



# Simultaneous parameter estimation and state smoothing of complex GARCH process in the presence of additive noise

Saman Mousazadeh, Israel Cohen \*

Department of Electrical Engineering, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel

## ARTICLE INFO

### Article history:

Received 22 July 2009  
Received in revised form  
18 January 2010  
Accepted 23 April 2010  
Available online 7 May 2010

### Keywords:

GARCH  
Parameter estimation  
Noisy data  
Maximum likelihood  
Recursive maximum likelihood

## ABSTRACT

ARCH and GARCH models have been used recently in model-based signal processing applications, such as speech and sonar signal processing. In these applications, additive noise is often inevitable. Conventional methods for parameter estimation of ARCH and GARCH processes assume that the data are clean. The parameter estimation performance degrades greatly when the measurements are noisy. In this paper, we propose a new method for parameter estimation and state smoothing of complex GARCH processes in the presence of additive noise. Simulation results show the advantage of the proposed method in noisy environments.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

Parameter estimation is the backbone of all model-based signal processing algorithms. Having variability clustering and heavy tail property, autoregressive conditional heteroscedasticity (ARCH) and generalized ARCH (GARCH) models have recently been used in speech signal modeling. In speech enhancement in the short time Fourier transform (STFT) domain, the most important quantity which is to be estimated correctly is the spectral variance, since having the true value of this quantity, one can use the minimum mean square error (MMSE) [1] or log spectral amplitude (LSA) [2] estimators, which are optimum in MMSE and LSA sense, respectively, in order to estimate the speech spectral component. Cohen [3] modeled the speech signal in the STFT domain as a complex GARCH process and used this model for speech enhancement. He showed that the time varying variance of the speech signal can be estimated using a GARCH

model with Gaussian innovations. Furthermore, decision-directed method of Ephraim and Malah [1] can be derived using GARCH modeling. In his work, the GARCH model is used to estimate the spectral variance of the speech signals. Tahmasbi et al. [4] used the GARCH model for the voice activity detection (VAD). Abdolahi et al. [5] used the parameters of the GARCH model for speech recognition in Persian isolated digits. In all of the above-mentioned applications, it is assumed that the parameters of the GARCH model are known or can be estimated from a clean database.

Parameter estimation methods for ARCH and GARCH models, such as quasi-maximum likelihood (QML) [6], two stage least squares (TSLS) [7], and constrained two stage least squares (CTSLS) [8], rely on the clean data. Since these models are mostly used in econometric where data can be assumed noise free, there was no effort to estimate the parameters of these models in presence of additive noise. In many speech signal processing algorithms, these methods are inefficient since the additive noise is inevitable. Hence, new methods for estimating ARCH and GARCH parameters in presence of noise are required. Poyiadjis et al. [9] introduced a method for ML parameter estimation of general state-space models using

\* Corresponding author.

E-mail addresses: [samanm@tx.technion.ac.il](mailto:samanm@tx.technion.ac.il) (S. Mousazadeh), [icohen@ee.technion.ac.il](mailto:icohen@ee.technion.ac.il) (I. Cohen).

particle methods. The drawback of their method is its very high computational load, which makes this approach inapplicable to speech signal processing.

In this paper, we propose a new ML based estimation method for estimating the parameters of GARCH model in presence of additive noise. We find the probability density function (pdf) of the observation data given the parameters and compute the maximum of this function. Specifically, we compute the gradient vector and the Hessian matrix of the probability density function of the data, and subsequently find the maximum of the probability density function numerically using the Newton method. The remainder of this paper is organized as follows. In Section 2, we formulate the problem. In Section 3, we introduce our method and an adaptive version thereof. In Section 4, we apply our method for hidden state estimation. Finally in Section 5, we examine the performance of our method using simulations.

## 2. Problem formulation

A complex GARCH process of order  $(p,q)$  is defined as follows:

$$\mathbf{x}_t = \sqrt{\lambda_{t|t-1}} \odot \mathbf{v}_t \quad (1)$$

$$\lambda_{t|t-1} = c_0 \mathbf{1} + \sum_{i=1}^p a_i \mathbf{x}_{t-i} \odot \mathbf{x}_{t-i}^* + \sum_{j=1}^q b_j \lambda_{t-j|t-j-1} \quad (2)$$

where  $\mathbf{x}_t$  is a  $k \times 1$  complex vector,  $\mathbf{1}$  denotes a vector of ones,  $\odot$  denotes a term-by-term multiplication,  $*$  denotes the conjugate transpose operator,  $k$  is the dimension of the model and  $\mathbf{v}_t$ 's are independent identically distributed complex Gaussian random vectors with zero mean and identity covariance matrix, i.e.

$$\mathbf{v}_t \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}) \quad (3)$$

An ARCH model is a special case of the GARCH model with  $b_j = 0 \forall j$ .

It is worth mentioning that this model is a special case of a more complicated general model in which the  $c_0$  is a  $k \times 1$  vector and  $a_i$ 's and  $b_j$ 's are  $k \times k$  matrices. Although it initially seems that the general model might be more successful in applications such as speech modeling (because it models the speech in each frequency bin using different GARCH models) one should take into account that the general model consists of  $k^2+k$  parameters with  $k=512$  or  $1024$  in speech processing applications. This large number of parameters makes the general model inapplicable in practice. Another issue which must be considered is that the process is really a vector process (not  $k$  separate scalar processes) because as we will see latter, all the vectors are combined with each other in order to jointly estimate the parameters using ML method.

Suppose that the observation signal is an ARCH or GARCH process corrupted with an additive zero-mean white complex Gaussian noise, i.e.

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{n}_t \quad (4)$$

$$\mathbf{n}_t \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}) \quad (5)$$

where  $\mathbf{n}_t$  is the additive noise sequence. We assume that the noise variance  $\sigma^2$  is unknown, and must be estimated from observations. Our objective is to find constrained estimates of the GARCH parameters and the noise variance (i.e. estimation of the vector of the parameters  $\theta = [c_0, a_1, \dots, a_p, b_1, \dots, b_q, \sigma^2]^T$ , under the stationarity conditions of the model). The stationarity conditions for the complex GARCH model are given by [3]

$$c_0 > 0 \quad (6)$$

$$a_1, a_2, \dots, a_p, b_1, b_2, \dots, b_p \geq 0 \quad (7)$$

$$\sum_{i=1}^p a_i + \sum_{j=1}^q b_j < 1 \quad (8)$$

## 3. ML estimation of the parameters

In order to find the maximum likelihood estimate of the parameters, we must compute the logarithm of the likelihood function of the observations. Let  $\mathbf{y}_0^\tau = \{\mathbf{y}_t | t=0, 1, 2, \dots, \tau\}$  be the set of observations up to sample number  $\tau$ . Using Bayes' formula, the logarithm of the likelihood function can be written as

$$\log[f(\mathbf{y}_0^{N-1}; \theta)] = \sum_{t=1}^{N-1} \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \theta)] + \log[f(\mathbf{y}_0; \theta)] \quad (9)$$

where  $N$  is the number of available observations. It should be noted that for a large sample data size the second term on the right side of (9) makes negligible contribution to the pdf and can be ignored. Let  $\mathbf{x}_0^\tau = \{\mathbf{x}_t | t=0, 1, 2, \dots, \tau\}$  be the clean signal up to sample number  $\tau$ . We denote the one-frame-ahead conditional variance of the clean signal as

$$\lambda_{t|t-1} = E\{\mathbf{x}_t \odot \mathbf{x}_t^* | \mathbf{x}_0^{t-1}; \theta\} \quad (10)$$

It is worth mentioning that in [3], it is shown that  $\lambda_{t|t-1}$  can be used as an estimate of the speech spectral variance. As in [3] we assume that the past estimated conditional variances are sufficient statistics for the conditional variance estimation, i.e.

$$E\{\mathbf{x}_t \odot \mathbf{x}_t^* | \mathbf{y}_0^t; \theta\} = E\{\mathbf{x}_t \odot \mathbf{x}_t^* | \hat{\lambda}_{t|t-1}, \mathbf{y}_t; \theta\} \quad (11)$$

Here, we assume that the order of the model is  $(1,1)$ . We choose this order because this order is mostly used in the speech processing applications. Note that the one-frame-ahead conditional variance  $\lambda_{t|t-1}$  can be estimated recursively in the minimum mean square error (MMSE) sense as follows:

$$\begin{aligned} \hat{\lambda}_{t|t-1} &= E\{\lambda_{t|t-1} | \mathbf{y}_0^{t-1}; \theta\} \\ &= E\{c_0 \mathbf{1} + a_1 \mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^* + b_1 \lambda_{t-1|t-2} | \mathbf{y}_0^{t-1}; \theta\} \\ &= c_0 \mathbf{1} + a_1 E\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^* | \mathbf{y}_0^{t-1}; \theta\} + b_1 E\{\lambda_{t-1|t-2} | \mathbf{y}_0^{t-1}; \theta\} \\ &= c_0 \mathbf{1} + a_1 E\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^* | \mathbf{y}_0^{t-1}; \theta\} + b_1 \hat{\lambda}_{t-1|t-2} \\ &= c_0 \mathbf{1} + a_1 E\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^* | \mathbf{y}_{t-1}, \hat{\lambda}_{t-1|t-2}; \theta\} + b_1 \hat{\lambda}_{t-1|t-2}. \end{aligned} \quad (12)$$

In order to compute  $E\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^* | \mathbf{y}_{t-1}, \hat{\lambda}_{t-1|t-2}; \theta\}$  note that this is the estimation of the *a posteriori* second order moment of vector  $\mathbf{x}_{t-1}$  with *a priori* known covariance, i.e.

$\hat{R}_x^{t-1} = \text{diag}(\hat{\lambda}_{t-1|t-2})$ . Using the fact that  $\mathbf{x}_{t-1}$  is a zero mean complex Gaussian random vector corrupted with a Gaussian noise we have [10, Eq. (14)]

$$\begin{aligned} \mathbf{u}_{t-1} &= E\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^* | \mathbf{y}_{t-1}, \hat{\lambda}_{t-1|t-2}; \boldsymbol{\theta}\} \\ &= \hat{R}_x^{t-1} (\hat{R}_x^{t-1} + \sigma^2 \mathbf{I})^{-1} [\text{diag}(\sigma^2 \mathbf{I}) \\ &\quad + \hat{R}_x^{t-1} (\hat{R}_x^{t-1} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_{t-1} \odot \mathbf{y}_{t-1}^*] \end{aligned} \quad (13)$$

In order to compute (9) note that

$$(\mathbf{x}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta}) \sim \mathcal{CN}(0, \text{diag}(\hat{\lambda}_{t|t-1})) \quad (14)$$

$$(\mathbf{n}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta}) \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}) \quad (15)$$

$$(\mathbf{y}_t = \mathbf{x}_t + \mathbf{n}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta}) \sim \mathcal{CN}(0, \sigma^2 \mathbf{I} + \text{diag}(\hat{\lambda}_{t|t-1})) \quad (16)$$

From (9)–(16) the log-likelihood function of the current observation conditioned on past observations, i.e.  $f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})$ , can be written as

$$\log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] = \log \left[ \frac{1}{\pi^k |\hat{R}_x^t + \sigma^2 \mathbf{I}|} \right] - \mathbf{y}_t^* (\hat{R}_x^t + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t \quad (17)$$

where  $|\cdot|$  is the determinant operation. The log-likelihood function of the observation can be computed using (9) and (17). To find the maximum of the log-likelihood function of the observation we must find the gradient and the Hessian matrix of this function.

The gradient of this function can be calculated by using

$$\begin{aligned} \mathbf{g} &= \nabla \log[f(\mathbf{y}_0^{N-1}; \boldsymbol{\theta})] = \nabla \sum_{t=1}^{N-1} \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] \\ &= \sum_{t=1}^{N-1} \nabla \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] \end{aligned} \quad (18)$$

In order to compute the gradient of the log-likelihood function of the current observation conditioned on past observations, i.e.  $\nabla \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})]$ , note that

$$\begin{aligned} \log \left[ \frac{1}{\pi^k |\hat{R}_x^t + \sigma^2 \mathbf{I}|} \right] &= -\log \pi^k - \log[|\hat{R}_x^t + \sigma^2 \mathbf{I}|] \\ &= -\log \pi^k - \log \left[ \prod_{i=1}^k (\hat{\lambda}_{i,t|t-1} + \sigma^2) \right] \\ &= -\log \pi^k - \sum_{i=1}^k \log[\hat{\lambda}_{i,t|t-1} + \sigma^2] \end{aligned} \quad (19)$$

and

$$\mathbf{y}_t^* (\hat{R}_x^t + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t = \sum_{i=1}^k \frac{y_{i,t} \mathbf{y}_{i,t}^*}{\hat{\lambda}_{i,t|t-1} + \sigma^2} \quad (20)$$

where  $y_{i,t}$  and  $\hat{\lambda}_{i,t|t-1}$  are the  $i$ -th elements of  $\mathbf{y}_t$ , and  $\hat{\lambda}_{i,t|t-1}$ , respectively. The gradient of the log-likelihood function of the current observation conditioned on past observations, can be computed as follows:

$$\frac{\partial}{\partial c_0} \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] = - \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^2} \quad (21)$$

$$\frac{\partial}{\partial a_1} \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] = - \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^2} u_{i,t-1} \quad (22)$$

$$\frac{\partial}{\partial b_1} \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] = - \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^2} \hat{\lambda}_{i,t-1|t-2} \quad (23)$$

$$\frac{\partial}{\partial \sigma^2} \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] = - \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^2} \left( 1 + a_1 \frac{\partial u_{i,t-1}}{\partial \sigma^2} \right) \quad (24)$$

where  $u_{i,t-1}$  is the  $i$ -th element of  $\mathbf{u}_{t-1}$ ,  $\hat{\lambda}_{i,t-1|t-2}$  and  $u_{i,t-1}$  can be evaluated by (12) and (13), respectively, and the partial derivative of  $u_{i,t-1}$  with respect to  $\sigma^2$  can be computed as follows:

$$\frac{\partial u_{i,t-1}}{\partial \sigma^2} = \frac{\hat{\lambda}_{i,t-1|t-2} (\hat{\lambda}_{i,t-1|t-2} + \sigma^2 - 2y_{i,t-1} \mathbf{y}_{i,t-1}^*)}{(\hat{\lambda}_{i,t-1|t-2} + \sigma^2)^3} \quad (25)$$

The Hessian matrix can be computed as follows:

$$\begin{aligned} \mathbf{H} &= \nabla^2 \log[f(\mathbf{y}_0^{N-1}; \boldsymbol{\theta})] = \nabla^2 \sum_{t=1}^{N-1} \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] \\ &= \sum_{t=1}^{N-1} \nabla^2 \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] = \sum_{t=1}^{N-1} \nabla^2 \mathbf{H}_t \end{aligned} \quad (26)$$

where the second partial derivative of the log-likelihood function of the current observation conditioned on the past observations, i.e.  $\mathbf{H}_t = \nabla^2 \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})]$ , can be computed as follows:

$$\mathbf{H}_t = \begin{bmatrix} \frac{\partial^2(\gamma)}{\partial c_0^2} & \frac{\partial^2(\gamma)}{\partial c_0 \partial a_1} & \frac{\partial^2(\gamma)}{\partial c_0 \partial b_1} & \frac{\partial^2(\gamma)}{\partial c_0 \partial \sigma^2} \\ \frac{\partial^2(\gamma)}{\partial a_1 \partial c_0} & \frac{\partial^2(\gamma)}{\partial a_1^2} & \frac{\partial^2(\gamma)}{\partial a_1 \partial b_1} & \frac{\partial^2(\gamma)}{\partial a_1 \partial \sigma^2} \\ \frac{\partial^2(\gamma)}{\partial b_1 \partial c_0} & \frac{\partial^2(\gamma)}{\partial b_1 \partial a_1} & \frac{\partial^2(\gamma)}{\partial b_1^2} & \frac{\partial^2(\gamma)}{\partial b_1 \partial \sigma^2} \\ \frac{\partial^2(\gamma)}{\partial \sigma^2 \partial c_0} & \frac{\partial^2(\gamma)}{\partial \sigma^2 \partial a_1} & \frac{\partial^2(\gamma)}{\partial \sigma^2 \partial b_1} & \frac{\partial^2(\gamma)}{\partial (\sigma^2)^2} \end{bmatrix} \quad (27)$$

where

$$\gamma = \log[f(\mathbf{y}_t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] \quad (28)$$

$$\frac{\partial^2(\gamma)}{\partial (c_0^2)^2} = \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} \quad (29)$$

$$\frac{\partial^2(\gamma)}{\partial c_0 \partial a_1} = \frac{\partial^2(\gamma)}{\partial a_1 \partial c_0} = \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} u_{i,t-1} \quad (30)$$

$$\frac{\partial^2(\gamma)}{\partial c_0 \partial b_1} = \frac{\partial^2(\gamma)}{\partial b_1 \partial c_0} = \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} \hat{\lambda}_{i,t-1|t-2} \quad (31)$$

$$\frac{\partial^2(\gamma)}{\partial c_0 \partial \sigma^2} = \frac{\partial^2(\gamma)}{\partial \sigma^2 \partial c_0} = \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} \left( 1 + a_1 \frac{\partial u_{i,t-1}}{\partial \sigma^2} \right) \quad (32)$$

$$\frac{\partial^2(\gamma)}{\partial a_1^2} = \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} u_{i,t-1}^2 \quad (33)$$

$$\frac{\partial^2(\gamma)}{\partial a_1 \partial b_1} = \frac{\partial^2(\gamma)}{\partial b_1 \partial a_1} = \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t} \mathbf{y}_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} \hat{\lambda}_{i,t-1|t-2} u_{i,t-1} \quad (34)$$

$$\begin{aligned} \frac{\partial^2(\gamma)}{\partial a_1 \partial \sigma^2} &= \frac{\partial^2(\gamma)}{\partial \sigma^2 \partial a_1} \\ &= \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t}y_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} \left(1 + a_1 \frac{\partial u_{i,t-1}}{\partial \sigma^2}\right) u_{i,t-1} \\ &\quad - \sum_{i=1}^k \frac{\partial u_{i,t-1}}{\partial \sigma^2} \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - y_{i,t}y_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^2} \end{aligned} \quad (35)$$

$$\begin{aligned} \frac{\partial^2(\gamma)}{\partial b_1 \partial \sigma^2} &= \frac{\partial^2(\gamma)}{\partial \sigma^2 \partial b_1} \\ &= \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t}y_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} \left(1 + a_1 \frac{\partial u_{i,t-1}}{\partial \sigma^2}\right) \hat{\lambda}_{i,t-1|t-2} \end{aligned} \quad (36)$$

$$\begin{aligned} \frac{\partial^2(\gamma)}{\partial (\sigma^2)^2} &= \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - 2y_{i,t}y_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^3} \left(1 + a_1 \frac{\partial u_{i,t-1}}{\partial \sigma^2}\right) \\ &\quad - a_1 \sum_{i=1}^k \frac{\hat{\lambda}_{i,t|t-1} + \sigma^2 - y_{i,t}y_{i,t}^*}{(\hat{\lambda}_{i,t|t-1} + \sigma^2)^2} \frac{\partial^2 u_{i,t-1}}{\partial (\sigma^2)^2} \end{aligned} \quad (37)$$

$$\frac{\partial^2 u_{i,t-1}}{\partial (\sigma^2)^2} = -2 \frac{\hat{\lambda}_{i,t-1|t-2} (\hat{\lambda}_{i,t-1|t-2} + \sigma^2 - 3y_{i,t-1}y_{i,t-1}^*)}{(\hat{\lambda}_{i,t-1|t-2} + \sigma^2)^4} \quad (38)$$

With these values of the gradient vector and Hessian matrix, we can use numerical optimization methods such as Newton [11] or gradient projection method [11] to find the maximum of the log-likelihood function with or without considering stationarity conditions. The algorithm for estimating the parameters without considering the stationarity conditions is summarized in Table 1.

In order to introduce the algorithm for estimating the parameters under the stationarity conditions, note that the constraints in (8) and positiveness of  $\sigma^2$  can be written as  $\mathbf{M}\boldsymbol{\theta} \leq \boldsymbol{\mu}$  where  $\mathbf{M} = [-\mathbf{I}_{4 \times 4}, [0, 1, 1, 0]^T]^T$ ,  $\boldsymbol{\mu} = [0, 0, 0, 0, 1]^T$  and  $\mathbf{a} \leq \mathbf{b}$  means that each element of the vector  $\mathbf{a}$  is less than or equal to the corresponding element in the vector  $\mathbf{b}$ . Our final algorithm is summarized in Table 2. In this algorithm,  $\mathbf{M}_q$  is constructed from  $\mathbf{M}$  by eliminating the rows corresponding to inactive constraints.

It is worth mentioning that if the covariance matrix of the additive noise is known, i.e. the signal to noise ratio (SNR) is known *a priori*, the proposed method can be used in the same manner by eliminating the last element of the gradient vector and the last column and the last row of the Hessian matrix.

**Table 1**  
Algorithm for ML parameter estimation of noisy complex GARCH model without considering stationarity conditions.

<p><i>Initialization:</i> Choose <math>\boldsymbol{\theta}_0</math>; Let <math>\mathbf{u}_1 = \hat{\lambda}_{1 0} = \mathbf{0}_{k \times 1}</math> and <math>l = 0</math>; while <math>\mathbf{g} &gt; 0</math>   Compute <math>\mathbf{u}_t</math> for <math>t=1, 2, \dots, N-1</math> using (13);   Compute <math>\hat{\lambda}_{t t-1}</math> for <math>t=1, 2, \dots, N-1</math> using (12);   Compute <math>\mathbf{g}</math> and <math>\mathbf{H}</math> using (18) and (26), respectively.   <math>\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \gamma \mathbf{H}^{-1} \mathbf{g}</math>;   <math>l = l + 1</math>; end (while)</p>
---

**Table 2**

Algorithm for ML parameter estimation of noisy complex GARCH model, under stationarity conditions.

<p><i>Initialization:</i> Chose <math>\boldsymbol{\theta}_0</math> which satisfies the stationarity conditions Let <math>\mathbf{u}_1 = \hat{\lambda}_{1 0} = 0</math>; 1) Find the subspace of active constraints and form <math>\mathbf{M}_q</math>. 2) Calculate <math>\mathbf{P} = \mathbf{I} - \mathbf{M}_q^T (\mathbf{M}_q \mathbf{M}_q^T)^{-1} \mathbf{M}_q</math>, and <math>\mathbf{d} = \mathbf{P} \mathbf{g}</math> using (18)–(25). 3) If <math>\mathbf{d} \neq \mathbf{0}</math>, find <math>\rho_1</math> and <math>\rho_2</math> satisfying   <math>\max \{\rho_1 : \boldsymbol{\theta} + \rho_1 \mathbf{d} \text{ is feasible}\}</math>   <math>\max \{\log(f(\mathbf{y}_n^N; \boldsymbol{\theta} + \rho_2 \mathbf{d})) : 0 \leq \rho_2 \leq \rho_1\}</math>   using (9). Set <math>\boldsymbol{\theta}</math> to <math>\boldsymbol{\theta} + \rho_2 \mathbf{d}</math>. 5) If <math>\mathbf{d} = \mathbf{0}</math>, find <math>\lambda = (\mathbf{M}_q \mathbf{M}_q^T)^{-1} \mathbf{M}_q \mathbf{g}</math>.   (a) If <math>\lambda_j \geq 0</math> for all <math>j</math> corresponding to active inequalities, stop;   <math>\boldsymbol{\theta}</math> satisfies the Kuhn–Tucker conditions.   (b) Otherwise, delete the row from <math>\mathbf{M}_q</math>, corresponding to the inequality with the most negative component of <math>\lambda</math> and return to 2.</p>
--

Another issue that must be addressed here is the case of additive colored noise. Suppose that  $n_t \sim \mathcal{CN}(\mathbf{0}, \boldsymbol{\Sigma}_n)$  where  $\boldsymbol{\Sigma}_n$  is a *known* matrix with  $\sigma^2$  on its diagonal. The off-diagonal entries are assumed to be zero as in all noise modeling procedures in the STFT domain for speech signal processing. In this case the terms  $\sigma^2 \mathbf{I}$  and  $\hat{\lambda}_{i,t|t-1} + \sigma^2$  in the above formulas must be replaced by  $\boldsymbol{\Sigma}_n$  and  $\hat{\lambda}_{i,t|t-1} + \sigma_i^2$ , respectively.

### 3.1. Recursive ML

A standard approach to recursive ML (RML) estimation considers a series of log-likelihood functions  $\log[f(\mathbf{y}_0^k; \boldsymbol{\theta})]$ , where  $\log[f(\mathbf{y}_0^k; \boldsymbol{\theta})] = \sum_{t=1}^k \log[f(\mathbf{y}^t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})]$  [12]. Under suitable regularity conditions described in [13] it can be shown that the average log-likelihood converges to the following limit:

$$l(\boldsymbol{\theta}) = \lim_{k \rightarrow \infty} \frac{1}{k+1} \sum_{t=1}^k \log[f(\mathbf{y}^t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})] \quad (39)$$

It can be shown that  $l(\boldsymbol{\theta})$  admits  $\boldsymbol{\theta}^*$  as a global maximum where  $\boldsymbol{\theta}^*$  is the global maximum of the log-likelihood function [12]. The function  $l(\boldsymbol{\theta})$  does not have an analytical expression and we do not have access to it. Nevertheless, identification of  $\boldsymbol{\theta}^*$  can still be achieved based on the ergodicity property in (39), which provides us with a set of accessible functions  $f(\mathbf{y}^t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta})$  that converge to  $l(\boldsymbol{\theta})$ . One way to exploit this in order to maximize  $l(\boldsymbol{\theta})$ , is to use a stochastic approximation (SA) algorithm to update the parameter estimate at time  $n$  using the recursion

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \gamma_n \nabla \log[f(\mathbf{y}_n | \mathbf{y}_0^{n-1}; \boldsymbol{\theta})] \quad (40)$$

where  $\boldsymbol{\theta}_{n-1}$  is the parameter estimate at time  $n-1$  and  $\nabla \log[f(\mathbf{y}_n | \mathbf{y}_0^{n-1}; \boldsymbol{\theta})]$  can be computed by (21)–(23), provided that the step size  $\gamma_n$  is a positive non-increasing sequence, such that  $\sum_{n=1}^{\infty} \gamma_n = \infty$  and  $\sum_{n=1}^{\infty} \gamma_n^2 < \infty$ . It can be shown that  $\boldsymbol{\theta}_n$  will converge to the set of (global or local) maxima of  $l(\boldsymbol{\theta})$ . It is worth mentioning that this method (RML) can be used for adaptive parameter estimation (varying parameters). In order to achieve this goal one should use the RML with constant step size (i.e.  $\gamma_n = \gamma$ ). The choice of  $\gamma$  will be a trade-off between

tracking capability (large  $\gamma$ ) and low estimation noise around the parameter (small  $\gamma$ ).

**4. State smoothing**

After the estimation of the parameters of the complex GARCH model these estimated parameters can be used to estimate the hidden state of the model,  $\mathbf{x}_t$ , in the MMSE sense. In order to do this, note that the MMSE optimal estimator in a Bayesian framework is the conditional mean

$$\hat{\mathbf{x}}_t = E\{\mathbf{x}_t | \mathbf{y}_0^t\} = E\{\mathbf{x}_t | \hat{\lambda}_{t|t-1}, \mathbf{y}_t\} \tag{41}$$

Because  $\mathbf{x}_t$  conditioned on  $\hat{\lambda}_{t|t-1}$  and  $\mathbf{y}_t$  is a zero mean complex Gaussian vector with covariance matrix equals to  $\text{diag}(\hat{\lambda}_{t|t-1})$ , the solution of the conditional expectation in (41) is the well-known Wiener filter. Hence, the conditional mean in (41) can be calculated using (12) and (13), the estimated parameters (i.e.  $\hat{\theta} = [\hat{c}_0, \hat{a}_1, \hat{b}_1, \hat{\sigma}^2]^T$ ), and the following equation:

$$\hat{\mathbf{x}}_t = \hat{R}_x^t (\hat{R}_x^t + \hat{\sigma}^2 \mathbf{I})^{-1} \mathbf{y}_t \tag{42}$$

**5. Simulations**

In this section, we analyze the performance of our proposed method using three different experiments. In the first experiment, we use a two dimensional ( $k=2$ ) GARCH(1,1) process corrupted with zero-mean complex Gaussian white noise with three different signal to noise ratios (SNR, i.e.  $-10, -5$  and  $0$  dB). The number of available data was set to  $N=1000$  samples. The process  $\mathbf{v}_t$  is zero-mean Gaussian white process with identity covariance matrix. The parameters of the GARCH(1,1) are  $\theta = [0.10, 0.40, 0.30]^T$ . We compare the performance of the proposed method with that of two different ML methods. The first method denoted by MLClean, employs the clean data (unavailable in practical situations) for estimating the parameters. The second method denoted by MLNoisy, employs the noisy data for estimating the parameters assuming that the data are clean. The proposed method is examined under two different situations. The first situation denoted by PUN, employs the noisy data for estimating the parameters assuming that the corrupting noise variance  $\sigma^2$  is unknown. The second situation denoted by PKN, employs noisy data for estimating the parameter assuming that the corrupting noise variance is known.

For evaluating the performance of our method we used the mean square error (MSE) in parameter estimation and the output SNR of the proposed state smoothing algorithm which both are estimated using 2000 realizations. The input SNR ( $\text{SNR}_{\text{in}}$ ) and the output SNR ( $\text{SNR}_{\text{out}}$ ) are defined as follows:

$$\begin{aligned} \text{SNR}_{\text{in}} &= 10 \log_{10} \frac{1}{2000} \sum_{i=1}^{2000} \frac{\sum_{t=1}^N \|\mathbf{x}_t^i\|^2}{\sum_{t=1}^N \|\mathbf{x}_t^i - \mathbf{y}_t^i\|^2} \\ \text{SNR}_{\text{out}} &= 10 \log_{10} \frac{1}{2000} \sum_{i=1}^{2000} \frac{\sum_{t=1}^N \|\mathbf{x}_t^i\|^2}{\sum_{t=1}^N \|\mathbf{x}_t^i - \hat{\mathbf{x}}_t^i\|^2} \end{aligned} \tag{43}$$

**Table 3**

MSE in parameter estimation for different methods ( $\text{SNR}_{\text{in}} = -10$  dB).

MSE	MLClean	MLNoisy	PKN	PUN	MMSEE
$c_0$	0.0003	13.9474	0.0196	7.0087	–
$a_1$	0.0010	0.1506	0.1161	0.1385	–
$b_1$	0.0024	0.1258	0.0530	0.0901	–
$\text{SNR}_{\text{out}}$	0.3656	–4.8379	0.3357	–3.5234	0.4991

**Table 4**

MSE in parameter estimation for different methods ( $\text{SNR}_{\text{in}} = -5$  dB).

MSE	MLClean	MLNoisy	PKN	PUN	MMSEE
$c_0$	0.0003	0.8057	0.0051	0.3590	–
$a_1$	0.0010	0.1314	0.0628	0.0881	–
$b_1$	0.0024	0.1595	0.0830	0.0931	–
$\text{SNR}_{\text{out}}$	1.2229	–0.7952	1.1940	0.1835	1.4405

**Table 5**

MSE in parameter estimation for different methods ( $\text{SNR}_{\text{in}} = 0.0$  dB).

MSE	MLClean	MLNoisy	PKN	PUN	MMSEE
$c_0$	0.0003	0.0316	0.0009	0.0223	–
$a_1$	0.0010	0.0740	0.0171	0.0513	–
$b_1$	0.0024	0.0401	0.0278	0.0387	–
$\text{SNR}_{\text{out}}$	3.1225	2.6878	3.1128	2.8427	3.3009

where  $\mathbf{x}_t^i$ ,  $\mathbf{x}_t^i$  and  $\hat{\mathbf{x}}_t^i$  are the clean, noisy and estimated signals in the  $i$ -th realization, respectively. For each previously mentioned parameter estimation methods  $\hat{\mathbf{x}}_t^i$  is obtained using the proposed state smoothing algorithm assuming that the corrupting noise variance is known. The results are depicted in Tables 3–5. In the last columns of these tables we also provide the output SNR for minimum mean square error estimator (MMSEE). This estimator uses the true one-frame-ahead conditional variance of the clean signal ( $\lambda_{t|t-1}$ ) for state smoothing (i.e. the MMSEE estimator uses ( $\lambda_{t|t-1}$ ) instead of its estimate ( $\hat{\lambda}_{t|t-1}$ ) in (42)). Note that the output SNR given in this column is the highest achievable SNR. It is obvious that the proposed methods yield better performance in output SNR and parameter estimation over the MLNoisy method which is often used in real world problems. The best performance is obtained by using the MLClean, as expected, but this method cannot be used in practical situations because the clean data are not available. Comparing the output SNR of PKN and MMSEE shows that the performance of the proposed parameter estimation method is very close to the best achievable performance (i.e. MMSEE). Another point which must be emphasized here is that the performance of the proposed method degrades if the noise variance is unknown. This is because of the additional information employed in PKN method. Another issue which must be mentioned here is that estimating  $b_1$  is much more difficult than  $c_0$  and  $a_1$  because  $b_1$  is multiplied by a hidden quantity  $\lambda_{t|t-1}$  which makes the estimation of this parameter difficult. The simulation results confirm this fact. Other simulation results show

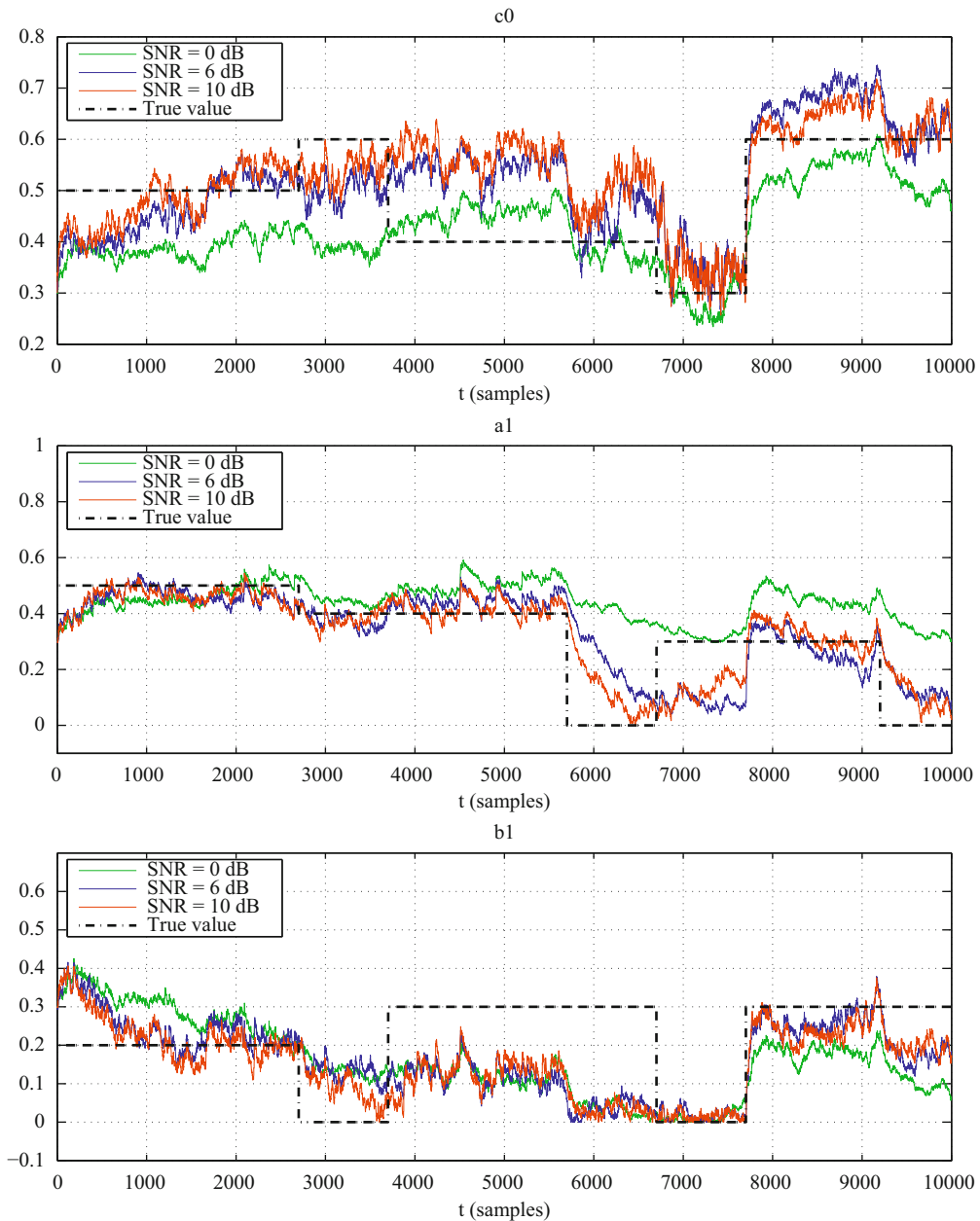
that the performance of MLClean, MLNoisy, PUN and PKN methods is almost the same for high values of SNR (greater than 5 dB).

**Table 6**  
MSE in parameter estimation for different methods (colored noise) ( $\text{SNR}_{\text{in}} = -4.44$  dB).

MSE	MLClean	MLNoisy	PKN	MMSEE
$c_0$	0.0003	0.0034	0.0014	–
$a_1$	0.0020	0.0529	0.0145	–
$b_1$	0.0047	0.2358	0.0244	–
$\text{SNR}_{\text{out}}$	1.5935	0.9356	1.5604	1.7905

In the second experiment we investigate the influence of the colored noise on the proposed algorithm. In the text we addressed how to treat the colored noise. For this experiment we used the same GARCH model as expressed in the first simulation but the process was corrupted by a zero-mean Gaussian colored noise with covariance matrix equals to  $\Sigma_n = \text{diag}(1, 0.16)$ . By this choice of the corrupting noise covariance matrix the input SNR equals to  $-4.44$  dB. The results are depicted in Table 6. It is apparent that the proposed method outperforms the MLNoisy method.

In the third experiment, we try to estimate the time varying parameters of a two dimensional complex GARCH



**Fig. 1.** The results of the RML method in estimating time varying parameters for three different SNRs.

**Table 7**

MSE in parameter estimation for RML methods with three different SNR.

MSE	SNR=0 dB	SNR=6 dB	SNR=10 dB
$c_0$	0.0190	0.0075	0.0065
$a_1$	0.0245	0.0147	0.0100
$b_1$	0.0143	0.0138	0.0114

process corrupted with zero-mean complex Gaussian white process with known covariance using the proposed RML method. The performance of the RML method is analyzed in three different SNRs, i.e. 0, 6 and 10 dB. The true and estimated values of the time varying parameters are depicted in Fig. 1. It is obvious that the RML algorithm can follow time varying parameters and its performance gets better as the SNR increases. The MSE in parameter estimation presented in Table 7, is estimated by

$$\text{MSE}_i = \frac{1}{10^4} \sum_{t=1}^{10^4} (\theta_i(t) - \hat{\theta}_i(t))^2 \quad (44)$$

where  $\theta_i$ ,  $\hat{\theta}_i$  and  $\text{MSE}_i$  are  $i$ -th element of  $\theta$ , its estimate and mean square error in its estimation, respectively. For a practical application of the RML parameter estimation in speech processing see [14].

### 5.1. Computational complexity

The computational complexities of the MLNoisy and the MLClean methods, which are often used in practice, are the same since these methods only use different data. The computational complexity of the proposed method roughly equals to that of MLNoisy (or MLClean) method. The reason is that all these methods are using the same algorithm, i.e. gradient projection method for maximization of the pdf on the parameters. Assuming that multiplications and divisions have the same complexity for the processor, the number of sums and multiplications for computing the pdf function in each iteration of the gradient projection method for the MLClean method are 3 and 6 kN, respectively. For the proposed method the number of sums is 6 kN and the number of multiplications is 9 kN, respectively. Increasing the order of the model increases the computational load but the computational load of parameter estimation of higher order models is still  $O(N)$ . It should be mentioned that the method presented in [9] takes approximately 35 h to estimate the parameters of one process while the proposed method just needs 23 s (for the process used in the first simulation). This fact shows that the method presented in [9], even if it has better performance over the purposed method, cannot be applied in practical applications.

## 6. Conclusion

We have presented a new ML estimation procedure for parameter estimation of complex GARCH(1,1) model in presence of the additive noise, and a new procedure for estimating the hidden states of the model in the MMSE sense. The method computes the log-likelihood function

together with its gradient vector and Hessian matrix, and uses the Newton method (or gradient projection method under stationarity conditions) to numerically find the maximum of the log-likelihood function. We also presented an RML method for estimating the time varying parameters. Simulation results show improvement with respect to not considering the noise effect in the estimation of the parameters. In this work we considered two situations. The first one is the general case where the SNR is unknown. In this situation, the proposed method estimates the variance of the noise together with the parameters and the hidden states of the model. The second situation, which is a special case of the first one, assumes that the variance of the additive noise is known and just estimates the parameters of the model. As expected, the second case yields better performance over the first one because of the additional information employed. The issue how to handle colored noise is also discussed in the paper.

## Acknowledgements

The authors thank the anonymous reviewers for their helpful comments.

## References

- [1] Y. Ephraim, D. Malah, Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator, *IEEE Trans. Acoust. Speech Signal Process.* ASSP-32 (6) (December 1984) 1109–1121.
- [2] Y. Ephraim, D. Malah, Speech enhancement using a minimum mean-square error log-spectral amplitude estimator, *IEEE Trans. Acoust. Speech Signal Process.* ASSP-32 (2) (April 1985) 443–445.
- [3] I. Cohen, Speech spectral modeling and enhancement based on autoregressive conditional heteroscedasticity models, *Signal Process.* 86 (2006) 698–709.
- [4] R. Tahmasbi, S. Rezaei, A soft voice activity detection using GARCH filter and variance Gamma distribution, *IEEE Trans. Audio Speech Lang. Process.* 5 (2007) 1129–1134.
- [5] M. Abdollahi, H. Amindavar, GARCH coefficients as feature for speech recognition in Persian isolated digit, in: *Proceedings of 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, Philadelphia, 18–23 March 2005, pp. 1-957–1-960.
- [6] A.A. Weiss, Asymptotic theory of ARCH models estimation and testing, *Econometric Theory* 2 (1989) 107–131.
- [7] A. Bose, K. Mukherjee, Estimating the ARCH parameters by solving linear equations, *J. Time Ser. Anal.* 24 (2003) 127–136.
- [8] S. Mousazadeh, M. Karimi, M. Farokhrooz, ARCH parameter estimation via constrained two stage least squares method, in: *Proceedings of Ninth IEEE International Symposium on Signal Processing and its Applications (ISSPA)*, IEEE, Sharjah, United Arab Emirates, 12–15 February 2007, pp. 1–4.
- [9] G. Poyiadjis, A. Doucet, S.S. Singh, Particle methods for optimal filter derivative: application to parameter estimation, in: *Proceedings of 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, Philadelphia, 18–23 March 2005, pp. 1-925–1-928.
- [10] A. Abramson, I. Cohen, State smoothing in Markov-switching time-frequency GARCH models, *IEEE Signal Process. Lett.* 13 (2006) 377–380.
- [11] D.G. Luenberger, *Linear and Nonlinear Programming*, Kluwer Academic Publishers Group, The Netherlands, 2003.
- [12] L. Ljung, T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, 1983.
- [13] V. Tadic, A. Doucet, Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models, *Stochastic Processes Appl.* 115 (2005) 1408–1436.
- [14] S. Mousazadeh, I. Cohen, AR-GARCH in presence of noise: parameter estimation and its application to voice activity detection, *IEEE Trans. Audio Speech Lang. Process.*, submitted for publication.