GENERALIZATION OF EXPONENTIALLY WEIGHTED RLS ALGRORITHM BASED ON A STATE-SPACE MODEL

Byungjin Chun, Beomsup Kim and Yong Hoon Lee

Dept. of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon, 305-701, Korea

ABSTRACT

We develop a generalized RLS (G-RLS) algorithm described by a state-space model through some modification of the procedure for Kalman filter derivation. It is shown that the G-RLS algorithm reduces to the conventional RLS when the state transition matrix is an identity matrix, and that the G-RLS algorithm without exponential weighting and Kalman filtering become identical when the state model is an unforced dynamical model. The G-RLS algorithm does not require model statistics, and can be implemented once the forgetting factor is chosen. The performances of the G-RLS and Kalman filtering are compared through computer simulation. Specifically, they are applied to the derivation of variable loop gains of a digital phase-locked loop (DPLL). The results indicate that the G-RLS algorithm can act like the Kalman filter if its forgetting factor is properly chosen.

1. INTRODUCTION

It has been recognized that the RLS algorithm can be viewed as a special case of Kalman filtering. In [1] [2] [3], it is shown that a Kalman filter described by a random-walk state model, whose state transition matrix equals the identity matrix, reduces to the exponentially weighted RLS algorithm when the process noise correlation matrix is properly given in terms of the forgetting factor, the Kalman gain and the estimation error covariance matrix. More recently, in [4], the RLS algorithm and its variants are derived in a unified manner from Kalman filtering described by an unforced dynamical state model whose state transition matrix is equal to a constant multiple of the identity matrix (see also [5]). Based on these results, some extensions of the RLS algorithm exhibiting better tracking performance than the conventional one are introduced in [5] [6].

The objective of this paper is to develop another extension of the RLS algorithm by exploiting Kalman filter theory. The proposed algorithm, which will be referred to as the generalized RLS (G-RLS) algorithm, is based on a general state model whose transition matrix is *not* a constant multiple of the identity matrix. This algorithm is developed through some modification of the procedure for the Kalman filter derivation in [7], and can be implemented without the knowledge of model statistics. It will be shown through computer simulation that the G-RLS can act like the Kalman filter if its forgetting factor is properly chosen.

2. THE G-RLS ALGORITHM

The G-RLS algorithm is based on the least squares (LS) estimation theory. In what follows, this algorithm is derived after briefly reviewing the LS theory.

2.1. The LS Estimation Theory

Consider the following measurement equation:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{y}$ is an N-dimensional measurement vector; $\mathbf{x}$ is an M-dimensional parameter vector to be estimated ($N \geq M$); $\mathbf{H}$ is a known N-by-M matrix of full rank; and $\mathbf{n}$ is an N-dimensional noise vector with zero mean and covariance matrix $\mathbf{R}$. In LS estimation, the problem is to find an estimate $\hat{\mathbf{x}}$ of the parameter $\mathbf{x}$ as a linear combination of the measurements $\mathbf{y}$ so that the estimate $\hat{\mathbf{x}}$ minimizes the following cost function:

$$J(\hat{\mathbf{x}}) = (\mathbf{y} - \mathbf{H}\hat{\mathbf{x}})^T \mathbf{W} (\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) \tag{2}$$

where $\mathbf{W}$ is an N-by-N weight matrix, and T denotes transpose. The solution to this problem is given by

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y}. \tag{3}$$

This estimate is unbiased [8], and its error covariance matrix is expressed as

$$E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1} \cdot \mathbf{H}^T\mathbf{W}\mathbf{R}\mathbf{W}\mathbf{H}(\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}. \tag{4}$$

When the weight matrix $\mathbf{W}$ is set equal to the inverse of the noise covariance matrix $\mathbf{R}$, the estimate in (3) becomes the minimum variance estimator, which plays a fundamental role in Kalman filter derivation [7]. Another important special case of the estimate in (3) is derived by setting $\mathbf{W}$ to the following diagonal matrix:

$$\mathbf{W} = diag(1,\ \lambda,\ \lambda^2,\ \cdots,\ \lambda^{N-1}) \tag{5}$$

where $0 < \lambda \leq 1$ and λ^k $(0 \leq k \leq N-1)$ represents the $(k+1)$-th diagonal element. In this case, the cost function in (2) becomes

$$J(\hat{\mathbf{x}}) = \sum_{k=0}^{N-1} \lambda^{N-1-k} (y_k - \mathbf{h}_k^T\hat{\mathbf{x}})^2 \tag{6}$$

0-7803-4455-3/98/$10.00 © 1998 IEEE

where y_k is the $(N-k)$-th element of $\mathbf{y}$ and $\mathbf{h}_k^T$ is the $(N-k)$-th row of $\mathbf{H}$. This cost function leads to the exponentially weighted RLS algorithm with the forgetting factor λ. In the sequel, the G-RLS algorithm is derived by replacing a noise covariance matrix that appears in Kalman filter derivation with a diagonal weight matrix which is an extension of the one in (5).

2.2. Derivation of the G-RLS Algorithm

We now consider a state-space model described by the following pair of equations:

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{w}_k \quad (7)$$

$$y_k = \mathbf{h}_k^T\mathbf{x}_k + v_k \quad (8)$$

where k is a nonnegative integer; $\mathbf{A}_k$ is a known M-by-M state transition matrix; $\mathbf{x}_k$ is the M-dimensional state (or parameter) vector; $\mathbf{w}_k$ is an M-dimensional process noise vector; y_k is the measurement; $\mathbf{h}_k$ is a known M-dimensional vector; and v_k is white noise with zero mean and variance σ_k^2. These equations can be combined to yield an equation that has the same form as (1). To be specific, we express y_{k-i}, $0 \le i \le k$, in terms of $\mathbf{x}_k$. Form (7), $\mathbf{x}_{k-i}$ is written as

$$\begin{aligned}\mathbf{x}_{k-i} &= \Big(\prod_{j=k-i}^{k-1}\mathbf{A}_j^{-1}\Big)\mathbf{x}_k - \sum_{m=1}^{i}\Big(\prod_{j=k-i}^{k-m}\mathbf{A}_j^{-1}\Big)\mathbf{w}_{k-m} \\ &= \mathbf{\Phi}_k^{k-i}\mathbf{x}_k - \sum_{m=1}^{i}\mathbf{\Phi}_{k-m+1}^{k-i}\mathbf{w}_{k-m} \quad (9)\end{aligned}$$

where

$$\mathbf{\Phi}_p^q = \prod_{r=q}^{p-1}\mathbf{A}_r^{-1}, \qquad p > q \quad (10)$$

is the backward transition matrix relating the states $\mathbf{x}_p$ and $\mathbf{x}_q$. Using (9) in (8), we get

$$y_{k-i} = \mathbf{h}_{k-i}^T\mathbf{\Phi}_k^{k-i}\mathbf{x}_k - \mathbf{h}_{k-i}^T\Big(\sum_{m=1}^{i}\mathbf{\Phi}_{k-m+1}^{k-i}\mathbf{w}_{k-m}\Big) + v_{k-i}. \quad (11)$$

If we define the $(k+1)$-by-M matrix

$$\mathbf{H}_k = [\mathbf{h}_k, (\mathbf{h}_{k-1}^T\mathbf{\Phi}_k^{k-1})^T, (\mathbf{h}_{k-2}^T\mathbf{\Phi}_k^{k-2})^T, \cdots, (\mathbf{h}_0^T\mathbf{\Phi}_k^0)^T]^T \quad (12)$$

and the $(k+1)$-dimensional vector

$$\begin{aligned}\mathbf{u}_k = -[0, &(\mathbf{h}_{k-1}^T\mathbf{\Phi}_k^{k-1}\mathbf{w}_{k-1})^T, (\mathbf{h}_{k-2}^T\sum_{m=1}^{2}\mathbf{\Phi}_{k-m+1}^{k-2}\mathbf{w}_{k-m})^T \\ &, \cdots, (\mathbf{h}_0^T\sum_{m=1}^{k}\mathbf{\Phi}_{k-m+1}^{0}\mathbf{w}_{k-m})^T]^T, \quad (13)\end{aligned}$$

then the set of measurements $\{y_k, y_{k-1}, \cdots, y_0\}$ is expressed as

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{n}_k \quad (14)$$

Table 1: Summary of the Kalman filter.

Initial conditions:
$\hat{\mathbf{x}}_{0|-1} = \bar{\mathbf{x}}_0$, $\mathbf{P}_{0|-1} = \mathbf{P}_0$
Prediction: $k = 1, 2, 3, \cdots$
$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}$
$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}$
Filtering: $k = 0, 1, 2, \cdots$
$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(y_k - \mathbf{h}_k^T\hat{\mathbf{x}}_{k|k-1})$
$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{h}_k^T)\mathbf{P}_{k|k-1}$
$\mathbf{K}_k = \dfrac{\mathbf{P}_{k|k-1}\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{P}_{k|k-1}\mathbf{h}_k + \sigma_k^2}$
Notations:
$\hat{\mathbf{x}}_{p|q}$ is the estimate of the state $\mathbf{x}_p$ given the input data up to time q.
$\mathbf{P}_{p|q}$ is the covariance matrix of the error vector $\mathbf{P}_{p|q} = E[(\mathbf{x}_p - \hat{\mathbf{x}}_{p|q})(\mathbf{x}_p - \hat{\mathbf{x}}_{p|q})^T]$.
$\mathbf{K}_k$ is the Kalman gain vector.

where $\mathbf{y}_k = [y_k, y_{k-1}, \cdots, y_0]^T$, $\mathbf{n}_k = \mathbf{u}_k + \mathbf{v}_k$ and $\mathbf{v}_k = [v_k, v_{k-1}, \cdots, v_0]^T$. Since (14) has the same form as (1) and $\mathbf{n}_k$ has zero mean, the state vector $\mathbf{x}_k$ can be estimated by using (3). Specifically,

$$\hat{\mathbf{x}}_k = (\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k)^{-1}\mathbf{H}_k^T\mathbf{W}_k\mathbf{y}_k \quad (15)$$

where $\mathbf{W}_k$ is a $(k+1)$-by-$(k+1)$ weight matrix. The Kalman filter can be derived from this equation. Suppose for the time being that the weight matrix $\mathbf{W}_k$ in (15) is set equal to the inverse of the noise covariance matrix $E[\mathbf{n}_k\mathbf{n}_k^T]$, and that the following assumptions on the noise sequences in (7) and (8) are satisfied:

- $\{\mathbf{w}_k\}$ is a white noise with zero mean and $E[\mathbf{w}_k\mathbf{w}_l^T] = \mathbf{Q}_k\delta_{kl}$ where $\delta_{kl} = 0$ for $k \neq l$ and $\delta_{kk} = 1$.
- The initial state vector $\mathbf{x}_0$ is random with mean $\bar{\mathbf{x}}_0$ and covariance matrix $\mathbf{P}_0$.
- The random variables $\{\mathbf{w}_k, v_k, (\mathbf{x}_0 - \bar{\mathbf{x}}_0)\}$ are uncorrelated.

Then, (15) leads to the Kalman filter, which is summarized in Table 1 for later use.

The G-RLS algorithm is also derived from (15). Let the weight matrix $\mathbf{W}_k$ be:

$$\mathbf{W}_k = diag(1, \lambda_k, \prod_{m=1}^{2}\lambda_{k-m+1}, \cdots, \prod_{m=1}^{k}\lambda_{k-m+1}). \quad (16)$$

When $\lambda_1 = \lambda_2 = \cdots = \lambda_k = \lambda$, this weight matrix is identical to the one in (5). We rewrite (15) and introduce some relations

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= (\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k)^{-1}\mathbf{H}_k^T\mathbf{W}_k\mathbf{y}_k \\ &= \mathbf{P}_{k|k}\mathbf{H}_k^T\mathbf{W}_k\mathbf{y}_k \quad (17)\end{aligned}$$

where $\hat{\mathbf{x}}_k$ in (15) is denoted by $\hat{\mathbf{x}}_{k|k}$ emphasizing that the input data up to time k are given, and $\mathbf{P}_{k|k} = (\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k)^{-1}$.

The matrix $\mathbf{P}_{k|k}$ is *not* a covariance matrix, unlike the case of Kalman filtering.

To derive a recursive version of the estimate in (17), we consider

$$\begin{aligned} \mathbf{y}_{k-1} &= \mathbf{H}_{k-1}\mathbf{x}_{k-1} + \mathbf{n}_{k-1} \\ &= \mathbf{H}_{k|k-1}\mathbf{x}_k + \mathbf{n}_{k|k-1} \end{aligned} \tag{18}$$

where $\mathbf{H}_{k|k-1} = \mathbf{H}_{k-1}\mathbf{A}_{k-1}^{-1}$ and $\mathbf{n}_{k|k-1} = \mathbf{n}_{k-1} - \mathbf{H}_{k|k-1}\mathbf{w}_{k-1}$. Since (18) has the same form as (1) and $\mathbf{n}_{k|k-1}$ has zero mean, the estimate (or prediction) of $\hat{\mathbf{x}}_k$ given $\{y_0, \cdots, y_{k-1}\}$ can be expressed as

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= (\mathbf{H}_{k|k-1}^T \mathbf{W}_{k|k-1}\mathbf{H}_{k|k-1})^{-1}\mathbf{H}_{k|k-1}^T \mathbf{W}_{k|k-1}\mathbf{y}_{k-1} \\ &= \mathbf{P}_{k|k-1}\mathbf{H}_{k|k-1}^T \mathbf{W}_{k|k-1}\mathbf{y}_{k-1} \end{aligned} \tag{19}$$

where

$$\mathbf{P}_{k|k-1} = (\mathbf{H}_{k|k-1}^T \mathbf{W}_{k|k-1}\mathbf{H}_{k|k-1})^{-1} \tag{20}$$

and $\mathbf{W}_{k|k-1}$ is a weight matrix. A reasonable choice of the weight $\mathbf{W}_{k|k-1}$ for the one-step prediction is:

$$\mathbf{W}_{k|k-1} = \lambda_k \mathbf{W}_{k-1} \tag{21}$$

With this weight matrix, every input including the most recent one y_{k-1} can be properly weighted for one-step prediction. Our goal is to express $\hat{\mathbf{x}}_{k|k}$ in terms of $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{x}}_{k-1|k-1}$. Using (21) in (20), $\mathbf{P}_{k|k-1}$ is written as

$$\mathbf{P}_{k|k-1} = \frac{\mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T}{\lambda_k}. \tag{22}$$

Now from (22) and (17), it is straightforward to show that $\hat{\mathbf{x}}_{k|k-1}$ in (19) reduces to

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}. \tag{23}$$

The recursion formulas for $\mathbf{P}_{k|k}$ and $\hat{\mathbf{x}}_{k|k}$ are derived by exploiting the following relations:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{h}_k^T \\ \mathbf{H}_{k|k-1} \end{bmatrix}, \mathbf{W}_k = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{W}_{k|k-1} \end{bmatrix} \tag{24}$$

and $\mathbf{y}_k = [y_k,\ \mathbf{y}_{k-1}^T]^T$. The matrix $\mathbf{P}_{k|k}$ in (17) is expressed as

$$\begin{aligned} \mathbf{P}_{k|k} &= (\mathbf{H}_{k|k-1}^T \mathbf{W}_{k|k-1}\mathbf{H}_{k|k-1} + \mathbf{h}_k\mathbf{h}_k^T)^{-1} \\ &= \mathbf{P}_{k|k-1} - \frac{\mathbf{P}_{k|k-1}\mathbf{h}_k\mathbf{h}_k^T\mathbf{P}_{k|k-1}}{\mathbf{h}_k^T\mathbf{P}_{k|k-1}\mathbf{h}_k + 1} \end{aligned} \tag{25}$$

where the first equality follows from (24) and the second follows from the matrix inversion lemma $(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$. Now the desired estimate $\hat{\mathbf{x}}_{k|k}$ becomes

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= (\mathbf{P}_{k|k-1} - \frac{\mathbf{P}_{k|k-1}\mathbf{h}_k\mathbf{h}_k^T\mathbf{P}_{k|k-1}}{\mathbf{h}_k^T\mathbf{P}_{k|k-1}\mathbf{h}_k + 1}) \\ &\quad \cdot(\mathbf{H}_{k|k-1}^T \mathbf{W}_{k|k-1}\mathbf{y}_{k-1} + \mathbf{h}_k y_k) \end{aligned} \tag{26}$$

and after some calculation it reduces to

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(y_k - \mathbf{h}_k^T\hat{\mathbf{x}}_{k|k-1}) \tag{27}$$

Table 2: Summary of the G-RLS algorithm.

Initialize the algorithm by setting
$\hat{\mathbf{x}}_{0|-1} = \mathbf{0}$,
$\mathbf{P}_{0|-1} = p^{-1}\mathbf{I}$, p = small positive constant.
Prediction: $k = 1, 2, 3, \cdots$
$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}$
$\mathbf{P}_{k|k-1} = \frac{\mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T}{\lambda_k}$
Filtering: $k = 0, 1, 2, \cdots$
$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(y_k - \mathbf{h}_k^T\hat{\mathbf{x}}_{k|k-1})$
$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{h}_k^T)\mathbf{P}_{k|k-1}$
$\mathbf{K}_k = \frac{\mathbf{P}_{k|k-1}\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{P}_{k|k-1}\mathbf{h}_k + 1}$

where

$$\mathbf{K}_k = \frac{\mathbf{P}_{k|k-1}\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{P}_{k|k-1}\mathbf{h}_k + 1}. \tag{28}$$

Equations (22), (23), (25), (27) and (28) constitute the G-RLS algorithm, as summarized in Table 2. This algorithm is initiated by setting $\hat{\mathbf{x}}_{0|-1} = \mathbf{0}$ and $\mathbf{P}_{0|-1} = p^{-1}\mathbf{I}$ for p a small positive constant, as in the case of conventional RLS filtering [5]. When the state model in (7) is a random walk model having $\mathbf{A}_k = \mathbf{I}$ and $\lambda_k = \lambda$ for all k,

$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} \tag{29}$$

$$\mathbf{P}_{k|k-1} = \lambda^{-1}\mathbf{P}_{k-1|k-1} \tag{30}$$

and the G-RLS algorithm reduces to the conventional RLS algorithm.

Comparison of Table 1 and 2 clearly indicates that the G-RLS algorithm resembles Kalman filtering. The two algorithms become identical, with the exception of initial conditions, if the statistics $\mathbf{Q}_k$ and σ_k^2 that appear in Kalman filtering are set to $\mathbf{0}$ and 1, respectively and $\lambda_k = 1$ for every k. Therefore, the G-RLS algorithm without exponential weighting corresponds to the Kalman filter described by an unforced (or deterministic) state model expressed as

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}_k\hat{\mathbf{x}}_k. \tag{31}$$

The relation between the two algorithm may be summarized as follows: given the state model in (7) and (8), the G-RLS algorithm approximates the Kalman filter by employing a proper weight sequence $\{\lambda_k\}$.

3. APPLICATION TO DPLL DESIGN

Consider the 2nd-order adaptive DPLL model shown in Fig. 1. Assuming a zero-crossing DPLL as in [9], the input to this model is the timing offset between the positive going zero-crossings of the incoming signal and those of a locally generated sine wave. To be specific, let $1/T_0$ and $1/T_1$ denote the clock rates of the receiver and the transmitter, respectively. The timing offset α_k at the k-th ($k = 0, 1, 2, \cdots$) zero-crossing point can be expressed as

$$\alpha_k = t_0 + k(T_1 - T_0) \tag{32}$$

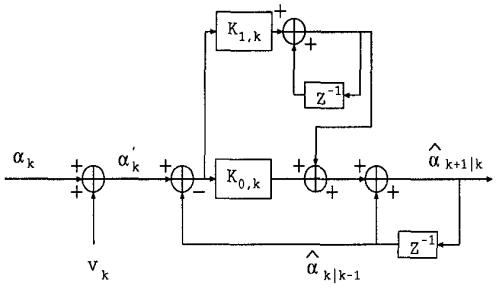

Figure 1: The 2nd-order DPLL.

where t_0 is the initial timing offset. The input $\alpha_k^{'}$ to the DPLL, corrupted by noise, is expressed as

$$\alpha_k^{'} = \alpha_k + v_k \tag{33}$$

where v_k is zero-mean white noise with variance σ^2. In [9], the variable gain sequences $K_{0,k}$ and $K_{1,k}$ are obtained based on the following state-space model.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \tag{34}$$

$$y_k = \alpha_k^{'} = \mathbf{h}^T\mathbf{x}_k + v_k \tag{35}$$

where $\mathbf{x}_k = [\alpha_k \ \beta_k]^T$; $\beta_k = T_1 - T_0$; $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{h}^T = [1 \ 0]$. Note that β_k is constant for all k. The optimal gain $K_{0,k}$ and $K_{1,k}$ minimizing the trace of the prediction error covariance matrix $\mathbf{P}_{k+1|k}$ are given by the Kalman gain vector $\mathbf{K}_k = [K_{0,k} \ K_{1,k}]^T$, which is obtained by solving the Kalman filtering equations in Table 1. Application of the G-RLS algorithm to this problem yields a sub-optimal solution. In what follows, the performance of the G-RLS algorithm is compared with those of Kalman filtering through computer simulation.

The input to the DPLL is generated under the assumption that the initial phase α_0 and the initial phase change β_0 have uniform density in $[-T_0/2, T_0/2]$ and $[-T_0/10, T_0/10]$, respectively. For these initial statistics, $\mathbf{P}_{0|-1}$ required for Kalman filtering is $\begin{bmatrix} T_0^2/12 & 0 \\ 0 & T_0^2/300 \end{bmatrix}$. The process noise covariance matrix $\mathbf{Q}$ is set to $\begin{bmatrix} 10^{-3}T_0^2 & 0 \\ 0 & 10^{-7}T_0^2 \end{bmatrix}$ and the measurement noise variance $\sigma^2 = 10^{-1}T_0^2$. For the G-RLS algorithm, these model statistics are ignored and its forgetting factor is experimentally chosen through simulation. Here, λ is set at 0.96.

The variable loop gain sequences obtained through the G-RLS and Kalman filtering algorithm are depicted in Fig. 2, and the corresponding prediction error variances evaluated experimentally through 1000 simulation runs are shown in Fig. 3. It is shown that the G-RLS algorithm almost acts like the Kalman filter; the former can be a useful alternative to the latter in practical applications in which model statistics are not precisely known.

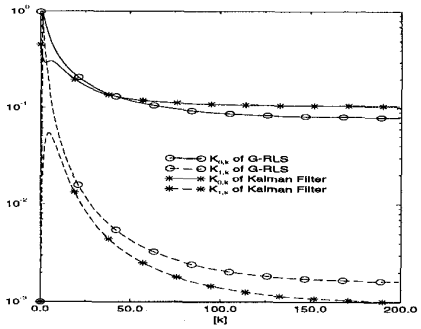

Figure 2: The variable loop gain sequences.

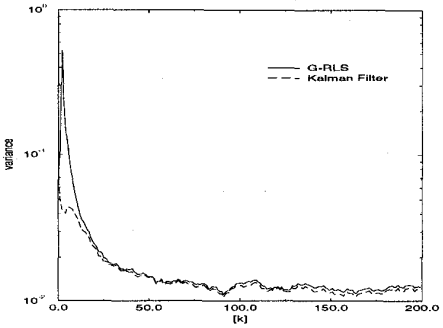

Figure 3: The empirical prediction error variances (normalized to T_0^2).

4. REFERENCES

[1] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis.* New York: Academic Press, 1977.

[2] L. Lyung and S. Gunnarsson, "Adaptation and Tracking in System Identification–A Survey," *Automatica*, vol. 26, pp. 7–21, Jan. 1990.

[3] S. Haykin, *Adaptive Filter Theory, 2nd ed.* Englewood Cliffs, N.J.: Prentice-Hall, 1991.

[4] A. H. Sayed and T. Kailath, "A State-Space Approach to Adaptive RLS Filtering," *IEEE Signal Processing Magazine*, vol. 11, pp. 18–60, Jul. 1994.

[5] S. Haykin, *Adaptive Filter Theory, 3rd ed.* Englewood Cliffs, N.J.: Prentice-Hall, 1996.

[6] S. Haykin, A. H. Sayed, J. R. Zeidler, P. Yee, and P. C. Wei, "Adaptive Tracking of Linear Time-Variant Systems by Extended RLS Algorithms," *IEEE Transactions on Signal Processing*, vol. 45, pp. 1118–1128, May. 1997.

[7] Y. Genin, "Further Comments on the Derivation of Kalman Filters, Section II: Gaussian Estimation and Kalman Filtering," *AGARDOgraph 139, C. T. Leodes, ed.*, pp. 55–63, 1970.

[8] J. L. Melsa and D. L. Cohn, *Decision and Estimation Theory.* McGraw-Hill, 1978.

[9] P. F. Driessen, "DPLL Bit Synchronizer with Rapid Acquisition Using Adaptive Kalman Filtering Techniques," *IEEE Transactions on Communications*, vol. 42, pp. 2673–2675, Sep. 1994.