Václav Peterka

A square root filter for real time multivariate regression

Kybernetika, Vol. 11 (1975), No. 1, (53)--67

Persistent URL: http://dml.cz/dmlcz/124229

KYBERNETIKA — VOLUME 11 (1975), NUMBER 1

A Square Root Filter for Real Time Multivariate Regression

VÁCLAV PETERKA

An algorithm is developed which makes it possible to perform multivariate regression in real time. The algorithm belongs into the class of square root filters and exhibits outstanding numerical characteristics, particularly in ill-conditioned problems. Because of these properties the algorithm is well suited for implementation in process computers and special digital devices operating with reduced word length.

1. INTRODUCTION

The theory of multivariate regression is well elaborated and may be found in standard textbooks. See e.g. [1]. The purpose of this paper is to follow the algorithmic and numerical point of view. The paper is organized in the following manner.

The multivariate regression model is defined in section 2 where also some known facts are recalled which make the theoretical basis for the rest of the paper. In the theory of regression the parameters of the regression model are assumed to be constants. However, in many practical applications this assumption is fulfilled only approximately. Such situations are often handled by exponential forgetting of old data without increasing the complexity of calculation. This technique is briefly described in section 3 to make the paper self-contained. In section 4 the recursion relations for the estimates of the parameter matrices are derived which are a slight generalization of the recursion formulae commonly used in univariate regression. They are needed in section 5 where the corresponding square root filter is developed and where the main result of this paper may be found. The algorithm derived in section 5 is closely related to square root filters which were developed to improve the numerical characteristics of the Kalman filters. A survey of these techniques is given in [9]. As the problem of regression is somewhat simpler it was possible to bring the final algorithm into a more compact and elegant form compared to filters reported in [9].

The classical problem of multivariate regression which is considered throughout this paper can be stated as follows. It is assumed that the relation between the vector valued variable $y_{(\tau)}$ of dimension v and the vector-valued variable $z_{(\tau)}$ of dimension ϱ can be described by the regression model

$$y_{(\tau)} = P^{\mathrm{T}} z_{(\tau)} + e_{(\tau)} \tag{2.1}$$

where P is $\varrho \times v$ – matrix of regression coefficients and $e_{(\tau)}$ is the stochastic term. The index τ may be interpreted as discrete time.

It is assumed that the data $y_{(\tau)}$, $z_{(\tau)}$ are given for $\tau = 1, 2, \ldots, t$ and the problem is to find some estimate $\hat{P}_{(t)}$ of the unknown matrix P. If it can be assumed that $\{e_{(\tau)};$ $\tau = 1, 2, \ldots, t\}$ is a sequence of mutually uncorrelated normal random vectors with zero mean and covariance matrix R

$$\mathscr{E} e_{(\tau)} = 0\,, \tag{2.2}$$

$$\mathscr{E} e_{(\tau)} e_{(\tau)}^{\mathrm{T}} = R\,, \tag{2.3}$$

$$\mathscr{E} e_{(\tau)} e_{(\tau-i)}^{\mathrm{T}} = 0\,, \quad i \neq 0\,, \tag{2.4}$$

the maximum likelihood estimation method can be applied and the result is [1]

$$\hat{P}_{(t)} = [Z_{(t)}^{\mathrm{T}} Z_{(t)}]^{-1} Z_{(t)}^{\mathrm{T}} Y_{(t)} \tag{2.5}$$

where

$$Z_{(t)} = \begin{bmatrix} z_{(1)}^{\mathrm{T}} \\ z_{(2)}^{\mathrm{T}} \\ \vdots \\ z_{(t)}^{\mathrm{T}} \end{bmatrix}, \quad Y_{(t)} = \begin{bmatrix} y_{(1)}^{\mathrm{T}} \\ y_{(2)}^{\mathrm{T}} \\ \vdots \\ y_{(t)}^{\mathrm{T}} \end{bmatrix}. \tag{2.6}$$

If also the estimate of the covariance matrix R is required the maximum likelihood method gives

$$\hat{R}_{(t)} = \frac{1}{t} [Y_{(t)}^{\mathrm{T}} Y_{(t)} - \hat{P}_{(t)}^{\mathrm{T}} Z_{(t)}^{\mathrm{T}} Z_{(t)} \hat{P}_{(t)}]\,. \tag{2.7}$$

If the random vectors $e_{(\tau)}$, $\tau = 1, 2, \ldots, t$ are independent of the vectors $z_{(\tau)}$ for all τ the estimate $\hat{P}_{(t)}$ is unbiased, i.e.

$$\mathscr{E}[\hat{P}_{(t)} - P] = 0 \tag{2.8}$$

and the four-dimensional covariance tensor of the estimate is [1]

$$\mathcal{E}[\hat{P}_{(t)} - P]_{ir}[\hat{P}_{(t)} - P]_{js} = R_{rs}C_{(t)ij} \tag{2.9}$$

where

$$C_{(t)} = [Z_{(t)}^{\mathrm{T}} Z_{(t)}]^{-1} . \tag{2.10}$$

Under rather general assumptions the formulae (8) and (9) hold asymptotically also in the case of autoregression when the vector $z_{(\tau)}$ contains delayed vectors $y_{(\tau-i)}$, $i = 1, 2, \ldots, n$. This was shown by Durbin [2] who considered the univariate case ($v = 1$) but his results can be well generalized also for multivariate case.

The regression model (1) may have various special forms according to the structure of the vector $z_{(\tau)}$. Of particular interest for control purposes is the linear regression model of a multivariate stochastic dynamic system

$$y_{(\tau)} + \sum_{i=1}^{n} A_i y_{(\tau-i)} = \sum_{i=0}^{n} B_i u_{(\tau-i)} + e_{(\tau)} \tag{2.11}$$

where $y_{(\tau)}$ is the v-vector of outputs and $u_{(\tau)}$ is the μ-vector of inputs. The corresponding vector $z_{(\tau)}$ in (2.1) depends on the way the matrix coefficients A_i and B_i are arranged in P. One possible arrangement is

$$P^{\mathrm{T}} = [B_0, -A_1, B_1, -A_2, \ldots, -A_n, B_n] , \tag{2.12}$$

$$z_{(\tau)}^{\mathrm{T}} = [u_{(\tau)}^{\mathrm{T}}, y_{(\tau-1)}^{\mathrm{T}}, u_{(\tau-1)}^{\mathrm{T}}, y_{(\tau-2)}^{\mathrm{T}}, \ldots, y_{(\tau-n)}^{\mathrm{T}}, u_{(\tau-n)}^{\mathrm{T}}] . \tag{2.13}$$

However, in general the vector $z_{(\tau)}$ may be any – possibly nonlinear but known – vector valued function of $u_{(\tau)}, y_{(\tau-1)}, \ldots, u_{(\tau-n)}$.

The classical formula (2.5) may be interpreted by different ways. For instance, it is an easy exercise to prove that (2.5) is an estimate minimising the criterion

$$J = \sum_{\tau=1}^{t} e_{(\tau)}^{\mathrm{T}} W e_{(\tau)}$$

for any positive definite matrix W and without any assumptions concerning the statistical properties of random vectors $e_{(\tau)}$.

On the other hand also the bayessian probabilistic interpretation may be given to the formulae listed above. In the theory of control of systems with unknown parameters the evolution of the conditional probability density function

$$p(P \mid z_{(t)}, z_{(t-1)}, \ldots, z_{(1)}) \tag{2.14}$$

plays a fundamental role. The right hand sides of the formulae (2.5) and (2.10) determine the mean value and the covariances of the distribution (2.14). This was

shown for univariate case by Åström and Wittenmark [3]. Under mild assumptions concerning the a priori distribution $p(P)$ this holds also in the multivariate case. The details will be reported elsewhere. This facts are briefly mentioned here only to outline the scope of possible applications of the algorithms developed in next sections.

3. EXPONENTIAL FORGETTING

To make the final algorithm more flexible without increasing its complexity the exponential forgetting will be introduced in this section.

The theory of linear regression briefly mentioned in the previous section assumes that the parameter matrix P and the covariance matrix R are unknown constants. The model (2.1) is well suited for many practical situations, however, only seldom it can be guaranteed that the parameters are precisely time-invariant even for a very long time period. If the statistical law for the possible variations of the parameter matrix P were a priori known the Kalman filtering could be applied. Unfortunately this a priori information is often not available and usually all what can be a priori assumed is that the parameters vary relatively slowly. Such a situation can be heuristically handled by the technique which is sometimes referred to as "exponential forgetting" or "overweighting of last data" [4]. The principle idea is very simple and can be explained as follows.

Let t be the current time while τ denotes the past time instant in which the data $y_{(\tau)}$ and $z_{(\tau)}$ were obtained. The estimates are calculated from a set of equations (2.1) for $\tau = t, t-1, t-2, \ldots$. If the parameters may vary in time a very old equation in this set is less reliable than the most recent one. This fact can be respected by the assumption that the variances of the stochastic term $e_{(\tau)}$ in the old equations (2.1) are larger then those in the last one. Specifically

$$\mathscr{E} e_{(\tau)} e_{(\tau)}^{\mathrm{T}} = \varphi^{-2(t-\tau)} R \tag{3.1}$$

where φ is a scalar factor, $\varphi \leqq 1$. By this way the stochastic equation (2.1) is modified into the form

$$y_{(\tau)} = P^{\mathrm{T}} z_{(\tau)} + \varphi^{-(t-\tau)} v_{(\tau)} \tag{3.2}$$

or equivalently

$$\varphi^{(t-\tau)} y_{(\tau)} = P^{\mathrm{T}} [\varphi^{(t-\tau)} z_{(\tau)}] + v_{(\tau)} \tag{3.3}$$

where the random vector $v_{(\tau)}$ has a constant covariance matrix

$$\mathscr{E} v_{(\tau)} v_{(\tau)}^{\mathrm{T}} = R \,. \tag{3.4}$$

Notice that the only diference between the original regression model (2.1) and the stochastic equation (3.3) is in the factor $\varphi^{(t-\tau)} \leqq 1$ by which the old data are weighted.

The scalar $\varphi \leqq 1$ is sometimes called the factor of exponential forgetting and its value has to be chosen intuitively. A reasonable choice usually lies in the interval $0{\cdot}9 < \varphi \leqq 1$.

The introduction of exponential forgetting requires only a slight modification of the formulae listed in section 2. The matrices $Z_{(t)}$ and $Y_{(t)}$ have to be redefined in the following way

(3.5)
$$Z_{(t)} = \begin{bmatrix} \varphi^{(t-1)} & z_{(1)}^{\mathrm{T}} \\ \varphi^{(t-2)} & z_{(2)}^{\mathrm{T}} \\ \vdots & \\ \varphi & z_{(t-1)}^{\mathrm{T}} \\ & z_{(t)}^{\mathrm{T}} \end{bmatrix}, \quad Y_{(t)} = \begin{bmatrix} \varphi^{(t-1)} & y_{(1)}^{\mathrm{T}} \\ \varphi^{(t-2)} & y_{(2)}^{\mathrm{T}} \\ \vdots & \\ \varphi & y_{(t-1)}^{\mathrm{T}} \\ & y_{(t)}^{\mathrm{T}} \end{bmatrix}$$

and the formula for the estimate $\hat{R}_{(t)}$ gets the form

(3.6)
$$\hat{R}_{(t)} = \frac{1}{\varkappa_{(t)}} \left[Y_{(t)}^{\mathrm{T}} Y_{(t)} - \hat{P}_{(t)}^{\mathrm{T}} Z_{(t)}^{\mathrm{T}} Z_{(t)} \hat{P}_{(t)} \right]$$

where

(3.7)
$$\varkappa_{(t)} = \sum_{i=0}^{t-1} \varphi^{2i} = 1 + \varphi^2 \varkappa_{(t-1)} .$$

Notice that the choice $\varphi = 1$ corresponds to the regular regression (no forgetting).

4. RECURSIVE RELATIONS FOR THE ESTIMATES

In this section the recursive relations for the real-time updating of the estimates $\hat{P}_{(t)}$, $\hat{R}_{(t)}$ and the matrix $C_{(t)}$ will be derived. This recursive relations are a generalization of similar relations which are well known for the univariate case. See e.g. [5; 6; 7]. They will serve us as the basis for the development of the corresponding square root filter in section 5.

We will now consider the situation when the estimates $\hat{P}_{(t)}$, $\hat{R}_{(t)}$ and the matrix $C_{(t)}$ are known and new data $y_{(t+1)}$ and $z_{(t+1)}$ are obtained. The problem is how to calculate most effectively the new estimates $\hat{P}_{(t+1)}$, $\hat{R}_{(t+1)}$ and $C_{(t+1)}$. In other words, the last estimates have to be updated to incorporate the new data.

From the definition of the matrices Z and Y (3.5) it follows

(4.1)
$$Z_{(t+1)}^{\mathrm{T}} Z_{(t+1)} = \varphi^2 Z_{(t)}^{\mathrm{T}} Z_{(t)} + z_{(t+1)} z_{(t+1)}^{\mathrm{T}} ,$$

(4.2)
$$Z_{(t+1)}^{\mathrm{T}} Y_{(t+1)} = \varphi^2 Z_{(t)}^{\mathrm{T}} Y_{(t)} + z_{(t+1)} y_{(t+1)}^{\mathrm{T}} ,$$

(4.3)
$$Y_{(t+1)}^{\mathrm{T}} Y_{(t+1)} = \varphi^2 Y_{(t)}^{\mathrm{T}} Y_{(t)} + y_{(t+1)} y_{(t+1)}^{\mathrm{T}} .$$

Using the definition of the matrix C (2.10) the following relation is obtained from (4.1)

$$C_{(t+1)}^{-1} = \varphi^2 C_{(t)}^{-1} + z_{(t+1)} z_{(t+1)}^{\mathrm{T}} . \tag{4.4}$$

Assuming that the inverse of the matrix $C_{(t)}^{-1} = Z_{(t)}^{\mathrm{T}} Z_{(t)}$ exists the well known matrix identity [8]

$$[A + BDB^{\mathrm{T}}]^{-1} = A^{-1} - A^{-1}B[D^{-1} + B^{\mathrm{T}}A^{-1}B]^{-1} B^{\mathrm{T}}A^{-1} \tag{4.5}$$

may be applied to (4.4)

$$C_{(t+1)} = \frac{1}{\varphi^2}[C_{(t)} - C_{(t)} z_{(t+1)} (\varphi^2 + z_{(t+1)}^{\mathrm{T}} C_{(t+1)} z_{(t+1)})^{-1} z_{(t+1)}^{\mathrm{T}} C_{(t)}] . \tag{4.6}$$

Notice that the expression which is to be inverted in (4.5) is a single positive number. Thus we have the following recursion formula for the updating of the matrix $C_{(t)}$

$$C_{(t+1)} = \frac{1}{\varphi^2}\left[C_{(t)} - \frac{1}{\sigma_{(t+1)}^2} C_{(t)} z_{(t+1)} z_{(t+1)}^{\mathrm{T}} C_{(t)}\right] \tag{4.7}$$

where

$$\sigma_{(t+1)}^2 = \varphi^2 + z_{(t+1)}^{\mathrm{T}} C_{(t)} z_{(t+1)} . \tag{4.8}$$

From the recursive equation (4.7) the following relation can be derived

$$C_{(t+1)} z_{(t+1)} = \frac{1}{\varphi^2}\left[C_{(t)} z_{(t+1)} - \frac{1}{\sigma_{(t+1)}^2} C_{(t)} z_{(t+1)} (\sigma_{(t+1)}^2 - \varphi^2)\right] =$$

$$= \frac{1}{\varphi^2}\left(1 - \frac{\sigma_{(t+1)}^2 - \varphi^2}{\sigma_{(t+1)}^2}\right) C_{(t)} z_{(t+1)} ,$$

$$C_{(t+1)} z_{(t+1)} = \frac{1}{\sigma_{(t+1)}^2} C_{(t)} z_{(t+1)} . \tag{4.9}$$

This useful relation will be exploited for the derivation of the updating equations for $\hat{P}_{(t)}$ and $\hat{R}_{(t)}$. Consider $\hat{P}_{(t)}$ first. From (2.5) and (2.10) follows

$$C_{(t)}^{-1} P_{(t)} = Z_{(t)}^{\mathrm{T}} Y_{(t)} \tag{4.10}$$

and similarly

$$C_{(t+1)}^{-1} P_{(t+1)} = Z_{(t+1)}^{\mathrm{T}} Y_{(t+1)} . \tag{4.11}$$

Applying the relation (4.2) we have

$$C_{(t+1)}^{-1}\hat{P}_{(t+1)} = \varphi^2 Z_{(t)}^{\mathrm{T}} Y_{(t)} + z_{(t+1)} y_{(t+1)}^{\mathrm{T}} = \tag{4.12}$$
$$= \varphi^2 C_{(t)}^{-1}\hat{P}_{(t)} + z_{(t+1)} y_{(t+1)}^{\mathrm{T}} .$$

The equation (4.4) gives

$$\varphi^2 C_{(t)}^{-1} = C_{(t+1)}^{-1} - z_{(t+1)} z_{(t+1)}^{\mathrm{T}} . \tag{4.13}$$

The substitution of (4.13) into (4.12) and a simple rearrangement gives one of possible alternatives of the recursion formula

$$\hat{P}_{(t+1)} = \hat{P}_{(t)} + C_{(t+1)} z_{(t+1)}\, \hat{e}_{(t+1)}^{\mathrm{T}} \tag{4.14}$$

where

$$\hat{e}_{(t+1)} = y_{(t+1)} - P_{(t)}^{\mathrm{T}} z_{(t+1)} . \tag{4.15}$$

Using (4.9) the recursion (4.14) can be written also in the form

$$\hat{P}_{(t+1)} = \hat{P}_{(t)} + \frac{1}{\sigma_{(t+1)}^2} C_{(t)} z_{(t+1)} \hat{e}_{(t+1)}^{\mathrm{T}} . \tag{4.16}$$

This form of the formula for updating of $\hat{P}_{(t)}$ is most commonly used in univariate regression [5; 6; 7] and in such a case is closely related to the Kalman filter [6].

Now the updating of the estimate $\hat{R}_{(t)}$ will be considered. Rewrite the formula (3.6) in the following way

$$\varkappa_{(t)}\hat{R}_{(t)} = Y_{(t)}^{\mathrm{T}} Y_{(t)} - \hat{P}_{(t)}^{\mathrm{T}} C_{(t)}^{-1}\hat{P}_{(t)} . \tag{4.17}$$

Similarly we have

$$\varkappa_{(t+1)}\hat{R}_{(t+1)} = Y_{(t+1)}^{\mathrm{T}} Y_{(t+1)} - \hat{P}_{(t+1)}^{\mathrm{T}} C_{(t+1)}^{-1}\hat{P}_{(t+1)} \tag{4.18}$$

and using the equalities (4.3) and (4.17)

$$\varkappa_{(t+1)}\hat{R}_{(t+1)} = \varphi^2 Y_{(t)}^{\mathrm{T}} Y_{(t)} + y_{(t+1)} y_{(t+1)}^{\mathrm{T}} - \hat{P}_{(t+1)} C_{(t+1)}^{-1} P_{(t+1)} = \tag{4.19}$$
$$= \varphi^2 \varkappa_{(t)}\hat{R}_{(t)} + \hat{P}_{(t)}^{\mathrm{T}} C_{(t)}^{-1}\hat{P}_{(t)} + y_{(t+1)} y_{(t+1)}^{\mathrm{T}} - P_{(t+1)}^{\mathrm{T}} C_{(t+1)}^{-1}\hat{P}_{(t+1)} .$$

The substitution of (4.4) and (4.16) into (4.19) and a cumbersome but straightforward rearrangement leads to the following simple result

$$\varkappa_{(t+1)}\hat{R}_{(t+1)} = \varphi^2 \left[\varkappa_{(t)}\hat{R}_{(t)} + \frac{1}{\sigma_{(t+1)}^2} \hat{e}_{(t+1)} \hat{e}_{(t+1)}^{\mathrm{T}} \right] . \tag{4.20}$$

Summarizing we thus find that the estimates of the matrix P and the covariance matrix R can be computed recursively using the following equations

$$g_{(t+1)} = C_{(t)} z_{(t+1)}\,, \tag{4.21}$$

$$\sigma^2_{(t+1)} = \varphi^2 + z^{\mathrm{T}}_{(t+1)} g_{(t+1)}\,, \tag{4.22}$$

$$\hat{e}_{(t+1)} = y_{(t+1)} - \hat{P}^{\mathrm{T}}_{(t)} z_{(t+1)}\,, \tag{4.23}$$

$$\hat{P}_{(t+1)} = \hat{P}_{(t)} + \frac{1}{\sigma^2_{(t+1)}} g_{(t+1)} \hat{e}^{\mathrm{T}}_{(t+1)}\,, \tag{4.24}$$

$$\varkappa_{(t+1)} \hat{R}_{(t+1)} = \varphi^2 \left[\varkappa_{(t)} \hat{R}_{(t)} + \frac{1}{\sigma^2_{(t+1)}} \hat{e}_{(t+1)} \hat{e}^{\mathrm{T}}_{(t+1)} \right], \tag{4.25}$$

$$\varkappa_{(t+1)} = 1 + \varphi^2 \varkappa_{(t)}\,, \tag{4.26}$$

$$C_{(t+1)} = \frac{1}{\varphi^2} \left[C_{(t)} - \frac{1}{\sigma^2_{(t+1)}} g_{(t+1)} g^{\mathrm{T}}_{(t+1)} \right]. \tag{4.27}$$

The four-dimensional covariance tensor of the estimate $\hat{P}_{(t+1)}$ gives the formula

$$\mathscr{E}\left[\hat{P}_{(t+1)} - P\right]_{ir} \left[\hat{P}_{(t+1)} - P\right]_{js} = R_{rs} C_{(t+1)ij}\,. \tag{4.28}$$

As the entries R_{rs} are rarely a priori known they have to be replaced by their estimates obtained from (4.25).

The starting values $P_{(0)}$ and $C_{(0)}$ may be interpreted as an apriori estimate of P the accuracy of which is characterized by $C_{(0)}$ according to (4.28). If no a priori estimate is available $C_{(0)}$ may be a diagonal matrix with sufficiently large numbers on the main diagonal. The error caused by this approximation is usually removed very fast. This is a standard trick often used also in the univariate case [6].

5. SQUARE ROOT FILTER

Numerical difficulties may be encountered when the recursion equations are applied to ill-conditioned estimation problems. Theoretically the matrix $C_{(t)}$ must be allways positive semidefinite. However, due to the rounding errors caused by the finite word length of the computing device the matrix $C_{(t)}$ calculated by means of the recursion equations (4.21), (4.22) and (4.27) may loose this property. This may occur when some components of the vectors $z_{(t)}$ are linearly dependent. As an example may serve the situation when the algorithm is used as on-line identifier in a selftuning digital regulator and the feed-back converges to a constant control law. Similar

problems were encountered in the Kalman filtering and motivated the development of the square root filtering techniques [9]. The regression is somewhat special and simpler in comparison to the Kalman filter which makes it possible to bring the square root filter to a more compact and elegant form than the filters reported in [9]. This will be shown in the sequel.

The positive definite matrix $C_{(t)}$ can be factored into the product

$$C_{(t)} = G_{(t)}G_{(t)}^{\mathrm{T}} \tag{5.1}$$

where $G_{(t)}$ is an upper triangular matrix. The factorization (5.1) is equivalent to the Cholesky decomposition [8] of the inversed matrix

$$C_{(t)}^{-1} = G_{(t)}^{-\mathrm{T}}G_{(t)}^{-1}\,. \tag{5.2}$$

The notation $[\cdot]^{-\mathrm{T}} = [(\cdot)^{-1}]^{\mathrm{T}} = [(\cdot)^{\mathrm{T}}]^{-1}$ is used for convenience.

Notice that the matrix $C_{(t)}$ never can become indefinite if its square root $G_{(t)}$ is propagated instead of $C_{(t)}$ itself. This is the fundamental idea of square root filtering.

With the factorization (5.1) the recursion equation (4.4) can be rewritten in the following way

$$\begin{aligned} G_{(t+1)}G_{(t+1)}^{\mathrm{T}} &= [\varphi^2 G_{(t)}^{-\mathrm{T}}G_{(t)}^{-1} + z_{(t+1)}z_{(t+1)}^{\mathrm{T}}]^{-1} = \\ &= \frac{1}{\varphi^2}\,G_{(t)}\left[I + f_{(t+1)}\frac{1}{\varphi^2}f_{(t+1)}^{\mathrm{T}}\right]^{-1}G_{(t)}^{\mathrm{T}} \end{aligned} \tag{5.3}$$

where

$$f_{(t+1)} = G_{(t)}^{\mathrm{T}}z_{(t+1)}\,. \tag{5.4}$$

The inner matrix in (5.3) may be inverted by means of the matrix identity (4.5)

$$G_{(t+1)}G_{(t+1)}^{\mathrm{T}} = \frac{1}{\varphi^2}\,G_{(t)}\left[I - f_{(t+1)}\frac{1}{\sigma_{(t+1)}^2}f_{(t+1)}^{\mathrm{T}}\right]G_{(t)}^{\mathrm{T}} \tag{5.5}$$

where in agreement with (4.8)

$$\sigma_{(t+1)}^2 = \varphi^2 + f_{(t+1)}^{\mathrm{T}}f_{(t+1)}\,. \tag{5.6}$$

The following rearrangement of the equation (5.5) is the crucial trick in the whole procedure.

$$G_{(t+1)}G_{(t+1)}^{\mathrm{T}} = \frac{1}{\varphi}\,G_{(t)}\left[I, \mathrm{j}\frac{f_{(t+1)}}{\sigma_{(t+1)}}\right]UU^{\mathrm{T}}\begin{bmatrix} I \\ \mathrm{j}\dfrac{f_{(t+1)}^{\mathrm{T}}}{\sigma_{(t+1)}} \end{bmatrix}G_{(t)}^{\mathrm{T}}\frac{1}{\varphi} \tag{5.7}$$

where $\mathrm{j} = \sqrt{-1}$ and U is any orthogonal matrix

$$UU^{\mathrm{T}} = I\,. \tag{5.8}$$

Notice the symmetry of the equation (5.7). It indicates that the recursion $G_{(t)} \to G_{(t+1)}$ will be found if it is possible to find an orthogonal matrix U such that

$$\left[I, \mathrm{j}\frac{f}{\sigma} \right] U = [H, 0] \tag{5.9}$$

where H is a real upper triangular matrix and 0 in (5.9) denotes a vector the components of which are all equal to zero. In other words, the orthogonal matrix U has to be found such that the imaginary vector in the left-hand side of (5.9) is zeroed. The time index is ommited in (5.9) and in the following text where no confusion may occur.

From the equalities (5.7) and (5.9) it follows that the matrix $G_{(t+1)}$ is given by the formula

$$G_{(t+1)} = \frac{1}{\varphi} G_{(t)} H . \tag{5.10}$$

As both matrices in the right-hand side of (5.10) are upper triangular their product $G_{(t+1)}$ is also upper triangular.

Now it will be shown that the transformation $f/\sigma \to H$ fulfilling the relation (5.9) can be performed very effectively without explicite calculation of the orthogonal matrix U and operating only with real numbers. Also the matrix product (5.10) may be incorporated into the algorithm which removes the necessity of storing the matrix H and significantly reduces the requirements for the memory capacity of the computing device.

From the required equality (5.9) it is evident that the orthogonal matrix U cannot be real. Notice that the matrix U is of dimension $(\varrho + 1) \times (\varrho + 1)$ where ϱ is the dimension of the vector $z_{(t+1)}$ and thus also of the vector f. Express the orthogonal matrix U as a product

$$U = U^{(\varrho)} U^{(\varrho - 1)} \ldots U^{(i)} \ldots U^{(1)}$$

where $U^{(i)}$ is a generalized elementary matrix of rotation

$$U^{(i)} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c_i & \cdots & \cdots & \mathrm{j}s_i \\ & & & \vdots & 1 & & \vdots \\ & & & \vdots & & \ddots & \vdots \\ & & & \vdots & & 1 & \vdots \\ & & & -\mathrm{j}s_i & \cdots & \cdots & c_i \end{bmatrix} \begin{matrix} \\ \\ \\ - - i \\ \\ \\ \\ - - \varrho + 1 \end{matrix} \tag{5.11}$$

$$\qquad\qquad\qquad i \qquad\qquad \varrho + 1$$

where only nonzero entries are indicated. The only difference between the ordinary elementary matrix of rotation [10] and the matrix (5.11) is that the offdiagonal elements of (5.11) are imaginary. The condition of orthogonality (5.8) requires that

$$c_i^2 - s_i^2 = 1 . \tag{5.12}$$

Consider the first product

$$\left[I, \mathrm{j}\frac{f}{\sigma}\right] U^{(\varrho)} \tag{5.13}$$

in the left-hand side of (5.9). Notice that by this matrix multiplication only the last two columns of the left-hand factor in (5.13) are changed. It means that in this first stage it is fully sufficient to consider the transformation

$$\begin{bmatrix} 0 & \\ 0 & \\ \vdots & \mathrm{j}\dfrac{f^{(\varrho)}}{\sigma_\varrho} \\ 0 & \\ 1 & \end{bmatrix} \begin{bmatrix} c_\varrho & \mathrm{j}s_\varrho \\ -\mathrm{j}s_\varrho & c_\varrho \end{bmatrix} \tag{5.14}$$

where the notation

$$f^{(\varrho)} = f = G_{(t)}^{\mathrm{T}} z_{(t+1)} \tag{5.15}$$

$$\sigma_\varrho = \sqrt{\varphi^2 + f^{(\varrho)\mathrm{T}} f^{(\varrho)}} = \sqrt{\varphi^2 + f^{\mathrm{T}}} \tag{5.16}$$

was used to indicate the first stage of the procedure. The two parameters of the transformation (5.14) are bounded by the orthogonality condition (5.12). Our goal is to annul the column $\mathrm{j}\sigma_\varrho^{-1} f^{(\varrho)}$. Therefore the second condition for the determination of c_ϱ and s_ϱ will be chosen so that the last component of this vector, namely $\mathrm{j}\sigma_\varrho^{-1} f_\varrho^{(\varrho)}$, be zeroed. This requirement is met when

$$s_\varrho + c_\varrho \frac{f_\varrho^{(\varrho)}}{\sigma_\varrho} = 0 . \tag{5.17}$$

The equations (5.17) and (5.12) for s_ϱ and c_ϱ give

$$c_\varrho = \frac{\sigma_\varrho}{\sqrt{\sigma_\varrho^2 - (f_\varrho^{(\varrho)})^2}} = \frac{\sigma_\varrho}{\sigma_{\varrho-1}} , \tag{5.18}$$

$$s_\varrho = -\frac{f_\varrho^{(\varrho)}}{\sigma_\varrho} c_\varrho = -\frac{f_\varrho^{(\varrho)}}{\sigma_{\varrho-1}} \tag{5.19}$$

where

$$\sigma_{\varrho-1} = \sqrt{\sigma_\varrho^2 - (f_\varrho^{(\varrho)})^2} = \sqrt{\varphi^2 + \sum_{k=1}^{\varrho-1} f_k^2} . \tag{5.20}$$

It is convenient for further use to generalize this notation

$$\sigma_i = \sqrt{\varphi^2 + \sum_{k=1}^{i} f_k^2} \,. \tag{5.21}$$

Notice that

$$\sigma_i = \sqrt{\sigma_{i-1}^2 + f_i^2} \tag{5.22}$$

and

$$\sigma_0 = \varphi \,. \tag{5.23}$$

The transformation (5.14) with the parameters c_ϱ and s_ϱ determined by (5.18) and (5.19) gives the result

$$\left[\begin{array}{c|c} \begin{matrix} H_{1,\varrho} \\ H_{2,\varrho} \\ \vdots \\ H_{\varrho-1,\varrho} \\ H_{\varrho,\varrho} \end{matrix} & \begin{matrix} \mathrm{j}\,\dfrac{f^{(\varrho-1)}}{\sigma_{\varrho-1}} \\ \\ \\ \\ 0 \end{matrix} \end{array}\right] \tag{5.24}$$

where

$$f_k^{(\varrho-1)} = f_k^{(\varrho)} = f_k \quad \text{for} \quad k < \varrho \,, \tag{5.25}$$

$$H_{k,\varrho} = \frac{f_k^{(\varrho)}}{\sigma_\varrho} s_\varrho = -\frac{f_k f_\varrho}{\sigma_\varrho \sigma_{\varrho-1}} \quad \text{for} \quad k < \varrho \,, \tag{5.26}$$

$$H_{\varrho,\varrho} = c_\varrho + \frac{f_\varrho^{(\varrho)}}{\sigma_\varrho} s_\varrho = \frac{\sigma_{\varrho-1}}{\sigma_\varrho} \,. \tag{5.27}$$

The following elementary matrix $U^{(\varrho-1)}$ is exploited to annul the last component of the vector $\mathrm{j}\,\sigma_{\varrho-1}^{-1} f^{(\varrho-1)}$ (with the index $\varrho - 1$) and simultaneously the next column of the upper triangular matrix H is obtained. By this way the whole matrix can be successively generated operating only with the norms (5.22). In the general i-th step of this procedure we have

$$H_{ki} = -\frac{f_k f_i}{\sigma_i \sigma_{i-1}} \quad \text{for} \quad i < k \,, \tag{5.28}$$

$$H_{ii} = \frac{\sigma_{i-1}}{\sigma_i} \,. \tag{5.29}$$

The general formulae (5.28) and (5.29) make it possible to calculate the matrix product (5.10) directly.

$$G_{(t+1)ij} = \frac{1}{\varphi} \sum_{k=i}^{j} G_{(t)ik} H_{kj} , \tag{5.30}$$

$$G_{(t+1)ij} = \frac{1}{\varphi} \frac{\sigma_{j-1}}{\sigma_j} \left(G_{(t)ij} - \frac{f_j}{\sigma_{j-1}^2} \sum_{k=i}^{j-1} G_{(t)ik} f_k \right) . \tag{5.31}$$

If the vector $g^{(m)}$ is introduced defined by

$$g_i^{(m)} = \sum_{k=i}^{m} G_{(t)ik} f_k \; ; \quad i = 1, 2, \ldots, m , \tag{5.32}$$

$$g_i^{(m)} = 0 \; ; \quad i > m , \tag{5.33}$$

the recursion formula (5.31) gets the form

$$G_{(t+1)ij} = \frac{1}{\varphi} \frac{\sigma_{j-1}}{\sigma_j} \left(G_{(t)ij} - \frac{f_j g_i^{(j-1)}}{\sigma_{j-1}^2} \right) . \tag{5.34}$$

Summing up it is seen that the equations (5.4), (5.22), (5.23), (5.32) and (5.34) describe the whole updating procedure. The sequence of calculation may be chosen so that the updating $G_{(t)} \to G_{(t+1)}$ is performed operating only on one triangular matrix and that only one component of each of the vectors σ_i, σ_i^2 and f_i has to be stored in the memory. This possibilities are respected in the algorithm REFIL.

ALGORITHM REFIL

$\sigma_0 = \varphi$

$\sigma_0^2 = \varphi^2$

$j = 1, 2, \ldots, \varrho$

$f_j = \sum_{i=1}^{j} G_{(t)ij} z_{(t+1)i}$

$a = \sigma_{j-1}/\varphi$

$b = f_j/\sigma_{j-1}^2$

$\sigma_j^2 = \sigma_{j-1}^2 + f_j^2$

$\sigma_j = \sqrt{\sigma_j^2}$

$c = a/\sigma_j$

$g_j = G_{(t)jj} f_j$

$$
\begin{aligned}
& G_{(t+1)jj} = cG_{(t)jj} \\
& i \qquad = 1, 2, \ldots, j-1 \\
& \qquad d \qquad = G_{(t)ij} \\
& \qquad G_{(t+1)ij} = c(d - bg_i) \\
& \qquad g_i \qquad = df_j + g_i
\end{aligned}
$$

Notice that the variables f_j, σ_j^2 and σ_j do not need to be indexed. They may be considered as scalars in the computer program. The variables a and c may have the same identifier and the time indices of the matrix G are superfluous. They are given only for better understanding of the algorithm.

Notice also that the updating algorithm $[G_{(t)}, z_{(t+1)}] \to G_{(t+1)}$ returns the vector g and the scalar σ^2 which are required for the updating of the estimates $\hat{P}_{(t)}$ and $\hat{R}_{(t)}$ according to the formulae (4.23), (4.24) and (4.25)

$$\hat{e}_{(t+1)} = y_{(t+1)} - \hat{P}_{(t)} z_{(t+1)} ,$$

$$\hat{P}_{(t+1)} = \hat{P}_{(t)} + \frac{g}{\sigma^2} \hat{e}_{(t+1)}^{\mathrm{T}} ,$$

$$\varkappa_{(t+1)} \hat{R}_{(t+1)} = \left[\varkappa_{(t)} \hat{R}_{(t)} + \frac{1}{\sigma^2} \hat{e}_{(t+1)} \hat{e}_{(t+1)}^{\mathrm{T}} \right] \varphi^2 .$$

The careful reader found out that the algorithm REFIL was written in such a way that it can be directly transformed into a computer program. The computational requirements (measured by the number of arithmetic operations and storage capacity required) are for both the square root and conventional implementations approximately the same.

As was pointed out in [9] the numerical conditioning of the matrix G is generally much better than that of the matrix C which is reflected in the effective double precision of the square root filter in ill-conditioned estimation problems. This may be appreciated particularly when the process computer or a special digital device (such as adaptive digital controller) operates with a reduced word length.

The necessity of calculating ϱ square roots in each updating period is the price which is paid for the excellent numerical properties of the square root filter. The experience shows that it pays out.

(Received May 20, 1974.)

[1] Anderson T. W.: An Introduction to Multivariate Statistical Analysis. John Wiley, N. Y. 1958.

[2] Durbin J.: Estimation of Parameters in Time-series Regression Models. J. Royal Stat. Soc. Series B *22* (1960), 1, 134—153.

[3] Åström K. J., Wittenmark B.: Problems of Identification and Control. Journal of Mathematical Analysis and Applications, *34* (1971), 90—113.

[4] Jazwinski A. H.: Stochastic Processes and Filtering Theory, Academic Press, New York—London 1970

[5] Lee R. C. K.: Optimal Estimation, Identification, and Control. MIT Press, 1964

[6] Åström K. J.: Lectures on the Identification Problem — The Least Square Method, Report 6806, Lund Institute of Technology, Sweden, 1968.

[7] Sage A. P., Melsa J. L.: Estimation Theory with Applications to Communications and Control. McGraw—Hill, N. Y. 1971.

[8] Householder A. S.: The Theory of Matrices in Numerical Analysis, Blaisdell Publ. Comp., 1964.

[9] Kaminski P. G., Bryson A. E., Schmidt S. F.: Discrete Square Root Filtering: A Survey of Current Techniques, IEEE Trans. *AC-16* (December 1971), 6, 727—735.

[10] Faddeev D. K., Faddeeva V. N.: Numerical Methods of Linear Algebra (in Russian). Fizmatgiz, Moscow 1963.

Ing. Václav Peterka, CSc.; Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 180 76 Praha 8 - Libeň. Czechoslovakia.