U N I V E R S I T Y O F R E A D I N G

Numerical Techniques for Eigenstructure Assignment by Output Feedback 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 that best obtain the desired levels of decoupling. We also control the robustness of the system and the accuracy of the assigned eigenvalues. The result is a flexible, multi-criteria optimisation routine. The first routine restricts the minimisation vectors to lie in subspaces corresponding to specified eigenvalues, the second allows these vectors to be totally unrestricted.

The minimisation routines generate the set of vectors that best achieve desired levels of input and output decoupling; we give methods for constructing a feedback that best assigns these vectors and analyse the errors in these constructions. We give examples taken from the aircraft industry in which we achieve a trade-off between the levels of input and output decoupling, the robustness of the system and the accuracy of the assigned eigenvalues.

Acknowledgements

Firstly, I would like to thank Dr.N.K. Nichols for helping me through the last 3 years. Without her help and encouragement, I would undoubtedly never have finished.

The biggest thanks go to my parents; it is to them that I dedicate this thesis. Without their constant love and support (not just financial!), I would not have obtained my first degree, let alone my second. Thanks are also due to my sister for being a good friend.

I would also like to thank the people who helped me understand the theory of aircraft flight control. Firstly, Lester Faleiro and Roger Pratt at Loughborough University of Technology; the meetings and discussions I had with them were invaluable to this work. Thanks are also due to Dr. Phil Smith at R.A.E. Bedford for taking the time to suggest possible directions for this thesis.

I have made many good friends in Reading, both as an undergraduate and a postgraduate. I would like to thank them all for making my six years at Reading University such a good time.

I also acknowledge the receipt of a studentship from the EPSRC.

Contents

Chapter 1

Introduction

In most applied mathematical research the aim is to investigate and control a given system. A system is defined to mean a collection of objects which are related by interactions and produce various outputs in response to different inputs. Examples of such systems are chemical plants, aircraft, spacecraft, biological systems, or even the economic structure of a country or region. The control problems associated with these systems might be the production of some chemical product as efficiently as possible; automatic landing of aircraft, rendezvous with an artificial satellite; regulation of body functions such as heartbeat or blood pressure, and the ever-present problem of economic inflation.

To be able to control a system, we need a valid mathematical model. However, practical systems are inherently complicated and highly non-linear. Thus, simplifications are made, such as the linearisation of the system. Error analysis can then be employed to give information on how valid the linear mathematical model is as an approximation to the real system.

It is desirable that systems are controlled automatically, that is, they adapt to behave in a specified manner, without direct intervention. An example is the room thermostat in a domestic central heating system which turns the boiler on and off so as to maintain room temperature at a predetermined level.

To achieve automatic control, information describing the system and the way it changes is needed. This is provided by a feedback control system, which calculates the difference between the measured variables and the desired output responses, and attempts to change the system to compensate for this.

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 given. An example summarises the theory of Chapter 4 and demonstrates the shortfall in the current work in that the left eigenvectors should be considered in

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, allowing the updating vectors to be chosen from anywhere in the complex plane. We require the full set of vectors to be real, and various methods for choosing an initial real set of vectors are presented. Again the minimisation is run with different weightings to demonstrate its performance.

Chapters 5 and 6 result in a set of vectors that minimise some set criteria. We require a feedback that best assigns the vectors; this problem is treated in Chapter 7 where we give different methods for the feedback construction, analysing the errors for each of them.

Examples that demonstrate all of the theory are presented in Chapter 8, from the initial assignment of some set of eigenvalues and their corresponding right eigenvectors, to the use of one of the minimisation algorithms, and finally the construction of a feedback. These examples are taken from the aircraft industry, and are used to illustrate how we can improve flight handling qualities.

We summarise the results in Chapter 9, giving conclusions on how our work improves the methods presently available. We finish by suggesting some possible improvements and extensions to the work.

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

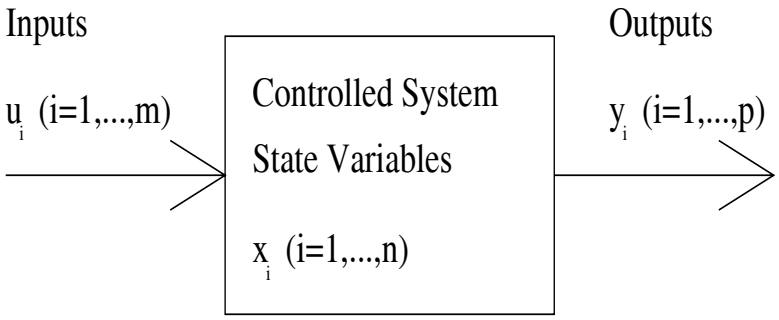

Figure 2.1: Open loop control system

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 Properties

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

2.2.1 Solution of state space equations

The solution of the system described by the state space equations in (2.1) is

$$\mathbf{x}(t) = e^{At}\mathbf{x}_0 + \int_0^t e^{A(t-s)}B\mathbf{u}(s)ds. \tag{2.4}$$

If we assume that A is non-defective, that is, if an eigenvalue has multiplicity k, then there exist k independent eigenvectors associated with it; then from Definition 2.1, we may write $A = V\Lambda V^{-1}$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$. Then

$$\begin{aligned} e^{At} &= I + At + \tfrac{A^2t^2}{2!} + \tfrac{A^3t^3}{3!} + \cdots \\ &= I + (V\Lambda V^{-1})t + \tfrac{(V\Lambda V^{-1})^2t^2}{2!} + \tfrac{(V\Lambda V^{-1})^3t^3}{3!} + \cdots \\ &= V[I + \Lambda t + \tfrac{\Lambda^2t^2}{2!} + \cdots]V^{-1} \\ &= Ve^{\Lambda t}V^{-1} \\ &= \sum_{i=1}^{n}\mathbf{v}_i e^{\lambda_i t}\mathbf{w}_i^T \end{aligned} \tag{2.5}$$

since $e^{\Lambda t} = \text{diag}(e^{\lambda_1 t}, \ldots, e^{\lambda_n t})$. Substitution into (2.4) gives the solution

$$\mathbf{x}(t) = \sum_{i=1}^{n}\mathbf{v}_i e^{\lambda_i t}\mathbf{w}_i^T\mathbf{x}_0 + \sum_{i=1}^{n}\mathbf{v}_i\mathbf{w}_i^T\int_0^t e^{\lambda_i(t-s)}B\mathbf{u}(s)ds, \tag{2.6}$$

and we can see that the response of the system depends on:

1. the eigenvalues, which determine the *decay/growth rate* of the response,
2. the eigenvectors, which determine the state variables participating in the response of each mode,
3. the initial condition of the system, which determines the *degree* to which each mode participates in the free response.

From (2.6) we see that the whole eigenstructure of the system (i.e. both the eigenvalues and the eigenvectors) should be considered when looking at the solution of control systems.

2.2.2 Stability

One of the main concerns of a control system designer is whether or not a system is stable. In its uncontrolled form a perturbed system may not return to its original operating condition; it is unstable. Intuitively, by stability we mean that

for small perturbations from the equilibrium state, the subsequent motions should not be too large. There are many concepts and definitions of stability, we choose to define the following:

Definition 2.2 *An equilibrium state* $\mathbf{x} = 0$ *is said to be*

$$\begin{aligned}
&(i) \quad \textbf{stable } \textit{if } \forall \epsilon > 0, \exists \delta > 0, \textit{such that } \|\mathbf{x}(t_0)\| < \delta \Rightarrow \|\mathbf{x}(t)\| < \epsilon \ (\forall t \geq t_0) \\
&(ii) \quad \textbf{asymptotically stable } \textit{if it is stable as in (i) and } \mathbf{x}(t) \to 0 \ (\textit{as } t \to \infty) \\
&(iii) \quad \textbf{unstable } \textit{if not stable as in (i), i.e. } \exists \epsilon > 0 \textit{ such that } \forall \delta > 0, \\
&\qquad \exists \mathbf{x}(t_0) \textit{ such that } \|\mathbf{x}(t_0)\| < \delta, \|\mathbf{x}(t_1)\| \geq \epsilon \quad (\textit{for some } t_1 > t_0)
\end{aligned} \tag{2.7}$$

This definition is not easy to relate to our control system given in (2.1); instead we can give a result for the algebraic stability of a linear system.

THEOREM 2.3 *Let the eigenvalues of* A *be* λ_i $(i = 1, \ldots, n)$*, then the time-invariant, linear system given in (2.1) is*

$$\begin{aligned}
&(i) \quad \textbf{stable} \Leftrightarrow Re(\lambda_i) \leq 0 \ (\forall \lambda_i) \\
&\qquad \textit{and any eigenvalue with } Re(\lambda_i) = 0 \textit{ is non-defective} \\
&(ii) \quad \textbf{asymptotically stable} \Leftrightarrow Re(\lambda_i) < 0 \ (\forall \lambda_i) \\
&(iii) \quad \textbf{unstable} \Leftrightarrow Re(\lambda_i) > 0 \ (\textit{for some } \ \lambda_i).
\end{aligned} \tag{2.8}$$

Proof (see Barnett and Cameron [4]).

From this we can see that the stability of the control system depends on the positions of the eigenvalues of the system coefficient matrix, A, in the complex plane.

2.2.3 Controllability

If we wish to control the open loop system, we must determine whether a desired objective can be achieved by manipulating the chosen control variables. We can define the general property of being able to transfer a system from any given state to any other by means of a suitable choice of control functions.

The linear, time-varying system described by (2.1) with A, B, C all functions of time has the following definition of controllability.

Definition 2.4 *A system is said to be* **completely controllable** *if, for any t_0, any initial state $\mathbf{x}(t_0) = \mathbf{x}_0$ and any given final state $\mathbf{x}_f$, there exists a finite time $t_1 > t_0$ and a control $\mathbf{u}(t)$, $t_0 \leq t \leq t_1$, such that $\mathbf{x}(t_1) = \mathbf{x}_f$.*

As with the stability definitions, we can give a more specific algebraic criterion for controllability, this time for the linear, time-invariant system given in (2.1). Equivalent mathematical conditions for a system to be completely controllable are given in the following theorem.

THEOREM 2.5 *A system is said to be completely controllable if and only if one of the following equivalent conditions holds.*

$$
\begin{aligned}
&(i) \quad rank[B, AB, \ldots, A^{n-1}B] = n \\
&(ii) \quad rank[B, A - \lambda I] = n, \quad (\forall \lambda \in \mathbb{C}) \\
&(iii) \quad \{\mathbf{s}^T A = \mu \mathbf{s}^T \ \ and \ \ \mathbf{s}^T B = 0\} \iff \mathbf{s}^T = 0.
\end{aligned}
$$

Proof (see Barnett and Cameron [4]).

2.2.4 Observability

Closely linked to the controllability idea is the concept of observability; that is, the possibility of determining the state of a system by measuring only the outputs. For the system governed by the differential equations given in (2.1), where A, B, C are considered time-varying, we give the following definition of observability.

Definition 2.6 *A system is said to be* **completely observable** *if, for any t_0 and any initial state $\mathbf{x}(t_0) = \mathbf{x}_0$, there exists a finite time $t_1 > t_0$ such that knowledge of $\mathbf{u}(t)$ and $\mathbf{y}(t)$ for $t_0 \leq t \leq t_1$ suffices to determine $\mathbf{x}_0$ uniquely. There is no loss of generality in assuming $\mathbf{u}(t) \equiv 0$ throughout the interval.*

For the linear, time-invariant system in (2.1) we can give a more specific algebraic criterion for observability. Equivalent mathematical conditions for a system to be completely observable are given in the following theorem.

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

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

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

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

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 control the system to behave in some desired manner, or to display some desired characteristics. To do this we use feedback, as illustrated in Figure 2.2.

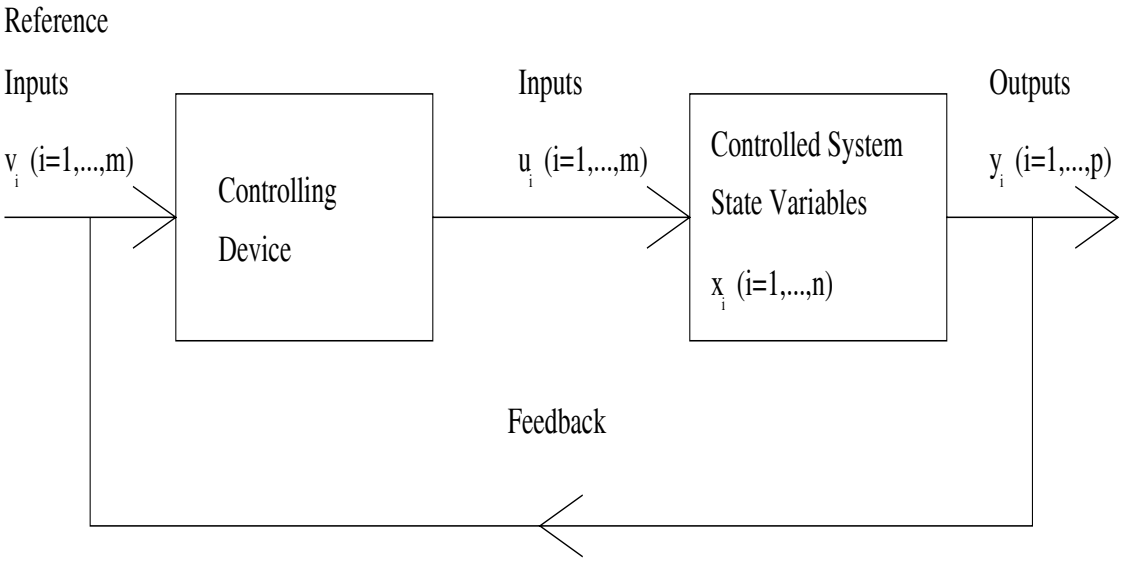

Figure 2.2: Closed loop control system

In general the objective is to make the system perform in some required way by suitably manipulating the control variables, u_i, this being done by a controlling device, or 'controller'. If all of the state variables of the system are available, then we may calculate a feedback matrix, F, such that the closed loop system, $A+BF$, has the desired characteristics. This is called state feedback.

In practice, it may be expensive to measure all of the state variables, or they may not all be available for measurement. We then feedback some of the outputs via a controller in the form

$$\mathbf{u} = K\mathbf{y} + \mathbf{v}, \tag{2.11}$$

where $K \in \mathbb{R}^{m\times p}$ is a constant gain feedback matrix, and $\mathbf{v} \in \mathbb{R}^m$ is a reference input. The closed-loop system is then given by

$$\dot{\mathbf{x}} = (A + BKC)\mathbf{x} + B\mathbf{v}. \tag{2.12}$$

This is called output feedback. We deduce that this change in the state matrix produces a change in the system behaviour, and therefore that the feedback matrix controls the way in which the system behaves.

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 how to construct the feedback, are given in Chapter 4.

2.2.7 Problems of interest

We have defined the general characteristics of control systems and how feedback may be used to alter the system. But the way in which the closed loop system behaves is dependent on the specific problem that we are solving.

In this thesis we are interested in the automatic flight control of aircraft. The equations of motion for an aircraft are time-varying and non-linear; to be able to use linear control design methods on these problems we need equations in the form of (2.1). To obtain this form, equations are linearised about a set of defined operating conditions. The resulting equations are still not in a usable form; numerous assumptions and substitutions are used to transform the equations into a linear, time invariant system. This derivation of the aircraft equations of motion and their transformation into a usable form is described in detail in Chapter 3.

There are a number of considerations to be taken into account when controlling an aircraft using feedback. We want to ensure that the closed loop system is stable and as robust as possible. We also wish to reduce the level of coupling evident between the inputs and the outputs. These can all be effected via the eigenvalues and eigenvectors of the system. Hence we use eigenstructure assignment techniques to obtain a satisfactory closed loop flight control system. The details of eigenstructure assignment are covered in Chapter 4.

Before giving the theory of the aircraft modelling and eigenstructure assignment in Chapters 3 and 4 respectively, we give a review of the literature on the two subjects, and their combination.

2.3 Literature review

There has been a lot of work performed in the control theory field in the past three decades into the control of systems via their eigenvalues and eigenvectors. More recently, these methods have been successfully applied to aircraft flight control systems. The progression of the work in eigenstructure assignment is reviewed, followed by its application to aircraft problems.

2.3.1 Eigenstructure assignment

As shown in the previous sections, the response of a control system depends most importantly on the eigenstructure of the system. The eigenstructure assignment problem in its simplest form was first addressed by Wonham [72] in 1967 who proved that a system was controllable if and only if a feedback could be found to make the closed loop system have an arbitrary set of self-conjugate scalars as its poles. Since then, hundreds of papers have been published on the subject of pole placement and its applications. For multi-input systems, the feedback gain matrix calculated to obtain a desired set of closed loop poles is not unique, an advantage that can be exploited.

In 1976, Moore [36] was the first to identify the freedom available in pole placement beyond eigenvalue assignment, but this was for the case of distinct eigenvalues. This restriction was overcome by Klein and Moore [31]. Numerous papers followed that used the freedom available in selecting the eigenvectors to perform full state feedback, as in Porter and D'Azzo [49] and Fahmy and O'Reilly [16]. As the subject became more applicable, so came the need for more reliable numerical methods. A popular method was to reduce the original system, using orthogonal similarity transformations, to staircase or upper Hessenberg form, as in Minimis and Paige [35], Patel and Misra [42], Arnold and Datta [3] and Petkov *et al.* [44]. In 1985, Kautsky *et al.* [28] described methods to select linearly independent vectors to ensure the matrix of eigenvectors was as well-conditioned as possible. Alternatives to the standard literature were Keel *et al.* [29] and Cavin and Bhattacharyya [9], who attempted to solve the problem via Sylvester's equation, and Varga [65], who used a Schur method to sequentially shift and

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 $(B) = m$ and rank $(C) = p$, then max (m, p) eigenvalues can be assigned almost arbitrarily. To complement these theoretical results, Davison and Chow [14] produced an algorithm to deal with large, practical systems. Later Kimura [30] proved that if the system is controllable and observable, and if $n \leq m + p - 1$, then an almost arbitrary set of n eigenvalues is assignable. However, this is not usually true for practical applications, but is worthy of note because of the author's consideration of the closed loop eigenvectors, rather than the characteristic equation approach of previous authors. The same results as Kimura [30] were produced by Davison and Wang [15].

In contrast to the previously mentioned approaches, Munro and Vardulakis [38] and Porter [46] investigated the existence of a link between the state and output feedback matrices. The approach of the former authors involved the computation of generalised inverses, whereas the latter produced a simpler condition for the link to exist. Other papers to note are those of Patel and Misra [42], who extended their state feedback work whereby they consider one column of the input matrix at a time, and Topalogu and Seborg [64], who assign min $(m+p-1, n)$ poles subject to certain mild restrictions, and who also introduce the elegant idea of pole protection by making them uncontrollable.

The previous references for output feedback are concerned with pole placement. On the broader subject of the whole eigenstructure of the system, Srinathkumar [63] is considered a benchmark paper. He proved the following :

THEOREM 2.8 *Given the controllable and observable system described by (2.1)*

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

this end was made by Oh *et al.* [40] who generate a nonlinear programming problem with a nonlinear objective function (maximising robustness) and linear and nonlinear constraints. This approach would benefit from specifying regions for each pole, rather than a global interval, but it is not clear how this would be achievable. Further work by the same authors is presented in Gu *et al.* [26] using homotopy methods.

The survey so far illustrates the importance in considering the whole eigenstructure of a system, for both state and output feedback.

2.3.2 Application to aircraft problems

All of these papers are concerned, however, with the mathematical nature of the problem, usually with a simple example at the end. Here we are more concerned with the application of eigenstructure assignment by output feedback to the aircraft industry. In this instance, it is not enough to apply previously formulated methods to an aircraft problem, but it is desirable to consider the specific control objectives and modify the theory accordingly.

The earliest comprehensive study of applying eigenstructure assignment to aircraft examples is that of Andry *et al.* [1] who show that, in practice, it is unnecessary to specify all of an eigenvector corresponding to a desired eigenvalue. Using Srinathkumar [63], they specify certain components in the desired eigenvectors to achieve design specifications, such as modal decoupling, and these vectors are projected into the subspace of allowable eigenvectors to find the best achievable vectors in a least-squares sense. However, instead of attempting to assign the whole eigenstructure, they use the theory of Davison [11] to assign rank $C = p$ eigenvalues. They also consider constraining the feedback gain computation by suppressing certain elements to be zero, thus reflecting a method of not feeding back certain outputs to certain inputs. The work of Sobel and Shapiro [55], [56] is presented as a tutorial into the theory of eigenstructure assignment and its application in the aircraft industry. It covers the same material as Andry *et al.* [1], except that a more comprehensive example is considered, as is the effect of feedforward for the purpose of command tracking (see Davison [12] and O'Brien and Broussard [39]). In Sobel and Shapiro [54] eigenvalue sensitivity, as

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, Sobel and Shapiro [57] extended the latter theory to include dynamic compensators in their design techniques.

Robust methods for eigenstructure assignment for aircraft design were presented by Mudge and Patton [37] and Spurgeon and Patton [61] where the choice of eigenvectors was based on one of the robustness methods of Kautsky *et al.* [28]. Although the work in these two papers is for state feedback, they are of note for their calculation of the achievable spaces using the singular value decomposition, and of their treatment of complex eigenvalues into their real and imaginary parts for this application. Similar work is presented by Burrows and Patton and Burrows *et al.* [8], implemented in the Ctrl-C design package.

As an alternative approach, a method based on the work of Fahmy and O'Reilly [18] is presented by White [66] who uses a metric technique to find the best solution from a number of solutions calculated from a space restricted by eigenvalue specifications.

These aircraft papers are restrictive in that once the closed loop eigenvalues are specified, the allowable subspaces are fixed. The freedom is thus in choosing the 'best set' of eigenvectors from these spaces. To relax this, it has been popular since the late 1980's to allow the eigenvalues to vary, and consider multi-criteria optimisation with a trade-off between robustness and performance. There are many reliable, numerical software packages available for optimisations; the key to using them effectively is in the formulation of the problem objectives.

However, most of this work has been performed using state feedback, not the main interest here, but, for completeness, a review of them is included. Burrows and Patton [7] evaluate a cost function using a quasi-Newton search with numerically evaluated gradients to find the optimal low sensitivity modalised observer, where the eigenvalues are constrained to be in rectangular regions of the complex plane. The same authors [6] also use the parametric representation of state feedback of Fahmy and O'Reilly [16] to assign eigenvalues in a region, considering low eigenvalue sensitivity and a structurally constrained low norm gain.

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 consideration of the singular values. Structured stability robustness is considered by Apkarian [2] and uses a hybrid design of a nonlinear programming technique for robustness, and point-wise modal synthesis for performance. Garg [22] uses the sensitivity of the minimum singular value of the return difference matrix, at plant input, to changes in desired closed loop eigenvalues and specified elements of the desired closed loop eigenvectors. The algorithm uses gradient information to improve the gain and phase margins.

Optimisation techniques have been used for eigenstructure assignment via output feedback as demonstrated in various papers by Sobel *et al.* In 1987 Sobel and Shapiro [58] formulated an objective function to minimise the sum of the squares of the eigenvalue condition numbers subject to exact eigenvalue assignment. This was solved by selecting 100 points in a user specified region, this region being the space of vectors that parameterise the eigenvectors. Four it-

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 achieve some desired level of output and input decoupling respectively.

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 by state feedback, to that of eigenstructure assignment by output feedback. It has also been demonstrated that eigenstructure assignment is a useful tool for aircraft flight control system design, although this is not widely accepted in industry. The importance of considering the robustness of the closed loop system has also been recognised in the design of automatic flight control systems. The aircraft industry has specific design objectives that can be considered before attempting to design a control, rather than just applying an analytical technique. Over the last few years, many authors have realised the need for multi-objective designs and have turned to numerical optimisation to solve this. However, much of the work has considered only state feedback, and those considering output feedback do not consider the whole eigenstructure in the sense that the right eigenvectors only are used.

The aim of the work here is to apply eigenstructure assignment techniques via output feedback to aircraft control problems. We pay particular attention to the simultaneous assignment of both the right and left eigenvectors to obtain decoupling in the outputs and inputs of the closed loop system. It will be shown that this assignment is not, in general, possible; a multi-criteria minimisation approach is developed that also emphasises the importance of ensuring the robustness of the closed loop system.

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].

3.1 Introduction

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 of freedom; three associated with angular motion about the aircraft's centre of gravity and three associated with the translation of the centre of gravity. This extra freedom means that aircraft control problems are generally more complicated than those of other vehicles.

An aircraft's stability characterises how it resists changes of its velocity vector, either in direction or magnitude, or both. The aircraft's quality of control relates to the ease with which the velocity vector can be changed. Without control,

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 flight 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 combat aircraft, have considerably more control surfaces. The required motion in flight control often needs a number of control surfaces to be used simultaneously, this often leading to considerable coupling and interaction between motion variables.

3.1.2 Flight control systems

The primary flying controls are defined as the input elements moved directly by a human pilot to cause the operation of a control surface. The main primary flying controls are pitch, roll and yaw control, effected by the elevators, ailerons and rudder, respectively.

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 accordance with the control law, the command signals to the actuator to produce the control surface deflections which will result in the correct control force or moment being applied. The aircraft thus responds so that the measured and commanded motion are in correspondence. How the required control law is determined is the primary topic of this thesis.

3.1.3 Gain scheduling

This thesis uses control methods for linear, time-invariant systems. However, an aircraft in flight is highly non-linear and is certainly dependent on time. Thus, the whole flight envelope of the aircraft is divided into a series of discrete points around which the system is linearised. Since many flight control problems are of very short duration (5-20 seconds), the coefficients of the equations of motion can be regarded as constant. So for every discrete point we have a constant, linear system for which an AFCS is designed.

3.2 Aircraft equations of motion

Problems involving AFCS are relatively short in time; the dynamic situations rarely last more than a few minutes. Consequently the inertial frame of reference used is one that has its origin fixed at the centre of the Earth, typically with X_E pointing north, Y_E pointing east and Z_E pointing downwards. Thus, an aircraft being considered relative to (X_E, Y_E, Z_E) must have its own axis system, usually taken at the centre of gravity with X_B pointing forwards through the nose, Y_B pointing out through the starboard (right) wing and Z_B pointing downwards. This is known as a body-fixed axis system. Other common axis systems are the

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 illustrate the various components of the equations of motion next described. In Figure 3.1, (U, V, W) are the roll, pitch and yaw moments; (P, Q, R) are the angular velocities (roll, pitch and yaw); (Φ, Θ, Ψ) are the roll, pitch and yaw angles.

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, Z_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

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

which are thus the equations of translational motion.

Rotational motion

For a rigid body, angular momentum may be defined as

$$\mathbf{H} = I\boldsymbol{\omega}, \tag{3.6}$$

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 XZ plane (implying $I_{xy} = I_{yz} = 0$), results in

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

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 Z_E axis, Θ is the angle between the gravity vector and the $Y_B Z_B$ plane and Φ is the bank angle between the Z_B axis and the projection of the gravity vector on the $Y_B Z_B$ plane. Direct resolution of $m\mathbf{g}$ into its X, Y, Z components produces

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

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 $\dot{\Theta}$, but its projection in the $Y_B Z_B$ plane is normal to both. By resolution

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

where Φ, Θ, 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

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

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

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

Note that (3.11) must also be used since those equations relate Ψ, Θ, 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

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

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 $X_0, Y_0, Z_0, L_0, M_0, N_0$ 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 $\dot{\Psi} = \Theta_0 = 0$,
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

$$\begin{aligned} x &= m[\dot{u} + W_0 q + (\mathbf{g}\ \cos\Theta_0)\theta] \\ z &= m[\dot{w} - U_0 q + (\mathbf{g}\ \sin\Theta_0)\theta] \\ m_1 &= I_{yy}\dot{q} \end{aligned} \tag{3.15}$$

and

$$\begin{aligned} y &= m[\dot{v} + U_0 r - W_0 p - (\mathbf{g}\ \cos\Theta_0)\phi] \\ l &= I_{xx}\dot{p} - I_{xz}\dot{r} \\ n &= I_{zz}\dot{r} - I_{xz}\dot{p}. \end{aligned} \tag{3.16}$$

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

$$\begin{aligned} x = \tfrac{\partial X}{\partial u}u + \tfrac{\partial X}{\partial \dot{u}}\dot{u} + \tfrac{\partial X}{\partial w}w + \tfrac{\partial X}{\partial \dot{w}}\dot{w} + \tfrac{\partial X}{\partial q}q + \tfrac{\partial X}{\partial \dot{q}}\dot{q} + \tfrac{\partial X}{\partial \delta_E}\delta_E + \tfrac{\partial X}{\partial \dot{\delta}_E}\dot{\delta}_E \\ = m[\dot{u} + W_0 q + (\mathbf{g}\ \cos\Theta_0)\theta] \end{aligned} \tag{3.17}$$

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

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

and M_x, Z_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

$$\begin{aligned} \dot{u} &= X_u u + X_w w - W_0 q - (\mathbf{g}\ \cos\Theta_0)\theta + X_{\delta_E}\delta_E \\ \dot{w} &= Z_u u + Z_w w + U_0 q - (\mathbf{g}\ \sin\Theta_0)\theta + Z_{\delta_E}\delta_E \\ \dot{q} &= M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta_E}\delta_E \\ \dot{\theta} &= q, \end{aligned} \tag{3.20}$$

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

Equations of lateral motion

As in Section (3.2.2), we expand the left-hand sides of (3.16) using Taylor series, make the stability derivative substitutions

$$\begin{aligned} Y_j &\equiv \tfrac{1}{m}\tfrac{\partial Y}{\partial j} \\ L_j &\equiv \tfrac{1}{I_{xx}}\tfrac{\partial L}{\partial j} \\ N_j &\equiv \tfrac{1}{I_{zz}}\tfrac{\partial N}{\partial j} \end{aligned} \tag{3.21}$$

and neglect certain terms, as before. This gives the equations governing perturbed lateral/directional motion of the aircraft as

$$\begin{aligned} \dot{v} &= Y_v v - U_0 r + W_0 p + (\mathbf{g}\ \cos\Theta_0)\phi + Y_{\delta_R}\delta_R \\ \dot{p} &= \tfrac{I_{xz}}{I_{xx}}\dot{r} + L_v v + L_p p + L_r r + L_{\delta_A}\delta_A + L_{\delta_R}\delta_R \\ \dot{r} &= \tfrac{I_{xz}}{I_{zz}}\dot{p} + N_v v + N_p p + N_r r + N_{\delta_A}\delta_A + N_{\delta_R}\delta_R \\ \dot{p} &= \dot{\phi} - \dot{\Psi}\ \sin\Theta_0 \\ \dot{r} &= \dot{\Psi}\ \cos\Theta_0, \end{aligned} \tag{3.22}$$

where again the last two equations are usually added for completeness. Note also the subscripts A and R on the δ indicate aileron and rudder respectively.

3.2.3 Equations of motion in a stability axis system

In a stability axis system, the orientation is such that W_0 is zero. Initially, the stability axis system is inclined to the horizon at some flight path angle, γ_0, since $\Theta_0 \equiv \gamma_0 + \alpha_0$, and α_0 is zero because the axis is orientated so that the X_B axis points into the relative wind. So the equations in (3.20) become

$$\begin{aligned} \dot{u} &= X_u u + X_w w - (\mathbf{g}\ \cos\gamma_0)\theta + X_{\delta_E}\delta_E \\ \dot{w} &= Z_u u + Z_w w + U_0 q - (\mathbf{g}\ \sin\gamma_0)\theta + Z_{\delta_E}\delta_E \\ \dot{q} &= M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta_E}\delta_E \\ \dot{\theta} &= q. \end{aligned} \tag{3.23}$$

The corresponding version of (3.22) contains cross-product inertia terms, which are eliminated by the use of primed stability derivatives. Ignoring second order

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

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

3.3 State space representation

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 Aircraft 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

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -\mathbf{g}cos\gamma_0 \\ Z_u & Z_w & U_0 & -\mathbf{g}sin\gamma_0 \\ \tilde{M}_u & \tilde{M}_w & \tilde{M}_q & \tilde{M}_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_E} & X_{\delta_{th}} \\ Z_{\delta_E} & Z_{\delta_{th}} \\ \tilde{M}_{\delta_E} & \tilde{M}_{\delta_{th}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_E \\ \delta_{th} \end{bmatrix}. \quad (3.27)$$

Here U_0 is the equilibrium (or trim) forward velocity, g is gravity, γ_0 is the flight path angle and all other undefined components are stability derivatives. The significance of the tilde in the third row of (3.27) is explained by the fact that in (3.23)

$$\dot{q} = M_u u + M_w w + M_{\dot{w}} \dot{w} + M_q q + M_{\delta_E} \delta_E. \quad (3.28)$$

However, it is obvious that although a term in $\dot{w}$ exists on the right hand side of (3.28), it does not appear in the state equation on the r.h.s of (3.27). Fortunately, $\dot{w}$ itself depends only on $\mathbf{x}$ and $\mathbf{u}$ and so we can substitute the equation for $\dot{w}$ in (3.23) into (3.28), yielding

$$\begin{aligned} \dot{q} = (M_u + M_{\dot{w}} Z_u) u + (M_w + M_{\dot{w}} Z_w) w + (M_q + M_{\dot{w}} U_0) q \\ -(\mathbf{g} M_{\dot{w}} \sin\gamma_0)\theta + (M_{\delta_E} + M_{\dot{w}} Z_{\delta_E})\delta_E \end{aligned}. \quad (3.29)$$

Then

$$\dot{q} = \tilde{M}_u u + \tilde{M}_w w + \tilde{M}_q q + \tilde{M}_\theta \theta + \tilde{M}_{\delta_E} \delta_E, \quad (3.30)$$

where

$$\begin{aligned} \tilde{M}_u &= M_u + M_{\dot{w}} Z_u \\ \tilde{M}_w &= M_w + M_{\dot{w}} Z_w \\ \tilde{M}_q &= M_q + U_0 M_{\dot{w}} \\ \tilde{M}_\theta &= -\mathbf{g} M_w \sin\gamma_0 \\ \tilde{M}_{\delta_E} &= M_{\delta_E} + M_{\dot{w}} Z_{\delta_E}. \end{aligned} \quad (3.31)$$

This is shown for the purpose of an illustrative example; other states can be considered, as can other control inputs, depending on the specific AFCS problem.

3.3.2 Aircraft equations of lateral motion

If the state vector is

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

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

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_v & 0 & -1 & \frac{g}{U_0}\cos\gamma_0 & 0 \\ L'_\beta & L'_p & L'_r & 0 & 0 \\ N'_\beta & N'_p & N'_r & 0 & 0 \\ 0 & 1 & \tan\gamma_0 & 0 & 0 \\ 0 & 0 & \sec\gamma_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} 0 & Y^*_{\delta_R} \\ L'_{\delta_A} & L'_{\delta_R} \\ N'_{\delta_A} & N'_{\delta_R} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_R \end{bmatrix}, \tag{3.33}$$

where again all terms not mentioned in Section (3.3.1) are stability derivatives. Again this is not a unique representation and is just for the purposes of illustration. For all of the examples used in this thesis, we will describe each of the variables used.

3.4 Aircraft stability

In Section 2.2.2 we introduced the general concept of stability with respect to control systems. In aircraft control we are particularly interested in whether the system remains stable under parameter variations. For the treatment of aircraft stability, it is assumed here that the aircraft is fixed wing and flying straight and level in a trimmed condition, and that its motion is properly characterised by (3.25).

3.4.1 Longitudinal stability

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

$$\lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0. \tag{3.34}$$

An aircraft may be said to be dynamically stable if all of its eigenvalues have a real part that is negative. Positive real parts mean the aircraft is dynamically unstable. So also does a zero real part, since this corresponds to a mode having simple harmonic motion which, for practical flight situations, is considered to be unstable.

For most aircraft types, the quartic factorises into two quadratics,

$$(\lambda^2 + 2\xi_{ph}\omega_{ph} + \omega_{ph}^2)(\lambda^2 + 2\xi_{sp}\omega_{sp} + \omega_{sp}^2) = 0. \tag{3.35}$$

The first factor corresponds to a mode of motion which is characterised by an oscillation of long period. The damping of this mode is usually very low, and is sometimes negative, so that the mode is unstable and the oscillation grows with time. The low frequency associated with the long period motion is defined as the natural frequency, ω_{ph}; the damping ratio being denoted as ξ_{ph}. This mode is defined as the *phugoid mode*.

The second factor corresponds to a rapid, relatively well-damped motion associated with the *short period mode* whose frequency is ω_{sp} and damping ratio is ξ_{sp}.

For the frequency and damping of a mode, if the eigenvalue of a complex mode is expressed in the form

$$\lambda_{1,2} = x \pm \mathbf{i}y, \tag{3.36}$$

where $\mathbf{i} = (-1)^{\frac{1}{2}}$, then

$$\begin{aligned} \xi &= \frac{x}{(x^2+y^2)^{\frac{1}{2}}} \\ \omega &= (x^2 + y^2)^{\frac{1}{2}} \end{aligned} . \tag{3.37}$$

3.4.2 Lateral stability

The characteristic polynomial of lateral motion is a quintic of the form

$$\lambda^5 + d_1\lambda^4 + d_2\lambda^3 + d_3\lambda^2 + d_4\lambda = 0, \tag{3.38}$$

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 natural tendency for the aircraft to be restored to its equilibrium heading. An aircraft has neutral heading stability and it remains at its perturbed heading until some corrective control action is taken. The term (λ+e) corresponds to the spiral convergence/divergence mode, which is usually a very slow motion corresponding to a long term tendency either to maintain the wings level or to 'roll off' in a divergent spiral. The term (λ+f) corresponds to the rolling subsidence mode; the quadratic term represents the 'dutch roll' motion for which the values of the damping ratio, ξ_D, is usually small, so that 'dutch' rolling motion is oscillatory.

3.5 Summary

Here we have given an introduction to the theory of aircraft flight control systems. The uses of feedback in AFCS theory has been given as have the physical controls used to control the aircraft. It has been shown that the equations of motion are very complicated and highly coupled, but that under certain assumptions and flight conditions these can be simplified into a usable form. We have demonstrated the stability considerations of an aircraft, for both the longitudinal and lateral motions. In Chapter 4 we shall explain the basic mathematical theory of eigenstructure assignment used to control the stability, robustness and transient response of the aircraft, and show our additional requirements of controlling the levels of input and output decoupling.

Chapter 4

Eigenstructure assignment

In Chapter 2 we introduced the concept of control systems and gave their properties. We then introduced feedback as a method of forcing the system to behave in a desired manner. We stated that we seek a feedback to obtain a certain closed loop eigenstructure. Here we give more detail on how this feedback may be calculated. We then give the design specifications for aircraft problems.

4.1 State feedback

To perform feedback we use the measured system variables as the new inputs to the closed loop system. If we assume that all of the state variables are available for feedback, then the controller takes the form

$$\mathbf{u} = F\mathbf{x} + \mathbf{v}. \tag{4.1}$$

Substituting this into (2.1) gives the closed loop system

$$\dot{\mathbf{x}} = (A + BF)\mathbf{x} + B\mathbf{v}. \tag{4.2}$$

The problem here is to calculate F such that $A + BF$ has the desired eigenvalues and eigenvectors. Early works concentrated solely on the eigenvalues, the problem being formally stated as:

Problem 1 Given the real pair (A, B) and a self-conjugate set of scalars $\{\lambda_1, \ldots, \lambda_n\}$, find a real matrix $F \in \mathbb{R}^{m \times n}$ such that the eigenvalues of $A + BF$ are λ_i $(i = 1, \ldots, n)$.

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 subject of eigenvalue assignment. For multi-input systems, however, the calculated feedback is not unique, a fact that was overlooked by most authors. As covered in the literature review in Section 2.3, Moore [36] was the first to identify the freedom offered by state feedback beyond specification of the closed loop eigenvalues. He demonstrated that the freedom available occurred in the choice of the eigenvector corresponding to each desired eigenvalue, and exploited this freedom to effect a desired distribution of modes among the output components. The idea of calculating a feedback to obtain both desired eigenvalues and eigenvectors of a system gave rise to the field of eigenstructure assignment.

In Section 2.2.2 we introduced the concept of stability and showed its dependency on the real parts of the system eigenvalues lying in the open left hand plane. This criterion can thus be solved by applying Theorem 4.1, since we can choose a set of stable eigenvalues and calculate a feedback matrix that assigns these as the closed loop eigenvalues.

However, we also illustrated the need to obtain a robust closed loop system in Section 2.2.5. To do this we wish to reduce the condition numbers of the individual eigenvalues, which are bounded above by the condition number of the modal matrix of the eigenvectors, as in Section 2.2.5. Thus, we wish to assign a stable set of eigenvalues and a corresponding set of well-conditioned eigenvectors. The robust eigenstructure assignment problem can be formulated as:

Problem 2 Given the real pair (A, B) and a self-conjugate set of scalars $\{\lambda_1, \ldots, \lambda_n\}$, find a real matrix $F \in \mathbb{R}^{m \times n}$ and a non-singular V satisfying

$$(A + BF)V = V\Lambda, \tag{4.3}$$

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

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

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

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

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

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

Corollary 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

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

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

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

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

$$BKC = V \Lambda V^{-1} - A \tag{4.13}$$

and pre- and post-multiplication by U^T and P, respectively, gives

$$\begin{bmatrix} Z_B \\ 0 \end{bmatrix} K [Z_C^T, 0] = \begin{bmatrix} U_0^T \\ U_1^T \end{bmatrix} (V \Lambda V^{-1} - A)[P_0, P_1], \tag{4.14}$$

from which (4.9), (4.10) and (4.12) immediately follow by comparison of components, since V, U and P are all invertible. □

Corollary 4.5 *The right and left eigenvectors,* $\mathbf{v}_i$ *and* $\mathbf{w}_i^T$*, of* $A + BKC$*, corresponding to the assigned eigenvalues* λ_i *must be such that*

$$\mathbf{v}_i \in \mathcal{S}_i \equiv \mathcal{N}[U_1^T (A - \lambda_i I)] \tag{4.15}$$

$$\mathbf{w}_i \in \mathcal{T}_i \equiv \mathcal{N}[P_1^T (A^T - \lambda_i I)] \tag{4.16}$$

where $\mathcal{N}[\cdot]$ *denotes null space.*

Proof Follows directly from Theorem 4.4.

Thus, for n eigenvalues to be assigned exactly by output feedback, their right and left eigenvectors must simultaneously lie in the spaces defined in (4.15),(4.16), respectively. However, it is obvious from the relationship $V^{-1} = W^T$ that if the right eigenvectors are chosen to maximise robustness (or to alter the transient response) and so that $\mathbf{v}_i \in \mathcal{S}_i$, then the left eigenvectors are immediately defined, without ensuring they are in the correct spaces. To this end Chu *et al.* [10] proposed an algorithm that minimises a weighted sum of the robustness and the distance of the left eigenvectors from $\mathcal{T}_i$.

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*

$$[U_1^T(A-\lambda_i I)]^T = [\hat{S}_i, S_i]\begin{bmatrix} R_{Ri} \\ 0 \end{bmatrix} \tag{4.17}$$

$$[(A-\lambda_i I)P_1] = [\hat{T}_i, T_i]\begin{bmatrix} R_{Li} \\ 0 \end{bmatrix}. \tag{4.18}$$

Then $S_i, \hat{S}_i$ are orthonormal bases for the null space $\mathcal{S}_i$ and its complement respectively; similarly $T_i, \hat{T}_i$ are orthonormal bases for the null space $\mathcal{T}_i$ and its complement respectively.

It can be shown 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 $\mathcal{T}_i$. The weighted sum minimisation from Chu *et al.* [10] is thus to minimise the functional

$$F = \|DV^{-1}\|_F^2 + \sum_{i=1}^{n}\omega_i^2\|\mathbf{w}_i^T\hat{T}_i\|_2^2, \tag{4.19}$$

where $D = \mathrm{diag}(d_1,\ldots,d_n)$ and the d_i, ω_i^2 are weightings to be chosen. The computed values of F are non-increasing and, provided F is small, the computed feedback assigns eigenvalues close to those specified, and such that their individual sensitivities are small. This can be expressed more specifically in a theorem, but first a result is needed.

LEMMA 4.7 *Given that $\mathbf{v}_i \in \mathcal{S}_i$, $(i = 1,\ldots,n)$, then K defined by (4.12) implies*

$$BKC = (V\Lambda W^T - A)(I - P_1P_1^T). \tag{4.20}$$

Proof From (4.11), $B = U_0Z_B$ and $C = Z_C^TP_0^T$; and also

$$I = UU^T = [U_0, U_1]\begin{bmatrix} U_0^T \\ U_1^T \end{bmatrix} = U_0U_0^T + U_1U_1^T, \tag{4.21}$$

$$I = PP^T = [P_0, P_1]\begin{bmatrix} P_0^T \\ P_1^T \end{bmatrix} = P_0P_0^T + P_1P_1^T. \tag{4.22}$$

Using K from (4.12) we obtain

$$BKC = U_0U_0^T(V\Lambda W^T - A)P_0P_0^T, \tag{4.23}$$

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

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

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

$$(A + BKC)V - V\Lambda = -(V\Lambda W^T - A)P_1 P_1^T V, \tag{4.27}$$

and (4.26) follows by writing

$$E = V(\Lambda W^T - W^T A)P_1 P_1^T.\Box \tag{4.28}$$

Corollary 4.9 *The error in eigenvalue assignment,* E, *given in Theorem (4.8) can be bounded such that*

$$\|E\|_F^2 \leq \sum_{i=1}^{n} r_i^2 \|\mathbf{w}_i^T \hat{T}_i\|_2^2, \tag{4.29}$$

with r_i *fixed constants.*

Proof Taking norms of E as defined in (4.28) gives

$$\begin{aligned}
\|E\|_F^2 &= \|V(\Lambda W^T - W^T A)P_1 P_1^T\|_F^2 \\
&\leq \|V\|_2^2 \ \|(\Lambda W^T - W^T A)P_1 P_1^T\|_F^2 \\
&\leq \|V\|_2^2 \ \sum_{i=1}^{n} \|\mathbf{w}_i^T(\lambda_i I - A)P_1 P_1^T\|_2^2 \\
&= \|V\|_2^2 \ \sum_{i=1}^{n} \|\mathbf{w}_i^T(\lambda_i I - A)P_1\|_2^2 \qquad (\text{since } P_1^T P_1 = I) \\
&= \|V\|_2^2 \ \sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i R_{Li}\|_2^2 \qquad (\text{from } (4.18)) \\
&\leq \|V\|_2^2 \ \sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2 \|R_{Li}\|_2^2
\end{aligned} \tag{4.30}$$

which gives the required result with $r_i^2 = \|V\|_2^2\|R_{Li}\|_2^2 \leq n\|R_{Li}\|_2^2$, from the assumption that $\|\mathbf{v}_i\|_2^2 = 1, \ (i = 1, \ldots, n)$. □

4.3 Partial eigenstructure assignment

We have shown that the exact re-assignment of all of the eigenvalues depends on the right and left eigenvectors simultaneously lying in the correct eigenspaces. However, this cannot be achieved because of the right and left eigenvector dependency (see (2.3)), and a minimisation technique is required to make the eigenvalues more accurate. Hence, a lot of work is being performed on the eigenvectors, thus losing us the freedom to choose them to shape the response of the system as in (2.6).

The alternative is to assign a certain set of the eigenvalues exactly according to dimensional restrictions proved by previous authors. Davison [11] showed that if the system is controllable and if rank$[C] = p$, then p eigenvalues of the closed loop system are arbitrarily close (but not necessarily equal) to the p desired values. Thus, it is possible to assign exactly p desired eigenvalues and their corresponding right eigenvectors. However, the $n - p$ unassigned closed loop eigenvalues and

corresponding eigenvectors are uncontrolled and may therefore force the system to display poor behaviour e.g. it may be unstable or very sensitive. We therefore note that while partial eigenstructure assignment is a method for obtaining an exact eigenstructure for a set of modes, we must be careful about what happens to the unassigned modes. We consider the partial eigenstructure problem

Problem 4 Given the real triple (A, B, C) and a self-conjugate set of scalars $\{\lambda_1, \ldots, \lambda_p\}$ and a corresponding self-conjugate set of n-dimensional vectors, $\{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$, find a real matrix $K \in \mathbb{R}^{m \times p}$ such that the eigenvalues of $A + BKC$ contain λ_i $(i = 1, \ldots, p)$ as a subset, with corresponding eigenvectors $\mathbf{v}_i$ $(i = 1, \ldots, p)$. i.e. that

$$(A + BKC)V = V\Lambda, \tag{4.31}$$

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

THEOREM 4.10 *Given Λ and V non-singular and assuming that CV is invertible, then there exists K, a solution satisfying (4.31) if and only if $U_1^T(AV - V\Lambda) = 0$ with K given explicitly by*

$$K = Z_B^{-1} U_0^T (V\Lambda - AV)(CV)^{-1}. \tag{4.32}$$

Proof From (4.31), K must satisfy

$$\begin{aligned} BKCV &= V\Lambda - AV \\ \Rightarrow \qquad BK &= (V\Lambda - AV)(CV)^{-1}. \end{aligned} \tag{4.33}$$

Pre-multiplication by U^T gives

$$\begin{bmatrix} Z_B \\ 0 \end{bmatrix} K = \begin{bmatrix} U_0^T \\ U_1^T \end{bmatrix} (V\Lambda - AV)(CV)^{-1}, \tag{4.34}$$

from which $U_1^T(AV - V\Lambda) = 0$ and (4.32) immediately follows by comparison of components. □

We have given justification for adjusting the eigenstructure of a system to obtain certain objectives, but for a general mathematical problem. Next we shall introduce the details of the aircraft problem, specifically how the eigenvectors are chosen from performance requirements.

4.3.1 Aircraft 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:*

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

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 shown in (4.15) that the corresponding eigenvector, $\mathbf{v}_i$, must lie in the null space of $[U_1^T(A - \lambda_i I)]$. From (4.17), S_i is an orthonormal basis for the null space of $\mathcal{S}_i$ and hence any vector in $\mathcal{S}_i$ can be expressed in the form

$$\mathbf{v}_{ai} = S_i\boldsymbol{\eta}_i \quad (\boldsymbol{\eta}_i \in \mathbb{R}^m), \tag{4.36}$$

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

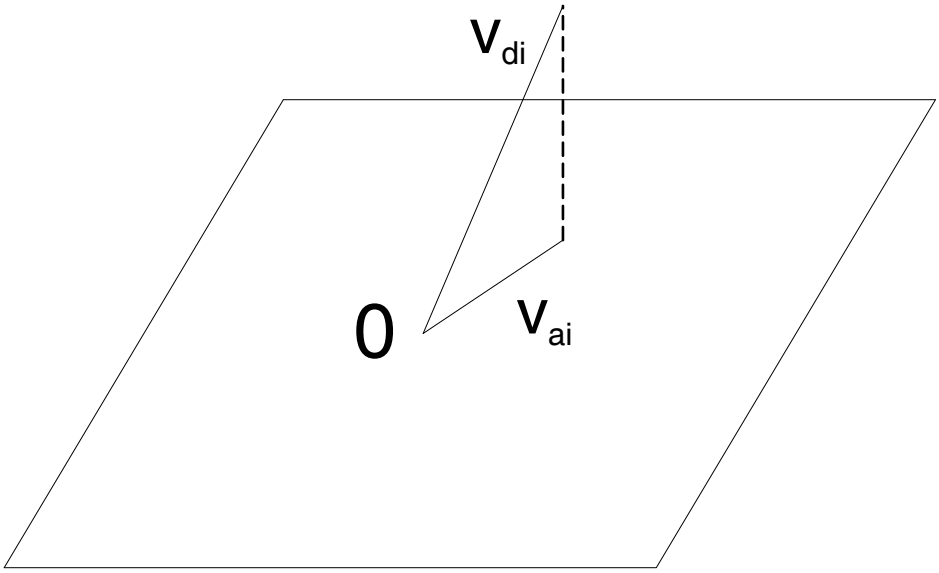

Figure 4.1: Projection to obtain 'best' achievable vector

To find the value of $\boldsymbol{\eta}_i$ corresponding to the projection of $\mathbf{v}_{di}$ onto the 'achievability subspace', $\boldsymbol{\eta}_i$ is chosen to solve the problem

$$\min J \equiv \min_{\boldsymbol{\eta}_i} \|\mathbf{v}_{di} - \mathbf{v}_{ai}\|^2 = \min_{\boldsymbol{\eta}_i} \|\mathbf{v}_{di} - S_i\boldsymbol{\eta}_i\|^2 \tag{4.37}$$

in a least squares sense. Differentiating J with respect to $\boldsymbol{\eta}_i$ gives

$$\frac{dJ}{d\boldsymbol{\eta}_i} = 2S_i^T(S_i\boldsymbol{\eta}_i - \mathbf{v}_{di}) \tag{4.38}$$

and J is at a minimum when $\frac{dJ}{d\boldsymbol{\eta}_i} = 0$, hence

$$\boldsymbol{\eta}_i = S_i^+\mathbf{v}_{di}. \tag{4.39}$$

The best achievable vector in a least squares sense that corresponds to a desired eigenvalue is thus

$$\mathbf{v}_{ai} = S_iS_i^+\mathbf{v}_{di}. \tag{4.40}$$

So, when performing partial eigenstructure assignment, the p desired eigenvalues are $\Lambda = \text{diag}\ \{\lambda_{d1}, \ldots, \lambda_{dp}\}$ and the corresponding desired eigenvectors are

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),

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

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 $\mathbf{d}_i$ and $\mathbf{u}_i$ are the vectors of specified and unspecified components respectively, and S_i has been reordered to conform with the reordered components of $\mathbf{v}_{di}$. This means that $\mathbf{d}_i, \mathbf{u}_i$ are composed of the v_{ij} and x's from (4.42), respectively.

Proceeding in precisely the same manner as the previous section (to find $\boldsymbol{\eta}_i$) with $\mathbf{d}_i$ replacing $\mathbf{v}_{di}$ and D_i replacing S_i, we obtain

$$\begin{aligned} \boldsymbol{\eta}_i &= D_i^+ \mathbf{d}_i \\ \mathbf{v}_{ai} &= S_i D_i^+ \mathbf{d}_i, \end{aligned} \tag{4.44}$$

so that the projection has been carried out on the specified components. Again the feedback matrix is constructed as in (4.32).

4.3.4 Example choice of desired eigenvectors

We have shown theoretically how to obtain the best set of achievable vectors, Here we give example desired vectors, chosen to obtain satisfactory aircraft performance (taken from Andry *et al.* [1]). For the linearised perturbed longitudinal equations of an aircraft the state vector may be

$$\mathbf{x} = \begin{bmatrix} \alpha \\ q \\ \theta \\ u \\ \delta_e \end{bmatrix} \begin{array}{l} \text{angle of attack} \\ \text{pitch rate} \\ \text{pitch angle} \\ \text{forward velocity(perturbed)} \\ \text{elevator deflection} \end{array} . \tag{4.45}$$

Note that this is different from the state vector given in Section 3.3.1. There it was explained that the state vector is not unique, and is dependent on the type of aircraft and on the design objectives. Here the angle of attack is one of the angles that orients the forces of lift and drag relative to the body fixed axes (the other being sideslip angle, β).

For this state vector, a good choice of closed loop eigenvectors might be

$$\begin{bmatrix} 1 \\ x \\ x \\ 0 \\ x \end{bmatrix} \quad \begin{bmatrix} x \\ 1 \\ x \\ 0 \\ x \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ x \\ 1 \\ x \\ x \end{bmatrix} \quad \begin{bmatrix} 0 \\ x \\ x \\ 1 \\ x \end{bmatrix} . \tag{4.46}$$

The first two vectors are called short period vectors and are chosen such that the variation in forward velocity is zero and the angle of attack and pitch rate are coupled together. The last two vectors are the phugoid vectors which couple pitch angle and (perturbed) forward velocity while holding angle of attack constant. This choice of eigenvectors coupled with the flight dynamics of the problem, i.e. during the short period mode the pitch angle is small, renders the subvector made up of the first four components of the short period vectors 'almost orthogonal' to the first four components of the phugoid vectors. This yields a good degree of decoupling.

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

$$\mathbf{y}(t) = \sum_{i=1}^{n} (C\mathbf{v}_i) e^{\lambda_i t} \mathbf{w}_i^T \mathbf{x}_0 + \sum_{i=1}^{n} (C\mathbf{v}_i)(\mathbf{w}_i^T B) \int_0^t e^{\lambda_i (t-s)} \mathbf{u}(s) ds, \tag{4.49}$$

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

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

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 $(CV)_{k,i}$ element. Here the 1^{st} input excites the 1^{st} mode (since $(W^T B)_{1,1} = 1$), which is directly coupled to the 1^{st} output ($(CV)_{1,1}$=1), but does not affect the 2^{nd} output ($(CV)_{2,1}$=1). However, the 1^{st} input excites the 4^{th} and 5^{th} modes ($(W^T B)_{4,1} = 1$, $(W^T B)_{5,1} = x$) which do affect the 2^{nd} output ($(CV)_{2,4} = x$, $(CV)_{2,5} = x$). These examples show the interactions between the modes and the inputs/outputs.

4.3.7 Partial eigenstructure assignment algorithm for aircraft problems

In this chapter we have described the theory of eigenstructure assignment by output feedback and have detailed the specifications of the aircraft industry. The algorithm used for achieving these objectives is:

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.

For this system, rank $(C) = 4$, so that, according to the theory of Davison [11], we can assign (almost exactly), four closed loop eigenvalues. These are

$$\Lambda_p = \begin{cases} -6 \pm i & \text{dutch roll mode} \\ -1 \pm 2i & \text{roll mode,} \end{cases} \tag{4.62}$$

with the corresponding desired mode output and mode input coupling vectors

$$G_{0d} = \begin{bmatrix} x & x & 0 & 0 \\ 0 & 0 & x & x \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, G_{1d} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \tag{4.63}$$

respectively. To find a feedback matrix that best assigns the closed loop eigenstructure, we use the algorithm in Section 4.3.7. This gives

$$K = \begin{bmatrix} 8.0313 & -0.2077 & -22.1264 & -0.5381 \\ 3.0432 & 0.9281 & -12.8538 & 4.0945 \end{bmatrix}, \tag{4.64}$$

which assigns the closed loop eigenvalues

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

.

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

$$\kappa_F(V) = 6.66 \times 10^4. \tag{4.65}$$

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

$$G_{0a} = \begin{bmatrix} 7.6425 \pm 1.4220i & 0.0057 \pm 0.0006i \\ 0 & -0.9998 \pm 1.9995i \\ 1 & -0.0067 \pm 0.0123i \\ 0 & 0.9998 \end{bmatrix}. \tag{4.66}$$

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 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

$$\begin{aligned} \|G_{0d} - G_{0a}\|_F^2 &= 4.5860 \times 10^{-4} \\ \|G_{1d} - G_{1a}\|_F^2 &= 23.0735 \end{aligned}, \tag{4.70}$$

respectively. The core of the new work that will follow is to attempt to achieve the desired eigenvalues and corresponding mode output and input coupling vectors simultaneously, in some kind of 'best fit' approximation.

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 of the left eigenvectors, in addition to the right eigenvectors, has often been overlooked. This was seen in the example whereby the right eigenvectors were matched exactly, but that there was an error in the matching of the left eigenvectors. This is our main concern in the thesis, considering the left eigenvectors. We have shown that simultaneous right/left eigenvector assignment is not, in general, possible. In the next three chapters we devise numerical techniques to obtain this eigenvector assignment in the best way in a defined sense.

Chapter 5

Restricted minimisation algorithm

As detailed in Chapter 4, there is a need to consider eigenstructure assignment when attempting to control stability, robustness or transient response, or a combination of the three. Having chosen the method of partial eigenstructure assignment (as in Section 4.3.7), we have shown how to construct an output feedback that achieves p desired eigenvalues and corresponding desired eigenvectors. These eigenvectors are the projections of the desired vectors into the subspaces of allowable vectors (as in Section 4.3.5). This is the method generally used by authors in the field, as outlined in Section 2.3. The recent move into optimisation package solvers also just considers p eigenvalues and their corresponding right eigenvectors.

However, in Section 4.3.5 it was shown that both the left and the right eigenvectors should be considered when altering a system. We also defined the mode output and input coupling vectors, G_0 and G_1, as reflecting the modal coupling interactions better than just the right and left eigenvectors, respectively. Some authors have assigned eigenvalues and their eigenvectors from the right, and then assigned a different set of eigenvalues and their eigenvectors from the left, but only Chu *et al.* [10], Magni and Mounan [32] and Smith [52] have considered the right and left eigenvectors corresponding to the same eigenvalue.

The problem is that from Corollary 4.5 (Chu *et al.* [10]), the right and left eigenvectors must simultaneously lie in certain subspaces corresponding to a de-

sired eigenvalue, and hence they cannot be considered independently. Thus we cannot expect to achieve the perfect desired decoupling in attaining both G_0 and G_1 simultaneously.

5.1 Right and left eigenvector partitioning

The objective here is to choose a set of eigenvectors to achieve desired decoupling through attaining G_0 and G_1. We assume that partial eigenstructure assignment by output feedback has been performed and this is used as the starting point for the minimisation algorithm.

The partial eigenstructure assignment method assigns a set of p eigenvalues and eigenvectors . The closed loop matrix is formed, the right and left eigenvectors are calculated, and are partitioned as follows

$$V = [V_1, V_2] \quad , \quad W^T = \begin{bmatrix} W_1^T \\ W_2^T \end{bmatrix}, \tag{5.1}$$

where $V_1 = [\mathbf{v}_1, \ldots, \mathbf{v}_p]$, $W_1^T = [\mathbf{w}_1, \ldots, \mathbf{w}_p]^T$. The mode output and input coupling vectors are defined to be

$$G_0 = CV = [CV_1, CV_2], \tag{5.2}$$

$$G_1 = W^T B = \begin{bmatrix} W_1^T B \\ W_2^T B \end{bmatrix}, \tag{5.3}$$

respectively. In using the partial eigenstructure assignment algorithm in Section 4.3 we have assigned exactly the p desired eigenvectors, V_1. Thus we have attained the desired p eigenvalues and their corresponding mode output coupling vectors, $G_{0a} = CV_1$. However, the mode input coupling vectors specified have not been used in constructing the feedback; indeed, they are only calculated from the closed loop system, and hence the achieved $G_{1a} = W_1^T B$ are not as desired. The minimisation algorithm developed here iterates and updates the unassigned vectors, V_2, to improve the matching of G_{1a}. This will then improve the input decoupling while retaining the output coupling obtained through the original eigenstructure assignment . The basis for this is that $V^{-1} = W^T$ so that altering any column of V changes all of the rows of W^T. The theoretical details on how V_2 is updated are described in the following sections.

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.

5.3 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 we note that every time a minimisation problem appears, the 'subject to $\|\mathbf{v}_i\|_2^2 = 1$ $(i = 1, \ldots, n)$' will be omitted for brevity, but it should be remembered that the condition still holds. However, since we are only concerned with matching the first p mode input coupling vectors, we have

$$\begin{aligned} G_{1a} \;=\; W_1^T B \;&=\; [I_p, 0] W^T B \\ &=\; [I_p, 0] V^{-1} B \end{aligned} \tag{5.8}$$

where $[I_p, 0]$ 'selects' the first p rows of W^T since I_p is the p-dimensional identity matrix. We therefore have the following problem to solve

$$\min J_1 \equiv \min_V \|G_{1d} - [I_p, 0] V^{-1} B\|_F^2. \tag{5.9}$$

It is here that we can exploit the specific structure of V^{-1} given in (5.5). First we note that

$$\mathbf{v}_n = S_n \boldsymbol{\eta}_n, \tag{5.10}$$

so that when we update $\mathbf{v}_n$, the new vector is restricted to lie in the eigenspace corresponding to the desired eigenvalue, λ_n. Now substituting (5.5) and (5.10) into (5.9) yields the problem

$$\begin{aligned} \min_{\boldsymbol{\eta}_n} J_1 \;&\equiv \min_{\boldsymbol{\eta}_n} \left\| G_{1d} - [I_p, 0] \begin{bmatrix} R^{-1} & -\rho R^{-1} Q_1^T S_n \boldsymbol{\eta}_n \\ \mathbf{0}^T & \rho \end{bmatrix} \begin{bmatrix} Q_1^T \\ \mathbf{q}^T \end{bmatrix} B \right\|_F^2 \\ &= \min_{\boldsymbol{\eta}_n} \left\| G_{1d} - [I_p, 0] \begin{bmatrix} D - \rho E \boldsymbol{\eta}_n \mathbf{z}^T \\ \rho \mathbf{z}^T \end{bmatrix} \right\|_F^2, \end{aligned} \tag{5.11}$$

where

$$\begin{cases} \mathbf{z}^T = \mathbf{q}^T B \\ D = R^{-1} Q_1^T B \\ E = R^{-1} Q_1^T S_n. \end{cases} \tag{5.12}$$

This can be simplified further by noting that $D \in \mathbb{C}^{n-1 \times m}$ so that, provided $(n-1) \geq p$, $[I_p, 0]$ selects the first p rows of $D - \rho E \boldsymbol{\eta}_n \mathbf{z}^T$; hence the $\rho \mathbf{z}^T$ term makes no contribution to the minimisation. Using this, (5.11) is equivalent to

$$\min_{\boldsymbol{\eta}_n} \|F_p + \rho E_p \boldsymbol{\eta}_n \mathbf{z}^T\|_F^2 \tag{5.13}$$

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 Eigenvector conditioning

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. Again the overcoming of the non-linearity introduced by $\rho\boldsymbol{\eta}_n$ is explained in the combined minimisation section.

5.5 Left eigenspace error

So far, when formulating the minimisation algorithm, we have aimed to match the mode input coupling vectors while taking care not to make the problem ill-conditioned. However, we have done this by updating the vectors in V_2, and have restricted them to be in subspaces corresponding to certain eigenvalues, but have not said what these eigenvalues should be. There are two choices

1. retain the Λ_2 corresponding to the V_2 calculated from the original partial eigenstructure assignment by output feedback (i.e. retain the unassigned eigenvalues),

2. change at least one of the values in Λ_2 to give a new set $\tilde{\Lambda}_2$.

This gives the advantage of being able to change some of the unassigned eigenvalues if, for example, one of them was unstable. But this also means that we are trying to perform full eigenstructure assignment in the algorithm, in that we want to assign the full sets $\tilde{V} = [V_1, \tilde{V}_2]$ and $\tilde{\Lambda} = [\Lambda_1, \tilde{\Lambda}_2]$ where the tilde denotes the matrix has been changed from that resulting from the original output feedback.

As demonstrated in Section 4.2.1, full eigenstructure assignment requires both the right and left eigenvectors to be in certain eigenspaces simultaneously, which is not, in general, possible. In choosing $\mathbf{v}_n = S_n\boldsymbol{\eta}_n$ we are ensuring that the right eigenvectors are in their correct spaces, so we need to consider minimising the distance of the left eigenvectors from their correct spaces.

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

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

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

$$\hat{J}_4 = \left\| \begin{pmatrix} \omega_1(\mathbf{z}^T\mathbf{z})^{\frac{1}{2}} F_p\tilde{\mathbf{z}} \\ \omega_3\mathbf{g} \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \omega_1(\mathbf{z}^T\mathbf{z})^{\frac{1}{2}} E_p \\ -\omega_3\tilde{D}E \\ \omega_3\delta_n S_n \\ \omega_2 E \\ \omega_2 S_n \end{pmatrix} (\rho\boldsymbol{\eta}_n) \right\|_2^2 . \tag{5.36}$$

Next we show the step to remove the non-linearity in (5.36).

5.6.2 Scaling

As noted many times, $\boldsymbol{\eta}_n$ occurs non-linearly in (5.36), so we aim to fix the scaling of $\rho\boldsymbol{\eta}_n$. To do this we first need a definition.

Definition 5.2 *(Golub and Van Loan [24]) : Let* $\mathbf{v} \in \mathbb{R}^n$ *be non-zero. An orthogonal* $n \times n$ *matrix* P_H *of the form*

$$P_H = I - \frac{2\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \tag{5.37}$$

is known as a **Householder transformation**.

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 $\mathbf{x} \in \mathbb{R}^n$ it is easy to construct $\mathbf{v}$ in (5.37) such that $P_H\mathbf{x}$ is a multiple of $\mathbf{e}_1$, the first column of I_n.

Proceeding, we find the orthogonal Householder transformation, P_H, such that

$$\mathbf{q}^T S_n P_H = \sigma \mathbf{e}_1^T, \tag{5.38}$$

implying

$$\mathbf{q}^T S_n \boldsymbol{\eta}_n = \sigma \mathbf{e}_1^T P_H^T \boldsymbol{\eta}_n. \tag{5.39}$$

Notice here that the l.h.s. of (5.39) is equivalent to the inverse of ρ, so multiplying (5.39) by ρ gives

$$1 = \mathbf{e}_1^T \sigma P_H^T (\rho \boldsymbol{\eta}_n), \tag{5.40}$$

implying that the first element of $\sigma P_H^T(\rho\boldsymbol{\eta}_n)$ is unity; the rest is a general vector. We can thus define $\mathbf{y}$ such that

$$\begin{bmatrix} 1 \\ \mathbf{y} \end{bmatrix} = \sigma P_H^T (\rho \boldsymbol{\eta}_n). \tag{5.41}$$

The Householder matrix is partitioned such that $P_H = [\mathbf{p}_n, \hat{P}]$ giving

$$(\rho\boldsymbol{\eta}_n) = \sigma^{-1} P_H \begin{bmatrix} 1 \\ \mathbf{y} \end{bmatrix} = \sigma^{-1}[\mathbf{p}_n, \hat{P}] \begin{bmatrix} 1 \\ \mathbf{y} \end{bmatrix} = \sigma^{-1}(\mathbf{p}_n + \hat{P}\mathbf{y}). \tag{5.42}$$

This can now be substituted into (5.36), yielding

$$\begin{cases} \min\limits_{\mathbf{y}} \hat{J}_4 \equiv \min\limits_{\mathbf{y}} \|M\mathbf{y} + \mathbf{r}\|_2^2 \\ \\ \text{subject to } \|\mathbf{v}_i\|_2^2 = 1 \ (i = 1, \ldots, n), \end{cases} \tag{5.43}$$

where

$$M = \begin{pmatrix} \omega_1 (\mathbf{z}^T\mathbf{z})^{\frac{1}{2}} E_p \\ -\omega_3 \tilde{D} E \\ \omega_3 \delta_n S_n \\ \omega_2 E \\ \omega_2 S_n \end{pmatrix} \hat{P} \tag{5.44}$$

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 Algorithm

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

4. calculate (and store) all left range spaces $\hat{T}_i$ for $\tilde{\Lambda} = \text{diag}[\Lambda_1, \tilde{\Lambda}_2]$ using the QR decomposition,

$$[(A - \lambda_i I)P_1] = [\hat{T}_i, T_i] \begin{bmatrix} R_{Li} \\ 0 \end{bmatrix} \tag{5.50}$$

5. choose weightings $\omega_1^2,\ \omega_2^2,\ \omega_3^2$ according to design objectives

STEP 2

1. partition $V = [V_{\_}, \mathbf{v}_n]$

2. perform QR on $V_{\_}$ to obtain $Q_1,\ \mathbf{q},\ R$

$$V_{\_} = [Q_1, \mathbf{q}] \begin{bmatrix} R \\ \mathbf{0}^T \end{bmatrix} \tag{5.51}$$

3. select S_n from right null space store

4. calculate

$$\begin{aligned} \mathbf{z}^T &= \mathbf{q}^T B \\ D_p &= [I_p, 0] R^{-1} Q_1^T B \\ E_p &= [I_p, 0] R^{-1} Q_1^T S_n \\ F_p &= G_{1d} - D_p \\ \tilde{\mathbf{z}} &= \frac{\mathbf{z}}{(\mathbf{z}^T \mathbf{z})} \end{aligned} \tag{5.52}$$

5. calculate necessary left range space components

$$
\begin{aligned}
&\text{for } i = 1 : n \\
&\quad \text{select } \hat{T}_i \text{ from left range space store} \\
&\quad \text{calculate} \\
&\qquad \mathbf{k}_i^T = \mathbf{q}^T \hat{T}_i \\
&\qquad H_i = R^{-1} Q_1^T \hat{T}_i \\
&\qquad \delta_i = (\mathbf{k}_i^T \mathbf{k}_i)^{\frac{1}{2}} \\
&\qquad g_i = \mathbf{e}_i^T \delta_i^{-1} H_i \mathbf{k}_i \\
&\qquad \mathbf{g} = [g_1, \ldots, g_{n-1}]^T \\
&\text{end} \\
&\quad \text{calculate} \\
&\quad \tilde{D} = \text{diag}(\delta_i) \\
&\quad \text{select } \hat{T}_n \text{ from left range space store} \\
&\quad \mathbf{k}_n^T = \mathbf{q}^T \hat{T}_n \\
&\quad \delta_n = (\mathbf{k}_n^T \mathbf{k}_n)^{\frac{1}{2}}
\end{aligned}
\tag{5.53}
$$

6. calculate Householder transformation as in (5.38) using the QR decomposition,

$$
\begin{aligned}
S_i^T \mathbf{q} &= Q \begin{pmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = Q r_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\
\sigma &= |r_1| \\
P_H &= \frac{r_1 Q}{\sigma}
\end{aligned}
\tag{5.54}
$$

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

8. form M, $\mathbf{r}$ as in (5.44) and (5.45) respectively, and solve for $\mathbf{y}$ the following

$$
\min_{\mathbf{y}} \|M\mathbf{y} + \mathbf{r}\|_2^2 \tag{5.55}
$$

STEP 3

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

$$
\tilde{\mathbf{v}}_n = \frac{S_n P_H [1, \mathbf{y}]^T}{\|S_n P_H [1, \mathbf{y}]^T\|_2} \tag{5.56}
$$

2. re-order V_2 and $\tilde{\Lambda}_2$ to give

$$\begin{aligned}\tilde{V}_2 &= [\tilde{\mathbf{v}}_n, \mathbf{v}_{p+1}, \ldots, \mathbf{v}_{n-1}] \\ \tilde{V} &= [V_1, \tilde{V}_2] \\ \tilde{\Lambda}_2 &= \text{diag}[\lambda_n, \lambda_{p+1}, \ldots, \lambda_{n-1}] \\ \tilde{\Lambda} &= \text{diag}[\Lambda_1, \tilde{\Lambda}_2]\end{aligned} \tag{5.57}$$

3. calculate $\tilde{W}^T = \tilde{V}^{-1}$ and form $\tilde{G} = \tilde{W}_1^T B$

4. calculate value of objective function

$$J_4 = \omega_1^2 \|G_{1d} - \tilde{G}_1\|_F^2 + \omega_2^2 \|\tilde{V}^{-1}\|_F^2 + \omega_3^2 \sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2 \tag{5.58}$$

This algorithm is then repeated from Step 2 with $V = \tilde{V}$ until J_4 has decreased to some required level, or until the decrease in J_4 is less than some tolerance. However, a word of caution is needed concerning the preservation of a self-conjugate set of vectors.

5.7.1 Preservation of self-conjugacy

A note is needed here on the existence of complex eigenvalues in the closed loop system. In Problems $(1-4)$ of Chapter 4, it is always a condition that the eigenvalues and eigenvectors both be self-conjugate sets, so the computed feedback will be real. Here, if $\mathbf{v}_n$ corresponds to a real eigenvalue λ_n, then a real update $\tilde{\mathbf{v}}_n$ is generated. If λ_n is complex then a complex update is generated. Hence, in order that the eigenvector set remains self-conjugate, the update corresponding to $\bar{\lambda}_n$ must be $\bar{\tilde{\mathbf{v}}}_n$, the complex conjugate of the computed update for λ_n.

5.7.2 Main algorithm summary

Here we have shown the need for a minimisation algorithm due to the usual disregard of the left eigenvectors when considering decoupling objectives in the design problem. We have given a minimisation algorithm that aims to select a new set of unassigned closed loop right vectors, in order to match the first p left vectors. We have also considered two other objectives in addition to that of left eigenvector matching for input decoupling purposes : namely robustness

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

$$\omega_1^2 \ \|G_{1d} - G_{1a}\|_F^2 + \omega_2^2 \ \|V^{-1}\|_F^2 + \omega_3^2 \sum_{i=1}^{n} \|\mathbf{w}_i^T \hat{T}_i\|_2^2 \tag{5.60}$$

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

$$\kappa_F(V) = n^{\frac{1}{2}} \|V^{-1}\|_F \tag{5.61}$$

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 vector, so

$$1 \text{ sweep} \equiv 6 \text{ iterations}. \tag{5.62}$$

Parameters: $(\omega_1^2, \omega_2^2, \omega_3^2) = (1, 0, 0)$

This is run with zero weighting on the eigenvector conditioning and the left eigenspace error because the left eigenvector matching is the primary aim of the

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

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	$3.7026e+03$	$3.7026e+03$	$1.6794e+03$	$3.3988e-27$
1	$4.5143e+02$	$4.5143e+02$	$8.8881e+04$	$1.2705e+06$
2	$1.2896e+02$	$1.2896e+02$	$1.2001e+05$	$1.9152e+05$
3	$9.5802e+01$	$9.5802e+01$	$7.1102e+04$	$2.1637e+06$
4	$8.9219e+01$	$8.9219e+01$	$7.2757e+04$	$2.3231e+06$
5	$8.4611e+01$	$8.4611e+01$	$7.1013e+04$	$2.1500e+06$
10	$6.8966e+01$	$6.8966e+01$	$6.4637e+04$	$1.5201e+06$
15	$5.8685e+01$	$5.8685e+01$	$6.0301e+04$	$1.1290e+06$
30	$4.0445e+01$	$4.0445e+01$	$5.3367e+04$	$5.5174e+05$
80	$1.9342e+01$	$1.9342e+01$	$4.9260e+04$	$1.9242e+05$

From these results we can clearly see that the left eigenvector matching has been improved quite considerably, but that the conditioning of the eigenvectors is quite large (as is the left eigenspace error). Also to note is the fact that most of the work is done in the first 3 sweeps of the routine and that a lot of computation is required to reduce the objective function by only a small amount further.

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

Following the results of the previous test it is obvious that we should also include weightings on the second and third criteria, but what values? We enforce equal weightings and use the results as an indication on how to choose the parameters.

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	$1.0535e+04$	$3.7026e+03$	$1.6794e+03$	$3.3988e-27$
1	$2.3291e+03$	$5.5189e+02$	$8.0783e+02$	$1.9618e+02$
2	$8.6477e+02$	$3.9368e+02$	$4.3491e+02$	$1.2861e+01$
4	$4.0250e+02$	$6.2708e+01$	$3.6585e+02$	$1.5536e+01$
10	$3.5409e+02$	$4.7934e+01$	$3.5048e+02$	$8.5580e+00$
30	$3.5344e+02$	$4.8581e+01$	$3.4978e+02$	$8.4572e+00$

From these results we can see that left eigenvector matching has still been improved considerably. In addition, the conditioning of the eigenvectors has been

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

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

Although the previous results show an improvement, we remark that we may have gone to the other extreme in weighting all the criteria equally, because our primary aim is still to decrease the error in the left eigenvector matching. We therefore keep weightings of unity for ω_2^2 and ω_3^2, but increase the relative weighting on ω_1^2. This is to keep a check on the conditioning and the left eigenspace error, but to further improve the matching.

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	$3.7710e+05$	$3.7026e+03$	$1.6794e+03$	$3.3988e-27$
1	$2.5467e+04$	$1.8287e+02$	$1.7103e+03$	$9.3682e+01$
2	$6.9534e+03$	$3.7437e+01$	$1.1257e+03$	$1.3945e+02$
10	$3.5323e+03$	$2.5073e+01$	$6.1301e+02$	$1.1467e+02$
25	$3.0253e+03$	$1.9372e+01$	$6.4599e+02$	$7.7172e+01$

Here the results are as expected; we have further reduced the value of the matching error, at the expense of small increments in the other two values.

5.8.1 Convergence histories

From the previous examples, we can see that most of the reduction work is done in the first few sweeps. We now give a graph of the error in the left eigenvector matching for the three previous test cases. This can been seen in Figure 5.1. We can also see that sometimes the value increases; there are two reasons for this,

1. the complex conjugate problem explained in Subsection 5.7.1 whereby the second vector is chosen to be the complex conjugate of the previous, not to minimise the objective function,

2. if the weightings ω_2 and ω_3 are not zero then the error in the left matching can increase at the expense of the conditioning or the left eigenspace error decreasing, but the value of the objective function still decreases.

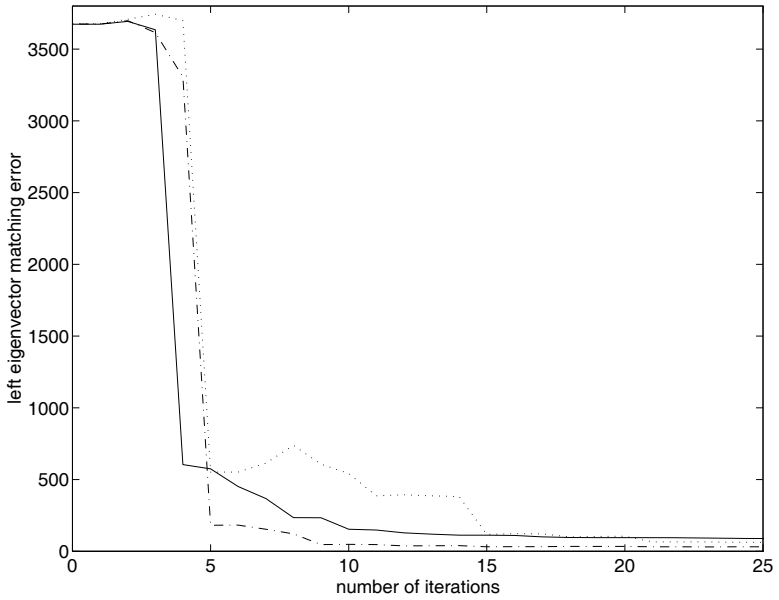

Figure 5.1: Convergence histories

Indeed, we should really show the convergence histories of the overall objective function, but due to the relative weightings, the value of the objective functions range from $O(10^3)$ to $O(10^7)$ and hence easy comparison would not be possible on the same graph.

5.9 Optimal scaling of assigned right vectors, V_1

In the preceding sections, a minimisation routine was developed that updated only the last $n-p$ columns of the eigenvector set, V. This was desirable in order to retain

$$G_{0d} = CV_1, \tag{5.63}$$

where $V = [V_1, V_2]$. The main aim of the algorithm is to solve

$$\min_{\mathbf{v}_i \in V_2} J_1 \quad \equiv \min_{\mathbf{v}_i \in V_2} \|G_{1d} - W_1^T B\|_F^2, \tag{5.64}$$

with the eigenvector conditioning and the left eigenspace error also being taken into consideration. However, since G_0 is column normalised to obtain a 1 in the correct position to reflect coupling, the actual value of the individual elements is

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, and hence we can select each d_i separately to minimise $\tilde{J}_1$ as in (5.67). Now to solve $\min_{D_1} \tilde{J}_1$ we must solve

$$\begin{aligned} \min_{d_i} \tilde{J}_i &\equiv \min_{d_i} (\mathbf{g}_i^H - d_i \tilde{\mathbf{w}}_i^H)(\mathbf{g}_i - d_i \tilde{\mathbf{w}}_i) \qquad (\forall i = 1, \ldots, p) \\ &= \min_{d_i} (\mathbf{g}_i^H \mathbf{g}_i - d_i \mathbf{g}_i^H \tilde{\mathbf{w}}_i - d_i \tilde{\mathbf{w}}_i^H \mathbf{g}_i + d_i^2 \tilde{\mathbf{w}}_i^H \tilde{\mathbf{w}}_i) \quad (\forall i = 1, \ldots, p). \end{aligned} \tag{5.69}$$

To find the minimum of this expression, differentiate with respect to d_i and set to zero, giving

$$\frac{\partial J_i}{\partial d_i} = 0 \quad \Leftrightarrow \quad -\mathbf{g}_i^H \tilde{\mathbf{w}}_i - \tilde{\mathbf{w}}_i^H \mathbf{g}_i + 2 d_i \tilde{\mathbf{w}}_i^H \tilde{\mathbf{w}}_i = 0, \tag{5.70}$$

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.

Parameters: $(\omega_1^2, \omega_2^2, \omega_3^2) = (1, 0, 0)$

Here we are only trying to improve the matching of the left eigenvectors, and hence set the weights of the other two criteria to zero.

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	$3.7026e+03$	$3.7026e+03$	$1.6794e+03$	$3.3988e-27$
1	$1.8392e+00$	$1.8392e+00$	$2.3616e+05$	$2.3394e+04$
2	$1.2485e+00$	$1.2485e+00$	$4.2128e+05$	$7.6534e+05$
3	$6.6759e-01$	$6.6759e-01$	$2.9691e+05$	$2.9802e+05$
5	$3.3792e-01$	$3.3792e-01$	$3.5561e+05$	$1.7860e+06$
10	$2.3350e-01$	$2.3350e-01$	$4.4380e+05$	$2.8500e+06$

We can see than the introduction of scaling on the V_1 set vastly improves the attainable accuracy.

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

Here we include the conditioning of the eigenvectors and on the left eigenspace error in the minimisation.

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	$1.1619e+04$	$3.7026e+03$	$1.6794e+03$	$3.3988e-27$
1	$1.6499e+03$	$2.6548e+00$	$4.9772e+04$	$1.7793e+02$
5	$3.0928e+02$	$2.7200e+00$	$2.0349e+05$	$7.9252e+00$
10	$2.9577e+02$	$2.5593e+00$	$9.1712e+04$	$7.7595e+00$
50	$2.8958e+02$	$2.2189e+00$	$6.3398e+04$	$9.6130e+00$

From these results we can again see that the left eigenvector matching is improved quite considerably, however this a large difference from before. As we can see from both scaling examples, the conditioning of the eigenvectors is very large in comparison with Subsection 5.8. However, we have noted that this just means that we have increased the value of the upper bound for the individual eigenvalue sensitivities, it does not actually affect the individual eigenvalue sensitivities. From experimentation we have observed that the scaling has little effect on the left eigenspace error, since it only affects the first p eigenmodes and then only

changes their values by a negligible amount. More theoretical work may be done on the exact effect of scaling on the value of the conditioning of the eigenvectors and the left eigenspace error. Note that the algorithm seems to have converged at around 50 sweeps.

5.10 Alternative scaling of assigned right vectors, V_1

We perform the minimisation to reduce the error between G_{1d} and G_{1a}. However, we are more concerned with the errors between the ratios of the components in each row of G_{1d} and G_{1a}. If we reduce $\|G_{1d} - G_{1a}\|_F^2$ significantly, then we obtain small errors between these ratios, but we have shown that, due to dimensional restrictions, this is not always possible. In fact, in reducing the matching error, we may actually increase these ratios. For example, if for one row $G_{1d} = [1\ 0\ 0]$, $G_{1a} = [10\ 1\ 1]$ then $\|G_{1d} - G_{1a}\|_F^2 = 83$ and the ratios are $G_{1a} = [1\ 0.1\ 0.1]$. If the minimisation gives the result $G_{1a} = [2\ 1\ 1]$, then $\|G_{1d} - G_{1a}\|_F^2 = 3$, but the ratios are $G_{1a} = [1\ 0.5\ 0.5]$, i.e. they have increased even though the left matching error has decreased.

To this end we may choose the scaling matrix, D_1, in (5.65), so that $G_{1a} = W_1^T B$ is row normalised by those elements that correspond to a 1 in G_{1d}. In implementing this scaling, we can show values that below which the left matching error must fall in order to ensure the desired level of input decoupling.

THEOREM 5.3 *If $\hat{G}_{1a}$ denotes that G_{1a} is normalised so that the largest element (in modulus) in each row is unity, then*

$$\|G_{1d} - \hat{G}_{1a}\|_F^2 \leq \tau^2 p(m-1) \tag{5.72}$$

is a necessary condition to obtain the desired level of input decoupling in that the non-unity elements of $\hat{G}_{1a}$ are less than τ. Also

$$\|G_{1d} - \hat{G}_{1a}\|_F^2 \leq \tau^2 \tag{5.73}$$

is a sufficient condition.

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 :

$$\|G_{1d} - \hat{G}_{1a}\|_F^2 < \tau^2 \Rightarrow \max \left|\frac{g_{i1}}{g_{ij}}\right|^2 \leq \sum_k \left|\frac{g_{ik}}{g_{ij}}\right|^2 < \tau^2 \square. \tag{5.77}$$

For our purposes here we choose $\tau = 0.1$. This means that if the matching error goes below $0.01p(m-1)$, then we may have achieved the desired level of input decoupling, but it is not guaranteed until the error reaches $O(10^{-3})$.

Parameters: $(\omega_1^2, \omega_2^2, \omega_3^2) = (1, 0, 0)$

Here we give results of the alternative scaling on V_1.

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	$3.7026e+03$	$3.7026e+03$	$1.6794e+03$	$3.3988e-27$
1	$1.4152e+00$	$1.4152e+00$	$1.5123e+06$	$6.7055e+06$
2	$7.4578e-01$	$7.4578e-01$	$1.5447e+06$	$5.0309e+06$
3	$6.0487e-01$	$6.0487e-01$	$1.1627e+06$	$2.9052e+06$
4	$5.8745e-01$	$5.8745e-01$	$1.0030e+06$	$2.2574e+06$
5	$5.9557e-01$	$5.9557e-01$	$9.8594e+05$	$2.0898e+06$

Even though we have not reached the desired level of decoupling, this scaling ensures the largest element in each row is in the correct place.

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 point

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 follows:

1. specify design requirements Λ_p, G_{0d}, G_{1d},
2. perform the projection as in Section 4.3.7 to obtain the best set of achievable vectors, G_{0a}; from this extract $V_1 \equiv V_a$,
3. specify a set of eigenvalues , Λ_{n-p}, Take an initial V_2 from the null spaces of the $\lambda \in \Lambda_{n-p}$,
4. set

$$\Lambda = \begin{bmatrix} \Lambda_p & 0 \\ 0 & \Lambda_{n-p} \end{bmatrix} \quad , \quad V = [V_1, V_2]. \tag{5.78}$$

Here the computation is reduced by not having to construct the feedback matrix. However, we expect to need more work in the minimisation algorithm to reduce the objective function, especially the left eigenspace error since the initial eigenstructure will most likely not be exactly assignable by any K.

5.12 Conclusions

This chapter has described the first of the new work. A minimisation routine has been developed to improve the unsatisfactory results generated by output feedback with respect to the coupling inherent in the left eigenvectors. The algorithm

iterates through the unassigned right vectors until some minimisation criteria is satisfied. In the development of the theory for the algorithm it has been shown that it is wise to include measures of the eigenvector conditioning and the left eigenspace error.

The algorithm was run for an example with various parameter weightings to illustrate the trade off achievable between the left vector matching, the vector conditioning and the left eigenspace error.

We have also included theory on how V_1 can be further scaled to make improvements to the left eigenvector matching, but more work may be done to analyse the exact effects on the conditioning bound and the left eigenspace error. The same example was used as a comparison to that performed without the scaling.

While the algorithm does reduce the objective function, it is not always to the desired level. This is due to the small dimension of the subspace from which the minimisation vectors are chosen.

This is overcome in the next chapter where we allow the minimisation vectors to be free.

Chapter 6

Unrestricted minimisation algorithm

As we have seen in the chapter on the subspace restricted minimisation, the theory allows us to improve on the results of the partial eigenstructure assignment algorithm via a minimisation routine. In this routine we can choose weightings to put an emphasis on the improvement of the decoupling, the robustness of the system or the accuracy of the desired eigenvalues.

However, from the examples, it can be seen that a lot of computation is required to improve the specified objective function. This is due to the fact that any vector chosen to decrease the objective function must lie in the subspace given by

$$\mathcal{S}_i \equiv \mathcal{N}[U_1^T(A - \lambda_i I)], \tag{6.1}$$

where U_1^T is calculated from the QR decomposition of the input matrix, B. This space has dimension m and hence the updated vector is made up of a linear combination of these m columns. Since m is usually small, this means the minimisation algorithm decreases the objective function quite slowly.

This suggests that we would like to increase the size of the allowable subspace. One idea is to replace a single desired eigenvalue by a desired eigenvalue region, but it is not clear if it is possible to specify a null space corresponding to a region in the imaginary plane.

Instead we go to the extreme and allow the new vector to lie anywhere in the complex plane. This will allow the algorithm to decrease the value of the

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 $\hat{J}_1$. As before, we explain the second criteria to be satisfied before describing how the non-linearity in $\rho \mathbf{v}_n$ can be dealt with.

6.3 Eigenvector conditioning

In the unrestricted minimisation, as with the restricted minimisation, the objective function is calculated from an inversion of V; thus we must take precautions

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

$$\begin{aligned} J_2 \equiv \|V^{-1}\|_F^2 &= \text{trace}(V^{-1}V^{-T}) \\ &= \sum_{i=1}^{n-1} \mathbf{e}_i^T (R^{-1}R^{-T} + \rho E \mathbf{v}_n \mathbf{v}_n^T E^T \rho^T)\mathbf{e}_i + \rho\rho^T \\ &= \sum_{i=1}^{n-1} \|\mathbf{e}_i^T(\rho E \mathbf{v}_n)\|_2^2 + \mathbf{e}_i^T R^{-1}R^{-T}\mathbf{e}_i + \rho\rho^T \\ &= \|\rho E \mathbf{v}_n\|_2^2 + \|\rho \mathbf{v}_n\|_2^2 + \sum_{i=1}^{n-1} \beta_i, \end{aligned} \tag{6.12}$$

where

$$\begin{cases} \|\rho \mathbf{v}_n\|_2^2 = \rho\rho^T \\ \\ \beta_i = \mathbf{e}_i^T R^{-1}R^{-T}\mathbf{e}_i, \end{cases} \tag{6.13}$$

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

$$\begin{aligned} \min_{\mathbf{v}_n} \hat{J}_2 &\equiv \min_{\mathbf{v}_n} \left[\|\rho E \mathbf{v}_n\|_2^2 + \|\rho \mathbf{v}_n\|_2^2 \right] \\ &= \min_{\mathbf{v}_n} \left\| \begin{bmatrix} E \\ I_n \end{bmatrix} \rho \mathbf{v}_n \right\|_2^2, \end{aligned} \tag{6.14}$$

since β_i is independent of $\mathbf{v}_n$ for all i. Note that I_n is the n-dimensional identity vector. Again the overcoming of $\rho\mathbf{v}_n$ as a non-linear term is explained in the combined minimisation section.

6.4 Combined minimisation

Obviously here, since we are not restricting the new vectors to lie in certain subspaces, there is no left eigenspace error as in Chapter 5. Thus we have just

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

$$\begin{aligned} J_3 &\equiv (\omega_1^2 J_1 + \omega_1^2 J_2) \\ &= \left[\omega_1^2 \ \|G_{1d} - [I_p, 0] W^T B\|_F^2 + \omega_2^2 \ \|V^{-1}\|_F^2 \right], \end{aligned} \tag{6.15}$$

where ω_1^2 and ω_2^2 are weightings to be chosen. Using the expressions found in (6.10) and (6.14), the objective function is

$$\hat{J}_3 \equiv \left[\omega_1^2 (\mathbf{z}^T \mathbf{z}) \|F_p \tilde{\mathbf{z}} + E_p (\rho \mathbf{v}_n)\|_2^2 + \omega_2^2 \left\| \begin{bmatrix} E \\ I_n \end{bmatrix} (\rho \mathbf{v}_n) \right\|_2^2 \right]. \tag{6.16}$$

Using $\|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x}$, (6.16) can be re-written to give

$$\hat{J}_3 = \left\| \begin{pmatrix} \omega_1 (\mathbf{z}^T \mathbf{z})^{\frac{1}{2}} F_p \tilde{\mathbf{z}} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \omega_1 (\mathbf{z}^T \mathbf{z})^{\frac{1}{2}} E_p \\ \omega_2 E \\ \omega_2 I_n \end{pmatrix} (\rho \mathbf{v}_n) \right\|_2^2. \tag{6.17}$$

Next we show the step to remove the non-linearity in (6.17).

6.4.1 Scaling

We deal with $\rho \mathbf{v}_n$ by fixing its scaling. We find the orthogonal Householder transformation, P_H, such that

$$\mathbf{q}^T P_H = \sigma \mathbf{e}_1^T, \tag{6.18}$$

and define $\mathbf{y}$ such that

$$\begin{bmatrix} 1 \\ \mathbf{y} \end{bmatrix} = \sigma P_H^T (\rho \mathbf{v}_n). \tag{6.19}$$

We partition $P_H = [\mathbf{p}_n, \hat{P}]$ and substitute

$$\rho \mathbf{v}_n = \sigma^{-1} (\mathbf{p}_n + \hat{P} \mathbf{y}) \tag{6.20}$$

from (6.19) into (6.17) to give

$$\begin{cases} \min_{\mathbf{y}} \hat{J}_3 \equiv \min_{\mathbf{y}} \|M \mathbf{y} + \mathbf{r}\|_2^2 \\ \\ \text{subject to } \|\mathbf{v}_i\|_2^2 = 1 \ (i = 1, \ldots, n) \end{cases}, \tag{6.21}$$

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 have again reduced the problem to a standard linear least squares problem which can therefore be solved by a QR (or SVD) method. To reconstruct the scaling we must calculate

$$\mathbf{v}_n = \frac{P_H[1,\mathbf{y}]^T}{\|P_H[1,\mathbf{y}]^T\|_2}. \tag{6.24}$$

6.5 Algorithm

To clarify the procedure, we write the algorithm in list form, assuming that we initially have a linearly independent set of vectors V.

STEP 1

1. normalise columns of V so that $\|\mathbf{v}_i\|_2^2 = 1$ $(i = 1, \ldots, n)$
2. re-order V so that $V = [V_1, V_2]$,
3. choose weightings $\omega_1^2,\ \omega_2^2$ according to design objectives

STEP 2

1. partition $V = [V_{\_}, \mathbf{v}_n]$
2. perform QR on $V_{\_}$ to obtain $Q_1,\ \mathbf{q},\ R$

$$V_{\_} = [Q_1, \mathbf{q}] \begin{bmatrix} R \\ \mathbf{0}^T \end{bmatrix} \tag{6.25}$$

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

$$\begin{aligned} \tilde{V}_2 &= [\tilde{\mathbf{v}}_n, \mathbf{v}_{p+1}, \ldots, \mathbf{v}_{n-1}] \\ \tilde{V} &= [V_1, \tilde{V}_2] \end{aligned} \tag{6.30}$$

3. calculate $\tilde{W}^T = \tilde{V}^{-1}$ and form $\tilde{G}_1 = \tilde{W}_1^T B$

4. calculate value of objective function

$$J_3 = \omega_1^2 \|G_{1d} - \tilde{G}_1\|_F^2 + \omega_2^2 \|\tilde{V}^{-1}\|_F^2 \tag{6.31}$$

This algorithm is then repeated from Step 1 with $V = \tilde{V}$ until J_3 has decreased to some required level, or until the decrease in J_3 is less than some tolerance.

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 Selection of initial vector set for algorithm

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 example is given to illustrate the theory; the minimisation routine is run for different parameter weightings. The example is the same for each section and is the same as for Section 5.8 of the restricted minimisation chapter. The results are then compared.

6.6.1 Results of partial eigenstructure assignment as a starting point

In Chapter 5, we used the results of partial eigenstructure assignment as the starting point for the algorithm. We may also do the same here. We calculate the eigenvectors of the closed loop matrix, $A+BKC$, and partition them so that $V = [V_1, V_2]$, and the routine is run to find a new set $\hat{V}_2$ to minimise

$$J_3 \equiv \left[\omega_1^2 \|G_{1d} - G_{1a}\|_F^2 + \omega_2^2 \ \|V^{-1}\|_F^2 \right]. \tag{6.32}$$

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

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

and the corresponding real representation of the eigenvalue matrix is

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

where the subscript 're' denotes the real form. If there are more complex modes, P can be augmented with

$$\begin{bmatrix} \frac{1}{2} & -\frac{1}{2}i \\ \frac{1}{2} & \frac{1}{2}i \end{bmatrix} \tag{6.37}$$

repeated along the block diagonal.

Following partial eigenstructure assignment, we can thus transform the complex eigenvector matrix into its real representation, and run the minimisation algorithm. This generates a real set of vectors that best satisfies the specified minimisation criteria.

6.6.2 Example 1

After partial eigenstructure assignment, the design specifications of the eigenvalues and the mode output coupling vectors are achieved. The closed loop modal eigenvector matrix is in the form

$$V_{\text{cc}} = [\mathbf{v}_1^{\text{re}} + i\mathbf{v}_1^{\text{im}}, \mathbf{v}_1^{\text{re}} - i\mathbf{v}_1^{\text{im}}, \mathbf{v}_3, \ldots, \mathbf{v}_8, \mathbf{v}_9^{\text{re}} + i\mathbf{v}_9^{\text{im}}, \mathbf{v}_9^{\text{re}} - i\mathbf{v}_9^{\text{im}}] \tag{6.38}$$

and is transformed it into its real representation,

$$V_{\text{re}} = [\mathbf{v}_1^{\text{re}}, \mathbf{v}_1^{\text{im}}, \mathbf{v}_3, \ldots, \mathbf{v}_8, \mathbf{v}_9^{\text{re}}, \mathbf{v}_9^{\text{im}}]. \tag{6.39}$$

The mode input coupling vectors then have the residual error,

$$\|G_{1d} - G_{1a}\|_F^2 = 3.7107 \times 10^3. \tag{6.40}$$

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 vector, hence

$$1 \text{ sweep} \equiv 6 \text{ iterations}. \tag{6.43}$$

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

We first test the algorithm with a zero weighting on the conditioning so that we are only attempting to reduce the left vector matching, since this is our primary aim.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$3.7107e+03$	$3.7107e+03$	$1.3133e+03$
1	$7.4517e+02$	$7.4517e+02$	$6.0626e+03$
2	$6.1801e-01$	$6.1801e-01$	$4.6071e+02$
3	$6.9720e-04$	$6.9720e-04$	$3.8902e+02$
4	$4.0517e-05$	$4.0517e-05$	$3.8901e+02$
5	$1.6341e-06$	$1.6341e-06$	$3.8900e+02$
10	$1.2209e-14$	$1.2209e-14$	$3.8900e+02$
15	$1.4849e-24$	$1.4849e-24$	$3.8900e+02$

We can see from these results that allowing the updating vectors to be chosen freely greatly improves the speed and accuracy of the algorithm. The error in the left matching has been reduced to zero (to machine accuracy, which here is $O(10^{-16})$), but the conditioning of the eigenvectors is relatively high.

Convergence 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.

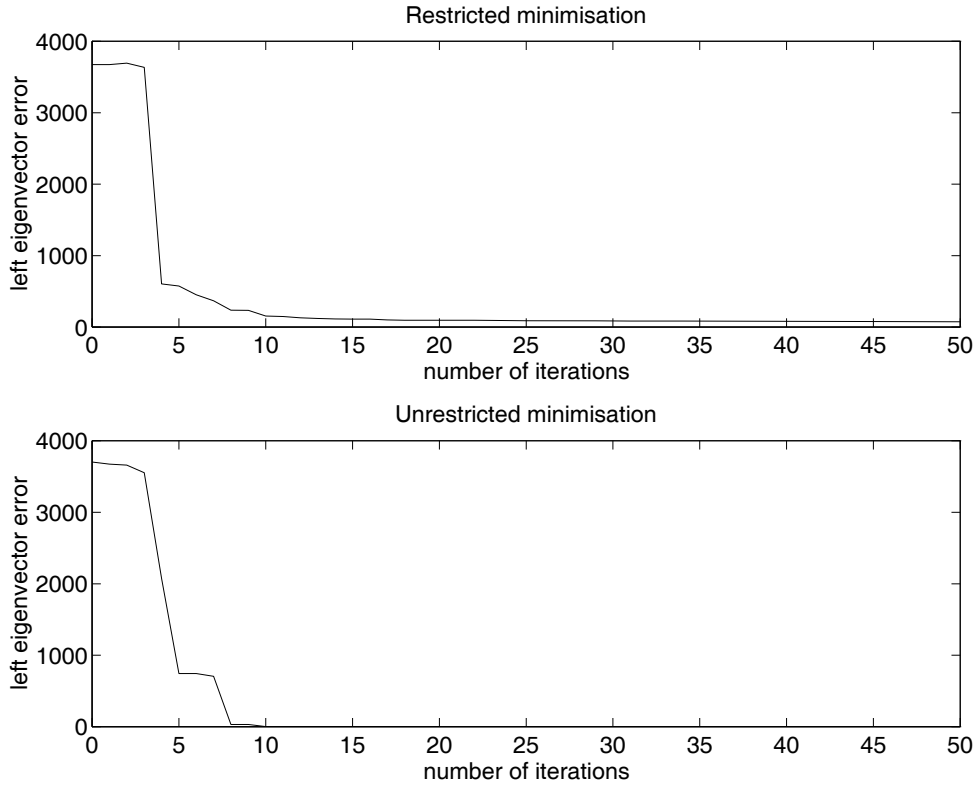

Figure 6.1: Comparison of convergence histories

From these we can see that both algorithms perform most of the reduction in the error norm in the first few iterations. Both seem to reduce the error quite considerably, but we need to look closer at where the error tends to zero to see the real difference between the two.

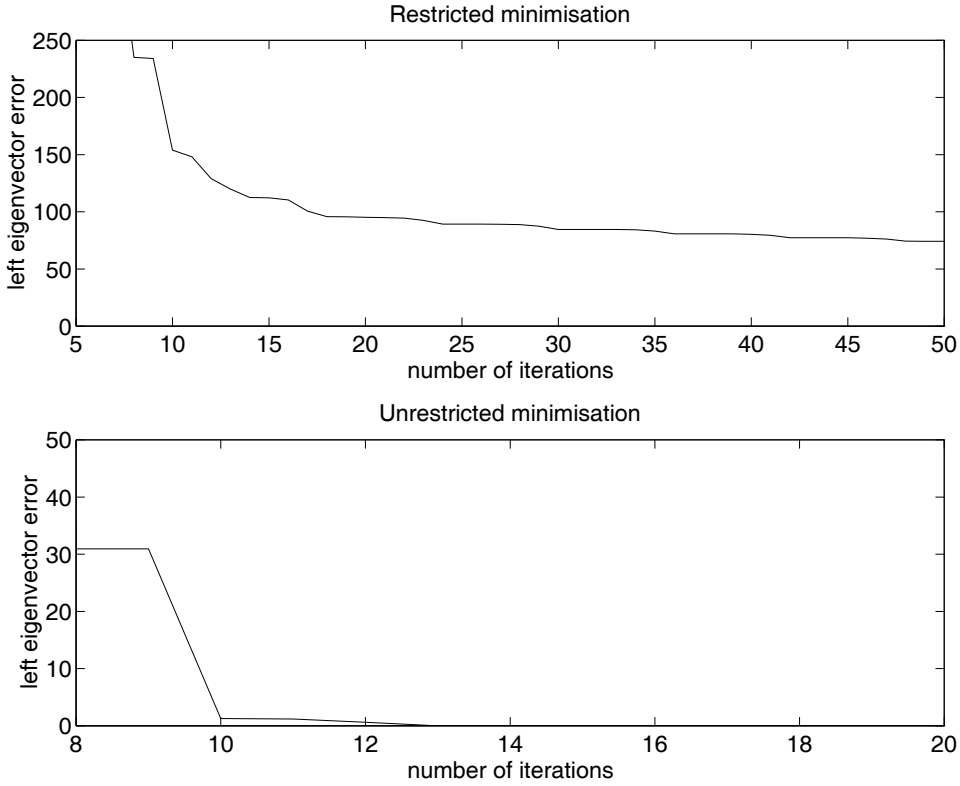

Figure 6.2: Comparison of convergence histories (as error tends to zero)

We can clearly see that, for the restricted case, the error is reduced significantly in the first few iterations, and has in fact reduced it by $O(10^2)$ within one sweep. However, a lot of extra work is needed to reduce the error further. For the unrestricted case, although most of the error reduction occurs within the first sweep, the algorithm further reduces the error within only a few more iterations.

These graphs show that both cases of the minimisation algorithm initially reduce the error significantly, but that the restricted minimisation struggles after this due to the fact that the updating vectors must lie in specific subspaces; the unrestricted minimisation further reduces the error because the vectors are restricted only to lie in the complex plane.

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

The previous example illustrates how the left vectors can be perfectly matched. However, the conditioning of the vectors was relatively high; we thus put an equal

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

$$V_2 = \begin{bmatrix} I_{n-p} \\ I_{n-p} \\ X_2 \end{bmatrix}, \tag{6.46}$$

where $X_2 \in \mathbb{R}^{2p-n\times n-p}$. This is repeated until step j, where the number of rows in X_j is less than $n-p$. If there are r rows in X_j, then let X_j be the first r rows of

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 have to be transformed into reals; these would be overwritten with new real vectors not corresponding to any eigenvalues . Thus it is not worth constructing K just to find the extra V_2, which is overwritten, and Λ_2, which is not used. It is more sensible to perform the projection method to find V_1 and choose an initial V_2.

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 we have very low condition numbers,

$$\kappa_F(V) \approx 76, \quad \kappa_F(V) \approx 38 \tag{6.53}$$

for the two tables respectively. This conditioning has risen in both cases because it is weighted zero.

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

Here we include the conditioning in the routine. Both criteria are given equal weighting.

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$1.1781e+04$	$1.1757e+04$	$7.5912e+01$
1	$1.4842e+02$	$2.6051e+01$	$1.6845e+02$
5	$5.3434e+01$	$1.6956e+01$	$9.1970e+01$
10	$4.4455e+01$	$1.4723e+01$	$8.3032e+01$
20	$2.9666e+01$	$9.5420e+00$	$6.8311e+01$
50	$2.7290e+01$	$1.0718e+01$	$6.1991e+01$
100	$2.5614e+01$	$1.0141e+01$	$5.9899e+01$
150	$2.4730e+01$	$9.7828e+00$	$5.8871e+01$

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$9.2053e+03$	$9.1991e+03$	$3.7908e+01$
1	$1.7529e+03$	$5.8288e+01$	$6.2685e+02$
5	$1.0753e+02$	$2.0405e+01$	$1.4214e+02$
10	$4.9469e+01$	$1.0315e+01$	$9.5284e+01$
20	$4.5162e+01$	$9.9813e+00$	$9.0320e+01$
50	$3.8145e+01$	$9.4757e+00$	$8.1533e+01$
100	$2.9732e+01$	$9.1104e+00$	$6.9149e+01$
150	$2.4746e+01$	$9.0061e+00$	$6.0413e+01$

The left vector matching error is reduced for both constructions of V_2, but the conditioning has increased for the second construction.

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.

6.6.5 Generate initial right vector set 'from scratch'

Using the projection method described in the previous section, we obtain the best G_0 that corresponds to a set of p desired eigenvalues. However, it would be a novel idea to be able to run the minimisation routine without specifying any eigenvalues, but by just using G_{0d}. We do this by performing an unconstrained projection, where again the aim is to generate an initial set of linearly independent vectors, V.

Following the theory in Section 4.3.5, trivially express the achievable mode output coupling vectors as

$$\mathbf{g}_{ai} = C\mathbf{v}_{ai} = CS_i\boldsymbol{\eta}_i, \tag{6.54}$$

where in this case, $S_i \equiv I_n$. Permute each $\mathbf{g}_{di}$ so that the desired elements appear first (and permute CS_i accordingly),

$$\mathbf{g}_{di} = \begin{bmatrix} \mathbf{d}_i \\ \mathbf{u}_i \end{bmatrix}, \quad CS_i = \begin{bmatrix} D_i \\ U_i \end{bmatrix}. \tag{6.55}$$

Minimising the difference between the desired and best achievable mode output coupling vectors gives

$$\min_{\mathbf{g}_{ai}} \|\mathbf{g}_{di} - \mathbf{g}_{ai}\|^2 = \min_{\boldsymbol{\eta}_i} \left\| \begin{bmatrix} \mathbf{d}_i \\ \mathbf{u}_i \end{bmatrix} - \begin{bmatrix} D_i \\ U_i \end{bmatrix} \boldsymbol{\eta}_i \right\|_2^2. \tag{6.56}$$

By minimising over the desired elements, the solution is

$$\mathbf{g}_{ai} = CS_iD_i^+\mathbf{d}_i, \tag{6.57}$$

but $S_i \equiv I_n$, so that

$$\begin{cases} \mathbf{g}_{ai} = CD_i^+\mathbf{d}_i \\ \mathbf{v}_{ai} = D_i^+\mathbf{d}_i \end{cases}. \tag{6.58}$$

However, a problem arises here. In minimising over the desired components,

$$\min_{\boldsymbol{\eta}_i} \|\mathbf{d}_i - D_i\boldsymbol{\eta}_i\|_2^2, \tag{6.59}$$

we have that $D_i \in \mathbb{C}^{k\times n}$, $\mathbf{d}_i \in \mathbb{R}^k$, where k is the number of specified elements in the desired mode output coupling vector. Now $k \leq n$ so that, unless equality holds, the system being solved by finding $\boldsymbol{\eta}_i$ is underdetermined. In fact, if there are any unspecified elements in a $\mathbf{g}_{di}$, then $k < n$. The unspecified elements in $\mathbf{g}_{di}$ are set to zero by the least squares routine, as this corresponds to the minimum 2-norm solution. This may not appear to be a problem, but if we consider G_{0d} from the example in Section 4.4,

$$G_{0d} = \begin{bmatrix} x & x & 0 & 0 \\ 0 & 0 & x & x \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \tag{6.60}$$

then the theory just outlined gives

$$G_{0a} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad V_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{6.61}$$

where the latter is obviously not of full column rank. There are two ways to overcome this. The first is to solve the underdetermined system, but not by finding the minimum 2-norm solution, so that the unspecified elements are not automatically set to zero. The other, which we implement, is to choose arbitrary values for the unspecified elements, so that we now have to solve

$$\min_{\mathbf{g}_{ai}} \|\mathbf{g}_{di} - C\mathbf{v}_{ai}\|_2^2, \tag{6.62}$$

giving

$$\begin{cases} \mathbf{v}_{ai} = C^+\mathbf{g}_{di} \\ \mathbf{g}_{ai} = CC^+\mathbf{g}_{di} \end{cases}. \tag{6.63}$$

We remark that for the projection method in Sections 4.3.5 and 6.6.3, where the vectors are restricted to lie in spaces corresponding to specified eigenvalues, $D_i \in \mathbb{C}^{k\times m}$. G_{0d} is chosen so that $k \geq m$ for each vector, and hence the system

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 the unrestricted projection method. From this we obtain the V_1 set. The V_2 set is obtained by cycling the identity matrix, as before. The residual error in G_1 is

$$\|G_{1d} - G_{1a}\|_F^2 = 4.3455 \times 10^5. \tag{6.66}$$

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	$7.8023e+01$	$7.8023e+01$	$1.0362e+02$
6	$9.4424e+00$	$9.4424e+00$	$7.0870e+01$
12	$2.6329e-01$	$2.6329e-01$	$7.3518e+01$
19	$8.7899e-05$	$8.7899e-05$	$7.5613e+01$
49	$4.0981e-14$	$4.0981e-14$	$7.4240e+01$
54	$7.5998e-15$	$7.5998e-15$	$7.4590e+01$
57	$9.7078e-16$	$9.7078e-16$	$7.5415e+01$
70	$9.6098e-17$	$9.6098e-17$	$7.4438e+01$

When the minimisation algorithm was run with these parameter values, an error was produced because of dividing by zero. This was a result of V being sparse; then decompositions involving V (such as the QR) were sparse. Since B and C are not full, this led to $\mathbf{z}^T = \mathbf{q}^T B$ becoming zero, giving a problem since we must divide by $\mathbf{z}^T$. The fact that $\omega_2^2 = 0$ also brought in more sparsity to the least squares problem.

To overcome this, we generate a random set V_2 of full density. This is reflected in the speed of convergence, the algorithm here needing 56 sweeps to reduce the matching error to zero, as opposed to 11 sweeps and 8 sweeps for Examples 1 and 2 in Sections 6.6.2 and 6.6.4 respectively, with the same weightings.

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	$2.7074e+01$	$1.5105e-01$	$7.1138e+01$
3	$1.2328e+01$	$4.4603e-03$	$7.0878e+01$
5	$1.2037e+01$	$2.8356e-03$	$7.0495e+01$
10	$1.1860e+01$	$2.2599e-03$	$7.0135e+01$
50	$1.1672e+01$	$1.4463e-03$	$6.9812e+01$
100	$1.1598e+01$	$8.8061e-04$	$6.9761e+01$
150	$1.1553e+01$	$5.7105e-04$	$6.9718e+01$

The matching is reduced from $O(10^5)$ to $O(10^{-3})$ within 3 sweeps, but takes another 100 sweeps to reduce it by another order.

6.6.7 Test results

These examples show the increased performance of the unrestricted algorithm over the restricted one. When the left matching is the only consideration, the

error is reduced virtually to zero, in comparison to $O(10^1)$ for the restricted case. However, care needs to be taken over choosing new vectors, as this could force the condition number of the modal matrix, V, to rise dramatically. The inclusion of the conditioning in the algorithm is successful, although it is at the expense of increasing the matching error. To reduce this, a higher weighting is imposed on the left vector matching relative to the conditioning. In all examples, it is seen that the errors are reduced considerably within the first few iterations, but that a lot of additional work is required to improve things further.

6.7 Efficiency comparison

We would like to solve our problem using existing methods to compare the numerical efficiency of our minimisation techniques. To do this we consider the objective function

$$J \equiv \omega_1^2 \|G_{1d} - G_{1a}\|_F^2 + \omega_2^2 \|V^{-1}\|_F^2, \tag{6.67}$$

and minimise J using the Matlab Optimisation Toolbox. If $\omega_2^2 \neq 0$, then we enforce constraints on the new vectors such that $\|\mathbf{v}_i\|_2^2 = 1$ for $\mathbf{v}_i \in V_2$, so the conditioning is being bounded as in (6.11). We use two different starting points for the minimisation, one is the results of partial eigenstructure assignment (as in Section 6.6.1), the other is to use the restricted projection to obtain V_1 and to form V_2 from a QR decomposition of V_1 (as in Section 6.6.3). For both of these we use three different combinations of the weighting parameters and terminate the algorithms when the value of the objective function has reached a certain level. We give comparisons of the number of floating point operations and cpu

time. The results are given in the table below.

starting	parameters		stopping	flops		cpu time (secs.)	
point	ω_1^2	ω_2^2	criterion	Toolbox	ours	Toolbox	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 investigation to increase the efficiency of the minimisation routine by improving the manner in which we control the conditioning of the system.

Also these results are obtained when the vectors are chosen freely; if they were restricted to lie in subspaces corresponding to specified eigenvalues then the differences between the two methods would almost certainly be greater.

6.8 Conclusions

We have, in this chapter, described an analogy to the restricted minimisation presented in Chapter 5. The difference is in the fact that the minimisation vectors are allowed to be chosen totally freely from within the complex plane. Following this, we have given three methods for finding an initial vector set with which to run the algorithm. To illustrate the theory, we have given examples with various parameter values, and have compared the results with those obtained in the restricted case. The conclusions are that the results here are improved in comparison to those in the previous chapter.

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)construction 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 Methods for calculating an initial right vector 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, transform to real vectors	QR decomposition on V_1
6	unrestricted projection	cycle I_n
7	unrestricted projection	QR decomposition on V_1

7.2 Construction of feedback for restricted minimisation

Following the completion of the restricted minimisation algorithm, the results are

$$\tilde{V} = [V_1, \tilde{V}_2], \quad \tilde{\Lambda} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \tilde{\Lambda}_2 \end{bmatrix}, \tag{7.1}$$

where the V_1 set has remained constant throughout the routine, and the $\tilde{V}_2$ have been chosen to minimise set criteria. The Λ_1 set is that used in the restricted projection; the $\tilde{\Lambda}_2$ set results from one of Methods 1-3.

If one of Methods 1-2 is used as the starting point for the algorithm, then the original feedback matrix is, (as in (4.32)),

$$K = B^+(V_1\Lambda_1 - AV_1)(CV_1)^{-1}. \tag{7.2}$$

From this it is evident that if we use (7.2) to reconstruct the feedback after the minimisation, then we just obtain the original K, since V_1 is unchanged throughout the minimisation. The reduction in the objective function is obtained

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

If we use Method 3 to find the starting point for the algorithm, then we still cannot form the feedback as in (7.2). This generates the same K as for Method 1 or 2, so the extra computation performed in using the minimisation is wasted.

We now give two constructions for the new feedback matrix, and give results on their errors in assignment, as well as the relationship between the two.

7.2.1 First (re)construction, K_1

The reason that we cannot use the feedback construction in (7.2) is that this is the construction for partial eigenstructure assignment. The minimisation algorithm generates a full set of vectors and hence we are attempting full eigenstructure assignment. In Section 4.2, we gave a feedback construction that achieves full eigenstructure assignment under certain conditions; our first construction is therefore

$$K_1 = B^+(\tilde{V}\tilde{\Lambda}\tilde{V}^{-1} - A)C^+, \tag{7.3}$$

where $\tilde{V} = [V_1, \tilde{V}_2]$. This construction generates the exact eigenstructure assignment provided that the right and left eigenvectors are simultaneously in their correct subspaces. There is an error involved, given by

$$E_1 \equiv \tilde{W}^T(A + BK_1C) - \tilde{\Lambda}\tilde{W}^T = (\tilde{W}^TA - \tilde{\Lambda}\tilde{W}^T)(I - C^+C), \tag{7.4}$$

where $\tilde{W}^T = \tilde{V}^{-1}$. Using the decomposition of C as in (4.11),

$$\begin{aligned} \mathbf{w}_i \in \mathcal{N}[P_1^T(A^T - \lambda_i I)] &\Rightarrow P_1^T(A^T - \lambda_i I)\mathbf{w}_i = 0 \\ &\Rightarrow P_1P_1^T(A^TW - W\Lambda) = 0 \\ &\Rightarrow (W^TA - \Lambda W^T)P_1P_1^T = 0. \end{aligned} \tag{7.5}$$

But from (4.11), $P_1P_1^T = I - C^+C$, so that

$$\mathbf{w}_i \in \mathcal{N}[P_1^T(A^T - \lambda_i I)] \Rightarrow (W^TA - \Lambda W^T)(I - C^+C) = 0, \tag{7.6}$$

which indicates that $\|E_1\|$ can be reduced by minimising the distance of the left eigenvectors from their correct subspaces. This is the third criterion within the minimisation and can be reduced by putting a high weighting on ω_3^2 relative to ω_1^2 and ω_2^2.

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, $C\tilde{V} \in \mathbb{C}^{p\times n}$ is not invertible unless $p = n$; hence our second construction is

$$K_2 = B^+(\tilde{V}\tilde{\Lambda} - A\tilde{V})(C\tilde{V})^+. \tag{7.7}$$

We again need to know the error involved in using this construction. This error is given by

$$\begin{aligned} E_2 \equiv & \ \tilde{W}^T(A + BK_2C) - \tilde{\Lambda}\tilde{W}^T \\ = & \ (\tilde{W}^TA - \tilde{\Lambda}\tilde{W}^T) + \tilde{W}^TBB^+(\tilde{V}\tilde{\Lambda} - A\tilde{V})(C\tilde{V})^+C. \end{aligned} \tag{7.8}$$

However, the $\mathbf{v}_i$ are in their correct subspaces, so that $BB^+(\tilde{V}\tilde{\Lambda} - A\tilde{\Lambda}) = (\tilde{V}\tilde{\Lambda} - A\tilde{\Lambda})$, as in (4.7). Hence

$$\begin{aligned} E_2 \equiv & \ (\tilde{W}^TA - \tilde{\Lambda}\tilde{W}^T) + (\tilde{\Lambda}\tilde{W}^T - \tilde{W}^TA)\tilde{V}(C\tilde{V})^+(C\tilde{V})\tilde{W}^T \\ = & \ (\tilde{W}^TA - \tilde{\Lambda}\tilde{W}^T)(I - \tilde{V}(C\tilde{V})^+(C\tilde{V})\tilde{W}^T). \end{aligned} \tag{7.9}$$

This error is similar to that given in (7.4), but a direct link is not obvious. We want to know how minimising $\|E_1\|$ affects $\|E_2\|$, and we explore this in the next section.

7.2.3 Relationship between K_1 and K_2 and their respective assignment errors

To find the link between the errors in assignment between the two constructions for the feedback, let

$$\begin{cases} M \equiv (\tilde{W}^TA - \tilde{\Lambda}\tilde{W}^T) \\ X \equiv \tilde{V}(C\tilde{V})^+(C\tilde{V})\tilde{W}^T \end{cases}. \tag{7.10}$$

Then $E_1X = M(I - C^+C)X = M(X - C^+CX) = M(X - C^+C)$ since

$$CX = (C\tilde{V})(C\tilde{V})^+(C\tilde{V})\tilde{W}^T = (C\tilde{V})\tilde{W}^T = C, \tag{7.11}$$

by the first Moore-Penrose condition for $C\tilde{V}$ given in Definition 4.11. Now

$$\begin{aligned} E_1 - E_1X = & \ M(I - C^+C) - M(X - C^+C) \\ = & \ M(I - X) \equiv E_2, \end{aligned} \tag{7.12}$$

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

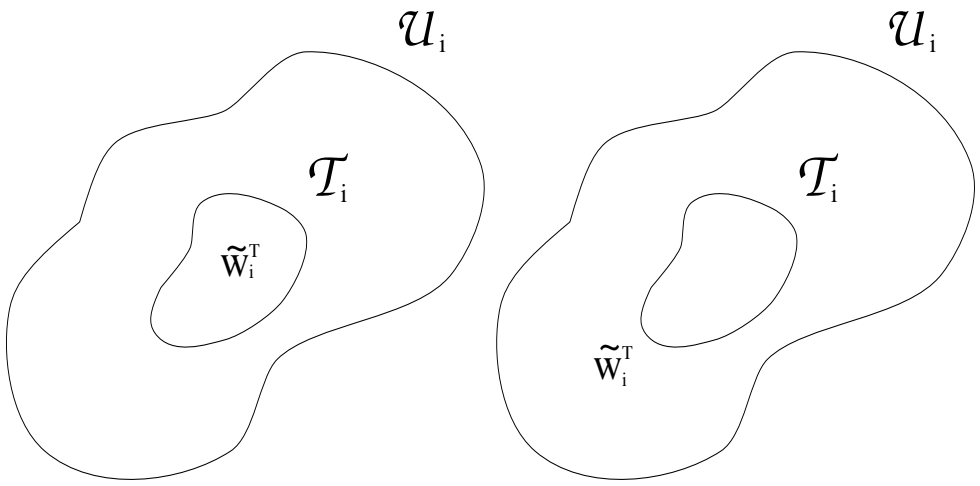

Figure 7.1: nullspaces

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

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.

$$\begin{array}{ll} (i) \quad CC^{+}C = C, & (iii) \quad (CC^{+})^{H} = CC^{+} \\ (ii) \quad C^{+}CC^{+} = C^{+}, & (iv) \quad (C^{+}C)^{H} = C^{+}C. \end{array} \tag{7.17}$$

We must show that $V^{-1}C^{+}$ satisfies these four conditions, so

$$(i)(CV)(CV)^{+}(CV) = CVV^{-1}C^{+}CV = CC^{+}CV = CV,$$

$$\begin{aligned} (ii)\ (CV)^{+}(CV)(CV)^{+} &= V^{-1}C^{+}CVV^{-1}C^{+} = V^{-1}C^{+}CC^{+} \\ &= V^{-1}C^{+} = (CV)^{+}, \end{aligned}$$

$$\begin{aligned} (iii)\ [(CV)(CV)^{+}]^{H} &= [CVV^{-1}C^{+}]^{H} = (CC^{+})^{H} = CC^{+} = CVV^{-1}C^{+} \\ &= (CV)(CV)^{+}, \end{aligned}$$

$$\begin{aligned} (iv)\ [(CV)^{+}(CV)]^{H} &= [V^{-1}C^{+}CV]^{H} = V^{H}(C^{+}C)^{H}(V^{H})^{-1} = V^{H}C^{+}C(V^{H})^{-1} \\ &= V^{-1}C^{+}CV = (CV)^{+}(CV), \end{aligned}$$

if V is unitary, which gives the result.

LEMMA 7.2 *The feedback constructions, K_1 and K_2, given respectively in (7.3) and (7.7), are equivalent in the sense that they generate the same eigenstructure if*

$$\begin{array}{ll} & (i) \quad \tilde{V} \textit{ is unitary}, \\ \textit{or} & (ii) \quad \tilde{\mathbf{w}}_i^{T} \in \mathcal{T}_i \quad (\forall i = 1, \ldots, n). \end{array} \tag{7.18}$$

Proof

$$
\begin{aligned}
(i) \qquad K_2 &\equiv B^+(\tilde{V}\tilde{\Lambda} - A\tilde{V})(C\tilde{V})^+ \\
&= B^+(\tilde{V}\tilde{\Lambda} - A\tilde{V})\tilde{V}^{-1}C^+ \quad \text{(from Theorem 7.1)} \\
&= B^+(\tilde{V}\tilde{\Lambda}\tilde{V}^{-1} - A)C^+ \equiv K_1.
\end{aligned} \tag{7.19}
$$

(ii) from (7.6),

$$
\tilde{\mathbf{w}}_i^T \in \mathcal{T}_i \;\; (\forall i = 1, \ldots, n) \Rightarrow \|E_1\|_F^2 = 0 \Rightarrow \|E_2\|_F^2 = 0
$$

so that both K_1 and K_2 generate the same closed loop eigenstructure; that is they are equivalent.

7.2.4 Effect of scaling on reconstruction of feedback

In Sections 5.9 and 5.10, we gave two methods for scaling V_1. If this scaling is performed, we want to know the affect on the construction of the feedback. To develop the scaling theory, the actual scaling was put on the left vectors and inverted, giving

$$
V_{\text{new}} = VD^{-1}. \tag{7.20}
$$

Thus, if we replace $\tilde{V}$ by $\tilde{V}D^{-1}$ in the expressions for the feedback construction, and add a subscript D to denote scaling, then

$$
K_{1D} = B^+(\tilde{V}D^{-1}\tilde{\Lambda}D\tilde{V}^{-1} - A)C^+. \tag{7.21}
$$

Here D and $\tilde{\Lambda}$ are diagonal so that $D^{-1}\tilde{\Lambda}D = D^{-1}D\tilde{\Lambda} = \tilde{\Lambda}$, giving

$$
K_{1D} = B^+(\tilde{V}\tilde{\Lambda}\tilde{V}^{-1} - A)C^+ \equiv K_1, \tag{7.22}
$$

so that the first construction of the feedback is independent of the scaling imposed on V_1. For the second construction

$$
\begin{aligned}
K_{2D} &= B^+(\tilde{V}D\tilde{\Lambda} - A\tilde{V}D)(C\tilde{V}D)^+ \\
&= B^+(\tilde{V}D\tilde{\Lambda}D^{-1} - A\tilde{V})D(C\tilde{V}D)^+.
\end{aligned} \tag{7.23}
$$

Again $D^{-1}\tilde{\Lambda}D = \tilde{\Lambda}$; if Theorem 7.1 holds then

$$
(C\tilde{V}D)^+ = D^{-1}\tilde{V}^{-1}C^+ \Rightarrow D(C\tilde{V}D)^+ = (C\tilde{V})^+, \tag{7.24}
$$

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 have expressed K_2 as the original feedback plus expressions involving only the updated set $(\tilde{V}_2, \tilde{\Lambda}_2)$ and the constant system matrices (A, B, C).

Now $(C\tilde{V}_2) \in \mathbb{C}^{p \times n-p}$ so that if rank $(C\tilde{V}_2) = n - p$ then $(C\tilde{V}_2)(C\tilde{V}_2)^+ = I$. That is

$$p \leq n - p \quad \Rightarrow \quad (C\tilde{V}_2)(C\tilde{V}_2)^+ = I, \tag{7.30}$$

giving

$$2p \leq n \quad \Rightarrow \quad (C\tilde{V}_2)(C\tilde{V}_2)^+ = I. \tag{7.31}$$

This has the implication that

$$K_2 = 2[K + B^+(\tilde{V}_2\tilde{\Lambda}_2 - A\tilde{V}_2)(C\tilde{V}_2)^+]. \tag{7.32}$$

From this section we can see that this construction for K_2 is, in essence, a perturbation to the feedback produced by Methods 1 and 2; achieved by selecting a new set, $\tilde{V}_2$.

7.2.6 Summary

When the minimisation algorithm is run with the new vectors restricted to be in certain spaces, the result is a set of real vectors that best satisfy some criteria, and the eigenvalues that they correspond to. A feedback must be (re)constructed; we have given two methods to do this. We have shown the errors in assignment for both and shown how we expect the minimisation algorithm to reduce both of these errors.

We now give a method for constructing a feedback using the results of the unrestricted minimisation.

7.3 Construction of feedback for unrestricted minimisation

We have described two methods for (re)constructing the feedback after running the restricted minimisation algorithm. The problem now is that the unrestricted minimisation algorithm generates a set of real vectors, but these do not correspond to any specified eigenvalues. Thus, we cannot construct the feedback by either of the methods in the previous section, since we do not have a Λ. We therefore develop a method for constructing a feedback from the minimisation vectors only.

Note that as a comparison, this method can also be applied to the restricted minimisation.

7.3.1 Diagonal solver

The objective here is to find a feedback matrix, K_3, that best preserves the $\tilde{V}$ generated by the unrestricted minimisation algorithm. Since we have no specified eigenvalues, we also want to calculate the set $\tilde{\Lambda}$. The problem is defined as:

Problem 5 Given the quadruple $(A, B, C, \tilde{V})$ where A, B C are real and $\tilde{V}$ is a self-conjugate set, calculate a real K_3 and $\tilde{\Lambda}$ such that

$$\begin{cases} (A + BK_3C)\tilde{V} = \tilde{V}\tilde{\Lambda} \\ \text{subject to } \tilde{\Lambda} \text{ diagonal} \\ \text{Re}(\tilde{\lambda}_j) < 0, \end{cases} \tag{7.33}$$

where the method developed is named the diagonal solver because of the attempt to solve the system subject to $\tilde{\Lambda}$ diagonal. In its present matrix algebraic equation form (7.33) cannot be solved directly as it is an underdetermined system. To manipulate it into a solvable form, re-write the top line of (7.33) as

$$\tilde{V}^{-1}(A + BK_3C)\tilde{V} = \tilde{\Lambda}, \tag{7.34}$$

where $\tilde{V}$ is inverted using the QR decomposition so as to avoid any ill-conditioning problems. Write (7.34) as

$$-\tilde{B}K_3\tilde{C} + \tilde{\Lambda} = \tilde{A}, \tag{7.35}$$

where

$$\begin{cases} \tilde{A} = \tilde{V}^{-1}A\tilde{V} \\ \tilde{B} = \tilde{V}^{-1}B \\ \tilde{C} = C\tilde{V} \end{cases} . \tag{7.36}$$

The aim is to express (7.35) by its individual equations. Denote the elements of $(\tilde{A}, \tilde{B}, \tilde{C}, K_3, \tilde{\Lambda})$ by $(a_{ij}, b_{ij}, c_{ij}, k_{ij}, \tilde{\lambda}_{ij})$ $(i, j = 1, \ldots, n)$, respectively; then we wish to solve

$$\begin{cases} -\sum_{t=1}^{m}\sum_{s=1}^{p} b_{it}k_{ts}c_{sj} + \tilde{\lambda}_{ij} = a_{ij} \\ \text{subject to} \begin{cases} \tilde{\lambda}_{ij} = 0 \quad (i \neq j) \\ \text{Re}(\tilde{\lambda}_{ij}) < 0 \quad (i = j) \end{cases} \end{cases} \qquad (i, j = 1, \ldots, n). \tag{7.37}$$

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 $n > m, p$ then*

$$n^2 > mp + n. \tag{7.38}$$

Proof Since m, p are integers, both less than n, their maximum values are attained at $m = p = n - 1$. Then

$$\max(mp + n) = (n-1)(n-1) + n = n^2 - n + 1. \tag{7.39}$$

Now, $n \geq 2$ (if $n = 1$, then $m = p = 0$, contradicting their inclusion in the set of natural numbers) so that $1 - n < 0$. Hence

$$\max(mp + n) = n^2 + (1 - n) < n^2 \Box. \tag{7.40}$$

Our system, (7.37), has n^2 equations ($\equiv$ number of elements in $\tilde{A}$) and mp unknowns from K_3 plus n unknowns from $\tilde{\Lambda}$. Thus, we have a system with n^2 equations and $mp + n$ unknowns which, from Theorem 7.3, is an overdetermined system.

It is not known whether or not the diagonal elements of $\tilde{\Lambda}$ will be real or complex. If complex, they may appear explicitly on the diagonal of $\tilde{\Lambda}$, or as 2×2 real blocks. The development of (7.37) into a linear least squares form is different for each case, and is covered in the next two sections.

7.3.2 Complex solver formulation

Following the unrestricted minimisation, we have a new set of self-conjugate vectors, $\tilde{V}$. From these we form the system, as in (7.37). In Section 6.6, we described three methods for selecting an initial set of vectors to be used as a starting point for the unrestricted algorithm. Two of these involved transforming complex vectors to their corresponding real representation. So, although we have a set of real vectors to use to find $(K_3, \tilde{\Lambda})$, we cannot expect $\mathrm{diag}(\tilde{\Lambda})$ to be real. Indeed, some pairs of the real vectors may represent the eigenvectors of a complex conjugate pair of eigenvalues. The question here is how to allow for these complex eigenvalues.

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 $M = [M_1, M_2]$. Here $M_1 \in \mathbb{C}^{n^2 \times mp}$ contains the coefficients $b_{it}c_{sj}$ and $M_2 \in \mathbb{C}^{n^2 \times n}$ contains the coefficients $\tilde{\lambda}_{ij}$ such that

$$M_1 = \begin{bmatrix} b_{11}c_{11} \ldots b_{11}c_{p1}, & \ldots & , b_{1m}c_{11} \ldots b_{1m}c_{p1} \\ \vdots & \vdots & \vdots \\ b_{11}c_{1n} \ldots b_{11}c_{pn}, & \ldots & , b_{1m}c_{1n} \ldots b_{1m}c_{pn} \\ \vdots & \vdots & \vdots \\ b_{21}c_{11} \ldots b_{21}c_{p1}, & \ldots & , b_{2m}c_{11} \ldots b_{2m}c_{p1} \\ \vdots & \vdots & \vdots \\ b_{n1}c_{1n} \ldots b_{n1}c_{pn}, & \ldots & , b_{nm}c_{1n} \ldots b_{nm}c_{pn} \end{bmatrix}, \tag{7.43}$$

$$M_2 = \begin{bmatrix} 1\,0\,0 & \ldots\ldots & 0\,0\,0 \\ 0\,0\,0 & \ldots\ldots & 0\,0\,0 \\ \vdots & \vdots & \vdots \\ 0\,0\,0 & \ldots\ldots & 0\,0\,0 \\ 0\,1\,0 & \ldots\ldots & 0\,0\,0 \\ 0\,0\,0 & \ldots\ldots & 0\,0\,0 \\ \vdots & \vdots & \vdots \\ 0\,0\,0 & \ldots\ldots & 0\,0\,0 \\ \vdots & \vdots & \vdots \\ 0\,0\,0 & \ldots\ldots & 0\,1\,0 \\ 0\,0\,0 & \ldots\ldots & 0\,0\,0 \\ \vdots & \vdots & \vdots \\ 0\,0\,0 & \ldots\ldots & 0\,0\,0 \\ 0\,0\,0 & \ldots\ldots & 0\,0\,1 \end{bmatrix} \begin{matrix} \} \text{ first row of } I_n \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n \text{ rows of zeros} \\ \} \text{ second rows of } I_n \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} n \text{ rows of zeros} \\ \\ \\ \\ \} \text{ penultimate row of } I_n \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} n \text{ rows of zeros} \\ \} \text{ last row of } I_n \end{matrix} . \tag{7.44}$$

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

$$\tilde{\lambda}_j = \alpha_j \pm \mathbf{i}\beta_j, \tag{7.47}$$

then it may appear on the diagonal of $\tilde{\Lambda}$ as

$$\begin{bmatrix} \alpha_j & \beta_j \\ -\beta_j & \alpha_j \end{bmatrix}. \tag{7.48}$$

However, we are not sure where we should allow the 2×2 blocks corresponding to the vectors in $\tilde{V}_2$. To develop the theory though, we consider the most general case whereby the system is of an even order and all of the eigenvalues are in complex conjugate pairs. We are still solving Problem 5, but here it is reduced

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

$$\tilde{\Lambda} = \left[\begin{array}{cc|c|cc|cc} \alpha_1 & \beta_1 & & & & & \\ -\beta_1 & \alpha_1 & & & & & \\ \hline & & \ddots & & & & \\ & & & \ddots & & & \\ \hline & & & & \alpha_j & \beta_j & \\ & & & & -\beta_j & \alpha_j & \\ \hline & & & & & \ddots & \\ & & & & & & \ddots \\ \hline & & & & & & \alpha_{\frac{n}{2}} \quad \beta_{\frac{n}{2}} \\ & & & & & & -\beta_{\frac{n}{2}} \quad \alpha_{\frac{n}{2}} \end{array}\right], \tag{7.50}$$

where all of the block off-diagonal elements are zero.

Again we augment the system as in (7.42), where M_1 remains the same. But

M_2 changes because of the different form that we are considering. We have

$$M_2 = \left[\begin{array}{ccc}
1\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,1\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,1\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,1\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
\vdots & \vdots & \vdots \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,1\,0 \\
0\,0\,0\,0\,0 & \ldots\ldots & 0\,0\,1
\end{array}\right]
\begin{array}{l}
\left.\begin{array}{l}\\\\\end{array}\right\} \text{first 2 rows of } I_{2n} \\
\left.\begin{array}{l}\\\\\end{array}\right\} \frac{n}{2} \text{ rows of zeros} \\
\left.\begin{array}{l}\\\\\end{array}\right\} \text{next 2 rows of } I_{2n} \\
\left.\begin{array}{l}\\\\\\\\\end{array}\right\} n \text{ rows of zeros} \\
\\
\left.\begin{array}{l}\\\\\end{array}\right\} \frac{n}{2} \text{ rows of zeros} \\
\left.\begin{array}{l}\\\\\end{array}\right\} \text{last 2 rows of } I_{2n}
\end{array} . \tag{7.51}$$

For each complex conjugate pair, there are four $\tilde{\lambda}_{ij}$'s. For $\tilde{\Lambda} \in \mathbb{R}^{n \times n}$ there are $\frac{n}{2}$ complex conjugate pairs, hence there are $\frac{n}{2}$ (2×2) blocks. Each block has four elements, hence M_2 has $4(\frac{n}{2}) = 2n$ columns; then $M_2 \in \mathbb{R}^{n^2 \times 2n}$. The form of M_2 is two rows that represent $\tilde{\lambda}_{i,j}$, $\tilde{\lambda}_{i,j+1}$; then $\frac{n}{2}$ rows of zeros before two rows that represent $\tilde{\lambda}_{i+1,1}$, $\tilde{\lambda}_{i+1,j+1}$; then n rows of zeros before the next 2×2 block. Thus, M_2 is two consecutive rows of the $2n$-dimensional identity matrix, spaced alternatively by rows of zeros of dimension $\frac{n}{2} \times 2n$ and $n \times 2n$, respectively.

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 Algorithm 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 algorithm is thus

1. initialise
$$\begin{cases} \tilde{A} = \tilde{V}^{-1} A \tilde{V} \\ \tilde{B} = \tilde{V}^{-1} B \\ \tilde{C} = C \tilde{V} \end{cases} \tag{7.54}$$

2. form M_1 from $\tilde{B}$ and $\tilde{C}$

3. form M_2 from $\tilde{\Lambda}$ according to how we want the eigenvalues to appear on the diagonal

4. form $\mathbf{a}$ from $\tilde{A}$

5. to find $\mathbf{k}$ solve
$$\min \|M\mathbf{k} - \mathbf{a}\|_2^2 \tag{7.55}$$

6. from $\mathbf{k}$ form K_3 and $\tilde{\Lambda}$

7. calculate errors
$$\begin{aligned} &\|M\mathbf{k} - \mathbf{a}\|_2^2 \\ &\|(A + BK_3C)\tilde{V} - \tilde{V}\tilde{\Lambda}\|_F^2. \end{aligned} \tag{7.56}$$

It is obvious that, since the system is overdetermined, the solution vector, $\mathbf{k}$, will solve $\|M\mathbf{k} - \mathbf{a}\|_2^2$ in the best least squares sense. Thus, we do not expect to have solved Problem 5, as in (7.33), exactly. If this is the case, then $\tilde{V}^{-1}(A + BK_3C)\tilde{V}$ has off-diagonal elements. Hence the $\tilde{\lambda}_{ii}$ found for the diagonal of $\tilde{\Lambda}$ will not be the final, closed loop eigenvalues of $A + BK_3C$. In fact, the desired $(\tilde{V}, \tilde{\Lambda})$ will not be the actual eigenstructure of $A + BK_3C$, but we have minimised the error between the two in a certain sense. These errors are analysed in the next section.

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

$$\begin{cases} \min\limits_{K_3,\tilde{\Lambda}} \|(A+BK_3C)\tilde{V}-\tilde{V}\tilde{\Lambda}\|_F^2 \\ \text{subject to } \tilde{\Lambda} \text{ (pseudo)-diagonal} \\ \text{and } K_3 \text{ real} \end{cases}, \tag{7.57}$$

where $\tilde{\Lambda}$ (pseudo)-diagonal means that it may be block-diagonal, depending on what form of the diagonal solver is used. However, (7.57) is only solved exactly if K_3 can be found such that the given $\tilde{V}$ are eigenvectors of the closed loop system, a situation that is unlikely to occur when using the unrestricted minimisation. This means that the values that appear on the (block)-diagonal of $\tilde{\Lambda}$ will not be the actual eigenvalues of $A+BK_3C$, but we can establish a relationship between them. First, a theorem is needed.

THEOREM 7.4 *(Bauer-Fike) If μ is an eigenvalue of $A+E \in \mathbb{C}^{n\times n}$ and $V^{-1}AV = D = diag(\lambda_1,\ldots,\lambda_n)$, then*

$$\min_{\lambda\in\lambda(A)} |\lambda-\mu| \leq \kappa_p(V)\|E\|_p, \tag{7.58}$$

where $\|\cdot\|_p$ denotes any of the Holder norms.

Proof (see Golub and Van Loan [24]).

To use this theorem, we need to manipulate our system slightly. If (7.57) is not solved exactly, the error is given by

$$E = (A+BK_3C) - \tilde{V}\tilde{\Lambda}\tilde{V}^{-1}, \tag{7.59}$$

so that

$$\tilde{V}^{-1}(A_c-E)\tilde{V} = \tilde{\Lambda}, \tag{7.60}$$

where $A_c \equiv A+BK_3C$ is the closed loop matrix. Hence $(\tilde{V},\tilde{\Lambda})$ is not the eigenstructure for A_c, as desired, but is the eigenstructure for a perturbed system, A_c-E.

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*

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

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

$A + BK_3C$, namely λ_{ij}, that are unstable. Lemma 7.5 shows how the $\tilde{\lambda}_{ij}$ and λ_{ij} are related; hence we can put constraints on the $\tilde{\lambda}_{ij}$ to induce constraints on the λ_{ij}. This means that we can now treat the last condition in Problem 5, that is to obtain $\mathrm{Re}(\lambda_j) < 0$.

We do this by implementing the NAG Fortran Library Routine E04NCF which solves linearly constrained linear-least squares problems and convex quadratic programming problems. To proceed, we form the system as in (7.62). In solving this, we place constraints on the $\tilde{\lambda}_{ij}$ that are either real values, or that correspond to the real part of a complex eigenvalue. The constraints we impose are such that

$$\mathrm{Re}(\tilde{\lambda}_{jj}) \leq \gamma < 0 \quad \Rightarrow \quad \mathrm{Re}(\lambda_j) < 0 \quad (\forall j), \tag{7.63}$$

where γ is the minimum distance of the most unstable eigenvalue in $\{\tilde{\lambda}_{jj}\}$ to the stable set $\{\lambda_j\}$. It is unclear how to choose γ because, from Lemma 7.5, we would ideally set

$$\gamma = \kappa_F(\tilde{V})\|E\|_F, \tag{7.64}$$

but we do not have prior knowledge of E, since it is dependent on K_3. This is done by experimentation and is illustrated in the examples in Chapter 8. We may also make use of the NAG routine in another way, relating to the 2×2 blocks that represent complex eigenvalues for the real solver. As explained in Section 7.3.3, the j^{th} complex conjugate pair is formulated to appear as

$$\begin{bmatrix} \alpha_j & \beta_j \\ -\beta_j & \alpha_j \end{bmatrix}. \tag{7.65}$$

Here the α_j are constrained so that $\alpha_j \leq \gamma$, as they represents the real part. The unconstrained solver may generate a solution such that for

$$\begin{bmatrix} \alpha_{j1} & \beta_{j1} \\ -\beta_{j2} & \alpha_{j2} \end{bmatrix}, \tag{7.66}$$

we obtain $\alpha_{j1} \neq \alpha_{j2}$, $\beta_{j1} \neq -\beta_{j2}$, which is not as required. Thus, we use the NAG routine to form a set of linear constraints such that

$$\begin{aligned} \alpha_{j1} - \alpha_{j2} &= 0 \\ \beta_{j1} + \beta_{j2} &= 0 \end{aligned}. \tag{7.67}$$

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

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

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

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

Full examples

8.1 Introduction

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

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

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

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

where the sensitivity of each eigenvalue, c_i, is calculated using

$$c_i = \frac{\|\mathbf{w}_i^T\| \|\mathbf{v}_i\|}{|\mathbf{w}_i^T \mathbf{v}_i|}. \tag{8.5}$$

The value of the conditioning of the right eigenvectors is

$$\kappa_F(V) = 17.85. \tag{8.6}$$

We next perform partial eigenstructure assignment as in Section 4.3.7 to see the error in the matching of the unconsidered left eigenvectors.

8.3 Partial eigenstructure assignment

For this system, rank $(C) = 4$, so that, according to the theory of Davison [11], we can assign (almost exactly), four closed loop eigenvalues. These are chosen as

$$\Lambda_p = \begin{cases} -6 \pm i & \text{dutch roll mode} \\ -1 \pm 2i & \text{roll mode,} \end{cases} \tag{8.7}$$

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 without feedforward for completeness and to see which inputs are coupled to which outputs, not to see the level of coupling, we obtain this information by analysing the mode input coupling vectors.

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 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 (in their normalised form) as

$$G_{1a} = \begin{bmatrix} 1 & -0.0066 - 0.0041i \\ 1 & -0.0066 + 0.0041i \\ -0.3693 + 0.0363i & 1 \\ -0.3693 - 0.0363i & 1 \end{bmatrix}. \tag{8.13}$$

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.

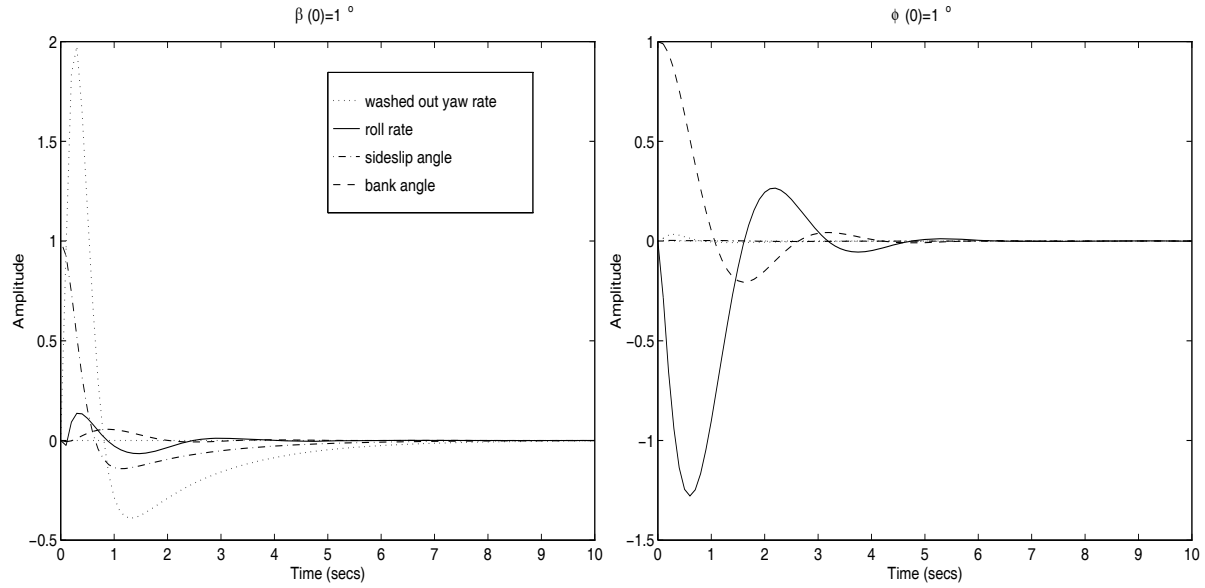

Figure 8.1: Original closed loop output responses

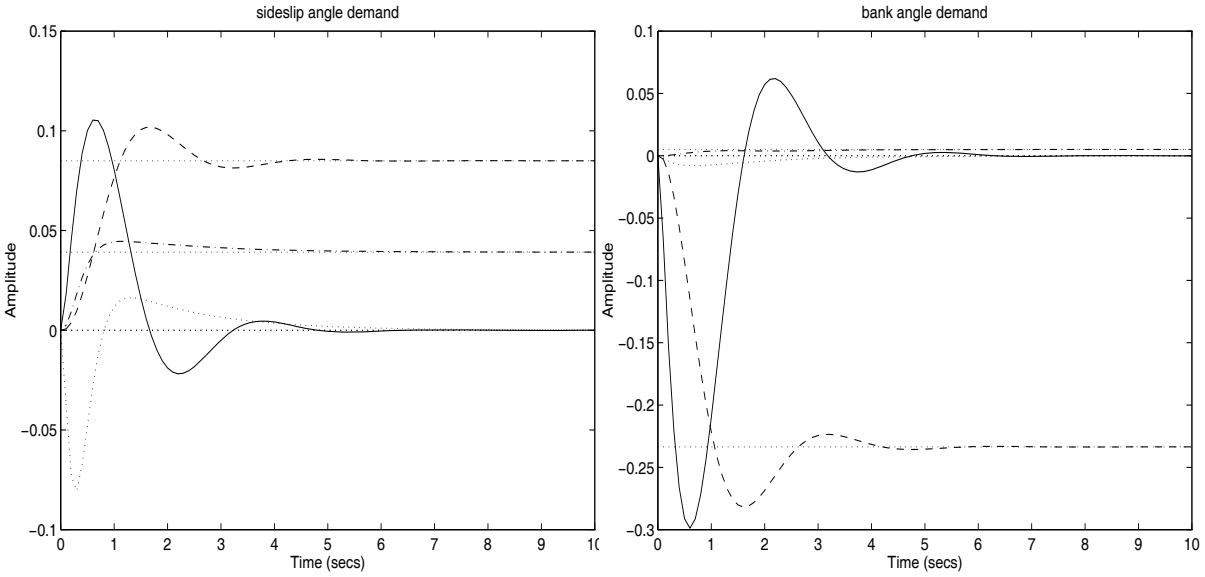

Figure 8.2: Original closed loop input responses

From Figure 8.1 we can see that sideslip angle is virtually decoupled from roll rate and that bank angle is decoupled from the washed out yaw rate. In Figure 8.2 we see that bank angle demand is decoupled from yaw rate and sideslip angle, as desired, but that sideslip angle demand is coupled to roll rate and bank angle. This is the error that we aim to reduce in using our minimisation routine.

8.3.1 Apply 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, $\tilde{V} = [V_1, \tilde{V}_2]$, we have

$$G_{1a} = \begin{bmatrix} 1 & -0.0787 - 0.0922i \\ 1 & -0.0787 + 0.0922i \\ -0.0489 - 0.0276i & 1 \\ -0.0489 + 0.0276i & 1 \end{bmatrix} \tag{8.16}$$

giving the desired level of input decoupling whilst retaining the right vector matrix. However, although we have the desired input and output decoupling for a set of vectors, these do not, as yet, represent a specific eigenstructure corresponding to a feedback, K. We must now reconstruct the feedback to attempt to best achieve $\tilde{V}$ as closed loop vectors. Using the first reconstruction from Chapter 7, we obtain

$$K_1 = \begin{bmatrix} 12.7270 & -0.4798 & -56.7815 & -1.2742 \\ -1.5221 & 0.4292 & -1.4384 & 1.4316 \end{bmatrix}, \tag{8.17}$$

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

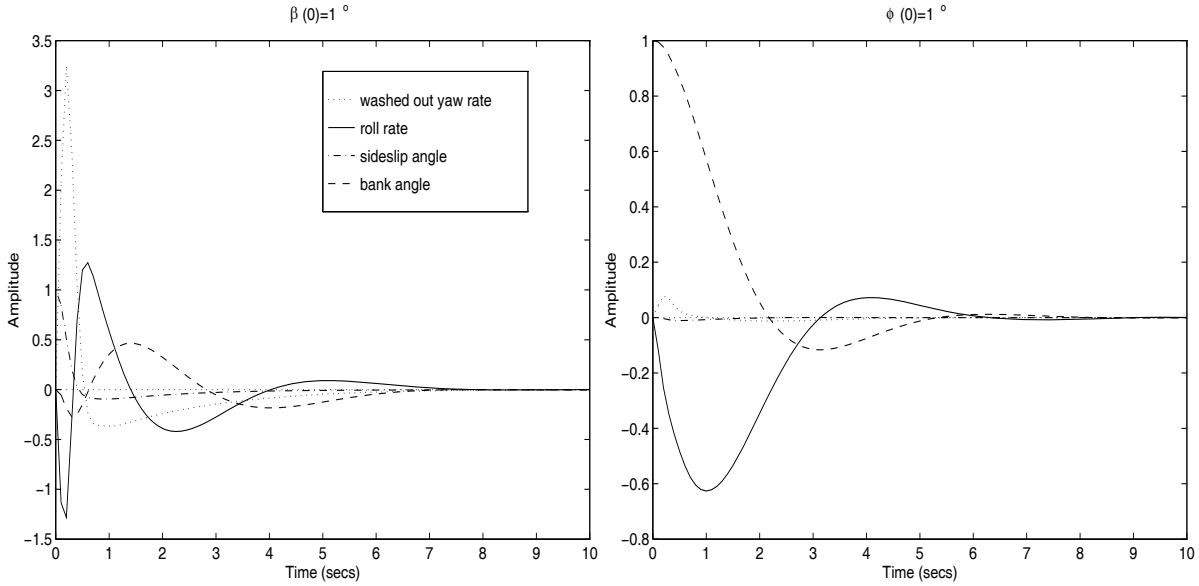

Figure 8.3: New closed loop output responses

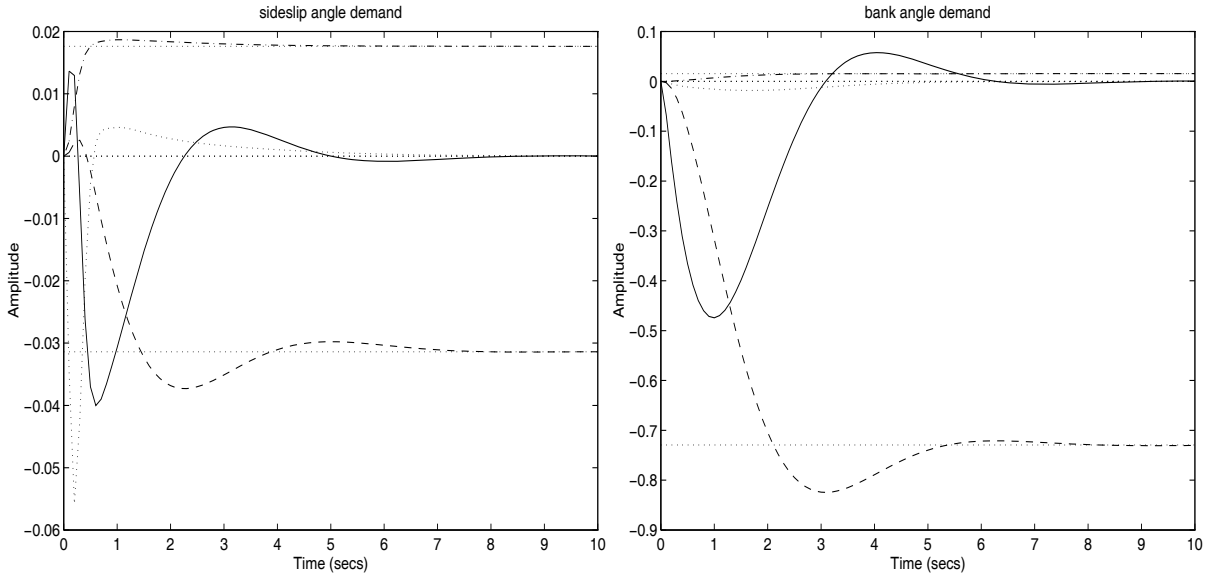

Figure 8.4: New closed loop input responses

the sideslip angle and the roll rate as indicated by G_{0a}. But G_{1a} shows that we have reduced the level of input coupling. We have also retained the desired levels of input and output decoupling on bank angle. This has all been achieved in

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 Apply 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

$$G_{1a} = \begin{bmatrix} 1 & -0.0033 - 0.0487i \\ 1 & -0.0033 + 0.0487i \\ 0.0065 + 0.0872i & 1 \\ 0.0065 - 0.0872i & 1 \end{bmatrix}, \tag{8.24}$$

$$G_{0a} = \begin{bmatrix} 1 & 0.0029 \mp 0.0283i \\ -0.8797 \mp 0.7418i & 1 \\ 0.0295 \pm 0.1395i & 0.0027 \pm 0.0103i \\ -0.0653 \pm 0.1616i & -0.4764 \pm 0.6890i \end{bmatrix}, \tag{8.25}$$

respectively. The level of input decoupling is good for both inputs, but we have introduced substantial coupling between the sideslip angle and roll rate outputs. The new closed loop output and input responses are given in Figures 8.5 and 8.6, respectively.

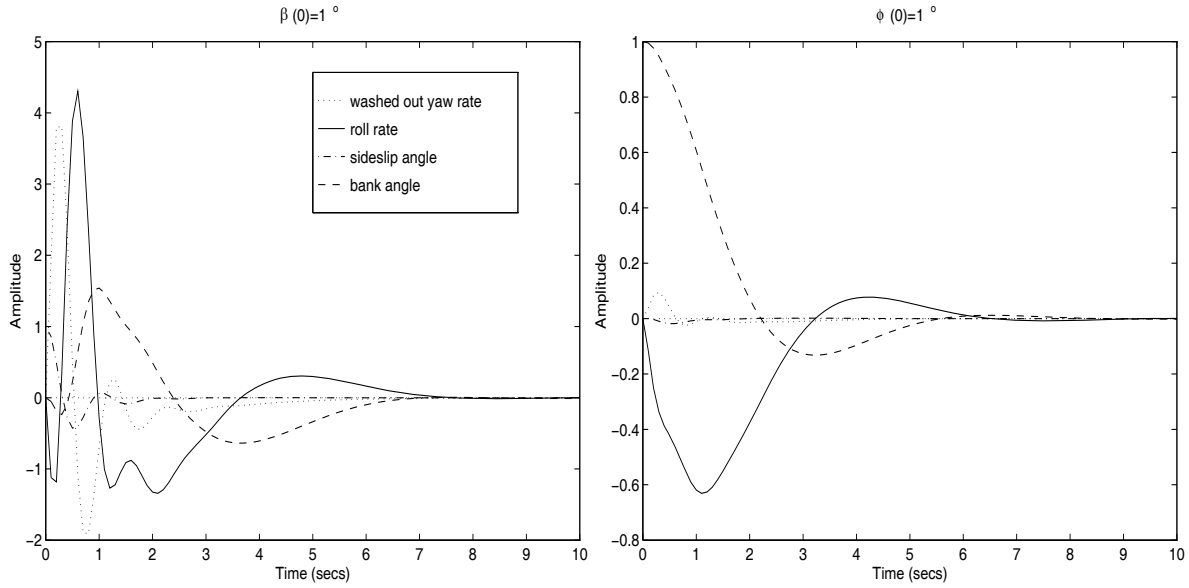

Figure 8.5: New closed loop output responses

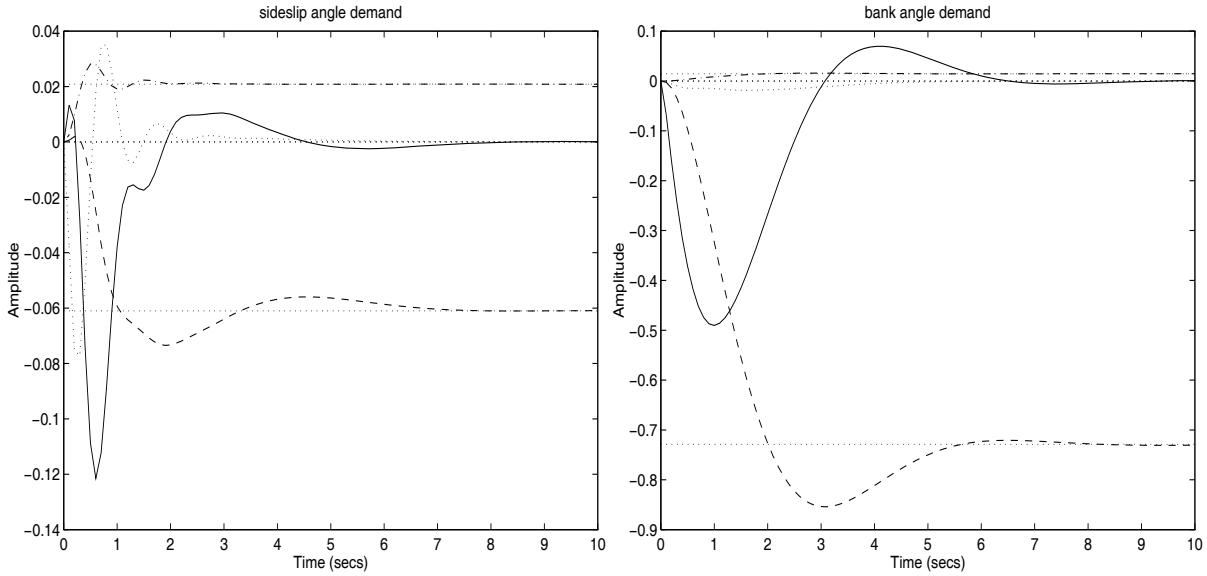

Figure 8.6: New closed loop input responses

In Figure 8.5 we can see the high level of coupling between sideslip angle and roll rate; there is also slight coupling apparent between bank angle and washed out yaw rate. There is only minor improvement in the level of input decoupling; this is expected since we are only attempting to improve the robustness of the system.

8.3.3 Apply unrestricted minimisation algorithm (for decoupling)

In the previous two sections, we have successfully applied the restricted minimisation algorithm to reduce the level of input decoupling and the sensitivity of the system. Here we use the unrestricted minimisation algorithm on the problem.

As described in Section 6.6.1, we require an initial set of *real* vectors for the algorithm. Here we find the initial V using Method 5, where the V_1 vector set is found from a restricted projection on the vectors corresponding to a specified set of eigenvalues (i.e. those in (8.7)); the V_2 vector set is found by performing a QR decomposition on the V_1 set. The chosen weightings are

$$(\omega_1^2, \omega_2^2) = (100, 1), \tag{8.26}$$

so that we are attempting to match the left eigenvectors, but are also keeping a check on the conditioning. The minimisation algorithm generates the results

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$3.0543e+05$	$3.0547e+01$	$1.2155e+02$
1	$1.4890e+03$	$1.0531e+01$	$1.2682e+03$
2	$9.7196e+02$	$7.9887e+00$	$7.9920e+02$
3	$3.0736e+02$	$1.3436e+00$	$7.9897e+02$
4	$2.4754e+02$	$7.4552e-01$	$7.9894e+02$
5	$2.1541e+02$	$2.6014e-01$	$8.3598e+02$

We use the diagonal solver on the new vectors, allowing for two complex conjugate modes in Λ_1, and none in Λ_2. This gives the feedback

$$K_3 = \begin{bmatrix} 6.1910 & 0.0071 & -7.6315 & 0.1900 \\ 2.1815 & 1.0072 & -6.1302 & 4.3681 \end{bmatrix}, \tag{8.27}$$

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

$$\kappa_F(V) = 179.5708. \tag{8.28}$$

The corresponding mode input and output coupling vectors are

$$G_{1a} = \begin{bmatrix} 1 & -0.1716 - 0.0395i \\ 1 & -0.1716 + 0.0395i \\ -0.3163 - 0.1039i & 1 \\ -0.3163 + 0.1039i & 1 \end{bmatrix}, \tag{8.29}$$

$$G_{0a} = \begin{bmatrix} 1 & -0.0223 \pm 0.0549i \\ 0.1721 \mp 0.0334i & 1.0000 \\ 0.5854 \pm 0.0334i & -0.0280 \pm 0.0108i \\ -0.1777 \mp 0.0235i & -0.1963 \mp 0.4010i \end{bmatrix}, \tag{8.30}$$

respectively. Compared to the results of the original output feedback in Section 8.3 there is a slight improvement in the level of decoupling on the sideslip angle input, although a small level of coupling is introduced into the bank angle input. There are also small levels of coupling apparent in the outputs corresponding to the first mode. The new closed loop output and input responses are given in Figures 8.7 and 8.8, respectively.

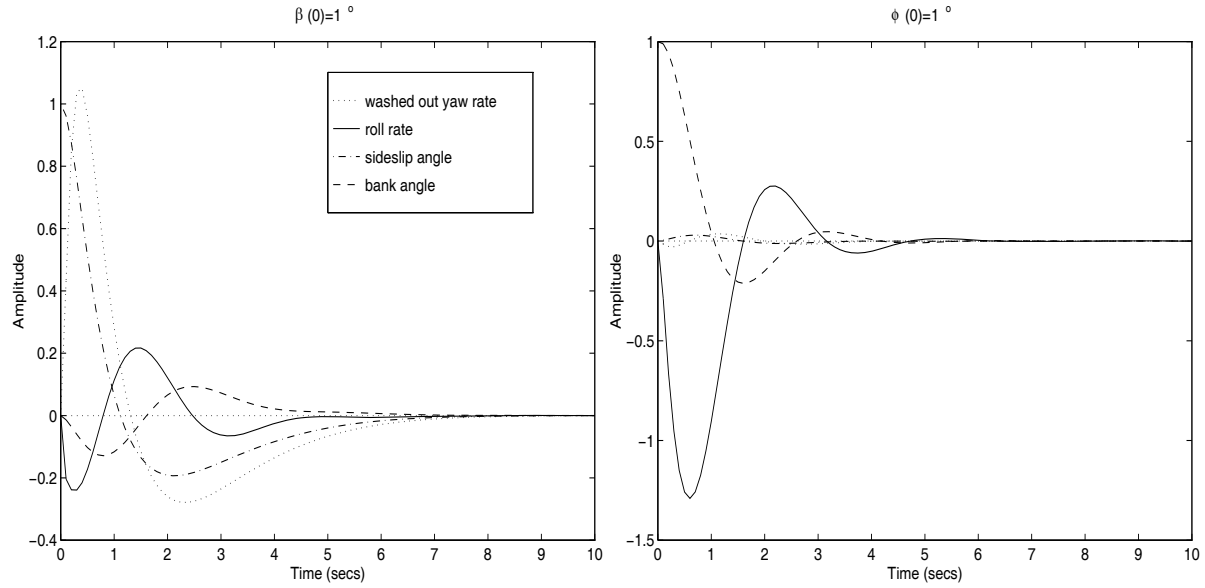

Figure 8.7: New closed loop output responses

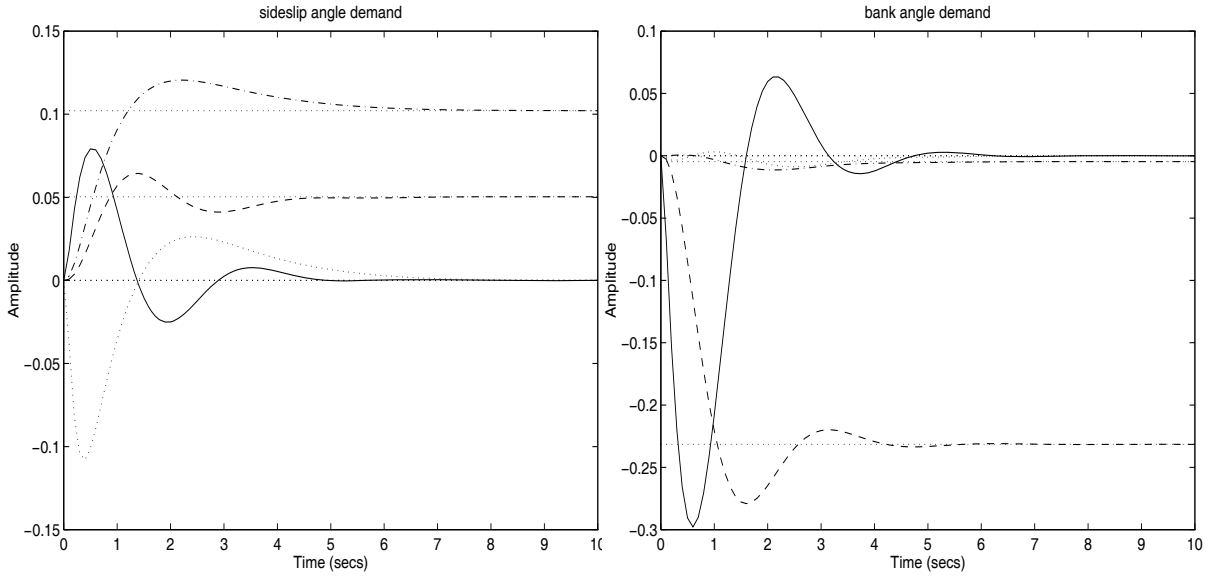

Figure 8.8: New closed loop input responses

Compare to Figure 8.1, we see in Figure 8.7 that we have actually retained the levels of output decoupling on the sideslip angle and bank angle outputs. In addition, we have slightly reduced the level of input decoupling and increased the robustness of the system.

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 Assign 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

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

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

$$K = \begin{bmatrix} 9.4136 & 0.1147 & -32.9886 & 4.0100 \\ 3.4395 & 3.3971 & -17.9012 & -34.0118 \end{bmatrix} . \tag{8.32}$$

The closed loop eigenvalues are

closed-loop eigenvalue	mode	frequency	damping	sensitivity
$-7 \pm 5i$	dutch roll	8.6023	0.8137	131.74
$-15 \pm 4i$	roll	15.524	0.9662	86.17
-6.2805	aileron	6.2805	1	193.52
-0.5785	rudder	0.5785	1	2.46
4.0879	washout filter	4.0879	-1	120.3

.

The dutch roll and roll modes have an acceptable level of damping, but have high frequencies. The main problem, however, is that this choice of eigenvalues results in high feedback gains and the washout filter mode is highly unstable.

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.

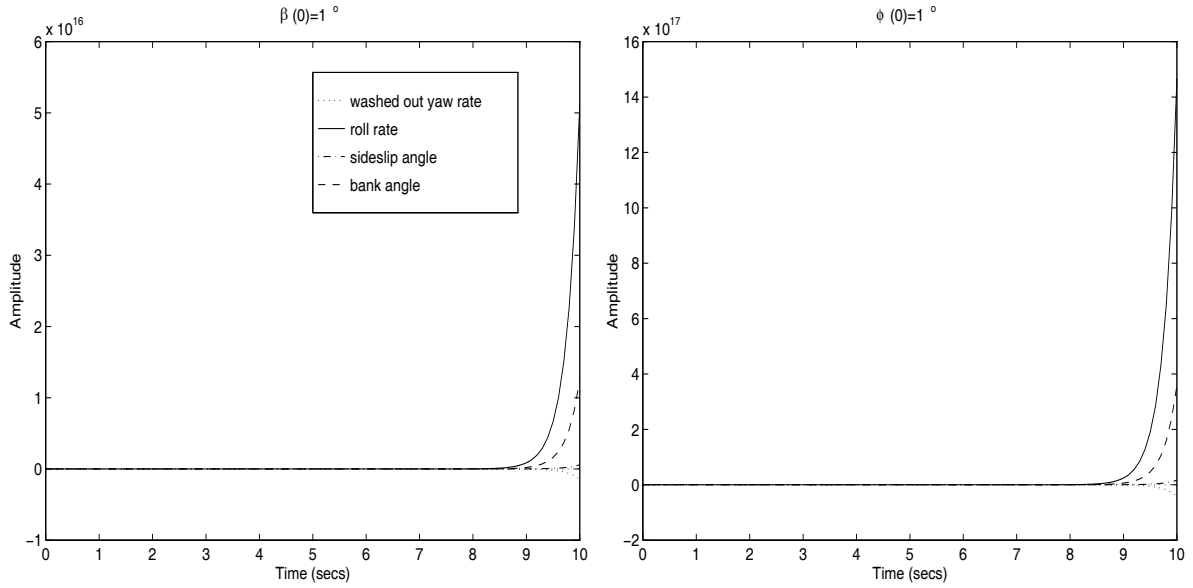

Figure 8.9: Original closed loop output responses for different assigned eigenvalues

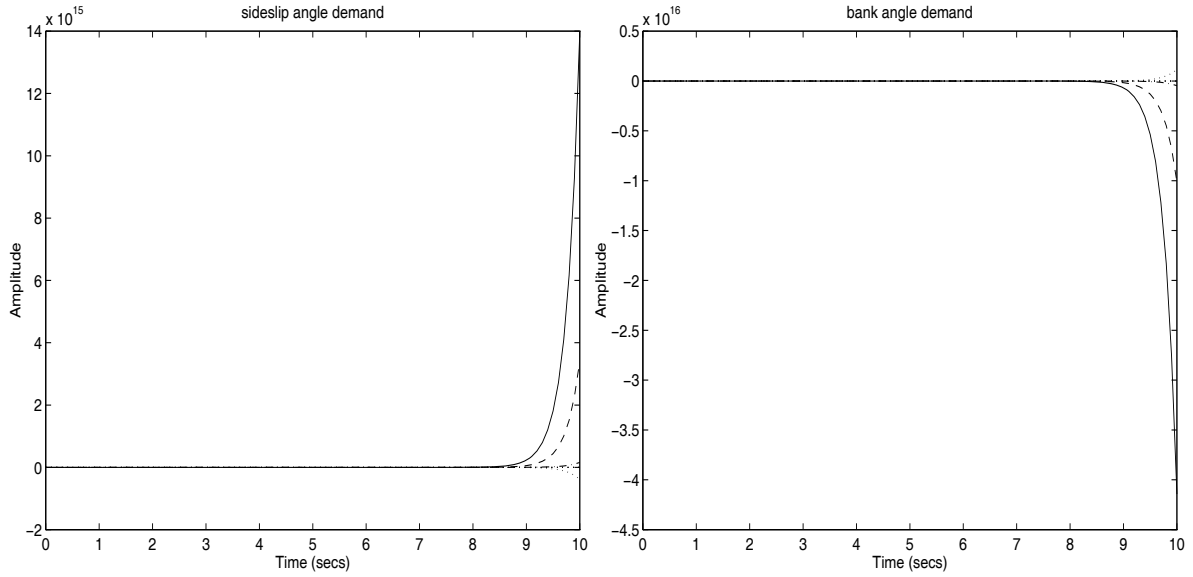

Figure 8.10: Original closed loop input responses for different assigned eigenvalues

The aim of using our methods here is to stabilise the system, in addition to reducing the level of input coupling and improving the robustness of the system.

8.3.6 Apply 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	$2.5921e+04$	$5.7446e-02$	$9.4707e+03$	$5.9998e+03$
3	$2.4996e+04$	$4.5494e-02$	$9.5285e+03$	$5.8357e+03$
4	$2.4756e+04$	$4.0736e-02$	$9.5810e+03$	$5.7906e+03$
5	$2.4684e+04$	$3.8439e-02$	$9.6167e+03$	$5.7667e+03$
6	$2.4661e+04$	$3.7249e-02$	$9.6385e+03$	$5.7532e+03$
7	$2.4653e+04$	$3.6612e-02$	$9.6513e+03$	$5.7456e+03$
8	$2.4650e+04$	$3.6265e-02$	$9.6586e+03$	$5.7414e+03$
9	$2.4649e+04$	$3.6073e-02$	$9.6627e+03$	$5.7390e+03$
10	$2.4649e+04$	$3.5968e-02$	$9.6650e+03$	$5.7377e+03$

As the scaling has no effect on the first reconstruction given in Chapter 7, we use the second reconstruction as in Section 7.3. This gives

$$K_2 = \begin{bmatrix} 9.4042 & -0.2950 & -32.8910 & -2.7612 \\ 3.4725 & 5.5972 & -18.2318 & 1.9933 \end{bmatrix}, \tag{8.38}$$

with the resulting closed loop eigenvalues

closed-loop eigenvalue	mode	frequency	damping	sensitivity
$-7.01 \pm 4.98i$	dutch roll	8.5985	0.8153	128.1
$-12.81 \pm 3.14i$	roll	13.1858	0.9711	52.77
-6.0501	aileron	6.0501	1	180.8
-0.5783	rudder	0.5783	1	4.03
-0.5119	washout filter	0.5119	1	2.55

.

The washout filter mode is now stable. There is not much movement in the other eigenvalues except for that corresponding to the roll mode. The condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 705.3392. \tag{8.39}$$

Here, the conditioning is slightly better, and we now have a stable system, but we have not looked at the coupling vectors. The mode input and output coupling vectors are

$$G_{1a} = \begin{bmatrix} 1 & -0.0691 + 0.0437i \\ 1 & -0.0691 - 0.0437i \\ -0.1254 + 0.0541i & 1 \\ -0.1254 - 0.0541i & 1 \end{bmatrix}, \tag{8.40}$$

$$G_{0a} = \begin{bmatrix} 1 & 0.0083 \mp 0.0433i \\ 0.0088 \pm 0.0080i & 1 \\ 0.0675 \pm 0.0626i & 0.0021 \mp 0.0017i \\ -0.0003 \mp 0.0014i & -0.0737 \mp 0.0181i \end{bmatrix}, \tag{8.41}$$

respectively. From these coupling vectors, we can see that the level of output decoupling has been retained, but the input decoupling has been only slightly improved. The new closed loop output and input responses are given in Figures 8.11 and 8.12, respectively.

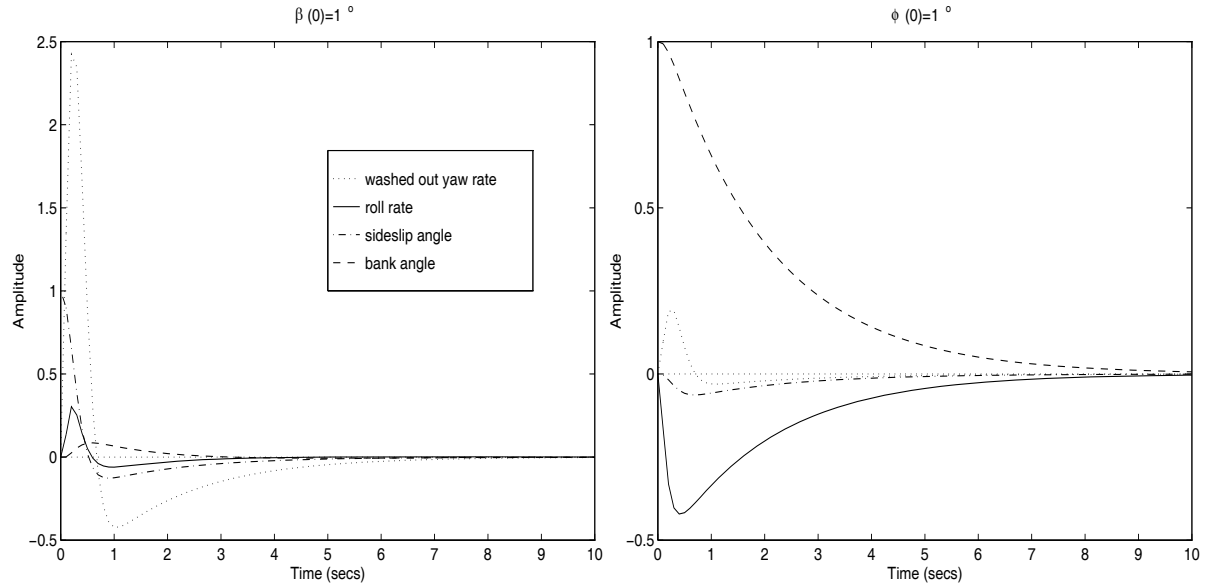

Figure 8.11: New closed loop output responses

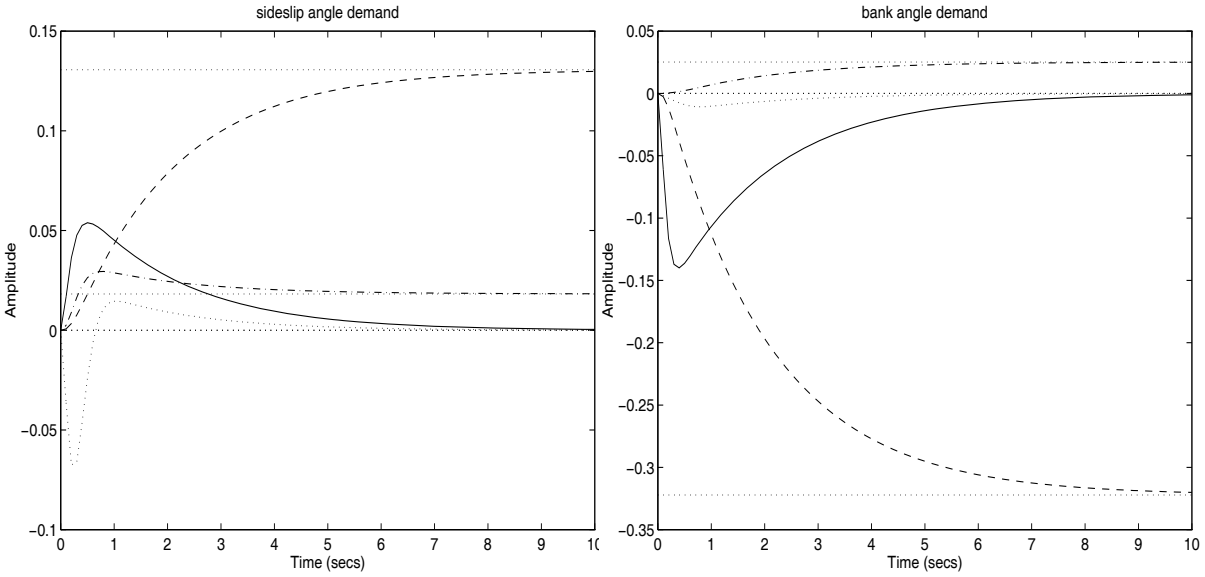

Figure 8.12: New closed loop input responses

In Figure 8.11 we see slight output coupling between sideslip angle and roll rate and between bank angle and washed out yaw rate. However, we have stabilised the system, reduced its sensitivity and slightly increased the input decoupling levels.

8.3.7 Apply unrestricted minimisation algorithm (for decoupling)

In Section 8.3.6, we stabilised the open loop system, but obtained only a slight improvement in the level of input decoupling. Here we aim to improve on this using the unrestricted minimisation algorithm. We use Method 5 to find an initial, real vector set $V = [V_1, V_2]$ and choose the weightings

$$(\omega_1^2, \omega_2^2) = (100, 1). \tag{8.42}$$

The results of the minimisation routine are

sweeps	objective function	$\|G_{1d} - G_{1a}\|_F^2$	$\kappa_F(V)$
0	$4.0496e+02$	$4.0188e+00$	$4.1182e+02$
1	$3.6285e+02$	$3.2317e+00$	$1.4777e+03$
2	$3.6207e+02$	$3.2076e+00$	$1.5077e+03$
3	$3.6057e+02$	$3.1819e+00$	$1.5273e+03$
4	$3.5867e+02$	$3.1541e+00$	$1.5429e+03$
5	$3.5778e+02$	$3.1378e+00$	$1.5561e+03$

To find a feedback that best assigns the minimisation vectors, we use the diagonal solver, allowing two complex conjugate pairs of eigenvalues in Λ_1 and none in Λ_2, giving

$$K_3 = \begin{bmatrix} 9.0815 & -0.1286 & -28.9725 & 0.1228 \\ 3.1673 & 5.5682 & -14.3302 & 1.0733 \end{bmatrix}. \tag{8.43}$$

The new closed loop eigenvalues are

closed-loop eigenvalue	mode	frequency	damping	sensitivity
$-7.41 \pm 4.39i$	dutch roll	8.6154	0.8599	136.9
$-12.91 \pm 3.36i$	roll	13.3415	0.9677	43.15
-5.3813	aileron	5.3813	1	160.8
-0.5908	rudder	0.5908	1	2.74
-0.1607	washout filter	0.1607	1	1.06

The system has been made stable, although there is not a great improvement in the conditioning,

$$\kappa_F(V) = 685.2460. \tag{8.44}$$

However, the new mode input and output coupling vectors are

$$G_{1a} = \begin{bmatrix} 1 & 0.0297 + 0.0278i \\ 1 & 0.0297 - 0.0278i \\ -0.0769 + 0.0843i & 1 \\ -0.0769 - 0.0843i & 1 \end{bmatrix}, \tag{8.45}$$

$$G_{0a} = \begin{bmatrix} 1 & 0.0048 \mp 0.0143i \\ -0.0580 \mp 0.0602i & 1 \\ 0.0717 \pm 0.0547i & 0.0014 \mp 0.0005i \\ 0.0022 \pm 0.0094i & -0.0725 \mp 0.0189i \end{bmatrix}, \tag{8.46}$$

respectively. We see that we have retained the desired level of output decoupling and have also reduced the level of input decoupling to a satisfactory level. The closed loop output and input responses are given in Figures 8.13 and 8.14, respectively.

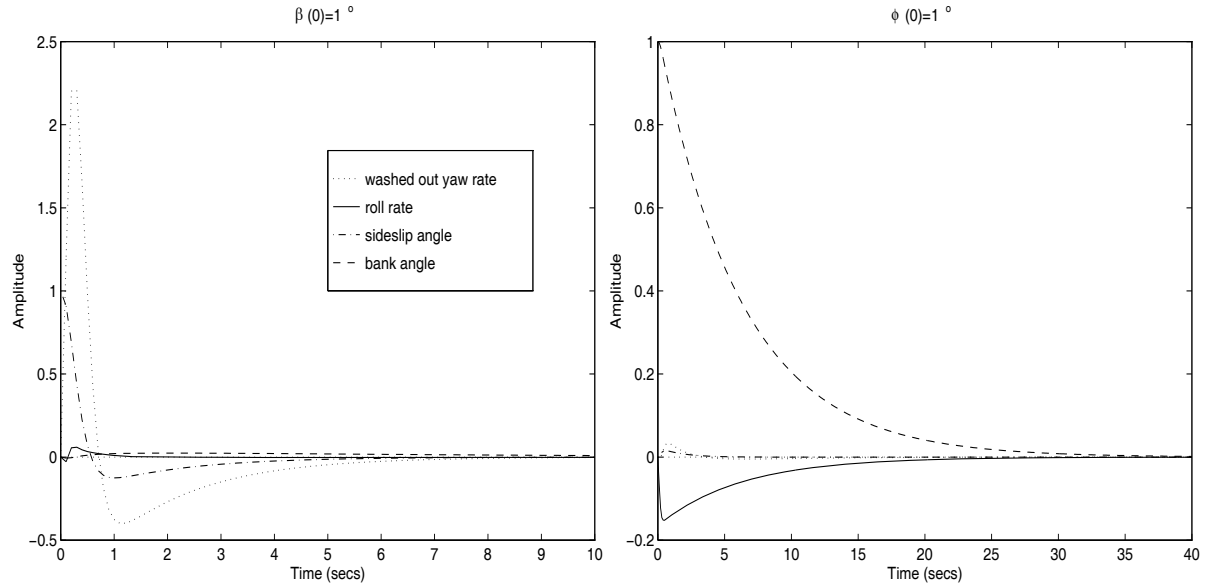

Figure 8.13: New closed loop output responses

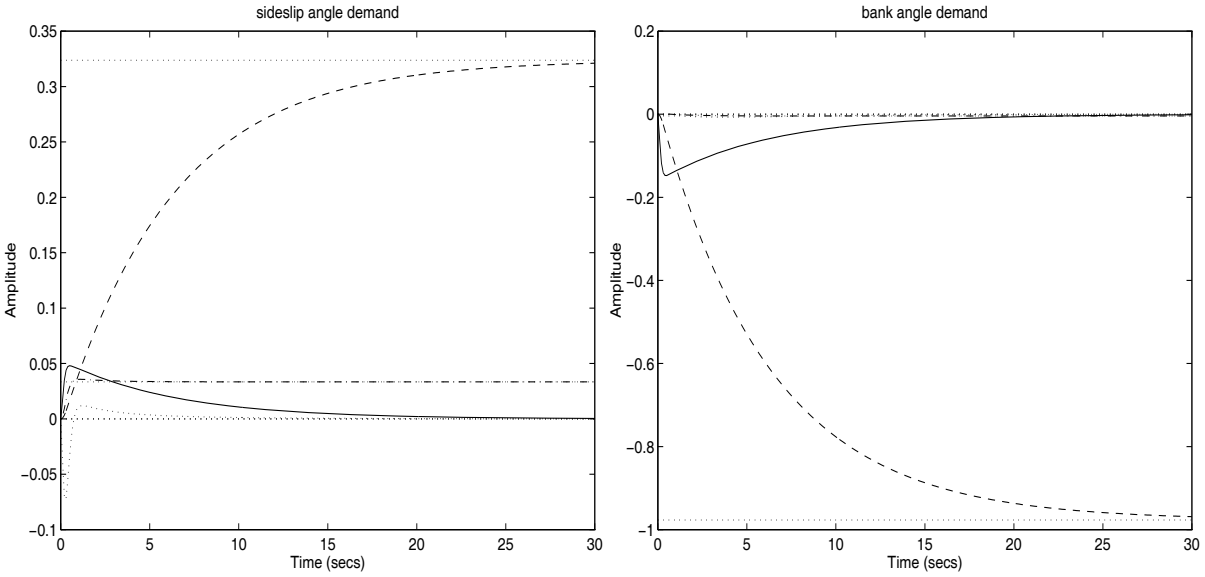

Figure 8.14: New closed loop input responses

In Figure 8.13 we can see that almost exact decoupling has been achieved for initial conditions on both sideslip angle and bank angle. We have also obtained the desired levels of input decoupling. In addition we have stabilised the system and slightly improved its robustness.

8.3.8 Apply 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	$3.0820e+00$	$4.0188e+00$	$4.1182e+02$
1	$4.9892e+00$	$1.6159e-01$	$1.2201e+02$

.

When the feedback is reconstructed, the new closed loop system is unstable for all combinations of real and complex conjugate pairs of eigenvalues allowed to appear in Λ_1 and Λ_2, using the diagonal solver. The unstable eigenvalue is not as far from the imaginary axis as in Section 8.3.5, but is still unacceptable. We therefore use the constrained diagonal solver as in Section 7.3.6 and, after experiments, choose to restrict

$$\tilde{\lambda}_7 \leq -20. \tag{8.48}$$

We are not restricting the actual closed loop eigenvalue to lie to the left of the $\mathrm{Re}(\tilde{\lambda}_j) \leq -20$ line (unless the problem is solved exactly), but are moving $\tilde{\lambda}_7$ by a large amount so that λ_7 does not cross into the closed right hand plane. The new closed loop feedback is

$$K_3 = \begin{bmatrix} 0.9438 & -0.0341 & 1.1108 & 0.3066 \\ 0.6833 & 4.5326 & -3.7859 & 0.5678 \end{bmatrix}. \tag{8.49}$$

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.

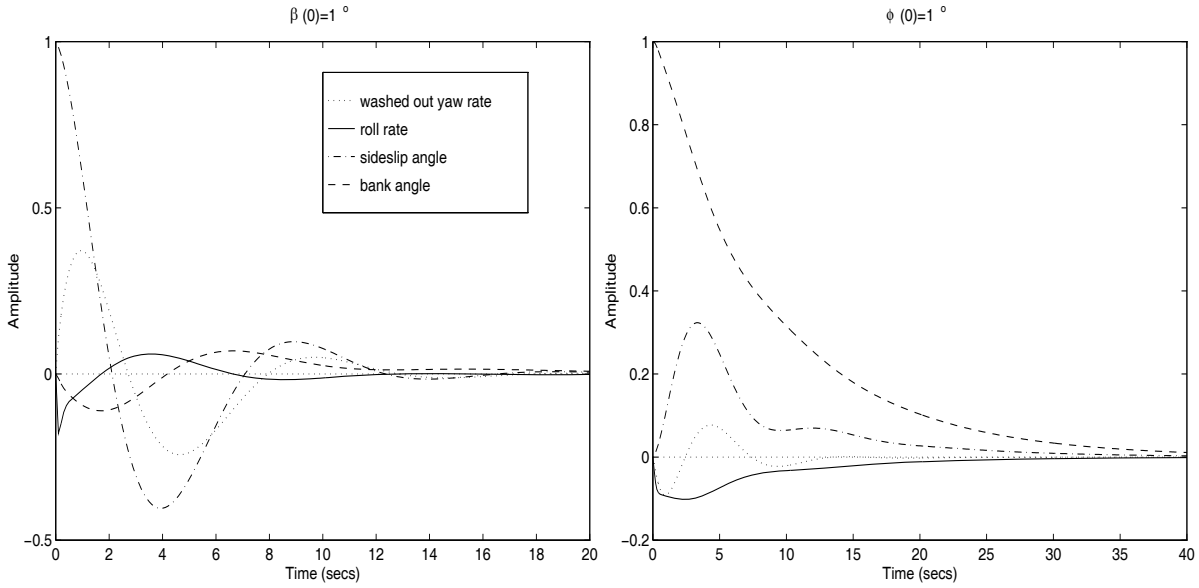

Figure 8.15: New closed loop output responses

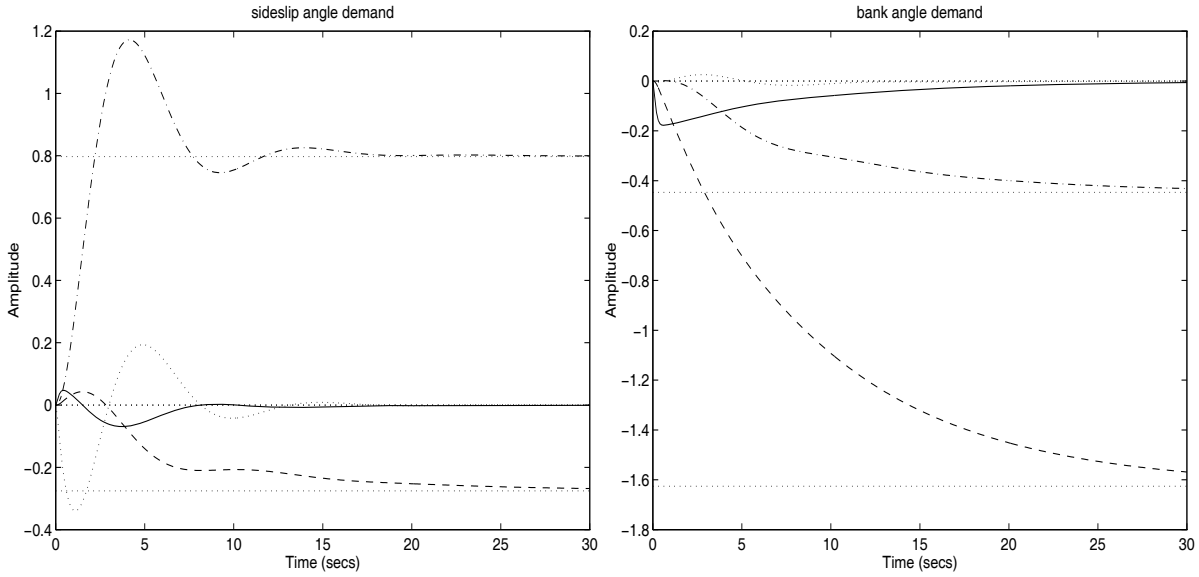

Figure 8.16: New closed loop input responses

From these response diagrams we can see that the reduction in the system sensitivity is at the expense of introducing increased oscillatory motion. We also see that we have increased the level of coupling between the bank angle and sideslip angle in both the output and input responses.

8.3.9 Example 1 conclusions

Example 1 has been used to test our methods with two different choices for the sets of desired eigenvalues. For the first set, we decreased the level of input coupling, but at the expense of increasing the level of output coupling. We also obtained new closed loop systems with lower sensitivity measures. The second desired eigenvalue set resulted in an unstable original closed loop system, which is a problem when only assign a certain number of modes. We successfully applied our algorithms to stabilise the system, reduce its sensitivity and obtain reduced levels of input coupling.

The results for Example 1 illustrate that it is not generally possible to obtain exactly all of the design specifications, but that there is a trade-off necessary between the individual requirements.

8.4 Example 2

The second example is taken from Smith [52] and is a generic model of a VSTOL aircraft which uses vectored thrust from a single jet engine. For a flight condition of 120 knots and 100 feet the longitudinal state vector, input vector and output vector are

$$\mathbf{x} = \begin{bmatrix} \theta & \text{pitch attitude} \\ q & \text{pitch rate} \\ u & \text{longitudinal body velocity} \\ w & \text{normal body velocity} \\ \eta & \text{tailplane angle} \\ \epsilon & \text{throttle position} \\ \theta_j & \text{nozzle angle} \\ \sigma_{fn} & \text{engine fan speed} \\ \sigma_{hn} & \text{engine compressor speed} \\ q_f & \text{engine fuel flow} \end{bmatrix}, \tag{8.51}$$

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

respectively. The system matrices are given by

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

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

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 20.0000 & 0 & 0 \\ 0 & 5.4060 & 8.4120 \\ 0 & -522.4000 & 335.8000 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{8.54}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5863 & 0.0824 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0.0393 & -0.2799 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{8.55}$$

The open loop eigenvalues of the system are

open-loop eigenvalue	frequency	damping	sensitivity
$-7.4520 \pm 5.7738i$	9.4270	0.7905	13.51
$-0.0343 \pm 0.4155i$	0.4169	0.0823	47.29
-0.2001	0.2001	1	23.65
-0.7593	0.7593	1	110.85
-4.8842	4.8842	1	7.18
-20	20	1	1.07
-10	10	1	6.4
-0.05	0.05	1	1

.

The condition number of the right eigenvectors is

$$\kappa_F(V) = 421.6572. \tag{8.56}$$

We now perform partial eigenstructure assignment to again see the error in the matching of the left eigenvectors.

8.5 Partial eigenstructure assignment

For this system, rank $(C) = 4$, so that we can assign (almost exactly), four closed loop eigenvalues. These are chosen to be

$$\Lambda_p = \begin{cases} -0.7 \pm 0.3i & \text{pitching} \\ -3.8 & \text{speed mode} \\ -0.2 & \text{flightpath mode} \end{cases} \tag{8.57}$$

with the corresponding desired mode output and input coupling vectors

$$G_{0d} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ x & x & 0 & 0 \end{bmatrix}, G_{1d} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8.58}$$

respectively. These mode input and output coupling vectors are chosen so as to decouple pitch attitude from airspeed and flightpath angle, and to decouple both airspeed and flightpath angle from the other three modes. For each feedback constructed in this example we present the closed loop output response to an initial condition of 0.05^o on pitch attitude (all other initial conditions are zero) and the closed loop response to a step input on pitch attitude, airspeed and flightpath angle. Again, the is no feedforward command tracker implemented on the input responses (see corresponding comment in Section 8.2).

Using Method 1 to find an initial closed loop system the feedback gains are calculated as

$$K = \begin{bmatrix} -0.0147 & 0.0380 & 0.0474 & 0.1136 \\ -0.0289 & -0.0707 & 0.0040 & -0.0066 \\ -0.0944 & -0.0551 & 0.0241 & -0.0216 \end{bmatrix}. \tag{8.59}$$

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 sensitivity of the first complex mode. The mode output coupling vectors corresponding to the four desired eigenvalues are

$$G_{0a} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.7 \pm 0.3i & 0 & 0 \end{bmatrix}. \tag{8.61}$$

These have been normalised so that the greatest element in each column is a one, and have been achieved exactly. The corresponding mode input coupling vectors (when normalised so that the largest element (in modulus) in each row is one) are

$$G_{1a} = \begin{bmatrix} -0.3694 - 0.5159i & 1 & -0.3113 + 0.1826i \\ -0.3694 + 0.5159i & 1 & -0.3113 - 0.1826i \\ 0.0012 & 1 & -0.6077 \\ 0.0146 & -0.7931 & 1 \end{bmatrix}. \tag{8.62}$$

From the results we can plainly see that the mode output coupling vectors, G_{0a} have been achieved to a satisfactory level; the mode output coupling vectors, G_{1a} have not been. The problem is that the second input (airspeed) is coupled to

pitch attitude, pitch rate and flightpath angle and the third input (flightpath angle) is coupled to pitch attitude, pitch rate and airspeed.

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

$$\begin{aligned} \|G_{0d} - G_{0a}\|_F^2 &= 3.7332 \times 10^{-30} \\ \|G_{1d} - G_{1a}\|_F^2 &= 5.6620 \times 10^{3} \end{aligned} , \tag{8.63}$$

respectively. 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 Figure 8.17.

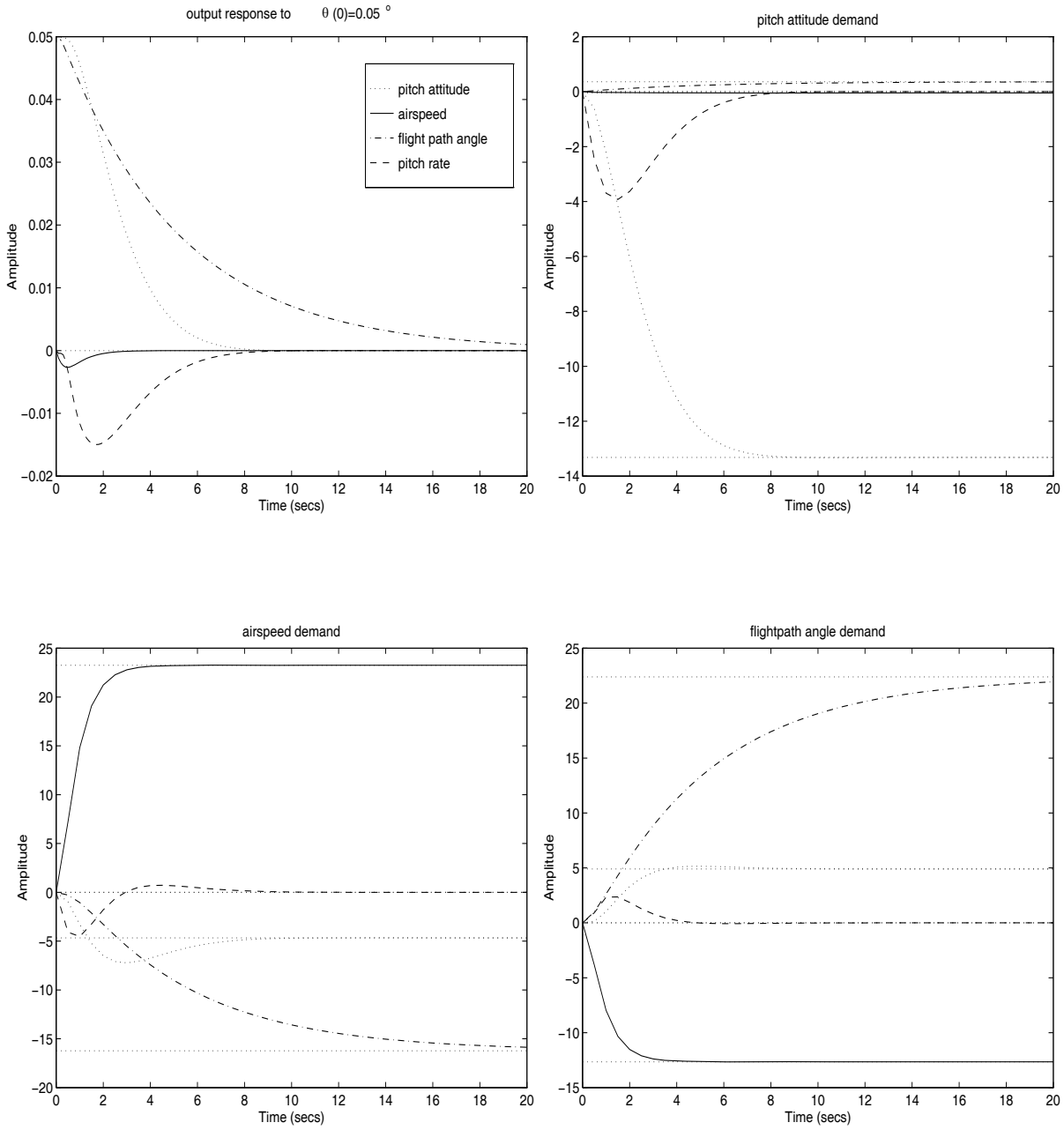

Figure 8.17: Original closed loop output and input responses

We can see from these diagrams that the pitch attitude demand response is satisfactory as it is decoupled from both airspeed and flightpath angle. However, both of the responses to airspeed demand and flightpath angle demand are inappropriately coupled to the other three modes.

8.5.1 Apply 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	$5.2047e+02$	$5.2047e+02$	$1.0502e+04$	$1.7029e+05$
5	$4.8870e+02$	$4.8870e+02$	$1.1354e+04$	$1.8651e+05$
6	$4.6690e+02$	$4.6690e+02$	$1.2098e+04$	$2.0124e+05$
7	$4.5122e+02$	$4.5122e+02$	$1.2716e+04$	$2.1405e+05$
8	$4.3936e+02$	$4.3936e+02$	$1.3224e+04$	$2.2504e+05$
9	$4.2994e+02$	$4.2994e+02$	$1.3647e+04$	$2.3455e+05$

To find the new feedback matrix that best assigns the new set of vectors, $\tilde{V} = [V_1, \tilde{V}_2]$, we use the second reconstruction from Section 7.2.2,

$$K_2 = \begin{bmatrix} -0.0250 & 0.0385 & 0.0476 & 0.0987 \\ 0.0264 & -0.0690 & 0.0034 & 0.0726 \\ -0.0320 & -0.0535 & 0.0232 & 0.0677 \end{bmatrix}. \tag{8.65}$$

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 minimisation. The new mode input coupling vectors are

$$G_{1a} = \begin{bmatrix} 1 & -0.6910 + 0.3598i & -0.0049 - 0.1523i \\ 1 & -0.6910 - 0.3598i & -0.0049 + 0.1523i \\ 0.0387 & 1 & -0.7513 \\ -0.4414 & -0.7118 & 1 \end{bmatrix}, \tag{8.67}$$

so that, even though we have not achieved the perfect decoupling required, we have improved the decoupling so that the largest element (in modulus) is in the correct place in each row. The corresponding mode output coupling vectors are

$$G_{0a} = \begin{bmatrix} 1 & 0.0015 & -0.0162 \\ 0.2474 \pm 0.1050i & 1 & -0.0131 \\ -0.0302 \mp 0.0526i & -0.0029 & 1 \\ -0.5213 \pm 0.1073i & -0.0059 & 0.0033 \end{bmatrix}. \tag{8.68}$$

This improvement in the input decoupling has been achieved at the expense coupling the airspeed into the pitching mode, which shows that we cannot look at

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

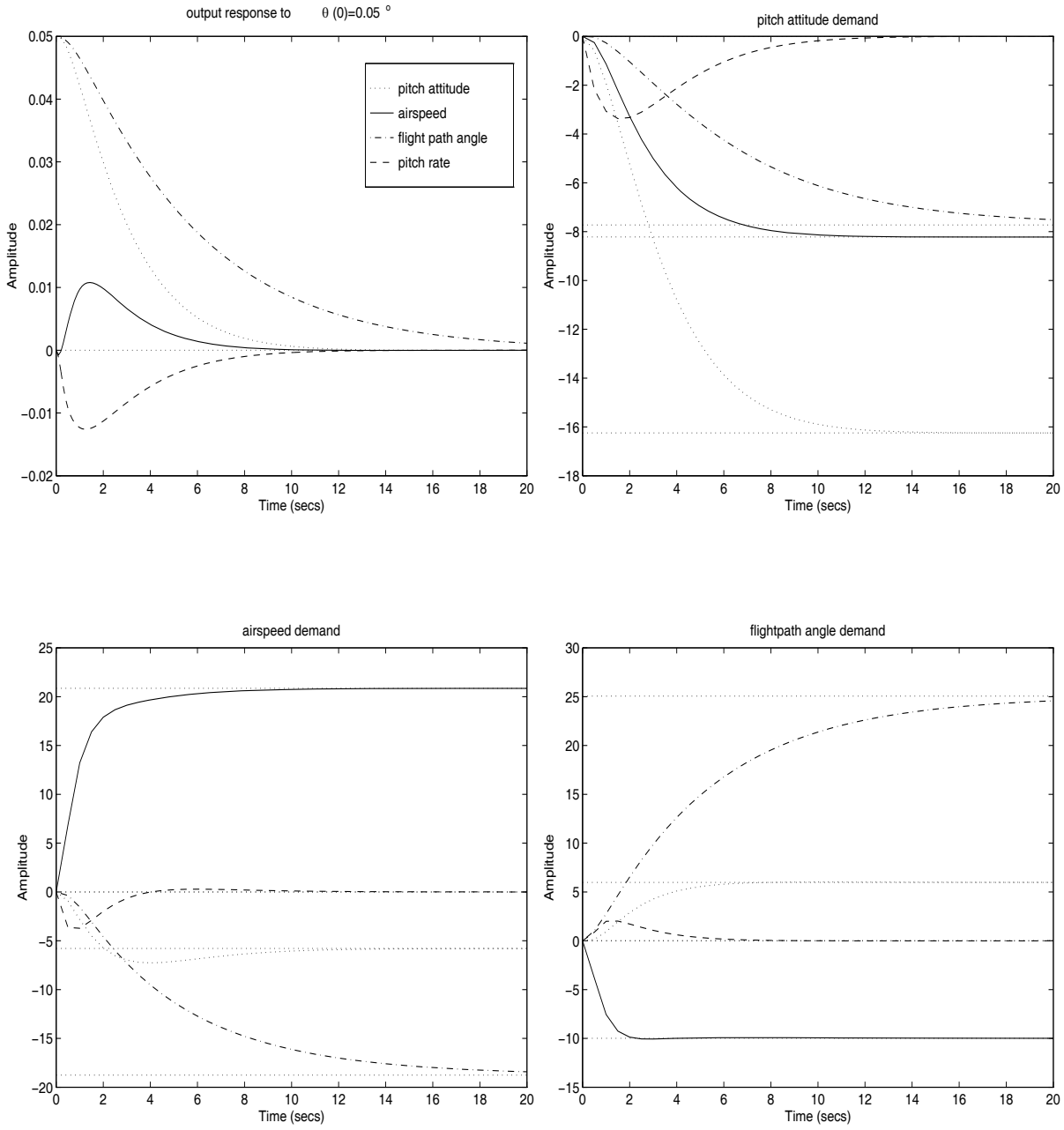

Figure 8.18: New closed loop output and input responses

coupling introduced between pitch attitude and airspeed. However, our minimisation on the left vectors has reduced the levels of input coupling between input 2 (airspeed) and flightpath angle and between input 3 (flightpath angle) and pitch attitude/pitch rate. The robustness of the system has increased slightly;

this again illustrates the balance to be obtained between levels of input/output coupling, stability and robustness.

8.5.2 Apply restricted minimisation algorithm (for conditioning)

In Section 8.5.1 we reduced the level of input decoupling, but at the expense of an increase in the sensitivity of the eigenvalues . This problem was run with the weightings $(\omega_1^2, \omega_2^2, \omega_3^2) = (0, 1, 0)$, but resulted in an unstable system. Therefore we include a weighting on the left eigenspace error to restrict the eigenvalue movement. Our chosen weightings are thus

$$(\omega_1^2, \omega_2^2, \omega_3^2) = (0, 10, 1). \tag{8.69}$$

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	$4.0235e+04$	$5.6620e+03$	$1.1141e+03$	$2.4164e-26$
1	$9.7320e+03$	$2.7090e+04$	$5.4208e+02$	$2.0600e+02$

Using the first reconstruction of the feedback as in Section 7.3, we obtain

$$K_1 = \begin{bmatrix} 0.0517 & -0.7672 & -0.0014 & 0.0596 \\ 0.0142 & -0.3596 & -0.0136 & -0.0056 \\ -0.0178 & -0.5038 & -0.0062 & -0.0194 \end{bmatrix}, \tag{8.70}$$

with the new closed loop eigenvalues

closed-loop eigenvalue	frequency	damping	sensitivity
$-0.6394 \pm 0.7565i$	0.6455	0.9905	63.5
$-3.5068 \pm 1.9702i$	0.8718	4.0223	39.03
-0.1997	1	0.1997	12.07
-8.9247	1	8.9247	17.32
-4.6329	1	4.6329	39.03
-19.6327	1	19.6327	1.46
$-7.0665 \pm 4.1641i$	0.8615	8.2021	31.62

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.

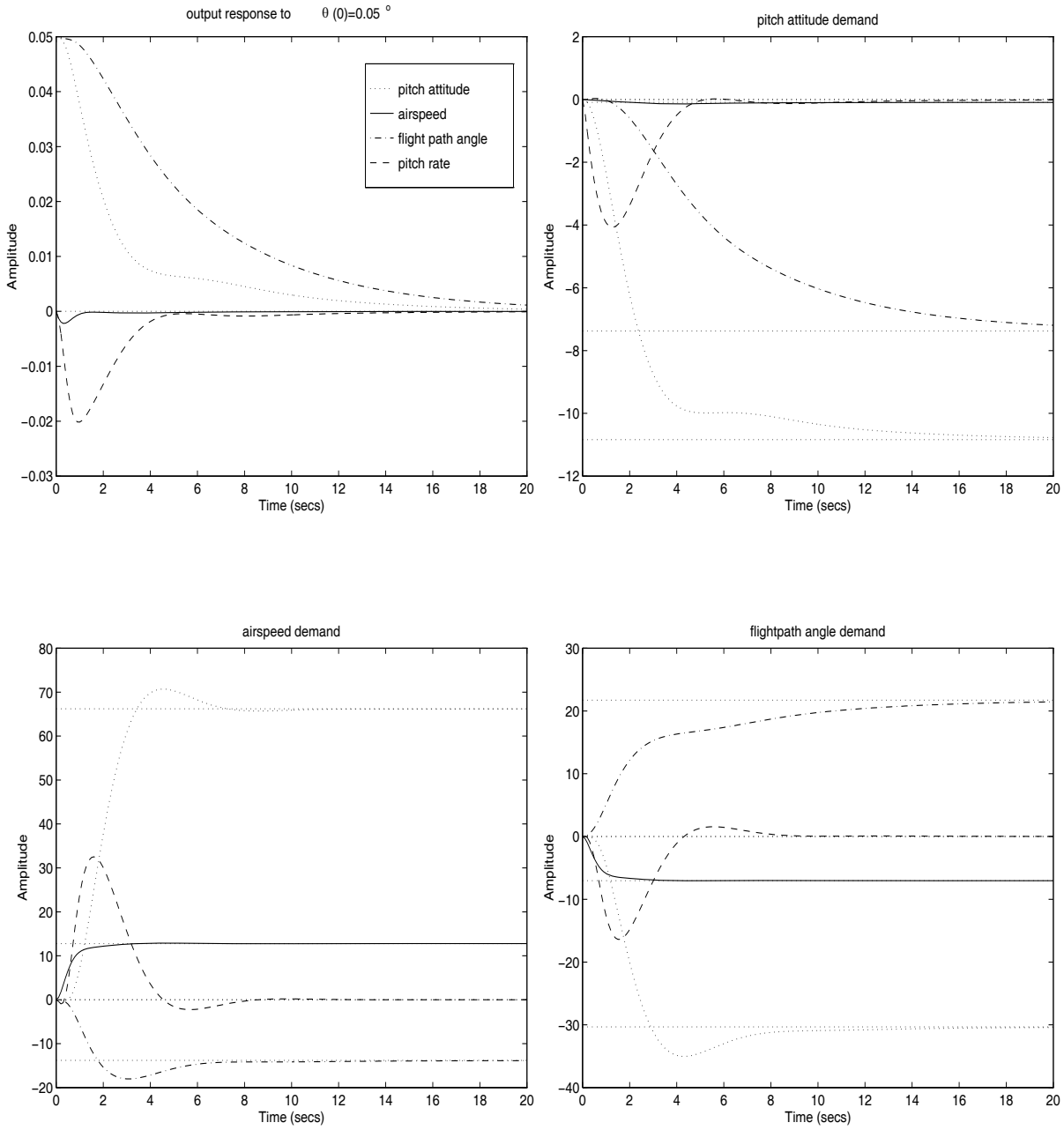

Figure 8.19: New closed loop output and input responses

We can see that we have retained the desired level of output decoupling due to an initial condition on the pitch attitude. Also, while the airspeed has not been coupled into pitch attitude as previously, flightpath angle has been. An increase in the coupling is expected due to the vast improvement in the robustness of the system.

8.5.3 Apply 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 condition number of the eigenvectors of the closed loop system is

$$\kappa_F(V) = 795.1898. \tag{8.74}$$

The new mode input coupling vectors are

$$G_{1a} = \begin{bmatrix} 1 & -0.6321 + 0.9665i & -0.1390 - 0.3143i \\ 1 & -0.6321 - 0.9665i & -0.1390 + 0.3143i \\ 0.0077 & 1 & -0.5985 \\ -0.1309 & -0.7437 & 1 \end{bmatrix}, \tag{8.75}$$

so that, even though we have not achieved the perfect decoupling required, we have improved the decoupling so that the largest element (in modulus) is in the correct place in each row. The corresponding mode output coupling vectors are

$$G_{0a} = \begin{bmatrix} 1 & 0 & 0.1884 \\ -0.1888 \pm 0.1141i & 1 & -0.5959 \\ -0.0820 \mp 0.0205i & 0 & 1 \\ -0.6373 \pm 0.3146i & 0 & -0.0087 \end{bmatrix}. \tag{8.76}$$

Only a small level of coupling has been introduced between the pitching mode and airspeed, but there is substantial coupling between the fourth mode and pitch attitude and airspeed. The new minimised output and input responses are presented in Figure 8.20.

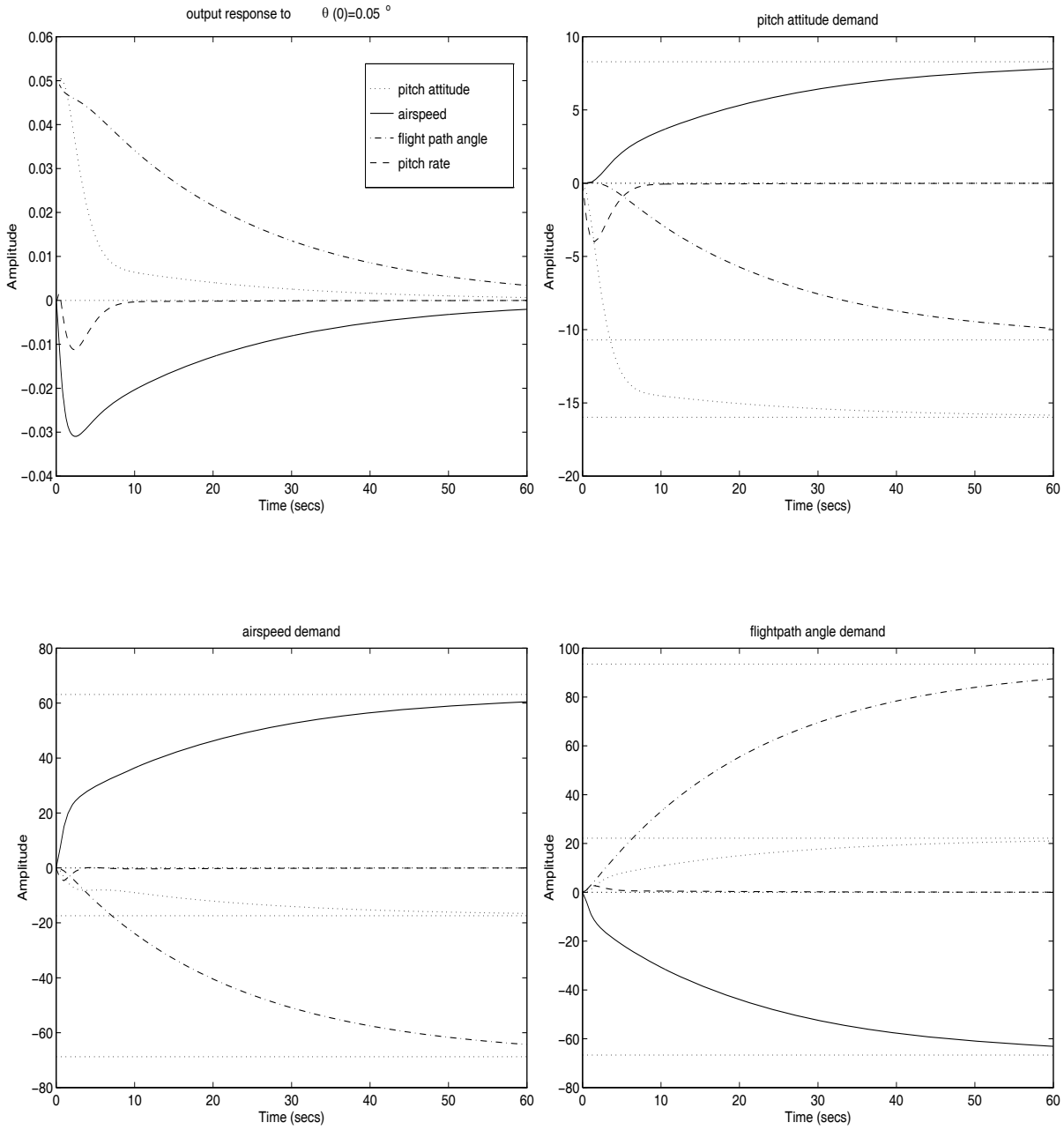

Figure 8.20: New closed loop output and input responses

The output response diagram clearly shows the increased level of output coupling. So, although we have apparently reduced the levels of input coupling from G_{1a}, this improvement has been lost due to the coupling in the outputs. This again shows the need to consider the links between the input and output coupling vectors simultaneously. Despite these coupling increases, we have still reduced the sensitivity of the system.

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 Conclusions

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 two aircraft examples. In Example 1, we obtained the desired levels of decoupling for both the inputs and the outputs. For Example 2, we improved the balance of the decoupling by improving the input decoupling at the expense of the level

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 package (as in Section 6.7). The minimisation routines are clever in the way that a non-linear problem is reduced to a simple linear least squares system by a choice of scaling.

Therefore, we have developed an efficient, flexible design tool for aircraft flight control system design that simultaneously balances the levels of input and output decoupling and the robustness of the system.

In the next chapter we summarise this thesis. We indicate areas of improvement and possible extensions to our work.

Chapter 9

Conclusions 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. We then related this theory to the specifications of aircraft problems and gave an example to illustrate the importance of considering the left eigenvectors in addition to the right eigenvectors to control the levels of input and output coupling, respectively. Smith [52] identified the worth in assigning desired left vectors, but little was done beyond using an optimisation package on the whole problem.

In Chapter 5 we extended this work to control the left vectors directly. We obtained a set of right vectors that gave exactly the desired output decoupling level. We then developed a minimisation technique that retained these vectors and improved the matching of the left vectors by choosing iteratively a new set of the unassigned right vectors. Consideration was also given to the conditioning of the vectors, and to the error of the left vectors from their correct subspaces. This was formulated as a multi-criteria minimisation method with weightings chosen by the designer. The problem was non-linear, but was reduced into a linear least squares form by a choice of right vector scaling. We gave results of the routine with various weighting values. Good results were obtained but the small dimension of the subspaces from which vectors were chosen had the effect that the objective function value was reduced considerably in the first few sweeps, but further reductions took a lot of work. With this in mind, we presented two scalings on the retained set of right vectors to further reduce the left vector matching error.

The results of Chapter 5 were constrained by the fact that the minimisation vectors were restricted to be in certain subspaces; in Chapter 6 we removed this restriction. We specified no eigenvalues for the new vectors to correspond to; these vectors were allowed to be anywhere in complex space. The theory was similar to that in Chapter 5, except that there were only two criteria in the minimisation because of the removal of the left eigenspace error. By letting the vectors lie anywhere in complex space, we were obtaining a set of vectors that were not self-conjugate. We thus gave three methods for calculating an initial set of real vectors using the real and imaginary part formulation; these were used as the starting point for the minimisation routine. The results showed that, when the vector conditioning was weighted to zero, the left vector matching error could be reduced to zero very quickly. This was not the case when the conditioning was introduced, but good results were still obtained with a high relative weighting on the left vector matching.

Both Chapters 5 and 6 gave a set of vectors that reduced an objective function, but had no feedback matrix that assigned these vectors. The aim of Chapter 7 was thus to find a feedback that best assigned the set of vectors obtained from

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 feedback that best assigned the unrestricted minimisation vectors, we developed a method (diagonal solver) that produced a feedback matrix while attempting to find a set of eigenvalues. Unless the problem was solved exactly, these values would differ from the final closed loop eigenvalues. However, we showed a bound on this error, one component of which was the condition number of the vectors. We also included a constrained version of our method using a NAG minimisation routine.

We thus had two core methods for obtaining a set of vectors to minimise some objective function, and various methods for constructing a feedback to best assign these vectors. In Chapter 8, to illustrate all of this theory, we presented two examples from the aircraft industry. These examples were used to test both the restricted and unrestricted minimisation algorithms with various parameter weightings to improve the matching of the left vectors and/or the sensitivity of the system whilst retaining an assigned set of right vectors. The examples illustrated that our methods achieved a balance between the levels of input and output decoupling. Exact output decoupling could be achieved by standard partial eigenstructure assignment; the minimisation algorithms and the construction of the feedback relaxed the exact attainment of the right vectors to reduce the error of the matching of the left vectors. We also significantly reduced the value of the conditioning of the systems considered. Thus, we demonstrated the achievable trade-off between the design specifications of performance, levels of coupling, stability and robustness. However, the results would benefit from a full non-linear simulation and appraisal from experienced aerospace engineers.

There are some extensions to this work that could be explored, which would

improve our results. If we wish the eigenvalues to move, then we have to put a low relative weighting on the left eigenspace error. It would be an improvement if it were possible to allow the eigenvalues to be constrained to certain regions that would correspond to specified areas of frequency and damping.

The main area where an improvement could be made is in the (re)construction of the feedback. We managed to obtain vectors that give good decoupling, but lose some of this accuracy in the feedback construction. It would certainly be beneficial if it were possible to choose the feedback at the same time as the vectors are updated so that, at the end of the minimisation routine, we have a set of vectors and a feedback matrix that assigns these vectors. We have looked at an idea of using the Gershgorin circle theorem to give bounds on the elements of the feedback to satisfy stability requirements for the positions of the closed loop eigenvalues. The left vector matching error could then be minimised subject to the inequality constraints placed on the elements of K. Initial studies suggest that this may be too restrictive as there may not be a feasible region satisfying the constraints.

It is also possible to solve the whole problem using an optimisation package. The design specifications would be the levels of input and output coupling, areas of frequency and damping for the eigenvalues, and robustness. The problem could be optimised over the region of the possible eigenvalue positions to generate a feedback matrix. Our experiments indicate that these programs can take a number of hours to generate a solution that may only be a local optimum.

In conclusion, the problem of using eigenstructure assignment to obtain a number of design specifications in the aircraft industry is not solvable exactly. Previous work omitted the consideration of the left eigenvectors; we have developed efficient methods to balance the levels of input and output decoupling and to reduce the sensitivity of a system. We have highlighted the possible extensions that could be investigated to improve this work.

Bibliography

[1] A. N. Andry, E. Y. Shapiro, and J. C. Chung. Eigenstructure assignment for linear systems. *IEEE Trans. on Aerospace and Electronic Systems*, AES-19(5):pp. 711–728, September 1983.

[2] P. R. Apkarian. Structured stability robustness improvement by eigenspace assignment techniques : A hybrid methodology. *Journal of Guidance, Control and Dynamics*, 12(2):pp. 162–168, 1989.

[3] M. Arnold and B. N. Datta. An algorithm for the multi-input eigenvalue assignment problem. Technical report, Dept. of Mathematics, University of California, San Diego, CA, 1988.

[4] S. Barnett and R. G. Cameron. *Introduction to mathematical control theory.* Oxford University Press, 1990.

[5] A. Ben-Israel and T. N. Grenville. *Generalized inverses : theory and applications.* Wiley, 1974.

[6] S. P. Burrows and R. J. Patton. Design of a low-sensitivity, minimum norm and structurally constrained control law using eigenstructure assignment. *Optimal Control Applications and Methods*, 12:pp. 131–140, 1991.

[7] S. P. Burrows and R. J. Patton. Design of low-sensitivity modalized observers using left eigenstructure assignment. *Journal of Guidance, Control and Dynamics*, 15(3):pp. 779–782, 1991.

[8] S. P. Burrows, R. J. Patton, and J. E. Szymanski. Robust eigenstructure assignment with a control design package. *IEEE Control Systems Magazine*, pages 29–32, 1989.

[9] R. K. Cavin and S. P. Bhattacharyya. Robust and well-conditioned eigenstructure assignment via Sylvester's equation. *Optimal Control Applications and Methods*, 4:pp. 205–212, 1983.

[10] K. W. E. Chu, N. K. Nichols, and J. Kautsky. Robust pole assignment by output feedback. In *4th IMA conference on Control Theory*. Academic Press, 1985.

[11] E. J. Davison. On pole assignment in linear systems with incomplete state feedback. *IEEE Trans. Automatic Control*, AC-15:pp. 348–351, June 1970.

[12] E. J. Davison. The feedforward control of linear multivariable time-invariant systems. *Automatica*, 9:pp. 561–573, Feb. 1973.

[13] E. J. Davison and R. Chatterjee. A note on pole assignment in linear systems with incomplete state feedback. *IEEE Trans. Automatic Control*, AC-16:pp. 98–99, Feb. 1971.

[14] E. J. Davison and S. G. Chow. An algorithm for the assignment of closed-loop poles using output feedback in large linear multivariable systems. *IEEE Trans. Automatic Control*, pages 74–75, Feb. 1973.

[15] E. J. Davison and S. H. Wang. On pole assignment in linear multivariable systems using output feedback. *IEEE Trans. Automatic Control*, AC-20:pp. 516–518, August 1975.

[16] M. M. Fahmy and J. O'Reilly. On eigenstructure assignment in linear multivariable systems. *IEEE Trans. on Aut. Control*, AC-27(3):pp. 690–693, June 1982.

[17] M. M. Fahmy and J. O'Reilly. Eigenstructure assignment in linear multivariable systems-a parametric solution. *IEEE Trans. on Aut. Control*, AC-28(10):pp. 990–994, Oct. 1983.

[18] M. M. Fahmy and J. O'Reilly. Parametric eigenstructure assignment by output feedback control: the case of multiple eigenvalues. *Int. J. Control*, 48(4):pp. 1519–1535, 1988.

[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*, AC-29(7):pp. 658–661, July 1984.

[24] G. H. Golub and C. F. V. Loan. *Matrix computations*. North Oxford Academic, 1986.

[25] A. Grace. *Optimization Toolbox for use with MATLAB*. Mathworks, Natick, MA, 1990.

[26] D.-W. Gu, M. Oh, and S. K. Spurgeon. Robust pole assignment by homotopy methods. Technical Report 92-36, University of Leicester, Nov. 1992.

[27] J. Kautsky and N. K. Nichols. Robust pole assignment in systems subject to structured perturbations. *Systems and Control Letters*, 15:pp. 373–380, 1990.

[28] J. Kautsky, N. K. Nichols, and P. V. Dooren. Robust pole assignment in linear state feedback. *Int. J. Control*, 41(5):pp. 1129–1155, 1985.

[29] L. H. Keel, J. A. Fleming, and S. P. Bhattacharyya. Minimum norm pole assignment via Sylvester's equation. *Contemporary Mathematics*, 47:pp. 265–272, 1985.

[30] H. Kimura. Pole assignment by gain output feedback. *IEEE Trans. Automtatic Control*, AC-20(4):pp. 509–516, Aug. 1975.

[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. *Delta Toolbox for use with MATLAB*. Mathwork, Natick, MA, 1990.

[35] G. S. Minimis and C. C. Paige. An algorithm for pole assignment of time invariant linear systems. *Int. Journal of Control*, 35(2):pp. 341–354, 1982.

[36] B. C. Moore. On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment. *IEEE Trans. Automatic Control*, AC-21:pp. 689–692, Oct. 1976.

[37] S. K. Mudge and R. J. Patton. Analysis of the technique of robust eigenstructure assignment with application to aircraft control. *IEE Proceedings-Pt. D*, 135(4):pp.275–281, 1988.

[38] N. Munro and A. Vardulakis. Pole-shifting using output feedback. *Int. Journal of Control*, 18(6):pp. 1267–1273, 1973.

[39] J. J. O'Brien and J. R. Broussard. Feedforward control to track the output of a forced model. In *Proc. 17th IEEE Conf. on Decision and Control*, pages 1149–1155, 1978.

[40] M. Oh, D.-W. Gu, and S. K. Spurgeon. Robust pole assignment in a specified region by output feedback. Technical Report 91-30, University of Leicester, Dec. 1991.

[41] T. J. Owens. Parametric output-feedback control with left eigenstrucutre insensitivity. *Int. Journal of Systems Science*, 21(8):pp. 1603–1630, 1990.

[42] R. V. Patel and P. Misra. Numerical algorithms for eigenvalue assignment by state feedback. *Proceedings of the IEEE 1984*, 72(12):pp. 1755–1764, 1985.

[43] Y. Patel, R. J. Patton, and S. P. Burrows. Robust eigenstructure assignment for multirate control systems. In *First IEEE Conf. on Control Applications*, volume 2, pages 1024–1029, 1992.

[44] P. H. Petkov, N. D. Christov, and M. M. Konstantinov. On the numerical properties of the Schur approach for solving the Riccati equation. *Systems and Control Letters*, 9:pp. 197–201, 1987.

[45] J. E. Piou, K. Sobel, and E. Y. Shapiro. Robust Lyapunov constrained sampled-data eigenstructure assignment using the delta-operator with application to flight control design. In *First IEEE Conference on Control Applications*, volume 2, pages 1000–1005, 1992.

[46] B. Porter. Eigenvalue assignment in linear multivariable systems by output feedback. *International Journal of Control*, 25(3):pp. 483–490, 1977.

[47] B. Porter and A. Bradshaw. *Int. Journal of Systems Science*, 9(6):pp. 445, 1978.

[48] B. Porter and A. Bradshaw. Design of linear multivariable continuous-time tracking systems incorporating error-actuated dynamic controllers. *Int. Journal of Systems Science*, 9(6):pp. 627–637, 1978.

[49] B. Porter and J. J. D'Azzo. Algorithm for closed-loop eigenstructure assignment by state feedback in multivariable linear systems. *International Journal of Control*, 27(6):pp. 943–947, 1978.

[50] G. Roppenecker and J. O'Reilly. Parametric output feedback controller design. In *Proc. 10th IFAC World Congress*, pages 275–281, 1987.

[51] S. Slade. *Robust partial pole placement via output feedback.* PhD thesis, University of Reading, 1988.

[52] P. R. Smith. Application of eigenstructure assignment to the control of powered lift combat aircraft. Technical Report FS-1009, RAE Farnborough, Feb. 1991.

[53] K. M. Sobel, S. S. Banda, and H.-H. Yeh. Robust control for linear systems with structured state space uncertainties. *Int. J. Control*, 50(5):pp. 1991–2004, 1989.

[54] K. M. Sobel and E. Y. Shapiro. Design of decoupled longitudinal flight control laws utilizing eigensystem assignment. In *Proc. of the 1984 American Control Conference*, pages 403–408, 1985.

[55] K. M. Sobel and E. Y. Shapiro. Eigenstructure assignment : A tutorial - part 1 theory. In *Proc. 1985 American Control Conference*, pages 456–460. IEEE, 1985.

[56] K. M. Sobel and E. Y. Shapiro. Eigenstructure assignment : A tutorial - part 2 applications. In *Proc. 1985 American Control Conference*, pages 461–467. IEEE, 1985.

[57] K. M. Sobel and E. Y. Shapiro. Application of eigenstructure assignment to flight control design: some extensions. *Journal of Guidance, Control and Dynamics*, 10(1):pp. 73–81, Jan.-Feb. 1987.

[58] K. M. Sobel and E. Y. Shapiro. Flight control examples of robust eigenstructure assignment. In *Proc. of the IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, volume 2, pages 1290–1291, Jan.-Feb. 1987.

[59] K. M. Sobel, E. Y. Shapiro, and A. N. Andry. Eigenstructure assignment. *Int. Journal of Control*, 59(1):pp. 13–37, 1994.

[60] K. M. Sobel and W. Yu. Flight control application of eigenstructure assignment with optimization of robustness to structured state space uncertainty. In *Proc. of the IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, volume 2, pages 1705–1707, 1989.

[61] S. K. Spurgeon and R. J. Patton. Robust control design using eigenstructure assignment. In *Proc. of 5th IMA Conf. on Control Theory*, 1988.

[62] Sridhar and Lindhorff. Pole placement with constant gain output feedback. *Int. Journal of Control*, 18:pp. 993–1003, 1973.

[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 Decision 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 Decision 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.