Akademie věd české republiky
ÚSTAV TEORIE INFORMACE A AUTOMATIZACE

Academy of Sciences of the Czech Republic
INSTITUTE OF INFORMATION THEORY AND AUTOMATION

RESEARCH REPORT

J.Kadlec

Transputer Implementation of Block Regularised Filtering

No. 1791 January 1994

ÚTIA AV ČR P. O. Box 18, 182 08 Prague, Czech Republic
Telex: 122018 atom c, Fax: (+42)(2)6641 4903
e-mail: utia@utia.cas.cz

This report constitutes an unrefereed manuscript which is intended to be submitted for publication. Any opinions and conclusions expressed in this report are those of the author(s) and do not necessarily represent the views of the Institute.

Transputer Implementation of Block Regularised Filtering

J. Kadlec

Inst. of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08, Prague 8
Czech Republic, Phone: (422) 8152274
e-mail: ADAPT@CSPGAS11.BITNET

Abstract. A novel approach to parallel transputer implementation of a regularised exponential parameter tracking is described.

The proposed block regularised exponential algorithm allows to define and automatically adjust mean value of an alternative covariance of the estimated parameters in the boundaries of blocks of processed data. The alternative mean tracks in blocks the current parameter estimates. That is why the influence of the regularisation on the parameter estimates is reduced, and the algorithm remains compatible with the completely pipelined parallel transputer implementation.

1 Introduction

Recursive parameter estimation and filtering is a key tool in modern application areas such as the adaptive signal processing or control.

When there is little information available about the detailed structure of the system, the standard Kalman filter model [10, 11] is reduced to a "random-walk of parameters" regression type model

$$\begin{aligned} \theta_{k+1} &= \theta_k + w_k \\ y_k &= \varphi_k' \theta_k + v_k \,. \end{aligned} \qquad (1)$$

Here θ_k is the vector of n unknown system parameters, y_k is the sampled system output, φ_k is the vector of n past input and output data, v_k and w_k are assumed to be the zero mean, Gaussian, white noise sequences characterised by the covariance matrix W_k and V_k respectively.

The related estimator takes the form of the standard Kalman-filter data update step (2) combined with different specifications of the time update part of the recursive parameter estimation.

Data Update.

$$
\begin{aligned}
\hat{e}_{k|k-1} &= y_k - \varphi_k' \hat{\theta}_{k|k-1} \\
P_{k|k} &= P_{k|k-1} - \frac{P_{k|k-1}\varphi_k\varphi_k' P_{k|k-1}}{1 + \varphi_k' P_{k|k-1}\varphi_k} \\
\hat{\theta}_{k|k} &= \hat{\theta}_{k|k-1} + P_{k|k}\,\varphi_k\,\hat{e}_{k|k-1}
\end{aligned}
\tag{2}
$$

Kalman filter-like Time Update. Specified by $0 \leq W_k$.

$$
\begin{aligned}
\hat{\theta}_{k+1|k} &= \hat{\theta}_{k|k} \\
P_{k+1|k} &= P_{k|k} + W_k
\end{aligned}
\tag{3}
$$

The alternative concept of exponential age–weighting [5] by the scalar exponential weighting factor $0 < \lambda_k \leq 1$ is frequently used when an effective implementation is required [17].

Exponential Forgetting Time Update. Specified by $0 < \lambda_k \leq 1$.

$$
\begin{aligned}
\hat{\theta}_{k+1|k} &= \hat{\theta}_{k|k} \\
P_{k+1|k} &= \lambda_k^{-1} P_{k|k}
\end{aligned}
\tag{4}
$$

The adding of W_k to $P_{k|k}$ (3) is avoided in (4) and the algorithm can be based on the efective QR update of the square root factors of the information matrix $P_{k|k}^{-1}$ and implemented on the triangular systolic array of Gentleman-Kung [4], [16], [18].

1.1 Limits of the standard time update formulation

When the measured data contains little information about some of the parameters, the parameter estimates become increasingly dependent on the parameters specifying the time update. Standard time updates (3) or (4) take into account neither the information contained in previous data, nor prior information about the possible values of the estimated parameters. Thus, numerical problems could eventually occur, even in the case of numerically robust factorised implementations.

Several techniques to cope with poor informative data have been proposed in the current literature. Directional forgetting [6, 9], the constant trace approach [7] and the shift invariant directional forgetting are only a few examples. Their original formulation has been designed for efficient sequential implementation. If these methods are directly mapped onto a systolic implementation, the global feedback loops in the signal flow degrade considerably the data throughput of the array. Their formulation must therefore be revised in the pipelined transputer implementation to avoid low processor utilisation. This paper addresses this issue.

1.2 The Concept of Regularised Estimator

To overcome these limitations, we shall design a systolic regularised estimator using the general concept of parameter tracking introduced in the Bayesian formulation by Kulhavý in [14] and analysed in [15].

Generalised exponential forgetting time update. Specified by θ_k^* ; $(P_k^*)^{-1} \geq 0$; $0 < \lambda_k \leq 1$.

$$\begin{aligned} A_k &\hat{=} \left\{ I + \frac{1-\lambda_k}{\lambda_k} P_{k|k} (P_k^*)^{-1} \right\}^{-1} \\ \hat{\theta}_{k+1|k} &= A_k \hat{\theta}_{k|k} + (I - A_k) \theta_k^* \\ P_{k+1|k} &= \left\{ \lambda_k P_{k|k}^{-1} + (1-\lambda_k)(P_k^*)^{-1} \right\}^{-1} \end{aligned} \tag{5}$$

The time update step (5) could be understood as the introduction of an additional, alternative reference matrix $(P_k^*)^{-1} > 0$ and parameter estimate θ_k^* exhibiting the role of an "alternative value" or "regularising point" in the specification of the time update step.

1.3 Regularised Information Filtering

The vectors $\hat{\theta}_{k|k}$; θ_k^* and (n, n) matrices $P_{k|k}$; P_k^* can be propagated in transformed, inverse or factorised forms. The information filter works with the transformed regression parameter estimates denoted as

$$d_{k|k} \hat{=} P_{k|k}^{-1} \hat{\theta}_{k|k} \ ; \ d_k^* \hat{=} (P_k^*)^{-1} \theta_k^*. \tag{6}$$

Information Data Update.

$$\begin{aligned} P_{k|k}^{-1} &= P_{k|k-1}^{-1} + \varphi_k \varphi_k' \\ d_{k|k} &= d_{k|k-1} + \varphi_k y_k \\ \hat{\theta}_{k|k} &= P_{k|k} d_{k|k} \end{aligned} \tag{7}$$

Regularised Information Time Update. Specified by d_k^* ; $(P_k^*)^{-1} \geq 0$; $0 < \lambda_k \leq 1$.

$$\begin{aligned} d_{k+1|k} &= \lambda_k d_{k|k} + (1-\lambda_k) d_k^* \\ P_{k+1|k}^{-1} &= \lambda_k P_{k|k}^{-1} + (1-\lambda_k)(P_k^*)^{-1} \end{aligned} \tag{8}$$

The information data update (7) is an inverse equivalent of (2) and the regularised information time update (8) is identical to (5) as follows directly from (6) substituted into (5).

One of possible square root systolic implementations of (7), (8) is proposed in next section.

2 Square Root Systolic Implementation

We propose implementation of the regularised information filter (7), (8) as an estimator implementing recursive QR decomposition technique [17], based on Gentleman-Kung array [4], [16], [18].

The QR algorithm [2, 4] implements the data update of the factorised information matrix $P_{k|k}^{-1}$ (7) and the data update of vector $r_{k|k}$ defined

$$
\begin{aligned}
P_{k|k}^{-1} &\hat{=} R_{k|k}^T\, R_{k|k} \\
r_{k|k} &\hat{=} R_{k|k}\, \hat{\theta}_{k|k} = R_{k|k}^{-T} d_{k|k}
\end{aligned}
\tag{9}
$$

where $R_{k|k} > 0$ is upper triangular matrix.

2.1 Regularised QR Factorisation

Let us define analogically to (9) the factorisation

$$
\begin{aligned}
(P_k^*)^{-1} &\hat{=} R_k^{*T}\, R_k^* \\
r_k^* &\hat{=} R_k^*\, \theta_k^* = (R_k^*)^{-T} d_k^*.
\end{aligned}
\tag{10}
$$

$$
\begin{bmatrix} P_{k|k}^{-1} & d_{k|k} \\ d'_{k|k} & * \end{bmatrix} \hat{=} \tilde{R}_{k|k}^T \tilde{R}_{k|k} = \begin{bmatrix} R_{k|k} & r_{k|k} \\ 0 & * \end{bmatrix}^T \begin{bmatrix} R_{k|k} & r_{k|k} \\ 0 & * \end{bmatrix} \tag{11}
$$

$$
\begin{bmatrix} (P_k^*)^{-1} & d_k^* \\ d_k^{*T} & * \end{bmatrix} \hat{=} \tilde{R}_k^{*T} \tilde{R}_k^* = \begin{bmatrix} R_k^* & r_k^* \\ 0 & * \end{bmatrix}^T \begin{bmatrix} R_k^* & r_k^* \\ 0 & * \end{bmatrix} \tag{12}
$$

where R_k^* ; $\tilde{R}_{k|k}$; $\tilde{R}_k^*$ are upper triangular and $*$ is a "don't care" scalar term.

Then the regularised information filter (7)–(8) can be implemented in the square root factorised form (13),(14).

Square Root Regularised Data Update.

$$
\tilde{R}_{k|k}^T \tilde{R}_{k|k} = \tilde{R}_{k|k-1}^T \tilde{R}_{k|k-1} + \begin{bmatrix} \varphi_k \\ y_k \end{bmatrix} \begin{bmatrix} \varphi_k \\ y_k \end{bmatrix}' \tag{13}
$$

Square Root Regularised Time Update. Specified by $\tilde{R}_k^* \geq 0$; $0 < \lambda_k \leq 1$.

$$
\tilde{R}_{k+1|k}^T \tilde{R}_{k+1|k} = \lambda_k \tilde{R}_{k|k}^T \tilde{R}_{k|k} + (1 - \lambda_k) \tilde{R}_k^{*T} \tilde{R}_k^* \tag{14}
$$

It follows directly from compare with (7), (8) and the definition of factorisation (9) and (10)-(12).

2.2 Systolic Implementation of the Regularised Estimator

The proposed regularised parallel estimator implements (13) and (14). This is in fact the standard systolic array for the QR filtering with $(n) * (n+1)/2$ cells. The cells store and update the triangular matrix $R_{k|k}$ and regression parameter estimate is updated in the transformed form as $r_{k|k}$ on the extended row of n cells.

The array requires $n+1$ time steps between measurements to complete the recursion and can be formally described as follows.

Systolic Regularised Estimator Array.

$$\left|\begin{array}{c} \varphi_k \\ y_k \\ \Leftarrow 1 \Rightarrow \end{array}\right| \quad \begin{array}{c} \mathrm{Q} \\ \Longrightarrow \\ \end{array} \quad \left[\begin{array}{c} R^T_{k|k-1} \rightarrow R^T_{k|k} \\ r^T_{k|k-1} \rightarrow r^T_{k|k} \\ \text{QR array} \end{array}\right] \Longrightarrow \hat{e}_{k|k} \tag{15}$$

$$\left|\begin{array}{c} (1-\lambda_k)^{1/2} R^{*T}_k \\ (1-\lambda_k)^{1/2} r^{*T}_k \\ \Leftarrow n \Rightarrow \end{array}\right| \quad \begin{array}{c} \mathrm{Q...Q} \\ \Longrightarrow \\ \leftarrow n \rightarrow \end{array} \quad \left[\begin{array}{c} \lambda_k^{1/2} R^T_{k|k} \rightarrow R^T_{k+1|k} \\ \lambda_k^{1/2} r^T_{k|k} \rightarrow r^T_{k+1|k} \\ \text{QR array} \end{array}\right] \tag{16}$$

The array implements one QR update (15) of the data measurement (13) and n additional QR updates (16) representing the "addition" of $(1-\lambda_k)^{1/2} R^{*T}_k$ and $(1-\lambda_k)^{1/2} r^{*T}_k$ as n columns of "artificial" data to implement the time update (14).

If additional information about the changes of the parameter estimates is available, it can be built into the regularising terms r^*_k; $R^*_k \geq 0$ related to θ^*_k and $(P^*_k)^{-1}$ through the definitions (6) (12).

The regularised array (15)-(16) provides as a by-product of the data update (15) the output filtration error $\hat{e}_{k|k}$ equally to the Gentleman-Kung array [4], [16], [18].

2.3 Regression Parameter Estimates Extraction

The function of array (16) can be combined with the n data updates necessary for the parameter estimate extraction technique known as " weight flushing " [17, 18].

Systolic implementation of parameter extraction.

$$\left|\begin{array}{c} I_n \\ 0 \\ \Leftarrow n \Rightarrow \end{array}\right| \quad \Longrightarrow \quad \left[\begin{array}{c} R^T_{k|k}(\text{frozen}) \\ r^T_{k|k}(\text{frozen}) \\ \text{QR array} \end{array}\right] \Longrightarrow \hat{\theta}^T_{k|k} \tag{17}$$

The regression parameter estimates can be captured as a serial stream of n items from the bottom of the array.

3 Block Regularised Algorithm

In [21], [22] a novel block regularised algorithm has been described. It is in fact a modification of the regularised exponential forgetting algorithm which allows parallel implementation.

The Block Regularised Algorithm [21] overcomes both the *Pipelining* and *Complexity* problems arising in the implementation of the regularised exponential algorithm (8)-(7). It is achieved by the additional assumption about the alternative covariance P^* and the alternative parameter vector θ^*. If they both remain constant within an block ie.

$$P^* = P^*_{k-N} = P^*_{k-N+1} = ... = P^*_{k-1} \tag{18}$$

$$\theta^* = \theta^*_{k-N} = \theta^*_{k-N+1} = ... = \theta^*_{k-1} \tag{19}$$

the computation will be simplified as the $(P^*)^{-1}$ and $(P^*)^{-1}\theta^*$ terms can be accumulated and added in every N-th iteration as shown in (20)-(22)

$$\begin{aligned} P^{-1}_{k|k} &= \lambda_{k-1}...\lambda_{k-N}P^{-1}_{k-N|k-N} + \\ &+ \lambda_{k-1}...\lambda_{k-N+1}\varphi_{k-N+1}\varphi'_{k-N+1} + \lambda_{k-1}...\lambda_{k-N+2}\varphi_{k-N+2}\varphi'_{k-N+2} \\ &+ ... + \lambda_{k-1}\varphi_{k-1}\varphi'_{k-1} + \varphi_k\varphi'_k + \lambda_{[k-N,k]}(P^*)^{-1} \end{aligned} \tag{20}$$

$$\begin{aligned} d_{k|k} &= \lambda_{k-1}...\lambda_{k-N}d_{k-N|k-N} \\ &+ \lambda_{k-1}...\lambda_{k-N+1}\varphi_{k-N+1}y_{k-N+1} + \lambda_{k-1}...\lambda_{k-N+2}\varphi_{k-N+2}y_{k-N+2} \\ &+ ... + \lambda_{k-1}\varphi_{k-1}y_{k-1} + \varphi_k y_k + \lambda_{[k-N,k]}(P^*)^{-1}\theta^* \end{aligned} \tag{21}$$

$$\hat{\theta}_{k|k} = P_{k|k}d_{k|k} \tag{22}$$

where $0 < \lambda_{[k-N,k]}$ is a scalar accumulated weight given by (23)

$$\begin{aligned} \lambda_{[k-N,k]} &= \lambda_{k-1}...\lambda_{k-N+1}(1-\lambda_{k-N}) + \lambda_{k-1}...\lambda_{k-N+2}(1-\lambda_{k-N+1}) \\ &+ ... + \lambda_{k-1}(1-\lambda_{k-2}) + (1-\lambda_{k-1}). \end{aligned} \tag{23}$$

The regularised exponential algorithm is thus modified by these assumptions (18), (19) into The Block Regularised Algorithm [21].

In the block regularised algorithm the alternative covariance matrix is added only every N-th iteration but in such a way as to achieve *exactly* the same results as the algorithm which adds it every step. Exactly the same parameter estimates will be provided, but they will be only available at every N-th iteration.

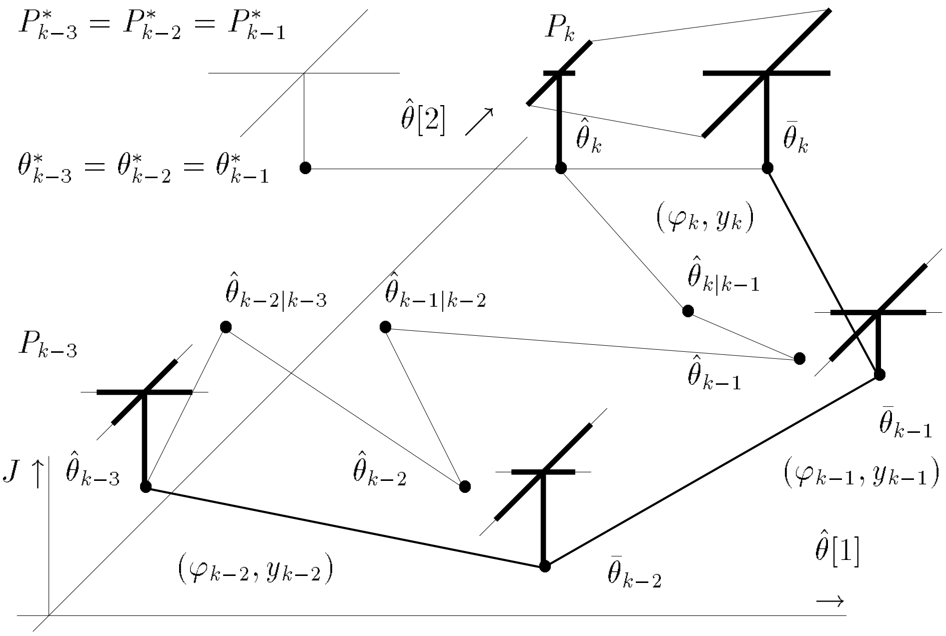

Figure 1: Block Regularised Algorithm for time $k-3$ to k

Fig. 1. demonstrates the block regularised algorithm. For a clearer comparison, there is a trace of the parameter estimates of the original regularised exponential algorithm. Notice, that both algorithms start and end at precisely the same conditions. The block regularised algorithm reduces considerably the number of computations necessary to get to the final point.

Note that θ^* , P^* can change in the time instants $k, k+N, \ldots$ but must remain constant in the intermediate N steps. Usually $N = n$, where n is the number of estimated parameters.

Using the QR orthogonal rotation concept (15), the factorised block restricted algorithm can be rewritten into the final QR form, updating the upper triangular matrix $R_{k|k}$ and vector $r_{k|k}$.

The final QR block restricted algorithm (24) is a sequence of QR orthogonal rotations of the type (15), applied for the real data measurements. It is followed by the accumulated regularisation of the type (16) performed on the boundaries of blocks, combined with the weight flushing (17) of the current parameter estimates.

Block Regularised Algorithm as a sequence of the QR orthogonal rotations is specified by the parameters $R^* \geq 0; \lambda_{k-N}, .., \lambda_{k-1}; 0 < \lambda_j \leq 1; N \geq n$.

$$
\begin{array}{ll}
\bar{R} = R_{k-N|k-N} \; ; \; \bar{r} = r_{k-N|k-N} & (a) \\
\textbf{cycle for } \; j \; = \; N-1, N-2, ..., 0 : & \\
\begin{bmatrix} \bar{R} & \bar{r} \\ 0 & * \end{bmatrix} := Q_{k-j-1} \begin{bmatrix} \sqrt{\lambda_{k-j-1}}\,\bar{R} & \sqrt{\lambda_{k-j-1}}\,\bar{r} \\ \varphi'_{k-j} & y_{k-j} \end{bmatrix} & (b) \\
\textbf{end of cycle for } j & \\
\bar{R}_k = \bar{R} ; \; \bar{r}_k = \bar{r} & (c) \\
\bar{\theta}_{k-N} = \bar{R}^{-1}_{k-N} \bar{r}_{k-N} & (d) \\
\theta^* \equiv \bar{\theta}_{k-N} & (e) \\
r^* = R^* \theta^* & (f) \\
\textbf{cycle for } \; i \; = \; n, n-1, ..., 1 : & \\
\begin{bmatrix} \bar{R} & \bar{r} \\ 0 & * \end{bmatrix} := Q_{i,k} \begin{bmatrix} \bar{R} & \bar{r} \\ \sqrt{\lambda_{[k-N,k]}}\, R^*[i] & \sqrt{\lambda_{[k-N,k]}}\, r^*[i] \end{bmatrix} & (g) \\
\textbf{end of cycle for } i & \\
R_{k|k} = \bar{R} \; ; \; r_{k|k} = \bar{r} \; ; \; (\hat{\theta}_{k|k} = R^{-1}_{k|k} r_{k|k}) & (h)
\end{array}
\qquad (24)
$$

4 Transputer Array Description

The transputer array implementing the algorithm (24) is plotted in Fig. 2. The detailed cell description depends on the specification of the basic "rank-one" update at the cell level. See [16], [17], [4], or [18] for different options. In this paper the QR type of cells [4] is worked out in detail.

Let us discuss first, how the array selects the new position of the alternative estimate θ^* for each block of N data measurements in (24).

The basic areas marked in Fig. 2. by ABC and BCDE store and update matrix R_k and r_k (respective $\bar{R}$ and $\bar{r}$) as in (24b) and (24g).

The automatic selection of θ^* for each block requires a sequence of additional computations (24d)-(24f). The equation (24d) computes the parameter estimate $\bar{\theta}_{k-N}$. In (24e) the selection of θ^* is made. Consequently the alternative quadratic form $J^*(\theta_k)$ for the block $[k-N, k]$ will have its minimum positioned in the point $\theta^* \equiv \bar{\theta}_{k-N} = \bar{R}^{-1}_{k-N} \bar{r}_{k-N}$.

Notice, that by using "old" variables $\bar{R}^{-1}_{k-N}$; $\bar{r}_{k-N}$ which have been available N time steps before, $\bar{\theta}_{k-N}$ can be computed off-line at this time.

Finally, once the alternative parameter θ^* and R^* for the block $[k-N, k]$ are chosen, the vector r^* as a matrix by vector product $R^*\theta^*$ is computed in (24f). The vector r^*, weighted by $\lambda_{[k-N,k]}$ (23), is finally added to $\bar{r}_k$ in (24g) to realise the accumulated regularisation.

The block regularised algorithm (24) produces as a side effect the parameter estimates $\bar{\theta}_{k-N}$; $\bar{\theta}_k$ ... used for the automatic positioning of the alternative quadratic form $J^*(\theta_k)$. These parameters have to be distinguished from the parameter estimate $\hat{\theta}_{k-N|k-N}$; $\hat{\theta}_{k|k}$ ... which can be computed using the formulae (22h). In Fig.3. the difference between $\bar{\theta}_k$ and $\hat{\theta}_k$ is illustrated.

The advantage of selecting the alternative quadratic form $J^*(\theta_k)$ in the semi constant points $\bar{\theta}_{k-N}$; $\bar{\theta}_k$ .... is an efficient parallel implementation. The computation of $\bar{\theta}_{k-N}$ (24d) can be started in parallel with the accumulated regularisation (24g) computation in the previous $[k-2N, k-N]$ block. In contrast, computation of $\hat{\theta}_{k-N|k-N}$ (24h) can start only after finishing of the accumulated regularisation (24g) of the transformed variables in the previous block ie. after all the computations of the block $[k-2N, k-N]$ and therefore it would cause a decrease of the throughput of the final pipelined implementation.

The extended row of n cells, DEFG, performs the matrix by vector multiplication (24f). The matrix R^* is loaded from the north to ABC area. When loaded, R^* starts to move (with the element $R^*[n,n]$ first) through BCDE in the easterly direction, to realise (24f) in DEFG.

The vector θ^* moves (the item $\theta^*[n]$ first) to the array marked DEFG from the south to realise (24f). The resulting product $r^*[i]$ emerges from the northerly output of the DEFG array. It is directly loaded back to the north input of the array BCDE (synchronously with the rows $R^*[i]$ loaded to the array ABC) to realise the accumulated "addition" (24g).

Simultaneously, both arrays ABC and BCDE provide the parameter estimate extraction known as "weight flushing" [17, 18] (24d) by inserting (n,n) unit matrix and a column of zeros on the array data inputs (synchronously with the rows $[\,R^*[i]\,,\,r^*[i]\,]$). The parameter extraction "filter" works with the "frozen" matrix $\bar{R}_{k-N}$ and vector $\bar{r}_{k-N}$, stored in the areas ABC and BCDE.

4.1 Systolic Array Characteristics

The concept of the systolic implementation of the regression parameter estimator is presented in Fig. 2. Fig. 3. presents the use of the input data buffering and multiplexing, to provide approximately 50%th throughput in compare with the systolic triangular RLS filter array.

The cells of the array work in two phases. These phases are distinguished in the cells by the sign of an index j propagated through the array. The value of the index gives to the cells the timing information within both phases.

In the first phase a block of $N \geq n$ data is processed (n is the number of estimated regression parameters) using the standard function of the RLS systolic information filter (N rank-one data updates).

In the second phase the processing of the input data temporary stops and the array computes (in n steps, each with complexity comparable to a rank-one data update) these combined functions:
(1) The computation of the regression parameter estimate by "weight flushing". The parameters emerge as a serial stream of n items.
(2) The positioning of the minimum of the regularising semi-constant quadratic form (the point θ^*) to the point of the recently flushed-out regression paremeter estimate.
(3) The regularisation, accumulated over last N data measurements.

The resulting parallel implementation of the estimator has several remarkable features:

- The transputer implementation is completely pipelined with processor utilisation close to 100%.
- The processes perform only the neighbour–to–neighbour one–directional communication.

5 Conclusion

The block regularised algorithm is described and its square root form derived, which allows parallel transputer implementation under the assumption, that the alternative regularising cost function is constant within data blocks and adjusted to the new value on the boundaries of blocks only. This restriction (see Fig.1.) does not reduce significantly the effectiveness of the general regularised concept because the array implementation adjusts the minimum of the semi-constant alternative cost to follow the current regression parameter estimate.

It is then shown how it could be implemented on a systolic architecture. Fig.2. presents the principle, Fig.3. describes the final structure of a compacted transputer array, with the input data buffering and multiplexors.

The final block regularised transputer architecture (Fig.3.) is half as fast as the standard architecture of McWhirter [4] for the exponentially weighted recursive least squares without regularisation. It should be noted again, that the processes perform nearest-neighbour, one-directional communication. There are no long global connections in Fig.3. Data only passes in one direction on any given link. The feedback associated with adjustment of the new position of the alternative cost function is completely compatible with the pipelined implementation.

The side effect of the parallel square root information filter implementation is that the regression parameter estimates are provided by the accumulated regularisation process on the boundaries of the computational blocks.

The direct use of these parameters can be found for example in the design of adaptive optimal controllers, or in the high resolution adaptive spectral analysis.

References

[1] Bierman, G. J. *Factorisation Methods for Discrete Sequential Estimation.* Academic Press, New York. 1977.

[2] Gentleman,W.M., H.T.Kung. Matrix Triangularisation by Systolic Arrays. Proc. SPIE, Vol.298, *Real Time Signal Processing IV*, 1981, pp. 19-26.

[3] Peterka, V. Bayesian approach to system identification In P. Eykhoff (Ed.) *Trends and Progress in System Identification*, Pergamon Press, Eindhoven, Netherlands, 1981.

[4] Mc Whirter, J.G. Recursive Least Squares minimisation using a Systolic Array, Proc. *SPIE*, Vol. 431, *Real Time Signal Processing VI*, pp. 105, 1983.

[5] Ljung, L., T. Söderström. *Theory and Practice of Recursive Identification.* Cambridge, MA, MIT Press. 1983.

[6] Hägglund, T., The problem of forgetting old data in recursive estimation. *Proceedings of the IFAC Workshop on Adaptive Systems in Control and Signal Processing*, San Francisco, Ca., 1983.

[7] Ljung, L. and Gunnarsson, S., Adaptation and tracking in system identification — a survey. *Automatica* **26**, 7–21, 1990.

[8] Parkum, J.E., Poulsen, N.K. and Holst, J., Recursive forgetting algorithms. *Int. J. Control* **55**, 109–128, 1992.

[9] Kulhavý, R. and M. Kárný. Tracking of Slowly Varying Parameters by Directional Forgetting. *Preprints 9th IFAC World Congress*, Budapest. **X**, pp. 178-183, 1984.

[10] Gaston F. M. F., G. W. Irwin. The systolic approach to information Kalman filtering, *Int. Jour. Control*, Vol. 15., No. 1, pp. 225-228, July 1989.

[11] Gaston F.M.F. ,G. W. Irwin, J. G. Mc Whirter. Systolic Square Root Covariance Kalman Filtering, *Jour. of VLSI Sig. Processing* , No. 2, pp. 37-49, 1990.

[12] Kadlec, J. A Joint Criterion for Exponential Diretional and Mixed Parameter Tracking. *Preprints of the International Symposium on Adaptive Systems in Control and Signal Processing (ACASP '92) Conference*, Grenoble, France, June 1992, pp. 687-692.

[13] Kulhavý, R. Directional Tracking of Regression-Type Model Parameters. *Preprints of the 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing*, Lund. pp. 97 - 102, 1986.

[14] Kulhavý, R. Restricted exponential forgetting in real-time identification. *Automatica*, **23**, 589–600, 1987.

[15] Kulhavý, R.; M. B. Zarrop: On a general concept of forgetting. To appear in *International Journal of Control*

[16] Ling, F., D. Manolakis, and J.G. Proakis. A recursive modified Gram-Schmidt algorithm for least squares estimation. *IEEE Trans., ASSP-34*, 829-835, 1986.

[17] Mc Whirter, J.G. Algorithmic engineering - an emerging discipline, *SPIE* Vol. 1152, Advanced Algorithms and Architectures for Signal Processing IV, pp. 2-15, 1989.

[18] Shepherd,T.J., J. G. Mc Whirter and J. E. Hudson. Parallel Weight Extraction from a Systolic Adaptive Beamformer. Proc. IMA Int. Conf. on Mathematics in Signal Processing, Warwick, Dec. 1988, pp. 775-790, 1988.

[19] Kadlec, J. A Recursive Gramm-Schmidt Identification with Directional Tracking of Parameters. *Preprints of the the 9-th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, Budapest, Hungary, 8-12 July, 1991, pp. 1707-1712.

[20] Kadlec, J., F.M.F. Gaston, G. W. Irwin. Implementation of the Regularised Parameter Estimator. *Proceedings of the 1992 IEEE Workshop on VLSI and Signal Processing*, Napa, CA, October 28-31, 1992.

[21] Kadlec, J., F.M.F. Gaston, G. W. Irwin, (1992). Parallel Implementation of Restricted Parameter Tracking. Poster na *Third IMA International conference on Mathematics in Signal Processing*, by University of Warwick, UK, December 15-17, 1992. Submitted for publication in *Mathematics in Signal Processing* , J. Mc Whirter editor, Oxford publishers press.

[22] Kadlec, J., F.M.F. Gaston, G. W. Irwin, (1993). The block regularised parameter estimator and its parallel implementation. Submitted to the The IFAC Journal "*Control Engineering Practice* "

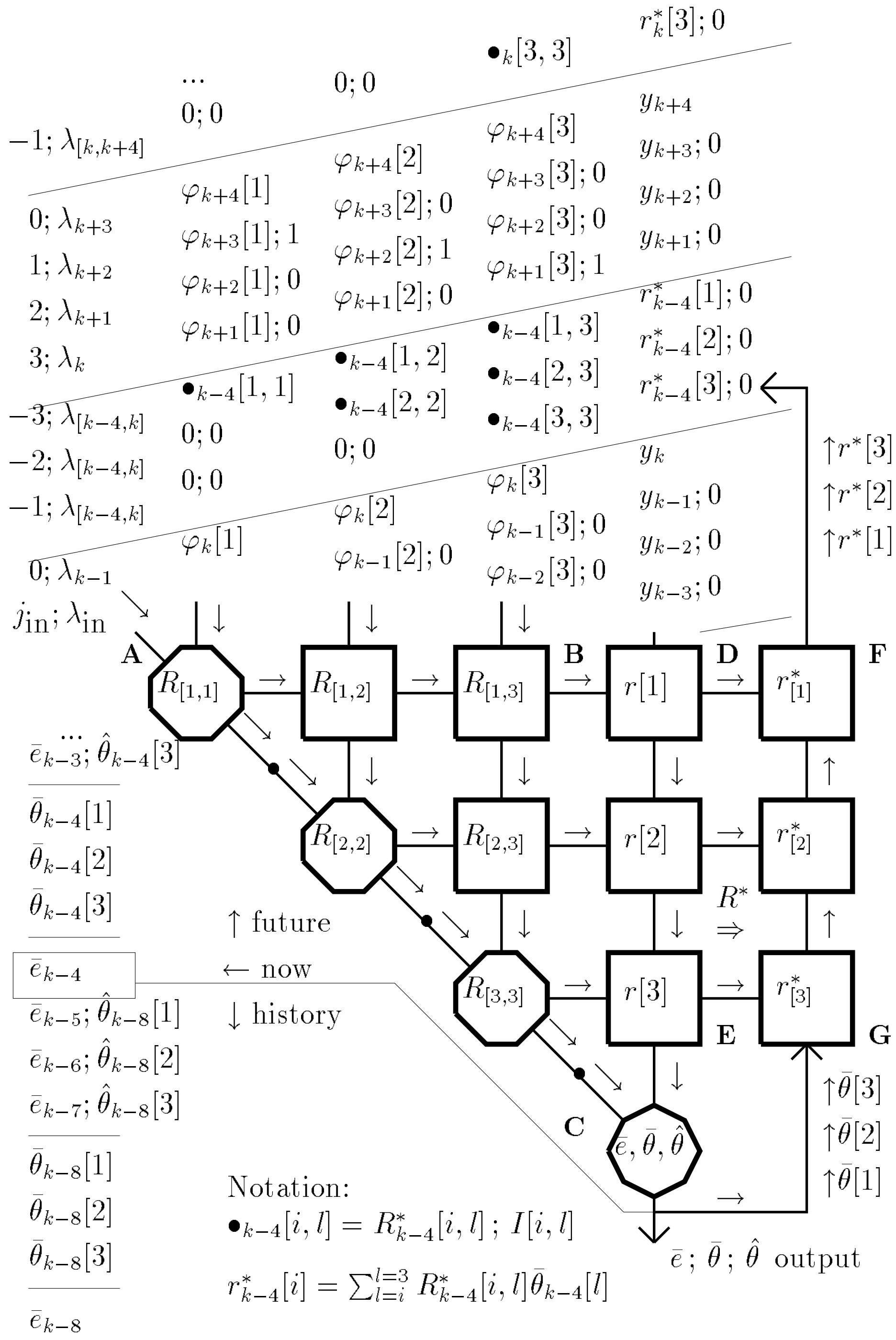

Figure 2: Block Regularised QR Array Concept

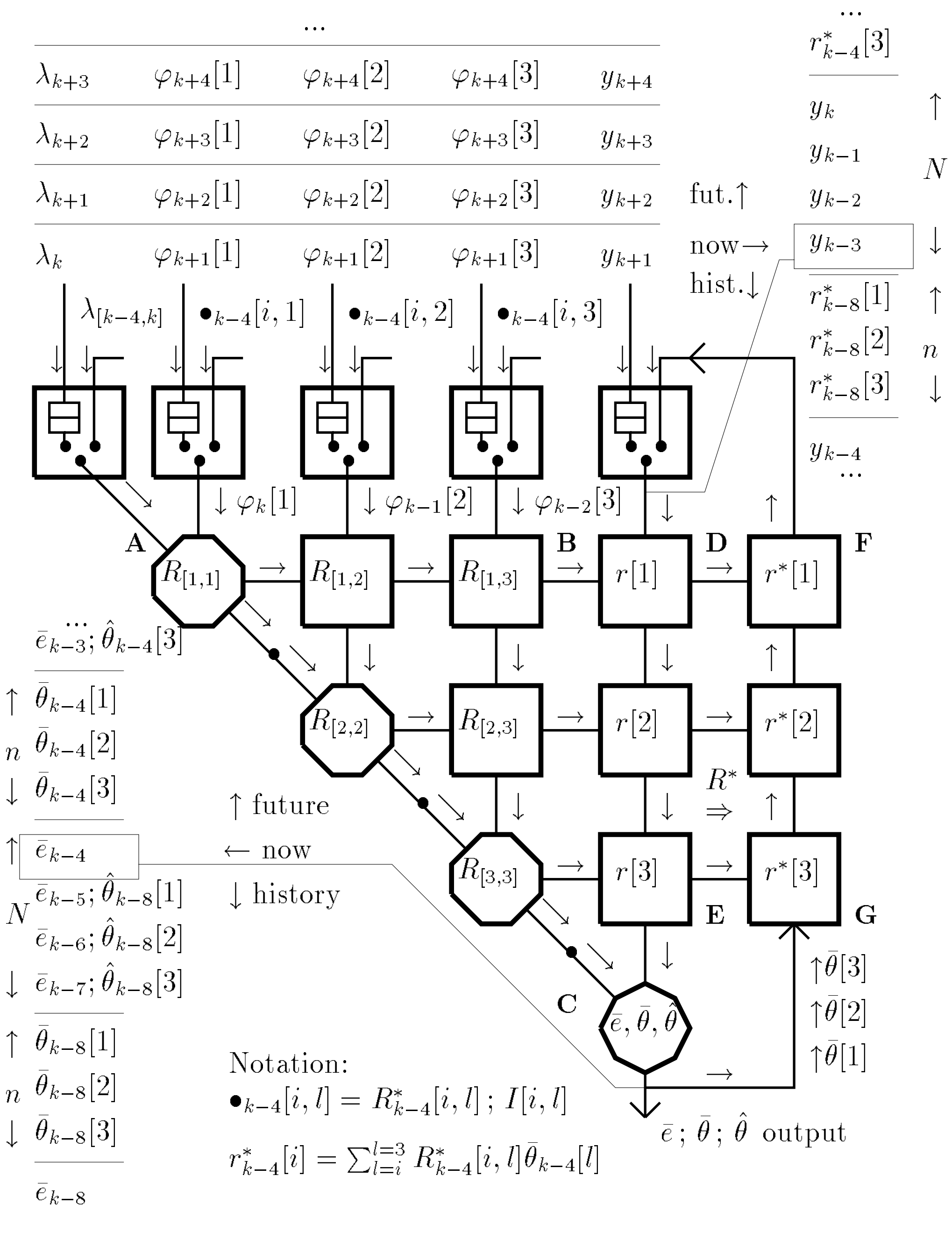

Figure 3: Transputer implementation of Block Regularised QR Array