The Backward Consistency Concept and Roundoff Error Propagation Dynamics in RLS Algorithms

Dirk T.M. Slock

Institut EURECOM
2229 route des Crêtes
Sophia Antipolis
06560 Valbonne
France

Abstract

A new approach for the analysis of the propagation of roundoff errors in recursive algorithms is presented. This approach is based on the concept of *backward consistency.* In general, this concept leads to a decomposition of the state space of the algorithm, and in fact, to a *manifold.* This manifold is the set of state values that are backward consistent. Perturbations within the manifold can be interpreted as resulting from perturbations on the input data. Hence, the error propagation on the manifold corresponds exactly (without averaging or even linearization) to the propagation of the effect of a perturbation of the input data at some point in time on the state of the algorithm at future times.

We apply these ideas to the Kalman filter and its various derivatives. In particular, we consider the conventional Kalman filter, some minor variations of it, and its square-root forms. Next we consider the Chandrasekhar equations, which apply to time-invariant state-space models. Recursive Least-Squares (RLS) parameter estimation is a special case of Kalman filtering, and hence the previous results also apply to the RLS algorithms. We shall furthermore consider in detail two groups of fast RLS algorithms: the Fast Transversal Filter (FTF) algorithms and the Fast Lattice/Fast QR (FLA/FQR) RLS algorithms.

Keywords:

recursive least-squares, Kalman filtering, Chandrasekhar equations, fast algorithms, roundoff errors, backward consistency

1 Introduction

In numerical analysis, the method of backward error analysis is a well-established procedure [1]. In this method, the computed solution to a problem (which differs from the exact solution due to roundoff errors in the computations) is interpreted as the exact solution of a perturbed problem. A method for solving the problem is called (backward) stable in this context if the perturbed problem is close to the original problem.

However, if the problem has a specific structure (consider e.g. the solution of a system of equations in which the matrix of coefficients has a specific structure), then a difficulty may arise if one requires the perturbed problem in the backwards association scheme to belong to the same class of structured problems as the original problem. Indeed, the method of solution may lead to a computed solution that cannot be interpreted as the exact solution to a perturbed problem belonging to the given class of structured problems. We shall call an algorithm for solving the given structured problem *backward consistent* if it always leads to a computed solution that can be interpreted as the solution of a perturbed problem with the required structure [2],[3],[4]. Once backward consistency is established, one can check whether the algorithm is (backward) stable. If it is, then the algorithm is called strongly stable in a recently introduced nomenclature by Bunch [5]. However, we shall concentrate on the consistency part and not on the stability part. Though the consistency issue (it does not appear to have a specific name in the numerical analysis literature) has been the subject of quite a bit of discussion in numerical analysis circles, its application (which we propose here) to recursive algorithms such as the Kalman filter appears to be even more enlightening.

Reference [6] represents perhaps the most up-to-date study of round-off errors in various Kalman filtering algorithms. However, there are several aspects in that work that one may consider as treated unsatisfactorily. We focus here specifically on the propagation of the numerical errors: we assume that from a certain time instant onwards no more round-off errors are made and observe how the effect of the accumulated errors evolves in time from then onwards. One approximation that is used in [6] to carry out the analysis of the error propagation is linearization: only the error terms of first order are kept. Furthermore, some (statistical) averaging step is introduced, effectively narrowing the analysis to only the mean of the errors (in a statistical description). One outcome from the analysis in [6] is that the so-called square-root forms of the Kalman filter would have little if any advantage over the conventional form in that the level of the accumulated errors in both algorithms would be comparable. This may be so, after linearization, if the covariance matrix is well bounded away from singularity. This conclusion is a bit unsatisfactory, considering the availability of a wealth of practical evidence that indicates the superior numerical behavior of the square-root algorithms in actual implementations (see e.g. [7]). The backward consistency point of view appears to shed quite a bit of light on this issue.

A special fast type of Kalman filter algorithm is given by the so-called Chandrasekhar equations [8]. The analysis of this algorithm in [6] is quite approximate.

When we limit the state estimation problem to the problem of estimating constant parameters, the Kalman filter reduces to the RLS algorithm. The Chandrasekhar equations reduce to the Fast Transversal Filter algorithm, a fast RLS algorithm (applicable to the case of a Toeplitz data matrix). The propagation of numerical errors in the FTF algorithm has been analyzed in [9] using linearization and averaging techniques and some other approximations. It was shown in [9] how in the original algorithm numerical errors diverge, and how the introduction of computational redundancies and a certain feedback mechanism allow for the stabilization of the error propagation dynamics. Though this analysis has revealed the key mechanisms in the error propagation, it does not tell the complete story (such as the influence of a nonstationary input signal).

Finally, we shall consider the second class of fast RLS algorithms, namely the FLA/FQR algorithms. Especially for the FQR algorithms, the backward consistency approach leads to a very simple but complete analysis of the error propagation dynamics. This last application, which was first considered in [10], provided a breakthrough in a problem that appears very hard to tackle using the previously existing techniques.

The notation we shall use below will vary according to established practices or specific references for the different algorithms considered, except for the time index, which will be denoted as k, and the system order index which will be denoted as n (intermediate orders) or N (largest order).

2 Backward Consistency

2.1 The Classical Approach

A recursive algorithm can be viewed as a nonlinear discrete-time system of the form

$$\Theta(k) \;=\; f\left(\Theta(k-1), \zeta(k)\right) \tag{1}$$

where $\Theta(k)$ (written as a row vector) is the set of state variables of the algorithm, and $\zeta(k)$ collects the new input data that are used at time k. In a finite-precision implementation, the actual system propagates a perturbed state $\widehat{\Theta}(k)$ and can be written as

$$\widehat{\Theta}(k) \;=\; f\left(\widehat{\Theta}(k-1), \zeta(k)\right) + V(k) \tag{2}$$

where $V(k)$ is the round-off error introduced in the computations at time k. From (1) and (2), one can deduce the nonlinear system for the accumulated errors on the state variables, $\Delta\Theta(k) = \widehat{\Theta}(k) - \Theta(k)$:

$$\begin{aligned} \Theta(k) \;&=\; f\left(\Theta(k-1), \zeta(k)\right) && (3) \\ \Delta\Theta(k) \;&=\; f\left(\Theta(k-1) + \Delta\Theta(k-1)\,,\; \zeta(k)\right) - \Theta(k) + V(k)\;\; . && (4) \end{aligned}$$

Assuming the numerical errors to be small, we can linearize the implemented system (2) around the infinite-precision trajectory (3), which leads us to replace (4) by (neglecting higher-order terms):

$$\Delta\Theta(k) \;=\; \Delta\Theta(k-1)F(k) + V(k) \tag{5}$$

where $F(k) = \left.\frac{\partial f}{\partial \Theta}\right|_{\Theta=\Theta(k-1)}$. One often adds to this a second step, in which the linear time-varying system of (5) is transformed into a linear time-invariant system by applying an averaging technique. One such technique goes as follows [11]. If the data $\zeta(.)$ are considered as stochastic variables then the process $\{\Delta\Theta(.)\}$ converges weakly, provided certain conditions are satisfied, to the process $\Xi(.)$ which satisfies

$$\Xi(k) \;=\; \Xi(k-1)\;\; \mathrm{E}\, F(k) + V(k) \tag{6}$$

when $F(.)$ converges to a series of identity matrices. So the error propagation dynamics can be described via the eigenvalues of the fixed matrix $\mathrm{E}\, F(k)$ (in the stationary case). A complete study of the roundoff errors requires besides the study of the error propagation (the initial-state response of the $\Delta\Theta(.)$ system) also the study of the error generation (properties of $V(.)$) and the error accumulation (how do errors generated at different time instants interact to form the compounded error $\Delta\Theta(.)$) [12].

2.2 Relation to Concepts from Numerical Analysis

We may comment on the relation between the notion of (forward) stability as used by numerical analysts, and the stability of the roundoff error system viewed as a discrete-time dynamical system above. When considering forward stability, numerical analysts refer to the size of the perturbation of the computed quantities (as a result of roundoff errors in the computations). In the approach outlined above, stability of the roundoff errors is about the stability of the dynamical system (4) that describes the propagation of the roundoff errors. At this point, the two stability notions appear to be unrelated. However, a proper quantity to study in the recursive algorithm approach above, but from a numerical analysis point of view, would be the (infinity) norm of the complete roundoff error trajectory $\Delta\Theta(.)$. Then it is clear that this norm will be finite (stable from a numerical analysis point of view) if and only if (with weak assumptions) the dynamical system (4) is Bounded-Input-Bounded-Output (BIBO) stable. The instantaneously generated roundoff errors $V(.)$ entering the dynamical system (4) can be modeled as deterministic or stochastic quantities (numerical analysts stick almost exclusively to deterministic descriptions though). In a deterministic treatment, the term bounded in the BIBO notion refers to actually bounded roundoff errors. In a stochastic treatment, it is common practice (in the analysis of roundoff errors in digital filters [13]) to assume that the $V(.)$ form a sequence of independent random variables (this is one of the weak assumptions mentioned above). In a stochastic context, bounded refers to bounded variance. In either of the deterministic or stochastic contexts, the algorithm implementation is assumed to lead to bounded input $V(.)$. Note that this deterministic/stochastic consideration for $V(.)$ is independent from a possible deterministic/stochastic point of view for the estimation problem treated recursively by the algorithm (1) (the averaging approximation mentioned above assumes a stochastic description of the quantities appearing in the estimation problem). Returning to the stability issue, we conclude that the algorithm implementation becomes more stable in a numerical analysis sense as the bound on the input $V(.)$ becomes smaller, and as the modes of the dynamical system (4) decay faster.

2.3 Error Propagation in Backward Consistent Algorithms

Here, we propose an alternative approach to the classical linearization and averaging strategy outline above. In general, the problem that the recursive algorithm is solving belongs to a certain class of problems with a well-defined structure. Consider the set of values for the state variables which can be interpreted as being the solution to a problem with the required structure. In general, the structure requirement will lead to equality as well as inequality constraints. The equality constraints can be written as $\mathcal{M}(\Theta) = 0$, i.e. the state should lie on a certain manifold $\mathcal{M}$ (see Fig. 1). The inequality constraints will lead in fact to the consideration of only a subset $\mathcal{S}$ of the manifold. In the Kalman filtering context, the inequality constraints result from the positive definiteness of the covariance matrix of the vector of estimated quantities, or related properties. One can in principle consider an extension of the set $\mathcal{S}$ from $\mathcal{M}$ to the complete state space. However, only the part of $\mathcal{S}$ within $\mathcal{M}$ matters. So the algorithm state is said to result from a problem with the required structure if it satisfies

$$\Theta \in \mathcal{M} \cap \mathcal{S} \ . \tag{7}$$

We shall in general only consider the interior of $\mathcal{S}$, since the roundoff errors may behave differently for degenerate (singular) cases (situated on the boundary of $\mathcal{S}$).

In order to study the local dynamics of the propagation of roundoff errors, we assume that roundoff errors have been made up to time k and that the computations are exact from then onwards. A perturbation of the state $\Theta(k)$ that remains on the manifold can be interpreted as the exact solution to a problem with perturbed data $\{\Theta(-1), \zeta(0), \ldots, \zeta(k)\}$. In fact, under some conditions, this perturbation can be reduced to a perturbation on $\Theta(-1)$ only. Or if not, it suffices to consider $\Theta(k)$ as the (perturbed) initial condition of an algorithm that operates free of error after time k. So the propagation of roundoff errors that lead to a perturbation of the state within the manifold can be analyzed through the dynamics of a perturbation of the initial conditions on the state trajectory in an infinite-precision implementation of the algorithm. Thus for a backward consistent algorithm implementation, the study of the dynamics of the (numerical) error system (4) reduces to the study of the convergence from arbitrary initial conditions of the (estimation) error system associated with the algorithm (3) itself.

This local analysis holds for the linearized system, for which superposition can be used to describe the accumulated errors $\Delta\Theta(.)$. The results thus obtained will hold if the accumulated errors are sufficiently small so that higher-order terms may be neglected. If general, the roundoff errors will perturb the dynamics with which the accumulated errors up to a certain time instant will decay into the future. One way to obtain global results is the following. If the decay from perturbed initial conditions can be bounded above uniformly (in the allowed perturbations on the data $\Theta(-1)$, $\zeta(.)$) by a decaying function that can be considered as the impulse response of a BIBO system, then the error system (4) is globally stable.

2.4 Example: the LMS Algorithm

The celebrated Least-Mean-Square (LMS) algorithm [14] can be considered as a simplification of the Kalman filter for the parameter (impulse response of an FIR filter) estimation problem. The algorithm updates the filter estimate $W_{N,k}$ according to (see [15] for notation)

$$\left\{ \begin{array}{rcl} \epsilon^p_N(k) & = & d(k) + W_{N,k-1}X_N(k) \\ W_{N,k} & = & W_{N,k-1} - \mu\, \epsilon^p_N(k) X^H_N(k)\ . \end{array} \right. \tag{8}$$

The state of the algorithm is clearly $\Theta(k) = W_{N,k}$ and the data are $\zeta(k) = \{d(k), X_N(k)\}$. In general, there are no constraints on $W_{N,K}$ for it to be a valid filter estimate. Hence $\mathcal{M} \cap \mathcal{S}$ is the complete state space here. So the roundoff error system associated with the LMS algorithm is stable if and only if the algorithm converges! The convergence in the presence of instantaneous roundoff errors $V(.)$ on $W_{N,.}$ can be interpreted as the dual problem of tracking optimal parameters $W^o_{N,k}$ that move around according to a random walk: $W^o_{N,K} = W^o_{N,k-1} + V(k)$ [14],[16] (leading to a perturbation of the desired-response signal $d(.)$). The presence of such variations of the optimal parameters does not modify the convergence properties of the LMS algorithm (the convergence dynamics are uniform in perturbations on $d(.)$, hence the error system is globally stable when the algorithm converges). But such perturbations make the algorithm converge to a steady state with increased variance for the parameter estimates $W_{N,k}$ and the error signal $\epsilon^p_N(k)$ (BIBO stability in a stochastic sense). See [14],[16] for expressions for these variance increases, given the variance of $V(.)$ (assumed stationary).

2.5 Decomposition of the Error System in the General Case

An algorithm is backward consistent if the computed state remains on the manifold. In a finite-precision environment, this will usually require that the manifold $\mathcal{M}$ is the complete state space. In general, the manifold is nontrivial and the study of the propagation of roundoff error will be more complicated than in the backward consistent case described above. However, the manifold allows us to introduce a simplification in the form of two decoupled subsystems as we shall explain now.

Whenever the manifold $\mathcal{M}$ is nontrivial, there exists a parameterization Θ_m of $\Theta \in \mathcal{M}$ with fewer parameters than Θ. This parameterization Θ_m is minimal if it is not subject to any equality constraints and contains enough degrees of freedom to represent any point on $\mathcal{M}$. So there exists a function $h_{\mathcal{M}}$ such that

$$\forall\Theta \in \mathcal{M} \cap \mathcal{S}\,,\ \exists\Theta_m\ :\ \Theta = h_{\mathcal{M}}(\Theta_m)\ \ . \tag{9}$$

Thus in order to study the error propagation on $\mathcal{M}$, we may well consider the error propagation for the minimal parameterization Θ_m. We apply linearization to find the local error dynamics. The component of the error $\Delta\Theta(k)$ along $\mathcal{M}$ (manifold $\approx$ tangent subspace) is given by:

$$\Delta_M\Theta(k) = \Delta\Theta(k)\,P_{M(k)} = \Delta\Theta_m(k)\,M^H(k) \tag{10}$$

where $M(k) = \frac{\partial h_{\mathcal{M}}^{-1}}{\partial\Theta}\Big|_{\Theta=\Theta(k)}$ and P_X denotes the projection matrix onto the column space of X. So the (linearized) error propagation on $\mathcal{M}$ is governed by the system matrix

$$P_{M(k-1)}F(k)P_{M(k)}\ \ . \tag{11}$$

Note that the product of consecutive instances of the matrix in (11) are a pre and postmultiplication away from a corresponding product of the matrix

$$M^H(k{-}1)F(k)M^+(k) \tag{12}$$

where M^+ is the Hermitian transpose of the left inverse of M. The linearized error propagation dynamics on $\mathcal{M}$ are described by (11) or (12) and correspond to the linearized dynamics of the effect of a perturbation of the initial conditions of the estimation problem. Global error propagation within $\mathcal{M}$ also corresponds with the global behavior of perturbations on the data $\Theta(-1)$, $\zeta(.)$.

Now let us write the equality constraints specifying the manifold compactly as $\mathcal{M}(\Theta) = 0$. Then the normal subspace of $\mathcal{M}$ locally at $\Theta(k)$ can be characterized as

$$\Delta\Theta(k)N(k) = 0\ ,\quad \text{where } N^H(k) = \frac{\partial\mathcal{M}}{\partial\Theta}\bigg|_{\Theta=\Theta(k)}\ \ . \tag{13}$$

Using the normalized version of $M(k)$ and $N(k)$ (e.g. $\overline{M} = M(M^HM)^{-H/2}$), we can form a unitary matrix $\Phi = \left[\overline{N}\ \ \overline{M}\right]$, $\Phi\Phi^H = I$. Introducing Φ into the error propagation equation $\Delta\Theta(k) = \Delta\Theta(k{-}1)F(k)$, we obtain

$$\begin{aligned}\Big[\Delta\Theta(k)\overline{N}(k) \quad \Delta\Theta(k)\overline{M}(k)\Big] &= \Big[\Delta\Theta(k{-}1)\overline{N}(k{-}1) \quad \Delta\Theta(k{-}1)\overline{M}(k{-}1)\Big] \\ &\times \begin{bmatrix} \overline{N}^H(k{-}1)F(k)\overline{N}(k) & \overline{N}^H(k{-}1)F(k)\overline{M}(k) \\ 0 & \overline{M}^H(k{-}1)F(k)\overline{M}(k) \end{bmatrix}\end{aligned} \tag{14}$$

which describes the joint reparameterized system of tangential and normal error components

$$\begin{aligned} \Delta_M\Theta(k) &= \Delta\Theta(k)P_{M(k)} &= \Delta\Theta(k)\overline{M}(k)\overline{M}^H(k) \\ \Delta_N\Theta(k) &= \Delta\Theta(k)P_{N(k)} &= \Delta\Theta(k)\overline{N}(k)\overline{N}^H(k) \end{aligned} \tag{15}$$

see Fig. 2. The main thing to note is that we can put a priori $\overline{M}^H(k{-}1)F(k)\overline{N}(k) = 0$ for the simple reason that a $\Theta(k)$ on $\mathcal{M}$ remains on $\mathcal{M}$! (reminder: the study of the local error propagation assumes that no further errors are introduced after an initial perturbation). So the decomposition of the error along tangential and normal components leads to two subsystems that are coupled only in a one-way sense (and hence can be investigated separately). The algorithm is backward consistent on the manifold and hence the results from the backward consistent case apply to the tangential subsystem. The error propagation of the normal subsystem remains to be investigated then (e.g. using the conventional linearization approach described above). Global stability can be obtained if the manifold can be shown to be a global attractor and a uniform rate of convergence towards $\mathcal{M}$ can be exhibited. We shall illustrate an application of the above decomposition in the next section, but we shall not carry out a systematic investigation of the normal subsystem in the applications considered below, as this would lead us too far.

It may happen that an algorithm is not backward consistent at a certain level of structure, but that it is backward consistent at a lower (less restrictive) level of structure. This provides a means of investigating (part of) the normal subsystem using the same techniques as for the tangential subsystem. We shall illustrate this below. To summarize, the concept of backward consistency may be introduced to allow for a decomposition of the error propagation problem into two smaller problems. For the analysis of one of these, the backward consistent subsystem, a different stochastic setting can be exploited, namely the one of the estimation problem itself, which is different from a possible stochastic characterization of the roundoff errors.

3 Conventional Kalman Filters

Consider the discrete-time linear state-space model for $k \geq 0$

$$\begin{array}{rcl} x_{k+1} & = & F_k x_k + G_k w_k \\ y_k & = & H_k x_k + v_k \end{array} \quad \text{with E} \begin{bmatrix} w_k \\ v_k \\ x_0 \end{bmatrix} \begin{bmatrix} w_l \\ v_l \\ x_0 \\ 1 \end{bmatrix}' = \begin{bmatrix} Q_k\delta_{kl} & 0 & 0 & 0 \\ 0 & R_k\delta_{kl} & 0 & 0 \\ 0 & 0 & \Pi_0 & 0 \end{bmatrix} . \quad (16)$$

The case E $w_k v_l' = C_{kl}\delta_{kl}$ can be handled as well. The state estimate $\widehat{x}_k = \widehat{x}_{k|k-1}$ is updated according to

$$\begin{array}{rcl} \epsilon_k & = & y_k - H_k \widehat{x}_k \\ \widehat{x}_{k+1} & = & F_k \widehat{x}_k + K_k R_k^{-e} \epsilon_k \\ \multicolumn{3}{l}{\text{with } \widehat{x}_0 = 0, \;\; K_k = F_k P_k H_k', \;\; P_k = P_{k|k-1} = \text{ E } \tilde{x}_k \tilde{x}_k'} \end{array} \quad (17)$$

and the covariance matrix P_k is updated according to the Riccati equation which will be discussed in more detail below.

3.1 Global Stability of Backward Consistent Implementations

The state $\Theta(k)$ of the conventional Kalman filter (CKF) consists of the pair $\widehat{x}_k$, P_k (we are talking about two state-space models now: one in x_k for the estimation problem and one in $\Theta(k)$ for the estimation algorithm). The input to the algorithm is $\zeta(k) = \{F_k, G_k, H_k, Q_k, R_k, y_k\}$. Note that $\widehat{x}_.$ depends on $P_.$ but not vice versa. Hence, the error propagation in both subsystems may be analyzed separately. The consistency requirements are $P_k = P_k' \geq 0$. There are no constraints on the $\widehat{x}_k$ part of $\Theta(k)$. Hence this part is always backward consistent and numerical errors on $\widehat{x}_k$ propagate in a stable fashion whenever the Kalman filter converges (sufficiently fast), as in the LMS algorithm. The state estimate update in (17) can be rewritten as

$$\widehat{x}_{k+1} \;=\; \Phi_k \widehat{x}_k + K_k R_k^{-e} y_k \,, \quad \Phi_k \;=\; F_k - K_k R_k^{-e} H_k \tag{18}$$

which renders explicit the crucial role of the closed-loop system matrix Φ_k of which the long term behavior is described by the state transition matrix

$$\Phi_{k,l} = \Phi_{k-1}\Phi_{k-2}\cdots\Phi_l \ . \tag{19}$$

In particular, the effect on $\widehat{x}_k$ of a perturbation on the initial condition is given by

$$\Delta\widehat{x}_k \;=\; \Phi_{k,0}\Delta\widehat{x}_0 \ . \tag{20}$$

The closed-loop system matrix also plays a crucial role in the error propagation for the Riccati equation as we shall see below.

In a typical numerical analysis approach [6], one takes a worst case point of view and the dynamics of (18) are summarized by the scalar quantity

$$\gamma_k \;=\; \|\Phi_k\|_2 \ . \tag{21}$$

If we consider for instance a parameter estimation problem with $F_k = I$, $G_k Q_k G_k^{'} = 0$, then $\gamma_k \geq 1$, $\forall k$ and $\gamma_k > 1$ in general. Because from this point of view, one can never arrive at a stable error propagation system, γ_k is replaced in [6] by the spectral radius of the averaged Φ_k using arguments that are unusual for rigorous numerical analysis. In [17], this shortcoming has been rectified for the special case of parameter estimation, by realizing that the long-term dynamics are indeed governed by $\Phi_{k,l}$ (as in [12]), which is the backward consistency simplification. In the particular case of parameter estimation, there exists a simple explicit expression for $\Phi_{k,l}$ [12].

Consider now the P_k part of $\Theta(k)$. An implementation of the Kalman filter that leaves the computed $P_.$ backward consistent can be considered as resulting from infinite-precision calculations of a perturbed state-space problem with the following perturbed part of the input: $\Delta G_k Q_k G_k^{'}$ (note that this perturbation may lead to a product $G_k Q_k G_k^{'}$

of increased rank if the original $G_k Q_k G_k'$ was not of full rank). For a local stability analysis, we consider such perturbations up to a time instant k and no further perturbations thereafter. These perturbations lead to a perturbed P_k which may be considered as the initial condition of a new state space estimation problem which is the same as the original one, but starts at time k. The robustness (stability) of the Kalman filter w.r.t. such a perturbation of the initial condition is a problem of convergence of the Kalman filter, and has been analyzed extensively in [18]. Usually, one requires Q-controllablity and R-observability to guarantee exponential convergence. In [18] however, very loose conditions have been obtained which guarantee decay of the effect of perturbations of the initial condition on the innovations variance $R_k^e = \mathrm{E}\ \epsilon_k \epsilon_k'$. Furthermore, we have the following theorem from [18].

Theorem 1 *Assume that the state space model (16) is uniformly completely observable and that the model parameters are uniformly bounded. Consider a perturbation $\Delta\Pi_0$ such that $\Pi_0 + \Delta\Pi_0 > 0$. The effect of such a perturbation decays, viz.*

$$\Delta P_k \longrightarrow 0 \text{ as } k \rightarrow 0 \text{ at a rate at least } \frac{1}{k^\alpha} \text{ for some } \alpha > 1.$$

Note that a rate of decay as in the above theorem is sufficient to have BIBO stability in both the deterministic and the stochastic sense. Furthermore, this rate is uniform in perturbations that leave the perturbed algorithm backward consistent ($P_k + \Delta P_k > 0,\ \forall\, k \geq 0$). Hence the algorithm is globally stable w.r.t. such perturbations. One may remark that if exponential forgetting is introduced in the Kalman filter, then the decay is exponential at a rate at least equal to the rate of forgetting (see [19]). However, exponential decay is not necessary to have the globally stable behavior of the error system, as we just indicated.

Whereas the local dynamics of the error decay on the state estimate are governed by Φ_k, the local dynamics for errors on P_k are governed by "the square" of Φ_k (see further). Hence, the weak conditions mentioned above lead to the rate of decay in Theorem 1 for the square of Φ_k. The rate of decay for Φ_k itself under the weak conditions above may not be sufficient to lead to a BIBO system, especially in a deterministic approach (and one may remark that the treatment of the bias due to non-zero mean roundoff errors in a stochastic approach parallels the deterministic treatment). Hence, stronger requirements on the estimation problem (than the ones in Theorem 1) need to be satisfied for the error propagation on $\widehat{x}_k$ to represent a BIBO stable system.

Now, the CKF in its straightforward implementation is not backwards consistent. We shall consider the two components of the consistency requirements in some detail.

3.2 Symmetry

The symmetry requirement has received quite a bit of attention lately [6],[20],[17]. Note that P_k can be decomposed into its symmetric part $P_k^s = \frac{1}{2}\left(P_k + P_k^{'}\right)$ and its antisymmetric part $P_k^a = \frac{1}{2}\left(P_k - P_k^{'}\right)$. Obviously, $P_k^s \in \mathcal{M} = \{P_k : P_k^a = 0\}$, and the above error propagation analysis applies to it provided that $P_k^s \in \mathcal{S} = \{P_k : P_k > 0\}$, $\forall\, k$. The complementary part of the manifold is parameterized by P_k^a. In [6],[20], it has been indicated (but not shown properly) that certain implementations of the CKF are stable, while in other implementations, P_k^a blows up exponentially if the open-loop state transition matrix is unstable. Consider the following two implementations of the Riccati equation:

$$\begin{aligned} R_k^e &= R_k + H_k P_k H_k^{'} \\ P_{k+1} &= F_k \left[I - P_k H_k^{'} R_k^{-e} H_k \right] P_k F_k^{'} + G_k Q_k G_k^{'} \end{aligned} \tag{22}$$

and

$$\begin{aligned} R_k^e &= R_k + H_k P_k H_k^{'} \\ K_k^p &= P_k H_k^{'} \\ P_{k+1} &= F_k \left[P_k - \left(K_k^p R_k^{-e} \right) K_k^{p'} \right] F_k^{'} + G_k Q_k G_k^{'} \end{aligned} \tag{23}$$

Either implementation shows that an instantaneous error on P_{k+1} can be interpreted as a perturbation on $G_k Q_k G_k^{'}$. The second implementation is computationally more efficient. Round-off errors will cause $P_k^a \neq 0$ in either implementation. When we perform a local analysis of the propagation of the tangential (ΔP_k^s) and normal (ΔP_k^a) error components in (22), we obtain

$$\begin{aligned} \Delta P^s_{k+1} &= \Phi_k \, \Delta P^s_k \, \Phi'_k \\ \Delta P^a_{k+1} &= \Phi_k \, \Delta P^a_k \, \Phi'_k \ . \end{aligned} \tag{24}$$

Notice the "triangular" coupling structure between tangential and normal error components (we have in fact a "diagonal" structure: complete decoupling). The equation for ΔP^s_k shows how its local dynamics are governed by the "square" of Φ_k. We note also that the normal subsystem is governed by the same (stable) dynamics as the tangential (backward consistent) subsystem. This is no surprise. Even though the Riccati equation in (22) is not backward consistent within the class of symmetric Riccati equations, it is consistent within the larger class of non-symmetric Riccati equations, and hence the robustness of non-symmetric Riccati equations applies. Using (22), P^a_k has been found (also experimentally) not to blow up in [20]. Non-symmetric Riccati equations can perhaps most easily be understood by specializing to the parameter estimation problem. In this problem, the symmetric Riccati equation corresponds to the RLS algorithm, while the non-symmetric Riccati equation corresponds to the Recursive Instrumental Variables (RIV) method [21],[22]. Now the robustness w.r.t. perturbations of the initial conditions is pretty much the same for the RIV algorithm as for the RLS algorithm (as in fact shown by (24)), which explains the stability of (22).

If $P^a_k \neq 0$ however, then (23) on the other hand is not a Riccati equation, not even within the class of non-symmetric ones. Performing a local analysis on (23), we obtain

$$\begin{aligned} \Delta P^s_{k+1} &= \Phi_k \, \Delta P^s_k \, \Phi'_k + F_k \left(\Delta P^a_k H'_k R^{-e}_k K^{p'}_k - K^p_k R^{-e}_k H_k \, \Delta P^a_k \right) F'_k \\ \Delta P^a_{k+1} &= F_k \, \Delta P^a_k \, F'_k \ . \end{aligned} \tag{25}$$

One may remark again the triangular structure of the coupling between the two subsystems. The backward consistent subsystem ΔP^s_k inherits the convergence properties of the Kalman filter, as discussed before. The complementary subsystem has different dynamics. An unstable open loop system (F_k) leads to exponential instability of the normal subsystem, as has been verified in [20]. An exponentially stable open loop on the other hand guarantees global stability of the implementation (23).

One may remark that $P_k \in \mathcal{M}$ is trivially enforced by e.g. computing only the upper triangle of P_{k+1} and filling in the lower triangle by symmetry, or by performing the operation $P_k \leftarrow \frac{1}{2}\left(P_k+P_k'\right)$ after each update.

3.3 Positive Definiteness

If P_k becomes indefinite, then the innovations variance R_k^e may become zero (or singular, if multivariable) or arbitrarily close to zero (singular). This singularity may cause very large (unbounded) perturbations in the Riccati equation. Also, if P_k is indefinite, then there exists a subspace in which the usual negative feedback of the Kalman filter is turned into positive feedback, easily leading to an unstable closed-loop state transition matrix. The danger of P_k becoming indefinite in the CKF depends on the combination of the precision of the arithmetic and the conditioning of the estimation problem.

These issues are illustrated in Fig. 3 for a scalar Riccati equation in the case of parameter estimation (hence RLS) (see also [23], and [24] for the multivariable case). Exponential weighting is used with weighting factor $\lambda = 0.99$. The Riccati equation reduces to $P_{k+1} = P_k/(P_k+\lambda)$. The innovations variance becomes singular for $P = -\lambda$. If $P_0 > 0$, then P_k will converge monotonuously to its steady-state value $P_\infty = 1-\lambda = 0.01$. When $P_0 = 0$, then $P_k \equiv 0$, so P_k gets trapped if it ever were to become zero. With $P_0 < 0$, we see that the system is unstable and hence the slightest negative initial value gets blown up, until at time m $P_m < -\lambda$. Then automatically $P_{m+1} > 0$, from where P_k converges smoothly. So, starting from a negative initial condition, the system has to go through a bursting phenomenon before it stabilizes. Note that as $P_{m-1} - (-\lambda)$ goes down to zero, P_m goes negative without bound. Or as $-\lambda - P_m$ goes down to zero, P_{m+1} goes positive without bound.

In [25], it was shown that the nonlinearities in the Riccati equation lead to a negative $\Delta G_k Q_k G_k'$. The authors proposed to compensate for this effect by adding an explicit $\delta G_k Q_k G_k' > 0$ (at each iteration). However, it is difficult to determine in general how much should be added in order to guarantee $\widehat{P}_k > 0, \ \forall\, k,$, while avoiding to introduce too much suboptimality. In [26], an interesting error analysis has been carried out (for the special case of the RLS algorithm), avoiding any linearization, to clearly bring out the effects of nonlinearities on the accumulation of the instantaneous roundoff errors (averaging and other approximations had to be introduced though in order to interpret the results). They lead to the same conclusion of a negative bias in ΔP_k. The remedy they propose though is, instead of adding some $\delta G_k Q_k G_k' > 0$, to organize all rounding operations so that they bias $\widehat{P}_k$ towards positivity. Again, this mechanism is not able to guarantee positivity under all circumstances. Finally, in [27] the rounding errors are explicitly taken into account in the estimation problem, and the optimal estimate is taken to minimize the total estimation variance, including the contribution from roundoff noise.

3.4 Square-Root Algorithms

Square-Root Kalman filters (SRKF) either propagate a matrix square-root of P_k (covariance form) or its inverse (information form) [8]. It is immediately clear that SRKF algorithms are backward consistent and hence the analysis for backward consistent algorithms in subsection 3.1 applies. This quality should make them the preferred forms for implementation. It is trivial to ensure that $\widehat{P}_k$ remains in the strict interior of $\mathcal{S}$, by keeping the diagonal of a triangular square-root of $\widehat{P}_k$ non-zero (using e.g. rounding-up instead of truncation).

The reason that the analysis in [6] did not bring out this positive character of the SRKF algorithms is that when linearization is done, it is done around the infinite-precision trajectory. So we always linearize around a $P_k > 0$, no matter how close it may be to singularity (with the computed $\widehat{P}_k$ possibly being indefinite). In this way, the desastrous effects of a $\widehat{P}_k$ becoming indefinite can simply never show up.

We should note that square-root algorithms do not propagate the state estimate directly. Rather, they propagate the quantity $P_k^{-1/2}\widehat{x}_k$, from which $\widehat{x}_k$ can be determined if desired. The error propagation for $P_k^{-1/2}\widehat{x}_k$ is also determined by the closed-loop state transition matrix $\Phi_{k,l}$.

3.5 Hierarchies of Consistency

We saw above one example of hierarchical levels of consistency in terms of symmetric and general Riccati equations. As another example of hierarchy of structure and consistency requirements, consider the case when the system matrices are time-invariant. Then the Chandrasekhar equations apply and will often lead to a significant reduction in the parameterization of the problem, if a certain displacement rank is low enough (see below). So one could define backwards consistency in this case as the requirement to lead to a problem with a certain given displacement rank. Applying the Chandrasekhar equations will ensure that this displacement rank is maintained throughout the computations. However, one may also apply a Kalman filter to such a problem. But the Kalman filter will not be backward consistent within the class of problems with a given displacement rank! Nevertheless, the Kalman filter is backward consistent (at least, the square-root algorithms are) for a broader class of least-squares problems.

4 The Chandrasekhar Equations

So consider the Kalman filtering problem for the case in which the system matrices are time-invariant. Let

$$\text{rank}\ (\delta P_0)\ =\ \text{rank}\ (P_1 - P_0)\ =\ \alpha\ . \tag{26}$$

Due to the time-invariance, we have rank $(\delta P_k)\ =\ \text{rank}\ (P_{k+1} - P_k)\ =\ \alpha\ ,\ \forall\, k > 0$.

4.1 A Normalized Algorithm

We can consider a minimal factorization

$$\delta P_0 = -\overline{L}_0 \Sigma \overline{L}_0^{'} \tag{27}$$

where Σ is a $\alpha \times \alpha$ signature matrix with the same non-zero inertia as δP_0. The normalized Chandrasekhar equations provide a recursive computation of the quantities $R_k^{e/2}$ and

$\overline{K}_k = K_k R_k^{-e'/2}$ (the state estimate update (17) requires the product $K_k R_k^{-e} = \overline{K}_k R_k^{-e/2}$) as follows [28]

$$\begin{bmatrix} R_{k+1}^{e/2} & 0 \\ \overline{K}_{k+1} & \overline{L}_{k+1} \end{bmatrix} = \begin{bmatrix} R_k^{e/2} & H\overline{L}_k \\ \overline{K}_k & F\overline{L}_k \end{bmatrix} \overline{\Psi}_k \ , \quad \overline{\Psi}_k J \overline{\Psi}_k' = J \ , \quad J = \begin{bmatrix} I_p & 0 \\ 0 & -\Sigma \end{bmatrix} \tag{28}$$

where I_p is an identity matrix of the size of y_k, and $\overline{\Psi}_k$ is a so-called J-orthogonal matrix which is chosen so as to annihilate the (1,2) block entry as indicated.

The state $\Theta(k)$ of the algorithm consists of the quantities $\widehat{x}_k$ as in the Kalman filter (and the error propagation properties of this part are unmodified), but P_k is replaced by $\overline{K}_k$, $R_k^{e/2}$, and $\overline{L}_k$. We seek the manifold $\mathcal{M}$ for this algorithm. Though P_k is not explicitly present in this algorithm, we could propagate it according to the following two update equations :

$$\begin{aligned} P_{k+1} &= P_k - \overline{L}_k \overline{L}_k' \\ P_{k+1} &= F P_k F' - \overline{K}_k \overline{K}_k' + GQG' \end{aligned} \tag{29}$$

Eliminating P_{k+1} between these two equations, we find that given the state of the Chandrasekhar equations, P_k can be computed from the Lyapunov equation

$$P_k - F P_k F' = -\overline{K}_k \overline{K}_k' + \overline{L}_k \overline{L}_k' + GQG' \ . \tag{30}$$

Conditions for solvability of this equation are $\lambda_i \lambda_j \neq 1, \ \forall i, j$, where the λ_i are the eigenvalues of F. For instance, if all $|\lambda_i| < 1$, then we can write the solution explicitly as

$$P_k = \sum_{i=0}^{\infty} F^i \left(-\overline{K}_k \overline{K}_k' + \overline{L}_k \overline{L}_k' + GQG' \right) F^{i'} \ . \tag{31}$$

Next, the requirement $P_k > 0$ imposes certain inequality constraints on the state $\Theta(k)$ (leading to the subset $\mathcal{S}$ of $\mathcal{M}$). The Chandrasekhar manifold $\mathcal{M}$ can now be seen to be determined by

$$\begin{aligned} K_k \ &\triangleq \ \overline{K}_k R_k^{e'/2} &&= F P_k H' \\ R_k^e \ &\triangleq \ R_k^{e/2} R_k^{e'/2} &&= R + H P_k H' \\ & &&= R + H F^{-1} K_k \quad \text{if } F^{-1} \text{ exists,} \end{aligned} \tag{32}$$

where P_k is the solution of (30) or (31). From (32), K_k and R_k^e are clearly a function of $\overline{K}_k$ and $R_k^{e/2}$. The reverse is also true if a unique (e.g. triangular or symmetric) matrix square-root $R_k^{e/2}$ is chosen (if not, then extra non-minimality is introduced into the algorithm with the ensuing possibly undesirable dynamics of a complementary subsystem). We assume this matrix square-root henceforth to be unique (different choices for this matrix square-root correspond to different forms of $\overline{\Psi}_k$). The manifold may be parameterized by $\overline{L}_k$ and $\widehat{x}_k$ combined, i.e., given $\overline{L}_k$, $\overline{K}_k$ and $R_k^{e/2}$ can be computed from (30),(32). Within this manifold, the Chandrasekhar equations are backward consistent and hence the error propagation corrsponds to the robustness of a perturbation of the initial conditions P_0 (such that still rank $\delta P_0 = \alpha$) and $\widehat{x}_0$, as in the conventional Kalman filter! As for the complementary part of the manifold (with the dimension of $\overline{K}_k$ and $R_k^{e/2}$ combined), consider

$$\begin{aligned} \widetilde{H}' &= P_k^{-1} F^{-1} K_k \\ \widetilde{R} &= R_k^e - \widetilde{H} P_k \widetilde{H}' \end{aligned} \qquad (33)$$

assuming F is invertible. Then the state $\Theta(k)$ of the Chandrasekhar equations at time k corresponds to the data Π_0, GQG', $\widetilde{H}$ and $\widetilde{R}$. However, after time k, the data H and R will be used. So at time k, accumulated round-off errors that cause the Chandrasekhar state Θ to deviate from the Chandrasekhar manifold correspond to a variation in the system matrices at time k. Such a (isolated) time-variation still can be accommodated by the Chandrasekhar equations, but leads to an increase of α [8],[29]. Hence, the error propagation in the complementary part is determined by the effect of such an increase in α when it is actually being ignored, or in other words, when the displacement δP_l, $l \geq k$ of rank $> \alpha$ is being approximated by $-\overline{L}_l \overline{L}_l'$ of lower rank α. We shall not further investigate these issue here.

4.2 An Unnormalized Algorithm

In an unnormalized algorithm, the Kalman gain K_k itself is propagated and a specific form of Ψ_k is chosen in which a maximal number of entries are either zero or one, viz. [8]

$$\left[\begin{array}{cc} R^e_{k+1} & 0 \\ K_{k+1} & L_{k+1} \\ 0 & R^r_{k+1} \end{array}\right] = \left[\begin{array}{cc} R^e_k & HL_k \\ K_k & FL_k \\ L^{'}_k H^{'} & R^r_k \end{array}\right] \Psi_k \;\; , \quad \Psi_k = \left[\begin{array}{cc} I & -R_k^{-e} H L_k \\ -R_k^{-r} L^{'}_k H^{'} & I \end{array}\right] \tag{34}$$

where Ψ_k is now an unnormalized J-orthogonal matrix, viz.

$$\Psi_k J_{k+1} \Psi^{'}_k = J_k \;\; , \quad J_k^{-1} = \left[\begin{array}{cc} R^e_k & 0 \\ 0 & R^r_k \end{array}\right] \; . \tag{35}$$

The state $\Theta(k)$ of the unnormalized algorithm is the same as that of the normalized algorithm, except that $\overline{K}_k$, $R_k^{e/2}$ are replaced by K_k, R^e_k, and $\overline{L}_k$ by L_k, R^r_k. It is easy to see that we have an overparameterization considering that

$$\delta P_k \;=\; -L_k R_k^{-r} L^{'}_k \tag{36}$$

(see (30) also, where $\overline{L}_k \overline{L}^{'}_k$ gets replaced by $L_k R_k^{-r} L^{'}_k$). So the quantities L_k, R^r_k influence the problem only through the product $L_k R_k^{-r} L^{'}_k$. The size of the augmentation of the complementary part of the state is $\frac{1}{2}\alpha\,(\alpha+1)$ (assuming R^r_k is computed in a way which respects its symmetry). Without going into the details of the error propagation in this complementary part, it is desirable to eliminate the $\frac{1}{2}\alpha\,(\alpha+1)$ extra degrees of freedom. One way of doing this is obviously by using the square-root Chandrasekhar equations discussed above. This is also done in the normalized FTF algorithm (a special instance of the Chandrasekhar equations, see below). In the unnormalized FTF algorithm, R^r_k is kept diagonal, while each column of L_k has an entry fixed to the value 1 (as in the LDU triangular decomposition), eliminating in this way the extra degrees of freedom.

In the unnormalized algorithm described above, a number of extra constraints appear in fact. First of all, the signature of R^r_k should remain constant in time (inequality constraints). Also, one can rewrite the updates of R^e_k and R^r_k in (34) above as (with

$R^r_k = R^{r/2}_k \Sigma R^{r'/2}_k$)

$$\left\{ \begin{array}{lcl} R^{e/2}_{k+1} & = & R^{e/2}_k \left(I - Y_k Y_k'\right)^{1/2} \\ R^{r/2}_{k+1} & = & R^{r/2}_k \left(I - Y_k' Y_k\right)^{1/2} \end{array} \right. , \quad Y_k = R^{-e/2}_k H L_k R^{-r'/2}_k \; . \tag{37}$$

Hence, the $\min(p, \alpha)$ smallest eigenvalues (which are all < 1) of $R^r_{k+1} R^{-r}_k$ and $R^e_{k+1} R^{-e}_k$ are equal. From this, it is difficult to obtain information on for instance the eigenvalues of R^r_k, R^e_k themselves. However, by considering the product of the eigenvalues, we get

$$\frac{\det R^r_{k+1}}{\det R^e_{k+1}} = \frac{\det R^r_k}{\det R^e_k} = \cdots = \frac{\det R^r_0}{\det R^e_0} \tag{38}$$

or hence an equality constraint on R^r_k in which Π_0 enters also in general.

So we do not offer a complete treatment of the error propagation in the Chandrasekhar equations here, since we are not investigating the complementary part. Perhaps the only other treatment of numerical errors in the Chandrasekhar equations available in the literature though appears in [6]. There, the state for the error system is taken to be the errors on the state estimate and on the Kalman gain. The quantities $\overline{L}_k$ and R^e_k have been left out. Thus, only an undecoupled subsystem of the error system is considered in [6] and hence the resulting analysis may not be valid, not even for the part of the system that has been considered.

5 Fast RLS Algorithms for FIR Filter Estimation

We have mentioned the problem of estimating the impulse response W_N of an FIR filter of length N before, when we discussed the LMS algorithm. Here we consider RLS algorithms, applied to the same problem. This problem is a special case of the Kalman filtering problem (16) with for instance $G_k \equiv 0$, $F_K \equiv I$, and consecutive H_k ($X_N(k)$ in the notation of LMS/RLS) being related by a shift structure. We shall introduce exponential weighting though and thus the LS criterion becomes (see [15] for notation)

$$\min_{W_{N,k}} \left\{ \sum_{i=0}^{k} \lambda^{k-i} \left\| d_i - W_{N,k} X_N(i) \right\|^2 \; + \; \lambda^{k+1} \mu \left\| W_{N,k} - W_0 \right\|^2 \right\} \tag{39}$$

where $\lambda \overset{<}{\approx} 1$ and the regression vector is $X_N(k) = \left[x^H(k)\; x^H(k-1) \cdots x(k-N+1)\right]^H$ and $x(.)$ is the input signal to the adapted FIR filter (for simplicity, we shall assume the signals to be scalars). The RLS algorithm for criterion (39) updates the filter estimate according to [30],[22]

$$\left\{ \begin{array}{rcl} \epsilon^p_N(k) & = & d(k) + W_{N,k-1} X_N(k) \\ W_{N,k} & = & W_{N,k-1} + \epsilon^p_N(k) \gamma_N(k) \widetilde{C}_{N,k} \end{array} \right. \tag{40}$$

(these equations are called the *joint-process* part in RLS terminology) where

$$\begin{array}{rcl} \widetilde{C}_{N,k} & = & -X^H_N(k) \lambda^{-1} R^{-1}_{N,k-1} \\ \gamma^{-1}_N(k) & = & 1 - \widetilde{C}_{N,k} X_N(k) \end{array} \tag{41}$$

and the inverse of the sample covariance matrix $R_{N,k} = \sum\limits_{i=0}^{k} \lambda^{k-i} X_N(i) X^H_N(i) + \lambda^{k+1} \mu I$ is recursively updated using the following special instance of the Riccati equation

$$R^{-1}_{N,k} \;=\; \lambda^{-1} R^{-1}_{N,k-1} - \widetilde{C}^H_{N,k} \gamma_N(k) \widetilde{C}_{N,k} \tag{42}$$

The initial conditions are obviously $W_{N,-1} = W_0$, $R^{-1}_{N,-1} = \frac{1}{\mu} I$. The state of the RLS algorithm consists of $W_{N,k}$ and $R^{-1}_{N,k}$, the analogs of $\hat{x}_{k+1}$ and P_{k+1} in the Kalman filter. So we see that the joint-process part (state component $W_{N,k}$) is backward consistent, while for $R^{-1}_{N,k}$, we have the Kalman filter manifold. The closed-loop system matrix $\Phi_{k,l}$, the crucial quantity determining the propagation of backward consistent errors, can for RLS be written explicitly as

$$\Phi_{k,l} \;=\; \lambda^{k-l} R^{-1}_{N,k} R_{N,l} \;\; , \quad k \geq l \;\; . \tag{43}$$

This shows the exponential decay with base λ for errors on the filter estimate when $R_{N,k}$ remains bounded from above and from below. The situation for the Riccati equation is slightly modified from the Kalman filter case considered above, due to the introduction of exponential weighting, and we get (for backward consistent errors)

$$\Delta R^{-1}_{N,k} \;=\; \frac{1}{\lambda^{k-l}} \Phi_{k,l} \Delta R^{-1}_{N,l} \Phi^H_{k,l} \;\; . \tag{44}$$

Equations (43) and (44) combined lead again to an exponential decay with base λ. Note that for the case $\lambda = 1$, $R_{N,k}^{-1}$ decays as $\frac{1}{k}$. From (43) and (44), we get that errors on $R_{N,k}^{-1}$ decay as $\frac{1}{k^2}$, which leads to a BIBO error system. The errors on $W_{N,K}$ decay as $\frac{1}{k}$ only however.

When we apply the RLS algorithm to minimize the cost function (39) recursively, we ignore the special structure that is present in the problem at hand (shift structure between $X_N(k)$ and $X_N(k-1)$). Ignoring this structure leads to the same manifold as we have found for the Kalman filter. Taking this structure into account leads to extra constraints that we shall discuss in detail below. So in fact, the RLS algorithm is not backward consistent for the specially structured problem that we are actually considering. However, it is backward consistent (at least, the square-root versions are) for a more general least-squares problem in which the consecutive $X_N(k)$ would be unrelated, and this has allowed us to analyze the error propagation using the backward consistency approach. Whether we include the special structure in the backward consistency requirements or not, we have found that the error propagation of the backward consistent part of any RLS algorithm is exponential with base λ (when $\lambda < 1$). This proves the conjecture in [9] that *the maximum eigenvalue governing the error propagation in any RLS algorithm for solving (39) can not be smaller than* λ.

5.1 Minimal Parameterizations of the Structured LS Problem

We shall consider the so-called prewindowed problem, in which the input signal $x(.)$ is assumed to be zero before time 0. Fast RLS algorithms exploit the special shift structure in the LS problem considered here, and replace the Riccati equation with what is generally called the *prediction* part. The prediction part delivers the Kalman gain $\widetilde{C}_{N,k}$ and the likelihood variable $\gamma_N(k)$ (in the FTF algorithm, or equivalent quantities in the FLA/FQR algorithms) to the joint-process part. So the coupling between the prediction part and the joint-process part is one-way, and hence both parts can be studied separately (as the $\hat{x}_k$ and P_k parts in the Kalman filter). The study of the joint-process part in the fast RLS algorithms is the same as for the conventional RLS algorithms, so we shall not further discuss the joint-process parts. Traditionally, the prediction part is overdimensioned by one, replacing the Riccati equation for the inverse of $R_{N+1,k}$ instead of $R_{N,k}$. We shall indicate how this slight nonminimality can be removed in each of the FTF, FLA and FQR algorithms. This modification does not influence the fundamental nature of the error propagation in the prediction part however.

The shift relation between consecutive $X_N(k)$ leads to $R_{N+1,k}$ being *close-to-Toeplitz*. Therefore, we consider the *displacement* structure of a Hermitian matrix R [31]:

$$\nabla_{\zeta} R \;=\; R - \zeta\, S R S^H \;=\; G \Sigma G^H \tag{45}$$

where S is the lower shift matrix (ones on the first subdiagonal). The displacement $\nabla_{\zeta} R$ has a minimal factorization of the form $G\Sigma G^H$ with Σ being a signature matrix with dimension equal to the rank of $\nabla_{\zeta} R$. Let $r_{N+1}(k) = R_{N+1,k} u_1$, the first column, and $r^0_{N+1}(k) = u_1^H R_{N+1,k} u_1$. Then

$$\begin{aligned} \nabla_{\frac{1}{\lambda}} R_{N+1,k} &= r_{N+1}(k) r^{-0}_{N+1}(k) r^H_{N+1}(k) \\ &\qquad - S S^H r_{N+1}(k) r^{-0}_{N+1}(k) r^H_{N+1}(k) S S^H - \tfrac{1}{\lambda} \begin{bmatrix} 0 \\ X_N(k) \end{bmatrix} \begin{bmatrix} 0 \\ X_N(k) \end{bmatrix}^H \qquad (46) \\ R_{N+1,k} &= \mathcal{T}\left(r_{N+1}(k)\right) - \tfrac{1}{\lambda} \mathcal{L}\left(\begin{bmatrix} 0 \\ X_N(k) \end{bmatrix} \right) \mathcal{L}\left(\begin{bmatrix} 0 \\ X_N(k) \end{bmatrix} \right)^H \end{aligned}$$

where $\mathcal{T}(v)$ and $\mathcal{L}(v)$ are a Hermitian and lower triangular Toeplitz matrix resp. with v as first column. So we have the following minimal parameterization of $R_{N+1,k}$:

$$\Theta_m(k) \;=\; \left[r^H_{N+1}(k) \;\; X^H_N(k) \right] \, . \tag{47}$$

Hence, the dimension of the minimal parameterization is $2N{+}1$, or actually $2N{-}1$, after the substitution $N{+}1 \rightarrow N$. In order to study the error propagation in backward consistent fast RLS algorithms, we may well consider the error propagation for the minimal parameterization $\Theta_m(k)$ for which the state equations are

$$\left\{ \begin{array}{rcl} r^H_{N+1}(k) & = & \lambda\, r^H_{N+1}(k-1) + x(k) X^H_N(k-1)\,[0 \;\; I_N] + |x(k)|^2\, u^H_1 \\ X_N(k) & = & S X_N(k-1) + u_1 x(k)\; . \end{array} \right. \tag{48}$$

$x(k)$ is the input signal for this quadratic system. Linearization (manifold $\approx$ tangent subspace) yields the local error dynamics:

$$\Delta\Theta_m(k) \;=\; \Delta\Theta_m(k-1) \begin{bmatrix} \lambda I_{N+1} & 0 \\ [0 \;\; x(k) I_N] & S^H \end{bmatrix} \tag{49}$$

which are exponentially stable with base λ. However, the linearization step can actually be dropped. Indeed, due to the obvious global exponential stability of the error system corresponding to (48), the error propagation of backward consistent fast RLS algorithms is globally asymptotically stable (within the applicable $\mathcal{S}$).

6 The FTF Algorithm

The Fast Transversal Filter (FTF) algorithm [15],[9], is a fast version of the conventional RLS algorithm above. The state for the FTF algorithm is (see [9] for an introduction and notation)

$$\Theta(k) = \left[\, a_{N,k} \;\; \alpha_N(k) \;\; b_{N,k} \;\; \beta_N(k) \;\; \widetilde{C}_{N,k} \;\; \gamma_N^{-1}(k) \,\right] \tag{50}$$

where $A_{N,k} = [1 \;\; a_{N,k}]$, $B_{N,k} = [b_{N,k} \;\; 1]$ are forward and backward prediction filters. The classical approach based on linearization and averaging leads to the following system matrix of the error system

$$\mathrm{E}\; F(k) = \begin{bmatrix} \lambda I & * & * & * & 0 & * \\ 0 & \lambda & * & * & 0 & * \\ 0 & 0 & \left[\frac{1}{\lambda} - K_1\,(1-\lambda)\right] I & * & 0 & * \\ 0 & 0 & * & \frac{1}{\lambda} - 2K_2\,(1-\lambda) & 0 & * \\ 0 & 0 & 0 & 0 & S^H - u_N b_N & 0 \\ * & * & * & * & 0 & 1 - K_3 \end{bmatrix} \tag{51}$$

for $\lambda \in [1-\epsilon, 1]$. The K_i are feedback coefficients pertaining to a stabilizing feedback mechanism based on redundancy, and $S^H - u_N b_N$ is a bottom companion matrix with the zeros of the backward prediction polynomial as eigenvalues. The eigenvalues of F are approximately given by its diagonal entries.

The prediction part of the FTF algorithm provides $\widetilde{C}_{n,k}$, $\gamma_n(k)$ for $n = N, N+1$. Hence, it is straightforward to reduce N by one, $N+1 \rightarrow N$, in the prediction part and still provide the appropriate quantities $\widetilde{C}_{N,k}$, $\gamma_N(k)$ for the joint-process part. But we shall stick to $N+1$ here for notational simplicity.

6.1 The FTF Manifold $\mathcal{M}_{\frac{2}{3}}$

Consider now the following question. If $\Theta(k)$ corresponds to a prewindowed problem with the given shift-invariance structure, can $\Theta(k)$ be arbitrary? The answer is no, $\Theta(k)$ lies on the following manifold $\mathcal{M}_{\frac{2}{3}}$.

6.1.1 Necessity of $\overline{\mathcal{M}}_{\frac{2}{3}}$

Consider

$$\nabla_\lambda \begin{bmatrix} R_{N,k}^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \overline{A}_{N,k}^H \overline{A}_{N,k} - \overline{B}_{N,k}^H \overline{B}_{N,k} + \lambda \begin{bmatrix} 0 & \overline{C}_{N,k} \end{bmatrix}^H \begin{bmatrix} 0 & \overline{C}_{N,k} \end{bmatrix} \tag{52}$$

$$\Longleftrightarrow \begin{bmatrix} R_{N,k}^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \sum_{i=0}^{N} \lambda^i S^i \, G \Sigma G^H \, S^{Hi} \ , \quad \Sigma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} , \; G^H = \begin{bmatrix} \overline{A}_{N,k} \\ \overline{B}_{N,k} \\ \sqrt{\lambda} \begin{bmatrix} 0 & \overline{C}_{N,k} \end{bmatrix} \end{bmatrix} \tag{53}$$

The normalized filters $\overline{A}_{N,k}$,$\overline{B}_{N,k}$,$\overline{C}_{N,k}$ constitute the state for the (prediction part of) the normalized FTF algorithm [15]. Due to the zero entries (problem independent values) in the matrix considered above, its generator G satisfies certain constraints which can be expressed as follows:

$$\left(\sum_{i=0}^{N} \lambda^i S^i G \Sigma G^H S^{Hi} \right) u_{N+1} = 0 \ , \ \ u_{N+1}^H = [0 \cdots 0 \ 1] \ . \tag{54}$$

These constraints define the FTF manifold $\overline{\mathcal{M}}_{\frac{2}{3}}$. A compact characterization of $\overline{\mathcal{M}}_{\frac{2}{3}}$ is obtained as follows. Pre and postmultiply (52) with $\left[1 \; \frac{z}{\sqrt{\lambda}} \cdots \left(\frac{z}{\sqrt{\lambda}}\right)^N\right]$, $\left[1 \; \frac{w}{\sqrt{\lambda}} \cdots \left(\frac{w}{\sqrt{\lambda}}\right)^N\right]^H$ to obtain

$$\begin{aligned}(1-zw)\,\mathcal{R}_{N,k}^{-1}(z,w) \;&=\; \overline{\mathcal{A}}_{N,k}\left(\tfrac{\sqrt{\lambda}}{z}\right)\overline{\mathcal{A}}_{N,k}\left(\tfrac{\sqrt{\lambda}}{w}\right)\\ &-\left(\tfrac{zw}{\lambda}\right)^N\overline{\mathcal{B}}_{N,k}\left(\tfrac{z}{\sqrt{\lambda}}\right)\overline{\mathcal{B}}_{N,k}\left(\tfrac{w}{\sqrt{\lambda}}\right)+zw\lambda\overline{\mathcal{C}}_{N,k}\left(\tfrac{\sqrt{\lambda}}{z}\right)\overline{\mathcal{C}}_{N,k}\left(\tfrac{\sqrt{\lambda}}{w}\right)\end{aligned} \tag{55}$$

Putting $w = z^{-1}$ yields

$$0=\overline{\mathcal{A}}_{N,k}\left(\frac{\sqrt{\lambda}}{z}\right)\overline{\mathcal{A}}_{N,k}\left(\sqrt{\lambda}z\right)-\lambda^{-N}\overline{\mathcal{B}}_{N,k}\left(\frac{z}{\sqrt{\lambda}}\right)\overline{\mathcal{B}}_{N,k}\left(\frac{1}{\sqrt{\lambda}z}\right)+\lambda\overline{\mathcal{C}}_{N,k}\left(\frac{\sqrt{\lambda}}{z}\right)\overline{\mathcal{C}}_{N,k}\left(\sqrt{\lambda}z\right) \tag{56}$$

Now, the specification of the manifold via (56) is not yet complete. Indeed, the condition expressed in (54) forces $\sum_{i=0}^{N}\lambda^i S^i G\Sigma G^H S^{Hi}$ to be of the form $\begin{bmatrix}\square & 0\\ 0 & 0\end{bmatrix}$, $\square = \square^H$. However, we also need $\square > 0$! It can be shown (see e.g. [32] for $\lambda = 1$) that this corresponds to requiring the scaled backward prediction polynomial $\overline{\mathcal{B}}_{N,k}\left(\frac{z}{\sqrt{\lambda}}\right)$ to be minimum phase. This requirement now, together with (56), allows for a specification of the manifold as $\overline{B}$ being an explicit function of $\overline{A}$,$\overline{C}$ via a spectral factorization operation:

$$\overline{\mathcal{M}}_{\frac{2}{3}} = \left\{\overline{\Theta} : \overline{B} = f_{\overline{\mathcal{M}}_{\frac{2}{3}}}\left(\overline{A},\overline{C}\right)\right\} \quad \text{where } f_{\overline{\mathcal{M}}_{\frac{2}{3}}} \text{ is defined by}$$

$$\begin{cases}\overline{\mathcal{B}}\left(\tfrac{z}{\sqrt{\lambda}}\right)\overline{\mathcal{B}}\left(\tfrac{1}{\sqrt{\lambda}z}\right) \;=\; \lambda^N\,\overline{\mathcal{A}}\left(\tfrac{\sqrt{\lambda}}{z}\right)\overline{\mathcal{A}}\left(\sqrt{\lambda}z\right)+\lambda^{N+1}\,\overline{\mathcal{C}}\left(\tfrac{\sqrt{\lambda}}{z}\right)\overline{\mathcal{C}}\left(\sqrt{\lambda}z\right)\\ \overline{\mathcal{B}}\left(\tfrac{z}{\sqrt{\lambda}}\right)\;\; \textit{minimum phase}\end{cases} \tag{57}$$

6.1.2 Sufficiency of $\overline{\mathcal{M}}_{\frac{2}{3}}$

Here we pose the reverse question: given $\overline{\Theta} \in \overline{\mathcal{M}}_{\frac{2}{3}}$, can we find a LS problem with the required structure, of which $\overline{\Theta}$ is the state. If $\overline{\Theta} \in \overline{\mathcal{M}}_{\frac{2}{3}}$, then from (53), we can find a matrix $R_{N,k}^{-1}$ that satisfies

$$\begin{bmatrix}R_{N,k}^{-1} & 0\\ 0 & 0\end{bmatrix}-\lambda\begin{bmatrix}0 & 0\\ 0 & R_{N,k}^{-1}+\overline{C}_{N,k}^H\overline{C}_{N,k}\end{bmatrix} = \overline{A}_{N,k}^H\overline{A}_{N,k}-\overline{B}_{N,k}^H\overline{B}_{N,k}\,. \tag{58}$$

Again, using a reasoning that is very similar to one in [32], one can show that the minimum phase property of $\overline{\mathcal{B}}_{N,k}\left(\frac{z}{\sqrt{\lambda}}\right)$ implies that $R_{N,k} > 0$. Hence $R_{N,k}$ exists and we can apply the matrix inversion lemma to obtain

$$\left(R_{N,k}^{-1} + \overline{C}_{N,k}^{H}\overline{C}_{N,k}\right)^{-1} = R_{N,k} - R_{N,k}\overline{C}_{N,k}^{H}\left(\overline{C}_{N,k}R_{N,k}\overline{C}_{N,k}^{H} + 1\right)^{-1}\overline{C}_{N,k}R_{N,k} \tag{59}$$

where the term being inverted on the RHS is greater than one, due to $R_{N,k} > 0$. Hence we can interpret the second term on the RHS as $X_N(k)X_N^H(k)$ with

$$X_N(k) \triangleq -R_{N,k}\overline{C}_{N,k}^{H}\left(\overline{C}_{N,k}R_{N,k}\overline{C}_{N,k}^{H} + 1\right)^{-1/2} . \tag{60}$$

We also introduce

$$R_{N,k-1}^{-1} \triangleq \lambda\left(R_{N,k}^{-1} + \overline{C}_{N,k}^{H}\overline{C}_{N,k}\right) . \tag{61}$$

which, using (59) and (60), satisfies

$$\lambda R_{N,k-1} = R_{N,k} - X_N(k)X_N^H(k) . \tag{62}$$

Using (58) and (61), we can introduce

$$R_{N+1,k}^{-1} \triangleq \begin{bmatrix} R_{N,k}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \overline{B}_{N,k}^{H}\overline{B}_{N,k} = \begin{bmatrix} 0 & 0 \\ 0 & R_{N,k-1}^{-1} \end{bmatrix} + \overline{A}_{N,k}^{H}\overline{A}_{N,k} . \tag{63}$$

This implies that $R_{N+1,k}$ is of the form

$$R_{N+1,k} = \begin{bmatrix} R_{N,k} & * \\ * & * \end{bmatrix} = \begin{bmatrix} * & * \\ * & R_{N,k-1} \end{bmatrix} . \tag{64}$$

Hence,

$$\begin{aligned} \nabla_{\frac{1}{\lambda}} R_{N+1,k} &= \begin{bmatrix} * & * \\ * & R_{N,k-1} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{\lambda}R_{N,k} \end{bmatrix} \\ &= r_{N+1}(k) r_{N+1}^{-0}(k) r_{N+1}^{H}(k) \\ &\quad - SS^H r_{N+1}(k) r_{N+1}^{-0}(k) r_{N+1}^{H}(k) SS^H - \tfrac{1}{\lambda}\begin{bmatrix} 0 \\ X_N(k) \end{bmatrix}\begin{bmatrix} 0 \\ X_N(k) \end{bmatrix}^H \end{aligned} \tag{65}$$

which is the required structure we found in (46).

6.1.3 $\mathcal{M}_{\frac{2}{3}}$ for the Unnormalized FTF Algorithm

To see what the corresponding manifold for the usual unnormalized FTF algorithm would be, consider the correspondences

$$\begin{aligned} \overline{A}_{N,k} &= \alpha_N^{-1/2}(k)A_{N,k} &\leftrightarrow\quad& A_{N,k}, \alpha_N(k) \\ \overline{B}_{N,k} &= \beta_N^{-1/2}(k)B_{N,k} &\leftrightarrow\quad& B_{N,k}, \beta_N(k) \\ \overline{C}_{N,k} &= \gamma_N^{1/2}(k)\widetilde{C}_{N,k} &\leftarrow\quad& \widetilde{C}_{N,k}, \gamma_N(k) \quad \text{but not} \;\rightarrow\; ! \end{aligned} \tag{66}$$

The unnormalized forward and backward prediction polynomials are obtained from their normalized counterparts by scaling a certain entry to unity. We also keep the squares of the scaling factors so that there is a conservation of the number of degrees of freedom. However, for the Kalman gain part, we cannot obtain $\widetilde{C}_{N,k}$ and $\gamma_N(k)$ uniquely from $\overline{C}_{N,k}$ alone. An extra condition is needed:

$$\gamma_N(k) = \lambda^N \beta_N(k)/\alpha_N(k) \ . \tag{67}$$

In the unnormalized case, the manifold $\mathcal{M}_{\frac{2}{3}}$ consists of the unnormalized version of (57) together with (67). In summarizing, we can note that $\Theta(k) \in \mathcal{M}_{\frac{2}{3}}$ implies that $\Theta(k)$ corresponds exactly to a certain prewindowed problem, and hence that it is *backward consistent* !

6.1.4 Minimal Parameterizations of $\mathcal{M}_{\frac{2}{3}}$

We can use the minimal parameterization $\Theta_m(k)$ introduced before in (47). It should now be easy to determine the transformations that allow the correspondences in the following diagram

$$\left.\begin{array}{r} r_{N+1}(k)\\ X_N(k)\end{array}\right\} \begin{array}{c}\nearrow\\ \longleftarrow\end{array} \left(\left\{\begin{array}{l}\left\{\begin{array}{c}A_{N,k}\\ \alpha_N(k)\\ \overline{C}_{N,k}\end{array}\right.\\ \quad\downarrow \mathcal{M}_{\frac{2}{3}} \\ \left\{\begin{array}{c}B_{N,k}\\ \beta_N(k)\\ \gamma_N(k)\end{array}\right.\end{array}\right.\right) = \Theta(k) \tag{68}$$

Thanks to $\mathcal{M}_{\frac{2}{3}}$, there is a 1-1 correspondence between $\{r_{N+1}(k), X_N(k)\}$ and the $\left\{A_{N,k}, \alpha_N(k), \overline{C}_{N,k}\right\}$ part of $\Theta(k)$.

6.2 The Minimal FTF Manifold $\mathcal{M}_{\frac{1}{3}}$

Though the FTF manifold $\mathcal{M}_{\frac{2}{3}}$ introduced above is a valid concept, there are actually more constraints to be satisfied for the algorithm to be backward consistent. This problem comes about as follows. The FTF algorithm drags a regression vector $X_N(k)$ along as part of its recursions. So the actual extended state of the (prediction part of the) FTF algorithm is $\Theta^e(k) = \left[\Theta(k)\;\; X_N^H(k)\right]$ and the FTF algorithm can be described as :

$$\left\{\begin{array}{rcl} \Theta(k) & = & f\left(\Theta(k-1), X_N(k-1), x(k)\right) \\ X_N(k) & = & S\, X_N(k-1) + u_1 x(k) \end{array}\right. \tag{69}$$

with $\Theta(k)$ as in (50). Actually, this $X_N(k)$ as part of $\Theta^e(k)$ may very well be different from the $X_N^{\mathcal{M}_{\frac{2}{3}}}(k)$ in the parameterization of the backward consistent part of $\Theta(k)$ (see (68)) ! Hence, the backward consistency requirement leads to an *extended manifold*

$$\mathcal{M}^e_{\frac{2}{3}} = \left\{\Theta^e(k) : \Theta(k) \in \mathcal{M}_{\frac{2}{3}},\; X_N(k) = X_N^{\mathcal{M}_{\frac{2}{3}}}(k)\right\} \tag{70}$$

where $X_N^{\mathcal{M}_{\frac{2}{3}}}(k)$ is a function of $\Theta(k)$ as indicated in (68).

Let us take another look at the minimal parameterization of $\mathcal{M}_{\frac{2}{3}}$. From the sufficiency part of $\mathcal{M}_{\frac{2}{3}}$, we can determine a matrix $R_{N+1,k}$ with displacement rank 3 and displacement structure $\nabla_{\frac{1}{\lambda}} R_{N+1,k} = G \Sigma G^H$ with $G = [G_1\, G_2\, G_3]$ and $\Sigma = diag\,\{1, -1, -1\}$. For a matrix $R_{N+1,k}$ computed from a $\Theta(k) \in \mathcal{M}_{\frac{2}{3}}$, the three generator vectors G_1, G_2 and G_3 are of the form shown in (46). In the developments thus far, we have actually tacitly assumed that $G_3 = \left[0 \;\, X_N^H(k)\right]^H$, where $X_N(k)$ is the regressor vector that is explicitly used in the FTF algorithm, and we have concentrated on the specific relation between the first and the second displacement generator vectors, requiring $G_2 = SS^H G_1$, which leads to the concept of $\mathcal{M}_{\frac{2}{3}}$. However, the requirement $\Theta(k) \in \mathcal{M}_{\frac{2}{3}}$ does not guarantee $X_N^{\mathcal{M}_{\frac{2}{3}}}(k) = X_N(k)$, and if $X_N^{\mathcal{M}_{\frac{2}{3}}}(k) \neq X_N(k)$, then the error propagation on $\mathcal{M}_{\frac{2}{3}}$ is not that of the minimal parameterization discussed before.

So the actual extended state $\Theta^e(k)$ of the FTF algorithm leads to more constraints in the backward consistency requirement, as displayed in the definition (70) of the extended manifold $\mathcal{M}^e_{\frac{2}{3}}$. For the purpose of further analysis, it is reasonable to assume that the $X_N(k)$ part of $\Theta^e(k)$ is error-free (any error on $X_N(k)$ is due to quantization of the signal $x(.)$ before it enters the FTF algorithm and as such has little to do with the error propagation in the FTF algorithm). So we shall again concentrate on the $\Theta(k)$ part and consider $X_N(k)$ as some external quantity. The constraints on $\Theta(k)$ implied by $\mathcal{M}^e_{\frac{2}{3}}$ will be renamed as $\mathcal{M}_{\frac{1}{3}}$ (only about $\frac{1}{3}$ of Θ contributes now to a minimal parameterization of $\mathcal{M}_{\frac{1}{3}}$ as we shall see below). $\Theta(k) \in \mathcal{M}_{\frac{1}{3}}$ means that $\Theta(k)$ corresponds to a prewindowed problem with $X_N(k)$ as last N data samples, but the preceding input samples are arbitrary. The manifold $\mathcal{M}_{\frac{1}{3}}$ can be defined as

$$\mathcal{M}_{\frac{1}{3}} = \left\{\Theta(k) : \left(B_{N,k}, \beta_N(k), \widetilde{C}_{N,k}, \gamma_N(k)\right) \;=\; f_{\mathcal{M}_{\frac{1}{3}}}\left(A_{N,k}, \alpha_N(k)\right)\right\} . \tag{71}$$

The function $f_{\mathcal{M}_{\frac{1}{3}}}$ can best be described by passing by the intermediate quantity $r_{N+1}(k)$ as follows. Using (46), one can write the normal equations for the forward prediction problem as

$$A_{N,k}\,\mathcal{T}\left(r_{N+1}(k)\right) \;=\; \left[\alpha_N(k)\; 0 \cdots 0\right] + A_{N,k}\,\mathcal{L}\left(\begin{bmatrix} 0 \\ X_N(k) \end{bmatrix}\right)\mathcal{L}\left(\begin{bmatrix} 0 \\ X_N(k) \end{bmatrix}\right)^H . \tag{72}$$

We can rewrite the LHS in the real scalar case as

$$A_{N,k}\,\mathcal{T}\left(r_{N+1}(k)\right) \;=\; r^H_{N+1}(k) I_{1,N}\left\{\widetilde{I}_{N+1}\mathcal{L}\left(\widetilde{I}_{N+1}A^H_{N,k}\right) + \mathcal{L}\left(A^H_{N,k}\right)^H\right\} \tag{73}$$

where $I_{1,N} = \frac{1}{2} \oplus I_N$ and $\widetilde{I}$ has ones on the antidiagonal and zeros elsewhere. Equations (72) and (73) combined allow one to solve for $r_{N+1}(k)$ given $A_{N,k}$, $\alpha_N(k)$ and $X_N(k)$. Given $r_{N+1}(k)$ and $X_N(k)$, the remaining quantities that make up $\Theta(k)$ are easily found. If $\Theta(k) \in \mathcal{M}_{\frac{1}{3}}$, then also the following constraint is satisfied

$$\gamma_N^{-1}(k) \;=\; 1 - \widetilde{C}_{N,k} X_N(k) \,. \tag{74}$$

Obviously, a minimal parameterization for $\mathcal{M}_{\frac{1}{3}}$ is $r_{N+1}(k)$ (given $X_N(k)$). The error propagation on $\mathcal{M}_{\frac{1}{3}}$ corresponds to the propagation of errors on $r_{N+1}(k)$. The error system associated with $r_{N+1}(k)$ is linear time-invariant and has $N+1$ eigenvalues equal to exactly λ (see (48)). This corrobarates our earlier findings recalled in (51) in which linearization and averaging techniques had lead us to conclude that one third of the $3N+3$ eigenvalues are approximately equal to λ, and these eigenvalues are more or less associated with the forward prediction quantities.

The decomposition of the error along tangential and normal components leads to two subsystems that are coupled only in a one-way sense. The tangential subsystem has the same dynamics as the system corresponding to the minimal parameterization $r_{N+1}(k)$ discussed above. The normal subsystem (of order $2N+2$) therefore contains the $N+1$ exponentially unstable modes $\frac{1}{\lambda}$ (in the unstabilized $7N$ FTF algorithm [9]) and the marginally unstable mode that we found in the averaging analysis [9]. The slightest deviation of the computed state from the manifold will thus lead to a further exponential running away from it.

6.3 Stabilization of the FTF Algorithm

The stabilization problem of the FTF algorithm can now simply be formulated as the problem of getting back to the manifold, once we deviate from it. The feedback mechanism considered in [9] is one way of accomplishing this goal. Actually, the so-called rescue procedures [15] can be considered as very rough projection operations onto the manifold. The original rescues [15] projected $\widehat{\Theta}(k)$ onto a fixed point on $\mathcal{M}_{\frac{1}{3}}$ ($R_{N+1,k}$

reduced to a diagonal matrix):

$$X_N^H(k) = \widetilde{C}_{N,k} = 0, \gamma_N(k) = 1, \mathcal{A}_{N,k}(z) = \mathcal{B}_{N,T}(z) = 1, \alpha_N(k) = \lambda^N \beta_N(k) = \mu \ . \quad (75)$$

This reduces the normal error component to zero, but leads to a huge error component within $\mathcal{M}_{\frac{1}{3}}$ which admittedly decays to zero exponentially. The following rescue point on $\mathcal{M}_{\frac{1}{3}}$ is much more data dependent and corresponds to approximating the prewindowed data matrix $X_{N+1,k}$ by a pre and postwindowed data matrix :

$$X_N^H(k) = \widetilde{C}_{N,k} = 0, \gamma_N(k) = 1, \mathcal{A}_{N,k}\left(z\sqrt{\lambda}\right) = \mathcal{B}_{N,k}\left(\frac{z}{\sqrt{\lambda}}\right), \ \beta_N(k) = \lambda^{-N}\alpha_N(k) \ . \quad (76)$$

Note that the prediction filters have to be initialized to minimum phase filters for $R_{N+1,k}$ to be positive definite though. So $B_{N,k}$ should be kept and $A_{N,k}$ adjusted accordingly, rather than the reverse, even though $B_{N,k}$ is more susceptible to error than $A_{N,k}$.

6.4 Relation to the Chandrasekhar Equations

The FTF algorithm has been shown to be a special instance of the Chandrasekhar equations in [29, chapter 7]. It actually involves an infinite-dimensional time-invariant state-space model. However, P_k is required to be non-zero only at N^2 well-defined positions. This requirement leads to the constraints defining $\mathcal{M}_{\frac{2}{3}}$. The requirements defining the Chandrasekhar manifold lead to the extra constraints $\mathcal{M}^c_{\frac{2}{3}}$, and from there to $\mathcal{M}_{\frac{1}{3}}$. The inequality constraints $P_k \geq 0$ lead to the minimum-phase requirement on the backward prediction polynomial (part of L_k). The Chandrasekhar constraints (38),(33) lead to the FTF constraints (67),(74) respectively.

7 The FLA/FQR Algorithms

As a final example, we consider the recently introduced Fast QR (FQR) algorithm [33],[34],[35], another fast RLS algorithm. This algorithm is intimately related to the Fast Lattice (FLA) algorithm [36]. The FLA/FQR algorithms are a fast version of square-root Kalman filtering, applied to the FIR filter parameter estimation problem. Hence, in the joint-process section, the filter estimate $W_{N,k}$ is replaced by $W_{N,k}R^{1/2}_{N,k}$ (see section 3.4), though the innovation $\epsilon_N(k)$ is still provided. The error propagation for this joint-process part of the algorithm state is exponentially stable with base λ (the closed-loop system matrix - see section 3.4). As usual, the joint-process part depends on the prediction section but not vice versa, so we can concentrate on the prediction part. Whereas the prediction part of the FTF algorithm provides the Kalman gain to the joint-process part, the Kalman gain $C_{N,k} = -X^H_N(k)R^{-1}_{N,k}$ is replaced by $-X^H_N(k)R^{-H/2}_{N,k}$ in the square-root algorithms, which is the set of backward prediction errors from order 0 to order $N-1$. These backward prediction errors are part of the algorithm state for the prediction part of the FLA/FQR algorithms. Historically, the backward prediction errors are computed for one order too high.

The propagation of numerical errors in the FLA algorithm has been analyzed in [12]. However, this analysis does not apply to the FQR algorithm, since a certain crucial assumption in this analysis (only order-update sweeps and hence causality in the order direction) is not satisfied in the FQR algorithm (which also has an order-downdate sweep). The FQR algorithm can easily be shown to be backward consistent, reducing the demonstration of the (before unknown) stable dynamics of the error propagation in this algorithm to a simple excercise. One may comment that the analysis of the error propagation in the FQR algorithm in the conventional way, using linearization steps and such, may be a formidable, if not impossible task.

7.1 The FQR Algorithm

The application of the backward consistency concept to this algorithm was first done in [10] to demontrate its stability. It may be useful to give an alternative exposition of this in light of the other applications discussed here. We shall use the notation from [10],[34]. The algorithm is stated in Table I of [34]. The quantities announced in this Table I as "available at time n" contain the state plus a number of intermediate quantities. To see what the state of the algorithm is, reduce the time index by one in the equations in Table I labeled as (5),(6) and (7), and start with equation (5) as the first equation for the set of operations to be done in one time update for the prediction part (in other words, in one time update of the prediction part, go through equations (5),(6),(1),(2),(3),(4)). This reveals the following quantities as being the state of the prediction part

$$\Theta(k) = \left[x_{f,0}(k), \ldots, x_{f,N-1}(k), E_{f,N}^{1/2}(k), \overline{\epsilon}_{b,0}(k), \ldots, \overline{\epsilon}_{b,N-1}(k)\right] \tag{77}$$

where $x_{f,n}(k)$ are scaled reflection coefficients, $\overline{\epsilon}_{b,n}(k)$ are scaled backward prediction errors, and $E_{f,N}(k)$ is the forward prediction error covariance of order N. However, this prediction part goes one order too high. By substituting $N \rightarrow N-1$ in the prediction part and in the state (77), we get the prediction part state dimension down from $2N+1$ to $2N-1$. The following additional operations have to be added to the joint-process part though (the γ in [34] is the square-root of the γ we have used before)

$$\begin{array}{rcl} \Theta_{N-1}(k+1) & = & \sin^{-1}\left(\frac{\overline{\epsilon}_{b,N-1}(k+1)}{\gamma_{N-2}(k+1)}\right) \\ \gamma_{N-1}(k+1) & = & \gamma_{N-2}(k+1)\cos\theta_{N-1}(k+1) \quad . \end{array} \tag{78}$$

We now have the state dimension down to the minimal $2N-1$, so we expect the manifold $\mathcal{M}$ to be the complete $(2N-1)$-dimensional Euclidean space. However, we may find a non-trivial set $\mathcal{S}$, determined by inequality constraints.

To determine $\mathcal{S}$, let us try to find a prediction problem of which a given vector $\Theta(k)$ is the resulting state, and see which constraints we meet in the process. First we have

$$\gamma_{n+1}^2(k) \; = \; \gamma_n^2(k) - (\epsilon_{b,n}(k))^2 \quad , \quad n = 0, \ldots, N-2 \tag{79}$$

where $\gamma_0(k) = 1$. Since $\gamma_n(k) \in [0,1]$, the above process requires $\|\underline{\epsilon}_b(k)\| \leq 1$ where $\underline{\epsilon}_b = [\overline{\epsilon}_{b,0} \cdots \overline{\epsilon}_{b,N-2}]'$. Next, we have

$$E_{f,n}(k) \; = \; E_{f,n+1}(k) + (x_{f,n}(k))^2 \quad , \quad n = N-2, \ldots, 0 \tag{80}$$

This process does not impose any constraints apart from the fact that we have $E_{f,N-1}^{1/2}(k) > 0$ of course. Now we are ready to determine

- $K_n(k) \;=\; x_{f,n}(k)/\sqrt{E_{f,n}(k)}$, $n = 0, \ldots, N-2$, the reflection coefficients
- $\overline{\epsilon}_{b,n}(k)/\gamma_n(k)$, $n = 0, \ldots, N-2$, the normalized backward prediction errors.

These two sets of quantities are all we need to determine the normalized backward prediction filters $\overline{B}_n(k)$, $n = 0, \ldots, N-1$ with the Levinson step-up procedure. The filter of order zero is given by $E_{b,0}^{-1/2}(k) = E_{f,0}^{-1/2}(k)$. We can stack these backward prediction filters of increasing order and length in an upper-triangular matrix, viz.

$$\mathbf{U}_N(k) \;=\; \begin{bmatrix} \overline{B}_0^{'}(k) & \overline{B}_1^{'}(k) & & \\ 0 & & \ddots & \overline{B}_{N-1}^{'}(k) \\ 0 & 0 & & \end{bmatrix} \tag{81}$$

This leads to the UL factorization of an inverse sample covariance matrix of size N:

$$\mathbf{P}_N(k) \;=\; \mathbf{U}_N(k)\mathbf{U}_N^{'}(k) \tag{82}$$

Let $\mathbf{U}_{N-1}(k)$ be the top left square submatrix of $\mathbf{U}_N(k)$ of size $N-1$, then we can determine a data vector as

$$\mathbf{x}_{N-1}(k) \;=\; \mathbf{U}_{N-1}^{-'}(k)\,\underline{\epsilon}_b(k) \tag{83}$$

The matrix $\mathbf{P}_N(k)$ in (82) has $2N-1$ degrees of freedom by construction. If one considers the displacement of $\mathbf{P}_N^{-1}(k)$ (which is of rank three), then one finds out that it is determined by the first row of $\mathbf{P}_N^{-1}(k)$ and the vector $\mathbf{x}_{N-1}(k)$ in (83), thus $2N-1$ parameters in all (as in the FTF algorithm, which is a different algorithm for the same problem). So $\mathbf{P}_N(k)$ is also determined by these $2N-1$ parameters. Let us now consider the construction of a data matrix with $2N-1$ data points of which the last $N-1$ are given by $\mathbf{x}_{N-1}(k)$ in (83) above. Let $\mathbf{X}_N(k)$ be a prewindowed data matrix built from the data

$$\{\ldots, 0, x(k-3N+3), \ldots, x(k-2N+2), 0, \ldots, 0, x(k-N+2), \ldots, x(k)\} \tag{84}$$

where $\{x(k), \ldots, x(k-N+2)\}$ are from $\mathbf{x}_{N-1}(k)$ in (83) above and $\{x(k-2N+2), \ldots, x(k-3N+3)\}$ are to be determined such that

$$\mathbf{X}_N^{'}(k)\mathbf{X}_N(k) \;=\; \mathbf{P}_N^{-1}(k) \ . \tag{85}$$

One could alternatively envisage a data sequence of $2N-1$ contiguous data samples, without the $N-1$ zeros in between as in (84) above, but the approach taken here is simpler to formulate. Consider now the matrix

$$\mathcal{T}_N(k) \;=\; \mathbf{P}_N^{-1}(k) - \mathcal{U}'\left(\mathbf{x}_{N-1}(k)\right)\mathcal{U}\left(\mathbf{x}_{N-1}(k)\right) \tag{86}$$

where $\mathcal{U}(v)$ is an upper-triangular Hankel matrix with v as first column. One can easily show that $\mathcal{T}_N(k)$ is a Toeplitz matrix (pre and postwindowed case). One can take the symmetric covariance sequence of length $2N-1$ corresponding to $\mathcal{T}_N(k)$, apply spectral factorization and e.g. take the minimum-phase factor with the first coefficient being positive. If one takes $\{x(k-2N+2),\ldots,x(k-3N+3)\}$ to be the sequence of length N corresponding to this spectral factor, then with this choice of the data samples, equation (85) will be satisfied. So we find that with any state vector $\Theta(k)$ that satisfies $E^{1/2}_{f,N-1}(k) > 0$, $\|\underline{\epsilon}_b(k)\| \leq 1$, we can associate a prewindowed problem with $2N-1$ data points. Hence the backward consistency and hence the exponential stability with base λ. The joint-process part is trivially shown to be also backward consistent (there are no constraints here, just as there are no constraints on the joint-process state of any of the algorithms considered here).

7.2 FLA Algorithms

The normalized FLA algorithm [37] can be handled in the same way as the FQR algorithm above. So the normalized FLA algorithm is also backward consistent and its prediction part state consists of $N-1$ backward prediction errors, $N-1$ reflection coefficients and the input signal energy. The inequality constraints are in this case the input signal energy being positive, and the reflection coefficients being bounded by 1 in magnitude.

The unnormalized FLA algorithm with order-updates for the forward and backward prediction error variances corresponds to a QDR (R unit-diagonal) factorization of the data matrix instead of a QR factorization. Here the backward consistent part of the state Θ can be shown to correspond to the Q and R factors of the QDR factorization of a

Toeplitz data matrix. However, the complementary part of Θ corresponds to the diagonal matrix D, which may be perturbed w.r.t. its value given by the QDR factorization of this Toeplitz matrix. A simple application of the analysis in [12] shows that such perturbations lead to stable dynamics. Similarly, the complementary part of Θ for any FLA algorithm can be shown to be exponentially stable (with base λ) using the techniques of [12]. Indeed, all FLA algorithms are composed of order-update sweeps only so that any given section of the lattice filter only depends on the lower-order sections. This decoupling allows for the separate analysis of each section, which renders the analysis extremely simple.

Suitable position for Table I

It is possible to come up with unnormalized FLA algorithms that are backward consistent. One such algorithm is provided in Table I. Its derivation is analogous to the derivation of any unnormalized FLA algorithm [30]. The state of the algorithm comprises $r_n(k)$, $n = 0, \ldots, N-2$, $\Delta_n(k)$, $n = 0, \ldots, N-2$, and $\beta_0(k)$, or $2N-1$ components total. Though the state dimension is minimal, the computational complexity of the algorithm is clearly not minimal. The point is that for the FLA algorithms, backward consistency is not required to assure good numerical behavior since in the FLA algorithms that are not backward consistent, the complementary part has the same exponentially stable dynamics as the minimal part.

8 Concluding Remarks

We have introduced the concept of backward consistency. This concept leads to minimal parameterizations of the state of a recursive algorithm (or rather, of the problem it is solving) and to the introduction of a manifold. Using the manifold, the error system can in general be decomposed into two subsystems. Since the coupling between these two subsystems is one way, the dynamics of both subsystems can be analyzed separately. The analysis of the minimal part reduces to the problem of the convergence of the algorithm from arbitrary initial conditions. The analysis of the complementary part still requires linearization and averaging, which are the classical tools in the analysis of the propagation of roundoff errors. Sometimes, these tools may be avoided by embedding the problem into a larger, less restrictive class of backward consistent problems. We have applied the new technique to a variety of least-squares algorithms. In concluding, we would like to comment on the following issues.

8.1 Dynamics of the Complementary Subsystem

Stability or instability of the the nonminimal part is often related to

- open-loop system stability. The open-loop stability refers to the eigenvalues of F_k or F_k^{-1}, depending on whether we deal with a covariance form (propagation of P_k) or information form (propagation of P_k^{-1}) formulation respectively. For instance, we have $F_k = \frac{1}{\sqrt{\lambda}} I$ for the FIR filter parameter estimation problem with exponential weighting (at least for the prediction part). The FLA algorithms which propagate the forward and backward prediction error variances $\alpha_n(k)$, $\beta_n(k)$ directly are fast versions of square-root information form Kalman filters, while the FLA algorithms propagating $\alpha_n^{-1}(k)$, $\beta_n^{-1}(k)$ are fast forms of square-root covariance form Kalman filters. Stable dynamics of the complementary part in the first class of algorithms can be explained by the stability of F_k^{-1}.

- whether some feedback applies. This feedback may be the inherent feedback of the Kalman filter. This explains why the ΔP_k^a subsystem has the same stable dynamics as the ΔP_k^s subsystem for the case of the backward consistent non-symmetric Riccati equation. This also explains the exponential stability of the second class of FLA algorithms mentioned above. Feedback may also be of a different nature, as in the stabilized FTF algorithm [9].

8.2 Persistent Excitation

Singularity of P_k (due to loss of controllability and/or observability) leads to instability if the open-loop system is unstable, even for the backward consistent subsystem. In fact, when P_k becomes singular, the minimal parameterization is no longer minimal. Taking a closer look at the prewindowed RLS problem in particular, non-persistent excitation means that the input signal consists of fewer than N (complex) exponentials [38]. We see that due to prewindowing, singularity may strictly speaking only occur in the limit, as time goes to infinity. Of course, with exponential weighting, the convergence towards singularity is exponentially fast (in the absence of persistent excitation).

8.3 Initialization of Fast RLS Algorithms

Adding a so-called soft constraint [15] to the least-squares criterion leads to a regularized problem right from the start. Again, in the absence of persistent excitation, the problem will converge to a singular one, and this exponentially fast if exponential weighting is used. When no soft constraint is used, all fast RLS algorithms take on an order-increasing structure during the first N time steps. In this way, extra minimality problems due to underdeterminedness are avoided by considering exactly determined problems of increasing size during the first N time steps.

8.4 Structured Condition Numbers

Recently, the notion of a structured condition number has been introduced in numerical analysis circles [39],[40],[41]. The structured condition number is the condition number that one obtains, when perturbations of the problem at hand are restricted to obey certain structural constraints. This is an application of the backward consistency concept to classical numerical analysis notions.

Finally, we may remark that it is possible to apply the backward consistency concept to least-squares algorithms that are not recursive in time, but in order. More specifically, consider the Levinson and Schur algorithms [30] for solving a set of equations with a Toeplitz matrix of coefficients. The Schur algorithm turns out not to be backward consistent for the class of Toeplitz matrices, but it is backward consistent for the larger class of quasi-Toeplitz matrices (characterized by a displacement rank equal to two). The Levinson algorithm on the other hand is not backward consistent, for any class of problems. This appears to suggest that it may be possible to come up with a version of the Schur algorithm that would be better behaved numerically than the Levinson algorithm.

References

[1] J.H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon, Oxford, 1965.

[2] D.T.M. Slock. "An Overview of Some Recent Advances in Fast RLS Algorithms". In E. Deprettere and A.-J. van der Veen, editors, *Algorithms and Parallel VLSI Architectures*, volume A: Tutorials. Elsevier Science Publishers, Amsterdam, 1991. Proc. of Int. Workshop, Pont-à-Mousson, France, June 10-16, 1990.

[3] D.T.M. Slock. "The FTF Manifold and its Role in the Numerical Behavior of Fast Transversal Filter RLS Algorithms". In *Proc. ICASSP 91 Conf.*, pages 3705–3708, Toronto, Canada, May 14-17 1991.

[4] D.T.M. Slock. "The Backward Consistency Concept and a New Decomposition of the Error Propagation Dynamics in Recursive Least-Squares Algorithms". In *Proc. Adaptive Signal Processing, part of SPIE's Technical Symposium*, San Diego, USA, July 21-26 1991.

[5] J.R. Bunch. "The Weak and Strong Stability of Algorithms in Numerical Linear Algebra". *Linear Algebra and its Applications*, 88/89:49–66, 1987.

[6] M.H. Verhaegen and P. Van Dooren. "Numerical Aspects of Different Kalman Filter Implementations". *IEEE Trans. Autom. Control*, AC-31(10):907–917, Oct. 1986.

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

[8] M. Morf and T. Kailath. "Square-Root Algorithms for Least-Squares Estimation". *IEEE Trans. Autom. Cont.*, AC-20(4):487–497, Aug. 1975.

[9] D.T.M. Slock and T. Kailath. "Numerically Stable Fast Transversal Filters for Recursive Least-Squares Adaptive Filtering". *IEEE Trans. Signal Proc.*, ASSP-39(1):92–114, Jan. 1991.

[10] P.A. Regalia. "Numerical Stability Properties of a QR-Based Fast Least-Squares Algorithm". IEEE Trans. SP, submitted.

[11] H. Kushner. *Approximation and Weak Convergence Methods for Random Processes, with Applications to Stochastic Systems Theory.* The MIT Press, Cambridge, MA, 1984.

[12] S. Ljung and L. Ljung. "Error Propagation Properties of Recursive Least-Squares Adaptation Algorithms". *Automatica*, 21(2):157–167, 1985.

[13] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1975.

[14] B. Widrow *et al.* "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter". *Proc. IEEE*, 64(8):1151–1162, Aug. 1976.

[15] J.M. Cioffi and T. Kailath. "Fast, recursive least squares transversal filters for adaptive filtering". *IEEE Trans. on ASSP*, ASSP-32(2):304–337, April 1984.

[16] D.T.M. Slock. "On the Convergence Behavior of the LMS and the Normalized LMS Algorithms". Submitted to IEEE Trans. on SP.

[17] M.H. Verhaegen. "Round-off Error Propagation in Four Generally Applicable, Recursive, Least-Squares Estimation Schemes". *Automatica*, 25(3):437–444, 1989.

[18] H. Aasnaes and T. Kailath. "Initial-Condition Robustness of Linear Least-Squares Filtering Algorithms". *IEEE Trans. Autom. Contr.*, AC-19(4):393–397, Aug. 1974.

[19] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, NJ, 1979.

[20] M.H. Verhaegen. "Improved Understanding of the Loss-of-Symmetry Phenomenon in the Conventional Kalman Filter". *IEEE Trans. Autom. Control*, AC-34(3):331–333, March 1989.

[21] T. Söderström and P.G. Stoica. *Instrumental Variable Methods for System Identification.* Springer-Verlag, Berlin, 1983.

[22] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* M.I.T. Press, Cambridge, MA, 1983.

[23] R.J. Fitzgerald. "Divergence of the Kalman Filter". *IEEE Trans. Automat. Contr.*, AC-16:736–747, Dec. 1971.

[24] J.F. Bellantoni and K.W. Dodge. "A Square Root Formulation of the Kalman-Schmidt Filter". *AIAA J.*, 5:1309–1314, July 1967.

[25] F.H. Schlee, C.J. Standish, and N.F. Toda. "Divergence in the Kalman Filter". *AIAA J.*, 5:1114–1120, June 1967.

[26] G.E. Bottomley and S.T. Alexander. "A Novel Approach for Stabilizing Recursive Least Squares Filters". *IEEE Trans. Signal Processing*, 39:1770–1779, Aug. 1991.

[27] P.J. Radonja. "Roundoff Noise in Signal Estimation and Minimization of Total Real Estimator Output Noise". *IEEE Trans. ASSP*, ASSP-38:2075–2087, Dec. 1990.

[28] T. Kailath, A. Vieira, and M. Morf. "Orthogonal Transformation (Square-Root) Implementations of the Generalized Chandrasekhar and Generalized Levinson Algorithms". In A. Bensoussan and J. Lions, editors, *Int. Symp. on Systems Optimiza-*

tion and Analysis, Rocquencourt, Dec. 1978, volume 14, pages 81–91. New York: Springer-Verlag, 1979. Lecture Notes in Control and Information Sciences.

[29] D.T.M. Slock. *Fast algorithms for Fixed-Order Recursive Least-Squares Parameter Estimation.* PhD thesis, Stanford University, Stanford, CA, USA, September 1989.

[30] S. Haykin. *Adaptive Filter Theory.* Prentice-Hall, Englewood Cliffs, NJ, 1991. second edition.

[31] H. Lev-Ari and T. Kailath. "Triangular Factorization of Structured Hermitian Matrices". In I. Gohberg, editor, *I. Schur Methods in Operator Theory and Signal Processing, Operator Theory: Advances and Applications*, volume 18, pages 301–324. Birkhäuser, Boston, 1986.

[32] P. Stoica and A. Nehorai. "On Stability and Root Location of Linear Prediction Models". *IEEE Trans. ASSP*, ASSP-35:582–584, April 1987.

[33] J.M. Cioffi. "The Fast Adaptive Rotors RLS Algorithm". *IEEE Trans. on ASSP*, ASSP-38(4):631–653, April 1990.

[34] P.A. Regalia and M.G. Bellanger. "On the Duality Between Fast QR Methods and Lattice Methods in Least-Squares Adaptive Filtering". *IEEE Trans. Signal Processing*, 39:879–891, April 1991.

[35] D.T.M. Slock. "Reconciling Fast RLS Lattice and QR Algorithms". In *Proc. ICASSP 90 Conf.*, pages 1591–1594, Albuquerque, NM, April 3–6 1990.

[36] D.T.L. Lee, M. Morf, and B. Friedlander. "Recursive least-squares ladder estimation algorithms". *IEEE Trans. ASSP*, ASSP-29(3):627–641, June 1981.

[37] B. Friedlander. "Lattice filters for adaptive processing". *Proceedings IEEE*, 70(8):829–867, Aug. 1982.

[38] G.C. Goodwin and K.S. Sin. *Adaptive Filtering Prediction and Control.* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[39] P. Van Dooren. “Structured Linear Algebra Problems in Digital Signal Processing”. In G. Golub and P. Van Dooren, editors, *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, volume F70. Springer-Verlag, New York, 1990. Proc. of NATO ASI, Leuven, Belgium, Aug. 1–12, 1988.

[40] D.J. Higham and N.J. Higham. “Backward Error and Condition of Structured Linear Systems”. submitted, SIAM J. Matrix Analysis, Sept. 1990.

[41] I. Gohberg and I. Koltracht. “Error Analysis for Cauchy and Vandermonde Matrices”. In *Proc. MTNS*, Amsterdam, July 1989.

Table I

Backward Consistent Unnormalized FLA Algorithm

$e_0(k) = r_0(k) = x(k), \gamma_0(k-1) = \gamma_0(k) = 1,$

$\alpha_0(k) = \beta_0(k) = \lambda\beta_0(k-1) + |x(k)|^2, \epsilon_0(k) = d(k)$

Prediction Part : *for* $n = 1, \ldots, N-1$ *do*

$$\begin{aligned}
\Delta_{n-1}(k) &= \lambda\Delta_{n-1}(k-1) + r_{n-1}(k-1)\gamma_{n-1}^{-1}(k-1)e_{n-1}^{H}(k) \\
e_n(k) &= e_{n-1}(k) - \Delta_{n-1}^{H}(k)\beta_{n-1}^{-1}(k-1)r_{n-1}(k-1) \\
r_n(k) &= r_{n-1}(k-1) - \Delta_{n-1}(k)\alpha_{n-1}^{-1}(k)e_{n-1}(k) \\
\alpha_n(k) &= \alpha_{n-1}(k) - |\Delta_{n-1}(k)|^2 \beta_{n-1}^{-1}(k-1) \\
\beta_n(k) &= \beta_{n-1}(k-1) - |\Delta_{n-1}(k)|^2 \alpha_{n-1}^{-1}(k) \\
\gamma_n(k-1) &= \gamma_{n-1}(k-1) - |r_{n-1}(k-1)|^2 \beta_{n-1}^{-1}(k-1) \\
\gamma_n(k) &= \gamma_{n-1}(k) - |r_{n-1}(k)|^2 \beta_{n-1}^{-1}(k) \\
\beta_n(k-1) &= \frac{1}{\lambda}\left\{\beta_n(k) - |r_n(k)|^2 \gamma_n^{-1}(k)\right\}
\end{aligned}$$

Joint-Process Part : *for* $n = 0, \ldots, N-1$ *do*

$$\begin{aligned}
\rho_n(k) &= \lambda\rho_n(k-1) + r_n(k)\gamma_n^{-1}(k)\epsilon_n^{H}(k) \\
\epsilon_{n+1}(k) &= \epsilon_n(k) - \rho_n^{H}(k)\beta_n^{-1}(k)r_n(k)
\end{aligned}$$

List of Figures

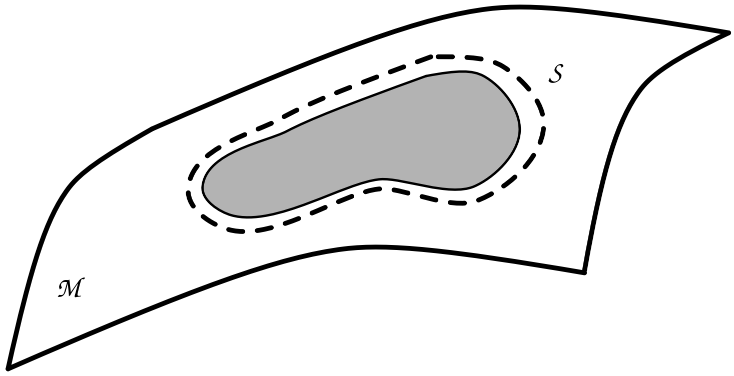

Figure 1:

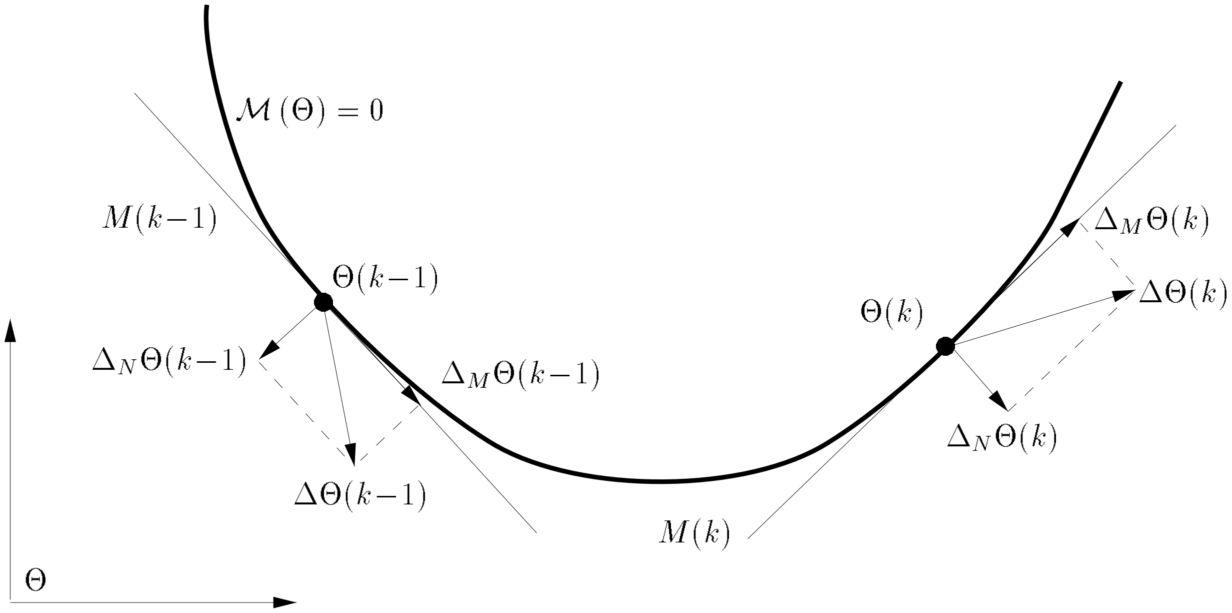
$\mathcal{M}(\Theta) = 0$
$M(k-1)$
$\Theta(k-1)$
$\Delta_N\Theta(k-1)$
$\Delta_M\Theta(k-1)$
$\Delta\Theta(k-1)$
$\Theta(k)$
$\Delta_M\Theta(k)$
$\Delta\Theta(k)$
$\Delta_N\Theta(k)$
$M(k)$
Θ

Figure 2:

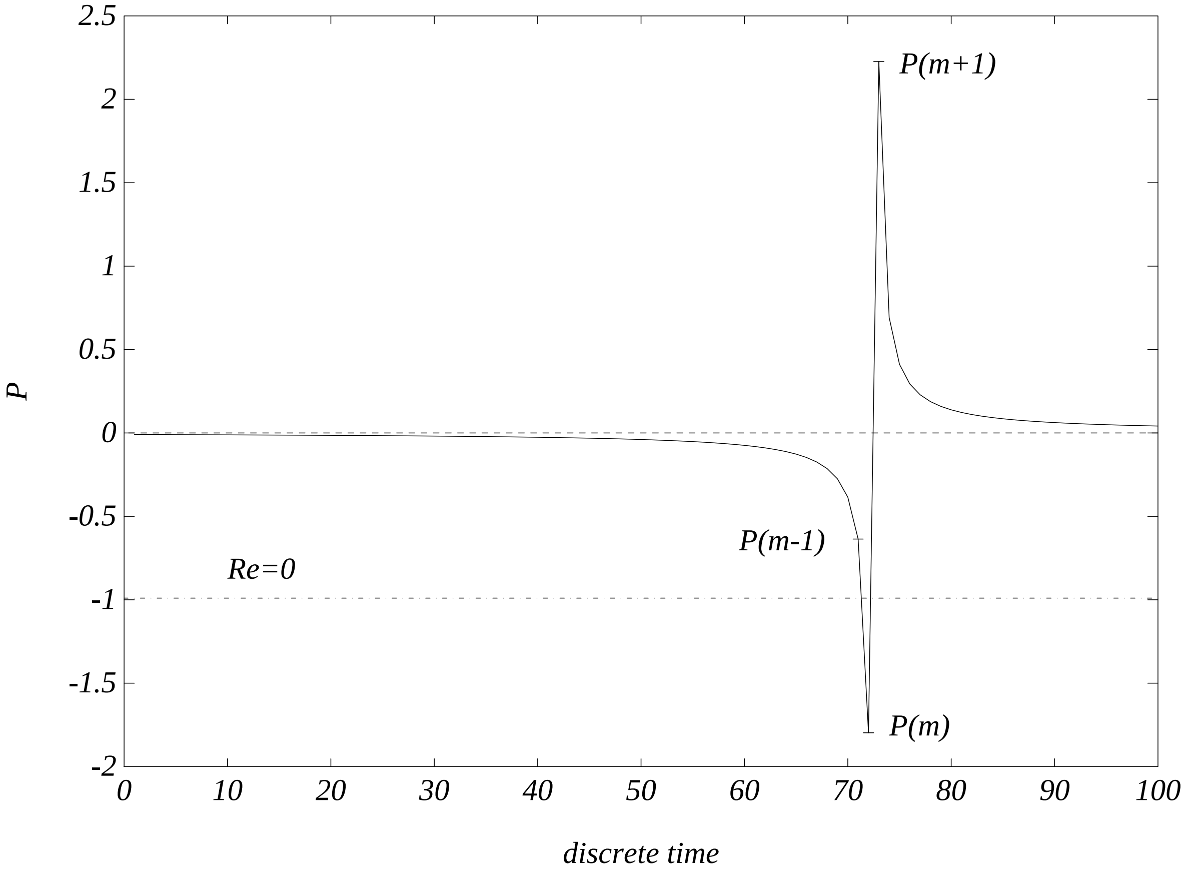
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
P
0
10
20
30
40
50
60
70
80
90
100
discrete time
P(m+1)
P(m-1)
P(m)
Re=0

Figure 3: