Recursive Identification Algorithms

Lennart Ljung

Department of Electrical Engineering Linköping University, SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se Email: ljung@isy.liu.se

October 3, 2001

AUTOMATIC CONTROL

LINKÖPING

Report no.: LiTH-ISY-R-2366

Technical reports from the Automatic Control group in Linköping are available by anonymous ftp at the address ftp.control.isy.liu.se. This report is contained in the file 2366.pdf.

Recursive Identification Algorithms

Lennart Ljung
Department of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

December 29, 2000

Abstract

Mechanisms for adapting models, filters, decisions, regulators and so on to changing properties of a system or a signal are of fundamental importance in many modern signal processing and control algorithms. This contribution describes a basic foundation for developing and analyzing such algorithms. Special attention is paid to the rationale behind the different algorithms, thus distinguishing between "optimal" algorithms and "ad hoc" algorithms. We also give an outline of the basic approaches to performance analysis of adaptive algorithms.

Keywords: Sequential estimation, adaptive algorithms, performance analysis, recursive estimation, Kalman filter, recursive least squares, least mean squares

1 Introduction

In many signal processing applications it is necessary to have a model available on-line. The reason might be that knowledge about the signal or system properties are accumulating as more observations become available and that a model based on current information is necessary to take an on-line decision. An even more common situation is that the properties in question are varying over time, so that a model that continuously adapts to a changing world is required. This contribution deals with basic ideas and algorithms to construct such recursive identification algorithms. Signal processing methods that employ recursively estimated, time-varying, models are typically called adaptive. Adaptive signal processing, adaptive prediction, adaptive filtering, adaptive echo-cancelling, etc., are thus all based on recursive parameter estimation techniques in one form or another.

There is a substantial literature on recursive estimation techniques. For general treatments, there are the books [7], [12], [10], [13], [4], [8], and [6] to name a few.

Tracking a system's properties is always a question of critically evaluating the observation obtained from the process in question: Do they contain information about changes in the process or are they just dominated by random disturbances? Thus even in a non-mathematical setting, adaptation and tracking is always characterized by a trade-off between tracking ability (dare to believe signs of process changes in the measurements!) and noise sensitivity (don't get confused by random fluctuations!). We shall see this fundamental trade-off show up in various formalized ways in the course of this contribution.

This focus of this paper is on theoretical issues, such as the basic rationale of the algorithms, and how to assess their tracking and convergence properties. For the important practical issues, like dealing with the tracking/noise trade-off and obtaining efficient implementations, the reader has to consult the references.

The organization of the paper is as follows: First, in Section 2, a typical, simple setup for tracking parameters in a time-varying linear regression is described. The optimal solution to this problem gives a certain, useful algorithm structure. In Section 3 ad-hoc choices of algorithms are discussed. These contain well-known cases such as LMS and RLS. Extensions beyond linear regressions are treated in Section 4. Analysis of the algorithms in the cases where the parameters are constant and time-varying is dealt with in Sections 5 and 6, respectively.

2 Optimal Algorithms for Tracking Drifting Parameters

To display the basic ideas of recursive estimation and algorithm structure, we first consider the following linear regression signal model:

$$y(t) = \varphi^{T}(t)\theta + e(t) \tag{1}$$

where $\{y(t)\}$ and $\{\varphi(t)\}$ are observed signals. The vector θ contains the unknown parameters, which are to be estimated by the tracker.

A common application of (1) in control and signal processing is when the regression vector $\varphi(t)$ consists of lagged outputs and inputs

$$\varphi^{T}(t) = (-y(t-1), \dots, -y(t-n), u(t-1), \dots, u(t-m)). \tag{2}$$

In this case (1) and (2) correspond to a linear difference relationship between the input and the output. In case there is no "input" signal $\{u(t)\}$ we have the well-known AR model for the signal $\{y(t)\}$. In any case, note that the nature of φ is not essential for the discussion to follow — The only thing that matters, is that $\varphi(t)$ is a known vector at time t.

We now assume that there is a true - and time varying - value $\theta_0(t)$ for the parameters and that these develop over time as a random walk. This means

that the "true" description of the signals $\{y(t)\}\$ and $\{\varphi(t)\}\$ becomes

$$\theta_0(t) = \theta_0(t-1) + w(t) \tag{3a}$$

$$y(t) = \theta_0^T(t)\varphi(t) + e(t). \tag{3b}$$

We here assume $\{e(t)\}$ to be white Gaussian noise with variance $R_2(t)$, while $\{w(t)\}$ is white Gaussian noise with covariance matrix $R_1(t)$. It is then well known, see, e.g., Section 2.3 in [7] that the estimate $\hat{\theta}(t)$ that minimizes the conditional expectation, given past observations

$$\Pi(t) = E(\hat{\theta}(t) - \theta_0(t))(\hat{\theta}(t) - \theta_0(t))^T$$
(4)

(even in a matrix sense) is given by the Kalman filter. The general algorithm is as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t)\varepsilon(t) \tag{5a}$$

$$\varepsilon(t) = y(t) - \varphi^{T}(t)\hat{\theta}(t-1) \tag{5b}$$

where the gain vector L(t) is given by

$$L(t) = \frac{P(t-1)\varphi(t)}{\hat{R}_2(t) + \varphi^T(t)P(t-1)\varphi(t)}$$
(5c)

and the matrix P(t) is updated according to

$$P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^{T}(t)P(t-1)}{\hat{R}_{2}(t) + \varphi^{T}(t)P(t-1)\varphi(t)} + \hat{R}_{1}(t)$$

$$P(0) = P_{0},$$
(6)

We have here used the notations $\hat{R}_1(t)$ and $\hat{R}_2(t)$ to indicate that the values used in the algorithm may very well differ from the true values $R_1(t)$ and $R_2(t)$. In the case $\hat{R}_2(t) \equiv R(t)$ and $\hat{R}_2(t) \equiv R_2(t)$, however, $\hat{\theta}(t)$ is the conditional expectation of $\theta_0(t)$, given the observations $\{u(k), y(k)\}, k \leq t$, and P(t) is the conditional covariance matrix of the parameter estimation error.

Note also that if $R_1(t)$ is known then (5)-(6) is the optimal algorithm also for abrupt changes in θ_0 . (Take $R_1(t) = 0$ except when a jump occurs, say for $t \in T_1$ take then $R_1(t) = R_1$.) However, this requires the time instants for the jumps to be known, not too realistic an assumption.

In the algorithm (6) it follows that, after a transient, the size of P(t) will be like the square root of \hat{R}_1 . For slowly changing systems, P will thus be small. To explicitly show this it is useful to scale P, so as to rewrite (5) as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \mu P_t L(\varphi(t))(y(t) - \varphi^T(t)\hat{\theta}(t-1))$$
(7)

We have here allowed a possible non-linear transformation L (such as normalization) of $\varphi(t)$.

3 Some ad hoc Algorithms for Tracking Drifting Parameters: RLS and LMS

The basic formulation (3) with the optimal algorithm (5)-(6) is quite powerful. It can deal with both slowly drifting parameters and with sudden changes, by assigning proper values to the covariance matrix $\hat{R}_1(t)$ and the variance $\hat{R}_2(t)$. The main shortcoming is then that these values will rarely be known to the user. One approach to deal with this problem is to choose some ad hoc values for $\hat{R}_1(t)$. We will discuss two such ad hoc choices below.

3.1 The RLS Algorithm

It is a useful, and well known feature, of linear least squares regression models, the the least squares estimate can be computed exactly in a recursive fashion. This leads to the RLS (Recursive Least Squares) algorithm. This can also be exploited when the underlying system is thought to be time-varying.

A popular approach to deal with time-varying linear regressions is to minimize a weighted criterion

$$V_t(\theta) = \sum_{k=1}^t \beta(t, k) (y(k) - \theta^T \varphi(k))^2$$
 (8)

where

$$\beta(t,k) = \prod_{j=k+1}^{t} \lambda(j)$$
 (9)

and where $0 \le \lambda(j) \le 1$ is called the *forgetting factor*. Note that older observations in (8) will have a smaller influence than new ones due to the forgetting. If $\lambda(t) = \lambda \, \forall t$, we have

$$\beta(t,k) = \lambda^{t-k} \tag{10}$$

which gives exponential forgetting. In this case, note that observations that are older than $1/(1-\lambda)$ have a weight that is less than about a third of the most recent measurement. (To see this, do the following calculation:

$$\lambda^{1/(1-\lambda)} = e^{\frac{\log(1-(1-\lambda))}{1-\lambda}} \approx e^{-1})$$

In the exponential forgetting case the criterion (8) can thus said to have a *memory horizon* of about

$$T_{\lambda} = \frac{1}{1-\lambda}$$

Now the essential point is that also the weighted criterion (8) can be exactly minimized w.r.t. θ by a recursive algorithm, see, e.g., [7]. This is recursive least

squares (RSL) algorithm, which is given by (5) with L(t) chosen as

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^{T}(t)P(t-1)\varphi(t)}$$

$$P(t) = \frac{1}{\lambda(t)} \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^{T}(t)P(t-1)}{\lambda(t) + \varphi^{T}(t)P(t-1)\varphi(t)} \right]$$
(11)

We note that this is a special case of (5)-(6), corresponding to the choices

$$\hat{R}_{1}(t) = \left(\frac{1}{\lambda(t)} - 1\right) \cdot \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^{T}(t)P(t-1)}{\lambda(t) + \varphi^{T}(t)P(t-1)\varphi(t)}\right]
\approx \left(\frac{1}{\lambda(t)} - 1\right)P(t-1)$$
(12)
$$\hat{R}_{2}(t) = \lambda(t)$$

This means that RLS would give optimal tracking if the true variations of the parameters could be described by a covariance matrix proportional to P.

For future use we also note that

$$P(t) = \left[\sum_{k=1}^{t} \beta(t, k) \varphi(k) \varphi^{T}(k) \right]^{-1}$$
(13)

The connection to the archetypical algorithm (7) is given by

$$\mu(t) = \left[\sum_{k=1}^{t} \beta(t, k)\right]^{-1} \tag{14}$$

3.2 The Least Mean Square (LMS) Algorithm

While the RLS algorithm is based on minimization of a standard linear least squares, Widrow's least mean squares (LMS) algorithm is derived from the idea to minimize the criterion

$$V(\theta) = E||y(t) - \theta^T \varphi(t)||^2$$

where E means mathematical expectation. A simple gradient search for the minimum gives

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma E \varphi(t) (y(t) - \hat{\theta}^T(t-1)\varphi(t))$$

Replacing the expected value by the observation itself gives the LMS algorithm (see, e.g., [12]) which is a very common tool for adaptation. It is given by (7) with

$$P_t = I, \quad L(t) = \mu \varphi(t) \tag{15}$$

The LMS algorithm can also be used in a normalized variant (NLMS):

$$L(t) = \frac{\mu \varphi(t)}{1 + \mu |\varphi(t)|^2}.$$
 (16)

Again, we may verify that (5a) and (16) is a special case of the basic algorithm (5) corresponding to

$$\hat{R}_1(t) = \mu^2 \frac{\varphi(t)\varphi^T(t)}{1 + \mu|\varphi(t)|^2}$$
(17)

$$\hat{R}_2(t) = 1 \tag{18}$$

$$P(0) = \mu \cdot I. \tag{19}$$

It would thus be the optimal tracking algorithm if the true parameter variations were proportional to the covariance matrix of the regression vector φ .

4 Algorithms for General Non-linear Regressions

Most models for dynamical systems can be cast into the form

$$y(t) = \hat{y}(t|\theta) + e(t) \tag{20}$$

where $\hat{y}(t|\theta)$ is a general function of input-output data and of the parameter vector θ . The notation \hat{y} emphasizes the interpretation of this quantity as a predictor. The book [5] contains many examples for how different model descriptions fit into the format (20). We note in passing that also multi-layered perceptions (neural networks) are special cases of (20) (θ then corresponds to the weights in the interconnections). See also the paper on *Prediction Error Estimation Methods* in this special issue.

A Derivation from the Off-line Prediction Error Criterion

Based on the general model (20) we can form a weighted prediction error criterion

$$V_t(\theta) = \sum_{k=1}^t \beta(t, k) \ell(\varepsilon(k, \theta), k)$$

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta)$$
(21)

Here $\ell(\cdot)$ is a scalar valued function that - in some sense - measures the "size" of the prediction error ε .

In the off-line case (21) is typically minimized by iterative search, e.g., of the Gauss-Newton type. A basic approach to adaptation is to perform one iteration for the minimization of (21) at the same time as one more observation (t) increased one unit) is obtained. This approach is detailed in [5], Chapter 11.

$$\beta(t,k) = \prod_{j=k+1}^{t} \lambda(j)$$
 (22)

the resulting algorithm is of the form

$$\hat{\theta}(t) = \hat{\theta}(t-1) + R^{-1}(t)\psi(t)\ell'_{\varepsilon}(\varepsilon(t), t)
R(t) = \lambda(t)R(t-1) + \psi(t)\ell''_{\varepsilon\varepsilon}(\varepsilon(t), t)\psi^{T}(t)$$
(23)

(See (11.52) of [5]). Here $\psi(t)$ is an approximation of the gradient

$$\psi(t, \hat{\theta}(t-1)) = \left. \frac{d}{d\theta} \hat{y}(t|\theta) \right|_{\theta = \hat{\theta}(t-1)}$$
(24)

and $\varepsilon(t)$ in an approximation of

$$\varepsilon(t, \hat{\theta}(t-1)) = y(t) - \hat{y}(t|\hat{\theta}(t-1)) \tag{25}$$

Moreover ℓ'_{ε} and $\ell''_{\varepsilon\varepsilon}$ are the derivatives of ℓ with respect to ε . In the special case, where $\hat{y}(t|\theta)$ is a linear regression

$$\hat{y}(t|\theta) = \varphi^T(t)\theta$$

and the norm ℓ is quadratic

$$\ell(\varepsilon) = \frac{1}{2}\varepsilon^2 \tag{26}$$

we recognize in (23) the RLS algorithm

A Derivation as an Approximation of the Linear Regression Case

To put the general model (20) more in line with the linear regression case, treated in Sections 2-4, we can make an approximate derivation of a general algorithm as follows.

Consider the general structure (20) together with a random walk model for the variation of the "true parameter vector"

$$\theta_0(t) = \theta_0(t-1) + w(t) y(t) = \hat{y}(t|\theta_0(t)) + e(t).$$
 (27)

Suppose that we have an approximation $\theta_*(t)$ of $\theta_0(t)$ available. We can then write, using the mean value theorem

$$\hat{y}(t|\theta_0(t)) = \hat{y}(t|\theta_*(t)) + (\theta_0(t) - \theta_*(t))^T \psi(t, \xi(t))$$
(28)

where $\xi(t)$ is a value "between" $\theta_*(t)$ and $\theta_0(t)$. Here $\psi(t,\theta)$ is the gradient of $\hat{y}(t|\theta)$, as defined in (24). Normally, $\psi(t,\xi(t))$ would not be known, but we may assume that an approximation

$$\psi(t) \approx \psi(t, \xi(t)) \tag{29}$$

is available. Introduce the known variable

$$z(t) = y(t) - \hat{y}(t|\theta_*(t)) + \theta_*^T(t)\psi(t).$$
(30)

Subject to the approximation (29) we can then rewrite (27) as

$$\theta_0(t) = \theta_0(t-1) + w(t) z(t) = \theta_0^T(t)\psi(t) + e(t)$$
 (31)

and we are back to the situation of Section 2. A natural choice $\theta_*(t)$ of a good approximation of $\theta_0(t)$ would be the previous estimate $\theta_*(t) = \hat{\theta}(t-1)$. Applying the general algorithm (5), the error in (5b) becomes

$$\begin{split} \varepsilon(t) &= z(t) - \hat{\theta}^T(t-1)\psi(t) \\ &= y(t) - \hat{y}(t|\hat{\theta}(t-1) + \hat{\theta}^T(t-1) - \hat{\theta}^T(t-1) \\ &= y(t) - \hat{y}(t|\hat{\theta}(t-1)) \end{split}$$

This means that the general algorithm (5) will be driven by the prediction error, and a recursive prediction error algorithm of the type (23) is obtained. (Note that in case (26) $\ell'_{\varepsilon}(\varepsilon(t),t) = \varepsilon(t)$) As $\hat{\theta}(t-1)$ comes closer to $\theta_0(t)$, the approximation involved in going from (27) to (31) will become arbitrarily good. This shows that an asymptotic theory of tracking parameters in arbitrary model structures can be developed from the linear regression case.

5 Asymptotic Properties in the Decreasing Gain Case

The actual use of the adaptive algorithms is to track time-varying properties of a system or a signal. Still, a natural first question is to ask how well the algorithms are capable to handle a time invariant system. This corresponds to the special case $R_1(t) = \hat{R}_1(t) = 0$ in (6), (3b) or $\lambda(j) \equiv 1$ in (9) or (22).

A substantial part of [7] is devoted to such analysis, and we shall here only quote the bottom lines:

1. As t tends to infinity, and as the gain tends to zero, a recursive prediction error algorithm (23) will converge to a local minimum of the expected loss function

$$\hat{\theta}(t) \to \arg\min \bar{V}(\theta)$$
 w.p 1 as $t \to \infty$, $\bar{V}(\theta) = E\ell(\varepsilon(t,\theta), t)$ (32)

i.e. in terms of the criterion $E\ell(\varepsilon)$, "the best possible approximation is obtained".

2. If, in addition, the Gauss-Newton search direction (23) is used, and asymptotically equal weighting is used ($\lambda(j) \equiv 1$) then the asymptotic accuracy

$$\bar{P} = \lim_{t \to \infty} t E(\hat{\theta}(t) - \theta_0) (\hat{\theta}(t) - \theta_0)^T$$

will be the same as for the corresponding off-line estimation method.

These asymptotic properties are thus the best one could ask for. It remains though to study how the algorithms actually can cope with time varying systems. This is the question we turn to next.

6 Tracking Ability of the Algorithms

In the analysis of the tracking ability we will only study algorithms for linear regressions. We first develop an exact expression for the parameter error.

Let us consider the description (3) for the behavior of the true system together with the generic parameter estimation algorithm (5).

$$\theta_0(t+1) = \theta_0(t) + \gamma w(t) \tag{33a}$$

$$y(t) = \varphi^{T}(t)\theta_0(t) + e(t)$$
(33b)

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t)\varepsilon(t) \tag{33c}$$

$$\varepsilon(t) = y(t) - \varphi^{T}(t)\hat{\theta}(t-1)$$
(33d)

Introduce the parameter error

$$\tilde{\theta}(t) = \hat{\theta}(t) - \theta_0(t+1). \tag{34}$$

Remark. The variable γ is used to easily treat scaling of the parameter changes. The time indexing here may seem somewhat peculiar, but it will simplify the expressions to follow. From an expression for the covariance of $\tilde{\theta}(t)$ we can exactly derive, e.g. the covariance of $\hat{\theta}(t) - \theta_0(t)$.

Then

$$\tilde{\theta}(t) = (I - L(t)\varphi^{T}(t))\tilde{\theta}(t-1) + L(t)e(t) - \gamma w(t).$$
(35)

The parameter error thus obeys a linear, time-varying difference equation. Notice that the L(t) is always of the form

$$L(t) = \bar{P}(t)\varphi(t) \tag{36}$$

for some matrix $\bar{P}(t)$. Solving (35) gives

$$\tilde{\theta}(t) = \Phi(t, 0)\tilde{\theta}(0) + \sum_{k=1}^{t} \Phi(t, k) [\bar{P}(k)\varphi(k)e(k) - \gamma w(k)]$$
(37)

where

$$\Phi(t,k) = \prod_{j=k}^{t} (I - \bar{P}(j)\varphi(j)\varphi^{T}(j)). \tag{38}$$

Expressions (35) and (37) form the basis for all analysis of the performance of the algorithm, and they hold for any sequences $\{\varphi(t)\}$, $\{e(t)\}$ and $\{w(t)\}$. The difficulty in the analysis lies in the complicated expression for $\Phi(t,k)$. Its properties depend entirely on the sequence $\{\varphi(t)\}$, but they are inherited in a fairly complicated way. We shall be interested in the properties of $\tilde{\theta}(t)$ as the gain L(t) becomes small. We therefore write

$$L(t) = \mu P_t \varphi(t) \tag{39}$$

where μ is a positive scaling parameter (see (7)), and obtain

$$\tilde{\theta}(t) = (I - \mu P_t \varphi(t) \varphi^T(t)) \tilde{\theta}(t - 1) + \mu P_t \varphi(t) e(t) - \gamma w(t). \tag{40}$$

The quantity that we are interested in is the size of the error $\tilde{\theta}(t)$ as measured by the covariance matrix

$$\Pi(t) = E\tilde{\theta}(t)\tilde{\theta}^{T}(t). \tag{41}$$

Here expectation "E" is over $\{e(t)\}$, $\{w(t)\}$ as well as over any random components of $\{\varphi(t)\}$. The exact expression for $\Pi(t)$ follows a somewhat complex equation. It can now be shown that $\Pi(t)$ is well approximated by $\hat{\Pi}(t)$, defined by

$$\hat{\Pi}(t) = (I - \mu \bar{P}_t Q(t)) \hat{\Pi}(t - 1) (I - \mu \bar{P}_t Q(t))^T + \mu^2 \bar{P}_t Q(t) \bar{P}_t \cdot R_2(t) + \gamma^2 R_1(t)$$

$$\hat{\Pi}(t_0) = \Pi(t_0).$$
(42)

Here

$$\bar{P}_t = EP_t, \quad Q(t) = E\varphi(t)\varphi^T(t)$$
 (43)

$$R_1(t) = Ew(t)w^T(t), \quad R_2(t) = Ee^2(t).$$
 (44)

There are several possibilities to establish that Π and $\hat{\Pi}$ are close. See e.g., [10] and [3]. In essence, (42) is obtained from (40) by squaring it and applying expectation neglecting certain dependencies between random variables.

Let us briefly discuss the implications of the expression (42). There is a substantial amount of papers that discuss such implications, e.g., [11], [1], [9], and [2]. We shall only comment on the case of RLS with forgetting factor $\lambda = 1 - \mu$. With

$$\bar{P}_t = \bar{P} = Q^{-1}, \quad Q(t) = Q$$

this gives the expression

$$\hat{\Pi}(t) = \hat{\Pi}(t-1) - 2\mu\hat{\Pi}(t-1) + \mu^2\hat{\Pi}(t-1) + \mu^2Q^{-1} \cdot R_2 + \gamma^2R_1 \tag{45}$$

As $t \to \infty$ we find that

$$\hat{\Pi}(t) \rightarrow \hat{\Pi}$$

where

$$\hat{\Pi} = \frac{1}{2} (\mu Q^{-1} R_2 + \frac{\gamma^2}{\mu} R_1) \tag{46}$$

(neglecting the term $\mu^2\hat{\Pi}$)

This expression shows clearly the trade-off in the choice of step size (adaptation gain) μ (or forgetting factor $\lambda = 1 - \mu$). A small μ gives a small influence from the noise $\{e(t)\}$ in the term $\mu Q^{-1}R_2$ and a large tracking error from the term $\gamma^2/\mu R_1$ and vice versa for a large μ .

Other specific algorithms, such as LMS show similar trade-offs. We may note, in the general case, as t tends to infinity, that $\hat{\Pi}(t)$ will converge to the solution $\hat{\Pi}$ of

$$\bar{P}Q\hat{\Pi} + \hat{\Pi}Q\bar{P} = \mu\bar{P}Q\bar{P}R_2^0 + \frac{\gamma^2}{\mu}R_1^0 \tag{47}$$

(where we assume \bar{P} , Q, R_1 and R_2 to be time-invariant). If $P_t = \mu \bar{P}_t$ obeys (6) and $\hat{R}_1(t) = \mu^2 \hat{R}_1$ is small and constant (or averages around such a value) similar argument will show that \bar{P}_t will converge to \bar{P} which, for small μ will approximately, be given by

$$\bar{P}Q\bar{P} = \hat{R}_1 \tag{48}$$

(assuming $\hat{R}_2 = 1$)

This paper does not cover the practical issues in recursive identification. One of the most important choices for successful applications of recursive techniques is to strike the right balance between tracking ability and noise sensitivity (in terms of μ , λ , \hat{R}_1 , \hat{R}_2 , depending on which type of algorithm is chosen). We refer to the references for such discussions, but note that expressions like (46) and (47) are essential for such trade-offs.

7 Conclusions

Linear regression models form a good basis for discussing how to derive recursive, adaptive algorithms. They are also the most commonly used models. With perfect knowledge of the character of how the true system varies (in our case the matrices $R_1(t)$ and $R_2(t)$), possibly with jumps, the Kalman filter provides the natural framework for the estimation. An important reason for the extensive literature on adaptive algorithms is that such knowledge typically is not

available. Hence, several ad hoc choices of algorithms have been developed. We have pointed to this framework, and also shown how the typical ad hoc algorithms can be seen as corresponding to certain assumptions about the system variations.

A second essential aspect of recursive identification is to assess how the crucial trade-off between tracking ability and noise sensitivity is affected by the user's design variables. We have pointed to the basic character of this trade-off and how analytical expressions for this can be derived for quite general algorithms.

References

- [1] D.C. Farden. Tracking properties of adaptive signal processing algorithms. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-29(3):439–446, 1981.
- [2] W. Gardner. Nonstationary learning characteristics of the LMS algorithms: A general study, analysis and critique. *IEEE Trans. of Circuits and Systems*, CAS-34(10):1199–1207, 1987.
- [3] L. Guo and L. Ljung. Performance analysis of general tracking algorithms. *IEEE Trans. Automatic Control*, AC-40:1388–1402, August 1995.
- [4] N. Kaloupsides and S. Theodoridis, editors. *Adaptive System Identification and Signal Processing Algorithms*, Hemel Hampstead, 1993. Prentice Hall International.
- [5] L. Ljung. System Identification Theory for the User. Prentice-Hall, Upper Saddle River, N.J., 2nd edition, 1999.
- [6] L. Ljung, G. Pflug, and H. Walk. Stochastic Approximation and Optimization of Random Systems. Birkhäuser, Berlin, 1992.
- [7] L. Ljung and T. Söderström. Theory and Practice of Recursive Identification. MIT press, Cambridge, Mass., 1983.
- [8] O. Macchi. Adaptive Processing: The Least Mean Squares Approach with Applications in Transmission. Wiley, Chichester, 1995.
- [9] O. Macchi and E. Eweda. Second-order convergence analysis of stochastic adaptive linear filtering. *IEEE Trans. Automatic Control*, AC-28(1):76–85, 1983.
- [10] V. Solo and X. Kong. Adaptive Signal Processing Algorithms. Prentice-Hall, Englewood Cliffs, 1995.
- [11] B. Widrow, J.M. McCool, M.G. Larimore, and C.R. Johnson Jr. Stationary and nonstationary learning characteristics of the lms adaptive filter. *Proceedings of the IEEE*, 64(8):1151–1162, 1976.

- [12] B. Widrow and S. Stearns. Adaptive Signal Processing. Prentice-Hall, Englewood-Cliffs, 1985.
- [13] P.C. Young. Recursive Estimation and Time-Series Analysis. Springer Verlag, Berlin, 1984.