This is to certify that the

dissertation entitled

MODELING OF SINGULARLY PERTURBED SYSTEMS
WITH APPLICATION TO MARKOV CHAINS

presented by

Rabah Wasel Aldhaheri

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Engineering

Hassan Khalil
Major professor

Date November 9, 1988

MODELING OF SINGULARLY PERTURBED SYSTEMS WITH APPLICATION TO MARKOV CHAINS

By

Rabah Wasel Aldhaheri

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering and
Systems Science

1988

ABSTRACT

MODELING OF SINGULARLY PERTURBED SYSTEMS
WITH APPLICATION TO MARKOV CHAINS

By

Rabah Wasel Aldhaheri

A new method for modeling a two-time scale system in the singularly perturbed form is presented. The method uses an "ordered" real Schur form decomposition which can be efficiently computed using standard subroutines from EISPACK. Three results are given in Chapter 2. First, it is shown that any two-time scale system can be modeled in the singularly perturbed form via a transformation into an "ordered" real Schur form, followed by balancing. Second under some conditions, the fast or the slow state variables can be chosen among the physical variables. Third, a procedure is given to achieve modeling by permuting the original state variables. The conditions under which this latter procedure works are necessary and sufficient.

In the third chapter of this dissertation, a nearly completely decomposable Markov chain that has many applications in queueing networks and inventory control is considered. This class of system is characterized by high dimensionality and ill-conditioning, which motivates researchers to look for efficient techniques to overcome these problems. We propose a general transformation to decompose this large and ill-conditioned Markov chain system into a reduced-order and well-conditioned aggregated system. It is shown that the present transformations known in the

literature are a subclass of our proposed transformation. A block diagonal transformation which is a subclass of the general one is also proposed to simplify and reduce the amount of computation involved in computing the aggregated matrix.

In the fourth chapter, an algorithm is developed to compute the optimal policies that attain the minimum average cost per stage for controlled Markov chains. This algorithm overcomes the large dimensionality and the ill-conditioning problem associated with many controlled Markov chain problems. Suboptimal policies can be computed with less computations via a modification of the above algorithm.

ACKNOWLEDGMENTS

I would like to thank my major advisor, Dr. Hassan K. Khalil for his patience, guidance and encouragement during the course of this research. Thanks to my guidance committee: Dr. R. Schlueter, Dr. R. O. Barr, and Dr. D. Yen for their comments and suggestions.

Finally, I wish to express my sincere thanks and appreciation to my parents for their continued prayers and encouragement. Special thank to my wife, Fadila, and my children for their patience, help and understanding throughout my studies.

Table of Contents

LIST OF TABLES vii

LIST OF FIGURES viii

INTRODUCTION 1

2 A REAL SCHUR FORM METHOD FOR MODELING SINGULARLY PERTURBED SYSTEMS 5

2.1 Introduction 5

2.2 Real Schur Form 10

2.3 General Modeling Method 13

2.4 Modeling with Physical Fast or Slow Variables 17

2.5 Modeling by Permutation 26

2.6 Summary 30

3 EXACT AGGREGATION OF NEARLY-COMPLETELY DECOMPOSABLE MARKOV CHAINS (NCDMC) 33

2.1 Background 33

3.2 Two-time Scale Decomposition of NCDMC 40

3.3 Properties of Exact Aggregated System 43

3.4 First Order Approximation of the Aggregated System 48

3.5 Higher Order Approximations of the Perron-Frobenius Eigenvector of NCD Systems 53

3.6 Block Diagonal Decoupling Transformation 55

3.7 Computational Considerations 57

4 OPTIMAL AND NEAR OPTIMAL CONTROL FOR NEARLY COMPLETELY DECOMPOSABLE CONTROLLED MARKOV CHAINS 64

4.1 Problem Statement 64

4.2 Algorithms for Computing the Optimal Control Policy 67

4.3 Decomposition of the Average Cost for a Fixed Policy 74

4.4 Optimal Policies for the Average Cost Problem 78

4.5 Near Optimal Policies for the Average Cost Problem 81

4.6 Example: Minimizing the Average Cost Per Stage 87

5 CONCLUSIONS AND FURTHER RESEARCH 94

APPENDIX 97

LIST OF REFERENCES 99

LIST OF TABLES

2.1 The $\|.\|_\infty$ of the ten rows of the matrix Q_1 29

4.1 The optimal policies and the average cost for different values of ε evaluated by the algorithm 4.4 90

4.2 The near optimal policy and the corresponding average cost 91

4.3 The optimal policy and the minimum average cost evaluated by he policy iteration method [41] applied to the full order system 92

LIST OF FIGURES

2.1 RC circuit with R_3 much larger than R_1 and R_2 21

2.2 Modeling flow chart 32

CHAPTER 1

INTRODUCTION

Analysis and modeling of large scale systems is one of the challenging tasks to system analysts and control engineers. Difficulties stem from the fact that most large scale systems suffer from the so called " curse of dimensionality." Of course, the problem will be worse if the large dimension of the system model is accompanied by ill-conditioning problems. So, this situation creates an urgent need for efficient model reduction and decomposition methods.

The presence of multiple time structures in many large scale systems made the singular perturbation technique one of the most effective methods for decomposing a large system into smaller subsystems of lower dimensions. The analysis and design techniques for singularly perturbed systems have been well documented in [1, 2]. The first task we face in applying singular perturbation techniques is the modeling task, that is, how to put the multi-time scale system in the singularly perturbed form. A few approaches are available in the control theory literature to handle this task. Some of those approaches rely on the underlying physical properties of the system. They require a very deep insight into the physical nature of the system in order to achieve the modeling step. Other approaches start from a given mathematical model, without further insight into the physical nature of the system; yet, they require computing the eigenvectors of the system [5, 6, 9] and in this case it is assumed that the system has distinct eigenvalues.

In Chapter 2 of this dissertation, a new method for modeling singularly perturbed systems is presented. This method makes use of the real Schur form

decomposition [16, 17]. It is numerically stable and very efficient, and it uses standard subroutines from EISPACK [18]. Three results are given in this chapter, first, a systematic procedure is given to put any two-time-scale system in the singularly perturbed form by transforming the system into an ordered real Schur form followed by balancing its elements. This procedure does not preserve the physical state variables of the system. It introduces new variables composed of linear combinations of the physical ones. Second, we give conditions under which the system can be modeled in the singularly perturbed form with all the fast or the slow state variables chosen from the original physical variables. Third, we give necessary and sufficient conditions under which all the state variables can be preserved as physical ones and the singularly perturbed form is achieved by permutation only.

Finite state Markov chains is one of the applications where large scale systems techniques have been applied. This area is very rich theoretically and it is popular to so many researchers. However, the practical usefulness of this area has been severely limited due to the extremely large dimensions of most Markov chains. The computational burden of these problems has disscourged researchers and engineers from using Markov chains for modeling purposes. In the past few years, some applications in queueing networks [25] and hydraulic scheduling [30, 44] considered a class of Markov chains where the states can be clustered into a small number of groups such that there is a strong interaction within each group while the interaction among the groups is weak compared to the interaction within the groups. This class is known as weakly coupled Markov chains or nearly completely decomposable Markov chains. In addition to the high dimensionality, this class is ill-conditioned. The strong and the weak transition probabilities between the states can be viewed as a two-time-scale property. Thus singular perturbation techniques can be applied to this class of systems.

In Chapter 3, a general transformation is given to decompose the nearly completely decomposable Markov chain into slow and fast parts and as a result of this, a high order and ill-conditioned system is reduced to a small order and well-conditioned system. This transformation enabled us to compute the exact as well as $O(\varepsilon^k)$ approximations of the steady state probability distribution of a Markov chain which is usually encountered in queueing network problems [25, 46]. It is shown that this transformation is more general than the one considered in [11, 30]. In fact, the transformation in [11, 30] is shown to be a subclass of this transformation. Moreover, it is shown that all the transformations that satisfy the conditions of Section 3.2 produce the same $O(\varepsilon)$ approximation of the aggregated matrix. A special class of "block diagonal transformations" is also given to simplify and reduce the amount of computations required to form the aggregated matrix. This transformation depends only on the dimensions of the subsystems. It is independent of the system parameters.

In Chapter 4, we consider the controlled Markov chain of the same class of systems treated in Chapter 3. The control problem which we consider in this chapter is: What are the optimal policies that minimize the average cost per stage over the infinite horizon? [40]. Again this problem is very ill-conditioned and has very large dimension. So, the conventional algorithms are not practical to handle this type of problems. In this chapter, we specialize the policy iteration algorithm of [41] to computing the optimal policies of nearly completely decomposable Markov chains. The proposed algorithm has two steps: a value determination step where the average cost and certain dual variables are computed for a fixed policy, and a policy improvement step where the dual variables from the previous step are used to compute a new policy by minimizing a Hamiltonian function. This new policy is passed to the value determination step and the cycle is repeated till convergence occurs. The new feature about this algorithm is that in the value determination step, the algorithm computes the average cost per stage by using the reduced order aggregated matrix which we

developed in Chapter 3 while in the policy improvement step, the algorithm performs the minimization on the full system. The reason for doing it this way is that in controlled Markov chains, each row of the probability transition matrix depends on one particular control; therefore, the minimization step can be performed point wise, i.e., each row can be minimized separately. This proposed algorithm would not be possible if the transformation of [11,30] is used because that transformation depends on the control policy.

Suboptimal policies can be computed with less computational effort. A modified version of the above algorithm is given to compute the suboptimal policies and the associated average cost per stage. This modified version is independent of ε. The convergence of this algorithm is shown in Section 4.5.

Finally, in Chapter 5 we draw conclusions and point to a number of research directions to which this dissertation leads.

CHAPTER 2

A REAL SCHUR FORM METHOD FOR MODELING SINGULARLY PERTURBED SYSTEMS

2.1 Introduction

Singular perturbation methods have been used in a wide range of control problems, see for example [1,2]. The interest in singular perturbation methods stems from their ability to exploit the multiple time scale structure of dynamic models to decompose analysis and design problems into simpler problems which are solved using simpler models. The first task we face in applying singular perturbation techniques is a modeling task, that is, how to mathematically describe a multiple time scale system in the singularly perturbed form. This modeling task is usually achieved with a great deal of physical insight into the system, see for example [1; chapter 1] and [2; chapters 4 and 5]. Attempts to approach this modeling task starting from a given mathematical model, and without additional information about the underlying physical system, have mainly concentrated on linear time-invariant systems, which is the class of systems considered in this chapter. In this case the standard singularly perturbed form (in a fast time scale) is described by

$$\dot{x} = Ax = \begin{bmatrix} \varepsilon A_{11}(\varepsilon) & \varepsilon A_{12}(\varepsilon) \\ A_{21}(\varepsilon) & A_{22}(\varepsilon) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.1}$$

where ε is a small positive parameter, $x_1 \in \mathbf{R}^n$ and $x_2 \in \mathbf{R}^m$. The matrix A_{22} is nonsingular uniformly in ε, i.e., $\underline{\sigma}(A_{22}) \geq K > 0$, where $\underline{\sigma}(.)$ denotes the minimum singular value and K is independent of ε. The matrices A_{ij} are O(1), i.e., their elements are

bounded from above by constants independent of ε. † The singularly perturbed system (2.1) is a two-time-scale system in the sense that its eigenvalues are clustered into two groups; n slow eigenvalues which satisfy

$$| \lambda_i | \leq K_i \varepsilon, \qquad i= 1,.....,n \tag{2.2}$$

and m fast eigenvalues which satisfy

$$| \lambda_i | \geq K_i, \qquad i= n+1,.....,n+m \ . \tag{2.3}$$

Any matrix A whose eigenvalues satisfy the conditions (2.2) and (2.3) will be referred to as a two-time-scale matrix. In general, a two-time-scale matrix does not have to be in the singularly perturbed form (2.1), since this form is dependent on the choice of state variables.

The modeling problem for linear time-invariant systems can be stated as follows: given a two-time-scale matrix A, find a similarity transformation S such that $S^{-1}AS$ is in the singularly perturbed form. From a theoretical viewpoint, this problem can always be solved by transforming A into its real Jordan form. This solution, however, is usually not acceptable from a practical viewpoint for three reasons. First, the computation of the real Jordan form presents severe practical difficulties when A is defective or close to a defective matrix [3]. This is particularly harmful when the real Jordan form computation is done for the purpose of modeling a two-time-scale system in the singularly perturbed form. A major advantage of singular perturbation techniques is a time scale decomposition based on the clustering of the eigenvalues into slow and fast groups. Such decomposition will be well-conditioned even when there are multiple or very close eigenvalues within each group [1]. This advantage

† Notice that O(ε) does not imply a lower bound. Numerically a quantity is considered O(ε) if its magnitude is sufficiently smaller than $\frac{1}{\varepsilon}$. For example, if ε = 0.01, then 10 is O(1); also 0.01 is O(1) since it is O(ε) and any O(ε) quantity is O(1).

will be lost if we have to compute the real Jordan form because multiplicity or almost multiplicity of eigenvalues within each group will result in ill-conditioned computations. Second, the computation of the real Jordan form could be burden, especially when the dimension of A is high. Third, the transformation into real Jordan form will, in general, change all state variables. This is a disadvantage when the original state variables are physical ones. In such a case it is preferable in solving the modeling problem to try to retain as many original state variables as possible. The optimum case, of course, is to be able to model the system by only permuting the original state variables. However, this in not always possible.

Previous work on the modeling problem is given in [4] – [15]. In [4] – [8] the main element of the modeling procedure is finding a solution of a matrix Riccati-type equation, $L \in \mathbf{R}^{m \times n}$ say, such that the similarity transformation $\begin{bmatrix} I_n & 0 \\ L & I_m \end{bmatrix}$ transforms the system into a block triangular form $\begin{bmatrix} B_1 & * \\ 0 & B_2 \end{bmatrix}$, with the eigenvalues of B_1 being the slow ones and the eigenvalues of B_2 being the fast ones. The methods of [4] – [6] start by grouping the state variables into two groups and then check a norm condition that is sufficient for the existence of L. The three methods differ in the way they achieve the state grouping. It is done by comparing row norms of the matrix A in [4], by comparing row norms of the eigenvectors corresponding to slow eigenvalues in [5] and by comparing row norms of all eigenvectors in [6]. In [7] the matrix L is determined from the eigenvectors corresponding to the slow eigenvalues. In [8] it is determined without calculating the eigenvalues and the eigenvectors of A, but with manipulation of the characteristic polynomial of A that requires decomposing this polynomial as a product of two polynomials corresponding to the slow and fast eigenvalues.

Another approach is taken in [9], where left and right eigenvectors are computed to determine the participation of each mode in every state variables. This information is used to classify state variables as slow and fast ones. The method is useful only when the modeling problem can be solved by permutation. In all above methods, whenever eigenvectors are computed it is assumed that the corresponding eigenvalues are distinct.

A special case that received a great deal of attention is the case when the matrix A can be represented as

$$A = A_0 + \varepsilon A_1 . \tag{2.4}$$

When A_0 is nonsingular, the system has only one time scale as $\varepsilon \to 0$. When A_0 is singular with semisimple null structure, the system has at least two time scales and can be transformed into the singularly perturbed form by a transformation that is dependent only on A_0. One such transformation is given in [10]. If the rank of the matrix A_0 is assumed to be m and its dimension to be $(n + m)$, then the transformation $\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$ is chosen such that the rows of $P_1 \in \mathbf{R}^{n \times (n+m)}$ span the left null space of A_0, i.e., $P_1 A_0 = 0$, while the rows of $P_2 \in \mathbf{R}^{m \times (n+m)}$ span the left range space of A_0. Applying this transformation to (2.1) gives us

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} \varepsilon A_{11}(\varepsilon) & \varepsilon A_{12}(\varepsilon) \\ \varepsilon A_{21}(\varepsilon) & A_{22}(\varepsilon) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{2.5}$$

where, $y_1 = P_1 x$ and $y_2 = P_2 x$. This case and its multiple time scale extensions have been studied by several investigators, e.g., [10] – [15]. It should be noted that the system (2.5) is a special case of the singularly perturbed system (2.1), where the two-one block is $O(\varepsilon)$. This special form is a result of the choice of P_2 described above. It is shown in [13] that one does not have to choose P_2 such that its rows

span the left range space of A_0. Rather, it is enough to choose P_2 as any matrix that will result in a nonsingular $\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$. such a choice will still bring the system into the singularly perturbed form (2.1), but $A_{21}(\varepsilon)$ will be O(1), in general. This observation on the choice of P_2 is behind the modeling with physical fast variables idea of section 2.4. It is also behind the modeling of Markov chains in Chapter 3.

In this chapter a new method for modeling singularly perturbed systems is presented. The method makes use of the real Schur decomposition [16] – [17]. In Section 2.2 we provide some background material on the computation of the real Schur decomposition. The computation is numerically stable and efficient. It uses standard subroutines from EISPACK [18] and from [20,21]. In fact, the real Schur decomposition is the standard routine to compute eigenvalues in EISPACK. Therefore, the computational effort needed in our method is basically the computational effort needed to compute eigenvalues using EISPACK. The method can handle matrices of dimension of the order of a few hundreds. The stable Schur decomposition can almost always be used in lieu of the Jordan decomposition [16]. It is gaining popularity in solving control problems, as for example, in solving Riccati equation [19].

In Section 2.3 we give a procedure for modeling any two-time-scale system in the singularly perturbed form by transforming the system into an "ordered" real Schur form, and then balancing the elements of the Schur form. As a byproduct, the singularly perturbed system will be block triangular. In Section 2.4 we give conditions under which the system can be modeled in the singularly perturbed form with all fast or all slow variables chosen from the original state variables. It is shown that those conditions hold whenever A takes the form (2.4) and A_0 is singular with semisimple null structure. Finally, in Section 2.5 we give necessary and sufficient conditions for the existence of a permutation that transforms a two-time-scale system into the

singularly perturbed form.

2.2 Real Schur Form

For any real matrix $A \in \mathbf{R}^{r \times r}$ there exists an orthogonal matrix $Q \in \mathbf{R}^{r \times r}$,i.e., $Q^TQ = I_r$, such that

$$Q^TAQ = T = \begin{bmatrix} S_{11} & S_{12} & . & . & . & S_{1N} \\ 0 & S_{22} & . & . & . & S_{2N} \\ . & 0 & S_{33} & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & . & 0 & . & . & S_{NN} \end{bmatrix} \tag{2.6}$$

where each S_{ii} is either 1×1 matrix corresponding to a real eigenvalue or 2×2 matrix corresponding to a pair of complex conjugate eigenvalues of A. The transformation (2.6) is called the real Schur decomposition, and T is called the real Schur form (RSF) of A. Moreover, suppose T is partitioned as

$$T = Q^TAQ = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \tag{2.7}$$

where the blocks $T_{11} \in \mathbf{R}^{p \times p}$ and $T_{22} \in \mathbf{R}^{(r-p) \times (r-p)}$ have no common eigenvalues i.e., $\lambda(T_{11}) \cap \lambda(T_{22}) = \varnothing$, then the first p orthogonal columns of Q span the invariant subspace associated with the eigenvalues of T_{11} [16].

The most common general purpose algorithm used to compute (2.6) is the double-Francis QR algorithm . Before the QR process is applied, A is initially reduced to upper Hessenberg matrix $H \in \mathbf{R}^{r \times r}$ and this is accomplished by a finite sequence of similarity transformations known as elementary reflector matrices. This algorithm is very efficient and numerically stable and it is the most preferred method for calculating the eigenvalues of a real matrix. For further details on the real Schur decomposition and its computation, the reader may consult any standard text book on

numerical linear algebra, e.g. [16,17].

The above algorithm does not guarantee any special ordering of the eigenvalues along the diagonal of T. In our work with two-time-scale matrices we are interested in a real Schur form in which the eigenvalues are clustered into two separate groups: slow and fast eigenvalues, i.e., if T is partitioned as in (2.7), then the eigenvalues of T_{11} will be the slow (fast) ones and the eigenvalues of T_{22} will be the fast (slow) ones. Fortunately, the ordering of eigenvalues along the diagonal of T can be changed by interchanging adjacent pairs of eigenvalues using orthogonal transformations [16,pp. 240 - 242]. Stewart [20] introduced an algorithm written as a Fortran subroutine [HQR3] to compute a RSF with the diagonal blocks S_{ii} ordered so that the eigenvalues appear in descending order of absolute value along the diagonal of T. Subroutine (HQR3) achieves the ordering of eigenvalues we need in (2.7). Actually, it goes beyond what we need because in our case the eigenvalues of T_{11} and T_{22} are not required to be of any particular ordering. However, for convenience, we will use this subroutine to compute the RSF (2.7) where in this case T_{11} and T_{22} have the fast and the slow eigenvalues respectively. To fix notation we will proceed with this particular ordering.

Definition (ORSF): If the diagonal blocks of the real Schur form (2.6) are ordered such that it can be partitioned as in (2.7) with

$T_{11} \in \mathbf{R}^{m\times m}$, $\lambda(T_{11}) = \{\lambda_i\}_{i=1}^{m} =$ the fast (largest in magnitude) eigenvalues

and

$T_{22} \in \mathbf{R}^{n\times n}$, $\lambda(T_{22}) = \{\lambda_j\}_{j=m+1}^{r} =$ the slow (smallest in magnitude) eigenvalues

where n=r-m, then T is called ordered real Schur form (ORSF).

We will always deal with a two-time-scale matrix A with m fast eigenvalues and n slow ones. These will be the same integers used in the ORSF of A. In this case the first m orthogonal columns of Q form a basis for the invariant subspace of A

associated with the fast eigenvalues, $\lambda(T_{11})$.

The following sequence of subroutines is used to compute an ordered real Schur form (ORSF):

1. **ORTHES** to reduce the two-time scale matrix A to upper Hessenberg form using orthogonal similarity transformation.
2. **ORTRAN** to accumulate the orthogonal similarity transformation used in reduction of the matrix A to upper Hessenberg form
3. **HQR3** to compute the real Schur form in descending order of absolute value along its diagonal

The subroutines **ORTHES** and **ORTRAN** are available in Eispack[18]

2.3 General Modeling Method

In this section we show that any two-time-scale system can be put into the singularly perturbed form by an orthogonal transformation followed by scaling. The first step is to transform the matrix A into an ORSF using an orthogonal matrix Q, i.e.,

$$T = Q^T A Q = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \tag{2.8}$$

where T_{11}, T_{12} and T_{22} are $m \times m$, $m \times n$ and $n \times n$, respectively, and $\lambda(T_{11})$ and $\lambda(T_{22})$ are the fast and the slow eigenvalues of the system. We discussed in section 2 how to perform this step. For this system to be in the singularly perturbed form, the elements of T_{11} and T_{12} should be $O(|\lambda(T_{11})|)$ and elements of T_{22} should be $O(|\lambda(T_{22})|)$. That is, T_{11} and T_{12} should be O(1) while T_{22} should be $O(\varepsilon)$. In general the elements of T_{ij} do not satisfy these requirements and balancing may be needed to reduce the order of magnitude of the elements of T_{11} and T_{12} to the order of magnitude of $\lambda(T_{11})$, and the order of magnitude of elements of T_{22} to the order of magnitude of $\lambda(T_{22})$. The idea of matrix balancing is well known in numerical analysis, e.g. [22,23]. The Eispack subroutine BALANC [18] balances a real matrix by scaling. This subroutine, however, balances only the irreducible part of the matrix, i.e., that part of the matrix which can not be reduced to a block triangular structure by permutation. With the quasi-triangular structure of the RSF, it is clear that subroutine BALANC is not useful in our case. Therefore, we devised our own algorithm to balance the elements of T. Similar to BALANC, the diagonal scaling elements are chosen to be exact powers of the radix base used to implement the algorithm on a digital computer. In this case no rounding errors will occur if the the algorithm is implemented in floating point arithmetic.

To describe the balancing algorithm, consider the RSF (2.6) and suppose that the diagonal blocks S_{ii} are ordered as in the ORSF. The goal of the balancing algorithm is to satisfy the condition

$$L_i \le \beta \mid \lambda(S_{ii}) \mid , \qquad i = 1,\ldots\ldots, N \tag{2.9}$$

where L_i is the largest (in magnitude) element of $(S_{ii}, S_{i,i+1}\ldots\ldots\ldots, S_{iN})$ and β is the radix base employed, usually 2, 10 or 16 . If condition (2.9) is satisfied, the system will be in the singularly perturbed form, provided that $\varepsilon \ll \frac{1}{\beta}$. We start by balancing the on-diagonal matrices S_{ii}. Of course 1×1 blocks need no balancing. For 2×2 matrices, the S_{ii} elements could be much larger than the magnitude of the complex eigenvalues. For example, the matrix

$$\begin{bmatrix} -(\alpha+1) & -(\alpha+1) \\ \alpha + \dfrac{1}{\alpha+1} & \alpha \end{bmatrix}$$

has eigenvalues $\frac{-1}{2} \pm j \frac{\sqrt{3}}{2}$, irrespective of the value of α. Hence its elements can be made arbitrarily larger than the eigenvalues. To balance a 2×2 matrix, consider $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ with complex conjugate eigenvalues $\lambda = \alpha \pm j\gamma$, $\gamma \neq 0$. Our goal is to achieve the condition $\mid a_{ij} \mid \le \beta \mid \lambda \mid$. We do this in two steps. First we achieve the conditions

$$\mid a_{ii} \mid \le \beta \mid \lambda \mid \qquad \text{and} \qquad \mid a_{12}\, a_{21} \mid \le \beta \mid \lambda \mid^2 \tag{2.10}$$

and then balance the outer diagonal elements by scaling. Notice that the condition $\mid a_{12}a_{21} \mid \le \beta \mid \lambda \mid^2$ guarantees that $\mid a_{ij} \mid \le \beta \mid \lambda \mid$ can be achieved by scaling. Notice also that (2.10) will always hold when a_{11} and a_{22} have the same sign, because in this case $|a_{ii}| < 2\,|\lambda| \le \beta\,|\lambda|$ and $|a_{12}\, a_{21}| \le |\lambda|^2 \le \beta\,|\lambda|^2$. If (2.10) does not hold we can achieve it by using a Givens rotation, see [16] problem P7.4-7. Let θ be given by

$$\cos(2\theta) = \frac{a_{12} + a_{21}}{\sqrt{(a_{11} - a_{22})^2 + (a_{12} + a_{21})^2}} \tag{2.11}$$

It can be verified that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \alpha & \delta_{12} \\ \delta_{21} & \alpha \end{bmatrix} \tag{2.12}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$ and $\delta_{12}\delta_{21} = -\gamma^2$.

Now, if one of the outer diagonal elements, a_{12} say, violates the condition $|a_{12}| \leq \beta|\lambda|$, it is multiplied by $\beta^{-\sigma}$, where σ is a positive integer determined from

$$\beta^{\sigma} \leq \frac{|a_{12}|}{|\lambda|} \leq \beta^{\sigma+1} \tag{2.13}$$

The other outer diagonal element is multiplied by β^{σ}. The second condition of (2.10) guarantees that $\beta^{\sigma}|a_{21}| \leq \beta|\lambda|$. Notice that during balancing the block S_{ii}, the blocks S_{ij}, for $i > j$ and S_{ji}, for $j < i$, will change.

After balancing the diagonal blocks S_{ii}, we proceed to balance the off diagonal blocks S_{ij}. For that we compute the ratio $\frac{L_i}{|\lambda(S_{ii})|}$. If

$$\beta^{\sigma} \leq \frac{L_i}{|\lambda(S_{ii})|} \leq \beta^{\sigma+1}, \qquad \sigma > 0, \tag{2.14}$$

the ith block row is multiplied by $\beta^{-\sigma}$ and the ith block column is multiplied by β^{σ}. This procedure is performed starting from the block S_{NN} and moving up, so that multiplication of the columns by a scaling factor does not alter previously balanced rows.

The previous procedure puts the system in a singularly perturbed form with the first m state variables as the fast ones and the last n variables as the slow ones. The block permutation $\begin{bmatrix} 0 & I_n \\ I_m & 0 \end{bmatrix}$ will put the system in the standard singularly perturbed form (2.1).

Example 2.1: Consider the following system

$$\dot{x} = \begin{bmatrix} -15.810 & -4.548 & -13.181 & -13.862 & -0.104 \\ 13.768 & 3.258 & 13.677 & 13.213 & 0.402 \\ 2.745 & 3.844 & -0.703 & -0.736 & 1.093 \\ 5.316 & -0.938 & 6.153 & 6.955 & -1.022 \\ -2.447 & -2.981 & -5.547 & -1.183 & -2.979 \end{bmatrix} x$$

The eigenvalues of this system are : -5.032, $-1.97 \pm j\,0.1431$ and $-0.154 \pm j\,0.149$, which can be clustered as three fast eigenvalues and two slow eigenvalues. A reasonable choice of ε is $\varepsilon = 0.1$.

The ORSF of this matrix, computed using the procedure of section 2.2, is

$$T = \begin{bmatrix} -5.032 & -0.133 & 0.747 & 32.530 & 11.377 \\ 0 & -1.736 & -3.5025 & -11.350 & 0.4454 \\ 0 & 0.0215 & -2.204 & 4.450 & -3.550 \\ 0 & 0 & 0 & -0.2431 & 0.0555 \\ 0 & 0 & 0 & -0.5421 & -0.0646 \end{bmatrix}$$

Taking $\beta = 2$, it is clear that T is not balanced; so we should apply the balancing algorithm we discussed in this section. The 2×2 diagonal blocks have diagonal elements of the same sign, therefore no rotation is needed in this example and the diagonal scaling matrix that put the system in singularly perturbed form is $D = diag\,[\,2, 2, 2, \frac{1}{2}, 1\,]$. For this D, the ORSF becomes

$$D^{-1}TD = \begin{bmatrix} -5.032 & -0.133 & 0.747 & 8.1325 & 5.6885 \\ 0 & -1.736 & -3.5025 & -2.8375 & 0.2227 \\ 0 & 0.0215 & -2.204 & 1.1125 & -1.775 \\ 0 & 0 & 0 & -0.2431 & 0.111 \\ 0 & 0 & 0 & -0.2711 & -0.0646 \end{bmatrix}.$$

Finally, block permutation yields the following singularly perturbed form

$$\begin{bmatrix} -0.2431 & 0.111 & 0 & 0 & 0 \\ -0.2711 & -0.0646 & 0 & 0 & 0 \\ 8.1325 & 5.6885 & -5.032 & -0.133 & 0.747 \\ -2.8375 & 0.2227 & 0 & -1.736 & -3.5025 \\ 1.1125 & -1.775 & 0 & 0.0215 & -2.204 \end{bmatrix}$$

2.4 Modeling With Physical Fast or Slow Variables

When the state variables of a given two-time-scale system are physical variables, it is usually preferable in transforming the system into the singularly perturbed form to retain as many physical variables as possible. The general modeling method of section 2.3 does not , in general, retain any of the original states variables. In this section we show that if some additional conditions are satisfied, then it is possible to obtain a singularly perturbed model in which all the fast or all the slow variables are chosen from the original physical state variables.

Theorem 2.1

Let A be a two-time-scale matrix, and (2.8) be an ordered real Schur decomposition. Suppose that T_{11} and T_{12} are O(1), T_{22} is O(ε) and there exists a permutation matrix matrix P such that

$$P\,Q = P\,[Q_1 \quad Q_2] \triangleq W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \tag{2.15}$$

where the $m \times m$ matrix W_{21}^{-1} is O(1). Then the transformation matrix

$$U^{-1}P = \begin{bmatrix} I_n & -W_{11}W_{21}^{-1} - \varepsilon E \\ 0 & I_m \end{bmatrix} P \tag{2.16}$$

where E is O(1), puts A into the singularly perturbed form.

Remarks

1. The permutation of the rows of Q is equivalent to the permutation of the original state variables. It does not affect the ORSF. In particular, $T = Q^T A Q$ can be written as $T = (PQ)^T\, P\, A\, P^T\, (PQ)$.
2. The m columns of Q_1 are orthogonal. Hence it is always possible to permute its rows such that W_{21} is nonsingular. The assumption that W_{21}^{-1} is O(1) is an additional requirement on the linear independence of the rows of W_{21}. It is

equivalent to saying that the minimum singular value of W_{21} is bounded from below by a constant independent of ε.

3. The assumption that W_{21}^{-1} is O(1) guarantees that both U and U^{-1} are O(1).

4. The term εE in (2.16) is included to allow dropping $O(\varepsilon)$ terms of $W_{11} W_{21}^{-1}$, which simplifies the transformation.

Proof

From the fact that $W^T W = I_r$, the following expressions can be verified :

- $W_{12}^{-T} = W_{12} - W_{11} W_{21}^{-1} W_{22}$
- $W_{12}^T W_{11} W_{21}^{-1} + W_{22}^T = 0$
- $W_{21}^{-1} = W_{21}^T + W_{11}^T W_{11} W_{21}^{-1}$

The first of these expressions shows that W_{12}^{-1} is O(1). Using the above expressions, we obtain

$$U^{-1}PAP^TU = \begin{bmatrix} -\varepsilon E W_{21} & W_{12}^{-T} - \varepsilon E W_{22} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} W_{11}^T & W_{21}^{-1} + \varepsilon W_{11}^T E \\ W_{12}^T & \varepsilon W_{12}^T E \end{bmatrix} \triangleq \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

where

$$B_{11} = W_{12}^{-T} T_{22} W_{12}^T - \varepsilon E \, [W_{21} T_{11} W_{11}^T + W_{21} T_{12} W_{12}^T + W_{22} T_{22} W_{12}^T] \triangleq \varepsilon \hat{A}_{11}$$

$$B_{12} = -\varepsilon \, [E W_{21} T_{11} W_{21}^{-1} - W_{12}^{-T} T_{22} W_{12}^T E] - \varepsilon^2 E$$

$$\times [W_{21} T_{11} W_{11}^T + W_{21} T_{12} W_{12}^T + W_{22} T_{22} W_{12}^T] \, E \triangleq \varepsilon \hat{A}_{12}$$

$$B_{21} = W_{21} T_{11} W_{11}^T + W_{21} T_{12} W_{12}^T + W_{22} T_{22} W_{12}^T \triangleq \hat{A}_{21}$$

$$B_{22} = W_{21} T_{11} W_{21}^{-1} + \varepsilon \, [W_{21} T_{11} W_{11}^T + W_{21} T_{12} W_{12}^T + W_{22} T_{22} W_{12}^T] \, E \triangleq \hat{A}_{22}$$

It is clear that $\hat{A}_{ij}$ are O(1); hence the system is in the singularly perturbed form.

Theorem 2.1 shows a procedure to model a two-time-scale system in the singularly perturbed form with physical fast variables. The significance of this theorem depends

on how general its conditions are. The following theorem shows that the conditions of Theorem 2.1 hold for the case $A = A_0 + \varepsilon A_1(\varepsilon)$, where A_0 is singular and has a semisimple null structure. This case, and its multiple-time scale extensions, have been studied by many researchers, e.g. [10] – [15].

Theorem 2.2

Let $A(\varepsilon) = A_0 + \varepsilon A_1(\varepsilon)$, where A_0 is singular and has a semisimple null structure, $A_1(\varepsilon)$ is O(1), and $\varepsilon > 0$ is sufficiently small. Then any ordered real Schur form decomposition of A has the properties

(i) T_{11} and T_{12} are O(1)

(ii) T_{22} is O(ε)

(iii) There exists a permutation P such that $PQ = W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$, where W_{21}^{-1} is O(1).

Proof

The semisimple null structure of A_0 implies that there exists an orthogonal matrix Q_0 such that

$$Q_0^T A_0 Q_0 = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}$$

where R_{11} is nonsingular. Moreover, there exists a permutation of the rows of Q_0 such that

$$PQ_0 = W_0 = \begin{bmatrix} W_{11}^0 & W_{12}^0 \\ W_{21}^0 & W_{22}^0 \end{bmatrix}$$

where W_{21}^0 is nonsingular. Since W_0 is independent of ε, $(W_{21}^0)^{-1}$ is O(1).

Therefore $A(\varepsilon)$ can be represented as

$$A(\varepsilon) = P^T W_0 D(\varepsilon) W_0^T P = P^T W_0 \begin{bmatrix} D_{11}(\varepsilon) & D_{12}(\varepsilon) \\ \varepsilon D_{21}(\varepsilon) & \varepsilon D_{22}(\varepsilon) \end{bmatrix} W_0^T P \qquad (2.17)$$

where $D_{ij}(\varepsilon)$ are O(1). Using (2.17) in any ordered real Schur decomposition (2.8) of $A(\varepsilon)$ yields

$$T_{11} = L_{11}^T M_{11} + \varepsilon L_{21}^T M_{21} \tag{2.18}$$

$$T_{12} = L_{11}^T M_{12} + \varepsilon L_{21}^T M_{22} \tag{2.19}$$

$$0 = L_{12}^T M_{11} + \varepsilon L_{22}^T M_{21} \tag{2.20}$$

$$T_{22} = L_{12}^T M_{12} + \varepsilon L_{22}^T M_{22} \tag{2.21}$$

where

$$L(\varepsilon) = W_0^T P Q(\varepsilon) = W_0^T W = \begin{bmatrix} L_{11}(\varepsilon) & L_{12}(\varepsilon) \\ L_{21}(\varepsilon) & L_{22}(\varepsilon) \end{bmatrix}$$

and

$$M(\varepsilon) = \begin{bmatrix} D_{11}(\varepsilon) & D_{12}(\varepsilon) \\ D_{21}(\varepsilon) & D_{22}(\varepsilon) \end{bmatrix} L(\varepsilon) = \begin{bmatrix} M_{11}(\varepsilon) & M_{12}(\varepsilon) \\ M_{21}(\varepsilon) & M_{22}(\varepsilon) \end{bmatrix}$$

The fact that L_{ij} and D_{ij} are O(1) implies that T_{11} and T_{12} are O(1). Moreover, since $|\lambda_i(T_{11})|$ are bounded from below by a constant independent of ε, it follows that, for sufficiently small ε, T_{11}^{-1} is O(1). Using this in (2.18) shows that L_{11}^{-1} and M_{11}^{-1} are O(1), which implies from (2.20) that L_{12} is $O(\varepsilon)$. Finally (2.21) shows that T_{22} is $O(\varepsilon)$. Now, the fact that L_{12} is $O(\varepsilon)$ together with $L^T L = I_r$ imply that L_{21} is $O(\varepsilon)$. Using this in

$$W_{21} = W_{21}^0 L_{11} + W_{22}^0 L_{21} = W_{21}^0 L_{11} + O(\varepsilon)$$

proves that W_{21}^{-1} is O(1), for sufficiently small ε.

Theorem 2.2 shows that there is a large class of interesting physical problems for which the conditions of Theorem 2.1 are satisfied. It is important, however, to note that Theorem 2.1 does not require that the matrix $A(\varepsilon)$ be modeled in the form $A = A_0 + \varepsilon A_1$, where A_0 is singular with a semisimple null structure. This modeling

step in itself may not be easy and may require physical insight into the problem. For example, in [10] this form was obtained by recognizing a network structure with weak coupling and ε was taken as a measure of the weak coupling. In another example [1], ε was introduced as a reciprocal of a high-gain parameter in a high gain feedback system. Without such insight it may be difficult to model A as $A = A_0 + \varepsilon A_1$. Theorem 2.1 alleviates the need for this modeling problem, because its conditions are checked on an ordered real Schur decomposition of the matrix A itself, i.e., we do not need to know the matrix A_0.

We illustrate Theorems 2.1 and 2.2 by the following two examples.

Example 2.2 Consider the following RC circuit

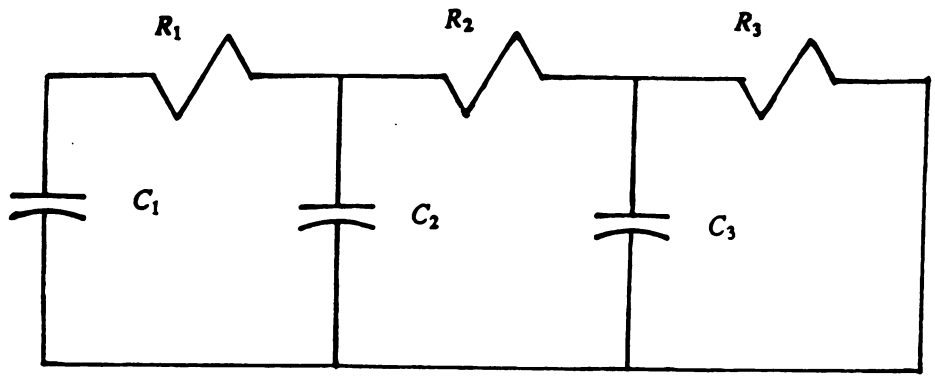

Fig. 2.1 RC circuit with R_3 much larger than R_1 and R_2

Choose the capacitance voltages V_1, V_2 and V_3 as state variables and let $C_1 = C_3 = C$, $C_2 = \frac{C}{2}$, $R_1 = R_2 = r$, $R_3 = R$ and $\frac{r}{R} = \varepsilon$. Here a small value of ε represents a weak connection between the capacitors. Let $t = \frac{t_d}{RC}$, $\tau = \frac{t_d}{r\,C}$ and $\varepsilon = \frac{t}{\tau}$, where t_d is the physical time variable.

The state equation for the above circuit is given by

$$\varepsilon\frac{dV}{dt} = \frac{dV}{d\tau} = A(\varepsilon)\,V = \begin{bmatrix} -1 & 1 & 0 \\ 2 & -4 & 2 \\ 0 & 1 & -1-\varepsilon \end{bmatrix} V, \quad \text{where } V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

which takes the form $A(\varepsilon) = A_0 + \varepsilon A_1$, where A_0 is singular with a semisimple null structure.

Let $\varepsilon = 0.1$. In this case the eigenvalues of A are -5.01021, -1.05184 and -0.03795, which shows the two-time-scale property of A. The ORSF of A, computed as in section 2.2, is given by

$$\begin{bmatrix} -5.01021 & -0.06573 & 1.41267 \\ 0 & -1.05184 & -0.00601 \\ 0 & 0 & -0.03795 \end{bmatrix}$$

It is clear that T_{12} is O(1) and T_{22} is O(ε). We can consider T_{11} to be O(1), by viewing 5.01021 as an O(1) quantity. This is a close call because $\frac{1}{\varepsilon} = 10$. Such a situation arises because $\varepsilon = 0.1$ is not very small. For smaller values of ε the O(1) elements will be clearly far from $\frac{1}{\varepsilon}$.

The matrix Q, partitioned as in (2.15), is given by

$$Q = \begin{bmatrix} -0.23483 & 0.67639 & 0.69810 \\ 0.94173 & -0.01963 & 0.33580 \\ -0.24084 & -0.73628 & 0.63237 \end{bmatrix}$$

Here $Q_{21}^{-1} = \begin{bmatrix} 1.05468 & -0.02812 \\ -0.34499 & -1.34898 \end{bmatrix}$ = O(1). Hence all the conditions of Theorem 2.1 are satisfied with $P = I_3$. Therefore $W = Q$ and

$$-W_{11}\,W_{21}^{-1} = [\ 0.481 \quad 0.906\ ] = [0.5 \quad 1.0\ ] - [\ 0.019 \quad 0.094\ ]$$

To simplify the transformation we may drop the O(ε) part [0.019 0.094]. This can be done by choosing $-E = [\ 0.19 \quad 0.94]$ such that $-W_{11}\,W_{21}^{-1} - \varepsilon\,E = [\ 0.5 \quad 1.0\ ]$.

Therefore

$$U^{-1} = \begin{bmatrix} 1.0 & 0.5 & 1.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

and the singularly perturbed system is given by

$$\begin{bmatrix} \frac{dy}{d\tau} \\ \frac{dV_2}{d\tau} \\ \frac{dV_3}{d\tau} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\varepsilon \\ 2 & -5 & 0 \\ 0 & 1 & -1-\varepsilon \end{bmatrix} \begin{bmatrix} y \\ V_2 \\ V_3 \end{bmatrix}$$

The eigenvalues of the slow and the fast subsystems are $\lambda(A_{11} - A_{12} A_{22}^{-1} A_{21}) = -0.036364$ and $\lambda(A_{22}) = -1.1$ and -5 which are $O(\varepsilon^2)$ and $O(\varepsilon)$ close to the exact eigenvalues, respectively.

In this example we introduced one state variable y as a slow state and retained V_2 and V_3 as fast states. On the other hand, if we choose the transformation

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} 0.4 & 0.2 & 0.4 \\ -1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & -1.0 \end{bmatrix},$$

the singularly perturbed form (2.5) can be written as

$$\begin{bmatrix} \frac{dy_1}{d\tau} \\ \frac{dy_2}{d\tau} \\ \frac{dy_3}{d\tau} \end{bmatrix} = \begin{bmatrix} -0.4\varepsilon & -0.16\varepsilon & 0.24\varepsilon \\ 0 & -3 & -2 \\ \varepsilon & -2+0.4\varepsilon & -3-0.6\varepsilon \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

By using this transformation, none of the state variables are kept as physical ones.

Example 2.3 To illustrate that it is not necessary to model the matrix A(ε) in the form $A = A_0 + \varepsilon A_1$, where A_0 is singular, let us consider the following system

$$A(\varepsilon) = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2-\varepsilon \end{bmatrix}$$

It is clear that A_0 is nonsingular. This suggests that A(ε) has only one time scale. However, for $\varepsilon = 0.1$ the eigenvalues for A are -0.2083, -1.6085 and -3.2832, which shows a two-time-scale property with one slow and two fast eigenvalues. This does not represent a contradiction. For, even though A(0) is nonsingular, its minimum singular value is 0.2083, i.e. , it is O(ε). We apply Theorem 2.1, computing the ORSF as in section 2.2, to obtain

$$T = \begin{bmatrix} -3.2832 & 0 & 0 \\ 0 & -1.6085 & 0 \\ 0 & 0 & -0.2083 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} -0.3172 & -0.5869 & -0.7450 \\ 0.7243 & 0.3571 & -0.5898 \\ -0.6122 & 0.7267 & -0.3118 \end{bmatrix}$$

Notice that the ORSF is diagonal because A(ε) is symmetric [16]. All the conditions of Theorem 2.1 are satisfied with $P = I_3$ and E is chosen to yield

$$U^{-1} = \begin{bmatrix} 1.0 & 0.8 & 0.4 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

The singularly perturbed system is given by

$$U^{-1}AU = \begin{bmatrix} -0.2 & -0.04 & 0.04 \\ 1.0 & -2.80 & 0.60 \\ 0 & 1.0 & -2.10 \end{bmatrix}$$

The eigenvalues of the slow subsystem is -0.2083, compared with the exact eigenvalue -0.2083, while the eigenvalues of the fast subsystem are -1.6 and -3.3, compared with the exact eigenvalues -1.6085 and -3.2832.

Note that if the conditions of Theorem 2.1 hold on A^T, where A^T is the transpose of the two-time-scale matrix A, then all the slow variables can be chosen from the original physical state variables while the fast states are introduced as a linear combination of all or some of the original state variables. Such a result can be easily seen if we apply the transformation matrix (2.16) on A^T to yield

$$U^{-1}PA^TP^TU = \begin{bmatrix} \varepsilon B_{11} & \varepsilon B_{12} \\ B_{21} & B_{22} \end{bmatrix} \tag{2.22}$$

where B_{ij} are O(1). If we take the transpose of the two sides of (2.22), we obtain

$$U^TPAP^TU^{-T} = \begin{bmatrix} \varepsilon B_{11}^T & B_{21}^T \\ \varepsilon B_{12}^T & B_{22}^T \end{bmatrix} \tag{2.23}$$

Using the scaling matrix $S^{-1} = \begin{bmatrix} I_n & 0 \\ 0 & \frac{I_m}{\varepsilon} \end{bmatrix}$, the system (2.23) becomes

$$S^{-1}U^TPAP^TU^{-T}S = \begin{bmatrix} \varepsilon B_{11}^T & \varepsilon B_{21}^T \\ B_{12}^T & B_{22}^T \end{bmatrix} \tag{2.24}$$

This can be illustrated by the following example

Example 2.4 Consider Example 2.2 again, but this time compute the ORSF of $A^T(\varepsilon)$ to obtain

$$T = \begin{bmatrix} -5.01021 & 0.05682 & 1.41303 \\ 0 & -1.05184 & -0.01038 \\ 0 & 0 & -0.03795 \end{bmatrix} \text{ and } Q = \begin{bmatrix} 0.40581 & -0.68641 & 0.60345 \\ -0.81370 & 0.02932 & 0.58055 \\ 0.41619 & 0.72662 & 0.54663 \end{bmatrix}$$

All the conditions of Theorem 2.1 are satisfied with $P = I_3$. Now, choose E such that

$$U^TP = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ -1.0 & 1.0 & 0.0 \\ -1.0 & 0.0 & 1.0 \end{bmatrix}$$

The singularly perturbed system, (2.24) is given by

$$S^{-1}U^{T}AU^{-T}S = \begin{bmatrix} 0.0 & \varepsilon & 0.0 \\ 0.0 & -5.0 & 2.0 \\ -1.0 & 0.0 & -1-\varepsilon \end{bmatrix}$$

The eigenvalues of the slow and the fast subsystems are -0.0364 and $\{-1.1, -5\}$, respectively. Again, these eigenvalues are $O(\varepsilon^2)$ and $O(\varepsilon)$ close to the exact slow and fast eigenvalues of the system. In this example the slow variable is taken as V_1, while the fast variables are taken as $\frac{1}{\varepsilon}(V_2 - V_1)$ and $\frac{1}{\varepsilon}(V_3 - V_1)$.

2.5 Modeling by Permutation

When can a two-time-scale system be modeled in the singularly perturbed form by reordering its state variables? The following theorem gives necessary and sufficient conditions in terms of the ordered real Schur decomposition.

Theorem 2.3

Let A be a two-time-scale matrix, and (2.8) be an ordered real Schur decomposition. There exists a permutation matrix P such that PAP^T is in the singularly perturbed form if and only if

(i) T_{11} and T_{12} are $O(1)$

(ii) T_{22} is $O(\varepsilon)$

(iii) $PQ = W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$, where the $n \times m$ matrix W_{11} is $O(\varepsilon)$

proof:

Sufficiency: From the orthogonality of W we have

$$W_{21}^T W_{21} = I_m - W_{11}^T W_{11}$$

$$\underline{\sigma}^2(W_{21}) = 1 - \overline{\sigma}^2(W_{11}) \tag{2.25}$$

where $\underline{\sigma}(.)$ and $\overline{\sigma}(.)$ denote the minimum and maximum singular values respectively.

Since W_{11} is O(ε),

$$\bar{\sigma}^2(W_{11}) \leq K\,\varepsilon^2 \tag{2.26}$$

where K is independent of ε. Using (2.26) in (2.25) yields

$$\underline{\sigma}^2(W_{21}) \geq 1 - K\,\varepsilon^2$$

which implies that

$$\bar{\sigma}^2(W_{21}^{-1}) \leq \frac{1}{1 - K\,\varepsilon^2}$$

Hence, for sufficiently small ε, W_{21}^{-1} is O(1). Thus, all the conditions of Theorem 2.1 are satisfied and the system can be put in the singularly perturbed form via the transformation (2.16). But in the current case $W_{11}W_{21}^{-1}$ is O(ε). Therefore, choosing $\varepsilon E = -W_{11}W_{21}^{-1}$, the matrix U^{-1} becomes the identity matrix, and the transformation (2.16) reduces to a permutation.

Necessity: This follows as a special case of Theorem 2.2. Suppose there is a permutation matrix P such that PAP^T is in the singularly perturbed form, i.e.,

$$P\,A\,P^T = \begin{bmatrix} \varepsilon A_{11} & \varepsilon A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2.27}$$

Then, all the conditions of Theorem 2.2 are satisfied which implies conditions (i) and (ii). Moreover, the orthogonal matrix Q_0 in the proof of Theorem 2.2 can be chosen as $Q_0 = P^T \begin{bmatrix} 0 & I_n \\ I_m & 0 \end{bmatrix}$. Furthermore, the matrix W_0 in the same proof can be chosen as

$$W_0 = P\,Q_0 = \begin{bmatrix} 0 & I_n \\ I_m & 0 \end{bmatrix}$$

Due to this special form of W_0, W is given by

$$W = W_0\,L = \begin{bmatrix} L_{21} & L_{22} \\ L_{11} & L_{12} \end{bmatrix} \tag{2.28}$$

It was shown that L_{21} is O(ε); hence W_{11} is O(ε), which completes the proof of the theorem.

Example 2.5 : Consider the 10th-order system described by

$$A = \begin{bmatrix} -0.05 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.05 & -0.46 & 0.05 & 0 & 0.36 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & -0.46 & 0.05 & 0 & 0.36 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & -0.41 & 0 & 0 & 0.36 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & -0.25 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0.05 & -0.66 & 0.05 & 0.36 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0.05 & -0.61 & 0 & 0.36 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & -0.25 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.05 & -0.61 & 0.36 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & -0.2 \end{bmatrix}$$

The eigenvalues of this model are

$-1.0121, -0.9065, -0.6421, -0.6063, -0.3792, -0.2156, -0.1077, -0.0654, -0.02516$ and 0.

For 4 slow and 6 fast states, take $\varepsilon = 0.5$. Calculating the ORSF of this model yields

$$T_{11} = \begin{bmatrix} -1.0121 & 0.0136 & 0.0097 & 0.1550 & 0.0645 & 0.0081 \\ 0 & -0.9065 & 0.0169 & -0.0578 & 0.1569 & 0.0346 \\ 0 & 0 & -0.6421 & -0.0001 & -0.0117 & -0.0083 \\ 0 & 0 & 0 & -0.6063 & 0.0271 & -0.1686 \\ 0 & 0 & 0 & 0 & -0.3792 & 0.0073 \\ 0 & 0 & 0 & 0 & 0 & -0.2156 \end{bmatrix},$$

$$T_{12} = \begin{bmatrix} -0.0071 & -0.0054 & 0.0061 & 0.0399 \\ -0.0245 & -0.0383 & -0.0276 & 0.0021 \\ 0.1230 & -0.0174 & -0.0954 & 0.0314 \\ -0.0185 & -0.0248 & -0.0292 & -0.0659 \\ 0.0650 & 0.0946 & 0.0569 & -0.0344 \\ -0.0003 & -0.0099 & -0.0316 & -0.1021 \end{bmatrix}$$

and

$$T_{22} = \begin{bmatrix} -0.1077 & -0.0160 & -0.0051 & -0.0086 \\ 0 & -0.0654 & -0.0161 & -0.0118 \\ 0 & 0 & -0.0252 & -0.0123 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

This shows that T_{11} and T_{12} are O(1) and T_{22} is O(ε). In fact in this example T_{12} happens to be O(ε). Therefore conditions (i) and (ii) of Theorem 2.3 are satisfied. Condition (iii) is satisfied if the 10×6 matrix Q_1 has four rows which are O(ε). The $\|.\|_\infty$ of each row of Q_1 is given in the following table

Table 2.1 The $\|.\|_\infty$ of the ten rows of the matrix Q_1

row	1	2	3	4	5
$\|.\|_\infty$	0.0744	0.8782	0.7544	0.6460	0.4307
row	6	7	8	9	10
$\|.\|_\infty$	0.6358	0.6247	0.3746	0.6727	0.3332

which shows that rows 1, 5, 8 and 10 are O(ε). Permuting the rows of Q accordingly, we verify that

$$-W_{11}W_{21}^{-1} = \begin{bmatrix} 0.084 & 0.003 & 0.001 & -0.003 & 0.001 & 0.001 \\ 0.509 & 0.094 & -0.003 & 0.109 & 0.002 & 0.0 \\ 0.002 & 0.265 & 0.077 & 0.507 & 0.155 & 0.180 \\ 0 & -0.002 & 0.113 & -0.001 & 0.228 & 0.478 \end{bmatrix} = O(\varepsilon)$$

Thus the system can be modeled in the singularly perturbed form by regrouping the states 1, 5, 8 and 10 as the slow states and 2, 3, 4, 6, 7 and 9 as the fast states.

2.6 Summary

A reliable and numerically stable algorithm is developed to transform any two-time-scale system into a singularly perturbed form. This is accomplished in two steps: first, transform the matrix A into an ORSF; second, use the balancing algorithm developed in this chapter to balance the ORSF such that T_{11} and T_{12} are O(1) and T_{22} is O(ε).

If we are interested in the physical state variables of the system, sufficient conditions are derived to put the system in the singularly perturbed form whereas the fast or the slow physical states are retained. The conditions hold when the matrix A can be modeled in the form $A = A_0 + \varepsilon A_1$, where A_0 is singular with a semisimple null structure. The significance of these conditions over previous results in this area is that we do not require that the matrix A be modeled in the above form. Such modeling step may require a priori knowledge of the physical nature of the system. An additional assumption on the orthogonal transformation matrix Q yields necessary and sufficient conditions for a two-time-scale matrix to be modeled in the singularly perturbed form while retaining all the states as physical ones.

The modeling procedure for the special cases of Theorems 2.1 and 2.3 can be combined with the general modeling method of section 2.3 in one modeling procedure presented in the modeling flow chart of Fig. 2.2. According to this flow chart, we first compute the ORSF of the matrix A and check whether T_{11} and T_{12} are O(1) and T_{22} is O(ε). If they are not satisfied, then we use the general method to put the system in the singularly perturbed form. If they are satisfied, we check whether there are n rows of the matrix Q_1 which are of order O(ε). If this is the case, then we can obtain the singularly perturbed form by permutation; otherwise we check the existence of a permutation of the rows of the matrix Q such that W_{21}^{-1} is O(1). If this

is true, then a singularly perturbed form with fast variables chosen from the physical ones is achieved; otherwise, the general method is used.

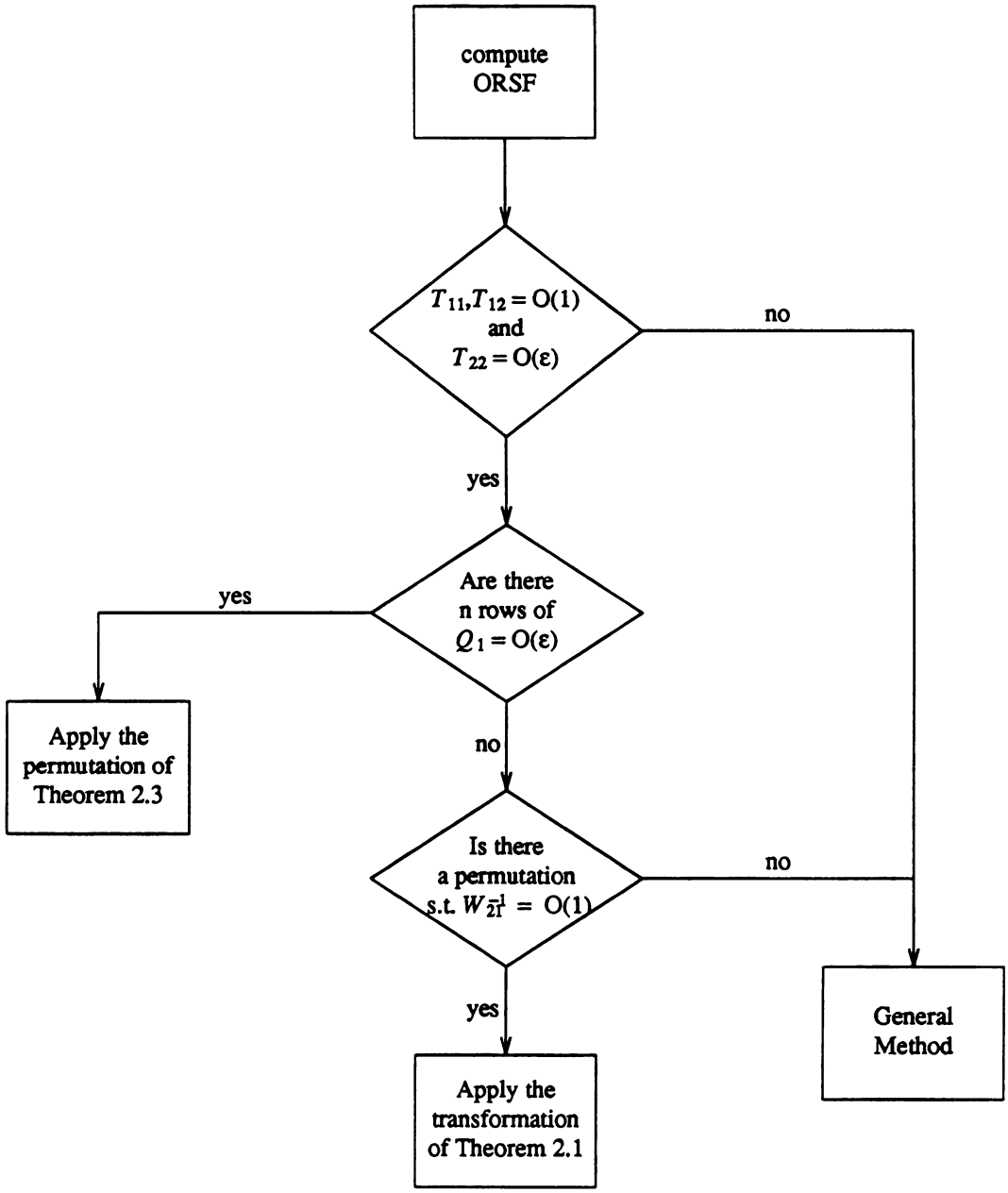

Fig. 2.2 Modeling flow chart

CHAPTER 3

EXACT AGGREGATION OF NEARLY-COMPLETELY DECOMPOSABLE MARKOV CHAINS (NCDMC)

3.1 Background

Finite Markov chains have many applications in biological, physical and social sciences, as well as in economics and engineering. Queueing networks which are used in computer system modeling are one of the areas which received a great deal of attention in the past few years [25,27 - 28].

A fundamental problem in these applications is to compute the unique stationary probability distribution vector which satisfies

$$\pi P = \pi, \quad \pi_i > 0, \quad \pi \mathbf{1}_n = 1 \tag{3.1}$$

where $P \in \mathbf{R}^{n \times n}$ is a transition probability matrix, $\mathbf{1}_n$ is a column vector consisting of n ones and π is the left eigenvector corresponding to the unit eigenvalue of the matrix P.

In many applications, the number of the states of a Markov chain model is so large that a direct solution method, such as an LU factorization, is not possible. Also, the number of iterations required by using an iterative method, such as the power method or the

Fortunately, in many large systems [24] the states can be clustered into a small number of groups such that there is a strong interaction within each group while the interaction among the groups is weak compared to the interaction within the groups. This class of systems is known in the literature as nearly completely decomposable Markov Chain.

The structure of this class of Markov chains can be represented by the following transition probability matrix

$$P = I_n + A + \varepsilon B = I_n + Q \tag{3.2}$$

where

$$Q = A + \varepsilon B\,, \tag{3.3}$$

$$A = \begin{bmatrix} A_1 & 0 & . & . & 0 \\ 0 & A_2 & 0 & . & . \\ 0 & 0 & .. & . & . \\ .. & .. & . & . & . \\ 0 & . & 0 & . & A_N \end{bmatrix} \tag{3.4}$$

and A_i is an $n_i \times n_i$ matrix, $i = 1, 2, ..., N$ and $\sum_{i=1}^{N} n_i = n$.

The matrices P and $(I_{n_i} + A_i)$, $i = 1, 2, ..., N$ are stochastic. Hence, the row sums of B and A_i are zero. It is assumed that the Markov chain has a single ergodic class, i.e., all states communicate with each other. In this case the Markov chain and the matrix P are called irreducible. This assumption is equivalent to the assumption that equation (3.1) has a unique solution, The probabilities π_i for all i are positive [35, pp. 181-182], and the vector π is called the Perron-Frobenius eigenvector, or simply, P-F eigenvector. Furthermore, the matrix P has a simple unity eigenvalue, and the matrix Q has a unique zero eigenvalue. It is also assumed that each of the matrices $(I_{n_i} + A_i)$ is irreducible. Hence A_i has a unique zero eigenvalue.

The maximum degree of coupling ε is very small and is defined by

$$\varepsilon = \|P - I_n - A\|_\infty << 1 \tag{3.5}$$

The problem of interest is to solve (3.1) or, equivalently, to solve

$$\pi Q = 0, \quad \pi \mathbf{1}_n = 1 \tag{3.6}$$

Simon and Ando [24] were the first to propose this class of systems which is known as nearly-completely decomposable systems. They gave examples from economics to illustrate this class of systems. Their analysis is based on the following argument:

1. Over a relatively short period of time a local equilibrium is reached among each group separately and in this period we can analyze the system as if there was no interaction among the groups.
2. In the long run, the interaction among groups can not be neglected and the whole system moves towards a global equilibrium which is defined by the steady-state probability. The equilibrium maintained in the short run will remain approximately the same because the interaction among the groups will not influence the relations between the states within each group.

The approximate Perron-Frobenius eigenvector was obtained by approximate aggregation as it will be shown in this section. Courtois [25] developed the theory of the NCDMC and applied his method to several computer system models [25 – 27]. According to his method the approximate P-F eigenvector is computed as follow:

i. Compute the P-F eigenvector of each block, i.e.,

$$v_i A_i = 0, \quad v_i \mathbf{1}_{n_i} = 1, \text{ for all } i = 1,2,\ldots\ldots, N \tag{3.7}$$

ii. Form the approximate aggregated matrix P_1 as

$$P_1 = I_N + \varepsilon \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & . & . \\ . & 0 & . & 0 \\ 0 & . & . & v_N \end{bmatrix} B \begin{bmatrix} \mathbf{1}_{n_1} & 0 & 0 & 0 \\ 0 & \mathbf{1}_{n_2} & . & . \\ . & 0 & . & 0 \\ 0 & . & . & \mathbf{1}_{n_N} \end{bmatrix} = I_N + \varepsilon \hat{V}_1 B W_1 = I_N + \varepsilon Q_0 \tag{3.8}$$

where $\hat{V}_1 \in \mathbf{R}^{N \times n}$ and $W_1 \in \mathbf{R}^{n \times N}$ are defined by

$$\hat{V}_1 = \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & . & . \\ . & 0 & . & 0 \\ 0 & . & . & v_N \end{bmatrix} \quad \text{and} \quad W_1 = \begin{bmatrix} \mathbf{1}_{n_1} & 0 & 0 & 0 \\ 0 & \mathbf{1}_{n_2} & . & . \\ . & 0 & . & 0 \\ 0 & . & . & \mathbf{1}_{n_N} \end{bmatrix} \tag{3.9}$$

iii. Compute the P-F eigenvector of P_1 that satisfies

$$X P_1 = X, \; X \mathbf{1}_N = 1$$

or

$$X Q_0 = 0, \; X \mathbf{1}_N = 1 \tag{3.10}$$

iv. The approximate P-F eigenvector of the whole system is

$$\tilde{\pi} = X \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & . & . \\ . & 0 & . & 0 \\ 0 & . & . & v_N \end{bmatrix} \tag{3.11}$$

Courtois [26] showed that $\tilde{\pi}$ is an $O(\varepsilon)$ approximation of π.

Vantilborgh [29] carried this approximation one more step to come up with an $O(\varepsilon^2)$ approximation of the P-F eigenvector π. He started by partitioning $Q = A + \varepsilon B$ as

$$Q = \begin{bmatrix} \hat{Q} & \hat{Q}_C \\ \hat{Q}_L & \hat{Q}_N \end{bmatrix}$$

where $\hat{Q} \in \mathbf{R}^{(n-n_N) \times (n-n_N)}$ and $\hat{Q}_N \in \mathbf{R}^{n_N \times n_N}$

He then formed a matrix

$$Q_N^* = \hat{Q}_N - \hat{Q}_L \, \hat{Q}^{-1} \, \hat{Q}_C \tag{3.12}$$

This partition is not based on decomposing the zero and nonzero eigenvalues. So, he was confronted with a near singular matrix $\hat{Q}$. To overcome this problem, he made use of the NCD structure of $\hat{Q}$ to expand it in ε. Thus he obtained

$$\hat{Q}^{-1} \approx \begin{bmatrix} \mathbf{1}_{n_1} & 0 & . & 0 \\ 0 & \mathbf{1}_{n_2} & . & . \\ . & 0 & . & . \\ . & . & . & \mathbf{1}_{n_{N-1}} \end{bmatrix} \hat{Q}_0^{-1} \begin{bmatrix} v_1 & 0 & . & 0 \\ 0 & v_2 & . & . \\ . & 0 & . & . \\ . & . & . & v_{N-1} \end{bmatrix} \tag{3.13}$$

where $\hat{Q}_0$ is obtained by partitioning the matrix Q_0 of equation (3.10) as

$$Q_0 = \begin{bmatrix} \hat{Q}_0 & \hat{Q}_{0c} \\ \hat{Q}_{0l} & q_{0n} \end{bmatrix} \tag{3.14}$$

where q_{0n} is 1×1 and $\hat{Q}_0 \in \mathbf{R}^{N-1 \times N-1}$.

Again $\hat{Q}_0$ is a near singular matrix; therefore, another expansion in ε should be pursued. Hence, equation (3.12) becomes

$$Q_N^* = \hat{Q}_N - \hat{Q}_L \begin{bmatrix} \mathbf{1}_{n_1} & 0 & . & 0 \\ 0 & \mathbf{1}_{n_2} & . & . \\ . & 0 & . & . \\ . & . & . & \mathbf{1}_{n_{N-1}} \end{bmatrix} \hat{Q}_0^{-1} \begin{bmatrix} v_1 & 0 & . & 0 \\ 0 & v_2 & . & . \\ . & 0 & . & . \\ . & . & . & v_{N-1} \end{bmatrix} \hat{Q}_C = \hat{Q}_N - q\, \hat{Q}_0^{-1}\, p \tag{3.15}$$

where $q \in \mathbf{R}^{n_N \times (N-1)}$ and $p \in \mathbf{R}^{(N-1) \times n_N}$ are given by

$$q = \hat{Q}_L \begin{bmatrix} \mathbf{1}_{n_1} & 0 & . & 0 \\ 0 & \mathbf{1}_{n_2} & . & . \\ . & 0 & . & . \\ . & . & . & \mathbf{1}_{n_{N-1}} \end{bmatrix} \quad \text{and} \quad p = \begin{bmatrix} v_1 & 0 & . & 0 \\ 0 & v_2 & . & . \\ . & 0 & . & . \\ . & . & . & v_{N-1} \end{bmatrix} \hat{Q}_C \tag{3.16}$$

The dominant part of $\hat{Q}_0^{-1}$ is given by [29]

$$\mathbf{1}_{N-1}(\hat{X}^* p\ \mathbf{1}_{n_N})^{-1}\hat{X}^* \tag{3.17}$$

where

$$\hat{X}^* = \frac{1}{1-X_N}\sum_{i=1}^{N-1} X_i \tag{3.18}$$

and X is defined by (3.10). Substitute (3.17) and (3.18) into (3.16) to obtain

$$Q_N^* = \hat{Q}_N - \frac{1}{\hat{X}^* p\ \mathbf{1}_{n_N}} q\ \mathbf{1}_{N-1}\ \hat{X}^* p + O(\varepsilon^2) \tag{3.19}$$

In general, the matrices Q_I^* for $I = 1,\ldots\ldots, N$ are defined by

$$Q_I^* = \hat{Q}_I - \frac{1}{\hat{X}^* p\ \mathbf{1}_{n_I}} q\ \mathbf{1}_{N-1}\ \hat{X}^* p + O(\varepsilon^2) \tag{3.20}$$

where q, p and $\hat{X}^*$ are defined by the same procedure as in block N, i.e., by (3.16) to (3.18).

Now, this method can be summarized as follow :

1. Compute v_i for all i, Q_0 and X by using (3.7), (3.8) and (3.10)
2. Compute q and p that correspond to the Ith block as defined in (3.16)
3. Form the matrix Q_I^*, $I=1,\ldots\ldots, N$ as in (3.20)
4. Repeat steps i – iv of Courtois' O(ε) approximation method with the new matrices Q_I^* replacing A_i to obtain an $O(\varepsilon^2)$ approximation of π.

This method requires more than double the computations needed for the O(ε) approximation.

Phillips and Kokotovic [11] gave a singular perturbation interpretation to Courtois' aggregation of NCD systems. They defined the continuous time Markov process by

$$\frac{d\,\pi(t)}{dt} = \pi(t)\,(\,\frac{A}{\varepsilon} + B)\,, \quad \pi(t)\,\mathbf{1}_n = 1 \tag{3.21}$$

where $\pi(t)$ is n-dimensional probability distribution vector at time t.

Consider the similarity transformation

$$\begin{bmatrix}\xi & \eta\end{bmatrix} = \pi\begin{bmatrix}W_1 & \hat{W}_2\end{bmatrix} \text{ and } \pi = \begin{bmatrix}\xi & \eta\end{bmatrix}\begin{bmatrix}\hat{V}_1 \\ \hat{V}_2\end{bmatrix} \tag{3.22}$$

where the matrices $\hat{V}_1$ and W_1 are chosen as in (3.9); hence

$$\hat{V}_1\,A = 0\,, \quad A\,W_1 = 0 \text{ and } \hat{V}_1 W_1 = I_N\,, \tag{3.23}$$

while the matrices $\hat{V}_2$ and $\hat{W}_2$ are chosen to satisfy

$$\hat{V}_2 W_1 = 0\,, \quad \hat{V}_1\hat{W}_2 = 0 \qquad \hat{V}_2\hat{W}_2 = I_{n-N}$$

which ensures that

$$\begin{bmatrix}W_1 & \hat{W}_2\end{bmatrix}\begin{bmatrix}\hat{V}_1 \\ \hat{V}_2\end{bmatrix} = I_n$$

Substituting (3.22) into (3.21) gives

$$\frac{d\,\xi(t)}{dt} = \xi(t)\,\hat{V}_1\,B\,W_1 + \eta(t)\,\hat{V}_2\,B\,W_1 \tag{3.24}$$

$$\frac{d\,\eta(t)}{dt} = \varepsilon\,\xi(t)\hat{V}_1 B\hat{W}_2 + \eta(t)\,\hat{V}_2(A + \varepsilon\,B)\,\hat{W}_2\,. \tag{3.25}$$

Equations (3.24) and (3.25) are in the singularly perturbed form. The O(ε) approximate stationary probability distribution vector is obtained, as $t \to \infty$, by setting $\varepsilon = 0$. This yields

$$\xi_0\,(\hat{V}_1\,B\;\;W_1) = 0 \text{ and } \quad \eta_0 = 0 \tag{3.26}$$

and the approximate π is given by substituting ξ_0 and η_0 in (3.22). Therefore,

$$\pi = \xi_0\,\hat{V}_1 + O(\varepsilon)\,. \tag{3.27}$$

Equations (3.26) and (3.27) are equivalent to (3.10) and (3.11), respectively. Notice that this transformation is similar to the one discussed in chapter 2 where A has a semisimple null structure, $\hat{V}_1$ spans the left null space of A and $\hat{V}_2$ is perpendicular to the null-space of A. This is the reason for having a weak coupling between η and ξ in equation (3.25). As mentioned in chapter 2, this transformation is a special case. There is a wide class of transformations which can put the system in the singularly perturbed form, but the coupling between the $\eta(t)$ and $\xi(t)$ will not be necessarily weak.

In section 3.2, we will propose a more general transformation to put the system in the singularly perturbed form and will show later in section 3.6 that the transformation proposed in [11] is a subclass of this transformation.

3.2 Two-time Scale Decomposition of NCDMC

The nearly-completely decomposable Markov chain matrix Q is very ill conditioned because it has one zero eigenvalue and $N-1$ eigenvalues which are very close to zero. Therefore, a direct method to solve the steady-state problem will suffer from ill-conditioning. However, this matrix exhibits a two-time scale property which can be used to decompose the system into slow and fast parts and, as a result of this, the high order system is reduced to a low order one.

The transformation proposed in this section is more general than the transformation considered in [11]. This transformation does not put any restriction on choosing the matrix V_1. The transformation can be described as follows:

W_1 is chosen as in (3.9); hence $A\,W_1 = 0$; W_2 can be any matrix such that the transformation

$$\Gamma = \begin{bmatrix} W_1 & W_2 \end{bmatrix} \tag{3.28}$$

is nonsingular. The matrix Γ^{-1} is written as

$$\Gamma^{-1} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \tag{3.29}$$

Therefore, $V_1 W_1 = I_N$, $V_2 W_2 = I_{n-N}$, $V_1 W_2 = 0$ and $V_2 W_1 = 0$. The dimensions of V_1, V_2, W_1 and W_2 are $N \times n$, $(n - N) \times n$, $n \times N$ and $n \times (n - N)$, respectively.

Since we are interested in the stationary probability distribution, we apply this transformation to equation (3.1). Let $\xi = \pi W_1$ and $\eta = \pi W_2$ then, equation (3.1) becomes

$$\pi \begin{bmatrix} W_1 & W_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} P = \pi . \tag{3.30}$$

Multiply (3.30) from the right by Γ to get

$$\begin{bmatrix} \xi & \eta \end{bmatrix} \begin{bmatrix} I_N + \varepsilon V_1 B W_1 & V_1(A + \varepsilon B) W_2 \\ \varepsilon V_2 B W_1 & I_{n-N} + V_2(A + \varepsilon B) W_2 \end{bmatrix} = \begin{bmatrix} \xi & \eta \end{bmatrix} . \tag{3.31}$$

Because A has N zero eigenvalues and

$$\Gamma^{-1} A \Gamma = \begin{bmatrix} 0 & V_1 A W_2 \\ 0 & V_2 A W_2 \end{bmatrix} \tag{3.32}$$

it is seen that $V_2 A W_2$ is nonsingular. This implies that for ε sufficiently small, $V_2(A + \varepsilon B) W_2$ is nonsingular. From equation (3.31) we have

$$\xi (I_N + \varepsilon V_1 B W_1) + \varepsilon \eta V_2 B W_1 = \xi \tag{3.33}$$

and

$$\xi V_1(A + \varepsilon B) W_2 + \eta(I_{n-N} + V_2(A + \varepsilon B) W_2) = \eta . \tag{3.34}$$

Therefore,

$$\eta = - \xi V_1(A + \varepsilon B) W_2 \left[V_2(A + \varepsilon B) W_2 \right]^{-1} . \tag{3.35}$$

Substitution of (3.35) into (3.33) yields

$$\xi\left[I_N + \varepsilon\left[V_1B\ W_1 - \ V_1(A + \varepsilon B)\ W_2\left[V_2(A + \varepsilon B)\ W_2\right]^{-1} V_2B\ W_1\right]\right] = \xi \qquad (3.36)$$

Let

$$Q_s = \left[V_1B\ W_1 - \ V_1(A + \varepsilon B)\ W_2\left[V_2(A + \varepsilon B)\ W_2\right]^{-1} V_2B\ W_1\right] \qquad (3.37)$$

and

$$P_s = I_N + \varepsilon\ Q_s\ . \qquad (3.38)$$

Then

$$\xi\ P_s = \xi \qquad (3.39)$$

or simply

$$\xi\ Q_s = 0\ . \qquad (3.40)$$

Again from (3.6), $\pi\ \mathbf{1}_n = 1$, therefore,

$$\pi\left[W_1 \quad W_2\right]\begin{bmatrix} V_1 \\ V_2 \end{bmatrix}\mathbf{1}_n = 1\ . \qquad (3.41)$$

Noting that $\mathbf{1}_n = W_1\ \mathbf{1}_N$, (3.41) can be written as

$$\left[\xi \quad \eta\right]\begin{bmatrix} I_N \\ 0 \end{bmatrix}\mathbf{1}_N = 1\ .$$

Hence

$$\xi\ \mathbf{1}_N = 1\ . \qquad (3.42)$$

Thus, the solution of the full order system (3.6) is completely characterized by

$$\pi = \xi\ V_1 + \eta\ V_2 \qquad (3.43)$$

where ξ is defined by (3.40) and (3.42), and η is defined by (3.35)

The existence of a unique solution of (3.40) and (3.42) is discussed in section 3.3. In sections 3.4 and 3.5, first and higher order approximations of the solution of equations (3.40), (3.42) are given, respectively.

3.3 Properties of the Exact Aggregated System

The aggregated matrix Q_s is a reduced form of the original matrix Q and it inherits some of its properties. In this section some properties of Q_s are given in the form of lemmas or theorems.

Lemma 3.1

The row sums of the matrix Q_s are equal to zero.

Proof :

From (3.37), Q_s can be rewritten as

$$Q_s = \left[V_1 - V_1(A + \varepsilon B) W_2 \left[V_2(A + \varepsilon B) W_2 \right]^{-1} V_2 \right] B W_1 = V_s B W_1 \qquad (3.44)$$

where

$$V_s = \left[V_1 - V_1(A + \varepsilon B) W_2 \left[V_2(A + \varepsilon B) W_2 \right]^{-1} V_2 \right] . \qquad (3.45)$$

Recall from section 3.1, that each row sum of B is equal to zero.

Let the ith row sum of $B W_1 = C_i$, then

$$C_i = \sum_{j=1}^{n_1} b_{ij} + \sum_{j=n_1+1}^{n_1+n_2} b_{ij} + \ldots\ldots\ldots\ldots + \sum_{j=n-n_N+1}^{n} b_{ij} = \sum_{j=1}^{n} b_{ij} = 0 . \qquad (3.46)$$

Thus, the ith row sum of $B W_1 = 0$. Now we need to prove that the ith row sum of $V_s B W_1 = 0$. Denote the ith row of V_s by V_{si} and the jth column of $B W_1$ by C^j. Then the ith row sum of $V_s B W_1$ is

$$\sum_{j=1}^{N} V_{si}\, C^j = V_{si} \sum_{j=1}^{N} C^j = 0 \tag{3.47}$$

and this completes the proof.

As a result of this lemma, the matrix Q_s has a zero eigenvalue. Another useful property of Q_s which will be used in this chapter and in chapter 4 is given in the following lemma.

Lemma 3.2

Let the matrix Q_s be defined as in (3.37) and Ψ_{si} be defined as the matrix Q_s with its ith column replaced by the vector $\mathbf{1}_N$. Then

i. The matrix Q_s has a unique zero eigenvalue.

ii. The matrix Ψ_{si} is nonsingular, for $i = 1, 2, ..., N$;

iii. Equations (3.40) and (3.42) have a unique solution and $\xi_i > 0$, $i = 1, 2, ..., N$.

Proof :

i. From the ergodicity assumption, the matrix Q has a unique zero eigenvalue and equation (3.6) has a unique solution; therefore,

$$Rank(\, Q \,,\ \mathbf{1}_n \,) = n \,. \tag{3.48}$$

The rank of the matrix is invariant to pre and post multiplication by nonsingular matrices; therefore,

$$Rank(\, Q \,,\ \mathbf{1}_n \,) = Rank\left[\Gamma^{-1}(Q \,,\ \mathbf{1}_n) \begin{bmatrix} \Gamma & 0 \\ 0 & 1 \end{bmatrix} \right] = Rank\left[\Gamma^{-1} Q\, \Gamma \,,\ \Gamma^{-1}\, \mathbf{1}_n \right]$$

$$= Rank\left[\begin{bmatrix} \varepsilon\ V_1 B\, W_1 & V_1(A + \varepsilon\, B)\, W_2 & \mathbf{1}_N \\ \varepsilon\ V_2 B\, W_1 & V_2(A + \varepsilon\, B)\, W_2 & 0 \end{bmatrix} \right] = n. \tag{3.49}$$

Since $V_2(A + \varepsilon B)\, W_2$ is nonsingular, an elementary row operation brings the above matrix into the form

$$\begin{bmatrix} \varepsilon\, Q_s & 0 & \mathbf{1}_N \\ \varepsilon\ V_2B\ W_1 & V_2(A+\varepsilon B)\ W_2 & 0 \end{bmatrix} \tag{3.50}$$

where the elementary row operation matrix is

$$\begin{bmatrix} I_N & -\ V_1(A+\varepsilon\, B)\ W_2(\ V_2(A+\varepsilon\, B)\ \ W_2)^{-1} \\ 0 & I_{n-N} \end{bmatrix}.$$

Hence

$$Rank(\ Q\ ,\ \mathbf{1}_n\ ) = Rank\begin{bmatrix} \varepsilon\, Q_s & 0 & \mathbf{1}_N \\ \varepsilon\ V_2B\ W_1 & V_2(A+\varepsilon B)\ W_2 & 0 \end{bmatrix}$$

$$= n - N + Rank(\ Q_s\ ,\ \mathbf{1}_N\ ) = n.$$

Therefore,

$$Rank\ (\ Q_s\ ,\ \mathbf{1}_N\ ) = N\ . \tag{3.51}$$

Thus, $Rank\ (\ Q_s\ ) = N - 1$. Moreover, since $\mathbf{1}_N$ spans the null space of the matrix Q_s, (3.51) implies

$$R(Q_s) \oplus N(Q_s) \in \mathbf{R}^N\ . \tag{3.52}$$

Hence, Q_s has a semi-simple null structure. This implies that Q_s has a unique zero eigenvalue.

ii. To prove that Ψ_{si} is nonsingular, let us prove first that any $N-1$ columns of Q_s are linearly independent. Any column of Q_s can be replaced by a zero column via an elementary column operation (ECO), which can be represented by

$$Q_s\ E_i = \left[q_s^1,\ q_s^2,\ldots\ldots,\ q_s^{i-1},\ 0\ ,\ q_s^{i+1},\ldots\ldots,q_s^N\right] \tag{3.53}$$

where q_s^i is the ith column of the matrix Q_s and $E_i \in \mathbf{R}^{N \times N}$ is a matrix which can be obtained by performing the same ECO on I_N. It takes one of the following forms:

$$E_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & . \\ . & 0 & 1 & . & . \\ . & . & . & . & . \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & . \\ . & 1 & 1 & . & . \\ . & . & . & . & . \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad E_N = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ . & 0 & 1 & . & . \\ . & . & . & . & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.54}$$

Since E_i is nonsingular,

$$Rank(\, Q_s \, E_i \,) = Rank(\, Q_s \,) = N - 1 \, . \tag{3.55}$$

Hence, any $N - 1$ columns of Q_s are linearly independent.

Now, the range space of Q_s is spanned by any $N - 1$ columns of Q_s and the vector $\mathbf{1}_N$ spans the null-space of Q_s. Using (3.52), we conclude that Ψ_{si} is a nonsingular matrix.

iii. The proof of this part follows from parts (i) and (ii). Since equations (3.40), (3.42) can be rewritten as

$$\xi \left[q_s^1, q_s^2, \ldots\ldots, q_s^{i-1}, \mathbf{1}_N, q_s^{i+1}, \ldots\ldots, q_s^N \right] = \left[0, 0, \ldots\ldots, 0, 1, 0, \ldots\ldots, 0 \right]$$

or

$$\xi \, \Psi_{si} = e_i$$

The nonsingularity of Ψ_{si} implies that the equation has a unique solution. Moreover, the relationship

$$\xi = \pi \; W_1$$

implies that $\xi_i > 0$, since $\pi_i > 0$.

Since Q_s preserves some of the properties of Q, like zero row sums, the uniqueness of the zero eigenvalue and the property that the left eigenvector corresponding to the unit eigenvalue is positive and its sum equals 1, a natural question that comes to mind is whether or not $P_s = I_N + \varepsilon\, Q_s$ is a stochastic matrix. The answer is no, in general.

The following example is a counter example that shows that the matrix P_s may not be a stochastic

Example 3.1: Consider the irreducible transition probability matrix

$$P = \begin{bmatrix} .5-\varepsilon & .5-\varepsilon & 0 & 0 & \varepsilon & \varepsilon \\ .5 & .5-\varepsilon & 0 & 0 & 0 & \varepsilon \\ \varepsilon & \varepsilon & .5-\varepsilon & .5-\varepsilon & 0 & 0 \\ 0 & \varepsilon & .5 & .5-\varepsilon & 0 & 0 \\ .5\varepsilon & .5\varepsilon & .5\varepsilon & .5\varepsilon & .5-\varepsilon & .5-\varepsilon \\ 0 & .5\varepsilon & 0 & .5\varepsilon & .5 & .5-\varepsilon \end{bmatrix}$$

From (3.2) to (3.4) the matrices A and B are

$$A = \begin{bmatrix} -.5 & .5 & 0 & 0 & 0 & 0 \\ .5 & -.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & -.5 & .5 & 0 & 0 \\ 0 & 0 & .5 & -.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -.5 & .5 \\ 0 & 0 & 0 & 0 & .5 & -.5 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -1 & -1 & 0 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ .5 & .5 & .5 & .5 & -1 & -1 \\ 0 & .5 & 0 & .5 & 0 & -1 \end{bmatrix}$$

Let Γ be equal to

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

From (3.37)

$$Q_s = \begin{bmatrix} -1.5 - 1.5\varepsilon & -.5\varepsilon & 1.5 + 2\varepsilon \\ 1.5 + 2\varepsilon & -1.5 - \varepsilon & -\varepsilon \\ 0.75 + .5\varepsilon & 0.75 + \varepsilon & -1.5 - 1.5\varepsilon \end{bmatrix}$$

It is clear that the matrix $P_s = I_N + \varepsilon Q_s$ is not a stochastic matrix because of the negative sign of the (12) and (23) elements. Notice, however, that, the matrix $P_1 = I_N + \varepsilon Q_0$ is stochastic.

3.4 First Order Approximation of the Aggregated System

An O(ε) approximation of the Perron-Frobenius eigenvector can be obtained by setting $\varepsilon = 0$ in equation (3.44) to obtain

$$Q_0 = Q_s\big|_{\varepsilon = 0} = \left[V_1 - V_1 A W_2 (V_2 A W_2)^{-1} V_2 \right] B W_1 = V_0 B W_1 \qquad (3.56)$$

where

$$V_0 = \left[V_1 - V_1 A W_2 (V_2 A W_2)^{-1} V_2 \right] \qquad (3.57)$$

and equation (3.38) becomes

$$P_1 = I_N + \varepsilon Q_0 \qquad (3.58)$$

This approximation reduces the amount of computations involved in computing the matrix V_s. In this section the properties of Q_0, V_0 and P_1 are presented and a closed form expression for the P-F eigenvector of the stochastic matrix is given.

Theorem 3.1

Let A be defined as in equation (3.4) and V_0 as in (3.57). Then

i. $V_0 W_1 = I_N$ (3.59)

ii. $V_0 A W_2 = 0$ (3.60)

iii. $V_0 A = 0$ (3.61)

iv. V_0 is a block (row) diagonal matrix and it is nonnegative.

v. The ith diagonal block of the matrix V_0 is the P-F eigenvector of the matrix $(A_i + I_{n_i})$.

Proof :

Parts (i) and (ii) follow by multiplying equation (3.57) from the right by W_1 and $A\,W_2$ respectively.

iii. The matrix Γ of (3.28) is nonsingular, therefore, the matrices A and $A\,\Gamma$ have the same range space. But,

$$A\,\Gamma = A\begin{bmatrix} W_1 & W_2 \end{bmatrix} = \begin{bmatrix} 0 & A\,W_2 \end{bmatrix}. \tag{3.62}$$

Thus, the range space of A is spanned by the $(n - N)$ columns of the matrix $A\,W_2$. Therefore, $V_0 A\,W_2 = 0$ implies that $V_0 A = 0$.

iv. To prove this part, let

$$V_0 = \begin{bmatrix} v_{11} & v_{12} & . & . & v_{1N} \\ v_{21} & v_{22} & . & . & v_{2N} \\ . & . & . & . & \\ . & . & . & . & \\ v_{N1} & v_{N2} & . & . & v_{NN} \end{bmatrix} \tag{3.63}$$

where v_{ij}, for all i and j, are vectors matched with the partitions of the matrix A.

From (i) and (iii)

$$v_{ii}\,A_i = 0\,,\quad v_{ii}\,\mathbf{1}_{n_i} = 1 \quad \text{for} \quad i = 1,2,.....,N \tag{3.64}$$

By (3.64),

$$v_{ii}\,(A_i + I_{n_i}) = v_{ii} \quad \text{and} \quad v_{ii}\,\mathbf{1}_{n_i} = 1\,, i = 1,....., N\,. \tag{3.65}$$

But $(A_i + I_{n_i})$ is a stochastic matrix with a single ergodic class. Therefore, the vector

v_{ii} is the P-F eigenvector which is unique and positive.

Now, to prove that the matrix V_0 is block diagonal, we should prove that $v_{ij} A_j = 0$ for $i \neq j$ implies that $v_{ij} = 0$.

From (iii), $v_{ij} A_j = 0, \quad i \neq j$, or

$$v_{ij} (A_j + I_{n_j}) = v_{ij} . \tag{3.66}$$

Since $(A_j + I_{n_j})$ is a stochastic matrix with a single ergodic class, v_{ij} must be equal to $K v_{jj}$, where K is any constant. From part (i), $v_{jj} \mathbf{1}_{n_j} = 1$ and $v_{ij} \mathbf{1}_{n_j} = 0$. This shows that

$$v_{ij} \mathbf{1}_{n_j} = K v_{jj} \mathbf{1}_{n_j} = 0 , \tag{3.67}$$

which implies that $K = 0$. Hence, $v_{ij} = 0$ for all $i \neq j$.

This completes the proof of (iv).

v. This part follows from (i), (iii) and (iv)

The last part of the above theorem shows that $V_0 = \hat{V}_1$, where $\hat{V}_1$ is given by (3.9). Hence, the matrix P_1 of (3.58) is the same as P_1 of (3.8). Therefore, P_1 is independent of the choice of W_2. In other words, all transformations of the form (3.28) produce the same matrix P_1. Furthermore, the expression (3.57) for V_0 gives a closed form expression for the P-F eigenvectors of the stochastic matrices $(I_{n_i} + A_i)$. Such closed form expression is important especially when the stochastic matrix is a function of a certain parameter as in the controlled Markov chain problem of Chapter 4.

Theorem 3.2

The matrix P_1 of (3.58) is a stochastic matrix with a single ergodic class.

Proof :

From (3.59) and (3.61), $V_0 \ W_1 = I_N$ and $V_0 A = 0$. Therefore $P_1 = I_N + \varepsilon Q_0$ can be written as

$$P_1 = V_0 (I_N + A + \varepsilon B) W_1 = V_0 P W_1 \tag{3.68}$$

where P is defined by (3.1). Similar to the proof of lemma 3.1, it can be shown that the row sums of $V_0 P W_1$ are equal to one. Moreover, since V_0, P and W_1 are nonnegative matrices, P_1 is also nonnegative. Hence, P_1 is nonnegative and its row sums are equal to 1, which implies that P_1 is a stochastic matrix. The ergodicity follows from the ergodicity of P; see, for example, [25].

As a result of this theorem, the matrix P_1 has a unique left eigenvector corresponding to the unit eigenvalue, i. e.,

$$\xi_0 P_1 = \xi_0 , \quad \xi_0 \mathbf{1}_N = 1 \tag{3.69}$$

or simply,

$$\xi_0 Q_0 = 0 \tag{3.70}$$

Theorem 3.3

i. P_1 is an $O(\varepsilon^2)$ approximation of P_s

ii. ξ_0 is an $O(\varepsilon)$ approximation of ξ

iii. π_0, given by

$$\pi_0 = \xi_0 V_1 + \eta_0 V_2 = \xi V_1 - \xi_0 V_1 A W_2 (V_2 A W_2)^{-1} V_2 = \xi_0 V_0 \tag{3.71}$$

is an $O(\varepsilon)$ approximation of π.

Proof :

i. From (3.38) and (3.44)

$$P_s = I_N + \varepsilon V_s B W_1 .$$

Expand V_s in ε, to obtain

$$P_s = I_N + \varepsilon (V_0 + O(\varepsilon)) B W_1 = P_1 + O(\varepsilon^2)$$

ii. From lemma 3.2

$$\xi \, \Psi_{si} = e_i \tag{3.72}$$

where e_i is an N-dimensional row vector with the ith element equal to 1 and the rest of the elements equal to zero, i.e.,

$$\xi \left[q_s^1, q_s^2, \ldots\ldots, q_s^{i-1}, \mathbf{1}_N, q_s^{i+1}, \ldots\ldots, q_s^N \right] = \left[0, 0, \ldots\ldots, 0, 1, 0, \ldots\ldots, 0 \right]. \tag{3.73}$$

Similarly,

$$\xi_0 \, \Psi_{0i} = e_i \tag{3.74}$$

$$\xi_0 \left[q_0^1, q_0^2, \ldots\ldots, q_0^{i-1}, \mathbf{1}_N, q_0^{i+1}, \ldots\ldots, q_0^N \right] = \left[0, 0, \ldots\ldots, 0, 1, 0, \ldots\ldots, 0 \right] \tag{3.75}$$

where q_0^i is the ith column of the matrix Q_0 and the right hand vector is the same as the one defined in (3.72). As in Lemma 3.2, the matrices Ψ_{si} and Ψ_{0i} are nonsingular. If Q_s is expanded in ε, we obtain

$$Q_s = Q_0 + O(\varepsilon) \; ; \; \text{therefore,} \; \Psi_{si} = \Psi_{0i} + O(\varepsilon) . \tag{3.76}$$

From equations (3.72) — (3.76)

$$(\xi - \xi_0) \, \Psi_{0i} = O(\varepsilon). \tag{3.77}$$

Since Ψ_{0i} is nonsingular,

$$\xi = \xi_0 + O(\varepsilon) . \tag{3.78}$$

iii. Substituting (3.35) in (3.43) and using (3.45), we obtain

$$\pi = \xi V_s = (\xi_0 + O(\varepsilon)) (V_0 + O(\varepsilon))$$

$$= \xi_0 \, V_0 + O(\varepsilon) = \pi_0 + O(\varepsilon). \tag{3.79}$$

3.5 Higher Order Approximations of the Perron-Frobenius Eigenvector of NCD Systems

In some of the applications, an $O(\varepsilon)$ approximation may not be good enough to meet certain specifications; so, a higher order approximation is needed. In [26] and [29], a procedure was given to get an $O(\varepsilon^2)$ approximation of the steady state probability. In this section we develop a straight forward procedure to obtain an $O(\varepsilon^k)$ approximation. The only computational effort we need in this procedure is some additional matrix multiplications which can be done easily in the presence of today's VLSI technology by using, for example, an array processor.

It is shown in the appendix that V_s can be written as

$$V_s = V_0 - \varepsilon\, V_0\, B\, W_2(\, V_2(A + \varepsilon\, B)\, W_2)^{-1}\ \ V_2\,. \tag{3.80}$$

Substituting (3.80) into (3.44), we get

$$Q_s = V_0\, B\, W_1 - \varepsilon\, V_0\, B\, W_2(\, V_2(A + \varepsilon\, B)\, W_2)^{-1}\, V_2 B\, W_1\,. \tag{3.81}$$

If we expand $(\, V_2(A + \varepsilon\, B)\, W_2)^{-1}$ in ε, we obtain

$$(\ V_2(A + \varepsilon\, B)\, W_2)^{-1} = (\, V_2 A\, W_2)^{-1} \sum_{k=0}^{\infty} \varepsilon^k \Big[-\,(\, V_2 B\, W_2)(\, V_2 A\, W_2)^{-1} \Big]^k\,. \tag{3.82}$$

Substitution of (3.82) into (3.81) yields

$$Q_s = Q_0 + \varepsilon Q_1 + \ldots\ldots + \varepsilon^{k-1} Q_{k-1} + O(\varepsilon^k) \tag{3.83}$$

where

$$Q_1 = -\, V_0\, B\, W_2(\, V_2 A\, W_2)^{-1}\, V_2 B\, W_1\,, \tag{3.84}$$

$$Q_j = -\, V_0\, B\, W_2(\, V_2 A\, W_2)^{-1} \Big[-\,(\, V_2 B\, W_2)(\, V_2 A\, W_2)^{-1} \Big]^{j-1}\, V_2 B\, W_1,\ j \geq 2. \tag{3.85}$$

Now, let the (k+1)th order approximation of P_s be P_k, then

$$P_k = I + \varepsilon\, Q_0 + \varepsilon^2\, Q_1 + \ldots\ldots + \varepsilon^k\, Q_{k-1}. \tag{3.86}$$

Theorem 3.4

If P_k is defined by (3.86), then ξ_k is an $O(\varepsilon^k)$ approximation of ξ and π_k is an $O(\varepsilon^k)$ approximation of π, where

$$\xi_k P_k = \xi_k , \quad \xi_k \mathbf{1}_N = 1 , \tag{3.87}$$

$$\pi_k = \xi_k V_1 + \eta_k V_2 \tag{3.88}$$

and η_k is an $O(\varepsilon^k)$ approximation of η obtained by expanding the right hand side of (3.35) and dropping the ε^{k+1} and higher order terms.

Proof :

From (3.87)

$$\xi_k \bar{Q}_k = 0 , \quad \xi_k \mathbf{1}_N = 1$$

$$\bar{Q}_k = Q_0 + \varepsilon Q_1 + ... + \varepsilon^{k-1} Q_{k-1} . \tag{3.89}$$

From Lemma 3.2 and Theorem 3.3 it follows that

$$\xi_k \Psi_{ki} = e_i \tag{3.90}$$

where e_i is defined in (3.72) and Ψ_{ki} is obtained from $\bar{Q}_k$ by replacing its ith column by $\mathbf{1}_N$. Now, if we expand Q_s in ε, we get

$$Q_s = \bar{Q}_k + O(\varepsilon^k) . \quad \text{Hence,} \quad \Psi_{si} = \Psi_{ki} + O(\varepsilon^k) . \tag{3.91}$$

It follows from equations (3.72), (3.90) and (3.91) that

$$(\xi - \xi_k) \Psi_{ki} = O(\varepsilon^k). \tag{3.92}$$

From (3.91) it is clear that for ε sufficiently small, Ψ_{ki} is nonsingular. Thus

$$\xi = \xi_k + O(\varepsilon^k). \tag{3.93}$$

From (3.93) and from the fact that η is a function of ξ, it can be shown that

$$\pi = \xi_k V_1 + \eta_k V_2 + O(\varepsilon^k) = \pi_k + O(\varepsilon^k) \tag{3.94}$$

3.6 Block Diagonal Decoupling Transformation

From sections 3.3 and 3.4 it is clear that there is a wide class of transformations which give the same aggregated (first order) matrix, (3.58). Moreover, any order of approximation of the P-F eigenvector π can be achieved via an expansion of the exact reduced order system, (3.44). In this section a subclass of the general transformation matrix discussed previously is given. This transformation has a block diagonal structure. The idea of the transformation is to specialize the choice of W_2 in (3.28). Since W_1 is block diagonal, we may choose W_2 to be block diagonal as well. Such a choice will result in block diagonal matrices V_1 and V_2. So the matrices V_1, V_2, W_1 and W_2 defined in section 3.2 become

$$V_1 = \begin{bmatrix} V_1^{(1)} & 0 & 0 & 0 & 0 \\ 0 & V_1^{(2)} & 0 & 0 & . \\ 0 & 0 & V_1^{(3)} & . & . \\ . & . & . & . & . \\ 0 & 0 & 0 & 0 & V_1^{(N)} \end{bmatrix}, \tag{3.95}$$

$$V_2 = \begin{bmatrix} V_2^{(1)} & 0 & 0 & 0 & 0 \\ 0 & V_2^{(2)} & 0 & 0 & . \\ 0 & 0 & V_2^{(3)} & . & . \\ . & . & . & . & . \\ 0 & 0 & 0 & 0 & V_2^{(N)} \end{bmatrix}, \tag{3.96}$$

$$W_1 = \begin{bmatrix} \mathbf{1}_{n_1} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1}_{n_2} & 0 & 0 & . \\ 0 & 0 & \mathbf{1}_{n_3} & . & . \\ . & . & . & . & . \\ 0 & 0 & 0 & 0 & \mathbf{1}_{n_N} \end{bmatrix} \quad \text{and} \tag{3.97}$$

$$W_2 = \begin{bmatrix} W_2^{(1)} & 0 & 0 & 0 & 0 \\ 0 & W_2^{(2)} & 0 & 0 & . \\ 0 & 0 & W_2^{(3)} & . & . \\ . & . & . & . & . \\ 0 & 0 & 0 & 0 & W_2^{(N)} \end{bmatrix}, \tag{3.98}$$

where

$$V_1^{(i)}\ W_1^{(i)} = 1\ ,\quad V_1^{(i)}\ W_2^{(i)} = 0\ ,\quad V_2^{(i)}\ W_1^{(i)} = 0 \text{ and } \quad V_2^{(i)}\ W_2^{(i)} = I_{n_i - 1}\ . \tag{3.99}$$

In this transformation, computation of V_0 is easier and more efficient. Recall from (3.57) that

$$V_0 = V_1 - V_1\, A\, W_2\, (V_2\, A\, W_2)^{-1}\, V_2\ .$$

In this case V_1 , V_2 , $(V_1\, A\ W_2)$ and $(V_2\, A\ W_2)$ are block diagonal. Each diagonal block of the matrix $V_2\, A\ W_2$ is equal to $V_2^{(i)}\, A_i\ W_2^{(i)} \in \mathbf{R}^{n_i - 1 \times n_i - 1}$. This simplifies the matrix multiplications and inversion involved in computing V_0

The transformation given in [11] belongs to this class of transformations. As we discussed in 3.1, in that transformation $V_1^{(i)}$ is chosen to be the P-F eigenvector of $(A_i + I_{n_i})$, $i = 1, , ..., N$, i.e, $V_1^{(i)} A_i = 0$, $V_1^{(i)}\ \mathbf{1}_{n_i} = 1$, while the choice of $W_1^{(i)}$ is the same as our choice. The matrices $V_2^{(i)}$ and $W_2^{(i)}$ are chosen to satisfy the conditions of (3.99).

We now propose a specific choice of $W_2^{(i)}$ that renders a particular simple transformation. This choice proceeds as follows:

For A_i ,$i = 1, 2, ..., N$, let $W_1^{(i)} \in \mathbf{R}^{n_i}$ and $W_2 \in \mathbf{R}^{n_i \times n_i - 1}$ be

$$W_1^{(i)} = \mathbf{1}_{n_i} \text{ and } \quad W_2^{(i)} = \begin{bmatrix} 0 \\ I_{n_i - 1} \end{bmatrix} \tag{3.100}$$

For this choice, $V_1^{(i)} \in \mathbf{R}^{1 \times n_i}$ and $V_2^{(i)} \in \mathbf{R}^{n_i - 1 \times n_i - 1}$ can be computed to be

$$V_1^{(i)} = \left[1, 0, 0, ..., 0\right] \tag{3.101}$$

and

$$V_2^{(i)} = \begin{bmatrix} -1 & . \\ -1 & . \\ . & . \\ . & . \\ -1 & . \end{bmatrix} \quad I_{n_i - 1} \tag{3.102}$$

Our choice of transformation (3.100) — (3.102) is much simpler than the one proposed in [11] because it is independent of the system parameters. This is very important when we deal with optimal control problem as we will see in chapter 4. Moreover, the transformation $\Gamma^{-1} A \Gamma$ can be obtained by inspection.

One more feature of our transformation, (3.100) — (3.102) is that we can relate it to what we have done in chapter 2, where we keep the fast variables as physical ones while the slow states are taken as linear combinations of the other states. From (3.30), $\xi = \pi W_1$ represents the slow states and it is a linear combination of some of the states, where

$$\xi = \left[\xi_1, \xi_2, ..., \xi_N\right] \quad \text{and} \tag{3.103}$$

$$\xi_1 = \sum_{i=1}^{n_1} \pi_i \,, \quad \xi_2 = \sum_{i=n_1+1}^{n_1+n_2} \pi_i \,, ..., \quad \xi_N = \sum_{i=n-n_N+1}^{n} \pi_i \tag{3.104}$$

The " fast " states are equal to $\eta = \pi W_2$, where

$$\eta_1 = \pi_2 \,, \quad \eta_2 = \pi_3 \,, ..., \eta_{n_1} = \pi_{n_1+1} \,, ..., \eta_{n-N} = \pi_n \tag{3.105}$$

3.7 Computational Considerations

So far we have outlined a method for computing the stationary probability distribution of an NCDMC, section 3.3, as well as a procedure for obtaining

approximations to any desired degree of accuracy, section 3.5. We have also introduced a block diagonal transformation, section 3.6, to be used with this method. In this section we discuss the computational advantages of the proposed method. To put our discussions into perspective, we discuss also Courtois' and Vantilborgh's methods.

The common motivation for all these methods can be summarized in two points:

1. To reduce the higher order n-dimensional system (3.1) to a small order N-dimensional system, where N is the number of groups in the system.
2. To overcome the ill-conditioning of the problem

For an $O(\varepsilon)$ approximation of the stationary probability distribution, Courtois [25] succeeded in achieving these two goals. His method requires computing the P-F eigenvectors of the groups $(I_{n_i} + A_i)$, $i = 1, 2, ..., N$. The computation of the P-F eigenvectors can be performed either by a direct method or by an iterative method, such as the methods given in [31 — 34]. From the eigenvectors the matrix $\hat{V}_1$ of (3.9) and the approximate aggregated matrix of (3.8) are computed. These two equations are used to compute the approximate stationary probability of the whole system. Vantilborgh [29] tried to extend Courtois' work in order to obtain higher order approximations. We summarized his procedure for obtaining $O(\varepsilon^2)$ approximations in section 3.1. From that summary we can see that he needs to repeat the first step of the $O(\varepsilon)$ approximation by computing the P-F eigenvectors of the new matrices $(I_{n_i} + Q_i^*), i = 1, 2, ..., N$. So, if it is required to compute the $O(\varepsilon^k)$ approximation, the P-F eigenvectors computation must be done k times. Also in the course of forming the new matrices Q_i^*, so many approximations are done to get rid of near singular matrices, e.g., (3.14) and (3.17). This process of forming Q_i^* is very involved and the complexity will grow tremendously when higher than $O(\varepsilon^2)$ approximation is required.

We claim that our proposed method, outlined in sections 3.3 to 3.6, is a natural and a straight-forward extension of Courtois's work. The O(ε) approximation of section 3.4 is equivalent to Courtois' method. The matrix V_0 can be computed exactly as in Courtois' work, using the fact $V_0 = \hat{V}_1$. Alternatively, it can be computed using the closed form expression (3.57), which requires inversion of $V_2 A W_2$. When the block diagonal transformation is used, this matrix is block diagonal. Hence its inversion reduces to the inversion of N diagonal blocks, of dimension $(n_i - 1)$ each. The two methods for computing V_0 are comparable, since computing the P-F eigenvector of a matrix of dimension n_i is roughly equivalent to the inversion of a matrix of dimension $n_i - 1$. The computation of $(V_2 A W_2)^{-1}$ is preferable as we move towards higher order approximations, as we will see shortly.

To compute the exact stationary probability distribution using the method of section 3.3, we need to compute V_s. Recalling that

$$V_s = V_0 - \varepsilon V_0 B W_2 \left[V_2(A + \varepsilon B) W_2 \right]^{-1} V_2 \tag{3.80}$$

we see that the main step, beyond computing V_0, is the inversion of $\left[V_2(A + \varepsilon B) W_2 \right]$. As we mentioned before, when the block diagonal transformation is used, the matrix $V_2 A W_2$ is block diagonal. But $\left[V_2(A + \varepsilon B) W_2 \right]$ is not a block diagonal matrix even though it is O(ε) close to a block diagonal one. This suggests that one should exploit the closeness to a block diagonal matrix to simplify the inversion of $\left[V_2(A + \varepsilon B) W_2 \right]$. This was done in section 3.5 by expanding $\left[V_2(A + \varepsilon B) W_2 \right]^{-1}$ as a power series in ε. Although through this expansion procedure we need only to compute the inverse of the block diagonal matrix $V_2 A W_2$, the amount of associated matrix multiplications grows exponentially as we get to higher order terms in the expansion. An alternative idea to exploit the closeness of

$\left[V_2(A + \varepsilon B) W_2 \right]$ to a block diagonal matrix is to compute $\left[V_2(A + \varepsilon B) W_2 \right]^{-1}$ iteratively. This idea is developed below.

Let $X \triangleq V_2 A W_2$ and $Y \triangleq V_2 B W_2$. Then

$$Z \triangleq \left[V_2(A + \varepsilon B) W_2 \right]^{-1} = \left[X + \varepsilon Y \right]^{-1} \tag{3.106}$$

where X is a block diagonal nonsingular matrix.

Multiply (3.106) from the right by $(X + \varepsilon Y)$ to obtain

$$Z \left[X + \varepsilon Y \right] = I .$$

Hence,

$$Z = X^{-1} - \varepsilon Z Y X^{-1} \triangleq f(Z) \tag{3.107}$$

where f maps Z into Z.

The solution of this equation can be obtained via successive approximations if $f(.)$ is a contraction mapping. We have

$$f(Z_1) - f(Z_2) = \varepsilon \left[Z_2 - Z_1 \right] L$$

where $L \triangleq Y X^{-1}$ and $Z_1, Z_2 \in Z$. Thus

$$\| f(Z_1) - f(Z_2) \| \leq \varepsilon \| Z_2 - Z_1 \| \| L \| .$$

Since A, B, V_2 and W_2 are O(1), $\|L\| = O(1)$. This implies that $\varepsilon \| L \| < 1$ for sufficiently small ε. Therefore, f is a contraction mapping. Thus equation (3.107) has a unique solution which gives $f(Z) = Z$.

The successive approximation for solving the above equation is described as follows:

- Let $Z^0 = 0$

- For $k = 1, 2, \ldots\ldots$

$$Z^{k+1} = X^{-1} - \varepsilon\, Z^k\, L \tag{3.108}$$

- As $k \to \infty$, $Z^k \to \left[V_2 (A + \varepsilon B) W_2 \right]^{-1}$. Moreover,

$$\| Z^k - Z \| = O(\varepsilon^k) . \tag{3.109}$$

To show (3.109), subtract (3.108) from (3.107) to get

$$Z - Z^k = \varepsilon \left[Z^k - Z \right] L .$$

Now, for $k = 0$, $Z - Z^1 = - \varepsilon\, Z\, L = O(\varepsilon)$ and for k=1

$$Z - Z^2 = - \varepsilon (Z^1 - Z)\, L = \varepsilon^2\, Z\, L^2 = O(\varepsilon^2)$$

and by induction we can show that (4.109) is true.

Notice that the only matrix inversion required in this process is computing the inverse of the block diagonal matrix $V_2 A W_2$ which is already computed before when we compute the matrix V_0.

We conclude this section by the following example

Example 3.2 : Consider the example given in [26,29]. Its transition Probability matrix is

$$P = \begin{bmatrix}
0.85 & 0 & 0.149 & 0.0009 & 0 & 0.00005 & 0 & 0.00005 \\
0.1 & 0.65 & 0.249 & 0 & 0.0009 & 0.00005 & 0 & 0.00005 \\
0.1 & 0.8 & 0.0996 & 0.0003 & 0 & 0 & 0.0001 & 0 \\
0 & 0.0004 & 0 & 0.7 & 0.2995 & 0 & 0.0001 & 0 \\
0.0005 & 0 & 0.0004 & 0.399 & 0.6 & 0.0001 & 0 & 0 \\
0 & 0.0005 & 0 & 0 & 0.00005 & 0.6 & 0.2499 & 0.15 \\
0.00003 & 0 & 0.00003 & 0.00004 & 0 & 0.1 & 0.8 & 0.0999 \\
0 & 0.00005 & 0 & 0 & 0.00005 & 0.1999 & 0.25 & 0.55
\end{bmatrix}$$

The matrices A_i for $i = 1,2$ and 3 are chosen as

$$A_1 = \begin{bmatrix} -0.15 & 0 & 0.15 \\ 0.1 & -0.35 & 0.25 \\ 0.1 & 0.8 & -0.9 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -0.3 & 0.3 \\ 0.4 & -0.4 \end{bmatrix} \text{ and } A_3 = \begin{bmatrix} -0.4 & 0.25 & 0.15 \\ 0.1 & -0.2 & 0.1 \\ 0.2 & 0.25 & -0.45 \end{bmatrix}$$

From (3.3) and (3.5), $\varepsilon = 0.001$ and the matrix B is

$$B = \begin{bmatrix} 0 & 0 & -1 & 0.9 & 0 & 0.05 & 0 & 0.05 \\ 0 & 0 & -1 & 0 & 0.9 & 0.05 & 0 & 0.05 \\ 0 & 0 & -0.4 & 0.3 & 0 & 0 & 0.1 & 0 \\ 0 & 0.4 & 0 & 0 & -0.5 & 0 & 0.1 & 0 \\ 0.5 & 0 & 0.4 & -1 & 0 & 0.1 & 0 & 0 \\ 0 & 0.05 & 0 & 0 & 0.05 & 0 & -0.1 & 0 \\ 0.03 & 0 & 0.03 & 0.04 & 0 & 0 & 0 & -0.1 \\ 0 & 0.05 & 0 & 0 & 0.05 & -0.1 & 0 & 0 \end{bmatrix}$$

We first compute the $O(\varepsilon)$ approximation. Then the exact steady state probability is computed. First, from (3.58) the approximate aggregated matrix is

$$P_1 = \begin{bmatrix} 0.999110 & 0.000790 & 0.000100 \\ 0.000614 & 0.999286 & 0.000100 \\ 0.000056 & 0.000044 & 0.999900 \end{bmatrix}$$

The steady state probability of (3.69) is $\xi_0 = \begin{bmatrix} 0.22258 & 0.27742 & 0.50000 \end{bmatrix}$ and from (3.71),

$$\pi_0 = \begin{bmatrix} 0.089032, 0.092903, 0.040645, 0.158526, 0.118894, 0.12037, 0.277778, 0.101852 \end{bmatrix}.$$

The steady state probability computed by Courtois' method is

$$\pi_0 = \begin{bmatrix} 0.089029, 0.0929, 0.040644, 0.15853, 0.118897, 0.12037, 0.277777, 0.101852 \end{bmatrix}.$$

The two methods are equivalent in the case of the first order approximation. The slight difference in π_0 computed by the two methods is due to numerical errors.

The exact aggregated matrix computed as in (3.38) is

$$P_s = \begin{bmatrix} 0.999109 & 0.000791 & 0.000100 \\ 0.000614 & 0.999286 & 0.000100 \\ 0.000056 & 0.000044 & 0.999900 \end{bmatrix}.$$

The steady state state probability for this matrix is equal to

$$\xi = \begin{bmatrix} 0.222529 & 0.277471 & 0.500000 \end{bmatrix}$$

and the exact steady state probability as defined in (3.43) equals

$$\pi = \begin{bmatrix} 0.089283, 0.092758, 0.040488, 0.158533, 0.118938, 0.120416, 0.277795, 0.101789 \end{bmatrix}.$$

If we compute π by using (3.1), we obtain

$$\pi = \begin{bmatrix} 0.089069, 0.092959, 0.040509, 0.158529, 0.118935, 0.120385, 0.277795, 0.101819 \end{bmatrix}.$$

Notice here in this example that the difference between the exact steady state probability evaluated by the proposed method and by using the direct method, i.e., by using (3.1) is very small. That is because the order of this example is eight only. However, the condition number of the matrix $A + \varepsilon B$, with one columns replaced by the vector $\mathbf{1}_n$ is 16554 compared with 15.2345, the condition number of Q_s with one column replaced by the vector $\mathbf{1}_N$. This shows that the condition number of the aggregated matrix is much better than of the full order matrix.

CHAPTER 4

OPTIMAL AND NEAR OPTIMAL CONTROL FOR NEARLY COMPLETELY DECOMPOSABLE CONTROLLED MARKOV CHAINS

4.1 Problem Statement

For the finite state Markov chain, the transition probability is defined by

$$P_{ij}(u) = prob\left\{X_{k+1} = j \,/\, X = i,\ u_i = u\right\},\ i, j = 1, 2, \ldots\ldots, n \quad \text{and} \quad u \in U(i)\ .$$

So, $P_{ij}(u)$ is the probability that the next state is j given that the current state is i and the control applied is $u \in U(i)$, where the set $U(i)$ is a compact set for all i. If we assume that the control u is stationary, then the stationary policy $u = \left[u_1, u_2, \ldots\ldots, u_n\right]^T$ can take any value in $U \in U(1) \times U(2) \times \ldots\ldots \times U(n)$. The transition probability matrix $P(u)$ is given by

$$P(u) = \begin{bmatrix} P_{11}(u_1) & P_{12}(u_1) & \ldots & P_{1n}(u_1) \\ P_{21}(u_2) & P_{22}(u_2) & \ldots & P_{2n}(u_2) \\ \vdots & \vdots & \ldots & \vdots \\ P_{n1}(u_n) & P_{n2}(u_n) & \ldots & P_{nn}(u_n) \end{bmatrix}. \tag{4.1}$$

Notice here that the matrix has the same properties as the uncontrolled Markov chain, i.e., for each u, the row sum is 1 and its elements are nonnegative. Note also that the ith row of $P(u)$ depends only on u_i.

In this chapter we assume that $P(u)$ takes the nearly completely decomposable form, i.e., $P(u)$ can be written as

$$P(u) = I_n + A(u) + \varepsilon B(u) = I_n + Q(u) \tag{4.2}$$

where $A(u)$, $B(u)$, ε and $Q(u)$ are defined by (3.3) — (3.5). We assume that for all $u_i \in U(i)$, the matrix $P(u)$ has a single ergodic class. Moreover, for each u associated with the groups, $i = 1, 2, \ldots\ldots, N$, the stochastic matrix $I_{n_i} + A_i(u)$ has a single ergodic class. This assumption is equivalent to the following:

For every $u \in U$ there are unique stationary probability vectors $\pi(u)$ and $v_i(u)$ such that

$$\pi(u)\, Q(u) = 0, \quad \pi(u)\, \mathbf{1}_n = 1 \tag{4.3}$$

and

$$v_i(u)\, A_i(u) = 0\ , \quad v_i\, \mathbf{1}_{n_i} = 1, \quad i = 1,\ldots\ldots, N\ . \tag{4.4}$$

The problem of interest which we are going to deal with in this section is: what is the policy $u \in U$ that minimizes the average cost per stage

$$J(u) = \lim_{M \to \infty} \frac{1}{M+1}\, E \sum_{k=0}^{M} f(x_k\ ,\ u(x_k)) \tag{4.5}$$

where $f(i, u_i)$ is the instantaneous cost. This cost is assumed to be stationary and continuous in $u_i \in U(i)$ for all i.

Under the ergodicity assumption [36, 40],

$$\begin{aligned} J(u) &= \lim_{M \to \infty} \frac{1}{M+1} E \sum_{k=0}^{M} \sum_{i=1}^{n} Pr.\ [x_k = i\ ]\, f(\ x_i\ ,\ u_i\ ) \\ &= \lim_{M \to \infty} \frac{1}{M+1} E \sum_{k=0}^{M} p_0(u)\, P^k(u)\, f(u) \end{aligned} \tag{4.6}$$

where $p_0(u)$ is the initial probability distribution and

$$f(u) = \begin{bmatrix} f(1, u_1) \\ f(2, u_2) \\ \cdot \\ \cdot \\ f(n, u_n) \end{bmatrix}.$$

But,

$$\lim_{M\to\infty} \frac{1}{M+1} E \sum_{k=0}^{M} p_0(u)\, P^k(u) = p_0(u)\, \mathbf{1}_n \pi(u) = \pi(u).$$

Substituting this into (4.6) we obtain

$$J(u) = \pi(u) f(u) . \tag{4.7}$$

Notice here that $J(u)$ is independent of the initial probability $p_0(u)$; so, the optimal cost per stage is equal for all states. Now, the minimum average cost per stage turns out to be :

$$J^*(u) = \min_{u \in U} \left\{ \pi(u) f(u) \right\} \tag{4.8}$$

where $\pi(u)$ is defined by (4.3). The existence of $u \in U$ that minimizes (4.8) is guaranteed because of the compactness of U [36].

The minimum cost of (4.8) can be obtained by solving the two equations (4.3) and (4.7) for all $u \in U$; then choosing the policy that gives the minimum average cost. This procedure is possible if the order of the matrix $Q(u)$ is small and the number of policies among which we choose is finite and small. Unfortunately, this is not the case for most practical problems. Equation (4.3) usually represents a very high order system. In addition to that, it is very ill-conditioned as we discussed in chapter 3. These difficulties enforced the researchers [36 — 43] to look for an alternative approach to tackle this problem. In section 4.2, four numerical methods are presented to solve the minimum average cost problem in its general form, and the fifth method deals specifically with the class of nearly-completely decomposable systems.

In section 4.3 we will make use of the structure of $Q(u)$ to overcome the high dimension and the ill-conditioning problems by using the two-time scale property to decompose the system into small and well-conditioned subsystems. This will help us solve the optimal control problem of (4.8) which is the topic of section 4.4.

In section 4.5, we consider the solution of (4.8) as $\varepsilon \rightarrow 0$. This results in a reduced problem for which a near-optimal policy is obtained. Finally, in section 4.6, an example is given to illustrate the optimal and the near-optimal solutions.

4.2 Algorithms for Computing the Optimal Control Policy

For $c \in \mathbf{R}^n$ and $\lambda \in \mathbf{R}$, equation (4.7) can be written as [36, 39]

$$\lambda \mathbf{1}_n = Q(u)\, c + f(u) \tag{4.9}$$

where we have n equations in $n + 1$ unknowns. The following lemma is recalled from [36, 39].

Lemma 4.1 Under the above assumption about $Q(u)$ and $f(u)$ and for $u \in U$,

i. If (λ, c) is a solution to (4.9), then $\lambda = J(u)$.

ii. If (λ, c) is a solution to (4.9), then $(\lambda, c + \delta \mathbf{1}_n)$ is also a solution for any scalar δ.

iii. A solution always exists.

In the literature there are several algorithms to compute the stationary policy that minimizes the cost (4.9). These methods are described briefly in this section.

1. Policy Iteration Method [41]

This method is composed of two steps: the value determination and the policy improvement steps.

a. Value Determination Operation

For a given admissible stationary policy u^k at the kth iteration, the average cost according to Lemma 4.1 is equal to λ^k, i.e., $J(u^k) = \lambda^k$, where the superscript denotes the iteration number. The corresponding unknown dual variables $c^k \in \mathbf{R}^n$ are given

by

$$\lambda^k = f(i, u_i^k) + \sum_{j=1}^{n} q_{ij}(u_i^k)\, c^k(j) \quad ,i = 1, 2,....., n$$

or in the matrix form

$$\lambda^k \mathbf{1}_n = f(u^k) + Q(u^k)\, c^k . \tag{4.10}$$

Recall from Lemma 4.1 that if (λ^k, c^k) is a solution to (4.10), then so is $(\lambda^k, c^k + \delta \mathbf{1}_n)$ for every constant δ. This is equivalent to setting one of the n elements of c^k to zero. Therefore, equation (4.10) with

$$c^k(s) = 0 , \tag{4.11}$$

where s is any state, has a unique solution; see [40, 41].

b. Policy Improvement

The next stationary policy u^{k+1} is determined by minimizing

$$H_i(u^{k+1}, \varepsilon) = \min_{u_i \in U(i)} \left\{ f(i, u_i) + \sum_{j=1}^{n} q_{ij}(u_i)\, c^k(j) \right\}, \; i = 1, 2,....., n . \tag{4.12}$$

In the matrix form this minimization can be written as

$$H(u^{k+1}, \varepsilon) = \min_{u \in U} \left\{ f(u) + Q(u)\, c^k \right\}. \tag{4.13}$$

Because in the controlled Markov chain every element in a row is dependent on one particular policy u_i, the minimization (4.12) is a point wise minimization. So we can find the minimum policy for each row separately. Then the iteration will proceed with this new policy to the value determination step and the cycle is repeated. If u_i^k gives the minimum cost in (4.12), we choose $u_i^{k+1} = u_i^k$ even if there are other controls giving the minimum beside u_i^k [40]. If the policies on two successive iterations are equal, then the optimal policy is reached, i.e., $J^*(u) = \lambda^k$ and the iteration is stopped.

In this algorithm, it is assumed that the system is a complete ergodic Markov processes, i.e., through the iterations, we assume that every stationary policy gives a Markov chain with a single ergodic class [41]. It was proved in [41] that in this algorithm the policy evaluated at each iteration gives a lower cost than the previous one. This guarantees that the algorithm converges. Moreover, if the number of policies is finite, the iteration converge in a finite number of steps.

The implementation of the algorithm is based on being able to solve the linear equation which is defined by (4.10) and (4.11).

The following methods do not require a matrix inversion of order n in each iteration; but they solve the optimization problem iteratively.

2. Successive Approximations Method [42]

This algorithm is summarized as follows:

Pick any initial value for c^0; for $k = 1, 2, ...$, perform the minimization

$$C^k(i) = \min_{u_i \in U(i)} \left\{ f(i, u_i) + \sum_{j=1}^{n} q_{ij}(u_i)\, c^{k-1}(j) + c^{k-1}(i) \right\} \quad ,i = 1, 2, ..., n \,. \qquad (4.14)$$

Set $\lambda^k = C^k(s)$ and $c^k(i) = C^k(i) - \lambda^k$ for $i = 1, 2, ...n$. The iteration is repeated until the algorithm converges.

It is shown in [42] that $J(u^*) = \lim_{k \to \infty} \lambda^k$. This method does not require solving n linear equations, however, the rate of convergence is low [37].

3. Gauss-Seidel Method [43]

This method is an iterative method as the successive approximation one except that at every iteration the new $C^k(i)$ is used in computing $C^k(j)$ for $j > i$, i.e., equation (4.14) is replaced by

$$C^k(i) = \min_{u_i \in U(i)} \left\{ f(i, u_i) + \sum_{j=1}^{i-1} q_{ij}(u_i) C^{k-1}(j) + c^{k-1}(i) + \sum_{j=i}^{n} q_{ij}(u_i)\, c^{k-1}(j) \right\},\ i = 1,\ldots,n \ . \quad (4.15)$$

It has been found in [40] that the convergence of this method is better than the successive approximations method. The convergence of the Gauss-Seidel method is, in general, guaranteed if the matrix $Q(u)$ is diagonally dominant [45].

4. Varaiya's Algorithm [36, 37]

This algorithm can be described in the following steps:

- Given the tolerance $\delta > 0$, choose c^0.
- For $k = 1, 2, \ldots\ldots$, $h(c^k) \in \mathbf{R}^n$ is defined by

$$h(c^k) = \min_{u \in U} \left\{ f(u) + Q(u)\, c^k \right\}. \quad (4.16)$$

- For the minimum of (4.16) determine

$$\overline{h}(c^k) = \max_i h_i(c^k) \quad \text{and} \quad \underline{h}(c^k) = \min_i h_i(c^k) \quad (4.17)$$

where i is the ith element of the vector $h(c^k)$, and

$$\hat{h}(c^k) = \frac{1}{n} \sum_{i=1}^{n} h_i(c^k) \quad \text{and} \quad K(c^k) = \overline{h}(c^k) - \hat{h}(c^k)\, \mathbf{1}_n \ . \quad (4.18)$$

- If $\overline{h}(c^k) - \underline{h}(c^k) < \delta$, stop and set $u^* = u^k$ and $J(u^*) = \hat{h}(c^k)$.

Else

$$c^{k+1} = c^k + \Delta\, K(c^k) \quad (4.19)$$

where the step size Δ is a small positive constant.

- The procedure continues till convergence is achieved.

5. Nested Optimization Method

The nearly completely decomposable Markov chain of (4.2) was considered in [30, 38] where a near optimal average cost is computed. This method employed the transformation (3.22) to transform equation (4.9) into the aggregated form

$$J\,\mathbf{1}_N = \left[\hat{V}_1(u) - \varepsilon\, F\,(u)\,\hat{V}_2\right] B(u)W_1\hat{c}_1 + \hat{f}\,(u) \tag{4.20}$$

where

$$F\,(u) = \hat{V}_1\,(u)\,B(u)\hat{W}_2(u)\left[\hat{V}_2(\,A(u) + \varepsilon\,B(u))\,\hat{W}_2(u)\right]^{-1} \tag{4.21}$$

and

$$\hat{f}\,(u) = \left[\hat{V}_1\,(u) - \varepsilon\,F\,(u)\hat{V}_2\right] f\,(u)\,. \tag{4.22}$$

If we try to apply Varaiya's algorithm to (4.20) to obtain the minimum average cost, we obtain

$$h\,(\hat{c}_1^k) = \min_{u\,\in\,U}\left\{\left[\hat{V}_1(u) - \varepsilon\,F\,(u)\,\hat{V}_2\right] B(u)W_1\hat{c}_1^k + \hat{f}\,(u)\right\} \tag{4.23}$$

where in this case as $k\rightarrow\infty$, $h\,(\hat{c}_1^k)\rightarrow J\,\mathbf{1}_N$. But, equation (4.23) is nonlinear and the transformation of (3.22) depends on u. This makes the minimization of (4.23) very complicated task whether we use Varaiya's algorithm or any other known algorithm.

Now, if we let ε equal zero, (4.23) becomes

$$h\,(\hat{c}_1^k) = \min_{u\,\in\,U}\left\{\hat{V}_1(u)\,B(u)W_1\,\hat{c}_1^k + \hat{V}_1(u)\,\hat{f}\,(u)\right\} = \min_{u\,\in\,U}\left\{\hat{V}_1(u)\left[\,B(u)W_1\,\hat{c}_1^k + \hat{f}\,(u)\right]\right\}. \tag{4.24}$$

Notice that each row of $B(u)\,W_1\,\hat{c}_1^k$ and $f\,(u)$ is a function of only one control u_i , $i = 1, 2,\ldots,n$. Now, partition u and $f\,(u)$ as follows

$$u = \begin{bmatrix} u^1 \\ u^2 \\ . \\ . \\ u^N \end{bmatrix} \quad \text{and} \quad f(u) = \begin{bmatrix} f(u^1) \\ f(u^2) \\ . \\ . \\ f(u^N) \end{bmatrix} \tag{4.25}$$

where $u^i \in \mathbf{R}^{n_i} \in U^i$ is the control associated with the fast subsystem A_i, $i = 1, ..., N$ and $f(u^i)$ is the corresponding instantaneous cost. Now, let $g^k(u)$ be equal to

$$g^k(u) = B(u)\, W_1\, \hat{c}_1^k + f(u) = \begin{bmatrix} g^k(u^1) \\ g^k(u^2) \\ . \\ . \\ g^k(u^N) \end{bmatrix}. \tag{4.26}$$

Hence equation (4.24) becomes

$$h(\hat{c}_1^k) = \min_{u \in U} \left\{ \hat{V}_1(u)\, g^k(u) \right\} \tag{4.27}$$

where $g^k(u)$ can be considered as the instantaneous cost for (4.27).

Recall that $\hat{V}_1(u)$ is block diagonal; therefore, equation (4.24) can be written as

$$h_i(\hat{c}_1^k) = \min_{u^i \in U^i} \left\{ v_i(u)\, g^k(u^i) \right\}, \; i = 1, 2, ..., N . \tag{4.28}$$

Equation (4.28) shows that for a fixed dual variable $\hat{c}_1^k$, the minimization problem of (4.27) can be separated into N different optimization problems for the fast subsystems $A_i(u)$, $i = 1, 2, ..., N$. Varaiya's algorithm can now be applied as follows:

For a fixed $\hat{c}_1^k$ and for each $i = 1, 2, ..., n$, the minimum average cost can be computed by

$$h_i(\hat{c}_1^k, d^j) = \min_{u^i \in U^i} \left\{ A_i(u^i)\, d^j + g^k(u^i) \right\}, \; i = 1, 2, ..., N . \tag{4.29}$$

If the algorithm converges, then as $j \to \infty$, $h_i(\hat{c}_1^k, d^j) \to h_i(\hat{c}_1^k)$ for $i = 1, 2, ..., N$.

Notice here that the minimum cost for (4.27) is computed at $\hat{c}_1^k$. Now $\hat{c}_1^k$ is updated by the same procedure as described in method 4 to obtain $\hat{c}_1^{k+1}$. This updated dual variable is used to update the cost

$$g^{k+1}(u) = B(u)W_1 \hat{c}_1^{k+1} + f(u) .$$

The minimization of (4.28) is repeated to obtain $h(\hat{c}_1^{k+1})$ from (4.27) and the cycle continues till the algorithm converges. Notice that in this algorithm the minimization of the aggregated system (4.27) is replaced by the minimization of N fast subsystems defined by (4.28) and (4.29). Thus, the optimization is nested. So, this algorithm decomposes the optimization problem into $(N + 1)$ subproblems: one slow time-scale problem and N fast time-scale problems.

Let us finally note that, because of the complexity of (4.23), the optimal solution of the average cost is very difficult. This makes seeking a near-optimal rather than optimal solution a must if we decide to choose this method.

In section 4.3 we will see that the optimal solution for the average cost per stage is possible by using a transformation which does not depend on the control u as the one proposed in section 3.6.

A comparison between the first, second and fourth methods is given in [37], where the rate of convergence, number of operations and the CPU time are compared for the three methods. This comparison shows that the first method has an advantage over the other two methods in the CPU time and the rate of convergence. However, this method does not work well when the matrix $Q(u)$ is near-singular. This led the authors of [37] to favor the fourth method.

In the following section we will treat the near-singularity and the high order problems in order to get benefit of the advantages of the first method.

4.3 Decomposition of the Average Cost for a Fixed Policy

We have seen in the previous section that the cost $J(u) = \lambda$ is defined by the following two equations

$$\lambda \mathbf{1}_n = Q(u)\, c + f(u) \tag{4.9}$$

and

$$c(s) = 0 . \tag{4.30}$$

Due to the high dimensions of $Q(u)$ and the ill-conditioning problem, solving the above two equations is impractical.

In this section the cost $J(u)$ is decomposed into fast and aggregate components by using the two time scale property of nearly decomposable Markov chains and the transformation developed in Chapter 3. As discussed in Chapter 3, the transformation can be chosen to independent of the system parameters. A specific example is given in section 3.6. Throughout this chapter, whenever we refer to the transformation of Chapter 3, it is implied that the transformation is independent of the system parameters; hence, independent of u. The decomposition can be done as follows: From equation (4.9)

$$\Gamma^{-1}\lambda \mathbf{1}_n = \Gamma^{-1} Q(u) \Gamma \Gamma^{-1} c + \Gamma^{-1} f(u) . \tag{4.31}$$

Substitute (3.28) and (3.29) into (4.31) to obtain

$$\begin{bmatrix} \lambda \mathbf{1}_N \\ 0 \end{bmatrix} = \begin{bmatrix} \varepsilon V_1 B(u) W_1 & V_1(A(u) + \varepsilon B(u)) W_2 \\ \varepsilon V_2 B(u) W_1 & V_2(A(u) + \varepsilon B(u)) W_2 \end{bmatrix} \begin{bmatrix} V_1 c \\ V_2 c \end{bmatrix} + \begin{bmatrix} V_1 f(u) \\ V_2 f(u) \end{bmatrix} \tag{4.32}$$

Let

$$V_1 c = \frac{1}{\varepsilon} c_1 \quad \text{and} \quad V_2 c = c_2 . \tag{4.33}$$

Therefore (4.32) becomes

$$\lambda \, \mathbf{1}_N = V_1 \, B(u) \, W_1 \, c_1 + V_1(A(u) + \varepsilon \, B(u)) \, W_2 \, c_2 + V_1 \, f(u) \tag{4.34}$$

$$0 = V_2 \, B(u) \, W_1 \, c_1 + V_2(A(u) + \varepsilon \, B(u)) \, W_2 \, c_2 + V_2 \, f(u) \, . \tag{4.35}$$

It was shown in Chapter 3 that $V_2(A(u) + \varepsilon \, B(u)) \, W_2$ is nonsingular; thus,

$$c_2 = - \left[V_2(A(u) + \varepsilon \, B(u)) \, W_2 \right]^{-1} \left[V_2 \, B(u) \, W_1 \, c_1 + V_2 \, f(u) \right] . \tag{4.36}$$

Substituting this into (4.34) yields

$$\lambda \, \mathbf{1}_N = V_s(u) \, B(u) \, W_1 \, c_1 + V_s(u) \, f(u) = Q_s(u) \, c_1 + f_s(u) \tag{4.37}$$

where

$$V_s(u) = V_1 - V_1(A(u) + \varepsilon \, B(u)) \, W_2 \left[V_2(A(u) + \varepsilon \, B(u))) \, W_2 \right]^{-1} V_2 \, , \tag{4.38}$$

$$Q_s(u) = V_s(u) \, B(u) \, W_1 \tag{4.39}$$

and

$$f_s(u) = V_s(u) \, f(u) \, . \tag{4.40}$$

Now, the linear system of n equations (4.9) is reduced to a smaller order system of order N, where N is defined in Chapter 3. Similarly, if we apply this transformation to equation (4.7), we obtain

$$J(u) = \pi(u) \, \Gamma \, \Gamma^{-1} f(u) = \begin{bmatrix} \xi(u) & \eta(u) \end{bmatrix} \begin{bmatrix} V_1 f(u) \\ V_2 f(u) \end{bmatrix} \tag{4.41}$$

where $\xi(u) = \pi(u) \, W_1$ and $\eta(u) = \pi(u) \, W_2$. Using (3.35), we obtain

$$\eta(u) = - \xi(u) \, V_1(A(u) + \varepsilon \, B(u)) \, W_2 \left[V_2(A(u) + \varepsilon \, B(u)) \, W_2 \right]^{-1} \tag{4.42}$$

substitution of (4.42) into (4.41), yields

$$J(u) = \xi(u) \, f_s(u) \tag{4.43}$$

where $\xi(u)$ is defined by

$$\xi(u)\, Q_s(u) = 0 \tag{4.44}$$

$$\xi(u)\, \mathbf{1}_N = 1\,. \tag{4.45}$$

Notice here that the properties of V_s and Q_s which are defined in Chapter 3 still hold for $V_s(u)$ and $Q_s(u)$.

From Chapter 3 we know that $Q_s(u)$ has some of the properties of $Q(u)$ such as the row sum of $Q_s(u)$ equals zero and the uniqueness of the zero eigenvalue, but $(\mathbf{1}_N + \varepsilon\, Q_s(u))$ may not be a stochastic matrix. Therefore, the existence of solution of (4.37) does not follow from standard properties of Markov chains. The following two theorems show the existence and uniqueness of the solution of (4.37).

Theorem 4.1

For a fixed policy u, consider the N linear equations with $(N+1)$ unknowns $\lambda \in \mathbf{R}$ and $c_1 \in \mathbf{R}^N$ of equation (4.37). Then

i. If $(\lambda,\ c_1)$ is a solution to (4.37), then $\lambda = J(u)$.

ii. If $(\lambda,\ c_1)$ is a solution to (4.37), then $(\lambda,\ c_1 + \delta\, \mathbf{1}_N)$ is also a solution for every δ.

iii. A solution always exists.

Proof :

i. Multiply (4.37) on the left by $\xi(u)$ to obtain

$$\lambda = \xi(u)\, Q_s(u)\, c_1 + \xi(u)\, f_s(u) = \xi(u)\, f_s(u) = J(u)$$

ii. The proof of this part comes from the fact that the row sum of $Q_s(u)$ equals zero; therefore $\delta\, Q_s(u)\, \mathbf{1}_N = 0$. That is to say $\delta\, \mathbf{1}_N \in N(Q_s(u))$, where $N(Q_s(u))$ is the null space of $Q_s(u)$.

iii. Recall that $A\,x = d$ has a solution if $d \in R(A)$, where, $R(A)$ is the range space of A.

Now, from equations (4.43) and (4,45)

$$\xi(u)\left[J(u)\,\mathbf{1}_N - f_s(u)\right] = 0\,.$$

So, $\xi(u)$ is orthogonal to $\left[J(u)\,\mathbf{1}_N - f_s(u)\right]$. This implies that

$$N(Q_s^t(u)) \perp \left[J(u)\,\mathbf{1}_N - f_s(u)\right]. \tag{4.46}$$

But

$$N(Q_s^t(u)) = R(Q_s(u))^c \tag{4.47}$$

where $R(Q_s(u))^c$ is the orthogonal complement of the range space $R(Q_s(u))$. Substituting (4.47) into (4.46) yields

$$R(Q_s(u))^c \perp \left[J(u)\,\mathbf{1}_N - f_s(u)\right]. \tag{4.48}$$

Therefore, the columns of $\left[J(u)\,\mathbf{1}_N - f_s(u)\right]$ are in the range space of $Q_s(u)$ which means that there exist a vector $c_1 \in \mathbf{R}^N$ such that

$$Q_s(u)\,c_1 = J(u)\,\mathbf{1}_N - f_s(u)\,.$$

Thus, c_1 is a solution.

The above theorem proves the existence of a solution. The following theorem proves the uniqueness of the solution under certain conditions

Theorem 4.2

For $c_1,\ f_s(u) \in \mathbf{R}^N$, let the ith element of c_1 be equal to zero, then

$$\lambda\,\mathbf{1}_N = Q_s(u)\;c_1 + f_s(u)$$

with

$$c_1(i) = 0 \tag{4.49}$$

has a unique solution.

Proof :

To prove that (4.37) and (4.49) have a unique solution, it is enough to prove that

$$\Psi_{si}(u) = \left[q_s^1(u),\ q_s^2(u)\ ,.....,q_s^{i-1}(u)\ ,\ -\mathbf{1}_N\ ,\ q_s^{i+1}(u),.....,q_s^N(u)\right]$$

where, $q_s^i(u)$ is the ith column of $Q_s(u)$, is nonsingular. The proof of this is a direct result of Lemma 3.2.

So, by using the two-time scale property of the full order system, we could reduce the order of the system from n to N and the system which we need to solve now has a unique solution and is defined by (4.37) and (4.49).

At this stage we have overcome the drawback of the policy iteration method which requires solving n, near-singular linear equations at each iteration. Instead, we solve N, well-conditioned linear equations.

In the next section, the policy iteration method is employed to solve the optimal control problem and in the following section it will be used for the near-optimal case.

4.4 Optimal Policies for the Average Cost Problem

In this section the policy iteration method [41] is used to minimize the cost of the aggregated system, i.e.,

$$\lambda\ \mathbf{1}_N = \min_{u \in U}\left\{ Q_s(u)\ c_1 + f_s(u)\right\} \tag{4.50}$$

The value determination method for aggregated form is used to compute λ^k and c_1^k for a fixed policy u^k at the kth iteration. This can be done by solving N linear equations in N unknowns as follows

$$\lambda^k\ \mathbf{1}_N = Q_s(u^k)\ c_1^k + f_s(u^k) \tag{4.51}$$

$$c_1^k(t) = 0 \tag{4.52}$$

where t is any state.

For a reason which will be discussed shortly, in addition to c_1^k, we need to compute c^k. This can be done by computing c_2^k first; by using (4.36)

$$c_2^k = -\left[V_2(A(u^k) + \varepsilon B(u^k))\, W_2\right]^{-1}\left[V_2 B(u^k)\, W_1\, c_1^k + V_2 f(u^k)\right]; \tag{4.53}$$

using equation (4.33), the dual variable c^k can then be written as

$$c^k = \begin{bmatrix} W_1 & W_2 \end{bmatrix}\begin{bmatrix} V_1 c^k \\ V_2 c^k \end{bmatrix} = \frac{1}{\varepsilon} W_1\, c_1^k + W_2\, c_2^k \tag{4.54}$$

Once we compute λ^k and c^k at the kth iteration, we can go to the policy improvement step to compute the next policy. Now, for the full order system, it was shown in section 4.2 that the minimization step is a point wise minimization, i.e., each row can be minimized separately. But, if we try to minimize the aggregated system (4.50), we will loose this property. Moreover, we need to disaggregate the reduced policies to the original ones in order to choose the minimum policies among the given physical ones. Thus, this step will be applied to the full order system; therefore, the minimization step (4.13) become

$$H(u^{k+1},\ \varepsilon) = \min_{u \in U}\left\{f(u) + (A(u) + \varepsilon B(u))\, c^k\right\}. \tag{4.55}$$

Substitute (4.54) into (4.55) and from the fact that $A(u)\, W_1 = 0$, we obtain

$$H(u^{k+1},\ \varepsilon) = \min_{u \in U}\left\{f(u) + B(u)\, W_1\, c_1^k + (A(u) + \varepsilon B(u))\, W_2\, c_2^k\right\}. \tag{4.56}$$

Since W_1 and W_2 are independent of u, we still have each row in (4.56) dependent on one control $u_i \in U(i)$. So, the point wise minimization can still be performed in this step.

It is important to notice that because the transformation proposed in Chapter 3 is independent on u, this step could be done without violating the point wise minimization. This step would not be possible if $\hat{W}_2(u)$ of [11] is used instead of W_2.

We conclude this section by summarizing the above algorithm.

1. **Value Determination Step**

- For an admissible stationary policy u^k at the kth iteration, compute the aggregated matrix $Q_s(u^k)$ and the aggregated instantaneous cost $f_s(u^k)$ as defined in (4.39) and (4.40), respectively.
- Solve for λ^k and c_1^k which are uniquely determined by (4.51) and (4.52). Set $J(u^k) = \lambda^k$ and by (4.53), compute c_2^k. Now, pass the dual variables c_1^k and c_2^k to the policy improvement step.

2. **Policy Improvement Step**

To compute the next policy, i.e., u^{k+1}, use equation (4.56) with the new c_1^k and c_2^k computed by the first step. This minimization, as we discussed previously, is performed one row at a time. The new policy u^{k+1} is sent to the value determination step to obtain new values for λ^{k+1}, c_1^{k+1} and c_2^{k+1} and the iteration continues until the the algorithm converges; for which $u^{k+1} = u^k$ and $J(u^*) = \lambda^k$.

The convergence of this algorithm is a by-product of the theorem given in [41] because the similarity transformation we applied to the system will not change its convergence.

We conclude this section by noting that with this algorithm, instead of solving n ill-conditioned linear equations in each iteration, we only solve N well-conditioned linear equations, where $N \ll n$.

4.5 Near Optimal Policies for the Average Cost Problem

The most expensive part in the algorithm proposed in section 4.4 is computing $V_s(u)$ which is defined by (4.38). In this section we wish to take advantage of the singularly perturbed form of the nearly decomposable Markov chain to obtain a near-optimal average cost per stage by approximating $V_s(u)$. By letting $\varepsilon = 0$ in (4.34) and (4.35) we obtain

$$\lambda_0 \mathbf{1}_N = V_1 B(u) W_1 c_{10} + V_1 A(u) W_2 c_{20} + V_1 f(u) \tag{4.57}$$

$$0 = V_2 B(u) W_1 c_{10} + V_2 A(u) W_2 c_{20} + V_2 f(u) . \tag{4.58}$$

Since $V_2 A(u) W_2$ is nonsingular,

$$c_{20} = -\left[V_2 A(u) W_2\right]^{-1}\left[V_2 B(u) W_1 c_{10} + V_2 f(u)\right] \tag{4.59}$$

Substituting (4.59) into (4.57) yields

$$\lambda_0 \mathbf{1}_N = \left[V_1 - V_1 A(u) W_2 (V_2 A(u) W_2)^{-1} V_2\right]\left[B(u) W_1 c_{10} + f(u)\right]$$

$$\triangleq V_0(u) B(u) W_1 c_{10} + V_0(u) f(u) \triangleq Q_0(u) c_{10} + f_0(u) \tag{4.60}$$

where

$$V_0(u) = V_1 - V_1 A(u) W_2\left[V_2 A(u) W_2\right]^{-1} V_2 , \tag{4.61}$$

$$Q_0(u) = V_0(u) B(u) W_1 \tag{4.62}$$

and

$$f_0(u) = V_0(u) f(u) . \tag{4.63}$$

Equations (4.60) — (4.63) are the unperturbed form of equations (4.37) — (4.40).

By the same procedure, equations (4.43) — (4.45) can be written as

$$J_0(u) = \xi_0(u) f_0(u) \tag{4.64}$$

where $\xi_0(u)$ is defined by

$$\xi_0(u)\, Q_0(u) = 0\,, \quad \xi_0(u) > 0 \tag{4.65}$$

$$\xi_0(u)\, \mathbf{1}_N = 1\,. \tag{4.66}$$

From Theorem 3.2 of Chapter 3 we notice that the results of Theorems 4.1 and 4.2 hold for λ_0, c_{10} and that

$$\lambda_0\, \mathbf{1}_N = Q_0(u)\, c_{10} + f_0(u) \tag{4.67}$$

with

$$c_{10}(i) = 0\,, \tag{4.68}$$

where i is any state, has a unique solution.

In the following theorem we want to show in what sense is the optimal solution of (4.64) a near optimal solution of the original problem (4.43).

Theorem 4.3

Let $f(u)$ be a uniformly bounded function of u for all $u \in U$. If u_0^* is a policy that minimizes (4.64), i.e.,

$$J_0(u_0^*) = \min_{u \in U} J_0(u) = \min_{u \in U} \left\{ \xi_0(u)\, f_0(u) \right\} \tag{4.69}$$

and if u^* is a policy that minimizes (4.43), i.e.,

$$J(u^*) = \min_{u \in U} J(u) = \min_{u \in U} \left\{ \xi(u)\, f_s(u) \right\}. \tag{4.70}$$

Then

$$J(u_0^*) = J(u^*) + O(\varepsilon) \tag{4.71}$$

Proof :

From Theorem 3.3 and equations (3.80) and (3.83)

$$\xi\,(u) = \xi_0\,(u) + O(\,\varepsilon\,)\,, \tag{4.72}$$

$$V_s\,(u) = V_0\,(u) + O(\,\varepsilon\,) \tag{4.73}$$

and

$$Q_s\,(u) = Q_0\,(u) + O(\,\varepsilon\,)\,. \tag{4.74}$$

Substitute (4.72) — (4.74) into (4.43) to obtain

$$J(u) = \Big[\,\xi_0(u) + O(\,\varepsilon)\,\Big] \times \Big[\,V_0\,(u) + O(\,\varepsilon)\,\Big] f\,(u) = \xi_0\,(u)\, f_0\,(u) + O(\,\varepsilon\,)\,. \tag{4.75}$$

Therefore,

$$J(u_0^*) = J(u_0^*) + O(\,\varepsilon) = \min_{u\,\in\,U}\Big\{\,J_0(u)\Big\} + O(\,\varepsilon\,)$$

$$= \min_{u\,\in\,U}\Big\{\,\xi_0(u)\; V_0\,(u)\, f\,(u)\,\Big\} + O(\,\varepsilon)$$

$$= \min_{u\,\in\,U}\Big\{\,(\,\xi\,(u) + O(\,\varepsilon\,)\,)\,(\,V_s\,(u) + O(\,\varepsilon\,)\,)\, f\,(u)\,\Big\} + O(\,\varepsilon\,)\,.$$

Because $|\,f\,(u)\,| \le K$ for all u, we have

$$J(u_0^*) = \min_{u\,\in\,U}\Big\{\xi\,(u)\; V_s\,(u)\, f\,(u)\Big\} + O(\,\varepsilon\,)$$

$$= \min_{u\,\in\,U}\Big\{\,J(u)\,\Big\} + O(\,\varepsilon\,) = J(u^*) + O(\,\varepsilon\,) \tag{4.76}$$

From the previous discussion we found that letting $\varepsilon = 0$ gives us a near optimal solution. If the near optimal solution satisfies our needs, then the computation

of the value determination step will be much easier because instead of computing $Q_s(u)$, we need to compute $Q_0(u)$. As it was shown in section 3.7, we can use the block diagonal transformation to simplify the computation of $Q_0(u)$. If we need higher order approximation of $Q_s(u)$, we can use the technique used in sections (3.5) and (3.7). The higher order approximation of $Q_s(u)$ gives a higher order approximation to $J(u^*)$. Computing the near optimal average cost per stage $J_0(u)$ is performed by the same algorithm presented in the previous section. In this case the two steps of the algorithm are modified as follows

1. **Value determination Step**

For the kth iteration and for a given stationary policy u^k, compute $Q_0(u)$ and $f_0(u)$ as defined in (4.62) and (4.63). Solve for λ_0^k and c_{10}^k by using equations (4.67) and (4.68) and use equation (4.59) to compute c_{20}^k.

2. **Policy Improvement Step**

Use the dual variables c_{10}^k and c_{20}^k computed from the value determination step to compute the next policy u^{k+1} as follows

$$H(u^{k+1}, 0) = \min_{u \in U} \left\{ f(u) + B(u) W_1 c_{10}^k + A(u) W_2 c_{20}^k \right\}. \tag{4.77}$$

Notice here that the minimization step is independent of ε. The new policy u^{k+1} is passed to the value determination step and the cycle continues until the convergence is achieved.

In the following theorem we want to prove that using this modified version, the algorithm gives a policy that attains the minimum average cost per stage for the problem described by (4.67) and (4.68).

Theorem 4.4

The algorithm described above produces a monotonically decreasing cost and converges to a policy that gives the minimum cost per stage to equations (4.67) and (4.68).

proof :

Suppose a policy $\bar{u}$ is chosen over the policy u by the policy Improvement routine (4.77). Let $\bar{\lambda}_0$, $\bar{c}_{10}$ and $\bar{c}_{20}$ be the corresponding average cost and dual variables computed by (4.67), (4.68) and (4.59). Then (4.57) can be written as

$$\bar{\lambda}_0 = V_1 B(\bar{u}) W_1 \bar{c}_{10} + V_1 A(\bar{u}) W_2 \bar{c}_{20} + V_1 f(\bar{u}) \tag{4.78}$$

Now, if $\bar{u} \neq u$, we want to prove that $\bar{\lambda}_0 \leq \lambda_0$.

Because the policy improvement step (4.77) chose $\bar{u}$ over u,

$$f(\bar{u}) + A(\bar{u}) W_2 c_{20} + B(\bar{u}) W_1 c_{10} \leq f(u) + A(u) W_2 c_{20} + B(u) W_1 c_{10} .$$

This can be written as

$$\gamma = f(u) - f(\bar{u}) + (A(u) - A(\bar{u})) W_2 c_{20} + (B(u) - B(\bar{u})) W_1 c_{10} \geq 0 \tag{4.79}$$

where $\gamma \in \mathbf{R}^n$.

The value determination step for $\bar{u}$ and u is given respectively, by (4.78) and (4.57). Subtract (4.78) from (4.57) to obtain

$$\begin{aligned}(\lambda_0 - \bar{\lambda}_0)\, \mathbf{1}_N = {} & V_1 B(u) W_1 c_{10} - V_1 B(\bar{u}) W_1 \bar{c}_{10} + V_1 A(u) W_2 c_{20} \\ & - V_1 A(\bar{u}) W_2 \bar{c}_{20} + V_1 f(u) - V_1 f(\bar{u}) .\end{aligned} \tag{4.80}$$

Substitution of (4.79) into (4.80) yields

$$\begin{aligned}(\lambda_0 - \bar{\lambda}_0)\, \mathbf{1}_N = {} & V_1 \Big[\gamma + A(\bar{u}) W_2 c_{20} + B(\bar{u}) W_1 c_{10} \Big] \\ & - V_1 B(\bar{u}) W_1 \bar{c}_{10} - V_1 A(\bar{u}) W_2 \bar{c}_{20} .\end{aligned}$$

This equation can be written as

$$(\lambda_0 - \bar{\lambda}_0)\, \mathbf{1}_N = V_1\, \gamma + V_1\, A(\bar{u})\, W_2(\, c_{20} - \bar{c}_{20}\,) + V_1\, B(\bar{u})\, W_1(\, c_{10} - \bar{c}_{10}\,)\,. \quad (4.81)$$

Let $\lambda_0^{\Delta} = \lambda_0 - \bar{\lambda}_0$, $c_{10}^{\Delta} = c_{10} - \bar{c}_{10}$ and $c_{20}^{\Delta} = c_{20} - \bar{c}_{20}$; therefore equation (4.81) becomes

$$\lambda_0^{\Delta}\, \mathbf{1}_N = V_1\gamma + V_1\, A(\bar{u})\, W_2 c_{20}^{\Delta} + V_1\, B(\bar{u})\, W_1 c_{10}^{\Delta}\,. \quad (4.82)$$

If we apply the same procedure to (4.58), we obtain

$$0 = V_2\, B(\bar{u})\, W_1 c_{10}^{\Delta} + V_2\, A(\bar{u})\, W_2 c_{20}^{\Delta} + V_2\, \gamma \quad (4.83)$$

$V_2\, A(\bar{u})\, W_2$ is nonsingular; hence,

$$c_{20}^{\Delta} = -\left[V_2\, A(\bar{u})\, W_2\right]^{-1}\left[V_2\, B(\bar{u})\, W_1 c_{10}^{\Delta} + V_2\, \gamma\right]. \quad (4.84)$$

Substituting this in (4.82) yields

$$\lambda_0^{\Delta}\, \mathbf{1}_N = Q_0(\bar{u})\, c_{10}^{\Delta} + V_0(\bar{u})\, \gamma\,. \quad (4.85)$$

Notice here that equation (4.85) takes the same form of (4.60). From (4.64) and Theorem 4.1, the solution of (4.60) satisfies

$$\lambda_0 = J(u) = \xi_0\,(u)\, f_0\,(u) = \xi_0\,(u)\, V_0\,(u)\, f\,(u)$$

where $\xi_0\,(u)$ is defined by equations (4.65) and (4.66). Similarly, λ_0^{Δ} can be written as

$$\lambda_0^{\Delta} = \xi\,(\bar{u})\, V_0\,(\bar{u})\, \gamma\,. \quad (4.86)$$

Because $Q_0\,(\bar{u})$ has the single ergodic class property, the elements of $\xi\,(\bar{u})$ are positive. From Theorem 3.1, part v and equation (4.79), we see that $V_0\,(\bar{u})$ and γ are nonnegative, therefore,

$$\lambda_0^{\Delta} > 0 \quad \Rightarrow\ \lambda_0 \geq \bar{\lambda}_0\,.$$

Therefore, the average cost is monotonically decreasing. Since it is bounded from below, it is convergent.

The second step of the proof is to show that it converges to a policy which gives the lowest average cost. Suppose it does not, i.e., let us assume that $\bar{\lambda}_0 < \lambda_0$, i.e., $\lambda_0^{\Delta} > 0$ and suppose that the policy improvement step (4.77) converges to u rather than $\bar{u}$. If this the case, then equation (4.79) yields $\gamma \leq 0$ which means by (4.86) that $\lambda_0^{\Delta} \leq 0$. But this contradicts the assumption that $\bar{\lambda}_0 < \lambda_0$; therefore, the policy will converge to $\bar{u}$. This completes the proof.

4.6 Example : Minimizing the Average Cost Per Stage

In this section, we consider the example given in [30]. The algorithm presented in sections 4.4 and 4.5 are applied to this example to compute the optimal and near-optimal policies and the average cost per stage corresponding to these policies. Also this example is solved by the policy iteration method [41] applied to the full order system. The control problem in this example can be visualized as one of maintenance scheduling. Similar to this problem, hydro scheduling with multiple time scales, was considered in [44]. The probability transition matrix for this example is given by

$$\begin{bmatrix}
.45 & .45 & 0 & .05 & .05 & 0 & 0 & 0 & 0 \\
.27 & .36 & .27 & .03 & .04 & .03 & 0 & 0 & 0 \\
0 & .72 & 0 & 0 & .08 & .02 & 0 & 0 & 0 \\
.5u & .5u & .3u & .45 - .5u & .45 - .5u & 0 & .05 & .05 & 0 \\
.3u & .4u & .2u & .27 - .3u & .36 - .4u & 0.27 - .3u & .03 & .04 & .03 \\
0 & .8u & 0 & 0 & .72 - .8u & .18 - .2u & 0 & .08 & .02 \\
0 & 0 & 0 & .5u & .5u & 0 & .5 - .5u & .5 - .5u & 0 \\
0 & 0 & 0 & .3u & .4u & .3u & .3 - .3u & .4 - .4u & .3 - .3u \\
0 & 0 & 0 & 0 & .8u & .2u & 0 & .8 - .8u & .2 - .2u
\end{bmatrix}$$

The state variables are defined by the variable G and D as follow:

state	G	D
x_1	2	0
x_2	2	1
x_3	2	2
x_4	1	0
x_5	1	1
x_6	1	2
x_7	0	0
x_8	0	1
x_9	0	2

where G is the number of power generating units available and D is the demand in terms of generating units needed. The control variable $u \in [\ 0.02\ ,\ 0.2\ ]$. The problem is to find the policy $u\ (G\ ,\ D)$ that minimizes the average cost per stage (4.5) if the instantaneous cost is

$$f\ (G,\ D,\ u(G,\ D)) = \left[(\ D - G)^{+}\right]^2 + K(u(G,\ D))^2$$

where $K = 30$ and $(D - G)^{+} = \max\ ((G - D),\ 0)$.

From (4.2), for $\varepsilon = 0.2$

$$A(u) = \begin{bmatrix} -0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & -0.6 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & -.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & -0.6 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & -.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & -0.6 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & -.8 \end{bmatrix}$$

and $B(u)$ equals

$$\begin{bmatrix} -.25 & -.25 & 0 & .25 & .25 & 0 & 0 & 0 & 0 \\ -.15 & -.2 & -.15 & .15 & .2 & .15 & 0 & 0 & 0 \\ 0 & -.4 & -.1 & 0 & .4 & .1 & 0 & 0 & 0 \\ 2.5u & 2.5u & 0 & -.25-2.5u & -.25-2.5u & 0 & .25 & .25 & 0 \\ 1.5u & 2u & 1.5u & -.15-1.5u & -.2-2u & -.15-1.5u & .15 & .2 & .15 \\ 0 & 4u & u & 0 & -.4-4u & -.1-u & 0 & .4 & .1 \\ 0 & 0 & 0 & 2.5u & 2.5u & 0 & -2.5u & -2.5u & 0 \\ 0 & 0 & 0 & 1.5u & 2u & 1.5u & -1.5u & -2u & -1.5u \\ 0 & 0 & 0 & 0 & 4u & u & 0 & -4u & -u \end{bmatrix}$$

The optimal policy u^* and the minimum average cost per stage are computed for different values of ε by using the algorithm developed in section 4.4. These results are given in table 4.1.

Table 4.1 The optimal policies and the average cost for different values of ε evaluated by the algorithm of section 4.4

ε	The minimum policy u^*									$J(u^*)$
0.2	.02	.02	.02	.10513	.11178	.11213	.12740	.14478	.15023	0.6709471
0.1	.02	.02	.02	.10761	.11087	.11102	.13376	.14287	.14590	0.6714597
0.05	.02	.02	.02	.10880	.11041	.11048	.13716	.14183	.14343	0.6716016
10^{-2}	.02	.02	.02	.10972	.11004	.11005	.14000	.14096	.14129	0.6716502
10^{-3}	.02	.02	.02	.10992	.10995	.10995	.14066	.14076	.14079	0.6716522
10^{-4}	.02	.02	.02	.10994	.10994	.10994	.14073	.14074	.14074	0.6716520
10^{-5}	.02	.02	.02	.10994	.10994	.10994	.14073	.14073	.14073	0.6716521

For all the values of ε the algorithm converges in 5 to 7 iterations. If we let $\varepsilon = 0$, the near optimal policy u_0^* and the average cost per stage are computed by the method given in section 4.5. The following table gives u_0^* and $J(u_0^*)$

Table 4.2 The near optimal policy and the corresponding average cost

	0.02
	0.02
The	0.02
minimum	0.10994
policy	0.10994
	0.10994
u_0^*	0.14073
	0.14073
	0.14073
$J(u_0^*)$	0.6716522

Notice here that u_0^* and $J(u_0^*)$ are independent of ε. Also as $\varepsilon \to 0$, $u^* \to u_0^*$.

Finally, this example is solved by the policy iteration method [41] which is described in section 4.2 and the optimal policies for different ε is given in table 4.3

Table 4.3 The optimal policy and the minimum average cost evaluated by the policy iteration method [41] applied to the full order system.

ε	The minimum policy u^*									$J(u^*)$
0.2	.02	.02	.02	.10513	.11178	.11213	.12740	.14478	.15023	0.6709469
0.1	.02	.02	.02	.10761	.11087	.11102	.13376	.14287	.14590	0.6714593
0.05	.02	.02	.02	.10880	.11041	.11048	.13716	.14183	.14343	0.6716016
10^{-2}	.02	.02	.02	.10972	.11004	.11005	.14000	.14096	.14129	0.6716472
10^{-3}	.02	.02	.02	.10993	.10996	.10996	.14066	.14076	.14079	0.6716321
10^{-4}	.02	.02	.02	.10992	.10992	.10992	.14072	.14073	.14073	0.671725
10^{-5}	.02	.02	.02	.10996	.10996	.10996	.14091	.14091	.14091	0.6692431

When ε gets smaller, the convergence of this algorithm is not predicted because of the near singularity of the matrix $Q(u)$, for example, in this example at $\varepsilon = 10^{-3}$, the algorithm does not converge and the minimum average cost fluctuates between .6716139, .6716321 and .6716230 all the way up to 100 iterations.

- At $\varepsilon = 10^{-4}$ the program takes only 6 iterations, but it does not converge to the right one.

- At $\varepsilon = 10^{-5}$ the cost fluctuates between .6701525 and .6692431 up to 100 iterations.

None of these problems happened when the optimal policy is evaluated by using the algorithm of section 4.4, see Table 4.1.

CHAPTER 5

CONCLUSIONS AND FURTHER RESEARCH

The real Schur form decomposition is used to develop a reliable and numerically stable algorithm that transforms any two-time scale system into the singularly perturbed form. The algorithm comprises two step: first, transform the matrix A into an ordered real Schur form; second, balance the elements of the ordered real Schur form such that T_{11} and T_{12} are O(1) and T_{22} is O(ε). If we are interested in the physical state variables of the system, sufficient conditions are derived to put the two-time scale system in the singularly perturbed form with all the fast or all the slow state variables chosen from the original physical variables. These sufficient conditions hold when the matrix A is given by $A = A_0 + \varepsilon A_1$, where A_0 is a singular matrix with a semisimple null structure. This case has been extensively studied in the literature. The significance of these conditions over the previous results in this area is that we do not require that the matrix A be modeled in the form $A = A_0 + \varepsilon A_1$. Such modeling form, in general, requires a priori knowledge about the physical nature of the system. Also, necessary and sufficient conditions are given to retain all the physical states and the modeling step is achieved by permutation only.

In Chapter 3, stochastic systems which can be modeled as large finite-state nearly completely decomposable Markov chains are considered. A general transformation is proposed to decompose the large and the ill-conditioned Markov chain system into a reduced order and well-conditioned aggregated matrix (3.37). This transformation enabled us to compute the exact steady state probability distribution of the NCDMC (3.43), a problem which is usually encountered in queueing network. It

is shown that the transformations available in the literature to handle this class of problems are a subclass of our proposed transformation. It is also shown that all the transformations that satisfy the conditions of Section 3.2 give the same first order approximation of the aggregated matrix (3.58). A block diagonal transformation, which is a subclass of the general one, is proposed to simplify and reduce the amount of computations required to form the aggregated matrix.

Finally, in Chapter 4, the controlled Markov chain of the same class as the one treated in Chapter 3 is considered. An algorithm composed of two steps, value determination and policy improvement steps, is proposed. This algorithm computes optimal policies that attain the minimum average cost per stage [40]. This algorithm overcomes the ill-conditioning problem associated with the NCDMC.

Suboptimal policies can be computed with less computational effort. In Section 4.5, a modified version of the above algorithm is given to compute the suboptimal policies and the average cost per stage when $\varepsilon = 0$. The convergence of this algorithm is shown. The above algorithm and its modified version are applied to an example to compute the optimal and suboptimal policies and the average cost per stage corresponding to these policies.

The possibility of further research and applications of the subject of this dissertation is very high. In Chapter 2, for example, the block diagonalization of equation (2.8) is needed to decompose the singularly perturbed form into two separate subsystems, slow and fast. Since the algorithm proposed in this chapter balances the elements of the ith off diagonal block to be $O(\lambda(S_{ii}))$, the solution of the Sylvester equation which block diagonalizes equation (2.8) is expected to be well conditioned and O(1). To confirm this expectation it is necessary to show that the separation between T_{11} and T_{22} is O(1).

The transformation of Theorem 2.1 is applied to two-time scale systems so that the state variables of one scale are preserved to be physical ones. A multi-time scale transformation which preserves the state variables of more than one time scale as physical variables is a point that needs further investigation.

In Chapter 4, we consider the optimal solution of the average cost per stage (4.5). An alternative optimal control formulation is to minimize a discounted cost, defined by

$$J(u) = \min_{u \in U} \sum_{k=0}^{\infty} \beta^k f(x_k, u(x_k)) \tag{5.1}$$

where β is the discount rate. The algorithm considered in Sections 4.4 and 4.5 could be extended to this problem to compute optimal and suboptimal policies that minimize the discounted cost of equation (5.1).

In equation (4.43), we have closed-form expressions for $\xi(u)$ and $f_s(u)$. The minimization of this equation is possible; but disaggregating the optimal reduced policies into the original ones is an open question and it has to be the subject of further consideration.

In the algorithm we proposed in Section 4.4, a symbolic inversion of the matrix $V_2 (A(u) + \varepsilon B(u)) W_2$ which is required in the value determination step, will reduce the amount of computations drastically.

I think the theoretical richness of this area combined with the numerous applications which could be modeled as Markov chains should result in many more contributions to Markov chain modeling and applications in the near future.

APPENDIX

APPENDIX

To show that

$$V_s = V_0 - \varepsilon V_0 B W_2 \left[V_2(A + \varepsilon B) W_2 \right]^{-1} V_2 ,$$

let us expand $\left[V_2(A + \varepsilon B) W_2 \right]^{-1}$ in ε and substitute into (3.45) to obtain

$$V_s = V_1 - V_1(A+ \varepsilon B) W_2 \left[(V_2 A W_2)^{-1} \sum_{k=0}^{\infty} \varepsilon^k \left[- (V_2 B W_2)(V_2 A W_2)^{-1} \right]^k \right] V_2$$

$$= V_1 - V_1(A + \varepsilon B) W_2 \left[V_2 A W_2 \right]^{-1} \{ I - \varepsilon V_2 B W_2 \left[V_2 A W_2 \right]^{-1}$$

$$+ \varepsilon^2 \left[V_2 B W_2 \left[V_2 A W_2 \right]^{-1} \right]^2 - \varepsilon^3 \left[V_2 B W_2 \left[V_2 A W_2 \right]^{-1} \right]^3 + \ldots \} V_2 .$$

This can be written as

$$V_s = V_1 - (V_1 A W_2) \left[V_2 A W_2 \right]^{-1} V_2 - \varepsilon \{ V_1 B W_2 \left[V_2 A W_2 \right]^{-1} V_2$$

$$- V_1 A W_2 \left[V_2 A W_2 \right]^{-1} V_2 B W_2 \left[V_2 A W_2 \right]^{-1} V_2 \}$$

$$+ \varepsilon^2 \{ V_1 B W_2 \left[V_2 A W_2 \right]^{-1} (V_2 B W_2) \left[V_2 A W_2 \right]^{-1} V_2$$

$$- (V_1 A W_2) \left[V_2 A W_2 \right]^{-1} \left[V_2 B W_2 \left[V_2 A W_2 \right]^{-1} \right]^2 V_2 \} + \varepsilon^3 \ldots .$$

Substitute for V_0 to obtain

$$V_s = V_0 - \varepsilon \left[V_1 - (V_1 A W_2) \left[V_2 A W_2 \right]^{-1} V_2 \right] B W_2 \left[V_2 A W_2 \right]^{-1} V_2$$

$$+ \varepsilon^2 \left[V_1 - (V_1 A W_2) \left[V_2 A W_2 \right]^{-1} V_2 \right] B W_2$$

$$\times \left[V_2 A W_2 \right]^{-1} (V_2 B W_2) \left[V_2 A W_2 \right]^{-1} V_2 + ...$$

The terms in the square brackets equal to V_0; therefore,

$$V_s = V_0 - \varepsilon V_0 B W_2 \Big\{ \left[V_2 A W_2 \right]^{-1} - \varepsilon \left[V_2 A W_2 \right]^{-1}$$

$$\left[V_2 B W_2 \right] \left[V_2 A W_2 \right]^{-1} + \varepsilon^2 \left[V_2 A W_2 \right]^{-1}$$

$$\left[V_2 B W_2) \left[V_2 A W_2 \right]^{-1} \right]^2 - .... \Big\} V_2$$

$$= V_0 - \varepsilon V_0 B W_2 \Big[V_2(A + \varepsilon B) W_2 \Big]^{-1} V_2 .$$

LIST OF REFERENCES

LIST OF REFERENCES

1. P. V. Kokotovic, H. K. Khalil and J. O'Reilly, Singular Perturbation Methods in Control: Analysis and Design. New York: Academic Press, 1986.

2. P. Kokotovic and H. Khalil, Ed., Singular Perturbations in Systems and Control, IEEE press, 1986.

3. G. H. Golub and J. H. Wilkinson, "Ill-conditioned Eigensystems and the Computation of the Jordan Canonical Form," SIAM Rev., vol. 18, pp. 578-619, 1976.

4. J. H. Chow and P. Kokotovic, "Eigenvalue Placement in Two-Time-Scale Systems," Proc. of IFAC Symposium on large scale systems, 1976.

5. J. Anderson, "Decoupling of Two-Time-Scale Linear Systems," Proc. of Joint Automatic Control, 1978.

6. G. P. Syrcos and P. Sannuti, "Singular Perturbation Modeling of Continuous and Discrete Physical System," Int. J. Control, vol. 37, pp. 1007-1022, 1983.

7. B. Avramovic, P. V. Kokotovic, J. R. Winkelman and J. H. Chow, "Area Decomposition for Electromechanical Models of Power Systems," Automatica, vol. 16, pp. 637-648, 1980.

8. A. J. Fossard, M. Berthelot and J. F. Magni, "On Coherency-based Decomposition Algoritms," Automatica, vol. 19, pp. 247-253, 1983.

9. I. J. Perez-Arriaga, G. C. Verghese and F. C. Schweppe, "Selective Modal Analysis with Application to Electric Power Systems, Part 1, IEEE Trans. power Appar. and Sys., vol. PAS-101, 1982

10. J. H. Chow, Ed., Time-Scale Modeling of Dynamic Networks with Application to Power System," Lecture Notes In Control Information Science, vol. 46, New York: Springer-Verlag, 1982.

11. R. G. Phillips and P. V. Kokotovic, "A Singular Perturbation Approach to Modeling and Control of Markov Chains," IEEE Trans. Automat. Contr., vol., AC-26, pp. 1087-1094, 1981.

12. M. Coderch, A. S. Willsky, S. S. Sastry and D. A. Castanon, "Hierarchical Aggregation of Linear Systems with Multiple Time Scales," IEEE Trans.

Automat. Contr., AC-28, pp. 1017-1030, 1983.

13. H. K. Khalil, "Time Scale Decomposition of Linear Implicit Singularly Perturbed Systems," IEEE Trans. Automat. Contr., vol. AC-29, pp. 1054-1056, 1984.

14. X.-C Lou, G. C. Verghese, A. S. Willsky and M. Vidyasagar, "An Algebraic Approach to Analysis and Control of Time-Scales," Proc. 1984 ACC, June 1984, pp. 1368-1372.

15. X.-C Lou, "An Algebraic Approach to the Analysis and Control of Time-Scales," PH.D. Dissertation, Dept. Elec. Eng. Comput. Sci. M.I.T., 1985.

16. G. H. Golub and C. F. Van Loan, Matrix Computations, Baltimore, The Johns Hopkins University Press, 1983.

17. G. W. Stewart, Introduction to Matrix Computations, New York, Academic Press, 1973.

18. B. T. Smith et al., Matrix Eigensystems Routines-EISPACK Guide, 2nd ed. (Lecture Notes in Computer Science), vol. 6, New York: Springer-Verlag, 1976.

19. A. J. Laub, "A Schur Method for Solving Algebraic Riccati Equations," IEEE Trans. Automat. Contr., vol. AC-24, pp. 913-921, 1979.

20. G. W. Stewart, "Algorithm 506: HQR3 and EXCHG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix," ACM Trans. Math. Software, vol. 2, pp. 275-280, 1976.

21. D. S. Flamm and R. W. Walker, "Remark On Algorithm 506," ACM Trans Math. Software, vol. 8, pp. 219-220, 1982.

22. B. N. Parlett and C. Reinch, "Balancing a Matrix for Calculation of Eigenvalues and Eigenvectors," Numer. Math., vol. 13, pp. 293-304, 1969.

23. E. E. Osborne, "On Pre-Conditioning of Matrices," Jour. ACM 7, pp. 338-345, 1960.

24. H. A. Simon and A. Ando, "Aggregation of Variables in Dynamic Systems," Econometrica 29, pp. 111-138, 1961.

25. P. J. Courtois, Decomposability Queueing and Computer System Applications. New York, Acadimic Press, 1977.

26. P. J. Courtois, "Error Analysis in Nearly-Completely Decomposable Stochastic Systems," Econometrica, vol. 43, pp. 691-709, 1975.

27. P. J. Courtois, "Decomposability, Instabilities, and Saturation in Multiprogramming Systems," Commun. ACM 18, pp. 371-377, 1975.

28. A. Brandwajn, "A Model of a Time Sharing Virtual Memory System Solved Using Equivalence and Decomposition Methods," Acta Informatica 4, 11-47, 1974.

29. H. Vantilborgh, "Aggregation with an Error of $O(\varepsilon^2)$," J. ACM vol. 32, pp. 162-190, 1985.

30. R. G. Phillips, "Near Optimal Policies for Large Scale Markovian Decision Processes," Proc. Int. Large Scale Sys. Symp., pp. 507-512, 1982.

31. W. J. Stewart, "A Comparison of Numerical Techniques in Markov Modeling," Commun. ACM 21, pp. 144-152, 1978.

32. J. R. Koury, D. F. McAllister and W. J. Stewart, "Iterative Methods For Computing Stationary Distributions of Nearly Completely Decomposable Markov Chains," SIAM J. Alg. Disc. Math. 5, pp. 164-186, 1984.

33. D. F. McAllister, G. W. Stewart and W. J. Stewart, "On a Rayleigh-Ritz Refinement Technique for Nearly Uncoupled Stochastic Matrices," Linear Alg. Applications 60, pp. 1-25, 1984.

34. W-L Cao and W. J. Stewart, "Iterative Aggregation / Disaggregation Techniques for Nearly Uncoupled Markov Chains," J. ACM 32, pp. 702-719, 1985.

35. J. L. Doob, Stochastic Processes. New York: Wiley Press, 1953.

36. P. Varaiya, "Optimal and Suboptimal Stationary Controls for Markov Chains," IEEE Trans. on Automat. Contr., vol. AC-23, pp. 388-394, 1978.

37. J. L. Popyack, R. L. Brown, and C. C. White, III, "Discrete Versions of an Algorithm Due Varaiya," IEEE Trans. on Automat. Contr., vol. AC-24, June 1979.

38. S. H. Javid, "Nested Optimization of Weakly Coupled Markov Chains," Proc. of the 18th Annual Allerton Conf. on Communication, Control and Computing, Univ. of Illinois, pp. 881 - 890, 1980.

39. D. Teneketzis, S. H. Javid, and B. Sridhar, "Control of Weakly-Coupled Markov Chains," Proc. 1980 CDC, pp. 137-142, 1980.

40. D. P. Bertsekas, Dynamic Programming: Deterministic and Stochastic Models. Prentice-Hall, Inc., 1987.

41. R. A. Howard, Dynamic Programming and Markov Processes. Cambridge, MA: MIT Press, 1960.

42. D. J. White, "Dynamic Programming, Markov Chains, and the Method of Successive Approximations," J. Math. Anal. Appl., vol. 6, pp. 373-376, 1963.

43. H. Kushner, Introduction to Stochastic Control. New York: Holt, Rinehart and Winston, 1971.

44. A. Cohen et.al., "Research and Development of a Unified Approach to Operations Scheduling for Electric Power Under Uncertainty," Final Report, USDOE Contract No. DEACO1-79ET29243, Oct., 1982. of Energ. Sys., Final Report, 1982.

45. J. Stoer and R. Bulirsch, Introduction to Numerical Analysis. New York: Springer-Verlag, 1980.

46. L. Kleinrock, Queueing Systems, Vol. I and II, New York: Wiley, 1976.