A Self-Tuning Filter for Fixed Lag Smoothing

Hagander, Per; Wittenmark, Björn

1976

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Hagander, P., & Wittenmark, B. (1976). *A Self-Tuning Filter for Fixed Lag Smoothing*. (Technical Reports TFRT-7091). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
2

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

TFRT - 7091

A SELF-TUNING FILTER FOR FIXED LAG SMOOTHING

P. HAGANDER
B. WITTENMARK

Report 7607(C) February 1976
Department of Automatic Control
Lund Institute of Technology

A SELF-TUNING FILTER FOR FIXED LAG SMOOTHING

P Hagander, B Wittenmark

Department of Automatic Control
Lund Institute of Technology
Lund, Sweden

ABSTRACT

The paper deals with the problem of estimating a discrete time stochastic signal which is corrupted by additive white measurement noise.

The first part of the paper shows how the stationary solution to the fixed lag smoothing problem can be obtained. The first step is to construct an innovation model for the process. It is then shown how the fixed lag smoother can be determined from the polynomials in the transferfunction of the innovation model. In many applications the signal model and the characteristics of the noise process are unknown. The paper shows that it is possible to derive an algorithm which on-line finds the optimal fixed lag smoother, a self-tuning smoother.

The self-tuning smoother consists of two parts: An on-line estimation of the parameters in the one-step-ahead predictor of the measured signal, and a computation of the smoother coefficients by simple manipulation of the predictor parameters. The smoother has good transient as well as good asymptotic properties.

TABLE OF CONTENTS

1. INTRODUCTION

The received signals in many applications are corrupted by noise. The objective of a receiver is therefore to eliminate the influence of the measurement noise as well as possible. The received signal is often filtered in order to get the best estimate of the desired signal at a certain time, t, based on the measurements up to and including the same time t. In some situations it is possible to get a substantial improvement by making a smoothing of the received signal, i e the signal at some time back, t - k, is estimated based on measurements up to the present time t. This is called fixed lag smoothing. Such an extra time lag is, especially in one way communication, of almost no disadvantage.

The filter and the smoothing cases of discrete time signals are illustrated in Figure 1.1.

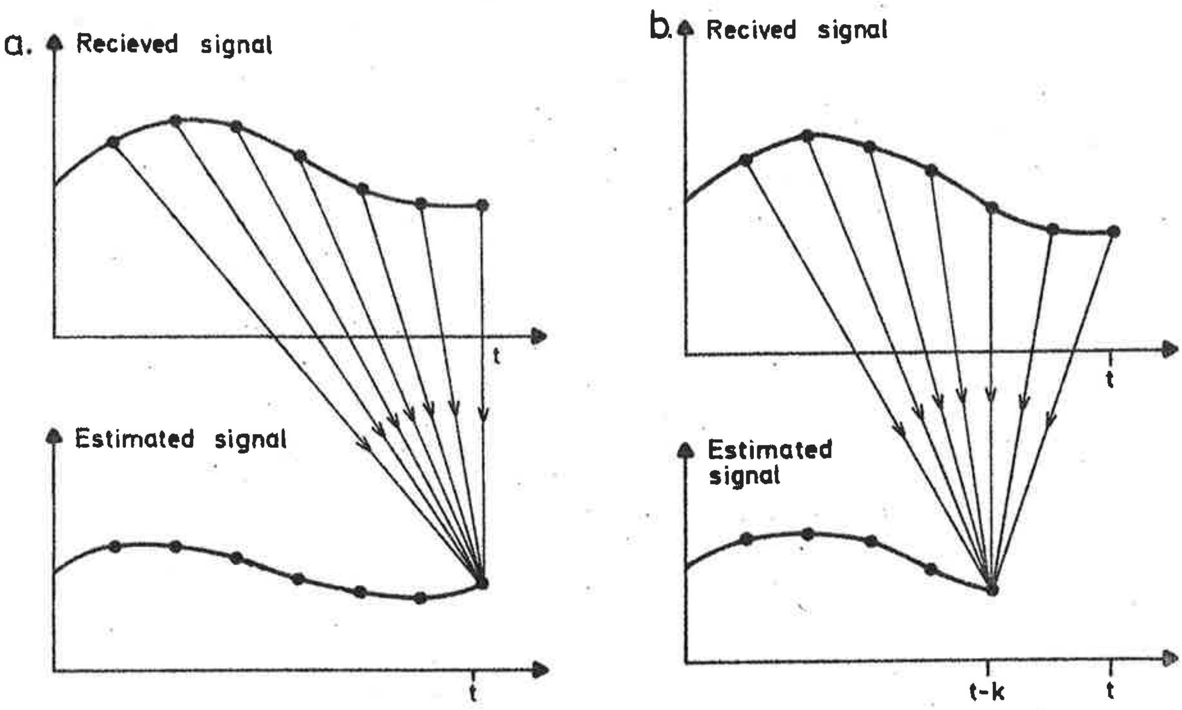


Fig 1.1 The filter (a) and the fixed lag smoothing (b) cases for discrete time signals.

The fixed lag smoothing is quite complicated to implement for continuous time signals, but straight forward in the discrete time case. A substantial number of papers are published during the last years giving different mechanizations and derivations [4],[5],[6],[9],[10],[11],[12]. Optimal estimation requires that the statistics are known both for the signal and the noise. The problem can then be solved using a Riccati-equation. However, the parameters of the process are in most cases unknown. In this paper we will discuss the stationary discrete time fixed lag smoother for a white noise corrupted <u>unknown</u> signal.

Let the desired signal be described by the model

$$z(t) = \frac{C(q^{-1})}{A(q^{-1})} \quad v(t) \tag{1.1}$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_nq^{-n}$$

$$C(q^{-1}) = c_1q^{-1} + c_2q^{-2} + c_3q^{-3} + \ldots + c_nq^{-n} \qquad c_1 = 1$$

are polynomials in the backward shift operator q^{-1}. The coefficients in the A and C polynomials are assumed to be unknown except c_1 which is known to be equal to 1.
The measurement of z(t), y(t), is corrupted by noise i e

$$y(t) = z(t) + e(t) \tag{1.2}$$

The noise processes, $v(\cdot)$ and $e(\cdot)$, are independent white noise processes with zero mean value and the standard deviations σ_v and σ_e respectively. It is assumed that σ_v and σ_e are unknown.

The problem can now be formulated as to find the best estimate of z(t-k), in the sense of mean square error, given data y(t), y(t-1),... . This estimate is denoted by $\hat{z}(t-k|t)$. I e we want to minimize the lossfunction

$$E\,[(z(t-k)-\hat{z}(t-k|t))^2|y(t),y(t-1),\ldots] \tag{1.3}$$

with respect to $\hat{z}(t-k|t)$.

For the derivation of the smoother it is also convenient to introduce the reciprocal polynomials $A^*(q)$ and $C^*(q)$ respectively

$$A^*(q) = q^n A(q^{-1}) = q^n + a_1 q^{n-1} + \ldots + a_n$$

$$C^*(q) = q^n C(q^{-1}) = q^{n-1} + c_2 q^{n-2} + \ldots + c_n$$

The self-tuning smoother is derived for the case where $e(t)$ is white noise. If $e(t)$ is coloured noise then it is necessary to have further information about the signal or the measurement noise. For instance if the covariance function of $e(t)$ is known then $y(t)$ could be filtered using the inverse of the noise model. The smoothing problem is then transformed to the problem discussed in this paper. If the covariance of the signal is known the problem can also be solved in an equivalent way.

The paper is organized as follows. In Section 2 the optimal fixed lag smoother is derived when the process is known. The first step is to construct the innovation model for the process. It is then shown how the stationary fixed lag smoother can be determined using polynomial operations on the polynomials in the innovation model.

The simple structure of the optimal smoother then indicates how to derive a self-tuning filter, when the parameters in the process are unknown. This is done in Section 3. The filter can be separated into two parts. First the parameters in the innovation model are estimated using a real time estimation method. Based on the estimated parameters the smoother is determined. These two steps are repeated at each step of time, when a new measurement is obtained. In Section 3 it is shown that if sufficiently many parameters are estimated then the self-tuning filter will converge to the optimal fixed lag smoother derived in Section 2. The algorithm is analyzed and aspects on the implementation are also discussed in Section 3.

Section 4 contains some simulated examples which illustrate the properties of the self-tuning algorithm. Section 5 summarizes the properties and discusses the usefulness of the self-tuning filter for fixed lag smoothing. References are given in Section 6.

2. THE OPTIMAL FIXED LAG SMOOTHER

Many different formulas and derivations for the smoothing problem have appeared during the last few years. Using the Wiener-formulation [15] the problem is conceptually very simple, but unfortunately nontrivial to mechanize.

When the recursive techniques for filters were introduced, e g Kalman filters [7] there appeared a number of papers on "state space smoothing", e g Bryson and Frazier [4], Rauch, Tung and Striebel [12], Mayne [9]. The theory is surveyed e g in Van Trees [14], Meditch [10] and Kailath and Frost [6]. Especially the fixed lag smoothing problem caused considerable difficulty.

It should be noted that there is most often a considerable difference in complexity between the problem posed by e g Wiener: "Find a signal in noise", as compared to the problem: "Find the smoother estimate of the state in a state space system". It is in fact for the former case possible to derive and mechanize the stationary fixed-lag smoothing estimator using shift operator polynomials.

The fact that the one step ahead prediction error, $\tilde{y}(t|t-1)$, constitutes innovations for the process $y(t)$ defined by (1.1) and (1.2), makes it straight forward to obtain the k-lag smoother, $\hat{z}(t|t+k)$, as a modification of the prediction estimate $\hat{z}(t|t-1)$:

$$\hat{z}(t|t+k) = \hat{z}(t|t-1) + \sum_{s=t}^{t+k} E\{\tilde{z}(t|t-1)\,\tilde{y}^T(s|s-1)\}\cdot$$

$$\cdot\{E\,\tilde{y}(s|s-1)\,\tilde{y}^T(s|s-1)\}^{-1}\,\tilde{y}(s|s-1) \qquad (2.1)$$

where

$$\tilde{z}(t|t-1) = z(t) - \hat{z}(t|t-1) = \tilde{y}(t|t-1) - e(t)$$

See for instance Kailath and Frost [6].

The formula (2.1) is valid also for multioutput, timevarying systems, and it describes the optimal start up of the estimator.

In the present context with only one signal, z, in noise, the structure of (2.1) is very simple, especially after that the start up transients have died off. The estimator is then most easily described using shift operator polynomials.

In order to get the stationary one step ahead predictor from the A and C polynomials given in (1.1) it is required to solve a stationary Riccati-equation or to make a spectral factorization.

The innovations representation of y can be written as

$$A(q^{-1})\ y(t) = D(q^{-1})\ \varepsilon(t) \tag{2.2}$$

where the innovations $\varepsilon(t) = \tilde{y}(t|t-1)$ have the variance σ_ε^2 and where

$$D(q^{-1}) = 1 + d_1 q^{-1} + \ldots + d_n q^{-n}.$$

The coefficients d_i and the variance σ_ε^2 are obtained from the spectral factorization

$$\begin{aligned} &\sigma_\varepsilon^2 (1+d_1x+\ldots+d_nx^n)(x^n+d_1x^{n-1}+\ldots+d_n) = \\ &= \sigma_v^2 (c_1x+\ldots+c_nx^n)(c_1x^{n-1}+\ldots+c_n) + \\ &\quad + \sigma_e^2 (1+a_1x+\ldots+a_nx^n)(x^n+\ldots+a_n) \end{aligned} \tag{2.3}$$

or

$$\sigma_\varepsilon^2\ D(x)\ D^*(x) = \sigma_v^2\ C(x)\ C^*(x) + \sigma_e^2\ A(x)\ A^*(x)$$

If it is required that D(x) has all its zeroes outside the unit circle, eq (2.3) has a unique solution.

The one step ahead predictor is now obtained as

$$D(q^{-1})\ \hat{z}(t|t-1) = [D(q^{-1}) - A(q^{-1})]\ y(t) \tag{2.4}$$

The computational work to get the stationary coefficients in (2.1) from A and D is the easy part as compared to the spectral factorization to obtain D from A and C.

Theorem 2.1: Let z and y be defined by (1.1) and (1.2). Then, $\hat{z}(t|t+k)$, the stationary k-step smoothing estimate of z(t), can be obtained by

$$\hat{z}(t|t+k) = y(t) - \frac{\sigma_e^2}{\sigma_\varepsilon^2} F_k(q)\cdot[y(t) - \hat{z}(t|t-1)] \tag{2.5}$$

where F_k is defined as

$$F_k(x) = \sum_{i=0}^{k} f_i x^i \tag{2.6}$$

$$F(x) = \sum_{i=0}^{\infty} f_i x^i \tag{2.7}$$

and computed from A and D of (2.2) using

$$A(x) = D(x)\cdot F(x) \tag{2.8}$$

and where

$$\frac{\sigma_e^2}{\sigma_\varepsilon^2} = \frac{d_n}{a_n} \tag{2.9}$$

Proof: The smoothing estimate $\hat{z}(t|t+k)$ is defined as

$$\hat{z}(t|t+k) = E\ [z(t)|y(0),\ldots,y(t+k)]$$

and it can be rewritten using the fact that the innovations $\varepsilon(t)$ are independent and form a sufficient statistics for y.

$$\hat{z}\ (t|t+k) = E\ [y(t)-e(t)|\varepsilon(0),\ldots,\varepsilon(t+k)] =$$

$$= y(t) - \sum_{i=0}^{t+k} E\ [e(t)|\varepsilon(i)] \tag{2.10}$$

The equations (1.1), (1.2) and (2.2) give

$$A(q^{-1})y(t) = D(q^{-1})\varepsilon(t) = C(q^{-1})v(t) + A(q^{-1})e(t)$$

or

$$\varepsilon(t) = \frac{C(q^{-1})}{D(q^{-1})} v(t) + \frac{A(q^{-1})}{D(q^{-1})} e(t)$$

But C/D and A/D are two stable systems, so that $\varepsilon(t)$ can also be expressed using their impulse responses, say $\{g_i\}_0^\infty$ and $\{f_i\}_0^\infty$:

$$\varepsilon(t) = \sum_{j=0}^{\infty} f_j \, q^{-j} \, e(t) + \sum_{j=0}^{\infty} g_j \, q^{-j} \, v(t)$$

Thus

$$E[e(t)\varepsilon(i)] = E[e(t) \sum_{j=0}^{\infty} f_j e(i-j)] = \begin{cases} f_{i-t} \cdot \sigma_e^2 & i \geq t \\ 0 & i < t \end{cases} \qquad (2.11)$$

so that

$$E[e(t)|\varepsilon(i)] = E[e(t)\varepsilon(i)] \; [E\varepsilon^2(i)]^{-1} \, \varepsilon(i) = \begin{cases} \dfrac{\sigma_e^2}{\sigma_\varepsilon^2} \cdot f_{i-t} \cdot \varepsilon(i) & i \geq t \\ 0 & i < t \end{cases}$$

and therefore

$$\hat{z}(t|t+k) = y(t) - \sum_{i=0}^{k} \frac{\sigma_e^2}{\sigma_\varepsilon^2} \cdot f_i \, q^i \, \varepsilon(t)$$

The equations (2.7) and (2.8) are another way of defining the impulse response, and the definition (2.6) completes the proof of (2.5). The equation (2.10) follows immediately from (2.3) with x = 0.

□

It should be noted that $a_n \neq 0$ by the definition of the order n, and the order of C is less than or equal to n.

This means that there must be no "white noise component" in the signal z. If there were, d_n/a_n would not give the relative contribution of the measurement noise, and such information would have to be supplied in some other way.

The interpretation of the weighting coefficients f_i as the first k values of the impulse response of the whitening filter A/D is very appealing, and they can be calculated using a simple recursion.

Most of the improvement with smoothing is obtained by the first few lags. The number of lags that should be used in a certain application depends on the C and A polynomials and the signal to noise ratio, cf for instance Chirarattananon and Anderson [5], and Van Trees [14, p 497].

The variance of the smoothing estimate can however also be expressed in the f_i coefficients:

Theorem 2.2: The error variance of the estimate $\hat{z}(t|t+k)$ of (2.5) is given by

$$\sigma_k^2 = \text{var}\,[z(t)-\hat{z}(t|t+k)] = \sigma_e^2\left[1-\frac{\sigma_e^2}{\sigma_\varepsilon^2}\sum_{i=0}^{k} f_i^2\right] \tag{2.12}$$

Proof: The proof is straightforward calculations using (2.5) and (2.11)

$$\sigma_k^2 = E\left[e(t) - \frac{\sigma_e^2}{\sigma_\varepsilon^2}\sum_{i=0}^{k} f_i\,\varepsilon(t+i)\right]^2 =$$

$$= E\left\{e^2(t)+\left[\frac{\sigma_e^2}{\sigma_\varepsilon^2}\sum_{i=0}^{k} f_i\,\varepsilon(t+i)\right]^2 - 2\,e(t)\frac{\sigma_e^2}{\sigma_\varepsilon^2}\sum_{i=0}^{k} f_i\,\varepsilon(t+i)\right\} =$$

$$= \sigma_e^2 + \frac{\sigma_e^4}{\sigma_\varepsilon^4}\sum_{i=0}^{k} f_i^2\cdot\sigma_\varepsilon^2 - 2\frac{\sigma_e^2}{\sigma_\varepsilon^2}\sum_{i=0}^{k} f_i\,E[e(t)\varepsilon(t+i)] =$$

$$= \sigma_e^2\left[1-\frac{\sigma_e^2}{\sigma_\varepsilon^2}\sum_{i=0}^{k} f_i^2\right]$$

□

The limit of σ_k^2 as k tends to infinity

$$\sigma_\infty^2 = \sigma_e^2\left[1-\frac{\sigma_e^2}{\sigma_\varepsilon^2}\sum_{i=0}^{\infty} f_i^2\right] = \sigma_e^2\left[1-\frac{\sigma_e^2}{\sigma_\varepsilon^2}\cdot I\right]$$

where I can be calculated directly from A and D using complex integrals:

$$I = \frac{1}{2\pi i} \oint \frac{A(z)A(z^{-1})}{D(z)D(z^{-1})} \frac{dz}{z}$$

which is most easily computed according to Åström [1, p 121].

Thus k can be chosen so that σ_k^2 is less than for instance 5 % larger than σ_∞^2:

$$(\sigma_k^2 - \sigma_\infty^2)/\sigma_\infty^2 = \frac{I - \sum_{i=0}^{k} f_i^2}{\sigma_\varepsilon^2/\sigma_e^2 - I} < 0.05$$

or

$$\sum_{i=0}^{k} f_i^2 > 1.05 \cdot I - 0.05\ \sigma_\varepsilon^2/\sigma_e^2$$

which is easy to test, while the f_i's are computed.

The mechanization of the equations (2.4) and (2.5) can be done in a number of ways. It immediately follows that

$$\hat{z}(t|t+k) = y(t) - \frac{\sigma_e^2}{\sigma_\varepsilon^2} \cdot F_k(q)\left[1 - \frac{D(q^{-1})-A(q^{-1})}{D(q^{-1})}\right] y(t) =$$

$$= \frac{D(q^{-1}) - \frac{\sigma_e^2}{\sigma_\varepsilon^2} F_k(q) \cdot A(q^{-1})}{D(q^{-1})}\ y(t) \qquad (2.13)$$

so that the minimal order, stable, k-lag smoother has a realization with a state dimension $n+k$.

These state variables have however no physical meaning, and it is more attractive to retain the original structure of (2.4) and (2.5). This will require k old y and k old $\hat{z}(t|t-1)$ or ε values, provided that k is larger than n. Otherwise 2n values would be required. The order is thus $2 \cdot \max(n,k)$.

In the self-tuning algorithm in the next chapter n y-values and n ε-values have to be stored anyhow for the identification algorithm.

A slight modification of the smoother (2.5) shows how it approaches "the unrealizable Wiener filter", see [14].

$$\lim_{k\to\infty} \hat{z}(t|t+k) = \lim_{k\to\infty} \left[y(t) - \frac{\sigma_e^2}{\sigma_\varepsilon^2} F_k(q)[y(t)-\hat{z}(t|t-1)] \right] =$$

$$= y(t) - \frac{\sigma_e^2}{\sigma_\varepsilon^2} \frac{A(q)}{D(q)} \varepsilon(t) = y(t) - \frac{\sigma_e^2}{\sigma_\varepsilon^2} \frac{A(q)}{D(q)} \cdot \frac{A(q^{-1})}{D(q^{-1})} y(t) =$$

$$= \frac{\sigma_\varepsilon^2 D(q)D(q^{-1}) - \sigma_e^2 A(q)A(q^{-1})}{\sigma_\varepsilon^2 D(q)D(q^{-1})} y(t) =$$

$$= \frac{\sigma_v^2 C(q)C(q^{-1}) / [A(q)A(q^{-1})]}{[\sigma_v^2 C(q)C(q^{-1}) + \sigma_e^2 A(q)A(q^{-1})]/[A(q)A(q^{-1})]} y(t) =$$

$$= \frac{S_z}{S_z + S_e} \cdot y(t)$$

where S_z and S_e are the spectral densities of the signal z and the measurement noise respectively.

3. A SELF-TUNING SMOOTHER

In section 2 the fixed lag smoother was derived for known processes. In this section we will show how it is possible to make a self-tuning smoother which automatically adjusts its parameters when the process and the variances of the noise processes are unknown.

To synthesize the optimal smoother we have to know the process, i e the polynomials A and C and the residuals. The idea is now based on the observation that the one step ahead prediction of z(t) is the same as the one step ahead prediction of y(t), i e

$$\hat{z}(t|t-1) = \hat{y}(t|t-1)$$

The predictor of y(t) is given by

$$\hat{y}(t|t-1) = \frac{D(q^{-1})-A(q^{-1})}{D(q^{-1})}\, y(t) = \frac{D(q^{-1})-A(q^{-1})}{A(q^{-1})}\, \varepsilon(t) \qquad (3.1)$$

In Wittenmark [16] it is shown how it is possible to construct a self-tuning predictor of an unknown process of the form (2.2). The predictor consists of two parts. First the parameters of the unknown process are estimated using some recursive estimation method. Secondly the prediction is done using (3.1). In [16] the polynomials A and D-A were estimated directly. In this application the parameters in the A and D polynomials will be estimated using the method of Extended Least Squares (ELS) or the Real-time Maximum Likelihood method (RML). Different recursive identification methods are compared in [13]. The choice of identification method will be discussed later in this section.

The algorithm

The self-tuning smoother can now be described in the following steps:

Step 1: Estimate the parameters a_i and d_i, $i=1,\dots,\hat{n}$ in the polynomials A and D using ELS or the RML identification method. The estimated polynomials are denoted by $\hat{A}$ and $\hat{D}$.

Step 2: Compute $\hat{F}_k(q)$ from $\hat{A}(q) = \hat{F}(q)\ \hat{D}(q)$ and compute the smoothing estimate from

$$\hat{z}(t-k|t) = y(t-k) - \frac{\hat{d}_{\hat{n}}}{\hat{a}_{\hat{n}}}\ \hat{F}_k(q)\ \hat{\varepsilon}(t-k) \tag{3.2}$$

where

$$\hat{\varepsilon}(t-k) = y(t-k) - \hat{z}(t-k|t-k-1) = \frac{\hat{A}(q^{-1})}{\hat{D}(q^{-1})}\ y(t-k)$$

(Compare eq (2.5) and (3.1).)

The two steps of the algorithm are repeated at each step of time. The estimation routines and (3.2) are well suited for recursive calculations.

Notice that the algorithm estimates the parameters in the process (2.2) and not in the process (1.2). Thus it is not necessary to make any spectral factorization. Further the quotient $\sigma_e^2/\sigma_\varepsilon^2$ can be computed directly using (2.9).

Estimation method

The estimation of the A and D polynomials in the innovation model (2.2) can be done using different estimation routines. The extended least squares method has the advantage that it is easy to implement. Further the computations in each step of time will be moderate. It has, however, been shown in [8] that the ELS method does not always converge. If the ELS method converges then it will converge to the true parameter values provided that the order of the model is sufficiently high.

It has been shown that the RML method always converges for ARMA-processes (2.2) [3]. The convergence rate can, however, be rather slow and different modifications can be done in order to speed up the rate of convergence. Further the RML algorithm is more complex than the ELS algorithm. The simulations presented in Section 4 have been done using the ELS method.

If the process has time varying parameters it is possible to modify the estimation routines in such a way that old data will be forgotten. This can easily be done by introducing a forgetting factor, λ. If $\lambda = 1$, all data have the same weight. If $\lambda < 1$, olda data will be exponentially forgotten. The forgetting factor will also influence the rate of convergence and a time varying λ can be used in RML to increase the rate of convergence.

Asymptotic properties

Theorem 3.1: Assume that the self-tuning smoother defined by Step 1 and 2 above is used on unknown processes of the type (1.2). Further it is assumed that the real time maximum likelihood method is used with the order of the model $\hat{n} = n$. The self-tuning smoother will then converge to the optimal smoother (2.5) that can be derived for known processes.

Proof: Using the result in [3] it is found that the estimates always converges, i e $\hat{A} \to A$ and $\hat{D} \to D$ as $t \to \infty$ and the result follows.

Remark 1: If $\hat{n} > n$ the estimated polynomials $\hat{A}$ and $\hat{D}$ will contain a common factor. If this factor is not equal to zero then the algorithm will still converge to the optimal smoother. The common factor will be zero only if $a_n = d_n = 0$, which can be tested for.

Remark 2: If the ELS-method is used for the estimation then the theorem has to be modified. It is then possible to state that if the estimation converges then the parameter estimates will converge to the true values in the innovation model, and the optimal smoother will thus be obtained. There are, however, processes for which the estimates do not converge.

Parameters of the algorithm

In order to use the self-tuning smoother some parameters have to be chosen. These are:

- The order of the estimated model, $\hat{n}$
- The initial values of the parameter estimates
- The covariance matrix of the initial errors of the parameter estimates
- The forgetting factor, λ
- The lag in the smoother, k.

The order of the model can be chosen by testing different orders and comparing the accumulated sum of the squares of the prediction error, i e by comparing $\Sigma[y(t)-\hat{y}(t|t-1)]^2 = \Sigma\, \hat{\varepsilon}(t)^2$ for different orders of $\hat{n}$. Statistical methods for choosing the order of the model are discussed for instance in [2].

The initial estimates of the parameter estimates are not crucial. The estimation routines use to get parameter estimates that are not too bad fairly quickly, especially if the covariance matrix of the initial values of the parameters is large or if a forgetting factor is used.

In Section 2 a way of determining the lag k was discussed. This procedure can be used on-line in order to get a good value of k. Simulations have indicated that a good way to start up the algorithm is to start with $k = 0$ and change k according to the rule in Section 2 when the parameter estimates have stabilized. The tuning of k has however not been fully analyzed and tested.

4. EXAMPLES

In this section some simulations are shown which illustrate the properties of the self-tuning smoother.

Example 4.1

Consider the process

$$\begin{cases} z(t) = \dfrac{1}{1+aq^{-1}} v(t-1) & v \in N(0,\sigma_v) \\ y(t) = z(t) + e(t) & e \in N(0,\sigma_e) \end{cases}$$

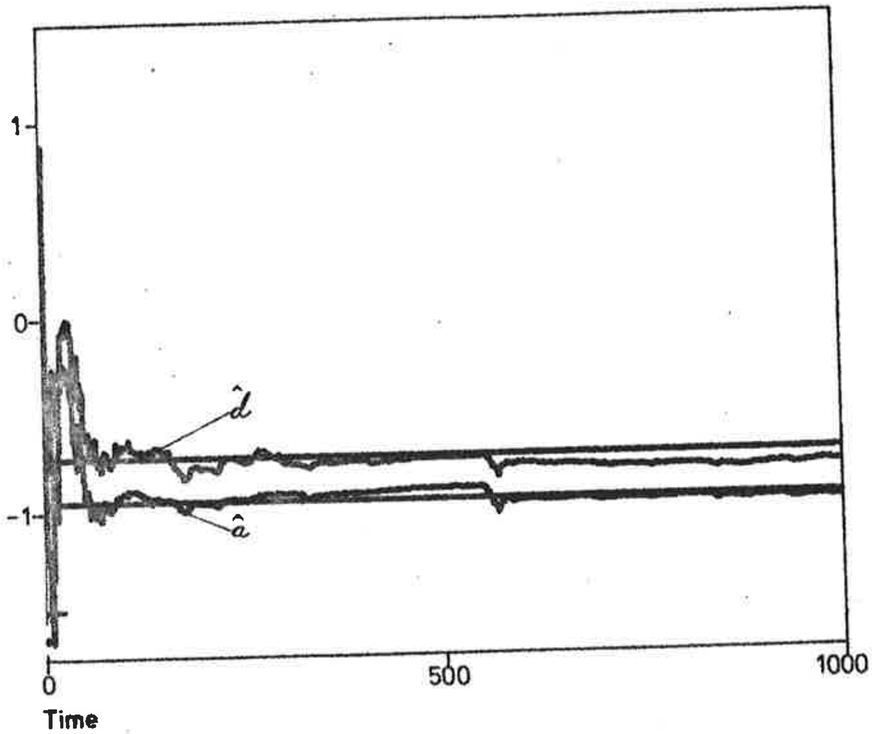
where $a = -0.95$, $\sigma_v^2 = 1$, and $\sigma_e^2 = 10$. This process is used in [5]. In this case the variance of the filter estimate ($k = 0$) is equal to 2.41 and the minimal error variance $\sigma_\infty^2 = 1.58$. The self-tuning smoother has been compared with the optimal smoother for different values of k. In this case the innovation model has the form

$$y(t) = \frac{1+dq^{-1}}{1+aq^{-1}} \varepsilon(t)$$

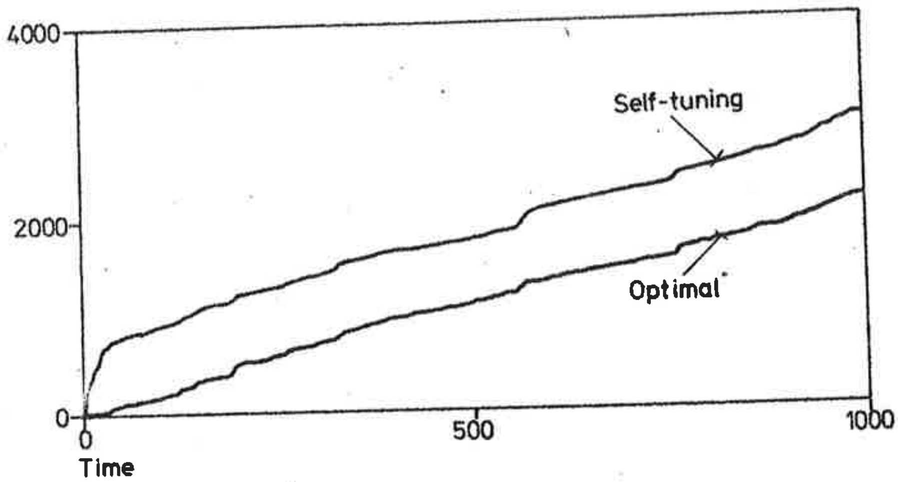
Figure 4.1 shows the parameter estimates $\hat{a}$ and $\hat{d}$ when the ELS method has been used. The straight lines show the true values. The estimation routine finds fairly good estimates after approximately 75 steps of time. The jump in the estimates at $t \approx 575$ is due to the noise realization.

Figure 4.2 shows the accumulated loss $V_t = \sum_{s=0}^{t} [z(s)-\hat{z}(s|s)]^2$, i e $k = 0$, when the self-tuning and the optimal smoother have been used. Apart from the initial loss the self-tuning smoother will give approximately the same loss as the optimal smoother.

In Table 4.1 a comparison is done between simulations of the self-tuning and the optimal smoother for different values of k.



4.1 Parameter estimates for Example 4.1 when the method is used.



4.2 Accumulated loss $\Sigma[z(t)-\hat{z}(t|t)]^2$ when the self-ning and the optimal smoother are used on Example 4.1.

k	$\frac{V_{1000} - V_{201}}{800}$		$\frac{V_{2000} - V_{1001}}{1000}$	
	Self-tuning	Optimal	Self-tuning	Optimal
0	2.22	2.08	2.17	2.20
1	1.84	1.74	1.85	1.87
2	1.65	1.57	1.66	1.68
4	1.44	1.40	1.51	1.51
6	1.37	1.34	1.45	1.45
8	1.33	1.31	1.43	1.43

Table 4.1 Average loss for one simulation during the intervalls 201-1000 and 1001-2000 when the optimal and the self-tuning smoother are used.

From the table it can be seen that in stationarity the self-tuning smoother will have as good performance as the optimal smoother. It might be surprising that the self-tuning smoother gives a lower loss than the optimal smoother in the interval 1000-2000. This can be explained by the fact that the variances of the noise processes e and v are not exactly 10 and 1 respectively in this particular simulation.

Example 4.2

Consider the process

$$\begin{cases} z(t) = \dfrac{1}{1 - 1.6q^{-1} + 0.8q^{-2}} v(t-1) \\ y(t) = z(t) + e(t) \end{cases}$$

where the variances of v and e are 1 and 12 respectively. By making a spectral factorization it is found that the innovation model is given by

$$y(t) = \frac{1 - 1.2415q^{-1} + 0.5264q^{-2}}{1 - 1.6q^{-1} + 0.8q^{-2}} \varepsilon(t) \qquad (4.1)$$

The variance of ε is 18.24. For this example different ways of estimating $z(t)$ has been investigated. Table 4.2 shows the expected error variance for different methods.

	Var $\tilde{z}(t)$
No smoothing $\hat{z}(t) = y(t)$	12.00
One step ahead predictor $\hat{z}(t) = \hat{y}(t\|t-1)$	6.24
Optimal filter estimate $(k = 0)$	4.10
Optimal smoothing estimate $(k = \infty)$	2.69

Table 4.2 Expected variance of $\tilde{z}(t) = z(t) - \hat{z}(t)$, for different methods.

In this example it is seen that substantial improvements can be obtained by using a fixed-lag smoother. By using the self-tuning smoother it is possible to obtain good smoothing estimates. Table 4.3 gives a comparison between a simulation when the optimal and the self-tuning smoother have been used.

k	$\frac{1}{1000} \sum_{1001}^{2000} [z(t) - \hat{z}(t\|t+k)]^2$ Optimal (based on (4.1))	Self-tuning
0	3.91	3.78
1	3.09	2.98
3	2.86	2.78
5	2.83	2.73

Table 4.3 Comparison between the optimal smoother based on eq (4.1) and the self-tuning smoother. The table shows the average loss per step for different values of k.

Also in this example it is found that the self-tuning smoother adapts to the realization of the noise processes. In the derivation of (4.1) it was assumed that the variances were 1 and 12 for v and e respectively. In the simulations the variance of e was about 10% higher than prescribed during the interval 1001-2000. This explains the fact that the self-tuning smoother gives a lower loss than the optimal smoother.

5. CONCLUSIONS

In this paper the optimal fixed lag smoother is derived for a discrete time signal which is corrupted by white noise. In Section 2 the smoother is derived for the case when the model of the signal is known. The smoother is determined in two steps. First the innovation model of the measured signals is determined by solving a Riccati equation or by making a spectral factorization. When the innovation model is known it is straight forward to compute the smoothing estimator. In Section 3 it was shown that it is possible to make a self-tuning smoother when the model of the signal is unknown. The self-tuning smoother estimates the parameters of the innovation model in real time. Based on the estimated parameter values the smoothing estimates are obtained. The self-tuning smoother will converge to the optimal smoother which would have been obtained knowing the process. Further it is not necessary to make any spectral factorization when using the self-tuning smoother. The computations in each step of time are moderate and the self-tuning smoother is well suited for real time applications.

There are many areas in the fields of communication and control where the self-tuning smoother can be used. Examples are transmission of digital signals, quality and production control, and measurements with low signal to noise ratio.

6. REFERENCES

[1] K J Åström: Introduction to stochastic control theory. Academic Press 1970.

[2] K J Åström and P Eykhoff: System identification - - A survey. Automatica 7 (1971) 123-162.

[3] K J Åström and T Söderström: Uniqueness of the maximum likelihood estimates of the parameters of an ARMA model. IEEE Trans Aut Control AC-19 (1974) 769-773.

[4] A E Bryson, Jr, and M Frazier: Smoothing for linear and nonlinear dynamic systems. TDR 63-119, Aero Sys Div, 353-364, Wright-Patterson Air Force Base, Ohio, 1963.

[5] S Chirarattananon and B D O Anderson: The fixed-lag smoother as a stable finite-dimensional linear system. Automatica 7 (1971) 657-669.

[6] T Kailath and P Frost: An innovations approach to least-squares estimation: Part II - Linear smoothing in additive white noise. IEEE Trans Aut Control AC-13 (1968) 646-655.

[7] R E Kalman: A new approach to linear filtering and prediction problems. J Basic Engng, Trans ASME Ser D 82 (1960) 35-45.

[8] L Ljung, T Söderström and I Gustavsson: Counterexamples to general convergence of a commonly used recursive identification method. IEEE Trans Aut Control AC-20 (1975) 643-652.

[9] D Q Mayne: A solution to the smoothing problem for linear dynamic systems. Automatica 4 (1966) 73-92.

[10] J S Meditch: Stochastic optimal linear estimation and control. McGraw-Hill, New York 1969.

[11] J B Moore: Discrete-time fixed lag smoothing algorithms. Automatica 9 (1973) 163-173.

[12] H E Rauch, F Tung and C T Striebel: Maximum likelihood estimates of linear dynamic systems. AIAA J 3 (1965) 1445-1450.

[13] T Söderström, L Ljung and I Gustavsson: A comparative analysis of recursive identification methods. Dept of Automatic Control, Lund Institute of Technology, Lund, Sweden, Report 7427 (1974).

[14] H L Van Trees: Detection, estimation and modulation theory. J Wiley 1968.

[15] N Wiener: Extrapolation, interpolation, and smoothing of stationary time series. Wiley, New York (1949).

[16] B Wittenmark: A self-tuning predictor. IEEE Trans Aut Control AC-19 (1974) 848-851.