UNIVERSITY OF READING

Numerical Techniques for Eigenstructure Assignment by Output Feedbac in Aircraft Applications

Darren M. Littleboy

This thesis is submitted for the degree of
Doctor of Philosophy

Department of Mathematics

December 1994

Abstract

In the past 30 years, techniques for eigenstructure assignment have been widely investigated and applied to many problems. By eigenstructure assignment we mean the use of feedback control in order to alter the eigenvalues and/or eigenvectors of a system. Eigenstructure assignment has been achieved using both state and output feedback.

This thesis is an investigation into the application of eigenstructure assignment to aircraft problems. We study the current work, illustrating that feedback is used to ensure stability, a satisfactory response and good decoupling in the closed loop system. A desired level of output decoupling is currently obtained by assigning a specified set of right eigenvectors; we identify a shortfall in current work that the corresponding left eigenvectors must also be considered to obtain a desired level of input decoupling. We give an example to demonstrate this.

We then present two minimisation routines that improve the level of input decoupling, while retaining the output decoupling. It is not generally possible to achieve the exact levels of input and output decoupling; our routines find the set of vectors anf throuteoallyetTc/Tinroecnotminimiiasyste.WT/aTc/thenall toorwinleobsetTc/

Contents

Chapter 1

Introd ction

In most applied mathematical researc

Ideally, we would like to be able to measure all of the variables, or states of a system in order to design a feedback. If this is the case, then we are performing state feedback. In practice, not all of the system states are available; the feedback then has to use the outputs to control the system. This is called output feedback.

In this thesis we are concerned with eigenstructure assignment by output feedback to assign simultaneously a set of eigenvalues and their corresponding right and left eigenvectors. We apply this theory to achieve the satisfactory handling qualities of an aircraft in flight. In the open-loop state, many aircraft are unstable, or display poor handling; hence feedback is required to force the aircraft to behave in the desired manner. The main considerations here are to improve stability, dampen unwanted oscillatory modes and to reduce any modal coupling. These qualities of the system can be observed and altered via investigation into the eigenstructure of the system.

In Chapter 2 we introduce the basic form for control systems and give their general governing equations. We describe their properties and introduce the concept of feedback. We then relate this theory to our interests, namely flight control systems. A review of the literature on eigenstructure assignment and its applications to aircraft problems is presented.

Having introduced our interest in aircraft flight control problems, we give in Chapter 3 the broad concept of aircraft control and how, physically, the aircraft is manoeuvred by either the pilot or feedback control. We describe the way in which the equations of motion are derived in the original non-linear form and are linearised and simplified into a usable state space form. We define the state representation of control systems, and give examples of state matrices for longitudinal and lateral motion. The concept of aircraft stability for both motions is introduced.

In Chapter 4 we give more detail on the theory and techniques of eigenstructure assignment, from the basics of pole placement by state feedback to the output feedback problem. The specific aircraft problem is introduced at the end of the chapter and the application of partial eigenstructure assignment to this problem is giv

addition to the right eigenvectors.

Having given the background theory on eigenstructure assignment and its application to aircraft problems in Chapter 4 we attempt, in Chapter 5, to improve on the results calculated in current work. A minimisation technique is developed that updates a set of vectors to improve the input decoupling (via the previously unconsidered left eigenvectors), while retaining the output decoupling already achieved. The updating vectors are restricted to lie in subspaces corresponding to a set of specified eigenvalues. We also control the robustness of the system via the condition number of the right eigenvectors and the accuracy of the assigned eigenvalues by minimising the error of the left eigenvectors from their correct subspaces. The result is a multi-criteria minimisation routine with weighting parameters that are altered in accordance with the design specifications. This routine is run with a number of parameter combinations to illustrate its flexibility.

In Chapter 6, to improve the results of Chapter 5, we remove the subspace restriction, allo

Chapter 2

Control systems

The main topic of this thesis is the application of eigenstructure assignment to aircraft problems. Before detailing both of these subjects we give some background theory on control systems and their properties. We define our areas of interest in the aircraft industry and give a review of the literature on eigenstructure assignment and its application to aircraft problems.

2.1 General control systems

In Chapter 1 we gave practical examples of what we mean by control systems; here we give a precise definition. An open loop control system, that is one which is not controlled automatically, can be represented as in Figure 2.1. Here the

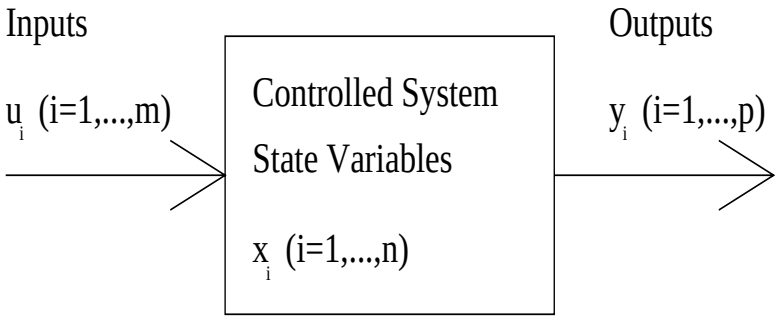


Figure 2.1: Open loop control system

state variables, x_i, describe the condition, or state, of the system, and provide the information which, together with a knowledge of the equations describing the system, enables us to calculate the future behaviour from a knowledge of the *input*

variables, u_i. Practically, it is often not possible to determine the values of the state variables directly, perhaps for reasons of expense or inaccessibility. Instead a set of *output variables*, y_i, which depend in some way on x_i, is measured.

The open loop equations describing the system in Figure 2.1 are

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t), \end{cases} \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ are the system matrices, known as the state, input and output matrices respectively. Practically, control systems are non-linear, but the set of non-linear differential equations can be linearised to give them in the form of (2.1). Also, in (2.1) the system matrices are dependent on time. For our work in this thesis we assume that the system matrices are constant coefficient matrices taken from the open loop control system at some specified operating points. Thus, we are working with linear, time-invariant systems. It is also assumed throughout this thesis that B and C are of full rank.

2.2 Pr perties

A system operating in its open loop state, as in Figure 2.1, has certain well defined properties. Before describing these we need a definition of the eigen-decomposition of a matrix.

Definition 2.1 *efine λ_i, $\mathbf{v}_i$ and $\mathbf{w}_i^T$ to be the eigenvalues and corresponding right and left eigenvectors, respectively, of A. They satisfy the relationships*

$$\begin{aligned} A\mathbf{v}_i &= \lambda\mathbf{v}_i \\ \mathbf{w}_i^T A &= \lambda\mathbf{w}_i^T. \end{aligned} \tag{2.2}$$

When $\mathbf{v}_i, \mathbf{w}_i^T$ are normalised appropriately then

$$V^{-1} = W^T, \tag{2.3}$$

where $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$, $W^T = [\mathbf{w}_1 \ldots, \mathbf{w}_n]^T$.

Definition 2.4

THEOREM 2.7 *A system is said to be completely observable if and only if one of the following equivalent conditions holds.*

$$(i) \quad rank \begin{bmatrix} C \\ CA \\ . \\ . \\ . \\ CA^{n-1} \end{bmatrix} = n$$

$$(ii) \quad rank \begin{bmatrix} A - \lambda I \\ C \end{bmatrix} = n \quad (\forall \lambda \in \mathbb{C})$$

$$(iii) \quad \{A\mathbf{s} = \mu \mathbf{s} \quad and \quad C\mathbf{s} = 0\} \quad \Rightarrow \ \mathbf{s} = 0.$$

Proof (see Barnett and Cameron [4]).

2.2.5 Robustness

Another important property of control systems is their robustness. This is defined in the sense that the eigenvalues of the system are as insensitive to perturbations as possible. If A in (2.1) is non-defective, then it is diagonalizable and it can be shown (Wilkinson [67]) that the sensitivity of the eigenvalue λ_i to perturbations in the components of A depends upon the magnitude of the *condition number*, c_i, where

$$c_i = \frac{\|\mathbf{w}_i^T\| \|\mathbf{v}_i\|}{|\mathbf{w}_i^T \mathbf{v}_i|} \geq 1. \tag{2.9}$$

A bound on the sensitivities of the eigenvalues is given by (Wilkinson [67])

$$\max_i c_i \leq \kappa_2(V) \equiv \|V\|_2 \|V^{-1}\|_2 \tag{2.10}$$

where $\kappa_2(V)$ is the *condition number* of the modal matrix of eigenvectors, $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$.

2.2.6 Feedback

We have given the general definition of a control system and its important properties. However, an open loop system may have poor properties in that it may

be unstable, or it may be very sensitive to perturbations. Thus, we want to con-

Thus, the aim of feedback, whether it is state or output feedback, is to control the system so that it behaves in a desired manner. We have shown that the properties of the systems are functions of its eigenvalues and eigenvectors, so we find a feedback such that the eigenstructure of the closed loop system results in, for example, that system being stable and robust. The details of this, including

overwrite only the 'bad' eigenvalues of a system.

In practice, state feedback is undesirable, not least because of the expense in measuring and feeding back all of the states. Indeed, all of the state measurements may not be available, so the more attractive procedure is to use the measured variables i.e. to perform output feedback.

One of the first to address pole placement by output feedback was Davison [11] who showed that if the system is controllable and if rank $(C) = p$, then a feedback can always be found so that p of the eigenvalues of the closed loop system are arbitrarily close to those desired. This result was extended by Davison and Chatterjee [13] and Sridhar and Lindhorff [62] who proved that if a system is controllable and observable and if rank (

and the assumptions that the matrices B and C are of full rank, then $max(m,p)$ closed loop eigenvalues can be assigned and $max(m,p)$ closed loop eigenvectors can be partially assigned with $min(m,p)$ entries in each vector arbitrarily chosen using output feedback.

Proof (see Srinathkumar [63]).

Attempts were made to assign the whole eigenstructure of a system such as by Porter and Bradshaw [47], [48] and Fletcher [20]. Necessary and sufficient conditions for a solution to exist were derived by Fletcher *et al.* [21], but were of a slightly abstract mathematical nature, not leading to a simple design technique. Conditions that full eigenstructure assignment was attainable when the right and left eigenvectors lie simultaneously in their correct subspaces were proved by Chu *et al.* [10], who used a least squares minimisation technique to solve the feedback design problem.

In a different direction, Roppenecker and O'Reilly [50] parameterised the problem, leading to the work of Fahmy and O'Reilly [18]. This extended the parametric state feedback work of Fahmy and O'Reilly [16], [17] and Fahmy and Tantawny [19]. It also extends the idea of pole protection from Topalogu and Seborg [64] to protecting the eigenvectors in addition. Owens [41] used this parameterisation idea to render a closed loop system eigenvalue totally insensitive by making its left eigenmode insensitive.

Whilst some authors attempted full eigenstructure assignment, others considered partial eigenstructure assignment. This idea arises from the fact that not all of the open loop eigenvalues of a system are necessarily considered undesirable. Usually, some eigenvalues will be acceptable and it is hence worth assigning some poles while retaining others. Fletcher *et al.* [21] set out necessary and sufficient conditions for assigning k eigenvalues, while retaining the other, original $n-k$ eigenvalues. These conditions were exploited by Slade [51], who devised an algorithm for assigning $m+p$ poles in two stages, while maximising the robustness of the solution.

Despite the numerous approaches, no one method has been adopted as standard. Indeed, the attempt by authors to assign poles exactly may not be as realistic a problem as assigning them to pre-specified regions. An approach to

formulated by Gilbert [23], is considered where the minimum sensitivity measure is attained if and only if all the closed loop eigenvectors are mutually orthogonal, thus affecting the authors' choice of desired eigenvectors. Similarly

The Davidon-Fletcher-Powell algorithm that requires first derivative information is used. Patel *et al.* [43] use the same method for multi-input, fixed output rate (MIFO) sampling schemes on a Stability Augmentation System of an aircraft.

A different approach was suggested by Wilson and Cloutier [68] who minimised a performance index constrained by the linear quadratic regulator algebraic Riccati equation. To constrain the eigenvalues, a Valentine transformation is employed to restrict them to be in some left-hand plane, but this is a very ill-conditioned transformation. The algorithm is implemented using the Ctrl-C software language with a conjugate gradient restoration algorithm but, as noted by the authors, has some drawbacks. For a third order system with two inputs, the number of optimising parameters is twenty-two, which slows down the performance; hence the authors adopt periodic preconditioning. The same authors [69] improve the previous work by replacing a highly nonlinear performance index with a quadratic one, at the expense of an increase in the nonlinearity and number of constraints. This is applied to the Extended Medium Range Air-to-Air Technology (EMRAAT) airframe. Wilson, Cloutier and Yedavalli [71] [70] extend their work to include time-varying parametric variations and also employ a Lyapunov constraint.

Most of these techniques use the conditioning of the modal matrix to control robustness, but another method is in considerationo

erations were then performed at each point and an optimisation to convergence on the five points with the smallest performance index was carried out. This was solved using a quadratic extended interior penalty function, and obviously requires a lot of computational effort. Here also the eigenvalue positions are rigid, but it does demonstrate the need for a robustness/performance trade-off. Sobel *et al.* [53] considered robust control for systems with structured, state space, norm bounded uncertainty, and extended this (see Sobel and Yu [60]) to add the constraint of restricting the eigenvalues to lie within chosen regions in the complex plane. This constrained optimisation problem was solved using the sequential unconstrained minimisation technique with a quadratic extended interior penalty function. This theory is applied to design a control for an EMRAAT missile in Yu *et al.* [73]. The structured uncertainty work is more comprehensively covered in Yu and Sobel [74], and includes a mention of considering robustness via the minimum singular value of the return difference matrix. The work is attempted in a slightly different way in Piou *et al.* [45] by constraining the problem with a Lyapunov condition and using the delta operator on a sampled data system. The various work of Sobel *et al.* over the last ten years in eigenstructure assignment for flight control system design is detailed in Sobel *et al.* [59] covering (constrained) output feedback, gain suppression, dynamic compensation, robust sampled data, pseudo-control, singular values for robustness and Lyapunov constraints; calculated using the Matlab Optimisation and Delta toolboxes ([25] and [34] respectively).

A study of the application of eigenstructure assignment to the control of powered lift combat aircraft was presented by Smith [52], most notable for the consideration of the left eigenvectors. Previous authors considered assigning only a set of right eigenvectors corresponding to a set of specified eigenvalues, for the purpose of obtaining modal output decoupling. In addition, Smith considered the assignment of a corresponding set of left eigenvectors to obtain some desired level of modal input decoupling. This is performed using a Simplex search method. This thesis considers the simultaneous assignment of right and left eigenvectors corresponding to specified eigenvalues to achiev

The most recent comprehensive study into multi-objective control design problems arising in aeronautics is by Magni and Mounan [32]. The theory is based on first order variations on the gains, eigenvalues, right and left eigenvectors, and their corresponding output and input directions. The problem is solved iteratively, utilising the Matlab Optimisation toolbox [25] whereby, at each iteration, a quadratic problem under linear equality constraints is solved. The constraints change step by step in such a way that the final step corresponds to the original eigenstructure assignment problem considered. Care has to be taken that the change in the set of constraints is small enough so that the first order approximations are valid. This method requires an interpretation of the results at each step to identify the objectives for the following step, for example identifying undesirable coupling or the slowest eigenvalue.

We have shown in this review how, in the last 27 years, the subject has progressed from the basis of changing the poles of a system b

In the next two chapters, we explain in detail the derivation of the aircraft equations of motion into a linear, time-invariant form as in (2.1), and develop the theory and techniques of eigenstructure assignment to be used in solving aircraft problems.

Chapter 3

Aircraft dynamics

This thesis is primarily concerned with the theory of eigenstructure assignment, with specific application to aircraft problems. For completeness, we describe how the equations of motion for an aircraft are derived. We give the linearisation and assumptions needed to transform these equations into a form to which eigenstructure assignment theory can be applied. This chapter closely follows parts of the first three chapters of McLean [33].

.1 Intr ducti n

Irrespective of the system being considered, we are interested in how effectively it can be controlled from an initial state to a desired final state within a certain time scale. The motion of a vehicle is characterised by its velocity vector, the control of the vehicle's path is dependent on physical constraints. For example a train is constrained by its track; cars must move over the surface of the earth, but both speed and direction are controlled. Aircraft differ as they have six degrees

This9bcbil'yvTijvAircr9Tc'vTj'ycum;rain

aircraft tend to fly in a constant turn; hence, to fly a straight and level course, continuous corrections must be made by a pilot, or by means of an *automatic ight control system* (AFCS). In aircraft, such AFCSs employ feedback to ensure:

1. the speed of response is better than at open-loop,
2. the accuracy in following commands is better,
3. the system is capable of suppressing unwanted effects arising from disturbances to the aircraft's flight.

However, the AFCS may have poor stability because such feedback systems have a tendency to oscillate. Thus, designers must employ a trade-off between the requirements for stability and control.

3.1.1 Control surfaces

If a body is to be changed from its present state of motion then external forces, or moments, or both, must be applied to the body, and the resulting acceleration vector can be determined by applying Newton's Second Law of Motion. Every aircraft has control surfaces which are used to generate the forces and moments required to produce the accelerations which cause the aircraft to be steered along its three-dimensional flightpath to its specified destination.

Conventional aircraft have three control surfaces; elevator, ailerons and rudder, with a fourth control available in the change of thrust obtained from the engines. Modern aircraft, particularly comIn'9999TD'9Tc'encnftfConTndcTD'der,am
ector

In addition to these surfaces, every aircraft has motion sensors to provide measures of change in motion variables which occur as the aircraft responds to the pilot's commands, or as it encounters some disturbance. These signals from the sensors can be used to provide the pilot with a visual display, or as feedback signals for the AFCS.

The flight controller compares the commanded motion with the measured motion and, if any discrepancy exists, generates, in accoTheresptondstcoceasuredT'9, iaircraftmra

Figure 3.1: Body axis system (see McLean [33])

principal axis system, the wind axis system and the stability axis system, which is the most frequently used.

3.2.1 Equations of motion of a rigid body aircraft

It is assumed that the aircraft considered is rigid-body, that is the distance between any two points on the aircraft's surface remain fixed in flight. Under this assumption the motion has six degrees of freedom. Newton's Second Law can be applied to obtain the equations of motion in terms of the translational and angular accelerations. It is also assumed that the inertial frame of reference does not accelerate, that is, the Earth is considered fixed in space.

Figure 3.1 is included to illust9TD'donotr9acceegru3.1nal9ermsmssurCggtheei

Translational motion

From Newton's Second Law it can be deduced that

$$\mathbf{F} = \mathbf{F}_0 + \Delta\mathbf{F} = m\frac{d}{dt}\{\mathbf{V}_T\} \tag{3.1}$$

$$\mathbf{M} = \mathbf{M}_0 + \Delta\mathbf{M} = \frac{d}{dt}\{\mathbf{H}\}, \tag{3.2}$$

where $\mathbf{F}$ represents the sum of all externally applied forces, $\mathbf{V}_T$ is the velocity vector, $\mathbf{M}$ represents the sum of all applied torques and $\mathbf{H}$ is the angular momentum. Also, m is the mass of the aircraft, assumed to be constant. It is convenient when analyzing AFCSs to regard $\mathbf{F}$ and $\mathbf{M}$ as consisting of an equilibrium component (denoted by 0) and a perturbational component (denoted by Δ).

By definition, equilibrium flight must be unaccelerated along a straight path; during this flight the linear velocity vector relative to fixed space is invariant, and the angular velocity is zero. Thus both $\mathbf{F}_0$ and $\mathbf{M}_0$ are zero. The rate of change of $\mathbf{V}_T$ relative to the Earth axis system is

$$\frac{d}{dt}\{\mathbf{V}_T\}_E = \left.\frac{d}{dt}\mathbf{V}_T\right|_B + \boldsymbol{\omega} \times \mathbf{V}_T, \tag{3.3}$$

where $\boldsymbol{\omega}$ is the angular velocity of the aircraft with respect to the fixed axis system. Expressing the vectors as the sums of their components with respect to $(X_B, Y_B, \quad_B)$ gives

$$\begin{aligned} \mathbf{V}_T &= U\mathbf{i} + V\mathbf{j} + W\mathbf{k} \\ \boldsymbol{\omega} &= P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}. \end{aligned} \tag{3.4}$$

Evaluating (3.3) using (3.4) gives

$$\begin{aligned} \Delta X \equiv \Delta F_x &= m(\dot{U} + QW - VR) \\ \Delta Y \equiv \Delta F_y &= m(\dot{V} + UR - PW) \\ \Delta \equiv \Delta F_z &= m(\dot{W} + VP - UQ), \end{aligned} \tag{3.5}$$

which are thus the equations of translational motion.

Rotational motion4)+4)De9mn9Tc')Tj'T99ToNFΔ+awith

with the inertia matrix, I, defined as

$$\begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}, \tag{3.7}$$

where I_{ii} denotes a moment of inertia, and I_{ij} a product of inertia for $j \neq i$. Using (3.6) in (3.2) gives

$$\mathbf{M} = \frac{d}{dt}\mathbf{H} + \boldsymbol{\omega} \times \mathbf{H}. \tag{3.8}$$

Transforming the body axes to the Earth axes system, and considering the individual components of $\mathbf{H}$ from (3.6), along with the fact that in general aircraft are symmetrical about the X plane (implying $I_{xy} = I_{yz} = 0$), results in

$$\begin{aligned} \Delta L \equiv \Delta M_x &= I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) + (I_{zz} - I_{yy})QR \\ \Delta M \equiv \Delta M_y &= I_{yy}\dot{Q} + I_{xz}(P^2 - R^2) + (I_{xx} - I_{zz})PR \\ \Delta N \equiv \Delta M_z &= I_{zz}\dot{R} - I_{xz}\dot{P} + PQ(I_{yy} - I_{xx}) + I_{xz}QR, \end{aligned} \tag{3.9}$$

where L, M, N are moments about the rolling, pitching and yawing axes respectively.

Forces due to gravity

The forces of gravity are always present in an aircraft; however, it can be assumed that gravity acts at the centre of gravity (c.g.) of the aircraft. But, since the centres of mass and gravity coincide in an aircraft, there is no external moment produced by gravity about the c.g.; this means gravity contributes only to the external force vector $\mathbf{F}$.

To resolve the forces, the gravity vector $m\mathbf{g}$ is directed along the $_E$ axis, is the angle between the gravity vector and the Y_B $_B$ plane and Φ is the bank angle between the $_B$ axis and the projection of the gravity vector on the Y_B $_B$ plane. Direct resolution of $m\mathbf{g}$ into its $X, Y,$ components produces

$$\begin{aligned} \delta X &= - \quad m\mathbf{g} \sin \\ \delta Y &= \quad m\mathbf{g} \cos \quad \sin\Phi \\ \delta &= \quad m\mathbf{g} \cos \quad \cos\Phi. \end{aligned} \tag{3.10}$$

The manner in which the angular orientation and velocity of the body axis system with respect to the gravity vector is expressed depends upon the angular velocity

of the body axis about $m\mathbf{g}$. This angular velocity is the azimuth rate, $\dot{\Psi}$; it is not normal to $\dot{\Phi}$ or $\dot{\ }$, but its projection in the Y_B $_B$ plane is normal to both. By resolution

$$\begin{aligned} P &= \dot{\Phi} - \dot{\Psi} \sin \\ Q &= \dot{\ } \cos\Phi + \dot{\Psi} \cos \quad \sin\Phi \\ R &= -\dot{\ } \sin\Phi + \dot{\Psi} \cos \quad \cos\Phi, \end{aligned} \tag{3.11}$$

where Φ, , Ψ are referred to as the Euler angles.

Linearisation of the inertial and gravitational terms

Equations (3.5) and (3.9) represent the inertial forces acting on the aircraft. Equation (3.10) represents the contribution of the forces due to gravity to these equations. The external forces acting on the aircraft can be re-expressed as

$$\begin{aligned} X &= \Delta X + \delta X \\ Y &= \Delta Y + \delta Y \\ &= \Delta \quad + \delta \ , \end{aligned} \tag{3.12}$$

where the δ terms are gravitational and the Δ terms represent the aerodynamic and thrust forces. For notational convenience, ΔL, ΔM and ΔN are denoted by L, M and N; thus the equations of motion of the rigid body for its six degrees of freedom are

$$\begin{aligned} X &= m(\dot{U} + QW - VR + \mathbf{g} \sin \ ) \\ Y &= m(\dot{V} + UR - PW - \mathbf{g} \cos \quad \sin\Phi) \\ &= m(\dot{W} + VP - UQ - \mathbf{g} \cos \quad \cos\Phi) \\ L &= I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) + (I_{zz} - I_{yy})QR \\ M &= I_{yy}\dot{Q} + I_{xz}(P^2 - R^2) + (I_{xx} - I_{zz})PR \\ N &= I_{zz}\dot{R} - I_{xz}\dot{P} + PQ(I_{yy} - I_{xx}) + I_{xz}QR. \end{aligned} \tag{3.13}$$

Note that (3.11) must also be used since those equations relate Ψ, , Φ to R, Q, P. The equations in (3.13) are highly non-linear and are simplified by considering the motion in two parts: a mean motion to represent the equilibrium (or trim) conditions, and a dynamic motion for the perturbations to the mean motion. Thus, every motion variable is considered to have two components. For example

$$\begin{aligned} U &\equiv U_0 + u \qquad R \equiv R_0 + r \\ Q &\equiv Q_0 + q \qquad M \equiv M_0 + m_1 \ \text{ etc.}, \end{aligned} \tag{3.14}$$

where 0 denotes trim and lower-case letters are the perturbations. Note that m_1 is the perturbation in the pitching moment, M, not to be confused with the mass m in (3.13), which is considered constant. In trim there is no acceleration so we can obtain equations for $X_0, Y_0, \ {}_0, L_0, M_0, N_0$ that are just the equations in (3.13) with the $\dot{U}, \dot{V}, \dot{W}, \dot{P}, \dot{Q}, \dot{R}$ terms all set to zero; all other components (except m and $\mathbf{g}$) have the subscript 0. The perturbed motion can then be found by substituting (3.14) into (3.13) and subtracting the equations for X_0, Y_0 etc. Assuming small perturbations, sines and cosines are approximated to the angles themselves and unity respectively; products of perturbed quantities are deemed negligible. The perturbed equations of motion that result are simpler than (3.13), but are still not readily usable. Common practice in AFCS studies is to consider flight cases with simpler trim conditions; flying straight in steady, symmetric flight with wings level is an example commonly used. These assumed trim conditions have the implications

1. straight flight implies $\dot{\Psi} = \ {}_0 = 0$,
2. symmetric flight implies $\Psi_0 = V_0 = 0$,
3. flying with wings level implies $\Phi_0 = 0$.

Under these conditions it may also be assumed that $Q_0 = P_0 = R_0 = 0$, giving the simple equations

$$\begin{aligned} x &= m[\dot{u} + W_0 q + (\mathbf{g} \cos \ {}_0)\theta] \\ z &= m[\dot{w} - U_0 q + (\mathbf{g} \sin \ {}_0)\theta] \\ m_1 &= I_{yy}\dot{q} \end{aligned} \tag{3.15}$$

and

$$\begin{aligned} y &= m[\dot{v} + U_0 r - W_0 p - (\mathbf{g} \cos \ {}_0) \ ] \\ l &= I_{xx}\dot{p} - I_{xz}\dot{r} \\ n &= I_{zz}\dot{r} - I_{xz}\dot{p}. \end{aligned} \tag{3.16}$$

Here the equations in (3.15) represent the longitudinal motion, and those in (3.16) represent the lateral/directional motion (sideslip, rolling and yawing motion specifically). This separation is merely a separation of gravitational and inertial forces, only possible because of the assumed trim conditions.

3.2.2 Complete linearised equations of motion

To expand the left-hand side of the equations of motion, a Taylor series expansion is used about the trimmed flight condition. For example, if only elevator deflection is involved in the aircraft's longitudinal motion then the first equation in (3.15) becomes

$$\begin{aligned} x = \tfrac{\partial X}{\partial u}u + \tfrac{\partial X}{\partial \dot{u}}\dot{u} + \tfrac{\partial X}{\partial w}w + \tfrac{\partial X}{\partial \dot{w}}\dot{w} + \tfrac{\partial X}{\partial q}q + \tfrac{\partial X}{\partial \dot{q}}\dot{q} + \tfrac{\partial X}{\partial \delta_E}\delta_E + \tfrac{\partial X}{\partial \dot{\delta}_E}\dot{\delta}_E \\ = m[\dot{u} + W_0 q + (\mathbf{g}\cos\ {}_0)\theta] \end{aligned} \tag{3.17}$$

and similarly for the other equations in (3.15) and (3.16). Note that here δ_E is the deflection to the elevators. If any other control surface on the aircraft being considered were involved, additional terms would be involved. For example, if deflection of flaps (F) and symmetrical spoilers (sp) were also used as controls for longitudinal motion, additional terms such as

$$\tfrac{\partial X}{\partial \delta_F}\delta_F \quad \text{and} \quad \tfrac{\partial X}{\partial \delta_{sp}}\delta_{sp} \tag{3.18}$$

would be added to equation (3.17). For simplification, we define

$$\begin{aligned} X_x &\equiv \tfrac{1}{m}\tfrac{\partial X}{\partial x} \\ {}_x &\equiv \tfrac{1}{m}\tfrac{\partial Z}{\partial x} \\ M_x &\equiv \tfrac{1}{I_{yy}}\tfrac{\partial M}{\partial x} \end{aligned} \tag{3.19}$$

and M_x, ${}_x$, X_x are called *stability derivatives.*

Equations of longitudinal motion

If the equations in (3.15) are expanded (as in (3.17)) and the substitutions in (3.19) are made, then there results a new set of equations. From the study of aerodynamic data, it becomes evident that some stability derivatives can be neglected (but this is problem dependent). The equations of perturbed longitudinal motion, for straight, symmetric flight, with wings level can be expressed as

$$\begin{aligned} \dot{u} &= X_u u + X_w w - W_0 q - (\mathbf{g}\cos\ {}_0)\theta + X_{\delta_E}\delta_E \\ \dot{w} &= {}_u u + {}_w w + U_0 q - (\mathbf{g}\sin\ {}_0)\theta + {}_{\delta_E}\delta_E \\ \dot{q} &= M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta_E}\delta_E \\ \dot{\theta} &= q, \end{aligned} \tag{3.20}$$

where $\dot{\theta} = q$ is usually added for completeness.

Equations of lateral motion

As in Section (3.2.2), w

effects and taking into account the cross-product of inertia terms gives

$$\begin{aligned}
\dot{v} &= Y_v v - U_0 r + (\mathbf{g}\ \cos\gamma_0) \quad + Y_{\delta_R}\delta_R \\
\dot{p} &= L'_v v + L'_p p + L'_r r + L'_{\delta_A}\delta_A + L'_{\delta_R}\delta_R \\
\dot{r} &= N'_v v + N'_p p + N'_r r + N'_{\delta_A}\delta_A + N'_{\delta_R}\delta_R \\
\dot{} &= p + r\ \tan\gamma_0 \\
\dot{\psi} &= r\ \sec\gamma_0.
\end{aligned} \tag{3.24}$$

. State space representati n

In Chapter 2 we introduced basic control systems, governed by the equations

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t), \end{cases} \tag{3.25}$$

where $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ are vectors representing the state, input and output variables respectively. We also defined our interest as being in linear, time-invariant systems. The flight of an aircraft, however, is time-varying and its equations are non-linear. In Section 3.1.3 we explained the concept of gain scheduling so that the system matrices, A, B and C, may be considered constant at set operating conditions relative to some parameter.

In the previous section we showed how the non-linear aircraft equations can be linearised into a relatively simple form, so that they can be represented in state space form.

We now illustrate how the simplified aircraft equations of motion are represented in the form of (3.25); this is done for both longitudinal and lateral motion.

3.3.1 ircraft equations of longitudinal motion

If the state vector for an aircraft is defined as

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix}, \tag{3.26}$$

where the variables are the perturbations to forward velocity, yaw velocity, pitch rate and pitch angle, and if the aircraft is being controlled by means of elevator

deflection, δ_E, and change of thrust, δ_{th}, then from (3.23) the state equation is defined as

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -\mathbf{g}cos\gamma_0 \\ {}_u & {}_w & U_0 & -\mathbf{g}sin\gamma_0 \\ & & & \\ & & & \end{bmatrix}$$

3.3.2 ircraft equations of lateral motion

If the state vector is

$$\begin{bmatrix} \beta \\ p \\ r \\ \\ \psi \end{bmatrix}, \tag{3.32}$$

where the variables are defined as sideslip angle, the perturbations to roll rate and yaw rate, and roll angle and yaw angle; and if the aircraft is being controlled by the ailerons, δ_A, and rudder, δ_R, then the state equation is

$$\begin{bmatrix}
\end{bmatrix}$$

3.4.1 Longitudinal stability

The characteristic polynomial of the state coefficient matrix A, known as the stability quartic, is calculated in the form

$$\lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0. \qquad (3.(3.[$$

which can usually be factorised into the following form

$$\lambda(\lambda + \mathrm{e})(\lambda + \mathrm{f})(\lambda^2 + 2\xi_D\omega_D\lambda + \omega_D^2) = 0. \tag{3.39}$$

The simple term in λ corresponds to the heading (directional) mode. Because $\lambda = 0$ is a root, then once an aircraft's heading has been changed, there is no

Chapter 4

Eigenstructure assignment

In Chapter 2 w

A solution to Problem 1 was given by Wonham [72], and gave the link between complete controllability and eigenvalue assignment.

THEOREM 4.1 *A system is controllable if and only if, for every self-conjugate set of scalars* $\{\lambda_1, \ldots, \lambda_n\}$*, there exists a real matrix* $F \in \mathbb{R}^{m \times n}$ *such that the eigenvalues of* $A + BF$ *are* λ_i $(i = 1, \ldots, n)$*.*

Proof (see Wonham [72]).

Following Wonham's paper [72], many authors published work on the sub-

where $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$, $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_n\}$ such that some measure of the conditioning, or robustness, is optimised.

Kautsky *et al.* [28] gave a number of measures that can be considered as an optimisation objective of Problem 2. We shall now show how to construct a feedback to solve this problem.

4.1.1 Construction of a state feedback

Given that we have a V that optimises some robustness measure, the following theorem gives a construction of F.

THEOREM 4.2 *Given $\Lambda = diag\{\lambda_1, \ldots, \lambda_n\}$ and V non-singular, then there exists a real $F \in \mathbb{R}^{m \times n}$, a solution to (4.3) if and only if*

$$U_1^T(AV - V\Lambda) = 0, \tag{4.4}$$

where

$$B = [U_0, U_1] \begin{bmatrix} \quad B \\ 0 \end{bmatrix}, \tag{4.5}$$

with $U = [U_0, U_1]$ orthogonal and B non-singular. Then F is given explicitly by

$$F = \bar{B}^{-1} U_0^T (V\Lambda V^{-1} - A). \tag{4.6}$$

Proof (see Kautsky *et al.* [28])

As a result of this theorem we have the following corollary.

orollary 4.3 *A matrix V may be chosen to satisfy Problem 2 if we select each column $\mathbf{v}_i$ of V, corresponding to each desired eigenvalue λ_i, so that it belongs to the null space*

$$\mathcal{S}_i \equiv \mathcal{N}[U_1^T(A - \lambda_i I)]. \tag{4.7}$$

Proof Follows directly from Theorem 4.2 (see Kautsky *et al.* [28]).

So, if we choose to assign a set of distinct eigenvalues, for each i, a vector $\mathbf{v}_i$ can be chosen from $\mathcal{S}_i$ to form V non-singular and as robust as possible. Three iterative methods for this are given in Kautsky *et al.* [28]. We have shown how we may construct a feedback to obtain a system that is stable and robust. We also

illustrated in Section 2.2.1 that this feedback can be used to affect the transient response of the system.

The construction of F has used the fact that all of the state variables are available for feedback. However, this is not generally the case and we need to extend our ideas to output feedback.

4.2 Output feedback

It has been illustrated that we may use feedback to alter the eigenstructure of a system for three purposes: to ensure stability, robustness and a satisfactory response. But all of this has been performed using the state variables. In practice these will not all be available for feedback; instead we may use the measured variables, that is, the outputs. Output feedback by eigenstructure assignment is a much more difficult problem than for state feedback, and our objectives are:

Problem 3 Given the real triple (A, B, C) and a self-conjugate set of scalars $\{\lambda_1, \ldots, \lambda_n\}$ and a corresponding self-conjugate set of n-dimensional vectors, $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, find a matrix $K \in \mathbb{R}^{m \times p}$ such that the eigenvalues of $A + BKC$ are λ_i $(i = 1, \ldots, n)$, with corresponding eigenvectors $\mathbf{v}_i$ $(i = 1, \ldots, n)$. i.e. that

$$(A + BKC)V = V\Lambda, \tag{4.8}$$

where $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$, $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_n\}$.

4.2.1 Construction of an output feedback

Without any dimensional restrictions on Problem 3, an output feedback can be constructed from Chu *et al.* [10].

THEOREM 4.4 *Given Λ and V non-singular, then there exists a real $K \in \mathbb{R}^{m \times p}$, a solution satisfying (4.8), if and only if*

$$U_1^T(AV - V\Lambda) = 0 \tag{4.9}$$

$$(V^{-1}A - \Lambda V^{-1})P_1 = 0, \tag{4.10}$$

where

$$B = [U_0, U_1] \begin{bmatrix} \quad _B \\ 0 \end{bmatrix}, C = [\ _C^T, 0] \begin{bmatrix} P_0^T \\ P_1^T \end{bmatrix}, \tag{4.11}$$

with $U = [U_0, U_1], P = [P_0, P_1]$ *orthogonal and* $_B$, $_C$ *non-singular. Then* K *is given explicitly by*

$$K = \ _B^{-1} U_0^T (V \Lambda V^{-1} - A) P_0 \ _C^{-T} \tag{4.12}$$

Proof The existence of decompositions (4.11) follows from the assumption that B and C are of full rank. From (4.8), K must satisfy

$$BKC$$

Definition 4.6 *(Chu et al. [10]) : Perform QR decompositions on the right and left eigenvector spaces given in (4.15),(4.16) such that*

$$[U_1^T(A - \lambda_i I)]^T = [\hat{S}_i, S_i] \begin{bmatrix} R_{Ri} \\ 0 \end{bmatrix} \qquad (4$$

since $V^{-1} = W^T$ from (2.3), but $\mathbf{v}_i \in \mathcal{S}_i$ i.e. $U_1^T(A - \lambda_i I)\mathbf{v}_i = 0$ so that

$$U_1 U_1^T(AV - V\Lambda) = (I - U_0 U_0^T)(AV - V\Lambda) = 0, \tag{4.24}$$

using (4.21), giving $(AV - V\Lambda) = U_0 U_0^T(AV - V\Lambda)$. Using this and (4.22) results in

$$BKC = (V\Lambda W^T - A)(I - P_1 P_1^T).\square \tag{4.25}$$

The theorem associated with the error involved in assigning the desired eigenvalues is

THEOREM 4.8 *(Chu et al. [10]) Given* $\Lambda = diag\{\lambda_1, \ldots, \lambda_n\}$ *and* $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ *non-singular, such that* $\mathbf{v}_i \in \mathcal{S}_i$ *and* $\|\mathbf{v}_i\|_2^2 = 1$, *then* K *defined by (4.12) implies*

$$(A + BKC)V - V\Lambda = -EV. \tag{4.26}$$

Proof Using (4.25) from Lemma (4.7) we obtain $BKC = (V\Lambda W^T - A) - (V\Lambda W^T - A)P_1 P_1^T$ so that

$$(A + BKC)V - V\Lambda = -(V\Lambda W^T - A)P_1 P_1^T V,$$

TiSfmpliollo

corresponding eigenvectors are uncontrolled and may therefore force the system to displa

4.3.1 ircraft control problem

The importance of considering the whole eigenstructure of a system has been demonstrated previously. As shown in Section 2.2.1, the transient response depends on both the eigenvalues and corresponding right and left eigenvectors (see (2.6)), these eigenvectors being chosen for performance requirements depending on the application being considered. For the aircraft application considered here, the eigenvectors are chosen explicitly to improve the aircraft's flight handling qualities. This chapter considers the problems of

1. characterising eigenvectors which can be assigned as closed loop vectors and
2. determining the best possible set of assigned closed loop vectors in case the desired set is not assignable (since arbitrary eigenvector assignment is not, in general, possible).

Before explaining the theoretical aspects of calculating the eigenvectors, a definition is required.

Definition 4.11 *A^+ is defined to be the unique matrix, $X \in \mathbb{R}^{n\times n}$ that satisfies the four Moore-Penrose conditions:*

$$\begin{array}{llll} (i) & AXA = A, & (iii) & (AX)^H = AX \\ (ii) & XAX = X, & (iv) & (XA)^H = XA \end{array}$$

and is the unique minimal F-norm solution to

$$\min_{X\in\mathbb{R}^{n\times n}} \|AX - I_m\|_F. \tag{4.35}$$

Note that here the H superscript is the complex conjugate transpose.

4.3.2 Complete specification of desired eigenvectors

For an assigned eigenvalue, λ_i, it has been sho

where the 'a' subscript denotes that the vector is achievable. The problem arises that, in general, a desired eigenvector, $\mathbf{v}_{di}$, chosen from a performance criteria will not reside in the prescribed subspace and hence cannot be achieved. Instead a 'best possible' choice is made by projecting $\mathbf{v}_{di}$ into the subspace of achievable vectors, $\mathcal{S}_i$, shown geometrically in Figure 4.1

each projected into $\mathcal{S}_i$. The set of achievable vectors are augmented in the form $V = [\mathbf{v}_{a1}, \ldots, \mathbf{v}_{ap}]$, and the feedback is constructed as in (4.32),

$$K = \bar{B}^{-1} U_0^T (V\Lambda - AV)(CV)^{-1}. \tag{4.41}$$

4.3.3 Partial specification of desired eigenvectors

In many practical situations, complete specification of $\mathbf{v}_{di}$ is neither required nor known but rather the designer is interested only in certain elements of the eigenvector. This case is considered by assuming the eigenvector has the form

$$\mathbf{v}_{di} = [v_{i1}, \ldots, x, v_{ij}, \ldots, x, v_{in}]^T, \tag{4.42}$$

where v_{ij} are designer specified components (usually a 0 or 1) and x is an unspecified component. The number of elements that can be specified in each eigenvector is outlined in a theorem from the paper by Srinathkumar [63], who showed that $\min(m, p)$ entries in each vector can be arbitrarily chosen. If there is the need to specify more than $\min(m, p)$ entries for a performance requirement, then the vector is projected as before to calculate the best least squares fit.

To proceed with the analysis, a permutation matrix, P, is defined so that

$$P\mathbf{v}_{di} = \begin{bmatrix} \mathbf{d}_i \\ \mathbf{u}_i \end{bmatrix}, \quad PS_i = \begin{bmatrix} D_i \\ U_i \end{bmatrix}, \tag{4.43}$$

where $\mathbf{d}_i$

If we consider the linearised perturbed lateral axis equations, the state vector may be

$$\mathbf{x} = \begin{bmatrix} r \\ \beta \\ p \\ \psi \\ \delta_r \\ \delta_a \end{bmatrix} \begin{matrix} \text{yaw rate} \\ \text{sideslip angle} \\ \text{roll rate} \\ \text{bank angle} \\ \text{rudder deflection} \\ \text{aileron deflection} \end{matrix} . \tag{4.47}$$

Again this differs to the state vector in Section 3.3.2, as the choice is problem and aircraft type dependent. A desirable choice of right eigenvectors would be

$$\begin{bmatrix} 1 \\ x \\ 0 \\ 0 \\ x \\ x \end{bmatrix} \quad \begin{bmatrix} x \\ 1 \\ 0 \\ 0 \\ x \\ x \end{bmatrix} , \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ x \\ x \\ x \end{bmatrix} . \tag{4.48}$$

The first two vectors are the dutch roll vectors in which the yaw rate and sideslip angle are coupled while roll rate and bank angle are suppressed. The third vector is the roll subsidence vector where roll rate (and hence bank angle) are emphasised while yaw rate and sideslip are set to zero. The effect of these choices is to obtain an orthogonality of the subvector composed of the first four components of the dutch roll vectors with respect to the appropriate subvector from roll subsidence.

4.3.5 Mode output input coupling vectors

We have shown how to choose the eigenvectors both mathematically and practically, these being used to construct the feedback as in (4.32). However, there are quantities that can be specified by the designer that give more important information about the aircraft's performance than the eigenvectors alone. Writing (2.6) in terms of the output equation in (2.1) gives

$$\mathbf{y}(t) = \sum_{i=1}^{n} (C\mathbf{v} \qquad \text{y}$$

from which it can be seen that the term $C\mathbf{v}_i$ determines the outputs participating in the response of each mode, and that $\mathbf{w}_i^T B$ determines those modes that are affected by each input. We therefore define the *mode output coupling vectors* and *mode input coupling vectors* to be

$$\begin{aligned} G_0 &= CV \\ G_1 &= W^T B, \end{aligned} \tag{4.50}$$

respectively. Again the required G_0 may not lie in the correct spaces and we consider the achievability criteria as in the previous section. There is significant advantage in considering the assignment in terms of the output variables rather than the state variables. This was originally proposed by Moore [36]. Again we consider a desirable mode output coupling vector in the form

$$\mathbf{g}_{di} = [g_{i1}, \ldots, x, g_{ij}, \ldots, x, g_{ip}]^T, \tag{4.51}$$

which is rearranged along with CS_i,

$$\mathbf{g}_{di} = \begin{bmatrix} \mathbf{d}_i \\ \mathbf{u}_i \end{bmatrix}, \quad CS_i = \begin{bmatrix} D_i \\ U_i \end{bmatrix}, \tag{4.52}$$

into specified components $(\mathbf{d}_i, D_i)$ and unspecified components $(\mathbf{u}_i, U_i)$, respectively. Using (4.36) gives

$$\mathbf{g}_{ai} = C\mathbf{v}_{ai} = CS_i\boldsymbol{\eta}_i, \tag{4.53}$$

so that here the functional to be minimised is

$$J_2 \equiv \|\mathbf{g}_{di} - \mathbf{g}_{ai}\|^2 = \left\| \begin{bmatrix} \mathbf{d}_i \\ \mathbf{u}_i \end{bmatrix} - CS_i\boldsymbol{\eta}_i \right\|_2^2 = \left\| \begin{bmatrix} \mathbf{d}_i \\ \mathbf{u}_i \end{bmatrix} - \begin{bmatrix} D_i \\ U_i \end{bmatrix} \boldsymbol{\eta}_i \right\|_2^2. \tag{4.54}$$

Minimising this over the desired components to find $\boldsymbol{\eta}_i$ gives

$$\mathbf{g}_{ai} = CS_iD_i^+\mathbf{d}_i. \tag{4.55}$$

Noting that $\mathbf{g}_{ai} = C\mathbf{v}_{ai}$ we can construct

$$\mathbf{v}_{ai} = S_iD_i^+\mathbf{d}_i. \tag{4.56}$$

From here $G_0 = [\mathbf{g}_{a1}, \ldots, \mathbf{g}_{ap}], V = [\mathbf{v}_{a1}, \ldots, \mathbf{v}_{ap}]$ are constructed and the feedback (4.32) becomes

$$K = \bar{B}^{-1}U_0^T(V\Lambda - AV)G_0^{-1}, \tag{4.57}$$

provided G_0 is invertible.

4.3.6 Example of coupling vectors interaction

In Section 4.3.4 we gave examples of right eigenvectors that could be selected; from these the projection as in Section 4.3.3 can be performed and K calculated. However, we have just illustrated that modal coupling can be better observed in the mode output and mode input coupling vectors, so we now give examples of these and how to interpret the coupling. Consider a system of dimensions $n = 5, m = 2, p = 3$; example mode input and output coupling vectors are

$$G_{1d} = W_1^T B = \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \overset{Inputs(j)}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x & x \\ 1 & 0 \\ x & x \end{bmatrix}} Modes(i), \tag{4.58}$$

where the i^{th} mode is excited by the j^{th} input according to the $(W^T B)_{i,j}$ element and

$$G_{0d} = CV_1 = \begin{matrix} \\ \\ \\ \end{matrix} \overset{Modes(i)}{\begin{bmatrix} 1 & 0 & 0 & x & x \\ 0 & 1 & 0 & x & x \\ x & x & 1 & x & x \end{bmatrix}} Outputs(k), \tag{4.59}$$

where the k^{th} output depends on the i^{th} mode according to the $(CV)_{k,i}$

1. specify the system matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$

2. specify design requirements, $\Lambda_p \in \mathbb{C}^{p\times p}$, $G_{0d} \in \mathbb{R}^{p\times p}$, $G_{1d} \in \mathbb{R}^{p\times m}$

3. construct loop to calculate achievable mode output coupling vectors

   for $i = 1 : p$

   - calculate re-ordering operator, P, such that $P\mathbf{g}_{di} = \begin{bmatrix} \mathbf{d}_i \\ \mathbf{u}_i \end{bmatrix}$
   - calculate S_i, the null space of $U_1^T(A - \lambda_i I)$
   - use the re-ordering operator so that $P(CS_i) = \begin{bmatrix} D_i \\ U_i \end{bmatrix}$
   - calculate best achievable mode output coupling vector $\mathbf{g}_{ai} = CS_iD_i^+\mathbf{d}_i$
   - calculate corresponding eigenvector $\mathbf{v}_{ai} = S_iD_i^+\mathbf{d}_i$
   - augment $G_{0a} = [\mathbf{g}_{a1}, \ldots, \mathbf{g}_{ap}]$, $V_a = [\mathbf{v}_{a1}, \ldots, \mathbf{v}_{ap}]$

   end

4. calculate feedback gain matrix

$$K = B^+(V_a\Lambda_p - AV_a)G_{0a}^{-1} \tag{4.60}$$

5. calculate closed loop system $A + BKC$, check $\Lambda_p \subset \lambda(A + BKC)$ and calculate the errors in the mode output and input coupling vectors

$$E_1 = \|G_{0d} - G_{0a}\|_F^2, \quad E_2 = \|G_{1d} - G_{1a}\|_F^2. \tag{4.61}$$

4.4 Example

The example used here to illustrate the preceding theory on eigenstructure assignment by output feedback and its application to aircraft problems is a model of an L−1011 aircraft at cruise condition from [1]. We do not give the system matrices here, or indeed the open loop behaviour as this is all covered in Example 1 of Chapter 8. Also, we do not justify the choice of desired eigenstructure. This example is used to illustrate the achievable results and to demonstrate the shortfall in these results.

F

These are normalised so that the largest element (in modulus) in each column is one, giving

$$G_{0a} = \begin{bmatrix} 1 & -0.0009 \pm 0.0024i \\ 0 & 1 \\ 0.1265 \pm 0.0235i & -0.0036 \pm 0.0051i \\ 0 & -0.2000 \pm 0.4000i \end{bmatrix}. \tag{4.67}$$

We can see that the exact desired decoupling cannot be achieved in the roll mode, although the level of coupling is small. The results given here are those usually obtained by authors investigating eigenstructure assignment applications to aircraft control. Here we are also concerned with the left eigenvectors via the mode input coupling vectors, calculated here as

$$G_{1a} = \begin{bmatrix} -1.0470 - 2.5088i & -0.0034 + 0.0209i \\ -1.0470 + 2.5088i & -0.0034 - 0.0209i \\ -0.0207 - 0.1107i & 0.0264 + 0.3023i \\ -0.0207 + 0.1107i & 0.0264 - 0.3023i \end{bmatrix}. \tag{4.68}$$

Again, to view the level of coupling, we normalise each row in G_{1a} such that the largest element (in modulus) is one, giving

$$G_{1a} = \begin{bmatrix} 1 & -0.0066 - 0.0041i \\ 1 & -0.0066 + 0.0041i \\ -0.3693 + 0.0363i & 1 \\ -0.3693 - 0.0363i & 1 \end{bmatrix}. \tag{4.69}$$

We can see that the mode output coupling vectors, G_{0a} have been achieved to

4.5 Summary

This chapter has detailed the theory on the technique of eigenstructure assignment. We have given a state feedback construction for eigenstructure assignment, but noted that the states are not always available. Hence we introduced output feedback, but as a harder problem, and have shown that full eigenstructure assignment is not, in general, possible. However, partial eigenstructure assignment can be performed and has been successfully applied to aircraft problems, as illustrated in the example in Section 4.4. We noted though, that the unassigned eigenvalues and eigenvectors may have the consequence that the closed loop system displays poor behaviour. We have highlighted the fact that the assignment ectors m,hosHT4

Chapter 5

Restricted minimisation algorithm

sired eigenvalue, and hence they cannot be considered independen

5.2 Structure of right eigenvector matrix, V

After partial eigenstructure assignment is performed, the columns of V are normalised such that $\|\mathbf{v}_i\|_2^2 = 1$ $(i = 1, \ldots, n)$. In order to be able to see analytically how to update each vector in V_2, partition V in the form

$$\begin{aligned} V = [V_{\_}, \mathbf{v}_n] &= \left[[Q_1, \mathbf{q}] \begin{bmatrix} R \\ \mathbf{0}^T \end{bmatrix}, \mathbf{v}_n \right] \\ &= [Q_1, \mathbf{q}] \begin{bmatrix} R & Q_1^T \mathbf{v}_n \\ \mathbf{0}^T & \mathbf{q}^T \mathbf{v}_n \end{bmatrix}, \end{aligned} \tag{5.4}$$

where a QR decomposition has been performed on $V_{\_}$. This partitioning is based on the form in Method 1 of Kautsky *et al.* [28]. It is then possible to write down the inverse of V explicitly,

$$V^{-1} = \begin{bmatrix} R^{-1} & -\rho R^{-1} Q_1^T \mathbf{v}_n \\ \mathbf{0}^T & \rho \end{bmatrix} \begin{bmatrix} Q_1^T \\ \mathbf{q}^T \end{bmatrix}, \tag{5.5}$$

where $\rho = \frac{1}{\mathbf{q}^T \mathbf{v}_n}$ and is a scalar. Here $\mathbf{v}_n$ is the last column of V_2 and is the vector to be updated to satisfy set minimisation objectives. After it has been updated, it is moved to the front of V_2 so that $\mathbf{v}_{n-1}$ is the next vector to be updated, i.e.

$$\tilde{V}_2 = [\tilde{\mathbf{v}}_n, \mathbf{v}_{p+1}, \ldots, \mathbf{v}_{n-1}]. \tag{5.6}$$

This process is continued on $\mathbf{v}_{n-1}$, $\mathbf{v}_{n-2}$ etc., and we thus have a procedure for choosing a new set of $\tilde{V}_2$ by performing a rank-one update at each iteration. We shall now define three criteria to be satisfied when performing the minimisation algorithm.

5. Left eigenvector matching

The primary aim of the minimisation algorithm is to reduce the error between the desired and achieved mode input coupling vectors. Thus, we aim to solve the problem

$$\begin{cases} \min J_1 \equiv \min \|G_{1d} - G_{1a}\|_F^2 \\ \\ \text{subject to } \|\mathbf{v}_i\|_2^2 = 1 \ (i = 1, \ldots, n), \end{cases} \tag{5.7}$$

where the subscripts 'd' and 'a' denote desired and achieved quantities, respectively. From here onwards w

where $F_p = G_{1d} - D_p$, and the subscript p denotes that the subscripted matrix has been pre-multiplied by $[I_p, 0]$. It is not apparent how $\boldsymbol{\eta}_n$ can be chosen as a solution to (5.13); a Lemma is needed to make things simpler.

LEMMA 5.1 *(Kautsky and Nichols [27]) For suitably sized matrices A,B and vectors* $\mathbf{v}$,$\mathbf{w} \neq 0$ *it can be shown that*

$$\|A + B\mathbf{w}\mathbf{v}^T\|_F^2 = (\mathbf{v}^T\mathbf{v})\|A\hat{\mathbf{v}} + B\mathbf{w}\|_2^2 + \sum_{i=1}^{p} \alpha_i \tag{5.14}$$

where

$$\hat{\mathbf{v}} = \tfrac{\mathbf{v}}{(\mathbf{v}^T\mathbf{v})} \quad , \quad \alpha_i = \mathbf{e}_i^T A(I - \tfrac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}})A^T\mathbf{e}_i \ . \tag{5.15}$$

Proof It is observed first that from the definition of the Frobenius norm, $\|A\|_F^2 = \text{trace}(AA)^T$, then the l.h.s. of (5.14) may be expressed as

$$\|A + B\mathbf{w}\mathbf{v}^T\|_F^2 = \sum_{i=1}^{p} \|\mathbf{e}_i^T(A + B\mathbf{w}\mathbf{v}^T)\|_2^2, \tag{5.16}$$

where $\mathbf{e}_i$ is the i^{th} column of the n-dimensional identity matrix. We expand the r.h.s. of (5.16) using the fact that $\|\mathbf{v}\|_2^2 = \mathbf{v}^T\mathbf{v}$. To complete the square in this expansion, the term

$$\sum_{i=1}^{p} \frac{\mathbf{e}_i^T A\mathbf{v}\mathbf{v}^T A^T\mathbf{e}_i}{(\mathbf{v}^T\mathbf{v})} \tag{5.17}$$

is added and subtracted, and this completes the proof.□

It must be noted here that in the above Lemma, the matrices A and B and the vectors $\mathbf{v}$ and $\mathbf{w}^T$ are generic and are not related to any similar matrices mentioned before. Applying Lemma 5.1 to (5.13) gives the equivalent optimisation problem

$$\min_{\boldsymbol{\eta}_n} \hat{J}_1 \equiv \min_{\boldsymbol{\eta}_n}(\mathbf{z}^T\mathbf{z})\|F_p\tilde{\mathbf{z}} + E_p\rho\boldsymbol{\eta}_n\|_2^2, \tag{5.18}$$

where

$$\tilde{\mathbf{z}} = \tfrac{\mathbf{z}}{(\mathbf{z}^T\mathbf{z})} \quad , \quad \alpha_i = \mathbf{e}_i^T F_p(I - \tfrac{\mathbf{z}\mathbf{z}^T}{\mathbf{z}^T\mathbf{z}})F_p^T\mathbf{e}_i, \tag{5.19}$$

since α_i is independent of $\boldsymbol{\eta}_n$ for all i. Here it is still not possible to solve (5.18) easily since $\boldsymbol{\eta}_n$ occurs non-linearly; the step necessary to overcome this is common to this and the next two sections and will thus be explained in the combined minimisation section.

5.4 Eigenvect r c nditi ning

We have shown how to formulate a problem in order to match the desired mode input coupling vectors, but this cannot be the only consideration. To calculate W^T, the inverse of V is needed, so updating a vector in V_2 that makes V ill-conditioned gives rise to inaccurate results. Thus, it is desirable to update the vectors in V_2 while simultaneously controlling the conditioning in the sense that

$$\kappa_F(V) = \|V\|_F \|V^{-1}\|_F \tag{5.20}$$

is reasonably small, where κ denotes the condition number. As shown in Section 2.2.5, this gives an upper bound on the maximum condition number of the closed loop eigenvalues, so also provides a robustness measure. In Section 5.2 we said that the columns V are scaled to unity; this implies

$$\|V\|_F^2 = \sum_{i=1}^{n} \|\mathbf{v}_i\|_2^2 = n, \tag{5.21}$$

so that

$$\kappa_F(V) = n^{\frac{1}{2}} \|V^{-1}\|_F. \tag{5.22}$$

Hence, to reduce the condition number of V, it is sufficient to reduce the Frobenius norm of V^{-1}. Here we need to know how to choose $\boldsymbol{\eta}_n$ to reduce the conditioning of V in addition to matching the mode input coupling vectors. Again we use the structure of V given in (5.5) to obtain the objective function

$$\begin{aligned} J_2 \equiv \|V^{-1}\|_F^2 &= \operatorname{trace}(V^{-1}V^{-T}) \\ &= \sum_{i=1}^{n-1} \mathbf{e}_i^T (R^{-1}R^{-T} + \rho E \boldsymbol{\eta}_n \boldsymbol{\eta}_n^T E^T \rho^T)\mathbf{e}_i + \rho\rho^T \\ &= \sum_{i=1}^{n-1} \|\mathbf{e}_i^T (\rho E \boldsymbol{\eta}_n)\|^2 + \mathbf{e}_i^T R^{-1}R^{-T}\mathbf{e}_i + \rho\rho^T \\ &= \|\rho E \boldsymbol{\eta}_n\|_2^2 + \|\rho S_n \boldsymbol{\eta}_n\|_2^2 + \sum_{i=1}^{n-1} \beta_i, \end{aligned} \tag{5.23}$$

where

$$\begin{cases} \|\rho S_n \boldsymbol{\eta}_n\|_2^2 = \rho\rho^T \\ \\ \beta_i = \mathbf{e}_i^T R^{-1}R^{-T}\mathbf{e}_i, \end{cases} \tag{5.24}$$

since $\|S_n \boldsymbol{\eta}_n\|_2^2 = \|\mathbf{v}_n\|_2^2 = 1$. Our second problem to be solved is

$$\begin{aligned} \min_{\boldsymbol{\eta}_n} \hat{J}_2 &\equiv \min_{\boldsymbol{\eta}_n} \left[\|\rho E \boldsymbol{\eta}_n\|_2^2 + \|\rho S_n \boldsymbol{\eta}_n\|_2^2 \right] \\ &= \min_{\boldsymbol{\eta}_n} \left\| \begin{bmatrix} E \\ S_n \end{bmatrix} \rho \boldsymbol{\eta}_n \right\|_2^2 \end{aligned} \tag{5.25}$$

since β_i is independent of $\boldsymbol{\eta}_n$ for all i

The significance of this error arises when we construct the feedback since the accuracy of the assigned eigenvalues is dependent on this error. This can be seen in Chapter 7.

Recall from Section 4.2.1 that $\|\mathbf{w}_i^T \hat{T}_i\|_2^2$ measures the minimum distance between $\mathbf{w}_i^T$ and a vector in the subspace $\mathcal{T}_i$, where $\hat{T}_i$ is an orthonormal basis for the range space of $\mathcal{T}_i$ (as in (4.18)). To minimise this distance, we put it into a form where we can again use the special structure of V^{-1} in (5.5); observe that the objective function

$$\begin{aligned} J_3 \equiv \sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2 &= \sum_{i=1}^{n} \|\mathbf{e}_i^T V^{-1} \hat{T}_i\|_2^2 \\ &= \sum_{i=1}^{n} \left\| \mathbf{e}_i^T \begin{bmatrix} R^{-1} Q_1^T - \rho E \boldsymbol{\eta}_n \mathbf{q}^T \\ \rho \mathbf{q}^T \end{bmatrix} \hat{T}_i \right\|_2^2 \\ &= \sum_{i=1}^{n-1} \|\mathbf{e}_i^T (R^{-1} Q_1^T - \rho E \boldsymbol{\eta}_n \mathbf{q}^T) \hat{T}_i\|_2^2 + \|\rho \mathbf{q}^T \hat{T}_n\|_2^2 \\ &= \sum_{i=1}^{n-1} \|\mathbf{e}_i^T (H_i - \rho E \boldsymbol{\eta}_n \mathbf{k}_i^T)\|_2^2 + \|\rho \mathbf{k}_n^T\|_2^2, \end{aligned} \tag{5.26}$$

where

$$\begin{cases} \mathbf{k}_i^T = \mathbf{q}^T \hat{T}_i \\ H_i = R^{-1} Q_1^T \hat{T}_i \\ E = R^{-1} Q_1^T S_n. \end{cases} \tag{5.27}$$

To simplify (5.27), we apply Lemma 5.1, except that the term added and subtracted to complete the square is

$$\sum_{i=1}^{n-1} \frac{\mathbf{e}_i^T H_i \mathbf{k}_i \mathbf{k}_i^T H_i^T \mathbf{e}_i}{(\mathbf{k}_i^T \mathbf{k}_i)}. \tag{5.28}$$

Now define $\delta_i = (\mathbf{k}_i^T \mathbf{k}_i)^{\frac{1}{2}}$, to give

$$J_3 = \sum_{i=1}^{n-1} \|\mathbf{e}_i^T (\delta_i^{-1} H_i \mathbf{k}_i - \delta_i \rho E \boldsymbol{\eta}_n)\|^2 + \|\rho \mathbf{k}_n^T\|_2^2 + \sum_{i=1}^{n-1} \boldsymbol{\gamma}_i, \tag{5.29}$$

where

$$\boldsymbol{\gamma}_i = \mathbf{e}_i^T H_i \left(I - \frac{\mathbf{k}_i \mathbf{k}_i^T}{\delta_i^2} \right) H_i^T \mathbf{e}_i. \tag{5.30}$$

To simplify this we make the definitions

$$\begin{cases} \tilde{D} = \text{diag}(\delta_i) \\ g_i = \mathbf{e}_i^T \delta_i^{-1} H_i \mathbf{k}_i \\ \mathbf{g} = [g_1, \ldots, g_{n-1}]^T, \end{cases} \tag{5.31}$$

so that (5.29) becomes

$$J_3 = \|\mathbf{g} - \tilde{D}\rho E \boldsymbol{\eta}_n\|_2^2 + \delta_n^2 \|\rho\|_2^2 + \sum_{i=1}^{n-1} \boldsymbol{\gamma}_i. \tag{5.32}$$

Thus to minimise the error of the left eigenvectors from their correct spaces, we must solve

$$\begin{aligned} \min_{\boldsymbol{\eta}_n} \hat{J}_3 &\equiv \min_{\boldsymbol{\eta}_n} \left[\|(\mathbf{g} - \tilde{D}\rho E \boldsymbol{\eta}_n)\|_2^2 + \delta_n^2 \|\rho S_n \boldsymbol{\eta}_n\|_2^2 \right] \\ &= \min_{\boldsymbol{\eta}_n} \left\| \begin{bmatrix} -\tilde{D}E \\ \delta_n S_n \end{bmatrix} (\rho \boldsymbol{\eta}_n) + \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} \right\|_2^2 , \end{aligned} \tag{5.33}$$

since γ_i is independent of $\boldsymbol{\eta}_n$ for all i. Again the step needed to overcome the non-linearity appearing in $\rho\boldsymbol{\eta}_n$ is described in the next section on the combined minimisation.

5.6 Combined minimisation

We have justified the need for a minimisation algorithm, and in the previous three sections have described three objectives for the routine, each reaching a stage where the objective function to be minimised involves $\rho\boldsymbol{\eta}_n$. We can now combine the three objective functions so that $\boldsymbol{\eta}_n$ is selected to satisfy all three criteria; the new objective function to be minimised is thus

$$\begin{aligned} J_4 &\equiv (\omega_1^2 J_1 + \omega_2^2 J_2 + \omega_3^2 J_3) \\ &= \left[\omega_1^2 \|G_{1d} - [I_p, 0] W^T B\|_F^2 + \omega_2^2 \|V^{-1}\|_F^2 + \omega_3^2 \sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2 \right]. \end{aligned} \tag{5.34}$$

Here the ω_i^2 $(i = 1, 2, 3)$ are weightings to be chosen by the designer according to the design objectives, where

1. ω_1^2 corresponds to input decoupling,

2. ω_2^2 corresponds to the robustness of the problem,

3. ω_3^2 corresponds to eigenvalue accuracy.

5.6.1 Overall objective function

Using the expressions obtained in equations (5.18), (5.25) and (5.33), the objective function in (5.34) can be written as

$$\begin{aligned}\hat{J}_4 \equiv \Bigg[& \omega_1^2(\mathbf{z}^T\mathbf{z})\|F_p\tilde{\mathbf{z}} + E_p(\rho\boldsymbol{\eta}_n)\|_2^2 + \omega_2^2 \left\| \begin{bmatrix} E \\ S_n \end{bmatrix} \rho\boldsymbol{\eta}_n \right\|_2^2 \\ & + \omega_3^2 \left\| \begin{bmatrix} -\tilde{D}E \\ \delta_n S_n \end{bmatrix} (\rho\boldsymbol{\eta}_n) + \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix} \right\|_2^2 \Bigg] .\end{aligned} \tag{5.35}$$

Using the fact that $\|\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{x}$, (5.35) can be re-expressed to give

$$\hat{J}_4 = \left\| \left(\omega_1(\mathbf{z}^T\mathbf{z})^{1} \quad (\right. \right.$$

Householder matrices are symmetric and orthogonal and are important because of their ability to zero specified entries in a matrix or vector. In particular, given any non-zero $\mathbf{x}$

and

$$\mathbf{r} = \begin{pmatrix} \sigma\omega_1(\mathbf{z}^T\mathbf{z})^{\frac{1}{2}}F_p\tilde{\mathbf{z}} \\ \sigma\omega_3\mathbf{g} \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \omega_1(\mathbf{z}^T\mathbf{z})^{\frac{1}{2}}E_p \\ -\omega_3\tilde{D}E \\ \omega_3\delta_n S_n \\ \omega_2 E \\ \omega_2 S_n \end{pmatrix} \mathbf{p}_n. \tag{5.45}$$

This is now (finally) in the form of a standard linear least squares problem and can therefore be solved by a QR (or SVD) method. Thus we can find $\mathbf{y}$ to satisfy the design objectives, but, because we fixed the scaling of $\rho\boldsymbol{\eta}_n$, we need to reconstruct it here. Note that

$$\mathbf{v}_n = S_n\boldsymbol{\eta}_n = \frac{S_n P_H[1,\mathbf{y}]^T}{\sigma\rho} \tag{5.46}$$

from (5.42). Now $\|\mathbf{v}_n\|_2^2 = 1$, which implies

$$\sigma\rho = \|S_n P_H[1,\mathbf{y}]^T\|_2, \tag{5.47}$$

and hence

$$\mathbf{v}_n = \frac{S_n P_H[1,\mathbf{y}]^T}{\|S_n P_H[1,\mathbf{y}]^T\|_2}. \tag{5.48}$$

5.7 Alg rithm

This algorithm assumes that partial eigenstructure assignment by output feedback has been performed as in Section 4.3; hence we have the closed loop eigenstructure (V,Λ) that contain the desired $V_1 \in \mathbb{C}^{n\times p}$, $\Lambda_1 \in \mathbb{C}^{p\times p}$. This is used as the starting point for the algorithm.

STEP 1

1. normalise columns of V so that $\|\mathbf{v}_i\|_2^2 = 1$ $(i = 1,\ldots,n)$
2. re-order V and Λ so that $V = [V_1, V_2]$, $\Lambda = \mathrm{diag}[\Lambda_1, \Lambda_2]$
3. if required, choose a new set $\tilde{\Lambda}_2$ and calculate (and store) all right null spaces S_i for $\tilde{\Lambda}_2$ using the QR decomposition,

$$[U_1^T(A - \lambda_i I)]^T = [\hat{S}_i, S_i] \begin{bmatrix} R_{Ri} \\ 0 \end{bmatrix} \tag{5.49}$$

for all $\lambda_i \in \mathrm{diag}\tilde{\Lambda}_2$

5. calculate necessary left range space components

for $i = 1 : n$

select $\hat{T}_i$ from left range space store

calculate

$\mathbf{k}_i^T = \mathbf{q}^T \hat{T}_i$

$H_i = R^{-1} Q_1^T \hat{T}_i$

$\delta_i = (\mathbf{k}_i^T \mathbf{k}_i)^{\frac{1}{2}}$

$g_i = \mathbf{e}_i^T \delta$

and eigenvalue accuracy, and have included these objectives in the problem. The result is a multi-criteria objective function with weightings to be chosen by the designer to obtain the design requirements. We have outlined the algorithm in a more usable form in Section 5.7, with a note on how to treat complex eigenvalues. Before presenting a section on how further to scale the first p vectors, we give a table of the dimensions of the elements of the algorithm for completeness, as well as some illustrative examples.

5.7.3 Component dimensions

component	number of rows	number of columns	real/complex
V_1	n	p	C
V_2	n	$n-p$	C
Λ_1	p	p	C
Λ_2	$n-p$	$n-p$	C
S_n	n	m	C
$\hat{T}_i$	n	$n-p$	C
$V\_$	n	$n-1$	C
Q_1	n	$n-1$	C
$\mathbf{q}$	n	1	C
R	$n-1$	$n-1$	C
$\mathbf{z}^T$	1	m	C
D_p	p	m	C
E_p	p	m	C
F_p	p	m	C
$\tilde{\mathbf{z}}$	m	1	C
$\mathbf{k}_i^T$	1	$n-p$	C
H_i	$n-1$	$n-p$	C
δ_i	1	1	C
g_i	1	1	C
$\mathbf{g}$	$n-1$	1	C
$\tilde{D}$	$n-1$	$n-1$	C
σ	1	1	R
P_H	m	m	C
$\mathbf{p}_n$	m	1	C
$\hat{P}$	m	$m-1$	C
M	$4n+p-2$	$m-1$	C
$\mathbf{r}$	$4n+p-2$	1	C
$\mathbf{y}$	$m-1$	1	C

5.8 Example

We next present an example to demonstrate the minimisation algorithm. The system used here has 10 states, 3 inputs and 4 outputs. The system matrices are given in Example 2 of Chapter 8 (Section 8.4). The desired eigenvalues and corresponding right eigenvectors have been achieved by partial eigenstructure assignment, but again the left eigenvectors have a residual error. This error is

$$\|G_{1d} - G_{1a}\|_F^2 = 3.7026 \times 10^3. \tag{5.59}$$

The results of the algorithm run with different values for the weightings, ω_i^2 $(i = 1, \ldots, 3)$ follow. The results are given in tabular form, where the objective function denotes the quantity

$$\omega_1^2 \ \|G_{1d} - G_{1a}\|_F^2 + \omega_2^2 \ \|V^{-1}\|_F^2 + \omega_3^2 \sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2$$

routine, and hence we are trying to see how close we can make the matching.

reduced, as has the left eigenspace error. Also, the algorithm has converged in 30 sweeps.

Parameters: $(\omega_1^2, \omega_2^2, \omega_3^2) = (100, 1,$

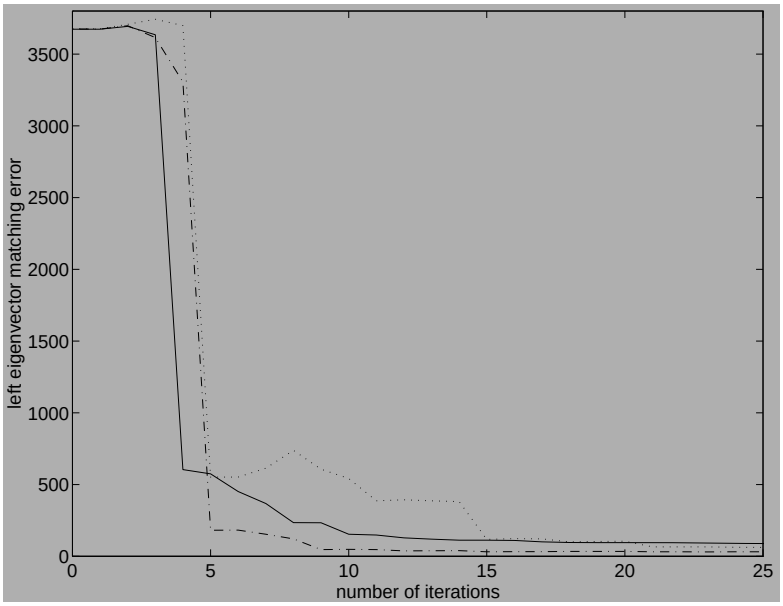


Figure 5.1: Convergence histories

Indeed, we should really show the convergence histories of the overall objective

not important. Thus we may impose any scaling on the columns of V_1 to reduce some error. Here we use it to reduce the value of the objective function in (5.64), and hence we would like to see the effect of this scaling on the left eigenvectors. Consider scaling W_1^T by a matrix D_1, while leaving W_2^T unchanged,

$$\begin{pmatrix} D_1 & 0 \\ 0 & I_{n-p} \end{pmatrix} \begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix} = \begin{pmatrix} D_1 W_1^T \\ W_2^T \end{pmatrix}, \tag{5.65}$$

giving

$$\begin{aligned} [V_1, V_2]\text{new} &= (W^T)^{-1} \begin{pmatrix} D_1 & 0 \\ 0 & I_{n-p} \end{pmatrix}^{-1} \\ &= [V_1, V_2] \begin{pmatrix} D_1^{-1} & 0 \\ 0 & I_{n-p} \end{pmatrix} \\ &= [V_1 D_1^{-1}, V_2]. \end{aligned} \tag{5.66}$$

The aim is to find D_1 to solve (5.64). The idea is that we use the previous routine to update $\mathbf{v}_n$, then rescale V_1 by finding D_1 as the solution to

$$\min_{D_1} \tilde{J}_1 \equiv \min_{D_1} \left\| G_{1d} - [I_p, 0] \begin{bmatrix} D_1 & 0 \\ 0 & I_{n-p} \end{bmatrix} V^{-1} B \right\|_F^2 = \min_{D_1} \|G_{1d} - D_1 W_1^T B\|_F^2. \tag{5.67}$$

If we denote the rows of G_{1d} and $W_1^T B$ by $\mathbf{g}_i^H$ and $\tilde{\mathbf{w}}_i^H$ respectively ($i = 1, \ldots, p$), then

$$\tilde{J}_1 = \sum_{i=1}^{p} \|\mathbf{e}_i^T (G_{1d} - D_1 W_1^T B)\|_2^2 = \sum_{i=1}^{p} (\mathbf{g}_i^H - d_i \tilde{\mathbf{w}}_i^H)(\mathbf{g}_i - d_i \tilde{\mathbf{w}}_i). \tag{5.68}$$

It is evident that each term is exclusive in that the scaling factor d_i affects only the i^{th} row of W_1^T

implying that

$$d_i = \frac{\mathbf{g}_i^H \tilde{\mathbf{w}}_i + \tilde{\mathbf{w}}_i^H \mathbf{g}_i}{2\tilde{\mathbf{w}}_i^H \tilde{\mathbf{w}}_i}. \tag{5.71}$$

The problem here is that when calculating the D_1, we are not taking into account the affect on the left eigenspace error. Hence we may actually increase the the objective function being minimised by updating the V_2 set.

It will also affect the condition number of $\tilde{V}$ but is not a cause for concern since it will just mean that we will increase the value of the upper bound, $\kappa_F(V)$, for the sensitivity of the individual eigenvalues, it will not affect the conditioning of the eigenvalues themselves. So it is not guaranteed to converge, but intuitively this scaling on V_1 should not have too bad an effect on the results. In summary we have a two stage minimisation process

1. update the last column of V_2 to satisfy a minimisation problem
2. scale V_1 to further minimise some criteria.

We apply this theory to some of the examples from Section 5.8 to show the expected improvement. The results given have values for the conditioning of the vectors and the left eigenspace error calculated before the scaling is implemented at each iteration. The left vector matching error is calculated with the scaling implemented since this is what the scaling reduces. The objective function is still valid because at each iteration the scaling of the full set of right vectors is fixed.

5.9.1 Optimal scaling examples

We use the same example as in Section 5.8, but this time we shall apply the optimal scaling onto V_1 as described previously.

c

Proof

Necessity : If we consider one row of $G_{1d} - \hat{G}_{1a}$ then

$$\begin{aligned}(G_{1d} - \hat{G}_{1a})_i &= [0 \ldots 1 \ldots 0] - \left[\tfrac{g_{i1}}{g_{ij}} \ldots 1 \ldots \tfrac{g_{im}}{g_{ij}}\right] \\ &= \left[-\tfrac{g_{i1}}{g_{ij}} \ldots 0 \ldots - \tfrac{g_{im}}{g_{ij}}\right] .\end{aligned} \tag{5.74}$$

Then

$$\|(G_{1d} - \hat{G}_{1a})_i\|_2^2 = \left[\left(\frac{g_{i1}}{g_{ij}}\right)^2 + \ldots + \left(\frac{g_{im}}{g_{ij}}\right)^2\right], \tag{5.75}$$

where there are $m-1$ terms in the sum. There is one of these sums for each row, and we require $(\frac{g_{i1}}{g_{ij}}) < \tau \Rightarrow (\frac{g_{i1}}{g_{ij}})^2 < \tau^2$, giving

$$\|G_{1d} - \hat{G}_{1a}\|_F^2 \leq \tau^2 p(m-1), \tag{5.76}$$

a necessary condition for the desired level of input decoupling to be achieved.

Sufficency :

$$\|G_{1d} - \hat{G}_{1a}\|_F^2 < \tau^2 \Rightarrow \max \left|\frac{g_{i1}}{g_{ij}}\right| \quad \begin{matrix} 6 \\ 6 \\ \max \end{matrix}$$

5.10.1 Summary of results

The previous examples show that the minimisation algorithm can be used to reduce the objective function that represents set criteria, but that quite a lot of computation is needed. As with most multi-criteria optimisation routines, it is not clear how best to choose the individual weightings, although the examples do give some indications.

5.11 Alternative starting p int

In the preceding theory, it has been assumed that partial eigenstructure assignment has been performed, this being used as the starting point for the minimisation algorithm. As an alternative, we may remove the need for first using partial eigenstructure assignment as h to

iterates through the unassigned right vectors until some minimisation criteria is satisfied. In the developmen

objective function much quicker, but will give rise to problems when trying to reconstruct the feedback, as will be explained in the next chapter.

6.1 Eigenvector partitioning

We assume that we have a set of n linearly independent vectors, V, which are column normalised so that $\|\mathbf{v}_i\|_2^2 = 1$ $(i = 1, \ldots, n)$. We partition V in the form

$$V = [V_1, V_2] \quad , \quad W^T = \begin{bmatrix} W_1^T \\ W_2^T \end{bmatrix} \tag{6.2}$$

where $V_1 = [\mathbf{v}_1, \ldots, \mathbf{v}_p]$, $W_1^T = [\mathbf{w}_1, \ldots, \mathbf{w}_p]^T$. The method is almost exactly the same as for the restricted minimisation in Chapter 5; we again partition V in the form

$$V = \left[[Q_1, \mathbf{q}] \begin{bmatrix} R \\ \mathbf{0}^T \end{bmatrix}, \mathbf{v}_n \right], \tag{6.3}$$

so that the inverse of V is given by

$$V^{-1} = \begin{bmatrix} R^{-1} & -\rho R^{-1} Q_1^T \mathbf{v}_n \\ \mathbf{0}^T & \rho \end{bmatrix} \begin{bmatrix} Q_1^T \\ \mathbf{q}^T \end{bmatrix}, \tag{6.4}$$

where $\rho = \frac{1}{\mathbf{q}^T \mathbf{v}_n}$ is a scalar. We shall now define the two criteria (as opposed to three in Chapter 5) that we want to satisfy when performing the minimisation algorithm.

6.2 Left eigenvector matching

Again the primary aim of the minimisation algorithm is to reduce the level of coupling inherent in the mode input coupling vectors. Thus, we aim to solve the problem

$$\begin{cases} \min J_1 \equiv \min \|G_{1d} - G_{1a}\|_F^2 = \min_V \|G_{1d} - [I_p, 0] V^{-1} B\|_F^2 \\ \\ \text{subject to } \|\mathbf{v}_i\|_2^2 = 1 \ (i = 1, \ldots, n) \end{cases} . \tag{6.5}$$

As in Chapter 5, we leave out the 'subject to' constraint for brevity in the following theory, although it is important to remember that it still applies. At this point

in the algorithm in Chapter 5, $\mathbf{v}_n$ was expressed as

$$\mathbf{v}_n = S_n \boldsymbol{\eta}_n, \tag{6.6}$$

where finding $\boldsymbol{\eta}_n$ corresponds to restricting $\mathbf{v}_n$ to be in the subspace corresponding to the desired eigenvalue, λ_n. Here we leave it as $\mathbf{v}_n$, so that we choose $\mathbf{v}_n \in \mathbb{C}^n$ to minimise some criteria.

Following the theory from Section 5.3, we substitute for V^{-1} from (6.4) to reduce (6.5) to

$$\min_{\mathbf{v}_n} J_1 \quad = \min_{\mathbf{v}_n} \left\| G_{1d} - [I_p, 0] \begin{bmatrix} D - \rho E \mathbf{v}_n \mathbf{z}^T \\ \rho \mathbf{z}^T \end{bmatrix} \right\|_F^2, \tag{6.7}$$

where

$$\begin{cases} \mathbf{z}^T = \mathbf{q}^T B \\ D = R^{-1} Q_1^T B \\ E = R^{-1} Q_1^T. \end{cases} \tag{6.8}$$

Note that $D \in \mathbb{C}^{n-1 \times m}$, and $[I_p, 0]$ only multiplies the first p rows of $D - \rho E \mathbf{v}_n \mathbf{z}^T$; therefore if $(n-1) \geq p$, the last row, $\rho \mathbf{z}^T$, of $V^{-1}B$ makes no contribution to the minimisation. Then (6.7) is equivalent to

$$\min_{\mathbf{v}_n} \| F_p + \rho E_p \mathbf{v}_n \mathbf{z}^T \|_F^2, \tag{6.9}$$

where $F_p = G_{1d} - D_p$, and the subscript p denotes that the subscripted matrix has been pre-multiplied by $[I_p, 0]$. We apply Lemma 5.1 to (6.9) to give

$$\min_{\mathbf{v}_n} \hat{J}_1 \equiv \min_{\mathbf{v}_n} (\mathbf{z}^T \mathbf{z}) \| F_p \tilde{\mathbf{z}} + E_p \rho \mathbf{v}_n \|_2^2. \tag{6.10}$$

We note here that if we were only aiming to minimise (6.10), then $\mathbf{z}^T \mathbf{z}$ could be removed as a constant factor multiplying the objective function at each step. If we include other criteria, then it must remain, either explicitly, or incorporated into a weighting factor on $\hat{J}_1$. AsD6dTj6TffiTD6.Tj6ffithen remosatisheffiTD0remo1Tj-deffibjectivz

that its condition number is kept low. The columns of V are normalised to one which implies that

$$\kappa_F(V) = n^{\frac{1}{2}} \|V^{-1}\|_F, \tag{6.11}$$

so that we restrict the condition number by keeping a bound on the Frobenius norm of the inverse of V. Using the structure of V given in (6.4) results in the objective function

$$J_2 \equiv \|V^{-1}\|_F^2$$

two criteria to be satisfied; our objective function to be minimised is

$$
\begin{aligned}
J_3 &\equiv (\omega_1^2 J_1 + \omega_1^2 J_2) \\
&= \Big[\ \omega_1^2 \ \| G_{1d} - [I_p, 0] W^T B \|_F^2 + \omega_{22}^2
\end{aligned}
$$

where

$$M = \begin{pmatrix} \omega_1(\mathbf{z}^T\mathbf{z})^{\frac{1}{2}}E_p \\ \omega_2 E \\ \omega_2 \end{pmatrix} \hat{P}, \tag{6.22}$$

and

$$\mathbf{r} = \begin{pmatrix} \sigma\omega_1(\mathbf{z}^T\mathbf{z})^{\frac{1}{2}}F_p\tilde{\mathbf{z}} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \omega_1(\mathbf{z}^T\mathbf{z})^{\frac{1}{2}}E_p \\ \omega_2 E \\ \omega_2 I_n \end{pmatrix} \mathbf{p}_n. \tag{6.23}$$

We have4pTj6TffiTD6zTj6ffiTc6ehf6ffiffiTD6zTthffiTD6zolaT6ffiffiTD6ffiTc6vTj6ffiffiTD6ej6Tpffi

3. calculate

$$\begin{aligned} \mathbf{z}^T &= \mathbf{q}^T B \\ D_p &= [I_p, 0] R^{-1} Q_1^T B \\ E_p &= [I_p, 0] R^{-1} Q_1^T \\ F_p &= G_{1d} - D_p \\ \tilde{\mathbf{z}} &= \frac{\mathbf{z}}{(\mathbf{z}^T \mathbf{z})} \end{aligned} \tag{6.26}$$

4. calculate Householder transformation as in (6.18) using the QR decomposition,

$$\begin{aligned} \mathbf{q} &= Q \begin{pmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = Q r_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ \sigma &= |r_1| \\ P_H &= \frac{r_1 Q}{\sigma} \end{aligned} \tag{6.27}$$

5. partition $P_H = [\mathbf{p_n}, \hat{P}]$

6. form M, $\mathbf{r}$ as in (6.22) and (6.23) respectively, and solve for $\mathbf{y}$ the following

$$\min_{\mathbf{y}} \|M\mathbf{y} + \mathbf{r}\|_2^2 \tag{6.28}$$

STEP 3

1. rescale $\mathbf{v}_n$ to give updated $\tilde{\mathbf{v}}_n$

$$\tilde{\mathbf{v}}_n = \frac{P_H [1, \mathbf{y}]^T}{\|P_H [1, \mathbf{y}]^T\|_2} \tag{6.29}$$

2. re-order V_2 to give

$$\tilde{V}_2 = [\tilde{\mathbf{v}}_n, \mathbf{v}_{p+1}, \ldots, \mathbf{v}_{n-1}]$$

6.5.1 Component dimensions

For completeness, a table of the dimensions of the components of the algorithm is given.

component	number of rows	number of columns	real/complex
V_1	n	p	C
V_2	n	$n-p$	C
$V_{\_}$	n	$n-1$	C
Q_1	n	$n-1$	C
$\mathbf{q}$	n	1	C
R	$n-1$	$n-1$	C
$\mathbf{z}^T$	1	m	C
D_p	p	m	C
E_p	p	n	C
F_p	p	m	C
$\tilde{\mathbf{z}}$	m	1	C
σ	1	1	R
P_H	n	n	C
$\mathbf{p}_n$	n	1	C
$\hat{P}$	n	$n-1$	C
M	$2n+p-1$	$n-1$	C
$\mathbf{r}$	$2n+p-1$	1	C
$\mathbf{y}$	$n-1$	1	C

6.5.2 Notes and summary

We have presented an algorithm that aims to minimise an objective function, allowing the minimisation vectors to be chosen freely from the complex plane. This method follows closely that in Chapter 5, but differs in that it does not have the left eigenspace error criterion.

It should be noted that we can also implement the optimal scaling on V_1 as in Section 5.9 or the alternative scaling as in Section 5.10, but the theory is exactly the same since it does not rely on null spaces corresponding to any eigenvalues.

The algorithm is complete except in the fact that we have not specified how to obtain the initial set of vectors, V, or how to reconstruct the feedback matrix. The former is covered in the next section, the latter in the next chapter.

6.6 Selecti n f initial vect r set f r alg rithm

We have explained that running the minimisation algorithm by letting the vectors be chosen freely should decrease the value of the objective function at a faster rate than for the restricted case. We have also given the theory for this in the preceding sections, but started with the assumption that we have an initial set of n linearly independent vectors, V. There are three ways of finding this initial set:

1. perform partial eigenstructure assignment as in Section 4.3, using the closed loop eigenvectors to run the minimisation,

2. find V_1 by the projection method corresponding to the desired eigenvalues, Λ_1; take some initial set V_2,

3. perform an unrestricted projection to obtain V_1; take some initial set V_2.

These methods are described in the following sections. After each section an exleeectiog6ffiffiffiffiTffiTD6Aftersecsetilluedtheftheory he ectioT66ei earouectinstrsectinstec

However, in general, the closed loop eigenstructure may have (will probably have) complex modes. If at least one pair of complex conjugate eigenvectors appears in V_1 (which will remain unaffected by the algorithm), then $V = [V_1, V_2]$ will be complex throughout. This will make each updated vector of new V_2 complex. Since each vector is updated individually, the result will be

$$\left.\begin{array}{l} V_1 \text{ self-conjugate} \\ V_2 \text{ not self-conjugate} \end{array}\right\} \Rightarrow \; V = [V_1, V_2] \text{ not self-conjugate.}$$

Following the minimisation algorithm, the feedback needs to be constructed in some way, as is detailed in Chapter 7. Whatever method is used, constructing a feedback from a set of vectors that is not self-conjugate results in a complex feedback matrix. We must therefore transform our vectors into a real set so that the minimisation algorithm generates real solution vectors.

Transforming complex vectors to real vectors

We assume, without loss of generality, that there is exactly one complex conjugate pair of eigenvalues within the closed loop set. If the eigenvalues are permutated so that this pair appears first, then the eigen-decomposition of the closed loop system is

$$V_{\text{cc}} = [\mathbf{v}_1^{\text{re}} + i\mathbf{v}_1^{\text{im}}, \mathbf{v}_1^{\text{re}} - i\mathbf{v}_1^{\text{im}}, \mathbf{v}_3, \ldots, \mathbf{v}_n]$$

$$\Lambda_{\text{cc}} = \begin{bmatrix} \lambda_1^{\text{re}} + i\lambda_1^{\text{im}} & & & & & \\ & \lambda_1^{\text{re}} - i\lambda_1^{\text{im}} & & & & \\ & & \lambda_3 & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & \lambda_n \end{bmatrix}, \tag{6.33}$$

where the 'cc' subscript denotes that the decomposition is in its complex form, and all off-diagonal elements are zero. The self-conjugate set of eigenvectors can be transformed into a real set by post-multiplying it by the transformation matrix

$$P = \left[\begin{array}{cc|c} \frac{1}{2} & -\frac{1}{2}i & \\ \frac{1}{2} & \frac{1}{2}i & 0 \\ \hline & 0 & I \end{array}\right]. \tag{6.34}$$

The real eigenvector set is thus

$$V_{\rm re} = [\mathbf{v}_1^{\rm re}, \mathbf{v}_1^{\rm im}, \mathbf{v}_3, \ldots, \mathbf{v}_n], \tag{6.35}$$

and the corresponding real representation of the eigenvalue matrix is

$$\Lambda_{\rm re} = \begin{bmatrix} \lambda_1^{\rm re} & \lambda_1^{\rm im} & & & \\ -\lambda_1^{\rm im} & \lambda_1^{\rm re} & & & \\ & & \lambda_3 & & \\ & & & \ddots & \\ & & & & \lambda_n \end{bmatrix}, \tag{6.36}$$

The minimisation routine is run with various values for the parameters. The results are given in tabular form, where the objective function denotes the quantity

$$\omega_1^2 \ \|G_{1d} - G_{1a}\|_F^2 + \omega_2^2 \ \|V^{-1}\|_F^2. \tag{6.41}$$

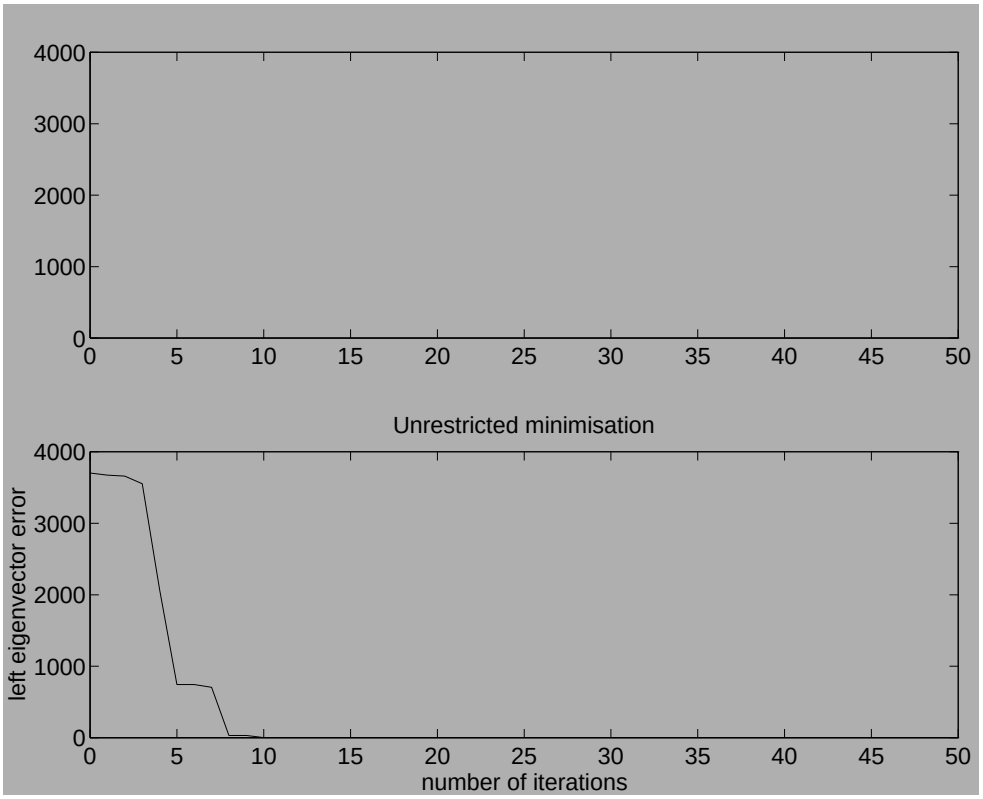
In each table, the value of the Frobenius norm condition number of V is given. It should be remembered that this is not the actual quantity being reduced. From Section 6.3

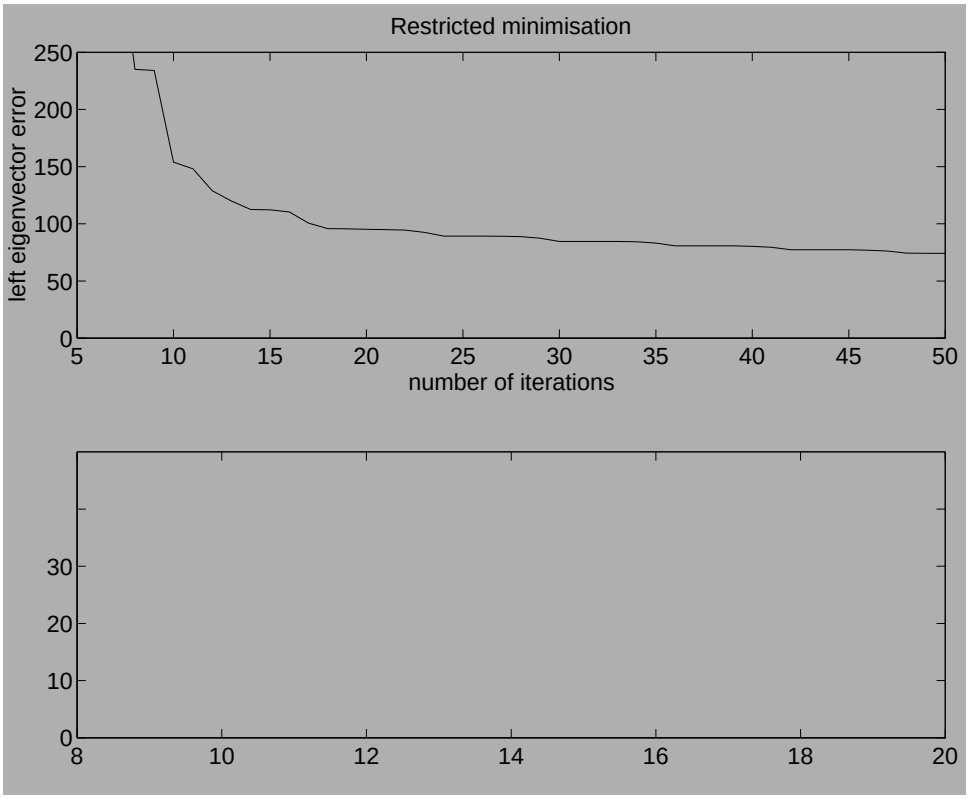
$$\kappa_F(V) = n^{\frac{1}{2}} \|V^{-1}\|_F \tag{6.42}$$

so that $\|V^{-1}\|_F^2$ is actually being reduced in order to reduce $\kappa_F(V)$. Also in the table, the number of sweeps is given. For this example, $n = 10$, $p = 4$, and $n - p = 6$, and hence one sweep is equivalent to updating each of the six vectors in V_2. One iteration is defined as updating one v

onvergence comparison

We now show graphs of the reduction of the error in the left vector matching for both the restricted and the unrestricted minimisation algorithms. It should be noted that the graphs show the reduction in the error against the number of iterations, not the number of sweeps.




Restricted minimisation
left eigenvector error
number of iterations
250
200
150
100
50
0
5
10
15
20
25
30
35
40
45
50
30
20
10
0
8
10
12
14
16
18
20

weighting on both criteria, to reduce both simultaneously.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$1.1181e+04$	$3.7107e+03$	$1.3133e+03$
1	$1.1335e+02$	$4.6963e+01$	$1.2407e+02$
5	$6.6694e+01$	$2.2698e+01$	$1.0100e+02$
10	$2.9888e+01$	$3.2990e+00$	$7.8520e+01$
15	$2.7022e+01$	$3.9518e+00$	$7.3140e+01$
25	$2.6155e+01$	$3.8630e+00$	$7.1896e+01$
50	$2.4752e+01$	$3.7463e+00$	$6.9791e+01$

We observe that the error in both the left vector matching and the conditioning of the vectors is reduced, but that the matching error is considerably higher than the virtual zero value attained before.

Parameters: $(\omega_1^2, \omega_2^2) = (100, 1)$

The previous example illustrates the trade-off between the two criteria considered. However, the left vector matching is our main consideration and we just want to ensure that the conditioning does not become very large. To this end, we place a higher relative weighting on the left vector matching.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$3.7854e+05$	$3.7107e+03$	$1.3133e+03$
1	$8.3518e+03$	$7.9453e+01$	$3.0702e+02$
5	$4.4868e+03$	$4.2055e+01$	$2.5538e+02$
10	$2.5182e+03$	$2.2732e+01$	$2.3836e+02$
25	$1.0294e+03$	$8.5914e+00$	$1.9871e+02$
70	$1.1034e+02$	$3.6497e-01$	$1.3085e+02$
80	$7.2064e+01$	$6.2381e-02$	$1.2355e+02$
100	$6.1702e+01$	$9.8189e-03$	$1.1866e+02$

As expected, the left matching is reduced considerably in comparison to the previous example, and the conditioning has increased slightly. Note, however, that a lot of computation is required to reduce the left vector matching from $O(10^0)$ to $O(10^{-3})$.

6.6.3 Projection method

The second method for constructing an initial vector set for the minimisation algorithm is just to perform the first part of the partial eigenstructure assignment algorithm, namely the projection method to obtain V_1. This is performed as follows

- specify design requirements, $\Lambda_1 \in \mathbb{C}^{p\times p}$, G_{0d},
- perform projection as in Section 4.3.7 to obtain the best set of achievable vectors, G_{0a}; extract V_1 from this,
- specify an initial set $V_2 \in \mathbb{R}^{n\times n-p}$ and form $V_{\mathrm{cc}} = [V_1, V_2]$,
- transform V_{cc} into its real form ; use this for the minimisation algorithm.

One question that arises from this is how to choose some initial, real set, V_2, such that

$$\mathrm{rank}[V_1, V_2] = n. \tag{6.44}$$

Since the minimisation is unrestricted, V_2 can be any set of real vectors that we decide to choose, but must not be such that the full vector set is badly conditioned. There is no obvious solution; here we choose to make use of the identity matrix. We let the first $n-p$ rows (and all columns) of V_2 be the $(n-p)$-dimensional identity matrix, denoted I_{n-p}. Thus,

$$V_2 = \begin{bmatrix} I_{n-p} \\ X_1 \end{bmatrix}, \tag{6.45}$$

where $X_1 \in \mathbb{R}^{p\times n-p}$ is the remaining section of V_2 to be filled. If $p \geq n-p$, let the first $n-p$ rows of X_1 be I_{n-p} so that

$$V_2 = \begin{bmatrix} V \\ \lambda I \end{bmatrix}$$

I_{n-p}. Hence, we are just repeating the $(n-p)$-dimensional identity matrix until V_2 is full. For example if $n = 5$, $p = 2$, then

$$V_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} I_3 \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{first } p \text{ rows of } I_3 \end{matrix} . \tag{6.47}$$

Note that this does not guarantee that $V = [V_1, V_2]$ will not have a high condition number, or even that V is of full rank; but we will be very unlucky if generating V_2 in this way causes V to be rank deficient.

Another method, although more expensive than the one just described, is to use the QR decomposition of V_1,

$$V_1 = [Q_{V_1}, Q_{V_2}] \begin{bmatrix} R_V \\ 0 \end{bmatrix} . \tag{6.48}$$

From this

$$Q_{V_2}^T V_1 = 0, \tag{6.49}$$

so that Q_{V_2} is an orthonormal basis for the complement of V_1. Thus, if we let

$$V_2 = Q_{V_2}, \tag{6.50}$$

then the columns in V_2 are linearly independent to the columns in V_1, and hence

$$\text{rank } [V_1, V_2] = n. \tag{6.51}$$

Computationally, it does not seem worth performing the extra work involved in the partial eigenstructure assignment algorithm. The unassigned eigenstructure, (V_2, Λ_2), may have complex conjugate modes that would hav

6.6.4 Example 2

Here, using the specified eigenvalues, the projection method is used to obtain the mode input coupling vectors; from these, the V_1 set is obtained. The V_2 set is obtained by cycling the identity matrix, as just described. The residual error in G_1 is

$$\|G_{1d} - G_{1a}\|_F^2 = 1.1757 \times 10^4. \tag{6.52}$$

As a comparison, we shall show the results for both methods of constructing a set, V_2, such that the full set of vectors is of full rank. The first table for each set of parameters is for when V_2 is calculated by repeating the identity matrix; the second table is when V_2 is calculated from the QR decomposition of V_1.

Parameters: $(\omega_1^2, \omega_2^2) = (1, 0)$

Again, we start off by considering the problem of just matching the left vectors.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$1.1757e+04$	$1.1757e+04$	$7.5912e+01$
1	$4.0990e+00$	$4.0990e+00$	$2.0319e+02$
2	$4.3528e-01$	$4.3528e-01$	$4.0576e+02$
3	$1.6485e-04$	$1.6485e-04$	$4.5249e+02$
4	$1.0992e-08$	$1.0992e-08$	$4.5653e+02$
5	$9.0931e-13$	$9.0931e-13$	$4.5251e+02$
10	$1.1968e-19$	$1.1968e-19$	$4.5690e+02$

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$9.1991e+03$	$9.1991e+03$	$3.7908e+01$
1	$9.1889e-01$	$9.1889e-01$	$9.3664e+03$
2	$6.8553e-03$	$6.8553e-03$	$8.4535e+03$
3	$2.0335e-04$	$2.0335e-04$	$8.5646e+03$
4	$3.0354e-06$	$3.0354e-06$	$8.5814e+03$
5	$1.1229e-07$	$1.1229e-07$	$8.5787e+03$
10	$4.7397e-16$	$4.7397e-16$	$8.5784e+03$

As in Example 1 in Section 6.6.2, the left vector matching is reduced to zero (to machine accuracy) in both tables. It is interesting to note that initially wA s A s t n

Parameters: $(\omega_1^2, \omega_2^2) = (100, 1)$

We impose a relative high weighting on the left vector matching as this is our primary concern.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$1.1757e+06$	$1.1757e+04$	$7.5912e+01$
1	$3.1115e+04$	$2.8288e+02$	$8.0956e+02$
2	$2.5771e+04$	$2.2903e+02$	$8.1548e+02$
4	$2.1609e+04$	$1.8722e+02$	$8.1814e+02$
6	$8.9644e+03$	$7.7609e+01$	$5.2826e+02$
12	$2.0802e+03$	$5.5313e+00$	$5.9507e+02$
20	$1.6900e+03$	$1.1582e+00$	$6.0418e+02$
25	$1.6179e+03$	$9.6541e-01$	$5.9394e+02$
50	$1.3773e+03$	$8.6059e-01$	$5.4718e+02$

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$9.1991e+05$	$9.1991e+03$	$3.7908e+01$
1	$7.8075e+04$	$3.3943e+02$	$3.1990e+03$
2	$4.4394e+04$	$3.1008e+02$	$1.7618e+03$
6	$2.4772e+04$	$1.8959e+02$	$1.1609e+03$
12	$6.8079e+03$	$5.4435e+01$	$5.6245e+02$
20	$5.5607e+03$	$4.5020e+01$	$4.9546e+02$
25	$5.1612e+03$	$4.1603e+01$	$4.8176e+02$
50	$3.4571e+03$	$2.6353e+01$	$4.3653e+02$
120	$7.0397e+02$	$3.8295e-01$	$3.9288e+02$
150	$6.3267e+02$	$2.3023e-01$	$3.7599e+02$

These results are unsatisfactory in comparison to the weighting $(\omega_1^2, \omega_2^2) = (1, 0)$, where the matching tends to zero and the conditioning is less than here.

Note also the extra work taken by the second method for the construction of V_2, namely the QR decomposition. However, in general, both methods for the construction of V_2 are about the same. The main differences are that the first method is much cheaper computationally, whereas the second method guarantees

that the vectors are of full rank. On these points, it is better to use the first method, repeating the identity matrix, because it is unlikely that this will give a rank deficient set of vectors.

is overdetermined. Finally, from (6.63), we form

$$V_1 = [\mathbf{v}_{a1}, \ldots, \mathbf{v}_{ap}], \tag{6.64}$$

and V_2 is chosen as in Section 6.6.3 so as to ensure $V = [V_1, V_2]$ is of full rank. The minimisation algorithm is then run using this matrix V.

One other problem is that in G_{0d}, two columns may represent a complex conjugate pair of eigenvectors. The constrained projection method in Section 4.3.5 forces the unspecified elements in these two vectors to be complex conjugate; then G_{0d} is self-conjugate. However, here we specify no eigenvalues, thus the unconstrained projection generates a rank deficient G_{0d}. To overcome this, we re-express G_{0d} in its real representation, so that for G_{0d} given in (6.60)

$$G_{0d} = \begin{bmatrix} x & x & 0 & 0 \\ 0 & 0 & x & x \\ 1 & x & 0 & 0 \\ 0 & 0 & 1 & x \end{bmatrix}, \tag{6.65}$$

so that again we can generate a full rank set of initial vectors.

6.6.6 Example 3

Here, the specified G_{0d} is re-expressed in its real form. G_{0a} is calculated using

Parameters: $(\omega_1^2, \omega_2^2) = (1, 0)$

Weight just the left vector matching.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$4.3455e+05$	$4.3455e+05$	$3.0182e+02$
1	$7.8023e+01$	$7.8023e+01$	$1.0362e+02$

Parameters: $(\omega_1^2, \omega_2^2) = (1, 1)$

Weight both criteria equally.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$4.3456e+05$	$4.3455e+05$	$6.2802e+01$
1	$9.4406e+00$	$1.4182e+00$	$5.8240e+01$
2	$8.9394e+00$	$8.6664e-01$	$5.8423e+01$
5	$8.5341e+00$	$4.2617e-01$	$5.8550e+01$
10	$8.4711e+00$	$3.7166e-01$	$5.8520e+01$
100	$8.4248e+00$	$3.3936e-01$	$5.8469e+01$

Again we have reduced both criteria, most of the work is done in the first two sweeps. A lot of computation is required to reduce the matching by a further order.

Parameters: $(\omega_1^2, \omega_2^2) = (100, 1)$

Choose to impose a relative high weighting on the matching.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$4.3455e+07$	$4.3455e+05$	$6.2802e+01$
1	$4.1667e+02$	$4.0464e+00$	$7.1322e+01$
2	$2.7074e+01$	$1.5105e-01$	$7.1138e+01$

error is reduced virtually to zero, in comparison to $O(10^1)$ for the restricted case. However, care needs to be tak

time. The results are given in the table below.

starting point	parameters ω_1^2	parameters ω_2^2	stopping criterion	flops Toolbox	flops ours	cpu time (secs.) Toolbox	cpu time (secs.) ours
partial	1	0	$J < 10^{-6}$	$O(10^7)$	$O(10^5)$	15.07	5.5
eigenstructure	1	1	$J < 25$	$O(10^8)$	$O(10^6)$	190.96	50.05
assignment	100	1	$J < 62$	$O(10^8)$	$O(10^7)$	190.59	109.02
projection for	1	0	$J < 10^{-6}$	$O(10^7)$	$O(10^5)$	14.27	4.54
V_1, QR	1	1	$J < 40$	$O(10^8)$	$O(10^6)$	97.7	41.71
for V_2	100	1	$J < 3000$	$O(10^8)$	$O(10^6)$	127.51	59.43

From these results we can conclude that we can make great savings, both in the number of flops and in cpu time. This is probably due to the fact that we reduce the non-linear problem to a linear one via a choice of scaling; the Toolbox must solve the non-linear problem. We also note that the introduction of the bound on the conditioning of the problem is causing the extra work, since both methods are very efficient when $\omega_2^2 = 0$. This is a possible area for future work, since caution in

We have also shown that our methods are numerically efficient in comparison to a modern optimisation package.

In Chapters 5 and 6 we have given theory for choosing a new set of vectors that satisfy some criterion. Next we have to construct a feedback to achieve these vectors. This work is carried out in the next chapter.

Chapter 7

(Re)constr ction of feedback

In the previous two chapters we have described methods for selecting a set of vectors that best satisfy a set of minimisation criteria. The objective function decreases by differing amounts in relation to the weightings placed on the parameters, and depending on whether the new vectors chosen are restricted to be in certain subspaces. Whatever form of the algorithm is used, the result is a new set of vectors.

However, the whole point of output feedback and eigenstructure assignment is to calculate a feedback matrix, K, such that the closed loop system has desired eigenvalues and/or desired eigenvectors. Thus, our next step is to find such a feedback that best assigns the vectors found from the minimisation algorithm.

In the following sections, we describe the motivation and methods for constructing the feedback, give results on how the feedbacks are related, and the errors involved in using these constructions.

7.1 Meth ds f r calculating an initial right vect r set, V

In Chapters 4, 5 and 6 we gave various methods for constructing an initial set of vectors, V, to be used as the starting point for the restricted or unrestricted minimisation algorithm. To distinguish between these methods, we present the

following table,

method	V_1	V_2
1	restricted projection	from closed loop system after K generated from V_1 as in Section 4.3.7
2	restricted projection	as above, except some of Λ_2 changed, vectors for changed eigenvalues from their null spaces
3	restricted projection	from null spaces of chosen Λ_2
4	restricted projection, transform to real vectors	cycle I_n
5	restricted projection,	

by choosing a new set $\tilde{V}_2$; hence $\tilde{V}_2$ must be included in the construction of the feedback.

If w

7.2.2 Second (re)construction, K_2

As an alternative to just using the first construction, we adapt the partial eigenstructure assignment construction as in (7.2), to include the full set of vectors $\tilde{V} = [V_1, \tilde{V}_2]$. Now, $C\tilde{V} \in$

giving

$$E_1(I - \tilde{V}(C\tilde{V})^+(C\tilde{V})\tilde{W}^T) = E_2. \tag{7.13}$$

This appears to tell us that, for a specified eigenvalue, the correct left null space for K_1 lies within the correct left null space for K_2 (denote these null spaces as $\mathcal{T}_i$ and $\mathcal{U}_i$ respectively). From Figure 7.1 we can see that

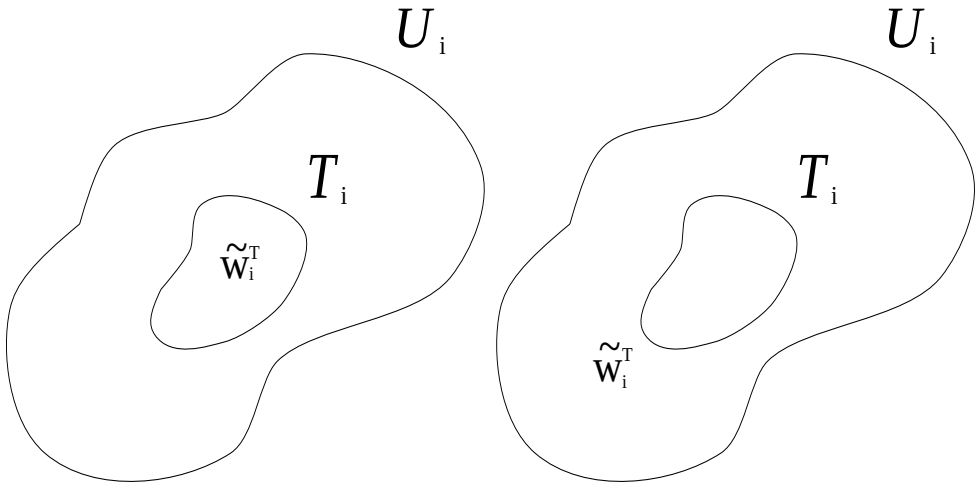


Figure 7.1: nullspaces

$$\begin{cases} \tilde{\mathbf{w}}_i^T \in \mathcal{T}_i & \Rightarrow \quad \tilde{\mathbf{w}}_i^T \in \mathcal{U}_i \\ \tilde{\mathbf{w}}_i^T \in \mathcal{U}_i & \not\Rightarrow \quad \tilde{\mathbf{w}}_i^T \in \mathcal{T}_i \end{cases} \tag{7.14}$$

respectively to the two diagrams. That is, if $\tilde{\mathbf{w}}_i^T$ lies in $\mathcal{T}_i$, then it also lies in $\mathcal{U}_i$; the converse does not hold in that $\tilde{\mathbf{w}}_i^T$ can lie in $\mathcal{U}_i$ without being in $\mathcal{T}_i$. Thus, although this does not guarantee that reducing $\|E_1\|$ automatically reduces $\|E_2\|$, it does show that they will simultaneously attain the same zero value when considered in the limit. This shows that using K_2, in addition to K_1, is worthwhile since we expect both of the errors to be reduced by the minimisation routine.

We can show, under certain conditions, that K_1 and K_2 are equivalent in the sense that they generate the same eigenstructure, but first need a theorem.

THEOREM 7.1 *The pseudo-inverse of a product of matrices $C \in \mathbb{C}^{p\times n}$, of full rank, and $V \in \mathbb{C}^{n\times n}$, unitary, is given by*

$$(CV)^+ = V^{-1}C^+. \tag{7.15}$$

Proof

Since V is square, invertible and unitary

$$V^{+} = V^{-1} = V^{H}, \tag{7.16}$$

where H denotes the complex conjugate transpose. Also, since C is of full rank, C^{+} satisfies the four Moore-Penrose conditions given in Definition 4.11, i.e.

$$(i) \quad CC^{+}C = C, \qquad (iii) \quad (CC^{+})^{H} = CC^{+}$$
$$(ii) \quad C^{+}CC^{+} = C^{+a\mu\mu}$$

(i)

$$K_2 \equiv B^+(\tilde{V}\tilde{\Lambda} - A\tilde{V})(C\tilde{V})^+$$

$$= B^+(\tilde{V}\tilde{\Lambda} - A\tilde{V})\tilde{V}^{-1}$$

so that $K_2 \equiv K_{2D}$. However, this is only true if $\tilde{V}$ is unitary, which is extremely unlikely. This indicates the worth in using K_2, in addition to K_1, since the benefit gained in scaling V_1 is lost for K_1.

7.2.5 Relationship between original feedback and K_2

If one of Methods 1 and 2 has been used to calculate the vectors for the starting point of the minimisation algorithm, then a K is generated. If we calculate the closed loop eigenstructure (V, Λ) then

$$(A + BKC)V = V\Lambda, \tag{7.25}$$

where

$$K = B^{+}(V_1\Lambda_1 - AV_1)(CV_1)^{-1}. \tag{7.26}$$

Now, for the second reconstruction

$$\begin{aligned} K_2 &= B^{+}\left([V_1, \tilde{V}_2]\begin{bmatrix} \Lambda_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix} - A[V_1, \tilde{V}_2]\right)(C[V_1, \tilde{V}_2])^{+} \\ &= B^{+}[(V_1\Lambda_1 - AV_1), (\tilde{V}_2\tilde{\Lambda}_2 - A\tilde{V}_2)][(CV_1, C\tilde{V}_2)]^{+}. \end{aligned} \tag{7.27}$$

Now by the theory of generalised inverses (see Ben-Israel and Grenville [5])

$$[A, B]^{+} = \begin{bmatrix} A^{+} \\ B^{+} \end{bmatrix}(AA^{+} + BB^{+})^{+}, \tag{7.28}$$

if $R(A) \bigcap R(B) = \{0\}$. Thus

$$\begin{aligned} K_2 &= B^{+}[(V_1\Lambda_1 - AV_1), (\tilde{V}_2\tilde{\Lambda}_2 - A\tilde{V}_2)]\begin{bmatrix} (CV_1)^{-1} \\ (C\tilde{V}_2)^{+} \end{bmatrix}[(CV_1)(CV_1)^{-1} + (C\tilde{V}_2)(C\tilde{V}_2)^{+}]^{+} \\ &= B^{+}[(V_1\Lambda_1 - AV_1)(CV_1)^{-1} + (\tilde{V}_2\tilde{\Lambda}_2 - A\tilde{V}_2)(C\tilde{V}_2)^{+}][I + (C\tilde{V}_2)(C\tilde{V}_2)^{+}]^{+} \\ &= [K + B^{+}(\tilde{V}_2\tilde{\Lambda}_2 - A\tilde{V}_2)(C\tilde{V}_2)^{+}][I + (C\tilde{V}_2)(C\tilde{V}_2)^{+}]^{+}, \end{aligned} \tag{7.29}$$

provided $R(CV_1) \bigcap R(C\tilde{V}_2) = \{0\}$. Here we hav

The idea is to augment the system into a linear least squares form to calculate $(K_3, \tilde{\Lambda})$. This system is overdetermined, proven using the following small theorem.

THEOREM 7.3 *If $n, m, p \in \mathbb{N}$ (the set of natural numbers) and $n > m$ a n*

This form of the diagonal solver is called the complex solver because complex eigenvalues are allowed to appear explicitly on the diagonal of $\tilde{\Lambda}$ in the form

$$\tilde{\lambda}_{jj} = \alpha_j + \mathbf{i}\beta_j, \tag{7.41}$$

where $\mathbf{i} = (-1)^{\frac{1}{2}}$. The system given in (7.37) is written in the form

$$M\mathbf{k} = \mathbf{a}, \tag{7.42}$$

where $M = [M_1$

which is just the n-dimensional identity matrix with n rows of zeros augmented between each of its rows. The rows with a one in them represent the coefficients of the diagonal elements of $\tilde{\Lambda}$; the n rows of zeros represent the n zeros that appear on the off-diagonal of $\tilde{\Lambda}$ between each diagonal element.

The solution vector, $\mathbf{k}$, is made up of the unknown elements of K_3 and $\tilde{\Lambda}$, and hence $\mathbf{k} \in \mathbb{C}^{mp \times n}$. The right hand side of (7.42) contains all of the elements of $\tilde{A}$, i.e. the coefficients a_{ij}; thus $\mathbf{a} \in \mathbb{C}^{n^2}$. These two components appear in the form

$$\mathbf{k} = \begin{bmatrix} k_{11} \\ \vdots \\ k_{1m} \\ k_{21} \\ \vdots \\ k_{2m} \\ \vdots \\ k_{p1} \\ \vdots \\ k_{pm} \\ \hline \tilde{\lambda}_{11} \\ \vdots \\ \tilde{\lambda}_{nn} \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_{11} \\ \vdots \\ a_{1n} \\ a_{21} \\ \vdots \\ a_{2n} \\ \vdots \\ a_{n1} \\ \vdots \\ a_{nn} \end{bmatrix}. \tag{7.45}$$

The system thus takes the general form

$$\left[\begin{array}{c|c} \begin{array}{c}\text{coefficients } BC \\ \text{of } K\end{array} & \text{coefficients of } \tilde{\Lambda} \end{array}\right] \left[\begin{array}{c} K\text{'s components} \\ \text{in list form} \\ \hline \tilde{\Lambda}\text{'s components} \\ \text{in list form} \end{array}\right] = \begin{bmatrix} A \text{ in} \\ \text{list form} \end{bmatrix}. \tag{7.46}$$

Here we have shown how to write (7.33) into a linear least squares form, which can be solved using a QR (or SVD) method. Since the system is overdetermined in general, it is very unlikely that the system has an exact solution. Thus, we will have errors between what we want and what we achieved; these are covered in the error analysis section.

Note that throughout this section, we have defined components as being members of the complex set. For the results of the unrestricted minimisation, all of the components are actually real. This section was put in its most general form because it may be applied to the vectors arising from the restricted minimisation, which may be complex.

7.3.3 Real solver formulation

In this section, we assume that we have a real set of vectors, $\tilde{V}$. We know from Section 6.6 that, if we used either Method 4 or 5 to find the initial vector set, then any complex conjugate vectors in V are transformed into their real representation. At the end of the minimisation, we do not know whether the real vectors in $\tilde{V}_2$ should correspond to all real eigenvalues, or some complex conjugate ones. But, we do know that any vectors in V_1 that corresponded to complex eigenvalues are unchanged by the minimisation. Thus, when solving here we must let the corresponding eigenvalues appear in their real representation of complex conjugate eigenvalues.

The method here is called the real solver since complex eigenvalues appear as real 2×2 blocks on the diagonal of $\tilde{\Lambda}$. If the j^{th} complex conjugate pair is in the form

$$\tilde{\lambda}_j = \alpha_j$$

to

$$\begin{cases} -\sum_{t=1}^{m}\sum_{s=1}^{p} b_{it}k_{ts}c_{sj} + \tilde{\lambda}_{ij} = a_{ij} \\ \text{subject to } \tilde{\lambda}_{ij} = 0 \begin{cases} j > i+1 & (i \text{ odd}) \\ j < i & \\ j < i-1 & (i \text{ even}) \\ j > i & \end{cases} \qquad (i,j = 1,\ldots,n). \\ \mathrm{Re}(\tilde{\lambda}_{ij}) < 0 \quad (i = j) \end{cases} \tag{7.49}$$

The desired diagonal matrix of eigenvalues is then in the form

$$\tilde{\Lambda} = \left[\begin{array}{cc|c|cc|c} \alpha_1 & \beta_1 & & & & \\ -\beta_1 & \alpha_1 & & & & \\ \hline & & \ddots & & & \\ & & & \ddots & & \\ \hline & & & \alpha_j & \beta_j & \\ & & & -\beta_j & \alpha_j & \\ \hline & & & & \ddots & \\ & & & & & \ddots \\ \hline & & & & & \end{array}\right] \alpha jc\mu m\mu ImD\mu\mu B\mu\mu.j\mu\mu\beta$$

Obviously $\mathbf{a}$ is the same as in (7.45); but $\mathbf{k}$ has extra $\tilde{\lambda}_{ij}$'s,

$$\mathbf{k} = \left[\begin{array}{c} k_{11} \\ \vdots \\ k_{1m} \\ k_{21} \\ \vdots \\ k_{2m} \\ \vdots \\ k_{p1} \\ \vdots \\ k_{pm} \\ \hline \tilde{\lambda}_{11} \\ \tilde{\lambda}_{12} \\ \tilde{\lambda}_{21} \\ \tilde{\lambda}_{22} \\ \vdots \\ \tilde{\lambda}_{n-1,n-1} \\ \tilde{\lambda}_{n-1,n} \\ \tilde{\lambda}_{n,n-1} \\ \tilde{\lambda}_{n,n} \end{array}\right]. \tag{7.52}$$

Again, we have reduced the system to a linear least squares form to be solved by the QR (or similar) method. Note that if n is odd, then there will be at least one real eigenvalue represented, or if we desire a mixture of real and complex eigenvalues, then we just use the previous theory, but with fewer 2×2 blocks. For example, if $n = 5$, our desired $\tilde{\Lambda}$ may be

$$\tilde{\Lambda} = \begin{bmatrix} \tilde{\lambda}_1 & 0 & 0 & 0 & 0 \\ 0 & \tilde{\lambda}_2 & 0 & 0 & 0 \\ 0 & 0 & \tilde{\lambda}_{33} & \tilde{\lambda}_{34} & 0 \\ 0 & 0 & \tilde{\lambda}_{43} & \tilde{\lambda}_{44} & 0 \\ 0 & 0 & 0 & 0 & \tilde{\lambda}_5 \end{bmatrix}. \tag{7.53}$$

We have now given two methods for finding a feedback and a new eigenvalue set; we next summarise the algorithm in listed form.

7.3.4 lgorithm for diagonal solver

The algorithm for finding $(K_3, \tilde{\Lambda})$ from knowing only the system matrices and a new set of minimisation vectors is the same for both the complex and real diagonal solver. They only differ in forming the components M_2 and $\mathbf{k}$. Summarising, the

7.3.5 Solver error analysis

As just mentioned, we are not able in general to solve Problem 5 exactly; more explicitly, in using the diagonal solver, we are trying to solve

$$\begin{cases} \min_{\tilde{K}_3, \tilde{\Lambda}} \|(A + BK_3C)\tilde{V} - \tilde{V}\tilde{\Lambda}\|_F^2 \\ \text{subject to } \tilde{} \end{cases}$$

LEMMA 7.5 *Let $\tilde{A} = A+BK_3C-E$, then if λ is an eigenvalue of $\tilde{A}+E$ ($\equiv A_c$), and $\tilde{V}^{-1}\tilde{A}\tilde{V} = diag(\tilde{\lambda})$, then*

$$\min|\tilde{\lambda} - \lambda| \leq \kappa_F(\tilde{V})\|E\|_F. \tag{7.61}$$

Proof The proof follows directly from Theorem 7.4.

However, in general, this is not a tight bound since it measures the distance of each eigenvalue in relation to the conditioning of all of the eigenvectors. Thus, a single ill-conditioned eigenvalue may make the whole system appear badly conditioned.

We may tighten this bound by reducing the conditioning of the vectors found, a criteria that is included in the unrestricted minimisation and can be affected by increasing the relative weighting of ω_2^2. In doing this, we lose some of the ability to reduce the left vector matching to a satisfactory level. This, again, is the problem of selecting the parameters to obtain the desired trade-off between the set minimisation criteria.

We can see the performance of the solver in the examples that follow in Chapter 8. First we show how the solver can be adapted to allow for equality constraints.

7.3.6 Constrained diagonal solver

So far in this section, we have given a method for constructing a feedback matrix from the system matrices and a set of vectors. Two variations have been outlined that illustrate how to deal with complex eigenvalues appearing explicitly or in their real, block form. The problem was written in the form

$$M\mathbf{k} = \mathbf{a}, \tag{7.62}$$

and solved in a least squares sense. In doing this we performed an unconstrained linear least squares optimisation. This is fine if the resulting eigenvalues of the closed loop system are satisfactory.

However, as shown in Section 7.3.5, we cannot expect to solve the problem exactly. Thus, the values obtained for $\tilde{\lambda}_{ij}$ will not be the eigenvalues of $A+BK_3C$, but of a perturbation to this. The result is that we may obtain eigenvalues of

Thus, assuming we have the system (7.33) written in the form (7.62), our optimisation problem becomes

$$\min_{\mathbf{k}} \|M\mathbf{k} - \mathbf{a}\|_2^2$$

$$\text{subject to} \begin{cases} k_i < 0 & \left(\begin{array}{l}\text{for the } i = mp+1, \ldots, n \\ \text{that are the } \mathrm{Re}(\tilde{\lambda}_{ij})\end{array}\right) \\ \begin{cases} k_i - k_{i+1} = 0 \\ k_{i+2} + k_{i+3} = 0 \end{cases} & \left(\begin{array}{l}\text{for the } i = mp+1, \ldots, n \\ \text{that represent a cc pair}\end{array}\right) \end{cases} . \quad (7.68)$$

But, too many constraints will influence the overall accuracy of the (re)construction and the benefit gained from running the original minimisation algorithm. By how much will be seen in the full examples in the next chapter.

7.4 Conclusions

In this chapter, we have given various methods for constructing a feedback that best assigns the new vector set, $\tilde{V}$, that results from either the restricted or unrestricted minimisation algorithm.

For calculating the feedback following the restricted minimisation we gave two constructions based on full and partial eigenstructure assignment. We analysed the errors in each and gave conditions for the two feedback constructions to be equivalent in the sense that they generate the same closed loop eigenstructure.

For the calculation of the feedback following the unrestricted minimisation we devised a routine that found a set of eigenvalues in addition to the feedback. This was formulated to allow complex conjugate eigenvalues to appear explicitly on the diagonal of $\tilde{\Lambda}$, or as real 2×2 diagonal blocks. We derived an expression for the error in using this construction, and showed how the solver could be modified to allow for equality constraints.

In the next chapter we test all of our minimisation methods and these feedback constructions on full aircraft examples.

Chapter 8

F ll examples

8.1 Intr ducti n

In the previous two chapters we have devised two minimisation routines that reduce the level of input decoupling via the left eigenvectors by iterating through the unassigned right eigenvectors. In these routines, we also included controls on the conditioning of the system and the distance of the left vectors from their correct subspaces. We have shown, through examples, how the minimisation algorithm works, with various values of the weighting parameters tested. The vectors resulting from either minimisation algorithm did not correspond to any closed loop eigenstructure; in Chapter 7 we gave methods for reconstructing the feedback to best obtain these vectors.

Here we give examples to demonstrate all of the theory, from the specification of the problem and the calculation of the initial vector set, through to the running of the minimisation and finally the reconstruction of the feedback.

8.2 Example 1

The first example here is taken from Andry *et al.* [1], and is a lateral axis model of an L−1011 aircraft at cruise condition. The model includes actuator dynamics and a washout filter on yaw rate. The state vector, input vector and output

vector are given by

$$\mathbf{x} = \begin{bmatrix} \delta_r \\ \delta_a \\ \\ r \\ p \\ \beta \\ x_7 \end{bmatrix} \begin{array}{l} \text{rudder deflection} \\ \text{aileron deflection} \\ \text{bank angle} \\ \text{yaw rate} \\ \text{roll rate} \\ \text{sideslip angle} \\ \text{washout filter} \end{array}, \tag{8.1}$$

$$\mathbf{u} = \begin{bmatrix} \delta_{r_c} \\ \delta_{r_a} \end{bmatrix} \begin{array}{l} \text{rudder command} \\ \text{aileron command} \end{array}, \quad \mathbf{y} = \begin{bmatrix} r_{\text{WO}} \\ p \\ \beta \\ \\ \phi \end{bmatrix} \begin{array}{l} \text{washed out yaw rate} \\ \text{roll rate} \\ \text{sideslip angle} \\ \\ \text{bank angle} \end{array}, \tag{8.2}$$

respectively. A word is needed here on the meaning of the washout filter, x_7. A yaw damper is used to ensure that the dutch roll damping is of an acceptable level. However, this does not completely remove the effect of the initial disturbance in yaw rate as there are non-zero steady state values. Also, the system tends to oppose any change in yaw rate, even if it has been commanded. Thus, the signal proportional to yaw rate, being used as a feedback signal to the controller, is first passed through a washout network to differentiate the signal from the yaw rate gyroscope. We can see the first output (from $\mathbf{y}$) is the washed-out yaw rate, and is a combination of yaw rate and the washout filter (as can be seen from the first row of C).

The system matrices are given by

$$A = \begin{bmatrix} -20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.744 & -0.032 & 0 & -0.154 & -0.0042 & 1.54 & 0 \\ 0.337 & -1.12 & 0 & 0.249 & -1 & -5.2 & 0 \\ 0.02 & 0 & 0.0386 & -0.996 & -0.0003 & -0.1170 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & -0.5 \end{bmatrix}, \tag{8.3}$$

$$B = \begin{bmatrix} 20 & 0 \\ 0 & 25 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{8.4}$$

The open loop eigenvalues of the system are

open-loop eigenvalue	mode	frequency	damping	sensitivity
-20	rudder	20	1	1.0
-25	aileron	25	1	1.0
$-0.0882 \pm 1.2695i$	dutch roll	1.27	0.07	2.8
-1.0855	roll subsidence	1.09	1	4.0
-0.0092	spiral	0.009	1	3.3
-0.5	washout filter	0.5	1	1.1

with the corresponding desired mode output coupling vectors

$$G_{0d} = \begin{bmatrix} x & x & 0 & 0 \\ 0 & 0 & x & x \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{8.8}$$

These mode output coupling vectors are chosen so that the sideslip angle and roll rate response are decoupled; this choice for G_{0d} also decouples bank angle and yaw rate. Thus, for each feedback constructed in this example, we give the closed loop response to an initial sideslip angle of 1^o (all other initial conditions are zero) and the closed loop response to an initial bank angle of 1^o (all other initial conditions are zero).

The first two vectors in G_{0d} are orthogonal to the last two vectors and a decoupling of the roll mode from the dutch roll will be realised if G_{0d} is achieved. The corresponding desired mode input coupling vectors are

$$G_{1d} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}. \tag{8.9}$$

This choice couples the first input (sideslip angle demand) to the washed out yaw rate and decouples it from roll rate and bank angle; the second input (bank angle demand) is coupled to the roll rate and is decoupled from the washed out yaw rate and sideslip angle.

Thus, for each feedback construction, we also give the responses to a step input on sideslip angle and bank angle. However, it should be noted that we have not included a feedforward command tracker in these responses. A tracker is needed for a comparison of the input responses between different feedbacks, but feedforward changes the obtained eigenvectors and would therefore affect the results of our minimisation. Further work needs to be carried out to include the use of a feedforward command tracker in our methods. We include the input response diagrams we;nglenpTorw

To find an initial closed loop system, we use Method 1, as in Section 4.3.7; the feedback gains are calculated as

$$K = \begin{bmatrix} 8.0313 & -0.2077 & -22.1264 & -0.5381 \\ 3.0432 & 0.9281 & -12.8538 & 4.0945 \end{bmatrix}. \tag{8.10}$$

The closed loop eigenvalues for $A + BKC$ are

closed-loop eigenvalue	mode	frequency	damping	sensitivity
$-6 \pm i$	dutch roll	6.0828	0.9864	701.97
$-1 \pm 2i$	roll	2.2361	0.4472	3.01
-23.9954	aileron	23.9954	1	10.92
-8.1679	rudder	8.1679	1	775.41
-0.6077	washout filter	0.6077	1	3.27

We can see that the damping of the roll mode is relatively low in comparison to the dutch roll. The dutch roll eigenvalue has a very large condition number, as does that of the rudder mode; the others are all quite well conditioned. The condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 6.66 \times 10^4. \tag{8.11}$$

The normalised mode output coupling vectors corresponding to the four desired eigenvalues are

$$G_{0a} = \begin{bmatrix} 1 & -0.0009 \pm 0.0024i \\ 0 & 1 \\ 0.1265 \pm 0.0235i & -0.0036 \pm 0.0051i \\ 0 & -0.2000 \pm 0.4000i \end{bmatrix}. \tag{8.12}$$

We can see that the exact desired decoupling cannot be achieved in the roll mode, although the level_of coupling is small. The results giv

From the results we can see that the mode output coupling vectors, G_{0a} have been achieved to a satisfactory level, but the mode input coupling vectors, G_{1a} have not been achieved. The first input is exciting inappropriate modes. We require that the real and imaginary parts of those elements in G_{1a} that correspond to a specified zero in G_{1d} to be $\mathrm{O}(10^{-2})$ or less (i.e < 0.1). The errors in the matching of the mode output and input coupling vectors are

$$\begin{aligned} \|G_{0d} - G_{0a}\|_F^2 &= 4.5860 \times 10^{-4} \\ \|G_{1d} - G_{1a}\|_F^2 &= 23.0735 \end{aligned} , \tag{8.14}$$

respectively. We see that there is a difference of $O(10^6)$ between the two errors; the aim is to minimise the error, $\|G_{1d} - G_{1a}\|_F^2$, while retaining the accuracy in the mode output coupling vectors.

The original closed loop output and input responses are given in Figures 8.1 and 8.2, respectively.

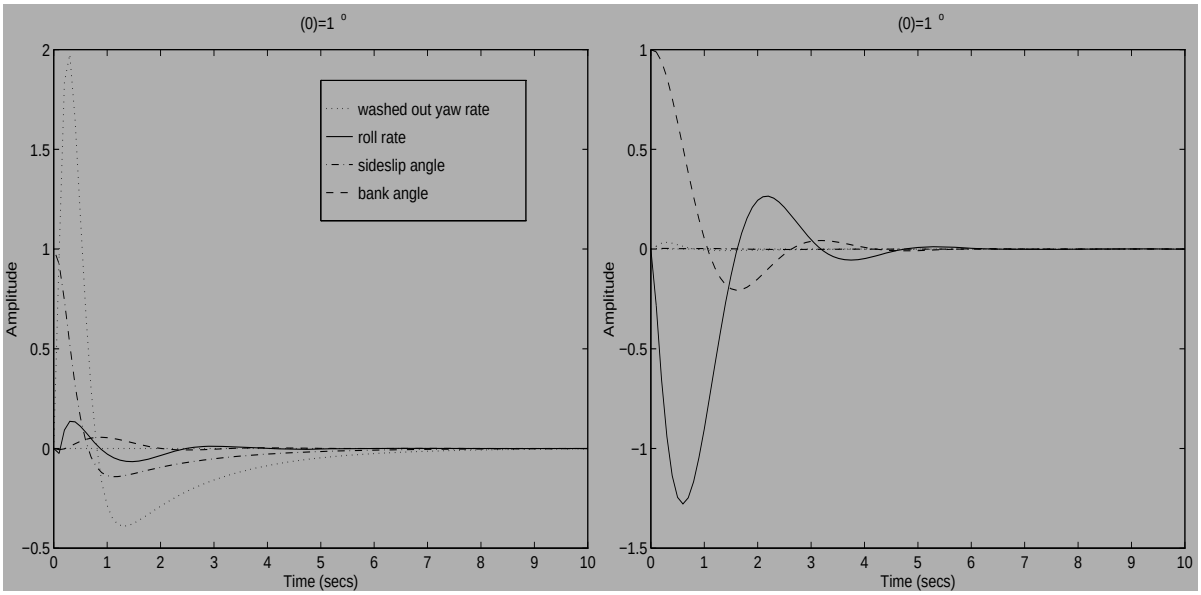
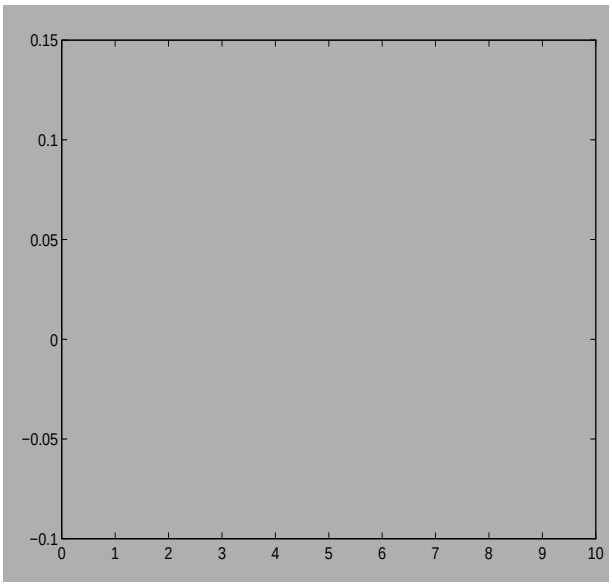


Figure 8.1: Original closed loop output responses



8.3.1 pply restricted minimisation algorithm (for decoupling)

Here we demonstrate the use of the restricted minimisation algorithm; we use Method 1 to find an initial set of vectors and choose to retain the unassigned set of eigenvalues, Λ_2. It is likely that, since $\lambda_{1,2} = -6 \pm i$ and $\lambda_5 = -8.1679$ are poorly conditioned, they will move quite considerably unless we put a high relative weighting on the left eigenspace error. However, we are attempting to improve the level of input decoupling, hence we choose the weightings

$$(\omega_1^2, \omega_2^2, \omega_3^2) = (1 \times 10^4, 1, 1), \tag{8.15}$$

so that, even though we are primarily trying to match the left eigenvectors, we include weightings on the eigenvector conditioning and the left eigenspace error. The results of the minimisation are

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$	$\sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2$
0	$2.9090e+06$	$2.3074e+01$	$6.6621e+04$	$1.1266e-23$
1	$5.0317e+05$	$2.2273e+01$	$4.1143e+04$	$5.0970e+04$
2	$6.1413e+04$	$5.2220e+00$	$7.9897e+03$	$5.3914e+02$
3	$6.1198e+04$	$5.2029e+00$	$7.9948e+03$	$5.0442e+02$

From the new set of minimisation vectors, $\tilde{V} = [$

which has the new closed loop eigenvalues

closed-loop eigenvalue	mode	frequency	damping	sensitivity
$-7.29 \pm 9.28i$	dutch roll	11.7978	0.6180	82.21
$-0.7079 \pm 1.0144i$	roll	1.2370	0.5723	3.45
-24.6274	aileron	24.6274	1	6.77
-5.5598	rudder	5.5598	1	84.41
-0.5859	washout filter	0.5859	1	2.79

.

From these results we can clearly see that the dutch roll and rudder eigenvalues have moved by a lot, as expected. This is reflected in the higher feedback gains in K_1. We have lost some of the damping in the dutch roll mode, but have increased the damping on the rudder mode. A significant improvement has also been made in the conditioning of the individual eigenvalues, specifically the dutch roll and rudder modes. The condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 380.8429, \tag{8.18}$$

which is a reduction of $O(10^2)$.

The corresponding mode input coupling vectors are

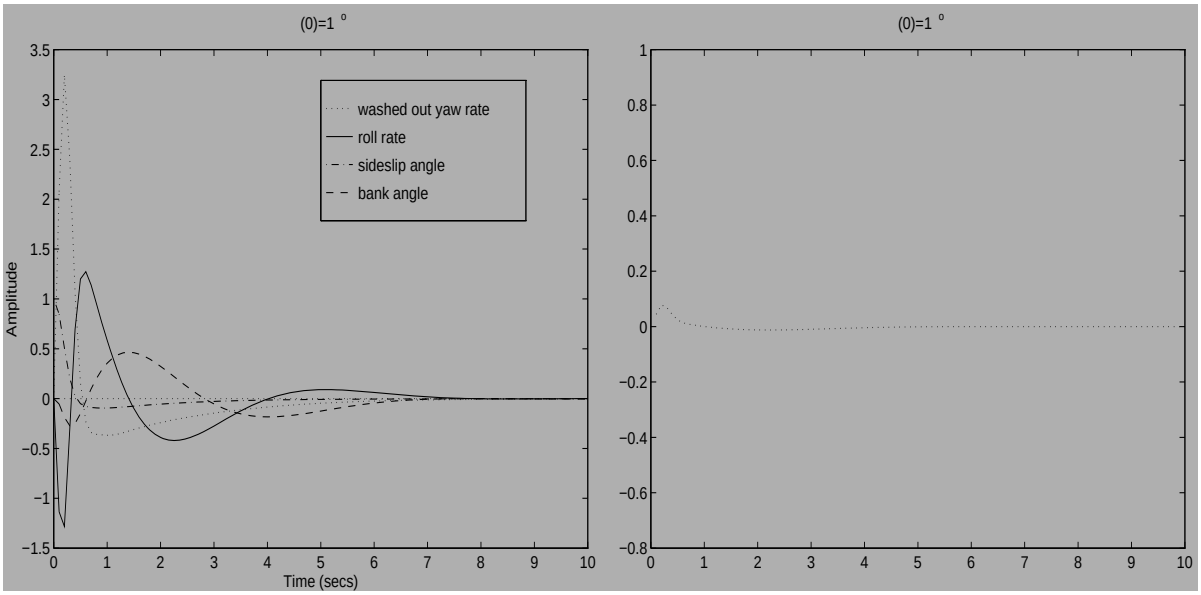
$$G_{1a} = \begin{bmatrix} 1 & -0.0130 - 0.0133i \\ 1 & -0.0130 + 0.0133i \\ 0.0271 + 0.0443i & 1 \\ 0.0271 - 0.0443i & 1 \end{bmatrix}, \tag{8.19}$$

which, surprisingly, are slightly better than in (8.16). Thus, we have calculated a feedback that gives the desired level of input decoupling, but have we retained the initial output decoupling. The new closed loop mode output coupling vectors are

$$G_{0a} = \begin{bmatrix} 1 & 0 \mp 0.0296i \\ -0.687 \pm 0.123i & 1 \\ 0.027 \mp 0.064i & 0.002 \pm 0.007i \\ 0.028 \mp 0.052i & -0.463 \mp 0.663i \end{bmatrix}, \tag{8.20}$$

so that we have introduced a small level of coupling between sideslip angle and roll rate.

The new closed loop output and input responses are given in Figures 8.3 and 8.4, respectively. In Figure 8.3 we can see the increased output coupling between



addition to increasing the robustness of the system. This illustrates the trade-off between the levels of input and output coupling that needs to be considered.

8.3.2 pply restricted minimisation algorithm (for conditioning)

In Section 8.3.1, we showed that the input decoupling could be reduced with a high relative weighting on the left vector matching. The conditioning of the problem was also reduced; here we attempt to reduce the conditioning further. The starting point for the minimisation algorithm is again the results of Method 1. We select the weightings

$$(\omega_1^2, \omega_2^2, \omega_3^2) = (0, 1, 0). \tag{8.21}$$

The results of the minimisation are

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$	$\sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2$
0	$6.0165e+05$	$2.3074e+01$	$6.6621e+04$	$1.1266e-23$
1	$4.1757e+04$	$1.5982e+02$	$1.7551e+04$	$1.3432e+03$
2	$7.9087e+03$	$5.5553e+00$	$7.6382e+03$	$1.5965e+03$
3	$7.6166e+03$	$5.3810e+00$	$7.4958e+03$	$1.6458e+03$
4	$7.6055e+03$	$5.3634e+00$	$7.4904e+03$	$1.6347e+03$
5	$7.6055e+03$	$5.3624e+00$	$7.4904e+03$	$1.6342e+03$

Again, the first reconstruction for the feedback is used,

$$K_1 = \begin{bmatrix} 5.5012 & -0.3948 & -48.6834 & -1.0257 \\ -3.7107 & 0.4436 & 0.2810 & 1.4630 \end{bmatrix}, \tag{8.22}$$

which has the new closed loop eigenvalues

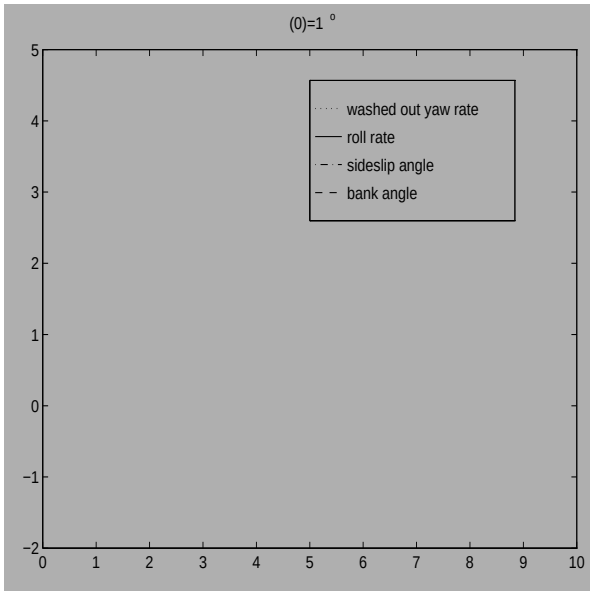
closed-loop eigenvalue	mode	frequency	damping	sensitivity
$-2.0545 \pm 6.2736i$	dutch roll	6.6015	0.3112	39.01
$-0.6790 \pm 0.9819i$	roll	1.1938	0.5688	7.79
-24.7621	aileron	24.7621	1	14.9394
-15.9402	rudder	15.9402	1	77.2805
-0.6016	washout filter	0.6016	1	6.6302

As required, we have reduced the sensitivity of the system further, specifically the dutch roll and rudder modes, with slight increases in the other modes. The condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 256.5804. \tag{8.23}$$

However, this has been achieved at the expense of the performance. The dutch roll mode now has low damping and we need to look at the levels of input and output decoupling. The closed loop mode input and output coupling vectors are

$$G_{1a} = \begin{bmatrix} 1 & -0.0033 - 0.0487i \\ 1 & -0.0033 + 0\theta\eta ui T\theta G\eta T\theta T\theta T\theta T\theta\delta T\theta TT\theta T\theta\eta T\theta T\theta\eta u\eta\kappa T\theta \end{bmatrix}$$


(0)=1 °
washed out yaw rate
roll rate
sideslip angle
bank angle
5
4
3
2
1
0
−1
−2
0
1
2
3
4
5
6
7
8
9
10

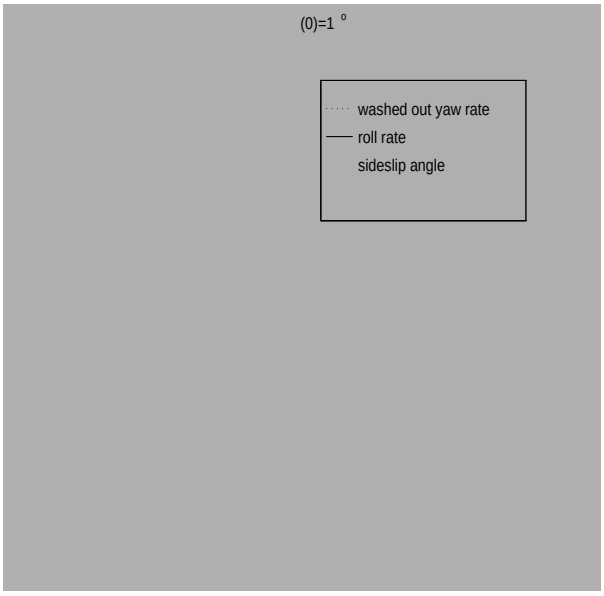
which has the new closed loop eigenvalues

closed-loop eigenvalue	mode	frequency	damping	sensitivity
$-0.9274 \pm 0.3107i$	dutch roll	0.9780	0.9482	16.198
$-0.9849 \pm 2.0116i$	roll	2.2397	0.4397	3.19
-23.9560	aileron	23.9560	1	7.52
-12.9226	rudder	12.9226	1	40.36
-6.0679	washout filter	6.0679	1	48.74

.

We can see that the eigenvalues are satisfactory, their sensitivities are particularly good. The condition number of the eigenvectors of the closed loop system is

$$\kappa$$


(0)=1 °
washed out yaw rate
roll rate
sideslip angle

8.3.4 Results summary

For this example, using the specified set of eigenvalues given in (8.7), we have successfully applied the restricted and unrestricted minimisation algorithms to reduce the level of input decoupling of the system; this has sometimes led to an increase in the levels of output coupling. We have also reduced the sensitivity of the system. These results demonstrate the trade-off between the input coupling, the output coupling and the robustness of the system.

For a different set of desired eigenvalues we next demonstrate the possibility of a system becoming unstable, thus justifying the need to control all of the eigenvalues.

8.3.5 ssign different eigenvalue set

It is unlikely that our routines perform a global minimisation for this problem. To illustrate this we select a new set of desired eigenvalues

$$\Lambda_p = \begin{cases} -7 \pm 5i \\ -15 \pm 4i \end{cases} . \tag{8.31}$$

We again perform partial eigenstructure assignment as in Section 4.3.7, generating the feedback

$$K = \begin{bmatrix} 9 \\ \\ \end{bmatrix}$$

This illustrates the problem in using partial eigenstructure assignment that the uncontrolled modes may be unstable. Our method uses full eigenstructure assignment and we can therefore overcome this problem. The condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 6.4330 \times 10^4. \tag{8.33}$$

The corresponding closed loop mode output and input coupling vectors are

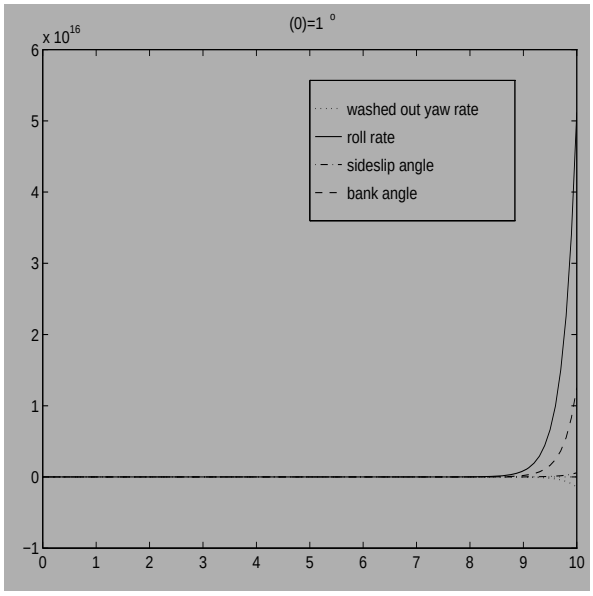
$$G_{0a} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0.0673 \pm 0.0628i & 0.0009 \pm 0.0001i \\ 0 & -0.0622 - 0.0166i \end{bmatrix}, \tag{8.34}$$

$$G_{1a} = \begin{bmatrix} 1 & 0.0754 + 0.0108i \\ 1 & 0.0754 - 0.0108i \\ -0.1469 + 0.0427i & 1 \\ -0.1469 - 0.0427i & 1 \end{bmatrix}, \tag{8.35}$$

respectively. The errors in the matching of the mode output and input coupling vectors are

$$\begin{matrix} \|G_{0d} - G_{0a}\|_F^2 = 3.7495 \times 10^{-4} \\ \|G_{1d} - G_{1a}\|_F^2 = 5.0074 \end{matrix}, \tag{8.36}$$

respectively. Again the output decoupling is attained to the desired level, but there is some coupling apparent in the inputs. The original closed loop output and input responses for the new assigned eigenvalue set are given in Figures 8.9 and 8.10, respectively.


(0)=1 °
x 10^16
6
5
4
3
2
1
0
−1
0
1
2
3
4
5
6
7
8
9
10
washed out yaw rate
roll rate
sideslip angle
bank angle

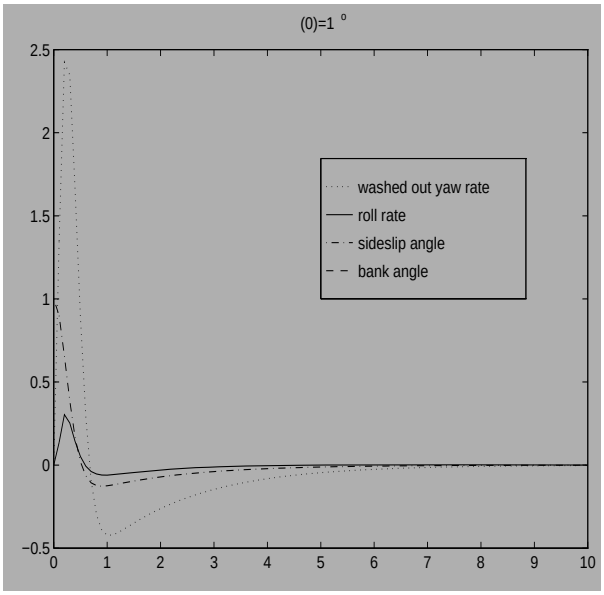
8.3.6 pply restricted minimisation algorithm (for decoupling)

Since the closed loop system is unstable, we allow the eigenvalues to vary freely by putting a zero weighting on the left eigenspace error. Our primary aim is to reduce the level of input decoupling; our choice of weighting parameters is therefore

$$(\omega_1^2, \omega_2^2, \omega_3^2) = (1 \times 10^5, 1, 0). \tag{8.37}$$

We run the restricted minimisation algorithm, implementing the alternative scaling method as in Section 5.10, giving

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$	$\sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2$
0	$5.3835e+05$	$5.0074e+00$	$6.4330e+04$	$7.1317e-26$
1	$3.3322e+04$	$1.2352e-01$	$9.5194e+03$	$6.3035e+03$
2	$2.5921e+04$	$5.7446e-02$	$9.4707e+03$	5.9998dllTTf]TD]eTj]TTTD]Tc]+0


(0)=1 °
2.5
2
1.5
1
0.5
0
−0.5
0
1
2
3
4
5
6
7
8
9
10
washed out yaw rate
roll rate
sideslip angle
bank angle

8.3.7

However, the new mode input and output coupling v

8.3.8 pply unrestricted minimisation algorithm (for conditioning)

We have managed to obtain satisfactory performance in terms of input/output decoupling, but have not significantly reduced the sensitivity of the system. Here we attempt to reduce the conditioning by weighting out the left vector matching requirement from the algorithm. Again Method 5 is used to find an initial, real vector set V. Thus, the minimisation is run with the weightings

$$(\omega_1^2, \omega_2^2) = (0, 1), \tag{8.47}$$

with the alternative scaling theory implemented, giving

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$3.0820e+00$	$4.0188e+00$	$4.1182e+02$
	;0188e		

The new closed loop eigenvalues are

closed-loop eigenvalue	frequency	damping	sensitivity
$-0.30 \pm 0.63i$	0.7044	0.4327	5.525
-17.1446	17.1446	1	16.1272
-8.7328	8.7328	1	14.0118
-19.2403	19.2403	1	3.4801
-0.9323	0.9323	1	6.385
-0.1116	0.1116	1	1.6359

We can see that the system has now been stabilised and the individual eigenvalue sensitivities are very low. The condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 63.3335, \tag{8.50}$$

which has been reduced by a further order. The input and output coupling vectors are not included here as they were not weighted in the algorithm, and have not improved. The closed loop output and input responses are given in Figures 8.15 and 8.16, respectively.

8.3.9 Example 1 conclusions

$$\mathbf{u} = \begin{bmatrix} \theta_d \\ u_d \\ \gamma_d \end{bmatrix} \begin{matrix} \text{pitch attitude demand} \\ \text{airspeed demand} \\ \text{flightpath angle} \end{matrix}, \quad \mathbf{y} = \begin{bmatrix} \theta \\ U_0 \\ \gamma \\ q \end{bmatrix} \begin{matrix} \text{pitch attitude} \\ \text{airspeed} \\ \text{flightpath angle} \\ \text{pitch rate} \end{matrix}, \quad (8.52)$$

respectively. The system matrices are given by

$$A(\text{cols. } 1:5) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -0.00000393 & -0.6917 & 0.1692 & -0.0181 & -7.3400 \\ -0.5561 & -0.4872 & -0.0416 & -0.0825 & -0.0426 \\ -0.0781 & 3.4690 & 0.0180 & -0.2949 & -0.4084 \\ 0 & 0 & 0 & 0 & -20.0000 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.00001131 & -0.000002486 & 0 \\ 0 & 0 & -0.00001179 & -0.00001045 & 0 \\ 0 & 0 & 0.0008166 & 0.0001073 & 0 \end{bmatrix},$$

$$A(\text{cols. } 6:10) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1580 & 62.9200 & -66.3600 & -11.9700 \\ 0 & -0.3736 & 14.5800 & 5.6640 & 2.9470 \\ 0 & -0.1201 & -28.1300 & -7.1830 & -3.8990 \\ 0 & 0 & 0 & 0 & 0 \\ -10.0000 & 0 & 0 & 0 & 0 \\ 0 & -4.9990 & 0 & 0 & 0 \\ 0 & 0 & -3.7470 & 2.5360 & 1.1240 \\ 0 & 0 & -0.0006261 & -2.7110 & 0.8217 \\ 15.5800 & 0 & -58.2000 & 0 & -13.3300 \end{bmatrix} \quad (8.53)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 20.0000 & 0 & 0 \\ 0 & 5.4060 & 8.4120 \\ 0 & -522.4000 & 335.8000 \\ 0 & 0 & 0 \\ 0 & & \end{bmatrix}.$$

The closed loop eigenvalues for $A + BKC$ are

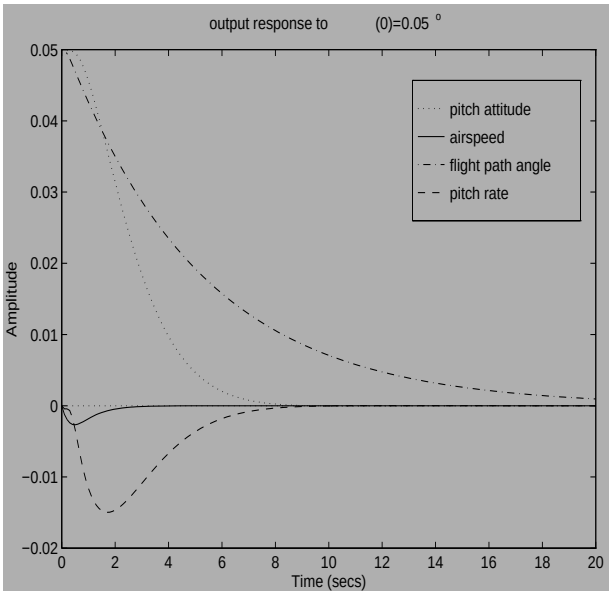
closed-loop eigenvalue	frequency	damping	sensitivity
$-0.7000 \pm 0.3000i$	0.7616	0.9191	177.43
-3.8	3.8	1	41.25
-0.2	0.2	1	4.98
-9.3152	9.3152	1	9.12
-4.8516	4.8516	1	27.48
-1.4618	1.4618	1	9.95
-19.1126	19.1126	1	1.3
$-7.8371 + 5.7006i$	9.6910	0.8087	16.14

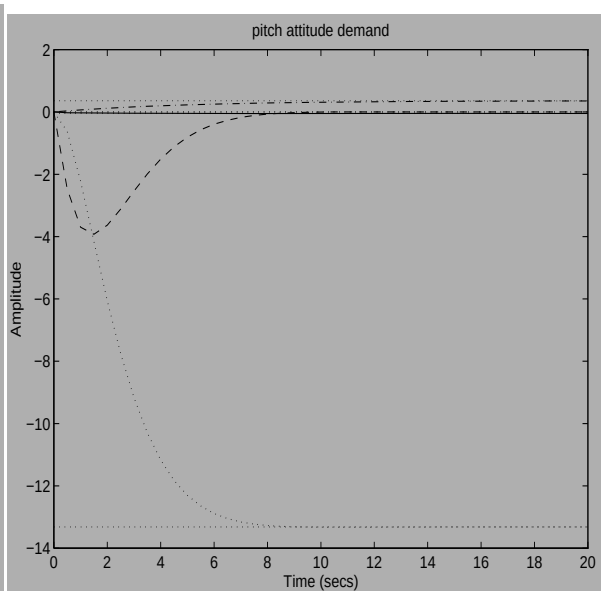
.

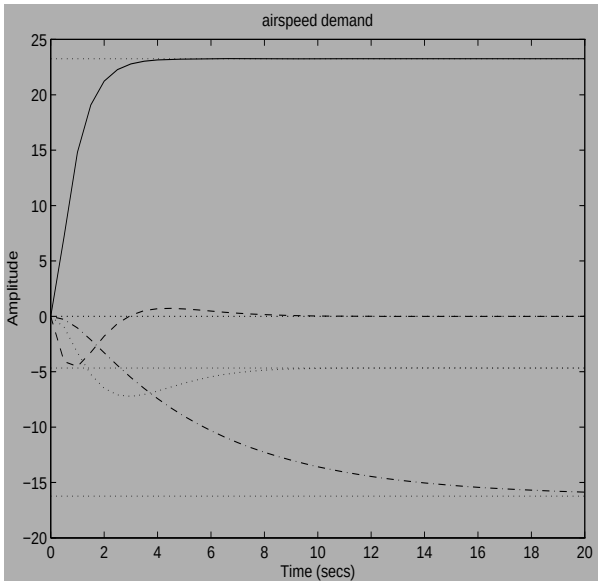
The condition number of the eigenvectors of the closed loop system is

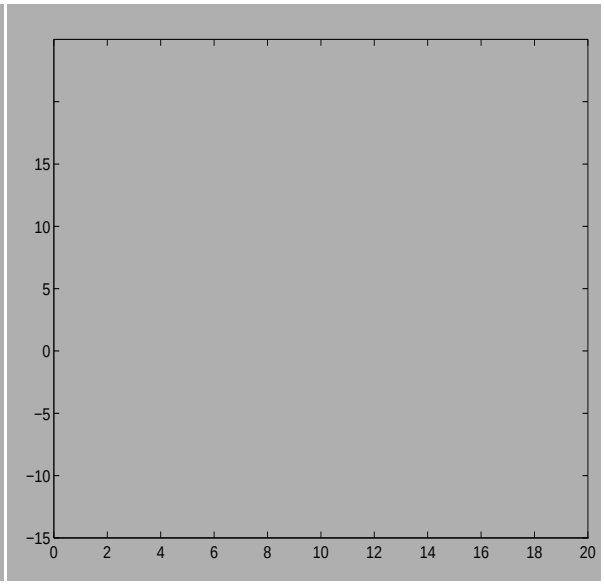
$$\kappa_F(V) = 1.11 \times 10^3, \tag{8.60}$$

which is quite large, as expected from the sensitivit


output response to (0)=0.05 °
pitch attitude
airspeed
flight path angle
pitch rate
Amplitude
Time (secs)


pitch attitude demand
Amplitude
Time (secs)


airspeed demand
Amplitude
Time (secs)



8.5.1 pply restricted minimisation algorithm (for decoupling)

Our primary aim is to reduce the level of input decoupling apparent in the left vectors. Thus, our choice of parameter weightings is

$$(\omega_1^2, \omega_2^2, \omega_3^2) = (1, 0, 0), \tag{8.64}$$

so that we are allowing the eigenvalues to vary. Note that we have also placed a zero weighting on the conditioning bound. The results of the minimisation algorithm are

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$	$\sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2$
0	$5.6620e+03$	$5.6620e+03$	$1.1141e+03$	$2.4164e-26$
1	$8.3169e+02$	$8.3169e+02$	$9.0047e+03$	$1.1417e+05$
2	$6.4465e+02$	$6.4465e+02$	$8.8878e+03$	$1.4010e+05$
3	$5.6885e+02$	$5.6885e+02$	$9.6162e+03$	$1.5408e+05$
4	$5.2047e+02$	$5.2047e+02$		

The closed loop eigenvalues of $A + BKC$ are

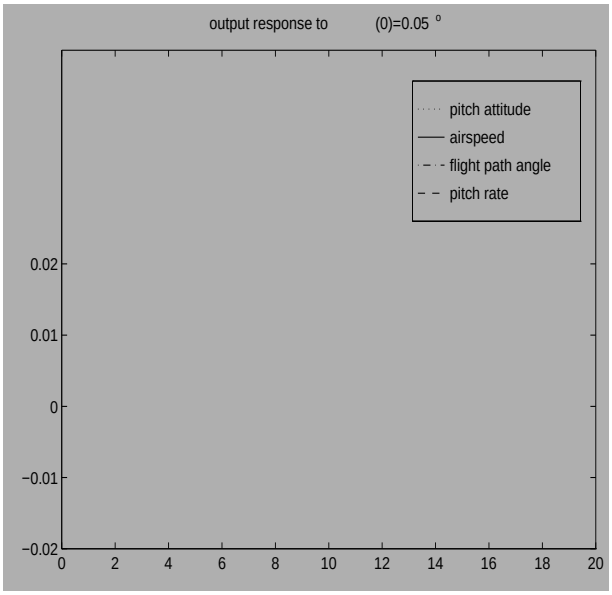
closed-loop eigenvalue	frequency	damping	sensitivity
$-0.5213 \pm 0.1073i$	0.5323	0.9795	236.22
-3.8306	3.8306	1	208.28
-0.2027	0.2027	1	25.17
-13.2812	13.2812	1	6.33
-4.9036	4.9036	1	61.04
-1.8594	1.8594	1	153.92
-19.1100	19.1100	1	1.49
$-5.7925 + 7.0577i$	9.1304	0.6344	22.16

From these results we see that, although the left eigenspace error was weighted zero, the eigenvalues have not moved by much. However, we have increased the individual sensitivities of the eigenvalues. The condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 1.3567 \times 10^3. \tag{8.66}$$

This is not surprising, since the conditioning was not included in the minimi

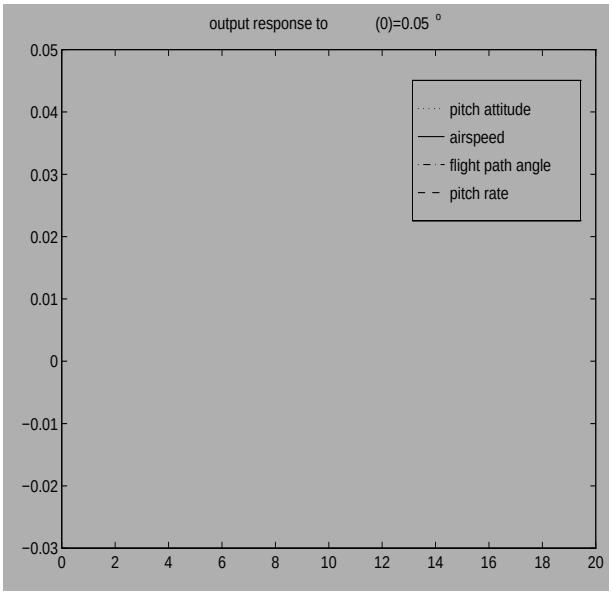
the output and input coupling vectors independently, but must look how each input effects each output. The new minimised output and input responses are presented in Figure 8.18. From these diagrams we can see the input and output



From these results we can clearly see that we have reduced the individual eigenvalue sensitivities by a substantial amount. This is reflected in the condition number of the eigenvectors of the closed loop system,

$$\kappa_F(V) = 296.5605. \tag{8.71}$$

We also notice that we now have an extra complex mode, one of the pair being one of the assigned modes. Since the left vector matching was not weighted, the mode input coupling vectors have not improved and so are not given here. The new minimised output and input responses are presented in Figure 8.19.


output response to (0)=0.05 °
0.05
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
0
2
4
6
8
10
12
14
16
18
20
pitch attitude
airspeed
flight path angle
pitch rate

8.5.3 pply unrestricted minimisation algorithm (for decoupling and conditioning)

We have successfully applied the restricted minimisation algorithm to reduce the level of input coupling and improve the robustness of the system. Here we use the unrestricted minimisation algorithm to obtain both of these system requirements simultaneously. We find an initial, real vector set by implementing Method 5 and choose the weightings

$$(\omega_1^2, \omega_2^2) = (1, 1). \tag{8.72}$$

We run the minimisation, using the alternative scaling method as in Section 5.10, giving the results

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$3.5862e+03$	$3.5623e+03$	$8.2354e+01$
1	$1.9689e+02$	$9.1189e-01$	$7.1862e+03$
2	$5.5452e+01$	$8.0264e-01$	$3.7800e+03$
3	$1.9101e+01$	$7.7040e-01$	$2.1943e+03$
4	$1.0522e+01$	$6.0644e-01$	$1.6197e+03$
5	$1.1898e+01$	$4.3585e-01$	$1.7450e+03$
6	$1.2383e+01$	$4.2241e-01$	$1.7825e+03$
7	$1.2642e+01$	$4.1915e-01$	$1.8021e+03$
8	$1.2844e+01$	$4.1811e-01$	$1.8171e+03$
9	$1.3030e+01$	$4.1766e-01$	$1.8308e+03$
10	$1.3208e+01$	$4.1739e-01$	$1.8437e+03$
11	$1.3381e+01$	$4.1720e-01$	$1.8563e+03$
12	$1.3551e+01$	$4.1705e-01$	$1.8684e+03$
13	$1.3716e+01$	$4.1691e-01$	$1.8802e+03$

We construct a new feedback using the diagonal solver, allowing one complex mode in both the assigned and unassigned modes,

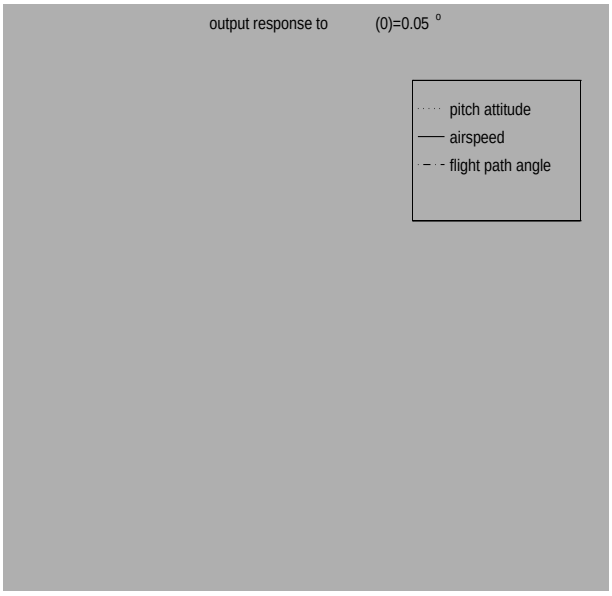
$$K_3 = \begin{bmatrix} -0.0148 & 0.0380 & 0.0470 & 0.1134 \\ -0.0315 & -0.0707 & -0.0059 & -0.0067 \\ -0.0866 & -0.0551 & 0.0486 & -0.0212 \end{bmatrix}. \tag{8.73}$$

The new closed loop eigenvalues are

closed-loop eigenvalue	frequency	damping	sensitivity
$-0.6373 \pm 0.3146i$	0.7107	0.8967	155.36
-3.7998	3.7998	1	92.91
-0.0461	0.0461	1	6.45
-19.1144	19.1144	1	1.3
$-7.8546 \pm 5.7133i$	9.7127	0.8087	16.49
-9.3156	9.3156	1	9.05
-1.7449	1.7449	1	54.36
-4.8105	4.8105	1	51.97

.

The individual eigenvalue sensitivities have reduced by a small amount; the con-


output response to (0)=0.05 °
pitch attitude
airspeed
flight path angle

8.5.4 Example 2 conclusions

For this example we have applied both the restricted and unrestricted minimisation algorithms. We have obtained results such that the level of input coupling has been reduced and the robustness of the system improved. We noted the need to consider the mode output and input coupling vectors simultaneously since the improvement in the input decoupling may be lost due to increased coupling in the outputs.

Even though we improved the level of input decoupling, we did not manage to obtain the specified level, but we did in Example 1. This is probably due to the fact that the system in Example 2 has three more state variables than in Example 1, but only one more control variable. Since the dimension of the subspace from which the new minimisation vectors are chosen is equal to the number of control variables, there is an even more limited choice for the vectors.

8.6 C nclusi ns

We have given new methods for performing eigenstructure assignment with specific consideration of the left eigenvectors to reduce the level of input coupling in aircraft problems. Previous work concentrated on assigning a set of right eigenvectors to control the output coupling; Smith [52] identified the need to consider the left eigenvectors in addition. However, no direct work was performed on the left eigenvectors beyond solving the whole problem using an optimisation package with a constraint on the left eigenvectors. We have extended this by producing two minimisation routines that balance the levels of input and output decoupling. The level of output decoupling remains constant throughout the algorithms, but its exact attainment is relaxed by the feedback construction. Also, we have included a measure on the robustness of the system, which is important in addition to the decoupling requirements.

We have illustrated our extensions to the work by applying our techniques to tw

of the output decoupling. For both examples we also reduced the conditioning of the systems, so increasing their robustness. Our results demonstrate that the trade-off between aircraft flight performance, stability and robustness can be achieved.

The minimisation criteria have individual weightings; we thus have a flexible design tool with parameters that can be altered in respect to the design specifications. We have demonstrated that, for our chosen formulation of the problem, our methods are numerically efficient in comparison to an optimisation pacyeaoT]esSctcao rofbeneofe

ofnesidinmT]taneousu

Chapter 9

Concl sions and extensions

In this thesis, we have addressed the problem of satisfactory flight control using eigenstructure assignment techniques. Specifically, we have illustrated shortcomings in previous work with respect to the consideration of both the left and right eigenvector sets corresponding to a set of desired eigenvalues. Generally, this problem is not exactly solvable; hence we presented two minimisation techniques to best meet the design specifications.

In Chapter 2 we introduced general control systems, their governing equations and their characteristics. We gave a general comment on feedback for the purpose of eigenstructure assignment and outlined our interest in aircraft problems. We then reviewed the literature on eigenstructure assignment and its application to aircraft problems.

In Chapter 3 we gave an outline of the derivation of the aircraft equations of motion. It was shown how a highly non-linear system could be reduced, via linearisations and certain flight state assumptions, into a state space matrix formulation.

In Chapter 4 we gave the background theory to current eigenstructure assignment techniques, including the theory for both full and partial system assignment.

In Chapter 5 we extended this work to control the left v

the minimisation routine. Since we were trying to assign all of the vectors, our aim was to perform full eigenstructure assignment. This has the advantage over partial eigenstructure assignment where there is no control over the unassigned modes, which can go unstable. Full eigenstructure assignment is not generally possible, but we derived errors for two feedback constructions for the restricted minimisation, proving that these errors were both related to the left eigenspace error. These constructions relied upon knowing a full set of eigenvalues, not the case for the unrestricted minimisation.

To find a feedbac

improve our results. If we wish the eigenvalues to move, then we have to put a low relativ

[19] M. M. Fahmy and H. S. Tantawy. Eigenstructure assignment via linear state-feedback control. *Int. Journal of Control*, 40(1):pp. 161–178, 1984.

[20] L. R. Fletcher. An intermediate algorithm for pole placement in linear multivariable control systems. *Int. Journal of Control*, 31(6):pp. 1121–1136, 1980.

[21] L. R. Fletcher, J. Kautsky, G. K. G. Kolka, and N. K. Nichols. Some necessary and sufficient conditions for eigenstructure assignment. *Int. Journal of Control*, 42(6):pp.1457–1468, 1985.

[22] S. Garg. Robust eigenspace assignment using singular value sensitivities. *Journal of Guidance, Control and Dynamics*, 14(12):pp. 416–424, 1991.

[23] E. G. Gilbert. Conditions for minimizing the norm sensitivity of characteristic roots. *IEEE Trans. Automatic Control*, A

[31] G. Klein and B. C. Moore. Eigenvalue-generalized eigenvector assignment with state feedback. *IEEE Trans. Automatic Control*, AC-22:pp. 140–141, Feb. 1977.

[32] J. F. Magni and A. Manouan. Robust flight control design by output feedback. Technical report, CERT-DERA, Nov. 1993.

[33] D. McLean. *Automatic Flight Control Systems*. Prentice Hall, 1990.

[34] R. H. Middleton. *elta Toolbox for use with MATLAB*

[53] K. M. Sobel, S. S. Banda, and H.-H. Yeh. Robust con

[63] S. Srinathkumar. Eigenvalue/eigenvector assignment using output feedback. *IEEE Trans. Automatic Control*, AC-23(1):pp. 79–81, 1978.

[64] T. Topalogu and D. E. Seborg. A design procedure for pole assignment using output feedback. *Int. Journal of Control*, 22(6):pp. 741–748, 1975.

[65] A. Varga. A Schur method for pole assignment. *IEEE Trans. Automatic Control*, AC-26(2):pp. 517–519, April 1981.

[66] B. A. White. Eigenstructure assignment by output feedback. *Int. J. Control*, 53(6):pp. 1413–1429, 1991.

[67] J. H. Wilkinson. *The algebraic eigenvalue problem.* Oxford University Press, 1965.

[68] R. F. Wilson and J. R. Cloutier. Optimal eigenstructure achievement with robustness guarantees. In *Proc. of American Control Conference*, volume 1, pages 952–957, 1990.

[69] R. F. Wilson and J. R. Cloutier. An alternate optimal formulation for robust eigenstructure assignment. In *NTC 91: National Telesystems Conference Proceedings*, volume 1, pages 1–11, 1991.

[70] R. F. Wilson and J. R. Cloutier. Control design for robust eigenstructure assignment in linear uncertain systems. In *Proc. of 30th Conf. on ecision and Control*, pages 2982–2987, 1991.

[71] R. F. Wilson, J. R. Cloutier, and R. K. Yedevalli. Lyapunov-constrained eigenstructure assignment for the design of robust mode-decoupled roll-yaw missile autopilots. In *First IEEE Conference on Control Applications*, volume 2, pages 994–999, 1992.

[72] W. Wonham. On pole assignment in multi-input, controllable linear systems. *IEEE Trans. Automatic Control*, AC-12(6):pp. 660–665, Dec. 1967.

[73] W. Yu, J. E. Piou, and K. M. Sobel. Robust eigenstructure assignment for the extended medium range air to air missile. In *Proc. of the 30th Conference on ecision and Control*, pages 2976–2981, 1991.

[74] W. Yu and K. M. Sobel. Robust eigenstructure assignment with structured state space uncertainty. *Journal of Guidance, Control and Dynamics*, 14:pp. 621–628, 1991.