REGRESSOR BASED ADAPTIVE INFINITE IMPULSE RESPONSE FILTERING

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Emrah ACAR
July 1997

REGRESSOR BASED ADAPTIVE INFINITE IMPULSE RESPONSE FILTERING

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Emrah Acar.

Emrah ACAR

July 1997

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Orhan Arıkan(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Erol Sezer

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ömer Morgül

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

REGRESSOR BASED ADAPTIVE INFINITE IMPULSE RESPONSE FILTERING

Emrah ACAR
M.S. in Electrical and Electronics Engineering
Supervisor: Assist. Prof. Dr. Orhan Arıkan
July 1997

Superior performance of fast recursive least squares (RLS) algorithms over the descent type least mean square (LMS) algorithms in the adaptation of FIR systems has not been realized in the adaptation of IIR systems. This is because of having noisy observations of the original system output resulting in significantly biased estimates of the system parameters. Here, we propose an adaptive IIR system structure consisting of two parts: a two–channel FIR adaptive filter whose parameters are updated by a RLS type algorithm, and an adaptive regressor which provides more reliable estimates to the original system output based on previous values of the adaptive system output and noisy observation of the original system output. Two different regressors are investigated and robust ways of adaptation of the regressor parameters are proposed. The performance of the proposed algorithms are compared with successful LMS type algorithms and it is found that in addition to the expected convergence speed up, the proposed algorithms provide better estimates to the system parameters at low SNR value. Also, the extended Kalman filtering approach is tailored to our application. Comparison of the proposed algorithms with the extended Kalman filter approach revealed that the proposed approaches provide improved estimates in systems with abrupt parameter changes.

Keywords: Adaptive IIR Filtering, ARX, RLS, Kalman Filtering

ÖZET

DOĞRULTUCU TABANLI UYARLAMALI SONSUZ İTMELİ SÜZGEÇLEME

Emrah ACAR
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Y. Doç. Dr. Orhan Arıkan
Temmuz 1997

Sonlu itmeli (FIR) süzgeçlerdeki hızlı ÖEK (RLS) algoritmalarının EOK (LMS) algoritmalarına göre olan üstün performansları, sonsuz itmeli (IIR) süzgeçlerin uyarlamasında henüz yer almamıştır. Bunun nedeni uyarlamalı süzgeçteki pürüzlü referans dolayısıyla sistem parametrelerinin yanlı olarak kestirilebilmesidir. Bu çalışmada, iki kısımdan oluşan bir uyarlamalı sonsuz itmeli süzgeçleme yapısı önerilmiştir: İlk kısım, dönüşüm temelli çok kanallı en küçük kare köşegen yapısındaki QR–MLSL algoritmasıyla uyarlanan iki kanallı sonlu itmeli süzgeç; ikinci kısım ise uyarlayan sistem sonuçları ve orjinal süzgecin pürüzlü referansları ışığında, gözlenemeyen gerçek referansın kestiriminde bulunan bir doğrultucudan oluşmuştur. İki farklı tip doğrultucu incelenmiş ve doğrultucu parametrelerinin dayanıklı belirlenme yolları önerilmiştir. Önerilen algoritmaların, bilinen metodlarla karşılaştırılması yapılmış ve yakınsama hızındaki artışın yanısıra düşük sinyal gürültü durumunda daha doğru kestirimler elde edilmiştir. Ayrıca genişletilmiş Kalman süzgeçleme yöntemi probleme uyarlanmıştır. Önerilen doğrultucu temelli algoritmalarla, genelleştirilmiş Kalman süzgeçlemenin karşılaştırılmasında, ani değişim gösteren sistemlerin tanımlanmasında önerilen algoritmaların daha yüksek başarımı olduğu gözlenmiştir.

Anahtar Kelimeler: Uyarlamalı süzgeçleme, Kalman süzgeçleme, ÖEK (RLS) algoritması

ACKNOWLEDGEMENT

I would like to express my deep gratitude to my supervisor Assist. Prof. Dr. Orhan Arıkan for his guidance, suggestions and valuable encouragement throughout the development of this thesis.

I would like to thank Prof. Dr. Erol Sezer and Assoc. Prof. Dr. Ömer Morgül for reading and commenting on the thesis and for the honor they gave me by presiding the jury.

I am also indebted to my family for their patience and support.

Sincere thanks are also extended to every close friends who have helped in the development of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Adaptive filters have found widespread use in many different signal processing applications where there is no reliable prior information on the system parameters or the parameters vary in time. Mainly because of its simplicity in implementations, a Finite Impulse Response (FIR) system structure is usually preferred for the adaptive filter. However, the choice of FIR structures severely limits the performance of the adaptive filters when the required adaptation necessitates the use of filters with poles as well as zeros. However, even in these cases, the natural choice of using an adaptive filter with Infinite Impulse Response (IIR) structure has not found much room in applications. The major reason behind this fact is the lack of IIR adaptation algorithms which robustly converge to the desired system parameters in a short time. The trade–off between the convergence and bias of the estimated system parameters has been the subject matter of many investigations on IIR adaptation approaches [1–6].

There are two main approaches to adaptive IIR filtering, based on two different definitions of the error sequence which is iteratively tried to be minimized by the adaptation algorithm. In the output error formulation, the error sequence is defined as the difference between the desired and the output sequences of an IIR filter whose parameters are adjusted iteratively by the adaptation algorithm. Although, the output error formulation is a very natural extension of the FIR adaptation concept, unlike the FIR case, the weighted least squares cost function is no more quadratic with respect to the adaptive IIR system parameters. This limits us to use

slowly converging gradient descent adaptation techniques which is not acceptable especially for systems whose parameter changes faster than the convergence of the adaptive system. Furthermore, the cost surface may have complicated local minima structure making it very difficult for the gradient descent algorithms to converge to the globally optimal IIR system parameters. Also, stability monitoring becomes a critical issue in the output error adaptation. Many of the difficulties of the output error formulation do not exists in the equation error formulation where the error sequence is defined as the difference between the desired sequence and the output of a two–channel FIR filter whose inputs are the available input sequence and one–sample delayed desired output sequence. Since the corresponding weighted least squares cost function is quadratic with respect to the two–channel FIR filter parameters, fast recursive least squares adaptation algorithms can be used to obtain the globally optimal system parameters. However, even in the sufficient order modeling, when there is an additive measurement noise in the desired sequence, the obtained results are biased estimates of the unknown system parameters.

In order to capture the beneficial features of the equation error formulation and reduce the bias in the converged parameters, various bias remedy approaches have been proposed [2–4, 6–8]. In some of these approaches the error sequence is defined as a convex combination of the equation and output error sequences [3]. Then, the least squares cost function is tried to be minimized by using a gradient descent algorithm based on the instantaneous gradient estimate. It has been shown that with a judicious choice of the convex combination parameter, significantly more accurate parameter estimation can be achieved [2, 3]. However, because of the use of an update strategy based on instantaneous gradient estimate, the speed of convergence of these algorithms is slow.

In this thesis, we propose an adaptive IIR system structure consisting of two parts: a two–channel FIR adaptive filter and an adaptive regressor which provides more reliable estimates to the original system output. As shown in Fig. 2.1, the two–channel FIR adaptive filter has as its inputs the input of the original system, $x(n)$, and the delayed output of the regressor, $\hat{w}(n)$, which is an estimate to the original system output, $w(n)$. This way, the parameters of the adaptive filter can be updated efficiently by using a multi–channel recursive least squares algorithm such as QR–MLSL [9, 10]. We consider two different adaptive regressors to provide reliable estimates to the original system output causally based on adaptive system output, $y(n)$, and noisy observations, $d(n)$. The first type of regressor provides an

estimate to $w(n)$ as a convex combination of the $y(n)$ and $d(n)$, where the convex combination parameter γ_n is adapted based on the convergence of the iterations. In the adaptation of γ_n, the regressor performs $O(N)$ multiplications where N is the order of the adaptive system. In the second type of regressor, a simplified Kalman filter is used to provide the estimate, $\hat{w}(n)$, to $w(n)$, where the required state space model of the system is obtained from the adapted system. The required number of multiplications of the Kalman regrssor is $O(N^2)$. In chapter 3, we investigate both regressors and provide robust ways of adapting their parameters. Also in the same chapter, the steps of the adaptation algorithms for both regressors are tabulated.

In chapter 4, the well known extended Kalman filter algorithm is tailored to our application [11–13]. It is shown that since the resultant algorithm requires no–matrix inversions, system parameters can be estimated by computing $O(N^2)$ multiplications. Also, a robust way of updating the required covariance matrices is provided.

In chapter 5, we provide extensive comparison results between the approaches investigated in this work and earlier proposed approaches to IIR adaptation [2, 3, 12, 13]. In chapter 6, we provide the conclusions of our work and address potential areas for future work.

Chapter 2

IIR System Model and Proposed Adaptive IIR Filter Structure

2.1 IIR System Model

As shown in Fig. 2.1, in a typical adaptive filtering application, input, $x(n)$, and noisy output, $d(n)$, of an unknown system are available for processing by an adaptive system to provide estimates, $y(n)$, to the output of the unknown system as time progresses. If the ultimate purpose is to keep track of the variation in the unknown system parameters, the required processing is called as adaptive system identification. However, there are many other important application areas of adaptive filtering such as adaptive prediction, noise cancelling, echo cancelling and channel equalization, where the primary purpose is not the estimation of the unknown system parameters [5, 14, 15]. The approaches we will investigate in this thesis are generally applicable in all these application areas.

In our investigation the unknown system or plant has an IIR model whose output can be compactly expressed as a function of its previous values and its input as:

$$w(n) = \sum_{j=1}^{N} a_j w(n-j) + \sum_{i=0}^{M} b_i x(n-i) = \underline{\theta}^T \underline{\phi}(n), \tag{2.1}$$

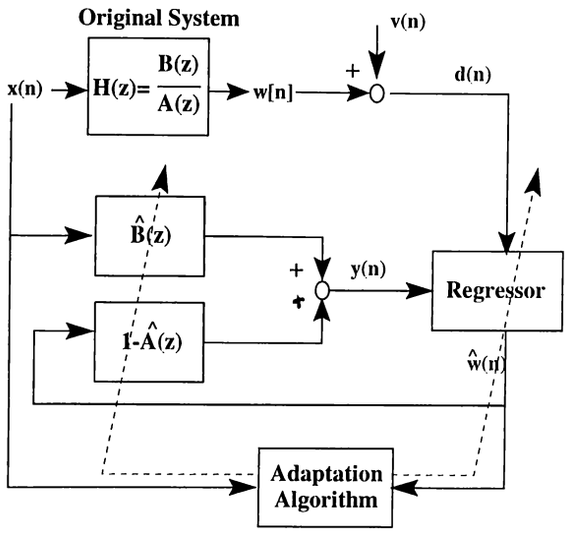


Figure 2.1: *Common structure of IIR–γ and IIR–Kalman adaptive systems.*

where $\underline{\theta}$ is the vector of direct form system parameters:

$$\underline{\theta} = \begin{bmatrix} a_1 & \cdots & a_N & b_0 & b_1 & & b_M \end{bmatrix}^T = \begin{bmatrix} \mathbf{a}^T & \mathbf{b}^T \end{bmatrix}^T, \tag{2.2}$$

and $\underline{\phi}(n)$ is formed by the previous values of the output and the present and past values of the input:

$$\begin{aligned} \underline{\phi}(n) &= \begin{bmatrix} w(n-1) & \cdots & w(n-N) & x(n) & x(n-1) & & x(n-M) \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{w}(n)^T & \mathbf{x}(n)^T \end{bmatrix}^T. \end{aligned} \tag{2.3}$$

In the above relation, the system is assumed to be time–invariant. Time varying systems can be modeled with $\underline{\theta}$ which has time–varying entries. Our aim is to develop adaptation algorithms that can be utilized for both time–invariant and time–varying systems. However, the variation of the system parameters should be either a slow function of time, or else, abrupt changes in the system parameters should occur infrequently in time.

2.2 A Regressor Based IIR Adaptive Filter Structure

In FIR adaptive filtering, the model is a tapped-delay-line and only the input samples determine the output of the plant and the model. No feedback loop exists inside the

system and stability in BIBO sense is always assured. In adaptive IIR filtering, due to the feedback existing in the system, we are faced with the problem of deciding on the feedback signal used in the adaptive system when we have noisy observations of the actual system output. Hence, as shown explicitly in Fig. 2.1, we need a regressor that causally performs the required estimation of the feedback signal based on the noisy output $d(n)$ and the output of the adaptive filter $y(n)$, which is obtained as:

$$y(n) = \underline{\hat{\theta}}^T(n)\underline{\hat{\phi}}(n) \tag{2.4}$$

where $\underline{\hat{\theta}}(n)$, the vector of estimated system parameters, can be written as:

$$\begin{aligned} \underline{\hat{\theta}}(n) &= \left[\begin{array}{cccccc} \hat{a}_1(n) & \cdots & \hat{a}_N(n) & \hat{b}_0(n) & \hat{b}_1(n) & \hat{b}_M(n) \end{array} \right]^T \\ &= \left[\begin{array}{cc} \hat{\mathbf{a}}(n)^T & \hat{\mathbf{b}}(n)^T \end{array} \right]^T \end{aligned} \tag{2.5}$$

and $\underline{\hat{\phi}}(n)$ is the vector of the regressor output, $\hat{w}(n)$, and the system input, $x(n)$:

$$\begin{aligned} \underline{\hat{\phi}}(n) &= \left[\begin{array}{cccccc} \hat{w}(n-1) & \cdots & \hat{w}(n-N) & x(n) & x(n-1) & x(n-M) \end{array} \right]^T \\ &= \left[\begin{array}{cc} \hat{\mathbf{w}}(n)^T & \mathbf{x}(n)^T \end{array} \right]^T. \end{aligned} \tag{2.6}$$

The performance of the adaptive filter heavily depends on how well the regressor provides estimates to the actual system output $w(n)$. The two well known formulations of adaptive IIR filtering, namely the output error (OE) and the equation error (EE) formulations, correspond to two different types of regressors.

In the OE formulation, the vector $\underline{\hat{\phi}}_O(n)$ is described as

$$\begin{aligned} \underline{\hat{\phi}}_O(n) &= \left[\begin{array}{cccccc} y(n-1) & \cdots & y(n-N) & x(n) & x(n-1) & x(n-M) \end{array} \right]^T \\ &= \left[\begin{array}{cc} \mathbf{y}(n)^T & \mathbf{x}(n)^T \end{array} \right]^T \end{aligned} \tag{2.7}$$

which corresponds to a regressor whose output is the output of the adaptive filter. In the EE formulation the signal vector, $\underline{\hat{\phi}}_E(n)$ is given as:

$$\begin{aligned} \underline{\hat{\phi}}_E(n) &= \left[\begin{array}{cccccc} d(n-1) & \cdots & d(n-N) & x(n) & x(n-1) & x(n-M) \end{array} \right]^T \\ &= \left[\begin{array}{cc} \mathbf{d}(n)^T & \mathbf{x}(n)^T \end{array} \right]^T \end{aligned} \tag{2.8}$$

which corresponds to a regressor whose output is the noisy observation of the system output, $d(n) = w(n) + v(n)$.

Since the least squares cost function of EE formulation is quadratic in terms of the parameter vector θ, fast converging recursive least squares techniques can be used in the adaptation. However, because of the additive measurement noise, $v(n)$, the converged parameter values are biased estimates of the actual system parameters [7,16–18]. In the OE formulation, the least squares cost function is not a quadratic function of the parameters. Hence, we are bound to use LMS type gradient descent techniques in the adaptation. When these LMS type adaptation algorithms converge to the global minima of the cost function, the obtained parameters are unbiased estimates of the cost function. Unfortunately, not only LMS type gradient adaptation methods converges slowly, but also, they may converge to a local minima of the cost surface. Various algorithms have been proposed to combine the beneficial features of the OE and EE formalism in one algorithm [1–4,6,7,19]. Notably, the bias remedy least mean square equation error (BRLE) [2] and the composite regressor algorithms (CRA) [3] are proposed to obtain low biased parameter estimates by using gradient descent type adaptation [11,16–18]. However, since the corresponding cost functions of these algorithms are not designed to be quadratic with respect to the parameters, recursive least squares techniques cannot be utilized to obtain fast converging estimates to the parameters.

In the first part of our work, we also try to combine the desired features of both OE and EE formalism in one formulation where the cost function is kept as a quadratic function of the parameters. As suggested in Fig. 2.1, this is achieved by choosing the adaptive filter as a two-channel FIR filter with inputs $x(n)$ and $\hat{w}(n-1)$. Then, the corresponding weighted least squares cost function becomes:

$$J(\underline{\theta}, n) = \sum_{k=1}^{n} (d(k) - \underline{\theta}^T \underline{\hat{\phi}}(k))^2 \lambda^{n-k}, \tag{2.9}$$

which is a quadratic function of θ, because $\underline{\hat{\phi}}(n)$ is a fixed sequence of vectors determined by the past parameter estimates $\underline{\hat{\theta}}(n-1), \underline{\hat{\theta}}(n-2), \cdots, \underline{\hat{\theta}}(0)$. Hence, efficient multi-channel FIR recursive least squares techniques can be used to obtain parameter estimates at time n, $\underline{\hat{\theta}}(n)$, as the minimizer of $J(\underline{\theta}, n)$.

In the following chapter, two different types of regressors will be investigated in detail and corresponding recursive least squares adaptation algorithms will be presented.

Chapter 3

Proposed Regressors

The performance of the regressor based IIR adaptation structure largely depends on how well the actual system output is estimated by the regressor in Fig. 2.1. In this chapter, we investigate in detail two types of regressors.

3.1 IIR$-\gamma$ Algorithm

In the first class, the regressor output is estimated as a convex combination of the noisy observations, $d(n)$ and the adaptive filter output, $y(n)$ as

$$\hat{w}(n) = \gamma_n d(n) + (1 - \gamma_n) y(n) \quad , \quad 0 \leq \gamma_n \leq 1 \tag{3.1}$$

where γ_n is the regression coefficient. In the following, the IIR adaptation algorithm which uses this type of regressor is referred to as IIR$-\gamma$.

The proper choice of γ_n should be based on a measure of the reliability of the estimated system parameters. A significant deviation of $y(n)$ from $d(n)$ is an indication that the system parameters are not reliably estimated, and hence, γ_n should be chosen close to 1, so that equation error type adaptation should take place. On the contrary, if $y(n)$ closely follows $d(n)$, then to reflect our level of confidence to the estimated system parameters, γ_n should be chosen close to 0, so that output error type adaptation should be performed. We propose to base the measure of

reliability of the estimated system parameters to the statistical significance of the observed deviation between $y(n)$ and $d(n)$ sequences. For this purpose, one way of choosing γ_n is based on weighted estimate of the expected energy of the error sequence $e(n) = d(n) - y(n)$:

$$L(n) = \frac{\sum_{i=0}^{n} \lambda_v^i e(n-i)^2}{\sum_{i=0}^{n} \lambda_v^i}, \tag{3.2}$$

where λ_v is an exponential forgetting factor that can improve the performance of the estimator. In this approach, the regressor parameter γ_n is an increasing function of $L(n)$, because large values of $L(n)$ is an indication of deviation from the true system parameters. In our investigation, we observed that the critical properties of the functional form between $L(n)$ and γ_n are the boundary values l_1 and l_2 such that $\gamma_n = 0$ if $L(n) < l_1$ and $\gamma_n = 1$ if $L(n) \geq l_2$. In between these two boundaries, various forms of increasing functions can be used. In order to determine which values for l_1 and l_2 should be used, we investigated the expected value of the $L(n)$ for the cases of $\gamma_n = 0$ and $\gamma_n = 1$, which correspond to output and equation error adaptation cases respectively. Assuming that $\gamma_n = 0$ and the estimated parameters have converged to the actual ones, the observed error sequence, $e(n)$ will be equal to $v(n)$, the additive Gaussian observation noise. Hence, the expected value of $L(n)$ will be σ_v^2, the variance of $v(n)$. Therefore, l_1 is chosen as σ_v^2. Likewise, when $\gamma_n = 1$ and convergence of weights are established, expected value of $L(n)$ is equal to the variance of $e(n)$ sequence for the equation error formulation. Since equation error $e_E(n)$ is related to the output error $e_O(n)$ as [2]:

$$e_E(n) = e_O(n) - \hat{\mathbf{a}}^T(n)\mathbf{e}_O(n), \tag{3.3}$$

when $v(n)$ is white noise, the variance of $e_E(n)$ can be written as:

$$\text{var } (e_E(n)) = \sigma_v^2 \left(1 + \sum_{i=1}^{N} \hat{a}_i^2(n)\right) \tag{3.4}$$

at the time of convergence to true parameters. Hence, we propose to use:

$$l_1 = \sigma_v^2 \quad , \quad l_2 = U\sigma_v^2(1 + \sum_{i=1}^{N} \hat{a}_i^2) \tag{3.5}$$

where $U > 1$ is introduced so that γ_n should not be kept fixed at 1 near the convergence point of the equation error adaptation.

For computational efficiency, the actual form of functional relation between $L(n)$ and γ_n is chosen as follows:

$$\gamma_n = \begin{cases} 0 & L(n) < l_1 \\ \kappa \frac{(L(n)-l_1)^p}{(\frac{l_2-l_1}{2})^p} & l_1 \leq L(n) < \frac{l_1+l_2}{2} \\ 1-(1-\kappa)\frac{(l_2-L(n))^p}{(\frac{l_2-l_1}{2})^p} & \frac{l_1+l_2}{2} \leq L(n) < l_2 \\ 1 & L(n) \geq l_2 \end{cases} \tag{3.6}$$

where κ and p are two parameters providing some control of the actual shape of the curve in between two boundaries l_1 and l_2. Fortunately, we observed that the behavior of the algorithm is not so sensitive to these shape parameters. For each iteration, this regression algorithm requires $(N+11)$ multiplications which is $O(N)$.

Some examples of the above functional relation (3.6) can be seen in Fig. 3.1 for various shape parameters.

3.2 IIR–Kalman Algorithm

In the second class, we consider a Kalman regressor structure based on the following state space model of the original system [11–13, 15]:

$$\mathbf{w}(n+1) = \mathcal{A}\mathbf{w}(n) + \mathcal{B}\mathbf{x}(n) \tag{3.7}$$

$$d(n) = \mathcal{C}\mathbf{w}(n+1) + v(n) \tag{3.8}$$

where $\mathcal{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$ and the state transition matrices are:

$$\mathcal{A} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_N \\ 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & & \\ 0 & & 1 & 0 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} b_0 & b_1 & & b_M \\ 0 & 0 & & 0 \\ & & & \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.9}$$

Since the actual parameters are unknown, we cannot use the state space model directly in the estimation of $w(n)$. However, if we form $\mathcal{A}$ and $\mathcal{B}$ matrices by using

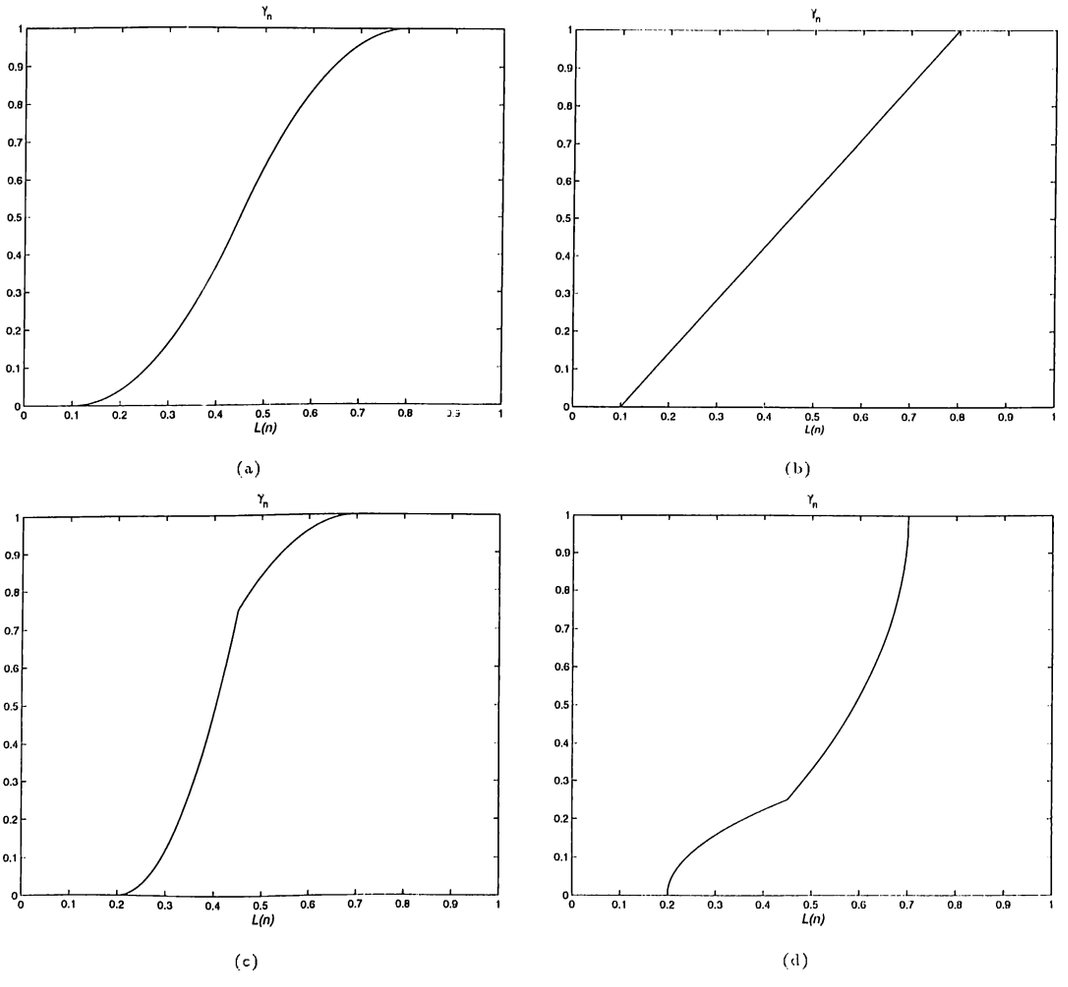


Figure 3.1: *Some example functional relation between* $L(n)$ *and* γ_n*: (a)* $l_1 = 0.1, l_2 = 0.8, \kappa = 0.5, p = 2$*, (b)* $l_1 = 0.1, l_2 = 0.8, \kappa = 0.5, p = 1$*, (c)* $l_1 = 0.2, l_2 = 0.7, \kappa = 0.75, p = 2$*, (d)* $l_1 = 0.2, l_2 = 0.7, \kappa = 0.25, p = 2$

the estimated parameters at time n, we get the following state space model:

$$\mathbf{w}(n+1) = \hat{\mathcal{A}}(n)\mathbf{w}(n) + \hat{\mathcal{B}}(n)\mathbf{x}(n) + \mathbf{u}(n) \quad (3.10)$$

$$d(n) = \mathcal{C}\mathbf{w}(n+1) + v(n) \quad (3.11)$$

where $\mathbf{u}(n)$ is introduced as an additional noise term to the system dynamics to account for the approximations in $\mathcal{A}$ and $\mathcal{B}$ by $\hat{\mathcal{A}}(n)$ and $\hat{\mathcal{B}}(n)$, which are equal to:

$$\hat{\mathcal{A}}(n) = \begin{bmatrix} -\hat{a}_1(n) & -\hat{a}_2(n) & \cdots & -\hat{a}_N(n) \\ 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & & \\ 0 & & 1 & 0 \end{bmatrix}, \quad \hat{\mathcal{B}}(n) = \begin{bmatrix} \hat{b}_0(n) & \hat{b}_1(n) & & \hat{b}_M(n) \\ 0 & 0 & & 0 \\ & & & \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.12)$$

Since the approximation in $\mathcal{A}(n)$ and $\mathcal{B}(n)$ only limited to the first row, the additional process noise $\mathbf{u}(n)$ can be written as:

$$\mathbf{u}(n) = \begin{bmatrix} u(n) & 0 & 0 \end{bmatrix}^T \quad (3.13)$$

In order to apply Kalman estimator on the approximate model given by Eqns. (3.10) and (3.11), we need the covariance matrices $\mathcal{R}_u(n)$ and $\mathcal{R}_v(n)$ of $u(n)$ and $v(n)$ respectively. In addition, we need an initial estimate to the state vector $\mathbf{w}(0)$ and the variance of the initial system error $\mathcal{R}_u(0)$. The covariance matrix $\mathcal{R}_u(n)$ is determined by the variance of $u(n)$ for which a robust way of approximation is presented in the Appendix A. The steps of the corresponding Kalman estimator are given in Table 3.1, where $\hat{\mathcal{A}}(n), \hat{\mathcal{B}}(n), \hat{\mathbf{w}}(n)$ are defined in Eqns. (3.12), (2.6) and the notation of $\mathcal{T}_{(1,1)}$ is used to denote the first diagonal entry of the matrix $\mathcal{T}$. Note that the output of the regressor $\hat{w}(n)$ is the first entry in the estimated state vector $\hat{\mathbf{w}}_K(n+1)$ and also the a–priori state estimate $\hat{\mathbf{w}}_K(n+1|n)$ is obtained efficiently by using the output of the adaptive filter and the previous states of the Kalman filter. The actual forms of the matrices in the above algorithm can be exploited for more efficient computation of the regressor output $\hat{w}(n)$. For each iteration, the Kalman regressor requires $(3N^2 + 2N)$ multiplications, hence it is $O(N^2)$. The IIR adaptation algorithm which uses this type of regressor is referred to as IIR–Kalman.

The required two–channel FIR adaptation can be efficiently performed by using QR–MLSL algorithm which is a rotation–based multi–channel least squares lattice

$$
\begin{aligned}
\hat{\mathbf{w}}_K(n+1|n) &= \begin{bmatrix} y(n) & \hat{w}_K(n-1) & \cdots & \hat{w}_K(n-N+1) \end{bmatrix}^T \\
\mathcal{P}(n+1|n) &= \hat{\mathcal{A}}(n)\mathcal{P}(n|n)\hat{\mathcal{A}}(n)^T + \mathcal{R}_u(n) \\
\mathcal{G}(n) &= \frac{\mathcal{P}(n+1|n)}{\mathcal{P}(n+1|n)_{(1,1)} + \sigma_v^2} \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \\
\mathcal{P}(n+1|n+1) &= \left(\mathbf{I} - \mathcal{G}(n) \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}\right) \mathcal{P}(n+1|n) \\
\hat{\mathbf{w}}_K(n+1) &= \hat{\mathbf{w}}_K(n+1|n) + \mathcal{G}(n)\left(d(n) - \hat{\mathbf{w}}_K(n+1|n)_{(1,1)}\right) \\
\hat{w}(n) &= \hat{\mathbf{w}}_K(n+1)_{(1,1)}
\end{aligned}
$$

Table 3.1: *Equations of IIR–Kalman State Estimator*

algorithm with many desired features [9]. The steps of this algorithm are given in Table 3.2. For each update, this algorithm requires $O(4N)$ multiplications. The required direct form parameters for the Kalman regressor can be easily computed by using standard mapping rules between lattice and direct form parameters [9]. The structure of multi-channel lattice FIR filter and the mapping rule is explained in the Appendix B.

Initializations:

$$\mathbf{R}^{\mathbf{f}}{}_p(0) = \mathbf{R}^{\mathbf{b}}{}_p(0) = \sqrt{\delta}\mathbf{I}_{2\times 2} \quad \mathbf{\Gamma}^{\mathbf{f}}{}_p(0) = \mathbf{\Gamma}^{\mathbf{b}}{}_p(0) = \mathbf{0}_{2\times 2} \quad \underline{\mathbf{\Gamma}}^{\mathbf{e}}{}_p(0) = \mathbf{0}_{2\times 1}$$

Time recursive equations: For $n = 1, 2, \ldots$

$$\tilde{\mathbf{f}}_0(n) = \tilde{\mathbf{b}}_0(n) = \mathbf{x}(n) = \begin{bmatrix} x_1(n) & x_2(n) \end{bmatrix}^T \quad \tilde{e}_0(n) = y(n) \quad \tilde{\gamma}_0(t) = 1$$

Order recursive equations: For $p = 1, 2, \ldots, p_{max}$

$$\mathbf{Q}_p^f(n) \begin{bmatrix} \sqrt{\lambda}\mathbf{R}^{\mathbf{f}}{}_p(n-1) & \sqrt{\lambda}\mathbf{\Gamma}^{\mathbf{f}}{}_p(n-1) \\ \tilde{\mathbf{f}}_{p-1}^T(n) & \tilde{\mathbf{b}}_{p-1}^T(n-1) \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{\mathbf{f}}{}_p(n) & \mathbf{\Gamma}^{\mathbf{f}}{}_p(n) \\ \mathbf{0}^T & \tilde{\mathbf{b}}_p^T(n) \end{bmatrix}$$

$$\mathbf{Q}_p^b(n) \begin{bmatrix} \sqrt{\lambda}\mathbf{R}^{\mathbf{b}}{}_p(n-1) & \sqrt{\lambda}\mathbf{\Gamma}^{\mathbf{b}}{}_p(n-1) & \sqrt{\lambda}\underline{\mathbf{\Gamma}}^{\mathbf{e}}{}_p(n-1) \\ \tilde{\mathbf{b}}_{p-1}^T(n-1) & \tilde{\mathbf{f}}_{p-1}^T(n) & \tilde{e}_{p-1}(n-1) \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{\mathbf{b}}{}_p(n) & \mathbf{\Gamma}^{\mathbf{b}}{}_p(n) & \underline{\mathbf{\Gamma}}^{\mathbf{e}}{}_p(n) \\ \mathbf{0}^T & \tilde{\mathbf{f}}_p^T(n) & \tilde{e}_p(n-1) \end{bmatrix}$$

$$\tilde{\gamma}_p(n-1) = \tilde{\gamma}_{p-1}(n-1)\cos\theta_{p,1}^b(n)\cos\theta_{p,2}^b(n)$$

Parameter Identification:

$$\mathbf{K}^{\mathbf{f}}{}_p(n) = \mathbf{R}^{\mathbf{f}}{}_p(n)^{-1}\mathbf{\Gamma}^{\mathbf{f}}{}_p(n)$$

$$\mathbf{K}^{\mathbf{b}}{}_p(n) = \mathbf{R}^{\mathbf{b}}{}_p(n)^{-1}\mathbf{\Gamma}^{\mathbf{b}}{}_p(n)$$

$$\underline{\mathbf{K}}^{\mathbf{e}}{}_p(n) = \mathbf{R}^{\mathbf{b}}{}_p(n)^{-1}\underline{\mathbf{\Gamma}}^{\mathbf{e}}{}_p(n)$$

Used Transformations:

$$\tilde{\mathbf{f}}_p(n) = \mathbf{f}_p(n)/\tilde{\gamma}_p(n-1) \qquad \tilde{\mathbf{b}}_p(n) = \mathbf{b}_p(n)/\tilde{\gamma}_p(n) \qquad \tilde{e}_p(n) = e_p(n)/\tilde{\gamma}_p(n))$$

$\theta_{p,1}^b, \theta_{p,2}^b$ are the angles of Givens rotation corresponding to $\mathbf{Q}_p^b(n)$.

The last diagonal element of $\mathbf{Q}_p^b(n)$is the products of cosines.

Table 3.2: *QR-MLSL Algorithm and Parameter Identification in the Two-Channel Lattice Form*

Chapter 4

System Identification by Extended Kalman Algorithm

We propose the regressor based RLS algorithms for their potential of providing more reliable feedback signal $\hat{w}(n)$ in the presence of output noise. In the IIR-Kalman algorithm, a boot-strap method is used for an alternating estimation of the system output and its parameters. The Kalman regressor provides estimates to the noise free output, and then an RLS type adaptation procedure first updates the adaptive system parameters and then compute the output of the adaptive system, $y(n)$. As discussed in [11–13], these two stages of the adaptation can be combined into one in an augmented state space description of the system. This approach has been proposed for combined state estimation and tracking of slowly varying system parameters once a close initial estimate to the system parameters is available [12, 13]. In this chapter, we provide the augmented state space description corresponding to IIR adaptive filtering and then derive the corresponding extended Kalman algorithm for the estimation of the augmented state. Also, we use a robust method, which is presented in the Appendix A, for the choice of the required covariance matrices.

The augmented state space description which will be exploited for joint estimation of the system output and its parameters is given as:

$$\begin{bmatrix} \mathbf{w}(n+1) \\ \underline{\theta}(n+1) \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{A}}(n)\mathbf{w}(n) + \hat{\mathcal{B}}(n)\mathbf{x}(n) \\ \underline{\theta}(n) \end{bmatrix} + \begin{bmatrix} \mathbf{u}(n) \\ \mathbf{s}(n) \end{bmatrix} \tag{4.1}$$

with the corresponding observation model of:

$$d(n) = \begin{bmatrix} \mathcal{C} & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{w}(n+1) \\ \hat{\underline{\theta}}(n+1) \end{bmatrix} + v(n). \tag{4.2}$$

Here, $\mathbf{u}(n)$ is the noise sequence vector on the output estimates, which we call as process noise as in IIR–Kalman framework, and $\mathbf{s}(n)$, which is assumed to be uncorrelated with $\mathbf{u}(n)$, is the noise vector on the parameter updates. Since $\hat{\mathcal{A}}(n)\mathbf{w}(n)$ involves multiplication of augmented state variables, extended Kalman filter algorithm should be used in recursive estimation of the augmented state variables.

For the following general state space model, the extended Kalman filter has been proposed for efficient estimation of the state

$$\mathbf{z}_{n+1} = \mathcal{F}_n(\mathbf{z}_n) + \mathcal{H}_n(\mathbf{z}_n)\underline{\xi}_n \tag{4.3}$$

$$\mathbf{r}_n = \mathcal{X}_n(\mathbf{z}_n) + \underline{\eta}_n \tag{4.4}$$

where $\mathcal{F}_n$ and $\mathcal{X}_n$ are vector–valued functions and $\mathcal{H}_n$ is a matrix–valued function with continuous first–order partial derivatives. In the case of zero–mean uncorrelated Gaussian noise sequences, $\underline{\xi}_n$ and $\underline{\eta}_n$, with

$$E\{\underline{\xi}_k\underline{\xi}_l^T\} = \mathcal{Q}_k\delta(k-l)\ ,\ E\{\underline{\eta}_k\underline{\eta}_l^T\} = \mathcal{S}_k\delta(k-l)\ ,\ E\{\begin{bmatrix} \underline{\xi}_k \\ \underline{\eta}_k \end{bmatrix}\mathbf{z}_0^T\} = 0 \quad \forall\, k,l \tag{4.5}$$

the steps of the extended Kalman filter algorithm are given in Table 4.1, as it was derived in [12].

This general form of the extended Kalman filter can be specialized to our application by using the following substitutions:

$$\begin{aligned} \mathbf{z}_n &= \begin{bmatrix} \mathbf{w}(n)^T & \theta(n)^T \end{bmatrix}^T \quad , \quad \mathbf{r}_n = d(n) \\ \mathcal{F}_n(\mathbf{z}_n) &= \begin{bmatrix} \left(\hat{\mathcal{A}}(n)\mathbf{w}(n) + \hat{\mathcal{B}}(n)\mathbf{x}(n)\right)^T & \underline{0}^T(n) \end{bmatrix}^T \quad , \quad \mathcal{H}_n(\mathbf{z}_n) = \mathbf{I} \\ \mathcal{X}_n(\mathbf{z}_n) &= \mathcal{C}\mathbf{w}(n) \quad , \quad \underline{\xi}_n = \begin{bmatrix} \mathbf{u}(n)^T & \mathbf{s}(n)^T \end{bmatrix}^T \quad , \quad \underline{\eta}_n = v(n). \end{aligned} \tag{4.6}$$

The steps of the corresponding algorithm, that will be referred to as EKF, is given in Table 4.2. As seen from this table, the initial estimates of the states, the covariance matrices of the initial state estimate, the system and observation noises are required. A robust way of approximating the covariance matrix $\mathcal{R}_u(n)$ is presented in the previous chapter and the Appendix A. Since the matrices have special structures, we can simplify the required computational complexity of the EKF. For instance, both $\mathbf{u}(n)$ and the observation matrix has only one non-zero entry, simplifying the vector–matrix operations. Hence, no matrix inversion is required in our application.

For each update, the EKF algorithm requires $(12N^2 + 3M^2 + 12NM + 17N + 9M + 4)$ multiplications which is an order more than that of IIR$-\gamma$ and around 9 times more than that of IIR–Kalman algorithms.

$$
\begin{aligned}
\text{Initialization: } \mathcal{P}_{0,0} &= E\{\mathbf{z}_0\mathbf{z}_0^T\} \quad , \quad \hat{\mathbf{z}}_0 = E\{\mathbf{z}_0\} \\
\mathcal{P}_{n,n-1} &= \left[\frac{\partial \mathcal{F}_{n-1}}{\partial \mathbf{z}_{n-1}}(\hat{\mathbf{z}}_{n-1})\right] \mathcal{P}_{n-1,n-1} \left[\frac{\partial \mathcal{F}_{n-1}}{\partial \mathbf{z}_{n-1}}(\hat{\mathbf{z}}_{n-1})\right]^T \\
&\quad + \mathcal{H}_{n-1}(\hat{\mathbf{z}}_{n-1})\mathcal{Q}_{n-1}\mathcal{H}_{n-1}^T(\hat{\mathbf{z}}_{n-1}) \\
\hat{\mathbf{z}}_{n|n-1} &= \mathcal{F}_{n-1}(\hat{\mathbf{z}}_{n-1}) \\
\mathcal{G}_n &= \mathcal{P}_{n,n-1}\left[\frac{\partial \mathcal{X}_n}{\partial \hat{\mathbf{z}}_n}(\hat{\mathbf{z}}_{n|n-1})\right]^T \\
&\quad \left[\left[\frac{\partial \mathcal{X}_n}{\partial \mathbf{z}_n}(\hat{\mathbf{z}}_{n|n-1})\right] \mathcal{P}_{n,n-1} \left[\frac{\partial \mathcal{X}_n}{\partial \mathbf{z}_n}(\hat{\mathbf{z}}_{n|n-1})\right]^T + \mathcal{S}_n\right]^{-1} \\
\mathcal{P}_{n,n} &= \left[\mathbf{I} - \mathcal{G}_n\left[\frac{\partial \mathcal{X}_n}{\partial \mathbf{z}_n}(\hat{\mathbf{z}}_{n|n-1})\right]\right] \mathcal{P}_{n,n-1} \\
\hat{\mathbf{z}}_{n|n} &= \hat{\mathbf{z}}_{n|n-1} + \mathcal{G}_n\left(\mathbf{r}_n - \mathcal{X}_n(\hat{\mathbf{z}}_{n|n-1})\right)
\end{aligned}
$$

Table 4.1: *Equations of General Extended Kalman Algorithm*

In the following chapter, we provide extensive comparison results between the presented algorithms and LMS type regression algorithms: CRA and BRLE.

$$
\begin{aligned}
\text{Initialization: } \begin{bmatrix} \hat{\mathbf{w}}(0) \\ \hat{\underline{\theta}}(0) \end{bmatrix} &= \begin{bmatrix} E\{\mathbf{w}(0)\} \\ \hat{\underline{\theta}}(0) \end{bmatrix}, \quad \mathcal{P}(0|0) = \begin{bmatrix} \mathcal{R}_{\mathbf{w}}(0) & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_{\underline{\theta}}(0) \end{bmatrix} \\
\begin{bmatrix} \hat{\mathbf{w}}(n+1|n) \\ \hat{\underline{\theta}}(n+1|n) \end{bmatrix} &= \begin{bmatrix} \hat{\mathcal{A}}(n)\hat{\mathbf{w}}(n) \\ \hat{\underline{\theta}}(n) \end{bmatrix} + \begin{bmatrix} \hat{\mathcal{B}}(n)\mathbf{x}(n) \\ \mathbf{0} \end{bmatrix} \\
\mathcal{P}(n+1|n) &= \mathcal{J}(n)\mathcal{P}(n|n)\mathcal{J}(n)^T + \begin{bmatrix} \mathcal{R}_u(n) & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_\theta(n) \end{bmatrix} \\
\mathcal{J}(n) &= \begin{bmatrix} \hat{\mathcal{A}}(n) & \begin{bmatrix} \hat{\mathbf{w}}(n)^T & \mathbf{x}(n)^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \mathbf{0} & \mathbf{I}_{M+N+1} \end{bmatrix} \\
\mathcal{G}(n) &= \frac{\mathcal{P}(n+1|n)}{\left(\mathcal{P}(n+1|n)_{(1,1)} + \sigma_v^2\right)} \begin{bmatrix} 1 & 0 & & 0 \end{bmatrix}^T \\
\mathcal{P}(n+1|n+1) &= \left(\mathbf{I} - \mathcal{G}(n) \begin{bmatrix} 1 & 0 & & 0 \end{bmatrix}\right) \mathcal{P}(n+1|n) \\
\begin{bmatrix} \hat{\mathbf{w}}(n+1) \\ \hat{\underline{\theta}}(n+1) \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{w}}(n+1|n) \\ \hat{\underline{\theta}}(n+1|n) \end{bmatrix} + \mathcal{G}(n)\left(d(n) - \hat{\mathbf{w}}(n+1|n)_{(1,1)}\right)
\end{aligned}
$$

Table 4.2: *Equations of Extended Kalman Algorithm Applied in Adaptive IIR Filtering*

Chapter 5

Simulation Experiments

In order to compare the regressor based fast RLS algorithms proposed in chapter 3 and the EKF algorithm discussed in chapter 4 with the earlier proposed gradient descent IIR adaptation algorithms BRLE [2] and CRA [3], their performances over synthetically generated examples are given in this chapter.

The steps of BRLE and CRA algorithms are given in Table 5.1 and Table 5.2.

$$
\begin{aligned}
e_O(n) &= d(n) - \underline{\hat{\phi}}_O^T(n)\underline{\hat{\theta}}(n) \\
e_E(n) &= d(n) - \underline{\hat{\phi}}_E^T(n)\underline{\hat{\theta}}(n) \\
\hat{a}_i(n+1) &= \hat{a}_i(n) + \mu e_E(n)\left[d(n-i) - \tau e_O(n-i)\right] \quad i = 1,\ 2, \cdots,\ N \\
\hat{b}_j(n+1) &= \hat{b}_i(n+1) + \mu e_E(n) x(n-j) \quad j = 0,\ 1, \cdots,\ M \\
\text{Remedy Parameter:} \quad & 0 \leq \tau \leq 1 \\
& \text{determined by} \quad \tau = \min(k\frac{\|\underline{\hat{\phi}}(n)\|}{\|\mathbf{e}_O(n)\|}, 1)
\end{aligned}
$$

Table 5.1: *Equations of Bias–Remedy Least Mean Square Equation Error Algorithm (BRLE)*

For the required multi–channel RLS adaptation of the system parameters in the regressor based algorithms, rotation–based multi–channel least squares lattice algorithm (QR–MLSL) given in Table 3.2, is used [9]. By using simple transformation

$$
\begin{aligned}
\hat{\mathbf{w}}(n) &= \gamma \mathbf{d}(n) + (1-\gamma)\mathbf{y}(n) \\
\hat{\underline{\phi}}(n) &= \left[\begin{array}{cc} \hat{\mathbf{w}}(n)^T & \mathbf{x}(n)^T \end{array} \right]^T \\
e(n) &= d(n) - \hat{\underline{\phi}}^T(n)\hat{\underline{\theta}}(n) \\
\hat{\underline{\theta}}(n+1) &= \hat{\underline{\theta}}(n) + \frac{\mu \hat{\underline{\phi}}(n) e(n)}{1 + \mu \hat{\underline{\phi}}^T(n)\hat{\underline{\phi}}(n)} \\
\text{Composition Parameter:} & \quad 0 \leq \gamma \leq 1
\end{aligned}
$$

Table 5.2: *Equations of Composite Regressor Algorithm (CRA)*

rules, the direct form parameters can be obtained from the reflection matrices of the adapted two-channel FIR lattice filter [9]. In the following results, the parameter error vectors are computed as:

$$\mathbf{e}_{\theta}(n) = \underline{\theta} - \hat{\underline{\theta}}(n) \tag{5.1}$$

where θ is the actual and $\hat{\underline{\theta}}(n)$ is the estimated direct form parameters.

The adaptive filters are "all-zero" initialized during each experiment. The statistical results come from the the ensemble average of 50 realizations.

5.1 Simulation Example 1

In this first example, the same LTI second order IIR system analyzed in [2, 19] is used in a system identification application. The transfer function of the original system is:

$$H(z) = \frac{1}{1 - 1.7z^{-1} + 0.7225z^{-2}}. \tag{5.2}$$

The input sequence is a unit-variance white Gaussian process. The output noise process, $v(n)$, is chosen as white Gaussian noise process. The output noise variance is varied to investigate the sensitivity of the performance of the algorithms to the level of SNR.

In Fig. 5.1, the squared norms of the parameter error vectors, $\|\mathbf{e}_{\theta}(n)\|^2$ corresponding to the compared algorithms are plotted as a function of time. The standard

deviation of the output noise, σ_v is set to 0.5. The forgetting factor λ of the QR–MLSL algorithm is chosen as 0.999, and the parameters of the regressor subsystem of Eqns. (3.5) and (3.6) are chosen as $\lambda_v = 0.9, p = 1, \kappa = 0.7, U = 2$. For the IIR–Kalman regression algorithm, the initial variance estimate, $\tilde{\sigma}_u^2(0)$ is chosen as unity and the smoothing factor, β is chosen as 0.9. The EKF algorithm has also initialized with all–zero initialization for the augmented state vector with the same smoothing factor and unit $\tilde{\sigma}_u^2(0)$ as well as the diagonal entries of the covariance matrix $\mathcal{R}_\theta(0)$. In order to better resolve the early convergence behaviors of the compared algorithms, a logarithmic time axis is used in Fig. 5.1. As seen from this figure, the proposed algorithms have converged to an error level of -10 dB earlier than the 1000^{th} sample, but the LMS type algorithms converge to the same error level at about 40000^{th} sample. EKF algorithm, performing the best, converges to -20 dB at around 50000^{th} sample. Here, the same step–size of 0.0005 is used for the CRA and BRLE algorithms. As recommended in [2] and [3], the composition parameter γ for CRA is chosen as 0.9, and the remedier parameter of BRLE, $\tau(n)$ is chosen as $\min(\frac{\|\hat{\underline{\phi}}(n)\|}{\|\mathbf{e}_O(n)\|}, 1)$.

Although the corresponding results of RLS equation error and output error adaptation are not shown in Fig. 5.1, they converged to error levels of -7 dB and 5 dB respectively, which are significantly higher than those of compared algorithms here. Therefore, as initially expected, the performance of the regressor based RLS approaches can be better than both the equation and output error formulations.

The tracking errors plot, $\|w(n) - y(n)\|^2$ of the compared algorithms are shown in Fig. 5.1. EKF and CRA algorithms have lower tracking errors, but the proposed regressor algorithms have a very fast convergence.

In order to compare the error on the parameters at the convergence of the algorithms, we repeated this experiment at different noise levels. The obtained $\|\mathbf{e}_\theta(n)\|^2$ results are given in Table 5.3 and figured in Fig. 5.2. In this experiment the best performing algorithm is found as the EKF algorithm. However, EKF requires an order more multiplications than IIR$-\gamma$ algorithm. As seen from these results, at high SNR (low levels of σ_v), LMS type algorithms converge to lower error levels. However, as the SNR decreases (high values of σ_v) the proposed algorithms start providing closer or better results than LMS type algorithms, which is an important advantage in many practical applications. Note that, the tabulated results correspond to the error levels at the 5000^{th} sample for the proposed algorithms and 50000^{th} samples

for the EKF, CRA and BRLE algorithms. Since, in many important applications, the speed of convergence is very critical, the proposed algorithms provide a good trade–off between error levels and the speed of convergence even at high SNR. Also, IIR$-\gamma$ provides comparable results to IIR–Kalman although it requires an order less number of multiplications.

σ_v	IIR$-\gamma$	IIR–Kalman	EKF	CRA	BRLE
0.0500	-74.5756	-82.2542	-74.5538	-84.9733	-77.3835
0.1000	-50.9257	-56.2936	-69.4997	-62.5078	-55.7208
0.2500	-24.1973	-27.1816	-38.8864	-32.1684	-28.3855
0.5000	-10.2770	-10.9023	-21.2782	-11.3103	-10.2322
1.0000	-1.4034	0.4040	-5.6237	3.6105	3.5396

Table 5.3: *Squared parameter error norm in dB at convergence of the algorithms at different noise levels for Example 1*

5.2 Simulation Example 2

In this example, the performance of the algorithms are compared when there is an abrupt change in the system parameters. For this purpose, we used the following time–varying transfer function for the original system:

$$H(z,n) = \begin{cases} \frac{0.2759+0.5121z^{-1}+0.5121z^{-2}+0.2759z^{-3}}{1-0.001z^{-1}+0.6546z^{-2}-0.0775z^{-3}} & n < 500 \\ \frac{0.7241+0.4879z^{-1}+0.4879z^{-2}+0.7241z^{-3}}{1+0.001z^{-1}+0.6546z^{-2}+0.0775z^{-3}} & n \geq 500 \end{cases} \tag{5.3}$$

The input is chosen as zero–mean unit variance white Gaussian process. The output noise $v(n)$ is chosen as a zero–mean white Gaussian noise with a variance of 0.25. The step–size of CRA and BRLE algorithms is set to 0.01, because a larger value for it would cause instability in the convergence. The composition parameter, γ of CRA is set to 0.5, and the remedier parameter, $\tau(n)$ of BRLE is determined as in the first example. The forgetting factor of the proposed algorithms is set to 0.99 for a better tracking of the variations in the system parameters. For IIR–Kalman algorithm, $\tilde{\sigma}_u^2(n)$ is chosen as unity and smoothing factor, β, as 0.9. The parameters of IIR$-\gamma$ in Eqns. (3.5) and (3.6) are chosen as $\lambda_v = 0.95, p = 1, \kappa = 0.3, U = 5$. EKF algorithm was also initialized with all–unit variances for all the elements in the state vector with the same smoothing factor of IIR–Kalman. In Fig. 5.4, $\|\mathbf{e}_\theta(n)\|^2$ of each algorithm is shown. As seen from these results, both CRA and BRLE, whose performance are

very close to each other, are outperformed by the proposed algorithms. IIR–γ and IIR–Kalman have the best performance where EKF algorithm has converged to a higher error level. This is an expected result in the light of the first experiment where the convergence speed of the proposed algorithms were found as significantly faster than BRLE and CRA algorithms. Again, at an orderless amount of multiplications, IIR–γ provides comparable results to IIR–Kalman.

5.3 Simulation Example 3

In this example, the performance of the proposed regressor algorithms are compared with the output error and equation error regressors when there is an abrupt change in the system as in the previous example. The time–varying system transfer function of the simple one–pole system:

$$H(z,n) = \begin{cases} \frac{1}{1+0.98z^{-1}} & n < 400 \\ \frac{1}{1-0.98z^{-1}} & n \geq 400 \end{cases} \tag{5.4}$$

The input sequence is chosen as in the second example. The output $w(n)$ is disturbed by a zero–mean white Gaussian noise process with $\sigma_v = 3$. The parameters of the regressor based algorithms are set as in the second example except the forgetting factor of the adaptation algorithm λ, which is chosen as 0.95. The output error and equation error regressor methods are also combined with the same adaptation algorithm, QR–MLSL, with the same forgetting factor. The corresponding $\|\mathbf{e}_\theta(n)\|^2$ performance and the tracking performance, $\|w(n) - y(n)\|^2$ of each algorithm are shown in Fig. 5.5.a and Fig. 5.5.b. The shown results are ensemble averages of 250 realizations. As seen from Fig. 5.5.a, the proposed regressor algorithm provide more reliable estimates to the system parameters than the equation and output error regressors. Following the abrupt change in the system, the proposed regressor algorithms track the equation error regressor for a short time, attaining a fast convergence then they keep reducing the estimation error even after the convergence of equation error regressor. At convergence, the proposed regressor algorithms have the lowest error level in the estimated parameters. Similar conclusions on the tracking performance of the compared algorithms can be drawn from Fig. 5.5.b, where the proposed algorithms converges rapidly to lower error levels in output tracking. This example demonstrated one more time the improved performance of the proposed regressor algorithms in time–varying systems.

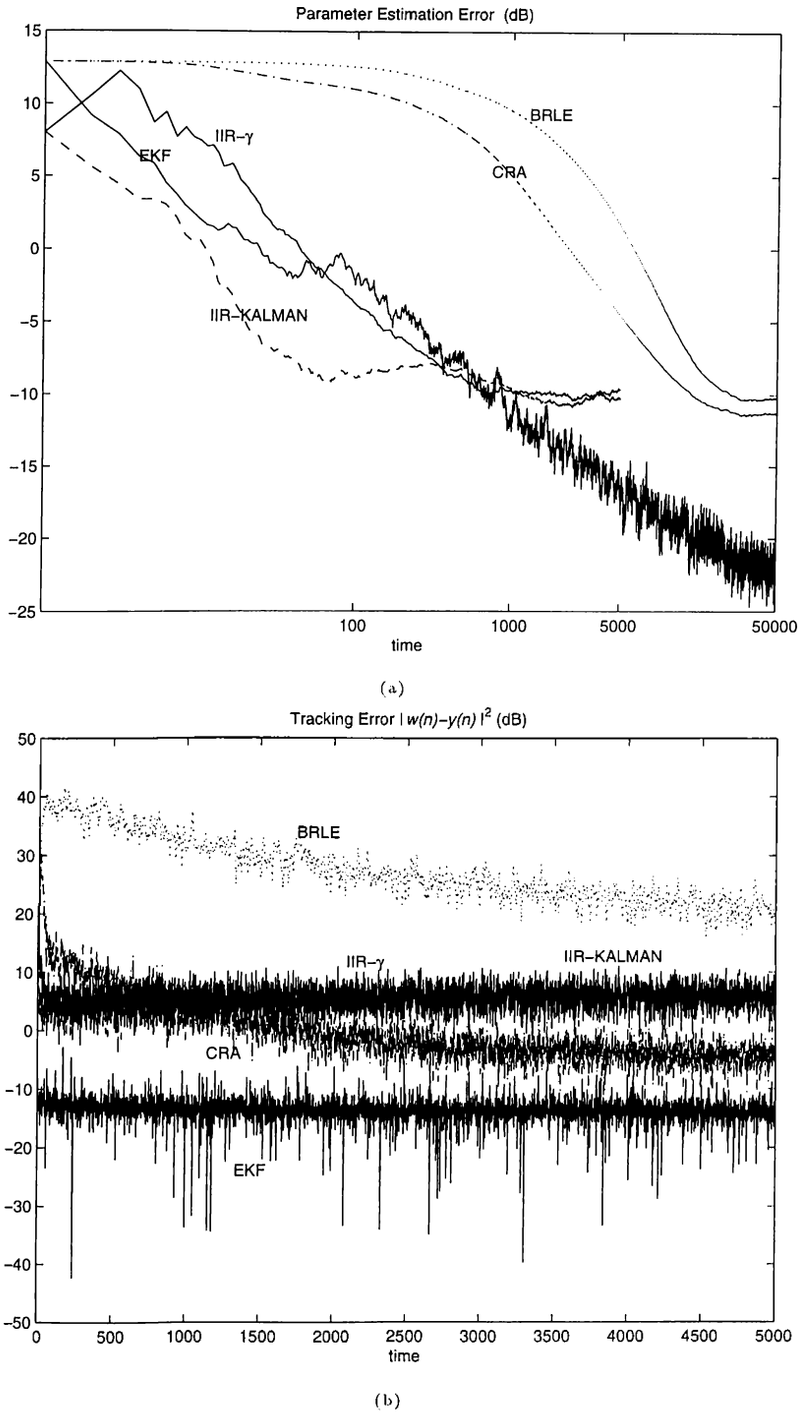


Figure 5.1: *Results of first example: (a) squared norm of parameter error as a function of time. Logarithmic time axis is used to resolve early convergence behavior of the algorithms, (b) output tracking error* $\|w(n) - y(n)\|^2$ *as a function of time.*

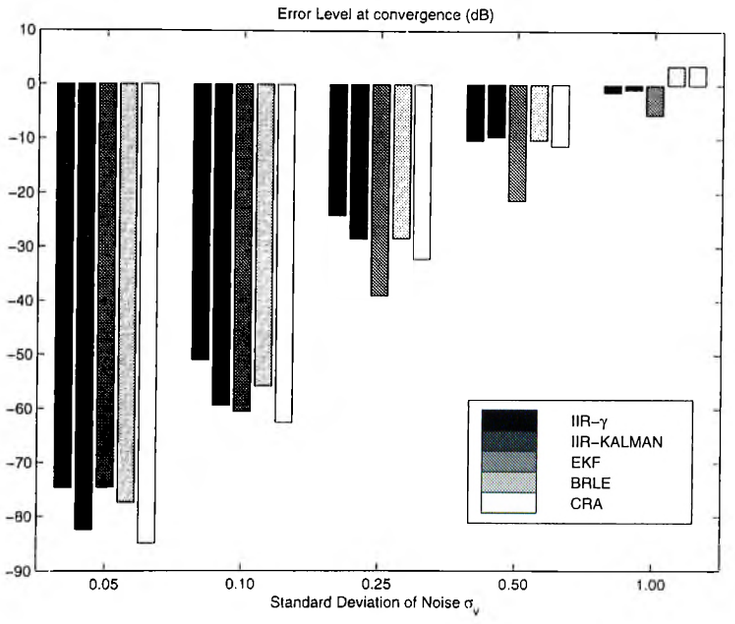


Figure 5.2: *Bar chart of squared parameter error norm in dB at convergence of the algorithms at different noise levels for Example 1*

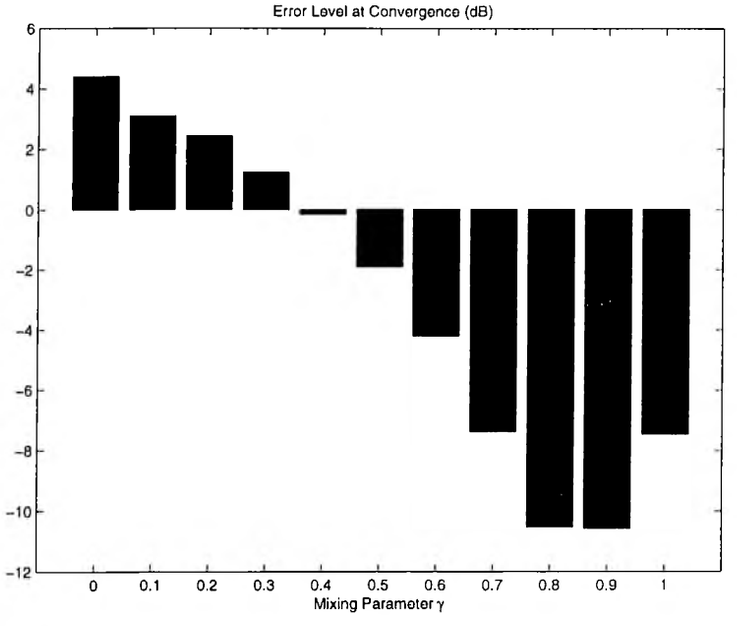


Figure 5.3: *Bar chart of squared parameter error norm in dB at convergence of the regressor algorithm when the mixing parameter, γ_n, is kept constant during the iterations for Example 1. Corresponding variance of the output noise is 0.25. Note that the marginal OE and EE formulations have larger error levels, than a composed regressor.*

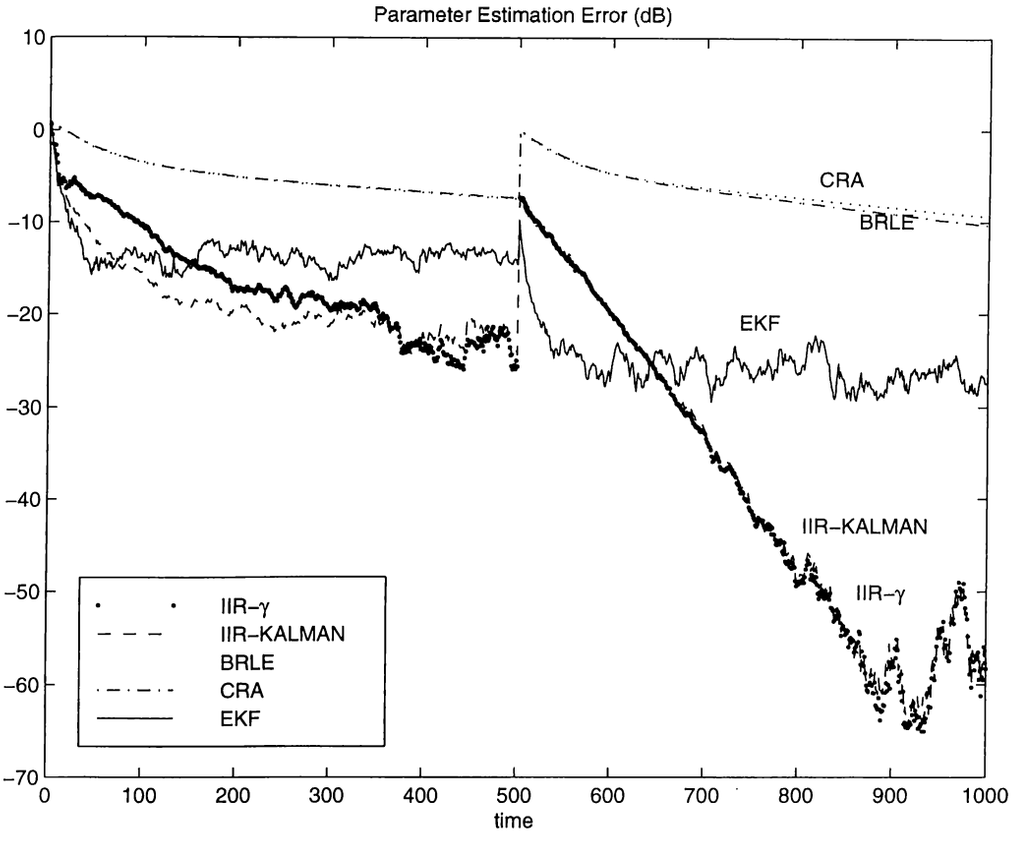
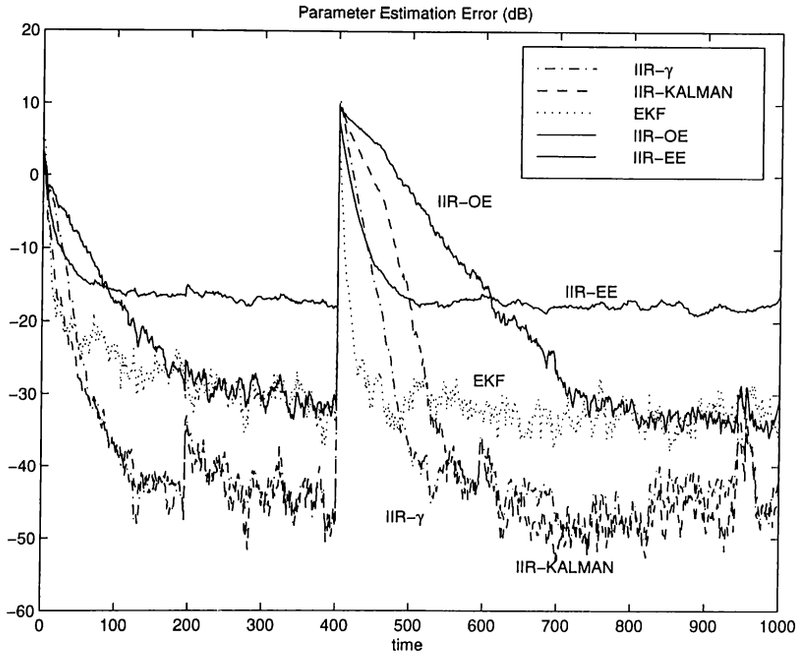
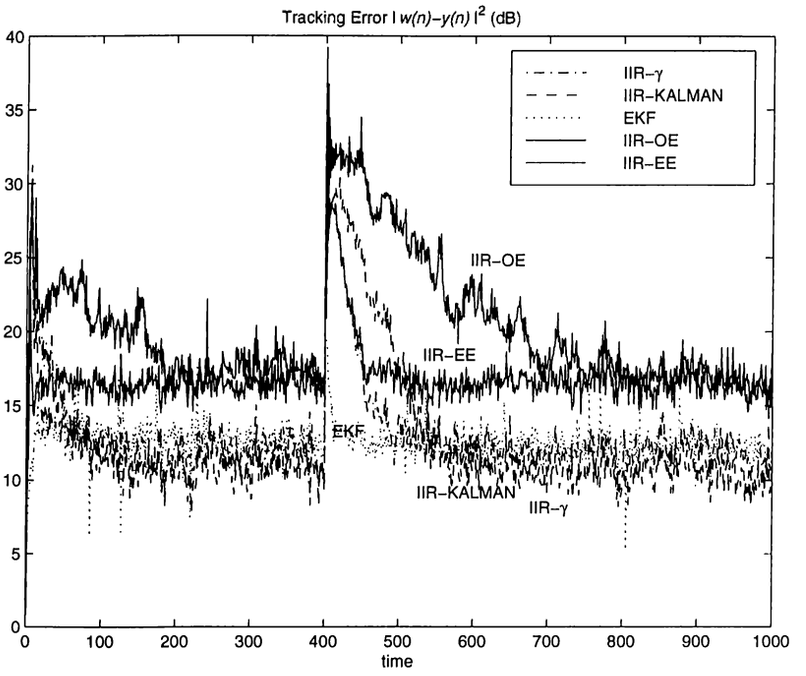


Figure 5.4: *Results of second example: squared norm of the parameter error as a function of time when the parameters of the original system are abruptly changed at time 500.*



(a)



(b)

Figure 5.5: *Results of third example: the original system parameters are abruptly changed at time 400, (a) squared norm of parameter error as a function of time, (b) output tracking error* $\|w(n) - y(n)\|^2$ *as a function of time.*

Chapter 6

Conclusions and Future Work

In order to be able to use fast converging recursive least squares adaptation algorithm in adaptive IIR filtering, a regressor based adaptive system structure is proposed. In the proposed approach, an adaptive regressor provides estimates to the actual system output based on the available noisy observations of the system output and the output of the adaptive system. Then, the estimated output and the system input are fed to a two–channel adaptive FIR filter whose parameters are updated by using a rotation–based multi–channel recursive least squares algorithm. Two different regressor algorithms, with number of multiplications in the $O(N)$ and $O(N^2)$ respectively, are proposed to provide reliable estimates to the system output. Robust ways of updating the parameters of the regressors are presented. Also, motivated from the use of Kalman regressor in the proposed adaptation structure, extended Kalman filter algorithm is applied to joint estimation of the system parameters and system output. By exploiting the special structure of the state space description of the adaptive system, it is shown that the corresponding extended Kalman algorithm does not require any matrix inversions and can be implemented by performing $O(N^2)$ multiplications.

The proposed regressor based adaptive IIR algorithms are compared with the extended Kalman approach as well as earlier proposed LMS type approaches: BRLE and CRA. Based on extensive set of simulations it is found that for time–invariant systems, the proposed algorithms not only converges faster than LMS type algorithms, but also, provide more reliable parameter estimates at low SNR. Also in the

simulation of the systems with abrupt changes in their parameters, it is observed that the proposed regressor based adaptation algorithms outperform the extended Kalman, BRLE and CRA algorithms establishing faster convergence to lower error levels.

As future work, the required modifications in the proposed algorithms in the presence of colored output noise can be investigated. Also, output tracking performance of the algorithms should be compared in the case of insufficient model order. More importantly, stability monitoring of the adapted system should be incorporated to the adaptation algorithm.

Appendix A

An Efficient Method of Estimation of var(u(n))

In the output error approach, the noise free output state is estimated by

$$\hat{w}_O(n) = \hat{\mathbf{a}}^T(n)\hat{\mathbf{w}}(n) + \hat{\mathbf{b}}^T(n)\mathbf{x}(n) \tag{A.1}$$

where the previous estimates of the output $w(n)$ is fed back to the adaptive system. However, in the equation error approach, the output is estimated by the noisy observation of it:

$$\hat{w}_E(n) = w(n) + v(n). \tag{A.2}$$

By denoting the parameter error vectors $\hat{\mathbf{a}}(n) - \mathbf{a}$ and $\hat{\mathbf{b}}(n) - \mathbf{b}$ as $\delta\mathbf{a}(n)$ and $\delta\mathbf{b}(n)$ respectively, Eqn. (A.1) can be written as:

$$\begin{aligned} \hat{w}_O(n) &= [\mathbf{a} + \delta\mathbf{a}(n)]^T [\mathbf{w}(n) + \delta\mathbf{w}(n)] + [\mathbf{b} + \delta\mathbf{b}(n)]^T \mathbf{x}(n) \\ &= w(n) + \delta\mathbf{a}^T(n)\hat{\mathbf{w}}(n) + \delta\mathbf{b}^T(n)\mathbf{x}(n) + \mathbf{a}^T\delta\mathbf{w}(n) \\ &= w(n) + u(n) \end{aligned} \tag{A.3}$$

where $\delta w(n)$ and $\delta\mathbf{w}(n)$ are defined as $\hat{w}(n) - w(n)$ and $\hat{\mathbf{w}}(n) - \mathbf{w}(n)$ respectively. Therefore, the estimate (A.3) which corresponds to the prior estimate of the state in Eqn. (3.10), and the estimate (A.2), which corresponds to the observation estimate in Eqn. (3.11), are optimally combined by the Kalman estimator so that the variance of the overall estimator is the lowest possible. Since both Eqns. (A.2) and (A.3)

are individual estimators of $w(n)$, under the assumption that $u(n)$ and $v(n)$ are uncorrelated white Gaussian random processes, the optimal estimator of $w(n)$ is given by:

$$\hat{w}_{opt}(n) = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_v^2}\hat{w}_E(n) + \frac{\sigma_v^2}{\sigma_u^2 + \sigma_v^2}\hat{w}_O(n) \tag{A.4}$$

where each estimator is weighted as inversely proportional to its variance. The variance of this optimal estimator is $\frac{\sigma_u^2\sigma_v^2}{\sigma_u^2+\sigma_v^2}$. Thus, the covariance matrix of the state estimate provided by the Kalman filter, is

$$E\{\delta\mathbf{w}(n)\delta\mathbf{w}^T(n)\} = \frac{\sigma_u^2\sigma_v^2}{\sigma_u^2 + \sigma_v^2}\mathbf{I}. \tag{A.5}$$

Now, by using the definition of $u(n)$ deduced from Eqn. (A.3):

$$u(n) = \delta\mathbf{a}^T(n)\hat{\mathbf{w}}(n) + \delta\mathbf{b}^T(n)\mathbf{x}(n) + \mathbf{a}^T\delta\mathbf{w}(n), \tag{A.6}$$

and assuming that $\delta\mathbf{a}(n)$, $\delta\mathbf{b}(n)$ and $\delta\mathbf{w}(n)$ are uncorrelated, we get:

$$\sigma_u^2 = \underbrace{\hat{\mathbf{w}}^T(n)\mathcal{R}_{\delta\mathbf{a}}(n)\hat{\mathbf{w}}(n) + \mathbf{x}^T(n)\mathcal{R}_{\delta\mathbf{b}}(n)\mathbf{x}(n)}_{\Upsilon} + \|\mathbf{a}\|^2\frac{\sigma_u^2\sigma_v^2}{\sigma_u^2 + \sigma_v^2} \tag{A.7}$$

Then, σ_u^2 can be found as

$$\sigma_u^2 = \frac{\Upsilon + \|\mathbf{a}\|^2\sigma_v^2 + \sqrt{[\Upsilon + \|\mathbf{a}\|^2\sigma_v^2]^2 + 4\Upsilon\sigma_v^2}}{2}. \tag{A.8}$$

Since $\mathbf{a}$ is not known, during the iterations σ_u^2 can be approximated by replacing $\mathbf{a}$ with $\hat{\mathbf{a}}(n)$, as well as the covariance matrices for the estimation errors are replaced with their approximations:

$$E\{\delta\mathbf{a}(n)\delta\mathbf{a}^T(n)\} \simeq \frac{1}{n-1}\sum_{i=1}^{n}\hat{\delta\mathbf{a}}(n)\hat{\delta\mathbf{a}}(n)^T = \hat{\mathcal{R}}_{\delta\mathbf{a}}(n) \tag{A.9}$$

$$E\{\delta\mathbf{b}(n)\delta\mathbf{b}^T(n)\} \simeq \frac{1}{n-1}\sum_{i=1}^{n}\hat{\delta\mathbf{b}}(n)\hat{\delta\mathbf{b}}(n)^T = \hat{\mathcal{R}}_{\delta\mathbf{b}}(n) \tag{A.10}$$

where $\hat{\delta\mathbf{a}}(n) = \hat{\mathbf{a}}(n) - \hat{\mathbf{a}}(n-1)$ and $\hat{\delta\mathbf{b}}(n) = \hat{\mathbf{b}}(n) - \hat{\mathbf{b}}(n-1)$ are the updates on the estimated parameters in two consecutive iterations. Under mild assumptions, it can be shown that Eqns. (A.9) and (A.10) provide reliable estimates at convergence. Then, the estimate of the σ_u^2 at time n results as:

$$\hat{\sigma}_u^2(n) = \frac{\Upsilon + \|\hat{\mathbf{a}}\|^2\sigma_v^2 + \sqrt{[\Upsilon + \|\hat{\mathbf{a}}\|^2\sigma_v^2]^2 + 4\Upsilon\sigma_v^2}}{2}. \tag{A.11}$$

where

$$\Upsilon = \hat{\mathbf{w}}^T(n)\hat{\mathcal{R}}_{\delta\mathbf{a}}(n)\hat{\mathbf{w}}(n) + \mathbf{x}^T(n)\hat{\mathcal{R}}_{\delta\mathbf{b}}(n)\mathbf{x}(n). \tag{A.12}$$

In order to have smoothly varying estimate during the iterations of the Kalman filter, a smoothing factor, β, can be added in the following update on the estimate:

$$\tilde{\sigma}_u^2(n) = \beta\tilde{\sigma}_u^2(n-1) + (1-\beta)\hat{\sigma}_u^2 \quad , \quad 0 \leq \beta \leq 1. \tag{A.13}$$

In the simulations, the smoothed estimate, $\tilde{\sigma}_u^2(n)$, is used.

Appendix B

Two–Channel Lattice Structure of an IIR Filter

An adaptive IIR filter can be structured as a two-channel FIR filter as in Fig. B.1. The order equations of p^{th} section of the lattice are:

$$\mathbf{f}_p(n) = \mathbf{f}_{p-1}(n) - \mathbf{K}^{\mathbf{b}^T}_p \mathbf{b}_{p-1}(n-1) \quad \text{(B.1)}$$

$$\mathbf{b}_p(n) = \mathbf{b}_{p-1}(n-1) - \mathbf{K}^{\mathbf{f}^T}_p \mathbf{f}_{p-1}(n). \quad \text{(B.2)}$$

and the output of the filter is defined as:

$$y(n) = \sum_{i=1}^{p_{max}} \underline{\mathbf{K}}^{\mathbf{e}^T}_p \mathbf{b}_{p-1}(n). \quad \text{(B.3)}$$

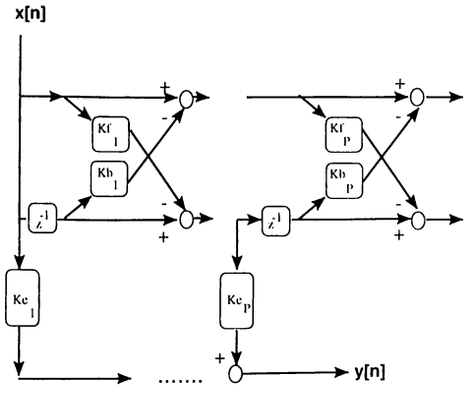


Figure B.1: *Multi–channel FIR lattice structure*

where $\mathbf{K}^{\mathbf{f}}{}_p$ and $\mathbf{K}^{\mathbf{b}}{}_p$ are the forward and backward predictor matrices of size 2×2 and $\underline{\mathbf{K}}^{\mathbf{e}}{}_p$ is a 2–dimensional vector of joint process estimator. The forward and backward prediction errors are $\mathbf{f}_p(n)$ and $\mathbf{b}_p(n)$ for p^{th} section at time n. The input vector is equal to:

$$\mathbf{x}(n) = \mathbf{f}_0(n) = \mathbf{b}_0(n) = \begin{bmatrix} x_1(n) & x_2(n) \end{bmatrix}^T \tag{B.4}$$

When the input of the first channel, $x_1(n)$ is chosen as the input of the original system, $x(n)$, and the second channel input, $x_2(n)$, is chosen as the one–sample delayed version of the reference output signal, $w(n-1)$, the shown structure is similar to an IIR system. Taking the first channel input as $x(n)$ and the second channel input as $\hat{w}(n)$ (the output of the regressor), the adaptive system output, $y(n)$, can be computed. In fact, the lattice form coefficients are adapted by a QR–MLSL algorithm. The direct form parameters can be found by a simple mapping of the reflection matrices. Notice that, the numerator coefficients of the IIR system, $\hat{\mathbf{b}}$, is the output response of the filter when $x_1(n) = \delta(n)$ and $x(n) = 0$. Similarly, the denumerator coefficients $\hat{\mathbf{a}}$ is the output response due to the inputs $x_1(n) = 0$ and $x_2(n) = \delta(n)$. The output can be computed by lattice recursions in Eqns. (B.1), (B.2), (B.3). This mapping is necessary to watch the estimated direct form coefficients and corresponding estimation error, $\mathbf{e}_0(n)$ defined in Eqn. (5.1).

References

[1] H. Fan and W. Jenkins "A New Adaptive IIR Filter," *IEEE Trans. on Circuits and Systems*, vol. CAS-33, pp. 939–947, Oct. 1986.

[2] J. Lin and R. Unbehauen "Bias-Remedy Least Mean Square Equation Error Algorithm for IIR Parameter Recursive Estimation," *IEEE Trans. on Signal Processing*, vol. 40, pp. 62–69, Jan. 1992.

[3] J. Kenney and C. Rohrs "The Composite Regressor Algorithm for IIR Adaptive Systems," *IEEE Trans. on Signal Processing*, vol. 41, pp. 617–628, Feb. 1993.

[4] C. Davilla "An Efficient Recursive Total Least Squares Algorithm for FIR Adaptive Filtering," *IEEE Trans. on Signal Processing*, vol. 42, pp. 268–280, Feb. 1994.

[5] S. Karaboyas and N. Kaluptsidis "Efficient Adaptive Algorithms for ARX Identification," *IEEE Trans. on Signal Processing*, vol. 39, pp. 571–582, Mar. 1991.

[6] K. Ho and Y. Chan "Bias Removal in Equation Error Adaptive IIR Filters," *IEEE Trans. on Signal Processing*, vol. 43, pp. 51–62, Jan. 1995.

[7] P. Regalia "An Unbiased Equation Error Identifier and Reduced-Order Approximations," *IEEE Trans. on Signal Processing*, vol. 42, pp. 1397–1412, Jun. 1994.

[8] P. Regalia "Stable and Efficient Lattice Algorithms for Adaptive IIR Filtering," *IEEE Trans. on Signal Processing*, vol. 40, pp. 375–388, Feb. 1992.

[9] B. Yang and J. Böhme "Rotation-Based RLS Algorithms: Unified Derivations, Numerical Properties, and Parallel Implementations," *IEEE Trans. on Signal Processing*, vol. 40, pp. 1151–1167, May 1992.

[10] P. Lewis "QR-Based Algorithms for Multichannel Adaptive Least Squares Lattice Filters," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 38, pp. 421–432, Mar. 1990.

[11] G. Goodwin and K. Sin. *Adaptive Filtering, Prediction and Control.* Prentice Hall, Englewood Cliffs, NJ, 1984.

[12] C. Chui and G Chen. *Kalman Filtering with Real-Time Applications.* Springer-Verlag, Berlin, 1991.

[13] C. Chui, G. Chen, and H. Chui "Modified Extended Kalman Filtering and a Real–Time Parallel Algorithm for System Parameter Identification," *IEEE Trans. on Automatic Control*, vol. 35, pp. 100–104, Jan. 1990.

[14] B. Mulgrew and C. Cowan. *Adaptive Filters and Equalisers.* Kluwer Academic Publishers, MA, 1988.

[15] S. Haykin, A. Sayed, J. Zeidler, P. Yee, and P. Wei "Adaptive Tracking of Linear Time–Variant Systems by Extended RLS algorithms," *IEEE Trans. on Signal Processing*, vol. 45, pp. 1118–1127, May 1997.

[16] T. Söderström and P. Stoica. *System Identification.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[17] L. Ljung and Söderström. *Theory and Practice of Recursive Identification.* The MIT Press, 1985.

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

[19] L. Larimore, J. Treichler, and C. Johnson "SHARF:An Algorithm for Adaptive IIR Digital Filters," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 428–440, Aug. 1980.