Parameter Set Estimation of Time Varying Systems

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the Graduate

School of The Ohio State University

By

John Watkins, B.S., M.S.

* * * * *

The Ohio State University

1995

Dissertation Committee:

Stephen Yurkovich

Kevin M. Passino

Hitay Özbay

Approved by

Adviser

Department of Electrical Engineering

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

19980115 182

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1995	Final

4. TITLE AND SUBTITLE
Parameter Set Estimation of Time Varying Systems

5. FUNDING NUMBERS

6. AUTHORS
John Watkins

AFRL-SR-BL-TR-98-

0018

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Ohio State University

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
110 Duncan Avenue, Suite B-115
Bolling Air Force Base, DC 20332-8080

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
See attached.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Parameter Set Estimation of Time Varying Systems

By

John Watkins, Ph.D.

The Ohio State University, 1995

Stephen Yurkovich, Adviser

Parameter set estimation (PSE), a class of system identification schemes which aim at characterizing the uncertainty in the identification experiment, will play a vital role in robust identification for control. An important step in current research along these lines is development of PSE algorithms for systems which are time varying in nature; this is particularly true if the identified model set is to be used in an adaptive setting.

In this dissertation, the Optimum Volume Ellipsoid (OVE) algorithm for parameter set estimation of time-invariant systems is extended to time-varying systems. Building on this development of the OVE algorithm for Time-Varying systems (OVETV), two algorithms are also presented for reducing the computational complexity of the optimal time update equations. These algorithms, scalar addition and scalar multiplication, reduce the computational complexity of the time update equa-

tions by constraining the new ellipsoid to be parameterized by the previous ellipsoid and a single new parameter. Several examples are presented in detail.

Following this, extensions to the OVE and OVETV algorithms are presented. The algorithms are extended to handle multiple-input, single-output (MISO) systems. It is shown how knowledge of dependencies in the parameter variations can be exploited to reduce the number of computations in the resulting algorithm. A "square-root" implementation of the OVE algorithm is developed which has improved numerical stability properties. We show how the scope of OVE-ISP, an OVE-based input synthesis procedure, can be extended to handle systems with known transportation lag.

Lastly, it is shown how the OVE and OVETV algorithms can be utilized for fault detection and isolation (FDI). Two methods are suggested for detecting faults in dynamical systems. The first method relies on a consistency check which is integral to the OVE and OVETV algorithms, while the second method utilizes an ellipsoid intersection test to detect a fault. Two "recovery" strategies are also presented which allow the OVE and OVETV algorithms to continue to track the parameters after a fault is detected.

During the course of this research, we derived some extremely useful routines using symbolic computations for linear time-varying systems; these appear in an appendix to the dissertation.

© Copyright by

John Watkins

1995

To Beth and Sarah

Acknowledgements

I would like to acknowledge my adviser, Professor Stephen Yurkovich, for the assistance and guidance he has provided throughout my graduate career. I would like to thank Professor Kevin Passino and Professor Hitay Özbay for serving on my examination committee. The financial support of this research by the US Air Force and The Ohio State University is greatly appreciated.

The friendship and advise of fellow graduate students, both past and present, has been essential in my experience as a graduate student. In particular, I would like to thank Ken Cheung and Jim Gassman, whose work provided the inspiration and background for my own. I thank Stewart DeVilbiss for his help as our research traveled similar directions. I especially appreciate the assistance and friendship that Layne Lenning and Keith Redmill have provided as we shared this Ph.D. experience. I would also like to thank Julie Hurtig for all the help that she has provided me and my family as I finished my dissertation.

I would like to thank my family and friends who have made my graduate career and life worthwhile. I am indebted to my extended family who has provided me with support and encouragement throughout my life. I treasure my daughter, Sarah, who has brought more joy into my life than I could have ever imagined.

Finally, I sincerely thank my wife, Beth, for her understanding and encouragement. Although I didn't always make it clear, Beth, I could not have completed this work without you and your loving support. This is really our dissertation. Thank you.

VITA

September 17, 1965	Born—North Platte, Nebraska
June, 1987 – August, 1987	Research Assistant, Electrical Engineering Department, University of Nebraska-Lincoln, Lincoln, Nebraska
May, 1988 – August, 1988	Engineering Assistant, ICR Research Associates, Lincoln, Nebraska
June, 1989	B.S. University of Nebraska-Lincoln, Lincoln, Nebraska
September, 1989 – August, 1990	University Fellowship, The Ohio State University, Columbus, Ohio
September, 1990 – August, 1993	U.S. Air Force Laboratory Graduate Fellowship, The Ohio State University, Columbus, Ohio
August, 1991	M.S. The Ohio State University, Columbus, Ohio
October, 1993 – December, 1993	Graduate Teaching Associate, The Ohio State University, Columbus, Ohio
January, 1994 – December, 1994	University Fellowship, The Ohio State University, Columbus, Ohio
January, 1995 – August, 1995	Graduate Associate, The Ohio State University, Columbus, Ohio

Publications

"Parameter Set Estimation Algorithms for Time-Varying Systems," submitted for publication to *International Journal of Control*, 1994 (with S. Yurkovich).

"Nonlinear Control with Acceleration Feedback for a Two-Link Flexible Robot," *Control Engineering Practice*, vol. 1, no. 6, pp. 989-997, 1993 (with E. García–Benitez and S. Yurkovich).

"Fault Detection using Set-Membership Identification," submitted for publication in *Proceedings of the 13th World Congres of International Federation of Automatic Control*, 1996 (with S. Yurkovich).

"Ellipsoid Algorithms for Parameter Set Estimation of Time-Varying Systems," in *Proceedings of the American Control Conference*, Seattle, WA, pp. 2554-2558, June 1995 (with S. Yurkovich).

"An Optimal Volume Ellipsoid Algorithm for Parameter Set Estimation of Time Varying Systems," in *Proceedings of the 26th Southeastern Symposium on System Theory*, Athens, OH, pp. 110-114, March 1994 (with S. Yurkovich).

"Calculation of the State Transition Matrix for Linear Time Varying Systems," in *Mathematical Computation with Maple V: Ideas and Applications*, (T. Lee, ed.), pp. 157-166, Birkhäuser, 1993 (with S. Yurkovich).

"Input Shaping Controllers for Slewing Flexible Structures", in *Proceedings of the IEEE Conference on Control Applications*, Dayton, OH, September 1992 (with S. Yurkovich).

"Vibration Control for Slewing Flexible Structures", in *Proceedings of the American Control Conference*, Chicago, IL, pp. 2525-2529, June 1992 (with S. Yurkovich).

"Feedback Linearization with Acceleration Feedback for a Two-Link Flexible Manipulator," in *Proceedings of the American Control Conference*, Boston, MA, pp. 1360-1365, June 1991 (with E. García–Benitez and S. Yurkovich).

"Evaluation of Controllers for Slewing Flexible Structures," Master's Thesis, The Ohio State University, 1991.

Fields of Study

Major Field: Electrical Engineering

Studies in:

Control Systems
Robotics and Computer Engineering
Mathematics

Table of Contents

DEDICATION ii

ACKNOWLEDGEMENTS iii

VITA v

LIST OF TABLES xi

LIST OF FIGURES xii

CHAPTER PAGE

I Introduction 1

1.1 Motivation 1
1.2 Review of the Literature 5
1.3 Dissertation Organization 9

II OVE for Time-Varying Systems 13

2.1 Overview 13
2.2 Problem Statement 16
2.3 Measurement Update Equations 19
2.4 Time Update Equations 22
2.4.1 Two-Dimensional Case 25
2.4.2 r-Dimensional Case 32
2.5 OVETV Algorithm Summary 34
2.6 Alternative Strategies 34
2.6.1 Two-Dimensional Case 35
2.6.2 r-Dimensional Case 41
2.7 Summary 47

III Examples 48

3.1 Overview 48
3.2 First-order Example 48
3.3 Linear Time-Varying Circuit 52
3.4 Crude Oil Distillation Column 64
3.5 Summary 67

IV Algorithm Extensions 71

4.1 Overview 71
4.2 Multiple-Input, Single-Output Systems 73
4.2.1 Linear Time-Invariant Case 73
4.2.2 Linear Time-Varying Case 74
4.2.3 Crude Oil Distillation Column Example 78
4.3 Dependent Parameter Variations 79
4.4 Square-Root Implementation 83
4.4.1 Linear Time-Invariant Case 84
4.4.2 Linear Time-Varying Case 87
4.5 Input Synthesis 92
4.5.1 OVE-ISP Development for General Delay Case 95
4.5.2 Application of OVE-ISP and OVETV to Time-Varying Systems 99
4.6 Summary 106

V Fault Detection and Isolation using PSE 109

5.1 Overview 109
5.2 Issues in FDI 111
5.3 Detection of Failures 114
5.3.1 Consistency Check 115
5.3.2 Ellipsoid Intersection Test 118
5.4 Algorithm Recovery Strategies 126
5.4.1 Ellipsoid Resetting Algorithm 128
5.4.2 Ellipsoid Projection Algorithm 132
5.4.3 Integrated Approach 140
5.5 Example 143
5.5.1 Abrupt Fault 145
5.5.2 Incipient Fault 150

List of Tables

TABLE		PAGE
1	Measured data	4
2	Performance measures for OVETV-WRLS comparison	61
3	Distillation column data	66
4	Square-root algorithm and corresponding floating point operations	86
5	Direct implementation and corresponding floating point operations	87
6	Center estimate performance measures	153
7	Stability bounds on $\Phi(t, t_0)$	171
8	Maple V linear algebra procedures	173
9	Maple V utility procedures	174

5.6 Summary 155

VI Conclusion 158

6.1 Summary 158
6.2 Future Work 163

APPENDICES

A Optimum Parameters for Two-Dimensional Case 165

B Symbolic Computations for Linear Time-Varying Systems 167

B.1 Overview 167
B.2 Motivation 168
B.3 State Transition Matrix Properties 169
B.4 Calculating the State Transition Matrix 172
B.5 Examples 185
B.6 Summary 190

BIBLIOGRAPHY 193

List of Figures

FIGURE		PAGE
1	System identification scheme	2
2	Union of E_{k-1} with spheres on the surface of E_{k-1}	23
3	Minimum volume ellipse, $\hat{E}_k$, that bounds $\hat{G}_k$	32
4	Comparison of ellipse parameters between optimum and scalar multiplication	37
5	Comparison of ellipse parameters between optimum and scalar addition	40
6	Ratio of scalar addition volume versus optimum volume	41
7	OVETV update at $k = 16$ for First-Order example	50
8	OVETV results for First-Order example	50
9	Scalar addition results for First-Order example	51
10	Scalar multiplication results for First-Order example	51
11	Ellipsoid volume for First-Order example	53
12	Normalized center estimate error for First-Order example	53
13	Parameter center estimates for Linear Circuit example	62

14 Parameter b_1 of Linear Circuit example 62

15 Parameter b_2 of Linear Circuit example 63

16 Ellipsoid volume for Linear Circuit example 63

17 Distillation Column 65

18 Open loop data for Distillation Column example 66

19 Center estimates for Distillation Column example 68

20 Top section temperature difference, Distillation Column example . . . 68

21 Ellipsoid volume for Distillation Column example 69

22 Parameter bounds at 298 minutes for Distillation Column example . . 69

23 Parameters a_1 and b_{22} of MISO Distillation Column example 79

24 Parameter bounds at 298 minutes for MISO Distillation Column example 80

25 Ellipsoid volume for MISO Distillation Column example 80

26 OVE-ISP procedure viewed as time-varying feedback scheme 99

27 First time-varying example comparing system inputs 101

28 First time-varying example with OVE-ISP input sequence 102

29 Second time-varying example comparing system inputs 103

30 System outputs for second example 104

31 Frozen-time poles of first and second examples 104

32 Second time-varying example with OVE-ISP input sequence 105

33 Second time-varying example with Pronzato-Walter input sequence . 106

34 Fault detected at $k = 16$ 116

35 Fault detected at $k = 99$ 120

36 Intersecting and nonintersecting ellipsoids 121

37 Ellipsoid resetting algorithm results for First-Order example 129

38 Normalized center estimate error for First-Order example 129

39 Ellipsoid volume for First-Order example 131

40 Projection Algorithm 133

41 Ellipsoid projection algorithm results for First-Order example 140

42 Normalized center estimate error during abrupt fault 146

43 Parameter b_1 with Ellipsoid Projection algorithm 147

44 Change in the true parameter vector 148

45 Parameter b_1 with Ellipsoid Resetting algorithm 149

46 Ellipsoid volume during abrupt fault 150

47 Normalized center estimate error during incipient fault 152

CHAPTER I

Introduction

1.1 Motivation

Throughout the centuries, humans have desired to control their environment. Within the last 150 years, a large body of theory has been developed for designing algorithms which are capable of controlling elements of this environment autonomously. Most of these algorithms depend either explicitly of implicitly on a mathematical model of the system to be controlled. There are two fundamental methods for developing models which describe the dynamic behavior of a system.

The first method is an analytical approach whereby these mathematical equations are developed from the basic laws of physics, chemistry, biology, etc. One drawback with this approach is the fact the system may be too complex for the derivation of a model to be practical or even possible using this approach.

In contrast, the second method for deriving a model, system identification, is "relatively' easy to use. System identification is an empirical approach whereby experiments are performed on the system, and parameter values of a model are estimated based on the observed data. This method is not without its drawbacks. First of all, it

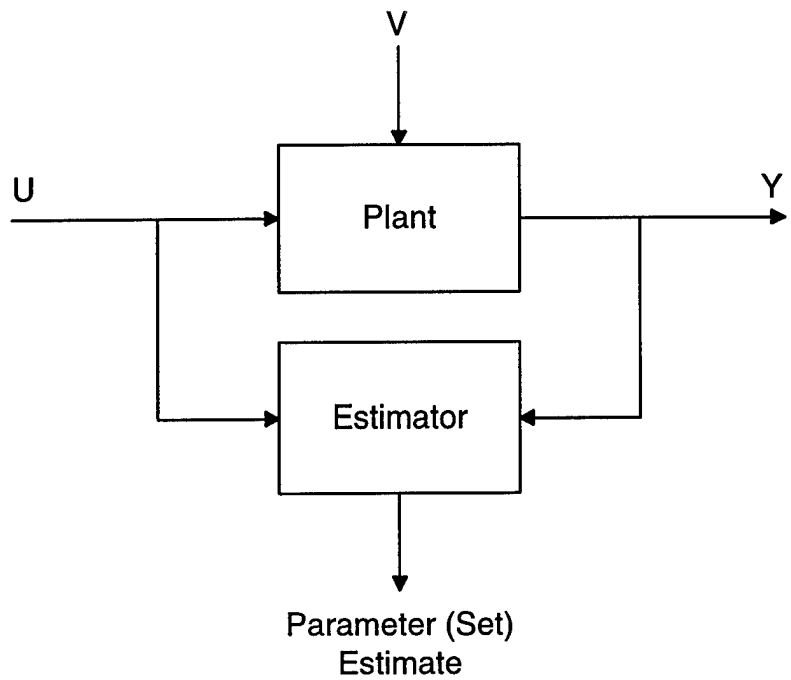

Figure 1: System identification scheme

may not be valid for all inputs and operating points. Secondly, in general, it does not provide as much physical insight into the problem as the analytical approach. Most complex systems usually require some combination of these approaches.

In spite of these drawbacks, system identification is a very powerful tool and the subject of this dissertation. A typical block diagram of a system identification scheme is shown in Figure 1.1, where the plant is the system to be modeled, u is the system input, and y is the system output. The estimator attempts to find parameters (or a set of parameters) which describe the behavior of the system. Unfortunately, this process is complicated by the fact that this must be done in the presence of noise and uncertainty, v.

Not surprisingly, the characterization of v dictates the nature of the algorithm used to estimate the parameters. Most traditional parameter estimation algorithms assume that v is a random variable which satisfies certain statistical properties. Given these assumptions, the algorithms seek to find a "best" estimate of the parameters that describe the system. A problem with this approach is the fact that this "best" estimate will probably not match the "true" parameters exactly. Consequently, there is no guarantee that a control algorithm designed for this "best" estimate will perform satisfactorily on the actual system.

Alternatively, it can be assumed that v is unknown-but-bounded. This results in a parameter set which is theoretically guaranteed to contain the true parameter. If the control algorithm is designed for the entire parameter set, some sense of robustness has been gained.

Conceptually, parameter set estimation is a simple idea. Consider the system

$$y_k = bu_k + v_k, \tag{1.1}$$

where

$$-1 \leq v_k \leq 1, \tag{1.2}$$

and where b is the parameter we seek to estimate. Solving (1.1) for v_k and substituting into (1.2) results in $-1 \leq y_k - bu_k \leq 1$ or

$$y_k - 1 \leq bu_k \leq y_k + 1. \tag{1.3}$$

An experiment was performed on the system from which the data in Table 1 was measured.

Table 1: Measured data

k	u_k	y_k
1	0.5	1.5
2	1.0	1.5

Substituting the measured values at $k = 1$ into (1.3), we find that the parameter b must lie within a certain feasible range, or set, of parameters; that is, b must satisfy $b \in [1, 5]$. Likewise, at $k = 2$, b must also satisfy $b \in [0.5, 2.5]$. The parameter b is constant (time-invariant), and therefore, must satisfy both these conditions. Therefore, the actual value of b must be contained in the "feasible parameter set" $[1, 2.5]$ where $[1, 2.5] = [1, 5] \cap [0.5, 2.5]$.

Unfortunately, as this concept is extended to higher dimensions, the feasible parameter set is quite complicated and difficult to track computationally. Because of their computational efficiency and ease in mathematical expression, ellipsoids are often used to overbound the feasible parameter set. However, if the estimated parameter set is to be used for robust control, there is a tradeoff between the volume of the ellipsoid and the achievable performance level. Consequently, we would like to find the smallest volume ellipsoid which is guaranteed to contain the true parameter values.

This was the motivation for the Optimal Volume Ellipsoid (OVE) algorithm which was developed in [1] for single-input, single-output linear time-invariant systems. This algorithm is computationally efficient and is guaranteed to contain the true plant.

While the OVE algorithm was developed for linear time-invariant systems, there is a large number of systems where the underlying model to be identified is linear time-varying. The identification of time-varying systems is an integral part of many adaptive control and adaptive signal processing applications. Consequently, the primary goal of this dissertation will be to extend the OVE parameter set estimation algorithm for identification of time-varying systems.

1.2 Review of the Literature

While the work in this dissertation was motivated in particular by [1], there has been a large amount of research in this area. Most parameter set estimation algorithms use either ellipsoids or boxes to bound the feasible parameter set; one exception is found in [2] where the exact feasible parameter set is calculated.

The most well known methods for bounding the feasible parameter set are the Optimal Bounding Ellipsoid (OBE) algorithms. Fogel and Huang [3] were the first to apply overbounding ellipsoids to the parameter set estimation (PSE) problem. In their algorithm, the center of the bounding ellipsoid can be viewed as a particular weighted least squares estimate. They considered two optimization strategies. One strategy sought to minimize the volume of the ellipsoid while the second strategy sought to minimize the lengths of the ellipsoid semi-axes.

A third optimization strategy with no interpretable geometrical significance was used by Dasgupta and Huang in [4]. In [5], it was shown how these optimization strategies can be combined with different weighting strategies to include a larger

family of algorithms that they call unified OBE (UOBE). Included in this family is the set-membership weighted recursive least squares (SM-WRLS) algorithm.

These algorithms are recursive in nature. They also have the capability of ignoring redundant data. However, even when they use the minimum volume optimization strategy, they do not produce the ellipsoid with the minimum volume. This conservatism is undesirable for robust control design. Although the volume convergence of the Fogel and Huang algorithm was improved by Belforte and Bona [6], it still did not produce an ellipsoid with the minimum volume because the center estimate was constrained to be a weighted least square estimate.

In [1], Cheung, Yurkovich, and Passino eliminated this constraint which led to the Optimal Volume Ellipsoid (OVE) algorithm, shown to have desirable convergence properties (proof) and geometrical interpretation. Two independently developed algorithms, which have been shown to be mathematically equivalent to the OVE algorithm, are the the modified OBE (MOBE) algorithm of [7] and the EPC algorithm of [8]. The difference between the algorithms lies in convergence proofs, geometrical interpretation, and implementation.

Several extensions to the OVE algorithm have already been considered. In [9], a modified-OVE (MOVE) algorithm was presented which allows for time-varying bounds on v and simplifies the parameter update calculations. The OVE algorithm was extended for interconnected systems in [10]. In [11], an OVE-based input synthesis procedure was presented which seeks to choose the system input such that a rapid decrease in ellipsoid volume occurs.

We are particularly interested in how PSE can be extended to time-varying systems. As in the time-invariant case, most adaptive PSE algorithms have used overbounding ellipsoids. However, two algorithms using polytopes can be found in [12]. In [13], three ellipsoid-type algorithms for tracking time varying systems are discussed: scalar bound inflation, fixed-memory bounding, and bound incrementing. Most adaptive PSE algorithms fit into one of these three categories.

The simplest of these is scalar bound inflation. In this algorithm, the ellipsoid is expanded uniformly in all direction before processing the new measurement. While computationally simple, one difficulty with this approach has been choosing the expansion factor such that the set estimate is guaranteed to contain the true plant. Similar to the forgetting factor in the standard weighted recursive least squares algorithm, the expansion factor is often chosen heuristically, usually to be a constant [5].

In [14], Rao and Huang show how the Dasgupta-Huang OBE algorithm can be modified by artificially increasing the noise bound to guarantee consistency if an update takes place. However if no update takes place, this consistency can still be lost. They also have developed a procedure to "recover" when tracking is lost, for example due to an unexpected large parameter jump.

With fixed memory bounding the data is windowed, that is, only the last L observations are used to find the plant estimate. Norton and Mo [13] found this method awkward to implement with no advantage over scalar bound inflation or bound incrementing. However, in [5] "back-rotation", an efficient method for implementing the

window, is applied to SM-WRLS. With this strategy two additional windowing techniques were also considered. In the first, instead of a rectangular window, tapered windowing is used where the window tapers slowly to zero. In the second, selective forgetting is used where only previously heavily weighted data sets are removed. These algorithms offer no guarantee of consistency.

The last strategy that [13] considered was the bound incrementing method. This method is based on an explicit parameter variation model in which the possible parameter deviation during each time step is assumed to be ellipsoidally bounded. They used a result given in [15] which finds the minimum volume ellipsoid that bounds the sum of two vectors which are also ellipsoidally bounded.

A significant motivation for extending PSE to time-varying systems is adaptive robust control. As defined in [16], the robust control problem is the problem of analyzing and designing accurate control systems given plants which contain significant uncertainty. While this is still an open research area, there has been some work in this direction.

In [17], Kosut examined some of the issues involved in using PSE for adaptive control. A finite horizon controller was designed for a finite impulse response (FIR) model whose parameters were ellipsoidally bounded in [18]. In [19], a controller was designed for systems whose output matrix was ellipsoidally bounded. Bound-based worst-cased self-tuning controllers were designed in [20].

Some very promising results for robust control can be found in [21]. In this work, techniques are developed for extracting modeling uncertainty information, vital for

H^∞ robust control design, from the results of PSE algorithms. A worst-case analysis was used to address the characterization of parametric uncertainty as additive and coprime factor perturbations to a nominal plant model. Techniques for identification of reduced-order perturbation weightings were also given.

1.3 Dissertation Organization

This dissertation is organized into four main chapters. In Chapter II, the OVE algorithm is extended for identification of time-varying systems. After illustrating how the equations arise for an optimal time update when there are two parameters to estimate, the results of [15] are combined with the optimum volume measurement update equations of [1] to develop an Optimal Volume Ellipsoid algorithm for Time-Varying systems (OVETV).

Building on the development of the OVETV, two algorithms are presented for reducing the computational complexity of the optimal time update equations. These algorithms reduce the computational complexity of the time update equations by constraining the new ellipsoid to be parameterized by the previous ellipsoid and a single new parameter. One of these algorithms uses the same parameterization as the scalar bound inflation method of [13]. However, the algorithms developed in this chapter are optimized for volume and combined with the optimal volume measurement update equations of [1].

Three examples, which illustrate the main feature of the OVETV algorithm, are considered in Chapter III. The first is a simple first-order example which compares

the OVETV algorithm with the scalar addition and scalar multiplication approaches. In the second example, identification of a linear time-varying circuit is used to demonstrate how this approach can be applied to sampled-data systems with quantization noise. In the final example, the OVETV algorithm is applied to actual data obtained from a crude oil distillation column.

In Chapter IV, four extensions to the OVETV algorithm are presented. In many real systems, the output of the system is effected by more than one input. We discuss how the OVE and OVETV algorithms can be used to identify multiple-input, single-output (MISO) systems. Results are demonstrated on the distillation column example of Chapter III.

Often, the time-varying parameters that we are trying to identify are dependent on a smaller set of physical time-varying parameters. If we can exploit this dependence, we may be able to simplify the problem and reduce the number of computations required to implement the algorithm. Adaptations of the OVETV algorithm are developed to utilize knowledge of these dependencies.

When applying the OVE and OVETV algorithms in practice, numerical round-off errors can cause the algorithms to become numerically intractable. To solve this problem, we apply an approach that has been successfully used in stochastic estimation schemes. A "square-root" algorithm is developed for implementing OVE. Furthermore, it is shown how the square-root approach can be extended to time-varying systems.

The type of system input for identification experiments has a very significant

effect on the "quality" of estimates produced from an identification scheme. The final extension to the OVETV algorithm which we will consider in Chapter IV is based on the OVE-ISP procedure of [11]. We show how the scope of OVE-ISP can be extended to handle systems with known transportation lag such as the distillation column in Chapter III. We also demonstrate its application to time-varying systems and an important "stabilizing" property that other synthesis strategies do not appear to possess.

Fault detection and isolation (FDI) is concerned with the detection and identification of failures in complex dynamical systems. In Chapter V, we discuss how the OVE and OVETV algorithms can be utilized for FDI. Two methods are suggested for detecting faults in dynamical systems. The first method relies on a consistency check which is integral to the OVE and OVETV algorithms, while the second method utilizes an ellipsoid intersection test to detect a fault.

Two "recovery" strategies are presented which allow the OVE and OVETV algorithms to continue to track the parameters after a fault is detected. The first strategy simply resets the current ellipsoid to an ellipsoid which is "large enough" to guarantee that the "true" parameter is contained in the parameter set immediately after a fault is detected. An alternative strategy based on projections is also introduced. While this algorithm is not guaranteed to recapture the ellipsoid, it usually produces ellipsoids with smaller volumes.

An integrated approach is also suggested for combining these recovery strategies. Finally, the algorithms of Chapter V are illustrated using the linear time-varying

circuit example of Chapter III.

Finally, during the course of this research, investigation into time-varying linear systems was imperative. In so doing, we derived some extremely useful routines using symbolic computations for linear time-varying (LTV) systems. To our knowledge, such routines have not appeared elsewhere in the open literature; we have gathered this material in a stand-alone unit, given in Appendix B.

CHAPTER II

OVE for Time-Varying Systems

2.1 Overview

Identification of time-varying systems is essential for many adaptive control and adaptive signal processing applications. In particular, adaptive tracking algorithms are needed for (i) complex systems which admit a linear, time-varying parameterization, (ii) failure detection and identification (FDI) systems, and (iii) gain scheduling control systems where supervisory controllers must track changing system characteristics. While most adaptive tracking algorithms assume that the perturbations to the system satisfy certain statistical properties, in many real world applications it is often more natural to assume that these perturbations are unknown-but-bounded.

Traditional system identification techniques abound for such applications, where a point in the parameter set is identified to model the system. A different philosophy, discussed in [22], [17], and elsewhere, is to seek to identify a *set* of models which are consistent with the incoming data (referred as *set membership identification* or *parameter set estimation*, or PSE). A major motivation for adopting this philosophy for the identification portion of the overall control design process is for use in *robust*

control design. While many adaptive and auto-tuning techniques have been developed based on point estimation schemes over the last 20 years, it has only been recently that connections have been made to the robust control design problem for ellipsoidally based PSE techniques; see, for example, [21] and [19]. It is for this reason, within this promising line of research for robust identification and control, that we focus here on PSE algorithms, and limit our discussions and comparisons with point estimation schemes. Indeed, there exists a large body of literature, which we cannot address here, on point estimation schemes, many of which are useful in practice (see, e.g., [23]).

Generally speaking, PSE algorithms are classified as robust identification techniques, because they attempt to characterize uncertainty by identifying the set of feasible parameters, given the data from the identification experiment. That is, PSE seeks to identify a set of parameters which are feasible with the measured data and the bounds on the perturbations. In general, this feasible set is an irregular convex set. Because of their computational efficiency and ease in mathematical expression, ellipsoids are used to overbound the feasible set. When PSE is used in robust control or robust adaptive control, there is a tradeoff between set size and system performance. Consequently, one seeks the smallest volume ellipsoid which is guaranteed to contain the "true" plant parameters.

In [13], the authors considered the bound incrementing and scalar bound inflation methods for PSE of time-varying systems. In [1], the Optimal Volume Ellipsoid (OVE) algorithm was developed for a time-invariant single-input, single-output

(SISO) auto-regressive with exogenous input (ARX) model, and extended in [10] for interconnected systems. The bounded incrementing method of [13] uses an optimum volume algorithm found in [15] for updating the time equations.

In this chapter, after illustrating how the equations arise for the optimal time update equations when there are two parameters to estimate, we will combine the results of [15] with the optimum volume measurement update equations of [1] to develop an Optimal Volume Ellipsoid algorithm for Time-Varying systems (OVETV). The OVETV enjoys many of the favorable characteristics of the OVE, which in fact are shared with other mathematically equivalent techniques such as extensions of the modified OBE algorithm of [7], and the ellipsoid with parallel cuts (EPC) algorithm of [8].

Building on the development of the OVETV, we present two algorithms for reducing the computational complexity of the optimal time update equations, thereby making real-time implementation feasible for many applications. These algorithms reduce the computational complexity of the time update equations by constraining the new ellipsoid to be parameterized by the previous ellipsoid and a single new parameter. One of these algorithms uses the same parameterization as the scalar bound inflation method of [13]. However, the algorithms developed in this chapter are optimized for volume and combined with the optimal volume measurement update equations of [1].

2.2 Problem Statement

Consider the time-varying SISO ARX model

$$y_k = \theta_k^T \phi_k + v_k, \tag{2.1}$$

$$\theta_{k+1} = \theta_k + w_k. \tag{2.2}$$

where $\theta_k = \begin{bmatrix} a_{1k} & \cdots & a_{nk} & b_{lk} & \cdots & b_{mk} \end{bmatrix}^T$ is the parameter vector, y_k is the system output, u_k is the system input, and $\phi_k = \begin{bmatrix} -y_{k-1} & \cdots & -y_{k-n} & u_{k-l} & \cdots & u_{k-m} \end{bmatrix}^T$ is the regression vector. Here l is the number of time steps delay in the system where $0 \leq l \leq m$. Let $r = n + m - l + 1$. The system disturbance, v_k, and the parameter disturbance vector, $w_k \in \Re^r$, are assumed to satisfy the following bounds

$$\underline{\gamma}_k \leq v_k \leq \overline{\gamma}_k, \tag{2.3}$$

$$w_k^T R_k^{-1} w_k \leq 1, \tag{2.4}$$

where $\underline{\gamma}_k < \overline{\gamma}_k$ and $\underline{\gamma}_k$, $\overline{\gamma}_k$, and $R_k \in \Re^{r \times r}$ are known at each time k. The vector, w_k, represents the change in the parameter vector at each time k. The matrix, R_k, is symmetric, positive definite and represents an ellipsoidal bound on the possible parameter vector deviation during each time step.

Equations (2.1) and (2.2) and the bounds in (2.3) and (2.4) are used to define the adaptive parameter set estimation problem. Let $H_{k-1} \subset \Re^r$ be the set such that all $\theta_{k-1} \in H_{k-1}$ are feasible parameters of the plant which are consistent with past y_k, ϕ_k, $\underline{\gamma}_k$, $\overline{\gamma}_k$, and R_k. [1] Let $F_k \subset \Re^r$ be the set such that all $\theta_k \in F_k$ are feasible

[1] In this work, the notation $A \subset B$ means that A is a subset of B, and therefore, A may be equal to B.

parameter estimates of the plant which are consistent with the measurements at time k. That is, from (2.1) and (2.3),

$$F_k = \{\theta_k \in \Re^r : \underline{\gamma}_k \leq y_k - \theta_k^T \phi_k \leq \overline{\gamma}_k\}, \tag{2.5}$$

where F_k is the region between two parallel hyperplanes.

Let $G_k \subset \Re^r$ be the set such that all $\theta_k \in G_k$ are feasible parameter estimates of the plant which are consistent with $\theta_{k-1} \in H_{k-1}$. That is, from (2.2) and (2.4),

$$G_k = \{\theta_k \in \Re^r : (\theta_k - \theta_{k-1})^T R_{k-1}^{-1} (\theta_k - \theta_{k-1}) \leq 1; \theta_{k-1} \in H_{k-1}\}. \tag{2.6}$$

Before incorporating a new measurement, the set, G_k, tells us how much the parameter set estimate at time $(k-1)$, H_{k-1}, must expand to guarantee consistency at time k. For H_k to be consistent, it must satisfy (2.5) and (2.6); therefore, H_k is given by:

$$H_k = G_k \cap F_k. \tag{2.7}$$

The problem of parameter set estimation for time-varying systems is to find H_k explicitly in the parameter space where H_k is defined recursively by (2.5), (2.6), and (2.7). In general, however, computing (2.7) is an extremely difficult problem because the feasible parameter space, H_k, is an irregular convex set. Consequently, ellipsoids will be used to overbound H_k because of their ease in mathematical expression and computational efficiency.

Define the ellipsoid E_{k-1} as

$$E_{k-1} = \{\theta_{k-1} : (\theta_{k-1} - \tilde{\theta}_{k-1})^T \tilde{P}_{k-1}^{-1} (\theta_{k-1} - \tilde{\theta}_{k-1}) \leq 1; \theta_{k-1} \in \Re^r\} \tag{2.8}$$

where $\tilde{\theta}_{k-1} \in \Re^r$ is the center of the ellipsoid, and $\tilde{P}_{k-1} \in \Re^{r\times r}$ is a symmetric, positive definite matrix. Similarly, define $\hat{E}_k$ as

$$\hat{E}_k = \{\theta_k : (\theta_k - \hat{\theta}_k)^T \hat{P}_k^{-1}(\theta_k - \hat{\theta}_k) \leq 1; \theta_k \in \Re^r\} \tag{2.9}$$

where $\hat{\theta}_k \in \Re^r$ is the center of the ellipsoid and $\hat{P}_k \in \Re^{r\times r}$ is a symmetric, positive definite matrix.

Choose E_{k-1} such that $E_{k-1} \supset H_{k-1}$. By replacing H_{k-1} with E_{k-1} in (2.6), we obtain a new set, $\hat{G}_k$, given by

$$\hat{G}_k = \{\theta_k \in \Re^r : (\theta_k - \theta_{k-1})^T R_{k-1}^{-1}(\theta_k - \theta_{k-1}) \leq 1; \theta_{k-1} \in E_{k-1}\}. \tag{2.10}$$

In general, $\hat{G}_k$ is not an ellipsoid. Consequently, we choose $\hat{E}_k$ such that $\hat{E}_k \supset \hat{G}_k$, i.e., we wish to bound the set, $\hat{G}_k$ with an overbounding ellipsoid. In particular, we would like to choose the $\hat{E}_k$ with minimum volume, that is

$$\hat{E}_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \hat{G}_k\}. \tag{2.11}$$

In addition to $\hat{G}_k$, at time k the ellipsoid, E_k, is also constrained by F_k as defined in (2.5). Clearly, we wish to choose E_k such that $E_k \supset \hat{E}_k \cap F_k$, i.e., we wish to bound all consistent parameter estimates with an overbounding ellipsoid. As with $\hat{E}_k$, we seek the E_k with minimum volume, that is

$$E_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \hat{E}_k \cap F_k\}. \tag{2.12}$$

Equations (2.5), (2.10), (2.11), and (2.12) provide a recursive algorithm for finding an ellipsoid E_k such that $E_k \supset H_k$. However, this algorithm assumes that the solutions

to (2.11) and (2.12) can be calculated. The solution to (2.12), the measurement update equations, will be solved using a modified version of the OVE algorithm found in [9]. The solution to (2.11), the time update equations, will be developed for $r = 2$, i.e., when there are two parameters to estimate. A result by Chernous'ko will be used to generalize this to the full r-dimensional case [15].

2.3 Measurement Update Equations

In [1], Cheung, Yurkovich, and Passino developed the OVE algorithm, with convergence proofs, for linear time invariant systems, that is, when $w_k = 0$. This result was extended for interconnected systems in [10]. The OVE algorithm solves the following optimization problem:

$$E_k = \arg_E \min\{\mathrm{vol}(E) : E \supset E_{k-1} \cap F_k\}. \tag{2.13}$$

The most significant difference between the OVE algorithm and the seminal work of [3] is that unlike the OVE algorithm, the OBE algorithm constrains the center of the new ellipsoid, E_k, to satisfy a "modified" recursive least squares estimate. It is, in fact, true that the OVE, the modified OBE (MOBE) algorithm of [7], and the EPC algorithm of [8] are mathematically equivalent. The difference between the algorithms lies in convergence proofs, geometrical interpretation, and implementation.

The OVE algorithm in [1] assumes that $\overline{\gamma}_k = -\underline{\gamma}_k = \gamma$. In [9], Gassman and Yurkovich discuss the modified OVE (MOVE) algorithm which uses the more general bound in (2.3). The optimization in (2.13) is identical to (2.12) with $\hat{E}_k$ replaced by

E_{k-1}. Consequently, we will be able to use the MOVE algorithm by simply replacing $\tilde{\theta}_{k-1}$ and $\tilde{P}_{k-1}$ with $\hat{\theta}_k$ and $\hat{P}_k$, respectively.

With these substitutions, the algorithm is given as follows:

1. Set

$$\underline{\alpha}_k = \max\left(\frac{y_k - \hat{\theta}_k^T \phi_k - \overline{\gamma}_k}{(\phi_k^T \hat{P}_k \phi_k)^{1/2}}, -1\right) \tag{2.14}$$

$$\overline{\alpha}_k = \min\left(\frac{y_k - \hat{\theta}_k^T \phi_k - \underline{\gamma}_k}{(\phi_k^T \hat{P}_k \phi_k)^{1/2}}, 1\right) \tag{2.15}$$

If $\underline{\alpha}_k \geq 1$ or $\overline{\alpha}_k \leq -1$, then the observed data is inconsistent with $\hat{E}_k$ and the algorithm stops.

2. Set $\epsilon_k = \underline{\alpha}_k \overline{\alpha}_k$. If $\epsilon_k \leq -\frac{1}{r}$, then no measurement update is necessary, that is

$$\tilde{\theta}_k = \hat{\theta}_k \qquad \tilde{P}_k = \hat{P}_k. \tag{2.16}$$

3. Set $\mu_k = \frac{\underline{\alpha}_k + \overline{\alpha}_k}{2}$.

4. If $|\mu_k| > \rho$, then

$$b_k = 2r\mu_k + \frac{1+\epsilon_k}{\mu_k} \tag{2.17}$$

$$\tau_k = \frac{b_k - \mathrm{sign}(\mu_k)\sqrt{b_k^2 - 4(r+1)(1+r\epsilon_k)}}{2(r+1)} \tag{2.18}$$

$$\sigma_k = \tau_k\left(\tau_k - \frac{1+\epsilon_k}{\mu_k}\right) + 1 \tag{2.19}$$

$$\delta_k = \frac{\sigma_k}{1 - \frac{\tau_k}{\mu_k}} \tag{2.20}$$

5. If $|\mu_k| \leq \rho$, then $\tau_k = 0$ and

$$\alpha = \max(|\underline{\alpha}_k|, |\overline{\alpha}_k|) \tag{2.21}$$

$$\sigma_k = r\alpha^2 \tag{2.22}$$

$$\delta_k = \frac{r(1-\alpha^2)}{r-1} \tag{2.23}$$

6. Update $\tilde{\theta}_k$ and $\tilde{P}_k$

$$\tilde{\theta}_k = \hat{\theta}_k + \frac{\tau_k \hat{P}_k \phi_k}{(\phi_k^T \hat{P}_k \phi_k)^{1/2}} \tag{2.24}$$

$$\tilde{P}_k = \delta_k \hat{P}_k + (\sigma_k - \delta_k) \frac{\hat{P}_k \phi_k \phi_k^T \hat{P}_k}{\phi_k^T \hat{P}_k \phi_k} \tag{2.25}$$

To arrive at the equations, an affine transform can be used to transform the ellipsoid $\hat{E}_k$ to a unit ball centered at the origin. In the transformed coordinate system, $\underline{\alpha}_k$ and $\overline{\alpha}_k$ are the coordinates along ϕ_k of the two hyperplanes (defined by F_k) which are orthogonal to ϕ_k. The consistency check in step 1 is used to verify that the intersection of $\hat{E}_k$ and F_k is not empty. If this intersection is empty, then either E_0 did not contain the "true" parameter or the assumptions made in (2.1) through (2.4) were invalid. Recovery strategies have been developed to handle the situation where an inconsistency does occur.

The check in step 2 is used to determine whether the current data record contains enough information to reduce the ellipsoid volume. In steps 4 and 5, ρ is chosen to be a very small value and is used to determine when $\underline{\alpha}_k \approx -\overline{\alpha}_k$. In the transformed coordinate system, τ_k, ϕ_k, and δ_k are used to characterize the optimum volume ellipsoid E_k. The parameter τ_k is the center coordinate of E_k along ϕ_k, σ_k is the squared

length of the semi-axis of E_k along ϕ_k, and δ_k is the squared length of the semi-axes of E_k orthogonal to ϕ_k. In step 6, the center $\tilde{\theta}_k$ and orientation matrix $\tilde{P}_k$ of the optimum volume ellipsoid E_k are given in the original coordinate system. Finally, it should be noted that a similar algorithm for set membership *state estimation* can be found in [24].

2.4 Time Update Equations

Initially, we will constrain R_k to be $\zeta_k^2 I$ where I is the identity matrix, and ζ_k bounds $\|w_k\|_2$. Doing so, equation (2.10) can be rewritten as

$$\hat{G}_k = \{\theta_k \in \Re^r : (\theta_k - \theta_{k-1})^T(\theta_k - \theta_{k-1}) \leq \zeta_{k-1}^2; \theta_{k-1} \in E_{k-1}\}. \tag{2.26}$$

Later, this constraint will be relaxed.

Clearly for $\zeta_{k-1} = 0$ where $k = 1 \ldots N$, $\hat{E}_k = \hat{G}_k = E_{k-1}$, that is, this algorithm reduces to the MOVE algorithm. Note that, while E_0 cannot be chosen to be $\Re^r$, it can be chosen arbitrarily large so that the "true" parameter vector, θ_0, is guaranteed to be in E_0.

To solve (2.11), it is important that we understand the nature of $\hat{G}_k$. From (2.26), we see that $\hat{G}_k$ is a union of hyperspheres with radius ζ_k whose centers are contained in the ellipsoid, E_{k-1}. The hyperspheres of greatest interest are those on the hypersurface of the ellipsoid because they define the hypersurface of $\hat{G}_k$. See Figure 2 for an example of the 2-D case.

Let $\vec{\nu}$ be the unit vector normal to the hypersurface of E_k that points away from

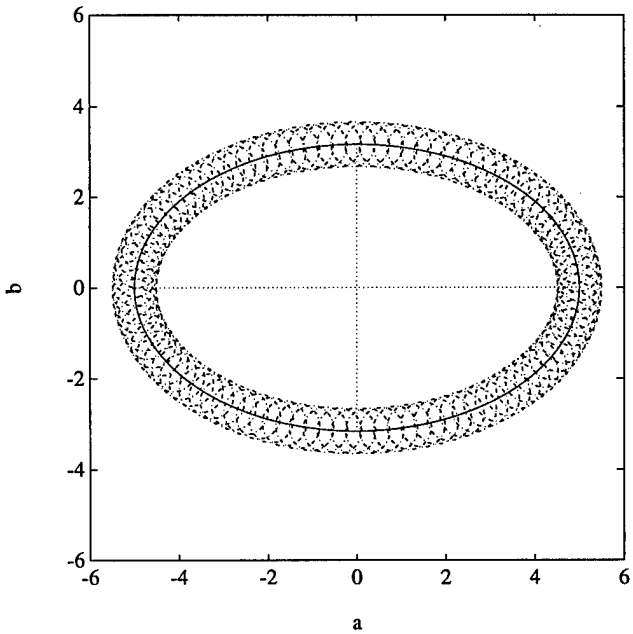

Figure 2: Union of E_{k-1} with spheres on the surface of E_{k-1}

the interior of E_k. Clearly, the distance along $\vec{\nu}$ between the hypersurface of E_k and the hypersurface of $\hat{G}_k$ is always ζ_{k-1}. In fact, this defines the hypersurface of $\hat{G}_k$. The surfaces of E_{k-1} and $\hat{E}_k$ are defined by

$$S_{k-1} = \{\theta_{k-1} : (\theta_{k-1} - \tilde{\theta}_{k-1})^T \tilde{P}_{k-1}^{-1} (\theta_{k-1} - \tilde{\theta}_{k-1}) = 1; \theta_{k-1} \in \Re^r\}, \tag{2.27}$$

$$\hat{S}_k = \{\theta_k : (\theta_k - \hat{\theta}_k)^T \hat{P}_k^{-1} (\theta_k - \hat{\theta}_k) = 1; \theta_k \in \Re^r\}. \tag{2.28}$$

Let the distance between two parameter vectors, θ_i and θ_j, be given by

$$d(\theta_i, \theta_j) = \| \theta_i - \theta_j \|_2 . \tag{2.29}$$

Next we need the following lemma.

Lemma 1 *The minimum distance between the hypersurface of $\hat{E}_k$ and the hypersurface of E_{k-1} must be ζ_{k-1}, that is,*

$$\zeta_{k-1} = \min\{d(\theta_{k-1}, \theta_k) : \theta_{k-1} \in S_{k-1}, \theta_k \in \hat{S}_k\} \tag{2.30}$$

Proof: If the minimum distance is less than ζ_{k-1}, then $\hat{E}_k \not\supset \hat{G}_k$. This is a contradiction because $\hat{E}_k$ must bound $\hat{G}_k$. If the minimum distance is greater than ζ_{k-1}, then there must exist some ellipsoid $E \supset \hat{G}_k$ where $\mathrm{vol}(E) < \mathrm{vol}(\hat{E}_k)$. This is a contradiction because $\hat{E}_k$ must be the minimum volume ellipse that contains $\hat{G}_k$. Therefore, (2.30) must hold. □

Using Lemma 1, equation (2.11) can be rewritten as

$$\hat{E}_k = \arg_E \min\{\mathrm{vol}(E) : \min(d(\theta_{k-1}, \theta_k)) = \zeta_{k-1}; \theta_{k-1} \in S_{k-1}, \theta_k \in \hat{S}_k\}. \tag{2.31}$$

That is, we want to find the minimum volume ellipsoid, $\hat{E}_k$, such that the minimum distance between the surface of E_{k-1} and $\hat{E}_k$ is equal to ζ_{k-1}. This suggests a two step optimization problem. First, solve (2.30) analytically for minimum values θ^*_{k-1} and θ^*_k. That is, find θ^*_{k-1} and θ^*_k as a function of E_{k-1} and $\hat{E}_k$ such that $d(\theta^*_{k-1}, \theta^*_k) = \zeta_{k-1}$. Secondly, substitute θ^*_{k-1} and θ^*_k into (2.31) and solve for the minimum volume $\hat{E}_k$ as a function of E_{k-1}.

Using the singular value decomposition (SVD), the positive definite matrix, $\tilde{P}_{k-1}$, used to define the ellipsoid in (2.8) can be factored as

$$\tilde{P}_{k-1} = V_{k-1}\Lambda_{k-1}V^T_{k-1} \tag{2.32}$$

where $V_{k-1} \in \Re^{r\times r}$ is orthogonal and $\Lambda_{k-1} \in \Re^{r\times r}$ is diagonal. The axes of the ellipsoid are aligned with the columns of V_{k-1}. The semi-axes lengths are given by the diagonal elements of $\Lambda_{k-1}^{1/2}$ [25]. Note also that $\mathrm{vol}(E_{k-1}) \propto \sqrt{\det(\tilde{P}_{k-1})} = \sqrt{\det(\Lambda_{k-1})}$. Due to the symmetry of E_{k-1} and $\hat{G}_k$, the minimum volume $\hat{E}_k$ must have the same center and axes as E_k, that is,

$$\hat{\theta}_k = \tilde{\theta}_{k-1}, \tag{2.33}$$

$$\hat{P}_k = V_{k-1}\hat{\Lambda}_k V_{k-1}^T. \tag{2.34}$$

Consequently, the optimization problem becomes one of finding $\hat{\Lambda}_k$ as a function of Λ_{k-1} such that (2.30) is satisfied and $\sqrt{\det(\hat{\Lambda}_k)}$ is minimized.

With this said, the general r dimensional problem will be discussed later. The two dimensional problem, $r = 2$, is solved next.

2.4.1 Two-Dimensional Case

Although most systems that we would like to identify have more than two parameters to estimate, the two-dimensional problem is important because it is easy to visualize and it provides insight into the r-dimensional problem. In the two-dimensional case, "ellipsoids" become "ellipses" and "volume" is actually "area".

As discussed in the previous section, we will begin by solving (2.30) for θ_{k-1}^* and θ_k^*. By (2.33) and (2.34), it is clear that we can consider ellipses centered at the origin with axes aligned along the coordinate axes without any loss of generality. Define the

surface of ellipses, E_1 and E_2, as

$$S_1 = \{x_1, y_1 : \frac{x_1^2}{a_1^2} + \frac{y_1^2}{b_1^2} = 1; x_1, y_1 \in \Re\}, \tag{2.35}$$

$$S_2 = \{x_2, y_2 : \frac{x_2^2}{a_2^2} + \frac{y_2^2}{b_2^2} = 1; x_2, y_2 \in \Re\}, \tag{2.36}$$

where a_1, b_1, a_2, and b_2 are positive constants. The distance between S_1 and S_2, (2.29), can be rewritten as

$$d(x_1, x_2, y_1, y_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{2.37}$$

Now we can prove the following lemma. Note, that because of symmetry, we can find the solution in the upper right quadrant without any loss of generality.

Lemma 2 *The parameters which minimize (2.37), the distance between S_1 and S_2, must satisfy one of four solutions:*

- *Minimum distance along y-axis:*

$$x_1^* = 0 \qquad y_1^* = b_1, \tag{2.38}$$

$$x_2^* = 0 \qquad y_1^* = b_2, \tag{2.39}$$

with

$$d(x_1^*, x_2^*, y_1^*, y_2^*) = |b_2 - b_1|, \tag{2.40}$$

- *Minimum distance along x-axis:*

$$x_1^* = a_1 \qquad y_1^* = 0, \tag{2.41}$$

$$x_2^* = a_2 \qquad y_1^* = 0, \tag{2.42}$$

with

$$d(x_1^*, x_2^*, y_1^*, y_2^*) = |a_2 - a_1|, \tag{2.43}$$

- *Ellipses Intersect:*

$$x_1^* = p_4 \qquad y_1^* = p_5, \tag{2.44}$$

$$x_2^* = p_4 \qquad y_1^* = p_5, \tag{2.45}$$

with

$$d(x_1^*, x_2^*, y_1^*, y_2^*) = 0, \tag{2.46}$$

- *Minimum distance between axes:*

$$x_1^* = \frac{a_1^2\sqrt{p_3}}{(a_1^2 - b_1^2)\sqrt{p_1}} \qquad y_1^* = \frac{b_1^2\sqrt{p_2}}{(a_1^2 - b_1^2)\sqrt{p_1}}, \tag{2.47}$$

$$x_2^* = \frac{a_2^2\sqrt{p_3}}{(a_2^2 - b_2^2)\sqrt{p_1}} \qquad y_1^* = \frac{b_2^2\sqrt{p_2}}{(a_2^2 - b_2^2)\sqrt{p_1}}, \tag{2.48}$$

with

$$d(x_1^*, x_2^*, y_1^*, y_2^*) = \sqrt{\frac{(b_1^2a_2^2 - a_1^2b_2^2)(b_2^2 - a_2^2 - b_1^2 + a_1^2)}{(b_2^2 - a_2^2)(b_1^2 - a_1^2)}}, \tag{2.49}$$

where

$$p_1 = a_1^2b_2^2 - b_1^2a_2^2, \tag{2.50}$$

$$p_2 = b_2^4a_1^2 - 2b_2^2a_2^2a_1^2 + a_2^4a_1^2 - b1^4a_2^2 + 2b_1^2a_2^2a_1^2 - a1^4a_2^2, \tag{2.51}$$

$$p_3 = b_1^4b_2^2 - b_1^2b_2^4 - 2b_1^2b_2^2a_1^2 + b_2a_1^4 + 2b_2^2b_1^2a_2^2 - a_2^4b_1^2, \tag{2.52}$$

$$p_4 = \frac{a_1a_2\sqrt{b_2^2 - b_1^2}}{\sqrt{p_1}}, \tag{2.53}$$

$$p_5 = \frac{b_1b_2\sqrt{a_1^2 - a_2^2}}{\sqrt{p_1}}. \tag{2.54}$$

Proof: Because $d(x_1, x_2, y_1, y_2)$ is nonnegative, minimizing d^2 is equivalent to minimizing d. Using the Lagrange multiplier method, the constraints posed by the ellipses, (2.35) and (2.36), can be augmented to d^2 to form

$$\begin{aligned} f(x_1, x_2, y_1, y_2, \lambda_1, \lambda_2) &= (x_1 - x_2)^2 + (y_1 - y_2)^2 + \lambda_1(x_1^2 b_1^2 \\ &\quad + y_1^2 a_1^2 - a_1^2 b_1^2) + \lambda_2(x_2^2 b_2^2 + y_2^2 a_2^2 - a_2^2 b_2^2). \end{aligned} \tag{2.55}$$

For $f(x_1^*, x_2^*, y_1^*, y_2^*, \lambda_1^*, \lambda_2^*)$ to be a minimum, the differential, $df(x_1^*, x_2^*, y_1^*, y_2^*, \lambda_1^*, \lambda_2^*)$ must be 0. Taking the differential of (2.55), we get the following set of equations that must be solved

$$2(x_1^* - x_2^*) + \lambda_1^*(2x_1^* b_1^2) = 0, \tag{2.56}$$

$$2(y_1^* - y_2^*) + \lambda_1^*(2y_1^* a_1^2) = 0, \tag{2.57}$$

$$-2(x_1^* - x_2^*) + \lambda_2^*(2x_2^* b_2^2) = 0, \tag{2.58}$$

$$-2(y_1^* - y_2^*) + \lambda_2^*(2y_2^* a_2^2) = 0, \tag{2.59}$$

$$x_1^{*2} b_1^2 + y_1^{*2} a_1^2 - a_1^2 b_1^2 = 0, \tag{2.60}$$

$$x_2^{*2} b_2^2 + y_2^{*2} a_2^2 - a_2^2 b_2^2 = 0. \tag{2.61}$$

The solution of (2.56) through (2.61) yields Lemma 2. □

The second step of the optimization process requires substitution of x_1^*, y_1^*, x_2^*, and y_2^* into (2.31). Doing so and solving for the minimum volume ellipse, E_2, results in the following lemma.

Lemma 3 *The minimum volume ellipse, E_2, whose surface, S_2, is at least $\zeta > 0$*

from the surface S_1 of ellipse E_1 is parameterized by

$$a_2 = \begin{cases} a_1 + \zeta & a_1 = b_1 \\ p_6 & \textit{otherwise} \end{cases} \tag{2.62}$$

$$b_2 = \begin{cases} \frac{1}{a_1\sqrt{2}}\sqrt{p_8 + (a_1^2 - b_1^2)\sqrt{p_7}} & a_1 > b_1 \\ b_1 + \zeta & a_1 = b_1 \\ \frac{1}{a_1\sqrt{2}}\sqrt{p_8 + (b_1^2 - a_1^2)\sqrt{p_7}} & a_1 < b_1 \end{cases} \tag{2.63}$$

where p_1 through p_8 are defined in Appendix A.

Proof: As mentioned above, we begin by substituting x_1^*, y_1^*, x_2^*, and y_2^* into (2.31). However, Lemma 2 gives four possible solutions which depend on the values of a_1, b_1, a_2, and b_2. Clearly, we are not interested in the 3rd case, that is, when the ellipses intersect. The case that we are most interested in is case 4, that is, when the minimum distance is between the axes. However, case 4 is not a solution when $a_1 = b_1$.

We begin by considering the case $a_1 = b_1$, and assume that the minimum distance is along the y-axis, i.e., case 1. We wish to minimize the volume of E_2, which is equivalent to minimizing

$$V = a_2 b_2. \tag{2.64}$$

If the minimum distance is along the y-axis then $b_2 - b_1 = \zeta$ and $b_2 = b_1 + \zeta$. Also implied by (2.40) is that $a_2 - a_1 \geq \zeta$. Consequently, $a_2 = a_1 + \zeta$ if (2.64) is to be minimized. Clearly, if we assume that the minimum distance was along the x-axis, we would get the same result.

Now consider the case when $a_1 \neq b_1$ and assume that the minimum distance is between the axes. The constraint posed by (2.49) can be appended to (2.64) giving

$$V_a(a_2, b_2, \lambda) = a_2 b_2 + \lambda((b_2^2 - a_2^2 - b_1^2 + a_1^2)(b_1^2 a_2^2 - a_1^2 b_2^2) - \zeta^2(b_2^2 - a_2^2)(b_1^2 - a_1^2)) \tag{2.65}$$

to be minimized. For $V_a(a_2^*, b_2^*, \lambda^*)$ to be a minimum, the differential, $dV_a(a_2^*, b_2^*, \lambda^*)$, must be 0. Taking the differential of (2.65), we get the following set of equations that must be solved

$$\begin{aligned} b_2^* + 2\lambda^*(a_2^* a_1^2 b_2^{*2} - 2b_1^2 a_2^{*3} + b_1^2 a_2^* b_2^{*2} + b_1^2 a_2^* a_1^2 - b_1^4 a_2^* \\ - \zeta^2 a_2^* a_1^2 + \zeta^2 a_2^* b_1^2) &= 0 \qquad (2.66) \\ a_2^* + 2\lambda^*(b_2^* b_1^2 a_2^{*2} - 2a_1^2 b_2^{*3} + a_1^2 b_2^* a_2^{*2} + a_1^2 b_2^* b_1^2 - a_1^4 b_2^* \\ - \zeta^2 b_2^* b_1^2 + \zeta^2 b_2^* a_1^2) &= 0 \qquad (2.67) \\ (b_2^{*2} - a_2^{*2} - b_1^2 + a_1^2)(b_1^2 a_2^{*2} - a_1^2 b_2^{*2}) - \zeta^2 (b_2^{*2} - a_2^{*2})(b_1^2 - a_1^2) &= 0 \qquad (2.68) \end{aligned}$$

The solution of (2.66), (2.67), and (2.68) yields Lemma 3. □

We can now prove the following theorem.

Theorem 1 *For $r = 2$, let $\tilde{P}_{k-1}$ be factored as $\tilde{P}_{k-1} = V_{k-1}\Lambda_{k-1}V_{k-1}^T$ where $V_{k-1} \in \Re^{2\times 2}$ is orthogonal and*

$$\Lambda_{k-1} = \begin{bmatrix} a_1^2 & 0 \\ 0 & b_1^2 \end{bmatrix}. \qquad (2.69)$$

Then, the minimum volume ellipse, $\hat{E}_k$, which contains the set, $\hat{G}_k$, where $\hat{E}_k$ is defined by (2.9) and $\hat{G}_k$ is defined by (2.26), is given by

$$\hat{P}_k = V_{k-1}\hat{\Lambda}_k V_{k-1}^T \qquad (2.70)$$

where

$$\hat{\Lambda}_k = \begin{bmatrix} a_2^2 & 0 \\ 0 & b_2^2 \end{bmatrix}, \qquad (2.71)$$

and a_2 and b_2 are given by Lemma 3 with ζ replaced by ζ_{k-1}.

Proof: The proof follows directly from equations (2.33) and (2.34) and Lemmas 1-3.

□

Example of Minimum Volume Ellipse

In this section we will apply Theorem 1 to a particular E_{k-1} and ζ_{k-1}. Let $\zeta_{k-1} = 0.5$ and E_{k-1} be given by (2.8) with

$$\tilde{P}_{k-1} = \begin{bmatrix} 25 & -5 \\ -5 & 5 \end{bmatrix}, \tag{2.72}$$

$$\tilde{\theta}_{k-1} = \begin{bmatrix} 6 \\ 7 \end{bmatrix}. \tag{2.73}$$

The SVD of $\tilde{P}_{k-1}$ is given by

$$V_{k-1} = \begin{bmatrix} -0.9732 & -0.2298 \\ 0.2298 & -0.9732 \end{bmatrix}, \tag{2.74}$$

$$\Lambda_{k-1} = \begin{bmatrix} 26.1803 & 0 \\ 0 & 3.8197 \end{bmatrix}. \tag{2.75}$$

From (2.69), we get a_1 and b_1 which are used by Lemma 3 to solve for a_2 and b_2. Substituting a_2 and b_2 into (2.71) results in

$$\hat{\Lambda}_k = \begin{bmatrix} 32.6125 & 0 \\ 0 & 6.1300 \end{bmatrix}. \tag{2.76}$$

Using (2.33) and (2.34), $\hat{E}_k$ is given by (2.9) with

$$\hat{P}_k = \begin{bmatrix} 31.2145 & -5.9217 \\ -5.9217 & 7.5279 \end{bmatrix}, \tag{2.77}$$

$$\hat{\theta}_k = \begin{bmatrix} 6 \\ 7 \end{bmatrix}. \tag{2.78}$$

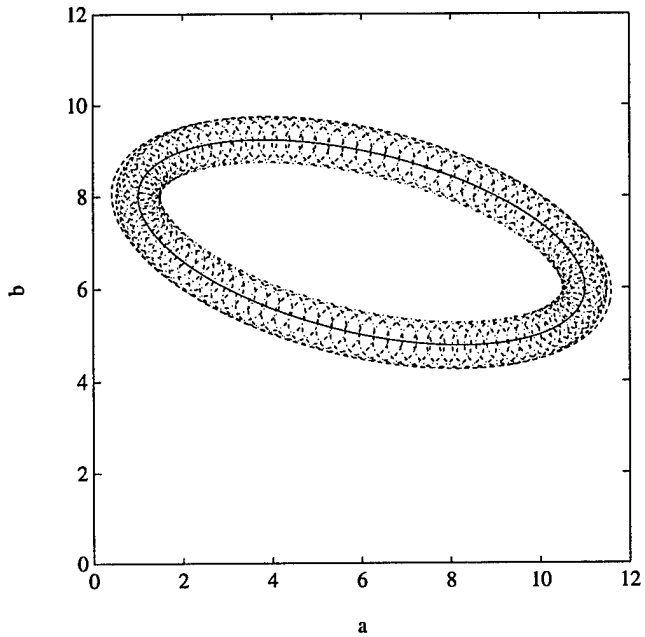

Figure 3: Minimum volume ellipse, $\hat{E}_k$, that bounds $\hat{G}_k$

In Figure 3 we see that $\hat{E}_k$ does bound $\hat{G}_k$. Note that although $\hat{E}_k$ has the smallest volume of any ellipse that contains $\hat{G}_k$, it is still larger than $\hat{G}_k$ except for the case $a_1 = b_1$. However, except for the cases where $a_1 \gg b_1$ or $a_1 \ll b_1$ this difference in volume is small.

2.4.2 r-Dimensional Case

To generalize this to r-dimensions, we will use a result by Chernous'ko. In [15], Chernous'ko finds the minimum volume ellipsoid which bounds the sum of two vectors which are also ellipsoidally bounded. This is exactly the problem posed by (2.11).

The solution is given as follows:

1. Solve the generalized eigenvalue problem

$$R_{k-1}x_j = \lambda_j \tilde{P}_{k-1}x_j \tag{2.79}$$

for each λ_j, $j \in [1, r]$.

2. Solve

$$\sum_{j=1}^{r} \frac{1}{\lambda_j + p} = \frac{r}{p(p+1)} \tag{2.80}$$

for the unique $p > 0$.

3. Then, the solution to (2.11) is given by

$$\hat{\theta}_k = \tilde{\theta}_{k-1} \tag{2.81}$$

$$\hat{P}_k = (p+1)\tilde{P}_{k-1} + (p^{-1}+1)R_{k-1} \tag{2.82}$$

To arrive at these equations, a nonsingular matrix can be found which transforms the ellipsoid E_{k-1} to a unit ball and the ellipsoid G_k to an ellipsoid whose semi-axes are aligned with the coordinate axes. After the transformation, the semi-axes length of the ellipsoid G_k are given by the square root of the generalized eigenvalues of (2.79). The optimization in (2.11) is then solved in the transformed coordinate system. This optimization requires the solution of (2.80). A proof that (2.80) has only one positive root can be found in [15]. After transforming back to the original coordinate system, the solution to (2.11) is given by (2.81) and (2.82). Note that R_k is no longer constrained to be $\zeta_k^2 I$.

2.5 OVETV Algorithm Summary

The steps of the Optimal Volume Ellipsoid for Time-Varying systems (OVETV) algorithm are summarized below.

1. Choose the initial ellipsoid, E_0, "large enough" such that the initial "true" parameter vector, θ_0, is in E_0.

2. Find $\hat{E}_k$ such that

$$\hat{E}_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \hat{G}_k\}. \tag{2.83}$$

3. Find E_k such that

$$E_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \hat{E}_k \cap F_k\}. \tag{2.84}$$

4. Repeat steps two and three for each new measurement.

2.6 Alternative Strategies

The optimal time update equations require the solution of an rth order generalized eigenvalue problem (2.79) and an $(r+1)$ order polynomial (2.80). Computationally, this may not be feasible for certain real-time applications, especially when r is large. Consequently, we will consider two alternative algorithms which represent different degrees of tradeoff between computational complexity and resulting volume. These

algorithms also use the MOVE algorithm to solve (2.12); thus, we will only discuss the approximating solutions to (2.11). These algorithms reduce the computational complexity by parameterizing the new ellipsoid with one parameter instead of r parameters. Again, we will begin with the two-dimensional case.

2.6.1 Two-Dimensional Case

If we consider the 2-dimensional system described in Section 2.4.1, one possible parameterization is given by

$$a_2 = \eta a_1 \tag{2.85}$$

$$b_2 = \eta b_1. \tag{2.86}$$

The parameterization in (2.85) and (2.86), scalar multiplication, is appealing because it does not require finding all the singular values of P_{k-1}. Note that this parameterization was refered to as scalar bound inflation in [13]. Unfortunately, it may yield a very conservative bounding ellipsoid. For $\hat{E}_k$ to contain $\hat{G}_k$, the following conditions on the semi-axes must hold:

$$a_2 \geq a_1 + \zeta \tag{2.87}$$

$$b_2 \geq b_1 + \zeta. \tag{2.88}$$

Substituting (2.86) into (2.88) and solving for η, we get

$$\eta \geq 1 + \frac{\zeta}{b_1}. \tag{2.89}$$

Substituting (2.89) into (2.85) yields

$$a_2 \geq a_1 + \zeta \frac{a_1}{b_1}. \tag{2.90}$$

Clearly, for $a_1 \gg b_1$, a_2 can become very large.

For an example, see Figure 4. Here, we set $a_1 = 9$, $b_1 = 1$, and vary ζ from 0 to 1.9. This plot contains a_2 and b_2 which are the optimal values calculated using Theorem 1. Also plotted are the lower bounds, (2.88) and (2.90), for the parameterization in (2.85) and (2.86). Clearly, for $\zeta > 0$ this parameterization will produce an ellipse which is much larger than optimum. Despite this conservativeness, we will state the following lemma without proof. It's implications will be discussed in the next section.

Lemma 4 *Let $a_2 = a_1\eta$, $b_2 = b_1\eta$ where $\eta \geq 0$. Let the surfaces of ellipses, E_1 and E_2, be defined by (2.35) and (2.36), then the value of η which minimizes the volume of E_2 such that the surface, S_2, is at least $\zeta > 0$ from S_1 is*

$$\eta = 1 + \frac{\zeta}{\chi}, \tag{2.91}$$

where $\chi = \min(a_1, b_1)$.

Examining Figure 4, it appears that the change in a_2 versus ζ and the change in b_2 versus ζ are similar. This suggests parameterizing the new ellipse by scalar addition, that is

$$a_2 = a_1 + \eta \tag{2.92}$$

$$b_2 = b_1 + \eta, \tag{2.93}$$

which leads to the following lemmas.

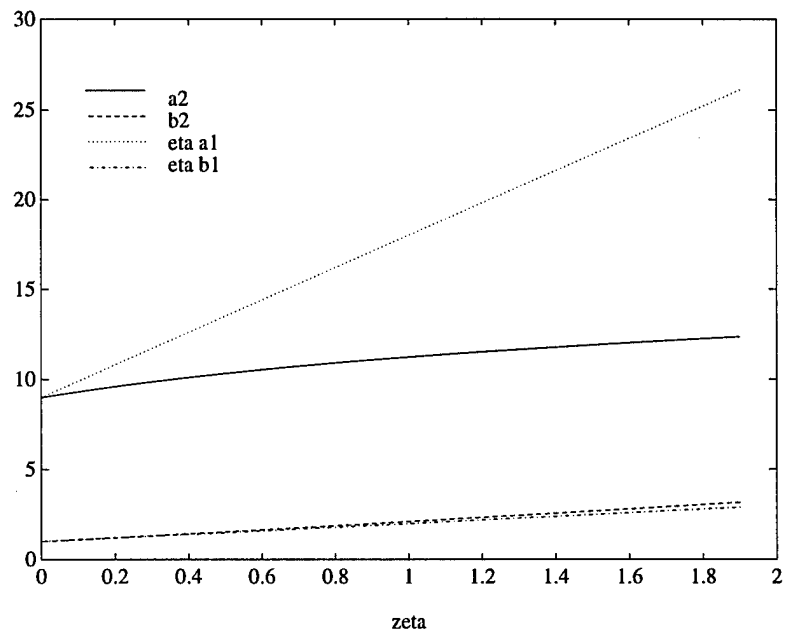

Figure 4: Comparison of ellipse parameters between optimum and scalar multiplication

Lemma 5 *Let $a_2 = a_1 + \eta$, $b_2 = b_1 + \eta$ where $\eta \geq 0$. Let the surfaces of ellipses, E_1 and E_2, be defined by (2.35) and (2.36). Then the minimum distance between S_1 and S_2 is given by*

$$d = \sqrt{2\eta^2 \frac{2b_1 a_1 + b_1\eta + a_1\eta}{(b_1 + a_1)(b_1 + 2\eta + a_1)}} \tag{2.94}$$

Proof: This lemma follows from Lemma 2. Substituting (2.92) and (2.93) into (2.49) yields (2.69). A more direct proof will be used to extend this method to r dimensions.

□

Lemma 6 *Let $a_2 = a_1 + \eta$, $b_2 = b_1 + \eta$ where $\eta \geq 0$. Let the surfaces of ellipses, E_1 and E_2, be defined by (2.35) and (2.36). Then the value of η which minimizes the volume of E_2 such that the surface, S_2, is at least $\zeta > 0$ from S_1 is*

$$\eta = p_7 + p_6 - \frac{2}{3}\frac{b_1 a_1}{p0}, \tag{2.95}$$

where

$$p_0 = a_1 + b_1 \tag{2.96}$$

$$p_1 = 27b_1^4 + 36b_1^3 a_1 + 2a_1^2 b_1^2 + 36b_1 a_1^3 + 2a_1^4 \tag{2.97}$$

$$p_2 = \frac{\zeta}{p_0}\sqrt{3(-16p_0^2\zeta^4 + p_1\zeta^2 - 64b_1^3 a_1^3)} \tag{2.98}$$

$$p_3 = \frac{-a_1^3 b_1^3}{8p_0^3} \tag{2.99}$$

$$p_4 = -\frac{1}{2}\zeta^2 p_0 \tag{2.100}$$

$$p_5 = \frac{a_1 b_1 \zeta^2}{2p_0} \tag{2.101}$$

$$p_6 = \sqrt[3]{-\frac{2}{3}p_5 - \frac{1}{2}p_4 + \frac{64}{27}p_3 - \frac{1}{36}p_2} \tag{2.102}$$

$$p_7 = \sqrt[3]{-\frac{2}{3}p_5 - \frac{1}{2}p_4 + \frac{64}{27}p_3 + \frac{1}{36}p_2} \tag{2.103}$$

Proof: By Lemma 1, for E_2 to have minimum volume, the minimum distance, d, in (2.94) must be equal to ζ. Solving (2.94) for η yields (2.95). □

We can now prove the following theorem.

Theorem 2 *For $r = 2$, let $\tilde{P}_{k-1}$ be factored as $\tilde{P}_{k-1} = V_{k-1}\Lambda_{k-1}V_{k-1}^T$ where $V_{k-1} \in \Re^{2\times 2}$ is orthogonal and*

$$\Lambda_{k-1} = \begin{bmatrix} a_1^2 & 0 \\ 0 & b_1^2 \end{bmatrix}. \tag{2.104}$$

Let the semi-axes of $\hat{E}_k$ be parameterized as $a_2 = a_1 + \eta$ and $b_2 = b_1 + \eta$. Then, the minimum volume ellipse, $\hat{E}_k$, which contains the set $\hat{G}_k$ where $\hat{E}_k$ is defined by (2.9) and $\hat{G}_k$ is defined by (2.26), is given by

$$\hat{P}_k = V_{k-1}\hat{\Lambda}_k V_{k-1}^T \tag{2.105}$$

where

$$\hat{\Lambda}_k = \begin{bmatrix} a_2^2 & 0 \\ 0 & b_2^2 \end{bmatrix}, \tag{2.106}$$

and η is given by Lemma 6 with ζ replaced by ζ_{k-1}.

Proof: The proof follows directly from equations (2.33) and (2.34) and Lemmas 5 and 6. □

In order to compare the values of Theorem 2 with the optimum values of Theorem 1, again we set $a_1 = 9$ and $b_1 = 1$. This time ζ is varied from 0 to 8.9. Shown in Figure 5 are the plots of a_2 and b_2 from Theorem 1 and Theorem 2. Clearly, this is a much closer approximation than the one given by (2.85) and (2.86). The volumes

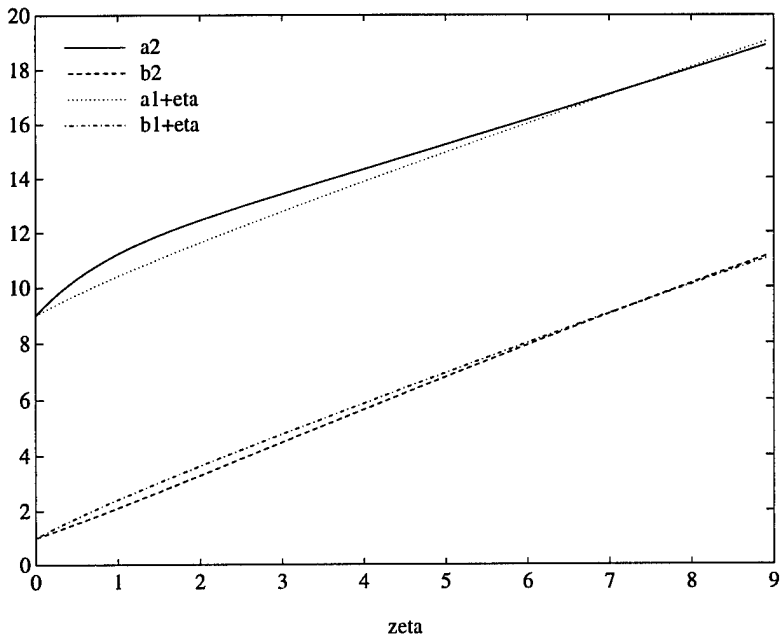

Figure 5: Comparison of ellipse parameters between optimum and scalar addition

were calculated for the parameters shown in Figure 5. A ratio of the volume using scalar addition versus the optimum volume is shown in Figure 6. For this example, the volume using scalar addition is at most approximately 7% higher than the optimum volume.

Theorem 2 presents a reasonable alternative to Theorem 1. In the next section, we will extend it as well as scalar multiplication to r dimensions.

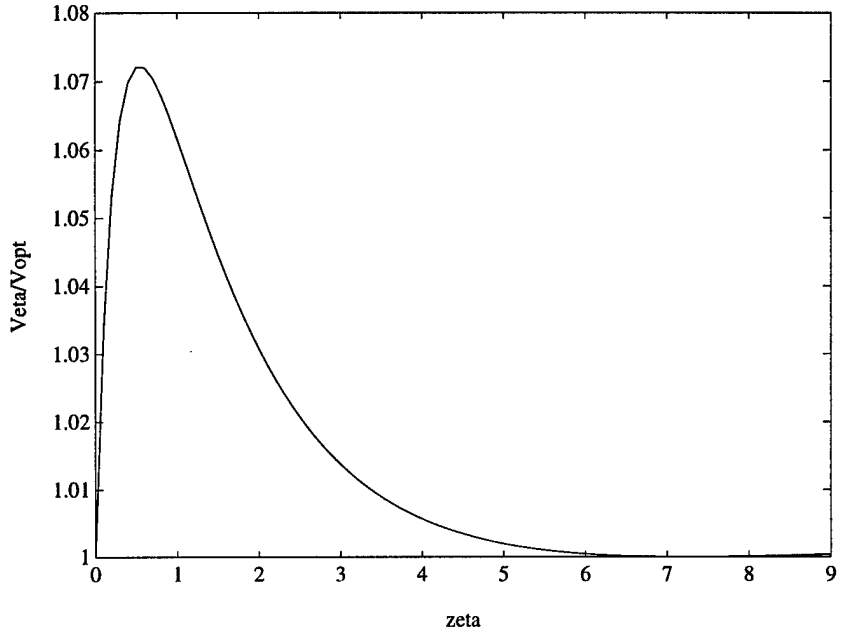

Figure 6: Ratio of scalar addition volume versus optimum volume

2.6.2 r-Dimensional Case

Scalar Addition

In the extension to r-dimensions, we see that $\hat{E}_k$ is constrained to be an ellipsoid with the same center and orientation as E_{k-1} and with semi-axis lengths which are found by adding a scalar, η, to the semi-axis lengths of E_{k-1}. R_k is again constrained to be $\zeta_k^2 I$; therefore, $\hat{G}_k$ is given by (2.26).

As in the 2-dimensional case, we need to parameterize the minimum distance

between the following ellipsoids. Define the surface of ellipsoids, E_1 and E_2, as

$$S_1 = \{x \in \Re^r : \sum_{i=1}^{r} \frac{x_i^2}{a_i^2} = 1\} \tag{2.107}$$

$$S_2 = \{\tilde{x} \in \Re^r : \sum_{i=1}^{r} \frac{\tilde{x}_i^2}{(a_i + \eta)^2} = 1\} \tag{2.108}$$

where each a_i, $i \in [1, r]$ and η are positive scalars.

Lemma 7 *Let the surfaces of ellipsoids, E_1 and E_2, be defined by (2.107) and (2.108). Then, the minimum distance between S_1 and S_2 is given by*

$$d = \sqrt{2\eta^2 \frac{2\overline{a}\underline{a} + \overline{a}\eta + \underline{a}\eta}{(\overline{a} + \underline{a})(\overline{a} + 2\eta + \underline{a})}} \tag{2.109}$$

where

$$\overline{a} = \max_{j \in [1,r]} a_j \qquad \underline{a} = \min_{j \in [1,r]} a_j. \tag{2.110}$$

Proof: Because $d(x, \tilde{x})$ is nonnegative, minimizing d^2 is equivalent to minimizing d. Using the Lagrange multiplier method, the constraints posed by the ellipsoids, (2.107) and (2.108), can be augmented to d^2 to form

$$f(x, \tilde{x}, \lambda_1, \lambda_2) = \sum_{i=1}^{n} (x_i - \tilde{x}_i)^2 + \lambda_1 (\sum_{i=1}^{r} \frac{x_i^2}{a_i^2} - 1) + \lambda_2 (\sum_{i=1}^{r} \frac{\tilde{x}_i^2}{(a_i + \eta)^2} - 1). \tag{2.111}$$

For $f(x^*, \tilde{x}^*, \lambda_1^*, \lambda_2^*)$ to be a minimum, the differential, $df(x^*, \tilde{x}^*, \lambda_1^*, \lambda_2^*)$ must be 0. Taking the differential of (2.111), we get the following set of equations that must be solved

$$2(x_j^* - \tilde{x}_j^*) + 2\lambda_1^* \frac{x_j^*}{a_j^2} = 0 \qquad j \in [1, r] \tag{2.112}$$

$$-2(x_j^* - \tilde{x}_j^*) + 2\lambda_2^* \frac{\tilde{x}_j^*}{(a_j + \eta)^2} = 0 \qquad j \in [1, r] \tag{2.113}$$

$$\sum_{i=1}^{r} \frac{(x_i^*)^2}{a_i^2} - 1 = 0 \tag{2.114}$$

$$\sum_{i=1}^{r} \frac{(\tilde{x}_i^*)^2}{(a_i + \eta)^2} - 1 = 0 \tag{2.115}$$

The solution of (2.112) through (2.115) yields Lemma 7. □

This lemma holds because if the semi-axes of ellipsoid, E_1, are incremented by a scalar η, then the minimum distance between the surface of E_1 and the surface of the resulting ellipsoid, E_2, must lie along the two-dimensional cross-section defined by the maximum and minimum length semi-axes.

We now state the following theorem:

Theorem 3 *Let E_{k-1} and $\hat{E}_k$ be defined as in (2.8) and (2.9). Let $\hat{G}_k$ be given by (2.26). If the SVD of $\tilde{P}_{k-1}$ is given by $\tilde{P}_{k-1} = V_{k-1}\Lambda_{k-1}V_{k-1}^T$ where V_{k-1} is orthogonal and $\Lambda_{k-1} = diag(\lambda_1 \ldots \lambda_r)$, then the value of η which optimizes*

$$\eta = \arg_\eta \min\{vol(E) : E \supset \hat{G}_k; P = V_{k-1}(\Lambda_{k-1}^{1/2} + \eta I)^2 V_{k-1}^T\} \tag{2.116}$$

is given by

$$\eta = p_7 + p_6 - \frac{2}{3}\frac{\underline{a}\overline{a}}{p0}, \tag{2.117}$$

where

$$\overline{a} = \max_{j\in[1,r]} \sqrt{\lambda_j} \qquad \underline{a} = \min_{j\in[1,r]} \sqrt{\lambda_j}, \tag{2.118}$$

and

$$p_0 = \overline{a} + \underline{a} \tag{2.119}$$

$$p_1 = 27\underline{a}^4 + 36\underline{a}^3\overline{a} + 2\overline{a}^2\underline{a}^2 + 36\underline{a}\overline{a}^3 + 2\overline{a}^4 \tag{2.120}$$

$$p_2 = \frac{\zeta_{k-1}}{p_0}\sqrt{3(-16p_0^2\zeta_{k-1}^4 + p_1\zeta_{k-1}^2 - 64\underline{a}^3\overline{a}^3)} \tag{2.121}$$

$$p_3 = \frac{-\overline{a}^3\underline{a}^3}{8p_0^3} \tag{2.122}$$

$$p_4 = -\frac{1}{2}\zeta_{k-1}^2 p_0 \tag{2.123}$$

$$p_5 = \frac{\overline{a}\underline{a}\zeta_{k-1}^2}{2p_0} \tag{2.124}$$

$$p_6 = \sqrt[3]{-\frac{2}{3}p_5 - \frac{1}{2}p_4 + \frac{64}{27}p_3 - \frac{1}{36}p_2} \tag{2.125}$$

$$p_7 = \sqrt[3]{-\frac{2}{3}p_5 - \frac{1}{2}p_4 + \frac{64}{27}p_3 + \frac{1}{36}p_2} \tag{2.126}$$

Proof: The proof follows directly from equations (2.33) and (2.34) and Lemmas 6 and 7. □

Scalar Multiplication

In the examples we have considered, Theorem 3 provides an acceptable alternative to the optimal algorithm. Notice that the $(r+1)$ order polynomial in (2.80) has been reduced to the same computational complexity as the 2nd order problem. However, the algorithm still requires the solution of an rth order SVD.

With this in mind, we reconsider the scalar multiplication strategy discussed in section 2.6.1. In this algorithm, $\hat{E}_k$ is constrained to be an ellipsoid with the same center and orientation as E_{k-1}; the semi-axis lengths are found by multiplying the semi-axes's of E_{k-1} with a scalar, η.

Theorem 4 *Let E_{k-1} and $\hat{E}_k$ be defined as in (2.8) and (2.9). Let $\hat{G}_k$ be given by*

(2.10). Then the value of η which optimizes

$$\eta = \arg_\eta \min\{vol(\hat{E}_k) : \hat{E}_k \supset \hat{G}_k, \hat{\theta}_k = \tilde{\theta}_{k-1}, \hat{P}_k = \eta^2 \tilde{P}_{k-1}\} \tag{2.127}$$

is given by

$$\eta = 1 + \sqrt{\bar{\lambda}} \tag{2.128}$$

where $\bar{\lambda}$ is the maximum generalized eigenvalue, λ_j, which satisfies

$$R_{k-1} x_j = \lambda_j \tilde{P}_{k-1} x_j. \tag{2.129}$$

Proof: From [26], we know that $\hat{E} \supset \hat{G}_k$ if and only if

$$S_{\hat{E}_k}(\theta) \geq S_{\hat{G}_k}(\theta) \tag{2.130}$$

where $S(\theta)$ defines the support function of the respective sets. We also know that

$$S_{\hat{G}_k}(\theta) = S_{E_{k-1}}(\theta) + S_{w_k}(\theta), \tag{2.131}$$

with the support functions given by

$$S_{\hat{E}_k}(\theta) = \theta^T \hat{\theta}_k + \sqrt{\theta^T \hat{P}_k \theta}, \tag{2.132}$$

$$S_{E_{k-1}}(\theta) = \theta^T \tilde{\theta}_{k-1} + \sqrt{\theta^T \tilde{P}_{k-1} \theta}, \tag{2.133}$$

$$S_{w_{k-1}}(\theta) = \sqrt{\theta^T R_{k-1} \theta}. \tag{2.134}$$

Substituting (2.131)-(2.134) into (2.130) results in

$$\theta^T \hat{\theta}_k + \sqrt{\theta^T \hat{P}_k \theta} \geq \theta^T \tilde{\theta}_{k-1} + \sqrt{\theta^T \tilde{P}_{k-1} \theta} + \sqrt{\theta^T R_{k-1} \theta}. \tag{2.135}$$

By substituting $\hat{\theta}_k = \tilde{\theta}_{k-1}$ and $\hat{P}_k = \eta^2 \tilde{P}_{k-1}$ into (2.135), it is easy to show that

$$(\eta - 1)^2 \theta^T \tilde{P}_{k-1} \theta \geq \theta^T R_{k-1} \theta. \tag{2.136}$$

From [27], we know that there exists a matrix transformation, $\theta = T\bar{\theta}$, such that

$$(\eta - 1)^2 \bar{\theta}^T \bar{\theta} \geq \bar{\theta}^T \Lambda \bar{\theta}, \tag{2.137}$$

where Λ is a diagonal matrix containing the generalized eigenvalues of (2.129). For (2.137) to hold, it is necessary that $(\eta - 1)^2 \geq \bar{\lambda}$ where $\bar{\lambda}$ is the maximum generalized eigenvalue of (2.129); thus $\eta \geq 1 + \sqrt{\bar{\lambda}}$. The volume of $\hat{E}_k$ is proportional to $\sqrt{\det(\hat{P}_k)}$ which is minimized if the equality holds; therefore, $\eta = 1 + \sqrt{\bar{\lambda}}$. □

Note that this algorithm does not require the solution of an $(r+1)$ order polynomial. Furthermore, the rth order generalized eigenvalue problem of (2.79) is replaced with finding the largest eigenvalue. There exist iterative techniques which can be used to find this value quickly. We could also use the relationship, $\|\tilde{P}_{k-1}^{-1} R_{k-1}\| \geq \bar{\lambda}$ [28], to replace η in (2.128) with $\tilde{\eta} = 1 + \sqrt{\|\tilde{P}_{k-1}^{-1} R_{k-1}\|}$ where $\|\cdot\|$ is any matrix norm; note that $\tilde{\eta} \geq \eta$. Also, note that $\tilde{P}_{k-1}$ is given recursively by [29]

$$\tilde{P}_k^{-1} = \frac{1}{\delta_k} \hat{P}_k^{-1} + \left(\frac{1}{\sigma_k} - \frac{1}{\delta_k}\right) \frac{\phi_k \phi_k^T}{\phi_k^T \hat{P}_k \phi_k} \tag{2.138}$$

where $\hat{P}_k^{-1} = \frac{1}{\tilde{\eta}^2} \tilde{P}_{k-1}^{-1}$.

Theorem 4 always gives more conservative results than the OVETV. For the case where $R_k = \zeta_k^2 I$, Theorem 4 often gives more conservative results than Theorem 3. However, the reduced number of computations may make it the most acceptable for real time applications.

2.7 Summary

In this chapter, we developed the optimal volume ellipsoid algorithm for parameter set estimation of time-varying systems. We also developed two alternative strategies that offer various degrees of trade off between computation complexity and ellipsoid volume. Given our assumptions in (2.1)-(2.4), all three algorithms are guaranteed to contain the "true" parameter in the parameter set. While OVETV gives the optimum volume ellipsoid, the scalar addition and scalar multiplication algorithms give optimum volume ellipsoids assuming additional constraints on the time update ellipsoid.

CHAPTER III

Examples

3.1 Overview

In his chapter, we consider three examples. The first is a simple first-order example which serves to illustrate the main features of this approach and to compare the OVETV algorithm with the scalar addition and scalar multiplication approaches. In the second example, identification of a linear time-varying circuit is used to demonstrate how this approach can be applied to sampled-data systems with quantization noise. Finally, we show how this approach can be applied to actual data obtained from a crude oil distillation column.

3.2 First-order Example

Consider the following time varying system,

$$y_k = -a_k y_{k-1} + b_k u_{k-1} + v_k \tag{3.1}$$

$$a_k = a_k + 0.0005 \tag{3.2}$$

$$b_k = b_k + 0.001 \tag{3.3}$$

where v_k is a uniformly distributed random noise between $[-0.1, 0.1]$. For this simulation u_k is a uniformly distributed random variable between $[-1, 1]$. Initial conditions for the system are $y_0 = 0$, $a_0 = 0.5$, and $b_0 = 1.0$.

To apply OVETV, the bounds in (2.3) and (2.4) must be specified. We set $\overline{\gamma}_k = -\underline{\gamma}_k = 0.1$ and $R_k = \zeta_k^2 I$ where $\zeta_k = \left\| \begin{bmatrix} 0.0005 & 0.001 \end{bmatrix}^T \right\|_2 = 0.00118$ and I is the identity matrix. The orientation matrix $\tilde{P}_0$ is set to $2I$, and the center $\tilde{\theta}_0$ is set to the origin.

In Figure 7 we see the results of the algorithm at one time step, $k = 16$. We begin with the ellipsoid at the previous time step, E_{15}. Applying the time update equations produces the ellipsoid $\hat{E}_{16}$. The measurements u_{16} and y_{16} give us the region between two hyperplanes, F_{16}. Finally, the measurement update equations are used to find the minimum volume ellipsoid, E_{16}, which bounds the intersection of $\hat{E}_{16}$ and F_{16}. Note that the +'s represent the "true" parameter vector at $k = 15$ and $k = 16$.

Results of the simulation for $k = 0$ to $k = 300$ are shown in Figure 8 with a_k at the bottom of the plot and b_k at the top. The upper and lower bounds are found by projecting the ellipsoids onto the coordinate axes [29]. Simulation results for the same system using scalar addition and scalar multiplication instead of the optimal time update equations are shown in Figures 9 and 10, respectively. The similarity of these plots to Figure 8 indicate that, for this system, very little performance is sacrificed using either scalar addition or scalar multiplication.

In Figure 11, the volume of E_k is plotted for the three simulations. Unlike the linear time invariant OVE and MOVE algorithms, OVETV has no guarantee of mono-

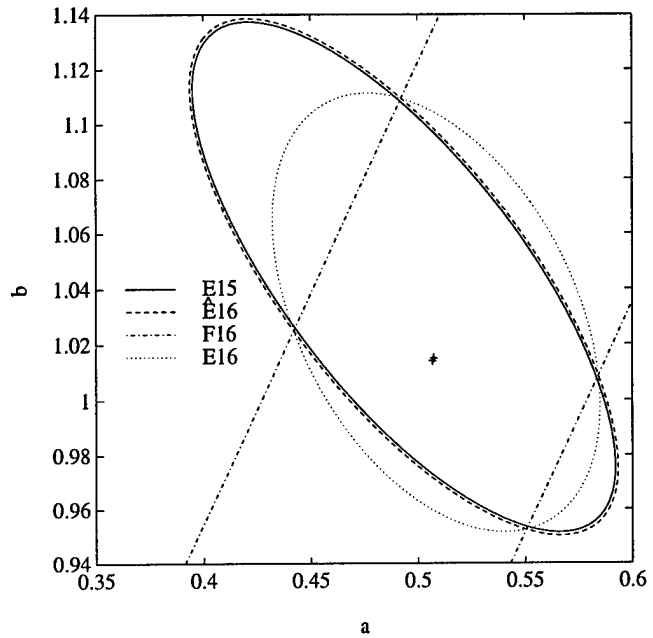

Figure 7: OVETV update at $k = 16$ for First-Order example

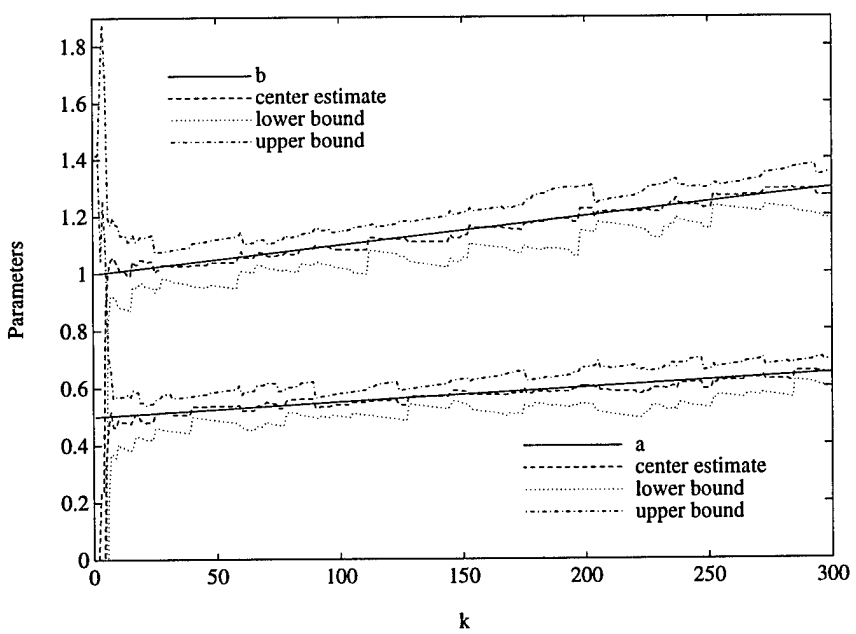

Figure 8: OVETV results for First-Order example

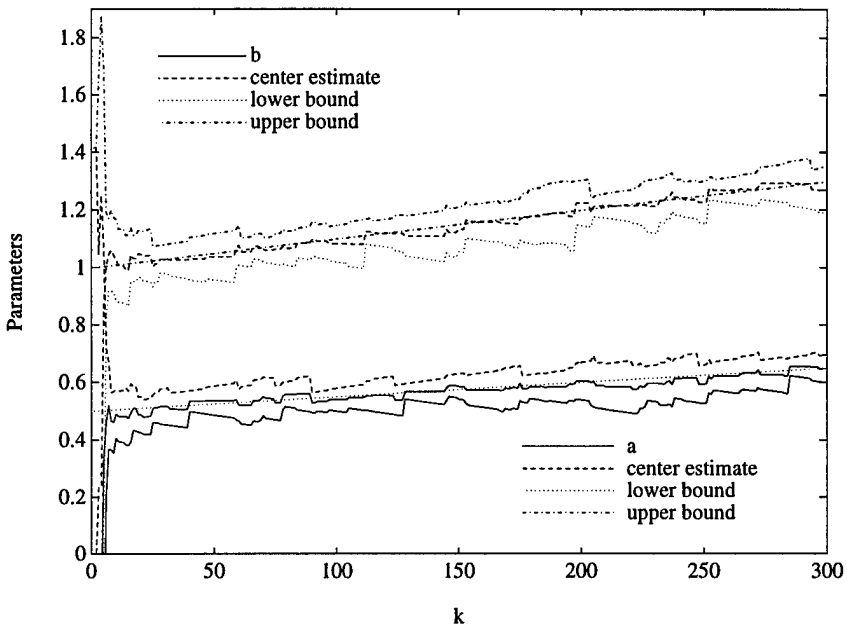

Figure 9: Scalar addition results for First-Order example

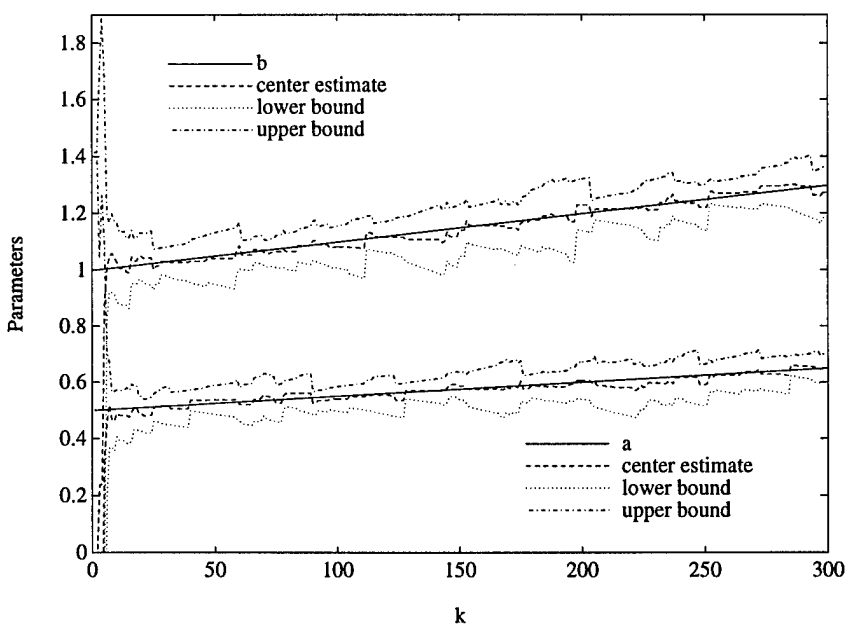

Figure 10: Scalar multiplication results for First-Order example

tonically decreasing volume. For this example, the volume of E_k using scalar addition is almost identical to the volume of E_k using the optimal update equations. The volume of E_k using scalar multiplication is larger, but the difference in volume is small. To ensure that the volume of E_k remains "small", persistent excitation is required. With "less" persistent excitation, the difference in volume between the algorithms would probably be larger.

The normalized center estimate error for all three simulations is plotted in Figure 12. This error is calculated according to

$$(\theta_k - \tilde{\theta}_k)^T \tilde{P}_k^{-1} (\theta_k - \tilde{\theta}_k)$$

where $\tilde{\theta}_k$ and $\tilde{P}_k$ parameterize E_k, and θ_k is the true parameter vector [9]. If the normalized center estimate error is less than one, then the true parameter vector, θ_k, is bounded by the ellipsoid, E_k; this consistency is guaranteed by all three algorithms.

3.3 Linear Time-Varying Circuit

Linear circuits with time-varying components arise in various applications. For example, the values of resistors, capacitors, and inductors all vary with temperature. Resistances and capacitances will also vary with adjustments in potentiometers and variable capacitors, respectively.

In this example, we consider a series R-L-C circuit across a voltage source [30].

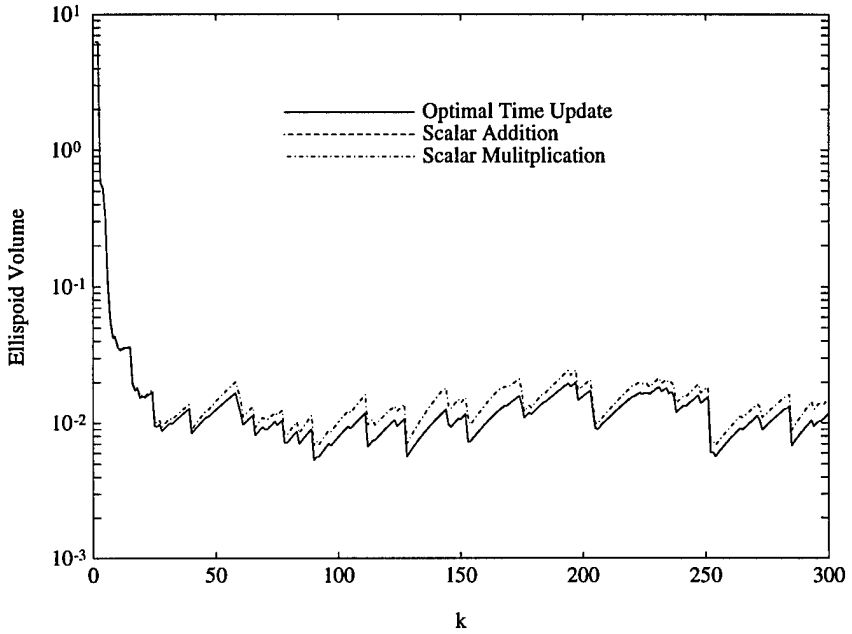

Figure 11: Ellipsoid volume for First-Order example

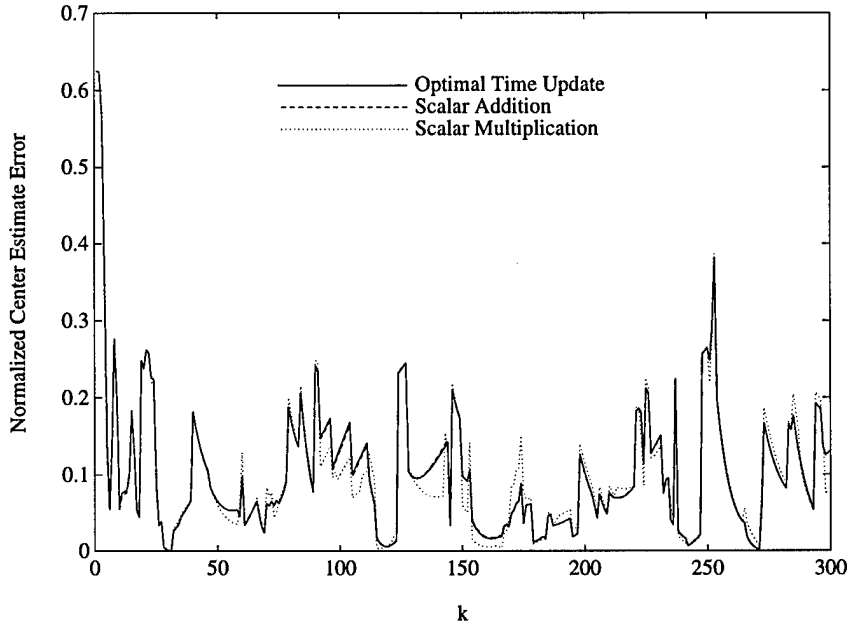

Figure 12: Normalized center estimate error for First-Order example

The state equations for this system are given as follows:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \tag{3.4}$$

$$y(t) = D(t)x(t), \tag{3.5}$$

where

$$A(t) = \begin{bmatrix} \frac{-\dot{c}(t)}{c(t)} & \frac{1}{c(t)} \\ \frac{-1}{l(t)} & \frac{-r(t)-\dot{l}(t)}{l(t)} \end{bmatrix}, \quad B(t) = \begin{bmatrix} 0 \\ \frac{1}{l(t)} \end{bmatrix}, \quad D(t) = \begin{bmatrix} 0 & r(t) \end{bmatrix}, \tag{3.6}$$

$x_1(t)$ is the voltage across the capacitance $c(t)$, $x_2(t)$ is the current through the inductance $l(t)$, $u(t)$ is the voltage source, and $y(t)$ is the voltage measured across the resistance $r(t)$.

To evaluate the performance of our algorithm, it is necessary that we discretize the state variable equations and then transform them to an ARX structure. Unfortunately, for linear time-varying systems, in general, there is no closed form solution for discretizing the state space equations. Consequently, we must resort to numerical methods. The discretized state space equations have the form

$$x((k+1)T) = G(kT)x(kT) + H(kT)u(kT), \tag{3.7}$$

$$y(kT) = D(kT)x(kT), \tag{3.8}$$

where T is the sampling time. The output matrix $D(kT)$ is simply the output matrix of (3.5) evaluated at time kT. To find the state matrix $G(kT)$ and the input matrix $H(kT)$, we must solve numerically for the state transition matrix, using

$$\dot{\Phi}(t, kT) = A(t)\Phi(t, kT) \quad \Phi(kT, kT) = I. \tag{3.9}$$

The matrices $G(kT)$ and $H(kT)$ can then be evaluated as [31]

$$G(kT) = \Phi((k+1)T, kT), \tag{3.10}$$

$$H(kT) = \Phi((k+1)T, kT) \int_{kT}^{(k+1)T} \Phi^{-1}(\tau, kT) B(\tau) d\tau. \tag{3.11}$$

Next, we need to transform from the state space equations to an ARX structure. Because the system matrices in (3.10) and (3.11) are time-varying, it is not possible to find the pulse-transfer-function in the usual way via the z transform. However, using the following Theorem (which extends a concept found in [30] for continuous time systems) we are able to transform the state space equations of (3.7) and (3.8) to the required form.

Theorem 5 *Given the linear time-varying SISO state space equation*

$$x((k+1)T) = G(kT)x(kT) + H(kT)u(kT), \tag{3.12}$$

$$y(kT) = D(kT)x(kT) + E(kT)u(kT), \tag{3.13}$$

where $x(kT) \in \Re^n$ and T is the sampling interval, define a sequence of $1 \times n$ matrices

$$\begin{aligned} L_0(kT) &= D(kT) \\ L_i(kT) &= L_{i-1}((k+1)T)G(kT) \qquad i = 1, 2, \cdots. \end{aligned} \tag{3.14}$$

If the rank of the matrix

$$\bar{L}(kT) = \begin{bmatrix} L_0(kT) \\ L_1(kT) \\ \vdots \\ L_{n-1}(kT) \end{bmatrix} \tag{3.15}$$

is n for all k, then $y(kT)$ satisfies a linear n^{th}-order difference equation of the form

$$y((k+n)T) + \sum_{i=1}^{n} \alpha_i(kT) y((k+n-i)T) = \sum_{i=0}^{n} \beta_i(kT) u((k+n-i)T), \tag{3.16}$$

where

$$\begin{bmatrix} \alpha_n(kT) & \alpha_{n-1}(kT) & \cdots & \alpha_1(kT) \end{bmatrix} = -L_n(kT)\bar{L}^{-1}(kT), \tag{3.17}$$

$$\begin{bmatrix} \beta_n(kT) & \beta_{n-1}(kT) & \cdots & \beta_0(kT) \end{bmatrix} = \bar{w}_n(kT) - L_n(kT)\bar{L}^{-1}(kT)\bar{W}(kT), \tag{3.18}$$

$$\bar{W}(kT) = \begin{bmatrix} w_{00}(kT) & 0 & \cdots & 0 & 0 \\ w_{10}(kT) & w_{11}(kT) & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ w_{(n-1)0}(kT) & w_{(n-1)1}(kT) & \cdots & w_{(n-1)(n-1)}(kT) & 0 \end{bmatrix}, \tag{3.19}$$

$$\bar{w}_n(kT) = \begin{bmatrix} w_{n0}(kT) & w_{n1}(kT) & \cdots & w_{n(n-1)}(kT) & w_{nn}(kT) \end{bmatrix}, \tag{3.20}$$

and the $w_{ij}(kT)$ are given recursively by

$$w_{0j}(kT) = \begin{cases} E(kT) & j = 0 \\ L_{(j-1)0}((k+1)T)H(kT) & j = 1, 2, \cdots, \end{cases} \tag{3.21}$$

$$w_{ij}(kT) = w_{(i-1)(j-1)}((k+1)T) \quad i = 1, 2, \cdots, \quad j = 1, 2, \cdots, i. \tag{3.22}$$

Proof: The proof is straight forward. From equation (3.13), we have

$$\begin{aligned} y(kT) &= D(kT)x(kT) + E(kT)u(kT) \\ &= L_0(kT)x(kT) + w_{00}(kT)u(kT), \end{aligned} \tag{3.23}$$

where $L_0(kT)$ and $w_{00}(kT)$ are as defined above. Applying the shift operator $q(\cdot)$, where $q(x(kT)) = x((k+1)T)$, to equation (3.23), we get

$$\begin{aligned} y((k+1)T) &= L_0((k+1)T)x((k+1)T) + w_{00}((k+1)T)u((k+1)T) \\ &= L_0((k+1)T)(G(kT)(x(kT) + H(kT)u(kT)) \\ &\quad + w_{00}((k+1)T)u((k+1)T) \\ &= L_1(kT)x(kT) + w_{10}(kT)u(kT) + w_{11}(kT)u((k+1)T). \end{aligned} \tag{3.24}$$

Repeatedly applying the shift operator, we find that in general,

$$y((k+i)T) = L_i(kT)x(kT) + \sum_{j=0}^{i} w_{ij}u((k+j)T). \tag{3.25}$$

From (3.25), we get the following equations

$$\bar{y}(kT) = \bar{L}(kT)x(kT) + \bar{W}(kT)\bar{u}(kT) \tag{3.26}$$

$$y((k+n)T) = L_n(kT)x(kT) + \bar{w}_n(kT)\bar{u}(kT). \tag{3.27}$$

where $\bar{L}(kT)$, $\bar{W}(kT)$, $L_n(kT)$, and $\bar{w}_n(kT)$ are as defined above, and

$$\bar{y}(kT) = \begin{bmatrix} y(kT) \\ y((k+1)T) \\ \vdots \\ y((k+n-1)T) \end{bmatrix} \quad \bar{u}(kT) = \begin{bmatrix} u(kT) \\ u((k+1)T) \\ \vdots \\ u((k+n)T) \end{bmatrix}. \tag{3.28}$$

Solving (3.26) for $x(kT)$ ($\bar{L}(kT)$ is invertible), and substituting the result into (3.27), we get

$$y((k+n)T) - L_n(kT)\bar{L}^{-1}(kT)\bar{y}(kT) = (\bar{w}_n(kT) - L_n(kT)\bar{L}^{-1}(kT)\bar{W}(kT))\bar{u}(kT). \tag{3.29}$$

The linear n^{th}-order difference equation of (3.16) follows directly from (3.29). □

Note that the condition that the rank of $\bar{L}(kT)$ be n guarantees that the system is observable (see [30] for a discussion of a concept in continuous time referred to as *instantaneously observable*). In fact, for the time-invariant case $\bar{L}$ reduces to the observability matrix, i.e., $\bar{L}^T = \begin{bmatrix} D^T & G^T D^T & \cdots & (G^T)^{n-1}D^T \end{bmatrix}$. This theorem can be extended to the single-input, multiple-output (SIMO) case by changing $\bar{L}^{-1}(kT)$ in the definition of α_i and β_i to $\bar{L}^+(kT)$, where $(\cdot)^+$ is the matrix pseudoinverse as

defined in [32]. In the next chapter, we will extend this theorem to multiple-input, single-output (MISO) systems.

To apply Theorem 5 to the circuit's discretized state space equations in (3.7) and (3.8), we need to check if

$$\bar{L}(kT) = \begin{bmatrix} 0 & d_2(kT) \\ d_2((k+1)T)g_{21}(kT) & d_2((k+1)T)g_{22}(kT), \end{bmatrix} \tag{3.30}$$

is invertible where $G(\cdot)$, $H(\cdot)$, and $D(\cdot)$ have the elements $g_{ij}(\cdot)$, $h_i(\cdot)$, and $d_j(\cdot)$, respectively. Clearly, $\bar{L}(kT)$ will be invertible provided that $d_2(kT)$ and $g_{21}(kT)$ are not equal to zero for all k. This will be the case for the system we consider.

Applying Theorem 5 to (3.7) and (3.8), we get the following difference equation

$$\begin{aligned} y(k+2) + \tfrac{d_2(k+2)}{d_2(k+1)g_{21}(k)}(-g_{11}(k)g_{21}(k+1) - g_{22}(k+1)g_{21}(k))y(k+1) \\ + \tfrac{d_2(k+2)g_{21}(k+1)}{d_2(k)g_{21}(k)}(g_{22}(k)g_{11}(k) - g_{12}(k)g_{21}(k))y(k) = \\ \tfrac{d_2(k+2)g_{21}(k+1)}{g_{21}(k)}(h_1(k)g_{21}(k) - h_2(k)g_{11}(k))u(k) + d_2(k+2)h_2(k+1)u(k+1) \end{aligned} \tag{3.31}$$

where the sampling time, T, is implied. If we apply the inverse shift operator $q^{-1}(\cdot)$ twice to (3.31), where $q^{-1}(x(kT)) = x((k-1)T)$, and solve for $y(k)$, we get the following equation

$$y_k = \theta_k^T \phi_k \tag{3.32}$$

where

$$\theta_k = \begin{bmatrix} \frac{d_2(k)}{d_2(k-1)g_{21}(k-2)}(-g_{11}(k-2)g_{21}(k-1) - g_{22}(k-1)g_{21}(k-2)) \\ \frac{d_2(k)g_{21}(k-1)}{d_2(k-2)g_{21}(k-2)}(g_{22}(k-2)g_{11}(k-2) - g_{12}(k-2)g_{21}(k-2)) \\ d_2(k)h_2(k-1) \\ \frac{d_2(k)g_{21}(k-1)}{g_{21}(k-2)}(h_1(k-2)g_{21}(k-2) - h_2(k-2)g_{11}(k-2)) \end{bmatrix} \tag{3.33}$$

$$\phi_k = \begin{bmatrix} -y_{k-1} & -y_{k-2} & u_{k-1} & u_{k-2} \end{bmatrix}^T. \tag{3.34}$$

The discrete time input-output equations of the linear time-varying circuit are given by equations (3.9)-(3.11) and (3.32). However, because the OVETV algorithm

is implemented on a digital computer, quantization errors in the D/A and A/D converters become an issue. Consequently, the measured input u_{mk} and the measured output y_{mk} differ from the actual input into the system u_k and the actual output from the system y_k. They are related by the following equations

$$u_k = u_{mk} + u_{qk} \tag{3.35}$$

$$y_k = y_{mk} - y_{qk} \tag{3.36}$$

where u_{qk} is the quantization error introduced by the D/A converter and y_{qk} is the quantization error introduced by the A/D converter. The quantization errors are known to be bounded by

$$|u_{qk}| \leq \frac{Q_{D/A}}{2}, \tag{3.37}$$

$$|y_{qk}| \leq \frac{Q_{A/D}}{2}. \tag{3.38}$$

where the quantization level Q (for the A/D and/or D/A) is equal to $\frac{\text{FSAR}}{2^{nb}}$, FSAR is the full scale analog range of the converters, and nb is the number of bits. [31]

In general, equation (3.32) can be written as

$$y_k = -\sum_{i=1}^{n} a_{ik} y_{k-i} + \sum_{i=0}^{m} b_{ik} u_{k-i}. \tag{3.39}$$

After substituting equations (3.35) and (3.36) into (3.39), we get

$$y_{mk} = -\sum_{i=1}^{n} a_{ik} y_{m(k-i)} + \sum_{i=0}^{m} b_{ik} u_{m(k-i)} + y_{qk} + \sum_{i=1}^{n} a_{ik} y_{q(k-i)} + \sum_{i=0}^{m} b_{ik} u_{q(k-i)}. \tag{3.40}$$

If we let $v_k = y_{qk} + \sum_{i=1}^{n} a_{ik} y_{q(k-i)} + \sum_{i=0}^{m} b_{ik} u_{q(k-i)}$, equation (3.40) can be rewritten as

$$y_{mk} = -\sum_{i=1}^{n} a_{ik} y_{m(k-i)} + \sum_{i=0}^{m} b_{ik} u_{m(k-i)} + v_k, \tag{3.41}$$

which is precisely the ARX structure of (2.1).

To apply the OVETV algorithm, we need a bound on v_k. It is easy to show that

$$|v_k| \leq \frac{Q_{A/D}}{2}(1 + \sum_{i=1}^{n} |a_{ik}|) + \frac{Q_{D/A}}{2} \sum_{i=0}^{m} |b_{ik}|. \tag{3.42}$$

Consequently, if we have *a priori* bounds on the parameter values (which we have already assumed), then it is possible to derive a bound on v_k. It is clear, though, that the tighter these bounds are, the better the OVETV algorithm will perform.

We can now apply OVETV to the linear time-varying circuit. We will consider the case where $r(t) = (1.0 + 0.5\cos(t/10.0))\Omega$, $l(t) = 1.0\text{H}$, and $c(t) = 1.0\text{F}$. The sampling time T is selected to be 0.1 seconds. Both the A/D and D/A converter are assumed to have 12 bit resolution and a FSAR of 10 volts.

The OVETV algorithm is initialized as follows. Using (3.42), the bounds on v_k are set at $\overline{\gamma}_k = -\underline{\gamma}_k = 0.005$. The matrix, R_k, which bounds the parameter variations is set at $R_k = \zeta_k^2 I$ with $\zeta_k = 0.001$. The orientation matrix, $\tilde{P}_0$, is set at $10I$, and the ellipsoid center, $\tilde{\theta}_0$, is set at the origin. For this simulation, u_{mk}, the input supplied by the digital computer, is selected as a uniformly distributed random variable between $[-5.0, 5.0]$ volts.

Results of the simulation are shown in Figure 13 where the solid lines are the true parameters calculated from (3.32) and the dashed lines are the center estimates obtained from the OVETV algorithm. As discussed earlier, with this approach we are more interested in the parameter sets than the center estimates. That said, we would like to know how the center estimates of the OVETV algorithm compare with the point estimates of a traditional parameter estimation technique. The same

Table 2: Performance measures for OVETV-WRLS comparison

	WRLS	OVETV (Center Estimate)
ε_1	0.0851	0.0618
ε_2	0.0643	0.0471

simulation was run using weighted recursive least squares (WRLS). The WRLS was tuned empirically to minimize the following performance measures:

$$\varepsilon_i = \frac{1}{N}\sum_{k=1}^{N} \|\tilde{\theta}_k - \theta_k\|_i \qquad i = 1, 2 \tag{3.43}$$

where N is the number of samples, $\tilde{\theta}_k$ is the parameter estimate at time k, θ_k is the true parameter value at time k, and $\|\cdot\|_i$ is a vector norm. The performance measures resulting from a simulation with a forgetting factor of 0.73 and an initial covariance matrix of 10^6 are shown in Table 2. The performance measures for the center estimates of the OVETV algorithm are also shown in Table 2. The center estimates for the OVETV algorithm had smaller performance measures than WRLS for both $i = 1$ and $i = 2$.

Despite the good performance of the center estimates, we are most interested in the parameter sets. The parameter bound for b_1 and b_2 are shown in Figures 14 and 15, respectively. The bounds for b_1 are especially tight. While the bounds on a_1 and a_2 are not as tight, the volumes of the ellipsoids, as shown in Figure 16, are quite small.

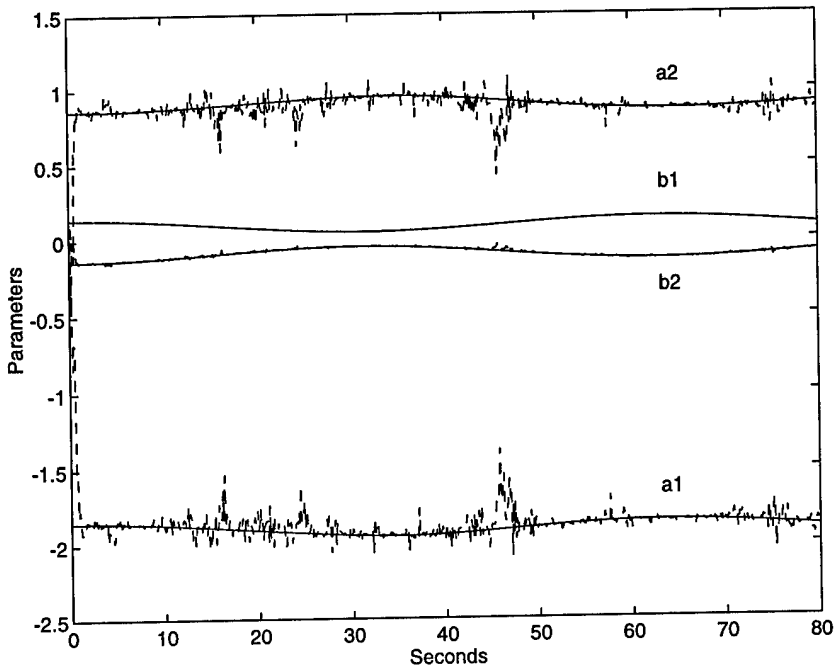

Figure 13: Parameter center estimates for Linear Circuit example

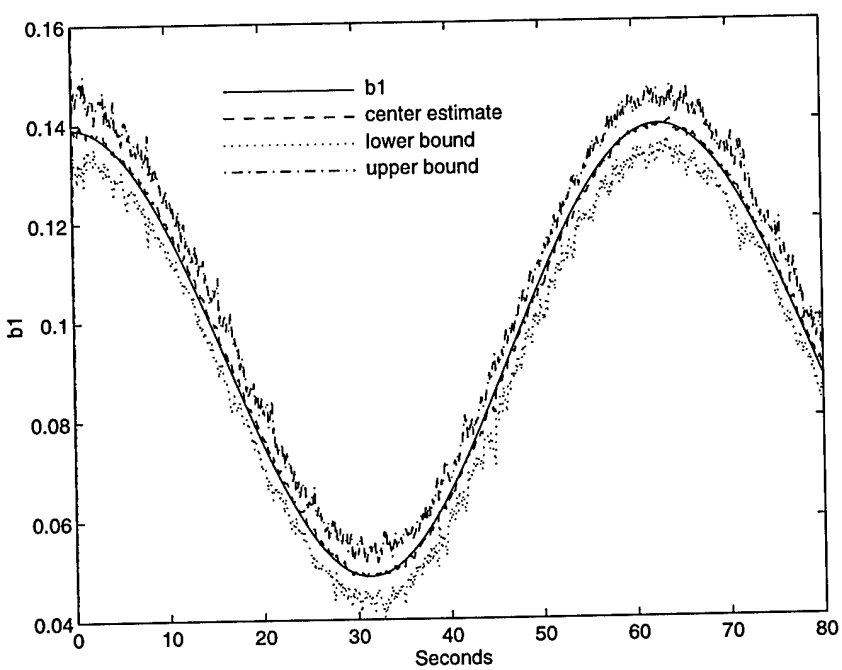

Figure 14: Parameter b_1 of Linear Circuit example

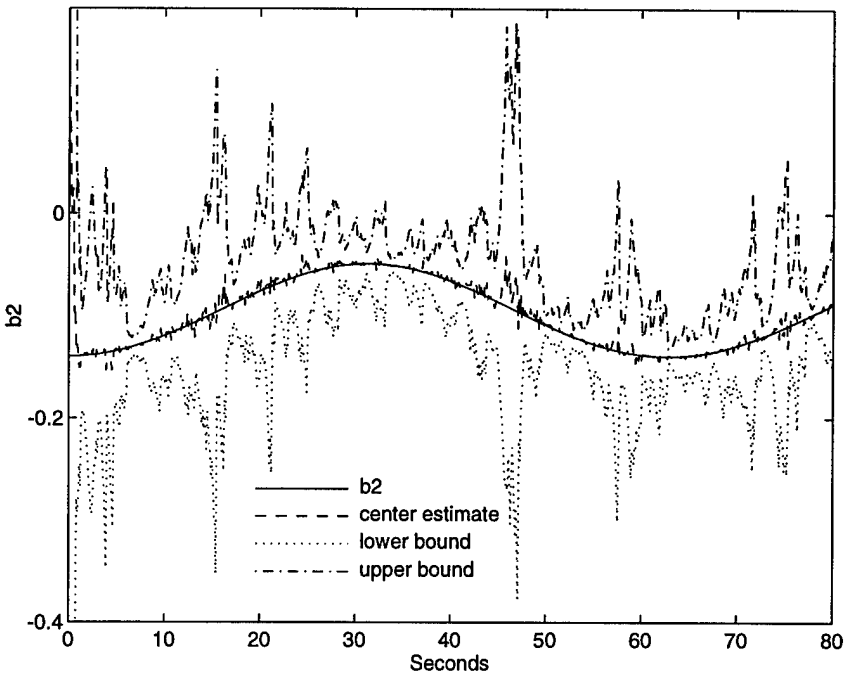

Figure 15: Parameter b_2 of Linear Circuit example

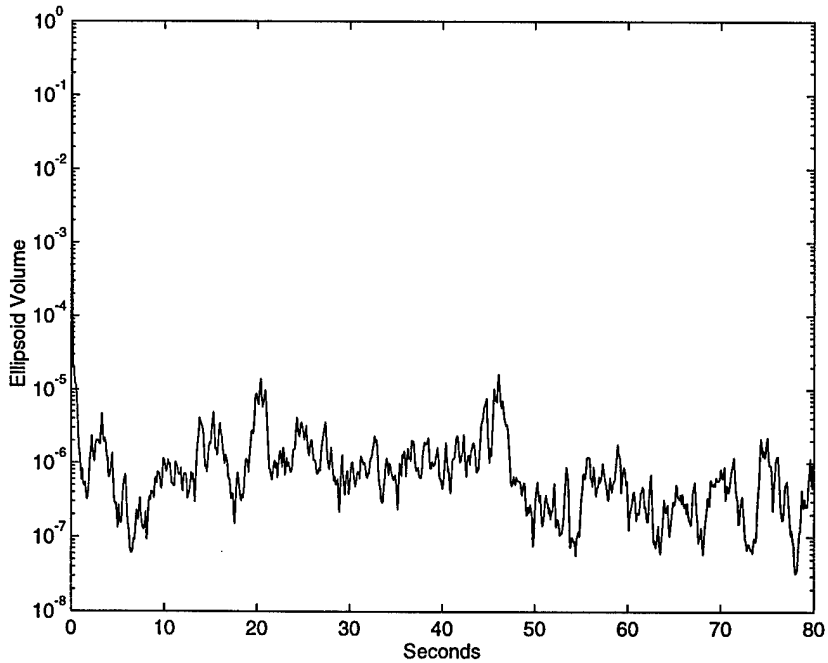

Figure 16: Ellipsoid volume for Linear Circuit example

3.4 Crude Oil Distillation Column

As a final example, we consider a crude oil distillation column. Our identification exercise for this example will be conducted on actual data recorded from an Amoco refinery in south Texas. The goal of the distillation process is to separate a mixture of two or more substances into its various components with a desired degree of purity. Because different pure liquids exhibit different volatilities, the application and removal of heat can be used to separate the components [33].

While the distillation process is highly nonlinear, within a given operating range it can effectively be modeled as a linear system. However, because of factors such as changes in raw materials or production levels or equipment fouling, the parameters of the linear model tend to vary with time [33]. Consequently, for control purposes it would be advantageous to track these parameters as they change with time. This information could be then be utilized to retune the process controllers automatically, rather than manually, as is often done in practice.

Several data sets were collected under open loop conditions on the distillation column shown in Figure 17. The results from one test are shown in Figure 18. The types of data measured are given in Table 3 where the y's are outputs, the u's are inputs, and $d1$ is a disturbance. This is clearly a multi-input, multi-output (MIMO) system.

In this section, we will look at the SISO relationship between the heat which is being applied at the base of the column, $u2$, and the top section temperature difference, $y2$. By controlling the temperature at specific locations along the column,

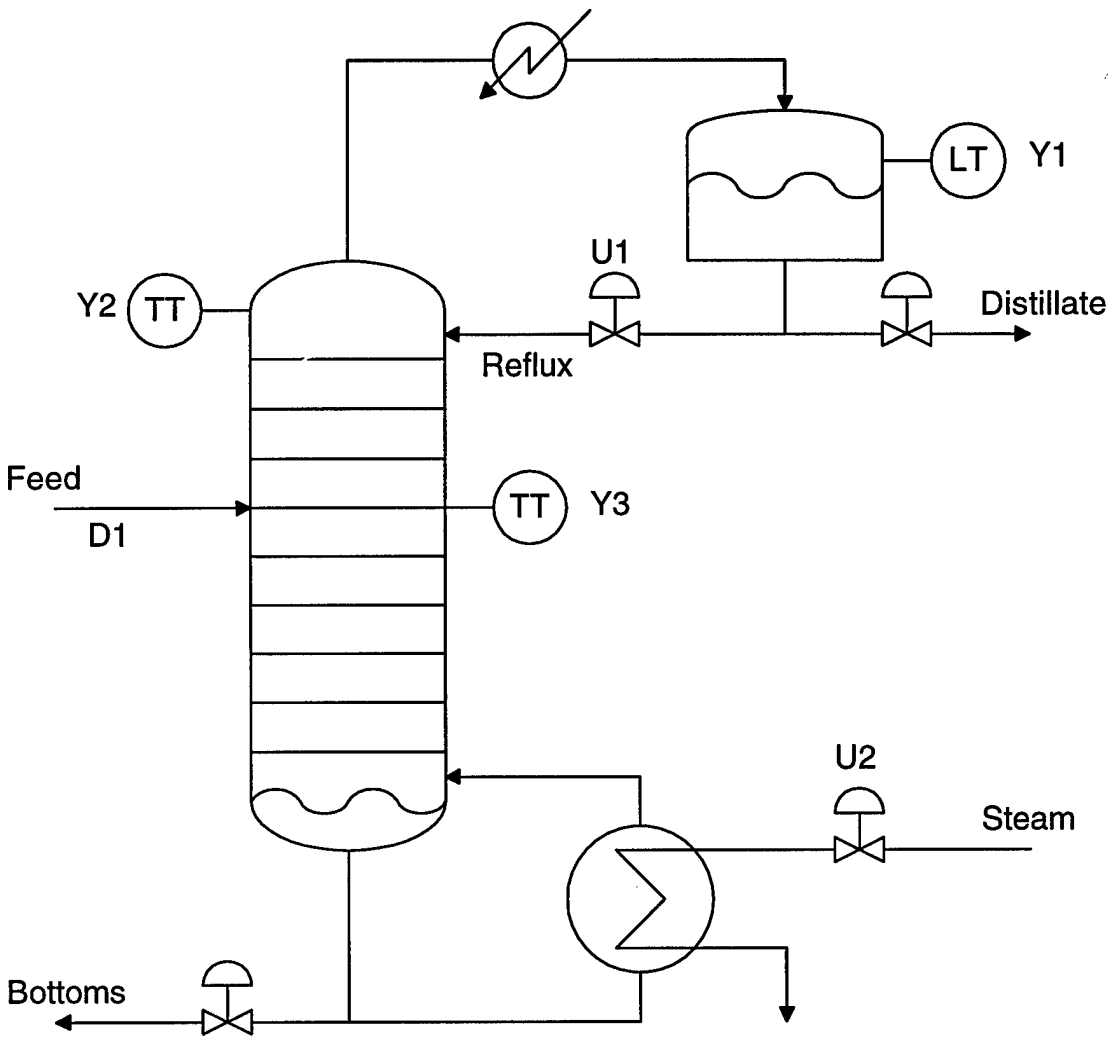

Figure 17: Distillation Column

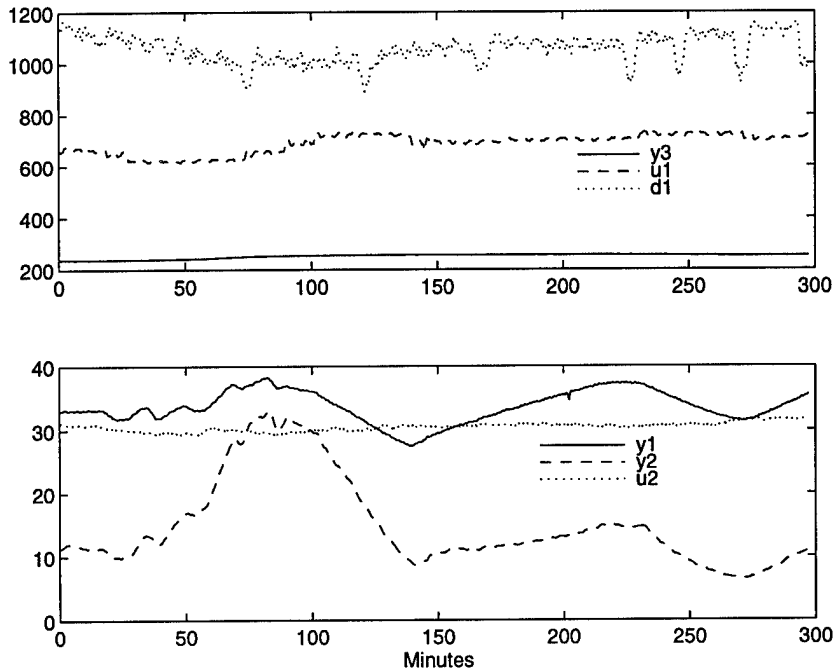

Figure 18: Open loop data for Distillation Column example

Table 3: Distillation column data

$y1$	overhead drum level
$y2$	top section temperature difference (degrees F)
$y3$	mid section temperature
$u1$	reflux
$u2$	heat addition to base of tower
$d1$	feed rate to tower

we are able to control the purity of the product compositions. In the next chapter, we will look at a multi-input, single-output (MISO) relationship for this example.

Based on knowledge of the system operation, and on control needs, the following model structure is assumed

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} + b_2 u_{k-2} + b_3 u_{k-3}, \tag{3.44}$$

where the sampling time is 20 seconds. The OVETV algorithm is initialized as follows: $\overline{\gamma}_k = -\underline{\gamma}_k = 0.25$, $R_k = \zeta_k^2 I$ with $\zeta_k = 0.00001$, $\tilde{P}_0 = 10I$, and $\tilde{\theta}_0 = 0$.

Applying the OVETV algorithm to this data, we obtain the center estimates shown in Figure 19. While we cannot compare the center estimates to the "true" parameters (which are effectively unknown), we can observe how well the center estimates predict the output at each time step. In Figure 20, we see that the predicted output $\tilde{y}_k = \tilde{\theta}_k^T \phi_k$ and the output y_k are almost indistinguishable. The volumes of the ellipsoids are shown in Figure 21 and the parameter bounds at 298 minutes are shown in Figure 22.

As a final note, for the bounds specified above on the disturbance vector, $\overline{\gamma}_k$ and $\underline{\gamma}_k$, the time-invariant OVE algorithm was inconsistent and failed.

3.5 Summary

In this chapter, we examined three examples. The first example was a first-order system which served to illustrate the key features of the OVETV algorithm. For this example, two other strategies, scalar addition and scalar multiplication, were shown to be effective alternatives to OVETV.

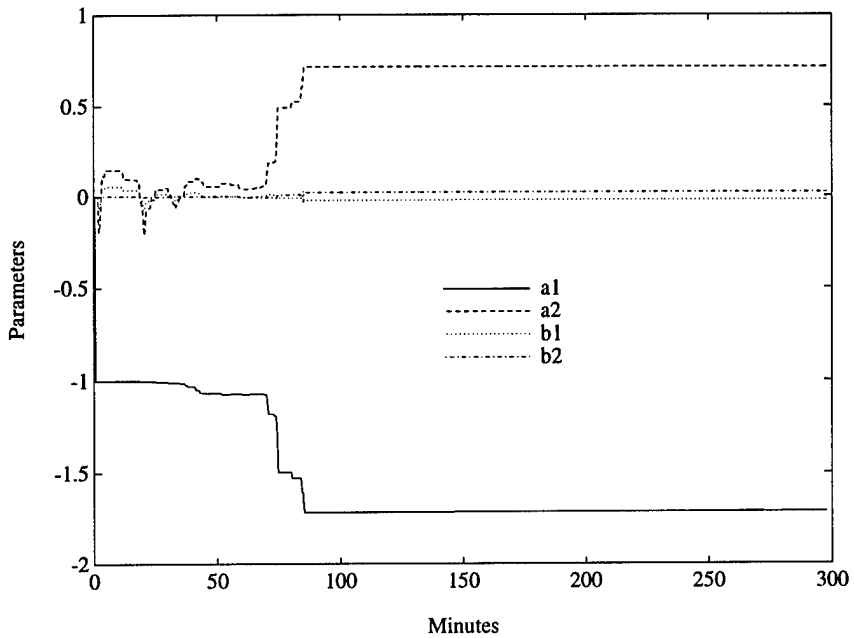

Figure 19: Center estimates for Distillation Column example

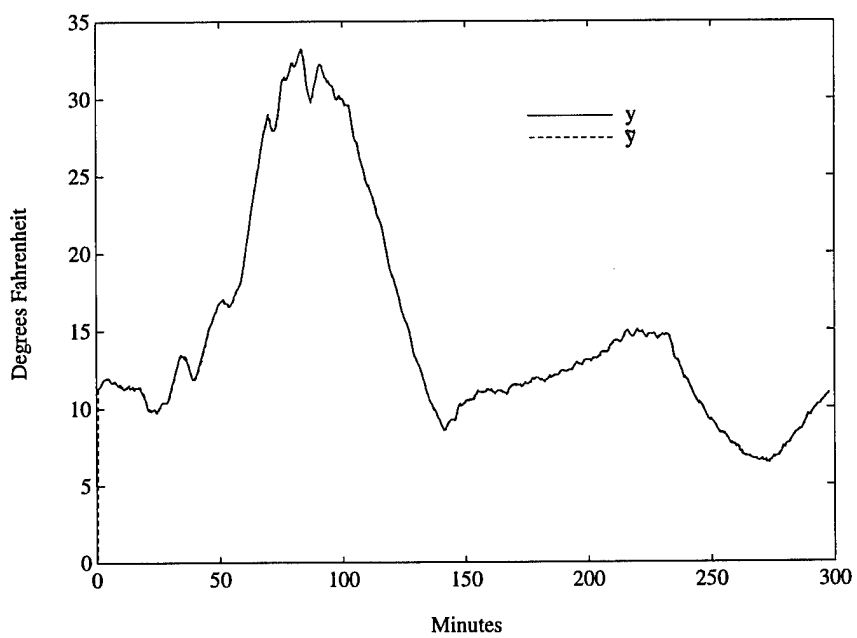

Figure 20: Top section temperature difference, Distillation Column example

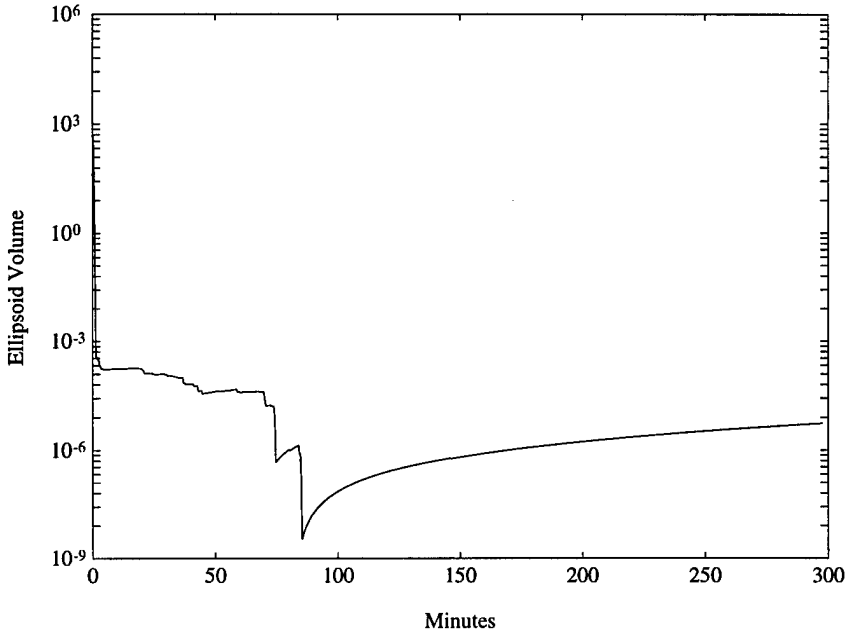

Figure 21: Ellipsoid volume for Distillation Column example

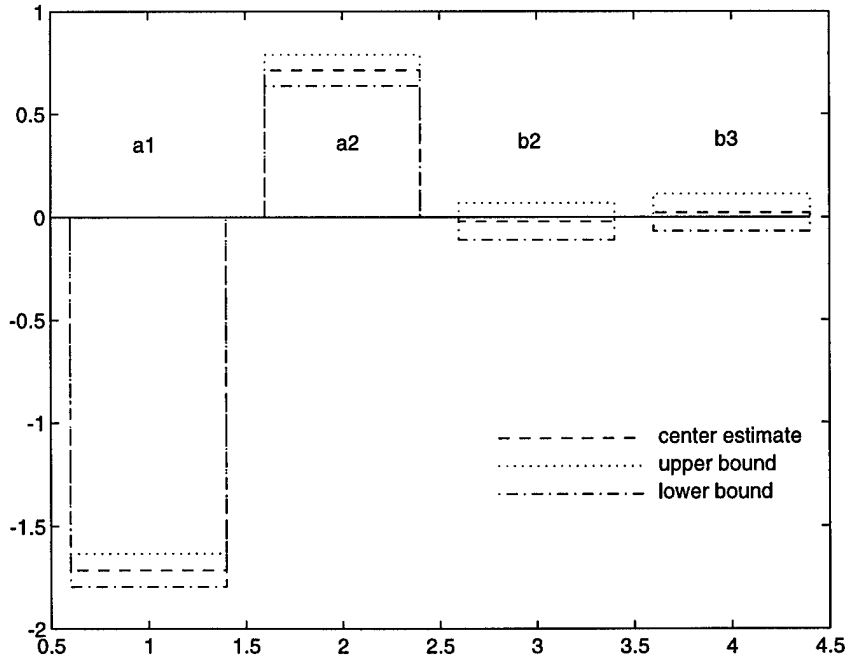

Figure 22: Parameter bounds at 298 minutes for Distillation Column example

In the second example, we explored the effects of sampling and quantization on a linear time-varying system, a linear time-varying circuit. Theorem 5 was given for transforming a dicrete-time linear time-varying state space equation to a time-varying difference equation. We were able to successfully identify the system using the OVETV algorithm. In fact, while the goal of the OVETV algorithm is to identify a parameter set, the center estimate of the OVETV algorithm was able to track the "true" parameter vector better than the "best" estimate using WRLS.

In the final example, we applied the OVETV algorithm to actual data obtained from a distillation column. Because the "true" parameter vector was effectively unknown, we were not able to compare it with the parameter sets and center estimates. However, the predicted output based on the center estimate matched the actual output very closely.

CHAPTER IV

Algorithm Extensions

4.1 Overview

As we saw in the last chapter, OVETV is a very effective tool for PSE of time-varying systems. However, as we began to apply OVETV, our experiences challenged us to extend and improve the properties of this algorithm. In this chapter, we discuss several of these modifications and adaptations to the OVETV algorithm.

As we applied OVETV to the crude oil distillation column of Section 3.4, we discovered that each of the outputs was actually affected by more than one input. Clearly, it would be good to identify this relationship directly in the OVETV framework. In Section 4.2, we discuss how the OVE and OVETV algorithms can be used to identify multiple-input, single-output (MISO) systems. We begin by showing how MISO linear time-invariant systems can be placed within the OVE framework. We also extend Theorem 5 of Section 3.3 to show how certain MISO linear time-varying systems can be placed within the MISO framework. Finally, we use the results of this section to identify a MISO relationship for the distillation column example.

Often, as was the case in the linear time-varying circuit example of Section 3.3,

the time-varying parameters that we are trying to identify are dependent on a smaller set of physical time-varying parameters. If we can exploit this dependence, we may be able to simplify the problem and reduce the number of computations required to implement the algorithm. Adaptations of the OVETV algorithm are developed in Section 4.3 to utilize knowledge of these dependencies.

When applying the OVETV algorithm in practice, the orientation matrix, $\tilde{P}_k$, can become non-positive definite. Theoretically this is impossible, geometrically it does not make since, and practically it causes the algorithm to fail. Numerical round-off errors are causing the algorithm to bomb. To solve this problem, we apply an approach that has been used successfully for stochastic estimation. A "square-root" algorithm is developed for implementing OVE. Furthermore, it is shown how the square-root approach can be extended to time-varying systems.

The type of input into the system has a very significant effect on the "quality" of estimates produced from an identification scheme. A "calculated" synthesis of the system input is explored in Section 4.5. In particular, we investigate an OVE-based input synthesis procedure (OVE-ISP) which was developed in [29] and [21]. We show how the scope of OVE-ISP can be extended to handle systems with known transportation lag such at the distillation column in Section 3.4. We also demonstrate its application to time-varying systems and an important "stabilizing" property that other synthesis strategies do not appear to possess.

4.2 Multiple-Input, Single-Output Systems

In many real systems, the output of the system is effected by more than one input. Consequently, we would like to extend the OVE and OVETV algorithms to handle MISO systems. In this section we will show how this can be done for linear time-invariant systems. Then we will show how this can be done for a class of linear time-varying systems. Finally, we will apply the MISO framework to the crude distillation column example which we explored in Section 3.4.

4.2.1 Linear Time-Invariant Case

Consider the linear time-invariant MISO state space equation

$$x((k+1)T) = Gx(kT) + \sum_{\kappa=1}^{\mu} H_\kappa u_\kappa(kT), \tag{4.1}$$

$$y(kT) = Dx(kT) + \sum_{\kappa=1}^{\mu} E_\kappa u_\kappa(kT), \tag{4.2}$$

where $x(kT) \in \Re^n$, $u_\kappa(kT) \in \Re$, $\{G, H_\kappa, D, E_\kappa\}$ are matrices of appropriate dimensions, and T is the sampling interval. Taking the Z transform of (4.1) and (4.2) gives

$$zX(z) = GX(z) + \sum_{\kappa=1}^{\mu} H_\kappa U_\kappa(z), \tag{4.3}$$

$$Y(z) = DX(z) + \sum_{\kappa=1}^{\mu} E_\kappa U_\kappa(z). \tag{4.4}$$

Solving (4.3) for $X(z)$, substituting it into (4.4), using a classical representation for the matrix inverse, and rearranging terms results in

$$\det(zI - G)Y(z) = \sum_{\kappa=1}^{\mu}\{D\mathrm{adj}(zI - G)H_\kappa + \det(zI - G)E_\kappa\}U_\kappa(z), \tag{4.5}$$

where $\det(\cdot)$ and $\mathrm{adj}(\cdot)$ return the determinant and adjoint of a matrix, respectively. If we multiply (4.5) by z^{-n}, take the inverse Z transform, and solve for $y(kT)$, it is easy to see that we get an equation of the form

$$y(kT) = -\sum_{i=1}^{n} a_i y((k-i)T) + \sum_{\kappa=1}^{\mu}\sum_{i=0}^{n} b_{i\kappa} u_\kappa((k-i)T). \tag{4.6}$$

Equation (4.6) can be rewritten as

$$y_k = \theta^T \phi_k \tag{4.7}$$

where

$$\theta = \begin{bmatrix} a_1 & \cdots & a_n & b_{01} & \cdots & b_{n1} & \cdots & \cdots & \cdots & b_{0\mu} & \cdots & b_{n\mu} \end{bmatrix}^T, \tag{4.8}$$

$$\begin{aligned} \phi_k = \Big[& -y(k-1) \ \cdots \ -y(k-n) \ u_0(k) \ \cdots \ u_0(k-n) \\ & \cdots \ \cdots \ \cdots \ u_\mu(k) \ \cdots \ u_\mu(k-n) \Big]^T. \end{aligned} \tag{4.9}$$

Assuming that the noise or uncertainty entering the system as $(y_k = \theta^T\phi_k + v_k)$ is bounded, the OVE algorithm can be directly applied to the system described by (4.7).

4.2.2 Linear Time-Varying Case

The goal of the dissertation has been to identify the parameters of time-varying systems. Clearly, we would like to be able to do this for MISO systems as well. To

see how MISO systems might fall into our time-varying ARX framework, we will extend Theorem 5 of Section 3.3 to MISO systems.

Theorem 6 *Given the linear time-varying MISO state space equation*

$$x((k+1)T) = G(kT)x(kT) + \sum_{\kappa=1}^{\mu} H_\kappa(kT)u_\kappa(kT), \tag{4.10}$$

$$y(kT) = D(kT)x(kT) + \sum_{\kappa=1}^{\mu} E_\kappa(kT)u_\kappa(kT), \tag{4.11}$$

where $x(kT) \in \Re^n$, $u_\kappa(kT) \in \Re$, and T is the sampling interval, define a sequence of $1 \times n$ matrices

$$\begin{aligned} L_0(kT) &= D(kT) \\ L_i(kT) &= L_{i-1}((k+1)T)G(kT) \qquad i = 1, 2, \cdots. \end{aligned} \tag{4.12}$$

If the rank of the matrix

$$\bar{L}(kT) = \begin{bmatrix} L_0(kT) \\ L_1(kT) \\ \vdots \\ L_{n-1}(kT) \end{bmatrix} \tag{4.13}$$

is n for all k, then $y(kT)$ satisfies a linear n^{th}-order difference equation of the form

$$y((k+n)T) + \sum_{i=1}^{n} \alpha_i(kT)y((k+n-i)T) = \sum_{\kappa=1}^{\mu}\sum_{i=0}^{n} \beta_{i\kappa}(kT)u_\kappa((k+n-i)T), \tag{4.14}$$

where

$$\begin{bmatrix} \alpha_n(kT) & \alpha_{n-1}(kT) & \cdots & \alpha_1(kT) \end{bmatrix} = -L_n(kT)\bar{L}^{-1}(kT), \tag{4.15}$$

$$\begin{bmatrix} \beta_{n\kappa}(kT) & \beta_{(n-1)\kappa}(kT) & \cdots & \beta_{0\kappa}(kT) \end{bmatrix} = \bar{w}_{n\kappa}(kT) - L_n(kT)\bar{L}^{-1}(kT)\bar{W}_\kappa(kT), \tag{4.16}$$

$$\bar{W}_\kappa(kT) = \begin{bmatrix} w_{00\kappa}(kT) & 0 & \cdots & 0 & 0 \\ w_{10\kappa}(kT) & w_{11\kappa}(kT) & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ w_{(n-1)0\kappa}(kT) & w_{(n-1)1\kappa}(kT) & \cdots & w_{(n-1)(n-1)\kappa}(kT) & 0 \end{bmatrix}, \quad (4.17)$$

$$\bar{w}_{n\kappa}(kT) = \begin{bmatrix} w_{n0\kappa}(kT) & w_{n1\kappa}(kT) & \cdots & w_{n(n-1)\kappa}(kT) & w_{nn\kappa}(kT) \end{bmatrix}, \quad (4.18)$$

and the $w_{ij\kappa}(kT)$ *are given recursively by*

$$w_{0j\kappa}(kT) = \begin{cases} E_\kappa(kT) & j = 0 \\ L_{(j-1)0}((k+1)T)H_\kappa(kT) & j = 1, 2, \cdots, \end{cases} \quad (4.19)$$

$$w_{ij\kappa}(kT) = w_{(i-1)(j-1)\kappa}((k+1)T) \quad i = 1, 2, \cdots, \quad j = 1, 2, \cdots, i, \quad (4.20)$$

and $\kappa = 1, 2, \cdots, \mu$.

Proof: The proof is a direct extension of the proof of Theorem 5 in Section 3.3 and is therefore omitted. Below we will show how this theorem can be used to transform a MISO system, which can be realized via linear time-varying state space equations, to a framework for which the OVETV algorithm is applicable. □

If we apply the inverse shift operator, $q^{-1}(\cdot)$, to (4.14) n times and solve for $y(kT)$, we get the following

$$y(kT) = -\sum_{i=1}^{n} a_i(kT)y((k-i)T) + \sum_{\kappa=1}^{\mu}\sum_{i=0}^{n} b_{i\kappa}(kT)u_\kappa((k-i)T), \quad (4.21)$$

where $a_i(kT) = \alpha_i((k-n)T)$ and $b_{i\kappa}(kT) = \beta_{i\kappa}((k-n)T)$. Equation (4.21) can be rewritten as

$$y_k = \theta_k^T \phi_k \quad (4.22)$$

where

$$\theta_k = \begin{bmatrix} a_1(kT) & \cdots & a_n(kT) & b_{01}(kT) & \cdots & b_{n1}(kT) \end{bmatrix}$$

$$\cdots \quad \cdots \quad \cdots \quad b_{0\mu}(kT) \quad \cdots \quad b_{n\mu}(kT) \Big]^T \tag{4.23}$$

$$\begin{aligned} \phi_k \;=\; \Big[& -y(k-1) \;\cdots\; -y(k-n) \;\; u_0(k) \;\cdots\; u_0(k-n) \\ & \cdots \quad \cdots \quad \cdots \quad u_\mu(k) \;\cdots\; u_\mu(k-n) \Big]^T . \end{aligned} \tag{4.24}$$

If the noise or uncertainty entering the system as ($y_k = \theta_k^T \phi_k + v_k$) is bounded, and if we can find a bound on how much the parameters change during any time step, then the OVETV algorithm can be directly applied to the system described by (4.22).

Remark 1 Before giving a MISO example, this seems to be an appropriate place to discuss the more general multiple-input, multiple-output (MIMO) case. First, Theorem 6 can be extended to the MIMO case by changing $\bar{L}^{-1}(kT)$ in the definition of α_i and $\beta_{i\kappa}$ to $\bar{L}^{+}(kT)$, where $(\cdot)^+$ is the matrix pseudoinverse as defined in [32]. The system can then be transformed into the *full polynomial form* as defined in [34]. In the MIMO case, the system disturbance, v_k, is a vector. If bounds are known on each of the components of v_k, then the problem can be composed into a separate problem for each output for which the MISO results discussed above directly apply [5]. When a bound exists on the norm of v_k, rather than on each of the components, the problem becomes more difficult. For extensions of the OBE algorithm to the case where a norm bounds exists on v_k, see [5].

4.2.3 Crude Oil Distillation Column Example

In Section 3.4, we estimated the parameters of the difference equation relating the heat which is being applied at the base of the distillation column, $u2$, to the top section temperature difference, $y2$. In actuality, the top section temperature difference is also affected by the reflux, $u1$, and the (disturbance) feed rate to the tower, $d1$. Clearly, we are interested in the MISO relationship between the output, $y2$, and the inputs, $u1$, $u2$, and $d1$.

Based on knowledge of the system operation, and on control needs, the following model structure is assumed:

$$\begin{aligned} y2(kT) &= -\sum_{i=1}^{4} a_i(kT)y2((k-i)T) + \sum_{i=2}^{5} b_{i1}(kT)u1((k-i)T) \\ &\quad + \sum_{i=2}^{5} b_{i2}(kT)u2((k-i)T) + \sum_{i=2}^{5} b_{i3}(kT)d1((k-i)T) \end{aligned} \tag{4.25}$$

where the sampling time is 20 seconds. The OVETV algorithm is initialized as follows: $\overline{\gamma}_k = -\underline{\gamma}_k = 0.15$, $R_k = \zeta_k^2 I$ with $\zeta_k = 0.0000006$, $\tilde{P}_0 = 10I$, and $\tilde{\theta}_0 = 0$.

Applying the OVETV algorithm to the data shown in Section 3.4, we get the results shown in Figure 23 for two parameters, a_1 and b_{22}. The center estimates and parameter bounds for all the parameters at 298 minutes are shown in Figure 24. As in the SISO case, we cannot compare the estimated parameter sets with the "true" parameters because they are effectively unknown. Also, as in the SISO case, we can observe how well the center estimates predict the output at each time step. As would be expected, the predicted output, $\tilde{y}_k = \tilde{\theta}_k^T \phi_k$, when we consider multiple inputs, matches the actual output, y_k, better than the predicted output when we

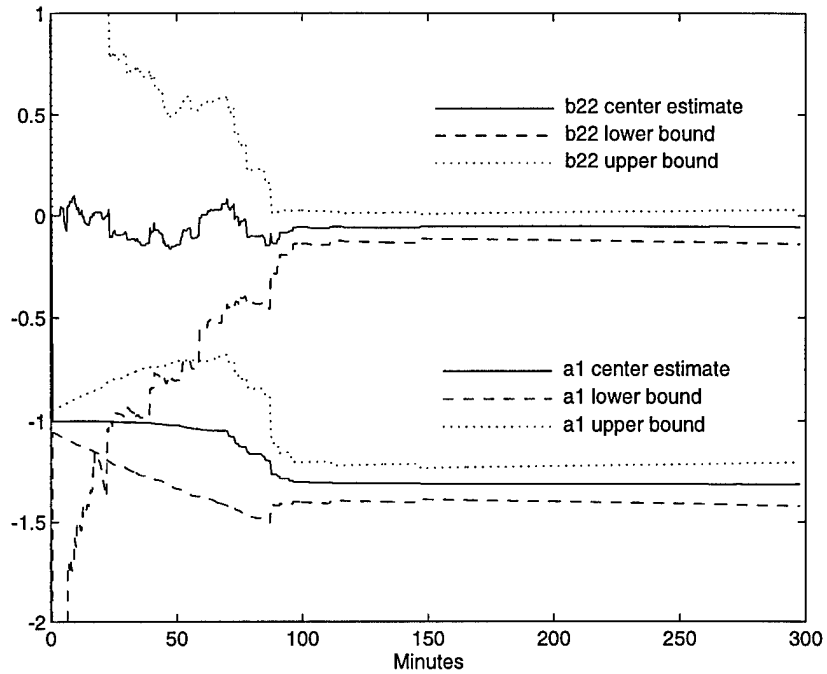

Figure 23: Parameters a_1 and b_{22} of MISO Distillation Column example

only consider a single input. In fact, the absolute error, $|y_k - \tilde{y}_k|$, is bounded by $\overline{\gamma}_k = -\underline{\gamma}_k = 0.15$ for the MISO case. The volumes of the ellipsoids are shown in Figure 25.

4.3 Dependent Parameter Variations

In this section, we will present an algorithm which exploits information regarding dependency in parameter variations to reduce the computational complexity of the OVETV algorithm. In the problem statement given in Chapter II, it was assumed that R_k, the matrix bound on the change in the parameter vector during each time

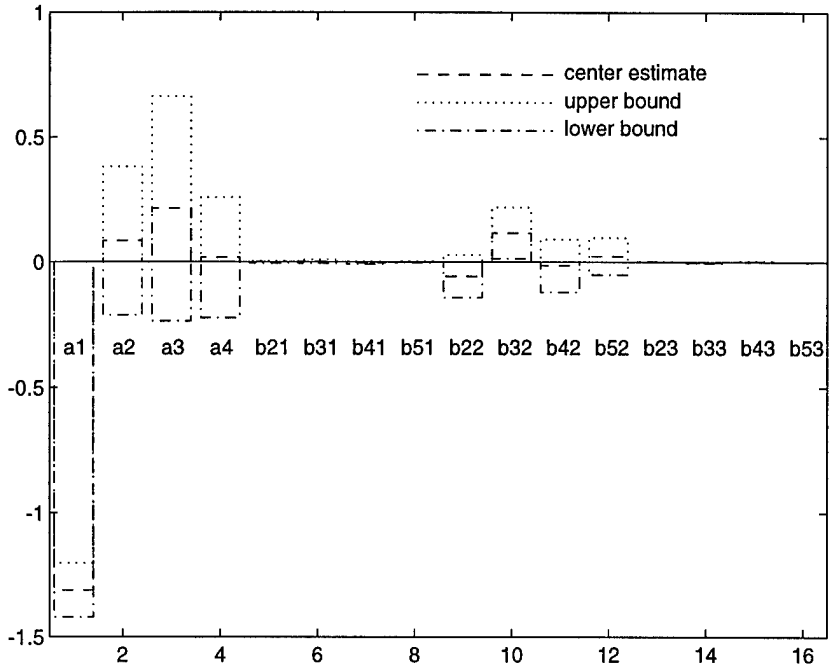

Figure 24: Parameter bounds at 298 minutes for MISO Distillation Column example

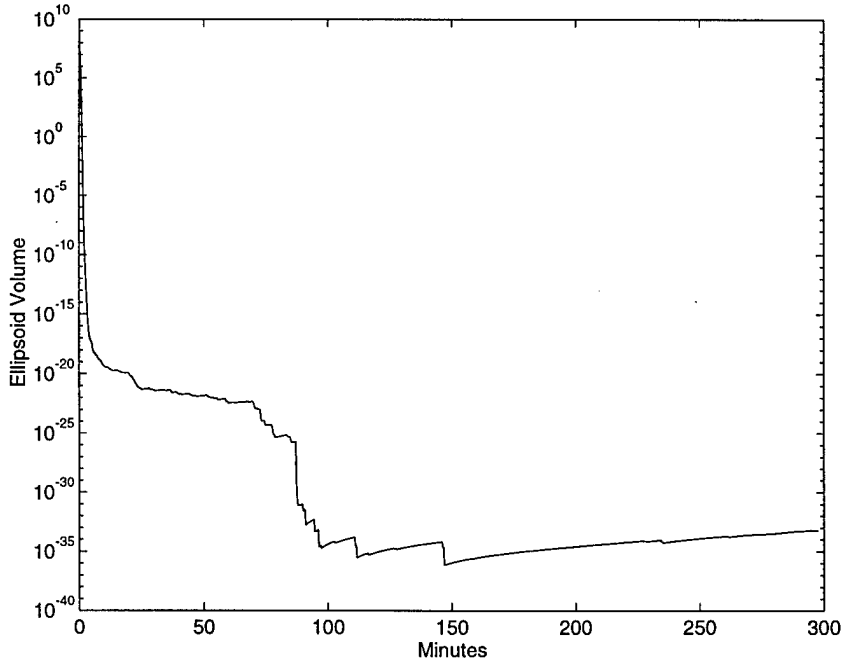

Figure 25: Ellipsoid volume for MISO Distillation Column example

step, was positive definite. While the theory of [15] still holds if $R_k \in \Re^{r\times r}$ is positive semi-definite, the bound on w_k must be defined in terms of its support function [26] instead of (2.4), that is,

$$\{w_k \in \Re^r : \eta_r^T w_k \leq \sqrt{\eta_r^T R_k \eta_r}; \eta_r \in \Re^r\} \tag{4.26}$$

The case where R_k is positive semi-definite is important because often there are time-invariant parameters, or the variation in one parameter is dependent on variations in other parameters.

Furthermore, if it is known that the rank of R_k is always less than r, we can use this information to reduce the number of computations required for the time update equations. If the rank of R_k is always $s < r$, then (2.2) can be rewritten as

$$\theta_{k+1} = \theta_k + B_k \check{w}_k, \tag{4.27}$$

where $\check{w}_k \in \Re^s$ is assumed to satisfy

$$\check{w}_k^T \check{R}_k^{-1} \check{w}_k \leq 1, \tag{4.28}$$

and the matrix, $\check{R}_k \in \Re^{s\times s}$, is symmetric, positive definite and known at each time k.

The bound in (4.28) can also be rewritten in terms of its support function as

$$\{\check{w}_k \in \Re^s : \eta_s^T \check{w}_k \leq \sqrt{\eta_s^T \check{R}_k \eta_s}; \eta_s \in \Re^s\}. \tag{4.29}$$

Using $w_k = B_k \check{w}_k$, the bound in (4.29), and the definition of a support function, the bound in (4.26) is given by

$$\{w_k \in \Re^r : \eta_r^T w_k \leq \sqrt{\eta_r^T B_k \check{R}_k B_k^T \eta_r}; \eta_r \in \Re^r\}. \tag{4.30}$$

Consequently, the generalized eigenvalue problem in (2.79) can be rewritten

$$B_{k-1}\check{R}_{k-1}B_{k-1}^T x_j = \lambda_j \tilde{P}_{k-1} x_j. \tag{4.31}$$

Solving (4.31) for each λ_j, $j \in [1, r]$ is equivalent to solving

$$\det(\lambda \tilde{P}_{k-1} - B_{k-1}\check{R}_{k-1}B_{k-1}^T) = 0 \tag{4.32}$$

for λ. Using properties of the determinant [35], it is easy to show that

$$\det(\lambda \tilde{P}_{k-1} - B_{k-1}\check{R}_{k-1}B_{k-1}^T) = \det(\check{R}_{k-1})\det(\tilde{P}_{k-1})\lambda^{r-s}\det(\lambda \check{R}_{k-1}^{-1} - B_{k-1}^T \tilde{P}_{k-1}^{-1} B_{k-1}). \tag{4.33}$$

Using (4.31) and (4.33), the revised time update equations are given in the following procedure:

1. Solve the generalized eigenvalue problem

$$B_{k-1}^T \tilde{P}_{k-1}^{-1} B_{k-1} x_j = \lambda_j \check{R}_{k-1}^{-1} x_j. \tag{4.34}$$

   for each λ_j, $j \in [1, s]$.

2. Solve

$$\sum_{j=1}^{s} \frac{1}{\lambda_j + p} = \frac{(s-r)p + s}{p(p+1)} \tag{4.35}$$

   for the unique $p > 0$.

3. Then, the solution to (2.11) is given by

$$\hat{\theta}_k = \tilde{\theta}_{k-1} \tag{4.36}$$

$$\hat{P}_k = (p+1)\tilde{P}_{k-1} + (p^{-1}+1)B_k \check{R}_{k-1} B_k^T \tag{4.37}$$

Note that the generalized eigenvalue problem in step 1 has been reduced from rth order to sth order. Furthermore, the $(r+1)$ order polynomial in step 2 has been replaced with a $(s+1)$ order polynomial. If $s \ll r$, the reduction in number of computations will be substantial. Note that the inverse of $\check{R}_{k-1}$ can often be calculated *a priori*. For the particular case when $s = 1$, see [15].

4.4 Square-Root Implementation

One problem when implementing many recursive estimation algorithms is that the covariance matrix in stochastic algorithms, such as the Kalman filter and recursive least squares, and the ellipsoid matrix in bounded input algorithms, such as OVE and OVETV, tend to loose their positive definite (PD) property due to finite precision calculations. This may cause the algorithms to "fail." One solution to this problem is to factor the PD matrix as $P = SS^T$ and then update S instead of P; this would guarantee the PD property of P. There have been many square-root methods developed for updating the covariance matrix of stochastic algorithms. For example, see [36], [37], [38]. While some of the algorithms are easily amenable to the OVE algorithm, others are not.

In this section, we will show how Potter's implementation of the Kalman filter [36] can be modified to work with the OVE algorithm. We will also compare the number of floating point operations using the direct implementation of the OVE algorithm with those using the new square-root algorithm. Finally, we will show how the square-root algorithm can be extended to the time-varying case.

4.4.1 Linear Time-Invariant Case

Recall from Section 2.3 that the measurement update equations for the center estimate, $\tilde{\theta}_k$, and the ellipsoid matrix, $\tilde{P}_k$, are given by

$$\tilde{\theta}_k = \hat{\theta}_k + \frac{\tau_k \hat{P}_k \phi_k}{(\phi_k^T \hat{P}_k \phi_k)^{1/2}} \tag{4.38}$$

$$\tilde{P}_k = \delta_k \hat{P}_k + (\sigma_k - \delta_k) \frac{\hat{P}_k \phi_k \phi_k^T \hat{P}_k}{\phi_k^T \hat{P}_k \phi_k}, \tag{4.39}$$

where τ_k, δ_k, and σ_k are scalars defined in Section 2.3 and ϕ_k is the regression vector. For the time-invariant case $\hat{\theta}_k = \tilde{\theta}_{k-1}$ and $\hat{P}_k = \tilde{P}_{k-1}$.

Following an outline similar to the development of Potter's algorithm for the Kalman filter in [36], we will develop a square-root algorithm for the OVE algorithm. We begin by setting

$$\tilde{P}_k = \tilde{S}_k \tilde{S}_k^T \tag{4.40}$$

$$\hat{P}_k = \hat{S}_k \hat{S}_k^T. \tag{4.41}$$

Substituting (4.40) and (4.41) into (4.39), we get

$$\begin{aligned} \tilde{P}_k &= \tilde{S}_k \tilde{S}_k^T = \delta_k \hat{S}_k \hat{S}_k^T + (\sigma_k - \delta_k) \hat{S}_k \hat{S}_k^T \phi_k \lambda_k \phi_k^T \hat{S}_k \hat{S}_k^T \\ &= \hat{S}_k (\delta_k I + (\sigma_k - \delta_k) \lambda_k \varpi_k \varpi_k^T) \hat{S}_k^T, \end{aligned} \tag{4.42}$$

where $\varpi_k = \hat{S}_k^T \phi_k$ and $\lambda_k = \frac{1}{\varpi_k^T \varpi_k}$.

From (4.42) it is easy to see that we can define

$$\tilde{S}_k = \hat{S}_k (\delta_k I + (\sigma_k - \delta_k) \lambda_k \varpi_k \varpi_k^T)^{1/2}$$

$$= \delta_k^{1/2}\hat{S}_k(I-(1-\frac{\sigma_k}{\delta_k})\lambda_k\varpi_k\varpi_k^T)^{1/2}, \tag{4.43}$$

where we know that δ_k is positive. Next, we use the fact that $(I-(1-\frac{\sigma_k}{\delta_k})\lambda_k\varpi_k\varpi_k^T)$ can be factored as

$$I-(1-\frac{\sigma_k}{\delta_k})\lambda_k\varpi_k\varpi_k^T = (I-\beta_k\varpi_k\varpi_k^T)^2 \tag{4.44}$$

where β_k is the root of

$$\varpi_k^T\varpi_k\beta_k^2 - 2\beta_k + (1-\frac{\sigma_k}{\delta_k})\lambda_k = 0. \tag{4.45}$$

The roots of (4.45) are given by $\beta_k = (1\pm\sqrt{\frac{\sigma_k}{\delta_k}})\lambda_k$. Using the root $\beta_k = (1+\sqrt{\frac{\sigma_k}{\delta_k}})\lambda_k$ and the factorization in (4.44), (4.43) can be written as

$$\begin{aligned}\tilde{S}_k &= \delta_k^{1/2}\hat{S}_k(I-(1+\sqrt{\frac{\sigma_k}{\delta_k}})\lambda_k\varpi_k\varpi_k^T)) \\ &= \delta_k^{1/2}(\hat{S}_k-(1+\sqrt{\frac{\sigma_k}{\delta_k}})\lambda_k L_k\varpi_k^T)),\end{aligned} \tag{4.46}$$

where $L_k = \hat{S}_k\varpi_k$.

Equation (4.46) is key step in the square-root algorithm for the OVE measurement update. The complete algorithm (excluding the calculation of the scalars τ_k, δ_k and σ_k which can be found in Section 2.3) is shown in Table 4. Also shown in Table 4 is a tabulation of the number of floating point operation required for each step. For comparison purposes, Table 5 details the steps and corresponding floating point operations for a direct implementation of the OVE measurement update. In our floating point counts for the direct implementation, we take advantage of the fact that we only have to calculate the upper triangular part of $\tilde{P}_k$ because it is symmetric.

Table 4: Square-root algorithm and corresponding floating point operations

Steps	Floating Point Operations			
	$+$	$\times$	$\sqrt{\ }$	$\div$
$\varpi_k = \hat{S}_k^T \phi_k$	n^2	n^2		
$\lambda_k = \frac{1}{\varpi_k^T \varpi_k}$	n	n		1
$\lambda_k^{1/2} = \sqrt{\lambda_k}$			1	
$\varepsilon_k = \tau_k \lambda_k^{1/2}$		1		
$L_k = \hat{S}_k \varpi_k$	n^2	n^2		
$\tilde{\theta}_k = \hat{\theta}_k + L_k \varepsilon_k$	n	n		
$\delta_k^{1/2} = \sqrt{\delta_k}$			1	
$\nu_k = (\delta_k^{1/2} + \sqrt{\sigma_k})\lambda_k$	1	1	1	
$\tilde{S}_k = \delta_k^{1/2} \hat{S}_k - (\nu_k L_k)\varpi_k^T$	n^2	$2n^2 + n$		
Total	$3n^2 + 2n + 1$	$4n^2 + 3n + 2$	3	1

As can be seen in Table 4 and Table 5, the number of floating point operations is of the same order for each algorithm. The number of floating point operations using the square-root algorithm will be more than the number of floating point operations using the direct implementation. In most cases, however, this increase in number of computations will be less important than the increase in numerical stability which is gained by using the square-root algorithm.

Finally, we need to be concerned with how the algorithm is initialized. That is, how does one find $\tilde{S}_0$ assuming that $\tilde{P}_0$ is specified. Often, when little is known about the plant, $\tilde{P}_0$ is initialized to ξI where I is the identity matrix and ξ is a very large positive scalar. In this case, $\tilde{S}_0$ can simply be set to $\sqrt{\xi} I$. If, on the other hand, more is known about the system and $\tilde{P}_0$ has a more complicated structure, then $\tilde{P}_0$ can

Table 5: Direct implementation and corresponding floating point operations

Steps	Floating Point Operations			
	$+$	$\times$	$\sqrt{}$	$\div$
$\varpi_k = \hat{P}_k \phi_k$	n^2	n^2		
$\lambda_k = \frac{1}{\phi_k^T \varpi_k}$	n	n		1
$\lambda_k^{1/2} = \sqrt{\lambda_k}$			1	
$\tilde{\theta}_k = \hat{\theta}_k + \varpi_k(\tau_k \lambda_k^{1/2})$	n	$n+1$		
$L_k = \varpi_k(\sigma_k - \delta_k)\lambda_k$	1	$n+2$		
$\tilde{P}_k = \delta_k \hat{P}_k + L_k \varpi_k^T$	$\frac{1}{2}n^2 + \frac{1}{2}n$	$n^2 + n$		
Total	$\frac{3}{2}n^2 + \frac{5}{2}n + 1$	$2n^2 + 4n + 3$	1	1

easily be factored as $\tilde{S}_0 \tilde{S}_0^T$ using the Cholesky decomposition [32].

4.4.2 Linear Time-Varying Case

Because of the improved numerical stability of square-root algorithms, we would like to extend this concept to the time-varying case. Unfortunately, the optimal time update equations of the OVETV algorithm are not amenable to being rewritten in square-root form. However, the two alternative algorithms discussed in Section 2.6, scalar addition and scalar multiplication, are very amenable to being rewritten in square-root form.

Scalar Addition

Recall from Section 2.6 that the basic idea of scalar addition was to reduce the number of calculations by constraining the time update ellipsoid, $\hat{E}_k$, to be an ellipsoid with the same center and orientation as E_{k-1}, and with semi-axis lengths which are found by adding a scalar, η, to the semi-axis lengths of E_{k-1}. R_k is also constrained to be $\zeta_k^2 I$.

From Theorem 3, we see that this is equivalent to constraining $\hat{P}_k$ to be of the form

$$\hat{P}_k = \tilde{V}_{k-1}(\Lambda_{k-1}^{1/2} + \eta I)^2 \tilde{V}_{k-1}^T \tag{4.47}$$

where a SVD of $\tilde{P}_{k-1}$ is given by

$$\tilde{P}_{k-1} = \tilde{V}_{k-1}\Lambda_{k-1}\tilde{V}_{k-1}^T, \tag{4.48}$$

$\tilde{V}_{k-1}$ is orthogonal, and $\Lambda_{k-1} = \mathrm{diag}(\lambda_1 \ldots \lambda_r)$.

To show how scalar addition can be cast in a square-root framework we take a SVD of $\tilde{S}_{k-1}$,

$$\tilde{S}_{k-1} = V_{k-1}\Sigma_{k-1}U_{k-1}^T, \tag{4.49}$$

where V_{k-1} and U_{k-1} are orthogonal and $\Sigma_{k-1} = \mathrm{diag}(\sigma_1 \ldots \sigma_r)$. Substituting (4.49) into (4.40), we get

$$\begin{aligned} \tilde{P}_{k-1} &= \tilde{S}_{k-1}\tilde{S}_{k-1}^T \\ &= V_{k-1}\Sigma_{k-1}U_{k-1}^T U_{k-1}\Sigma_{k-1}V_{k-1}^T \\ &= V_{k-1}\Sigma_{k-1}^2 V_{k-1}^T. \end{aligned} \tag{4.50}$$

Comparing (4.48) and (4.50), we see that $\tilde{V}_{k-1}$ and Λ_{k-1} can be chosen such that $\tilde{V}_{k-1} = V_{k-1}$ and $\Lambda_{k-1}^{1/2} = \Sigma_{k-1}$.

Let

$$\hat{S}_k = V_{k-1}(\Sigma_{k-1} + \eta I)U_{k-1}^T. \tag{4.51}$$

Substituting (4.51) into (4.41), we get

$$\begin{aligned} \hat{P}_k &= \hat{S}_k \hat{S}_k^T \\ &= V_{k-1}(\Sigma_{k-1} + \eta I)U_{k-1}^T U_{k-1}(\Sigma_{k-1} + \eta I)V_{k-1}^T \\ &= V_{k-1}(\Sigma_{k-1} + \eta I)^2 V_{k-1}^T. \end{aligned} \tag{4.52}$$

Comparing (4.47) and (4.52), and using the fact that $\tilde{V}_{k-1} = V_{k-1}$ and $\Lambda_{k-1}^{1/2} = \Sigma_{k-1}$, we see that (4.51) provides us with the square-root update for scalar addition where the SVD of $\tilde{S}_{k-1}$ is given by (4.49).

However, we still must show that the value of η which minimizes the volume of $\hat{E}_k$ such that $\hat{E}_k \supset \hat{G}_k$ where $\hat{G}_k$ is given by (2.26) can be calculated directly from $\tilde{S}_{k-1}$. From Theorem 3, we know that optimum value for η is given by

$$\eta = p_6 + p_7 - \frac{2\underline{a}\overline{a}}{3(\overline{a} + \underline{a})}, \tag{4.53}$$

where

$$\overline{a} = \max_{j\in[1,r]} \sqrt{\lambda_j}, \qquad \underline{a} = \min_{j\in[1,r]} \sqrt{\lambda_j}, \tag{4.54}$$

and p_6 and p_7 are defined in Theorem 3 and are functions of $\overline{a}$, $\underline{a}$, and ζ_{k-1}. Therefore, if we can define $\underline{a}$ and $\overline{a}$ in terms of $\tilde{S}_{k-1}$ we have achieved our goal. In fact,

$$\overline{a} = \max_{j\in[1,r]} \sigma_j, \qquad \underline{a} = \min_{j\in[1,r]} \sigma_j, \tag{4.55}$$

where the σ_j are the singular values given by (4.49).

Scalar Multiplication

In the scalar multiplication strategy, $\hat{E}_k$ is constrained to be an ellipsoid with the same center and orientation as E_{k-1}; the semi-axis lengths are found by multiplying the semi-axes of E_{k-1} with a scalar, η. From Theorem 4, we see that this is equivalent to constraining $\hat{P}_k$ to be of the form

$$\hat{P}_k = \eta^2 \tilde{P}_{k-1}. \tag{4.56}$$

It is easy to see that this is equivalent to

$$\hat{S}_k = \eta \tilde{S}_{k-1}, \tag{4.57}$$

in our square-root framework.

We still must show that the value of η, which minimizes the volume of $\hat{E}_k$ such that $\hat{E}_k \supset \hat{G}_k$ where $\hat{G}_k$ is given by (2.10) (R_k is not constrained to $\zeta_k^2 I$), can be calculated directly from $\tilde{S}_{k-1}$. From Theorem 4, we know that optimum value for η is given by

$$\eta = 1 + \sqrt{\bar{\lambda}} \tag{4.58}$$

where $\bar{\lambda}$ is the maximum generalized eigenvalue, λ_j, which satisfies

$$R_{k-1} x_j = \lambda_j \tilde{P}_{k-1} x_j. \tag{4.59}$$

Since R_{k-1} is symmetric and positive semi-definite, it can be factored as $T_{k-1}T_{k-1}^T$. Therefore, (4.59) can be written as

$$T_{k-1}T_{k-1}^T x_j = \lambda_j \tilde{S}_{k-1}\tilde{S}_{k-1}^T x_j. \tag{4.60}$$

This problem can be solved directly from T_{k-1} and $\tilde{S}_{k-1}$ using the generalized singular value decomposition [32]. Given T_{k-1} and $\tilde{S}_{k-1}$, there exist orthogonal V_{k-1} and U_{k-1} and invertible X_{k-1} such that

$$V_{k-1}^T T_{k-1} X_{k-1} = C_{k-1} \tag{4.61}$$

$$U_{k-1}^T \tilde{S}_{k-1} X_{k-1} = D_{k-1}, \tag{4.62}$$

where $C_{k-1} = \text{diag}(c_1 \ldots c_r)$ and $D_{k-1} = \text{diag}(d_1 \ldots d_r)$. The generalized singular values are given by $\sigma_j = c_j/s_j$ where $j = 1 \ldots r$. The generalized singular values, σ_j, are also equal to the square-root of the generalized eigenvalues, λ_j. Therefore, the optimum value of η is given by

$$\eta = 1 + \bar{\sigma} \tag{4.63}$$

where $\bar{\sigma}$ is the maximum generalized singular value. For a discussion of how the generalized singular values can be calculated, see [32].

Finally, some discussion of how R_k can be factored as $T_k T_k^T$ is in order. Clearly if $R_k = \zeta_k^2 I$, then $T_k = \zeta_k I$. Actually, if $R_k = \zeta_k^2 I$, then this problem can be solved using the standard singular value decomposition. If on the other hand, R_k has a more complicated structure, then R_k can be factored using the Cholesky decomposition [32]. If R_k does not vary with time, then this can be done once before the identification process begins.

4.5 Input Synthesis

The input into a system during an identification experiment greatly affects the "quality" of the estimates produced by an identification algorithm. Often the experiment design has some freedom on how this input is chosen. We will call this design of the system input, which is intended to improve the identification experiment, "input synthesis."

In this section, we will consider an input synthesis strategy developed to work with the OVE algorithm [11]. The goal of this OVE-based input synthesis procedure is to drive the system such that the regression vector, ϕ_k, is aligned parallel to the ellipsoid axis of greatest length. This direction is chosen because it was found in [29] that, when using the OVE algorithm, the greatest reduction in volume occurs along the direction which is parallel to ϕ_k.

The OVE Input Synthesis Procedure (OVE-ISP) was originally developed in [29] for systems with a single delay, $l = 1$. The algorithm is based on a system model which is realized from the ellipsoid center estimate at time k_0, $\tilde{\theta}_{k_0}$, for which ϕ_k is the state vector. In [39], DeVilbiss and Yurkovich modified the algorithm to handle systems where the delay index, l, was either 0 or 1. For improved numerical robustness, they used a balanced minimal realization instead of the canonical realization discussed above. However, by using a minimal realization, the new strategy required an additional "observation" stage which was not needed with the canonical realization. In [11], DeVilbiss and Yurkovich returned to the canonical realization because

it was found that the input signal level during the "observation" stage was large in magnitude compared to the final stage of the input synthesis strategy.

This input synthesis strategy is investigated in this dissertation for several reasons. First, the authors of [11] and the author of this dissertation feel that one of the most significant applications of OVE-ISP may be for robust adaptive control of time-varying systems. The reasoning is as follows. As has been said, when applying PSE to robust control or robust adaptive control, there is a trade off between set size and system performance. In [39], the OVE-ISP algorithm demonstrated significantly greater ellipsoid volume reduction than a pseudorandom white noise sequence of equal expected energy during the "transient phase" of the identification experiment. However, in [11] the OVE-ISP algorithm did not demonstrate any volume reduction advantages over a pseudorandom white noise sequence of equal expected energy, or over an alternative input synthesis algorithm found in [40], when comparing steady-state performance.

Therefore, in the time-invariant case where steady-state performance is often more important than transient performance, other input synthesis strategies could be used. However, in the time-varying case, where the transient performance is very important, OVE-ISP could be used in conjunction with an adaptive robust controller. Initially, OVE-ISP could be used to quickly reduce the ellipsoid volume so that a robust controller could be designed. It could also be used periodically if the ellipsoid volume becomes to large for effective robust control.

Second, during our course of study of OVETV and OVE-ISP we discovered a

significant property of OVE-ISP which may make it the preferred input synthesis strategy for a certain class of linear systems which may be either time-invariant or time-varying. Because of the "feedback nature" of OVE-ISP, the OVE-ISP demonstrates a "stabilizing" property that the pseudorandom white noise or input synthesis strategy of [40] did not possess. Consequently, OVE-ISP is probably the preferred strategy for systems which may be unstable or marginally stable.

Third, OVE-ISP offers design flexibility which may make it a useful strategy not only for system identification but also for FDI (which is discussed in the next chapter). For an example of FDI-based input design, see [41] where Sadegh, Madsen, and Holst discuss an optimal input design strategy for fault detection and diagnosis based on stochastic properties of the system. In OVE-ISP, the goal is to direct the regression vector, ϕ_k, such that maximum reduction in ellipsoid volume occurs. For FDI an alternative goal might be to direct ϕ_k such that fast tracking and isolation take place after a fault is detected. It should be pointed out, however, that the OVE-ISP strategy is highly dependent on the ellipsoid center estimate, $\tilde{\theta}_{k0}$, which may not adequately represent the system dynamics after a fault occurs.

Finally, one of the most significant limitations of the OVE-ISP algorithm in [11] is its lack of capability to handle plants where the known system delay, l, is greater than 1. This should not be a problem for physical systems where the original continuous system can adequately be modeled with rational functions in the Laplace domain. However, it will not be very satisfactory for systems where there is a known transportation lag such that $l > 1$ or or $l \gg 1$. This include systems such as the crude oil

distillation column which was discussed in Sections 3.4 and 4.2.3. The development of OVE-ISP in this section will eliminate this restriction and improve the speed of the OVE-ISP for systems where $l = 0$.

We begin by developing the OVE-ISP algorithm for the general case, $l \geq 0$. Next, we show how the OVE-ISP algorithm can be used in conjunction with OVETV for PSE of time-varying systems. We compare the performance of OVETV using OVE-ISP with the performance of OVETV using pseudorandom white noise and OVETV using the input synthesis strategy of [40]. We compare this performance for systems which are both stable and unstable.

4.5.1 OVE-ISP Development for General Delay Case

As discussed earlier, the goal of OVE-ISP is to drive the system such that ϕ_k is aligned with the ellipsoid axis of greatest length. At time k_0, this axis is given by the eigenvector, $\pm\tilde{x}_{k_0}$, which corresponds to the largest magnitude eigenvalue of $\tilde{P}_{k_0}$ and has been normalized such that $\|\tilde{x}_k\|_2 = 1$. At some future time k_1, the desired regression vector, ϕ_d, is selected to be ($\phi_d = \pm\varrho\tilde{x}_{k_0}$) where ϱ is a scalar which is maximized subject to the constraints of the physical system, and the sign of $\tilde{x}_{k_0}$ is chosen to minimize the synthesized input energy for a given ϱ.

Invoking a certainty equivalence argument, the current ellipsoid center, $\tilde{\theta}_{k_0}$, is chosen to realize a state space model of the system which has the form

$$\phi_{k+1} = A\phi_k + Bu_{k-l+1} \tag{4.64}$$

where

$$A = \begin{bmatrix} -a_1 & \cdots & -a_{n-1} & -a_n & -b_l & \cdots & -b_{m-1} & -b_m \\ 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & & & & \vdots \\ \vdots & \ddots & 1 & \ddots & & & & \vdots \\ \vdots & & \ddots & 0 & \ddots & & & \vdots \\ \vdots & & & \ddots & 1 & \ddots & & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{4.65}$$

At this point in our design, we will assume that the center ellipsoid estimate, $\tilde{\theta}_{k_0}$, is equal to the "true" center estimate, $\hat{\theta}_k$, that the noise or uncertainty entering our system, v_k, is 0, and that the system is time-invariant. Most likely, all three of these assumptions will fail. However, the degree to which they hold will directly affect the ability of OVE-ISP to achieve it goal.

Also at issue is the observability and controllability of (4.64). By realizing (4.64) as we did, the states are directly measurable; therefore observability is not an issue. If the modeling orders are chosen correctly, but (4.64) is not controllable, it should be possible to choose another parameter estimate within $\tilde{E}_{k_0}$ such that (4.64) is controllable. For a more complete discussion of the issues involved in choosing this realization, see [29] or [21].

The system realization in (4.64) can now be used to generate the desired input sequence. Let $k_l = k_0 + l - 1$. From (4.64), we know that

$$\begin{aligned} \phi_{k_l+1} &= A\phi_{k_l} + Bu_{k_l-l+1} \\ &= A\phi_{k_l} + Bu_{k_0}. \end{aligned} \tag{4.66}$$

Iterating (4.66) and using back substitution, it is easy to show that

$$\phi_{k_l+r} = A^r \phi_{k_l} + A^{r-1} B u_{k_0} + \cdots + B u_{k_0+r-1}. \tag{4.67}$$

Equation (4.67) can now be rewritten as

$$\phi_{k_l+r} = A^r \phi_{k_l} + CU, \tag{4.68}$$

where $U = \begin{bmatrix} u_{k_0} & u_{k_0+1} & \cdots & u_{k_0+r-1} \end{bmatrix}^T$ is a vector which contains the next r inputs and $C = \begin{bmatrix} B & AB & \cdots & A^{r-1}B \end{bmatrix}$ is the controllability matrix which is assumed to be invertible. Thus, the desired input sequence is given by

$$U = C^{-1}(\phi_d - A^r \phi_{k_l}), \tag{4.69}$$

where ϕ_{k_l+r} has been set to our desired regression vector, ϕ_d.

Finally, we need to specify ϕ_{k_l}. It is easy to show that

$$\phi_{k_l} = \begin{cases} \phi_{k_0-1} & l = 0 \\ \phi_{k_0} & l = 1 \\ A^{l-1}\phi_{k_0} + A^{l-2}Bu_{k_0-l+1} + \cdots + Bu_{k_0-1} & l \geq 2, \end{cases} \tag{4.70}$$

where all the terms on the right hand side are known. The desired input sequence series takes r steps. The desired regressor is achieved at time $k_1 = k_l + r = k_0 + l - 1 + r$. This algorithm reduces to the algorithm discussed in [11] for the case when $l = 1$. It is faster for the case where $l = 0$. Finally, it handles the cases where $l \geq 2$ that are not handled by the development in [11].

The steps of the revised OVE-ISP algorithm can now be summarized:

1. Use the current OVE (OVETV) ellipsoid center estimate, $\tilde{\theta}_{k_0}$, to generate the state space realization of (4.64).

2. Find the eigenvector x_k, which corresponds to the largest eigenvalue of $\tilde{P}_{k_0}$. Set $\phi_d = \pm\varrho\tilde{x}_{k_0}$ where ϱ is prespecified.

3. Compute the input sequence specified by (4.69) and (4.70) which is necessary to transfer ϕ_{k_0} to ϕ_d.

Note that even through ϕ_k will not reach ϕ_d until $k_1 = k_0 + l - 1 + r$, the input sequence is only specified until (k_0+r-1). Therefore, if desired, a new sequence could be generated starting at $(k_0 + r)$. If fact, if a new sequence is to be generated every r steps, from (4.69) we see that the OVE-ISP procedure could be viewed (and analyzed) as a multi-rate time-varying feedback scheme. To illustrate this, see Figure 26 for the case where $l = 1$, and where $\{A_c, B_c, D_c\}$ represent the plant's continuous time system dynamics, T is the sampling time, and ZOH is a zero-order hold. If an OVE-ISP sequence begins at time k_0, for $k = k_0 \ldots k_{0+r-1}$, the time-varying gain K_k is given by $\begin{bmatrix} K_{k_0}^T & K_{k_0+1}^T & \cdots & K_{k_0+r-1}^T \end{bmatrix}^T = C^{-1}$.

Feedback can cause difficulties for system identification algorithms [34]. It is shown in [42] how *identifiability* can be lost due to static gain feedback. In this case, the parameters cannot be uniquely determined. However, *identifiability* can be regained by using a time-varying feedback gain or feedback of sufficiently high order. The feedback gain in the OVE-ISP procedure, K_k, is time-varying. Furthermore, the feedback depends not only on the output, but on past values of the output and input. While the full implications of this "feedback nature" will require further study, its importance will be demonstrated in the next section.

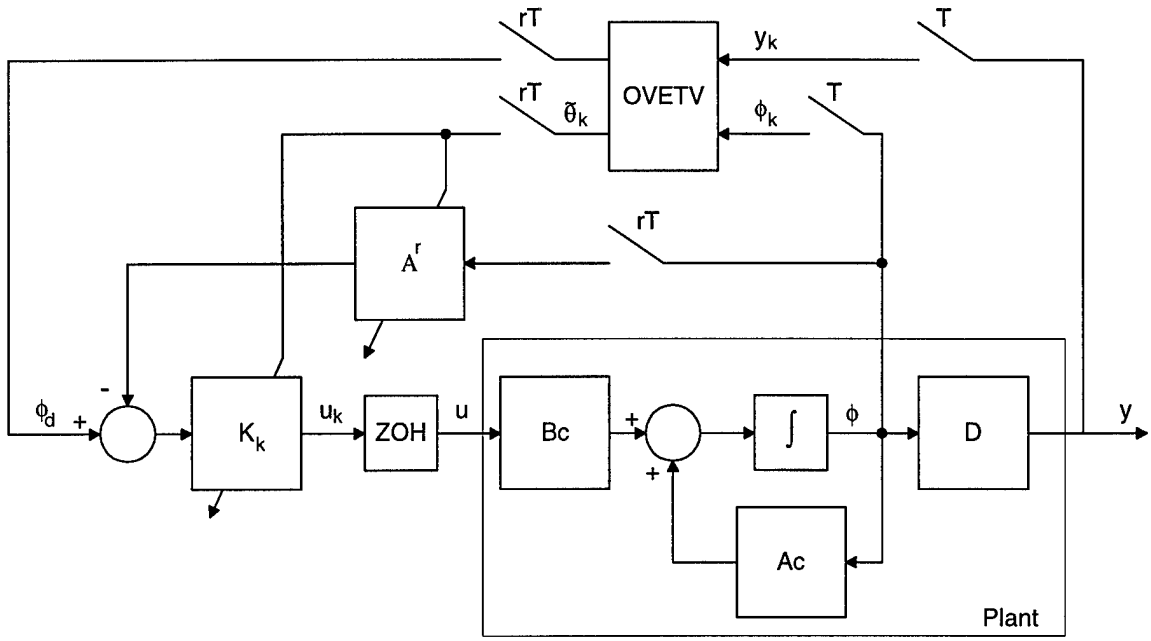

Figure 26: OVE-ISP procedure viewed as time-varying feedback scheme

4.5.2 Application of OVE-ISP and OVETV to Time-Varying Systems

As previously mentioned, one of the most significant applications for OVE-ISP may be in the areas of adaptive PSE and adaptive robust control. However, in the development of the previous section, it was explicitly assumed that the system we were trying to identify was time-invariant. Clearly, we would expect the performance of OVE-ISP to deteriorate for systems which vary "quickly" with time. Consequently, for demonstration purposes we will work only with "slowly-varying" systems.

In this section, we will consider a variation of the example examined in [40] and [11]; here, we will allow the system parameters to vary with time. Consider the

time-varying system

$$y_k = \theta_k^T \phi_k + v_k, \tag{4.71}$$

$$\theta_{k+1} = \theta_k + w_k, \tag{4.72}$$

where

$$\theta_k = \begin{bmatrix} a_{1k} & a_{2k} & b_{1k} \end{bmatrix}^T \tag{4.73}$$

$$\phi_k = \begin{bmatrix} -y_{k-1} & -y_{k-2} & u_{k-1} \end{bmatrix}^T, \tag{4.74}$$

and

$$\theta_0 = \begin{bmatrix} 0.4 & 0.85 & 0.75 \end{bmatrix}^T \tag{4.75}$$

$$w_k = \begin{bmatrix} 0.005 & -0.002 & 0.007 \end{bmatrix}^T, \tag{4.76}$$

and v_k is a uniformly distributed random variable between [-0.05 0.05]. The system is initially at rest.

To apply OVETV, the bounds in (2.3) and (2.4) must be specified. We set $\overline{\gamma}_k = -\underline{\gamma}_k = 0.05$ and $R_k = \zeta_k^2 I$ where $\zeta_k = \|w_k\|_2$ and I is the identity matrix. The orientation matrix $\tilde{P}_0$ is set to $10^4 I$, and the center $\tilde{\theta}_0$ is set to the origin.

For this example, we will use three different input sequences. OVE-ISP with $\varrho = 1$ is repeatedly applied to the system to generate the first input sequence. The second input sequence is a uniformly distributed random white noise sequence whose bounds are set such that it has the same input energy as the OVE-ISP sequence. The final input sequence uses an approach which was developed by Pronzato and Walter and can be found in [40]. This input sequence is "bang-bang" in nature and is designed

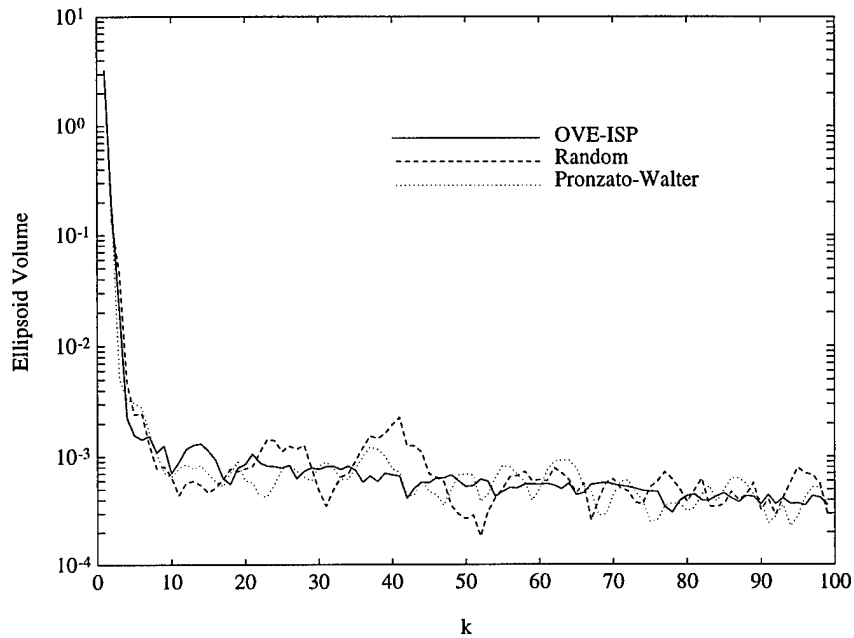

Figure 27: First time-varying example comparing system inputs

using an "optimal" one-step-ahead cost function with a constraint on the magnitude of the input. We will choose the bound on the input level so that the energy of this input sequence also matches the energy of the OVE-ISP sequence.

Before applying any of these input sequences, a uniformly distributed random white noise sequence of length 3 was applied to the system to "initialize" the OVETV algorithm. The ellipsoid volumes for the three input sequences are shown in Figure 27. While OVE-ISP may have a slight advantage during the transient phase, overall, there appears to be no clear advantage for any of the input sequences. The parameters, center estimates, and upper and lower bounds are shown in Figure 28 for the OVE-ISP input sequence. The plots for the other input sequences were similar.

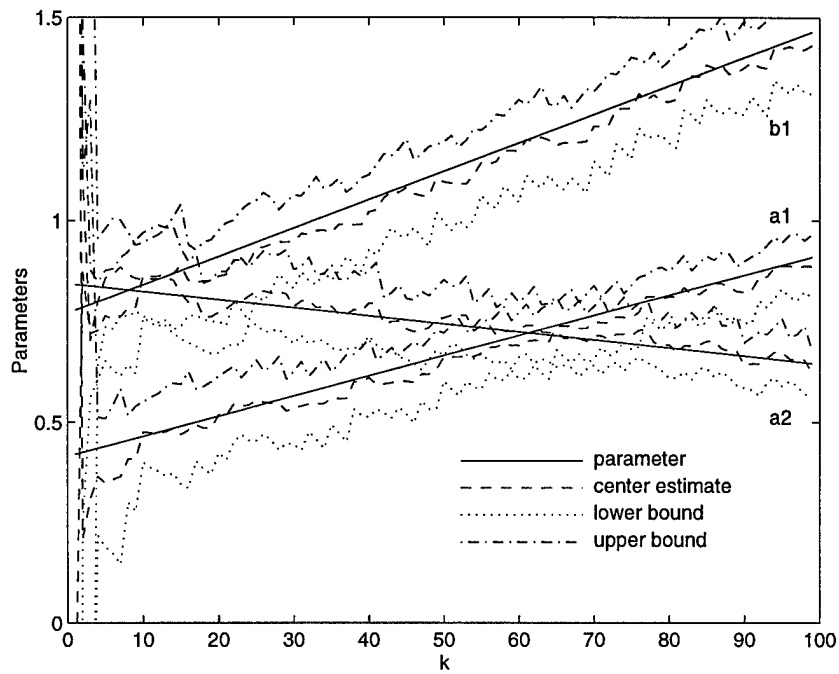

Figure 28: First time-varying example with OVE-ISP input sequence

This simulation is repeated as above except with w_k now given as $w_k = \begin{bmatrix} 0.005 & 0.0035 & -0.007 \end{bmatrix}^T$. The ellipsoid volumes for this second example are shown in Figure 29. Upon first inspection, it would appear that the performance of OVE-ISP is inferior to the performance of the random input and the Pronzato-Walter algorithm. However, if we examine the output sequences in Figure 30 for each of the three input input sequences, we clearly see the source of the discrepancy. The output sequences for the random input and Pronzato-Walter algorithms are growing to levels which would probably be unacceptable for most physical systems, while the output sequence for the OVE-ISP algorithm is remaining "well-behaved." Clearly, the energy of the output sequence has as much effect on the volume of the ellipsoids as

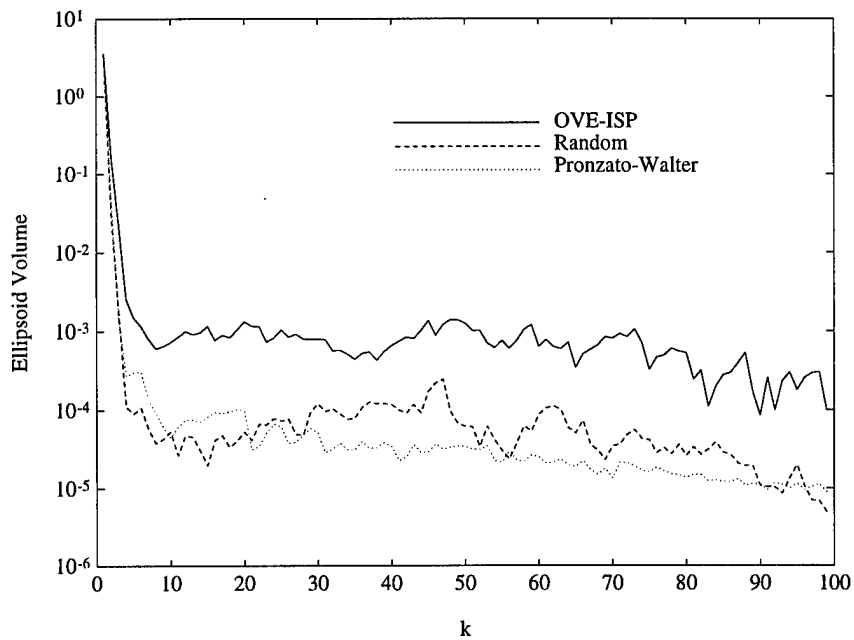

Figure 29: Second time-varying example comparing system inputs

the energy of the input sequence.

The reason for the unacceptable increase in magnitude of the output sequences for the random input and Pronzato-Walter algorithm can be seen in Figure 31. In Figure 31, we see that while the system poles at each time-step during the first example remain within the unit circle, the system poles during the second example clearly move outside the unit circle. While the locations of the "frozen-time" poles by themselves do not guarantee stability or instability for linear time-varying systems, they do provide information, particularly for slow-varying systems, about the local (in-time) behavior of the system.

In the second example, once the poles of the system move outside the unit circle,

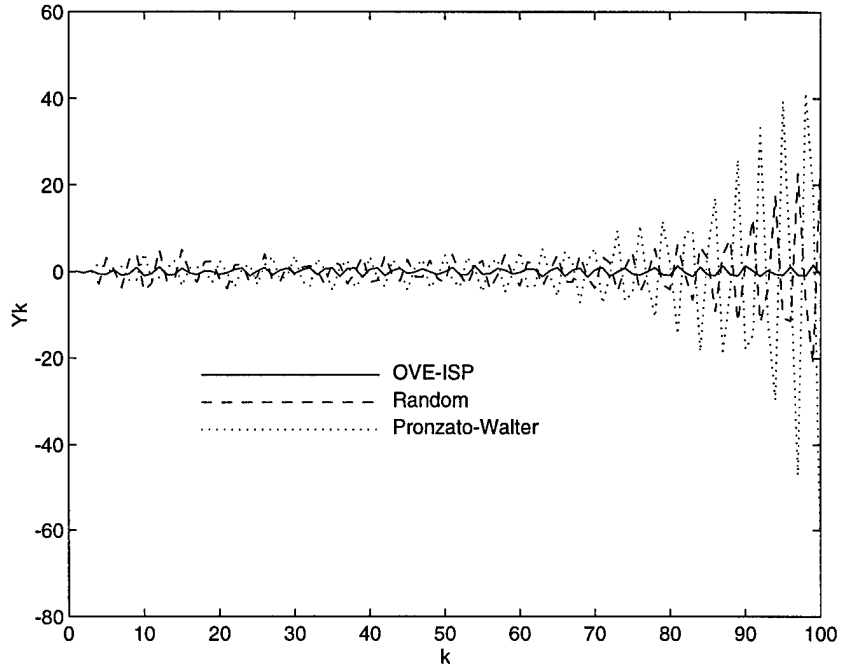

Figure 30: System outputs for second example

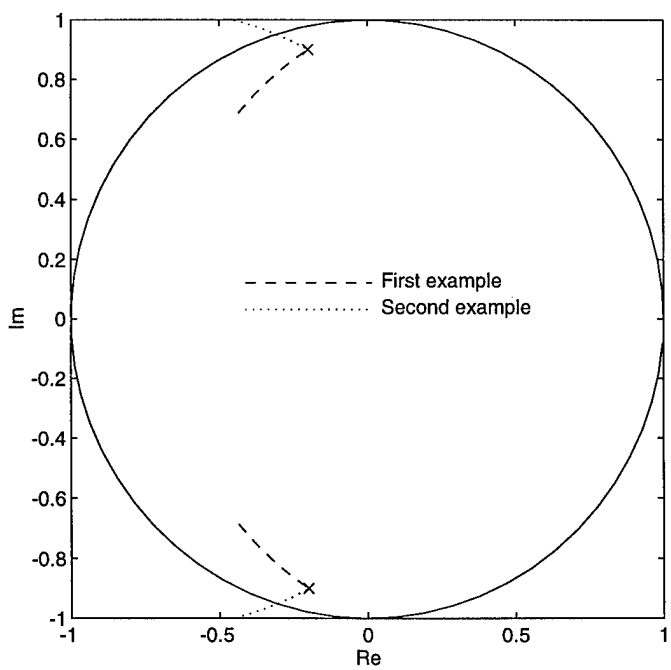

Figure 31: Frozen-time poles of first and second examples

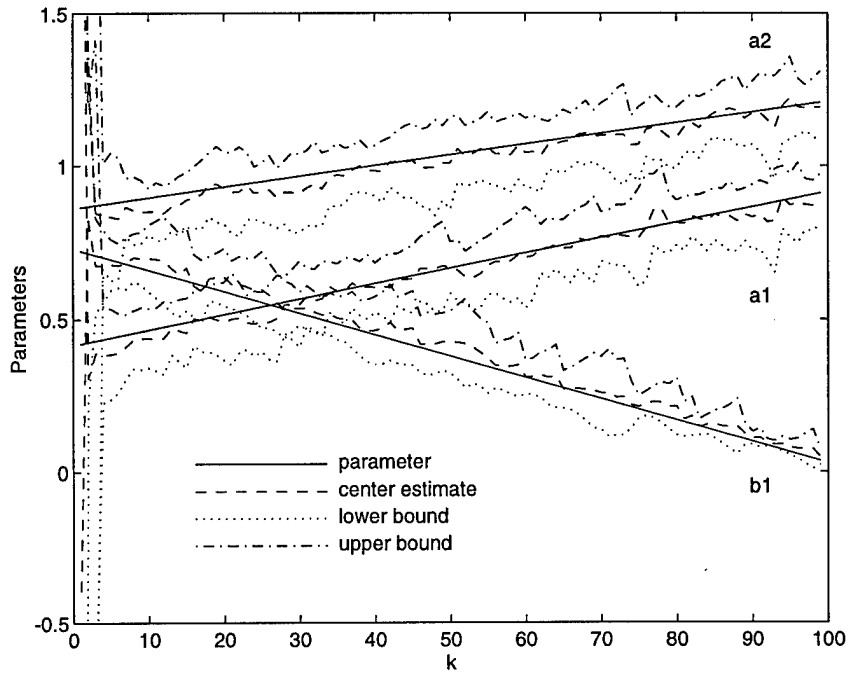

Figure 32: Second time-varying example with OVE-ISP input sequence

the magnitude of the output of the open-loop system grows rapidly. However, the output sequence using the OVE-ISP input sequence remains well-behaved because of the "feedback nature" of its synthesis. Furthermore, as seen in Figure 32, OVETV is able to track the parameters effectively using the OVE-ISP input sequence. This is in contrast to the Pronzato-Walter case where the overall volume is smaller, but the uncertainty around the parameter b_1 becomes very large, as can be seen in Figure 33. OVE-ISP has achieved similar performance for unstable linear time-invariant systems. In conclusion, while this "stabilizing" property is appealing, clearly more analysis is needed to determine when and if this property can be guaranteed.

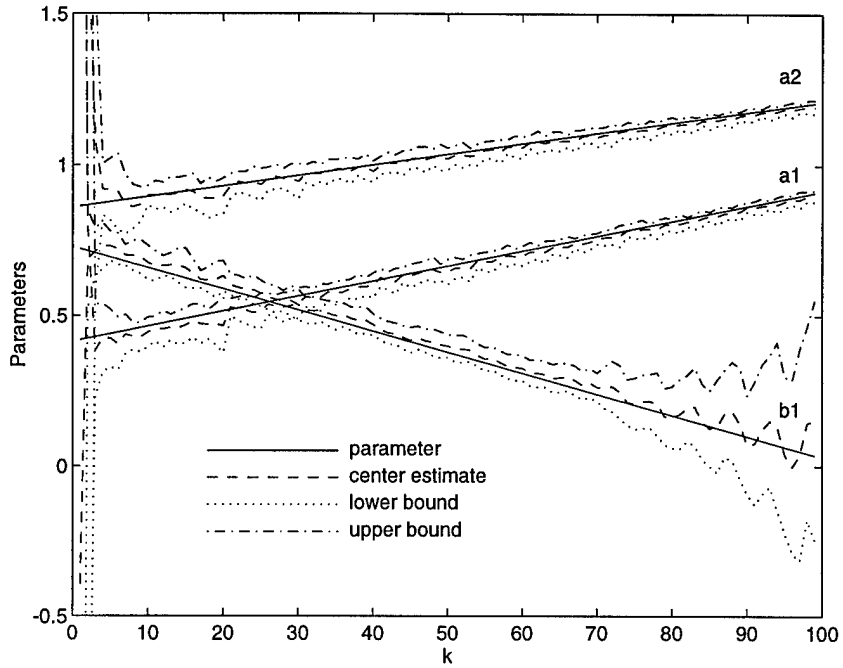

Figure 33: Second time-varying example with Pronzato-Walter input sequence

4.6 Summary

In this chapter, we discussed several algorithm modifications and extensions which serve to improve one or more properties of the OVE or OVETV algorithms. In Section 4.2, we showed how MISO systems could be placed in the OVE or OVETV framework. We began by showing how linear time-invariant MISO state space equations can be placed in the OVE framework. Next, we extended Theorem 5 to linear time-varying MISO systems which satisfy a certain "observability" property. Theorem 6 showed how systems which satisfy this property can be transformed into the OVETV framework. Finally, results were presented showing how the OVETV algorithm could be applied to the crude oil distillation column using the MISO framework.

In section 4.3, we showed how dependencies in parameter variations could be exploited to reduce the number of calculations necessary to implement the OVETV algorithm. When these dependencies are present, support functions are utilized to handle the degeneracies which arise in the ellipsoid bound on the parameter disturbance vector, w_k. Properties of determinants are used to decrease the size of the generalized eigenvalue problem in step 1 of the optimal time update equations, as well as the size of the polynomial which needs to be solved in step 2.

Section 4.4 introduced a square-root implementation of the OVE algorithm. This implementation serves to prevent the ellipsoid orientation matrix, $\tilde{P}_k$, from becoming non-positive definite due to finite precision calculations. The implementation follows an outline similar to Potter's algorithm [36] which is often used in stochastic estimation routines. A comparison with the direct implementation of the OVE algorithm showed that while the square-root implementation requires more floating point operations, the number of operations is of the same order for each implementation. Furthermore, in most cases this increase in number of computations will be less important than the increase in numerical stability which is gained by the square-root implementation. For the time-varying case it was shown how the scalar addition and scalar multiplication algorithms of Section 2.6 can also be implemented in a square-root framework.

Finally, in section 4.5 the OVE-ISP input synthesis procedure of [21] and [11] was investigated. First, we extended OVE-ISP to handle the class of systems which contains a known transportation lag. The modified algorithm also improves the speed

of response for the case where there is no delay. Next, we demonstrated how OVE-ISP and OVETV could be applied to linear time-varying systems. The simulation results presented demonstrate a "stabilizing" property of OVE-ISP which is not possessed by a random input sequence and an alternative synthesis procedure found in [40]. This "stabilizing" property makes OVE-ISP the preferred input scheme for unstable linear time-invariant and linear time-varying systems.

CHAPTER V

Fault Detection and Isolation using PSE

5.1 Overview

To attain autonomy in control systems it is necessary that a controller be able to detect and identify faults in a complex dynamical system. These faults (or failures) may occur in the sensors, actuators, and components of the system we are trying to control. It is very important that a fault detection and isolation (FDI) scheme be able to accurately detect failures in the presence of modeling errors, system and measurement noise, and parameter variations. Because of these stringent requirements, this is a natural setting for a robust system identification technique such as set-membership identification. In this chapter we show how the OVE the OVETV algorithms can be used for detection and isolation of faults in dynamical systems.

The chapter begins with a brief overview of FDI and some of the issues involved in its implementation. Next, we will discuss two methods for detecting faults in dynamical systems. The first method relies on the consistency check which is integral to the OVE and OVETV algorithms, while the second method utilizes an ellipsoid intersection test to detect a fault.

If a fault is detected by an inconsistency check in the OVE or OVETV algorithms, a fault is signalled and the algorithms halt. However, in many situations it is desirable to track the parameters in the parameter space after a fault is indicated. Two "recovery" strategies are presented. The first strategy simply resets the current ellipsoid to a "large enough" ellipsoid guaranteed to contain the "true" parameter after a fault is detected. This is the only method that we know of that can guarantee that the "true" parameter will be contained in the parameter set immediately after a fault it indicated. However, after the ellipsoid is initially reset, there may be a transition time for which the ellipsoid is too large to be effective for applications such as robust adaptive control.

Consequently, an alternative strategy is also introduced. This strategy uses a projection strategy to combine the information contained in the new measurements with the "best" of the information retained by the previous ellipsoid. While this strategy is not guaranteed to capture the "true" parameter, it often succeeds in doing so very quickly. Rules are also proposed for combining these strategies to take advantage of the guaranteed properties of the resetting algorithm and the transition properties of the projection algorithm.

Finally, the algorithms of this chapter are illustrated using the linear time-varying circuit example of Section 3.3. Some techniques for fault isolation are also discussed in our concluding remarks.

5.2 Issues in FDI

The use of FDI is becoming more and more critical as the complexity of systems to be controlled increases. Applications where FDI is necessary include complex process plants where early detection of slowly-varying faults can avoid major plant breakdowns and disasters. FDI is also applicable to high performance vehicles, such as ships, submarines, airplanes, and spacecraft, where safety and significant financial investment are at stake. The application of FDI techniques has also been made possible by the increased computational capability of digital computers and the development of advanced processing techniques.

There are four basic approaches to FDI: (1) limit and trend checking, (2) physical redundancy, (3) analytical redundancy, and (4) knowledge-based redundancy. Limit and trend checking is the simplest and the most widely used. In this strategy, limits are placed on measured variables or on there rate of change. If these limits are surpassed, a fault is detected. Examples include smoke alarms and "dummy" lights on an automobile.

With physical redundancy, multiple sensors are used for each measurement. A voting strategy is used to decide which measurements to use. Multiple actuators may also be used. While this method has the advantage of being simple, the additional hardware requirements may be costly in terms of dollars as well as mass and volume requirements.

Analytical redundancy is an approach, often model-based, which generates residuals utilized to detect and isolate failures. This is the approach used in this dissertation

and will be discussed in detail. Knowledge-based redundancy uses heuristic model information and often functions as a supervisor utilizing the other approaches.

One model considered by analytical redundancy techniques is

$$x((k+1)T) = Ax(kT) + Bu(kT) + Ed(kT) + Kf(kT) \tag{5.1}$$

$$y(kT) = Cx(kT) + Fd(kT) + Gf(kT), \tag{5.2}$$

where $x \in \Re^n$ is the state, $u \in \Re^p$ is the control input, $y \in \Re^q$ in the output, $d \in \Re^l$ is the unknown input vector, $f \in \Re^m$ is the fault vector, and T is the sampling time. The unknown input vector, d, represents noise and modeling errors which may cause false alarms in a FDI strategy. The fault vector, f, is nonzero only when a fault occurs.

Another type of model considered by analytical redundancy techniques is

$$y_k = \theta_k^T \phi_k + v_k, \tag{5.3}$$

where y_k is the system output, θ_k is the parameter vector, ϕ_k is the regression vector, and v_k is the system disturbance. Faults are modeled by changes in the parameter vector, θ_k, which are not anticipated by the no-fault behavior of the system.

There are two types of fault modes with which we are concerned: abrupt and incipient. Abrupt faults are step-like changes which require quick detection for safety reasons. Incipient faults are slowly varying failures, e.g., bias or drift, which take longer to detect, but are usually more maintenance related.

As discussed previously, analytical redundancy relies on the generation of residuals to detect and isolate faults. Residuals are simply functions which are accentuated by

the fault vector f and are ideally zero when $f = 0$. Most analytical redundant schemes require two steps: (1) generation of residuals and (2) decision and isolation of the faults. Residuals are most often generated using either state estimation or parameter estimation. Examples of state estimation schemes include parity checks, observer schemes, and detection filters. Once residuals are generated, they may be used to form decision functions where the relevant information is more apparent. Isolation of the failures is based on a fault signature, which is a signal defining the effects associated with a particular fault (often derived from a model of the faulty system).

To achieve fault detection and isolation, three types of models may be used: nominal representing no fault behavior, actual representing observed behavior, and faulty representing the system after a fault has occurred. Ultimately, we may desire to know the type, size and source as well as the time and location of the fault. To achieve this, often additional heuristic information is required, i.e., knowledge based redundancy.

While FDI has received much attention in the research community (see e.g., [43], [44], and [45]), there has been little investigation into the use of set-membership identification approaches for FDI. This is surprising given the stringent requirements for robustness. Set-membership identification algorithms have the very desirable property that the estimated parameter sets are guaranteed to contain the "true" parameter. It should be mentioned, though, that this robustness property is only as good as the assumptions upon which the parameter estimation algorithms are based.

Work which is related to the use of PSE for FDI can be found in [46], where a

two-ellipsoid overlap test for on-line failure detection is developed. The authors did not use set-membership identification but, instead, used confidence regions based on covariance matrices. They compared the confidence region of an on-line identified model of the system with the confidence region of a nominal model. A fault was detected when the confidence regions did not overlap.

In [14], the authors show how a version of the OBE algorithm developed in [4] can be modified by artificially increasing the noise bound to guarantee consistency if an update takes place. This algorithm suffers from the fact that this consistency can still be lost if no update takes place. However, they have developed a "rescue procedure" for when tracking is lost, for example due to an unexpected large parameter jump which could be caused by a fault. Other rescue procedures are discussed in [47].

5.3 Detection of Failures

A critical part of an FDI scheme is the ability to detect failures. In this section, we will discuss two approaches for detection of failures. The first approach uses the consistency check which is integral to the OVE and OVETV algorithms. This check determines if the intersection between the ellipsoid resulting from the time update, $\hat{E}_k$, and the region based on the new measurements, F_k, is empty. The second approach combines the ellipsoid intersection test of [46] with the OVETV algorithm. An alternative, more straight-forward, development of the ellipsoid intersection test is given.

5.3.1 Consistency Check

Consider the system given by (2.1) and (2.2) with known bounds given in (2.3) and (2.4). To apply the OVETV algorithm to this system, we assume that the following are known: the system structure, i.e., n, m, and l, the bounds on the system disturbance, i.e., $\underline{\gamma}_k$ and $\overline{\gamma}_k$, the bound on the parameter disturbance vector, i.e., R_k, and the initial ellipsoid which is guaranteed to contain the "true" parameter vector, i.e. E_0. Under the assumption that these parameters are known, the algorithm produces an ellipsoid at each time k, E_k, which is guaranteed to contain the "true" parameter vector.

However, if any of these are unknown, we may have the case where $\hat{E}_k \cap F_k = \emptyset$, where F_k is the region between the two parallel hyperplanes, and $\hat{E}_k$ is the ellipsoid resulting after the time update equations. For an example, consider the first-order system in Section 3.2. With the algorithm initialized as before, at $k = 16$ a fault is induced causing the parameter a_k to jump by 0.25, i.e., $a_{16} = a_{15} + 0.25$. This violates the bounds on the parameter disturbance vector, w_k, and produces the simulation results shown in Figure 34. Clearly, $\hat{E}_{16} \cap F_{16} = \emptyset$, indicating that a fault has occurred.

Let us formalize an approach implied by this simple example. Let the nominal system model be given by

$$y_k = \theta_k^T \phi_k + v_k, \tag{5.4}$$

$$\theta_{k+1} = \theta_k + w_k, \tag{5.5}$$

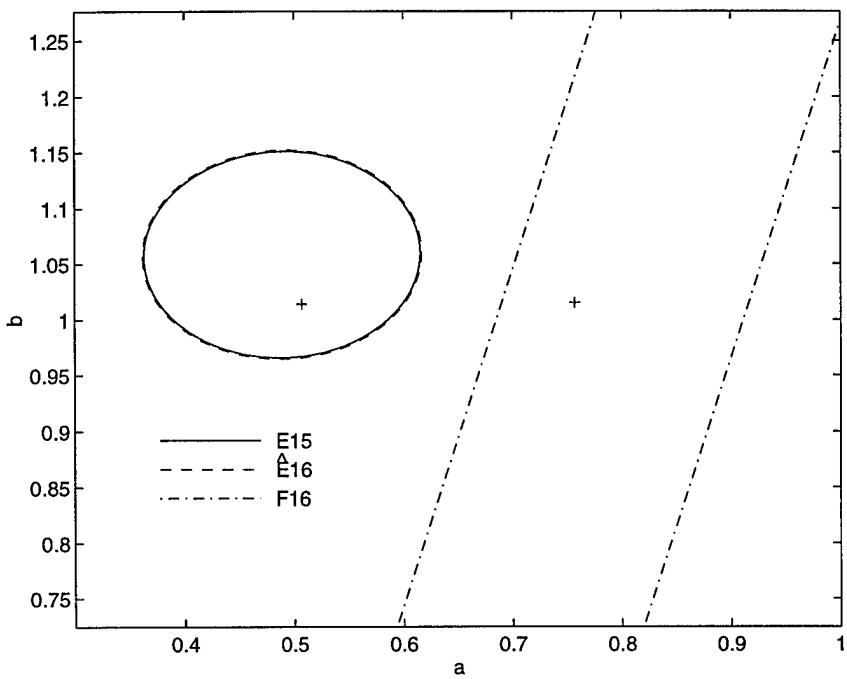

Figure 34: Fault detected at $k = 16$

where $\theta_k = \begin{bmatrix} a_{1k} & \cdots & a_{nk} & b_{lk} & \cdots & b_{mk} \end{bmatrix}^T$, $\theta_0 \in \tilde{E}_0$, and $\phi_k = \begin{bmatrix} -y_{k-1} & \cdots & -y_{k-n} & u_{k-l} & \cdots & u_{k-m} \end{bmatrix}^T$. The system disturbance, v_k, and the parameter disturbance vector, w_k, are assumed to satisfy the following bounds

$$\underline{\tilde{\gamma}}_k \leq v_k \leq \overline{\tilde{\gamma}}_k, \tag{5.6}$$

$$\{w_k \in \Re^r : \theta^T w_k \leq \sqrt{\theta^T \tilde{R}_k \theta}; \theta \in \Re^r\}, \tag{5.7}$$

where $\underline{\tilde{\gamma}}_k < \overline{\tilde{\gamma}}_k$, and $\underline{\tilde{\gamma}}_k \in \Re$, $\overline{\tilde{\gamma}}_k \in \Re$, and $\tilde{R}_k \in \Re^{r \times r}$ are known at each time k. The matrix $\tilde{R}_k$ is symmetric, semi-positive definite.

From these quantities, define the following sequences:

$$\begin{array}{ll} U_N = \{u_k\}_{k=0,\ldots,N} & Y_N = \{y_k\}_{k=0,\ldots,N} \\ V_N = \{v_k\}_{k=0,\ldots,N} & W_N = \{w_k\}_{k=0,\ldots,N}. \end{array} \tag{5.8}$$

For this work, we will define a fault as follows.

Definition 1 *For a given input sequence, U_N, and output sequence, Y_N, of the actual system, a* **fault** *is said to have occurred if there does not exist a pair of sequences, V_N and W_N, which satisfy the nominal system equations, (5.4) and (5.5), and nominal system bounds, (5.6) and (5.7).*

We can now state the following theorem.

Theorem 7 *Given that*

- *the actual system satisfies (5.4)-(5.7) under nominal operating conditions,*
- *the OVETV algorithm is initialized such that $\overline{\gamma}_k \geq \tilde{\overline{\gamma}}_k$, $\underline{\gamma}_k \leq \tilde{\underline{\gamma}}_k$, $R_k \geq \tilde{R}_k$, and $E_0 \supset \tilde{E}_0$, and*
- *the OVETV algorithm is applied to the input sequence, U_N, and output sequence, Y_N, of the actual system.*

If

$$\hat{E}_k \cap F_k = \emptyset, \tag{5.9}$$

for any $k \in \{0, \ldots, N\}$, then a fault has occurred.

Proof: Given the input sequence, U_N, and output sequence, Y_N, of the actual system, assume that $\hat{E}_k \cap F_k = \emptyset$, but that *no* fault has occurred. Therefore, there exist sequences, V_N and W_N, such that the nominal system equations and nominal systems bounds are satisfied. However, if V_N and W_N exist which satisfy (5.4)-(5.7), by construction of the OVETV algorithm,

$$\hat{E}_k \cap F_k \neq \emptyset. \tag{5.10}$$

This is a contradiction. Therefore, if $\hat{E}_k \cap F_k = \emptyset$, then there cannot exist sequences, V_N and W_N, such that the nominal system and nominal system bounds are satisfied. Thus, a fault must have occurred. □

In this section we applied Theorem 7 to the first-order example of Section 3.2. In Section 5.5, we will demonstrate this detection strategy on a more complicated example, the linear time-varying circuit of Section 3.3.

When a fault is detected by this algorithm, i.e, when $\hat{E}_k \cap F_k = \emptyset$, the OVETV algorithm halts. However, there are many situations where it would be desirable to track the parameters after a fault is detected. In Section 5.4, algorithms are discussed which allow the OVETV algorithm to "recover" after a fault is detected. In the next section, we will discuss another algorithm for fault detection. This algorithm has the advantage that for linear time-invariant systems with incipient faults, no "recovery" strategy is needed to continue tracking the parameters after a fault is detected.

5.3.2 Ellipsoid Intersection Test

Consider the system given by (2.1) and (2.2) with known bounds given in (2.3) and (2.4). Assume that under nominal conditions, the system can be described by (5.4)-(5.7) where $\tilde{R}_k$ is known to be equal to 0 for all k, i.e., the nominal system is time-invariant. If we apply the OVE algorithm to the system when it is operating under nominal conditions, we will get a resulting ellipsoid, E_{nom}.

To monitor the system for incipient (slowly-varying) faults, we can apply the OVETV algorithm with R_k set greater than 0 such that parameter tracking is never

lost. In this situation, we can compare the ellipsoid, E_k, produced by the OVETV algorithm during actual operation with the nominal ellipsoid, E_{nom}, which was produced during nominal operating conditions. If these ellipsoids intersect, it is possible that the actual plant matches the nominal plant model. However, if the ellipsoids do not intersect, clearly a fault has occurred.

For an example, consider the first-order system in section 3.2. Initially, the OVE algorithm was applied to the system under nominal conditions, i.e., $a_k = 0.5$ and $b_k = 1.0$. This resulted in the ellipsoid

$$E_{nom} = \{\theta : (\theta - \theta_{nom})^T P_{nom}^{-1} (\theta - \theta_{nom}) \leq 1; \theta \in \Re^r\}, \tag{5.11}$$

where

$$P_{nom} = \begin{bmatrix} 0.0007 & 0.0006 \\ 0.0006 & 0.0011 \end{bmatrix} \qquad \theta_{nom} = \begin{bmatrix} 0.4970 \\ 1.0045 \end{bmatrix}. \tag{5.12}$$

During the actual operation of the plant, the parameters begin to drift, just as in Section 3.2. In order to track the parameters as they drift, the OVETV algorithm was initialized the same as in Section 3.2. In Figure 35, we see at that time $k = 99$, the nominal ellipsoid, E_{nom}, and the actual ellipsoid, E_k, fail to intersect. Clearly, a fault has occurred.

In this section, we will develop an algorithm to automatically detect if two ellipsoids intersect. While such an algorithm was previously developed in [46], the development in this section serves to illustrate the process and is, we feel, more straight forward than that of [46]. Furthermore, we will recommend an additional step to the algorithm presented in [46] which often results in a reduced number of computations. Finally, the detection strategy discussed above will be stated in terms of a theorem.

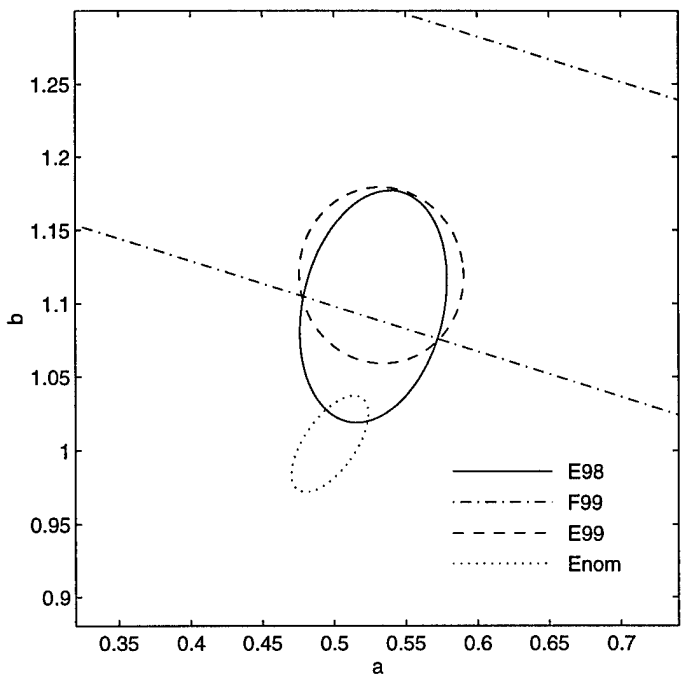

Figure 35: Fault detected at $k = 99$

Ellipsoid Intersection Equations

We wish to develop a test to determine if two ellipsoids,

$$\tilde{E}_1 = \{\theta : (\theta - \theta_1)^T P_1^{-1} (\theta - \theta_1) \leq 1; \theta_1 \in \Re^r\} \tag{5.13}$$

$$\tilde{E}_2 = \{\theta : (\theta - \theta_2)^T P_2^{-1} (\theta - \theta_2) \leq 1; \theta_2 \in \Re^r\} \tag{5.14}$$

intersect. This is equivalent to testing whether the following ellipsoids intersect

$$E_1 = \{x : x^T P_1^{-1} x \leq 1; x \in \Re^r\} \tag{5.15}$$

$$E_2 = \{x : (x - x_2)^T P_2^{-1} (x - x_2) \leq 1; x_2 \in \Re^r\}, \tag{5.16}$$

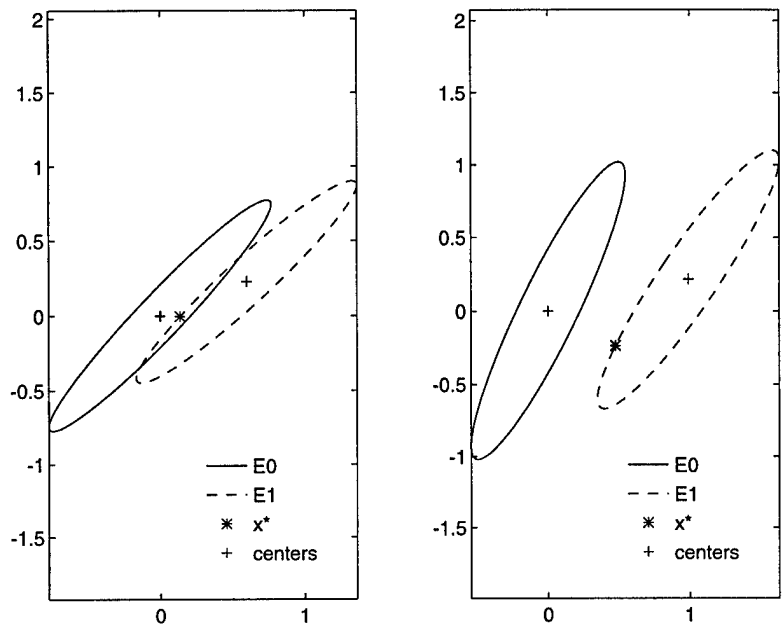

Figure 36: Intersecting and nonintersecting ellipsoids

where $x = \theta - \theta_1$ and $x_2 = \theta_2 - \theta_1$. A pair of intersecting and a pair of nonintersecting ellipsoids are shown in Figure 36.

If the center of E_1 is contained within E_2,

$$x_2^T P_2^{-1} x_2 \leq 1, \tag{5.17}$$

then the ellipsoids intersect, i.e., $E_1 \cap E_2 \neq \emptyset$. If $E_1 \cap E_2 \neq \emptyset$ and (5.17) is not satisfied, then clearly there must be at least one point, x^*, which lies on the hypersurface of E_2, i.e., $(x^* - x_2)^T P_2^{-1}(x^* - x_2) = 1$, and lies within E_1, i.e., $x^{*T} P_1^{-1} x^* \leq 1$.

This suggests the following development, as a straight-forward alternative to that of [46]. If (5.17) is satisfied, then the ellipsoids intersect and the algorithm stops. If

(5.17) is not satisfied, minimize

$$J(x) = x^T P_1^{-1} x, \tag{5.18}$$

subject to the constraint that x satisfies

$$(x - x_2)^T P_2^{-1} (x - x_2) = 1. \tag{5.19}$$

If $J(x^*) \leq 1$, where x^* is the optimum value, then the ellipsoids intersect, otherwise, they do not. The value of x^* can be seen in Figure 36 for a pair of intersecting ellipsoids and a pair of nonintersecting ellipsoids.

To solve (5.18) subject to (5.19), we form the augmented cost function

$$J_a(x, \lambda) = x^T P_1^{-1} x + \lambda((x - x_2)^T P_2^{-1} (x - x_2) - 1), \tag{5.20}$$

where λ is the Lagrange multiplier. Taking the gradient of (5.20) with respect to x and setting it equal to 0, we get

$$\frac{\partial J_a}{\partial x}(x^*, \lambda^*) = 2P_1^{-1} x^* + 2\lambda^* P_2^{-1} (x^* - x_2) = 0. \tag{5.21}$$

Rearranging (5.21) results in

$$(P_1^{-1} + \lambda^* P_2^{-1}) x^* = \lambda^* P_2^{-1} x_2. \tag{5.22}$$

Assuming that λ^* is positive, then the term multiplying x^* is positive definite and invertible, therefore,

$$x^* = \lambda^* (P_1^{-1} + \lambda^* P_2^{-1})^{-1} P_2^{-1} x_2. \tag{5.23}$$

Taking the gradient of (5.20) with respect to λ and setting it equal to 0 gives

$$\frac{\partial J_a}{\partial \lambda}(x^*, \lambda^*) = (x^* - x_2)^T P_2^{-1} (x^* - x_2) - 1 = 0. \tag{5.24}$$

By substituting (5.23) into (5.24) and performing some algebra, we get

$$x_2^T P_1^{-1}(P_1^{-1} + \lambda^* P_2^{-1})^{-1} P_2^{-1}(P_1^{-1} + \lambda^* P_2^{-1})^{-1} P_1^{-1} x_2 = 1. \tag{5.25}$$

To show that a unique positive solution exists for λ^*, we make the following substitution. Because P_1^{-1} and P_2^{-1} are symmetric, positive definite, there exists a nonsingular matrix S such that

$$P_1^{-1} = S^T S \tag{5.26}$$

$$P_2^{-1} = S^T D S, \tag{5.27}$$

where $D = \text{diag}(d_1, d_2, \ldots, d_r)$ [32]. Substituting (5.26) and (5.27) into (5.25) results in

$$x_2^T S^T (I + \lambda^* D)^{-1} D (I + \lambda^* D)^{-1} S x_2 = 1. \tag{5.28}$$

The left hand side of (5.28) can be written as $\eta(\lambda^*) = \sum_{i=1}^{r} \frac{d_i v_i^2}{(1+\lambda^*)^2}$ where $v = Sx_2 = \begin{bmatrix} v_1 & v_2 & \cdots & v_r \end{bmatrix}$. Clearly, $\eta(\infty) = 0$. At $\lambda^* = 0$, $\eta(0) = x_2^T P_2^{-1} x_2$ must be greater than 1, otherwise, we would have stopped the algorithm earlier because we would have known that the ellipsoids intersect. For $\lambda^* \geq 0$, $\frac{d\eta}{d\lambda}(\lambda^*) \leq 0$ and, therefore, $\eta(\lambda^*)$ is strictly decreasing. Consequently, a unique positive solution must exist for λ^*.

Therefore, the algorithm for detecting when two ellipsoids, $\tilde{E}_1$ and $\tilde{E}_2$, intersect is given as follows:

1. If $x_2^T P_2^{-1} x_2 \leq 1$ where $x_2 = \theta_2 - \theta_1$, then $\tilde{E}_1 \cap \tilde{E}_2 \neq \emptyset$ and the algorithm stops; otherwise continue.

2. Solve $f(\lambda) = x_3^T P_2^{-1} x_3 - 1$ for the unique positive solution, λ^*, where $x_3 = (P_1^{-1} + \lambda P_2^{-1})^{-1} P_1^{-1}(\theta_2 - \theta_1)$. The authors in [46] recommend using the bisection method to solve for λ^*. This method is guaranteed to converge if the initial values for λ are set at 0 and some positive value such that $f(\lambda) < 0$.

3. If $x^{*T} P_2^{-1} x^* \leq 1$ where $x^* = \lambda^*(P_1^{-1} + \lambda^* P_2^{-1})^{-1} P_2^{-1}(\theta_2 - \theta_1)$ and λ^* is found as above, then $\tilde{E}_1 \cap \tilde{E}_2 \neq \emptyset$, otherwise, $\tilde{E}_1 \cap \tilde{E}_2 = \emptyset$, i.e., $\tilde{E}_1$ and $\tilde{E}_2$ do not intersect.

We also recommend an additional step, which is not included in [46], to precede step 1.

0. If $x_2^T P_1^{-1} x_2 \leq 1$ where $x_2 = \theta_2 - \theta_1$, then $\tilde{E}_1 \cap \tilde{E}_2 \neq \emptyset$ and the algorithm stops.

Step 0 determines if the center of $\tilde{E}_2$ is contained in $\tilde{E}_1$. While this step is not necessary for the algorithm to achieve its goal, it will reduce the number of times the search in step 2 has to be conducted.

Ellipsoid Intersection Detection Strategy

Now, let us formalize the detection strategy which utilizes the ellipsoid intersection test developed above. Let the nominal system model be given by (5.4)-(5.7) with $\tilde{R}_k = 0$. Let the actual system under all conditions be given by (5.4)-(5.6) and

$$\{w_k \in \Re^r : \theta^T w_k \leq \sqrt{\theta^T \bar{R}_k \theta}; \theta \in \Re^r\}, \tag{5.29}$$

where $\bar{R}_k \in \Re^{r \times r}$ is known at each time k. The matrix, $\bar{R}_k$, is symmetric, semi-positive definite. We can now state the following theorem.

Theorem 8 *Given that*

- *the actual system satisfies (5.4)-(5.7) under nominal operating conditions with* $\tilde{R}_k = 0$*;*
- *an ellipsoid,* E_{nom}*, was produced by the OVE algorithm which was initialized such that* $\overline{\gamma}_k \geq \overline{\tilde{\gamma}}_k$*,* $\underline{\gamma}_k \leq \underline{\tilde{\gamma}}_k$*, and* $E_0 \supset \tilde{E}_0$*, and which was applied to the system under nominal operating conditions;*
- *the OVETV algorithm is initialized such that* $\overline{\gamma}_k \geq \overline{\tilde{\gamma}}_k$*,* $\underline{\gamma}_k \leq \underline{\tilde{\gamma}}_k$*,* $R_k \geq \bar{R}_k$*, and* $E_0 \supset \tilde{E}_0$*; and,*
- *the OVETV algorithm is applied to the input sequence,* U_N*, and output sequence,* Y_N*, of the actual system.*

If

$$\hat{E}_k \cap E_{nom} = \emptyset, \tag{5.30}$$

for any $k \in \{0, \ldots, N\}$*, then a fault has occurred.*

Proof: Given the input sequence, U_N, and output sequence, Y_N, of the actual system, assume that $E_k \cap E_{nom} = \emptyset$, but that *no* fault has occurred. Therefore, there does exist sequences, V_N and W_N, such that the nominal system equations and nominal systems bounds are satisfied. Therefore, the true parameter, θ^*, must be in E_k for

all $k = 0, \ldots, N$. By construction of the OVE algorithm, θ^* must also be contained within E_{nom}. Therefore,

$$\hat{E}_k \cap E_{nom} \neq \emptyset. \tag{5.31}$$

This is a contradiction. Therefore, if $\hat{E}_k \cap E_{nom} = \emptyset$, then there cannot exist sequences, V_N and W_N, such that the nominal system and nominal system bounds are satisfied. Thus, a fault must have occurred. □

Clearly, this algorithm has an advantage over the consistency check detection strategy in the fact that for incipient faults, no recovery strategy is needed. However, it does require calculations in addition to those of the OVETV algorithm. Furthermore, it would be difficult to apply this algorithm to systems which are nominally time-varying, because it would be required to know E_{nom} as a function of k for all time. Finally, abrupt failures will still, most likely, cause an inconsistency in the OVETV algorithm. Consequently, for abrupt failures, no matter which detection strategy we use, the algorithm recovery strategies of the next section will be important.

5.4 Algorithm Recovery Strategies

When $\hat{E}_k \cap F_k = \emptyset$, the OVE and OVETV algorithms indicate an inconsistency and stop. This is satisfactory if the system being monitored also shuts down immediately (for example, due to safety reasons). However, in many situations it is not safe or desirable for the system being monitored to be shut down immediately.

In situations where the system does not shut down immediately, there are strong

reasons why one may desire that the OVE and OVETV algorithms continue to track the parameters moving in the parameter space. If the algorithm is being used in an indirect adaptive control setting, the parameter set is critical for the on-line control design. To isolate the type of fault that has occurred, it is important that we know where in the parameter space the parameter set has moved to after a fault has been detected. In [48], it is argued that the need for estimation of uncertainty in the parameter estimates (which is integral to OVE and OVETV algorithms) is particularly important for reconfigurable control. Fault isolation will be discussed further in our concluding remarks.

In [47], a recovery strategy is developed for the OBE algorithm of [3]. In [14], a recovery strategy is developed for a modified version of the OBE algorithm [4]. When an inconsistency is detected, they can guarantee, under certain assumptions, that this algorithm will asymptotically contain the "true" parameter.

However, we would argue that while asymptotically capturing the "true" parameter is a nice property, it does not satisfy the requirement that the "true" parameter always be contained within the parameter set. We suggest instead, that after a fault is detected, the ellipsoid, $\hat{E}_k$, be reset to a "large enough" volume ellipsoid such that the "true" parameter is contained within the parameter set.

However, initially this reset ellipsoid may be too large for effective robust adaptive control, isolation, etc. Consequently, an alternative approach, which uses projections to effectively utilize the information gained by new measurement, F_k, and the information contained in the ellipsoid, $\hat{E}_k$, is developed. Finally, an integrated approach is

proposed to take advantage of the guaranteed containment properties of the resetting algorithm and the improved volume properties of the projection algorithm.

5.4.1 Ellipsoid Resetting Algorithm

Consider the example investigated in Section 5.3.1. This is similar to the first-order example considered in Section 3.2. However, at $k = 16$ a fault occurs causing the a_k parameter to jump by 0.25, i.e., $a_{16} = a_{15} + 0.25$. A simulation was run and the fault was detected at time $k = 16$; the intersection of $\hat{E}_{16}$ and F_{16} was empty as can be seen in Figure 34. At this point the OVETV algorithm halts. As discussed above, often we would like to continue to track the parameters in the parameter space.

In this section, we will reset the ellipsoid to a size which is guaranteed to capture the true parameters after the fault has been detected. In the example above, the ellipsoid $\hat{E}_{16}$ was reset to have $\hat{P}_{16} = 2I$ and $\hat{\theta}_{16} = 0$. The results are shown in Figure 37. The algorithm tracks the jump in the parameter very quickly. In fact, it appears that the actual parameter is always contained in the parameter set.

This is verified by examining the normalized center estimate error (see Figure 38), which is found to always be less than one. While the OVETV algorithm is guaranteed to contain the true parameter before a fault occurs, and the ellipsoid resetting algorithm guarantees this property immediately after a fault is detected, there is no guarantee that the true parameter will always be contained during the time between when the fault occurs and when it is detected. For this example, the fault was detected immediately; this will not always be the case.

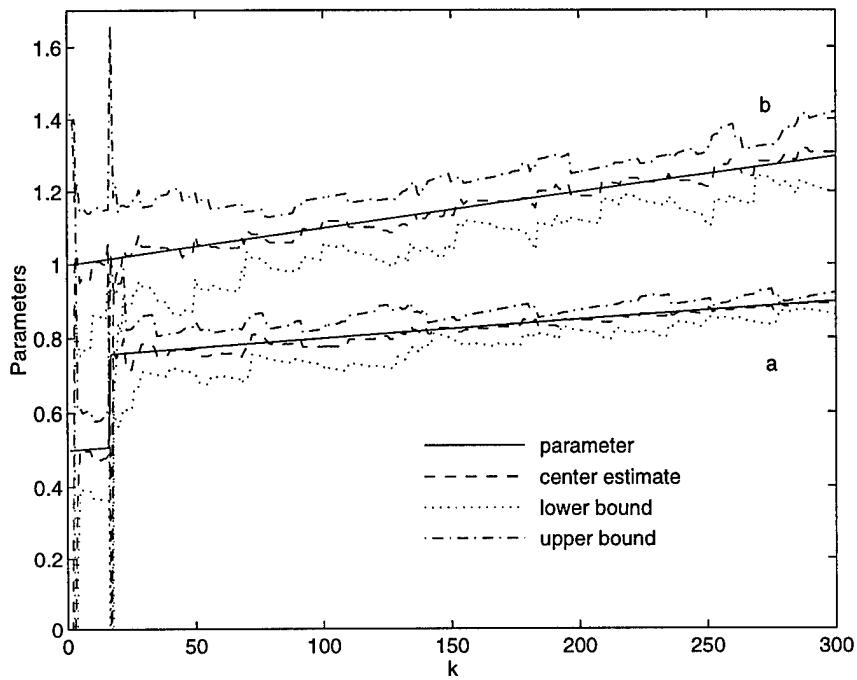

Figure 37: Ellipsoid resetting algorithm results for First-Order example

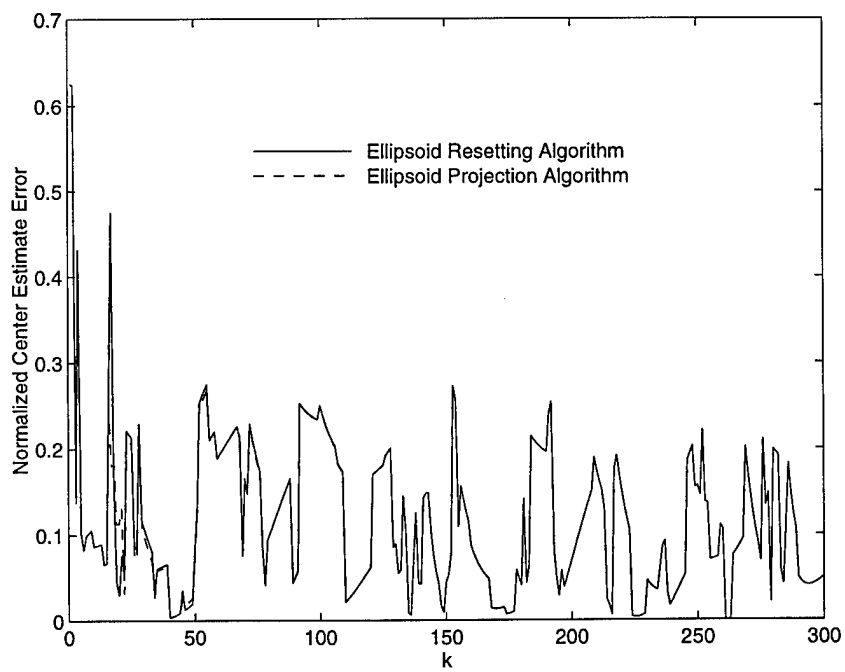

Figure 38: Normalized center estimate error for First-Order example

Furthermore, the ellipsoid resetting algorithm guarantees that the true parameter will always be contained immediately after a fault is detected. Under nominal conditions, once the parameter is captured, it is guaranteed to remain inside the parameter set until another fault occurs. However, if the system does not satisfy the nominal conditions after a fault is detected, e.g. due to continued large jumps in the parameter vector, the true parameter may not remain in the parameter set. Hopefully, however, another inconsistency will indicate the "continued" fault allowing the true parameter to be recaptured permanently. This is an important point and will be discussed further in Section 5.5.1.

One of our concerns about the ellipsoid resetting algorithm was the volume of the ellipsoids immediately after the resetting takes place. Indeed, as seen in Figure 39, the ellipsoid volume does rise sharply immediately after the ellipsoid is reset. However, due to the excitation of the input and the low order of the problem, the ellipsoid volume decreases rapidly. In the next section, we will examine an alternative algorithm which should result in smaller ellipsoid volumes, but lacks the guaranteed recapture property of the ellipsoid resetting algorithm.

The procedure for incorporating the resetting algorithm within the standard OVETV structure is given as follows:

1. Choose the initial ellipsoid, E_0, "large enough" such that the initial "true" parameter vector, θ_0, is in E_0.

2. Find $\hat{E}_k$ such that

$$\hat{E}_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \hat{G}_k\}. \tag{5.32}$$

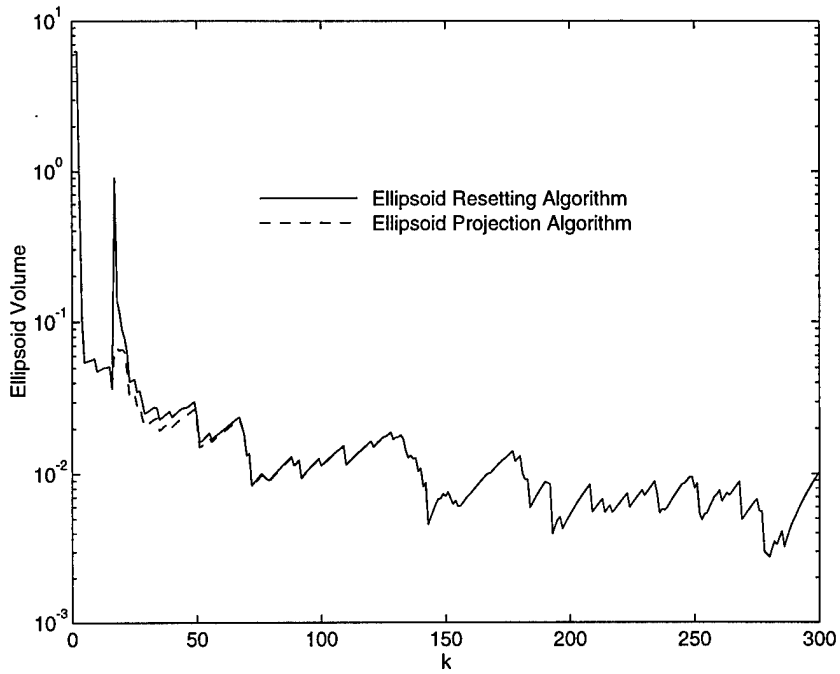

Figure 39: Ellipsoid volume for First-Order example

3. If $\hat{E}_k \cap F_k = \emptyset$, then signal a fault and reset $\hat{E}_k$,

$$\hat{E}_k = E_{reset}, \tag{5.33}$$

where E_{reset} is "large enough" such that "true" parameter vector, θ_k, is in $\hat{E}_k$.

4. Find E_k such that

$$E_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \hat{E}_k \cap F_k\}. \tag{5.34}$$

5. Repeat steps two through four for each new measurement.

Note, that in step 3, the question of how one chooses E_{reset} "large enough" such that $\theta_k \in \hat{E}_k$ is an important one and will be discussed in detail in Section 5.5.2.

5.4.2 Ellipsoid Projection Algorithm

Consider the situation in Figure 40 where, at time k, the "true" parameter θ_k has jumped. The intersection of F_k and $\hat{E}_k$ is empty, and a fault has been detected by the OVETV algorithm. In this section, we want to develop an algorithm to find a new ellipsoid, E_k, which utilizes the information gained by F_k and the "best" information possessed by $\hat{E}_k$.

We are guaranteed that the true parameter must be contained within F_k. The region, F_k, provides information in the direction parallel to the regression vector, ϕ_k. However, it provides no information in the directions orthogonal to ϕ_k. Since we have no additional information, we will make the assumption that in the directions orthogonal to ϕ_k, that the ellipsoid, $\hat{E}_k$, bounds the "true" parameter. While this assumption holds for the example in Figure 40, there is no guarantee that it will hold in general. We will use the results of Sections 2.4.2 (optimal time update equations) and 4.3 (dependent parameter variations) to solve for the minimum volume ellipsoid which bounds the sum of (i) the degenerate ellipsoid found by projecting $\hat{E}_k$ onto the plane orthogonal to ϕ_k and (ii) the degenerate ellipsoid found by projecting F_k onto the line of ϕ_k. In Figure 40, the resulting ellipsoid is given by E_k.

Mathematically, we want to find E_k such that

$$E_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \mathcal{P}(\hat{E}_k, \phi_k^{\perp}) + \mathcal{P}(F_k, \phi_k)\}, \tag{5.35}$$

where $\mathcal{P}(X, Y)$ is the projection of X onto Y and $Y^{\perp}$ is the orthogonal complement of Y. To solve this problem, we begin by finding the singular value decomposition of

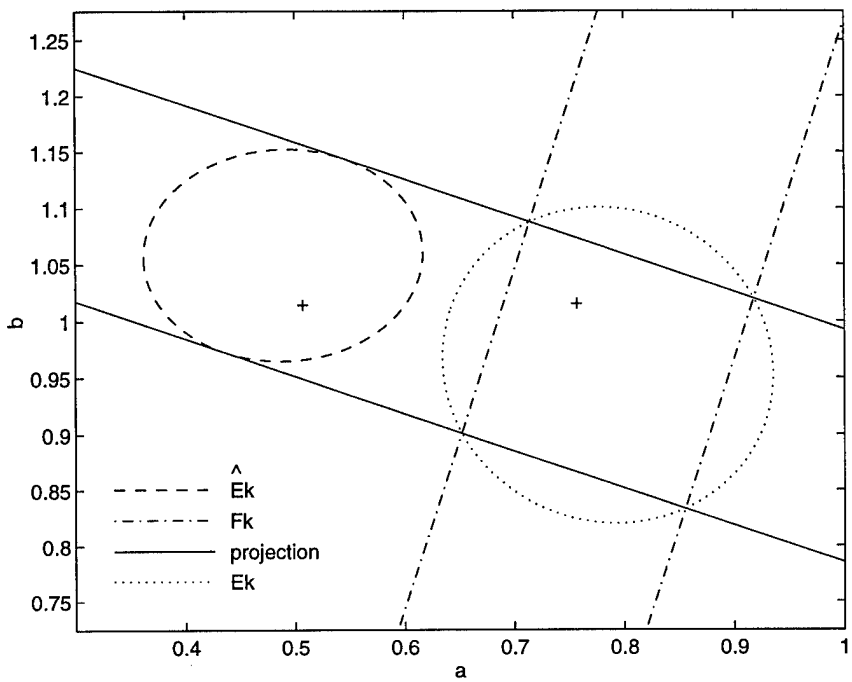

Figure 40: Projection Algorithm

ϕ_k,

$$\phi_k = U\Sigma V^T, \tag{5.36}$$

where $U \in \Re^{r\times r}$ is orthogonal, $\Sigma \in \Re^r$ is equal to $\Sigma = \begin{bmatrix} \sigma & 0 & \cdots & 0 \end{bmatrix}^T$, and $V \in \Re$ is equal to one. It is easy to show that σ must be equal to $\|\phi_k\|_2$. Recall that F_k and $\hat{E}_k$ are defined as

$$F_k = \{\theta_k \in \Re^r : \underline{\gamma}_k \leq y_k - \theta_k^T \phi_k \leq \overline{\gamma}_k\} \tag{5.37}$$

$$\hat{E}_k = \{\theta_k : (\theta_k - \hat{\theta}_k)^T \hat{P}_k^{-1} (\theta_k - \hat{\theta}_k) \leq 1; \theta_k \in \Re^r\}, \tag{5.38}$$

where $\hat{\theta}_k \in \Re^r$ is the center of the ellipsoid and $\hat{P}_k \in \Re^{r\times r}$ is a symmetric, positive definite matrix.

To simplify our problem, we define the affine transformation, T_U, and its inverse by

$$\bar{\theta}_k = T_U(\theta_k) = U^T(\theta_k - \hat{\theta}_k) \tag{5.39}$$

$$\theta_k = T_U^{-1}(\bar{\theta}_k) = U\bar{\theta}_k + \hat{\theta}_k. \tag{5.40}$$

This transformation is similar in form to the transformation defined in [29] for the development of the MOVE algorithm. The key difference lies in the fact that U is orthogonal and therefore directions and distances are preserved.

Substituting (5.40) into (5.38), we see that the ellipsoid, $\hat{E}_k$, is given by

$$\bar{E}_k = \{\bar{\theta}_k : \bar{\theta}_k^T \bar{P}_k^{-1} \bar{\theta}_k \leq 1; \bar{\theta}_k \in \Re^r\} \tag{5.41}$$

in the transformed coordinate systems where $\bar{P}_k = U^T \hat{P}_k U$. The transformation translates $\hat{E}_k$ to the origin and then applies a rotation. Substituting (5.40) into (5.37), we see that the region between the two hyperplanes, F_k, is given by

$$\bar{F}_k = \{\bar{\theta}_k \in \Re^r : \underline{\gamma}_k \leq y_k - \bar{\theta}_k^T \bar{\phi}_k - \hat{\theta}_k^T \phi_k \leq \overline{\gamma}_k\} \tag{5.42}$$

in the transformed coordinate systems where $\bar{\phi}_k = U^T \phi_k$. Using (5.36), $\bar{\phi}_k$ can be rewritten as

$$\bar{\phi}_k = U^T \phi_k = U^T U \Sigma V^T = \Sigma. \tag{5.43}$$

Substituting (5.43) into (5.42) and rearranging terms results in

$$\bar{F}_k = \{\bar{\theta}_k \in \Re^r : \underline{\alpha}_k \leq \bar{\theta}_{k1} \leq \overline{\alpha}_k\}, \tag{5.44}$$

where $\underline{\alpha}_k$ and $\overline{\alpha}_k$ are now defined as

$$\underline{\alpha}_k = \frac{y_k - \phi_k^T \hat{\theta}_k - \overline{\gamma}_k}{\|\phi_k\|_2} \tag{5.45}$$

$$\overline{\alpha}_k = \frac{y_k - \phi_k^T \hat{\theta}_k - \underline{\gamma}_k}{\|\phi_k\|_2}, \tag{5.46}$$

and $\bar{\theta}_k = \begin{bmatrix} \bar{\theta}_{k1} & \bar{\theta}_{k2} & \cdots & \bar{\theta}_{kr} \end{bmatrix}^T$. The variables $\underline{\alpha}_k$ and $\overline{\alpha}_k$ represent the location of the parallel hyperplanes which are orthogonal to $\bar{\phi}_k$.

Let the matrices, $\bar{P}_k$ and U, be partitioned as

$$\bar{P}_k = \left[\begin{array}{c|c} \bar{P}_{11} & \bar{P}_{12} \\ \hline \bar{P}_{21} & \bar{P}_{22} \end{array}\right] \qquad U^T = \left[\begin{array}{c} U_1^T \\ \hline U_2^T \end{array}\right], \tag{5.47}$$

where $\bar{P}_{11} \in \Re$, $\bar{P}_{12} \in \Re^{1\times(r-1)}$, $\bar{P}_{21} \in \Re^{(r-1)\times 1}$, $\bar{P}_{22} \in \Re^{(r-1)\times(r-1)}$, $U_1 \in \Re^{r\times 1}$, and $U_2 \in \Re^{r\times(r-1)}$. The projection of $\bar{E}_k$ onto the subspace orthogonal to $\bar{\phi}_k$ is given by the degenerate ellipsoid

$$\mathcal{P}(\bar{E}_k, \bar{\phi}_k^{\perp}) = \{\bar{\theta}_k \in \Re^r : \eta_r^T \bar{\theta}_k \leq \sqrt{\eta_r^T Q_1 \eta_r}; \eta_r \in \Re^r\}, \tag{5.48}$$

where

$$Q_1 = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \bar{P}_{22} \end{array}\right]. \tag{5.49}$$

The projection of $\bar{F}_k$ onto the axis parallel to $\bar{\phi}_k$ is given by the degenerate ellipsoid

$$\mathcal{P}(\bar{F}_k, \bar{\phi}_k) = \{\bar{\theta}_k \in \Re^r : \eta_r^T \bar{\theta}_k \leq \eta_r^T \check{\theta}_k + \sqrt{\eta_r^T Q_2 \eta_r}; \eta_r \in \Re^r\}, \tag{5.50}$$

where

$$Q_2 = \left[\begin{array}{c|c} \beta^2 & 0 \\ \hline 0 & 0 \end{array}\right] \qquad \check{\theta}_k = \left[\begin{array}{c} \alpha \\ \hline 0 \end{array}\right], \tag{5.51}$$

and $\alpha = \frac{\overline{\alpha}_k + \underline{\alpha}_k}{2}$ and $\beta = \frac{\overline{\alpha}_k - \underline{\alpha}_k}{2}$.

Next, we want to solve (5.35) in the transformed coordinate systems, i.e.,

$$\check{E}_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \mathcal{P}(\bar{E}_k, \bar{\phi}_k^{\perp}) + \mathcal{P}(\bar{F}_k, \bar{\phi}_k)\}. \tag{5.52}$$

This is the problem we solved in Section 2.4.2 for the case where both ellipsoids were nondegenerate and in Section 4.3 for the case where one ellipsoid was nondegenerate. However, in (5.52) both ellipsoids are degenerate. To apply the results of Section 2.4.2, we will let

$$Q_1 = \left[\begin{array}{c|c} \epsilon & 0 \\ \hline 0 & P_{22} \end{array}\right], \tag{5.53}$$

where ϵ is a scalar which we will specify later, and factor Q_2 as

$$Q_2 = BRB^T, \tag{5.54}$$

where $B^T = \left[\begin{array}{c|c} 1 & 0 \end{array}\right]$ and $R = \left[\begin{array}{c} \beta^2 \end{array}\right]$.

From Section 2.4.2 we know that the solution of (5.52) can be found by solving the following generalized eigenvalue problem

$$BRB^T x_j = \lambda_j Q_1 x_j. \tag{5.55}$$

for each λ_j, $j \in [1, r]$. However, from (4.34) in Section 4.3, we know that the solution of (5.52) can also be found by the solving the following reduced order problem

$$B^T Q_1^{-1} B x = \lambda R^{-1} x \tag{5.56}$$

for λ. This is equivalent to solving

$$\det(\lambda R^{-1} - B^T Q_1^{-1} B) = 0 \tag{5.57}$$

for λ. The solution to (5.57) is given by

$$\lambda = RB^T Q_1^{-1} B = \frac{\beta^2}{\epsilon}. \tag{5.58}$$

For the case where $s = 1$, the unique value of $p > 0$ in (4.35) is given by

$$p = \frac{-\lambda(r-1) + \sqrt{\lambda^2(r-1)^2 + 4r\lambda}}{2r}. \tag{5.59}$$

After some algebra, (5.59) can be rewritten as

$$p = \frac{2}{r - 1 + \sqrt{(r-1)^2 + \frac{4r}{\lambda}}}. \tag{5.60}$$

Substituting (5.58) into (5.60) and taking the limit as ϵ approaches 0, we find that

$$p = \frac{1}{r-1}. \tag{5.61}$$

Therefore, from (4.36) and (4.37), we find that the solution to (5.52) is given by

$$\check{E}_k = \{\bar{\theta}_k : (\bar{\theta}_k - \check{\theta}_k)^T \check{P}_k^{-1} (\bar{\theta}_k - \check{\theta}_k) \leq 1; \bar{\theta}_k \in \Re^r\}, \tag{5.62}$$

where

$$\check{P}_k = (p+1)Q_2 + (p^{-1}+1)BRB^T = \left[\begin{array}{c|c} r\beta^2 & 0 \\ \hline 0 & \frac{r}{r-1}\bar{P}_{22}. \end{array}\right] \tag{5.63}$$

Substituting (5.39) into (5.62) gives us the solution to (5.35) in the original coordinate system as

$$E_k = \{\theta_k : (\theta_k - \tilde{\theta}_k)^T \tilde{P}_k^{-1} (\theta_k - \tilde{\theta}_k) \leq 1; \theta_k \in \Re^r\}, \tag{5.64}$$

where

$$\tilde{P}_k = r\beta^2 U_1 U_1^T + \frac{r}{r-1} U_2 \bar{P}_{22} U_2^T \tag{5.65}$$

$$\tilde{\theta}_k = \hat{\theta}_k + \alpha U_1. \tag{5.66}$$

By substituting for α and β and using the fact that $\phi_k = \sigma U_1$, (5.65) and (5.66) can be rewritten as

$$\tilde{P}_k = \frac{r(\overline{\gamma}_k - \underline{\gamma}_k)^2}{4(\phi_k^T \phi_k)^2}\phi_k \phi_k^T + \frac{r}{r-1} U_2 \bar{P}_{22} U_2^T \tag{5.67}$$

$$\tilde{\theta}_k = \hat{\theta}_k + \frac{y_k - \phi_k^T \hat{\theta}_k - \frac{\overline{\gamma}_k + \underline{\gamma}_k}{2}}{\phi_k^T \phi_k}\phi_k. \tag{5.68}$$

Therefore, the ellipsoid projection algorithm for recovering from a fault is given as follows:

1. Find the singular value decomposition of ϕ_k as

$$\phi_k = U \Sigma V^T. \tag{5.69}$$

2. Calculate

$$\bar{P}_k = U^T \hat{P}_k U. \tag{5.70}$$

3. Update $\tilde{\theta}_k$ and $\tilde{P}_k$

$$\tilde{\theta}_k = \hat{\theta}_k + \frac{y_k - \phi_k^T \hat{\theta}_k - \frac{\overline{\gamma}_k + \underline{\gamma}_k}{2}}{\phi_k^T \phi_k}\phi_k \tag{5.71}$$

$$\tilde{P}_k = \frac{r(\overline{\gamma}_k - \underline{\gamma}_k)^2}{4(\phi_k^T \phi_k)^2}\phi_k \phi_k^T + \frac{r}{r-1} U_2 \bar{P}_{22} U_2^T. \tag{5.72}$$

As discussed earlier, one iteration of the projection algorithm is not guaranteed to capture the "true" parameter. Consequently, even after the projection algorithm is applied and the OVETV algorithm resumes, more inconsistencies could arise. If this happens, the projection algorithm would then be applied again.

A procedure for incorporating the projection algorithm within the standard OVETV structure is given as follows:

1. Choose the initial ellipsoid, E_0, "large enough" such that the initial "true" parameter vector, θ_0, is in E_0.

2. Find $\hat{E}_k$ such that

$$\hat{E}_k = \arg_E \min\{\mathrm{vol}(E) : E \supset \hat{G}_k\}. \tag{5.73}$$

3. Find E_k such that

$$E_k = \arg_E \min \left\{ \mathrm{vol}(E) : E \supset \left\{ \begin{array}{ll} \hat{E}_k \cap F_k & \text{if } \hat{E}_k \cap F_k \neq \emptyset \\ \mathcal{P}(\hat{E}_k, \phi_k^{\perp}) + \mathcal{P}(F_k, \phi_k) & \text{otherwise} \end{array} \right. \right\}. \tag{5.74}$$

4. Repeat steps two and three for each new measurement.

This procedure is often able to recapture the "true" parameter very quickly. Consider the first-order example we investigated in Section 5.4.1 where a fault occurs at $k = 16$ causing the a_k parameter to jump by 0.25, i.e., $a_{16} = a_{15} + 0.25$. This simulation was repeated using the projection algorithm, rather than the resetting algorithm, as a recovery strategy. The results are shown in Figure 41.

For this example, the results are very similar to those obtained using the resetting algorithm. The key difference can be seen in the ellipsoid volumes of Figure 39. The large increase in ellipsoid volume, which occurs immediately after the fault is detected in the resetting algorithm results, is not seen in the projection algorithm results. As can be seen in Figure 38, the true parameter is always contained in the parameter set for both simulations. This is particularly surprising for the projection algorithm, where repeated applications of the algorithm are often necessary to recapture the

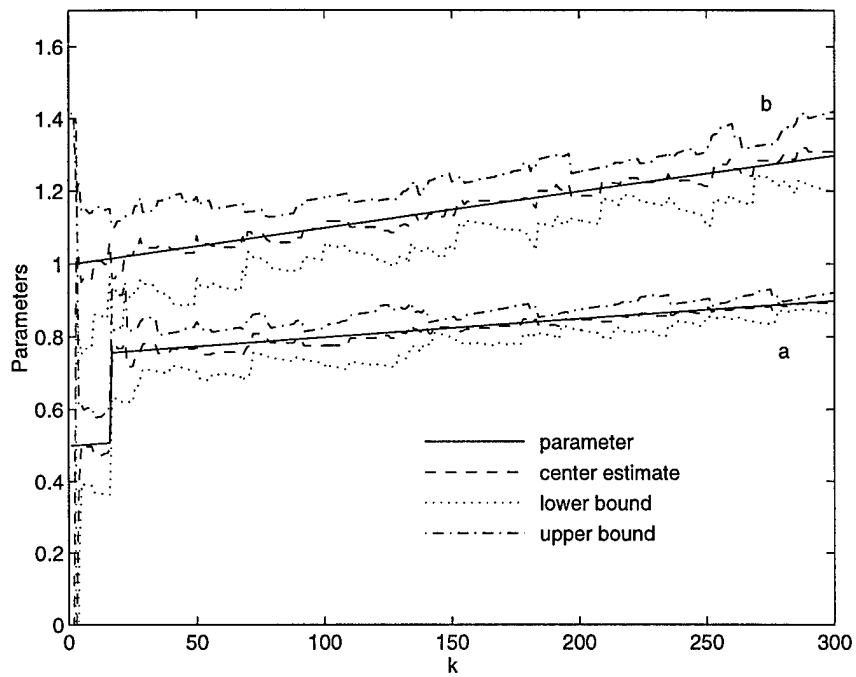

Figure 41: Ellipsoid projection algorithm results for First-Order example

true parameter. In the next section, we will suggest a set of rules which serve to combine the ellipsoid resetting algorithm and the ellipsoid projection algorithm in an integrated approach.

5.4.3 Integrated Approach

The ellipsoid resetting algorithm is guaranteed to capture the true parameter immediately after a fault is detected. This is important for robust control applications and detection of any future faults. However, immediately after being reset, the resulting ellipsoid may be too large for effective control applications. On the other hand, the

projection algorithm does not suffer from the large volume increases encountered by the resetting strategy. While often capturing the true parameter soon after a fault is detected, the projection algorithm may have to "recover" from several inconsistencies before doing so. Furthermore, it lacks the guaranteed properties of the resetting algorithm.

In this section, we propose an algorithm whereby once a fault is detected by an inconsistency, the resetting algorithm and projection algorithm are run in parallel. A set of rules are used to decide which ellipsoid to utilize for robust control applications. Although heuristic, these rules are designed to capture the strengths of each approach. One possible set of rules is given below, where TU are the time update equations of Section 2.4.2, MU are the measurement update equations of Section 2.3, and PU are the projection update equations of Section 5.4.2. In this notation, "MU($\hat{E}_k, F_k$)", for example, means to apply the measurement update equations to the ellipsoid $\hat{E}_k$ and region between two hyperplanes F_k. Once a fault has been detected, $\hat{E}_k^r$ and E_k^r are the ellipsoids resulting from the resetting algorithm, and $\hat{E}_k^p$ and E_k^p are the ellipsoids resulting from the projection algorithm.

The steps of this integrated approach are given below.

1. Choose the initial ellipsoid, E_0

2. $k = k + 1$

3. $\hat{E}_k = \text{TU}(E_{k-1}, R_{k-1})$

4. If $\hat{E}_k \cap F_k \neq \emptyset$, then

- $E_k = \mathrm{MU}(\hat{E}_k, F_k)$
- Go to step 2

5. A fault has been detected: $E_k^p = \mathrm{PU}(\hat{E}_k, \phi_k, F_k)$ $\quad E_k^r = \mathrm{MU}(E_{reset}, F_k)$

6. If (a) $E_k^p \cap E_k^r = \emptyset$, (b) $\mathrm{vol}(E_k^r) \leq \mathrm{vol}(E_k^p)$, or (c) $\mathrm{vol}(E_k^r) \leq \gamma$, then

   - $E_k = E_k^r$
   - Go to step 2

7. $E_k = E_k^p$

8. $k = k + 1$

9. $\hat{E}_k^p = \mathrm{TU}(E_{k-1}^p, R_{k-1})$ $\quad \hat{E}_k^r = \mathrm{TU}(E_{k-1}^r, R_{k-1})$

10. If $E_k^r \cap F_k = \emptyset$, then

    - $\hat{E}_k = \hat{E}_k^r$
    - Go to step 5

11. $E_k^r = \mathrm{MU}(\hat{E}_k^r, F_k)$

12. If $E_k^p \cap F_k = \emptyset$, then

    - $E_k^p = \mathrm{PU}(\hat{E}_k^p, \phi_k, F_k)$
    - Go to step 6

13. $E_k^p = \mathrm{MU}(\hat{E}_k^p, F_k)$

14. Go to step 6

Of particular importance in this integrated approach are steps 6 and 10. Once a fault has been detected, they are used to decide whether the ellipsoids resulting from the projection strategy or the ellipsoids resulting from the resetting strategy are to be used for control purposes. In step 6, rule (a) is the most important. If $E_k^p \cap E_k^r = \emptyset$, E_k^p does not contain the true parameter; therefore, we should use E_k^r for robust control. In rule (b), we choose E_k^r if it has the smallest volume because it is guaranteed to contain the true parameter, while the projection strategy is not. In rule (c), we also choose E_k^r if the volume is "small enough" that a robust controller design is possible, where γ is a prespecified constant. Clearly, other rules could be used instead of (b) and (c) depending on the control design, performance requirements, etc..

Step 10 is used to determine if an additional fault has occurred. Notice that the determination of a fault depends on $\hat{E}_k^r$ and not on $\hat{E}_k^p$ because of the guaranteed containment property of the resetting algorithm.

Probably the biggest disadvantage of the integrated approach is the added complexity. After recovery, and until E_k^r has been selected, two versions of the OVETV algorithm are required to run in parallel.

5.5 Example

In this section, we will consider the linear time-varying circuit which was examined in Section 3.3. Recall that we investigated a series R-L-C circuit where the components

were allowed to vary with time. To obtain the true parameters, the continuous time state-space equations were discretized and then transformed into a difference equation using Theorem 5. Quantization errors in the D/A and A/D converters were also incorporated into the ARX structure.

In the simulations of Section 3.3, the inductance l and capacitance c were held constant, while the resistance r was allowed to vary slowly with time. In this section, we will consider two scenarios. The values of l and c will remain constant for both scenarios. In the first scenario, the nominal model will assume that r varies slowly with time. The consistency check, which is integral to the OVETV algorithm, will be used to detect when an abrupt fault occurs. The ellipsoid resetting and ellipsoid projection algorithms will be applied to recover tracking.

In the second scenario, the system is nominally time-invariant. Our goal is to detect incipient faults and track the parameters after a fault is detected. Two different methods are used to detect a fault. In the first method, the OVETV algorithm is applied to the system with the bounds on the parameter disturbance vector set to track the parameters during an incipient fault. The ellipsoid intersection test detects a fault when the ellipsoid output of the OVETV algorithm does not intersect with the ellipsoid which was found using the OVE algorithm under nominal conditions. In the second method, the OVE algorithm is applied to the system and the internal consistency check is used to detect a fault. The resetting and projection schemes will be applied to recover tracking.

5.5.1 Abrupt Fault

In this section, we will use the consistency check of the OVETV algorithm to detect an abrupt fault in the linear LTV circuit of Section 3.3. Nominally, we expect the circuit to behave just as it did in Section 3.3. Consequently, the OVETV algorithm is initialized the same with $\overline{\gamma}_k = -\underline{\gamma}_k = 0.005$, $R_k = \zeta_k^2 I$ where $\zeta_k = 0.001$, $\tilde{P}_0 = 10I$, and $\tilde{\theta}_0 = 0$. The input, u_{mk}, is a uniformly distributed random variable between $[-5.0, 5.0]$ volts, and the sampling time T is 0.1 seconds.

In Section 3.3, the actual circuit components were: $r(t) = (1.0 + 0.5\cos(t/10.0))\Omega$, $l(t) = 1.0\text{H}$, and $c(t) = 1.0\text{F}$. In this section, the components are identical to those of Section 3.3 from $t = 0.0$ to $t = 10.0$. However, at $t = 10.0$, the resistance $r(t)$ jumps sharply, i.e., $r(t) = (1.0 + 0.5\cos(t/10.0) + 0.5u_s(t - 10))\Omega$, where $u_s(\cdot)$ is the unit step function.

A simulation was run with the OVETV consistency check utilized for detection and the ellipsoid projection scheme utilized for recovery. The consistency check easily detected the fault and signalled immediately at $t = 10.0$ seconds.

Consequently, the projection scheme was applied to recover tracking. Inconsistencies also occurred at $t = 10.2$ and $t = 10.3$ seconds requiring a reapplication of the projection scheme. This reapplication was expected since the projection algorithm is not guaranteed to recapture the parameter. Finally, as seen in Figure 42, the normalized center estimate error goes below 1.0 at $t = 10.6$ seconds. This tells us that the true parameter has been recaptured by the parameter set. Under nominal conditions, once the true parameter has been recaptured, it will stay there until another fault

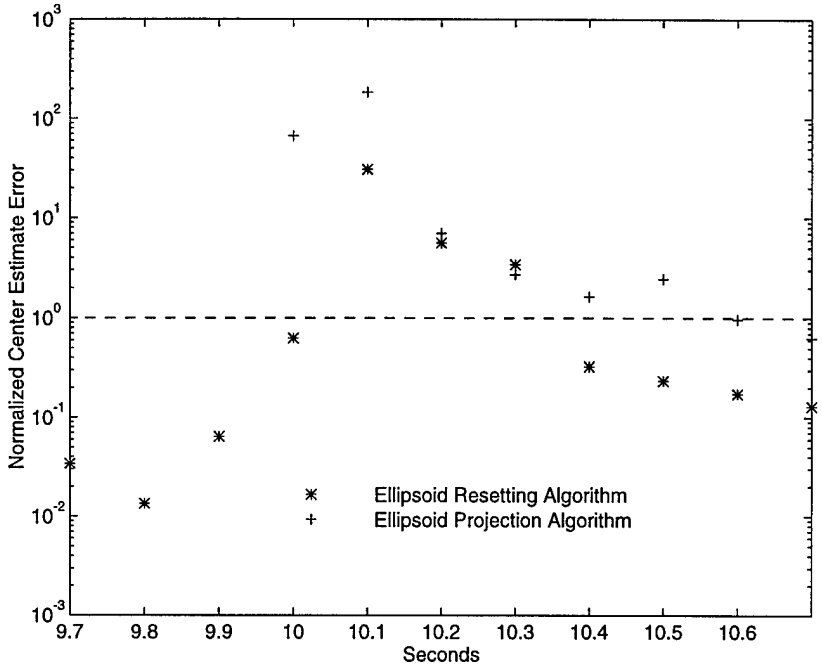

Figure 42: Normalized center estimate error during abrupt fault

occurs. Figure 43 shows the tracking of the parameter b_1 before and after the fault occurs.

The simulation was run again; this time using the ellipsoid resetting scheme for recovery. The resetting ellipsoid, E_{reset}, was set the same as E_0, i.e, $P_{reset} = 10I$ and $\theta_{reset} = 0$. As before, the fault was detected immediately at $t = 10.0$ seconds. As expected, the ellipsoid resetting strategy resets $\hat{E}_k$ to E_{reset} before applying the measurement update equations. As can be seen in Figure 42, the normalized center estimate error for the ellipsoid resetting algorithm is clearly less than one at $t = 10.0$ seconds.

At this point, note that the OVETV signals another inconsistency at $t = 10.4$

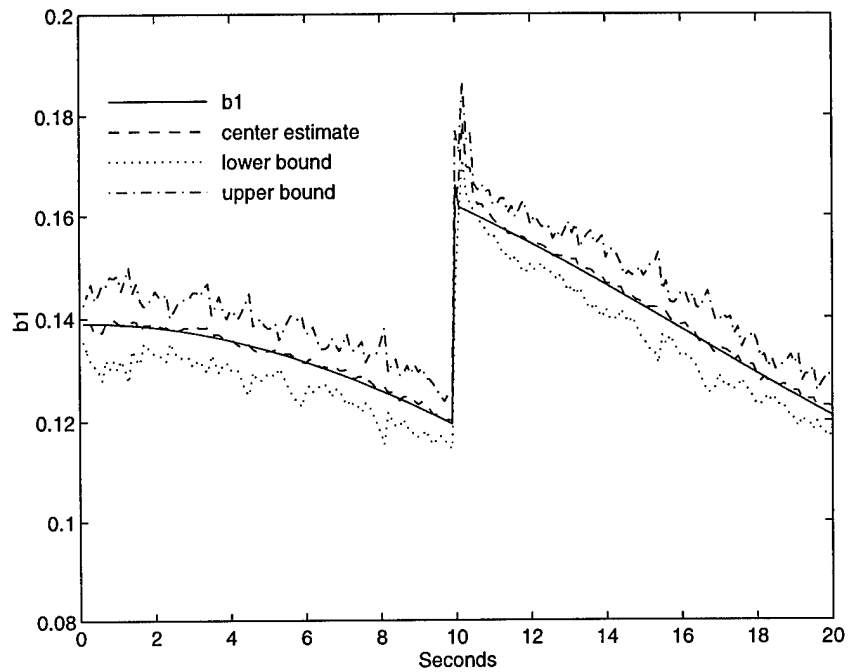

Figure 43: Parameter b_1 with Ellipsoid Projection algorithm

seconds. Intuitively, this may be unexpected since the ellipsoid resetting algorithm was designed to recapture the parameter immediately after a jump in the parameter vector was detected. While the algorithm succeeds in doing this at $t = 10.0$ and $t = 10.4$ seconds, as can be seen in Figure 42, the normalized center estimate error is greater than one between these times.

To understand these results, consider the change in the true parameter vector at each time step, w_k. Under nominal conditions, w_k is bounded by (2.4). Recall from above that R_k was specified to be equal to $\zeta_k^2 I$ where $\zeta_k = 0.001$. When $R_k = \zeta_k^2 I$, the bound in (2.4) can be written as

$$\|w_k\|_2 \leq \zeta_k. \tag{5.75}$$

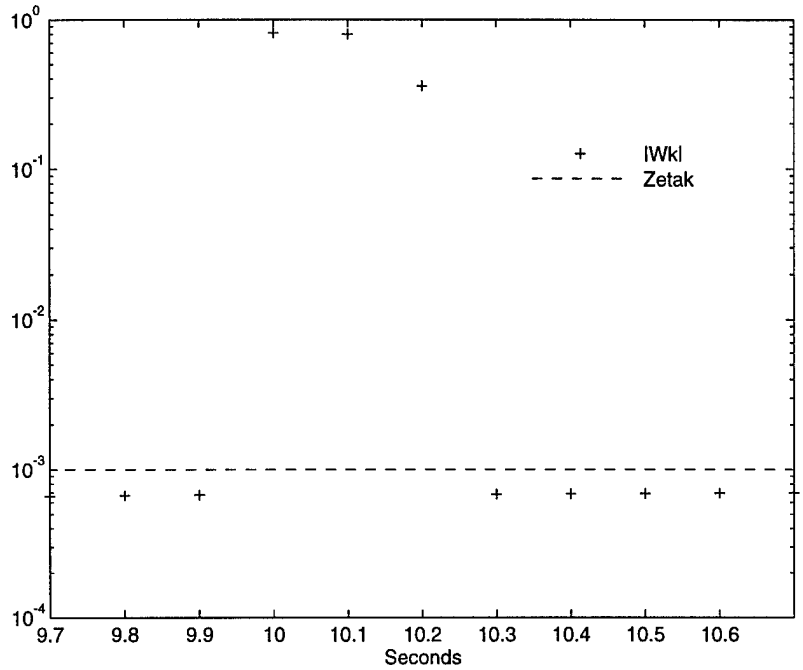

Figure 44: Change in the true parameter vector

In Figure 44, $\|w_k\|_2$ and ζ_k are plotted. As expected, at $t = 10.0$, $\|w_k\|_2 > \zeta_k$. Clearly, a fault has occurred. However, at $t = 10.1$ and $t = 10.2$, $\|w_k\|_2$ is again greater that ζ_k. This is puzzling at first, since the only jump in the continuous time state space equations occurred at $t = 10.0$. However by examining equation (3.32), we see that true parameter vector, θ_k is a function of d_2 at time k, $k-1$, and $k-2$, where $d2(k) = r(kT)$.

Because the OVETV algorithm is highly dependent on the input of the system, we ran this simulation again using a saw-tooth wave as an input. The results were similar, but the inconsistencies were detected at $t = 10.0$ and $t = 10.2$.

This problem poses an interesting challenge for a recovery strategy, because even

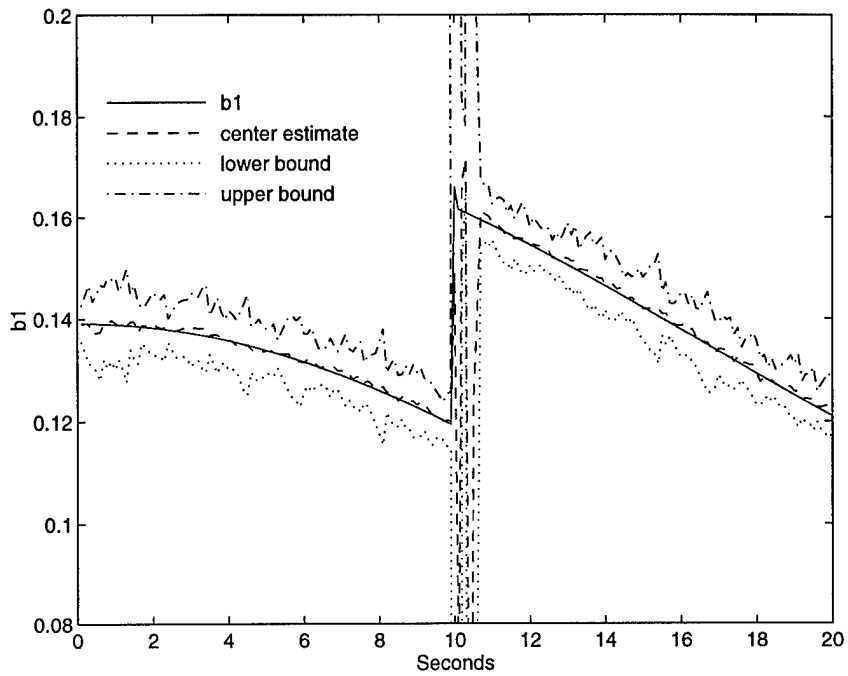

Figure 45: Parameter b_1 with Ellipsoid Resetting algorithm

though the jump in the continuous time parameter occurred during just one time step, it caused the nominal system bounds to be violated for r time steps. Perhaps the integrated approach, which was discussed in Section 5.4.3, could be modified to account for this behavior.

Lest we forget, this algorithm actually performed quite well. It was able to detect the fault immediately, recapturing the true parameter one time step after the fault ended. Figure 43 shows the tracking of the parameter b_1 before and after the fault occurs. As would be expected, the ellipsoid projection algorithm had a smaller volume around the fault area. See Figure 46.

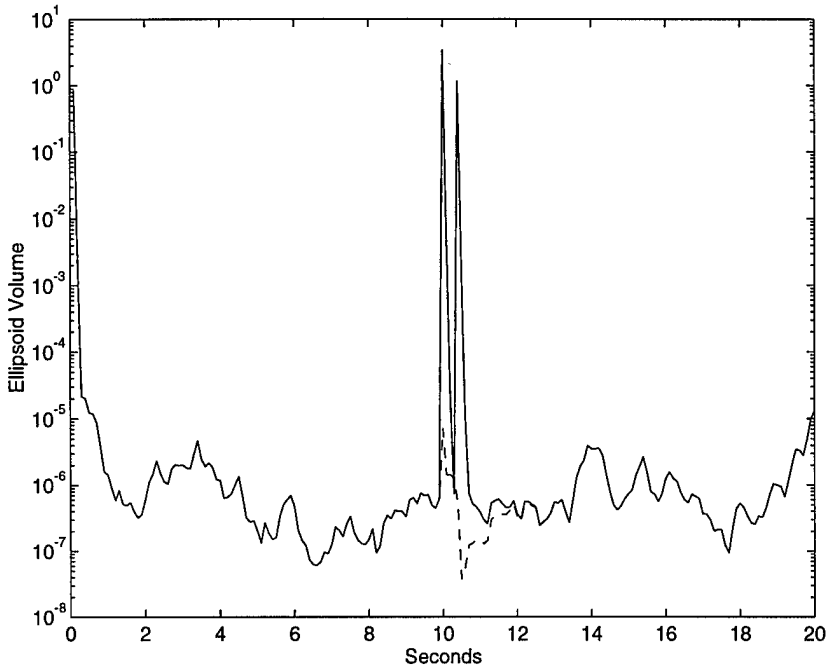

Figure 46: Ellipsoid volume during abrupt fault

5.5.2 Incipient Fault

In the example of the previous section, the nominal plant was slowly time-varying, but an abrupt fault occurred in the actual plant. In this example, the nominal plant is time-invariant and the fault is incipient (slowly-varying). Two methods will be used to detect the fault.

The first method uses the ellipsoid intersection test of section 5.3.2 to detect a fault. First, we apply the OVE algorithm to the system when it is operating under nominal conditions. Under nominal conditions, the circuit components are $r(t) = 1.5\Omega$, $l(t) = 1.0\text{H}$, and $c(t) = 1.0\text{F}$. The OVE algorithm is initialized with $\overline{\gamma}_k = -\underline{\gamma}_k = 0.005$, $\tilde{P}_0 = 10I$, and $\tilde{\theta}_0 = 0$. After 20 seconds, the bounding ellipsoid is

given by E_{nom}, where

$$P_{nom} = \begin{bmatrix} 1.7476e-03 & -1.5366e-03 & 2.6452e-06 & 2.4247e-04 \\ -1.5366e-03 & 1.3697e-03 & -3.5244e-06 & -2.1448e-04 \\ 2.6452e-06 & -3.5244e-06 & 1.9640e-06 & 5.4088e-07 \\ 2.4247e-04 & -2.1448e-04 & 5.4088e-07 & 3.5318e-05 \end{bmatrix} \quad (5.76)$$

$$\theta_{nom} = \begin{bmatrix} -1.8537e+00 & 8.6256e-01 & 1.3916e-01 & -1.3935e-01 \end{bmatrix}^T. \quad (5.77)$$

When the system is undergoing an incipient fault, it is still known to satisfy the bounds which were specified in Section 3.3. Consequently, to monitor the circuit, the OVETV algorithm is initialized the same as in Section 3.3. The system was simulated with the actual circuit components given by $r(t) = (1.0+0.5\cos(t/10.0))\Omega$, $l(t) = 1.0\text{H}$, and $c(t) = 1.0\text{F}$.

The ellipsoid intersection test is used to indicate when E_k, the output of the OVETV algorithm, and E_{nom}, the nominal ellipsoid, fail to intersect. At $t = 3.6$ seconds, the ellipsoids fail to intersect and a fault is indicated. The performance of this algorithm clearly depends on how tightly E_k bounds the actual parameter and the size of the nominal ellipsoid, E_{nom}. For comparison purposes, it was found that the true parameter actually escaped the nominal parameter set at $t = 2.0$. As we saw in Section 3.3, the OVETV algorithm has no problem tracking this system.

The second method we will apply uses the consistency check which is integral to the OVE algorithm to indicate a fault. The OVE algorithm was initialized to bound the nominal system with $\overline{\gamma}_k = -\underline{\gamma}_k = 0.005$, $\tilde{P}_0 = 10I$, and $\tilde{\theta}_0 = 0$. The OVE algorithm was then applied to the measured data from the actual system. At $t = 3.9$ seconds, an inconsistency was indicated, signaling a fault. This was slightly slower than the results achieved using the intersection test. However, this algorithm

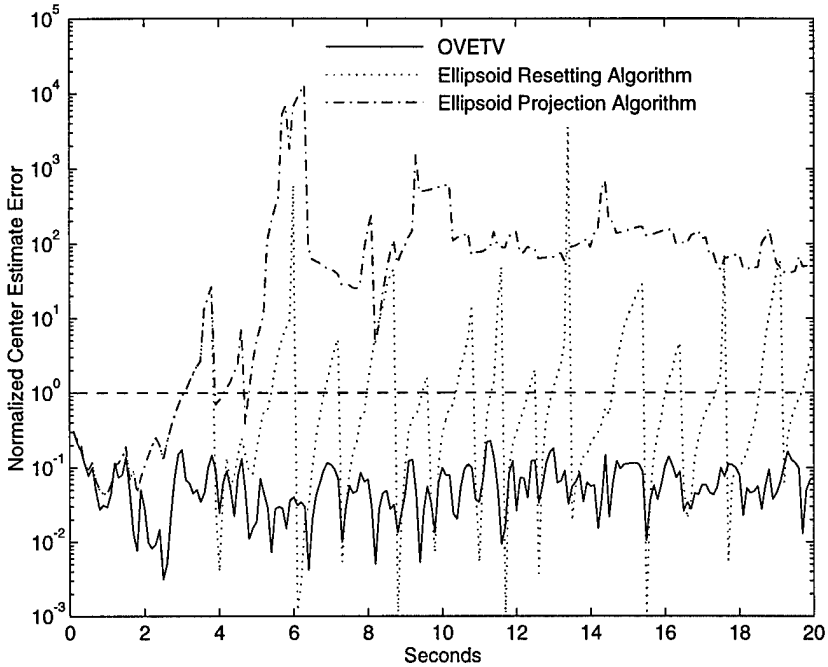

Figure 47: Normalized center estimate error during incipient fault

is significantly cheaper in terms of the number of computations required at each time step. It requires neither the time update computations of the OVETV algorithm nor the computations required for the ellipsoid intersection test.

The ellipsoid intersection strategy has the advantage of built-in tracking after an incipient fault is detected. This raises the question of how well the OVE algorithm can track the system after a fault is detected using the projection and resetting algorithms. With the resetting ellipsoid initialized to $\theta_{reset} = \hat{\theta}$ and $P_{reset} = 1000I$, the normalized center estimate errors for the OVE algorithm with ellipsoid resetting and ellipsoid projection, and for the OVETV algorithm, are given in Figure 47.

Clearly, the OVETV algorithm has the best tracking characteristics in terms of

Table 6: Center estimate performance measures

	OVETV	Ellipsoid Resetting	Ellipsoid Projection
ε_1	0.0800	0.1358	0.0589
ε_2	0.1019	0.1780	0.0740

normalized center estimate error, as it always remains below one. The ellipsoid projection normalized center estimate error resets below one every time an inconsistency occurs, but drifts above one as the parameters continue to vary with time. The results for the ellipsoid projection algorithm are clearly unacceptable since they spend much of the time above one.

While our primary interest is in the parameter sets, it is interesting to note that the center estimate of the ellipsoid projection algorithm actually tracks the true parameter better than either of the other approaches. Based on the performance measures defined in Section 3.3, the projection algorithm results have the lowest cost as can be seen in Table 6.

Having gained some experience with the ellipsoid projection algorithm, this seems to be an appropriate place to discuss how E_{reset} can be chosen "large enough". The answer is definitely problem dependent. If one has a relatively significant amount of *a priori* information about the system, i.e., how it behaves under nominal and fault modes, then it may be possible to specify E_{reset} for the particular case.

If, on the other hand, relatively little is known about the system, at least two

choices exist. In the abrupt fault example, we chose an E_{reset} of the form $\theta_{reset} = 0$, $P_{reset} = \beta I$ where β is a large number. This approach basically discounts any measurements received prior to that point.

In the incipient fault example, we chose an E_{reset} of the form, $\theta_{reset} = \hat{\theta}$ and $P_{reset} = \beta I$ where β is a large number. This gives some credibility to the center estimate of the previous ellipsoid, but greatly increases the uncertainty. Observations made using this form for E_{reset} led to the following Theorem.

Theorem 9 *Let*

$$E(\tilde{\theta}_r, \tilde{P}_r) = \arg_E \min\{vol(E) : E \supset E(\theta_{reset}, P_{reset}) \cap F_k\}, \tag{5.78}$$

and

$$E(\tilde{\theta}_p, \tilde{P}_p) = \arg_E \min\{vol(E) : E \supset \mathcal{P}(E(\hat{\theta}_k, \hat{P}_k), \phi_k^{\perp}) + \mathcal{P}(F_k, \phi_k)\}, \tag{5.79}$$

where $E(\tilde{\theta}, \tilde{P})$ is defined to be

$$E(\tilde{\theta}, \tilde{P}) = \{\theta_k : (\theta_k - \tilde{\theta})^T \tilde{P}^{-1} (\theta_k - \tilde{\theta}) \leq 1; \theta_k \in \Re^r\}. \tag{5.80}$$

Given that $E(\hat{\theta}_k, \hat{P}_k) \cap F_k = 0$, if $\theta_{reset} = \hat{\theta}_k$ and $P_{reset} = \beta I$, then

$$\lim_{\beta \to \infty} \tilde{\theta}_r = \tilde{\theta}_p. \tag{5.81}$$

Proof: The proof is straight forward. From (5.71),

$$\tilde{\theta}_p = \hat{\theta}_k + \frac{y_k - \phi_k^T \hat{\theta}_k - \frac{\overline{\gamma}_k + \underline{\gamma}_k}{2}}{\phi_k^T \phi_k} \phi_k. \tag{5.82}$$

It is easy to show that in the limit, as β approaches infinity, $\tilde{\theta}_r$ equals (5.82). This is done by substituting $\theta_{reset} = \hat{\theta}_k$ and $P_{reset} = \beta I$ into the measurement update equations (2.14)-(2.25) of Section 2.3 and taking the limit as β approaches infinity. □

This theorem is significant because it states that if you choose the parameters of E_{reset} as $\theta_{reset} = \hat{\theta}_k$ and $P_{reset} = \beta I$ where β is some large number, then the center estimate produced by the resetting algorithm is approximately equal to the center estimate produced by the projection algorithm. The orientation matrices, however, are quite different.

5.6 Summary

In this chapter, we demonstrated how the OVE and OVETV algorithms could be used for FDI. We began the chapter by discussing some of the issues involved in the implementation of FDI algorithms.

Next, we discussed two different methods for detection of failures. The first method used the consistency check which is integral to the OVE and OVETV algorithms. Because the check is an integral part of the OVE and OVETV algorithms, this detection strategy was easy to implement. It was particularly adept at detecting abrupt failures in nominally slowly-varying systems.

We also utilized an ellipsoid intersection test to detect when an ellipsoid, produced under nominal conditions, fails to intersect with the current ellipsoid of the OVETV algorithm. The advantage of this algorithm lies in the fact that it is able to track

incipient faults without resorting to recovery strategies. However, it requires additional computations and is not easily applicable to nominally time-varying systems. An alternative development of the ellipsoid intersection test equations was realized by framing the problem as a constrained optimization problem.

We should mention that the ellipsoid intersection test could also be used for fault isolation. Assume that "signature" ellipsoids exist which represent the system under particular faults. Once a fault has been detected and recaptured, the ellipsoid intersection test could be used to check which, if any, signature ellipsoids intersect with the current ellipsoid. If the current ellipsoid does not intersect with a particular signature ellipsoid, then it is clear that the associated fault did not occur. Clearly, if a signature ellipsoid intersects with the current ellipsoid, the associated fault may have occurred. For certain systems, it may be possible to use a modified version of the OVE-ISP algorithm in Section 4.5 to help isolate which fault did occur.

When a fault is detected by an inconsistency in the OVE and OVETV algorithms, the algorithms halt. For applications such as fault isolation and reconfigurable control, it would be desirable to continue tracking the parameters after the fault is detected. Two methods for recovering the true parameter were proposed.

The ellipsoid resetting algorithm resets the current ellipsoid to one large enough to capture the true parameters. If the system satisfies the nominal conditions after the ellipsoid is reset, the algorithm is guaranteed to contain the true parameters until another fault occurs. Several different strategies were considered for choosing the resetting ellipsoid, E_{reset}. It was shown that for a particular form of E_{reset}, that as

the size of E_{reset} is increased, the center estimate approaches the center estimate of our other recovery strategy, ellipsoid projection.

To guarantee that the true parameter is contained in the parameter set after a fault is detected, the ellipsoid resetting algorithm often results with large ellipsoid volumes immediately after a fault is detected. Consequently, a method with better volume properties for recovering the true parameter is also proposed. It combines the information contained in the new measurements and assumes correct information from the old ellipsoid whenever ambiguity arises. Because this is not always the case, repeated applications of the projection algorithm may be required to recapture the true parameter. To take advantage of the guaranteed containment properties of the ellipsoid resetting algorithm and the smaller volumes of the ellipsoid projection algorithm, an integrated approach was also proposed.

Finally, the strategies of this chapter were demonstrated on the LTV circuit of Section 3.3. The algorithms were shown to successfully detect and track both abrupt and incipient faults.

CHAPTER VI

Conclusion

In [49], Ljung, one of the leaders in the field, argues that while system identification is a mature area, there are several important problems that are not sufficiently understood. Two issues that he feels require more investigation are parameter set estimation and the tracking of time-varying properties of systems and signals. In this dissertation, we extended the OVE algorithm for parameter set estimation of linear time-invariant systems to allow for tracking of time-varying parameters. Our experience with the OVETV algorithm challenged us to extend and improve some of the properties of this algorithm; several of these modifications and adaptations are also included in this dissertation. Finally, we demonstrated how the OVE and OVETV algorithms could be utilized for fault detection and isolation.

6.1 Summary

The main result of this dissertation can be found in Chapter II. In this chapter, we combined the optimum volume time update equations of [15] with the optimum volume measurement update equations of [1] to develop the Optimal Volume Ellipsoid

algorithm for Time-Varying systems (OVETV). Given *a priori* assumptions on bounds of the noise entering the system and the possible deviation of the parameters during each time step, this algorithm is guaranteed to contain the true parameter as it varies with time.

Building on the development of the OVETV, we created two algorithms which require fewer computations than the optimal time update equations, thereby making real-time implementation more feasible for many applications. These algorithms reduced the computational complexity of the time update equations by constraining the new ellipsoid to be parameterized by the previous ellipsoid and a single new parameter. One of these algorithms uses the same parameterization as the scalar bound inflation method of [13]. However, the algorithms developed in this chapter are optimized for volume and combined with the optimal volume measurement update equations of [1].

In Chapter III, we applied the OVETV algorithm to three examples. The first example, a simple first-order system, illustrated the key features of the OVETV algorithm and demonstrated the applicability of two alternative algorithms, scalar addition and scalar multiplication.

In the second example, the effects of sampling and quantization on a linear time-varying system, a linear time-varying circuit, were explored. Theorem 5 was given for transforming a discrete-time linear time-varying state space equation to a time-varying difference equation. Interestingly enough, while the goal of the OVETV algorithm is to identify a parameter set, the center estimate of the OVETV algorithm

was able to track the "true" parameter vector better than the "best" estimate using WRLS.

In the final example, we applied the OVETV algorithm to actual data obtained from a distillation column. With the "true" parameter vector effectively unknown, we were not able to compare it with the parameter sets and center estimates. However, the predicted output based on the center estimate matched the actual output very closely.

In Chapter IV, we discussed several algorithm modifications and extensions to the OVE and OVETV algorithms. First, we showed how MISO systems could be placed in the OVE or OVETV framework. Theorem 5 was extended from LTV SISO systems to LTV MISO systems; an extension to MIMO systems was also discussed. Results were also presented showing the application of the OVETV algorithm within the MISO framework to the distillation column of Chapter III.

Next, we showed how dependencies in parameter variations could be exploited to reduce the number of calculations necessary to implement the optimal time update equations of Section 2.4.2. When these dependencies are present, the sizes of the generalized eigenvalue problem and the governing polynomial may be significantly reduced.

A square-root implementation of the OVE algorithm is developed which prevents the ellipsoid orientation matrix, $\tilde{P}_k$, from becoming non-positive definite due to finite precision calculations. A comparison with the direct implementation of the OVE algorithm showed that while the square-root implementation requires more floating

point operations, the number of operations is of the same order for each implementation. Furthermore, in most cases this increase in number of computations will be less important than the increase in numerical stability which is gained by the square-root implementation. For the time-varying case it was shown how the scalar addition and scalar multiplication algorithms of Chapter II can also be implemented in a square-root framework.

The last subject investigated in Chapter IV was the OVE-ISP input synthesis procedure of [21] and [11]. First, the OVE-ISP algorithm was extended to handle systems which contain a known transportation lag. Next, we demonstrated how OVE-ISP and OVETV could be applied to linear time-varying systems. The simulation results demonstrated a "stabilizing" property of OVE-ISP procedure which is not possessed by a random input sequence and an alternative synthesis procedure found in [40]. This "stabilizing" property makes OVE-ISP the preferred input scheme for unstable linear time-invariant and linear time-varying systems.

In Chapter V, we demonstrated how the OVE and OVETV algorithms could be used for FDI. Two different methods for detection of failures were discussed. The first method used the consistency check which is integral to the OVE and OVETV algorithms. Because the check is integral to the OVE algorithms, this detection strategy was easy to implement. It was particularly adept at detecting abrupt failures in nominally slowly-varying systems.

An ellipsoid intersection test was also used to detect when an ellipsoid produced under nominal conditions fails to intersect with the current ellipsoid of the OVETV

algorithm. The advantage of this algorithm lies in the fact that it is able to track incipient faults without resorting to recovery strategies. An alternative development of the ellipsoid intersection test equations of [46] was realized by framing the intersection test as a constrained optimization problem.

Also proposed were two methods for recovering the true parameter once a fault causes an inconsistency in the OVE or OVETV algorithm. The ellipsoid resetting algorithm resets the current ellipsoid to one large enough to capture the true parameters. If the system satisfies the nominal conditions after the ellipsoid is reset, the algorithm is guaranteed to contain the true parameters until another fault occurs. Several different strategies were considered for choosing the resetting ellipsoid, E_{reset}. It was shown that for a particular form of E_{reset}, that as the size of E_{reset} is increased, the center estimate approaches the center estimate of our other recovery strategy, ellipsoid projection.

The ellipsoid projection strategy results in smaller ellipsoid volumes than the ellipsoid resetting strategy. It combines the information contained in the new measurements and assumes correct information from the old ellipsoid whenever ambiguity arises. Because this is not always the case, repeated applications of the projection algorithm may be required to recapture the true parameter. To take advantage of the guaranteed containment properties of the ellipsoid resetting algorithm and the smaller volumes of the ellipsoid projection algorithm, an integrated approach was also proposed.

Finally, during the course of this research, investigation into time-varying linear

systems was imperative. In so doing, we derived some very useful routines using symbolic computations for linear time-varying (LTV) systems which are given in Appendix B.

In short, the major contributions of this dissertation are:

- development of the OVE algorithm for time-varying systems,
- development of alternative algorithms for PSE of time-varying systems, scalar addition and scalar multiplication,
- extension of OVE and OVETV algorithms to MISO systems,
- exploitation of dependencies in parameter variations to reduce computations in optimal time update equations,
- modification of the OVE algorithm using square-root approach for improved numerical stability,
- development of PSE estimation schemes for fault detection, and
- development of schemes for parameter recovery after fault detection.

6.2 Future Work

While there are many directions towards which this research could turn, there are a few areas which seem particularly promising. In Chapter IV, the simulation results presented demonstrate a "stabilizing" property of OVE-ISP which is not possessed by

other input sequences. This stabilizing property could be potentially very important for the identification of unstable systems in the closed loop. However, more analysis is needed to determine when and if this property can be guaranteed.

At the end of Chapter V, we discussed how the ellipsoid intersection test could be used in conjunction with recovery strategies for fault isolation. Certainly, more work is required to explore the use of PSE for fault isolation and/or reconfigurable control. Along these lines, it would be worthwhile to explore how the OVE-ISP procedure could be modified to aid in the isolation process.

Most analytical redundancy approaches for FDI involve either parameter estimation or state estimation schemes. In Chapter V, we explored the use of PSE for FDI. An alternative approach would be to apply the closely related set-membership state estimation schemes to the FDI problem. For some possible starting points, see [50].

Another area for future investigation involving PSE is to use the divided-difference or δ-operator instead of the traditional shift operator where $\delta \equiv \frac{z-1}{\Delta}$ and Δ is a positive number. Traditional digital signal processing and control algorithms using the shift operator often suffer from ill-conditioning at high sampling rates. The δ-operator has been shown to eliminate many of these numerical difficulties [51, 52]. It has also allowed for a unification of continuous and discrete time algorithms [53, 51].

Appendix A

Optimum Parameters for Two-Dimensional Case

The following equations define the parameters p_1 through p_8 used in lemma 3.

$$\begin{aligned} p_1 &= -8\,b_1^6\zeta^4 - 24\,b_1^4\zeta^4a_1^2 - 24\,\zeta^4a_1^4b_1^2 - 8\,\zeta^4a_1^6 - b_1^8\zeta^2 - 76\,b_1^6\zeta^2a_1^2 \\ &\quad +282\,b_1^4\zeta^2a_1^4 - 76\,\zeta^2a_1^6b_1^2 - \zeta^2a_1^8 - 8\,b_1^8a_1^2 - 24\,b_1^6a_1^4 - 24\,a_1^6b_1^4 \\ &\quad -8\,b_1^2a_1^8 \end{aligned} \tag{A.1}$$

$$\begin{aligned} p_2 &= -171\,{b_1}^6\zeta^2{a_1}^4 - 171\,{b_1}^6\zeta^4{a_1}^2 - 333\,{b_1}^8\zeta^2{a_1}^2 + 6\,{b_1}^4\zeta^6{a_1}^2 - 333\,{b_1}^4\zeta^4{a_1}^4 \\ &\quad -39\,{b_1}^4\zeta^2{a_1}^6 + 15\,{b_1}^2\zeta^4{a_1}^6 + 6\,{b_1}^2\zeta^6{a_1}^4 + 15\,{b_1}^{10}\zeta^2 - 39\,{b_1}^8\zeta^4 \\ &\quad +6\,{b_1}^{10}{a_1}^2 + 6\,{b_1}^8{a_1}^4 + 2\,b1^6\zeta^6 + 2\,{b_1}^6{a_1}^6 + 2\,\zeta^6{a_1}^6 \end{aligned} \tag{A.2}$$

$$\begin{aligned} p_3 &= 9\,\zeta^3\sqrt{p_1}\sqrt{3}{b_1}^2{a_1}^2 - 9\,\zeta\sqrt{p_1}\sqrt{3}{b_1}^6 - 3\,\zeta\sqrt{p_1}\sqrt{3}{b_1}^4{a_1}^2 \\ &\quad +3\,\zeta^3\sqrt{p_1}\sqrt{3}{b_1}^4 + 2\,{b_1}^{12} \end{aligned} \tag{A.3}$$

$$p_4 = 6{b_1}^2\sqrt[3]{-\frac{p_2+p_3}{432\,{b_1}^6}} \tag{A.4}$$

$$p_5 = 6{b_1}^2\sqrt[3]{-\frac{p_2-p_3}{432\,{b_1}^6}} \tag{A.5}$$

$$p_6 = \frac{\sqrt{p_4 + p_5 - {b_1}^4 - \zeta^2 {a_1}^2 + 5\,\zeta^2 {b_1}^2 + 5\,{a_1}^2 {b_1}^2}}{b_1\sqrt{6}} \tag{A.6}$$

$$p_7 = -2\zeta^2 a_1^2 + a_1^4 + \zeta^4 + a_2^4 - 2a_2^2 a_1^2 - 2\zeta^2 a_2^2 \tag{A.7}$$

$$p_8 = b_1^2 a_2^2 + a_2^2 a_1^2 + a_1^2 b_1^2 - a_1^4 - \zeta^2 b_1^2 + \zeta^2 a_1^2 \tag{A.8}$$

Appendix B

Symbolic Computations for Linear Time-Varying Systems

B.1 Overview

During this work, it has become clear that the state transition matrix is integral for both analysis and control system design of linear time-varying (LTV) systems. Unfortunately, a closed form solution for the state transition matrix exists only when the LTV system satisfies certain properties. In this appendix we show how Maple V, a computer algebra system, can be used to calculate the state transition matrix for several classes of LTV systems. By working with symbolic rather than numerical data, computer algebra systems offer several advantages including the greater accuracy achieved by the absence of finite precision arithmetic and the additional insight obtained by maintaining the mathematical information and structure. Examples and applications to control system design are discussed.

B.2 Motivation

Although the laws of physics do not change with time, time-varying models arise due to special circumstances in the physical plant or due to the particular formulation of the model [30]. Applications of linear time-varying (LTV) systems include rocket dynamics, time-varying linear circuits, satellite systems, and pneumatic actuators. The LTV structure is also often assumed in adaptive and standard gain-scheduled control systems.

In this appendix, we are interested in LTV systems of the form

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \\ y(t) &= C(t)x(t) + D(t)u(t), \end{aligned} \tag{B.1}$$

where $x(t) \in \Re^n$ is the state vector, $u(t) \in \Re^m$ is the control input, $y(t) \in \Re^p$ is the system output, and $A(t)$, $B(t)$, $C(t)$, $D(t)$ are matrices of appropriate dimension, each with (possibly) time-varying entries. The state transition matrix is the unique solution to

$$\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0), \qquad \Phi(t_0, t_0) = I, \tag{B.2}$$

where I is the identity matrix. The state transition matrix is essential in determining the complete solution, stability, controllability, and observability of (B.1). It is also useful in design of controllers and observers for (B.1). Unfortunately, a closed form solution to (B.2) exists only when $A(t)$ satisfies certain properties.

By working with symbolic rather than numerical data, computer algebra systems offer several advantages for the controls engineer, particularly for problems such as

that outlined above. These advantages include the greater accuracy achieved by the absence of finite precision arithmetic and the additional insight obtained by maintaining the mathematical information and structure [54]. Computer algebra systems also handle complex mathematical manipulations, which if done by hand would be tedious and error prone [55]. One particular computer algebra system, which we use in this work, is Maple V, which is a powerful software package for symbolic and numeric computation. Maple V includes a large library of functions, programming capability, interactive graphics, a worksheet interface, and an online help facility. It is available on many computer platforms including MS-DOS, Windows, Macintosh, NeXT, DEC, Sun and other UNIX workstations.

Maple V can easily find the solution of (B.2) when $A(t)$ is constant or satisfies a well known commutative property. This appendix describes routines written by the authors to expand the class of systems for which Maple V can easily calculate $\Phi(t, t_0)$ to include systems where $A(t)$ is triangular or $\dot{A}(t)$ satisfies certain bracket properties. Routines have also been written to calculate the general solution of (B.2) to any arbitrary order and to check whether a given $\Phi(t, t_0)$ satisfies (B.2).

B.3 State Transition Matrix Properties

The state transition matrix is an integral component in the study of LTV systems of the form given by (B.1). It is used for determining the complete solution, stability, controllability, and observability. It can also be used in the design of controllers and observers for (B.1). In this section we will discuss these uses along with some of the

properties of the state transition matrix. Since this material is somewhat standard in the linear systems literature, we offer here only a brief overview, and present relevant theorems in the Addendum.

The state transition matrix, $\Phi(t, t_0)$, satisfies

$$\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0), \tag{B.3}$$

and has the following important properties [56]:

$$\Phi(t, t) = I \tag{B.4}$$

$$\Phi^{-1}(t, t_0) = \Phi(t_0, t) \tag{B.5}$$

$$\Phi(t_2, t_0) = \Phi(t_2, t_1)\Phi(t_1, t_0). \tag{B.6}$$

Stability of the homogeneous system,

$$\dot{x}(t) = A(t)x(t), \tag{B.7}$$

whose solution is given by

$$x(t) = \Phi(t, t_0)x_0, \tag{B.8}$$

where $x_0 = x(t_0)$, can be determined from the state transition matrix, according to well known stability theorems [56] (see Addendum). The necessary and sufficient conditions on $\Phi(t, t_0)$ for stability are summarized in Table 7.

It is easy to verify that the solution to the non-homogeneous system (B.1) is given by

$$\begin{aligned} x(t) &= \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau \\ y(t) &= C(t)\Phi(t, t_0)x_0 + C(t)\int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau + D(t)u(t). \end{aligned} \tag{B.9}$$

Table 7: Stability bounds on $\Phi(t, t_0)$

Stability Result	Necessary and Sufficient Condition
Stable in the sense of Lyapunov at t_0	$\|\Phi(t,t_0)\| < k(t_0) < \infty$
Uniformly stable in the sense of Lyapunov	$\|\Phi(t,t_0)\| < k < \infty$
Asymptotically stable at t_0	$\|\Phi(t,t_0)\| \leq k(t_0) < \infty$ $\|\Phi(t,t_0)\| \to 0$ as $t \to \infty$
Uniformly asymptotically stable	$\|\Phi(t,t_0)\| \leq k_1 e^{-k_2(t-t_0)}$

To guarantee that the system can be driven from one state x_0 to another state x_1 with an input $u(t)$, it is necessary to show that the system is *controllable.* The LTV system (B.1) is said to be *controllable* if given any x_0 there exists an input $u(t)_{[t_0,t_1]}$ such that $x(t_1) = 0$. Controllability of (B.1) can be determined from the state transition matrix according to a well known theorem [30] (see Addendum).

To guarantee that the system state $x(t)$ can be estimated from the system output $y(t)$, it it necessary to show that the system is observable. The LTV system (B.1) is said to be *observable* on $[t_0, t_1]$ if the initial state x_0 is uniquely determined by the output $y(t)$ for $t \in [t_0, t_1]$. Observability of (B.1) can be determined from the state transition matrix according to a well known theorem [30] (see Addendum). The controllability and observability grammians, $W(t_0, t_1)$ and $M(t_0, t_1)$, respectively, (see Addendum) can also be used in the design of controllers and observers for (B.1).

It is clear that the state transition matrix is important for studying stability, controllability, and observability of (B.1). Calculation of the state transition matrix for linear, time-invariant systems is a straight forward task. Unfortunately, for lin-

ear time-varying systems, it is often difficult if not impossible to calculate the state transition matrix.

B.4 Calculating the State Transition Matrix

In general, a closed form solution for $\Phi(t, t_0)$ does not exist. In this section, several classes of systems for which $\Phi(t, t_0)$ can be calculated in closed form are discussed. Maple V routines which aid in this calculation are discussed. After summarizing classes for which the state transition can be calculated, two decomposition schemes will be discussed which expand on these classes.

Before examining these classes, some comments about Maple V should be given. As mentioned earlier, Maple V contains a large library of functions. While many of these functions are internal to Maple V and available immediately, other functions are grouped together as packages which must be loaded into Maple V before using those commands. One such package, the linear algebra package, contains many common functions for working with vectors and matrices [57]. Several of the functions that we use in this work are given in Table 8. While many of Maple V's other commands are self-explanatory, the procedure **map** also requires some discussion. The procedure **map(f, A, arg2, arg3, ..., argn)** is used to apply a function, f, with multiple arguments to each component of an expression (or matrix), A.

In addition to the linear algebra package, it is assumed that several procedures, which were written by the authors and listed in Table 9, are also loaded. The **eye** and **zeros** procedures are modeled after the Matlab functions of the same names and

Table 8: Maple V linear algebra procedures

linsolve(A, b)	Solves $Ax = b$
evalm$(\cdot)$	Evaluates matrix expression
add(A, B)	Adds matrices A and B
multiply(A, B)	Multiplies matrices A and B
exponential(A, t)	Computes matrix exponential, e^{At}
transpose(A)	Matrix transpose, A^T
eigenvals(A)	Computes eigenvalues of matrix A

return the identity matrix and a matrix of zeros, respectively. The **msubs** procedure substitutes expressions into matrices. The Kronecker product of $A = [a_{ij}] \in \Re^{m\times n}$ and $B \in \Re^{p\times q}$ is defined to be

$$A \otimes B \equiv \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{B.10}$$

where $A \otimes B \in \Re^{mp\times qn}$. The vector vec A is defined to be

$$\text{vec } A \equiv \begin{bmatrix} \bar{a}_1 \\ \vdots \\ \bar{a}_n \end{bmatrix}, \tag{B.11}$$

where $A \in \Re^{m\times n}$, vec $A \in \Re^{mn}$, and the $\bar{a}_i$ are the columns of the matrix A. The command **invec(vec(A) ,m, n)** will return the matrix A. The procedures **kron** and **vec** transform matrix equations of the form

$$AXB = C \tag{B.12}$$

where $C \in \Re^{m\times q}$ and $X \in \Re^{n\times p}$ is unknown, to the equivalent system of equations

$$(B^T \otimes A)\text{vec } X = \text{vec } C. \tag{B.13}$$

Table 9: Maple V utility procedures

eye(·)	Generates identity matrix
zeros(·)	Generates matrix of zeros
msubs($A, s_1, ..., s_n$)	Substitutes expressions into a matrix
kron(A,B)	Kronecker product $A \otimes B$
vec(A)	vec A
invec(a,m,n)	inverse of vec A

This equation can be solved for vec X by **linsolve(D, vec(C))** where $D = B^T \otimes A$ [58].

With these tools, we can now study the solution of the general problem (B.2). While in general there is no closed form solution to (B.2), the solution can be expressed in terms of the Peano-Baker series [30]

$$\Phi(t, t_0) = I + \int_{t_0}^{t} A(\tau_1)d\tau_1 + \int_{t_0}^{t} A(\tau_1) \int_{t_0}^{\tau_1} A(\tau_2)d\tau_2 d\tau_1 + \cdots. \tag{B.14}$$

The following procedure, **peano(A, n, t0)**, allows us to calculate this series to any prespecified order n:

```
peano:=
proc(A,n,t0)
local w1,w2,i,tp;
    if not type(n,posint) then ERROR(`n must be positive integer`)
    elif not type(A,matrix) then ERROR(`A must be matrix`)
    elif not type(t0,scalar) then ERROR(`t0 must be a scalar`)
    fi;
    w2 := eye(A);
    w1 := evalm(w2);
    for i to n-1 do
```

```
        w2 := map(int,multiply(A,w2),t = t0 .. tp);
        w2 := msubs(msubs(w2,t = t.i),tp = t);
        w1 := add(w1,w2)
    od;
    RETURN(")
end
```

For time-invariant systems, that is, when $A(t) = A$ is a *constant* matrix, the state transition matrix is given as

$$\Phi(t, t_0) = e^{A(t-t_0)}. \tag{B.15}$$

When A is time-invariant, Maple V can easily calculate $\Phi(t, t_0)$ using the Maple V function **exponential(A, t-t0)**.

If $A(t)$ satisfies the *commutative* property,

$$A(t)\left(\int_{t_0}^{t} A(t_1)dt_1\right) = \left(\int_{t_0}^{t} A(t_1)dt_1\right)A(t), \tag{B.16}$$

for all t, t_0, the state transition matrix is given as

$$\Phi(t, t_0) = e^{\int_{t_0}^{t} A(t_1)dt_1}. \tag{B.17}$$

Again, Maple V can easily calculate $\Phi(t, t_0)$ using the command **exponential(map (int,msubs(A,t=t1), t1=t0..t))**.

Checking (B.16) is equivalent to checking if

$$A(t_1)A(t_2) = A(t_2)A(t_1), \ \forall t_1, t_2 \tag{B.18}$$

is satisfied [59]. The commutative condition can also be checked by decomposing $A(t)$

into the form

$$A(t) = \sum_{i=1}^{q} \alpha_i(t) A_i \tag{B.19}$$

where the $\alpha_i(t)$'s are linearly independent time functions and the A_i's are constant matrices, and then checking if the A_i's form a commuting family, that is $A_i A_j = A_j A_i$, $\forall i, j$. When the system is commutative, the state transition matrix is given by

$$\Phi(t, t_0) = \prod_{i=1}^{q} e^{A_i \beta_i(t, t_0)}, \tag{B.20}$$

where $\beta_i(t, t_0) = \int_{t_0}^{t} \alpha_i(\tau) d\tau$ [60]. The solution of (B.18) and (B.20) are equivalent; however, the solution of (B.20) may result in a simpler form. For another decomposition method, see [61]. Clearly, constant matrices and diagonal matrices both satisfy the commutative property.

If $\dot{A}(t)$ exists and there is a constant matrix A_1 which satisfies

$$\frac{d}{dt} A(t) = A_1 A(t) - A(t) A_1 \ \forall t \geq t_0, \tag{B.21}$$

then

$$\Phi(t, t_0) = e^{A_1 (t - t_0)} e^{A_2 (t - t_0)} \ \forall t \geq t_0, \tag{B.22}$$

where $A_2 = A(t_0) - A_1$ [62]. If $A(t)$ satisfies (B.21), it is said to be a member of the $\mathcal{A}_1$ class. A necessary condition to satisfy (B.21) is that the eigenvalues of $A(t)$ are time-invariant. Stability for members of $\mathcal{A}_1$ can be completely determined from the eigenvalues of A_1 and A_2.

The Kronecker product is used to transform (B.21) to

$$(A^T \otimes I - I \otimes A)\text{vec } A_1 = \text{vec } \dot{A}, \tag{B.23}$$

which can be solved for vec A_1 using the Maple V routine **linsolve**. The following procedure, **fa1(A, A0, A1, A2)**, transforms (B.21) to (B.23) and returns A_0, A_1, and A_2:

```
fa1:=
proc(A,A0,A1,A2)
local ad,Ad,a1,Al,n;
    n := linalg[rowdim](A);
    Ad := map(diff,A,t);
    ad := vec(Ad);
    Al := add(kron(linalg[transpose](A),eye(A)),-kron(eye(A),A));
    a1 := linalg[linsolve](Al,ad);
    if a1 = NULL then ERROR('no solution possible') fi;
    A0 := msubs(A,t = 0);
    A1 := invec(a1,n,n);
    A2 := evalm(A0-A1)
end
```

We note that multiple solutions are possible; the user should, if possible, choose the free parameters such that A_1 and A_2 are time-invariant. If a time-invariant solution exists, $\Phi(t, t_0)$ is given by **multiply(exponential(A1, t-t0), exponential(A2, t-t0))**.

The class $\mathcal{A}_1$ can be extended as follows [63]. If $\dot{A}(t)$ and $\dot{h}(t)$ exist, where $h(t)$ is a nonzero scalar time function, and there is a constant matrix A_1 which satisfies

$$\frac{d}{dt}\left(\frac{A(t)}{h(t)}\right) = A_1 A(t) - A(t) A_1 \; \forall t \geq t_0, \tag{B.24}$$

then

$$\Phi(t, t_0) = e^{A_1 g(t)} e^{A_2 g(t)} \; \forall t \geq t_0, \tag{B.25}$$

where $g(t) = \int_{t_0}^{t} h(\tau) d\tau$ and $A_2 = \lim_{t \to t_0} (\frac{A(t)}{h(t)}) - A_1$. If $A(t)$ satisfies (B.24), it is said to be a member of the $\mathcal{A}_h$ class. A necessary condition to satisfy (B.24) is that the eigenvalues of $A(t)$ are scalar multiples of $h(t)$. This can be checked using **eigenvals(A)**. When $h(t) = 1$ this class reverts to the $\mathcal{A}_1$ class, and when $h(t) = 1/t$ the Euler-type differential equation can be solved [64].

The following procedure, **fah(A, Ah0, A1, A2, t0, h, g)**, transforms (B.24) using the Kronecker product and returns A_1, A_2 and g:

```
fah:=
proc(A,Ah0,A1,A2,t0,h,g)
local ad,Ad,a1,Al,n,Ah,t1;
    g := eval(int(subs(t = t1,h),t1 = t0 .. t));
    n := linalg[rowdim](A);
    Ah := evalm(1/h*A);
    Ad := map(diff,Ah,t);
    ad := vec(Ad);
    Al := add(kron(linalg[transpose](A),eye(A)),-kron(eye(A),A));
    a1 := linalg[linsolve](Al,ad);
    if a1 = NULL then ERROR('no solution possible') fi;
    Ah0 := map(limit,Ah,t = t0);
    A1 := invec(a1,n,n);
    A2 := evalm(Ah0-A1)
end
```

Again, if a time-invariant solution for A_1 and A_2 exists, $\Phi(t, t_0)$ is given by **multiply(exponential(A1, g), exponential(A2, g))**.

Often situations arise where $A(t)$ is *triangular*. If $A(t) = [a_{ij}(t)]$ is lower triangular, that is $a_{ij}(t) = 0$, $\forall j > i$, then the state transition matrix has elements given by

$$\phi_{ij}(t, t_0) = \begin{cases} 0 & i < j \\ e^{\int_{t_0}^{t} a_{ii}(\tau)d\tau} & i = j \\ \int_{t_0}^{t} \phi_{ii}(t,\tau) \sum_{k=j}^{i-1} a_{ik}(\tau)\phi_{kj}(\tau, t_0)d\tau & i > j, \end{cases} \tag{B.26}$$

where $\Phi(t, t_0) = [\phi_{ij}(t, t_0)]$. The following procedure, **ltriangle(A)**, calculates $\Phi(t, t_0)$ for lower triangular matrices:

```
ltriangle:=
proc(A)
local phi,n,i,sumk,k,j,l;
    n := rowdim(A);
    phi := map(0,A);
    for i to n do
        phi[i,i] := exp(int(subs(t = t1,A[i,i]),t1 = t0 .. t));
        for j to i-1 do
            l := i-j+1;
            sumk := subs(t = t.l,sum(A[i,k]*phi[k,j],k = j .. i-1));
            phi[i,j] := phi[i,i]*
            int(subs(t = t0,subs(t0 = t.l,phi[i,i]))*sumk,t.l = t0..t)
        od
    od;
    map(simplify,phi);
    RETURN(")
end
```

If $A(t)$ is upper triangular, that is $a_{ij}(t) = 0$, $\forall i > j$, the state transition has

elements given by

$$\phi_{ij}(t,t_0) = \begin{cases} 0 & i > j \\ e^{\int_{t_0}^{t} a_{ii}(\tau)d\tau} & i = j \\ \int_{t_0}^{t} \phi_{ii}(t,\tau) \sum_{k=i+1}^{j} a_{ik}(\tau)\phi_{kj}(\tau,t_0)d\tau & i < j. \end{cases} \tag{B.27}$$

Note that a *diagonal* matrix is both lower and upper triangular. The following procedure, **utriangle(A)**, calculates $\Phi(t,t_0)$ for upper triangular matrices:

```
utriangle:=
proc(A)
local phi,n,i,sumk,k,j,l;
    n := rowdim(A);
    phi := map(0,A);
    for i from n by -1 to 1 do
        phi[i,i] := exp(int(subs(t = t1,A[i,i]),t1 = t0 .. t));
        for j from i+1 to n do
            l := j-i+1;
            sumk := subs(t = t.l,sum(A[i,k]*phi[k,j],k = i+1 .. j));
            phi[i,j] := phi[i,i]*
            int(subs(t = t0,subs(t0 = t.l,phi[i,i]))*sumk,t.l = t0..t)
        od
    od;
    map(simplify,phi);
    RETURN(")
end
```

This concept can be extended to *block triangular matrices.* Let $A(t) = [A_{ij}(t)]$, where each $A_{ij}(t)$ has dimensions $n_i \times n_j$, and $\Phi(t,t_0) = [\Phi_{ij}(t,t_0)]$, where $\Phi_{ij}(t,t_0)$ has dimensions $n_i \times n_j$. If $A(t)$ is lower block triangular, that is $A_{ij}(t) = 0$, $\forall j > i$,

then the state transition matrix has blocks given by

$$\Phi_{ij}(t,t_0) = \begin{cases} 0 & i < j \\ \Phi_{ii} & i = j \\ \int_{t_0}^{t} \Phi_{ii}(t,\tau) \sum_{k=j}^{i-1} A_{ik}(\tau)\Phi_{kj}(\tau,t_0)d\tau & i > j. \end{cases} \tag{B.28}$$

Likewise, if $A(t)$ is upper block triangular, that is $A_{ij}(t) = 0$, $\forall i > j$, then the state transition has blocks given by

$$\Phi_{ij}(t,t_0) = \begin{cases} 0 & i > j \\ \Phi_{ii}(t,t_0) & i = j \\ \int_{t_0}^{t} \Phi_{ii}(t,\tau) \sum_{k=i+1}^{j} A_{ik}(\tau)\Phi_{kj}(\tau,t_0)d\tau & i < j. \end{cases} \tag{B.29}$$

While (B.28) and (B.29) hold for all lower and upper block triangular matrices, respectively, $\Phi(t,t_0)$ can be calculated explicitly only when the $\Phi_{ii}(t,t_0)$ are known. Therefore, if every $A_{ii}(t)$ is from a class of matrices from which $\Phi_{ii}(t,t_0)$ can be calculated, $\Phi(t,t_0)$ can also be calculated. Note that a *block diagonal* matrix is both lower and upper block triangular.

Two decomposition schemes exist which expand the class of systems for which $\Phi(t,t_0)$ can be calculated. The first one can be found in [65]. Let $A(t)$ be decomposed as

$$A(t) = \sum_{i=1}^{m} A_i(t) \tag{B.30}$$

such that

$$\begin{aligned} \dot{\Phi}_i(t,0) &= F_i(t)\Phi_i(t,0), \\ \Phi_i(0,0) &= I, \end{aligned} \qquad i = 1,2,\ldots,m \tag{B.31}$$

is solvable where

$$F_i(t) = T_{i-1}^{-1}(t)A_i(t)T_{i-1}(t) \quad i = 1,2,\ldots,m, \tag{B.32}$$

and

$$\begin{aligned} T_i(t) &= \Phi_i(t,0)T_{i-1}(t) \quad i = 1,2,\ldots,m-1, \\ T_0(t) &= I. \end{aligned} \tag{B.33}$$

Then we have that

$$\Phi(t,0) = \prod_{i=1}^{m} \Phi_i(t,0). \tag{B.34}$$

The procedures discussed to this point are useful in constructing procedures for computing this decomposition. It is interesting to note that this decomposition scheme can be used to show that any arbitrary $A(t)$ can be decomposed into two normal systems [66]. Consequently, if closed form solutions exist for these normal systems, a closed form solution exists for any arbitrary $A(t)$.

A second decomposition scheme can be found in [67] and [68]. Differentiating (B.7) results in

$$\ddot{x} = (\dot{A}(t) + A^2(t))x(t). \tag{B.35}$$

If

$$\dot{A}(t) + A^2(t) = B(t)A(t), \tag{B.36}$$

for some $B(t)$, then (B.35) can be rewritten as

$$\ddot{x} = B(t)\dot{x}. \tag{B.37}$$

If a solution for $B(t)$ exists, the following procedure, **hemami(A)**, returns $B(t)$:

```
hemami:=
proc(A)
local ad,Ad,b,B,n,Al;
    n := linalg[rowdim](A);
    Ad := map(diff,A,t);
    ad := vec(evalm(Ad+A^2));
    Al := kron(linalg[transpose](A),eye(A));
    b := linalg[linsolve](Al,ad);
    if b = NULL then ERROR(`no solution possible`) fi;
    B := invec(b,n,n)
end
```

If

$$\dot{\Phi}_1(t,0) = B(t)(t)\Phi_1(t,0), \quad \Phi_1(0,0) = I \tag{B.38}$$

is solvable then

$$\Phi(t,t_0) = \left(\int_{t_0}^{t} \Phi_1(\tau,t_0)d\tau\right) A(t_0) + I. \tag{B.39}$$

Assuming **Phi1** has been found, $\Phi(t,t_0)$ is given by **add(multiply(map(int, msubs (Phi1, t=t1), t1=t0..t), msubs(A, t=t0)), eye(A))**. If (B.38) is not solvable using any of the previous procedures, (B.37) can be differentiated. If a solution to the Riccati equation $\dot{B}(t) + B^2(t) = C(t)B(t)$ exists, this procedure can be repeated until $\Phi(t,t_0)$ is found, or until the Riccati equation has no solution.

Lastly, we discuss the procedure used to verify that a given $\Phi(t, t_0)$ satisfies (B.2). From (B.2), we know that

$$\begin{aligned} A(t) &= \Phi^{-1}(t, t_0)\dot{\Phi}(t, t_0) \\ \Phi(t_0, t_0) &= I. \end{aligned} \tag{B.40}$$

The following procedure **check(Phi)** returns $\Phi^{-1}(t, t_0)\dot{\Phi}(t, t_0)$ and $\Phi(t_0, t_0)$:

```
check:=
proc(phi,t,t0)
local a,Id;
    a := multiply(map(diff,phi,t),inverse(phi));
    Id := map(limit,phi,t = t0);
    RETURN(map(simplify,Id),map(simplify,a))
end
```

In this section, several classes for which the state transition matrix can be calculated were given. These include constant matrices, matrices which satisfy the commutative condition, the $\mathcal{A}_1$ class, the $\mathcal{A}_h$ class, triangular matrices, and block triangular matrices whose block diagonal matrices are solvable. Two decomposition schemes which extend these classes were also given. Maple V procedures were given which aid in this calculation.

B.5 Examples

In this section, we will apply the Maple V procedures discussed earlier to two LTV systems. In exposition of the examples, we include diary-like output from the Maple V implementation to assist the reader in reproducing these results.

Example 1

Consider Euler's equations for the rigid body [30]:

$$\begin{aligned}
\dot{\omega}_1(t) &= \frac{I_2 - I_3}{I_1}\omega_2(t)\omega_3(t) + \frac{1}{I_1}u_1(t) \\
\dot{\omega}_2(t) &= \frac{I_3 - I_1}{I_2}\omega_3(t)\omega_1(t) + \frac{1}{I_2}u_2(t) \\
\dot{\omega}_3(t) &= \frac{I_1 - I_2}{I_1}\omega_1(t)\omega_2(t) + \frac{1}{I_3}u_3(t)
\end{aligned} \tag{B.41}$$

where the ω_i are the angular velocities, the I_i are the principal moments of inertia, and the u_i are the applied torques. If we let $I_1 = I_2 = I$ (symmetrical body) and linearize about the nominal trajectory, $\hat{u}_1 = \hat{u}_2 = \hat{u}_3 = 0$, $\hat{\omega}_1(t) = \sin(wgt)$, $\hat{\omega}_2(t) = \cos(wgt)$, and $\hat{\omega}_3(t) = w$, where $g = \frac{I-I_3}{I}$, we obtain the state equation

$$\dot{\bar{\omega}}(t) = A(t)\bar{\omega}(t) + B(t)u(t), \tag{B.42}$$

where $\bar{\omega}(t) = \omega(t) - \hat{\omega}(t)$. In this case, the matrix $A(t)$ can be found according to

```
> A:=matrix([[0,g*w,g*cos(w*g*t)],[-g*w,0,-g*sin(w*g*t)],
>    [0,0,0]]);
```

```
      [   0     g w    g cos(w g t)  ]
      [                              ]
A := [ - g w    0    - g sin(w g t) ]
      [                              ]
      [   0     0          0        ]
```

The eigenvalues of $A(t)$, given by

```
> eigenvals(A);

        2  2 1/2        2  2 1/2
  0, (- g  w )   , - (- g  w )
```

are time-invariant, which suggests that $A(t)$ may be in the $\mathcal{A}_1$ class. The procedure call **fa1(A, A0, A1, A2, 0)** returns A_1 and A_2 with A_1 given below as

```
> fa1(A,A0,A1,A2,0):
> print(A1);
  [         cos(w g t)t4   sin(w g t)t5   sin(w g t)t9                ]
  [ t5 t4   ------------ + ------------ - ------------ - g cos(w g t)]
  [               w             w              w                      ]
  [                                                                   ]
  [        -sin(w g t)t4   cos(w g t)t5   cos(w g t)t9                ]
  [-t4 t5   ------------ + ------------ - ------------ + g sin(w g t)]
  [              w             w              w                      ]
  [                                                                   ]
  [ 0  0                                t9                            ]
```

where t_4, t_5, and t_9 are parameters to be specified. These parameters are chosen such that A_1 and A_2 are time-invariant, according to

```
> A1:=msubs(A1,t4=w*g,t5=0,t9=0);

        [   0    g w  0 ]
        [               ]
  A1 := [ - g w   0   0 ]
        [               ]
        [   0     0   0 ]


> A2:=msubs(A2,t4=w*g,t5=0,t9=0);
```

```
       [ 0  0  g ]
       [         ]
  A2 := [ 0  0  0 ]
       [         ]
       [ 0  0  0 ]
```

Now $\Phi(t,0)$ can be calculated as in (B.22) using

```
> phi:=map(simplify,multiply(exponential(A1,t),
>      exponential(A2,t)));

       [cos(wgt)  sin(wgt)  cos(wgt) t g]
       [                                ]
  phi:=[-sin(wgt) cos(wgt) -sin(wgt) g t]
       [                                ]
       [   0         0          1       ]
```

Finally, checking our results (according to (B.40))

```
> check(phi,t,0);

  [1  0  0]   [  0    g w   g cos(w g t) ]
  [       ]   [                          ]
  [0  1  0],  [- g w   0   - g sin(w g t)]
  [       ]   [                          ]
  [0  0  1]   [  0     0          0      ]
```

The identity matrix and the matrix $A(t)$ are returned as expected. Note that by examining $\Phi(t,0)$, we see that the nominal trajectory is unstable (see Table 7). The elements $\phi_{13}(t,0)$ and $\phi_{23}(t,0)$ of $\Phi(t,0)$ are clearly unbounded.

Example 2

Again consider Euler's equations for a symmetrical body, that is $I_1 = I_2 = I$. Setting $u_1 = u_2 = 0$, the equations can be written in the form of a quasi-linear

parameter varying system [69]:

$$\frac{d}{dt}\begin{bmatrix}\omega_1(t)\\ \omega_2(t)\\ z(t)\end{bmatrix} = \begin{bmatrix}0 & gz(t) & 0\\ -gz(t) & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\omega_1(t)\\ \omega_2(t)\\ z(t)\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{1}{I_3}\end{bmatrix}u_3(t) \tag{B.43}$$

where $z(t) = \omega_3(t)$. A linear parameter varying (LPV) system depends on a time-varying parameter rather than explicitly on time. This parameter is known at time t, but not necessarily *a priori* [69].

The control structure $u_3(t) = -I_3k(z - z_{ref})$ is proposed to drive $z(t)$ to z_{ref} and maintain stability of the system where k is a positive constant. Substituting $u_3(t)$ into (B.43), we obtain the following closed loop matrix for $A(t)$,

```
> A:=matrix([[0,g*z(t),0],[-g*z(t),0,0],[0,0,-k]]);

      [     0      g z(t)    0  ]
      [                         ]
  A := [ - g z(t)     0      0  ]
      [                         ]
      [     0         0    - k ]
```

Taking the Peano-Baker series to the 5th order results in

```
> peano(A,5,0);
          2          4                  3
  [1-1/2 %1  + 1/24 %1 ,     %2 - 1/6 %2 ,     0]

           3                 2          4
  [%1-1/6 %1 ,     1 - 1/2 %2  + 1/24 %2 ,     0]

                          2 2     3 3      4 4
  [0,     0,     1-k t+1/2 k t -1/6k t +1/24k t ]
```

```
                  t
                  /
                 |
%1 :=            |  - g z(t1) dt1
                 |
                /
               0

                  t
                  /
                 |
%2 :=            |  g z(t1) dt1
                 |
                /
               0
```

By examining the Peano-Baker series and checking a table of Taylor series, it is proposed that $\Phi(t,0)$ is of the following form:

```
> h:=int(g*z(t1),t1=0..t):
> phi:=matrix([[cos(h),sin(h),0],[-sin(h),cos(h),0],
>      [0,0,exp(-k*t)]]);

       [  cos(%1)   sin(%1)       0     ]
       [                                ]
  phi:=[ - sin(%1)  cos(%1)       0     ]
       [                                ]
       [     0         0     exp(- k t) ]

                  t
                  /
                 |
%1 :=            |  g z(t1) dt1
                 |
                /
               0
```

Checking our proposed $\Phi(t,0)$,

```
> check(phi,t,0);

  [ 1  0  0 ]   [     0     g z(t)   0  ]
  [         ]   [                      ]
  [ 0  1  0 ],  [ - g z(t)     0     0  ]
  [         ]   [                      ]
  [ 0  0  1 ]   [     0        0    - k ]
```

where the identity matrix and $A(t)$ are returned as expected. By examining $\Phi(t,0)$, it can be seen that ω_1 and ω_2 are stable and that z is asymptotically stable as desired.

B.6 Summary

In this appendix, we have examined how Maple V could be used to calculate the state transition matrix for several classes of LTV systems. We motivated the problem by arguing that the state transition matrix is important for understanding stability, controllability, and observability of LTV systems. Classes of systems for which the state transition matrix can be calculated include constant matrices, matrices which satisfy a commutative condition, the $\mathcal{A}_1$ class, the $\mathcal{A}_h$ class, triangular matrices, and block triangular matrices whose block diagonal matrices are solvable. Maple V procedures to aid in this calculation were given. Finally, examples showing how the state transition matrix can be calculated and used for analysis and control of LTV systems were discussed. Clearly, this is one of many applications where symbolic software packages can be used to aid the controls engineer.

Addendum

In the following we list several well known Theorems (involving the state transition matrix), without proof, which are referenced in the text.

1. Stability [56]:

   Theorem 10 *Every equilibrium state of (B.7) is* stable in the sense of Lyapunov *at* t_0 *if and only if there exists some constant* k *which depends on* t_0 *such that* $\|\Phi(t,t_0)\| \leq k < \infty$ *for all* $t \geq t_0$. *If* k *is independent of* t_0, *it is* uniformly stable in the sense of Lyapunov.

2. Asymptotic Stability [56]:

   Theorem 11 *The zero state of (B.7) is* asymptotically stable *at* t_0 *if and only if* $\|\Phi(t,t_0)\| \leq k(t_0) < \infty$ *and* $\|\Phi(t,t_0)\| \to 0$ *as* $t \to \infty$. *The zero state is* uniformly asymptotically stable *over* $[0,\infty)$ *if and only if there exist positive numbers* k_1 *and* k_2 *such that* $\|\Phi(t,t_0)\| \leq k_1 e^{-k_2(t-t_0)}$ *for any* $t_0 \geq 0$ *and for all* $t \geq t_0$.

3. Controllability [30]:

Theorem 12 *The LTV system (B.1) is* controllable *on* $[t_0, t_1]$ *if and only if the controllability grammian*

$$W(t_0, t_1) = \int_{t_0}^{t_1} \Phi(t_0, t) B(t) B^T(t) \Phi^T(t_0, t) dt \tag{B.44}$$

is invertible.

4. Observability [30]:

Theorem 13 *The LTV system (B.1) is* observable *on* $[t_0, t_1]$ *if and only if the observability grammian*

$$M(t_0, t_1) = \int_{t_0}^{t_1} \Phi^T(t_0, t) C^T(t) C(t) \Phi(t_0, t) dt \tag{B.45}$$

is invertible.

BIBLIOGRAPHY

[1] M. F. Cheung, S. Yurkovich, and K. Passino, "An optimal volume ellipsoid algorithm for parameter set estimation," *IEEE Transactions on Automatic Control*, vol. 38, pp. 1292–1296, August 1993.

[2] E. Walter and H. Piet-Lahanier, "Exact recursive polyhedral description of the feasible parameter set for bounded-error models," *IEEE Transactions on Automatic Control*, vol. 34, pp. 911–915, August 1989.

[3] E. Fogel and Y. F. Huang, "On the value of information in system identification: Bounded noise case," *Automatica*, vol. 18, no. 2, pp. 229–238, 1982.

[4] S. Dasgupta and Y.-F. Huang, "Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise," *IEEE Transactions on Information Theory*, vol. 33, pp. 383–392, May 1987.

[5] J. R. Deller, Jr., M. Nayeri, and S. F. Odeh, "Least square identification with error bounds for real-time signal processing and control," *Proceedings of the IEEE*, vol. 81, pp. 815–849, June 1993.

[6] G. Belforte and B. Bona, "An improved parameter identification algorithm for signals with unknown but bounded errors," in *IFAC/IFORS Symposium on Identification and System Parameter Estimation*, (York), pp. 1507–1511, 1985.

[7] G. Belforte, B. Bona, and V. Cerone, "Parameter estimation algorithms for a set-membership description of uncertainity," *Automatica*, vol. 26, no. 887-898, pp. 887–898, 1990.

[8] R. G. Balnd, D. Goldfarb, and M. J. Todd, "The ellipsoid method: A survey," *Operations Research*, vol. 29, pp. 1039–1091, 1981.

[9] J. M. Gassman and S. Yurkovich, "An ellipsoid algorithm for parameter set estimation," in *Proceedings of the IEEE Conference on Control Applications*, (Dayton, OH), pp. 835–840, September 1992.

[10] M.-F. Cheung and S. Yurkovich, "Parametric subsystem set estimation for interconnected systems," *International Journal of Control*, vol. 59, pp. 499–514, February 1994.

[11] S. L. DeVilbiss and S. Yurkovich, "An input synthesis procedure for parameter set identification," in *Proceedings of the American Control Conference*, (Seattle, WA), pp. 2564–2565, June 1995.

[12] H. Piet-Lahanier and E. Walter, "Bounded-error tracking of time-varying parameters," in *Proceedings of the IEEE Conference on Decision and Control*, (Tucson, Arizona), pp. 66–67, December 1992.

[13] J. P. Norton and S. H. Mo, "Parameter bounding for time-varying systems," *Mathematics and Computers in Simulations*, vol. 32, pp. 527–534, 1990.

[14] A. K. Rao and Y.-F. Huang, "Tracking characteristics of and OBE parameter-estimation algorithm," *IEEE Transactions on Signal Processing*, vol. 41, pp. 1140–1148, March 1993.

[15] F. L. Chernous'ko, "Optimal guaranteed estimates of indeterminacies with the aid of ellipsoids. I," *Engineering Cybernetics*, vol. 18, May-June 1980.

[16] P. Dorato, *Robust Control.* New York, New York: IEEE Press, 1987.

[17] R. L. Kosut, "Adaptive control via parameter set estimation," *International Journal of Adaptive Control and Signal Processing*, vol. 2, pp. 371–399, 1988.

[18] M. K. Lau, S. Boyd, R. L. Kosut, and G. F. Franklin, "A robust control design for FIR plants with parameter set uncertainty," in *Proceedings of American Control Conference*, (Boston, MA), pp. 83–88, June 1991.

[19] M. K. Lau, S. Boyd, R. L. Kosut, and G. F. Franklin, "Robust control design for ellipsoid plant set," in *Proceedings of American Control Conference*, (Boston, MA), pp. 83–88, June 1991.

[20] S. M. Veres and J. P. Norton, "Bound-based worst-case self-tuning controllers," in *IFAC/IFORS Symposium on Identification and System Parameter Estimation*, (Budapest, Hungary), pp. 773–778, July 1991.

[21] S. L. DeVilbiss, *System Identification for H_∞ Robust Control Design.* PhD thesis, The Ohio State University, 1994.

[22] R. L. Kosut, M. K. Lau, and S. P. Boyd, "Set membership identification of systems with parametric and nonparametric uncertainty," *IEEE Transactions on Automatic Control*, vol. 37, pp. 929–941, July 1992.

[23] P. C. Young and K. J. Beven, "Data-based mechanistic modeling and the rainfall-flow non-linearity," *Environmetrics*, vol. 5, pp. 335–363, 1994.

[24] A. A. Zhukov and V. D. Furasov, "Ellipsoidal approximation and estimation of discrete system states," *Soviet Journal of Computer and System Sciences*, vol. 29, pp. 1–9, January 1991.

[25] G. Strang, *Linear Algebra and its Applications.* Harcourt Brace Jovanovich, third ed., 1988.

[26] F. C. Schweppe, *Uncertain Dynamic Systems.* Englewood Cliffs, New Jersey: Prentice Hall, 1973.

[27] R. Bellman, *Introduction to Matrix Analysis.* New York, New York: McGraw-Hill Book Company, 1970.

[28] R. A. Horn and C. R. Johnson, *Matrix Analysis.* New York: Cambridge University Press, 1990.

[29] J. M. Gassman, "An ellipsoid algorithm for parameter set estimation," Master's thesis, The Ohio State University, 1992.

[30] W. J. Rugh, *Linear System Theory.* Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[31] K. Ogata, *Discrete-time Control Systems.* Englewood Cliffs, New Jersey: Prentice Hall, 1987.

[32] G. H. Golub and C. F. Van Loan, *Matrix Computation.* Baltimore, Maryland: The Johns Hopkins University Press, 1993.

[33] P. B. Deshpande, *Distillation Dynamics and Control.* Research Triangle Park, NC: Instrument Society of America, 1985.

[34] T. Söderstörm and P. Stoica, *System Identification.* New York: Prentice Hall, 1989.

[35] W. Brogan, *Modern Control Theory.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1985.

[36] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation.* San Diego, California: Academic Press, 1977.

[37] G. L. Besnerais and Y. Goussard, "Improved square-root forms of fast linear least squares estimation algorithms," *IEEE Transactions on Signal Processing*, vol. 41, pp. 1415–1421, March 1993.

[38] P. Park and T. Kailath, "New square-root algorithms for kalman filtering," *IEEE Transactions on Automatic Control*, vol. 40, pp. 895–899, May 1995.

[39] S. L. DeVilbiss and S. Yurkovich, "Input synthesis for a parameter set estimation algorithm," in *Proceedings of the 26th Southeastern Symposium on System Theory*, (Athens, OH), pp. 507–511, March 1994.

[40] L. Pronzato and E. Walter, "Sequential experimental design for parameter bounding," in *Proceedings of the First European Control Conference*, (Grenoble, France), pp. 1181–1186, July 1991.

[41] P. Sadegh, H. Madsen, and J. Holst, "Optimal input design for fault detection and diagnosis," in *Proceedings of the American Control Conference*, (Seattle, WA), pp. 1147–1148, June 1995.

[42] K. J. Åström and B. Wittenmark, *Adaptive Control.* Addison Wesley, 1989.

[43] A. S. Wilsky, "A survey of design methods for failure detection in dynamics systems," *Automatica*, vol. 12, pp. 601–611, 1976.

[44] R. Isermann, "Process fault detection based on modeling and estimation methods–a survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.

[45] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy–a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[46] A. Zolghadri, B. Bergeon, and M. Monsion, "A two-ellipsoid overlap test for on-line failure detection," *Automatica*, vol. 29, November 1993.

[47] Y. F. Huang and J. J. R. Deller, "On the tracking capabilities of optimal bounding ellipsoid algorithms," in *Thirtieth Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, Illinois), pp. 50–59, October 1992.

[48] M. Bodson, "Identification with modeling uncertainty and reconfigurable control," in *Proceedings of the IEEE Conference on Decision and Control*, (San Antonio, TX), pp. 2242–2247, December 1993.

[49] L. Ljung, "Issues in system identification," *IEEE Control Systems*, vol. 11, pp. 25–29, January 1991.

[50] J. Watkins, "Parameter set estimation for time varying systems." Ph.D. Dissertation Proposal: The Ohio State University, 1993.

[51] G. C. Goodwin, R. H. Middleton, and V. Poor, "High-speed digital signal processing and control," *Proceedings of the IEEE*, vol. 80, pp. 240–259, February 1992.

[52] G. Li and M. Gevers, "Comparative study of finite wordlength effects in shift and delta operator parameterizations," *IEEE Transactions on Automatic Control*, vol. 38, pp. 803–807, May 1993.

[53] B. M. Ninness and G. C. Goodwin, "The relationship between discrete time and continuous time linear estimation," in *Identification of Continuous-Time Systems* (N. K. Sinha and G. P. Rao, eds.), pp. 79–122, Kluwer Academic Press, 1991.

[54] H. A. Barker, I. T. Harvey, and P. Townsend, "Symbolic mathematics in control system analysis and design," *Transactions of the Institute of Measurement and Control*, vol. 15, no. 2, pp. 59–68, 1993.

[55] T. Lee and G. R. Heppler, "Algebra systems for enhancing the learning environment for control systems," in *1990 ASEE Annual Conference Proceedings*, pp. 113–118, 1990.

[56] C.-T. Chen, *Linear System Theory and Design.* Orlando, Florida: Holt, Rinehart and Winston, Inc., 1984.

[57] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt, *First Leaves: A Tutorial Introduction to Maple V.* New York, New York: Springer-Verlag, 1992.

[58] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis.* New York: Cambridge University Press, 1991.

[59] B. A. Kinariwala, "Analysis of time varying networks," in *IRE Int. Conv. Record, Part 4*, pp. 268–276, 1961.

[60] M.-Y. Wu and A. Sherif, "On the commutative class of linear time-varying systems," *International Journal of Control*, vol. 23, no. 3, pp. 433–444, 1976.

[61] L. A. Zadeh and C. A. Desoer, *Linear system theory; the state space approach.* New York: McGraw-Hill, 1963.

[62] M. Y. Wu, "A new method of computing the state transition matrix of linear time-varying systems," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, (San Francisco, CA), pp. 269–272, April 1974.

[63] M.-Y. Wu, "Solution of certain classes of linear time-varying systems," *International Journal of Control*, vol. 31, no. 1, pp. 11–20, 1980.

[64] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations.* New York: McGraw-Hill, 1955.

[65] M.-Y. Wu, "A successive decomposition method for the solution of linear time-varying systems," *International Journal of Control*, vol. 33, no. 1, pp. 181–186, 1981.

[66] J. Zhu and C. D. Johnson, "New results in the reduction of linear time-varying dynamical systems," *Siam Journal of Control and Optimization*, vol. 27, pp. 476–494, May 1989.

[67] H. Hemami, "On a class of time-varying systems," *IEEE Transactions on Automatic Control*, vol. AC-13, p. 589, October 1968.

[68] T. L. Niemeyer, "Exact state transition matrix solutions to time-varying systems," Master's thesis, The Ohio State University, 1969.

[69] J. S. Shamma and J. R. Cloutier, "Trajectory scheduled missile autopilot design," in *Conference on Control Applications*, (Dayton, OH), pp. 237–242, September 1992.