Tracking time-varying coefficient-functions

Henrik Aa. Nielsen[1], Torben S. Nielsen[1],
Alfred K. Joensen[1], Henrik Madsen[1], and Jan Holst[2]

Abstract

A conditional parametric ARX-model is an ARX-model in which the parameters are replaced by smooth functions of an, possibly multivariate, external input signal. These functions are called coefficient-functions. A method, which estimates these functions adaptively and recursively, and hence allows for on-line tracking of the coefficient-functions is suggested. Essentially, in its most simple form, this method is a combination of recursive least squares with exponential forgetting and local polynomial regression. However, it is argued, that it is appropriate to let the forgetting factor vary with the value of the external signal which is argument of the coefficient-functions.

The properties of the modified method are studied by simulation. A particular feature is the this effective forgetting factor will adapt to the bandwidth used so that the effective number of observations behind the estimates will be almost independent of the actual bandwidth or of the type of bandwidth selection used (fixed or nearest neighbour). The choice of optimal bandwidth and forgetting is briefly discussed. Furthermore, a method for adaptive and recursive estimation in additive or varying-coefficient models is suggested. This method is a semi-parametric equivalent to the recursive prediction error method.

KEY WORDS: Adaptive and recursive estimation; Time-varying functions; Conditional parametric model; Additive model; Non-parametric method; Semi-parametric method.

[1]Department of Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark

[2]Department of Mathematical Statistics, Lund University, Lund Institute of Technology, S-221 00 Lund, Sweden

1 Introduction

The conditional parametric ARX-model is the linear ARX-model in which the parameters are replaced by smooth, but otherwise unknown, functions of one or more explanatory variables. These functions are called coefficient-functions. This class of models are used by Nielsen, Nielsen and Madsen (1997) to model a varying time delay. For on-line applications it is advantageous to allow the function estimates to be modified as data become available. Furthermore, because the system may change slowly over time, observations should be down-weighted as they become older. For this reason we propose an time-adaptive and recursive procedure, which is a combination of the adaptive recursive least squares method (Ljung, 1987) and locally weighted polynomial regression (Cleveland and Devlin, 1988). In the paper *adaptive* is used to denote that old observations are down-weighted, i.e. in the sense of *adaptive in time*.

Non-adaptive recursive estimation of a regression function is a related problem, which has been studied recently by Thuvesholmen (1997) using kernel methods and by Vilar-Fernández and Vilar-Fernández (1998) using local polynomial regression. Since these methods are non-adaptive one of the aspects considered in these papers is how to decrease the bandwidth as new observations become available. This problem do not arise for adaptive estimation since old observations are down-weighted and eventually disregarded as part of the algorithm.

Hastie and Tibshirani (1993) considered varying-coefficient models which are similar in structure to conditional parametric models and have close resemblance to additive models (Hastie and Tibshirani, 1990) with respect to estimation. However, varying-coefficient models include additional assumptions on the structure. Some specific time-series counterparts of these models are the functional-coefficient autoregressive models (Chen and Tsay, 1993*a*) and the nonlinear additive ARX-models (Chen and Tsay, 1993*b*).

In Section 2 a method for adaptive estimation in conditional parametric ARX-models is proposed and it is shown that the method is a natural extension of the adaptive recursive least squares method. A recursive formulation of the proposed method is derived in Section 3. Section 4 describes a modification of the method suitable e.g. for the case when the argument(s) of the functions exhibit cyclic behaviour. For non-adaptive and non-recursive non-parametric regression nearest neighbour techniques are well known; in Section 5 this subject is considered in the adaptive and recursive context. The method is summarized in Section 6. In Section 7 the suggested method combining recursive least squares and local polynomial regression is studied by simulation. Some further topics, such as long term fluctuations, optimal bandwidths, and optimal forgetting factors, are considered in Section 8. Finally, we conclude on the paper in Section 9.

2 Proposed Method

For simplicity the method is outlined as a generalization of exponential forgetting. However, the more general forgetting methods described by Ljung (1987) could also serve as a basis.

Using exponential forgetting and assuming observations at time $s = 1, \dots, t$ are available, the adaptive least squares estimate of the parameters $\boldsymbol{\theta}$ relating the explanatory variables $\mathbf{x}_s$ to the response y_s using the linear model $Ey_s = \mathbf{x}_s^T \boldsymbol{\theta}$ is found as

$$\hat{\boldsymbol{\theta}}_t = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{s=1}^{t} \lambda^{t-s} (y_s - \mathbf{x}_s^T \boldsymbol{\theta})^2,$$

where $0 < \lambda < 1$ is called the forgetting factor, see also (Ljung, 1987). The estimate can be written explicitly in matrix notation

$$\hat{\boldsymbol{\theta}}_t = \left(\mathbf{X}_t^T \boldsymbol{\Lambda}_t \mathbf{X}_t \right)^{-1} \mathbf{X}_t^T \boldsymbol{\Lambda}_t \mathbf{y}_t, \tag{1}$$

where $\mathbf{y}_t = [y_1 \; \dots y_t]^T$ is a vector of observations up to time t, $\boldsymbol{\Lambda}_t = \operatorname{diag}(\lambda^{t-1}, \lambda^{t-2}, \dots, \lambda, 1)$ is a diagonal weighting matrix, and finally

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_t^T \end{bmatrix}$$

is a (design) matrix in which row s is $\mathbf{x}_s$. When the estimator is written as the local (time) weighted least squares solution (1) this suggests that the estimator may also be defined locally with respect to some other explanatory variables $\mathbf{u}_t$. If the estimates are defined locally to some fixed point $\mathbf{u}$, called the fitting point, the adaptive estimate corresponding to this point can be expressed as

$$\hat{\boldsymbol{\theta}}_t(\mathbf{u}) = \left(\mathbf{X}_t^T \boldsymbol{\Lambda}_t \mathbf{W}_{u,t} \mathbf{X}_t\right)^{-1} \mathbf{X}_t^T \boldsymbol{\Lambda}_t \mathbf{W}_{u,t} \mathbf{y}_t, \tag{2}$$

where $\mathbf{W}_{u,t} = \operatorname{diag}(w_u(\mathbf{u}_1), \ldots, w_u(\mathbf{u}_t))$ is a diagonal weighting matrix in which the weights depend on the fitting point $\mathbf{u}$ and on the observations $\mathbf{u}_s;\ s = 1, \ldots, t$, see Appendix A. It is clear that (2) also may be written as

$$\hat{\boldsymbol{\theta}}_t(\mathbf{u}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{s=1}^{t} \lambda^{t-s} w_u(\mathbf{u}_s)(y_s - \mathbf{x}_s^T \boldsymbol{\theta})^2. \tag{3}$$

Estimators like this can be applied in parallel to a number of fitting points ($\mathbf{u}$) whereby the coefficient-functions $\boldsymbol{\theta}(\cdot)$ in the model $Ey_s = \mathbf{x}_s^T \boldsymbol{\theta}(\mathbf{u}_s)$ are estimated adaptively at a finite number of possible values of the argument. Interpolation can be used if the estimated function values are needed for other values of the argument.

In Section 3 it will be shown how the estimator (2) can be formulated recursively, but here we will briefly comment on the estimator and its relations to non-parametric regression. From (3) it is seen that locally to $\mathbf{u}$ the functions $\boldsymbol{\theta}(\mathbf{u})$ are approximated by constants. A special case is obtained if $\mathbf{X}$ is a column of ones, then simple calculations show that

$$\hat{\theta}_t(\mathbf{u}) = \frac{\sum_{s=1}^{t} \lambda^{t-s} w_u(\mathbf{u}_s) y_s}{\sum_{s=1}^{t} \lambda^{t-s} w_u(\mathbf{u}_s)}, \tag{4}$$

If $\lambda = 1$ this is a kernel estimator of $\theta(\cdot)$ in $Ey_s = \theta(\mathbf{u}_s)$, cf. (Härdle, 1990, p. 30). For this reason (4) is called an adaptive kernel estimator of $\theta(\cdot)$ and the general estimator (2) may be called an adaptive local constant or kernel estimator of the coefficient-functions $\boldsymbol{\theta}(\cdot)$ in the conditional parametric model $Ey_s = \mathbf{x}_s^T \boldsymbol{\theta}(\mathbf{u}_s)$. If lagged values of the dependent variable are included in $\mathbf{x}_s$ the model will be a conditional parametric ARX-model (CPARX-model), see also (Nielsen et al., 1997).

The local constant approximation is in general not very appropriate and local polynomial approximations will often be more suitable, see (Nielsen et al., 1997). In Appendix A it is shown how non-adaptive estimation in $Ey_s = \mathbf{x}_s^T \boldsymbol{\theta}(\mathbf{u}_s)$ can be performed using local polynomial approximations of the coefficient-functions and that the method corresponds to local constant estimation after redefining $\mathbf{X}_t$ in (2). For this reason the adaptive local constant estimator described above can be used to implement a general adaptive local polynomial estimator of the coefficient-functions $\boldsymbol{\theta}(\cdot)$. Therefore, methods aiming at adaptive kernel estimation and adaptive local polynomial estimation of a single regression function or of a set of coefficient-functions can all be described as (2).

3 Recursive formulation

Following Ljung (1987) the adaptive estimates (2) can be found recursively as

$$\hat{\boldsymbol{\theta}}_t(\mathbf{u}) = \hat{\boldsymbol{\theta}}_{t-1}(\mathbf{u}) + w_u(\mathbf{u}_t)\mathbf{R}_{u,t}^{-1}\mathbf{x}_t\left[y_t - \mathbf{x}_t^T\hat{\boldsymbol{\theta}}_{t-1}(\mathbf{u})\right] \tag{5}$$

and

$$\mathbf{R}_{u,t} = \lambda\mathbf{R}_{u,t-1} + w_u(\mathbf{u}_t)\mathbf{x}_t\mathbf{x}_t^T \ . \tag{6}$$

It is seen that existing numerical procedures can be applied in parallel to a number of fitting points $\mathbf{u}$, by replacing $\mathbf{x}_t$ and y_t with $\mathbf{x}_t\sqrt{w_u(\mathbf{u}_t)}$ and $y_t\sqrt{w_u(\mathbf{u}_t)}$, respectively. Note that $\mathbf{x}_t^T\hat{\boldsymbol{\theta}}_{t-1}(\mathbf{u})$ is a predictor of y_t locally with respect to $\mathbf{u}$ and for this reason it is used in (5). To predict y_t a predictor like $\mathbf{x}_t^T\hat{\boldsymbol{\theta}}_{t-1}(\mathbf{u}_t)$ is appropriate.

4 Modified updating formula

When $\mathbf{u}_t$ is far from the particular fitting point $\mathbf{u}$ it is clear from (5) and (6) that $\hat{\boldsymbol{\theta}}_t(\mathbf{u}) \approx \hat{\boldsymbol{\theta}}_{t-1}(\mathbf{u})$ and $\mathbf{R}_{u,t} \approx \lambda\mathbf{R}_{u,t-1}$, i.e. old observations are down-weighted without new information becoming available. This may result in abruptly changing estimates if $\mathbf{u}$ is not visited regularly, since the matrix $\mathbf{R}$ is decreasing exponentially in this case. Since we regard this as a serious practical problem it is proposed to modify (6) to ensure that the past is weighted down only when new information becomes available, i.e.

$$\mathbf{R}_{u,t} = \lambda v(w_u(\mathbf{u}_t);\lambda)\mathbf{R}_{u,t-1} + w_u(\mathbf{u}_t)\mathbf{x}_t\mathbf{x}_t^T, \tag{7}$$

where $v(\cdot\,;\lambda)$ is a nowhere increasing function on $[0;1]$ fulfilling $v(0;\lambda) = 1/\lambda$ and $v(1;\lambda) = 1$. Note that this requires that the weights span the interval ranging from zero to one. In this paper we consider only the linear function

$$v(w;\lambda) = 1/\lambda - (1/\lambda - 1)w,$$

for which (7) becomes

$$\mathbf{R}_{u,t} = (1 - (1-\lambda)w_u(\mathbf{u}_t))\mathbf{R}_{u,t-1} + \ w_u(\mathbf{u}_t)\mathbf{x}_t\mathbf{x}_t^T.$$

It is natural to denote $1-(1-\lambda)w_u(\mathbf{u}_t)$ the effective forgetting factor for point $\mathbf{u}$ at time t, $\lambda_{eff}^u(t)$.

5 Nearest neighbour bandwidth

A bandwidth specified according to the nearest neighbour principle is often used as a tool to vary the actual bandwidth used with the local density of the data. Assume in the following discussion that $\mathbf{u}_t$ is a stochastic variable and that the pdf $f(\cdot)$ of $\mathbf{u}_t$ is known and constant over t. Based on a nearest neighbour bandwidth the actual bandwidth can then be calculated for a number of fitting points $\mathbf{u}$ placed within the domain of $f(\cdot)$ and used to generate the weights $w_u(\mathbf{u}_t)$ used in the previous sections, see also Appendix A. The actual bandwidth $\hbar(\mathbf{u})$ corresponding to the point $\mathbf{u}$ will be related to the nearest neighbour bandwidth α by

$$\alpha = \int_{\mathbb{D}_u} f(\mathbf{z})d\mathbf{z}, \tag{8}$$

where $\mathbb{D}_u = \{\mathbf{z} \in \mathbb{R}^d \mid ||\mathbf{z}-\mathbf{u}|| \leq \hbar(\mathbf{u})\}$ is the neighbour-hood, d is the dimension of $\mathbf{u}$, and $||\cdot||$ is the Euclidean norm. In applications the density $f(\cdot)$ is often unknown. However, the selected model is based on an analysis which in turn is based on a set of observations. Hence, $f(\cdot)$ can be estimated, e.g. by the empirical pdf.

In order to select an appropriate value for α the effective number of observations used for estimation must be considered. In Appendix B it is shown that under certain conditions

$$\tilde{\eta}_u = \frac{1}{1 - E[\lambda_{eff}^u(t)]} = \frac{1}{(1-\lambda)E[w_u(\mathbf{u}_t)]} \tag{9}$$

is a lower bound on the effective number of observations (in the direction of time) corresponding to a point $\mathbf{u}$. When selecting α it is then natural to require that the number of observations within the bandwidth, i.e. $\alpha\tilde{\eta}_u$, is sufficiently large to justify the complexity of the model and the order of the local polynomial approximations. Of course α could also be based on a non-adaptive analysis of the data. In this case $\alpha\tilde{\eta}_u$ should be used to verify that the average forgetting factor is large enough. By assuming a stochastic process for $\{\mathbf{u}_t\}$ the effective number of observations in the direction of time, as described by the process $\{\eta_u(t)\}$ in Appendix B, can be simulated whereby the validity of $\tilde{\eta}_u$ can be addressed.

For $u_t \sim N(0,1)$, $\lambda = 0.99$, and when using the tricube weight-function (cf. Appendix A) the effective number of observations within the bandwidth, $\alpha\tilde{\eta}_u$, is displayed in Figure 1. It is seen that $\alpha\tilde{\eta}_u$ depends strongly on the fitting point u but only moderately on α. Figure 2 shows $\alpha\tilde{\eta}_u$ for λ ranging from 0.90 to 0.99 for $u = 0$ and $u = 2$, when $u_t \sim N(0,1)$. From this figure it is seen that, given the fitting point, $\alpha\tilde{\eta}_u$ is almost solely determined by λ. In conclusion, for the example considered, the effective forgetting factor $\lambda^u_{eff}(t)$ will be affected by the nearest neighbour bandwidth, so that the effective number of observations within the bandwidth will be strongly dependent on λ, but only weakly dependent on the bandwidth (α). The ratio between the rate at which the weights on observations goes to zero in the direction of time and the corresponding rate in the direction of u_t will be determined by α.

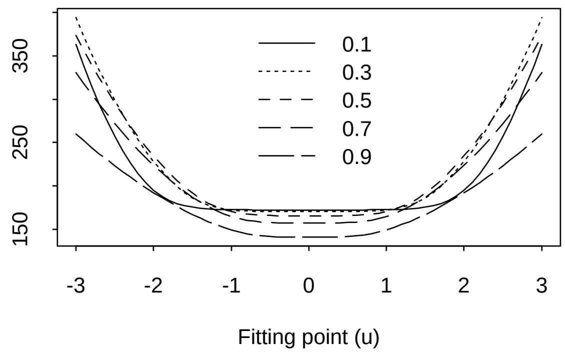

Figure 1: Effective number of observations within the bandwidth ($\alpha\tilde{\eta}_u(u)$) for $\alpha = 0.1, \ldots, 0.9$ and $\lambda = 0.99$.

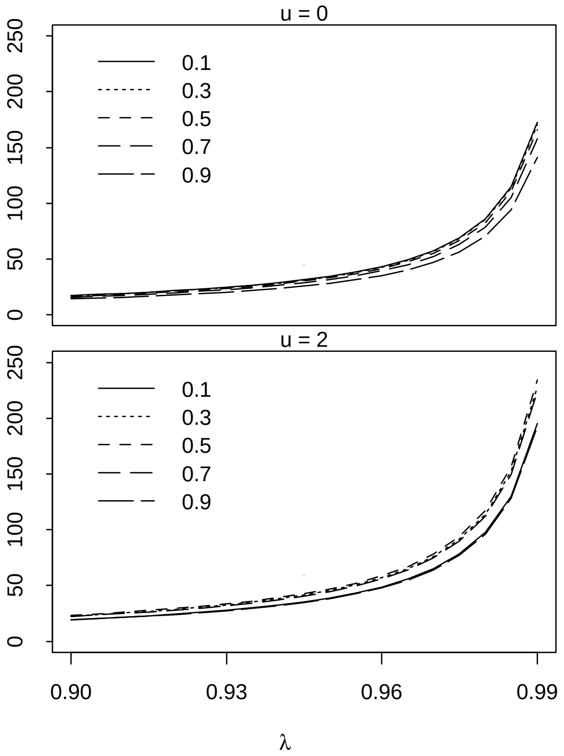

Figure 2: Effective number of observations within the bandwidth $(\alpha\tilde{\eta}_u(u))$ for $u = 0$ (top) and $u = 2$ (bottom), and for $\alpha = 0.1, \dots, 0.9$.

From Figure 1 it is tempting to infer that the adaptive estimates will have larger variance near the center of the distribution of u_t than in the tails of the distribution. However, although nearest neighbour bandwidths are used, local polynomial estimates have increased variance in border regions and this variance depend on the degree of the local approximation used. Therefore, some local approximations may result in increased variance near the center of the distribution, whereas other local approximations may result in increased variance in border regions.

These aspects are exemplified by simulations using the ARX-model

$$y_t = 0.9y_{t-1} + x_{t-1} + e_t, \tag{10}$$

where, $t = 1, \ldots, 2500$, $\{x_t\}$ is the input process, $\{y_t\}$ is the output process, and $\{e_t\}$ is a white noise error process. In the simulations $\{e_t\}$ is iid $N(0, 1 - 0.9^2)$ and $\{x_t\}$ is standard Gaussian white noise. Furthermore, the $\{u_t\}$ process is simulated as standard Gaussian white noise. Hereafter, estimation is performed using the modified method (4), assuming the model

$$y_t = a(u_{t-1})y_{t-1} + b(u_{t-1})x_{t-1} + e_t, \qquad (11)$$

where $a(\cdot)$ and $b(\cdot)$ are the coefficient-functions. A nearest neighbour bandwidth (α) of 0.7 is used, and $\lambda = 0.99$.

Using the tricube weight function (cf. Appendix A) and local quadratic approximations, the traces of the estimates of $a(u)$ and $b(u)$ are displayed in Figure 3. The traces indicate that the variance increases as the fitting point moves away from the center of the distribution, although Figure 1 shows that the effective number of observations within the bandwidth increases with the distance from the center of the distribution. Figure 4 shows the empirical standard deviation of the last 500 values of the estimates of $b(\cdot)$ for fitting points between -4 and 4 when local constant, linear, and quadratic approximations are used. The local constant approximation result in increased variance near the center of the distribution as compared to the border regions, the local linear approximation seems to have approximately constant variance over fitting points, and the local quadratic approximation clearly shows increased variance in border regions. Note that, although from the figure the local constant approximation seems superior, it may result in excess bias when the true function is not a constant, see also (Nielsen et al., 1997).

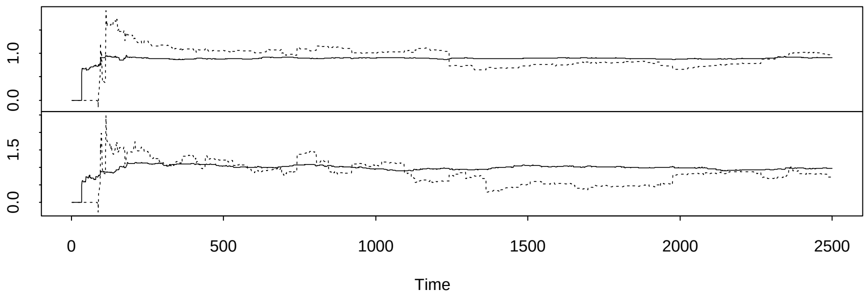

Figure 3: Traces of local quadratic adaptive estimates ($\alpha = 0.7$ and $\lambda = 0.99$) of $a(\cdot)$ (top) and $b(\cdot)$ (bottom) for $u = -4$ (dotted) and $u = 0$ (solid).

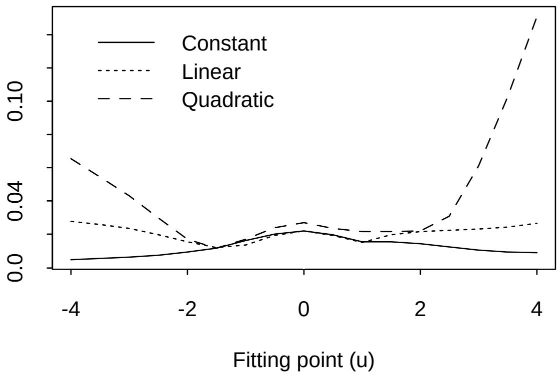

Figure 4: Empirical standard deviation of the last 500 adaptive estimates of $b(\cdot)$, for the local quadratic approximation with $\alpha = 0.7$ and $\lambda = 0.99$.

6 Summary of the method

To clarify the method described above the actual algorithm is briefly described in this section. As opposed to the previous part of the paper the distinction between the local constant and the local polynomial estimates, as described in Appendix A.2, will be made explicit. Thus, in this section we assume that at each time step measurements of the output y and the two sets of inputs $\mathbf{x}$ and $\mathbf{u}$ are received. The aim is to obtain adaptive estimates of the coefficient-functions in the model $Ey_t = \mathbf{x}_t^T \boldsymbol{\theta}(\mathbf{u}_t)$. This is accomplished by applying the method described in the previous part of the paper to the model $Ey_t = \mathbf{z}_t^T \boldsymbol{\phi}(\mathbf{u}_t)$, where $\mathbf{z}_t$ is defined by $\mathbf{x}_t$ and $\mathbf{u}_t$, see (16).

Besides λ in (4), prior to the application of the algorithm a number of fitting points $\mathbf{u}^{(j)}$; $j = 1, \dots, n_{fp}$ in which the coefficient-functions are to be estimated have to be selected. Furthermore the bandwidth associated with each of the fitting points $\hbar^{(j)}$; $j = 1, \dots, n_{fp}$ and the degrees of the approximating polynomials $d(1), \dots, d(p)$ have to be selected, where p denotes the number of coefficient-functions. Here the degree of the approximating polynomial for a particular coefficient-function will be fixed across fitting points. Finally, initial estimates of the coefficient-functions in the model corresponding to local constant estimates, i.e. $\hat{\boldsymbol{\phi}}_0(\mathbf{u}^{(j)})$ below, must be chosen. Also, the matrices $\mathbf{R}_{u^{(j)},0}$ must be chosen. One possibility is diag$(\epsilon,\ldots,\epsilon)$, where ϵ is a small positive number, see also the first paragraph in Section 8.

The selection of the degrees of each of the approximating polynomials and of fitting points and bandwidths associated with each of these requires some prior knowledge about the process $\{\mathbf{u}_t\}$ and about the smoothness of the coefficient-functions. The following considerations should be addressed:

- The placement of the extreme fitting points should be related to the range (region) spanned by $\{\mathbf{u}_t\}$, a 95% confidence region of $\mathbf{u}_t$ will often be appropriate.
- The distance between the fitting points should be related to the smoothness of the coefficient functions – the interpolation method used between fitting points should not influence the result to any significant degree.
- The degree of the approximating polynomials together with bandwidth should be related to the smoothness of the coefficient functions – the approximation must be appropriate within the bandwidth.

For simplicity, in the following description of the algorithm it will be assumed that a tricube weight function and a spherical kernel is used, cf. Appendix A. Furthermore it will be assumed that $\mathbf{R}_{u,t}$ can be inverted for all fitting points. Under these assumptions the algorithm can be described as:

For each time t: Loop over the fitting points $\mathbf{u}^{(j)};\ j=1,\ldots,n_{fp}$ and for each fitting point:

- Calculate the weight:
  $w_{u^{(j)}}(\mathbf{u}_t)=(1-(||\mathbf{u}_t-\mathbf{u}^{(j)}||/\hbar^{(j)})^3)^3$, if $||\mathbf{u}_t-\mathbf{u}^{(j)}||<\hbar^{(j)}$ and zero otherwise.
- Find the effective forgetting factor:
  $\lambda_{eff}^{(j)}(t)=1-(1-\lambda)w_{u^{(j)}}(\mathbf{u}_t)$.
- Construct the explanatory variables corresponding to local constant estimates as in (16) of Appendix A.2:
  $\mathbf{z}_t^T=[x_{t1}\mathbf{p}_{d(1)}^T(\mathbf{u}_t)\ldots x_{tp}\mathbf{p}_{d(p)}^T(\mathbf{u}_t)]$.
- Update $\mathbf{R}_{u^{(j)},t-1}$ using (4):
  $\mathbf{R}_{u^{(j)},t}=\lambda_{eff}^{(j)}(t)\mathbf{R}_{u^{(j)},t-1}+w_{u^{(j)}}(\mathbf{u}_t)\mathbf{z}_t\mathbf{z}_t^T$.

- Update $\hat{\boldsymbol{\phi}}_{t-1}(\mathbf{u}^{(j)})$ using (5):
  $\hat{\boldsymbol{\phi}}_t(\mathbf{u}^{(j)}) = \hat{\boldsymbol{\phi}}_{t-1}(\mathbf{u}^{(j)}) + w_{u^{(j)}}(\mathbf{u}_t)\mathbf{R}^{-1}_{u^{(j)},t}\mathbf{z}_t \left[y_t - \mathbf{z}_t^T \hat{\boldsymbol{\phi}}_{t-1}(\mathbf{u}^{(j)})\right].$

- Calculate the local polynomial estimates of the coefficient-functions as in (17) of Appendix A.2:
  $\hat{\boldsymbol{\theta}}^T(\mathbf{u}^{(j)}) = [\mathbf{p}^T_{d(1)}(\mathbf{u}^{(j)})\hat{\boldsymbol{\phi}}_1(\mathbf{u}^{(j)}) \dots \mathbf{p}^T_{d(p)}(\mathbf{u}^{(j)})\hat{\boldsymbol{\phi}}_p(\mathbf{u}^{(j)})].$

The algorithm could also be implemented using the matrix inversion lemma as in (Ljung and Söderström, 1983).

7 Simulations

The methods of updating $\mathbf{R}$, cf. (6) and (4), are studied by simulation using the model

$$y_t = a(t, u_{t-1})y_{t-1} + b(t, u_{t-1})x_t + e_t, \tag{12}$$

where $\{x_t\}$ is the input process, $\{u_t\}$ is the process controlling the coefficients, $\{y_t\}$ is the output process, and $\{e_t\}$ is a white noise standard Gaussian process. The coefficient-functions are simulated as

$$a(t,u) = 0.3 + (0.6 - \frac{1.5}{N}t)\exp\left(-\frac{(u - \frac{0.8}{N}t)^2}{2(0.6 - \frac{0.1}{N}t)^2}\right)$$

and

$$b(t,u) = 2 - \exp\left(-\frac{(u + 1 - \frac{2}{N}t)^2}{0.32},\right)$$

where $t = 1, \dots, N$ and $N = 5000$, i.e. $a(t,u)$ ranges from -0.6 to 0.9 and $b(t,u)$ ranges from 1 to 2. The functions are displayed in Figure 5. As indicated by the figure both coefficient functions are based on a Gaussian density in which the mean and variance varies linearly with time.

Adaptive estimates of the functions $a()$ and $b()$ are then found using the proposed procedure with the model

$$y_t = a(u_{t-1})y_{t-1} + b(u_{t-1})x_t + e_t. \tag{13}$$

For the adaptive estimation fitting points ranging from -2 to 2 in steps of 0.2 are considered. Initial estimates of the coefficient-functions are set

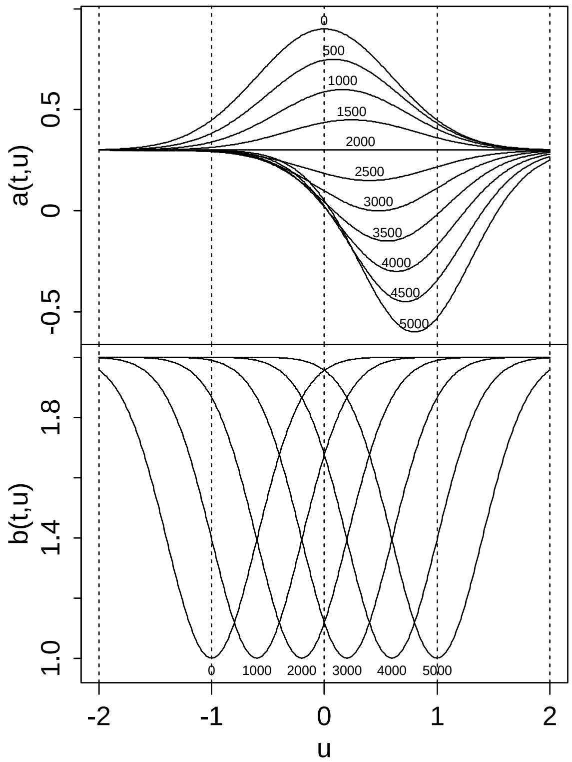

Figure 5: The time-varying coefficient-functions plotted for equidistant points in time.

to zero and during initialization the estimates are not updated, for the fitting point considered, until ten observations have received a weight of 0.5 or larger. Furthermore, in all cases, local linear approximations are used together with the tricube weight function, cf. Appendix A.

7.1 Varying the bandwidth

The data used in this section are generated using (12) where $\{x_t\}$ and $\{u_t\}$ now are zero mean $AR(1)$-processes with poles in 0.9 and 0.98, respectively. The variances in both the series are one and the series are mutually

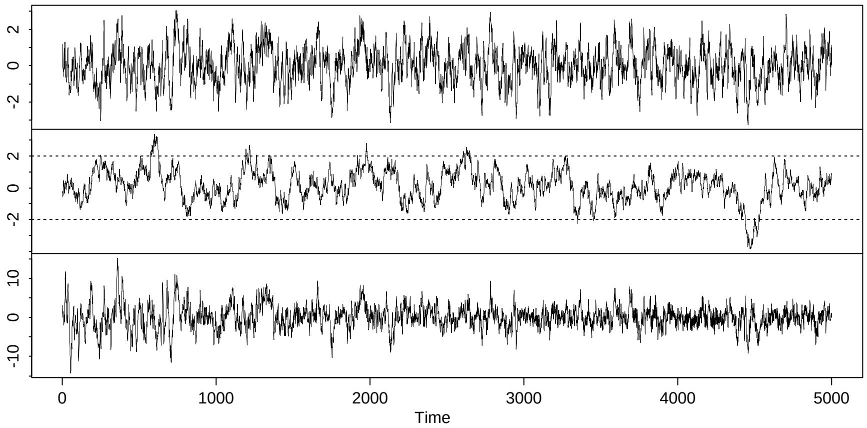

Figure 6: Simulated output (bottom) when x_t (top) and u_t (middle) are $AR(1)$-processes.

independent. In Figure 6 the data are displayed. Based on these data adaptive estimation in (13) are performed using nearest neighbour bandwidths, calculated assuming a standard Gaussian distribution for u_t.

The results obtained using the modified updating formula (4) are displayed for fitting points $u = -2, -1, 0, 1, 2$ in Figures 7 and 8. For the first 2/3 of the period the estimates at $u = -2$, i.e. $\hat{a}(-2)$ and $\hat{b}(-2)$, only gets updated occasionally. This is due to the correlation structure of $\{u_t\}$ as illustrated by the realization displayed in Figure 6. For less correlated series a better performance at fitting points placed in the tails of the pdf of u_t is found.

For both estimates the bias is most pronounced during periods in which the true coefficient-function changes quickly for values of u_t near the fitting point considered. This is further illustrated by Figure 5 and it is, for instance clear that adaption to $a(t, 1)$ is difficult for $t > 3000$. In general, the low bandwidth ($\alpha = 0.3$)) seems to result in large bias, presumably because the effective forgetting factor is increased on average, cf. Section 5. Similarly, the high bandwidth ($\alpha = 0.7$) result in large bias for $u = 2$ and $t > 4000$. A nearest neighbour bandwidth of 0.7 corresponds to an actual bandwidth of approximately 2.5 at $u = 2$ and since most values of u_t are below one, it is clear that the estimates at $u = 2$ will be highly influenced by the actual function values for u near one. From Figure 5 it is seen that

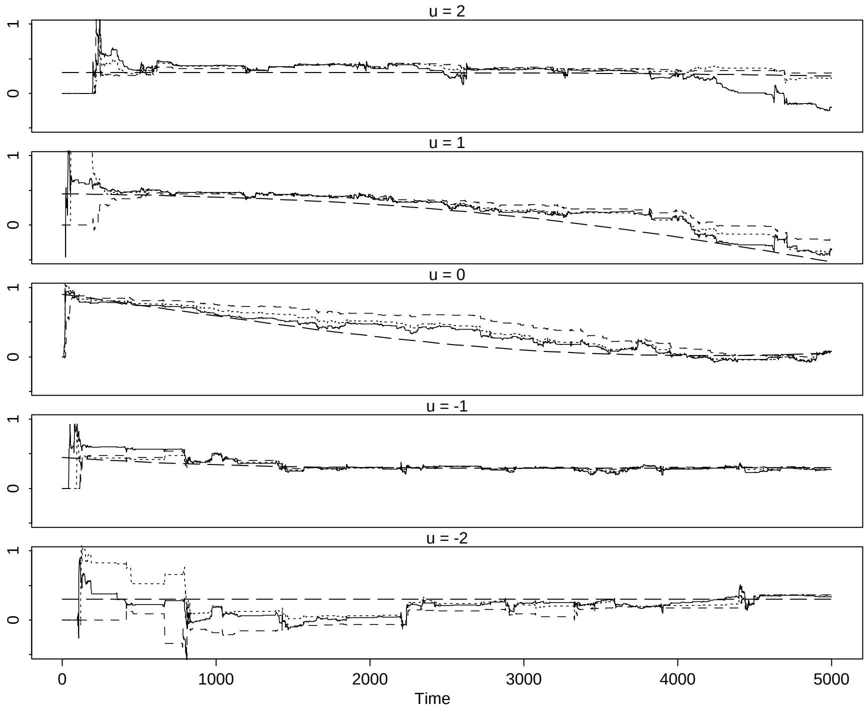

Figure 7: Adaptive estimates of $a(u)$ using local linear approximations and nearest neighbour bandwidths 0.3 (dashed), 0.5 (dotted), and 0.7 (full). True values are indicated by smooth dashed lines.

for $t > 4000$ the true values at $u = 1$ is markedly lower that the true values at $u = 2$. This explains the observed bias at $u = 2$, see Figure 9.

When the modified updating formula (4) is used the effective forgetting factor for a particular fitting point is increased when the bandwidth for that fitting point decreases. For this reason a fixed bandwidth across fitting points might be almost as appropriate as a nearest neighbour bandwidth. As mentioned in Section 5 a nearest neighbour bandwidth will often have to be based on an estimate of the pdf of u_t. This estimate might be rather uncertain in the tail of the distribution, especially when the series is highly autocorrelated. For this reason the ability to use a fixed bandwidth has important practical implications. One approach would be to calculate a

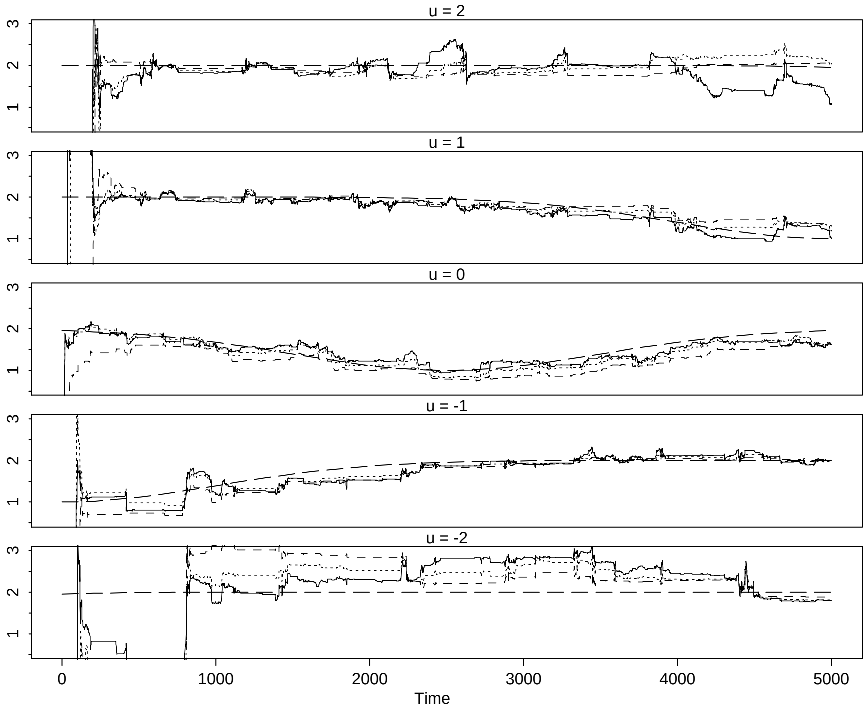

Figure 8: Adaptive estimates of $b(u)$ using local linear approximations and nearest neighbour bandwidths 0.3 (dashed), 0.5 (dotted), and 0.7 (full). True values are indicated by smooth dashed lines.

nearest neighbour bandwidth for the fitting point $\hat{E}[u_t] = \sum_{t=1}^{N} u_t/N$ and used this as a fixed bandwidth for all fitting points.

The approach is tested using $\alpha = 0.5$ and still assuming the pdf of u_t to be known. The results are not shown, but they are very similar to the results obtained for a nearest neighbour bandwidth.

A similar comparison is performed for the normal updating formula (6) and for $u = -2$ jumps in the estimates are observed when a fixed bandwidth is used. This is most likely due to the constant forgetting factor and the relatively low bandwidth at $u = -2$. This aspect is further illustrated in the following section.

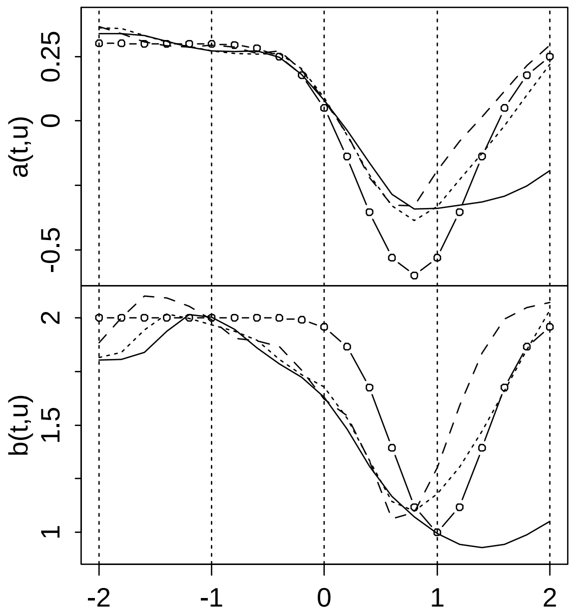

Figure 9: Adaptive estimates for the example considered in Section 7.1 at $t = 5000$ for $\alpha = 0.3$ (dashed), 0.5 (dotted), 0.7 (full). True values are indicated by circles.

7.2 Abrupt changes in $\{u_t\}$

One of the main advantages of the modified updating formula (4) over the normal updating formula (6) is that it does not allow fast changes in the estimates at fitting points which has not been visited by the process $\{u_t\}$ for a longer period. If, for instance, we wish to adaptively estimate the stationary relation between the heat consumption of a town and the ambient air temperature then $\{u_t\}$ contains an annual fluctuation and at some geographical locations the transition from, say, warm to cold periods may be quite fast. In such a situation the normal updating formula will, essentially, forget the preceding winter during the summer, allowing for large changes in the estimate at low temperatures during some initial period of the following winter. Actually, it is possible that, using the normal updating formula will result in a nearly singular $\mathbf{R}_t$.

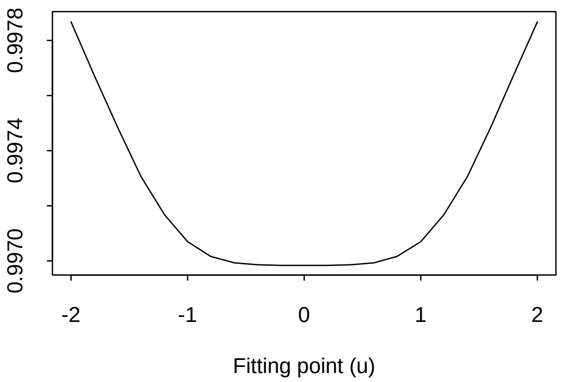

Figure 10: $\tilde{\lambda}(u)$ for a nearest neighbour bandwidth of 0.5 and $\lambda = 0.99$.

To illustrate this aspect 5000 observations are simulated using the model (12). The sequence $\{x_t\}$ is simulated as a standard Gaussian $AR(1)$-process with a pole in 0.9. Furthermore, $\{u_t\}$ is simulated as an iid process where

$$u_t \sim \begin{cases} N(0,1), & t = 1, \ldots, 1000 \\ N(3/2, 1/6^2), & t = 1001, \ldots, 4000 \\ N(-3/2, 1/6^2), & t = 4001, \ldots, 5000 \end{cases}$$

To compare the two methods of updating, i.e. (6) and (4), a fixed λ is used in (4) across the fitting points and the effective forgetting factors are designed to be equal. If $\tilde{\lambda}$ is the forgetting factor corresponding to (6) it can be varied with u as

$$\tilde{\lambda}(u) = E[\lambda^u_{eff}(t)] = 1 - (1 - \lambda)E[w_u(u_t)],$$

where $E[w_u(u_t)]$ is calculated assuming that u_t is standard Gaussian, i.e. corresponding to $1 \leq t \leq 1000$. Using a nearest neighbour bandwidth of 0.5 and $\lambda = 0.99$ the resulting $\tilde{\lambda}(u)$ is shown in Figure 10.

The corresponding adaptive estimates obtained for fitting point $u = -1$ are shown in Figure 11. The figure illustrates that for both methods the updating of the estimates stops as $\{u_t\}$ leaves the fitting point $u = -1$. Using the normal updating (6) of $\mathbf{R}_t$ its value is multiplied by $\tilde{\lambda}(-1)^{3000} \approx 0.00015$ as $\{u_t\}$ returns to the vicinity of the fitting point. This results in large fluctuations of the estimates, starting at $t = 4001$. As opposed to this our modified updating (4) does not lead to such fluctuations after $t = 4000$.

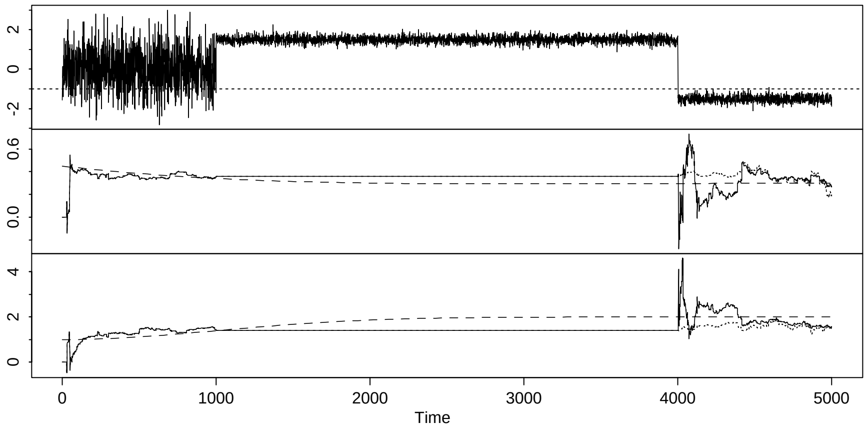

Figure 11: Realization of $\{u_t\}$ (top) and adaptive estimates of $a(-1)$ (middle) and $b(-1)$ (bottom), using the normal updating formula (solid) and the modified updating formula (dotted). True values are indicated by dashed lines.

8 Further topics

Long-term fluctuations: If $\{\mathbf{u}_t\}$ exhibits long-term fluctuations, e.g. annual fluctuations, the method can still be applied. However, if the usual approach of setting the initial estimates to zero is applied the time-span until the estimates are appropriate for all $\mathbf{u}$ will be long, maybe one year. Therefore, in case of long-term fluctuations in $\{\mathbf{u}_t\}$ it is crucial to use information from the analysis leading to the considered model. This information should be provided both in terms of $\hat{\boldsymbol{\theta}}_0(\mathbf{u})$ and $\mathbf{R}_{u,0}$.

Non-compact domain: If the domain of the pdf of $\mathbf{u}_t$ is non-compact we propose for on-line applications that fitting points $\mathbf{u}$ are selected within a reasonable range of the center of the distribution. If function estimates are needed outside this range we may use the estimates corresponding to the nearest point $\mathbf{u}$ used for estimation.

Effective number of observations: In Figure 1 it is shown how the effective number of observations within the bandwidth $\alpha\tilde{\eta}_u$ varies with the fitting point u when $u_t \sim N(0,1)$. To make $\alpha\tilde{\eta}_u$ independent of the fitting point the weights $w_u(\mathbf{u}_t)$ may be multiplied by a strictly positive factor, since this will not affect the estimates in the non-adaptive case. Alternatively, λ can be varied with the fitting point. If the weights are replaced by $w_u(\mathbf{u}_t)/E[w_u(\mathbf{u}_t)]$ then $\alpha\tilde{\eta}_u = \alpha/(1-\lambda)$ and it is seen that $\tilde{\eta}_u = 1/(1-\lambda)$ can be interpreted as the memory time constant T_0. If λ is varied with the fitting point as $\lambda(\mathbf{u}) = 1 - 1/(T_0 E[w_u(\mathbf{u}_t)])$ then $\tilde{\eta}_u = T_0$. In both cases the effective forgetting factor at time t is $1 - w_u(\mathbf{u}_t)/(T_0 E[w_u(\mathbf{u}_t)])$ and consequently the approaches are equivalent. For practical applications $E[w_u(\mathbf{u}_t)]$ must be estimated. Direct estimation by averaging observed weights will result in highly variable estimates, especially for fitting points placed in the tails of the distribution of $\mathbf{u}_t$. Since also the expression used for calculation of $\tilde{\eta}_u$ (9) is an approximation (see Appendix B) it is proposed to estimate the pdf of $\mathbf{u}_t$ based on a parametric family which fits the data reasonably well. Consequently, in many cases the Gaussian family of distributions is appropriate.

Optimal bandwidth and forgetting factor: So far in this paper it has been assumed that the bandwidths used over the range of $\mathbf{u}_t$ is derived from the nearest neighbour bandwidth α and it has been indicated how it can be ensured that the average forgetting factor is large enough.

However, the adaptive and recursive method is well suited for forward validation (Hjorth, 1994) and hence tuning parameters can be selected by minimizing, e.g. the root mean square of the one-step prediction error (using observed $\mathbf{u}_t$ and $\mathbf{x}_t$ to predict y_t, together with interpolation between fitting points to obtain $\hat{\boldsymbol{\theta}}_{t-1}(\mathbf{u}_t)$).

There are numerous ways to define the tuning parameters. A simple approach is to use (λ, α), cf. (4) and (8). A more ambiguous approach is to use both λ and $\hbar$ for each fitting point $\mathbf{u}$. Furthermore, tuning parameters controlling scaling and rotation of $\mathbf{u}_s$ may also be considered.

If n fitting points are used this amounts to $2n$, or more, tuning parameters. To make the dimension of the (global) optimization problem independent of n and to have $\lambda(\mathbf{u})$ and $\hbar(\mathbf{u})$ vary smoothly with $\mathbf{u}$ we may choose to restrict $\lambda(\mathbf{u})$ and $\hbar(\mathbf{u})$, or appropriate transformations of these (logit for λ and log for $\hbar$), to follow a spline basis (de Boor, 1978; Lancaster and Salkauskas, 1986). This is similar to the smoothing of spans described by Friedman (1984).

Local time-polynomials: In this paper local polynomial approximations in the direction of time is not considered. Such a method is proposed for usual ARX-models by Joensen, Nielsen, Nielsen and Madsen (1999). This method can be combined with the method described here and will result in local polynomial approximations where cross-products between time and the conditioning variables ($\mathbf{u}_t$) are excluded. It is, however, still an open question if the outlined extension are applicable from a practical point of view. Since the method described in this paper down weights observations both in the direction of time and $\mathbf{u}_t$ it requires a relatively large average forgetting factor. Hence a simultaneous local polynomial approximation of the development over time will require the forgetting factor to be increased further, possibly resulting in a method which is non-adaptive for practical purposes. However, if the initial values of the recursions are carefully selected the approach may prove valuable.

Adaptive estimation in additive models: Consider the additive model (Hastie and Tibshirani, 1990)

$$Ey_s = \mu + \sum_i f_i(\mathbf{u}_{i,s}), \tag{14}$$

where, in principle, each summand may be a conditional parametric model. Consequently, the varying-coefficient models of Hastie and Tibshirani (1993) are also included in (14).

Below a method for adaptive and recursive estimation in models like (14) is proposed. The method is inspired by the backfitting algorithm (Hastie and Tibshirani, 1990). At time step t the following steps are performed:

1. $\hat{\mu}_t$ is obtained through adaptive and recursive updating of (the constant) $\hat{\mu}_{t-1}$ using $y_t - \sum_i \hat{f}_{i,t-1}(\mathbf{u}_{i,t})$ as the dependent variable.

2. $\hat{f}_{j,t}$ is obtained equivalently, i.e. as described in this paper, but using $y_t - \hat{\mu}_{t-1} - \sum_{i \neq j} \hat{f}_{i,t-1}(\mathbf{u}_{i,t})$ as the dependent variable. Alternatively $\hat{\mu}_t$ could be used instead of $\hat{\mu}_{t-1}$.

3. $\hat{f}_{j,t}$ is adjusted by subtracting $\int_{\min\{u_j\}}^{\max\{u_j\}} \hat{f}_{j,t}(z)dz/(\max\{u_j\} - \min\{u_j\})$, and similarly for multivariate $\mathbf{u}$.

In step 3 the minimum and maximum refers to the minimum and maximum of the fitting points. The range of integration is not very important but step 3 is important to ensure that the level of y_t can only be handled by μ in the model. Furthermore, step 3 is not required for varying-coefficient models. Considering step 2 it is natural to use the most recent estimates at every instant of the algorithm. In this case the order in which we consider the functions to be estimated may be important.

The algorithm amounts to performing the iterations in the backfitting algorithm distributed over time steps resembling the recursive prediction error method (Ljung, 1987) in which a single Newton-Raphson iteration is performed at each time step.

9 Conclusion and Discussion

The conditionally parametric ARX-model (CPARX-model) is a conventional ARX-model in which the parameters are replaced by smooth functions of a (low-dimensional) input process. One possible application of these models is the modelling of varying time delays, cf. (Nielsen et al., 1997). For on-line applications the function estimates should be allowed to adapt to slow changes in the true, but unknown, functions. Although, other practical solutions may exist, the recursive approach is particularly useful in that a fairly small computational effort is required each time an observation becomes available. In this paper a method for adaptive and recursive estimation in CPARX-models are proposed. The method can be seen as a generalization or a combination of adaptive recursive least squares (Ljung, 1987), local polynomial regression (Cleveland and Devlin, 1988), and conditional parametric fits (Anderson, Fang and Olkin, 1994).

For some applications it may be possible to specify global polynomial approximations to the coefficient-functions of a CPARX-model. In this situation the adaptive recursive least squares method can be applied for tracking the parameters from which the estimates of the coefficient-functions can be calculated. However, if the argument(s) of the coefficient-functions only stays in parts of the space corresponding to the possible values of the argument(s) for longer periods this may seriously affect the coefficient-functions for other values of the argument(s), in that it corresponds to extrapolation using a fitted polynomial. This problem is effectively solved using the non-parametric model in combination with the modified updating formula suggested in this paper.

Adaptive and recursive estimation in CPARX-models will require a relatively large forgetting factor as compared to ARX-models. Furthermore, during part of the time, the function estimates may be updated for some values of their argument(s) while the estimates are left unchanged for other values. Therefore, for some practical applications, it will be crucial to initialize the recursions both in terms of the estimates and the precision hereof.

The modified updating formula bear resemblance selective forgetting (Ljung, 1987). Instead of using a forgetting factor of one for observations with zero weight a number slightly lower than one may be chosen. For applications where the functions to be estimated change substantially at some values of their argument, while these values are not visited for longer periods, this may be applicable since it will allow for faster adaption at the cost of increased variance of the estimates in these situations.

A Local polynomial estimation

In this appendix non-adaptive estimation in conditional parametric models is described. The model is of the form

$$y_s = \mathbf{x}_s^T \boldsymbol{\theta}(\mathbf{u}_s) + e_s; \quad s = 1, \dots, N, \tag{15}$$

where the response y_s is a stochastic variable, $\mathbf{u}_s$ and $\mathbf{x}_s$ are explanatory variables, e_s is i.i.d. $N(0, \sigma^2)$, $\boldsymbol{\theta}(\cdot)$ is a vector of unknown but smooth functions with values in $\mathbb{R}$, and $s = 1, \dots, N$ are observation numbers. When $\mathbf{u}_s$ is constant across the observations the model reduces to an ordinary parametric linear model.

A.1 Local constant estimates

Estimation in (15) aims at estimating the functions $\boldsymbol{\theta}(\cdot)$ within the space spanned by the observations of $\mathbf{u}_s; s = 1, \dots, N$. The functions are only estimated for distinct values of the argument $\mathbf{u}$. Below $\mathbf{u}$ denotes one of these fitting points and $\hat{\boldsymbol{\theta}}(\mathbf{u})$ denotes the estimates of the coefficient-functions, when the functions are evaluated at $\mathbf{u}$.

One solution to the estimation problem is to replace $\boldsymbol{\theta}(\mathbf{u}_s)$ in (15) with a constant vector $\boldsymbol{\theta}_u$ and fit the resulting model locally to $\mathbf{u}$, using weighted least squares. Below two similar methods of allocating weights to the observations are described, for both methods the weight function $W : \mathbb{R}_0 \to \mathbb{R}_0$ is a nowhere increasing function, $\mathbb{R}_0$ denotes the non-negative real numbers. In this paper the tricube weight function

$$W(u) = \begin{cases} (1-u^3)^3, & u \in [0;1) \\ 0, & u \in [1;\infty) \end{cases}$$

is used. Hence, $W : \mathbb{R}_0 \to [0,1]$.

In the case of a spherical kernel the weight on observation s is determined by the Euclidean distance $||\mathbf{u}_s - \mathbf{u}||$ between $\mathbf{u}_s$ and $\mathbf{u}$, i.e.

$$w_u(\mathbf{u}_s) = W\left(\frac{||\mathbf{u}_s - \mathbf{u}||}{\hbar(\mathbf{u})}\right).$$

A product kernel is characterized by distances being calculated for one dimension at a time, i.e.

$$w_u(\mathbf{u}_s) = \prod_j W\left(\frac{|u_{j,s} - u_j|}{\hbar(\mathbf{u})}\right),$$

where the multiplication is over the dimensions of $\mathbf{u}$. The scalar $\hbar(\mathbf{u}) > 0$ is called the bandwidth. If $\hbar(\mathbf{u})$ is constant for all values of $\mathbf{u}$ it is denoted a fixed bandwidth. If $\hbar(\mathbf{u})$ is chosen so that a certain fraction (α) of the observations fulfill $||\mathbf{u}_s - \mathbf{u}|| \leq \hbar(\mathbf{u})$ it is denoted a nearest neighbour bandwidth. If $\mathbf{u}$ has dimension of two or larger, scaling of the individual elements of $\mathbf{u}_s$ before applying the method should be considered, see e.g. (Cleveland and Devlin, 1988). Rotating the coordinate system in which $\mathbf{u}_s$ is measured may also be relevant.

A.2 Local polynomial estimates

If the bandwidth $\hbar(\mathbf{u})$ is sufficiently small the approximation of $\boldsymbol{\theta}(\cdot)$ as a constant vector near $\mathbf{u}$ is good. This implies that a relatively low number of observations is used to estimate $\boldsymbol{\theta}(\mathbf{u})$, resulting in a noisy estimate or large bias if the bandwidth is increased. See also the comments on kernel estimates in (Anderson et al., 1994).

It is, however, well known that locally to $\mathbf{u}$ the elements of $\boldsymbol{\theta}(\cdot)$ may be approximated by polynomials, and in many cases these will be good approximations for larger bandwidths than those corresponding to local constants. Local polynomial approximations are easily included in the method described. Let $\theta_j(\cdot)$ be the j'th element of $\boldsymbol{\theta}(\cdot)$ and let $\mathbf{p}_{d(j)}(\mathbf{u})$ be a column vector of terms in the corresponding d-order polynomial evaluated at $\mathbf{u}$, if for instance $\mathbf{u} = [u_1\ u_2]^T$ then $\mathbf{p}_2(\mathbf{u}) = [1\ u_1\ u_2\ u_1^2\ u_1u_2\ u_2^2]^T$. Furthermore, let $\mathbf{x}_s = [x_{1s} \dots x_{ps}]^T$. With

$$\mathbf{z}_s^T = \left[x_{1s}\mathbf{p}_{d(1)}^T(\mathbf{u}_s) \dots x_{js}\mathbf{p}_{d(j)}^T(\mathbf{u}_s) \dots x_{ps}\mathbf{p}_{d(p)}^T(\mathbf{u}_s)\right] \tag{16}$$

and

$$\hat{\boldsymbol{\phi}}^T(\mathbf{u}) = [\hat{\boldsymbol{\phi}}_1^T(\mathbf{u}) \dots \hat{\boldsymbol{\phi}}_j^T(\mathbf{u}) \dots \hat{\boldsymbol{\phi}}_p^T(\mathbf{u})],$$

where $\hat{\boldsymbol{\phi}}_j(\mathbf{u})$ is a column vector of local constant estimates at $\mathbf{u}$ corresponding to $x_{js}\mathbf{p}_{d(j)}(\mathbf{u}_s)$, estimation is handled as described in Section A.1, but fitting the linear model

$$y_s = \mathbf{z}_s^T \boldsymbol{\phi}_u + e_s; \quad i = 1, \dots, N,$$

locally to $\mathbf{u}$, indicated by the subscript parameter-vector.. Hereafter the elements of $\boldsymbol{\theta}(\mathbf{u})$ are estimated by

$$\hat{\theta}_j(\mathbf{u}) = \mathbf{p}_{d(j)}^T(\mathbf{u})\, \hat{\boldsymbol{\phi}}_j(\mathbf{u}); \quad j = 1, \dots p. \tag{17}$$

When $\mathbf{x}_s = 1$ for all s, i.e. $p = 1$, this method is identical to the method by Cleveland and Devlin (1988), with the exception that they center the elements of $\mathbf{u}_s$ used in $\mathbf{p}_{d(j)}(\mathbf{u}_s)$ around $\mathbf{u}$ and so $\mathbf{p}_{d(j)}(\mathbf{u}_s)$ must be recalculated for each value of $\mathbf{u}$ considered.

B Effective number of observations

Using the modified updating formula, as described in Section 4, the estimates at time t can be written as

$$\hat{\boldsymbol{\theta}}_t(\mathbf{u}) = \operatorname*{Argmin}_{\boldsymbol{\theta}} \sum_{s=1}^{t} \beta(t,s) w_u(\mathbf{u}_s)(y_s - \mathbf{x}_s^T \boldsymbol{\theta})^2,$$

where

$$\beta(t,t) = 1,$$

and, for $s < t$

$$\beta(t,s) = \prod_{j=s+1}^{t} \lambda_{eff}^{u}(j) = \lambda_{eff}^{u}(t)\beta(t-1,s).$$

It is then natural to define the effective number of observations (in the direction of time) as

$$\begin{aligned} \eta_u(t) &= \sum_{i=0}^{\infty} \beta(t,t-i) \\ &= 1 + \lambda_{eff}^{u}(t) + \lambda_{eff}^{u}(t)\lambda_{eff}^{u}(t-1) + \ldots \end{aligned} \tag{18}$$

Suppose that the fitting point $\mathbf{u}$ is chosen so that $E[\eta_u(t)]$ exists. Consequently, when $\{\lambda_{eff}^{u}(t)\}$ is i.i.d. and when $\bar{\lambda}_u$ denotes $E[\lambda_{eff}^{u}(t)]$, the average effective number of observations is

$$\bar{\eta}_u = 1 + \bar{\lambda}_u + \bar{\lambda}_u^2 + \ldots = \frac{1}{1-\bar{\lambda}_u}.$$

When $\{\lambda_{eff}^{u}(t)\}$ is not i.i.d., it is noted that since the expectation operator is linear, $E[\eta_u(t)]$ is the sum of the expected values of each summand in (18). Hence, $E[\eta_u(t)]$ is independent of t if $\{\lambda_{eff}^{u}(t)\}$ is strongly stationary, i.e. if $\{\mathbf{u}_t\}$ is strongly stationary. From (18)

$$\eta_u(t) = 1 + \lambda_{eff}^{u}(t)\eta_u(t-1)$$

is obtained, and from the definition of covariance it then follows that

$$\bar{\eta}_u = \frac{1 + Cov[\lambda_{eff}^{u}(t), \eta_u(t-1)]}{1-\bar{\lambda}_u} \geq \frac{1}{1-\bar{\lambda}_u}, \tag{19}$$

since $0 < \lambda < 1$ and assuming that the covariance between $\lambda_{eff}^{u}(t)$ and $\eta_u(t-1)$ is positive. Note that if the process $\{\mathbf{u}_t\}$ behaves such that if it has been near $\mathbf{u}$ for a longer period up to time $t-1$ it will tend to be near $\mathbf{u}$ at time t also a positive covariance is obtained. It is the experience of the authors that such a behaviour of a stochastic process is often encountered in practice.

As an alternative to the calculations above $\lambda_{eff}^{u}(t)\eta_u(t-1)$ may be linearized around $\bar{\lambda}_u$ and $\bar{\eta}_u$. From this it follows that when the variance of $\lambda_{eff}^{u}(t)$ and $\eta_u(t-1)$ is small then

$$\bar{\eta}_u \approx \frac{1}{1-\bar{\lambda}_u}.$$

Therefore we may use $1/(1-\bar{\lambda}_u)$ as an approximation to the effective number of observations and we suppose that in many practical applications it will be an lower bound, c.f. (19).

References

Anderson, T. W., Fang, K. T. and Olkin, I., eds (1994), *Multivariate Analysis and Its Applications*, Institute of Mathematical Statistics, Hayward, chapter Coplots, Nonparametric Regression, and conditionally Parametric Fits, pp. 21–36.

Chen, R. and Tsay, R. S. (1993*a*), 'Functional-coefficient autoregressive models', *Journal of the American Statistical Association* **88**, 298–308.

Chen, R. and Tsay, R. S. (1993*b*), 'Nonlinear additive ARX models', *Journal of the American Statistical Association* **88**, 955–967.

Cleveland, W. S. and Devlin, S. J. (1988), 'Locally weighted regression: An approach to regression analysis by local fitting', *Journal of the American Statistical Association* **83**, 596–610.

de Boor, C. (1978), *A Practical Guide to Splines*, Springer Verlag, Berlin.

Friedman, J. H. (1984), A variable span smoother, Technical Report 5, Laboratory for Computational Statistics, Dept. of Statistics, Stanford Univ., California.

Härdle, W. (1990), *Applied Nonparametric Regression*, Cambridge University Press, Cambridge, UK.

Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized Additive Models*, Chapman & Hall, London/New York.

Hastie, T. and Tibshirani, R. (1993), 'Varying-coefficient models', *Journal of the Royal Statistical Society, Series B, Methodological* **55**, 757–796.

Hjorth, J. S. U. (1994), *Computer Intensive Statistical Methods: Validation Model Selection and Bootstrap*, Chapman & Hall, London/New York.

Joensen, A. K., Nielsen, H. A., Nielsen, T. S. and Madsen, H. (1999), 'Tracking time-varying parameters with local regression', *Automatica* . To be published.

Lancaster, P. and Salkauskas, K. (1986), *Curve and Surface Fitting: An Introduction*, Academic, New York/London.

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

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

Nielsen, H. A., Nielsen, T. S. and Madsen, H. (1997), ARX-models with parameter variations estimated by local fitting, *in* Y. Sawaragi and S. Sagara, eds, '11th IFAC Symposium on System Identification', Vol. 2, pp. 475–480.

Thuvesholmen, M. (1997), 'An on-line crossvalidation bandwidth selector for recursive kernel regression', Lic thesis, Department of Mathematical Statistics, Lund University, Sweden.

Vilar-Fernández, J. A. and Vilar-Fernández, J. M. (1998), 'Recursive estimation of regression functions by local polynomial fitting', *Annals of the Institute of Statistical Mathematics* **50**, 729–754.