# A MINIMAL, GIVENS ROTATION BASED FRLS LATTICE ALGORITHM

*François Desbouvries and Phillip A. Regalia*

Département Signal & Image
Institut National des Télécommunications
9 rue Charles Fourier
91011 Evry Cedex, France
desbou@int-evry.fr, regalia@int-evry.fr

## ABSTRACT

We propose a new Givens rotation based least-squares lattice algorithm. Based on spherical trigonometry principles, this algorithm turns out to be a normalized version of the fast QRD-based least-squares lattice filter, introduced independently by Ling and Proudler *et al*. In constrast to those algorithms, the storage requirements of the new algorithm are minimal (in the system theory sense). From this, we show that the new algorithm satisfies the backward consistency property, and hence enjoys stable error propagation.

## 1 INTRODUCTION

Yule's PARCOR identity was recently shown to coincide with the cosine law of spherical trigonometry. This observation establishes a connection between FRLS adaptive filtering and spherical trigonometry [1], [2], because the prewindowed fully-normalized FRLS lattice algorithm of Lee *et al*. [3] consists of three particular applications of Yule's PARCOR Identity. Let us briefly recall this algorithm for convenience of the reader. Let $v_{n,t}$ and $\eta_{n,t}$ be the doubly-normalized forward and backward prediction errors, respectively, at order $n$ and time $t$, and let $\rho_{n+1,t}$ be the $n+1^{st}$-order PARCOR at time $t$. The algorithm of Lee *et al*. consists of the well known recursions

$$\rho_{n+1,t} = v_{n,t}\,\eta_{n,t-1} + \sqrt{1-v_{n,t}^2}\,\rho_{n+1,t-1}\,\sqrt{1-\eta_{n,t-1}^2}$$
$$v_{n+1,t} = (1-\rho_{n+1,t}^2)^{-1/2}(v_{n,t}-\rho_{n+1,t}\,\eta_{n,t-1})(1-\eta_{n,t-1}^2)^{-1/2} \quad (1)$$
$$\eta_{n+1,t} = (1-\rho_{n+1,t}^2)^{-1/2}(\eta_{n,t-1}-\rho_{n+1,t}\,v_{n,t})(1-v_{n,t}^2)^{-1/2}$$

Recast in the spherical trigonometry framework [1], [2], the six PARCORs propagated by this algorithm are the cosines of the six elements of a spherical triangle, namely the angles $A$, $B$, and $C$, and their corresponding sides $a$, $b$, and $c$; the correspondences are $\cos A = \rho_{n+1,t-1}$, $\cos b = v_{n,t}$, $\cos c = \eta_{n,t-1}$, $\cos a = \rho_{n+1,t}$, $\cos B = v_{n+1,t}$ and $\cos C = \eta_{n+1,t}$. Thus (1) can be seen as one particular solution to the spherical triangle problem which, given two sides $b$ and $c$ of a spherical triangle, plus the angle $A$ in between, consists in determining the remaining three elements $a$, $B$ and $C$.

It turns out that this algorithm (using only the cosine law three times) is not, in spherical trigonometry, a standard solution of the spherical triangle problem $(A,b,c) \to (a,B,C)$.

Among other alternatives, one classical way for (partially) solving the problem consists in using a so-called "Gauss system" [4], [5], [6]. In this paper , we obtain a new algorithm based on two such Gauss systems, which yields, in effect, a normalized version of the QRD-based LS lattice filter, introduced independently by Ling [7] and Proudler *et al*. [8], [9]. The main advantage of our algorithm over that of [7]–[9] is that the storage requirements are cut in half, thereby purging the "redundancy" present in those algorithms. Indeed, the new algorithm now propagates a minimal (in the system theory sense) number of internal variables, all of magnitude bounded by 1. Now, minimality is known to be a key structural constraint behind backward consistency [10], and from this viewpoint we prove that this algorithm enjoys stable error propagation.

This paper is organized as follows. In Section 2 we give a geometric interpretation of the fast QRD-based lattice filter, in terms of spherical trigonometry. In this framework, it becomes obvious that some appropriate normalization should be performed, which leads to our new algorithm in Section 3. Stable error propagation is proved in Section 4, with concluding remarks in Section 5.

## 2 GAUSS SYSTEMS

Consider a unit sphere in 3D space centered at the point $O$. Let $C'$ be an arbitrary point, and consider a line beginning at $O$ and passing through $C'$, with $C$ its intersection with the unit sphere. The position of $C'$ can be described, for example, by the spherical coordinates $(R,\theta_1,\phi_1)$, where $R$ is the length of the vector $OC'$ and the two angles satisfy $\theta_1 \in [-\pi,\pi]$ and $\phi_1 \in [-\pi/2,\pi/2]$, or alternately by the Cartesian coordinates ($x_1 = R\cos\phi_1\cos\theta_1, y_1 = R\cos\phi_1\sin\theta_1, z_1 = R\sin\phi_1$) (see Fig. 1).

Let us rotate the initial coordinate system $(O,x_1,y_1,z_1)$ by an angle $c$ around the $y_1$ axis, giving a new coordinate system $(O,x_2,y_2,z_2)$, with $y_2 = y_1$. The coordinates of $C'$ in the new system are ($x_2 = R\cos\phi_2\cos\theta_2, y_2 = R\cos\phi_2\sin\theta_2, z_2 = R\sin\phi_2$). Let us now write the mapping $[x_1,y_1,z_1]^T \overset{c}{\to} [x_2,y_2,z_2]^T$ in terms of the elements of the spherical triangle $ABC$, using the relations $a = \pi/2-\phi_2$, $b = \pi/2-\phi_1$, $A = \pi-|\theta_1|$, and $B = |\theta_2|$. If we let $\varepsilon$ denote $\theta_1/|\theta_1| = \theta_2/|\theta_2|$, we obtain a
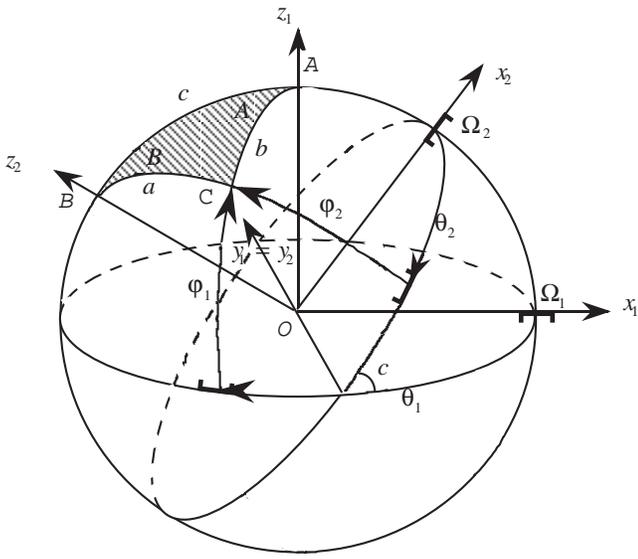
Figure 1: Illustrating the spherical coordinate systems.

"Gauss system" (unnormalized if $R \neq 1$),

$$\begin{bmatrix} \cos B \sin a \\ \varepsilon \sin B \sin a \\ \cos a \end{bmatrix} R = \begin{bmatrix} \cos c & 0 & \sin c \\ 0 & 1 & 0 \\ -\sin c & 0 & \cos c \end{bmatrix} \begin{bmatrix} -\cos A \sin b \\ \varepsilon \sin A \sin b \\ \cos b \end{bmatrix} R \quad (2)$$

whose three rows yield, respectively, the five-element formula, the sine law, and the cosine law of spherical trigonometry. These equations appear naturally in astronomy [4], [5], [6]. Indeed, passing back and forth between the horizontal coordinates of a star and its hour coordinates, or between its equatorial coordinates and its ecliptic coordinates, consists in performing such a change of spherical coordinates $(\theta_1, \phi_1) \overset{c}{\to} (\theta_2, \phi_2)$; this can be done using (2), since in these applications one is interested only in solving the partial problem $(A, b, c) \to (a, B)$.

These equations also arise in adaptive filtering and may be linked [2] via the classical geometric framework [11] of FRLS algorithms. Let $A$, $B$ and $C$ belong to the space $\mathbb{R}^{t+1}$ of $(t+1)$-dimensional vectors, with inner product $\langle u, v \rangle = u^T \Lambda v$, where $\Lambda = \text{diag}\{\lambda^t, \cdots, \lambda, 1\}$ with $\lambda$ the forgetting factor. Let $Y$ be a data matrix with $t+1$ rows, and let $P_Y^\perp = \mathbf{I} - Y(Y^T \Lambda Y)^{-1} Y^T \Lambda$ be the projection matrix onto the orthogonal complement subspace to the column space of $Y$. Let us identify $\overrightarrow{OA}$ with $\overline{P_Y^\perp A}$, $\overrightarrow{OB}$ with $\overline{P_Y^\perp B}$, $\overrightarrow{OC}$ with $\overline{P_Y^\perp C}$, and $\overrightarrow{OC'}$ with $P_Y^\perp C$, where $\overline{P_Y^\perp C} = P_Y^\perp C \langle P_Y^\perp C, P_Y^\perp C \rangle^{-1/2}$ designates scaling to unit length. Lastly, let the PARCOR $\rho_Y(u, v)$ be defined as $\langle \overline{P_Y u}, \overline{P_Y v} \rangle$. For any variable bounded in magnitude by 1, say $\rho$, we define its (positive) square complement $\rho^c$ according to $\rho^2 + (\rho^c)^2 = 1$. The PARCOR analogs to the first and third rows of (2) are [2]

$$\begin{bmatrix} \rho_{Y,B}(A,C)\rho_Y^c(B,C) \\ \rho_Y(B,C) \end{bmatrix} \langle P_Y^\perp C, P_Y^\perp C \rangle^{1/2} = \quad (3)$$

$$\begin{bmatrix} \rho_Y(A,B) & \rho_Y^c(A,B) \\ -\rho_Y^c(A,B) & \rho_Y(A,B) \end{bmatrix} \begin{bmatrix} -\rho_{Y,A}(B,C)\rho_Y^c(A,C) \\ \rho_Y(A,C) \end{bmatrix} \langle P_Y^\perp C, P_Y^\perp C \rangle^{1/2}$$

Let now $\{y_i\}$ and $\{d_i\}$ denote the input and desired-response time series, respectively, of the adaptive filter. Let $\mathbf{y}_i \overset{\text{def}}{=} [0 \cdots 0 \, y_0 \cdots y_i]^T$, $\mathbf{d}_i \overset{\text{def}}{=} [0 \cdots 0 \, d_0 \cdots d_i]^T$ and $\sigma \overset{\text{def}}{=} [0 \cdots 0 \, 1]^T$ belong to $\mathbb{R}^{t+1}$, and set $Y$ to the $(t+1) \times n$ matrix $[\mathbf{y}_{t-1} \cdots \mathbf{y}_{t-n}]$. The successive correspondences $(A = \sigma, B = \mathbf{y}_t, C = \mathbf{y}_{t-n-1})$, $(A = \sigma, B = \mathbf{y}_{t-n-1}, C = \mathbf{y}_t)$ and $(A = \sigma, B = \mathbf{y}_{t-n-1}, C = \mathbf{d}_{t-1})$ yield, respectively:

$$\begin{bmatrix} \nu_{n,t} & \nu_{n,t}^c \\ -\nu_{n,t}^c & \nu_{n,t} \end{bmatrix} \begin{bmatrix} -\sqrt{\lambda} \rho_{n+1,t-1} \beta_{n,t-2}^{1/2} \\ e_{n,t-1}^b \gamma_{n,t-1}^{-1/2} \end{bmatrix} = \begin{bmatrix} e_{n+1,t}^b \gamma_{n+1,t}^{-1/2} \\ \rho_{n+1,t} \beta_{n,t-1}^{1/2} \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \eta_{n,t-1} & \eta_{n,t-1}^c \\ -\eta_{n,t-1}^c & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\sqrt{\lambda} \rho_{n+1,t-1} \alpha_{n,t-1}^{1/2} \\ e_{n,t}^f \gamma_{n,t-1}^{-1/2} \end{bmatrix} = \begin{bmatrix} e_{n+1,t}^f \gamma_{n+1,t-1}^{-1/2} \\ \rho_{n+1,t} \alpha_{n,t}^{1/2} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} \eta_{n,t-1} & \eta_{n,t-1}^c \\ -\eta_{n,t-1}^c & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\sqrt{\lambda} \pi_{n,t-2} \\ e_{n,t-1} \gamma_{n,t-1}^{-1/2} \end{bmatrix} = \begin{bmatrix} e_{n+1,t-1} \gamma_{n+1,t-1}^{-1/2} \\ \pi_{n,t-1} \end{bmatrix} \quad (6)$$

Here $e_{n,t}^f$, $e_{n,t}^b$ and $e_{n,t}$ are, respectively, the unnormalized *a posteriori* forward prediction error, backward prediction error and filtering error, at order $n$ and time $t$. The term $\alpha_{n,t} \overset{\text{def}}{=} \langle P_Y^\perp \mathbf{y}_t, P_Y^\perp \mathbf{y}_t \rangle$ (resp. $\beta_{n,t-1} \overset{\text{def}}{=} \langle P_Y^\perp \mathbf{y}_{t-n-1}, P_Y^\perp \mathbf{y}_{t-n-1} \rangle$) is the forward (resp. backward) prediction error energy. The term $\gamma_{n,t-1} \overset{\text{def}}{=} \langle P_Y^\perp \sigma, P_Y^\perp \sigma \rangle$ is a "likelihood variable" and $\pi_{n,t-1} = \langle P_Y^\perp \mathbf{d}_{t-1}, P_Y^\perp \mathbf{y}_{t-n-1} \rangle$ is an auxiliary variable. We recognize in (4) and (5) the rotations of the prediction part, and in (6) that of the filtering part, of the fast QRD-based lattice [7], [8], [9], which is thus essentially made of three unnormalized Gauss systems.

## 3 A MINIMAL GIVENS ROTATION BASED LATTICE ALGORITHM

The algorithm above is nonminimal in its storage requirements, which impedes a direct verification of stable error propagation (e.g., [12]). In effect, the parameters $\nu_{n,t}$ and $\nu_{n,t}^c$ of the forward rotation are determined according to

$$\alpha_{n,t}^{1/2} = \left\| \begin{matrix} \lambda^{1/2} \alpha_{n,t-1}^{1/2} \\ e_{n,t}^f \gamma_{n,t-1}^{-1/2} \end{matrix} \right\|, \quad \nu_{n,t} = \frac{e_{n,t}^f \gamma_{n,t-1}^{-1/2}}{\alpha_{n,t}^{1/2}}, \quad \nu_{n,t}^c = \frac{\lambda^{1/2} \alpha_{n,t-1}^{1/2}}{\alpha_{n,t}^{1/2}}$$

where $\| \cdot \|$ denotes the Euclidean norm. Similarly, $\eta_{n,t-1}$ and $\eta_{n,t-1}^c$ are determined according to

$$\beta_{n,t-1}^{1/2} = \left\| \begin{matrix} \lambda^{1/2} \beta_{n,t-2}^{1/2} \\ e_{n,t-1}^b \gamma_{n,t-1}^{-1/2} \end{matrix} \right\|, \eta_{n,t-1} = \frac{e_{n,t-1}^b \gamma_{n,t-1}^{-1/2}}{\beta_{n,t-1}^{1/2}}, \eta_{n,t-1}^c = \frac{\lambda^{1/2} \beta_{n,t-2}^{1/2}}{\beta_{n,t-1}^{1/2}}$$

Thus this algorithm requires storing $5M + (M+1)$ variables at each iteration (where $M$ is the maximum order of the filter), namely the variables $\{\alpha_{n,t-1}^{1/2}\}_{n=0}^{M-1}$, $\{\beta_{n,t-2}^{1/2}\}_{n=0}^{M-1}$, $\{e_{n,t-1}^b \gamma_{n,t-1}^{-1/2}\}_{n=0}^{M-1}$, $\{\rho_{n+1,t-1} \alpha_{n,t-1}^{1/2}\}_{n=0}^{M-1}$ and $\{\rho_{n+1,t-1} \beta_{n,t-2}^{1/2}\}_{n=0}^{M-1}$ for the prediction section, as well as $\{\pi_{n,t-2}\}_{n=0}^{M}$ for the filtering part.

Now, in many applications, the quantity of ultimate interest is often the filtered error $e_{M+1,t}$, which is computed from $e_{M+1,t} \gamma_{M+1,t}^{-1/2}$ and $\gamma_{M+1,t}^{1/2}$, where $\gamma_{M+1,t}^{1/2}$ is given by

$$\gamma_{n+1,t}^{1/2} = \eta_{n,t}^c \, \gamma_{n,t}^{1/2}. \quad (7)$$

We now show that in such cases the storage requirements of the fast QRD-based least-squares lattice can be cut in half. In particular, the vectors to which the rotations (4) and (5) are applied are closely related to the parameters of these rotations, so that it is not in fact necessary to compute (as above) the parameters of these rotations. More precisely, (4) and (5) can be replaced by the corresponding *normalized* Gauss systems, i.e., the algorithm which transforms the coordinates of $\overline{P_Y^\perp C}$, rather than $P_Y^\perp C$:

$$\begin{bmatrix} v_{n,t} & v_{n,t}^c \\ -v_{n,t}^c & v_{n,t} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1}\eta_{n,t-1}^c \\ \eta_{n,t-1} \end{bmatrix} = \begin{bmatrix} \eta_{n+1,t}\rho_{n+1,t}^c \\ \rho_{n+1,t} \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} \eta_{n,t-1} & \eta_{n,t-1}^c \\ -\eta_{n,t-1}^c & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1}v_{n,t}^c \\ v_{n,t} \end{bmatrix} = \begin{bmatrix} v_{n+1,t}\rho_{n+1,t}^c \\ \rho_{n+1,t} \end{bmatrix} \quad (9)$$

The algorithm made of (8) and (9) for the prediction part, and keeping (6) and (7) for the filtering part, is summarized in Table 1; the storage requirements of the prediction part are reduced to $2M+1$ parameters (the minimal value [10]), namely the square-root signal energy $\alpha_{0,t}^{1/2}$, the doubly-normalized backward prediction error residuals $\{\eta_{n,t-1}\}_{n=0}^{M-1}$, and the PARCORS $\{\rho_{n+1,t-1}\}_{n=0}^{M-1}$. The filtering part still uses the $M+1$ parameters $\{\pi_{n,t-1}\}_{n=0}^{M}$.

As each prediction stage of our algorithm performs the mapping $(\rho_{n+1,t-1}, v_{n,t}, \eta_{n,t-1}) \rightarrow (\rho_{n+1,t}, v_{n+1,t}, \eta_{n+1,t})$, it is clearly related to (1), as well as to the algorithm of [13, eqs. (10) and (11)], which perform the same global transformation. This can be explained by spherical trigonometry principles: Since there are three degrees of freedom in a spherical triangle, there can be no more that three distinct relationships among the six elements. One such set is comprised of the three cosine laws (i.e., (1) in the FRLS framework), from which any other spherical trigonometry formula may be deduced.

## 4 STABLE ERROR PROPAGATION

The verification of stable error propagation follows the methodology of [10], in which stable error propagation is shown subordinate to a backward consistency property. The time updates in Table 1 may be written as a state recursion

$$\xi(t) = T[\xi(t-1), y_t] \quad (10)$$

in which

$$\xi(t) = \left[\alpha_{0,t}^{1/2}; \eta_{0,t}; \dots ; \eta_{M-1,t}; \rho_{1,t}; \dots ; \rho_{M,t}\right] \quad (11)$$

is the state vector comprising the variables that need be written for storage at each iteration, and $T[\cdot,\cdot]$ is the mapping which implements the subroutine of Table 1 at each iteration.

As in [10], [12], we denote by $\mathcal{S}_i$ the set of reachable states $\xi(t)$ in exact arithmetic, as the past input $y_t, y_{t-1}, y_{t-2}, \dots$, is varied over all possibilities. If the computed state vector, call it $\tilde{\xi}(t)$ (including thus roundoff errors), satisfies

$$\tilde{\xi}(t) \in \mathcal{S}_i, \quad (12)$$

then $\tilde{\xi}(t)$ may be understood as the exact solution reached by some perturbed past input sequence $\tilde{y}_t, \tilde{y}_{t-1}, \tilde{y}_{t-2}, \dots$. The

influence of the perturbation $\tilde{\xi}(t) - \xi(t)$ on the future evolution of the equation (10) is then identical to the influence of a corresponding perturbation on the past input sequence. With persistently exciting future data $y_{t+1}, y_{t+2}, \dots$, the influence of this perturbation will be exponentially decaying owing to exponential data weighting ($\lambda < 1$). For more detail on this simple argument, we refer the reader to [10], [12], [14], [15]. It thus suffices to examine whether the consistency condition (12) can be enforced in the present algorithm.

To this end, we can connect the variables of the state vector (11) to the variables of the QR algorithm studied in [10]. The PARCORS $\{\rho_{n,t}\}$ correspond, in the notations of [10], to $\rho_{n,t} = \sin\phi_n(t)$, while the doubly normalized backward prediction errors become $\eta_{n,t} = \sin\theta_n(t)$. From [10, Prop. 5], the set of reachable state variables is characterized by the simple inequalities $|\theta_{n-1}(t)| < \pi/2$ and $|\phi_n(t)| < \pi/2$ for orders $n = 0, 1, \dots, M$, combined with one prediction error energy being positive. This yields $\mathcal{S}_i$ in the present algorithm as

$$|\eta_{n-1,t}| < 1, \qquad |\rho_{n,t}| < 1, \quad \text{and} \quad \alpha_{0,t}^{1/2} > 0, \quad (13)$$

for all $n$ and $t$. Provided the computed variables always satisfy these inequalities, consistency applies, which then implies stable error propagation [10], [12], [14], [15].

To examine satisfaction of the consistency constraints (13), assume that the expression holds at time $t$, i.e., $|\eta_{n-1,t}| < 1$ and $|\rho_{n,t}| < 1$; we then examine whether it remains in effect at time $t+1$. The first computation reads as

$$\alpha_{0,t+1}^{1/2} = \left\| \begin{matrix} \lambda^{1/2}\alpha_{0,t}^{1/2} \\ y_{t+1} \end{matrix} \right\| > 0,$$

$$\eta_{0,t+1} = v_{0,t+1} = \frac{y_{t+1}}{\alpha_{0,t+1}^{1/2}} \quad \Rightarrow \quad |\eta_{0,t+1}| = |v_{0,t+1}| < 1.$$

(The intermediate variables $v_{n,t}$ intervene in the computation $\xi(t+1) = T[\xi(t), y_{t+1}]$, but are not written for storage). The computation (8) becomes

$$\begin{bmatrix} v_{n-1,t+1} & v_{n-1,t+1}^c \\ -v_{n-1,t+1}^c & v_{n-1,t+1} \end{bmatrix} \underbrace{\begin{bmatrix} -\rho_{n,t}\eta_{n-1,t}^c \\ \eta_{n-1,t} \end{bmatrix}}_{\overset{\text{def}}{=} \mathbf{a}} = \underbrace{\begin{bmatrix} \eta_{n,t+1}\rho_{n,t+1}^c \\ \rho_{n,t+1} \end{bmatrix}}_{\overset{\text{def}}{=} \mathbf{b}}$$

Now, with $|\rho_{n,t}| < 1$ and $|\eta_{n-1,t}| < 1$, we get $\|\mathbf{a}\| < 1$. If the norm of the numerical rotation on the left-hand side is less than $1/\|\mathbf{a}\|$, we obtain $\|\mathbf{b}\| < 1$. This holds, in particular, if the rotation is numerically passive. The property $\|\mathbf{b}\| < 1$ then ensures that the inferred variables $\eta_{n,t+1}$ and $\rho_{n,t+1}$ fulfill $|\eta_{n,t+1}| < 1$ and $|\rho_{n,t+1}| < 1$, so that these computed variables remain consistent. The same argument applied to (9) shows that $|v_{n,t+1}| < 1$ as well, so that the rotation required by the next order index is well defined. Thus if $\xi(t) \in \mathcal{S}_i$, then the updated state vector so computed likewise satisfies $\xi(t+1) \in \mathcal{S}_i$. This gives backward consistency, and thus stable error propagation.

## 5 CONCLUDING REMARKS

A new rotation-based lattice algorithm was derived. In contrast to the algorithms of Ling [7] and Proudler *et al.* [8],

[9], the present algorithm is minimal in the storage requirements. From this, stable error propagation is shown easily upon verifying that the computed variables remain backward consistent. The consistency test of the algorithm amounts to verifying that the square-root signal energy remains positive, and that all other variables are bounded by unit magnitude. This can be structurally induced by correct programming, or at least tested with negligible overhead if less careful programming methods are employed.

Our contribution extends the family of minimal fast least-squares algorithms beyond those identified in [10] and [14]; in contrast to the backward consistent algorithm of [10], for which the order recursions are both ascending and descending, all recursions in the present algorithm are ascending, which thereby affords pipelining of the computations.

## References

[1] F. Desbouvries, "Recursive least-squares lattices and trigonometry in the spherical triangle," *Proc. ICASSP*, pp. III.404–III.407, Minneapolis, MN, April 27–30, 1993.

[2] F. Desbouvries, "Spherical Trigonometry, Yule's PARCOR Identity and FRLS fully-normalized lattice," *IEEE Trans. Signal Processing*, vol. 43, pp. 427–430, February 1996.

[3] D. T. Lee, M. Morf and B. Friedlander, "Recursive Least Squares Ladder Estimation Algorithms," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 29, pp. 627–641, June 1981.

[4] V. Kourganoff, *Astronomie Fondamentale Élémentaire*, Masson, Paris, 1961.

[5] H. Andoyer, *Cours d'Astronomie*, 3$^{rd}$ ed., Librairie Scientifique J. Hermann, Paris, 1923.

[6] W. Chauvenet, *A Treatise on Plane and Spherical Trigonometry*, 9$^{th}$ ed., Lippincott, Philadelphia, 1875.

[7] F. Ling, "Givens rotation based least squares lattice and related algorithms," *IEEE Trans. Signal Processing*, vol. 39, pp. 1541–1551, 1991.

[8] I. K. Proudler, J. G. McWhirter and T. J. Shepherd, "Fast QRD-based algorithms for least squares linear prediction," in: *Mathematics in Signal Processing II*, J. G. McWhirter, ed., pp. 465–488, Clarendon Press, Oxford, 1990.

[9] I. K. Proudler, J. G. McWhirter and T. J. Shepherd, "Computationally efficient QR decomposition approach to least squares adaptive filtering," *IEE Proceedings-F*, vol. 138, pp. 341–353, August 1991.

[10] P. A. Regalia, "Numerical stability properties of a QR-based fast least-squares algorithm," *IEEE Trans. Signal Processing*, vol. 41, pp. 2096–2109, June 1993.

[11] H. Lev-Ari, T. Kailath and J. Cioffi, "Least-squares adaptive lattice and transversal filters: A unified geometric theory," *IEEE Trans. Information Theory*, vol. IT-30, pp. 222–236, March 1984.

[12] P. A. Regalia, "Numerical stability issues in fast least-squares adaptation algorithms," *Optical Engineering*, vol. 31, pp. 1144–1152, June 1992.

[13] H. M. Ahmed, J. M. Delosme, and M. Morf, "Highly concurrent computing structures for matrix arithmetic and signal processing," *Computer*, vol. 15, pp. 65–80, January 1982.

[14] D. T. M. Slock, "The backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms," *Optical Engineering*, vol. 31, pp. 1153–1169, June 1992.

[15] P. A. Regalia and F. Desbouvries, "Displacement structures of covariance matrices, lossless systems, and numerical algorithm design," *SIAM J. Matrix Analysis and Applications*, vol. 16, pp. 536–564, April 1995.

**Table 1**: Minimal Lattice Algorithm

**Initialization:**

$$\eta_{n,-1} = \pi_{n,-1} = \rho_{n+1,-1} = 0, \quad \text{for } n = 0, 1, \ldots, M-1;$$
$$\alpha_{0,-1}^{1/2} = 0 \text{ (exact start) or } \alpha_{0,-1}^{1/2} = \delta > 0 \text{ (soft start)}$$

$$\gamma_{0,t} = 1, \ e_{0,t} = d_t, \ \alpha_{0,t}^{1/2} = \left\| \begin{array}{c} \lambda^{1/2} \alpha_{0,t-1}^{1/2} \\ y_t \end{array} \right\|, \ v_{0,t} = \eta_{0,t} = \frac{y_t}{\alpha_{0,t}^{1/2}}$$

**for time** $t = 0, 1, 2, \ldots$:

$$\gamma_{1,t}^{1/2} = \eta_{0,t}^c$$

$$\begin{bmatrix} \eta_{0,t} & \eta_{0,t}^c \\ -\eta_{0,t}^c & \eta_{0,t} \end{bmatrix} \begin{bmatrix} -\lambda^{1/2} \pi_{0,t-1} \\ e_{0,t} \end{bmatrix} = \begin{bmatrix} e_{1,t} \gamma_{1,t}^{-1/2} \\ \pi_{0,t} \end{bmatrix}$$

**for order** $n = 0, \ldots, M-1$:

prediction:

$$\begin{bmatrix} \eta_{n,t-1} & \eta_{n,t-1}^c \\ -\eta_{n,t-1}^c & \eta_{n,t-1} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1} v_{n,t}^c \\ v_{n,t} \end{bmatrix} = \begin{bmatrix} v_{n+1,t} \rho_{n+1,t}^c \\ \rho_{n+1,t} \end{bmatrix}$$

$$\begin{bmatrix} v_{n,t} & v_{n,t}^c \\ -v_{n,t}^c & v_{n,t} \end{bmatrix} \begin{bmatrix} -\rho_{n+1,t-1} \eta_{n,t-1}^c \\ \eta_{n,t-1} \end{bmatrix} = \begin{bmatrix} \eta_{n+1,t} \rho_{n+1,t}^c \\ * \end{bmatrix}$$

($* =$ superfluous computation)

$$v_{n+1,t} = \left( v_{n+1,t} \rho_{n+1,t}^c \right) \Big/ \rho_{n+1,t}^c$$

$$\eta_{n+1,t} = \left( \eta_{n+1,t} \rho_{n+1,t}^c \right) \Big/ \rho_{n+1,t}^c$$

filtering:

$$\gamma_{n+2,t}^{1/2} = \eta_{n+1,t}^c \gamma_{n+1,t}^{1/2}$$

$$\begin{bmatrix} \eta_{n+1,t} & \eta_{n+1,t}^c \\ -\eta_{n+1,t}^c & \eta_{n+1,t} \end{bmatrix} \begin{bmatrix} -\lambda^{1/2} \pi_{n+1,t-1} \\ e_{n+1,t} \gamma_{n+1,t}^{-1/2} \end{bmatrix} = \begin{bmatrix} e_{n+2,t} \gamma_{n+2,t}^{-1/2} \\ \pi_{n+1,t} \end{bmatrix}$$

**end for order** loop

$$e_{M+1,t} = \left( e_{M+1,t} \gamma_{M+1,t}^{-1/2} \right) \gamma_{M+1,t}^{1/2}$$

**end for time** loop.