(2)

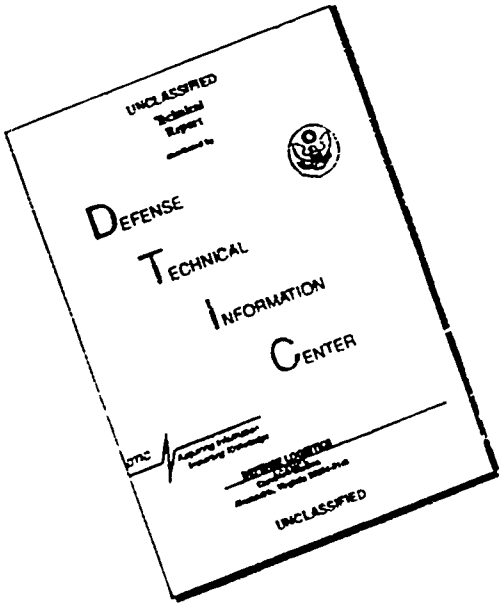
FINAL PROJECT REPORT
Contract N00014-89-J-1788

Principal Investigators: Professor Yih-Fang Huang
Professor Ruey-wen Liu
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Department of
ELECTRICAL ENGINEERING

UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

FINAL PROJECT REPORT
Contract N00014-89-J-1788

Principal Investigators: Professor Yih-Fang Huang
Professor Ruey-wen Liu
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Statement A per telecon
Dr. Rabinder Madan ONR/Code 1114
Arlington, VA 22217-5000

NWW 4/27/92

Project Summary

Project Title: Real-Time Aggressive Image Data Compression

Principal Investigators: Dr. Yih-Fang Huang and Dr. Ruey-wen Liu

Institution: University of Notre Dame

Address: Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
e-mail: huang@saturn.ece.nd.edu

Summary

The objective of the proposed research is to develop reliable algorithms that can achieve aggressive image data compression (with a compression ratio of 60 times or more) for real-time implementation. Typical applications of such algorithms include terrestrial HDTV broadcasting, space communications, and handling and disposing of toxic materials and nuclear wastes with remotely controlled robots. The state-of-the-art techniques are hampered by serious technical barriers of codebook design complexity.

The proposed approach is built on a vector quantization (VQ) algorithm recently developed by the PI. The codebook design complexity of this VQ algorithm is only linearly proportional to the codebook size (significantly less than conventional algorithms) and the encoding complexity is independent of codebook size. Highlighting the proposed approach is a piecewise-linear transform preceding VQ based on the concept of entropy partitioning.

The novelty of the proposed algorithm is due to the following: (i) introduction of a piecewise-linear transform to VQ so as to retain more input information; (ii) exploiting both inter-block and intra-block redundancy; (iii) use of parallel distributed network for real-time codebook design.

The proposed research is significant as (i) it addresses the imminent demands of solving the aforementioned real-world problems; (ii) its accomplishment will alleviate the serious complexity barrier of conventional VQ algorithms; (iii) it pushes forward the technical frontiers of data compression.

Table of Contents

I. EXECUTIVE SUMMARY

This final report summarizes accomplishments and overall progress made during the period of April 1, 1989 to March 31, 1990 for the research project sponsored by the Office of Naval Research under Contract N00014-89-J-1788. It enlists publications and Theses and Dissertations that have been sponsored, in part, by this research project. A selected subset of the related publications is also included.

The objective of this research project is to perform fundamental studies on the theory of selective update in signal and image processing. The approach is based on selective use of input data in retrieving information of the underlying signals. The selection will be based on the information content of the incoming data. Of particular interests here are parameter estimation in adaptive systems. The significance of this research project is that it is a timely response to the demand of higher level of machine automation and man-machine interaction.

Over the last few decades, much endeavor has been made on improving the effectiveness of data processing, particularly on integrated circuits technology. The advent of very large scale integrated (VLSI) circuits technology has made available fast and high density circuitry devices at lower costs. Processing of large volumes of data in real time has thus become more feasible and cost-effective in practice. As such, modern signal and image processing calls for algorithms that are compatible with such technological advances. In particular, algorithms which can be implemented with higher degrees of *modularity*, *concurrency*, and higher levels of *machine intelligence*, thereby providing higher data-throughput rates, are more appealing in practice.

Most, if not all, of the efforts have been focused on the improvements of general computational capabilities or the architectures of maneuvering arithmetic operations. One critical issue which has often been overlooked is the extent of intelligence incorporated in the algorithms implemented. In particular, *selective* use of the input data to improve the efficiency of information retrieval is as critical as improving the speed of simple arithmetic operations. An essential reason for the *selective use* of input data is that it eliminates redundant processing, thus could improve significantly the potential of modular concurrent processing. It also incorporates a decision-making

procedure in the selection of data, thus enhances the level and capability of machine automation.

This research project concentrates on the context of adaptive signal processing in studying selective use of information. The ground work upon which this research project rests is a set of recursive parameter estimation algorithms, i.e., the so-called OBE algorithms, which feature a *discerning update* strategy. This discerning update is in sharp contrast to the *continual update* used by most existing algorithms.

The OBE algorithms belong to a class of parameter estimation/identification algorithms termed *Set-Membership* (SM) algorithms. The SM algorithms use certain set-theoretic type of *a priori* knowledge about the underlying model to constrain the solutions to a certain set. In particular, the disturbance and the input signals are assumed to be bounded in some sense. The OBE algorithms are, perhaps, the most viable SM estimation technique in terms of analytical tractability and practical appealingness.

The emerging field of SM-based signal processing has received considerable attention and is becoming increasingly popular in the research community around the world. Many special sessions at professional conferences have been organized and special issues in professional journals have been published. It is clear that researchers around the world are excited about the tremendous potential of SM-based algorithms for applications to problems of practical importance. To name a few applications, time series analysis, spectrum estimation, speech and image enhancement/processing, biology and chemistry, and pharmacokinetics are among the more notable ones.

One of the striking features of recursive SM-based algorithms, thus OBE algorithms, is a discerning update strategy for the parameter estimates. An important outcome of such discerning updates is that the resulting algorithm can be implemented with two modules: an *information processor* followed by an *updating processor*. The former decides whether an update is needed, and the decision is based on the evaluation of the "*information quality*" *of the input data*, the *prediction error*, and the *noise bound*. It is essential that the information evaluation involves very little computational effort, which is the case here. The latter then updates the parameter estimates when the information processor decides that such is needed.

Simulations on the OBE algorithms have shown, in general, that only less than 20% of the input data are used to update the parameter estimates. This is true for most practical systems that can be modeled by autoregressive processes with exogeneous inputs (ARX) or autoregressive moving average (ARMA) processes whose order is less than ten.

Conceptually, thanks to the modularity and to the fact that only less than 20% of input data are used to update the parameter estimates, an adaptive signal processing network may be constructed. The network will consist of a number of such modular recursive estimators, each of which is comprised of two modules, namely, the information evaluator and the updating processor. As such, idle time of both the information evaluator and the updating processor can be reduced, thus the data throughput rate will be increased. In addition, the reliability of signal processing can be improved greatly. In essence, this type of adaptive networks will be able to process *multi-channel adaptation and filtering*, improving *reliability* and *data throughput rates*. One of the important applications for this is *adaptive array processing* in sonar systems.

In this project, several fundamental issues associated with this kind of estimation algorithms are investigated. To begin with, investigations are conducted to extend one of the OBE algorithms to the estimation of parameters of autoregressive-moving-average (ARMA) processes. The resulting algorithm is referred to as the EOBE algorithm. The ARMA process has been used to model signals encountered in underwater array signal processing.

Among others, the issue of convergence for ARMA parameter estimation is of critical importance to practical implementation. We showed that if the input noise is bounded in magnitude and the moving average parameters satisfy a certain magnitude bound, then the *a posteriori* prediction errors are uniformly bounded. With an additional persistence of excitation condition, the parameter estimates are shown to converge to a neighborhood of the true parameters, and the *a priori* prediction errors are asymptotically bounded. In contrast, the conventional algorithm of *extended least-squares* requires the strictly positive real (SPR) condition to assure convergence.

It is worth mentioning that an important virtue of this EOBE algorithm is that,

under rather mild conditions, the bounding ellipsoids always contain the true parameter, providing a 100% confidence region for the true parameter. This is a feature not shared by other conventional algorithms which only guarantee that asymptotically.

Implementation on finite word-length processors is almost a mandate for all signal processing algorithms. We investigated the OBE algorithms' performance in finite word-length environment via simulations. In particular, the effects of roundoff error accumulation and numerical stability were studied with fixed point simulations. Analysis of error propagation in an OBE algorithm is also performed which shows that the errors in the estimates due to an initial perturbation are bounded. Based on these results, we showed that the OBE and the EOBE appear to be superior to the RLS and the ELS, respectively.

One of the possible reasons for such encouraging results is the discerning update strategy which updates parameter estimates less frequently, thereby accumulates less roundoff errors. Another reason is imbedded in the update equations which may require more detailed analysis. Nevertheless, these results further verify our conjecture that eliminating redundant use of information, contained in the received data, would reduce the effects of roundoff errors.

We further investigated one of the OBE algorithms in terms of tracking properties. Conditions which ensure the existence of these 100% confidence regions in the face of small model parameter variations are derived. For larger parameter variations, it is shown that the existence of the 100% confidence region can be guaranteed asymptotically. A modification of the OBE algorithm was also proposed to enable tracking of larger variations. Our simulation results have shown that the modified algorithm has tracking performance comparable, and in some cases, superior, to the exponentially weighted recursive least-squares algorithm.

In summary, our studies in this one-year project established the practical viability of estimation algorithms that selectively use the input data.

THESES AND DISSERTATIONS DIRECTED

The following is a list of masters theses and doctoral dissertations that had been sponsored, in part, by the grant.

1. Ashok K. Rao (Ph.D.) Dissertation Title: Membership-Set Parameter Estimation via Optimal Bounding Ellipsoids. (Graduation Date: January 1990)[1].

2. Vincent R. Marcopoli (M.S.)
   Thesis Title: Equation Error and Output Error Methods for Adaptive System Identification. (Graduation Date: January, 1990)[2].

RELATED PUBLICATIONS

1. A. K. Rao and Y. F. Huang,

   "Analysis of Fixed-Point Roundoff Errors in the OBE Algorithm", *Proc. IEEE 1989 International Conf. on Acoust., Speech, and Signal Processing*, pp. 853-856, Glasgow, Scotland, UK, May, 1989.

2. A. K. Rao, Y. F. Huang, and S. Dasgupta

   "On the Bounds of the Estimation Errors of the EOBE Algorithm," *Proc. 32nd Midwest Symposium on Circuits and Systems*, Urbana, IL, August, 1989.

3. Q. Huang, D. Graupe, Y.F. Huang, and R. Liu

   "Identification of Firing Patterns of Neuronal Signals", *Proc. 28-th IEEE Conf. Decision and Control*, pp. 266-271, Tempa, FL, Dec. 1989.

4. A. K. Rao and Y. F. Huang

   "Tracking Characteristics of An OBE Parameter Estimation Algorithm", *Proc. 24th Conf. Inform. Sci. and Syst.*, Princeton Univ., Princeton, N.J., pp. 17-22, March, 1990.

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 Processing*, Vol. ASSP-38, No. 3, pp. 447-457, March 1990.

[1]This Dissertation had been included in the mid-year technical report submitted previously
[2]This Thesis is being included in this final report

6. A. K. Rao and Y. F. Huang (Invited)

"Recent Developments in Optimal Bounding Ellipsoidal Parameter Estimation," Special Issue on Bounded Error Estimation, *Mathematics and Computers in Simulation*, Vol. 32, Nos. 5 & 6, pp. 515-526, Dec., 1990.

7. A. K. Rao and Y. F. Huang

"Analysis of Finite Precision Effects on a Recursive Set Membership Parameter Estimation Algorithm", *IEEE Trans. Signal Processing*,Vol. 40, No. 12, Dec. 1992, (to appear).

8. A.K. Rao and Y.F. Huang

"Tracking Characteristics of An OBE Parameter Estimation Algorithm", *IEEE Trans. Signal Processing*, (to appear).

Publication [1]

ARMA Parameter Estimation Using a Novel Recursive Estimation Algorithm with Selective Updating

ASHOK K. RAO, YIH-FANG HUANG, MEMBER, IEEE, AND SOURA DASGUPTA

Abstract—This paper investigates an extension of a recursive estimation algorithm (the so-called OBE algorithm) [9]–[11], which features a discerning update strategy. In particular, an extension of the algorithm to ARMA parameter estimation is presented here along with convergence analysis. The extension is similar to the extended least-squares algorithm. However, the convergence analysis is complicated due to the discerning update strategy which incorporates an information-dependent updating factor. The virtues of such an update strategy are: 1) more efficient use of the input data in terms of information processing, and 2) a modular adaptive filter structure which would facilitate the development of a parallel-pipelined signal processing architecture. It is shown in this paper that if the input noise is bounded and the moving average parameters satisfy a certain magnitude bound, then the *a posteriori* prediction errors are uniformly bounded. With an additional persistence of excitation condition, the parameter estimates are shown to converge to a neighborhood of the true parameters, and the *a priori* prediction errors are shown to be asymptotically bounded. Simulation results show that the parameter estimation error for the EOBE algorithm is comparable to that for the ELS algorithm.

I. Introduction

In many adaptive signal processing applications, such as speech processing, seismic data processing, and channel equalization, a signal $y(t)$ is often considered as the output of an IIR filter driven by unknown white noise $w(t)$ [1]. The signal $y(t)$ can therefore be modeled as an autoregressive moving average (ARMA) process of the form

$$y(t) = a_1 y(t-1) + \cdots + a_n y(t-n) + w(t) + c_1 w(t-1) + \cdots + c_r w(t-r). \quad (1.1)$$

Fitting this ARMA model to the measured data $y(t)$, $t = 1, 2, \cdots$, requires the estimation of the parameters $a_1, \cdots, a_n, c_1, \cdots, c_r$. Many methods for the estimation of ARMA parameters have been proposed in the literature, particularly from the spectral estimation viewpoint. Among the more recent are Cadzow's overdetermined rational equation method [2], the spectral matching technique of Friedlander and Porat [3], and the extended Yule–Walker method of Kaveh [4]. A common feature of these methods is the use of the sample autocorrelation sequence of the output process $y(t)$. In the context of system identification, the extended least-squares (ELS), the recursive maximum likelihood (RML), and multistage least-squares algorithms have been used to recursively estimate ARMA parameters [5], [6], [12]. The ELS algorithm uses the *a posteriori* prediction error $\epsilon(t)$, as an estimate of $w(t)$. The regressor vector is formed from $y(t-1), \cdots, y(t-n)$ and $\epsilon(t-1), \cdots, \epsilon(t-r)$. The standard recursive least-squares (RLS) algorithm is then employed to update the estimates. The algorithm is conceptually simple but restrictive in the sense that convergence of the algorithm can be assured only if the underlying transfer function $H(q^{-1}) = 1/C(q^{-1}) - 1/2$ is strictly positive real (SPR), with q^{-1} being the delay operator and

$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_r q^{-r}. \quad (1.2)$$

The RML algorithm, which uses a filtered version of the regressor vector used in the ELS algorithm, does not require $H(q^{-1})$ to be SPR. However, the estimates have to be monitored and projected into a stability region to ensure convergence [5].

In addition to the aforementioned least-squares based methods, there exists a different class of estimation algorithms that estimate membership sets of parameters which are consistent with the measurements and noise constraints [7]–[11]. These algorithms are particularly useful when the noise distribution is unknown but constraints in the form of bounds on the instantaneous values of the noise are available. To the best of our knowledge, none of the algorithms has been applied to the problem of ARMA parameter estimation. Among these algorithms based on membership sets, a group of seminal recursive algorithms are the so-called optimal bounding ellipsoid (OBE) algorithms [9]–[11]. The OBE algorithms have been developed using a set-theoretic formulation and are applicable to autoregressive with exogenous input (ARX) models with bounded noise. One of the main features of these temporally recursive algorithms is a discerning update strategy. This feature, obtained by the introduction

Manuscript received April 30, 1988, revised May 4, 1989. This work was supported in part by the National Science Foundation under Grant MIP-87-11174; in part by the Office of Naval Research under Contract N00014-87-K-0284, and in part by the National Science Foundation under Grant ECS-8618240.

A. K. Rao is with COMSAT Laboratories, Clarksburg, MD 20871.

Y.-F. Huang is with the Department of Electrical and Computer Engineering, University of Notre Dame, Notre Dame, IN 46556.

S. Dasgupta is with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242.

IEEE Log Number 8933425.

0096-3518/90/0300-0447$01.00 © 1990 IEEE

of an information dependent updating forgetting factor, yields a modular structure thereby increasing the potential for concurrent and pipelined processing of signals. The presence of such a forgetting factor also gives the algorithms the ability to track slowly time varying parameters. One of the algorithms [11] has been shown to possess the advantageous feature of automatic asymptotic cessation of updates if the model is time invariant. If a loose upper bound on the noise magnitude is known, and if the input is persistently exciting and sufficiently uncorrelated with the noise, then it has been shown in [11] that the parameter estimates converge asymptotically to a neighborhood of the true parameter vector.

In this paper, we extend one of the OBE algorithms [11] to the ARMA case. For the ARMA parameter estimation problem, the OBE algorithm cannot be applied in its present form. However, by assuming that the input white noise is bounded in magnitude, the OBE algorithm can be extended in a manner similar to the ELS algorithm. Convergence analysis of the resulting algorithm is performed by imposing a bound on the sum of the magnitudes of the MA coefficients. This ensures that the true parameter vector is contained in all the optimal bounding ellipsoids. A uniform bound on the *a posteriori* prediction error can then be derived. In contrast, even though the *a posteriori* prediction errors are generated in a stable fashion in the ELS algorithm [5], it is difficult to obtain an expression for even the asymptotic bound, if such a bound exists. By imposing a persistence of excitation condition on the regressor vector, the *a priori* prediction error of the extended OBE algorithm is shown to be bounded and the parameter estimates are shown to converge to a neighborhood of the true parameter vector.

The paper is organized in the following manner. In Section II, a brief review of the OBE algorithm and its properties is presented. In Section III, the algorithm is extended to ARMA parameter estimation. Convergence analysis of the extended algorithm is performed in Section IV. The performance of the algorithm is compared to the ELS algorithm through simulation studies in Section V. Section VI concludes the paper.

II. The OBE Algorithm

Consider the ARX model described by

$$y(t) = a_1 y(t-1) + \cdots + a_n y(t-n) + b_0 u(t) + b_1 u(t-1) + \cdots + b_m u(t-m) + v(t)$$

where $y(t)$ is the output, $u(t)$ is the measurable input, and $v(t)$ represents the uncertainty or noise. The above equation can be recast as

$$y(t) = \theta^{*T}\Phi(t) + v(t) \tag{2.1}$$

where

$$\theta^* = [a_1, a_2, \cdots, a_n, \quad b_0, b_1, \cdots, b_m]^T$$

is the vector of true parameters and

$$\Phi(t) = [y(t-1), y(t-2), \cdots, y(t-n), u(t), u(t-1), \cdots, u(t-m)]^T$$

is the regressor vector. A key assumption here is that the noise is bounded in magnitude, i.e., there exists a $\gamma_0 \geq 0$, such that

$$v^2(t) \leq \gamma_0^2, \quad \text{for all } t, \text{ hence,}$$

$$(y(t) - \theta^{*T}\Phi(t))^2 \leq \gamma_0^2.$$

Let S_t be a subset of the euclidean space R^{n+m+1}, defined by

$$S_t = \left\{\theta: (y(t) - \theta^T\Phi(t))^2 \leq \gamma_0^2, \quad \theta \in R^{n+m+1}\right\}.$$

From a geometric point of view, S_t is a convex polytope in the parameter space and contains the vector of true parameters. The OBE algorithm starts off with a large ellipsoid, E_0, in R^{n+m+1} which contains all possible values of the modeled parameter θ^*. After the first observation $y(1)$ is acquired, an ellipsoid is found which bounds the intersection of E_0 and the convex polytope S_1. This ellipsoid must be optimal in some sense, say minimum volume [9], [10] or by any other criterion [9], [11], to hasten convergence. Denoting the optimal ellipsoid by E_1, one can proceed exactly as before with the future observations and obtain a sequence of optimal bounding ellipsoids $\{E_t\}$. The center of the ellipsoid E_t can be taken as the parameter estimate at the tth instant and is denoted by $\theta(t)$. If at a particular time instant t, the resulting optimal bounding ellipsoid would be of a "smaller size," thereby implying that the data point $y(t)$ conveys some fresh "information" regarding the parameter estimates, then the parameters are updated. Otherwise, E_t is set equal to E_{t-1} and the parameters are not updated. It can also be shown [11] that all the ellipsoids $\{E_t, t = 1, 2, \cdots\}$ contain the true parameter θ^*, provided that E_0 does.

Let the ellipsoid E_{t-1} at the $(t-1)$th instant be formulated by

$$E_{t-1} = \left\{\theta: (\theta - \theta(t-1))^T P^{-1}(t-1) \cdot (\theta - \theta(t-1)) \leq \sigma^2(t-1)\right\} \tag{2.2}$$

for some positive definite matrix $P(t-1)$ and a nonnegative scalar $\sigma^2(t-1)$. Then, given $y(t)$, an ellipsoid which bounds $E_{t-1} \cap S_t$ "tightly" is

$$\left\{\theta: (1-\lambda_t)(\theta - \theta(t-1))^T P^{-1}(t-1) \cdot (\theta - \theta(t-1)) - \lambda_t(y(t) - \theta^T\Phi(t))^2 \leq (1-\lambda_t)\sigma^2(t-1) - \lambda_t\gamma_0^2\right\} \tag{2.3}$$

where the forgetting factor $\lambda(t)$ satisfies $0 \leq \lambda(t) < 1$. The size of the bounding ellipsoid is related to the scalar $\sigma^2(t-1)$ and the eigenvalues of $P(t-1)$. The update equations for $\theta(t)$, $P(t)$, and $\sigma^2(t)$ are derived in [11]. The optimal ellipsoid which bounds the intersection of

E_{t-1} and S_t is defined in terms of an optimal value of λ_t. For the OBE algorithm of [11], the optimum value λ_t^* is determined by minimizing $\sigma^2(t)$ with respect to λ_t at every time instant. The minimization procedure results in λ_t^* being set equal to zero (no update) if

$$\sigma^2(t-1) + \delta^2(t) \le \gamma_0^2. \tag{2.4}$$

If (2.4) is not satisfied, then the optimal value of λ_t is computed. The parameter estimation procedure is depicted in Fig. 1. An outgrowth of this modular recursive estimation procedure is a parallel-pipelined networking structure [13]. The algorithm is such that the computational complexity of the information evaluation (IE) procedure is much less than that of the updating procedure (UPD). Since, in general, a good number of data samples would be rejected by the IE, both the IE and the UPD would involve significant amounts of idle time. A viable scheme then is to configure a parallel-pipelined network comprising of such modular estimators to process signals from multiple channels. Apart from reducing hardware costs, such a scheme would offer increased reliability since the failure of one UPD processor would not cause any of the channels to fail, in contrast to a system with a dedicated UPD processor for each channel.

III. Extension to ARMA Models

The ARMA model described by (1.1) can be rewritten as

$$w(t) = y(t) - \theta^{*T}\Phi'(t) \tag{3.1}$$

where θ^*, the vector of true parameters, and $\Phi'(t)$ are defined by

$$\theta^* = [a_1, a_2, \cdots, a_n, \quad c_1, c_2, \cdots, c_r]^T$$

$$\Phi'(t) = [y(t-1), \cdots, y(t-n), \quad w(t-1), \cdots, w(t-r)]^T.$$

Here again, $w(t)$ is assumed to be bounded in magnitude, i.e., there exists positive γ_0^2 such that

$$w^2(t) \le \gamma_0^2. \tag{3.2}$$

Since the values of the noise sequence $\{w(t)\}$ are not available, the regressor vector $\Phi'(t)$ is not known exactly. If, however, at time t, an estimate of θ^*,

$$\theta(t) = [a_1(t), \cdots, a_n(t) \quad c_1(t), \cdots, c_r(t)]^T \tag{3.3}$$

is available, $w(t)$ could be estimated by the *a posteriori* prediction error

$$\epsilon(t) = y(t) - \theta^T(t)\,\Phi(t) \tag{3.4}$$

where

$$\Phi(t) = [y(t-1), \cdots, y(t-n), \quad \epsilon(t-1), \cdots, \epsilon(t-r)]^T. \tag{3.5}$$

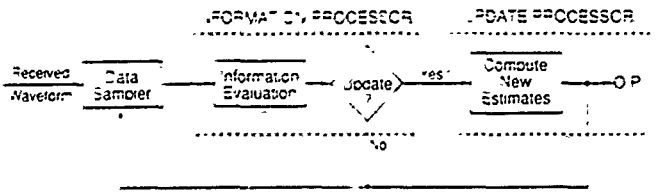


Fig. 1. A modular recursive estimator.

Now just as in the ARX case, define for some suitable γ^2 the convex polytope

$$S_t = \left\{\theta: \left(y(t) - \theta^T\Phi(t)\right)^2 \le \gamma^2, \quad \theta \in \boldsymbol{R}^{n+r}\right\}$$

and the bounding ellipsoid

$$E_t = \left\{\theta \in \boldsymbol{R}^{n+r}: (\theta - \theta(t))^T P^{-1}(t)(\theta - \theta(t)) \le \sigma^2(t)\right\}.$$

The update equations for $\theta(t)$, $P(t)$, and $\sigma^2(t)$, which then follow directly from [11], are as in the ARX case, with the only difference being that the regressor vector is now given by (3.5):

$$P^{-1}(t) = (1-\lambda_t)P^{-1}(t-1) + \lambda_t\Phi(t)\,\Phi^T(t) \tag{3.6a}$$

$$\theta(t) = \theta(t-1) + \lambda_t P(t)\,\Phi(t)\,\delta(t) \tag{3.6b}$$

$$\delta(t) = y(t) - \theta^T(t-1)\,\Phi(t) \tag{3.6c}$$

$$\sigma^2(t) = (1-\lambda_t)\sigma^2(t-1) + \lambda_t\gamma^2 - \frac{\lambda_t(1-\lambda_t)\,\delta^2(t)}{1-\lambda_t+\lambda_t G(t)} \tag{3.6d}$$

where

$$G(t) = \Phi^T(t)\,P(t-1)\,\Phi(t). \tag{3.6e}$$

The matrix inversion lemma can be used in (3.6a) to obtain the following recursion for $P(t)$:

$$P(t) = \frac{1}{1-\lambda_t}\left[P(t-1) - \frac{\lambda_t P(t-1)\,\Phi(t)\,\Phi^T(t)\,P(t-1)}{1-\lambda_t+\lambda_t G(t)}\right]. \tag{3.6f}$$

As in the OBE algorithm, the bounding ellipsoids are optimized by choosing λ_t^* to minimize $\sigma^2(t)$. In order to facilitate the subsequent analysis, the initial conditions are modified to

$$P(0) = MI_{n+r}, \quad \theta(0) = 0, \quad \text{and } \sigma^2(0) = \gamma^2 - \epsilon \tag{3.7}$$

where $M \gg 1$, $\epsilon \ll 1$, and I_{n+r} is the identity matrix of dimension $n+r$. This choice of initial conditions ensures that the initial ellipsoid E_0 will contain the true parameter vector θ^* and, more importantly, as shown in Appendix A, simplifies the optimum forgetting factor

determination formula to

$$\text{If } \sigma^2(t-1) - \hat{\delta}^2(t) \le \gamma^2$$

$$\text{then } \lambda_t^* = 0, \tag{3.8}$$

otherwise

$$\lambda_t^* = \frac{1 - \beta(t)}{2}$$

$$\text{if } G(t) = 1, \tag{3.9a}$$

$$\frac{1}{1 - G(t)}\left[1 - \sqrt{\frac{G(t)}{1 + \beta(t)(G(t) - 1)}}\right]$$

$$\text{if } G(t) \neq 1 \tag{3.9b}$$

where

$$\beta(t) \triangleq \frac{\gamma^2 - \sigma^2(t-1)}{\delta^2(t)}. \tag{3.9c}$$

Remarks:

1) It is shown in Appendix A that if $\sigma^2(t-1) - \hat{\delta}^2(t) \ge \gamma^2$, then λ_t^* given by (3.9) satisfies

$$\left.\frac{d\sigma^2(t)}{d\lambda_t}\right|_{\lambda_t = \lambda_t^*} = 0$$

and furthermore, $0 \le \lambda_t^* < 1$. Thus, unlike [11], no upper bound need be imposed on the forgetting factor.

2) Since $\sigma^2(t) = \sigma^2(t-1)$ if $\lambda_t^* = 0$, any nonzero value of λ_t^* which minimizes $\sigma^2(t)$ will cause $\sigma^2(t) < \sigma^2(t-1)$. Thus, choosing λ_t^* to minimize $\sigma^2(t)$ causes $\{\sigma^2(t)\}$ to be a nonincreasing sequence.

The recursive relations (3.6), the initial conditions (3.7), the selective update strategy (3.8), and the forgetting factor determination formula (3.9) form the Extended Optimal Bounding Ellipsoid (EOBE) estimation algorithm [14]. The choice of the threshold γ^2 will become clear from the analysis below. The algorithm retains the discerning update strategy and the modular adaptive filter structure of the OBE algorithm [11], [13].

IV. Analysis of the EOBE Algorithm

The main difficulty in the analysis of the EOBE algorithm arises from the presence of the *a posteriori* prediction errors in the regressor vector. Unlike the OBE algorithm, in this case, boundedness of $w(t)$ does not guarantee that all the convex polytopes S_t, $t = 1, 2, \cdots$, will contain θ^*. The first step in the analysis is to find conditions under which this happens. The minimization of $\sigma^2(t)$, at every time instant, and the choice of initial conditions (3.7), facilitate the characterization of the behavior of the *a posteriori* prediction errors.

Lemma 1: For the EOBE algorithm of Section III, if $\sigma^2(t-1) + \hat{\delta}^2(t) > \gamma^2$, i.e., if an update occurs at time instant t, then

i) $\sigma^2(t) - \epsilon^2(t) = \gamma^2$, (4.1)

ii) $\sigma^2(k) \le \sigma^2(t)$ for all time instants $k < t$, (4.2)

and if $t - j$ is the time instant at which the next update occurs, then

iii) $\epsilon^2(k) \le \epsilon^2(t)$ for all $k < t - j$. (4.3)

Proof:

i) It has been shown in Appendix A that if $\sigma^2(t-1) - \hat{\delta}^2(t) > \gamma^2$, then the optimum forgetting factor λ_t^* satisfies

$$\left.\frac{d\sigma^2(t)}{d\lambda_t}\right|_{\lambda_t = \lambda_t^*} = 0. \tag{4.4}$$

Taking the derivative in (3.6d) and using (4.4) yields

$$\gamma^2 - \sigma^2(t-1) - \frac{(1-\lambda_t)\delta^2(t)}{1 - \lambda_t + \lambda_t G(t)} = -\frac{\lambda_t \delta^2(t)\, G(t)}{(1 - \lambda_t + \lambda_t G(t))^2} \tag{4.5a}$$

which can be rewritten in the form

$$\gamma^2 - \sigma^2(t-1) = \hat{\delta}^2(t)\frac{(1-\lambda_t)^2 - \lambda_t^2 G(t)}{(1 - \lambda_t + \lambda_t G(t))^2}. \tag{4.5b}$$

In (4.5) and in the remainder of the paper, when there is no risk of confusion, the optimum forgetting factor λ_t^* will be denoted by λ_t. It is also easily shown from (3.6b), (3.6c), and (3.6f) that the *a posteriori* and *a priori* prediction errors are related by

$$\epsilon(t) = \frac{1 - \lambda_t}{1 - \lambda_t + \lambda_t G(t)}\hat{\delta}(t). \tag{4.6}$$

Note that the nonnegativeness of $G(t)$ implies that $\epsilon^2(t) \le \hat{\delta}^2(t)$. Substituting (4.6) in (4.5b) and rearranging terms yields

$$(1 - \lambda_t)\gamma^2 - (1 - \lambda_t)\sigma^2(t-1) = (1 - \lambda_t)\epsilon^2(t) - \frac{\lambda_t^2 G(t)\,\epsilon^2(t)}{1 - \lambda_t}. \tag{4.7}$$

Now using (4.6) in (3.6d) gives

$$\sigma^2(t) = (1 - \lambda_t)\sigma^2(t-1) + \lambda_t\gamma^2 - \lambda_t\epsilon^2(t) - \frac{\lambda_t^2 G(t)}{1 - \lambda_t}\epsilon^2(t). \tag{4.8}$$

Finally, subtracting (4.8) from (4.7) gives (4.1).

ii) *Case 1:* If $k < t$ is an updating instant, then (4.1) gives

$$\sigma^2(k) - \epsilon^2(k) = \gamma^2. \tag{4.9}$$

But since $\{\sigma^2(t)\}$ is a nonincreasing sequence, (4.9) and (4.1) together would imply that

$$\epsilon^2(k) \le \epsilon^2(t).$$

Case 2: If $k < t$ is a nonupdating instant, then $\epsilon^2(k) = \hat{\sigma}^2(k)$, and so by (3.8), $\sigma^2(k-1) - \epsilon^2(k) \leq \cdots$, and since $\sigma^2(t)$ is nonincreasing, $\epsilon^2(k) \leq \epsilon^2(t)$.

iii) Since λ_k, $k = t+1, t+2, \cdots, t+j-1$, are all zero, $\sigma^2(k) = \sigma^2(t)$, for all $t < k < t+j$. And because k is a nonupdating instant, $\sigma^2(k-1) - \epsilon^2(k) = \sigma^2(t) + \epsilon^2(k) \leq \gamma^2$, and so (4.3) follows.

We can now derive sufficient conditions under which the convex polytopes S_t and E_t will contain θ^*.

Theorem 1: The convex polytopes S_t and consequently the ellipsoids E_t, $t = 1, 2, \cdots$, will contain the true parameter, if

i) E_0 contains θ^*, (4.10a)

ii) the true moving average coefficients satisfy

$$\left[\sum_{i=1}^{r} |c_i|\right]^2 < 0.5, \tag{4.10b}$$

iii) the threshold γ^2 satisfies

$$\gamma^2 \geq \left[2\gamma_0^2 \frac{\left[1 + \sum_{i=1}^{r} |c_i|\right]^2}{1 - 2\left[\sum_{i=1}^{r} |c_i|\right]^2}\right]. \tag{4.10c}$$

Proof: Let the induction hypothesis be $\theta^* \in E_{t-1}$. Then defining

$$V(t) = (\theta(t) - \theta^*)^T P^{-1}(t)(\theta(t) - \theta^*) \tag{4.11}$$

and recalling the definition of E_{t-1} yields

$$V(t-1) \leq \sigma^2(t-1) \tag{4.12}$$

and since $P^{-1}(t)$ is positive definite for all t, $\sigma^2(t-1) > 0$.

Now using (3.1) and (3.5)

$$(y(t) - \theta^{*T}\Phi(t))^2 = (C(q^{-1})[w(t)] - (C(q^{-1}) - 1)[\epsilon(t)])^2$$

where the operator $C(q^{-1})$ has been defined in (1.2). Defining $n(t) = C(q^{-1})[w(t)]$, and recalling an elementary algebraic inequality

$$(a-b)^2 \leq 2a^2 + 2b^2$$

yields

$$(y(t) - \theta^{*T}\Phi(t))^2 \leq 2n^2(t) + 2(c_1\epsilon(t-1) - c_2\epsilon(t-2) \cdots - c_r\epsilon(t-r))^2. \tag{4.13}$$

But

$$n^2(t) \leq \tfrac{1}{2}\gamma'^2, \quad \text{for all } t$$

where

$$\gamma'^2 = 2\gamma_0^2\left(1 - \sum_{i=1}^{r} c_i\right)^2. \tag{4.14}$$

Hence,

$$(y(t) - \theta^{*T}\Phi(t))^2 \leq \cdots - 2(c_1\epsilon(t-1) - c_2\epsilon(t-2) - \cdots - c_r\epsilon(t-r))^2. \tag{4.15}$$

But by Lemma 1, if $t-j$ is the updating instant immediately preceding time instant t, then

$$|\epsilon(t-i)| \leq |\epsilon(t-j)| \quad \text{for } 1 \leq i \leq r.$$

Thus

$$(y(t) - \theta^{*T}\Phi(t))^2 \leq \gamma'^2 + 2\left(\sum_{i=1}^{r} |c_i|\right)^2 \epsilon^2(t-j) \leq \gamma'^2 + 2\left(\sum_{i=1}^{r} |c_i|\right)^2 (\gamma^2 - \sigma^2(t-1)).$$

Since $\epsilon^2(t-j) = \gamma^2 - \sigma^2(t-j) = \gamma^2 - \sigma^2(t-1)$. Now by the induction hypothesis, $\sigma^2(t-1) \geq 0$. Hence,

$$(y(t) - \theta^{*T}\Phi(t))^2 \leq \gamma'^2 + 2\left(\sum_{i=1}^{r} |c_i|\right)^2 \gamma^2. \tag{4.16}$$

So the convex polytope S_t will contain θ^* if

$$\gamma'^2 + 2\left(\sum_{i=1}^{r} |c_i|\right)^2 \gamma^2 \leq \gamma^2. \tag{4.17}$$

The inequality (4.17) will hold iff (4.10b) and (4.10c) are true. Assuming (4.10b) and (4.10c) thus guarantees that for all time instants t

$$(y(t) - \theta^{*T}\Phi(t))^2 \leq \gamma^2. \tag{4.18}$$

Using (3.6) and (4.11), it can be shown that

$$V(t) - \sigma^2(t) \leq (1 - \lambda_t)(V(t-1) - \sigma^2(t-1)) + \lambda_t[(y(t) - \theta^{*T}\Phi(t))^2 - \gamma^2] \tag{4.19}$$

and so from (4.18) it follows that

$$V(t) - \sigma^2(t) \leq (1 - \lambda_t)(V(t-1) - \sigma^2(t-1)). \tag{4.20}$$

Finally, by (4.12), it follows that

$$V(t) - \sigma^2(t) \leq 0, \tag{4.21}$$

i.e., E_t contains θ^*, and $\sigma^2(t)$ is nonnegative for all t.

Remarks:

1) The assumption (4.10b) says that the noise sequence $n(t) = C(q^{-1})[w(t)]$ should not be "too colored." This condition is analogous to the Strictly Positive Real (SPR) condition which appears in the ELS algorithm (cf. Section I). It is not very difficult to show that for the SPR condition to hold, it is necessary that

$$\sum_{i=1}^{r} c_i < 1. \tag{4.22}$$

It can also be seen that condition (4.10b) is a stricter form of the Strictly Dominant Passive (SDP) condition [15] which appears in the analysis of some signed LMS algorithms, and from [15], it follows that (4.10b) is sufficient for the SPR condition to hold and hence is more restrictive than the SPR condition.

2) Selection of the right "noise bound" γ^2 is made possible by (4.10c). The user would, however, need to have some knowledge of the magnitude of the true moving average coefficients. Simulation results show that overestimation of γ^2 has very little effect on the parameter estimates (centers of the bounding ellipsoids), although it may have an adverse effect on the size of the bounding ellipsoids.

3) The conditions (4.10b) and (4.10c) are not necessary conditions, and the algorithm has been observed to perform well in several examples where these conditions were violated.

The following result follows straightforwardly from Lemma 1 and Theorem 1.

Corollary 1: If the conditions of Theorem 1 hold then

a) $\lim_{j \to \infty} \epsilon^2(t_j)$ exists (4.23a)

where $\{t_j\}$ is the subsequence of updating instants of the EOBE algorithm, and

b) uniformly bounded *a posteriori* prediction errors

$$\epsilon^2(t) \le \gamma^2, \quad \text{for all time instants } t. \tag{4.23b}$$

Boundedness of $\delta^2(t)$, the *a priori* prediction error, and convergence of the parameter estimates to a neighborhood of the true parameter can be assured by requiring the regressor vector to be persistently exciting. The next lemma relates the positive definiteness of $P^{-1}(t)$ to the richness of the regressor vector $\Phi(t)$.

Lemma 2: If there exist positive α_3 and N such that, for all t

$$\sum_{i=t}^{t+N} \Phi(i)\,\Phi^T(i) \ge \alpha_3 I > 0, \tag{4.24a}$$

then there exists a positive α_4 such that

$$P^{-1}(t) \ge \alpha_4 I > 0. \tag{4.24b}$$

Proof of the lemma is the same as that of Theorem 4.1 of [11], it is thus omitted here.

Remark: The positive definiteness of $P^{-1}(t)$ implies that the eigenvalues of $P(t)$ are upper bounded.

Theorem 2: If the assumptions of Theorem 1 are satisfied and (4.24a) holds, then the EOBE algorithm ensures the following.

a) Parameter difference convergence

$$\lim_{t \to \infty} \|\theta(t) - \theta(t-k)\| = 0$$

for any finite k. (4.25)

b) Asymptotically bounded parameter estimation errors

$$\theta(t) - \theta^* \to \left[0,\ 2\gamma\sqrt{(1 - \Sigma_{i=1}^{n} \alpha_i^2)/\alpha_4}\right] \tag{4.26}$$

where γ, α_i and α_4 are as in (3.2) and (4.24b), respectively.

c) If, in addition, the process (1.1) is stable, then the algorithm yields asymptotically bounded *a priori* prediction errors

$$\delta^2(t) \to [0, \gamma^2]. \tag{4.27}$$

Proof:

a) From (3.6b) and (3.6f)

$$\|\theta(t) - \theta(t-1)\|^2 = \frac{\lambda_t^2 \Phi^T(t)\, P^2(t-1)\, \Phi(t)\, \delta^2(t)}{(1 - \lambda_t + \lambda_t G(t))^2} \tag{4.28}$$

$$\le e_{\max}\{P(t-1)\} \frac{\lambda_t^2 G(t)\, \delta^2(t)}{(1 - \lambda_t + \lambda_t G(t))^2} \tag{4.29}$$

where $e_{\max}\{P(t-1)\}$ is the maximum eigenvalue of $P(t-1)$, and $\|\cdot\|$ denotes the Euclidean norm. Using (3.6d) in (4.5a) yields

$$\sigma^2(t) = \sigma^2(t-1) - \frac{\lambda_t^2 \delta^2(t)\, G(t)}{(1 - \lambda_t + \lambda_t G(t))^2}. \tag{4.30}$$

The nonnegativity of $\sigma^2(t)$ therefore implies

$$\sum_{i=1}^{t} \frac{\lambda_i^2 \delta^2(i)\, G(i)}{(1 - \lambda_i + \lambda_i G(i))^2} = \sigma^2(0) - \sigma^2(t) < \infty, \tag{4.31}$$

Hence,

$$\lim_{t \to \infty} \frac{\lambda^2 \delta^2(t)\, G(t)}{(1 - \lambda_t + \lambda_t G(t))^2} = 0. \tag{4.32}$$

If (4.24a) holds, then by Lemma 2, $e_{\max}\{P(t-1)\}$, the maximum eigenvalue of $P(t-1)$, is bounded for all t, and hence (4.29) and (4.32) yield

$$\|\theta(t) - \theta(t-1)\| \to 0. \tag{4.33}$$

Applying the Minkowski inequality to $\|\theta(t) - \theta(t-k)\|$ and using (4.33) completes the proof of (4.25).

b) Using (3.6), (4.11), and (4.6), an expression similar to (4.19) can be derived as

$$V(t) = (1 - \lambda_t) V(t-1) - \lambda_t \left\{ [C(q^{-1})[w(t)] - [C(q^{-1}) - 1][\epsilon(t)] \right\}^2 - \frac{1 - \lambda_t - \lambda_t G(t)}{1 - \lambda_t} \epsilon^2(t) \Big\}. \tag{4.34}$$

It can also be seen that condition (4.10b) is a stricter form of the Strictly Dominant Passive (SDP) condition [15] which appears in the analysis of some signed LMS algorithms, and from [15], it follows that (4.10b) is sufficient for the SPR condition to hold and hence is more restrictive than the SPR condition.

2) Selection of the right "noise bound" γ^2 is made possible by (4.10c). The user would, however, need to have some knowledge of the magnitude of the true moving average coefficients. Simulation results show that overestimation of γ^2 has very little effect on the parameter estimates (centers of the bounding ellipsoids), although it may have an adverse effect on the size of the bounding ellipsoids.

3) The conditions (4.10b) and (4.10c) are not necessary conditions, and the algorithm has been observed to perform well in several examples where these conditions were violated.

The following result follows straightforwardly from Lemma 1 and Theorem 1.

Corollary 1: If the conditions of Theorem 1 hold then

a) $\lim_{j\to\infty} \epsilon^2(t_j)$ exists (4.23a)

where $\{t_j\}$ is the subsequence of updating instants of the EOBE algorithm, and

b) uniformly bounded *a posteriori* prediction errors

$$\epsilon^2(t) \le \gamma^2, \quad \text{for all time instants } t. \tag{4.23b}$$

Boundedness of $\delta^2(t)$, the *a priori* prediction error, and convergence of the parameter estimates to a neighborhood of the true parameter can be assured by requiring the regressor vector to be persistently exciting. The next lemma relates the positive definiteness of $P^{-1}(t)$ to the richness of the regressor vector $\Phi(t)$.

Lemma 2: If there exist positive α_3 and N such that, for all t

$$\sum_{i=t}^{t+N} \Phi(i)\,\Phi^T(i) \ge \alpha_3 I > 0, \tag{4.24a}$$

then there exists a positive α_4 such that

$$P^{-1}(t) \ge \alpha_4 I > 0. \tag{4.24b}$$

Proof of the lemma is the same as that of Theorem 4.1 of [11], it is thus omitted here.

Remark: The positive definiteness of $P^{-1}(t)$ implies that the eigenvalues of $P(t)$ are upper bounded.

Theorem 2: If the assumptions of Theorem 1 are satisfied and (4.24a) holds, then the EOBE algorithm ensures the following.

a) Parameter difference convergence

$$\lim_{t\to\infty} \|\theta(t) - \theta(t-k)\| = 0$$

for any finite k. (4.25)

b) Asymptotically bounded parameter estimation errors

$$\theta(t) - \theta^* \to \left[0,\ 2\gamma\bar{\delta}\left(1 - \sum_{i=1}^{n} \alpha_i\right)^{-1}\right] \tag{4.26}$$

where γ, $\bar{\delta}$ and α_i are as in (3.2) and (4.24b), respectively.

c) If, in addition, the process (1.1) is stable, then the algorithm yields asymptotically bounded *a priori* prediction errors

$$\delta^2(t) \to [0,\ \gamma^2]. \tag{4.27}$$

Proof:

a) From (3.6b) and (3.6f)

$$\|\theta(t) - \theta(t-1)\|^2 = \frac{\lambda_t^2 \Phi^T(t)\,P^2(t-1)\,\Phi(t)\,\delta^2(t)}{(1 - \lambda_t + \lambda_t G(t))^2} \tag{4.28}$$

$$\le e_{\max}\{P(t-1)\}\,\frac{\lambda_t^2 G(t)\,\delta^2(t)}{(1 - \lambda_t + \lambda_t G(t))^2}, \tag{4.29}$$

where $e_{\max}\{P(t-1)\}$ is the maximum eigenvalue of $P(t-1)$, and $\|\cdot\|$ denotes the Euclidean norm. Using (3.6d) in (4.5a) yields

$$\sigma^2(t) = \sigma^2(t-1) - \frac{\lambda_t^2 \delta^2(t)\,G(t)}{(1 - \lambda_t + \lambda_t G(t))^2}, \tag{4.30}$$

The nonnegativity of $\sigma^2(t)$ therefore implies

$$\sum_{i=1}^{t} \frac{\lambda_i^2 \delta^2(i)\,G(i)}{(1 - \lambda_i + \lambda_i G(i))^2} = \sigma^2(0) - \sigma^2(t) < \infty, \tag{4.31}$$

Hence,

$$\lim_{t\to\infty} \frac{\lambda_t^2 \delta^2(t)\,G(t)}{(1 - \lambda_t + \lambda_t G(t))^2} = 0. \tag{4.32}$$

If (4.24a) holds, then by Lemma 2, $e_{\max}\{P(t-1)\}$, the maximum eigenvalue of $P(t-1)$, is bounded for all t, and hence (4.29) and (4.32) yield

$$\|\theta(t) - \theta(t-1)\| \to 0. \tag{4.33}$$

Applying the Minkowski inequality to $\|\theta(t) - \theta(t-k)\|$ and using (4.33) completes the proof of (4.25).

b) Using (3.6), (4.11), and (4.6), an expression similar to (4.19) can be derived as

$$V(t) = (1 - \lambda_t)V(t-1) - \lambda_t\left[\,[C(q^{-1})[w(t)]^2 - (C(q^{-1}) - 1)[\epsilon(t)]\right]^2 - \frac{1 - \lambda_t - \lambda_t G(t)}{1 - \lambda_t}\,\epsilon^2(t)\Big]. \tag{4.34}$$

Just as in the proof of Theorem 1, (4.34) can be expressed as

$$V(t) \le (1 - \lambda_t)V(t-1) - \lambda_t\gamma'^2 + \lambda_t\left[2\left(\sum_{i=1}^{r} c_i\right)^2 \epsilon^2(t-j) - \frac{1 - \lambda_t + \lambda_t G(t)}{1 - \lambda_t}\epsilon^2(t)\right] \quad (4.35)$$

where γ'^2 is as in (4.14), and $t - j$ is the updating instant immediately preceding time instant t. Assume t is an updating instant. Then (4.10b), (4.2), and the nonnegativity of $G(t)$ would imply that the term in square brackets on the right-hand side of (4.35) is not positive, and so

$$V(t) \le (1 - \lambda_t)V(t-1) + \lambda_t\gamma'^2. \quad (4.36)$$

It is obvious that if t is not an updating instant, then (4.36) would still follow from (4.35). A nonrecursive form for (4.36) can be obtained as

$$V(t) \le \prod_{i=1}^{t} (1 - \lambda_i)V(0) + \gamma'^2 \sum_{i=1}^{t} q_{it} \quad (4.37)$$

where

$$q_{it} = \begin{cases} \lambda_i(1 - \lambda_{i+1}) \cdots (1 - \lambda_t), & i < t \\ \lambda_t & i = t. \end{cases} \quad (4.38)$$

For large t, the first term on the right-hand side of (4.37) can be neglected. In Appendix B, it is shown that

$$\sum_{i=1}^{t} q_{it} < 1. \quad (4.39)$$

Hence, for large enough t

$$V(t) = (\theta(t) - \theta^*)^T P^{-1}(t)(\theta(t) - \theta^*) \le \gamma'^2 \quad (4.40)$$

And so (4.26) follows from Lemma 2 and (4.14).

c) Stability of the process (1.1) and the boundedness of $w(t)$ implies that the outputs $y(t)$ are bounded. Hence, from (3.6e), (4.23b), and Lemma 2, it follows that

$$G(t) \le e_{\max}\{P(t-1)\}\left[r\gamma^2 + n \max_{t-n \le i \le t-1} y^2(i)\right] < \infty \quad (4.41)$$

where n is the order of the AR process and r is the order of the MA process. It can now be shown, just as in Theorem 3.2 of [11], that the *a priori* prediction errors satisfy (4.27).

Remarks:

1) The results of Theorem 1, and the results (4.25), (4.26) of Theorem 2, do not require the process to be stable. However, if the process is unstable, then on account of finite precision effects, the matrix $P(t)$ may not stay positive definite, thus invalidating the notion of bounding ellipsoids and causing the algorithm to fail. In this situation, the ELS algorithm will fail, too.

2) Theorems 1 and 2 do not impose any statistical properties on the input noise sequence $\{w(t)\}$. However, our simulation experience has been that the parameter estimates are usually not close to the true parameters if the noise is not white. Of course, such is also the case for the ELS algorithm.

V. Simulation Results

Simulations have been performed to investigate the performance of the EOBE algorithm vis à vis the ELS algorithm. In this paper, we present simulation results for two examples—a broad-band ARMA (3, 3) process and a narrow-band ARMA (2, 2) process where the indexes n, r in an ARMA (n, r) process refer to the orders of the $A(q^{-1})$ and $C(q^{-1})$ polynomials, respectively.

Example 1—Broad-band ARMA (3, 3) Process: The output data $\{y(t)\}$ are generated by the following difference equation:

$$y(t) = -0.4y(t-1) + 0.2y(t-2) + 0.6y(t-3) + w(t) - 0.22w(t-1) + 0.17w(t-2) - 0.1w(t-3).$$

The noise sequence $\{w(t)\}$ is generated by a pseudo-random number generator with a uniform probability distribution in $[-1.0, 1.0]$. The upper bound γ^2 was set equal to 25. The parameter estimates were obtained by applying the EOBE algorithm to 1000 point data sequences. Twenty-five runs of the algorithm were performed on the same model but with different input noise sequences. The average squared parameter error $L_1(t)$ is computed for the AR coefficients according to the formula

$$L(t) = \frac{1}{25}\sum_{j=1}^{25} l_j(t)$$

where $l_j(t)$, the squared AR parameter error at time t for the jth run, is defined by

$$l_j(t) = \sum_{i=1}^{n} (a_i(t) - a_i)^2$$

with a_i and $a_i(t)$ being defined by (1.1) and (3.3), respectively. The average squared parameter error $L_2(t)$ for the MA coefficients is defined analogously. Figs. 2 and 3 display the average squared estimation errors for AR and MA parameters using both the EOBE and the ELS algorithms. The curves show that the performance of the two algorithms is comparable. The average number of updates for the EOBE algorithm was 160 for 1000 point data sequences. Thus, only 16% of the samples are used for updates, as compared to the ELS algorithm which updates at every sampling instant.

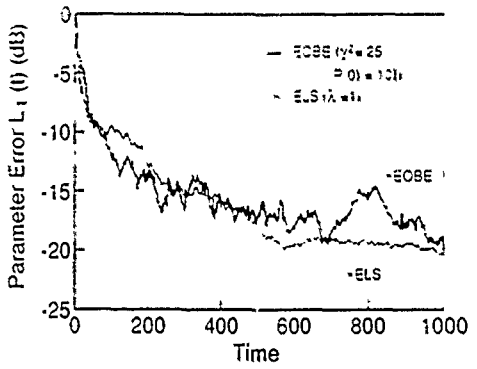


Fig. 2. Average squared AR parameter estimation error for the EOBE and ELS algorithms—Example 1.

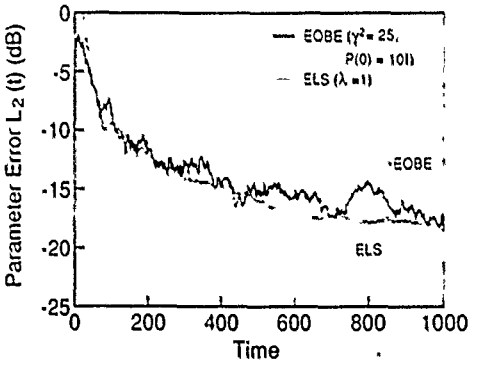


Fig. 3. Average squared MA parameter estimation error for the EOBE and ELS algorithms—Example 1.

TABLE I

Upper Bound γ^2	Average Tap Error	Average Number of Updates	Total Number of Times θ^* is Out of S_t	Total Number of Times θ^* is Out of E_t	Average Final Volume	Average Final Sum of Axes
0.5	0.031	160	7309	23952	—	—
1.0	0.031	160	315	0	0.22	10.46
2.0	0.031	160	0	0	2.6×10^3	74
5.0	0.031	154	0	0	5.4×10^7	265
25.0	0.031	153	0	0	2.1×10^{12}	1537
100.0	0.0308	156	0	0	1.0×10^{16}	6303

The effect of different choices for the upper bound γ^2 on the performance has also been studied. For each value of γ^2, the asymptotic average squared parameter error T was computed over 25 runs of the algorithm, according to the formula

$$T = \frac{1}{25} \sum_{j=1}^{25} \left\| \theta_j(1000) - \theta^* \right\|^2$$

where $\theta_j(1000)$ is the parameter estimate at the 1000th iteration in the jth run. The lower bound on γ^2 as calculated from (4.10c) is $\gamma^2 \geq 8.54$. The second column of Table I lists the different values of T obtained when γ^2 is varied from 0.5 to 100. It is clear that the centers of the bounding ellipsoids are insensitive to the value of γ^2, since the tap error is almost constant. However, the final size of the ellipsoids does depend on γ^2. The negative volume obtained when $\gamma^2 = 0.5$ is an indication of the fact that $\sigma^2(t)$ is no longer positive and so bounding ellipsoids cannot be constructed.

The performance of the algorithm, when the noise sequence $\{w(t)\}$ has a Gaussian distribution, was evaluated in a similar fashion. A constant value of $\gamma^2 = 25$ was used and the standard deviation of the noise was varied. The results for 25 runs of the algorithm are shown in Table II. It is clear that the unbounded noise has marginal effect on the parameter estimates

Finally, the tracking capability of the EOBE algorithm was compared to that of the ELS algorithm (with forgetting factor = 0.99). The same model was used to generate 400 data points. The parameters were then changed by 150% and the next 400 points were generated. Finally the last 200 points were generated by using the original parameters. The average squared parameter error was

[illegible] II

Standard Deviation of Noise	Average Tap Error	Average Number of Updates	Total Number of Times [illegible] Out of [illegible]	Total Number of Times [illegible] Out [illegible]	Average [illegible] Volume	Average Final Sum of Semiaxes
0.5	0.29	93	0		[illegible]	575
1.0	0.317	108	3		[illegible]	333
2.0	0.313	117	323		[illegible]	20
3.0	0.32	122	2439	1978	—	—

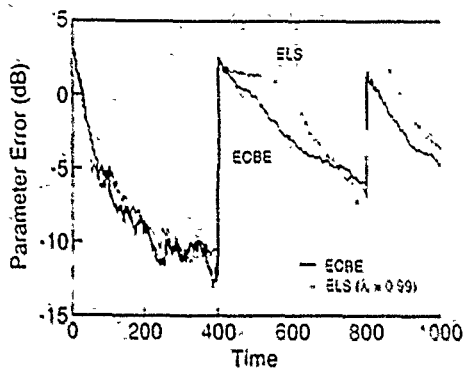


Fig. 4. Tracking performance of the EOBE and ELS algorithms—average squared parameter estimation error.

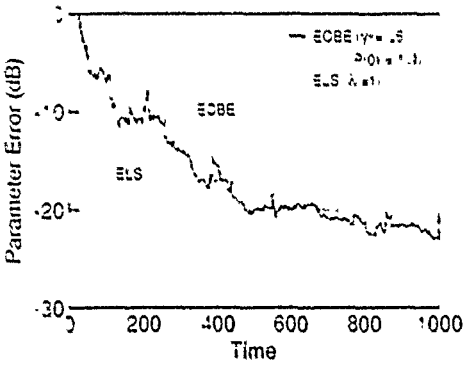


Fig. 6. Average squared MA parameter estimation error for the EOBE and ELS algorithms—Example 2.

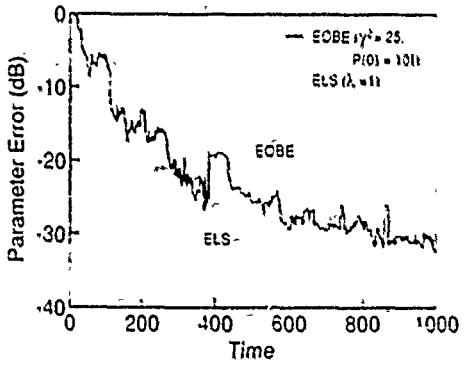


Fig. 5. Average squared AR parameter estimation error for the EOBE and ELS algorithms—Example 2.

evaluated over 25 runs and is shown in Fig. 4. Even though the formulation of bounding ellipsoids is based on the assumption that the parameters are constant, the simulation results show that the algorithm is able to accommodate changes in model parameters. Analysis of the tracking ability of the algorithm is currently under investigation.

Example 2—Narrow-band ARMA (2, 2) Process: The output data $\{y(t)\}$ are generated by the following difference equation:

$$y(t) = 1.4y(t-1) - 0.95y(t-2) + w(t) - 0.86w(t-1) - 0.431w(t-2).$$

Note that, in this case, condition (4.10b) of Theorem 1 is violated. The noise sequence is uniformly distributed in $[-1.0, 1.0]$, as in the first example. The upper bound γ^2 was set equal to 25. The average squared AR and MA parameter estimation errors are calculated over twenty five runs and plotted in Figs. 5 and 6, respectively. The average number of updates was 78 for 1000 point data sequences.

For this example too, different values of the upper bound γ^2 were used and no significant difference in the quality of estimates, number of updates or convergence rate was observed. Thus, it is verified once again that a precise knowledge of the upper bound is not a prerequisite for satisfactory performance of the algorithm.

VI. Conclusion

A recursive parameter estimation algorithm has been extended for ARMA parameter estimation. The main features of the algorithm are a membership set theoretic formulation and a discerning update strategy. Convergence analysis of the algorithm has been performed under the assumption that the noise is bounded. The main results of the analysis are that all the bounding ellipsoids will contain the true parameter, provided the true moving average coefficients satisfy a condition, which is analogous to the SPR condition of the ELS algorithm. In addition, the algorithm yields uniformly bounded *a posteriori* prediction errors. With a persistence of excitation condition on the regressor vector, boundedness of the *a priori* prediction errors can then be established and the parameter estimates are shown to converge to a neighborhood of the true parameters. Simulation results show that the performance of the algorithm is comparable to the ELS algorithm while requiring far fewer updates.

Appendix A

Proof of (3.8) and (3.9): The proof is along the lines of the proof of Lemma 2.1 in [11].

Since λ_t^* minimizes $\sigma^2(t)$

$$\sigma^2(t, \lambda_t^*) \leq \sigma^2(t, 0) = \sigma^2(t-1) \tag{A.1}$$

and

$$\frac{d\sigma^2(t)}{d\lambda_t} = \gamma^2 - \sigma^2(t-1) - \hat{\delta}^2(t)\frac{(1-\lambda_t)^2 - \lambda_t^2 G(t)}{(1-\lambda_t + \lambda_t G(t))^2} \tag{A.2}$$

and

$$\frac{d^2\sigma^2(t)}{d\lambda_t^2} = \frac{2\hat{\delta}^2(t)\,G(t)}{(1-\lambda_t+\lambda_t G(t))^3}. \tag{A.3}$$

Thus, $d^2\sigma^2(t)/d\lambda_t^2 > 0$, unless $\hat{\delta}^2(t) = 0$ or $G(t) = 0$. Since $P(t-1)$ is positive definite, $G(t) = 0$ iff $\Phi(t) = 0$. The algorithm can be modified to detect the occurrence of a null $\Phi(t)$ and set it to a small nonzero value, prior to the calculation of $G(t)$. Thus, it can be assumed that $G(t) \neq 0$ for all t. If $\hat{\delta}^2(t) = 0$ ($\sigma^2(t-1) + \hat{\delta}^2(t) < \gamma^2$ in this case), then, since $\sigma^2(0) < \gamma^2$ by (3.7), and since $\sigma^2(t)$ is nonincreasing, therefore, by (A.2) $d\sigma^2(t)/d\lambda_t$ is positive, and hence $\sigma^2(t)$ is minimized if $\lambda_t^* = 0$. Now, for the sequel, the second derivative of $\sigma^2(t)$ can be assumed to be positive, and hence the unique minimum occurs at $d\sigma^2(t)/d\lambda_t = 0$. From (A.2), if $G(t) = 1$, $\sigma^2(t)$ is minimized if

$$\lambda_t^* = (1-\beta(t))/2. \tag{A.4}$$

Otherwise, if $G(t) \neq 1$, $\sigma^2(t)$ is minimized if

$$\lambda_t^* = \frac{1}{1-G(t)}\left[1 - \sqrt{\frac{G(t)}{1+\beta(t)(G(t)-1)}}\right]. \tag{A.5}$$

Moreover, in (A.4) and (A.5)

$$\lambda_t^* > 0 \Leftrightarrow \beta(t) < 1 \Leftrightarrow \sigma^2(t-1) + \hat{\delta}^2(t) > \gamma^2. \tag{A.6}$$

It is easy to show that $1 + \beta(t)(G(t)-1)$ is always positive. Since $\sigma^2(0) < \gamma^2$ and $\sigma^2(t)$ is nonincreasing, therefore, $\beta(t) > 0$. From (A.6), $\beta(t) < 1$, hence $1 - 1/\beta(t) < 0$. Then

$$1 + \beta(t)(G(t)-1) \leq 0 \Rightarrow G(t) \leq 1 - 1/\beta(t) \Rightarrow G(t) < 0$$

which is a contradiction. Thus, (A.5) would always yield real λ_t^*. It is now shown that (A.4) and (A.5) yield values of λ_t^* which are upper bounded by unity. If $G(t) = 1$, then since $\beta(t) > 0$, (A.4) yields $\lambda_t^* < 1$. If $G(t) < 1$, then $\lambda_t^* \geq 1 \Leftrightarrow 1 - [G(t)/(1+\beta(t)(G(t)-1))]^{1/2} \geq 1 - G(t)$

$$\Leftrightarrow G(t)(1+\beta(t)(G(t)-1)) \geq 1. \tag{A.7}$$

But $G(t) < 1$ and $\beta(t) > 0$ contradict (A.7). Hence, if $G(t) < 1$, then $\lambda_t^* < 1$. It can be shown in exactly the same way that $G(t) > 1$ would imply that $\lambda_t^* < 1$. Thus, unlike the case in [11], no upper bound has to be imposed on the forgetting factor.

Appendix B

Proof of (4.39) (by Induction): Let

$$R(t) = \sum_{i=1}^{t} q_{it}. \tag{B.1}$$

Then

$$R(t) = (1-\lambda_t)R(t-1) + \lambda_t \tag{B.2}$$

and

$$R(1) = \lambda_1 < 1.$$

Assume

$$R(t-1) < 1.$$

Then by (B.2)

$$R(t) < (1-\lambda_t)\cdot 1 + \lambda_t,$$

i.e.,

$$R(t) < 1.$$

References

[1] B. Friedlander, "System identification techniques for adaptive signal processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 240-246, Apr. 1982.

[2] J. A. Cadzow, "Spectral estimation: An overdetermined rational model equation approach," *Proc. IEEE*, vol. 70, pp. 907-939, Sept. 1982.

[3] B. Friedlander and B. Porat, "A spectral matching technique for ARMA parameter estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 338-343, Apr. 1984.

[4] M. Kaveh, "High resolution spectral estimation for noisy signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 286-287, June 1979.

[5] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, MA: M.I.T. Press, 1983.

[6] D. Q. Mayne *et al.*, "Linear identification of ARMA processes," *Automatica* vol. 18, no. 4, pp. 461-466, July 1982.

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

[8] M. Milanese and G. Belaforte, "Estimation theory and uncertainty intervals," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 408-414, Apr. 1982.

[9] E. Fogel and Y. F. Huang, "On the value of information in system identification—bounded noise case," *Automatica*, vol. 18, no. 2, pp. 229-238, Mar. 1982.

[10] Y. F. Huang, "A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1131-1134, Oct. 1986.

[11] S. Dasgupta and Y. F. Huang, "Asymptotically convergent modified recursive least-squares," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 383-392, May 1987.

[12] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*. Englewood Cliffs, NJ: Prentice Hall, 1982.

[13] Y. F. Huang and R. W. Liu, "A high level parallel pipelined network architecture for adaptive signal processing," in *Proc. 26th IEEE Conf. Decision Contr.*, vol. 1, Dec. 1987, pp. 662-666.

[14] A. K. Rao, "Applications and extensions of a novel recursive estimation algorithm with selective updating," M.S. thesis, Dep. Elec. Comput. Eng., Univ. Notre Dame, Notre Dame, IN, Aug. 1987.

[15] S. Dasgupta, J. S. Garnett, and C. R. Johnson, "Convergence of an adaptive filter with signed error filtering," in *Proc. 1988 Amer. Contr. Conf.*, Atlanta, vol. 1, pp. 400-405.

Ashok K. Rao was born in New Delhi, India, on January 12, 1962. He received the B. Tech. degree in electrical engineering from the Indian Institute of Technology, Delhi, India, in 1984, the M.S. degree in electrical engineering, and the Ph.D. degree, both from the University of Notre Dame in 1987 and 1989, respectively.

He is currently a Member of the Technical Staff at COMSAT Laboratories. His main research interests are in parameter estimation, adaptive filtering, communication theory and image coding.

Yih-Fang Huang (S'80-M'82) received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, in 1976, the M.S.E.E. degree in electrical engineering from University of Notre Dame, Notre Dame, IN, in 1979, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, 1982.

In 1982 he joined the Department of Electrical and Computer Engineering at University of Notre Dame. In May 1987 he became an Associate Professor at Notre Dame. His research interests are in statistical signal processing and artificial neural networks.

Dr. Huang is currently Associate Editor on Neural Networks and Signal Processing for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS.

Soura Dasgupta was born in Calcutta, India, in April 1959. He received the B.E. degree in electrical engineering from the University of Queensland, Australia, in 1980. He received the Ph.D. degree in systems engineering from the Australian National University, Canberra, in 1984.

In 1981 he was a Research Fellow at the Electronics and Communication Sciences Unit in the Indian Statistical Institute, Calcutta. In the academic year of 1984-1985 he was a Visiting Assistant Professor in the Department of Electrical Engineering at the University of Notre Dame. He is currently with the Department of Electrical and Computer Engineering at the University of Iowa. His research interests are in adaptive systems theory, robust stability, and neural networks.

Publication [2]

Publication [3]

Mathematics and Computers in Simulation 32 (1990) 515–526
North-Holland

RECENT DEVELOPMENTS IN OPTIMAL BOUNDING ELLIPSOIDAL PARAMETER ESTIMATION

Ashok K. RAO
COMSAT LABS, Clarksburg, MD 20874, USA

Yih-Fang HUANG
Department of Electrical & Computer Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

The Optimal Bounding Ellipsoid (OBE) algorithms are viable alternatives to conventional adaptive filtering algorithms in situations where the noise does not satisfy the usual stationarity and whiteness assumptions. An example is shown in which the performance of an OBE algorithm is seen to be markedly superior to that of the recursive least-squares algorithm. Subsequently, an overview of some recent work in the area of OBE parameter estimation is presented. A lattice filter implementation of one particular OBE algorithm is first described. The extension of the OBE algorithm to the estimation of parameters of ARMA models is performed and the results of a convergence analysis are presented. It is demonstrated through a simulation example that the transient performance of the proposed algorithm is superior to that of the well-known extended least-squares algorithm.

1. Introduction

In recent years, there has been a resurgence of interest in an alternative approach to parameter estimation, which has been termed membership set parameter estimation by some authors [1,2]. This approach is particularly appropriate when the probability distribution of the disturbances is unknown, and a bound on the magnitude of the disturbances is available [2,3]. In contrast to conventional system identification schemes (e.g. maximum likelihood, least squares etc. [4]) which yield point estimates of the parameters, a membership set algorithm yields a set of parameter estimates which are compatible with the model, data, and noise bounds. This set of parameters, which is usually a convex polytope in the parameter space, may become extremely complicated to formulate and so it may be necessary to approximate the set.

In this paper, the discussions will be concentrated on the ellipsoidal outer bounding approach which approximates the exact membership set at each instant by an ellipsoid in the parameter space. The algorithms in this class [2,5–7] are temporally recursive and yield ellipsoids which are optimal, in a sense to be defined later. The computational complexity of the Optimal Bounding Ellipsoid (OBE) algorithms is much lower than that of the exact polytope bounding algorithms [8] and non-recursive linear programming based algorithms [9]. The ellipsoidal formulation also helps to make the analysis tractable. Furthermore, a discerning update strategy, which proves to be appealing for recursive algorithms, evolves quite naturally in the optimization of the

ellipsoids. A disadvantage of the OBE algorithms is the possible looseness of the ellipsoidal outer bounds.

The objective of this paper is to provide an overview of some recent developments and applications of the OBE algorithms. It begins by providing a brief review of the various OBE algorithms. The superiority of the algorithms vis à vis commonly used algorithms like the Recursive Least-Squares (RLS) algorithm in situations where the noise does not satisfy the conventional stationarity and whiteness assumptions will be demonstrated by means of an example. In Section 3, an approximate lattice implementation of one of the OBE algorithms will be described [10]. An extension of the OBE algorithm to the estimation of parameters of ARMA models will be presented in Section 4. A simulation example will be presented to compare the transient performance of the extended algorithm to that of the well-known Extended Least-Squares (ELS) algorithm.

2. The OBE algorithms

The OBE algorithms estimate the coefficients of autoregressive with exogeneous input (ARX) processes described by [11]

$$y(t) = a_1 y(t-1) + \cdots + a_n y(t-n) + b_0 u(t) + b_1 u(t-1) + \cdots + b_m u(t-m) + v(t), \tag{1}$$

where t is the integer sample number and $y(t)$, $u(t)$ and $v(t)$ denote the output, input and the noise term, respectively. This equation can be recast as

$$y(t) = \theta^{*\mathrm{T}} \phi(t) + v(t), \tag{2}$$

where

$$\theta^* = [a_1, a_2, \ldots, a_n, b_0, b_1, \ldots, b_m]^{\mathrm{T}}$$

is the vector of true parameters, and

$$\phi(t) = [y(t-1), y(t-2), \ldots, y(t-n), u(t), u(t-1), \ldots, u(t-m)]^{\mathrm{T}}$$

is the regressor vector. It is assumed that the noise is uniformly bounded in magnitude, i.e., there exists a known $\gamma_0 \geq 0$, such that for all t,

$$v^2(t) \leq \gamma_0^2. \tag{3}$$

Combining (2) and (3) yields

$$\left(y(t) - \theta^{*\mathrm{T}} \phi(t)\right)^2 \leq \gamma_0^2. \tag{4}$$

Let S_t be a subset of the Euclidean space $\mathbb{R}^{n+m+1}$, defined by

$$S_t = \left\{\theta: \left(y(t) - \theta^{\mathrm{T}} \phi(t)\right)^2 \leq \gamma_0^2,\ \theta \in \mathbb{R}^{n+m+1}\right\}. \tag{5}$$

The OBE algorithms start off with a large ellipsoid, E_0, in $\mathbb{R}^{n+m+1}$ which contains all admissible

values of the model parameter vector θ^*. After the first observation $y(1)$ is acquired, an ellipsoid is found which bounds the intersection of E_0 and the convex polytope S_1. To hasten convergence, this ellipsoid must be optimized in some sense, say minimum volume, minimum trace [2,7]; or by any other criterion [6]. Denoting the optimal ellipsoid by E_1, one can proceed exactly as before with future observations and obtain a sequence of optimal bounding ellipsoids $\{E_t\}$. The center of the ellipsoid E_t can be taken as the parameter estimate at the tth instant and is denoted by $\theta(t)$. If at a particular time instant t, the resulting optimal bounding ellipsoid would be of a "smaller size", thereby implying that the data point $y(t)$ contains some fresh "information" regarding the parameter estimates, then the parameter estimates are updated. Otherwise E_t is set equal to E_{t-1}, and the estimates are not updated. In essence, the recursive estimator consists of two modules, an information evaluator followed by an updating processor. At each data point, the received data proceed to the updating processor only if the information evaluator indicates that some fresh information is contained in the data. For details of the minimum volume OBE algorithm, one may refer to, e.g., [2,7,12].

The subsequent discussions will be focused on a particular OBE algorithm [6]. The optimization criterion for the OBE algorithm of [6] is defined in terms of a certain upper bound on the estimation error. Such a criterion yields several advantages over not only the minimum volume and minimum trace OBE algorithms, but also other membership set algorithms mentioned in [12]. The updating criterion is simpler, and the presence of an information dependent updating/forgetting factor enables the algorithm to track slow time variations in the parameters. Analysis of the algorithm shows that if the input is sufficiently rich, as defined in [6], and the noise is uncorrelated with the inputs then the prediction error is asymptotically bounded by the noise bound and the parameter estimation error is bounded by a quantity proportional to the noise bound. In addition, asymptotic cessation of updating is guaranteed in the fixed parameter case. These properties are not apparent in the other membership set algorithms.

For the OBE algorithm of [6], the bounding ellipsoid at the tth instant is formulated as

$$E_t = \left\{ \theta \in \mathbb{R}^{n+m+1} : (\theta - \theta(t))^{\mathrm{T}} P^{-1}(t)(\theta - \theta(t)) \leqslant \sigma^2(t) \right\} \tag{6}$$

for some positive definite matrix $P(t)$ and a non-negative scalar $\sigma^2(t)$. The size of the bounding ellipsoid is related to the scalar $\sigma^2(t)$ and the eigenvalues of $P(t)$. The update equations for $\theta(t)$, $P(t)$ and $\sigma^2(t)$, derived in [6], are as follows:

$$\theta(t) = \theta(t-1) + K(t)\delta(t), \tag{7a}$$

$$\delta(t) = y(t) - \theta^{\mathrm{T}}(t-1)\phi(t), \tag{7b}$$

$$K(t) = \frac{\lambda_t P(t-1)\phi(t)}{1 - \lambda_t + \lambda_t G(t)}, \tag{7c}$$

$$G(t) = \phi^{\mathrm{T}}(t) P(t-1)\phi(t), \tag{7d}$$

$$P(t) = \frac{1}{1-\lambda_t}\left[I - K(t)\phi^{\mathrm{T}}(t) \right] P(t-1), \tag{7e}$$

$$\sigma^2(t) = (1+\lambda_t)\sigma^2(t-1) + \lambda_t \gamma_0^2 - \frac{\lambda_t(1-\lambda_t)\delta^2(t)}{1-\lambda_t+\lambda_t G(t)}. \tag{7f}$$

The optimal ellipsoid E_t which bounds the intersection of E_{t-1} and S_t is defined in terms of an optimal value of the updating gain factor λ_t, where $0 \leqslant \lambda_t \leqslant \alpha < 1$, with α being a user chosen

upper bound on the updating gain factor. The optimum value of λ_t is determined by minimization of $\sigma^2(t)$ with respect to λ_t at every time instant. The minimization procedure results in a discerning update procedure. In particular, λ_t is set equal to zero (no update) if

$$\sigma^2(t-1)+\delta^2(t)\leqslant\gamma_0^2. \tag{8}$$

On the other hand, if (8) is not satisfied, then the optimal values of λ_t is computed as follows:

$$\lambda_t=\min(\alpha,\nu),$$

with

$$\nu=\begin{cases}\alpha & \text{if } \delta^2(t)=0,\\ [1-\beta(t)]/2 & \text{if } G(t)=1,\\ \dfrac{1}{1-G(t)}\left[1-\left\{\dfrac{G(t)}{1+\beta(t)(G(t)-1)}\right\}^{1/2}\right] & \text{if } 1+\beta(t)(G(t)-1)>0,\\ \alpha & \text{if } 1+\beta(t)(G(t)-1)\leqslant 0\end{cases} \tag{9}$$

and

$$\beta(t)=\left(\gamma_0^2-\sigma^2(t-1)\right)/\delta^2(t). \tag{10}$$

The recursions (7), and the selective update strategy, along with the initial values

$$P^{-1}(0)=I,\quad \theta(0)=0\quad\text{and}\quad \sigma^2(0)=1/\Delta \text{ with } \Delta\ll 1 \tag{11}$$

form the OBE estimation algorithm. The value chosen for the upper bound γ_0 need not be a tight bound on the noise magnitude since the parameter estimates are not affected by an overestimation of the noise bound [13, p. 52]. Overestimation of the noise bound however, will cause the bounding ellipsoids to be larger. Underestimating the noise bound may cause $\sigma^2(t)$ to become negative at some instant, thereby causing the bounding ellipsoid to vanish. In this case, a recovery procedure may be activated to either increase the size of the ellipsoid E_{t-1} or increase the width of S_t by increasing γ_0.

A striking feature of the OBE algorithms is their similarity to the Recursive Weighted Least Squares (RWLS) with forgetting factor algorithm. In fact the OBE algorithm of [6] can be considered a special case of the RWLS with forgetting factor algorithm with a weighting factor λ_t and a forgetting factor $1-\lambda_t$. However, the intelligent selection of the weighting factor λ_t makes the actual behavior of the OBE algorithms quite different from that of the RWLS.

The RLS algorithms have become increasingly popular in the fields of adaptive signal processing and adaptive control. It is therefore worthwhile to investigate situations in which the use of the OBE algorithms would be preferred to the RLS algorithms. For example, those cases in which the statistical nature of the noise is unknown or in which the noise does not satisfy the usual stationarity and whiteness assumptions seem particularly appropriate for the OBE algorithms. Milanese and Belforte [9] have demonstrated the superiority of the Minimum Uncertainty Interval Correct Estimator (MUICE) over the least-squares estimate for a third-order moving average model where the noise is proportional to the magnitude of the output. A comparison between the OMNE algorithm and least-squares for a non-linear biological model has been presented in [14]. However the OMNE and MUICE are non-recursive and more

computationally intensive than least-squares algorithms and it is perhaps fairer to compare the RLS algorithms with the OBE algorithms which have similar computational complexity. For the sake of illustration, we present an example below, in which the noise is quasi-stationary [4], and compare the performance of the OBE algorithm of [6] to the standard unweighted RLS algorithm.

Example 1. The following ARX (2,2) model is considered

$$y(t) = -0.4\, y(t-1) - 0.85\, y(t-2) - 0.2\, u(t) - 0.7\, u(t-1) + v(t),$$

where the measureable input $u(t)$ is white and uniformly distributed in $[-1, 1]$ and $v(t)$ is a sinusoid in white noise $w(t)$. Such a situation could arise when the observations are affected by power supply hum or other electromagnetic interference. The following model for $v(t)$ is assumed:

$$v(t) = (1-\beta)w(t) + \beta \sin(\pi t/10).$$

The white noise sequence $w(t)$ is also uniformly distributed in $[-1, 1]$ and is uncorrelated with the input sequence. The value of β is varied from 0 to 1 and for each value of β, ten Monte Carlo runs of the OBE and RLS algorithms are performed with data records of 500 points each. The value of the upper bound on the updating gain factor is $\alpha = 0.5$ and the upper bound on the noise is $\gamma_0 = 1.0$. The final parameter estimation error $(\theta^* - \theta(500))^T(\theta^* - \theta(500))$ is averaged over the ten runs and is displayed in *Fig. 1* for β ranging from zero to one. Notice that the parameter estimates of the RLS algorithm are unacceptable for larger values of β. In contrast, the performance of the OBE algorithm is relatively constant over the range of β. The performance of the OBE algorithm has also been observed to be superior to the RLS algorithm for other cases in which the noise is impulsive and bursty [13].

In conclusion, it can be noted that the similarity of the OBE algorithms to the RLS algorithms facilitates the analysis of the algorithms and eases the development of numerically superior and faster implementations of the OBE algorithms. Analysis of finite precision effects in the OBE algorithm of [6] has been performed in [13,15] and upper bounds on the parameter estimation error due to finite word-length computations have been derived. It has also been shown that the time recursion for the matrix $P(t)$ in the OBE algorithm of [6] is less susceptible to round-off

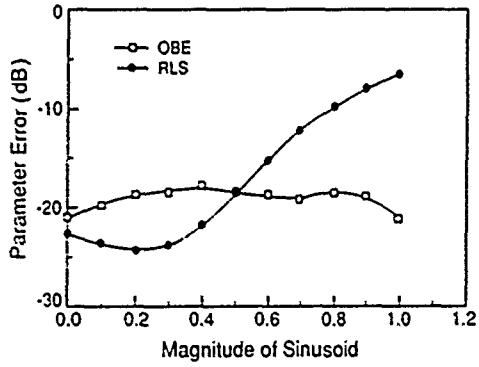


Fig. 1. Mean-squared parameter estimation errors of the OBE and RLS algorithms in white noise mixed with sinusoidal noise (Example 1).

errors than the corresponding recursion in the RLS algorithm. As in the RLS case, Bierman's UDU^T factorization can be performed straightforwardly, to update the P matrix in the OBE algorithm, in a numerically stable fashion. Systolic array implementations of the algorithm have been reported in [16,17]. Thus there exists the potential to apply well established techniques from the adaptive filtering and system identification literature to bounding ellipsoid algorithms.

3. Lattice implementation

Lattice-filter implementations [18] of adaptive algorithms have become popular for a number of reasons. Among others, the more prominent ones are: (1) the modular structure of lattice filters which renders them particularly suitable for VLSI implementations; (2) the low sensitivity of the filter to numerical perturbations in the lattice coefficients; and (3) the fact that the lattice coefficients are independent of the filter order, thus making it possible to add successive lattice stages or subtract existing ones without recalculating the already existing coefficients. In this section, an outline of the lattice-filter formulation of the OBE algorithm of [6] is presented. This lattice-filter implementation appears to retain all the above mentioned advantages. Details of the implementation and simulation results have been presented in [10] and [19].

Consider the following well-known RLS lattice recursions [18] for an AR model

$$e_m(t) = e_{m-1}(t) - k^{\mathrm{b}}_{m-1}(t-1)r_{m-1}(t-1) \quad \text{for } m = 1, 2, \ldots, N, \tag{12}$$

$$r_m(t) = r_{m-1}(t-1) - k^{\mathrm{f}}_{m-1}(t-1)e_{m-1}(t) \quad \text{for } m = 1, 2, \ldots, N, \tag{13}$$

where, $e_0(t) = r_0(t) = y(t)$, $e_m(t)$ is the forward prediction error of order m, $r_m(t)$ is the backward prediction error of order m and k^{b}_m and k^{f}_m are, respectively, the mth backward and forward partial correlation (PARCOR) coefficients. Iterating (12) up to order $m = N$ yields

$$\begin{aligned} e_N(t) = y(t) &- k^{\mathrm{b}}_0(t-1)r_0(t-1) - k^{\mathrm{b}}_1(t-1)r_1(t-1) \\ &- \cdots - k^{\mathrm{b}}_{N-1}(t-1)r_{N-1}(t-1). \end{aligned} \tag{14}$$

Thus the Nth order predictor of $y(t)$ is

$$y(t/N) = k^{\mathrm{b}}_0(t-1)r_0(t-1) + k^{\mathrm{b}}_1(t-1)r_1(t-1) + \cdots + k^{\mathrm{b}}_{N-1}(t-1)r_{N-1}(t-1). \tag{15}$$

Compare (15) to the equation for a transversal predictor given by

$$y(t/N) = a_1(t-1)y(t-1) + a_2(t-1)y(t-2) + \cdots + a_N(t-1)y(t-N)$$

and let $Y_t = (0, \ldots, y(0), y(1), \ldots, y(t))$ and $R_{m,t} = (0, \ldots, r_m(0), r_m(1), \ldots, r_m(t))$, the optimal predictor for the RLS transversal case can be thought of in geometrical terms as the projection of Y_t onto the regressor space spanned by $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-N}$. The backward error vectors $R_{0,t-1}, R_{1,t-1}, \ldots, R_{N-1,t-1}$ span exactly the same space with the difference that they are mutually orthogonal, and the optimal predictor for the RLS lattice case is again the projection of Y_t onto the above regressor space. Thus the predictors for the RLS lattice and transversal case are identical.

In general, the OBE estimates at every time step are not identical to the RLS estimates. Nevertheless, the approximate minimum mean square residual property of the ellipsoidal center

[20] justifies the imposition of the lattice structure (12) and (13) on the forward and backward errors of different orders for the OBE algorithm. Therefore, in principle, the OBE algorithm can be used to calculate the transversal filter coefficients and then a step down procedure described in [18, Section 4.2] can be used to calculate the optimal PARCOR coefficients. However, this method will involve an excessive amount of computation. Furthermore, many of the advantageous features of the lattice structure mentioned earlier will be lost since the transversal filter coefficients are being used to obtain the lattice coefficients. It is thus preferable to apply the OBE in such a way that, at every time step, the estimate of the PARCOR coefficients is obtained directly. This problem can be tackled by working in the space of PARCOR coefficients instead of the space of the transversal filter coefficients. More specifically, define the ellipsoid

$$E_0' = \left\{ \theta' : \theta'^{\mathrm{T}} P_0^{-1} \theta < 1/\Delta \right\}$$

and the convex polytope

$$S_t' = \left\{ \theta' : \left(y(t) - \theta'^{\mathrm{T}} \phi'(t) \right)^2 \leqslant \gamma_0^2 \right\},$$

where

$$\theta' = \left(k_0^{\mathrm{b}}, k_1^{\mathrm{b}}, \ldots, k_{N-1}^{\mathrm{b}} \right)^{\mathrm{T}}, \qquad \phi'(t) = \left(r_0(t-1), r_1(t-1), \ldots, r_{N-1}(t-1) \right)^{\mathrm{T}},$$

$$P_0 = I \quad \text{and} \quad \Delta \ll 1.$$

If the true PARCOR coefficients are defined to be the ones obtained by applying the step down procedure to the true AR parameters of the system, and if the true PARCOR coefficients have been used to recursively obtain $\phi'(t)$ from (12) and (13), then each one of the convex polytopes S_t', $t = 1, 2, \ldots, T$, will contain the true backward PARCOR coefficient vector θ'^*. This is because $\theta'^{*\mathrm{T}} \phi'(t) \equiv \theta^{*\mathrm{T}} \phi(t)$. The OBE algorithm can now be applied with the parameter vector θ set equal to the PARCOR coefficient vector θ' and the regressor vector $\phi(t)$ set equal to the vector of background errors $\phi'(t)$ thus yielding the time update for the backward PARCOR coefficients. However, it is clear from (13) that, in order to obtain the backward errors of different orders at time t, the forward PARCOR coefficients at time $t-1$ are required. The time update equations for the forward coefficients can be obtained as follows.

Iterating (13) yields

$$r_N(t) = y(t-N) - k_0^{\mathrm{f}}(t-N) e_0(t-N+1) - k_1^{\mathrm{f}}(t-N+1) e_1(t-N+2) \cdots - k_{N-1}^{\mathrm{f}}(t-1) e_{N-1}(t).$$

Since the backward errors are expected to be bounded, one can therefore define a convex polytope in the space of the forward PARCOR coefficients as

$$S''(t) = \left\{ \theta'' : \left(y(t-N) - \theta''^{\mathrm{T}} \phi''(t) \right)^2 \leqslant \gamma'^2_0 \right\},$$

where

$$\theta'' = \left(k_0^{\mathrm{f}}, k_1^{\mathrm{f}}, \ldots k_{N-1}^{\mathrm{f}} \right)^{\mathrm{T}} \quad \text{and} \quad \phi''(t) = \left(e_0(t-N+1), e_1(t-N+2), \ldots e_{N-1}(t) \right)^{\mathrm{T}}.$$

$\gamma_0'^2$ is set higher than γ_0^2 to ensure that $S''(t)$ contains the true forward PARCOR coefficient

vector. It is worth noting that the exact value of $\gamma_0'^2$ is not critical here as, according to our experience, the algorithm is relatively insensitive to the values of such bounds. The algorithm formulated by (7) with $\theta = \theta''$ and $\phi = \phi''$ can now be applied to obtain the time updates of the forward PARCOR coefficients. An important point to be noted here is that for the backward error recursions at time t, the estimates $k_0^f(t-1)$, $k_1^f(t-1), \ldots, k_{N-1}^f(t-1)$ are required. However, the OBE time update at time $t-1$ has made available the coefficients $k_0^f(t-N)$, $k_1^f(t-N+1), \ldots, k_{N-1}^f(t-1)$. The former set of coefficients thus has to be approximated by the latter set. This is a valid approximation for small N (as verified by simulation) because then $k_m(t-1)$ is approximately equal to $k_m(t-N+m)$.

For the stationary case, since the backward and forward coefficients are expected to be equal, the OBE needs to be applied only once to the forward error (i.e., to the backward coefficients). The algorithm complexity is thus the same as that of the direct implementation. In general, the computational complexity of the lattice implementation is twice that of the direct form because the OBE algorithm is applied two times at every iteration. However, the order of computation is still $O(N^2)$.

4. Extension to ARMA models

Autoregressive moving average (ARMA) models are described by difference equations of the form

$$y(t) = a_1 y(t-1) + \cdots + a_n y(t-n) + w(t) + c_1 w(t-1) + \cdots + c_r w(t-r) \tag{16}$$

where $y(t)$ is the output and $w(t)$ is an unobservable white noise sequence. This equation can be recast as

$$y(t) = \theta^{*T} \phi'(t) + w(t), \tag{17}$$

where

$$\theta^* = [a_1, a_2, \ldots, a_n, c_1, \ldots, c_r]^T \tag{18}$$

is the vector of true parameters, and

$$\phi'(t) = [y(t-1), y(t-2), \ldots, y(t-n), w(t-1), \ldots, w(t-r)]^T$$

is the true regressor vector. It is assumed that the noise is uniformly bounded in magnitude, i.e., there exists $\gamma_0 \geq 0$, such that

$$w^2(t) \leq \gamma_0^2 \quad \text{for all } t. \tag{19}$$

Since the values of $w(t)$ are unknown, the OBE algorithm, in its present form, cannot be used to estimate the parameters. However, if estimates of $w(t)$ are used in place of the actual values, as in the ELS algorithm, then the algorithm (7) can be used to construct a sequence of optimal bounding ellipsoids. A natural estimate of $w(t)$ is the a posteriori prediction error (also termed residual by some authors)

$$\epsilon(t) = y(t) - \theta^T(t)\phi(t), \tag{20}$$

where now

$$\phi(t) = [y(t-1), y(t-2), \ldots, y(t-n), \epsilon(t-1), \ldots, \epsilon(t-r)]^T. \tag{21}$$

The extended optimal bounding ellipsoid algorithm (EOBE) [13,21] thus consists of (7) and the same selective update strategy, with the true parameter vector and the regressor vector as defined in (18) and (21) respectively. The initial conditions (11) are modified to

$$P^{-1}(0) = M \cdot I, \quad \theta(0) = 0 \quad \text{and} \quad \sigma^2(0) \leqslant \gamma^2, \qquad \text{with } M \gg 1. \tag{22}$$

This choice of initial conditions still ensures that the initial ellipsoid E_0 will contain θ^* and makes the algorithm amenable to analysis. It also simplifies the formula for determining the updating gain factor. In particular, λ_t is always less than unity, hence there is no need to introduce an upper bound for the updating gain factor. Also note that γ^2 in (22) is different from γ_0^2 in (19).

Analysis of the EOBE algorithm. It is easy to see that, since estimates of $w(t)$ are used in the regressor vector, there is no guarantee that all the convex polytopes S_t, $t = 1, 2, \ldots$, will contain θ^*. However, it has been shown [21] that all the convex polytopes will contain θ^* if (i) E_0 contains θ^*, (ii) the true moving average coefficients satisfy a certain upper bound (analogous to the Strictly Positive Real (SPR) condition in the ELS algorithm), and (iii) the threshold γ^2 is chosen appropriately [21]. The conditions are of course only sufficient conditions, and the algorithm has been observed to perform well in several examples where the conditions (ii) and (iii) were violated.

Using this result, the following bounds on the prediction error and parameter estimation error can be obtained (see [13,21] for details).

(a)

$$\lim_{t_j \to \infty} \epsilon^2(t_j) \quad \text{exists,}$$

where $\{t_j\}$ is the subsequence of updating instants of the EOBE algorithm.

(b) Uniformly bounded a posteriori prediction errors:

$$\epsilon^2(t) \leqslant \gamma^2, \quad \text{for all time instants } t.$$

Furthermore, if a certain persistence of excitation condition holds, then for any finite k,

(c)

$$\lim_{t \to \infty} \| \theta(t) - \theta(t-k) \| = 0.$$

(e) Asymptotically bounded a priori prediction errors:

$$\delta^2(t) \to [0, \gamma^2].$$

(f) Asymptotically bounded parameter estimation error.

$$\| \theta(t) - \theta^* \|^2 \to \left[0, 2\gamma_0^2 \left(1 + \sum |c_i|\right)^2 / \alpha_4\right],$$

where γ_0^2 is as in (19) and α_4 is a positive constant.

The above results do not require the system (16) to be stable or the noise sequence $w(t)$ to be white. However our simulation experience has shown that the parameter estimates are usually not close to the true parameters if the noise is colored, but such is also the case for the ELS

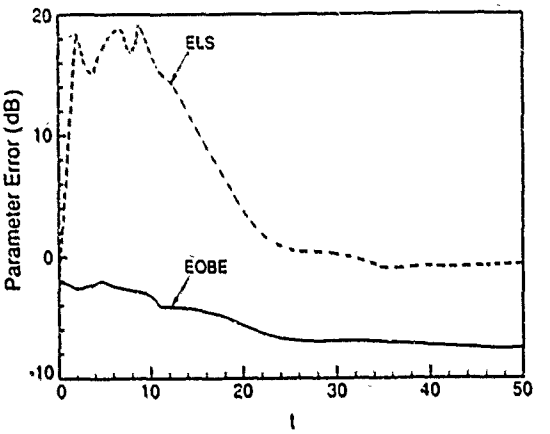


Fig. 2. Mean-squared parameter estimation errors of the EOBE and ELS algorithms (Example 2).

algorithm. The EOBE algorithm performs well when the noise sequence $w(t)$ is white. In particular, the transient performance of the algorithm for stable and unstable ARMA type systems with $w(t)$ white appears to be superior to that of the ELS algorithm. This observation is illustrated by the following.

Example 2. The following ARMA (3,3) model is considered

$$y(t) = -0.6\,y(t-1) + 0.2\,y(t-2) + 0.4\,y(t-3) + w(t) - 0.22\,w(t-1) + 0.17\,w(t-2) - 0.1\,w(t-3).$$

The white noise sequence $w(t)$ is uniformly distributed in $[-1, 1]$. Ten Monte Carlo runs of the OBE and ELS algorithms are performed with data records of 50 points each. The threshold $\gamma^2 = 25$. The parameter estimation error at each instant, $(\theta^* - \theta(t))^T(\theta^* - \theta(t))$ and the a priori prediction error are averaged over the ten runs and displayed in *Fig. 2* and *Fig. 3* respectively. The parameter estimates of the ELS algorithm tend to wander outside the stability region in the

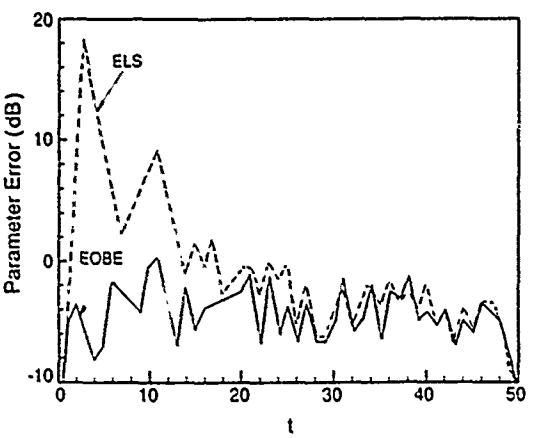


Fig. 3. Mean-squared prediction errors of the EOBE and ELS algorithms (Example 2).

transient stage, thus causing unacceptably high prediction error bursts. The inherent stability mechanism of the ELS algorithm, however, ensures that the estimates do return to the stability region. The transient estimation error of the EOBE algorithm, in contrast, is well behaved. This seems to provide a good incentive for employing EOBE, rather than ELS, when few data are available.

5. Conclusion

It has been shown that, on account of their low computational complexity and analytical tractability, the OBE algorithms can serve as alternatives to standard adaptive filtering algorithms in situations where the noise is unknown but bounded. As in the least-squares case, the OBE algorithm can be implemented in a lattice form and can thus acquire all the advantages of the lattice structure. The extension to the colored noise case is performed as in extended least squares, and sufficient conditions for "convergence" of the algorithm have been outlined. The transient performance of the algorithm, in terms of parameter estimation error and prediction error, has been observed to be superior to that of the ELS algorithm.

Acknowledgment

This work has been supported, in part, by the National Science Foundation under Grant MIP-87-11174 and, in part, by the Office of Naval Research under Contract N00014-89-J-1788.

References

[1] D.P. Bertsekas and I.B. Rhodes, Recursive state estimation for a set-membership description of uncertainty, *IEEE Trans. Automat. Control* AC-16 (1971) 117–123.

[2] E. Fogel and Y.-F Huang, On the value of information in system identification—bounded noise case, *Automatica* 18 (2) (1982) 229–238.

[3] J.P. Norton, Identification and application of bounded-parameter models, *Automatica* 23 (1987) 497–507.

[4] L. Ljung, *System Identification: Theory for the User* (Prentice-Hall, Englewood Cliffs, NJ, 1987).

[5] Y.-F. Huang, A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing, *IEEE Trans. Acoust. Speech Signal Process.* ASSP-34 (5) (1986) 1131–1134.

[6] S. Dasgupta and Y.-F. Huang, Asymptotically convergent modified recursive least-squares, *IEEE Trans. Inform. Theory* IT-33 (3) (1987) 383–392.

[7] Y.-F. Huang, *System identification using membership-set description of system uncertainties and a reduced-order optimal state estimator*, M.S. Thesis, Dept. of Electrical Engineering, Univ. of Notre Dame, Notre Dame, IN, 1980.

[8] E. Walter and H. Piet-Lahanier, Exact and recursive description of the feasible parameter set for bounded error models, in: *Proc. 26th IEEE Conference on Decision and Control*, Los Angeles (1987) 1921–1922.

[9] M. Milanese and G. Belforte, Estimation theory and uncertainty intervals, *IEEE Trans. Automat Control* AC-27 (2) (1982) 408–414.

[10] Y.-F. Huang and A.K. Rao, Lattice implementation of a novel recursive estimation algorithm, in: *Proc of the 24th Annual Allerton Conf. on Commun., Contr. and Comput.*, University of Illinois, Urbana-Champaign (1986) 115–124.

[11] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification* (MIT Press, Cambridge, MA, 1983)

[12] E. Walter and H. Piet-Lahanier, Estimation of parameter bounds from bounded-error data: a survey, in: *Proc. 12th IMACS World Congress*, Paris, France (1988) 467–472.
[13] A.K. Rao, *Membership-set parameter estimation via optimal bounding ellipsoids*, Ph. D. dissertation, Dept. of Electrical & Computer Engineering, Univ. of Notre Dame, Notre Dame IN, 1989.
[14] E. Walter and H. Piet-Lahanier, OMNE versus least-squares for estimating the parameters of a biological model from short records, in: *Proc. 12th IMACS World Congress*, Paris, France, (1988) 85–87.
[15] A.K. Rao and Y.-F. Huang, Analysis of finite precision effects on an OBE algorithm, in: *Proc. IEEE 1989 International Conference on Acoustics, Speech and Signal Processing*, Glasgow, Scotland (1989) 853–856.
[16] J.R. Deller, A 'Systolic array' formulation of the optimal bounding ellipsoid algorithm, *IEEE Trans. Acoust. Speech Signal Process.* ASSP-37 (9) (1989) 1432–1436.
[17] J.R. Deller, Set membership identification in digital signal processing, *IEEE ASSP Magazine* 6 (4) (1989) 4–22.
[18] M.L. Honig and D.L. Messerschmitt, *Adaptive Filters — Structures, Algorithms and Applications* (Kluwer Academic Publishers, New York, 1984).
[19] A.K. Rao, *Applications and extensions of a novel recursive estimation algorithm with selective updating*, M.S. Thesis, Dept. of Electrical & Computer Engineering, Univ. of Notre Dame, Notre Dame, IN, 1987.
[20] J.P. Norton, Identification of parameter bounds of ARMAX models from records with bounded noise, *Internat. J. Control* 45 (2) (1987) 375–390.
[21] 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 Signal Process.* ASSP-38 (3) (1990).

ANALYSIS OF FINITE PRECISION EFFECTS ON A RECURSIVE SET MEMBERSHIP PARAMETER ESTIMATION ALGORITHM[1]

Ashok K. Rao[2] and Yih-Fang Huang[3]

Abstract

Analysis of error propagation in an OBE algorithm is performed which shows that the errors in the estimates due to an initial perturbation are bounded. Simulation results demonstrate that the OBE algorithm can perform better than the conventional RLS in small word-length environments. The analysis presented in the paper could also be applied for the finite precision analysis of recursive weighted least-squares algorithms.

[1] This work has been supported, in part, by the National Science Foundation under Grant MIP-8711174 and, in part, by the Office of Naval Research under Contract N00014-89-J-1788. Paper is to appear in the *IEEE Transactions on Signal Processing*, Vol. SP-40, No. 12, December 1992.

[2] COMSAT Labs, Clarksburg, MD 20874

[3] Laboratory for Image and Signal Analysis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556

I INTRODUCTION

Set Membership Parameter Estimation (SMPE) algorithms [1-3] are a class of estimation algorithms which yield a set of feasible parameter vectors consistent with the observations, model structure and noise constraints. This is in contrast to least-squares type or stochastic-gradient-based algorithms which compute point estimates of model parameters.

The SMPE algorithms do not assume any knowledge of the distribution or any other statistical properties of the noise process. However it is assumed that the noise is bounded, either in magnitude or energy. The performance of SMPE algorithms is often superior to the least-squares algorithms for cases when the noise process does not satisfy the usual white and stationary assumptions [4] and when the sample size is small [5]. Furthermore, these algorithms yield 100% confidence regions for the parameters even for small sample sizes, in the case of batch algorithms, and at every time instant with recursive algorithms.

The behavior of least-squares and stochastic-gradient-based adaptive filtering algorithms in limited precision environments has attracted a lot of attention [6], [7]. However, in the case of SMPE algorithms, the issue of finite word-length effects has been largely ignored till recently. In [8], the potential numerical problems which can arise with the exact cone updating (ECU) algorithm are discussed and a robust modification is suggested. In this paper, finite precision effects on one of the Optimal Bounding Ellipsoid (OBE) algorithms are studied through analysis and simulations. The OBE algorithms obtain recursively, ellipsoidal outer bounds of the membership set of parameters of ARX models with bounded noise. The algorithms have the distinctive feature of a discerning update strategy.

A brief description of the OBE algorithm of [9] is given in Section II. A first order analysis of the error propagation in the OBE algorithm is then performed which shows that the error in the estimates at any time instant due to an initial perturbation is bounded. The finite precision effects are also analyzed from an alternate geometric point of view. Results of a fixed point type simulation of the algorithm are presented which show that the OBE algorithm yields consistently good estimates over a large range of word-lengths. In fact, the performance is superior to that of the RLS algorithm for small word-lengths.

II THE OBE ALGORITHM

The OBE algorithms [1],[9] estimate the coefficients of ARX processes described by

$$y(t) = \theta^{*T} \Phi(t) + v(t) \tag{2.1}$$

where

$$\Phi(t) = [y(t-1),...,y(t-n), u(t), u(t-1), ...,u(t-m)]^T \tag{2.2}$$

is the regressor vector consisting of past outputs $\{y(t)\}$ and present and past inputs $\{u(t)\}$, and θ^* is the true parameter vector, specifically $\theta^* = [a_1, ..., a_n, b_0, ..., b_m]^T$, [1,9]. The noise sequence $\{v(t)\}$ is assumed to be uniformly bounded with a known bound $\gamma \geq 0$, i.e.,

$$v^2(t) \leq \gamma^2 \qquad \text{for all } t \tag{2.3}$$

The OBE algorithms obtain, recursively, a "decreasing" sequence $\{E_t\}$ of optimal outer bounding ellipsoids in the n+m+1 dimensional parameter space. The ellipsoid E_t can be expressed as

$$E_t = \{\theta \in \mathbf{R}^{n+m+1} : [\theta - \theta(t)]^T P^{-1}(t) [\theta - \theta(t)] \leq \sigma^2(t) \} \tag{2.4}$$

where $P^{-1}(t)$ is a positive definite matrix and $\theta(t)$ is the center of the ellipsoid which can be taken to be a point estimate of the parameter vector. The factor $\sigma^2(t)$ is a positive time-varying scalar which along with P(t) determines the size of E_t. Time recursions for P(t), $\theta(t)$ and $\sigma^2(t)$ are given below, see [9] for a derivation of these equations.

$$P(t) = \frac{1}{1-\lambda_t}\left[P(t-1) - \frac{\lambda_t P(t-1)\Phi(t)\Phi^T(t)P(t-1)}{1-\lambda_t + \lambda_t \Phi^T(t)P(t-1)\Phi^T(t)}\right] \tag{2.5}$$

$$\sigma^2(t) = (1-\lambda_t)\,\sigma^2(t-1) + \lambda_t \gamma^2 - \frac{\lambda_t (1-\lambda_t)\,[y(t)-\Phi^T(t)\theta(t-1)]^2}{1-\lambda_t + \lambda_t \Phi^T(t)P(t-1)\Phi^T(t)} \tag{2.6}$$

$$\theta(t) = \theta(t-1) + \lambda_t P(t)\Phi(t)[y(t)-\Phi^T(t)\theta(t-1)] \tag{2.7}$$

The initial conditions are chosen to ensure that $\theta^* \in E_0$. A possible choice which improves the robustness of the algorithm to finite word-length effects is

$$P(0) = M\,\mathbf{I}, \text{ and } \sigma^2(0) = \gamma^2 \quad \text{where } M >> 1. \tag{2.8}$$

The optimal ellipsoid E_t is defined in terms of an optimal value of the updating factor $\lambda_t \in [0,\alpha)$ where $\alpha < 1$, is a user chosen upper bound on the updating factor. For the OBE algorithm of [9], the optimum value λ_t is determined by minimization of $\sigma^2(t)$ with respect to λ_t at every time instant. The minimization procedure results in a discerning update procedure. In particular,

$$\text{if } \sigma^2(t-1) + \delta^2(t) \leq \gamma^2 \qquad \lambda_t = 0 \text{ (no update)} \tag{2.9}$$

$$\text{else} \qquad \lambda_t = \min(\alpha, \varpi(t)), \qquad \text{with} \tag{2.10}$$

$$\varpi(t) = \begin{cases} (1-\beta(t))/2 & \text{if } G(t) = 1 \\ \dfrac{1}{1-G(t)}\left[1-\left\{\dfrac{G(t)}{1+\beta(t)\,(G(t)-1)}\right\}^{\frac{1}{2}}\right] & \text{if } G(t) \neq 1 \end{cases} \tag{2.11}$$

where

$$\delta(t) = y(t) - \theta^T(t-1)\Phi(t) \tag{2.12}$$

$$G(t) = \Phi^T(t)P(t-1)\Phi(t) \tag{2.13}$$

and

$$\beta(t) = (\gamma^2 - \sigma^2(t-1)\,)/\,\delta^2(t) \tag{2.14}$$

The above recursive relations (2.5)-(2.7), and the updating factor formula (2.9)-(2.14) form the OBE estimation algorithm.

III ERROR PROPAGATION

The error propagation properties of the OBE algorithm are analyzed here by focusing on the propagation of a single error in $\theta(t)$ and $P(t)$ to future instants. Assume that at time instant t_0 there is a perturbation in the estimates due to round-off error, yielding $\theta'(t_0) = \theta(t_0) + \Delta\theta(t_0)$ and $P'(t_0) = P(t_0)+\Delta P(t_0)$, where the primed quantities are the perturbed ones. We investigate in this section the effect of these errors on the estimates $\theta'(t)$ and $P'(t)$ at $t > t_0$, assuming that the computations are performed with infinite precision. Similar studies have been performed by Ljung and Ljung [10] in their investigation of the error propagation properties of RLS algorithms. Though the update equations of the OBE algorithm are similar to those of the RLS algorithm, the presence of the updating factor as a discontinuous function of the estimates complicates the analysis. Employing a first order perturbation analysis, an upper bound on the error in the estimates due to finite precision computations can be obtained as described below.

Theorem 1. If the following assumptions hold:

(i) The matrix $P(t)$ is well conditioned, i.e. there exist positive η_1 and η_2 such that

$$0 < \eta_1 \leq \lambda_{min}[P(t)] \quad \text{and} \quad \lambda_{max}[P(t)] \leq \eta_2 \quad \text{for all } t \geq 0 \tag{3.1}$$

where $\lambda_{min}[.]$ and $\lambda_{max}[.]$ refer to minimum and maximum eigenvalues respectively.

(ii) The ARX process is stable and has bounded inputs, thereby implying the existence of a positive κ such that

$$\Phi^T(t)\Phi(t) \leq \kappa \quad \text{for all } t \geq 0 \tag{3.2}$$

(iii) The unperturbed algorithm yields bounded prediction errors, i.e., there exists an $h > 0$, s.t.

$$| \delta(t) | \leq h \qquad \text{for all } t \geq 0 \tag{3.3}$$

(iv) There exists an integer M such that if the unperturbed algorithm has M updates in an interval of time, then the perturbed version updates at least once in that interval,

(v) At the updating instants of the perturbed algorithm, a lower bound ρ is set for the updating factor λ_t', where ρ is a suitably small positive number.

Then the error between the perturbed and unperturbed quantities at the updating instants $\{t_k\}$ of the perturbed algorithm is bounded as

$$\|\Delta P(t_k)\| \leq \left(\frac{\eta_2}{\eta_1}\right)^2 (1-\rho)^{\frac{k}{M}-1} \|\Delta P(0)\| + \eta_2^2 \left(\kappa+\frac{1}{\eta_1}\right) \max_{1\leq u\leq k} \Delta\lambda_{t_u} M \frac{1-(1-\rho)^{\lfloor k/M \rfloor}}{\rho} \tag{3.4}$$

$$\|\Delta\theta(t_k)\| \leq (1-\rho)^{k/M-1} \|\Delta\theta(t_0)\| + \eta_2 h \kappa^{1/2} \max_{1\leq j\leq k} |\Delta\lambda_{t_j}| \frac{M}{\rho} [1-(1-\rho)^{\lfloor k/M \rfloor}] + h \kappa^{1/2} \frac{\eta_2}{\eta_1} \max_{1\leq j\leq k} \|\Delta P(t_j)\| \tag{3.5}$$

where η_1 and η_2 are as in (3.1); $\lfloor x \rfloor$ is used to denote the largest integer less than x and $\|.\|$ is used for both the euclidean vector norm and the compatible matrix norm.

Proof: See [11].

Remarks

(1) The first term in (3.4) and (3.5) reveals an exponentially decaying effect of the initial perturbation. The second term depends on the error introduced by the initial perturbation in the calculation of the updating factor. The additional error term in (3.5) is due to the errors in P(t).

(2) Assumptions (i) and (iii) have been shown to hold in [9] if the system input u(t) and the noise v(t) satisfy certain persistence of excitation type of conditions.

(3) Assumption (v) is a technical device required to ensure that the homogeneous parts of (3.4) and (3.5) are exponentially stable. If $\rho \leq 0.001$, then in practice the values of λ_t at the updating instants will usually be larger than ρ.

(4) Note that the analysis of error propagation has ignored the effect of round-off errors in computations. However, since the homogeneous parts of (3.4) and (3.5) are exponentially

stable, the errors at any time instant due to round-off errors created at previous time instants would be bounded [10].

IV EFFECTS ON THE BOUNDING ELLIPSOID

In this section, the effect of round-off errors (in one iteration) on the resulting bounding ellipsoid is studied. More specifically, we ask the following question: If $\theta^* \in E_{t-1}$, can errors in the computation of E_t (i.e., computation of $\theta(t)$, $P(t)$ and $\sigma^2(t)$) cause $\theta^* \notin E_t$.

Define $\tilde{\theta}(t) = \theta(t) - \theta^*$. Then from (2.7)

$$\tilde{\theta}(t) = \tilde{\theta}(t-1) + \lambda_t P(t)\Phi(t)\delta(t) + \Delta_1 \tag{4.1}$$

where Δ_1 is the round-off error. Similarly from (2.5a)

$$P^{-1}(t) = (1-\lambda_t)\, P^{-1}(t-1) + \lambda_t \Phi(t)\Phi^T(t) + \Delta_2 \tag{4.2}$$

and from (2.6)

$$\sigma^2(t) = (1-\lambda_t)\,\sigma^2(t-1) + \lambda_t \gamma^2 - \frac{\lambda_t (1-\lambda_t)\,\delta^2(t)}{1-\lambda_t + \lambda_t \Phi^T(t)P(t-1)\Phi^T(t)} + \Delta_3 \tag{4.3}$$

Define

$$A_t = \tilde{\theta}(t-1) + \lambda_t P(t)\Phi(t)\delta(t) \tag{4.4}$$

and

$$B_t = (1-\lambda_t)\, P^{-1}(t-1) + \lambda_t \Phi(t)\Phi^T(t) \tag{4.5}$$

Then, after neglecting second and higher order terms in Δ_1 and Δ_2, it can be shown that

$$V(t) = A_t^T B_t A_t + A_t^T \Delta_2 A_t + \Delta_1^T B_t A_t + A_t^T B_t \Delta_1 \tag{4.6}$$

where

$$V(t) = \tilde{\theta}(t)\, P^{-1}(t)\, \tilde{\theta}(t) \tag{4.7}$$

Expanding $A_t^T B_t A_t$ and using (4.3) yields

$$\begin{aligned} V(t) - \sigma^2(t) = {} & (1-\lambda_t)\,[V(t-1) - \sigma^2(t-1)] + \lambda_t [v^2(t) - \gamma^2] \\ & + 2\Delta_1^T B_t A_t + A_t^T \Delta_2 A_t + \Delta_3 \end{aligned} \tag{4.8}$$

From the definition of E_t it is clear that $\theta^* \in E_t$ if and only if $V(t) \le \sigma^2(t)$. Thus if the errors Δ_1, Δ_2, and Δ_3 are large enough, it is possible that $\theta^* \notin E_t$. A sufficient condition for $\theta^* \in E_t$ is

$$| 2\Delta_1^T B_t A_t + A_t^T \Delta_2 A_t + \Delta_3 | \le \lambda_t [\gamma^2 - v^2(t)] \tag{4.9}$$

If $\lambda_t = 0$ then since no update occurs $\theta^* \in E_t$ automatically. The condition (4.9) shows that if the errors due to finite word-length computations are small enough then $\theta^* \in E_t$. Furthermore, by setting γ^2 higher than the actual bound on the noise, the robustness of the algorithm with respect to finite precision effects can be increased at the expense of increasing the size of the bounding ellipsoids.

V SIMULATION STUDIES

A fixed point implementation of the OBE algorithm was simulated by assigning a fixed number of bits (*ibit*) to represent the fractional part of the algorithmic variables. By varying *ibit* a fairly accurate portrayal of the behavior of the algorithm in a real-world restricted word-length environment can be obtained. A similar scheme has been used in [12] to characterize the performance of the RLS algorithm. The noise sequence {v(t)} and the input sequence{u(t)}are generated by a pseudo-random number generator with a uniform distribution in [-1.0,1.0]. The upper bound γ^2 is set equal to 1.0. A value of $\alpha = 0.1$ was used since it yielded a satisfactory convergence rate and inhibited overflows in the update equation for P(t). The parameter estimates are obtained by applying the OBE, RLS and EWLS (RLS with weighting factor λ =0.99) to 1000 point data sequences. For the OBE algorithm, the centers of the optimal bounding ellipsoids are taken to be the estimates. Ten runs of the algorithms are performed on the same model but with different noise sequences. The number of bits used for the fractional part, *ibit*, is varied from 16 down to 6 bits and the average of the parameter error $\|\theta(1000)-\theta^*\|^2$ is computed for each value of *ibit*.

Example 5.1 (Fig. 5.1) An ARX(2,3) process

$$y(t) = 1.6y(t-1)-0.83\, y(t-2)+0.14\, u(t) +u(t-1) +0.16\, u(t-2) +v(t)$$

The average tap error of the OBE algorithm appears constant as *ibit* varies from 16 to 8 bits. The P matrix became negative definite for *ibit* = 6. The RLS and EWLS algorithms do not work well for *ibit* $\leq$ 10. In fact P became indefinite for *ibit* $\leq$ 14, in the EWLS case.

Example 5.2 (Fig. 5.2) An ARX(10,10) process

The OBE algorithm worked well for *ibit* $\geq$ 12. However for smaller values, P became indefinite and overflows occurred. For the RLS case, P became indefinite for *ibit* $\leq$ 16. In order to study the performance of the OBE algorithm at smaller word-lengths, a UDU' factorization of the P matrix was performed. The OBE update equations are identical to the update equations of the weighted RLS algorithm with weight $\alpha_t = \lambda_t$, and forgetting factor $\lambda(t) = (1-\lambda_t)$ and hence the UDU' form of the OBE can be easily developed [13, pg. 334]. The UDU' form of the OBE algorithm is then compared to the UDU' form of the RLS algorithm.

The simulation results show that for larger word-lengths, the performance of the RLS algorithm is superior. For smaller values of *ibit*, however, the average parameter estimation error is about the same for both the OBE and the RLS algorithms.

Discussions

Example 5.2 shows that the performances of the UDU' versions of OBE and RLS algorithms are comparable at smaller word-lengths. The superior performance of the straightforward implementation of the OBE algorithm, as compared to the RLS or EWLS algorithms at smaller word-lengths is therefore primarily due to the superior numerical properties of the recursion for the matrix P(t). The update equation for the RLS algorithm with a forgetting factor λ is

$$P(t) = \left[I - \frac{P(t-1)\Phi(t)\Phi^T(t)}{\lambda + \Phi^T(t)P(t-1)\Phi(t)} \right] \frac{P(t-1)}{\lambda} \tag{5.1}$$

The corresponding equation for the OBE algorithm can be rewritten as

$$P(t) = \left[I - \frac{P(t-1)\Phi(t)\Phi^T(t)}{\frac{1-\lambda_t}{\lambda_t} + \Phi^T(t)P(t-1)\Phi(t)} \right] \frac{P(t-1)}{1-\lambda_t} \tag{5.2}$$

Since $1-\lambda_t$ plays the same role in the OBE algorithm as does λ in the RLS algorithm, the only difference between (5.1) and (5.2) is that the factor $(1-\lambda_t)/\lambda_t$ appears in the denominator of the term within braces in (5.2) as opposed to the corresponding term λ in (5.1). The degradation of performance occurs primarily because the term within braces becomes indefinite on account of round-off errors. Since λ_t is usually much smaller than unity, the term which is being subtracted from the identity matrix in (5.2) is much smaller than the one in (5.1). Thus P(t) in the RLS algorithm has a greater tendency to become indefinite than the P(t) in the OBE algorithm. This observation has been confirmed by examining the eigenvalues of P(t), for runs in which the RLS algorithm performed poorly.

VI CONCLUSION

The analysis of error propagation in the OBE algorithm has shown that the algorithm is stable with respect to small computational errors. As in the RLS case, the robustness of the algorithm is due to the presence of an updating gain/forgetting factor. Stability of the algorithm has also been viewed from an alternate geometric approach. The analysis shows that the bounding ellipsoids are valid bounds for the membership sets as long as computational errors are not too large and that the robustness of the algorithm can be increased by increasing the value of the noise bound. Simulation results show that the OBE algorithm is indeed stable for moderate word-lengths and that the mean parameter estimation error is relatively constant over a wide range of word-lengths. In fact, it was observed that the performance of the OBE algorithm is superior to that of the RLS algorithm for small word-lengths.

REFERENCES

[1] E. Fogel and Y.F. Huang, " On the value of information in system identification-bounded noise case,"*Automatica*, vol. 18, No. 2, pp. 229-238, March 1982.

[2] J. R.Deller, "Set membership identification in digital signal processing," *IEEE Acoust., Speech, and Signal Processing Magazine*, vol. 6, No. 4, pp. 4-20, Oct. 1989.

[3] J. P. Norton, "Identification and Application of Bounded-parameter models", *Automatica* Vol. 23, No.4, pp. 497-507, 1987.

[4] A. K. Rao and Y. F. Huang, " Recent developments in optimal bounding ellipsoidal parameter estimation," *Mathematics and Computers in Simulation*, vol. 32, Nos. 5 & 6, Dec. 1990, pp. 515-526.

[5] M. Milanese and G. Belforte, " Estimation theory and uncertainty intervals," *IEEE Trans. Automatic Control*, vol. AC-27, No. 2, pp. 408-414, April 1982.

[6] J. M. Cioffi, "Limited-precision effects in adaptive filtering,"*IEEE Trans. Circuits and Systems*, vol. CAS-34, No. 7, pp. 821-833, July 1987.

[7] J. M. Cioffi and T. Kailath, " Fast RLS transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, No. 2, pp. 304- 337, April 1984.

[8] H. Piet-Lahanier and E. Walter, " Practical implementation of an exact and recursive algorithm for characterizing likelihood sets," *Proc. 12th IMACS World Congress*, Paris, France, July 1988, pp. 481-483.

[9] S. Dasgupta and Y.F. Huang, " Asymptotically convergent modified recursive least-squares,"*IEEE Trans. Information Theory* , vol. IT-33, No.3, pp. 383 - 392, May 1987.

[10] S. Ljung and L. Ljung, "Error propagation properties of recursive least-squares adaptation algorithms," *Automatica*, vol. 21, pp. 157-167, 1985.

[11] A. Rao, " Membership-Set Parameter Estimation via Optimal Bounding Ellipsoids," Ph.D. dissertation, Department of Electrical Engineering, University of Notre Dame, Notre Dame IN, January 1990.

[12] S. H. Ardalan and S. T. Alexander, " Fixed point roundoff error analysis of the exponentially windowed RLS algorithm for time varying systems," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, No. 6, pp. 770-783, June 1987.

[13] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press,Cambridge, Mass. 1983.

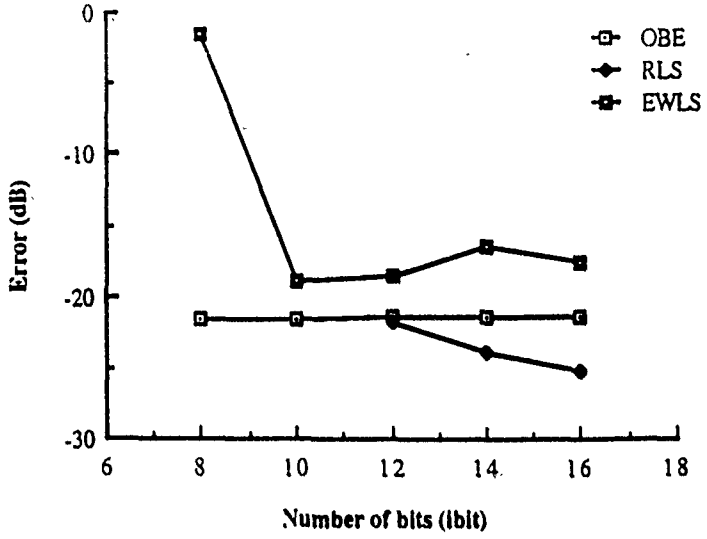


Figure 5.1 Average parameter estimation error for the OBE and RLS algorithms for an ARX(2,3) process

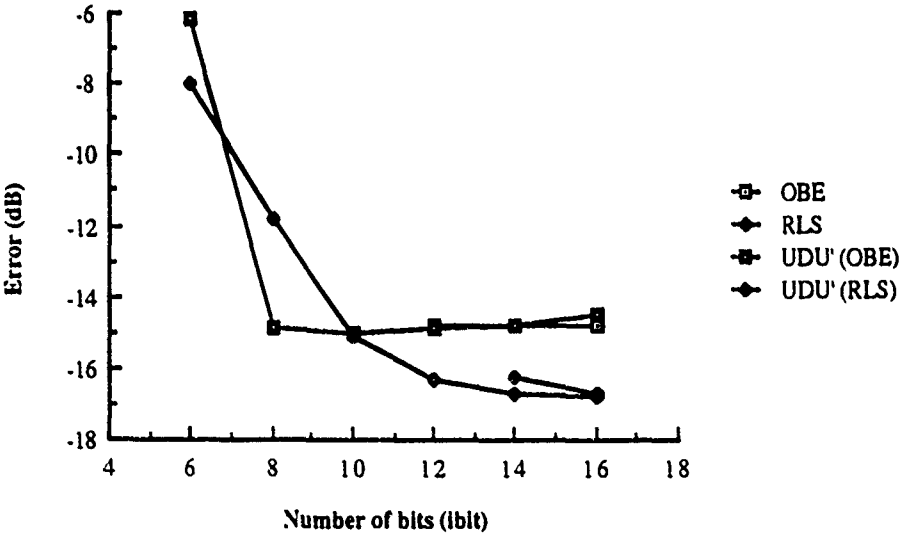


Figure 5.2 Average parameter estimation error for the OBE and RLS algorithms for an ARX(10,10) process

Publication [4]

TRACKING CHARACTERISTICS OF AN OBE PARAMETER ESTIMATION ALGORITHM*

Ashok K. Rao[1] and Yih-Fang Huang[2]

Abstract

Recently there seems to have been a resurgence of interest in recursive parameter bounding algorithms. These algorithms are applicable when the noise is bounded and the bound is known to the user. One of the advantages of such algorithms is that 100% confidence regions (which are optimal in some sense) for the parameter estimates can be obtained at every time instant, rather than asymptotically as in the least-squares type algorithms. Another advantage is that these recursive algorithms have the inherent capability of implementing discerning updates, particularly that of allowing no updates of parameter estimates in the recursion. This paper investigates tracking properties of one such algorithm, referred to as the DHOBE algorithm. Conditions which ensure the existence of these 100% confidence regions in the face of small model parameter variations are derived. For larger parameter variations, it is shown that the existence of the 100% confidence regions is guaranteed asymptotically. A modification is also proposed here to enable the algorithm to track large variations in model parameters. Simulation results show that in general, the modified algorithm has tracking performance comparable, and in some cases superior, to the exponentially weighted recursive least-squares algorithm.

* This work has been supported in part by the National Science Foundation under Grant MIP 87-11174, and in part by the Office of Naval Research under Contract N00014-89-J-1788. Paper is to appear in the *IEEE Transactions on Signal Processing*.

[1] COMSAT Laboratories, 22300 Comsat Drive, Clarksburg, MD 20871

[2] Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556

I. INTRODUCTION

Performance analysis of adaptive filtering algorithms is usually done by assuming that the unknown system being modeled is time-invariant. However, in practice, adaptive filters are often used in time varying environments. It is thus important to investigate the performance of these algorithms, allowing the system model parameters to vary with time. A considerable amount of attention has been paid to this problem in the adaptive filtering literature, with analysis of varying amounts of rigor being performed mainly for the LMS and RLS algorithms, see, e.g., [1-5].

This paper investigates tracking properties of a recursive estimation algorithm, referred to hereafter as the DHOBE (Dasgupta-Huang Optimal Bounding Ellipsoid) algorithm [6]. This algorithm belongs to a class of bounded-error estimation algorithms termed *Set-Membership Parameter Estimation* (SMPE) algorithms [7],[8]. The membership set is a set of parameter estimates which are compatible with the model of the underlying process, the assumptions on noise, and the observation data. At the first glance the DHOBE algorithm appears to be very similar to the recursive least-squares (RLS) algorithm. However, in contrast to the RLS algorithm which obtains an optimal solution (in the sense of minimum mean-square estimation error) to the underlying problem, the DHOBE algorithm is developed by using a set-theoretic framework, namely, the notion of *optimal bounding ellipsoids*. This causes the algorithm to behave quite differently from the RLS algorithm in many ways. In addition, the algorithm incorporates a data dependent forgetting factor which results in a discerning update strategy.

In case of time-varying systems, it is important to ensure that the time varying true parameters $\{\theta^*(t)\}$ are contained in the bounding ellipsoids $\{E_t\}$ of the DHOBE algorithm. In this paper, such conditions will be derived. It will also be shown that if a jump in the true parameter vector $\theta^*(t)$ causes it to fall outside the bounding ellipsoid, then provided that the jump is not too large the bounding ellipsoids will move towards $\theta^*(t)$ and eventually enclose $\theta^*(t)$ again. A rescue scheme is proposed which will guarantee the

existence of bounding ellipsoids in the face of large parameter variations. Some techniques for applying different parameter bounding algorithms to time varying systems have been reported by Norton and Mo in [9]. One of the techniques suggested for the OBE type algorithms is to use a fixed scaling factor to inflate the bounding ellipsoid with every new data point. Another technique which can be used if prior knowledge of the parameter increments is available is to vector sum the bounding ellipsoid with the set describing the parameter variation [9]. If the extent of parameter variation is unknown, as is often the case, the first technique will have to use a large scaling factor to cope with possibly large parameter variations and consequently the parameter bounds will be loose. In contrast, the rescue procedure described in this paper can automatically detect and accurately compensate for large parameter jumps.

Simulation results are presented to show that the DHOBE algorithm is able to track slow and abrupt variations in the parameters. The tracking performance, in terms of parameter estimation error, is comparable to the RLS algorithm with a forgetting factor. Abrupt changes in the parameter can in some cases be tracked better by the DHOBE algorithm than by the RLS algorithm.

II. THE DHOBE ALGORITHM

One of the seminal works in the estimation of parameter bounds is that of Fogel and Huang [10]. The algorithm of [10] recursively obtains ellipsoidal outer bounds to the membership set. The model structure considered is the following ARX model:

$$y(t) = \theta^{*T}\Phi(t) + v(t) \tag{2.1}$$

where

$$\theta^* = [a_1\ a_2\ \ldots\ a_n\ b_0\ b_1\ \ldots\ b_m]^T$$

is the true parameter vector and

$$\Phi(t) = [y(t-1)\ y(t-2)\ \ldots\ y(t-n)\ u(t)\ u(t-1)\ \ldots\ u(t-m)]^T$$

is the measurable regressor vector. The noise $v(t)$ is assumed to be uniformly bounded in magnitude with a known bound γ, i.e.,

$$|v(t)| \le \gamma \tag{2.2}$$

Assume that at time instant t-1, the exact membership set is outer bounded by the ellipsoid E_{t-1} described by

$$E_{t-1} = \{\theta \in \mathbf{R}^N : [\theta - \theta(t-1)]^T P^{-1}(t-1) [\theta - \theta(t-1)] \le \sigma^2(t-1)\} \tag{2.3}$$

where N=n+m+1, $P^{-1}(t-1)$ is a positive definite matrix, and $\theta(t-1)$ is the center of the ellipsoid. At time instant t, the observation y(t) yields a set S_t which is a degenerate ellipsoid in the parameter space, namely

$$S_t = \{\theta \in \mathbf{R}^N : [y(t) - \theta^T\Phi(t)]^2 \le \gamma^2\} \tag{2.4}$$

From (2.1) and (2.2) it is clear that S_t contains the true parameter vector. An ellipsoid E_t which contains the intersection of E_{t-1} and S_t is then given by [10]

$$E_t = \{\theta \in \mathbf{R}^N : (1-\lambda_t)[\theta - \theta(t-1)]^T P^{-1}(t-1) [\theta - \theta(t-1)] + \lambda_t [y(t) - \theta^T\Phi(t)]^2 \} \le (1-\lambda_t)\sigma^2(t-1) + \lambda_t \gamma^2\} \tag{2.5}$$

where λ_t is a positive time-varying updating gain. Note that $(1-\lambda_t)$ can be regarded as a forgetting factor. The formation of the bounding ellipsoid E_t which contains the intersection of an ellipsoid E_{t-1} and the set S_t is illustrated by means of a two-dimensional example in Figure 1. By performing some algebraic manipulations on (2.5), an expression for E_t can be obtained as

$$E_t = \{\theta \in \mathbf{R}^N : [\theta - \theta(t)]^T P^{-1}(t) [\theta - \theta(t)] \le \sigma^2(t)\} \tag{2.6}$$

where

$$P^{-1}(t) = (1-\lambda_t)P^{-1}(t-1) + \lambda_t \Phi(t)\Phi^T(t) \tag{2.7}$$

$$\sigma^2(t) = (1-\lambda_t)\sigma^2(t-1) + \lambda_t \gamma^2 - \frac{\lambda_t (1-\lambda_t) [y(t)-\Phi^T(t)\theta(t-1)]^2}{1-\lambda_t + \lambda_t \Phi^T(t)P(t-1)\Phi(t)} \tag{2.8}$$

$$\theta(t) = \theta(t-1) + \lambda_t P(t)\Phi(t)[y(t)-\Phi^T(t)\theta(t-1)] \tag{2.9}$$

Using the matrix inversion lemma in (2.7) yields

$$P(t) = \frac{1}{1-\lambda_t}[P(t-1) - \frac{\lambda_t P(t-1)\Phi(t)\Phi^T(t)P(t-1)}{1-\lambda_t + \lambda_t \Phi^T(t)P(t-1)\Phi(t)}] \tag{2.10}$$

Equations (2.6) - (2.9) characterize the update of the bounding ellipsoids. The center $\theta(t)$ of the bounding ellipsoid E_t can be taken to be a point estimate of the parameter vector. Note that different values of λ_t yield different bounding ellipsoids [10]. To ensure

convergence, λ_t need be chosen to optimize in some sense the bounding ellipsoids and, clearly, different optimization criteria would lead to different OBE algorithms.

In the DHOBE algorithm, the updating gain λ_t is chosen to minimize $\sigma^2(t)$ at every instant t. This has the effect of usually decreasing the size of the ellipsoid from iteration to iteration, though there is no guarantee that the size will be minimized. This choice of λ_t has yielded good results experimentally and in addition has simplified the convergence and tracking analysis of the algorithm. The minimization procedure yields the following updating criterion [6]

$$\text{If} \quad \sigma^2(t-1) + \delta^2(t) \le \gamma^2 \quad \text{then} \quad \lambda_t = 0 \ \text{(i.e., no update)} \tag{2.11}$$

where $\delta(t)$ is the *a priori* prediction error, namely,

$$\delta(t) = y(t) - \Phi^T(t)\theta(t-1) \tag{2.12}$$

Otherwise if $\sigma^2(t-1) + \delta^2(t) > \gamma^2$, then the optimum value of λ_t is non-zero and can be calculated according to

$$\lambda_t = \min(\alpha, \nu_t)$$

where

$$\nu_t = \begin{cases} \alpha & \text{if } \delta^2(t) = 0 \qquad (2.13.a) \\ \dfrac{1-\beta(t)}{2} & \text{if } G(t) = 1 \qquad (2.13.b) \\ \dfrac{1}{1-G(t)}\left[1 - \sqrt{\dfrac{G(t)}{1+\beta(t)[G(t)-1]}}\right] & \text{if } 1+\beta(t)[G(t)-1] > 0 \qquad (2.13.c) \\ \alpha & \text{if } 1+\beta(t)[G(t)-1] \le 0 \qquad (2.13.d) \end{cases}$$

and α is a user chosen upper bound on λ_t satisfying

$$0 < \alpha < 1 \tag{2.14}$$

and

$$G(t) = \Phi^T(t)P(t-1)\Phi(t) \tag{2.15}$$

and

$$\beta(t) = \frac{\gamma^2 - \sigma^2(t-1)}{\delta^2(t)} \tag{2.16}$$

The initial conditions are chosen to ensure that $\theta^* \in E_0$. A possible choice is

$$P(0) = I, \ \theta(t) = 0 \text{ and } \sigma^2(0) = 1/\varepsilon^2 \quad \text{where } \varepsilon << 1.$$

The above equations, (2.8-2.16) define the recursions of the DHOBE algorithm. In [6], some convergence type properties such as convergence of the parameter estimates to a ball and boundedness of the prediction error have been shown for time-invariant systems. In

[11] and [12], an extension of this algorithm was developed for ARMA parameter estimation and similar convergence properties have been shown to hold.

III. ANALYSIS OF TRACKING CHARACTERISTICS

As mentioned earlier, tracking in the context of OBE algorithms for parameter estimation will mean ensuring that the time varying true parameter vector is contained in the bounding ellipsoid. The theorems below present conditions under which parameter tracking can be accomplished.

Theorem 1. A sufficient condition for $\theta^*(t) \in E_t$ is

$$(\theta^*(t)-\theta(t-1))^T P^{-1}(t-1)(\theta^*(t)-\theta(t-1)) \le \sigma^2(t-1) \tag{3.1}$$

Proof: If $\theta^*(t) \in E_{t-1}$ then since $\theta^*(t) \in S_t$ and $E_t \supseteq E_{t-1} \cap S_t$, it follows that $\theta^*(t) \in E_t$. And from (2.3), $\theta^*(t) \in E_{t-1}$ is equivalent to (3.1).

Theorem 2. At any time instant t, the true parameter $\theta^*(t) \in E_t$ if and only if

$$(\theta^*(t)-\theta(t-1))^T P^{-1}(t-1)(\theta^*(t)-\theta(t-1)) \le \sigma^2(t-1)+\frac{\lambda_t}{1-\lambda_t}\left(\gamma^2 - v^2(t)\right) \tag{3.2}$$

where v(t) is the noise term in (2.1).

Proof: Subtracting $\theta^*(t)$ from both sides of (2.9) yields

$$\theta(t)-\theta^*(t) = \theta(t-1)-\theta^*(t)+\lambda_t P(t)\Phi(t)\delta(t) \tag{3.3}$$

Define the following quadratic function in $\theta^*(t)$

$$V(t) = [\theta(t)-\theta^*(t)]^T P^{-1}(t)[\theta(t)-\theta^*(t)]$$

Using (2.7) and (3.3) it is straightforward though tedious to show that

$$\begin{aligned} V(t) = {} & (1-\lambda_t)[\theta(t-1)-\theta^*(t)]^T P^{-1}(t-1)[\theta(t-1)-\theta^*(t)] \\ & + \lambda_t v^2(t) - \frac{\lambda_t(1-\lambda_t)\delta^2(t)}{(1-\lambda_t)+\lambda_t G(t)} \end{aligned} \tag{3.4}$$

Using (2.8) in (3.4) yields

$$\begin{aligned} V(t)-\sigma^2(t) = {} & (1-\lambda_t)[\theta(t-1)-\theta^*(t)]^T P^{-1}(t-1)[\theta(t-1)-\theta^*(t)] \\ & +\lambda_t(v^2(t)-\gamma^2)-(1-\lambda_t)\sigma^2(t-1) \end{aligned} \tag{3.5}$$

Since $\theta^*(t) \in E_t$ if and only if $V(t) \le \sigma^2(t)$, thus (3.2) is obtained. ∇∇∇

It is easy to see from Theorem 1 that if the true parameter $\theta^*(t)$ is constant for all t, then the bounding ellipsoids obtained by the DHOBE algorithm encloses $\theta^*(t)$ at all time instants. This is a property that all well devised set-membership estimation algorithms should have when applied to estimation of time-invariant parameters. If, on the other hand, $\theta^*(t)$ is time-varying, and if at some time instant t_k, $\theta^*(t)$ is found to be out of the bounding ellipsoid E_t, it must not have been included in E_{t-1}. Theorem 2 then demarcates the region in which $\theta^*(t)$ can migrate without loss of tracking. This region is shown in Figure 2 for a two-dimensional case. This theorem also shows that by choosing γ^2 to be larger than the actual bound, say, γ'^2 on $v^2(t)$, it is possible to increase the tracking capability of the algorithm. The next theorem gives an upper bound on the maximum variation in the parameters for which tracking is guaranteed.

Theorem 3. If $\theta^*(t-1) \in E_{t-1}$, and $\lambda_t \neq 0$, then $\theta^*(t) \in E_t$ if

$$\|\Delta(t)\| \leq \frac{1}{\sqrt{\lambda_{min}[P^{-1}(t-1)]}} \left\{ \left[\frac{\lambda_t}{1-\lambda_t} \frac{\lambda_{min}[P^{-1}(t-1)]}{\lambda_{max}[P^{-1}(t-1)]} [\gamma^2 - \gamma'^2(t)] + \sigma^2(t-1) \right]^{\frac{1}{2}} - \sqrt{\sigma^2(t-1)} \right\} \tag{3.6}$$

where

$$\Delta(t) = \theta^*(t) - \theta^*(t-1) \tag{3.7}$$

and λ_{min} and λ_{max} denote, respectively, minimum and maximum eigenvalues, and $\|.\|$ denotes the usual Euclidean norm. The quantity γ'^2 is the actual bound on $v^2(t)$ and the threshold γ^2 that is needed for evaluating the optimal updating gain via (2.11) and (2.16) is chosen to be larger than γ'^2.

Proof: It is straightforward to show that

$$\begin{aligned} &[\theta(t-1) - \theta^*(t)]^T P^{-1}(t-1)[\theta(t-1) - \theta^*(t)] \\ &\quad = V(t-1) + \Delta^T(t) P^{-1}(t-1)\Delta(t) - 2\Delta^T(t) P^{-1}(t-1)\tilde{\theta}(t-1) \end{aligned} \tag{3.8}$$

where $V(t)$ has been defined previously and

$$\tilde{\theta}(t-1) = \theta(t-1) - \theta^*(t-1)$$

Substituting (3.8) into (3.5) and using the fact that $v^2(t) \leq \gamma'^2$ yield

$$V(t)=\sigma^2(t)\le(1-\lambda_t)[V(t-1)+\sigma^2(t-1)]+\lambda_t(\gamma^2-\gamma'^2)$$
$$+(1-\lambda_t)[\Delta^T(t)P^{-1}(t-1)\Delta(t)-2\Delta^T(t)P^{-1}(t-1)\tilde{\theta}(t-1)] \quad (3.9)$$

Since $\theta^*(t-1) \in E_{t-1}$, therefore $V(t-1) \le \sigma^2(t-1)$ and as a sufficient condition for $\theta^*(t) \in E_t$ is

$$\Delta^T(t)P^{-1}(t-1)\Delta(t)-2\Delta^T(t)P^{-1}(t-1)\tilde{\theta}(t-1)$$
$$\le\frac{\lambda_t}{1-\lambda_t}(\gamma^2-\gamma'^2) \quad (3.10)$$

i.e., $\theta^*(t) \in E_t$ if

$$\lambda_{max}[P^{-1}(t-1)]\|\Delta(t)\|^2+2\|\Delta(t)\|\,\|\tilde{\theta}(t-1)\|\lambda_{max}[P^{-1}(t-1)]\le\frac{\lambda_t}{1-\lambda_t}(\gamma^2-\gamma'^2) \quad (3.11)$$

Since $V(t-1) \le \sigma^2(t-1)$, therefore

$$\|\tilde{\theta}(t-1)\|^2\le\frac{\sigma^2(t-1)}{\lambda_{min}[P^{-1}(t-1)]} \quad (3.12)$$

Substituting (3.12) in (3.11) gives a sufficient condition for $\theta^*(t) \in E_t$ as

$$\lambda_{max}[P^{-1}(t-1)]\|\Delta(t)\|^2+2\|\Delta(t)\|\sqrt{\sigma^2(t-1)}\frac{\lambda_{max}[P^{-1}(t-1)]}{\sqrt{\lambda_{min}[P^{-1}(t-1)]}}$$
$$\le\frac{\lambda_t}{1-\lambda_t}(\gamma^2-\gamma'^2) \quad (3.13)$$

Solving this quadratic inequality then yields (3.6). ∇∇∇

It can be seen from (3.6) that if $\lambda_t = 0$, then the difference between γ^2 and γ'^2 can not be exploited to increase the tracking capability of the algorithm. In this case, $\theta^*(t) \in E_t$ if and only if $\theta^*(t) \in E_{t-1}$. Thus if $\theta^*(t)$ jumps out of E_{t-1}, and no updates are performed at future time instants $t+i$, then $\theta^*(t+i) \notin E_{t+1} = E_{t-1}$, and the parameter may never be tracked. However, it can be argued that an update will be performed in a finite interval of time. This is shown heuristically, by examining the expression for the magnitude of the prediction error

$$|\delta(t)| = |\,[\theta^*(t)-\theta(t-1)]^T\Phi(t) + v(t)\,|$$

Assume that no updates are performed for a large interval of time say, from time instant t to time instant $t + N_1$. From (2.11) it then follows that

$$[\theta^*(t+i)-\theta(t-1)]^T\Phi(t+i) + v(t+i) \le [\gamma^2 - \sigma^2(t-1)]^{1/2} \quad \forall\ i = 0,1,....N_1.$$

If the input and noise sequences are sufficiently rich, then the regressor vector $\Phi(t)$ will span the parameter space in all directions and so $[\theta^*(t+i)-\theta(t-1)]^T\Phi(t+i)$ will not be arbitrarily small for all $i \in [0, N_1]$. If $|v(t+i)|$ is close to its true upper bound γ for some i in the same interval, and if $\{v(t)\}$ is sufficiently uncorrelated with the input $\{u(t)\}$, then the above inequality will be violated and an update will be performed. It is also clear that to ensure that an update is performed eventually (i.e., violation of the above inequality), the threshold γ^2 should not be chosen much larger than γ^2.

If the parameter variation is such that (3.2) is violated then $\theta^*(t) \notin E_t$. The next theorem shows that if $\theta^*(t)$ remains fixed after it jumps out of E_t, and if the jump is not large enough to cause the subsequent ellipsoids E_{t+i} to vanish, for $i \ge 0$, then the DHOBE algorithm guarantees that the true parameter will be tracked (enclosed) in finite time.

Theorem 4. Assume that the parameter variation at time instant t causes $\theta^*(t) \notin E_t$. Assume further that:

(1) After this variation , the parameter remains constant (i.e., the jump parameter case).

(2) $\sigma^2(t+i) > 0$, for all $i \ge 0$.

(3) The algorithm does not stop updating.

(4) A lower bound ρ is imposed on λ_t at all updating instants.

Then there exists an $N_1 > 0$, which depends on the amount of parameter variation and the actual and user set noise bounds, such that $\theta^*(t) \in E_{t+N_1}$.

Proof: Since $\theta^*(t) \notin E_t$, define

$$\eta = [\theta(t)-\theta^*(t)]P^{-1}(t)[\theta(t)-\theta^*(t)]-\sigma^2(t) > 0 \qquad (3.14)$$

Assumption (1) will imply that $\Delta(t+N_1) = \Delta(t+1) = 0$ for arbitrary positive N_1. Substituting in (3.9), and iterating from $t+N_1$ to $t+1$ yields

$$V(t+N_1)-\sigma^2(t+N_1) = \eta\prod_{i=t+1}^{t+N_1}(1-\lambda_i) + \sum_{i=t+1}^{t+N_1} q_{i,t+N_1}[\gamma'^2(t)-\gamma^2] \qquad (3.15)$$

where $q_{i,t}$ is defined as

$$q_{i,t} = \begin{cases} \lambda_i \prod_{j=i+1}^{t} (1-\lambda_j) & \text{if } i < t \\ \lambda_t & \text{if } i = t \end{cases}$$

Assumption (3) will ensure that some of the λ_{t+i}, $i \geq 0$, will be non-zero. This ensures that the first term on the right hand side of (3.15) will tend to zero. Since the second term on the right hand side of (3.15) is negative, the difference $V(t+N_1) - \sigma^2(t+N_1)$ will tend to zero as N_1 increases. Thus there exists an N_1 such that

$$V(t+N_1) - \sigma^2(t+N_1) \leq 0 \tag{3.16}$$

Thereby ensuring that $\theta^*(t) \in E_{t+N_1}$. ∇∇∇

IV. A RESCUE PROCEDURE

In many cases when the parameter jump is large, or if the ellipsoid has shrunk to a very small size, the intersection of E_{t-1} and S_t can be void. This situation is illustrated in Fig. 3. In such cases, $\sigma^2(t)$ will become negative, thus indicating that a bounding ellipsoid could not be constructed. To circumvent such a failure of the algorithm, a rescue procedure is proposed. If at any time instant *t*, $\sigma^2(t)$ becomes negative, then $\sigma^2(t-1)$ is increased by an appropriate amount, thereby increasing the size of E_{t-1} so that the intersection of S_t and this enlarged E_{t-1} will no longer be void. As such, an ellipsoid E_t will be constructed. Alternatively, γ^2 could be increased to permit a non-null intersection. However, the former procedure is preferable because it causes $\theta(t)$ to migrate towards $\theta^*(t)$, thereby reducing the parameter estimation error. The rescue procedure is similar to the covariance resetting technique used in RLS algorithms to cope with time varying systems. However, in the RLS case, a jump in the parameters has to be detected by some other means before the covariance matrix can be reset whereas for the DHOBE algorithm, $\sigma^2(t)$ becoming negative is an automatic indicator of a jump. The amount of increase in $\sigma^2(t-1)$ required to make $\sigma^2(t)$ positive in such a case is now calculated.

Recall that the optimal updating gain λ_t is the one which minimizes $\sigma^2(t)$. The minimum occurs either at a stationary point of $\sigma^2(t)$ or at one of the boundaries $\lambda_t = 0$ and

$\lambda_t = \alpha$. Since it is assumed that a failure occurs when $\sigma^2(t-1) > 0$ and $\sigma^2(t) \le 0$, therefore an update has to occur at t and so $\lambda_t \neq 0$. The case that the minimum occurs at a stationary point which is strictly inside the interval $[0,\alpha]$ and the case that the minimum occurs at $\lambda_t = \alpha$ are considered separately.

Case 1. $\left.\dfrac{d\sigma^2(t)}{d\lambda_t}\right|_{\lambda_t = \nu_t} = 0$ and $0 < \nu_t < \alpha$

From (2.13) it is clear that this case occurs if and only if $1+\beta(t)[G(t)-1] > 0$ and $\nu_t < \alpha$. Setting the derivative of $\sigma^2(t)$ in (2.8) to zero yields

$$\gamma^2 - \sigma^2(t-1) - \frac{1-\lambda_t}{1-\lambda_t+\lambda_t G(t)}\,\delta^2(t) + \frac{\lambda_t G(t)}{(1-\lambda_t+\lambda_t G(t))^2}\,\delta^2(t) = 0$$

Substituting $\sigma^2(t-1)$ from above into (2.8) yields

$$\sigma^2(t) + \frac{(1-\lambda_t)^2}{(1-\lambda_t+\lambda_t G(t))^2}\,\delta^2(t) = \gamma^2 \qquad (4.1)$$

Thus $\sigma^2(t)$ is negative if and only if

$$|\delta(t)| > \frac{1-\lambda_t+\lambda_t G(t)}{1-\lambda_t}\gamma \qquad (4.2)$$

On substituting for λ_t from (2.13b) and (2.13c), (4.2) can be expressed, respectively, as

$$|\delta(t)| > \frac{G(t)-1}{\sqrt{G(t)[1+\beta(t)(G(t)-1)]}-1}\gamma \qquad \text{if } G(t) \neq 1$$

and (4.3)

$$|\delta(t)| > \frac{2\gamma}{1+\beta(t)} \qquad \text{if } G(t) = 1$$

Using the definition of $\beta(t)$ from (2.16) in (4.3) and manipulating terms yields a necessary and sufficient condition for $\sigma^2(t)$ to be negative in terms of $\sigma^2(t-1)$

$$\sigma^2(t-1) < \frac{1}{G(t)-1}\left[\delta^2(t) + \gamma^2[G(t)-1] - \frac{[\gamma[G(t)-1]+|\delta(t)|]^2}{G(t)}\right] = K_1 \quad \text{if } G(t) \neq 1$$

and

$$\sigma^2(t-1) < \delta^2(t) + \gamma^2 - 2\gamma|\delta(t)| = K_1 \qquad \text{if } G(t) = 1$$

Note that the last inequality was obtained because $\nu_t = (1-\beta(t))/2 < 1$, hence $1+\beta(t) > 0$. Thus if the calculated value of $\sigma^2(t)$ is negative, the rescue procedure will replace $\sigma^2(t-1)$ by $K_1 + \zeta$, where ζ is a positive constant, thereby increasing the size of E_{t-1}. The optimum

updating gain will then be recalculated and the resulting value will be used to calculate $\sigma^2(t)$, $\theta(t)$ and $P(t)$. Our simulation studies have shown that using a value of $\zeta = 1$ yields satisfactory results.

Case 2. $\underline{\lambda_t = \alpha}$

In this case, from (2.8), $\sigma^2(t)$ is negative if and only if

$$\delta^2(t) \geq [1-\alpha+\alpha G(t)]\left[\frac{\sigma^2(t-1)}{\alpha}+\frac{\gamma^2}{1-\alpha}\right]$$

Thus $\sigma^2(t)$ is negative if and only if

$$\sigma^2(t-1) < \alpha\left[\frac{\delta^2(t)}{1-\alpha+\alpha G(t)}-\frac{\gamma^2}{1-\alpha}\right] = K_2$$

In this case $\sigma^2(t-1)$ would be replaced by $K_2 + \zeta$ and the value of the updating gain would be recalculated and used to calculate $\sigma^2(t)$, $\theta(t)$ and $P(t)$.

V. SIMULATION EXAMPLES

The tracking properties of the DHOBE algorithm are studied for an ARX(1,1) model

$$y(t) = ay(t-1) + bu(t) + v(t)$$

The nominal values for the parameters were $a = -0.5$ and $b = 1.0$. The noise sequence$\{v(t)\}$ and the input sequence$\{u(t)\}$ were both generated by a pseudo-random number generator with a uniform distribution in [-1,1]. This corresponds to a signal-to-noise ratio (SNR) of 0 dB. For the DHOBE algorithm, we chose $\alpha = 0.2$, $\gamma^2 = 1.0$, and $\sigma^2(0) = 100$. In all the examples shown here, the parameter estimates are taken to be the centers of the optimal bounding ellipsoids. The parameters were varied as follows:

Case 1. Slow variation in the parameter vector

The parameters a and b were varied by 1% for every 10 samples, starting from the first sample, and the output data $\{y(t)\}$ were generated for $t = 1,2,...1000$. It was then observed that the bounding ellipsoids created by the DHOBE algorithm contain the true parameter at all time instants. The final parameter estimation error was 7.0×10^{-3}. The parameter estimates, i.e., the centers of the OBE, are plotted against the true parameters in Fig. 4.

From the figure it is clear that the DHOBE algorithm tracks quite well slow time variations in the parameters.

Case 2. Slow variation in the parameter vector from $t = 500$

The parameters a and b were varied by 1% for every 10 samples, starting from the 500^{th} sample. The final parameter estimation error was 3.0×10^{-3}. All the bounding ellipsoids were seen to contain the true parameter. The parameter estimates are plotted against the true parameters in Fig. 5. The figure shows that the algorithm can track slow time variations in the parameters even after it has "converged".

Case 3. Jump in the MA parameter at $t = 500$

The parameter b was changed by 100% at the 500^{th} sample, and a was kept constant at its nominal value at all times. Several runs of the DHOBE algorithm were performed with different input and noise sequences. It was observed that the true parameter vector was out of the bounding ellipsoid at t=500 and would be recaptured by the bounding ellipsoid after some number of samples (usually less than 50) thus verifying the claims made in Theorem 4. It was also observed that the jump causes the resulting bounding ellipsoids to have smaller sizes. Intuitively, a jump at time t causes the set S_i, $i \geq t$, to have a smaller intersection with E_{i-1} and so the ellipsoid which bounds the intersection is also smaller. In one particular run, the parameter was recaptured at t = 530 and the final parameter estimation error at t = 1000 was 1.3×10^{-4}. The parameter estimates (the centers of the bounding ellipsoids) are plotted against the true parameters in Fig. 6. Figure 7 shows the parameter estimates obtained for this run by applying the RLS algorithm with a forgetting factor $\lambda(t) = 0.9$ and $\lambda(t) = 0.99$. Observe that the RLS parameter estimates are extremely jumpy when $\lambda(t) = 0.9$, probably because the forgetting factor is not large enough to average out the noise. Figure 8 shows the estimates when the variable forgetting factor proposed by Fortescue and Kershenbaum [13] is incorporated into the RLS algorithm. This variable forgetting factor, $\lambda(t)$, is a function of the prediction error and is given by

$$\lambda(t) = 1 - \alpha \frac{\delta^2(t)}{1 + G(t)}$$

A value of $\alpha' = 0.01$ was used because it yields steady state tracking error of about the same magnitude as does the DHOBE algorithm. From these figures, it is evident that the DHOBE algorithm can track jumps in the parameters at least as well as the exponentially weighted RLS algorithm.

The effect of varying γ^2 was also studied. A value of $\gamma^2 = 2$ was taken. In this case, the true parameter did not jump out of the bounding ellipsoid at $t = 500$. The parameter estimates are identical to those in Fig. 6. But the ellipsoids are larger, as expected.

For a different run, i.e., with a different input and noise sequence, the jump at $t = 500$, caused $\sigma^2(t)$ to become negative. The rescue procedure was then used and yielded remarkable results. The true parameter was captured immediately at $t = 501$. The final parameter estimation error was 2.4×10^{-4}. Figure 9 shows that the parameters are tracked extremely rapidly in this case.

Tracking Performance in Gaussian Noise

It is well known that least-squares algorithms are optimal in the constant parameter case for Gaussian distributed noise. It is thus interesting to compare the tracking abilities of the DHOBE and RLS algorithms in Gaussian noise. The same ARX model was used with the noise sequence v(t) now being generated as zero-mean white Gaussian noise with variance 0.25, which corresponds to an SNR of 1.25dB. To satisfy the bounded noise assumption, v(t) was truncated to the range [-1,1], resulting in a slightly larger SNR. The parameter b was changed by 100% at the 500th sample, and a was kept constant at its nominal value at all times. Several runs of the DHOBE algorithm were performed with different noise sequences. As in the uniform noise case, it was found that in a few runs, the rescue procedure was activated, consequently causing extremely rapid acquisition of the parameter. In most of the runs, the true parameter was acquired by the bounding ellipsoid without requiring rescue. The acquisition usually happened in less than twenty samples

after the change occurred. Figure 10 compares the tracking performance of the RLS algorithm (with $\lambda(t) = 0.9$ and $\lambda(t) = 0.99$) to the DHOBE algorithm for a run in which the rescue procedure was **not** activated. The curves shown are plots of estimates of parameter b by both algorithms. It is seen that RLS with $\lambda(t) = 0.9$ seems to track a little faster than the DHOBE algorithm. However the steady state RLS estimates are extremely jerky. The tracking performance of RLS with $\lambda(t) = 0.99$ is definitely inferior to that of the DHOBE algorithm, however its steady state performance prior to the jump is superior. Another point of note is that the DHOBE estimates become much less jerky after the jump on account of the decrease in the size of the ellipsoids.

VI. CONCLUSION

The tracking properties of a recursive set-membership parameter estimation algorithm viz. the DHOBE algorithm have been investigated. Some sufficient and other necessary conditions which ensure parameter tracking have been derived. A modification of the DHOBE algorithm is proposed to improve its tracking capability for larger parameter variations. Simulation results show that the tracking performance of the DHOBE algorithm is comparable to that of the exponentially weighted RLS algorithm. In some cases of large parameter jumps, the automatic activation of a rescue procedure causes the parameters to be tracked extremely rapidly.

REFERENCES

[1] B. Widrow *et al.* "Stationary and non stationary learning characteristics of the LMS adaptive filter." *Proc. IEEE* , vol. 64, pp. 1151-1162, 1976.

[2] A. Benveniste and G. Ruget, "A measure of the tracking capability of recursive stochastic algorithms with constant gains," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 639-649, 1982.

[3] E. Eleftheriou and D.D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. ASSP-34, No. 5, pp. 1097-1109, Oct. 1986.

[4] B.D. Rao and R. Peng, "Tracking characteristics of the constrained IIR adaptive notch filter," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. ASSP-36, No.9, pp. 1466-1479, Sept. 1988.

[5] S. Gunnarsson and L. Ljung, "Frequency domain tracking characteristics of adaptive algorithms," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. ASSP-37, No. 7, pp.1072-1089, July 1989.

[6] S. Dasgupta and Y.F. Huang, "Asymptotically convergent modified recursive least-squares...," *IEEE Trans. Inform. Theory* , vol. IT-33, No.3, pp. 383-392, May 1987.

[7] J. Deller, "Set Membership Identification and Signal Processing", *IEEE ASSP Magazine*, vol. 6, No. 4, pp. 4-20, October, 1989.

[8] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: a survey," *Mathematics and Computers in Simulation*, vol. 32, No. 5&6, pp. 449-465, December 1990.

[9] J. P. Norton and S. H. Mo, "Parameter Bounding for time-varying systems," *Mathematics and Computers in Simulation*, vol. 32, No. 5&6, pp. 527-534, December 1990.

[10] E. Fogel and Y.F. Huang, "On the value of information in system identification - bounded noise case," *Automatica* , vol. 18, No. 2, pp. 229-238, March 1982.

[11] Ashok K. Rao, "Membership-Set Parameter Estimation via Optimal Bounding Ellipsoids". Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Notre Dame, Notre Dame, IN, January 1990.

[12] 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 Processing*, vol. ASSP-38, No.3, pp. 447-457, March 1990.

[13] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

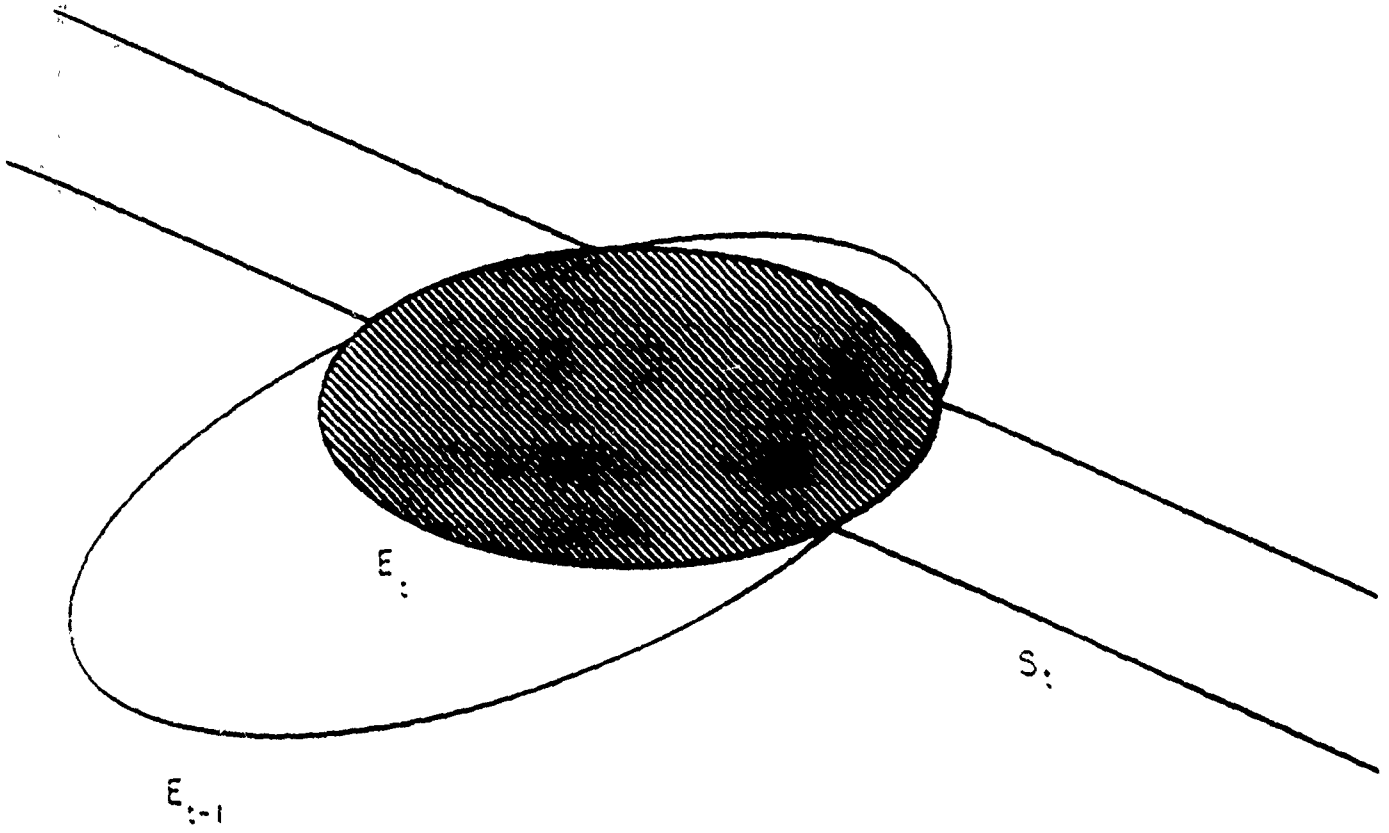
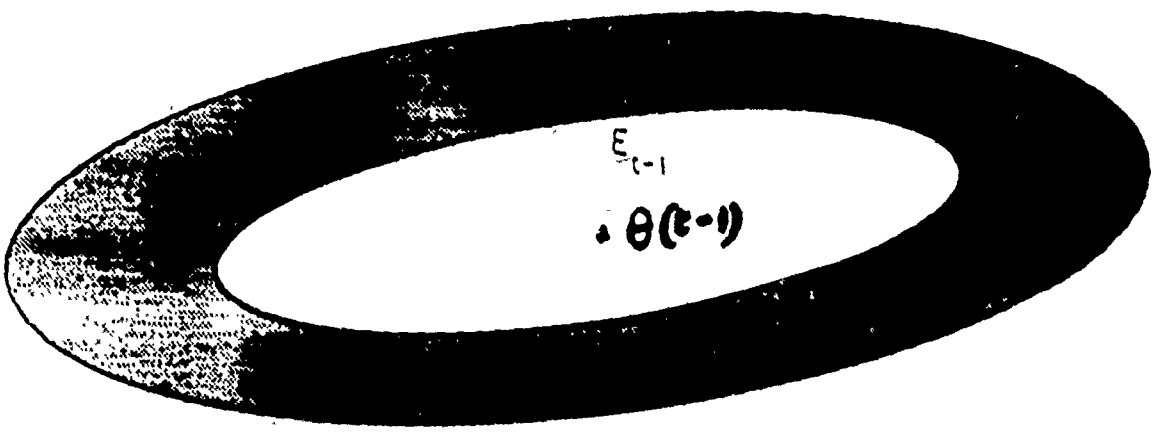


Figure 1. Formation of the bounding ellipsoid E_i



D(t) Permissible domain of migration of $\theta^*(t)$

Figure 2. Region outside E_{t-1} to which $\theta^*(t)$ can belong without loss of tracking.

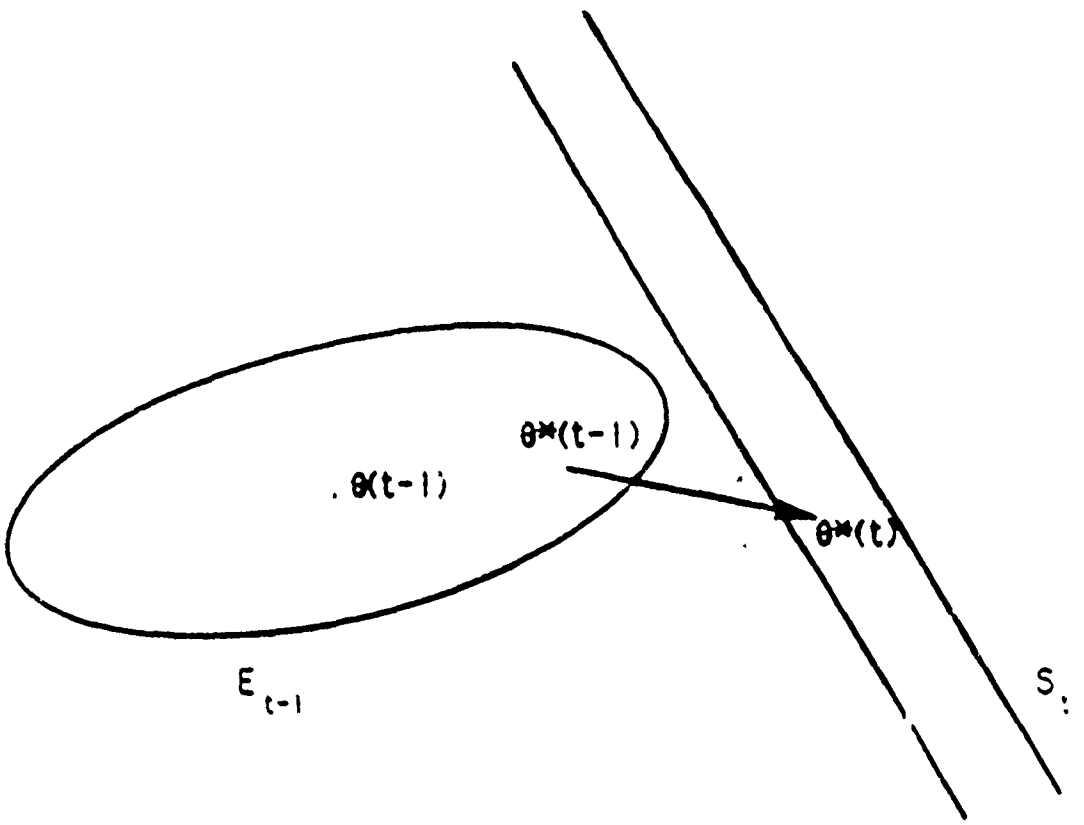


Figure 3. A case in which a jump in the parameter causes the intersection of E_{t-1} and S_t to be void.

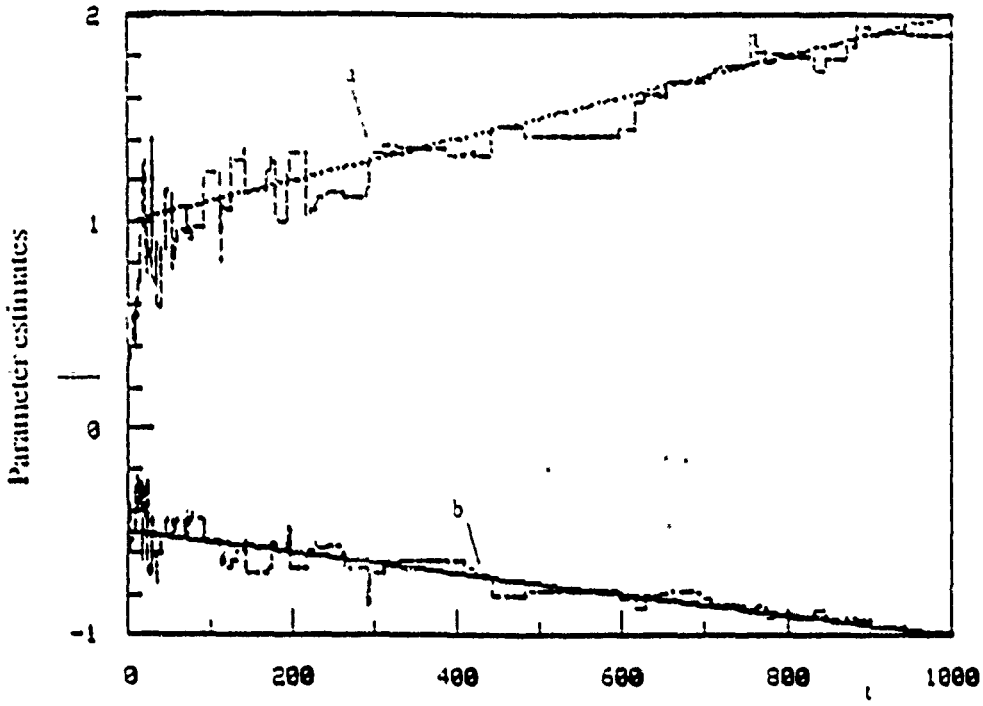


Figure 4. DHOBE parameter estimates for the case of slow variation in the true parameter from $t = 1$

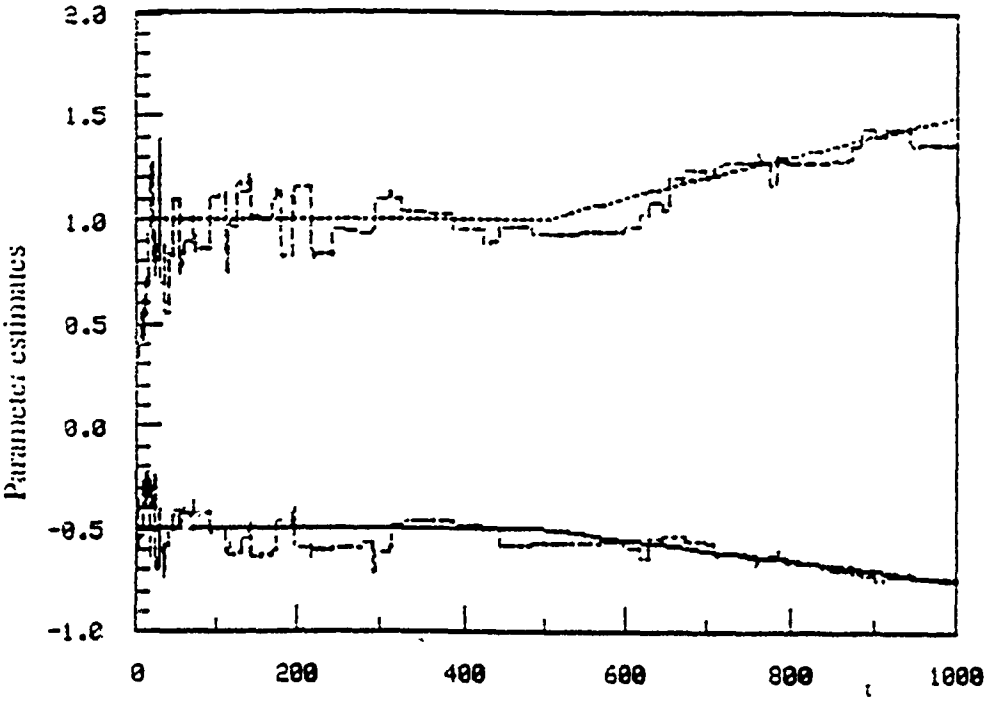


Figure 5. DHOBE parameter estimates for the case of slow variation in the true parameter from $t = 500$

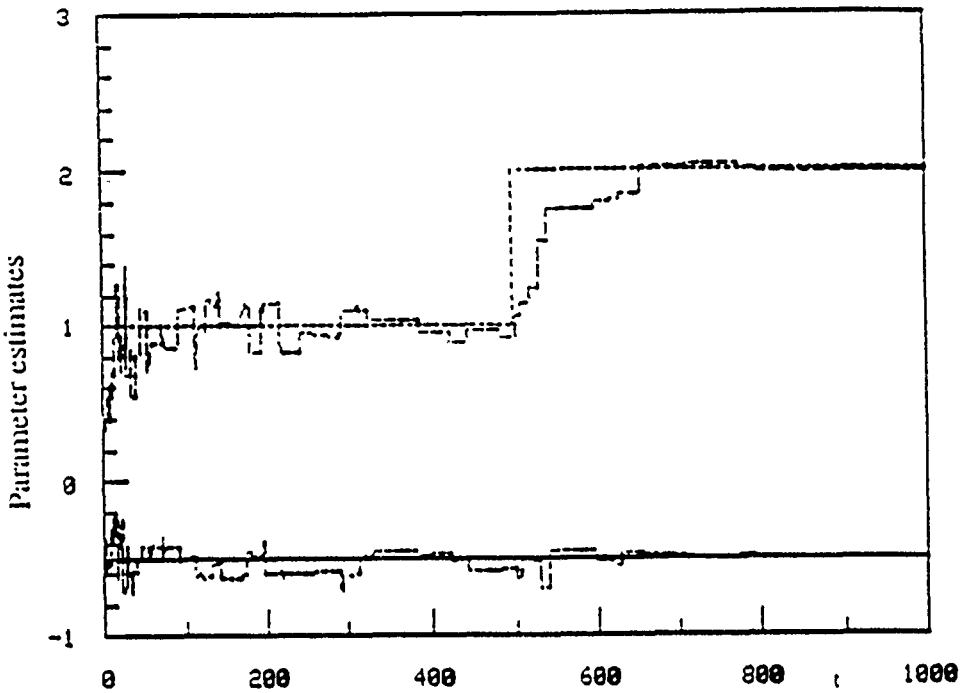


Figure 6. DHOBE parameter estimates for the case of a jump in the MA parameter at $t = 500$

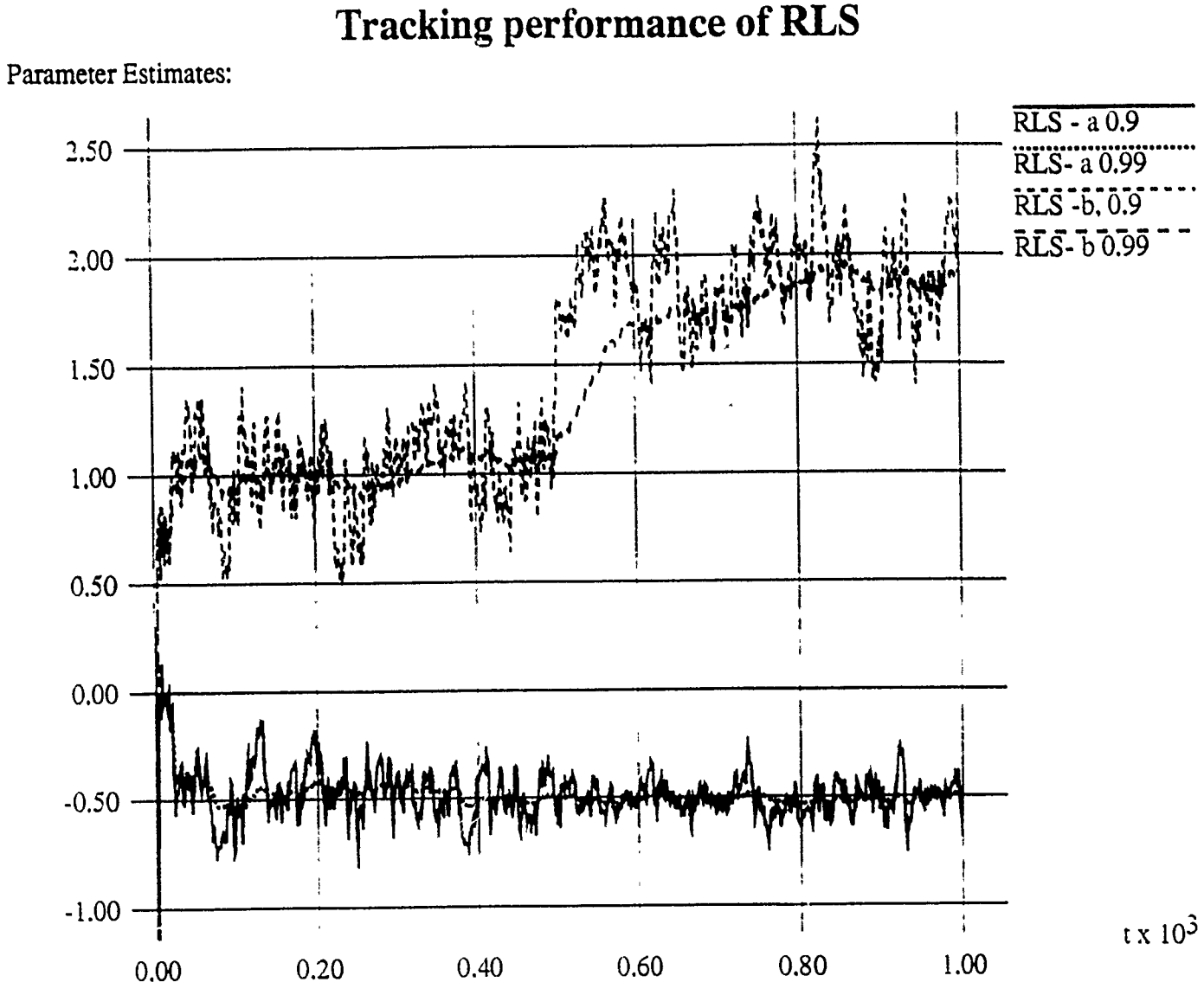


Figure 7. RLS (with $\lambda(t) = 0.9$ and $\lambda(t) = 0.99$) parameter estimates for the jump parameter case

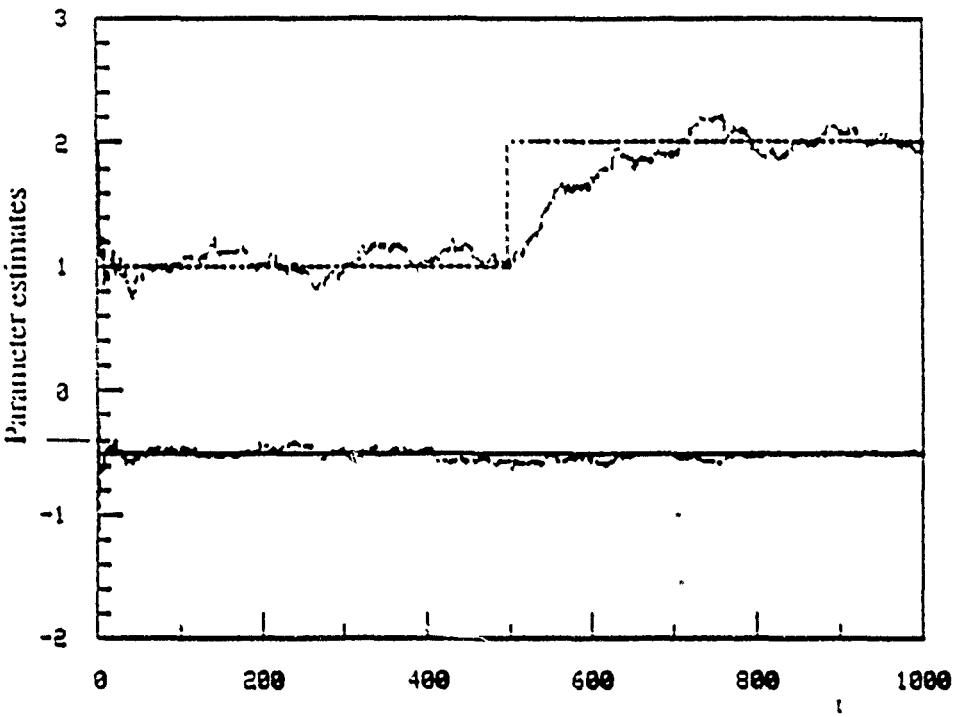


Figure 8. RLS (with variable forgetting factor) parameter estimates for the jump parameter case

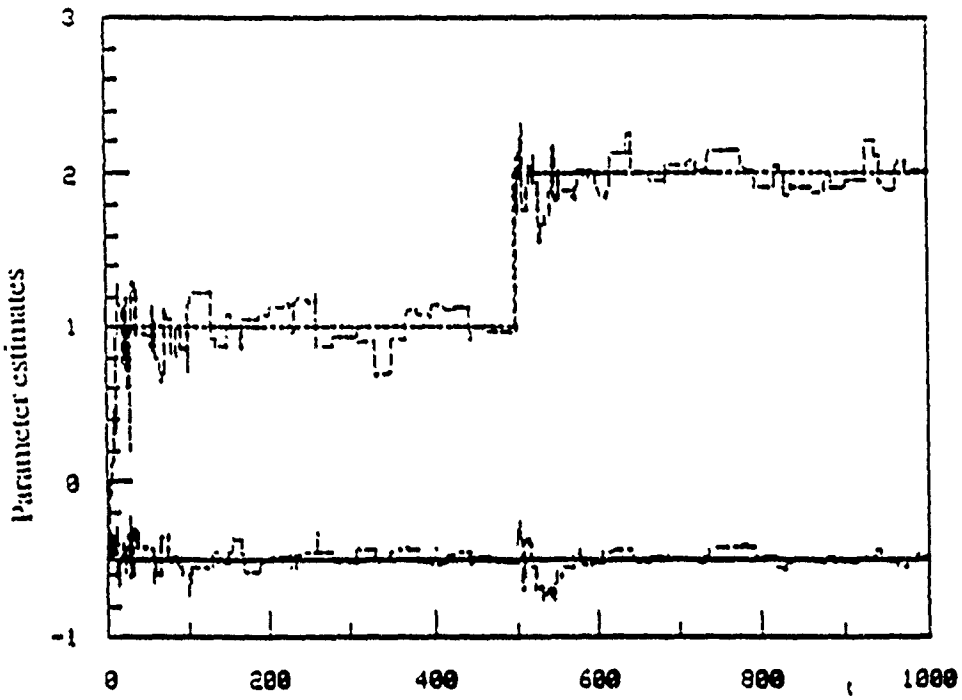


Figure 9. DHOBE parameter estimates when the rescue procedure is activated in the jump parameter case

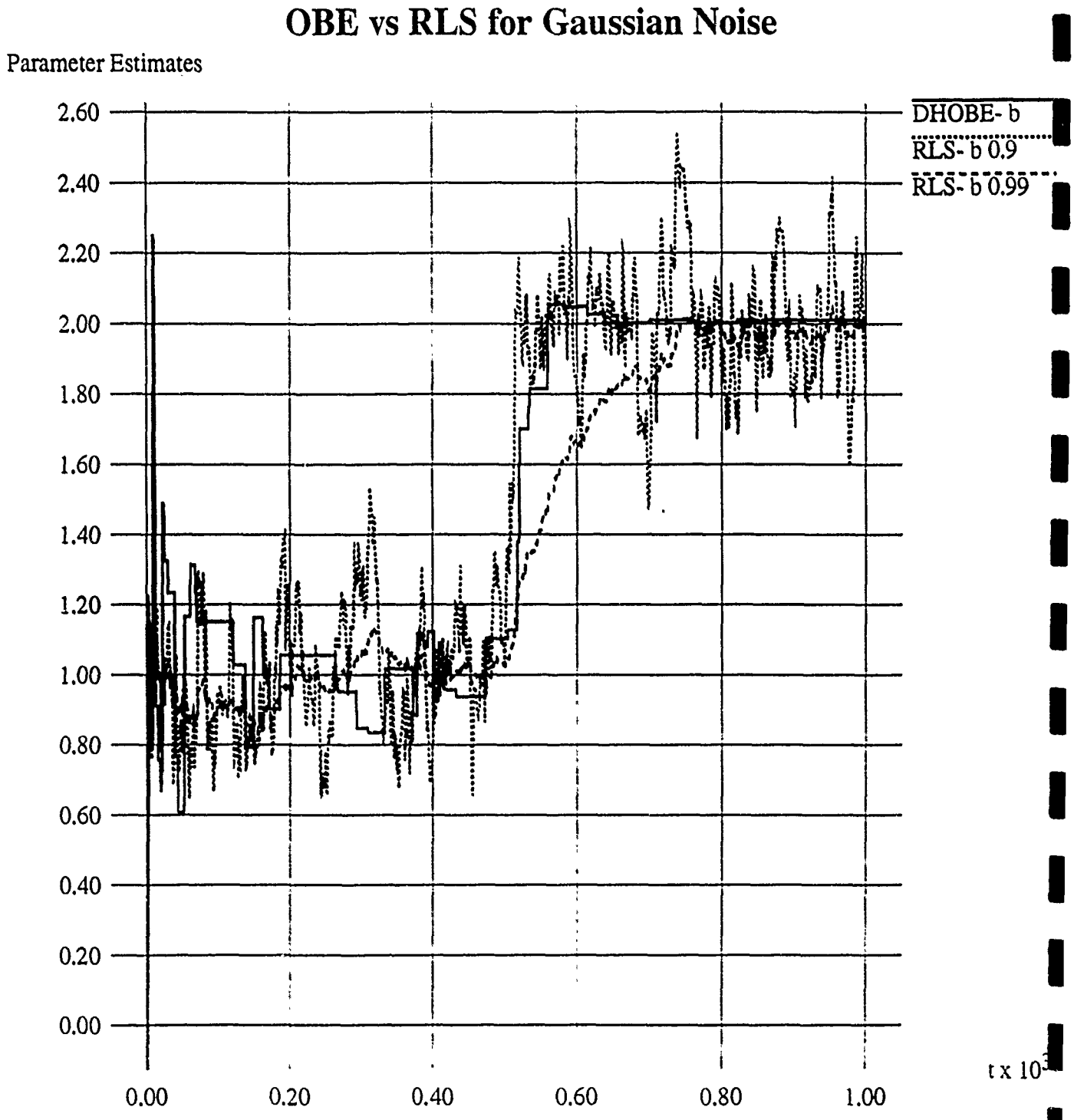


Figure 10. Tracking performance of DHOBE and RLS algorithms for gaussian noise.

EQUATION ERROR AND OUTPUT ERROR METHODS FOR ADAPTIVE SYSTEM IDENTIFICATION

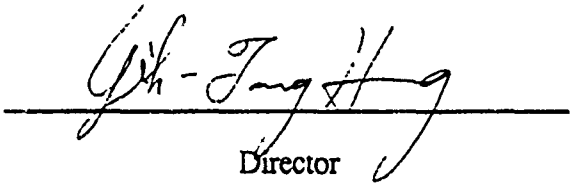
A Thesis

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science in Electrical Engineering

by

Vincent R. Marcopoli, B.S.



Director

Department of Electrical and Computer Engineering
Notre Dame, Indiana
December, 1989

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I owe much thanks to my advisor, Dr. Y. F. Huang, and also to Dr. A. K. Rao for their patience, insight, and guidance in helping me to learn about the area of adaptive systems.

I would also like to thank my family and friends for their encouragement during this time. This work truly would not have been possible without their uncompromising moral support.

This thesis has been supported, in part, by the National Science Foundation under grant MIP-8711174, and, in part, by the office of Naval Research under contracts N00014-89-J-1788.

CHAPTER 1

INTRODUCTION

1.1 Formulation of the Adaptive Filtering Problem

An *adaptive filter* is one which can adjust its impulse response through time in order to reach some desired level of performance. The means of adjusting the adaptive filter is accomplished through an *adaptive algorithm*. This type of filtering is especially needed when dealing with unknown and/or changing environments. Many useful applications have been found for the adaptive filter, such as noise cancellation, echo cancellation in phone lines, equalization of a communication channel to combat intersymbol interference, and system identification. Detailed descriptions of these and other applications of adaptive filters can be found in [Ha86], [Wi75], [Ho84], and [Qu85].

The adaptive filtering problem will be approached here in the context of *system identification*. This is an important area of study for adaptive systems, since many applications of adaptive filters can be put in this context. In system identification, it is desired to characterize a system, usually called the *plant* (see Figure 1.1), with an adaptive filter, based only on the observable input/output data sequences, x(n) and y(n) (see Figure 1.2). The plant/adaptive filter combination will be referred to as the *adaptive system*.

1.2 Notations

The systems of interest in this presentation for the plant and adaptive filter are those which can be represented by linear constant-coefficient difference equations. In the literature, there are three popular notations which are used to express this difference equation input/output relationship. The one that is used depends largely on ease of explanation and the type of results that are needed. However, the varying notations can also be a source of confusion. It is attempted here to explain these three notations in order to avoid future misunderstanding of their meaning. The following notational examples describe a system with input x(n) yielding an output y(n):

1) Difference equation notation.

This is the standard notation found in most texts dealing with digital signal processing [Op75,Ch.1]:

$$y(n) = -\sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=0}^{n_b} b_i x(n-i) \qquad (1.1)$$

The minus sign here is arbitrarily chosen to be consistent with the operator notation, shown next.

2) Operator notation.

The operator q^{-i}, is chosen to represent a delay in its operand signal of i samples, i.e., $q^{-i}x(n)=x(n-i)$. This is analogous to the z-transform representation of a delayed signal. It is now possible to represent (1.1) in operator notation by defining the following polynomials:

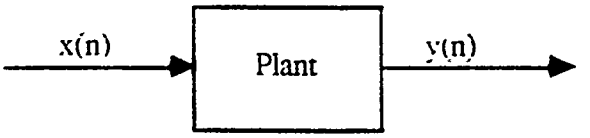


Figure 1.1 The plant

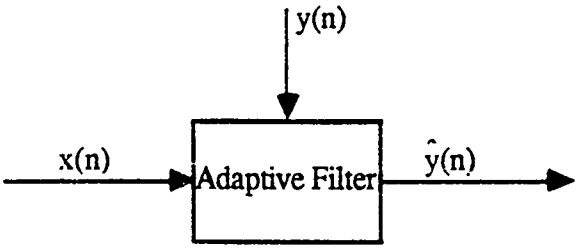


Figure 1.2 The adaptive filter

Referring to Figure 1.1 and Figure 1.2, it is seen that two structures must be decided upon in the system identification problem, yielding a two-step modelling process:

1) The model of the plant structure. This decision is based on some knowledge of how the output signal, y(n), is generated from the input signal, x(n).

2) The adaptive filter structure. This decision is based on practical restrictions on the complexity of the adaptive filter and/or its corresponding adaptive algorithm.

In Chapter 2, an assumption which is common in the field of system identification will be made on the plant structure and two popular choices for the adaptive filter will be investigated. These two structures of adaptive systems are seen in Chapter 3 to yield two families of adaptive algorithms.

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b} q^{-n_b}$$

An equivalent expression to (1.1) is thus:

$$A(q^{-1})y(n) = B(q^{-1})x(n)$$

Note y(n) can be solved for, thus yielding:

$$y(n) = \frac{B(q^{-1})}{A(q^{-1})} x(n) \tag{1.2}$$

It is important to note here that the operator polynomial appearing as a denominator term in (1.2) implies the existence of an *autoregressive* component in the determination of the signal y(n). In other words, y(n) depends on past values ("regressive") of itself ("auto") in addition to the current and past values of the input.

It may appear as if there is a mixing of frequency domain and time domain notations in (1.2). However, this is not the case, since the delay operator was not defined as a complex transform variable as in z-transforms. In interpreting (1.2), it is helpful at first to mentally multiply the expression through by the denominator polynomial, $A(q^{-1})$. Since $A(q^{-1})$ begins with a "1," the first term is y(n) and all the other terms are autoregressive, and can be moved to the right of the equation, yielding the explicit expression for y(n) of (1.1).

An example of operator notation which will be seen often is a pure autoregressive filtering of a signal. This operation, applied to a signal y(n), appears in operator notation as:

$$y'(n) = \frac{1}{A(q^{-1})} y(n)$$

where the prime (') character is used here to denote the autoregressively filtered version of the unprimed signal. Expanding this notation as described in the previous paragraph yields the explicit difference equation relationship:

$$A(q^{-1})y'(n) = y(n)$$

$$y'(n) = y(n) - \sum_{i=1}^{n_a} a_i y'(n-i)$$

3) Matrix notation.

Another convenient means of expressing (1.1) is through matrix operations. Define the *parameter vector*, θ, as:

$$\theta = \left[a_1\ a_2 \cdots a_{n_a}\ b_0\ b_1 \cdots b_{n_b}\right]^T$$

Also define the *regressor vector*, $\varphi(n)$, as:

$$\varphi(n) = \left[-y(n-1)\ -y(n-2) \cdots -y(n-n_a)\ x(n)\ x(n-1) \cdots x(n-n_b)\right]^T$$

These definitions lead to the following equivalent expression for (1.1):

$$y(n) = \theta^T \varphi(n)$$

1.3 Difference Equation Structures

There are five difference equation structures that are the most commonly encountered and dealt with in the literature. In the following, they are presented in terms of the plant structure of Figure 1.1. Note that an unobservable, zero mean white noise component, v(n), is present in all the cases, since an approximation is generally acceptable if it is correct up to some random, independent, zero mean amount. The corresponding

adaptive filter structures are obtained by adding a caret (^) on top of the plant quantities $y(n)$, a_i, b_i, c_i, n_a, n_b, n_c[†], and providing an estimate of the terms involving the unobservable signal, $v(n)$. When $v(n)$ appears alone as an additive modelling error of the plant, its estimate is the expected value, which is zero. In other words, the $v(n)$ term is simply dropped in these cases (which are shown as structures 1) - 3) below). The five structures, in order of increasing complexity, shown in both difference equation and operator notation, are:

1) Exogenous (X)

$$y(n) = \sum_{i=0}^{n_b} b_i x(n-i) + v(n)$$

$$y(n) = B(q^{-1})x(n) + v(n)$$

2) Autoregressive (AR)

$$y(n) = -\sum_{i=1}^{n_a} a_i y(n-i) + v(n)$$

$$A(q^{-1})y(n) = v(n)$$

3) Autoregressive, exogenous input (ARX)

$$y(n) = -\sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=0}^{n_b} b_i x(n-i) + v(n)$$

$$A(q^{-1})y(n) = B(q^{-1})x(n) + v(n)$$

4) Autoregressive, moving average (ARMA)

$$y(n) = -\sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=0}^{n_c} c_i v(n-i)$$

[†] The caret used in this manner denotes a quantity which is an estimate, in some sense, of the corresponding "uncareted" quantity.

$$A(q^{-1})y(n) = C(q^{-1})v(n)$$

5) Autoregressive, moving average, exogenous input (ARMAX)

$$y(n) = -\sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=0}^{n_b} b_i x(n-i) + \sum_{i=0}^{n_c} c_i v(n-i)$$

$$A(q^{-1})y(n) = B(q^{-1})x(n) + C(q^{-1})v(n)$$

The ARMAX model is the most general model which will be considered here, as it contains all previously mentioned models 1) - 4) as special cases. Examples of other more general approaches to modelling are given in [Lj83] and [Ab88]. Specifically, the Box-Jenkins model is discussed in [Lj83], which extends the ARMAX model by replacing the $A(q^{-1})$, $B(q^{-1})$, and $C(q^{-1})$ polynomials with rational functions of polynomials. In [Ab88], the plant is modelled as a linear, continuous-time, time-varying process with no constraints on the order of the system. It is shown there how this very general model yields a nonlinear adaptive filter.

The extent to which a given plant can be identified with an adaptive filter will depend of course on how well the chosen plant structure approximates the true physical system, and also on which structure is chosen for the adaptive filter, as will be seen in Chapter 2. This two-step modelling process is crucial to the success of any adaptive filtering problem.

1.4 Applications of Difference Equation Structures

It is helpful to see how the difference equation structures presented above are utilized by considering two examples of their use in practical situations: Linear predictive coding and echo cancellation.

1.4.1 Linear Predictive Coding (LPC) Speech Modelling

An important modelling example in adaptive filtering is the characterization of the vocal process for the reproduction of speech. The LPC technique for speech modelling is a "black box" method, which assumes a known input of either an impulse train (for voiced sounds) or white noise (for unvoiced sounds) applied to an unknown time-varying system whose output is the final voice signal (See Figure 1.3). The unknown system corresponds to the "plant" in Figure 1.1 and the two-step modelling process of Section 1.1 must be applied in order to characterize this speech-producing system with an adaptive filter. When this is accomplished, speech sounds can be reproduced by exciting a system with the appropriate input signal having the same characteristics as the vocal tract plant. Since these characteristics are time-varying, the adaptive filter is especially suited to this application.

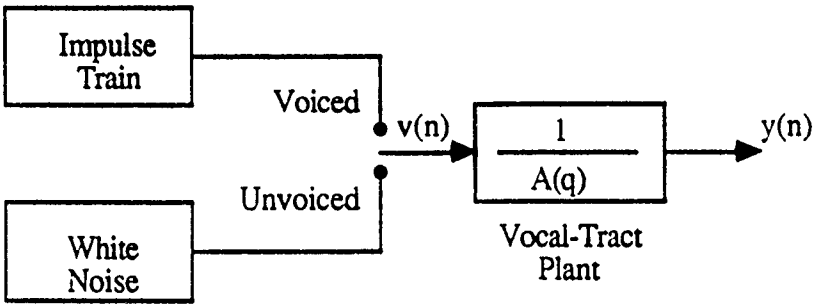
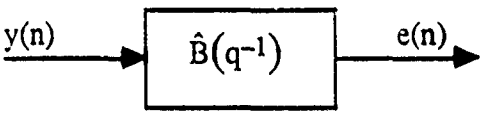


Figure 1.3 A model of speech production

Following the two-step modelling process, it has been seen experimentally that an AR model for the vocal tract is a good choice for the plant structure of the adaptive system (See Figure 1.3). As for the adaptive filter structure, note that the input is not accessible to the adaptive filter. However, the input is assumed to be either white noise or an impulse train. Therefore, if an adaptive filter could reproduce the input given only the output voice signal, it would characterize the inverse of the vocal tract plant, and thus characterize the vocal tract itself. It can be seen that an adaptive filter with an X structure having input y(n)

and output e(n) (See Figure 1.4) can accomplish this inversion when the adaptive filter coefficients are adjusted such that $\hat{b}_i=a_i$, $i=0, \cdots, \hat{n}_b=n_a$ ($a_0=1$). It is shown in [Ha86] that choosing $\hat{b}_i$ parameters such that the mean-square value of the adaptive filter output, e(n), is minimized will yield parameters such that $\hat{b}_i=a_i$.



Adaptive Filter

Figure 1.4 X structure for adaptive filter

1.4.2 Echo Cancellation

In telephone communication, a problem arises when the signal, x(n), which is transmitted over long distances via a "four-wire" line, reaches the "two-wire" line of the destination phone. An echo, y(n), is generated at the meeting of the two transmission lines (the *hybrid*), due to an impedance mismatch. This echo subsequently travels back to the source, which the speaker hears (see Figure 1.5).

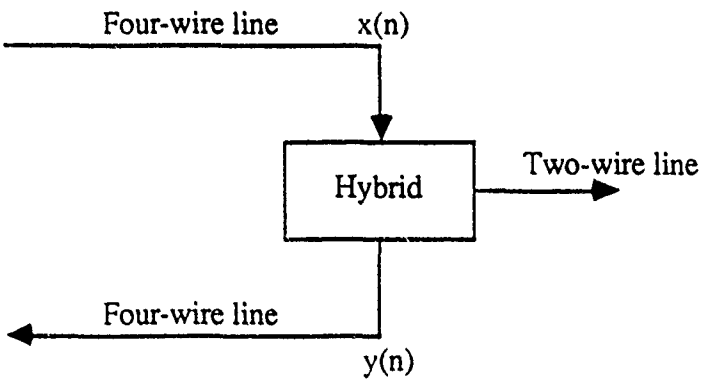


Figure 1.5 Echo generation in phone lines

If the hybrid characteristics were known, a fixed filter could be placed in parallel with the hybrid. The filter output would then be subtracted from the hybrid output, y(n),

thus cancelling the echo. Since the hybrid characteristics are not known and may be time varying as well, a fixed filter will not solve the problem. However, an adaptive filter in this configuration producing an output, $\hat{y}(n)$, has been shown to accomplish very well the task of echo cancellation when the adaptive filter coefficients are adjusted such that the mean-square value of the error signal, $y(n)-\hat{y}(n)$, is minimized. This is illustrated in Figure 1.6, where $H(q^{-1})$ and $\hat{H}(q^{-1})$ stand for rational functions of the operator polynomials introduced in Section 1.2, analogous to the transfer function representation for linear, time-invariant systems.

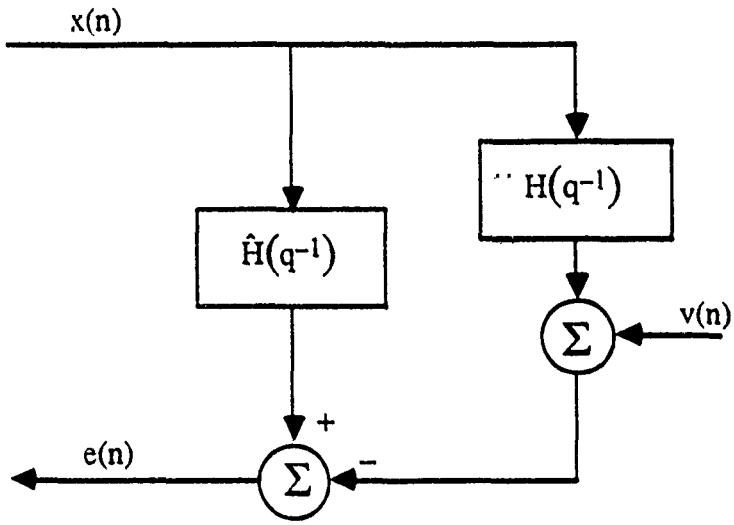


Figure 1.6 Model of hybrid and echo cancellor

At this point, again, models must be decided upon for both the hybrid "plant" and the adaptive filter, through specific choices for $H(q^{-1})$ and $\hat{H}(q^{-1})$. The hybrid has been modelled in different ways, giving rise to various adaptive filter structures. The simplest model of the hybrid is to consider it as an X process [Ha86], [Ho84]. In other words, y(n) depends only on a weighted sum of past inputs. Since the input is available for use by the adaptive filter, the X structure should be adapted so that $\hat{b}_i=b_i$, $i=0, \cdots, \hat{n}_b= n_b$, and thus the error signal will be only white noise. This plant/adaptive filter combination corresponds to the choices:

$$H(q^{-1})=B(q^{-1})$$

$$\hat{H}(q^{-1})=\hat{B}(q^{-1})$$

where $B(q^{-1})$ and $\hat{B}(q^{-1})$ are defined as in Section 1.2.

A slightly more complicated structure for the hybrid models it as an ARX process, which computes its output as a weighted sum of both the incoming signal, x(n-i), i=0, $\cdots$, n_b, as well as its output y(n–i), i=1, $\cdots$,n_a. This is a more realistic system, as this recursive structure contains poles as well as zeros. Note that the adaptive filter in this case can still be chosen with an X structure, because both signals x(n) and y(n) are available to use as inputs. When the adaptive filter weights which multiply x(n–i), i=0, $\cdots$,$\hat{n}_b$, are equal to the corresponding b_i, and those which multiply y(n–i), i=1, $\cdots$,$\hat{n}_a$, are equal to a_i, the error signal will again be white noise.

Finally, a model [Fa88] which is more realistic and which will be considered in some detail later, is to represent the hybrid as being an ARX process with no internal noise term as in the ARX process above, but whose output, p(n), is corrupted by white measurement noise, v(n). It will be shown in Chapter 2 that this model of the hybrid is actually an ARMAX process with $c_i=a_i$, i=1, $\cdots$, n_a. It will be further be shown that the proper structure for the adaptive filter to cancel the echo is the ARX structure. In other words the adaptive filter must be recursive (IIR) in order to cancel the echo, y(n), produced at the hybrid. Referring again to Figure 1.6, this plant and adaptive filter are characterized by:

$$H(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})}$$

$$\hat{H}(q^{-1}) = \frac{\hat{B}(q^{-1})}{\hat{A}(q^{-1})}$$

Note that if v(n)≡0, the situation reduces to the ARX plant described in the preceding paragraph. Thus the simpler X structure adaptive filter with inputs x(n) and y(n)=p(n) can be used. The system structure for this situation is slightly different than the one shown in Figure 1.6, and is shown in Figure 1.7. In this case two FIR filters are used - one which realizes the zeros, and one which realizes the poles of $H(q^{-1})$. It is important to see here how, when $\hat{A}(q^{-1}) = A(q^{-1})$ and $\hat{B}(q^{-1}) = B(q^{-1})$, the signal e(n) is zero.

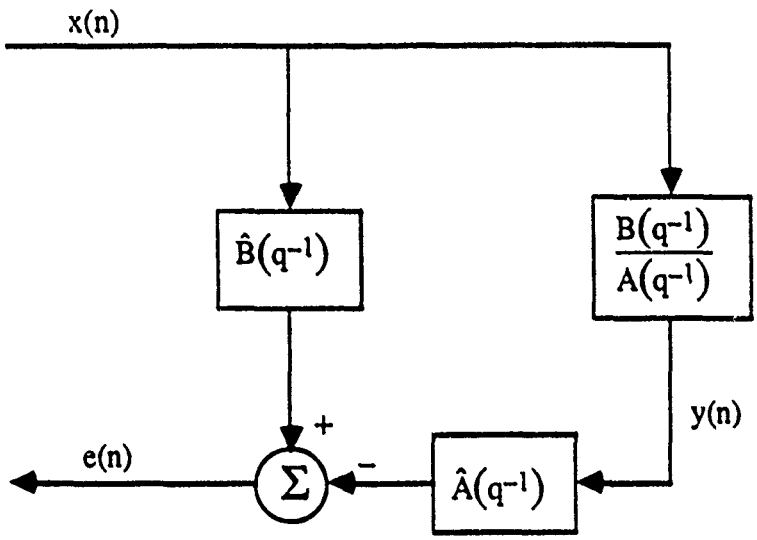


Figure 1.7 Echo cancellation for a recursive plant when v(n)≡0

1.5 The Mean Square Error Criterion

In order for an adaptive algorithm to adjust the impulse response of its adaptive filter, the algorithm must somehow be able to gauge its progress to determine how to make the adjustment. A natural criterion on which to base this adjustment is the difference between the output of the plant, y(n), and that of the adaptive filter, $\hat{y}(n)$. Thus the *error signal* is defined as $e(n)=y(n)-\hat{y}(n)$[†] . Intuitively, the magnitude of the error signal should

[†] The notation e(n) will be used to refer to the error signal of *any* adaptive system. In Chapter 3, error quantities for specific adaptive systems will be defined as ee(n) and oe(n). The cause of the difference between ee(n) and oe(n) will be seen to be the manner in which the adaptive filter output, $\hat{y}(n)$, is generated.

be as small as possible for desired operation. Mathematically, however, the criterion $|y(n)-\hat{y}(n)|$ isn't very attractive. A mathematically sounder criterion leading to efficient adaptive algorithms is the *mean - square error* (MSE) which is expressed as $E\{e^2(n)\}$. The squaring operation provides an alternative to the absolute value operation. Statistical expectation is needed because, as noted previously, the plant models are all assumed to be accurate to within an independent, zero-mean noise term, v(n).

The squaring operation can also be viewed as providing a criterion which tends to emphasize larger values of the error signal while diminishing the importance of smaller errors, as opposed to the absolute value operation, which linearly assigns an error penalty according to the magnitude of the error, $|e(n)|$. This is an intuitively reasonable characteristic for a criterion of "goodness" to have. However, this also causes some algorithms to adapt more slowly as the MSE of the adaptive filter decreases. Depending on the goal of the adaptive system, this may or may not be of significance.

1.6 Overview

In what follows in this thesis, some basics in the area of adaptive systems from the perspective of system identification will be developed, as well as experimental results obtained by the author. In particular, Chapter 2 introduces two important classes of system identification models: The equation error and output error adaptive systems. Chapter 3 presents methods of adjusting an adaptive filter (i.e. adaptive algorithms) in the context of both the equation error and output error adaptive systems. This will be seen to give rise to two different families of adaptive algorithms. In Chapter 4, an adaptive algorithm is presented which combines elements from two adaptive schemes studied in Chapter 3. Simulation results are given which show empirically that the algorithm works, and comparisons are made with the standard method. Finally, two output error algorithms

developed by the author are presented for review. These are preliminary results and no simulations have been performed yet.

CHAPTER 2

CONCEPTS OF ADAPTIVE FILTERING

2.1 Modelling Techniques

In system identification the plant structure is usually modelled as a rational transfer function whose output, p(n), is corrupted by additive white measurement noise, v(n), to yield the observable signal, y(n) (see Figure 2.1). This plant is a special case of the ARMAX structure, which can be seen by taking the expression for the output:

$$y(n) = \frac{B(q^{-1})}{A(q^{-1})}x(n) + v(n)$$

and multiplying through by the $A(q^{-1})$ polynomial to yield:

$$A(q^{-1})y(n) = B(q^{-1})x(n) + A(q^{-1})v(n)$$

Or, equivalently, using difference equation notation:

$$y(n) = -\sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=0}^{n_b} b_i x(n-i) + \sum_{i=0}^{n_a} a_i v(n-i) \qquad (2.1)$$

Note in the last summation above, $a_0=1$. It is thus seen that the plant structure of Figure 2.1 is a special case of the ARMAX structure with $C(q^{-1})=A(q^{-1})$.

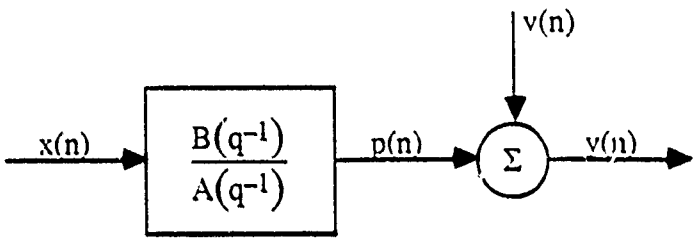


Figure 2.1 An ARMAX plant with $C(q^{-1}) = A(q^{-1})$

Given this plant, there are two approaches that can be taken in modelling this plant with an adaptive filter — the *equation error* approach and the *output error* approach.

2.1.1 Equation Error Model

Since x(n) and y(n) are both measurable, the simplest approach is to use these signals to form the output of the adaptive filter similar to how the plant forms the output in (2.1). This approach yields the expression for the adaptive filter output:

$$\hat{y}(n) = -\sum_{i=1}^{\hat{n}_a} \hat{a}_i(n-1)y(n-i) + \sum_{i=0}^{\hat{n}_b} \hat{b}_i(n-1)x(n-i) \tag{2.2}$$

Since the v(n) terms of the plant can not be measured, they will be neglected. Note that the last available values of the adaptive filter parameter estimates, $\hat{a}_i(n-1)$ and $\hat{b}_i(n-1)$, are used to determine the adaptive filter output. An adaptive algorithm uses the error signal, $y(n)-\hat{y}(n)$, to determine the new "current" estimates, $\hat{a}_i(n)$ and $\hat{b}_i(n)$.

The expression (2.2) can be expressed in matrix notation as:

$$\hat{y}(n) = \hat{\theta}(n-1)^T \varphi_{ee}(n) \tag{2.3}$$

where the regressor vector, $\varphi_{ee}(n)$, is defined as follows:

$$\varphi_{ee}(n) = \left[-y(n-1) \cdots -y(n-n_a)\ x(n) \cdots x(n-n_b)\right]^T$$

Using operator notation, the description of the adaptive system is:

$$y(n) = \left[1-A\left(q^{-1}\right)\right]y(n) + B\left(q^{-1}\right)x(n) + A\left(q^{-1}\right)v(n) \tag{2.4}$$

$$\hat{y}(n) = \left[1-\hat{A}\left(q^{-1},n-1\right)\right]y(n) + \hat{B}\left(q^{-1},n-1\right)x(n) \tag{2.5}$$

The appropriate structures of the equation error model,resulting from the two-step modelling process described in Chapter 1, can be recognized from (2.1) and (2.2) for the plant and adaptive filter as ARMAX and X, respectively.

This adaptive system generates an error signal $y(n) - \hat{y}(n)$ known as the *equation error*, ee(n). Subtracting (2.5) from (2.4) yields:

$$\begin{aligned} ee(n) &= y(n) - \hat{y}(n) \\ &= -\left[A\left(q^{-1}\right)-\hat{A}\left(q^{-1},n-1\right)\right]y(n) + \\ &\qquad \left[B\left(q^{-1}\right)-\hat{B}\left(q^{-1},n-1\right)\right]x(n) + A\left(q^{-1}\right)v(n) \end{aligned} \tag{2.6}$$

Re-expressing (2.6) utilizing matrix notation yields the following useful relationship between ee(n) and the parameter error vector, $\tilde{\theta} = \theta_p - \hat{\theta}(n-1)$:

$$ee(n) = \tilde{\theta}(n-1)^T\varphi_{ee}(n) + A\left(q^{-1}\right)v(n) \tag{2.7}$$

An alternative expression for ee(n) can be obtained from (2.6) by noting from (2.1) that $-A\left(q^{-1}\right)y(n)+B\left(q^{-1}\right)x(n)+A\left(q^{-1}\right)v(n)=0$. This yields the following expression for the equation error:

$$ee(n) = \hat{A}\left(q^{-1},n-1\right)y(n) - \hat{B}\left(q^{-1},n-1\right)x(n) \tag{2.8}$$

Equation (2.8) implies the *series-parallel* structure of Figure 2.2 for the equation error model of an adaptive system. Note that this structure requires only FIR filters for its implementation.

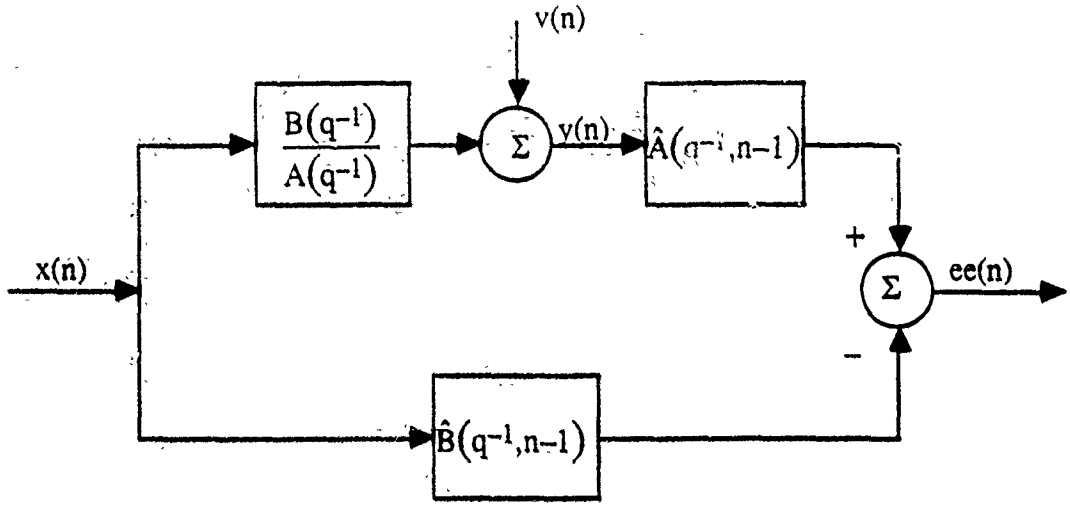


Figure 2.2 Series-parallel structure of the equation error adaptive system

2.1.2 Output Error Model

The output error model adaptive filter attempts to duplicate the structure of the assumed plant model. This adaptive system can most easily be introduced through a diagram of its structure, shown in Figure 2.3.

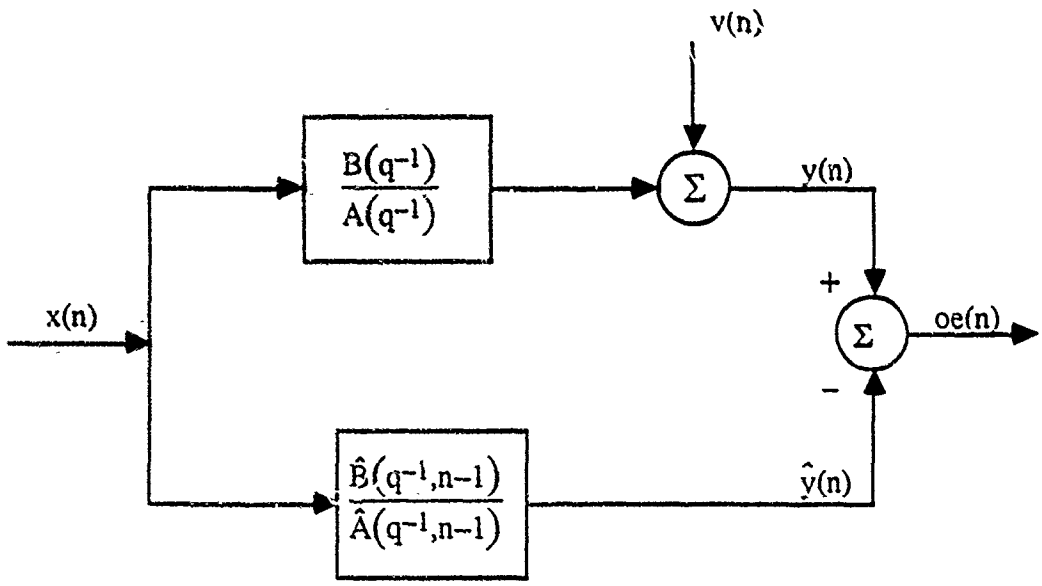


Figure 2.3 Parallel structure of the output error adapive system

A few comments are in order here regarding this alternative system identification structure. Note the *parallel* structure of the output error model, in contrast to the series-parallel equation error model structure. This implies that the adaptive filter is independent of the plant, sharing only the common input signal. An important consequence of this characteristic is that the noise, v(n), is not introduced in the adaptive filter, as it is in the equation error model through y(n)=p(n)+v(n) (see Figure 2.2). Thus it might be reasonable to expect that the measurement noise, v(n), will have less of an effect on the performance of the output error model than the equation error model. This is in fact true, as will be shown shortly. It is also important to note that the adaptive filter in the output error model is an IIR filter. In other words, the autoregressive portion of the adaptive filter uses past values of $\hat{y}(n-i)$, $i=1,\cdots,\hat{n}_a$, in determining its current output $\hat{y}(n)$. This is in contrast to the equation error model, which uses past values of the plant output, $y(n-i)$, $i=1,\cdots,\hat{n}_a$, in determining $\hat{y}(n)$. The output of the adaptive filter can be expressed compactly in matrix notation similar to that of (2.3) for the equation error adaptive filter:

$$\hat{y}(n) = \hat{\theta}(n-1)^T \varphi_{oe}(n)$$

where

$$\varphi_{oe}(n) = \left[-\hat{y}(n-1)\ -\hat{y}(n-2)\ \cdots\ -\hat{y}(n-\hat{n}_a)\ x(n)\ x(n-1)\ \cdots\ x(n-\hat{n}_b)\right]^T \qquad (2.9)$$

The difference of the regressors in (2.3) and (2.9) captures very concisely the fundamental difference between the equation error and the output error methods.

Referring to Figure 2.3, the output error adaptive system is seen to be described by the following equations:

$$A(q^{-1})y(n) = B(q^{-1})x(n) + A(q^{-1})v(n) \qquad (2.10)$$

$$\hat{A}(q^{-1},n-1)\hat{y}(n) = \hat{B}(q^{-1},n-1)x(n) \qquad (2.11)$$

The two-step modelling process described in Chapter 1 has thus yielded the structures of ARMAX and ARX for the plant and adaptive filter, respectively.

The expression for the error signal, $y(n)-\hat{y}(n)$, can now be derived. This signal is known as the output error, oe(n). Subtracting (2.11) from (2.10) yields:

$$A(q^{-1})y(n) - \hat{A}(q^{-1},n-1)\hat{y}(n)$$

$$= \left[B(q^{-1})-\hat{B}(q^{-1},n-1)\right]x(n) + A(q^{-1})v(n) \qquad (2.12)$$

In order to get an expression for the output error, $oe(n) = y(n)-\hat{y}(n)$, either $A(q^{-1})\hat{y}(n)$ or $\hat{A}(q^{-1},n-1)y(n)$ can be added and then subtracted to the left side of (2.12). This results in two different interpretations of oe(n). Choosing $\hat{A}(q^{-1},n-1)y(n)$ yields:

$$A(q^{-1})y(n) - \hat{A}(q^{-1},n-1)\hat{y}(n) + \hat{A}(q^{-1},n-1)y(n) - \hat{A}(q^{-1},n-1)y(n)$$

$$= \left[B(q^{-1})-\hat{B}(q^{-1},n-1)\right]x(n) + A(q^{-1})v(n)$$

Factoring $oe(n) = y(n) - \hat{y}(n)$ gives:

$$\hat{A}(q^{-1},n-1)oe(n) + \left[A(q^{-1})-\hat{A}(q^{-1},n-1)\right]y(n)$$

$$= \left[B(q^{-1})-\hat{B}(q^{-1},n-1)\right]x(n) + A(q^{-1})v(n)$$

Solving for oe(n):

$$\hat{A}(q^{-1},n-1)oe(n) =$$

$$\left\{-\left[A(q^{-1})-\hat{A}(q^{-1},n-1)\right]y(n)+\left[B(q^{-1})-\hat{B}(q^{-1},n-1)\right]x(n)+A(q^{-1})v(n)\right\}$$

The term in braces can be recognized as the equation error, ee(n). Thus

$$oe(n) = \frac{1}{\hat{A}(q^{-1},n-1)}ee(n) \qquad (2.13)$$

It is of interest to examine Equation (2.13). A relation is now apparent between the two adaptive system models. Namely, given the same input and noise sequence to both the equation error and output error models, the resulting error sequences are related through a filtering by the adaptive filter's denominator polynomial, $\hat{A}(q^{-1},n-1)$. This relationship will be exploited later in the development of an adaptive algorithm.

Choosing the term $A(q^{-1})\hat{y}(n)$ to add and subtract in (2.12) yields a relationship, analogous to the equation error expression (2.7), between oe(n) and the parameter error vector, $\tilde{\theta}(n-1) = \hat{\theta}(n-1) - \theta_p$ [Jo84]:

$$A(q^{-1})y(n) - \hat{A}(q^{-1},n-1)\hat{y}(n) + A(q^{-1})\hat{y}(n) - A(q^{-1})\hat{y}(n) =$$

$$\left[B(q^{-1})-\hat{B}(q^{-1},n-1)\right]x(n) + A(q^{-1})v(n)$$

Similarly factoring and simplifying yields:

$$A(q^{-1})oe(n) + \left[A(q^{-1})-\hat{A}(q^{-1},n-1)\right]\hat{y}(n)$$

$$= \left[B(q^{-1})-\hat{B}(q^{-1},n-1)\right]x(n) + A(q^{-1})v(n)$$

$$oe(n) = \frac{1}{A(q^{-1})}\left\{-\left[A(q^{-1})-\hat{A}(q^{-1},n-1)\right]\hat{y}(n)\right.$$

$$\left. + \left[B(q^{-1})-\hat{B}(q^{-1},n-1)\right]x(n)\right\} + v(n)$$

$$oe(n) = \frac{1}{A(q^{-1})}\tilde{\theta}^T(n-1)\varphi_{oe}(n) + v(n) \qquad (2.14)$$

Expressions (2.14) and (2.7) make very clear the effect of the noise, v(n), in each of the adaptive system models. In the output error model, assuming v(n) is white, the noise power is simply σ_v^2 from (2.14). However, examination of (2.7) shows the noise power of the equation error model to be $(1+a_1^2+a_2^2+\cdots+a_{\hat{n}_a}^2)\sigma_v^2$. Thus it is seen that the measurement noise affects the output error model much less than it affects the equation

error model. The implications of these additional effects of the noise in the equation error modes are discussed in section 2.3.1, in particular with respect to the quality of the parameter estimates, $\hat{\theta}$. It will be shown that the presence of the measurement noise, v(n), produces a *bias* in the equation error estimates, $\hat{\theta}$, with respect to the plant parameters, θ_p.

2.2 The Mean Square Error Surface

As discussed in Chapter 1, both the criterion of performance and the adjustment mechanism depend on the nature of the error signal, $y(n)-\hat{y}(n)$. Therefore it is important to examine the characteristics of this signal for the output error and equation error models. In particular, the *mean square error* (MSE) *surface* will be discussed for both models. The MSE surface is the relationship between the MSE and the adaptive filter coefficients. Later, in Section 2.3.2, these characteristics will be examined as to how they affect the ability of an adaptive filter to reach an "optimal" state.

2.2.1 Equation Error Surface

In the equation error derivations which follow here and in Section 2.3.1, the parameter estimate vector, $\hat{\theta}$, will be considered to be a constant quantity with respect to the statistics of the input process, x(n). This assumption is obviously not true, since, as will be seen in Chapter 3, the parameters are updated by algorithms which use the input, among other things, to accomplish the updating process. The following results effectively evaluate characteristics of an adaptive filter whose impulse response, $h(i)=\hat{\theta}_i$, is set to some arbitrary constant value. Consideration of the adaptive filter in this way permits the use of techniques from Wiener filtering theory [Ha86, Ch.3]. This provides a means to evaluate the performance of an adaptive filter and establishes a useful basis for comparison of an

adaptive filter to the ideal, time-invariant situation. This is especially valid under the reasonable, common assumption of a slowly time-varying adaptive filter.

In deriving the expression for the MSE surface, it is helpful to use matrix notation. Recall equation (2.8), reprinted here in matrix form as well:

$$ee(n) = \hat{A}(q^{-1})y(n) - \hat{B}(q^{-1})x(n) = y(n) - \hat{\theta}^T\varphi(n) \tag{2.15}$$

Note here that ee(n) is linear in the parameters. Thus $ee^2(n)$ is quadratic in the parameters, yielding a parabolic MSE surface, and is shown in the following. Squaring, expanding, and taking expectations of (2.15) yields:

$$E[ee^2(n)] = E[y^2(n)] - 2E[y(n)\hat{\theta}^T\varphi(n)] + \hat{\theta}^T E[\varphi(n)\varphi^T(n)]\hat{\theta}$$

Defining $\mathbf{R}(n) = E[\varphi(n)\varphi^T(n)]$, $\sigma_y^2(n) = E[y^2(n)]$, and rearranging yields:

$$E[ee^2(n)] = \sigma_y^2(n) - 2\hat{\theta}^T E[y(n)\varphi(n)] + \hat{\theta}^T\mathbf{R}(n)\hat{\theta} \tag{2.16}$$

For stationary processes, The expectation operation yields values independent of n. Thus $\sigma_y^2(n)=\sigma_y^2$ and $\mathbf{R}(n)=\mathbf{R}$ are a constant value and matrix, respectively, and the equation error model possesses an error surface which is quadratic in the parameters[†] . This is a very desirable property because many adaptive schemes require the MSE surface to have no local minimum points for guaranteed convergence.

Another useful expression for the MSEE can also be derived using (2.7), reprinted here for convenience:

$$ee(n) = \tilde{\theta}^T\varphi_{ee}(n) + A(q^{-1})v(n)$$

Squaring and taking expectations yields:

[†] The notation for the MSE surface of $E[e^2(n)]$ is mathematically misleading because there is no indication of the dependence of this function on $\hat{\theta}$. However, this is the standard notation in the literature and the dependence on $\hat{\theta}$ must be tacitly assumed.

$$E\left[ee^2(n)\right] = \tilde{\theta}^T\mathbf{R}\tilde{\theta} + 2\tilde{\theta}^T E\left\{\left[A(q^{-1})v(n)\right]\varphi_{ee}(n)\right\} + (1+a_1^2+a_2^2+\cdots+a_{\hat{n}_a}^2)\sigma_v^2$$

The second term can be simplified by first expanding the operator polynomial, $A(q^{-1})$:

$$E\left\{\left[A(q^{-1})v(n)\right]\varphi_{ee}(n)\right\} = E\left\{\sum_{i=1}^{n_a} a_i v(n-i)\begin{bmatrix} -y(n-1) \\ -y(n-2) \\ \vdots \\ x(n) \\ \vdots \\ x(n-n_b) \end{bmatrix}\right\}$$

Noting that v(n) is white and y(n)=p(n)+v(n) from Figure 2.1, it can be seen that the summation will expand as:

$$= a_1\begin{bmatrix} -\sigma_v^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + a_2\begin{bmatrix} 0 \\ -\sigma_v^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + a_{n_a}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ -\sigma_v^2 \end{bmatrix}$$

$$= -\sigma_v^2\begin{bmatrix} a_1 \\ \vdots \\ a_{n_a} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (2.17)$$

Defining the vector **a** as the $(\hat{n}_a+\hat{n}_b+1)$-dimensional vector in (2.17), i.e. as the plant parameter vector θ_p with the b_i parameters set equal to zero, gives the desired expression for this term:

$$E\left[A(q^{-1})v(n)\varphi_{ee}(n)\right] = -\sigma_v^2\mathbf{a}$$

Thus an alternative expression for the equation error, ee(n), is:

$$E\left[ee^2(n)\right] = \tilde{\theta}^T R \tilde{\theta} - 2\sigma_v^2 \tilde{\theta}^T a + (1+a_1^2+a_2^2+\cdots+a_{\hat{n}_a}^2)\sigma_v^2 \qquad (2.18)$$

2.2.2 Output Error Surface

To find the expression for the MSE of the output error model, recall equation (2.13), again neglecting the time dependence of the coefficients:

$$oe(n) = \frac{1}{\hat{A}(q^{-1})} ee(n)$$

Expanding the operator notation yields:

$$oe(n) = ee(n) - \sum_{i=1}^{\hat{n}_a} \hat{a}_i oe(n-i)$$

From this expression it is evident that oe(n) must be a highly nonlinear function of the $\hat{a}_i$ parameters, since it is the solution to a difference equation. The procedure for finding the explicit expression for $E\left[oe^2(n)\right]$ in terms of the parameters $\hat{a}_i$ and $\hat{b}_i$ is given in [Wi85,Ch.7].

The highly nonlinear nature of the MSE surface in the output error model suggests possible local minima in this surface. Indeed, it is this characteristic of the output error model that causes most algorithms to fail. The problem of local minima, together with the inherent stability restrictions of using an IIR adaptive filter, have severely limited output error modelling in practical situations. These issues are examined more closely in the following section.

2.3 Equation Error and Output Error: A Comparison

In any practical situation, an appropriate model for the given problem must be chosen. Therefore it is important to compare the output and equation error models. Three important criteria on which to base this comparison are:

1) Performance capability. In particular, it is of interest to examine the minimum achievable mean square error as well as the quality of the parameter estimates. In other words, what is the best performance that can be expected from the chosen model?

2) Characteristics of the MSE surface. The nature of the MSE surface can drastically affect the ability of an adaptive algorithm to minimize this quantity.

3) Stability considerations.

2.3.1 Performance Capability

It is clear that in the problem of system identification, the best achievable performance will be limited by the amount of measurement noise, v(n). This is easiest to see with the output error model of Figure 2.3. It can be seen that if $\hat{B}(q^{-1}) = B(q^{-1})$ and $\hat{A}(q^{-1}) = A(q^{-1})$, then oe(n)=y(n)–$\hat{y}$(n)=v(n). Therefore the output error model will give the best achievable performance when $\hat{\theta}=\theta_p$, yielding the minimum MSE of σ_v^2. In the literature, i.e. [So88, eq.2.2], [Lj83, p.109], v(n) itself is often defined as the output error, since this is the error which results from the output error model with the adaptive filter adjusted so that $\hat{\theta}=\theta_p$. This can also be seen in (2.14), where oe(n)=v(n) if $\tilde{\theta}=\hat{\theta}-\theta_p=0$. Many adaptive schemes attempt to minimize the MSE, as will be seen in Chapter 3.

Therefore, using the output error model, MSE - minimizing adaptive algorithms will provide what are called *unbiased estimates* of the plant parameters after convergence. The term *unbiased* refers to parameter estimates, $\hat{\theta}$, whose ensemble average is equal to the true plant parameters, θ_p. In other words, $E[\hat{\theta}]=\theta_p$ after convergence. This is a desirable property for an adaptive system to have.

If v(n) appeared by itself as an additive term in the equation error expression (2.7) as it does in the output error expression (2.13) instead of being filtered by $A(q^{-1})$, it could similarly be reasoned that the equation error adaptive system simultaneously provides unbiased estimates and minimum possible MSE of σ_v^2. Such a situation would occur if the plant had an ARX ($\hat{n}_a>0$, $\hat{n}_b>0$) or X ($\hat{n}_a=0$, $\hat{n}_b>0$), since for these cases:

$$y(n) = \theta_p^T \varphi_{ee}(n) + v(n)$$

$$\hat{y}(n) = \hat{\theta}^T \varphi_{ee}(n)$$

and therefore:

$$ee(n) = y(n) - \hat{y}(n) = \tilde{\theta}^T \varphi_{ee}(n) + v(n) \qquad (2.19)$$

However, for the current, more practical case of the ARMAX plant structure appearing in the equation error adaptive system of Figure 2.2, the issue of MMSE and unbiased estimates is not as intuitively clear. Observe in (2.18) that if $\tilde{\theta}=\hat{\theta}-\theta_p=0$, then $E[ee^2(n)]=(1+a_1^2+a_2^2+\cdots+a_{\hat{n}_a}^2)\sigma_v^2$. If this MSE is in fact the minimum MSE achievable with the equation error model, then this model would also yield unbiased estimates. It is not clear from a cursory examination of (2.18) whether this is so, as it was in the output error case upon examination of (2.14). However, since the MSEE is quadratic in the parameters, there should be no trouble taking derivatives to find the parameter which yields the minimum MSEE. Minimization will be on (2.18), repeated here for convenience:

$$E\left[ee^2(n)\right] = \tilde{\theta}^T R\tilde{\theta} - 2\sigma_v^2\tilde{\theta}^T a + (1+a_1^2+a_2^2+\cdots+a_{n_a}^2)\sigma_v^2$$

Since differentiation is with respect to $\hat{\theta}$, only the first two terms will have nonzero derivatives. Differentiating the first term, remembering that $\tilde{\theta}=\theta_p-\hat{\theta}$, yields:

$$\frac{d}{d\hat{\theta}}\left[\tilde{\theta}^T R\tilde{\theta}\right] = -2R\tilde{\theta}$$

Similarly differentiating the second term yields:

$$\frac{d}{d\hat{\theta}}\left[2\sigma_v^2\tilde{\theta}^T a\right] = -2\sigma_v^2 a$$

Now setting the derivative equal to zero gives:

$$-2R\theta_p + 2R\hat{\theta}^* + 2\sigma_v^2\tilde{\theta}^{*T}a = 0$$

Finally, solving for the parameter vector, $\hat{\theta}^*$, which gives the MMSEE yields:

$$\hat{\theta}^* = R^{-1}\left[R\theta_p - \sigma_v^2 a\right] = \theta_p - \sigma_v^2 R^{-1}a \qquad (2.20)$$

This is an important result. Examining (2.20), it is seen that one of two conditions must be met if the equation error adaptive system of Figure 2.2 is to provide unbiased estimates:

1) $v(n) \equiv 0$

2) $a \cong 0$.

Condition 2) states that the plant has no autoregressive component, which implies an X plant structure. Further examination of (2.20) shows that as the variance of v(n) increases, the a_i parameter estimates will be increasingly biased. This is the major problem with the equation error adaptive system. Note that the b_i parameter estimates, however, will be unbiased.

It can further be shown that, given neither of the above two conditions are met, the minimum value of the MSEE is always greater than the minimum value of the MSOE, which is σ_v^2, as follows. Subsituting $\hat{\theta}^*$ back into (2.20) yields the minimum value of the mean-squared equation error:

$$E\left[ee^2(n)\right]_{min} = \left[\sigma_v^2 R^{-1}a\right]^T R\left[\sigma_v^2 R^{-1}a\right] + (1+a_1^2+a_2^2+\cdots+a_{\hat{n}_a}^2)\sigma_v^2$$

$$-2\left[\sigma_v^2 R^{-1}a\right]^T a\sigma_v^2$$

$$= \sigma_v^4 a^T R^{-1}a + (1+a_1^2+a_2^2+\cdots+a_{\hat{n}_a}^2)\sigma_v^2 - 2\sigma_v^4 a^T R^{-1}a$$

$$= \left(1+a^T a\right)\sigma_v^2 - \sigma_v^4 a^T R^{-1}a$$

$$= \sigma_v^2 + \sigma_v^2 a^T\left[I-\sigma_v^2 R^{-1}\right]a \qquad (2.21)$$

It can therefore be seen that the mean-square value of the equation error will be always greater than that of the output error only if the term in brackets in (2.21) is positive definite. To show this, it is sufficient to take R to be the autocorrelation matrix of the plant output, y(n), and a to be the $n_a \times 1$ vector of the a_i parameters without the zeros appended as in (2.17). Since v(n) is white and y(n)=p(n)+v(n), it can be seen:

$$R = R_p + \sigma_v^2 I$$

Postmultiplying the term in brackets in (2.21) by R will not change its definiteness. Performing this operation yields the desired result:

$$\left[I-\sigma_v^2 R^{-1}\right]R = R - \sigma_v^2 I = R_p > 0$$

It can thus be seen that the equation error model, in addition to providing biased estimates of the plant parameter, θ_p, yields a higher value of minimum MSE than the output error model.

Regarding this minimum MSE achievable by the equation error model, it is worthy to note here a tradeoff that exists in this model. In general the minimum value of the MSEE can be lowered by increasing $\hat{n}_a$ above n_a. In fact, as $\hat{n}_a \rightarrow \infty$, the MMSEE$\rightarrow\sigma_v^2$ [Wi75, Sec.IV]. In other words, the performance of the equation error model can be arbitrarily improved by increasing the order of $\hat{A}(q^{-1},n)$. This, however, increases the computational burden and memory requirements of the adaptive system.

These results illustrate the superiority of the output error model over the equation error model in the system identification problem, with respect to the ability to reach an "optimal" state, which is now seen to mean that it can provide simultaneous minimization of the MSE and unbiased estimates. This superiority is also intuitively reasonable, since unlike the equation error model, the output error adaptive filter is an IIR filter, just as the assumed plant, and it would seem better to model an IIR plant with an IIR filter.

2.3.2 Characteristics of the MSE Surface

As noted in the previous section, many adaptive algorithms attempt to choose a parameter, $\hat{\theta}$, in an $(\hat{n}_a+\hat{n}_b+1)$ - dimensional space which minimizes the mean square error. This is implemented in recursive fashion in an adaptive algorithm in such a way that, given its last selection for the parameter vector[†] , $\hat{\theta}(n-1)$, pick a new one, $\hat{\theta}(n)$, which yields a lower MSE. This is done by effectively "looking down" the MSE surface from an initial estimate, $\hat{\theta}(0)$, and choosing the parameter at the bottom as the final estimate. Different

[†] Always having a "last" parameter esumate implies that the algorithm must be given an initial estimate, $\hat{\theta}(0)$, before starting.

algorithms do this in different ways, but the main point here is that they are all trying to get to that same "bottom' point. This process works fine if the MSE surface is reasonably smooth and has no local minimum points, since "looking down" from any parameter vector in the $(\hat{n}_a+\hat{n}_b+1)$ - dimensional space will always lead to the minimum point.

It was seen in Section 2.2.1 that the equation error model has a quadratic MSE surface. This yields a "bowl-shaped" surface in the parameter space. Therefore the minimum point can be found by "looking down" from any initial point, $\hat{\theta}(0)$. Thus the parameter yielding the minimum MSEE will always be reached.

The output error model, however, has a much more complicated MSE surface and, as noted in Section 2.2.2, it may have local minima. Therefore, simply "looking down" the MSE surface may not lead to its global minimum point. Since there is no way for the algorithm to know if it is at a local or global minimum, it will converge to either, depending on the initial estimate, $\hat{\theta}(0)$. By converging to a local minimum point of the MSE surface, the output error model loses its most desirable feature which was discussed in the previous section: Simultaneously minimizing the MSOE (recall that this value is σ_v^2) and providing unbiased estimates of the plant parameters, θ_p.

The lack of global convergence of algorithms trying to minimize output error is the major reason why use of the output error model has been mainly in computer simulations in research labs and not in practical applications. Recently, however, an algorithm has been developed which can provide unbiased estimates of the plant [Fa86]. This algorithm uses a criterion slightly different from the simplistic "looking down the MSE surface" approach to adapt the IIR filter of Figure 2.3. Since unbiased estimates are both a necessary and sufficient condition for minimum MSOE in the output error model, this algorithm therefore retains the desirable output error model property of simultaneous output error minimization and unbiased estimates, but does not get stuck in local minima. This algorithm will be studied in more detail in Chapter 3.

2.3.3 Stability Considerations

Referring to the equation error structure of Figure 2.2, it is seen that this method of system identification requires two FIR filters for its implementation. Adaptive algorithms perform particularly well when adapting FIR filters, with respect to speed of convergence to the parameter estimate, $\hat{\theta}$, yielding the minimum MSEE.

The output error model, on the other hand, requires an IIR adaptive filter, as shown in Figure 2.3. The fact that IIR filters have poles as well as zeros imposes a stability constraint on the $\hat{A}(q^{-1},n)$ polynomial. This requires that every new estimate $\hat{\theta}(n)$, generated by the adaptive algorithm be checked to see if the resulting $\hat{A}(q^{-1},n)$ is stable (i.e. it has all its roots inside the unit circle). Factoring the $\hat{A}(q^{-1},n)$ polynomial to determine its roots is a major computational burden when $\hat{n}_a \geq 5$, however, methods have been devised [Ju64] which can check the roots of a discrete-time polynomial for the presence of unstable roots, similar to the Routh-Hurwitz test for continuous time polynomials. This eliminates the need to factor $\hat{A}(q^{-1},n)$. If it is determined to possess unstable roots, then the unstable estimate,

$$\hat{\theta}_{unst}(n) = \hat{\theta}(n-1) + \Delta\hat{\theta}(n)$$

should be replaced with the stable estimate

$$\hat{\theta}_{st}(n) = \hat{\theta}(n-1) + \rho(n)\Delta\hat{\theta}(n) \quad \text{where } 0 \leq \rho(n) < 1 \tag{2.22}$$

This process is known as *stability projection* [Lj83, Sec.6.6]. Note that the choice of $\rho(n)=0$ corresponds to effectively "throwing out" the unstable estimate, $\hat{\theta}(n)$, by setting $\hat{\theta}(n)=\hat{\theta}(n-1)$.

An example of the process of stability projection for the case of $\hat{n}_a=2$ is shown in Figure 2.4. It is shown in [Ha86, Sec.2.8] that stability is maintained if and only if the

point $(\hat{a}_1,\hat{a}_2)$ lies in the triangular region shown. Given the previous estimate, $\hat{\theta}(n-1)$, the current estimate as generated by the adaptive algorithm, $\hat{\theta}_{unst}(n)$, is shown to lie outside of the stability region. To generate the estimate $\hat{\theta}_{st}(n)$, $\Delta\hat{\theta}(n)$ is repeatedly multiplied by a small constant, μ, $0\leq\mu<1$, until $\hat{\theta}(n)$ lies inside the triangular stability region. This yields a value for $\rho(n)$ in (2.22) of $\rho(n)=\mu^p$, where p is the number of times that $\Delta\hat{\theta}_{unst}(n)$ had to be multiplied by μ. Typically, the choice $\mu=0.5$ works well. In the example in Figure 2.4, it is seen that after multiplying $\Delta\theta(n)$ by μ, the resulting parameter, $\theta_1(n)$ is still unstable. Multiplying again by μ yields the final stable estimate, $\theta_{st}(n)$.

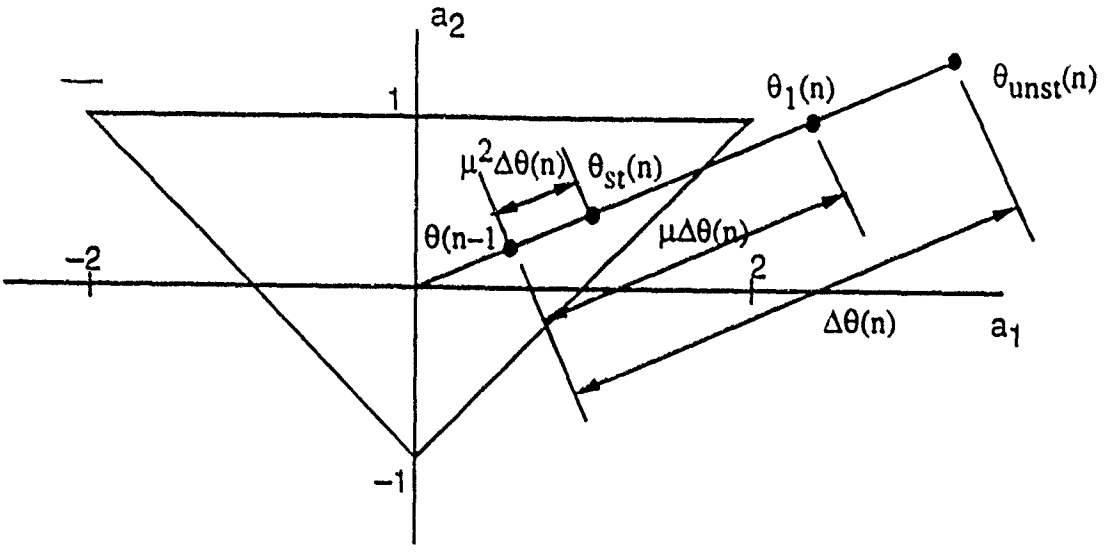


Figure 2.4 An example of stability projection for $\hat{n}_a=2$

2.4 Summary

This chapter has illustrated in detail two very common examples of the two-step modelling process of system identification as described in Chapter 1. The equation-error model and the output-error model. The first step of selecting the plant structure was the same for both models. In particular, an ARMAX plant structure with $C(q^{-1})=A(q^{-1})$ was selected. This structure corresponds to a plant modelled as having a rational transfer

function with its output corrupted by additive white measurement noise (Figure 2.1). The second step - selecting the adaptive filter structure - is what distinguishes the output-error model from the equation-error model

The equation-error model simply uses the observable plant signals x(n) and $\hat{y}$(n) as inputs to seperate FIR filters, generating the adaptive filter output, $\hat{y}$(n), as in (2.2). This yields an X structure for the adaptive filter. The error signal for this model, ee(n)=y(n)–$\hat{y}$(n), is shown to be linear in the parameters, $\hat{\theta}$, yielding a quadratic MSE surface. this type of MSE surface is very desirable because it is mathematically well-behaved and does not contain local minima, as does the output-error MSE surface. However, the price paid for the filter simplicity and quadratic error surface is biased estimates, shown in (2.20), and minimum MSE which is greater than the measurement noise variance, σ_v^2, shown in (2.21).

The output-error model employs an IIR adaptive filter, yielding an ARX adaptive filter structure. This model has the ability to simultaneously provide unbiased estimates and optimal minimum MSE of σ_v^2. However, the tradeoff here is its MSE surface is highly nonlinear and can have local minima. Furthermore, the adaptive IIR filter must always be checked for stability before proceeding after a parameter update. This introduces the additional computational burdens of stability determination and projection.

CHAPTER 3

ADAPTIVE ALGORITHMS

In this chapter, recursive algorithms are developed in Sections 3.1 and 3.2 which implement the process of "looking down" the MSE surface, as described in Section 2.3.2. In Sections 3.3 and 3.4, algorithms are presented which use different criterion to identify the plant parameters, θ_p.

3.1 Gradient-Based Methods

The simplest approach to recursively find a minimum point of a surface is called the *steepest descent* algorithm. This method is described by the following 3-step procedure:

1) Locate the direction at which the surface is most rapidly descending from the last parameter estimate, $\hat{\theta}(n-1)$.

2) Choose the current estimate $\hat{\theta}(n)$ as the estimate resulting from taking a small step away from $\hat{\theta}(n-1)$ in the direction determined in step 1).

3) Go back to step 1).

Mathematically, the direction of step 1) above is related to the *gradient* of the MSE surface. Consider a function $f:\mathbb{R}^n \rightarrow \mathbb{R}$. The gradient of f at a point $x \in \mathbb{R}^n$, denoted $\nabla f(x)$, is a generalization of the derivative of a function of one variable, and is defined as:

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1} \quad \frac{\partial f(x)}{\partial x_2} \quad \cdots \quad \frac{\partial f(x)}{\partial x_n} \right]^T$$

Given a point x_0 on the f(x) versus x surface, the direction in which the slope is maximum is precisely the direction of the gradient vector. Furthermore, the direction of minimum slope is opposite to the direction of the gradient vector [Fl77, Sec.3.5].

The gradient of the MSE surface, denoted by $\nabla E\left[e^2(n)\right]$ (recall that $\nabla E\left[e^2(n)\right]$ is a function of $\hat{\theta}$), is thus defined as the vector of its partial derivatives with respect to the parameters $\hat{a}_i$, i=1, $\cdots$, $\hat{n}_a$, and $\hat{b}_i$, i=0, $\cdots$, $\hat{n}_b$ as follows:

$$\nabla E\left[e^2(n)\right] = \left[\frac{\partial E\left[e^2(n)\right]}{\partial \hat{a}_1} \quad \cdots \quad \frac{\partial E\left[e^2(n)\right]}{\partial \hat{a}_{\hat{n}_a}} \quad \frac{\partial E\left[e^2(n)\right]}{\partial \hat{b}_0} \quad \cdots \quad \frac{\partial E\left[e^2(n)\right]}{\partial \hat{b}_{\hat{n}_b}}\right]^T$$

Since the direction of minimum slope is $-\nabla E\left[e^2(n)\right]$, the steepest descent method can thus be expressed recursively as:

$$\hat{\theta}(n) = \hat{\theta}(n-1) - \mu \nabla E\left[e^2(n)\right] \tag{3.1}$$

where μ is a small stepsize, which determines if and how fast the algorithm will converge. It is shown in [Ha86,Sec.5.4], for the equation error adaptive system, that the steepest descent algorithm will converge if $0<\mu<2/\lambda_{max}$, where λ_{max} is the largest eigenvalue of the correlation matrix $\mathbf{R} = E\left[\varphi_{ee}(n)\varphi_{ee}^T(n)\right]$. Also, in general for gradient-type algorithms, the rate of convergence is proportional to μ.

Recall from Chapter 1, however, that adaptive filtering applications are precisely those in which the environment of the adaptive filter (i.e. the plant in the case of system identification) is unknown and/or changing. This makes taking expectations difficult if not impossible. Therefore, in order to design a practical algorithm, the expectation operator must be dispensed with, yielding approximate or instantaneous gradient methods. There is also an important theoretical justification for doing this: $\nabla\left[e^2(n)\right]$ is by definition an unbiased estimate of $\nabla E\left[e^2(n)\right]$. In the literature, these methods are often referred to as *stochastic gradient* methods [Ha86,Sec 5.3]. These algorithms thus have the form:

$$\hat{\theta}(n) = \hat{\theta}(n-1) - \mu\nabla\left[e^2(n)\right] \tag{3.2}$$

In order to implement (3.2), the gradient must be determined. The gradient will have a different form depending on which system identification model is used (i.e. output error or equation error). These two gradient expressions are now derived.

3.1.1 The Equation Error Stochastic Gradient Algorithm

To find $\nabla\left[ee^2(n)\right]$, the chain rule[Fl77, Sec.4.4] is first applied:

$$\nabla\left[ee^2(n)\right] = 2ee(n)\nabla[ee(n)] \tag{3.3}$$

Recall:

$$ee(n) = y(n) - \hat{y}(n) = y(n) - \hat{\theta}^T(n-1)\varphi_{ee}(n)$$

Taking derivatives with respect to the parameters, $\hat{\theta}(n-1)$, noting $y(n)$ is independent of $\hat{\theta}(n-1)$ and thus $\nabla y(n)=0$, the following expression for the gradient of $ee(n)$ is obtained:

$$\nabla ee(n) = -\varphi_{ee}(n) \tag{3.4}$$

Substituting (3.4) into (3.3) and then into (3.2) yields the following expression for the equation error stochastic gradient algorithm. This algorithm is known as the LMS algorithm, which was developed by Widrow and Hoff [Wi75], [Wi76], [Ha86,Ch5], [Wi85,Ch6]:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \mu\varphi_{ee}(n)ee(n) \tag{3.5}$$

Note the constant "2" has been absorbed into the stepsize, μ. Convergence requirements for this algorithm are similar to those of the steepest descent method. In particular [Ha86,Sec.5.12,prop.2], for mean-square convergence of the parameters, $\hat{\theta}(n-1)$:

$$0<\mu<\frac{2}{\sum\lambda_i}=\frac{2}{Tr(R)}=\frac{2}{\text{Input Power}}$$

Where, as before, the λ_i's are the eigenvalues of the correlation matrix, R.

The LMS algorithm is the most widely used adaptive algorithm because of its computational simplicity. Furthermore, since it uses the mean-squared equation error, it possesses the desirable characteristics of the equation error model discussed in Section 2.3 (i.e. unimodal error surface and no adaptive filter stability problems). Thus, even though the equation error modes does not provide optimal MMSE and estimates of θ_p, it is dependable and does in fact perform satisfactorily in a wide variety of applications[Wi75].

3.1.2 The Output Error Stochastic Gradient Algorithm

Proceeding similarly as in the previous section, the gradient of $oe^2(n)$ must be determined. As before, the chain rule yields:

$$\nabla oe^2(n) = 2oe(n)\nabla oe(n)$$

The matrix expression for the output error is:

$$oe(n) = y(n) - \hat{y}(n) = y(n) - \hat{\theta}^T(n-1)\varphi_{oe}(n)$$

As before, $\nabla y(n)=0$, so the expression for the gradient is:

$$\nabla oe(n) = -\nabla\hat{\theta}^T(n-1)\varphi_{oe}(n)$$

This gradient cannot be evaluated as simply as in (3.4) in the equation error case, because the $\hat{y}$ terms in φ_{oe} depend on the parameters, $\hat{\theta}$. Expanding the matrix notation yields:

$$-\hat{\theta}^T(n-1)\varphi_{oe}(n) = \hat{a}_1(n-1)\hat{y}(n-1) + \cdots + \hat{a}_{\hat{n}_a}(n-1)\hat{y}(n-\hat{n}_a)$$

$$- \hat{b}_0(n-1)x(n) - \cdots - \hat{b}_{\hat{n}_b}(n-1)x(n-\hat{n}_b)$$

In the following, the time index, (n–1), associated with the adaptive filter parameters $\hat{a}_i$ and $\hat{b}_i$ will be omitted for convenience. The appropriate derivatives for the gradient vector must now be taken, using the chain rule since $\hat{y}(n)$ is not a constant with respect to the parameters [Jo84,Sec.III.A]:

$$\frac{-\partial\hat{\theta}^T\varphi_{oe}(n)}{\partial\hat{a}_i} = \hat{a}_1\frac{\partial\hat{y}(n-1)}{\partial\hat{a}_i} + \cdots + \hat{a}_i\frac{\partial\hat{y}(n-i)}{\partial\hat{a}_i} + \hat{y}(n-i) - \cdots - \hat{a}_{\hat{n}_a}\frac{\partial\hat{y}(n-\hat{n}_a)}{\partial\hat{a}_i}$$

$$= \sum_{j=1}^{\hat{n}_a}\hat{a}_j\frac{\partial\hat{y}(n-i)}{\partial\hat{a}_i} + \hat{y}(n-i)$$

$$= \left[\hat{A}(q^{-1})-1\right]\frac{\partial\hat{y}(n)}{\partial\hat{a}_i} + \hat{y}(n-i) \qquad (3.6)$$

$$\frac{\partial\hat{\theta}^T\varphi_{oe}(n)}{\partial\hat{b}_i} = \hat{a}_1\frac{\partial\hat{y}(n-1)}{\partial\hat{b}_i} + \cdots + \hat{a}_{\hat{n}_a}\frac{\partial\hat{y}(n-\hat{n}_b)}{\partial\hat{b}_i} - x(n-i)$$

$$= \sum_{j=1}^{\hat{n}_a}\hat{a}_j\frac{\partial\hat{y}(n-i)}{\partial\hat{a}_i} - x(n-i)$$

$$= \left[\hat{A}(q^{-1})-1\right]\frac{\partial\hat{y}(n)}{\partial\hat{b}_i} - x(n-i) \qquad (3.7)$$

From the expressions (3.6) and (3.7), it is seen that:

$$\nabla oe(n) = -\nabla\hat{\theta}^T\varphi_{oe}(n)$$

$$= -\varphi_{oe}(n) +$$

$$\left[\hat{A}(q^{-1})-1\right]\left[\frac{\partial\hat{y}(n)}{\partial\hat{a}_1}\cdots\frac{\partial\hat{y}(n)}{\partial\hat{a}_{\hat{n}_a}}\frac{\partial\hat{y}(n)}{\partial\hat{b}_0}\cdots\frac{\partial\hat{y}(n)}{\partial\hat{b}_{\hat{n}_b}}\right]^T \qquad (3.8)$$

Since $oe(n)=y(n)-\hat{y}(n)$, $\nabla oe(n)=-\nabla\hat{y}(n)$, which is the negative of the vector in second term on the right of (3.8). Using this fact, (3.8) can be rewritten as:

$$\nabla oe(n) = -\varphi_{oe}(n) + \left[1-\hat{A}\left(q^{-1}\right)\right]\nabla oe(n)$$

The above equation can be simplified as:

$$\hat{A}\left(q^{-1}\right)\nabla oe(n) = -\varphi_{oe}(n)$$

Reintroducing the time index and solving for the gradient vector yields:

$$\nabla oe(n) = \frac{-1}{\hat{A}\left(q^{-1},n-1\right)}\varphi_{oe}(n)^{\dagger} \tag{3.9}$$

Define the vector:

$$\varphi'_{oe}(n) = \frac{1}{\hat{A}\left(q^{-1},n-1\right)}\varphi_{oe}(n)$$

This yields the following expression for the stochastic gradient algorithm for the output error adaptive system:

$$\theta(n) = \theta(n-1) + \mu\varphi'_{oe}(n)oe(n) \tag{3.10}$$

It is important to note a significant difference between the output error algorithm of (3.10) and the equation error algorithm of (3.5). The algorithm of (3.5) uses the equation error regressor vector, $\varphi_{ee}(n)$, "as is," whereas the algorithm of (3.10) requires the output error regressor vector, $\varphi_{oe}(n)$, to be filtered by the autoregressive polynomial of the adaptive filter. This characteristic of regressor filtering is very common among algorithms which have been developed for adaptive IIR filters [Jo84], [Fa86], [So88]. Note that the autoregressive filtering by $\hat{A}\left(q^{-1},n-1\right)$ in (3.10) is applied to a *vector*. This implies that $\hat{n}_a$ past values of $\varphi_{oe}(n)$, which is a total of $\hat{n}_a(\hat{n}_a+\hat{n}_b+1)$ values, be retained in memory to

---

† This result can be arrived at more simply by considering the output error expression (2.11): $oe(n) = \frac{1}{A\left(q^{-1}\right)}\tilde{\theta}(n-1)\varphi_{oe}(n)+v(n)$. Taking derivatives with respect to $\hat{\theta}(n-1)$ yields $\nabla oe(n)=\frac{-1}{A\left(q^{-1}\right)}\varphi_{oe}(n)$. Since $A\left(q^{-1}\right)$ is unknown, the best that can be done is to use its latest estimate $\hat{A}\left(q^{-1},n-1\right)$. This substitution yields (3.7"). This is a common practical way of dealing with filtered quantities and will be used later.

accomplish this operation. Assuming slowly time varying adaptive filter parameters, this memory requirement can be reduced by implementing the filtering process in the following approximate manner [Jo84,Sec.III.4], [So88,Remark 1] by defining the quantities:

$$\hat{y}^F(n) = \frac{1}{\hat{A}(q^{-1},n-1)}\hat{y}(n)$$

$$x^F(n) = \frac{1}{\hat{A}(q^{-1},n-1)}x(n)$$

The following approximation can now be used in (3.10):

$$\frac{1}{\hat{A}(q^{-1},n-1)}\varphi_{oe}(n) \approx \left[\hat{y}^F(n-1) \cdots \hat{y}^F(n-\hat{n}_a)\ x^F(n) \cdots x^F(n-\hat{n}_b)\right]^T$$

This approximate filtered-regressor expression requires that only $(\hat{n}_a+\hat{n}_b+1)$ values of $\hat{y}(n)$ and $x(n)$ be stored in memory.

As mentioned in Section 2.3, the possible multimodality of the MSOE surface is the major drawback of the output error model, because this can cause (3.10) to stall in a local minimum. This fact can be seen more precisely now in terms of the gradient vector. At any local minimum point, the partial derivatives with respect to every variable are zero. Thus $\nabla E\left[oe^2(n)\right] = 0$ and it can be seen from (3.1) that $\tilde{\theta}(n)=\hat{\theta}(n-1)$. In other words, gradient algorithms effectively "turn off" at local minima. Note that in addition to "turning off" at local minima, gradient algorithms will "turn off" at local maxima and inflection points as well. However, using the instantaneous gradient of (3.2) prevents the second term of the algorithm from staying at **0**. Therefore, at local maxima and inflection points, the noisy gradient estimate will always perturb the parameter estimates, $\hat{\theta}$, slightly past these points, and the algorithm will continue "looking down" until in reaches a minimum point (global or local). The points of minimum MSE are referred to in literature dealing with convergence issues as *stable* [Mi82,Sec.5.3] convergence points of the gradient algorithm (3.1). When the practical stochastic gradient algorithm (3.10) yields parameter

estimates near these points, they will oscillate about them because of the noisy gradient estimates instead of moving away as in the case of local maxima and inflection points.

3.2 Least Squares Methods

In the least squares (LS) scheme, it is desired to do more than merely move the estimate, $\hat{\theta}$, in the direction of the minimum of the mean-square error surface. In particular, the least squares estimate is the one which minimizes at every time instant, n, the following criterion:

$$J_{LS}(n) = \sum_{i=1}^{n} e^2(i) \tag{3.11}$$

This can be explained more intuitively as follows [Ha86, Ch.7]. Given some plant input/output sequence $\{x(i)\}_{i=1}^{n}$, $\{y(i)\}_{i=1}^{n}$, and a *constant* parameter estimate $\hat{\theta}(n)$, the output sequence of the adaptive filter, $\{\hat{y}(n)\}_{i=1}^{n}$, will produce the error sequence $\{y(i)-\hat{y}(i)\}_{i=1}^{n}$. This error sequence will produce a corresponding value of J(n). The LS estimate, $\hat{\theta}_{LS}(n)$, is the one which, when held constant through the interval i=1, $\cdots$,n, as above, yields the minimum value of J(n). The least squares method is seen to be a deterministic method, in that no statistical assumptions or approximations have been made, as in the gradient methods.† Minimization thus takes place assuming only the plant input/output record from the initial time up to the current time.

Consider the minimization with respect to $\hat{\theta}(n)$ of the criterion (3.11), repeated here:

---

† It should be noted here that minimization of $\sum_{i=1}^{n} e^2(i)$ is equivalent to minimizing $\frac{1}{n}\sum_{i=1}^{n} e^2(i)$, which by the law of large numbers approaches $E\left[e^2(n)\right]$ as $n \to \infty$. Therefore, statistical methods based on the MSE surface are asymptotically equivalent to least squares methods. In other words, both types of algorithms will converge to the same point.

$$J_{LS}(n) = \sum_{i=1}^{n} e^2(i)$$

The chain rule yields the derivative:

$$\frac{dJ_{LS}(n)}{d\hat{\theta}(n)} = 2\sum_{i=1}^{n} e(i)\frac{de(i)}{d\hat{\theta}(n)} \tag{3.12}$$

Setting this derivative to zero and solving for $\hat{\theta}(n)$ yields the LS estimate based on n observations, $\hat{\theta}_{LS}(n)$. To determine its value, the explicit expression for the error, e(i), must be used. Thus in a similar fashion as in Section 3.1, both the equation error and output error expressions will be utilized in (3.12) to derive adaptive algorithms for the equation error and output error adaptive systems.

3.2.1 The Equation Error Least Squares Algorithm

Using the matrix expression (2.3) for the output of the adaptive filter yields the desired expression for the equation error at the i^{th} iteration:

$$ee(i) = y(i) - \hat{y}(i) = y(i) - \hat{\theta}^T(n)\varphi_{ee}(i)$$

As seen in Section 3.1.1, the derivative of this quantity is:

$$\frac{dee(i)}{d\hat{\theta}(n)} = \nabla^T ee(i) = -\varphi_{ee}^T(i)$$

Substituting these expressions into (3.12) and equating to zero yields the least squares estimate of θ_p based on n observations, $\hat{\theta}_{LS}(n)$:

$$\sum_{i=1}^{n} \left[y(i) - \hat{\theta}_{LS}^T(n)\varphi_{ee}(i) \right] \varphi_{ee}^T(i) = 0$$

Solving for $\hat{\theta}_{LS}(n)$ yields:

$$\hat{\theta}_{LS}^{T}(n)\sum_{i=1}^{n}\varphi_{ee}(i)\varphi_{ce}^{T}(i) = \sum_{i=1}^{n}y(i)\varphi_{ce}^{T}(i) \tag{3.13}$$

$$\hat{\theta}_{LS}(n) = \left[\sum_{i=1}^{n}\varphi_{ee}(i)\varphi_{ce}^{T}(i)\right]^{-1}\sum_{i=1}^{n}y(i)\varphi_{ee}(i) \tag{3.14}$$

Even though (3.14) is a valid expression for the LS estimate, note that it is not recursive, that is, (3.14) does not express $\hat{\theta}_{LS}(n)$ in terms of only the data at the previous time instant. In other words, it requires all the data from the starting time to the current time.

It is possible to put (3.14) in recursive form as follows [Lj83,Sec.2.2.1] by defining the term which is inverted in (3.14) as:

$$R(n) = \sum_{i=1}^{n}\varphi_{ee}(i)\varphi_{ee}^{T}(i) = R(n-1) + \varphi_{ee}(n)\varphi_{ee}^{T}(n) \tag{3.15}$$

This definition used in (3.13) and transposing yields:

$$R(n)\hat{\theta}_{LS}(n) = \sum_{i=1}^{n}y(i)\varphi_{ee}(i) \tag{3.16}$$

The summation above can be expanded to give:

$$R(n)\hat{\theta}_{LS}(n) = \sum_{i=1}^{n-1}y(i)\varphi_{ee}(i) + y(n)\varphi_{ee}(n)$$

Applying (3.16) to the first term on the right above yields:

$$R(n)\hat{\theta}_{LS}(n) = R(n-1)\hat{\theta}_{LS}(n-1) + y(n)\varphi_{ee}(n)$$

Now solving (3.15) for R(n–1) and substituting in the above expression yields:

$$R(n)\hat{\theta}_{LS}(n) = \left[R(n)-\varphi_{ee}(n)\varphi_{ee}^{T}(n)\right]\hat{\theta}_{LS}(n-1) + y(n)\varphi_{ee}(n)$$

$$= \mathbf{R}(n)\hat{\theta}_{LS}(n-1) + \varphi_{ee}(n)\left[-\varphi_{ee}^{T}(n)\hat{\theta}_{LS}(n-1)+y(n)\right]$$

Finally, premultiplying by $\mathbf{R}^{-1}(n)$ and recognizing that the term in brackets is the equation error, $e_e(n)$, gives the desired result:

$$\hat{\theta}_{LS}(n) = \hat{\theta}_{LS}(n-1) + \mathbf{R}^{-1}(n)\varphi_{ee}(n)ee(n) \tag{3.17}$$

The equations (3.17) and (3.15) constitute a recursive version of (3.14). However, notice that the matrix $\mathbf{R}(n)$ must be inverted at each iteration, which is very time consuming. This problem can be alleviated by defining the matrix:

$$\mathbf{P}(n) = \mathbf{R}^{-1}(n)$$

The matrix inversion lemma can be used here to establish the following recursion to update $\mathbf{P}(n)$ [Lj83,p.19]:

$$\mathbf{P}(n) = \mathbf{P}(n-1) - \frac{\mathbf{P}(n-1)\varphi_{ee}(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)}{1+\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)} \tag{3.18}$$

This expression allows the inverted matrix, $\mathbf{R}^{-1}(n)$, to be updated directly, rather than first calculating $\mathbf{R}(n)$ using (3.15) and then performing the matrix inversion. Note that (3.18) eliminates the need for matrix inversion altogether, as it requires only a single scalar division. The expressions (3.18) and (3.17) with $\mathbf{R}^{-1}(n)=\mathbf{P}(n)$ constitute what is known as the recursive least squares (RLS) algorithm.

Weighted Recursive Least Squares (WRLS)

In applications, it is often desirable to assign weights to the individual observations of the least squares parameter estimation problem. Weighting an observation can indicate some measure of its importance, accuracy, or relevence in determining the new parameter estimate, $\hat{\theta}(n)$. The particular choice of weighting assignments depends on the application.

and can have either a heuristic or analytic basis. The criterion function (3.11), modified to reflect weighted observations, is now:

$$J_{WLS}(n) = \sum_{i=1}^{n} \alpha(i) ee^2(i) \tag{3.19}$$

where $\{\alpha(i)\}_{i=1}^{n}$ is the sequence of observation weights.

Carrying through the procedure of minimization of (3.19), and generating a recursive formulation as done previously yields the following modified least squares algorithm, commonly known as *weighted least squares* (WLS):

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \alpha(n)\mathbf{P}(n)\varphi_{ee}(n)ee(n)$$

$$\mathbf{P}(n) = \mathbf{P}(n-1) - \frac{\alpha(n)\mathbf{P}(n-1)\varphi_{ee}(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)}{1+\alpha(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)}$$

Recursive Least Squares with Forgetting Factor (RLSFF)

A particular weighting scheme which assigns progressively lower weights to past observations is useful in dealing with time varying systems. This gives the least squares algorithm the characteristic of "forgetting" data from the distant past which may not be relavent in determining the current "optimal" value of the parameter estimates. Note that in this scheme, a given observation will be systematically be assigned smaller and smaller weights as the time index, n, increases. This is in contrast to WRLS, which assigns a *constant* weight to each observation. The criterion function for this weighting scheme is thus [Lj83.Sec.2.6.2]:

$$J_{RLSFF}(n) = \sum_{i=1}^{n} \beta(i,n) ee^2(i) \tag{3.20}$$

where the weighting sequence, $\{\beta(i,n)\}_{i=1}^{n}$, is increasing in the variable i and decreasing in the variable n. A typical choice for this weighting sequence is:

$$\beta(i,n) = \lambda^{n-i}$$

where λ, called the *forgetting factor*, is constant, and $0<\lambda\leq 1$. This choice of $\beta(i,n)$ is seen to yield an exponential forgetting characteristic when considered as a function of the time variable, n. A forgetting factor, λ, yields an algorithm with an effective "memory" of the past $1/(1-\lambda)$ observations [Sh89].

Allowing the RLS algorithm to "forget" past observations in this manner produces an increase in the "bouncing around" of the estimates, $\hat{\theta}(n)$, about their target values in the steady state, resulting in a higher value of MMSE. This is because useful past data is effectively "thrown out," leaving the algorithm more susceptible to the noise contribution in fewer observations. This characteristic illustrates a general tradeoff which exists in most adaptive algorithms between tracking ability and MMSE.

As before, carrying through the minimization of (3.20) and derivation of a recursive algorithm yields the following algorithm, known as *recursive least squares with forgetting factor* (RLSFF):

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \mathbf{P}(n)\varphi_{ee}(n)ee(n)$$

$$\mathbf{P}(n) = \mathbf{P}(n-1) - \frac{1}{\lambda}\left[\frac{\mathbf{P}(n-1)\varphi_{ee}(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)}{\lambda+\alpha(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)}\right]$$

Weighted Recursive Least Squares with Forgetting Factor (WRLSFF)

The three previously presented least squares algorithms can be lumped into one algorithm by employing both weighting schemes simultaneously. This algorithm is called *weighted recursive least squares with forgetting factor* (WRLSFF):

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \alpha(n)\mathbf{P}(n)\varphi_{ee}(n)ee(n) \tag{3.21a}$$

$$\mathbf{P}(n) = \frac{1}{\lambda}\left[\mathbf{P}(n-1) - \frac{\alpha(n)\mathbf{P}(n-1)\varphi_{ee}(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)}{\lambda+\alpha(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)}\right] \tag{3.21b}$$

This algorithm obviously includes RLS, WRLS, and RLSFF as special cases with appropriate choices for $\alpha(n)$ and λ.

As a computational issue, note that all the least squares algorithms given previously require that $\mathbf{P}(n-1)\varphi_{ee}(n)$ be evaluated to determine $\mathbf{P}(n)$ and then subsequently evaluating $\mathbf{P}(n)\varphi_{ee}(n)$ to update $\hat{\theta}(n)$. This latter matrix multiplication can be avoided by manipulating the $\mathbf{P}(n)$ update equation (3.21b) by postmultiplying both sides by $\varphi_{ee}(n)$ and expressing the term in brackets over a common denominator:

$$\begin{aligned}\mathbf{P}(n)\varphi_{ee}(n) = \frac{1}{\lambda}\Big[&\lambda\mathbf{P}(n-1)\varphi_{ee}(n)\\ &+\alpha(n)\mathbf{P}(n-1)\varphi_{ee}(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)\\ &-\alpha(n)\mathbf{P}(n-1)\varphi_{ee}(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)\Big]\Big/\left(\lambda+\alpha(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)\right)\\ &= \frac{\mathbf{P}(n-1)\varphi_{ee}(n)}{\lambda+\alpha(n)\varphi_{ee}^{T}(n)\mathbf{P}(n-1)\varphi_{ee}(n)}\end{aligned}$$

Using this relationship, the least squares recursion of (3.21) can be implemented in the following four-step procedure:

1) Calculate $\mathbf{M}(n) = \mathbf{P}(n-1)\varphi_{ee}(n)$ (3.22a)

2) Calculate $\mathbf{L}(n) = \dfrac{\alpha(n)\mathbf{M}(n)}{\lambda+\alpha(n)\varphi_{ee}^{T}(n)\mathbf{M}(n)}$ (3.22b)

3) $\hat{\theta}(n) = \hat{\theta}(n-1) + L(n)ee(n)$ (3.22c)

4) $P(n) = \frac{1}{\lambda}\left[P(n-1) - L(n)M^T(n)\right]$ (3.22d)

3.2.2 The Output Error Least Squares Algorithm (RPE)

Unfortunately, analytic minimization of the criterion (3.11) when e(n)=oe(n) turns out to be impossible for a recursive algorithm. This is because of the highly nonlinear relationship between the parameter estimates, $\hat{\theta}$, and the corresponding adaptive filter output, $\hat{y}(n)=\hat{\theta}^T\varphi_{ee}(n)$. The difficulty lies in the problem of solving (3.12) set equal to zero for the least squares estimate, $\hat{\theta}_{LS}(n)$. Recall for the equation error case, e(i) is linear in $\hat{\theta}$ and thus $de(i)/d\hat{\theta}$ is constant. This gives rise to the estimate $\hat{\theta}_{LS}(n)$, which is the solution to a linear (matrix) equation. The nonlinear equation resulting when using the output error model cannot be solved so simply. Instead, numerical methods must be utilized to minimize (3.11). This procedure usually requires several evaluations of (3.11) which uses all of the data since the initial time, n=1. Therefore, a recursive algorithm based on this type of minimization is not possible.

It is thus necessary to introduce approximations in the quest for a recursive algorithm in order to attempt to minimize (3.11). A very general method is presented in [Lj83,Sec.3.7.2], which accomodates a plant having the structure of an extended form of the Box-Jenkins model, which was briefly mentioned in Chapter 1. Algorithms such as this which can approximately minimize the least squares criterion for plant structures that are more complicated than ARX are known as *recursive prediction error* (RPE) methods.

The problem at hand is to minimize (3.11) using the output error, oe(n), as the error term, e(n). The output error should be thought of here more generally as the error between an adaptive filter and a plant having the ARMAX structure of Figure 2.1. Notice that no

mention has been made here of the structure of the adaptive filter. This is because the general RPE method specifies the structure of the optimal adaptive filter based on the plant structure. It turns out that the ARX adaptive filter of the output error adaptive system, shown in Figure 2.3, is the optimal adaptive filter structure for use with given ARMAX plant structure. This should not be surprising, given the investigation and discussion of the properties of the output error model given in Chapter 2 (i.e. unbiased estimates and minimum possible MMSE of σ_v^2).

The RPE algorithm is now shown here in the context of the output error adaptive system of Figure 2.3:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \mathbf{P}(n)\psi(n)oe(n) \tag{3.23a}$$

$$\mathbf{P}(n) = \frac{1}{\lambda}\left[\mathbf{P}(n-1) - \frac{\alpha(n)\mathbf{P}(n-1)\psi(n)\psi^T(n)\mathbf{P}(n-1)}{\lambda+\psi^T(n)\mathbf{P}(n-1)\psi(n)}\right] \tag{3.23b}$$

where

$$\psi(n) = -\nabla oe(n) = \frac{1}{\hat{A}(q^{-1},n-1)}\varphi_{oe}(n)$$

The forgetting factor, λ, and observation weighting coefficient, $\alpha(n)$, have also been included here and serve the same purpose as in WRLSFF. Note the striking similarity between the RPE algorithm and WRLSFF. In fact, the RPE algorithm reduces to WRLSFF when the equation error quantities, ee(n) and $\varphi_{ee}(n)$, are used in place of the corresponding output error quantities, oe(n) and $\varphi_{oe}(n)$. Also note that (3.23a) reduces to the gradient methods of (3.5) and (3.10) for the equation error and output error cases, respectively, when $\mathbf{P}(n)\equiv\mu\mathbf{I}$. These are some examples of the generality and wide applicablilty of the RPE algorithm. It should also be noted that (3.22) can obviously be used to implement the recursion (3.23) with the appropriate changes of the notation.

3.3 The Method of Optimal Bounding Ellipsoids (OBE)

Recently, a system identification technique providing an alternative to least squares equation error minimization has been proposed [Fo82], [Da87]. In these papers, algorithms were developed which perform optimization based on geometric considerations rather than the analytic minimization of (3.11). What sets these algorithms apart from least squares methods is the manner in which the contribution to the plant output of the noise is characterized. Instead of imposing some statistical assumptions on the noise (i.e. white noise), as has been the case thus far, these algorithms were developed assuming that the noise contribution in the plant description has a magnitude bound, γ, at every time instant, n. This key assumption gives rise to an algorithm having the ability to decide very quickly whether the current data observation, $\varphi_{ee}(n)$, yields any additional information which could improve upon the previous estimate, $\hat{\theta}(n-1)$. If it is determined that $\varphi_{ee}(n)$ contains no new information, updating (a computationally expensive operation) need not occur. In other words, this algorithm possesses the very attractive characteristic of *selective updating*. This feature has been seen in simulations to reduce the amount of computation considerably, as the algorithm tends to use less than twenty percent of the input data [Hu86].

The geometric criterion used in the OBE algorithms is described using the concept of a membership set. A membership set is the set of points in the parameter space which are consistant with the data observations, assumed plant structure, and the noise bound. Initially, the membership set is the set of all points in the $(\hat{n}_a+\hat{n}_b+1)$-dimensional parameter space. Practically, however, it is chosen to be a very large $(\hat{n}_a+\hat{n}_b+1)$-dimensional ellipsoid which must include all possible valid parameter values of the plant. Starting with this initial ellipsoid, the following algorithm is then implemented:

1) A check is made of each subsequent data observation to determine if the "size" of the previous ellipsoid can be reduced by utilizing the current data (regressor vector), $\varphi_{ee}(n)$.[†]

2a) If so, a new, smaller ellipsoid is then determined by the algorithm.

2b) If not, the current data is discarded, the previous ellipsoid is kept as the current ellipsoid, the next data observation is brought in, and the process is repeated by returning to step 1).

At each iteration, k, the corresponding parameter estimate, $\hat{\theta}(k)$, is taken to be the center of the ellipsoid generated so far from the current and previous iterations, $i=1,\cdots,k$.

The algorithm derived in [Da87] using $\sigma^2(n)$ as an optimization parameter, has a very familiar form. In fact, it is identical to WRLSFF of (3.21) with a data-dependent scheme of generating the weighting coefficients, $\alpha(n)$. A time-varying forgetting factor, $\lambda(n)$, is also employed, and is related to the weighting coefficient by:

$$\lambda(n) = 1 - \alpha(n)$$

Define the following quantities, as in [Da87]:

$$G(n) = \varphi_{ee}^{T}(n)P(n-1)\varphi_{ee}(n)$$

$$\beta(n) = \frac{\gamma^2 - \sigma^2(n-1)}{ee^2(n)}$$

$\zeta \in (0,1)$, a design parameter

The OBE algorithm is now presented:

[†] Various measures of the "size" of the ellipsoid have been employed. In [Fo82], the size was defined in two ways, each yielding a slightly different algorithm. The two measures of size were 1) the volume of the ellipsoid, and 2) the sum of the semi-axes of the ellipsoid. In [Da87], a more abstract minimization criterion, $\sigma^2(n)$ (note this has nothing to do with noise variance), was used. This was seen to yield a simpler algorithm which is essentially an implementation of WRLSFF, as will be seen.

while (not done) do begin

get the current data vector, $\varphi_{ee}(n)$

if $\gamma^2 \geq \sigma^2(n-1) + ee^2(n)$ then begin {no update needed}

$\sigma^2(n) = \sigma^2(n-1)$

$\hat{\theta}(n) = \hat{\theta}(n-1)$

end

else begin {determine the weighting coefficients, $\alpha(n)$, the new value of $\sigma^2(n)$, and update with WRLSFF}

$$\nu(n) = \begin{cases} \dfrac{1}{1-G(n)} \left\{ 1 - \sqrt{\dfrac{G(n)}{1+\beta(n)[G(n)-1]}} \right\}, & \text{if } \beta(n)[G(n)-1]+1>0 \\ \zeta, & \text{if } \beta(n)[G(n)-1]+1 \leq 0 \end{cases}$$

$\alpha(n) = \min(\zeta, \nu(n))$

$$\sigma^2(n) = [1-\alpha(n)]\sigma^2(n-1) + \alpha(n)\gamma^2 - \frac{\alpha(n)[1-\alpha(n)]ee^2(n)}{1-\alpha(n) + \alpha(n)G(n)}$$

implement (3.21) with $\lambda(n) = 1 - \alpha(n)$

end {if}

end {while }

In summary, the OBE algorithm possesses two key properties which could be very desirable in applications. They are:

1) The bounded noise assumption. Most algorithms employ statistical assumptions to characterize the noise contribution in the plant output (i.e.

white noise). In practice, however, satisfaction of these assumptions are both difficult to guarantee and verify. In practice, it is usually much easier to obtain a magnitude bound on the output noise.

2) The selective updating strategy. This frees the algorithm for about 70-80% of the total running time, which opens up (largely unexplored) possibilities for time-sharing the algorithm with multiple processes.

3.4 The Steiglitz-McBride Method (SMM)

As mentioned at the beginning of this chapter, both the gradient and least squares algorithms implement the process of "looking down" the MSE surface. In Section 2.3.2, it was described how local minima in the MSE surface of the output error adaptive system could cause this type of algorithm to fail. It is therefore of interest to examine alternative methods of system identification which do not require unimodality if the output error surface for convergence of the adaptive filter parameter estimates, $\hat{\theta}$, to the plant parameters, θ_p.

An adaptive algorithm was developed recently by Fan [Fa86] which minimizes a criterion first considered by Steiglitz and McBride. For *sufficient order* adaptive systems (i.e. $\hat{n}_a \geq n_a$ and $\hat{n}_b \geq n_b$), the SMM criterion has a unimodal character containing a global minimum which coincides with the global minimum of the MSOE surface. Simulation studies have also shown this to be true in some *reduced order* cases (i.e. $\hat{n}_a \leq n_a$ *or* $\hat{n}_b \leq n_b$). Reduced order adaptive systems are of extreme interest, since in most practical situations, the plant order, n_a, is unknown.[†] Furthermore, in many cases, the plant may not even be of the form $B(q^{-1}) / A(q^{-1})$ as has been assumed throughout this thesis. In this case, an

[†] In fact, reduced order systems can *cause* the existence of local minima in the MSOE surface. Cases have been documented [St81] of adaptive systems, possessing unimodal MSOE surfaces with a sufficient order adaptive filter, having multimodal MSOE surfaces when the sufficient order adaptive filter is replaced with one of reduced order.

adaptive filter of sufficient order would need an infinite-dimensional parameter vector, $\hat{\theta}$ (i.e. $\hat{n}_a \to \infty$ and/or $\hat{n}_b \to \infty$). Thus acceptable (hopefully optimal in some sense) reduced order performance is an important feature for any practical adaptive system to have.

The SMM scheme is presented here as follows. Consider the ARMAX plant of Figure 2.1, described by the relationship:

$$A(q^{-1})y(n) = B(q^{-1})x(n) + A(q^{-1})v(n)$$

If the quantities x(n), y(n), and v(n) are autoregressively filtered by the $A(q^{-1})$ polynomial, the above relationship can be expressed as:

$$A(q^{-1})y'(n) = B(q^{-1})x'(n) + v(n) \tag{3.24}$$

where

$$y'(n) = \frac{1}{A(q^{-1})}y(n)$$

$$x'(n) = \frac{1}{A(q^{-1})}x(n)$$

In what follows, it will be seen that minimization of the *eqation error* of the "primed" adaptive system having the plant described by (3.24) will be the goal of the SMM algorithms. This is the essence of the SMM approach. Minimization will be accomplished using both the gradient and least squares techniques.

3.4.1 Gradient Minimization

Observe that (3.24) describes an ARX plant with input x'(n) and output y'(n). As mentioned in Section 2.3.1, the equation error adaptive system will simultaneously provide minimum MSE of σ_v^2 and unbiased estimates for ARX plant structures. Therefore, one might expect the algorithms presented thus far for equation error systems to perform

optimally for the "primed" equation error adaptive system having a plant described by (3.24), with input x'(n) and output y'(n). In particular, applying the LMS algorithm of (3.5) yields:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \mu\varphi_{SMM}(n)ee'(n) \tag{3.25}$$

where

$$\varphi_{SMM}(n) = \left[-y'(n-1) \cdots -y'(n-n_a)\ x'(n) \cdots x'(n-n_b)\right]^T$$

$$= \frac{1}{A(q^{-1})}\varphi_{ee}(n) = \varphi'_{ee}(n)$$

and ee'(n) is the equation error of the adaptive system having the ARX plant described by (3.24).

The relationship between ee(n) and ee'(n) will be needed later. It can be derived simply as follows: From (2.19) applied to the ARX system of (3.24), it is seen that

$$ee'(n) = \tilde{\theta}^T\varphi'_{ee}(n) + v(n) \tag{3.26}$$

The expression for the equation error was given in (2.7) as:

$$ee(n) = \tilde{\theta}^T\varphi_{ee}(n) + A(q^{-1})v(n) \tag{3.27}$$

Autoregressively filtering each term of (3.27) by $A(q^{-1})$ yields:

$$\frac{1}{A(q^{-1})}ee(n) = \hat{\theta}^T\varphi'_{ee}(n) + v(n) \tag{3.28}$$

Thus it is seen from (3.26) and (3.28) that:

$$ee'(n) = \frac{1}{A(q^{-1})}ee(n) \tag{3.29}$$

This relationship should have been expected, since the "prime" has denoted here an autoregressive filtering by $A(q^{-1})$.

Note that generating the filtered regressor, $\varphi_{ee}(n)$, requires knowledge of the $A(q^{-1})$ polynomial, which is not known. As mentioned in the footnote of section 3.1.2 regarding the generation of an expression for $\nabla oe(n)$, the most recent estimate of $A(q^{-1})$, which is $\hat{A}(q^{-1},n-1)$, can be used to (approximat' ,) implement the filtering operation. Thus the following approximation is made, yielding the "IF" algorithms of [Fa86]:

$$\varphi'_{ee}(n) = \frac{1}{A(q^{-1})}\varphi_{ee}(n) \approx \frac{1}{\hat{A}(q^{-1},n-1)}\varphi_{ee}(n) \tag{3.30}$$

This filtering operation can be further simplified, as was also shown in Section 3.1.2, by defining the filtered quantities:

$$y^F(n) = \frac{1}{\hat{A}(q^{-1},n-1)}y(n)$$

$$x^F(n) = \frac{1}{\hat{A}(q^{-1},n-1)}x(n)$$

Therefore a simpler approximation to $\varphi'_{ee}(n)$ is :

$$\varphi'_{ee}(n) \approx \left[y^F(n-1) \cdots y^F(n-\hat{n}_a)\ x^F(n) \cdots x^F(n-\hat{n}_b)\right]^T \tag{3.31}$$

This approximation yields the simpler non-"IF" algorithms in [Fa86].

In addition to making the regressor filtering realizable, a very interesting relationship results from using $\hat{A}(q^{-1},n-1)$ for the filtering operation. The relationship (3.29) is now modified to:

$$ee'(n) = \frac{1}{A(q^{-1})} \approx \frac{1}{\hat{A}(q^{-1},n-1)}ee(n) = oe(n) \tag{3.32}$$

In light of the approximate equivalence between ee'(n) and oe(n) in (3.32) as the AR adaptive filter parameters converge to those of the plant, it has been shown [Fa86] that the LMS-type algorithm of (3.25), through minimization of the "primed" adaptive system equation error, ee'(n), will approximately minimize the output error of the original

ARMAX adaptive system of Figure 2.3, thus providing unbiased estimates. Also, in order to generate a realizable algorithm, (3.32) can be applied to (3.25), yielding the output error algorithm given in [Fa86]:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \mu\varphi'_{ee}(n)oe(n) \tag{3.33}$$

As previously mentioned, (3.30) or (3.31) can be used to approximately determine $\varphi'_{ee}(n)$, yielding the "IF" and non-"IF" algorithms, respectively, in [Fa86].

3.4.2 Least Squares Minimization

It is also natural to consider a least squares minimization of the equation error of the "primed" adaptive system in addition to gradient minimization. This is accomplished straightforwardly by recalling the least-squares criterion introduced in Section 3.2:

$$J_{LS}(n) = \sum_{i=1}^{n} e^2(i)$$

For the current problem, e(i)≡ee'(i), yielding the following criterion for the "primed" equation error system:

$$J_{LS}(n) = \sum_{i=1}^{n} ee'^2(i) \tag{3.34}$$

Recall that the sequence of error values, $\{ee'(i)\}_{i=1}^{n}$, is obtained by holding the parameters of the adaptive filter constant in the interval i=1, $\cdots$,n, these parameters being denoted as $\hat{\theta}(n)$. Applying the equation error relation (2.8) to the primed system thus expands (3.34) to:

$$J'_{LS}(n) = \sum_{i=1}^{n}\left[\hat{A}(q^{-1},n)y'(i) - \hat{B}(q^{-1},n)x'(i)\right]^2$$

Finally, again recall the process of implementing filtered quantities – the last available estimates are used to filter the desired signal. Thus, substituting for the signals x'(i) and y'(i) their actual realizations yields the desired least squares criterion for the "primed" adaptive system:

$$J'_{LS}(n) = \sum_{i=1}^{n} \left[\hat{A}(q^{-1},n) \frac{y(i)}{\hat{A}(q^{-1},i-1)} - \hat{B}(q^{-1},n) \frac{x(i)}{\hat{A}(q^{-1},i-1)} \right]^2 \qquad (3.35)$$

This is the general form of the SMM criterion [Fa89], which has been seen to exhibit a unimodal characteristic even when the original output error least squares criterion, $\Sigma oe^2(n)$, (or equivalently the MSOE surface) is multimodal.

It is interesting to consider the criterion (3.35) as the estimates converge to the true plant parameters. In this case, most of the polynomials

$$\hat{A}(q^{-1},i) \text{ and } \hat{B}(q^{-1},i)$$

for $i=1, \cdots, n$, will be approximately equal. Therefore, upon convergence to the plant parameters, the criterion will approach the following expression:

$$J'_{LS}(n) = \sum_{i=1}^{n} \left[y(n) - \frac{B(q^{-1})}{A(q^{-1})} x(n) \right]^2$$

The term in brackets is now seen to be the output error, oe(n). Thus it is seen that if the plant parameters do in fact minimize $J'_{LS}(n)$, minimization of this criterion is then equivalent to output error minimization. This has been proved for the case of a sufficient order adaptive system having white output noise, v(n) [Stc81].

Returning to the original SMM criterion of (3.35), it can be seen that minimization of $J'_{LS}(n)$ is exactly the same as least squares minimization on the "primed" adaptive system, i.e., considering the signals x'(n) and y'(n) as the input and output signals to an ARX plant. Also in light of the equivalence between $J'_{LS}(n)$ and $J_{LS}(n)$ as $\hat{\theta}(n) \rightarrow \theta_p$, it has

been shown that the output error, oe(n), can be utilized in place of ee'(n), similar to the gradient case. Therefore, the SMM algorithm is the following modified version of RLS:

$$\theta(n) = \theta(n-1) + \mathbf{R}^{-1}\varphi'_{ee}(n)oe(n)$$

$$\mathbf{R}(n) = \mathbf{R}(n-1) + \varphi'_{ee}(n)\varphi'^{T}_{ee}(n)$$

The above algorithm will be referred to as SMM(RLS), since it uses the recursive least squares algorithm in the context of the "primed" adaptive system of the SMM approach. Note that the data weighting and forgetting factor techniques discussed in Section 3.2.1 can also be utilized to implement the SMM approach. In the following chapter, the behavior of one of these standard SMM algorithms, SMM(RLSFF), will be observed through computer simulations and compared with a new SMM-type algorithm.

CHAPTER 4

SOME NEW OUTPUT ERROR ADAPTIVE ALGORITHMS

4.1 Use of OBE in the Output Error Adaptive System

Until now, the OBE algorithms have not been implemented in an output error adaptive system for identification of the ARMAX plant of Figure 2.1. One of the OBE algorithms has been extended [Rao89, Ch.3], however, in a manner similar to the *extended least squares* (ELS) scheme [Lj83, Sec 2.5.1], which permits identification of a general ARMAX plant (EOBE). Though this is an important result in its own right, it places restrictions on the c_i coefficients of the plant to ensure proper convergence of the adaptive filter parameters. This limits the plants to which this technique can be applied since, in general, there is no control over the plant parameters. For the current ARMAX system identification problem, it was shown in Chapter 2 that $c_i=a_i$, for $i=0, \cdots, n_a$ ($a_0=1$). It will be shown here that this ARMAX case can alternatively be dealt with by considering the output error adaptive system.

To understand the reason why OBE cannot be directly applied to the output error adaptive system, consider the expression for the output, y(n), of the plant, given as:

$$y(n) = -\sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=0}^{n_b} b_i x(n-i) + \sum_{i=0}^{n_a} a_i v(n-i)$$

Recall the key assumption of the OBE algorithm: The contribution of the noise to the plant output must be *bounded*. In practice, the quantity which is most realistically bounded is the

output noise term, v(n). However, knowledge of a magnitude bound on v(n) is not equivalent to knowing a bound on the total noise contribution to y(n). This is because the noise contribution is actually an FIR filtered version of v(n), with the filter being the unknown autoregressive plant polynomial, $A(q^{-1})$. Since the noise bound depends on the unknown plant parameters, applying OBE in this situation is not a well-posed problem. This is one of the reasons why for a proper operation of OBE on this system, some restrictions must be placed on the a_i coefficients to ensure that the total noise term satisfies a bound when v(n) itself is bounded [Rao89, Sec.3.3].

4.1.1 Presentation of the Algorithm

As an alternative to the general ARMAX identification scheme of EOBE, again consider the plant description, given in operator notation:

$$A(q^{-1})y(n) = B(q^{-1})x(n) + A(q^{-1})v(n)$$

Autoregressively filtering each quantity by $A(q^{-1})$, exactly as done in Section 3.4, yields an SMM-type approach to identification of the plant:

$$A(q^{-1})y'(n) = B(q^{-1})x'(n) + v(n) \tag{4.1}$$

It is important to see here that now v(n) appears "as is" in the alternative plant description (4.1), and thus the bounded noise assumption is directly satisfied without requiring any restrictions on the plant. Furthermore, note that (4.1) describes an ARX system with input x'(n) and output y'(n), which is the structure needed to utilize OBE. Also, recall (Section 3.4, Eq.(3.32)) the approximate equivalence of the error in equation error adaptive system using (4.1) as the plant description and the error of the output error adaptive system having the original ARMAX plant.

Therefore, using OBE to identify the original ARMAX plant is equivalent to applying the algorithm to the "primed" ARX system of (4.1). This will require two modifications to the original algorithm:

1) The signals x'(n) and y'(n) must be used as the input and output quantities in the regressor vector. In other words, $\varphi'_{ee}(n)$ must be used in place of $\varphi_{ee}(n)$ in the standard OBE algorithm. Two methods of generating $\varphi'_{ee}(n)$ were given in Section 3.4, Eqs. (3.30) and (3.31).

2) The output error, oe(n), of the adaptive system having the original ARMAX plant must be used in place of ee(n) in the original OBE algorithm. This is possible in light of the approximate equivalence between the equation error of the "primed" system, ee'(n), and the output error, oe(n).

This algorithm will be referred to as SMM(OBE).

As a test of this algorithm, simulations were performed for three cases considered in [Fa86], where the authors' algorithm (3.33) was shown to converge to unbiased estimates of the plant:

Case 1) **Sufficient order adaptive filter, unimodal performance surface.**

For this case the output error adaptive system is described by:

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{1}{1-1.2q^{-1}+0.6q^{-2}}$$

$$\frac{\hat{B}(q^{-1},n)}{\hat{A}(q^{-1},n)} = \frac{\hat{b}_0(n)}{1-\hat{a}_1(n)q^{-1}-\hat{a}_2(n)q^{-2}}$$

A uniformly distributed, zero mean, unit variance, white sequence was used as the input, x(n). It was shown in [Str81] that the error surface of this adaptive system is unimodal.

Case 2) **Sufficient order adaptive filter, multimodal performance surface.**

A multimodal error surface was constructed in [So75], using the following adaptive system:

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{1}{(1-0.7q^{-1})^2} = \frac{1}{1-1.4q^{-1}+0.49q^{-2}}$$

$$\frac{\hat{B}(q^{-1},n)}{\hat{A}(q^{-1},n)} = \frac{\hat{b}_0(n)}{1-\hat{a}_1(n)q^{-1}-\hat{a}_2(n)q^{-2}}$$

The input sequence, x(n), was a correlated sequence, obtained by passing uniformly distributed, zero mean, unit variance, white noise through the following filter:

$$(1-0.7q^{-1})^2(1+0.7q^{-1})^2 = 1 - 0.98q^{-2} + 0.2401q^{-4}$$

Case 3) **Reduced order adaptive filter, multimodal performance surface.**

The multimodal reduced order adaptive system examined here was introduced in [Jo77]. It is described by:

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{0.05-0.4q^{-1}}{1-1.1314q^{-1}+0.25q^{-2}}$$

$$\frac{\hat{B}(q^{-1},n)}{\hat{A}(q^{-1},n)} = \frac{\hat{b}_0(n)}{1-\hat{a}_1(n)q^{-1}}$$

As in case 1), the input , x(n), was a uniformly distributed, zero mean, unit variance, white sequence.

In all the simulation cases, the output noise, v(n), was a uniformly distributed, zero mean white sequence independent of the white sequence generating the input, x(n). It should also be noted here that stability projection (see Section 2.3.3) was used, because of the IIR structure of the adaptive filter. The simulations were run with the following questions in mind: 1) Is this algorithm a viable alternative to output error adaptive system identification? More simply put, does this algorithm work at all? 2) If the answer to 1) is affirmative, then how does SMM(OBE) compare to the "standard" SMM algorithms [So88], which use equation-error minimization algorithms such as LMS [Fa86], or RLS. This latter algorithm will be denoted as SMM(RLS).

A partial answer to question (1) was obtained by running the simulations and checking for global convergence. To illustrate proper operation of SMM(OBE), simulations were run and the behavior of the parameter estimates was observed. For the sufficient order cases 1) and 2), recall from Section 2.3.1 that MMSOE=σ_v^2 and the parameters which yield this MMSOE are precisely those of the plant, θ_p. On the other hand, in the reduced order adaptive system of case 3), there is no "true" parameter vector that the adaptive filter can take on that will match the plant exactly, since $\hat{\theta}$ and θ_p have different dimensions. The resulting MSOE surface for this adaptive system will thus have a minimum point which is greater than σ_v^2, due to the inability of the adaptive filter to "match" the plant. The disparity between the minimum MSOE achievable with a sufficient order adaptive filter and that achieved by a reduced order adaptive filter is caused by what is known as *model mismatch*. In other words, the minimum MSOE of a reduced order adaptive system can be thought of as being separated into two components as follows:

$$\text{MMSOE} = \text{MMSOE}_v + \text{MMSOE}_{mm}$$

where MMSOE_v is the minimum mean square output error due to the output noise, v(n), obtained by considering the adaptive filter to be of sufficient order. Again recall from

Section 2.3.1 that $MMSOE_v=\sigma_v^2$. The term $MMSOE_{mm}$ is the minimum mean square output error due to the model mismatch, which is obtained by taking $v(n)\equiv 0$ and generating the resulting MSOE surface. This was done in [Jo77] for the simulation case 3), where it was shown that $MMSOE_{mm}=0.2066$, occurring for the adaptive filter parameters of:

$$\hat{\theta} = [\hat{a}_1 \;\; \hat{b}_0]^T = [-0.906 \;\; -0.311]^T$$

Recall for simulation cases 2) and 3), the MSOE surfaces are multimodal [Fa86]. Therefore, to illustrate global convergence in these situations, initial estimates, $\hat{\theta}(0)$, were provided which were very close to a local minimum of the MSOE surface. The parameter trajectories of the adaptive filter were then observed to see if the parameters were adapted such that they moved away from the parameter yielding the local minimum MSOE to the one yielding the global minimum MSOE. The trajectories obtained for cases 1)-3) are shown in Figure 4.1 for a simulation run of 1000 iterations, which was well after convergence.† Shown is the initial parameter estimate, $\hat{\theta}(0)$, the final estimate, $\hat{\theta}(1000)$, and the theoretical parameter estimate, θ_0, yielding the MMSOE. In the simulation cases 2) and 3), the parameter estimates are seen to be adapted away from the parameter yielding a local minimum MSOE towards the one yielding the global minimum MSOE.

† Convergence was determined by viewing the learning curve, to be discussed in Section 4.1.2.

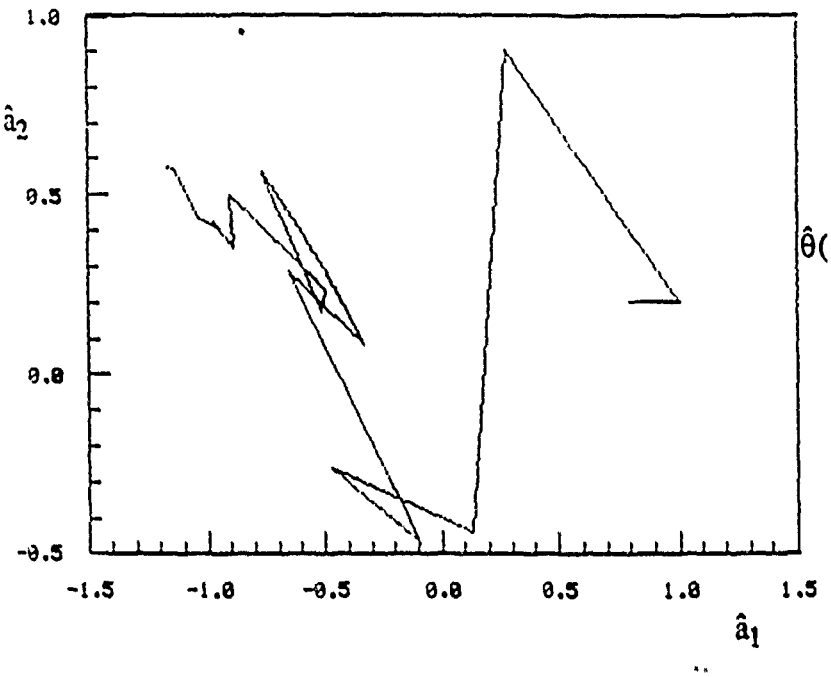


Figure 4.1a Simulation case 1) trajectories

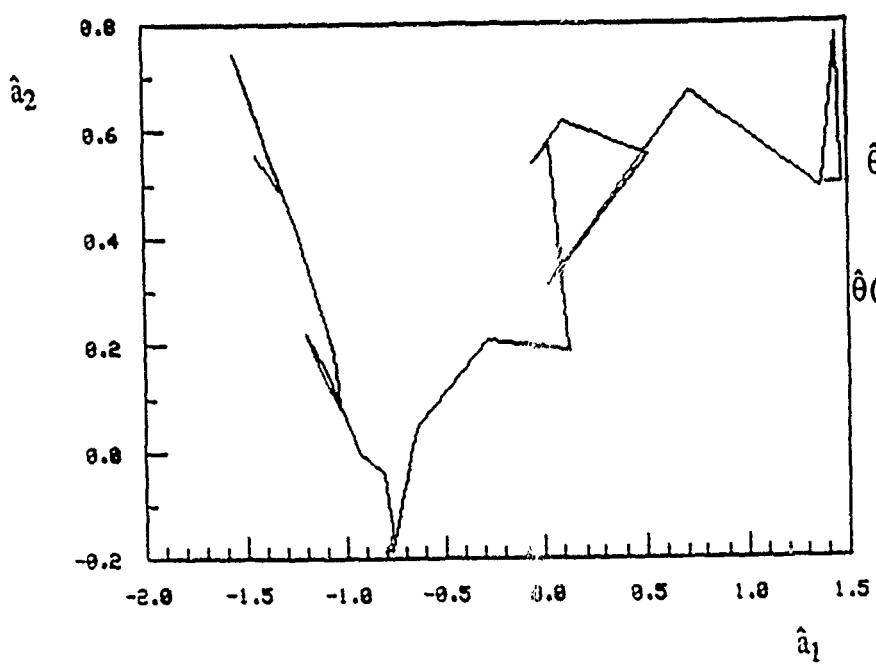


Figure 4.1b Simulation case 2) trajectories

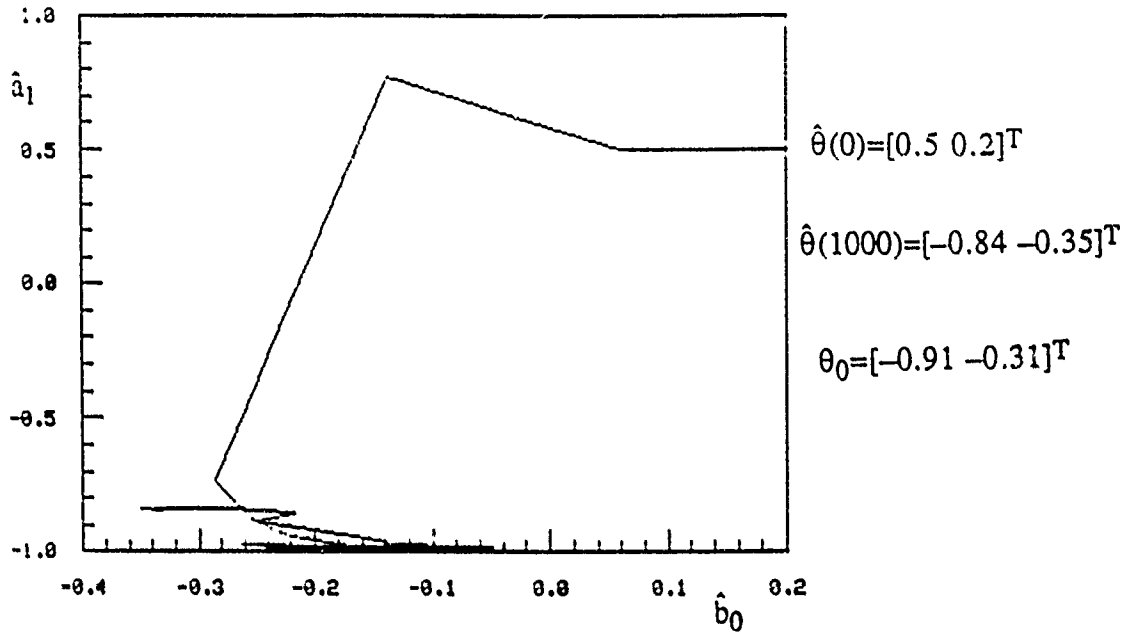


Figure 4.1c Simulation case 3) trajectories

A remark is in order regarding the trajectories shown in Figure 4.1. Note that the trajectories given for cases 1) and 2) are a plot of $\hat{a}_2$ versus $\hat{a}_1$. The reason for considering only the AR parameters as opposed to the X parameter $\hat{b}_0$ is that in the sufficient order case, the MSOE surface of an output error adaptive system is quadratic in the X parameters, and the X parameter estimate which minimizes the MSOE is in fact the true parameter, $\hat{b}_0$. Thus there is no problem obtaining unbiased estimates for these parameters because there are no local minima with respect to them, as might happen with the AR parameter estimates. However, this is not true in reduced order adaptive systems. Therefore, in the reduced order case 3), the plot of $\hat{a}_1$ versus $\hat{b}_0$ is considered. This is because the MSOE surface is a highly nonlinear function of *both* the AR and X coefficients,[*] and thus in addition to the behavior of the $\hat{a}_1$ parameter, the behavior of the

[*] See [Jo77] or [Sh89] For the explicit expression for the $MSOE_{mm}$ surface, i.e., the expression of MSOE in terms of $\hat{b}_0$ and $\hat{a}_1$ with $v(n)\equiv 0$.

estimate $\hat{b}_0$ needs to be observed for proper convergence to the parameter yielding MMSOE.

Thus from the results shown in Figure 4.1, it appears that the answer to question (1) above is "yes," i.e., SMM(OBE) did in fact identify the parameters of an ARMAX plant in the output error adaptive system configuration for the simulation cases 1)-3). Next, attention is given to the more interesting (and more difficult) question (2), which is the subject of the next section.

4.1.2 Performance of SMM(OBE) versus SMM(RLSFF)

Depending on the application, the environment in which an adaptive system is operating could have either a large or small noise content. It is therefore of general interest to examine the performance of adaptive algorithms operating on systems with varying levels of noise. Furthermore, the examination of performance with respect to different noise levels can also serve as a basis of comparison between different algorithms. Of particular interest here is an investigation of the performance of a "standard" SMM algorithm, SMM(RLSFF), with respect to the new SMM algorithm, SMM(OBE).

To compare the two algorithms in this manner, simulations were performed on the simulation cases 1)-3) for varying signal-to-noise ratios (SNR's). The SNR is defined as:

$$\text{SNR} = \frac{\sigma_x^2}{\sigma_v^2}$$

Usually (as will be the case in this discussion), this quantity is given in decibels (dB), converting the above expression to:

$$SNR(dB) = 10\log\frac{\sigma_x^2}{\sigma_v^2}$$

The SNR was varied from –10 to 10 dB, in steps of 2 dB, which was accomplished by appropriately scaling the uniform output noise, v(n), with respect to the input signal. The algorithm SMM(RLSFF) was implemented with a forgetting factor, λ, of 0.99. At each value of the SNR, SMM(RLSFF) and SMM(OBE) were compared with respect to two criterion:

1) The minimum value of MSOE which was achieved (MMSOE).

2) Transient MSOE behavior.

Both criteria were evaluated through the consideration of the adaptive system *learning curve*, which is a plot of the MSOE versus n, the number of iterations. Initially, at the start of adaptation, the MSOE is usually high, since the initial adaptive filter parameters, $\hat{\theta}(0)$, are probably much different than the true plant parameters, θ_p. As adaptation proceeds, the estimates, $\hat{\theta}(n)$, generally adapt so as to get closer to θ_p. This process yields a monotonically decreasing sequence of MSOE values as a function of the time index, n. For an algorithm which does in fact converge, this sequence will approach some constant minimum value as n gets large (i.e. the MMSOE)†

Note that the characteristics described above for the learning curve apply to the curve $E[oe^2(n)]$ versus n. To determine this curve experimentally would require taking an ensemble average of an infinite number of independent realizations of $oe^2(n)$ versus n. Obviously this is not possible. However, averaging a relatively small number of the curves $oe^2(n)$ versus n provides a very good indication of MSOE performance, even though these

† For this to be true, the common assumpuon made here (and throughout this discussion) is that the plant is fixed and that the input and noise sequences, x(n) and v(n), are stationary.

curves may not have the precise monotone characteristics of the actual learning curve. In fact, there will be considerable fluctuation in these curves, as will be seen.

With these properties of the learning curve in mind, the two performance criteria are now considered for SMM(RLSFF) and SMM(OBE).

MMSOE

After viewing the learning curve resulting from each simulation case at every SNR using both algorithms, it appeared that all of the curves reached their minimum, steady-state levels well before 1000 iterations. See Figure 4.2 for a typical example of this transient behavior for both SMM(RLSFF) and SMM(OBE). The quantity used as an approximation to the MMSOE was a time average of the last 50 values of the experimental learning curve. This value will be called the *steady-state MSE* (SSMSE). To compare the two algorithms with respect to this quantity, the SSMSE was plotted as a function of the SNR. These plots are shown for each of the simulation cases in Figures 4.3a, 4.4a, and 4.5a. Observe that the curves have an exponential characteristic. To see why this is so, recall from Section 2.3.1 that the minimum value of MSOE is σ_v^2, occurring when $\hat{\theta}=\theta_p$. Therefore, the experimental curves should approximate a curve of σ_v^2 versus SNR. But recall that the SNR is related to σ_v^2 as follows:

$$\text{SNR(dB)} = 10\log\frac{\sigma_x^2}{\sigma_v^2}$$

Solving for σ_v^2 yields:

$$\sigma_v^2 = \sigma_x^2 \text{alog}\left[\frac{-\text{SNR(dB)}}{10}\right]$$

where the function alog is the inverse of the (base 10) log function. It is thus seen that the theoretical MMSOE versus SNR curve has the above exponential form. The theoretical curves are given in Figure 4.3b, 4.4b, and 4.5b for each simulation case.

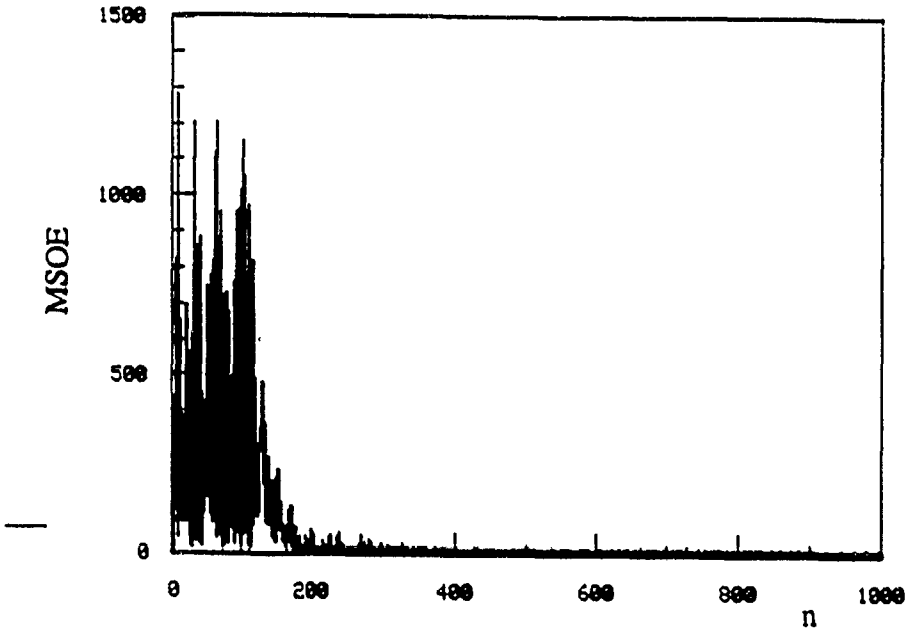


Figure 4.2a Simulation case 1) learning curve for SMM(RLSFF). SNR=–10dB

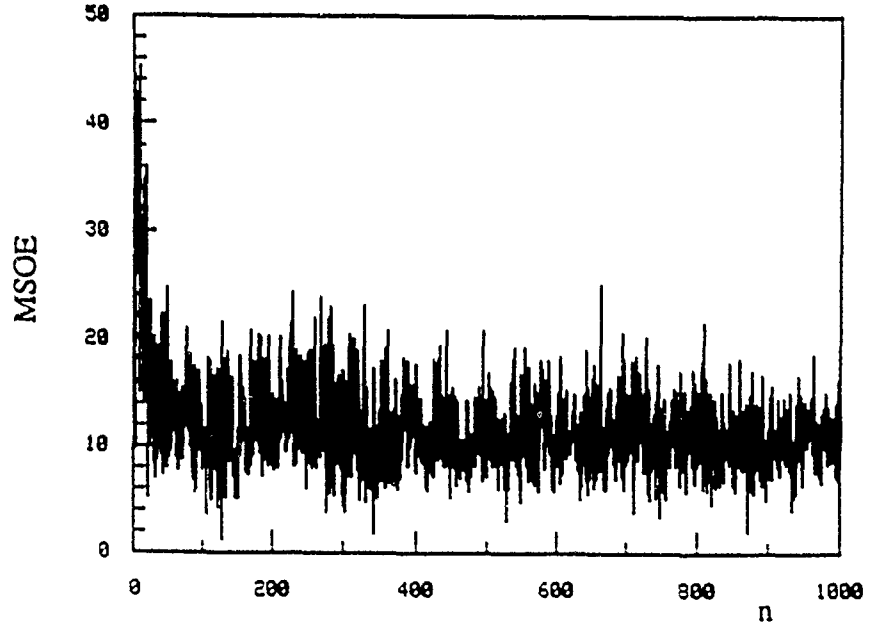


Figure 4.2b Simulation case 1) learning curve for SMM(OBE). SNR=–10dB

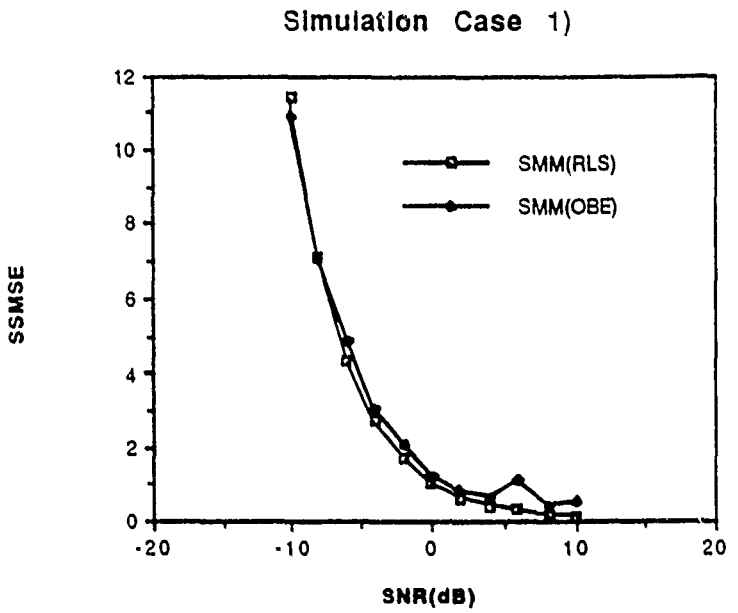


Figure 4.3a Simulation case 1) SSMSE curve

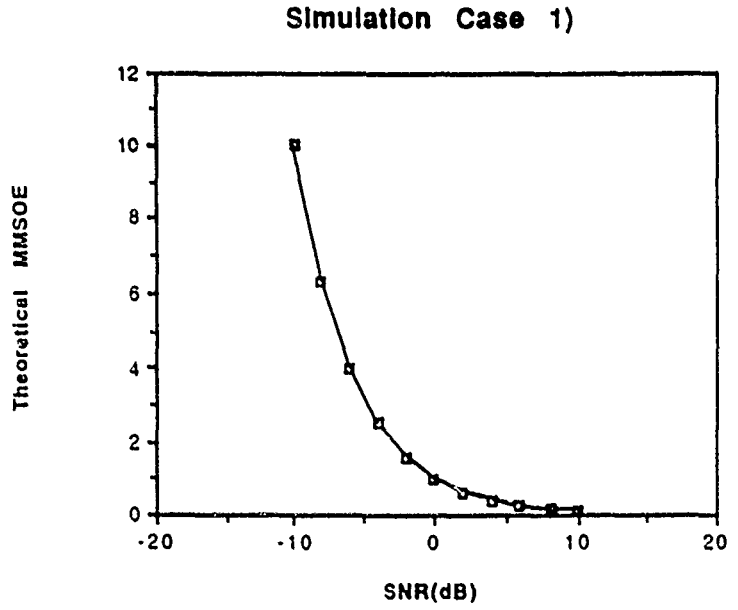


Figure 4.3b Theoretical MMSOE for case 1)

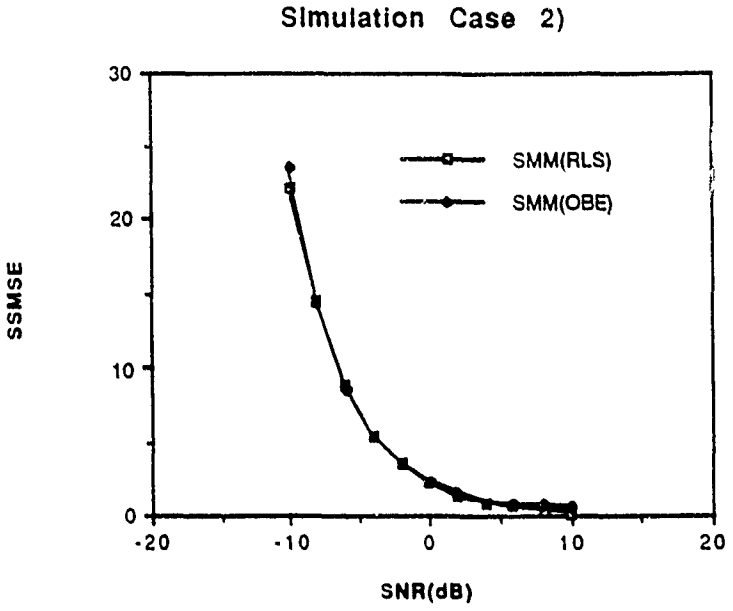


Figure 4.4a Simulation case 2) SSMSE curve

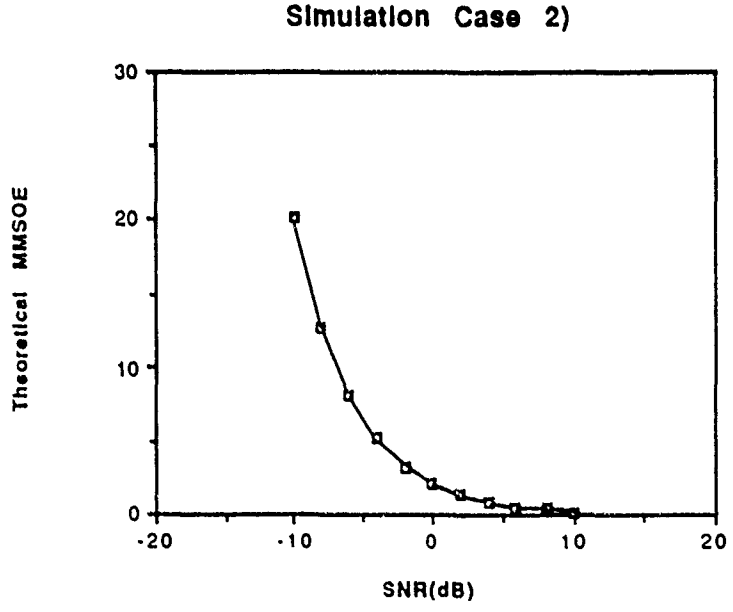


Figure 4.4b Theoretical MMSOE for case 2)

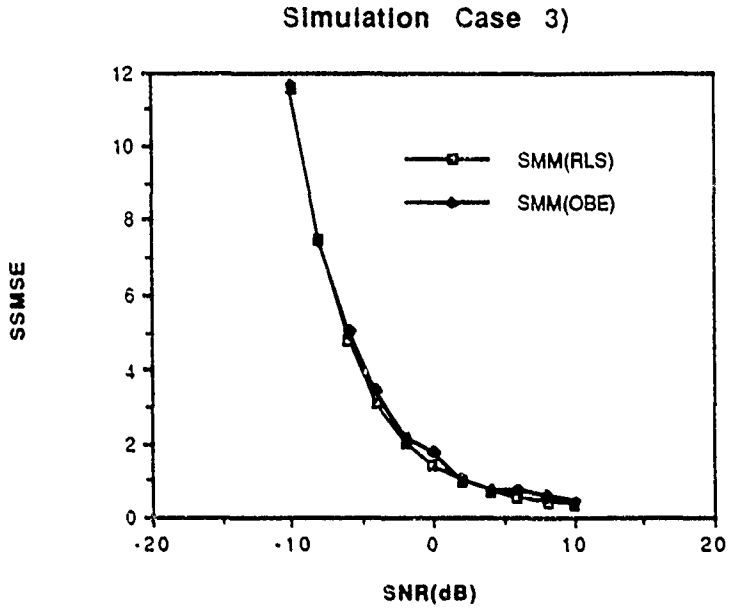


Figure 4.5a Simulation case 3) SSMSE curve

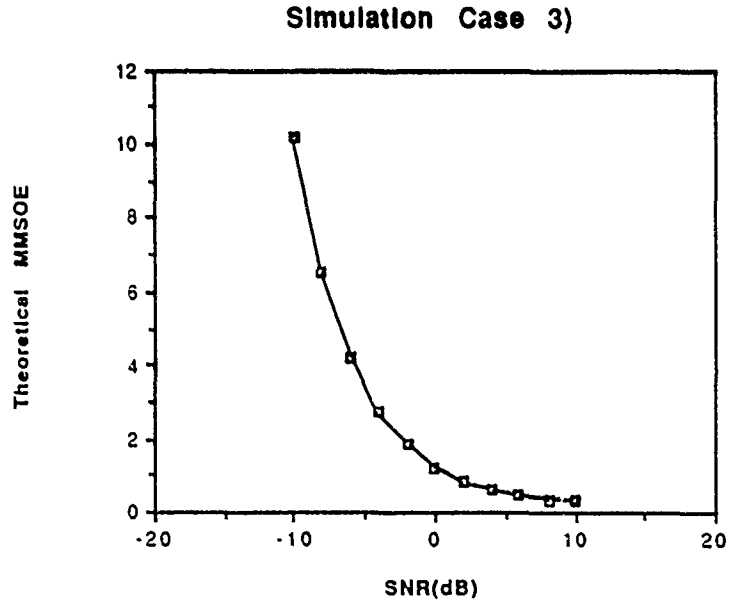


Figure 4.5b Theoretical MMSOE for case 3)

Upon viewing the experimental curves of Figures 4.3a-4.5a, it can be seen that SMM(OBE) performs very comparably to SMM(RLSFF). Also note that the experimental curves are very close to their theoretical lower bounds of MMSOE performance. This is encouraging, since RLS algorithms are based on MSE minimization, while the OBE algorithm is based on minimizing ellipsoidal membership sets in a geometrical sense. Though these two schemes minimize different quantities, it can be seen in Figures 4.3-4.5 that SMM(OBE) does in fact perform very well with respect to the MSE minimization criterion of RLS, especially in the region of low SNR values. At higher SNR levels, however, SMM(OBE) occasionally, but not consistently, yielded values of SSMSE which exceed significantly those of SMM(RLSFF). This was especially evident in simulation case 1) (Figure 4.3), but can also be seen to some degree in each of the curves. Thus it might be conjectured that anomalous SSMSE behavior is more prone to occur in SMM(OBE) at higher SNR's than at lower values. In particular, for negative SNR levels, all the simulation curves in Figures 4.3-4.5 indicate consistently good SMM(OBE) performance. The apparent anomalous behavior was observed always when the SNR was greater than zero.

The above observations suggest near optimum performance of SMM(OBE) with respect to the SSMSE criterion, especially at low SNR's. Next, the transient characteristics of the learning curve will be addressed and it will be seen that SMM(OBE) actually exhibits *superior* transient behavior to that of SMM(RLSFF) at low SNR's.

Transient MSOE behavior

In many applications (especially time-varying cases) the best steady-state performance may not be the only important concern. The manner in which the steady state is achieved may also be of extreme importance. Examination of the learning curve also provides much insight into this transient behavior.

Upon viewing the learning curves of SMM(RLSFF) and SMM(OBE) for each SNR, some very interesting characteristics were observed. In most cases the learning curves obtained when using SMM(RLSFF) exhibited much higher peaks at smaller values of n compared to that of SMM(OBE). Though this usually occurred, it was especially evident at low SNR's, where SMM(RLSFF) peaked at huge values of MSOE compared to the peaking of SMM(OBE). See again Figure 4.2 for an example of this behavior.

However, as with the SSMSE comparisons, some seemingly anomalous behavior was observed in the learning curve of SMM(OBE) at some higher SNR's. In fact, the same SNR's which yielded higher SSMSE yielded very peculiar learning curves. As can be seen in Figure 4.3a, this unusual behavior was exhibited in simulation case 1) at the SNR's of 6 and 10 dB. The corresponding learning curves are shown in Figures 4.6 and 4.7. Referring to Figure 4.6, it is seen that both algorithms peak at about the same time and magnitude. The SMM(OBE) learning curve of Figure 4.6b shows that up to approximately 150 iterations, SMM(OBE) appears to be converging smoothly as it did in most of the other simulations (i.e. see Figure 4.2b). However, after this point, the learning curve exhibits erratic behavior and subsequently does not settle to a level comparable to SMM(RLSFF). In Figure 4.7, both learning curves appear on the same plot for comparison. Referring to Figure 4.7a, it is seen that the peak of the learning curve of SMM(OBE) is significantly higher than that of SMM(RLSFF), though they both reach steady state at about the same time. To observe the steady state characteristics of the curves, the portion of Figure 4.7a from 800 to 1000 iterations was expanded in Figure 4.7b. Erratic steady state behavior of the learning curve of SMM(OBE) is again observed.

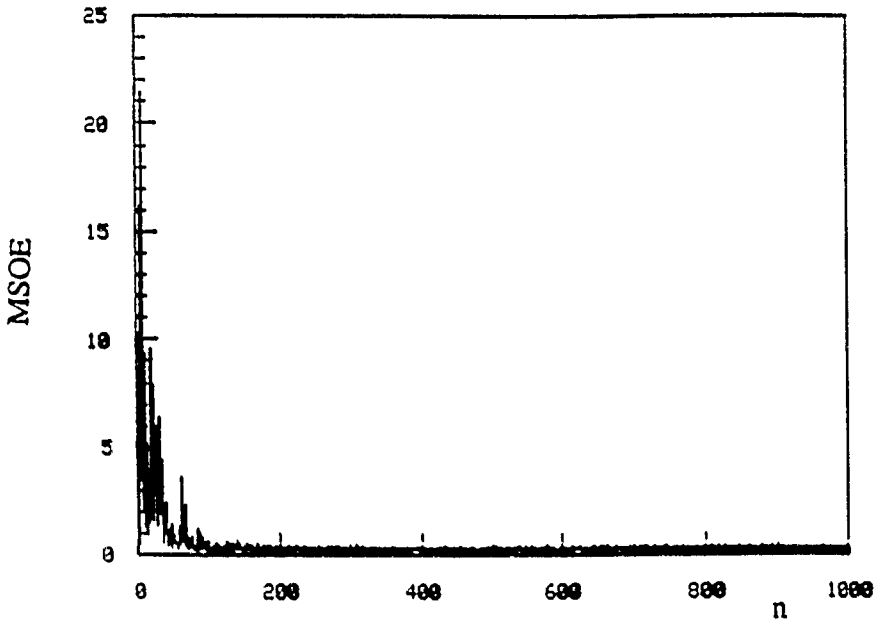


Figure 4.6a Simulation case 1) learning curve for SMM(RLSFF). SNR=6dB

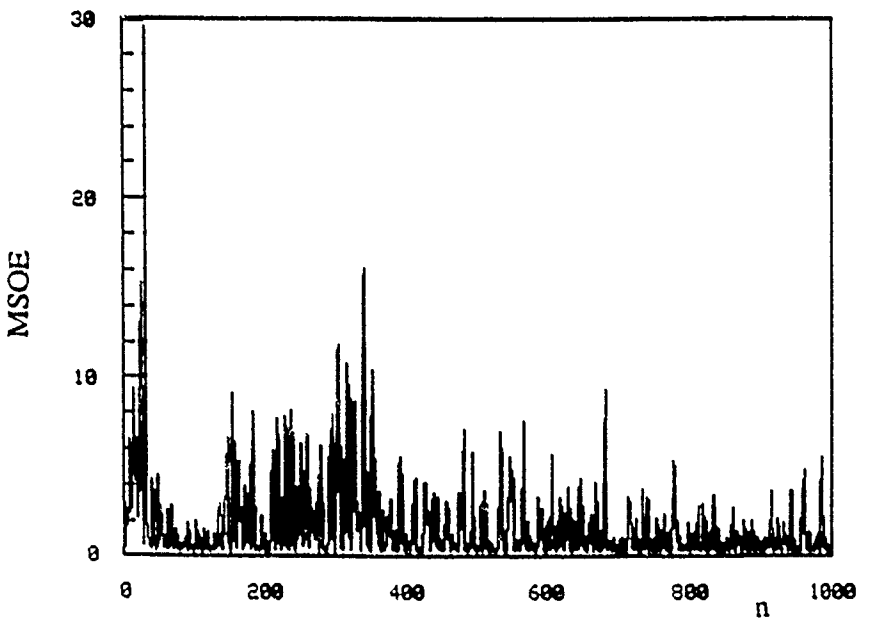


Figure 4.6b Simulation case 1) learning curve for SMM(OBE). SNR=6dB

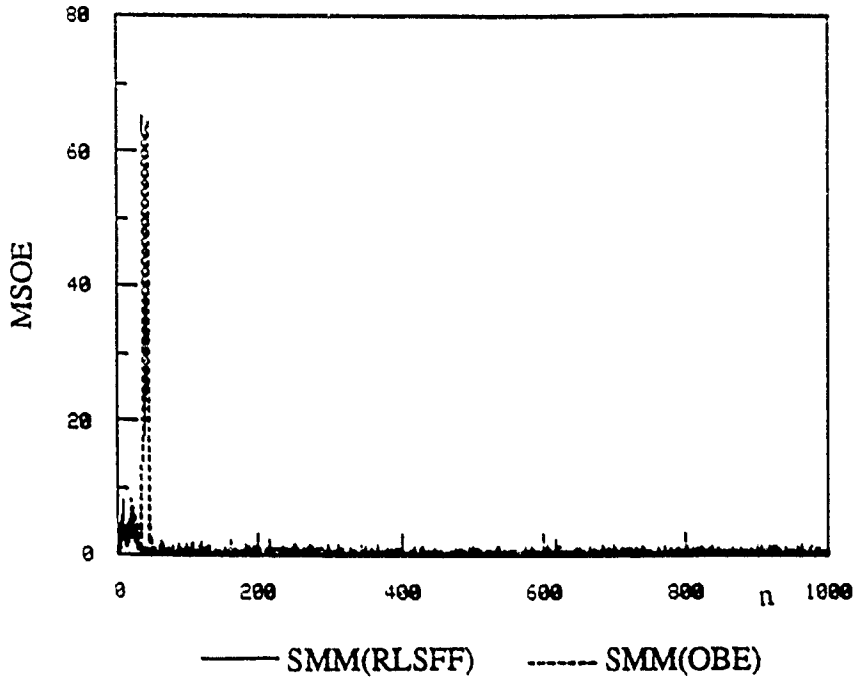


Figure 4.7a Simulation case 1) learning curves. SNR=10dB

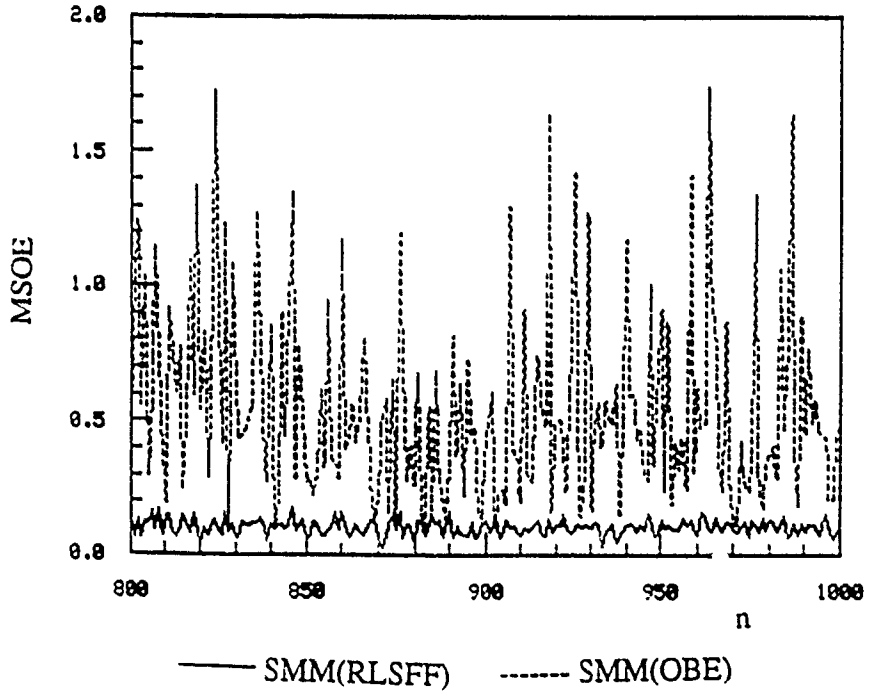


Figure 4.7b Simulation case 1) learning curves. SNR=10dB

An additional phenomenon of anomalous behavior was found in SMM(RLSFF), which was surprising. In simulation case 2) at the low SNR's of –2 and –10 dB, SMM(RLSFF) took very long to converge compared to both SMM(OBE) at those SNR's as well as itself at all other SNR's. See Figures 4.8 and 4.9 for these learning curves. Since there is so much peaking at low values of n, the curves of Figures 4.8a and 4.9a were viewed starting from n=500 in Figures 4.8b and 4.9b in order to see the actual point at which steady state was achieved. It can be seen that in both cases, more than 800 iterations were needed for SMM(RLSFF) to achieve steady state. Figures 4.8c and 4.9c show the learning curves yielded by SMM(OBE) for the same simulation cases, and it can be seen that SMM(OBE) converged smoothly in less than 200 iterations, as it did for all cases of negative SNR.

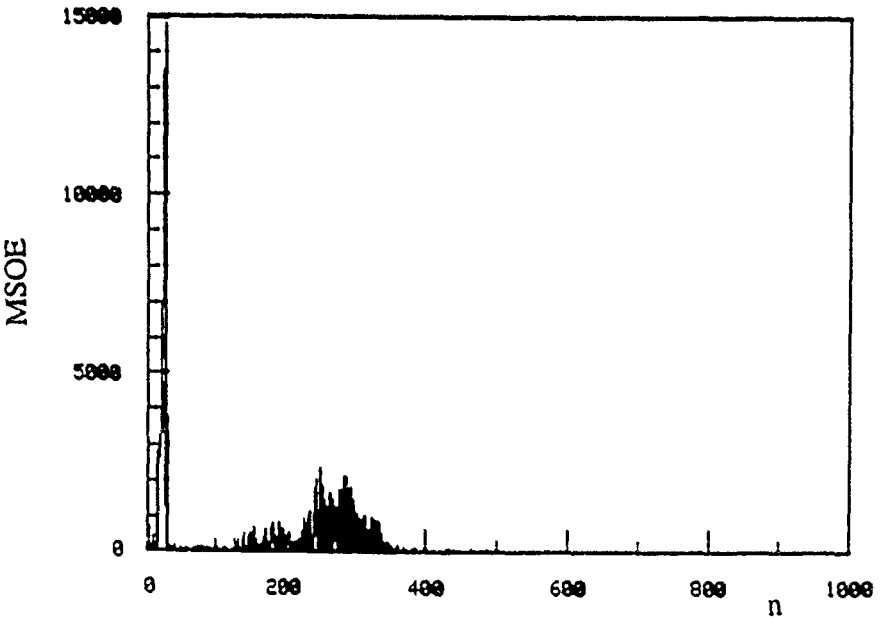


Figure 4.8a Simulation case 2) learning curve for SMM(RLSFF). SNR=–2dB

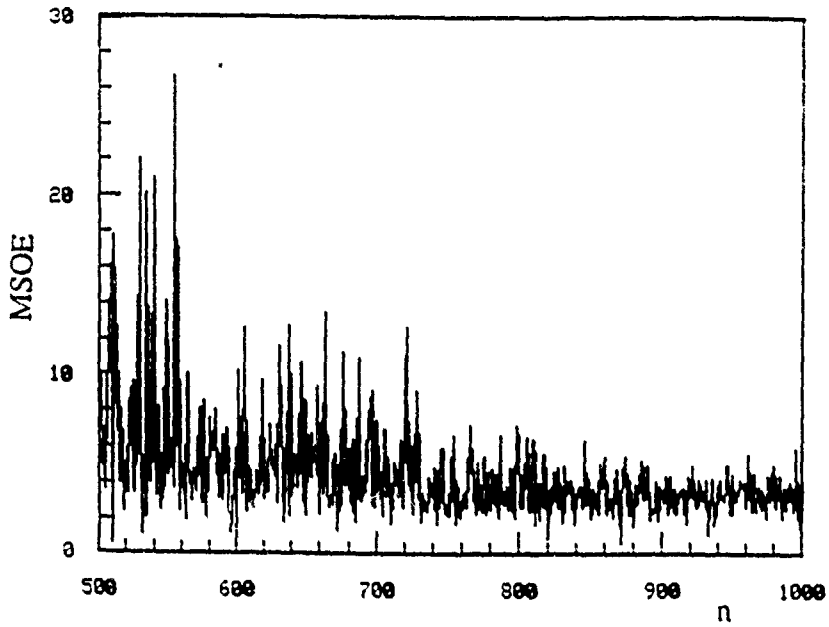


Figure 4.8b Simulation case 2) learning curve for SMM(RLSFF). SNR=–2dB

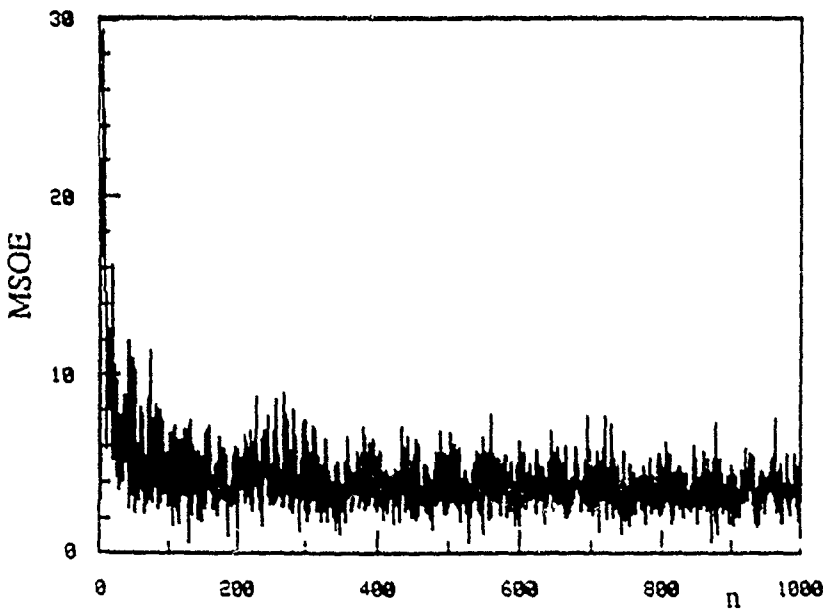


Figure 4.8c Simulation case 2) learning curve for SMM(OBE). SNR=–2dB

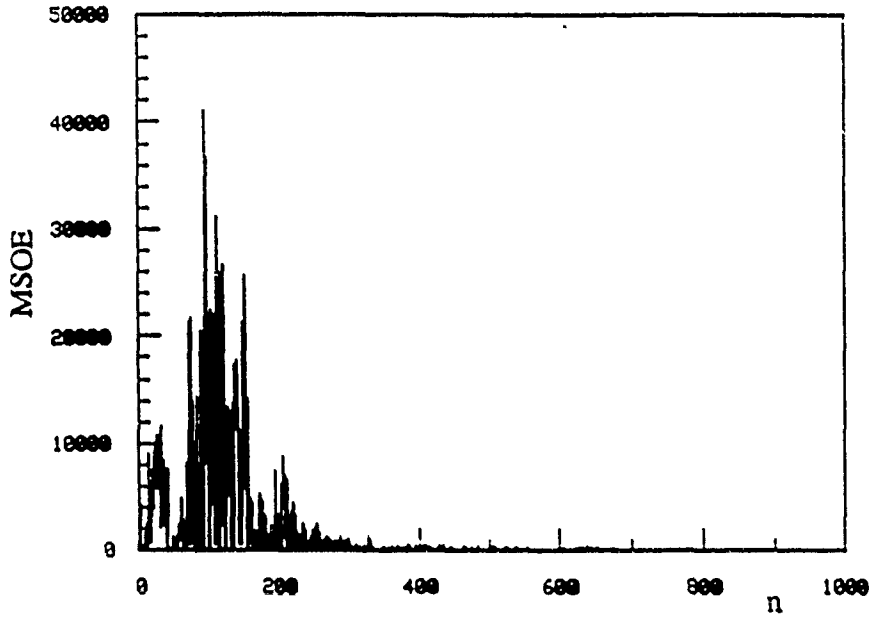


Figure 4.9a Simulation case 2) learning curve for SMM(RLSFF). SNR=–10dB

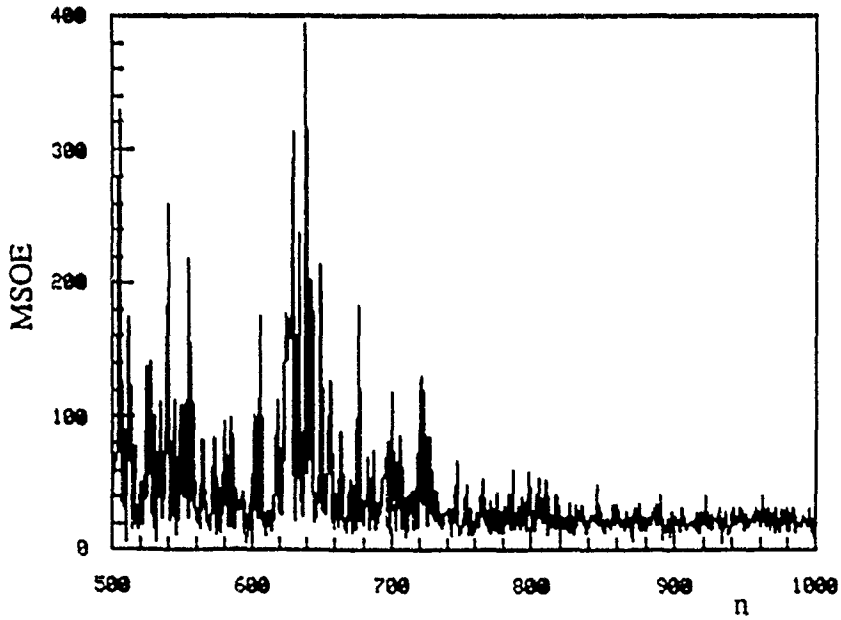


Figure 4.9b Simulation case 2) learning curve for SMM(RLSFF). SNR=–10dB

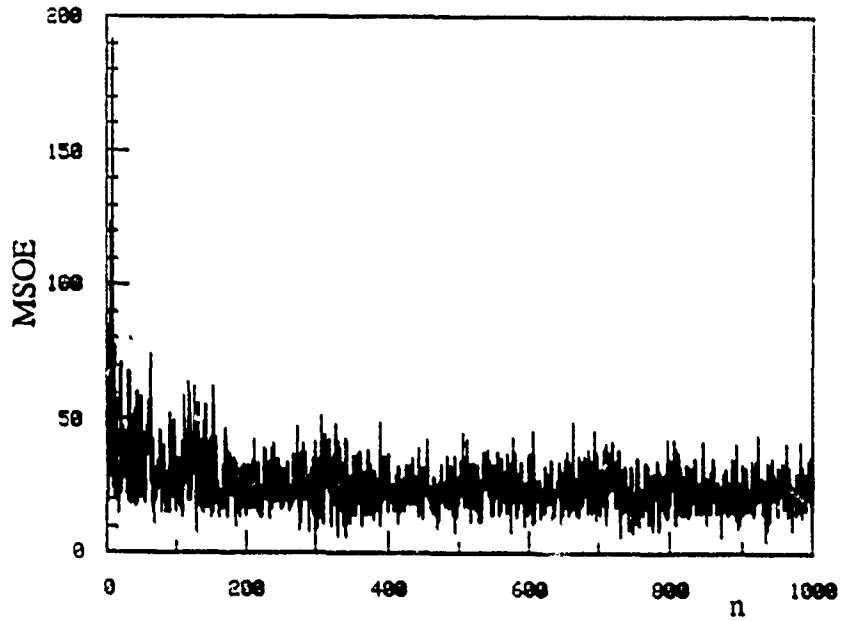


Figure 4.9c Simulation case 2) learning curve for SMM(OBE). SNR=-10dB

4.1.3 Summary of Simulations

To summarize the observations of the preceding section, a few key points will be reiterated here. First of all, both SMM(RLSFF) and SMM(OBE) performed well in the cases studied. However, unusual behavior of both algorithms was found which seemed to follow a trend with respect to varying SNR levels. In particular, at a few low SNR's, which were negative, SMM(RLSFF) was found to converge very slowly. SMM(OBE), on the other hand, converged very rapidly for all simulation cases at every SNR less than zero. This might suggest a greater dependability of SMM(OBE) with respect to SMM(RLSFF) in the presence of higher noise levels. At larger SNR's (≥0), SMM(RLSFF) appears to be the more dependable algorithm, as SMM(OBE) was seen not to converge very well in a few cases.

Also worthy of mention are two observations made on the performance of SMM(OBE) as the SNR varied. The first observation was a consistent increase in the amount of data used by SMM(OBE) (i.e. the number of updates to $\hat{\theta}$) as the noise level increased (i.e. SNR↓). This is an interesting and intuitively satisfying property for an information-dependent updating scheme to have. It seems reasonable that as the signal gets more and more corrupted by noise, the algorithm needs to take more and more "looks" at it to extract the proper information. Out of 1000 iterations, the average number of updates used in 10 independent runs of SMM(OBE) and the corresponding SNR values for each of the simulation cases are shown in Table 1, where the inverse relationship between the SNR and the number of updates is evident for all but a few increments in the SNR. Note in only one simulation did the amount of data used for updating $\hat{\theta}$ exceed 10% of the total data. In fact, for most cases, the parameter estimates were updated less than 8% of the time.

TABLE 1

	Average Number of Updates of SMM(OBE)		
SNR(dB)	Case 1)	Case 2)	Case 3)
−10	88.5	97.7	76.0
−8	80.0	104.5	65.8
−6	71.8	77.9	60.8
−4	66.3	70.2	58.5
−2	68.3	65.2	62.2
0	67.5	52.6	63.6
2	65.7	42.4	53.6
4	45.0	41.9	51.5
6	43.6	42.3	60.3
8	36.2	42.7	60.4
10	29.5	45.8	52.3

The second of the observations made on SMM(OBE) was an insensitivity of the operation of the algorithm with respect to the choice of the magnitude bound on the noise, ρ. This characteristic has also been observed in other OBE algorithms as well [Rao89, Sec.3.4]. This observation was made through the following experiment. Initially, the

noise bound, γ, was chosen to agree with the noise level when the SNR was 0 dB. This value of γ was used for the SNR's ranging from –10 to 10 dB. Note that for low SNR's (<0), the noise bound at 0 dB is not a bound at all, and at high SNR's (>0), the 0 dB bound is too large, which could degrade performance. In particular, since the noise distribution was chosen to be uniform with zero mean, its magnitude bound can be calculated using the formula:

$$\gamma_{SNR(dB)} = \sqrt{\frac{3\sigma_x^2}{SNR}}$$

Note that on the right side of this equation, the SNR is **not** in dB's. This calculation for SNR = 0, –10, and 10 dB yields $\gamma_0=\sqrt{3}\approx1.73$, $\gamma_{-10}=\sqrt{30}\approx5.48$, and $\gamma_{10}=\sqrt{0.3}\approx0.548$. These calculations yield a factor of $\sqrt{10}\approx3.16$ underestimation or overestimation for the simulations using SNR's of –10 and 10 dB, respectively. In spite of these misjudgments in γ, SMM(OBE) was observed to perform virtually identically to when the proper bound was used. It thus appears that the performance of SMM(OBE) is insensitive to the accuracy of γ.

Practically, this is a very important property for an OBE algorithm to have, since it is often not possible to meet certain assumptions of any algorithm exactly. It is therefore crucial that a deviation of the true conditions from the ideal case does not cause a complete failure in performance, a robustness property. Thus it appears that SMM(OBE) can be described as being robust with respect to the choice of the noise bound used.

4.2 A Proposal of Two New Output Error Adaptive Algorithms

The final results of this thesis involve an RLS-type derivation of algorithms for use with the output error adaptive system. By utilizing some previously derived expressions for the output error, oe(n), and its gradient, two algorithms can be derived which are

identical to the RPE algorithms of Section 3.2.2, except for the construction of the matrix R(n). Further investigations – both simulations and analysis – are needed to determine the convergence properties of these algorithms.

Proceeding as in Section 3.2, it is desired to minimize the criterion $J_{LS}(n)$ of (3.11) with oe(n) as the error term:

$$J_{LS}(n) = \sum_{i=1}^{n} oe^2(i)$$

Again, taking derivatives with respect to the least-squares estimate after n iterations, $\hat{\theta}(n)$, and setting to zero gives:

$$2\sum_{i=1}^{n} oe(i)\frac{doe(i)}{d\hat{\theta}(n)} = 0 \qquad (4.2)$$

This expression will be implemented in two different ways, giving rise to two algorithms which are slightly different than RPE and have an appearance similar to the instrumental variable method [Lj83,Sec.2.2.2], as will be discussed in Section 4.2.3.

4.2.1 Algorithm #1

From Section 2.1.2, oe(i)=y(i)–$\hat{y}$(i), where

$$\hat{y}(i) = \hat{\theta}(i-1)\varphi_{oe}(i) \qquad (4.3)$$

Now from (3.9) of Section 3.1.2, and assuming slow adaptation of the adaptive filter coefficients, the expression for the derivative of oe(i) (which is the transpose of the gradient) is:

$$\frac{doe(i)}{d\hat{\theta}(n)} \approx \frac{doe(i)}{d\hat{\theta}(n-1)} = \frac{-1}{\hat{A}\left(q^{-1},n-1\right)}\varphi_{oe}^{T}(i) = -\varphi_{oe}^{'T}(i) \qquad (4.4)$$

where the prime denotes autoregressive filtering by the denominator polynomial of the adaptive filter. Substituting (4.4) and (4.3) into (4.2) and dividing through by –2 yields:

$$\sum_{i=1}^{n}\left[y(i) - \hat{\theta}^T(n)\varphi_{oe}(i)\right]\varphi_{oe}'^T(i) = 0$$

Solving for $\hat{\theta}(n)$ gives:

$$\hat{\theta}(n) = \left[\sum_{i=1}^{n} \varphi_{oe}'(i)\varphi_{oe}^T(i)\right]^{-1} \sum_{i=1}^{n}\varphi_{oe}(i)y(i)$$

Similar to Section 3.2.1, a recursive formulation can be derived, yielding:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \mathbf{R}^{-1}(n)\varphi_{oe}'(n)oe(n) \tag{4.5a}$$

where

$$\mathbf{R}(n) = \sum_{i=1}^{n} \varphi_{oe}'(i)\varphi_{oe}^T(i) = \mathbf{R}(n-1) + \varphi_{oe}'(n)\varphi_{oe}^T(n) \tag{4.5b}$$

Examination of (4.5) shows this algorithm to be identical to the RPE algorithm, except for the construction of $\mathbf{R}(n)$. Here both a filtered and unfiltered version of the regressor, $\varphi_{oe}(n)$, is used.

4.2.2 Algorithm #2

For this algorithm, the alternative expression (2.13) is used for oe(n) in (4.2), which is repeated here:

$$oe(n) = \frac{1}{\hat{A}(q^{-1},n-1)}ee(n) = \frac{1}{\hat{A}(q^{-1},n-1)}\left[y(n)-\hat{\theta}^T(n-1)\varphi_{ee}(n)\right]$$

$$= y'(n) - \hat{\theta}^T(n-1)\varphi'_{ee}(n) \tag{4.6}$$

Here, again, a primed quantity stands for a quantity which is autoregressively filtered by the denominator polynomial of the adaptive filter.

Proceeding as before, (4.6) and (4.4) are now substituted into (4.2), yielding the following equation:

$$\sum_{i=1}^{n} \left[y'(i) - \hat{\theta}^T(n)\varphi'_{ee}(i) \right] \varphi'^T_{oe}(n) = 0$$

Solving for $\hat{\theta}(n)$ yields:

$$\hat{\theta}(n) = \left[\sum_{i=1}^{n} \varphi'_{oe}(i)\varphi'^T_{ee}(n) \right]^{-1} \sum_{i=1}^{n} y'(i)\varphi'_{oe}(i)$$

The recursive version of the above expression for $\hat{\theta}(n)$ is:

$$\hat{\theta}(n) = \hat{\theta}(n-1) + \mathbf{R}^{-1}(n)\varphi'_{oe}(n)\, oe(n) \tag{4.7a}$$

where

$$\mathbf{R}(n) = \sum_{i=1}^{n} \varphi'_{oe}(i)\varphi'^T_{ee}(i) = \mathbf{R}(n-1) + \varphi'_{oe}(n)\varphi'^T_{ee}(n) \tag{4.7b}$$

4.3.3 Discussion of the Algorithms

An interesting characteristic of these algorithms that distinguishes them from the RPE method is that they use two different vectors in the calculation of R(n). This feature is reminiscent of the instrumental variable method [Lj83, Sec.2.2.2]. The instrumental

variable method is an algorithm for adapting an *equation error* adaptive filter, and is identical to either (4.5) or (4.7) with the following substitutions:

$$\varphi'_{oe}(n) \overset{(4.5)}{\Leftarrow} \zeta(n) \overset{(4.7)}{\Rightarrow} \varphi'_{oe}(n)$$

$$\varphi'_{ee}(n) \overset{(4.5)}{\Leftarrow} \varphi_{ee}(n) \overset{(4.7)}{\Rightarrow} \varphi_{oe}(n)$$

$$oe(n) \overset{(4.5)}{\Leftarrow} ee(n) \overset{(4.7)}{\Rightarrow} oe(n)$$

The vector $\zeta(n)$ is called the *instrumental variable*, which can be chosen in many ways. See [Lj83, Sec.2.2.2 and 3.6.3] for discussions on this subject.

A final comment regarding the algorithm (4.7) is worth mentioning. Since this algorithm uses both the equation error regressor as well as the usual output error regressor, it would be interesting - and certainly exciting - to see whether this method exhibits characteristics of equation error adaptive schemes. Of particular interest is whether this algorithm possesses a unimodal performance surface such as the SMM algorithms, which also combine elements of equation error and output error schemes.

REFERENCES

[Ab88] A. S. Abutaleb, "An adaptive filter for noise cancelling," *IEEE Trans. Circuits Systems*, vol. CAS-35, no. 10, pp. 1201-1209, Oct. 1988.

[Da87] 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. Inform. Theory*, vol. IT-33, No. 3, pp. 383-392, May 1987.

[Fa86] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuits Systems*, vol. CAS-33, no. 10, pp. 939-947, Oct. 1986.

[Fa88] H. Fan and W. K. Jenkins, "An investigation of an adaptive IIR echo canceller: Advantages and problems," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-36, no. 12, pp. 1819-1834, Dec. 1988.

[Fa89] H. Fan and M. Nayeri, "Stability of some system identification techniques for underparameterized IIR adaptive filters," *Proc. 1989 IEEE Int. Symp. Circuits Syst.*, Portland, OR, vol. 3, pp.1748-1751, May, 1989.

[Fl77] W. Fleming, *Functions of Several Variables*, Springer-Verlag, N.Y., 1977.

[Fo82] E. Fogel and Y. F. Huang, On the value of information in system identification - bounded noise case," *Automatica*, vol. 18, No. 2, pp. 229-238, March 1982.

[Ha86] S. S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, N.J., 1986.

[Ho84] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*. Boston: Kluwer Academic, 1984.

[Hu86] Y. F. Huang, "A recursive estimation algorithm using selective updating for spectral analysis and adaptive signal processing," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-34, No. 5, pp. 1131-1134, Oct. 1986.

[Jo77] C. R. Johnson, Jr., and M. G. Larimore, "Comments on and additions to 'An adaptive recursive LMS filter,'" *Proc. IEEE*, vol. 65, no. 9, pp. 1399-1402, Sep. 1977.

[Jo84] C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 237-250, Mar. 1984.

[Ju64] E. I. Jury, *Theory and Applications of the Z-Transform Method.* Wiley, New York, 1964.

[Lj83] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification.* MIT Press, Cambridge, MA, 1983.

[Mi82] A. Michel, *Ordinary Differential Equations*, Academic Press, N. Y., 1982

[Op75] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Englewood Cliffs, N.J., Prentice Hall, 1975.

[Qu85] S. U. H. Qureshi, "Adaptive Equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349-1387, Sep. 1985.

[Rao89] A. K. Rao, "Membership-set parameter estimation via optimal bounding ellipsoids," Ph. D. dissertation, Dept. of Elec. and Comp. Eng., Univ. of Notre Dame, Notre Dame, IN, 1989

[Sh89] J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP magazine*, April, 1989.

[So75] T. Soderstrom, "On the uniqueness of maximum likelihood identification," *Automatica*, vol. 11, no. 2, pp. 193-197, Mar. 1975.

[So88] T. Soderstrom and P. Stoica, "On some system identification techniques for adaptive filtering," *IEEE Trans. Circuits Systems*, vol. CAS-35, no. 4, pp. 457-461, April 1988.

[Str81] S. D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Trans. Circuits Systems*, vol. CAS-28, no. 6, pp. 603-606, June 1981.

[Stc81] P. Stoica and T. Soderstrom, "The Steiglitz-McBride algorithm revisited – Convergence analysis and accuracy aspects," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 712-717, 1981.

[Wi75] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proc IEEE*, vol. 63, no. 12, pp. 1692-1716, Dec 1975.

[Wi76] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151-1162, Aug 1976.