Maximum *a posteriori* sequence estimation using Monte Carlo particle filters

SIMON GODSILL[1], ARNAUD DOUCET[1] AND MIKE WEST[2]

[1]Signal Processing Laboratory
University of Cambridge
Cambridge CB2 1PZ, UK
{ad2,sjg}@eng.cam.ac.uk

[2]Institute of Statistics and Decision Sciences
Duke University
Durham NC USA 27708-0251
mw@isds.duke.edu

August 11, 2000

Abstract

We develop methods for performing maximum *a posteriori* (MAP) sequence estimation in non-linear non-Gaussian dynamic models. The methods rely on a particle cloud representation of the filtering distribution which evolves through time using importance sampling and resampling ideas. MAP sequence estimation is then performed using a classical dynamic programming technique applied to the discretised version of the state space. In contrast with standard approaches to the problem which essentially compare only the trajectories generated directly during the filtering stage, our method efficiently computes the optimal trajectory over all combinations of the filtered states. A particular strength of the method is that MAP sequence estimation is performed sequentially in one single forwards pass through the data without the requirement of an additional backward sweep (as would be the case for most smoothing methods such as the Kalman smoother). In simulation, the methods are shown to outperform standard methods for a given computational complexity. An application to estimation of a non-linear time series model and to spectral estimation for time-varying autoregressions is described.

Key Words: Bayesian estimation, Filtering, Monte Carlo methods, Non-linear non-Gaussian state space model, Maximum *a posteriori* estimation, Particle filter, Smoothing.

1 Introduction

Let $t \in \mathbb{N}^*$ be a discrete time index. Consider the standard Markovian state-space model

$$x_t \sim f\left(x_t | x_{t-1}\right) \qquad \textit{State evolution density} \tag{1}$$
$$y_t \sim g\left(y_t | x_t\right) \qquad \textit{Observation density} \tag{2}$$

where $x_t \in \mathbb{R}^{n_x}$ are unobserved states of the system and $y_t \in \mathbb{R}^{n_y}$ are observations made over some time interval. $f(.|.)$ and $g(.|.)$ are pre-specified state evolution and observation densities which may be non-Gaussian and involve non-linearity. We assume that both $f(.|.)$ and $g(.|.)$ can be evaluated pointwise up to a normalizing constant for any states and observations x_t and y_t. The marginal distribution of the initial states is denoted by $f(x_1)$ and where convenient we will adopt the notation $f(x_1 | x_0) \triangleq f(x_1)$.

$x_{1:t} \triangleq (x_1, ..., x_t)$ and $y_{1:t} \triangleq (y_1, ..., y_t)$ denote collections of observations and states from time 1 through t. Given $y_{1:t}$, all inference on the states $x_{1:t}$ is based on the joint posterior distribution $p(x_{1:t} | y_{1:t})$. The Markov assumptions lead to the following expression for the joint distribution of states and observations by the probability chain rule

$$p\left(x_{1:t} | y_{1:t}\right) \propto \prod_{i=1}^{t} f(x_i|x_{i-1}) g(y_i|x_i) \tag{3}$$

One can obtain easily a recursion for this joint distribution

$$p\left(x_{1:t+1} | y_{1:t+1}\right) = p\left(x_{1:t} | y_{1:t}\right) \frac{g\left(y_{t+1} | x_{t+1}\right) f\left(x_{t+1} | x_t\right)}{p\left(y_{t+1} | y_t\right)} \tag{4}$$

In practice, computing (4) can only be performed in closed form for linear Gaussian models using the Kalman filter-smoother and for finite state-space hidden Markov models. In other cases approximate numerical techniques must be employed, such as the extended Kalman filter, Gaussian sum methods and general numerical integration procedures (Kitagawa, 1987). Here we focus on Monte Carlo *particle filters* (Doucet, De Freitas and Gordon, 2000a; Doucet, Godsill and Andrieu, 2000b; Gordon, Salmond and Smith, 1993; Kitagawa, 1996; Liu and Chen, 1998). These particle filters can be viewed as a randomized adaptive grid approximation where the particles (values of the grid) evolve randomly in time according to a simulation-based rule. One makes at time t the following approximation

$$p(x_{1:t}|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{1:t}^{(i)}}(dx_{1:t}) \tag{5}$$

where $\delta_{x_0}(dx)$ denotes the Dirac delta function located at x_0 and $w_t^{(i)}$ is the weight attached to particle $x_{1:t}^{(i)}$, $w_t^{(i)} \geq 0$ and $\sum_{i=1}^{N} w_t^{(i)} = 1$. Particles at time t can be updated efficiently to particles at time $t+1$ using sequential importance sampling and resampling methods, see Doucet *et al.* (2000b) for a review of the current methodology.

From (5), one can in principle estimate any feature of interest such as the filtering distribution $p(x_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}(dx_t)$, the minimum mean square error (MMSE) state estimate $\mathbb{E}[\, x_t|\, y_{1:t}] \approx \sum_{i=1}^{N} w_t^{(i)} x_t^{(i)}$, the fixed-interval smoothing distribution $p(x_k|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta_{x_k^{(i)}}(dx_k)$ etc. However, in practice the use of a resampling procedure is such that estimates which involve any significant element of smoothing (that is, anything except estimates based on the approximation of the marginal distribution $p(x_t|y_{1:t})$) will suffer from a severe depletion of samples over time and be unreliable, since in the approximation (5) there are only a few distinct paths $x_{1:t-k}^{(i)}$ for $k \ll t$ as these paths have been resampled many times (Doucet *et al.*, 2000b). In the case of fixed-interval smoothing, this problem has led researchers to develop alternative methods based either on the two-filter formula (Kitagawa, 1996) or the forward filtering-backward smoothing formula (Hürzeler and Künsch, 2000; Doucet *et al.*, 2000b).

In this paper, we focus on the estimation of the MAP sequence

$$x_{1:t}^{MAP}(t) \triangleq \arg\max_{x_{1:t}} p(x_{1:t}|y_{1:t}) \tag{6}$$

and the marginal fixed-lag MAP sequence

$$x_{t-L+1:t}^{MMAP}(t) \triangleq \arg\max_{x_{t-L+1:t}} p(x_{t-L+1:t}|y_{1:t}). \tag{7}$$

where the dependency upon $y_{1:t}$ is indicated by '(t)'.

In this paper we do not make any general claim that MAP estimation is preferable to MMSE or other estimators. Rather, the choice of estimator will be determined by the demands of the application. However, we do note that while the MMSE estimate is more popular in the statistical literature, it will not always make good sense. In some cases, for example, the posterior distribution might be multimodal and the MMSE estimate located between the modes, possibly in a region of very low probability. This is quite commonly the case when tracking multiple targets (Bar-Shalom and Li, 1995) and for deconvolution of impulsive processes (Mendel, 1990). In such cases, it is more useful to be able to estimate sequentially in time the MAP sequence estimate $x_{1:t}^{MAP}(t)$ given by (6) or $x_{t-L+1:t}^{MMAP}(t)$ given by (7). Moreover, in many target tracking problems, a zero-one loss function is appropriate, in which case there is no choice but to perform MAP estimation. It may be argued that marginal filtering or smoothing densities provide an adequate analysis of the data for most purposes. We maintain, however, that for many applications it is important to capture the sequence-specific interactions of the states over time in order to make successful inferences.

Except in a few special cases, estimating $x_{1:t}^{MAP}(t)$ or $x_{t-L+1:t}^{MMAP}(t)$ does not admit any analytical solution. In Section 2, we review standard methods to compute these estimates and then describe a method based on dynamic programming. In Section 3, we apply this algorithm to a standard non-linear time series and to time-varying autoregressions.

2 Maximum a Posteriori sequence estimation

We first focus on the estimation of $x_{1:t}^{MAP}(t)$ and discuss subsequently how to estimate $x_{t-L+1:t}^{MMAP}(t)$. Computing $x_{1:t}^{MAP}(t)$ requires the solution of a complex global optimization problem. Many standard global optimization methods such as simulated annealing or genetic algorithms are available in the literature and could be applied to this problem. However, most of these methods cannot readily be adapted to sequential estimation of $x_{1:t}^{MAP}(t)$ as t increases. We describe here several stochastic methods for performing this sequential task.

2.1 Standard methods

A simple sequential optimization method consists of sampling (sequentially in time) some paths $x_{1:t}^{(i)}$, $i = 1, \ldots, N$ according to a distribution, say $q(x_{1:t})$. Then one can select $\widehat{x}_{1:t}^{MAP}(t) = \arg\max_{x_{1:t}^{(i)}} p\left(x_{1:t}^{(i)} \middle| y_{1:t} \right)$. As long as the support of $q(x_{1:t})$ includes the support of $p\left(x_{1:t}^{(i)} \middle| y_{1:t} \right)$ then this estimate converges asymptotically ($N \rightarrow \infty$) to $x_{1:t}^{MAP}(t)$. However, the choice of $q(x_{1:t})$ will have a huge influence on the performance of the algorithm. The construction of an "optimal" distribution $q(x_{1:t})$ is clearly very difficult. At a given time t, it would consist of a distribution whose support is concentrated on the set of the unknown global maxima of $p(x_{1:t}| y_{1:t})$.

A reasonable choice for $q(x_{1:t})$ is the posterior distribution $p(x_{1:t}|y_{1:t})$ or any distribution $q(x_{1:t})$ that has the same global maxima as $p(x_{1:t}|y_{1:t})$, such as $q(x_{1:t}) \propto p(x_{1:t}|y_{1:t})^{\gamma}$ ($\gamma > 0$). Direct sampling from $p(x_{1:t}|y_{1:t})$ is usually impossible but based on the particle filtering approximation (5), one obtains the following approximation of $\widehat{x}_{1:T}^{MAP}(t)$:

$$\widehat{x}_{1:t}^{MAP}(t) = \arg \max_{x_{1:t} \in \{x_{1:t}^{(i)}; i=1,\ldots,N\}} p(x_{1:t}|y_{1:t}). \tag{8}$$

A clear advantage of this method is that it is very easy to implement and has computational complexity and storage requirements of order $O(NT)$, but a severe drawback is that, because of the *degeneracy phenomenon* discussed in the introduction, the performance of this estimate will get worse as time

t increases. Similar problems would occur while estimating $x_{t-L+1:t}^{MMAP}(t)$ as soon as L is large. It is possible to derive other sampling schemes focusing on the regions of high posterior values using ideas from the genetic algorithms literature (Higuchi, 1997) but we do not pursue this here.

Another way to consider the problems with basic procedures such as the above is that the trajectories compared are limited to be those which were generated directly by the Monte Carlo filter. These by their very nature will generally be far more random than the true MAP sequence. Hence a huge number of trajectories is required for reasonable performance, especially for large datasets.

2.2 Optimization via dynamic programming.

We now describe our dynamic programming approach to MAP estimation. Assume the filtering distribution $p(x_k|y_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{x_k^{(i)}}(dx_k)$ has been computed and stored at each time $k = 1, \ldots, t$. The filtering procedure has thus generated a finite grid approximation of the state space at each time k, that is $x_k \in \{x_k^{(i)}; i = 1, \ldots, N\}$. Though this grid is not optimal for MAP estimation, it is a sensible discretization of the state-space as long as the weights have a small variance, a basic requirement of any particle filtering algorithm.

2.2.1 Maximization of $p\left(x_{1:t}\right| y_{1:t})$

An approximation to $x_{1:t}^{MAP}(t)$ can be obtained as

$$\widehat{x}_{1:t}^{MAP}(t) = \underset{x_{1:t} \in \bigotimes_{k=1}^{t} \{x_k^{(i)};\, i=1,2,\ldots,N\}}{\arg\max} p\left(x_{1:t}\right| y_{1:t}) .$$

A brute-force evaluation of this MAP state sequence estimate would involve an exhaustive search of all possible state trajectories in this discrete state space model. However, the function to maximize is additive as, thanks to (3), one has

$$x_{1:t}^{MAP}(t) \triangleq \underset{x_{1:t}}{\arg\max} \sum_{k=1}^{t} \left[\log g\left(y_k\right| x_k) + \log f(x_k|x_{k-1})\right]$$

This property allows us to use a standard dynamic programming (DP) technique, the Viterbi algorithm (Viterbi, 1967), so as to compute

$$\widehat{x}_{1:t}^{MAP}(t) = \underset{x_{1:t} \in \bigotimes_{k=1}^{t} \{x_k^{(i)};\, i=1,2,\ldots,N\}}{\arg\max} \sum_{k=1}^{t} \left[\log g\left(y_k\right| x_k) + \log f(x_k|x_{k-1})\right] . \tag{9}$$

The Viterbi algorithm is a well known technique for the estimation of discrete state-space hidden Markov models, and has been particularly associated with speech recognition (Levinson, Rabiner and Sondhi, 1983) and decoding of convolutional codes in information theory (Forney, 1973). The originality of our work lies in the observation that the Viterbi algorithm can be employed for estimation of a *continuous* state-space Markov model via a discrete approximation of the state space using particle filters. Furthermore, in contrast with standard applications of the Viterbi algorithm, the discretization of the state-space is time-dependent and generated automatically using any particle filtering method.

The algorithm proceeds as follows for computation of

$$\widehat{x}_{1:t}^{MAP}(t) \triangleq (\widehat{x}_1^{MAP}(t), \widehat{x}_2^{MAP}(t), \ldots, \widehat{x}_t^{MAP}(t))^T.$$

Viterbi Algorithm

1. *Initialization*. For $1 \leq i \leq N$

$$\delta_1(i) = \log f\left(x_1^{(i)}\right) + \log g\left(y_1 | x_1^{(i)}\right)$$

2. *Recursion*. For $2 \leq k \leq t$ and $1 \leq j \leq N$

$$\delta_k(j) = \log g\left(y_k | x_k^{(j)}\right) + \max_i \left[\delta_{k-1}(i) + \log f\left(x_k^{(j)} \middle| x_{k-1}^{(i)}\right)\right]$$

$$\psi_k(j) = \arg\max_i \left[\delta_{k-1}(i) + \log f\left(x_k^{(j)} \middle| x_{k-1}^{(i)}\right)\right]$$

3. *Termination*.

$$i_t = \arg\max_i \delta_t(i)$$

$$\widehat{x}_t^{MAP}(t) = x_t^{(i_t)}$$

4. *Backtracking*. For $k = t-1, t-2, \ldots 1$

$$i_k = \psi_{k+1}(i_{k+1})$$

$$\widehat{x}_k^{MAP}(t) = x_k^{(i_k)}$$

This algorithm has a computational complexity $O\left(N^2 t\right)$ and memory requirements of order $O(Nt)$ as it requires storage at each time $k \in \{1, \ldots, t\}$ of the particle filter approximation $\left(\left(w_k^{(i)}, x_k^{(i)}\right); i = 1, \ldots, N\right)$. Now assume a new data point y_{t+1} is available, then one only needs to compute

$\delta_{t+1}(j)$ and $\psi_{t+1}(j)$. If one is interested in estimating only $x_{t-p+2:t+1}^{MAP}(t+1)$ and not the whole path $x_{1:t+1}^{MAP}(t+1)$, then the memory requirements are only of order $O(Np)$ and the computational complexity of order $O(N^2)$ (computation of $\delta_{t+1}(j)$ and $\psi_{t+1}(j)$) $+O(p)$ (backtracking)): one can discard the past history of the simulated paths. In this case the storage requirements of the algorithm do not increase over time.

An important point to note is that the general procedure here is guaranteed to give an estimate with at least as high posterior probability as the standard method, when applied to the same particle set. This is as a result of the optimality of the Viterbi algorithm which exactly solves the discrete state problem (9). In practice we expect it to significantly out-perform the standard methods, especially for large datasets; see examples later.

2.2.2 Maximization of $p(x_{t-L+1:t}|\, y_{1:t})$

The algorithm to estimate $x_{1:t}^{MAP}(t)$ can be easily modified to estimate $x_{t-L+1:t}^{MMAP}(t)$. Obtaining $x_{t-L+1:t}^{MMAP}(t)$ requires maximization of

$$\log p(x_{t-L+1}|\, y_{1:t-L}) + \log g(y_{t-L+1}|\, x_{t-L+1}) + \sum_{k=t-L+2}^{t} [\log g(y_k|\, x_k) + \log f(x_k|x_{k-1})]$$

where the initial marginal $p(x_{t-L+1}|\, y_{1:t-L})$ can be computed pointwise through the particle filtering approximation of $p(x_{t-L}|\, y_{1:t-L})$ via

$$p(x_{t-L+1}|\, y_{1:t-L}) \approx \sum_{i=1}^{N} w_{t-L}^{(i)} f\left(x_{t-L+1}|\, x_{t-L}^{(i)}\right).$$

The algorithm then proceeds exactly as before, but starting at time $t-L+1$ and replacing the initial state distribution with $p(x_{t-L+1}|\, y_{1:t-L})$. This algorithm has a computational complexity $O\left(N^2(L+1)\right)$ and memory requirements of order $O(N(L+1))$: it requires storage at each time $k=t-L+1$ to t the approximation of the filtering density $\left(\left(w_k^{(i)}, x_k^{(i)}\right); i=1,\ldots,N\right)$.

3 Examples

3.1 A non-linear time series

We consider here the following non-linear reference model (Doucet *et al.*, 2000b; Gordon *et al.*, 1993; Kitagawa, 1996)

$$x_t = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + v_t$$

$$y_t = \frac{x_t^2}{20} + w_t$$

where $x_1 \sim \mathcal{N}\left(0, \sigma_1^2\right)$, v_t and w_t are mutually independent white Gaussian noise sequences with $v_k \sim \mathcal{N}\left(0, \sigma_v^2\right)$ and $w_k \sim \mathcal{N}\left(0, \sigma_w^2\right)$ where $\sigma_1^2 = 5$, $\sigma_v^2 = 10$ and $\sigma_w^2 = 1$.

In Figure 1, we present the simulated state sequence x_t and the observations y_t. As illustrated in Figure 2 and Figure 3, because the observation equation is a function of x_t^2, the filtering distribution $p\left(x_t \middle| y_{1:t}\right)$ is often bimodal so that the MMSE estimate is located between two modes.

Figure 1 about here.
Figure 2 about here.

Figure 3 about here.

In Figure 4, we illustrate the differences between the MMSE and the MAP sequence estimate obtained using the Viterbi algorithm with $N = 1000$ particles. At time $t = 14$, the MMSE estimate averages the two modes whereas the MAP value corresponds approximately to a mode of $p\left(x_t \middle| y_{1:t}\right)$, which is closer to the true value. We do not however claim that MAP estimation is always the point estimate to use, rather that in some settings such as this the MAP estimate may be more relevant than other estimators such as the MMSE estimator.

Figure 4 about here.

As the computational complexity of the standard estimate (8) described in 2.1 and of the Viterbi algorithms described in 2.2 are significantly different, it is of interest to compare for an approximately fixed computational complexity the performance of the estimates. That is, when we ran the Viterbi algorithm for N particles, we ran the standard methods with N^2 particles. We present in Table 1 the average, over 10 different realizations of the data, of the log-posterior probability values of the MAP estimate. In Table 2 the sample variation of the procedure is tested by performing 25 particle filtering and smoothing estimations on one single realisation of the data and measuring the mean and standard deviations of the estimated log-posterior probability. It was impractical to perform the standard procedure for $N \geq 500$ (that is $N^2 \geq 2.5.10^5$) as the memory requirements $O\left(N^2 T\right)$ were too great for a standard computer, and these cases are marked with crosses.

Table 1 about here.

Table 2 about here.

It is clear that the Viterbi algorithm outperforms the standard method and that the robustness in terms of sample variability improves as the number of particles increases. Because of the degeneracy phenomenon inherent in the standard method, this improvement over the standard methods will get larger and larger as t increases.

3.2 Time-varying autoregressive models

Our second example of MAP sequence estimation is for the time-varying autoregressive (TVAR) model (see for example Kitagawa and Gersch (1996) and Prado, West and Krystal (1999)), which can be used to model parametrically a signal with time-varying frequency content. These models are of very wide utility and importance in engineering, scientific and socio-economic applications. It is assumed that the TVAR signal is observed in additive white Gaussian noise, which models any background noise present in the measurement environment. This aspect of the model is important in many applications where measurements are noisy, including the field of speech and audio processing (Godsill and Rayner, 1998a; Godsill and Rayner, 1998b; Godsill, 1997; Vermaak, Andrieu, Doucet and Godsill, 1999). The TVAR signal process $\{z_t\}$ is generated as a linear weighted sum of previous signal values:

$$z_t = \sum_{i=1}^{P} a_{i,t} z_{t-i} + e_t = a_t^{\mathsf{T}} z_{t-1:t-P} + e_t.$$

Here $a_t = (a_{1,t}, a_{2,t}, ..., a_{P,t})^{\mathsf{T}}$ is the $P-$dimensional AR coefficient vector at time t. The innovation sequence $\{e_t\}$ is assumed independent and Gaussian with time-varying variance $\sigma_{e_t}^2$. Hence we may write the conditional density for z_t as:

$$f(z_t | z_{t-1:t-P}, a_t, \sigma_{e_t}^2) = \mathcal{N}\left(a_t^{\mathsf{T}} z_{t-1:t-P}, \sigma_{e_t}^2\right).$$

The signal is assumed to be observed in independent Gaussian noise, i.e.,

$$y_t = z_t + v_t,$$

so that we may write the density for the observation y_t as $g(y_t | z_t, \sigma_v) = \mathcal{N}\left(z_t, \sigma_v^2\right)$. For our simulations, a Gaussian random walk model is assumed for the log-standard deviation $\phi_{e_t} = \log(\sigma_{e_t})$, i.e.,

$$f(\phi_{e_t} | \phi_{e_{t-1}}, \sigma_{\phi_e}^2) = \mathcal{N}\left(\phi_{e_{t-1}}, \sigma_{\phi_e}^2\right).$$

The model now requires specification of the time variation in a_t itself. One of the simplest choices of all is a first order autoregression directly on the coefficients

$$f(a_t | a_{t-1}, \sigma_a^2) = \mathcal{N}\left(\alpha a_{t-1}, \sigma_a^2 I_P\right)$$

where α is a constant just less than 1.

More elaborate schemes of this sort are possible, such as a smoothed random walk involving AR coefficients from further in the past, or a non-diagonal covariance matrix, but none of these modifications make a significant difference to the computations required for the particle filter. In this paper we do not consider the stability of the TVAR model; see Doucet, Godsill and West (2000c) and Godsill, Doucet and West (2000) for models which explicitly deal with this aspect.

The state space TVAR model is now fully specified. The unobserved state vector is $x_t = (z_{t:t-P+1}, a_t, \phi_{e_t})$. Hyperparameters σ_v^2, σ_a^2 and $\sigma_{\phi_e}^2$ are assumed pre-specified and fixed in all of the simulations. The initial state probability is specified as Gaussian.

3.2.1 Filtering and MAP estimation

The first step in analysing the data is to perform a complete forwards sweep of a Monte Carlo filtering algorithm to produce weighted particles $\left(\left(w_t^{(i)}, x_t^{(i)}\right); i = 1, \ldots, N\right)$ for $t = 1, 2, ...T$, drawn approximately according to $p(x_t|y_{1:t})$. Filtering is carried out using a version of the *auxiliary particle filter* (Pitt and Shephard, 1999). We do not describe all of the details here as the aim of this paper is to develop new smoothing methodology. However, full details of the filter used and a study of some alternatives applied to the TVAR model can be found in Godsill and Clapp (2000). Following the filtering pass, the MAP state sequence is estimated using the dynamic programming method described above and compared with the standard scheme.

3.2.2 Results

Results are presented initially for a simulated TVAR model with order $P = 3$. The fixed hyperparameters are $\sigma_v = 1$, $\sigma_a = 0.01$ and $\sigma_{\phi_e} = 0.01$. The initial states are assigned independent, diffuse Gaussian priors with zero mean. $T = 50$ data points are generated synthetically from the TVAR model as described above and filtered with $N = 500$ particles. A typical result showing the estimated MAP AR parameters is plotted in Figure 5, comparing the proposed dynamic programming method with the true parameters which generated the data. The log-posterior probability of this estimate was 1290, compared with 819 for the sequence estimated by the standard scheme – a hugely significant improvement. Finally, a Monte Carlo experiment was carried out in which 100 independent realizations of AR processes with the same hyperparameters were estimated. In every case there was a large and sometimes a dramatic improvement in posterior probability from using the dynamic programming method as compared with the standard method.

Figure 5 about here

We now apply the methodology to estimation of the same TVAR model in a real speech signal observed in white Gaussian background noise. The noisy data can be seen in Figure 6. The application of these models in this field is useful both for performing signal extraction from noise and also for examination of the underlying time-varying spectral structure of the data, which may then be used to aid speech recognition, speaker identification or for more general scientific study of the speech generation process. A TVAR model with order $P = 4$ is chosen, in accordance with empirical performance evaluation using a variety of model orders (see e.g. Vermaak *et al.* (1999) for more detail). The fixed hyperparameters are $\sigma_v = 0.01$, $\sigma_a = 0.005$ and $\sigma_{\phi_e} = 0.001$, chosen to match the known background noise statistics and expected characteristics of speech signals. $T = 800$ data points are filtered using $N = 1000$ particles and the final 100 states are estimated using the fixed-lag MAP procedure outlined in section 2.2.2 (in this way we aim to ensure that the effects of the Gaussian prior on the initial states at $t = 1$ are negligible). Figure 7 shows the noisy speech data over the region for MAP estimation. Figure 8 shows the signal sequence extracted from the estimated MAP state sequence, demonstrating a good fit to the data and a degree of smoothness typical of speech signals. Figure 9 shows the estimated TVAR coefficient sequence, which is slowly varying in this region of fairly steady signal characteristics. In order to test the robustness of the procedure once again, 50 repetitions of the entire estimation procedure were carried out on this same piece of data. The Viterbi-based method achieved a significant improvement, both in terms of mean probability and variability, compared with the standard method, see Table 3. Note that the number of particles used for both methods here was 1000, as the memory requirements of the standard method become prohibitive for much larger numbers, as discussed earlier.

Figures 6-9 about here

Table 3 about here

4 Discussion

Recent years have seen a huge surge of interest in particle filtering, motivated by practical problems of sequential analysis in dynamic models in many areas of engineering, the natural science and socio-economics (Doucet *et al.*, 2000a). Our work here is not specific to any one algorithm, and takes the established notion of sequentially updated particulate representations of posterior distributions in state space models as the starting point for smoothing. In speech processing as in other applications, it is often critically important to "look back over time" for several or many time steps in order to assess and evaluate how new data revises the view of the recent past. Hence smoothing algorithms are key, and our work here develops effective approaches that apply whatever filtering method is adopted.

We have developed and presented fairly simple methods for generation of MAP estimates of joint smoothing densities in a general model context. Smoothing has not been stressed by earlier authors in the sequential simulation literature, and where it has been studied approaches have been limited to approximating the time-specific marginal smoothing distributions for individual states. We forcefully argue that patterns of changes in historical states should focus on the joint trajectories of past states and hence necessarily involve consideration of joint smoothing densities. Here we have presented an advance in this direction by showing how to make MAP estimates of entire state trajectories. As already discussed and illustrated with the non-linear model example, the MAP sequence estimate is likely to be a valuable quantity in inference for models with strongly multimodal characteristics. Another important challenge for many statistical contexts is in generating sampled realizations from the joint smoothing density; this is an area which we have also explored within a particle simulation framework (Doucet *et al.*, 2000c; Godsill *et al.*, 2000).

There are current research challenges in many aspects of the sequential simulation arena, including real needs for improved particle filtering algorithms, and reconciliation of the several variants of sequential importance sampling, resampling, and auxiliary particle methods. The current paper ignores issues of learning on fixed model parameters in addition to time-varying states, a broader problem also ignored by most other authors in the field, but critical in many applications (e.g., as in the challenging multifactor models of Aguilar and West (2000)). In our current work with TVAR models we are developing analyses for both parameters and states using the *auxiliary particle plus* methods of Liu and West (2000). It should be noted that the MAP smoothing method developed and illustrated here applies directly in this context also.

5 Acknowledgements

This work was performed under partial support of EPSRC grant 'Dynamic Sequential simulation methodology' (UK) and NSF grant DMS-9704432 (USA).

References

AGUILAR, O. AND WEST, M. (2000). Bayesian dynamic factor models and variance matrix discounting for portfolio allocation. *Journal of Business and Economic Statistics* **18**.

BAR-SHALOM, Y. AND LI, X. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*. New York: Academic Press.

DOUCET, A., DE FREITAS, J. F. G. AND GORDON, N. J. (eds.) (2000a). *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag.

DOUCET, A., GODSILL, S. J. AND ANDRIEU, C. (2000b). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* **10** 197–208.

DOUCET, A., GODSILL, S. J. AND WEST, M. (2000c). Monte Carlo filtering and smoothing with application to time-varying spectral estimation. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* vol. II 701–704. ISBN 0-7803-6296-9.

FORNEY, G. D. (1973). The Viterbi algorithm. *Proc. IEEE* **61**(3) 268–278.

GODSILL, S. J. (1997). Bayesian enhancement of speech and audio signals which can be modelled as ARMA processes. *International Statistical Review* **65**(1) 1–21.

GODSILL, S. J. AND CLAPP, T. C. (2000). Improvement strategies for Monte Carlo particle filters. In A. Doucet, J. F. G. De Freitas and N. J. Gordon (eds.), *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag.

GODSILL, S. J., DOUCET, A. AND WEST, M. (2000). Methodology for Monte Carlo smoothing with application to time-varying autoregressions. *Biometrika* Submitted for publication.

GODSILL, S. J. AND RAYNER, P. J. W. (1998a). *Digital Audio Restoration: A Statistical Model-Based Approach*. Berlin: Springer, ISBN 3 540 76222 1.

Godsill, S. J. and Rayner, P. J. W. (1998b). Robust reconstruction and analysis of autoregressive signals in impulsive noise using the Gibbs sampler. *IEEE Trans. on Speech and Audio Processing* **6**(4) 352–372.

Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F* **140**(2) 107–113.

Higuchi, T. (1997). Monte Carlo filtering using the genetic algorithm operators. *Jounal of Statistical Computation and Simulation* **59** 1–23.

Hürzeler, M. and Künsch, H. R. (2000). Monte Carlo approximations for general state space models. *Journal of Computational and Graphical Statistics* **7**(2) 175–193.

Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association* **82**(400) 1032–1063.

Kitagawa, G. (1996). Monte Carlo filter and smoother for nonlinear non-Gaussian state space models. *Journal of Computational and Graphical Statistics* **5** 1–25.

Kitagawa, G. and Gersch, W. (1996). *Smoothness Priors Analysis of Time Series, Lecture Notes in Statistics #116*. Springer-Verlag New York.

Levinson, S. E., Rabiner, L. R. and Sondhi, M. M. (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic spech recognition. *Bell System Technical Journal* **62**(4) 1035–1075.

Liu, J. and West, M. (2000). Combined parameter and state estimation in simulation-based filtering. In A. Doucet, J. F. G. De Freitas and N. J. Gordon (eds.), *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag New York.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamical systems. *Journal of American Statistical Association* **93** 1032–44.

Mendel, J. (1990). *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing*. New York: Springer-Verlag.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: auxiliary particle filters. *Journal of American Statistical Association* **94** 590–9.

Prado, R., West, M. and Krystal, A. (1999). Evaluation and comparison of EEG traces: Latent structure in non-stationary time series. *Journal of American Statistical Association* **94** 1083–1095.

Vermaak, J., Andrieu, C., Doucet, A. and Godsill, S. J. (1999). On-line Bayesian modelling and enhancement of speech signals. Tech. Rep. CUED/F-INFENG/TR.361 Cambridge University Engineering Department Cambridge, England.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Info. Theory* **IT-13** 260–269.

6 Figures and Tables

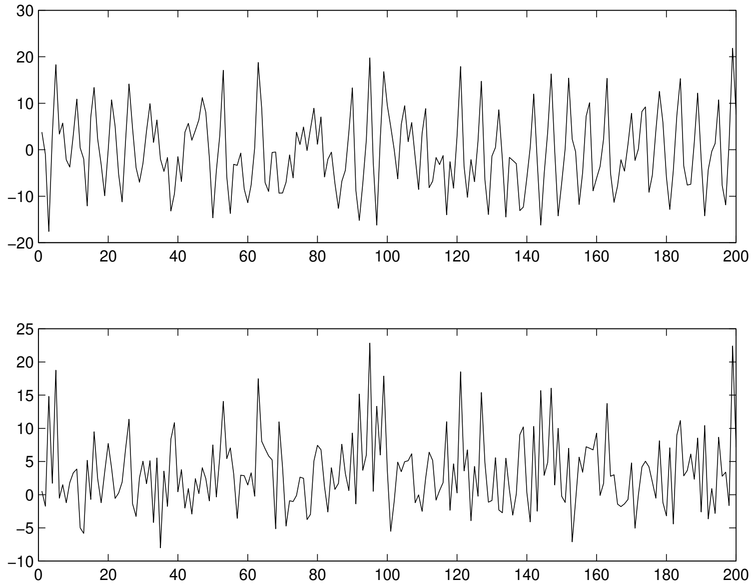

Figure 1: Simulated signal (top) and observations (bottom).

Particles	μ (Viterbi)	μ (standard method)
$N = 100$	-79.2	-84.3
$N = 250$	-77.3	-82.1
$N = 500$	-75.2	xxx
$N = 1000$	-74.9	xxx

Table 1: Mean log-posterior values of the MAP estimate over 10 data realizations.

Particles	μ (Viterbi)	μ (standard method)	σ (Viterbi)	σ (standard method)
$N = 100$	-81.09	-85.36	1.14	3.98
$N = 250$	-76.67	-82.24	0.58	3.09
$N = 500$	-75.42	xxx	0.42	xxx
$N = 1000$	-74.92	xxx	0.23	xxx

Table 2: Sample mean log-posterior values and standard deviations over 25 simulations with the same data.

Method	μ	σ
Viterbi	3180	11.1
Standard	2830	26.9

Table 3: Speech data: sample mean and standard deviation of the MAP probability over 50 simulations, $N = 1000$.

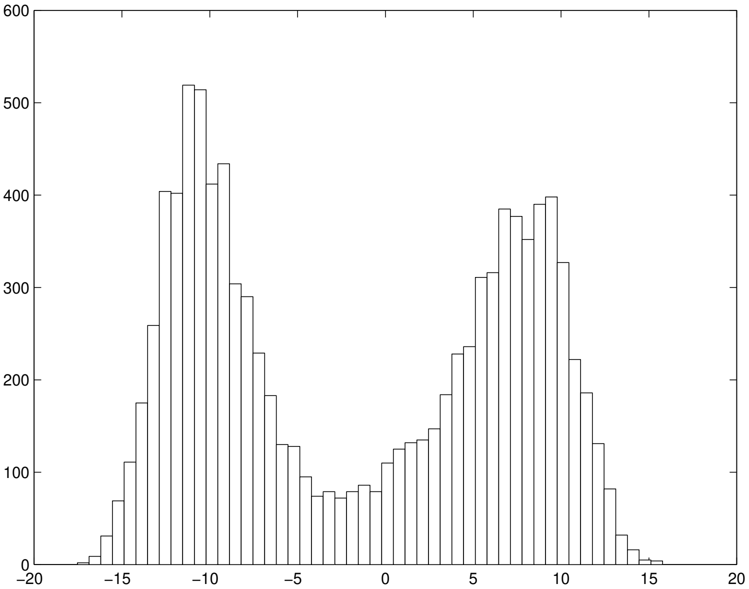

Figure 2: Filtering distribution $p\left(x_t \middle| y_{1:t}\right)$ at time $t = 14$.

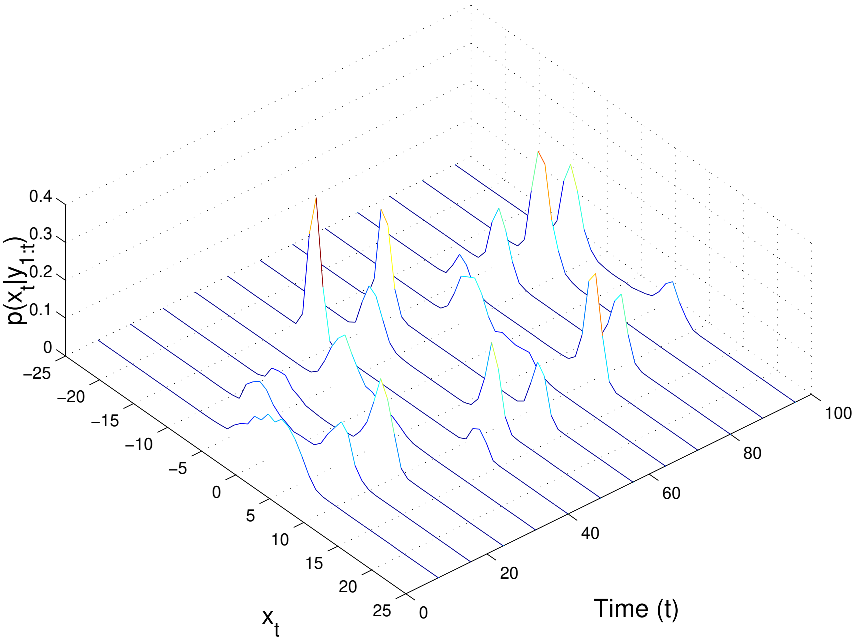

Figure 3: Evolution of the filtering distribution $p\left(x_t \middle| y_{1:t}\right)$ over time t.

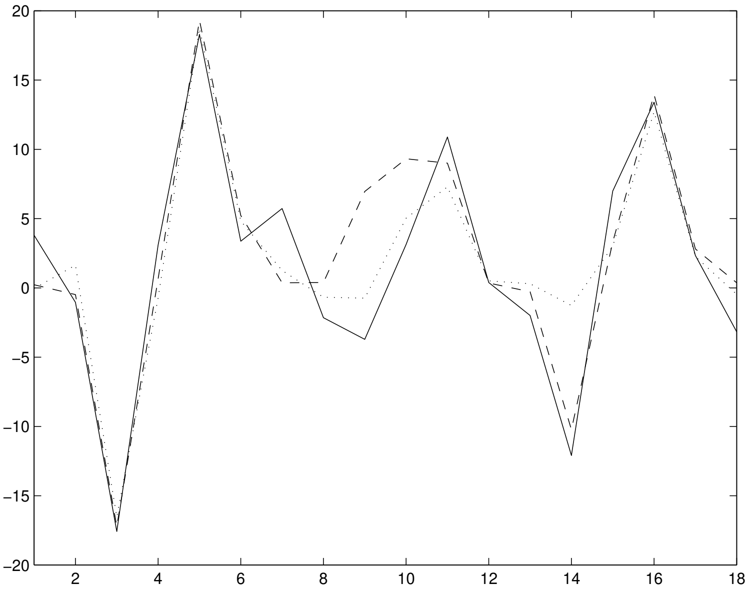

Figure 4: Simulated sequence x_t (solid line), MMSE estimate (dotted line), MAP sequence estimate (dashed line).

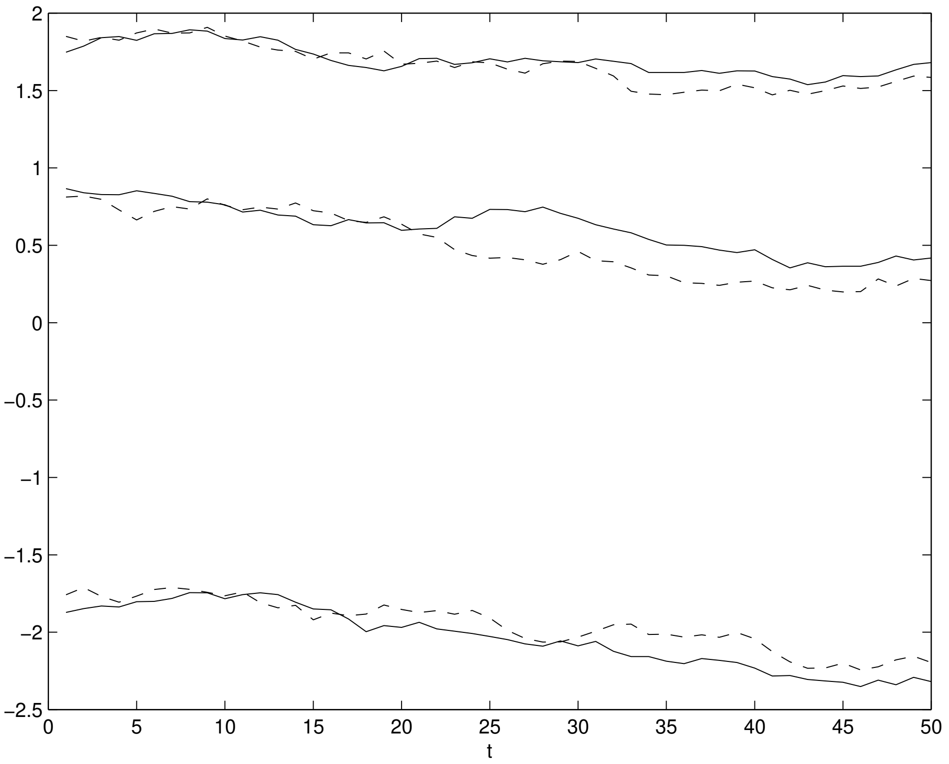

Figure 5: Typical results for a third order TVAR model - AR coefficients. True values (dotted) and estimated MAP sequence (solid)

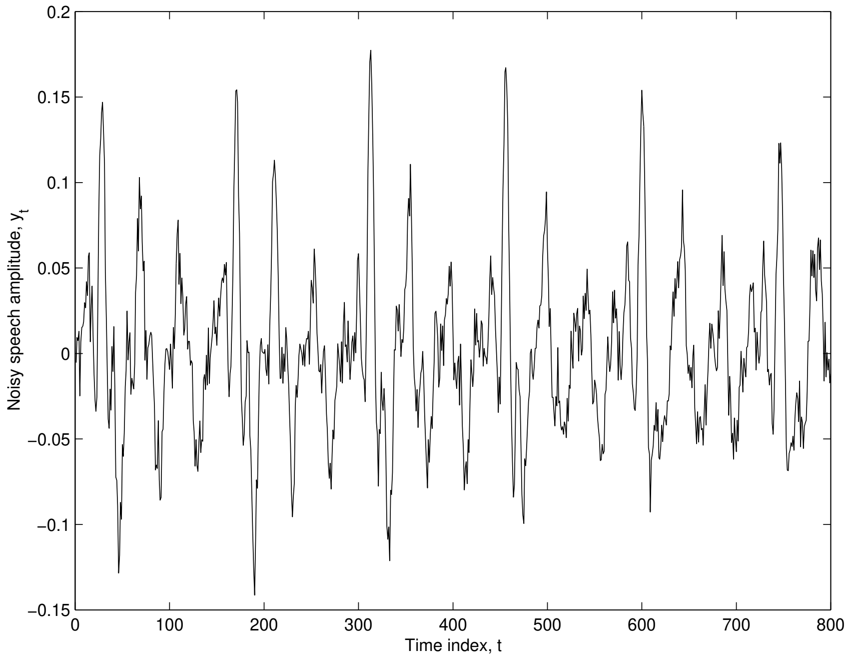

Figure 6: Noisy speech data. TIMIT database, male speaker, sampling rate 16kHz, resolution 16-bit.

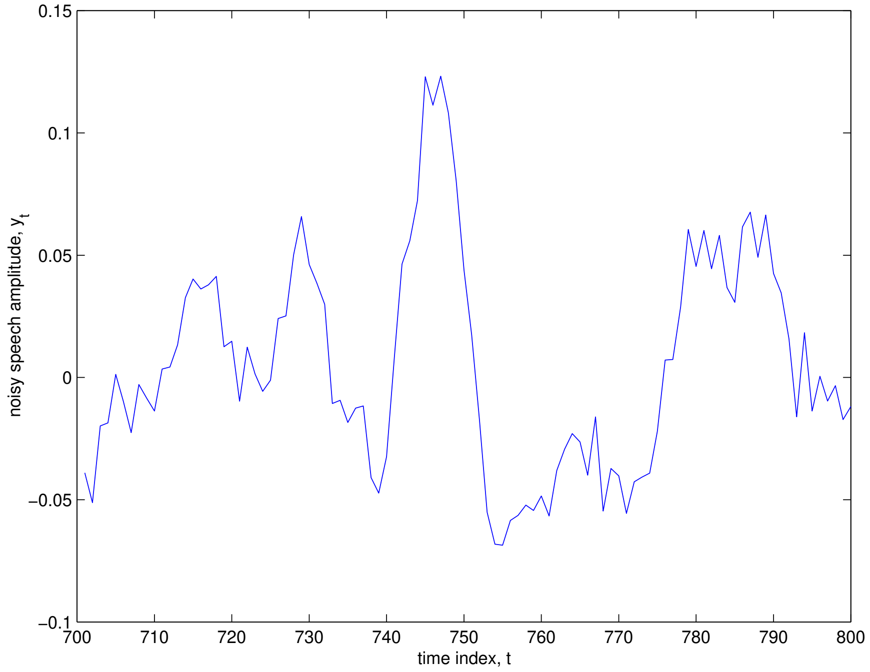

Figure 7: Noisy speech: data points 701,...,800.

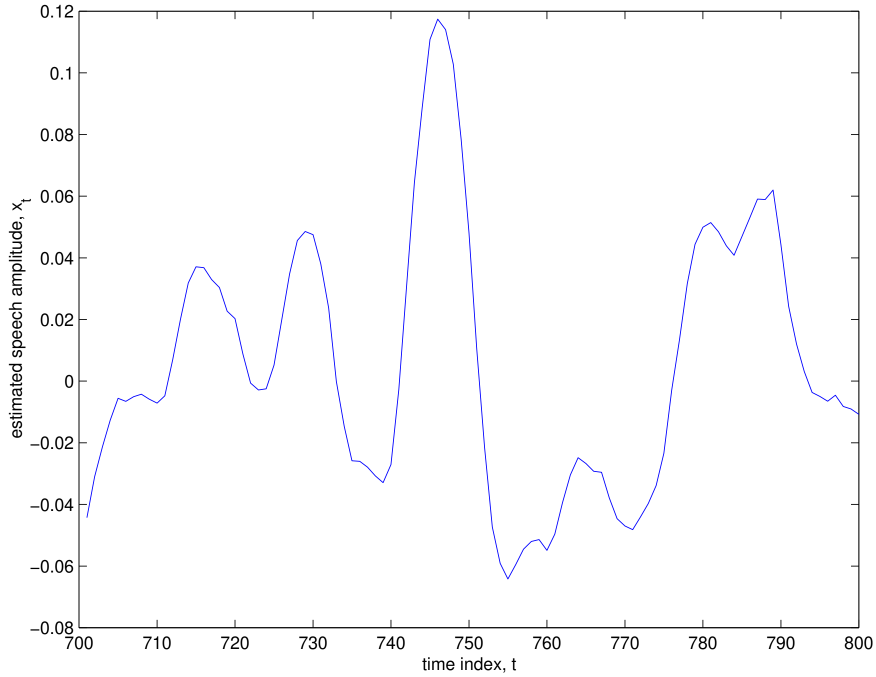

Figure 8: Estimated signal waveform x_t.

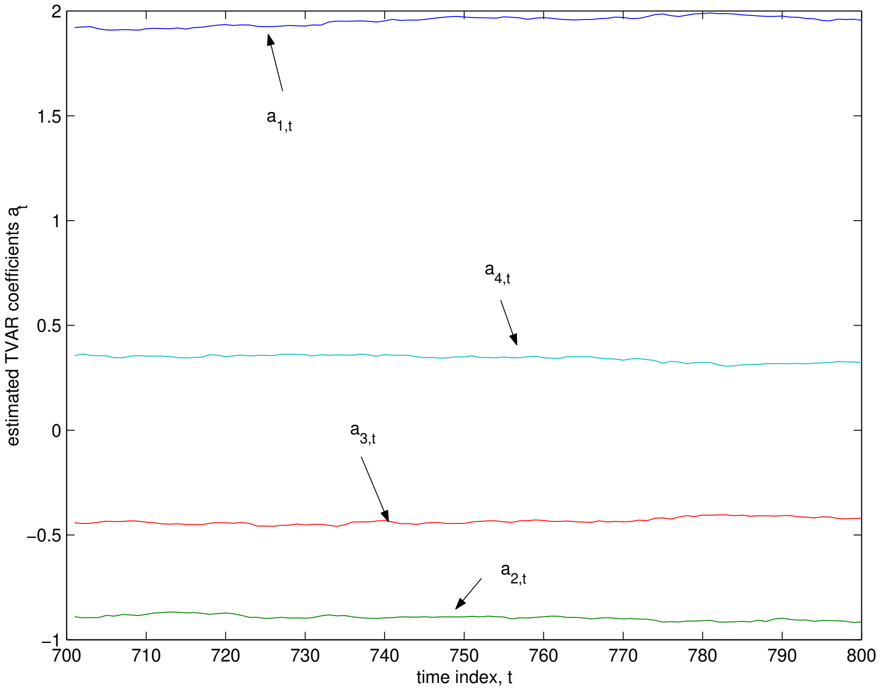

Figure 9: Estimated TVAR coefficients