Yugoslav Journal of Operations Research
3 (1993), Number 1, 33–42

ESTIMATING PARAMETERS OF A MULTIPLICATIVE SEASONAL ARIMA MODEL USING PREDICTION ERROR METHOD ALGORITHM

Davor RADENOVIĆ

Knowledge Engineering Laboratory, Mihajlo Pupin Institute
P.O. Box 15, 11000 Belgrade, Yugoslavia

Abstract. Estimating the parameters of a model presents only one stage of the time series modeling procedure. This paper describes an effective way of estimating the parameters of the multiplicative seasonal autoregressive integrated moving average (ARIMA) model in a recursive fashion. This "on line" algorithm is, in contrast to the known "off line" algorithms, noniterative and model independent. The method is based on the Gauss–Newton parameter estimator, updating its gradient and Hessian every time instant some new data becomes available.

Key words: recursive prediction error algorithm, parameter estimation, multiplicative seasonal ARIMA models

1. INTRODUCTION

Estimating the parameters of a model presents only one stage of the time series modeling procedure [2]. Algorithms solving this problem can be generally divided into two groups, nonrecursive (or "off line") algorithms and recursive (or "on line") algorithms. Nonrecursive algorithms collect a batch of data from the system, and subsequently, as a separate procedure, this batch of data is used to estimate the parameters of the model. Recursive algorithms need to have the parameters of the model at the same time as the data is collected. The parameters are then "updated" at each time instant some new data becomes available. The updating is performed by a recursive procedure. The most common "off line" algorithms are Marquardt algorithm [2] and algorithms based on exact likelihood functions [1]. The former belongs to the class of model iterative algorithms and the later belong to the class of model dependent algorithms. In order to avoid these disadvantages, a

recursive algorithm is introduced. The algorithm (approximately) minimizes prediction error and is based on the Gauss–Newton parameter estimator, updating its gradient and Hessian every time instant some new data becomes available.

In the following, we give a time series model description, a recursive estimation algorithm term, problem definition, prediction error algorithm and its derivation, stability test, numerical examples and conclusion.

2. A MULTIPLICATIVE SEASONAL ARIMA MODEL

Let $z(1), z(2), \ldots, z(N)$ represent the observed discrete time series measures at equal time intervals, and let high correlations among observations from the same season (week, month, quarter, year,...) be present. Multiplicative seasonal ARIMA family of models constitutes a flexible and comprehensive set of linear stochastic processes for modeling such series [2]. The multiplicative seasonal ARIMA model can be written in either of the following forms:

$$\begin{aligned}\left(1+a_1q^{-1}+\ldots+a_{n_a}q^{-n_a}\right)\left(1+b_1q^{-s}+\ldots+b_{n_b}q^{-sn_b}\right)(\nabla^d\nabla_s^D z(t)-\mu) = \\ =\left(1+c_1q^{-1}+\ldots+c_{n_c}q^{-n_c}\right)\left(1+d_1q^{-s}+\ldots+d_{n_d}q^{-sn_d}\right)e(t)\end{aligned} \quad (1a)$$

or

$$A(q^{-1})B(q^{-s})(\nabla^d\nabla_s^D z(t)-\mu)=C(q^{-1})D(q^{-s})e(t) \quad (1b)$$

or

$$A(q^{-1})B(q^{-s})(w(t)-\mu)=C(q^{-1})D(q^{-s})e(t) \quad (1c)$$

or

$$A(q^{-1})B(q^{-s})y(t)=C(q^{-1})D(q^{-s})e(t). \quad (1d)$$

3. A RECURSIVE ESTIMATION ALGORITHM

The material for information processing is of course the recorded data. Let y(t) denote data received at time t. Assuming that the data acquisition takes place in discrete time, as is normally the case, we have at time t received a sequence of measurements y(1),y(2), ... ,y(t). Let us use {y(t)} to denote the whole data record. The objective of estimation is to infer a model of the system from the record {y(t)}. Normally, the model is

parametrized in terms of a parameter vector θ, so the objective really is to determine this vector. The estimation problem can thus be phrased as the determination of a mapping from the data {y(t)} to the model parameters θ:

$$\{y(t)\} \rightarrow \hat{\theta}(t;\{y(t)\}) \tag{2}$$

Here the value $\hat{\theta}(t;\{y(t)\})$ is the estimate of θ, based on the information contained in $\{y(t)\}$. In on line or recursive estimation, the estimate $\hat{\theta}(t)$ is required for each t. However, in practice it is important that memory space and computation time do not increase with t. This introduces restrictions upon how $\hat{\theta}(t)$ may be formed. Basically, we will need to condense the observed data into an auxiliary "memory" quantity $S(t)$ of given and fixed dimensions. This auxiliary vector (or matrix) will then be updated according to an algorithm of the structure

$$\begin{aligned} \hat{\theta}(t) &= F(\hat{\theta}(t-1), S(t), y(t)) \\ S(t) &= H(S(t-1), \hat{\theta}(t-1), y(t)) \end{aligned} \tag{3}$$

where the $F(.,.,.)$ and $H(.,.,.)$ are given functions. We see that the estimate $\theta(t)$ is formed from current data, the previous estimate, and the auxiliary variable $S(t)$. The only thing that needs to be stored at time t, consequently, is the information $\left\{\hat{\theta}(t), S(t)\right\}$. This quantity is updated with a fixed algorithm, with a number of operations that does not depend on time t.

4. PROBLEM DEFINITION

Consider now the model (1d). The problem we are solving includes the definitions of the one–ahead–step prediction and prediction error criterion of the underlying model. It is well–known [2], [3] that the *one–ahead–step prediction* for this model can be defined as

$$\hat{y}(t|\theta) = \left\{1 - \frac{A(q^{-1})B(q^{-s})}{C(q^{-1})D(q^{-s})}\right\} y(t) \tag{4}$$

where we have used the parameter vector θ to denote the unknown parameters:

$$\theta^T = (a_1 \cdots a_{n_a} b_1 \cdots b_{n_b} c_1 \cdots c_{n_c} d_1 \cdots d_{n_d}). \tag{5}$$

It is also well–known [3] that we can define the prediction error as

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta) = \frac{A(q^{-1})B(q^{-s})}{C(q^{-1})D(q^{-s})} y(t) \tag{6}$$

It is convenient to introduce the auxiliary variables

$$\bar{y}(t - is, \theta) = A(q^{-1}) y(t - is) \tag{7a}$$

$$\bar{\varepsilon}(t - is, \theta) = C(q^{-1}) \varepsilon(t - is, \theta) \quad . \tag{7b}$$

If we denote

$$\begin{aligned} \varphi^T(t,\theta) = (&-y(t-1) \cdots -y(t-n_a) - \bar{y}(t-s,\theta) \cdots -\bar{y}(t-n_b s,\theta) \\ &\varepsilon(t-1,\theta) \cdots \varepsilon(t-n_c,\theta)\ \ \bar{\varepsilon}(t-s,\theta) \cdots \bar{\varepsilon}(t-n_d s,\theta)) \end{aligned} \tag{8}$$

the linear filter (6) can be written as

$$\hat{y}(t|\theta) = \varphi^T(t,\theta)\theta. \tag{9}$$

If we denote

$$V_t(\theta) = \frac{1}{2} \sum_{k=1}^{t} \varepsilon^2(k,\theta)$$

the *prediction error criterion* can be defined [3] as

$$\min_{\theta} V_t(\theta). \tag{10}$$

Our main task will be to solve the criterion (10) in a recursive fashion (3).

5. RECURSIVE PREDICTION ERROR METHOD (RPEM) ALGORITHM

Consider the prediction error criterion (10). We can follow a similar manner as in [3] and construct the RPEM algorithm. Thus we get well–known [3] formulas denoting RPEM algorithm

$$R(t) = R(t-1) + \psi(t,\hat{\theta}(t-1))\psi^T(t,\hat{\theta}(t-1)) \tag{11}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) - R^{-1}(t)\psi(t,\hat{\theta}(t-1))\varepsilon(t,\hat{\theta}(t-1)) \tag{12}$$

where $\hat{\theta}(t-1)$ denotes the parameter vector estimates at time t–1, $\psi(t,\hat{\theta}(t-1))$ denotes the negative innovation gradient vector at time t–1, and R(t) denotes the Hessian matrix at time t. All that is required is to determine $\psi(t,\hat{\theta}(t-1))$ and $\varepsilon(t,\hat{\theta}(t-1))$. We find the derivative with respect to a_i by differentiating (4)

$$C(q^{-1})D(q^{-s})\frac{\partial}{\partial a_i}\hat{y}(t|\theta) = -B(q^{-s})y(t-i)$$

or

$$\frac{\partial}{\partial a_i}\hat{y}(t|\theta) = -\frac{B(q^{-s})}{C(q^{-1})D(q^{-s})}y(t-i). \tag{13a}$$

Similarly, we find the other derivatives

$$\frac{\partial}{\partial b_i}\hat{y}(t|\theta) = -\frac{A(q^{-1})}{C(q^{-1})D(q^{-s})}y(t-is) \tag{13b}$$

,

$$\frac{\partial}{\partial c_i}\hat{y}(t|\theta) = \frac{1}{C(q^{-1})}\varepsilon(t-i,\theta) \text{ and} \tag{13c}$$

,

$$\frac{\partial}{\partial d_i}\hat{y}(t|\theta) = \frac{1}{D(q^{-s})}\varepsilon(t-is,\theta) \tag{13d}$$

Thus we can compute $\psi(t,\hat{\theta}(t-1))$ and $\varepsilon(t,\hat{\theta}(t-1))$ by solving (4), (6) and (13), respectively. These equations are recursive filters with y as input and ε and ψ as outputs. The filter coefficients are determined by $\hat{\theta}(t-1)$. Since these variables cannot be

computed "recursively" in the sense of (3) using only $\hat{\theta}(t-1)$ and a fixed–size memory vector, we must look for approximations that can be computed recursively. A natural approximation of (6) and (13) is to make only one time iteration with the current estimates and use previous values of ε and ψ as initial values. This means that $\varepsilon(t,\hat{\theta}(t-1))$ and $\psi(t,\hat{\theta}(t-1))$ are approximated by $\varepsilon(t)$ and $\psi(t)$, respectively, calculated according to $\hat{\theta}(t-1)$, equations (4), (6), (13) and their previous values. If we introduce

$$P(t)=[R(t)]^{-1}, \tag{14}$$

and update $P(t)$, using the matrix inversion lemma [3], we obtain

$$P(t)=P(t-1)-\frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1+\psi^T(t)P(t-1)\psi(t)} \tag{15}$$

We summarize the RPEM algorithm in Table 1.

6. STABILITY TEST AND PROJECTION ALGORITHM

The stability set for the predictor (4) is given by

$$D_s=\{\theta|C^*(z)\cdot D^*(z^s) \text{ has all zeros inside the unit circle}\}. \tag{16}$$

where

$$C^*(z)=z^{n_c}+c_1z^{n_c-1}+\ldots+c_{n_c} \tag{17a}$$

$$D^*(z^s)=z^{sn_d}+d_1z^{s(n_d-1)}+\ldots+d_{n_d} \tag{17b}$$

For the RPEM algorithm (Table 1) experience shows that the parameter estimates step out of the stability region and make the algorithm explode. We noticed that it happens within the first two or three seasons and it happens very rarely. The reason probably lies in the initial values of the parameters. Therefore the RPEM algorithm contains stability projection into the stability region. The projection algorithm is defined in a similar manner as in [3] and is therefore omitted.

Table 1. RPEM algorithm

$$\varepsilon(t) = y(t) - \hat{y}(t), \qquad \text{(Prediction error)}$$

$$P(t) = P(t-1) - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1+\psi^T(t)P(t-1)\psi(t)}, \qquad \text{(Hessian update)}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) - P(t)\psi(t)\varepsilon(t), \qquad \text{(Parameter update)}$$

$$\bar{y}(t-is) = A(q^{-1})y(t-is),$$

$$\bar{\varepsilon}(t-is) = C(q^{-1})\varepsilon(t-is),$$

$$\varphi^T(t+1) = (-y(t)\cdots -y(t+1-n_a) - \bar{y}(t+1-s)\cdots -\bar{y}(t+1-n_b s)$$
$$\varepsilon(t)\cdots\varepsilon(t+1-n_c)\ \ \bar{\varepsilon}(t+1-s)\cdots\bar{\varepsilon}(t+1-n_d s)), \qquad \text{(Phi–update)}$$

$$\hat{y}(t+1) = \varphi(t+1)\hat{\theta}^T(t), \qquad \text{(Prediction)}$$

$$\tilde{y}(t-i) = \frac{B(q^{-s})}{C(q^{-1})D(q^{-s})}y(t-i),$$

$$\tilde{\bar{y}}(t-is) = \frac{A(q^{-1})}{C(q^{-1})D(q^{-s})}y(t-is),$$

$$\tilde{\varepsilon}(t-i) = \frac{1}{C(q^{-1})}\varepsilon(t-i),$$

$$\tilde{\bar{\varepsilon}}(t-is) = \frac{1}{D(q^{-s})}\varepsilon(t-is),$$

$$\psi^T(t+1) = (-\tilde{y}(t)\cdots -\tilde{y}(t+1-n_a) - \tilde{\bar{y}}(t+1-s)\cdots -\tilde{\bar{y}}(t+1-n_b s)$$
$$\tilde{\varepsilon}(t)\cdots\tilde{\varepsilon}(t+1-n_c)\ \ \tilde{\bar{\varepsilon}}(t+1-s)\cdots\tilde{\bar{\varepsilon}}(t+1-n_d s)) \qquad \text{(Gradient update)}$$

7. NUMERICAL EXAMPLES

The method described above has been applied successfully to both real and simulated data. We shall describe two examples with both the simulated (see Example 1.) and real data (see Example 2.).

EXAMPLE 1. In order to get the time series $\{y(t)\}$ which we are going to model, we consider the model

$$(1+0.9q^{-1})(1+0.3q^{-12})y(t)=(1-0.3q^{-1})(1-0.5q^{-12})e(t)$$

where $\{e(t), t=1, 1200\}$ is the normally independently distributed white noise with mean 0 and variance 1. After the RPEM algorithm (Table 1.) has been applied to the model

$$(1+a_1q^{-1})(1+b_1q^{-12})y(t)=(1+c_1q^{-1})(1+d_1q^{-12})e(t)$$

the parameters are estimated forming the following model

$$(1+0.91q^{-1})(1+0.25q^{-12})y(t)=(1-0.27q^{-1})(1-0.58q^{-12})e(t)$$

where the estimated variance of the white noise residual was 0.96. Figure 1 shows how the parameter estimates vary with time together with their true values.

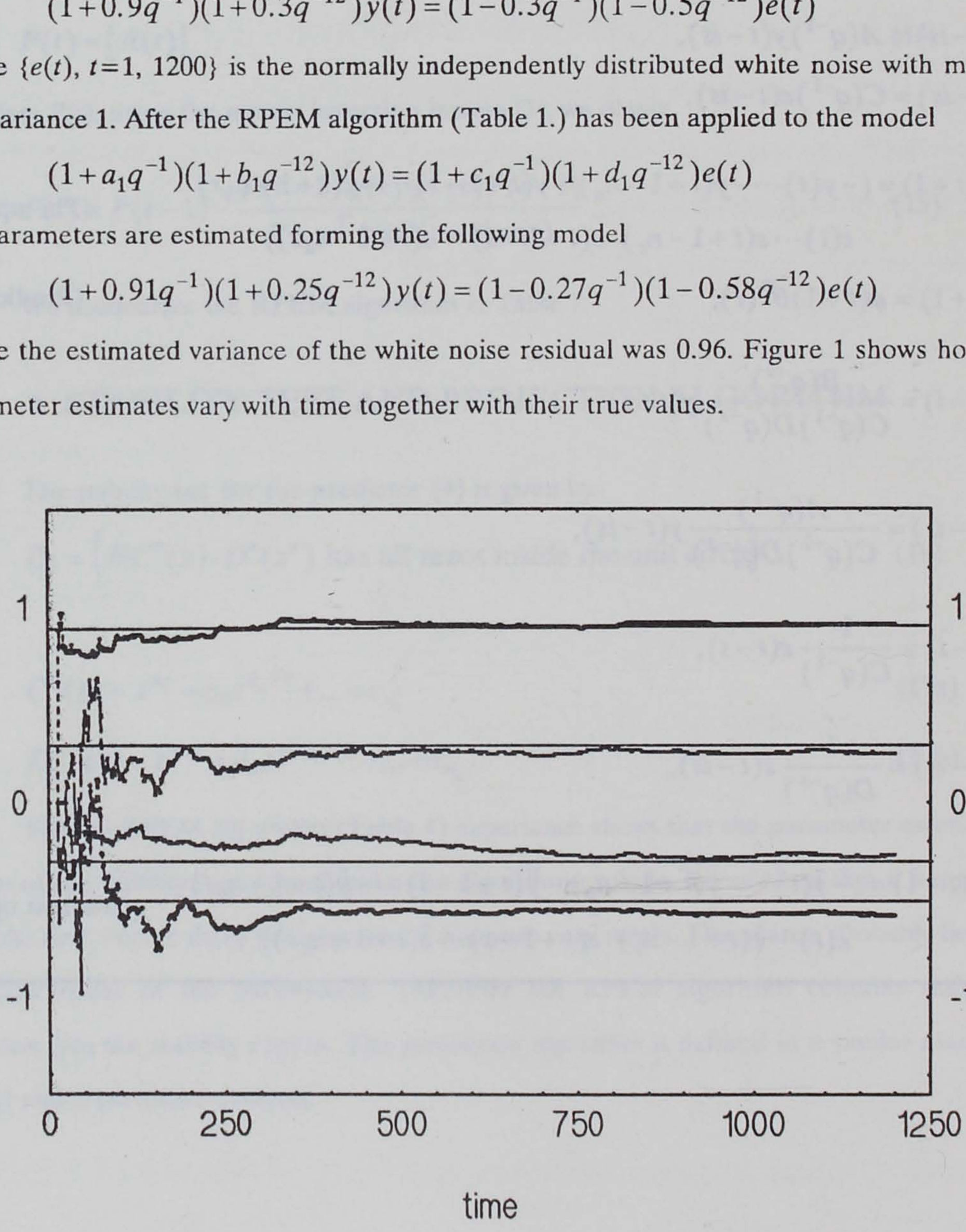

Figure 1. The parameter estimates of the simulated data varying with time

EXAMPLE 2. In order to show how the RPEM was applied to real data we consider data containing the average monthly residential electricity usage in Iowa City from January 1971 to December 1978 (N=96) [1]. After the identification stage of the time series modeling procedure we consider the model

$$\nabla_{12}^{1} z(t) = \theta_0 + \left(1 + c_1 q^{-1}\right)\left(1 + d_1 q^{-12}\right) e(t)$$

to fit the data. RPEM was applied resulting in parameter estimates that form the following model

$$\nabla_{12}^{1} z(t) = 9.11 + \left(1 + 0.31 q^{-1}\right)\left(1 - 0.47 q^{-12}\right) e(t)$$

where the estimated variance of the white noise residual was 1921. Figure 2 shows how the parameter estimates vary with time

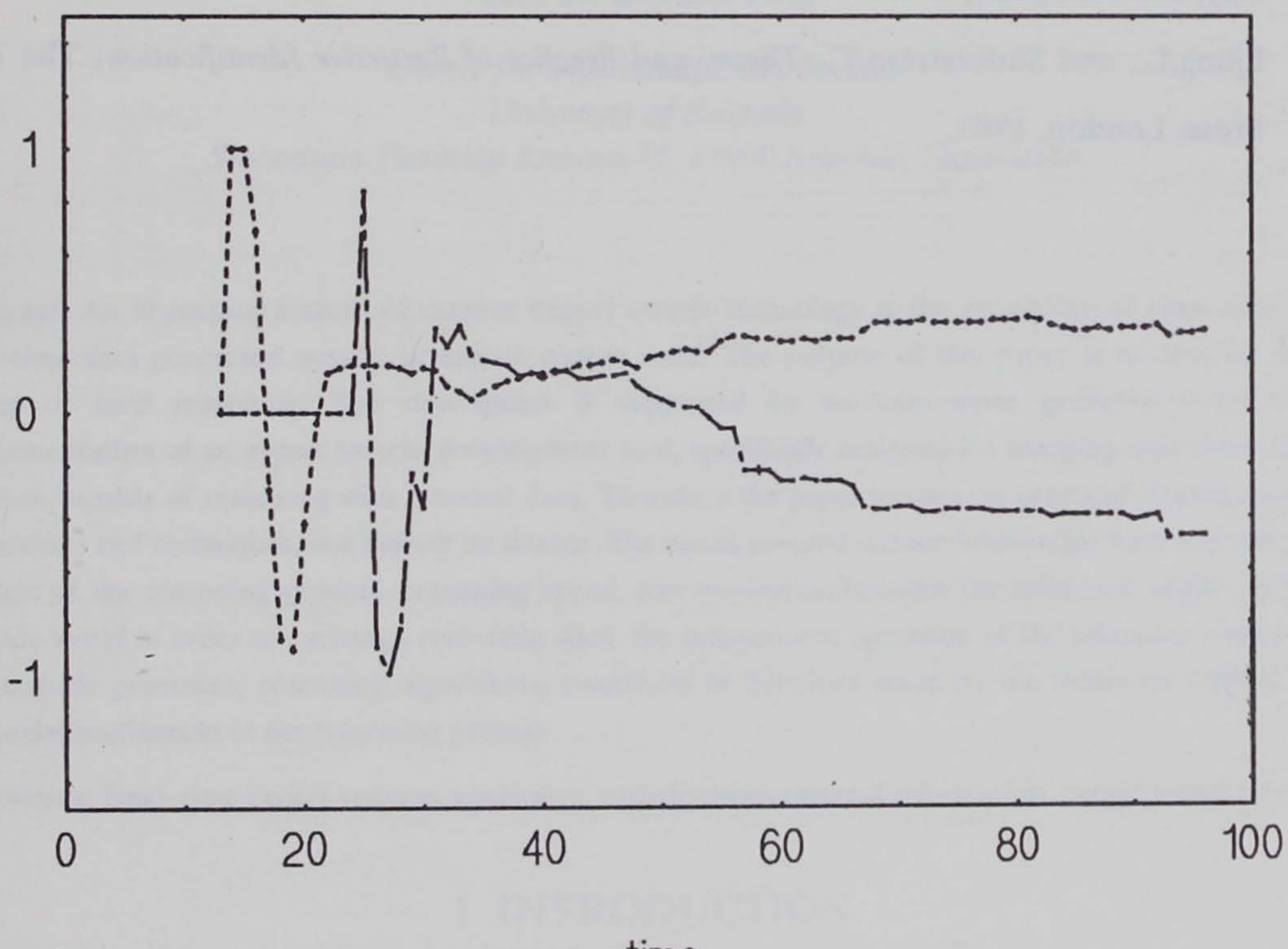

Figure 2. The parameter estimates of the real data varying with time

8. CONCLUSION

In this paper we showed how the recursive prediction error method can be used to define a recursive algorithm in order to estimate the parameters of a multiplicative seasonal

ARIMA model. Applying this algorithm to various numbers of real and simulated time series gave us results that are not worse than results obtained from the nonrecursive algorithms in the forecasting sense. The goodness of the method grows with the new data and without much computation effort. A similar algorithm can be defined also to estimate the parameters of a nonmultiplicative seasonal ARIMA model and further studying of these two phenomena will be the subject of our next investigation.

REFERENCES

[1] Abraham,B., and Ledolter,J., *Statistical Methods for Forecasting,* John Wiley & Sons, New York, 1983.

[2] Box,G.E., and Jenkins,G.M., *Time Series Analysis: Forecasting and Control,* Holden–Day, San Francisco, 1976.

[3] Ljung,L., and Söderström,T., *Theory and Practice of Recursive Identification*, The MIT Press, London, 1983.