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, F. M. F. Gaston, G.W. Irwin

Regularised Lattice-Ladder Adaptive Filter

No. 1793 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

Regularised Lattice-Ladder Adaptive Filter

J. Kadlec, F. M. F. Gaston, G.W. Irwin

Control Research Group,
Department of Electrical and Electronic Engineering,
The Queens University of Belfast,
Stranmillis Road, Belfast, BT9 5AH, Northern Ireland.

Abstract. A novel regularised algorithm is presented for the adaptive recursive least-squares Ladder-Lattice prediction and filtering. The design of the filter is based on formulating the Lattice section as a special implementation of the factorised recursive least-squares algorithm, assuming the data and the parameter tracking method meet certain specific criteria.

The proposed estimator provides on a linear array of n processors a data throughput comparable with the triangular systolic implementation of the recursive square root information filters on $n(n-1)/2$ processors.

Unlike the standard Recursive Least Squares Lattice-Ladder algorithm the proposed regularised estimator parameter tracking technique can be interpreted as introduction of a set of forgetting factors tuned to control the character of the adaptation. The estimator reduces considerably the numerical problems related to overparametrisation or poor data.

Keywords. Identification; least-squares estimation; adaptive systems; forgetting; recursive algorithms.

1. INTRODUCTION

Lattice - Ladder predictors, estimators and filters have been studied extensively in the 80's [14]. The wide use of these structures in various signal processing applications is mainly due to their good numerical properties [9] (in comparison with their fast transversal counterparts), decoupling between various stages and the possibility of pipelined, VLSI implementation. Traditionally, the Lattice – Ladder structures have been described in terms of reflection coefficients and ladder gains. These coefficients are recursively updated in time in an effort to minimize the squared error between the system and the estimator output. The problem is known as linear regression, and estimation is based on the recursive minimization of a least-squares criterion (RLS).

Interpreted in terms of the corresponding state-space realization [6] the Ladder section works with a state vector that has a diagonal covariance matrix. This is achieved by a linear, time-varying transformation of the model representation [9]. Regression parameters are transformed to ladder gains, regressor data to uncorrelated, so called backward, errors and the model output is built up as their linear combination.

This change of the model representation can be implemented by the recursive QR algorithm [3], the Recursive Modified Gram–Schmidt (RMGS) algorithm [7] or the factorised covariance filter (UD) [8], [9]. These algorithms require $O(n^2)$ multiplication and divisions (MAD) between measurements for the recursive identification of n parameters. The data update of these algorithms can be mapped on to a triangular array [2], [7] of $O(n^2)$ processors and thus reduce the time between measurements from $O(n^2)$MAD to $O(n^0)$MAD, using the pipelining of subsequent data update tasks.

The Lattice structure is able to realize the transformation of the model representation (necessary for the Ladder part which is common in all the above filters) on a linear array of only $O(n)$ processors, reducing the time between measurement to $O(n^0)$MAD too. This is possible because the Lattice approach assumes that the data regressor exhibits shifting properties which is often the case in signal processing.

But there is a more serious requirement imposed by the Lattice transformation of the model representation; the shifting properties of the data have to be reflected in the shifting properties of the updated matrices. This significantly restricts the choice of adaptation techniques used in the conventional

RLS Lattice. Exponential age weighting with a fixed, constant weighting factor is the most commonly used technique.

This restriction imposed by the standard RLS Lattice prevents consideration of the character of information contained in previous data, or prior information about the possible values of the estimated parameters [11]. Some parameters, namely the variances of the backward errors and the correlation coefficients may become extremely small, giving rise to numerical problems. This could happen if the model set contains too many parameters, or if the input signal is not informative enough. The data produced by a constant linear control law, or a pure sinusoidal input can be examples of these situations.

Consequently, it is important to have built into the algorithm a device to prevent these situations in adaptive identification, control or monitoring, since it is not very realistic to assume that the input will remain informative in practical applications. In these cases we must somehow guarantee that the parameters will remain nonsingular. This is called the regularization of the algorithm.

The standard technique associated with factorized algorithms is to test certain parameters and to stop their recursion if they are approaching singular conditions. See [9] for the application of this technique in the UD filter and Lattice - Ladder implementation. This technique protects the algorithm from numerical problems. Its drawback is that it is not compatibile with the basic requirements of the transformation of the model representation in the parallel implementation and the saving of the shifting properties in the Lattice section. It can affect the "fragile" transformation of the regression model representation and can produce big errors in the corresponding original regression parameter estimates. That is why the method is switched-on only in nearly singular situations. But these near to singular conditions correspond to a large uncertainty about some of parameters. In other words, part of the algorithm could heavily depend on the next measurements, just like a filter relying on an extremely short window of data.

If we allow the algorithm to approach these conditions, it can eventually produce serious secondary problems; output bursting in adaptive control close-loop applications, or uncertain estimates of spectra in high resolution spectral analysis are only two examples.

For these reasons several other techniques for coping with poor data have been proposed, especially in adaptive control. In these implementations more space has remained for parameter tracking specifications, because (in opposite to signal processing) neither the restriction imposed by the systolic array implementation, nor the Lattice related shifting properties has been assumed apriori. Directional forgetting [10, 12], the constant trace approach, restricted exponential forgetting [4] and shift invariant directional forgetting are only a few examples of these techniques.

If directional forgetting, or the constant trace approach, is mapped onto the Gentleman-Kung array [2], the global feedback loops in the signal flow considerably degrade the data throughput of the array, because the original formulation has been designed primarily for efficient implementation on one processor.

Other approaches [4] affect the transformation of the model representation and consequently disable efficient parallel implementation.

Their formulation has to be revised especially in the Lattice - Ladder type of parallel implementation where the shifting properties of the data and updated matrices are the additional vital assumption in their derivation.

Our paper addresses this issue. We propose a solution to both contradictory requirements of the adaptive filter. The proposed array combines an efficient Lattice implementation with well defined regularization.

1.1 The Concept of Regularised Filter

The proposed regularized Lattice – Ladder filter is designed by utilizing of specification of the general probabilistic concept of parameter tracking introduced in [4]. The related recent work [16] provides the asymptotical analysis of the approach as well as how it relates to the underlying statistical results. From the implementation point of view, this concept can be understood as the introduction of an additional, alternative reference covariance matrix exhibiting the role of a regularizing point for the evolution of transformed parameters. Various choices of the alternative covariance matrix specify different parameter tracking methods [10, 12, 13].

Neither the parallel implementation aspects nor the Ladder-Lattice were addressed in [4] and [16]. The generality of the approach makes it to be a suitable tool for the design of the regularized adaptive filter respecting the data flow requirements of the pipelined VLSI implementation as well as the additional basic requirement of the shifting properties, assumed by the Lattice - Ladder structures.

2. PROPAGATION OF PROBABILITIES

We consider a stochastic single output $y(t)$ system modelled by a m-th order regression model

$$y(t) = \theta'_m(t)\varphi_m(t) + v_m(t), \; v_m(t) \sim N(0, \sigma^2_m) \tag{1}$$

where m denotes an index of n generally different identification tasks, t is discrete time, $\theta_m(t)$ is a vector of m unknown possibly time variable regression parameters, $v_m(t)$ denotes the normal white noisewith variance σ^2_m. The data regressor vector $\varphi_m(t)$ is assumed to be composed of m time-shifted scalar items.

$$\begin{aligned} \varphi'_n(t) &= [\varphi(t), \varphi(t-1), \; \dots \; , \varphi(t-n+1)] \\ &= [\varphi'_m(t), \varphi(t-m+1), \; \dots \;] \\ &= [\varphi(t), \varphi'_m(t-1), \; \dots \;] \end{aligned} \tag{2}$$

The item $\varphi(t)$ denotes the shifted system input - in the finite input response (FIR) applications, or the shifted system output $y(t-1)$ in the case of auto regression (AR) filtering. The derivations can be extended straightforwardly to the multi-variable case, and the block shifting properties of the data regressors. We have restricted this discussion to the scalar case in order to illustrate the basic principles of the unified Lattice Ladder factorization.

2.1 Propagation of Conditional Probability

Our incomplete knowledge of the system behavior is formally expressed through a vector of unknown, time-varying parameters $\theta_m(t)$ characterised by the probability density function (p.d.f.) $p(\theta_m(t) \,|\, t-1)$ where the abbreviation $|\, t-1$ denotes the conditioning on data measurement up to the time $t-1$.

The complete recursion of the adaptive estimator can be written in two steps. The first step is the *data update*. It forms the common part of a broad class of adaptive algorithms.

$$p(\theta_m(t) \,|\, t-1) \rightarrow p(\theta_m(t) \,|t) \tag{3}$$

The second step, *time update* makes the estimator adaptive.

$$p(\theta_m(t) \,|\, t) \rightarrow p(\theta_m(t+1) \,|\, t) \tag{4}$$

In this paper we use for modeling the time update step the forgetting operator (5). The operator has been formulated within a more general statistical framework in [4] and analyzed in [16].

2.2 Regularised Exponential Forgetting

$$\begin{aligned} p(\theta_m(t+1)|t) &\propto \{p(\theta_m(t)|t)\}^{\lambda}\,\{p^*(\theta_m(t+1)|t)\}^{1-\lambda} \\ E\,\{(p^*(\theta_m(t+1)\,|\,t)\} &= E\,\{p(\theta_m(t)\,|t)\}\,. \end{aligned} \tag{5}$$

where $\propto$ is for "equal up to a scaling factor" and $E\{\,\}$ is for the mean value.

The operator (5) constructs $p(\theta_m(t+1)\,|\,t)$ from

- the last conditional p.d.f. $p(\theta_m(t)\,|t)$ which represents the prediction of $p(\theta_m(t+1)\,|\,t)$ for the case of no parameter changes
- a reference p.d.f. $p^*(\theta_m(t+1)\,|\,t)$ which describes (in probabilistic terms) the available knowledge of unknown parameters at the next time instant, based on both prior information and information contained in the data (observed up to the time t)

by mixing both kinds of information together weighted by constants λ and $(1-\lambda)$ respectively.

In [4] λ is estimated in time and mean $E\,\{(p^*(\theta_m(t+1)|t)\}$ can be chosen freely.

Here we restrict λ to a constant value and will call it the exponential weighting factor, because it corresponds to the exponential age-weighting concepts used in signal processing and RLS applications.

Note that standard exponential age-weighting corresponds to the operator

$$\begin{aligned} p(\theta_m(t+1)\,|\,t) &\propto \{p(\theta_m(t)\,|t)\}^{\lambda} \\ E\,\{(p(\theta_m(t+1)\,|\,t)\} &= E\,\{p(\theta_m(t)\,|t)\} \end{aligned} \tag{6}$$

suggested by [1].

The operator (5) specializes to (6) if the noninformative alternative p.d.f. $p^*(\theta_m(t+1)|t) \propto 1$ is selected.

2.3 Normal Distribution of Parameters

The computations are considerably simplified if we also assume the distribution of parameters $p(\theta_m(t)|t)$ and the reference p.d.f. $p^*(\theta_m(t+1)|t)$ are both normal.

Assuming that the estimated system can be modelled by (1), the time update is specified by (4), and the conditional p.d.f. $p(\theta_m(t)|t)$ and the reference p.d.f. $p^*(\theta_m(t+1)|t)$ are both normal, with covariance matrices $P_m(t)$ and $P^*_m(t{+}1|t)$ and mean values $\hat{\theta}^*_m(t{+}1|t) = \hat{\theta}_m(t)$,then the p.d.f. $p(\theta_m(t)|t)$, as well as, the p.d.f. $p(\theta_m(t+1)|t)$ are also again normal. For proof see e.g. [1] or [4].

Consequently, the functional recursion can be replaced by the algebraic recursion of mean $\hat{\theta}_m(t|t{-}1) \to \hat{\theta}_m(t) \to \hat{\theta}_m(t{+}1|t)$ and the covariance matrix $P_m(t|t{-}1) \to P_m(t) \to P_m(t{+}1|t)$. The related algebraic Kalman filter – like recursions are summarized now.

2.4 The Recursive Algorithm

2.4.1 Data Update

$$\begin{aligned}
\hat{e}_m(t|t{-}1) &= y(t) - \hat{\theta}'_m(t|t{-}1)\varphi_m(t) \\
P_m(t) &= P_m(t|t{-}1) - \frac{P_m(t|t{-}1)\varphi_m(t)\varphi'_m(t)P_m(t|t{-}1)}{1+\varphi'_m(t)P_m(t|t{-}1)\varphi_m(t)} \\
\hat{\theta}_m(t) &= \hat{\theta}_m(t|t{-}1) + P_m(t)\ \varphi_m(t)\hat{e}_m(t|t{-}1)
\end{aligned} \tag{7}$$

Where $\hat{e}_m(t|t{-}1)$ and $\hat{\theta}'_m(t|t{-}1)$ are the apriori error and the parameter estimate of order m respectively and, $P_m(t)$ is the associated covariance matrix. $P_m(t)\ \varphi_m(t)$ is the optimal Kalman Gain vector.

2.4.2 Time Update

$$\hat{\theta}_m(t{+}1|t) = \hat{\theta}_m(t) \tag{8}$$

$$P_m(t{+}1|t) = \left\{\lambda P_m^{-1}(t) + (1-\lambda)(P^*_m(t{+}1|t))^{-1}\right\}^{-1} \tag{9}$$

where $P^*_m(t{+}1|t) > 0$ is the alternative covariance matrix which characterizes the alternative normal reference p.d.f. $p^*(\theta_m(t+1)|t)$, and has a mean value equal to the current estimate

$$\hat{\theta}^*_m(t{+}1|t) = \hat{\theta}_m(t)\ . \tag{10}$$

The weight $0 < \lambda < 1$ is the (generalized) exponential weighting factor.

2.5 Indirect Information Filtering

The mean $\hat{\theta}_m(\cdot)$ and covariance matrix $P_m(\cdot)$ can be propagated in a transformed form too. Here $(\cdot)$ stands for $(t|t{-}1)$, (t) or $(t{+}1|t)$. The information filter updates the regression parameter estimate $\hat{\theta}_m(\cdot)$ indirectly as a vector denoted here $d_m(\cdot)$, where

$$d_m(\cdot) \hat{=} P_m^{-1}(\cdot)\hat{\theta}_m(\cdot)\,. \tag{11}$$

2.5.1 Indirect Data Update

$$\begin{aligned} P_m^{-1}(t) &= P_m^{-1}(t|t{-}1) + \varphi_m(t)\varphi'_m(t) && (12)\\ d_m(t) &= d_m(t|t{-}1) + \varphi_m(t)y(t) && (13)\\ \hat{\theta}_m(t) &= P_m(t)\, d_m(t) \end{aligned}$$

2.5.2 Indirect Time Update

$$\begin{aligned} d_m(t{+}1|t) &= \lambda d_m(t)\ + \\ &\quad +(1-\lambda)(P_m^*(t{+}1|t))^{-1}P_m(t)\, d_m(t) && (14)\\ P_m^{-1}(t{+}1|t) &= \lambda P_m^{-1}(t)\ + (1-\lambda)(P_m^*(t{+}1|t))^{-1} && (15) \end{aligned}$$

where (14) is direct consequence of (8),(9) and (11).

2.6 Nesting of the Lower Order Models

Up to now we have considered the data and time update steps of n different identification tasks indexed by m, where, $\hat{\theta}_m(t)$ is the estimate of the unknown regression parameter vector $\theta_m(t)$.

The advantage of the indirect, or information, filter is better compatibility with the requirements of the parallel implementation which seems to be closely related to the propagation of the nested, lower order problems.

2.7 Complications in Time Update

Propagation of nesting is complicated by the generalized time update (14),(15). This time update reproduces nesting if we simplify it back to standard exponentially weighted RLS specifying $(P_m^*(t{+}1|t))^{-1} = 0$.

But there is a question of how to specify a non–trivial $P_m^*(t+1|t)$ in (14) and (15) and still reproduce nesting of $P_m^{-1}(t+1|t)$ in $P_n^{-1}(t+1|t)$ and $d_m(t+1|t)$ in $d_n(t+1|t)$ for $m = 1, 2, ..., n-1$ and, if possible, how to implement it on a fast systolic array for signal processing applications.

3. PROBLEM FORMULATION

The problem is to specify what are the additional limitations on the choice of $P_m^*(t+1|t)$ in (14), (15) imposed by the Lattice - Ladder implementation to get the data throughput of the estimator equal with the standard exponentially weighted RLS Ladder – Lattice pipelined array [14].

3.1 Nesting and Shifting in Time Update

To be able to close the recursion and use a Lattice implementation we have to assume the time update step will keep both the nesting and the shifting relations for the next step of the recursion. In order to do this we specify the class of possible time updates now.

If

$$\begin{aligned} P_{m+1}^{-1}(t) &= \begin{bmatrix} P_m^{-1}(t) & * \\ * & * \end{bmatrix} = \begin{bmatrix} * & * \\ * & P_m^{-1}(t-1) \end{bmatrix} \\ d_{m+1}^T(t) &= \begin{bmatrix} d_m^T(t) & * \end{bmatrix} = \begin{bmatrix} * & d_m^T(t-1) \end{bmatrix} \end{aligned} \tag{16}$$

is valid. Let us restrict the class of $(P_m^*(t+1|t))^{-1}$ specifying (14), (15) $m = 1, 2, ..., n-1$ and $t = ..., t-1, t$ that
although

$$\begin{aligned} P_{m+1}^{-1}(t+1|t) &= \begin{bmatrix} P_m^{-1}(t+1|t) & * \\ * & * \end{bmatrix} = \begin{bmatrix} * & * \\ * & P_m^{-1}(t|t-1) \end{bmatrix} \\ d_{m+1}^T(t+1|t) &= \begin{bmatrix} d_m^T(t+1|t) & * \end{bmatrix} = \begin{bmatrix} * & d_m^T(t|t-1) \end{bmatrix}. \end{aligned} \tag{17}$$

then the (regularised) indirect information filter of order n (11–15) reproduces the shifting and the nesting of the lower order $m = 1, 2, ..., n-1$ identification of models (1).

4. THE LADDER LATTICE

To implement the filter in the Ladder Lattice form we have to formulate the filter as a special case of the square root factorized implementation of RLS algorithm (12).

4.1 UD Square Root Covariance Filter

The UD square – root covariance filter [8] works with the square root factors of the covariance matrix

For $m = 1, 2, ..., n$:

$$P_n(\cdot) \hat{=} U_n(\cdot) D_n(\cdot) U_n^T(\cdot) \tag{18}$$

where $U_n(\cdot)$ is an upper a triangular matrix with units on the diagonal, $D_n(\cdot)$ is a diagonal matrix with positive elements on the diagonal and $(\cdot)$ is for time index $(t|t{-}1)$, (t) or $(t{+}1|t)$.

4.2 The RMGS Filter

The RMGS factorization [7] implements the recursive update of the factorized information matrix. Using the notation introduced so far we can express it as a special version of the QR factorisation with a separately handled diagonal.

The RMGS algorithm stores and recursively updates the triangular array $U_n^{-1}(\cdot)$ and the vector of the Ladder gains $\hat{K}_n(\cdot)$ defined by

$$\begin{aligned} \hat{K}_n(\cdot) &\hat{=} U_n^{-1}(\cdot)\hat{\theta}_n(\cdot) && (19) \\ &= D_n(\cdot)U_n^T(\cdot)d_n(\cdot) && (20) \end{aligned}$$

where (19) is the definition and (20) follows from (11).

The use of the RMGS or UD factorization (18),(19) is in fact the change of the coordinate basis for linear regression by the particular matrix $U_n(t)$.

$$\hat{e}_n(t) = y(t) - \varphi_n'(t)U_n(t)\, U_n^{-1}(t)\, \hat{\theta}_n(t) \tag{21}$$

The transformed regressor is then the vector of backward prediction errors

$$\varphi_n'(t)U_n(t) = \left[e_0^b(t)\, , e_1^b(t)\, , ..., e_{n-1}^b(t) \right] \tag{22}$$

The vector corresponding to the inverse transformation of the estimates $\hat{\theta}_n(t)$ is the vector $\hat{K}_n(t)$ composed from n Ladder coefficients $\hat{k}_m(t)$.

$$\left[\hat{k}_1(t)\,,\hat{k}_2(t)\,,...,\hat{k}_n(t)\,\right]' \hat{=} U_n^{-1}(t)\,\hat{\theta}_n(t) \tag{23}$$

4.3 The Ladder Section

Because of nesting of data (2), $U_n(t)$ (18) and the Ladder vector $\hat{K}_n(t)$ (19); the output filtration errors of the different orders $\hat{e}_m(t)$ can be computed as the intermediate results in the scalar multiplication

$$\hat{e}_{m+1}(t) \;= y(t) - \left[e_0^b(t)\,,e_1^b(t)\,,...,e_m^b(t)\,\right]\hat{K}_m(t) \tag{24}$$

in the order-recursive formula

$$\hat{e}_{m+1}(t) \;= \hat{e}_m(t) \;+ \hat{k}_{m+1}(t)\;e_m^b(t) \tag{25}$$

starting from $\hat{e}_0(t) \;= y(t)$.

It means, that the by-products $\hat{e}_{m+1}(t)$ are the a-posterior filtration errors belonging to the lower order (nested) identification problems of the regression parameter estimation with m parameters.

4.4 The Lattice Section

The Lattice section can be understood as an implementation of the RMGS information filter or the related UD factorised covariance filter algorithm. The Lattice section replaces, by a linear array, the triangular array required by the RMGS factorization with equivalent data throughput.

4.5 The Reproduction of Nesting

The problem is to specify the alternative information matrix $(P_n^*(t{+}1|t))^{-1}$ characterising the generalized time update (14) and (15) for the efficient systolic implementation reproducing the nesting and the shifting in the factorised square root algorithms too.

Consider the RMGS type of factorisation (19) of the maximal order n specified in the previous section. The related systolic array [7] updates $U_n^{-T}(\cdot)$, $D_n^{-1}(\cdot)$ and vector $\hat{K}_n(\cdot)$.

The algorithm implements an inverse or dual version of the UD factorization, and reproduces the nesting.

According to time update (15) the weighted matrix $(1-\lambda)(P_n^*(t+1|t))^{-1}$ has to be added to $\lambda(P_n(t)\,)^{-1}$ to get $(P_n(t+1|t))^{-1}$.

In the contradiction, according to (19), the nesting requirement (20) and time update (8) requirement, the $U_n(t)$ has to remain unmodified. by the time update

$$U_n(t+1|t) = U_n(t)\ . \tag{26}$$

Fortunatelly both above requirements (15) and (26) can be solved jointly by specifying $P_n^*(t+1|t)$ in a UD factorised form

$$P_n^*(t+1|t) \hat{=} U_n^*(t+1|t) D_n^*(t+1|t) U_n^{*T}(t+1|t) \tag{27}$$

with the selection

$$U_n^*(t+1|t) \hat{=} U_n(t) \tag{28}$$

The assumption (28) considerably simplifies the computation but the basic assumption of the Lattice (17) is still not valid unless we restrict the selection of the diagonal of the alternative covariance matrix to be a unit matrix scaled by a positive constant.

Here are the assumptions which reproduce the nesting as well as the shifting properties of the factorized matrices:

1. The triangular factors of the UD factorization are equal to the current one.

$$U_n^*(t+1|t) = U_n(t) \tag{29}$$

2. The diagonal of the alternative covariance matrix is a unit matrix scaled by a positive constant.

$$D_n^*(t+1|t) = (\alpha^{b*})^{-1} I \tag{30}$$

where I denotes the unit matrix and α^{b*} is a positive scalar constant.

3. The weighting factor λ is assumed to be constant

4.3 Regularised Time Update in UD

The time update in the factorised form includes the regularization of the problem as well as the basic requirement of reproducing the shifting and nesting properties properties required by the Ladder–Lattice implementation.

$$\begin{aligned} U_n(t{+}1|t) &= U_n(t) \\ D_n(t{+}1|t) &= \left\{ \lambda I_n + (1-\lambda)\alpha^{b*} D_n(t) \right\}^{-1} D_n(t) \\ d_n(t{+}1|t) &= \left\{ \lambda I_n + (1-\lambda)\alpha^{b*} D_n(t) \right\} d_n(t) \end{aligned} \tag{31}$$

The Lattice - Ladder factorisation as a special case of the RMGS square root filter is provided in the next section.

5. THE LADDER LATTCE IMPLEMENTATION

5.1 Regularised Time Update in RMGS

The time update (31) in the factorised form includes the regularization of the problem as well as the basic requirement of reproducing the shifting and nesting properties properties required by the Ladder–Lattice implementation.

$$\begin{aligned} \hat{K}_n(t{+}1|t) &= \hat{K}_n(t) \\ U_n^{-1}(t{+}1|t) &= U_n^{-1}(t) \\ D_n^{-1}(t{+}1|t) &= \left\{ \lambda I_n + (1-\lambda)\alpha^{b*} D_n(t) \right\} D_n^{-1}(t) \end{aligned} \tag{32}$$

5.2 Regularised Time Update in Ladder–Lattice

The time update (32) affects the diagonal matrix updated effectively by the lattice factorisation.

$$D_m^{-1}(t{+}1|t) = \lambda D_m^{-1}(t) \ + (1-\lambda)\alpha^{b*} I_m \tag{33}$$

It can be expressed in the scalar Lattice notation by form for $m = 0, 1, ..., n{-}1$:

$$\alpha_m^b(t{+}1|t) = \lambda \alpha_m^b(t) \ + (1-\lambda)\alpha^{b*} \tag{34}$$

The adding of the terms $(1-\lambda)\alpha^{b*}$ can be understood as the different *time variable* weighting factors defined $\lambda_m(t)$ along the diagonal.

$$\lambda_m(t) \hat{=} \lambda + (1-\lambda)\alpha^{b*}/\alpha^b_m(t) \tag{35}$$

The time update step of the related Lattice variables can be written.

$$\alpha^b_m(t{+}1|t) = \lambda_m(t)\alpha^b_m(t) \tag{36}$$

5.3 Regularised Lattice – Ladder Algorithm

The regularised Ladder – Lattice algorithm is summarized now.

Table (37) lists the initialization of the identification, Tables (38) and (39) lists the computations within the complete Lattice and Ladder section.

Initialization in time 0

$$\begin{array}{rcl} & & 0 < \lambda < 1;\ 0 < \alpha^{b*} \\ \alpha^f_m(0) & = & \alpha^b_m(0) = \alpha^{b*} \\ \lambda_m(0) & = & \lambda + (1-\lambda)\alpha^{b*}/\alpha^b_m(0) = 1 \\ e^b_m(0) & = & 0;\ \beta_m(0) = k^f_m(0) = k^b_m(0) = 0 \\ \hat{\beta}_m(0) & = & \hat{k}_m(0) = 0 \end{array} \tag{37}$$

Regularised Ladder - Section

$$\begin{array}{rcl} \hat{e}_0(t) & = & y(t) \\ & & \text{compute (in cell m) } m = 0, 1, \ldots, n-1 \\ \hat{\beta}_{m+1}(t) & = & \lambda_m(t{-}1)\ \hat{\beta}_{m+1}(t{-}1)\ + \\ & & +\hat{e}_m(t)\ e^b_m(t)\ /(1-\gamma_m(t)\ ) \\ \hat{k}_{m+1}(t) & = & -\hat{\beta}_{m+1}(t)\ /\alpha^b_m(t) \\ \hat{e}_{m+1}(t) & = & \hat{e}_m(t)\ + \hat{k}_{m+1}(t)\ e^b_m(t) \end{array} \tag{38}$$

Regularised Lattice - Section

$$
\begin{aligned}
e_0^f(t) &= e_0^b(t) \; = \varphi(t) \\
\gamma_0(t) &= 0 \\
\alpha_0^f(t) &= \alpha_0^b(t) \; = \lambda_0(t\text{-}1)\; \alpha_0^b(t|t\text{-}1) + (\varphi(t))^2 \\
\lambda_0(t) &= \lambda + (1-\lambda)\alpha^{b*}/\alpha_0^b(t) \\
& \quad \text{compute (in cell m) } m = 0, 1, \ldots, n-2 \\
\beta_{m+1}(t) &= \lambda_m(t\text{-}1)\; \beta_{m+1}(t\text{-}1) \; + \\
& \quad + e_m^f(t)\; e_m^b(t\text{-}1) \;/(1-\gamma_m(t\text{-}1)\;) \\
k_{m+1}^f(t) &= -\beta_{m+1}(t)\;/\alpha_m^f(t) \\
k_{m+1}^b(t) &= -\beta_{m+1}(t)\;/\alpha_m^b(t) \\
e_{m+1}^f(t) &= e_m^f(t) \; + k_{m+1}^b(t\text{-}1)\; e_m^b(t\text{-}1) \\
e_{m+1}^b(t) &= e_m^b(t\text{-}1) \; + k_{m+1}^f(t)\; e_m^f(t) \\
\alpha_{m+1}^f(t) &= \lambda_{m+1}(t\text{-}1)\; \alpha_{m+1}^f(t\text{-}1) \; + \\
& \quad + (e_{m+1}^f(t)\;)^2/(1-\gamma_{m+1}(t\text{-}1)\;) \\
\gamma_{m+1}(t) &= \gamma_m(t) \; + (e_m^b(t)\;)^2/\alpha_m^b(t) \\
\alpha_{m+1}^b(t) &= \lambda_m(t\text{-}1)\; \alpha_{m+1}^b(t\text{-}1) \; + \\
& \quad + (e_{m+1}^b(t)\;)^2/(1-\gamma_{m+1}(t)\;) \\
\lambda_{m+1}(t) &= \lambda + (1-\lambda)\alpha^*/\alpha_{m+1}^b(t)
\end{aligned}
\tag{39}
$$

6. CONCLUSION

The presented Regularised Lattice–Ladder adaptive filter (summarized in tables (37),(38) and (39) has several remarkable features.

- The proposed regularised parameter tracking technique requires only a modest increase of additional computation at the cell level of the Lattice and Ladder sections without any additional data connections or loops in the data flow.

- The throughput of the Lattice - Ladder is equal to the standard algorithm with exponential weighting.

- The estimator is effectively regularized by the specification of the constant α^{b*}.

- The parameter tracking speed is effectively controlled by a constant forgetting factor λ. However the time update can be interpreted as the different time variable weighting factors $\lambda_m(t)$ along the diagonal.

- If $\alpha^{b*} = 0$ the algorithm reduces back to the standard exponential forgetting RLS Lattice - Ladder algorithm forcing all $\lambda_m(t) = \lambda$.

- If $\alpha^{b*}_m(t{+}1|t) = \alpha^b(t)$ then the standard exponential forgetting RLS Lattice - Ladder algorithm with a growing rectangular window is produced (by forcing all $\lambda_m(t) = 1$ even if $0 < \lambda < 1$).

- The algorithm is able to produce the exact initialization from the first measurement, identical with the RMGS implementation even for non zero initial conditions. It is reflected by the initial value of $\lambda_m(0) = 1$ even if $0 < \lambda < 1$. Later on, during the identification, the factors $\lambda_m(t)$ behave differently along the diagonal in dependence on the data $\varphi(t)$ transformation.

- Each backward errors variance $\alpha^b_m(\cdot)$ on diagonal appears to be tuned individually.

- In the case of the over–parametrization or the poor data $\varphi_n(t)$ the affected variance say $\alpha^b_m(\cdot)$ starts to converge to α^{b*} with the rate of convergence given by $\lambda_m(\cdot)$ which is in turn controlled λ and the "distance" of $\alpha_m^b(\cdot)$ from α^{b*}.

$$\lambda_m(t) \ = \lambda + (1-\lambda)\alpha^{b*}/\alpha_m^b(t) \tag{40}$$

- Because of propagation of the nesting the algorithm of order n consists and reproduces all the lower order, information filters of order $m = 1, 2, , ..., n-1$ and using the shifting properties effectively updates the set of the un-correlated backward a-posterior errors $e_m^b(t)$ serving as the transformed data set for the Ladder filter section.

- The Ladder sections propagates the a-posteriory (filtration) errors $\hat{e}_m(t)$ related to the nested model.

The paper relates the formulation of the the Ladder-Lattice approach with the square root information filtering and with the general probabilistic frameworks too.

References

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

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

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

[13] Kulhavý, R. Directional Tracking of Regression-Type Model Parameters. Preprints of the 2nd IFAC Workshop on Adaptive Systems in Con-

trol and Signal Processing. Lund. pp. 97 - 102, 1986.

[14] Ling, F., D. Manolakis, and J.G. Proakis . Numerically Robust Least-Squares Lattice-Ladder Algorithms with Direct Update of the Reflection Coefficients *IEEE Trans., ASSP-34*, 837-845, 1986.

[15] 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, 1707- 1712.

[16] Kulhavý, R.; M. B. Zarrop: On a general concept of forgetting. Submitted.

Acknowledgments

The first author gratefully acknowledges support by the research grants of Czech Academy of Sciences and Arts, The International Fund for Ireland and The Royal Society Czechoslovak Postdoctoral Fellowship Program.