MICHIGAN STATE UNIVERSITY

DEPARTMENT OF ELECTRICAL ENGINEERING
speech processing laboratory
adaptive signal processing laboratory

1991 YEAR END REPORT OF PROGRESS

Theoretical Issues in Adaptive Set-Membership-Based Signal Processing

Grant No.:	N00014-91-J-1329
R&T Project:	4148126—01
Submission Date:	January 20, 1992
Scientific Officer:	Dr. R.N. Madan, Electronics Division
Distribution:	Per Attachment 2 of Notification Letter
Co-Principal Investigators:	John R. Deller, Professor
	Majid Nayeri, Assistant Professor

Michigan State University
Department of Electrical Engineering / 260EB
East Lansing, MI 48824-1226

FAX: (517) 353-1980
VOICE: (517) 353-8840 (Deller)
(517) 353-1857 (Nayeri)
EMAIL: deller@ee.msu.edu
majid@ee.msu.edu

List of Contents

Statement A per telecon
Dr. Rabinder Maden
ONR/Code 1114
Arlington, VA 22217-5000
NWW 2/3/92

Project Summary and Description of Report Contents

The included materials comprise Report of Progress covering the period 4/91 - 12/91 for the work being conducted under ONR Grant No. N00014-91-J-1329. The project title is "Theoretical Issues in Adaptive Set-Membership-Based Signal Processing." The research is being conducted in the Speech Processing Laboratory and the Adaptive Signal Processing Laboratory under the direction of Co-Principal Investigators John R. Deller, Professor of Electrical Engineering and Majid Nayeri, Assistant Professor of Electrical Engineering. The following graduate students are conducting research directly related to the project:

1. **Steve M.S. Liu** (Ph.D. Candidate) — Convergent unified optimal bounding ellipsoid (UOBE) algorithms and applications to speech recognition.
2. **Shawn D. Hunt** (Ph.D. Candidate) — Efficient neural network learning algorithms with selective updating. Completion of Ph.D. anticipated 6/92.
3. **Y.B. Lee** (Ph.D. Candidate) — Novel set-membership-based algorithms for neural network learning.
4. **Marwan M. Krunz** (M.S. Candidate) — Convergence and colored noise issues in UOBE algorithms. UOBE optimization involving multiple data weights. Simulation software development. Completion of M.S. degree anticipated 6/92. Will enter Ph.D. program.

Related work is being supported by the NSF under Grant No. MIP-9016734 entitled "Applications and Performance Evaluation of Set-Membership Algorithms for Signals in $\mathcal{C}^n$." All work cited below acknowledges joint sponsorship.

The general purpose of this research is the development and exploration of new set-membership-based algorithms for adaptive identification of parametric signal and system models. We are pleased to report progress in several important aspects, both theoretical and applied, of this general scope. The report consists of several preprints of papers in review by repected journals, published and preprinted conference papers, and some other supporting material. A clear understanding of our progress is inherent in the discussion of each item in the following. These discussions are meant to illuminate the directions, rationale, and achievements of our research, with the technical details left to the papers. The items appearing in the following are grouped into papers written for journals, followed by conference papers, descriptions of dissertations in preparation, then documents showing further evidence of research progress. Within each group, the items appear in chronological order. The contents are as follows:

1. JOURNAL PAPER PREPRINTS, DRAFT MANUSCRIPTS, AND SUMMARIES

(a) J.R. Deller and S.F. Odeh, "Adaptive set-membership identification in $\mathcal{O}(m)$ time for linear in parameters models," *IEEE Transactions on Signal Processing* (revision submitted 10/91). [Preprint]

This paper is a revision of an earlier submission which was based principally upon the Ph.D. dissertation of Souheil F. Odeh. The revision includes many new results obtained under ONR sponsorship. Reported are four significant contributions to the field:

- A generalization of all fundamental results in Optimal Bounding Ellipsoid (OBE) processing to the case of complex signal MIMO models. Such models occur in many important problems including, for example, adaptive beamforming and neural network learning.
- A class of *explicitly* adaptive OBE algorithms appears in a journal paper for the first time.
- A suboptimal test for innovation is developed which leads to a class of OBE algorithms which empirically perform as well as those employing optimal checking. This check admits $\mathcal{O}(m)$ computational complexity which represents a square root factor improvement over optimal methods, as well as RLS.
- Compact parallel architectures are developed which can be used for running both optimal and suboptimal algorithms at $\mathcal{O}(m)$ expense.

(b) **J.R. Deller, M. Nayeri, and S.F. Odeh, "System identification using set-membership-based signal processing,"** *Proceedings of the IEEE* **(submitted 12/91 by invitation in response to paper proposal). [Letter of invitation included with preprint]**

In response to a proposal to the *Proceedings*, submission of this paper was encouraged by the editors as the prefatory letters indicate. The paper is expository, reviewing the general field of set-membership based identification algorithms, then focusing on the OBE algorithms which are currently of most interest to the signal processing community. Reviewed are adaptive and "nonadpative" algorithms, efficient algorithms with suboptimal data checking, and parallel architectures for implementation. In addition to the tutorial value of this paper, current research sponsored in part by the ONR grant has lead to a unified framework into which all OBE algorithms may be placed. The paper discusses the field from this point of view, and, in an appendix, provides unified and rigorous theoretical developments which underlie all major developments in the OBE field. These developments are scattered throughout the literature, and in some cases are absent, incomplete, or misunderstood. Both the novice reader and the expert should benefit from this work.

(c) **S.D. Hunt and J.R. Deller, " 'Linearized' alternatives to back-propagation based on recursive *QR* decomposition,"** *IEEE Transactions on Neural Networks* **(submitted 8/91). [Preprint]**

The class of learning methods presented in this paper were developed *en route* to the application of set-membership principles to neural network training. The algorithm is based upon linearization of the dynamics of a feedforward neural network based on error surface analysis, followed by training using a ***QR*** decomposition version of the RLS algorithm. The algorithm can be used to train networks "node-wise" (all weights connected to a node updated simultaneously) or "layer-wise," and, in some cases all weights of the network can be updated simultaneously. The node-wise case turns out to be *theoretically* similar to a method developed by Azimi-Sadjadi *et al.* (A-S), but the ***QR*** implementation renders the present algorithm vastly superior in terms of numbers and speeds of convergences. The reported method, as well as the A-S method, outperform conventional back-propagation.

(d) **J.R. Deller, M. Nayeri, and M.S. Liu, "Connections between the Fogel-Huang and Dasgupta-Huang optimal bounding ellipsoid algorithms" (in preparation, tenatively for** *Automatica*) **[Draft manuscript included].**

The Fogel-Huang OBE (F-H OBE) algorithm is attractive in its clear interpretability, but in spite of statements to the contrary in the literature, it does not have proven convergence properties. On the other hand, the Dasgupta-Huang OBE (D-H OBE) is desirable in its proven convergence, but its controversial optimization criterion is not amenable to clear intepretation of the method's operation. In our work related to the *Proceedings* paper above, intriguing connections between D-H OBE and F-H OBE (in fact, between D-H and a broadly genearlized version of F-H) were discovered. These connections are apparently unknown to the research community, and are reported in this paper. It is suggested that these findings could ultimately lead to an OBE algorithm with the desirable properties of both methods.

(e) **M. Nayeri, J.R. Deller, and M.S. Liu, "A converging optimal bounding ellipsoid algorithm with volume minimization (tentative title)" (in preparation, tentatively for** *Automatica* **(special issue on signal processing)). [Summary paper included]**

We found the OBE algorithm alluded to in the discussion of the last paper. It is quite possible that this will be a landmark paper which will have the same impact on the field as the original F-H OBE and subsequent D-H OBE.

(f) **J.R. Deller and M. Nayeri, "Unifying the landmark developments in optimal bounding ellipsoid processing,"** *International Journal of Adaptive Control and Signal Processing* **(in planning in response to recent invitation). [Letter of invitation & planning paper included]**

The Guest Editor of this special issue has written that papers with tutorial content are especially welcome. This paper will tie together in one source several of the key unifying themes mentioned in the descriptions above. Accordingly, it will decribe the general unifying themes, and lead the reader to sources of information on rigorous theoretical details. In particular, we will develope the "generic" *Unified Optimal Bounding Ellipsoid (UOBE)* algorithm, and show how all reported algorithms, both adaptive and nonadaptive, are instances of UOBE. The interesting connections between F-H OBE and D-H OBE described in paper 1d above can be presented in this framework. Finally, the algorithm which combines the desirable features of these two "landmark" algorithms will be described. This paper will reach a large population of researchers in Europe whose work is system and control-oriented and who might not be as familiar with the signal processing literature.

2. PUBLISHED AND PREPRINTED CONFERENCE PAPERS

(g) **M. Nayeri, J.R. Deller, and M.M. Krunz, "Convergence and colored noise issues in bounding ellipsoid identification,"** *Proceedings of ICASSP '92*, **San Franscisco, March 1992 (to appear). [Preprint]**

This paper presents the following new results a discusssion of almost sure convergence of the UOBE *estimator* (ellipsoid center) under ordinary "white noise" conditions on the model disturbances, then presents the following new results concerning the ellipsoid behavior under various noise conditions:

- With white noise disturbances, UOBE algorithms involve ellipsoidal bounding sets which converge in some unspecified way to some unspecified "size." This result represents the first report in the literature of a covergence result for a "non-D-H" algorithm. The original F-H OBE paper has been misinterpreted to mean that the ellipsoid converges to a point.
- With colored noise inputs, the limiting ellipsoid must be a nontrivial set. Empirical evidence suggests that the true parameters lie on the boundary of this limiting set.
- Arguments are made in support of the idea that the ellipsoid may collapse into a subspace of the parameter space (thereby diminishing the volume of the ellipsoid to zero without its being reduced to a point) if and only if the input is not persistently exciting.

(h) **J.R. Deller and S.F. Odeh, "SM-WRLS algorithms with an efficient test for innovation,"** *Proceedings of the 9th IFAC / IFORS Symposium on Identification and System Parameter Identification*, **vol. 2, pp. 1044–1049, July 1991 (written and presented by invitation). [Reprint]**

This paper presents some of the ideas concerning suboptimal testing cited in the description of paper 1a.

(i) **J.R. Deller and S.D. Hunt, "A simple 'linearized' learning algorithm which outperforms back-propagation" (submitted to International Joint Conference on Neural Networks, 1/92). [Preprint]**

This paper presents some of the key developments of the algorithm cited in the description of paper 1c.

3. BOOK ACKNOWLEDGING ONR SUPPORT

(j) **J.R. Deller, J.G. Proakis, and J.H.L. Hansen,** *Discrete Time Processing of Speech Signals*, **New York: Macmillan (writing completed, anticipated publication in late 1992). [Table of contents included]**

This book will acknowledge ONR research support during the period of authorship.

Adaptive Set-Membership Identification in $\mathcal{O}(m)$ Time for Linear-in-Parameters Models

J.R. Deller, Jr. and S.F. Odeh[1]

Michigan State University
Department of Electrical Engineering / 260 EB
Control, Systems, and Signal Processing Group: Speech Processing Laboratory
East Lansing, MI 48824-1226

(517) 353-8840 arpanet: deller@ee.msu.edu bitnet: deller@msuegr

Revision of October 25, 1991

EDICS Category: 5.2

Abstract

This paper describes some fundamental contributions to the theory and applicability of optimal bounding ellipsoid (OBE) algorithms for signal processing. All reported OBE algorithms are placed in a general framework which fruitfully demonstrates the relationship between the set-membership principles and least square error identification. Within this framework, flexible measures for adding explicit adaptation capability are formulated and demonstrated through simulation. Computational complexity analysis of OBE algorithms reveals that they is of $\mathcal{O}(m^2)$ complexity per data sample with m the number of parameters identified, in spite of their well-known propensity toward highly-selective updating. Two very different approaches are described for rendering the a specific OBE algorithm, the set-membership weighted recursive least squares algorithm, of $\mathcal{O}(m)$ complexity. The first approach involves an algorithmic solution in which a suboptimal test for innovation is employed. The performance is demonstrated through simulation. The second method is an architectural approach in which complexity is reduced through parallel computation.

Acknowledgements

This work was supported in part by the National Science Foundation under Grant No. MIP-9016734 and the Office of Naval Research under Contract No. N00014-91-J-1329.

[1] S.F. Odeh is currently with the Department of Electrical Engineering, University of Jordan, Amman.

List of Figures

List of Tables

List of Frequently Used Acronyms and Abbreviations

1. AR - autoregressive
2. ARX - autoregressive, exogenous input
3. BE - bounded error
4. LP - linear-in-parameters (model)
5. LSE - least square error
6. MIL-WRLS - WRLS algorithm based on matrix inversion lemma
7. OBE - optimal bounding ellipsoid (algorithm)
8. QR-WRLS - WRLS algorithm based on QR decomposition
9. RLS - recursive least squares (algorithm)
10. SM - set membership
11. SM-WRLS - set membership weighted recursive least squares (algorithm)
12. WRLS - weighted recursive least squares (algorithm)

1 Introduction

Set-membership (SM) identification of parametric systems is concerned with the computational description of *feasible sets* of solutions which are consistent with the measurements and the modelling assumptions. SM algorithms have been the subject of intense research effort in recent years and and many approaches has been explored. The papers in [1] and [2] provide a broad and current overview of the area. In particular, comprehensive reviews of the field with extensive reference lists are found in papers by Walter and Piet-Lahanier [3] and by Milanese and Vicino [4]. An extensive list of application examples with references is also given in the Milanese paper. A tutorial on the principal algorithm of interest in this paper, the so-called *set-membership weighted recursive least squares (SM-WRLS)* algorithm, is found in [5].

One class of SM methods, the *optimal bounding ellipsoid (OBE)* algorithms[2], is of particular interest to the signal community since it represents a merging of the SM approach and widely used *least square error (LSE)* procedures for identifying linear models. The benefits of combining SM considerations (when they are known) with LSE processing are twofold: First, the SM information provides a feasible set of solutions which complements the unique LSE estimate. This feasible set can help to compensate for the restrictive nature of the assumptions placed upon the LSE model. Secondly, as we demonstrate in this paper, SM knowledge can greatly improve the efficiency of LSE identification.

Two aspects of OBE processing are treated in this paper. In a general way, it is shown that all reported OBE algorithms can be placed into a unified framework which is clearly related to conventional LSE processing. This framework will embrace *explicitly adaptive* OBE algorithms which will be demonstrated as a first major contribution of the paper. The second, and more extensive, aspect of this paper is concerned with the computational efficiency of OBE algorithms. OBE algorithms (both nonadaptive and adaptive) entail an interesting data selection procedure which typically discards 70 – 95% of the incoming data. The basis for this selective updating is a determination of whether the incoming datum is "informative" in the sense of refining the feasibility set. The selective updating procedure, however, does *not* imply a similar reduction in computational load, since the effort of checking for innovation in the data is approximately as expensive as the updating itself. In either case, the processing requires $\mathcal{O}(m^2)$ floating point operations[3] per incoming datum, where m represents the number of parameters to be estimated.

[2] The original algorithm in this class due to Fogel and Huang [6] was called simply "OBE". We use this term to indicate the broader class of similar algorithms. The SM-WRLS algorithm will be seen below to be a specific type of OBE algorithm in this broader sense.

[3] One flop is taken to be a multiplication plus an addition operation.

This point has not been clearly brought out in the literature. A second focus of this paper is to demonstrate two very different methods for making a specific OBE algorithm run in $\mathcal{O}(m)$ time. The first solution is algorithmic, while the second is architectural. The ability to execute this interesting method in $\mathcal{O}(m)$ time makes it highly competitive with conventional identification techniques (especially *recursive least squares (RLS)*) which typically require $\mathcal{O}(m^2)$ flops per point.

2 An Adaptive SM-WRLS Algorithm

2.1 The Model and the LSE Identification Problem

The basic identification problem is as follows: We observe a system which is generating output sequence $y(\cdot)$ in response to input sequence $u(\cdot)$. Both input and output sequences are measurable, and $u(\cdot)$ is assumed to be a realization of a stationary, ergodic random process. The system is governed by a "true" model of form

$$y(n) = \boldsymbol{\theta}_*^T \boldsymbol{x}(n) + \varepsilon_*(n) \tag{1}$$

in which $\boldsymbol{x}(n)$ is some m-vector of functions of p lags of $y(\cdot)$ at time n, and q lags plus the present value of $u(\cdot)$, and where, $\varepsilon_*(\cdot)$ is the realization of a zero-mean, *white noise error sequence*. The error sequence is not measurable and the "true" parameters $\boldsymbol{\theta}_* \in \mathcal{R}^m$ are unknown. At time n we wish to use the observed data on $t \in [1, n]$ to deduce an estimated model which is similar in form to (1),

$$y(n) = \boldsymbol{\theta}^T(n)\boldsymbol{x}(n) + \varepsilon(n, \boldsymbol{\theta}(n)). \tag{2}$$

In the following, the identified parameter vector will be unique for each n (e.g. [7]), but will change at every step. Hence, we index the parameter estimate by n. The error sequence will depend on the choice of parameters, and we explicitly show this dependence. Neglecting the error term, this model exhibits only linear functional dependence upon the parameter vector and has been called a *linear in unknown coefficients* (e.g. [8]) or *linear-in-parameters (LP)* model (e.g. [3]). Special cases of the LP model of (2) are the *autoregressive-exogenous input (ARX)* and *autoregressive (AR)* models (e.g. [9] - [11]). For a current overview of methods that deal with nonlinear models, the reader is referred to [3],[4].

In particular, we desire the weighted LSE model for which $\theta(n)$ minimizes $\xi(n) = \frac{1}{n}\sum_{t=1}^{n} \lambda_n(t)\varepsilon^2(t, \boldsymbol{\theta}(n))$, where $\lambda_n(\cdot)$ is a sequence of nonnegative weights which may depend on n. $\boldsymbol{\theta}(n)$ can be found as the solution of the following classical linear algebra problem (e.g. [7]): Given data (or a system of observations) on the interval $t \in [1, n]$ $(n \geq m)$, and some set of error minimization weights, say

$\{\lambda_n(t),\ t = 1, 2, \ldots, n\}$, form the overdetermined system of equations

$$\boldsymbol{X}(n)\boldsymbol{\nu} = \boldsymbol{y}(n)\ , \tag{3}$$

and find the LS estimate, $\boldsymbol{\theta}(n)$, for the vector $\boldsymbol{\nu}$. $\boldsymbol{X}(n)$ is the $m \times n$ matrix with i^{th} row $\sqrt{\lambda_n(i)}\boldsymbol{x}^T(i)$ and $\bar{\boldsymbol{y}}(n)$ is the n-vector with i^{th} element $\sqrt{\lambda_n(i)}y(i)$. Because of this interpretation, the pair $(y(n), \boldsymbol{x}(n))$ could appropriately be called an *equation* in many contexts in the following. This term is not always satisfactory, however. Whereas the term "datum" is inappropriate to describe $(y(n), \boldsymbol{x}(n))$, and "data" can be misleading, we will frequently refer to $(y(n), \boldsymbol{x}(n))$ as the *data set* at time n. The expression "per n" should be interpreted to mean "per data set."

In principle, the LSE solution is the solution to the *normal equations* (e.g. [7]), $\boldsymbol{C}(n)\boldsymbol{\theta}(n) = \boldsymbol{c}(n)$, where $\boldsymbol{C}(n)$ is the weighted *normal* matrix[4] [8, p. 62]

$$\boldsymbol{C}(n) = \boldsymbol{X}^T(n)\boldsymbol{X}(n) = \sum_{t=1}^{n} \lambda_n(t)\boldsymbol{x}(t)\boldsymbol{x}^T(t)\ . \tag{4}$$

and $\boldsymbol{c}(n) \overset{def}{=} \boldsymbol{X}^T(n)\boldsymbol{y}(n) = \sum_{t=1}^{n} \lambda_n(t)\boldsymbol{x}(t)y(t)$.

A recursive solution can be obtained for certain classes of time varying weights. Consider first the case in which the weights are time *invariant*, i.e. $\lambda_n(t)$ does not depend on n for any t. In this case, we one can use a contemporary *weighted recursive least squares (WRLS)* algorithm based on the *QR decomposition* (e.g. [7]) of the $\boldsymbol{X}(n)$ matrix of (3). We shall refer to this algorithm as "QR-WRLS" to distinguish it from the more conventional WRLS algorithm based on the matrix inversion lemma (e.g. [8],[9] - [11]) (MIL-WRLS)[5]. QR-WRLS, in principle, involves the application of a sequence of orthogonal operators (Givens rotations) to (3) which leaves the system in the form

$$\begin{bmatrix} \boldsymbol{T}(n) \\ \hline \boldsymbol{0}_{(n-m)\times m} \end{bmatrix} \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{d}_1(n) \\ \hline \boldsymbol{d}_2(n) \end{bmatrix} \tag{5}$$

where the matrix $\boldsymbol{T}(n)$ is an $m \times m$ upper triangular Cholesky factor [7] of $\boldsymbol{C}(n)$, i.e., $\boldsymbol{C}(n) = \boldsymbol{X}^T(n)\boldsymbol{X}(n) = \boldsymbol{T}^T(n)\boldsymbol{T}(n)$, and $\boldsymbol{0}_{i\times j}$ denotes the $i \times j$ zero matrix. The system

$$\boldsymbol{T}(n)\boldsymbol{\theta}(n) = \boldsymbol{d}_1(n) \tag{6}$$

is easily solved using back substitution [7] to obtain the LSE estimate, $\boldsymbol{\theta}(n)$. This procedure can be performed in a recursive manner using only about m^2 memory locations. When the $n + 1^{st}$ data set

[4] In many contexts $C(n)$ is imprecisely called a "covariance" matrix. In fact, $\lim_{n\to\infty}(1/n)C(n)$ is the covariance matrix for the process if appropriate ergodicity assumptions are made.

[5] With the exception of the parallel processing architectures, developments throughout this paper may also be based upon MIL-WRLS. Indeed, almost all of the existing SM algorithms of the type considered here are based on the conventional method.

becomes available, it is weighted by $\sqrt{\lambda_n(n)}$ and incorporated into the system. Details are found in [12]-[14]. We shall use the name *QR-WRLS* to refer to this form of the recursion. It will be shown how this formulation makes possible the solution of the ellipsoid algorithms to be described on contemporary parallel architectures for great speed advantages. It also avoids initialization problems encountered in the use of MIL-WRLS [14].

The QR-WRLS algorithm can conveniently accommodate certain classes of time varying weights of interest in this work. The first is the case in which previous weights are *scaled* at time n by a time dependent scalar,

$$\lambda_n(t) = \frac{\lambda_{n-1}(t)}{\zeta(n-1)} \quad \forall t \leq n-1. \tag{7}$$

$\zeta(\cdot)$ is a *scaling sequence* which depends on the nature of the method. A common use for such scaling is to effect *adaptation by exponential forgetting.* In this case $\zeta(n) = \alpha^{-1}$, $\forall n$, where $0 < \alpha < 1$. This scaling is conveniently carried out in the course of QR-WRLS by simply multiplying the matrix and vector $\boldsymbol{T}(n)$ and $\boldsymbol{d}_1(n)$ by $\alpha^{-1/2}$ prior to considering $(y(n), \boldsymbol{x}(n))$ [13]. By a straghtforward generalization of the work in [13], it can be shown that time-varying scaling may be accomplished by a similar premultiplication by $\zeta^{-1/2}(n-1)$. Let us denote the *scaled* system of equations at time $n-1$ by $\boldsymbol{T}_s(n-1)\boldsymbol{\theta}_s(n) = \boldsymbol{d}_{1,s}(n)$.

A second type of time varying weights is used to achieve *adaptation by exclusion. In this case* it is desired to remove some prior data sets from the system prior to considering $(y(n), \boldsymbol{x}(n))$. Let the set of times corresponding to data sets to be excluded be $\mathcal{T}_{n-1}$. Then, whereas $\lambda_{n-1}(t) > 0$, $t \in \mathcal{T}_{n-1}$, it is to be true that $\lambda_n(t) = 0$, $t \in \mathcal{T}_{n-1}$. This case is accomodated within QR-WRLS by simply reentering the data set to be forgotten with its previous weight as though it represented new data, then making some simple sign changes in the algorithm [5],[15]. Because the data sets are removed by "reversing" the Givens rotations which originally included them, this process is often call *back-rotation.* It is notable that previous data sets can likewise be *partially* excluded using a similar back-rotation method [16],[17]. After all desired data sets are removed, the system of equations is often said to be *downdated* at time $n-1$, and we shall denote this by writing

$$\boldsymbol{T}_d(n-1)\boldsymbol{\theta}_d(n) = \boldsymbol{d}_{1,d}(n). \tag{8}$$

If it were to be solved for, $\boldsymbol{\theta}_d(n)$ would represent an estimate at time $n-1$ without knowledge of the excluded data sets.

2.2 The BE Constraint and the Feasibility Set

A widely-research class of SM problems is those involving *bounded error (BE)* constraints (e.g. [3]-[6],[15]-[33]). In BE identification, a pointwise bound on the true error sequence is assumed.

Ordinarily this takes the form[6]

$$\varepsilon_*^2(n) < \gamma(n), \tag{9}$$

where $\gamma(\cdot)$ is a known positive sequence. It follows immediately from (1) and (9) that the true parameters must be in the set

$$\omega(n) = \left\{ \boldsymbol{\theta} \mid \left(y(n) - \boldsymbol{\theta}^T \boldsymbol{x}(n) \right)^2 < \gamma(n) \right\}. \tag{10}$$

When intersected over a given time range usually form convex polytopes of feasible parameters, say $\Omega(n) = \bigcap_{t=1}^{n} \omega(t)$. Methods which track these polytopes [3],[4], [18]–[21] result in interesting but very complex algorithms which, at present, are not suitable for fast signal processing applications. OBE algorithms are of much lower complexity and work with an outer bounding hyperellipsoid, a superset of the polytope [6],[22]–[29]. The ellipsoid is "optimized" at each step by making some measure of its size as small as possible in light of the incoming data.

One of the drawbacks of the OBE approach from a set-theoretic point of view is that the hyperellipsoidal bounding sets are sometimes quite "loose" supersets of the actual feasibility sets (polytopes) (e.g. [22],[30]). This problem renders the resulting feasible superset "pessimistic" in that it may contain many points which are infeasible, and not reflect the size of the true feasible set. Whether certain measures can be taken, or particular OBE algorithms can be used, to minimize this problem, is an open issue. One possible solution is the use of *inner* bounds, as suggested in [30],[31]. In the present work the relative size of the bounding set will turn out to be somewhat inconsequential. It is the information afforded by the *existence* of the ellipsoid which is important.

2.3 Combining the BE and LSE Problems: The SM-WRLS Algorithm

OBE algorithms are fruitfully viewed as a marriage between the LSE and BE problems for LP models. With this point of view, signal processing engineers have begun to exploit the benefits of BE information in the context of LSE identification problems. In particular, LSE identifiers exploit no point-by-point information which can be used to ascertain the usefulness of observations. This fact manifests itself in the effective retention of the entire parameter space as a "feasible set," and results in wasteful processing. BE constraints, when they are known, provide a finite feasible set and offer the possibility of including only data points which contribute to the reduction of this set.

As mentioned above, the polytope $\Omega(n)$ arising directly from BE considerations is not easy to track and manipulate. Further, $\Omega(n)$ is not clearly related to the LSE solution. However, it has been shown in three special cases of scaling sequences, $\zeta(\cdot)$ (recall definition below (7)) , that there

[6] This form is slightly less general than stating asymmetrical amplitude bounds, $\varepsilon_{\min}(n) < \varepsilon_*(n) < \varepsilon_{\max}(n)$, but the very slight loss of generality is worth the significant analytic gain afforded by this assumption.

is an outer bounding hyperellipsoid, say $\bar{\Omega}(n)$, which contains $\Omega(n)$ and which *is* closely associated with the LSE estimate $\boldsymbol{\theta}(n)$ [6],[26],[27]. A description of the hyperellipsoid is embodied in the following:

Proposition 1 *Let $\Omega(n) \subseteq \mathcal{R}^m$ be the feasibility set arising from BE constraints as above. Let $\boldsymbol{\theta}(n)$ denote the weighted LSE estimate with associated normal matrix $\boldsymbol{C}(n)$. The weights used in the estimation are $\lambda_n(\cdot)$ with $\lambda_n(1) > 0$. There exists a hyperellipsoidal set of parameter vectors, $\bar{\Omega}(n) \subseteq \mathcal{R}^m$, such that $\boldsymbol{\theta}_* \in \Omega(n) \subseteq \bar{\Omega}(n)$, which is given by*

$$\bar{\Omega}(n) = \left\{\boldsymbol{\theta} \;\middle|\; \left\{[\boldsymbol{\theta} - \boldsymbol{\theta}(n)]^T \boldsymbol{\Phi}(n) [\boldsymbol{\theta} - \boldsymbol{\theta}(n)]\right\} < 1\right\} \tag{11}$$

where $\kappa(n)$ is the scalar quantity, $\kappa(n) \overset{\text{def}}{=} \boldsymbol{\theta}^T(n)\boldsymbol{C}(n)\boldsymbol{\theta}(n) + \sum_{t=1}^{n} \gamma(n)\lambda_n(t)\left[1 - \gamma^{-1}(t)y^2(t)\right]$, and $\boldsymbol{\Phi}(n) = \boldsymbol{C}(n)/\kappa(n)$.

Note that the ellipsoid is centered on the LSE estimate, $\boldsymbol{\theta}(n)$, and its defining matrix is a scaled version of the normal matrix, $\boldsymbol{C}(n)$.

The proof of Proposition 1 is a generalization of the proofs of similar results for special cases (discussed below) found in [6] and [26]. Another related result for complex-valued, multiple input – multiple output systems is proved in [16],[34].

Clearly, the weights $\lambda_n(\cdot)$ parameterize $\bar{\Omega}(n)$ and presumably can serve to minimize its size and orientation in the parameter space. Because we want to work with *recursive* LSE estimation, in particular QR-WRLS, let us henceforth restrict our attention to weight sequences which conform to the simple forms of time variance described in Section 2.1 – scaling and exclusion. This effectively restricts to *one* the number of free parameters available to control the bounding ellipsoid. The central objective of an *optimal bounding ellipsoid (OBE)* algorithm is to employ these free weights in the context of LSE estimation to sequentially minimize the ellipsoid size in some sense. A significant benefit is that often no weight exists which minimizes the ellipsoid size in some sense, indicating that the incoming data set is uninformative in the SM sense.

In a general sense, reported (nonadaptive) OBE algorithms differ in the scaling sequences, $\zeta(\cdot)$, used in creating time varying weights. Fogel and Huang's original OBE algorithm (henceforth called *Fogel-Huang OBE*) [6], and the more recent method by Dasgupta and Huang (henceforth called *Dasgupta-Huang OBE*) [27], are not presented from this explicit LSE point of view, and this unified approach has not been widely discussed. Some general ideas along these lines may be inferred from [33] and a unified treatment will be found in [34]. The *set membership weighted recursive least squares (SM-WRLS)* algorithm is the simplest in this sense, employing unity scaling, $\zeta(n) = 1\ \forall n$. We henceforth focus on SM-WRLS because this absence of scaling is essential to achieve the desired low complexity algorithm. Details of the other reported algorithms are left to the original papers and enhancements by Belforte *et al.* [22], and Rao *et al.* [23],[24].

Nonadaptive SM-WRLS (when based upon QR-WRLS) is comprised of the following steps: At time n,

1. In conjunction with the incoming data set $(y(n), \boldsymbol{x}(n))$, find the optimal weight, say $\lambda_n^*(n)$, which will (prospectively) minimize the size (according to some set measure) of $\bar{\Omega}(n)$, say $\mu\{\bar{\Omega}(n)\}$. (This will generally require knowledge of $\boldsymbol{C}(n-1)$ or $\boldsymbol{T}(n-1)$, and $\kappa(n-1)$.)

2. Discard the data set if $\lambda_n^*(n) \leq 0$.

3. Update $\boldsymbol{\theta}(n)$ using QR-WRLS (see Section 2.1).

4. Update $\kappa(n)$ of Proposition 1 according to

$$\kappa(n) = \| \boldsymbol{d}_1(n) \|^2 + \tilde{\kappa}(n) \tag{12}$$

with

$$\tilde{\kappa}(n) = \tilde{\kappa}(n-1) + \lambda_n(n)\gamma(n)\left(1 - \gamma^{-1}(n)y^2(n)\right) \tag{13}$$

where $\tilde{\kappa}(0) \stackrel{\text{def}}{=} 0$.

Expressions (12) and (13) are derived in [5],[15]. A detailed version of SM-WRLS is described in [5].

2.4 Adaptation by Back-Rotation

While OBE algorithms in general, and the SM-WRLS algorithm in particular, have been shown to have inherent and fortuitous adaptive properties as a result of their optimal weighting strategies, measures have been suggested by Deller and Odeh [5],[15]-[17], and Norton and Mo [33] to render explicit and controlable adaptation. All adaptive strategies for ellipsoid algorithms work on the general principle of inflating the "current" ellipsoid in some sense before considering an incoming data set. The basis for this inflation is to contain the shifting true parameters while at the same time increasing some measure of "size" of the ellipsoid (see (16) and (17) below), making it more likely that the incoming data, with potentially novel information, will be selected.

For SM-WRLS, simple forms of adaptation have been based upon exponential forgetting and by exclusion or back-rotation [5],[15]-[17]. Norton and Mo have also worked with exponential forgetting and other forms of adaptation in a broader context [33]. While exponential forgetting is conveniently integrated into OBE algorithms, in the following, we shall focus exclusively upon adaptation methods which are based on back-rotation for two reasons: First, exponential forgetting precludes the achievement of the low complexity algorithm ultimately sought in this work. Secondly, due to the fact that heavily weighted points remain influential in the estimate for very long periods of time, exponential forgetting has not been found to be as effective in tracking fast time variations in system dynamics [16],[34]. In the case of adaptation by back-rotation, the system of equations (6) is downdated prior to considering the data set at time n. The result is (8). The altered ellipsoid is centered on $\boldsymbol{\theta}_d(n-1)$ and has associated matrix $\boldsymbol{C}_d(n-1)/\kappa_d(n-1) = \boldsymbol{T}_d^T(n-1)\boldsymbol{T}_d(n-1)/\kappa_d(n-1)$.

Proper downdating of the scalar $\kappa(n-1)$ is easy. Upon rewriting the definition of $\kappa(\cdot)$ from Proposition 1 at time $n-1$,

$$\kappa(n-1) = \boldsymbol{\theta}^T(n-1)\boldsymbol{C}(n-1)\boldsymbol{\theta}(n-1) + \sum_{t=1}^{n-1} \lambda_{n-1}(t)\gamma(t)\left[-\gamma^{-1}(t)y^2(t)\right], \tag{14}$$

it becomes immediately clear that if data sets at times $t \in \mathcal{T}_{n-1}$ are eliminated from the system, then the normal matrix is simply replaced by its downdated version and all deleted terms should be removed from the sum on the right. Correspondingly, the downdated version of (12) written for time $n-1$ becomes

$$\kappa_d(n-1) = \| \boldsymbol{d}_{1,d}(n-1) \|^2 + \left[\tilde{\kappa}(n-1) - \sum_{\substack{t=1 \\ t\in\mathcal{T}_{n-1}}}^{n} \lambda_{n-1}(t)\gamma(t)\left(1-\gamma^{-1}(t)y^2(t)\right)\right] \tag{15}$$

and the term $\tilde{\kappa}(n-1)$ in (13) should be replaced by $\tilde{\kappa}_d(n-1)$ which is defined to be the term in square brackets.

A wide range of adaptation strategies is inherent in the general formulation described above, many of them computationally inexpensive. We have found two forms of adaptation by back-rotation to be particularly effective. These are:

1. *Windowing.* Let l be a fixed "window length." For each $n > l$, let $\mathcal{T}_{n-1} = \{n-l\}$.

2. *Selective Forgetting.* At time n check some predetermined criterion indicating whether adaptation is necessary. If so, select the set to be forgotten according to some other criterion.

The first case above corresponds to the use of a sliding rectangular window of length l, outside of which all previous data sets are completely removed. The estimate at time n covers the range $[n-l+1, n]$. The windowing technique is made possible by the ability to completely and systematically remove data sets at the trailing edge of the window. Only one back-rotation is required prior to optimizing at time n, and this is only necessary if $\lambda^*_{n-l}(n-l) \neq 0$.

At significantly higher computational expense, smoother windows can be implemented by back-rotation. This is accomplished by partial rotation of an included data set according to a schedule which gradually eliminates the data set [16],[17]. Since each included data set is back-rotated multiple times, the computation required to effect such a window is frequently not warranted by the benefits of slightly improved frequency resolution. For details, see [16].

Selective forgetting represents a very general class of techniques in which the data sets to be removed .rom the system are selected according to certain user defined criteria. The selection process can be, for example, to remove (or downweight) only the previously heavily weighted data sets, to remove the data sets that were accepted in regions of abrupt change in the signal dynamics, or to remove the data sets starting from the first data set and proceeding sequentially. Whatever the criterion, a fundamental issue is to detect *when* adaptation is needed to improve the parameter estimates. An example is explored in the simulation studies below.

2.5 Optimization

In the nonadaptive case, Fogel and Huang [6] suggest two set measures on $\bar{\Omega}(n)$ for optimization. These measures may also be applied to the downdated system extant at time $n-1$ if adapation is employed. For generality, we assume downdating in the following. If adaptation is not used, it is only necessary to drop subscripts "d" where they occur. The first Fogel and Huang set measure is the determinant of the matrix $\boldsymbol{\Phi}(n)$,

$$\mu_v\{\bar{\Omega}(n)\} \overset{\text{def}}{=} \det\{\boldsymbol{\Phi}(n)\} \tag{16}$$

and the second is the trace,

$$\mu_t\{\bar{\Omega}(n)\} \overset{\text{def}}{=} \text{tr}\ \{\boldsymbol{\Phi}(n)\}. \tag{17}$$

$\mu_v\{\bar{\Omega}(n)\}$ is proportional to the square root of the volume of $\Omega(n)$ while $\mu_t\{\bar{\Omega}(n)\}$ is proportional to the sum of its semi-axes. The following is a slightly generalized version (to accommodate adpatation by downdating) of results found in [6],[26]. Further generalizations are found in [34].

Proposition 2 *Let $\mathcal{T}_{n-1}$ be the set of times corresponding to data sets to be excluded by back-rotation prior to time n. Then let $\lambda_n(t), t \in [1,n]$ indicate the weights to be used to optimize (16) or (17) at time n. Under the adaptation by exclusion policy, for $t \in [1, n-1]$ and $t \notin \mathcal{T}_{n-1}$. $\lambda_n(t) = \lambda_{n-1}(t)$. For $t \in [1, n-1]$ and $t \in \mathcal{F}_{n-1}$, $\lambda_n(t) = 0$. Then,*

1. *if it exists, $\lambda_n^*(n)$ which minimizes the volume measure (16) is the unique positive root of the quadratic equation*

$$F_v(\lambda) = a_2\lambda^2 + a_1\lambda + a_0 = 0 \tag{18}$$

where, $a_2 = \{(m-1)\gamma(n)G_d^2(n)\}$,

$a_1 = \{(2m-1) + \gamma^{-1}(n)\varepsilon^2(n, \boldsymbol{\theta}_d(n-1)) - \kappa_d(n-1)\gamma^{-1}(n)G_d(n)\}\,\gamma(n)G_d(n)$,

and $a_0 = m\left[\gamma(n) - \varepsilon^2(n, \boldsymbol{\theta}_d(n-1))\right] - \kappa_d(n-1)G_d(n)$,

in which all quantities are defined above except $G_d(n) \overset{\text{def}}{=} \boldsymbol{x}^T(n)\boldsymbol{C}_d^{-1}(n)\boldsymbol{x}(n)$.

2. *if it exists, the weight $\lambda_n^*(n)$ which minimizes the trace measure (17), is the unique positive root of the cubic equation*

$$F_t(\lambda) = b_3\lambda^3 + b_2\lambda^2 + b_1\lambda + b_0 \tag{19}$$

$$
\begin{aligned}
\textit{with} \quad & b_3 = \gamma(n)G_d^2(G_d(n) - I_d(n-1)H_d(n)) \ , \\
& b_2 = 3\gamma(n)G_d(n)[G_d(n) - I_d(n-1)H_d(n)], \\
& b_1 = H_d(n)G_d(n)I_d(n-1)\kappa_d(n-1) \\
& \qquad\qquad -2H_d(n)I_d(n-1)[\gamma(n) - \varepsilon^2(n, \boldsymbol{\theta}_d(n-1))] \\
& \qquad\qquad -G_d(n)\varepsilon^2(n, \boldsymbol{\theta}_d(n-1)) + 3\gamma(n)G_d(n), \\
\textit{and} \quad & b_0 = \gamma(n) - \varepsilon^2(n, \boldsymbol{\theta}_d(n-1)) - H_d(n)I_d(n-1)\kappa_d(n-1),
\end{aligned}
$$

where $H_d(n) \overset{\text{def}}{=} \boldsymbol{x}^T(n)\boldsymbol{C}_d^{-2}(n)\boldsymbol{x}(n)$ *and* $I_d(n) \overset{\text{def}}{=}$ *tr* $\boldsymbol{C}_d(n)$.

For later computational considerations we note the following. In the context of QR-WRLS, the inverse normal matrix, $\boldsymbol{C}_d^{-1}(n-1)$, never appears, yet it is needed to compute $G_d(n)$ and $H_d(n)$. The following circumvents this problem:

Lemma 1 *In the context of QR-WRLS, the scalars $G_d(n)$ and $H_d(n)$ can be computed using $\mathcal{O}(m^2/2)$ flops.*

Proof: Write

$$G_d(n) = \boldsymbol{x}^T(n)\boldsymbol{T}_d^{-1}(n-1)\boldsymbol{T}^{-T}(n-1)\boldsymbol{x}(n) \overset{\text{def}}{=} \boldsymbol{g}^T(n)\boldsymbol{g}(n) = \| \boldsymbol{g}(n) \|^2 \tag{20}$$

in which $\| \cdot \|$ denotes the l_2 norm. Now $\boldsymbol{x}(n) = \boldsymbol{T}^T(n-1)\boldsymbol{g}(n)$, and $\boldsymbol{T}^T(n-1)$ is lower triangular, so $\boldsymbol{g}(n)$ is found by back-substitution using $(m^2+m)/2$ flops. Now note that

$$
\begin{aligned}
H_d(n) &= \boldsymbol{x}^T(n)\boldsymbol{T}_d^{-1}(n-1)\boldsymbol{T}^{-T}(n-1)\boldsymbol{T}_d^{-1}(n-1)\boldsymbol{T}_d^{-T}(n-1)\boldsymbol{x}(n) \\
&= \boldsymbol{g}^T(n)\boldsymbol{T}_d^{-1}(n-1)\boldsymbol{T}^{-T}(n-1)\boldsymbol{g}(n) \overset{\text{def}}{=} \boldsymbol{h}^T(n)\boldsymbol{h}(n) = \| \boldsymbol{h}(n) \|^2
\end{aligned} \tag{21}
$$

and back-substitution can once again be employed. □

3 Implementing SM-WRLS in $\mathcal{O}(m)$ Time

3.1 Complexity Considerations

A precise comparison of the computational loads of various OBE algorithms is given in [34]. The number of flops (see footnote 3) required for the (generally adaptive) SM-WRLS algorithm under consideration here may be approximated by

$$f_{opt} \sim \mathcal{O}(c_1 m^2) + b\mathcal{O}(c_2 m^2) + \rho\mathcal{O}(c_3 m^2) \tag{22}$$

where, ρ is the average number of data sets accepted per n; b is the average number of back-rotations per n; and c_1, c_2 and c_3 are small numbers (all in the range 0.5 - 2.5) which depend upon whether

QR-WRLS or MIL-WRLS is used. For QR-WRLS upon which we have principally focussed in this paper, $c_1 = 0.5$, $c_2 = 2$, and $c_3 = 2.5$. The first term is due to the procedure which checks for information in the incoming data. The others are attributable to adaptation, and solution update, respectively. If either an exponential forgetting factor or a non-unity scaling sequence (other OBE algorithms), is used, an additional term of $\mathcal{O}(m^2/2)$ must be added. Apparently, the SM-WRLS algorithm, as presently formulated, is an "$\mathcal{O}(m^2)$" process. The objective of the section below is to demonstrate two distinct methods for reducing the effective complexity to $\mathcal{O}(m)$, thereby making a SM-WRLS algorithm a desirable alternative to standard RLS-based methods from a computational point of view.

Two approaches are taken. The first is an algorithmic solution which will reduce the true complexity to $\mathcal{O}(m)$ for processing on a sequential machine. The second is a hardware solution which reduces the basic algorithm to $\mathcal{O}(m)$ *parallel* complexity, with even further reduction possible if the algorithmic measures are combined.

3.2 $\mathcal{O}(m)$ Processing on a Sequential Machine

From a signal processing point of view, one of the most interesting aspects of an OBE algorithm is its inherent ability to select only data points which are informative in the sense of refining the feasible set. The fact that typically 70 - 95% of the data are rejected by this criterion (e.g. [6],[17],[23]-[29]) would seem to imply a remarkable savings in computation. However, this is only true to the extent that the *checking* for usefulness of the incoming data set is negligibly expensive compared with the inclusion of it in the estimate. We have seen above, however, that the checking procedure is not inexpensive (see lead term of (22)) - a point which has not been made clear in reported research. The approach taken here is to render the checking procedure an $\mathcal{O}(m)$ process in a manner which does not (empirically) degrade performance of the algorithm.

Before detailing the methods, some points about the use of the approximation "$\mathcal{O}(m)$" are necessary. The first concerns a practical matter. The objective in the following is to reduce the computational complexity of the algorithms to an *average* of $\mathcal{O}(m)$ flops per n. It will be appreciated that, without data buffering, the data flow is still limited by the worst case $\mathcal{O}(m^2)$ computation. However, if a buffer is included, the algorithm easily be structured to operate in $\mathcal{O}(m)$ average time per n. Further, by using interrupt driven processing of the checking procedure, it may be possible to reduce the average time even further. Other points concern algorithmic details. We reiterate that the use of a unity scaling sequence (SM-WRLS algorithm) is required in order to avoid an invariant $\mathcal{O}(m^2/2)$ flops per n. We specifically assume the use of this algorithm below, although the $\mathcal{O}(m)$ checking procedure to be developed does not depend on this choice.

Secondly, (22) indicates that an adaptive strategy must involve a sufficiently small average number of back-rotations per n so that the $\mathcal{O}(m^2)$ adaptation term in (22) does not overwhelm gains made by reducing the checking cost. In the windowing case above, for example, we would expect that $b \approx \rho$ and the adaptation is not unduly expensive. A selective forgetting strategy which meets this condition will also be illustrated in the simulations below. Finally, we note that even if the checking procedure can be made $\mathcal{O}(m)$, terms $b\mathcal{O}(m^2)$ and $\rho\mathcal{O}(m^2)$ (typically $b \approx \rho$) persist in (22). This means that to truly achieve $\mathcal{O}(m)$ complexity, b and ρ must be $\mathcal{O}(1/m)$. For large m, this will not be always be the case. In fact, some experimental evidence suggests, not unexpectedly, that ρ increases, rather than decreases, with increasing m. For "large" m (conservatively, say, $m > 10$), therefore, it is the case that the complexity is reduced to $\mathcal{O}(\rho m^2)$ by $\mathcal{O}(m)$ checking. It should be clear however, that neither $\mathcal{O}(m)$ nor $\mathcal{O}(\rho m^2)$ complexity can be achieved if the checking procedure remains $\mathcal{O}(m^2)$. We therefore pursue an $\mathcal{O}(m)$ test for information in an incoming data set.

In principle, the information checking procedure for the volume or trace algorithms consists of forming either $F_v(\lambda)$ or $F_t(\lambda)$ of (18) and (19), then solving for the positive root. However, since $a_2 > 0$, and $b_i > 0$, $i = 1, 2, 3$, there is at most one such root in either case, and the test reduces to one of checking the zero order coefficient for negativity [35]. When the test is successful, then the root solving and updating procedes, requiring the standard MIL- or QR-WRLS load, plus a few operations for finding the optimal weight. In spite of Lemma 1, the most expensive aspect of this information test is the computation of the quantity $G_d(n)$ or $H_d(n)$, each requiring $\mathcal{O}(m^2/2)$ flops. The trick to making the SM-WRLS algorithm an $\mathcal{O}(m)$ procedure is to find a way to avoid the computation of $G_d(n)$ or $H_d(n)$ at each n. We first develop a method which accomplishes this for the "volume" algorithm, then argue that it pertains to the "trace" optimization criterion as well.

Let us denote the estimation *error* vector at time n by

$$\tilde{\boldsymbol{\theta}}(n) \overset{\text{def}}{=} \boldsymbol{\theta}_* - \boldsymbol{\theta}(n). \tag{23}$$

It follows immediately from (11) that $\tilde{\boldsymbol{\theta}}^T(n)\boldsymbol{C}(n)\tilde{\boldsymbol{\theta}}(n) < \kappa(n)$. While it is tempting to view $\kappa(n)$ as a bound on $\tilde{\boldsymbol{\theta}}(n)$ (see discussion of the Dasgupta-Huang algorithm below), it is important to note that each side of this inequality is dependent upon $\lambda_n(n)$. In fact, let us temporarily write the two key quantities as functions of $\lambda_n(n)$: $\boldsymbol{C}(n, \lambda_n(n))$ and $\kappa(n, \lambda_n(n))$ and consider the usual volume quantity to be minimized at time n,

$$\mu_v\{\tilde{\Omega}(n)\} = \det\left[\kappa(n, \lambda_n(n))\boldsymbol{C}^{-1}(n, \lambda_n(n))\right]. \tag{24}$$

It is assumed that enough data sets have been included in the normal matrix at time $n - 1$ so that its elements are large with respect to the data in the incoming data set. For the choice of weighting strategy employed here, the quantity $\det \boldsymbol{C}(n, \lambda_n(n))$ is readily shown to be monotonically increasing with respect to $\lambda_n(n)$ on the domain $(0, \infty)$ [16], with $\boldsymbol{C}(n, 0) \overset{\text{def}}{=} \boldsymbol{C}(n-1, \lambda^*_{n-1}(n-1))$. Under the assumption above, $\det \boldsymbol{C}(n, \lambda_n(n))$ will not increase significantly over reasonably small values of $\lambda_n(n)$. The attempt to maximize $\det \boldsymbol{C}(n, \lambda_n(n))$ in (24) causes a tendency to increase $\lambda_n(n)$ in the usual optimization process. However, the attempt to minimize $\kappa(n, \lambda_n(n))$ generally causes a tendency toward small values of $\lambda_n(n)$, unless a minimum of $\kappa(n, \lambda_n(n))$ occurs at a "large" value of $\lambda_n(n)$. To pursue this idea and further points of the argument, we use two key facts about $\kappa(n, \lambda(n))$:

Proposition 3 *$\kappa(n, \lambda_n(n))$ has the following properties: 1. On the interval $\lambda_n(n) \in (0, \infty)$, $\kappa(n, \lambda_n(n))$ is either monotonically increasing or it has a single minimum. 2. $\kappa(n, \lambda_n(n))$ has a minimum on $\lambda_n(n) \in (0, \infty)$ iff*

$$\varepsilon^2(n, \boldsymbol{\theta}_d(n-1)) > \gamma(n). \tag{25}$$

To verify this result we need the following which is proven in [34]:

Lemma 2 *For $n > 1$, the sequence $\kappa(\cdot)$ can be computed recursively as*

$$\kappa(n) = \kappa_d(n-1) + \lambda_n(n)\gamma(n) - \lambda_n(n)\frac{\varepsilon^2(n, \boldsymbol{\theta}_d(n-1))}{1 + \lambda_n(n)G_d(n)}. \tag{26}$$

Proof of Proposition 3: For simplicity, we write $\lambda_n(n)$ as λ. Using (26) from Lemma 2, we can write

$$Q(\lambda) \overset{\text{def}}{=} \frac{\partial \kappa(n, \lambda)}{\partial \lambda} = \frac{G_d^2(n)\gamma(n)\lambda^2 + 2G_d(n)\gamma(n)\lambda + [\gamma(n) - \varepsilon^2(n, \boldsymbol{\theta}_d(n-1))]}{G_d(n)^2\lambda^2 + 2G_d(n)\lambda + 1} \tag{27}$$

and

$$\dot{Q}(\lambda) = \frac{\partial^2 \kappa(n, \lambda)}{\partial \lambda^2} = \frac{2[G_d^2(n) + \gamma(n)G_d(n)]\varepsilon^2(n, \boldsymbol{\theta}_d(n-1))}{(G_d^2(n)\lambda^2 + 2G_d(n)\lambda + 1)^2}. \tag{28}$$

The denominator of $Q(\lambda)$ is positive on $\lambda \in (0, \infty)$ and therfore has a root on $\lambda \in (0, \infty)$ iff its numerator does. The the numerator is a convex parabola with its minumum at $\lambda = -1/G_d(n) < 0$, and it therefore has a unique positive root on the interval $(0, \infty)$ iff $\gamma(n) - \varepsilon^2(n, \boldsymbol{\theta}_d(n-1)) < 0$. Further $\dot{Q}(\lambda) > 0$ for all $\lambda > 0$, so the root, if it exists, will correspond to a minimum of $\kappa(n, \lambda)$.□

Accordingly, it can be argued that: If $\det \boldsymbol{C}(n, \lambda_n(n))$ is increasing, but not changing significantly over reasonably small values of $\lambda_n(n)$, then it is sufficient to seek $\lambda_n(n)$ which minimizes $\kappa(n, \lambda_n(n))$. If $\kappa(n, \lambda_n(n))$ is monotonically increasing on $\lambda_n(n) \geq 0$, this value is $\lambda_n(n) = 0$ which corresponds to rejection of the data set at time n. It suffices, therefore to have a test for a minimum of $\kappa(n, \lambda_n(n))$ on positive $\lambda_n(n)$. A simple test is embodied in condition (25) which determines whether the square of the current residual exceeds the upcoming error bound. If this test is met, it is then cost effective to proceed with the standard optimization centered on (18). Otherwise, the explicit construction and solution of a_0 of (18) can be avoided.

In fact, this suboptimal test for innovation is similar to that used in the Dasgupta–Huang OBE algorithm reported in [27]. The suboptimal test of Dasgupta is to accept the incoming data set only if[7] $\varepsilon^2(n, \boldsymbol{\theta}(n-1)) < \gamma(n) - \kappa(n-1)$. This inequality likewise tests for a minimum of κ with respect to $\lambda_n(n)$, and differs in form from (25) because of the scaling factors (see (7) and surrounding discussion) which depend on the optimal weights, $\zeta(n-1) = (1 - \lambda_n^*(n))^{-1}$ in the Dasgupta case. While this dependence precludes the construction of a reasonable expression in $\lambda_n^*(n)$ with which to minimize the set measure $\mu_v\{\bar{\Omega}(n)\}$, the Dasgupta hyperellipsoid nevertheless *does* have a volume at each n, and it is therefore possible to attempt to apply the above arguments. A problem arises in the Dasgupta–Huang case, however, because the relative independence of $\boldsymbol{C}(n)$ and $\lambda_n(n)$ is not tennably argued. Therefore, the simplified test in this case is not subject to the "same" justification as (25). Interestingly, however, if $\lambda_n(n)$, which is already constrained to $[0, 1)$ in the Dasgupta–Huang algorithm, happens to be very small at a particular n, then the algorithm approaches the case of unity scale factors ($\zeta(n) \approx 1$) as in SM-WRLS, and it can be argued that the normal matrix changes only slightly. In this case, but only in this case, the arguments above are applicable. Of course, artificially constraining the weights to be small for all n destroys the optimization process in the Dasgupta-Huang method, so that this analysis provides support for the suboptimal test only for isolated and infrequent times. Dasgupta and Huang argue simply that $\kappa(n)$ is "a bound on the estimation error," and should be minimized. This claim has been disputed by Norton and Mo [33] and is not clearly supported here. Generally, the arguments in support of (25) are valid only for certain types of scaling sequences which do not cause the estimation process to "forget" too quickly. This is not generally the case with the Dasgupta–Huang strategy.

Before proceding, another comparison to the Dasgupta–Huang OBE algorithm should be made. One of the principal advantages of their method is the ability to conveniently prove convergence of the ellipsoid to a point ($\boldsymbol{\theta}_*$). The original Fogel and Huang paper [6] is often cited as proving that the bounding ellipsoid in the Fogel–Huang OBE algorithm converges to a point under ordinary conditions on $\varepsilon_*(\cdot)$. In fact, the paper only proves this convergence for the case of unity weights so that the fundamental optimization process is not taken into account. No known proof of this desirable result for the Fogel–Huang OBE algorithm, or for any version of SM-WRLS exists, whether optimal or suboptimal checking is used. While the estimate itself is guaranteed to converge asymptotically under proper conditions on $\varepsilon_*(\cdot)$ (e.g. [10]), the ellipsoid is not guaranteed to diminish

[7] Subscripts "d" are omitted here since their algorithm does not involve this form of adaptation.

asymptotically. However, we have found empirically that the optimal and suboptimal tests tend to produce an ellipsoid with a similar "size" at a given point in the signal, and to produce similar estimates, in spite of the fact that the suboptimal test tends to use fewer data (see simulations below).

A further interpretation of (25) is possible which also allows the extension of the test to include "trace" minimization as well. A simple rationale for the suboptimal test is as follows:

Proposition 4 *If the test of (25) is met, then a positive optimal weight exists for either the volume or trace criterion.*

Proof: We show that the zero order cofficients a_0 and b_0, of (18) and (19), respectively, will never be positive if the test is met. Consider $a_0 = m\left[\gamma(n) - \varepsilon^2(n,\boldsymbol{\theta}_d(n-1))\right] - \kappa_d(n-1)G_d(n)$. Write (11) for the downdated case at time $n-1$, then multiply through by $\kappa_d(n-1)$. The result is $[\boldsymbol{\theta} - \boldsymbol{\theta}_d(n-1)]^T \boldsymbol{C}_d(n-1)[\boldsymbol{\theta} - \boldsymbol{\theta}_d(n-1)] < \kappa_d(n-1)$. If $\boldsymbol{C}_d(n-1)$ is positive definite, this implies that $\kappa_d(n) > 0$. Further $G_d(n) = \boldsymbol{x}^T(n)\boldsymbol{C}_d(n-1)\boldsymbol{x}(n) > 0$, so $a_0 < 0$ if the test (25) is met. Now, consider $b_0 = \gamma(n) - \varepsilon^2(n,\boldsymbol{\theta}_d(n-1)) - H_d(n)I_d(n-1)\kappa_d(n-1)$. By similarly showing that $H_d(n) > 0$ and $I_d(n-1) > 0$, the desired result for the trace criterion is obtained. □

In the volume case, for example, the suboptimal check tests whether a_0 is negative *if the term* $\kappa_d(n-1)G_d(n)$ *is neglected.* This ignored term is always negative and becomes small as n increases. For a given set of preceding optimal weights, $\lambda^*(1),\ldots,\lambda^*(n-1)$, *the suboptimal test will never fail to accept an data set which would have been accepted by the optimal test.* A similar analysis applies to the coefficient b_0 of the trace algorithm.

With the inexpensive test afforded by (25), the checking procedure becomes an $\mathcal{O}(m)$ procedure. Consequently, for sufficiently small ρ, the SM-WRLS algorithm can be run in $\mathcal{O}(m)$ time per n.

3.3 Simulation Studies and Further Discussion

OBE algorithms which do not include explicit adaptation measures have been demonstrated in numerous papers cited above. Our principle objective here is to briefly illustrate the use of the adaptive and, particularly, the $\mathcal{O}(m)$ suboptimal checking procedures.

We consider the identification of a time varying *AR(14)* model of the form

$$y(n) = \sum_{i=1}^{14} a_{i*}(n)y(n-i) + \varepsilon_*(n). \tag{29}$$

A set of "true" AR parameters were derived using linear prediction analysis (e.g. [36]) of order 14 on an utterance of the word "seven" by an adult male speaker. The original speech waveform is shown in Fig. 1 to illustrate the time varying nature of the signal. A 7000 point sequence, $y(n)$, was generated by driving the derived set of parameters with an uncorrelated sequence, $\varepsilon_*(n)$, which was uniformly distributed on $[-1, 1]$.

The speech signal was not used directly in this study for a simple reason. The problem of determining proper bounds for the model error is a nontrivial one for real speech, and a proper description of this point would seriously sidetrack the present discussion. Similarly, space would not permit a careful discussion of the performance of the algorithm in cases in which noise bounds are uncertain or violated. The predecessor (optimal, nonadaptive case) methods to those illustrated here have been successfully applied to real speech and these results are reported in [26] where some of these more difficult issues are also adressed. In the same vein, the artificial noise permits carefully controlled statistical properties. The model noise used here is uncorrelated, and this algorithm in its present form will converge to a bias if this is not the case. A discussion of colored noise, while interesting and useful, is beyond the scope of this paper. The interested reader is referred to [24],[28],[34]. While the uniform distribution chosen here has become conventional in testing OBE algorithms, it is worth noting that the performance of the methods is bound to be affected to some extent by the choice of this distribution. This becomes clear upon recognizing that the algorithm tends to favor the acceptance of data at time n when the residual is large. In some preliminary runs with bounded but nonuniform distributions, we do not find these effects to be very significant.

In the simulations below, we apply the conventional and adaptive SM-WRLS algorithms with "volume" optimization to the identification of the a_{i*} parameters. We discuss several simulation results. Only the result for a_{4*} is shown in each case to conserve space. Of the 14 parameters, a_{4*} emerged as the most difficult to track and gave the *worst* performance. Each figure shows two curves, one for the true parameter, the other for the estimate obtained by the algorithm under study.

In several previous studies, it has been demonstrated that that OBE algorithms have inherent adaptive capabilities by virtue of their optimal data weighting strategies, even when not explicitly designed to be adaptive (e.g. [24]-[29]). The adaptive capability of "nonadaptive" OBE algorithms is somewhat unpredictable and fortuitous, especially for fast time variations. Further they are subject to divergence if the true parameters move outside the feasible set. Nevertheless, SM-WRLS and other OBE algorithms often demonstrate this inherent ability. The present example is contrary. Figure 2 illustrates the result of applying SM-WRLS to the time varying waveform. The estimate clearly fails to appropriately track the true parameter in this case.

Before proceeding, let us use the present result to emphasize a principal point made in the paper. The result of Fig. 2 is achieved using only the fraction $\rho = 0.079$ of the data. Other examples are found in the literature where *good* tracking is achieved with similar, or even smaller, fractions of the data used. It is important to keep in mind, however, that the computational complexity of the SM-WRLS algorithm is only a factor of about five better than conventional RLS, and the "$\rho = 0.079$" figure must not be interpreted to the contrary. Herein lies the motivation for the suboptimal checking procedure.

Next, we show the simulation results of the variations on the adaptive SM-WRLS algorithm. Figure 3 shows the results of the windowed SM-WRLS algorithm using windows of lengths 500, 1000, and 1500. This strategy uses the fractions $\rho = 0.221, 0.174$, and 0.143 of the data, respectively, but remains an $\mathcal{O}(m^2)$ process because optimal checking is used. Additionally, each time an *accepted* point occurs at the trailing edge of the window, a back-rotation is needed to effect adaptation. This implies an average number of back-rotations $b \approx \rho$ per n (see Section 3.1). More data and more rotations than with the unmodified SM-WRLS algorithm are used, but more accurate estimates result and the time varying parameters are tracked more quickly and accurately. As expected, adapting over smaller windows tended to improve time resolution, but increased the variation of the estimate and increased the number of points accepted. Conversely, the longer windows yielded smoother estimates using fewer data, but at the expense of slower tracking. While no window length in this range yielded grossly unacceptable estimates, the 1000 point window illustrated represents a good tradeoff between the demands of time and frequency resolution.

Figure 4 illustrates the use of suboptimal checking in conjunction with windowed SM-WRLS with a window of length 1000. Interestingly, the fraction of the data used is $\rho' = 0.087$ which is about half that required in the same experiment with optimal data checking (Fig. 3(b)). This means that the suboptimal checking not only reduced the computational effort of checking, but also decreased by a factor of two the number of m^2 complexity rotations required. Nevertheless, the estimate trace is quite similar to the optimal case, the only difference being a slight increase in the variance near the end of the trace. Similar results were obtained for windows of length 500 and 1500.

The selective forgetting strategy chooses data sets to be removed from the system based on user defined criteria. Here the set of times to be back-rotated is as follows. Let $t' < n$ correspond to the "oldest" data set remaining in the estimate. Then $\mathcal{F}_{n-1} = \{t', \ldots, t''\}$, where the elements in the set are ordered, $t' < \cdots < t''$, and $t'' < n$ is the smallest time for which some other criterion is met. The determination of when to apply the forgetting procedure and when to stop removing data sets at a given time is discussed in the following.

The parameter a_{4*} to be tracked in this study is characterized by relatively fast time variations in the time region 2000 - 6000. The fact that the parameters change relatively slowly in the first 2000 points induces the algorithm to accept some points which, in turn, causes the ellipsoid volume to decrease. An increase in the "confidence" of the estimate results. Near time 2000, the ellipsoid volume becomes very small. When the parameters move rapidly away from their current location, they eventually move outside the ellipsoid which is therefore no longer a valid bounding ellipsoid. When this condition happens, it eventually leads to a negative value of $\kappa(n)$. For a stationary system, $\kappa(n)$ is always positive, so that this condition indicates that a violation of the theory (in particular, the violation of the assumption of stationary dynamics) has taken place[8]. A similar condition was also reported by Dasgupta and Huang [27] while applying their algorithm to nonstationary systems. In our simulation studies, we find that a negative $\kappa(n)$ is often an effective indicator of need for adaptation, and we use this criterion as the prompt to begin selective forgetting. Whenever accepting a data set causes $\kappa(n)$ to become negative, the algorithm starts rotating out the selected data sets until $\kappa(n)$ becomes positive again.

Figure 5 shows the simulation results of the selective forgetting strategy described here. The fraction $\rho = 0.129$ of the data is accepted by the estimation procedure and about 73% of these are back-rotated for adaptation. This implies a small "b" factor of about 0.094 per n so that adaptation is not expensive in this case. The checking process is still $\mathcal{O}(m^2)$, however, so the overall process remains of $\mathcal{O}(m^2)$ complexity. Suboptimal checking for the same experiment is illustrated in Fig. 6. In this case $\rho' = 0.088$ of the data is used with similar results. About 63% of these data are back-rotated, so that $b = 0.055$. Once again, the suboptimal test has preserved the quality of the estimate and lowered not only the checking complexity, but also the number of actual rotations that need be implemented.

Compared to the windowed adaptive strategies, for this example the selective forgetting strategy yields smoother estimates using even fewer computations, but with poorer time resolution. (Recall that a_{4*} was found to be the most difficult to track in this simulation, so that this result is the worst case.) In general, we have found that selective forgetting (as employed here) generally uses fewer data and produces smoother estimates, but the tracking ability is not as reliable (though sometimes superior) to the windowed method [16],[34]. In fact, the selective forgetting strategy (as used here) tends to outperform windowing in cases of *very* fast time variations in the dynamics. The conservative schedule of back-rotations employed in the present technique accounts for this observation. $\kappa(n) > 0$ is only a necessary condition for the true parameters to be inside the current ellipsoid. The fact that $\kappa(\cdot)$ goes negative at a particular time does not precisely determine the point at which system dynamics began to change. If the variations are slow, this may occur (if at all) long after the dynamics begin to change. In fact, $\kappa(n) < 0$ often indicates a rather severe breakdown of

[8] Mathematically, $\kappa(n) < 0$ indicates an ellipsoid of negative dimensions.

the process indicating that the "true" parameters have moved well outside the current ellipsoid at time n. In cases of *fast* changing dynamics that this "breakdown" occurs rapidly enough to render the condition "$\kappa(n) < 0$" good locator of changing dynamics which require "immediate" adaptation to preserve the integrity of the process. The present example represents a very challenging case in the sense that variations apparently occur too rapidly to be tracked by standard SM-WRLS (see Fig. 2), yet not quickly enough to allow very high time resolution by the chosen selective forgetting method. Other methods for selection leading to a more aggressive elimination of past data may assist in the tracking at the expense of higher fractions of data used.

4 Architectural Solutions to Achieving $\mathcal{O}(m)$ Time

4.1 Systolic Architecture

In this section we develop parallel architectures on which both suboptimal and optimal checking versions of SM-WRLS will run in $\mathcal{O}(m)$ time. Here the efficiency is achieved by parallelism so that the number of operations is *effectively* reduced by simultaneous execution of many ccmputations. Accordingly, the $\mathcal{O}(m)$ flop per n load to be achieved is actually a *parallel complexity* since many processors might be performing $\mathcal{O}(m)$ flops simultaneously. From a temporal viewpoint, the processing is reduced from the $\mathcal{O}(m^2)$ time required to compute the optimal solution sequentially, to $\mathcal{O}(m)$.

In the following we will assume the use of SM-WRLS (no scaling) for simplicity. Unlike the sequential case, however, scaling can be done in parallel here and does not add a significant computational burden. The modification of the following to include scaling is straightforward. We also use ellipsoid volume minimization for optimization, but a similar machine may be developed to implement trace optimization.

We first discuss the "nonadaptive" case. The fundamental parallel solution is made possible by the QR-WRLS version of SM-WRLS. The main computational requirements are a GR processor (to effectively execute the QR decomposition) to update the matrix $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$ at each step, and a back substitution (BS) processor to solve for the scalar $G(n)$ and also for the estimate $\boldsymbol{\theta}(n)$ at each n. Systolic processors for these operations, based on the original work of Gentleman and Kung [37] and Kung and Leiserson [38], are well known. It is the purpose of this section to manifest this algorithm as a parallel architecture based on these processors.

The need for implementing the algorithm on a parallel architecture arises from the fact that portions of the algorithm are compute-bound, specifically, updating the matrix $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$ and computing the value $G(n)$ and the parameter vector $\boldsymbol{\theta}(n)$. The architecture that speeds up the computation of these quantities and satisfies the desirable characteristics of systolic arrays (SA's) is shown in Fig. 7. Although this architecture is based on SA design methodologies, it is used *here* to process one data set at a time (more on this below), and therefore, is not used as a SA. This architecture provides an improvement over that described in [39] by replacing the global buses with local buses for communication between adjacent cells. *For simplicity of notation, the figure shows a purely autoregressive case of order three, AR(3). Once the processor is understood, it should be clear that the architecture is perfectly capable of handling the general LP model case discussed above.* In the discussion below, the vector notations $\boldsymbol{x}(n)$ and $\boldsymbol{\theta}(n)$ are used, however, the architecture of Fig. 7 uses the vectors $\boldsymbol{y}(n)$ and $\boldsymbol{a}(n)$ instead to denote the special case AR(3), where $\boldsymbol{y}(n) = [y(n-1)\ y(n-2)\ y(n-3)]^T$.

The architecture is composed of two SA's, several memory management units (i.e., *First-in First-out* (FIFO) and *Last-in First-out* (LIFO) stacks[9]), multiply-add units (MAU's), multiplexers (MPX's), and demultiplexers (DMX's). The first SA is a *triangular* array that performs QR decomposition using GR's [37, 40] which are particularly suitable for solving recursive linear LSE problems. The diagonal (circular) cells perform the "Givens generation" (GG) operations and all other (square) cells in the triangular array perform the GR operations. There is a delay element at the lower right-hand corner of the triangular array that is used to synchronize the flow of the generated entries into the FIFO stacks and to simplify the control of these stacks once they are filled and ready to output their contents to the BS array. The operations performed by this array are shown in Fig. 8 [37, 40]. Therefore, the triangular array rotates the new data set into the upper triangular matrix $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$, where the t_{ij} cells update the matrix $\boldsymbol{T}(n)$ and the right-hand column (d_{1j}) cells update the vector $\boldsymbol{d}_1(n)$. The element t_{ij} denotes the ij^{th} element of the matrix $\boldsymbol{T}(n)$ and the element d_{1j} denotes the j^{th} element of the vector $\boldsymbol{d}_1(n)$.

The second array is a linear array that performs the BS operations shown in Fig. 9 [38]. Note that the same BS array is used to solve for the vectors $\boldsymbol{g}(n+1)$ and $\boldsymbol{\theta}(n)$ with the data provided to the appropriate cells in the required order by the FIFO and LIFO stacks. The FIFO stacks feed the lower triangular matrix $\boldsymbol{T}^T(n)$ to solve for the vector $\boldsymbol{g}(n+1)$, and hence, the value $G(n+1)$. The LIFO stacks feed the upper triangular matrix $\boldsymbol{T}(n)$ to solve for the parameter vector $\boldsymbol{\theta}(n)$. The values $G(n+1) = \| \boldsymbol{g}(n+1) \|^2$ and $\| \boldsymbol{d}_1(n) \|^2$ are generated by the MAU's shown in Fig. 10. The number of segments in each stack is equal to the number of elements the stack holds. Therefore, the leftmost stack consists of m segments, whereas the rightmost stack has only one segment.

[9] The architecture shown in Fig. 7 does not include any of the LIFO stacks that were used to hold the matrix $T(n)$ in the architecture reported in [39]. This is achieved by slightly increasing the complexity of the cells used in the triangular array so that they can be used as - rage elements as well. This is facilitated by the diagonal interconnections between adjacent cells which now constitute the LIFO stacks.

The system shown in Fig. 7 works as follows. The first m+1 data sets (with appropriate weights) enter the triangular array (from the top) in a skewed order, and the matrix $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$ is generated and stored inside the cells. A shift register with appropriate feedback connection and data sequencing can be used to hold and feed the data set to the array. The initial upper triangular matrix residing in the array, and corresponding to the first $m+1$ data sets, is ready after $3m+1$ GG time cycles. The GG time cycle is that of the triangular array performing the GG operations without square roots, which is the time required to perform five flops [40],[41]. In order to prevent data collision, the flow of data in the triangular array moves along a corresponding wavefront and is controlled by the slowest cells in the array, *viz.*, GG cells. The data are fed to the array one (skewed) data set at a time, therefore, the contents of each cell remains constant after the completion of the current recursion. After the new data set is rotated into the matrix $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$, the vectors $\boldsymbol{g}(n+1)$ and $\boldsymbol{\theta}(n)$ are computed. All the t_{ij} cells in the triangular array load their contents on the t_{out} lines ($t_{out} \leftarrow x$), and then pass these elements across the diagonal lines ($t_{out} \leftarrow t_{in}$) (see Figs. 7 and 8). This obviates LIFO stacks. The FIFO stacks are still needed, however, to compute the vector $\boldsymbol{g}(n+1)$. The FIFO stacks are filled with the elements of the lower triangular matrix $\boldsymbol{T}^T(n)$ as they are generated. This is done by loading the t_{ij} entry on the t_{out} line ($t_{out} \leftarrow x$) when it is generated. This entry propagates down the diagonal cells (with the function $t_{out} \leftarrow t_{in}$) until it arrives at and fills the appropriate FIFO stack. For the cells in the right-hand column, which generate the vector $\boldsymbol{d}_1(n)$, the operations are different because it is this column that constitutes the LIFO stack for the vector $\boldsymbol{d}_1(n)$. Hence, after the new data set is rotated into the array, all the cells in the right-hand column load their contents on the x_{out} lines ($x_{out} \leftarrow x$), and then they pass these elements down the column ($x_{out} \leftarrow x_{in}$) (see Figs. 7 and 8). The output x_{out} leaving the bottom cell in this column passes through the delay element and is routed to both the MAU and the MPX feeding the d_{1j} elements to the BS array. The elements d_{1m} and t_{mm} leave the triangular array at the same time because of this delay element. The timing diagram of the triangular array is shown in Table 1. In this table, the inputs refer to the elements fed to the cells in the top row. The circle (○) represents the GG cell and the square (□) represents the GR cell (see Fig. 7). The outputs refer to the elements that are produced in the array cells and are written columnwise; i.e., the first column in the table represents the first column in the array, and so on.

The BS array is used to solve for the vectors $\boldsymbol{g}(n+1)$ and $\boldsymbol{\theta}(n)$. The vector $\boldsymbol{g}(n+1)$ is solved using (20) and the parameter vector $\boldsymbol{\theta}(n)$ using (6). Therefore, the vector $\boldsymbol{g}(n+1)$ is generated from the matrix $\boldsymbol{T}^T(n)$, which is residing in the FIFO stacks, and the vector $\boldsymbol{x}(n+1)$ which is available. The entries are fed to the BS array every other BS time cycle, where the BS time cycle is the time required to perform one flop. As the g_i entries are output from the left-end processor of the BS array, they enter the MAU to generate the value $G(n+1)$ after $2m+1$ BS time cycles. Likewise, the parameter vector $\boldsymbol{\theta}(n)$ is generated using the matrix $\boldsymbol{T}(n)$ and the vector $\boldsymbol{d}_1(n)$ which are stored in the triangular array. Starting one BS time cycle after the initiation of the first BS operation, the appropriate entries (of the second BS operation) are also fed to the BS array every other BS time cycle. The parameter vector $\boldsymbol{\theta}(n)$ is output from the left-end processor of the BS array in reversed order and interleaved with the vector $\boldsymbol{g}(n+1)$ as shown in Fig. 7. The value $\| \boldsymbol{d}_1(n) \|_2^2$ is generated using a MAU one BS time cycle after the last (m^{th}) element of the vector $\boldsymbol{d}_1(n)$ is generated. The timing diagram of the BS array is shown in Table 2 in which the inputs refer to the elements fed to the shown cells, and the outputs refer to the elements produced by the left-end processor in the array.

The values $\kappa(n)$ and $\epsilon^2(n+1, \boldsymbol{\theta}(n))$ are then computed, and hence, the value $\lambda^*_{n+1}(n+1)$ which determines whether the new data set is to be accepted or not. If the new data set is accepted, then the *weighted* new data set enters the triangular array and the same procedure described above takes place producing a new $[\boldsymbol{T}(n+1) \mid \boldsymbol{d}_1(n+1)]$ matrix after $2m+1$ GG time cycles, and therefore, an updated $G(n+2)$, $\boldsymbol{\theta}(n+1)$, and $\kappa(n+1)$. On the other hand, if the new data set is rejected, then the triangular array preserves its contents (hold state), but the value $G(n+2)$ is updated to make the decision concerning the next data set. In the latter case, the same $\boldsymbol{T}^T(n+1)$ matrix is used as the previous $\boldsymbol{T}^T(n)$ matrix, and hence, the feedback on the FIFO stacks. This procedure is repeated for every new data set.

The computational complexities (in flops per data set) for the architecture of Fig. 7 is approximated by [16]

$$f_{\underset{parallel}{opt}} \sim \mathcal{O}(3m) + \rho\mathcal{O}(11m) \qquad (30)$$

where the first term accounts for checking and the second for solution update, with ρ defined as usual. As noted at the outset, the complexities of the solution are *parallel* complexities in the sense that they denote the *effective* number of operations per data set, though many processors can be performing this number of operations simultaneously. Accordingly, the parallel complexity indicates the time it takes the parallel architecture to process the data, regardless of the total number of operations performed by the individual cells. The GG and GR operations constitute the main computational load of the algorithm as shown in Table 3. In this table, the number of flops associated with the GR's is multiplied by five to account for the GG cycle time. These operations are avoided when the data set is rejected, and thus, a significant savings in computation time is achieved.

Suboptimal checking may also be used in conjunction with the parallel processing. In this case it is simply unnecessary for the processor to compute the first three items in Table 3 in order to check the incoming data set. The reduction in computation, which is is not as significant as in the sequential processing case, is reflected by the approximation

$$f_{\substack{subopt \\ parallel}} \sim \mathcal{O}(m) + \rho'\mathcal{O}(11m) \tag{31}$$

for small ρ' [16].

4.1.1 Adaptive Compact Architecture

The architecture described above can be modified to improve cell utilization and to incorporate adaptation by back-rotation. The basic idea behind the compact architecture is to map the triangular array of Fig. 7 into a linear array (called the GR array), that is, mapping all of the GG cells into one GG cell and all the GR cells that are on the same diagonal into one GR cell. This constitutes a permissible schedule because the projection vector, $\vec{d}$, is parallel to the schedule vector, $\vec{s}$, and all the dependency arcs flow in the *same* direction across the hyperplanes (e.g. [42, Ch. 3]). In other words, this schedule satisfies the conditions $\vec{s}^T\vec{d} > 0$ and $\vec{s}^T\vec{e} \geq 0$, for any dependence arc $\vec{e}$.

The compact architecture implementation of the adaptive SM-WRLS algorithm is shown in Fig. 11. The operations performed by this architecture are similar to those of Fig. 7 with the exception that the GG and GR cells are now capable of performing back rotation (see Fig. 12) and are embedded in a slightly more complicated modules needed for scheduling. These modules are called GG′ and GR′, and are shown in Fig. 13.

This architecture uses $\mathcal{O}(m)$ cells (one GG′ cell and m GR′ cells) compared with $\mathcal{O}(m^2)$ cells (m GG cells and $(m^2+m)/2$ GR cells) used in the architecture shown in Fig. 7, and yet has the same computational efficiency per n. Note however that the LIFO stacks that were embedded in the triangular array of Fig. 7 are now needed to hold the matrix $\boldsymbol{T}(n)$.

The system shown in Fig. 11 works as follows. Each data set (with its optimal weight) enters the GR array (from the top) in a skewed order, and the matrix $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$ is generated and stored in the appropriate memory units. Note that the GR array can operate in two modes, forward ($\delta = +1$) and backward ($\delta = -1$) rotation modes (see Fig. 12). In the backward rotation mode, the data set to be removed is re-introduced to the GR array with the appropriate weight. At the end of each recursion, the FIFO stacks contain the lower triangular matrix $\boldsymbol{T}^T(n)$ needed to solve for the vector $\boldsymbol{g}(n+1)$, and hence, the value $G(n+1)$. The LIFO stacks contain the upper triangular matrix $\boldsymbol{T}(n)$ needed to solve for the parameter vector $\hat{\boldsymbol{\theta}}(n)$. The values $G(n+1) = \| \boldsymbol{g}(n+1) \|^2$ and $\| \boldsymbol{d}_1(n) \|^2$ are generated by the MAU's. Note that the values which were propagating downward in the triangular array of Fig. 7 are now propagating leftward due to the new scheduling. Note also that the vector $\boldsymbol{d}_1(n)$ is treated differently from the matrix $\boldsymbol{T}(n)$. When the element d_{1_j} is computed, it is stored in an internal register in the GR$'$ cell (see Fig. 13). After generating and storing the matrix $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$, the processor is ready to compute the vectors $\boldsymbol{g}(n+1)$ and $\boldsymbol{\theta}(n)$ using the BS array. The vector $\boldsymbol{d}_1(n)$ is downloaded into the latches which serve as a LIFO stack used in conjunction with the other LIFO stacks (containing the matrix $\boldsymbol{T}(n)$) to solve for the parameter vector $\boldsymbol{\theta}(n)$. The timing diagram of the GR array is shown in Table 4 in which the input (output) columns show the elements that are input (output) to (from) the corresponding GG (◯) or GR (□) cells. Compared to the triangular array of Fig. 7, it is noted that the cell utilization per update (or downdate) has increased by a factor of 2.25 for the case when $m = 3$, or by $(.5m^2 + 1.5m)/(m+1)$ in general. The operations and timing diagram of the BS array are described in detail above.

The adaptive compact architecture of Fig. 11 has slightly more complicated cells than that of Fig. 7, but requires the same number of operations to check and incorporate a data set. However, the compact architecture processor may additionally be used to back-rotate a data set for adaptation. The forward and backward rotation modes have the same parallel complexity. Therefore, it is only necessary to add terms of the form $b\mathcal{O}(11m)$ to either (30) or (31) to account for back-rotation, where b has the usual meaning.

5 Conclusions

Two general contributions have been made to the theory and application of OBE algorithms for linear-in-parameters models. We have first suggested that all reported OBE algorithms, both nonadaptive and adaptive, can be placed into a general framework which is intimately related to recursive LSE processing. A flexible form of explicit adaptation has been demonstrated within this framework. In particular, a general technique based on "back-rotation" within the context of the QR-decomposition based version of WRLS offers a flexible array of adaptation strategies and good tracking ability. Secondly, two very different approaches to rendering a specific OBE algorithm, SM-WRLS, of $\mathcal{O}(m)$ per n computational complexity have been proposed. The computational complexity of the optimal OBE algorithms is of $\mathcal{O}(m^2)$ flops per n in spite of the highly discriminating data selection through set-membership criteria. This fact has not been made clear in the literature. This paper has demonstrated both an algorithmic and an architectural solution to this problem, making the SM-WRLS method superior to many other LSE techniques in a computational sense. In signal processing applications, this computational advantage is complemented the existence of the feasible set of solutions for which many other interesting purposes may be found.

References

[1] E. Walter (editor), *Mathematics and Computers in Simulation*, Special issue on parameter identifications with error bound, vol. 32, Dec. 1990.

[2] Cs. Bányász and L. Keviczky (editors), *Proceedings of the* 9^{th} *IFAC / IFORS Symposium on Identification and System Parameter Estimation*, vols. 1 & 2, Budapest, July 1991.

[3] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: A survey," *Mathematics and Computers in Simulation*, vol. 32, pp. 449-468, 1990.

[4] M. Milanese and A. Vicino, "Estimation theory for dynamic systems with unknown but bounded uncertainty: An overview," *Proceedings of the* 9^{th} *IFAC / IFORS Symposium on Identification and System Parameter Estimation*, vol. 2, pp. 859-867, Budapest, July 1991.

[5] J.R. Deller, Jr., "Set-membership identification in digital signal processing," *IEEE ASSP Magazine*, vol. 6, pp. 4-22, Oct. 1989.

[6] E. Fogel and Y.F. Huang, "On the value of information in system identification - Bounded noise case," *Automatica*, vol. 18, pp. 229-238, 1982.

[7] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Baltimore, MD: Johns-Hopkins Univ. Press, Ch. 6, 1983.

[8] J.P. Norton, *An Introduction to Identification*, London and Orlando, Florida: Academic Press, 1986.

[9] C.R. Johnson, *Lectures on Adaptive Parameter Estimation*, Englewood Cliffs, New Jersey: Prentice-Hall, 1988.

[10] D. Graupe, *Time Series Analysis, Identification, and Adaptive Systems*, Malabar, Florida: Krieger, Ch. 5, 1989.

[11] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, Cambridge, MA: MIT Press, pp. 14-15 & Sect. 2.2.1., 1983.

[12] T.C. Luk and J.R. Deller, Jr., "A nonclassical WRLS algorithm," *Proc. 23rd Ann. Allerton Conf.*, pp. 732-741, 1985.

[13] J.R. Deller, Jr. and D. Hsu, "An alternative adaptive sequential regression algorithm and its application to the recognition of cerebral palsy speech," IEEE Trans. Circ. and Syst., vol. CAS-34, pp. 782-787, July 1987.

[14] J.R. Deller, Jr. and G.P. Picaché, "Advantages of a Givens rotation approach to temporally recursive linear prediction analysis of speech," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 429-431, March 1989.

[15] J.R. Deller, Jr., "A 'systolic array' formulation of the optimal bounding ellipsoid algorithm," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 1432-1436, Sept. 1989.

[16] S.F. Odeh, *Algorithms and Architectures for Adaptive Set Membership-based Signal Processing* (Ph.D. Dissertation), Michigan State University, East Lansing, 1990.

[17] S.F. Odeh and J.R. Deller, Jr., "Systolic algorithms for adaptive set-membership identification," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '90*, Albuquerque, vol. 5, pp. 2419-2422, Apr. 1990.

[18] S.H. Mo and J.P. Norton, "Fast and robust algorithm to compute exact polytope parameter bounds," *Mathematics and Computers in Simulation*, vol. 32, pp. 481-493, 1990.

[19] V. Broman and M.J. Shensa, "A compact algorithm for the intersection and approximation of N-dimensional polytopes," *Mathematics and Computers in Simulation*, vol. 32, pp. 469-480, 1990.

[20] E. Walter, and H. Piet-Lahanier, "Exact recursive polyhedral description of the feasible parameter set for bounded-error models." *IEEE Trans. Automatic Control*, vol. AC-34, pp. 911-915, 1989.

[21] M. Milanese and G. Belaforte, "Estimation theory and uncertainty intervals evaluation in the presence of unknown but bounded errors: Linear families of models and estimators," *IEEE Trans. Automatic Control*, vol. AC-27, pp. 408-414, 1982.

[22] G. Belforte, B. Bona, and V. Cerone, "Parameter estimation algorithms for a set-membership description of uncertainty," *Automatica*, vol. 26, pp. 887-898, 1990.

[23] A.K. Rao, Y.F. Huang, and S. Dasgupta, "ARMA parameter estimation using a novel recursive estimation algorithm with selective updating," *IEEE Trans. Acoust. Speech and Signal Process.*, 38, pp. 447-457, March 1990.

[24] A.K. Rao and Y.F. Huang, "Recent developments in optimal bounding ellipsoid parameter estimation, *Mathematics and Computers in Simulation*, vol. 32, pp. 515-526, Dec. 1990.

[25] L. Pronzato, E. Walter, H. Piet-Lahanier, "Mathematical equivalence of two ellipsoid algorithms for bounded error estimation," *Proc. 28th IEEE Conf. on Decision and Control*, Tampa, pp. 1952-1955, 1989.

[26] J.R. Deller, Jr. and T.C. Luk, "Linear prediction analysis of speech based on set-membership theory," *Computer Speech and Language*, vol. 3, pp. 301-327, Oct. 1989.

[27] 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 Trans. Information Theory*, vol. IT-33, pp. 383-392, 1987.

[28] J.P. Norton, "Identification of parameter bounds for ARMAX models from records with bounded noise," *Int. J. Control*, vol. 45, pp. 375-390, 1987.

[29] Y.F. Huang, "A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-34, pp. 1331-1334, 1986.

[30] J.P. Norton, "Recursive computation of inner bounds for the parameters of linear models," *Int. J. Control*, vol. 50, pp. 2423-2430, 1989.

[31] A. Vicino and M. Milanese, "Optimal approximation of feasible parameter set in estimation with unknown but bounded noise," *Proceedings of the 9th IFAC / IFORS Symposium on Identification and System Parameter Estimation*, vol. 2, pp. 1365-1369, Budapest, July 1991.

[32] J.R. Deller, Jr. and S.F. Odeh, "SM-WRLS algorithms with an efficient test for innovation," *Proceedings of the 9th IFAC / IFORS Symposium on Identification and System Parameter Estimation*, vol. 2, pp. 1044-1049, Budapest, July 1991.

[33] J.P. Norton and S.H. Mo, "Parameter bounding for time-varying systems," *Mathematics and Computers in Simulation*, vol. 32, pp. 527-534, 1990.

[34] J.R. Deller, Jr., S.F. Odeh, and M. Nayeri, "Set-membership algorithms for signal processing," *Proc. of the IEEE*, in review.

[35] L.E. Dickson, *A First Course in the Theory of Equations*, New York: Wiley, 1952.

[36] J. Makhoul, "Linear prediction: A tutorial review," *Proc. of the IEEE*, vol. 63, pp. 561-580, 1975.

[37] W.M. Gentleman and H.T. Kung, "Matrix triangularization by systolic arrays," *Proc. Soc. Photo-Optical Instrumentation Engineers: Real-Time Signal Processing IV*, vol. 298, pp. 19-26, 1981.

[38] H.T. Kung and C. Leiserson, "Algorithms for VLSI processor arrays," Rpt. No. MU-CS-79-103, Dept. of Computer Sci., Carnegie-Mellon Univ., 1978. Reprinted in C. Mead and L. Conway, *Introduction to VLSI Systems*, Reading, MA: Addison-Wesley, Section 8.3, 1980.

[39] J.R. Deller, Jr. and S.F. Odeh, "Implementing the optimal bounding ellipsoid algorithm on a fast processor," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '89*, Glasgow, vol. 2, pp. 1067-1070, May 1989.

[40] J.G. McWhirter, "Recursive least squares minimization using a systolic array," *Proc. Soc. Photo-Optical Instrumentation Engineers: Real-Time Signal Processing VI*, vol. 431, pp. 105-112, 1983.

[41] W.M. Gentleman, "Least squares computations by Givens transformations without square roots," *J. Inst. of Math. and its Appl.*, Vol. 12, pp. 329-336, 1973.

[42] S.Y. Kung, *VLSI Array Processors*, Englewood Cliffs, NJ: Prentice Hall, Ch. 3, 1988.

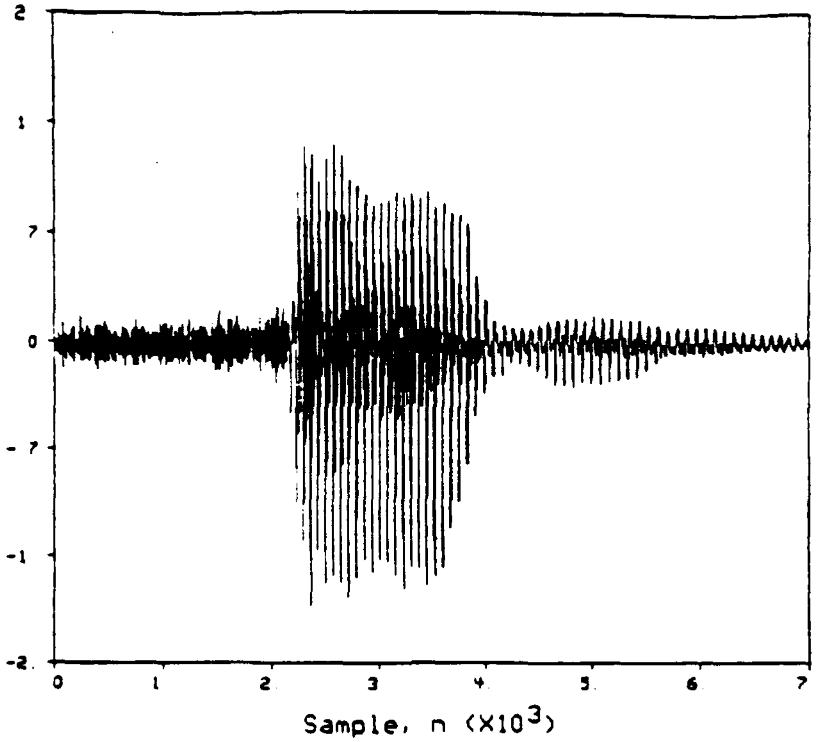


Figure 1: Acoustic waveform of the utterance "seven" upon which the time varying system in the simulation studies is based.

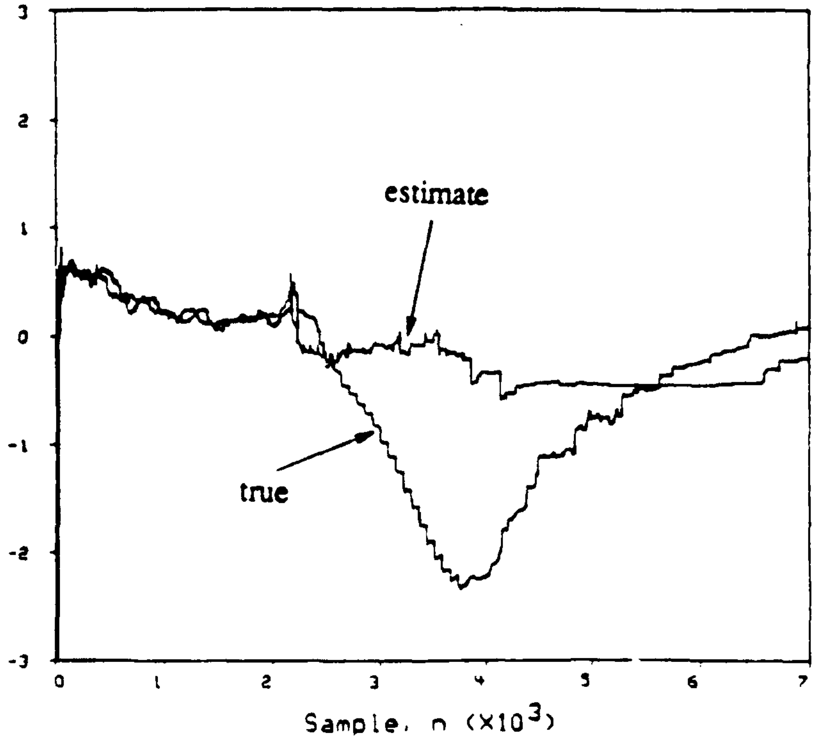


Figure 2: "Nonadaptive" SM-WRLS algorithm applied to the estimation of parameter a_{4*}. Only $\rho = 0.079$ of the data is used, but the estimate fails to track the true parameter.

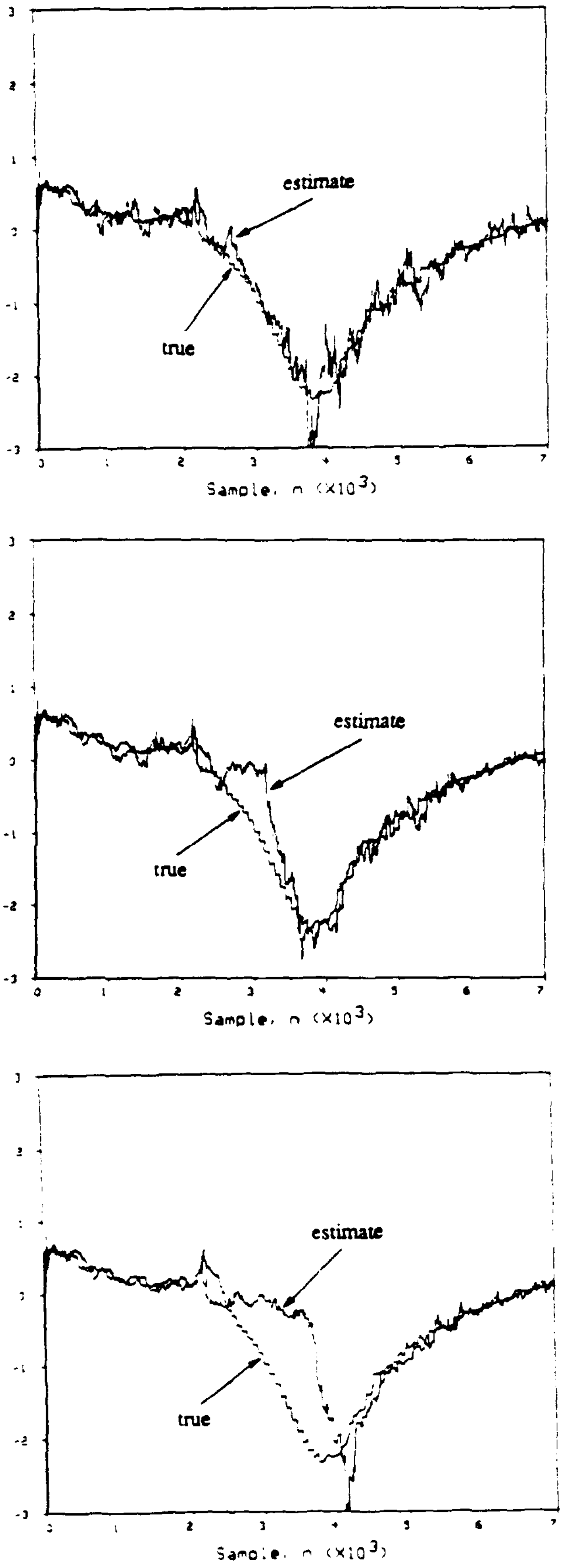


Figure 3: Windowed SM-WRLS with optimal data checking applied to the estimation of parameter a_4. The window lengths are (a) 500, (b) 1000, and (c) 1500 points, and the fractions (a) $\rho = 0.221$, (b) 0.174, and (c) 0.143 of the data are used in the estimation.

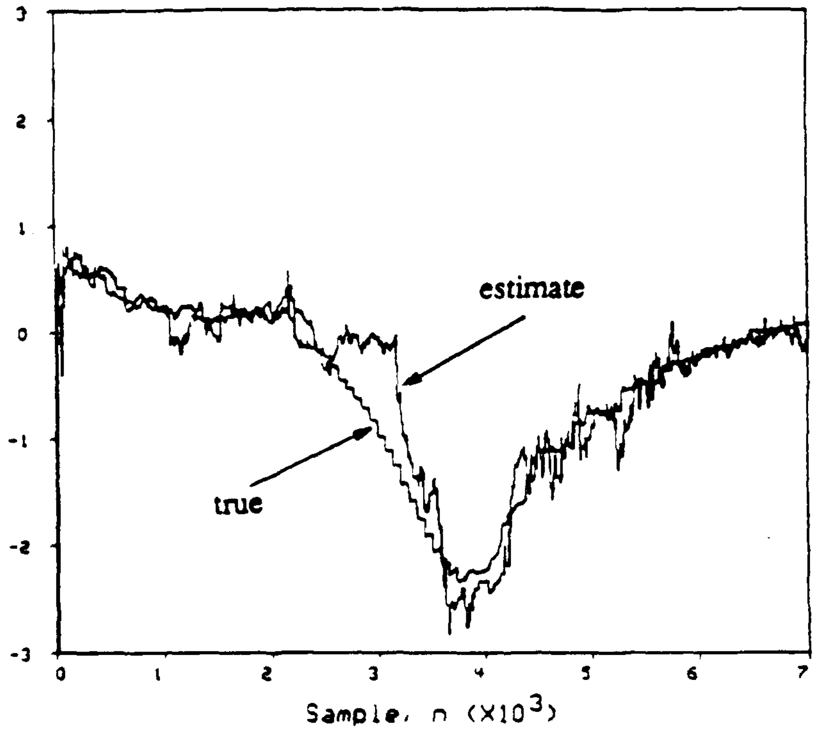


Figure 4: Windowed SM-WRLS with suboptimal data checking applied to the estimation of parameter a_{4*}. The window length is 1000 points and the fraction $\rho = 0.087$ of the data is used in the estimation.

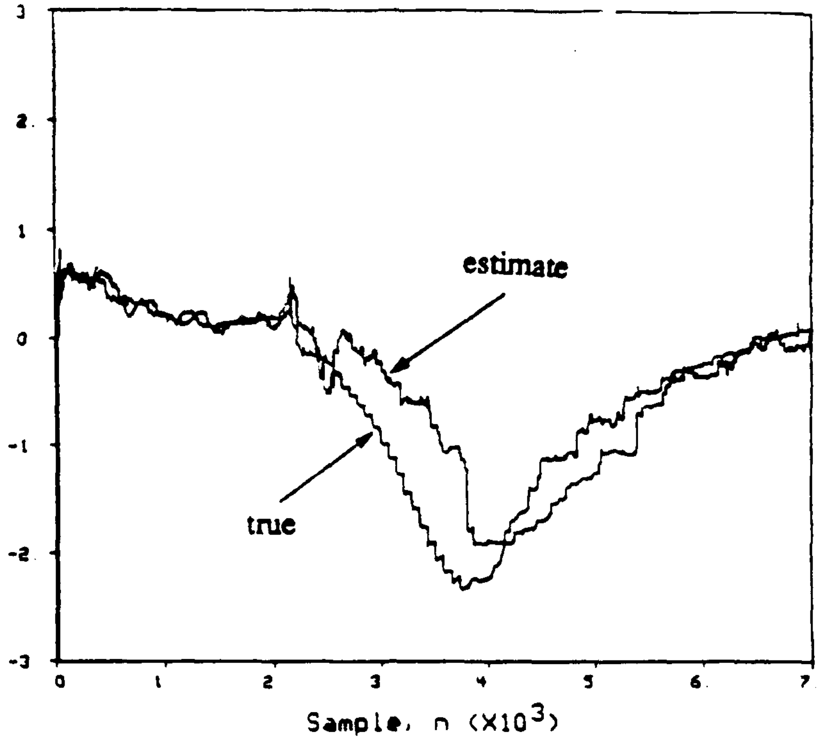


Figure 5: "Selective forgetting" SM-WRLS with optimal data checking applied to the estimation of the parameter a_{4*}. The criterion for selective removal of past points is described in the text. The fraction ρ = 0.129 of the data is used by the estimation procedure and the adaptation is computationally very inexpensive.

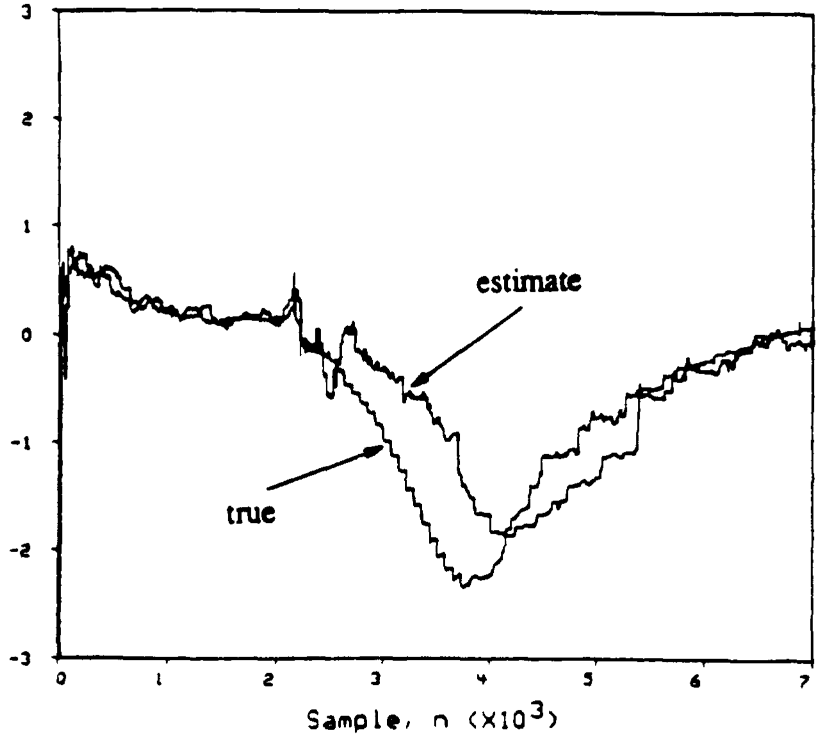


Figure 6: "Selective forgetting" SM-WRLS with suboptimal data checking applied to the estimation of the parameter a_{4*}. The criterion for selective removal of past points is described in the text. The fraction ρ = 0.088 of the data is used by the estimation procedure and the adaptation is computationally very inexpensive.

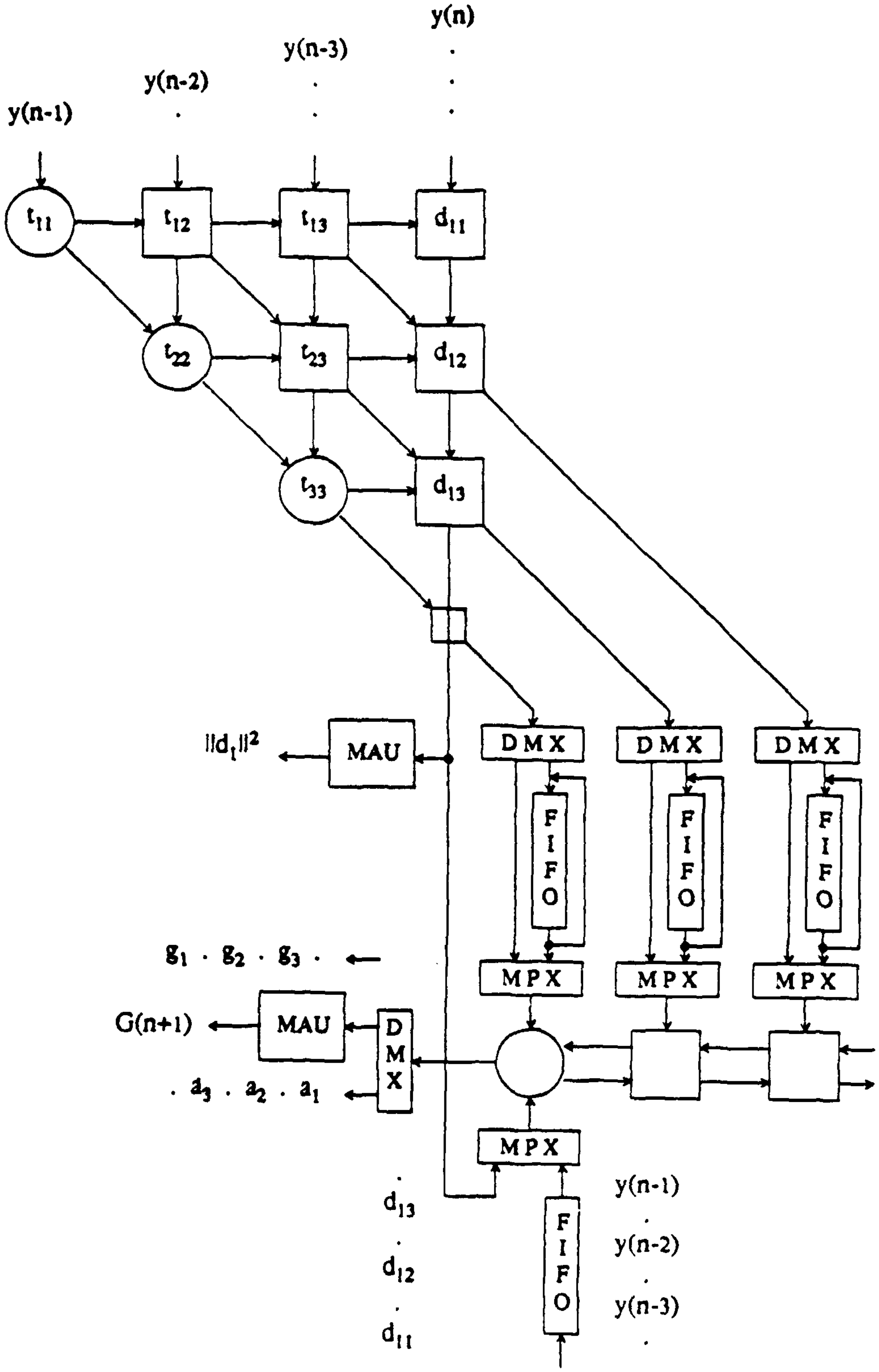
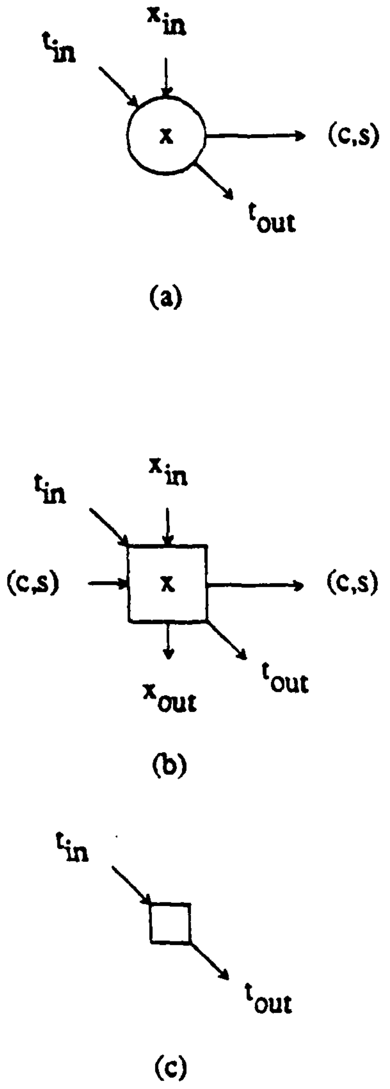


Figure 7: Systolic array implementation of the QR-WRLS based SM-WRLS algorithm. For simplicity, but without loss of generality, a pure autoregressive case of order three (AR(3)) is illustrated.



```
If ($x_{in} = 0$){
    $c = 1$
    $s = 0$
    $t_{out} = t_{in}$
}
else{
    $temp = [x^2 + (x_{in})^2]^{1/2}$
    $c = x / temp$
    $s = x_{in} / temp$
    $x = temp$
    $t_{out} = x$
}
```

```
If ($x_{in} = 0$ & $c = 1$ & $s = 0$)
    $t_{out} = t_{in}$
else{
    $x = c\,x + s\,x_{in}$
    $x_{out} = -s\,x + c\,x_{in}$
    $t_{out} = x$
}
```

$t_{out} = t_{in}$

When the array is used as LIFO stacks: First cycle: $t_{out} = x$ (t_{ij} cells)

$x_{out} = x$ (d_1 cells)

All following cycles: $t_{out} = t_{in}$ (t_{ij} cells)

$x_{out} = x_{in}$ (d_1 cells)

Figure 8: The operations performed by the cells used in the triangular array of Fig. 7. (a) The Givens generation (GG) cells, (b) the Givens rotation (GR) cells, and (c) the delay element.

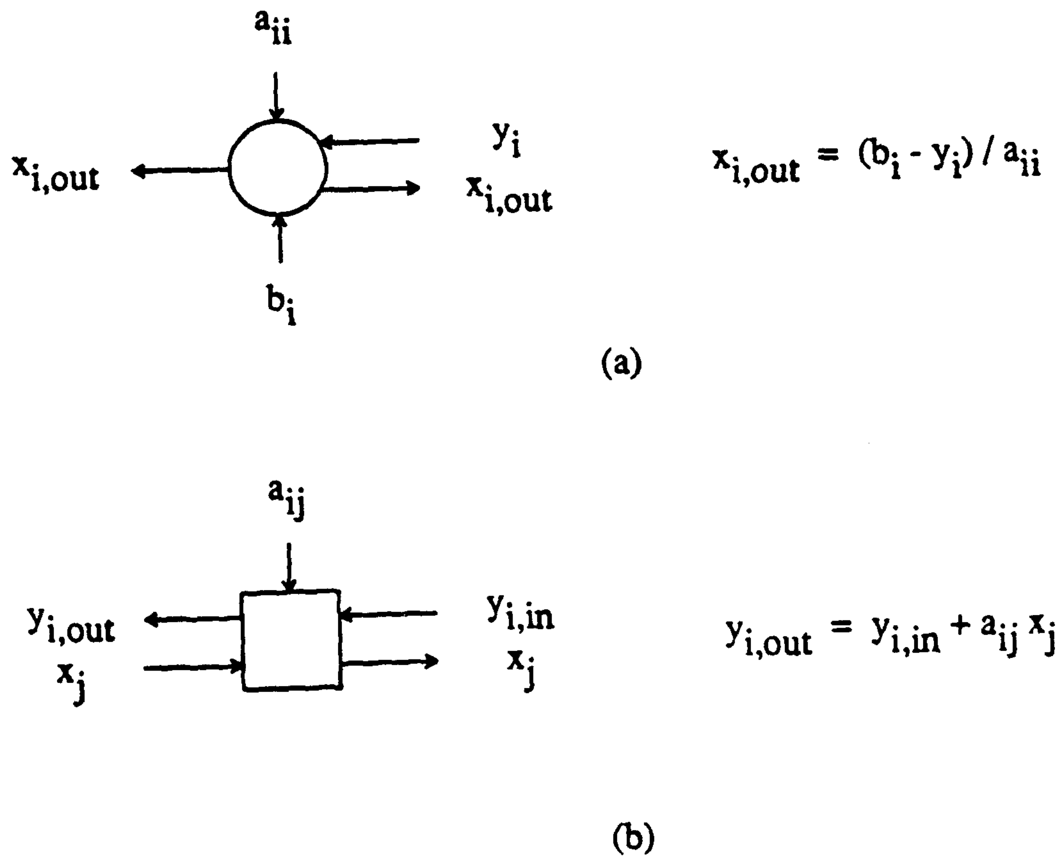


Figure 9: Operations performed by the back substitution array. (a) The left-end processor and (b) the multiply-add units. The initial $y_{i,in}$ entering the rightmost cell is set to 0.

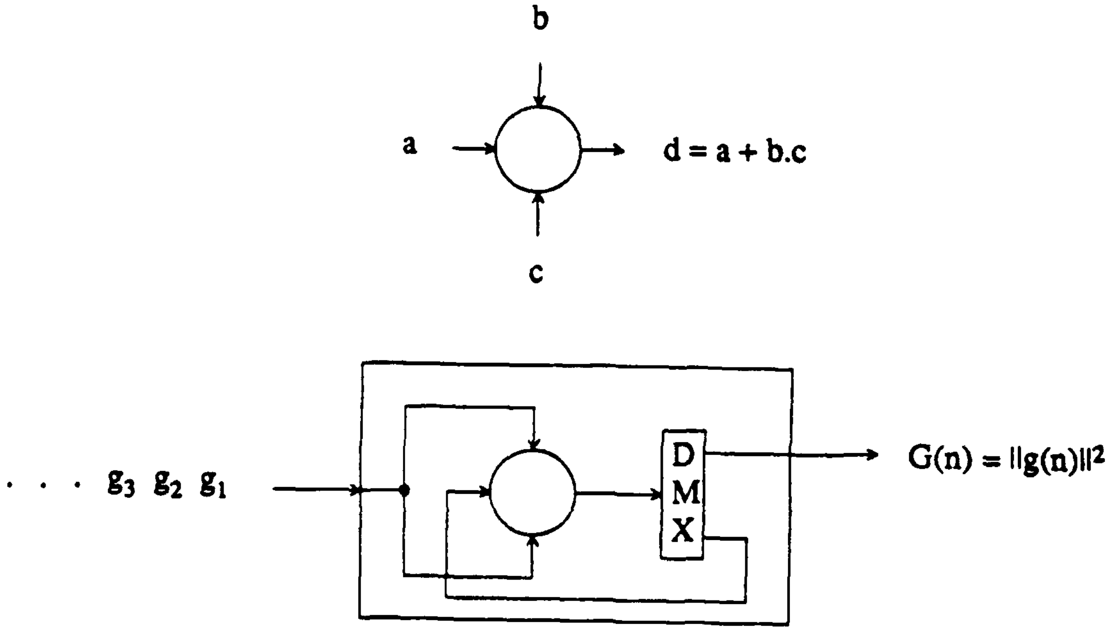


Figure 10: Multiply-add unit used in Fig. 7.

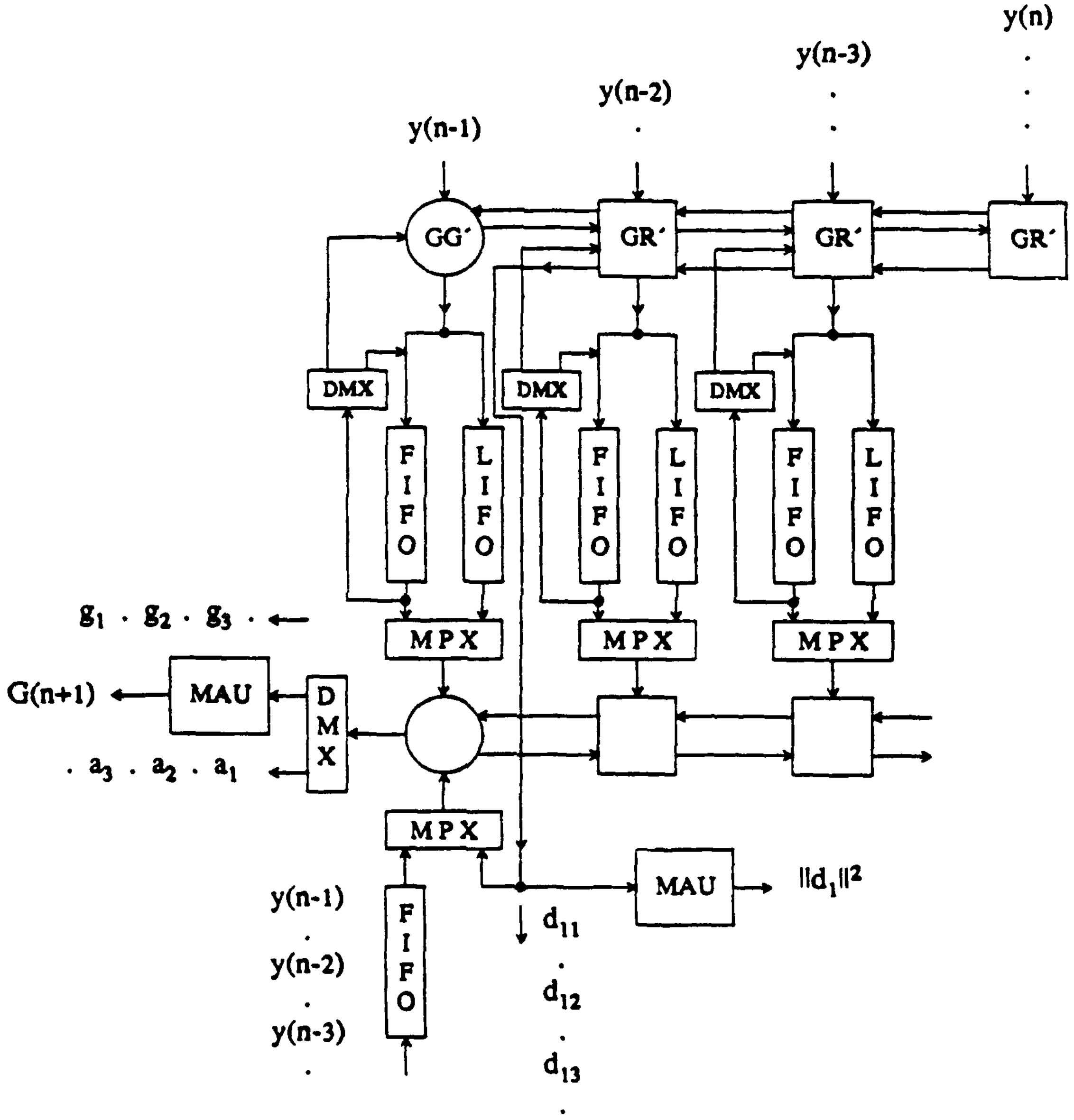
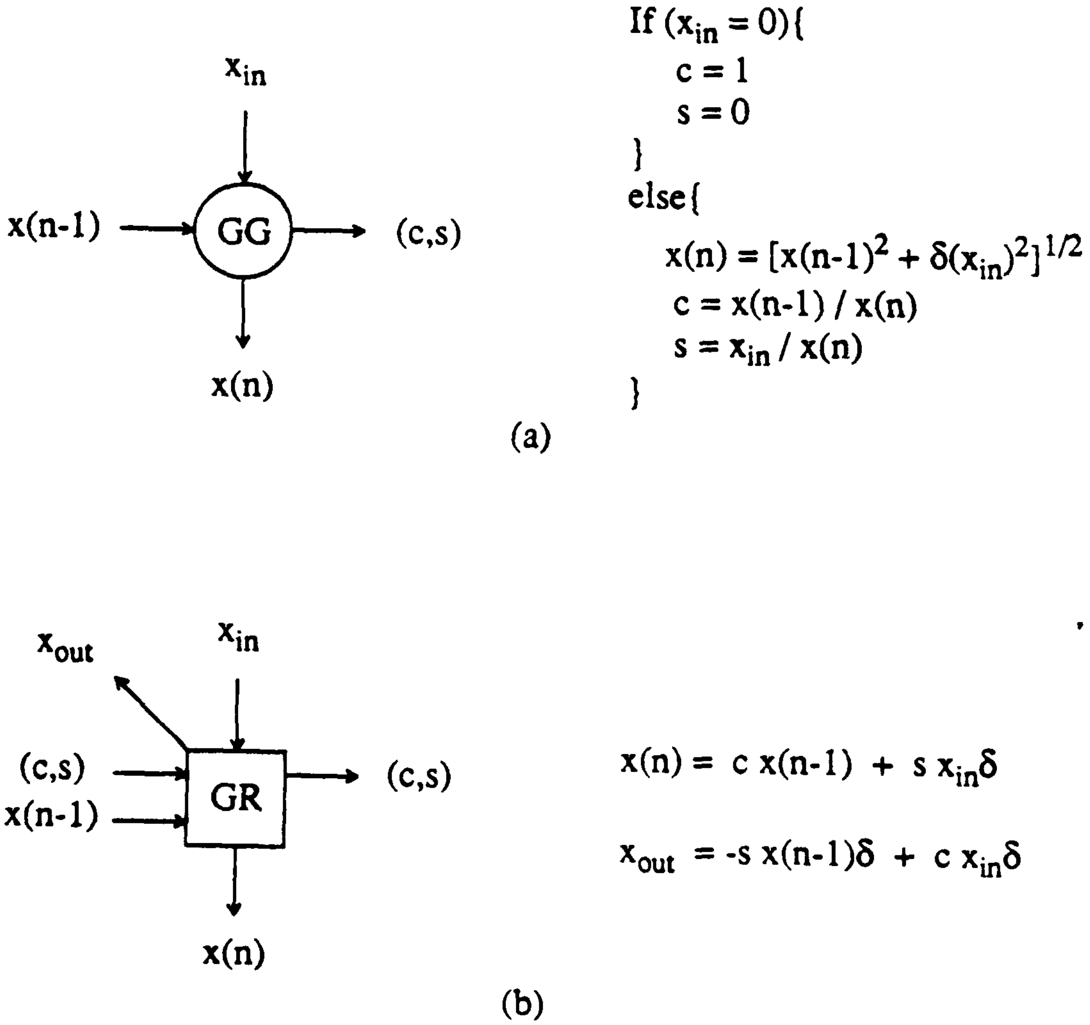


Figure 11: A compact architecture implementation of the adaptive SM-WRLS algorithm.



```
If ($x_{in} = 0$){
    $c = 1$
    $s = 0$
}
else{
    $x(n) = [x(n-1)^2 + \delta(x_{in})^2]^{1/2}$
    $c = x(n-1) / x(n)$
    $s = x_{in} / x(n)$
}
```

(a)

$$x(n) = c\, x(n-1) + s\, x_{in}\delta$$

$$x_{out} = -s\, x(n-1)\delta + c\, x_{in}\delta$$

(b)

Figure 12: Operations performed by (a) the GG and (b) the GR cells used in the modules of Fig. 11. $\delta = +1$ (-1) for rotating the data set into (out of) the system.

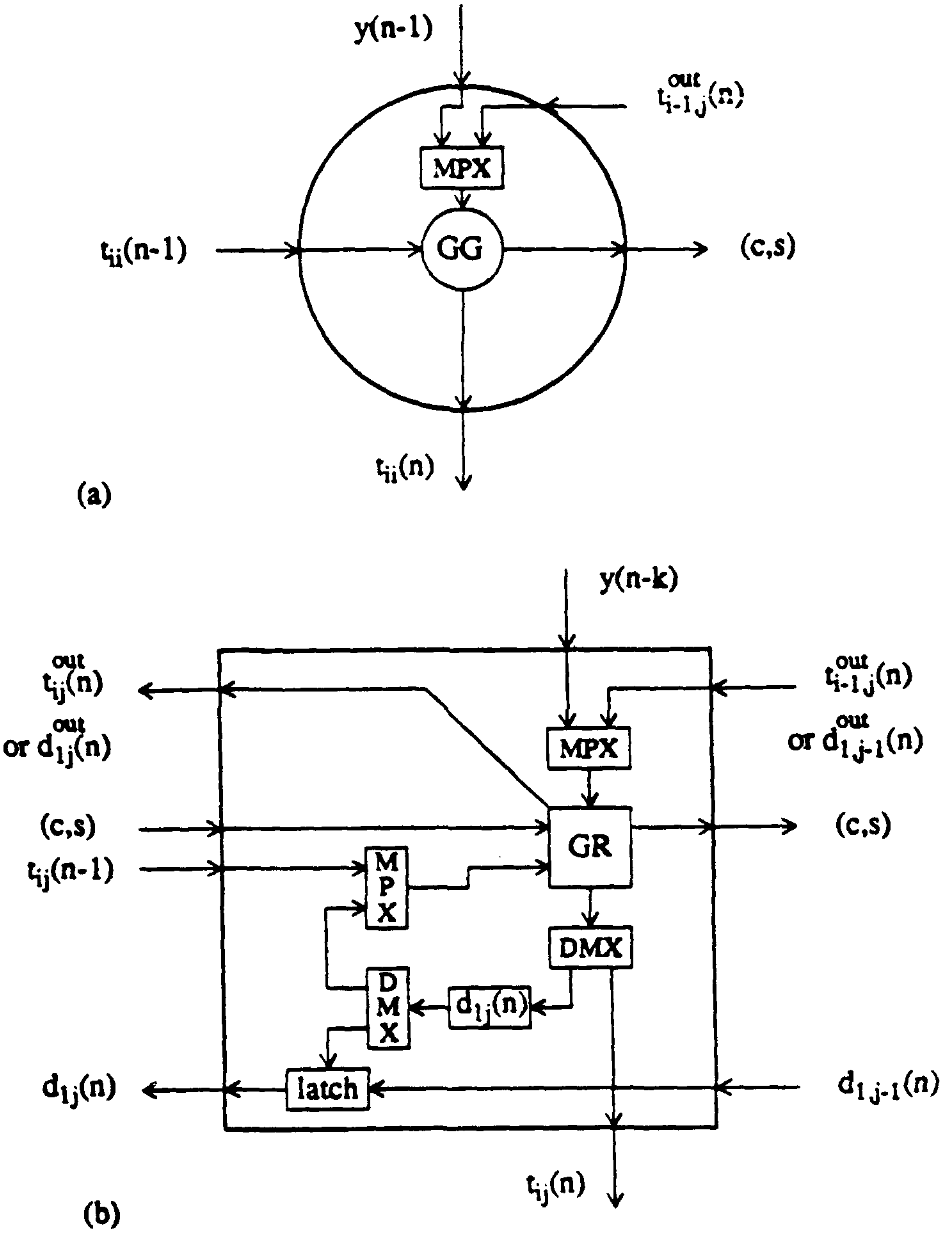


Figure 13: (a) GG′ module and (b) GR′ module used in the architecture of Fig. 11.

	Inputs				Outputs			
Time	○	□	□	□	$[\mathbf{T}(n) \mid \boldsymbol{d}_1]$			
0	$y(n-1)$							
1		$y(n-2)$			t_{11}			
2			$y(n-3)$			t_{12}		
3				$y(n)$		t_{22}	t_{13}	
4							t_{23}	d_{11}
5							t_{33}	d_{12}
6								d_{13}

Table 1: Timing diagram of the triangular array of Fig. 7.

	Inputs			Outputs
Time	○	□	□	○
0	$t_{11}, y(n-1)$			
1	t_{33}, d_{13}	t_{12}		g_1
2	$t_{22}, y(n-2)$	t_{23}	t_{13}	$\hat{a}_3$
3	t_{22}, d_{12}	t_{23}	t_{13}	g_2
4	$t_{33}, y(n-3)$	t_{12}		$\hat{a}_2$
5	t_{11}, d_{11}			g_3
6				$\hat{a}_1$

Table 2: Timing diagram of the back substitution array of Fig. 7.

Element Computed	flops per n
$\varepsilon(n+1, \boldsymbol{\theta}(n))$	$m+1$
coefficient of quadratic (XX)	7
$\lambda^{*}_{n+1}(n+1)$	$5+\sqrt{\ }m+1$
$G(n+1)$ and $\boldsymbol{\theta}(n)$	$2m+1$
If data set is accepted:	
update	$m+1+\sqrt{\ }$
Givens rotations	$5(2m+1)$
$\kappa(n)$	4

Table 3: Numbers of operations required by the GG and GR cells in the architecture of Fig. 7.

	Inputs				**Outputs**			
Time	○	□	□	□	○	□	□	□
0	$y(n-1)$							
1		$y(n-2)$			t_{11}			
2			$y(n-3)$			t_{12}		
3				$y(n)$	t_{22}		t_{13}	
4						t_{23}		d_{11}
5					t_{33}		d_{12}	
6						d_{13}		

Table 4: Timing diagram of the GR array of the compact architecture of Fig. 11.

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
345 EAST 47TH STREET, NEW YORK, N.Y. 10017-2394, U.S.A. TELEX 236411 FAX (212) 705-7682

DIRECT NUMBER (212) 705-
7557/7558

2 January 1992

Prof. John R. Deller, Jr.
258 EB
Dept of Electrical Engineering
Michigan State University
East Lansing, MI 48824-1226

Paper: "System Identification Using Set-Membership Based Signal Processing" by J. R. Deller, M. Nayeri & S. F. Odeh

Dear Prof. Deller:

I am writing to acknowledge, with thanks, receipt of the abovementioned paper, which we encouraged you to prepare for submission to the PROCEEDINGS OF THE IEEE.

We are seeking the advice of a member of our Editorial Board on the review of the paper, after which we will send the manuscript to the reviewers designated. We will contact you again when the review has been completed.

The IEEE policy is to hold copyright to the material it publishes. The attached form gives information on this subject. Kindly sign it, or have it signed by an authorized person in your organization, and return it immediately so that your contribution is eligible for acceptance for publication. Note that this transfer agreement has no effect if the manuscript is not accepted for publication or is withdrawn.

The original illustrations and the photograph and biography of the author(s) will be required if the paper is accepted. If these items have not already been submitted, please be prepared to supply them promptly upon notification of acceptance of the paper.

Sincerely,

W. Reed Crone
Executive Editor
PROCEEDINGS OF THE IEEE

enc.

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
345 EAST 47TH STREET, NEW YORK, N.Y. 10017-2394, U.S.A. TELEX 236411 FAX (212) 705-7682

DIRECT NUMBER (212) 705- 7557

February 22, 1991

Prof. John R. Deller, Jr.
258 EB
Dept. of Electrical Engineering
Michigan State University
East Lansing, MI 48824-1226

Proposal: "Estimation and Identification Using Set-Membership Based Signal Processing" by J. R. Deller, Jr. & S. F. Odeh

Dear Prof. Deller:

I am writing further to my letter of 13 February concerning your proposal for a paper on the above-mentioned subject. The PROCEEDINGS Editor has asked me to let you know that the proposal has been favorably reviewed and that we would like to encourage you to prepare the paper for submission to the PROCEEDINGS.

We are taking your paper into account in our plans for future issues of the PROCEEDINGS, and we will expect to receive a manuscript from you in the next 6 to 9 months. If you feel that the preparation will require more time than this, please let us know.

Although our goal of covering a wide variety of topics makes it necessary to limit the length of papers, the PROCEEDINGS devotes more pages on the average to each paper than do the TRANSACTIONS. We feel that a maximum of 15,000 words are sufficient for most subjects.

It is important that papers in the PROCEEDINGS can be understood not only by the specialist but by the generalist as well. This means that each tutorial-review paper should contain an extensive introduction to the field, written at a level suitable for the reader not familiar with the topic, as well as a comprehensive bibliography for those who want to probe further. In between, the substance of the paper should be a review of the current state of progress in the field -- not only of the author's own work, but that of others as well.

I would like to mention the commitment the PROCEEDINGS will have to your paper. We will have a definite predisposition to its publication; however, all of our papers must undergo review before actual acceptance for publication. The reviewers often suggest changes which will improve the paper. Naturally, we must reserve the right to turn down a paper which our reviewers feel has technical or literary shortcomings.

I would like to say again that we are very interested in your paper and look forward to seeing the completed manuscript. Please keep us informed of any changes in the schedule we have suggested.

Sincerely,

W. Reed Crone
Executive Editor
PROCEEDINGS OF THE IEEE

System Identification Using
Set-Membership Based Signal Processing

J.R. Deller, Jr., M. Nayeri, and S.F. Odeh[‡]

Michigan State University
Department of Electrical Engineering
CSSP Group: Speech Processing & Adaptive Signal Processing Laboratories
East Lansing, MI 48824-1226

email: deller@ee.msu.edu phone: (517) 353-8840 FAX. (517) 353-1980

[‡]S.F. Odeh is currently with the University of Jordan, Department of Electrical Engineering, Amman.

Abstract

This paper is concerned with *set-membership (SM)* identification which refers to a class of algorithms which uses certain *a priori* knowledge about a parametric model to constrain the solutions to certain sets. The emerging field of SM-based Signal Processing is receiving considerable attention and is becoming increasingly popular around the world. This paper initially surveys the types of problems and solutions being researched, then focusses on identification techniques of particular currency in the signal processing field. Specifically, the case in which bounds on the model errors are known has be of particular interest to SM researchers. We show that these "bounded error" (BE) algorithms can be combined with various forms of least square error (LSE) signal processing algorithms with interesting and beneficial consequences. A general framework embracing all currently used BE/LSE algorithms is developed, then strategies for adaptation and for implementation on parallel machines are discussed. Computational complexity benefits are considered for the various algorithms. The paper is tutorial, leaving many of the formal details to appendices which presents a theoretical treatment of the key results. These appendices serves to unify many related results appearing in the literature.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. MIP-9016734 and by the Office of Naval Research under Contract No. N00014-91-J-1329. The authors wish to thank Mr. Marwan Krunz for assistance with the simulation studies.

Contents

Note: All blanks below are filled in the captions with the actual figure.

List of Figures

List of Tables

1 Introduction to Set-Membership-Based Signal Processing

System identification is concerned with the deduction of a mathematical model of a dynamical system based on measurable signals and other attributes of the physical situation. The principal focus of this paper will be upon the paradigm in which sufficient information exists to specify a good ("true") parametric *form* for the underlying dynamics, and the identification problem is reduced to correctly parameterizing the mathematical form.

In a broad sense, *set-membership (SM) identification* is concerned with the description of *sets* of parameter solutions which are consistent with the measurements and the modelling assumptions. Accordingly, SM identification is sometimes called *parameter bounding identification* or similar names. The name "SM" identification derives from the fact that an SM algorithm, in principle, ascertains whether a particular parameter vector is a *member* of the *feasible set.*

SM identification is novel in its pursuit of *set* solutions rather than *particular* solutions which are sought by conventional methods. A feasible set which arises as a consequence of SM processing is a reflection of the assumptions made about the "true" model, and its "size" is inversely proportional to the amount of information available about the "true" model. The fundamental benefit of this approach is that it yields solutions which are based only upon *tenable* modelling assumptions. A set of solutions consistent with known information can be preferable to, or complemetary to, a *single solution* based on tenuous assumptions. If an appropriate SM algorithm exists, it is only necessary to have sufficient modelling information to provide a sufficiently small feasible set for a given purpose. For example, a resulting set might be small enough so that its centroid would provide a good model in some application.

We remark that the given description of an SM identification algorithm above does not necessarily exclude methods whose solutions are a single parameter vector. However, a consistent theory requires that, as time progresses, feasible sets be subsets of their predecessors (see below). The class of algorithms that produces an invariant single point estimate is certainly a very uninteresting one. We will discuss this point with respect to the conventional *least square error (LSE)* solution in the paper.

After discussing the general modelling and identification issues and defining notation in Section 2, this paper will focus on four classes of identification problems:

1. *"Other" SM Problems.* The principal focus of this paper is upon SM methods currently being employed in signal processing applications. In Section 3 we begin a taxonomy of SM methods and survey problems which are outside the scope of the present paper.

2. *The LSE Problem.* The purpose of this brief discussion in Section 4 will be to view this well-known problem in relation to the SM approach in preparation for further developments.

3. *The Bounded Error (BE) Problem.* We continue our discussion of the taxonomy of SM methods in

Section 5 with the BE problem. This class of SM algorithms is predicated upon a model with additive errors whose magnitudes are assumed bounded. A vast majority of the research on SM identification to date has focused on this problem and a variety of algorithms has resulted. The purpose here will be to introduce the problem and review this body of research. At the "bottom" of the BE class of techniques, we will encounter the *ellipsoid bounding* algorithms in which the LSE and BE problems interface.

4. *Combined LSE / BE Problem.* The heart of this paper is Section 6 in which we formulate and discuss the *Unified Optimal Bounding Ellipsoid (UOBE)* algorithm which represents an *explicit* combination of the two classes of problems above for the linear parametric model. The UOBE algorithm is actually a *class* of algorithms that embraces many LSE/BE algorithms proposed in the literature. It will be discovered that the benefits of combining BE considerations (when they are known) with LSE processing are twofold: First, the BE information provides a feasible set of solutions which complements the unique LSE ("infeasible" by virtue of its uniqueness) estimate. This feasible set can help to compensate for the extremely restrictive nature of the assumptions placed upon the LSE model. A colored noise sequence, for example, represents a violation of the basic tenets of LSE modelling which might be ameliorated by the BE considerations. Secondly, it will be shown how BE knowledge can greatly *improve the efficiency* of LSE identification.

In its focus on Problem 4 above, this paper provides a what might be called a "signal processing" perspective on the field of SM identification. By this we mean that we approach the problem with a predisposition toward linear models and LSE processing which are firmly entrenched and successfully employed in many signal processing applications. The authors' principal interest in SM theory has been its implications for complexity improvement, architectures, adaptation, and bias-reduction in linear LSE algorithms. The work of Fogel, Huang and colleagues [24],[41],[49],[51],[99]-[101] also falls into this realm and this relationship will be explored in detail. A different perspective on this field is provided by the work of a number of research groups in Europe, most of whom approach SM identification with an interest in control and system science. These researchers, whose work will be discussed in the material to follow, have focused on a broad array of algorithms and models, mostly in conjunction with the BE constraint. This work has tended to focus on the development and analysis of novel, sometimes very complex, identification algorithms for bounding feasibility sets. While extremely interesting, this work has not yet yielded methods which are as immediately applicable to practical problems as the well known LSE approaches discussed here[1]. We review these "BE" developments in Section 5 and direct the reader to specific information about this interesting work. Recent surveys of the SM field which focus on the BE research are found in papers

[1] This is not to say that applications have not occured. Some examples are cited below.

by Walter and Piet-Lahanier [121], and by Milanese and Vicino [75]. Both survey papers contain extensive and useful reference lists. The reader interested in a much lighter tutorial on a specific form of UOBE algorithm, the "SM-WRLS" algorithm, is directed to the paper by Deller [27].

Whatever the particular interest in pursuing SM algorithms, it is clear that all researchers are excited about their tremendous potential for application to problems of practical importance. Milanese and Vicino [75], for example, list a broad range of areas to which SM techniques have been applied. Among them are applications to biology and chemistry [14],[82], [130]; pharmacokinetics [44],[73]; time series analysis [118]; economic modelling [76]; speech and image processing [29],[30],[108],[111]; ecology [58],[109]; measurement [11],[13],[104],[110]; and robust adaptive control [2],[24],[39],[55],[60],[62],[67],[112],[117]. Recently, artificial neural network training algorithms have been the subject of studies involving the SM algorithms [25],[50]. SM algorithms have also been explored with regard to their tracking ability for *adaptive* identification [16],[17],[24],[32],[86],[87]-[89]. Finally, another novel way in which BE methods have been applied is to the problem of model *structure* identification [114]. Because of this significant potential for application, SM algorithms continue to be the subject of intense research effort.

2 Formalities

In this brief section, we formally define notation for the identification problem to be studied and discuss some important aspects of the modelling problem.

2.1 General Identification Problem

The general modelling setup employed in the discussion as follows: We assume that we are observing some physical discrete time system which is generating a complex-valued vector sequence, $\boldsymbol{y}(\cdot)$ of dimension k, in response to complex vector-valued input $\boldsymbol{u}(\cdot)$. The sequence $\boldsymbol{u}(\cdot)$ is assumed to be a realization of an ergodic, wide sense stationary stochastic process. Both input and output sequences are measurable. The consideration of a complex, *multiple-input—multiple-output (MIMO)* system will generalize many of the results found in the literature. Of course, the real or complex *single-input—single-output (SISO)* system is contained in this analysis as a special case. Although many of the developments in the literature are explicitly for SISO systems, these are trivially generalized to an arbitrary (finite) number of inputs (MISO). Of the remaining developments, most are concerned with SO, but implicitly or explicitly MI, hence MISO, systems. With regard to the dimensions of inputs and outputs, therefore, the developments here differ from previously published results principally in terms of the generalized number of outputs. Upon occassion, we will wish to discuss a result from the literature. In this case we shall remark that we are dealing with a SISO or MISO system, and let $k = 1$ and the output and error (defined below) be denoted in regular typeface, $y(\cdot)$ and $\varepsilon(\cdot)$, to denote scalars. Though SI systems do occur in such discussions, we shall not have occassion to use the scalar notation for the input sequence.

At time n, mathematical model of the form

$$\boldsymbol{y}(t) = \boldsymbol{\Psi}[t, \boldsymbol{\Theta}(n), \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\varepsilon}, p, q, r] + \boldsymbol{\varepsilon}(t, \boldsymbol{\Theta}(n)) \tag{1}$$

is proposed to account for the dynamics of the physical system. For any time t, $\boldsymbol{\Psi}$ is a k-vector of functions of the "present" input $\boldsymbol{u}(t)$, and p, q, and r lags of the sequences $\boldsymbol{y}(\cdot)$, $\boldsymbol{u}(\cdot)$, and $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}(n))$, respectively. $\boldsymbol{\Psi}$ is parameterized by a matrix $\boldsymbol{\Theta}(\mathrm{n})$, and $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}(n))$ is a complex k-vector error sequence which depends upon the parameterization. In all models of interest in SM analysis, the *additive* error sequence appears. In general, the model will depend upon the time n at which we are constructing the model (we may have different information at different times). As we shall see below, however, the only unknown in the model will be the parameterization. Hence, the dependence of the model upon n will arise through the parameters alone. Accordingly, we show $\boldsymbol{\Theta}(n)$ as a function of the modelling time n.

It is assumed that a "true" time-invariant model, of form, say,

$$\boldsymbol{y}(t) = \boldsymbol{\Psi}_*[t, \boldsymbol{\Theta}_*, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\varepsilon}_*, p_*, q_*, r_*] + \boldsymbol{\varepsilon}_*(t), \tag{2}$$

is *exactly* accounts for the observed dynamics. While the *form* of $\boldsymbol{\Psi}_*$ is known, the "true" parameters, $\boldsymbol{\Theta}_*$, are unknown and must be sought by the identification. Naturally, we take $\boldsymbol{\Psi}$, p, q, and r of the proposed model to be equivalent to their "true" counterparts. The "true" noise sequence, $\boldsymbol{\varepsilon}_*(\cdot)$, is generally not known on a sample-by-sample basis, but certain of its properties are known (e.g., second order statistical properties) and are attributed to the proposed model error, $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}(n))$. Whether "local" information about the error sequence is available or not is one of the distinguishing characteristics of a SM identification problem. Frequently, identification approaches (in particular, the LSE approach) are based on asymptotic properties of the sequence $\boldsymbol{\varepsilon}_*(\cdot)$. Asymptotic properties fail to provide pointwise information with which to pare down the space of parameter estimates. For example, second order statistical properties of the sequence do not provide much specific information about the value $\boldsymbol{\varepsilon}_*(n)$ at a particular n. This is to be contrasted with a SM problem, in which known attributes of the error sequence (or, infrequently, of other aspects of the model) are available at every modelling time n. The following problem statement reflects this class of constraints.

Problem 1 (General SM Identification Problem) *Observations $\boldsymbol{y}(t)$, $\boldsymbol{u}(t)$, $t \in [1, n]$ are "known" to have been generated by a "true" model of form (2) whose error sequence has a specified set of attributes, say $\mathcal{A}_n$, on that time range. We propose a model of form (1) with $\boldsymbol{\Psi} \equiv \boldsymbol{\Psi}_*$, $p = p_*$, $q = q_*$, $r = r_*$, whose parameters, $\boldsymbol{\Theta}_*$, are unknown, but whose error sequence has properties $\mathcal{A}_n$ on the given time range. Find the feasible set of parameters, $\Omega(n)$, such that for each $\boldsymbol{\Theta} \in \Omega(n)$, the proposed model is consistent with the observations.*

A SM problem will be said to be *ill-posed* if

$$\Omega(n+1) \not\subseteq \Omega(n), \quad n = 1, 2, \ldots \tag{3}$$

If (3) were not true, it would be the case that there exists a potential parameterization of the "true" model which is consistent with the observations on $t \in [1, n+1]$ but not those on $t \in [1, n]$. In turn, this implies the potential for a time–varying "true" system, in violation of the assumption about this system. This is an indication that there is something inconsistent in the specification of the error attributes, or that the data do not conform to the assumed "true" model.

2.2 "LP" *vs.* "non-LP" Models

Models of form (1) can be dichotomized into those which are are linear in the parameters (LP) sought, and those which are not (non-LP). With regard to the general form (1), we see that any model in which $\boldsymbol{\Psi}$ has explicit nonlinear terms in the matrix $\boldsymbol{\Theta}(n)$ is immediately non-LP. For example[2],

$$\boldsymbol{y}(t) = \boldsymbol{\Theta}^H(n)\boldsymbol{A}(t)\boldsymbol{\Theta}(n) + \boldsymbol{\varepsilon}(t, \boldsymbol{\Theta}(n)) \tag{4}$$

where $\boldsymbol{A}(t)$ is some $m \times m$ matrix of functions of the lags of $\boldsymbol{y}(\cdot)$ and $\boldsymbol{u}(\cdot)$, is clearly non-LP. A model cannot be LP, therefore, unless it can be written in the form

$$\boldsymbol{y}(t) = \boldsymbol{\Theta}^H(n)\boldsymbol{x}(t) + \boldsymbol{\varepsilon}(t, \boldsymbol{\Theta}(n)). \tag{5}$$

This is a necessary, but not a sufficient condition for a model to be LP, however. second necessary condition is that the vector sequence $\boldsymbol{x}(\cdot)$ contain no functions which have samples of the error sequence $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}(n))$ as arguments. One frequent occurrence of this non-LP type of mapping appears in the so-called *output error* model (e.g. [52],[121]). For a SISO system[3] the output error model takes the form (5) (with $\boldsymbol{y}(\cdot) = y(\cdot)$, $\boldsymbol{u}(\cdot) = u(\cdot)$, and $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}) = \varepsilon(\cdot, \boldsymbol{\Theta})$ scalars), where

$$\boldsymbol{x}(t) = [\hat{y}(t-1) \ \cdots \ \hat{y}(t-p) \ u(t) \ u(t-1) \ \cdots \ u(t-q)]^H \tag{6}$$

in which $\hat{y}(\cdot)$ represents the sequence $y(\cdot) - \varepsilon(\cdot, \boldsymbol{\Theta})$. A second important non-LP model is the SISO *autoregressive - moving average with exogenous input (ARMAX)* model which is of form (5) with

$$\boldsymbol{x}(t) = [y(t-1) \ \cdots \ y(t-p) \ u(t) \ u(t-1) \ \cdots \ u(t-q) \ \varepsilon(t-1, \boldsymbol{\theta}) \ \cdots \ \varepsilon(t-r, \boldsymbol{\theta})]^H \tag{7}$$

[2] Throughout, superscript H denotes the Hermitian transpose.

[3] Henceforth, whenever a SISO system is mentioned in the paper, it is implicit that the model signals and parameters are *real*. This is for two simple reasons: 1. To avoid superfluous details and notation, and 2. To accurately represent other research. The general results of this paper are perfectly applicable to the *complex* SISO case. In the real case, "H" denotes the real transpose.

where $r \geq 1$. The *autoregressive - moving average (ARMA)* model is a special case of the ARMAX with no u terms present in (7). Details on these non-LP models are found, for example, in [45],[52],[69].

LP models are characterized by difference equations of form (5) in which $\boldsymbol{x}(t)$ is any m-vector of functions of the lags of $\boldsymbol{y}(\cdot)$ and $\boldsymbol{u}(\cdot)$ at time t. A special SISO case is the *autoregressive with exogenous input (ARX)* model in which

$$\boldsymbol{x}(t) = [y(t-1) \cdots y(t-p)\ u(t)\ u(t-1) \cdots u(t-q)]^H . \tag{8}$$

It is conventional to denote the parameters of the ARX model by

$$\boldsymbol{\Theta} = [a_1 \cdots a_p\ c_0\ c_1 \cdots c_q]^H \tag{9}$$

so that the ARX system can be described in terms of the difference equation

$$y(t) = \sum_{i=1}^{p} a_i y(t-i) + \sum_{j=0}^{q} c_j u(t-j) + \varepsilon(t,\boldsymbol{\Theta}). \tag{10}$$

A pure *autoregressive (AR)* model is a special case of the ARX model in which no u terms appear.

3 "Other" SM Problems

Before turning to the main SM problems of interest, we return to the broad definition of SM identification given in the opening paragraphs and note the potential for many other types of algorithms within the framework of the SM algorithm definition.

A taxonomy of SM methods is shown in Fig. 1. SM techniques are seen to be first subdivided into those concerned with bounding parameters of input-output descriptions of systems (identification), and those dealing with bounding state estimates in state space formulations (state estimation). While it is the former class of techniques which is treated in this paper, it is the latter which is the subject of the seminal papers on SM theory. The reader is referred to the early papers of Schweppe [102], Witsenhausen [131], and Bertsekas and Rhodes [15] which treat the bounding of state estimates as a consequence of bounded errors. More recent work on the state estimation problem appears in [1],[54],[63],[66],[79],[98]. A significant number of papers in Russian have also been published. In fact, according to Kurzhanski and Vályi [66], some of the earliest reported work on this subject appears in the Russian paper by Krasovski [61]. For an extensive list of papers in Russian, see [66]. While most of the work on state estimation has strong ties to the identification methods to be discussed in the present paper, the papers by Anan'ev and Kurzhaskii [1] and Morrell and Stirling [79] represent an interesting departure form the bounded error assumption. These papers are concerned with bounded sets of probability distributions for *a priori* and *a posteriori* state estimates. These constraints result in bounded sets of conditional mean estimates and error covariance matrices.

Contemporary research into SM methods has focused to a much greater extent on the second major subdivision concerned with bounding of parameter sets in input–output models. Most of *this* work has treated the BE problem, though at least one broader class of constraints has been studied. Combettes and Trussell [20]–[22] have rigorously investigated feasibility sets which arise as a consequence of "true" probablistic attributes of measurement noise. These sets are constrained parameters which result in residuals which are consistent with the noise properties. Noise properties considered are range, moments, and various second and higher order properties. Combettes and Trussell [23] have also derived feasibility sets for AR model parameters under constraints of stability and bounded (in norm) perturbations on the correlation matrix and vector in the normal equations.

4 The Least Square Error (LSE) Problem

We digress momentarily from the taxonomy of SM methods to interject some material on the conventional LSE problem. This information will be needed in the "lower levels" of the BE discussion to follow, and will play a major role in the developments of the paper.

LSE modelling is a classic and well-understood tool for identification which has an extensive research history quite apart from SM theory. The goals of this brief section are twofold: First, we wish to discuss the LSE approach in relation to the SM approach in preparation for their combination in the main section of the paper. Secondly, necessary notation for the future development will be introduced.

4.1 Relationship Between the LSE and SM Problems

The general LSE problem (for the time interval $t \in [1, n]$) is stated as follows:

Problem 2 ((Weighted) LSE Problem) *Observations $\boldsymbol{y}(t)$, $\boldsymbol{u}(t)$, $t \in [1, n]$, are taken from a system assumed to follow a "true" model of form (2). For a similar model of form (1), find the set of parameter vectors (usually a singleton), say $\Xi(n)$, such that for each $\boldsymbol{\Theta} \in \Xi(n)$, and for any parameters $\boldsymbol{\Gamma}$,*

$$\frac{1}{n}\sum_{t=1}^{n} \lambda_n(t) \parallel \varepsilon(t, \boldsymbol{\Theta}) \parallel^2 \leq \frac{1}{n}\sum_{t=1}^{n} \lambda_n(t) \parallel \varepsilon(t, \boldsymbol{\Gamma}) \parallel^2 \tag{11}$$

where $\lambda_n(\cdot)$ is a sequence of nonnegative weights which may depend on n, and $\|\cdot\|$ denotes the C^k norm.

This problem resembles the form of the general SM problem, Problem 1, posed above. In particular, the result at time n appears to be a "feasible" set, $\Xi(n)$. However, $\Xi(n)$ is not a feasible set, and this is not a proper SM problem. The differences between the LSE problem and a SM problem have been alluded to above and are subtle and revealing.

Feasible sets of solutions in SM problems arise because of some set of attributes we ascribe to the *true* model error at a given time. Any parameter vectors which can produce the given observations *and* an error sequence which has these attributes is *feasible*. Of course, the true parameters must be an element of any feasible set. Conspicuously missing from Problem 2 is any explicit statement of the attributes of the "true" error $\varepsilon_*(\cdot)$ on the range $t \in [1,n]$, which are required in an SM problem statement. In fact, we make no such statement in the LSE problem. Implicitly, we assume that the "true" error sequence is "white noise," implying that *asymptotically*, it will have the smallest possible average squared value in light of the observed data. We do not necessarily believe that the noise is "small" and "locally white" (on the finite time range $t \in [1,n]$) although these are precisely the conditions which underlie (11). The justification for using (11) is that it asymptotically leads to the "true" parameters if our assumption about $\varepsilon_*(\cdot)$ is indeed correct. Along the way, the sets $\Xi(n)$ (usually single points) are not generally montonically decreasing, and do not contain the true parameters $\boldsymbol{\Theta}_*$. They are not, therefore, valid feasible sets.

Another way to view the situation above is as follows. Suppose we were to assume that (11) *is a reflection of some attribute*, $\mathcal{A}_n$, *which we do* believe about $\varepsilon_*(t)$ on $t \in [1,n]$, *viz.*,

$$\boldsymbol{\Theta}_* \text{ is such that } \frac{1}{n}\sum_{t=1}^{n}\lambda_n(t)\,\|\,\varepsilon_*(t)\,\|^2 \text{ is minimal.} \tag{12}$$

In this case $\Xi(n)$ plays the role of a feasible set. Ordinarily, however, $\Xi(n)$ consists of a single point which is therefore both the estimate and the true parameters. In this case since, generally, $\Xi(n+1) \not\subseteq \Xi(n)$, $n = 1,2,\ldots$, *the SM problem is ill-posed and our belief in (12) has led to a sequence of time varying true* parameters – contrary to another belief about the system.

However one views the situation, the conclusion is that the LSE problem is not a valid SM problem. The basic deficiency is the absence of any useful information which serves to constrain the feasible parameters in finite time. In fact, *no* attributes are assigned to the "true" error on a finite time basis, and the resulting "feasible set" is $\Omega(n) = \mathcal{R}^m$ for any $n < \infty$. This results in the necessity of incorporating all data into a LSE estimate, since there is no point-by-point or finite range basis for doing otherwise. Combining the BE considerations below will greatly remedy this inadaquacy of LSE processing.

Because we intend to blend the LSE and BE theory below, and also because LSE processing will emerge as central to another important technique to be discussed, it is important to lay down a formal framework for LSE identification.

4.2 LSE Problem: Formalities

Our discussions of LSE processing will focus exclusively upon models which are LP. The objective here is to lay the formal foundation for these future discussions. Much of the formality described here represent a generalization of developments appearing in the literature.

With reference to Problem 2 and surrounding discussion, we assume the existence of a "true" model of form

$$\boldsymbol{y}(t) = \boldsymbol{\Theta}_*^H \boldsymbol{x}(t) + \boldsymbol{\varepsilon}_*(t) \tag{13}$$

in which $\boldsymbol{x}(t)$ is some m-vector of functions of p_* lags of $\boldsymbol{y}(\cdot)$ and q_* lags plus the present value of $\boldsymbol{u}(\cdot)$, and where, in accordance with the discussion immediately above, $\boldsymbol{\varepsilon}_*(\cdot)$ is the realization of a zero-mean, second moment ergodic, vector-valued random sequence whose components are independent:

$$E\{\boldsymbol{\varepsilon}_*(k-i)\boldsymbol{\varepsilon}_*^H(k-j)\} = \lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n}\boldsymbol{\varepsilon}_*(t-i)\boldsymbol{\varepsilon}_*^H(t-j) = \sigma^2\delta(i-j)\boldsymbol{I} \tag{14}$$

where $E\{\cdot\}$ denotes the expectation, σ^2 is some finite constant, $\delta(\cdot)$ is the Kronecker delta sequence (e.g. [45, p. 37]) and $\boldsymbol{I}$ is the $m \times m$ identity matrix. No finite time attributes are ascribed to $\boldsymbol{\varepsilon}_*(\cdot)$. At time n we wish to use the observed data on $t \in [1, n]$ to deduce an estimated model of the form (5),

$$\boldsymbol{y}(t) = \boldsymbol{\Theta}^H(n)\boldsymbol{x}(t) + \boldsymbol{\varepsilon}(t, \boldsymbol{\Theta}(n)) \tag{15}$$

For the LP problem, the identified parameter vector will be unique for each n (e.g. [45],[52],[69]), but will generally change at every step. Hence, the index n is very significant. In particular, we desire the weighted LSE model for which $\boldsymbol{\Theta}(n)$ satisfies (11).

$\boldsymbol{\Theta}(n)$ can be found as the solution of the following classical linear algebra problem [46]: Given data (or a system of observations) on the interval $t \in [1, n]$ $(n \geq m)$, and some set of error minimization weights, say $\lambda_n(\cdot)$, form the overdetermined system of equations

$$\begin{bmatrix} \sqrt{\lambda_n(1)}\boldsymbol{x}^H(1) & \rightarrow \\ \sqrt{\lambda_n(2)}\boldsymbol{x}^H(2) & \rightarrow \\ \vdots & \\ \sqrt{\lambda_n(n)}\boldsymbol{x}^H(n) & \rightarrow \end{bmatrix} \boldsymbol{\Gamma} = \begin{bmatrix} \sqrt{\lambda_n(1)}\boldsymbol{y}^H(1) & \rightarrow \\ \sqrt{\lambda_n(2)}\boldsymbol{y}^H(2) & \rightarrow \\ \vdots & \\ \sqrt{\lambda_n(n)}\boldsymbol{y}^H(n) & \rightarrow \end{bmatrix} \tag{16}$$

denoted

$$\boldsymbol{X}(n)\boldsymbol{\Gamma} = \boldsymbol{Y}(n) \ , \tag{17}$$

and find the LS estimate, $\boldsymbol{\Theta}(n)$, for the vector $\boldsymbol{\Gamma}$. Because of this interpretation, the pair $(\boldsymbol{y}(t), \boldsymbol{x}(t))$ could appropriately be called an *equation* in many contexts in the following. This term is not always satisfactory, however. Whereas the term "datum" is inappropriate to describe $(\boldsymbol{y}(t), \boldsymbol{x}(t))$, and "data" can be misleading, we will frequently refer to $(\boldsymbol{y}(t), \boldsymbol{x}(t))$ as the *data set* at time t. The expression "per t" should be interpreted to mean "per data set."

There are well known methods to solve this problem. The first is the "batch" solution given by [46]

$$\boldsymbol{\Theta}(n) = \left[\boldsymbol{X}^H(n)\boldsymbol{X}(n)\right]^{-1}\boldsymbol{X}^H(n)\boldsymbol{Y}(n) \tag{18}$$

with the matrix in brackets playing the role of the weighted *covariance* matrix[4], i.e.,

$$\boldsymbol{C}(n) = \boldsymbol{X}^H(n)\boldsymbol{X}(n) = \sum_{t=1}^{n} \lambda_n(t)\boldsymbol{x}(t)\boldsymbol{x}^H(t) \ . \tag{19}$$

For future reference, we also note that the "auxillary matrix" on the right side of (18) can be expressed as

$$\boldsymbol{C}_{\boldsymbol{xy}}(n) \overset{\text{def}}{=} \boldsymbol{X}^H(n)\boldsymbol{Y}(n) = \sum_{t=1}^{n} \lambda_n(t)\boldsymbol{x}(t)\boldsymbol{y}^H(t). \tag{20}$$

When written explicitly in the form

$$\boldsymbol{C}(n)\boldsymbol{\Theta}(n) = \boldsymbol{C}_{\boldsymbol{xy}}(n) \tag{21}$$

this equation is frequently refered to as the set of *normal equations.*

When the weights are time varying by virtue of time-dependent *scaling* of previous weights at time n, i.e.,

$$\lambda_n(t) = \frac{\lambda_{n-1}(t)}{\zeta(n-1)} \quad \forall t \le n-1, \tag{22}$$

where $\zeta(\cdot)$ is a time dependent normalizing sequence, then the weighted LSE solution can be computed recursively using the relations [81]

$$\begin{aligned} \boldsymbol{C}_s(n) &= \boldsymbol{C}(n)/\zeta(n) && (23)\\ C^{-1}(n) &= \boldsymbol{C}_s^{-1}(n-1) - \lambda_n(n)\frac{\boldsymbol{C}_s^{-1}(n-1)\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{C}_s^{-1}(n-1)}{1+\lambda_n(n)G_s(n)} && (24)\\ \boldsymbol{\Theta}(n) &= \boldsymbol{\Theta}(n-1) + \lambda_n(n)\boldsymbol{C}^{-1}(n)\boldsymbol{x}(n)\varepsilon^H(n,\boldsymbol{\Theta}(n-1)) && (25) \end{aligned}$$

where $G_s(n) \overset{\text{def}}{=} \boldsymbol{x}^H(n)\boldsymbol{C}_s^{-1}(n-1)\boldsymbol{x}(n)$. For future reference, let us also define the "unscaled" version of this last quantity, $G(n) \overset{\text{def}}{=} G_s(n)/\zeta(n) = \boldsymbol{x}^H(n)\boldsymbol{C}^{-1}(n-1)\boldsymbol{x}(n)$. *(In general, quantities with subscripts "s" will indicate that the scale factor is included, and those without such subscripts are the unscaled counterparts.)*

As an aside, we note that the scaling sequences $\zeta(\cdot)$ will play a key role in the developments to follow. One peculiarity will occur with regard to this sequence in a very important SM algorithm. In this case $\zeta(\cdot)$ will be such that, for each n, $\zeta(n-1)$ depends on a quantity which will not be computed until time n. In general, we shall distinguish between "causal" and "noncausal" scaling sequences. A *causal* scaling sequence $\zeta(\cdot)$ is one for which, for every n, and for all $n' > n$, $\zeta(n)$ is independent of any quantity which is not computed until time n'. In simple terms, a causal scaling sequence is one which does not depend on "future" processing to determine its "present" values. If $\zeta(\cdot)$ is not causal, then it is *noncausal.* It might seem that a noncausal sequence would be all but impossible to work with, but, as noted, we shall encounter one interesting case to the contrary.

[4] More precisely, this is a *normal* matrix which becomes a "covariance" matrix asymptotically if scaled by $1/n$, and if the mean of the vector $x(t)$ is zero for all t. We shall use the conventional term "covariance" in this work.

When the scaling sequence $\zeta(\cdot)$ is unity for all time, then the (24) and (25) are usually called *recursive least squares (RLS)* (e.g. see [69],[81]) or *sequential least squares (SLS)* (e.g. see [45]), and the word "weighted" is sometimes added to give *WRLS* or *WSLS*. When the scaling factor is taken to be constant, say $\zeta(n) = \alpha^{-1}$ $\forall n$, and such that $0 < \alpha < 1$, then α is called a *fogetting factor (FF)*, and acronym like "SLSFF" might be used. In any case, we will use the acronym "WRLS" to refer to a recursive computation of the weighted LSE solution, and in particular we will call (23) - (25) *MIL-WRLS* to indicate recursions based on the *matrix inversion lemma (MIL)* [45],[69],[81]. This is to be juxtaposed with *QR-WRLS* described in the following paragraph.

When the weights conform to (22), one can use a contemporary WRLS algorithm based on the *QR decomposition* of the $\boldsymbol{X}(n)$ matrix of (17) [28],[33],[43],[48],[71],[72]. The procedure, in principle, involves the application of a sequence of orthogonal operators (Given's rotations) to (17) which leaves the system in the form

$$\begin{bmatrix} \boldsymbol{T}(n) \\ \hline \boldsymbol{0}_{(n-m)\times m} \end{bmatrix} \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{D}_1(n) \\ \hline \boldsymbol{D}_2(n) \end{bmatrix} \tag{26}$$

where the matrix $\boldsymbol{T}(n)$ is an $m \times m$ upper triangular Cholesky factor [46] of $\boldsymbol{C}(n)$ (see (27) below), and $\boldsymbol{0}_{i\times j}$ denotes the $i \times j$ zero matrix. $\boldsymbol{D}_1(n)$ and $\boldsymbol{D}_2(n)$ are $m \times k$ and $(n-m) \times k$ matrices, respectively, which result from the operations on $\boldsymbol{Y}(n)$. It will be useful in our work below to note that

$$\boldsymbol{C}(n) = \boldsymbol{X}^H(n)\boldsymbol{X}(n) = \boldsymbol{T}^H(n)\boldsymbol{T}(n) \tag{27}$$

because $\boldsymbol{T}(n)$ represents an orthogonal transformation on $\boldsymbol{X}(n)$. The system

$$\boldsymbol{T}(n)\boldsymbol{\Theta}(n) = \boldsymbol{D}_1(n) \tag{28}$$

is easily solved using back substitution [46] (k times, once for each column of $\boldsymbol{\Theta}(n)$ and $\boldsymbol{D}_1(n)$) to obtain the LSE estimate, $\boldsymbol{\Theta}(n)$. When the $n+1^{st}$ data set becomes available, it is weighted by $[\lambda_{n+1}(n+1)]^{1/2}$ and the matrices $\boldsymbol{T}(n)$ and $\boldsymbol{D}_1(n)$ are scaled by $[\zeta(n)]^{-1/2}$ before incorporating this new information. This procedure can be performed in a recursive manner using only about $m^2 + km$ memory locations. Details for the SISO case (which are easily generalized) are found in [28],[33],[48],[71]. We shall use the name *QR-WRLS* to refer to this form of the recursion. This formulation makes possible the solution of the ellipsoid algorithms to be described on contemporary parallel architectures (discussed in Section 6.7) for great speed advantages.

5 The Bounded Error (BE) Identification Problem

5.1 Overview

We now return to Fig. 1 and the survey of SM methods and discuss the most widely researched group of techniques, those based on a BE constraint. The general problem statement is as follows:

Problem 3 (BE Identification Problem) *Observations $\boldsymbol{y}(t)$, $\boldsymbol{u}(t)$, $t \in [1, n]$ are "known" to have been generated by a "true" model of form (2) whose error sequence is*[5] *"pointwise energy bounded"*

$$\mathcal{A}_n : \quad \| \boldsymbol{\varepsilon}_*(n) \|^2 < \gamma(n), \tag{29}$$

where $\gamma(\cdot)$ is a known positive sequence[6]*. We propose a model of form (1) with $\boldsymbol{\Psi} \equiv \boldsymbol{\Psi}_*$, $p = p_*, q = q_*, r = r_*$, whose parameters, $\boldsymbol{\Theta}_*$, are unknown, but whose error sequence adheres to $\mathcal{A}_n$ at time n. Find the feasible set of parameters, $\Omega(n)$, such that for each $\boldsymbol{\Theta} \in \Omega(n)$, the proposed model is consistent with the observations.*

BE methods are categorized into those which the models are LP and those which are non-LP (see Section 2.2). The feasible solution sets that arise as a consequence of error bounding assume different geometries in the parameter space, depending on the form of the model. In general, constraints of form (29), in conjunction with a model of form (1) and the measured data, imply *pointwise* feasible sets

$$\omega(n) = \left\{ \boldsymbol{\Theta} \mid \; \| \boldsymbol{y}(n) - \boldsymbol{\Psi}[n, \boldsymbol{\Theta}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\varepsilon}, p, q, r] \|^2 < \gamma(n) \right\}. \tag{30}$$

These can be intersected over time to create a feasible set over the range $t \in [1, n]$,

$$\Omega(n) = \bigcap_{t=1}^{n} \omega(t). \tag{31}$$

For a *non-LP* model the "local" $\omega(n)$ are generally hypersurfaces in $\boldsymbol{C}^{m \times k}$ which, when intersected over time, create sets which may have highly irregular geometries and which need not be connected in the parameter space (see e.g. Fig. 2 and [121]). The work that has been done on such problems has been largely concerned with developing novel algorithms for MISO, real parameter, systems,.which *bound* $\Omega(n)$. Specific approaches can be found in [6], [8],[10],[18],[19],[34]-[36],[56],[57],[74],[84],[90],[93],[97],[104],[105], [113], [123], [124], [128],[129]. Since the focus of this paper is upon a special class of LP methods and signal

[5] This is slightly less general than stating asymmetrical amplitude bounds, $\gamma_{\min}(n) < \| \varepsilon_*(n) \| < \gamma_{\max}(n)$, but the very slight loss of generality is worth the significant analytic gain afforded by this assumption.

[6] We shall assume this sequence known throughout this paper. While determination of appropriate error bounds often follows naturally from the physical constraints of the problem, in other cases this determination is challenging. One theoretical approach is found in [120], while an experimental discussion for a particular application is found in [29].

processing applications, we shall not further pursue the topic of non-LP models[7]. An excellent place to begin a review of non-LP methods is with the recent paper by Walter and Piet-Lahanier [121].

In the LP model case, error bounding implies pointwise "hyperstrip" regions of possible parameter sets in the space,

$$\omega(n) = \left\{ \boldsymbol{\Theta} \mid \| \boldsymbol{y}(n) - \boldsymbol{\Theta}^H \boldsymbol{x}(n) \|^2 < \gamma(n) \right\}, \tag{32}$$

which, when intersected over a given time range (see (31)), usually form convex polytopes of feasible parameters (see Fig. 3). Three different approaches have been introduced which describe or characterize the feasible parameter sets. The first approach (developed for real, generally MISO, systems) produces exact parameterized descriptions of these polytopes [7],[16],[17],[73],[77],[78],[92],[122],[125]–[127]. Although this approach is recursive and simple, its computational complexity increases with the number of vertices of the polytope. The second approach (also for real MISO systems) gives orthotopic outer bounds of the solution sets [5],[73],[77]. This approach yields exact parameter uncertainty intervals at the expense of very complex computations. The third approach is of much lower complexity compared to the first two and works with an outer bounding[8] hyperellipsoid, a superset of the polytope [24],[26],[27],[29]–[32],[37],[38],[41],[49],[51],[84]–[89],[99]–[101].

Ellipsoid algorithms are often presented as BE procedures, and indeed they do follow from the BE constraints. However, they are more fruitfully viewed as a marriage between the LSE and BE problems for LP models. With this point of view, signal processing engineers have begun to exploit the benefits of BE information in the context of LSE identification problems. To stress this point of view, we feature the ellipsoid algorithms in their own section to follow. This subject will be covered in considerable detail and will comprise the remainder of the paper.

6 Combining the LSE and BE Problems: Ellipsoid Algorithms

6.1 A Unified Optimal Bounding Ellipsoid (UOBE) Algorithm

Please note that the rigorous development of several of the key results to follow are found in the appendices.

The benefits of combining BE considerations, when they are known, with LSE identification have been alluded to in Section 4.2. LSE identifiers exploit no point-by-point information which can be used to ascertain the usefulness of observations. This fact manifests itself in the effective retention of the entire parameter space as a "feasible set," and results in wasteful processing. The idea to combine BE considerations with LSE identification did not arise out of a quest to make LSE processing more efficient,

[7] With one exception. Methods developed for ARX (LP) models have been extended for use with ARMA and ARMAX (non-LP) models [84],[99],[101]. We will discuss these techniques below.

[8] *Inner* bounding algorithms of the last two approaches have also been presented in [83],[116].

however. Rather, it resulted from the discovery that ellipsoid bounding algorithms are very closely related to WRLS. While, clearly the feasible set arising from *any* SM algorithm will contain the LSE estimate, it is the ellipsoid algorithms which have a particularly attractive relationship.

We begin by seeking a solution to the SM (BE) problem. Since we are working with an LP model, the BE constraint (see Problem 3) is given by

$$\| \boldsymbol{y}(n) - \boldsymbol{\Theta}_*^H \boldsymbol{x}(n) \|^2 < \gamma(n). \tag{33}$$

It follows readily that (see Lemma 1 in Appendix A.2)

$$\sum_{t=1}^{n} \lambda_n(t) \| \boldsymbol{y}(t) - \boldsymbol{\Theta}_*^H \boldsymbol{x}(t) \|^2 < \sum_{t=1}^{n} \lambda_n(t)\gamma(t) \tag{34}$$

for any positive numbers $\lambda_n(t)$, $t \in [1, n]$. For any nonnegative sequence $\lambda_n(\cdot)$, (34) specifies a set to which $\boldsymbol{\Theta}_*$ must belong. Let us denote this set,

$$\bar{\Omega}(n) \stackrel{\text{def}}{=} \left\{ \boldsymbol{\Theta} \mid \sum_{t=1}^{n} \lambda_n(t) \| \boldsymbol{y}(t) - \boldsymbol{\Theta}^H \boldsymbol{x}(t) \|^2 < \sum_{t=1}^{n} \lambda_n(t)\gamma(t) \right\}. \tag{35}$$

Note that all elements of $\bar{\Omega}(n)$ need *not* be in the usual feasible set, $\Omega(n)$, consisting of the intersection of pointwise "hyperstrips" (see (31) and (32)). In fact, $\bar{\Omega}(n)$ can be almost any size depending on the choice of numbers $\lambda_n(t), t \in [1, n]$. (Note that $\lambda_n(\cdot)$ is indexed (subscript) by the end-time n because we might wish to have a completely different sequence of parameters at each n to control the size, placement, etc. of the hyperellipse.) Whatever sequence $\lambda_n(\cdot)$ is chosen, however, the set $\bar{\Omega}(n)$ must *contain* the feasible set, and therefore $\boldsymbol{\Theta}_*$:

$$\boldsymbol{\Theta}_* \in \Omega(n) \subseteq \bar{\Omega}(n). \tag{36}$$

The following development is rigorously supported by Proposition 1 in Appendix A.1. Some manipulation of (35) shows that the set $\bar{\Omega}(n)$ may be expressed as follows:

$$\bar{\Omega}(n) = \left\{ \text{tr}\ \{\boldsymbol{\Theta} \mid [\boldsymbol{\Theta} - \boldsymbol{\Theta}_c(n)]^H \boldsymbol{\Phi}(n) [\boldsymbol{\Theta} - \boldsymbol{\Theta}_c(n)]\} < 1 \right\} \tag{37}$$

where tr$\{\cdot\}$ denotes the trace of a matrix. This set as a hyperellipsoid in $\boldsymbol{\mathcal{R}}^{2m \times k}$, with its "center" at $\boldsymbol{\Theta}_c(n)$. We give meaning to the term "hyperellipsoid" below. *The fundamental connection of this ellipsoidal set to to the weighted LSE problem is as follows:* The center of the ellipsoid is exactly the weighted LSE estimate using weights $\lambda_n(\cdot)$,

$$\boldsymbol{\Theta}_c(n) = \boldsymbol{\Theta}(n), \tag{38}$$

and the ellipsoid matrix $\boldsymbol{\Phi}(n)$ is a scaled version of the associated covariance matrix

$$\boldsymbol{\Phi}(n) = \frac{\boldsymbol{C}(n)}{\kappa(n)} \tag{39}$$

(see (18) and (19)), and where $\kappa(n)$ is the scalar quantity,

$$\kappa(n) \overset{\text{def}}{=} \operatorname{tr}\{\boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta}(n)\} + \sum_{t=1}^{n} \gamma(t)\lambda_n(t)\left[1 - \gamma^{-1}(t) \parallel \boldsymbol{y}(t) \parallel^2\right]. \tag{40}$$

To give meaning to the term "hyperellisoid," consider a single column, say $\boldsymbol{\theta}_i(n)$, of $\boldsymbol{\Theta}(n)$, corresponding to output $y_i(\cdot)$ in vector $\boldsymbol{y}(\cdot)$. Using (37), we see that $\boldsymbol{\theta}_i(n)$ *is* constrained to be an element of a set which is properly called a hyperellipsoid, say,

$$\tilde{\Omega}_i(n) = \left\{\boldsymbol{\theta}_i \mid [\boldsymbol{\theta}_i - \boldsymbol{\theta}_i(n)]^H \frac{\boldsymbol{C}(n)}{\kappa(n)}[\boldsymbol{\theta}_i - \boldsymbol{\theta}_i(n)] < 1\right\}. \tag{41}$$

In particular, when $\boldsymbol{\theta}_i(n)$ is real and of dimension two, the perimeter of $\tilde{\Omega}_i(n)$ is precisely what is conventionally regarded as an ellipse in $\mathcal{R}^2$ (see Fig. 3). Notice that all outputs $y_i(\cdot)$, $i = 1, 2, \ldots, k$, will apparently share the same "ellipsoid matrix," $\boldsymbol{C}(n)/\kappa(n)$, but their corresponding ellipsoids will be centered on different estimates. This fact will be important in the optimization problem to be discussed below. Also note that the influence of the "other" outputs in $\boldsymbol{y}(\cdot)$ on the ellipsoid $\tilde{\Omega}_i(n)$ arises through the parameter $\kappa(n)$. This means that the MIMO solution, as we shall describe it here, is not equivalent to a decomposition of the problem into k MISO systems. However, the MIMO problem *does* correctly include the MISO problem as a special case.

We conclude therefore that under known BE constraints, a hyperellipsoid can be associated with a weighted LSE estimation problem and *conversely*. *This set is illustrated in Fig.* 3 *for the two-dimensional* case. Clearly, the weights $\lambda_n(\cdot)$ parameterize the ellipsoid and presumably can serve to minimize its size and orientation in the parameter space. Anticipating that we will want to work with recursive least squares estimation, let us henceforth restrict our attention to weight sequences which conform to the scaling pattern[9] (22). *This effectively restricts to* one *(viz.* $\lambda_n(n)$*) the number of free parameters available to control the bounding ellipsoid at time* n. The central objective of a *bounding ellipsoid algorithm* is to employ the weights in the context of LSE estimation to sequentially optimize some feature of the ellipsoid (directly or indirectly related to its "size"). A significant benefit is that often no weight exists which can minimize the ellipsoid, indicating that the incoming data set is uninformative in the SM sense.

While it may not be immediately apparent from the original developments in the literature, all published bounding ellipsoid algorithms, both adaptive and nonadaptive, adhere to the following steps. Let us refer to this set of operations as the *Unified Optimal Bounding Ellipsoid (UOBE)* algorithm (for a complex MIMO LP system): At time n,

1. In conjunction with the incoming data set $(\boldsymbol{y}(n), \boldsymbol{x}(n))$, find the weight, say $\lambda_n^*(n)$, which will prospectively optimize some quantitative feature of $\Omega(n)$ related to its "size." (This will require knowledge of $\boldsymbol{C}(n-1)$, $\kappa(n-1)$, and $\zeta(n-1)$.)

[9] An exception to this rule is that, for adaptive strategies to be discussed below, we will additionally allow $\lambda_n(t)$ to be set to zero for one or more $t \leq n-1$.

2. Discard the data set $(\boldsymbol{y}(n), \boldsymbol{x}(n))$ if $\lambda_n^*(n) \leq 0$.

3. Update $\boldsymbol{C}(n)$ and $\boldsymbol{\Theta}(n)$ using MIL-WRLS or QR-WRLS (see Section 4.2).

4. Update $\kappa(n)$ according to (40) or one of the recursions given in Lemma 2 in Appendix A.2.

With one exception (see Dasgupta–Huang OBE algorithm below), all published OBE algorithms operate on the principle of minimizing a set measure on $\tilde{\Omega}(n)$ by choice of $\lambda_n^*(n)$. For a SISO system, Fogel and Huang suggest two set measures for the optimization. The first is the determinant of the matrix $\boldsymbol{\Phi}^{-1}(n)$,

$$\mu_v\{\tilde{\Omega}(n)\} \overset{\text{def}}{=} \det\{\boldsymbol{\Phi}^{-1}(n)\} \tag{42}$$

and the second is the trace,

$$\mu_t\{\tilde{\Omega}(n)\} \overset{\text{def}}{=} \text{ tr } \{\boldsymbol{\Phi}^{-1}(n)\}. \tag{43}$$

(Having established these quantities as set measures on $\tilde{\Omega}(n)$, for simplicity, we shall henceforth write $\mu_v(n)$ and $\mu_t(n)$. We shall also occassionally write $\mu(n)$ to mean "either $\mu_v(n)$ or $\mu_t(n)$.") In the MISO case in which $\tilde{\Omega}(n)$ is clearly intepretable as an ellisoid (see (41)), $\mu_v(n)$ is proportional to the square of the volume of the ellipsoid, while $\mu_t(n)$ is proportional to the sum of the square root of its semi-axes. A moment's reflection will indicate that the same two measures are meaningful in the MIMO case, since they result in the minimization of the volume or trace of the *common* ellipsoid shared by all the outputs (see discussion below (41)).

It is shown in Proposition 2 in Appendix A.1 that, when the scaling sequence is causal (see discussion below (25)), then $\lambda_n^*(n)$ is the unique positive root of the polynomials $F_v(\lambda)$ and $F_t(\lambda)$ for the volume and trace measures respectively, where F_v is a quadratic,

$$F_v(\lambda) = a_2\lambda^2 + a_1\lambda + a_0 \ , \tag{44}$$

and F_t is a cubic polynomial

$$F_t(\lambda) = b_3\lambda^3 + b_2\lambda^2 + b_1\lambda + b_0 \ . \tag{45}$$

The coefficients a_i and b_i are given in terms of quantities which are known prior to time n.

Interestingly, we will find that the optimization procedure is not "locally" affected by a causal scaling process. This is so because neither measure μ_v nor μ_t is changed when the scale factor is included. To show precisely what we mean by this, consider the optimization problem at time n. All previous weights will be modified by scale factor $\zeta(n-1)$. We have called the resulting covariance matrix $\boldsymbol{C}_s(n-1) \overset{\text{def}}{=} \boldsymbol{C}(n-1)/\zeta(n-1)$. The definition of $\kappa(n-1)$ in (40) will indicate that the effect of weight scaling on this quantity is likewise a simple scaling,

$$\kappa_s(n-1) \overset{\text{def}}{=} \frac{\kappa(n-1)}{\zeta(n-1)}. \tag{46}$$

Accordingly, the ellipsoid matrix $C(n-1)/\kappa(n-1)$ is changed to $C_s(n-1)/\kappa_s(n-1)$, by the scaling procedure. Note, however, that the scale factors cancel in this ratio, so that either of the measures of size will remain unchanged. It is very important *not* to infer that the optimization process is independent of the scaling factors. Clearly the existing covariance matrix and κ value at each time is influenced by the complete history of the scale factors. The consequence of the analysis above is simply that the ellipsoid volume at a specific time is not affected by scaling. This will have implications for theoretical developments surrounding the optimal weight (see Proposition 2) in Appendix A.1.

Finally, we note an important fact to which we will return in Section 6.6. For the volume algorithm using weights of form (22), it can be shown that, if an optimal weight exists, it will definitely shrink the volume of the ellipsoid. A similar result can be obtained for the trace measure [80]. This has important implications for convergence of the ellipsoid, the analysis of which has been widely misunderstood.

A detailed version of the UOBE algorithm for a MISO system, which is based on QR-WRLS and the volume criterion, appears in Fig. 4. It should be clear how to incorporate changes necessary to implement a "trace" algorithm, or to introduce additional outputs. This general algorithm will embrace any of the specific algorithms discussed below. We now discuss variations on this general algorithm.

6.2 The Fogel–Huang OBE Algorithm

6.2.1 History and Development of F-H OBE.

The first major journal paper on the application of ellipsoid algorithms to parametric LP models was published by Fogel and Huang in 1982 [41]. The Fogel–Huang algorithm is frequently called the *optimal bounding ellipsoid (OBE)* algorithm, and we shall adopt the name "F-H OBE" in this paper to distinguish it from another algorithm to be presented below. F-H OBE was originally presented for the SISO ARX model, but it is easily generalized using the developments described in this paper. This method follows the basic framework of the UOBE algorithm enumerated above, with the following specific conditions:

1. $\zeta(n)=\kappa(n)$ for each n;

2. MIL-WRLS is used to implement the recursions (but QR-WRLS can be used as well);

3. Set measures (42) and (43) are employed.

It is interesting to note that, because the scaling sequence $\zeta(\cdot)$ is equivalent to the sequence $\kappa(\cdot)$ in F-H OBE, the ellipsoid matrix at time n, $\boldsymbol{\Phi}(n)$, is identical to the scaled covariance matrix $C_s(n)=C(n)/\zeta(n)$ whose inverse is computed directly in the course of the MIL-WRLS equations. This is a consequence of the geometric approach taken (see below) rather than a deliberate choice of the scaling sequence. We also note that there is nothing to preclude the use of QR-WRLS in conjunction with F-H OBE. Alternative versions of OBE have been published and will be described below, but first it is interesting to place the F-H OBE development in historical perspective.

In [41], using the BE constraints only, Fogel and Huang arrive at the ellipsoid of form (37) and recognize that the center of the ellipsoid is a weighted LSE estimate. However, the LSE problem is not pursued directly. Instead, the fact that ellipsoids can be used to bound the feasible set is used as a motivation for the following geometric approach: Assume that a membership set $\tilde{\Omega}(n-1)$ is known at time $n-1$. We need not be aware of parameters $\lambda_{n-1}(t)$, $t \in [1, n-1]$, nor even that there is a LSE problem underlying the membership set. The objective is to find a new (small, if possible) ellipsoid which superscribes the intersection of $\tilde{\Omega}(n-1)$ with the incoming feasible "hyperstrip" $\omega(n)$ (see (32) and Fig. 5). The work of Kahan [53] had shown that a family of such circumscribing ellipsoids could be computed using relations which the authors then manipulate into the equations which comprise F-H OBE. The quantity $q(n)$ (equivalent to $\lambda_n(n)$) emerges as a single parameter with which to control the size of the ellipsoid $\tilde{\Omega}(n)$. We will henceforth refer to $q(n)$ as $\lambda_n(n)$, even though this does not connote the geometric spirit of the original F-H OBE development.

F-H OBE is sometimes called the *minimum volume sequential (MVS)* algorithm when it is is based upon sequential minimization of $\mu_v(n)$. This involves the construction, and solution for the roots of, the quadratic equation (44) to find the optimal parameter $\lambda_n^*(n)$. Similarly, the *minimum trace sequential (MTS)* algorithm is based upon minimization of $\mu_t(n)$ by optimizing $\lambda_n(n)$. This procedure requires the construction and solution for the positive root of the cubic equation (45).

The F-H OBE algorithm was the first UOBE-type algorithm to be presented as having potential benefits for signal processing [49]. These benefits derive from the optimization procedure, as alluded to above. Generally speaking (precise comments are found in Section 6.6 below), as n increases, the true feasible set $\Omega(n)$, and the ellipsoid $\tilde{\Omega}(n)$, decrease in size and it becomes increasingly likely that[10]

$$\tilde{\Omega}(n) = \omega(n) \cap \tilde{\Omega}(n-1) = \tilde{\Omega}(n-1). \tag{48}$$

This means that the new data set is not providing any useful information in the sense of shrinking the membership set. There is no positive parameter $\lambda_n(n)$ with which to combine the data set at time n with the current ellipsoid to create a smaller ellipsoid. The manifestation, therefore, is that the "optimal" parameter in the sense of minimizing $\mu(n)$, $\lambda_n^*(n)$, is nonpositive. In this case the data set at time n should be rejected and the computational effort of processing it avoided. In many simulations and experiments with real data (e.g. [24],[29],[32],[49]), typically 70 - 95% of the data are "rejected" in this sense.

[10] It also becomes increasingly likely that

$$\Omega(n) = \omega(n) \cap \Omega(n-1) = \Omega(n-1), \tag{47}$$

but this does not necesarily mean that the incoming data set cannot be used to minimize the *ellipsoid*.

A critical point about this data selection process must be made which was not necessarily evident in the early papers. One must be very careful not to infer that the complexity of the F-H OBE (or any UOBE) algorithm is drastically reduced (say, by 90%) by virtue of this data selection procedure. In fact, in its basic form, if the parameter matrix $\boldsymbol{\Theta}_*$ is of dimension $m \times k$, then each of the optimality checks requires $\mathcal{O}(m^2)$ complex floating point operations (cflops), then, when accepted another $\mathcal{O}[(2 + .5k)m^2]$ cflops are required to update the covariance matrix and parameter estimates[11]. To process the data set directly using WRLS requires $\mathcal{O}(3m^2)$ cflops. While a dramatic decrease in the *number* of data used results, the computational load is not significantly decreased, especially for large m. There are methods to remedy this problem which will be discussed below.

As an aside, we note that the MVS version of the F-H OBE algorithm is suboptimal in the following sense. When one of the hyperplanes bounding $\omega(n)$ does not intersect $\tilde{\Omega}(n-1)$, a smaller (volume) $\tilde{\Omega}(n)$ can be achieved by repositioning the nonintersecting plane to be tangent to $\tilde{\Omega}(n-1)$. Belforte and Bona have suggested this procedure in [5] (see also [9]). As pointed out by Walter and Piet-Lahanier [121], the modified procedure is equivalent to the *ellipsoid with parallel cuts (EPC)* algorithm developed by researchers working in linear programming [59],[103].

6.2.2 Dasgupta-Huang OBE

A significant variation on the F-H OBE algorithm has been suggested by Dasgupta and Huang [24] since the publication of the original algorithm. Again, the method is originally developed for the SISO ARX model, but can be generalized using the devlopments in this paper. The Dasgupta-Huang OBE (D-H OBE) algorithm has two unusual features with respect to all other UOBE algorithms. These are the use of *noncausal* scale factors, and an optimization procedure which does *not* seek to directly minimize a set measure on $\tilde{\Omega}(n)$. D-H OBE employs the scale factors

$$\zeta(n-1) = (1 - \lambda_n^*(n))^{-1}. \tag{49}$$

With reference to (19) and (22), it is seen that, for a given optimal weight $\lambda_n^*(n)$, the updated covariance matrix is a convex combination of $\boldsymbol{C}(n-1)$ and the new data outer product.

$$\boldsymbol{C}(n) = (1 - \lambda_n^*(n))\boldsymbol{C}(n-1) + \lambda_n^*(n)\boldsymbol{x}(n)\boldsymbol{x}^H(n) \tag{50}$$

Accordingly, this choice of scale factors constraints the optimal weights to the range $[0, \alpha]$ for $\alpha < 1$. The central benefit of this method is that it provides the means with which to prove asymptotic and exponential convergence of the ellipsoid, and cessation of updating, using Lyaponov theory. Upon convergence, the residuals, $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}(\cdot))$ are guaranteed to remain in the "dead zone" indicated by the error bounds, i.e.,

[11] Throughout, a cflop is taken to be one complex multiplication and one complex addition.

$\lim_{\substack{t \to \infty \\ n \to \infty}} \| \varepsilon(t, \Theta(n)) \|^2 < \gamma(t)$. The number $(1 - \lambda_n^*(n))$ is referred to as a "forgetting factor" by Dasgupta and Huang, and, although it does serve to downweight the past contributions of to the covariance matrix, it is not a forgetting factor in the conventional sense, since it is not a free parameter and therefore does not *explicitly* control adaptation. The algorithm does exhibit some adaptation capabilities as do other UOBE algorithms due to the optimal data weighting. Explicitly adaptive UOBE algorithms will be discussed below.

A second significant difference in the D-H OBE algorithm occurs in the technique employed for determining "optimal" weights $\lambda_n^*(n)$. Rather than minimize a set measure such as (44) or (45), the weight is chosen to minimize $\kappa(n)$, subject to the constraint that it be in the allowable range $[0, \alpha]$. The reason for this choice is that $\kappa(n)$ is a bound on the Lyapunov function used in the minimization. A side benefit is that the check for usefulness of the data set is very cost effective. We will return to a discussion of this "unconventional" optimization technique, as well as issues of computational efficiency, in two places below.

It is notable that, in spite of the "noncausal" scaling factor at time $n-1$, $\zeta(n-1)$, which might be expected to create intractable nonlinearities with respect to $\lambda_n^*(n)$, it is still possible to derive polynomials like (44) and (45) with which to optimize set measures of $\tilde{\Omega}(n)$ [68]. Doing so, however, defeats one of the main purposes of using the complicated scale factors, and whether such an optimization has any usefulness remains an open question. We return to this issue in Section 6.5.2.

6.3 The SM-WRLS Algorithm

6.3.1 History and Development of SM-WRLS.

While developed geometrically, we know that the F-H OBE algorithm solves a LSE problem with time varying weights. From this point of view, it is interesting to note that the algorithm is charged with focusing on the hyperstrip $\omega(n)$ associated with the "new" data set. Intuitively, the scaling down of previous weights is consistent with this concentration on the new data set. However, it is prudent to wonder whether a tighter, or at least "simpler" membership set could be found. The SM-WRLS algorithm, to which we now turn, addresses both the concern for a more conventional algorithm and the more "uniform" attention to the true feasible set.

Even though Fogel and Huang clearly state in their 1982 paper that there is an LSE problem underlying F-H OBE, the geometric approach taken tends to obscure its presence. The approach, notwithstanding, however, the similarity of the F-H OBE (as well as the D-H OBE) equations to WRLS is striking, and it has not gone unnoticed in the literature. In their recent paper, Walter and Piet-Lahanier make the following remarks [121]: "Let us stress, however, that the EPC and MVS algorithms are not just another variation of RLS. As Schweppe puts it [102], a comparison of set theoretic concepts with stochastic theory reveals that

1. the detailed mathematical manipulations are very different.

2. the final equations look similar.

3. the final equations behave quite differently in general.

"Moreover, the type of information needed is completely different. The RLS algorithm only requires measurements, whereas the EPC and MVS algorithms also require bounds on the errors." As we know from above, however, the difference between WRLS and F-H OBE (or any UOBE algorithm) is not as different as one may infer from these comments. As Norton and Mo have recently written concerning F-H OBE [86], "The algorithm [F-H OBE] differs from recursive least squares by an extra data-dependent scaling of [the ellipsoid matrix $\boldsymbol{\Phi}(n)$]." In a 1989 paper. Deller [26] similarly recognized that "[F-H] OBE is 'WRLS with time varying weights'." It is this recognition, combined with Norton and Mo's formulation for *adaptive* ellipsoid processing, that led to the UOBE formulation taken in this paper. (We will see that UOBE also embraces adaptive strategies below.)

Until recently, however, this uniformity of ellipsoid algorithms was not fully appreciated. In the early and mid 1980's, Deller and students [30],[37],[38],[70], recognized the similarity of F-H OBE to RLS, attempted to associate an ellipsoid directly with WRLS rather than conversely. The result is an OBE-*like algorithm which is exactly* interpretable as conventional WRLS (i.e., only equations (24) and (25) with $\boldsymbol{C}_s(n) \equiv \boldsymbol{C}(n)$ or $\zeta(n) = 1, \quad \forall n$), with the sequence of optimal ellipsoid parameters $\lambda_n^*(\cdot)$ simply interpretable as the weights used in the process. Fogel and Huang's volume measure $\mu_v(n)$ has been used as the optimization criterion, but the trace measure can be employed as well. In later work, the use of QR-WRLS was suggested to enhance the method in a number of ways to be described [26].

The algorithm proposed by Deller and others has been called *set membership weighted recursive least squares (SM-WRLS)* to emphasize the nature of their approach. SM-WRLS is, in fact, a UOBE algorithm with the following conditions:

1. $\zeta(n) = 1$ for each n;

2. QR-WRLS is used to implement the recursions (but MIL-WRLS can also be used);

3. Volume measure $\mu_v(n)$ is used as the optimization criterion (but $\mu_t(n)$ can be used as well).

6.3.2 Illustration

At appropriate points in the paper, we will illustrate the behavior and performance of the UOBE approach through simulation studies. A common set of two systems will be used which will be introduced here. For simplicity, the SM-WRLS algorithm is used as the nominal algorithm. The volume measure is employed as the optimization criterion. Many other example studies are found in the literature (e.g.

[9],[24],[26],[27],[41],[49],[86],[89],[101]) including some with real data (e.g. [29],[77]). In particular, many studies with time-*invariant* systems have been published, so we advance immediately to the case of time varying parameters.

We consider the estimation of the parameters of a real signal, real parameter, time varying *AR(2)* model of the form[12]

$$y(t) = a_{1*}(t)y(t-1) + a_{2*}(t)y(t-2) + \varepsilon_*(t) \tag{51}$$

with $\Theta_*(t) = [a_{1*}(t)\ a_{2*}(t)]^H$. Two similar systems will be used. The first is a more severe test of the tracking ability of an identification algorithm. In this case the pole pair of the system alternates abruptly between $0.8 \pm j0.2$ and $-0.8 \pm j0.2$ every 1000 samples, so that a_{1*} alternates between ±1.6, and a_{2*} remains constant at 0.68. In the second system, the poles alternate between the same sets of conjugate pairs, but the transisitions are gradual rather than abrupt. In this case a_{1*} changes linearly (between $+1.6$ and -1.6 and *vica versa*) over 1000 point ranges, then remains fixed for 1000 point intervals. We shall refer to the two systems as the "fast" and "slow" systems, respectively, though we hasten to point out the the "slowly" time varying system does *not* represent a trivial tracking problem. Since only a_{1*} changes in each case, it is the more interesting parameter to observe. To conserve space, we show only the results for a_{1*} in each simulation. We found nothing particularly unusual or unexpected in the results for a_{2*}.

A 7,000 point sequence, $y(\cdot)$, was generated by driving the parameter sets with an uncorrelated sequence, $\varepsilon_*(\cdot)$, which was uniformly distributed on $[-0.5, 0.5]$. $\varepsilon_*(\cdot)$ was generated using a random number generator based on a subtractive method [94].

The inherent ability of UOBE algorithms (without any special adaptive provisions) to adapt and track time varying parameters is sometimes quite dramatic. In this work, we have intentionally chosen systems for which SM-WRLS exhibits less than excellent tracking performance in order to illustrate several important points. For reference, in Figs. 6 (a) and (b), we show the results of using standard RLS in the identification (no data selection and optimization, $\lambda_n(n) = \zeta(n) = 1$ for all n). The algorithm is clearly incapable of following the parameters in either the fast or slow case. The RLS results can be contrasted with those using SM-WRLS in Fig. 7. Though not excellent, the SM-WRLS results are improved with respect to RLS (at least initially), and it is important to note that this improved performance comes with somewhat improved computational efficiency (more on this below). In this case SM-WRLS uses only the fractions $\rho = 0.020$ (fast system) and $\rho = 0.025$ (slow) of the data and yet yields better parameters estimates in the early stages of identification. However, two important points are to be emphasized here. First, SM-WRLS does not *reliably and predictably* adapt to time varying systems. Even for more slowly time varying systems,

[12] Note that for the first time in this paper, we have allowed the dynamics of the "true" system to be time varying. The theoretical developments above do not strictly support the identification of such systems, so the issue of adaptation is an important one to which will shall pay close attention in the following.

SM-WRLS (and UOBE algorithms in general) cannot be used in adaptive schemes with confidence. This motivates the need for specifically adaptive techniques. Further, a deeper analysis of this situation reveals a very important second point. The quantities $\mu_v(n)$ and the sign of $\kappa(n)$ ($\text{sgn}\{\kappa(n)\}$) are shown in Figs. 8(a) and 8(b), respectively. At time $n = 1000$ we see a very disturbing development. The volume begins to increase, and the parameter κ becomes negative. Both of these trends are in violation of theory, but they arise precisely because the theoretical development does not strictly support the identification of time varying systems. The most revealing anomaly is the appearance of a negative κ which indicates an "ellipsoid of negative dimensions." Theoretically speaking, the algorithm has become completely disintegrated at time $n = 1000$, and its performance is therefore not predictable based on SM principles. Nevertheless, we see that the parameters continue to be tracked rather well for at least another cycle. Apparently, there is significant benefit to using the "optimization" process even if the success is not analyzable. In fact, we have seen this phenomenon in many other simulations. It is more likely to occur with rapid changes in dynamics (as we discuss below), but may occur in slower systems as well. The conclusion is that, not only is the apparent adaptive capability of UOBE algorithms unpredictable, but even when good tracking *does* occur, the good performance is not necessarily attributable to the proper principles of the underlying the methods. In turn, this latter observation adds to the uncertainty in predicting adaptive performance. We shall return to these points in future discussions.

6.4 Adaptive UOBE Algorithms

6.4.1 Introduction

While UOBE algorithms have been observed to have inherent and fortuitous adaptive capabilities as a result of their optimal weighting strategies, we have just seen that these capabilites are unpredictable at best. Accordingly, measures have been suggested by Norton and Mo [86], and Deller and Odeh [26],[27],[31],[32],[87]-[89] to render explicit and controlable adaptation[13]. Of three general methods suggested by Norton and Mo, the *bound incrementing* method does not closely follow the UOBE paradigm established above, so we refer the reader to the original paper for details. The other two Norton methods are discussed below. All adaptive strategies for ellipsoid algorithms work on the general principle of iteratively inflating the "current" ellipsoid in some sense before considering an incoming data set. The basis for this inflation is to contain the shifting true parameters while at the same time increasing some measure of "size" of the ellipsoid (see (42) and (43) below), making it more likely that the incoming data, with potentially novel information, will be selected. Deller and Odeh have suggested the use of QR-WRLS in the adaptive methods because of the convenient computational interpretation of the procedure. In principle, however,

[13] Additionally, Norton and Mo briefly discuss adaptive strategies for other than ellipsoid algorithms.

MIL-WRLS can be used as well.

6.4.2 Exponential Forgetting

A general UOBE algorithm can be designed to be explicitly adaptive within the established framework by judicious choice of scaling sequence $\zeta(\cdot)$. One seemingly reasonable choice is to let the scaling effect a conventional forgetting factor,

$$\zeta^{-1}(n) = \alpha, \quad 0 < \alpha < 1 \tag{52}$$

for all n. The computational mechanisms for including such a forgetting factor into both forms of WRLS is found in Section 4.2. Additionally, if another scaling sequence $\zeta(\cdot)$ is part of the algorithm for some purpose other than adaptation (e.g. in F-H OBE), then the sequence $\zeta(\cdot)/\alpha$ can used for scaling in order to achieve forgetting. Deller and Odeh have called this method *exponential forgetting*, while Norton and Mo call it *scalar bound inflation* because of its equivalence to progressively "loosening" the past γ bounds (see (29)) as time goes on. Norton and Mo also point out that once *any* past optimal weight is tampered with, all weights in its future become suboptimal in the sense considered above and should, in principle, be reevaluated. As they acknowledge, however, this is not practically feasible in most applications.

One important detail must be made clear. We have noted in the discussion surrounding (46) that weight scaling *will effect both the covariance matrix* $C(n-1)$ *and parameter* $\kappa(n-1)$ *in such a way that the* overall ellipsoid matrix is not affected. This means that the "expansion" we desire in the present context will not take place if the scaling is carried out "properly." The remedy is to scale only the covariance matrix prior to optimization. That is, the scaled matrix $C_s(n-1)$ is used in constructing polynomial (44) or (45), but $\kappa(n-1)$ is not scaled until after the data set is considered. It will be noted that a formal problem arises with respect to our previous discussion, since the weights $\lambda_n(1), \ldots, \lambda_n(n-1)$ are used in the scaled covariance matrix, while the weights $\lambda_{n-1}(1), \ldots, \lambda_{n-1}(n-1)$ remain in $\kappa(n-1)$. This nuance, however, is necessary to achieve the desired result. Since the theoretical developments underlying the UOBE algorithm do not, strictly speaking, support identification of time-varying systems, the use of UOBE for adaptive purposes is based on heuristic procedures of which this "improper" scaling is a part.

With the exception of the minor issue discussed above, exponential fogetting amounts to a UOBE algorithm with non-unity scaling. Accordingly, it is somewhat inefficient because $\mathcal{O}(0.5m^2+km)$ multiplies are required at each n just to implement the forgetting factor (see Section 6.5.1). Further, it has not been found to be effective for adaptation in simulations, unless the system dynamics are changing rather slowly [87]. We shall discuss this effect in the simulations below. The reason is that the exponential decay of the influence of past data sets is frequently not fast enough to discount very heavily weighted data, so that the estimate does not respond to fast changes in the system dynamics. To counter this problem, a small α might be proposed, but this has the effect of creating a very small effective "window" which, in turn, leads to high variability and loss of spectral resolution. From the point of view of the ellipsoid, the pre-scaling by α results in an inflation in the volume by a factor inversely proportional to $\alpha^{m/2}$. Therefore, a large α results in little change in the ellipsoid, while a small α causes severe inflation of the ellipsoid and induces a series of "oscillations" in the ellipsoid size. Further, this cycle of expanding and shrinking ellipsoids causes a tendency to accept more data sets. Therefore, from the SM point of view, small values of α are least desirable. These phenomena will be illustrated in the simulation studies below.

The adaptive UOBE algorithms to which we now turn do not depend on a fixed factor, such as α, to expand the ellipsoid volume. However, these algorithms expand the ellipsoid by (selectively) removing previously accepted influential data sets from the system, either partially or completely, and therefore, relinquishing their influence on the current ellipsoid, thereby allowing it to expand and adapt to the changes in the signal dynamics.

6.4.3 Forgetting by Back-Rotation

The forms of adaptation to be discussed here do not fit as neatly into our previous formalisms as does exponential forgetting. Let us begin with the general UOBE algorithm for which the scaling sequence is $\zeta(\cdot)$. Having obtained an estimate $\boldsymbol{\Theta}(n-1)$ with associated covariance matrix $\boldsymbol{C}(n-1)$, we wish to consider the incoming data set $(\boldsymbol{y}(n), \boldsymbol{x}(n))$. Before doing so, however, and even *prior to scaling*, we adjust the existing system of equations in order to "downweight" the influence of some, or all, of the previous data sets. The means by which the existing data sets are modified is to, in effect, introduce different minimization weights. In the present situation, we wish to change (in general, all) weights used at time $n-1$, $\lambda_{n-1}(t)$, $t \in [1, n-1]$ to a new set, say $\lambda_{n-1,d}(t)$, $t \in [1, n-1]$. We assume that the new set of weights is not obtained by simple scaling, but restrict ourselves to the case in which the new weights will be of the form

$$\lambda_{n-1,d}(t) = [1 - \varphi_{n-1}(t)]\lambda_{n-1}(t), \quad t \in [1, n-1] \tag{53}$$

where $0 \leq \varphi_{n-1}(t) \leq 1$. In effect, we wish to remove the a fraction equivalent to $\varphi_{n-1}(t)$ of the contribution of the data set at time t from the estimate. Not surprisingly, this can be accomplished by treating $(\boldsymbol{y}(t), \boldsymbol{x}(t))$ as a new data set with "weight" $-\varphi_{n-1}(t)\lambda_{n-1}(t)$. In the MIL-WRLS context, no modifications to the basic algorithm are required. In the context of QR-WRLS where the square root of the weight is taken (see discussion below (28)), this is achieved by using weight $[\varphi_{n-1}(t)\lambda_{n-1}(t)]^{-1/2}$, and introducing some sign changes in the algorithm [26],[27],[88].

The method by which an data set is completely removed from the previous system using QR decomposition by Givens' rotations has been called *back rotation* in the papers cited above. We will use this term to refer to removal by both QR-WRLS and MIL-WRLS even though it loses its technical significance for the latter. The technique to *partially* remove a prior data set is a simple generalization suggested in

[32],[87],[88]. Let us now formalize this procedure, focusing on QR-WRLS (similar developments can be obtained for the MIL-WRLS version).

Suppose we, in principle, sequentially modify weights as described above, beginning at time $t = 1$. The following (and similar) quantities will pertain to the "downdated" system of equations whose weights have been modified to time t: $\boldsymbol{C}_d(n-1:t), \boldsymbol{T}_d(n-1:t), \boldsymbol{D}_{1,d}(n-1:t), \boldsymbol{\Theta}_d(n-1:t), \kappa_d(n-1:t)$, where each is similar to familiar quantities in the foregoing discussions. We also omit the second argument if it is $n-1$. For example, $\boldsymbol{C}_d(n-1) \stackrel{\text{def}}{=} \boldsymbol{C}_d(n-1:n-1)$. Following the modification of the t^{th} data set, the downdated equation to be solved in the QR-WRLS method (if the solution were desired) is

$$\boldsymbol{T}_d(n-1:t)\boldsymbol{\Theta}_d(n-1:t) = \boldsymbol{D}_{1,d}(n-1:t). \tag{54}$$

The downdated ellipsoid matrix is $\boldsymbol{C}_d(n-1:t)/\kappa_d(n-1:t)$ where

$$\begin{aligned} \boldsymbol{C}_d(n-1:t) &= \boldsymbol{T}_d^H(n-1:t)\boldsymbol{T}_d(n-1:t)\ , && (55) \\ \kappa_d(n-1:t) &= \boldsymbol{\delta}_{1,d}(n-1) + \tilde{\kappa}_d(n-1:t), && (56) \end{aligned}$$

with $\boldsymbol{\delta}_{1,d}(n-1) \stackrel{\text{def}}{=} \operatorname{tr}\{\boldsymbol{D}_{1,d}^H(n-1)\boldsymbol{D}_{1,d}(n-1)\}$ and

$$\tilde{\kappa}_d(n-1:t) \stackrel{\text{def}}{=} \tilde{\kappa}(n-1:t-1) - \varphi_{n-1}(t)\lambda_{n-1}(t)\gamma(t)\left(1-\gamma^{-1}(t)\parallel \boldsymbol{y}(t)\parallel^2\right). \tag{57}$$

The quantity $\tilde{\kappa}(n-1:0) \stackrel{\text{def}}{=} \tilde{\kappa}(n-1)$ represents the updated value of $\tilde{\kappa}$ which includes $(\boldsymbol{y}(n-1), \boldsymbol{x}(n-1))$. Equations (56) and (57) follow immediately from the definition of κ found in (40) and a basic understanding of the back rotation process being undertaken. Following all necessary downdating just prior to time n, the algorithm uses the downdated system to compute the downdated *and scaled* quantities

$$G_{d,s}(n) \stackrel{\text{def}}{=} \boldsymbol{x}^H(n)\boldsymbol{C}_d^{-1}(n-1)\boldsymbol{x}(n)\zeta(n) = \boldsymbol{x}^H(n)\boldsymbol{C}_{d,s}^{-1}(n-1)\boldsymbol{x}(n). \tag{58}$$

$$H_{d,s}(n) \stackrel{\text{def}}{=} \boldsymbol{x}^H(n)\boldsymbol{C}_d^{-2}(n-1)\boldsymbol{x}(n)\zeta^2(n) = \boldsymbol{x}^H(n)\boldsymbol{C}_{d,s}^{-2}(n-1)\boldsymbol{x}(n) \quad \text{(necessary for trace only)}. \tag{59}$$

and

$$\kappa_{d,s}(n-1) \stackrel{\text{def}}{=} \frac{\kappa_d(n-1)}{\zeta(n-1)}. \tag{60}$$

In turn, these numbers are used in place of their "non-downdated" counterparts in (44) or (45) to test for the existence of, and to compute, the optimal weight for $(\boldsymbol{y}(n), \boldsymbol{x}(n))$. Once the optimal $\lambda_n^*(n)$ is found, we define

$$\lambda_n(t) = \begin{cases} \lambda_{n-1,d}(t), & t = 1, 2, \ldots, n-1 \\ \lambda_n^*(n), & t = n \end{cases} \tag{61}$$

for the next iteration.

The process described above would appear to be extraordinarily computationally expensive in general, since each past weight is modified at each $n-1$. Recall, however, that "most" data sets are never included in the estimate in the first place ($\lambda_n^*(n) = 0$) and therefore the system need not be downdated at these times. If the data set at time t, for example, was not included in the estimate, then formally $\boldsymbol{C}_d(n-1;t+1) = \boldsymbol{C}(n-1;t)$, $\boldsymbol{T}_d(n-1;t+1) = \boldsymbol{T}(n-1;t)$, etc., and no computation is required. A similar situation obtains if a data set, say at time t, was completely removed by back-rotation so that $\lambda_{n-1}(t) = 0$. In this case, no computational effort is required to downdate this data set at time $n-1$. Further, in many cases the modification of a particular data set is not desired. If, for example, the data set at t is not to be altered, then $\varphi_{n-1}(t) = 0$, and no computation is necessary. Finally, note that when the "new" data set at n is rejected ($\lambda_n^*(n) = 0$), then $\boldsymbol{T}(n) = \boldsymbol{T}_d(n-1)$ and $\boldsymbol{\Theta}(n) = \boldsymbol{\Theta}_d(n-1)$, and, once again, no computation is actually required.

A wide range of adaptation strategies is inherent in the general formulation described above, many of them computationally inexpensive. Three cases are considered:

1. l is a constant *window length* and, for all n,

$$\varphi_{n-1}(t) = \begin{cases} 1, & t = n-l \\ 0, & \text{other } t \end{cases} ; \tag{62}$$

2. l is a constant window length and, for all n, $1 - \varphi_{n-1}(t)$ is zero prior to time $n - l + 1$ and smoothly (perhaps linearly) tapers to unity at time n.

3. $\mathcal{T}_{n-1}$ is some past set of equations to be "forgotten," and,

$$\varphi_{n-1}(t) = \begin{cases} 1, & t \in \mathcal{T}_{n-1} \\ 0, & t \notin \mathcal{T}_{n-1} \end{cases} . \tag{63}$$

The first case above corresponds to the use of a sliding window of length l, outside of which all previous data sets are completely removed. Norton and Mo have called this case *fixed memory bounding* [86] while Deller and Odeh have called it simply *windowing* and have suggested an efficient algorithm for implementing it [32],[87],[88]. The estimate at time n covers the range $[n - l + 1, n]$. The windowing technique is made possible by the ability to completely and systematically remove data sets at the trailing edge of the window. Only one back-rotation is required prior to optimizing at time n, and this is only necessary if $\lambda_{n-1}(n-l) = \lambda_{n-l}^*(n-l) \neq 0$.

Case 2 represents another windowed, or finite memory, approach, but in this case the window is permitted to taper smoothly to zero as it moves into the past. For example, the effective weights might decrease linearly when moving toward the trailing edge of the window. Hence, the data set at the trailing edge has an effective weight of $l^{-1}\lambda_{n-1}(n+1-l)$ and the data set to be rotated in has an effective weight of $\lambda_n^*(n)$. To reiterate an important point made in the general discussion above, although each data set must be partially rotated out l times, only those data sets that were previously accepted (in the past l recursions) need to be considered by the algorithm. Let us refer to this method as *tapered forgetting.*

We remark that a tapered window can smooth the estimate, but at the expense of a significant amount of computation. Each accepted point must be back rotated about l times where it can be true that $l >> m$. Depending on the circustances, the extra computation required to implement the smoother window may not be warranted by the extra amount of computation (see simulations below and [87] for further discussion).

Case 3 is a different type of strategy which Deller and Odeh call *selective forgetting.* This technique selectively chooses the data sets to be removed from the system based on certain user defined criteria in order to remove their influence from the system. The selection process can be, for example, to remove (or downweight) only the previously heavily weighted data sets, to remove the data sets that were accepted in regions of abrupt change in the signal dynamics, or to remove the data sets starting from the first data set and proceeding sequentially. Whatever the criteria, a fundamental issue is to detect *when* adaptation is needed to improve the parameter estimates. This issue is further investigated in the simulation studies below.

6.4.4 Illustration

The simulation results of the several variations on the general adaptive SM-WRLS algorithm are shown. We continue with the example initiated in Section 6.3.2. The reader is reminded that only the results for a_{1*} are shown.

The first experiment concerns the use of exponential forgetting. We noted above that this form of adaptation will often fail to track quickly varying parameters. This was the case with both the "slow" and "fast" systems used here for any reasonable forgetting factor. The problem is the inability to "forget" heavily weighted data quickly enough. Accordingly, we tried the experimental procedure of replacing any optimal weight by unity before incorporating the chosen data set. The results for forgetting factor $\alpha = 0.99$ are shown in Fig. 9. Whereas the "weight override" might be expected to cause a vastly increased fraction of the data to be used, in fact only fractions $\rho = 0.073$ and $\rho = 0.094$ of the data were used for the fast and slow systems, respectively. Clearly, the tracking is very good for the slow system, and perhaps acceptable for some purposes for the f^st system. $\alpha = 0.99$ was the smallest forgetting factor which would give "acceptable" tracking in the sense of reaching the "target" values during the each cycle in the fast case. This large forgetting factor is responsible for the variability seen in the regions which are easier to track. The estimate can, of course, be smoothed by choice of a smaller α. In the slow case, the estimate can be smoothed considerably before time resolution is lost.

Figure 10 shows the volume traces, and Fig. 11 sgn$\{\kappa(n)\}$, as functions of n. As in the "nonadaptive" experiments, we observe a tendency for κ to become negative when the parameters change abruptly. The problem is not as "serious" as it was with the nonadaptive cases because the algorithm tends to "recover." That is, when κ goes negative, the volume goes into a trend of expansion (due to forgetting) ultimately leading to a positive κ. We can imagine that the ellipsoid ultimately becomes large enough to "recapture" the moving parameters, and bring the identification back into line with the underlying principles. However, a positive κ is a necessary but not sufficient condition for this to be true (for the ellipsoid to contain the true parameters), so we must be careful with this analysis. In the use of specifically adaptive algorithms, generally we observe that the identification does enter phases in which it operates outside the principles of SM identification, but that it tends to recover and operate properly due to the adaptation measures.

Figure 12 shows the simulation result of the windowed SM-WRLS algorithm using a window of length 250. This strategy uses only the fraction $\rho = 0.094$ of the data for the fast system, and $\rho = 0.10$ for the slow system. Since most of the data sets rotated into the system are eventually rotated out, this strategy *effectively* uses about twice the number of data sets rotated in ($b \approx \rho$). More data than with the unmodified SM-WRLS algorithm are used, but more accurate estimates result and the time varying parameters are tracked more quickly and accurately. For the slow system, we observe that the algorithm behaves properly (in the sense that κ remains positive) for nearly the entire range. For the fast system, there are relatively short recovery phases ($\approx$ 100 points) required after each abrupt change in dynamics. The volume traces behave as expected with trends toward increase (due to "forgetting") interrupted by occassional decreases as data sets are accepted. An example for the slow case is shown in Fig. 13.

As expected, the estimates are smoothed, time resolution lessened, and the fraction of accepted points decreases, as window lengths increase. The parameter estimates for the fast system and window length $l = 500$ are shown in Fig. 14 as an example. The fractions of data accepted are $\rho = 0.070$ and $\rho = 0.060$ for the fast and slow systems, respectively. Also not unexpectedly, recovery periods, which were virtually nonexistent for the slow system with $l = 250$, are now present with $l = 500$, though still for a small fraction of the time. The recovery phases for the fast case increase in duration so that they now occupied more than one-third of the range.

As the window length continues to increase, the effects reported above continue to change in the expected ways. In particular, it is not unexpected that at some point, the process would begin to disintegrate from a theoretical point of view, since as $l \rightarrow \infty$, the "windowed" algorithm approaches "nonadaptive" SM-WRLS. In fact, the recovery phases for the fast system continue to increase until at $l = 1000$, the parameter κ is negative for nearly the entire range following the initial change. The process erodes and fails to track properly after the first one and one-half cycles. Interestingly, only the fraction $\rho = 0.030$ of the data are used in this estimate, and most of these are taken in the initial cycles. The parameter estimate and κ are shown in Figs. 15(a) and 15(b) for this case.

As an aside, we observe that this and similar UOBE algorithms are frequently capable of tracking while using small fractions of data, even "in violation of theory" ($\kappa < 0$). However, empirically, the estimate frequently diverges after a problem dependent interval. This suggests the possibility of monitoring κ and "resetting" the algorithm after a sustained period of "violation." (The selective forgetting approach to be described can be interpreted as a highly conservative version of this procedure.) Such a procedure would be quite unpredictable unless the theoretical analysis the of the process under conditions of negative κ is forthcoming. In fact, this "unpredicitability" is the same problem encountered when no adaptation measures are taken, but with some alleviation or postponement of the undesirable performance.

The selective forgetting strategy selects the data sets to be (partially or completely) removed from the estimate according to certain criteria in order to remove their influence on the result. In keeping with foregoing developments, the selection procedure used here is to monitor the parameter κ for positivity. When it is found that $\kappa(n) < 0$ for some n, we simply back-rotate previously accepted data sets, beginning with the oldest data set remaining in the estimate, until this number becomes positive. We reiterate that $\kappa(n)$ merely being positive does not insure that the true parameters have returned to the interior of the bounding ellipsoid, so the procedure is purely experimental. This technique yields the simulation results shown in Figs. 16(a) and 16(b) for the fast and slow systems, respectively. The identification of the fast system uses only $\rho = 0.050$ of the data, 84.0% of which are back-rotated for adaptation, so that $b = 0.042$. For the slow system, the rates are $\rho = 0.064$ and $b = 0.047$. While usually requiring even less computational effort, this method is seen to provide superior estimates to those obtained from the other adaptive techniques. For the fast system, it is noted that large errors occur in the estimates at the points of discontinuity in the true parameters. At some computational expense, this could be potentially be resolved by, after forgetting, *removing the data set* $(\boldsymbol{y}(n), \boldsymbol{x}(n))$ *which caused* $\kappa(n)$ *to go negative* and recomputing the weight (or some similar heuristic).

6.5 Implementing the UOBE Algorithm in $\mathcal{O}(m)$ Time

6.5.1 Complexity of the Basic UOBE

From a signal processing point of view, one of the most interesting aspect of a UOBE algorithm is its inherent ability to select only data points which are informative in the sense of refining the feasible set. The fact that typically 70% – 95% of the data are rejected by this criterion would seem to imply a remarkable savings in computation. We have noted in Section 6.2, however, that this is only true to the extent that the SM preprocessing of the incoming data set is negligibly expensive compared with the inclusion of it in the estimate. In this section, we examine some factors related to this complexity issue.

A comparison of the computational loads of the various algorithms discussed in this paper is shown in Table 1. A complex floating point operation (cflop) is taken to be approximately one complex multiplication plus one complex addition operation. Additions which are unpaired with multiplications are ignored.

The numbers shown arise from efficient procedures which avoid recomputation of quantities, for example $\varepsilon(n, \Theta(n-1))$, which are shared among different operations. Only numbers dependent upon m and k are shown with constant (usually small) numbers of cflops ignored. Not shown in the table are tallies to update $\kappa(n)$, and the number of operations needed to compute an optimal weight when the data set is accepted. $\kappa(n)$ requires about 4 cflops in the MIL-WRLS cases, and $(m+1)k$ in the QR-WRLS cases. Optimal weights require a small number of cflops (about 25) which may be thought of as nearly independent of m and k since all quantities computed are used for other purposes. Figures shown are based on volume optimization, but the trace tallies are nearly identical.

The following analyses are applicable to the usual case in which the number of outputs from the system, k, is small relative to the number of parameters estimated, m. In fact, for simplicity, let us set $k \equiv 1$ (MISO system). The general conclusions reached, however, are valid when $k << m$, and we shall continue to show $\boldsymbol{y}(\cdot)$ and $\varepsilon(\cdot, \Theta(\cdot))$ as vectors.

As a standard of comparison, we note that conventional MIL-WRLS requires $\mathcal{O}(3m^2)$ cflops per n with an additional $\mathcal{O}(m^2/2)$ required to include a scaling sequence $\zeta(\cdot)$. For QR-WRLS, $\mathcal{O}(2.5m^2)$ cflops per n are required with an additional $\mathcal{O}(m^2/2)$ needed for scaling.

From Table 1, we may state that the average operation count for an adaptive UOBE algorithm implemented on a sequential machine is approximated by

$$f_{opt} \sim \mathcal{O}(c_1 m^2) + s\mathcal{O}(m^2/2) + b\mathcal{O}(c_2 m^2) + \rho\mathcal{O}(c_3 m^2) \text{ flops per } n \tag{64}$$

where, s is unity if the algorithm involves a scaling sequence and/or a forgetting factor and is zero otherwise; ρ is the average number of data sets accepted per n; b is the average number of back-rotations performed per n; and c_1, c_2 and c_3 are small numbers (all in the range 0.5 - 2.5) which depend upon whether MIL-WRLS or QR-WRLS is used. The first term is due to the procedure which checks for information in the incoming data. The others are attributable to scaling, adaptation, and solution update, respectively. The subscript "*opt*" is used to indicate that the proper optimization described above is used. Apparently, the UOBE algorithm, as presently formulated, is an "$\mathcal{O}(m^2)$" process. The objective of the section below is to demonstrate a method for reducing the effective complexity to $\mathcal{O}(m)$ by reducing the checking cost, thereby making a UOBE algorithm a desirable alternative to standard RLS-based methods from a computational point of view. We also mention a parallel processing approach which likewise achieves the $\mathcal{O}(m)$ goal.

Before detailing the methods, some points about the use of the approximation "$\mathcal{O}(m)$" are necessary. The first concerns a practical matter. The objective in the following is to reduce the computational complexity of the algorithms to an *average* of $\mathcal{O}(m)$ flops per n. It will be appreciated that, without data buffering, the data flow is still limited by the worst case $\mathcal{O}(m^2)$ computation. However, if a buffer is included, the algorithm easily be structured to operate in $\mathcal{O}(m)$ average time per n. Further, by using interrupt driven processing of the checking procedure, it may be possible to reduce the average time even further. Other points concern algorithmic details. We see from (64) that the use of a unity scaling sequence (SM-WRLS algorithm) is required in order to avoid an invariant $\mathcal{O}(m^2/2)$ flops per n. We specifically assume the use of this algorithm below, although the $\mathcal{O}(m)$ checking procedure to be developed does not depend on this choice. Secondly, we note that even if the checking procedure can be made $\mathcal{O}(m)$, terms $b\mathcal{O}(m^2)$ and $\rho\mathcal{O}(m^2)$ (typically $b \approx \rho$) persist in (64). This means that to truly achieve $\mathcal{O}(m)$ complexity, b and ρ must be $\mathcal{O}(1/m)$. For large m, this will not be always be the case. In fact, some experimental evidence suggests, not unexpectedly, that ρ increases, rather than decreases, with increasing m. For "large" m (conservatively, say, $m > 10$), therefore, it is the case that the complexity is reduced to $\mathcal{O}(\rho m^2)$ by $\mathcal{O}(m)$ checking. It should be clear however, that neither $\mathcal{O}(m)$ nor $\mathcal{O}(\rho m^2)$ complexity can be achieved if the checking procedure remains $\mathcal{O}(m^2)$. We therefore pursue an $\mathcal{O}(m)$ test for information in an incoming data set.

With UOBE, the number of computations needed for each n depends on whether the corresponding data set is accepted for processing by the optimization criterion. UOBE is essentially reverts to MIL-WRLS or QR-WRLS when a data set is accepted. Since most of the time the data set is rejected, for significant complexity gain, a UOBE algorithm must require many fewer than $\mathcal{O}(3m^2)$ flops for checking. We digress momentarily, therefore, to view some of the details of the checking procedure.

In principle, the information checking procedure for the volume or trace algorithms consists of forming either $F_v(\lambda)$ or $F_t(\lambda)$ of (44) and (45), then solving for a positive root. In either case, however, the polynomial can have at most *one positive root* (see Proposition 2 in Appendix A.1). The test therefore reduces to one of testing the zero order term for negativity. When the test is successful, then the root solving and updating procedes, requiring the standard MIL- or QR-WRLS load, plus a few operations for finding the optimal weight. The most expensive aspect of this information test is the computation of the quantity $G_d(n)$ in the case of volume minimization, or $H_d(n)$ for trace minimization. (For generality, we assume downdating is used. If this is not the case, it is merely necessary to drop the subscripts "d" on all quantities.) In the MIL-WRLS case, this requires $\mathcal{O}(m^2)$ flops. In the QR-WRLS case, a problem arises because $G_d(n)$ depends upon the *inverse* normal matrix, $\boldsymbol{C}_d^{-1}$, which is not otherwise used in the process. Similarly, $H_d(n)$ depends on $\boldsymbol{C}_d^{-2}$. In the paper by Deller [27], the following method has been suggested to sidestep this problem for $G_d(n)$: Recalling the definition of $G_d(n)$ and noting and (27), we can write

$$G_d(n) = \boldsymbol{x}^H(n)\boldsymbol{T}_d^{-1}(n-1)\boldsymbol{T}_d^{-T}(n-1)\boldsymbol{x}(n) \stackrel{\text{def}}{=} \boldsymbol{g}_d^H(n)\boldsymbol{g}_d(n) = \|\,\boldsymbol{g}_d(n)\,\|^2 . \tag{65}$$

Since $\boldsymbol{x}(n) = \boldsymbol{T}_d^H(n-1)\boldsymbol{g}_d(n)$, and the matrix $\boldsymbol{T}_d^H(n-1)$ is lower triangular, $\boldsymbol{g}_d(n)$ is easily found from the available quantities at time n by forward substitution. The procedure can be repeated to compute $H_d(n)$ if needed, since

$$H_d(n) = \boldsymbol{g}_d^H(n)\boldsymbol{T}_d^{-1}(n-1)\boldsymbol{T}^{-H}(n-1)\boldsymbol{g}_d(n) \stackrel{\text{def}}{=} \boldsymbol{h}_d^H(n)\boldsymbol{h}_d(n) = \|\,\boldsymbol{h}_d(n)\,\|^2 . \tag{66}$$

The total computational load for this method is $\mathcal{O}(m^2/2)$ for $G_d(n)$ and $\mathcal{O}(m^2)$ for $H_d(n)$ which is far less than the effort required to invert $C_d(n-1)$. When $MIL-WRLS$ is used for the covariance and parameter update in UOBE, the checking ("precomputation" of $G_d(n)$) removes $\mathcal{O}(m^2)$ flops from the update load, but the checking does not contribute to the update for QR-WRLS.

6.5.2 Suboptimal Tests for Innovation in the Data

In spite of the simplifications suggested above, the computation of the quantities $G_d(n)$ and $H_d(n)$ remain of $\mathcal{O}(m^2)$ complexity. Clearly, the trick is to try to avoid the computation of these numbers in the information checking procedure. Deller and Odeh [32] have proposed a simple suboptimal updating rule: Include the data set at time n only if

$$\| \varepsilon(n, \boldsymbol{\Theta}(n-1)) \|^2 < \gamma(n). \tag{67}$$

This rule is used for both the volume and trace minimization versions of the UOBE and is not affected by inclusion of a *causal* scaling sequence, $\zeta(\cdot)$. The rationale for this test is simple. The zero order cofficients a_0 and b_0, of (44) and (45), respectively, will never be positive if the test is met. In the volume case, for example, the suboptimal check tests whether a_0 is negative *if the term* $-\kappa_d(n-1)G_d(n)$ *is neglected.* This ignored term is always negative and becomes small as n increases if no forgetting is used. For a given set of preceding optimal weights, $\lambda^*(1), \ldots, \lambda^*(n-1)$, *the suboptimal test will never fail to accept a data set which would have been accepted by the optimal test.* A similar analysis applies to the coefficient b_0 of the trace algorithm.

A deeper analysis of this suboptimal test has been made for the volume algorithm by Deller and Odeh [32]. Let us denote the estimatior *rror* matrix at time n by

$$\tilde{\boldsymbol{\Theta}}(n) \overset{\text{def}}{=} \boldsymbol{\Theta}_* - \boldsymbol{\Theta}(n). \tag{68}$$

The following inequality results immediately from (37) - (40):

$$\tilde{\boldsymbol{\Theta}}^H(n)\boldsymbol{C}(n)\tilde{\boldsymbol{\Theta}}(n) < \kappa(n). \tag{69}$$

While it is tempting to view $\kappa(n)$ as a bound on $\tilde{\boldsymbol{\Theta}}(n)$ (see discussion of the D-H OBE algorithm below), it is important to note that each side of this inequality is dependent upon $\lambda_n(n)$. In fact, let us temporarily write the two key quantities as functions of $\lambda_n(n)$, $\boldsymbol{C}(n, \lambda_n(n))$ and $\kappa(n, \lambda_n(n))$, and consider the usual volume quantity to be minimized at time n,

$$\mu_v(n) = \det[\kappa(n, \lambda_n(n))\boldsymbol{C}^{-1}(n, \lambda_n(n))]. \tag{70}$$

It is assumed that enough data sets have been included in the covariance matrix at time $n-1$ so that its elements are iarge with respect to the incoming data[14]. If a causal scaling sequence is included, the quantity

[14] The validity of this assumption depends to some extent on the choice of scaling sequence $\zeta(\cdot)$ if one is included.

$\det \boldsymbol{C}(n, \lambda_n(n))$ is readily shown to be monotonically increasing with respect to $\lambda_n(n)$ on the interval $[0, \infty)$ [87], with $\boldsymbol{C}(n, 0) \overset{\text{def}}{=} \boldsymbol{C}(n-1, \lambda^*_{n-1}(n-1))$. Under the assumption above, $\det \boldsymbol{C}(n, \lambda_n(n))$ will not increase significantly over reasonably small values of $\lambda_n(n)$. The attempt to maximize $\det \boldsymbol{C}(n, \lambda_n(n))$ in (70) causes a tendency to increase $\lambda_n(n)$ in the usual optimization process. However, the attempt to minimize $\kappa(n, \lambda_n(n))$ generally causes a tendency toward small values of $\lambda_n(n)$, unless a minimum of $\kappa(n, \lambda_n(n))$ occurs at a "large" value of $\lambda_n(n)$. To pursue this idea and further points of the argument, we use two key facts about $\kappa(n, \lambda_n(n))$ which are given in Proposition 3 in Appendix A.1. These are that $\kappa(n, \lambda_n(n))$ is either monotonically increasing on positive λ's or it has a single minimum. A necessary and sufficient test for that minimum is (67). Accordingly, it can be argued that: If $\det \boldsymbol{C}(n, \lambda_n(n))$ is increasing, but not changing significantly over reasonably small values of $\lambda_n(n)$, then it is sufficient to seek $\lambda^*_n(n)$ which minimizes $\kappa(n, \lambda_n(n))$. If $\kappa(n, \lambda_n(n))$ is monotonically increasing on $\lambda_n(n) \geq 0$, this value is $\lambda_n(n) = 0$ which corresponds to rejection of $(\boldsymbol{y}(n), \boldsymbol{x}(n))$. It suffices, therefore to have a test for a minimum of $\kappa(n, \lambda_n(n))$ on positive $\lambda_n(n)$. As noted above, a simple test is embodied in condition (67). If this test is met, it is then cost effective to proceed with the standard optimization centered on (44). Otherwise, the explicit construction and solution of a_0 of (44) can be avoided.

In fact, this suboptimal test for innovation is similar to that used in the D-H OBE algorithm reported in [24] and discussed in Section 6.2.2. The test used in D-H OBE is to accept the incoming data set only if[15]

$$\varepsilon^2(n, \boldsymbol{\theta}(n-1)) < \gamma(n) - \kappa(n-1) \,. \tag{71}$$

This inequality likewise tests for a minimum of $\kappa(n)$ with respect to $\lambda_n(n)$, and differs in form from (67) because of the noncausal scaling factors (see (22) and surrounding discussion). There has been some controversy in the literature as to the meaning of this test. Dasgupta and Huang argue simply that $\kappa(n)$ is "a bound on the estimation error," and should be minimized. Indeed, the minimization of $\kappa(n)$ is the optimization criterion used in D-H OBE, and no apparent connection to a set measure on the underlying ellipsoid is made. Dasgupta and Huang's claim has been disputed by Norton and Mo [86] and is not clearly supported by the *heuristic* arguments above, because the relative independence of $\boldsymbol{C}(n)$ and $\lambda_n(n)$ is not tenably argued. Nevertheless, examination of the *analytical* arguments above does reveal some interesting similarities between the use of the D-H test and the suboptimal test (67). These revelations nearly (but not quite) provide justification for the D-H test.

While it is not exploited in the D-H OBE algorithm for reasons discussed in Section 6.2.2, the D-H hyperellipsoid nevertheless *does* have a volume at each n. Liu *et al.* [68] have recently shown[16] that there

[15] A scalar error is shown since this algorithm is developed for a SISO model.

[16] Somewhat unexpectedly, perhaps, because of the noncausal scale factors and the nonlinearities in $\lambda_n(n)$ which would be expected to arise.

is a quadratic equation in λ similar to (44) which must be solved to find the optimal "volume" root (see Corollary 2 in Appendix A.1). For the sake of discussion, let us call the weight which optimizes *volume* λ^v, and that which optimizes $\kappa(n)$ λ^κ. The volume quadratic has the amazing property that its zero order coefficient is *identical* to that in (44), a_0, and may be checked for negativity as a necessary and sufficient test for the existence of an optimal weight λ^v. Similarly to the suboptimal SM-WRLS strategy, the D-H test (71) comes quite close to being a sufficient test for negativity of a_0, and therefore a sufficient test for whether the volume can be diminished. Further, even if the D-H weight λ^κ is not equal to λ^v (and it likely will not be), it can still be shown to shrink the volume [68] as long as $\lambda^v > 0$ exists. Consequently, if (71) were exactly a test for $a_0 < 0$, then it would follow that the D-H algorithm, by reducing κ simultaneously reduces volume. The fact that λ^κ does not *optimally* minimize volume is apparently a small price to pay for the ability to prove convergence. Regrettably, the test (71) is not quite sufficient to assure $a_0 < 0$. Additionally, it is must be true that $G_s(n) \geq mk$. This condition is most likely to be met for small n, precisely when the most data sets are likely to be accepted. However, there is no assurance in general that this condition will prevail. Consequently the D-H OBE test, while part of a very different approach, comes intriguingly close to being justified by the same means as the suboptimal test associated with SM-WRLS, but falls somewhat short. Some further theoretical work may ultimately resolve the apparent problem.

Interestingly, the Deller-Odeh test (67) could be used as a "suboptimal" criterion for accepting data in the D-H OBE algorithm. That is, (71) is satisfied whenever (67) is met. The benefit of this suboptimal approach would be that it would assure that volume would be decreasing at each step by minimizing κ, thereby providing a clear *pointwise* justification for the D-H approach. Before such an approach were adopted, it would be necessary to ascertain that the convergence result (which is the *raison-d'etre* for the D-H OBE algorithm) is preserved.

6.5.3 Computational Complexity of UOBE Algorithms with Suboptimal Checking

Recall that the purpose of this pursuit is to find a way to avoid the m^2 flops necessary to carry out the checking process in the optimal algorithm. The test (67) requires only $\mathcal{O}(m)$ cflops, so that the revised operation count is[17]

$$f_{subopt} \sim \mathcal{O}(m) + s\mathcal{O}(m^2/2) + b'\mathcal{O}(c_3 m^2) + \rho'\mathcal{O}(c_2 m^2) \quad \text{cflops per } n \tag{72}$$

where b' is the average number of back-rotations per n under the suboptimal checking policy; ρ' represents the fraction of the data sets which are included in the update; and s indicates whether scaling is used. As we have stessed above, even as $\rho' \rightarrow 0$ the suboptimal algorithm remains of $\mathcal{O}(m^2/2)$ complexity per n on

[17] Note that $G_d(n)$ is no longer computed in the checking phase so that the operation count for MIL-WRLS is the full $\mathcal{O}(3m^2)$.

a sequential machine, *unless* SM-WRLS ($\zeta(n) = 1 \ \forall n$) is used. Herein lies one of the most compelling reasons for the choice of the simplest form of UOBE algorithm in signal processing.

In light of (72), let us briefly consider the computational loads imposed by the specific adaptation strategies described above. In each case, we assume QR-WRLS underlies the process, but the discussion for MIL-WRLS is similar.

Of the adaptation methods described above, exponential forgetting is the most expensive computationally, unless the UOBE algorithm already employs a non-unity scaling sequence. If the algorithm does not employ a scaling sequence ($s = 0$ in (64) and (72)), then the inclusion of the forgetting factor essentially imposes one ($s = 1$) and adds $\mathcal{O}(m^2/2)$ cflops per n. If the algorithm does contain a scaling sequence, then the forgetting factor can be combined with it prior to scaling, requiring only one cflop per n.

Since back-rotation is essentially equivalent to a covariance (or $\boldsymbol{T}(n)$) update[18] for an incoming data set, each of these rotations takes $\mathcal{O}(2.5m^2)$ cflops. If b back-rotations are performed on the average at each n, then effectively $\mathcal{O}(2.5bm^2)$ additional operations are required by the adaptation procedure. Since ρ is usually small, whether a particular adaptation strategy is cost-effective depends on the number b. For simple windowing, for example, $b \approx \rho$ and the adaptation adds negligibly to the computational load. For tapered forgetting, on the other hand, $b \approx \rho l$, where l is the effective window length which may be quite large. In this case, the adaptation might be the dominant cost requirement, completely overshadowing any savings gained by suboptimal testing, for example. A high computational cost, therefore, might be incurred for the benefits of a tapered window in the analysis. Finally, the cost of the selective forgetting routine depends entirely upon the criterion employed for deciding to back-rotate a previous data set, which, in turn, determines the value of b. An example will be discussed below in the simulation studies.

6.5.4 Illustration

To illustrate the efficient methods based on suboptimal checking, we continue the study of the systems described in Section 6.3.2. Not unexpectedly, the suboptimal "nonadaptive" SM-WRLS algorithm fails to track either system properly. The result is similar to Fig. 7.

As illustrations of adaptive methods, we repeat the exponential forgetting and selective forgetting experiments performed above. We use the same techniques and conditions except that the suboptimal test is employed. Figure 17 shows the parameter estimates, and Fig. 18 the sgn$\{\kappa(\cdot)\}$ traces, resulting from exponential forgetting. Comparing Figs. 9 and 17 we see that in the case of the fast system, the parameters resulting from suboptimal testing track the true parameters more accurately for the first two cycles, but then show signs of "breakdown" in the third cycle unlike the parameters resulting with optimal checking. No definitive conclusions can be drawn from this comparison of fast system results. In particular,

[18] Note that a *parameter solution* update is *not* required, just the covariance update.

it should not be concluded that suboptimal testing will lead to faster disintegration of tracking. Indeed, by comparing Figs. 11 and 18, we see that there is a much greater tendency for κ to remain positive in the suboptimal case. In fact, there is evidence in the κ trace that the failure to track well in the third cycle is a transient effect from which the identification may recover. In the slow system case, somewhat more variance is seen in the parameter estimate with respect the optimal checking case, but this is apparently related to the many fewer data selected. Note the remarkable improvement in the κ behavior with respect to the optimal case, indicating that the identification is more likely operating within the principles of SM theory for suboptimal checking. Another important observation is that the number of data sets selected by suboptimal checking is many fewer (roughly half) that required by optimal checking. We currently have no explanation for these preferable behaviors of the suboptimal checking case, but, importantly, they have been quite generally observed across many simulations.

In a second experiment, selective forgetting is used in conjunction with the suboptimal testing. For this case, the parameter results are practically indistinguishable from those obtained using optimal checking (see Fig. 16). The notable difference is once again in the greatly reduced number of data used in the suboptimal checking experiments. For the fast system $\rho' = 0.022$ and $b' = 0.019$, and for the slow system $\rho' = 0.041$ and $b' = 0.028$. Again each of these fractions is roughly half the corresponding figure required in the optimal checking cases.

In summary, we generally observe that the suboptimal technique uses about half as many data, but produces comparable estimates to those obtained using the optimal procedure. This is true whether good or bad tracking results. This means that not only does the suboptimal procedure reduce the complexity of testing the data sets for innovation (motivation for its development), but it also reduces (by about a factor of two) the number of operations spent in rotating data sets into the system of equations. Further, suboptimal checking frequency results in more "meaningful" identification in the sense that κ has a much higher tendency to remain positive. Other examples using suboptimal checking are found in [32],[87],[89].

6.6 Convergence Issues and Colored Noise

In the following, we return the the case of time-invariant systems and discuss a few issues related to convergence and colored inputs.

One of the interesting and practical benefits of having interpreted UOBE algorithm as a WRLS algorithm with a bounded error "overlay" is the immediate consequence for convergence of the estimator. It is well-known that if the sequence $\varepsilon_*(\cdot)$ is wide-sense stationary, second moment ergodic almost surely (a.s.), white noise (see discussion surrounding (14)), then the WRLS estimator $\Theta(\cdot)$ will converge asymptotically to Θ_* a.s. (e.g. [45]). In the present case, we need only to add the qualifier that the UOBE algorithm not cease to accept data in order to lay claim to this useful result.

Likewise, we may even assert a.s. convergence of the WRLS estimate, albeit to a bias, when $\varepsilon_*(\cdot)$ is colored and persistently exciting[19] (p.e.) [47]. Even in the presence of colored errors, therefore, as long as the acceptance of data does not cease, and the "sampling" induced by data selection does not interfere with the p.e., we may expect the UOBE estimate to converge.

It would be interesting to have a precise understanding of the asymptotic behavior of the hyperellipsoidal *feasible set*, especially in the case of colored noise. Knowledge that the ellipsoid is vanishing (white noise), or becoming as small as possible (colored noise), could be very useful information indeed. In the white noise case, a sufficiently small ellipsoid could serve as a reinforcing indicator of convergence, and offer a means of determining error bounds on the estimate. In the colored case, a small feasible set (known to contain the true, unbiased estimate) could be indispensible. Unfortunately, a convergence proof for most instances of the UOBE algorithm is not forthcoming. The original OBE paper by Fogel and Huang [41] is sometimes misunderstood to indicate the convergence of the bounding ellipsoid to a point under ordinary conditions on $\varepsilon_*(\cdot)$. In fact, the paper only proves this convergence for the case of *unity weights* so that the fundamental optimization process is not taken into account. No known proof of this desirable result for the F-H OBE algorithm, or for any instance of UOBE with causal scaling exists, whether optimal or suboptimal checking is used. However, it can be shown for the volume algorithm with causal scaling (see Corollary 3 in Appendix A.1) that if an optimal weight exists at time n, then if the data set is included using this weight, then the volume will certainly decrease:

$$\mu_v(n) < \mu_v(n-1) \ . \tag{73}$$

This indicates that the ellipsoid volume will converge to some unspecified size in some unspecified manner. A similar result can be demonstrated for the trace algorithm [80].

In spite of this encouraging result, one of the drawbacks of the volume approach is that the set measure μ_v is not a proper "metric" in the parameter space. By this we mean the following: Suppose we propose the distance measure d such that at time n, $d(\Theta(n), \Theta_*) = \mu_v(n)$. We immediately find that d fails to be a proper metric since $d(\Theta(n), \Theta_*) = 0$ does not imply that $\Theta(n) = \Theta_*$. This unfortunate situation arises because the ellipsoid may potentially degenerate and reside in a subspace of $C^{m \times k}$, thereby achieving zero volume without being reduced to a point. According to Nayeri *et al.* [80], this will likely only occur if p.e. is not achieved, and is therefore a more important problem with colored disturbances. This potential anomaly provides motivation to consider the use of the trace measure for which a degenerate ellipsoid will not produce a zero set measure.

It has frequently been noted is that the hyperellipsoidal bounding sets resulting from UOBE algorithms can be quite "loose" supersets of the exact feasibility sets (polytopes) (e.g. [8],[83]), particularly in "finite"

[19] Please read the abbreviation "p.e." as "persitently exciting" or "persistency of excitation," as appropriate.

time[20]. However, many simulation studies in the literature (white noise case) have shown the volume of the ellipsoids to become quite small in the "long term." Further, as we and other researchers have demonstrated, the empirical convergence and tracking properties of the UOBE estimator are favorable in spite of the few data used. This is an indication that the *presence* of the ellipsoid and the optimization procedure centered on it, are quite useful for signal processing, regardless of our present inability to completely understand its behavior in theory. The results presented above offer further support for "good behavior" of this class of algorithms by indicating that the ellipsoid measures will converge to some unspecified size in some unspecified manner. This result has not been clearly understood, and its finding offers some hope that a proof of convergence (in some sense) for the UOBE algorithms may be found in the white noise case.

The D-H OBE algorithm [24] has been cited above as an instance of UOBE which *does* exhibit convergence of the ellipsoid under usual "white noise" excitation conditions. From the UOBE point of view, the trick employed by Dasgupta and Huang is to use a noncausal scaling sequence which is also pointwise optimized in a certain sense. In particular $\zeta(n) = (1 - \lambda_n^*(n))^{-1}$ for each n, so that the previous weights are also modified in a manner which is consistent with the optimal objective[21] of minimizing $\kappa(n)$. This choice of scaling sequence and minimization criterion admits the clever use of Lyapunov theory to obtain the convergence result. As we have also indicated above, the checking criterion for the D-H OBE method is of $\mathcal{O}(m)$ complexity per n, adding another attractive feature. These theroretical and computational benefits notwithstanding, in published simulation studies, this method has not been shown to exhibit any significant advantage in estimation or tracking with respect to the adaptive SM-WRLS methods, for example, discussed above. As we have also discussed above, the inability to determine the precise meaning of the optimization criterion leaves open some fundamental questions in the interpretation of the behavior of the method.

Finally, we note that some work with colored disturbances has been reported. The D-H OBE algorithm has been extended to the case of an ARMA (SISO) model by Rao *et al.* in [99]-[101]. In this case the error is filtered by a linear (MA) filter creating a colored noise sequence, say,

$$\varepsilon_*'(n) = \varepsilon_*(n) + \sum_{i=1}^{r} b_{i*}\varepsilon_*(n-i). \qquad (74)$$

Rao's approach is to estimate the unobservable sequence $\varepsilon_*(\cdot)$ by the errors $\varepsilon(n-i, \boldsymbol{\theta}(n-i))$, $i = 0, 1, \ldots, r$ at time n, then use the D-H OBE developments. Error bounds on $\varepsilon_*(\cdot)$ are not sufficient to bound $\varepsilon_*'(\cdot)$, so that the ellipsoid is no longer guaranteed to contain the true parameters. A condition on the b_{i*} parameters is determined such that this violation does not occur. Not surprisingly, the condition implies a restriction on the amount of *correlation which can be induced* by the MA filter.

[20] Norton has proposed the use of *inner* bounds as a possible remedy for this problem [83],[115]

[21] Recall that this is *not* the minimization of the ellipsoid size *per se*. See discussion in Sections 6.2.2 and 6.5.2.

Related work is found in the paper by Norton [84] in which the ARMAX model is studied. In this work, the effects of the coloring of the noise upon the bounds is studied both analytically and experimentally. The results indicate the possibility of non-convex bounds on the true feasible set, $\Omega(\cdot)$, in the colored noise case. An attempt is made to relate these anomalies to the bias that occurs in conventional WRLS processing due to the colored noise.

Nayeri *et al.* [80] have argued that the ellipsoid must remain nontrivial in the colored noise case (i.e. $\lim_{n\to\infty}\bar{\Omega}(n) \neq \{\boldsymbol{\Theta}_*\}$) and have conjectured that the $\boldsymbol{\Theta}_*$ will appear on the boundary of the limiting ellipsoid when $\epsilon_*(\cdot)$ is p.e. Some interesting effects of non-p.e. disturbances alluded to above are also studied in the cited paper.

6.7 Parallel Hardware Implementations

One of the advantages of the QR-WRLS-based UOBE formulation, and the feature which motivated its development [26], is that it immediately admits solution by contemporary parallel architectures. This is critical because it reduces the complexity of the *optimal* algorithm from $\mathcal{O}(m^2)$ to $\mathcal{O}(m)$, where m is the number of parameters to be estimated. The significant reduction of computational complexity and parallel hardware implementation of SM algorithms improve their potential for real time applications. Systolic *architectures for both nonadaptive* [31] *and adaptive* [87],[88] *versions of the* SM-WRLS algorithm have been developed by Odeh and Deller. The adaptive architecture has somewhat more complex cells, but the computational savings with respect to sequential solutions is identical. The complexity of the parallel computation is given by

$$f_{\substack{opt\\parallel}} \sim \mathcal{O}(3m) + \rho\mathcal{O}(11m) \text{ flops per } n \tag{75}$$

if the optimal checking is implemented, where ρ, as above, is the fraction of the data accepted by the SM considerations. If suboptimal checking is employed, the average count is

$$f_{\substack{subopt\\parallel}} \sim \mathcal{O}(m) + \rho'\mathcal{O}(11m) \text{ flops per } n. \tag{76}$$

where ρ' likewise indicates the acceptance ratio. When adaptation by back-rotation is added to either strategy, and additional $b\mathcal{O}(11m)$ (or $b'\mathcal{O}(11m)$) flops per n are required on the average, where b and b', as above, indicate the average number of back-rotations computed per n in the optimal and suboptimal cases. Note that these tallies represent *parallel* complexities in the sense that they denote the *effective* number of operations per n, though many processors can be performing this number of operations simultaneously. Accordingly the parallel complexity indicates the *time* it takes the parallel architecture to process the data regardless of the total number of operations performed by the individual cells.

Unlike the sequential algorithms, scaling may be added to the parallel processors (to implement the F-H OBE algorithm, for example) at virtually no computational cost, but at the negligible hardware cost of m multiplication units.

The parallel architectures described in papers cited above are developed for real, scalar observations, but can be used for *complex* scalar observations. The necessary modifications are concerned with the basic Givens rotation operations. These are elementary and are found, for example, in [48]. However, the general complex *vector* observation case of the SM-WRLS (UOBE) algorithm is not readily mapped into similar architectures. The generalized architecture that efficiently implements this case has not yet been developed.

7 Conclusions and Further Issues

The emerging field of SM-based signal processing is receiving considerable attention and is becoming increasingly popular around the world. In this paper, we have given a general review of SM theory and a broad coverage of general SM algorithms and related topics. The majority of this paper has been concerned with a class of SM algorithms for estimating the parameters of linear-in-parameter system or signal models in which the error sequence is pointwise "energy bounded." Specifically, we have focused on the case of ellipsoid algorithms which have been shown to represent a blending of the classical LSE methods with the BE constraints.

The combined LSE/BE algorithm has been formulated as a UOBE strategy which embraces all reported algorithms, adaptive and nonadaptive. Within this framework, a flexible strategy based on "back-rotation" has been proposed to make the UOBE algorithms specifically adaptive. The adaptive strategies as well as the nonadaptive cases performed well in simulation trials.

In general, SM approaches are interesting because they produce *sets* of feasible solutions based on tenable assumptions where no unique solution may otherwise exist. In the signal processing (LSE) domain, a unique solution exists, but the set provided by UOBE is interesting from two points of view. First, the feasible set may complement the unique LSE solution in cases in which the ordinary asumptions about the model error are tenuous (for example, where the model noise is colored). Secondly, from the feasible set arises an interesting data selection technique which can lead to significant computational complexity improvement. UOBE algorithms typically reject 70 - 95% of the incoming data sets because they fail to refine the existing ellipsoidal feasible set in some sense. This should not be misintepreted, however, to imply a 70 - 95% load improvement. In fact, certain constraints must be observed to achieve more than a gain of about five in complexity improvement with respect to conventional WRLS. If a sequential computing machine is to be used, then suboptimal checking (for feasible set refinement) must be used. A method suggested in this paper has been found to perform quite well, and yields $\mathcal{O}(m)$ complexity compared with (at least) $\mathcal{O}(m^2/2)$ for the optimal algorithm. This lowered complexity can be preserved with adaptive strategies which do not require excessive reiteration over the past data sets. Secondly, scaling factors ($\zeta(\cdot)$, see Section 4.2), including exponential forgetting factors, cannot be used except at $\mathcal{O}(m^2/2)$ expense. No compelling reason for the use of such factors has been observed in simulation studies in the literature. (A theoretical argument exists for the use of scaling factors in the D-H OBE algorithm [24]. In this case, the scaling strategy leads to convergence of the ellipsoidal bounding set in a certain sense.) Finally, a parallel processing version of the UOBE method has been presented with which to achieve $\mathcal{O}(m)$ complexity under virtually any condition of scaling, adaptation, optimal or suboptimal checking. Real applications of these identification techniques will benefit when these relatively simple architectures can be dedicated to the process. It is interesting to note that the infrequent updating which results as a consequence of the UOBE considerations, may lead to strategies for time-sharing of these parallel processors.

The simulation results presented illustrate important points about the various UOBE methods and show that the adaptive algorithms yield accurate estimates using very few of the data and quickly adapt to fast variations in the signals dynamics.

Some of the key theoretical results underlying the UOBE class of algorithms appear in the appendices of this paper. These appendices unify many theoretical results found in the literature.

Many interesting open research problems remain in SM-based signal processing. Among them are the pursuit of different adaptation strategies, refined hardware solutions, and a world of other challenges that will emerge as these exciting new techniques continue to be applied to practical problems. As computing power continues to increase, many of the more complex error bounding, and other SM, algorithms will begin to attract more attention of signal processing engineers. In this sense, the techniques upon which this paper has focused may ultimately comprise a very small part of the overall impact which SM-based techniques will have on the signal processing technologies.

References

[1] B.I. Anan'ev and A.B. Kurzhanski, "The nonlinear filtering problem for a multistage system with statistical uncertainty," *Proc. 2nd IFAC Symposium*, pp. 19-23, 1986.

[2] L.V.R. Arruda and G. Favier, "A review and a comparison of robust estimation methods," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 1027-1032, Budapest, July 1991.

[3] Cs. Bányász and L. Keviczky (editors), *Proceedings of the 9th IFAC / IFORS Symposium on Identification and System Parameter Estimation*, vols. 1 & 2, Budapest, July 1991.

[4] B.R. Barmish and J. Sankaran, "The propagation of parametric uncertainty via polytopes," *IEEE Trans. Automatic Control*, vol. AC-24, pp. 346-349, 1979.

[5] G. Belforte and B. Bona, "An improved parameter identification algorithm for signals with unknown-but-bounded errors," *Proc. 7th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, York, pp. 1507-1512, 1985.

[6] G. Belforte and M. Milanese, "Uncertainty interval evaluation in presence of unknown-but-bounded errors: Nonlinear families of models," *Proc. 1st IASTED Int. Symp. Modeling, Identification, and Control*, Davos, pp. 75-79, 1981.

[7] G. Belforte and T.T. Tay, "Recursive estimation for linear models with set-membership measurement error," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 872-877, Budapest, July 1991.

[8] G. Belforte, B. Bona, and V. Cerone, "Identification, structure selection and validation of uncertain models with set-membership error description," *Mathematics and Computers in Simulation*, vol. 32, pp. 561-569, 1990.

[9] G. Belforte, B. Bona, and V. Cerone, "Parameter estimation algorithms for a set-membership description of uncertainty," *Automatica*, vol. 26, pp. 887-898, 1990.

[10] G. Belforte, B. Bona, and V. Cerone, "Parameter estimation with set membership uncertainty: Nonlinear families of models," *Proc. 8th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, Beijing, pp. 399-404, 1988.

[11] G. Belforte, B. Bona, and V. Cerone, "A bounded error approach to the tuning of a digital voltmeter," *Proc. 12th IMACS World Congress*, Paris, 1988.

[12] G. Belforte, B. Bona, and S. Frediani, "Optimal sampling schedule for parameter estimation of linear models with unknown-but-bounded measurements errors," *IEEE Trans. Automatic Control*, vol. AC-32, pp. 179-182, 1987.

[13] G. Belforte, B. Bona, E. Canuto, F. Donati, F. Feraris, I. Gorini, S. Morei, M. Peisino, and S. Sartori, "Coordinate measuring machine and machine tool self calibration and error correction," *Annals CIRP*, vol. 36, pp. 359-364, 1987.

[14] G. Belforte, B. Bona, and M. Milanese, "Advanced modelling and identification techniques for metabolic processes," *CRC J. Biomed. Engineer.*, vol. 10, pp. 275-316, 1983.

[15] D.P. Bertsekas and I.B. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Trans. Automatic Control*, vol. AC-16, pp. 117-128, 1971.

[16] V. Broman and M.J. Shensa, "A compact algorithm for the intersection and approximation of N-dimensional polytopes," *Mathematics and Computers in Simulation*, vol. 32, pp. 469-480, 1990.

[17] V. Broman and M.J. Shensa, "Polytopes, a novel approach to tracking," *Proc. 25th IEEE Conf. Decision and Control*, Athens, pp. 1749-1752, 1986.

[18] T. Clément and S. Gentil, "Recursive membership set estimation for output-error models," *Mathematics and Computers in Simulation*, vol. 32, pp. 505-513, 1990.

[19] T. Clément and S. Gentil, "Reformulation of parameter identification with unknown-but-bounded errors," *Mathematics and Computers in Simulation*, vol. 30, pp. 257-270, 1988.

[20] P.L. Combettes, *Set Theoretic Estimation in Digital Signal Processing* (Ph.D. Dissertation), North Carolina State University, Raleigh, 1989.

[21] P.L. Combettes and H.J. Trussell, "The use of noise properties in set-theoretic estimation," *IEEE Trans. Signal Process.*, vol. 39, pp. 1630–1641, 1991.

[22] P.L. Combettes and H.J. Trussell, "General order moments in set-theoretic estimation," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '89*, Glasgow, vol. 4, pp. 2101-2104, May 1989.

[23] P.L. Combettes and H.J. Trussell, "Stability of the linear prediction filter: A set theoretic approach," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '88*, New York, vol. 4, pp. 2288-2291, 1988.

[24] 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 Trans. Information Theory*, vol. IT-33, pp. 383-392, 1987.

[25] J.R. Deller, Jr., "Toward the use of set-membership identification in efficient training of feedforward neural nets," *Proc. 1990 IEEE Int. Symp. Circuits and Systems*, New Orleans, vol. 3, pp. 2041-2044, 1990.

[26] J.R. Deller, Jr., "A 'systolic array' formulation of the optimal bounding ellipsoid algorithm," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 1432-1436, Sept. 1989.

[27] J.R. Deller, Jr., "Set-membership identification in digital signal processing," *IEEE ASSP Magazine*, vol. 6, pp. 4-22, Oct. 1989.

[28] J.R. Deller, Jr. and D. Hsu, "An alternative adaptive sequential regression algorithm and its application to the recognition of cerebral palsy speech," IEEE Trans. Circuits and Systems, vol. CAS-34, pp. 782-787, July 1987.

[29] J.R. Deller, Jr. and T.C. Luk, "Linear prediction analysis of speech based on set-membership theory," *Computer Speech and Language*, vol. 3, pp. 301-327, Oct. 1989.

[30] J.R. Deller, Jr. and T.C. Luk, "Set-membership theory applied to linear prediction analysis of speech," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '87*, Dallas, vol. 2, pp. 653-656, 1987.

[31] J.R. Deller, Jr. and S.F. Odeh, "Implementing the optimal bounding ellipsoid algorithm on a fast processor," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '89*, Glasgow, vol. 2, pp. 1067-1070, May 1989.

[32] J.R. Deller, Jr. and S.F. Odeh, "SM-WRLS algorithms with an efficient test for innovation," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 1044-1049, Budapest, July 1991.

[33] J.R. Deller, Jr. and G.P. Picaché, "Advantages of a Givens rotation approach to temporally recursive linear prediction analysis of speech," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 429-431, March 1989.

[34] J.G. Ecker, "Geometric programming: Methods, computations, and applications," *SIAM Rev.*, vol. 22, pp. 338-362, 1980.

[35] J. Falk, *Global solutions of signomial programs*, Technical Report T-274, Geoige Washington University, Washington, DC, 1973.

[36] K. Fedra, G. van Straten, and M.B. Beck, "Uncertainty and arbitrariness in ecosystems modeling: A lake modeling example," *Ecological Modeling*, vol. 13, pp. 87-110, 1981.

[37] M.L. Feron, "Membership set system identification with pulse train input" (M.S. Dissertation), Illinois Institute of Technology, Chicago, 1982.

[38] M.L. Feron and J.R. Deller, Jr., "'Membership set' identification applied to the impulse train excited AR model" (abstract), *Final Program: SIAM 1983 National Mtg.*, Denver, p. 30, 1983.

[39] G. Fiorio, S. Malan, M. Milanese, and A. Vicino, "Guaranteed specification control design of inertial platforms for a high accuracy calibration device," *Proc. 29th IEEE Conf. on Decision and Control*, Honolulu, 1990.

[40] E. Fogel, "System identification via membership set constraints with energy constrained noise," *IEEE Trans. Automatic Control*, vol. AC-24, pp. 752-758, 1979.

[41] E. Fogel and Y.F. Huang, "On the value of information in system identification - Bounded noise case," *Automatica*, vol. 18, pp. 229-238, 1982.

[42] W.M. Gentleman, "Least squares computations by Givens transformations without square roots," *J. Inst. of Math. and its Appl.*, Vol. 12, pp. 329-336, 1973.

[43] W.M. Gentleman and H.T. Kung, "Matrix triangularization by systolic arrays," *Proc. Soc. Photo-Optical Instrumentation Engineers: Real-Time Signal Processing IV*, vol. 298, pp. 19-26, 1981.

[44] R. Gomeni, H. Piet-Lahanier, and E. Walter, "Study of the phamacokinetics of betaxolol using set-membership set estimation," *Proc. 3rd IMEKO Congress on Measurements in Clinical Medicine*, Edinburgh, 1986.

[45] D. Graupe, *Time Series Analysis, Identification, and Adaptive Systems*, Malabar, Florida: Krieger, Ch. 5, 1989.

[46] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Baltimore, MD: Johns-Hopkins Univ. Press, Ch. 6, 1983.

[47] G.C. Goodwin and K.S. Sin, *Adaptive Filtering, Prediction, and Control*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[48] S. Haykin, *Adaptive Filter Theory*, Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

[49] Y.F. Huang, "A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-34, pp. 1331-1334, Oct. 1986.

[50] S.C. Huang and Y.F. Huang, "Learning algorithms for perceptrons using back-propagation with selective updates," *IEEE Control Systems Magazine*, pp. 56-61, Apr. 1990.

[51] Y.-F. Huang and A.K. Rao, "Application of a recursive estimation algorithm with information-dependent updating to ARMAX models and ARMA models with unknown inputs," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '87*, Dallas, vol. 2, pp. 1007-1010, 1987.

[52] C.R. Johnson, *Lectures on Adaptive Parameter Estimation*, Englewood Cliffs, New Jersey: Prentice-Hall, 1988.

[53] W. Kahan, "Circumscribing an ellipsoid about the intersection of two ellipsoids," *Canadian Math. Bull.*, vol. 11, p. 437, 1968.

[54] I.Ya. Kats and A.B. Kurzhanski, "Minimaz multistep filtering in statistically indeterminate solutions," *Automat. i Telemekh.*, vol. 11, p. 79, 1978.

[55] K. Keesman, "Robust identification and prediction for output-error models," *Proceedings of the* 9^{th} *IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 878-882, Budapest, July 1991.

[56] K. Keesman, "Membership-set estimation using random scanning and principal component analysis," *Mathematics and Computers in Simulation*, vol. 32, pp. 535-543, 1990.

[57] K. Keesman, *A Set-Membership Approach to the Identification and Prediction of Ill-Defined Systems: Application to a Water Quality System* (Ph.D. Dissertation), University of Twente, 1989.

[58] K.J. Keesman and G. Van Straten, "Modified set-theoretic identification of an ill-defined water quality system from poor data," *Proc. of IAWPRC Symp. on System Analysis in Water Quality Management*, 1987.

[59] H. König and D. Pallaschke, "On Khachian's algorithm and minimal ellipsoids," *Numer. Math.*, vol. 36, pp. 211-223, 1981.

[60] R.L. Kosut, "Adaptive robust control via parameter set estimation," *Int. J. Adaptive Control and Signal Process.*, vol. 2, pp. 371-399, 1988.

[61] N.N. Krasnovski, "On the theory of controllability and observability of linear dynamic systems," *Prikl. Mat. Mech.*, vol. 28, no. 1, 1964.

[62] J.M. Krause, G. Stein, and P.P. Khargonekar, "Robust performance of adaptive controllers with generally uncertain structure," *Proc.* 29^{th} *IEEE Conf. on Decision and Control*, Honolulu, 1990.

[63] A.J. Krener, "Kalman–Bucy and mimimax filtering," *IEEE Trans. Automatic Control*, vol. 25, pp. 291-292, 1980.

[64] S.Y. Kung, *VLSI Array Processors*, Englewood Cliffs, NJ: Prentice Hall, Ch. 3, 1988.

[65] H.T. Kung and C. Leiserson, "Algorithms for VLSI processor arrays," Rpt. No. MU-CS-79-103, Dept. of Computer Sci., Carnegie-Mellon Univ., 1978. Reprinted in C. Mead and L. Conway, *Introduction to VLSI Systems*, Reading, MA: Addison-Wesley, Section 8.3, 1980.

[66] A.B. Kurzhanski and I. Vályi, "Guaranteed state estimation for dynamic systems: Beyond the overviews," *Proceedings of the* 9^{th} *IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 1033-1037, Budapest, July 1991.

[67] M.K. Lau, R.L. Kosut, and S. Boyd, "Parameter set estimation of systems with uncertain nonparametric dynamics and disturbances," *Proc.* 29^{th} *IEEE Conf. on Decision and Control*, Honolulu, 1990.

[68] M.S. Liu, M. Nayeri, and J.R. Deller, Unpublished research notes, Michigan State University, 1991.

[69] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, Cambridge, MA: MIT Press, pp. 14-15 & Sect. 2.2.1., 1983.

[70] T.C. Luk, *Fast Optimal Identification of Autoregressive Systems with a Class of Correlated Inputs* (Ph.D. Dissertation), Illinois Institute of Technology, Chicago, 1985.

[71] T.C. Luk and J.R. Deller, Jr., "A nonclassical WRLS algorithm," *Proc. 23rd Ann. Allerton Conf.*, pp. 732-741, 1985.

[72] J.G. McWhirter, "Recursive least squares minimization using a systolic array," *Proc. Soc. Photo-Optical Instrumentation Engineers: Real-Time Signal Processing VI*, vol. 431, pp. 105-112, 1983.

[73] M. Milanese and G. Belaforte, "Estimation theory and uncertainty intervals evaluation in the presence of unknown but bounded errors: Linear families of models and estimators," *IEEE Trans. Automatic Control*, vol. AC-27, pp. 408-414, 1982.

[74] M. Milanese and A. Vicino, "Robust estimation and exact uncertainty intervals evaluation for nonlinear models," in: S. Tzafestas *et al.*, eds., *System Modeling and Simulation*, Amsterdam: North-Holland, pp. 91-96, 1989.

[75] M. Milanese and A. Vicino, "Estimation theory for dynamic systems with unknown but bounded uncertainty: An overview," *Proceedings of the* 9^{th} *IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 859-867, Budapest, July 1991.

[76] M. Milanese, R. Tempo, and A. Vicino, "Optimal error prediction for economic models," *Int. J System Sci.*, vol. 19, pp. 1189-1200, 1988. with

[77] S.H. Mo, *Computation of Bounded-parameter Models of Dynamical Systems from Bounded-noise Records* (Ph.D. Dissertation), University of Birmingham, Birmingham, England 1989.

[78] S.H. Mo and J.P. Norton, "Fast and robust algorithm to compute exact polytope parameter bounds," *Mathematics and Computers in Simulation*, vol. 32, pp. 481-493, 1990.

[79] D.R. Morrell and W.C. Stirling, "Set-valued filtering and smoothing," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 21, pp. 184-193, 1991.

[80] M. Nayeri, J.R. Deller, Jr., and M.M. Krunz, "Convergence and colored noise issues in bounding ellipsoid identification," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '92*, San Francisco (in press).

[81] J.P. Norton, *An Introduction to Identification*, London and Orlando, Florida: Academic Press. 1986.

[82] J.P. Norton, "Problems in identifying the dynamics of biological systems from very short records," *Proc. 25th IEEE Conf. on Decision and Control*, Athens, 1986.

[83] J.P. Norton, "Recursive computation of inner bounds for the parameters of linear models," *Int. J. Control*, vol. 50, pp. 2423-2430, 1989.

[84] J.P. Norton, "Identification of parameter bounds of ARMAX models from records with bounded noise," *Int. J. Control*, vol. 45, pp. 375-390, 1987.

[85] J.P. Norton, "Identification and application of bounded-parameter models," *Automatica*, vol. 23, pp. 497-507, 1987.

[86] J.P. Norton and S.H. Mo, "Parameter bounding for time-varying systems," *Mathematics and Computers in Simulation*, vol. 32, pp. 527-534, 1990.

[87] S.F. Odeh, *Algorithms and Architectures for Adaptive Set Membership-based Signal Processing* (Ph.D. Dissertation), Michigan State University, East Lansing, 1990.

[88] S.F. Odeh and J.R. Deller, Jr., "Systolic algorithms for adaptive set-membership identification," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '90*, Albuquerque, vol. 5, pp. 2419-2422, Apr. 1990.

[89] S.F. Odeh and J.R. Deller, Jr., "An SM-WRLS algorithm with an efficient test for innovation: Simulation studies and complexity issues," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '91*, Toronto, vol. 5, pp. 3413-3416, May 1991.

[90] H. Piet-Lahanier and E. Walter, "Characterization of non-connected parameter uncertainty regions," *Mathematics and Computers in Simulation*, vol. 32, pp. 553-560, 1990.

[91] H. Piet-Lahanier and E. Walter, "Further results on recursive polyhedral description of parameter uncertainty in the bounded-error context," *Proc. 28th IEEE Conf. Decision and Control*, Tampa, pp. 1964-1966, 1989.

[92] H. Piet-Lahanier and E. Walter, "Practical implementation of an exact and recursive algorithm for characterizing likelihood sets," *Proc. 12th IMACS World Congress on Scientific Computation*, Paris, July 1988.

[93] H. Piet-Lahanier, E. Walter, and R. Gomeni, "OMNE: a new robust membership set estimator for the parameters of nonlinear models," *J. Pharmacokin. Biopharm.*, vol. 15, pp. 203-219, 1987.

[94] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge, Massachusetts: Cambridge Uviversity Press, 1988.

[95] L. Pronzato and E. Walter, "Experiment design for bounded-error models," *Mathematics and Computers in Simulation*, vol. 32, pp. 571-584, 1990.

[96] L. Pronzato, E. Walter, and H. Piet-Lahanier, "Mathematical equivalence of two ellipsoidal algorithms for bounded-error estimation," *Proc. 28th IEEE Conf. Decision and Control*, Tampa, pp. 1952-1955, 1989.

[97] L. Pronzato, E. Walter, A. Venot, and J.-F. Lebruchec, "A general-purpose global optimizer: Implementation and applications," *Mathematics and Computers in Simulation*, vol. 26, pp. 412-422, 1984.

[98] B.N. Pshenichnyi and V.G. Pokotilo, "On observation problems in discrete systems," *PMM J. Appl. Math. Mechanics*, vol. 45, pp. 1-6, 1981.

[99] A.K. Rao and Y.F. Huang, "Recent developments in optimal bounding ellipsoid parameter estimation, *Mathematics and Computers in Simulation*, vol. 32, pp. 515-526, Dec. 1990.

[100] A.K. Rao and Y.F. Huang, "Analysis of finite precision effects on an OBE algorithm," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '89*, Glasgow, vol. 2, pp. 853-856, May 1989.

[101] A.K. Rao, Y.-F. Huang, and S. Dasgupta, "ARMA parameter estimation using a novel recursive estimation algorithm with selective updating," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 38, pp. 447-457, March 1990.

[102] F.C. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Trans. Automatic Control*, vol. AC-13, pp. 22-28, 1968.

[103] N. Shor and V. Gershovich, "Family of algorithms for solving convex programming problems," *Cybernetics*, vol. 15, pp. 502-507, 1979.

[104] M.K. Smit, "A novel approach to the solution of indirect measurement problems with minimal error propagation," *Measurement*, vol. 1, pp. 181-190, 1983.

[105] M.K. Smit and Ch. van Vliet, "OMTP: Fortran program for optimally solving indirect measurement problems," *Measurement*, vol. 1, pp. 209-211, 1983.

[106] R. Tempo, "Robust estimation and filtering in the presence of bounded noise," *IEEE Trans. Automatic Control*, vol. AC-33, pp. 864-867, 1988.

[107] R. Tempo and A. Vicino, "Optimal algorithms for system identification: A review of some recent results," *Mathematics and Computers in Simulation*, vol. 32, pp. 585-595, 1990.

[108] H.J. Trussell and M.R. Civanlar, "The feasible solution in signal restoration," *IEEE Trans. Acoustics, Speech, and Signal Process.*, vol. 32, pp. 201-212, 1984.

[109] G. Van Straten, "Analytical methods for parameter space delimitation and application to shallow-lake phytoplankton dynamics modelling," *Appl. Math. and Computing*, vol. 17, pp. 459-482, 1985.

[110] J.W. Verhoof and M.K. Smit, "Error analysis for ellipsometry," *Proc. 12th IMACS World Congress*, Paris, 1988.

[111] A. Venot, L. Pronzato, E. Walter, and J.F. Lebruchec, "A distribution-free criterion for robust identification with applications in system modelling and image processing," *Automatica*, vol. 22, pp. 105-109, 1986.

[112] S.M. Veres and J.P. Norton, "Bound-based worst-case self-tuning controllers," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 1, pp. 773-778, Budapest, July 1991.

[113] S.M. Veres and J.P. Norton, "Parameter bounding algorithms for linear errors in variables models," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 1038-1043, Budapest, July 1991.

[114] S.M. Veres and J.P. Norton, "Structure identification of parameter-bounding models by use of noise-structure bounds," *Int. J. Control*, vol. 50, pp. 639-649, 1989.

[115] A. Vincino and M. Milanese, "Optimal approximation of feasible parameter set in estimation with unknown but bounded noise," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 1365-1369, Budapest, July 1991.

[116] A. Vicino and M. Milanese, "Optimal inner bounds of feasible parameter sets in linear estimation with bounded noise," *Proc. 28th IEEE Conf. Decision and Control*, Tampa, pp. 2576-2580, 1989.

[117] A. Vicino, A. Tesi, and M. Milanese, "Computation of non-conservative stability perturbation bounds for systems with nonlinearly correlated uncertainties," *IEEE Trans. Automatic Control*, vol. 35, pp. 835-841, 1990.

[118] A. Vicino, R. Tempo, R. Genesio, and M. Milanese, "Optimal error and GMDH predictors: A comparison with some statistical techniques," *Int. J. Forecasting*, vol. 2, pp. 313-328, 1987.

[119] E. Walter (editor), *Mathematics and Computers in Simulation*, Special issue on parameter identification with error bound, vol. 32, Dec. 1990.

[120] E. Walter and H. Piet-Lahanier, "Recursive robust minimax estimation for models linear in their parameters," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 1, pp. 763-768, Budapest, July 1991.

[121] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: A survey," *Mathematics and Computers in Simulation*, vol. 32, pp. 449-468, 1990.

[122] E. Walter, and H. Piet-Lahanier, "Exact recursive polyhedral description of the feasible parameter set for bounded-error models," *IEEE Trans. Automatic Control*, vol. AC-34, pp. 911-915, 1989.

[123] E. Walter, and H. Piet-Lahanier, "Robust linear and nonlinear parameter estimation in the bounded-error context," in: M. Milanese, R. Tempo, and A. Vicino, eds., *Robustness in Identification and Control*, New York, NY: Plenum, pp. 67-76, 1989.

[124] E. Walter and H. Piet-Lahanier, "Bounded-error alternative to least squares for estimating the parameters of a nonlinear model from short records," in: J. Eisenfeld and D. Levine, eds., *Biomedical Modeling and Simulation*, pp. 95-99, 1989.

[125] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: A survey," *Proc. 12th IMACS World Congress on Scientific Computation*, Paris, pp. 467-472, July 1988.

[126] E. Walter and H. Piet-Lahanier, "OMNE versus least squares for estimating parameters of a biological model from short data records," *Proc. 12th IMACS World Congress on Scientific Computation*, Paris, pp. 85-87, July 1988.

[127] E. Walter and H. Piet-Lahanier, "Exact and recursive description of the feasible parameter set for bounded error models," *Proc. 26th IEEE Conf. Decision and Control*, Los Angeles, pp. 1921-1922, 1987.

[128] E. Walter and H. Piet-Lahanier, "Robust nonlinear estimation in the bounded-noise case," *Proc. 25th IEEE Conf. Decision and Control*, Athens, pp. 1037-1042, 1986.

[129] E. Walter, H. Piet-Lahanier, and J. Happel, "Estimation of non-uniquely identifiable parameters via exhaustive modeling and membership set theory," *Mathematics and Computers in Simulation*, vol. 28, pp. 479-490, 1986.

[130] E. Walter, Y. Lecourtier, J. Happel, and J.Y. Kao, "Identifiability and distinguishability of fundamental parameters in catalytic methanation," *J. American Inst. Chemical Engineer.*, vol. 32, pp. 1360-1366, 1988.

[131] H.S. Witsenhausen, "Sets of possible states of linear systems given perturbed observations," *IEEE Trans. Automatic Control*, vol. AC-13, pp. 556-568, 1968.

A Appendices

These appendices present results which will rigorously support informal arguments made in the main text. For generality, we will include a scaling sequence (of the form (22) unless otherwise noted) in the WRLS recursions. "Unscaled" results are obtained by simply dropping subscripts "s" or setting $\zeta(n) = 1$ wherever it occurs. Without loss of generality, however, we shall not explicitly include the downdating process for adaptation which was developed in the paper. If the solution at time $n - 1$ is to be downdated prior to consideration of $(\boldsymbol{y}(n), \boldsymbol{x}(n))$, then all quantities implicitly or explicitly involving past data will be modified, and then will enter the developments in precisely the same way as their "un-downdated" counterparts. In this case, for example, every occurence of $G_s(n) = \boldsymbol{x}^T(n)\boldsymbol{C}_s^{-1}(n-1)\boldsymbol{x}(n)$ should be replaced by $G_{d,s}(n) = \boldsymbol{x}^T(n)\boldsymbol{C}_{d,s}^{-1}(n-1)\boldsymbol{x}(n)$.

A.1 Propositions and Corrollaries

Proposition 1 *Let $\Omega(n) \subseteq \mathcal{C}^{mk \times k}$ be the feasibility set arising from BE constraints as in (29). Given observations on time range $t \in [1, n]$, let $\boldsymbol{\Theta}(n)$ denote the weighted LSE estimate with associated covariance matrix $\boldsymbol{C}(n)$. The weights used in the estimation are $\lambda_n(t)$ with $\lambda_n(1) > 0$. There exists a hyperellipsoidal set of parameter vectors, $\bar{\Omega}(n) \subseteq \mathcal{C}^{mk \times k}$, such that $\boldsymbol{\Theta}_* \in \Omega(n) \subseteq \bar{\Omega}(n)$, which is given by*

$$\bar{\Omega}(n) = \left\{\boldsymbol{\Theta} \mid tr\left\{[\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]^H \frac{\boldsymbol{C}(n)}{\kappa(n)} [\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]\right\} < 1\right\}, \quad \boldsymbol{\Theta} \in \mathcal{C}^{mk \times k} \tag{77}$$

where,

$$\kappa(n) = tr\left\{\boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta}(n)\right\} + \sum_{t=1}^{n} \lambda_n(t)\gamma(t)(1 - \gamma^{-1}(t) \parallel \boldsymbol{y}(t) \parallel^2). \tag{78}$$

Remark: When $k = 1$ this the result of Proposition 1 reduces to a generalization of the MISO case result found in many papers in the literature. When $k \neq 1$, a hyperelliposidal bounding set is also associated with each scalar component of the output vector as we show in the following corollary.

Corollary 1 *Under the conditions of Proposition 1, feasible parameter vectors associated with output y_i (column i of $\boldsymbol{\Theta}$), say $\boldsymbol{\theta}_i$, are confined to a hyperellipsoidal membership set, say $\bar{\Omega}_i(n)$, which is centered on its current weighted LSE estimate, $\boldsymbol{\theta}_i(n)$.*

$$\Omega_i(n) = \left\{\boldsymbol{\theta}_i \mid [\boldsymbol{\theta}_i - \boldsymbol{\theta}_i(n)]^H \frac{\boldsymbol{C}(n)}{\kappa(n)} [\boldsymbol{\theta}_i - \boldsymbol{\theta}_i(n)] < 1\right\}, \quad \boldsymbol{\theta}_i \in \mathcal{C}^{mk}. \tag{79}$$

Remark: This means simply that there is a hyperellipsoidal domain in the parameter subspace which contains all possible parameter vectors and which is centered on the WRLS estimate. Note that the ellipsoid associated with each y_i, $i = 1, 2, \ldots, k$, is identical to all others except for its center.

Proposition 2 *If it exists, the weight $\lambda_n^*(n)$ which minimizes*

1. *the volume measure $\mu_v(n)$ is the unique positive root of the quadratic equation*

$$F_v(\lambda) = a_2\lambda^2 + a_1\lambda + a_0 = 0 \tag{80}$$

where, $a_2 = \{(mk-1)\gamma(n)G_s^2(n)\}$,

$a_1 = \{(2mk-1) + \gamma^{-1}(n) \parallel \varepsilon(n, \Theta(n-1)) \parallel^2 - \kappa_s(n-1)\gamma^{-1}(n)G_s(n)\}\,\gamma(n)G_s(n)$,

and $a_0 = mk\left[\gamma(n) - \parallel \varepsilon(n, \Theta(n-1)) \parallel^2\right] - \kappa_s(n-1)G_s(n)$;

2. *the trace measure $\mu_t(n)$ is the unique positive root of the cubic equation*

$$F_t(\lambda) = b_3\lambda^3 + b_2\lambda^2 + b_1\lambda + b_0 \tag{81}$$

with $b_3 = \gamma(n)G_s^2(n)(G_s(n) - I_s(n-1)H_s(n))$,

$b_2 = 3\gamma(n)G_s(n)[G_s(n) - I_s(n-1)H_s(n)]$,

$b_1 = H_s(n)G_s(n)I_s(n-1)\kappa_s(n-1) - 2H_s(n)I_s(n-1)\left[\gamma(n) - \parallel \varepsilon(n, \Theta(n-1)) \parallel^2\right]$
$- G_s(n) \parallel \varepsilon(n, \Theta(n-1)) \parallel^2 + 3\gamma(n)G_s(n)$,

and $b_0 = \gamma(n) - \parallel \varepsilon(n, \Theta(n-1)) \parallel^2 - H_s(n)I_s(n-1)\kappa_s(n-1)$,

where $G_s(n) \overset{\text{def}}{=} \boldsymbol{x}^T(n)\boldsymbol{C}_s^{-1}(n-1)\boldsymbol{x}(n)$, $H_s(n) \overset{\text{def}}{=} \boldsymbol{x}^T(n)\boldsymbol{C}_s^{-2}(n-2)\boldsymbol{x}(n)$, *and* $I_s(n) \overset{\text{def}}{=} tr\,\boldsymbol{C}_s(n)$.

Remark: Many of the inherent scale factors in the coefficients above cancel, but for practical implementation it is more useful to express the coefficients with the scaled quantities included. By cancelling the scale factors, however, the following can immediately be observed for either optimization criterion: If $\lambda_{n,s}^*(n)$ denotes the optimal weight (or, in fact, any root of the polynomial) at time n with scaling, while $\lambda_n^*(n)$ denotes the weight resulting if no scaling takes place, then $\lambda_{n,s}^*(n) = \lambda_n^*(n)\zeta(n)$.

Corollary 2 *There is an inherent hyperellipsoid associated with the D-H OBE algorithm whose volume at time n would be minimized by the positive root of the quadratic $F_v'(\lambda) = a_2'\lambda^2 + a_1'\lambda + a_0'$ where $a_2' = a_2 + a_0 - a_1$, $a_1' = a_1 - 2a_0$, and $a_0' = a_0$, where a_i, $i = 0, 1, 2$ are defined as in Proposition 2.*

Remarks:

1. Interestingly, the quadratic in Corollary 2 can be obtained by using the scale factor $\zeta(n) = (1 - \lambda_n^*(n))^{-1}$ in the results of Proposition 2. That this should be true is not obvious because of the nonlinearities in λ which are created by these scale factors.

2. A similar result likely obtains for the trace case.

3. The utility of this result remains an open question because to use weights which are optimal in this sense does not necessarily admit the convergence results obtained by the Dasgupta-Huang analysis.

Corollary 3 *Consider the UOBE algorithm with simple scale factors and volume optimization. If an optimal weight exists at time n, then its use will certainly diminish the volume,* $\mu_v(n) < \mu_v(n-1)$.

Remark: A similar result can be obtained for the trace measure [80].

Proposition 3 $\kappa(n, \lambda_n(n))$ *has the following properties: 1. On the interval* $\lambda_n(n) \in (0, \infty)$, $\kappa(n, \lambda_n(n))$ *is either monotonically increasing or it has a single minimum. 2.* $\kappa(n, \lambda_n(n))$ *has a minimum on* $\lambda_n(n) \in (0, \infty)$ *iff*

$$\| \varepsilon(n, \boldsymbol{\Theta}(n-1)) \|^2 > \gamma(n). \tag{82}$$

Proposition 4 *Consider the UOBE algorithm with causally scaled weights as in (22). Then, if suboptimal check (82) holds, a positive* <u>*optimal*</u> *weight exists for either the volume or trace algorithm.*

Remark: The D-H OBE [24] algorithm uses a similar test (see (71)), derived by very different arguments. See the discussion in Section 6.5.2 for interesting similarities between Proposition 4 and D-H OBE.

A.2 Lemmas

Lemma 1 *Condition (29) implies*

$$\sum_{t=1}^{n} \lambda_n(t) \| \varepsilon_*(t) \|^2 \leq \sum_{t=1}^{n} \lambda_n(t)\gamma(t) \tag{83}$$

for any non-negative (real) sequence $\lambda_n(\cdot)$. *The equality can be removed for* $n \geq t_0$, *where* t_0 *is the minimum* t *for which* $\lambda_n(t) \neq 0$.

Lemma 2 *The scalar sequence* $\kappa(\cdot)$ *of (78) can be computed recursively in two ways:*

1. *In the context of MIL-WRLS:*

$$\kappa(n) = \kappa_s(n-1) + \lambda_n(n)\gamma(n) - \frac{\lambda_n(n) \| \varepsilon(n, \boldsymbol{\Theta}(n-1)) \|^2}{1 + \lambda_n(n) G_s(n)} \tag{84}$$

with $\kappa_s(n-1) \stackrel{\text{def}}{=} \kappa(n-1)/\zeta(n-1)$ *and* $\kappa_s(0) \stackrel{\text{def}}{=} 0$.

2. *In the context of QR-WRLS: Let* $\boldsymbol{T}(n)\boldsymbol{\Theta}(n) = \boldsymbol{D}_1$ *represent the triangular system of equations to be solved at time n and let* $\boldsymbol{\delta}_1(n) \stackrel{\text{def}}{=} tr\ \{\boldsymbol{D}_1^H(n)\boldsymbol{D}_1(n)\}$. *Then,*

$$\kappa(n) = \boldsymbol{\delta}_1(n) + \tilde{\kappa}(n) \tag{85}$$

with

$$\tilde{\kappa}(n) = \tilde{\kappa}_s(n-1) + \lambda_n(n)\gamma(n)\left(1 - \gamma^{-1}(n) \| \boldsymbol{y}(n) \|^2\right) \tag{86}$$

where $\tilde{\kappa}_s(n-1)$ *is defined as above.*

A.3 Proofs

Proof of Lemma 1: That the *equality* holds for $n < t_0$ is obvious. At t_0

$$\lambda_n(t_0) \parallel \boldsymbol{\varepsilon}_*(t_0) \parallel^2 < \lambda_n(t_0)\gamma(t_0). \tag{87}$$

Since $\lambda_n(t) = 0$ for $t < t_0$, (87) may be written

$$\sum_{t=1}^{t_0} \lambda_n(t) \parallel \boldsymbol{\varepsilon}_*(t) \parallel^2 < \sum_{t=1}^{t_0} \lambda_n(t)\gamma(t) \ . \tag{88}$$

Sequentially add inequalities $\lambda_n(t) \parallel \boldsymbol{\varepsilon}_*(t) \parallel^2 < \lambda_n(t)\gamma(t)$ for $t = t_0 + 1, \ldots, n$, noting that the inequality between the sums is preserved. □

Proofs of Proposition 1 and Corollary 1: Upon writing $\parallel \boldsymbol{\varepsilon}_*(t)) \parallel^2$ as $\operatorname{tr}\left\{\boldsymbol{\varepsilon}_*(t)\boldsymbol{\varepsilon}_*^H(t)\right\}$, it follows immediately from Lemma 1 that

$$\sum_{t=1}^{n} \lambda_n(t) \operatorname{tr}\left\{\left[\boldsymbol{y}(t) - \boldsymbol{\Theta}_*{}^H\boldsymbol{x}(t)\right]\left[\boldsymbol{y}(t) - \boldsymbol{\Theta}_*^H\boldsymbol{x}(t)\right]^H\right\} < \sum_{t=1}^{n} \lambda_n(t)\gamma(t) \ . \tag{89}$$

This constrains the possible parameter matrices to the set

$$\bar{\Omega}(n) = \left\{\boldsymbol{\Theta} \mid \sum_{t=1}^{n} \lambda_n(t) \operatorname{tr}\left\{\left[\boldsymbol{y}(t) - \boldsymbol{\Theta}^H\boldsymbol{x}(t)\right]\left[\boldsymbol{y}(t) - \boldsymbol{\Theta}^H\boldsymbol{x}(t)\right]^H\right\} < \sum_{t=1}^{n} \lambda_n(t)\gamma(t)\right\} \ . \tag{90}$$

Expanding the trace term,

$$\begin{aligned}\bar{\Omega}(n) = \Bigg\{\boldsymbol{\Theta} \mid \sum_{t=1}^{n} \lambda_n(t) \operatorname{tr}\left\{\boldsymbol{y}(t)\boldsymbol{y}^H(t) - \boldsymbol{\Theta}^H\boldsymbol{x}(t)\boldsymbol{y}^H(t) - \boldsymbol{y}(t)\boldsymbol{x}^H(t)\boldsymbol{\Theta} + \boldsymbol{\Theta}^H\boldsymbol{x}(t)\boldsymbol{x}^H(t)\boldsymbol{\Theta}\right\} \\ < \sum_{t=1}^{n} \lambda_n(t)\gamma(t)\Bigg\} \ .\end{aligned} \tag{91}$$

Moving the summation across terms,

$$\bar{\Omega}(n) = \left\{\boldsymbol{\Theta} \mid \operatorname{tr}\left\{\boldsymbol{C}_{\boldsymbol{y}}(n) - \boldsymbol{\Theta}^H\boldsymbol{C}_{\boldsymbol{xy}}(n) - \boldsymbol{C}_{\boldsymbol{xy}}^H(n)\boldsymbol{\Theta} + \boldsymbol{\Theta}^H\boldsymbol{C}(n)\boldsymbol{\Theta}\right\} < \sum_{t=1}^{n} \lambda_n(t)\gamma(t)\right\} \tag{92}$$

where definitions of $\boldsymbol{C}_{\boldsymbol{xy}}(\cdot)$ and $\boldsymbol{C}_{\boldsymbol{y}}(\cdot)$ are inherent. Since $\boldsymbol{C}_{\boldsymbol{xy}}(n) = \boldsymbol{C}(n)\boldsymbol{\Theta}(n)$,

$$\boldsymbol{C}_{\boldsymbol{xy}}^H(n) = \boldsymbol{\Theta}^H(n)\boldsymbol{C}^H(n) = \boldsymbol{\Theta}^H(n)\boldsymbol{C}(n). \tag{93}$$

This substitution in (92) and some simple manipulation yields

$$\bar{\Omega}(n) = \left\{\boldsymbol{\Theta} \mid \operatorname{tr}\left\{\boldsymbol{\Theta}^H\boldsymbol{C}(n)\boldsymbol{\Theta} - \boldsymbol{\Theta}^H\boldsymbol{C}(n)\boldsymbol{\Theta}(n) - \boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta}\right\} < \sum_{t=1}^{n} \lambda_n(t)\gamma(t) - \operatorname{tr}\left\{\boldsymbol{C}_{\boldsymbol{y}}(n)\right\}\right\} \tag{94}$$

Completing the square on the left side yields

$$\begin{aligned}\bar{\Omega}(n) = \Big\{\boldsymbol{\Theta} \mid \operatorname{tr}\left\{\boldsymbol{\Theta}^H\boldsymbol{C}(n)\boldsymbol{\Theta} - \boldsymbol{\Theta}^H\boldsymbol{C}(n)\boldsymbol{\Theta}(n) - \boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta} + \boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta}(n)\right\} \\ < \sum_{t=1}^{n} \lambda_n(t)\gamma(t) - \operatorname{tr}\left\{\boldsymbol{C}_{\boldsymbol{y}}(n)\right\} + \operatorname{tr}\left\{\boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta}(n)\right\} \overset{\text{def}}{=} \kappa(n)\Big\}\end{aligned} \tag{95}$$

The definition of $\kappa(n)$ in (95) is seen to be equivalent to that given in (78) by noting that $\operatorname{tr}\{\boldsymbol{C_y}(n)\} = \sum_{t=1}^{n} \lambda_n(t) \parallel \boldsymbol{y}(t) \parallel^2$. It follows that the set is described by

$$\bar{\Omega}(n) = \left\{\boldsymbol{\Theta} \;\mid\; \operatorname{tr}\left\{[\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]^H \boldsymbol{C}(n) [\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]\right\} < \kappa(n)\right\} . \qquad (96)$$

Since $\boldsymbol{C}(n)$ is positive definite almost surely, the left side of this inequality must be a positive number. Therefore $\kappa(n) > 0$. Dividing both sides by $\kappa(n)$ yields (77). □

To prove Corollary 1, it is convenient to write

$$\operatorname{tr}\left\{[\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]^H \boldsymbol{C}(n) [\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]\right\} = \sum_{j=1}^{k} c_j \qquad (97)$$

where c_j indicates the j^{th} diagonal element of $[\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]^H \boldsymbol{C}(n) [\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]$. Now it is clear that

$$c_i = [\boldsymbol{\theta}_i - \boldsymbol{\theta}_i(n)]^H \boldsymbol{C}(n) [\boldsymbol{\theta}_i - \boldsymbol{\theta}_i(n)] \qquad (98)$$

for any i, where $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_i(n)$ are the i^{th} columns of $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}(n)$, respectively. It is also true that all the c_j's are positive since $\boldsymbol{C}(n)$ is a positive definite matrix. Therefore,

$$c_i < \sum_{j=1}^{k} c_j < \kappa(n) \quad \text{for any } i \in [1, k] . \qquad (99)$$

Dividing through by $\kappa(n)$ yields inequality (79). □

Proof of Lemma 2: Case 1. Inserting the right side of (25) into (78) for $\boldsymbol{\Theta}(n)$ gives

$$\begin{aligned}
\kappa(n) \;=\; & \operatorname{tr}\left\{\left[\boldsymbol{\Theta}(n-1) + \lambda_n(n)\boldsymbol{C}^{-1}(n)\boldsymbol{x}(n)\boldsymbol{\varepsilon}^H(n, \boldsymbol{\Theta}(n-1))\right]^H \boldsymbol{C}(n)\right. \\
& \left.\times \left[\boldsymbol{\Theta}(n-1) + \lambda_n(n)\boldsymbol{C}^{-1}(n)\boldsymbol{x}(n)\boldsymbol{\varepsilon}^H(n, \boldsymbol{\Theta}(n-1))\right]\right\} + \sum_{t=1}^{n} \lambda_n(t)\gamma(t)\left(1 - \gamma^{-1}(t) \parallel \boldsymbol{y}(t) \parallel^2\right) (100)
\end{aligned}$$

$$\begin{aligned}
=\; & \operatorname{tr}\Big\{\boldsymbol{\Theta}^H(n-1)\left[\boldsymbol{C}_s(n-1) + \lambda_n(n)\boldsymbol{x}(n)\boldsymbol{x}^H(n)\right]\boldsymbol{\Theta}(n-1) + \\
& \lambda_n(n)\boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))\boldsymbol{x}^H(n)\boldsymbol{\Theta}(n-1) + \lambda_n(n)\boldsymbol{\Theta}^H(n-1)\boldsymbol{x}(n)\boldsymbol{\varepsilon}^H(n, \boldsymbol{\Theta}(n-1)) + \\
& \lambda^2(n)\boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))\boldsymbol{x}^H(n)\left[\boldsymbol{C}_s^{-1}(n-1) - \lambda_n(n)\frac{\boldsymbol{C}_s^{-1}(n-1)\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{C}_s^{-1}(n-1)}{1 + \lambda_n(n)G_s(n)}\right]\boldsymbol{x}(n)\boldsymbol{\varepsilon}^H(n, \boldsymbol{\Theta}(n-1) \\
& + \sum_{t=1}^{n} \lambda_n(t)\gamma(t)\left(1 - \gamma^{-1} \parallel \boldsymbol{y}(t) \parallel^2\right) \qquad (101
\end{aligned}$$

$$\begin{aligned}
=\; & \kappa_s(n-1) + \lambda_n(n)\gamma(n) - \lambda_n(n) \parallel \boldsymbol{y}(n) \parallel^2 + \\
& \lambda_n(n) \operatorname{tr}\Big\{[\boldsymbol{y}(n) - \boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))][\boldsymbol{y}(n) - \boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))]^H + \boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))[\boldsymbol{y}(n) - \boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))]^H + \\
& [\boldsymbol{y}(n) - \boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))]\boldsymbol{\varepsilon}^H(n, \boldsymbol{\Theta}(n-1)) + \frac{\lambda^2(n)\boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))\boldsymbol{\varepsilon}^H(n, \boldsymbol{\Theta}(n-1))G_s(n)}{1 + \lambda_n(n)G_s(n)}\Big\} \qquad (102
\end{aligned}$$

$$= \kappa_s(n-1) + \lambda_n(n)\gamma(n) - \lambda_n(n) \operatorname{tr}\left\{\boldsymbol{\varepsilon}(n, \boldsymbol{\Theta}(n-1))\boldsymbol{\varepsilon}^H(n, \boldsymbol{\Theta}(n-1))\right\}\left[1 - \frac{\lambda_n(n)G_s(n)}{1 + \lambda_n(n)G_s(n)}\right] . \qquad (103$$

Equation (84) follows immediately upon recognizing the trace term to be $\| \boldsymbol{\epsilon}(n, \boldsymbol{\Theta}(n-1)) \|^2$.

Case 2. Because $\boldsymbol{T}(n)$ is obtained by an orthonormal transformation of $\boldsymbol{X}(n)$ (see (19)),

$$\operatorname{tr}\left\{\boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta}(n)\right\} = \operatorname{tr}\left\{\boldsymbol{\Theta}^H(n)\boldsymbol{T}^H(n)\boldsymbol{T}(n)\boldsymbol{\Theta}(n)\right\} = \operatorname{tr}\left\{\boldsymbol{D}_1^H(n)\boldsymbol{D}_1(n)\right\}. \tag{104}$$

This is the first term in the basic expression for $\kappa(n)$ in (78). The second term can be written as

$$\frac{\sum_{t=1}^{n-1} \lambda_{n-1}(t)\gamma(t)\left(1-\gamma^{-1}(t) \| \boldsymbol{y}(t) \|^2\right)}{\zeta(n-1)} + \lambda_n(n)\gamma(n)\left(1-\gamma^{-1} \| \boldsymbol{y}(n) \|^2\right). \tag{105}$$

The desired recursion follows immediately. □

Proof of Proposition 2: For simplicity, let us denote $\lambda_n(n)$ by λ throughout the proof.

Volume case. Define

$$\boldsymbol{B}(n) \stackrel{\text{def}}{=} \kappa(n)\boldsymbol{C}^{-1}(n) \tag{106}$$

We wish to minimize $\mu_v(n) = \det \boldsymbol{B}(n)$, and it will be convenient to do so by minimizing the ratio[22]

$$\nu_v(n) \stackrel{\text{def}}{=} \frac{\mu_v(n)}{\mu_v(n-1)} = \frac{\det \boldsymbol{B}(n)}{\det \boldsymbol{B}(n-1)} = \det \frac{\boldsymbol{B}(n)}{\boldsymbol{B}(n-1)}. \tag{107}$$

From (24) and (106)

$$\begin{aligned}
\frac{\boldsymbol{B}(n)}{\kappa(n)} &= \frac{\boldsymbol{B}(n-1)}{\kappa_s(n-1)} - \lambda\frac{\boldsymbol{B}(n-1)\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{B}(n-1)}{\kappa_s^2(n-1)\left[1+\lambda G_s(n)\right]} \qquad (108)\\
&= \frac{\boldsymbol{B}(n-1)\zeta(n-1)}{\kappa(n-1)} - \lambda\frac{\zeta^2(n-1)\boldsymbol{B}(n-1)\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{B}(n-1)}{\kappa^2(n-1)\left[1+\lambda G_s(n)\right]}\\
&= \frac{\boldsymbol{B}(n-1)}{\kappa_s(n-1)} - \lambda\frac{\boldsymbol{B}(n-1)\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{B}(n-1)}{\kappa_s^2(n-1)\left[1+\lambda G_s(n)\right]}
\end{aligned}$$

Defining $h(n) \stackrel{\text{def}}{=} 1+\lambda G_s(n)$ and $r(n) \stackrel{\text{def}}{=} \kappa(n)/\kappa_s(n-1)$ yields

$$\boldsymbol{B}(n) = \boldsymbol{B}(n-1)r(n)\left\{\boldsymbol{I} - \frac{\lambda\boldsymbol{x}(n)}{\kappa_s(n-1)h(n)}\left[\boldsymbol{B}(n-1)\boldsymbol{x}(n)\right]^H\right\}. \tag{109}$$

So,

$$\nu_v(n) = \det\frac{\boldsymbol{B}(n)}{\boldsymbol{B}(n-1)} = \det\left\{r(n)\boldsymbol{I} - \frac{r(n)\lambda\boldsymbol{x}(n)}{\kappa_s(n-1)h(n)}\left[\boldsymbol{B}(n-1)\boldsymbol{x}(n)\right]^H\right\}. \tag{110}$$

Using the matrix identity [37] (for the complex case)

$$\det(c\boldsymbol{I} + \boldsymbol{v}\boldsymbol{z}^H) = c^{mk-1}(c + \boldsymbol{v}^H\boldsymbol{z}) \tag{111}$$

where $\boldsymbol{v}, \boldsymbol{z} \in \boldsymbol{C}^{m\times k}$ and c is a real number, we obtain

$$\nu_v(n) = r^{mk-1}(n)\left\{r(n) - \frac{r(n)\lambda\boldsymbol{x}^H(n)}{\kappa_s(n-1)h(n)}\boldsymbol{B}(n-1)\boldsymbol{x}(n)\right\}. \tag{112}$$

[22] Recall from Section 6.1 that $B(n-1)$ may be considered either the volume of the scaled or unscaled ellipsoid at time $n-1$.

This can be written as

$$\nu_v(n) = r^{mk}(n)\left\{1 - \frac{\lambda G_s(n)}{h(n)}\right\} = \frac{r^{mk}(n)}{h(n)}. \tag{113}$$

Therefore, to minimize $\nu_v(n)$ with respect to λ, (113) is differentiated and the result is set to zero,

$$\frac{\partial \nu_v(n)}{\partial \lambda} = \frac{\partial}{\partial \lambda}\left(\frac{r^{mk}(n)}{h(n)}\right) = \frac{mkr^{mk-1}(n)}{h(n)}\frac{\partial r(n)}{\partial \lambda} - \frac{r^{mk}(n)}{h^2(n)}G_s(n) \tag{114}$$

Since $r^{mk-1}(n) > 0$ (see proof of Proposition 1),

$$\frac{h^2(n)}{r^{mk-1}(n)}\frac{\partial \nu_v(n)}{\partial \lambda} = mkh(n)\frac{\partial r(n)}{\partial \lambda} - r(n)G_s(n). \tag{115}$$

Now using Lemma 2 we can write

$$r(n) = \frac{\kappa(n)}{\kappa_s(n-1)} = 1 + \frac{\lambda\gamma(n)}{\kappa_s(n-1)} - \frac{\lambda \parallel \varepsilon(n,\Theta(n-1)) \parallel^2}{\kappa_s(n-1)h(n)}. \tag{116}$$

Differentiating this result with respect to λ yields

$$\frac{\partial r(n)}{\partial \lambda} = \frac{1}{\kappa_s(n-1)}\left[\gamma(n) - \frac{\parallel \varepsilon(n,\Theta(n-1)) \parallel^2}{h^2(n)}\right]. \tag{117}$$

Putting this result in (115) and replacing $r(n)$ with the right side of (116) yields

$$\begin{aligned}\frac{h^2(n)}{r^{m-1}(n)}\frac{\partial \nu_v(n)}{\partial \lambda} = \frac{mkh(n)}{\kappa_s(n-1)}\left\{\gamma(n) - \frac{\parallel \varepsilon(n,\Theta(n-1)) \parallel^2}{h^2(n)}\right\} - \\ \left\{1 + \frac{\lambda\gamma(n)}{\kappa_s(n-1)} - \frac{\lambda \parallel \varepsilon(n,\Theta(n-1)) \parallel^2}{\kappa_s(n-1)h(n)}\right\}G_s(n).\end{aligned} \tag{118}$$

After some algebra,

$$\begin{aligned}\frac{\kappa_s(n-1)h^3(n)}{r^{mk-1}(n)}\frac{\partial \nu_v(n)}{\partial \lambda} = mk\left[\gamma(n)h^2(n) - \parallel \varepsilon(n,\Theta(n-1)) \parallel^2\right] - \\ \left[\kappa_s(n-1)h(n) + \lambda\gamma(n)h(n) - \lambda \parallel \varepsilon(n,\Theta(n-1)) \parallel^2\right]G_s(n)\end{aligned} \tag{119}$$

When $h(n)$ is replaced by $(1 + \lambda G_s(n))$ on the right, the following result is obtained

$$\frac{\kappa_s(n-1)h^3(n)}{r^{m-1}(n)}\frac{\partial \nu_v(n)}{\partial \lambda} = F_v(\lambda) \tag{120}$$

where $F_v(\lambda)$ is exactly the quadratic of (80). Since the factor in front of the derivative on the left is positive for any positive λ, a positive root of $F_v(\lambda)$ corresponds to $(\partial \nu_v(n)/\partial \lambda) = 0$.

It is noted that the discriminant of the quadratic is always positive so that the roots are always real. Moreover, when $a_0 > 0$ it is found that $a_1 > 0$ as well. Since a_2 is always positive, this implies that the roots are both negative, since no positive λ satisfies (80) in this case. On the other hand, when $a_0 < 0$, this immediately implies that the roots have opposite signs. Thus exactly one positive root is found. In the proof of Corollary 3 below, this root will be demonstrated to minimize the volume measure.

Trace case. As above $r(n) \overset{\text{def}}{=} \kappa(n)/\kappa_s(n-1)$ and $h(n) \overset{\text{def}}{=} (1+\lambda G_s(n))^{-1}$. Note that $\mu_t(n) =$ tr$\{bmB(n)\}$ and also define

$$\nu_t(n) \overset{\text{def}}{=} \frac{\mu_t(n)}{\mu_t(n-1)} = \frac{\operatorname{tr}\{\boldsymbol{B}(n)\}}{\operatorname{tr}\{\boldsymbol{B}(n-1)\}}. \tag{121}$$

Beginning with (109) it is easy to show that

$$\operatorname{tr}\{\boldsymbol{B}(n)\} = r(n)\left[\operatorname{tr}\{\boldsymbol{B}(n-1)\} - \frac{\lambda}{\kappa_s(n-1)h(n)}\boldsymbol{x}^H(n)\boldsymbol{B}^2(n-1)\boldsymbol{x}(n)\right], \tag{122}$$

so that

$$\begin{aligned} \nu_t(n) &= r(n) - \frac{r(n)\lambda}{\operatorname{tr}\{\boldsymbol{B}(n-1)\}\kappa_s(n-1)h(n)}\boldsymbol{x}^H(n)\boldsymbol{B}^2(n-1)\boldsymbol{x}(n) \\ &= r(n) - \frac{r(n)\lambda}{\operatorname{tr}\{\boldsymbol{B}(n-1)\}h(n)}\boldsymbol{x}^H(n)\boldsymbol{C}^{-2}(n-1)\boldsymbol{x}(n). \end{aligned} \tag{123}$$

Letting $H(n) \overset{\text{def}}{=} \boldsymbol{x}^H(n)\boldsymbol{C}^{-2}(n-1)\boldsymbol{x}(n)$ and $I(n-1) \overset{\text{def}}{=} \operatorname{tr}\{\boldsymbol{B}(n-1)\}$,

$$\nu_t(n) = r(n) - \frac{r(n)\lambda}{I(n-1)h(n)}H(n) = r(n)\left(1 - \frac{\lambda}{I(n-1)h(n)}H(n)\right). \tag{124}$$

Differentiating with respect to lambda and setting the result to zero yields

$$\frac{\partial\nu_t(n)}{\partial\lambda} = \frac{\partial r(n)}{\partial\lambda}\left(1 - \frac{\lambda}{I(n-1)h(n)}H(n)\right) - \frac{r(n)}{I(n-1)h(n)}\left(1 - \frac{G_s(n)\lambda}{h(n)}\right)H(n) = 0. \tag{125}$$

Now using (116) and (117) in this result yields

$$\begin{aligned} \frac{\partial\nu_t(n)}{\partial\lambda} &= \frac{1}{\kappa_s(n-1)}\left(\gamma(n) - \frac{\|\,\varepsilon(n,\boldsymbol{\Theta}(n-1)\,\|^2}{h^2(n)}\right)\left(1 - \frac{\lambda}{I(n-1)h(n)}H(n)\right) \\ &\quad - \left(1 + \frac{\lambda\gamma(n)}{\kappa_s(n-1)} - \frac{\|\,\varepsilon(n,\boldsymbol{\Theta}(n-1))\,\|^2}{\kappa_s(n-1)H(n)}\right)\frac{1}{I(n-1)h(n)}\left(1 - \frac{G_s(n)\lambda}{h(n)}\right)H(n) = 0. \end{aligned} \tag{126}$$

After algebraic manipulation this becomes

$$\frac{\partial\nu_t(n)}{\partial\lambda} = \frac{N(\lambda)}{D(\lambda)} \tag{127}$$

where $N(\lambda)$ is precisely the cubic equation $F_t(\lambda)$ described in the proposition, and $D(\lambda) = \kappa_s(n-1)\gamma^{-1}(n)h^3(n)$. Since $D(\lambda) > 0$ for all $\lambda > 0$, it is sufficient to seek the positive root(s) of the numerator.

It is straightforward to show that coefficients b_2 and b_3 are always positive. When $b_0 > 0$ then $b_1 > 0$, so in this case there can be no positive solution to $F_t(\lambda) = 0$. When $b_0 < 0$, we claim that there is exactly one positive real root. The quantity $(-b_0/b_3)$, which is real and positive, is the product of the roots, so there must be at least one real positive root. The remaining two are a complex conjugate pair, or are both negative or both positive. Now the quantity $(-b_2/b_3)$ is the sum of the roots and is negative. This guarantees that the remaining two roots cannot both be positive. Therefore, the remaining two are either complex or negative, and the claim is verified. □

Sketch of Proof of Corollary 2: The procedure parallels the steps used to prove Proposition 2 for the volume case [68]. □

Proof of Corollary 3: Again we use λ to indicate $\lambda_n(n)$, and λ^* for $\lambda_n^*(n)$. Since n is fixed, we write the volume ratio of (107) to show its explicit dependence upon λ, $\nu_v(\lambda)$, and suppress the dependence upon n. From (115),

$$\frac{\partial \nu_v(\lambda)}{\partial \lambda} = Q(\lambda)R(\lambda) \tag{128}$$

where, for n fixed, we make the definitions

$$Q(\lambda) \stackrel{\text{def}}{=} \frac{r^{mk-1}(n)}{h^2(n)} \quad \text{and} \quad R(\lambda) \stackrel{\text{def}}{=} mkh(n)\frac{\partial r(n)}{\partial \lambda} - r(n)G_s(n). \tag{129}$$

For future reference, also notice that

$$\kappa_s(n-1)h(n)R(\lambda) = F_v(\lambda) \tag{130}$$

where $F_v(\lambda)$ is the volume quadratic of Proposition 2. This becomes evident upon comparing (114) and (120). Consequently,

$$\frac{\partial^2 \nu_v(\lambda)}{\partial \lambda^2} = \frac{\partial R(\lambda)}{\partial \lambda}Q(\lambda) + R(\lambda)\frac{\partial Q(\lambda)}{\partial \lambda} \tag{131}$$

It is easy to demonstrate that $Q(\lambda)$ is positive, and that its derivative is bounded, for $\lambda \in [0, \infty)$. Now with the aid of (117) we can write

$$\frac{\partial R(\lambda)}{\partial \lambda} = (mk-1)G_s(n)\frac{\partial r(n)}{\partial \lambda} + 2mk\frac{\| \varepsilon(n, \Theta(n-1)) \|^2 \, G_s(n)}{\kappa_s(n-1)h^2(n)}. \tag{132}$$

Because of (130) it is clear that $R(\lambda^*) = 0$. Reference to the definition of $R(\lambda)$ in (129), therefore immediately shows that

$$\left.\frac{\partial r(n)}{\partial \lambda}\right]_{\lambda=\lambda^*} > 0. \tag{133}$$

Consequently,

$$\left.\frac{\partial R(\lambda)}{\partial \lambda}\right]_{\lambda=\lambda^*} > 0. \tag{134}$$

It follows immediately that

$$\left.\frac{\partial^2 \nu_v(\lambda)}{\partial \lambda^2}\right]_{\lambda=\lambda^*} > 0 \tag{135}$$

so that λ^* corresponds to a minimum of $\nu_v(\lambda)$ with respect to λ (see Fig. 19). Further, since

$$r(n)]_{\lambda=0} = 1 \quad \text{(see (116)), and} \quad h(n)]_{\lambda=0} = 1 \tag{136}$$

we have from (113) that $\nu_v(\lambda)]_{\lambda=0} = 1$, and also that $Q(0) = 1$. Therefore, from (128) and (130),

$$\left.\frac{\partial \nu_v(\lambda)}{\partial \lambda}\right]_{\lambda=0} = R(0) = \frac{F_v(0)}{\kappa_s(n-1)} = \frac{a_0}{\kappa_s(n-1)} < 0 \tag{137}$$

where a_0 is the zero order coefficient of the quadratic which is negative if an optimal root exists. It follows that $\nu_v(\lambda^*) < 1$ and the corollary is proven . □

Proof of Proposition 3: For simplicity, we write $\lambda_n(n)$ as λ. Using (84) from Lemma 2, we can write

$$\dot{K}(\lambda) \overset{\text{def}}{=} \frac{\partial \kappa(n, \lambda)}{\partial \lambda} = \frac{G_s^2(n)\gamma(n)\lambda^2 + 2G_s(n)\gamma(n)\lambda + [\gamma(n) - \| \varepsilon(n, \Theta(n-1) \|^2]}{G_s(n)^2\lambda^2 + 2G_s(n)\lambda + 1} \tag{138}$$

and

$$\ddot{K}(\lambda) \overset{\text{def}}{=} \frac{\partial^2 \kappa(n, \lambda)}{\partial \lambda^2} = \frac{2[G_s^2(n) + \gamma(n)G_s(n)] \| \varepsilon(n, \Theta(n-1) \|^2}{(G_s^2(n)\lambda^2 + 2G_s(n)\lambda + 1)^2}. \tag{139}$$

The denominator of $\dot{K}(\lambda)$ is positive on $\lambda \in (0, \infty)$ and therefore $\dot{K}(\lambda)$ has a root on $\lambda \in (0, \infty)$ iff its numerator does. The roots of the numerator are always real. Moreover, the numerator has a unique positive root on $(0, \infty)$ iff $[\gamma(n) - \| \varepsilon(n, \Theta(n-1)) \|^2] < 0$. Further since $\ddot{K}(\lambda) > 0$ for all $\lambda > 0$, the root, if it exists, will correspond to a minimum of $\kappa(n, \lambda)$. □

Proof of Proposition 4: If (67) holds, then a_0 of $F_v(\lambda)$, and b_0 of $F_t(\lambda)$, are both negative. Now see the proof of Proposition 2. □

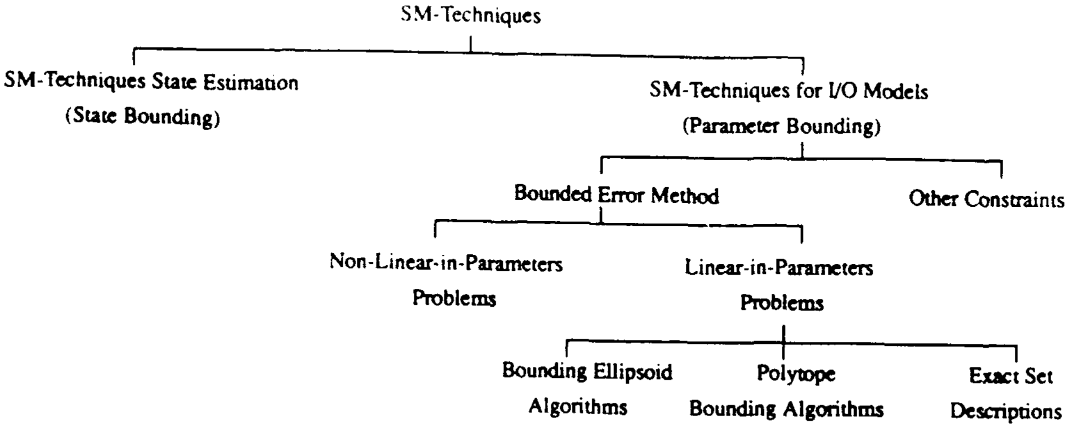


Figure 1: A taxonomy of SM methods.

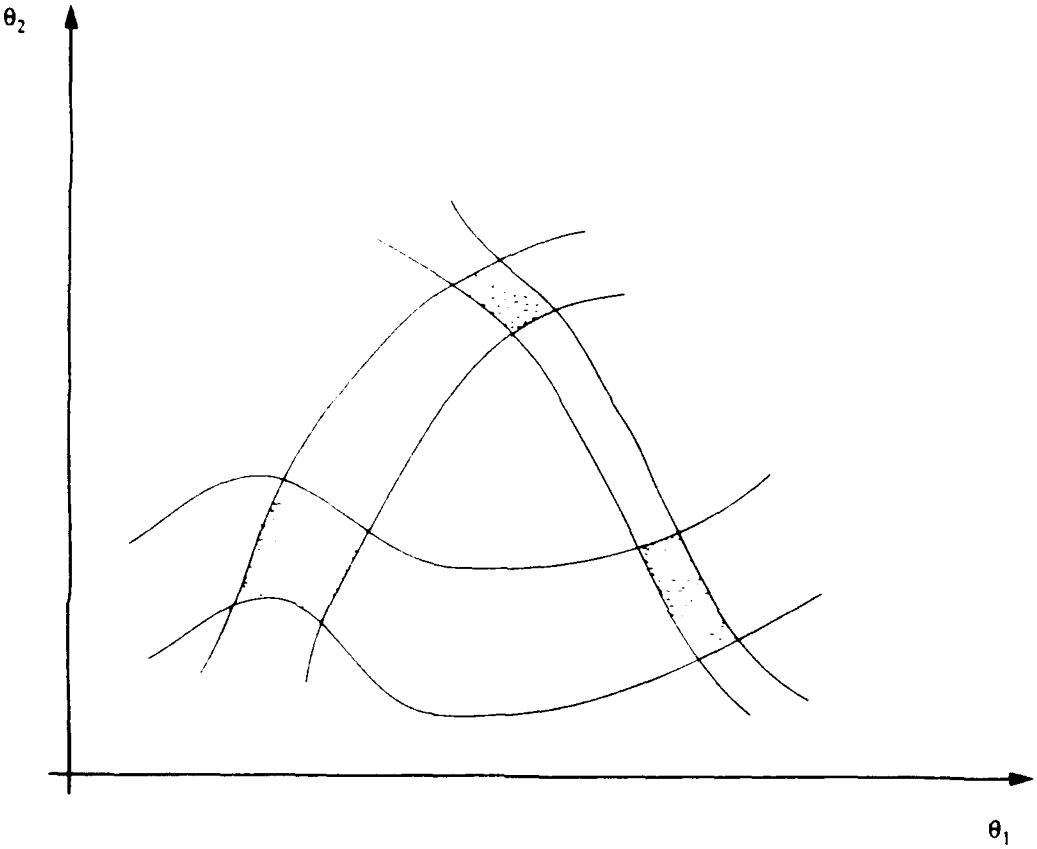


Figure 2: For non-LP models, pointwise feasible sets are hypersurfaces, and the accumulated feasible set, $\Omega(n)$, formed by their intersection can be highly irregular. $\Omega(n)$ need not be connected in the parameter space. Illustrated is the case of a real signal, SISO system for which the parameters comprise real vectors of dimension two.

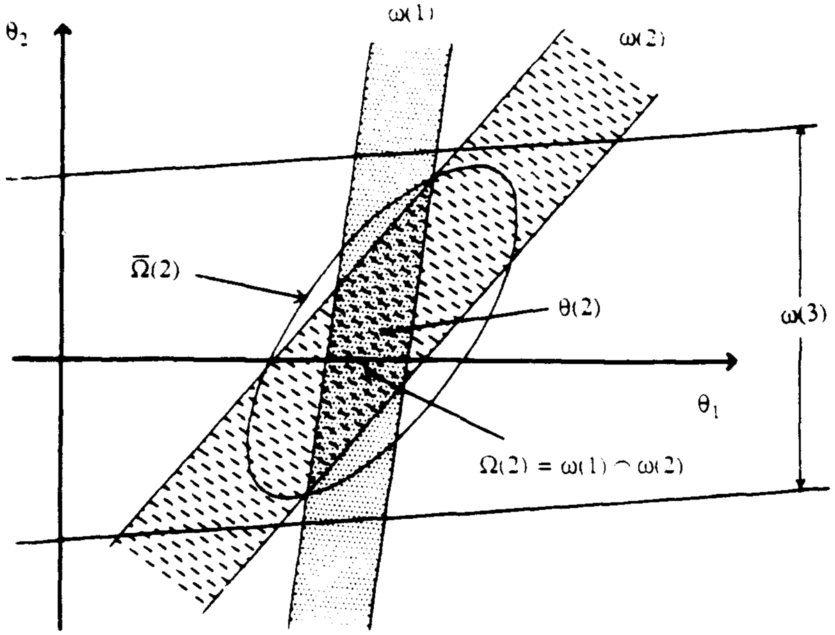


Figure 3: In the LP model case, error bounding implies pointwise "hyperstrip" regions of possible parameter sets in the space, which, when intersected over a given time range usually form convex polytopes of feasible parameters. These sets are called $\Omega(n)$ when time range $t \in [1, n]$ is included. Associated with a LSE problem with weights $\lambda_n(\cdot)$ is a hyperellipsoidal set $\bar{\Omega}(n)$ which is centered on the LSE estimate which contains the feasible set $\Omega(n)$ and, consequently, the true parameters Θ_*.Illustrated is the case in which the parameters comprise a real vector of dimension two.

Figure 4:

The UOBE algorithm based on QR-WRLS and volume minimization. The case of a scalar output is shown.

```
INITIALIZATION: Fill $(m+1) \times (m+1)$ working matrix, $\boldsymbol{W}$, with zeros.
                $\lambda(n) = \zeta(n) = 1, \quad n = 1, 2, \ldots, m+1$
                $\tilde{\kappa}(0) = 0$

RECURSION:      For $n = 1, 2, \ldots,$

    STEP 1.     (Skip[23] if $n \leq m+1$) Update $G_s(n)$, $\varepsilon(n, \boldsymbol{\theta}(n-1))$.
                $\boldsymbol{T}_s(n-1) = \zeta^{-1}(n-1)\boldsymbol{T}(n-1)$  (multiply top $m$ rows of $\boldsymbol{W}$ by $\zeta^{-1}(n-1)$)
                Solve $\boldsymbol{T}_s{}^H(n-1)\boldsymbol{g}(n) = \boldsymbol{x}(n)$ for $\boldsymbol{g}(n)$ by back substitution.
                $G_s(n) = \| \boldsymbol{g}(n) \|^2$
                $\varepsilon(n, \boldsymbol{\theta}(n-1)) = y(n) - \boldsymbol{\theta}^H(n-1)\boldsymbol{x}(n)$
    STEP 2.     (Skip if $n \leq m+1$) Check for and compute optimal $\lambda_n^*(n)$.
                Consider $a_0$ of (44). If $a_0 \geq 0$, set $\lambda_n^*(n) \equiv 0$. Go to STEP 3.
                If $a_0 < 0$, solve (44) for positive root $\lambda_n^*(n)$.
    STEP 3.     (Skip if $n \leq m+1$) If $\lambda_n^*(n) = 0$, set
                $\boldsymbol{T}(n) = \boldsymbol{T}_s(n-1)$
                $\boldsymbol{\theta}(n) = \boldsymbol{\theta}(n-1)$
                $\tilde{\kappa}(i) = \tilde{\kappa}(i-1)$
                and go to STEP 7.
                Otherwise, continue.
    STEP 4.     Update $\boldsymbol{T}(n)$.
                Replace bottom row of $\boldsymbol{W}$ by $\sqrt{\lambda_n^*(n)}\left[\boldsymbol{x}^H(n) \mid y(n)\right]$.
                Rotate this "new equation" into $\boldsymbol{W}$ using Givens rotations,
                leaving the result $[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$ in the upper $m$ rows of $\boldsymbol{W}$.
                These rotations involve the scalar computations (e.g. [33])
                $W'_{jk} = W_{jk}\sigma + W_{m+1,k}\tau\delta$ and $W'_{m+1,k} = -W_{jk}\tau\delta + W_{m+1,k}\sigma\delta$
                for $k = j, j+1, \ldots, m+1$ and for $j = 1, 2, \ldots, m$;
                where, $\sigma = W_{jj}/\rho$, $\tau = W_{m+1,k}/\rho$, $\rho = \sqrt{W_{jj}^2 + \delta W_{m+1,j}^2}$, $\delta$ is unity[24]
                and $W_{jk}$ ($W'_{jk}$) is the $j,k$ element of $\mathbf{W}$ pre- (post-) rotation.
    STEP 5.     (Skip if $n \leq m$) Update $\boldsymbol{\theta}(n)(n)$, solving $\boldsymbol{T}(n)\boldsymbol{\theta}(n) = \boldsymbol{d}_1(n)$ by
                back substitution.
    STEP 6.     Update $\kappa(n)$ and $\tilde{\kappa}(n)$ according to
                $\tilde{\kappa}(n) = \left[\tilde{\kappa}(i-1)/\zeta(r-1)\right] + \lambda_n^*(n)\gamma(n)(1 - \gamma^{-1}(n)y^2(n))$
                $\kappa(n) = \| \boldsymbol{d}_1(n) \|^2 + \tilde{\kappa}(n)$
                Compute and store only $\tilde{\kappa}(i)$ if $i \leq m$.
    STEP 7.     If new data set $(y(n+1), \boldsymbol{x}(n+1))$ available, return to STEP 1.
```

[23] Generally $T(\boldsymbol{n})$ does not become nonsingular until $n = m+1$. The first $\theta(n)$ cannot be computed until $\boldsymbol{n} = m+1$ and the first $\lambda_n^*(\boldsymbol{n})$ at $\boldsymbol{n} = \boldsymbol{m} + 2$. We arbitrarily set $\lambda(n) = 1$ on the initial range.

[24] δ is set to -1 to rotate an equation *out* of the estimate [33].

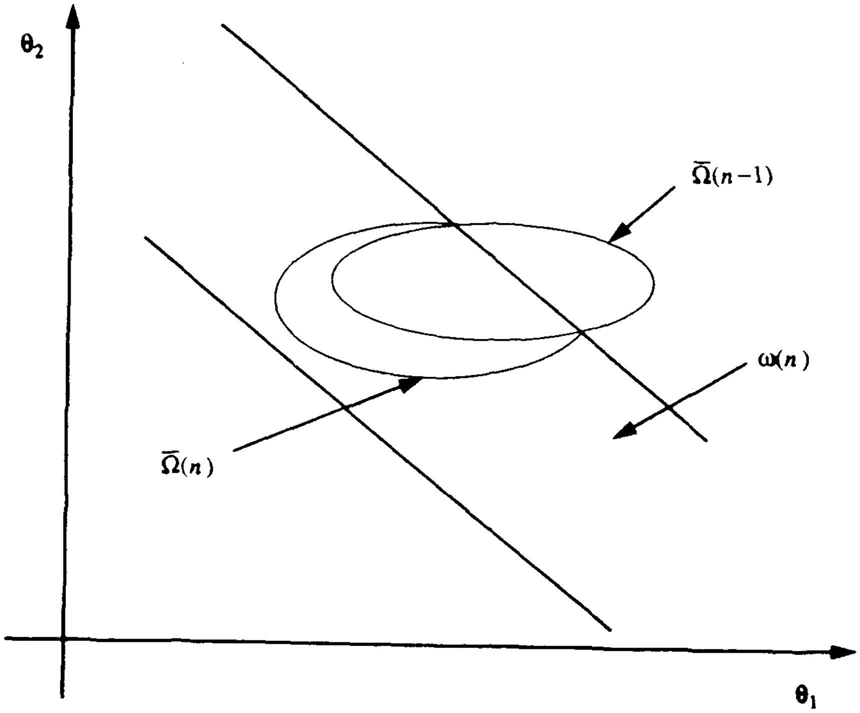


Figure 5: The F-H OBE algorithm circumscribes the intersection of the current hyperstrip and the existing hyperellipsoid, $\omega(n) \cap \bar{\Omega}(n-1)$, with another hyperellipsoid $\bar{\Omega}(n)$.

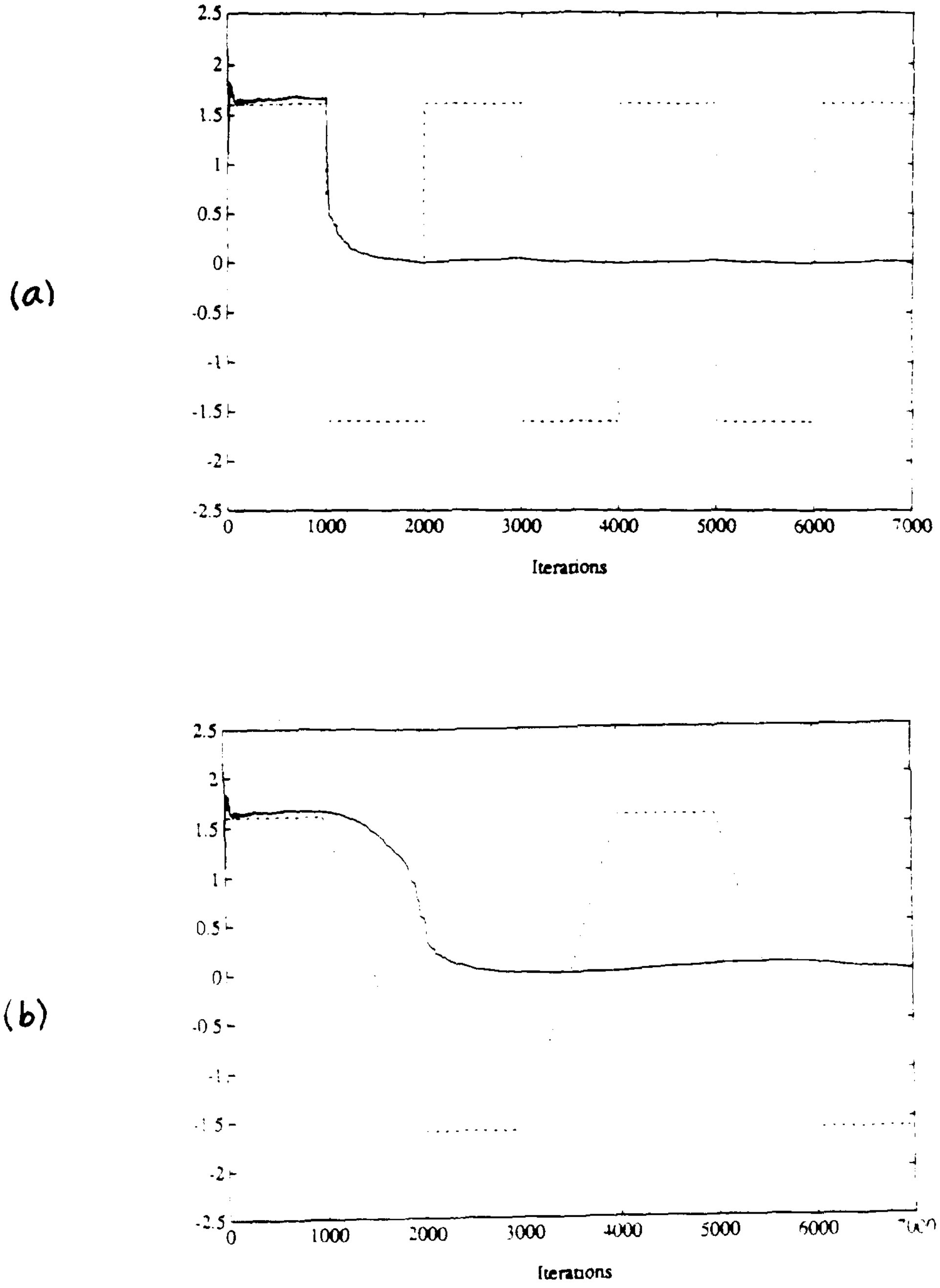


Figure 6: Results of using RLS on the (a) fast system, and (b) slow system. The dotted curve represents a_1, in each case, and the solid curve the parameter estimate.

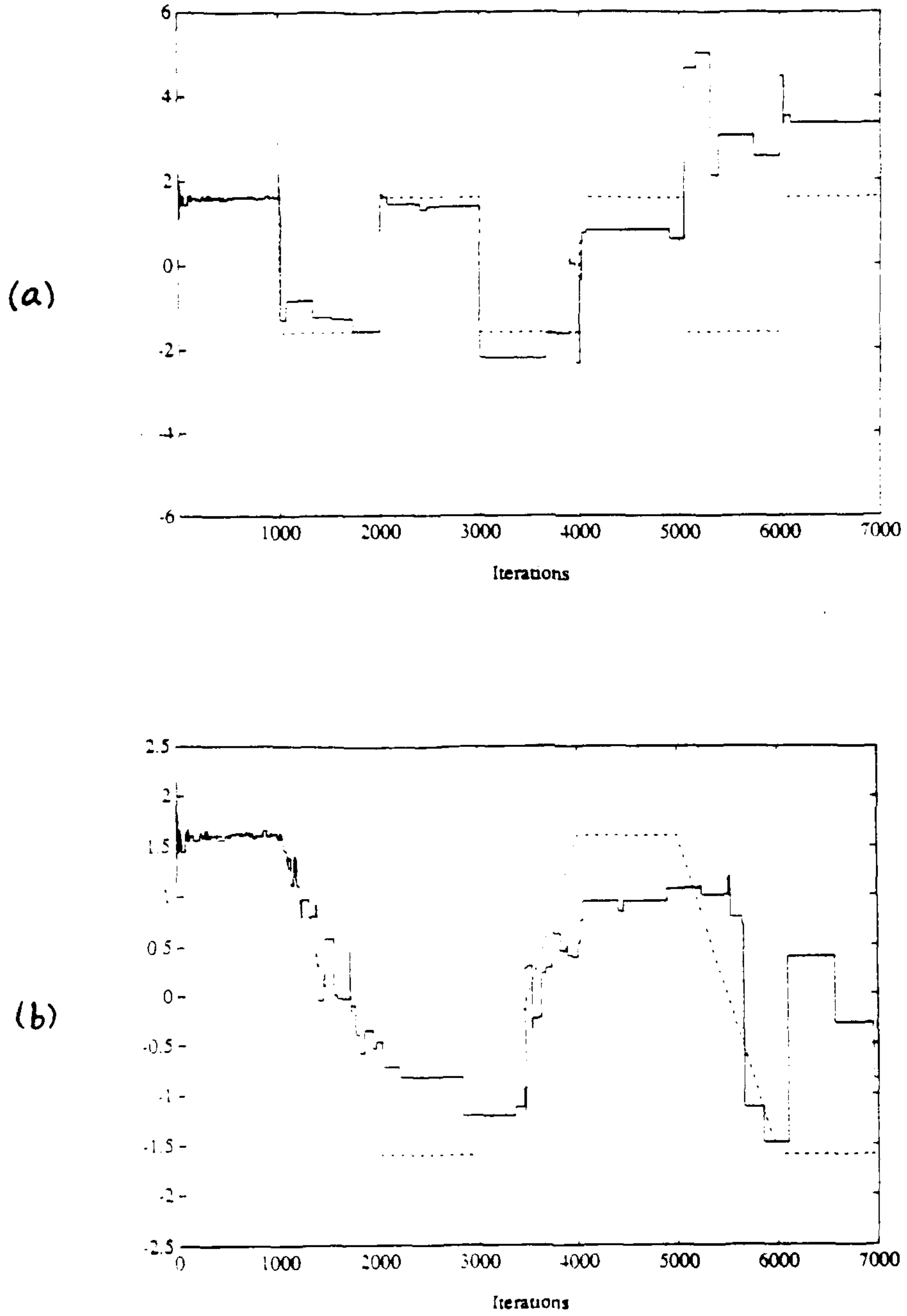


Figure 7: Results of using "nonadaptive" SM-WRLS on the (a) fast system: ρ = 0.020, and (b) slow system: ρ = 0.025.

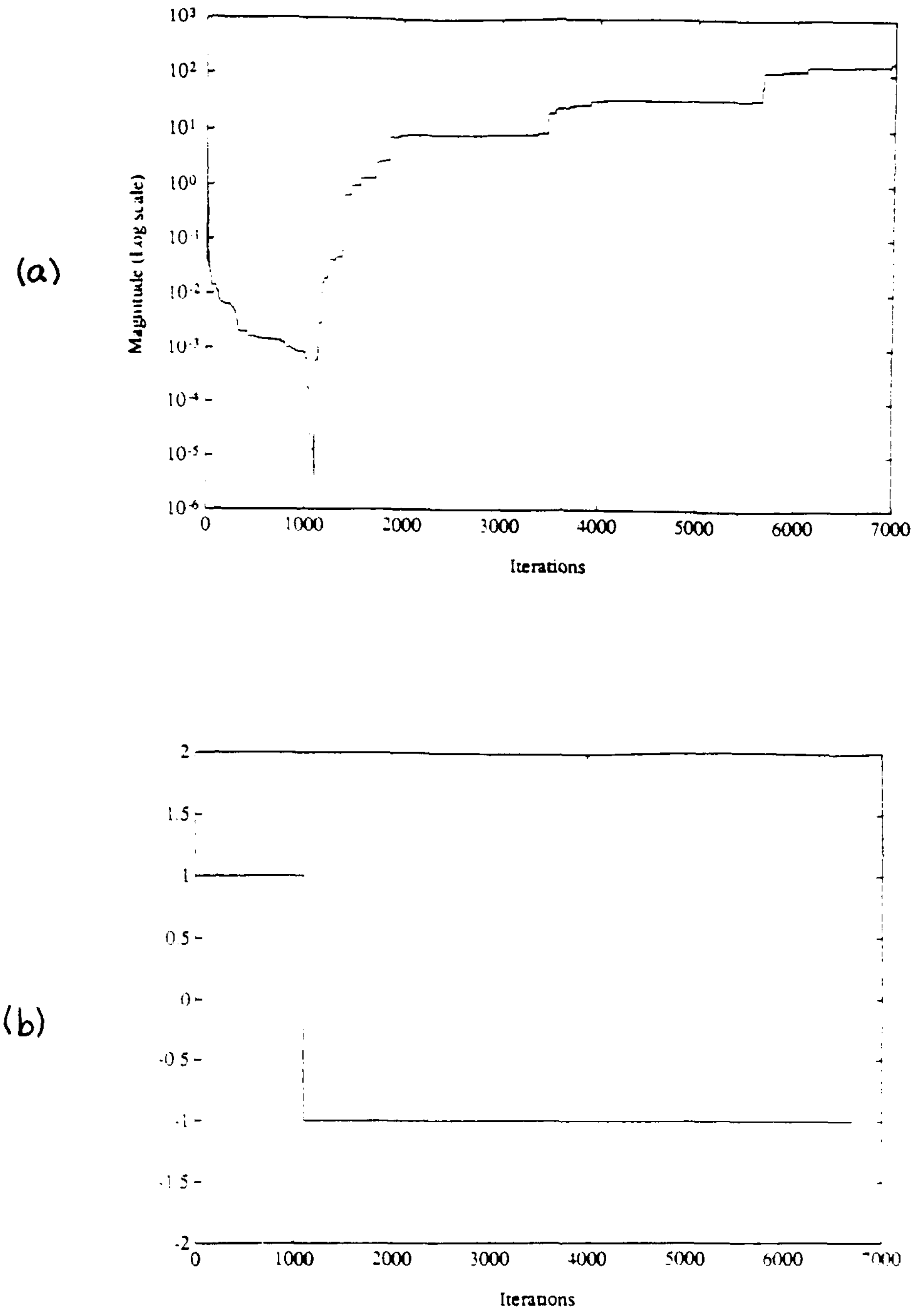


Figure 8: (a) $\mu_v(n)$ and (b) sgn$\{\kappa(n)\}$ for the "nonadaptive" SM-WRLS simulation of Fig. 7(b).

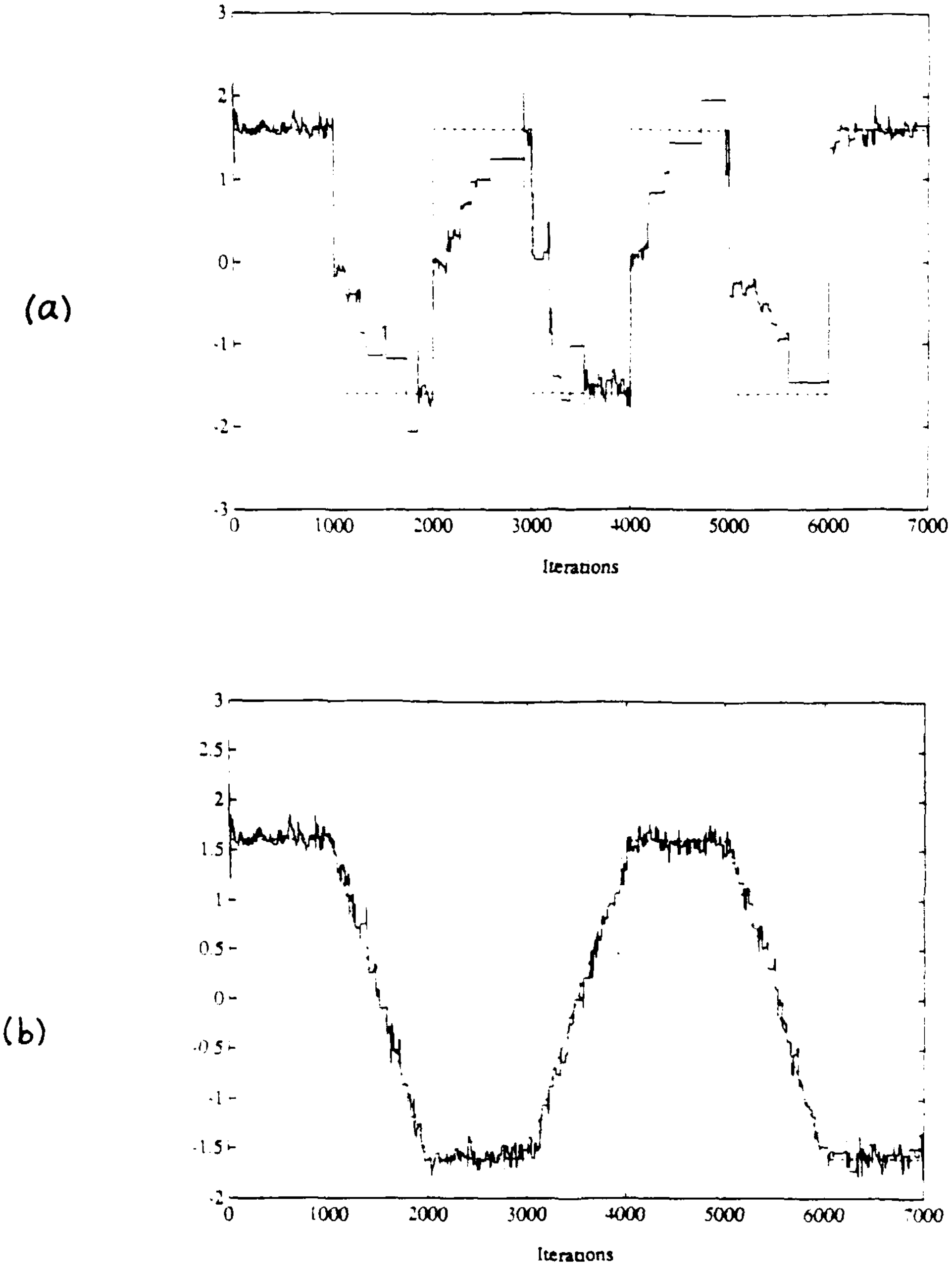


Figure 9: Parameter estimate results using exponential forgetting adaptation with optimal weights overridden as described in the text. Forgetting factor $\alpha = 0.99$. (a) Fast system: $\rho = 0.073$. (b) Slow system: $\rho = 0.094$.

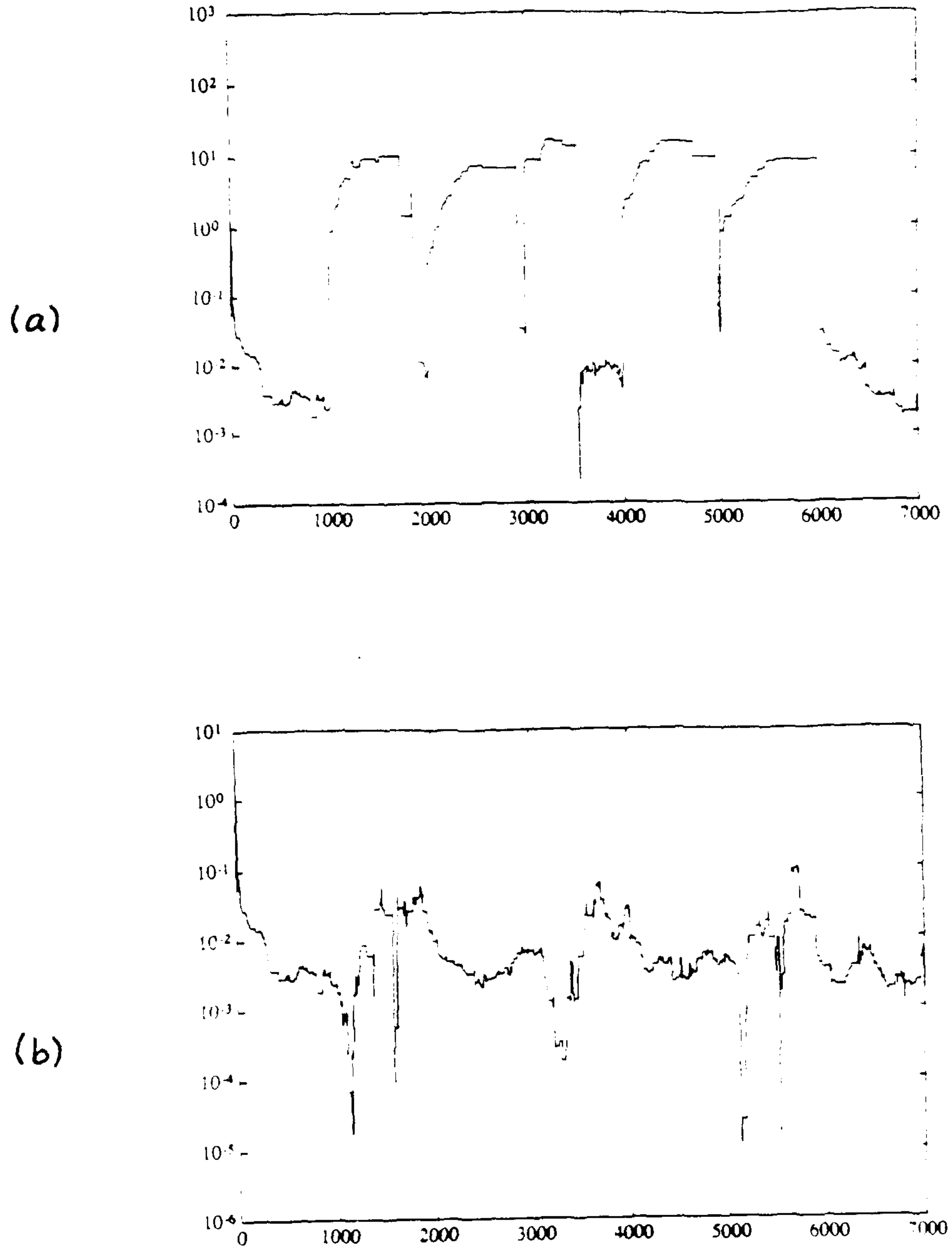


Figure 10: $\mu_v(n)$ vs. n for the exponential forgetting experiments of Fig. 9 . (a) Fast system. (b) Slow system.

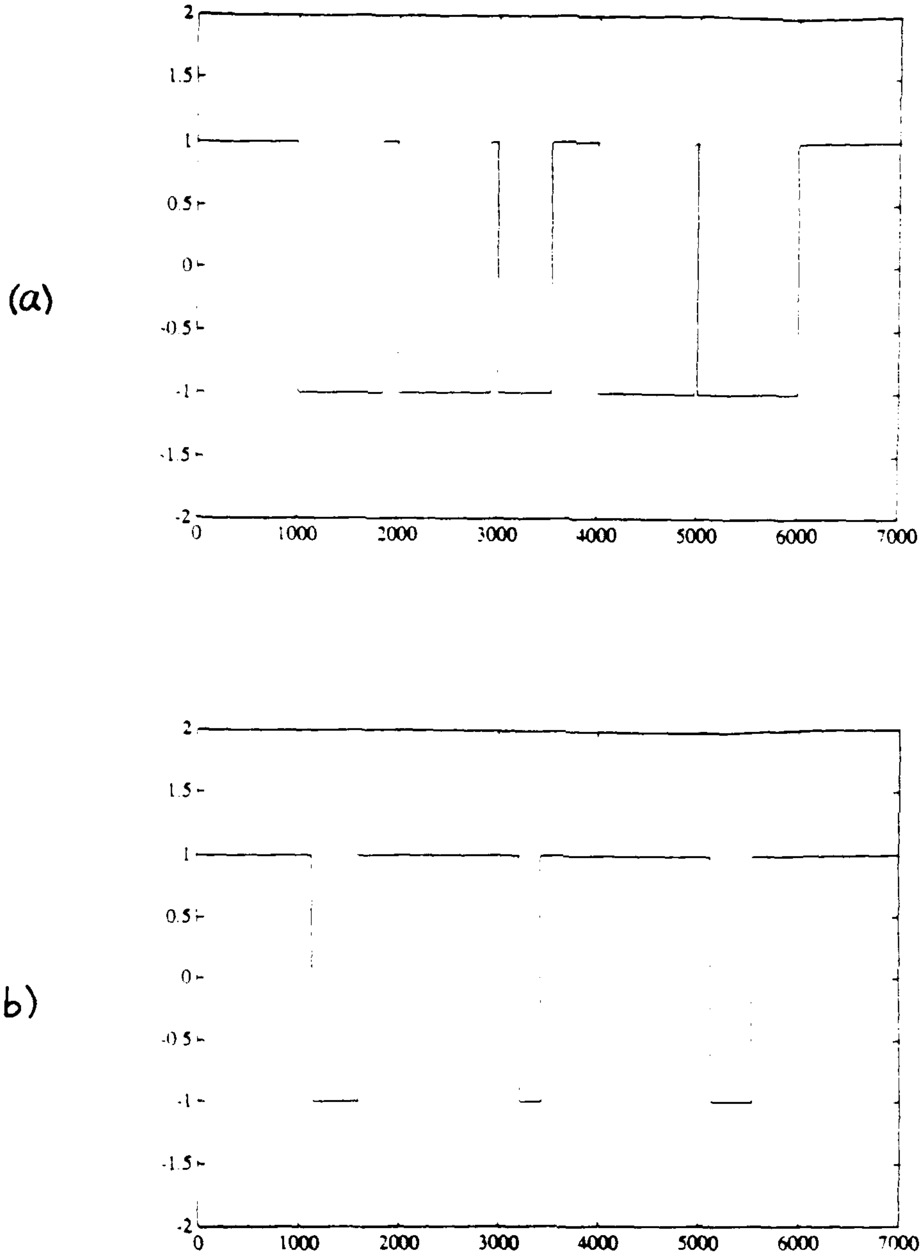


Figure 11: sgn{$\kappa(n)$} vs. n for the exponential forgetting experiments of Fig. 9 . (a) Fast system. (b) Slow system.

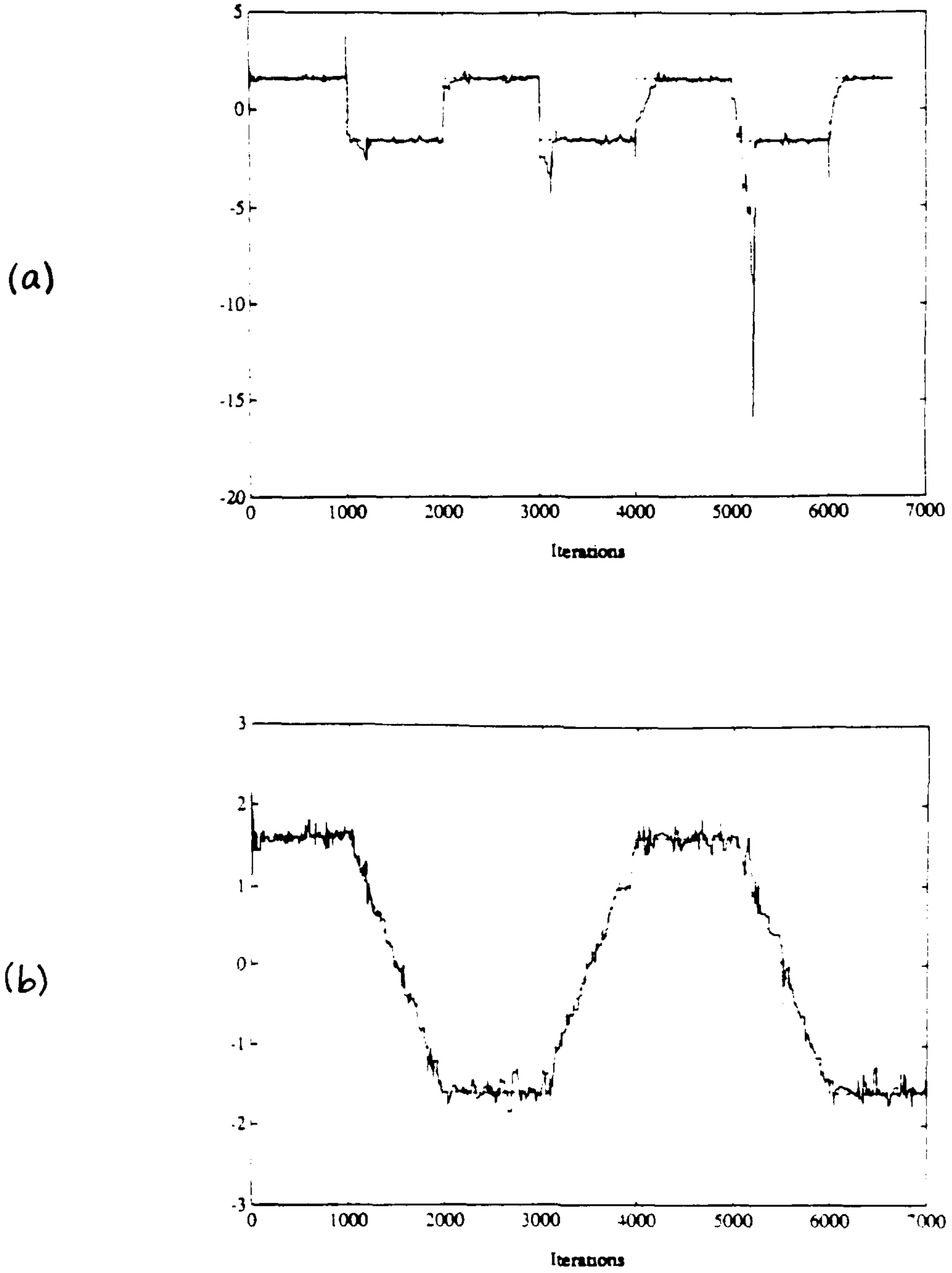


Figure 12: Simulation result of the windowed SM-WRLS algorithm ($l = 250$). (a) Fast system: $\rho = 0.094$. (b) Slow system: $\rho = 0.10$.

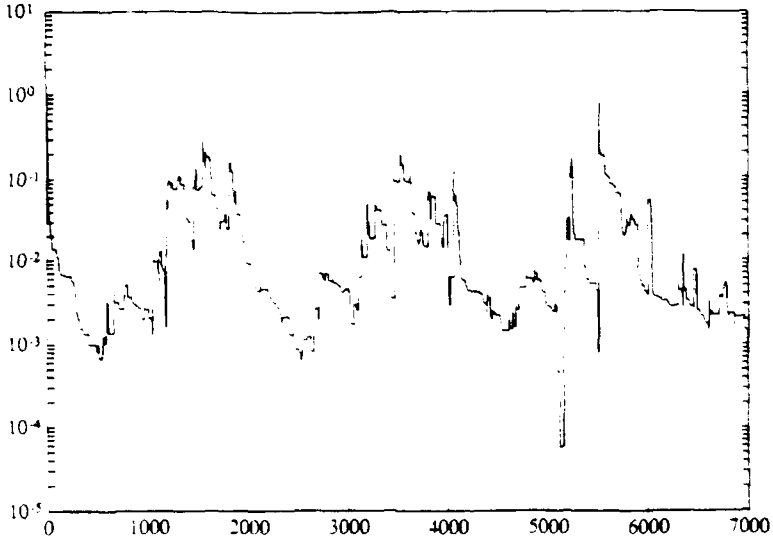


Figure 13: $\mu_v(n)$ vs. n for the $l = 250$ length window and the slow system.

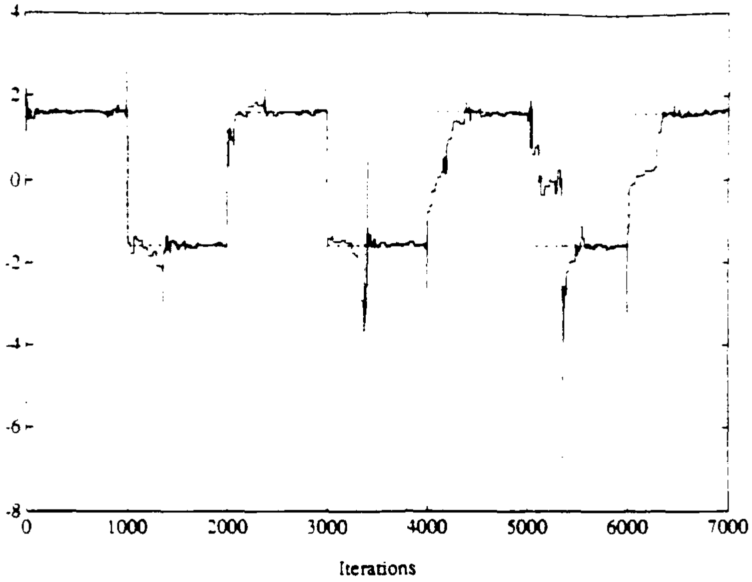


Figure 14: Simulation result of the windowed SM-WRLS algorithm (l = 500) for the fast system: ρ = 0.070.

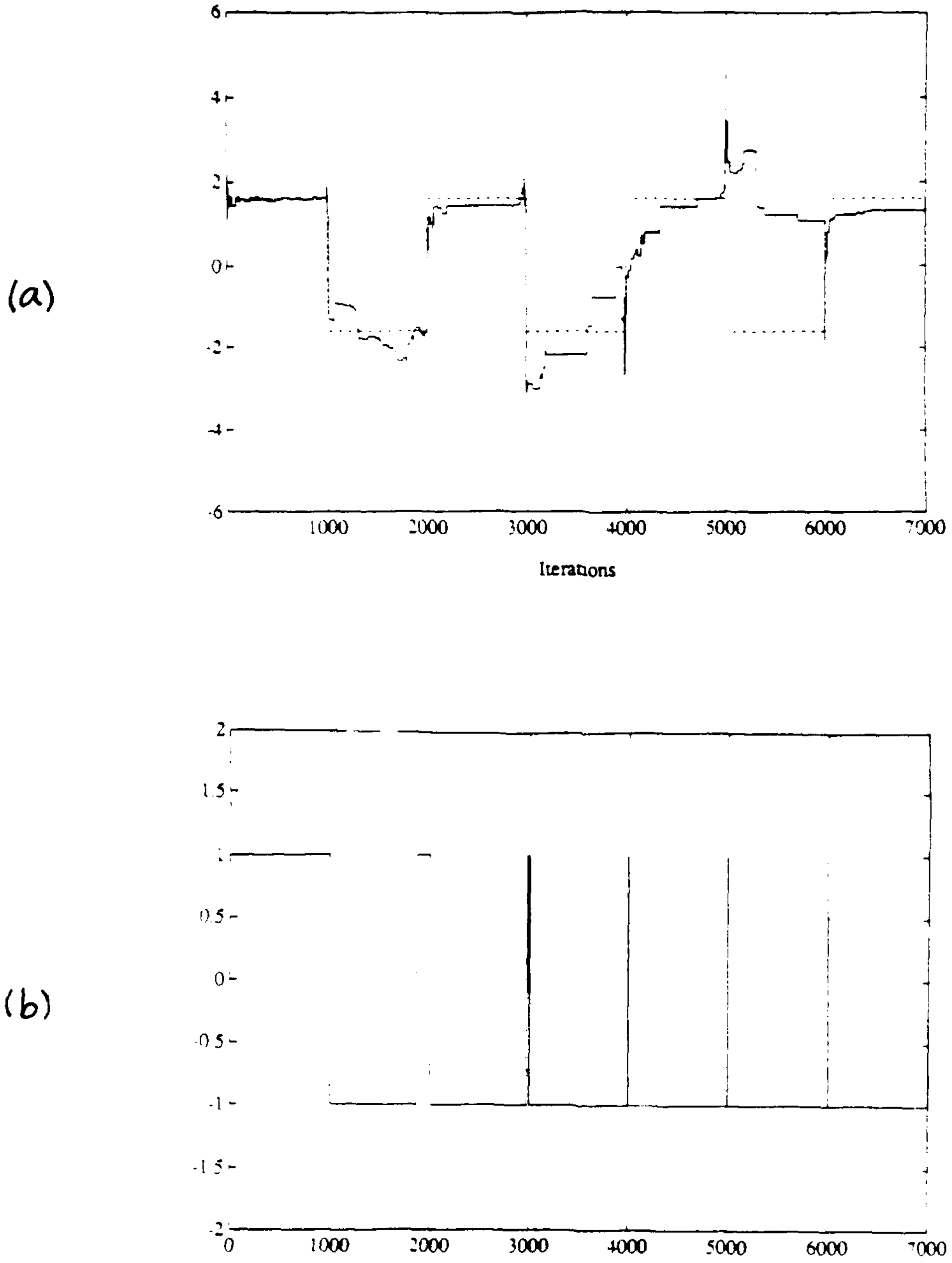


Figure 15: Parameter estimate and $\kappa(n)$ vs. n for the fast system using windowed estimation with $l = 1000$. The process has eroded and no longer operates according to proper SM principles.

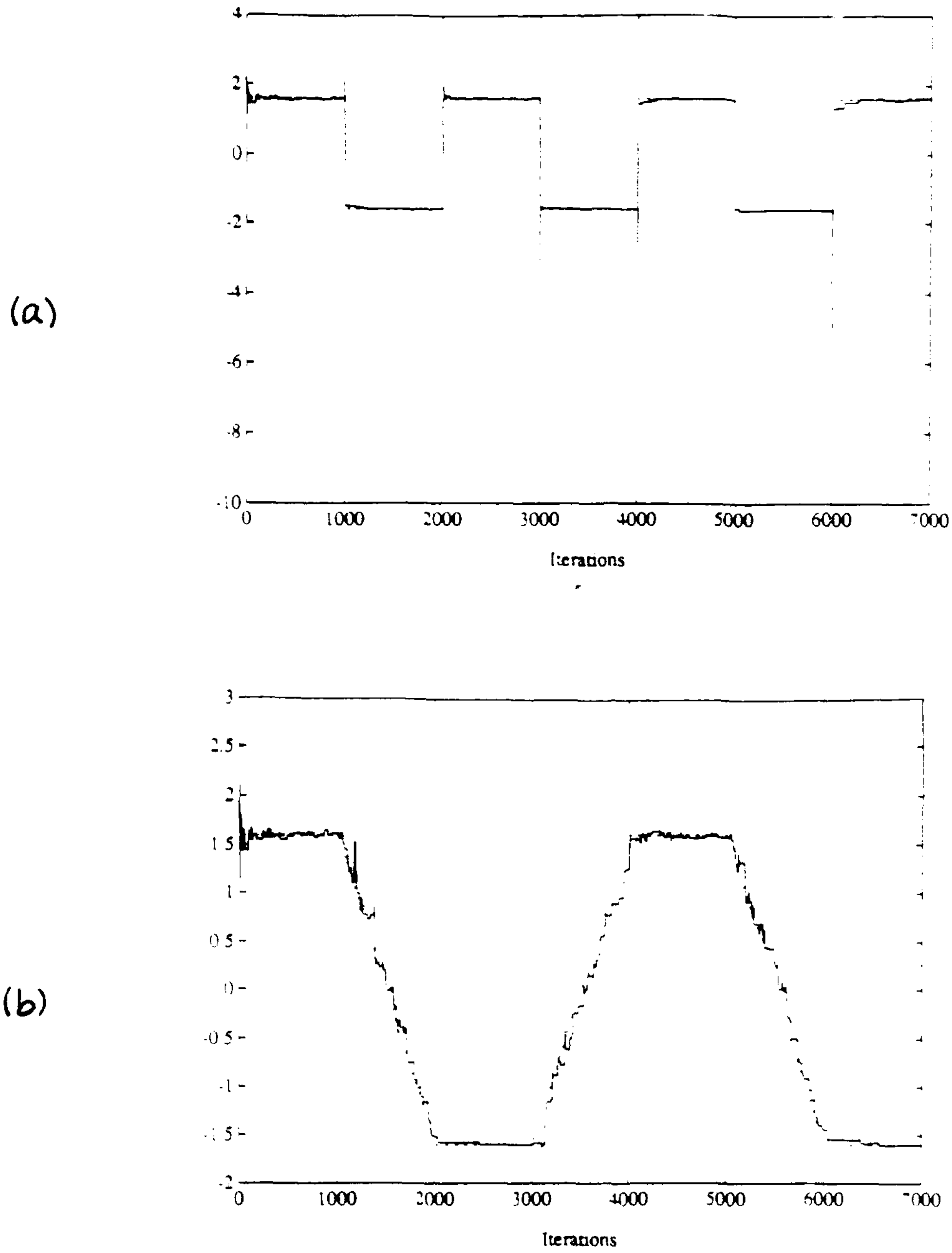


Figure 16: Simulation result of the selective forgetting SM-WRLS algorithm. (a) Fast system: ρ = 0.050, b = 0.042. (b) Slow system: ρ = 0.064, b = 0.047.

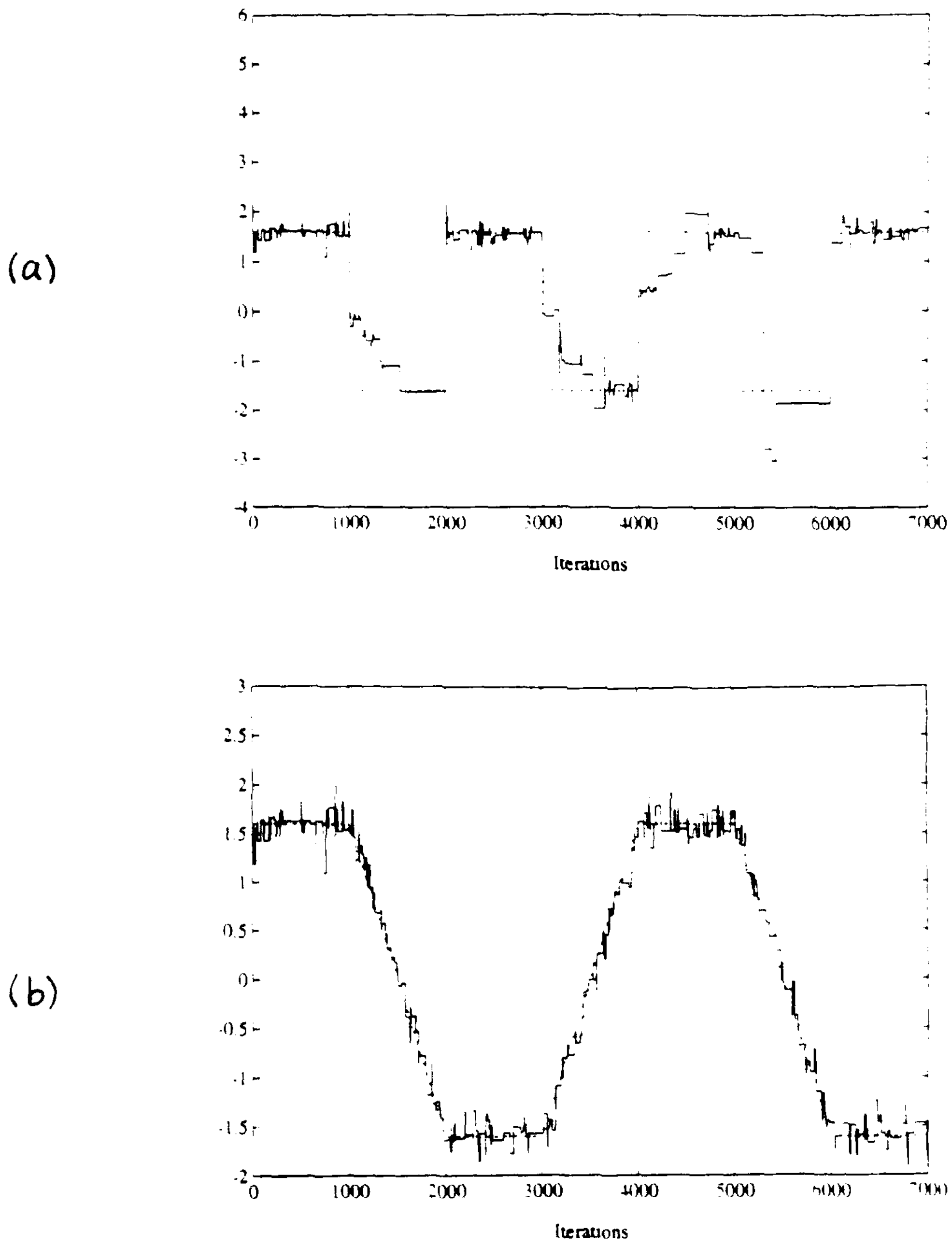


Figure 17: Parameter estimate results for exponential forgetting using suboptimal checking. Weights are "overridden" as described in Section 6.4.2. Forgetting factor $\alpha = 0.99$. (a) Fast system: $\rho = 0.038$. (b) Slow system: $\rho = 0.051$.

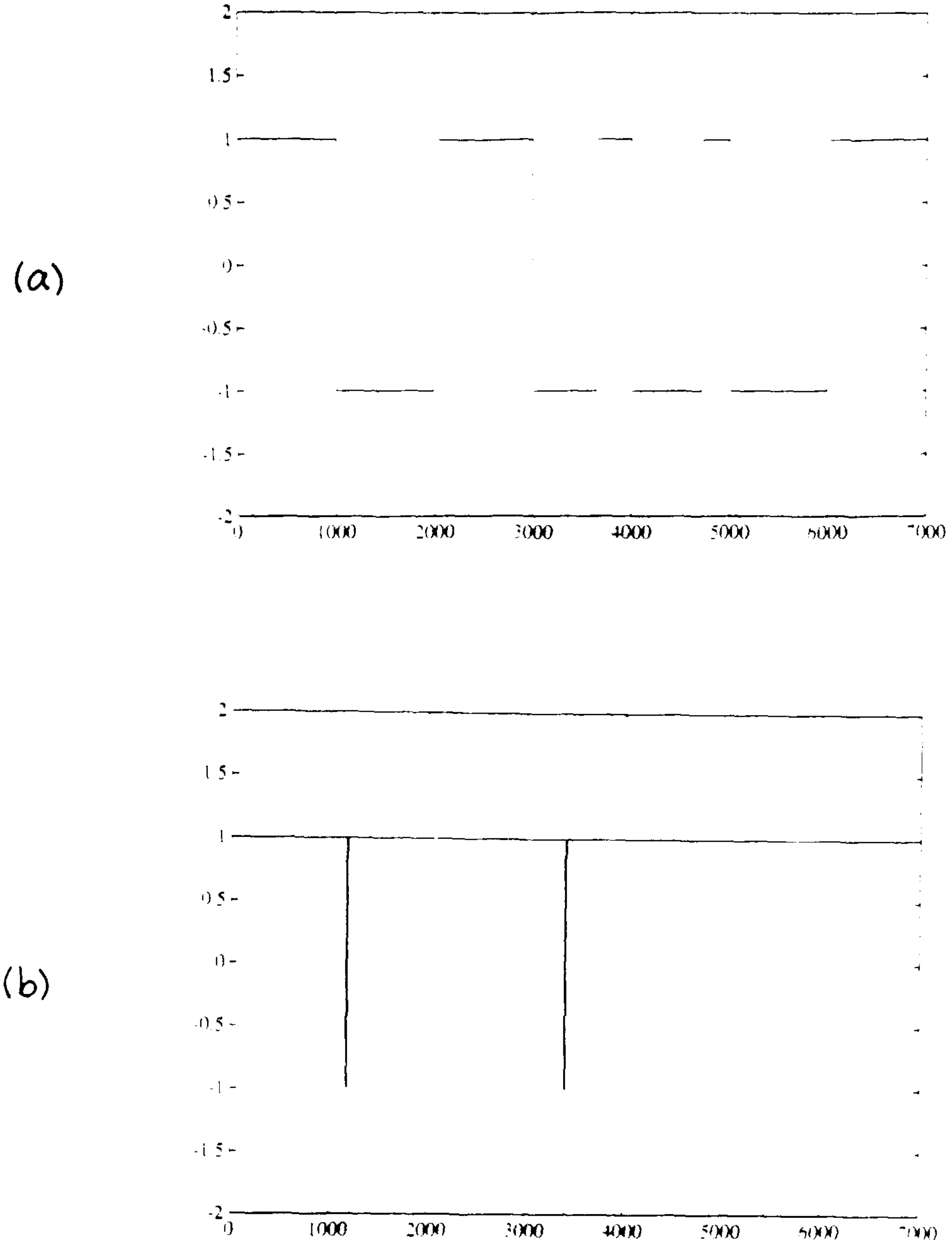


Figure 18: $\kappa(n)$ vs. n for the exponential forgetting experiments with suboptimal checking. (a) Fast system. (b) Slow system.

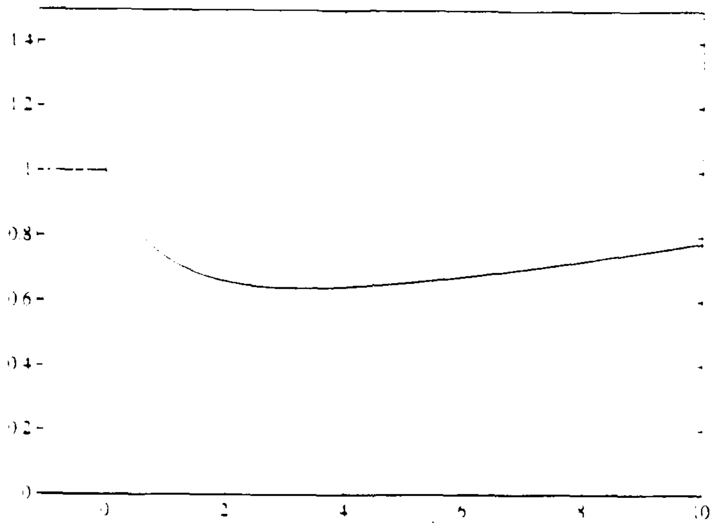


Figure 19: Form of the volume ratio as a function of λ when an optimal weight exists.

Algorithm	Checking	Adaptation by Back-Rotation	Covariance & Solution Update
Conventional MIL-WRLS		$b(2m^2+2m)$	$3m^2+(k+3)m$
Conventional QR-WRLS		$b[2.5m^2+(4k+1)m]$	$(2.5+0.5k)m^2+(4.5k+1)m$
Optimal UOBE (MIL-WRLS)	$m^2+(k+1)m+k$	$b(2m^2+2m)$	$\rho[2m^2+(k+2)m])$
Optimal UOBE (QR-WRLS)	$0.5m^2+(1.5+k)m+k$	$b[2.5m^2+(4k+1)m]$	$\rho[(2.5+0.5k)m^2+(4.5k+1)m]$
Suboptimal UOBE (MIL-WRLS)	$km+k$	$b'(2m^2+2m)$	$\rho'[3m^2+(k+3)m]$
Suboptimal UOBE (QR-WRLS)	$km+k$	$b'[2.5m^2+(4k+1)m]$	$\rho'[(2.5+0.5k)m^2+(4.5k+1)m]$
Parallel Optimal UOBE (QR-WRLS)	$3m$	$b(11m)$	$\rho(11m)$
Parallel Suboptimal UOBE (QR-WRLS)	$(m+1)$	$b'(11m)$	$\rho'(11m)$

Table 1:
Approximate computational complexities in average number of cflops per data set for the various techniques discussed in the text. m is the number of parameters in the model; k the dimension of the output vector; ρ and ρ' represent the average number of data sets accepted per n in the optimal and suboptimal cases, respectively (typically $\rho' < \rho$); and b and b' are the average number of back-rotations performed per n in the optimal and suboptimal cases, respectively (typically $b' < b$). For each sequential algorithm scaling or adaptation by exponential forgetting require $0.5m^2 + (k + 0.5)m$ cflops for each procedure. If both procedures are to be used, they can be combined and implemented at about the same cost as a single procedure. In the parallel cases, scaling and exponential forgetting can be achieved at virtually no cost. In the parallel processing cases, the loads in the table represent *parallel* complexities (see text), and results are for the case $k = 1$ since architectures for the MO case have no been devloped.

'Linearized' Alternatives to Back-Propagation Based on Recursive QR Decomposition

S.D. Hunt and J.R. Deller, Jr.

Michigan State University
Department of Electrical Engineering / 260 EB
Control, Systems, and Signal Processing Group: Speech Processing Laboratory
East Lansing, MI 48824-1226

(517) 353-8840 hunt@ee.msu.edu, deller@ee.msu.edu bitnet: hunt@msuegr, deller@msuegr

Abstract

A class of algorithms is presented for training multilayer perceptrons which implement nonlinear mappings using purely "linear" techniques. The methods are based upon linearizations of the network using error surface analysis, followed by a contemporary least squares estimation procedure. Specific algorithms are presented to estimate weights node-wise, layer-wise, and for estimating the entire set of network weights simultaneously. In several experimental studies, the node-wise method is superior to back-propagation and an alternative linearization method due to Azimi-Sadjadi *et al.* in terms of number of convergences and convergence rate. The layer and network- wise updating offers further improvement.

Acknowledgements

This work was supported by the National Science Foundation under Grant No. MIP-9016734 and by the Office of Naval Research under Contract No. N00014-91-J-1329. Mr. Hunt was also supported by a fellowship from the University of Puerto Rico.

List of Tables

List of Figures

1 Introduction

This paper introduces a new class of learning algorithms for multilayer perceptrons (MLP) with improved convergence properties. In spite of the nonlinearites present in the dynamics of a MLP, the learning algorithm is purely "linear" in the sense that it is based on a contemporary version of the conventional recursive least squares (RLS) algorithm (e.g. [1]). Accordingly, unlike the popular "nonlinear" algorithms used to train MLPs, the linear algorithm and its potential variants will benefit from the well-understood theoretical properties of RLS and VLSI architectures for its implementation.

A MLP is a an artifical neural network consisting of nodes grouped into layers. In this paper, we consider a two-layer network[1], an example of which is illustrated in Fig. 1, but the generalization of the method to an arbitrary number of layers will be obvious. Each node above the input layer in the MLP passes the sum of its weighted inputs through a non-linearity to produce its output. The inputs to layer zero are external. The outputs of the last layer are the outputs of the network.

Let us now formalize the network and define notation. The number of nodes in layer i is denoted N_i, with N_0 indicating the number of input nodes at the bottom of the network. The weights connecting to node k (k') of layer two (one) are held in the N_1-vector (N_0-vector) $\boldsymbol{w}_k$ ($\boldsymbol{w}'_{k'}$). The inputs to the nodes in all layers except the first are the outputs of the layer below. We denote by N the number of *training patterns*

$$\{(\boldsymbol{x}(n), \boldsymbol{t}(n)), \quad n = 1, 2, \ldots, N\}, \tag{1}$$

in which each $\boldsymbol{x}(n)$ is an N_0-vector of inputs to the bottom layer of the network, and each $\boldsymbol{t}(n)$ is an

[1] Some authors might choose to call this a *three* layer network. We shall designate the bottom layer of "nodes" as "layer zero" and not count it in the total number of layers. Layer zero is a set of linear nodes which simply pass the inputs unaltered. For this reason, we choose not to show circular nodes in the diagram.

N_2-vector of *target* outputs (final layer outputs which are *desired* in response to the corresponding input). The *computed* outputs of layer two (one) in response to $\boldsymbol{x}(n)$ comprise the N_2-vector (N_1-vector) $\boldsymbol{y}(n)$ ($\boldsymbol{y}'(n)$). Throughout the discussion, r_j will be used to denote the j^{th} element of vector $\boldsymbol{r}$.

Finally, we need to formalize the nonlinearity associated with the nodes. For given weights, $\boldsymbol{w}_k$, connected to node k of the final layer, for example, the output in response to input $\boldsymbol{x}(n)$ is

$$y_k(n) = S(\boldsymbol{w}_k^T \boldsymbol{y}'(n)) \tag{2}$$

in which $S(\cdot)$ is a nonlinear mapping. Typically, for example, a sigmoidal function would be used:

$$S(\alpha) = \frac{1}{1+e^{-\alpha}}. \tag{3}$$

Any function which is once differentiable can be employed in the methods to be presented. Finally, for convenience we also define

$$u_k(n) \overset{\text{def}}{=} \boldsymbol{w}_k^T \boldsymbol{y}'(n). \tag{4}$$

Clearly, $u_k(n)$ is the input to node k in the output layer in response to pattern n. $u_l'(n)$ is similarly defined as the input to node l in layer one.

Many training (weight estimation) algorithms exist for this type of network [2] - [6]. The most popular is the so-called *back-propagation* algorithm [5], [6]. Back-propagation performs satisfactorily in some cases if given enough time to converge. However, convergence can be too slow for many applications (e.g. [7])

One attempt to develop faster training methods is represented by the class of algorithms in which the network mapping is "linearized" in some sense in order to take advantage of linear estimation algorithms. In particular, $S(\cdot)$ can be replaced by a linear approximation. A recent example employing this strategy is the method reported by Azimi-Sadjadi *et al.* [2]. We shall refer to this technique as the *A-S algorithm*. While the *initial* method developed in this paper will be shown to be equivalent in certain *theoretical* senses to the A-S algorithm, its derivation is quite different (providing a second interpretation of the underlying linearization) and the *implementation* approach will result in significantly improved performance.

2 Linearization Algorithm

The training problem for the two layer MLP is stated as follows: Given a set of N training patterns as in (1), find the network weights which minimize the sum of squared errors,

$$E = \sum_{n=1}^{N} (\boldsymbol{t}(n) - \boldsymbol{y}(n))^T (\boldsymbol{t}(n) - \boldsymbol{y}(n)). \tag{5}$$

The purpose of this section is to describe the theoretical basis for a "linearized" solution of this problem.

Before continuing, we note a simple fact which will reduce the number of details in our discussion. It is easy to show from (5) that if only the weights connected to the output layer are allowed to change, with the other weights in the network held constant, then E is minimized by minimizing the errors associated with each node independently. That is, if E_k denotes the error associated with output node k,

$$E_k = \sum_{n=1}^{N} [t_k(n) - y_k(n)]^2, \tag{6}$$

then $E = \sum_{k=1}^{N_2} E_k$. In any given layer, each node in that layer involves a distinct set of weights (those directly connected to it), and each set of weights may be optimized (to reduce its node's error) independently of the others. This means that without loss of generality, we may focus on a single output node. (Whatever optimization method is discovered for this node will then be applied to other nodes in the layer.) Let us concentrate on, say, node k and seek weights which minimize (6).

First we wish to concentrate on the training of the weights in the final layer, so let us write E_k in a form which explicitly features these weights,

$$E_k = \sum_{n=1}^{N} [t_k(n) - S(\boldsymbol{w}_k^T \boldsymbol{y}'(n))]^2. \tag{7}$$

Algorithms for finding the optimal solution, say $\boldsymbol{w}_k^*$, to this problem are well-known if the modeled output depends only upon a *linear* combination of pattern-invariant (constant) weights. In the linear case $y_k(n) = S(\boldsymbol{w}_k \boldsymbol{y}'(n)) = \beta \boldsymbol{w}_k \boldsymbol{y}'(n)$, for some constant β (which can be taken as unity without loss of generality), and the error expression takes the form

$$E_k = \sum_{n=1}^{N} [t_k(n) - \boldsymbol{w}_k^T \boldsymbol{y}'(n)]^2. \tag{8}$$

The solution in the linear case is the solution to the classical linear least squares "normal equations" [8]. The solution of the normal equations can proceed in a variety of ways. It is also possible to arrive at the solution without explicitly forming the normal equations. This is the case, for example, when using the ***least mean square*** (LMS) (e.g. see [9] or [10]) algorithm, a recursive solution which amounts to "back-propagation" for a linear network. A second popular method is the conventional ***recursive least square (RLS)*** algorithm (e.g. see [11]). A contemporary version of the latter will serve as a computational basis for the algorithm be described in this paper, and RLS is also the basis for the A-S algorithm to which we wish to relate the method of this paper. Appropriate description and formalism will be introduced as needed.

It is well-known that least squares estimation problems may be discussed in terms of their error surfaces, in this case the graph of E_k as a function of $\boldsymbol{w}_k$. Whatever the form of the least square estimation algorithm, the ideal goal is to find the weight vector, say $\boldsymbol{w}_k^*$, corresponding to the global minimum of $E_k(\boldsymbol{w}_k)$. It is important to future developments to note that E_k depends not only on $\boldsymbol{w}_k$ but also upon the training patterns $\{(\boldsymbol{x}(n), t_k(n)),\ \ n \in [1, N]\}$ (see (7)). (In fact, since we have "frozen" the weights in the first layer, it is more appropriate in this case to view E_k as a function of $\boldsymbol{w}_k$ and the pairs $\{(\boldsymbol{y}'(n), t_k(n)), n \in [1, N]\}$.) Once the training patterns are fixed, the error function may be described as a surface over the N_1-dimensional hyperplane corresponding to the weights. Theoretically, the pairs $\{(\boldsymbol{y}'(n), t_k(n)), n \in [1, N]\}$ represent partial realizations of a two dimensional stochastic process which generates them. In this sense

$$E_k(\boldsymbol{w}_k, \{(\boldsymbol{y}'(n), t_k(n)), n \in [1, N]\}) \tag{9}$$

is only a sample error surface. In a pure sense, we would like to find weights corresponding to the global minimum of $\mathcal{E}\{E_k(\boldsymbol{w}_k)\}$ where $\mathcal{E}$ denotes the expected value. We must be content, however, to work with the sample surface provided by the training data.

The point of this discussion of error surfaces is to note that different algorithms construct and use different sample error surfaces from the data. With LMS (or back-propagation), error surfaces are sequentially constructed from *individual* training patterns, i.e., error surfaces of the form

$$E_k(\boldsymbol{w}_k, (\boldsymbol{y}'(n), t_k(n))), \quad n = 1, 2, \ldots, N \tag{10}$$

are created, and for each n, the weights are moved in the direction of the negative gradient on that surface. The convergence properties are well-understood. RLS[2], on the other hand, creates

[2] Of course, here we are speaking of a linear model identification.

sequentially more refined error surfaces of the form

$$E_k(\boldsymbol{w}_k, \{(\boldsymbol{y}'(j), t_k(j)), j \in [1, n]\}) \tag{11}$$

as n is incremented. At each step, if a weight update is computed, the solution corresponds to the unique minimum of the newly refined surface. We can appreciate, therefore, that even if we neglect nonlinearities, the estimation processes behave quite differently with respect their error surface analysis.

The linearization technique adopted in this work can be explained in terms of the error surface analysis. The error surface over which we would like to find the (global) minimum by choice of weights is given by (7). Suppose we wish to construct a "linearized" error surface, say $\bar{E}_k$, which is "similar" in some sense to E_k in a neighborhood of the present weights. Recalling that E_k is a function not only of the weights, but also of the training patterns, the fundamental question is: *Can the pairs $\{(\boldsymbol{y}'(n), \boldsymbol{t}(n)), n \in [1, N]\}$ be modified in some sense, say $(\boldsymbol{y}'(n), \boldsymbol{t}(n)) \rightarrow (\bar{\boldsymbol{y}}'_k(n), \bar{t}_k(n))$, so that*

$$\begin{aligned} \bar{E}_k(\boldsymbol{w}_k, \{(\bar{\boldsymbol{y}}'_k(n), \bar{t}_k(n)), n \in [1, N]\}) &= \sum_{n=1}^{N} [\bar{t}_k(n) - \boldsymbol{w}_k^T \bar{\boldsymbol{y}}'_k(n)]^2 \\ \approx E_k(\boldsymbol{w}_k, \{(\boldsymbol{y}'(n), t_k(n)), n \in [1, N]\}) &= \sum_{n=1}^{N} [t_k(n) - S(\boldsymbol{w}_k^T \boldsymbol{y}'(n))]^2 \end{aligned} \tag{12}$$

in some neighborhood of the present weights? The answer to this question is the key theoretical development described in the following paragraphs.

In the ensuing discussion, the notation $\boldsymbol{w}_k^*$ will be used to designate a *local* minimum of E_k. Ideally, $\boldsymbol{w}_k^*$ will be the *global* minimum, but we have no way to assure this. The objective is to find, by a "linear" algorithm, a close approximation to $\boldsymbol{w}_k^*$.

The algorithm to be described proceeds in iterations, indexed by $i = 1, 2, \ldots$. Each iteration represents one complete training cycle through the N training patterns. Suppose that a weight vector estimate $\boldsymbol{w}_k(i-1)$ results from iteration $i-1$. In iteration i, by manipulation of the data, we work with a "linearized" error surface which is similar to the nonlinear surface in the neighborhood of $\boldsymbol{w}_k(i-1)$. The similarity follows from two criteria:

1. $\bar{E}_k(\boldsymbol{w}_k(i-1), \{(\bar{\boldsymbol{y}}'_k(n), \bar{t}_k(n)), n \in [1, N]\}) = E_k(\boldsymbol{w}_k(i-1), \{(\boldsymbol{y}'_k(n), t_k(n)), n \in [1, N]\})$;

2. $\left.\frac{\partial E_k}{\partial \boldsymbol{w}_k}\right]_{\boldsymbol{w}_k = \boldsymbol{w}_k(i-1)} = \left.\frac{\partial E_k}{\partial \boldsymbol{w}_k}\right]_{\boldsymbol{w}_k = \boldsymbol{w}_k(i-1)}$.

The first task is to manipulate the pairs $\{(\bar{\boldsymbol{y}}'_k(n), \bar{t}_k(n)), n \in [1, N]\}$ so that these criteria hold. This is accomplished as follows. It follows from Criterion 1 that

$$\sum_{n=1}^{N}(t_k(n) - y_k(n))^2 = \sum_{n=1}^{N}(\bar{t}_k(n) - \boldsymbol{w}_k^T(i-1)\bar{\boldsymbol{y}}'_k(n))^2. \tag{13}$$

By letting

$$t_k(n) - y_k(n) = \bar{t}_k(n) - \boldsymbol{w}_k^T(i-1)\bar{\boldsymbol{y}}'_k(n), \tag{14}$$

or

$$\bar{t}_k(n) = (t_k(n) - y_k(n)) + \boldsymbol{w}_k^T(i-1)\bar{\boldsymbol{y}}'_k(n), \tag{15}$$

for each n, Criterion 1 is met. Now we take the partial derivatives required in Criterion 2. For the "nonlinear" error,

$$\begin{aligned}\left.\frac{\partial E_k}{\partial \boldsymbol{w}_k}\right]_{\boldsymbol{w}_k = \boldsymbol{w}_k(i-1)} &= -2\sum_{n=1}^{N}(t_k(n) - y_k(n))\dot{S}(u_k(n))\boldsymbol{y}'(n) \\ &= -2\sum_{n=1}^{N}(t_k(n) - y_k(n))\dot{S}(\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n))\boldsymbol{y}'(n)\end{aligned} \tag{16}$$

where

$$\dot{S}(u_k(n)) \overset{\text{def}}{=} \frac{dS(\alpha)}{d\alpha}\Big]_{\alpha=u_k(n)}. \tag{17}$$

All inputs and outputs in this and similar expressions are those associated with weights $\boldsymbol{w}_k(i-1)$ (or the "current" set of weights around which linearization is taking place), but we will avoid writing $u_k(i-1,n)$, for example, for simplicity. For the "linear" error,

$$\frac{\partial \bar{E}}{\partial \boldsymbol{w}_k}\Big]_{\boldsymbol{w}_k=\boldsymbol{w}_k(i-1)} = -2\sum_{n=1}^{N}(\bar{t}_k(n) - \boldsymbol{w}_k^T(i-i)\bar{\boldsymbol{y}}'_k(n))\bar{\boldsymbol{y}}'_k(n). \tag{18}$$

Equating (16) and (18), in light of (14) we have

$$\bar{\boldsymbol{y}}'_k(n) = \dot{S}(\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n))\boldsymbol{y}'(n). \tag{19}$$

All quantities needed to compute the modified pair $(\bar{t}_k(n), \bar{\boldsymbol{y}}'_k(n))$ are known or can be calculated at pattern n. This procedure is repeated for each k (output node).

Before extending the analysis down to layer one, let us ponder the significance of what we have done. By modifying the data pairs, we have created a "linear" error surface which is similar to the "nonlinear" one in the neighborhood of $\boldsymbol{w}_k(i-1)$. In particular, the error surfaces match at that point, and their gradients are identical with respect to the weight vectors. We can find the $\boldsymbol{w}_k$ which minimizes $\bar{E}_k$ by simple *linear* least squares processing of the modified data $\{(\bar{t}_k(n), \bar{\boldsymbol{y}}'(n)),\ n \in [1, N]\}$. The linear estimate will correspond to a minimum of the error surface $\bar{E}_k$ which need not be near a minimum of E_k. However, because the error surfaces and the gradients match with respect to the weight vector of node k, if the weight change is small enough, the weight change will be in the direction of decreasing E_k. Accordingly, the linear weights must be constrained to remain in a reasonably small neighborhood of $\boldsymbol{w}_k(i-1)$. Because E_k is reduced at each iteration, it is to be expected that a minimum of E will be reached by repeating this procedure. In turn, this implies convergence to the "nonlinear" solution for the weights, using purely linear techniques.

Let us now move down to the lower layer and consider the estimation of the weights $\left\{ \boldsymbol{w}'_j \quad j \in [1, N_1] \right\}$. By similar reasoning to the above, we may focus on a single node, say node l. However, we must now optimize $\boldsymbol{w}'_l$ with respect to the *entire* external error, E, since all nodes in the upper layer are affected by these weights. Suppose that we are working on the i^{th} cycle through the training patterns and that all weights in the upper layer are fixed at their newly updated values $\{\boldsymbol{w}_k(i),\ k \in [1, N_2]\}$. Taking the derivative of E with respect to $\boldsymbol{w}'_l$,

$$\frac{\partial E}{\partial \boldsymbol{w}'_l} = -2 \sum_{n=1}^{N} \sum_{j=1}^{N_2} (t_j(n) - y_j(n)) \dot{S}(u_j(n)) w_{j,l}(i) \dot{S}([\boldsymbol{w}'_l]^T \boldsymbol{x}(n)) \boldsymbol{x}(n) \qquad (20)$$

where $w_{j,l}(i)$ denotes l^{th} element in weight vector $\boldsymbol{w}_j(i)$ (weight on connection from node l in layer one to node j in layer two). This expression can be written

$$\frac{\partial E}{\partial \boldsymbol{w}'_l} = -2 \sum_{n=1}^{N} (t'_l(n) - y'_l(n)) \dot{S}([\boldsymbol{w}'_l]^T \boldsymbol{x}(n)) \boldsymbol{x}(n) \qquad (21)$$

where $t'_l(n)$ is called the *target value* for inner node l and is defined such that

$$(t'_l(n) - y'_l(n)) = \sum_{j=1}^{N_2} (t_j(n) - y_j(n)) \dot{S}(u_j(n)) w_{j,l}(i). \qquad (22)$$

The quantity on the right side of (22) is commonly called the *back-propagated error* for node l. The solution sought, say $\boldsymbol{w}'^{*}_l$, is one for which

$$\left. \frac{\partial E}{\partial \boldsymbol{w}'_l} \right]_{\boldsymbol{w}'_l = \boldsymbol{w}'^{*}_l} = 0. \qquad (23)$$

In the top layer, for node k we sought $\boldsymbol{w}_k^*$ such that

$$\left.\frac{\partial E}{\partial \boldsymbol{w}_k}\right]_{\boldsymbol{w}_k=\boldsymbol{w}_k^*} = 0. \tag{24}$$

With reference to (16), it is clear that the present optimization problem is equivalent to the ones encountered at the upper nodes. In particular, the same linearization considerations can be applied to obtain modified input and target values, say

$$(t_l'(n), \boldsymbol{x}(n)) \rightarrow (\bar{t}'_l(n), \bar{\boldsymbol{x}}_l(n)) \tag{25}$$

and the set of layer one weights $\boldsymbol{w}_l'(i)$ computed accordingly for each l.

Before continuing, let us note the relationship to the A-S algorithm noted above. In fact, to this point in the discussion, the methods are nearly equivalent though derived from different starting points. In the i^{th} iteration through the training patterns, prior to updating the weights $\boldsymbol{w}_k$ at pattern n, node k is "linearized" in the A-S algorithm by approximating $S(\alpha)$ by a linear function which is tangential to[3] S at $\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n)$. In effect, $S(\alpha)$ is approximated by the first two terms of a Taylor series,

$$\begin{aligned} S(\alpha) &\approx \bar{S}(\alpha) = \dot{S}(\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n))(\alpha - \boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n)) + S(\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n)) \qquad (26) \\ &= \dot{S}(\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n))\alpha + [S(\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n)) - \dot{S}(\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n))\boldsymbol{w}_k^T(i-1)\boldsymbol{y}'(n)] \\ &\overset{\text{def}}{=} K_k(n)\alpha + b_k(n). \end{aligned}$$

Azimi-Sadjadi *et al.* [2] recognized that by using this approximation in (16), the optimization

[3] In fact, if $w(i, n-1)$ denotes the weight estimates after pattern $n-1$ in iteration i, then in the A-S algorithm, $S(\cdot)$ is linearized around *these* weights rather than the weights at the end of the previous cycle. Of course, this process could also be used in our algorithm, but we find that it makes no significant difference, and the computational expense of updating the weights at each n is avoided in our case.

problem became equivalent to a set of linear least square error normal equations *if* the data were modified according to (15) and (19). Therefore, by quite different means, the theoretical developments arrive at the same set of linear equations to be solved.

In principle, once the linearization is achieved at iteration i and pattern n, any least mean square type algorithm can be employed to update the weight estimates. The A-S method uses the conventional RLS algorithm. In this case, neglecting any error weighting, RLS takes the form of the two recursions (written for node k in the top layer) [11, Ch.5],

$$\boldsymbol{P}(i,n) = \boldsymbol{I} - \frac{\boldsymbol{P}(i,n-1)\bar{\boldsymbol{y}}'_k(n)[\bar{\boldsymbol{y}}'_k(n)]^T\boldsymbol{P}(i,n-1)}{1+[\bar{\boldsymbol{y}}'_k(n)]^T\boldsymbol{P}(i,n-1)\bar{\boldsymbol{y}}'_k(n)} \tag{27}$$

$$\boldsymbol{w}_k(i,n) = \boldsymbol{w}_k(i,n-1) + \boldsymbol{P}(n)\bar{\boldsymbol{y}}'_k(n)[\bar{t}_k(n) - [\bar{y}_k(n)]. \tag{28}$$

$\boldsymbol{w}_k(i,n)$ is the estimate of the weights $\boldsymbol{w}_k$ following pattern n in the i^{th} iteration through the training data, and $\boldsymbol{P}^{-1}(i,n)$ is the covariance matrix at the same "time" in the process,

$$\boldsymbol{P}^{-1}(i,n) \overset{\text{def}}{=} \sum_{j=1}^{n} \bar{\boldsymbol{y}}'_k(j)[\bar{\boldsymbol{y}}'_k(j)]^T. \tag{29}$$

Note that $\boldsymbol{w}_k(i,0) \overset{\text{def}}{=} \boldsymbol{w}_k(i-1,N)$ and similarly for the covariance matrix. This presents the question of how $\boldsymbol{w}_k(0,0)$ and $\boldsymbol{P}(0,0)$ should be initialized. The inverse covariance matrix contains theoretically infinite values at the outset and a proper initialization for the weights is practically not known (this means that the initial linearizations of the training data are based on potentially very bad weight estimates). This issue will be addressed further below. Also, it is clear that this solution, as written, will continue to "accumulate" past linearized sets of data which might, in fact, be linearized around very poor weight estimates. Therefore, the A-S algorithm includes a "forgetting factor" [11] in the RLS recursions. This is equivalent to using a weighted error criterion with time varying (exponentially decaying) weights. This can make convergence slow if the forgetting factor is large. If the forgetting factor is small, then past values are forgotten more quickly, but leads to convergence problems. We will also comment further on this issue below.

We have found that the choice of conventional RLS as a solution method seriously impairs the ability of this linearization method to converge on a proper set of network weights. As an alternative, therefore, we suggest the method presented in the following section.

3 Solution by QR Decomposition

In order to improve convergence the algorithm developed above can be implemented using QR decomposition [1, 8]. This algorithm has distinct advantages over conventional RLS. First, the QR algorithm does not suffer from initialization problems noted above for RLS. It also permits the inclusion of several very flexible "forgetting" strategies. To illustrate the operation of the algorithm, it is sufficient to consider the estimation of weights $\boldsymbol{w}_k$ in the output layer of the network. All notation is consistent with that used above except the number of nodes in layer one is denoted M.

In effect, the linearization technique described above reduces the problem at the i^{th} iteration through the training patterns to one of finding the least square error solution of the overdetermined system of equations

$$\begin{bmatrix} (\bar{\boldsymbol{y}}'_k(1))^T & \rightarrow \\ (\bar{\boldsymbol{y}}'_k(2))^T & \rightarrow \\ \vdots & \vdots \\ (\bar{\boldsymbol{y}}'_k(N))^T & \rightarrow \end{bmatrix} \boldsymbol{w}_k(i) = \begin{bmatrix} \bar{t}_k(1) \\ \bar{t}_k(2) \\ \vdots \\ \bar{t}_k(N) \end{bmatrix} . \qquad (30)$$

The QR decomposition method is based upon transforming this system into an upper triangular system by applying a series of orthonormal operators (Givens rotations). The resulting system is

$$\left[\begin{array}{c} \boldsymbol{T}(N) \\ \hline \mathbf{0}_{(N-M)\times M} \end{array}\right] \boldsymbol{w}_k(i) = \left[\begin{array}{c} \boldsymbol{d}_1(N) \\ \hline \boldsymbol{d}_2(N) \end{array}\right] \tag{31}$$

where the matrix $\boldsymbol{T}(N)$ is $M \times M$ upper triangular and $\mathbf{0}_{i\times j}$ denotes the $i \times j$ zero matrix. The solution for $\boldsymbol{w}_k(i)$ is easily obtained by back-substitution. A recursive version of the solution is also possible. The recursive algorithm is shown in Fig. 2. For details the reader is referred to [1].

For discussion of further benefits of the decomposition algorithm, it is useful to view the $\boldsymbol{A}$ matix, defined in Fig. 2, as four partitions. Following the rotation of the n^{th} equation, in Step 2, for example,

$$\boldsymbol{A} = \left[\begin{array}{c|c} \boldsymbol{T}(n) & \boldsymbol{d}_1(n) \\ \hline \mathbf{0}_{1\times M} & d_2(n) \end{array}\right] \tag{32}$$

As is the case with the A-S method, a forgetting factor must be employed to gradually reduce the effects of earlier linearizations. This is very easily accomplished in the QR algorithm by simply multiplying the top M rows of the matrix $\boldsymbol{A}$ (matrix $\boldsymbol{T}(n)$ and vector $\boldsymbol{d}_1(n)$) by a factor $\beta < 1$ prior to the rotation of the $n + 1^{st}$ pattern equation. In this context, both the forgetting factor and the frequency of weight updates can be varied. In addition to exponential forgetting factors, equations can be "rotated out" of the matrix. This is done by changing $\mathcal{S}$ in Fig. 2 to -1 and rotating in the equation to eliminate. Thus, for example, only the last $Q > M$ equations can be used to calculate the weight updates by sequentially removing equation $n - Q + 1$ prior to inclusion of equation n. This procedure effects a sliding window over which the estimates are computed. Another forgetting method useful for MLPs is possible because no initialization of the updating equations is necessary. Because there are no initialization problems the system can be re-initialized at any step, thus completely "forgetting" the past linearized values. These and a number of other flexible forgetting strategies made possible by this algorithm may prove very useful in the training of MLPs.

In addition to new forgetting factors, using the QR implementation also allows the frequency of updating of the weights to vary. As with conventional RLS, the weights can be updated every time a new linearization has been used[4].

The theoretical results above, along with those in Section 2, can be combined to form a learning algorithm for MLPs. First, the weights of the network must be initialized. This is done randomly, each weight being selected from a uniform distribution over the set [-1,1]. Once the initial weights are chosen, the weight updating can begin. First, a training pattern is input to the system. Because the weights are not updated until all the training patterns have been used, convergence does not depend on the order in which the training patterns are used. Given a training pattern, the algorithm calculates linearized training patterns for the last layer nodes and these are rotated into the corresponding $\boldsymbol{A}$ matrix. Each node has a "separate" $\boldsymbol{A}$ matrix. The target outputs of the layer below are calculated next using back- propagation. The $\boldsymbol{A}$ matrices for the first layer are then updated. A new training pattern is then used to calculate a new set of linearized inputs and outputs. This is repeated until all the training patterns have been used. The $\boldsymbol{A}$ matrices are then used to calculate updated weights. This continues until the network converges to a solution. By definition the solution is said to have converged when the change of the norm of the vector of all the weights is below a threshold. As with other training algorithms for MLPs, this algorithm may not converge to the weights corresponding to the *global* minimum of the function of E. Also, although the algorithm approximates a gradient system, because it is not a gradient system, there is no

[4] This can be as often as every pattern (see Footnote 3), or at the end of each iteration through the patterns as has been our convention.

guarantee that the algorithm will converge to any solution. Thus for implementation a maximum is placed on the number of iterations.

4 Complete Layer and Network Updating

The back-propagation algorithm updates each weight at each node individually. All the weights in the network except one are fixed and this is changed to reduce E. The algorithm described in the previous section updates *all the weights connected to one node* simultaneously. All the weights in the network except *those connected to one node* are fixed, and those weights are updated to reduce E. These two methods of updating weights may not be optimal because E is a function of all the network weights and may not be minimized by updating the weights of each node independently. Minimizing an error implies there is a target value. There are no given target values for the inner layers so these are computed assuming the weights of the layers above are fixed. These target values allow the weights of each node to be updated independently. This makes the computations easier, but does not take into consideration the interdependence of the nodes.

The next step in the development is to demonstrate how to update all the weights connected to one *layer* simultaneously.

The following derivation uses the linearization of the nonlinearity suggested by (26). Note that

$$y_k(n) = S(\boldsymbol{w}_k^T \boldsymbol{y}'(n)) = S(\sum_{j=1}^{N_1} w_{k,j} y'_j(n)) \tag{33}$$

and

$$y'_j(n) = S([\boldsymbol{w}'_j]^T \boldsymbol{x}(n)) = S(\sum_{l=1}^{N_0} w'_{j,l} x_l(n)). \tag{34}$$

The linearization replaces $S(u)$ by $Ku + b$. Thus

$$\bar{y}_k(n) = K_k(n)(\sum_{j=1}^{N_1} w_{k,j}y'_j(n)) + b_k(n) \tag{35}$$

and

$$\bar{y}'_j(n) = K'_j(n)(\sum_{l=1}^{N_0} w'_{j,l}x_l(n)) + b'_j(n), \tag{36}$$

so

$$\bar{y}_k(n) = K_k(n)(\sum_{j=1}^{N_1} w_{k,j}[K'_j(n)(\sum_{l=1}^{N_0} w'_{j,l}x_l(n)) + b'_j(n)]) + b_k(n). \tag{37}$$

To update the weights in layer two, the weights in layer one of the network are fixed. With these weights fixed, the weights connected to different nodes in the output layer can be updated independently and the same update equations as in the previous section result. To update the weights in the first layer, the weights in the last layer are fixed. Thus

$$\bar{y}_k(n) = K_k(n)\sum_{j=1}^{N_1} w_{k,j}K'_j(n)\sum_{l=1}^{N_0} w'_{j,l}x_l(n) + K_k(n)\sum_{j=1}^{N_1} w_{k,j}b'_j + b_k(n), \tag{38}$$

or

$$\bar{y}_k(n) = \sum_{j=1}^{N_1}\sum_{l=1}^{N_0}(K_k(n)w_{k,j}K'_j(n)x_l(n))w'_{j,l} + [\sum_{j=1}^{N_1}(K_k(n)w_{k,j}b'_j + b_k(n)] \tag{39}$$

so

$$\bar{y}_k(n) - [\sum_{j=1}^{N_1} K_k(n)w_{k,j}b'_j(n) + b_k(n)] = \sum_{j=1}^{N_1}\sum_{l=1}^{N_0}(K_k(n)w_{k,j}K'_j(n)x_l(n))w'_{j,l}. \tag{40}$$

Hence for one output, this is the same as a linear system with $N_0 \times N_1$ inputs and one output. The linearized output and inputs are

$$\bar{t}_k(n) = t_k(n) - \sum_{j=1}^{N_1}[K_k(n)w_{k,j}b'_j(n)] + b_k(n), \tag{41}$$

and

$$\bar{x}_{j,l} = K_k(n)w_{k,j}K_j'(n)x_l(n). \tag{42}$$

For one output, $y_k(n)$, and N training patterns,

$$E_k = \sum_{n=1}^{N}(t_k(n) - y_k(n))^2, \tag{43}$$

while for N_2 outputs and one training pattern

$$E = \sum_{k=1}^{N_2}(t_k(n) - y_k(n))^2, \tag{44}$$

where $y_k(n)$ has the form above. The weights in the output layer are held constant, so all the y_k are a function of all the weights in the first layer. Because the $y_k(n)$ for all k and for all n are functions of the same weights, we can use the same technique to update the weights of the two systems above, given by (43) and (44). Thus, the N_2 output network with one training pattern is treated as a one output network with N_2 training patterns. This is done for each training pattern. Thus if there are N training patterns and N_2 outputs, the number of linearized training patterns is $N \times N_2$. This method allows us to update all the weights in the same layer simultaneously.

Ultimately the goal is to update all the weights of the network simultaneously. Some improvement in convergence can be expected because the weights are not independent.

Simultaneous updating of *all* weights can easily be accomplished for a one output network using the derivation above. From (38)

$$y_k(n) = \sum_{j=1}^{N_1}\sum_{l=1}^{N_0}[K_k(n)K_j'(n)x_l(n)]w_{k,j}w_{j,l}' + \sum_{j=1}^{N_1}[K_k(n)b_j'(n)]w_{k,j} + b_k(n). \tag{45}$$

Letting

$$w^{\dagger}_{j,l} = w_{k,j} w'_{j,l}, \tag{46}$$

then

$$y_k(n) = \sum_{j=1}^{N_1} \sum_{l=1}^{N_0} [K_k(n) K'_j(n) x_l(n)] w^{\dagger}_{j,l} + \sum_{j=1}^{N_1} [K_k(n) b'_j(n)] w_{k,j} + b_k(n) \tag{47}$$

or

$$(y_k(n) - b_k(n)) = \sum_{j=1}^{N_1} \sum_{l=1}^{N_0} [K_k(n) K'_j(n) x_l(n)] w^{\dagger}_{j,l} + \sum_{j=1}^{N_1} [K_k(n) b'_j(n)] w_{k,j}. \tag{48}$$

This is a linear system with one output and $N_0 \times N_1 + N_1$ inputs. The system can be solved for $w_{k,j}$ and $w^{\dagger}_{k,j}$ and (46) can be used to solve for $w'_{j,l}$.

5 Experimental Results

5.1 Single Node Updating

The results given in this section compare three training strategies for an MLP. These are:

1. Conventional back-propagation (no linearization).

2. Conventional RLS with a forgetting factor (A-S Algorithm).

3. QR decomposition with an exponential forgetting factor.

Each of the three strategies was used to train each of the following networks:

1. a two-bit parity checker,

2. a four-bit parity checker, and

3. a four-bit bit counter.

The architectures for these three networks are illustrated in Fig. 3.

The two-bit parity checker (XOR) network has two inputs, two hidden layer nodes and one output node. An additional input is added at each layer whose value was always unity, to serve as a bias for each node. The output function $S(\cdot)$ is the sigmoid defined in (3). The initial weights were chosen as follows. Each weight in the network was selected randomly from a uniform distribution over the set $[-1, 1]$. This procedure was repeated 100 times to select 100 sets of initial weights. The same 100 sets of weights were used for all 3 implementations. For the back-propagation algorithm, a factor of 0.04 was used in the weight updating equation. The A-S algorithm was implemented [2] using no weight change constraints. The forgetting factor for this and for the QR decomposition implementation was 0.98. The QR decomposition implementation used a weight constraint of 0.2. Thus the weight vector associated with each node was allowed to change at most by 0.2 during each iteration.

The four-bit parity checker network has four inputs, four hidden layer nodes and one output node. A bias input is also added to each layer. Two output functions were used. These were the same sigmoid function as above, and the logic activation function. The logic activation function is a three piece piecewise linear function. It is zero at zero, has slope one from zero to one, and slope zero everywhere elsewhere. This makes the derivative of $S(\cdot)$ easy to determine everywhere except at zero and one where it does not exist. This does not pose a problem in implementation if we let $\dot{S}(\alpha) = 1$ if $\alpha \in [0, 1]$ and zero else. Two sets of 100 random initial weights were used for the three implementations. The first set of weights was random as in the two-bit parity checker, and the second set of weights was as described by Azimi-Sadjadi *et al.* in their paper. The A-S method selects the weights so that the outputs of the network will be between zero and one. This is done so that the derivative will not be zero and weight updating can take place. The four-bit bit counter had four inputs, four hidden layer nodes and two outputs. An extra input was added to each layer. The logic activation function was used as the output function. Two sets of initial weights, random and as described by Azimi-Sadjadi, were used. The results are shown in Table 1. The table shows the number of times each implementation found weights that solved the problem for the 100 initial weight sets.

Simulations were also run comparing the output error of each algorithm. In the resulting figures, the error in dB means the following: Let $\varepsilon(i)$ be the sum of the squared errors incurred in iteration i through the training patterns, averaged over the 100 initial weight sets. Then, plotted in the figures is $10\log(\varepsilon(i)/\mu)$ (dB), where μ is the maximum possible error in any iteration.

Fig. 4 shows the errors of the three X-OR implementations. Fig. 5 shows the errors of the QR decomposition algorithm using different forgetting factors and different weight constraints. The number of convergences for each of the setting was 78 for number one, 60 for number two, 56 for number three and 64 for number four. It is apparent that the parameters which yield the most convergences do not necessarily lead to the lowest average error.

These results indicate a clear advantage for the QR decomposition strategy. Algorithmic differences among the three implementations account for performance differences. One difference is the initialization needed for the RLS equations. With the RLS strategy, both the covariance matrix recursion and the weight vector recursion must be initialized using theoretically incorrect values. Because of initializations, the RLS algorithm is not guaranteed to move the estimate in the direction of *greatest* decrease of E, or even of decreasing E, for the first iterations. Of the two RLS recursions, the weight recursion seems to be the most sensitive to the initialization problem. This is because $\boldsymbol{P}$ is initialized with large values, $\boldsymbol{P}^{-1}$ is small and the effect of this initialization is relatively small. The weight recursion is sensitive to initialization because (28) depends explicitly upon $w_k(i, n-1)$. The QR algorithm has only an implicit dependence on the weights, as do all linearization algorithms, because the linearizations depend on the weights.

There is also a difference in the performance of the network using different functions for $S(\cdot)$. The logic activation function proved superior to the sigmoid in these experiments. This is probably because the error will always be positive using the sigmoid, but can be zero for the threshold logic activation function. No matter how the weights are adjusted, the output of the sigmoid will always be bounded by one, so that the training pattern outputs can never be matched exactly. With the threshold logic activation function, once the weights are adjusted so that the output is off the ramp (the linear region), the output can be zero or one in which case the difference between the training output and the actual output can be zero.

5.2 Layer Updating and Network Updating

This section gives the results for the algorithms given in Section 4. The first algorithm updates the network weights by layers. All the weights in the same layer are updated simultaneously. The second algorithm updates all the weights in the network simultaneously. Both algorithms were used to train a two-bit parity checker (XOR) network. The same network architecture and the same 100 sets of initial weights as in the previous section were used in the simulations. The layer-wise updating algorithm has a forgetting factor of 0.3 and a weight constraint of 1.0. Thus the vector of the weights in each layer was allowed to change by at most 1.0 during any iteration. The network-wise updating algorithm had the same forgetting factor and weight constraint. The results are shown in Table 2. As in Table 1, Table 2 shows the number of times each algorithm found weights that solved the problem for the 100 initial weight sets.

Fig. 6 shows the errors of the two X-OR implementations of this section and the error of the QR decomposition implementation of the single node updating algorithm. The convergence results show the advantage of layer-wise weight updating and network-wise weight updating over node-wise updating. Layer-wise weight updating also proved better in the error analysis

6 Conclusions

A new implementation of a node-wise weight updating algorithm for multilayer perceptrons and new algorithms that update weights layer-wise and network-wise have been presented in this paper. The QR decomposition implementation has been shown to be superior to standard recursive equations for the node-wise updating algorithm. This result should prove to be beneficial not only for this algorithm, but for all MLP training algorithms that use recursive equations for implementation. The layer-wise and network-wise weight updating algorithms were developed to improve the convergence rate and the speed of convergence. Both objectives were accomplished, with the layer-wise weight updating algorithm showing a significant advantage over both the single node weight updating algorithm used as a reference and standard back-propagation.

References

[1] J.R. Deller, Jr., and D. Hsu, "An alternative adaptive sequential regression algorithm and its application to the recognition of cerebral palsy speech," *IEEE Transactions on Circuits and Systems*, vol.CAS-34, no.7, pp.782-786, July 1987.

[2] M. Azimi-Sadjadi, S. Citrin, and S. Sheedvash, "Supervised learning process of multi-layer perceptron neural networks using fast least squares," *Proceedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp.1381-1384, 1990.

[3] S. Kollias and D. Anastassiou, "An adaptive least squares algorithm for the efficient training of artificial neural networks," *IEEE Transactions on Circuits and Systems*, vol. CAS-36, pp.1092-1101, August 1989.

[4] M. Azimi-Sadjadi and S. Citrin, "Fast leaning process of multi- layer neural nets using recursive least squares technique," *IEEE Int. Conf. on Neural Networks*, Washington D.C., June 1989.

[5] P. Werbos, *Beyond Regression: New Tool for Prediction and Analyses in the Behavioral Sciences* (Ph.D. dissertation), Harvard University, Cambridge, Massachusetts, 1974.

[6] D. Rumelhart, G. Hinton, and G. Williams, "Learning internal representations by error propagation," in D. Rumelhart and J. McCleland (editors), *Parallel Distributed Processing*, vol. 1 Cambridge, Massachusetts: MIT Press, 1986.

[7] R. Lippmann, "Review of neural networks for speech recognition," *Neural Computation*, vol. 1, pp. 1-38, 1989.

[8] G. Golub and C. VanLoan, *Matrix Computations*, Baltimore, Maryland: Johns Hopkins University Press, 1983.

[9] B. Kosko, *Neural Networks and Fuzzy Systems*, Englewood Cliffs, New Jersey: Prentice-Hall, 1992.

[10] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, New Jersey: Prentice-Hall, 1985.

[11] D. Graupe, *Time Series Analysis, Identification and Adaptive Filtering*, Malabar, Florida: Krieger Publishing Company, 1989.

Tables 1 & 2.

Table 1: Number of convergences per 100 sets of initial weights

Impementation	2in-1out	4in-1out				4in-2out	
	sigmoid	sigmoid		logic activation		logic activation	
	random weights	random weights	A-S weights	random weights	A-S weights	random weights	A-S weights
Q-R	78	5	5	51	57	1	16
Back-Prop	11	0	0	1	53	0	0
A-S	8	0	0	1	37	0	9

Table 2: Number of convergences per 100 sets of initial weights

Impementation	2in-1out
	sigmoid
	random weights
Layer-wise updating	96
Network-wise updating	99

Figure 1

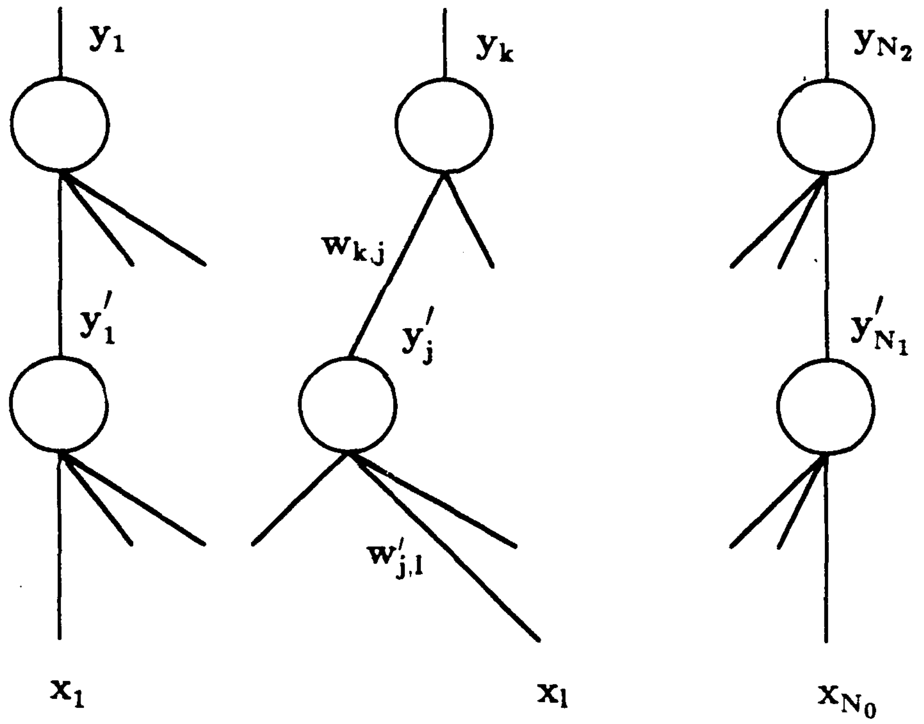


Figure 1: Two-layer perceptron. The bottom set of "nodes" (input) are linear and are designated layer zero.

Figure 2.

Figure 2: Weight Estimation Using Recursive QR Decomposition

Weight Estimation Using Recursive QR Decomposition

Initialization: Initialize an $(M + 1) \times (M + 1)$ working matrix, say $\boldsymbol{A}$, to a null matrix.

Recursion: For $i = 1, 2, \ldots$ (iteration); and, For $n = 1, 2, \ldots, N$ (pattern),

1. Enter the next equation into the bottom row of $\boldsymbol{A}$,

$$\left[\ [\bar{\boldsymbol{y}}'_k(n)]^T \ \middle|\ \bar{t}_k(n)\ \right]. \tag{49}$$

2. "Rotate" the new equation into the system using

$$\begin{aligned} A'_{mk} &= A_{mk}\sigma + A_{M+1,k}\tau\mathcal{S} \\ A'_{M+1,k} &= -A_{mk}\tau\mathcal{S} + A_{M+1,k}\sigma\mathcal{S} \end{aligned}$$

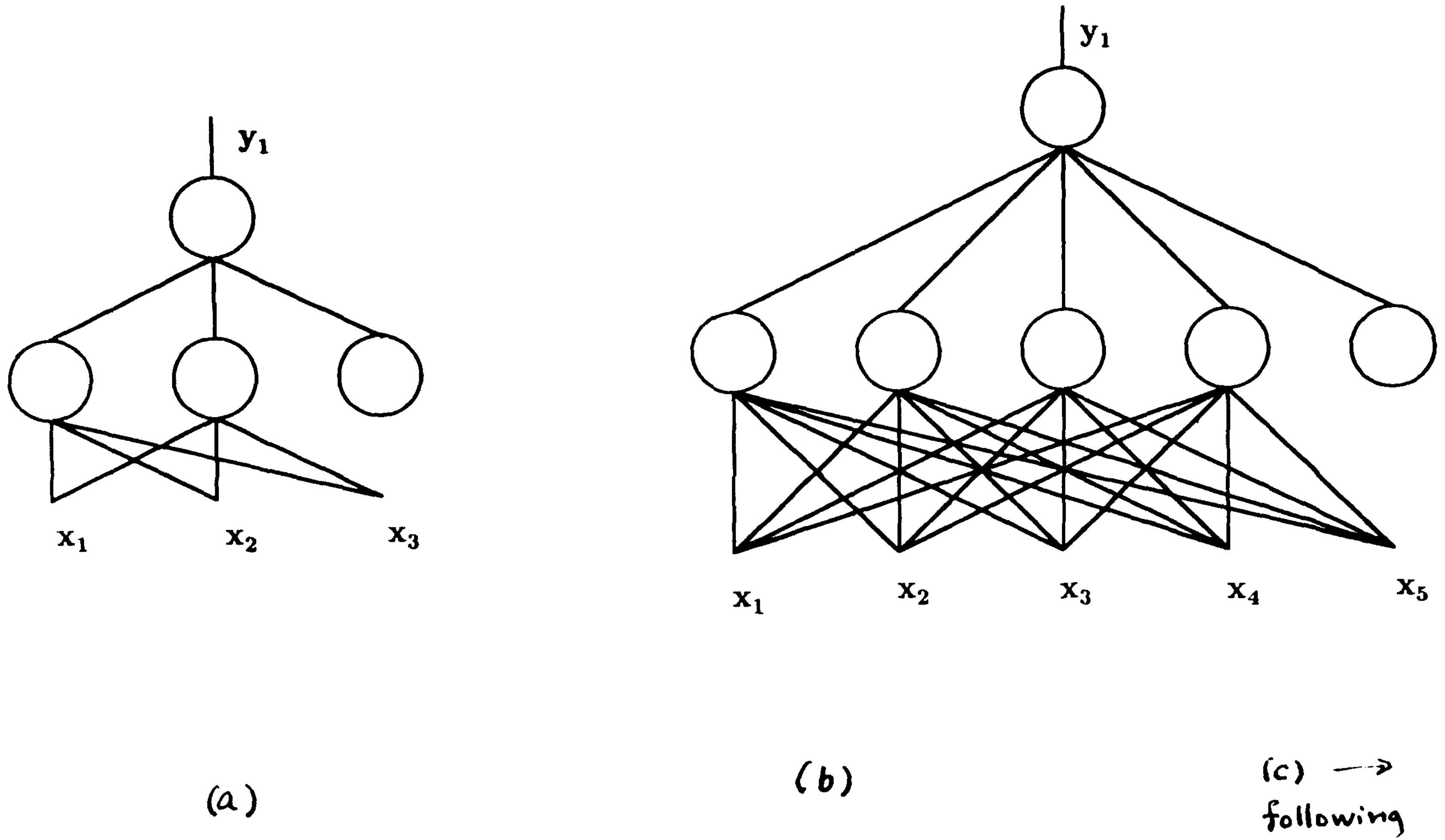
   for $k = m, m+1, \ldots, M+1$ and $m = 1, \ldots, M$; where $\sigma = A_{mm}/\rho$, $\tau = A_{M+1}/\rho$, $\rho = (A^2_{mm} + A^2_{M+1})^{1/2}$, $\mathcal{S}$ is unity (useful later), and $A_{mk}(A'_{mk})$ is the m, k element of $\boldsymbol{A}$ pre-(post-)rotation. No other elements of $\boldsymbol{A}$ are affected.

3. Solve for the least square estimate of the weights $\boldsymbol{w}_k$ if desired. (Solution after the n^{th} pattern will produce what has been called $\boldsymbol{w}_k(i, n)$ in the text, and $\boldsymbol{w}_k(i, N) = \boldsymbol{w}_k(i)$.)

4. If $n < N$, increment n. Otherwise check convergence criterion and increment i and reset n if not met.

Termination: Stop when some convergence criterion has been met.

Figures 3(a) & (b)



(c) → following page

Figure 3: Network architectures for (a) the two-bit parity checker, (b) the four-bit parity checker, and (c) the four-bit bit counter used in the simulation studies.

Figure 3(c)

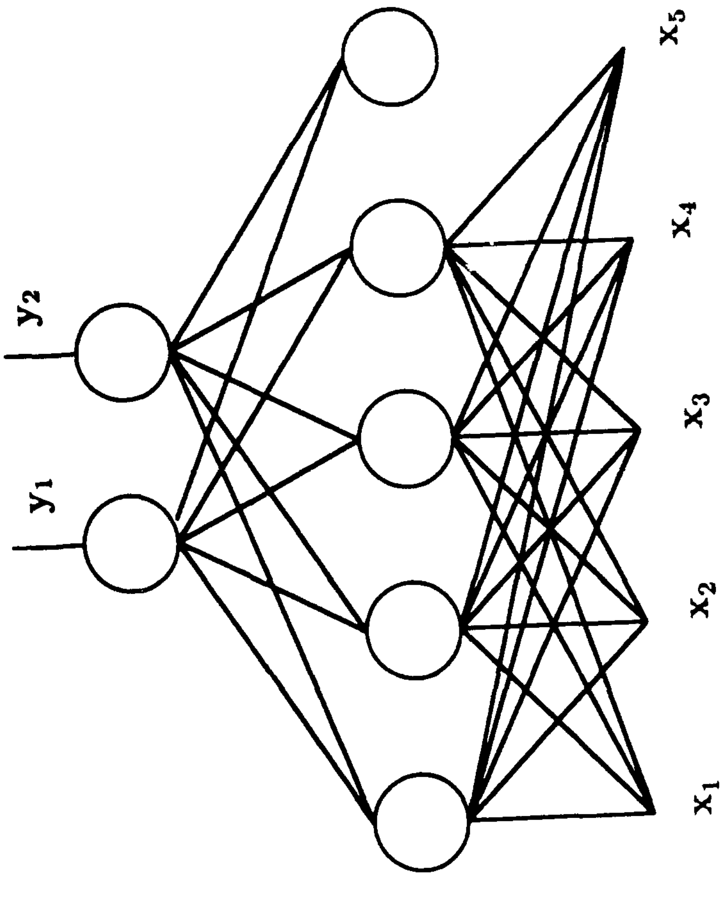


Figure 4.

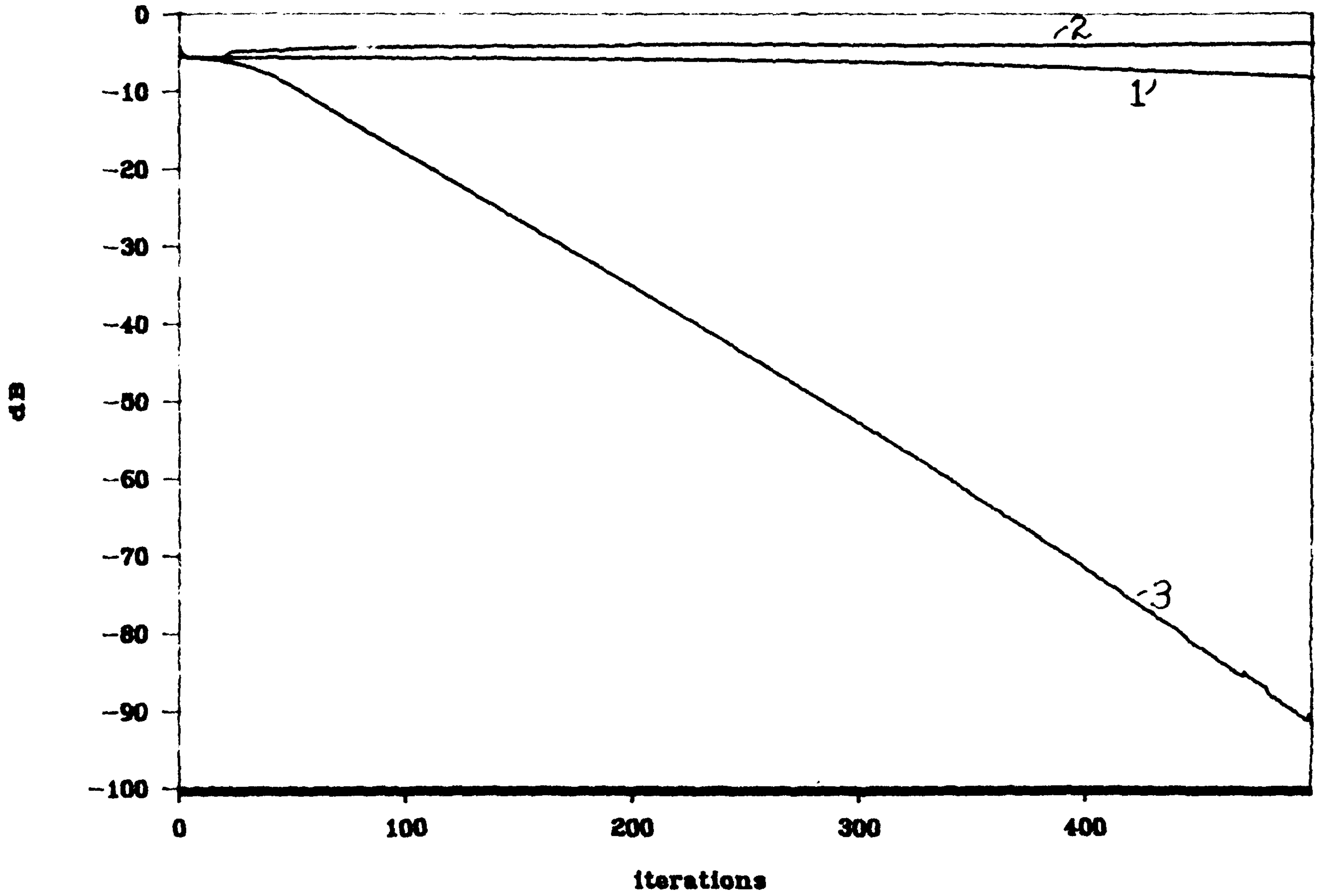


Figure 4: Average error in dB for the X-OR implementations vs. iteration *number*. 1.back propagation; 2.A-S algorithm; 3. QR *decomposition*.

Figure 5.

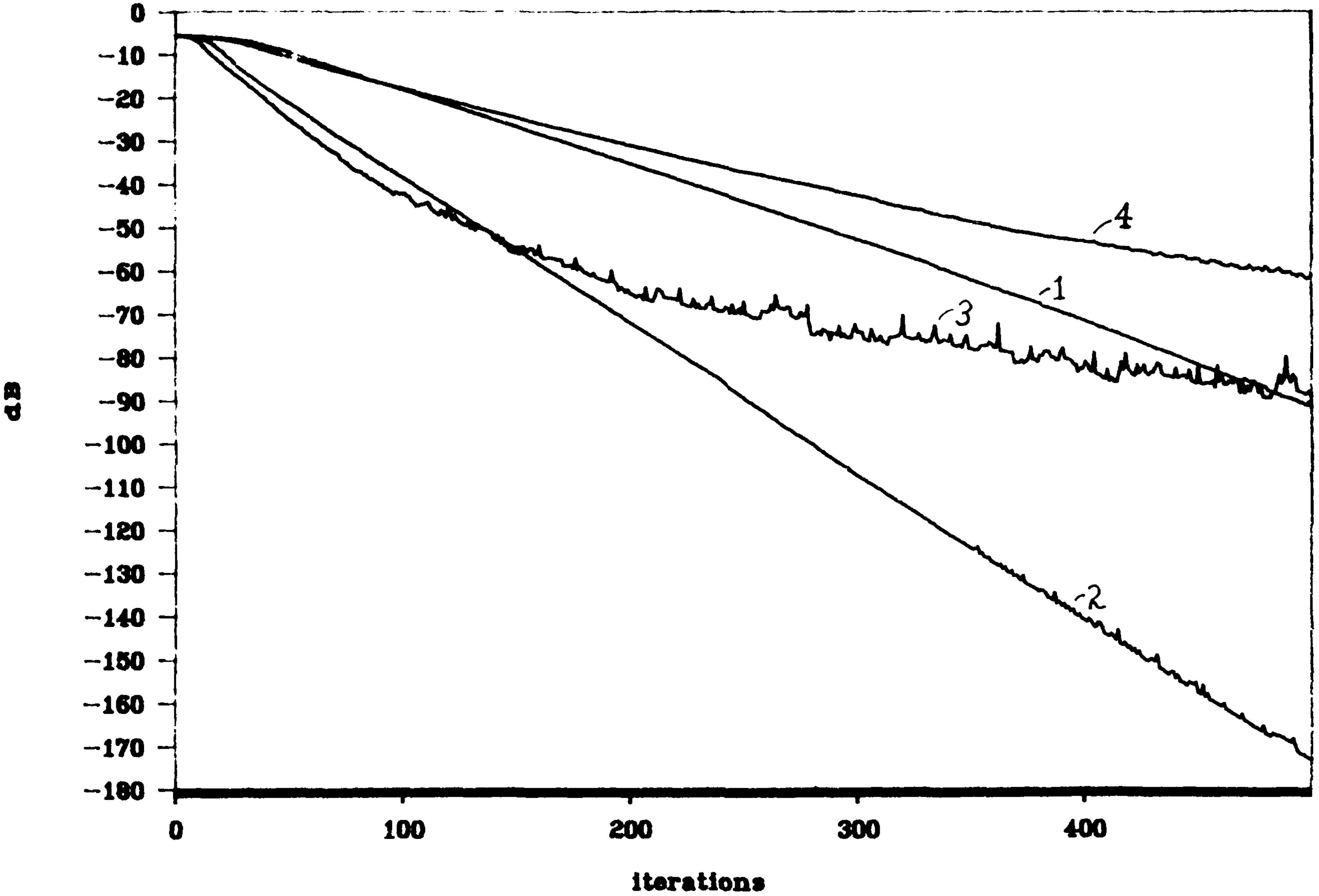


Figure 5: Average error in dB for the QR X-OR implementation vs. iteration number, using different forgetting factors and weight change constraints. 1. $\nu = 0.98, \gamma = 0.2$; 2. $\nu = 0.98, \gamma = 1.0$; 3. $\nu = 0.1, \gamma = 1.0$; 4. $\nu = 0.1, \gamma = 0.2$; where ν is the forgetting factor and γ is the weight constraint.

Figure 6

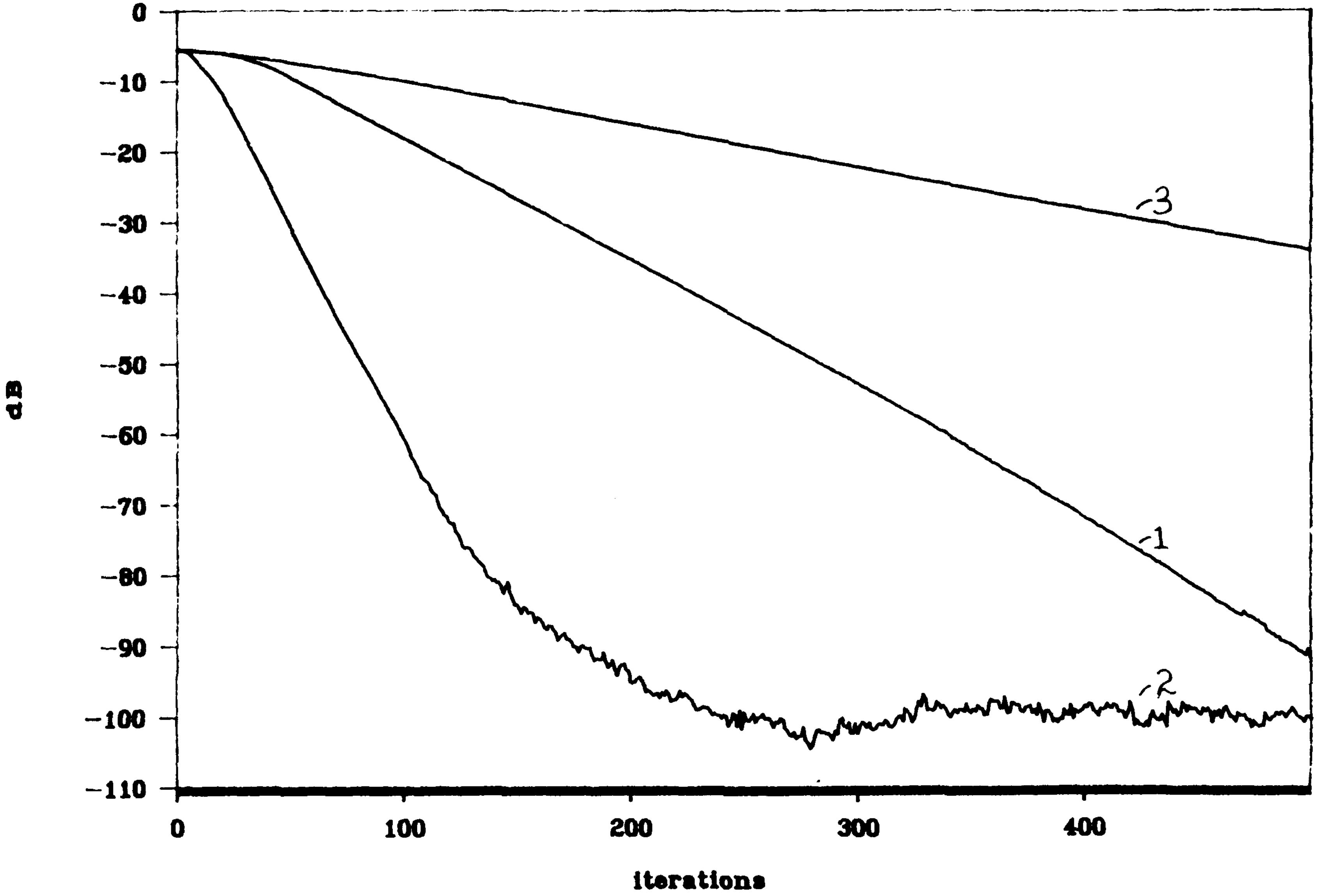


Figure 6: Average error in dB for the X-OR algorithms vs. iteration number. 1. Single node updating; 2. Layer updating; 3. Complete network updating.

æ

DRAFT MANUSCRIPT January 19, 1992

On the Connections Between the Fogel–Huang and Dasgupta–Huang OBE Identification Algorithms

J.R. Deller, Jr., M. Nayeri, and M.S. Liu

Michigan State University
Department of Electrical Engineering
CSSP Group: Speech Processing & Adaptive Signal Processing Laboratories
East Lansing, MI 48824-1226

email: deller@ee.msu.edu phone: (517) 353-8840 FAX: (517) 353-1980

Abstract

The Dasgupta–Huang Optimal Bounding Ellipsoid (OBE) algorithm for identifying linear parametric systems has been proven to converge under ordinary conditions on the model disturbances. This appealing property notwithstanding, the algorithm is based upon an unusual optimization criterion which makes behavior of the method difficult to interpret theoretically. On the other hand, the optimization strategy for the original Fogel–Huang OBE algorithm is appealing in its straightforward interpretability, but the method suffers from the lack of a clear understanding of its convergence properties. While the underlying bounded error assumption gives rise to both algorithms, the developments of the techniques are fundamentally very different. However, this note describes some interesting relationships between the algorithms which: 1. provide theoretical support and interpretation of the optimization criterion employed in Dasgupta–Huang; and, 2. suggest that an algorithm with the desirable properties of both algorithms may exist.

Acknowledgments

This work was supported by the Office of Naval Research under Contract No. N00014-91-J-1329 and by the National Science Foundation under Grant No. MIP-9016734.

1 Introduction

Set-membership-based (SM) system identification algorithms offer an interesting alternative to conventional techniques. SM methods have been receiving increasing attention internationally. Recent reviews of this field are found, for example, in [1]–[3]. This note is restricted to the class of algorithms known as *optimal bounding ellipsoid (OBE)* algorithms which follow from a *bounded error* constraint. We explore some interesting connections which exist between two landmark OBE algorithms – the Fogel-Huang (F-H) [4] and Dasgupta-Huang (D-H) [5] OBE algorithms – which have not been well appreciated. These connections suggest the possibility that the desirable properties of both may be blended into a single OBE algorithm.

The bounded error identification problem is as follows: Assume that we are observing some physical system which is generating sequence $\boldsymbol{y}(\cdot) \in \mathcal{C}^k$ in response to input $\boldsymbol{u}(\cdot) \in \mathcal{C}^l$. $\boldsymbol{u}(\cdot)$ is a realization of an ergodic, wide sense stationary stochastic process. Both input and output sequences are measurable. We assume the existence of a "true" model of form

$$\boldsymbol{y}(n) = \boldsymbol{\Theta}_*^H \boldsymbol{x}(n) + \boldsymbol{\varepsilon}_*(n) \tag{1}$$

in which $\boldsymbol{x}(n)$ is some m-vector of functions of p lags of $\boldsymbol{y}(\cdot)$ and q lags plus the present value of $\boldsymbol{u}(\cdot)$, and where $\boxed{\boldsymbol{\varepsilon}_*(\cdot) \in \mathcal{C}^k\ ???}$ is the realization of a zero-mean, second moment ergodic, complex vector-valued random sequence whose components are independent. The matrix $\boldsymbol{\Theta}_* \in \mathcal{C}^{m\times k}$ parameterizes the model. At time n we wish to use the observed data on $t \in [1, n]$ to deduce an estimated model of the same form. The parameter estimate is denoted by $\boldsymbol{\Theta}(n)$ and the residual process by $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}(n))$. The dependence of the residual upon the parameter estimates is highly significant, so it is shown explicitly.

Deller *et al.* [3] have recently shown that all reported OBE algorithms, including F-H OBE and D-H OBE, can be unified into a general framework which they call the *Unified OBE (UOBE)* algorithm. We initially present the UOBE framework: UOBE algorithms arise from a bounded error constraint:

$$\| \boldsymbol{\varepsilon}_*(n) \|^2 < \gamma(n), \tag{2}$$

where $\gamma(\cdot)$ is a known positive sequence. At time n, a *set* of parameters can be found which are consistent with the observations and this sequence of bounds. The exact set is difficult to describe and track, but, in conjunction with *weighted recursive least square (WRLS)* processing (e.g. [6, 7]), $\Omega(n)$ can be shown to be contained in a superset of the form (e.g. [3],[4],[8])

$$\bar{\Omega}(n) = \left\{ \boldsymbol{\Theta} \mid \operatorname{tr}\{[\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]^H \frac{\boldsymbol{C}(n)}{\kappa(n)} [\boldsymbol{\Theta} - \boldsymbol{\Theta}(n)]\} < 1 \right\} \tag{3}$$

where $\operatorname{tr}\{\cdot\}$ denotes the trace of a matrix, $\boldsymbol{\Theta}(n)$ is the WRLS parameter estimate at time n using weights $\lambda_n(1), \ldots, \lambda_n(n)$. $\boldsymbol{C}(n)$ is the weighted covariance matrix, and $\kappa(n)$ is the scalar quantity

$$\kappa(n) \stackrel{\text{def}}{=} \operatorname{tr}\{\boldsymbol{\Theta}^H(n)\boldsymbol{C}(n)\boldsymbol{\Theta}(n)\} + \sum_{t=1}^{n} \lambda_n(t) \left[\gamma(t) - \| \boldsymbol{y}(t) \|^2\right] . \tag{4}$$

$\Omega(n)$ is a hyperellipsoid in $\boldsymbol{C}^{m\times k}$, with its center at $\boldsymbol{\Theta}(n)$. By examining a single output say $y_i(\cdot)$, the i^{th} component of $\boldsymbol{y}(\cdot)$ we see that a common "ellipsoid matrix" $\boldsymbol{C}(n)/\kappa(n)$ is shared by each of the individual outputs, but that each is centered on a different parameter estimate represented by column i of $\boldsymbol{\Theta}(\cdot)$. We conclude therefore that under bounded error constraints, a hyperellipsoid can be associated with a WRLS recursion and conversely.

The subscript "n" on the weights $\lambda_n(\cdot)$ is used to indicate that the weights may be dependent upon the time of estimation. In general, time dependent weights are not easily integrated into WRLS algorithms except in simple cases. One such case occurs in the UOBE algorithm in which the weights are time varying by virtue of a *scaling* procedure. The weights used at time n are given by

$$\lambda_n(t) = \frac{\lambda_{n-1}(t)}{\zeta(n-1)} \text{ for } t \leq n-1 , \tag{5}$$

and $\lambda_n(n)$, where $\zeta(\cdot)$ is a positive scaling sequence. We make the reasonable assumption that the sequence $\zeta(\cdot)$ is "causal" in the sense that $\zeta(n)$ does not depend upon any quantities not available at time n. The method for integrating scaled weights into WRLS is given inherently in [4] and [8], and explicitly in [3] and [9]. While the weights are directly related to the size, orientation, and location of the ellipsoid in the parameter space, *this scaling procedure effectively restricts to* one *(viz. $\lambda_n(n)$) the number of free parameters available to control the bounding ellipsoid at time n.* The central objective of the UOBE algorithm is to employ the weights in the context of WRLS estimation to sequentially minimize the ellipsoid size in some sense. A significant benefit is that often no weight exists which can minimize the ellipsoid, indicating that the incoming data set is uninformative in the SM sense.

All UOBE algorithms adhere to the following steps: At time n,

1. In conjunction with the incoming data set $(\boldsymbol{y}(n), \boldsymbol{x}(n))$, find the weight, say $\lambda_n^*(n)$, which is optimal in some sense (see below);

2. Discard the data set if $\lambda_n^*(n) \leq 0$.

3. Update $\boldsymbol{C}(n)$ and $\boldsymbol{\Theta}(n)$ using some version of WRLS (e.g. see [8]).

4. Update $\kappa(n)$ using (4) or one of the recursions in [3].

Three fundamental variations on the UOBE method have been reported in the literature. The most recent, the D-H OBE algorithm [5], is unlike the others in one important aspect. This difference lies in the criterion used for determining optimal weights. This difference, on one hand, allows for a proof of convergence of the ellipsoid in a certain sense. On the other hand, the optimization criterion used is controversial and somewhat difficult to interpret. Further, the usual optimization criterion so profoundly changes the development of the algorithm, that its identity as a member of the UOBE class of algorithms has not been appreciated.

The other two reported OBE methods are the F-H OBE algorithm [4] ($\zeta(n) = \kappa(n)$) and the *SM-WRLS* algorithm of Deller *et al.* [8],[10] ($\zeta(n) = 1$). Variations on, and enhancements to, each of these algorithms, as well as D-H OBE, are found in the literature (e.g. [11]—[14]). The stated purpose of this paper is to make connections between D-H OBE and F-H OBE. However, the important contrast exists between D-H OBE and *any* UOBE algorithm with "conventional optimization" based on a meaningful set measure as described below. Let us refer to the latter class of algorithms as UOBE-μ, and generalize the discussion.

UOBE-μ algorithms operate on the optimization principle of (prospectively) minimizing some set measure of $\Omega(n)$, say $\mu\{\bar{\Omega}(n)\}$. Fogel and Huang [4] suggest two set measures. The first is the determinant of the inverse ellipsoid matrix

$$\mu_v \{\bar{\Omega}(n)\} \overset{\text{def}}{=} \det \{\kappa(n)\boldsymbol{C}^{-1}(n)\} \tag{6}$$

and the second is the trace,

$$\mu_t\{\Omega(n)\} \overset{\text{def}}{=} \text{tr}\ \{\kappa(n)\boldsymbol{C}^{-1}(n)\}. \tag{7}$$

(We shall henceforth write $\mu_v(n)$ and $\mu_t(n)$ for simplicity.) In the single output case in which $\Omega(n)$ is clearly intepretable as an ellipsoid, $\mu_v(n)$ is proportional to the square of the volume of the ellipsoid, while $\mu_t(n)$ is proportional to the sum of squares of its semi-axes. The same two measures are meaningful in the multiple output case, since they result in the minimization of the volume or trace of the *common* ellipsoid shared by all the outputs (see discussion below (4)).

The general method for finding the UOBE-μ optimal weight for minimizing the either set measure is given in [3]. These methods include results for F-H OBE and SM-WRLS as special cases, but optimization strategies are also given of course in the original papers. It is found that $\lambda_n^*(n)$ is the *unique* positive root of the polynomials $F_v(\lambda)$ and $F_t(\lambda)$ for the volume and trace measures respectively, where F_v is a quadratic,

$$F_v(\lambda) = a_2\lambda^2 + a_1\lambda + a_0 \ , \tag{8}$$

and F_t is a cubic polynomial

$$F_t(\lambda) = b_3\lambda^3 + b_2\lambda^2 + b_1\lambda + b_0 \ . \tag{9}$$

The coefficients a_i and b_i are given in terms of quantities which are known prior to time n, and which, in turn, are dependent upon the scaling sequence $\zeta(\cdot)$. The interesting feature of the UOBE-μ algorithms is the infrequent existence of the optimal weight leading to infrequent updating of the parameter estimates. This reduction in the need for updating, in turn, results in computational efficiencies and interesting performance properties.

In contrast, the D-H OBE algorithm uses scale factors $\zeta(n) = (1 - \lambda_n^*(n))^{-1}$, and an optimization procedure which does *not* seek to directly minimize a set measure on $\Omega(n)$ such as (6) or (7). Rather, the weight is chosen to minimize $\kappa(n)$, subject to the constraint that it be in the allowable range $[0, \alpha]$ with $0 < \alpha < 1$ (see below). The choice of scaling sequence results in the covariance matrix at time n,

$$\boldsymbol{C}(n) = (1 - \lambda_n^*(n))\boldsymbol{C}(n-1) + \lambda_n^*\boldsymbol{x}(n)\boldsymbol{x}^H(n) \tag{10}$$

which is seen to be a convex combination of $\boldsymbol{C}(n-1)$ and the new data outer product. Here we see the reason for the constraint on the range of optimal weights. This construction provides the means with which to prove asymptotic and exponential convergence of the ellipsoid, and cessation of updating, using Lyaponov theory. Upon convergence, the residuals, $\boldsymbol{\varepsilon}(\cdot, \boldsymbol{\Theta}(\cdot))$ are guaranteed to remain in the "dead zone" indicated by the error bounds, i.e., $\lim_{\substack{t\to\infty \\ n\to\infty}} \|\, \boldsymbol{\varepsilon}(t, \boldsymbol{\Theta}(n))\, \|^2 > \gamma(t)$.

Dasgupta and Huang [5] show that such an weight optimal weight in the sense of minimizing $\kappa(n)$ exists iff

$$\varepsilon^2(n, \boldsymbol{\theta}(n-1)) > \gamma(n) - \kappa_s(n-1) \ , \tag{11}$$

where $\kappa_s(n-1) \overset{\text{def}}{=} \kappa(n-1)/\zeta(n-1)$, the "scaled" value of the κ parameter. Accordingly this simple and computationally inexpensive test may be employed to determine whether the the current data set $(\boldsymbol{y}(n), \boldsymbol{x}(n))$ is useful in the sense of the optimization criterion. However, whether this goal of minimizing $\kappa(n)$ is meaningful remains an issue of controversy. From an analytical point of view, the reason for this choice is that $\kappa(n)$ is a bound on the Lyapunov function used in the minimization, and the convergence of the Lyapunov function is used to prove convergence of the algorithm. From an interpretive point of view, however, diminishing $\kappa(n)$ is not helpful because its magnitude is not clearly related to the "size" of the set $\Omega(n)$. Dasgupta and Huang [5] argue simply that $\kappa(n)$ is "a bound on the estimation error," and should be minimized. Norton and Mo [15] dispute this claim with the observation that "[minimizing $\kappa(n)$] is claimed in [5] to minimize a bound on the estimation error, but the quantity minimized is not a bound on the parameter error, nor does it bear a simple relation to it."

Hence, we have arrived at the apparent philosophical and practical dilemma which initiated this discussion. When faced with the choice of OBE algorithms, does one opt for the D-H OBE method with its proven convergence properties, or the UOBE-μ (including F-H OBE) algorithms with their clear interpretation? In the following section, we demonstrate some heretofore unrecognized connections between the methods which provide a better basis for making this choice.

2 Connections Between the D-H OBE and UOBE-μ (F-H OBE) Algorithms

In spite of some statements to the contrary in the literature (based on apparent misunderstandings of the original F-H paper [4]), there is no known proof of convergence of the F-H OBE algorithm according to any reasonable criteria. In particular, F-H OBE is not known to converge in the sense described for D-H OBE. These statements apply to UOBE-μ algorithms in general. We shall not pursue such a convergence result. Rather, we shall show that some of the "interpretability" of UOBE-μ algorithms may be "transferred" to D-H OBE.

It is somewhat curious that we have called the D-H method an "OBE" algorithm. The bounding ellipsoid clearly underlies the process, but its use in the optimization procedure is obscure. Herein lies the crux of the problem with interpretation. While it is not exploited in the D-H OBE algorithm, the D-H hyperellipsoid nevertheless *does* have volume and trace set measures at each n. In fact, because D-H OBE is fundamentally a UOBE algorithm, the unique positive root of (8), if it exists, will minimize μ_v. A similar statement applies to (9) and μ_t. The utility of this volume or trace result remains an open question at this point in our discussion, because to use weights which are optimal in these "conventional" senses does not necessarily admit the *convergence* results obtained by the D-H analysis.

The connection between the two methods rests fundamentally in the zero order coefficients a_0 and b_0 of the optimization polynomials (8) and (9). The following result has been shown in [3]:

Theorem 1 *The quadratic $F_v(\lambda)$ associated with the UOBE-μ algorithms has a* <u>unique</u> *positive root (the optimal weight, $\lambda_v^*(n)$) iff $a_0 < 0$. Similarly, cubic equation $F_t(\lambda)$ has a* <u>unique</u> *positive root iff $b_0 < 0$.*

Let us henceforth restrict our attention to the volume minimization case with the understanding that a parallel discussion applies to the trace measure. The coefficient a_0 is given by [3]

$$a_0 = mk\left[\gamma(n) - \|\varepsilon(n,\boldsymbol{\Theta}(n-1))\|^2\right] - \kappa_v(n-1)G_v(n) \tag{12}$$

where $G_v(n) \stackrel{\text{def}}{=} \boldsymbol{x}^H(n)\boldsymbol{C}_v^{-1}(n-1)\boldsymbol{x}(n)$, $\boldsymbol{C}_v(n) \stackrel{\text{def}}{=} \boldsymbol{C}(n)/\zeta(n)$, and all other quantities have been defined above. Noting that the computation of this quantity is nearly as computationally expensive as simply including the new data in the estimate, Deller and Odeh [12],[13] suggest the suboptimal testing procedure in which the new data are used iff

$$\|\boldsymbol{\varepsilon}(n,\boldsymbol{\Theta}(n-1))\|^2 > \gamma(n). \tag{13}$$

While originally developed using a different argument, it is seen that the test (13) has a very useful interpretation in terms of the "proper" test $a_0 < 0$. In fact, (13) is equivalent to testing whether

$$a_0 + K_1 < 0 \tag{14}$$

where K_1 is the last term in (12). Since $K_1 > 0$ [3], *an optimal weight in the sense of diminishing $\mu_v(n)$ will always exist if the suboptimal test (13) is satisfied.* Experimental studies have shown that the UOBE-μ algorithm with suboptimal testing performs as well as the optimal algorithm in terms of tracking and (empirical) convergence, while frequently using significantly fewer data (e.g. see [12],[13]).

Let us now examine the test employed by Dasgupta and Huang, given in (11). Recall that this test is designed to determine whether an optimal weight exists in the sense of minimizing $\kappa(n)$, say $\lambda^*_{n,\kappa}(n)$. However, in light of the developments above, the D-H test may also be seen to be a *suboptimal test for the existence of an optimal weight in the sense of diminishing μ_v, say $\lambda^*_{n,v}(n)$.* In fact, the D-H test is equivalent to testing whether

$$a_0 + K_2 < 0 \tag{15}$$

where $K_2 = \kappa_s(n-1)\left[(G_s(n)/mk) - 1\right]$. Unfortunately, the truth of (15) is not sufficient to assure that $a_0 < 0$ because K_2 is not necessarily positive. If it were additionally known that $G_s(n) > mk$, then (15) would be a sufficient test. However, because of the weighting strategy used in D-H OBE, there is no reason to believe that this latter condition holds in general. So the D-H test comes intriguingly close to being a check for the existence of $\lambda^*_{n,v}(n)$, but falls somewhat short. Nevertheless, the D-H test *can* be interpreted as a suboptimal test for the existence of an optimal "volume weight."

To summarize the result above, at time n, there is an optimal weight in the sense of minimizing $\mu_v(n)$, $\lambda^*_{n,v}(n) \geq 0$, associated with the D-H ellipsoid. The test (11) is a suboptimal check for the existence of a positive value of $\lambda^*_{n,v}(n)$. The relevance of this result for D-H OBE in which κ, not μ_v, is minimized, is as follows. While generally $\lambda^*_{n,\kappa}(n) \neq \lambda^*_{n,v}(n)$, roughly speaking, $\lambda^*_{n,\kappa}(n)$ will be positive, and the data used, only when the volume *can* be minimized. While $\lambda^*_{n,\kappa}(n)$ is not designed to minimize μ_v, it will still *diminish* the volume, though not optimally. In turn, this fact is a consequence of the following, which can be inferred from the work in [3]:

Theorem 2 *For any UOBE algorithm, if $\lambda^*_{n,v}(n) > 0$, then if $\underline{\text{any}}$ positive weight is used, $\mu_v(n) < \mu_v(n-1)$.*

Consequently, to the extent that (11) is a useful test for a positive $\lambda^*_{n,v}(n)$, it can be stated that the D-H OBE algorithm diminishes the volume at each step in the process of minimizing κ.

The arguments above are not rigorous because (11) is not an exact check for a positive $\lambda^*_{n,v}(n)$. Further, algorithms which use different weights can only be compared "locally," that is, at a given n. However, it seems intuitive that if an algorithm which suboptimally diminishes volume at each n can be shown to converge, then a covergent algorithm which optimally diminishes μ_v could be demonstrated. By showing that D-H OBE and UOBE-μ (F-H OBE) are more similar than had previously been understood, this discussion suggests that such an algorithm may be forthcoming

References

[1] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: A survey," *Mathematics and Computers in Simulation*, vol. 32, pp. 449-468, 1990.

[2] M. Milanese and A. Vicino, "Estimation theory for dynamic systems with unknown but bounded uncertainty: An overview," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 859-867, Budapest, July 1991.

[3] J.R. Deller, Jr., M. Nayeri, and S.F. Odeh, "System identification using set-membership based signal processing," *Proceedings of the IEEE* (in review).

[4] E. Fogel and Y.F. Huang, "On the value of information in system identification - Bounded noise case," *Automatica*, vol. 18, pp 229-238, 1982.

[5] 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 Trans. Information Theory*, vol. IT-33, pp. 383-392, 1987.

[6] C.R. Johnson, *Lectures on Adaptive Parameter Estimation*, Englewood Cliffs, New Jersey: Prentice-Hall, 1988.

[7] G.C. Goodwin and K.S. Sin, *Adaptive Filtering, Prediction, and Control*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[8] J.R. Deller, Jr. and T.C. Luk, "Linear prediction analysis of speech based on set-membership theory," *Computer Speech and Language*, vol. 3, pp. 301-327, Oct. 1989.

[9] J.P. Norton, *An Introduction to Identification*, London and Orlando, Florida: Academic Press, 1986.

[10] J.R. Deller, Jr., "Set-membership identification in digital signal processing," *IEEE ASSP Magazine*, vol. 6, pp. 4-22, Oct. 1989.

[11] G. Belforte, B. Bona, and V. Cerone, "Identification, structure selection and validation of uncertain models with set-membership error description," *Mathematics and Computers in Simulation*, vol. 32, pp. 561-569, 1990.

[12] J.R. Deller, Jr. and S.F. Odeh, "SM-WRLS algorithms with an efficient test for innovation," *Proceedings of the 9th IFAC / IFORS Symp. on Identification and System Parameter Estimation*, vol. 2, pp. 1044-1049, Budapest, July 1991.

[13] S.F. Odeh and J.R. Deller, Jr., "An SM-WRLS algorithm with an efficient test for innovation: Simulation studies and complexity issues," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '91*, Toronto, vol. 5, pp. 3413-3416, May 1991.

[14] A.K. Rao, Y.-F. Huang, and S. Dasgupta, "ARMA parameter estimation using a novel recursive estimation algorithm with selective updating," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 38, pp. 447-457, March 1990.

[15] J.P. Norton and S.H. Mo, "Parameter bounding for time-varying systems," *Mathematics and Computers in Simulation*, vol. 32, pp. 527-534, 1990.

A CONVERGING OPTIMAL ELLIPSOID ALGORITHM WITH VOLUME MINIMIZATION; A SUMMARY[1]

M. Nayeri, J.R. Deller, and M.S. Liu

Michigan State University
Department of Electrical Engineering - 260 EB
CSSP Group: Speech Processing & Adaptive Signal Processing Laboratories
East Lansing, MI 48824-1226

ABSTRACT

A newly modified *set-membership* algorithm is introduced. It is shown that the *forgetting* covariance updating in conjunction with *minimum volume* data selecting strategy result in a landmark performance level in system identification. A suboptimal test for data selection is introduced which is computationally efficient.

[1] This work was supported by the National Science Foundation under Grant No. MIP-9016734 and by the Office of Naval Research under Contract No. N00014-91-J-1329.

1. INTRODUCTION

The behavior of optimal bounding ellipsoid algorithms (OBE) such as Fogel-Huang and SM-WRLS has attracted the attention of prominent researchers. While the strong performance of these algorithms can not be disputed, there are no supporting theoretical proofs for their converging behavior. Dasgupta and Huang [4] introduced an OBE algorithm whose convergence was shown theoretically by employing a Lyapunov technique. However, the data-selection strategy implemented in this algorithm is the center of some debate. This originates from a controversial criterion used to measure the performance of the algorithm if new data are to be selected.

In this paper, we introduce a modified SM-WRLS algorithm whose convergence will be proven in the most general system identification setting. This algorithm employs the covariance updating used by Dasgupta and Huang, and selects the incoming data according to SM-WRLS volume minimization strategy.

2. MODIFIED SM-WRLS ALGORITHM

Since the Modified SM-WRLS algorithm is also an OBE algorithm, it adheres to the general steps outlined in [8]. To address the update recursions and data-selection strategy, let us adopt similar notation used in [8].

Update Recursions: The modification of the SM-WRLS Covariance updating to

$$C(n) = (1-\lambda_n)\, C(n-1) + \lambda_n\, x_n x_n^T \tag{1}$$

with $0 \le \lambda_n \le 1$, has a profound effect on other measures employed in SM-WRLS as indicated below, where $P(n) = C^{-1}(n)$:

$$\begin{aligned}
P(n) &= \frac{1}{1-\lambda_n}\left[P(n-1) - \frac{\lambda_n\, P(n-1)x_n x_n^T P(n-1)}{1-\lambda_n+\lambda_n G_n}\right]\\
\theta_n &= \theta_{n-1} + \lambda_n\, P(n)x_n\,\varepsilon_n\\
\varepsilon_n &= y_n - \theta^T{}_{n-1}x_n\\
\kappa_n &= (1-\lambda_n)\,\kappa_{n-1} + \lambda_n\gamma_n - \frac{\lambda_n(1-\lambda_n)\,\varepsilon^2{}_n}{1-\lambda_n+\lambda_n G_n}\\
G_n &= x_n^T P(n-1)x_n
\end{aligned} \tag{2}$$

Data Selection: Selection of the incoming data involves the minimization of the bounding ellipsoid's volume, with respect to the weights λ_n. The optimal weight, denoted by λ_n^*, is given by

$$\lambda_n^* = \max(0, \lambda_{\max}) \tag{3}$$

where $\lambda_{\max}$ is the larger root of the following quadratic equation:

$$\alpha_2\lambda^2 + \alpha_1\lambda + \alpha_0 = 0 \tag{4}$$

where

$$\begin{aligned}
\alpha_2 &= m\gamma_n - m\varepsilon_n^2 + mG_n^2\gamma_n - 2mG_n\gamma_n - \kappa_{n-1}G_n + \kappa_{n-1}G_n^2 + G_n\gamma_n - G_n^2\gamma_n - \varepsilon_n^2 G_n\\
\alpha_1 &= 2m\varepsilon_n^2 - 2m\gamma_n - 2mG_n\gamma_n + 2\kappa_{n-1}G_n - \kappa_{n-1}G_n^2 - G_n\gamma_n + \varepsilon_n^2 G_n\\
\alpha_0 &= m\gamma_n - m\varepsilon_n^2 - \kappa_{n-1}G_n
\end{aligned}$$

Theorem 1: Equation (4) has *at most* one positive root in which case the selection of the data point guarantees the shrinkage of the bounding ellipsoid in volume. Moreover, this positive root lies in the interval (0,1).

This result implies that the data must be discarded if

$$\frac{\alpha_0}{\alpha_2} > 0$$

Also, noting that (1) represents a convex combination of $C(n)$ and $x_n x_n^T$ matrices, Theorem 1 suggests that as long as a positive root to (4) is found, no extra monitoring is needed to satisfy the $0 \leq \lambda_n^* \leq 1$ condition. This is a very convenient data selection strategy.

3. CONVERGENCE OF THE ALGORITHM

This modified SM-WRLS algorithm shows an attracting convergence behavior as given by the following Theorem.

Theorem 2: Let us assume that the noise process $v(n)$ is persistently exciting with pointwise energy bound

$$v^2(n) < \gamma_n \qquad (5)$$

Then, the modified SM-WRLS shows convergence in the following sense:

$$(1) \quad \lim_{n \to \infty} || \theta_n - \theta^* || = 0$$

$$(2) \quad \lim_{n \to \infty} \lambda_n^* = 0$$

$$(3) \quad \lim_{n \to \infty} \varepsilon_n^2 \in [0, \gamma_n]$$

Example:

Let us consider a simple AR(2) model given by

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + v(n)$$

where $a_1 = 1.6$ and $a_2 = -0.68$ and $v(n)$ is a white noise sequence where $\gamma_n = 0.5$ in (5). Figure 1 clearly indicates the asymptotic convergence of the parameters to the desired values. Also shown in Fig. 2 is the volume of the bounding ellipsoid as more data is selected. The asymptotic convergence of the optimal weight λ_n^* to zero is shown in fig. 3. Figure 4 indicates that in the limit ε_n^2 will be equivalent to the input noise $v(n)$, as expected.

4. SUBOPTIMAL TEST

There are some common features of SM-WRLS that are inherited by the modified version one of which is the feasibility of the same suboptimal test shown in [3] and [8]. This is so because of the resemblance of the α_1 coefficient in (4) to its counterpart in conventional SM-WRLS. Figure 5 shows the relative computational requirements for the optimal and suboptimal tests associated with this algorithm.

REFERENCES

[1] F.C. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Trans. Automatic Control*, vol. AC-13, pp. 22-28, 1968.

[2] D.P. Bertsekas and I.B. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Trans. Automatic Control*, vol. AC-16, pp. 117-128, 1971.

[3] J.R. Deller, Jr., "Set membership identification in digital signal processing, *IEEE ASSP Magazine*, vol. 6, no. 4, pp. 4-20, Oct. 1989.

[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 Trans. Information Theory*, vol. IT-33, pp. 383-392, 1987.

[5] Y. F. Huang, "A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-34, pp. 1331-1334, 1986.

[6] Y. F. Huang and A.K. Rao, "Application of a recursive estimation algorithm with information-dependent updating to ARMAX models and ARMA models with unknown inputs," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '87*, Dallas, vol. 2, pp. 1007-1010, 1987.

[7] S.F. Odeh and J.R. Deller, Jr., "An SM-WRLS algorithm with an efficient test for innovation: Simulation studies and complexity issues," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '91*, Toronto, vol. 5, pp. 3413-3416, May 1991.

[8] J. R. Deller, M. Nayeri, and S. F. Odeh, "System identification using set-membership-based signal processing," *Proceeding of the IEEE*, submitted 12/91.

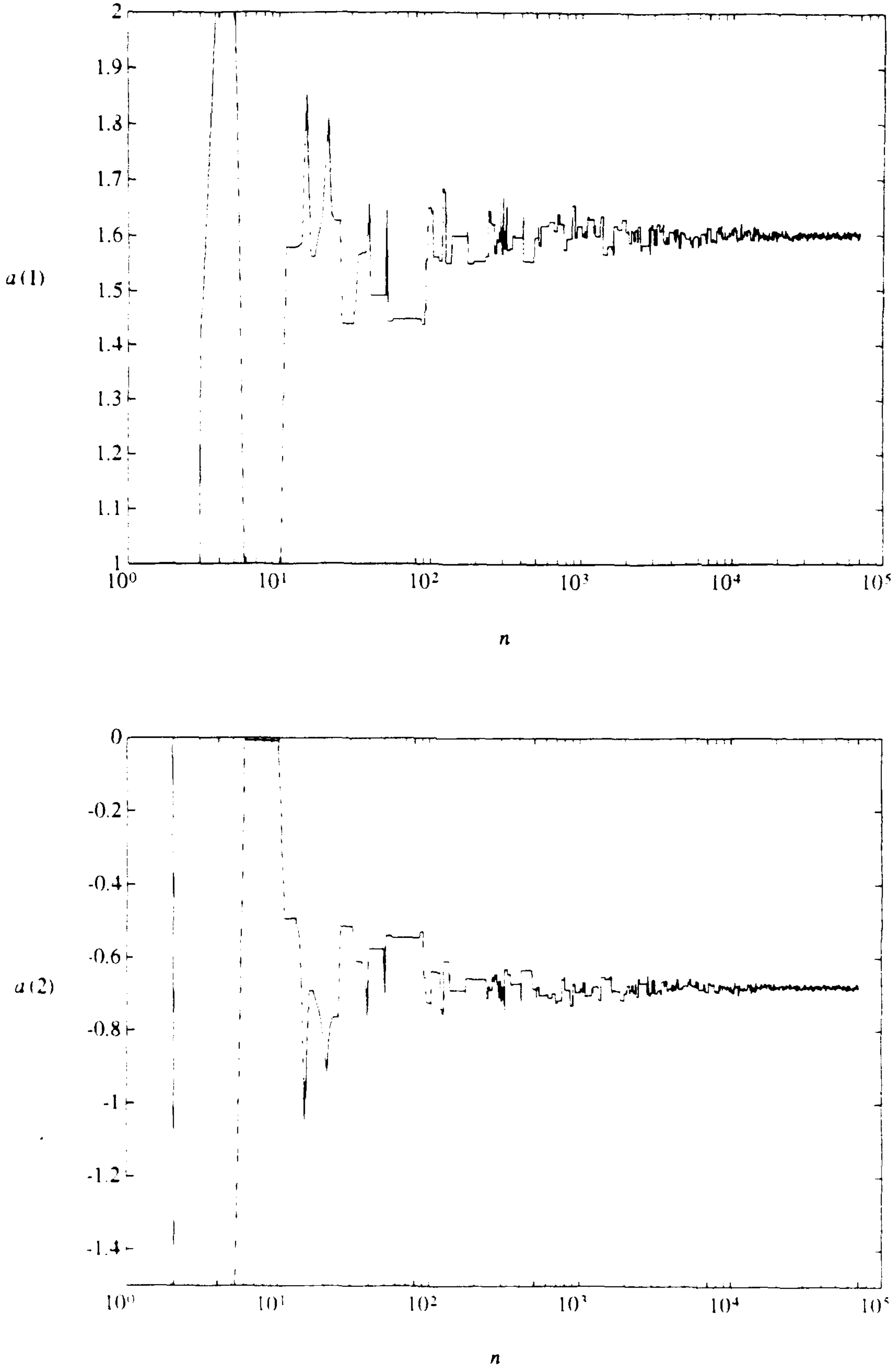


Fig. 1. Asymptotic convergence of $a(1)$ and $a(2)$ to a_1 and a_2 in the example.

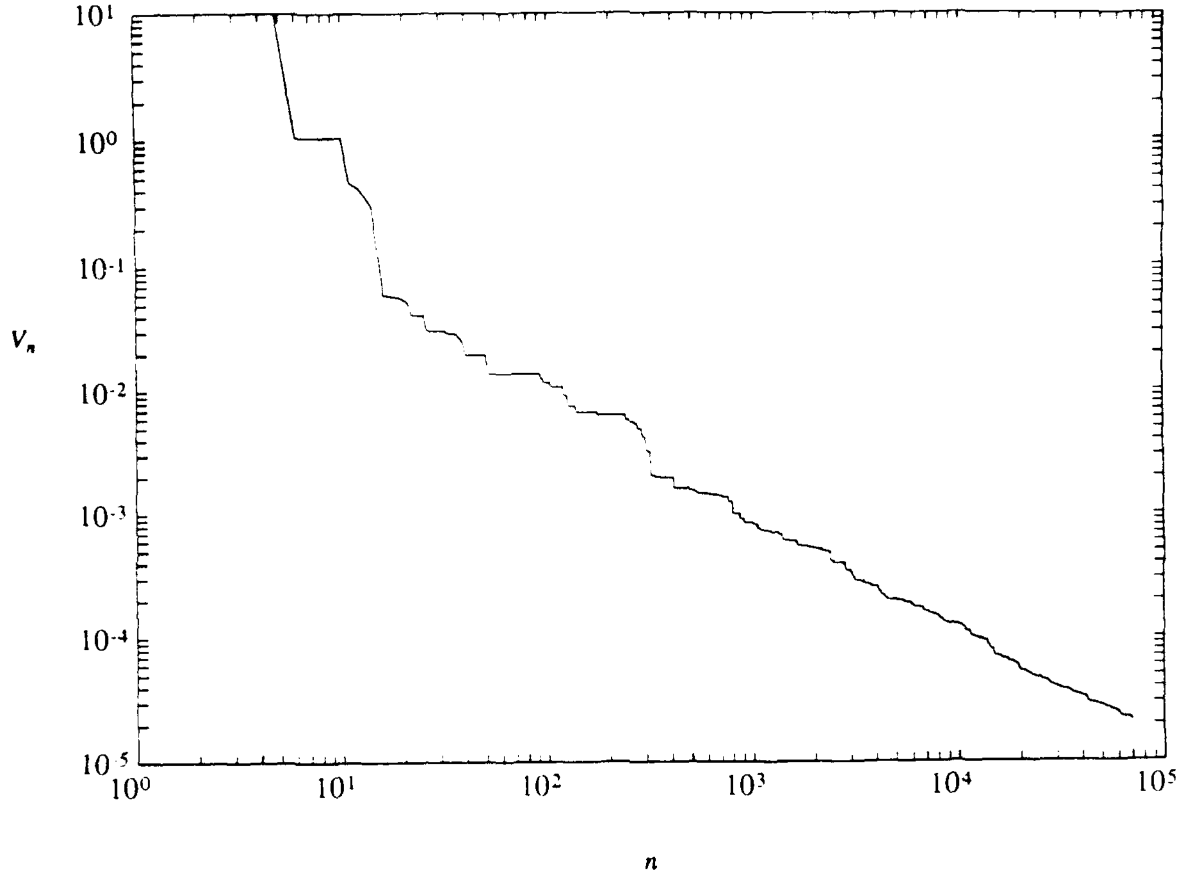


Fig. 2. *Almost* monotonically decreasing characteristic of the volume measure.

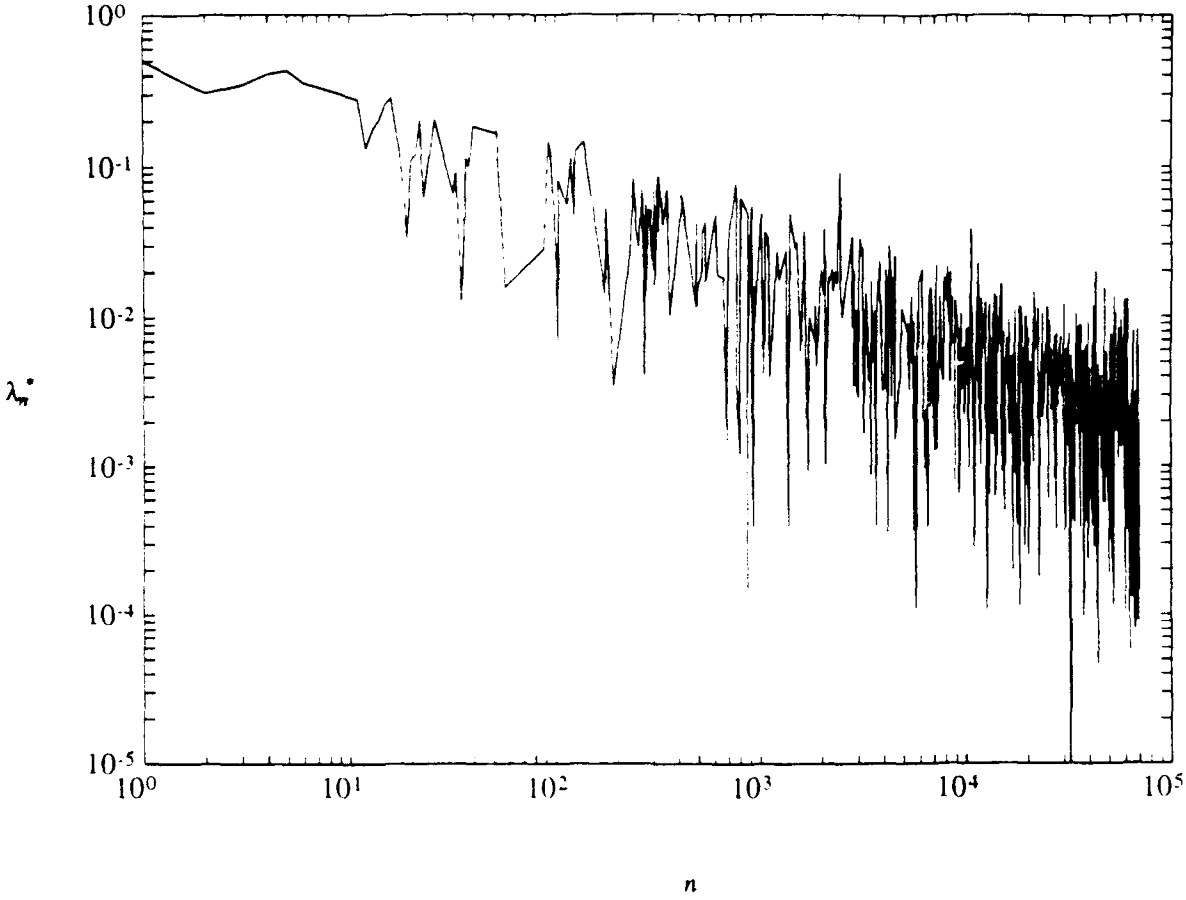


Fig. 3. Asymptotic convergence of λ_n^* to zero.

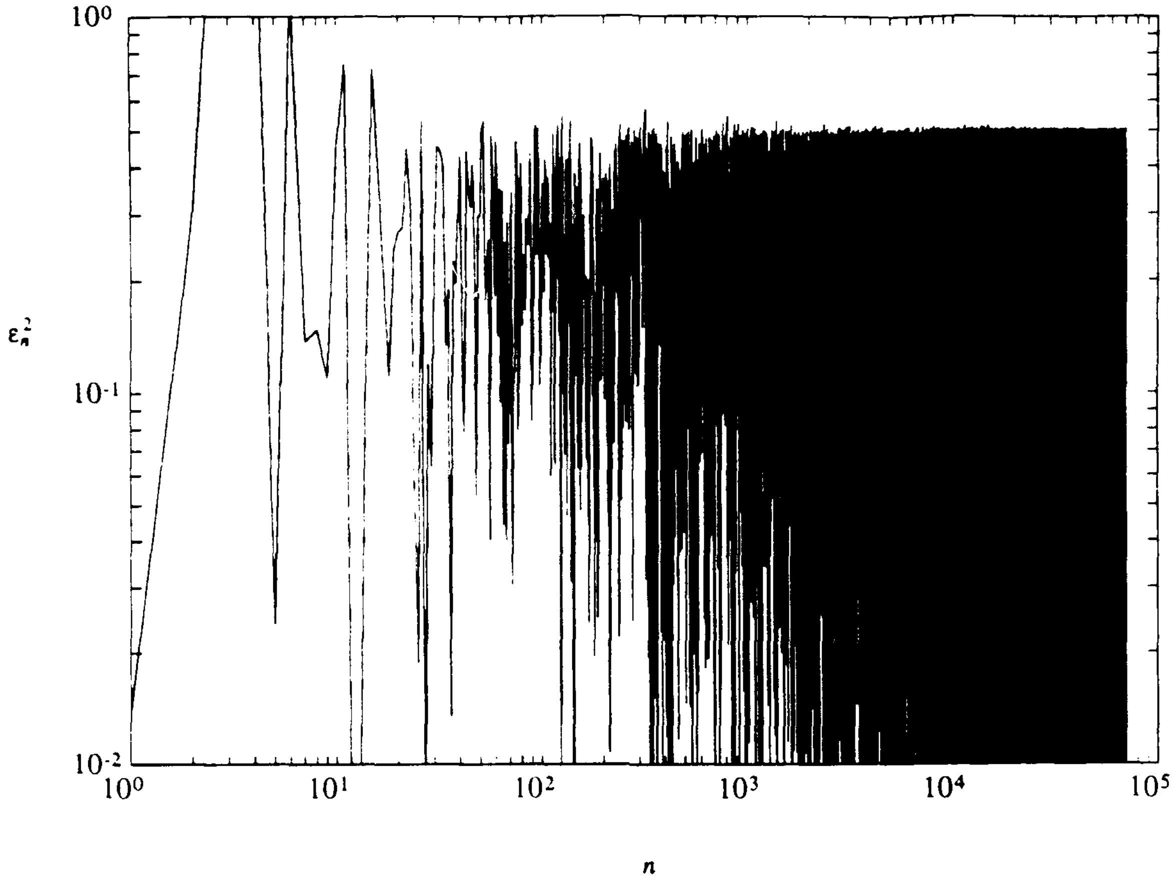


Fig. 4. ε_n^2 vs. n for the identification of AR(2) model in the example.

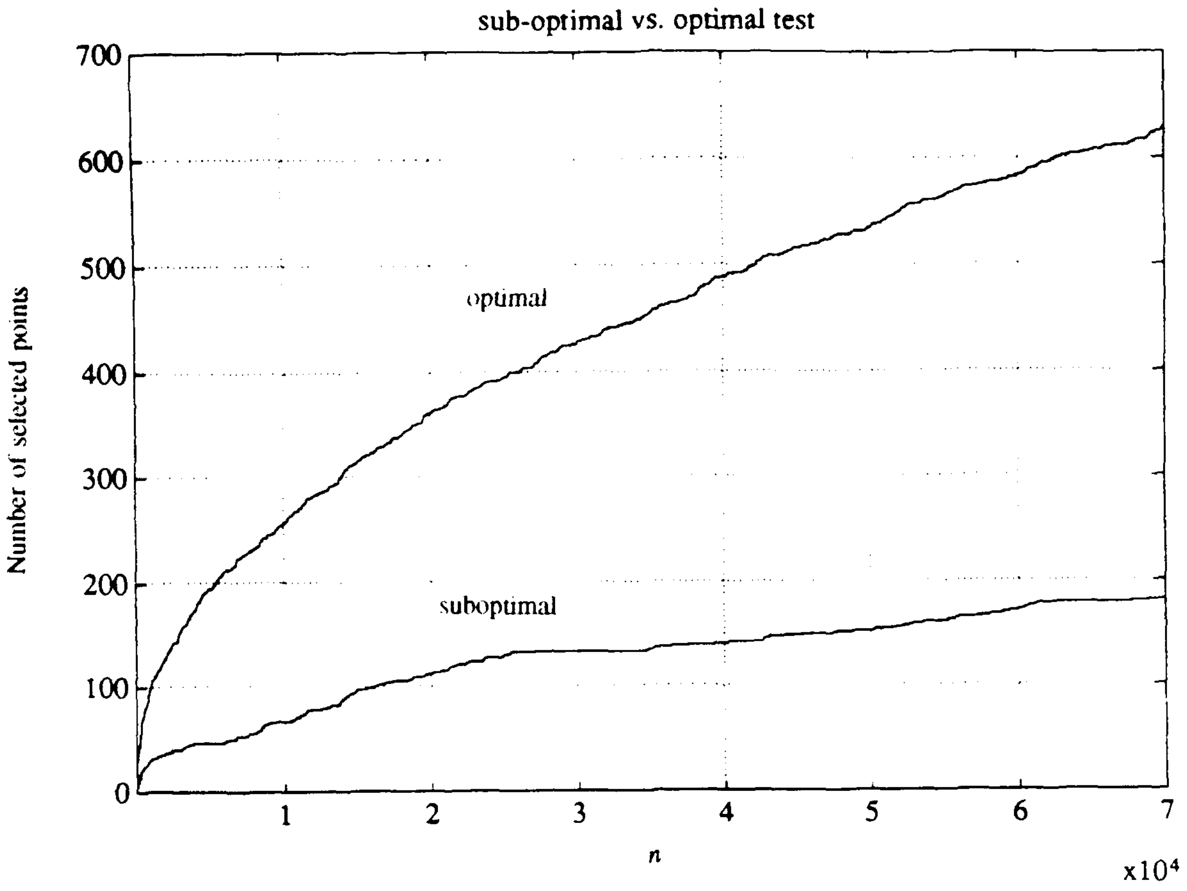


Fig. 5. The number of selected points vs. n, for the optimal and suboptimal tests.

THE UNIVERSITY
OF BIRMINGHAM

Faculty of Engineering
School of Electronic and
Electrical Engineering

Edgbaston
Birmingham B15 2TT
United Kingdom
Telephone 021 414 4285
Fax 021 414 4291
Telex 333762 UOBHAM G

Head of School
Professor D. J. Creasey
Head of Postgraduate Division
Research
Professor J. P. Norton

Direct Line 021 414 4300 4301

Prof. J.R. Deller
Dept. of Electrical Engineering
Michigan State University
260 Engineering Building
East Lansing
Michigan 48824
UNITED STATES

8th January 1992

Dear John,

I am writing to invite you to contribute a paper to a special issue of the International Journal of Adaptive Control and Signal Processing devoted to Bounded-Error Estimation. As you know, this topic is gaining international interest very rapidly and has been the subject of a great deal of discussion at recent meetings such as the IFAC identification symposium in Budapest and the CDC in Brighton. IJACSP is a particularly suitable journal because of the strong recent interest shown in the topic by the signal-processing community. I am hoping, as guest editor, that we can use this special issue to spread awareness of bounding techniques and their potential; a contribution from you would strengthen the special issue considerably. At present the expectation is that the main focus will be on parameter bounding and state bounding rather than parameter-bound-based control design, although the latter is not positively excluded. Material with a strong tutorial content will be especially welcome, and so would reports of applications. Papers discussing applications of parameter or state bounding which have not yet been well explored (including control design, but emphasising the requirements for bound computation rather than the details of control synthesis) will also be very welcome.

Contributions will be refereed to ensure compatibility with the other contributions and readability will be given a high priority, especially in the more theoretical papers. Provisionally the deadline for submissions is ***31st May 1992***, with the intention that the issue should appear about the end of 1992. We hope to provide rapid and constructive refereeing to allow easy revision where necessary.

You may wish to consider updating and extending a suitable conference paper, with the permission of the copyright holder, if you have such a paper not being rewritten for another journal.

I hope very much that you can respond positively to this invitation, and I look forward to seeing your contribution in due course.

Yours sincerely,

John.

Prof. J.P. Norton

Convergence and Colored Noise Issues in Bounding Ellipsoid Identification

M. Nayeri, J.R. Deller, Jr., and M.M. Krunz

Michigan State University
Department of Electrical Engineering
CSSP Group: Speech Processing & Adaptive Signal Processing Laboratories
East Lansing, MI 48824-1226 USA

email: majid@ee.msu.edu phone: (517) 353-1857 FAX: (517) 353-1980

Abstract[1]

This paper is concerned with the convergence and bias properties of a general class of *optimal bounding ellipsoid* (OBE) algorithms. OBE algorithms are *set-membership* (SM) based identification algorithms which are applied to models which are linear-in-parameters, and are closely related to weighted recursive least square error (WRLS) methods.

1. Introduction and Formalities

This paper will generalize the basic OBE identification problem with respect to existing publications in several ways. First, a complex signal, MIMO system is considered. This treatment subsumes common parametric models such as the SISO ARX model (e.g. [7]) as special cases. Secondly, a *unified OBE* (UOBE) algorithm is developed which contains all reported OBE algorithms, both adaptive and nonadaptive, as special cases.

Assume that we are observing some physical system which is generating sequence $y(\cdot) \in \mathcal{C}^k$ in response to input $u(\cdot) \in \mathcal{C}^l$. $u(\cdot)$ is a realization of an ergodic, wide sense stationary stochastic process. Both input and output sequences are measurable. We assume the existence of a "true" model of form

$$y(n) = \Theta_*^H x(n) + \varepsilon_*(n) \tag{1}$$

in which $x(n)$ is some m-vector of functions of p lags of $y(\cdot)$ and q lags plus the present value of $u(\cdot)$, and where $\varepsilon_*(\cdot) \in \mathcal{C}^k$ is the realization of a zero-mean, second moment ergodic, complex vector-valued random sequence whose components are independent. The matrix $\Theta_* \in \mathcal{C}^{m \times k}$ parameterizes the model. At time n we wish to use the observed data on $t \in [1, n]$ to deduce an estimated model of the same form. The parameter estimate is denoted by $\Theta(n)$ and the residual process by $\varepsilon(n, \Theta(n))$. The dependence of the residual upon the parameter estimates is highly significant, so it is shown explicitly.

In almost all SM-based techniques, a *feasible parameter set* arises from direct or indirect constraints on the additive error sequence. UOBE algorithms arise from a bounded error constraint:

$$\| \varepsilon_*(n) \|^2 < \gamma(n), \tag{2}$$

where $\gamma(\cdot)$ is a known positive sequence. At time n, a set of parameters can be found which are consistent with the observations and this sequence of bounds. The exact set is difficult to describe and track, but, in conjunction with WRLS processing, it can be shown to be contained in a superset of the form (e.g.,[4],[5],[6])

$$\Omega(n) = \left\{ \Theta \mid \operatorname{tr}\{ [\Theta - \Theta(n)]^H \frac{C(n)}{\kappa(n)} [\Theta - \Theta(n)] \} < 1 \right\} \tag{3}$$

where $\operatorname{tr}\{\cdot\}$ denotes the trace of a matrix, $\Theta(n)$ is the WRLS parameter estimate at time n using weights $\lambda_n(1), \ldots, \lambda_n(n)$, $C(n)$ is the weighted covariance matrix, and $\kappa(n)$ is the scalar quantity

$$\kappa(n) \overset{\text{def}}{=} K(n) + \sum_{t=1}^{n} \lambda_n(t) \left[\gamma(t) - \| y(t) \|^2\right], \tag{4}$$

with $K(n) \overset{\text{def}}{=} \operatorname{tr}\{ \Theta^H(n)\, C(n)\, \Theta(n)\}$. We shall refer to $\Omega(n)$ as a "hyperellipsoid" in $\mathcal{C}^{m \times k}$, with its "center" at $\Theta(n)$. Indeed, if all quantities are real, and $m = 2$ and $k = 1$, this set forms an ellipse in $\mathcal{R}^2$. By examining a single output – say $y_i(\cdot)$, the i^{th} component of $y(\cdot)$ – we see that a common "ellipsoid matrix" $C(n)/\kappa(n)$ is shared by each of the individual outputs, but that each is centered on a different parameter estimate represented by column i of $\Theta(\cdot)$. We conclude therefore that under bounded error constraints, a hyperellipsoid can be associated with a WRLS recursion and conversely.

The subscript "n" on the weights $\lambda_n(\cdot)$ is used to indicate that the weights may be dependent upon the time of estimation. In general, time dependent weights are not easily integrated into WRLS algorithms except in simple cases. One such case occurs in the UOBE algorithm in which the weights are time varying by virtue of a simple scaling procedure. The weights used at time n are given by

$$\lambda_n(t) = \frac{\lambda_{n-1}(t)}{\zeta(n-1)} \quad \text{for } t \leq n-1, \tag{5}$$

and $\lambda_n(n)$, where $\zeta(\cdot)$ is a positive scaling sequence. When $\zeta(n)$ is independent of $\lambda_i(j)$ for all $i, j \geq n$, then we shall call these *simply scaled* weights. The method for integrating scaled weights into WRLS is given inherently in [4] and [6], and explicitly in [5] and [9]. While the weights are directly related to the size, orientation, and location of the ellipsoid in the parameter space, *this scaling procedure effectively restricts to one (viz. $\lambda_n(n)$) the number of free parameters available to control the bounding ellipsoid at time n.* The central objective of the UOBE algorithm is to employ the weights in the context of WRLS estimation to sequentially minimize the ellipsoid size in some sense. A significant benefit is that often no weight exists which can minimize the ellipsoid, indicating that the incoming data set is uninformative in the SM sense.

All bounding ellipsoid algorithms, both adaptive and nonadaptive, adhere to the following steps. Consequently, we call this set of operations the *Unified Optimal Bounding Ellipsoid (UOBE)* algorithm: At time n,

1. In conjunction with the incoming data set ($y(n)$, $x(n)$), find the weight, say $\lambda_n^*(n)$, which is optimal in some sense (see below);
2. Discard the data set if $\lambda_n^*(n) \leq 0$.
3. Update $C(n)$ and $\Theta(n)$ using some version of WRLS (e.g. see [4]).
4. Update $\kappa(n)$ using (4) or one of the recursions in [5].

Three fundamental variations on the UOBE method have been reported in the literature. The most recent, due to Dasgupta

[1] This work was supported by the Office of Naval Ressearch under Contract No. N00014-91-J-1329 and by the National Science Foundation under Grant No. MIP-9016734.

Presented at ICASSP-92, March 23-26, San Francisco, California, USA

and Huang ("D-H OBE") [2], is unlike all the others in certain respects. From the present point of view, one of the key differences is that the weight pattern follows (5), but the weights are not *simply* scaled according the definition above. These differences, on one hand, allow for a proof of convergence of the ellipsoid in a certain sense and make the analysis in this paper seemingly unnecessary. On the other hand, the optimization criterion used is controversial and somewhat difficult to interpret. Space does not permit elaboration upon D-H OBE, and no precise connection between this method and more "conventional" OBE algorithms exists in the literature. Hence, the analysis in this paper is not *apparently* related to D-H OBE. However, interesting connections *do* exist and these will be the subject of a forthcoming paper [8].

"Conventional" OBE algorithms operate on the optimization principle of (prospectively) minimizing some set measure of $\Omega(n)$, say $\mu\{\Omega(n)\}$. For the SISO case, Fogel and Huang [6] suggest two set measures. The first is the determinant of the inverse "ellipsoid matrix"

$$\mu_v\{\Omega(n)\} \overset{\text{def}}{=} \det\ \{\kappa(n)\ C^{-1}(n)\} \tag{6}$$

and the second is the trace,

$$\mu_t\{\Omega(n)\} \overset{\text{def}}{=} \operatorname{tr}\ \{\kappa(n)\ C^{-1}(n)\}. \tag{7}$$

(We shall henceforth write $\mu_v(n)$ and $\mu_t(n)$ for simplicity.) In the MISO case in which $\Omega(n)$ is clearly intepretable as an ellipsoid, $\mu_v(n)$ is proportional to the square of the volume of the ellipsoid, while $\mu_t(n)$ is proportional to the sum of squares of its semi-axes. The same two measures are meaningful in the MIMO case, since they result in the minimization of the volume or trace of the *common* ellipsoid shared by all the outputs (see discussion below (4)). The original OBE algorithm of Fogel and Huang ("F-H OBE") [6] follows these UOBE steps with $\zeta(n) = \kappa(n)$ for each n. The *set-membership weighted recursive least squares algorithm (SM-WRLS)* of Deller *et al.* (e.g. see [3],[4]) is a UOBE algorithm with $\zeta(n) = 1$ for all n.

The general method for finding the UOBE optimal weight for minimizing the either set measure is given in [5]. These methods include results for F-H OBE and SM-WRLS as special cases, but optimization strategies are also given of course in the original papers. In the volume case, it is found that the optimal weight is given by the *unique* positive root of a quadratic equation in $\lambda_n(n)$, say $F_v(\lambda_n(n))$, whose coefficients are expressed in terms of known quantities at time $n-1$. The optimal trace weight, if it exists, is the unique positive root of a cubic polynomial, say $F_t(\lambda_n(n))$. The critical feature to keep in mind is the infrequent updating of UOBE which leads to interesting performance properties and computational efficiencies.

2. Convergence Issues

Asymptotic Estimates. One of the interesting and practical benefits of having interpreted UOBE algorithm as a WRLS algorithm with a bounded error "overlay" is the immediate consequence for convergence of the estimator. It is well-known that if the sequence $\varepsilon_*(n)$ is wide-sense stationary, second moment ergodic almost surely (a.s.), white noise, then the WRLS estimator $\Theta(n)$ will converge asymptotically to Θ_* a.s. (e.g. [7]). In the present case, we need only to add the qualifier that the UOBE algorithm not cease to accept data in order to lay claim to this useful result.

Likewise, we may even assert a.s. convergence of the WRLS estimate, albeit to a bias, when $\varepsilon_*(n)$ is colored and persistently exciting[2] (p.e.) [7]. Even in the presence of colored errors, therefore, as long as the acceptance of data does not cease, and the infrequency of updating does not interfere with the persistency of excitation, we may expect the UOBE estimate (ellipsoid center) to converge.

Convergence of the Ellipsoid. It would be interesting to have a precise understanding of the asymptotic behavior of the hyperellipsoidal *feasible set*, especially in the case of colored noise. Knowledge that the ellipsoid is vanishing (white noise), or becoming as small as possible (colored noise), could be very useful information indeed. In the white noise case, a sufficiently small ellipsoid could serve as a reinforcing indicator of convergence, and offer a means of determining error bounds on the estimate in finite time. In the colored noise case, a small feasible set (known to contain the true, unbiased estimate) could be indispensible. Unfortunately, no known proof of this desirable result for any case of UOBE with simply scaled weights exists. The remainder of this paper indicates recent progress made toward the understanding of the convergence properties of the ellipsoid, in the presence of both white and colored noise disturbances.

We first present an important contribution toward the understanding of the asymptotic behavior of the ellipsoid:

Proposition 1 *Consider the UOBE algorithm with simple scale factors as in (5). If an optimal weight exists at time n, then its use will certainly diminish the set measure,*

$$\mu(n) < \mu(n-1) \tag{8}$$

where μ is either μ_v or μ_t. Therefore, in general $\mu(n) \leq \mu_v(n-1)$. Further, for the trace measure, $\mu_t\{\Omega(n)\} = 0$ iff $\Omega(n) = \{\ \Theta(n)\} = \{\ \Theta_*\}$.

Proof: We prove the result for μ_v. The proof for μ_t is similar. The last line of the proposition will be verified in the future discussion.

For simplicity, we write $\lambda_n(n)$ as λ. Also, the functional dependence of $\mu_v(n)$ upon λ for a fixed n is the central issue, so we write $\mu_v(\lambda)$. It has been shown that[3] [5]

$$\mu_v(\lambda) = r^{mk}(n)\left\{1 - \frac{\lambda G_s(n)}{h(n)}\right\} = \frac{r^{mk}(n)}{h(n)} \tag{9}$$

where $h(n) \overset{\text{def}}{=} 1 + \lambda G_s(n)$ and $r(n) \overset{\text{def}}{=} \kappa(n)/\kappa_s(n-1)$, $\kappa_s(n) \overset{\text{def}}{=} \kappa(n)/\zeta(n)$, and $G_s(n) \overset{\text{def}}{=} \zeta(n-1)\ x^H(n)\ C^{-1}(n-1)\ x(n)$. Thus it is found that

$$\frac{\partial\mu_v(\lambda)}{\partial\lambda} = Q(\lambda)R(\lambda) \tag{10}$$

where,

$$Q(\lambda) \overset{\text{def}}{=} \frac{r^{mk-1}(n)}{h^2(n)} \quad \text{and} \quad R(\lambda) \overset{\text{def}}{=} mkh(n)\frac{\partial r(n)}{\partial\lambda} - r(n)G_s(n). \tag{11}$$

For future reference, also note that [5]

$$\kappa_s(n-1)h(n)R(\lambda) = F_v(\lambda) \tag{12}$$

where $F_v(\lambda)$ is the volume quadratic solved to find the optimal weight. Consequently,

$$\frac{\partial^2\mu_v(\lambda)}{\partial\lambda^2} = \frac{\partial R(\lambda)}{\partial\lambda}Q(\lambda) + R(\lambda)\frac{\partial Q(\lambda)}{\partial\lambda} \tag{13}$$

It is easy to demonstrate that $Q(\lambda)$ is positive, and that its derivative is bounded, for $\lambda \in [0, \infty)$. Now it can be shown that [5]

$$r(n) = \frac{\kappa(n)}{\kappa_s(n-1)} = 1 + \frac{\lambda\gamma(n)}{\kappa_s(n-1)} - \frac{\lambda\,\|\ \varepsilon(n,\ \Theta(n-1))\ \|^2}{\kappa_s(n-1)h(n)}. \tag{14}$$

allowing us to write, using (14),

$$\frac{\partial R(\lambda)}{\partial\lambda} = (mk-1)G_s(n)\frac{\partial r(n)}{\partial\lambda} + mk\frac{\|\ \varepsilon(n,\ \Theta(n-1))\ \|^2\ G_s(n)}{\kappa_s(n-1)h^2(n)} \tag{15}$$

[2] Please read the abbreviation "p.e." as "persitently exciting" or "persistency of excitation," as appropriate.

[3] A similar result for a less general case is found in [4].

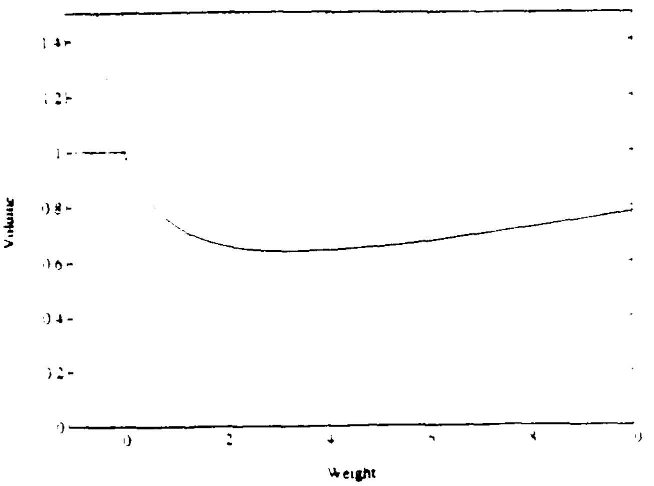


Fig. 1: Typical plot of $\mu_v(\lambda)$ vs. λ. When a positive root of $F_v(\lambda)$ exists, it corresponds to a *minimum* of the volume measure.

Because of (12) it is clear that $R(\lambda^*) = 0$. Reference to the definition of $R(\lambda)$ in (11), therefore immediately shows that $\frac{\partial r(n)}{\partial \lambda}\Big]_{\lambda=\lambda^*} > 0$. Consequently $\frac{\partial R(\lambda)}{\partial \lambda}\Big]_{\lambda=\lambda^*} > 0$. It follows immediately that $\frac{\partial^2 \mu_v(\lambda)}{\partial \lambda^2}\Big]_{\lambda=\lambda^*} > 0$ so that λ^* corresponds to a minimum of $\mu_v(\lambda)$. Further, since $r(n)]_{\lambda=0} = 1$ (see (14)), and $h(n)]_{\lambda=0} = 1$, we have from (9) that $\mu_v(0) = 1$, and also that $Q(0) = 1$. Therefore, from (10) and (12),

$$\frac{\partial \mu_v(\lambda)}{\partial \lambda}\Big]_{\lambda=0} = R(0) = \frac{F_v(0)}{\kappa_s(n-1)} = \frac{a_0}{\kappa_s(n-1)} \qquad (16)$$

where a_0 is the zero order coefficient of the quadratic, $F_v(\lambda) = a_2\lambda^2 + a_1\lambda + a_0$. It has been shown [5] that no positive roots of $F_v(\lambda)$ exist if $a_0 \geq 0$, and exactly one exists if $a_0 < 0$. It follows that the derivative in (16) is negative; hence, $\mu_v(\lambda^*) < 1$ and the proposition is proven (see Fig. 1). □

In spite of this encouraging result, one of the drawbacks of the volume approach is that the set measure μ_v is not a proper "metric" in the parameter space. By this we mean the following: Suppose we propose the distance measure d such that at time n, $d(\Theta(n), \Theta_*) = \mu_v(n)$. We immediately find that d fails to be a proper metric since $d(\Theta(n), \Theta_*) = 0$ does not imply that $\Theta(n) = \Theta_*$. This unfortunate situation arises because the ellipsoid may potentially degenerate and reside in a subspace of $C^{m \times k}$, thereby achieving zero volume without being reduced to a point. This will likely only occur if p.e. is not achieved as we detail below, and is therefore a more important problem with colored disturbances. This potential anomaly provides motivation to consider the use of the trace measure for which a degenerate ellipsoid will not produce a zero set measure.

One of the drawbacks of the UOBE approach is that the hyperellipsoidal bounding sets are sometimes quite "loose" supersets of the exact feasibility sets (polytopes) (e.g. [1],[10]), particularly in "finite" time. However, many simulation studies in the literature (white noise case) have shown the volume of the ellipsoids to become quite small in the "long term." Further, as we and other researchers have demonstrated, the empirical convergence and tracking properties of the UOBE estimator are favorable in spite of the few data used. This is an indication that the *presence* of the ellipsoid and the optimization procedure centered on it, are quite useful for signal processing, regardless of our present inability to completely understand its behavior in theory. The results presented above offer further support for "good behavior" of this class of algorithms by indicating that the ellipsoid measures will converge to some unspecified size in some unspecified manner. This result has not been clearly understood, and its finding offers some hope that a proof of convergence for the UOBE algorithms may be found in the white noise case.

Colored Noise. Whether or not the UOBE ellipsoid can ultimately be proven to converge to a point with white noise disturbances, such a result would cause a contradiction in the colored noise case. This is so because if $\lim_{n\to\infty} \Omega(n)$ is a single point, then this implies that $\lim_{n\to\infty} \Theta(n) = \Theta_*$ in violation of the basic principles of least square estimation (the estimate must be biased). We therefore conclude the following:

Proposition 2 *With colored noise disturbances,* $\lim_{n\to\infty} \Omega(n)$ *is a nontrivial set.*

Empirical evidence leads to the following conjecture:

Conjecture 1 *With a p.e. input,* Θ_* *is on the boundary of the limiting set.*

Of course Proposition 2 is not suprising for a non-p.e. excitation for which the algorithm will cease to accept data in finite time due to lack of innovation. However, that the ellipsoid should remain nontrivial for p.e. inputs is not as apparent, since each time a data set is accepted, the set measure $\mu(n)$ must be diminished. This brings us to another interesting issue centered on the set measure used in the optimization.

Persistency of Excitation. In the following, we focus on the volume and trace measures, but the discussion might be generalizable in certain ways to broader classes of measures. The lengths of the ellipsoid axes at time n are inversely proportional to the square roots of the eigenvalues, say e_i, $i = 1,\dots,m$, of the matrix $C(n)/\kappa(n)$. Accordingly, convergence of the ellipsoid to a single point requires that $e_i \to \infty$ for all i. $\lim_{n\to\infty} \Omega(n)$ remains nontrivial iff one or more of the e_i remains finite. This implies that, in the presence of colored disturbances, $\lim_{n\to\infty} \mu_t(n)$ must be positive since one or more finite eigenvalues will make this so. On the other hand, μ_v becomes zero much more readily, because a single *infinite* eigenvalue is sufficient to cause $\lim_{n\to\infty} \mu_v(n) = 0$. That is, the ellipsoid need only "collapse in one dimension" to assure zero volume. In this sense, μ_v is a "weaker" set measure than μ_t.

The behavior of $\kappa(\cdot)$ is not well enough understood to make definitive conclusions about the conditions under which limiting ellipsoid may remain nontrivial in fewer than m dimensions. This has implications for both white and colored noise. For white noise, this means we do not know when (if ever) the ellipsoid will collapse to a point. For colored noise, it simply means we do not know when the ellipsoid will collapse into a subspace of the parameter space (since it must remain nontrivial). However, use of the volume measure permits an intriguing situation to arise precisely because of its "weakness." We believe that this situation points to the general condition under which the collapse may occur.

The asymptotic *volume* measure may be zero even if the ellipsoid is of *infinite* extent in one or more dimensions. That is, the *finite* eigenvalue(s) which imply a nontrivial limiting set may be *zero*. (Clearly, this cannot be the case with the trace measure since it would imply that $\lim_{n\to\infty} \mu_t(n) = \infty$.) Under what conditions might this "degenerate" volume situation occur? It is tempting to surmise that a p.e. input would be necessary to drive the volume ultimately to zero. However, precisely the opposite is true. A singular $\lim_{n\to\infty} C(n)/\kappa(n)$ occurs iff $\lim_{n\to\infty} C(n)$ is singular. In turn, this is indicative of a non-p.e. input. In this case the ellipsoid expands without bound in the null space of $\lim_{n\to\infty} C(n)/\kappa(n)$, while it must collaspe in (at least one dimension of) the range space in order to prevent the volume from

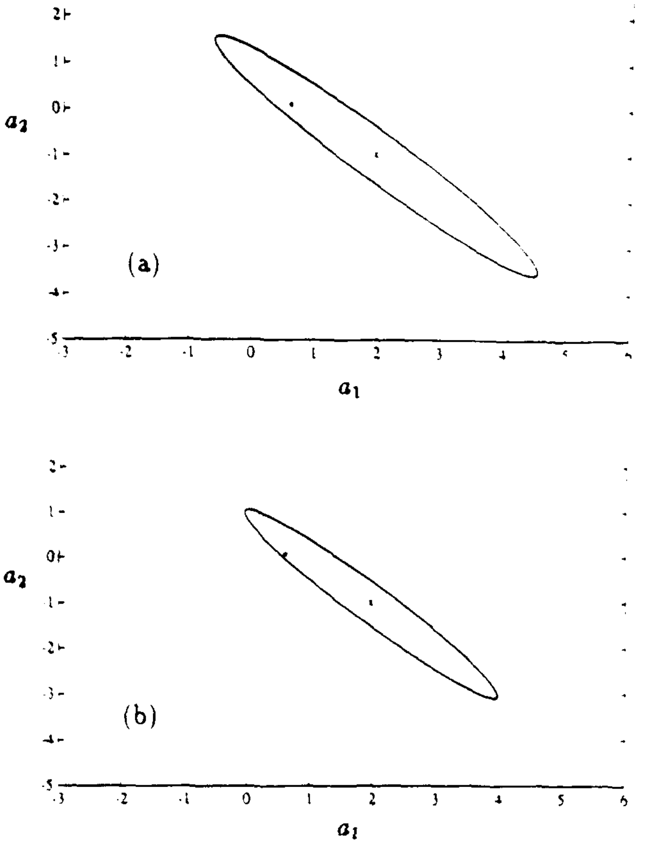


Fig. 2: "Asymptotic" ellipsoids resulting from first example. In each case, * denotes the true parameters and × the estimate (ellipsoid center). (a) $\gamma = 2$. (b) $\gamma = 1.25$.

also diverging. If the ellipsoid converges in every dimension of the range space, then the feasible set exists entirely within the null space of $\lim_{n\to\infty} C(n)/\kappa(n)$, except for the intersection (point, line, plane, hyperplane) with the range space. This analysis leads to the following conjecture:

Conjecture 2 *"Degeneracy" of the limiting ellipsoid (collapse of $\lim_{n\to\infty}\Omega(n)$ into a subspace), will occur in the volume case iff the input is non-p.e.*

Examples. In order to illustrate these ideas, we present two simple examples. In the both examples an AR(2) model of the form $y(n) = a_{1*}y(n-1) + a_{2*}y(n-2) + \epsilon_*(n)$ is used. In the first example the parameters are $a_{1*} = 0.6$, $a_{2*} = 0.1$, and $\epsilon_*(\cdot)$ is a realization of the stochastic process $\sqrt{2}\cos[(\pi n/16)+\xi)$ with ξ a uniformly distributed random phase. This noise is p.e. of order two. Two identifications were performed on this system using volume optimization. In the first, $\gamma(n)$ is ("properly") chosen to be the constant $\gamma = 2 = \epsilon_{*,\max}(\cdot)$, while in the second experiment $\gamma = 1.25$ in slight violation of the proper bound. The "asymptotic" ($n = 7000$) ellipsoids are shown in Figs. 2(a) and 2(b), respectively. In the first case 128/7000 (1.8%) of the data were selected by the optimization procedure, while in the second, 101/7000 (1.4%) were used. In *both* experiments the parameter estimates are identical to six decimal places: $a_1(7000) = 1.961508$, $a_2(7000) = -0.999855$. Both outcomes adhere to Proposition 2 in the production of a nontrivial limiting set with a biased estimate. Some support is seen for Conjecture 1 in the proximity of the true parameters to the ellipsoid boundary.

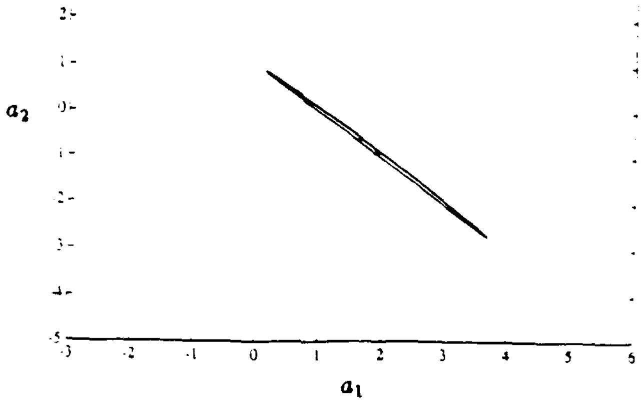


Fig. 3: "Asymptotic" ellipsoid resulting from second example.

Interestingly, the second experiment produces a more desirable outcome in this regard, and does so using fewer data, in spite of the bound violation. UOBE algorithms are remarkably robust to, and indeed sometimes benefit from, such violations and other

In the second example, $a_{1*} = 1.6$, $a_{2*} = -0.65$, and the system is excited by a constant $\epsilon_*(\cdot) = 0.322$ selected by choosing a value from a random realization, at a random time, in the cosine process above. Accordingly, $\gamma(\cdot)$ is taken as the constant $\gamma = 2 = \epsilon_{*,\max}(\cdot)$. This noise is p.e. of order one, and is therefore not sufficient to uniquely identify the system. In this case 30/7000 (0.43%) of the data were used. The ellipsoid in Fig. 3 is the resulting set at time $n = 7000$. A collapsing dimension is apparent as the covariance matrix becomes singular and the feasibility set begins to occupy the null space of the ellipsoid matrix.

[1] G. Belforte, B. Bona, and V. Cerone, "Identification, structure selection and validation ... with SM error...," *Math. and Computers in Simul.*, vol. 32, 561-569, 1990.

[2] S. Dasgupta and Y.F. Huang, "Asymptotically convergent modified RLS with data dependent updating ... ," *IEEE Trans. Information Theory*, vol. 33, 383-392, 1987.

[3] J.R. Deller, Jr., "SM ID in DSP," *IEEE ASSP Magazine*, vol. 6, 4-22, 1989.

[4] J.R. Deller, Jr. and T.C. Luk, "Linear prediction ... based on SM," *Computer Speech and Lang.*, vol. 3, 301-327, 1989.

[5] J.R. Deller, Jr., M. Nayeri, and S.F. Odeh, "System ID using SM-based signal processing," *IEEE Proc.* (in review).

[6] E. Fogel and Y.F. Huang, "On the value of information in system ID ... ," *Automatica*, vol. 18, 229-238, 1982.

[7] G.C. Goodwin and K.S. Sin, *Adaptive Filtering, Prediction, and Control*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[8] M.S. Liu, M. Nayeri, and J.R. Deller, Jr., Unpublished research notes, Michigan State University, 1991.

[9] J.P. Norton, *An Introduction to Identification*, London and Orlando, Florida: Academic Press, 1986.

[10] J.P. Norton, "... computation of inner bounds for the parameters of linear models," *Int. J. Control*, vol. 50, 2423-2430, 1989.

Centre National de la Recherche Scientifique — Ecole Supérieure d'Electricité

LABORATOIRE DES SIGNAUX & SYSTEMES
Associé à l'Université Paris-Sud

ECOLE SUPERIEUFE D'ELECTRICITE
Plateau de Moulon · 91192 GIF-s/YVETTE cedex (FRANCE)
Tél : (1) 69.41.80.40 · Télécopie : (1) 69.41.30.60

Professor J. Deller
Department of Electrical Engineering and System Science
Michigan State University
260 Engineering Bldg
East Lansing
MI 48824-1226
USA

Gif, February 1, 1990

Dear Professor Deller:

We have just received the first announcement for the 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation to be held in Budapest, Hungary, July 8-12 1991. This congress which takes place every three years is, as you probably know, the largest international conference devoted to identification problems. We feel it is a unique opportunity to present the state of the art in estimation in the bounded-error context and to bring together the bounded-error community. We therefore propose to organize one or several session(s) on this topic, and would like to invite you to participate.

The proposal for sessions must be submitted to the organizing committee by March 31, 1990. The final decision for the selection of sessions will be announced by April 30, 1990, so that if the session was not accepted you could still submit papers individually as the deadline for paper submission is July 15, 1990.

If you are interested in participating to such a session, could you please let us know by sending us a prospective title with a list of authors and a short abstract as soon as possible and to be received no latter than March 15th 1990. The length of the final paper should not exceed 6 Pergamon laysheets.

We hope that our proposal will be accepted and the resulting session(s) will be as rewarding as the IMACS Conference in Paris has been.

E. Walter — H. Piet-Lahanier

SM-WRLS ALGORITHMS WITH AN EFFICIENT TEST FOR INNOVATION[1]

J.R. Deller, Jr.[†] *and S.F. Odeh*[‡]

[†]Michigan State University
Department of Electrical Engineering
CSSP Group: Speech Processing Lab
East Lansing, MI 48824-1226 USA

[‡]University of Wisconsin—Platteville
Department of Electrical Engineering
1 University Plaza
Platteville, WI 53818-3099 USA

Abstract

This paper is concerned with the *set membership weighted recursive least squares (SM-WRLS)* algorithm which can be used for estimating the parameters of linear system or signal models in which the error sequence is pointwise "energy bounded." This algorithm works with bounding hyperellipsoidal regions to describe the solution sets. A new strategy is developed which can be applied to virtually any version of the SM-WRLS algorithm to improve the computational efficiency. A significant reduction in computational complexity can be achieved by employing a "suboptimal" test for information content in an incoming equation. The proposed check is argued to be a useful determiner of the ability of incoming data to shrink the ellipsoid. The performance of this strategy is compared to that of SM-WRLS using simulation studies.

keywords: *bounded-error identification, parameter estimation, adaptive estimation, set-membership theory, system identification*

1 Introduction

The *set membership weighted recursive least squares (SM-WRLS)* algorithm [1, 2, 3] is an efficient technique which can be used for estimating the parameters of linear system or signal models under *a priori* information which constrains the solutions to certain sets. When data do not help refine these membership sets, the effort of updating the parameter estimates at those points can oe avoided. The SM-WRLS algorithm is concerned with the case in which the error sequence, say $v(n)$, is pointwise "energy bounded,"

$$\gamma(n)v^2(n) < 1 \tag{1}$$

where the sequence $\gamma(n)$ is known or can be estimated from the data. Constraints of form (1), in conjunction with the model and data, imply pointwise "hyperstrip" regions of possible parameter sets in the parameter space which, when intersected over a given time range, usually form convex polytopes of permissible solutions for the "true" parameters. While exact descriptions of these polytopes are possible (e.g., see [4]), algorithms of much lower complexity have been developed which work with a bounding hyperellipsoid, a tight superset of the polytope [1, 2, 3, 5, 6, 7]. The SM-WRLS algorithm is one such algorithm which is formulated such that it is exactly the familiar weighted recursive least squares solution [8, 9] with the SM considerations handled through a special weighting strategy. A tutorial on the SM-WRLS algorithm, and on general SM identification is found in [2].

One of the advantages of the SM-WRLS formulation is that it immediately admits solution by contemporary systolic array processors for speed advantages [1]. The systolic array implementations of this algorithm are presented in [10, 11]. In this paper, however, we will develop a "suboptimal" strategy which can be applied to virtually any version, adaptive or "non-adaptive," of the SM-WRLS algorithm to improve the computational efficiency. The theoretical development of this strategy is presented in Section 2. In Section 3, the performance of this strategy is compared to that of SM-WRLS using simulation studies. A detailed analysis of the computational complexity issues is found in Section 4.

2 Theoretical Development

2.1 The SM-WRLS Algorithm

We consider the estimation of the parameters of a general ARMAX(p,q) [9] model of the form

$$\begin{aligned} y(n) &= \sum_{i=1}^{p} a_i y(n-i) + \sum_{j=0}^{q} b_j w(n-j) + v(n) \qquad (2)\\ &\doteq \boldsymbol{\theta}_0^T \boldsymbol{x}(n) + v(n) \end{aligned}$$

in which $y(n)$ is a scalar output; $w(n)$ is a measurable, uncorrelated, input; and $v(n)$ is an uncorrelated process, known to be bounded as in (1), which is independent of $w(n)$. For convenience, we also employ the vector notations

$$\boldsymbol{x}^T(n) \doteq [y(n-1)\cdots y(n-p)w(n)w(n-1)\cdots w(n-q)] \tag{3}$$

and

$$\boldsymbol{\theta}_0^T \doteq [a_1 a_2 \cdots a_p b_0 b_1 \cdots b_q] \tag{4}$$

$\boldsymbol{\theta}_0$ represents the vector of parameters to be estimated. We define the integer $m = p + q + 1$ noting that m should be reduced to simply $m = p$ for the pure AR case.

Let us define $\hat{\boldsymbol{\theta}}(N)$ to be the conventional weighted LS estimate of $\boldsymbol{\theta}_0$ using the data on the range $n = 1, 2, \ldots, N$, with squared error minimization weights $\lambda(n)$. Also denote the (weighted) covariance matrix for the data by $\boldsymbol{C}(N)$. As a consequence of the bounds on the sequence $v(n)$, at time N there is a computable hyperellipsoidal domain in the parameter space which *certainly* contains $\boldsymbol{\theta}_0$ and which is centered on the LS estimate. This set is given by [3]

$$\Omega(N) = \left\{ \boldsymbol{\theta} \mid (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(N))^T \frac{1}{\kappa(N)} \boldsymbol{C}(N)(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(N)) < 1 \right\}, \tag{5}$$

$\boldsymbol{\theta} \in \Re^m$, where,

$$\begin{aligned} \kappa(N) &= \hat{\boldsymbol{\theta}}^T(N)\boldsymbol{C}(N)\hat{\boldsymbol{\theta}}(N) + \sum_{n=1}^{N} \frac{\lambda(n)}{\gamma(n)} \left(1 - \gamma(n)y^2(n)\right) \qquad (6)\\ &= \| \mathbf{d}_1(N) \|^2 + \tilde{\kappa}(N) \qquad (7) \end{aligned}$$

in which $\| \cdot \|$ denotes the Euclidean norm,

$$\boldsymbol{d}_1(N) = \boldsymbol{T}(N)\hat{\boldsymbol{\theta}}(N) \tag{8}$$

with $\boldsymbol{T}(N)$ the upper triangular Cholesky factor [12] of $\boldsymbol{C}(N)$ (more on this below), and $\tilde{\kappa}(N)$ denotes the sum on the right side of (6). It is useful to note that

$$\tilde{\kappa}(n) = \tilde{\kappa}(n-1) + \frac{\lambda(n)}{\gamma(n)} \left(1 - \gamma(n)y^2(n)\right), \quad \tilde{\kappa}(0) = 0 \tag{9}$$

[1] This work was supported in part by the National Science Foundation of the United States under Grant No. MIP-9016734, and in part by the Office of Naval Research under Contract No. 4148126-01

Very importantly, the "size" of this domain is a function of only one unknown at time n, $\lambda(n)$, the error minimization weight at time n. The "SM strategy" of updating the parameter vector at time n involves the computation of the $\lambda(n)$ which minimizes the size of $\Omega(n)$. The volume of the ellipsoid, $\Omega(n)$, is proportional to the quantity

$$\det \boldsymbol{B}(n) \doteq \det \kappa(n)\boldsymbol{C}^{-1}(n) = \frac{\kappa^{m}(n)}{\det \boldsymbol{C}(n)} . \qquad (10)$$

A reasonable strategy is to find an optimal weight, $\lambda^{*}(n)$, at each step which minimizes the "volume ratio" of the ellipsoids at n and $n-1$:

$$V(\lambda(n)) = \frac{\det \boldsymbol{B}(n)}{\det \boldsymbol{B}(n-1)} . \qquad (11)$$

This weight is taken to be the most positive root of the quadratic equation [3]

$$F(\lambda) = \alpha_2\lambda^2 + \alpha_1\lambda + \alpha_0 = 0 \qquad (12)$$

where,

$$\begin{aligned}
\alpha_2 &= (m-1)G^2(n) \\
\alpha_1 &= \{2m - 1 + \gamma(n)\epsilon^2_{n-1}(n) - \kappa(n-1)\gamma(n)G(n)\}G(n) \\
\alpha_0 &= m[1 - \gamma(n)\epsilon^2_{n-1}(n)] - \kappa(n-1)\gamma(n)G(n)
\end{aligned}$$

and where,

$$G(n) \doteq \boldsymbol{x}^{T}(n)\boldsymbol{C}^{-1}(n-1)\boldsymbol{x}(n) \qquad (13)$$

and $\epsilon_{n-1}(n)$ is the residual at time n based on the parameter estimate at $n-1$,

$$\epsilon_{n-1}(n) = y(n) - \dot{\boldsymbol{\theta}}^{T}(n-1)\boldsymbol{x}(n) . \qquad (14)$$

One important consequence of this approach is that often no $\lambda(n)$ exists which will further shrink $\Omega(n)$. Generally, this means that the equation[2] $(y(n), \boldsymbol{x}(n))$ at n has "no information" which has not already been incorporated into the estimate. In this case the equation is "rejected" ($\lambda(n)$ effectively set to zero) saving the computational expense otherwise necessary to incorporate it.

In a recent paper [1], SM-WRLS was formulated into a more contemporary WRLS algorithm which is amenable to a systolic architecture implementation. The algorithm is given in the Appendix of this paper. In Steps 4 and 5 of the algorithm, the LS problem is solved using a sequential "QR" decomposition using *Givens rotations (GR's)*, a method which is well-understood and becoming widely used for this purpose [13, 14, 15]. (It is important to note the meaning of the matrix $\boldsymbol{T}(n)$ in this process. We first encountered $\boldsymbol{T}(n)$ in (8) where it was defined as the upper-triangular Cholesky factor matrix of $\boldsymbol{C}(n)$ at each step, i.e., $\boldsymbol{C}(n) = \boldsymbol{T}^{T}(n)\boldsymbol{T}(n)$ (see Appendix)). The more novel part of the algorithm in Steps 1, 2, 3 and 6, is concerned with the computation of the optimal weights. Here the method had to be designed to avoid the costly inversion of the matrix $\boldsymbol{C}(n)$, nominally necessary to compute $G(n)$. The quantity $\kappa(n)$ is also efficiently computed in this context. The reader is referred to [1] for details.

2.2 Adaptive SM-WRLS Algorithm

In this section, we present an *adaptive* SM-WRLS algorithm with a very flexible mechanism by which it can "forget" the influence of past data.

The adaptive algorithm presented here uses "back rotation" in order to partially or completely "forget" past information enabling it to track (potentially fast) time varying signals. Back rotation [13] is a Givens rotation-based technique that removes (or rotates out) a previously included equation from the system. In this paper we modify the back rotation so that a previous equation can be *partially* removed. This will permit a broader class of adaptive strategies. In SM terms, back rotation causes the ellipsoidal membership set to expand due to the removal of information. This expansion entices the algorithm to incorporate present data. The back rotation technique requires that all the weights with the corresponding equations (for weights other than zero) be stored for later use.

In the Appendix, we see that at each step in the SM-WRLS algorithm, the upper triangular system of simultaneous equations $\boldsymbol{T}(n)\dot{\boldsymbol{\theta}}(n) = \boldsymbol{d}_1(n)$, is solved (when data are accepted) to obtain the optimal estimate [2, 3, 10]. Suppose in approaching time n that the past equation to be (partially) removed is at time τ. Rotating this equation out of the system is accomplished by re-introducing it as though it were a new equation. A weight $\sqrt{\mu\lambda(\tau)}$, where μ is the fraction of the equation to be removed from the system, is used, and some sign changes in the rotation equations are necessary [13]. Let us refer to the system of equations with τ removed as the "downdated" system at time $n-1$, and label the related quantities with subscript d, i.e.,

$$\boldsymbol{T}_d(n-1)\dot{\boldsymbol{\theta}}_d(n-1) = \boldsymbol{d}_{1,d}(n-1) . \qquad (15)$$

The downdated ellipsoid matrix is $\boldsymbol{C}_d(n-1)/\kappa_d(n-1)$ where

$$\begin{aligned}
\boldsymbol{C}_d(n-1) &= \boldsymbol{T}^{T}_d(n-1)\boldsymbol{T}_d(n-1) , &\qquad (16)\\
\kappa_d(n-1) &= \| \boldsymbol{d}_{1,d}(n-1) \|^2 + \tilde{\kappa}_d(n-1) &\qquad (17)
\end{aligned}$$

with

$$\tilde{\kappa}_d(n-1) \doteq \tilde{\kappa}(n-1) - \frac{\mu\lambda(\tau)}{\gamma(\tau)}\left(1 - \gamma(\tau)y^2(\tau)\right) . \qquad (18)$$

Equations (17) and (18) follow immediately from the definition of κ found in (6). These relations can be used repeatedly regardless of the number of equations (partially or completely) removed prior to time n. If more than one equation is removed prior to n, $\tilde{\kappa}(n-1)$ in the right-hand side of (18) is replaced by $\tilde{\kappa}_d(n-1)$ for all downdates after the first one. Following all necessary downdating just prior to time n, the algorithm uses the downdated system to compute the quantities $G_d(n)$ and $\epsilon_{n-1,d}(n)$ which are necessary to compute the optimal weight for the equation at n. To compute a downdated SM-WRLS estimate, therefore, it is only necessary to downdate the matrix $\boldsymbol{T}(n-1)$ and the vector $\boldsymbol{d}_1(n-1)$ and to solve for $\dot{\boldsymbol{\theta}}_d(n-1)$ prior to Step 1, then replace all relevant quantities in Step 1 by their downdated versions. $\kappa_d(n-1)$ and $\tilde{\kappa}_d(n-1)$ are downdated according to (17) and (18). Then $\lambda^{*}(n)$ is found in Step 2 using (12) with downdated quantities. Note that downdating is unnecessary if the equation τ was rejected by SM-WRLS. In this case $\boldsymbol{T}_d(n-1) = \boldsymbol{T}(n-1)$ and $\dot{\boldsymbol{\theta}}_d(n-1) = \dot{\boldsymbol{\theta}}(n-1)$. Conversely, when the "new" equation at n is rejected, then $\boldsymbol{T}(n) = \boldsymbol{T}_d(n-1)$ and $\dot{\boldsymbol{\theta}}(n) = \dot{\boldsymbol{\theta}}_d(n-1)$.

A wide range of adaptation strategies is inherent in the general formulation described above. Three major subcases are identified (windowing, graceful forgetting, and selective forgetting) in [11, 16]. In each of these subcases, the objective is to expand the ellipsoidal region of possible solutions in order to track fast time variations in the signal. For illustration and comparison purposes in the simulations below, we use one of the adaptive strategies, namely, the *selective forgetting* in which the equations are removed from the estimate according to certain user-defined criteria in order to remove their influence on the result. The selection criterion used here is to remove the equations starting from the first accepted equation remaining in the estimate at a given time, and proceeding sequentially until some other condition is satisfied. The determination of when to apply the forgetting procedure and when to stop removing equations at a given time is discussed in [16].

2.3 Suboptimal Test for Innovation

A significant reduction in computational complexity can be achieved by employing a "suboptimal" test for information content in an

[2] Since the LS process represents an effort to fit the m model parameters to N equations of the form $y(n) = \boldsymbol{\theta}^{T}\boldsymbol{x}(n)$, we refer to the pair $(y(n), \boldsymbol{x}(n))$ as an "equation" throughout this paper.

incoming equation. The proposed check is argued to be a useful determiner of the ability of incoming data to shrink the ellipsoid, but it does not rigorously determine the existence of an optimal SM weight in the sense described above. The main issue here is to avoid the computations of the quantities necessary at each step to construct and solve the quadratic (12) in cases in which the quadratic turns out only to be useful for the purpose of checking for the existence of a meaningful weight. Since most of the time these computations result in the rejection of incoming data, a more efficient test could significantly reduce the complexity of the algorithm.

The estimation error vector at time n can be denoted by

$$\tilde{\theta}(n) \doteq \theta_0 - \hat{\theta}(n) . \qquad (19)$$

The following inequality results immediately from (5),

$$\tilde{\theta}^T(n) C(n) \tilde{\theta}(n) < \kappa(n) . \qquad (20)$$

Using a similar inequality, Dasgupta and Huang [6] have noted that their $\kappa(n)$-like quantity provides a bound on the error vector sequence and have suggested minimizing this quantity with respect to $\lambda(n)$ in an effort to decrease computational complexity. However, this minimization does not, in general, imply an improvement in the estimate with respect to previous times, since both sides of the inequality (20) are dependent upon $\lambda(n)$. Further, the nonexistence of a minimum of $\kappa(n)$ with respect to $\lambda(n)$ is not very informative in this sense. However, further arguments are presented here to provide support for this process in the SM-WRLS context.

Consider the usual volume quantity to be minimized at time n, defined in (10). Let us temporarily write the two key quantities there as functions of $\lambda(n)$: $C(n, \lambda(n))$ and $\kappa(n, \lambda(n))$. It is assumed that enough equations have been included in the covariance matrix at time $n-1$ so that its elements are large with respect to the data in the incoming equation. Now the quantity $\det C(n, \lambda(n))$ is readily shown to be monotonically increasing with respect to $\lambda(n)$ on $\lambda(n) \in [0, \infty)$ [16], with $C(n, 0) = C(n-1, \lambda^*(n-1))$, where $\lambda^*(n-1)$ indicates the optimal weight at time $n-1$. Under the assumption above, $\det C(n, \lambda(n))$ will not increase significantly over reasonably small values of $\lambda(n)$. The attempt to maximize $\det C(n, \lambda(n))$ in (10) causes a tendency to increase $\lambda(n)$ in the usual optimization process. However, the attempt to minimize $\kappa(n, \lambda(n))$ generally causes a tendency toward small values of $\lambda(n)$, unless a minimum of $\kappa(n, \lambda(n))$ occurs at a "large" value of $\lambda(n)$. To pursue this idea and further points of the argument, key results about $\kappa(n, \lambda(n))$ are noted in the following.

Theorem 1 *$\kappa(n, \lambda(n))$ has the following properties:*

- *On the domain $\lambda(n) \in [0, \infty)$, $\kappa(n, \lambda(n))$ is either monotonically increasing or it has a single minimum.*
- *$\kappa(n, \lambda(n))$ has a minimum on $\lambda(n) \in [0, \infty)$ iff*

$$\epsilon^2_{n-1}(n) > \gamma^{-1}(n) . \qquad (21)$$

Lemma 1 *[9]. Let $\lambda^*(n-1)$ denote the optimal weight in the sense of (12) (which might be zero) at time n − 1. Then*

$$\kappa(n, \lambda(n)) = \kappa(n-1, \lambda^*(n-1)) + \frac{\lambda(n)}{\gamma(n)} - \left[\frac{\lambda(n)\epsilon^2_{n-1}(n)}{1 + \lambda(n)G(n)}\right] . \qquad (22)$$

Proof of Theorem 1 The minimum of $\kappa(n, \lambda(n))$ with respect to $\lambda(n)$ can be found by differentiating (22) and setting the result equal to 0,

$$\frac{\partial \kappa(n, \lambda(n))}{\partial \lambda(n)} = G^2(n)\lambda^2(n) + 2G(n)\lambda(n) + \left[1 - \gamma(n)\epsilon^2_{n-1}(n)\right] \equiv 0 . \qquad (23)$$

This is a concave upward quadratic function with its minimum at

$$\lambda'(n) = -G^{-1}(n) < 0 . \qquad (24)$$

Two *real roots* of (23) always exist,

$$\lambda_{roots}(n) = \frac{-1 \pm \sqrt{\gamma(n)}\epsilon_{n-1}(n)}{G(n)} \qquad (25)$$

the smaller corresponding to a maximum of $\kappa(n, \lambda(n))$, the larger to a minimum. Only the larger root can be positive since the lower root is bound to be less than $\lambda'(n)$. Therefore, it is only possible for $\kappa(n, \lambda(n))$ to exhibit a minimum or to be increasing on positive $\lambda(n)$. It is easy to use (25) to verify that the larger root is positive *iff* condition (21) is met. ⊐

With these results, it can be argued that: If $\det C(n, \lambda(n))$ is increasing, but not changing significantly over reasonably small values of $\lambda(n)$, then it is sufficient to seek $\lambda(n)$ which minimizes $\kappa(n, \lambda(n))$. If $\kappa(n, \lambda(n))$ is monotonically increasing on $\lambda(n) \geq 0$, this value is $\lambda(n) = 0$ which corresponds to rejection of the equation at time n. It suffices, therefore to have a test for a minimum of $\kappa(n, \lambda(n))$ on positive $\lambda(n)$. As noted above, a simple test is embodied in condition (21). If this test is met, it is then cost effective to proceed with the standard optimization process centered on (12). Otherwise, the explicit construction and solution of (12) can be avoided.

It is to be noted that even if (21) is met, it is possible that the optimization procedure will still reject the datum. Perhaps more importantly, it is also possible for (21) to reject data which would have been accepted by the usual process. These ideas will be explored in the simulation studies below.

Finally, note that when the simplified test (21) accepts the new equation, there are tools to compute the weight which is "optimal" in the sense of minimizing $\kappa(n, \lambda(n))$. In particular, this would be the larger of the roots in (25). However, it clearly makes more sense to compute *the* optimal weight according to (12), since this computation is not much more expensive. The improvement in the computational complexity due to "suboptimal checking" is discussed in Section 4. It is important to note that the general adaptive formulation of Section 2.2 is amenable to the suboptimal technique described here. The performance of the suboptimal and the *adaptive* suboptimal techniques will be investigated in the next section.

3 Simulation Studies

In this section, we consider the estimation of the parameters of two time varying AR(2) models of the form

$$y(n) = a_1(n)y(n-1) + a_2(n)y(n-2) + v(n) . \qquad (26)$$

Two sets of AR parameters were derived using linear prediction (LP) analysis of order two on utterances of the words "four" and "six" by an adult male speaker. The data were sampled at 10 kHz after 4.7 kHz lowpass filtering, and the "forgetting factor" in the LP algorithm (see [17]) was [illegible] = 0.996. A 7000 point sequence, $y(n)$, for each case ("four" and "six") was generated by driving the appropriate set of parameters with an uncorrelated sequence, $v(n)$, which was uniformly distributed on $[-1, 1]$. In the simulations below, we apply the conventional and suboptimal SM-WRLS algorithms to the estimation of the a_i parameters.

We discuss a number of simulation results. To conserve space, only the result for a_1 is illustrated in each case. Each figure shows two curves, one for the true parameter, the other for the estimate obtained by the algorithm under study.

Figures 1 and 2 show the simulation results of the conventional SM WRLS algorithm for the words four and six using only 1.86% and 2.16% of the data, respectively. Figures 3 and 4 show the simulation results of the conventional SM-WRLS algorithm with suboptimal data selection. In this case, only 1.19% and

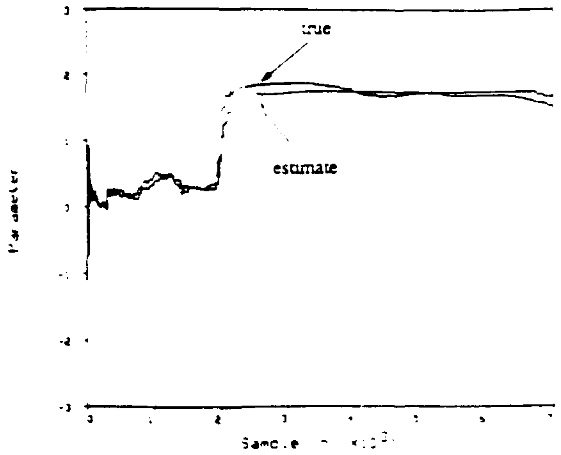


Figure 1: Simulation results of the SM-WRLS algorithm for the word four. 1.86% of the data is employed in the estimation process.

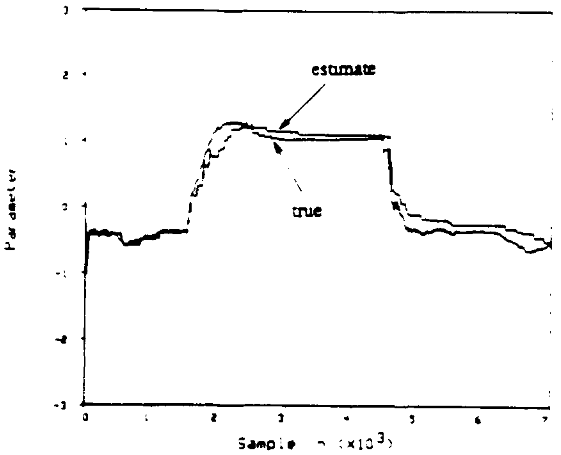


Figure 2: Simulation results of the SM-WRLS algorithm for the word six. 2.16% of the data is employed in the estimation process.

1.53% of the data are used for the words four and six, respectively. Compared to the conventional SM-WRLS algorithm (see Figs. 1 and 2), the suboptimal technique uses slightly fewer data but produces comparable estimates. It is interesting to note that *most* of the equations (97.6% for the word four and 94.4% for the word six) that are accepted by the suboptimal technique are also accepted by the conventional SM-WRLS algorithm. It is also interesting to note that the equations that are accepted by the suboptimal technique but not by the conventional SM-WRLS algorithm lie mostly in regions of fast changing dynamics.

Figures 5 and 6 show the simulation results of the selective forgetting adaptive strategy. This strategy uses only 3.6% and 2.83% of the data for the words four and six, respectively. More data than with the conventional SM-WRLS algorithm are used, but more accurate estimates result and the time varying parameters are tracked more quickly and accurately. This can be easily seen when the parameter dynamics change abruptly near the point 2100 for the word four (see Fig. 5) and near the points 2000 and 4500 for the word six (see Fig. 6).

We have noted that the general formulation of the adaptive SM-WRLS algorithm is amenable to the suboptimal technique.

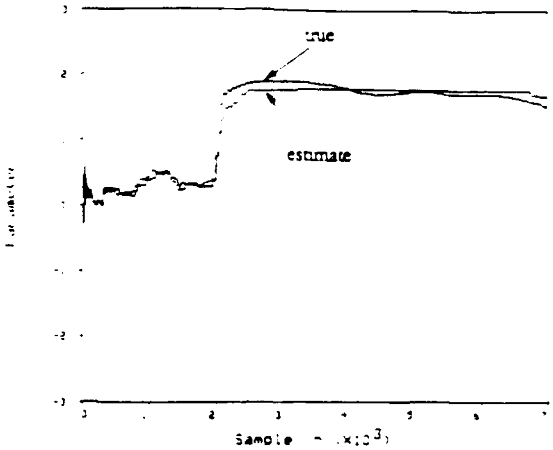


Figure 3: Simulation results of the SM-WRLS algorithm with suboptimal data selection for the word four. 1.19% of the data is employed in the estimation process.

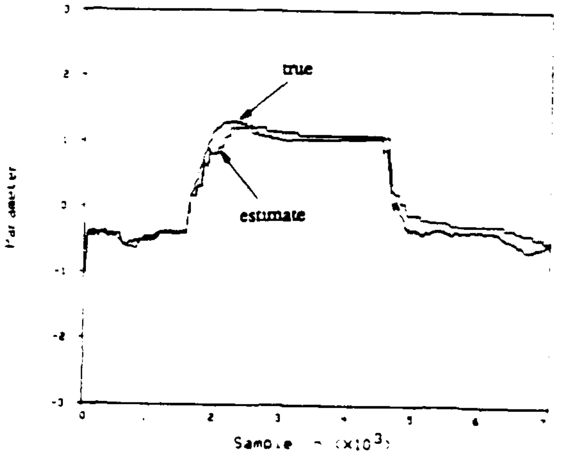


Figure 4: Simulation results of the SM-WRLS algorithm with suboptimal data selection for the word six. 1.53% of the data is employed in the estimation process.

The simulation results of the selective forgetting SM-WRLS technique with suboptimal data selection are shown in Figs. 7 and 8. This strategy uses only 1.89% and 1.86% of the data for the words four and six, respectively. Compared to the selective forgetting strategy (Figs. 5 and 6), the selective forgetting technique with suboptimal data selection uses fewer data but produces comparable estimates. On the other hand, when compared to conventional SM-WRLS with suboptimal data selection (Figs. 3 and 4), the selective forgetting suboptimal technique uses more data but produces better estimates.

4 Complexity Analysis

In order to perform a detailed analysis of the computational complexities, we employ the following notations: If the fraction of the data accepted by the conventional SM-WRLS algorithm is denoted by r, the fraction of the data accepted by the SM-WRLS algorithm with suboptimal data selection by s ($s < r$), and the fraction of the data accepted by the SM-WRLS algorithm after passing the test (21) by t ($t \leq s$), then the total computational

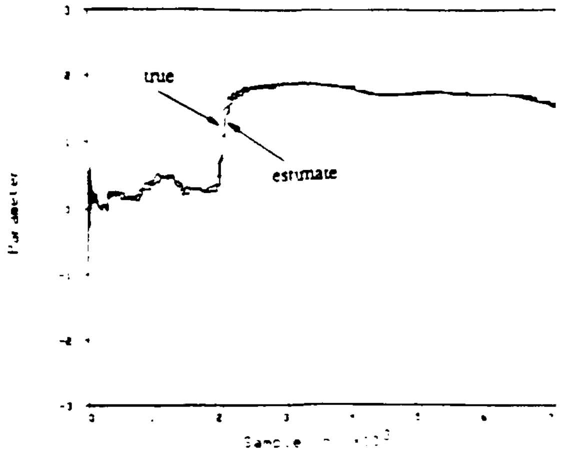


Figure 5: Simulation results of the selective forgetting SM-WRLS algorithm for the word four. 3.6% of the data is employed in the estimation process.

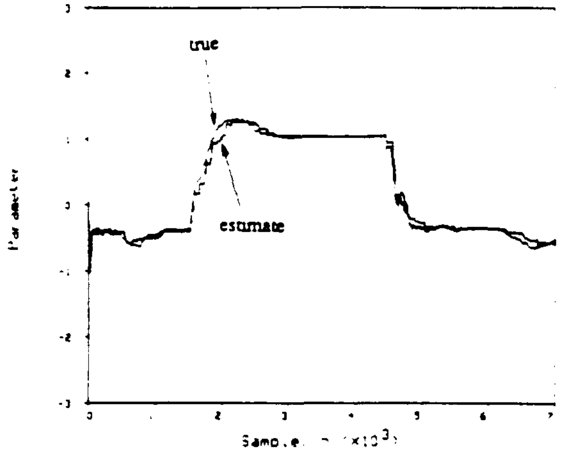


Figure 6: Simulation results of the selective forgetting SM-WRLS algorithm for the word six. 2.83% of the data is employed in the estimation process.

complexity of the conventional SM-WRLS algorithm is given by [16]

$$\left(m^2 + 2m + 13\right) + r\left[2m^2 + 3m + 7\right] \tag{27}$$

floating point operations (flops) per equation. For the SM-WRLS algorithm with suboptimal data selection, it is given by [16]

$$(m + 1) + s\left[m^2 + m + 12\right] + t\left[2m^2 + 3m + 7\right] \tag{28}$$

flops per equation. When considering a typical example to compare the complexities of the two strategies, the suboptimal strategy reduces the computational complexity of the conventional algorithm by 60 - 70%, which is clearly advantageous especially when noting that the simulation results of the two strategies are comparable.

If the fractions of the data used by the adaptive SM-WRLS algorithms are denoted by the same symbols used above, and the fraction of the data removed from the system is denoted by u, then the total computational complexity of the selective forgetting SM-WRLS algorithms is given by

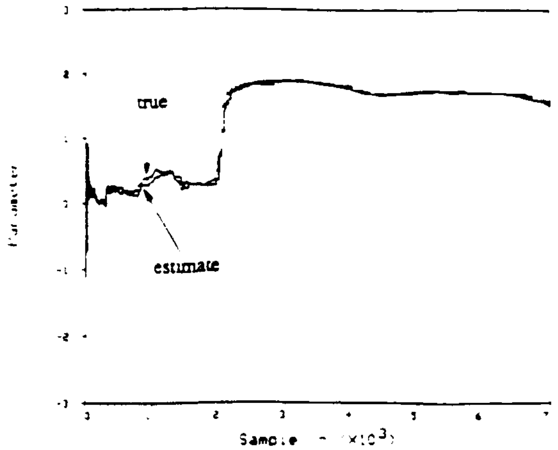


Figure 7: Simulation results of the selective forgetting SM-WRLS algorithm with suboptimal data selection for the word four. 1.89% of the data is employed in the estimation process.

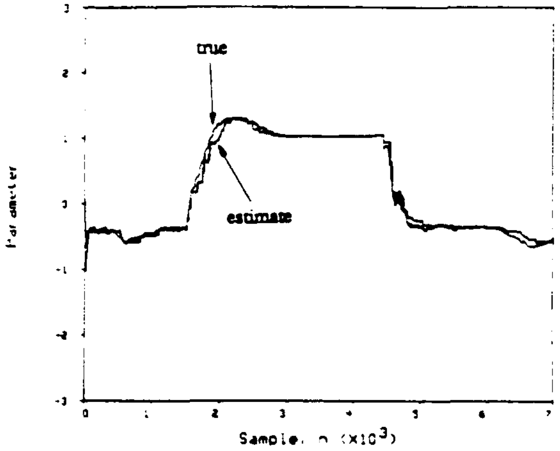


Figure 8: Simulation results of the selective forgetting SM-WRLS algorithm with suboptimal data selection for the word six. 1.86% of the data is employed in the estimation process.

$$\left(.5m^2 + 2.5m + 13\right) + r\left[2.5m^2 + 10.5m + 5\right] + u\left[2m^2 + 10m + 5\right] \tag{29}$$

flops per equation. For the suboptimal selective forgetting SM-WRLS algorithms, it is given by

$$\begin{aligned}(m + 1) + s\left[.5m^2 + 1.5m + 12\right] \\ +t\left[2.5m^2 + 10.5m + 5\right] + u\left[2m^2 + 10m + 5\right]\end{aligned} \tag{30}$$

flops per equation. Again, the computational complexity is reduced by 60 - 70%.

5 Conclusion

This paper presents a suboptimal data checking strategy for the SM-WRLS algorithm. It also shows how adaptation can be incorporated into SM-WRLS in a very general way by introducing a flexible mechanism by which the algorithm can forget the influence of past data. The suboptimal technique (which can be applied to virtually any version, adaptive or non-adaptive, of the SM-WRLS algorithm) uses many fewer data, is a square-root factor better in computational complexity, and produces comparable estimates to the optimal algorithm.

References

[1] J.R. Deller, Jr., "A 'systolic array' formulation of the optimal bounding ellipsoid algorithm," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 1432-1436, Sept. 1989.

[2] J.R. Deller, Jr., "Set membership identification in digital signal processing," *IEEE ASSP Magazine*, vol. 6, no. 4, pp. 4-20, Oct. 1989.

[3] J.R. Deller, Jr. and T.C. Luk, "Linear prediction analysis of speech based on set-membership theory," *Computer Speech and Language*, vol. 3, no. 4, pp. 301-327, Oct. 1989.

[4] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: A survey," *Proc. 12th IMACS World Congress on Scientific Computation*, Paris, pp. 467-472, July 1988.

[5] A.K. Rao, Y.F. Huang, and S. Dasgupta, "ARMA parameter estimation using a novel recursive estimation algorithm with selective updating," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 38, pp. 447-457, March 1990.

[6] 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 Trans. Information Theory*, vol. IT-33, pp. 383-392, 1987.

[7] Y.F. Huang, "A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-34, pp. 1331-1334, 1986.

[8] D. Graupe, *Time Series Analysis, Identification, and Adaptive Systems*, Malabar, FL: Krieger, Ch. 5, 1989.

[9] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, Cambridge, MA: MIT Press, pp. 14-15 & Sect. 2.2.1., 1983.

[10] J.R. Deller, Jr. and S.F. Odeh, "Implementing the optimal bounding ellipsoid algorithm on a fast processor," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '89*, Glasgow, vol. 2, pp. 1067-1070, May 1989.

[11] S.F. Odeh and J.R. Deller, Jr., "A systolic algorithm for adaptive set membership identification," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. '90*, Albuquerque, NM, vol. 5, pp. 2419-2422, April 1990.

[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Baltimore, MD: Johns-Hopkins Univ. Press, Ch. 5, 1983.

[13] J.R. Deller, Jr. and G.P. Picaché, "Advantages of a Givens rotation approach to temporally recursive linear prediction analysis of speech," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 429-431, March 1989.

[14] T.C. Luk and J.R. Deller, Jr., "A nonclassical WRLS algorithm," *Proc. 23rd Ann. Allerton Conf.*, pp. 732-741, 1985.

[15] W.M. Gentleman and H.T. Kung, "Matrix triangularization by systolic arrays," *Proc. Soc. Photo-Optical Instrumentation Engineers: Real-Time Signal Processing IV*, vol. 298, pp. 19-26, 1981.

[16] S.F. Odeh, *Algorithms and Architectures for Adaptive Set Membership-based Signal Processing* (Ph.D. Dissertation), Michigan State University, 1990.

[17] J.R. Deller, Jr. and D. Hsu, "An alternative adaptive sequential regression algorithm and its application to the recognition of cerebral palsy speech," *IEEE Trans. Circ. and Syst.*, vol. CAS-34, pp. 782-787, July 1987.

Appendix
SM-WRLS Algorithm Based on QR Decomposition

Initialization: Fill an $(m+1) \times (m+1)$ working matrix $\boldsymbol{W}$ with zeros. For $n = 1, \ldots, m+1$, set $\lambda(n) = 0$. Set $\bar{\kappa}(0) = 0$.
Recursion[3]: For $n = 1, \ldots, N$,

1. (Skip if $n \le m+1$ (see footnote).) Update $G(n), \varepsilon_{n-1}(n)$ as follows.
   Solve
   $$\boldsymbol{T}^T(n-1)\boldsymbol{g}(n) = \boldsymbol{x}(n)$$
   for $\boldsymbol{g}(n)$ by back-substitution. Compute $G(n) = \| \boldsymbol{g}(n) \|^2$ and
   $$\varepsilon_{n-1}(n) = y(n) - \hat{\boldsymbol{\theta}}^T(n-1)\boldsymbol{x}(n).$$

2. (Skip if $n \le m+1$ (see footnote).) Compute the optimal $\lambda^*(n)$ by finding the most positive root of (12).

3. (Skip if $n \le m+1$ (see footnote).) If $\lambda^*(n) \le 0$, set $\boldsymbol{T}(n) = \boldsymbol{T}(n-1)$, $\hat{\boldsymbol{\theta}}(n) = \hat{\boldsymbol{\theta}}(n-1)$, $\bar{\kappa}(n) = \bar{\kappa}(n-1)$, and go to Step 7. Otherwise continue.

4. Update $\boldsymbol{T}(n)$ as follows. Replace the bottom row of $\boldsymbol{W}$ by
   $$\sqrt{\lambda^*(n)}\left[\boldsymbol{x}^T(n) \mid y(n)\right].$$
   "Rotate" this new row into $\boldsymbol{W}$ using Givens' rotations, leaving the result
   $$[\boldsymbol{T}(n) \mid \boldsymbol{d}_1(n)]$$
   in the upper m rows of $\boldsymbol{W}$. These rotations involve the scalar computations [13, 14]).
   $$\begin{aligned} W'_{jk} &= W_{jk}\sigma + W_{m+1,k}\tau\delta \\ W'_{m+1,k} &= -W_{jk}\tau\delta + W_{m+1,k}\sigma\delta \end{aligned}$$
   for $k = j, j+1, \ldots, m+1$ and for $j = 1, 2, \ldots, m$; where $\sigma = W_{jj}/\rho$, $\tau = W_{m+1,k}/\rho$,
   $$\rho = \sqrt{W_{jj}^2 + \delta W_{m+1,j}^2},$$
   δ is ±unity[4], and W_{jk} (W'_{jk}) is the j,k element of $\boldsymbol{W}$ pre- (post-) rotation.

5. (Skip if $n \le m$ (see footnote).) Update $\hat{\boldsymbol{\theta}}(n)$ by solving
   $$\boldsymbol{T}(n)\hat{\boldsymbol{\theta}}(n) = \boldsymbol{d}_1(n)$$
   using back-substitution.

6. Update $\kappa(n)$ and $\bar{\kappa}(n)$ according to (7) and (9). (Compute and store only $\bar{\kappa}(n)$ if $n \le m$.)

7. If $n \le N$, increment n and return to Step 1.

[3] Generally $T(n)$ does not become nonsingular until $n = m+1$. The first $\hat{\theta}(n)$ cannot be computed until $n = m+1$ and the first optimal weight, $\lambda^*(n)$, cannot be computed until $n = m+2$ (it is convenient to let $\lambda(n) = 1$ on the initial range), and $\kappa(n)$ is not needed until $n = m+1$. However, $\bar{\kappa}(n)$ must be computed for all n (beginning with $\bar{\kappa}(0) = 0$) so that (7) is efficiently started at $n = m+1$. It is assumed in this algorithm that $\gamma(n)$ is known for all n. For procedures to estimate $\gamma(n)$ and other details, see [3]

[4] δ is set to -1 to rotate an equation *out* of the estimate

A Simple 'Linearized' Learning Algorithm which Outperforms Back-Propagation[1]

J.R. Deller, Jr. and S.D. Hunt

Michigan State University
Department of Electrical Engineering / 260 EB
Control, Systems, and Signal Processing Group: Speech Processing Laboratory
East Lansing, MI 48824-1226 USA
author_surname@ee.msu.edu

Abstract: A class of algorithms is presented for training multilayer perceptrons using purely "linear" techniques. The methods are based upon linearizations of the network using error surface analysis, followed by a contemporary least squares estimation procedure. Specific algorithms are presented to estimate weights node-wise, layer-wise, and for estimating the entire set of network weights simultaneously. In several experimental studies, the node-wise method is superior to back-propagation and an alternative linearization method due to Azimi-Sadjadi *et al.* in terms of number of convergences and convergence rate. The layer and network-wise updating offer further improvement.

1. Introduction

This paper introduces a new class of learning algorithms for feedforward neural networks (FNN) with improved convergence properties. In spite of the nonlinearities present in the dynamics of a FNN, the learning algorithm is purely "linear" in the sense that it is based on a contemporary version (see [1]) of the recursive least squares (RLS) algorithm (e.g. [2]). Accordingly, unlike the popular back-propagation algorithm used to train FNNs [3, 4], the new learning algorithm and its potential variants will benefit from the well-understood theoretical properties of RLS and VLSI architectures for its implementation.

A FNN is an artificial neural network consisting of nodes grouped into layers. In this paper, we consider a two-layer network[2], but the generalization of the method to an arbitrary number of layers is not difficult. Working from the bottom up, we shall frequently refer to layers zero, one, and two as the "input," "hidden," and "output" layers, respectively. Each node above the input layer in the FNN passes the sum of its weighted inputs through a non-linearity to produce its output. The inputs to the input layer are the external inputs to the network, and the outputs of the output layer are the external outputs.

The number of nodes in layer i is denoted N_i, with N_0 indicating the number of input nodes at the bottom of the network. The weight connecting node j in the hidden layer to node k in the output layer is denoted $w_{k,j}$. The weight connecting input node l to node j in the hidden layer is denoted $w'_{j,l}$. We denote by N the number of *training patterns* of the form

$$\{(x_1(n), x_2(n), \ldots, x_{N_0}(n); t_1(n), t_2(n), \ldots, t_{N_2}(n)), \quad n = 1, 2, \ldots, N\}, \tag{1}$$

in which $x_l(n)$ is the input to the l^{th} node in layer zero, and $t_k(n)$ is the *target* output for node k in the output layer (output *desired* in response to the corresponding input). The *computed* outputs of layer two [one] in response to $x_1(n), \ldots, x_{N_0}(n)$ are denoted $y_1(n), \ldots, y_{N_2}$ $[y'_1(n), \ldots, y'_{N_2}]$. Finally, we need to formalize the nonlinearity associated with the nodes. Consider node k in the output layer. For given weights, $w_{k,j}$, $j \in [1, N_1]$, the output in response to the n^{th} input is

$$y_k(n) = S\left(\sum_{j=1}^{N_1} w_{k,j} y'_j(n)\right) \tag{2}$$

in which $S(\cdot)$ is a differentiable nonlinear mapping. For future purposes, we define $\dot{S}(\cdot)$ to be the derivative of $S(\cdot)$. For convenience, we also define $u_k(n) \overset{\text{def}}{=} \sum_{j=1}^{N_1} w_{k,j} y'_j(n)$. Clearly, $u_k(n)$ is the input to node k in the output layer in response to pattern n. $u'_l(n)$ is similarly defined as the input to node l in the hidden layer.

[1] **Acknowledgements**: This work was supported by the Office of Naval Research under Contract No. N00014-91-J-1329, and by the National Science Foundation under Grant No. MIP-9016734. Mr. Hunt was also supported by a fellowship from the University of Puerto Rico.

[2] Some authors might choose to call this a *three* layer network. We shall designate the bottom layer of "nodes" as "layer zero" and not count it in the total number of layers. Layer zero is a set of linear nodes which simply pass the inputs unaltered.

Many training (weight estimation) algorithms exist for this type of network (e.g. [3] - Azimi-Sadjadi 89. The most popular, the back-propagation algorithm [3], [4], performs satisfactorily in some cases if given enough time to converge. However, the literature abounds with example applications in which back-propagation convergence is too slow for practical usage (e.g. see [8]). One attempt to develop faster training methods is represented by the class of algorithms in which the network mapping is "linearized" in some sense in order to take advantage of linear estimation algorithms. It is with this class of algorithms that this paper is concerned.

2. Linearization Algorithm

The fundamental training problem for the two layer FNN is stated as follows: Given a set of $\mathcal{N}$ training patterns as in (1), find the network weights which minimize the sum of squared errors, $E = \sum_{n=1}^{N} \sum_{k=1}^{N_2} \lambda(n)(t_k(n) - y_k(n))^2$,where the weights $\lambda(\cdot)$ are included for generality. For a given set of training pairs, E is a function of the weights of the network. A graph of E over the weight space is frequently called an *error surface*. Ideally, a training algorithm would find the weights corresponding to the global minimum of the error surface. Training algorithms usually operate by sequentially presenting the training patterns and moving the weights toward a minimum of the error surface. The procedure is repeated several times using different initial weights in order to locate the best minimum. Ideally, *all* weights will be altered with each presentation of the set of training patterns so that the weights may move in the direction of steepest descent. In this case the algorithm represents a true ***gradient descent*** approach. In practice, however, no reasonable algorithm exists which can simultaneous change each weight in the network. In fact, the popular back-propagation algorithm works on only one weight at a time. One of the principal benefits of the method to be presented here is that many weights can be simultaneously updated.

The linearization technique adopted in this work can be explained in terms of erroı surface analysis. In effect, for a present set of weights and a given training pattern, we construct a "linearized" network with an error surface, say $\tilde{E}$, which is "similar" in some sense to E in a neighborhood of the present weights. There are two similarity criteria: first, that the magnitude of E and $\tilde{E}$ be the same at the present weights; and second, that the derivatives of E and $\tilde{E}$ with respect to the weights *to be updated* be the same at the present weights (since the other weights are not altered, it is not necessary that the derivatives with respect to those weights match).

Let us digress momentarily from the simple two layer network and use more general description. Suppose that the weights connected to one or more nodes in layer L are to be updated simultaneously[3]. This may include as few as one, and as many as all, nodes in layer L. Denote the set of such selected nodes by $\mathcal{N}$. Denote by $\mathcal{M}$ the set of all nodes above layer L to which any node in $\mathcal{N}$ is connected, directly or indirectly. Let all weights not connected to nodes in $\mathcal{N}$ and $\mathcal{M}$ be fixed at present values[4]. Then it is shown in [9] that a "linearized" network whose error surface $\tilde{E}$ is similar to E in the senses above is constructed by replacing the nonlinearity $S(\cdot)$ for each node in $\mathcal{N}$ and $\mathcal{M}$ by a linear approximation, say $\tilde{S}(\cdot)$, consisting of the first two terms of a Taylor series around the "present" value of the node's input. For example, suppose the k^{th} output node is to be linearized with respect to the n^{th} training pattern. Let $\tilde{w}_{k,j}$ denote the present value of weight $w_{k,j}$. Then,

$$\begin{aligned} S(u) &\approx S(u) = \dot{S}\left(\sum_{j=1}^{N_1} \tilde{w}_{k,j} y_j'(n)\right) [u - \sum_{j=1}^{N_1} \tilde{w}_{k,j} y_j'(n)] + S\left(\sum_{j=1}^{N_1} \tilde{w}_{k,j} y_j'(n)\right) \qquad (3) \\ &= \dot{S}\left(\sum_{j=1}^{N_1} \tilde{w}_{k,j} y_j'(n)\right) u + \left[S\left(\sum_{j=1}^{N_1} \tilde{w}_{k,j} y_j'(n)\right) - \dot{S}\left(\sum_{j=1}^{N_1} \tilde{w}_{k,j} y_j'(n)\right) \sum_{j=1}^{N_1} \tilde{w}_{k,j} y_j'(n)\right] \overset{\text{def}}{=} K_k(n)u + b_k(n). \end{aligned}$$

In fact, since $S(u) = \tilde{S}(u)$ if u is the input corresponding to the present weights, any node not in $\mathcal{N}$ or $\mathcal{M}$ may also be linearized with no effect on the solution. Therefore, we may assume without loss of generality that the entire network is linearized, even if only a portion of the weights is to be updated.

It will become clear below that once the network is linearized by replacing the operation $S(\cdot)$ by $\tilde{S}(\cdot)$ in all appropriate nodes, in principle any least square error algorithm can be used to update the weights. Algorithms based on similar ideas for updating weights one node at a time are given by Azimi-Sadjadi *et al.* [5] (henceforth, *A-S algorithm*) and by Hunt and Deller [9]. The former is based on the conventional RLS algorithm [2] with a

[3] If any weight connected a node is to be updated, then every weight connected to that node must be updated. This "constraint" is ordinarily beneficial, since it implies the ability to simultaneously update more than one weight.

[4] In certain cases it is possible to update weights in different layers simultaneously. We discuss one case at the end of this section

forgetting factor, while the latter employs a contemporary QR decomposition algorithm [1, 10] for significant performance improvement. The view of the method taken above allows us to to further exploit the linearization by complete *layer-wise* updating of weights for even further improvement. Let us pursue this layer-wise approach.

Suppose we wish to update all weights in the output layer simultaneously. We must linearize all output nodes (and may arbitrarily linearize any other nodes). For node k in the output layer, the output in response to input n is computed as in (2). Let $\bar{y}_k(n)$ represent the output of node k after $S(u_k(n))$ as been replaced by $\hat{S}(u_k(n)) = K_k u_k(n) + b_k$. Accordingly,

$$\bar{y}_k(n) = K_k(n)[\sum_{j=1}^{N_1} w_{k,j} y'_j(n)] + b_k(n) \quad \text{or} \quad \bar{z}_k(n) = K_k(n)[\sum_{j=1}^{N_1} w_{k,j} y'_j(n)] \tag{4}$$

with $\bar{z}_k(n) \overset{\text{def}}{=} \bar{y}_k(n) - b_k(n)$. We speak of the rightmost form in (4) as descriptive of a *linearized node* since the output is a purely linear combination of the inputs to the node. The network with all appropriate nodes linearized will be called the *linearized network*. Since $\bar{y}_k(n) = y_k(n)$ at the present weights, the error at the k^{th} node will be the same for the linearized and original network if the target value for $\bar{z}_k(n)$, say $\bar{t}_k(n)$, is taken to be

$$\bar{t}_k(n) \overset{\text{def}}{=} t_k(n) - b_k(n) \tag{5}$$

and the linearized inputs to node k at pattern n are

$$\bar{x}_{k,j}(n) \overset{\text{def}}{=} K_k(n) y'_j(n), \quad j = 1, 2, \ldots, N_1. \tag{6}$$

Note that the linearized inputs are dependent upon k, so that we have effectively increased the number of training pairs by a factor of N_2.

The problem has effectively been reduced to one of estimating weights for a single-layer linear network. In order to simultaneously update the all weights in the output layer, the system of $N \times N_2$ equations

$$\bar{t}_k(n) = \sum_{j=1}^{N_1} \bar{x}_{k,j} w_{k,j}, \quad k = 1, 2, \ldots, N_2 \quad n = 1, 2, \ldots, N \tag{7}$$

must be solved for the least square estimate of the $N_1 \times N_2$ weights $w_{k,j}$, $k \in [1, N_2]$ $j \in [1, N_1]$. However, since all weights in the hidden layer are fixed, the outputs $y'_j(n)$ are independent of k. This means that the equations indexed by different values of k are independent of one another, and the sets of weights connected to different outputs may be updated independently. In the output layer, therefore, there is no theoretical difference between layer-wise and node-wise updating. This is not true at lower layers, however, as we now show for the hidden layer of the present network.

To update all weights in the hidden layer simultaneously, the weights in the output layer are fixed and *all* nodes in the network must be linearized. The outputs of the hidden layer with $S(\cdot)$ replaced by $\hat{S}(\cdot)$ are given by

$$\bar{y}'_j(n) = K'_j(n)[\sum_{l=1}^{N_0} w'_{j,l} x_l(n)] + b'_j(n), \quad j = 1, 2, \ldots, N_1. \tag{8}$$

Substituting (8) in the leftmost expression in (4) results in

$$\bar{y}_k(n) - [\sum_{j=1}^{N_1} K_k(n) w_{k,j} b'_j(n) + b_k(n)] = \sum_{j=1}^{N_1} \sum_{i=1}^{N_0} [K_k(n) w_{k,j} K'_j(n) x_i(n)] w'_{j,l}. \tag{9}$$

As above, we can now view the problem as one of training a single-layer linear mapping with target outputs

$$\bar{t}'_k(n) = t_k(n) - [\sum_{j=1}^{N_1} K_k(n) w_{k,j} b'_j(n) + b_k(n)] \tag{10}$$

and inputs

$$\bar{x}'_{k,j,l}(n) = K_k(n) w_{k,j} K'_j(n) x_l(n). \tag{11}$$

The weight estimates for $w'_{j,l}$, $j \in [1, N_1]$ $l \in [1, N_0]$ comprise the least square error solution to the system of equations

$$\bar{t}'_k(n) = \sum_{j=1}^{N_1} \sum_{l=1}^{N_0} \bar{x}'_{k\,j,l}(n) w'_{j,l} \quad k = 1, 2, \ldots, N_2 \quad n = 1, 2, \ldots, N. \tag{12}$$

Unlike the output layer, we see that the problem cannot be decomposed into separate solutions for sets of weights connected to individual nodes in the hidden layer. This is a reflection of the fact that all weights in the hidden layer are coupled through their "mixing" in the output layer. This means that the simultaneous solution for all weights in the hidden layer should be beneficial with respect to a node-wise solution. Indeed we will find this to be the case in the experiments. Of course, this same intra-layer dependence of weights would continue if there were further hidden layers to be considered.

Note that, for a fixed k, the inputs to the linearized network, $\bar{x}'(n)$, $n \in [1, N]$, are most conveniently viewed as two-dimensional (indexed by couples (j, l)). There are N such "grid" inputs for each k, paired with the N values of $\bar{t}'_k(n)$. If there were further hidden layers in the network, we would find that the effective inputs would continue to increase in dimension. Further, it is noted that the role of k in (12) is somewhat superfluous. In principle, the index is used to keep track of which of N_2 outputs in the linearized network is being considered. However, the training pairs $(\bar{t}'_k(n); \bar{x}'_{k,1,1}(n), \ldots \bar{x}'_{k,N_1,N_0}(n))$, $k \in [1, N_2]$ $n \in [1, N]$, can be reindexed by mapping pairs $(k, n) \to i$ so that the training pairs may be written $(\bar{t}'(i); \bar{x}'_{1,1}(i), \ldots, \bar{x}'_{N_1,N_0}(i))$, $i \in [1, N \times N_2]$. Of course, an identical system of equations to (12) results, but the linearized network may be viewed as a *single output* linear layer with $N \times N_2$ training pairs.

Updating of some subset of the weights in the hidden layer (in particular, "node-wise" as in the A-S algorithm) is tantamount to solving the subsystem of (12) corresponding to the desired weights, introducing the updated values into the system, solving for the next desired subset, etc. Clearly, this will result in a different solution than the simultaneous solution. In terms of the error surfaces, this process consists of continually updating the error surface as "partial" information becomes available, then moving in the direction of the gradient with respect to a new subset of weights in the updated surfaces. Intuitively, movement "at once" with respect to the "complete" gradient would seem to be a preferable procedure. Indeed, the later operation corresponds to the simultaneous updating.

The linearization allows us to approximate the error surface of the nonlinear system for only a small neighborhood around the present weights. Because of the criteria used to construct $\bar{E}$, the weights will be changed in the *direction* of the true gradient in the nonlinear space, but will move to the minimum of $\bar{E}$ which may be quite far from the neighborhood over which $E \approx \bar{E}$. Accordingly, the weights must be allowed to change only a small amount using the training patterns of the linearized system. If the linearized procedure results in a large change of weights, measures must be taken to decrease the alteration. The updating procedure is repeated until changing the weights does not result in a decrease in error. The algorithm proceeds as follows: linearize the system around the present weights, change the weights by a small amount to decrease error, then repeat the procedure. This is done until changing the weights does not decrease the error or a maximum on the number of linearizations is reached.

For the same reason that simultaneous *layer*-wise estimation of weights is beneficial, we should expect even more benefit from complete ***network*** updating if such were possible. It follows from the developments above that entire network updating is possible for at least one case. *If there is a single node in the output layer of the network*, let $k = 1$ and define

$$w^{\dagger}_{j,l} = w_{k,j} w'_{j,l} = w_{1,j} w'_{j,l} \tag{13}$$

From (9) it follows that

$$(\bar{y}_1(n) - b_1(n)) = \sum_{j=1}^{N_1} \sum_{l=1}^{N_0} [K_1(n) K'_j(n) x_l(n)] w^{\dagger}_{j,l} + \sum_{j=1}^{N_1} [K_1(n) b'_j(n)] w_{1,j}. \tag{14}$$

This can be interpreted as an attempt to train a single linear layer with one output and $(N_0 \times N_1) + N_1$ inputs. In this case, there will be only N linearized training patterns. The system can be solved for $w_{1,j}$ and $w^{\dagger}_{j,l}$, $j \in [1, N_1]$ $l \in [1, N_0]$ and (13) can be used to solve for $w'_{j,l}$, $j \in [1, N_1]$ $l \in [1, N_0]$.

3. Experimental Results

The results given in this section compare five training strategies for a FNN. These are: 1. Conventional back-propagation (no linearization in the sense described here, weight-wise updating); 2. A-S algorithm (no

Impementation →	Back-Prop	A-S	Node Updating	Layer Updating	Network Updating
No. of Convergences →	11	8	78	96	99

Table 1: Number of convergences per 100 sets of initial weights in experiments with the XOR network.

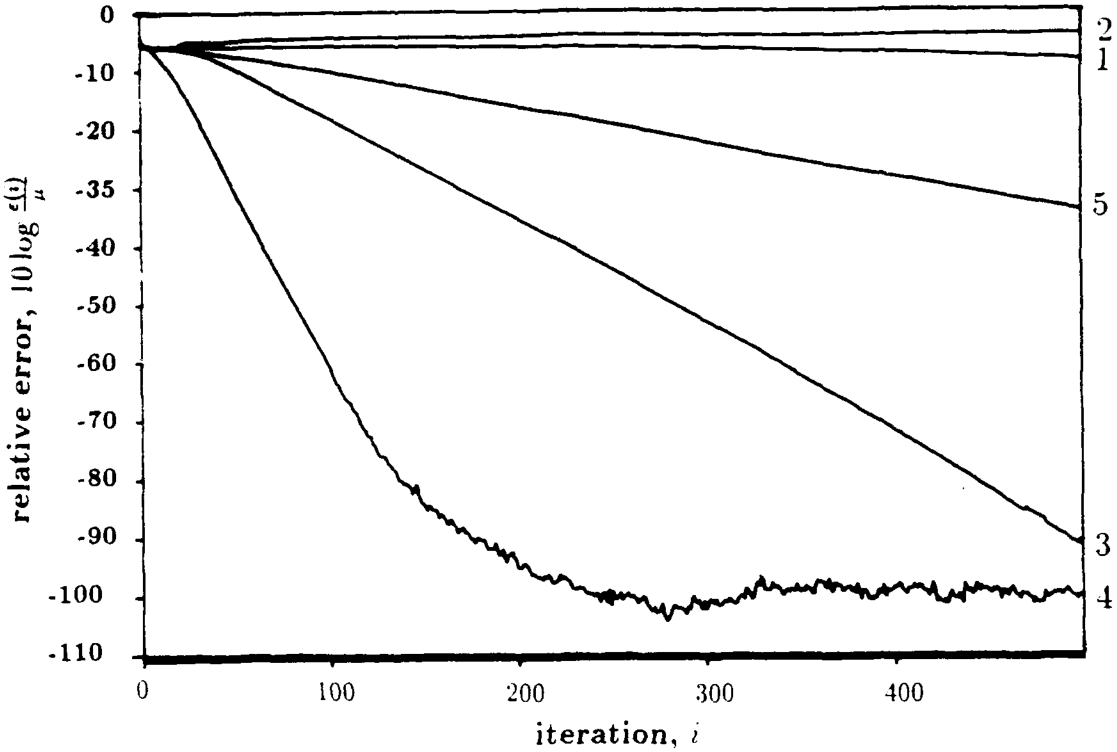


Figure 1: Average error in dB for the XOR implementations vs. iteration number. 1.Back-propagation: 2.A-S algorithm; 3. Node-wise updating; 4.Layer-wise updating; 5.Network-wise updating.

linearization, then conventional RLS with a forgetting factor for node-wise updating); 3. Linearization method described above with node-wise updating based on QR decomposition; 4. Same as 3 with *layer*-wise updating; 5. Same as 3 with *complete network* updating. The two-bit parity checker (XOR) network used in the simulations has two inputs, two hidden layer nodes and one output node. An additional node is added at each layer whose output value was always unity, to serve as a bias for each node in the layer above. The initial weights were chosen as follows. Each weight in the network was selected randomly from a uniform distribution over the interval $[-1, 1]$. This procedure was repeated 100 times to select 100 sets of initial weights. The same 100 sets of weights were used for all five implementations. For the back-propagation algorithm, a factor of 0.04 was used in the weight updating equation. The A-S algorithm was implemented using no weight change constraints. The forgetting factor for A-S and for the QR decomposition implementation was 0.98. The QR decomposition implementation used a weight constraint of 0.2, meaning that the weight vector associated with each node was allowed to change at most by 0.2 in Euclidean norm during each iteration. The layer-wise updating algorithm has a forgetting factor of 0.3 and a weight constraint of 1.0. The network-wise updating algorithm had the same forgetting factor and weight constraint as the layer case.

Simulations were run to compare the number of times each implementation found weights that solve the XOR problem for the 100 initial weight sets. The results are shown in Table 1.

Simulations were also run to compare the output error of each algorithm. In the resulting figures, the error in dB means the following: Let $\varepsilon(i)$ be the sum of the squared errors incurred in iteration i through the training patterns, averaged over the 100 initial weight sets. Then, plotted in the figures is $10\log(\varepsilon(i)/\mu)$ (dB), where μ is the maximum possible error in any iteration. Figure 1 shows the errors of the four XOR implementations.

4. Conclusions

A new implementation for node-wise weight updating algorithm for feedforward neural networks and new algorithms that update weights layer-wise and network-wise have been presented in this paper. The QR decomposition implementation has been shown experimentally to be superior to standard recursive equations for the node-wise updating algorithm. The layer-wise and network-wise weight updating algorithms were developed to improve the convergence rate and the speed of convergence. Both objectives were accomplished, with the layer-wise weight updating algorithm showing a significant advantage over both the single node weight updating algorithm used as a reference, and the widely used back-propagation algorithm.

References

[1] J.R. Deller, Jr. and D. Hsu, "An alternative adaptive sequential regression algorithm and its application to the recognition of cerebral palsy speech," *IEEE Trans. Circuits and Systems*, vol. CAS-34, pp.782-786, July 1987.

[2] D. Graupe, *Time Series Analysis, Identification and Adaptive Filtering*, Malabar, Florida: Krieger Publishing Company, 1989.

[3] P. Werbos, *Beyond Regression: New Tool for Prediction and Analyses in the Behavioral Sciences* (Ph.D. dissertation), Harvard University, Cambridge, Massachusetts, 1974.

[4] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in D. Rumelhart and J. McCleland (editors), *Parallel Distributed Processing*, vol. 1 Cambridge, Massachusetts: MIT Press, 1986.

[5] M. Azimi-Sadjadi, S. Citrin, and S. Sheedvash, "Supervised learning process of multi-layer perceptron neural networks using fast least squares," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp.1381-1384, 1990.

[6] S. Kollias and D. Anastassiou, "An adaptive least squares algorithm for the efficient training of artificial neural networks," *IEEE Trans. Circuits and Systems*, vol. CAS-36, pp.1092-1101, August 1989.

[7] M. Azimi-Sadjadi and S. Citrin, "Fast leaning process of multi- layer neural nets using recursive least squares technique," *Proc. IEEE Int. Conf. on Neural Networks*, Washington D.C., June 1989.

[8] R. Lippmann, "Review of neural networks for speech recognition," *Neural Computation*, vol. 1, pp. 1-38, 1989.

[9] S.D. Hunt and J.R. Deller, Jr., " 'Linearized' alternatives to back-propagation based on recursive QR decomposition," *IEEE Trans. Neural Networks* (in review).

[10] G. Golub and C. VanLoan, *Matrix Computations*, Baltimore, Maryland: Johns Hopkins University Press, 1983.

Contents