ABSTRACT

Name: João Carvalho

Department: Mathematical Sciences

Title: State Estimation and Finite-Element Model Updating for Vibrating Systems

Major: Mathematical Sciences

Degree: Doctor of Philosophy

Approved by:

Date:

Dissertation Director

NORTHERN ILLINOIS UNIVERSITY

ABSTRACT

This dissertation is devoted to the study of two important problems arising in vibration analysis and control of structures modeled by a system of matrix second-order differential equations: the State Estimation and the Finite Element Model Updating problems.

The state estimation problem of a second-order control system is solved via its associated descriptor system (generalized state-space model). A new theory is developed using generalized Sylvester-observer equation and then three new algorithms for solving this equation are proposed. These new algorithms generalize the existing algorithms for the Sylvester-observer equation for standard state-space systems. Indeed, a new block algorithm for the standard Sylvester-observer equation, generalizing a well-known algorithm due to Van Dooren, is also proposed. These algorithms, which are rich in BLAS-3 computations, are tested on some benchmark problems using high-performance software libraries LAPACK and SLICOT.

The finite-element model updating problem, although is of significant industrial importance, has not yet been solved satisfactorily. A new solution to the problem in the undamped symmetric positive semidefinite case is proposed in this dissertation. The new algorithm works with a set of incomplete measured data and, unlike other existing algorithms, does not require model reduction. An extension of the algorithm to the damped case is also made and possible solutions are suggested. These new algorithms for model updating are based on a set of new orthogonality relations between the eigenvectors of a symmetric positive semidefinite quadratic pencil. These new orthogonality relations, which are also derived in this dissertation, generalize the

existing ones for the symmetric positive definite model and are also of independent interest.

Finally, a solution to the eigenvalue embedding problem, which is a special case of the finite-element model updating problem is proposed in case the model is symmetric positive semidefinite and the eigenvalues to be updated are real, generalizing earlier results in the symmetric definite case.

The new results in this dissertation, besides being useful contributions to the literature of linear and numerical linear algebra and vibration engineering, are expected to impact a wide variety of industries including automobile and aerospace.

NORTHERN ILLINOIS UNIVERSITY

STATE ESTIMATION AND FINITE-ELEMENT MODEL UPDATING FOR VIBRATING SYSTEMS

A DISSERTATION SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
DOCTOR OF PHILOSOPHY

MATHEMATICAL SCIENCES

BY
JOÃO CARVALHO

DEKALB, ILLINOIS
JULY 2002

Certification: In accordance with departmental and Graduate School policies, this dissertation is accepted in partial fulfillment of degree requirements.

Dissertation Director

Date

ACKNOWLEDGMENTS

I would like to thank my scientific advisor, professor Biswa Datta, for his support and guidance through all the steps of this work.

I would like to thank professor Karabi Datta for encouragement and insight received when developing this dissertation.

I would like to thank Dr. Vasile Sima for reading the pre-defense version of this dissertation and pointing me out several important typographic mistakes.

During the development if this dissertation, I have been supported by the *National Science Foundation* under grant ECS-0074411, to work as *Graduate Research Assistant* of professor Biswa Datta. During my PhD studies, I have been supported by the *Brazilian Capes* grant BEX1624/98-9 and by *Universidade Federal do Rio Grande do Sul* in Brazil. My gratitude to these three institutions is immense.

TABLE OF CONTENTS

Chapter	Page

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

One of the two widely-used approaches to represent a control system mathematically is the *state-space* approach. In this approach, a linear time-invariant continuous-time control system is represented by means of a system of first-order ordinary differential equations

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_0u(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1.0.1}$$

with initial condition $x(0) = x_0$, where A, B_0 and C are time-invariant matrices of appropriate dimensions, $x(t)$, $u(t)$ and $y(t)$ are the state, input and output vectors respectively.

Unfortunately, the natural mathematical models of many practical control systems do not arise in the above form; they can be systems of nonlinear partial differential equations or systems of matrix second-order differential equations.

A control system represented by a system of matrix second-order differential equations is of the form

$$\begin{aligned} M\ddot{q}(t) + D\dot{q}(t) + Kq(t) &= Bu(t) \\ y(t) &= C_1q(t) + C_2\dot{q}(t) \end{aligned} \tag{1.0.2}$$

with initial conditions $q(0) = q_0$ and $\dot{q}(0) = v_0$. Very often in vibration literature the matrices M, D and K are called, respectively, *mass*, *damping* and *stiffness* matrices. The vectors $q(t)$, $\dot{q}(t)$ and $\ddot{q}(t)$ are, respectively, the displacement, velocity and acceleration vectors.

The model (1.0.2) is sometimes conveniently denoted by the triplet (M, D, K). In a mathematical setup, very often the modal parameters of the model (1.0.2) are better described and studied in terms of the eigenvalues and eigenvectors of the associated **quadratic pencil**

$$P(\lambda) = \lambda^2 M + \lambda D + K. \tag{1.0.3}$$

If the matrices M, D and K are symmetric and furthermore M and K are positive semidefinite (to be denoted by $M \geq 0$, $K \geq 0$), the system (1.0.2) is called a symmetric positive semidefinite system and the associated pencil (1.0.3) is called a symmetric positive semidefinite pencil. If M is positive definite, then the model is called a symmetric positive definite model.

A standard first-order state-space representation of (1.0.2) is given by (1.0.1), where

$$\begin{aligned} A &= \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D \end{bmatrix}, \; B_0 = \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix}, \; C = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \\ x(t) &= \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}, \; x_0 = \begin{bmatrix} q_0 \\ v_0 \end{bmatrix}. \end{aligned} \tag{1.0.4}$$

Because of the obvious computational disadvantages of the representation using

(1.0.4), very often the system (1.0.2) is represented as

$$\begin{aligned} \begin{bmatrix} 0 & M \\ M & D \end{bmatrix} \begin{bmatrix} \ddot{q}(t) \\ \dot{q}(t) \end{bmatrix} &= \begin{bmatrix} M & \\ & -K \end{bmatrix} \begin{bmatrix} \dot{q}(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} C_2 & C_1 \end{bmatrix} \begin{bmatrix} \dot{q}(t) \\ q(t) \end{bmatrix} \end{aligned} \tag{1.0.5}$$

with proper initial conditions. There are, however, other first-order representations of the system (1.0.2) similar to (1.0.5) in the literature [27]. Note that the system (1.0.5) is symmetric but not necessarily positive semidefinite.

The system (1.0.5) has the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + B_0 u(t) \\ y &= Cx(t) \end{aligned} \tag{1.0.6}$$

and is usually referred to as a *generalized state-space* system or *descriptor* system. Descriptor systems arise, for example, in modeling interconnected systems, constrained mechanical systems and economic processes [49, 50, 61]. In those applications, frequently the matrix E is singular.

An obvious way to solve a control problem associated with the system (1.0.2) is to transform it to a first-order system of the form (1.0.1) or (1.0.6) and then apply a well-established technique for a first-order system available for that problem. The reduction to the system (1.0.6) should be preferred over (1.0.1) since it preserves the symmetry of the system matrices. Unfortunately, however, the control techniques for descriptor systems are not very well-established.

This dissertation deals with two important practical problems associated with the second-order model (1.0.2):

- the State Estimation problem

- the Finite-element Model updating problem.

1.1 State Estimation problem

The State Estimation problem for the system (1.0.1) is the problem of estimating the state vector $x(t)$ from the knowledge of the matrices A, B, C , the output vector $y(t)$ and the input vector $u(t)$. This problem arises from the fact that in many practical instances the state vector $x(t)$ is not completely known. On the other hand, there are important control problems such as

- State Feedback Stabilization
- Pole-assignment by State Feedback
- H_2 and H_∞ Control Problems

that require the knowledge of the full vector $x(t)$ (see [27, 47] for description and applications of the problems mentioned above). Thus the components of the state vector that are not available from measurement have to be estimated.

There are two distinct but related approaches for state-estimation:

- State Estimation via Eigenvalue Assignment
- State Estimation via Sylvester-observer equation.

The **Eigenvalue Assignment approach** is based on the solution of an eigenvalue assignment problem by state feedback. Specifically, given the observable pair (A, C), a feedback matrix K_E is sought such that the matrix $A - K_E C$ is stable. The estimate $\hat{x}(t)$ of $x(t)$ is then found solving the system of differential equations

$$\frac{d\hat{x}}{dt}(t) = (A - K_E C)\hat{x}(t) + K_E y(t) + Bu(t) \tag{1.1.7}$$

with any initial condition $\hat{x}(0) = x_0$, where $y(t)$ is the output vector and $u(t)$ is the input vector. It can be shown [27] that in this case the error vector $e(t) = \hat{x}(t) - x(t) \to 0$ as $t \to \infty$.

The **Sylvester-observer equation approach** is based on numerically solving the matrix equation

$$XA - FX = GC \tag{1.1.8}$$

which is a variation of the well known Sylvester equation $AX - XB = C$. Equation (1.1.8) is called *Sylvester-observer* equation. The term "Sylvester-observer equation" was coined by Datta in [19]. In this equation, A and C are known and X, F and G are to be found.

If F is a stable matrix (that is, all the eigenvalues of F have negative real parts), then it can be shown that $e(t) = z(t) - Xx(t) \to 0$ as $t \to \infty$, where $z(t)$ satisfies

$$\dot{z}(t) = Fz(t) + Gy(t) + XBu(t) \tag{1.1.9}$$

for any initial condition $z(0) = z_0$.

The existing numerical methods for solving the Sylvester-observer equation include the recursive method by Van Dooren [59], its block generalization by Carvalho and Datta [10], the SVD-based algorithm by Datta and Sarkissian [23], the large-scale algorithms by Datta and Saad [18] and Calvetti and Reichel [8] and the parallel algorithm by Bischof, Datta and Purkayastha [6].

1.2 Finite-Element Model Updating problem

The Finite-element Model Updating (FEMU) problem is the problem of updating a finite-element generated second-order model (M, D, K) described by (1.0.2), using modal data acquired from a physical vibration test, so that innacurate modeling assumptions can be corrected in a new updated model $(\tilde{M}, \tilde{D}, \tilde{K})$.

The need to solve the finite-element model updating problem arises from the fact that very often natural frequencies and mode shapes (eigenvalues and eigenvectors) of a finite element model (M, D, K) do not match very well with experimentally measured frequencies and mode shapes obtained from a real-life vibrating structure. Thus, a vibration engineer needs to update the theoretical finite element model to ensure its validity for future use. A complete description of these techniques can be found in [36].

The problem can be mathematically defined as follows:

Given a symmetric positive semidefinite model (M, D, K) with a set $\{\lambda_k, x_k\}$, $k = 1, \ldots, m$ of eigenvalues and corresponding eigenvectors, and a measured set $\{\sigma_k, y_k\}$, $k = 1, \ldots, m$ of natural frequencies and correspondent mode shapes, find an updated symmetric model $(\tilde{M}, \tilde{D}, \tilde{K})$ such that

- the subset $\{\lambda_k, x_k\}, k = 1, \ldots, m$ is replaced by $\{\sigma_k, y_k\}, k = 1, \ldots, m$ as eigenvalues and corresponding eigenvectors of the new model $(\tilde{M}, \tilde{D}, \tilde{K})$
- the remaining subset of $2n - m$ eigenvalues and corresponding eigenvectors of the new model $(\tilde{M}, \tilde{D}, \tilde{K})$ are the same as those of (M, D, K).

The problem as defined above is extremely difficult to solve in general. Very frequently in model updating applications the first requirement above, which means that the updated model must reproduce the physical measurements, is kept, while the second requirement, which means that the modal parameters (eigenvalues and eigenvectors) not related with the physical measurement do not change, is dropped.

There exist some partial solutions to this important problem as well as solutions to some related problems. These include:

- Parametric Model Updating Methods [1, 36]
- Partial Eigenstructure Assignment Methods [24, 28, 43, 54]
- Direct Methods using Modal Data [29, 36]
- Eigenvalue Embedding Methods [11, 35].

1.3 Dissertation Outline

This dissertation has the follow outline:

Chapter 2 is on concepts and fundamental results on state feedback control and state estimation theory for standard first-order linear time-invariant systems. Orthogonal state-space reductions to controller and observer Hessenberg forms, which play key role to the development of the numerical methods of this dissertation, are described in the case of standard first-order linear systems.

Chapter 3 is on state estimation of standard first-order linear time-invariant systems; emphasis is given here to the numerical solution of the reduced-order Sylvester-observer equation. Besides briefly describing the existing methods, a new block algorithm for the Sylvester-observer equation is proposed here. A comparative study between this new algorithm and the existing ones is made, using the results of numerical experiments performed on several families of test matrices obtained from [40].

Chapter 4 is on state estimation of descriptor linear time-invariant systems. Both the theoretical results and the numerical algorithms obtained in this chapter are new. A theorem establishing the state-observer of a descriptor linear time-invariant systems is derived and computational algorithms for state-estimation of such systems are obtained. These algorithms generalize those for the ordinary state-space systems given in Chapter 3. Finally, a comparative study, similar to that in Chapter 3, is presented.

Chapter 5 is on state estimation of second-order vibrating systems. We propose a solution to the state estimation problem in a second-order vibrating system via linearization to a descriptor system. To this end, the Sylvester-observer matrix equation approach developed in Chapter 4 is used. Applications to real-life state feedback control systems are presented.

Chapter 6 is on finite-element model updating of matrix second-order systems. We propose a new method for the model updating problem of an undamped positive semidefinite second-order model. Our method is based on results on orthogonality relations between eigenvectors of a symmetric positive semidefinite pencil $P(\lambda) = \lambda^2 M + \lambda D + K$; these results are derived in this chapter. We also propose a method for model updating via the Eigenvalue Embedding problem, this method simultaneously embed r real eigenvalues using a single rank r symmetric matrix update.

1.4 Contributions of this Dissertation

Contributions, both theoretical and computational, have been made to four important topics in Control and vibration: State Estimation for standard state-space systems, State Estimation for generalized state-space systems (also known as descriptor systems), Finite-Element Model Updating and Eigenvalue Embedding problems.

- *State Estimation for Standard State-space Systems:*

  Here two new algorithms (Algorithm 3.2.2 and Algorithm 3.2.4) have been proposed. Algorithm 3.2.2 is a block generalization of the algorithm for solving the Sylvester-observer equation due to Van Dooren [59]. This new block algorithm is computationally more efficient than Van Dooren's algorithm and suitable for high-performance computing. Algorithm 3.2.4 is a parametric algorithm. It has an advantage that it does not require any reduction of the system matrices and is then suitable for large and sparse problems.

- *State Estimation for Descriptor Systems:*

  Here a theory of state-estimation, analogous to the theory of state-estimation for a standard state-space system, has been derived via a generalized Sylvester-observer equation. Then three new algorithms for the generalized Sylvester-observer equation (Algorithm 4.3.1, Algorithm 4.3.2 and Algorithm 4.3.3) have been developed.

  These new algorithms generalize the corresponding block algorithms for the standard Sylvester-observer equation.

- *Finite-Element Model Updating:*

  Here a new algorithm for model updating of a symmetric positive semidefinite undamped second-order model has been developed and possible extensions of this method to the symmetric positive semidefinite damped model have been suggested. The proposed algorithm has the advantage, when compared to other existing algorithms, that it works with a positive semidefinite model and does not require any model reduction or expansion of the mode shapes.

  This new algorithm is based on several new orthogonality relations, derived in this dissertation, between the eigenvectors of a symmetric positive semidefinite damped model. These relations generalize the earlier relations due to Datta, Elhay and Ram [21] for the symmetric positive definite model.

  These new orthogonality relations, besides their key role in the development of our algorithms for model updating, are of independent theoretical interests and are believed to be important contributions to the literature of linear algebra.

- *Eigenvalue Embedding:*

  Here a new method has been developed to embed a real eigenvalue in a symmetric positive semidefinite model while keeping the other eigenvalues and eigenvectors of the original model unchanged. An extension of this method has been made to simultaneous embedding of several real eigenvalues. These results generalize earlier results of Carvalho, Datta, Lin and Wang [11].

- *High-performance Implementation:*

  Algorithms developed in this dissertation have been implemented, using high-performance software packages LAPACK and SLICOT, on several benchmark test problems. The results of these experiments have been displayed graphically. FORTRAN routines, whenever needed, have been written.

- *Numerical Experiments:*

  Comparisons of different methods, with respect to efficiency (both flop-count and cpu-time) and other numerical attributes such as residual, conditioning and others, have been made. Finally, conclusions have been drawn based on our observations.

1.5 Notations

The following notations are used throughout this dissertation:

n	dimension or number of degrees of freedom of the state-space models
t	independent time variable
$x(t)$	state vector for the state-space models, $x(t) \in \mathbb{R}^n$
m	number of input variables of the state-space models
r	number of output variables of the state-space models

s	dimension of the observer in the context of state estimation
$u(t)$	input vector , $u(t) \in \mathbb{R}^m$
$y(t)$	output vector, $y(t) \in \mathbb{R}^r$
A	$n \times n$ system matrix for first-order state-space systems
E	$n \times n$ system matrix for first-order descriptor systems
M	mass matrix for the second-order systems, $M = M^T \in \mathbb{R}^{n\times n}$
D	damping matrix for the second-order systems, $D \in \mathbb{R}^{n\times n}$
B	input or control matrix for the models to be considered, $B \in \mathbb{R}^{n\times m}$
K	feedback gain matrix in the context of feedback control $(K \in \mathbb{R}^{m\times n})$
	the stiffness matrix for second-order systems , $K = K^T \in \mathbb{R}^{n\times n}$
Λ	the matrix of eigenvalues of the state-space models
X	the $s \times n$ matrix solution of the Sylvester-observer equation
	the eigenvector matrix of state-space models
Λ_1	the $m \times m$ matrix of the eigenvalues that are meant to change
X_1	the $n \times m$ matrix of the eigenvectors that are meant to change
Σ_1	the $m \times m$ matrix of the measured natural frequencies
Y_1	the $n \times m$ matrix of the measured mode shapes
Λ_2	the $m \times m$ matrix of the unmeasured eigenvalues
X_2	the $n \times m$ matrix of the unmeasured eigenvectors
(E, A)	the pair of system matrices of the descriptor system
	$E\dot{x}(t) = Ax(t) + Bu(t)$
(M, D, K)	the system matrices of the symmetric second-order model
	$M\ddot{x}(t) + D\dot{x}(t) + Kx(t) = Bu(t)$
$\Omega(M)$	spectrum of the matrix M
$rank(M)$	numerical rank of the matrix M

CHAPTER 2

Preliminaries on Control and State Estimation

In this chapter we present the mathematical foundations of State Feedback Control and State Estimation problems.

In Section 2.1, we describe the interconnection between the State Feedback Control and the State Estimation problems.

In Section 2.2, we describe the main mathematical results concerning State Estimation.

In Section 2.3, we describe the orthogonal reduction to controller and observer Hessenberg forms, which are important computational tools in control applications.

2.1 The Interplay Between Feedback Control and State Estimation

Consider the dynamical system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t). \end{aligned} \tag{2.1.1}$$

Here $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector and $y(t) \in \mathbb{R}^r$ is the output vector.

It is often desirable to modify the behavior of the system to achieve certain desired objectives. An effective way to do so is the use of state feedback control. Thus, if the input vector $u(t)$, which is also regarded as a control vector, is chosen

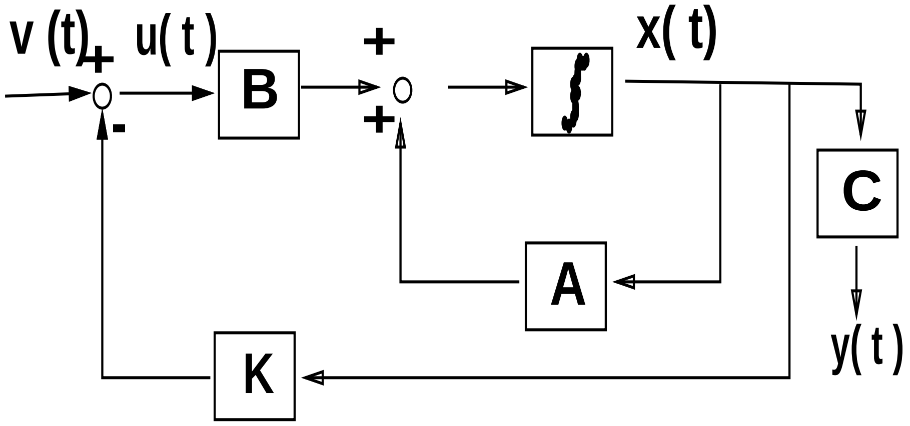

Figure 2.1: Diagram of State Feedback Design.

as

$$u(t) = v(t) - Kx(t), \tag{2.1.2}$$

where $v(t)$ is the exogenous signal (frequently $v(t) \equiv 0$ is assumed in practical problems) then the *open-loop* system (2.1.1) becomes

$$\begin{aligned} \dot{x}(t) &= (A - BK)x(t) + Bv(t) \\ y(t) &= Cx(t). \end{aligned} \tag{2.1.3}$$

The system (2.1.3) is called the *closed-loop* system. The matrix K is called the *feedback* matrix. The desired responses of this closed-loop system can be achieved by choosing the feedback matrix K appropriately. The feedback design of this type is called *state feedback* design and is illustrated in Figure 2.1.

The system (2.1.1) is said to be *controllable* if, starting from any initial state $x(0)$, the system can be driven to any final state $x(t_1)$ in some finite time t_1, choosing the input variable $u(t), 0 \leq t \leq t_1$, appropriately.

The behavior of the system (2.1.1) is controlled by the eigenvalues of the matrix A. Thus, the behavior of the closed-loop system (2.1.3) can be determined by choosing the matrix K such that the eigenvalues of the matrix $A - BK$ are placed at some desired locations of the complex plane. The choices of locations depend on the desired objectives.

It is well-known [27, 14] that there always exist a matrix K such that the matrix $A-BK$ has a desired set of eigenvalues if and only if the system (2.1.1) is controllable.

The well-known **Kalman Criterion of Controllability** [27] states that the system (2.1.1) is controllable, or equivalently the pair of matrices (A, B) is controllable, if and only if

$$rank\left(\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}\right) = n. \tag{2.1.4}$$

The matrix in parenthesis is called the *controllability* matrix and denoted by C_{AB}.

A concept dual to controllability is *observability*. The system (2.1.1) is said to be *observable* if there exists $t_1 \geq 0$ such that the initial state $x(0)$ can be uniquely determined from the knowledge of $u(t)$ and $y(t)$, for $0 \leq t \leq t_1$. A criterion of observability, analogous to the controllability criterion (2.1.4), is:

$$rank\left(\begin{bmatrix} C \\ CA \\ \dots \\ CA^{n-1} \end{bmatrix}\right) = n. \tag{2.1.5}$$

The matrix in parenthesis is called the *observability* matrix and denoted by O_{AC}.

2.2 Observing the State of a Linear Control System

As we have seen in the last section, the knowledge of the state vector $x(t)$ is necessary to the design of a feedback control system. Unfortunately, in practice, the state variables are not often available for measurement, or only a part of the state vector can be measured. In such a situation, the state vector $x(t)$ must be estimated, either in full or in part, so that the control law (2.1.2) can be applied.

The idea of state estimation was originated by David Luenberger in 1964 [47]; further developments were presented in [48]. Some simplifications of Luenberger's results were proposed in [17] and a geometric insight to the problem of reduced-order observers was given in [45].

The basic idea is to construct an auxiliary system using the knowledge of the matrices A, B and C and the input and output vectors $u(t)$ and $y(t)$ from which an estimate can be computed. This auxiliary system is known as *observer*.

It was not until 1984 that a constructive proof on the existence of reduced order observers was published by Paul Van Dooren [59]. Van Dooren proposed a numerically viable algorithm for solving the reduced order Sylvester-observer equation (see below), based on the reduction of the observable pair (A, C) to the observer-Hessenberg form, developed earlier by Boley [7], Paige [53], and others.

The next theorem shows how to construct this observer system.

Theorem 2.1 : *Observer Theorem*

Assume that matrices A and F have no common eigenvalues and let X, F and G be such that

$$XA - FX = GC, \tag{2.2.6}$$

where F is a stable matrix, that is, all its eigenvalues have negative real parts.

Then the vector $z(t)$ defined by

$$\dot{z}(t) = Fz(t) + Gy(t) + XBu(t), \tag{2.2.7}$$

with any initial condition $z(0) = z_0$, is such that $z(t) - Xx(t) \to 0$ as $t \to \infty$.

Proof: Since $\Omega(A) \bigcap \Omega(F) = \emptyset$, equation (2.2.6) has a solution triple (X, F, G). From (2.2.6) and (2.2.7), we have

$$\dot{z}(t) - X\dot{x}(t) = Fz(t) - XAx(t) + GCx(t) + (XB - XB)u(t) = F[z(t) - Xx(t)]$$

and therefore

$$z(t) - Xx(t) = \exp(Ft)[z(0) - Xx(0)].$$

Since F is stable, the result follows for any initial conditions $x(0) = x_0$ and $z(0) = z_0$. $\square$

Let $X \in \mathbb{R}^{s \times n}$ satisfy (2.2.6) for a stable matrix $F \in \mathbb{R}^{s \times n}$ and some $G \in \mathbb{R}^{s \times r}$. We now show how the result $z(t) - Xx(t) \to 0$ can be used to derive an asymptotic approximation $\hat{x}(t)$ of the state vector $x(t)$ from the knowledge of the input vector $u(t)$ and the output vector $y(t)$.

In the case that the observer has the same size as the original system $(s = n)$, an estimate $\hat{x}(t)$ can be found by simply solving the linear system defined by

$$X\hat{x}(t) = z(t) \tag{2.2.8}$$

as long as its coefficient matrix X is nonsingular.

As observed in [47], there is some redundancy in this construction since the information that can be obtained by the direct observation $y(t) = Cx(t)$ is not being used. In order to remove this redundancy, a "Reduction of Dynamical Order" is necessary [47].

Since the output vector $y(t)$ has the dimension $r < n$, from the relations

$$z(t) = X\hat{x}(t) \tag{2.2.9}$$

$$y(t) = C\hat{x}(t), \tag{2.2.10}$$

where (2.2.9) corresponds to the asymptotic result given by Theorem 2.1, we obtain

$$\begin{bmatrix} z(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} X \\ C \end{bmatrix} \hat{x}(t). \tag{2.2.11}$$

Thus

$$\hat{x}(t) = \begin{bmatrix} X \\ C \end{bmatrix}^{-1} \begin{bmatrix} z(t) \\ y(t) \end{bmatrix}, \tag{2.2.12}$$

if the inverse of the coefficient matrix of (2.2.11) exists.

The next theorem, taken from [14], elucidates the role of controllability and observability in the nonsingularity of the coefficient matrices of (2.2.8) and (2.2.11).

The proof given here is adapted from [14], [27] and [56].

Theorem 2.2 : *Role of Controllability and Observability in State Estimation*

Let matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ be given such that C has full rank r. Suppose a solution triple (X, F, G) of (2.2.6) is found, where $X \in \mathbb{R}^{s\times n}$, $F \in \mathbb{R}^{s\times s}$ and $G \in \mathbb{R}^{s\times r}$.

Then necessary conditions for compatibility of the systems (2.2.8) and (2.2.11), (or equivalently, for the existence of a full-rank solution X of the Sylvester-observer equation (2.2.6)) are that (A, C) is observable and (F, G) is controllable.

Proof: We first observe that using (2.2.6) together with an induction argument allows us to conclude

$$XA^k - F^kX = D_k \ , \ k = 0, 1, 2, 3, \ldots \tag{2.2.13}$$

where

$$D_k = \begin{cases} 0 & , k = 0 \\ \sum_{i=0}^{k-1} F^iGCA^{k-i-1} & , k > 0. \end{cases} \tag{2.2.14}$$

Let $\alpha(\lambda) = \alpha_0\lambda^n + \alpha_1\lambda^{n-1} + \ldots + \alpha_n$, ($\alpha_0 = 1$) be the characteristic polynomial of the matrix A and let $\beta(\lambda) = \beta_0\lambda^s + \beta_1\lambda^{s-1} + \ldots + \beta_s$, ($\beta_0 = 1$) be the characteristic polynomial of the matrix F.

Let

$$W_k = \begin{bmatrix} G & FG & F^2G & \ldots & F^{k-1}G \end{bmatrix} \tag{2.2.15}$$

and observe that $W_s = C_{FG}$, the controllability matrix of (F, G).

Similarly, let

$$Z_k = \begin{bmatrix} C^T & A^TC^T & (A^T)^2C^T & \ldots & (A^T)^{k-1}C^T \end{bmatrix}^T \tag{2.2.16}$$

and observe that $Z_n = O_{AC}$, the observability matrix of (A, C).

Multiplying (2.2.13) by α_k and adding for $k = 0, 1, \ldots, n$ gives

$$X\alpha(A) - \alpha(F)X = W_nR_\alpha O_{AC} \tag{2.2.17}$$

where

$$R_\alpha = \begin{bmatrix} \alpha_{n-1} & \alpha_{n-2} & \ldots & \alpha_1 & 1 \\ \alpha_{n-2} & \alpha_{n-3} & \ldots & 1 & 0 \\ \ldots & \ldots & & \ldots & \ldots \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.2.18}$$

Clearly $rank(R_\alpha) = n$. Also $\alpha(A) = 0$ by the Cayley-Hamilton Theorem (see [20]).

Similarly, multiplying (2.2.13) by $1, \beta_1, \beta_2, \ldots, \beta_s$, respectively, and adding, gives

$$X\beta(A) - \beta(F)X = C_{FG}R_\beta Z_s \tag{2.2.19}$$

where

$$R_\beta = \begin{bmatrix} \beta_{s-1} & \beta_{s-2} & \dots & \beta_1 & 1 \\ \beta_{s-2} & \beta_{s-3} & \dots & 1 & 0 \\ \dots & \dots & & \dots & \dots \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.2.20}$$

Clearly $rank(R_\beta) = s$. Also $\beta(F) = 0$, again by the Cayley-Hamilton Theorem.

Let $\lambda_i, i = 1, \dots, n$ be the eigenvalues of the matrix A and let $\mu_i, i = 1, \dots, s$ be the eigenvalues of the matrix F. The eigenvalues of $\alpha(F)$ and $\beta(A)$ are given [27], respectively, by

$$\prod_{j=1}^{n}(\mu_i - \lambda_j) \quad \text{and} \quad \prod_{j=1}^{s}(\lambda_i - \mu_j) \ .$$

Since, by hypothesis, $\Omega(A) \bigcap \Omega(F) = \emptyset$, we conclude that both $\alpha(F)$ and $\beta(A)$ are nonsingular matrices.

Consider now two cases:

Case 1: Let $s = n$ and consider the coefficient matrix X of the (2.2.8). Since $\alpha(A) = 0$, $det(\alpha(F)) \neq 0$, $rank(R_\alpha) = n$ and $W_n = C_{FG}$, from (2.2.17) we have

$$rank(X) \leq rank(C_{FG}) rank(O_{AC}).$$

Clearly X is nonsingular only if $rank(C_{FG}) = rank(O_{AC}) = n$ and the result follows for $s = n$.

Case 2: Let $s = n - r$ and consider the coefficient matrix

$$P = \begin{bmatrix} X \\ C \end{bmatrix} \tag{2.2.21}$$

of the linear system (2.2.11). Again, since $\beta(F) = 0$, $det(\beta(A)) \neq 0$ and $rank(R_\beta) = s$, we have from (2.2.19)

$$rank(X) \leq rank(C_{FG}).$$

If (F, G) is not controllable, then this clearly shows that $rank(X) < s$ and therefore P will be singular. If (A, C) is not observable, there exists a nonzero vector w such that $O_{AC}w = 0$. However, from the definition of O_{AC}, it follows that $Cw = 0$. Since $\alpha(A) = 0$ and $det(\alpha(F)) \neq 0$, we can use (2.2.17) to write

$$P = \begin{bmatrix} -\alpha(F)^{-1} W_n R_\alpha O_{AC} \\ C \end{bmatrix}.$$

Therefore,

$$Pw = \begin{bmatrix} -\alpha(F)^{-1} W_n R_\alpha O_{AC} w \\ Cw \end{bmatrix} = 0$$

which again implies that P is singular.

□

It can be shown [56] that the converse of the above theorem is also true in the case of single output systems, that is, in the case $r = 1$. In the general case $r \geq 1$, a set of necessary and sufficient conditions were derived recently in [31]. Unfortunately, these conditions are difficult to verify in a computational setting. Thus, each individual algorithm for solving the Sylvester-observer equation has to guarantee the compatibility of (2.2.8) or (2.2.11) from its construction of the matrix X. The same applies to the controllability of the constructed pair (F, G).

Figure 2.2 shows the Block Diagram of a feedback control system that uses state estimation to implement the control law (2.1.2).

2.3 Reduction to Controller and Observer Hessenberg forms

Very often in control applications a change of coordinates is introduced in the systems' description such that in the new set of coordinates controllability and ob-

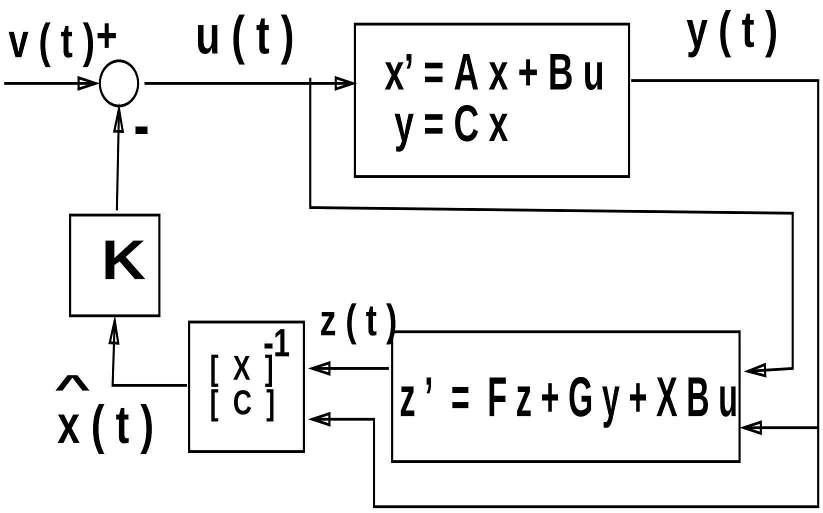

Figure 2.2: Diagram of State Feedback Control Design via Reduced-order State Estimation.

servability are better characterized. A change of coordinates is done by a proper linear transformation that can be represented by a matrix P.

One well-known transformation is the reduction of the system to the Controllable and Observable Canonical forms [27]. Unfortunately, the transforming matrices of this reduction process can be extremely ill-conditioned and, therefore, the process is not numerically viable.

As mentioned before, the concept of *orthogonal reduction to controller-Hessenberg form* brought a new perspective for numerical control applications. This is because the transformation, being orthogonal, is as well-conditioned as possible, and therefore yields to numerically stable computations.

The orthogonal reduction to controller-Hessenberg form can be described as follows: given matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ of a controllable linear system governed by (2.1.1), an orthogonal matrix $P \in \mathbb{R}^{n \times n}$ is computed such that

$$P^T A P = H = (H_{ij}) \tag{2.3.22}$$

$$P^T B = \tilde{B} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tag{2.3.23}$$

where

- H is block upper Hessenberg matrix with blocks H_{ij}, of dimension $n_i \times n_j, i, j = 1, \ldots, p$
- $n_1 + n_2 + \ldots + n_p = n$, and $n_1 \geq n_2 \geq \ldots \geq n_p$
- the sub-diagonal blocks $H_{j+1,j}, j = 1, \ldots, p-1$ and the matrix B_1 have full row rank.

A procedure for computing a controller-Hessenberg form, suggested in [60] and [53] and frequently referred to as the *Staircase* algorithm, is numerically stable and gives a numerically effective test of controllability [27] .

This procedure is presented below. It requires testing if certain matrices have full rank; which can be done either via QR decomposition with partial pivoting or using singular value decomposition (SVD) [38].

Algorithm 2.3 : *Staircase Algorithm for Controller-Hessenberg Form*

Input: Matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$.

Output: Matrices $P \in \mathbb{R}^{n \times n}$, $H \in \mathbb{R}^{n \times n}$ and $\tilde{B} \in \mathbb{R}^{n \times m}$.

Step 1: Apply the QR decomposition with column pivoting to the matrix B to compute matrices $Q_0 \in \mathbb{R}^{n \times n}$, $R_0 \in \mathbb{R}^{n \times m}$ and $E_0 \in \mathbb{R}^{m \times m}$ such that $BE_0 = Q_0 R_0$.

Step 2: Set $P_0 = Q_0$, $\tilde{B} = P_0^T B$ and $H_0 = P_0^T A P_0$.

Step 3: Set $n_1 = rank(R_0)$, $s_1 = n_1$, $s_0 = 0$ and $i = 1$.

Step 4: If $s_i < n$ do Steps 5 , 6 and 7, else do Step 8.

Step 5: Apply the QR decomposition with column pivoting to the sub-matrix $H_{i-1}(s_i + 1 : n, s_{i-1} + 1 : s_{i-1} + n_i)$ to compute matrices $Q_i \in \mathbb{R}^{(n-s_i)\times(n-s_i)}$, $R_i \in \mathbb{R}^{(n-s_i)\times n_i}$ and $E_i \in \mathbb{R}^{n_i \times n_i}$ such that

$$H_{i-1}(s_i + 1 : n, s_{i-1} + 1 : s_{i-1} + n_i)E_i = Q_i R_i.$$

Step 6: Compute

$$P_i = P_{i-1} \begin{bmatrix} I_{s_i} & \\ & Q_i \end{bmatrix}, \ H_i = \begin{bmatrix} I_{s_i} & \\ & Q_i^T \end{bmatrix} H_{i-1} \begin{bmatrix} I_{s_i} & \\ & Q_i \end{bmatrix}$$

where I_{s_i} denotes the identity matrix of order s_i. (Note that matrix Q_i need not be explicitly formed).

Step 7: Set $n_{i+1} = rank(R_i)$ and $s_{i+1} = s_i + n_{i+1}$. Update $i \leftarrow i + 1$ and go to Step 4.

Step 8 Set $P = P_{i-1}$, $H = H_{i-1}$ and return. $\square$

Breakdown: The algorithm above breaks down with $n_i = 0$ for some $i > 1$ if and only if (2.1.4) does not hold. Therefore, it can be reliably used to determine the controllability of the system (2.1.1).

Flop-count and Numerical Stability: If all the QR decompositions are done with implicit computation of the matrix Q, the algorithm above requires approximately $6n^3 + 12n^2m$ floating point operations and is backward stable [60].

The orthogonal reduction to observer-Hessenberg form can be described as follows: given matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ of an observable linear system (2.1.1), an orthogonal matrix $P \in \mathbb{R}^{n\times n}$ is computed such that

$$P^T AP = H = (H_{ij}) \tag{2.3.24}$$

$$CP = \tilde{C} = \begin{bmatrix} 0 & \dots & 0 & C_1 \end{bmatrix} \tag{2.3.25}$$

where

- H is a block upper Hessenberg matrix with blocks H_{ij}, of dimension $n_i \times n_j, i, j = 1, \dots, p$
- $n_1 + n_2 + \dots + n_p = n$, and $n_1 \leq n_2 \leq \dots \leq n_p$
- the sub-diagonal blocks $H_{j+1,j}, j = 1, \dots, p-1$ and the matrix C_1 have full column rank.

Since observability is a dual concept of controllability, the *Staircase* algorithm described above can be easily adapted to the reduction to the observer-Hessenberg form (2.3.24)-(2.3.25). It requires RQ decompositions with row pivoting, whose algorithmic description can be easily adapted from the QR case, presented in [38].

Algorithm 2.4 : *Staircase Reduction to the observer-Hessenberg Form*

Input: Matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$.

Output: Matrices $P \in \mathbb{R}^{n\times n}$, $H \in \mathbb{R}^{n\times n}$ and $\tilde{C} \in \mathbb{R}^{r\times n}$.

Step 1: Apply the RQ decomposition with row pivoting to the matrix C, computing matrices $Q_0 \in \mathbb{R}^{n\times n}$, $R_0 \in \mathbb{R}^{r\times n}$ and $E_0 \in \mathbb{R}^{r\times r}$ such that $E_0 C = R_0 Q_0$.

Step 2: Set $P_0 = Q_0^T$, $\tilde{C} = CP$ and $H_0 = P_0^T A P_0$.

Step 3: Set $n_1 = rank(R_0)$, $s_1 = n - n_1$, $s_0 = n$ and $i = 1$.

Step 4: If $s_i > 0$ do Steps 5 , 6 and 7, else do Step 8.

Step 5: Apply the RQ decomposition with row pivoting to the sub-matrix $H_{i-1}(s_i + 1 : s_i + n_i, 1 : s_i)$ to compute matrices $Q_i \in \mathbb{R}^{s_i\times s_i}$, $R_i \in \mathbb{R}^{n_i\times s_i}$ and $E_i \in \mathbb{R}^{n_i\times n_i}$ such that

$$E_i H_{i-1}(s_i + 1 : s_i + n_i, 1 : s_i) = R_i Q_i.$$

Step 6: Compute

$$P_i = P_{i-1} \begin{bmatrix} I_{n-s_i} & \\ & Q_i^T \end{bmatrix}, \; H_i = \begin{bmatrix} I_{n-s_i} & \\ & Q_i \end{bmatrix} H_{i-1} \begin{bmatrix} I_{n-s_i} & \\ & Q_i^T \end{bmatrix}$$

where I_{n-s_i} denotes the identity matrix of order $n - s_i$. (Note that matrix Q_i need not be explicitly formed).

Step 7: Set $n_{i+1} = rank(R_i)$ and $s_{i+1} = s_i - n_{i+1}$. Update $i \leftarrow i + 1$ and go to Step 4.

Step 8 Set $P = P_{i-1}$, $H = H_{i-1}$ and $p = i - 1$. Reverse the order of the vector $n_i, i = 1, \ldots, p$. $\square$

Breakdown: The algorithm breaks down with $n_i = 0$ for some $i > 1$ if and only if (2.1.5) does not hold. Therefore, it can be reliably used to determine the observability of the system (2.1.1).

Flop-count and Numerical Stability: Similarly to Algorithm 2.3, this algorithm requires approximately $6n^3 + 12n^2r$ floating point operations and is backward stable.

An important improvement in the staircase algorithm was proposed by Miminis and Paige [52]. The Miminis and Paige algorithm performs a two-stage sweep to bring certain structure to the sub-diagonal blocks of matrix H. The first stage is just the staircase reduction itself, yielding to a block upper Hessenberg matrix H with unstructured sub-diagonal blocks $H_{j+1,j}, j = 1, \ldots, p-1$. The second stage aims to bring the upper triangular structure to the sub-diagonal blocks of H by means of orthogonal transformations.

The two-stage Miminis and Paige reduction is presented here in the case of reduction of an observable pair (A, C). The algorithm computes orthogonal matrices $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{r\times r}$ such that (2.3.24) is satisfied with the sub-diagonal blocks of H being triangular and

$$Q^T C P = \tilde{C}. \tag{2.3.26}$$

Algorithm 2.5 : *The observer-Hessenberg Reduction with Triangular Structure*

Input: Matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$.

Output: Matrices $P \in \mathbb{R}^{n\times n}$, $H \in \mathbb{R}^{n\times n}$, $Q \in \mathbb{R}^{r\times r}$ and $\tilde{C} \in \mathbb{R}^{r\times n}$ satisfying (2.3.22) and (2.3.26).

Assumption: The pair of matrices (A, C) is observable.

First Stage: Do Steps 1 through 8 of Algorithm 2.4.

Second Stage:

Step 9: Set $u_0 = 0$, $u_1 = n_1$ and reset $H_0 = H$, $P_0 = P$.

Step 10: For $i = 1$ to $p-1$ do Steps 11, 12 and 13

Step 11: Apply the QR decomposition to the sub-matrix $H_{i-1}(u_i + 1 : u_i + n_{i+1}, u_{i-1} + 1 : u_{i-1} + n_i)$ to compute matrices $Q_i \in \mathbb{R}^{n_{i+1}\times n_{i+1}}$ and $R_i \in \mathbb{R}^{n_{i+1}\times n_i}$ such that $H_{i-1}(u_i + 1 : u_i + n_{i+1}, u_{i-1} + 1 : u_{i-1} + n_i) = Q_i R_i$.

Step 12: Compute

$$P_i = P_{i-1}\begin{bmatrix} I_{u_i} & & \\ & Q_i & \\ & & I \end{bmatrix}, \quad H_i = \begin{bmatrix} I_{u_i} & & \\ & Q_i^T & \\ & & I \end{bmatrix} H_{i-1} \begin{bmatrix} I_{u_i} & & \\ & Q_i & \\ & & I \end{bmatrix}$$

where I denotes the identity matrix of appropriate dimension.

Step 13: Set $u_{i+1} = u_i + n_{i+1}$ and go to Step 10.

Step 14 Reset $P = P_{p-1}$ and $H = H_{p-1}$. Apply the QR factorization to the matrix CP, computing matrices $Q_p \in \mathbb{R}^{r\times r}$ and $R_p \in \mathbb{R}^{r\times n}$ such that $CP = Q_p R_p$. Set $Q = Q_p$, $\tilde{C} = R_p$ and return. □

Flop-count: The second stage of this algorithm requires approximately $12n^2r$ floating-point operations. Thus, a total approximate flop-count of Algorithm 2.4, when combined with flop-count of Algorithm 2.3, is $6n^3 + 24n^2r$ floating-point operations.

CHAPTER 3

Reduced-order Observers for Standard First Order Systems

3.1 Introduction

In this chapter we present recent numerical methods for solving the Sylvester-observer equation

$$XA - FX = GC. \tag{3.1.1}$$

As we have seen in the last chapter, equation (3.1.1) is required to be solved in order to estimate the state $\hat{x}(t)$ of the linear system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{3.1.2}$$

where $A \in \mathbb{R}^{n\times n}, C \in \mathbb{R}^{r\times n}$ and $B \in \mathbb{R}^{n\times m}$, using the matrix equation approach.

We will assume that a reduced-order observer is sought (that is, the order of the observer is $s = n - r$).

Thus, given matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ such that (3.1.2) is observable and C has full rank r, and a self-conjugate vector $\mathcal{S} \in \mathbb{C}^s$ with the property $\Omega(A) \bigcap \mathcal{S} = \emptyset$, we would like to construct a solution triple (X, F, G) of (3.1.1), where $F \in \mathbb{R}^{s\times s}, G \in \mathbb{R}^{s\times r}$ and $X \in \mathbb{R}^{s\times n}$, satisfying the requirements:

(i) $\Omega(F) = \mathcal{S}$

(ii) the pair (F, G) is controllable

(iii) the matrix $\begin{bmatrix} X \\ C \end{bmatrix}$ is nonsingular.

Once the above requirements are met, an estimate $\hat{x}(t)$ of the state-vector $x(t)$ is obtained by solving

$$\begin{bmatrix} X \\ C \end{bmatrix} \hat{x}(t) = \begin{bmatrix} z(t) \\ y(t) \end{bmatrix}, \tag{3.1.3}$$

where the observer variable $z(t)$ satisfies the system of differential equations

$$\dot{z}(t) = Fz(t) + Gy(t) + XBu(t) \tag{3.1.4}$$

with any initial condition $z(0) = z_0$.

An efficient algorithm for computing a full rank solution X of the Sylvester-observer equation (3.1.1) was first proposed by Van Dooren [59]. The algorithm is based on the reduction to observer-Hessenberg form, described in Chapter 2.

In 2000, a block algorithm for solution of the reduced order Sylvester-observer equation (3.1.1) was published by Datta and Sarkissian [23]. The algorithm proposed in [23] is not based on the reduction to observer-Hessenberg form, but instead singular value decompositions are used in conjunction with the solutions of small Sylvester equations of the form $YA - TY = D$ where A, T, D are given and Y is to be computed. A computationally efficient algorithm for the construction of *functional observers* is also presented in [23]. The construction of a functional observer allows one to implement the control law (2.1.2) without explicitly computing an estimate of the state vector $x(t)$.

In 2001, a block generalization of Van Dooren's algorithm, based on BLAS-3 computations, was published by Carvalho and Datta [10]. The proposed algorithm was shown to be able to fully take advantage of the block nature of the reduction to observer-Hessenberg form.

In this chapter, we present the following algorithms for solving the Sylvester-observer equation (3.1.1):

(i) The recursive algorithm due to Van Dooren (Algorithm 3.2.1)

(ii) The block generalization of the recursive algorithm (Algorithm 3.2.2)

(iii) The SVD algorithm by Datta and Sarkissian (Algorithm 3.2.3)

(iv) The parametric QR algorithm (Algorithm 3.2.4)

3.2 Numerical Solutions of Sylvester-observer Equation

3.2.1 Observer-Hessenberg Method

As said before, this method, due to Van Dooren [59], is based on the reduction to observer-Hessenberg form, that is, orthonormal matrix $P \in \mathbb{R}^{n\times n}$ is computed such that

$$P^TAP = H \tag{3.2.5}$$

$$CP = \tilde{C} = \begin{bmatrix} 0 & \dots & 0 & C_1 \end{bmatrix} \tag{3.2.6}$$

where $H \in \mathbb{R}^{n\times n}$ is block upper Hessenberg and $C_1 \in \mathbb{R}^{r\times r}$ is a nonsingular matrix.

Post-multiplying (3.1.1) by P, we obtain

$$XAP - FXP = GCP\ .$$

That is,

$$XPH - FXP = G\begin{bmatrix} 0 & \dots & 0 & C_1 \end{bmatrix}.$$

Set $Y = XP$. Then we have

$$YH - FY = G\begin{bmatrix} 0 & \dots & 0 & C_1 \end{bmatrix}. \tag{3.2.7}$$

This reduced equation is now solved for Y, F and G and X is recovered from Y as

$$X = YP^T. \tag{3.2.8}$$

Development of the Method:

Assume that Y and F have the form

$$Y = \begin{bmatrix} 1 & y_{12} & y_{13} & \dots & \dots & y_{1n} \\ & 1 & y_{23} & \dots & \dots & y_{2n} \\ & & 1 & \dots & \dots & \dots \\ & & & 1 & y_{q,q+1} & y_{q,n} \end{bmatrix} \tag{3.2.9}$$

$$F = \begin{bmatrix} f_{11} & 0 & \dots & \dots & 0 \\ f_{21} & f_{22} & 0 & \dots & 0 \\ f_{31} & f_{32} & f_{33} & 0 & 0 \\ \dots & \dots & \dots & \dots & 0 \\ f_{q1} & \dots & \dots & \dots & f_{qq} \end{bmatrix}. \tag{3.2.10}$$

The diagonal entries $f_{ii}, i = 1, 2, \dots, n-r$ are now set so that $\Omega(F) = \mathcal{S}$ is satisfied, but the remaining entries of F are still to be found. If $\mathcal{S}$ has complex entries, complex arithmetic has to be used. It has been shown [59] that a solution Y of the form above always exists. Set $q = n - r$. Then equation (3.2.7) can be written

$$\begin{bmatrix} 1 & y_{12} & \dots & y_{1n} \\ & 1 & \dots & y_{2n} \\ & & 1 & y_{q,n} \end{bmatrix} H - \begin{bmatrix} f_{11} & \dots & 0 \\ f_{21} & f_{22} & 0 \\ f_{q1} & \dots & f_{qq} \end{bmatrix} \begin{bmatrix} 1 & y_{12} & \dots & y_{1n} \\ & 1 & \dots & y_{2n} \\ & & 1 & y_{q,n} \end{bmatrix} = \begin{bmatrix} g_1^T \\ \dots \\ g_q^T \end{bmatrix} C. \tag{3.2.11}$$

Set

$$y_i = \begin{bmatrix} y_{i,i+1} & \dots & y_{in} \end{bmatrix}^T, \; f_i = \begin{bmatrix} f_{i1} & \dots & f_{i,i-1} \end{bmatrix}^T. \tag{3.2.12}$$

Then equation (3.2.11) can be decomposed into the following systems of equations:

$$\begin{bmatrix} 1 & y_1^T \end{bmatrix} H - f_{11} \begin{bmatrix} 1 & y_1^T \end{bmatrix} = g_1^T \tilde{C} \quad (3.2.13)$$

$$\begin{bmatrix} 0 & \ldots & 0 & 1 & y_i^T \end{bmatrix} H - \begin{bmatrix} f_i^T & f_{ii} & 0 & \ldots & 0 \end{bmatrix} Y = g_i^T \tilde{C}, i = 2, \ldots, q. \quad (3.2.14)$$

Let u_i^T be the i-th row of the matrix $H_i = H - f_{ii} I$, $i = 1, 2, \ldots, q$. Then the equations (3.2.13) and (3.2.14) can be, respectively, written (using Matlab notation) as

$$\begin{bmatrix} y_1^T & g_1^T \end{bmatrix} \begin{bmatrix} H_1(2:n, 1:n) \\ -\tilde{C} \end{bmatrix} = -u_1^T \quad (3.2.15)$$

$$\begin{bmatrix} f_i^T & y_i^T & g_i^T \end{bmatrix} \begin{bmatrix} -Y(1:i-1, 1:n) \\ H_i(i+1:n, 1:n) \\ -\tilde{C} \end{bmatrix} = -u_i^T, i = 2, \ldots, q. \quad (3.2.16)$$

The non-square coefficient matrices of the linear systems of equations (3.2.15) and (3.2.16) above can be shown to have a left inverse, so that a solution can always be computed.

The following example illustrates the particular case when $n = 3$, $r = 1$.

Equation (3.2.11) now becomes:

$$\begin{bmatrix} 1 & y_{12} & y_{13} \\ 0 & 1 & y_{23} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \end{bmatrix} - \begin{bmatrix} f_{11} & 0 \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} 1 & y_{12} & y_{13} \\ 0 & 1 & y_{23} \end{bmatrix} = \quad (3.2.17)$$

$$= \begin{bmatrix} g_{11} \\ g_{21} \end{bmatrix} \begin{bmatrix} 0 & 0 & C_1 \end{bmatrix}.$$

Comparing the first row of the last equation gives

$$\begin{cases} y_{12}h_{21} = f_{11} - h_{11} \\ y_{12}(h_{22} - f_{11}) + y_{13}h_{32} = -h_{12} \\ y_{13}(h_{33} - f_{11}) + y_{12}h_{23} - g_{11}c_1 = -h_{13}. \end{cases}$$

Similarly, comparing the second row, we have

$$\begin{cases} -f_{21} = -h_{21} \\ y_{23}h_{32} - f_{21}y_{12} = -h_{22} + f_{22} \\ y_{23}(h_{33} - f_{22}) - f_{21}y_{13} - g_{21}c_1 = -h_{23} \end{cases}$$

The above equations can be written as

$$\begin{bmatrix} h_{21} & h_{22} - f_{11} & h_{23} \\ 0 & h_{32} & h_{33} - f_{11} \\ 0 & 0 & -c_1 \end{bmatrix}^T \begin{bmatrix} y_{12} \\ y_{13} \\ g_{11} \end{bmatrix} = \begin{bmatrix} f_{11} - h_{11} \\ -h_{12} \\ -h_{13} \end{bmatrix} \tag{3.2.18}$$

$$\begin{bmatrix} -1 & -y_{12} & -y_{13} \\ 0 & h_{32} & h_{33} - f_{22} \\ 0 & 0 & -c_1 \end{bmatrix}^T \begin{bmatrix} f_{21} \\ y_{23} \\ g_{21} \end{bmatrix} = \begin{bmatrix} -h_{21} \\ f_{22} - h_{22} \\ -h_{23} \end{bmatrix}. \tag{3.2.19}$$

Since the pair (A, C) is observable, the numbers h_{21}, h_{32} and c_1 are different from zero and therefore the coefficient matrices of the above two systems are nonsingular.

The above discussions lead to the following algorithm:

Algorithm 3.2.1 : *The Observer-Hessenberg Algorithm for Sylvester-observer Equation*

Input: Matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n\times n}$ of the system (3.1.2) and a self-conjugate set $\mathcal{S} \in \mathbb{C}^{n-r}$.

Output: Matrices $X \in \mathbb{R}^{(n-r)\times n}$, $F \in \mathbb{R}^{(n-r)\times(n-r)}$ and $G \in \mathbb{R}^{(n-r)\times r}$ such that $\Omega(F) = \mathcal{S}$ and $XA - FX = GC$.

Assumption: The system (3.1.2) is observable, C has full rank r and $\Omega(A) \bigcap \mathcal{S} = \emptyset$.

Step 1: Set $q = n - r$. Set the diagonal elements $f_{ii}, i = 1, \ldots, n-r$ of the matrix F such that $\Omega(F) = \mathcal{S}$.

Step 2: Transform the pair (A, C) to the observer-Hessenberg pair $(H, \tilde{C})$ to compute an orthogonal matrix $P \in \mathbb{R}^{n \times n}$ such that

$$P^T A P = H$$

$$C P = \tilde{C}$$

Step 3: For $i = 1, \ldots, q$ do Steps 4 and 5

Step 4: Set u_i^T to be the i-th row of the matrix $H_i = H - f_{ii} I$.

Step 5: If $i = 1$ find a solution (y_1, g_1) of the linear system

$$\begin{bmatrix} H_1(2:n, 1:n) \\ -\tilde{C} \end{bmatrix}^T \begin{bmatrix} y_1 \\ g_1 \end{bmatrix} = -u_1$$

and form the first row of the matrix Y. Otherwise, find a solution (f_i, y_i, g_i) of the linear system

$$\begin{bmatrix} -Y(1:i-1, 1:n) \\ H_i(i+1:n, 1:n) \\ -\tilde{C} \end{bmatrix}^T \begin{bmatrix} f_i \\ y_i \\ g_i \end{bmatrix} = -u_i$$

and form the $i - th$ row of the matrix Y.

Step 6: Form the matrices F and G from their rows and columns computed above.

Step 7: Compute $X = Y P^T$.

Flop-count:

- Staircase reduction : $6n^2 + 12n^2 r$ flops
- Computations of F, X, G : $2(n-r)rn^2$ (assuming that the eigenvalues of F are chosen to be real)

Therefore, this algorithm requires approximately $(2r+6)n^3 + 12n^2 r$ floating-point operations.

Example 3.2.1: The data of this example comes from the design of a reduced-order observer for the *Helicopter* problem discussed by *Doyle and Stein* [33] .

$$A = \begin{bmatrix} -0.02 & 0.005 & 2.4 & -32 \\ -0.14 & 0.44 & -1.3 & -3.0 \\ 0 & 0.018 & -1.6 & 1.2 \\ 0 & 0 & 1 & 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 57.3 \end{bmatrix}$$

$$\mathcal{S} = \left\{ \begin{array}{cc} -1 & -2 \end{array} \right\}$$

Here $n = 4$ and $r = 2$.

Step 1: $q = 4 - 2 = 2$.

Step 2: The observer-Hessenberg reduction:

$$H = \begin{bmatrix} -0.020 & 2.400 & 0.005 & -32.000 \\ 0 & -1.600 & 0.018 & 1.200 \\ -0.140 & -1.300 & 0.440 & -30.000 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 57.3 \end{bmatrix}$$

Step 3: $i = 1$. The linear system to solve is

$$\begin{bmatrix} 0 & -0.600 & 0.018 & 1.200 \\ -0.140 & -1.300 & 1.440 & -30.000 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -57.3 \end{bmatrix}^T \begin{bmatrix} y_{12} \\ y_{13} \\ y_{14} \\ g_{11} \\ g_{12} \end{bmatrix} = \begin{bmatrix} -0.980 \\ -2.400 \\ -0.005 \\ 32.000 \end{bmatrix}$$

and a solution is

$$\left\{ \begin{array}{ccccc} y_{12} & y_{13} & y_{14} & g_{11} & g_{12} \end{array} \right\} = \left\{ \begin{array}{ccccc} 0 & 7.000 & 6.700 & 10.085 & -4.1065 \end{array} \right\}$$

$i = 2$. The linear system to solve is

$$\begin{bmatrix} -1.000 & 0 & -7.000 & -6.700 \\ -0.140 & -1.300 & 2.440 & -30.000 \\ 0 & 1.000 & 0 & 2.000 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -57.3 \end{bmatrix}^T \begin{bmatrix} f_{21} \\ y_{23} \\ y_{24} \\ g_{21} \\ g_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.400 \\ -0.018 \\ -1.200 \end{bmatrix}$$

and a solution is

$$\left\{ \begin{array}{ccccc} f_{21} & y_{23} & y_{24} & g_{21} & g_{22} \end{array} \right\} = \left\{ \begin{array}{ccccc} 0.0007 & -0.0053 & -0.4068 & 0 & 0.0094 \end{array} \right\}$$

Step 6: The matrices Y, F and G are

$$Y = \begin{bmatrix} 1 & 0 & 7 & 6.7 \\ 0 & 1 & -0.0053 & -0.4068 \end{bmatrix}, F = \begin{bmatrix} -1 & 0 \\ 0.0007 & -2 \end{bmatrix}$$

$$G = \begin{bmatrix} 10.085 & -4.1065 \\ 0 & -0.0094 \end{bmatrix}$$

Step 7: The solution X is recovered as

$$X = \begin{bmatrix} 1 & 7 & 0 & 6.7 \\ 0 & -0.0053 & 1 & -0.4068 \end{bmatrix}.$$

Verification: $\|XA - FX - GC\|_F = 1.6325 \times 10^{-14}$ and

$$\Omega(F) = \left\{ \begin{array}{cc} -2.0000, & -1.0000 \end{array} \right\}.$$

3.2.2 Block Observer-Hessenberg Method

The Block Observer-Hessenberg method is a block generalization of the scalar recursive algorithm presented in last subsection.

The version presented here is basically the one published by Carvalho and Datta [9], with the difference that the reduction to observer-Hessenberg form is now done via the *Staircase* algorithm instead of the Block Arnoldi method. A similar method was also proposed in [55].

Assume that matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ and a self-conjugate vector $\mathcal{S} \in \mathbb{C}^{n-r}$ are given such that the system (3.1.2) is observable, C has full rank and $\Omega(A) \bigcap \mathcal{S} = \emptyset$.

The observable pair of matrices (A, C) is first transformed to a observer-Hessenberg pair $(H, \bar{C})$ using the Miminis and Paige method (Algorithm 2.5). That is, orthogonal matrices P and Q are computed such that

$$P^T AP = H \tag{3.2.20}$$

$$Q^T CP = \tilde{C} = \left[\begin{array}{cccc} 0 & \dots & 0 & C_1 \end{array}\right], \tag{3.2.21}$$

where $H = (H_{ij})$ is a block upper Hessenberg matrix with blocks $H_{ij} \in \mathbb{R}^{n_i\times n_j}$, $i, j = 1, 2, \dots, p$ and the upper trapezoidal subdiagonal blocks $H_{i+1,i}, i = 1, \dots, p-1$ have full column rank. The matrix C_1 is upper triangular with full column rank.

Post-multiplying the Sylvester-observer equation (3.1.1) by P and noting that $QQ^T = I$, we obtain

$$XAP - FXP = GQQ^T CP\ .$$

Using equations (3.2.20) and (3.2.21), the last equation becomes

$$XPH - FXP = GQ \left[\begin{array}{cccc} 0 & \dots & 0 & C_1 \end{array}\right]$$

or

$$YH - FY = \tilde{G} \left[\begin{array}{cccc} 0 & \dots & 0 & C_1 \end{array}\right] \tag{3.2.22}$$

where $Y = XP$ and $\tilde{G} = GQ$.

The reduced equation (3.2.22) is now solved for matrices Y, F and $\tilde{G}$ such that $\Omega(F) = \mathcal{S}$, and the matrix X is then recovered from Y as

$$X = YP^T. \tag{3.2.23}$$

The matrix G is then recovered from $\tilde{G}$ as

$$G = \tilde{G}Q^T. \tag{3.2.24}$$

Development of the Method for Solving (3.2.22):

Let

$$H = \begin{bmatrix} H_{11} & H_{12} & \dots & H_{1p} \\ H_{21} & H_{22} & \dots & H_{2p} \\ & H_{32} & \dots & H_{3p} \\ & & H_{p,p-1} & H_{pp} \end{bmatrix} \tag{3.2.25}$$

$$\tilde{C} = \begin{bmatrix} 0 & 0 & \dots & 0 & C_1 \end{bmatrix}. \tag{3.2.26}$$

Partition the matrices F, Y and $\tilde{G}$ conformably with the partition of H:

$$F = \begin{bmatrix} F_{11} & & \\ F_{21} & F_{22} & \\ F_{q1} & \dots & F_{qq} \end{bmatrix}, \; Y = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1p} \\ & Y_{22} & \dots & Y_{2p} \\ & & Y_{qq} & Y_{qp} \end{bmatrix}, \; \tilde{G} = \begin{bmatrix} \tilde{G}_1 \\ \dots \\ \tilde{G}_q \end{bmatrix}. \tag{3.2.27}$$

For simplicity, we assume that each diagonal block $Y_{ii} \in \mathbb{R}^{n_i \times n_i}$, $i = 1, \dots, p$, is an identity matrix of order n_i. However, other choices are also possible.

Since matrix F is required to have spectrum $\mathcal{S}$, we distribute the elements of $\mathcal{S}$ among the diagonal blocks of F in such a way that $\Omega(F) = \mathcal{S}$. A complex conjugate pair is distributed as a 2×2 matrix and a real one as a 1×1 scalar along the diagonal of F. Note that some compatibility between the structure of $\mathcal{S}$ and the size of the diagonal blocks of the matrix H is required for this to be possible.

Substituting (3.2.25)-(3.2.27) in (3.2.22) and comparing the corresponding blocks on left and right hand sides, we obtain:

$$\sum_{k=i}^{j+1} Y_{ik}H_{kj} - \sum_{k=1}^{min(i,j)} F_{ik}Y_{kj} = 0 \ , \ j = 1, 2, \ldots, p-1 \tag{3.2.28}$$

$$\sum_{k=i}^{p} Y_{ik}H_{kp} - \sum_{k=1}^{i} F_{ik}Y_{kp} = \tilde{G}_i C_1. \tag{3.2.29}$$

Again, from (3.2.22), we easily obtain

$$F_{ij} = 0 \text{ for } j = 1, 2, \ldots, i-2 \text{ and } F_{ij} = H_{ij} \text{ for } j = i-1.$$

Thus, equations (3.2.28) and (3.2.29) are reduced to

$$\sum_{k=i}^{j+1} Y_{ik}H_{kj} - \sum_{k=max(i-1,1)}^{i} F_{ik}Y_{kj} = 0 \ , \ j = i, i+1, \ldots, p-1 \tag{3.2.30}$$

$$\sum_{k=i}^{p} Y_{ik}H_{kp} - \sum_{k=max(i-1,1)}^{i} F_{ik}Y_{kp} = \tilde{G}_i C_1, \tag{3.2.31}$$

for $i = 1, 2, \ldots, q$. For a computational purpose we rewrite equation (3.2.30) as

$$\sum_{k=i}^{j} Y_{ik}H_{kj} + Y_{i,j+1}H_{j+1,j} - \sum_{k=max(i-1,1)}^{i} F_{ik}Y_{kj} = 0, j = i, i+1, \ldots, p-1 \tag{3.2.32}$$

that is, for $j = i, i+1, \ldots, p-1$, we have

$$Y_{i,j+1}H_{j+1,j} = -\sum_{k=i}^{j} Y_{ik}H_{kj} + \sum_{k=max(i-1,1)}^{i} F_{ik}Y_{kj}. \tag{3.2.33}$$

Equations (3.2.31) and (3.2.33) allow us to compute the off-diagonal blocks Y_{ij} of Y and the blocks $\tilde{G}_i$ of $\tilde{G}$ recursively. Equation (3.2.33) is an under-determined linear system which is guaranteed to have a solution since its coefficient matrix $H_{j+1,j}$ has full column rank.

The following example illustrates our development in the particular case when $p = 4$ and $q = 3$.

After setting $F_{21} = H_{21}$ and $F_{32} = H_{32}$, the equation $YH - FY = \tilde{G}C$ becomes

$$\begin{bmatrix} I & Y_{12} & Y_{13} & Y_{14} \\ & I & Y_{23} & Y_{24} \\ & & I & Y_{34} \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ & H_{32} & H_{33} & H_{34} \\ & & H_{43} & H_{44} \end{bmatrix} -$$

$$\begin{bmatrix} F_{11} & & \\ H_{21} & F_{22} & \\ & H_{32} & F_{33} \end{bmatrix} \begin{bmatrix} I & Y_{12} & Y_{13} & Y_{14} \\ & I & Y_{23} & Y_{24} \\ & & I & Y_{34} \end{bmatrix} = \begin{bmatrix} \tilde{G}_1 \\ \tilde{G}_2 \\ \tilde{G}_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & C_1 \end{bmatrix}.$$

This allows us to obtain the unknown blocks row-wise, as follows:

First row : $i = 1$

$H_{11} + Y_{12}H_{21} - F_{11} = 0$ (Solve for Y_{12})

$H_{12} + Y_{12}H_{22} + Y_{13}H_{32} - F_{11}Y_{12} = 0$ (Solve for Y_{13})

$H_{13} + Y_{12}H_{23} + Y_{13}H_{33} + Y_{14}H_{43} - F_{11}Y_{13} = 0$ (Solve for Y_{14})

$H_{14} + Y_{12}H_{24} + Y_{13}H_{34} + Y_{14}H_{44} - F_{11}Y_{14} = \tilde{G}_1C_1$ (Solve for $\tilde{G}_1$)

Second row : $i = 2$

$H_{22} + Y_{23}H_{32} - F_{21}Y_{12} - F_{22} = 0$ (Solve for Y_{23})

$H_{23} + Y_{23}H_{33} + Y_{24}H_{43} - F_{21}Y_{13} - F_{22}Y_{23} = 0$ (Solve for Y_{24})

$H_{24} + Y_{23}H_{34} + Y_{24}H_{44} - F_{21}Y_{14} - F_{22}Y_{24} = \tilde{G}_2C_1$ (Solve for $\tilde{G}_2$)

Third row : $i = 3$

$H_{33} + Y_{34}H_{43} - F_{32}Y_{23} - F_{33} = 0$ (Solve for Y_{34})

$H_{34} + Y_{34}H_{44} - F_{32}Y_{24} - F_{33}Y_{34} = \tilde{G}_3C_1$ (Solve for $\tilde{G}_3$).

$\square$

The above discussion leads to the following algorithm:

Algorithm 3.2.2: *Block Observer-Hessenberg Algorithm for Sylvester-observer Equation*

Input: Matrices $A \in R^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ of the linear system (3.1.2) and a self-conjugate set $\mathcal{S} \in \mathbb{C}^{n-r}$.

Output: Block matrices X,F and G, such that $\Omega(F) = \mathcal{S}$ and $XA - FX = GC$.

Assumption: The system (3.1.2) is observable, C has full rank and $\Omega(A) \bigcap \mathcal{S} = \emptyset$.

Step 1: Reduce (A, C) to the observer-Hessenberg form $(H, \tilde{C})$ using Algorithm 2.5, and implicitly save the orthogonal transforming matrices P and Q. Let $n_i, i = 1, \ldots, p$ be the dimension of the diagonal blocks of the matrix H.

Step 2: Partition matrices Y, F and $\tilde{G}$ in blocks conformably with the block structure of the matrix $H = (H_{ij})$, $i, j = 1, \ldots, p$. Set $q = p - 1$.

Step 3: Distribute the elements of $\mathcal{S}$ along the diagonal blocks $F_{ii}, i = 1, 2, \ldots q$ such that $\Omega(F) = \mathcal{S}$; the complex conjugate pairs as 2×2 blocks and the real ones as 1×1 scalars.

Step 4: Set $Y_{11} = I_{n_1\times n_1}$.

Step 5: For $i = 2, 3, \ldots, q$ set

$$F_{i,i-1} = H_{i,i-1} \ , \qquad Y_{ii} = I_{n_i\times n_i}$$

Step 6: For $i = 1, 2, \ldots, q$ do Steps 7 and 8

Step 7: For $j = i, i + 1, \ldots, p - 1$, solve the upper triangular system for $Y_{i,j+1}$:

$$Y_{i,j+1}H_{j+1,j} = -\sum_{k=i}^{j} Y_{ik}H_{kj} + \sum_{k=max(i-1,1)}^{i} F_{ik}Y_{kj}$$

Step 8: Solve the triangular system for $\tilde{G}_i$:

$$\tilde{G}_iC_1 = \sum_{k=i}^{p} Y_{ik}H_{kp} - \sum_{k=max(i-1,1)}^{i} F_{ik}Y_{kp}$$

Step 9: Form the block matrices Y, F and $\tilde{G}$ from the computed blocks.

Step 10: Compute $X = YP^T$ and $G = \tilde{G}Q^T$.

Remark: The computation in (2.2.11) requires the solution of the linear system

$$\begin{bmatrix} X \\ C \end{bmatrix} \hat{x}(t) = \begin{bmatrix} z(t) \\ y(t) \end{bmatrix}$$

which is equivalent to the nonsingular linear system:

$$\begin{bmatrix} Y \\ \tilde{C} \end{bmatrix} P^T \hat{x}(t) = \begin{bmatrix} z(t) \\ Q^T y(t) \end{bmatrix}. \tag{3.2.34}$$

Note that the coefficient matrix of the system above is upper triangular with nonzero diagonal entries by the construction of the matrices Y and $\tilde{C}$.

Flop count:

- Reduction to observer-Hessenberg form: $6n^3 + 24n^2 r$ flops,
- Computation of Y : $\dfrac{n^3}{3} + \dfrac{rn^2}{2}$ flops,
- Computation of X from Y and G from $\tilde{G}$: $n^3 + 4rn^2$ flops.

Therefore, this algorithm requires approximately $\dfrac{22n^3}{3} + \dfrac{57n^2 r}{2}$ floating-point operations.

Example 3.2.2: The same as in Example 3.2.1.

Step 1: The pair (A, C) is reduced to the observer-Hessenberg pair $(H, \bar{C})$:

$$P = \begin{bmatrix} -0.9943 & 0.1071 & 0. & 0. \\ 0. & 0. & 0. & 1. \\ 0.1071 & 0.9943 & 0. & 0. \\ 0. & 0. & -1. & 0. \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & -1. \\ -1. & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} -0.2936 & -2.5407 & -31.9445 & -0.0030 \\ -0.1407 & -1.3264 & 2.2332 & 0.0184 \\ -0.1071 & -0.9943 & 0. & 0. \\ 0. & -1.3075 & 3. & 0.4400 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} 0. & 0. & 57.3000 & 0. \\ 0. & 0. & 0. & -1. \end{bmatrix}$$

Step 2: $p = 2$, $q = 1$.

Step 3: The elements of $\mathcal{S}$ are distributed along the diagonal entries of F:

$$F_{11} = \begin{bmatrix} -1.0000 & 0 \\ 0 & -2.0000 \end{bmatrix}$$

Step 4:

$$Y_{11} = \begin{bmatrix} 1. & 0 \\ 0 & 1. \end{bmatrix}$$

Step 5: Since $q = 1$, this step is skipped.

Step 6: $i = 1$:

Step 7: $j = 1$. Solving the triangular system $Y_{12}H_{21} = -Y_{11}H_{11} + F_{11}Y_{11}$ for Y_{12} , we have

$$Y_{12} = \begin{bmatrix} 6.5972 & -6.9598 \\ -1.3139 & 1.5143 \end{bmatrix}$$

Step 8: Solving triangular system $\tilde{G}_1 C_1 = Y_{11}H_{12} + Y_{12}H_{22} - F_{11}Y_{12}$ for $\tilde{G}_1$, we have

$$\tilde{G}_1 = \begin{bmatrix} -0.8067 & 10.0251 \\ 0.0724 & -3.7134 \end{bmatrix}$$

Step 9 : Forming matrices Y, F and $\tilde{G}$ from the computed blocks, we have

$$Y = \begin{bmatrix} 1. & 0. & 6.5972 & -6.9598 \\ 0. & 1. & -1.3139 & 1.5143 \end{bmatrix} , \quad F = \begin{bmatrix} -1.0000 & 0 \\ 0 & -2.0000 \end{bmatrix} ,$$

$$\tilde{G} = \begin{bmatrix} -0.8067 & 10.0251 \\ 0.0724 & -3.7134 \end{bmatrix}$$

Step 10: Recovering $X = YP^T$ and $G = \tilde{G}Q^T$:

$$X = \begin{bmatrix} -0.9943 & -6.9598 & 0.1071 & -6.5972 \\ 0.1071 & 1.5143 & 0.9943 & 1.3139 \end{bmatrix}, \quad G = \begin{bmatrix} -10.0251 & 0.8067 \\ 3.7134 & -0.0724 \end{bmatrix}$$

Verification: $\|XA - FX - GC\|_F = 5.4336 \times 10^{-16}$ and

$$\Omega(F) = \left\{ -2.0000, \quad -1.0000 \right\}.$$

3.2.3 SVD Method

In this section we will present a method for solving the Sylvester-observer equation (3.1.1) due to Datta and Sarkissian [23] . The method is composed of repeated solutions of small Sylvester equations followed by singular value decompositions (SVD).

Development of the method:

In order to solve (3.1.1) in a block fashion, partition F as a $q \times q$ block matrix

$$F = \begin{bmatrix} F_{11} & F_{12} & & \\ & F_{22} & F_{23} & \\ & & \cdots & \cdots \\ & & & F_{qq} \end{bmatrix} \tag{3.2.35}$$

where diagonal blocks $F_{ii}, i = 1, \ldots, q$ of F are chosen to be matrices in real Schur form such that the requirement $\Omega(F) = \mathcal{S}$ is satisfied. Partitioning matrices X and G conformably to the partition of F, we have

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_q \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ G_q \end{bmatrix} \tag{3.2.36}$$

Using (3.2.35) and (3.2.36) in (3.1.1) and equating the corresponding blocks on each side, we obtain

$$X_iA - F_{ii}X_i = F_{i,i+1}X_{i+1}\ ,\ i = 1, \ldots, q-1 \tag{3.2.37}$$

$$X_qA - F_{qq}X_q = G_qC. \tag{3.2.38}$$

Let $\tilde{F}_{qq}$ be a matrix with a prescribed spectrum and let Y_q satisfy the Sylvester equation

$$Y_qA - \tilde{F}_{qq}Y_q = C. \tag{3.2.39}$$

Taking now the SVD of Y_q^T : $Y_q^T = U_q\Sigma_qV_q^T$, we have

$$V_q\Sigma_qU_q^TA - \tilde{F}_{qq}V_q\Sigma_qU_q^T = C.$$

If Y_q has full rank, we have

$$U_q^TA - (\Sigma_q^{-1}V_q^T\tilde{F}_{qq}V_q\Sigma_q)U_q^T = \Sigma_q^{-1}V_q^TC$$

meaning that we have a solution of (3.2.38) with

$$X_q = U_q^T\ ,\ F_{qq} = \Sigma_q^{-1}V_q^T\tilde{F}_{qq}V_q\Sigma_q\ ,\ G_q = \Sigma_q^{-1}V_q^T. \tag{3.2.40}$$

This strategy can be repeated for each $i = q-1, \ldots, 2, 1$ to obtain a solution of (3.2.37) as follows:

Choosing $\tilde{F}_{ii}$ with a prescribed spectrum, we solve the Sylvester equation

$$Y_iA - \tilde{F}_{ii}Y_i = X_{i+1} \tag{3.2.41}$$

for Y_i and compute the SVD of Y_i^T : $Y_i^T = U_i\Sigma_iV_i^T$ to obtain

$$U_i^T A - (\Sigma_i^{-1} V_i^T \tilde{F}_{ii} V_i \Sigma_i) U_i^T = \Sigma_i^{-1} V_i^T X_{i+1},$$

assuming that Y_i has full rank. A solution of (3.2.37) is then obtained with

$$X_i = U_i^T \ , \ F_{ii} = \Sigma_i^{-1} V_i^T \tilde{F}_{ii} V_i \Sigma_i \ , \ F_{i,i+1} = \Sigma_i^{-1} V_i^T. \tag{3.2.42}$$

The above discussion leads to the following algorithm:

Algorithm 3.2.3: *The SVD Algorithm for Sylvester-observer Equation*

Input: Matrices $A \in R^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ of the linear system (3.1.2) and a self-conjugate set $\mathcal{S} \in \mathbb{C}^{n-r}$.

Output: Block matrices X,F and G, such that $\Omega(F) = \mathcal{S}$ and $XA - FX = GC$.

Assumption: The system (3.1.2) is observable, C has full rank and $\Omega(A) \bigcap \mathcal{S} = \emptyset$.

Step 1: Distribute the elements of $\mathcal{S}$ in self-conjugate groups $S_i, i = 1, \ldots, q$ that will, later on, be assigned to the diagonal blocks of the matrix F.

Step 2: For $i = q, q-1, \ldots, 1$ do Steps 3 to 9

Step 3: If $i = q$ then set $J = C$; else set $J = X_{i+1}$.

Step 4: Define $\tilde{F}_{ii}$ to be a matrix in real Schur form such that $\Omega(\tilde{F}_{ii}) = S_i$.

Step 5: Solve, for Y_i, the Sylvester equation

$$Y_i A - \tilde{F}_{ii} Y_i = J$$

exploiting the structure of $\tilde{F}_{ii}$.

Step 6: Compute the economy size SVD of Y_i^T; that is, compute matrices U_i and V_i and a diagonal matrix Σ_i such that $Y_i^T = U_i \Sigma_i V_i^T$. If Σ_i is singular, go back to Step 4 and select another matrix $\tilde{F}_{ii}$.

Step 7: Compute $F_{ii} = \Sigma_i^{-1} V_i^T \tilde{F}_{ii} V_i \Sigma_i$ taking advantage of the structure of $\tilde{F}_{ii}$.

Step 8: If $i < q$ then set $F_{i,i+1} = \Sigma_i^{-1} V_i^T$; else set $G_q = \Sigma_q^{-1} V_q^T$.

Step 9: Set $X_i = U_i^T$.

Step 10: Form matrices X, F and G from their computed blocks.

Remarks:

1. The algorithm produces a solution X such that every $X_i, i = 1, \ldots, q$ is a matrix whose rows form an orthonormal set of vectors.

2. The algorithm does not require any reduction of the system matrices A and C. This feature is specially attractive when A is large and sparse, as long as we are able to exploit this structure in the solution of the subproblems in Step 5.

3. If matrix A is dense, then an orthogonal Hessenberg reduction $A \leftarrow P^T AP, C \leftarrow CP$ can be done to bring certain structure to the solution of the subproblems in Step 5. If (X_h, F, G) is the solution of this reduced problem, then $X = X_h P^T$ is the solution of the original problem.

4. The algorithm is rich in BLAS-3 computations and thus suitable for high-performance computing using LAPACK [2, 32].

Flop-count:

- Reduction of A to Hessenberg form with implicit computation of P (if needed): $\dfrac{10n^3}{3} + 4n^2 r - 4r^3/3$

- Solutions of the Sylvester equations in Step 5, using the Hessenberg-Schur method [39]: $10n^2 r + nr^2$ flops

- Step 6: $4n^2 r + 4nr^2 - 4r^3/3$

- Steps 7 and 8: $2r^3$ flops

- Recovery X from X_h: $4n^3$ flops.

Therefore, this algorithm requires approximately

$$\frac{64n^3}{3} + 9n^2 r$$

floating-point operations. The count for Step 5 assumes the worst-case scenario where the eigenvalues of $\tilde{F}_{ii}$ are all non-real.

Example 3.2.3: The same as in Example 3.2.1.

Step 1. $q = 1$ and $s_1 = \{\ -1.\ \ -2.\ \}$.

Steps 2 and 3. $i = 1$. Since $i = q$, $J = C$. Thus

$$J = \begin{bmatrix} 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 57.3000 \end{bmatrix}$$

Step 4.

$$\tilde{F}_{11} = \begin{bmatrix} -1. & 0. \\ 0. & -2. \end{bmatrix}$$

Step 5. Solving $Y_1 A - \tilde{F}_{11} Y_1 = J$ using the Hessenberg-Schur method [39] gives

$$Y_1 = \begin{bmatrix} 0.0948 & 0.6636 & 2.4386 & 2.0983 \\ -0.0708 & -1.0012 & 135.7360 & -55.4261 \end{bmatrix}$$

Step 6. From the economy size SVD of Y_1^T:

$$U_1 = \begin{bmatrix} -0.0005 & -0.0068 & 0.9258 & -0.3778 \\ 0.0324 & 0.2288 & 0.3691 & 0.9002 \end{bmatrix}^T$$

$$\Sigma_1 = \begin{bmatrix} 146.6269 & 0. \\ 0. & 2.9440 \end{bmatrix}, \quad V_1 = \begin{bmatrix} 0.0100 & 1. \\ 1. & -0.0100 \end{bmatrix}$$

Step 7. Computing $F_{11} = \Sigma_1^{-1} V_1^T \tilde{F}_{11} V_1 \Sigma_1$ gives

$$F_{11} = \begin{bmatrix} -1.9999 & 0.0002 \\ 0.4960 & -1.0001 \end{bmatrix}$$

Steps 8 and 9. Computing $G_1 = \Sigma_1^{-1} V_1'$ and $X_1 = U_1^T$ gives

$$G_1 = \begin{bmatrix} 0.0001 & 0.0068 \\ 0.3397 & -0.0034 \end{bmatrix} \quad , \quad X_1 = \begin{bmatrix} -0.0005 & -0.0068 & 0.9258 & -0.3778 \\ 0.0324 & 0.2288 & 0.3691 & 0.9002 \end{bmatrix}$$

Step 10. $X = X_1$, $F = F_{11}$ and $G = G_1$.

Verification: $\|XA - FX - GC\|_F = 4.6542 \times 10^{-15}$ and

$$\Omega(F) = \left\{ \ -2.0000, \quad -1.0000 \ \right\}.$$

3.2.4 Parametric QR Method

The Parametric QR method, proposed by Carvalho, Hong and Datta in [12], solves the Sylvester-observer equation (3.1.1) by imposing some structure on the right hand side of the equation. This means that, like in the SVD method presented in the last subsection, no reduction on the system matrices A and C is required.

Assume that matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ and a self-conjugate vector $\mathcal{S} \in \mathbb{C}^{n-r}$ are given such that the system (3.1.2) is observable, C has full rank and $\Omega(A) \bigcap \mathcal{S} = \emptyset$.

The method computes matrices X, F and R such that

$$XA - FX = R, \Omega(F) = \mathcal{S} \tag{3.2.43}$$

and for which

$$GC = R \tag{3.2.44}$$

can be solved for $G \in \mathbb{R}^{(n-r)\times r}$.

As the solution X is being computed, a Householder-QR [38] based strategy will reshape it so that at the end of the process X is a full-rank upper triangular matrix.

Other applications of (3.2.43) have been studied in [30].

Development of the method:

Let $C = R_c Q_c$, where $Q_c \in \mathbb{R}^{r\times n}$ and $R_c \in \mathbb{R}^{r\times r}$, be a thin RQ factorization [2] of the matrix C .

In order to solve (3.2.43) in a block fashion, the matrix F is partitioned as the $q \times q$ block matrix

$$F = \begin{bmatrix} F_{11} & & & \\ F_{21} & F_{22} & & \\ \dots & \dots & \dots & \\ F_{q1} & \dots & \dots & F_{qq} \end{bmatrix} \tag{3.2.45}$$

where diagonal blocks $F_{ii}, i = 1, \dots, q$ of F are chosen to be matrices in real Schur form such that the requirement $\Omega(F) = \mathcal{S}$ is satisfied. The matrix R is taken in the special partitioned form

$$R = \begin{bmatrix} N_1 \\ \dots \\ N_q \end{bmatrix} Q_c \tag{3.2.46}$$

, where $N_i \in \mathbb{R}^{n_i\times r}$, $i = 1, \dots, q$, and $n_1 + n_2 + \dots + n_q = s$. This special form ensures the existence of a solution $G \in \mathbb{R}^{s\times r}$ of (3.2.44) and implies

$$G_i R_c = N_i\ ,\ i = 1, 2, \dots, q \tag{3.2.47}$$

In particular, the choice $N_1 = I_r$ ensures that $rank(R) = rank(C) = r$.

Partitioning matrices X and G conformably to the partition of F gives

$$X = \begin{bmatrix} X_1 \\ \dots \\ X_q \end{bmatrix}\ ,\ G = \begin{bmatrix} G_1 \\ \dots \\ G_q \end{bmatrix}. \tag{3.2.48}$$

Substituting (3.2.46) and (3.2.48) into (3.2.43) and equating corresponding blocks on the right and left hand sides gives

$$X_1 A - F_{11} X_1 = N_1 Q_c, \tag{3.2.49}$$

$$X_i A - F_{ii} X_i = N_i Q_c + \sum_{j=1}^{i-1} F_{ij} X_j, i = 2, \ldots, q. \tag{3.2.50}$$

Therefore, as long as the elements of the given vector $\mathcal{S}$ can be successfully distributed in self-conjugate groups $S_i \in \mathbb{C}^{n_i}$, $i = 1, \ldots, q$, to be assigned as eigenvalues of the block matrices F_{ii}, $i = 1, \ldots, q$, the matrices X, F and G can be formed from their blocks computed recursively using (3.2.49) and (3.2.50).

After each sub-matrix X_i of the solution X has been computed, an orthogonal transformation is applied such that an upper triangular structure is created. This upper triangular structure can be exploited in solving the linear system (3.1.3).

Therefore, a step-by-step QR factorization will implicitly be performed on the matrix X. Assume that in the $i - th$ stage of this process all the blocks $X_j, j = 1, \ldots, i-1$ form an upper trapezoidal matrix and that the block X_i has just been computed. Placing X_i in the bottom of this matrix of X_j's gives an augmented matrix like

$$\begin{bmatrix} X_1 \\ \cdots \\ X_{i-1} \\ X_i \end{bmatrix} = \begin{bmatrix} * & * & * & * & * & * & * & * & * & * \\ & * & * & * & * & * & * & * & * & * \\ & & * & * & * & * & * & * & * & * \\ & & & & & & \cdots & & \cdots & \\ & & & & * & * & * & * & * & * \\ & & & & & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * \\ & & & \cdots & & & \cdots & & & \\ * & * & * & * & * & * & * & * & * & * \end{bmatrix}$$

which can be reduced to an upper trapezoidal form by orthogonal transformation applied to the its rows. This can be done, for instance, via left multiplication by a sequence of Householder matrices, here represented by a suitable orthogonal matrix Q_i^T.

The matrix equation

$$\begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix} A - F^i \begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix} = \begin{bmatrix} G_1 \\ \dots \\ G_{i-1} \\ G_i \end{bmatrix} C, \qquad (3.2.51)$$

where

$$F^i = \begin{bmatrix} F_{11} & \dots & F_{1,i-1} & 0 \\ \dots & \dots & \dots & 0 \\ F_{i-1,1} & \dots & F_{i-1,i-1} & 0 \\ F_{i1} & \dots & \dots & F_{ii} \end{bmatrix} \qquad (3.2.52)$$

is also updated at the same time as follows:

$$Q_i^T \begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix} A - Q_i^T F^i Q_i \cdot Q_i^T \begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix} = Q_i^T \begin{bmatrix} G_1 \\ \dots \\ G_{i-1} \\ G_i \end{bmatrix} C$$

and

$$\begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix}^{new} A - (F^i)^{new} \begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix}^{new} = \begin{bmatrix} G_1 \\ \dots \\ G_{i-1} \\ G_i \end{bmatrix}^{new} C \qquad (3.2.53)$$

meaning that it is possible to update the solution matrices, at every step of the orthogonal reduction, simply by computing

$$(F^i)^{new} = Q_i^T F^i Q_i \tag{3.2.54}$$

$$\begin{bmatrix} G_1 \\ \cdots \\ G_{i-1} \\ G_i \end{bmatrix}^{new} = Q_i^T \begin{bmatrix} G_1 \\ \cdots \\ G_{i-1} \\ G_i \end{bmatrix}. \tag{3.2.55}$$

The above discussion leads to the following algorithm, taken from [12]:

Algorithm 3.2.4: *The Parametric QR Algorithm for Sylvester-observer Equation*

Input: Matrices $A \in R^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ of the linear system (3.1.2) and a self-conjugate set $\mathcal{S} \in \mathbb{C}^{n-r}$.

Output: Block matrices X,F and G, such that $\Omega(F) = \mathcal{S}$ and $XA - FX = GC$.

Assumption: The system (3.1.2) is observable, C has full rank and $\Omega(A) \bigcap \mathcal{S} = \emptyset$.

Step 1: Set $s = n - r$, $\ell = r$ and $N_1 = I_{r\times r}$ and $G_1 = R_c^{-1}$.

Step 2: Compute the thin RQ factorization of C : $R_cQ_c = C$ where $Q_c \in \mathbb{R}^{r\times n}$ and $R_c \in \mathbb{R}^{r\times r}$.

Step 3: For $i = 1, 2, \ldots$ do Steps 4 to 10

Step 4: Set $S_i \in \mathbb{R}^{\ell}$ be a self-conjugate subset of the part of $\mathcal{S}$ that was not used yet.

Step 5: Set $F_{ii} \in \mathbb{R}^{\ell\times\ell}$ to be any matrix in upper real Schur form satisfying $\Omega(F_{ii}) = S_i$.

Step 6: Free parameter setup. If $i > 1$ set $N_i \in \mathbb{R}^{\ell\times n_i}$ and $F_{ij} \in \mathbb{R}^{\ell\times n_j}$, $j = 1, \ldots, i-1$ to be arbitrary matrices. Compute $G_i = N_iR_c^{-1}$.

Step 7: Solve the Sylvester equation

$$X_i A - F_{ii} X_i = N_i Q_c + \sum_{j=1}^{i-1} F_{ij} X_j \tag{3.2.56}$$

for $X_i \in \mathbb{R}^{\ell \times n}$.

Step 8: Set n_i to be the number of rows of X_i that are linearly independent of the rows of the matrix $\begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \end{bmatrix}$. If $n_i < \ell$ then set $\ell = n_i$, choose another set S_i from $\mathcal{S}$ and do Steps 5 to 8 again.

Step 9: Find, implicitly, an orthogonal matrix Q_i that reduces the matrix

$$\begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix}$$

to upper triangular form via left multiplication by Q_i^T. Then compute the matrix updates

$$\begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix} \leftarrow Q_i^T \begin{bmatrix} X_1 \\ \dots \\ X_{i-1} \\ X_i \end{bmatrix}, \quad \begin{bmatrix} G_1 \\ \dots \\ G_{i-1} \\ G_i \end{bmatrix} \leftarrow Q_i^T \begin{bmatrix} G_1 \\ \dots \\ G_{i-1} \\ G_i \end{bmatrix},$$
$$F^i \leftarrow Q_i^T F^i Q_i$$

where F^i satisfies (3.2.52).

Step 10: If $n_1 + \ldots + n_i = s$ then let $q = i$ and exit loop.

Step 11: Form matrices X, F and G from their computed blocks.

Remarks:

1. Some compatibility between the structure of the vector $\mathcal{S}$ and the parameters n_i, $i = 1, \ldots, q$ is required so that Step 4 is always possible to be accomplished.

2. The algorithm does not require reduction of the system matrices A and C. This feature is specially attractive when A is large and sparse, as long as we are able to exploit this structure in the solution of the subproblems in Step 7.

3. In Step 6, it is possible to exploit the freedom of assigning F_{ij} to facilitate the solution of the Sylvester equation in Step 7.

4. If matrix A is dense, an orthogonal similarity reduction $A \leftarrow P^T AP$, $C \leftarrow CP$, can be used so to bring Hessenberg structure to the matrix A. This allows Step 7 to be computed efficiently so that the whole algorithm requires $O(n^3)$ flops. If (X_h, F, G) is the solution of this reduced problem, then $X = X_h P^T$ is the solution of the original problem.

5. The algorithm is rich in BLAS-3 computations and thus is suitable for high-performance computing using LAPACK [2, 32].

Flop-count:

- Reduction of A to Hessenberg form with implicit computation of P (if needed): $\dfrac{10n^3}{3} + 4n^2r - 4r^3/3$ flops

- Step 2 (assuming explicit computation of Q_c [38]): $4n^2r - 2nr^2 + 4r^3/3$ flops

- Step 8 : $2n^2r(i-1) - r^3(i-1)^2$ flops

- Steps 9 and 10: $4\left[4nr^2i - 2r^3i^2 + 2nr^2 - 2r^3i\right]$ flops

- Step 7 (using the Hessenberg-Schur method [39]): $10n^2r + nr^2$ flops

- Recovery of X from X_h: $4n^3$ flops.

Therefore, this algorithm requires approximately

$$\frac{77n^3}{3} + \frac{29n^2r}{2}$$

floating-point operations. The count for Step 7 assumes the worst-case scenario where the eigenvalues of F_{ii} are all non-real.

Example 3.2.4: The same as in Example 3.2.1.

Step 1. $s = 2$, $\ell = 2$ and $N_1 = diag\left(\begin{array}{cc} 1, & 1 \end{array}\right)$.

Step 2.

$$R_c = \begin{bmatrix} 1. & 0. \\ 0. & -57.3000 \end{bmatrix}, \quad Q_c = \begin{bmatrix} 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & -1. \end{bmatrix}$$

Steps 3 and 4. $i = 1$, $S_1 = \{\ -1, \ \ -2\ \}$.

Step 5. Choosing $F_{11}(1,2) = 0.5$:

$$F_{11} = \begin{bmatrix} -1. & 0.5000 \\ 0. & -2. \end{bmatrix}$$

Step 6. Since $i = 1$, this step is skipped.

Step 7. Solving the Sylvester equation using Hessenberg-Schur method:

$$X_1 = \begin{bmatrix} 0.0947 & 0.6582 & 3.3574 & 1.4585 \\ 0.0012 & 0.0175 & -2.3689 & 0.9673 \end{bmatrix}$$

Step 8. $n_1 = 2$.

Step 9.

$$Q_1 = \begin{bmatrix} -0.9999 & -0.0131 \\ -0.0131 & 0.9999 \end{bmatrix},$$

$$X_1 = \begin{bmatrix} -0.0947 & -0.6584 & -3.3262 & -1.4710 \\ 0. & 0.0089 & -2.4125 & 0.9482 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -0.9999 & 0.0002 \\ -0.0131 & -0.0175 \end{bmatrix}, \quad F_{11} = \begin{bmatrix} -0.9936 & -0.4869 \\ 0.0131 & -2.0064 \end{bmatrix}$$

Step 10. Since $n_1 = s$ the algorithm is complete with $q = 1$.

Verification: $\|XA - FX - GC\|_F = 2.2988 \times 10^{-15}$ and

$$\Omega(F) = \left\{ -1.0000, \quad -2.0000 \right\}.$$

3.3 Comparison of Methods and Numerical Experiments

In this section, we compare the methods presented in the previous section for the solution of the Sylvester-observer equation for reduced-order state estimation.

The comparison is made not only with respect to traditional flop-count, but other numerical properties such as cpu time, residual accuracy, numerical ranking and condition numbers are also taken into consideration.

While comparing quantities like cpu-time and residual, Algorithm 3.2.1 (the observer-Hessenberg method due to Van Dooren) is not taken into consideration. The reason is as follows:

This method cannot be considered as either a scalar or a block algorithm since the reduction to observer-Hessenberg form has a block nature, while the recursive computations that yield to the solution matrices X, F and G, which consume most of the flop-count, do not involve BLAS-3 computations.

3.3.1 Comparison of Methods

The table below shows the floating-point operations count for the four algorithms derived in Section 3.2. Decimal numbers are used for easier comparison.

This table clearly shows that Algorithm 3.2.2 is the most efficient one when amount of floating-point operations is concerned.

Table 3.1: Flop Counts for the Algorithms in Section 3.2.

Algorithm	Subsection	Flops
Observer-Hessenberg	3.2.1	$(2r+6)n^3 + 12n^2r$
Block Observer-Hessenberg	3.2.2	$7.33n^3 + 28.50n^2r$
SVD-based	3.2.3	$21.33n^3 + 9n^2r$
Parametric QR	3.2.4	$25.66n^3 + 14.50n^2r$

In order to compare the block algorithms with respect to other numerical properties, the algorithms were tested on several benchmark problems with increasing size n taken from [40].

Three MATLAB-called Fortran 77 routines were written for this purpose, implementing the three block algorithms. They are as follows:

- **sylobbhr**: implements Algorithm 3.2.2
- **sylobsvd**: implements Algorithm 3.2.3
- **sylobpqr**: implements Algorithm 3.2.4

These routines were written with the goal of calling routines from BLAS, LAPACK and SLICOT libraries as much as possible.

Four *performance criteria* are considered to evaluate the behavior of these block algorithms:

1. *Normalized cpu-time*: The normalized cpu-time of a **sylob___** routine is defined to be the average cpu-time of a call to the routine divided by twice the cpu-time of a call to the BLAS-3 routine DGEMM for multiplying two square matrices. This measure has been used in [2]. Thus, the smaller the normalized cpu-time is, the more efficient the algorithm becomes.

2. *Residual*: In our context, the residual is the Frobenius norm of the matrix $R(\hat{X}) = \hat{X}A - F\hat{X} - GC$ of the computed solution $\hat{X}$ of the Sylvester-observer equation (3.1.1). If the problem is well-conditioned, then the smaller the residual is, the more accurate the solution is.

3. *Numerical Rank*: The numerical rank of the matrix X is defined to be the number of singular values of this matrix that are bigger than $\epsilon\|X\|_F$ (ϵ is the relative machine precision, $\epsilon = 2.22 \times 10^{-16}$ in our case). Since all the block algorithms under consideration are designed to compute a full-rank solution X to the Sylvester-observer equation (3.1.1), the numerical rank gives us a measure on how robust the full-rank solution X is.

4. *Condition Number*: The condition number of a matrix X has direct influence on the accuracy of the solution of an algebraic linear system with the system matrix X [27]. In our case, since the linear system (3.1.3), involving the matrix
$$\begin{bmatrix} X \\ C \end{bmatrix}$$
has to be solved in order to compute the estimate $\hat{x}(t)$, the 2-norm condition number of this matrix is measured so that we can decide which algorithm gives the most well-conditioned linear system (3.1.3) and thus ensures the most accurate computation of $\hat{x}(t)$.

3.3.2 Numerical Experiments

The numerical experiments were done using the families of matrices *Riemann* and *Pentoep* taken from [40].

Setup of the First Benchmark: For every positive integer n, the matrix $A = riemann(n)$ is defined by

$$A = B(2:n+1, 2:n+1), \text{ where } B_{ij} = \begin{cases} i-1 & \text{if } i \text{ divides } j \\ \text{-1} & \text{otherwise.} \end{cases}$$

For each $n = 4, 5, 6, \ldots$, the parameter r was taken as $r = \lfloor \sqrt{n} \rfloor$ and the matrix $C \in \mathbb{R}^{r \times n}$ was taken as a random matrix, generated by MATLAB's *randn* function.

It is known that each eigenvalue λ_i of $A = riemann(n)$ satisfies $|\lambda_i| \leq M - 1/M$, where $M = n + 1$. Also, $\lambda_i \in [i, i+1]$ with "at most $M - \sqrt{M}$ exceptions" [40]. All integers in the interval $(M/3, M/2]$ are eigenvalues.

Experimentally, for $n = 1, 2, \ldots, 300$, we have further observed that

$$|\lambda_i| < \sqrt{n} - 1$$

for all λ_i with negative real parts. This last upper bound helps in the selection of a stable set $\mathcal{S} \in \mathbb{C}^{n-r}$ apart from the set of eigenvalues $\lambda_i, i = 1, \ldots, n$ of A.

In the numerical testing with the *Riemann* family, for each n, the $n - r$ elements of the vector $\mathcal{S}$ were taken as follows:

- Roughly two thirds of them are conjugate pairs with negative real parts and each with modulus $\sqrt{n-r}$.

- Roughly one third of them are real negative and uniformly distributed in the interval $[-n, -\sqrt{n}]$.

This choice ensures that the set $\mathcal{S}$ is in the left side of the complex plane, and apart from the eigenvalues of A.

Figure 3.1 shows, for the particular case $n = 40$ and $r = \lfloor \sqrt{40} \rfloor = 6$, how these two sets are distributed in the complex plane.

Setup of the Second Benchmark: For every positive integer n, the matrix $A = pentoep(n)$, shown below, is both Toeplitz and pentadiagonal :

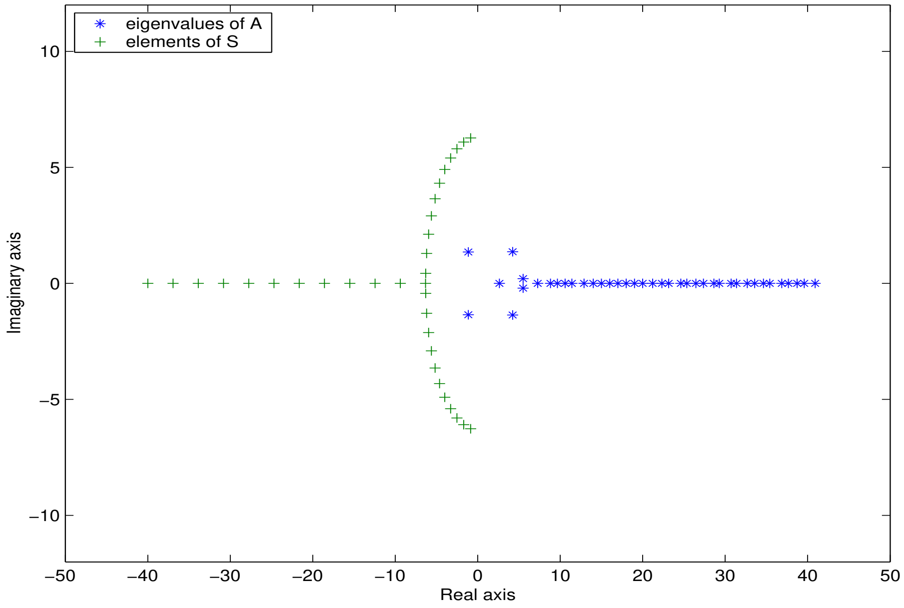

Figure 3.1: Spectrum of $A = riemann(40)$ and Chosen Elements of $\mathcal{S}$.

$$A = \begin{bmatrix} 0 & 10 & 1 & & \\ -10 & 0 & 10 & \dots & \\ 1 & -10 & 0 & \dots & 1 \\ & \dots & \dots & \dots & 10 \\ & & 1 & -10 & 0 \end{bmatrix}.$$

This matrix has eigenvalues $\lambda_i,\ i = 1, \dots, n$ lying approximately on the curve $2\cos(2t) + 20i\sin(t),\ -\pi \le t \le \pi$.

In the numerical testing with the *Pentoep* family, for each n, the $n - r$ elements of the vector $\mathcal{S}$ were taken as negative real numbers uniformly distributed in the interval $[-20, -1]$. We have chosen $\mathcal{S}$ with real entries in order to exploit both toeplitz and band structures in the solution of the Sylvester-observer equation using both Algorithm 3.2.3 and Algorithm 3.2.4. This choice ensures that the set $\mathcal{S}$ is in the left side of the complex plane, and apart from the eigenvalues of A.

Figure 3.2 shows, for the particular case $n = 40$ and $r = 6$, how these two sets

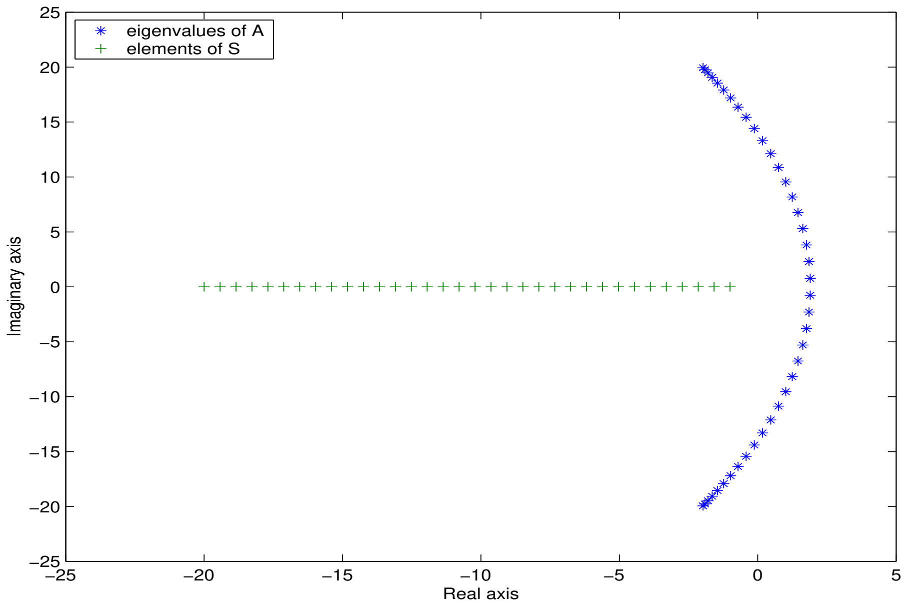

Figure 3.2: Spectrum of $A = pentoep(40)$ and Chosen Elements of $\mathcal{S}$.

are distributed in the complex plane.

First Benchmark: Figure 3.3 corresponds to the testing with the family of matrices *Riemann*. The numerical experiments were done in a Pentium II 400 MHz IBM-PC under Linux-Mandrake 8.0 and Matlab 6.0 .

From the plots, it can be seen that:

- The Block Observer-Hessenberg method requires significantly less cpu-time to achieve its goal.

- The Block Observer-Hessenberg method is also significantly less accurate than the other two methods. The Parametric QR method seems to be the most accurate one.

- The Parametric QR method has severe limitations when losses of the numerical rank of $[X; C]$ are concerned. The Block Observer-Hessenberg method and the SVD method have a comparable performance in this case.

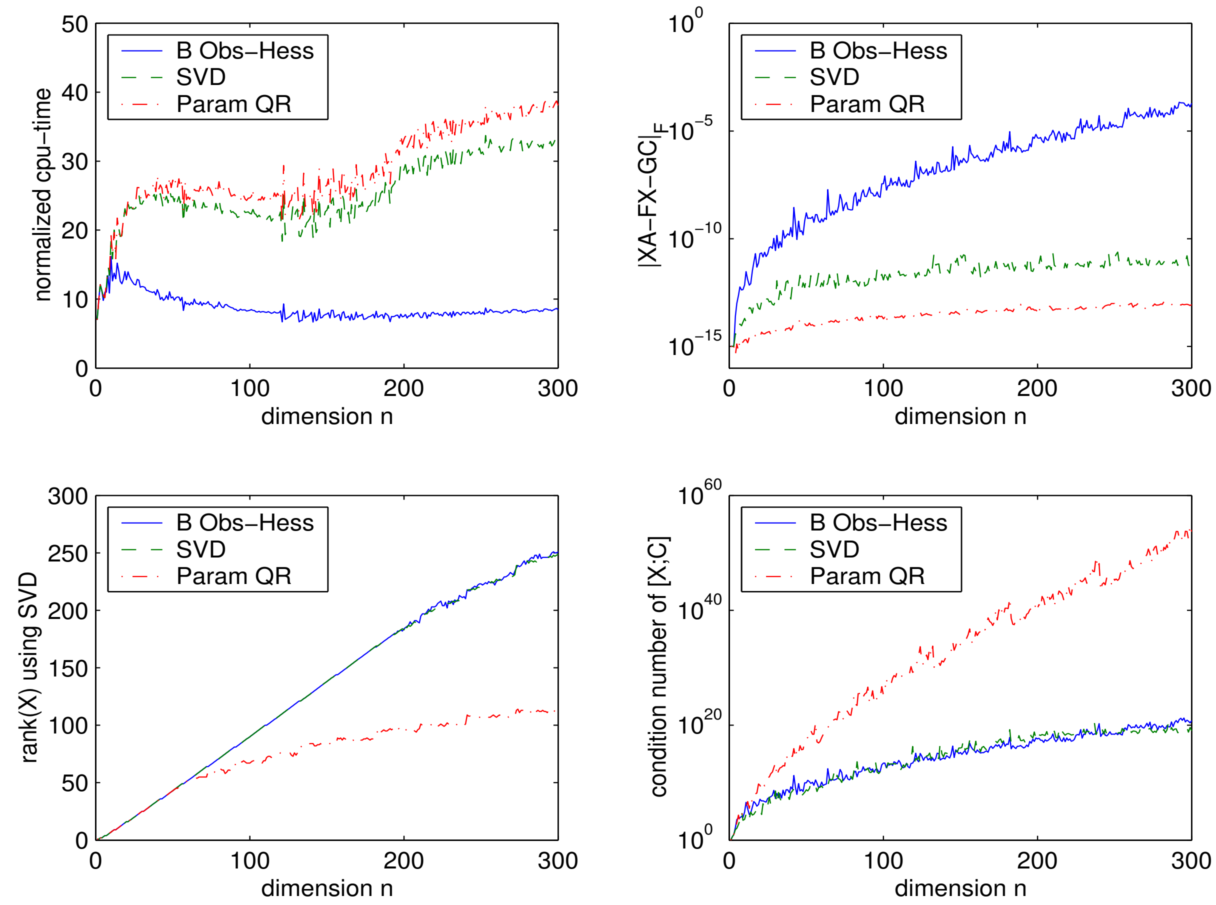

Figure 3.3: Numerical Experiment with Family *Riemann*.

Second Benchmark: Figure 3.4 and Figure 3.5 correspond to the testing with the family *Pentoep*. As said before, *Pentoep* matrices are both Toeplitz and pentadiagonal. In Figure 3.4, *Pentoep* matrices are considered to be Toeplitz and for the second (Figure 3.5), such matrices are considered to be pentadiagonal. Computations of the SVD and Parametric QR methods exploited these structures in each case.

The experiments confirm the results already obtained in the first benchmark. Notice that, by exploiting the band structure of matrix A, both the SVD and the Parametric QR algorithms give a normalized cpu-time competitive with the Block Observer-Hessenberg algorithms which, of course, is not designed to exploit either

Toeplitz or band symmetry.

Therefore, because of the inferior performance of the block observer-Hessenberg algorithm in all the residual accuracy tests, and that of the Parametric QR algorithm in all the condition number tests, we conclude that the SVD algorithm is the one that gives the best performance when state estimation is concerned. It must be also noted that a different setup of the free parameters of the Parametric QR algorithm can possibly improve its performance.

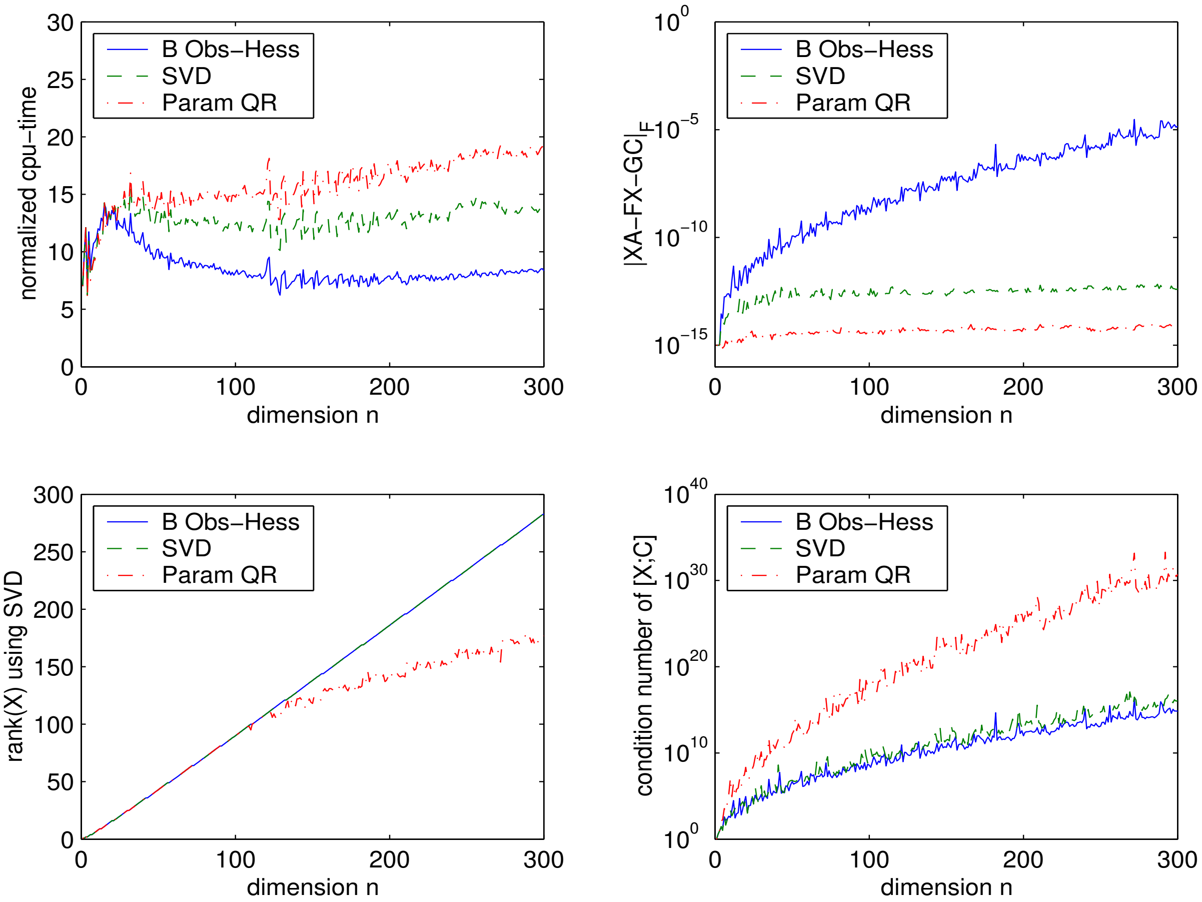

Figure 3.4: Numerical Experiment with Family *Pentoep*, Exploiting Toeplitz Structure.

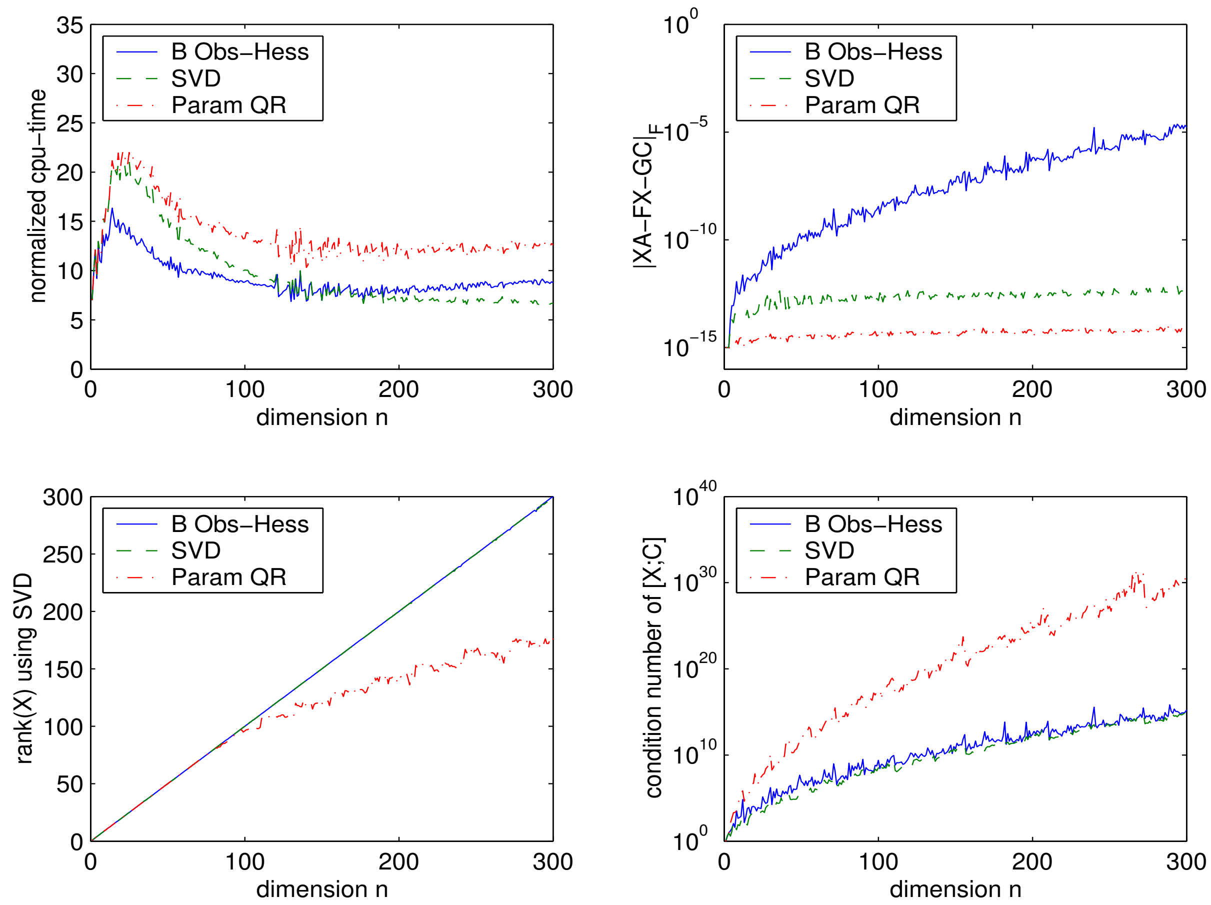

Figure 3.5: Numerical Experiment with Family *Pentoep*, Exploiting Pentadiagonal Structure.

CHAPTER 4

Reduced-order Observers for Descriptor Systems

4.1 Introduction

As said in the Introduction, many practical applications [14, 50] give rise to descriptor systems of the form:

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t). \end{aligned} \tag{4.1.1}$$

Here $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{r\times n}$ and $rank(E) \geq n - r$.

Once more, a state feedback control law

$$u(t) = v(t) - Kx(t) \tag{4.1.2}$$

is considered. Feeding this information back into (4.1.1) gives the closed-loop system

$$\begin{aligned} E\dot{x}(t) &= (A - BK)x(t) + Bv(t) \\ y(t) &= Cx(t). \end{aligned} \tag{4.1.3}$$

To design the state feedback control (4.1.2), of course, one needs to estimate the state vector $x(t)$.

The concepts of controllability and observability of the system (4.1.1) can be defined in terms of the associated standard state-space system. We give here two criteria presented in [46].

Lemma 4.1 *: Criterion for Controllability of a Descriptor System*

The descriptor system (4.1.1) is *controllable* if and only if

$$rank\left(\left[\begin{array}{cc} A-\mu E & B \end{array}\right]\right)=n \tag{4.1.4}$$

for each generalized eigenvalue μ of (A, E).

Lemma 4.2 *: Criterion for Observability of a Descriptor System*

The descriptor system (4.1.1) is *observable* if and only if

$$rank\left(\left[\begin{array}{c} A-\mu E \\ C \end{array}\right]\right)=n \tag{4.1.5}$$

for each generalized eigenvalue μ of (A, E).

Proof of the Lemmas: The proof of the results above follows from the controllability and observability criteria presented in [27] for standard state-space systems.

In this chapter, we consider the problem of state estimation for such a system. The organization of this chapter is as follows:

In Section 4.2, we develop a theory of state estimation for the system (4.1.1).

In Section 4.3, we present three new algorithms for solving the descriptor Sylvester-observer equation, to be presented in Section 4.2. These are:

- Hessenberg-Triangular Algorithm (Algorithm 4.3.1)
- SVD Algorithm (Algorithm 4.3.2)
- Parametric QR Algorithm (Algorithm 4.3.3).

These algorithms generalize, respectively, algorithms 3.2.2, 3.2.3 and 3.2.4, presented in Chapter 3.

In Section 4.4, similarly as in Section 3.3, we make a comparative study of the methods presented in Section 4.3, based on results of our numerical experiments with some selected benchmark matrices.

4.2 Observing the State of a Descriptor Control System

The analogue of the Sylvester-observer equation (3.1.1) for the descriptor system (4.1.1) is

$$XA - FXE = GC \tag{4.2.6}$$

where $E, A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ are given and the matrices $X \in \mathbb{R}^{s\times n}, F \in \mathbb{R}^{s\times s}$ and $G \in \mathbb{R}^{s\times r}$ are to be found. Equation (4.2.6) will be referred to, by analogy, as the *Descriptor Sylvester-observer equation.*

A solution triple (X, F, G) of (4.2.6) is guaranteed to exist (see [16]) if

$$\Omega(A, E) \bigcap \Omega(F) = \emptyset. \tag{4.2.7}$$

This requirement will be assumed for the rest of this chapter.

The Luenberger-observer for the system (4.1.1) is the same as that of the standard system (3.1.2). That is, it is given by a system of differential equations

$$\dot{z}(t) = Fz(t) + Gy(t) + XBu(t) \tag{4.2.8}$$

where any initial condition $z(0) = z_0$ can be taken.

If a full-order observer is constructed $(s = n)$, then an estimate $\hat{x}(t)$ to the state vector $x(t)$ is obtained by solving the system:

$$XE\hat{x}(t) = z(t), \tag{4.2.9}$$

once a solution $z(t)$ of (4.2.8) is computed.

However, if a reduced-order observer is constructed $(s = n - r)$, an estimate $\hat{x}(t)$ of the state vector $x(t)$ is obtained by solving the system:

$$\begin{bmatrix} XE \\ C \end{bmatrix} \hat{x}(t) = \begin{bmatrix} z(t) \\ y(t) \end{bmatrix}. \tag{4.2.10}$$

In the reduced-order case, in order to solve (4.2.10), the matrix product XE must have full rank $n-r$. Therefore, the solution matrix X of (4.2.6) must have full rank $n-r$ and we must assume $rank(E) \geq n-r$. In the full-order case, similarly, both the solution X and the system matrix E must have full rank n. We will return to this issue later.

The following theorem shows that the solution $z(t)$ of (4.2.8) asymptotically approximates a linear combination of the state vector $x(t)$ of (4.1.1) when $X \in \mathbb{R}^{s \times n}$ is the solution of (4.2.6) and matrix F is chosen to be stable (that is, all the eigenvalues have negative real parts).

Theorem 4.1 : *Reduced-order Observer Theorem for Descriptor systems.*

Suppose that

(i) the triple (X, F, G) satisfies (4.2.6)

(ii) the matrix F is chosen to be stable (all eigenvalues with negative real parts).

Then the system (4.2.8) is a state-observer of the system (4.1.1) in the sense that, for any initial condition $x(0) = x_0$ and $z(0) = z_0$, the error $e(t) = z(t) - XEx(t)$ approaches zero as time t increases.

Proof: Using (4.1.1), (4.2.6) and (4.2.8) , we can write

$$\dot{z}(t) - XE\dot{x}(t) = Fz(t) + GCx(t) + XBu(t) - XAx(t) - XBu(t) = Fz(t) - FXEx(t)$$

.

Therefore $\dot{e}(t) = Fe(t)$, which gives

$$e(t) = \exp(Ft)e(0) = \exp(Ft)[z(0) - XEx(0)]$$

and clearly $e(t) \to 0$ as $t \to \infty$, for any initial conditions $z(0) = z_0$ and $x(0) = x_0$, since F is stable by hypothesis.

□

As in Theorem 2.2, the next theorem shows that controllability and observability of the descriptor system (4.1.1) play a role in obtaining a full-rank solution X to the descriptor Sylvester-observer equation (4.2.6).

Theorem 4.2 : *Role of Controllability and Observability in State Estimation of Descriptor Systems*

Let the matrices $A, E \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{r \times n}$ be given such that C has full rank r. Suppose a solution triple (X_0, F_0, G_0) of (4.2.6) is found, where $X_0 \in \mathbb{R}^{s \times n}$, $F_0 \in \mathbb{R}^{s \times s}$ and $G_0 \in \mathbb{R}^{s \times r}$. Suppose that $\Omega(A, E) \bigcap \Omega(F_0) = \emptyset$.

Then the following statements hold:

(i) In the full-order case , if E is nonsingular, then the observability of the system (4.1.1) and the controllability of (F, G) are necessary conditions for the compatibility of the linear system (4.2.9) for $X = X_0$.

(ii) In the reduced-order case, if $rank(E) \geq n - r$, then the existence of $s = n - r$ observable states of the system (4.1.1) and the controllability of (F, G) are necessary conditions for the compatibility of the linear system (4.2.10) for $X = X_0$.

Proof: We note that, since $\Omega(A,E)\bigcap\Omega(F_0)=\emptyset$, X_0 is the unique solution of

$$XA - F_0XE = G_0C. \tag{4.2.11}$$

Generalizing the idea of orthogonal reduction of the pair (A,C) to the observer-Hessenberg form, as described in Chapter 2, it was proved by Varga in [62] that there exist orthogonal matrices $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times n}$ that transform (E,A,C) to its *Irreducible Generalized Observable State-space Realization* $(\hat{E},\hat{A},\hat{C})$:

$$\begin{aligned} U^TAV = \hat{A} &= \begin{bmatrix} A_2 & A_3 \\ & A_1 \end{bmatrix} \\ U^TEV = \hat{E} &= \begin{bmatrix} E_2 & E_3 \\ & E_1 \end{bmatrix} \\ CV = \hat{C} &= \begin{bmatrix} 0 & C_1 \end{bmatrix} \end{aligned} \tag{4.2.12}$$

where matrices $A_1, E_1 \in \mathbb{R}^{\bar{n}\times\bar{n}}$ and $C_1 \in \mathbb{R}^{r\times\bar{n}}$, $\bar{n}$ being the number of observable states of (4.1.1).

Similarly, let $Q \in \mathbb{R}^{s\times s}$ be an orthogonal matrix that transforms (F_0,G_0) to its *irreducible controller-Hessenberg form* $(\hat{F}_0,\hat{G}_0)$ [27]:

$$\begin{aligned} Q^TF_0Q = \hat{F}_0 &= \begin{bmatrix} F_1 & F_3 \\ & F_2 \end{bmatrix} \\ Q^TG_0 = \hat{G}_0 &= \begin{bmatrix} G_1 \\ 0 \end{bmatrix}, \end{aligned} \tag{4.2.13}$$

where matrices $F_1 \in \mathbb{R}^{\bar{s}\times\bar{s}}$ and $G_1 \in \mathbb{R}^{\bar{s}\times r}$. Here $\bar{s} = rank(C_{F_0G_0})$, where C_{F_0,G_0} is the controllability matrix of the pair (F_0,G_0), as defined in Section 2.1.

We remark that X_0 is a solution of (4.2.11) if and only if $Y_0 = Q^T X_0 U$ is a solution of

$$Y\hat{A} - \hat{F}_0 Y \hat{E} = \hat{G}_0 \hat{C}. \tag{4.2.14}$$

Using (4.2.12) and (4.2.13), equation (4.2.14) above can be shown to have the solution

$$Y_0 = \begin{bmatrix} 0 & Y_1 \\ 0 & 0 \end{bmatrix} \tag{4.2.15}$$

where $Y_1 \in \mathbb{R}^{\bar{s} \times \bar{n}}$ satisfies $Y_1 A_1 - F_1 Y_1 E_1 = G_1 C_1$.

Now, clearly

$$rank(X_0) = rank(Y_0) = rank(Y_1) \leq \min(\bar{s}, \bar{n}). \tag{4.2.16}$$

Consider now two cases:

(a) In the full-order case, since E is nonsingular by hypothesis, system (4.2.9) is compatible for $X = X_0$ only if X_0 has maximal rank n. Again, from (4.2.16), the matrix X_0 has maximal rank n only if $\bar{s} = \bar{n} = n$, which is equivalent to the observability of the system (4.1.1) and the controllability of (F_0, G_0). Thus the statement (a) follows.

(b) In the reduced-order case, since $rank(E) \geq s = n - r$ by hypothesis, the system (4.2.10) is compatible for $X = X_0$ only if X_0 has a maximal rank s. From (4.2.16), the matrix X_0 has maximal rank s only if $\bar{s} = s$ and $\bar{n} \geq s$, which is equivalent to the controllability of (F_0, G_0) and the existence of $n - r$ observable states of (4.1.1). Thus the statement (b) follows.

4.3 Numerical Solution of the Descriptor Sylvester-observer Equation

4.3.1 Hessenberg-triangular Method

The Hessenberg-triangular method is a generalization of the block observer-Hessenberg method presented in Subsection 3.2.2; it requires an orthogonal state-space reduction [62, 63], as stated below:

Let $A, E \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{r\times n}$ and $\mathcal{S} \in \mathbb{C}^s$ be such that (4.1.1) is observable, C has full rank r and $\Omega(A,E)\bigcap\mathcal{S} = \emptyset$. The matrices A and E are reduced to block Hessenberg-triangular form via orthogonal equivalence transformations. That is, orthogonal matrices $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times n}$ are computed such that

$$\begin{aligned} U^T AV &= H \\ U^T EV &= T \\ CV = \tilde{C} &= \begin{bmatrix} 0 & 0 & \dots & 0 & C_1 \end{bmatrix} \end{aligned} \tag{4.3.17}$$

where

- H is a block upper Hessenberg matrix with blocks $H_{ij} \in \mathbb{R}^{n_i\times n_j}; i, j = 1, \dots p$
- Sub-diagonal blocks $H_{i+1,i}$, $i = 1, \dots, p-1$ have full column rank
- $n_1 \le n_2 \le \dots \le n_p$ and $n_1 + n_2 + \dots + n_p = n$
- T is an upper triangular matrix
- C_1 has full column rank r.

An algorithmic description for the computation of the matrices U and V in (4.3.17) can be found in [62].

Using (4.3.17) in (4.2.6) , we obtain

$$XUU^TAV - FXUU^TEV = GCV \ .$$

Define $Y = XU$, then the above equation becomes

$$YH - FYT = \begin{bmatrix} 0 & 0 & \dots & 0 & GC_1 \end{bmatrix}. \tag{4.3.18}$$

The equation (4.3.18) is now solved, obtaining Y , F and G such that $\Omega(F) = \mathcal{S}$. Finally, the matrix X is recovered from Y as:

$$X = YU^T. \tag{4.3.19}$$

Development of the Method for Solving (4.3.18):

Let

$$H = \begin{bmatrix} H_{11} & H_{12} & \dots & H_{1p} \\ H_{21} & H_{22} & \dots & H_{2p} \\ & H_{32} & \dots & H_{3p} \\ & & H_{p,p-1} & H_{pp} \end{bmatrix}, \ T = \begin{bmatrix} T_{11} & T_{12} & \dots & T_{1p} \\ & T_{22} & \dots & T_{2p} \\ & & \ddots & \dots \\ & & & T_{pp} \end{bmatrix} \tag{4.3.20}$$

$$\tilde{C} = \begin{bmatrix} 0 & 0 & \dots & 0 & C_1 \end{bmatrix}$$

obtained by the orthogonal state-space reduction (4.3.17). The matrices F, Y and G are conformably partitioned as:

$$F = \begin{bmatrix} F_{11} & & & \\ F_{21} & F_{22} & & \\ & \ddots & \ddots & \\ & & F_{q,q-1} & F_{qq} \end{bmatrix}, \ Y = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1p} \\ & Y_{22} & \dots & Y_{2p} \\ & & Y_{qq} & Y_{qp} \end{bmatrix}, \ G = \begin{bmatrix} G_1 \\ \dots \\ G_q \end{bmatrix}. \tag{4.3.21}$$

For simplicity, we have assumed that the matrix F has a block lower bidiagonal structure, as shown above. We can also assume, for simplicity, that the matrix Y has identity diagonal blocks $Y_{ii} \in \mathbb{R}^{n_i \times n_i}$, $i = 1, \ldots, p$. However, other definitions are possible.

Since the matrix F is required to have a preassigned spectrum $\mathcal{S}$, we distribute the elements of $\mathcal{S}$ among the diagonal blocks of F, if possible, in such a way that $\Omega(F) = \mathcal{S}$. A complex conjugate pair is distributed as a 2×2 matrix and a real one as a 1×1 scalar on the diagonal of F. Note that some compatibility between the structure of $\mathcal{S}$ and the size of the diagonal blocks of the matrices H and T is required for this to be possible.

Substituting (4.3.20) and (4.3.21) into (4.3.18) and comparing corresponding blocks on left and right hand sides, we obtain, for $i = 1, \ldots, q$; $j = 1, \ldots, p$

$$\sum_{k=i}^{\min(j+1,p)} Y_{ik} H_{kj} - \sum_{k=\max(i-1,1)}^{i} F_{ik} \sum_{\ell=k}^{j} Y_{k\ell} T_{\ell j} = \delta_{jp} G_i C_1, \tag{4.3.22}$$

where $\delta_{jp} = 1$ if $j = p$, and zero otherwise.

After rearrangement, equation (4.3.22) can be broken down into three sets of equations, as follows:

- For $i = 2, \ldots, q$ and $j = i - 1$

$$F_{i,i-1} T_{i-1,i-1} = H_{i,i-1} \tag{4.3.23}$$

- For $i = 1, 2, \ldots, q$ and $j = i, \ldots, p - 1$

$$Y_{i,j+1} H_{j+1,j} = - \sum_{k=i}^{j} Y_{ik} H_{kj} + \sum_{k=\max(i-1,1)}^{i} F_{ik} \sum_{\ell=k}^{j} Y_{k\ell} T_{\ell j} \tag{4.3.24}$$

- For $i = 1, 2, \ldots, q$ and $j = p$

$$G_i C_1 = \sum_{k=i}^{p} Y_{ik} H_{kj} - \sum_{k=\max(i-1,1)}^{i} F_{ik} \sum_{\ell=k}^{p} Y_{k\ell} T_{\ell p}. \tag{4.3.25}$$

These three sets of equations now can be used to compute all the unknown blocks in a recursive fashion.

Equation (4.3.24) is an under-determined linear system which is guaranteed to have a solution since its coefficient matrix $H_{j+1,j}$ has full column rank.

The following example illustrates our development in the particular case $p = 3, q = 2$.

In this case, we have

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ & H_{32} & H_{33} \end{bmatrix}, \ T = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ & T_{22} & T_{23} \\ & & T_{33} \end{bmatrix}, \ \tilde{C} = \begin{bmatrix} 0 & 0 & C_1 \end{bmatrix}$$

and

$$Y = \begin{bmatrix} I & Y_{12} & Y_{13} \\ & I & Y_{23} \end{bmatrix}, \ F = \begin{bmatrix} F_{11} & \\ F_{21} & F_{22} \end{bmatrix}, \ G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}.$$

The unknown first-row blocks Y_{12}, Y_{13} of Y, F_{11} of F and G_1 of G can then be computed as follows:

$$\begin{aligned} Y_{12} H_{21} &= -H_{11} + F_{11} T_{11} \text{ (solve for } Y_{12}) \\ Y_{13} H_{32} &= -H_{12} - Y_{12} H_{22} + F_{11}(T_{12} + Y_{12} T_{22}) \text{ (solve for } Y_{13}) \\ G_1 C_1 &= H_{13} + Y_{12} H_{23} + Y_{13} H_{33} - F_{11}(T_{13} + Y_{12} T_{23} + Y_{13} T_{33}) \text{ (solve for } G_1). \end{aligned}$$

Similarly, for the second row:

$$\begin{aligned} F_{21} T_{11} &= H_{21} \text{ (solve for } F_{21}) \\ Y_{23} H_{32} &= -H_{22} + F_{21}(T_{12} + Y_{12} T_{22}) + F_{22} T_{22} \text{ (solve for } Y_{23}) \end{aligned}$$

$G_2C_1 = H_{23} + Y_{23}H_{33} - F_{21}(T_{13} + Y_{12}T_{23} + Y_{13}T_{33}) - F_{22}(T_{23} + Y_{23}T_{33})$ (solve for G_2).

The above discussion leads to the following algorithm:

Algorithm 4.3.1: Hessenberg-triangular Algorithm for Descriptor Sylvester-observer Equation

Input: Matrices $A, E \in R^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ of the linear system (4.1.1) and a self-conjugate set $\mathcal{S} \in \mathbb{C}^s$.

Output: Block matrices X,F and G, such that $\Omega(F) = \mathcal{S}$ and $XA - FXE = GC$.

Assumption: The system (4.1.1) is observable, C has full rank and $\Omega(A, E) \bigcap \mathcal{S} = \emptyset$. Also, either $s = n$ or $s = n - r$.

Step 1: Reduce (E, A, C) to the Hessenberg-triangular form $(T, H, \tilde{C})$. Let U and V be the orthogonal matrices of this reduction and let $n_i, i = 1, \ldots, p$ be the dimension of the diagonal blocks of the matrix H.

Step 2: Partition matrices Y, F and G in blocks conformably with the block structure of H. Set $q = p$ if $s = n$; otherwise set $q = p-1$. Let the matrix T be partitioned in blocks according to the partitioning of matrix H.

Step 3: Distribute the elements of $\mathcal{S}$ along the diagonal blocks $F_{ii}, i = 1, 2, \ldots q$ such that $\Omega(F) = \mathcal{S}$; the complex conjugate pairs as 2×2 blocks and the real ones as 1×1 scalars along the diagonal of the matrix F.

Step 4: Set $Y_{11} = I_{n_1\times n_1}$.

Step 5: For $i = 2, 3, \ldots, q$, set $Y_{ii} = I_{n_i\times n_i}$ and solve

$$F_{i,i-1}T_{i-1,i-1} = H_{i,i-1}$$

for $F_{i,i-1} \in \mathbb{R}^{n_i\times n_{i-1}}$.

Step 6: For $i = 1, 2, \ldots, q$ do Steps 7 and 8

Step 7: For $j = i, i + 1, \ldots, p - 1$, solve the linear system for $Y_{i,j+1}$:

$$Y_{i,j+1}H_{j+1,j} = -\sum_{k=i}^{j} Y_{ik}H_{kj} + \sum_{k=\max(i-1,1)}^{i} F_{ik}\sum_{\ell=k}^{j} Y_{k\ell}T_{\ell j}$$

Step 8: Solve the linear system for G_i:

$$G_iC_1 = \sum_{k=i}^{p} Y_{ik}H_{kj} - \sum_{k=\max(i-1,1)}^{i} F_{ik}\sum_{\ell=k}^{p} Y_{k\ell}T_{\ell p}$$

Step 9: Form the block matrices Y, F and G from their computed blocks.

Step 10: Compute $X = YU^T$.

Remark: Using (4.3.17), it can be seen that the computation in (4.2.10) is equivalent to the solution of the linear system

$$\begin{bmatrix} YT \\ \tilde{C} \end{bmatrix} V^T\hat{x}(t) = \begin{bmatrix} z(t) \\ y(t) \end{bmatrix}. \tag{4.3.26}$$

If $Q \in \mathbb{R}^{r\times r}$ is an orthogonal matrix such that Q^TC_1 is an upper triangular matrix, then (4.3.26) is again equivalent to

$$\begin{bmatrix} YT \\ Q^T\tilde{C} \end{bmatrix} V^T\hat{x}(t) = \begin{bmatrix} z(t) \\ Q^Ty(t) \end{bmatrix}, \tag{4.3.27}$$

which is a nonsingular upper triangular system.

Example 4.3.1: Consider solving the descriptor Sylvester-observer equation (4.2.6) with

$$E = \begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A = \begin{bmatrix} 0.45 & -0.04 & 0.42 & -0.09 & -0.36 & -0.48 \\ -0.27 & -0.48 & 0.24 & 0.39 & -0.30 & 0.25 \\ 0.11 & 0.32 & -0.32 & -0.44 & -0.30 & -0.05 \\ -0.01 & -0.06 & -0.09 & -0.15 & 0.10 & 0.43 \\ 0.39 & 0.12 & 0.44 & 0.31 & -0.23 & -0.03 \\ 0.25 & 0.30 & 0.42 & -0.49 & -0.30 & -0.08 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.8462 & 0.2026 & 0.8381 & 0.6813 & 0.8318 & 0.7095 \\ 0.5252 & 0.6721 & 0.0196 & 0.3795 & 0.5028 & 0.4289 \end{bmatrix},$$

$$\mathcal{S} = \left\{ \begin{array}{cccc} -4. & -4. - 4.i & -4. + 4.i & -8. \end{array} \right\}$$

Step 1. The Hessenberg-triangular form is obtained using the routine *tg01id* of **SLICOT** [4], which gives $n_1 = n_2 = n_3 = 2$ and

$$U = \begin{bmatrix} 0.8617 & -0.4638 & -0.1547 & 0.1354 & -0.0111 & 0 \\ -0.3931 & -0.6500 & 0.0253 & 0.2550 & -0.5978 & 0 \\ -0.2882 & -0.4490 & -0.0707 & 0.2810 & 0.7946 & 0 \\ 0.0347 & -0.3496 & 0.6601 & -0.6555 & 0.1056 & 0 \\ -0.1367 & -0.1965 & -0.7312 & -0.6388 & 0.0002 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.0000 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.4625 & -0.5139 & 0.4164 & -0.3171 & 0.1359 & 0.4791 \\ -0.2813 & 0.0415 & -0.3326 & -0.3994 & 0.7974 & 0.1147 \\ -0.3094 & 0.0586 & -0.3254 & -0.4991 & -0.5662 & 0.4746 \\ 0.0745 & -0.3952 & -0.5770 & 0.5951 & 0.0487 & 0.3858 \\ -0.5870 & 0.1422 & 0.5211 & 0.3586 & 0.1147 & 0.4710 \\ 0.5109 & 0.7446 & -0.0742 & 0.0900 & 0.0979 & 0.4017 \end{bmatrix}$$

$$H = \begin{bmatrix} -0.1292 & -0.4971 & 0.1452 & -0.4536 & -0.1400 & 0.0574 \\ -0.3736 & 0.1073 & 0.2410 & 0.2694 & 0.3252 & 0.0273 \\ 0 & 0.6028 & 0.3166 & 0.3726 & 0.1207 & -0.2010 \\ -0.0973 & -0.0731 & 0.0290 & 0.0445 & -0.0285 & -0.4304 \\ 0 & 0 & 0.3889 & -0.4692 & 0.6453 & -0.3248 \\ 0 & 0 & 0 & 0.8151 & 0.0307 & 0.0089 \end{bmatrix}$$

$$T = \begin{bmatrix} 4.2939 & -3.7547 & 3.9191 & -0.6761 & -0.3037 & 2.4476 \\ 0 & 1.8882 & 0.6373 & 3.1440 & -2.6534 & -3.4399 \\ 0 & 0 & -1.6168 & 0.9964 & 0.0933 & -0.5448 \\ 0 & 0 & 0 & -2.5247 & 0.5936 & 0.4213 \\ 0 & 0 & 0 & 0 & -4.6615 & 1.1357 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1.7661 \\ 0 & 0 & 0 & 0 & 0.7143 & 0.8936 \end{bmatrix}, \; C_1 = \begin{bmatrix} 0 & 1.7661 \\ 0.7143 & 0.8936 \end{bmatrix}$$

Step 2. Matrices Y, F and G are partitioned conformably with the partition of H and T.

Step 3. The elements of $\mathcal{S}$ are distributed in self-conjugate groups

$$S_1 = \{ \; -4. - 4.i \quad -4. + 4.i \; \} \text{ and } S_2 = \{ \; -4. \quad -8. \; \},$$

giving

$$F_{11} = \begin{bmatrix} -4 & -4 \\ 4 & -4 \end{bmatrix}, \; F_{22} = \begin{bmatrix} -8 & 0 \\ 0 & -4 \end{bmatrix}$$

Steps 4 and 5.

$$Y_{11} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; F_{21} = \begin{bmatrix} 0 & 0.3192 \\ -0.0227 & -0.0838 \end{bmatrix}, \; Y_{22} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Steps 6, 7 and 8. $i = 1$:

$$Y_{12} = \begin{bmatrix} 34.4496 & 175.1652 \\ -59.4891 & -180.3320 \end{bmatrix}, \ Y_{13} = 10^3 \begin{bmatrix} -0.5047 & -0.2690 \\ -1.4670 & -4.7718 \end{bmatrix}$$

$$G_1 = 10^4 \begin{bmatrix} -3.0784 & 5.0954 \\ -1.3720 & 2.2343 \end{bmatrix}$$

Steps 6, 7 and 8. $i = 2$:

$$Y_{23} = \begin{bmatrix} 111.9138 & 210.5227 \\ -17.9123 & -27.6408 \end{bmatrix}, \ G_2 = 10^3 \begin{bmatrix} 5.2835 & -8.7360 \\ -0.7980 & 1.3200 \end{bmatrix}$$

Step 9. The matrices Y , F and G, constructed from their blocks:

$$Y = \begin{bmatrix} 1. & 0. & 34.4496 & 175.1652 & -504.7174 & -269.0058 \\ 0. & 1. & -59.4891 & -180.3320 & -1467.0309 & -4771.7867 \\ 0. & 0. & 1. & 0. & 111.9138 & 210.5227 \\ 0. & 0. & 0. & 1. & -17.9123 & -27.6408 \end{bmatrix}$$

$$F = \begin{bmatrix} -4. & -4. & 0. & 0. \\ 4. & -4. & 0. & 0. \\ 0. & 0.3192 & -8. & 0. \\ -0.0227 & -0.0838 & 0. & -4. \end{bmatrix}, \ G = \begin{bmatrix} -30784.2368 & 50953.6711 \\ -13720.3313 & 22342.7464 \\ 5283.4797 & -8736.0473 \\ -797.9911 & 1320.0403 \end{bmatrix}$$

Step 10. The solution matrix X:

$$X = \begin{bmatrix} 24.8722 & 346.8634 & -354.5442 & -145.3255 & -137.3184 & 269.0058 \\ 0.6764 & 828.8481 & -1212.6037 & -76.2792 & 158.1880 & 4771.7867 \\ -1.4021 & -66.8764 & 88.8547 & 12.4743 & -0.7081 & -210.5227 \\ 0.3350 & 10.9629 & -13.9519 & -2.5464 & -0.6424 & 27.6408 \end{bmatrix}$$

Verification:

$$\|AX - FXE - GC\|_F = 1.2244 \times 10^{-11}$$

$$\Omega(F) = \left\{ \ -4. \quad -4. - 4.i \quad -4. + 4.i \quad -8. \ \right\}.$$

4.3.2 SVD Method

We will now show how the algorithm of Datta and Sarkissian [23], presented in Chapter 3, can be generalized to solve the descriptor Sylvester-observer equation (4.2.6).

The development of this method follows the same template as the development in Subsection 3.2.3. In fact, derivations given there can be considered as special cases of our derivations given here with $E = I$.

Let $A, E \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{r\times n}$ and $\mathcal{S} \in \mathbb{C}^{n-r}$ be such that (4.1.1) is observable, C has full rank r and $\Omega(A, E) \bigcap \mathcal{S} = \emptyset$.

Development of the Method:

In order to solve (4.2.6) in a block fashion, we partition the matrix F in the form

$$F = \begin{bmatrix} F_{11} & F_{12} & & \\ & F_{22} & F_{23} & \\ & & \cdots & \cdots \\ & & & F_{qq} \end{bmatrix} \tag{4.3.28}$$

where diagonal blocks $F_{ii} \in \mathbb{R}^{n_i\times n_i}$, $i = 1, \ldots, q$ of F are chosen to be matrices in real Schur form (see [20, 38]) such that the requirement $\Omega(F) = \mathcal{S}$ is satisfied. Partitioning the matrices X and G conformably with the partition of F, we obtain

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_q \end{bmatrix}, \; G = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ G_q \end{bmatrix}. \tag{4.3.29}$$

Using (4.3.28) and (4.3.29) in (4.2.6) and equating the corresponding blocks on each side, we obtain

$$X_iA - F_{ii}X_iE = F_{i,i+1}X_{i+1}E \ , \ i = 1, \ldots, q-1 \tag{4.3.30}$$

$$X_qA - F_{qq}X_qE = G_qC. \tag{4.3.31}$$

Let $\tilde{F}_{qq}$ be a matrix with a prescribed spectrum and let Y_q satisfy the generalized Sylvester equation

$$Y_qA - \tilde{F}_{qq}Y_qE = C. \tag{4.3.32}$$

Take now the SVD of Y_q^T : $Y_q^T = U_q\Sigma_qV_q^T$. Then equation (4.3.32) becomes

$$V_q\Sigma_qU_q^TA - \tilde{F}_{qq}V_q\Sigma_qU_q^TE = C.$$

Since C is assumed to have full rank r, we have

$$U_q^TA - (\Sigma_q^{-1}V_q^T\tilde{F}_{qq}V_q\Sigma_q)U_q^TE = \Sigma_q^{-1}V_q^TC \tag{4.3.33}$$

meaning that we have a solution of (4.3.31) with

$$X_q = U_q^T \ , \ F_{qq} = \Sigma_q^{-1}V_q^T\tilde{F}_{qq}V_q\Sigma_q \ , \ G_q = \Sigma_q^{-1}V_q^T. \tag{4.3.34}$$

This strategy now can be repeated for each $i = q-1, \ldots, 2, 1$ to obtain a solution of (4.3.30) as follows: choosing $\tilde{F}_{ii} \in \mathbb{R}^{n_i \times n_i}$ with a prescribed spectrum, we solve

$$Y_iA - \tilde{F}_{ii}Y_iE = X_{i+1}E \tag{4.3.35}$$

for Y_i and compute the SVD of Y_i^T : $Y_i^T = U_i\Sigma_iV_i^T$ to obtain

$$U_i^TA - (\Sigma_i^{-1}V_i^T\tilde{F}_{ii}V_i\Sigma_i)U_i^TE = \Sigma_i^{-1}V_i^TX_{i+1}E. \tag{4.3.36}$$

Assuming that Y_i has full rank, a solution of (4.3.30) is then obtained with

$$X_i = U_i^T \ , \ F_{ii} = \Sigma_i^{-1} V_i^T \tilde{F}_{ii} V_i \Sigma_i \ , \ F_{i,i+1} = \Sigma_i^{-1} V_i^T. \tag{4.3.37}$$

The above discussion leads to the following algorithm:

Algorithm 4.3.2: The SVD Algorithm for the Descriptor Sylvester-observer Equation

Input: Matrices $A, E \in R^{n \times n}$ and $C \in \mathbb{R}^{r \times n}$ of the linear system (4.1.1) and a self-conjugate set $\mathcal{S} \in \mathbb{C}^s$.

Output: Block matrices X,F and G, such that $\Omega(F) = \mathcal{S}$ and $XA - FXE = GC$.

Assumption: The system (4.1.1) is observable, C has full rank and $\Omega(A, E) \bigcap \mathcal{S} = \emptyset$. Also, either $s = n$ or $s = n - r$.

Step 1: Distribute the elements of $\mathcal{S}$ in self-conjugate groups $S_i, i = 1, \ldots, q$ that will, later on, be assigned to the diagonal blocks of the matrix F.

Step 2: For $i = q, q-1, \ldots, 1$ do Steps 3 through 9

Step 3: If $i = q$ then set $J = C$; else set $J = X_{i+1}E$.

Step 4: Choose $\tilde{F}_{ii}$ to be a matrix in real Schur form such that $\Omega(\tilde{F}_{ii}) = S_i$.

Step 5: Solve, for Y_i, the generalized Sylvester equation

$$Y_i A - \tilde{F}_{ii} Y_i E = J$$

using Algorithm 4.3.4, to be described in Subsection 4.3.4.

Step 6: Compute the economy size SVD of Y_i^T, that is, compute matrices U_i and V_i and diagonal matrix Σ_i such that $Y_i^T = U_i \Sigma_i V_i^T$.

Step 7: If Σ_i is nonsingular, compute

$$F_{ii} = \Sigma_i^{-1} V_i^T \tilde{F}_{ii} V_i \Sigma_i$$

taking advantage of the structure of $\tilde{F}_{ii}$. Otherwise, go back to Step 4 and choose another set S_i .

Step 8: If $i = q$, compute

$$G_q = \Sigma_q^{-1} V_q^T$$

else compute

$$F_{i,i+1} = \Sigma_i^{-1} V_i^T \ .$$

Step 9: Set $X_i = U_i^T$.

Step 10: Form matrices X, F and G from their blocks computed above.

Remarks

1. As in [23], the above algorithm produces a matrix X such that every block-row $X_i, i = 1, \ldots, q$ is a matrix whose rows form an orthonormal set of vectors.

2. The algorithm does not require reduction of the system triple (E, A, C). This is specially attractive when E and A are large and sparse, as long as we are able to exploit this in the solution of the subproblems in Step 5.

3. If the matrices E and A are dense, then the pair (E, A) can be reduced to (T, H), where H is an upper Hessenberg matrix and T is upper triangular. The structure of this Hessenberg-triangular form can again be exploited in the solution of the subproblems in Step 5. An efficient method for computing this reduction, also known as the first stage of the QZ method, can be found in [20, 38].

4. The algorithm is rich in BLAS-3 computations and thus is suitable for high-performance computing using LAPACK [32] .

Example 4.3.2: Consider the same data as in Example 4.3.1.

Step 1. $S_1 = \{\ -4. - 4.i \quad -4. + 4.i\ \}\ ,\ S_2 = \{\ -8. \quad -4.\ \}$.

Steps 2, 3 and 4. $i = 1$:

$$J = C = \begin{bmatrix} 0.8462 & 0.2026 & 0.8381 & 0.6813 & 0.8318 & 0.7095 \\ 0.5252 & 0.6721 & 0.0196 & 0.3795 & 0.5028 & 0.4289 \end{bmatrix}$$

$$\tilde{F}_{22} = \begin{bmatrix} -8. & 0. \\ 0. & -4. \end{bmatrix}$$

Step 5.

$$Y_2 = \begin{bmatrix} 0.0553 & 0.0695 & 0.1667 & -0.2710 & -0.2808 & -10.4390 \\ 0.0799 & 0.1238 & 0.2100 & -0.4018 & -0.4254 & -7.5856 \end{bmatrix}$$

Step 6.

$$U_2 = \begin{bmatrix} -0.0071 & -0.0100 & -0.0200 & 0.0353 & 0.0369 & 0.9984 \\ -0.1220 & -0.2251 & -0.2725 & 0.6288 & 0.6794 & -0.0559 \end{bmatrix}^T$$

$$\Sigma_2 = \begin{bmatrix} 12.9244 & 0. \\ 0. & 0.2627 \end{bmatrix}\ ,\ V_2 = \begin{bmatrix} -0.8083 & 0.5888 \\ -0.5888 & -0.8083 \end{bmatrix}$$

Steps 7 and 8.

$$F_{22} = \begin{bmatrix} -6.6134 & 0.0387 \\ 93.6387 & -5.3866 \end{bmatrix}\ ,\ G_2 = \begin{bmatrix} -0.0625 & -0.0456 \\ 2.2408 & -3.0764 \end{bmatrix}$$

Step 9.

$$X_2 = \begin{bmatrix} -0.0071 & -0.0100 & -0.0200 & 0.0353 & 0.0369 & 0.9984 \\ -0.1220 & -0.2251 & -0.2725 & 0.6288 & 0.6794 & -0.0559 \end{bmatrix}$$

Steps 2, 3 and 4. $i = 2$:

$$J = \begin{bmatrix} -0.0568 & -0.0599 & -0.0800 & 0.0705 & 0.0369 & 0. \\ -0.9761 & -1.3504 & -1.0902 & 1.2577 & 0.6794 & 0. \end{bmatrix}$$

$$\tilde{F}_{11} = \begin{bmatrix} -4. & -4. \\ 4. & -4. \end{bmatrix}$$

Step 5.

$$Y_1 = \begin{bmatrix} 0.0192 & 0.0333 & 0.0463 & -0.1006 & -0.1129 & -0.5384 \\ -0.0156 & -0.0287 & -0.0347 & 0.0801 & 0.0855 & 0.4238 \end{bmatrix}$$

Step 6.

$$U_1 = \begin{bmatrix} -0.0345 & -0.0613 & -0.0809 & 0.1797 & 0.1979 & 0.9576 \\ -0.1025 & -0.5393 & 0.3682 & 0.2109 & -0.7131 & 0.1007 \end{bmatrix}^T$$

$$\Sigma_1 = \begin{bmatrix} 0.7155 & 0. \\ 0. & 0.0036 \end{bmatrix} , \; V_1 = \begin{bmatrix} -0.7861 & 0.6181 \\ 0.6181 & 0.7861 \end{bmatrix}$$

Steps 7 and 8.

$$F_{11} = \begin{bmatrix} -4. & 0.0203 \\ -789.5867 & -4. \end{bmatrix} , \; F_{12} = \begin{bmatrix} -1.0987 & 0.8638 \\ 170.5128 & 216.8810 \end{bmatrix}$$

Step 9.

$$X_1 = \begin{bmatrix} -0.0345 & -0.0613 & -0.0809 & 0.1797 & 0.1979 & 0.9576 \\ -0.1025 & -0.5393 & 0.3682 & 0.2109 & -0.7131 & 0.1007 \end{bmatrix}$$

Step 10. The matrices X , F and G obtained from their computed blocks are:

$$X = \begin{bmatrix} -0.0345 & -0.0613 & -0.0809 & 0.1797 & 0.1979 & 0.9576 \\ -0.1025 & -0.5393 & 0.3682 & 0.2109 & -0.7131 & 0.1007 \\ -0.0071 & -0.0100 & -0.0200 & 0.0353 & 0.0369 & 0.9984 \\ -0.1220 & -0.2251 & -0.2725 & 0.6288 & 0.6794 & -0.0559 \end{bmatrix},$$

$$F = \begin{bmatrix} -4. & 0.0203 & -1.0987 & 0.8638 \\ -789.5867 & -4. & 170.5128 & 216.8810 \\ 0. & 0. & -6.6134 & 0.0387 \\ 0. & 0. & 93.6387 & -5.3866 \end{bmatrix} , \; G = \begin{bmatrix} 0. & 0. \\ 0. & 0. \\ -0.0625 & -0.0456 \\ 2.2408 & -3.0764 \end{bmatrix} .$$

Verification:

$$\|AX - FXE - GC\|_F = 1.9568 \times 10^{-13}$$
$$\Omega(F) = \left\{ \begin{array}{cccc} -4. & -4. - 4.i & -4. + 4.i & -8. \end{array} \right\}.$$

4.3.3 Parametric QR Method

The Parametric QR method of Carvalho, Hong and Datta, developed in Subsection 3.2.4, can also be generalized to solve the descriptor Sylvester-observer equation (4.2.6).

The derivation of this method is along the same line as of Algorithm 3.2.4. Therefore, we omit the details here, and go directly to the description of the corresponding algorithm.

Algorithm 4.3.3: *The Parametric QR Algorithm for Descriptor Sylvester-observer Equation*

Input: Matrices $A, E \in R^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ of the linear system (4.1.1) and a self-conjugate set $\mathcal{S} \in \mathbb{C}^s$.

Output: Block matrices X,F and G, such that $\Omega(F) = \mathcal{S}$ and $XA - FXE = GC$.

Assumption: The system (4.1.1) is observable, C has full rank and $\Omega(A, E) \bigcap \mathcal{S} = \emptyset$. Also, either $s = n$ or $s = n - r$.

Step 1: Set $\ell = r$ and $N_1 = I_{r\times r}$.

Step 2: Compute the thin RQ factorization of C : $R_cQ_c = C$ where $Q_c \in \mathbb{R}^{r\times n}$ and $R_c \in \mathbb{R}^{r\times r}$.

Step 3: For $i = 1, 2, \ldots$ do steps 4 through 10

Step 4: Set $S_i \in \mathbb{C}^\ell$ to be a self-conjugate subset of the part of $\mathcal{S}$ that was not used yet.

Step 5: Set $F_{ii} \in \mathbb{R}^{\ell\times\ell}$ to be any matrix in upper real Schur form satisfying

$\Omega(F_{ii}) = S_i$.

Step 6: Free parameter step. If $i > 1$ set $N_i \in \mathbb{R}^{\ell \times n_i}$ and $F_{ij} \in \mathbb{R}^{\ell \times n_j}$, $j = 1, \ldots, i-1$ to be arbitrary matrices.

Step 7: Solve the descriptor Sylvester equation

$$X_i A - F_{ii} X_i E = N_i Q_c + \sum_{j=1}^{i-1} F_{ij} X_j E, \tag{4.3.38}$$

for $X_i \in \mathbb{R}^{\ell \times n}$, using Algorithm 4.3.4 to be given in Subsection 4.3.4.

Step 8: Set n_i to be the number of rows of X_i that are linearly independent of the rows of the matrix $\begin{bmatrix} X_1 \\ \ldots \\ X_{i-1} \end{bmatrix}$.

If $n_i < \ell$ then set $\ell = n_i$, restore S_i to $\mathcal{S}$ and do Steps 4 - 8 again.

Step 9: Find, implicitly, an orthogonal matrix Q_i that reduces the matrix

$$\begin{bmatrix} X_1 \\ \ldots \\ X_{i-1} \\ X_i \end{bmatrix}$$

to upper triangular form via left multiplication by Q_i^T. Then compute the matrix updates

$$\begin{bmatrix} X_1 \\ \ldots \\ X_{i-1} \\ X_i \end{bmatrix} \leftarrow Q_i^T \begin{bmatrix} X_1 \\ \ldots \\ X_{i-1} \\ X_i \end{bmatrix}, \quad \begin{bmatrix} G_1 \\ \ldots \\ G_{i-1} \\ G_i \end{bmatrix} \leftarrow Q_i^T \begin{bmatrix} G_1 \\ \ldots \\ G_{i-1} \\ N_i R_c^{-1} \end{bmatrix},$$
$$F^i \leftarrow Q_i^T F^i Q_i$$

Step 10: If $n_1 + \ldots + n_i = s$ then set $p = i$ and exit loop.

End For

Step 11: Form matrices X, F and G from their computed blocks.

Remark: The remarks made in the context of Algorithm 3.2.4 are also valid here with appropriate modifications.

Example 4.3.3: Consider, again, the same data as in Example 4.3.1.

Step 1. $s = 2$, $\ell = 2$ and $N_1 = I_2$.

Step 2.

$$R_c = \begin{bmatrix} 1.1027 & -1.3795 \\ 0. & -1.1440 \end{bmatrix},$$

$$Q_c = \begin{bmatrix} 0.1931 & -0.5512 & 0.7386 & 0.2028 & 0.2045 & 0.1744 \\ -0.4591 & -0.5875 & -0.0171 & -0.3317 & -0.4395 & -0.3749 \end{bmatrix}$$

Steps 3 and 4. $i = 1$, $S_1 = \{\ -4. - 4.i \quad -4. + 4.i\ \}$.

Steps 5 , 6 and 7 .

$$F_{11} = \begin{bmatrix} -4 & -4 \\ 4 & -4 \end{bmatrix}, \ X_1 =$$

$$\begin{bmatrix} 0.0438 & 0.0529 & 0.1350 & -0.2160 & -0.2236 & -3.4389 \\ -0.0089 & -0.0329 & 0.0072 & 0.0406 & 0.0274 & 4.8405 \end{bmatrix}$$

Steps 8, 9 and 10. $\ell = 2$:

$$X_1 = \begin{bmatrix} -0.0447 & -0.0583 & -0.1308 & 0.2197 & 0.2246 & 4.3313 \\ 0. & -0.0218 & 0.0338 & -0.0030 & -0.0175 & 4.0615 \end{bmatrix}$$

$$F_{11} = \begin{bmatrix} -4. & 4. \\ -4. & -4. \end{bmatrix}, \ G_1 = \begin{bmatrix} -0.8888 & 0.8982 \\ 0.1800 & -1.0738 \end{bmatrix}$$

Steps 3 and 4. $i = 2$, $S_2 = \{\ -4 \quad -8\ \}$.

Steps 5, 6 and 7. Choosing $N_2 = 0$, $F_{21} = I_2$, we obtain

$$F_{22} = \begin{bmatrix} -4 & 0 \\ 0 & -8 \end{bmatrix}, \; X_2 =$$

$$\begin{bmatrix} -0.0155 & -0.0203 & -0.0467 & 0.0789 & 0.0865 & 0.4506 \\ 0. & -0.0027 & 0.0045 & -0.0006 & -0.0027 & -0.0135 \end{bmatrix}$$

Steps 8, 9 and 10. $n_2 = 2$:

$$X = \begin{bmatrix} 0.0473 & 0.0618 & 0.1389 & -0.2335 & -0.2405 & -4.2395 \\ 0. & 0.0220 & -0.0341 & 0.0030 & 0.0177 & -4.0281 \\ 0. & 0. & 0.0012 & -0.0025 & -0.0081 & 0.9585 \\ 0. & 0. & 0. & 0.0002 & 0.0003 & -0.5848 \end{bmatrix}$$

$$F = \begin{bmatrix} -3.6902 & 3.7484 & -0.0765 & 0.4757 \\ -3.7481 & -3.9385 & 1.3336 & 0.4280 \\ 0.8589 & -1.3318 & -4.3271 & -0.4194 \\ -0.5380 & -0.4010 & -0.0890 & -8.0442 \end{bmatrix}, \; G = \begin{bmatrix} 0.8396 & -0.8485 \\ -0.1785 & 1.0653 \\ -0.2924 & 0.3028 \\ -0.0036 & 0.1153 \end{bmatrix}$$

and the algorithm terminates with $p = 2$.

Verification:

$$\|AX - FXE - GC\|_F = 5.8846 \times 10^{-16},$$
$$\Omega(F) = \left\{ \begin{array}{cccc} -4. & -4. - 4.i & -4. + 4.i & -8. \end{array} \right\}.$$

4.3.4 Solution of the Descriptor Sylvester Equation

To implement algorithms 4.3.2 , 4.3.3 and 6.3.2, we need to solve a descriptor Sylvester-equation of the form:

$$YA - FYE = J \tag{4.3.39}$$

where $A, E \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{r \times r}$ is an upper real Schur matrix and $J \in \mathbb{R}^{r \times n}$ is a general matrix.

This matrix equation is a special case of the generalized Sylvester equation $BXA+CXD=E$, for which there exist in the literature numerically viable methods of solution [16, 37].

Here we develop a new method for solving (4.3.39) based on the Hessenberg-triangular reduction of a pair of matrices (A, E). The method presented here is also a generalization of the *Hessenberg-Schur method* [39] for the standard Sylvester equation. This method exploits the fact that the matrix F is already in the real Schur form.

Partitioning

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \cdots \\ y_r^T \end{bmatrix}, \quad J = \begin{bmatrix} j_1^T \\ j_2^T \\ \cdots \\ j_r^T \end{bmatrix} \tag{4.3.40}$$

and defining $F = (F_{ij})$, equation (4.3.39) can be written as

$$y_k^T A - F_{k,k-1} y_{k-1}^T E - F_{k,k} y_k^T E - \sum_{i=k+1}^{r} F_{k,i} y_i^T E = j_k^T \ , \ k = 1, 2, \ldots, r. \tag{4.3.41}$$

As in the Hessenberg-Schur method, we can now recursively compute the rows $y_r, y_{r-1}, \ldots, y_1$ by solving linear algebraic systems of equations

- of size n when the y_k's decouple $(F_{k,k-1} = 0)$
- of size $2n$ when y_k and y_{k-1} are coupled through F.

We now show how the above two systems can be conveniently solved by exploiting the Hessenberg-triangular structure obtained by orthogonal reduction of the pair (A, E). If these matrices already have convenient structures, then no reduction is necessary.

Recall that, given two matrices A and E of order n, the *Hessenberg-triangular* reduction method [20, 38] (also well-known as the first stage of the QZ method [20, 38]) computes orthogonal matrices $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} U^T A^T V &= H \\ U^T E^T V &= T \end{aligned} \tag{4.3.42}$$

where H is a upper Hessenberg matrix , T is an upper triangular matrix.

Transposing (4.3.41) and using (4.3.42), we then have

$$HZ - TZF^T = D \tag{4.3.43}$$

where $Z = V^T Y^T$ and $D = U^T J^T$.

Therefore, defining $Z = \begin{bmatrix} z_1 & \dots & z_r \end{bmatrix}$ and $D = \begin{bmatrix} d_1 & \dots & d_r \end{bmatrix}$, we obtain

$$Hz_k - F_{k,k-1}Tz_{k-1} - F_{k,k}Tz_k = d_k + \sum_{i=k+1}^{r} F_{k,i}Tz_i \ , \ k = r, r-1, \dots, 2, 1 \tag{4.3.44}$$

Consider now two cases:

Case 1: If $F_{k,k-1} = 0$ then z_k is obtained by solving the Hessenberg system:

$$(H - F_{k,k}T)z_k = d_k + \sum_{i=k+1}^{r} F_{k,i}Tz_i = d'_k. \tag{4.3.45}$$

Case 2: Otherwise, z_{k-1} and z_k are computed simultaneously by solving the algebraic linear system:

$$\begin{bmatrix} H - F_{k-1,k-1}T & -F_{k-1,k}T \\ -F_{k,k-1}T & H - F_{k,k}T \end{bmatrix} \begin{bmatrix} z_{k-1} \\ z_k \end{bmatrix} = \begin{bmatrix} d'_{k-1} \\ d'_k \end{bmatrix} \tag{4.3.46}$$

where

$$\begin{aligned} d'_{k-1} &= d_{k-1} + \sum_{i=k+1}^{r} F_{k-1,i} T z_i \\ d'_k &= d_k + \sum_{i=k+1}^{r} F_{k,i} T z_i. \end{aligned} \tag{4.3.47}$$

Define

$$B = (b_{ij}) = \begin{bmatrix} F_{k-1,k-1} & F_{k-1,k} \\ F_{k,k-1} & F_{k,k} \end{bmatrix}. \tag{4.3.48}$$

As in the Hessenberg-Schur method, equation (4.3.46) must be reordered in order to bring back the upper Hessenberg structure.

Then the coefficient matrix of the $2n \times 2n$ linear system (4.3.46) has a very special structure, which is illustrated below in the case when $n = 3$. The coefficient matrix in this case is:

$$\begin{bmatrix} h_{11} - b_{11}t_{11} & h_{12} - b_{11}t_{12} & h_{13} - b_{11}t_{13} & -b_{12}t_{11} & -b_{12}t_{12} & -b_{12}t_{13} \\ h_{21} & h_{22} - b_{11}t_{22} & h_{23} - b_{11}t_{23} & & -b_{12}t_{22} & -b_{12}t_{23} \\ & h_{32} & h_{33} - b_{11}t_{33} & & & -b_{12}t_{33} \\ -b_{21}t_{11} & -b_{21}t_{12} & -b_{21}t_{13} & h_{11} - b_{22}t_{11} & h_{12} - b_{22}t_{12} & h_{13} - b_{22}t_{13} \\ & -b_{21}t_{22} & -b_{21}t_{23} & h_{21} & h_{22} - b_{22}t_{22} & h_{23} - b_{22}t_{23} \\ & & -b_{21}t_{33} & & h_{32} & h_{33} - b_{22}t_{33} \end{bmatrix}.$$

After properly reordering rows and columns of the above matrix, we obtain

$$\begin{bmatrix} h_{11} - b_{11}t_{11} & -b_{12}t_{11} & h_{12} - b_{11}t_{12} & -b_{12}t_{12} & h_{13} - b_{11}t_{13} & -b_{12}t_{13} \\ -b_{21}t_{11} & h_{11} - b_{22}t_{11} & -b_{21}t_{12} & h_{12} - b_{22}t_{12} & -b_{21}t_{13} & h_{13} - b_{22}t_{13} \\ h_{21} & 0 & h_{22} - b_{11}t_{22} & -b_{12}t_{22} & h_{23} - b_{11}t_{23} & -b_{12}t_{23} \\ & h_{21} & -b_{21}t_{22} & h_{22} - b_{22}t_{22} & -b_{21}t_{23} & h_{23} - b_{22}t_{23} \\ & & h_{32} & 0 & h_{33} - b_{11}t_{33} & -b_{12}t_{33} \\ & & & h_{32} & -b_{21}t_{33} & h_{33} - b_{22}t_{33} \end{bmatrix}$$

which is now a matrix $G \in \mathbb{R}^{2n \times 2n}$ with only two non-zero sub-diagonals.

This matrix G can be generated using the *Kronecker product* of matrices [38], as follows:

$$G = kron(H, I_2) - kron(T, B) \tag{4.3.49}$$

where I_2 stands for the identity matrix of order 2.

Therefore, equation (4.3.46) is equivalent to

$$G \begin{bmatrix} x_1 \\ \dots \\ x_{2n} \end{bmatrix} = \begin{bmatrix} w_1 \\ \dots \\ w_{2n} \end{bmatrix} \tag{4.3.50}$$

where

$$w_{2i-1} = d'_{k-1}(i)\ ,\ w_{2i} = d'_k(i)\ , i = 1, 2, \dots, n \tag{4.3.51}$$

and after the solution $x = (x_i)$ is computed, z_{k-1} and z_k are recovered via

$$z_{k-1}(i) = x_{2i-1}\ ,\ z_k(i) = x_{2i}\ , i = 1, 2, \dots, n \tag{4.3.52}$$

where $z_k(i)$ stands for the $i - th$ component of the vector z_k.

The above discussion leads to the following algorithm:

Algorithm 4.3.4: Hessenberg-triangular Algorithm for Descriptor Sylvester Equation

Input: matrices $A, E \in \mathbb{R}^{n \times n}$, upper Real Schur matrix $F \in \mathbb{R}^{r \times r}$ and a matrix $J \in \mathbb{R}^{r \times n}$.

Output: matrix $Y \in \mathbb{R}^{r \times n}$ such that $YA - FYE = J$.

Step 1: Compute orthogonal matrices $U, V \in \mathbb{R}^{n \times n}$, obtaining H and T as in (4.3.42). Compute $D = U^T J^T$.

Step 2: Set $k = r$.

Step 3: While $k > 0$ do Steps 4 and 5

Step 4: Compute $d'_k = d_k + \sum_{i=k+1}^{r} F_{k,i} T z_i$

Step 5: If $k = 1$ or $F_{k,k-1} = 0$ solve the Hessenberg linear system

$$[H - F_{kk}T]\, z_k = d'_k$$

for $z_k \in \mathbb{R}^n$ and update $k \leftarrow k - 1$.

Else do Steps 6 , 7 and 8

Step 6: Compute $d'_{k-1} = d_{k-1} + \sum_{i=k+1}^{r} F_{k-1,i} T z_i$.

Step 7: Using (4.3.48) and (4.3.49), solve the structured linear system $Gx = w$, where $x = (x_i), i = 1, \ldots, 2n$ and $w = (w_i), i = 1, \ldots, 2n$ is given by (4.3.51).

Step 8: Set $z_{k-1}(i) = x_{2i-1}$, $z_k(i) = x_{2i}$, $i = 1, 2, \ldots, n$ and update $k \leftarrow k - 2$.

End If

End while

Step 9: Form the matrix Z and recover the solution Y as $Y = Z^T V^T$.

Remark: If matrices A and E already have some structure that can be exploited in the solution of the linear systems of steps 5 or 7, then in Step 1 we only need to set $U = V = I$.

4.4 Comparison of Methods and Numerical Experiments

In this section we compare the methods presented in the last section for the solution of the descriptor Sylvester-observer equation for reduced-order state estimation of linear descriptor systems.

The flop-counts of these methods are not included in our comparison here; however, as in Chapter 3, other numerical attributes such as, normalized cpu-time, residual error, numerical rank and condition number are included.

4.4.1 Comparison of Methods

In order to compare the block algorithms with respect to other numerical properties, the algorithms were tested on several benchmark problems with increasing size n taken from [40].

Three MATLAB-called Fortran 77 routines were written for this purpose, implementing the three block algorithms of Chapter 3. They are named as follows:

- **sylodbhr**: implements Algorithm 4.3.1
- **sylodsvd**: implements Algorithm 4.3.2
- **sylodpqr**: implements Algorithm 4.3.3

These routines were written with the goal of calling routines of the BLAS, LAPACK [2]and SLICOT [4] libraries as much as possible.

Four *performance criteria*, analogous to the ones defined in Subsection 3.3.1, are considered to evaluate the behavior of these block algorithms.

4.4.2 Numerical Experiments

The numerical experiments were performed using the families of matrices *Lehmer*, *Lesp*, *Pdtoep* and *Pentoep* from [40].

Setup of the First Benchmark: For every positive integer n, the matrices $A = \mathit{pdtoep}(n)$ and $E = \mathit{lehmer}(n)$ are symmetric positive semidefinite Toeplitz and symmetric positive definite, respectively. A precise definition of the *pdtoep* family, which involves the use of randomly generated numbers, can be found in [40]. $E = \mathit{lehmer}(n)$ is the symmetric positive definite $n \times n$ matrix defined by

$$E(i,j) = E(j,i) = i/j \text{ for } j \geq i.$$

E is totally nonnegative, E^{-1} is tridiagonal, and explicit formulas are known for its entries. Also $n \leq cond(E) \leq 4n^2$. Therefore, the generalized eigenvalues of (A, E) lie in the positive real axis.

In the numerical testing with these two families, for each n, the $n - r$ elements of the vector $\mathcal{S}$ were taken as follows:

- Roughly two thirds of them are conjugate pairs uniformly distributed along the open arc $|z| = n\sqrt{n}$, $Imag(z) < 0$.
- Roughly one third of them are real negative numbers uniformly distributed along the segment $[-n\sqrt{n}, -n]$.

This choice ensures that the set $\mathcal{S}$ is in the left side of the complex plane, and do not overlap with the generalized eigenvalues of (A, E). Figure 4.1 shows, for the particular case $n = 50$ and $r = 7$, how these two sets are distributed in the complex plane.

Setup of the Second Benchmark: For every positive integer n, the matrices $A = lesp(n)$ and $E = pentoep(n)$ are symmetric positive definite tridiagonal and pentadiagonal Toeplitz, respectively. These $n \times n$ matrices are defined by $A = D^{-1}A_0D$, where

$$A_0 = \begin{bmatrix} -5 & 1 & & & & \\ 1 & -7 & 1 & & & \\ & 1 & -9 & 1 & & \\ & & 1 & -11 & \ldots & \\ & & & \ldots & \ldots & 1 \\ & & & & 1 & -2n-3 \end{bmatrix},$$

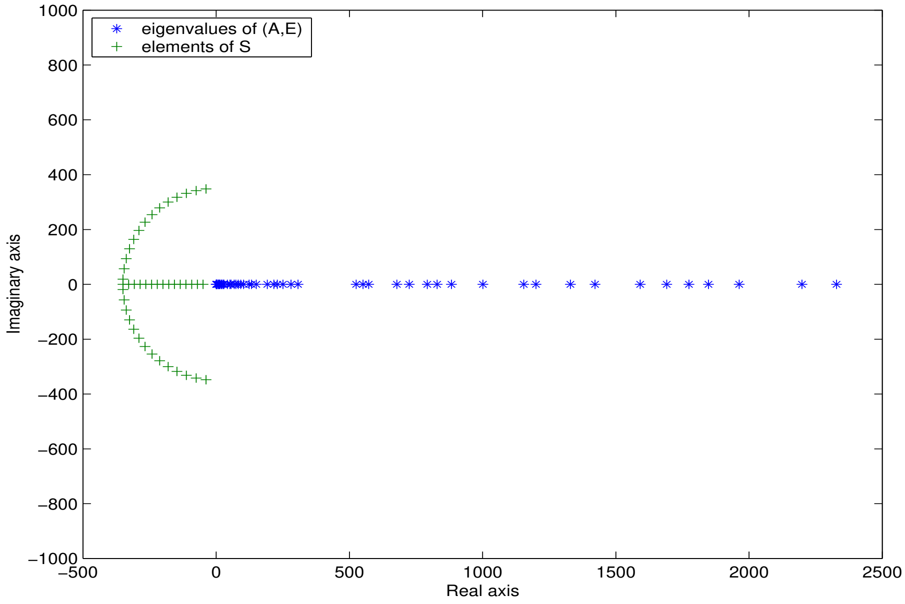

Figure 4.1: Spectrum of (A, E) and Chosen Elements of $\mathcal{S}$.

$$E = \begin{bmatrix} 0 & 10 & 1 & & & \\ -10 & 0 & 10 & 1 & & \\ 1 & -10 & 0 & 10 & 1 & \\ & \dots & \dots & \dots & \dots & 1 \\ & & \dots & \dots & \dots & 10 \\ & & & 1 & -10 & 0 \end{bmatrix}$$

$$D = diag\begin{pmatrix} 1! & 2! & 3! & \dots & n! \end{pmatrix}.$$

It can be shown that, if n is even, then $n = 2p$ generalized eigenvalues of (A, E) are p self-conjugate non-real complex conjugate pairs which lie mostly inside the circle $|z - n/4| = n/4$.

In the numerical testing with these two families, for each $n = 2, 4, 6, \dots$, the $n - r$ elements of the vector $\mathcal{S}$ were taken as negative numbers uniformly distributed in the interval $[-5n/12, -n/12]$. This choice ensures that the set $\mathcal{S}$ is in the left side of the complex plane, and do not overlap with the generalized eigenvalues of (A, E).

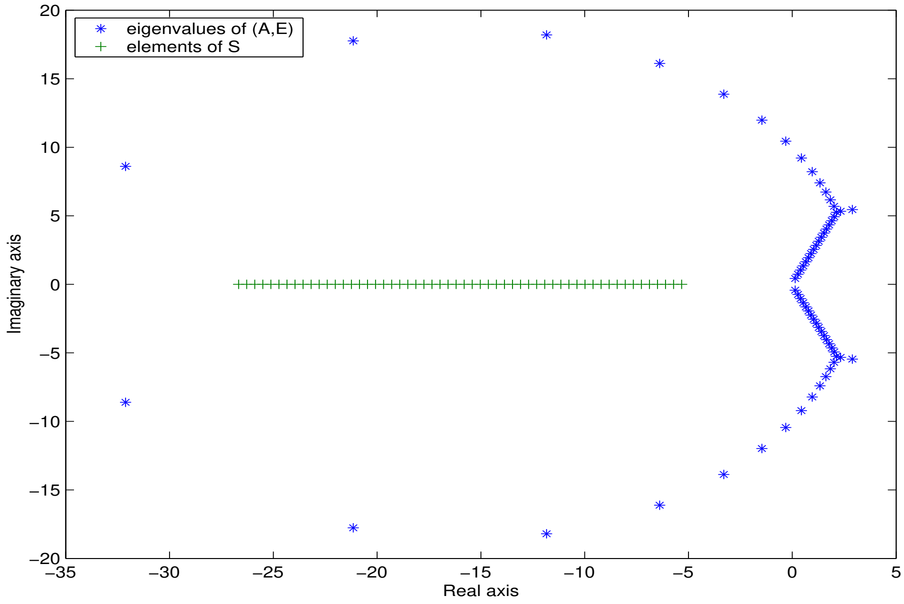

Figure 4.2: Spectrum of (A, E) and Chosen Elements of $\mathcal{S}$.

Figure 4.2 shows, for the particular case $n = 64$ and $r = 8$, how these two sets are distributed in the complex plane.

First Benchmark: Figure 4.3 corresponds to the testing with the families of matrices *pdtoep* and *Lehmer*. The numerical experiments were done in a Pentium II 400 MHz IBM-PC under Linux-Mandrake 8.0 and Matlab 6.0 .

From the plots, it is seen that:

- The Hessenberg-triangular method again requires significantly less cpu-time to achieve its goal. It is also the one that gives the most well-conditioned linear system (4.2.10).

- The Hessenberg-triangular method is, however, significantly less accurate than the other two methods. The Parametric QR method seems to be the most accurate one, but it is the one that gives the most ill-conditioned linear system (4.2.10).

- The SVD method, contrary to what was observed in experiments with standard systems in Chapter 3, performs poorly in the numerical rank test.

Second Benchmark: Figure 4.4 corresponds to the testing with the families of matrices *lesp* and *pentoep*. The numerical experiments were done in a Pentium II 400 MHz IBM-PC under Linux-Mandrake 8.0 and Matlab 6.0 .

From the plots, it is seen that:

- The Hessenberg-triangular method, which cannot exploit the band structure of matrices A and E, is the one that requires more cpu time.

- The SVD method is now the one which is the best to prevent losses in the numerical rank of X.

- The Hessenberg-triangular method again is also significantly less accurate than the other two methods, giving very poor results this time. The Parametric QR method seems to be the most accurate one, but again gives a remarkably ill-conditioned linear system (4.2.10).

Therefore, because of the inferior performance of the Hessenberg-triangular algorithm in all the residual accuracy tests, and that of the Parametric QR algorithm in all the condition number tests, we conclude that the SVD algorithm is the one that gives the best performance when state estimation of descriptor systems is concerned. It must be also noted that a different setup of the free parameters of the Parametric QR method can possibly improve its performance. This is left as a topic for future investigation.

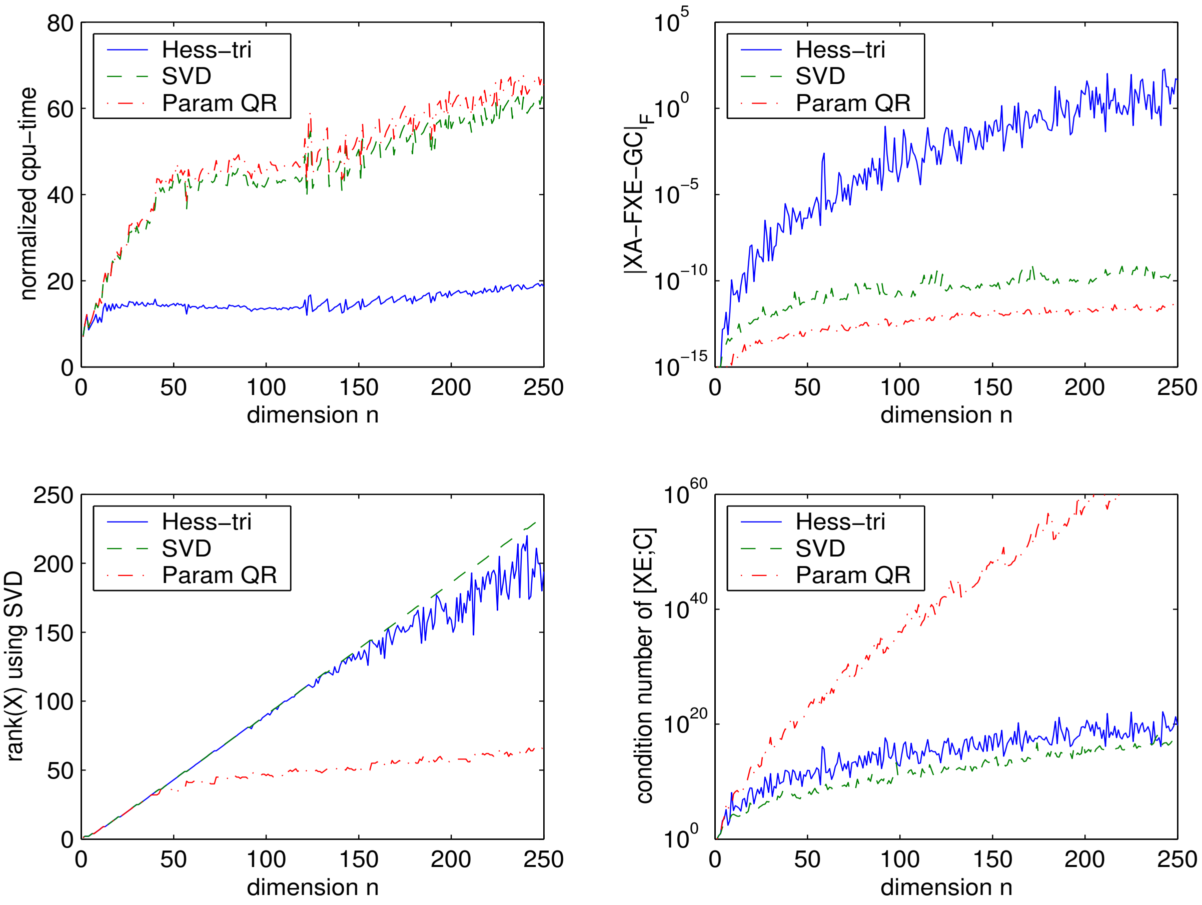

Figure 4.3: Numerical Experiment with Families *pdtoep* and *Lehmer*.

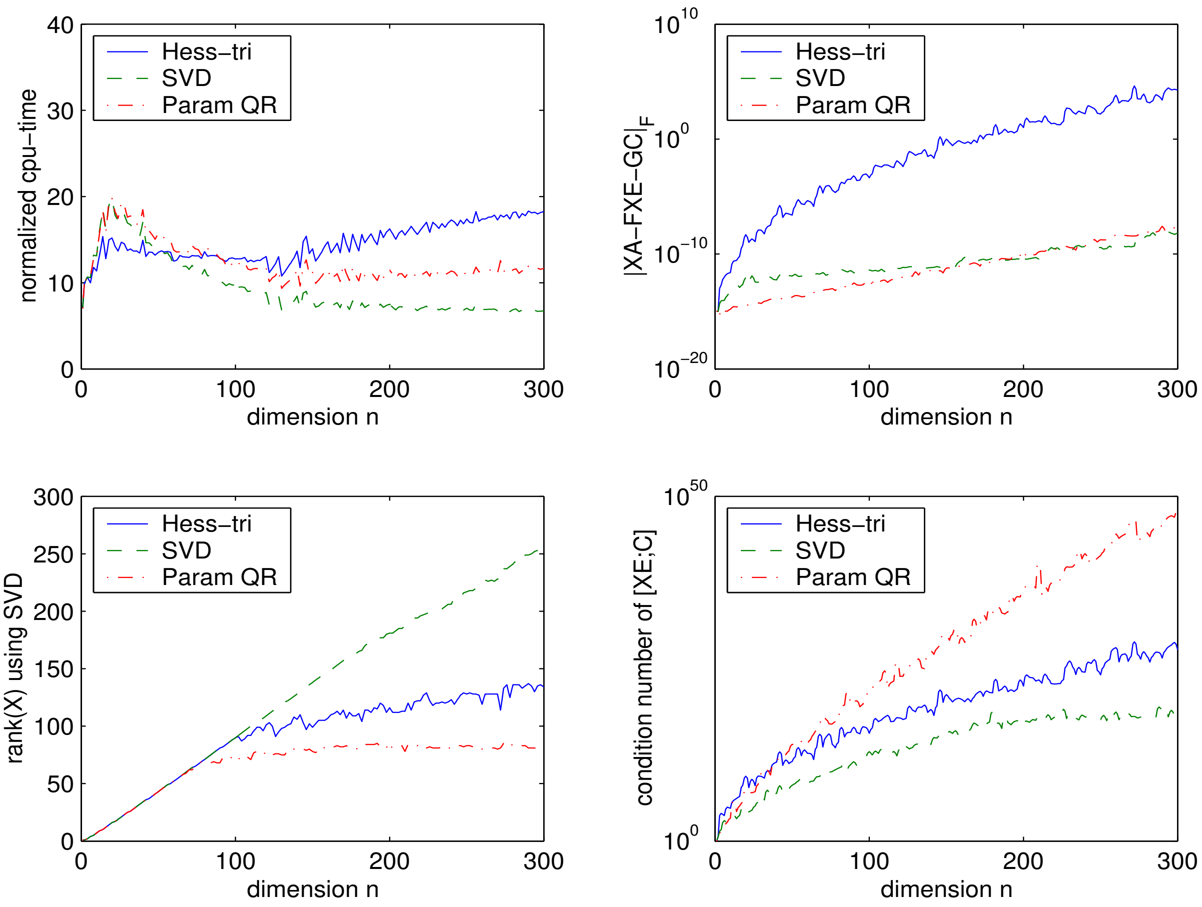

Figure 4.4: Numerical Experiment with Families *Lesp* and *pentoep*.

CHAPTER 5

Estimating the State of a Vibrating System

5.1 Introduction

In this chapter, we develop a state estimation procedure for the second-order vibrating system

$$\begin{aligned} &M\ddot{q}(t) + D\dot{q}(t) + Kq(t) = Bu(t) \\ &y(t) = C_1 q(t) + C_2 \dot{q}(t) \end{aligned} \tag{5.1.1}$$

where $M, D, K \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and C_1, $C_2 \in \mathbb{R}^{r\times n}$, arising in vibration and structural dynamics. The matrices M, D and K are usually symmetric and are called, respectively, the *mass*, *damping* and *stiffness* matrices. The vectors $q(t)$, $\dot{q}(t)$ and $\ddot{q}(t)$ are, respectively, the *displacement*, *velocity* and *acceleration* vectors. We will assume in this chapter that $M = M^T$, $D = D^T$ and $K = K^T$. Other assumptions on them will be made as the situations warrant.

Motivation for state estimation of vibrating systems of the form (5.1.1) comes from stabilization and control of structures such as bridges, buildings, automobiles, air or space crafts. To improve the performance of such systems, frequently a state feedback control law

$$u(t) = v(t) + K_1 q(t) + K_2 \dot{q}(t) \tag{5.1.2}$$

is applied. Here $K_1, K_2 \in \mathbb{R}^{m\times n}$ are feedback matrices computed to meet certain performance criterion. The vector $v(t)$, that represents an external controlling action in the system, is frequently set to zero in the real-life applications.

When the control law (5.1.2) is applied, the open-loop system (5.1.1) becomes the closed-loop system

$$\begin{aligned} &M\ddot{q}(t) + (D - BK_2)\dot{q}(t) + (K - BK_1)q(t) = Bv(t) \\ &y(t) = C_1 q(t) + C_2 \dot{q}(t) \end{aligned} \quad . \tag{5.1.3}$$

In circumstances in which the designer does not have information about the state $q(t)$ or the state $\dot{q}(t)$, or some of their components, these quantities have to be estimated so that the control law (5.1.2) can be applied to the system.

One natural way to estimate the states of (5.1.1), is to first reduce this system to a descriptor system and then apply the techniques for state estimation of such systems, developed in Chapter 4. After that, the estimates for $q(t)$ and $\dot{q}(t)$ can be recovered.

5.2 Controllability and Observability of the Second Order System

The controllability and observability of the second-order system (5.1.1) are defined with respect to the respective concepts of its first-order realizations, which are given in Section 2.1.

We recall from section 2.1 that $P(\lambda) = \lambda^2 M + \lambda D + K$, $\lambda \in \mathbb{C}$, is the quadratic pencil associated with (5.1.1). A complex number μ is said to be an eigenvalue of $P(\lambda)$ if there exists a nonzero vector x, called eigenvector, such that $P(\mu)x = 0$.

The criteria presented here use the eigenvalues of this quadratic pencil to decide controllability and observability of (5.1.1). These criteria were derived by Laub and Arnold [46]. Other definitions and criteria for controllability and observability of (5.1.1) for can be found in [41].

Lemma 5.1 *: Criterion for Controllability of a Second Order System*

The second order system (5.1.1) is *controllable* if and only if

$$rank\left(\begin{bmatrix} P(\mu) & B \end{bmatrix}\right) = n \tag{5.2.4}$$

for each eigenvalue μ of the quadratic pencil $P(\lambda)$.

Lemma 5.2 *: Criterion for Observability of a Second Order System*

The second order system (5.1.1) is *observable* if and only if

$$rank\left(\begin{bmatrix} P(\mu) \\ C_1 + \mu C_2 \end{bmatrix}\right) = n \tag{5.2.5}$$

for each eigenvalue μ of the quadratic pencil $P(\lambda)$.

Proof of the Lemmas: See [46] for the proof of the results above.

Example 5.1 *: Consider the second order system (5.1.1) with*

$$M = 10^5 \begin{bmatrix} 0.0038 & 0. & 0. \\ 0. & 674247.8000 & 0. \\ 0. & 0. & 50.3170 \end{bmatrix},$$

$$D = 10^1 \begin{bmatrix} 0. & -263459.8120 & 0. \\ 263459.8120 & 0. & 0. \\ 0. & 0. & 0. \end{bmatrix}$$

$$K = \begin{bmatrix} -77.2095 & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 344868.8939 \end{bmatrix}, \quad B = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 13400. & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0.4000 & 0.1000 & 0.2000 \\ 0. & 0.3000 & 0. \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0.2500 & 0.1000 & 0.2500 \\ 0.1000 & 0.2000 & 0.1000 \end{bmatrix}.$$

The associated pencil $P(\lambda)$ has eigenvalues $\pm 0.$, $\pm 0.2618i$ and $\pm 0.2618i$.

Controllability : Since the matrix B already has rank $n = 3$, clearly this system is controllable.

Observability : Table 5.1 shows, for each of the eigenvalues of the pencil $P(\lambda)$, the smallest singular value, denoted by $\sigma_o(\mu)$, of the matrix in (5.2.5):

Table 5.1: Smallest Singular Values for the Observability Criterion.

μ	0	$0.2618i$	$-0.2618i$
$\sigma_o(\mu)$	0.3162	0.0114	0.0114

and therefore we conclude that this system is also observable.

Example 5.2 *: Consider the second order system (5.1.1) with*

$$M = \begin{bmatrix} 0.2000 & 0. \\ 0. & 0.0100 \end{bmatrix}, \quad D = \begin{bmatrix} 0. & 0. \\ 0. & 0.2500 \end{bmatrix}, \quad K = \begin{bmatrix} 24. & -3.6000 \\ -3.6000 & 0.5400 \end{bmatrix}$$

$$B = \begin{bmatrix} 0. \\ 1. \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0. & 0. \\ 0. & 0. \end{bmatrix}.$$

The associated pencil $P(\lambda)$ has eigenvalues $0.$, $-0.9545 \pm 11.3582i$ and -23.0911.

Controllability: Table 5.2 shows, for each of the eigenvalues of the pencil $P(\lambda)$, the smallest singular value, denoted by $\sigma_c(\mu)$, of the matrix in (5.2.4):

Table 5.2: Smallest Singular Values for the Controllability Criterion.

μ	0	$-0.9545 + 11.3582i$	$-0.9545 - 11.3582i$	-23.0911
$\sigma_c(\mu)$	0.9889	0.7867	0.7867	0.9996

and therefore we conclude that this system is controllable.

Observability : Since the matrix C_1 already has rank $n = 2$ and $C_2 = 0$, clearly this system is observable.

5.3 State Estimation via Reduction to a Descriptor System

The system (5.1.1) can be casted into a descriptor system of the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + B_0 u(t) \\ y(t) &= Cx(t) \end{aligned} \tag{5.3.6}$$

where $x(t) = \begin{bmatrix} \dot{q}(t) & q(t) \end{bmatrix}^T$. This can be done in several ways. In our development, the matrices E, A, B_0 and C in (5.3.6) are defined as follows:

1. If M is nonsingular, then

$$E = \begin{bmatrix} 0 & M \\ M & D \end{bmatrix}, \; A = \begin{bmatrix} M & \\ & -K \end{bmatrix}, \; B_0 = \begin{bmatrix} 0 \\ B \end{bmatrix} u(t), \; C = \begin{bmatrix} C_2 & C_1 \end{bmatrix}. \tag{5.3.7}$$

2. If K is nonsingular, then

$$E = \begin{bmatrix} M & \\ & -K \end{bmatrix}, \; A = \begin{bmatrix} -D & -K \\ -K & 0 \end{bmatrix}, \; B_0 = \begin{bmatrix} B \\ 0 \end{bmatrix} u(t), \; C = \begin{bmatrix} C_2 & C_1 \end{bmatrix}. \tag{5.3.8}$$

We note that the matrices E and A in both (5.3.7) and (5.3.8) are symmetric, since M, D and K are symmetric. For other formulations, see [27].

In order to estimate the states $x(t)$ of the system (5.1.1), the descriptor Sylvester-observer equation

$$XA - FXE = GC \tag{5.3.9}$$

will be solved with the matrices E, A, B_0 and C defined either by (5.3.7) or (5.3.8).

After solving (5.3.9) for matrices X, F and G, approximations $\hat{q}(t)$ and $\hat{d}(t)$ of $q(t)$ and $\dot{q}(t)$ can be found by solving either

$$XE \begin{bmatrix} \hat{d}(t) \\ \hat{q}(t) \end{bmatrix} = z(t), \tag{5.3.10}$$

in the full-order case ($s = 2n$), or

$$\begin{bmatrix} XE \\ C \end{bmatrix} \begin{bmatrix} \hat{d}(t) \\ \hat{q}(t) \end{bmatrix} = \begin{bmatrix} z(t) \\ y(t) \end{bmatrix} \tag{5.3.11}$$

in the reduced-order case ($s = 2n - r$), where $z(t) \in \mathbb{R}^s$ is the solution of the differential equation

$$\dot{z}(t) = Fz(t) + Gy(t) + XB_0u(t) \tag{5.3.12}$$

satisfying any initial condition $z(0) = z_0$.

As seen in chapter 4, solving (5.3.9) requires that $\Omega(A, E) \bigcap \mathcal{S} = \emptyset$. Our next step is to show how the eigenvalues of the pair (A, E) are related to the eigenvalues of the pencil $P(\lambda) = \lambda^2 M + \lambda D + K$ when representations (5.3.7) and (5.3.8) are used.

Consider the case that M is nonsingular and the matrices A and E are defined by (5.3.7). Let μ be a generalized eigenvalue of (A, E). By definition, there is a nonzero vector $w = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T$ such that $Aw = \mu Ew$; that is,

$$\begin{bmatrix} M & \\ & -K \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \mu \begin{bmatrix} 0 & M \\ M & D \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

This gives

$$Mw_1 = \mu M w_2 \tag{5.3.13}$$

$$\mu M w_1 + \mu D w_2 + K w_2 = 0. \tag{5.3.14}$$

Since M is nonsingular, (5.3.13) gives $w_1 = \mu w_2$ and substituting this in (5.3.14) gives

$$\mu^2 M w_2 + \mu D w_2 + K w_2 = 0. \tag{5.3.15}$$

Now consider the case that K is nonsingular and the matrices A and E are defined by (5.3.8). Let μ be a generalized eigenvalue of (A, E). By definition, there is a nonzero vector $w = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T$ such that $Aw = \mu E w$, that is,

$$\begin{bmatrix} -D & -K \\ -K & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \mu \begin{bmatrix} M & \\ & -K \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

This gives

$$\mu M w_1 + D w_1 + K w_2 = 0 \tag{5.3.16}$$

$$K w_1 = \mu K w_2. \tag{5.3.17}$$

Since now K is nonsingular, again we have $w_1 = \mu w_2$. Substituting this in (5.3.16) gives (5.3.15) once more.

Therefore, the generalized eigenvalues μ of a pair (A, E) satisfying either (5.3.7) or (5.3.8) are eigenvalues of the pencil $P(\lambda)$.

Next, we show how the estimates of the state variables $q(t)$ and $\dot{q}(t)$ can be used to implement a feedback control law in a second-order system.

Algorithm 5.3 : *State Feedback Control via Reduced Order State Estimation for Second-order Systems*

Input: System matrices $M, D, K \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $, C_1, C_2 \in \mathbb{R}^{r \times n}$ of the system (5.1.1).

Assumptions: System (5.1.1) is observable and C_1 and C_2 are matrices with full rank and at least one of the matrices M and K is nonsingular.

Step 1: If a full order estimator is sought, set $s = 2n$, otherwise, set $s = 2n - r$. If M is nonsingular, compute the matrices A, E, B_0 and C using (5.3.7). Otherwise, use (5.3.8).

Step 2: Choose a self-conjugate set $\mathcal{S} \in \mathbb{C}^s$ with elements in the left side of the complex plane such that these elements are different from the eigenvalues of the pencil $P(\lambda) = \lambda^2 M + \lambda D + K$.

Step 3: Solve the Sylvester-observer equation (5.3.9) using any of the methods described in Chapter 4.

Step 4: From the knowledge of the matrices M, D, K and B, obtain feedback matrices K_1 and K_2 according to the design requirements and then form the closed-loop system (5.1.3) using K_1, K_2 and the knowledge of the external force $v(t)$.

Step 5: Solve the system of differential equations (5.3.12) with a chosen initial condition and then find the approximations $\hat{q}(t)$ and $\hat{d}(t)$ of $q(t)$ and $\dot{q}(t)$ by solving (5.3.10) or (5.3.11), according to whether a full-order state estimate or a reduced-order estimate is desired.

Step 6: Implement now the control law (5.1.3), using the estimate $\hat{d}(t)$ and $\hat{q}(t)$, in place of $\dot{q}(t)$ and $q(t)$, respectively.

$\square$

5.4 Applications to Feedback Control of Vibrating Structures

In this section, we present some applications of Algorithm 5.3.

Example 5.4 : *State Feedback Control of a Communication Satellite*

This example is taken from [15]. A communication satellite of mass m orbits around the earth with altitude specified by spherical coordinates $r(t)$, $\theta(t)$ and $\phi(t)$.

The orbit is controlled by three orthogonal thrusts $u_r(t)$, $u_\theta(t)$ and $u_\phi(t)$.

The dynamics of this model, given by the Newton's Second Law, is

$$\begin{aligned} m\ddot{r} &= mr\dot{\theta}^2\cos^2\phi + mr\dot{\phi}^2 - k/r^2 + u_r \\ mr^2\ddot{\theta} &= -2mr\dot{r}\dot{\theta} + 2mr^2\dot{\theta}\dot{\phi}\sin\phi/\cos\phi + ru_\theta/\cos\phi \\ mr\ddot{\phi} &= -mr\dot{\theta}^2\cos\phi\sin\phi - 2m\dot{r}\dot{\phi} + u_\phi. \end{aligned} \tag{5.4.18}$$

A stationary solution, corresponding to the desired circular equatorial orbit, is given by

$$\begin{bmatrix} r_0(t) & \theta_0(t) & \phi_0(t) \end{bmatrix}^T = \begin{bmatrix} r_0 & \omega_0 t & \omega_0 \end{bmatrix}^T \tag{5.4.19}$$

where r_0, ω_0 and k are constants such that $r_0\omega_0^2 = k/\omega_0^2$. Once in orbit, the satellite will remain there if no disturbances happen. However, if it deviates from its orbit, the thrusts are applied to push it back to the orbit.

Let

$$q(t) = \begin{bmatrix} r(t) - r_0 & \theta(t) - \theta_0 & \phi(t) - \phi_0 \end{bmatrix}^T \tag{5.4.20}$$

denote the deviation from the orbit. If the perturbation is very small, then equations (5.4.18) can be linearized, and a control system (5.1.1) with three inputs and two outputs can be derived, where

$$M = \begin{bmatrix} m & & \\ & mr_0^2 & \\ & & mr_0 \end{bmatrix}, \quad D = \begin{bmatrix} & -2mr_0\omega_0 & \\ 2mr_0\omega_0 & & \\ & & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} -3m\omega_0^2 & & \\ & 0 & \\ & & mr_0\omega_0^2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & & \\ & r_0 & \\ & & 1 \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_r(t) \\ u_\theta(t) \\ u_\phi(t) \end{bmatrix}.$$

Taking the parameters of this model to be $m = 375.5$ kg, $r_0 = 13400$ m and $\omega_0 = 0.2618$ rad/sec, the matrices above assume the values given in Example 5.1. In particular, choosing the matrices C_1 and C_2 also as defined in Example 5.1 ensures that this model is controllable and observable.

In the descriptor form (5.3.6), where E , A , B_0 and C are given by (5.3.7), we have

$$E = 10^5 \begin{bmatrix} 0. & 0. & 0. & 0.0038 & 0. & 0. \\ 0. & 0. & 0. & 0. & 674247.8000 & 0. \\ 0. & 0. & 0. & 0. & 0. & 50.3170 \\ 0.0038 & 0. & 0. & 0. & -26.3460 & 0. \\ 0. & 674247.8000 & 0. & 26.3460 & 0. & 0. \\ 0. & 0. & 50.3170 & 0. & 0. & 0. \end{bmatrix}$$

$$A = 10^5 \begin{bmatrix} 0.0038 & 0. & 0. & 0. & 0. & 0. \\ 0. & 674247.8000 & 0. & 0. & 0. & 0. \\ 0. & 0. & 50.3170 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0.0008 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & -3.4487 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.2500 & 0.1000 & 0.2500 & 0.4000 & 0.1000 & 0.2000 \\ 0.1000 & 0.2000 & 0.1000 & 0. & 0.3000 & 0. \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0. & 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 0. & 13400. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1. \end{bmatrix}^T .$$

As already mentioned in Example 5.1, the associated pencil $P(\lambda)$ has eigenvalues $\pm 0.$, $\pm 0.2618i$ and $\pm 0.2618i$. Therefore, this system is not asymptotically stable.

In order to stabilize this system, we apply the state feedback control law (5.1.2), with

$$K_1 = 10^1 \begin{bmatrix} -100.5304 & -118169.1827 & 0. \\ -1.4195 & -906602.3462 & 0. \\ 0. & 0. & -77328.6662 \end{bmatrix},$$

$$K_2 = 10^2 \begin{bmatrix} -12.1183 & -30058.1247 & 0. \\ 1.9895 & -139516.5952 & 0. \\ 0. & 0. & -50317. \end{bmatrix}.$$

With the definitions of K_1 and K_2 above, the eigenvalues of the closed-loop system (5.1.3) are $-1/3, -2/3, -1, -4/3, -5/3, -2$.

Taking $v(t) = \delta(t)$, the Dirac delta pulse, we study the impulse response of the system under three conditions:

- no feedback control; that is, considering the open-loop system (5.1.1) with $u(t) = v(t) = \delta(t)$
- state feedback control using (5.1.2) and the actual states
- state feedback control using (5.1.2) and the estimated states.

Solving the Sylvester-observer equation (5.3.9) gives

$$X = \begin{bmatrix} 0.5184 & 0. & 0. & -0.8551 & 0. & -0.0001 \\ 0.8551 & 0. & 0. & 0.5184 & 0. & 0. \\ -0.8975 & 0. & 0. & -0.4410 & 0. & 0. \\ 0.4410 & 0. & 0. & -0.8975 & 0. & -0.0001 \end{bmatrix}$$

$$F = \begin{bmatrix} -7.0237 & -0.1198 & -1.0194 & 6.5465 \\ -0.1929 & -7.9763 & -8.3075 & -1.2937 \\ 0. & 0. & -5.9851 & 0.0292 \\ 0. & 0. & 0.5009 & -5.0149 \end{bmatrix},$$

$$G = \begin{bmatrix} 0. & 0. \\ 0. & 0. \\ -5026.3685 & -617.4799 \\ 2555.7662 & -20804.2752 \end{bmatrix}$$

$$\Omega(F) = \{ \begin{array}{cccc} -5 & -6 & -7 & -8 \end{array} \}$$

Figure 5.1 shows the norm of the impulse responses of the system in the three cases defined above. The initial condition for the observer is $z(0) = 0$. The plot shows that the control using estimated states is slower than the one using the actual states, but is clearly successful in bringing the satellite back to the desired orbit.

Figure 5.2 shows how the norm of the difference between the actual states and the estimated states evolves in time. The plot shows that $|x(t) - x_e(t)| \to 0$ as the time $t \to \infty$, where $x_e(t) = \hat{x}(t)$ are the estimated states.

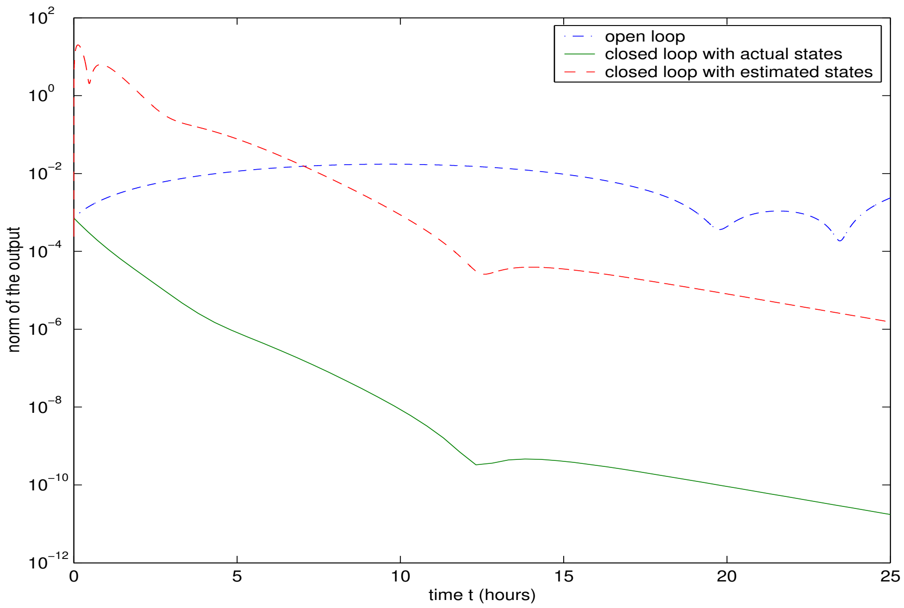

Figure 5.1: Comparison of the Norms of the Impulse Responses.

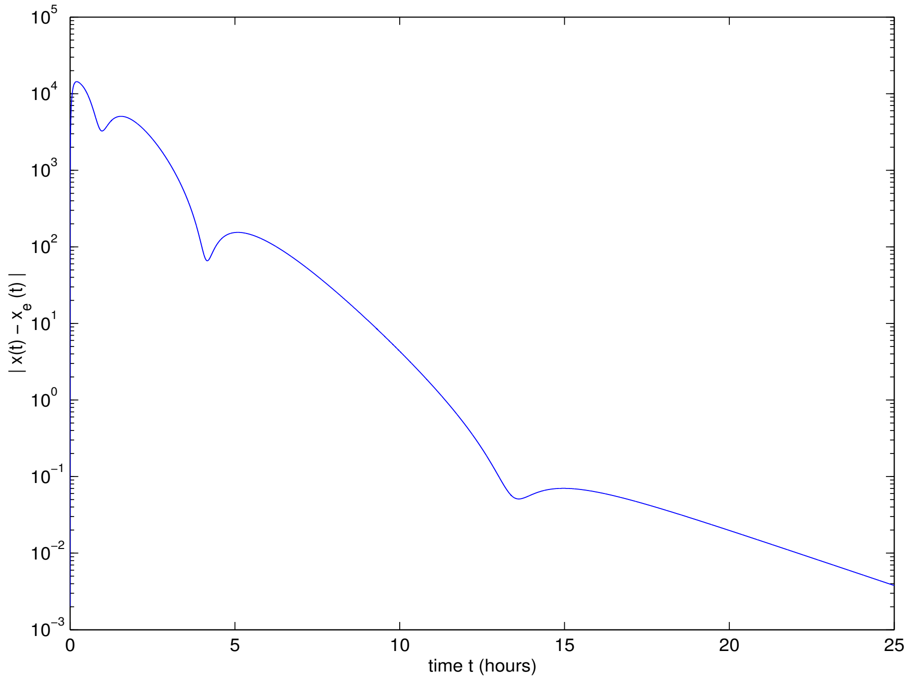

Figure 5.2: Norm of the Difference Between Actual States and Estimated States.

Example 5.5 : *Control of a Printer Belt System*

The following example was taken from [34]. Here a printer that uses a belt drive to move the printing device laterally across the page being printed is considered.

The model, proposed in [34], considers the following variables:

- the torque $T_m(t)$ produced by the motor, which is assumed to have a negligible inductance L
- the undesired torque $T_d(t)$ due to causes like wearing of the motor or disturbance of the electrical voltage
- the displacement $y(t)$ of the printing device
- the angular rotation $\theta(t)$ of the motor shaft
- the output voltage $v_1(t)$ of the sensor that measures $y(t)$

- the output voltage $v_2(t)$ of the controller used

along with the following parameters

- the total inertia J of the motor and pulley
- the friction constant b of motor and pulley
- the radius r of the pulley
- the spring constant k of the belt
- the parameters k_1 and k_2 of the controller
- the mass m of the printing device
- the field resistance R of the motor.

The dynamics of this model, according to Newton's Second Law, is

$$\begin{aligned} &m\ddot{y}(t) - 2kr\theta(t) + 2kry(t) = 0 \\ &J\ddot{\theta}(t) + b\dot{\theta}(t) + 2kr^2\theta(t) - 2kry(t) = T_m - T_d. \end{aligned} \tag{5.4.21}$$

Therefore, this model can be described by (5.1.1), where

$$\begin{aligned} &M = \begin{bmatrix} m & \\ & J \end{bmatrix}, \; D = \begin{bmatrix} 0 & \\ & b \end{bmatrix}, \; K = \begin{bmatrix} 2k & -2kr \\ -2kr & 2kr^2 \end{bmatrix} \\ &B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; C_1 = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, \; C_2 = \begin{bmatrix} 0 & \\ & 0 \end{bmatrix} \\ &x(t) = \begin{bmatrix} \dot{q}(t) & q(t) \end{bmatrix}^T. \end{aligned} \tag{5.4.22}$$

The controller is designed to control the behavior of this system when faced with the undesirable torque T_d. The position sensor, the controller and the torque T_m

produced by the motor are, respectively, described by

$$\begin{aligned} v_1(t) &= k_1 y(t) \\ v_2(t) &= -\left[k_2 \dot{v}_1(t) + k_3 v_1(t)\right] \\ T_m(t) &= \tfrac{K_m}{R} v_2(t). \end{aligned} \tag{5.4.23}$$

Therefore, the state feedback law (5.1.2) can be applied, where $u(t) = T_m(t) - T_d(t)$ and

$$v(t) = -T_d(t)\ ,\ K_1 = \begin{bmatrix} \frac{-K_m k_2 k_1}{R} \\ 0 \end{bmatrix}\ ,\ K_2 = \begin{bmatrix} \frac{-K_m k_3 k_1}{R} \\ 0 \end{bmatrix}. \tag{5.4.24}$$

Table 5.3 gives the numerical values for the parameters of this problem.

Table 5.3: Numerical Values for the Parameters of Example 5.5.

$m = 0.2$ kg	$r = 0.15$ m	$b = 0.25$ N-ms/rad
$k_1 = 1.0$ V/m	$k_2 = 0.1$	$k_3 = 0.001$
$J = 1.01$ kg-m^2	$R = 2\ \Omega$	$K_m = 2$ N-m/A

With these values, the matrices M , D , K , B become the same as those in Example 5.2. In particular, by choosing the matrices C_1 and C_2 as in Example 5.2, we ensure that this system is controllable and observable. The matrices K_1 and K_2 are chosen so that the system (5.1.3) is assymptotically stable. They are

$$K_1 = \begin{bmatrix} -0.1 \\ 0 \end{bmatrix}\ ,\ K_2 = \begin{bmatrix} -0.001 \\ 0 \end{bmatrix}.$$

Note that the open-loop eigenvalues are: $0.$, $-0.9545 \pm 11.3582i$ and -23.0911 and the above feedback matrices K_1 and K_2 make a closed-loop system with eigenvalues -0.0602 , $-0.9292 \pm 11.3464i$, -23.0814.

The matrices of the corresponding descriptor system, computed using (5.3.7), are:

$$E = \begin{bmatrix} 0. & 0. & 0.2000 & 0. \\ 0. & 0. & 0. & 0.0100 \\ 0.2000 & 0. & 0. & 0. \\ 0. & 0.0100 & 0. & 0.25 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 1. \end{bmatrix}$$

$$A = \begin{bmatrix} 0.2000 & 0. & 0. & 0. \\ 0. & 0.0100 & 0. & 0. \\ 0. & 0. & -24. & 3.6000 \\ 0. & 0. & 3.6000 & -0.5400 \end{bmatrix}, \quad C = \begin{bmatrix} 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \end{bmatrix}.$$

As in Example 5.1, we take $v(t) = \delta(t)$, to compare the impulse response of this system in three cases:

- no feedback, that is, considering the open-loop system (5.1.1) with $u(t) = v(t)$
- state feedback control using the actual states
- state feedback control using the estimated states.

Solving the Sylvester-observer equation (5.3.9) with $\Omega(F) = \{ \; -2 \quad -3 \; \}$ gives

$$X = \begin{bmatrix} 0.1256 & 0.8867 & -0.0629 & -0.4404 \\ 0.6980 & -0.3774 & -0.3256 & -0.5142 \end{bmatrix}, \quad F = \begin{bmatrix} -2.0180 & 0.0041 \\ 4.3382 & -2.9820 \end{bmatrix},$$

$$G = \begin{bmatrix} -0.0261 & -0.1923 \\ 6.2698 & -0.8497 \end{bmatrix}.$$

Figure 5.3 shows the norm of the impulse responses of the system in the three cases defined above. The initial condition for the observer is $z(0) = 0$. From the plot, it can be seen that the controlled impulse response using estimated states approximates very accurately the controlled impulse response using the actual states.

Figure 5.4 shows how the norm of the difference between the actual states and the estimated states evolves in time. The plot shows that $|x(t) - x_e(t)| \to 0$ as the time $t \to \infty$, where $x_e(t) = \hat{x}(t)$ are the estimated states.

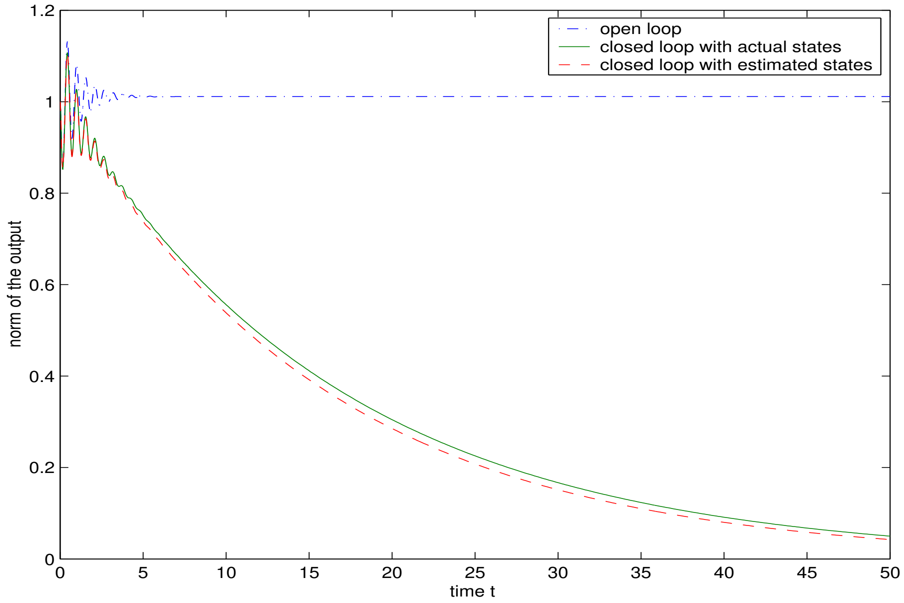

Figure 5.3: Comparison of the Norms of the Impulse Responses.

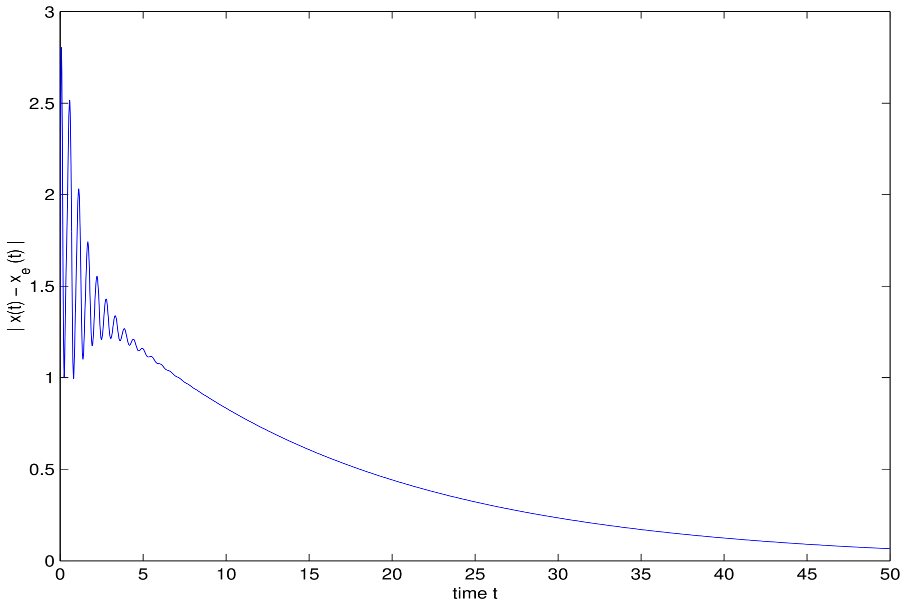

Figure 5.4: Norm of the Difference Between Actual States and Estimated States.

CHAPTER 6

Finite-Element Model Updating

6.1 Introduction

Recall from our discussions in Chapter 1 that the Finite Element (FEM) Model Updating problem is the problem of updating the symmetric finite-element generated model

$$M\ddot{q}(t) + D\dot{q}(t) + Kq(t) = 0, \tag{6.1.1}$$

represented by the triple (M, D, K), in such a way that

(i) the updated model $(\tilde{M}, \tilde{D}, \tilde{K})$ is also symmetric

(ii) a set of natural frequencies and corresponding mode shapes, measured from a real-life structure, is contained in the updated model

(iii) the remaining natural frequencies and corresponding mode shapes of the updated model are the same as those of the original model.

Because of its immense practical importance, this problem has been well-studied in the recent years. As a consequence, there exist numerous methods for its solution.

Unfortunately, the problem, as defined above, has not been completely solved yet.

The existing methods can be classified in the following broad categories:

(i) Direct Methods

(ii) Parametric Iterative Methods

(iii) Frequency Domain Methods.

An excellent account of these methods can be found in the authoritative book [36].

Most of the optimization-based direct methods deal only with the undamped case ($D = 0$). On the other hand, the control-theoretic direct methods deal with the damped second-order model and try to preserve symmetry, as much as possible, by using optimization techniques.

Neither of these methods, however, can guarantee the invariance of the natural frequencies and corresponding mode shapes that are required to remain unchanged after the updating.

Another difficulty with all these methods is the problem of handling incomplete measured data. Usually, the degree of freedom of the measured data is much smaller than that of the finite-element model. Thus, to compare these measured data with those of the finite-element model, one needs to either reduce the finite-element model (model reduction) or complete the measured data (mode shape expansion). For details, see [36].

In this chapter, we develop a new method for model updating of an undamped second-order model with incomplete measured data (Algorithm 6.3.2). Our method does not require any model reduction. Furthermore, it can be applied to a symmetric positive semidefinite model.

Our development depends upon results on orthogonality relations between the eigenvectors of a symmetric positive semidefinite quadratic pencil $P(\lambda) = \lambda^2 M + \lambda D + K$. Such relations are derived in this chapter (Theorem 6.2.1).

Finally, we consider the solution of a problem, called *Eigenvalue Embedding* problem or *Model Tuning* problem [35], which is a particular case of the model updating problem.

The eigenvalue embedding strategy can be applied in the particular case when only natural frequencies are concerned for updating. The Eigenvalue Embedding problem is defined as follows:

Given a symmetric model (M, D, K), a subset $\{\lambda_k\}, k = 1, \ldots, m$ of its eigenvalue set, and a set of measured natural frequencies $\{\mu_k\}, k = 1, \ldots, m$, find an updated model $(\tilde{M}, \tilde{D}, \tilde{K})$ such that

- the set $\{\lambda_k\}, k = 1, \ldots, m$ is replaced by $\{\mu_k\}, k = 1, \ldots, m$ as eigenvalues of the new model $(\tilde{M}, \tilde{D}, \tilde{K})$
- the remaining subset of $2n - m$ eigenvalues (the natural frequencies) of the new symmetric model $(\tilde{M}, \tilde{D}, \tilde{K})$ are the same as those of (M, D, K).

In this chapter, we present a new method (Theorem 6.4.1 and Algorithm 6.4.2) for eigenvalue embedding of a set of real numbers, either individually or simultaneously, of a symmetric positive semidefinite second-order model. The method generalizes an earlier method due to Carvalho, et al. [11] for eigenvalue embedding of a symmetric positive definite model.

The organization of this chapter is as follows:

In Section 6.2, we present the mathematical results that support the development of our methods. The concept of orthogonality relations among eigenvectors, established in [21], is revisited with the purpose of generalizing these results for symmetric positive semidefinite pencils, so that these results can be applied to update a symmetric semidefinite model.

In Section 6.3, we present methods for finite-element model updating which are guaranteed to preserve the symmetry and the unmeasured natural frequencies and mode shapes of the model. These methods assume that the available set of modal parameters can be totally or just partially obtained from measurement.

In Section 6.4, we present a method for finite-element model updating through eigenvalue embedding.

6.2 Orthogonality Relations Between the Eigenvectors of the Symmetric Quadratic Pencil

It is well-known [14, 20], that the eigenvectors of a symmetric matrix $A \in \mathbb{R}^{n \times n}$, can be chosen to be orthonormal.

Similarly, the eigenvectors of a symmetric definite pencil $Q(\lambda) = \lambda K - M$ (that is, when $M = M^T > 0$ and $K = K^T$) can be chosen such that

$$x_i^T M x_j = \delta_{ij} \ , \ x_i^T K x_j = \delta_{ij} \lambda_i \tag{6.2.2}$$

for $i, j = 1, 2, \ldots, n$, where δ_{ij} is the Kronecker Delta.

These orthogonality relations were generalized by Datta, Elhay and Ram [21] to the symmetric positive definite quadratic pencil $P(\lambda) = \lambda^2 M + \lambda D + K$; this generalization formed the basis of a new approach for partial eigenvalue and partial eigenstructure assignment problems [22, 24, 25, 26].

In this section, we extend these relations to the two important cases:

1. The undamped symmetric positive semidefinite quadratic pencil $Q(\lambda) = \lambda^2 M + K$.

2. The symmetric positive semidefinite quadratic pencil $P(\lambda) = \lambda^2 M + \lambda D + K$.

These relations are then used to derive our proposed algorithm for model updating of the undamped symmetric positive semidefinite model (M, D, K).

6.2.1 Orthogonality Relations for Symmetric Positive Semidefinite Undamped Quadratic Pencil

Consider the symmetric semidefinite undamped quadratic pencil $P(\lambda) = \lambda^2 M + K$. Let λ be a finite eigenvalue of $P(\lambda)$ and let x be the corresponding eigenvector, then we must have $\lambda^2 Mx = -Kx$. Pre-multiplication by x^T gives

$$\lambda^2 x^T Mx = -x^T Kx.$$

Since the matrices M and K are symmetric positive semidefinite, λ^2 is a non-positive real number. This implies that x is a real vector and that λ is either zero or a purely imaginary number. Therefore, every eigenvector x is real and corresponds to two purely imaginary eigenvalues $\lambda = \pm i\ \alpha, \alpha \in \mathbb{R}$.

Thus, we can have a compact representation (Λ, X) of the finite eigenstructure of the pencil $P(\lambda)$ satisfying

$$MX\Lambda^2 + KX = 0 \tag{6.2.3}$$

where $X \in \mathbb{R}^{n\times n}$ and $\Lambda \in \mathbb{C}^{n\times n}$ ($\Lambda^2 \in \mathbb{R}^{n\times n}$ is diagonal with non-positive entries), under the convention that every pair of eigenvalues $\lambda = \pm i\ \alpha$ corresponding to an eigenvector x, is represented only once when assembling in (6.2.3). This compact representation makes the matrices X and Λ have at most n rows and n columns, instead of their usual dimensions for the quadratic eigenvalue problem for the pencil $Q(\lambda) = \lambda^2 M + K$.

Transposing (6.2.3) and using the symmetry of M and K, we obtain

$$(\Lambda^2)^T X^T M + X^T K = 0. \tag{6.2.4}$$

Pre-multiplying (6.2.3) by X^T, post-multiplying (6.2.4) by X and combining gives

$$(X^T MX)\Lambda^2 - (\Lambda^2)^T (X^T MX) = 0. \tag{6.2.5}$$

The next theorem gives an important set of orthogonality relations for the symmetric semidefinite undamped quadratic pencil:

Theorem 6.2.1 : *Orthogonality Relations for Symmetric Semidefinite Undamped Quadratic Pencil*

Let $P(\lambda) = \lambda^2 M + K$ be a symmetric semidefinite pencil with distinct nonzero eigenvalues and let (Λ, X) be a compact representation of the finite eigenstructure of this pencil. Then the matrices D_1 and D_2 defined by

$$D_1 = X^T M X \tag{6.2.6}$$

and

$$D_2 = X^T K X. \tag{6.2.7}$$

are diagonal and

$$D_2 = -D_1 \Lambda^2. \tag{6.2.8}$$

Proof : By hypothesis, Λ is a diagonal matrix with distinct nonzero entries. Since the compact representation is adopted, Λ^2 is also a diagonal matrix with distinct nonzero entries. By (6.2.5), the matrices D_1 and Λ^2 commute, and therefore D_1 is also a diagonal matrix.

Again, pre-multiplying (6.2.3) by X^T and rearranging gives

$$X^T M X \Lambda^2 + X^T K X = 0$$

from which (6.2.8) follows. Since D_1 and Λ^2 are diagonal matrices, from (6.2.8) we conclude that D_2 is also a diagonal matrix.

□

Corollary 6.2.1: Suppose that the hypothesis of Theorem 6.2.1 hold, let the matrices X and Λ be partitioned as

$$X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}, \ \Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \tag{6.2.9}$$

and assume that Λ_1 and Λ_2 do not have a common entry.

Then

$$X_1^T M X_2 = 0 \tag{6.2.10}$$

and

$$X_1^T K X_2 = 0. \tag{6.2.11}$$

Proof: From (6.2.6),

$$\begin{bmatrix} D_{11} & 0 \\ 0 & D_{12} \end{bmatrix} = D_1 = \begin{bmatrix} X_1 & X_2 \end{bmatrix}^T M \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} X_1^T M X_1 & X_1^T M X_2 \\ X_2^T M X_1 & X_2^T M X_2 \end{bmatrix}$$

and (6.2.10) follows. A similar argument proves (6.2.11).

Example 6.2.1: Consider the symmetric positive semidefinite pencil $P(\lambda) = \lambda^2 M + K$, where

$$M = \begin{bmatrix} 1.294 & 0 & 0 & 0 & 0 \\ 0 & 1.294 & 0 & 0 & 0 \\ 0 & 0 & 1.294 & 0 & 0 \\ 0 & 0 & 0 & 1.294 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$K = \begin{bmatrix} 1188.5 & 196.6 & 0 & 0 & -642.4 \\ 196.6 & 626.3 & 0 & -555.6 & 0 \\ 0 & 0 & 1188.5 & -196.6 & -546.1 \\ 0 & -555.6 & -196.6 & 626.3 & 196.6 \\ -642.4 & 0 & -546.1 & 196.6 & 4019.1 \end{bmatrix}.$$

A compact eigenpair representation is given by

$$\Lambda^2 = \begin{bmatrix} -23.6929 & & & \\ & -702.3357 & & \\ & & -943.6921 & \\ & & & -991.1000 \end{bmatrix},$$

$$X = \begin{bmatrix} 0.2045 & -1. & 1. & -0.5069 \\ -1. & 0.5775 & 0.2627 & -0.9986 \\ -0.1403 & -0.9602 & -0.9325 & -0.7470 \\ -0.9997 & -0.6475 & 0.0726 & 1. \\ 0.0625 & -0.2586 & 0.0296 & -0.2314 \end{bmatrix}.$$

Observe that this pencil has two infinite eigenvalues which are excluded from the representation above.

The matrices D_1 and D_2 are:

$$D_1 = \begin{bmatrix} 2.6668 & 0 & 0 & 0 \\ 0 & 3.4610 & -0.0000 & 0 \\ 0 & -0.0000 & 2.5153 & -0.0000 \\ 0 & 0 & -0.0000 & 3.6390 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 63.1847 & -0.0000 & 0.0000 & -0.0000 \\ -0.0000 & 2430.8139 & -0.0000 & -0.0000 \\ 0.0000 & -0.0000 & 2373.6668 & -0.0000 \\ -0.0000 & -0.0000 & -0.0000 & 3606.6303 \end{bmatrix}$$

which are clearly diagonal and verify $\|D_2 + D_1\Lambda^2\|_F = 1.6402 \times 10^{-12}$.

6.2.2 Orthogonality Relations for the Symmetric Quadratic Pencil

Consider the symmetric quadratic pencil $P(\lambda) = \lambda^2 M + \lambda D + K$. Assume, for simplicity, that M and K are positive semidefinite.

The eigenstructure decomposition of $P(\lambda)$ is

$$MX\Lambda^2 + DX\Lambda + KX = 0 \tag{6.2.12}$$

where $X \in \mathbb{C}^{n\times 2n}$ and $\Lambda \in \mathbb{C}^{2n\times 2n}$. The matrix Λ, which can be either diagonal or quasi-diagonal, contains information about the natural frequencies of the model. The columns of the matrix X contain the corresponding mode shapes. The pair (Λ, X) is said to be a **complex** representation of the eigenstructure of $P(\lambda)$ if complex numbers are used; in this case, the matrix Λ is diagonal and, of course, symmetric. Similarly, (Λ, X) is said to be a *real* representation of the eigenstructure of $P(\lambda)$ if, with the proper use of Schur bumps over the diagonal of Λ, both matrices are defined to have only real entries. In this case, Λ is quasi-diagonal.

Since the matrices M and K can be singular, the *characteristic* polynomial $p(\lambda) = det(P(\lambda))$ can possibly have degree less than $2n$, and then the convention is to set the missed eigenvalues to be ∞. Accounting for this possibility, we denote by (Λ, X), where $\Lambda \in \mathbb{C}^{2p\times 2p}$ and $X \in \mathbb{C}^{n\times 2p}$, $1 \leq p \leq n$, the representation of the finite eigenstructure of $P(\lambda)$.

The main results presented in [21] will be re-derived now. Our ultimate goal is to generalize them to the semidefinite case.

From (6.2.12), we have

$$-DX\Lambda = MX\Lambda^2 + KX$$

and pre-multiplying both sides by $(X\Lambda)^T$ gives

$$-(X\Lambda)^T DX\Lambda = (X\Lambda)^T MX\Lambda^2 + \Lambda^T X^T KX. \tag{6.2.13}$$

Transposing (6.2.13) and noting that $D = D^T$, we obtain

$$-(X\Lambda)^T DX\Lambda = (\Lambda^2)^T X^T MX\Lambda + X^T KX\Lambda. \tag{6.2.14}$$

Combining (6.2.13) and (6.2.14) gives, after rearrangement,

$$\left[(X\Lambda)^T MX\Lambda - X^T KX\right]\Lambda = \Lambda^T\left[(X\Lambda)^T MX\Lambda - X^T KX\right]. \tag{6.2.15}$$

Define

$$D_1 = (X\Lambda)^T MX\Lambda - X^T KX \tag{6.2.16}$$

so that (6.2.15) becomes

$$D_1\Lambda - \Lambda^T D_1 = 0. \tag{6.2.17}$$

Again, from (6.2.12), we have

$$-MX\Lambda^2 = DX\Lambda + KX$$

and multiplying both sides by $(X\Lambda^2)^T$ gives

$$-(X\Lambda^2)^T MX\Lambda^2 = \Lambda^T (X\Lambda)^T DX\Lambda + \Lambda^T (X\Lambda)^T KX. \tag{6.2.18}$$

Transposing (6.2.18) and noting that $M = M^T$, we have

$$-(X\Lambda^2)^T MX\Lambda^2 = \Lambda^T X^T D(X\Lambda)\Lambda + X^T K(X\Lambda)\Lambda. \tag{6.2.19}$$

Combining (6.2.18) and (6.2.19), and adding $(X\Lambda)^T KX\Lambda$ to both sides gives

$$\begin{gathered}\Lambda^T (X\Lambda)^T DX\Lambda + \Lambda^T (X\Lambda)^T KX + \Lambda^T X^T KX\Lambda = \\ \Lambda^T X^T KX\Lambda + \Lambda^T X^T D(X\Lambda)\Lambda + X^T K(X\Lambda)\Lambda\ .\end{gathered}$$

Factoring out Λ^T on the left and Λ on the right, we get

$$\begin{aligned}\Lambda^T \left[(X\Lambda)^T DX\Lambda + (X\Lambda)^T KX + X^T KX\Lambda\right] = \\ = \left[(X\Lambda)^T DX\Lambda + X^T K(X\Lambda) + (X\Lambda)^T KX\right]\Lambda.\end{aligned} \tag{6.2.20}$$

Define now

$$D_2 = (X\Lambda)^T DX\Lambda + (X\Lambda)^T KX + X^T KX\Lambda \tag{6.2.21}$$

so that (6.2.20) becomes

$$D_2\Lambda - \Lambda^T D_2 = 0. \tag{6.2.22}$$

The next theorem gives the main result of this subsection. It shows that under certain assumptions upon the nonzero eigenvalues of $P(\lambda)$, matrices D_1 and D_2 defined by (6.2.16) and (6.2.21) will have a special structure.

Theorem 6.2.2 : *Orthogonality Relations for the Symmetric Quadratic Pencil*

Let $P(\lambda) = \lambda^2 M + \lambda D + K$ be a symmetric pencil and let (Λ, X) be a finite representation of the eigenstructure of this pencil. Let the matrices D_1 and D_2 be defined by (6.2.16) and (6.2.21) and

$$D_3 = (X\Lambda)^T MX + X^T MX\Lambda + X^T DX. \tag{6.2.23}$$

Then the following statements hold:

(i) In case (Λ, X) is a complex representation, then D_1, D_2 and D_3 are block diagonal matrices where the size of each block is equal to the algebraic multiplicity of the eigenvalue of $P(\lambda)$ that the block represents. In particular, if the pencil $P(\lambda)$ has distinct nonzero eigenvalues, then D_1, D_2 and D_3 are diagonal matrices.

(ii) In case (Λ, X) is a real representation, then D_1, D_2 and D_3 are block diagonal matrices where the size of each block is equal to the algebraic multiplicity of the eigenvalue of $P(\lambda)$ that the block represents; however, a pair of complex conjugate eigenvalues will correspond to a single real block of size 2 (Schur bump).

(iii) In either of the cases (i) and (ii), we have

$$D_1 = D_3\Lambda \tag{6.2.24}$$

and

$$D_2 = -D_1\Lambda. \tag{6.2.25}$$

Proof of (iii): To prove (6.2.24), we post-multiply both sides of (6.2.23) by Λ and use (6.2.12) to obtain

$$D_3\Lambda = (X\Lambda)^T MX\Lambda + X^T(-KX)\ .$$

Using (6.2.16), we immediately obtain (6.2.24). To prove (6.2.25), we rewrite (6.2.21) as

$$D_2 = (X\Lambda)^T(DX\Lambda + KX) + X^TKX\Lambda = (X\Lambda)^T(-MX\Lambda^2) + X^TKX\Lambda =$$
$$(-(X\Lambda)^T MX\Lambda + X^TKX)\Lambda\ .$$

Using (6.2.16), we immediately obtain (6.2.25).

Proof of (ii): In this case, Λ can be assumed to be a real block diagonal matrix where a block of size k corresponds to group of k repeated real eigenvalues, a block of size 2 corresponds to a distinct complex conjugate pair and so on. This means that Λ can be written as

$$\Lambda = \begin{bmatrix} \Lambda_{11} & & & \\ & \Lambda_{22} & & \\ & & \cdots & \\ & & & \Lambda_{pp} \end{bmatrix}, \tag{6.2.26}$$

while the analogous partition of the symmetric matrix D_1 gives

$$D_1 = \begin{bmatrix} D_{11} & D_{12} & \dots & D_{1p} \\ D_{12}^T & D_{22} & \dots & D_{2p} \\ \dots & \dots & \dots & \dots \\ D_{1p}^T & D_{2p}^T & \dots & D_{pp} \end{bmatrix} \tag{6.2.27}$$

where the diagonal blocks of $D_{ii}, i = 1, \dots, p$ are all symmetric.

From (6.2.17), we then have

$$D_{ii}\Lambda_{ii} - \Lambda_{ii}^T D_{ii} = 0, i = 1, 2, \dots, p \tag{6.2.28}$$

$$D_{ij}\Lambda_{jj} - \Lambda_{ii}^T D_{ij} = 0, j \neq i;\ \ i, j = 1, 2, \dots, p \tag{6.2.29}$$

and since the blocks Λ_{jj} and Λ_{ii} do not have common eigenvalues by the construction of Λ, the second equation above gives $D_{ij} = 0, j \neq i;\ i,j = 1,2,\ldots,p$ and therefore D_1 has the same block structure as that of the matrix Λ.

Similar conclusion can be made about D_2 and D_3, using (6.2.24) and (6.2.25).

Proof of (i): The proof of the first statement is similar to the proof of (ii); however a complex conjugate pair of eigenvalues will correspond to two complex blocks each of size 1. In the particular case that $P(\lambda)$ has distinct nonzero eigenvalues Λ is a diagonal matrix which commutes with the matrices D_1 and D_2 by (6.2.17) and (6.2.22). Since Λ has distinct nonzero entries, the diagonal structure of D_1 and D_2 clearly follows. The diagonal structure of D_3 immediately follows from (iii), which we already proved.

Corollary 6.2.2: Assume that the hypotheses of Theorem 6.2.2 hold. Partition the matrices X and Λ as

$$X = \begin{bmatrix} X_1 & X_2 \end{bmatrix},\ \Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \tag{6.2.30}$$

and suppose Λ_1 and Λ_2 do not have common eigenvalues. Then

$$(X_1\Lambda_1)^T M X_2 \Lambda_2 - X_1^T K X_2 = 0 \tag{6.2.31}$$

$$(X_1\Lambda_1)^T D X_2 \Lambda_2 + X_1^T K (X_2\Lambda_2) + (X_1\Lambda_1)^T K X_2 = 0 \tag{6.2.32}$$

$$(X_1\Lambda_1)^T M X_2 + X_1^T M X_2 \Lambda_2 + X_1^T D X_2 = 0. \tag{6.2.33}$$

Proof: Follows immediately from (6.2.16), (6.2.21) and (6.2.23).

Remarks:

(i) The last statement of part (i) in Theorem 6.2.2 was proved earlier by Datta, Elhay and Ram [21] under the stronger hypothesis that M and K are both symmetric positive definite.

(ii) The relation (6.2.31) was proved earlier in the PhD Dissertation of Sarkissian [54].

Example 6.2.2: Consider the pencil $P(\lambda) = \lambda^2 M + \lambda D + K$, where

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 3 & 1 \\ 0 & 1 & 4 \end{bmatrix}.$$

Note that the matrix M is singular and that

$$p(\lambda) = det(P(\lambda)) = (\lambda+1)^2\,(2\lambda^3 + 5\lambda^2 + 14\lambda + 15).$$

Therefore, there are only five finite eigenvalues.

Case 1: A finite complex representation of the eigenstructure is

$$\Lambda = \begin{bmatrix} -0.6922 & & & & \\ & -1.0000 & & & \\ & & -1.0000 & & \\ & & & -1.4039 - 1.7566i & \\ & & & & -1.4039 + 1.7566i \end{bmatrix}$$

$$X = \begin{bmatrix} -1.0000 & -1.0000 & -1.0000 & -0.1305 - 0.5393i & -0.1305 + 0.5393i \\ -0.3078 & -0.3333 & 1.0000 & -0.8945 + 0.4470i & -0.8945 - 0.4470i \\ 0.1991 & 0.3333 & -1.0000 & 0.1275 + 0.7923i & 0.1275 - 0.7923i \end{bmatrix}.$$

The matrix D_1 satisfying (6.2.16) is

$$D_1 = \begin{bmatrix} -0.1801 & -0. & 0. & 0. & 0. \\ -0. & 0.2222 & 0.6667 & 0. & 0. \\ 0. & 0.6667 & -6.0000 & 0. & 0. \\ 0. & 0. & -0. & 5.4566 + 5.9477i & 0.0000 - 0.0000i \\ 0. & 0. & -0. & 0.0000 + 0.0000i & 5.4566 - 5.9477i \end{bmatrix}.$$

Note that D_1 is a block diagonal matrix and not strictly a diagonal matrix. This is because the pencil $P(\lambda)$ has a double eigenvalue $\mu = -1$. Similar conclusions can be drawn about D_2 and D_3.

Case 2: A finite real representation of the eigenstructure is

$$\Lambda = \begin{bmatrix} -0.6922 & & & & \\ & -1. & & & \\ & & -1. & & \\ & & & -1.4039 & -1.7566 \\ & & & 1.7566 & -1.4039 \end{bmatrix}$$

$$X = \begin{bmatrix} -1. & -1. & -1. & -0.1305 & -0.5393 \\ -0.3078 & -0.3333 & 1. & -0.8945 & 0.4470 \\ 0.1991 & 0.3333 & -1. & 0.1275 & 0.7923 \end{bmatrix}$$

The matrix D_1 satisfying (6.2.16) now is

$$D_1 = \begin{bmatrix} -0.1801 & 0. & 0. & 0. & 0. \\ 0. & 0.2222 & 0.6667 & 0. & 0. \\ 0. & 0.6667 & -6.0000 & 0. & 0. \\ 0. & 0. & 0. & 2.7283 & 2.9739 \\ 0. & 0. & 0. & 2.9739 & -2.7283 \end{bmatrix}.$$

Note that D_1 has two diagonal blocks of size two; one corresponds to the real double eigenvalue $\mu = -1$ and the other corresponds to the pair of complex conjugate eigenvalues $\mu = -1.4039 \pm 1.7566i$. Similar conclusions can be drawn for D_2 and D_3.

6.3 Finite Element Model Updating via Direct Methods using Modal Data

In this section, we describe two methods for updating of a symmetric positive semidefinite undamped model (M, D, K) and three formulas for updating of a model

(M, D, K) to a model $(\tilde{M}, \tilde{D}, \tilde{K})$ where damping is considered.

The first method is the Berman-Nagy method [5], which is a modification of the the Baruch's method [3], proposed earlier and widely used. The second one is a new method derived from the results established in the previous section.

The Berman-Nagy method assumes that the measured data is completely known (that is, the mode shapes are measured to the size of the finite-element model). If the measured data is not complete, in order to apply the method, a model reduction technique has to be used or some mode shape expansion has to be done. Besides, it may produce *spill-over*, that is, the natural frequencies and the corresponding mode shapes that are not measured cannot be guaranteed to remain unchanged.

The method was originally proposed for the updating of the linear pencil $Q(\lambda) = K - \lambda M$; however, it can easily be adapted to handle the second-order undamped case, as shown in Subsection 6.3.1.

The proposed method only assumes that an incomplete measured data is available and it is guaranteed not to produce spill-over.

Assume that only m natural frequencies and corresponding mode shapes vectors are to be updated and let (Λ, X) be a finite compact representation of the eigenstructure of the model; therefore, matrices Λ and X satisfy (6.2.3) and the conventions established in Subsection 6.2.1 hold. Partition Λ and X as follows:

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, X = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \tag{6.3.34}$$

where $\Lambda_1 \in \mathbb{C}^{m \times m}, X_1 \in \mathbb{C}^{n \times m}, \Lambda_2 \in \mathbb{C}^{(n-m)\times(n-m)}, X_2 \in \mathbb{C}^{n \times (n-m)}$ and such that

- (Λ_1, X_1) corresponds to the set of frequencies and mode shapes that needs to be updated

- (Λ_2, X_2) corresponds to the set of frequencies and mode shapes that is to remain unchanged.

Let $\Sigma_1^2 \in \mathbb{R}^{m\times m}$ denote the matrix that contains the information about the measured frequencies and let

$$Y_1 = \begin{bmatrix} Y_{11} \\ Y_{12} \end{bmatrix} \tag{6.3.35}$$

be the matrix of corresponding mode shapes, where $Y_{11} \in \mathbb{R}^{m\times m}$ and $Y_{12} \in \mathbb{R}^{(n-m)\times m}$.

In the Berman-Nagy method, the matrix Y_1 is assumed to be completely known. In the proposed method, it is assumed that only Y_{11} is known; the method itself conveniently constructs Y_{12}.

An important consequence of the compact representation proposed in Subsection 6.2.1 is that whenever an eigenvalue-eigenvector pair (λ_i, x_i) is replaced by another pair (σ_i, y_i), its conjugate pair $(\overline{\lambda}_i, \overline{x}_i)$ is *implicitly* replaced by $(\overline{\sigma}_i, \overline{y}_i)$.

Also note that the set (Λ_2, X_2) is not required for the implementation of any of the methods presented here.

6.3.1 Model Updating of an Undamped Model Using Complete Modal Data

Suppose that a complete modal data set is available. Under the stronger assumption that the matrix M is symmetric positive definite, our algorithm normalizes Y_1 such that $Y_1^T M Y_1 = I_m$, the m-th order identity matrix; this is equivalent to say that the m mode shapes represented in the matrix Y_1 are *mass-normalized.*

The following update in the stiffness matrix K is due to Berman and Nagy [5]:

$$\tilde{K} = K - KY_1Y_1^TM - MY_1Y_1^TK + MY_1Y_1^TKY_1Y_1^TM - MY_1\Sigma_1^2Y_1^TM. \tag{6.3.36}$$

This stiffness-matrix update can be shown to minimize the functional

$$N(S) = \|M^{-1/2}(S-K)M^{-1/2}\|_F^2.$$

The method is recommended when the matrix M is known with high accuracy. Clearly this is a rank-m symmetric update.

It is easy to check that the updated matrix $\tilde{K}$ is such that $MY_1\Sigma_1^2 + \tilde{K}Y_1 = 0$:

$$MY_1\Sigma_1^2 + \tilde{K}Y_1 = MY_1\Sigma_1^2 + (K - KY_1Y_1^TM - MY_1Y_1^TK + MY_1Y_1^TKY_1Y_1^TM - MY_1\Sigma_1^2Y_1^TM)Y_1 = MY_1\Sigma_1^2 + KY_1 - KY_1 - MY_1Y_1^TKY_1 + MY_1Y_1^TKY_1 - MY_1\Sigma_1^2 = 0$$

This means that the pair (Λ_1, X_1) is replaced by (Σ_1, Y_1) in the updated model.

Unfortunately, the modes and frequencies that were not measured, here represented by the pair (Λ_2, X_2), can change in the new model, as shown below:

$$\begin{gathered} MX_2\Lambda_2^2 + \tilde{K}X_2 = MX_2\Lambda_2^2 + KX_2 - KY_1Y_1^TMX_2 - MY_1Y_1^TKX_2 + \\ MY_1Y_1^TKY_1Y_1^TMX_2 - MY_1\Sigma_1^2Y_1^TMX_2 = \\ 0 + MY_1Y_1^TK\left[Y_1Y_1^TM - I\right]X_2 - \left[MY_1\Sigma_1^2 + KY_1\right]Y_1^TMX_2, \end{gathered}$$

which is not zero in general.

Algorithm 6.3.1 : *Model Updating of an Undamped Model Using Complete Measured Data*

Input: The symmetric matrices $M, K \in \mathbb{R}^{n\times n}$; the set of m analytical frequencies and mode shapes to be updated; the complete set of m measured frequencies and mode shapes from the vibration test.

Output: Updated stiffness matrix $\tilde{K}$.

Assumptions: M is symmetric positive definite and the given mode shapes form a linearly independent set of vectors.

Step 1: Form the matrices $\Sigma_1^2 \in \mathbb{R}^{m\times m}$ and $Y_1 \in \mathbb{R}^{n\times m}$ from the available data. Form the corresponding matrices $\Lambda_1^2 \in \mathbb{R}^{m\times m}$ and $X_1 \in \mathbb{R}^{n\times m}$.

Step 2: Compute the matrix $L \in \mathbb{R}^{m\times m}$ from the Cholesky factorization: $LL^T = Y_1^T M Y_1$. Update the matrix Y_1 : $Y_1 \leftarrow Y_1 L^{-T}$.

Step 3: Update the matrix K to $\tilde{K}$:

$$\tilde{K} = K - KY_1Y_1^TM - MY_1Y_1^TK + MY_1Y_1^TKY_1Y_1^TM - MY_1\Sigma_1^2Y_1^TM.$$

□

Example 6.3.1 : *Model Updating Using Complete Measured Data*

Consider the model (M, D, K) where $D = 0$ and M and K are given as

$$M = \begin{bmatrix} 1.2940 & 0. & 0. & 0. & 0. \\ 0. & 1.2940 & 0. & 0. & 0. \\ 0. & 0. & 1.2940 & 0. & 0. \\ 0. & 0. & 0. & 1.2940 & 0. \\ 0. & 0. & 0. & 0. & 1.2940 \end{bmatrix}$$

$$K = \begin{bmatrix} 1188.5000 & 196.6000 & 0. & 0. & -642.4000 \\ 196.6000 & 626.3000 & 0. & -555.6000 & 0. \\ 0. & 0. & 1188.5000 & -196.6000 & -546.1000 \\ 0. & -555.6000 & -196.6000 & 626.3000 & 196.6000 \\ -642.4000 & 0. & -546.1000 & 196.6000 & 4019.1000 \end{bmatrix}.$$

The matrices that represent mode shapes and natural frequencies are:

$$X = \begin{bmatrix} -0.1775 & -0.1159 & 0.6424 & 0.5472 & 0.1253 \\ -0.0182 & -0.5193 & 0.2441 & -0.2625 & -0.6118 \\ -0.1536 & -0.4141 & -0.5454 & 0.5224 & -0.0856 \\ 0.0567 & 0.5444 & -0.0335 & 0.3131 & -0.6116 \\ 0.8450 & -0.1474 & 0.0434 & 0.1832 & 0.0386 \end{bmatrix}$$

$$\Lambda^2 = 10^3 \begin{bmatrix} -3.2971 & & & & \\ & -0.9680 & & & \\ & & -0.9427 & & \\ & & & -0.6794 & \\ & & & & -0.0236 \end{bmatrix}.$$

Step 1. The matrices of measured frequencies and mode shapes are:

$$\Sigma_1^2 = \begin{bmatrix} -3270.11 & \\ & -23.05 \end{bmatrix}, \quad Y_1 = \begin{bmatrix} 1.2000 & 0.3000 \\ -1.1300 & 0.3500 \\ 0.3400 & 0.6700 \\ -0.5400 & 0.7800 \\ 0.9600 & 0.2300 \end{bmatrix}.$$

The corresponding modal matrices are:

$$\Lambda_1^2 = \begin{bmatrix} -3297.1298 & \\ & -23.6477 \end{bmatrix}, \quad X_1 = \begin{bmatrix} -0.1775 & 0.1253 \\ -0.0182 & -0.6118 \\ -0.1536 & -0.0856 \\ 0.0567 & -0.6116 \\ 0.8450 & 0.0386 \end{bmatrix}.$$

Also, note that

$$\Lambda_2^2 = \begin{bmatrix} -968.0307 & & \\ & -942.6949 & \\ & & -679.3933 \end{bmatrix},$$

$$X_2 = \begin{bmatrix} -0.1159 & 0.6424 & 0.5472 \\ -0.5193 & 0.2441 & -0.2625 \\ -0.4141 & -0.5455 & 0.5224 \\ 0.5444 & -0.0335 & 0.3131 \\ -0.1474 & 0.0434 & 0.1832 \end{bmatrix}.$$

Step 2. The matrix L and the updated matrix Y_1 are

$$L = \begin{bmatrix} 2.2880 & -0.0046 \\ & 1.3083 \end{bmatrix}, \quad Y_1 = \begin{bmatrix} 0.5245 & 0.2311 \\ -0.4939 & 0.2658 \\ 0.1486 & 0.5126 \\ -0.2360 & 0.5954 \\ 0.4196 & 0.1773 \end{bmatrix}.$$

Step 3. The updated stiffness matrix $\tilde{K}$ is

$$\tilde{K} = \begin{bmatrix} 2696.7197 & -1265.0993 & 469.5122 & -816.4529 & -196.1644 \\ -1265.0993 & 2099.6588 & -562.6168 & 230.0036 & -599.3697 \\ 469.5122 & -562.6168 & 1082.8045 & -683.4154 & -518.3873 \\ -816.4529 & 230.0036 & -683.4154 & 929.1160 & -324.2611 \\ -196.1644 & -599.3697 & -518.3873 & -324.2611 & 3772.4184 \end{bmatrix}.$$

Verification:

$$\|MY_1\Sigma_1^2 + \tilde{K}Y_1\|_F = 6.8436 \times 10^{-13}, \quad \|MX_2\Lambda_2^2 + \tilde{K}X_2\|_F = 2.1822 \times 10^3.$$

Therefore, we conclude:

- The complete measured data was entirely and exactly incorporated to the new model.
- The unmeasured frequencies and mode shapes *changed*, and therefore the update produced spill-over.

6.3.2 Model Updating of an Undamped Model Using Incomplete Modal Data

Suppose now that an incomplete modal data set is available, meaning that only m components of the mode shapes are known from measurement. This information is contained in matrices $\Sigma_1^2 \in \mathbb{R}^{m\times m}$ and $Y_{11} \in \mathbb{R}^{m\times m}$, as defined before.

Theorem 6.3.2 Consider the positive semidefinite model (M, D, K) with no damping, that is, $D = 0$. Let matrices $\Lambda \in \mathbb{R}^{n\times n}$ and $X \in \mathbb{R}^{n\times n}$, which represent the modal structure of the model, satisfy (6.2.3) and be partitioned as in (6.3.34). Suppose that the diagonal sub-matrices Λ_1 and Λ_2 do not have a common nonzero entry.

Then, for every symmetric matrix $\Phi \in \mathbb{R}^{m\times m}$, the updated symmetric matrix $\tilde{K}$ defined by

$$\tilde{K} = K - MX_1\Phi X_1^T M \tag{6.3.37}$$

satisfies

$$MX_2\Lambda_2^2 + \tilde{K}X_2 = 0. \tag{6.3.38}$$

Proof: Combining (6.3.38) and (6.3.37), we have

$$MX_2\Lambda_2^2 + \tilde{K}X_2 = MX_2\Lambda_2^2 + (K - MX_1\Phi X_1^T M)X_2 =$$
$$MX_2\Lambda_2^2 + KX_2 - MX_1\Phi X_1^T MX_2 = 0$$

since $X_1^T MX_2 = 0$, by Corollary 6.2.1.

□

In other words, Theorem 6.3.2 states that the symmetric updating of K by (6.3.37) is guaranteed to produce no spill-over.

We now show how the symmetric matrix Φ can be chosen so that the measured eigenvalues and eigenvectors are contained in the updated model, that is, which such choice of Φ, the matrix $\tilde{K}$ is such that

$$MY_1\Sigma_1^2 + \tilde{K}Y_1 = 0. \tag{6.3.39}$$

Substituting the expressions of Y_1 from (6.3.35) and $\tilde{K}$ from (6.3.37) in (6.3.39), we have

$$M\begin{bmatrix} Y_{11} \\ Y_{12} \end{bmatrix}\Sigma_1^2 + K\begin{bmatrix} Y_{11} \\ Y_{12} \end{bmatrix} = MX_1\Phi X_1^T M\begin{bmatrix} Y_{11} \\ Y_{12} \end{bmatrix}. \tag{6.3.40}$$

Assume that MX_1 has full rank. Then the QR factorization of this product defines orthogonal matrices $U_1 \in \mathbb{R}^{n\times m}$ and $U_2 \in \mathbb{R}^{n\times(n-m)}$, and an upper triangular matrix $Z \in \mathbb{R}^{m\times m}$ satisfying

$$MX_1 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} Z \\ 0 \end{bmatrix}. \tag{6.3.41}$$

Let $M = \begin{bmatrix} M_1 & M_2 \end{bmatrix}$ and $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$, where $M_1, K_1 \in \mathbb{R}^{n\times m}$ and $M_2, K_2 \in \mathbb{R}^{n\times(n-m)}$.

After pre-multiplication by $\begin{bmatrix} U_1 & U_2 \end{bmatrix}^T$ and using (6.3.41) with the above partitioning of M and K, equation (6.3.40) can be rewritten as

$$\begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} (M_1Y_{11} + M_2Y_{12})\,\Sigma_1^2 + \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} (K_1Y_{11} + K_2Y_{12}) = \begin{bmatrix} Z \\ 0 \end{bmatrix} \Phi X_1^T M Y_1. \tag{6.3.42}$$

Therefore, a solution $\Phi \in \mathbb{R}^{m\times m}$ to (6.3.40) exists only if

$$U_2^T\,(M_1Y_{11} + M_2Y_{12})\,\Sigma_1^2 = -U_2^T\,(K_1Y_{11} + K_2Y_{12}),$$

which is equivalent to

$$U_2^T M_2 Y_{12} \Sigma_1^2 + U_2^T K_2 Y_{12} = -U_2^T (K_1 Y_{11} + M_1 Y_{11} \Sigma_1^2). \tag{6.3.43}$$

Equation (6.3.43) is a descriptor Sylvester equation that can be solved, since Σ_1^2 is diagonal matrix, by Algorithm 4.3.4.

Once equation (6.3.43) is solved for Y_{12}, we can form the matrix Y_1 using (6.3.35) and then compute $\Phi \in \mathbb{R}^{m\times m}$ from

$$Y_1^T M Y_1 \Sigma_1^2 + Y_1^T K Y_1 = (Y_1^T M X_1)\Phi(Y_1^T M X_1)^T \tag{6.3.44}$$

which was obtained by pre-multiplying (6.3.39) by Y_1^T. In principle, equation (6.3.44) gives just a least-squares solution of (6.3.43). However, once (6.3.43) is satisfied, (6.3.44) gives an exact solution of (6.3.39).

However, the symmetry of the solution Φ is not guaranteed yet. Since the updated model should also be symmetric positive definite, the following orthogonality relations must hold by Theorem 6.2.1:

$$Y_1^T M Y_1 = D_1 \tag{6.3.45}$$

$$Y_1^T \tilde{K} Y_1 = D_2 \tag{6.3.46}$$

$$D_2 = -\Sigma_1^2 D_1, \tag{6.3.47}$$

where D_1 and D_2 are two diagonal matrices of order m. This means that the matrix Y_1 will have to be updated before the computation of Φ, and a matrix $G \in \mathbb{R}^{m \times m}$ must be found such that, after the update $Y_1 \leftarrow Y_1 G$, equation (6.3.45) is satisfied. We must remark that the invariant subspace spanned by this matrix would still be the same after the update; only the basis vectors would be modified.

We now show that the matrix Φ satisfying (6.3.44) is symmetric. First note that Σ_1^2 is a diagonal matrix. Then, using (6.3.46) and noting that D_1 is also a diagonal matrix, we see that the left hand side of (6.3.44) is symmetric. Thus, the right hand side

$$(Y_1^T M X_1)\Phi(Y_1^T M X_1)^T$$

of (6.3.44) is symmetric, implying that Φ is symmetric.

Algorithm 6.3.2 : *Model Updating of an Undamped Symmetric Positive Semidefinite Model using Incomplete Measured Data*

Input: The symmetric matrices $M, K \in \mathbb{R}^{n \times n}$; the set of m analytical frequencies and mode shapes to be updated; the complete set of m measured frequencies and mode shapes from the vibration test.

Output: Updated stiffness matrix $\tilde{K}$.

Assumption: $M = M^T \geq 0$ and $K = K^T \geq 0$.

Step 1: Form the matrices $\Sigma_1^2 \in \mathbb{R}^{m\times m}$ and $Y_{11} \in \mathbb{R}^{m\times m}$ from the available data. Form the corresponding matrices $\Lambda_1^2 \in \mathbb{R}^{m\times m}$ and $X_1 \in \mathbb{R}^{n\times m}$.

Step 2: Compute the matrices $U_1 \in \mathbb{R}^{n\times m}$, $U_2 \in \mathbb{R}^{n\times(n-m)}$ and $Z \in \mathbb{R}^{m\times m}$ from the QR factorization:

$$MX_1 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} Z \\ 0 \end{bmatrix}.$$

Step 3: Partition $M = \begin{bmatrix} M_1 & M_2 \end{bmatrix}, K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$ where $M_1, K_1 \in \mathbb{R}^{n\times m}$.

Step 4: Solve the following matrix equation to obtain $Y_{12} \in \mathbb{R}^{(n-m)\times m}$:

$$U_2^T M_2 Y_{12} \Sigma_1^2 + U_2^T K_2 Y_{12} = -U_2^T [K_1 Y_{11} + M_1 Y_{11} \Sigma_1^2]$$

and form the matrix

$$Y_1 = \begin{bmatrix} Y_{11} \\ Y_{12} \end{bmatrix}.$$

Step 5: Compute the matrix $L \in \mathbb{R}^{m\times m}$ and the diagonal matrix $J \in \mathbb{R}^{m\times m}$ such that $LJL^T = Y_1^T M Y_1$ is a symmetric (LDL^T) factorization of $Y_1^T M Y_1$. Update the matrix Y_1 by $Y_1 \leftarrow Y_1 (L^{-1})^T$.

Step 6: Compute $\Phi \in \mathbb{R}^{m\times m}$ by solving the following system of equations:

$$(Y_1^T M X_1)\Phi(Y_1^T M X_1)^T = Y_1^T M Y_1 (\Sigma_1)^2 + Y_1^T K Y_1$$

Step 7: Update

$$\tilde{K} = K - MX_1 \Phi X_1^T M.$$

Remark: The algorithm above can also be used when a complete measured data is available. In this case, Steps 2 , 3 and 4 must be skipped. However, the symmetric matrix Φ computed by Step 6 will be such that (6.3.39) is satisfied in the least-squares sense only. We recommend doing this when the measurements of Σ_1 and Y_1 are not highly accurate.

Example 6.3.2 : *Model Updating using Incomplete Measured Data*

Consider the positive semidefinite model (M, D, K) where $D = 0$ and matrices M and K are given by

$$M = \begin{bmatrix} 1.2940 & 0. & 0. & 0. & 0. \\ 0. & 1.2940 & 0. & 0. & 0. \\ 0. & 0. & 1.2940 & 0. & 0. \\ 0. & 0. & 0. & 1.2940 & 0. \\ 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

$$K = \begin{bmatrix} 1188.5000 & 196.6000 & 0. & 0. & -642.4000 \\ 196.6000 & 626.3000 & 0. & -555.6000 & 0. \\ 0. & 0. & 1188.5000 & -196.6000 & -546.1000 \\ 0. & -555.6000 & -196.6000 & 626.3000 & 196.6000 \\ -642.4000 & 0. & -546.1000 & 196.6000 & 4019.1000 \end{bmatrix}$$

as considered previously in Example 6.2.1. We recall that this model has two infinite eigenvalues.

Step 1. The matrices of measured frequencies and mode shapes are:

$$\Sigma_1^2 = \begin{bmatrix} -23.5500 & \\ & -990.0800 \end{bmatrix} , \quad Y_{11} = \begin{bmatrix} 0.3000 & 1.2000 \\ 0.3500 & -1.1300 \end{bmatrix}.$$

The corresponding modal matrices are :

$$\Lambda_1^2 = \begin{bmatrix} -23.6929 & \\ & -991.1000 \end{bmatrix}, \quad X_1 = \begin{bmatrix} 0.2045 & -0.5069 \\ -1. & -0.9986 \\ -0.1403 & -0.7470 \\ -0.9997 & 1 \\ 0.0625 & -0.2314 \end{bmatrix}.$$

Also, note that

$$\Lambda_2^2 = \begin{bmatrix} -702.3357 & \\ & -943.6921 \end{bmatrix}, \quad X_2 = \begin{bmatrix} -1. & 1. \\ 0.5775 & 0.2627 \\ -0.9602 & -0.9325 \\ -0.6475 & 0.0726 \\ -0.2586 & 0.0296 \end{bmatrix}$$

Step 2. From the QR factorization of MX_1:

$$U_1 = \begin{bmatrix} -0.1424 & -0.3023 \\ 0.6966 & -0.5955 \\ 0.0977 & -0.4455 \\ 0.6964 & 0.5963 \\ 0. & 0. \end{bmatrix}, \quad U_2 = \begin{bmatrix} -0.1668 & 0.9276 & 0. \\ -0.3696 & -0.1536 & 0. \\ 0.8895 & 0.0298 & 0. \\ 0.2108 & 0.3392 & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

and

$$Z = \begin{bmatrix} -1.8576 & 0 \\ 0 & 2.1700 \end{bmatrix}$$

Step 3. The partition of matrices M and K is straightforward.

Step 4. The solution of the descriptor Sylvester equation is

$$Y_{12} = \begin{bmatrix} 0.3662 & 0.6685 \\ -0.6942 & -3.8523 \\ 0.1317 & 0.4711 \end{bmatrix}.$$

Step 5. Computing the LDL^T factorization and updating Y_1:

$$L = \begin{bmatrix} 1 & 0 \\ 3.4804 & 1 \end{bmatrix} , \quad J = \begin{bmatrix} 1.0722 & 0 \\ 0 & 10.3107 \end{bmatrix} , \quad Y_1 = \begin{bmatrix} 0.3000 & 0.1559 \\ 0.3500 & -2.3481 \\ 0.3662 & -0.6061 \\ -0.6942 & -1.4362 \\ 0.1317 & 0.0128 \end{bmatrix}$$

Step 6. The symmetric matrix Φ is

$$\Phi = \begin{bmatrix} -335.7777 & -80.9068 \\ -80.9068 & 246.3905 \end{bmatrix}$$

Step 7. The updated stiffness matrix is:

$$\tilde{K} = \begin{bmatrix} 1077.9001 & -86.2129 & -183.4235 & 190.5648 & -642.4000 \\ -86.2129 & 1047.6955 & -108.7290 & 418.2325 & 0. \\ -183.4235 & -108.7290 & 997.7215 & 272.5997 & -546.1000 \\ 190.5648 & 418.2325 & 272.5997 & 504.7781 & 196.6000 \\ -642.4000 & 0. & -546.1000 & 196.6000 & 4019.1000 \end{bmatrix} .$$

Verification:

$$\|MY_1\Sigma_1^2 + \tilde{K}Y_1\|_F = 3.1607 \times 10^{-13} , \quad \|MX_2\Lambda_2^2 + \tilde{K}X_2\|_F = 2.4177 \times 10^{-12} .$$

Therefore, we conclude that

- The incomplete measured data was entirely and accurately incorporated in the new model.
- The unmeasured frequencies and mode shapes *did not* change, and therefore the update did not produce spill-over.

6.3.3 Symmetric Updates of a Damped Positive Semidefinite Second-order Model

In the previous subsections, we considered the updating of an undamped finite-element generated model to an undamped model. In this subsection, we study the update of a damped finite-element generated model.

Consider a symmetric model (M, D, K) and let (Λ, X) represent its eigenstructure. In the case that all eigenvalues are finite, we have $\Lambda \in \mathbb{C}^{2n\times 2n}$ and $X \in \mathbb{R}^{n\times 2n}$ satisfying (6.2.12). Suppose that Λ and X are partitioned as in (6.3.34). Here (Λ_1, X_1) are desired to be changed to (Σ_1, Y_1), while (Λ_2, X_2) is usually not known, but is desired to not change.

As application of Corollary 6.2.2, we show how to compute symmetric updates $\tilde{M}, \tilde{D}$ and $\tilde{K}$ of the matrices M, D and K such that the unmeasured frequencies and mode shapes of the original model are preserved; that is, such that

$$\tilde{M}X_2\Lambda_2^2 + \tilde{D}X_2\Lambda_2 + \tilde{K}X_2 = 0. \tag{6.3.48}$$

We give explicit formulas for $\tilde{M}$, $\tilde{D}$ and $\tilde{K}$, using again a parametric symmetric matrix Φ. Finally, we show how to choose Φ so that the measured frequencies and mode shapes are also contained in the updated model; that is, $\tilde{M}, \tilde{D}$ and $\tilde{K}$ are such that

$$\tilde{M}Y_1\Sigma_1^2 + \tilde{D}Y_1\Sigma_1 + \tilde{K}Y_1 = 0. \tag{6.3.49}$$

Theorem 6.3.3 : *No-spill-over symmetric updates of a Damped Positive Semidefinite Second-order Model*

Consider a positive semidefinite model (M, D, K) and let $\Lambda \in \mathbb{R}^{n\times n}$ and $X \in \mathbb{R}^{n\times n}$ satisfy (6.2.12) and be partitioned as in (6.3.34). Suppose that Λ_1 and Λ_2 do not have a common eigenvalue.

Let $\Phi \in \mathbb{C}^{m\times m}$ be any symmetric matrix. Define

$$\begin{aligned}\tilde{M} &= M - MX_1\Lambda_1\Phi(X_1\Lambda_1)^T M\\ \tilde{D} &= D + MX_1\Lambda_1\Phi X_1^T K + KX_1\Phi(X_1\Lambda_1)^T M\\ \tilde{K} &= K - KX_1\Phi X_1^T K.\end{aligned} \tag{6.3.50}$$

Then the updated model $(\tilde{M}, \tilde{D}, \tilde{K})$ is symmetric and the unmeasured frequencies and mode shapes of the original model remain unchanged.

Proof: Using (6.3.49) and (6.3.50), we have

$$\begin{gathered}\tilde{M}X_2\Lambda_2^2 + \tilde{D}X_2\Lambda_2 + \tilde{K}X_2 = (MX_2\Lambda_2^2 + DX_2\Lambda_2 + KX_2) - MX_1\Lambda_1\Phi(X_1\Lambda_1)^T MX_2\Lambda_2^2 +\\ MX_1\Lambda_1\Phi X_1^T KX_2\Lambda_2 + KX_1\Phi(X_1\Lambda_1)^T MX_2\Lambda_2 - KX_1\Phi X_1^T KX_2 =\\ = MX_1\Lambda_1\Phi[X_1^T KX_2 - (X_1\Lambda_1)^T MX_2\Lambda_2]\Lambda_2 - KX_1\Phi[X_1^T KX_2 - (X_1\Lambda_1)^T MX_2\Lambda_2] = 0\end{gathered}$$

since the terms in brackets vanish by Corollary 6.2.2.

Example 6.3.3: Consider the model (M, D, K) where matrices M, D and K are given as

$$M = \begin{bmatrix} 1.2000 & 0. & 0. & 0.\\ 0. & 2.3400 & 0. & 0.\\ 0. & 0. & 1.6700 & 0.\\ 0. & 0. & 0. & 0. \end{bmatrix}, \quad D = \begin{bmatrix} 2. & -1. & 0. & 0.\\ -1. & 2. & -1. & 0.\\ 0. & -1. & 2. & -1.\\ 0. & 0. & -1. & 2. \end{bmatrix}$$

$$K = \begin{bmatrix} 1. & -0.5000 & 0. & 0.\\ -0.5000 & 3. & -1.5000 & 0.\\ 0. & -1.5000 & 5. & -2.5000\\ 0. & 0. & -2.5000 & 7. \end{bmatrix}.$$

The finite eigendecomposition of this model is:

$$\Lambda = \begin{bmatrix} -0.1646 & -0.8464 & 0. & 0. & 0. & 0. & 0. \\ 0.8464 & -0.1646 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & -0.9382 & -0.3230 & 0. & 0. & 0. \\ 0. & 0. & 0.3230 & -0.9382 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & -0.6201 & -1.5067 & 0. \\ 0. & 0. & 0. & 0. & 1.5067 & -0.6201 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & -3.4737 \end{bmatrix}$$

$$X = \begin{bmatrix} -0.4319 & 0.5068 & 0.9653 & -0.2612 & 0.2509 & -0.0733 & -0.0016 \\ -0.2122 & 0.9772 & -0.1794 & -0.0143 & 0.0426 & -0.4509 & 0.0046 \\ 0.0282 & 0.5563 & -0.0247 & 0.0158 & -0.6181 & 0.7861 & -0.0540 \\ 0.0300 & 0.1988 & -0.0072 & 0.0055 & -0.1687 & 0.3300 & 1. \end{bmatrix}.$$

Let the set of frequencies and mode shapes to be updated be:

$$\Lambda_1 = \begin{bmatrix} -0.1646 & -0.8464 \\ 0.8464 & -0.1646 \end{bmatrix}, \quad X_1 = \begin{bmatrix} -0.4319 & 0.5068 \\ -0.2122 & 0.9772 \\ 0.0282 & 0.5563 \\ 0.0300 & 0.1988 \end{bmatrix}.$$

We choose the symmetric matrix Φ as

$$\Phi = \begin{bmatrix} 1. & -0.5000 \\ -0.5000 & 1. \end{bmatrix}.$$

Then the matrices $\tilde{M}, \tilde{D}, \tilde{K}$ given by (6.3.50) are

$$\tilde{M} = \begin{bmatrix} 0.9285 & -0.8706 & -0.3280 & 0. \\ -0.8706 & -1.6425 & -1.7395 & 0. \\ -0.3280 & -1.7395 & 0.8765 & 0. \\ 0. & 0. & 0. & 0. \end{bmatrix}$$

$$\tilde{D} = \begin{bmatrix} 1.7208 & -1.7958 & -0.0985 & 0.0602 \\ -1.7958 & -3.4050 & -2.5017 & 0.2777 \\ -0.0985 & -2.5017 & 1.7191 & -0.8783 \\ 0.0602 & 0.2777 & -0.8783 & 2. \end{bmatrix}$$

$$\tilde{K} = \begin{bmatrix} 0.8877 & -0.9888 & -0.0196 & 0.0466 \\ -0.9888 & -1.4676 & -2.6672 & 0.1916 \\ -0.0196 & -2.6672 & 4.4965 & -2.4970 \\ 0.0466 & 0.1916 & -2.4970 & 6.9806 \end{bmatrix}.$$

Unfortunately, these matrices are no longer positive semidefinite.

After the update, the pair (Λ_1, X_1) is changed to (Σ_1, Y_1), where

$$\Sigma_1 = \begin{bmatrix} -0.8624 & -0.4792 \\ 0.4792 & -0.8624 \end{bmatrix}, \quad Y_1 = \begin{bmatrix} -0.4101 & 0.4947 \\ -0.8972 & 0.4417 \\ -0.5363 & 0.1285 \\ -0.1951 & 0.0295 \end{bmatrix}.$$

Note that Σ_1 has eigenvalues $-0.8624 \pm -0.4792i$.

Verification:

$$\|\tilde{M}X_2\Lambda_2^2 + \tilde{D}X_2\Lambda_2 + \tilde{K}X_2\|_F = 1.2151 \times 10^{-14}.$$

Therefore, we conclude that the update did not produce spill-over.

Corollary 6.3.3 : *Characterization of Possible Solutions to the Model Updating Problem of a Damped Second-Order Model*

Under the assumptions of Theorem 6.3.3, a solution to the damped positive semidefinite model updating problem exists if and only if there exists a symmetric matrix $\Phi \in \mathbb{C}^{m\times m}$ satisfying the generalized Sylvester equation

$$P\Phi Q - R\Phi S = W \tag{6.3.51}$$

where the matrices P, Q, R, S and W are given by

$$\begin{aligned} &P = KX_1, && Q = X_1^T KY_1 - (X_1\Lambda_1)^T MY_1\Sigma_1, \quad R = MX_1\Lambda_1 \\ &S = [X_1^T KY_1 - (X_1\Lambda_1)^T MY_1\Sigma_1]\Sigma_1, && W = MY_1\Sigma_1^2 + DY_1\Sigma_1 + KY_1. \end{aligned} \tag{6.3.52}$$

Proof : Using (6.3.49) and (6.3.50), we have

$$MY_1\Sigma_1^2 - MX_1\Lambda_1\Phi(X_1\Lambda_1)^T MY_1\Sigma_1^2 + DY_1\Sigma_1 + MX_1\Lambda_1\Phi(X_1\Lambda_1)^T MY_1\Sigma_1 + \\ KX_1\Phi(X_1\Lambda_1)^T KY_1\Sigma_1 + KY_1 - KX_1\Phi X_1^T KY_1 = 0.$$

Rearranging gives

$$MY_1\Sigma_1^2 + DY_1\Sigma_1 + KY_1 = \\ KX_1\Phi[X_1^T KY_1 - (X_1\Lambda_1)^T MY_1\Sigma_1] - MX_1\Lambda_1\Phi[X_1^T KY_1 - (X_1\Lambda_1)^T MY_1\Sigma_1]\Sigma_1$$

and the claim follows.

□

Thus, Corollary 6.3.3 shows that, under the symmetric no-spill-over update formula (6.3.50), solutions to the model updating problem can be found by solving the generalized Sylvester-equation (6.3.51). Note that (6.3.51) has more equations than unknowns. Thus, specialized methods have to be used in its solution, which is left for future research. Besides, if no symmetric solution of (6.3.51) exists, then symmetric updates that best satisfy (6.3.49) in the least-squares sense, for instance, can be found by using weighted least-squares techniques in the solution of (6.3.51).

6.4 Finite Element Model Updating via Eigenvalue Embedding

In this section, we consider the Eigenvalue Embedding problem. Recall that this problem is a particular case of the finite-element model updating problem considered in the last section, in the sense that here we concern only with the updating of natural frequencies.

In [11], we proposed a solution of this problem using symmetric low-rank updates. The explicit formulas for updates were given under the assumptions that the matrices M and K are symmetric and positive definite.

In this section, we show how to extend these formulas to the symmetric positive semidefinite case and when only the embedding of real eigenvalues is concerned. The orthogonality relations given in Corollary 6.2.2 play a key role in our development.

In Subsection 6.4.2 we present an algorithm, adapted from [11], for simultaneous embedding of a group of r real numbers $\mu_1, \ldots, \mu_r$.

6.4.1 Single Real Eigenvalue Embedding

Theorem 6.4.1: *Embedding of a real number μ_1*

Let (λ_1, x_1), $\lambda_1 \neq 0$ be a single (isolated) real eigenpair of the symmetric positive semidefinite model (M, D, K). Suppose that we want to replace λ_1 by a real number μ_1 in the new model. Assume that μ_1 is such that $x_1^T K x_1 - \lambda_1 \mu_1 x_1^T M x_1 \neq 0$.

Then the updated model $(\tilde{M}, \tilde{D}, \tilde{K})$ defined by

$$\begin{aligned} \tilde{M} &= M - \epsilon_1 \lambda_1 M x_1 x_1^T M \\ \tilde{D} &= D + \epsilon_1 (M x_1 x_1^T K + K x_1 x_1^T M) \\ \tilde{K} &= K - \tfrac{\epsilon_1}{\lambda_1} K x_1 x_1^T K \end{aligned} \tag{6.4.53}$$

where

$$\epsilon_1 = \frac{\lambda_1 - \mu_1}{x_1^T K x_1 - \lambda_1 \mu_1 x_1^T M x_1} \tag{6.4.54}$$

is clearly symmetric and has the following properties:

(i) the number μ_1 is an eigenvalue of $(\tilde{M}, \tilde{D}, \tilde{K})$ with corresponding eigenvector x_1

(ii) $(\lambda_k, x_k), k = 2, \ldots, 2n$ are also eigenpairs of $(\tilde{M}, \tilde{D}, \tilde{K})$.

Proof: A detailed and elaborated proof, which uses non-equivalence transformations of the pencil $P(\lambda) = \lambda^2 M + \lambda D + \lambda K$, was originally given in [11] under the assumption that M and K are positive definite matrices. We present here a simplified proof that uses the orthogonality relations given in Corollary 6.2.2.

Let i be any integer such that $1 \leq i \leq 2n$. Then, since (λ_i, x_i) is an eigenpair of $P(\lambda)$, we have

$$P(\lambda_i)x_i = \lambda_i^2 M x_i + \lambda_i D x_i + K x_i = 0 \tag{6.4.55}$$

which gives

$$P(\lambda)x_i = \lambda^2 M x_i + \lambda D x_i - \lambda_i^2 M x_i - \lambda_i D x_i = (M x_i(\lambda + \lambda_i) + D x_i)(\lambda - \lambda_i). \tag{6.4.56}$$

Defining now

$$\tilde{P}(\lambda) = \lambda^2 \tilde{M} + \lambda \tilde{D} + \tilde{K} \tag{6.4.57}$$

and setting $\nu_1 = \epsilon_1 / \lambda_1$ gives, after using (6.4.53),

$$\begin{gathered}
\tilde{P}(\lambda)x_i = (\lambda^2 M + \lambda D + K)x_i - \lambda^2 \lambda_1^2 \nu_1 M x_1 x_1^T M x_i + \lambda \lambda_1 \nu_1 (M x_1 x_1^T K + \\
K x_1 x_1^T M) x_i - \nu_1 K x_1 x_1^T K x_i = \\
= P(\lambda)x_i - \lambda(\lambda \lambda_1^2 \nu_1 M x_1 - \lambda_1 \nu_1 K x_1) x_1^T M x_i + (\lambda \lambda_1 \nu_1 M x_1 - \nu_1 K x_1) x_1^T K x_i = \\
= P(\lambda)x_i - \lambda \lambda_1 \nu_1 (\lambda \lambda_1 M x_1 - K x_1) x_1^T M x_i + \nu_1 (\lambda \lambda_1 M x_1 - K x_1) x_1^T K x_i = \\
= P(\lambda)x_i + (\lambda \lambda_1 M x_1 - K x_1) \nu_1 [x_1^T K x_i - \lambda \lambda_1 x_1^T M x_i].
\end{gathered}$$

Using (6.4.55), we then have

$$\begin{gathered}
\tilde{P}(\lambda)x_i = P(\lambda)x_i + (\lambda \lambda_1 M x_1 + \lambda_1^2 M x_1 + \lambda_1 D x_1) \nu_1 [x_1^T K x_i - \lambda \lambda_1 x_1^T M x_i] = \\
P(\lambda)x_i + ((\lambda + \lambda_1) M x_1 + D x_1) \lambda_1 \nu_1 [x_1^T K x_i - \lambda \lambda_1 x_1^T M x_i].
\end{gathered}$$

Therefore,

$$\tilde{P}(\lambda)x_i = P(\lambda)x_i + ((\lambda + \lambda_1)Mx_1 + Dx_1)\epsilon_1[x_1^T K x_i - \lambda\lambda_1 x_1^T M x_i]. \qquad (6.4.58)$$

Proof of (i): if we set $i = 1$, then (6.4.56) and (6.4.58) give

$$\tilde{P}(\lambda)x_1 =$$
$$((\lambda + \lambda_1)Mx_1 + Dx_1)(\lambda - \lambda_1) + ((\lambda + \lambda_1)Mx_1 + Dx_1)\epsilon_1(x_1^T K x_1 - \lambda\lambda_1 x_1^T M x_1) =$$
$$= (\lambda + \lambda_1)Mx_1 + Dx_1)[(\lambda - \lambda_1) + \epsilon_1(x_1^T K x_1 - \lambda\lambda_1 x_1^T M x_1)]$$

and setting $\lambda = \mu_1$ we obtain

$$\tilde{P}(\mu_1)x_1 = ((\mu_1 + \lambda_1)Mx_1 + Dx_1)[(\mu_1 - \lambda_1) + \epsilon_1(x_1^T K x_1 - \mu_1\lambda_1 x_1^T M x_1)] = 0$$

since the term in brackets vanishes by the definition of ϵ_1 given in (6.4.54).

Proof of (ii): if $2 \leq i \leq 2n$ and $\lambda = \lambda_i$, then the term in brackets of (6.4.58) clearly vanishes by the orthogonality relation given in Corollary 6.2.2, and since $P(\lambda_i)x_i = 0$ by hypothesis, part (ii) follows.

$\square$

Remarks:

- Unfortunately, the positive semidefiniteness of the model (M, D, K) may not be preserved by the eigenvalue embedding formulae (6.4.53).
- If K is symmetric positive definite, then we can assume that the eigenvector x_1 was normalized such that $x_1^T K x_1 = 1$. Defining $\theta_1 = x_1^T M x_1$, we see that the formulae given in [11] are a particular case of (6.4.53).

Example 6.4.1: Consider the matrices M, D and K as

$$M = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 0. \end{bmatrix}, \quad D = \begin{bmatrix} 2. & -1. & 0. \\ -1. & 2. & -1. \\ 0. & -1. & 2. \end{bmatrix}, \quad K = \begin{bmatrix} 1. & -1. & 0. \\ -1. & 3. & 1. \\ 0. & 1. & 4. \end{bmatrix}.$$

The matrices Λ and X were computed in Example 6.2.2 following Case 2.

We have $\lambda_1 = -0.6922$ and

$$x_1 = \begin{bmatrix} -1. & -0.3078 & 0.1991 \end{bmatrix}^T.$$

We want to replace λ_1 by $\mu_1 = -0.5$.

Evaluating (6.4.54), we have $\epsilon_1 = -0.5899$ and from (6.4.53) :

$$\tilde{M} = \begin{bmatrix} 0.5917 & -0.1257 & 0. \\ -0.1257 & 0.9613 & 0. \\ 0. & 0. & 0. \end{bmatrix}$$

$$\tilde{D} = \begin{bmatrix} 1.1833 & -0.9631 & 0.2883 \\ -0.9631 & 2.1001 & -0.9112 \\ 0.2883 & -0.9112 & 2. \end{bmatrix}$$

$$\tilde{K} = \begin{bmatrix} 0.5917 & -0.8374 & 0.2883 \\ -0.8374 & 2.9352 & 0.8852 \\ 0.2883 & 0.8852 & 3.7964 \end{bmatrix}.$$

Verification:

$$\|\mu_1^2 \tilde{M} x_1 + \mu_1 \tilde{D} x_1 + \tilde{K} x_1\| = 3.0657 \times 10^{-16}$$

and

$$\|\tilde{M} X_2 \Lambda_2^2 + \tilde{D} X_2 \Lambda_2 + \tilde{K} X_2\|_F = 5.0114 \times 10^{-15}.$$

Therefore, we conclude that

- The number μ_1 was successfully embedded in the new model and the eigenvector x_1 was preserved.

- The remaining eigenvalues and eigenvectors of the model did not change by the embedding.

6.4.2 Simultaneous Embedding of Several Real Eigenvalues

In this subsection we consider the problem of simultaneously embedding of a group of r real numbers. Specifically, we consider the following problem:

Let $\{\lambda_1, \ldots, \lambda_r, \lambda_{r+1}, \ldots, \lambda_{2n}\}$ be the set of natural frequencies of the symmetric positive semidefinite model (M, D, K) of order n , where the natural frequencies $\{\lambda_1, \ldots, \lambda_r\}$ and their corresponding mode shapes $\{x_1, \ldots, x_r\}$ are real .

Given a set of real numbers $\{\mu_1, \ldots, \mu_r\}$, compute a new updated model $(\tilde{M}, \tilde{D}, \tilde{K})$ whose natural frequencies are given by $\{\mu_1, \ldots, \mu_r, \lambda_{r+1}, \ldots, \lambda_{2n}\}$.

The following method computes the matrices W, Z and the diagonal matrices E_m, E_d and E_k such that the symmetric updated model $(\tilde{M}, \tilde{D}, \tilde{K})$, where

$$\begin{aligned} \tilde{M} &= M - W E_m W^T \\ \tilde{D} &= D + Z E_d W^T + W E_d Z^T \\ \tilde{K} &= K - Z E_k Z^T \end{aligned} \tag{6.4.59}$$

solve the problem above.

To develop the method, we recall that by successive applications of (6.4.53) we obtain

$$\begin{aligned} M_s &= M_{s-1} - \epsilon_s \lambda_s M_{s-1} x_s x_s^T M_{s-1} \\ D_s &= D_{s-1} + \epsilon_s [M_{s-1} x_s x_s^T K_{s-1} + K_{s-1} x_s x_s^T M_{s-1}] \\ K_k &= K_{s-1} - \tfrac{\epsilon_s}{\lambda_s} K_{s-1} x_s x_s^T K_{s-1} \end{aligned} \tag{6.4.60}$$

where ϵ_s are given by

$$\epsilon_s = \frac{\lambda_s - \mu_s}{x_s^T K_{s-1} x_s - \lambda_s \mu_s \theta_s}. \tag{6.4.61}$$

where

$$\theta_s = x_s^T M_{s-1} x_s \tag{6.4.62}$$

and the vectors x_s are normalized such that $|x_s^T K_{s-1} x_s| = 1$, when possible.

In (6.4.60), $M_0 = M, D_0 = D, K_0 = K$ and $M_r = \tilde{M}, D_r = \tilde{D}, K_r = \tilde{K}$ are the final updated matrices.

These formulas can be used to successively embed the real eigenvalues one at a time, where the coefficient matrices need to be updated at the end of each embedding.

The method proposed below delays the updating of the coefficient matrices until all the real numbers $\{\theta_s, \epsilon_s\}$, needed for the multi embedding, have been computed. After all these quantities have been computed, the coefficient matrices are updated with only one rank-r symmetric update. *This strategy brings dense matrix-matrix computations to the problem, so that high-performance Level 3 BLAS computational procedures can be used in its solution.*

To develop the relation (6.4.59), we note that, using (6.4.60), we can write

$$\tilde{M} = M_0 - \sum_{k=1}^{r} \epsilon_k \lambda_k M_{k-1} x_k x_k^T M_{k-1} \tag{6.4.63}$$

which clearly can be written in the form of equation (6.4.59) with

$$W = \left[\begin{array}{cccc} M_0 x_1 & M_1 x_2 & \ldots & M_r x_r \end{array} \right] \tag{6.4.64}$$

and $E_m = diag(\ \epsilon_1 \lambda_1 \quad \ldots \quad \epsilon_r \lambda_r\ )$.

We also observe that, for $s = 1, \ldots, r$

$$\begin{aligned} \theta_s &= x_s^T [M_{s-2} - \epsilon_{s-1} \lambda_{s-1} M_{s-2} x_{s-1} x_{s-1}^T M_{s-2}] x_s = \\ &= x_s^T M_{s-2} x_s - \epsilon_{s-1} \lambda_{s-1} (x_s^T M_{s-2} x_{s-1})(x_{s-1}^T M_{s-2} x_s) \end{aligned} \tag{6.4.65}$$

and

$$\begin{aligned} M_{s-1}x_s &= [M_{s-2} - \epsilon_{s-1}\lambda_{s-1}M_{s-2}x_{s-1}x_{s-1}^T M_{s-2}]x_s = \\ &= M_{s-2}x_s - \epsilon_{s-1}\lambda_{s-1}(x_{s-1}M_{s-2}x_s)M_{s-2}x_{s-1}. \end{aligned} \tag{6.4.66}$$

The last equation allows the recursive computation of the columns of W. Equation (6.4.65) allows the recursive computation of the real parameters θ_s and ϵ_s.

Similarly, the expressions for $\tilde{D}$ and $\tilde{K}$ can be obtained with appropriate matrices Z, E_k and E_d.

Algorithm 6.4.2 *Simultaneous Real Embedding*

Input: Set of real numbers $\{\mu_i\}, i = 1, \ldots, r$; set of unwanted real eigenvalues $\{\lambda_i\}, i = 1, \ldots, r$ with corresponding eigenvectors $\{x_i\}, i = 1, \ldots, r$; symmetric matrices M, D and K such that $M, K \geq 0$.

Output: Matrices $\tilde{M}, \tilde{D}$ and $\tilde{K}$ such that the model $(\tilde{M}, \tilde{D}, \tilde{K})$ has eigenvalues

$\{\mu_1, \ldots, \mu_r, \lambda_{r+1}, \ldots, \lambda_{2n}\}$.

Step 1: Compute $m_i = Mx_i$, $k_i = Kx_i$, $i = 1, \ldots, r$.

Step 2: Compute $\alpha_{ij} = x_i^T m_j$, $\beta_{ij} = x_i^T k_j$, $j = i, \ldots, r$; $i = 1, \ldots, r$.

Step 3: Compute $\eta_1 = \sqrt{|x_1^T K x_1|}$ and update

$\alpha_{11} \leftarrow \alpha_{11}/\eta_1, \beta_{11} \leftarrow \beta_{11}/\eta_1$

$\alpha_{1j} \leftarrow \alpha_{1j}/\eta_1, \beta_{1j} \leftarrow \beta_{1j}/\eta_1, j = 1, \ldots, r$

Step 4: Set $\epsilon_1 = \dfrac{\lambda_1 - \mu_1}{\beta_{11} - \lambda_1\mu_1\alpha_{11}}$.

Step 5: For $s = 2, \ldots, r$ do Steps 6, 7, 8 and 9

Step 6: For $i = s, \ldots, r$ and $j = i, \ldots, r$

Update $\alpha_{ij} \leftarrow \alpha_{ij} - \epsilon_{s-1}\lambda_{s-1}\alpha_{s-1,i}\alpha_{s-1,j}$

Update $\beta_{ij} \leftarrow \beta_{ij} - \frac{\epsilon_{s-1}}{\lambda_{s-1}}\beta_{s-1,i}\beta_{s-1,j}$

Step 7: Compute $\eta_s = \sqrt{|\beta_{ss}|}$.

Step 8: If $\eta_s > 0$ then

Update $\alpha_{i,s} \leftarrow \alpha_{i,s}/\eta_s$ and $\beta_{i,s} \leftarrow \beta_{i,s}/\eta_s$, $i = 1, \ldots, s$.

Update $\alpha_{s,j} \leftarrow \alpha_{s,j}/\eta_s$ and $\beta_{s,j} \leftarrow \alpha_{s,j}/\eta_s \;\;,\;\; j = s, \ldots, r$.

Step 9: Compute $\epsilon_s = \dfrac{\lambda_s - \mu_s}{\beta_{ss} - \lambda_s \mu_s \alpha_{ss}}$

Step 10: Update $m_i \leftarrow m_i/\eta_i,\; k_i \leftarrow k_i/\eta_i, i = 1, \ldots, r.$

Step 11: For $s = 2, \ldots, r$ and $i = s, \ldots, r$

Update $m_i \leftarrow m_i - \epsilon_{s-1}\lambda_{s-1}\alpha_{s-1,i}m_{s-1}$

Update $k_i \leftarrow k_i - \frac{\epsilon_{s-1}}{\lambda_{s-1}}\beta_{s-1,i}k_{s-1}$

Step 12: Form the matrices $W = \begin{bmatrix} m_1 & m_2 & \ldots & m_r \end{bmatrix}$ and $Z = \begin{bmatrix} k_1 & k_2 & \ldots & k_r \end{bmatrix}$

.

Step 13: Form the diagonal matrices

$$E_m = diag\begin{pmatrix} \epsilon_1\lambda_1 & \epsilon_2\lambda_2 & \ldots & \epsilon_r\lambda_r \end{pmatrix}$$
$$E_d = diag\begin{pmatrix} \epsilon_1 & \epsilon_2 & \ldots & \epsilon_r \end{pmatrix}$$
$$E_k = diag\begin{pmatrix} \epsilon_1/\lambda_1 & \epsilon_2/\lambda_2 & \ldots & \epsilon_r/\lambda_r \end{pmatrix}$$

Sep 13: Compute $\tilde{M}$, $\tilde{D}$ and $\tilde{K}$ using (6.4.59).

Example 6.4.2:

Consider the model (M, D, K) where

$$M = \begin{bmatrix} 22.1672 & 0. & 0. \\ 0. & 24. & 0. \\ 0. & 0. & 2. \end{bmatrix} \;,\; D = \begin{bmatrix} 0. & 12. & 0. \\ 12. & 0. & 0. \\ 0. & 0. & 8. \end{bmatrix} \;,\; K = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 7.3891 \end{bmatrix}$$

where we observe that matrix M is positive definite, while K is positive semidefinite. Because K is singular, there are two infinite eigenvalues. The finite eigenstructure of this model can be represented by

$$\Lambda = \begin{bmatrix} -2.5527 & 0. & 0. & 0. \\ 0. & -1.4473 & 0. & 0. \\ 0. & 0. & -0.4749 & 0. \\ 0. & 0. & 0. & 0.4749 \end{bmatrix}$$

$$X = \begin{bmatrix} 0. & 0. & 0.9499 & -0.9499 \\ 0. & 0. & 1. & 1. \\ -1. & -1. & 0. & 0. \end{bmatrix}.$$

We choose to change the natural frequencies

$$\{ \begin{array}{ccc} -2.5527 & -1.4473 & -0.4749 \end{array} \} \text{ to } \{ \begin{array}{ccc} -1 & -2 & -3 \end{array} \}.$$

Algorithm 6.4.2 gives

$$W = \begin{bmatrix} 0. & 0. & 22.1672 \\ 0. & 0. & 25.2669 \\ -0.7358 & 1.8480 & 0. \end{bmatrix}, \quad Z = \begin{bmatrix} 0. & 0. & 1. \\ 0. & 0. & 0. \\ -2.7183 & 2.6746 & 0. \end{bmatrix}$$

$$E_m = \begin{bmatrix} 12.8245 \\ -0.7999 \\ 0.0175 \end{bmatrix}, \quad E_d = \begin{bmatrix} -5.0239 \\ 0.5527 \\ -0.0369 \end{bmatrix}, \quad E_k = \begin{bmatrix} 1.9681 \\ -0.3819 \\ 0.0776 \end{bmatrix}.$$

Verification: If $\tilde{\Lambda} = diag\left(\begin{array}{cccc} \mu_1 & \mu_2 & \mu_3 & \lambda_4 \end{array} \right)$, we have

$$\|\tilde{M}X\tilde{\Lambda}^2 + \tilde{D}X\tilde{\Lambda} + \tilde{K}X\|_F = 1.5966 \times 10^{-14}$$

which shows that the multiple embedding was successful and produced no spill-over.

Example 6.4.3: Consider the model (M, D, K) where

- The matrices $M \in \mathbb{R}^{66\times66}$ and $K \in \mathbb{R}^{66\times66}$ come from the Statically Condensed Oil Rig model of the *Harwell-Boeing* set *BCSSTRUC1* [51]. The matrix M is symmetric positive definite and the matrix K is symmetric positive semidefinite.

- The damping matrix D is defined by $D = \rho I_{66}$, where $\rho = 1.55$.

This model has eight real eigenvalues $\{\lambda_1, \ldots, \lambda_8\}$, where

$$\{ \begin{array}{cccc} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{array} \} = \{ \begin{array}{cccc} -3.4628 & -3.5709 & -5.3584 & -9.2761 \end{array} \}$$

$$\{\ \lambda_5 \quad \lambda_6 \quad \lambda_7 \quad \lambda_8\ \} = \{\ -13.1972 \quad -13.4480 \quad -27.5536 \quad -44.5031\ \}$$

and 62 pairs of complex conjugate eigenvalues that are not shown here because of space limitation. The set $\{\lambda_1, \ldots, \lambda_8\}$ is changed to the set $\{\mu_1, \ldots, \mu_8\}$, where

$$\{\ \mu_1 \quad \mu_2 \quad \mu_3 \quad \mu_4\ \} = \{\ -3.32 \quad -3.75 \quad -5.05 \quad -9.07\ \}$$
$$\{\ \mu_5 \quad \mu_6 \quad \mu_7 \quad \mu_8\ \} = \{\ -13.59 \quad -13.04 \quad -27.31 \quad -42.11\ \}.$$

Algorithm 6.4.2 is then applied, giving matrices E_m, E_d and E_k as follows:

$$E_m =$$
$$diag\begin{pmatrix} 0.6697 & -0.9138 & 3.6368 & -2.4231 & 2.6340 & -2.6111 & 17.3927 & -197.1462 \end{pmatrix}$$
$$E_d =$$
$$diag\begin{pmatrix} -0.1934 & 0.2559 & -0.6787 & 0.2612 & -0.1996 & 0.1942 & -0.6312 & 4.4299 \end{pmatrix}$$
$$E_k =$$
$$diag\begin{pmatrix} 0.0558 & -0.0717 & 0.1267 & -0.0282 & 0.0151 & -0.0144 & 0.0229 & -0.0995 \end{pmatrix}$$

Matrices W and Z are not shown here, again for space limitation. The matrices $\tilde{M}, \tilde{D}$ and $\tilde{K}$ are then computed, using the update formulas (6.4.59), as a single rank-8 update of the matrices M, D and K.

Verification: The matrices $\Lambda, \tilde{\Lambda}$ and X are computed similarly as the ones in Example 6.4.2, and we have

$$\|\tilde{M}X\tilde{\Lambda}^2 + \tilde{D}X\tilde{\Lambda} + \tilde{K}X\|_F = 1.7709 \times 10^{-7}$$

which shows that the multiple embedding was successful and produced no spill-over.

Figures 6.1, 6.2 and 6.3 show the bar graph of the magnitude of the components of the matrices $M - \tilde{M}$, $D - \tilde{D}$ and $K - \tilde{K}$, respectively.

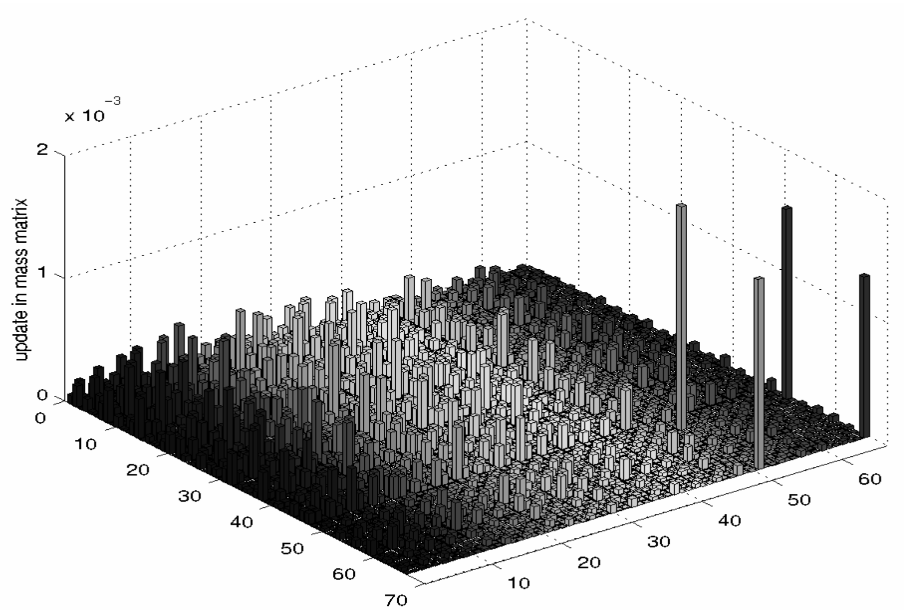

Figure 6.1: Magnitudes of the entries of the matrix $M - \tilde{M}$.

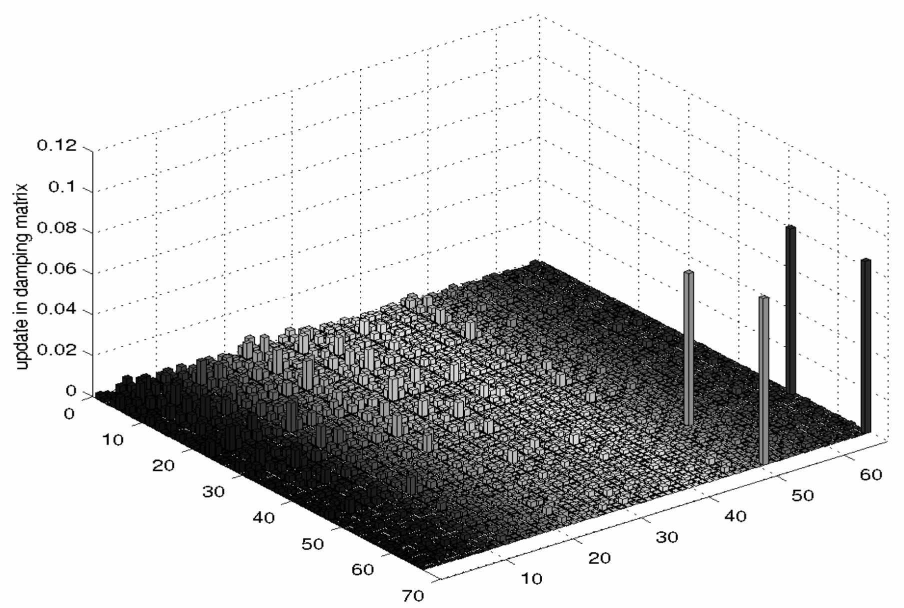

Figure 6.2: Magnitudes of the entries of the matrix $D - \tilde{D}$.

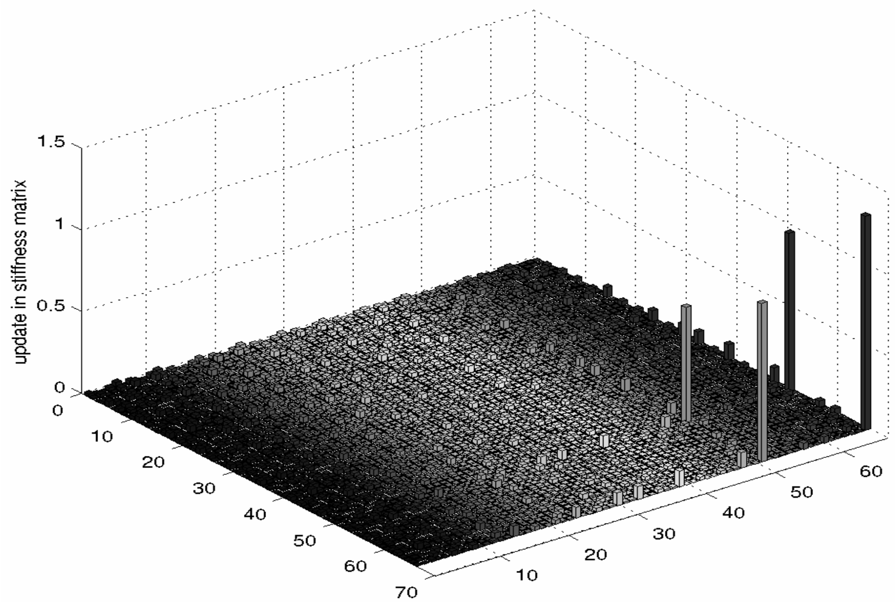

Figure 6.3: Magnitudes of the entries of the matrix $K - \tilde{K}$.

CHAPTER 7

Conclusions

In this chapter, we present some concluding remarks. In Section 7.1, we summarize our contributions and state their importance. In Section 7.2, we state some future research problems. Finally, in Section 7.3, we remark about the academic and industrial impacts of the results obtained in this dissertation.

7.1 Contributions

In this dissertation, we have developed a state estimation theory for second-order systems that model important real-life vibrating structures and we have developed a new method for Finite-Element Model Updating of an undamped symmetric positive semidefinite second-order model.

The classic state estimation theory for standard first-order systems has been extended to the descriptor systems, from where a procedure for state estimation of second-order systems has been successfully developed and applied to solve small real-life problems.

New methods have been proposed for numerical solution of Sylvester-observer equation and its descriptor counterpart; these methods have been implemented using high-performance software libraries as BLAS, LAPACK and SLICOT on benchmark problems.

Previous results on orthogonality relations between eigenvectors of the symmetric positive definite quadratic pencil have been extended to the symmetric positive semidefinite case. Using these results, a new method for the finite-element model updating of a symmetric positive semidefinite undamped model has been proposed. This method has the advantage, when compared to the other existing methods, that it does not required model reduction or mode shape expansion.

An earlier method for single real eigenvalue embedding in a positive definite model (M, D, K) has been extended to the positive semidefinite case and generalized to perform multi real eigenvalue embedding.

7.2 Future Research

Based on the theoretical investigation and numerical experiments, we propose the following problems for future research:

- In the State Estimation procedure developed in this dissertation, the experiments showed that an important source of errors is the solution of the linear system that defines the estimate $\hat{x}(t)$ in terms of the observer variable $z(t)$ and the output $y(t)$. In the standard first-order case, the design of *functional observers*, which observe directly the quantity $K\hat{x}(t)$ instead of $\hat{x}(t)$, has been previously proposed to overcome this difficulty. We believe that the design of functional observers can also be extended to the state estimation in descriptor systems.

- The presence of damping in vibrating structures is a real-life fact. Thus, including damping is essential for any design and analysis of such structures. A characterization of a possible solution to the model updating problem in the case that damping is considered has been derived in this dissertation. However, it requires the existence of a symmetric solution of a certain generalized Sylvester equation, which is presented in Subsection 6.3.3. We believe that the study of this Sylvester equation, in particular, the study of the particular cases in which it has symmetric solutions, will be important in future research on model updating of damped systems.

- In this dissertation, we have proposed a method for simultaneous multi real eigenvalue embedding in a symmetric positive semidefinite second-order model. The simultaneous eigenvalue embedding algorithm is rich in BLAS-3 computations and therefore it can be implemented using high-performance software such as LAPACK. The extension of our method to the simultaneous embedding of complex conjugate pairs constitutes an important future research topic.

7.3 Impact

The theoretical and computational results obtained in this dissertation are new contributions to the literatures of control and linear and numerical linear algebra and will advance the state-of-the-art knowledge in these areas. Besides, they are believed to have an important impact on real-life control systems design and vibration analysis.

In particular, the results on finite-element model updating are to impact a variety of industries including aerospace industries such as *The Boeing Company* and automotive industries.

REFERENCES

[1] H. Ahmadian, , G. Gladwell and F. Ismail, *Parameter Selection Strategies in Finite Element Model Updating.* Trans. ASME, Journal of Vibration and Acoustics, 1994.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammerling, A. Mckenney and D. Sorensen. *LAPACK User's Guide, Second Edition.* SIAM, Philadelphia, PA, 1995.

[3] M. Baruch, *Optimization Procedure to Correct Stiffness and Flexibility Matrices Using Vibration Data.* AIAA Journal, **16**, 11, pp 1208-1210, 1978.

[4] P. Benner, V. Mehrmann, V. Sima, S. VanHuffel and A. Varga. *SLICOT - A Subroutine Library in Systems and Control Theory.* NICONET Report 97-3, 1997. Also in Applied and Computational Control, Signal and Circuits, Birkauser, **1**, pp. 499-539, 1999.

[5] A. Berman and E. Nagy, *Improvement of a Large Analytical Model using Test Data.* AIAA Journal, **21**, 8, pp. 1168-1173, 1983.

[6] C. Bischof, B. Datta and A. Purkayastha, *A Parallel Algorithm for the Sylvester Observer Equation.* SIAM J. Sci. Comp., **17**, pp 686-698, 1996.

[7] D. Boley, *Computing the Controllability/Observability Decomposition of a Linear Time-Invariant Dynamic System, A Numerical Approach.* PhD. Thesis, Report STAN-CS-81-860, Dept. Comp. Sci., Stanford University, 1981.

[8] D. Calvetti, B. Lewis and L. Reichel,*On the Solution of Large Sylvester-observer Equation.* Num. Lin. Alg. Appl. **8**, 6-7, pp 435-452, 2001.

[9] J. Carvalho and B. Datta, *A Block Algorithm for Multi-input Eigenvalue Assignment Problem.* Proc. 1st IFAC/IEEE Symposium on Systems, Structure and Control, Prague, 2001.

[10] J. Carvalho and B. Datta, *A New Block Algorithm for the Sylvester- observer Equation Arising in State-estimation.* Proc. IEEE International Conference on Decision and Control, Orlando, Dec 7-11, 2001.

[11] J. Carvalho, B. Datta, W. Lin, C. Wang, *Eigenvalue Embedding in a Quadratic Pencil using Symmetric Low Rank Updates.* Proc. Fourth SIAM Conference on Linear Algebra in Signals, Systems and Control. Boston, Aug 13-17, 2001.

[12] J. Carvalho, K. Datta and Y. Hong, *A New Block Algorithm for Solving the Sylvester-observer Equation.* Accepted for publication at Proc. IEEE International Conference on Decision and Control, Las Vegas, Dec 10-13, 2002.

[13] J. Carvalho and B. Datta, *An Algorithm for Reduced-order State Estimation of Descriptor Systems.* Accepted for publication at Proc. IEEE International Conference on Decision and Control, Las Vegas, Dec 10-13, 2002.

[14] C-T. Chen, *Linear System Theory and Design*, 2nd ed, Holt, Rinehart and Winston, NY, 1984.

[15] C-T. Chen, *Linear System Theory and Design*, 3rd ed, Oxford Univ. Press, NY, 1999.

[16] K. Chu, *The Solution of Matrix Equations $AXB - CXD = E$ and $(YA - DZ, YC - BZ) = (E, F)$*. Lin. Alg. Appl., **93**, pp 93-105 , 1987.

[17] S. Cumming, *Design of Observers of Reduced Dynamics*. Electr. Lett. **5**, pp 213-214, 1969.

[18] B. Datta and Y. Saad, *Arnoldi Methods for Large Sylvester-like Observer Matrix Equations, and an Associated Algorithm for Partial Spectrum Assignment*. Lin. Alg. Appl. **154-156**, pp. 225-244, 1991.

[19] B. Datta, *Linear and Numerical Linear Algebra in Control Theory: Some Research Problems*. Lin. Alg. Appl. **197-198**, pp. 755-790, 1994.

[20] B. Datta, *Numerical Linear Algebra and Applications*. Brooks/Cole Publishing Company, Pacific Grove, 1995.

[21] B. Datta, S. Elhay and Y. Ram, *Orthogonality and Partial Pole Assignment for the Symmetric Definite Quadratic Pencil*. Lin. Alg. Appl., **257**, pp 29-48, 1997.

[22] B. Datta, Y. Ram and D. Sarkissian, *Multi-input Partial Pole-placement for Distributed Parameter Gyroscopic Systems* . Proceedings of the IEEE Conference on Decision and Control, Sydney, Australia, pp 4661-4665, 2000.

[23] B. Datta and D.Sarkissian, *Block Algorithms for State Estimation and Functional Observers* . Proceedings of the 2000 IEEE International Symposium on Computer-Aided Control Systems Design, Alaska, pp 19-23, 2000.

[24] B. Datta, S. Elhay, Y. Ram, and D. Sarkissian, *Partial Eigenstructure Assignment for the Quadratic Pencil*. Journal of Sound and Vibration , **230**, pp 101-110, 2000.

[25] B. Datta and D. Sarkissian, *Theory and Computations of Some Inverse Eigenvalue Problems for the Quadratic Pencil* . Contemporary Math. Vol "*Structured Matrices in Operator Theory, Control, and Signal and Image Processing*", pp. 221-240, 2001 .

[26] B. Datta, Y. Ram and D. Sarkissian, *Spectral Modification for Gyroscopic Systems.* ZAMM (Z. Angew. Math. Mechanics), **82**, pp. 191-200, 2001.

[27] B. Datta, *Numerical Methods for Control Systems Design and Analysis.* Academic Press, to appear in 2002.

[28] B. Datta and D. Sarkissian, *Feedback Control in Distributed Parameter Systems: A Solution to the partial eigenstructure assignment problem* . Mechanical System and Signal Processing, **16**, 1, 3-17, 2002.

[29] B. Datta, *Finite Element Model Updating, Eigenstructure Assignment and Eigenvalue Embedding Techniques for Vibrating Systems.* Mechanical Systems and Signal Processing in the special issue on Vibration Control, **16**, 1, 83-96, 2002.

[30] K. Datta, *The Matrix Equation $AX - XB = R$ and its Applications.* Lin. Alg. Appl, **109**, pp 91-105, 1988.

[31] K. Datta, Y. Hong and R. Lee, *Applications of Linear Transformations to Matrix Equations.* Lin. Alg. Appl. **267**, pp 221-240, 1997.

[32] J. Dongarra, I. Duff, D. Sorensen and H. Van der Vorst. *Numerical Linear Algebra for High-performance Computers.* SIAM Press, 1998.

[33] J. Doyle and G. Stein, *Robustness with Observers.* IEEE Trans. Aut. Contr., **AC-26**, pp.4-16, 1981.

[34] R. Dorf and R. Bishop, *Modern Control Systems*, 9th ed, Prentice Hall, 2001.

[35] W. Ferng, W. Lin, D. Pierce and C. Wang, *Nonequivalence Transformation of λ-matrix Eigenproblems and Model Embedding Approach to Model Tuning.* Num. Lin. Alg. Appl., **8**, pp 53-70, 2000.

[36] M. Friswell and J. Mottershed, *Finite Element Model Updating in Structural Dynamics.* Kluwer Acad. Pub., 1995.

[37] J. Gardiner , A. Laub , J. Amato and C. Moler, *Solution of the Sylvester Matrix Equation* $AXB^T + CXD^T = E$. ACM Trans. Math. Soft., **18**, 2, pp 223-231, 1992.

[38] G. Golub and C. VanLoan, *Matrix Computations*, 3rd ed., Johns Hopkins Press, 1996.

[39] G. Golub, S. Nash and C. VanLoan,*A Hessenberg-Schur method for the problem* $AX + XB = C$. IEEE Trans. Aut. Contr., **AC-24**, pp. 909-913, 1979.

[40] N. Higham, *The Test Matrix Toolbox for Matlab (version 3.0).* Numerical Analysis Report 276, Sept 95, Manchester Centre for Comput. Math., Univ. Manchester, UK.

[41] P. Hughes and R. Skelton, *Controllability and Observability of Linear Matrix-second-order Systems.* J. Appl. Mechanics, **47**, pp 415-420, 1980.

[42] D. Inman, *Vibration with Control, Measurement and Stability.* Prentice-Hall, 1989.

[43] D. Inman and C. Minas, *Matching Finite Element Models to Modal Data.* Transactions of the ASME, Journal of Vibration and Acustics, **112**, 1, pp 84-92, 1990.

[44] Kailath, *Liner Systems.* Prentice Hall, Englewood Cliffs, NJ, 1980.

[45] H. Kimura , *Geometric Structure of Observers for Linear Feedback Control Laws.* IEEE Trans. Aut. Contr., **22**, pp 846-855, 1977.

[46] A. Laub and F. Arnold, *Controllability and Observability Criteria for Multivariable Linear Second-Order Models.* IEEE Trans. Aut. Contr., **AC-29**, 2, pp 163-165, 1984.

[47] D. Luenberger, *Observing the State of a Linear System.* IEEE Trans. Mil. Electr., **8**, pp 74-80, 1964.

[48] D. Luenberger,*Observers for Multivariable Systems.* IEEE Trans. Aut. Contr. **AC-11**, pp. 190-197, 1966.

[49] D. Luenberger, *Dynamic Equations in Descriptor Form.* IEEE Trans. Aut. Contr. **AC-22**, 3, pp. 312-321, 1977.

[50] D. Luenberger, *Introduction to Dynamic Systems: Theory, Models and Applications.* Wiley, NY, 1979.

[51] The *Matrix Market* Directory on the Web. Online Reference at *http://math.nist.gov/MatrixMarket.*

[52] G. Miminis and and C. Paige, *An Algorithm for Pole Assignment of Time-invariant Multi-input Linear Systems.* Proc. 21st IEEE CDC, Orlando, pp. 62-67, 1982.

[53] C. Paige, *Properties of Numerical Algorithms Related to Computing Controllability.* IEEE Trans. Aut. Contr., **AC-26**, pp 130-138, 1981.

[54] D. Sarkissian, *Theory and Computation of Partial Eigenvalue and Eigenstructure Assignment Problems in Matrix Second-order and Distributed Parameter Systems.* Ph.D Dissertation, Northern Illinois University, 2001.

[55] B. Shafai and S. Bhattacharyya, *An Algorithm for Pole Placement in High-order Multi-variable Systems.* IEEE Trans. Aut. Control, **33**, 9, pp 870-876, 1988.

[56] E. DeSouza and S. Bhattacharyya, *Controllability, Observability and the Solution of* $AX - XB = C$. Lin. Alg. Appl. **39**, pp 167-188, 1981.

[57] C. Tsui,*On the Solution to Matrix Equation* $TA - FT = LC$. SIAM J. Matrix Anal. Appl., **14**, pp 33-44, 1993.

[58] P. VanDooren,*The Generalized Eigenstructure Problem in Linear System Theory.* IEEE Trans. Aut. Contr., **AC-26**, pp 111-129, 1981.

[59] P. VanDooren, *Reduced Order Observers: A New Algorithm and Proof.* Systems Control Lett., **4**, pp 243-251, 1984.

[60] P. VanDoren and M. Verhaegen, *On the Use of Unitary State-space Transformations.* Contemporary Math., **47**, pp 447-463, 1985.

[61] A. Varga, *Computation of Transfer-function Matrices of Generalized State-space Models.* Int. J. Control, **50**, 6, pp 2543-2561, 1989.

[62] A. Varga, *Computation of Irreducible Generalized State-Space Realizations.* Kybernetika, **26**, pp 89-106, 1990.

[63] A. Varga, *A Descriptor Systems Toolbox for MATLAB.* Proc. CACSD'2000 Symposium, Anchorage, Alaska, 2000.

[64] D. Zimmerman and M. Widengren,*Correcting Finite Element Models Using a Symmetric Eigenstructure Assignment Technique.* AIAA, **28**, pp 1670-1676, 1990.