Dipartimento di Statistica
"Giuseppe Parenti"

Dipartimento di Statistica "G. Parenti" – Viale Morgagni 59 – 50134 Firenze - www.ds.unifi.it

WORKING PAPER 2004/05

On-line Bayesian estimation of AR signals in symmetric alpha-stable noise

Marco J. Lombardi,
Simon J. Godsill

Università degli Studi
di Firenze

Statistics for the experimental and technological research

On-line Bayesian Estimation of AR Signals in Symmetric α-Stable Noise

Marco J. Lombardi* and Simon J. Godsill†

Abstract

In this paper we propose an on-line Bayesian filtering and smoothing method for time series models with heavy-tailed α-stable noise, with a particular focus on TVAR models. α-stable processes have been shown in the past to be a good model for many naturally occurring noise sources, see e.g. [1, 2]. We first point out how a filter that fails to take into account the heavy-tailed character of the noise performs poorly and then examine how an α-stable based particle filter can be devised to overcome this problem.

The filtering methodology is based on a scale mixtures of normals (SMiN) representation of the α-stable distribution, which allows efficient Rao-Blackwellised implementation within a conditionally Gaussian framework, and requires no direct evaluation of the α-stable density, which is in general unavailable in closed form. The methodology is shown to work well, outperforming the traditional Gaussian methods both on simulated data and on real audio data sets.

1 Introduction

In many real signal processing environments, sources of noise cannot be considered as Gaussian. Here we consider in particular the estimation of processes observed in heavy-tailed noise, i.e. noise having occasional very large values that could strongly affect any inference procedure for the underlying process. We focus on one particular class of heavy-tailed distribution, the *symmetric α-stable* class, that can be obtained through a generalisation

*Corresponding author. Dipartimento di statistica "G. Parenti", Università degli studi di Firenze – Viale G.B. Morgagni 59, I-50134 Firenze (Italy). Phone +39-055-4237256, Fax +39-055-4223560, E-mail: mjl@ds.unifi.it. This paper was completed when the first author was visiting the Signal Processing lab at the University of Cambridge; a preliminary version of it was presented there in a seminar. We are indebted with Jaco Vermaak for allowing us to exploit and modify his Matlab source code.

†Signal Processing Laboratory, Cambridge University Engineering Department – Trumpington street, CB2 1PZ Cambridge (U.K.). Phone +44-1223-332604, Fax +44-1223-332662, E-mail sjg@eng.cam.ac.uk.

of the central limit theorem when the random variables need not have finite variance, see [3, 4]. Such noise processes have been found appropriate in a number of areas, including signal processing [1, 2] and econometrics [5].

We propose methodology for optimal on-line estimation of stochastic processes observed in α-stable noise. The models chosen are time varying autoregressions (TVAR), which are appropriate for a wide range of signals, including speech, audio, eeg and seismic data. On-line estimation and signal extraction is performed by means of sequential Monte Carlo methods [6–8]. These methods have been applied to the TVAR model with Gaussian noise in [9–12]. This paper constitutes an extension to these approaches to the symmetric α-stable case and, more generally, to any case where the noise distribution can be represented as a scale mixture of normals (SMiN). The SMiN representation of the α-stable class allows us importantly to employ conditionally linear and Gaussian computational methods within the Monte Carlo filter (via the Kalman filter), hence avoiding any direct evaluations of the noise density function (which is unavailable in most cases for the α-stable model). It is important to stress that our proposed methods are simulation-exact, in the sense that the scale mixture of normals representation of the symmetric α-stable law is exact. The Monte Carlo schemes proposed require random variate generation from α-stable distributions – this can readily and exactly be achieved using standard procedures [13]. In [14, 15] methodology was presented for inference about static AR models in stable law noise, using a batch based Markov chain Monte Carlo (MCMC) method. In this paper we show how to extend the methodology to the sequential Monte Carlo setting. We will focus on TVAR models with α-stable noise. These models require extensions to Gaussian-based methods such as [9–12] to allow for non-Gaussianity, and also for estimation of the tail parameter α of the stable distribution. These extensions both present significant additional challenges, and in particular the estimation of α is a difficult problem for Monte Carlo filters to solve.

The methods are able accurately to reconstruct the signal process, the TVAR parameters, and also the stable law parameter α, which is static and thus not easily amenable to standard particle filter analysis. A real-data application of the methods is presented for audio signal enhancement. In this setting we present experimental evidence that the α-stable distribution is appropriate for certain noise sources in 78rpm gramophone disk recordings which are typically degraded by non-Gaussian clicks. Results are found to be very effective.

The structure of the paper is as follows: we first describe the properties of α-stable distributions and proposing that they are more appropriate than other parametric families in modelling noise in certain real-world signals. We then introduce the statistical model and its state-space representation. Bayesian methods are presented for sequential estimation and filtering and we discuss how symmetric α-stable distributions can be embedded into this framework. The approach is then compared to the traditional Gaussian framework on both simulated data and artificially corrupted real data samples. An application to real audio data concludes the paper.

2 α-Stable Distributions

The α-stable family of distributions arises from a more general version of the central limit theorem in which the assumption of finite variance is replaced by a much less restrictive one concerning the regular behavior of the tails [3]; the Gaussian distribution then becomes a particular case of α-stable distribution. This family of distributions has a very interesting pattern of shapes, allowing for asymmetry and thick tails, that makes it suitable for the modelling of several phenomena; moreover, linear combinations of independent α-stable variables are themselves α-stable.

The family is identified by means of the characteristic function

$$\phi(t) = \begin{cases} \exp\left\{\mathrm{i}\delta t - \gamma^{\alpha}|t|^{\alpha}\left[1 - \mathrm{i}\beta \operatorname{sgn}(t)\tan\frac{\pi\alpha}{2}\right]\right\} & \text{if } \alpha \neq 1 \\ \exp\left\{\mathrm{i}\delta t - \gamma|t|\left[1 + \mathrm{i}\beta\frac{2}{\pi}\operatorname{sgn}(t)\ln|t|\right]\right\} & \text{if } \alpha = 1 \end{cases} \tag{1}$$

which depends on four parameters: $\alpha \in (0, 2]$, measuring the tail thickness (thicker tails for smaller values of the parameter), $\beta \in [-1, 1]$ determining the degree and sign of asymmetry, $\gamma > 0$ (scale) and $\delta \in \mathbb{R}$ (location). To denote a stable distribution with parameters α, β, γ and δ we will use the notation $\mathcal{S}(\alpha, \beta, \gamma, \delta)$. As in the Gaussian case, a random variable X with $\mathcal{S}(\alpha, \beta, \gamma, \delta)$ distribution can be standardized to produce

$$Z = \frac{X - \delta}{\gamma} \sim \mathcal{S}(\alpha, \beta, 1, 0).$$

For the standardized stable distribution, we will henceforth use the shorthand notation $\mathcal{S}(\alpha, \beta)$.

Unfortunately, (1) can only be inverted to yield a closed-form density function in a very few cases: $\alpha = 2$, corresponding to the normal distribution, $\alpha = 1$ and $\beta = 0$, yielding the Cauchy distribution, and $\alpha = \frac{1}{2}, \beta = 1$

for the Lévy distribution. This difficulty, coupled with the fact that moments of order greater than α do not exist whenever $\alpha \neq 2$, has hindered the use of standard estimation methods such as maximum likelihood and the method of moments. Researchers have thus devised alternative estimation methods, mainly based on quantiles [16], the performance of which is judged unsatisfactory in a number of respects, especially because they are not liable to be incorporated in complex models and thus require a two-step estimation approach. With the availability of powerful computing machines, it has become possible to devise computationally-intensive estimation methods for the estimation of α-stable distributions, such as maximum likelihood based on the inverse FFT of the characteristic function, as in [17], or on direct numerical integration as in [18]. In the signal processing field, α-stable distributions have been widely used (see, e.g. [19]); for a general overview of estimation procedures in this setting, see [1]. A plot of some stable densities and distribution functions, approximated by the inverse FFT method, is reported in figure 1.

Given these computational difficulties, it is perhaps surprising that exact simulated values from α-stable distributions can be straightforwardly produced with a simple analytic transformation of two uniformly distributed random numbers [13] – this method opens up the possibility for optimal Monte Carlo-based implementations. The possibility of a simulation based Bayesian approach was first put forth in [20], who shows how to devise an auxiliary variable conditional on which the likelihood can be expressed in closed form. Unfortunately, simulated values from this auxiliary variable cannot be readily produced and one must resort to rejection sampling or to an MCMC subchain. Furthermore, several reparameterizations are needed in order to obtain posterior distributions that can be easily simulated from. This makes the whole procedure quite slow, especially when large sample sizes are involved.

In the case of symmetric stable distributions, the situation is less cumbersome: we can exploit the fact that, if we have two random variables

$$X_1 \sim \mathcal{S}(\alpha_1, 0), \quad X_2 \sim \mathcal{S}(\alpha_2, 1),$$

then the distribution of their product is

$$X_1 X_2 \sim \mathcal{S}(\alpha_1 \alpha_2, 0).$$

Here the notation '$\sim$' denotes that the random variable to the left has the distribution on the right. As an immediate corollary, it follows that any

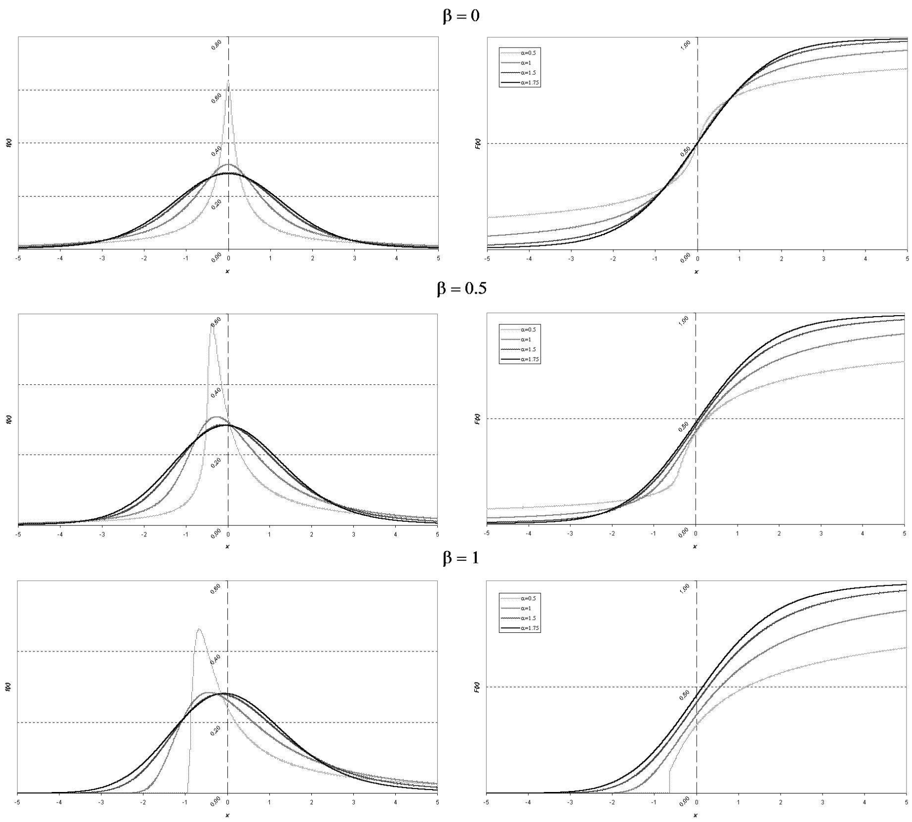

Figure 1: Probability density and cumulative distribution function of a standard stable random variable for different parameters values.

symmetric α-stable random variable can be thought of as the product of a Gaussian X_1 ($\alpha_1 = 2$) and a perfectly skewed (positive) random variable

$$X_2 \sim \mathcal{S}\left(\tfrac{\alpha_2}{2}, 1\right).$$

In other words, a symmetric α-stable distribution can be represented as a scale mixture of normals [21, 22]. To see how this works in practice, let us consider a generic model with noise at time i expressed as

$$\epsilon_i \sim \mathcal{S}(\alpha, 0, \gamma, \delta), \qquad i = 1, \ldots, N.$$

If we introduce an auxiliary white noise $u_i \sim \mathcal{N}(0, 1)$, where $\mathcal{N}(0, 1)$ denotes a standard normal distribution, using the above property allows us to express the α-stable noise as

$$\epsilon_i = \delta + \gamma\sqrt{\lambda_i}u_i, \qquad \lambda_i \sim \mathcal{S}\left(\tfrac{\alpha}{2}, 1\right),\ u_i \sim \mathcal{N}(0, 1). \tag{2}$$

Conditionally on λ_i, we have thus

$$\epsilon_i|\lambda_i \sim \mathcal{N}(\delta, \gamma^2\lambda_i). \tag{3}$$

So, if we can generate appropriately simulated values from λ, we can condition upon this vector to return into a Gaussian framework. This idea has been employed successfully in MCMC samplers, see [5, 14, 15].

2.1 α-stable distributions in noise modelling

The theoretical argument in favor of α-stable distributions is supported by a good fit on real noise data in several applications, including communications [1, 2]. Here we provide a further motivating example, which is used for our real data simulations later in the paper. The noisy data are taken from early sound recordings from an early 20th century archive. This particular archive contains much challenging material that has proved difficult to process using the standard techniques [23], and it was proposed that an α-stable model might be more appropriate here. An excerpt of just over one second (44487 observations) of the recording, in which there was only noise and no musical signal present, was extracted and fitted with a stable distribution, using an approximate maximum likelihood method based on the FFT of the characteristic function[1]. Results are reported in table 1, along

[1]For a more detailed description on how the maximum likelihood procedure is implemented, we refer to [24].

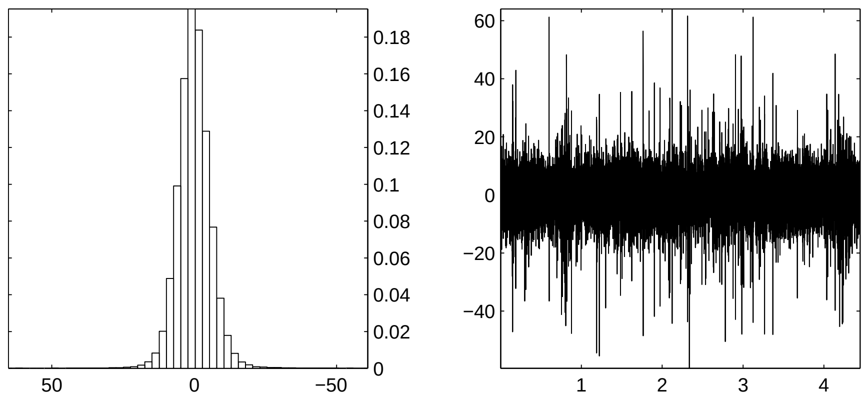

Figure 2: Histogram and pattern of the noise.

with the estimated parameters for a simple Gaussian and a more standard heavy-tailed Student's t model, estimated using standard Maximum Likelihood procedures. In figure 3 we report the kernel density estimate of the data set and the normal, the Student's t and the symmetric α-stable fitted densities. The estimated α parameter for the stable distribution was approximately 1.8, and provides a much better fit both in the central part and in the tails of the distribution when compared with both the Gaussian and the Student's t model. We also fitted asymmetric stable distributions to the same data, but found the fit was not significantly improved. Thus in order to be able to exploit the mixture of normals representation (which only applies for the symmetric α-stable case) we will restrict our attention in the following to the symmetric case.

Table 1: Maximum likelihood estimates of a symmetric α-stable, Student's t and Gaussian distribution for the noise sample depicted in figure 2.

	α-Stable			*Student's t*		*Gaussian*
	Estimate	Std. err.		Estimate		Estimate
α	1.8297	0.0061	ν	5.5145		
δ	-0.0281	0.0177	μ	0.1006	μ	0.0166
γ	3.6579	0.0104	σ	4.6706	σ	5.9642

3 Statistical Models

The approach to α-stable modelling described above can be readily applied to any time series models containing symmetric α-stable noise or disturbance

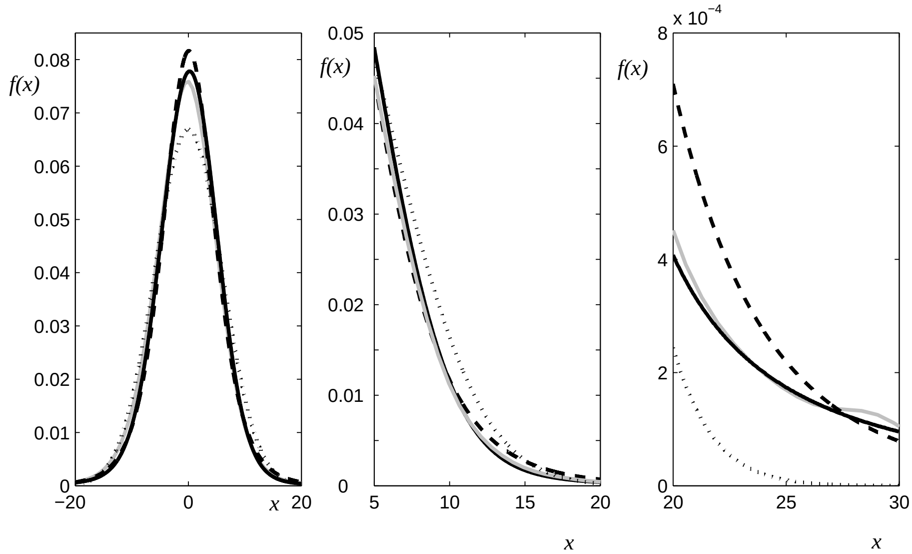

Figure 3: Kernel density (grey line), Gaussian fit (dotted line), Student's t fit (dashed line) and α-stable fit (solid line) - shown in the center of the distribution (left), middle tails (middle) and far tail (right)

terms. Here we focus on the important class of autoregressive (AR) time series, which have been widely used in many signal processing settings. Here we adopt the time-varying autoregressive (TVAR) process, in which the AR coefficients evolve over time according to certain specified dynamics. Models of this type have been employed in signal processing by many researchers, see e.g. [10–12, 25, 26]. The TVAR (p) model we use can be expressed as,

$$x_t = \sum_{k=1}^{p} a_{k,t} x_{t-k} + \sigma_{\epsilon_t} \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0,1), \tag{4}$$

and is assumed to be observed in additive symmetric α-stable noise such that the observed signal is

$$y_t = x_t + \gamma_{\eta_t} \eta_t, \qquad \eta_t \sim \mathcal{S}(\alpha, 0), \tag{5}$$

where σ_{ϵ_t} and γ_{η_t} represent, respectively, the standard deviation of the innovations in the true signal process and the scale of the stable noise; both are allowed to be time-varying. We furthermore assume that ϵ_t and η_t are independent of each other and over time. The time-varying parameter vector of the model has thus dimension $p+2$ and is given by

$$\theta_t = \left(\mathbf{a}_t, \phi_{\epsilon_t}, \phi_{\eta_t}\right), \tag{6}$$

where

$$\begin{aligned}
\mathbf{a}_t &= (a_{1,t}, a_{2,t}, \ldots, a_{p,t}) \\
\phi_{\epsilon_t} &= \ln \sigma^2_{\epsilon_t} \\
\phi_{\eta_t} &= \ln \gamma^2_{\eta_t};
\end{aligned}$$

its support is given by $A_p \times \mathbb{R} \times \mathbb{R}$, where A_p is the region of stability of a stationary AR(p) process[2].

The above model can be readily expressed in state-space form [27–29]. The system matrices are

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{a}'_t \\ \mathbf{I}_{p-1} \;\; \mathbf{0}_{k-1\times 1} \end{bmatrix} \qquad \mathbf{B}_t = \begin{bmatrix} \sigma_{\epsilon_t} \\ \mathbf{0}_{k-1\times 1} \end{bmatrix} \tag{7}$$
$$\mathbf{C} = [1 \;\; \mathbf{0}_{k-1\times 1}] \qquad \mathbf{D}_t = [\gamma_{\eta_t}]$$

and, defining $\check{\mathbf{x}}_t = (x_t, x_{t-1}, \ldots, x_{t-p+1})$,

$$\check{\mathbf{x}}_t = \mathbf{A}_t \check{\mathbf{x}}_{t-1} + \mathbf{B}_t \mathbf{v}_t \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{8}$$

$$\mathbf{y}_t = \mathbf{C} \check{\mathbf{x}}_t + \mathbf{D}_t \mathbf{u}_t \qquad \mathbf{u}_t \sim \mathcal{S}(\alpha, \mathbf{0}). \tag{9}$$

Now, exploiting the mixture of normal representation of a stable distribution (2), we can redefine

$$\mathbf{D}^\star_t = \left[\gamma_{\eta_t} \sqrt{\lambda_t}\right] \qquad \lambda_t \sim \mathcal{S}\left(\tfrac{\alpha}{2}, 1\right) \tag{10}$$

and express (9) as

$$\mathbf{y}_t = \mathbf{C} \check{\mathbf{x}}_t + \mathbf{D}^\star_t \mathbf{w}_t \qquad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{11}$$

so that the model is expressed in conditionally Gaussian state space form. According to this approach, λ_t would be treated as a unknown parameter and incorporated into θ_t.

The evolution of θ_t over time (excluding λ_t) obeys a first order Markov process, whose parameters are assumed to be fixed and known:

$$\begin{aligned}
p(\theta_0) &= p(\mathbf{a}_0) p(\phi_{\epsilon_0}) p(\phi_{\eta_0}) p(\lambda_0) \\
p(\theta_t | \theta_{t-1}) &= p(\mathbf{a}_t | \mathbf{a}_{t-1}) p(\phi_{\epsilon_t} | \phi_{\epsilon_{t-1}}) p(\phi_{\eta_t} | \phi_{\eta_{t-1}}) p(\lambda_t)
\end{aligned}$$

[2]This condition is only sufficient and not necessary when dealing with TVAR processes. However, regions of stability for TVAR processes are much more complex to deal with, so we have decided to enforce this simpler condition.

with

$$
\begin{aligned}
p(\mathbf{a}_0) &\propto \mathcal{N}(\mathbf{0}, \Delta_{\mathbf{a}_0})\mathbb{I}_{\mathbf{a}_0 \in A_p} \\
p(\mathbf{a}_t|\mathbf{a}_{t-1}) &\propto \mathcal{N}(\mathbf{a}_{t-1}, \Delta_{\mathbf{a}})\mathbb{I}_{\mathbf{a}_t \in A_p} \\
p(\phi_{\epsilon_0}) &= \mathcal{N}(0, \delta^2_{\epsilon_0}) \\
p(\phi_{\epsilon_t}|\phi_{\epsilon_{t-1}}) &= \mathcal{N}(\phi_{\epsilon_{t-1}}, \delta^2_{\epsilon}) \\
p(\phi_{\eta_0}) &= \mathcal{N}(0, \delta^2_{\eta_0}) \\
p(\phi_{\eta_t}|\phi_{\eta_{t-1}}) &= \mathcal{N}(\phi_{\eta_{t-1}}, \delta^2_{\eta}) \\
p(\lambda_t) &= \mathcal{S}(\tfrac{\alpha}{2}, 1).
\end{aligned} \tag{12}
$$

where $\mathbb{I}_{x \in X}$ is the indicator function of the set X, taking value 1 whenever $x \in X$ and 0 elsewhere.

4 Sequential Monte Carlo Methods

The principal goal of this work will be to extract on the basis of the observable noisy signal $\{y_t\}$, the unobservable clean signal $\{x_t\}$. One could be interested in simply obtaining a point estimate $\hat{x}_t$ for every time interval, but in Bayesian terms it is much more interesting to focus the analysis on the filtering distribution $p(\check{\mathbf{x}}_t, \theta_t|\mathbf{y}_{1:t})$ or on the fixed-lag smoothing distribution $p(\check{\mathbf{x}}_t, \theta_t|\mathbf{y}_{1:t+L})$. On the basis of these we can construct both point estimates and Highest Probability Density (HPD) intervals for x_t, for example.

We note here that many possible particle filtering algorithms can be adapted to the present context. The most relevant will be those existing methods for TVAR models in Gaussian noise, see e.g. [9–12]. Of these, potentially the most accurate is [10] since it applies a full Rao-Blackwellisation procedure [8] to the signal states. Hence we choose [10][3] as the basis for our α-stable particle filter. The other TVAR particle filters cited can also readily be used in our application, and yield much faster filtering at the expense of some accuracy in the estimation performance.

4.1 Kalman filtering

Expressed in the above formulation the model is not linear, and closed-form algorithms such as the Kalman filter cannot be employed. However, under the redefined model (11) it can be observed that, conditionally on

[3] We would like to thank Dr. Jaco Vermaak for giving us access to his Rao-Blackwellised Matlab code.

$\theta_{0:t}$, the model is linear and Gaussian; $p(\check{\mathbf{x}}_t|\theta_{0:t}, \mathbf{y}_{1:t})$ can thus be obtained analytically using the Kalman filter and the prediction error decomposition [27]. This standard procedure is now briefly reviewed for the case of our model.

The Kalman filter runs as follows: for $k = 1, \ldots, t$ we first set the sufficient statistics for the predictive distributions

$$\begin{aligned} \mathbf{m}_{k|k-1}(\theta_{0:k}) &= \mathbf{A}_k \mathbf{m}_{k-1|k-1} \\ \mathbf{P}_{k|k-1}(\theta_{0:k}) &= \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}'_k + \mathbf{B}_k \mathbf{B}'_k \\ \mathbf{y}_{k|k-1}(\theta_{0:k}) &= \mathbf{C} \mathbf{m}_{k|k-1}; \end{aligned} \tag{13}$$

we compute

$$\mathbf{S}_k(\theta_{0:k}) = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}' + \mathbf{D}^{\star}_k \mathbf{D}^{\star\prime}_k, \tag{14}$$

and we finally obtain the parameters of the filtering distribution according to

$$\begin{aligned} \mathbf{m}_{k|k}(\theta_{0:k}) &= \mathbf{m}_{k|k-1} + \mathbf{P}_{k|k-1}\mathbf{C}'\mathbf{S}_k^{-1}\left(\mathbf{y}_k - \mathbf{y}_{k|k-1}\right) \\ \mathbf{P}_{k|k}(\theta_{0:k}) &= \mathbf{P}_{k|k-1} - \mathbf{P}_{k-1|k-1}\mathbf{C}'\mathbf{S}_k^{-1}\mathbf{C}\mathbf{P}_{k|k-1}. \end{aligned}$$

The filtering distribution of the state vector is thus

$$p(\check{\mathbf{x}}_k|\theta_{0:k}, \mathbf{y}_{1:k}) = \mathcal{N}\left(\mathbf{m}_{k|k}, \mathbf{P}_{k|k}\right), \tag{15}$$

and the likelihood of the last observation is

$$p(\mathbf{y}_k|\theta_{0:k}, \mathbf{y}_{1:k-1}) = \mathcal{N}\left(\mathbf{A}_k\mathbf{m}_{k|k}, \mathbf{D}^{\star}_k + \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}'_k\right). \tag{16}$$

Now, since

$$\begin{aligned} p(\check{\mathbf{x}}_t, \theta_{0:t}|\mathbf{y}_{1:t}) &= p(\check{\mathbf{x}}_t|\theta_{0:t}, \mathbf{y}_{1:t})p(\theta_{0:t}|\mathbf{y}_{1:t}) = \\ &= p(\check{\mathbf{x}}_t|\theta_{0:t}, \mathbf{y}_{1:t})p(\theta_{0:t})\prod_{\tau=1}^{t} p(\mathbf{y}_\tau|\mathbf{y}_{1:\tau-1}, \theta_{0:\tau-1}) \end{aligned}$$

the problem reduces to one of obtaining simulated values from $p(\theta_{0:t}|\mathbf{y}_{1:t})$ in order to produce a random sample to be used for Monte Carlo inference[4]. This is in general difficult, and an importance sampling technique can be employed. Given a probability distribution $\pi(\theta_{0:t}|\mathbf{y}_{1:t})$ which is easy to simulate from, we produce a set of M random vectors $\theta_{0:t}$ from it and assign to each one a weight

$$w(\theta_{0:t}) \propto \frac{p(\theta_{0:t}|\mathbf{y}_{1:t})}{\pi(\theta_{0:t}|\mathbf{y}_{1:t})}$$

to be used in Monte Carlo inference.

[4]This is an example of the Rao – Blackwellized procedure, see [8].

4.2 Particle Filters

In the above importance sampling framework the data are processed in batches and, as new observations arrive, it is necessary to produce a new sample from the importance distribution (with increasingly large sample size) and recompute the importance weights. In many practical situations, however, ranging from the signal processing to the financial field, data are naturally available on a sequential basis, and having to re-run the whole estimation as new data arrives is often not feasible when new observations arrive at a high rate.

Particle filtering methods have been recently proposed for state space models [6, 7], see also [8, 30, 31] for recent developments and review. The idea underlying this approach is to represent the distribution of interest by a large number of random samples, or *particles*, evolving over time on the basis of a simulation-based updating scheme, so that new observations are incorporated in the filter as they become available.

More formally, the objective is to update, at each time t, $p(\theta_{0:t}|\mathbf{y}_{1:t})$ without modifying the past values of θ. The importance distribution should thus be such that

$$\pi(\theta_{0:t}|\mathbf{y}_{1:t}) = \pi(\theta_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\theta_t|\theta_{0:t-1}, \mathbf{y}_{1:t}),$$

and the weights factorize as $w(\theta_{0:t}) = w(\theta_{0:t-1})w_t$ where

$$w_t \propto \frac{p(\mathbf{y}_t|\theta_{0:t}, \mathbf{y}_{1:t-1})p(\theta_t|\theta_{t-1})}{\pi(\theta_t|\theta_{0:t-1}, \mathbf{y}_{1:t})}.$$

The weights can then be updated recursively, since $w(\theta_{0:t}) = w(\theta_{0:t-1})w_t$. It was shown by [8] that the optimal importance distribution, that is the one that minimizes the variance of the importance weights, is $p(\theta_t|\theta_{0:t-1}, \mathbf{y}_t)$. Unfortunately in our case this is not easy to simulate from. A simple alternative, as employed by [6] and [7], is to use

$$\pi(\theta_t|\theta_{0:t-1}, \mathbf{y}_{1:t}) = p(\theta_t|\theta_{t-1}), \quad w_t \propto p(\mathbf{y}_t|\theta_{0:t}, \mathbf{y}_{1:t-1})$$

i.e. the importance weights are then simply proportional to the marginal likelihood, computed using the Kalman filter in (16).

The weights are then normalized according to

$$\tilde{w}\left(\theta_{0:t}^{(i)}\right) = \frac{w\left(\theta_{0:t}^{(i)}\right)}{\sum_{j=1}^{M} w\left(\theta_{0:t}^{(j)}\right)},$$

where $\theta_{0:t}^{(i)}$ denotes the i-th *particle*. This is basically the approach employed in the seminal paper by [6], plus the resampling step which we will discuss in the following paragraph.

4.3 Resampling

It turns out that an algorithm of this kind will eventually degenerate, i.e. assign almost all the weight to a single particle. In order to overcome this problem, a resampling step is necessary. In the resampling step, particles with low importance weights are discarded and those with high importance are multiplied. More formally, after producing a set of particles from the importance distribution and having assigned to each one an appropriate weight, we associate to each particle i a number of offspring M_i such that $\sum_{i=1}^{M} M_i = M$. After this selection step, offspring particles replace the original particles and the importance weights are reset to $1/M$, so that the set of particles can be thought of as a random sample. The resampling step can be implemented at every time interval [6], or it can be employed whenever the set of particles crosses a certain degeneracy threshold. A measure of degeneracy of the algorithm is the effective sample size [31], defined as

$$M_e = \frac{M}{1 + \mathrm{Var}(w_t)}.$$

This quantity can be estimated by

$$\hat{M}_e = \frac{1}{\sum_{i=1}^{M} \tilde{w}_t^{2(i)}};$$

when $\hat{M}_e$ drops below a certain threshold, the resampling takes place.

Several resampling schemes have been proposed in the literature [6, 31]; in our specific case, we will employ *Systematic sampling*. For a description of the algorithm, we refer to [31].

To wrap up, what the algorithm practically implements at each time interval is the following:

1. Sample M particles $\theta_t^\dagger$ from the importance distribution $\pi(\theta_t|\theta_{0:t-1}, \mathbf{y}_{1:t})$ and set $\theta_{0:t}^\dagger = (\theta_t^\dagger, \theta_{0:t-1})$.

2. Evaluate the importance weights according to

$$w_t \propto \frac{p(\mathbf{y}_t|\theta_{0:t}^\dagger, \mathbf{y}_{1:t-1})p(\theta_t^\dagger|\theta_{t-1})}{\pi(\theta_t^\dagger|\theta_{0:t-1}, \mathbf{y}_{1:t})}.$$

3. Normalize the importance weights:

$$\tilde{w}(\theta_{0:t}^{(i)}) = \frac{w\left(\theta_{0:t}^{(i)}\right)}{\sum_{j=1}^{M} w\left(\theta_{0:t}^{(j)}\right)}.$$

4. Resample if $\hat{M}_e$ below threshold by multiplying or discarding particles according to their weight to produce a new set of M particles $\theta_{0:t}$, each with weight $\tilde{w}(\theta_{0:t}) = 1/M$.

An issue which is closely related to degeneracy is that of the depletion of samples. When performing the resampling step, particles with high importance weight tend to be sampled a large number of times and it could happen that the initial set of particles ends up in collapsing into a single particle. We will examine in what follows two situations in which the depletion of samples should be seriously taken into account.

4.4 Fixed-lag smoothing

In some cases, in order to obtain a smoother estimate of the state, it is useful to consider the distribution at time t after a certain number of time intervals L. In more formal terms, instead of considering $p(\check{\mathbf{x}}_t|\mathbf{y}_{1:t})$, we focus on $p(\check{\mathbf{x}}_t|\mathbf{y}_{1:t+L})$. It is hoped that expanding the information set by the use of an appropriately chosen lag window L improves the estimates of the states. In principle, fixed-lag smoothed densities can be straightforwardly obtained by the general algorithm proposed above, by simply extracting signal estimates at time $t-L+1$ from the particles. This simple method obviously involves no increase in computational burden and will be used in one of the experiments. However, it turns out that such a scheme, where states at time $t-L+1$ could have been resampled up to L times, leads to serious degeneracy as L grows larger.

In order to overcome this degeneracy problem, an MCMC approach similar to [11] can be adopted. Let us consider that, at time $t+L$, the particles are distributed according to $p(\theta_{0:t+L}|\mathbf{y}_{1:t+L})$; the idea is to apply to each particle a Markov transition kernel K^* with invariant distribution $p(\theta_{0:t+L}|\mathbf{y}_{1:t+L})$ in order to introduce diversity among the particles, as proposed originally in [32].

If we denote with $\theta_{0:t+L}^{\prime(i)}$ the i-th particle after the resampling stage, the MCMC proceeds by sampling each particle and state $\theta_k^{(i)}$ according to target density $p(\theta_k|\theta_{-k}^{(i)}, \mathbf{y}_{1:t+L})$, where

$$\theta_{-k}^{(i)} = \left(\theta_{0:t-1}^{*(i)}, \theta_t^{(i)}, \ldots, \theta_{k-1}^{(i)}, \theta_{k+1}^{*(i)}, \ldots, \theta_{t+L}^{*(i)}\right),$$

and $k = t, t+1, \ldots, t+L$. A Metropolis-within Gibbs sampling MCMC scheme can be designed to achieve this. In the Gaussian noise case, a suitable scheme, including a forward-backward Kalman filter for efficient implementation, may be found in [11]. The extension of the above scheme to the α stable case is straightforward, involving a change of the observation model as in (11). This requires MCMC steps for the additional λ parameters introduced in that model. This additional step can readily be accomplished using the scheme of [14, 15]. However, it proved to be an unnecessary step in the cases we considered and the λ's we will use are simply those obtained directly from the importance sampling step.

4.5 Static parameters

The degeneracy problem is however much more severe whenever the particle filter has to deal with the estimation of static parameters. The prior $p(\theta_{t+1}|\theta_t)$ would have probability mass 1 at θ_t, so the particles are never updated and rejuvenated and they eventually collapse on a few – and sometimes even one – single value. In our specific case, we have up to now assumed the stable law tail parameter α to be known. This is rarely the case in practical applications. Furthermore, whereas in static estimation problems one could somehow pre-estimate static parameters, in our sequential estimation case this is obviously impossible. Fixing the static parameters to arbitrarily chosen guesses is in general very bad practice. In our specific case, however, a few experiments, not reported here for sake of brevity, have reported that, even if the guessed α is not very close to the actual one, the results are still satisfactory and the improvement of the signal remains at the same order of magnitude. However, it would surely be preferable to estimate α together with the other parameters as the data are processed.

Several approaches to this problem are available. The MCMC schemes above for fixed lag smoothing can be adapted to the static parameter case (see, e.g. [33, 34]), for example, although in our case this led to a filter of ever growing computational complexity and so was not adopted. Another approach to overcome the degeneracy problem is to introduce artificial parameter evolution, that is, simply pretending that static parameters are indeed time-varying by adding a noise term at each time interval [35]. The problem is that in doing so we introduce additional variability by "throwing away" information. In [36], a method for quantifying this loss of information is proposed and an artificial parameter evolution scheme immune to this problem is introduced. To focus the attention on our specific case, we note that the static parameter is the stability index α. Introducing artificial parameter evolution is equivalent to considering a model in which α is replaced by its time-varying analog α_t which evolves according to

$$\alpha_{t+1} = \alpha_t + \zeta_{t+1}, \qquad \zeta_t \sim \mathcal{N}(0, \omega_t).$$

In a situation in which α is fixed, the posterior distribution $p(\alpha|\mathbf{y}_{1:t})$ could be characterized by its Monte Carlo mean and variance $\bar{\alpha}_t$ and s_t^2. It is immediate to observe that, in the case of artificial parameter evolution, the Monte Carlo variance increases to $s_t^2 + \xi_t$. The Monte Carlo approximation can be expressed as kernel smoothed density of the particles as

$$p(\alpha|\mathbf{y}_{1:t}) \approx \sum_{j=1}^{M} w_t^{(j)} \mathcal{N}\left(\alpha_{t+1}|\alpha_t^{(j)}, \xi_t\right).$$

Now the target variance s_t^2 can be expressed as

$$s_t^2 = s_{t-1}^2 + \xi_t + 2\mathrm{Cov}(\alpha_{t-1}, \zeta_t),$$

so if we choose

$$\mathrm{Cov}(\alpha_{t-1}, \zeta_t) = -\frac{\xi_t}{2}$$

we have managed to avoid the loss of information. A simple particular case in which this can be achieved is to consider

$$\omega_t = s_t^2 \left(\frac{1}{\delta} - 1\right),$$

where δ is a discount factor in $(0, 1]$; the authors suggest its value to be chosen around 0.95-0.99. If we define $d = \frac{3\delta - 1}{2\delta}$, the conditional density evolution becomes

$$p(\alpha_{t+1}|\alpha_t) \sim \mathcal{N}\left(\alpha_{t+1}|d\alpha_t + (1-d)\bar{\alpha}_t, h^2 s_t^2\right), \tag{17}$$

where

$$h^2 = 1 - d^2 = 1 - \left(\frac{3\delta - 1}{2\delta}\right)^2,$$

so that sampling from (17) is equivalent to sampling from a kernel smoothed density in which the smoothing parameter h is controlled via the discount factor δ.

5 Experiments and Results

In this section we will show how the sequential Monte Carlo method outlined above performs on both simulated and real data. As a benchmark of model performance, we will use the signal to noise ratio, defined as

$$\mathrm{SNR} = 10 \log_{10} \frac{\sum_{t=1}^{T} x_t^2}{\sum_{t=1}^{T} (x_t - z_t)^2}, \tag{18}$$

where x_t is the clean signal and z_t represents, in turn, the observed noisy signal and the filtered state. It is obviously hoped that replacing the noisy signal y_t with the filtered state $\hat{x}_t$ produces an improvement in the SNR. Since the variance of α-stable distribution is infinite for $\alpha < 2$, it is obvious that the expression above is inconsistent from an inferential point of view. As T goes to infinity, the denominator will diverge to infinity too, thus yielding a SNR of $-\infty$ and an infinitely large improvement for any kind of filtering algorithm. A modified version of the SNR that yields consistent results in the case of α-stable noise could be constructed by expl3oiting the fact that, if Z has stable distribution with characteristic index α, $\mathrm{E}\left(|x|^{\alpha}\right) < \infty$:

$$\mathrm{SNR}_{\alpha} = 10 \log_{10} \frac{\sum_{t=1}^{T} |x_t|^{\alpha}}{\sum_{t=1}^{T} |x_t - z_t|^{\alpha}}. \tag{19}$$

In our case, however, we will deal with small sample sizes and, for ease of comparison with other results commonly available in the literature, we have decided to use the traditional SNR. In one of the experiments that follow, however, we have compared the performance of the indicators, highlighting that they perform approximately the same when small sample sizes are involved.

We will start by considering the simplest case, that is the one in which α is known a priori and we do not perform fixed-lag smoothing, so that there is no need for the MCMC step outlined in the above subsection. The importance function was taken to be the prior $p(\theta_t|\theta_{t-1})$; as a resampling scheme, we will use systematic sampling, applied at each time step.

We have generated a synthetic signal of 200 observations with parameters $\Delta_{\mathbf{a}_0} = 2\mathbf{I}$, $\Delta_{\mathbf{a}} = 0.0005\mathbf{I}$, $\delta_{\epsilon_0}^2 = 0.2$, $\delta_{\epsilon}^2 = 0.005$, $\delta_{\eta_0}^2 = 0.5$, $\delta_{\eta}^2 = 0.00005$; the signal was then corrupted with symmetric α-stable noise with $\alpha = 1.4$. The SNR of the noisy observations was 0.83dB. The synthetic data are depicted in figure 4; the corresponding parameter values are reported in figure 5.

Using a simple Gaussian model, using the algorithm proposed by [11], leads to very poor results, as should be expected. The filtered states, along

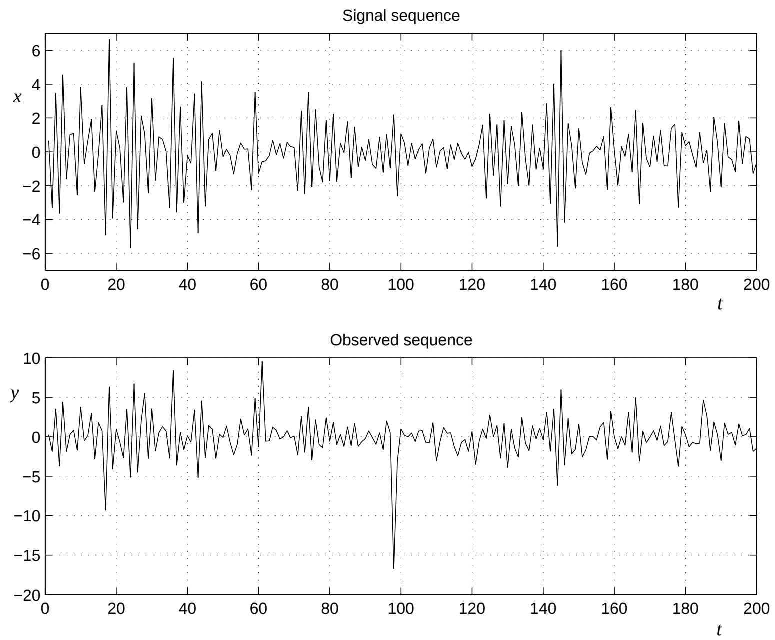

Figure 4: Clean and noisy signal, synthetic data.

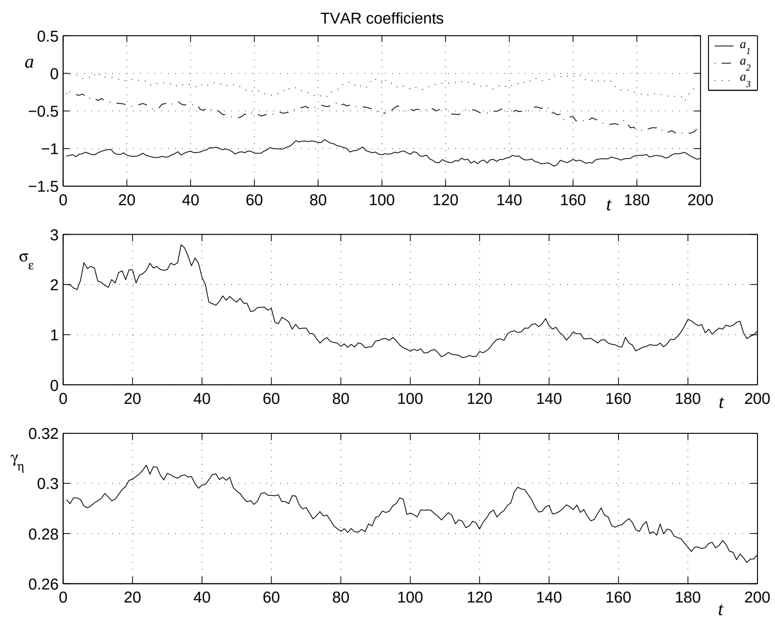

Figure 5: Evolution of the true parameters of the model.

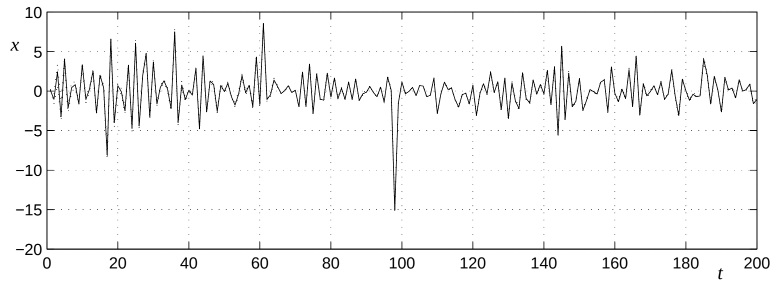

Figure 6: Filtered signal (solid line) with 95% quantile bands (dotted lines), Gaussian noise.

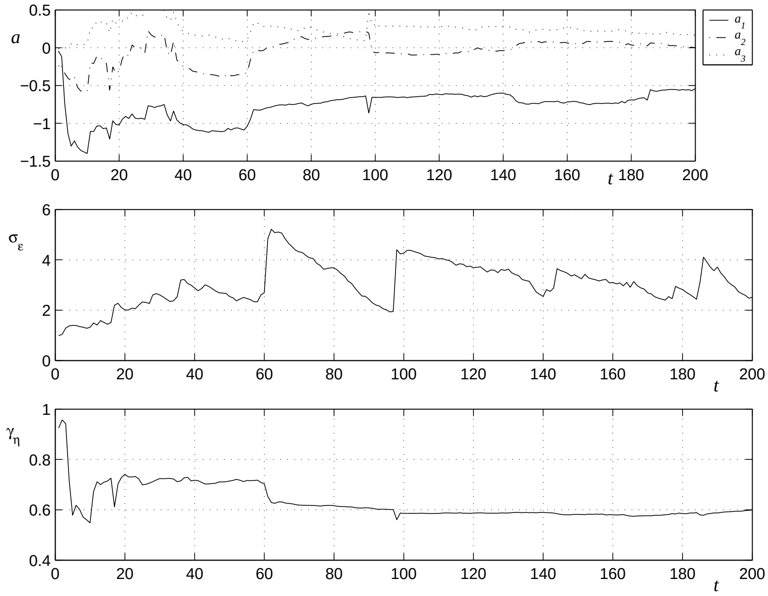

Figure 7: Estimated parameters of the model, Gaussian noise.

with the corresponding 95% quantiles, are displayed in figure 6. We can observe that, especially when the signal is highly corrupted by the noise peaks, the filtered states are very near to the observations, according to the low likeliness of such extreme values under the Gaussian noise assumption. Furthermore, as becomes apparent looking at the estimated parameter values (figure 7), the extreme observations are somehow "absorbed" by jumps in the variance of the signal. The overall improvement in SNR was of 0.86dB, with RMSE 1.6947.

On the other hand, the use of the stable model greatly reduces the influence of the extreme noise observations, achieving a SNR improvement of

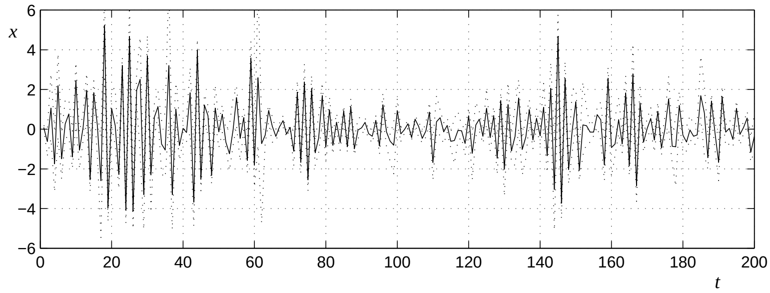

Figure 8: Filtered signal (solid line) with 95% quantile bands (dotted lines), stable noise, $\alpha = 1.4$.

5.12dB with RMSE 1.0382. Looking at figure 8, we can observe how the filter was not misled by extreme observations. Now, concerning the estimated parameter evolution, displayed in figure 9, we can observe that the estimated parameters start displaying a trajectory similar to that of their actual counterparts around the fiftieth observation.

Similar results hold when α is estimated along with the other parameters. The prior we used for α was a simple uniform on the support $(0.2, 2)$[5], and we fixed the discount factor δ in (17) to 0.95. The evolution of the stability index is depicted in the top graph of figure 11 along with the 95% quantile bands. The SNR improvement is 5.13dB with RMSE 1.0372, nearly identical to the case analyzed earlier in which we fixed α to its true value. The evolution of the kernel smoothed posterior distribution of α for the last time intervals is presented in figure 12.

In order to get insights about the appropriate number of particles to be used, we have performed a Monte Carlo experiment consisting of 50 independent replications. All experiments were performed on a laptop computer with a 2.66GHz Intel® Pentium® IV processor with 512Mb RAM. The results, reported in Table 2 seem to indicate that using more than 300 particles does not lead to a significantly improved performance despite the increase in computational effort. A number of particles between 100 and 300 seems to be a good compromise between speed and accuracy.

Concerning the fixed-lag smoothing, we have performed a simulation experiment consisting of 50 independent replications over 100 particles for different lengths of the lag window. Results are reported in Table 3 and suggest that an optimal lag window could be between 5 and 10. However, if

[5]Values of α smaller than 0.2 were ruled out *a priori* as too heavy-tailed.

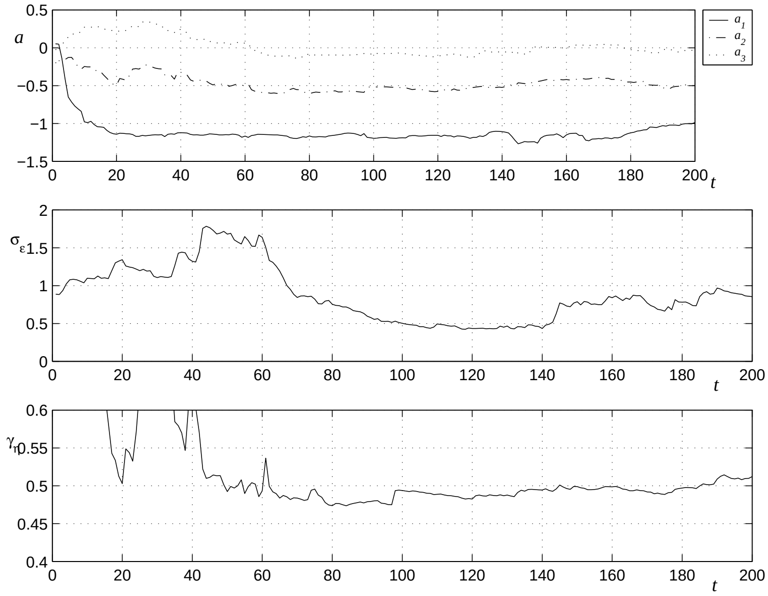

Figure 9: Estimated parameters of the model, stable noise, fixed $\alpha = 1.4$.

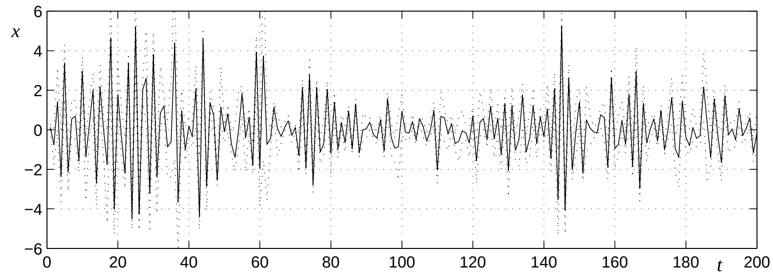

Figure 10: Filtered signal (solid line) with 95% quantile bands (dotted lines), stable noise, $\alpha = 1.4$.

Table 2: RMSE and mean and standard deviation (in parentheses) of SNR improvement, over 50 independent replications, for different number of particles M. The last row reports the average time (in seconds) required to process one observation.

	$M = 10$	$M = 50$	$M = 100$	$M = 300$	$M = 500$
RMSE	1.3688	0.9921	0.9893	0.9892	0.9892
SNR	2.8651	4.8815	5.1515	5.4857	5.4786
	(1.6433)	(0.7991)	(0.5344)	(0.2708)	(0.2222)
Time	0.0219	0.0897	0.1432	0.3810	0.6407

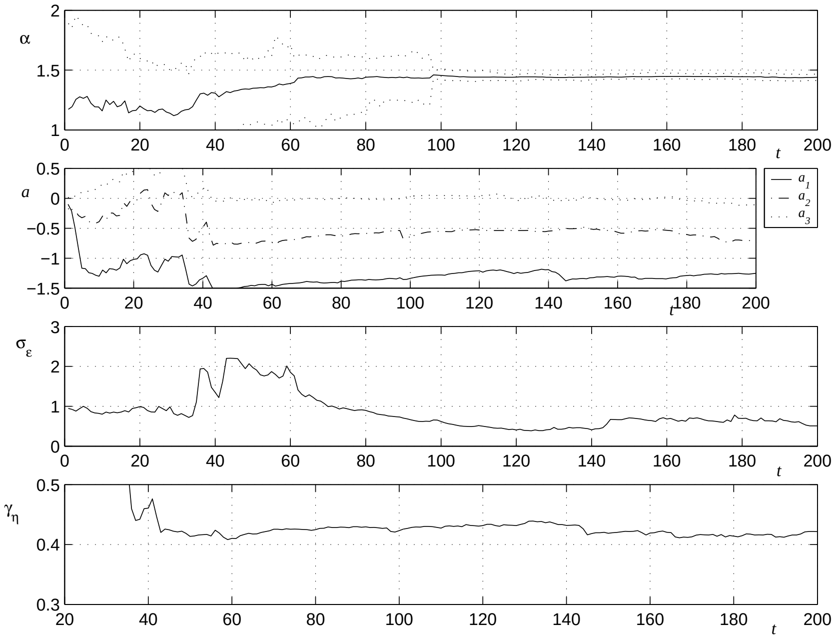

Figure 11: Estimated parameters of the model, stable noise.

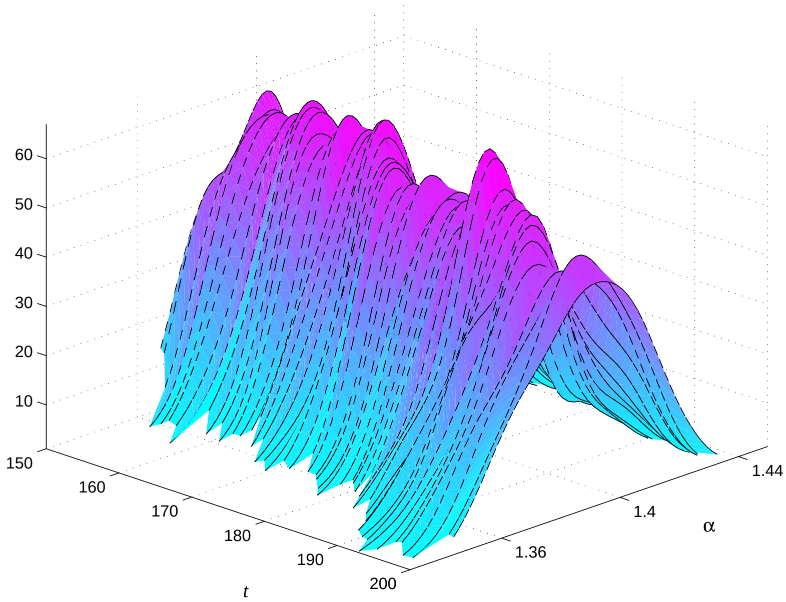

Figure 12: Kernel smoothed posterior densities of α for $t = 150, \ldots, 200$.

one is interested in processing the observations in a quicker way, it is possible to employ the non-Rao-Blackwellized version of the algorithm [9]. In this case, the time required to process one observation drops dramatically, at some expense to the signal estimation accuracy.

Table 3: RMSE and mean and standard deviation (in parentheses) of SNR improvement, over 50 independent replications, for different length of the lag window L with 100 particles. The last row reports the average time (in seconds) required to process one observation.

	$L = 0$	$L = 5$	$L = 10$	$L = 20$
RMSE	0.9892	1.0151	0.8893	0.9561
SNR	5.1515	6.1061	6.1907	5.8457
	(0.5344)	(0.4889)	(0.5483)	(0.6312)
Time	0.1432	1.0734	1.9126	3.5102

Up to now, we have employed for all the simulations the same synthetic data set depicted in figure 4. In order to confirm that the performance of our method does not depend on the particular data set at hand, we have conducted a simulation experiment, consisting of 50 independent replications, in order to assess the average gain in SNR with different synthetic data sets. As a benchmark we used both the standard SNR and its modified version that takes into account the non-finiteness of the variance of the data generating process defined in (19). Each of the synthetic time series consisted of 200 observations and was generated by the same algorithm as that in figure 4. We have employed a filter, without fixed-lag smoothing, maintaining various numbers of particles. Results are reported in table 4 and seem to suggest that the average SNR improvement depends weakly on the number of particles.

Table 4: Mean and standard deviation (in parentheses) of SNR and SNR_α improvement, over 50 independent replications on 50 different synthetic data sets, for various numbers of particles M.

	$M = 50$	$M = 100$	$M = 200$	$M = 500$
SNR	10.5877	10.8249	10.9606	11.0201
	(6.3530)	(6.1029)	(6.3236)	(6.1482)
SNR_α	5.2933	5.4453	5.5474	5.5693
	(3.5238)	(3.3315)	(3.4988)	(3.3687)

The final simulation experiment we have performed consists in artificially corrupting with symmetric α-stable noise a clean sequence of audio data; we have used the first 6.75 seconds of the Boards of Canada's "Music is Math" from the album "Geogaddi" (44.1KHz, 16 bit, mono). This audio source was originally computer-generated, and presents no kind of corruption or background noise. The parameters of the artificial noise were set to $\alpha = 1.7$, $\delta = 0$, and the scale parameter γ was evolved from its initial value 0.01 according to a Markov process as in (12), with $\delta = 0.01$. The resulting SNR is 3.9564. For illustration purposes, the filter was first run on an excerpt of 1000 observations (200001 to 201000 out of 261072, SNR 8.72); clean, noisy and filtered signal for this excerpt are displayed in figure 13. We have employed 200 particles, with a fixed-lag smoothing window of length 5; the filter performed remarkably well, achieving a SNR improvement of 12.66.

The same filter was then applied to the whole series, yielding again a remarkable SNR improvement of 10.88. The reconstructed audio source was then recoded in audio format, and informal listening tests confirmed the reduction of the noise. In particular, the filter performed very well in removing the peaks but left a small amount background noise[6], sounding like a feeble "hiss".

One issue which was quite sensitive in configuring the filters was the choice of the prior for the scale parameters σ and γ. Preliminary experiments pointed out how values very far from their true counterparts can lead to poor performance, mainly owing to the trade-off effect between the scale and the tail-thickness parameter and the slow evolution speed of the scale parameter. For example, a too small value of γ can lead α to decrease to compensate the effect. In the present case, we have bypassed the problem by using an informative normal prior centered on the true value of γ, which was obviously known a priori. The issue deserves however further attention in our future work - for example these parameters can be learnt off-line using a batch procedure such as MCMC and a short section of initial data.

We are now in position to consider an application of the above methodology to a genuine corrupted audio data set. The audio source chosen is from a music recording database of songs made in Palestine in the early twentieth century; this audio source is extremely noisy and corrupted; as we have pointed out in section 2, the noise of this audio source is modelled very well by an α-stable distribution. We have applied the particle filter,

[6] All audio examples presented in this paper can be downloaded at the URL `http://www.ds.unifi.it/mjl/sound.htm`.

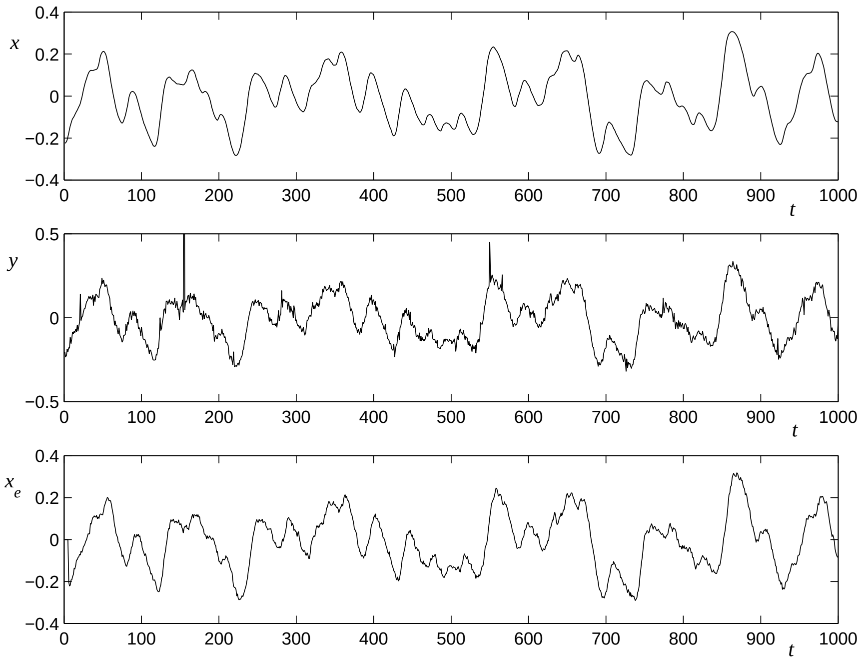

Figure 13: Excerpt of clean, noisy and reconstructed signal for Boards of Canada's "Music is Math".

with $M = 100$ and $L = 100$ to a short excerpt of two seconds of one of the audio tracks. Given the large number of lags involved, we have decided not to perform the MCMC step in order to reduce the computational time. The results were encouraging: the peaks in the corrupted audio signal were removed, leaving behind a "seemingly white" background noise that could be dealt with by traditional filtering methods. The final estimated value for α was 1.5509. In figure 14, we report a comparison between a short excerpt of the original signal and its filtered counterpart.

6 Conclusions

We have proposed and tested methods for performing on-line Bayesian filtering in TVAR models with symmetric α-stable noise distribution. Using such a distribution allows for more flexibility and permits successful modelling of the heavy-tailed noise which is often observed for example in corrupted audio time series. The performance of this filtering method was assessed on both simulated and real data, and the analysis of a genuinely degraded audio source suggested that α-stable distributions are particularly well suited

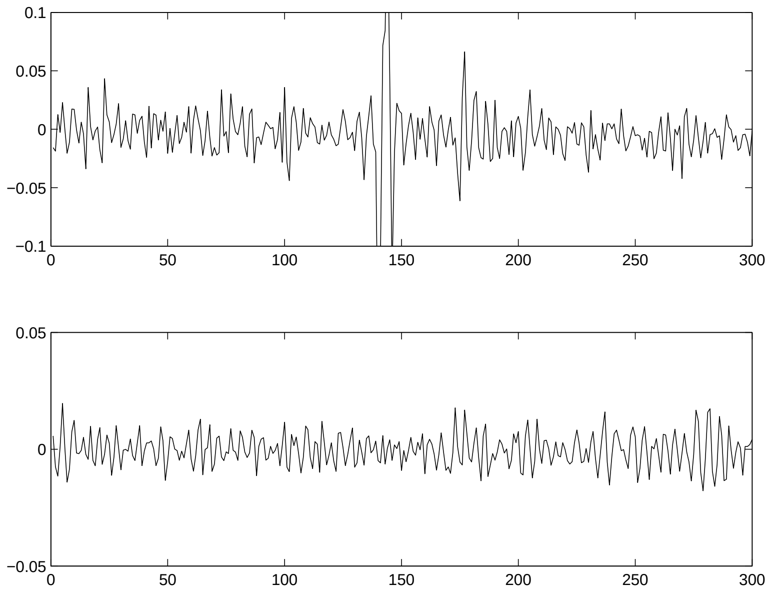

Figure 14: Original and filtered signal for the Lachmann database.

to model this kind of noise.

One reason why we considered only symmetric cases of α-stable distributions instead of the more general asymmetric version is that they can be represented exactly as a scale mixture of normals. This useful property, that allows us to use the Kalman filter by expressing the model in conditionally Gaussian form, does not hold for the more general asymmetric case. In that case, one should resort to more standard techniques to obtain the likelihood of every particle, but the necessity to perform the inversion of the characteristic function via the FFT at each time interval and, within a given time interval, for each particle, would lead to excessive computational requirements, at least according to the power of the machines available to us. In fact we believe from observation that the α-stable distributions involved in audio noise are very close to symmetric, so we do not regard this restriction as a serious limit to the methods in this application.

Although we have focused our analysis on symmetric α-stable distributions, this approach has much more generality and can routinely be extended to other situations in which the distribution of the noise can be represented as a scale mixture of normals; it is in fact sufficient to modify the distribution of the scaling factor λ. Distributions that can be expressed as scale mixtures of normals include the logistic, Student's t and power exponential [37]. In particular, the Student's t and the power exponential distribution are especially appreciated in the setting of noise modelling and we will present here for reference the densities that should be employed for the scale factor λ.

If the noise has t distribution with ν degrees of freedom, scale parameter σ and location parameter μ

$$\epsilon_i \sim t(\nu, \mu, \sigma),$$

the scaling factor has inverse gamma distribution with shape parameter $\nu - \frac{1}{2}$ and scale parameter 2 [21]:

$$\epsilon_i = \mu + \sigma\sqrt{\lambda_i}u_i, \qquad \lambda_i \sim \mathcal{I}g\left(\nu - \tfrac{1}{2}, 2\right),\ u_i \sim \mathcal{N}(0,1).$$

The (standardized) power exponential distribution, sometimes referred to as generalized error distribution (GED), has probability density function

$$f(x) \propto \exp\left(|x|^{-\alpha}\right),$$

with $\alpha \in [1, 2]$; the case $\alpha = 2$ obviously corresponds to a Gaussian distribution, and $\alpha = 1$ to a Laplace, or double exponential, distribution. If ϵ_i has power exponential distribution with parameters α, μ and σ, the scaling factor can be shown [37] to have density

$$p(\lambda_i) \propto \lambda_i^{-2} s\left(\lambda_i^{-2}; \tfrac{\alpha}{2}, 1\right),$$

where $s(\cdot; \alpha, \beta)$ denotes the probability density function of a standard stable distribution with tail parameter α and asymmetry parameter β. Although this density cannot be expressed in closed form, simulated values can again be readily obtained using the approach of [13].

In general, t distributions and power exponentials are far more popular than the α-stable for heavy tailed modelling purposes; in our opinion this is mainly because of their simplicity. However, as we have observed in section 2, the α-stable distribution fits certain data much better than the Student's t. Moreover, in our framework the α-stable and GED models will involve approximately the same computational burden as that for the (apparently simpler) Student's t case, since the generation of stable law random numbers takes roughly the same magnitude of computation time as that needed to produce inverse gamma distributed random numbers.

To conclude, we have presented practical Monte Carlo methods for on-line estimation of TVAR models in the presence of α-stable noise. The methods are accurately able to infer the signal state as well as unknown parameters, including the challenging α parameter of the stable distribution. Results so far are promising both for simulated data and for an audio processing example.

References

[1] C. Nikias and M. Shao, *Signal Processing with α-Stable Distributions and Applications.* John Wiley and Sons, New York, 1995.

[2] E. Kuruoglu, “Signal processing in α-stable environments: a least L^p-norm approach,” Ph.D. dissertation, Department of Engineering, University of Cambridge, 1998.

[3] B. Gnedenko and A. Kolmogorov, *Limit Distributions for Sums of Independent Random Variables.* Addison-Wesley, Reading, 1954.

[4] G. Samorodnitsky and M. Taqqu, *Stable Non-Gaussian Random Processes.* Chapman & Hall, Boca Raton, 1994.

[5] E. Tsionas, “Monte Carlo inference in econometric models with symmetric stable distributions,” *Journal of Econometrics*, vol. 88, pp. 365–401, 1999.

[6] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings-F*, vol. 140, pp. 107–113, 1993.

[7] G. Kitagawa, “Sequential Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[8] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[9] S. Godsill and T. Clapp, “Improvement strategies for Monte Carlo particle filters,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. de Freitas, and N. Gordon, Eds. Springer-Verlag, New York, 2001.

[10] W. Fong, A. Doucet, S. Godsill, and M. West, “Monte Carlo smoothing with application to speech enhancement,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 438–449, 2002.

[11] J. Vermaak, C. Andrieu, A. Doucet, and S. Godsill, “Particle methods for Bayesian modelling and enhancement of speech signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 173–185, 2002.

[12] S. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for non-linear time series," *Journal of the American Statistical Association*, vol. 50, pp. 438–449, 2004.

[13] J. Chambers, C. Mallows, and B. Stuck, "A method for simulating stable random variables," *Journal of the American Statistical Association*, vol. 71, pp. 340–344, 1976.

[14] S. Godsill, "MCMC and EM-based methods for inference in heavy-tailed processes with α-stable innovations," in *Proceedings of the IEEE Signal Processing Workshop on Higher-order Statistics*, 1999.

[15] S. Godsill and E. Kuruoglu, "Bayesian inference for time series with heavy-tailed symmetric α-stable noise processes," Department of Engineering, University of Cambridge, Tech. Rep. CUED/F-INFENG/TR.473, 1999.

[16] J. McCulloch, "Simple consistent estimators of stable distribution parameters," *Communications in Statistics – Simulation and Computation*, vol. 15, pp. 1109–1136, 1986.

[17] S. Mittnik, S. Rachev, T. Doganoglu, and D. Chenyao, "Maximum likelihood estimation of stable Paretian models," *Mathematical and Computer Modelling*, vol. 29, pp. 275–293, 1999.

[18] J. Nolan, "Numerical computation of stable densities and distribution functions," *Communications in Statistics – Stochastic Models*, vol. 13, pp. 759–774, 1997.

[19] E. Kuruoglu, W. Fitzgerald, and P. Rayner, "Near optimal detection of signals in impulsive noise modelled with a symmetric α-stable distribution," *IEEE Communications Letters*, vol. 10, pp. 282–284, 1998.

[20] D. Buckle, "Bayesian inference for stable distributions," *Journal of the American Statistical Association*, vol. 90, pp. 605–613, 1995.

[21] D. Andrews and C. Mallows, "Scale mixtures of normal distributions," *Journal of the Royal Statistical Society B*, vol. 36, pp. 99–102, 1974.

[22] E. Kuruoglu, C. Molina, and W. Fitzgerald, "Approximation of α-stable probability densities using finite mixtures of Gaussians," in *Proceedings of the 9th European Signal Processing Conference*, A. Theodoridis, I. Pitas, A. Stouraitis, and N. Kalouptsidis, Eds. Typorama, Rhodes, 1998.

[23] S. Godsill and P. Rayner, *Digital Audio Restoration.* Springer, Berlin, 1998.

[24] M. Lombardi, "Bayesian inference for α-stable distributions: A random walk MCMC approach," 2004, Dipartimento di Statistica "G. Parenti", Università degli studi di Firenze.

[25] P. Ha and S. Ann, "Robust time-varying parametric modelling of voiced speech," *Signal Processing*, vol. 42, pp. 311–317, 1995.

[26] M. Niedźwiecki and K. Cisowski, "Adaptive scheme for elimination of broadband noise and impulsive disturbances from AR and ARMA signals," *IEEE Transactions on Signal Processing*, vol. 44, pp. 528–537, 1996.

[27] A. Harvey, *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge, 1989.

[28] M. West and J. Harrison, *Bayesian forecasting and dynamic models*. Springer, New York, 1997.

[29] G. Kitagawa and W. Gersch, *Smoothness Prior Analysis of Time Series*. Springer, New York, 1996.

[30] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer, New York, 2001.

[31] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[32] W. Gilks and C. Berzuini, "Following a moving target: Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society B*, vol. 63, pp. 127–146, 2001.

[33] G. Storvik, "Particle filters for state space models in the presenc of unknown static parameters," *IEEE Transactions on Signal Processing*, vol. 50, pp. 281–289, 2002.

[34] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, pp. 539–551, 2002.

[35] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

[36] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. de Freitas, and N. Gordon, Eds. Springer-Verlag, New York, 2001.

[37] M. West, "On scale mixtures of normal distributions," *Biometrika*, vol. 74, pp. 646–648, 1987.