This is to certify that the

dissertation entitled

PARAMETER ESTIMATION FOR THE FAST AND SLOW SUBSYSTEMS OF A PROCESS OPERATING IN COUPLED SINGULARLY PERTURBED FORM

presented by

Michael Joel Cook

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Systems Science

Robert O. Barr, Jr.
Major professor

Date June 2, 1982

# PARAMETER ESTIMATION FOR THE FAST AND SLOW SUBSYSTEMS OF A PROCESS OPERATING IN COUPLED SINGULARLY PERTURBED FORM

By

Michael Joel Cook

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

1982

ABSTRACT

PARAMETER ESTIMATION FOR THE FAST AND SLOW
SUBSYSTEMS OF A PROCESS OPERATING
IN COUPLED SINGULARLY PERTURBED FORM

By

Michael Joel Cook

The input and output of a deterministic singularly perturbed system, operating in coupled form, are observed over a finite time-interval. The problem under consideration is to determine the system parameters of the decoupled subsystems from these measurements. The nature and formulation of the singularly perturbed system are examined along with the fundamentals of systems identification. A finite time-interval identification method is investigated which utilizes a filter to annihilate the initial condition response, and models disturbances as solutions to a homogeneous differential equation. The adaptation of this method is applied to the singularly perturbed problem, and a unique procedure for its implementation is presented via a heuristic study of linear time invariant systems. The experimental results indicate success of the methodology for reasonable separation of the

subsystems, as characterized through the inherent time scale parameter.

To Leonard and Rheva, my dad and mom, for their unconditional love and sacrifices, and for their wisdom in sending me to a tutor in seventh grade when I was ready to give up on mathematics.

And to kindred air pirates Steve and Sandy, my brother and sister, for the greatest times in my life.

They always want the best for me.
Who could ask for a better family!

## ACKNOWLEDGMENTS

I express my sincere gratitude to Dr. Robert O. Barr, my advisor, whose supervision and guidance and encouragement throughout my program were heightened not only by his friendly nature, but also by his genuine concern and understanding--especially through the ebbs and flows of my research.

My affirmations also extend to the other members of my committee--Dr. Robert A. Schlueter, Dr. Hassan K. Khalil, and Dr. James H. Stapleton--for all their valued inputs, directions, assistance, and kindness, and for their creative instructions throughout my coursework interactions with them.

I also thank Dr. Erik Goodman and the consultants at the Case Center for their readiness and availability in answering my questions concerning the modus operandi of the Prime 750 Computer.

And thanks are extended to Ali Saberi--a colleague and a friend--for his feedback and insights, and for the enjoyable discussions we share paralleling systems theory with everyday life.

My special thanks to Carol Cole, whose fast and excellent typing of this dissertation is greatly appreciated.

I would also like to express my absolute validations to all my friends, whose constant support and love I will cherish always. Three of these folk are very special to me: Karin Montgomery, whose love and confidence in me and my abilities have always given me strength; Kitty Buffington, who has been there each and every day with a smile, giving me her special attention and friendship; and Dr. Peter Brobeil, my white-water sternman and best buddy, whose altruistic beliefs and attitudes towards life and nature enhance the quality friendship that we share. These friends are a treasure to me. I might not have reached my goals without them.

Finally, a heart-felt thanks to a fellow joey, Al "Whistles" Fast, my clown instructor, who kept me (a.k.a. "Noodles") laughing through all the experiences of the last two years.

TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER I

# INTRODUCTION

The purpose of this chapter is to serve as a foundation for the results of this thesis. Section 1.1 will commence the discussion with the concept of singular perturbation and its significance in systems theory. The role of estimation in the identification of systems will be the theme of Section 1.2. A survey of general identification schemes and their application will be examined in Section 1.3. The aggregated problem of identification and singular perturbation as presented in Section 1.4 will complete this preview.

## 1.1. Singular Perturbation

A system can be defined as a function whose domain is a set of inputs and whose range is a set of outputs. The functional relationship and behavior between the inputs and outputs are based on the inherent characteristics of the physical system under consideration. In studying this behavior it often becomes necessary to construct a mathematical model, in which the relationships between the physical variables in the system are mapped onto the mathematical structures via equations. To acquire

a full representation of the system often requires many variables and equations, which tend to increase the complexity of the model. This largeness of the model can be due to the inclusion of all factors which affect the system, even those contributions which have little effect on the behavior. They may be nearly negligible because their cumulative effect is small during the operation of the system, or they might be relatively "short-lived" in comparison to the other variables and thus do not dominate the mid-term and long-term phenomena. Since these small contributors must be included for a complete representation in the original model description of the system, they are classified as "parasitic."

Examples of such parasitics are small time constants, masses, moments of inertia, capacitances, inductances, and any other relatively unimportant parameters. Besides increasing the dynamic order of the system, these parasitics introduce "fast modes" making the model "stiff"; that is, hard to handle on a digital computer because the equations require small integration intervals. Solution of the system equations becomes overly-complicated, although numerical methods have been developed to increase the efficiency of the solution procedures [CLA], [GE].

A set of dynamic equations containing such parasitics is called a <u>singularly perturbed system</u>, since the

solution to the equations can be constructed as a power series in terms of a small perturbation parameter $\varepsilon$ [WA], [GA].

In a singularly perturbed system there are generally many time-scales needed in describing the system behavior. For instance, there can be very fast and very slow phenomena requiring three or more separate time-scales [DE], [HO]. In this thesis, the discussion will be limited to the case of two time scales, for very fast and for normal-speed phenomena. The separation between the two time scales is directly related to $\varepsilon$ in that the smaller $\varepsilon$ is, the wider the separation. The extension of this thesis to multi-time-scaled systems is an area for further research.

The small parasitics of the system are considered as proportional to the perturbation parameter $\varepsilon$ [KO-1]. The effect that the parasitics have on the system behavior occurs immediately upon initiation of the system. Therefore the states of the system associated with these parasitics are called "fast" states, and their swift effect dies out rapidly allowing for these states to reach their quasi-steady-states very quickly. The other states not associated with the parasitics take longer to affect a change in the system behavior and are therefore dubbed "slow" states. The underlying assumption in singular perturbation theory is that the slow variables remain constant at the onset of the system and the fast variables are dominant during this short time, and by the time the changes in the

slow variables become noticeable, the fast variables have reached their quasi-steady-state.

A wealth of studies has arisen in reference to linear singularly perturbed systems (to be discussed below non-chronologically). Kokotovic et al. [KO-2,3] provide an overview on the use of singular perturbations in reducing the model order by first neglecting the parasitics and then reintroducing them as boundary layer corrections in separate time scales. Kokotovic et al. [KO-1] develop an iterative procedure to more accurately separate the full-ordered system into two subsystems of slow and fast states which avoids inconsistencies associated with the approach of first neglecting parasitics. Javid [JA] constructs a reduced-order state observer for the slow reduced system wherein parasitics are neglected, and derives two types of observer errors. And Saksena and Cruz [SA] design a robust low-order observer estimating the slow states using only the reduced model. O'Reilly [OR] formulates a full-order observer for the singularly perturbed system as a composition of two observers, one for each of the slow and fast subsystems. Kokotovic and Haddad [KO-4] find criteria for controllability of the slow and fast states of the system by separately analyzing the two subsystems defined by these states, and apply the separation procedure to the time-optimal

control problem [KO-5]. The ever-popular linear state regulator problem is treated by O'Malley [OM-1,2] via Hamiltonian methodology and asymptotic expansions. He gives conditions under which the optimal regulator-control problem has a unique asymptotic solution for sufficiently small $\varepsilon$. Basic theorems providing for the uniqueness and uniformness of the Riccati solution to the linear regulator problem are furnished by Kokotovic and Yackel [KO-6]. Stochastic control of the linear singularly perturbed system with additive noise is discussed by Haddad and Kokotovic [HA-1] wherein the optimal control is approximated by a near-optimal control obtained as a combination of a slow controller and a fast controller computed in separate time scales. Conditions on asymptotically stable feedback controllers by using Hurwitz criteria is dealt with by Porter [PO-1]. By applying frequency-domain techniques, Porter and Shenton [PO-2] use the special structures of the transfer function matrices for singularly perturbed systems to construct controllers. They find that in the frequency-domain the full system with slow and fast states is asymptotically equivalent to parallel connections of the reduced slow subsystem with the fast subsystem. Khalil and Kokotovic [KH] examine stability test criteria for the implementation of effective control laws for linear singularly perturbed

systems with multiparameters all of the same order, and for systems with multitime scales. They also design a near-optimal control law which does not depend on the values of the small parameters. State estimation is explored by Haddad [HA-2] for the case of input disturbances by developing two lower order filters in separate time scales. For unknown $\varepsilon$, Sebald and Haddad [SE] examine the problem of estimation of the slow and fast states of the singularly perturbed system.

And Chow et al. [CHO-1] rely on singular perturbation techniques to take a system with slightly damped high frequency oscillations and decompose it into two separate subsystems, one containing the slowly varying dynamics and the other containing only the fast oscillatory modes. Then the subsystems are analyzed in different time scales. This decomposition is also shown to work for systems whose slightly damped large eigenvalues result in sustained high frequency oscillations. These situations can occur in mechanical and electromechanical systems such as the spring-mass suspension system and the multimachine power system. Chow and Kokotovic [CHO-2] also design a near-optimal state regulator by decomposing the singularly perturbed system into two subsystems with separate fast and slow modes and then developing a composite controller based on the inputs of each subsystem.

## 1.2. Estimation and Identification

The modelling of a system and its analysis has significance in many fields, e.g. economics, biology, medicine, ecology, and certainly in the field of process control. The model building is an enormously important condition for making use of control theory. In order to better understand the dynamic behavior of a system, the system must be properly designed. Treating the system through mathematical representations allows for the modelling to be accomplished. Most certainly, an ample model of the system to be controlled is necessary or else the construction of a control law is not feasible.

A mathematical model can be considered as a function between the physical variables of the system under consideration and the mathematical equations presuming the system structure. These equations may be simple algebraic, differential, or difference types of equations. The plant system is then said to be described or modelled by the set of mathematical equations involving these physical variables.

The model is constructed theoretically and/or empirically. By theoretically analyzing the system through the usage of balance equations and the physical laws of conservation, the simple subprocesses of the plant can be described by mathematical equations. Adjoining these

equations with the appropriate boundary conditions yields the mathematical model that is desired.

The theoretic-construction approach is utilized if experiments in the plant cannot be accomplished, or if the plant is not yet in existence. However, if an experimental analysis of a plant with arbitrary structure is executed, the input and output signals are measured. Evaluation of these measurements through an identification procedure produces the mathematical model of the plant process. This estimated model is then a description of the input-output behavior of the process.

A precise definition of identification is now stated [ZA]: Given a class of systems S where each member of the class is completely specified, the identification of a system A consists in finding a system $s \in S$ that is input-output equivalent to A. It is important to note that the definition requires input-output equivalence and does not require $s \in S$ to be identical to A. Certainly, for a given input-output relation, there is generally no unique system representation [KA-1]. In this thesis the systems under consideration are to be completely specified within a parameter set and the purpose of identification is to determine, that is, estimate these parameters.

There are three major complications which appear into any real identification problem. The first deals with the absence of knowledge concerning initial conditions of the system; the second is the presence of random noises shadowing the input and output observations; and the last is the difficulty in establishing a meaningful and convenient method for estimating the system parameters as a function of the observations.

When disturbances are present they act on the process and thus affect the output signals, making more difficult the determination of a mathematical model from correctly measured input and output signals. Hence, the method of identification must separate the piece of output into the information component and the disturbance component. For linear systems, the disturbance is a single additive component of the output superimposed upon the information-carrying part. The identification scheme should overcome the influence of these disturbance signals.

The model's validity rests upon the connection between the variables of the mathematical structure and their physical counterparts. Hopefully, the relation between these entities is isomorphic [AH]. That is, the values assumed by the variables in the mathematical model are in a one-to-one property correspondence with the values that are measured.

Since one goal of identification is to determine the system model for a process under investigation, it is relevant to discuss the various model classifications that are closely associated with the identification problem. A model described by sets of differential or algebraic equations is called parametric, and the identification procedure is to determine the parameters in this structure. The number of these parameters is finite, and their true values uniquely determine the system model [BL]. These parameters may be constant or vary with time. The response description obtained from an experimental analysis of the physical process is a non-parametric model, for no à priori structure of the model can be assumed, and no finite number of parameters determines the model.

If the dynamics of the system are described by partial differential equations (e.g., parabolic, elliptical), then a distributed parameter model is being used, whereas a lumped parameter model is one using ordinary differential equations for its structure [FA]. A lumped parameter model lends itself to being discretized in time from an original continuous time model.

Models may also involve statistical values for some of its variables (stochastic-type), or there may be no probability structures at all (deterministic-type). In

the former case, the stochastic phenomena can present themselves in the form of random input and output disturbances, or, perhaps, the initial states of the system may be random variables with known or unknown means and covariances. Improper measurements of the inputs and outputs in conjunction with uncertainty in the process are a cause for difficulties in the effective identification of systems.

## 1.3. Identification Schemes and Applications

There exists quite a variety of strategies dealing with the problem of system identification [AS-1]. Step response and frequency response techniques [RA], [CHE] can accommodate both parametric and nonparametric models [IS], whereas Fourier and spectral analysis as well as correlation techniques [GO], [RA] apply only to nonparametric models. It is the tactics of parameter estimation--applicable solely to parametric models--that will be under consideration in this thesis.

The first of the parameter estimation methods is that of Least Squares [ST], [GR], [LEE]. This method is based on the thought that the most probable value of the parameters is the one "that minimizes the sum of the squares of the differences between the actually observed and computed values multiplied by weighting factors measuring the degree of precision" [GAU]. Within the least squares

methodology are the specialized schemes of Generalized Least Squares [CL], Instrumental Variables [WO], [YO], Levin's method [LEV], and the Tally principle [PE-2].

The Maximum-Likelihood method [AS-2] estimates the parameters by selecting the value of the parameters which "makes the observed data most probable in the sense that the likelihood function is maximized" [GOO]. The likelihood function is a function of the conditional probability density of the data given the parameters. Thus, the method chooses the parameters' values that makes as probable as is possible the data which is in fact observed [BL].

Another scheme for parameter estimation is through a Bayesian approach [DO], [PE-1]. In this method the estimates are taken from the à posteriori conditional density of the parameters given the input-output data. This is done by the use of Bayes' Theorem [LEE] on the conditional probability density of the data given the parameters--the function which is the argument of the likelihood functional.

In both the Maximum Likelihood and Bayesian estimation methods, it is necessary to make assumptions on the probability distributions of the data and parameters. The difficulties involved with expressing the à priori information in terms of a probability distribution can be circumvented by using prediction error methods [AS-2].

In this case, a prediction model (similar to the Kalman filter [KA-2]) is implemented and the parameters are estimated by minimizing a criterion which is a function of the predicted output.

One of the main purposes of identification is to determine the dynamics of a process so that a proper control law may be designed and implemented to cause the system to perform according to some set of criteria. For example, better knowledge of a production industry plant or an economic system may be obtained for improved control.

The identification procedure can also be utilized for a diagnostic examination to analyze the properties of a system, such as the determination of rate coefficients in chemical reactions and reactivity coefficients in nuclear reactors. This goal has practicality in biology, economics, medicine, and many other related fields.

Of course, identification of a process may simply be carried out to verify the structure of a theoretical model which was posed. And by continuously monitoring a process, a system identifier can learn parameters which vary slowly through time. Thus, at each instant of time, the system behavior is approximated and an effective controller can be implemented for that instant. A controller constructed by way of this type of parameter

learning procedure is called a parameter adaptive controller [KU], [SK]. And sometimes when system parameters vary, a reliability index of that system may change. A check on the reliability of the system can be maintained by identifying the system parameters.

## 1.4. Object of the Thesis

This first chapter has staged an introduction to two areas of systems theory: singularly perturbed systems and parameter estimation. The rest of this treatise is to serve as a tutorial to the unification of these two studies.

Chapter II will establish the algebraic language of this text and the mathematical structure of the singularly perturbed system. The third chapter will concentrate on the solution to the identification problem at hand, including the mathematical and systems approaches and techniques utilized. Chapter IV will provide tangible reckoning of the newly-constructed procedures through computer-oriented examples.

The final chapter will summarize the results of this thesis and provide directional comments for further research in this area.

# CHAPTER II

## FORMULATION OF THE PROBLEM

In this chapter the mathematics behind the singularly perturbed system will be introduced. The construction of the two-time scale concept will be included along with the algebraic "evolution" of that system. Sections 2.2 and 2.3 will focus on the decoupling; i.e., separation, of that system into fast and slow subsystems. Section 2.4 will deal with properties and theorems for the singularly perturbed system. The discussion will end with the fifth section representing the formal problem statement.

### 2.1. The Singularly Perturbed System

It is the nature of systems engineering to commence a discussion with analytic statements regarding the variables of the problem under consideration. In this case, the statements may consist of vector-form ordinary differential equations interrelating the variables. The general form of these is

$$\frac{d}{dt}\eta = F_1(\eta,u,t) \quad , \quad \eta(t_o) = \eta^o \tag{2.1.1}$$

$$y = F_2(\eta,u,t) \tag{2.1.2}$$

where $\eta$ is an n'-dimensional time-differentiable state vector, y is a q-dimensional output vector, u is an r-dimensional input vector, and t is the scalar-time variable with the initial time instant being $t_o$. (Until otherwise stated, let $t_o = 0$.)

As discussed in the introduction to this thesis, the fundamental concept embedded in singular perturbation theory is that of slow and fast states. During the onset of the process the slow variables remain relatively constant compared to the fast variables which die out quickly; i.e., reach their quasi-steady-states. Thus, if there are n slow states x and m fast states z, the n'-state vector $\eta$ can be partitioned as

$$\eta \triangleq \begin{bmatrix} x \\ z \end{bmatrix} \tag{2.1.3}$$

with $n + m = n'$. Rewriting (2.1.1) and (2.1.2) in terms of x and z yields

$$\frac{d}{dt} x = f(x,z,u,t) \ , \ x(0) = x^o \tag{2.1.4}$$

$$\frac{d}{dt} z = F_3(x,z,u,t) \ , \ z(0) = z^o \tag{2.1.5}$$

$$y = F_4(x,z,u,t) \tag{2.1.6}$$

where f, $F_3$, and $F_4$ are merely the adjusted functionals of $F_1$ and $F_2$. (Generally, f and $F_3$ are also functions

of a parameter $\varepsilon$ which represents small "parasitic" masses, capacitances, etc. of the system.)

Now, by assuming that t is the time frame for the characteristics that are slow, and allowing $\tau$ to represent the time frame for the fast characteristics, it is reasonable to assume that the ratio of t to $\tau$ is some small positive number $\varepsilon$ [KO-1]. That is, if t is in seconds and $\tau$ is in milliseconds, then $\varepsilon$ is 0.001. Assuming that $\tau = 0$ corresponds to the instant $t = 0$, it is found that

$$\tau = \frac{t}{\varepsilon} . \qquad (2.1.7)$$

If $\tau$ were now changed to microseconds, then $\varepsilon$ would decrease in magnitude. And when $\varepsilon$ is shrunk, the fixed t period will correspond to quite a long $\tau$ period. So, as $\varepsilon$ is decreased toward zero, one fixed t period will correspond to an infinitely long $\tau$ interval. Thus, following the example, if $\varepsilon$ were decreased to $10^{-9}$, one t period (1 second) would contain one billion $\tau$ periods (nanoseconds).

Feeling familiarized with this two-time scale concept, it seems reasonable that the states x and z should interact according to t and $\tau$, respectively. That is, the states x are $\frac{1}{\varepsilon}$ times slower than z, and likewise are their respective derivatives. Accordingly, $F_3$ can be

rescaled as $g = \varepsilon F_3$, so that g and f are of the same order of magnitude [KO-1]. Thus, equations (2.1.4) and (2.1.5) become

$$\frac{d}{dt} x = f(x,z,u,t) \ , \ x(0) = x^o \qquad (2.1.8)$$

$$\varepsilon \frac{d}{dt} z = g(x,z,u,t) \ , \ z(0) = z^o \ . \qquad (2.1.9)$$

(Recent results by Chow et al. [CHO-1] show this state description is utilizable for systems with lightly damped high frequency modes.) Note that as $\varepsilon \to 0$ here,

$$\frac{d}{dt} x_s = f(x_s,z_s,u,t) \ , \ x_s(0) = x^o \qquad (2.1.10)$$

$$0 = g(x_s,z_s,u,t) \qquad (2.1.11)$$

where $x_s(t)$ and $z_s(t)$ are the quasi-steady-states of x(t) and z(t), respectively. Here then, equation (2.1.11) is algebraic and can now be backward substituted into (2.1.10) to yield a new differential equation in $x_s$.

It is worth noting that for the long-term studies in the classical quasi-steady-state approach, the derivative of z with respect to t in equation (2.1.5) is set equal to zero which then yields the system of equations (2.1.10) and (2.1.11). Thus, $dz_s/dt = 0$ which requires $z_s$ to be a constant. However, equation (2.1.11) defines $z_s$ as a time-varying quantity. Even though this procedure is justifiable in yielding approximate solutions, it does

leave this obvious inconsistency. It is through the introduction of the two-time scale concept discussed above that this inconsistency is circumvented. For in equation (2.1.11) $\varepsilon(dz_s/dt) = 0$ results from letting $\varepsilon \to 0$ rather than from $dz_s/dt = 0$.

If the time scale is changed to $\tau$ using (2.1.7), equations (2.1.8) and (2.1.9) become

$$\frac{d}{d\tau} x = \varepsilon f(x,z,u,\varepsilon\tau) \qquad (2.1.12)$$

$$\frac{d}{d\tau} z = g(x,z,u,\varepsilon\tau) \ . \qquad (2.1.13)$$

Now as $\varepsilon \to 0$, equation (2.1.12) implies that x remains constant in the fast time period. Therefore, during this initial fast time period the only fast variations are in z. Accordingly,

$$z = z_f + z_s \qquad (2.1.14)$$

and thus equation (2.1.13) becomes (with $z_f = z - z_s$ and $\varepsilon \to 0$ and $dz_s/d\tau = 0$):

$$\frac{d}{d\tau} z_f = g(x^o, z_s^o + z_f(\tau), u(\tau), 0), \ z_f(0) = z^o - z_s(0) \qquad (2.1.15)$$

often called the "boundary layer system." Finally, it is recognized that (2.1.10) and (2.1.11) represent the slow model and (2.1.15) represents the fast model, with

$$x(t) \cong x_s(t) \tag{2.1.16}$$

$$z(t) \cong z_s(t) + z_f(\tfrac{t}{\varepsilon}) = z_s(t) + z_f(\tau) \ . \tag{2.1.17}$$

(From hereon, $z_f$ will be expressed in the $\tau$-domain, so that the investigation of the system characteristics at $t = 2$ seconds (say, with $\varepsilon = 0.001$) will then involve examination of $z_f$ at $\tau = 2000$ milliseconds.)

These representations (2.1.16) and (2.1.17) are merely the zero-order approximations of the asymptotic expansions in $\varepsilon$ of the solutions x and z for the system (2.1.8) and (2.1.9) [OM-3], [GA]. The solutions x and z are therein expressed as

$$x(t) = x_o(t) + \varepsilon x_1(t) + \varepsilon^2 x_2(t) + \ldots$$
$$+ \tilde{x}_o(\tau) + \varepsilon \tilde{x}_1(\tau) + \varepsilon^2 \tilde{x}_2(\tau) + \ldots \tag{2.1.18}$$

$$z(t) = z_o(t) + \varepsilon z_1(t) + \varepsilon^2 z_2(t) + \ldots$$
$$+ \tilde{z}_o(\tau) + \varepsilon \tilde{z}_1(\tau) + \varepsilon^2 \tilde{z}_2(\tau) + \ldots \ . \tag{2.1.19}$$

Thus, (with $\tilde{x}_o(\tau) \equiv 0$--see [GA, pg. 29])

$$x(t) \triangleq x_s(t) \triangleq x_o(t) \tag{2.1.20}$$

$$z(t) \triangleq z_s(t) + z_f(\tau) \triangleq z_o(t) + \tilde{z}_o(\tau) \ . \tag{2.1.21}$$

## 2.2. Exact Decomposition of the System

It is natural to ask at this point if the system (2.1.8), (2.1.9) with (2.1.6) can be decoupled into separate subsystems. In facilitating this task, the linear time-invariant matrix version of this system will be used:

$$\frac{d}{dt} x = A_{11}x + A_{12}z + B_1u \ , \ x(0) = x^o \qquad (2.2.1)$$

$$\varepsilon \frac{d}{dt} z = A_{21}x + A_{22}z + B_2u \ , \ z(0) = z^o \qquad (2.2.2)$$

$$y = C_1x + C_2z + Eu \qquad (2.2.3)$$

where

$A_{11}$ is $n \times n$ $\quad$ $A_{12}$ is $n \times m$ $\quad$ $B_1$ is $n \times r$

$A_{21}$ is $m \times n$ $\quad$ $A_{22}$ is $m \times m$ $\quad$ $B_2$ is $m \times r$

$C_1$ is $q \times n$ $\quad$ $C_2$ is $q \times m$ $\quad$ $E$ is $q \times r$

and where the argument t for x, z, u, and y has been suppressed for ease of notation. From hereon, equations (2.2.1)-(2.2.3) will be called system CS for coupled system. Kokotovic et al. [KO-1] provide an iterative scheme to separate the slow and fast subsystems, wherein the newly-determined subsystem matrices are obtained in terms of $A_{11}$, $A_{12}$, $A_{21}$, and $A_{22}$ without ill-conditioned modal transformations. And an alternative algorithm

based on the modal transformation matrices is presented in [KO-7]. Since these algorithms are cleanly presented and available in those papers, a transformation technique akin to that in [KO-4] will be discussed here.

Consider the matrix T:

$$T = \begin{bmatrix} I_1 - \varepsilon ML & -\varepsilon M \\ L & I_2 \end{bmatrix} \tag{2.2.4}$$

where L and M are any matrices of the proper sizes, along with $I_1$ and $I_2$, to yield a square matrix T. It is easily verified by checking $T^{-1}T = TT^{-1} = I_3$ that

$$T^{-1} = \begin{bmatrix} I_1 & \varepsilon M \\ -L & I_2 - \varepsilon LM \end{bmatrix} . \tag{2.2.5}$$

Assume now that L and M satisfy

$$A_{21} - A_{22}L + \varepsilon L(A_{11} - A_{12}L) = 0 \tag{2.2.6}$$

$$A_{12} - M(A_{22} + \varepsilon LA_{12}) + \varepsilon(A_{11} - A_{12}L)M = 0 . \tag{2.2.7}$$

(By checking matrix sizes, it must be that L is $m \times n$ and M is $n \times m$, $I_1$ is $n \times n$ and $I_2$ is $m \times m$, and thus $I_3$ and T are $(n + m) \times (n + m)$ matrices.) Introducing the change of variables

$$\xi_1 = x \tag{2.2.8}$$

$$\xi_2 = z + Lx = z + L\xi_1 \tag{2.2.9}$$

into system CS yields

$$\frac{d}{dt}\,\xi_1 = (A_{11} - A_{12}L)\xi_1 + A_{12}\xi_2 + B_1u \tag{2.2.10}$$

$$\varepsilon\frac{d}{dt}\,\xi_2 = (A_{21} - A_{22}L + \varepsilon LA_{11} - \varepsilon LA_{12}L)\xi_1 +$$

$$+ (A_{22} + \varepsilon LA_{12})\xi_2 + (B_2 + \varepsilon LB_1)u \tag{2.2.11}$$

$$y = (C_1 - C_2L)\xi_1 + C_2\xi_2 + Eu\ . \tag{2.2.12}$$

By using (2.2.6) in (2.2.11), the system simplifies to

$$\frac{d}{dt}\,\xi_1 = (A_{11} - A_{12}L)\xi_1 + A_{12}\xi_2 + B_1u \tag{2.2.13}$$

$$\varepsilon\,\frac{d}{dt}\,\xi_2 = (A_{22} + \varepsilon LA_{12})\xi_2 + (B_2 + \varepsilon LB_1)u \tag{2.2.14}$$

$$y = (C_1 - C_2L)\xi_1 + C_2\xi_2 + Eu\ . \tag{2.2.15}$$

Another change of variables

$$v_2 = \xi_2 \tag{2.2.16}$$

$$v_1 = \xi_1 - \varepsilon M\xi_2 = \xi_1 - \varepsilon Mv_2 \tag{2.2.17}$$

turns (2.2.13)-(2.2.15) into

$$\frac{d}{dt}\,v_1 = (A_{11} - A_{12}L)v_1 + [\varepsilon(A_{11} - A_{12}L)M + A_{12} -$$

$$- M(A_{22} + \varepsilon LA_{12})]v_2 + [B_1 - M(B_2 + \varepsilon LB_1)]u \tag{2.2.18}$$

$$\varepsilon \frac{d}{dt} v_2 = (A_{22} + \varepsilon L A_{12}) v_2 + (B_2 + \varepsilon L B_1) u \qquad (2.2.19)$$

$$y = (C_1 - C_2 L) v_1 + [\varepsilon (C_1 - C_2 L) M + C_2] v_2 + Eu \ . \qquad (2.2.20)$$

By applying (2.2.7) in (2.2.18), the simplification becomes

$$\frac{d}{dt} v_1 = (A_{11} - A_{12} L) v_1 + [B_1 - M(B_2 + \varepsilon L B_1)] u \qquad (2.2.21)$$

$$\varepsilon \frac{d}{dt} v_2 = (A_{22} + \varepsilon L A_{12}) v_2 + (B_2 + \varepsilon L B_1) u \qquad (2.2.22)$$

$$y = (C_1 - C_2 L) v_1 + [\varepsilon (C_1 - C_2 L) M + C_2] v_2 + Eu \ . \qquad (2.2.23)$$

What has thus been constructed is a state transformation

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \xi_1 - \varepsilon M \xi_2 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} x - \varepsilon M(z + Lx) \\ z + Lx \end{bmatrix} = T \begin{bmatrix} x \\ z \end{bmatrix} \qquad (2.2.24)$$

turning the coupled singularly perturbed system CS into the decoupled slow and fast subsystems (2.2.21)-(2.2.23).

Assuming that $A_{22}$ is non-singular, the choice of L and M can be made through their asymptotic expansion representation as

$$L = A_{22}^{-1} A_{21} + O(\varepsilon) \tag{2.2.25}$$

$$M = A_{12} A_{22}^{-1} + O(\varepsilon) \tag{2.2.26}$$

which, for small $\varepsilon$, satisfy equations (2.2.6) and (2.2.7).

Definition: A matrix P is of order $\varepsilon$, $O(\varepsilon)$, if there exists positive constants $\varepsilon^*$ and c such that the norm $||P||$ satisfies $||P|| \leq c\varepsilon$ for all $\varepsilon \in [0,\varepsilon^*]$.

If these two expressions for L and M are substituted in (2.2.21)-(2.2.23), the resulting system is

$$\begin{aligned}\frac{d}{dt} v_1 = {} & [A_{11} - A_{12} A_{22}^{-1} A_{21} - A_{12} O(\varepsilon)]v_1 + \\ & + \big[B_1 - (A_{12} A_{22}^{-1} + O(\varepsilon))B_2 - \\ & - \varepsilon A_{12} A_{22}^{-1}(A_{22}^{-1} A_{21} + O(\varepsilon))B_1 - \\ & - \varepsilon O(\varepsilon)(A_{22}^{-1} A_{21} + O(\varepsilon))B_1\big]u \end{aligned} \tag{2.2.27}$$

$$\begin{aligned}\varepsilon \frac{d}{dt} v_2 = {} & [A_{22} + \varepsilon A_{22}^{-1} A_{21} A_{12} + \varepsilon O(\varepsilon) A_{12}]v_2 + \\ & + [B_2 + \varepsilon A_{22}^{-1} A_{21} B_1 + \varepsilon O(\varepsilon) B_1]u \end{aligned} \tag{2.2.28}$$

$$\begin{aligned} y = {} & [C_1 - C_2(A_{22}^{-1} A_{21} + O(\varepsilon))]v_1 + \\ & + \big[\varepsilon[C_1 - C_2(A_{22}^{-1} A_{21} + O(\varepsilon)) \times \\ & \times (A_{12} A_{22}^{-1} + O(\varepsilon))] + C_2\big]v_2 + Eu \ . \end{aligned} \tag{2.2.29}$$

Defining

$$A_o \triangleq A_{11} - A_{12} A_{22}^{-1} A_{21} \tag{2.2.30}$$

$$B_o \triangleq B_1 - A_{12} A_{22}^{-1} B_2 \tag{2.2.31}$$

$$C_o \triangleq C_1 - C_2 A_{22}^{-1} A_{21} \tag{2.2.32}$$

and recalling the fact that $\tilde{A}O(\varepsilon) = O(\varepsilon)$ for any matrix $\tilde{A}$, then for small $\varepsilon$ system (2.2.27)-(2.2.29) simplifies to

$$\frac{d}{dt} v_1 = [A_o + O(\varepsilon)]v_1 + [B_o + O(\varepsilon)]u \tag{2.2.33}$$

$$\varepsilon \frac{d}{dt} v_2 = [A_{22} + O(\varepsilon)]v_2 + [B_2 + O(\varepsilon)]u \tag{2.2.34}$$

$$y = [C_o + O(\varepsilon)]v_1 + [C_2 + O(\varepsilon)]v_2 + Eu \ . \tag{2.2.35}$$

This is the $\varepsilon$-asymptotic expansion representation of the decomposed system (2.2.21)-(2.2.23). The zero-order approximation of this system is thus

$$\frac{d}{dt} v_1 = A_o v_1 + B_o u \tag{2.2.36}$$

$$\varepsilon \frac{d}{dt} v_2 = A_{22} v_2 + B_2 u \tag{2.2.37}$$

$$y = C_o v_1 + C_2 v_2 + Eu \ . \tag{2.2.38}$$

If the expressions for L and M in equations (2.2.25) and (2.2.26) are expanded [KO-4] to

$$L = A_{22}^{-1} A_{21} + \varepsilon A_{22}^{-2} A_{21} A_o + O(\varepsilon^2) \qquad (2.2.39)$$

$$M = A_{12} A_{22}^{-1} + \varepsilon(A_o A_{12} A_{22}^{-2} - A_{12} A_{22}^{-2} A_{21} A_{12} A_{22}^{-1}) + O(\varepsilon^2) , \qquad (2.2.40)$$

then the first-order approximation of the decoupled system becomes:

$$\dot{v}_1 = [A_o - \varepsilon A_{12} A_{22}^{-2} A_{21} A_o]v_1 + [B_o - \varepsilon(A_{12} A_{22}^{-2} A_{21} B_1 + A_{12} A_{22}^{-1} B_2)]u \qquad (2.2.41)$$

$$\varepsilon\dot{v}_2 = [A_{22} + \varepsilon A_{22}^{-1} A_{21} A_{12}]v_2 + [B_2 + \varepsilon A_{22}^{-1} A_{21} B_1]u \qquad (2.2.42)$$

$$y = [C_o - \varepsilon C_2 A_{22}^{-2} A_{21} A_o]v_1 + [C_2 + \varepsilon C_o A_{12} A_{22}^{-1}]v_2 + Eu \ . \qquad (2.2.43)$$

## 2.3. Approximate Decomposition of the System

Pausing for a moment, it is interesting to examine what would result by approximately decomposing system CS. This procedure is done by setting $\varepsilon = 0$ in equation (2.2.2). This yields

$$0 = A_{21}\bar{x} + A_{22}\bar{z} + B_2\bar{u} \qquad (2.3.1)$$

or

$$\bar{z} = -A_{22}^{-1}(A_{21}\bar{x} + B_2\bar{u}) \qquad (2.3.2)$$

where the bar indicates that $\varepsilon = 0$. Also

$$\bar{y} = C_1\bar{x} + C_2\bar{z} + E\bar{u} \ . \tag{2.3.3}$$

Substituting (2.3.2) into system CS leaves the slow subsystem

$$\frac{d}{dt} x_s = A_o x_s + B_o u_s \ , \ x_s(0) = x^o \tag{2.3.4}$$

$$y_s = C_o x_s + E_o u_s \tag{2.3.5}$$

where $\bar{x} = x_s$, $\bar{z}$, $\bar{y} = y_s$, and $\bar{u} = u_s$ are the slow parts of the variables x, z, y, and u, respectively, and

$$E_o \triangleq E - C_2 A_{22}^{-1} B_2 \ . \tag{2.3.6}$$

To derive the fast subsystem, it is assumed that the slow variables are constant during the fast transients, so that $d\bar{z}/dt = 0$ and $\bar{x}$ = constant during that fast period. Subtracting (2.3.1) from (2.2.2) and (2.3.3) from (2.2.3) produces

$$\varepsilon(\frac{d}{dt} z - \frac{d}{dt} \bar{z}) = A_{21}(x - \bar{x}) + A_{22}(z - \bar{z}) + \\ + B_2(u - \bar{u}) \tag{2.3.7}$$

$$y - \bar{y} = C_1(x - \bar{x}) + C_2(z - \bar{z}) + E(u - \bar{u}) \ . \tag{2.3.8}$$

Since x is predominantly slow, $x \cong \bar{x}$, and letting

$$z_f = z - \bar{z} \tag{2.3.9}$$

$$u_f = u - \bar{u} = u - u_s \tag{2.3.10}$$

$$y_f = y - \bar{y} = y - y_s \tag{2.3.11}$$

then equations (2.3.7) and (2.3.8) become

$$\varepsilon \frac{d}{dt} z_f(t) = A_{22} z_f(t) + B_2 u_f(t) ,$$

$$z_f(0) = z^o - \bar{z}(0) \tag{2.3.12}$$

$$y_f(t) = C_2 z_f(t) + Eu_f(t) . \tag{2.3.13}$$

Introducing the fast "stretching" time scale $\tau = t/\varepsilon$ produces

$$\frac{d}{d\tau} z_f(\tau) = A_{22} z_f(\tau) + B_2 u_f(\tau), \; z_f(0) = z^o - \bar{z}(0) \tag{2.3.14}$$

$$y_f(\tau) = C_2 z_f(\tau) + Eu_f(\tau) . \tag{2.3.15}$$

Notice that (2.3.1) and (2.3.4) represent the linearized time-invariant matrix versions of (2.1.11) and (2.1.10), just as (2.3.14) is to (2.1.15). And notice that (2.2.36)-(2.2.38), the decomposition via asymptotic expansion, is the same system as (2.3.4), (2.3.5), (2.3.12), and (2.3.13) with $v_1$ and $v_2$ being identified as $x_s$ and $z_f$, respectively. Looking back at (2.3.2), it becomes apparent that $\bar{z} = z_s$, and thus

$$z_s(t) = -A_{22}^{-1} A_{21} x_s(t) - A_{22}^{-1} B_2 u_s(t)$$

$$\stackrel{\Delta}{=} A_3 x_s(t) + B_3 u_s(t) \qquad (2.3.16)$$

so that

$$z_s(0) = -A_{22}^{-1} A_{21} x^o - A_{22}^{-1} B_2 u_s(0)$$

$$= A_3 x^o + B_3 u_s(0) \qquad (2.3.17)$$

where $A_3$ is m × n and $B_3$ is m × r.

It is now time to collect together the equations of the system to be examined in the remainder of this thesis. Therefore, equations (2.3.4), (2.3.5), (2.3.14), (2.3.15), and (2.3.16) will comprise that system, hereby dubbed system DS, the decoupled system.

## 2.4. Properties of the System

There is enough foundation at this point to discuss properties of the singularly perturbed system. The system CS has already been shown to possess a two-time scale characteristic. This effect is evident in the eigenvalue structure of that system.

Lemma 1: Suppose $A_{22}^{-1}$ exists and has all L.H.P. eigenvalues, none on the imaginary axis. Then, as $\varepsilon \to 0$, the first n eigenvalues of system CS tend to the eigenvalues of the reduced system

(2.3.4), while the remaining m eigenvalues tend to infinity as the eigenvalues of $\frac{1}{\varepsilon} A_{22}$ .

Proof: By rewriting equations (2.2.33) and (2.2.34) as

$$\frac{d}{dt}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} A_o + \mathcal{O}(\varepsilon) & 0 \\ 0 & \frac{1}{\varepsilon}A_{22} + \frac{1}{\varepsilon}\mathcal{O}(\varepsilon) \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} B_o + \mathcal{O}(\varepsilon) \\ \frac{1}{\varepsilon}B_2 + \frac{1}{\varepsilon}\mathcal{O}(\varepsilon) \end{bmatrix} u$$

$$\triangleq Av + Bu \qquad (2.4.1)$$

it is clear that the eigenvalues of system CS are contained in the eigenvalues of this system matrix A--which consist of the eigenvalues of $A_o + \mathcal{O}(\varepsilon)$ and the eigenvalues of $\frac{1}{\varepsilon} A_{22} + \frac{1}{\varepsilon} \mathcal{O}(\varepsilon)$ . (Also see [KO-4].) Q.E.D.

Thus, system DS consists of two subsystems: the slow subsystem containing n small eigenvalues (in magnitude) and a fast subsystem with m large eigenvalues. And the smaller $\varepsilon$ is, the greater is the separation of these two groups of eigenvalues. In an asymptotically stable system the fast modes corresponding to the large eigenvalues are important only during a short period (measured in $\tau$-units). And after that period those modes become negligible and the behavior of the system can be described merely by its slow modes (using the t-domain). (This is related to the concepts of Dominant Pole

Theory [SH] which holds that the system eigenvalues of small magnitude dominate the system behavior.) Neglecting the fast modes (parasitics) is equivalent to assuming that they are infinitely fast; that is, allowing $\varepsilon \to 0$ in system CS.

The last paragraph mentioned the concept of stability. Basically, a system is <u>asymptotically stable</u> if when the system is started near an equilibrium point, the state of the system approaches that equilibrium point as $t \to \infty$, where an <u>equilibrium point</u> is a constant vector solution of the state differential equation.

<u>Theorem 2.1</u>: If the real parts of all eigenvalues of $A_o$ and of $A_{22}$ are negative, then there exists an $\varepsilon^* > 0$ such that for all $\varepsilon < \varepsilon^*$ the system CS is asymptotically stable.

<u>Proof</u>: Referring to (2.4.1), the system matrix there becomes, for sufficiently small $\varepsilon$, $\begin{bmatrix} A_o & 0 \\ 0 & \frac{1}{\varepsilon} A_{22} \end{bmatrix}$ and thus, if eigenvalues of $A_o$ and $A_{22}$ are in the left-half of the complex plane then the system CS is asymptotically stable. This can be considered also as $\sigma(A_o + 0(\varepsilon)) = \sigma(A_o) + 0(\varepsilon)$ and $\sigma(\frac{1}{\varepsilon} A_{22} + \frac{1}{\varepsilon} 0(\varepsilon)) = \frac{1}{\varepsilon} \sigma(A_{22}) + \frac{1}{\varepsilon} 0(\varepsilon)$, and thus for small positive $\varepsilon$ these spectra become simply $\sigma(A_o)$ and $\frac{1}{\varepsilon} \sigma(A_{22})$, respectively. (Here $\sigma(P)$ stands for the spectrum of P--the set of all

eigenvalues of P.) Hence, if $\sigma(A_o)$ and $\sigma(A_{22})$ are in the left-half plane then the system CS is asymptotically stable. Q.E.D.

Thus, stability of system DS implies stability of system CS. Other discussions on stability and stabilization can be found in [KO-2], [KO-4], [PO-1], [WI], and [GRU].

Controllability of the system CS can now be established, too.

Definition: A pair of matrices (A,B) is a controllable pair (and thus the system $\dot{X} = AX + BU$ is controllable) if rank $[B,AB,A^2B,\ldots,A^{n-1}B] = n$, where A is $n \times n$ and B is $n \times r$ and rank $A = n$.

Theorem 2.2: If $A_{22}^{-1}$ exists, and if the pairs $(A_o,B_o)$ and $(A_{22},B_2)$ are controllable pairs, then there exists an $\varepsilon^* > 0$ such that for all $\varepsilon < \varepsilon^*$ the system CS is controllable.

Proof: From (2.4.1) it follows that for $\varepsilon$ small the controllability of the reduced and boundary layer systems, that is of the pairs $(A_o,B_o)$ and $(A_{22},B_2)$, implies the controllability of the original system CS. (That is, the subsystem (2.2.33) is a regular perturbation of the reduced system (2.2.36) and the subsystems (2.2.33)

and (2.2.34) are connected through u, but have different eigenvalues.) (See also [KO-2], [KO-4].) Q.E.D.

Thus, controllability of system DS implies controllability of system CS.

It should be noted here that a matrix K exists such that $A_{22} + B_2K$ is non-singular. And the controllability of the system CS is not influenced by $u = Kz + w$. Thus, even if $A_{22}^{-1}$ doesn't exist, Theorem 2.2 still holds, but with the matrix $A_{22} + B_2K$ replacing $A_{22}$ in the definition of $A_o$ and $B_o$ in equations (2.2.30) and (2.2.31).

The last concept to be discussed in observability.

Definition: A pair of matrices (A,C) is an observable pair (and thus the system $\dot{X} = AX + BU$, $Y = CX + EU$ is observable) if the rank $[C,CA,CA^2 \ldots CA^{n-1}]^T = n$, where A is $n \times n$ and C is $q \times n$ and rank A = n.

An analogous argument leads to the proof of the last important theorem:

Theorem 2.3: If the pairs $(A_o,C_o)$ and $(A_{22},C_2)$ are observable, then there exists an $\varepsilon^* > 0$ such that for all $\varepsilon < \varepsilon^*$ the system CS is observable.

Thus, observability of system DS implies observability of system CS. (For additional reading on observability see Javid [JA].)

In summary, the formalized system DS is compiled here:

Slow Decoupled Subsystem (SDSS):

$$\frac{d}{dt} x_s(t) = A_o x_s(t) + B_o u_s(t), \; x_s(0) = x^o \in R^n$$

$$y_s(t) = C_o x_s(t) + E_o u_s(t)$$

$$z_s(t) = A_3 x_s(t) + B_3 u_s(t)$$

Fast Decoupled Subsystem (FDSS):

$$\frac{d}{d\tau} z_f(\tau) = A_{22} z_f(\tau) + B_2 u_f(\tau), \; z_f(0) = z^o - z_s(0) \in R^m$$

$$y_f(\tau) = C_2 z_f(\tau) + E u_f(\tau) \; .$$

## 2.5. The Identification Problem

It is now of interest to examine these two decoupled subsystems with respect to parameter estimation concepts. The problem under investigation can now be stated simply as:

> Given à priori knowledge that a system exhibits the behaviors characteristic of slow and fast phenomena, determine the "inner workings" of that system from the input and output data records available.

What do the words "inner workings" refer to? In the present case of this problem, it is initially assumed that the process under scrutiny is of the bi-structural form of system DS. The "inner workings" of that system are then the internal mechanisms as defined by the system matrices and the time-scale parameter $\varepsilon$. What they describe are the functionings of the decomposed slow and fast subsystems.

So once the matrices $A_o$, $B_o$, $C_o$, $E_o$, $A_3$, $B_3$, $A_{22}$, $B_2$, $C_2$, E, and the initial values $x^o, z^o$ and the parameter $\varepsilon$ are known, then the system DS is totally describable, and is then ready for further explorations such as in the area of optimal control. Therefore, for the remainder of this thesis, the goal will be the determination of these parametric quantities.

In order to discover these quantities, it will be necessary to choose the proper experimental design [GOO], [IS]. This includes the selection of the input signals [LE], for care need be taken to use inputs which will act to "excite" all the fast and/or slow states so that accurate determination of the parameters will be made.

Also an appropriate identification scheme must be used which: (a) has good discriminating ability in order to identify the faster components over the slower ones; (b) is "good" in the sense that it yields a model

consistent with the data; and (c) yields estimates which converge, in some statistical sense, even in the presence of noise. Finally, since the system DS is operating in two time scales, it is important to consider relevant sampling time(s) on the process to be identified.

It will be the intent of the next chapter to utilize identification theory to solve the problem of estimating the parameters of the decoupled singularly perturbed systems SDSS and FDSS from a process operating in coupled form (CS).

# CHAPTER III

## PROBLEM SOLUTION

The scope of this chapter will be to provide the theoretical solution to the problem of identifying the decoupled singularly perturbed subsystems. The first section will take into account the salient characteristics of the singularly perturbed structure and discuss what type of an identification method might be used to exploit these features. Also discussed in that section is model representation. This is expanded in Section 3.2 where the models for the identification-solution method are dealt with. The algorithms involved in the identification procedure are unveiled in the next section, and the last section presents the application of the identification method to the specific problem of a singularly perturbed system.

### 3.1. Considerations for a Solution to the Problem

At this point, let us examine system DS again:

$$\underline{\text{SDSS}}: \quad \frac{d}{dt} x_s(t) = A_o x_s(t) + B_o u_s(t); \; x_s(0) = x(0) \qquad (3.1.1)$$

$$y_s(t) = C_o x_s(t) + E_o u_s(t)$$

$$z_s(t) = A_3 x_s(t) + B_3 u_s(t)$$

$$\underline{\text{FDSS}}: \quad \varepsilon \frac{d}{dt} z_f(t) = A_{22} z_f(t) + B_2 u_f(t);$$

$$z_f(0) = z(0) - z_s(0)$$

$$y_f(t) = C_2 z_f(t) + E u_f \qquad (3.1.2)$$

The block diagram of the mechanics of the singularly perturbed process may be seen in Figure 3.1. In this

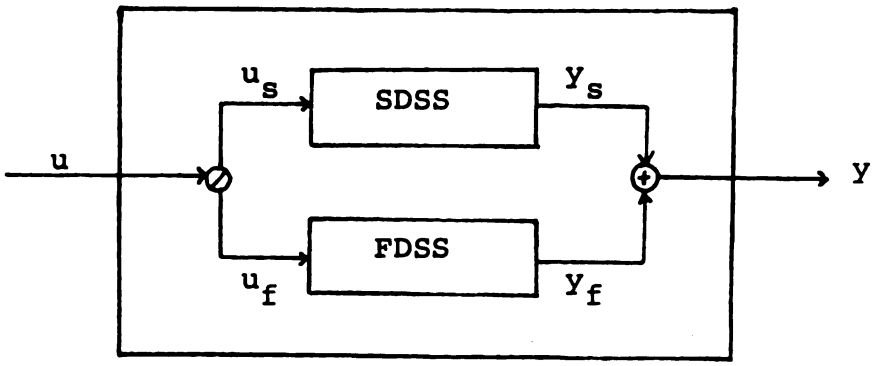



Figure 3.1. Singularly Perturbed Process

figure the coupled system CS is visualized as two decomposed subsystems operating in parallel. What is desired is some method and set of procedures for determining the dynamics of each subsystem from the input-output data information that is given.

Before addressing the issue of the method of identification, it is relevant to note something about the model to be identified. Up to this point, the systems

of equations under consideration have been represented in state space equation forms. Since, in the framework of identification, models of the fast and slow subsystems are to be found, it is totally reasonable to try to determine any structural format as long as it is equivalent in an input-output sense.

The following are two different representations of the same observable system. The first is the observable input-output canonical form [GU], [BE] for a multi-input, multi-output (MIMO) system:

$$\tilde{P}(D)y(t) = \tilde{Q}(D)u(t) \ . \qquad (3.1.3)$$

$\tilde{P}(D)$ is a square non-singular $(q \times q)$ polynomial matrix in the differential operator D and $Q(\tilde{D})$ is a $(q \times r)$ polynomial matrix in D, and u(t), y(t) are $(r \times 1)$ input and $(q \times 1)$ output vector functions, respectively. For the case of a single-input, single-output (SISO) system, this can be expressed in a scalar linear time-invariant differential equation:

$$y^{(n)}(t) + a_{n-1}\, y^{(n-1)}(t) + \ldots a_o y(t) =$$

$$= b_n u^{(n)}(t) + b_{n-1}\, u^{(n-1)}(t) + \ldots + b_o u(t) \ . \qquad (3.1.4)$$

The second representation is the observable companion form in state space form:

$$\frac{d}{dt} x(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Eu(t) \qquad (3.1.5)$$

The transformation from (3.1.5) to (3.1.3) can be divided into two steps [BE]. In the first step, by eliminating state vector x(t) from equation (3.1.5), an equivalent representation of the form (3.1.3) is obtained. During the second step, a unimodular matrix is formed, with which the representation obtained in the first step can be transformed to the desired input-output canonical form satisfying certain requirements on the degrees of the element polynomials in $\tilde{P}(D)$ and $\tilde{Q}(D)$. The transformation from (3.1.3) to (3.1.5) can be obtained either through the Structure Theorem [WOL] or by the algorithm developed by Guidorzi [GU]. Other transformation procedures between the two representations can be found in Ogata [OG], and algorithms on the relationship of the initial conditions between the two forms are dealt with by Heinen [HE]. With these equivalency considerations, it is therefore legitimate to choose to find either the differential equations (3.1.3) describing the system behavior or the state space description (3.1.5) of that behavior.

As to what kind of method would be suited to the problem at hand, a major consideration is the fact that the two subsystems operate with different time constants.

This difference manifests itself through the effect of the fast subsystem dynamics. Since the fast dynamics dissipate rapidly (on order $\frac{1}{\epsilon}$ times faster than the slow dynamics), a key concern should be a procedure which can identify a continuous-time system in a limited time frame.

Of the multitude of schemes in the literature, several methods show promise of accomplishing identification in a finite time period. Obviously, though, any identification attempted--in particular, adaptive control procedures--are performed in a finite time interval, even if they are only theoretically valid on the infinite time frame. There exists a well-defined procedure of identification which has been proved to be valid and successful on a finite time interval [PEA-1,2], which will be discussed at length later.

In most identification methods, the initial conditions of the system have to be determined along with the system parameters, even though it is the set of system parameters that is of primary interest. With the problem at hand--of identifying the fast and slow subsystems--finding the initial conditions for each separate subsystem adds to the complexities of the problem. The basic issue is to determine the system dynamics of FDSS and SDSS. The same afore-mentioned finite-time procedure (called

*H*-identification), due to Pearson, uses a noncausal filter which eliminates the initial conditions during the identification process so that the system parameters can be identified alone.

## 3.2. Model of a System to be Identified

Since *H*-identification will be used in this dissertation, it is now time to examine more closely this procedure and how it relates to the singularly perturbed system identification problem. This procedure is a least squares equation error parameter identification technique (see Figure 3.2), but differs from other known applications of least squares in a number of ways. The first of these is, as mentioned above, that only input-output

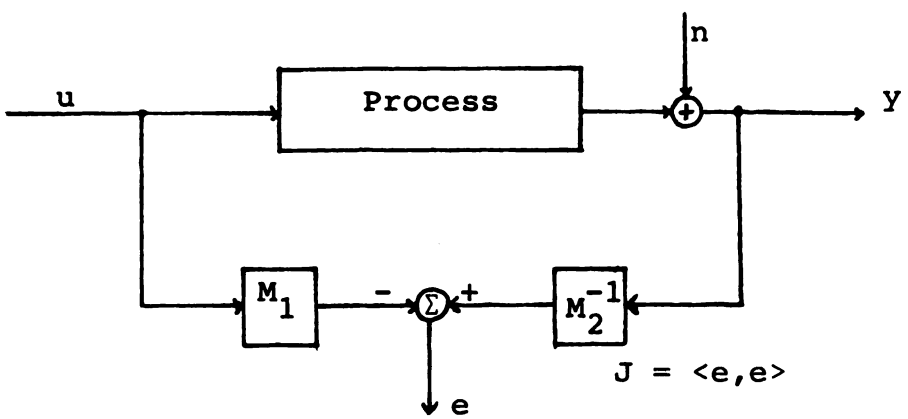



Figure 3.2. Least-Squares Equation Error Model

data is presumed to be given over a fixed finite time interval with no attempt to estimate unknown initial conditions. The second characteristic of *H*-identification is sufficiently general to include a variety of nonlinear, time varying, differential delay, possible unstable, multivariable system models. The next is that the formulation leads to an explicitly defined function of the parameters which simplifies the computations significantly. Also, this approach is a "one shot" identification scheme, as opposed to other methods which are iterative in time. The last, and most germane feature, is the way in which the unknown disturbances are modeled on the finite observation time interval. While Maximum Likelihood and other statistical methods of identification represent the disturbances by stochastic processes with underlying Markov process representations, the model for unknown disturbance signals in this approach is the deterministic homogeneous differential operator equation:

$$T(D,\delta)d(t) = \sum_{i=0}^{\alpha} \delta_i D^{\alpha-i} d(t) = 0 ,$$

$$\delta_o \overset{\Delta}{=} 1, \; 0 \leq t \leq t_1 < \infty \qquad (3.2.1)$$

(with the order being $\alpha \in [0,\alpha_{max}]$, $\alpha_{max}$ preselected) where the $\delta_i$'s and the initial conditions are completely arbitrary. That is, the disturbances can be approximated

by the arbitrary solution of a homogeneous ordinary differential equation on a specified finite time interval. Actually, this model can be regarded as generating a stochastic process if the $\delta_i$'s and the initial conditions, $d^{(i)}(0)$ $(i = 1,\ldots,\alpha)$, comprise $2\alpha$ independent random variables with essentially infinite variances. The above model is actually quite suitable, since the data set is presumed to consist of input-output data observed on a finite observation time interval. Thus, the shorter the time interval, the more reasonable is the above disturbance model for a modest value of $\alpha$. With respect to the finite time-interval length, it has been verified by simulation studies [PEA-2,3] that this time interval can be surprisingly short in many cases; i.e., on the order of the dominant system time constant, or less.

At this juncture, it is appropriate to introduce the model formulation for the identification procedure. To refresh the memory, equation (3.1.3) is rewritten as:

$$\tilde{P}(D)y(t) + \tilde{Q}(D)u(t) = 0 \ , \ 0 \leq t \leq t_1 \ . \qquad (3.2.2)$$

Since the system parameters are contained within $\tilde{P}(D)$ and $\tilde{Q}(D)$, it would be best to express them in terms of the parameters in question as:

$$\tilde{P}(D,\omega) + \tilde{Q}(D,\omega)u(t) = 0 \ , \ 0 \leq t \leq t_1 \qquad (3.2.3)$$

where

$$\tilde{P}(D,\omega) = \sum_{i=0}^{n} \tilde{P}_i(\omega) D^{n-i} \qquad (3.2.4)$$

$$\tilde{Q}(D,\omega) = \sum_{i=0}^{n} \tilde{Q}_i(\omega) D^{n-i} \qquad (3.2.5)$$

and $D \triangleq \frac{d}{dt}$ with $\omega = (\omega_1,\ldots,\omega_\beta)$ being the vector of system parameters. By defining a vector valued function $f(\omega)$ with components $f_i(\omega)$ selected to reflect the ways in which the parameters enter into $\tilde{P}$ and $\tilde{Q}$, it is then easy to define $v(t)$ and $V(t)$ (depending on the data pair $[u(t),y(t)]$) and operators P and Q such that equation (3.2.3) becomes

$$P(D)v(t) + Q(D)V(t)f(\omega) = 0 \ , \ 0 \leq t \leq t_1 \ . \qquad (3.2.6)$$

This is the case for systems which are <u>separable in the parameters</u>--as are all linear systems--wherein the generic decomposition of equation (3.2.3):

$$P(D)v(t) + Q(D)g(t,\omega) = 0 \ , \ 0 \leq t \leq t_1 \qquad (3.2.7)$$

admits to equation (3.2.6) via

$$g(t,\omega) \equiv V(t)f(\omega) \ . \qquad (3.2.8)$$

Here $V(t)$ is a matrix valued function of the data and $f(\cdot)$ is a continuously differentiable vector-valued function of $\omega$ with the single valued property

$$f(\omega) = f(\omega^*) \text{ if and only if } \omega = \omega^* \qquad (3.2.9)$$

for all $\omega$ and $\omega^*$.

Now, if the actual input $\tilde{u}(t)$ and output $\tilde{y}(t)$ are corrupted by additive disturbances $d_1(t)$ and $d_2(t)$, respectively, so that $u(t)$ and $y(t)$ are observed according to:

$$y(t) = \tilde{y}(t) + d_1(t) \ , \ 0 \leq t \leq t_1 \qquad (3.2.10)$$

$$u(t) = \tilde{u}(t) + d_2(t) \ , \ 0 \leq t \leq t_1 \qquad (3.2.11)$$

then the model (3.2.3) becomes

$$\tilde{P}(D,\omega)[y(t) - d_1(t)] + \tilde{Q}(D,\omega)[u(t) - d_2(t)] = 0 \ ,$$

$$0 \leq t \leq t_1 \ . \qquad (3.2.12)$$

And if $d_1(t)$, $d_2(t)$ are assumed to be solutions of the differential equation (3.2.1) on $[0,t_1]$, then operating on both sides of equation (3.2.12) with $T(D,\delta)$ yields:

$$T(D,\delta)\tilde{P}(D,\omega)y(t) + T(D,\delta)\tilde{Q}(D,\omega)u(t) = 0 \ , \ 0 \leq t \leq t_1 \ . \qquad (3.2.13)$$

This is analogous to

$$T(D,\delta)P(D)v(t) + T(D,\delta)Q(D)V(t)f(\omega) = 0 \ , \ 0 \leq t \leq t_1 \qquad (3.2.14)$$

by following the same decomposition scheme that transformed equation (3.2.3) into equation (3.2.6). By expanding out $T(D,\delta)$, equation (3.2.14) takes the form:

$$D^{\alpha}P(D)v(t) + \delta_1 D^{\alpha-1}P(D)v(t) +\ldots+ \delta_{\alpha}P(D)v(t) +$$

$$+ D^{\alpha}Q(D)V(t)f(\omega) + \delta_1 D^{\alpha-1}Q(D)V(t)f(\omega) +\ldots+$$

$$+ \delta_{\alpha}Q(D)V(t)f(\omega) = 0 \ . \qquad (3.2.15)$$

Writing this in vector form yields:

$$D^{\alpha}P(D)v(t) + \left[D^{\alpha-1}P(D)v(t),\ldots,P(D)v(t),D^{\alpha}Q(D)V(t),\right.$$

$$\left.D^{\alpha-1}Q(D)V(t),\ldots,Q(D)V(t)\right] \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_{\alpha} \\ f(\omega) \\ \delta_1 f(\omega) \\ \vdots \\ \delta_{\alpha} f(\omega) \end{bmatrix} = 0 \ . \qquad (3.2.16)$$

This can be simplified into the following model form:

$$\overline{P}(D)v(t) + \overline{Q}(D)\overline{V}(t)\overline{f}(\theta) = 0 \ , \ 0 \leq t \leq t_1 \qquad (3.2.17)$$

where

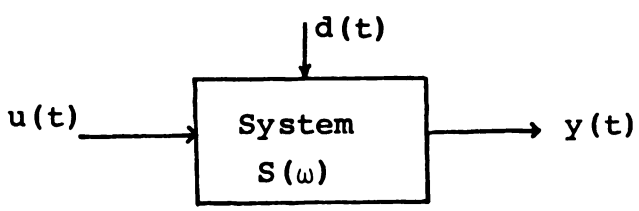
$$\theta = (\delta_1,\ldots,\delta_{\alpha},\omega_1,\ldots,\omega_{\beta}) \ . \qquad (3.2.18)$$

The vector function $\overline{f}(\theta)$ satisfies the same single-valued property (3.2.9) if the original function $f(\omega)$ in (3.2.6) does. This should be the case with a model that has been properly parametrized.

To summarize, the basic model including disturbances is represented in equation (3.2.17). This is, of course, valid only for models which are separable in the parameters. Otherwise, it would take the generic form:

$$\bar{P}(D)v(t) + \bar{Q}(D)\bar{g}(t,\theta) = 0 \ , \ 0 \leq t \leq t_1 \qquad (3.2.19)$$

This generic model can be viewed in Figure 3.3.




$$\bar{P}(D)v(t) + \bar{Q}(D)\bar{g}(t,\theta) = 0$$

$$\theta = (\delta,\omega)$$

Figure 3.3. Basic Model

## 3.3. The Algorithms of *H*-identification

Having established the basic model to be identified, it is time to deal with the algebraic mechanics behind *H*-identification.

Definition: The basic model (3.2.19) is *H*-identifiable if and only if the parameter vector $\theta$ can be identified via the input u(t) and the output y(t) on a finite time interval $[0,t_1]$

without estimating (implicitly or explicitly) the initial condition of the model.

Now, let a square non-singular polynomial matrix F(D) be chosen such that

$$F^{-1}(D)[\bar{P}(D), \bar{Q}(D)] \tag{3.3.1}$$

is a proper transfer function matrix. The form of F(D) is

$$F(D) = \sum_{i=0}^{m} F_i D^{m-i} \ , \ m \geq n \ . \tag{3.3.2}$$

Then an auxiliary error function is implicitly designated via

$$F(D)z(t,\theta) = \bar{P}(D)v(t) + \bar{Q}(D)\bar{g}(t,\theta) \ . \tag{3.3.3}$$

To get a better handle on the nature of $z(t,\theta)$, it is wise to first examine the homogeneous solution to equation (3.3.3):

$$F(D)z(t) = 0 \ . \tag{3.3.4}$$

The solution to this can be expressed as:

$$\frac{d}{dt} x(t) = Ax(t) \ , \ x(0) = x_o \ \varepsilon \ R^{\bar{n}}$$

$$z(t) = Cx(t) \tag{3.3.5}$$

where (A,C) is an appropriate observable pair with minimal dimension state space $\bar{n}$. Then z(t) takes the form:

$$z(t) = Ce^{At}x_o \ . \tag{3.3.6}$$

Therefore, the particular solution to equation (3.3.3) will take the form:

$$z(t,\theta) = Ce^{At}x_o + \nu(t) + \mu(t,\theta) \ , \ 0 \leq t \leq t_1 \ . \tag{3.3.7}$$

By operating on both sides of this by F(D), it is seen that the particular solutions $\nu(t)$ and $\mu(t,\theta)$ are the zero state solutions to

$$F(D)\nu(t) = \bar{P}(D)v(t) \tag{3.3.8}$$

$$F(D)\mu(t,\theta) = \bar{Q}(D)\bar{g}(t,\theta) \ , \tag{3.3.9}$$

respectively. In the case of separability in the parameters, the particular solution $z(t,\theta)$ would be:

$$z(t,\theta) = Ce^{At}x_o + \nu(t) + M(t)\rho(\theta) \tag{3.3.10}$$

whereupon M(t) would then be found (through the degeneracy of equation (3.3.9)) as the zero state solution to:

$$F(D)M(t) = \bar{Q}(D)\bar{V}(t) \ . \tag{3.3.11}$$

However, for either case, the particular solution $z(t,\theta)$ contains unknown parameters $\theta$ and $x_o$. Since it is desired

to identify $\theta$ without actually estimating $x_o$, the term $Ce^{At}x_o$ need be eliminated. This can be accomplished via an annihilation filter $H$ [PEA-1,2].

Laying some groundwork first, let T denote the Hilbert space of all vector valued square integrable functions on $[0,t_1]$. And let $v(t)$ and $\bar{g}(t,\theta)$ range over the space of piecewise continuous functions on $[0,t_1]$. Also, let $T_o$ denote the subspace of T containing solutions to equation (3.3.6). That is,

$$T_o = \{\chi(t) \,|\, \chi(t) = Ce^{At}x_o,\ x_o \in R^{\bar{n}},\ 0 \leq t \leq t_1\} \ . \tag{3.3.12}$$

Definition: The filter $H$ is a linear operator with domain T and range $(T - T_o)$ with the property:

$$\tilde{\psi}(t) \triangleq H(\psi(t)) \triangleq \int_o^{t_1} [H(t,\tau)]\psi(\tau)d\tau \triangleq$$

$$\triangleq \int_o^{t_1} [I\delta(t-\tau) - Ce^{At}W^{-1}e^{A^T\tau}C^T]\psi(\tau)d\tau =$$

$$= \psi(t) - Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T\psi(\tau)d\tau \tag{3.3.13}$$

where (A,C) is the observable pair for the system in equation (3.3.5) and

$$W \triangleq \int_o^{t_1} e^{A^T t}C^T Ce^{At}dt \tag{3.3.14}$$

is the observability Gramian for that same system and--only here--$\delta$ is the Dirac function.

Now, $H$ is a self-adjoint projection operator, as can readily be seen from $H\left[H(\psi(t))\right] = H(\psi(t))$ and $\text{adj}\left[H(\psi(t))\right] = H(\psi(t))$ (see Appendix A). But the significance to the filter $H$ is that its null space is $T_o$; that is,

$$H(Ce^{At}x_o) \equiv 0 \ , \ \forall \ x_o \in R^{\overline{n}} \ , \ 0 \leq t \leq t_1 \ . \qquad (3.3.15)$$

Thus, operating on the solution $z(t,\theta)$ in equation (3.3.7) yields:

$$\tilde{z}(t,\theta) \stackrel{\Delta}{=} H(z(t,\theta)) = H(\nu(t)) + H(\mu(t,\theta)) \stackrel{\Delta}{=}$$

$$\stackrel{\Delta}{=} \tilde{\nu}(t) + \tilde{\mu}(t,\theta) \ , \ 0 \leq t \leq t_1 \ . \qquad (3.3.16)$$

Therefore, since the initial condition response is arbitrary and has no physical significance, the solution $z(t,\theta)$ is projected down into the subspace $(T - T_o)$, via $H$, thus annihilating the initial condition response on $[0,t_1]$. In so doing, $\nu(t)$ and $\mu(t,\theta)$ are also projected down into that same subspace yielding:

$$\tilde{\nu}(t) = \nu(t) - Ce^{At}W^{-1} \int_o^{t_1} e^{A^T\tau}C^T\nu(\tau)d\tau \qquad (3.3.17)$$

$$\tilde{\mu}(t,\theta) = \mu(t,\theta) - Ce^{At}W^{-1} \int_o^{t_1} e^{A^T\tau}C^T\mu(\tau,\theta)d\tau \ . \qquad (3.3.18)$$

Now, *H*-identification minimizes the inner product norm of equation (3.3.16) yielding the functional $J_1(\theta)$ for the least squares minimization problem:

$$J_1(\theta) = \langle \tilde{z}(t,\theta), \tilde{z}(t,\theta) \rangle = \int_o^{t_1} \tilde{z}^T(t,\theta)\tilde{z}(t,\theta)dt \ . \tag{3.3.19}$$

Thus, any value of $\theta$ which satisfies the basic model (3.2.19) is also a solution to

$$J_1(\theta) = 0 \ . \tag{3.3.20}$$

Conversely, any value of $\theta$ which satisfies equation (3.3.20) is a candidate for a value of $\theta$ satisfying the basic model (3.2.19).

By substituting equations (3.3.16)-(3.3.18) into equation (3.3.19), $J_1(\theta)$ unfolds as:

$$J_1(\theta) = \int_o^{t_1} \nu^T(t)\nu(t)dt + 2\int_o^{t_1} \nu^T(t)\mu(t,\theta)dt + \int_o^{t_1} \mu^T(t,\theta)\nu(t)dt - \eta^T W^{-1}\eta - 2\eta^T W^{-1}\gamma(\theta) - \gamma^T(\theta)W^{-1}\gamma(\theta) \tag{3.3.21}$$

where

$$\eta = \int_o^{t_1} e^{A^T t} C^T \nu(t)dt \tag{3.3.22}$$

$$\gamma(\theta) = \int_o^{t_1} e^{A^T t} C^T \mu(t,\theta)dt \ . \tag{3.3.23}$$

In the case of separable-in-the-parameters models,

$$\mu(t,\theta) = M(t)\rho(\theta) \tag{3.3.24}$$

so that

$$\gamma(\theta) = N\rho(\theta) \tag{3.3.25}$$

where

$$N = \int_o^{t_1} e^{A^T t} C^T M(t)\,dt \tag{3.3.26}$$

The functional $J_1(\theta)$ can then reduce to an explicit function of $\theta$:

$$J_2(\theta) = a + 2b^T\rho(\theta) + \rho^T(\theta)\Phi\rho(\theta) \tag{3.3.27}$$

where

$$a = \int_o^{t_1} \nu^T(t)\nu(t)\,dt - \eta^T W^{-1}\eta \tag{3.3.28}$$

$$b = \int_o^{t_1} M^T(t)\nu(t)\,dt - N^T W^{-1}\eta \tag{3.3.29}$$

$$\Phi = \int_o^{t_1} M^T(t)M(t)\,dt - N^T W^{-1}N \quad . \tag{3.3.30}$$

Thus, once the data is collected and $(a,b,\Phi)$ are found, no further integrations are needed involving the data over $[0,t_1]$. It is left to just minimize $J_1(\theta)$ or $J_2(\theta)$ with respect to $\theta$.

Theorem 3.1: A minimizing value $\theta^*$ for the positive definite $J_2(\theta)$ is a least squares estimate of $\theta$ which is unique if (as a sufficient condition) the data makes $\Phi$ positive definite, which occurs if the columns of $\bar{Q}(D)\bar{V}(t)$ are linearly independent functions on $[0,t_1]$.

Proof: By letting $\lambda = \rho(\theta)$, due to its single-valued property, and setting $J_2(\theta)$ as:

$$\bar{J}_2(\lambda) = a + 2b^T\lambda + \lambda^T\Phi\lambda \ , \qquad (3.3.31)$$

then the minimization becomes equivalent to

$$\frac{1}{2}\nabla\bar{J}_2(\lambda) = b + \Phi\lambda = 0 \qquad (3.3.32)$$

which is the normal equation for

$$\tilde{z}(t,\theta) = \tilde{v}(t) + \tilde{M}(t)\lambda \ , \ 0 \leq t \leq t_1 \ . \qquad (3.3.33)$$

A unique solution to this normal equation (3.3.32) is found if and only if the columns of $\tilde{M}(t)$ are linearly independent on $[0,t_1]$. For then $\tilde{M}(t)$ has full rank, and thus $\Phi$ is positive definite and therefore non-singular, allowing a unique solution to the normal equation [SEB]. The subspace $(T - T_o)$ contains the columns of $\tilde{M}(t)$, which can be represented as the projection of the function $F^{-1}(D)\bar{Q}(D)\bar{V}(t)\lambda$ in that subspace. Since $T_o$ is the null space for $F(D)$, it follows that linear dependence, or independence, of the columns of $\tilde{M}(\cdot)$ cannot be altered

by operating on that projection of $F^{-1}(D)\bar{Q}(D)\bar{V}(t)\lambda$ with $F(D)$. Q.E.D.

Thus, $\Phi$ is non-singular if the columns of $\bar{Q}(D)\bar{V}(t)$ are linearly independent functions. However, this is mainly of theoretical interest since it is not assumed that the data is differentiable.

A final point concerning *H*-identification is that the theory is still valid for any initial time $t_o$ ($0 \leq t_o < t_1$), whereby any reference to $t = 0$ in the algorithm is replaced by $t = t_o$.

## 3.4. Identifying Decoupled Subsystems via *H*-identification

The aim of this section is to describe how *H*-identification is used to identify the decoupled subsystems FDSS and SDSS. Re-examination of Figures 3.1 and 3.3 will help to facilitate this. Their combination is demonstrated in Figure 3.4.

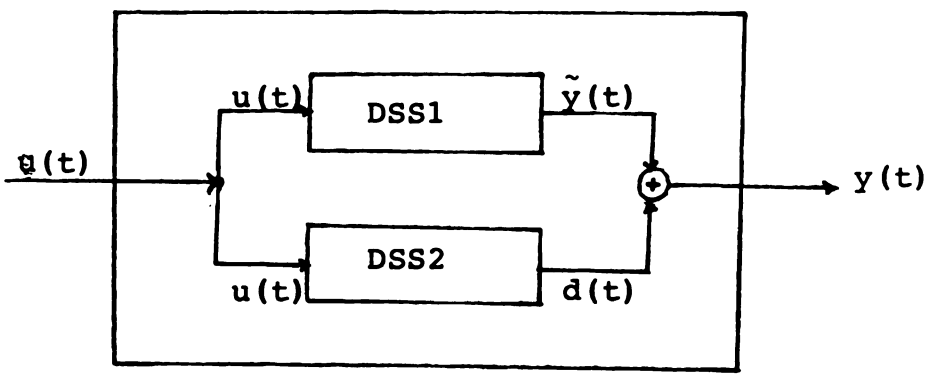



Figure 3.4. Basic Identification Model for Singularly Perturbed Systems.

In Figure 3.4, one of the decoupled subsystems (DSS1) is considered as the main system containing the parameter vector $\omega$ to be identified, whereas the second decoupled subsystem (DSS2) is considered as the disturbance model (containing parameter vector $\delta$) with output d(t). These two mechanisms are acting in parallel, and the observed output y(t) is as in equation (3.2.10):

$$y(t) = \tilde{y}(t) + d(t) \ . \tag{3.4.1}$$

This matches equation (2.3.11) in that

$$y(t) = y_s(t) + y_f(t) \ ; \tag{3.4.2}$$

that is, there is equivalency between the two sets of signals via:

$$\{\tilde{y}(t),\ d(t)\} = \{y_s(t),\ y_f(t)\} \tag{3.4.3}$$

provided there is no other outside noise disturbances acting on the system CS or its component subsystems.

That *H*-identification should apply well here depends on several factors. First, and most significant, is that the output of the disturbance subsystem, d(t), is in fact a solution of a homogeneous differential equation. Without loss of generality, suppose that FDSS is the subsystem DSS2 considered as the disturbance mechanism. Examining the differential equations for FDSS as seen in

equation (3.1.2) shows that its output $y_f$ satisfies equation (3.2.1) as d(t) where the order $\alpha$ of $T(D,\delta)$ is at least as large as the sum of the order of $z_f$ and $u_f$. Therefore, $d(t) \stackrel{\Delta}{=} y_f(t)$ will have some $T(D,\delta)$ in existence to annihilate it. Secondly, since the goal is to identify --not one--but two (sub)systems, the need to determine the initial conditions of each is eliminated via the filter $H$, thus alleviating such difficulties. Thirdly, of course, is that $H$-identification has been proved to be quite successful [PEA-1,2,4] over very small time intervals $[0,t_1]$. This is of great significance to the problem of identifying the fast subsystem since the effects of FDSS die out quite rapidly, necessitating the need for such a "fast" algorithm.

At this point it might seem that $H$-identification, certainly used in the vein of points one and two just above, is application to any system that admits to a decoupling into subsystems. This might be the case; however, the application in this thesis of $H$-identification to the particular problem of the decoupled singularly perturbed system is special in the following way: the majority of simulation studies carried out by Pearson et al. indicate satisfactory performance of $H$-identification when the modes of the disturbance model DSS2 and the modes of the system model DSS1 are located some

distance $\omega_o = \frac{2\pi}{t_1}$ , or greater, from each other in the complex plane. Since the nature of the singularly perturbed system is that it has subsets of modes located apart from each other in the complex plane, this is in line with the assumptions of successful $H$-identification.

As far as the issue of exogeneous noise, n(t), acting on the system CS, it seems reasonable that this will only affect identification if the modes of n(t) are close to the modes of the subsystem to be identified. This concern will be addressed within Chapter V of this thesis.

As a final point, once the parameters of the main subsystem DSS1 are found, that knowledge can then be incorporated to facilitate the identification of the remaining system DSS2. Details of this will be discussed in the next chapter.

In the next chapter, the computational considerations concerning the implementation of the algorithms will be dealt with, along with some examples and results to verify the direct numerical applications of $H$-identification to singularly perturbed systems.

# CHAPTER IV

# COMPUTATIONAL CONSIDERATIONS AND RESULTS

This chapter will deal with the implementation of the $H$-identification procedure for singularly perturbed systems. Section 4.1 will modify the algorithms of $H$-identification which simplifies the procedure. The second section will discuss specific computational aspects involved with implementing the algorithm and running it on the computer. The third section will focus on the direct application of the implemented algorithm for the singularly perturbed system, and provide specific examples and their associated results. The last section will examine these results and discuss their significance.

## 4.1. Modification of $H$-identification

Recently, a formulation of $H$-identification was described [PEA-5] which reveals the underlying least squares functional to be minimized over the system parameters unrestrained by the parameters characterizing the disturbance modes. At the same time, this formulation accrues significant benefits in streamlining and simplifying the computations needed to obtain the least squares functional from the observed input-output data.

Taking the system description (3.2.3) without disturbances:

$$\tilde{P}(D,\omega)y(t) + \tilde{Q}(D,\omega)u(t) = 0 \qquad (4.1.1)$$

and reshaping this yields:

$$R(D)k(t,\omega) = 0 \ , \ 0 \leq t \leq t_1 \ . \qquad (4.1.2)$$

By including disturbances d(t) acting upon the system, equation (4.1.2) admits to a general form as:

$$R(D)k(t,\omega) = S(D,\omega)d(t) \ , \ 0 \leq t \leq t_1 \ . \qquad (4.1.3)$$

Application of $T(D,\delta)$ on both sides yields:

$$T(D,\delta)R(D)k(t,\omega) = 0 \qquad (4.1.4)$$

which in vector form presents itself as

$$0 = \left[D^{\alpha}R(D),\ D^{\alpha-1}R(D),\ldots,R(D)\right]\begin{bmatrix} k(t,\omega) \\ \delta_1 k(t,\omega) \\ \vdots \\ \delta_{\alpha}k(t,\omega) \end{bmatrix} . \qquad (4.1.5)$$

Then the equation error function $z(t,\theta)$ is the solution to

$$F(D)z(t,\theta) = \left[D^{\alpha}R(D),D^{\alpha-1}R(D),\ldots,R(D)\right]\begin{bmatrix} k(t,\omega) \\ \delta_1 k(t,\omega) \\ \vdots \\ \delta_{\alpha}k(t,\omega) \end{bmatrix} \qquad (4.1.6)$$

with $\theta \triangleq (\delta,\omega) = (\delta_1,\ldots,\delta_\alpha,\omega_1,\ldots,\omega_\beta)$. Here, F(D) is chosen so that

$$F^{-1}(s)R(s)s^{\alpha} \tag{4.1.7}$$

is a proper transfer function matrix. As in Chapter III, the filter $H$ is applied to $z(t,\theta)$ and the functional $J_1(\theta)$ becomes:

$$J_1(\theta) = \langle \tilde{z}(t,\theta), \tilde{z}(t,\theta) \rangle = \int_o^{t_1} \tilde{z}^T(t,\theta)\tilde{z}(t,\theta)dt \ . \tag{4.1.8}$$

Now, if $\zeta(t,\theta)$ satisfies equation (4.1.6) with zero initial conditions, then $J_1(\theta)$ becomes (see Appendix B):

$$J_1(\theta) = \int_o^{t_1} \zeta^T(t,\theta)\zeta(t,\theta)dt - \phi^T(\theta)W^{-1}\phi(\theta) \tag{4.1.9}$$

where

$$\phi(\theta) = \int_o^{t_1} e^{A^T t} C^T \zeta(t,\theta)dt \ . \tag{4.1.10}$$

In the case of separable-in-the-parameters models, $k(t,\omega)$ becomes:

$$k(t,\omega) \equiv U(t)h(\omega) \tag{4.1.11}$$

where $U(t)$ is a matrix valued function of the observed input-output data on $[0,t_1]$ and $h(\omega)$ is a continuously

differentiable vector valued function of the system parameters. Then, $J_1(\theta)$ will become an explicitly defined function of $\theta$:

$$J_2(\theta) = \left[h^T(\omega), \delta^T H^T(\omega)\right] \begin{bmatrix} \Omega_{oo} & \Omega_{od} \\ \Omega_{do} & \Omega_{dd} \end{bmatrix} \begin{bmatrix} h(\omega) \\ H(\omega)\,\delta \end{bmatrix}$$

$$= \left[h^T(\omega), \delta^T H^T(\omega)\right] \Omega \begin{bmatrix} h(\omega) \\ H(\omega)\,\delta \end{bmatrix} \qquad (4.1.12)$$

where

$$H(\omega) = \underbrace{\begin{bmatrix} h(\omega) & & 0 \\ & \ddots & \\ 0 & & h(\omega) \end{bmatrix}}_{\alpha \text{ columns}} \qquad (4.1.13)$$

and the Gramian is

$$\Omega = \int_o^{t_1} \tilde{Y}^T(t)\tilde{Y}(t)\,dt = \int_o^{t_1} Y^T(t)Y(t)\,dt - N^T W^{-1} N \qquad (4.1.14)$$

with $Y(t)$ as the zero-state solution to

$$F(D)Y(t) = \left[D^{\alpha}R(D), D^{\alpha-1}R(D), \ldots, R(D)\right]U(t) \qquad (4.1.15)$$

and

$$N = \int_o^{t_1} e^{A^T t} C^T Y(t)\,dt \quad . \qquad (4.1.16)$$

The matrix $\Omega$ is effectively a time correlation matrix with some bias removal terms which arise from the application of the annihilating filter $H$. Also, $\Omega$ is symmetric and non-negative definite, so that $J_2(\theta)$ satisfies the positive definite property:

$$J_2(\theta) \geq 0 \ . \tag{4.1.17}$$

Thus, once $\Omega$ is computed, any hill-climbing technique can be used on $J_2(\theta)$ without further integrations of the data on $[0,t_1]$.

Examining $J_2(\theta)$, it is seen that the disturbance parameters, $\delta$, enter quadratically in $J_2(\theta)$. Thus, a necessary condition for a minimal value of $J_2(\theta)$ would be the vanishing of the gradients:

$$\frac{\partial J}{\partial \delta} = 0 \quad \text{and} \quad \frac{\partial J}{\partial \omega} = 0 \ . \tag{4.1.18}$$

Thus, solving the first of these yields:

$$\hat{\delta} = - [H^T(\omega)\Omega_{dd}H(\omega)]^{-1}H^T(\omega)\Omega_{do}h(\omega) \tag{4.1.19}$$

assuming the needed inverse exists functionally in $\omega$. Substituting $\hat{\delta}$ into $J_2(\theta)$ yields an explicit function of the system parameters:

$$J_3(\omega)=h^T(\omega)\left[\Omega_{oo}-\Omega_{od}H(\omega)[H^T(\omega)\Omega_{dd}H(\omega)]^{-1}H^T(\omega)\Omega_{do}\right]h(\omega) \ . \tag{4.1.20}$$

In general, $J_3(\omega)$, though positive definite, is nonlinear, nonquadratic, and not necessarily convex in $\omega$. Although not explicitly present, the effect of the disturbance parameters is manifest in the inverse of $H^T(\omega)\Omega_{dd}H(\omega)$.

At this point, the computations needed for $\Omega$ are undertaken. First, the solution $Y(t)$ to equation (4.1.15) can be partitioned as:

$$Y(t) = [Y_o(t), Y_1(t), \ldots, Y_\alpha(t)], \quad 0 \leq t \leq t_1 . \tag{4.1.21}$$

(Thus, $DY_{i+1}(t) = Y_i(t)$, $0 \leq t \leq t_1$, $0 \leq i \leq \alpha - 1$.)
Then $N$ can be partitioned as:

$$N = [N_o, N_1, \ldots, N_\alpha] = \int_o^{t_1} e^{A^T t} C^T [Y_o(t), \ldots, Y_\alpha(t)]\, dt. \tag{4.1.22}$$

Then $\Omega$ is partitioned into $(\alpha + 1)^2$ blocks as

$$\Omega_{ij} = \int_o^{t_1} Y_i^T(t) Y_j(t)\, dt - N_i^T W^{-1} N_j , \quad 0 \leq i,j \leq \alpha . \tag{4.1.23}$$

Whence $\Omega_{od}$, $\Omega_{do}$, and $\Omega_{dd}$ for equation (4.1.12) are defined as:

$$\Omega_{od} = [\Omega_{o1}, \ldots, \Omega_{o\alpha}]$$

$$\Omega_{do} = \Omega_{od}^T$$

$$\Omega_{dd} = \begin{bmatrix} \Omega_{11} & \cdots & \Omega_{1\alpha} \\ \vdots & \ddots & \vdots \\ \Omega_{\alpha 1} & \cdots & \Omega_{\alpha\alpha} \end{bmatrix} . \tag{4.1.24}$$

Let a matrix function $Z(t_1)$ be defined as:

$$Z(t_1) = e^{-A^T t_1} N \tag{4.1.25}$$

with a similar partitioning:

$$Z(t_1) = [Z_o(t_1),\ldots,Z_\alpha(t_1)] = e^{-A^T t_1}[N_o,\ldots,N_\alpha] . \tag{4.1.26}$$

(As occurred before, $DZ_{i+1}(t) = Z_i(t)$, $0 \leq t \leq t_1$, $0 \leq i \leq \alpha - 1$.) Then the bias removal term becomes:

$$N^T W^{-1} N = Z^T(t_1) e^{At_1} W^{-1} e^{A^T t_1} Z(t_1) = Z^T(t_1)\tilde{W}^{-1}Z(t_1) \tag{4.1.27}$$

where

$$\tilde{W} \triangleq e^{-A^T t_1} W e^{-At_1} \tag{4.1.28}$$

is the observability Gramian for the pair $(-A,C)$.

Through the partitioning of Z and N, each $Z_i$ satisfies

$$Z_i(t_1) = e^{-A^T t_1} \int_o^{t_1} e^{A^T \tau} C^T Y_i(\tau) d\tau \ , \ i = 0,1,\ldots,\alpha \tag{4.1.29}$$

which is the solution (for $t = t_1$) to the differential equation

$$\dot{Z}_i(t) = -A^T Z_i(t) + C^T Y_i(t) \ , \ Z_i(0) = 0 . \tag{4.1.30}$$

Let (A,B,C,E) be a minimal realization [GOO] for the transfer function $F^{-1}(s)R(s)s^{\alpha}$. Then using equation (4.1.15), $Y_o(t)$ is the solution to

$$\dot{X}_o(t) = AX_o(t) + BU(t) \qquad X_o(0) = 0$$

$$Y_o(t) = CX_o(t) + EU(t) \qquad 0 \leq t \leq t_1 \qquad (4.1.31)$$

and thus

$$Y_{i+1}(t) = D^{-1}Y_i(t) \qquad 0 \leq t \leq t_1 \ . \qquad (4.1.32)$$

This leads to the following theorem [PEA-5]:

Theorem 4.1: Let (A,B,C,E) be a minimal realization for $F^{-1}(s)R(s)s^{\alpha}$, with det $A \neq 0$. Then a least squares estimate of $\omega$, in the separable case, is obtained by minimizing $J_3(\omega)$. The matrices $Y_i(t)$, $Z_i(t)$ $(0 \leq i \leq \alpha)$ comprising $\Omega$ are efficiently determined from the zero state solution to:

$$(i = 0) \qquad \dot{X}_o(t) = AX_o(t) + BU(t)$$

$$Y_o(t) = CX_o(t) + EU(t)$$

$$\dot{Z}_o(t) = -A^T Z_o(t) + C^T Y_o(t) \qquad (4.1.33)$$

$$(i = 1,\ldots,\alpha) \qquad X_i(t) = A^{-1}[X_{i-1}(t) - BD^{-i}U(t)]$$

$$Y_i(t) = CX_i(t) + ED^{-i}U(t)$$

$$Z_i(t) = (-A^T)^{-1}[Z_{i-1}(t) - C^T Y_i(t)] \qquad (4.1.34)$$

where $D^{-i}$ denotes the i-fold pure integration operator with zero initial conditions.

Proof: Since $J_3(\omega)$ has already been derived together with equations (4.1.33), it remains to establish equations (4.1.34). Since $X_i(t)$ can be defined iteratively from $X_{i+1}(t) = D^{-1}X_i(t)$ $(i = 0,\ldots,\alpha-1)$, the first and second relations in equations (4.1.34) are therefore immediately seen to be valid from equations (4.1.33). And since $Z_{i+1}(t) = D^{-1}Z_i(t)$, the third relation in equations (4.1.34) is also immediately valid from equations (4.1.33). Q.E.D.

Equations (4.1.34) represent a significant saving in computation not only because the data matrix is generally sparse, but also because the number of distinct time functions in U(t) is less than the number of non-zero entries. Furthermore, all that is needed of the Z function is $Z(t_1)$. And aside from the pure integrations $D^{-i}U(t)$, only one other set of integrations (equations (4.1.33)) is needed.

It is the application of Theorem 4.1 that will be used as the specific algorithm for *H*-identification.

## 4.2. Computational Aspects

Before the actual implementation of the algorithm can be undertaken, a choice for F(D) must be made. Apart

from det $F(D) \neq 0$, the selection of $F(D)$ is quite unrestricted and the modes of $F(D)$ can, in theory, be selected as either stable or unstable since all computations are confined to the finite interval $[0,t_1]$. However, strongly unstable modes in $F(D)$ are undesirable since the control of the integration errors will be more difficult. Now, if n is the order of the system to be identified (DSS1 as in Figure 3.4) and $\alpha$ is the order of the disturbance process (DSS2), then the order of $F(D)$ must be c with $c \geq n + \alpha$ such that $F^{-1}(s)R(s)s^{\alpha}$ is proper. Pole-zero cancellation is permitted in $F^{-1}(s)R(s)s^{\alpha}$, but any such cancelled modes must be included in the W matrix because such modes, although not controllable, are observable and must be included in the annihilating filter.

The choice of $F(D)$ with c assumed even is

$$F(D) = \prod_{k=1}^{c/2} (D^2 + k^2\omega_o^2)I \qquad (4.2.1)$$

with

$$\omega_o \triangleq \frac{2\pi}{t_1} \quad . \qquad (4.2.2)$$

This selection of $F(D)$ simplifies the computations significantly. The fundamental solution to the homogeneous equation $F(D)z(t) = 0$ (that is, the modes for $Ce^{At}$) involves the functions $\{\sin(k\omega_o t), \cos(k\omega_o t)\}$, $k = 1,2,\ldots,\frac{c}{2}$, which are orthogonal over the observation

time interval $[0,t_1]$. Hence, the Gramian matrix W is diagonal, as is $\tilde{W}$, such that:

$$W = \tilde{W} = \frac{t_1}{2} I \ . \tag{4.2.3}$$

(This same frequency $\omega_o$ was mentioned near the end of Chapter III as the minimum resolving distance between the two sets of modes of DSS1 and DSS2.) Notice that the resonance frequencies of the filter $F^{-1}(D)$ coincide with the null frequencies of the filter $H$ so that the composite filter $HF^{-1}(D)R(D)D^{\alpha}$ tends to preserve the useful information in the data at all frequencies.

In the case of linear, time-invariant SISO systems,

$$R(D) = [D^n, D^{n-1}, \ldots, 1] \tag{4.2.4}$$

yielding a matrix transfer function (as $1 \times (n + 1)$).

$$F^{-1}(s)R(s)s^{\alpha} = \frac{1}{\prod_{k=1}^{c/2} (s^2 + k^2\omega_o^2)} [s^{n+\alpha}, s^{n+\alpha-1}, \ldots, s^{\alpha}] \ . \tag{4.2.5}$$

In order for (A,B,C,E) to be a minimal realization for $F^{-1}(s)R(s)s^{\alpha}$, the following four conditions are needed [GOO]:

$$\text{(i)} \quad C(sI - A)^{-1}B + E = F^{-1}(s)R(s)s^{\alpha} \tag{4.2.6a}$$

$$\text{(ii)} \quad \text{rank } [C^T, (CA)^T, \ldots, (CA^{n-1})^T] = \text{rank } A \tag{4.2.6b}$$

$$\text{(iii)}\quad \operatorname{rank}\,[B, AB, \ldots, A^{n-1}B] = \operatorname{rank} A \qquad (4.2.6c)$$

$$\text{(iv)}\quad \operatorname{rank}\begin{bmatrix} CB & \cdots & CA^{n-1}B \\ \vdots & \ddots & \vdots \\ CA^{n-1}B & \cdots & CB \end{bmatrix} = \operatorname{rank} A\,. \qquad (4.2.6d)$$

By taking A (as c × c) in the canonical form

$$A = \left[\begin{array}{ccc:c} 0 & & & \\ \vdots & & & \\ 0 & & & I \\ \hdashline -a_c & \cdots & -a_1 & \end{array}\right] \qquad (4.2.7)$$

where

$$s^c + a_1 s^{c-1} + \ldots + a_c = \prod_{k=1}^{c/2} (s^2 + k^2\omega_o^2) \qquad (4.2.8)$$

and C (as 1 × c) in the form

$$C = [1, 0, \ldots, 0]\,, \qquad (4.2.9)$$

the matrices B (as c × (n + 1)) and E (as 1 × (n + 1)) can easily be found satisfying conditions (4.2.6). For example, with n = 2 and α = 1, then c = 4 so that:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -4\omega_o^4 & 0 & -5\omega_o^2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5\omega_o^2 & 0 & 1 \\ 0 & -5\omega_o^2 & 0 \end{bmatrix}$$

$$C = [1, 0, 0, 0] \qquad E = [0, 0, 0] \qquad (4.2.10a)$$

$$F^{-1}(s)R(s)s^{\alpha} = C(sI - A)^{-1}B + E =$$

$$= \frac{1}{\prod_{k=1}^{c/2} (s^2 + k^2\omega_o^2)} [s^3, s^2, s^1] \quad . \qquad (4.2.10b)$$

As to the choice in $t_1$, some fraction of the longest expected system time constant is suggested [PEA-2,3].

All numerical examples were run on a Prime 750 Computer using Fortran IV language (see Appendix C). The IMSL Library [IMSL] was utilized for integration (DGEAR), interpolation (ICSEVU, ICSCCU), and minimization (ZSRCH, ZXMIN) routines. The IMSL library contains a comprehensive range of high quality validated algorithms. The library is internally self-consistent and well-documented for the user. The effectiveness of the IMSL library is discussed in Jacobs [JAC]. The routine DGEAR was significant for the problem of integrating singularly perturbed systems of equations since it is a backward differentiation formula based on Gear's stiff methods [GE], [SHA]. The routine ZSRCH systematically searches a spatial region for good starting points to serve as initial guesses to ZXMIN. This is necessary since $J_3(\omega)$ is not necessarily convex. Hence more than one initial guess may be necessary before the absolute minimum of $J_3(\omega)$ is achieved. Roughly speaking, the value $J_3(\hat{\omega}) \leq 10^{-4}$ is sufficient [PEA-2] to be assured that $||\hat{\omega} - \omega^*||_2$ is also small and

that convergence has occurred. However, this threshold might depend on the $[0,t_1]$ interval. And the routine ZXMIN uses a quasi-Newton method to find the unconstrained minimum of $J_3(\omega)$.

The only restrictions to *H*-identification comes from under-ordering the disturbance model, and from disturbance models with too large a value of $\alpha$, for then the formulation may not be suitable because a long time interval will be required when there are a large number of unknown parameters due to many frequency components in the disturbance.

## 4.3. Direct Application and Results

The most basic illustration of the use of *H*-identification as applied to singularly perturbed systems is the minimum-dimensioned linear, time-invariant example:

$$\frac{d}{dt}\, x(t) = A_{11}x(t) + A_{12}z(t) + B_1u(t)$$

$$\varepsilon\, \frac{d}{dt}\, z(t) = A_{21}x(t) + A_{22}z(t) + B_2u(t)$$

$$y(t) = C_1x(t) + C_2z(t) \qquad (4.3.1)$$

where all variables (matrices) are scalars ($1 \times 1$). (Note that the effect of the input, u(t), is through the state equations directly. This is a standard formulation with u(t).)

Thus, this SISO system represents the process operating in coupled form CS. Known design inputs u(t) will be given along with the observed output y(t) on $[0,t_1]$. What is desired are the representations of the two subsystems (approximate):

$$\underline{\text{SDSS}}: \quad \frac{d}{dt} x_s(t) = A_o x_s(t) + B_o u_s(t)$$

$$y_s(t) = C_o x_s(t) + D_o u_s(t) \qquad (4.3.2)$$

$$\underline{\text{FDSS}}: \quad \varepsilon \frac{d}{dt} z_f(t) = A_{22} z_f(t) + B_2 u_f(t)$$

$$y_f(t) = C_2 z_f(t) + D u_f(t) \; . \qquad (4.3.3)$$

Each of these subsystems is input-output equivalent to:

$$\dot{\tilde{y}}(t) + a_1 \tilde{y}(t) = b_o \dot{\tilde{u}}(t) + b_1 \tilde{u}(t) \; . \qquad (4.3.4)$$

Therefore, it remains to determine, for each subsystem, the scalar quantities $a_1$, $b_o$, $b_1$. According to the formulations in Section 4.1, it is found that:

$$0 = R(D)k(t,\omega) = R(D)U(t)h(\omega) =$$

$$= \left[D,1\right] \begin{bmatrix} \tilde{y}(t) & 0 & -\tilde{u}(t) & 0 \\ 0 & \tilde{y}(t) & 0 & -\tilde{u}(t) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ b_o \\ b_1 \end{bmatrix}$$

$$0 \leq t \leq t_1 \; . \qquad (4.3.5)$$

This last equation is the model of the subsystem DSS1 to be identified, with the other subsystem output considered as disturbance, d(t), such that:

$$y(t) \stackrel{\Delta}{=} y_f(t) + y_s(t) \equiv \tilde{y}(t) + d(t) \ . \qquad (4.3.6)$$

With the à priori knowledge of the dimensions n and $\alpha$ of DSS1 and DSS2, respectively--along with the observed input and output--U(t), $h(\omega)$, and $H(\omega)$ can be formed. Also, c can be chosen to make $F^{-1}(s)R(s)s^{\alpha}$ proper, whereby (A,B,C,E) can be selected. Then the necessary integrations (4.1.33) and (4.1.34) can be executed, and $\Omega$ formed.

At this point, all integrations of the data are complete, and minimization of $J_3(\omega)$ is all that remains. Once successful minimization is attained, the model DSS1 is known, and this information can then be used to find the parameters of the other subsystem DSS2. Then, if so desired, the newly learned models can be transformed into state-space configuration (as discussed in Section 3.1). Tests for further determining the multiplier $\varepsilon$ in the state-space form of FDSS are discussed in Mendel [ME].

For each example attempted, some à priori knowledge on the separation of modes between the fast and slow subsystems was assumed. This information was used to help in the design of the input signals. The input

signals were of step, ramp, parabolic, and sinusoidal types. Various combinations of inputs and final times, $t_1$, were applied until consistency (to three of four significant places) in one or more parameters appeared. These newly found parameters were then fixed as known constants, making the parameter vector, $h(\omega)$, smaller. This iterative procedure was continued until all parameters within $h(\omega)$ were learned.

At this point, the estimated model for DSS1 was attained. To learn the parameters of DSS2, the newly found model DSS1 was simulated, and its output effect, $y_1(t)$, was subtracted from the output, $y(t)$, of the original coupled system CS, yielding $y_2(t)$ (plus some small disturbance effect due to the unknown initial conditions on DSS1). Then $H$-identification was performed anew to learn the estimated parameters of DSS2.

As far as which subsystem to identify first and over what interval(s) of time to do this identification, a heuristic study pointed directly to a unique procedure:

Procedure 4.1:

(i) Observe the coupled system output, $y(t)$, over some time interval $[t_o,t_1]$, where $t_o > 0$ is some time instant after the fast response has effectively died out.

(ii) Use $H$-identification to estimate the parameters of the slow subsystem. Then the estimated output $\hat{y}_s(t)$, for a given input, can be determined--up to the initial condition response--over any time interval desired.

(iii) Observe the coupled system output, $y(t)$, over some interval $[0,t_a]$, where $0 < t_a \leq t_o$.

(iv) Form the output measurement $y(t) - \hat{y}_s(t)$ over $[0,t_a]$.

(v) With this formed output, use $H$-identification to estimate the parameters of the fast subsystem.

Actually, the observations of $y(t)$ over $[0,t_1]$ can be made all at once, with the record of $y(t)$ over $[0,t_a]$ being stored for the later computations in step (iv). For most of the examples considered (shown in the tables to follow), $t_o \leq 1$ (second) worked well as a time instant after which the fast response had effectively dissipated. And the instant $t_a$ was generally taken as $t_o$.

It was hoped that FDSS could be successfully estimated over $[0,t_a]$ first, and then SDSS estimated over $[t_o,t_1]$. For then this would point toward an adaptive control procedure for singularly perturbed systems.

A very interesting discovery was made while running the computer analysis. It was found that the subsystems being identified were not the zeroth-order approximation (equations (2.2.36)-(2.2.38)) nor even the first-order approximation (equations (2.2.41)-(2.2.43)). What was being identified was the exact decoupled subsystems (equations (2.2.21)-(2.2.23)):

$$\underline{\text{SDSS}}: \quad \dot{v}_1 = (A_{11} - A_{12}L)v_1 + [B_1 - M(B_2 + \varepsilon LB_1)]u \tag{4.3.7}$$

$$y_1 = (C_1 - C_2L)v_1 \tag{4.3.8}$$

$$\underline{\text{FDSS}}: \quad \varepsilon\dot{v}_2 = (A_{22} + \varepsilon LA_{12})v_2 + (B_2 + \varepsilon LB_1)u \tag{4.3.9}$$

$$y_2 = [\varepsilon(C_1 - C_2L)M + C_2]v_2 \tag{4.3.10}$$

where

$$y(t) = y_1(t) + y_2(t) \triangleq y_s(t) + y_f(t) \tag{4.3.11}$$

and L and M satisfy equations (2.2.6) and (2.2.7).

The specific examples tested are listed in Table 4.1. A range of values for $\varepsilon$ was considered, from $\varepsilon = \frac{1}{2}$ to $\varepsilon = \frac{1}{200}$. For each example in Table 4.1, the model of CS was (one fast state and one slow state):

$$\varepsilon \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = A \begin{bmatrix} x \\ z \end{bmatrix} + B[u]$$

$$[y] = [3,2] \begin{bmatrix} x \\ z \end{bmatrix} \tag{4.3.12}$$

Table 4.1. Examples (n = 1, m = 1)

| Example # | $\varepsilon$ | A | B | $\lambda_1, \lambda_2$ | $(a_1^*, b_o^*, b_1^*)_{slow}$ | $(a_1^*, b_o^*, b_1^*)_{fast}$ |
|---|---|---|---|---|---|---|
| 1 | $\frac{1}{2}$ | $\begin{bmatrix} 0 & 1 \\ -1 & -1.5 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}$ | -1,-2 | (1,0,3) | (2,0,2) |
| 2 | $\frac{1}{5}$ | $\begin{bmatrix} 0 & 1 \\ -1 & -1.2 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ \frac{1}{5} \end{bmatrix}$ | -1,-5 | $(1,0,\frac{3}{2})$ | $(5,0,\frac{7}{2})$ |
| 3 | $\frac{1}{10}$ | $\begin{bmatrix} 0 & 1 \\ -1 & -1.1 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ \frac{1}{10} \end{bmatrix}$ | -1,-10 | $(1,0,\frac{11}{9})$ | $(10,0,\frac{34}{9})$ |
| 4 | $\frac{1}{20}$ | $\begin{bmatrix} 0 & 1 \\ -1 & -1.05 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ \frac{1}{20} \end{bmatrix}$ | -1,-20 | $(1,0,\frac{21}{19})$ | $(20,0,\frac{74}{19})$ |
| 5 | $\frac{1}{50}$ | $\begin{bmatrix} 0 & 1 \\ -1 & -1.02 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ \frac{1}{50} \end{bmatrix}$ | -1,-50 | $(1,0,\frac{51}{49})$ | $(50,0,\frac{194}{49})$ |
| 6 | $\frac{1}{100}$ | $\begin{bmatrix} 0 & 1 \\ -1 & -1.01 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ \frac{1}{100} \end{bmatrix}$ | -1,-100 | $(1,0,\frac{101}{99})$ | $(100,0,\frac{394}{99})$ |
| 7 | $\frac{1}{200}$ | $\begin{bmatrix} 0 & 1 \\ -1 & -1.005 \end{bmatrix}$ | $\begin{bmatrix} 1 \\ \frac{1}{200} \end{bmatrix}$ | -1,-200 | $(1,0,\frac{201}{199})$ | $(200,0,\frac{794}{199})$ |

* true values.

Table 4.2. Results--Slow Subsystem

| Example # | $(\hat{a}_1, \hat{b}_o, \hat{b}_1)$ | $\frac{\|\|\hat{\omega} - \omega^*\|\|}{\|\|\omega^*\|\|}$ | $\|\|\hat{\omega}-\omega^*\|\|$ | $\|J_3(\hat{\omega})\|$ | $[t_o, t_1]$ |
|---|---|---|---|---|---|
| 1 | (0.882,0.025,2.883) | 0.053 | 0.168 | 9.42E-3 | [3,13] |
| 2 | (0.994,0.001,1.512) | 0.007 | 0.013 | 6.56E-4 | [1,10] |
| 3 | (1.001,-0.001,1.223) | 0.001 | 0.002 | 7.09E-5 | [1,7.5] |
| 4 | (1.001,0.000,1.105) | 0.0007 | 0.001 | 2.00E-5 | [0.6,6] |
| 5 | (1.000,0.000,1.040) | 0.0005 | 0.0008 | 8.83E-6 | [0.3,6] |
| 6 | (1.001,0.004,1.022) | 0.003 | 0.004 | 9.89E-5 | [0.2,5] |
| 7 | (1.000,0.001,1.011) | 0.0009 | 0.001 | 3.92E-5 | [0.1,5] |

Table 4.3. Results--Fast Subsystem

| Example # | $(\hat{a}_1, \hat{b}_o, \hat{b}_1)$ | $\frac{\|\|\omega - \omega^*\|\|}{\|\|\omega^*\|\|}$ | $\|\|\hat{\omega} - \omega^*\|\|$ | $\|J_3(\hat{\omega})\|$ | $[0, t_a]$ |
|---|---|---|---|---|---|
| 1 | (1.829,0.033,2.411) | 0.158 | 0.446 | 4.32E-2 | [0,3] |
| 2 | (5.010,0.000,3.602) | 0.017 | 0.102 | 1.27E-4 | [0,1] |
| 3 | (10.02,0.000,3.775) | 0.002 | 0.020 | 2.01E-4 | [0,1] |
| 4 | (20.00,0.000,3.892) | 0.0001 | 0.003 | 5.48E-5 | [0,0.6] |
| 5 | (50.04,-0.013,3.960) | 0.0008 | 0.042 | 8.83E-6 | [0,0.3] |
| 6 | (104.6,0.094,3.981) | 0.046 | 4.601 | 9.10E-4 | [0,0.2] |
| 7 | (194.4,-0.119,4.021) | 0.028 | 5.601 | 3.50E-2 | [0,0.1] |

Table 4.4. Example and Result (n = 1, m = 2)

$$\begin{bmatrix} \frac{d}{dt} & & \\ & \frac{1}{20}\frac{1}{dt} & \\ & & \frac{1}{20}\frac{d}{dt} \end{bmatrix} \begin{bmatrix} x \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1.05 & -1 \\ -0.9 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 1 \\ \frac{1}{20} \\ \frac{1}{20} \end{bmatrix} [u];$$

$$[y] = [2,3,1] \begin{bmatrix} x \\ z_1 \\ z_2 \end{bmatrix}$$

$(\lambda_1,\lambda_2,\lambda_3) = (-2,-19,-21)$ SDSS*: $\dot{y}_s + 2y_s = 0\dot{u} + \frac{1511}{289}u$

FDSS*: $\ddot{y}_f + 40\dot{y}_f + 399y_f = 0\ddot{u}_f + \frac{7113}{5491}\dot{u}_f + \frac{542493}{5491}u_f$

$(\hat{a}_1,\hat{b}_o,\hat{b}_1)_{slow} = (2.003,-0.001,5.234)$

$(\hat{a}_1,\hat{a}_2,\hat{b}_o,\hat{b}_1,\hat{b}_2)_{fast} = (40.09,399.8,0.002,1.293,98.06)$

$$\frac{||\hat{\omega} - \omega^*||_{slow}}{||\omega^*||_{slow}} = 0.001 \qquad ||\hat{\omega} - \omega^*||_{slow} = 0.006$$

$$\frac{||\hat{\omega} - \hat{\omega}^*||_{fast}}{||\omega^*||_{fast}} = 0.002 \qquad ||\hat{\omega} - \omega^*||_{fast} = 0.885$$

$|J_3(\omega)|_{slow} = 4.46E\text{-}5$

$|J_3(\omega)|_{fast} = 3.08E\text{-}4$

$(t_a,t_o,t_1) = (0.8,0.8,8.0)$

and the model of each subsystem was:

$$\dot{y} + a_1 y = b_o \dot{u} + b_1 u \ . \qquad (4.3.13)$$

The results for these examples can be found in Tables 4.2 and 4.3.

It has already been reported [PEA-2,3] that the algorithm does not work well for systems with too many modes--particularly high frequency modes. Therefore, the size of each example was kept to a minimum to test the actual effectiveness of the algorithm and procedure. As also reported in these same articles, there is essentially no effect by over-ordering the disturbance model. In many examples this was done. Table 4.4 indicates the results of an example with one slow state and two fast states.

## 4.4. Discussion of Results

A grasp on the effectiveness of $H$-identification of a singularly perturbed system can now be made through the examination of the information displayed in Tables 4.2 and 4.3. It is seen that the algorithm had much difficulty in identifying the separate subsystems when the time-scale factor $\varepsilon$ was large (Example #1: $\varepsilon = \frac{1}{2}$). In this example the relative errors in the slow and fast parameters amounted to 5.3% and 15.8%, respectively. This apparent failure of the algorithm--for this example

--stems, most definitely, from the fact that the eigenvalues for the fast and slow subsystems are too close to each other, even with the small resolving distance of $\omega_o = \frac{2\pi}{10}$ .

The effectiveness of the algorithm to differentiate between the subsystems improves as $\varepsilon$ decreases in size. This was anticipated from the beginning of this thesis. As $\varepsilon$ decreased, the relative error in $\hat{\omega}$ remained less than 0.1%. (That these errors are considerably small was unexpected; however, not totally surprising. The preliminary investigations of Pearson et al. showed very accurate results in their simulation studies.) The relative errors in the slow subsystem parameter estimates improved as $\varepsilon$ decreased, due to the fact that the "disturbance" system (FDSS) died out rapidly. This parallels the fundamental construct in singular perturbation theory that as $\varepsilon \to 0$, the coupled system CS basically reduces to the lower-order slow subsystem SDSS. A major reckoning for $\varepsilon \leq \frac{1}{5}$, though, is that almost all parameters $\hat{\omega}_i$ $(i = 1,2,3)$ for both subsystems are accurate to two or three significant digits.

As to the effect of a decreasing time-scale $\varepsilon$ on the parameter estimates for the fast subsystem, a slight increase in the relative error is noticed, albeit the absolute error is poor. This could be due, in part, to

the extreme speed with which the exponential factors decay, thereby decreasing the richness of information in the output signal $y_f(t)$, and/or due to integration errors.

As to the example in Table 4.4, wherein the fast-state dimension is increased to two, the parameter estimates $\hat{\omega}$ for both the fast and slow subsystems are accurate to three significant digits, and the relative errors are less than 0.5%.

All these results tend to indicate that the two subsystems of a coupled singularly perturbed process can be identified provided the time-scale, $\varepsilon$, is much less than unity.

In the next chapter of this text, a review of the essentials covered in this thesis will be made, along with relevant conclusions. The chapter will close with some insights on directions for further research in this problem area.

# CHAPTER V

## SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

The purpose of this dissertation was to examine the problem of identifying the system parameters of the fast and slow subsystems of a singularly perturbed system. This singularly perturbed system was operating in coupled form with its input and output used to estimate the characteristics of the decoupled subsystems.

In the beginning of this document, an introduction to the nature of singularly perturbed systems, along with the nature of the identification problem, was presented. Following that, some algebra and theory of singularly perturbed systems was revealed, along with the mechanics of its decomposition into slow and fast subsystems.

Next, the solution of the problem at hand was undertaken using a deterministic, least squares, equation error, finite time-interval, identification method. This method utilized a filter to annihilate the initial condition response, and assumed the disturbances to be solutions to a homogeneous differential equation. The adaptation of this method was then applied to an example set

of deterministic, linear, time invariant, single-input single-output, stable, observable, controllable, singularly perturbed systems.

The results of this analysis revealed success in identifying the parameters of the separate subsystems via a unique procedure determined through a heuristic study. The success of the procedure was based, in part, on the time-scale parameter $\varepsilon$, for if $\varepsilon$ was too large the parameter estimates were not significantly close to the true parameter values.

It is, therefore, possible to determine reasonable estimates for the parameters of each decoupled subsystem from the input and output (observed over a finite time interval) of a system operating in coupled singularly perturbed form.

There are several recommendations for further research in this area. The first of these is to explore the identification problem for singularly perturbed systems that are nonlinear in form. Since the theory of $H$-identification is valid for nonlinear systems, it might prove applicable to this. Another area of interest would be the problem of identifying the parameters for a singularly perturbed system with more than one time scale. An iterative approach, similar to Procedure 4.1, might solve this problem. Thirdly, the success of identifying the decoupled

fast and slow subsystems could be further developed to the broader problem of parameter estimation for any system admitting to a decoupling into subsystems.

And finally, and most significantly, a study could be undertaken on the success of parameter estimation for the singularly perturbed system when corrupted by noise. The effects of different noise (stochastic, white, etc.) upon the actual input and output variables might significantly change the effectiveness of the identification procedure, particularly if the noise contains modes in common with the modes of any subsystem.

These directions are significant and warrant further investigations and developments.

# APPENDIX A

## APPENDIX A

Verification of $H(H) = H$:

$$H\left[H(z(t))\right] = H\left[z(t) - Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T z(\tau)\,d\tau\right]$$

$$= H(z(t)) - H\left[Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T z(\tau)\,d\tau\right]$$

$$= H(z(t)) - \left[\{Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T z(\tau)\,d\tau\} -\right.$$

$$\left. - Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T\{Ce^{A\tau}W^{-1}\int_o^{t_1} e^{A^T\sigma}C^T z(\sigma)\,d\sigma\}d\tau\right]$$

$$= H(z(t)) - Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T z(\tau)\,d\tau +$$

$$+ Ce^{At}W^{-1}\left[\int_o^{t_1} e^{A^T\tau}C^T Ce^{A\tau}\,d\tau\right]W^{-1}\int_o^{t_1} e^{A^T\sigma}C^T z(\sigma)\,d\sigma$$

$$= H(z(t)) - Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T z(\tau)\,d\tau +$$

$$+ Ce^{At}W^{-1}[W]W^{-1}\int_o^{t_1} e^{A^T\sigma}C^T z(\sigma)\,d\sigma$$

$$= H(z(t)) - Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\tau}C^T z(\tau)\,d\tau +$$

$$+ Ce^{At}W^{-1}\int_o^{t_1} e^{A^T\sigma}C^T z(\sigma)\,d\sigma = H(z(t)) \quad . \quad \text{Q.E.D.}$$

Verification of adj($H$) = $H$ (i.e. $H^* = H$):

$$\langle Hx,y\rangle = \langle \tilde{x},y\rangle = \langle x^T(t) -$$

$$- (Ce^{At}W^{-1}\int_0^{t_1} e^{A^T\tau}C^T x(\tau)d\tau)^T, y\rangle$$

$$= \int_0^{t_1}\Big[x^T(t)y(t) -$$

$$- (Ce^{At}W^{-1}\int_0^{t_1} e^{A^T\tau}C^T x(\tau)d\tau)^T y(t)\Big]dt$$

$$= \int_0^{t_1}\Big[x^T(t)y(t) -$$

$$- \int_0^{t_1} x^T(\tau)Ce^{A\tau}d\tau(W^{-1})^T e^{A^T t}C^T y(t)\Big]dt$$

$$= \int_0^{t_1} x^T(t)y(t)dt -$$

$$- \int_0^{t_1}\int_0^{t_1} x^T(\tau)Ce^{A\tau}d\tau(W^{-1})^T e^{A^T t}C^T y(t)dt\ .$$

But $W = W^T \Rightarrow W^{-1} = (W^{-1})^T$ . And switching $t \leftrightarrow \tau$ in second term:

$$\langle Hx,y\rangle = \int_0^{t_1} x^T(t)y(t)dt -$$

$$- \int_0^{t_1} x^T(t)Ce^{At}dtW^{-1}\int_0^{t_1} e^{A^T\tau}C^T y(\tau)dtd\tau$$

$$= \int_0^{t_1} \left[ x^T(t)y(t)dt - \right.$$

$$\left. - x^T(t)Ce^{At}W^{-1} \int_0^{t_1} e^{A^T\tau}C^T y(\tau)d\tau \right] dt$$

$$= \langle x, y - Ce^{At}W^{-1} \int_0^{t_1} e^{A^T\tau}C^T y(\tau)d\tau \rangle$$

$$= \langle x, \tilde{y} \rangle = \langle x, Hy \rangle \ .$$

Thus, $\langle Hx, y \rangle = \langle x, Hy \rangle$ .

But by definition, $\langle Hx, y \rangle = \langle x, H^* y \rangle$ .

Thus, $\langle x, Hy \rangle = \langle x, H^* y \rangle \Rightarrow H^* = H$.

Therefore, $H$ is self-adjoint. Q.E.D.

See also Halmos [HA], wherein it is shown that every projection operator is a self-adjoint operator.

# APPENDIX B

## APPENDIX B

Derivation of equation (4.1.9):

Using operator notation, $J_1(\theta)$ in equation (4.1.8) is given by $J_1(\theta) = \langle H(\zeta), H(\zeta) \rangle$ since $\tilde{z} = H(\zeta)$. Since $H$ is self-adjoint and a projection,

$$J_1(\theta) = \langle \zeta, H^2(\zeta) \rangle = \langle \zeta, H(\zeta) \rangle$$

$$= \langle \zeta, \zeta - Ce^{At} W^{-1} \int_0^{t_1} e^{A^T \tau} C^T \zeta(\tau, \theta) d\tau \rangle$$

$$= \langle \zeta, \zeta \rangle - \langle \zeta, Ce^{At} W^{-1} \int_0^{t_1} e^{A^T \tau} C^T \zeta(\tau, \theta) d\tau \rangle$$

$$= \langle \zeta, \zeta \rangle - \langle \zeta, Ce^{At} W^{-1} \phi(\theta) \rangle$$

$$= \langle \zeta, \zeta \rangle - \int_0^{t_1} \zeta(t, \theta) Ce^{At} dt W^{-1} \phi(\theta)$$

$$= \langle \zeta, \zeta \rangle - \phi^T(\theta) W^{-1} \phi(\theta) \ .$$

# APPENDIX C

## APPENDIX C

### Computer Programs (n = 1, α = 3)

```
INTEGER NEGO,M,MI,IX,IR1,NTAB,NPART,IC,IWK1(2),LOOP,LOOPM1
INTEGER IR2,NEGXOS,IWK2(4),IP,NEGZOS,NP,NSIG,MX,IOPT,NDUM
INTEGER LOOP1,IDUM,NGENPT,NCHOSE,OUTER,IDUM20,NTAB20
INTEGER NTAB14,NTAB8,NDUM20,NDUM14,IC20,IC14,IDUM14
INTEGER IR2,IR3,IR4,IR5,IR6,IR7,IR8,IR9,IR10,IR11,IR12,IR13
INTEGER IR14,IR15,IR16,IR17,N,IA,IB,JA,JB,IWK3(4)
INTEGER IWK4(4),IWK5(4),IWK6(4),IWK7(4),IWK8(4),IWK9(4)
INTEGER IWK10(3),IWK11(9),NDEF
REAL T,XORG(2),TL,H,DELTAT,T1,TABSC(451),TEND,WK1(26),Y(451)
REAL GPMEG2,GPMEG4,DUMY(451),DUMT(451),CY(450,3),CDUMY(450,3)
REAL XOSET1(4),XOSET2(4),XOSET3(4),XOSET4(4),D,XO11(445)
REAL XO12(445),XO13(445),XO14(445),XO21(445),XO22(445),XO23(445)
REAL XO24(445),XO31(445),XO32(445),XO33(445),XO34(445),XO41(445)
REAL XO42(445),XO43(445),XO44(445),DUM1(445),DUM2(445),DUM3(445)
REAL DUM4(445),WK2(60),WK3(60),WK4(60),WK5(60),WK6(60),WK7(60)
REAL WK8(60),WK9(60),DUMT14(445),CDUM1(444,3),CDUM2(444,3)
REAL CDUM3(444,3),CDUM4(444,3),CXO11(444,3),CXO12(444,3)
REAL CXO13(444,3),CXO14(444,3),BDLOW(3),BDUP(3),GENPT(3)
REAL PARAMS(3),HESS(6),GRAD(3),WK10(9),ZOSET1(4),ZOSET2(4)
REAL ZOSET3(4),ZOSET4(4),ZOT1(4,4),Z1T1(4,4),Z2T1(4,4),Z3T1(4,4)
REAL YO11(431),YO12(431),YO13(431),YO14(431),Y111(431),Y112(431)
REAL Y113(431),Y114(431),Y211(431),Y212(431),Y213(431),Y214(431)
REAL Y311(431),Y312(431),Y313(431),Y314(431),YK(4,4),DIFF1(4,4)
REAL DIFF2(4,4),DIFF3(4,4),GNATIV(4,4),COEFF1,COEFF2,COEFF3
REAL BIAS00(4,4),BIAS01(4,4),BIAS02(4,4),BIAS03(4,4),BIAS10(4,4)
REAL BIAS11(4,4),BIAS12(4,4),BIAS13(4,4),BIAS20(4,4),BIAS21(4,4)
REAL BIAS22(4,4),BIAS23(4,4),BIAS30(4,4),BIAS31(4,4),BIAS32(4,4)
REAL BIAS33(4,4),GOO(4,4),GO1(4,4),GO2(4,4),GO3(4,4),G10(4,4)
REAL G11(4,4),G12(4,4),G13(4,4),G20(4,4),G21(4,4),G22(4,4)
REAL G23(4,4),G30(4,4),G31(4,4),G32(4,4),G33(4,4),DELPAR(3,1)
REAL GAMOO(4,4),GAMOD(4,12),GAMDO(12,4),GAMDD(12,12),COEFF4
REAL U,UPAR(8),Z,FJ3,GPMEGA
EXTERNAL F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,FTHETA
COMMON /APPLE/TABSC,Y,NTAB20,CY,IC20
COMMON /BANANA/GPMEG2,GPMEG4
COMMON /CHERRY/XO14,CXO14
COMMON /GUAVA/XO13,CXO13
COMMON /KIWI/XO11,CXO11
COMMON /MANGO/UPAR
COMMON /PAPAYA/DELPAR
COMMON /PEACH/GAMOO,GAMOD,GAMDO,GAMDD
COMMON /PEAR/XO12,CXO12
```

```
C ***************************************************************************
C
C  GET ENDTIME T1 FOR H-IDENTIFICATION AND VALUES OF
C  (OMEGA 0)**2 AND (OMEGA 0)**4 .
C
      WRITE(1,10)
 10   FORMAT('DESIRED ENDPOINT OF INTEGRATION (F6.2)=')
      READ(1,20) T1
 20   FORMAT(F6.2)
      OMEGA=2.0*3.14159265359/T1
      OMEG2=OMEGA*OMEGA
      OMEG4=OMEG2*OMEG2
C ***************************************************************************
C
C  DEFINE INPUT FUNCTION U(T) VIA PARAMETERS.  U(T) WILL TAKE THE FORM:
C
C   U(T)=A*EXP(B*T)*SIN(C*T+D) + E*EXP(F*T)*SIN(G*T+H) , A-H USER SUPPLIED HERE.
C
      WRITE(1,21)
 21   FORMAT('INPUT U(T)=A*EXP(B*T)*SIN(C*T+D)+E*EXP(F*T)*SIN(G*T+H)')
      WRITE(1,22)
 22   FORMAT('DEFINE (USING 2F8.3) AMPLITUDES',/,'A        E')
      READ(1,23) UPAR(1),UPAR(5)
 23   FORMAT(2F8.3)
      WRITE(1,24)
 24   FORMAT('DEFINE (USING 2F8.3) EXPONL FACTORS',/,'B        F')
      READ(1,23) UPAR(2),UPAR(6)
      WRITE(1,25)
 25   FORMAT('DEFINE (USING 2F8.3) PERIODS',/,'C        G')
      READ(1,23) UPAR(3),UPAR(7)
      WRITE(1,26)
 26   FORMAT('DEFINE (USING 2F8.3) PHASE SHIFTS',/,'D        H')
      READ(1,23) UPAR(4),UPAR(8)
C ***************************************************************************
C
C  INITIALIZE FOR GEAR TO GET STATE SOLUTION OF ORIGINAL SYSTEM.
C
      M=2
      MI=1
      IX=1
      OUTER=0
      NEQO=2
      T=0.0
      TL=0.000001
      H=0.000000001
      XORG(1)=0.0
      XORG(2)=0.5
```

```
      NPART=430
      NTAB=NPART+1
      NDUM=NTAB
      NTAB8=NTAB+8
      NTAB14=NTAB+14
      NTAB20=NTAB+20
      NDUM14=NDUM+14
      NDUM20=NDUM+20
C ***************************************************************************
C
C  INTEGRATE ORIGINAL COUPLED SYSTEM AND GET OUTPUT VARIABLE Y AT
C  NTAB20 TABSC-KNOTS.
C
      DELTAT=T1/(FLOAT(NPART))
      TABSC(1)=0.000000001
      TEND=TABSC(1)
      DUMT(1)=TABSC(1)
      CALL DGEAR(NEQ0,F1,F2,T,H,XORG,TEND,TL,M,MI,IX,IWK1,WK1,IR1)
      CALL FOUTPT(NEQ0,TEND,XORG,Z)
      Y(1)=Z
      DUMY(1)=Y(1)
      IF(IR1.GT.128) CALL EGRESS(IR1,T,TEND,NEQ0,XORG,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
      DO 30 LOOP=2,NTAB20
      LOOPM1=LOOP-1
      TEND=(FLOAT(LOOPM1))*DELTAT
      TABSC(LOOP)=TEND
      DUMT(LOOP)=TABSC(LOOP)
      CALL DGEAR(NEQ0,F1,F2,T,H,XORG,TEND,TL,M,MI,IX,IWK1,WK1,IR1)
      CALL FOUTPT(NEQ0,TEND,XORG,Z)
      Y(LOOP)=Z
      DUMY(LOOP)=Y(LOOP)
      IF(IR1.GT.128) CALL EGRESS(IR1,T,TEND,NEQ0,XORG,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
 30   CONTINUE
      WRITE(1,32) IR1
 32   FORMAT('ORIGINAL SYSTEM INTEGRATION COMPLETED.  FINAL ERR=',I5)
C ***************************************************************************
C
C  COMPUTE SPLINE COEFFICIENTS FOR OUTPUT Y FOR LATER INTERPOLATION.
C
      IC=NPART
      IDUM=IC
      IC14=IC+14
      IC20=IC+20
      IDUM14=IDUM+14
      IDUM20=IDUM+20
      CALL ICSCCU(DUMT,DUMY,NDUM20,CDUMY,IDUM20,IR2)
```

```
C ***********************************************************************************
C
C   COPY COEFFICIENTS INTO CY.
        DO 35 LOOP=1,IC20
        DO 34 LOOP1=1,3
        CY(LOOP,LOOP1)=CDUMY(LOOP,LOOP1)
 34     CONTINUE
 35     CONTINUE
C ***********************************************************************************
C
C   INITIALIZE FOR GEAR TO GET XO VARIABLES AT NTAB14 TABSC-KNOTS.
C   INTEGRATION MUST BE DONE IN FOUR SETS DUE TO LIMITATIONS IN DGEAR.
C
C   INTEGRATE SET 1.
C
        NEQXOS=4
        T=0.0
        DO 40 LOOP=1,NEQXOS
        XOSET1(LOOP)=0.0
 40     CONTINUE
        TL=0.000001
        H=0.000000001
        OUTER=0
        M=2
        MI=1
        IX=1
        DO 45 LOOP=1,NTAB14
        TEND=TABSC(LOOP)
        CALL DGEAR(NEQXOS,F3,F4,T,H,XOSET1,TEND,TL,M,MI,IX,IWK2,WK2,IR3)
        IF(IR3.GT.128) CALL EGRESS(IR3,T,TEND,NEQXOS,XOSET1,H,IX,OUTER)
        IF(OUTER.EQ.1) CALL EXIT
        XO11(LOOP)=XOSET1(1)
        DUM1(LOOP)=XO11(LOOP)
        XO21(LOOP)=XOSET1(2)
        XO31(LOOP)=XOSET1(3)
        XO41(LOOP)=XOSET1(4)
 45     CONTINUE
        WRITE(1,46) IR3
 46     FORMAT('INTEGRATION OF XO SET 1 COMPLETED.  FINAL ERR=',I5)
C ***********************************************************************************
C
C   INTEGRATE SET 2.
C
        T=0.0
        DO 50 LOOP=1,NEQXOS
        XOSET2(LOOP)=0.0
 50     CONTINUE
```

```
      OUTER=0
      TL=0.000001
      H=0.000000001
      M=2
      MI=1
      IX=1
      DO 55 LOOP=1,NTAB14
      TEND=TABSC(LOOP)
      CALL DGEAR(NEQXOS,F5,F4,T,H,XOSET2,TEND,TL,M,MI,IX,IWK3,WK3,IR4)
      IF(IR4.GT.128) CALL EGRESS(IR4,T,TEND,NEQXOS,XOSET2,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
      XO12(LOOP)=XOSET2(1)
      DUM2(LOOP)=XO12(LOOP)
      XO22(LOOP)=XOSET2(2)
      XO32(LOOP)=XOSET2(3)
      XO42(LOOP)=XOSET2(4)
 55   CONTINUE
      WRITE(1,56) IR4
 56   FORMAT('INTEGRATION OF XO SET 2 COMPLETED.  FINAL ERR=',I5)
C  ***********************************************************************
C
C   INTEGRATE SET 3.
C
      T=0.0
      DO 60 LOOP=1,NEQXOS
      XOSET3(LOOP)=0.0
 60   CONTINUE
      OUTER=0
      TL=0.000001
      H=0.000000001
      M=2
      MI=1
      IX=1
      DO 65 LOOP=1,NTAB14
      TEND=TABSC(LOOP)
      CALL DGEAR(NEQXOS,F6,F4,T,H,XOSET3,TEND,TL,M,MI,IX,IWK4,WK4,IR5)
      IF(IR5.GT.128) CALL EGRESS(IR5,T,TEND,NEQXOS,XOSET3,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
      XO13(LOOP)=XOSET3(1)
      DUM3(LOOP)=XO13(LOOP)
      XO23(LOOP)=XOSET3(2)
      XO33(LOOP)=XOSET3(3)
      XO43(LOOP)=XOSET3(4)
 65   CONTINUE
      WRITE(1,66) IR5
 66   FORMAT('INTEGRATION OF XO SET 3 COMPLETED.  FINAL ERR=',I5)
```

```
C ***************************************************************************
C
C   INTEGRATE SET 4.
C
      T=0.0
      DO 70 LOOP=1,NEGXOS
      XOSET4(LOOP)=0.0
 70   CONTINUE
      OUTER=0
      TL=0.000001
      H=0.000000001
      M=2
      MI=1
      IX=1
      DO 75 LOOP=1,NTAB14
      TEND=TABSC(LOOP)
      CALL DGEAR(NEQXOS,F7,F4,T,H,XOSET4,TEND,TL,M,MI,IX,IWK5,WK5,IR6)
      IF(IR6.GT.128) CALL EGRESS(IR6,T,TEND,NEQXOS,XOSET4,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
      XO14(LOOP)=XOSET4(1)
      DUM4(LOOP)=XO14(LOOP)
      XO24(LOOP)=XOSET4(2)
      XO34(LOOP)=XOSET4(3)
      XO44(LOOP)=XOSET4(4)
      DUMT14(LOOP)=DUMT(LOOP)
 75   CONTINUE
      WRITE(1,76) IR6
 76   FORMAT('INTEGRATION OF XO SET 4 COMPLETED.  FINAL ERR=',I5)
C ***************************************************************************
C
C   GET SPLINE COEFFICIENTS FOR LATER INTERPOLATION.
C
      CALL ICSCCU(DUMT14,DUM1,NDUM14,CDUM1,IDUM14,IR7)
      CALL ICSCCU(DUMT14,DUM2,NDUM14,CDUM2,IDUM14,IR8)
      CALL ICSCCU(DUMT14,DUM3,NDUM14,CDUM3,IDUM14,IR9)
      CALL ICSCCU(DUMT14,DUM4,NDUM14,CDUM4,IDUM14,IR10)
C ***************************************************************************
C
C   COPY COEFFICIENTS INTO CX ARRAYS.
C
      DO 78 LOOP=1,IC14
      DO 77 LOOP1=1,3
      CXO11(LOOP,LOOP1)=CDUM1(LOOP,LOOP1)
      CXO12(LOOP,LOOP1)=CDUM2(LOOP,LOOP1)
      CXO13(LOOP,LOOP1)=CDUM3(LOOP,LOOP1)
      CXO14(LOOP,LOOP1)=CDUM4(LOOP,LOOP1)
 77   CONTINUE
 78   CONTINUE
```

```
C  ***************************************************************************
C
C   INITIALIZE FOR GEAR TO GET ZO VARIABLES--ONLY NEED ZO(T1).
C   INTEGRATION MUST BE DONE IN FOUR SETS DUE TO LIMITATIONS IN DGEAR.
C
C   INTEGRATE SET 1 AND GET ZO(T1) VALUES FOR SET 1.
C
       T=0.0
       NEQZOS=4
       DO 80 LOOP=1,NEQZOS
       ZOSET1(LOOP)=0.0
 80    CONTINUE
       OUTER=0
       TL=0.000001
       H=0.000000001
       M=2
       MI=1
       IX=1
       DO 85 LOOP=1,NTAB8
       TEND=TABSC(LOOP)
       CALL DGEAR(NEQZOS,F8,F9,T,H,ZOSET1,TEND,TL,M,MI,IX,IWK6,WK6,IR11)
       IF(IR11.GT.128) CALL EGRESS(IR11,T,TEND,NEQZOS,ZOSET1,H,IX,OUTER)
       IF(OUTER.EQ.1) CALL EXIT
       IF(LOOP.NE.NTAB) GO TO 85
       ZOT1(1,1)=ZOSET1(1)
       ZOT1(2,1)=ZOSET1(2)
       ZOT1(3,1)=ZOSET1(3)
       ZOT1(4,1)=ZOSET1(4)
 85    CONTINUE
       WRITE(1,86) IR11
 86    FORMAT('INTEGRATION OF ZO SET 1 COMPLETED.  FINAL ERR=',I5)
C  ***************************************************************************
C
C   INTEGRATE SET 2 AND GET ZO(T1) VALUES FOR SET 2.
C
       T=0.0
       DO 90 LOOP=1,NEQZOS
       ZOSET2(LOOP)=0.0
 90    CONTINUE
       TL=0.000001
       OUTER=0
       H=0.000000001
```

```
      M=2
      MI=1
      IX=1
      DO 95 LOOP=1,NTAB8
      TEND=TABSC(LOOP)
      CALL DGEAR(NEQZOS,F10,F9,T,H,ZOSET2,TEND,TL,M,MI,IX,IWK7,WK7,IR12)
      IF(IR12.GT.128) CALL EGRESS(IR12,T,TEND,NEQZOS,ZOSET2,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
      IF(LOOP.NE.NTAB) GO TO 95
      ZOT1(1,2)=ZOSET2(1)
      ZOT1(2,2)=ZOSET2(2)
      ZOT1(3,2)=ZOSET2(3)
      ZOT1(4,2)=ZOSET2(4)
 95   CONTINUE
      WRITE(1,96) IR12
 96   FORMAT('INTEGRATION OF ZO SET 2 COMPLETED.  FINAL ERR=',I5)
C  ***************************************************************
CC
CC   INTEGRATE SET 3 AND GET ZO(T1) VALUES FOR SET 3.
CC
      T=0.0
      DO 100 LOOP=1,NEQZOS
      ZOSET3(LOOP)=0.0
 100  CONTINUE
      OUTER=0
      TL=0.000001
      H=0.000000001
      M=2
      MI=1
      IX=1
      DO 105 LOOP=1,NTAB8
      TEND= TABSC(LOOP)
      CALL DGEAR(NEQZOS,F11,F9,T,H,ZOSET3,TEND,TL,M,MI,IX,IWK8,WK8,IR13)
      IF(IR13.GT.128) CALL EGRESS(IR13,T,TEND,NEQZOS,ZOSET3,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
      IF(LOOP.NE.NTAB) GO TO 105
      ZOT1(1,3)=ZOSET3(1)
      ZOT1(2,3)=ZOSET3(2)
      ZOT1(3,3)=ZOSET3(3)
      ZOT1(4,3)=ZOSET3(4)
 105  CONTINUE
      WRITE(1,106) IR13
 106  FORMAT('INTEGRATION OF ZO SET 3 COMPLETED.  FINAL ERR=',I5)
```

```
C ***********************************************************************
C
C  INTEGRATE SET 4 AND GET ZO(T1) VALUES FOR SET 4.
C
      T=0.0
      DO 110 LOOP=1,NEQZOS
      ZOSET4(LOOP)=0.0
 110  CONTINUE
      OUTER=0
      TL=0.000001
      H=0.000000001
      M=2
      MI=1
      IX=1
      DO 115 LOOP=1,NTAB8
      TEND=TABSC(LOOP)
      CALL DGEAR(NEQZOS,F12,F9,T,H,ZOSET4,TEND,TL,M,MI,IX,IWK9,WK9,IR14)
      IF(IR14.GT.128) CALL EGRESS(IR14,T,TEND,NEQZOS,ZOSET4,H,IX,OUTER)
      IF(OUTER.EQ.1) CALL EXIT
      IF(LOOP.NE.NTAB) GO TO 115
      ZOT1(1,4)=ZOSET4(1)
      ZOT1(2,4)=ZOSET4(2)
      ZOT1(3,4)=ZOSET4(3)
      ZOT1(4,4)=ZOSET4(4)
 115  CONTINUE
      WRITE(1,116) IR14
 116  FORMAT('INTEGRATION OF ZO SET 4 COMPLETED.  FINAL ERR=',I5)
C ***********************************************************************
C
C  GET YO VALUES AT NTAB TABSC-KNOTS.
C
      DO 120 LOOP=1,NTAB
      T=TABSC(LOOP)
      CALL UINPUT(T,U)
      YO11(LOOP)=XO11(LOOP)+Y(LOOP)
      YO12(LOOP)=XO12(LOOP)
      YO13(LOOP)=XO13(LOOP)-U
      YO14(LOOP)=XO14(LOOP)
 120  CONTINUE
```

```
C ***********************************************************************
C
C  GET Y1 VARIABLES AT NTAB TABSC-KNOTS.
C
      COEFF1=-1.25/QPMEG2
      COEFF2=-0.25/QPMEG4
      DO 121 LOOP=1,NTAB
      Y111(LOOP)=COEFF1*X021(LOOP)+COEFF2*X041(LOOP)
      Y112(LOOP)=COEFF1*X022(LOOP)+COEFF2*X042(LOOP)
      Y113(LOOP)=COEFF1*X023(LOOP)+COEFF2*X043(LOOP)
      Y114(LOOP)=COEFF1*X024(LOOP)+COEFF2*X044(LOOP)
 121  CONTINUE
C ***********************************************************************
C
C  FORM INVERSE OF THE TRANSPOSE OF (-A).
C
      DO 123 LOOP=1,4
      DO 122 LOOP1=1,4
      YK(LOOP,LOOP1)=0.0
      QNATIV(LOOP,LOOP1)=0.0
 122  CONTINUE
 123  CONTINUE
      QNATIV(1,2)=-1.0
      QNATIV(2,1)=-1.0*COEFF1
      QNATIV(2,3)=-1.0
      QNATIV(3,4)=-1.0
      QNATIV(4,1)=-1.0*COEFF2
C ***********************************************************************
C
C  GET Z1(T1) VALUES.
C
      YK(1,1)=Y111(NTAB)
      YK(1,2)=Y112(NTAB)
      YK(1,3)=Y113(NTAB)
      YK(1,4)=Y114(NTAB)
      CALL MSUB(DIFF1,ZOT1,YK,4,4)
      CALL MMLT(Z1T1,QNATIV,DIFF1,4,4,4)
C ***********************************************************************
C
C  GET Y2 VARIABLES AT NTAB TABSC-KNOTS.
C
      DO 124 LOOP=1,NTAB
      Y211(LOOP)=COEFF1*X011(LOOP)+COEFF2*X031(LOOP)
      Y212(LOOP)=COEFF1*X012(LOOP)+COEFF2*X032(LOOP)
      Y213(LOOP)=COEFF1*X013(LOOP)+COEFF2*X033(LOOP)
      Y214(LOOP)=COEFF1*X014(LOOP)+COEFF2*X034(LOOP)
 124  CONTINUE
```

```
C ***************************************************************************
C
C  GET Z2(T1) VALUES.
C
      YK(1,1)=Y211(NTAB)
      YK(1,2)=Y212(NTAB)
      YK(1,3)=Y213(NTAB)
      YK(1,4)=Y214(NTAB)
      CALL MSUB(DIFF2,Z1T1,YK,4,4)
      CALL MMLT(Z2T1,GNATIV,DIFF2,4,4,4)
C ***************************************************************************
C
C  GET Y3 VARIABLES AT NTAB TABSC-KNOTS.
C
      COEFF3=21.0/(16.0*GPMEG4)
      COEFF4=5.0/(16.0*GPMEG4*GPMEG2)
      DO 125 LOOP=1,NTAB
      Y311(LOOP)=COEFF3*X021(LOOP)+COEFF4*X041(LOOP)
      Y312(LOOP)=COEFF3*X022(LOOP)+COEFF4*X042(LOOP)
      Y313(LOOP)=COEFF3*X023(LOOP)+COEFF4*X043(LOOP)
      Y314(LOOP)=COEFF3*X024(LOOP)+COEFF4*X044(LOOP)
125   CONTINUE
C ***************************************************************************
C
C  GET Z3(T1) VALUES.
C
      YK(1,1)=Y311(NTAB)
      YK(1,2)=Y312(NTAB)
      YK(1,3)=Y313(NTAB)
      YK(1,4)=Y314(NTAB)
      CALL MSUB(DIFF3,Z2T1,YK,4,4)
      CALL MMLT(Z3T1,GNATIV,DIFF3,4,4,4)
C ***************************************************************************
C
C  GET BIAS MATRICES.
C
      CALL BIAS(Z0T1,Z0T1,T1,BIAS00)
      CALL BIAS(Z0T1,Z1T1,T1,BIAS01)
      CALL BIAS(Z0T1,Z2T1,T1,BIAS02)
      CALL BIAS(Z0T1,Z3T1,T1,BIAS03)
      CALL BIAS(Z1T1,Z0T1,T1,BIAS10)
      CALL BIAS(Z1T1,Z1T1,T1,BIAS11)
      CALL BIAS(Z1T1,Z2T1,T1,BIAS12)
      CALL BIAS(Z1T1,Z3T1,T1,BIAS13)
      CALL BIAS(Z2T1,Z0T1,T1,BIAS20)
      CALL BIAS(Z2T1,Z1T1,T1,BIAS21)
      CALL BIAS(Z2T1,Z2T1,T1,BIAS22)
```

```
      CALL BIAS(Z2T1,Z3T1,T1,BIAS23)
      CALL BIAS(Z3T1,Z0T1,T1,BIAS30)
      CALL BIAS(Z3T1,Z1T1,T1,BIAS31)
      CALL BIAS(Z3T1,Z2T1,T1,BIAS32)
      CALL BIAS(Z3T1,Z3T1,T1,BIAS33)
C ***********************************************************************
C
C  GET GAMMA MATRICES.
C
      N=NTAB
      D=DELTAT
      CALL GAMMA(N,D,Y011,Y012,Y013,Y014,Y011,Y012,Y013,Y014,BIAS00,G00)
      CALL GAMMA(N,D,Y011,Y012,Y013,Y014,Y111,Y112,Y113,Y114,BIAS01,G01)
      CALL GAMMA(N,D,Y011,Y012,Y013,Y014,Y211,Y212,Y213,Y214,BIAS02,G02)
      CALL GAMMA(N,D,Y011,Y012,Y013,Y014,Y311,Y312,Y313,Y314,BIAS03,G03)
      CALL GAMMA(N,D,Y111,Y112,Y113,Y114,Y011,Y012,Y013,Y014,BIAS10,G10)
      CALL GAMMA(N,D,Y111,Y112,Y113,Y114,Y111,Y112,Y113,Y114,BIAS11,G11)
      CALL GAMMA(N,D,Y111,Y112,Y113,Y114,Y211,Y212,Y213,Y214,BIAS12,G12)
      CALL GAMMA(N,D,Y111,Y112,Y113,Y114,Y311,Y312,Y313,Y314,BIAS13,G13)
      CALL GAMMA(N,D,Y211,Y212,Y213,Y214,Y011,Y012,Y013,Y014,BIAS20,G20)
      CALL GAMMA(N,D,Y211,Y212,Y213,Y214,Y111,Y112,Y113,Y114,BIAS21,G21)
      CALL GAMMA(N,D,Y211,Y212,Y213,Y214,Y211,Y212,Y213,Y214,BIAS22,G22)
      CALL GAMMA(N,D,Y211,Y212,Y213,Y214,Y311,Y312,Y313,Y314,BIAS23,G23)
      CALL GAMMA(N,D,Y311,Y312,Y313,Y314,Y011,Y012,Y013,Y014,BIAS30,G30)
      CALL GAMMA(N,D,Y311,Y312,Y313,Y314,Y111,Y112,Y113,Y114,BIAS31,G31)
      CALL GAMMA(N,D,Y311,Y312,Y313,Y314,Y211,Y212,Y213,Y214,BIAS32,G32)
      CALL GAMMA(N,D,Y311,Y312,Y313,Y314,Y311,Y312,Y313,Y314,BIAS33,G33)
C ***********************************************************************
C
C  FORM GAMMA OO, OD, DO, AND DD MATRICES.
C
      DO 131 LOOP1=1,4
      DO 130 LOOP=1,4
      IA=LOOP1+4
      IB=LOOP1+8
      JA=LOOP+4
      JB=LOOP+8
      GAMOO(LOOP1,LOOP)=G00(LOOP1,LOOP)
      GAMOD(LOOP1,LOOP)=G01(LOOP1,LOOP)
      GAMOD(LOOP1,JA)=G02(LOOP1,LOOP)
      GAMOD(LOOP1,JB)=G03(LOOP1,LOOP)
      GAMDO(LOOP1,LOOP)=G10(LOOP1,LOOP)
      GAMDO(IA,LOOP)=G20(LOOP1,LOOP)
      GAMDO(IB,LOOP)=G30(LOOP1,LOOP)
      GAMDD(LOOP1,LOOP)=G11(LOOP1,LOOP)
      GAMDD(IA,LOOP)=G21(LOOP1,LOOP)
      GAMDD(IB,LOOP)=G31(LOOP1,LOOP)
      GAMDD(LOOP1,JA)=G12(LOOP1,LOOP)
```

```
      GAMDD(IA,JA)=G22(LOOP1,LOOP)
      GAMDD(IB,JA)=G32(LOOP1,LOOP)
      GAMDD(LOOP1,JB)=G13(LOOP1,LOOP)
      GAMDD(IA,JB)=G23(LOOP1,LOOP)
      GAMDD(IB,JB)=G33(LOOP1,LOOP)
  130 CONTINUE
  131 CONTINUE
C ***************************************************************************
C
C  DETERMINE POSITIVE DEFINITENESS OF GAMMA 00.
C
      CALL POSDEF(NDEF,IR15)
      IF(NDEF.EQ.0) WRITE(1,135)
  135 FORMAT('GAMMA 00 IS POSITIVE DEFINITE !!!!!!!!!!!!!!!!!!!!!!!')
      IF(NDEF.NE.0) WRITE(1,136) NDEF
  136 FORMAT('GAMMA 00 IS NOT POSITIVE DEFINITE.   NDEF=',I8)
      IF(IR15.EQ.0) WRITE(1,137)
  137 FORMAT('DETERMINANTS TO VERIFY THIS WERE FOUND SUCCESSFULLY. ')
      IF(IR15.NE.0) WRITE(1,138) IR15
  138 FORMAT('DETERMINANTS NOT FOUND SUCCESSFULLY.   IR15=',I8)
C ***************************************************************************
C
C  EXAMINE GAMMA 00 MATRIX.
C
      WRITE(1,140)
  140 FORMAT('GAMMA 00 MATRIX IS')
      DO 142 LOOP=1,4
      WRITE(1,141) GOO(LOOP,1),GOO(LOOP,2),GOO(LOOP,3),GOO(LOOP,4)
  141 FORMAT(4F19.7)
  142 CONTINUE
  149 WRITE(1,150)
  150 FORMAT('CHOICE TIME.   DEFINE SEARCH REGION=1,DO MINIMIZATION=2')
      READ(1,212) NCHOSE
      IF(NCHOSE.EQ.1) GO TO 151
      IF(NCHOSE.EQ.2) GO TO 285
      GO TO 149
C ***************************************************************************
C
C  DEFINE REGION IN 3-SPACE TO BE SEARCHED FOR GENERATING POINTS TO
C  GO INTO MINIMIZATION SCHEME.
C
  151 WRITE(1,155)
  155 FORMAT('REGION IN 3-SPACE TO BE SEARCHED IS READY FOR DEFINITION')
      DO 180 LOOP=1,3
      WRITE(1,160) LOOP
  160 FORMAT('PARAMETER ',I3,' BOUNDS ARE (2F10.5)',/,'LOWER      UPPER')
      READ(1,170) BDLOW(LOOP),BDUP(LOOP)
  170 FORMAT(2F10.5)
  180 CONTINUE
```

```
C ***************************************************************************
C
C   DO SEARCH IN DEFINED REGION.
C
      WRITE(1,190)
 190  FORMAT('NUMBER OF POINTS (I2) TO BE GENERATED (KEEP < 10) =')
      READ(1,200) NGENPT
 200  FORMAT(I2)
      WRITE(1,210)
 210  FORMAT('NUMBER OF SIGNIFICANT DIGITS OF ACCURACY (I1) =')
      READ(1,212) NSIG
 212  FORMAT(I1)
      WRITE(1,215)
 215  FORMAT('MAX NUMBER OF FNT EVALUATIONS (I5) (KEEP < 30000) =')
      READ(1,216) MX
 216  FORMAT(I5)
      IP=0
      NP=3
      WRITE(1,220)
 220  FORMAT(2X,'PARAM1',7X,'PARAM2',7X,'PARAM3',7X,'J3')
      DO 260 LOOP=1,NGENPT
      CALL ZSRCH(BDLOW,BDUP,NP,NGENPT,IP,GENPT,IWK10,IWK11,IR16)
      IOPT=0
      DO 230 LOOP1=1,3
      PARAMS(LOOP1)=GENPT(LOOP1)
 230  CONTINUE
      WRITE(1,240) GENPT(1),GENPT(2),GENPT(3)
 240  FORMAT(3F12.7,13X,'<---GENERATED PT.')
      CALL ZXMIN(FTHETA,NP,NSIG,MX,IOPT,PARAMS,HESS,GRAD,FJ3,WK10,IR17)
      WRITE(1,250) PARAMS(1),PARAMS(2),PARAMS(3),FJ3
 250  FORMAT(4F12.7,' <--SOLN W/GENPT')
 260  CONTINUE
 270  WRITE(1,280)
 280  FORMAT('CHOICE TIME.   DEFINE NEW REGION=1,MINIMIZE=2,EXIT=3')
      READ(1,212) NCHOSE
      IF(NCHOSE.EQ.1) GO TO 151
      IF(NCHOSE.EQ.2) GO TO 285
      IF(NCHOSE.EQ.3) CALL EXIT
      GO TO 270
C ***************************************************************************
C
C   DO MINIMIZATION FOR SUGGESTED INITIAL PARAMETER GUESS.
C
 285  WRITE(1,290)
 290  FORMAT('NOW TIME FOR MINIMIZATION.')
      DO 320 LOOP=1,3
      WRITE(1,300) LOOP
```

```
  300   FORMAT('INITIAL GUESS (F12.7) FOR PARAMETER ',I2,' =')
        READ(1,310) PARAMS(LOOP)
  310   FORMAT(F12.7)
  320   CONTINUE
        IOPT=0
        WRITE(1,210)
        READ(1,212) NSIG
        WRITE(1,215)
        READ(1,216) MX
        CALL ZXMIN(FTHETA,NP,NSIG,MX,IOPT,PARAMS,HESS,GRAD,FJ3,WK10,IR17)
        WRITE(1,220)
        WRITE(1,330) PARAMS(1),PARAMS(2),PARAMS(3),FJ3
  330   FORMAT(3F12.7,F20.13)
        WRITE(1,332)
  332   FORMAT(2X,'GRAD1',8X,'GRAD2',8X,'GRAD3')
        WRITE(1,335) GRAD(1),GRAD(2),GRAD(3)
  335   FORMAT(3F12.7)
        WRITE(1,340) WK10(2)
  340   FORMAT('NUMBER OF FUNCTION EVALUATIONS PERFORMED=',F8.2)
        WRITE(1,350) WK10(3)
  350   FORMAT('EST. OF NUMBER OF SIGNIF. DIGITS IN FINAL PARAMS=',F8.2)
        WRITE(1,360) DELPAR(1,1),DELPAR(2,1),DELPAR(3,1)
  360   FORMAT('DELTA(1)=',F12.6,'  DELTA(2)=',F12.6,'  DELTA(3)=',F12.6)
        GO TO 270
        END
```

```
      SUBROUTINE UINPUT(T,U)
      REAL T,U,UPAR(8),U1,U2
      COMMON /MANGO/UPAR
C ***************************************************************************
C
C   SUBROUTINE CONTAINS INPUT FUNCTION TO ORIGINAL COUPLED SYSTEM
C   OF EQUATIONS.
C
      U1=UPAR(1)*EXP(UPAR(2)*T)*SIN(UPAR(3)*T+UPAR(4))
      U2=UPAR(5)*EXP(UPAR(6)*T)*SIN(UPAR(7)*T+UPAR(8))
      U=U1+U2
      RETURN
      END
```

```
      SUBROUTINE FOUTPT(N,T,X,Y)
      INTEGER N
      REAL Y,X(N),T
C ***************************************************************************
C
C  SUBROUTINE CONTAINS OUTPUT EQUATION FOR ORIGINAL COUPLED SYSTEM
C  OF EQUATIONS.
C
      T=T
      Y=3.0*X(1)+2.0*X(2)
      RETURN
      END
```



```
      SUBROUTINE F1(N,T,X,XPRIME)
      INTEGER N
      REAL T,X(N),XPRIME(N),U
C ***************************************************************************
C
C  SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR ORIGINAL
C  COUPLED SYSTEM OF EQUATIONS THAT REPRESENT THE PROCESS FROM WHICH
C  MEASUREMENTS ARE BEING MADE FOR IDENTIFICATION OF THE SEPARATE
C  DECOUPLED SUBSYSTEMS.
C
      CALL UINPUT(T,U)
      XPRIME(1)=X(2)+U
      XPRIME(2)=-10.*X(1)-11.*X(2)+U
      RETURN
      END
```

```
      SUBROUTINE F2(N, T, X, PD)
      INTEGER N
      REAL T, X(N), PD(N, N)
C ***************************************************************************
C
C   SUBROUTINE CONTAINS JACOBIAN FOR ORIGINAL COUPLED SYSTEM OF EQUATIONS
C   REPRESENTING PROCESS BEHAVIOR.
C
      T=T
      PD(1, 1)=0.0
      PD(1, 2)=1.0
      PD(2, 1)=-10.
      PD(2, 2)=-11.
      RETURN
      END
```



```
      SUBROUTINE F3(N, T, X, XPRIME)
      INTEGER N, IR5, M, LOOP, NTAB20, IC20, L
      REAL T, X(N), XPRIME(N), S(1), R(1), Y(451), CY(450, 3), TABSC(451)
      REAL A(451), B(451), C(450, 3), GPMEG2, GPMEG4
      COMMON /APPLE/TABSC, Y, NTAB20, CY, IC20
      COMMON /BANANA/GPMEG2, GPMEG4
C ***************************************************************************
C
C   SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR XO SET1 INTEGRATION.
C
C   CREATE DUMMY ARRAYS FOR PASSING TO INTERPOLATION.
C
      M=NTAB20
      L=IC20
      DO 10 LOOP=1, L
      A(LOOP)=TABSC(LOOP)
      B(LOOP)=Y(LOOP)
      C(LOOP, 1)=CY(LOOP, 1)
      C(LOOP, 2)=CY(LOOP, 2)
      C(LOOP, 3)=CY(LOOP, 3)
 10   CONTINUE
```

```
      A(M)=TABSC(M)
      B(M)=Y(M)
C *************************************************************************
CCC
CCC   GET INTERPOLATED VALUE OF Y AT T.
CCC
      R(1)=T
      CALL ICSEVU(A,B,M,C,L,R,S,1,IR5)
C *************************************************************************
CCC
CCC   DIFFERENTIAL EQUATIONS FOR XO SET1.
CCC
      XPRIME(1)=X(2)
      XPRIME(2)=X(3)-5.0*QPMEG2*S(1)
      XPRIME(3)=X(4)
      XPRIME(4)=-4.0*QPMEG4*X(1)-5.0*QPMEG2*X(3)+21.0*QPMEG4*S(1)
      RETURN
      END
```

```
      SUBROUTINE F4(N,T,X,PD)
      INTEGER N,LOOP1,LOOP2
      REAL T,X(N),PD(N,N),QPMEG2,QPMEG4
      COMMON /BANANA/QPMEG2,QPMEG4
C *************************************************************************
CCC
CCC   SUBROUTINE CONTAINS JACOBIAN FOR XO INTEGRATION.
CCC
      T=T
      DO 10 LOOP1=1,N
      DO 9 LOOP2=1,N
      PD(LOOP1,LOOP2)=0.0
 9    CONTINUE
 10   CONTINUE
      PD(1,2)=1.0
      PD(2,3)=1.0
      PD(3,4)=1.0
      PD(4,1)=-4.0*QPMEG4
      PD(4,3)=-5.0*QPMEG2
      RETURN
      END
```

```
      SUBROUTINE F5(N,T,X,XPRIME)
      INTEGER N,IR5,M,LOOP,NTAB20,IC20,L
      REAL T,X(N),XPRIME(N),S(1),R(1),Y(451),CY(450,3),TABSC(451)
      REAL A(451),B(451),C(450,3),QPMEG2,QPMEG4
      COMMON /APPLE/TABSC,Y,NTAB20,CY,IC20
      COMMON /BANANA/QPMEG2,QPMEG4
C ***************************************************************************
C
C  SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR XO SET2 INTEGRATION.
C
C  CREATE DUMMY ARRAYS FOR PASSING TO INTERPOLATION.
C
      M=NTAB20
      L=IC20
      DO 10 LOOP=1,L
      A(LOOP)=TABSC(LOOP)
      B(LOOP)=Y(LOOP)
      C(LOOP,1)=CY(LOOP,1)
      C(LOOP,2)=CY(LOOP,2)
      C(LOOP,3)=CY(LOOP,3)
 10   CONTINUE
      A(M)=TABSC(M)
      B(M)=Y(M)
C ***************************************************************************
C
C  GET INTERPOLATED VALUE OF Y AT T.
C
      R(1)=T
      CALL ICSEVU(A,B,M,C,L,R,S,1,IR5)
C ***************************************************************************
C
C  DIFFERENTIAL EQUATIONS FOR XO SET2.
C
      XPRIME(1)=X(2)+S(1)
      XPRIME(2)=X(3)
      XPRIME(3)=X(4)-5.0*QPMEG2*S(1)
      XPRIME(4)=-4.0*QPMEG4*X(1)-5.0*QPMEG2*X(3)
      RETURN
      END
```

```
      SUBROUTINE F6(N, T, X, XPRIME)
      INTEGER N
      REAL QPMEG2, QPMEG4, T, X(N), XPRIME(N), U
      COMMON /BANANA/QPMEG2, QPMEG4
C ***********************************************************************
C
C  SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR XO SET3 INTEGRATION.
C
      CALL UINPUT(T, U)
      XPRIME(1)=X(2)
      XPRIME(2)=X(3)+5.0*QPMEG2*U
      XPRIME(3)=X(4)
      XPRIME(4)=-4.0*QPMEG4*X(1)-5.0*QPMEG2*X(3)-21.0*QPMEG4*U
      RETURN
      END
```

```
      SUBROUTINE F7(N, T, X, XPRIME)
      INTEGER N
      REAL QPMEG2, QPMEG4, T, X(N), XPRIME(N), U
      COMMON /BANANA/QPMEG2, QPMEG4
C ***********************************************************************
C
C  SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR XO SET4 INTEGRATION.
C
      CALL UINPUT(T, U)
      XPRIME(1)=X(2)-U
      XPRIME(2)=X(3)
      XPRIME(3)=X(4)+5.0*QPMEG2*U
      XPRIME(4)=-4.0*QPMEG4*X(1)-5.0*QPMEG2*X(3)
      RETURN
      END
```

```
      SUBROUTINE F8(N,T,X,XPRIME)
      INTEGER N,M20,LOOP1,LOOP,IC14,IC20,NTAB14,NTAB19,NTAB20,L20,IR16
      REAL T,X(N),XPRIME(N),TABSC(451),Y(451),CY(450,3),R(1),S(1),C(445)
      REAL XO11(445),CXO11(444,3),A(451),B(451),C1(444,3),C3(450,3)
      REAL S1(1),D(445),GPMEG2,GPMEG4
      COMMON /APPLE/TABSC,Y,NTAB20,CY,IC20
      COMMON /BANANA/GPMEG2,GPMEG4
      COMMON /KIWI/XO11,CXO11
C ***************************************************************************
CC
C   SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR ZO SET 1 INTEGRATION.
CC
C   CREATE DUMMY ARRAYS FOR PASSING TO INTERPOLATION.
C
      NTAB14=NTAB20-6
      NTAB19=NTAB20-1
      IC14=IC20-6
      L20=IC20
      M20=NTAB20
      DO 10 LOOP=1,IC14
      A(LOOP)=TABSC(LOOP)
      B(LOOP)=Y(LOOP)
      C(LOOP)=TABSC(LOOP)
      D(LOOP)=XO11(LOOP)
      DO 9 LOOP1=1,3
      C1(LOOP,LOOP1)=CXO11(LOOP,LOOP1)
      C3(LOOP,LOOP1)=CY(LOOP,LOOP1)
 9    CONTINUE
 10   CONTINUE
      DO 20 LOOP=NTAB14,NTAB19
      A(LOOP)=TABSC(LOOP)
      B(LOOP)=Y(LOOP)
      DO 19 LOOP1=1,3
      C3(LOOP,LOOP1)=CY(LOOP,LOOP1)
 19   CONTINUE
 20   CONTINUE
      A(NTAB20)=TABSC(NTAB20)
      B(NTAB20)=Y(NTAB20)
      C(NTAB14)=TABSC(NTAB14)
      D(NTAB14)=XO11(NTAB14)
```

```
C ***********************************************************************************
C
C  GET INTERPOLATED VALUES.
C
      R(1)=T
      CALL ICSEVU(A,B,M20,C3,L20,R,S,1,IR16)
      CALL ICSEVU(C,D,NTAB14,C1,IC14,R,S1,1,IR16)
C ***********************************************************************************
C
C  DIFFERENTIAL EQUATIONS FOR ZO SET 1 INTEGRATION.
C
      XPRIME(1)=4.0*QPMEG4*X(4)+S(1)+S1(1)
      XPRIME(2)=-1.0*X(1)
      XPRIME(3)=-1.0*X(2)+5.0*QPMEG2*X(4)
      XPRIME(4)=-1.0*X(3)
      RETURN
      END
```



```
      SUBROUTINE F9(N,T,X,PD)
      INTEGER N,LOOP1,LOOP2
      REAL T,X(N),PD(N,N),QPMEG2,QPMEG4
      COMMON /BANANA/QPMEG2,QPMEG4
C ***********************************************************************************
C
C  SUBROUTINE CONTAINS JACOBIAN FOR ZO INTEGRATION.
C
      T=T
      DO 10 LOOP1=1,N
      DO 9 LOOP2=1,N
      PD(LOOP1,LOOP2)=0.0
 9    CONTINUE
 10   CONTINUE
      PD(1,4)=4.0*QPMEG4
      PD(2,1)=-1.0
      PD(3,2)=-1.0
      PD(3,4)=5.0*QPMEG2
      PD(4,3)=-1.0
      RETURN
      END
```

```
      SUBROUTINE F10(N,T,X,XPRIME)
      INTEGER N,LOOP1,LOOP2,NTAB20,IC20,IC14,NTAB14,IR15
      REAL GPMEG2,GPMEG4,T,X(N),XPRIME(N),Y(451),TABSC(451),CY(450,3)
      REAL XO12(445),CXO12(444,3),A(445),B(445),C(444,3),R(1),S(1)
      COMMON /APPLE/TABSC,Y,NTAB20,CY,IC20
      COMMON /BANANA/GPMEG2,GPMEG4
      COMMON /PEAR/XO12,CXO12
C *****************************************************************
C
C  SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR ZO SET2 INTEGRATION.
C
C  CREATE DUMMY ARRAYS FOR PASSING TO INTERPOLATION.
C
      NTAB14=NTAB20-6
      IC14=IC20-6
      DO 10 LOOP1=1,IC14
      A(LOOP1)=TABSC(LOOP1)
      B(LOOP1)=XO12(LOOP1)
      DO 9 LOOP2=1,3
      C(LOOP1,LOOP2)=CXO12(LOOP1,LOOP2)
 9    CONTINUE
 10   CONTINUE
      A(NTAB14)=TABSC(NTAB14)
      B(NTAB14)=XO12(NTAB14)
C *****************************************************************
C
C  GET INTERPOLATED VALUES.
C
      R(1)=T
      CALL ICSEVU(A,B,NTAB14,C,IC14,R,S,1,IR15)
C *****************************************************************
C
C  DIFFERENTIAL EQUATIONS FOR ZO SET 2 INTEGRATION.
C
      XPRIME(1)=4.0*GPMEG4*X(4)+S(1)
      XPRIME(2)=-1.0*X(1)
      XPRIME(3)=-1.0*X(2)+5.0*GPMEG2*X(4)
      XPRIME(4)=-1.0*X(3)
      RETURN
      END
```

```
      SUBROUTINE F11(N,T,X,XPRIME)
      INTEGER N,LOOP1,LOOP2,NTAB20,IC20,IC14,NTAB14,IR15
      REAL GPMEG2,GPMEG4,T,X(N),XPRIME(N),Y(451),TABSC(451),CY(450,3)
      REAL XO13(445),CXO13(444,3),A(445),B(445),C(444,3),R(1),S(1),U
      COMMON /APPLE/TABSC,Y,NTAB20,CY,IC20
      COMMON /BANANA/GPMEG2,GPMEG4
      COMMON /GUAVA/XO13,CXO13
C *****************************************************************************
C
C  SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR ZO SET3 INTEGRATION.
C
C  CREATE DUMMY ARRAYS FOR PASSING TO INTERPOLATION.
C
      NTAB14=NTAB20-6
      IC14=IC20-6
      DO 10 LOOP1=1,IC14
      A(LOOP1)=TABSC(LOOP1)
      B(LOOP1)=XO13(LOOP1)
      DO 9 LOOP2=1,3
      C(LOOP1,LOOP2)=CXO13(LOOP1,LOOP2)
 9    CONTINUE
 10   CONTINUE
      A(NTAB14)=TABSC(NTAB14)
      B(NTAB14)=XO13(NTAB14)
C *****************************************************************************
C
C  GET INTERPOLATED VALUES.
C
      R(1)=T
      CALL ICSEVU(A,B,NTAB14,C,IC14,R,S,1,IR15)
C *****************************************************************************
C
C  DIFFERENTIAL EQUATIONS FOR ZO SET 3 INTEGRATION.
C
      CALL UINPUT(T,U)
      XPRIME(1)=4.0*GPMEG4*X(4)+S(1)-U
      XPRIME(2)=-1.0*X(1)
      XPRIME(3)=-1.0*X(2)+5.0*GPMEG2*X(4)
      XPRIME(4)=-1.0*X(3)
      RETURN
      END
```

```
      SUBROUTINE F12(N,T,X,XPRIME)
      INTEGER N,LOOP1,LOOP2,NTAB20,IC20,IC14,NTAB14,IR15
      REAL GPMEG2,GPMEG4,T,X(N),XPRIME(N),Y(451),TABSC(451),CY(450,3)
      REAL XO14(445),CXO14(444,3),A(445),B(445),C(444,3),R(1),S(1)
      COMMON /APPLE/TABSC,Y,NTAB20,CY,IC20
      COMMON /BANANA/GPMEG2,GPMEG4
      COMMON /CHERRY/XO14,CXO14
C *****************************************************************************
CCC
CCCC  SUBROUTINE CONTAINS DIFFERENTIAL EQUATIONS FOR ZO SET4 INTEGRATION.
CCCC
CCC   CREATE DUMMY ARRAYS FOR PASSING TO INTERPOLATION.
C
      NTAB14=NTAB20-6
      IC14=IC20-6
      DO 10 LOOP1=1,IC14
      A(LOOP1)=TABSC(LOOP1)
      B(LOOP1)=XO14(LOOP1)
      DO 9 LOOP2=1,3
      C(LOOP1,LOOP2)=CXO14(LOOP1,LOOP2)
 9    CONTINUE
 10   CONTINUE
      A(NTAB14)=TABSC(NTAB14)
      B(NTAB14)=XO14(NTAB14)
C *****************************************************************************
CCC
CCC   GET INTERPOLATED VALUES.
C
      R(1)=T
      CALL ICSEVU(A,B,NTAB14,C,IC14,R,S,1,IR15)
C *****************************************************************************
CCC
CCC   DIFFERENTIAL EQUATIONS FOR ZO SET 4 INTEGRATION.
C
      XPRIME(1)=4.0*GPMEG4*X(4)+S(1)
      XPRIME(2)=-1.0*X(1)
      XPRIME(3)=-1.0*X(2)+5.0*GPMEG2*X(4)
      XPRIME(4)=-1.0*X(3)
      RETURN
      END
```

```
      SUBROUTINE EGRESS(L,B,C,K,D,E,M,N)
      INTEGER K,L,M,N,LOOP
      REAL B,C,D(K),E
C ***************************************************************************
C
C  SUBROUTINE DISPLAYS CURRENT VARIABLES AND ERROR VALUES.
C  CONTROL IS PASSED FROM MAIN PROGRAM TO THIS SUBROUTINE WHEN
C  ERROR IS ENCOUNTERED DURING DGEAR INTEGRATION.  CONTROL
C  IS THEN RETURNED TO MAIN PROGRAM AND EXIT IS MADE VIA ERROR SIGNAL N=1 .
C
      WRITE(1,10) L,M
 10   FORMAT('MALFNT: IER=',I6,' INDEX=',I6)
      WRITE(1,20) B,C,E
 20   FORMAT('T=',F12.7,' TEND=',F12.7,' H=',F12.7)
      N=1
      DO 40 LOOP=1,K
      WRITE(1,30) LOOP,D(LOOP)
 30   FORMAT('VAR(',I2,')=',F12.6)
 40   CONTINUE
      WRITE(1,50)
 50   FORMAT('PROGRAM IS NOW TERMINATED')
      RETURN
      END
```

```
      SUBROUTINE BIAS(A,B,T1,D)
      REAL A(4,4),B(4,4),D(4,4),T1,E(4,4),F(4,4),SCON
C ***************************************************************************
C
C  SUBROUTINE COMPUTES D = 2./T1 * A(TRANSPOSE) * B
C
      CALL MTRN(E,A,4)
      CALL MMLT(F,E,B,4,4,4)
      SCON=2.0/T1
      CALL MSCL(D,F,4,4,SCON)
      RETURN
      END
```

```
      SUBROUTINE GAMMA(N,D,YI1,YI2,YI3,YI4,YJ1,YJ2,YJ3,YJ4,BIJ,G)
      INTEGER N
      REAL YI1(N),YI2(N),YI3(N),YI4(N),YJ1(N),YJ2(N),YJ3(N),YJ4(N),D
      REAL CORREL,G(4,4),BIJ(4,4)
C ***************************************************************************
C
C  SUBROUTINE COMPUTES GAMMA MATRIX GIVEN YI,YJ TIME VECTORS AND
C  BIAS(I,J) MATRIX.  CALLS TO SUBROUTINE SIMPSN ARE MADE FOR COMPUTING
C  THE CORRELATION MATRIX.
C
      CALL SIMPSN(N,D,YI1,YJ1,CORREL)
      G(1,1)=CORREL-BIJ(1,1)
      CALL SIMPSN(N,D,YI1,YJ2,CORREL)
      G(1,2)=CORREL-BIJ(1,2)
      CALL SIMPSN(N,D,YI1,YJ3,CORREL)
      G(1,3)=CORREL-BIJ(1,3)
      CALL SIMPSN(N,D,YI1,YJ4,CORREL)
      G(1,4)=CORREL-BIJ(1,4)
      CALL SIMPSN(N,D,YI2,YJ1,CORREL)
      G(2,1)=CORREL-BIJ(2,1)
      CALL SIMPSN(N,D,YI2,YJ2,CORREL)
      G(2,2)=CORREL-BIJ(2,2)
      CALL SIMPSN(N,D,YI2,YJ3,CORREL)
      G(2,3)=CORREL-BIJ(2,3)
      CALL SIMPSN(N,D,YI2,YJ4,CORREL)
      G(2,4)=CORREL-BIJ(2,4)
      CALL SIMPSN(N,D,YI3,YJ1,CORREL)
      G(3,1)=CORREL-BIJ(3,1)
      CALL SIMPSN(N,D,YI3,YJ2,CORREL)
      G(3,2)=CORREL-BIJ(3,2)
      CALL SIMPSN(N,D,YI3,YJ3,CORREL)
      G(3,3)=CORREL-BIJ(3,3)
      CALL SIMPSN(N,D,YI3,YJ4,CORREL)
      G(3,4)=CORREL-BIJ(3,4)
      CALL SIMPSN(N,D,YI4,YJ1,CORREL)
      G(4,1)=CORREL-BIJ(4,1)
      CALL SIMPSN(N,D,YI4,YJ2,CORREL)
      G(4,2)=CORREL-BIJ(4,2)
      CALL SIMPSN(N,D,YI4,YJ3,CORREL)
      G(4,3)=CORREL-BIJ(4,3)
      CALL SIMPSN(N,D,YI4,YJ4,CORREL)
      G(4,4)=CORREL-BIJ(4,4)
      RETURN
      END
```

```
      SUBROUTINE FTHETA(N,X,F)
      INTEGER N,LOOP1,LOOP2,LOOP,IA,IB,IR
      REAL X(N),F,H(4,1),GDD(12,12),GOO(4,4),GOD(4,12),GDO(12,4),HT(1,4)
      REAL HBIG(12,3),HBIGT(3,12),GAMOO(4,4),GAMOD(4,12),GAMDO(12,4)
      REAL GAMDD(12,12),A(3,12),B(3,3),C(3,12),D(3,4),E(3,1),WK(4,6)
      REAL Q(1,4),G(12,1),P(4,1),R(1,1),DELPAR(3,1),S(1,1),EFFECT(3,3)
      COMMON /PAPAYA/DELPAR
      COMMON /PEACH/GAMOO,GAMOD,GAMDO,GAMDD
      DATA HBIG/36*0.0/
C *****************************************************************************
CC
CC   SUBROUTINE COMPUTES THE OBJECTIVE FUNCTION F GIVEN THE PARAMETER VECTOR X.
CC
CC   LOAD MATRICES.
C
      H(1,1)=1.0
      H(2,1)=X(1)
      H(3,1)=X(2)
      H(4,1)=X(3)
      DO 10 LOOP1=1,4
      HT(1,LOOP1)=H(LOOP1,1)
      DO 5 LOOP2=1,4
      GOO(LOOP1,LOOP2)=GAMOO(LOOP1,LOOP2)
 5    CONTINUE
      IA=LOOP1+4
      IB=LOOP1+8
      HBIG(LOOP1,1)=H(LOOP1,1)
      HBIG(IA,2)=H(LOOP1,1)
      HBIG(IB,3)=H(LOOP1,1)
 10   CONTINUE
      DO 21 LOOP1=1,3
      DO 20 LOOP2=1,12
      HBIGT(LOOP1,LOOP2)=HBIG(LOOP2,LOOP1)
 20   CONTINUE
 21   CONTINUE
      DO 31 LOOP1=1,4
      DO 30 LOOP2=1,12
      GOD(LOOP1,LOOP2)=GAMOD(LOOP1,LOOP2)
      GDO(LOOP2,LOOP1)=GAMDO(LOOP2,LOOP1)
 30   CONTINUE
 31   CONTINUE
      DO 36 LOOP1=1,12
      DO 35 LOOP2=1,12
      GDD(LOOP1,LOOP2)=GAMDD(LOOP1,LOOP2)
 35   CONTINUE
 36   CONTINUE
```

```
C ***********************************************************************
C
C  COMPUTE OBJECTIVE FUNCTION J3(OMEGA).
C
      CALL MMLT(A,HBIGT,GDD,3,12,12)
      CALL MMLT(B,A,HBIG,3,12,3)
      CALL MINV(EFFECT,B,3,WK,4,6,IR)
      IF(IR.EQ.1) WRITE(1,40) IR
 40   FORMAT(//,'INVERSE IN J3 IS SINGULAR!!! ERR=',I5,/,25X,'********')
      CALL MMLT(C,EFFECT,HBIGT,3,3,12)
      CALL MMLT(D,C,GDO,3,12,4)
      CALL MMLT(E,D,H,3,4,1)
      CALL MMLT(G,HBIG,E,12,3,1)
      CALL MMLT(P,GOD,G,4,12,1)
      CALL MMLT(R,HT,P,1,4,1)
      CALL MMLT(Q,HT,GOO,1,4,4)
      CALL MMLT(S,Q,H,1,4,1)
      F=S(1,1)-R(1,1)
      DO 50 LOOP=1,3
      DELPAR(LOOP,1)=-1.0*E(LOOP,1)
 50   CONTINUE
      RETURN
      END

      SUBROUTINE SIMPSN(N,D,GRAND1,GRAND2,CORREL)
      INTEGER N,LOOP,NM1
      REAL D,CORREL,GRAND1(N),GRAND2(N),SIM
C ***********************************************************************
C
C  SUBROUTINE USES SIMPSON'S RULE TO COMPUTE INTEGRAL FROM 0 TO T1 OF
C  GRAND1 * GRAND2  .     THIS VALUE IS THEN THE CORRELATION.
C
C  DO SIMPSON'S INTEGRATION OF GRAND1*GRAND2 AT NTAB-KNOTS.
C
      SIM=0.0
      NM1=N-1
      DO 10 LOOP=2,NM1
      SIM=SIM+2.0*GRAND1(LOOP)*GRAND2(LOOP)
 10   CONTINUE
      DO 20 LOOP=2,NM1,2
      SIM=SIM+2.0*GRAND1(LOOP)*GRAND2(LOOP)
 20   CONTINUE
      CORREL=(SIM+GRAND1(1)*GRAND2(1)+GRAND1(N)*GRAND2(N))*D/3.0
      RETURN
      END
```

```
      SUBROUTINE POSDEF(NDEF, IR)
      INTEGER LOOP1, LOOP2, IR, IRS, NDEF
      REAL G1(4,4), G2(3,3), WK1(4), DET, V
      REAL WK2(4), WK3(4), WK4(4), WK5(3), WK6(3), WK7(3), WK8(3)
      REAL GAMOO(4,4), GAMOD(4,12), GAMDO(12,4), GAMDD(12,12)
      COMMON /PEACH/GAMOO, GAMOD, GAMDO, GAMDD
C ***********************************************************************
C
C  SUBROUTINE COMPUTES THE DETERMINANT OF ALL PRINCIPAL LEADING MINORS
C  OF GAMMA OO AND RETURNS INDICATOR OF ITS POSITIVE DEFINITENESS AND
C  SUCCESS OF SUCH VERIFICATION.
C
      NDEF=0
      IRS=0
      DO 10 LOOP1=1,4
      DO 9 LOOP2=1,4
      G1(LOOP1,LOOP2)=GAMOO(LOOP1,LOOP2)
 9    CONTINUE
 10   CONTINUE
      CALL MDET(DET, G1, 4, WK1, WK2, WK3, WK4, IR)
      IF(DET.LT.0.0) NDEF=NDEF+16
      IF(IR.EQ.1) IRS=IRS+16
      DO 20 LOOP1=1,3
      DO 19 LOOP2=1,3
      G2(LOOP1,LOOP2)=GAMOO(LOOP1,LOOP2)
 19   CONTINUE
 20   CONTINUE
      CALL MDET(DET, G2, 3, WK5, WK6, WK7, WK8, IR)
      IF(DET.LT.0.0) NDEF=NDEF+9
      IF(IR.EQ.1) IRS=IRS+9
      DET=GAMOO(1,1)*GAMOO(2,2)-GAMOO(1,2)*GAMOO(2,1)
      IF(DET.LT.0.0) NDEF=NDEF+4
      V=GAMOO(1,1)
      IF(V.LT.0.0) NDEF=NDEF+1
      RETURN
      END
```

# LIST OF REFERENCES

## LIST OF REFERENCES


[AH] Ahlfors, L.V., Complex Analysis, 2nd Ed., McGraw-Hill Book Co., Inc., New York, 1966.

[AS-1] Aström, K.J. and P. Eykhoff, "System Identification--A Survey," Automatica, Vol. 7, No. 2, pp. 123-162, March, 1971.

[AS-2] Aström, K.J., "Maximum Likelihood and Prediction Error Methods," Automatica, Vol. 16, No. 5, pp. 551-574, Sept. 1980.

[BE] Beghelli, S. and R. Guidorzi, "A New Input-Output Canonical Form for Multivariable Systems," IEEE Transactions on Automatic Control, AC-21, No. 5, pp. 692-696, Oct. 1976.

[BL] Blum, J.R. and J.I. Rosenblatt, Probability and Statistics, W.B. Saunders Co., Philadelphia, 1972.

[CHE] Chen, C.F. and B.L. Phillips, "Accurate Determination of Complex Root Transfer Functions from Frequency Response Data," IEEE Transactions on Automatic Control, AC-10, No. 3, pp. 356-358, July 1965.

[CHO-1] Chow, J.H., J.J. Allemong and P.V. Kokotovic, "Singular Perturbation Analysis of Systems with Sustained High Frequency Oscillations," Automatica, Vol. 14, No. 3, pp. 271-279, May 1978.

[CHO-2] Chow, J.H. and P.V. Kokotovic, "A Decomposition of Near-Optimum Regulators for Systems with Slow and Fast Modes," IEEE Transactions on Automatic Control, AC-21, No. 5, pp. 701-705, Oct. 1976.

[CL] Clarke, D.W., "Generalized Least Squares Estimation of the Parameters of a Dynamic Model," IFAC Symposium on Identification in Automatic Control Systems, Prague, paper 3.17, 1967.

[CLA] Clasen, R.J., D. Garfinkel, N. Shapiro and G-C. Roman, "A Method for Solving Certain Stiff Differential Equations," SIAM Journal of Applied Mathematics, Vol. 34, No. 4, pp. 732-742, June 1978.

[DE] Desoer, C.A. and M.J. Shensa, "Networks with Very Small and Very Large Parasitics: Natural Frequencies and Stability," Proceedings of the IEEE, Vol. 58, No. 12, pp. 1933-1938, Dec. 1970.

[DO] Donoghue, P.J., System Identification by Bayesian Learning, Ph.D. Thesis, Michigan State University, 1968.

[FA] Fasol, K.H. and H.P. Jörgl, "Principles of Model Building and Identification," Automatica, Vol. 16, No. 5, pp. 505-518, Sept. 1980.

[GA] Gamkrelidze, R.V., Progress in Mathematics: Mathematical Analysis, Vol. 8, Plenum Press, New York, 1970.

[GAU] Gauss, K.F., "Teoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium," 1809. Reprinted translation: Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections, Dover Press, New York, 1963.

[GE] Gear, C.W., "The Automatic Integration of Stiff Oridinary Differential Equations," Information Processing, Vol. 1, No. 68, A.J.H. Morrell, ed., North-Holland, Amsterdam, pp. 187-193, 1969.

[GO] Godfrey, K.R., "Correlation Methods," Automatica, Vol. 16, No. 5, pp. 527-534, Sept. 1980.

[GOO] Goodwin, G.C. and R.L. Payne, Dynamic System Identification: Experiment Design and Data Analysis, Academic Press, New York, 1977.

[GR] Graupe, D., V.K. Jain and J. Salahi, "A Comparative Analysis of Various Least-Squares Identification Algorithms," Automatica, Vol. 16, No. 6, pp. 663-681, Nov. 1980.

[GRU] Grujic, L.T., "Uniform Asymptotic Stability of Non-Linear Singularly Perturbed General and Large-Scale Systems," International Journal of Control, Vol. 33, No. 3, pp. 481-504, March 1981.

[GU] Guidorzi, R.P., "Canonical Structures in the Identification of Multivariable Systems," Proceedings of the 3rd IFAC Symposium on Identification, The Hague/Delft, The Netherlands, paper TS-4.

[HA-1] Haddad, A.H. and P.V. Kokotovic, "Stochastic Control of Linear Singularly Perturbed Systems," IEEE Transactions on Automatic Control, AC-22, No. 5, pp. 815-821, Oct. 1977.

[HA-2] Haddad, A.H., "Linear Filtering of Singularly Perturbed Systems," IEEE Transactions on Automatic Control, AC-21, No. 4, pp. 515-519, Aug. 1976.

[HAL] Halmos, P.R., Introduction to Hilbert Space and the Theory of Spectral Multiplicity, Chelsea Publishing Co., New York, 1951.

[HE] Heinen, J.A., "The Relationship of the Initial Conditions of a Scalar Differential Equation to those of its State Variable Representation," International Journal of Control, Vol. 31, No. 4, pp. 705-707, 1980.

[HO] Hoppensteadt, F., "On Systems of Ordinary Differential Equations with Several Parameters Multiplying the Derivatives," Journal of Differential Equations, Vol. 5, No. 1, pp. 106-116, Jan. 1969.

[IMSL] The IMSL Library, Edition 6, International Math and Statistics Library, Inc., Houston, TX, 1977.

[IS] Isermann, R., "Practical Aspects of Process Identification," Automatica, Vol. 16, No. 5, pp. 575-587, Sept. 1980.

[JA] Javid, S.H., "Observing the Slow States of a Singularly Perturbed System," IEEE Transactions on Automatic Control, AC-25, No. 2, pp. 277-280, April 1980.

[JAC] Jacobs, D., Numerical Software Needs and Availability, Academic Press, New York, 1978.

[KA-1] Kalman, R.E., "Mathematical Description of Linear Dynamic Systems," SIAM Journal of Control, Vol. 1, No. 2, pp. 152-192, 1963.

[KA-2] Kalman, R.E., "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, 82, pp. 35-45, March 1960.

[KH] Khalil, H.K. and P.V. Kokotovic, "Control of Linear Systems with Multiparameter Singular Perturbations," Automatica, Vol. 15, No. 2, pp. 197-207, March 1979.

[KO-1] Kokotovic, P.V., J.J. Allemong, J.R. Winkelman and J.H. Chow, "Singular Perturbation and Iterative Separation of Time Scales," Automatica, Vol. 16, No. 1, pp. 23-33, Jan. 1980.

[KO-2] Kokotovic, P.V., R.E. O'Malley, Jr. and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory--An Overview," Automatica, Vol. 12, No. 2, pp. 123-132, March 1976.

[KO-3] Kokotovic, P.V. and P. Sannuti, "Singular Perturbation Method for Reducing the Model Order in Optimal Control Design," IEEE Transactions on Automatic Control, AC-13, No. 4, pp. 377-384, Aug. 1968.

[KO-4] Kokotovic, P.V. and A.H. Haddad, "Controllability and Time-Optimal Control of Systems with Slow and Fast Modes," IEEE Transactions on Automatic Control, AC-20, No. 1, pp. 111-113, Feb. 1975.

[KO-5] Kokotovic, P.V. and A.H. Haddad, "Singular Perturbations of a Class of Time-Optimal Controls," IEEE Transactions on Automatic Control, AC-20, No. 1, pp. 163-164, Feb. 1975.

[KO-6] Kokotovic, P.V. and R.A. Yackel, "Singular Perturbation of Linear Regulators: Basic Theorems," IEEE Transactions on Automatic Control, AC-17, No. 1, pp. 29-37, Feb. 1972.

[KO-7] Kokotovic, P.V., "A Riccati Equation for Block-Diagonalization of Ill-Conditioned Systems," IEEE Transactions on Automatic Control, AC-20, No. 6, pp. 812-814, Dec. 1975.

[KU] Kurz, H., R. Isermann and R. Schumann, "Experimental Comparison and Application of Various Parameter-Adaptive Control Algorithms," Automatica, Vol. 16, No. 2, pp. 117-133, March 1980.

[LE] Levadi, V.S., "Design of Input Signals for Parameter Estimation," IEEE Transactions on Automatic Control, AC-11, No. 2, pp. 205-211, April 1966.

[LEE] Lee, R.C.K., Optimal Estimation, Identification, and Control, Research Monograph No. 28, M.I.T. Press, Cambridge, MA, 1964.

[LEV] Levin, M.J., Estimation of the Characteristics of Linear Systems in the Presence of Noise, Eng. Sc. D. Dissertation, Dept. EE, Columbia University, New York, 1959.

[ME] Mendel, J., "On the Identifiability of State-Derivative-Coupled Systems," IEEE Transactions on Automatic Control, AC-20, No. 6, pp. 782-785, Dec. 1975.

[OG] Ogata, K., State Space Analysis of Control Systems, Prentice Hall, Englewood Cliffs, NJ, 1967.

[OM-1] O'Malley, R.E., Jr. and C.F. Kung, "The Singularly Perturbed Linear State Regulator Problem, II," SIAM Journal of Control, Vol. 13, No. 2, pp. 327-337, Feb. 1975.

[OM-2] O'Malley, R.E., Jr., "The Singularly Perturbed Linear State Regulator Problem," SIAM Journal of Control, Vol. 10, No. 3, pp. 399-413, Aug. 1972.

[OM-3] O'Malley, R.E., Jr., Introduction to Singular Perturbations, Academic Press, New York, 1974.

[OR] O'Reilly, J., "Full-Order Observers for a Class of Singularly Perturbed Linear Time-Varying Systems," International Journal of Control, Vol. 30, No. 5, pp. 745-756, Nov. 1979.

[PE-1] Peterka, V., "Bayesian System Identification," Survey Lecture S6, Proceedings of the 5th IFAC Symposium on Identification and System Parameter Estimation, Darmstadt, 1979.

[PE-2] Peterka, V. and A. Halousková, "Tally Estimation of Aström Model for Stochastic Systems," IFAC Symposium on Identification and Process Parameter Estimation, Prague, paper 2.3, 1970.

[PEA-1] Pearson, A.E., "Finite Time Interval Linear System Identification Without Initial State Estimation," Automatica, Vol. 12, pp. 577-587, Nov. 1976.

[PEA-2] Pearson, A.E., "Nonlinear System Identification With Limited Time Data," Automatica, Vol. 15, pp. 73-84, Jan. 1979.

[PEA-3] Pearson, A.E. and Y.K. Chin, "Identification of MIMO Systems with Partially Decoupled Parameters," IEEE Transactions on Automatic Control, AC-24, pp. 599-604, Aug. 1979.

[PEA-4] Pearson, A. and J. Mocenigo, "A Filter for Separating Time-Limited Modal Signals," IEEE Transactions on Automatic Control, AC-24, No. 6, pp. 926-932, Dec. 1979.

[PEA-5] Pearson, A., "Equation Error Identification With Modal Disturbances Suppressed," in Numerical Techniques for Stochastic Systems, Archetti, F. and M. Cugiani, eds., North Holland Publishing Co., 1980.

[PO-1] Porter, B., "Singular Perturbation Methods in the Design of Stabilizing Feedback Controllers for Multivariable Linear Systems," International Journal of Control, Vol. 20, No. 4, pp. 689-692, Oct. 1974.

[PO-2] Porter, B. and A.T. Shenton, "Singular Perturbation Analysis of the Transfer Function Matrices of a Class of Multivariable Linear Systems," International Journal of Control, Vol. 21, No. 4, pp. 655-660, April 1975.

[RA] Rake, H., "Step Response and Frequency Response Methods," Automatica, Vol. 16, No. 5, pp. 519-526, Sept. 1980.

[SA] Saksena, V.R. and J.B. Cruz, Jr., "Stabilization of Singularly Perturbed Linear Time-Invariant Systems Using Low-Order Observers," IEEE Transactions on Automatic Control, AC-26, No. 2, pp. 510-513, April 1981.

[SE] Sebald, A.V. and A.H. Haddad, "State Estimation for Singularly Perturbed Systems with Unknown Perturbation Parameter," IEEE Transactions on Automatic Control, AC-23, No. 3, pp. 464-469, June 1978.

[SEB] Seber, G., Linear Regression Analysis, John Wiley and Sons, Inc., New York, 1977.

[SH] Shinners, S.M., Modern Control Theory and Application, Addison-Wesley Publishing Co., Reading, MA, 1978.

[SHA] Shampine, L.F., and C.W. Gear, "A User's View of Solving Stiff Ordinary Differential Equations," SIAM Review, Vol. 21, No. 1, pp. 1-17, Jan. 1979.

[SK] Sklansky, J., "Learning Systems for Automatic Control," IEEE Transactions on Automatic Control, AC-11, No. 1, pp. 6-19, Jan. 1966.

[ST] Strejc, V., "Least Squares Parameter Estimation," Automatica, Vol. 16, No. 5, pp. 535-550, Sept. 1980.

[WA] Wasow, W., Asymptotic Expansions for Ordinary Differential Equations, Wiley Publishers, New York, 1965.

[WI] Wilde, R.R. and P.V. Kokotovic, "Stability of Singular Perturbed Systems and Networks with Parasitics," IEEE Transactions on Automatic Control, AC-17, No. 2, pp. 245-246, April 1972.

[WO] Wong, K.Y. and E. Polak, "Identification of Linear Discrete Time Systems Using the Instrumental Variable Method," IEEE Transactions on Automatic Control, AC-12, No. 6, pp. 707-718, Dec. 1967.

[WOL] Wolovich, W.A., Linear Multivariable Systems, Springer-Verlag, New York, 1974.

[YO] Young, P.C., "An Instrumental Variable Method for Real-time Identification of a Noisy Process," Automatica, Vol. 6, No. 2, pp. 271-287, March 1970.

[ZA] Zadeh, L.A., "From Circuit Theory to System Theory," Proceedings of the IRE, 50, pp. 856-865, May 1962.