49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Growing Window Recursive Quadratic Optimization with Variable Regularization

Asad A. Ali[1], Jesse B. Hoagg[2], Magnus Mossberg[3] and Dennis S. Bernstein[4]

***Abstract*— We present a growing-window variable-regularization recursive least squares (GW-VR-RLS) algorithm. Standard recursive least squares (RLS) uses a time-invariant regularization. More specifically, the inverse of the initial covariance matrix in classical RLS can be viewed as a regularization term, which weights the difference between the next state estimate and the initial state estimate. The present paper allows for time-varying in the weighting as well as what is being weighted. This extension can be used to modulate the speed of convergence of the estimates versus the magnitude of transient estimation errors. Furthermore, the regularization term can weight the difference between the next state estimate and a time-varying vector of parameters rather than the initial state estimate as is required in standard RLS.**

I. INTRODUCTION

Recursive least squares (RLS) is widely used in signal processing, identification, estimation, and control [1], [2], [3], [4], [5], [6], [7], [8]. Under ideal conditions, that is, nonnoisy measurements and persistency of the data, RLS is guaranteed to converge to the minimizer of a quadratic function [5], [6]. In practice, the accuracy of the estimates and the rate of convergence depend on the level of noise and persistency of the data. The goal of the present paper is to extend standard RLS in two ways. First, in standard RLS, the positive-definite initialization of the covariance matrix serves as the weighting of a regularization term within the context of a quadratic optimization. Until at least n measurements are available, this regularization term compensates for the lack of persistency in order to obtain a unique minimizer. Traditionally, the regularization weighting is fixed for all steps of the recursion. In the present work, we derive a growing-window variable-regularization RLS (GW-VR-RLS) algorithm, where the weighting of the regularization term changes at each step. As a special case, the regularization can be decreased in magnitude or rank as the rank of the covariance matrix increases, and can be removed entirely when no longer needed. This ability is not available in standard RLS where the regularization term is weighted by the inverse of the initial covariance at every step.

A second extension presented in this paper also involves the regularization term. Specifically, the regularization term in standard RLS weights the difference between the next state estimate and the initial state. In the present paper, the regularization term weights the difference between the next state estimate and an arbitrarily chosen time-varying vector of parameters. As a special case, the time-varying vector can be the current state estimate, and thus the regularization term weights the difference between the next state estimate and the current state estimate. This formulation allows us to modulate the rate at which the current estimate changes from step to step.

For these extensions, we derive GW-VR-RLS update equations. While standard RLS entails the update of the state estimate and the covariance matrix, GW-VR-RLS entails the update of an additional symmetric matrix of dimension $n \times n$ to allow for the variable regularization. Thus, GW-VR-RLS entails some additional computational burden relative to classical RLS.

II. PROBLEM FORMULATION AND GW-VR-RLS ALGORITHM

For all $i \geq 0$, let $A_i \in \mathbb{R}^{n\times n}$, $b_i, \alpha_i \in \mathbb{R}^n$, and $R_i \in \mathbb{R}^{n\times n}$, where A_i and R_i are positive semidefinite, define $A_0 \triangleq 0$, $b_0 \triangleq 0$, and assume that, for all $k \geq 0$, $\sum_{i=0}^k A_i + R_k$ and $\sum_{i=0}^k A_i + R_{k+1}$ are positive definite. Hence R_0 and R_1 are positive definite. For $k \geq 0$, define the regularized quadratic cost

$$J_k(x) \triangleq \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x\right) + (x-\alpha_k)^{\mathrm{T}} R_k (x-\alpha_k), \quad (1)$$

where $x \in \mathbb{R}^n$. The minimizer x_k of (1) is given by

$$x_k = -\tfrac{1}{2}\left(\sum_{i=0}^{k} A_i + R_k\right)^{-1}\left(\sum_{i=0}^{k} b_i - 2R_k\alpha_k\right). \quad (2)$$

Note that $x_0 = \alpha_0$ is the minimizer of $J_0(x)$.

To rewrite (2) recursively, define

$$P_k \triangleq \left(\sum_{i=0}^{k} A_i + R_k\right)^{-1}, \quad (3)$$

where the inverse exists by assumption, so that

$$x_k = -\tfrac{1}{2}P_k\left(\sum_{i=0}^{k} b_i - 2R_k\alpha_k\right). \quad (4)$$

[1]Graduate student, Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109-2140, asadali@umich.edu.

[2]Assistant Professor, Department of Mechanical Engineering, The University of Kentucky, Lexington, KY 40506-0503, jhoagg@engr.uky.edu.

[3]Associate Professor, Department of Physics and Electrical Engineering, Karlstad University, Sweden, Magnus.Mossberg@kau.se

[4]Professor, Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109-2140, dsbaero@umich.edu.

Using (4), it follows that

$$
\begin{aligned}
x_{k+1} &= -\tfrac{1}{2}P_{k+1}\left(\sum_{i=0}^{k+1} b_i - 2R_{k+1}\alpha_{k+1}\right)\\
&= -\tfrac{1}{2}P_{k+1}\left(\sum_{i=0}^{k} b_i + b_{k+1} - 2R_{k+1}\alpha_{k+1}\right)\\
&= -\tfrac{1}{2}P_{k+1}\Big(-2P_k^{-1}x_k + 2R_k\alpha_k + b_{k+1} - 2R_{k+1}\alpha_{k+1}\Big)\\
&= -P_{k+1}\Big(-P_{k+1}^{-1}x_k + R_{k+1}x_k - R_k x_k + A_{k+1}x_k\\
&\quad + R_k\alpha_k + \tfrac{1}{2}b_{k+1} - R_{k+1}\alpha_{k+1}\Big)\\
&= x_k - P_{k+1}\Big(A_{k+1}x_k + (R_{k+1}-R_k)x_k\\
&\quad + R_k\alpha_k - R_{k+1}\alpha_{k+1} + \tfrac{1}{2}b_{k+1}\Big). \qquad (5)
\end{aligned}
$$

Next, it follows from (3) that

$$
\begin{aligned}
P_{k+1}^{-1} &= \sum_{i=0}^{k} A_i + A_{k+1} + R_{k+1}\\
&= P_k^{-1} + R_{k+1} - R_k + A_{k+1}.
\end{aligned}
$$

Consider the decomposition

$$A_{k+1} = \psi_{k+1}\psi_{k+1}^{\mathrm{T}}, \qquad (6)$$

where $\psi_{k+1} \in \mathbb{R}^{n\times n_{k+1}}$ and $n_{k+1} \triangleq \operatorname{rank}(A_{k+1})$. Consequently,

$$
\begin{aligned}
P_{k+1} &= \left(P_k^{-1} + R_{k+1} - R_k + \psi_{k+1}\psi_{k+1}^{\mathrm{T}}\right)^{-1}\\
&= \left(Q_{k+1}^{-1} + \psi_{k+1}\psi_{k+1}^{\mathrm{T}}\right)^{-1}, \qquad (7)
\end{aligned}
$$

where

$$
\begin{aligned}
Q_{k+1} &\triangleq \left(\sum_{i=0}^{k} A_i + R_{k+1}\right)^{-1}\\
&= \left(P_k^{-1} + R_{k+1} - R_k\right)^{-1}, \qquad (8)
\end{aligned}
$$

where the inverse exists by assumption. Note that

$$P_{k+1} \le Q_{k+1}. \qquad (9)$$

Next, using the matrix inversion lemma

$$(X+UCV)^{-1} = X^{-1} - X^{-1}U\left(C^{-1} + VX^{-1}U\right)^{-1}VX^{-1} \qquad (10)$$

with $X = Q_{k+1}^{-1}$, $U = \psi_{k+1}$, $C = I$, and $V = \psi_{k+1}^{\mathrm{T}}$, it follows from (7) that

$$
\begin{aligned}
P_{k+1} &= Q_{k+1}\Big(I_n - \psi_{k+1}(I_{n_{k+1}} + \psi_{k+1}^{\mathrm{T}}Q_{k+1}\psi_{k+1})^{-1}\\
&\quad \times \psi_{k+1}^{\mathrm{T}}Q_{k+1}\Big). \qquad (11)
\end{aligned}
$$

Next, consider the decomposition

$$R_{k+1} - R_k = \phi_{k+1}S_{k+1}\phi_{k+1}^{\mathrm{T}}, \qquad (12)$$

where $\phi_{k+1} \in \mathbb{R}^{n\times m_{k+1}}$, $m_{k+1} \triangleq \operatorname{rank}(R_{k+1} - R_k)$, and $S_{k+1} \in \mathbb{R}^{m_{k+1}\times m_{k+1}}$ is a matrix of the form

$$
S_{k+1} \triangleq \begin{bmatrix} \pm 1 & 0 & \cdots & \\ 0 & \pm 1 & & \vdots \\ \vdots & & \ddots & \\ & \cdots & & \pm 1 \end{bmatrix}. \qquad (13)
$$

Therefore, (8) can be expressed as

$$Q_{k+1} = \left(P_k^{-1} + \phi_{k+1}S_{k+1}\phi_{k+1}^{\mathrm{T}}\right)^{-1}. \qquad (14)$$

Letting $X = P_k^{-1}$, $U = \phi_{k+1}$, $C = S_{k+1}$, and $V = \phi_{k+1}^{\mathrm{T}}$ it follows from (10) and (14) that

$$
\begin{aligned}
Q_{k+1} &= P_k - P_k\phi_{k+1}(S_{k+1} + \phi_{k+1}^{\mathrm{T}}P_k\phi_{k+1})^{-1}\phi_{k+1}^{\mathrm{T}}P_k\\
&= P_k\Big(I_n - \phi_{k+1}(S_{k+1} + \phi_{k+1}^{\mathrm{T}}P_k\phi_{k+1})^{-1}\phi_{k+1}^{\mathrm{T}}P_k\Big). \qquad (15)
\end{aligned}
$$

Therefore, for $k \ge 0$, the recursive regularized quadratic cost minimizer of (1) is given by (5), (11), and (15), that is,

$$
\begin{aligned}
Q_{k+1} &= P_k\Big(I_n - \phi_{k+1}(S_{k+1} + \phi_{k+1}^{\mathrm{T}}P_k\phi_{k+1})^{-1}\phi_{k+1}^{\mathrm{T}}P_k\Big), \qquad (16)\\
P_{k+1} &= Q_{k+1}\Big(I_n - \psi_{k+1}(I_{n_{k+1}} + \psi_{k+1}^{\mathrm{T}}Q_{k+1}\psi_{k+1})^{-1}\\
&\quad \times \psi_{k+1}^{\mathrm{T}}Q_{k+1}\Big), \qquad (17)\\
x_{k+1} &= x_k - P_{k+1}\Big(A_{k+1}x_k + (R_{k+1}-R_k)x_k + R_k\alpha_k\\
&\quad - R_{k+1}\alpha_{k+1} + \tfrac{1}{2}b_{k+1}\Big), \qquad (18)
\end{aligned}
$$

where $x_0 = \alpha_0$, $P_0 = R_0^{-1}$, ψ_{k+1} is given by (6), and ϕ_{k+1} is given by (12).

III. Specializations

A. Standard RLS

Consider the special case $R_k \equiv R_0$ and $\alpha_k \equiv x_0$. Then the quadratic cost

$$J_k(x) \triangleq \sum_{i=0}^{k}\left(x^{\mathrm{T}}A_i x + b_i^{\mathrm{T}}x\right) + (x - x_0)^{\mathrm{T}}R_0(x - x_0) \qquad (19)$$

is minimized by

$$
\begin{aligned}
x_{k+1} &= x_k - P_{k+1}\Big(A_{k+1}x_k + \tfrac{1}{2}b_{k+1}\Big), \qquad (20)\\
P_{k+1} &= P_k\Big(I_n - \psi_{k+1}(I_{n_{k+1}} + \psi_{k+1}^{\mathrm{T}}P_k\psi_{k+1})^{-1}\psi_{k+1}^{\mathrm{T}}P_k\Big), \qquad (21)
\end{aligned}
$$

where $P_0 = R_0^{-1}$. Since the recursive update for Q_{k+1} given by (16) simplifies to $Q_{k+1} = P_{k+1}$, standard RLS does not require the update of Q_{k+1}.

B. Standard RLS with $\alpha_k = x_{k-1}$ and $R_k \equiv R_0$

Consider the special case $R_k \equiv R_0$ and $\alpha_k = x_{k-1}$. Then the quadratic cost

$$J_k(x) \triangleq \sum_{i=0}^{k}\left(x^{\mathrm{T}}A_i x + b_i^{\mathrm{T}}x\right) + (x - x_{k-1})^{\mathrm{T}}R_0(x - x_{k-1})$$

is minimized by

$$
\begin{aligned}
P_{k+1} &= P_k\Big(I_{n_{k+1}} - \psi_{k+1}(I_{n_{k+1}} + \psi_{k+1}^{\mathrm{T}}P_k\psi_{k+1})^{-1}\psi_{k+1}^{\mathrm{T}}P_k\Big),\\
x_{k+1} &= x_k - P_{k+1}\Big(A_{k+1}x_k + P_0^{-1}(x_{k-1} - x_k) + \tfrac{1}{2}b_{k+1}\Big),
\end{aligned}
$$

where $P_0 = R_0^{-1}$. Note that the update for P_k does not require Q_k.

C. Standard RLS with forgetting factor

Let $0 < \lambda \leq 1$, and consider the modified cost

$$\bar{J}_k(x) \triangleq \sum_{i=0}^{k} \lambda^{k-i} \left(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x\right) + (x - x_0)^{\mathrm{T}} \lambda^k \bar{R}_0 (x - x_0),$$

where for $i \geq 0$, $\bar{A}_i = \bar{\psi}_i \bar{\psi}_i^{\mathrm{T}}$. Next, it follows that

$$\begin{aligned}\bar{J}_k(x) &= \lambda^k \sum_{i=0}^{k} \lambda^{-i} \left(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x\right) + (x - x_0)^{\mathrm{T}} \bar{R}_0 (x - x_0) \\ &= \lambda^k \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x\right) + (x - x_0)^{\mathrm{T}} R_0 (x - x_0),\end{aligned}$$

where $A_i \triangleq \lambda^{-i} \bar{A}_i$, $b_i \triangleq \lambda^{-i} \bar{b}_i$, and $R_0 \triangleq \bar{R}_0$. Therefore, $\bar{J}_k(x) = \lambda^k J_k(x)$, where $J_k(x)$ is given by the traditional RLS quadratic cost (19). Minimizing $\bar{J}_k(x)$ is equivalent to minimizing $J_k(x)$. In this case, the minimizer of J_k is given by (20) and (21); however, the minimizer x_k is expressed in terms of A_{k+1} and b_{k+1} rather than $\bar{A}_{k+1}$ and $\bar{b}_{k+1}$. Substituting $A_{k+1} = \lambda^{-k-1} \bar{A}_{k+1}$, $b_{k+1} = \lambda^{-k-1} \bar{b}_{k+1}$, and $\psi_{k+1} = \lambda^{-(k+1)/2} \bar{\psi}_{k+1}$ into (20) and (21) yields

$$\begin{aligned}P_{k+1} &= P_k \Big(I_n - \lambda^{-k-1} \bar{\psi}_{k+1} (I_{n_{k+1}} + \lambda^{-k-1} \bar{\psi}_{k+1}^{\mathrm{T}} \\ &\qquad \times P_k \bar{\psi}_{k+1})^{-1} \bar{\psi}_{k+1}^{\mathrm{T}} P_k \Big), \\ x_{k+1} &= x_k - P_{k+1} \Big(\lambda^{-k-1} \bar{A}_{k+1} x_k + \tfrac{1}{2} \lambda^{-k-1} \bar{b}_{k+1}\Big).\end{aligned}$$

Next, for $i \geq 0$, define $\bar{P}_i \triangleq \lambda^{-i} P_i$, and it follows that the minimizer of $\bar{J}_k$ is given by

$$\begin{aligned}\bar{P}_{k+1} &= \lambda^{-1} \bar{P}_k \Big(I_n - \bar{\psi}_{k+1} (\lambda I_{n_{k+1}} + \bar{\psi}_{k+1}^{\mathrm{T}} \bar{P}_k \bar{\psi}_{k+1})^{-1} \bar{\psi}_{k+1}^{\mathrm{T}} \bar{P}_k \Big), \\ x_{k+1} &= x_k - \bar{P}_{k+1} \Big(\bar{A}_{k+1} x_k + \tfrac{1}{2} \bar{b}_{k+1}\Big),\end{aligned}$$

where $\bar{P}_0 = R_0^{-1}$.

D. Standard RLS with $\alpha_k = x_{k-1}$ and forgetting factor

Let $0 < \lambda \leq 1$, and consider the modified cost

$$\begin{aligned}\bar{J}_k(x) \triangleq \sum_{i=0}^{k} \lambda^{k-i} \left(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x\right) + (x - x_{k-1})^{\mathrm{T}} \\ \times \lambda^k \bar{R}_0 (x - x_{k-1}),\end{aligned}$$

where for $i \geq 0$, $\bar{A}_i = \bar{\psi}_i \bar{\psi}_i^{\mathrm{T}}$. Next, it follows that

$$\begin{aligned}\bar{J}_k(x) &= \lambda^k \sum_{i=0}^{k} \lambda^{-i} \left(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x\right) + (x - x_{k-1})^{\mathrm{T}} \\ &\qquad \times \bar{R}_0 (x - x_{k-1}) \\ &= \lambda^k \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x\right) + (x - x_{k-1})^{\mathrm{T}} \\ &\qquad \times R_0 (x - x_{k-1}),\end{aligned} \tag{22}$$

where $A_i \triangleq \lambda^{-i} \bar{A}_i$, $b_i \triangleq \lambda^{-i} \bar{b}_i$, and $R_0 \triangleq \bar{R}_0$. Combining the steps in Section III-C and Section III-D, it follows that the minimizer of $\bar{J}_k$ is given by

$$\begin{aligned}\bar{P}_{k+1} &= \lambda^{-1} \bar{P}_k \Big(I_n - \bar{\psi}_{k+1} (\lambda I_{n_{k+1}} + \bar{\psi}_{k+1}^{\mathrm{T}} \bar{P}_k \bar{\psi}_{k+1})^{-1} \bar{\psi}_{k+1}^{\mathrm{T}} \bar{P}_k \Big), \\ x_{k+1} &= x_k - \bar{P}_{k+1} \Big(\bar{A}_{k+1} x_k + \lambda^{k+1} P_0^{-1} (x_{k-1} - x_k) + \tfrac{1}{2} \bar{b}_{k+1}\Big),\end{aligned}$$

where $P_0 = R_0^{-1}$.

IV. Setup for Numerical Simulations

For all $k \geq 0$, let $x_{k,\mathrm{opt}} \in \mathbb{R}^n$, $\psi_k \in \mathbb{R}^n$, where its i^{th} entry $\psi_{k,i}$ is generated from a zero mean, unit variance Gaussian distribution. The entries of ψ_k are independent. Define

$$\beta_k \triangleq \psi_k^{\mathrm{T}} x_{k,\mathrm{opt}}.$$

Let l be the number of data points. Define

$$\sigma_{\psi,i} \triangleq \sqrt{\frac{1}{l} \sum_{k=1}^{l} \psi_{k,i}^2} \xrightarrow{l \to \infty} 1,$$

$$\sigma_{\beta} \triangleq \sqrt{\frac{1}{l} \sum_{k=1}^{l} \beta_k^2} \xrightarrow{l \to \infty} \sqrt{x_{k,\mathrm{opt}}^{\mathrm{T}} x_{k,\mathrm{opt}}}.$$

Next, for $i = 1, \ldots, n$, let $N_{k,i} \in \mathbb{R}$, and $M_k \in \mathbb{R}$ be generated from zero-mean Gaussian distributions with variances $\sigma_{N,i}^2$ and σ_M^2, respectively, where $\sigma_{N,i}$ and σ_M are determined from the signal-to-noise ratio (SNR). More specifically, for $i = 1, \ldots, n$,

$$\mathrm{SNR}_{\psi,i} \triangleq \frac{\sigma_{\psi,i}}{\sigma_{N,i}}, \quad \text{and} \quad \mathrm{SNR}_{\beta} \triangleq \frac{\sigma_{\beta}}{\sigma_{\mathrm{M}}},$$

where, for $i = 1, \ldots, n$, $\sigma_{N,i} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} N_{k,i}^2}$ and $\sigma_M = \sqrt{\frac{1}{K} \sum_{k=1}^{K} M_k^2}$.

Finally, for $k \geq 0$, define $A_k \triangleq (\psi_k + N_k)(\psi_k + N_k)^{\mathrm{T}}$ and $b_k \triangleq -2(\beta_k + M_k)(\psi_k + N_k)$, where N_k is the noise in ψ_k and M_k is the noise in β_k.

Define

$$z_1 \triangleq \begin{bmatrix} 0.08 & -1.12 & 1.6 & 1.5 & -2.2 & -2.1 & 0.32 \end{bmatrix}^{\mathrm{T}},$$

$$z_2 \triangleq \begin{bmatrix} -1.11 & -0.2 & 1.1 & -0.2 & 0.4 & 0.23 & -2.5 \end{bmatrix}^{\mathrm{T}}.$$

Unless otherwise specified, for all $k \geq 0$, $x_{k,\mathrm{opt}} = z_1$, $\alpha_k = x_0$, and $x_0 = 0_{7\times1}$.

Define the performance

$$\varepsilon_k \triangleq \frac{\|x_{k,\mathrm{opt}} - x_k\|}{\|x_{k,\mathrm{opt}}\|}.$$

V. Numerical Simulations with Noiseless Data

We now investigate the effect of R_k, α_k and λ on GW-VR-RLS. Furthermore, in this section, A_k and b_k contain no noise, specifically, for all $k \geq 0$, $N_k = 0_{7\times1}$ and $M_k = 0$.

A. Effect of R_k

We begin by testing the effect of R_k on convergence of ε_k when R_k is constant. In the following example, we test GW-VR-RLS for three values of R_k. Specifically, for all $k \geq 0$, $R_k = I_{7\times7}$, $R_k = 0.1 I_{7\times7}$ or $R_k = 0.01 I_{7\times7}$. In all three cases, for all $k \geq 0$, A_k and b_k are the same. For this example, Figure 1 shows that smaller values of R_k yield faster convergence of ε_k to zero. This effect occurs because decreasing R_k reduces the magnitude of the regularization

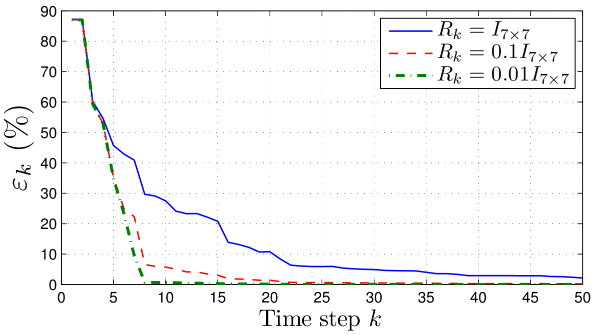

Fig. 1. Effect of R_k on convergence of x_k to $x_{k,\rm opt}$. For this example, smaller values of R_k yield faster convergence of ε_k to zero.

term in the cost function (1). Next, we let R_k be constant and positive definite until $\sum_{i=0}^{k-1} A_i$ has full rank, then we let $R_k = 0$. More specifically,

$$R_k = \begin{cases} 0.1I_{7\times7}, & \text{if rank } \sum_{i=0}^{k-1} A_i < n, \\ 0, & \text{if rank } \sum_{i=0}^{k-1} A_i = n. \end{cases} \tag{23}$$

For R_k given by (23), if there is no noise in the data, then x_k may converge to $x_{k,\rm opt}$ in finite time. In particular, if there exists a positive integer N such that $\sum_{i=0}^{N-1} A_i$ has full rank, then, for all $k \geq N$, $x_k = x_{k,\rm opt}$. Figure 2 shows that ε_k converges to zero in finite time when R_k is given by (23). In this case for all $k \geq 7$, $\sum_{i=0}^{k-1} A_i$ has full rank. Thus, for all $k \geq 8$, $x_k = x_{k,\rm opt}$.

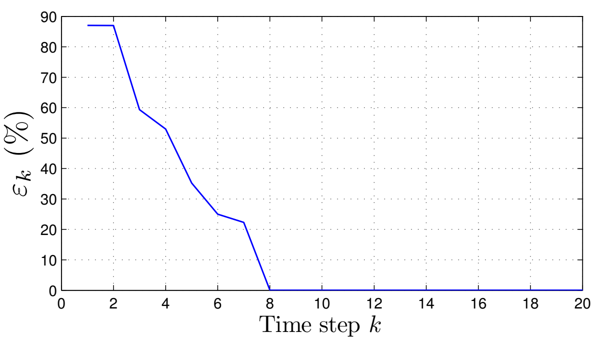

Fig. 2. Effect of R_k on convergence of x_k to $x_{k,\rm opt}$. In this example, $\sum_{i=0}^{7} A_i$ has full rank. Therefore, for $k \geq 8$, $R_k = 0$ and $x_k = x_{k,\rm opt}$.

Next, we choose the smallest R_k such that $\sum_{i=0}^{k} A_i$ is positive definite. More specifically, we conduct the singular value decomposition $USU^{\rm T} = \sum_{i=0}^{k-1} A_i$, where $U \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{n\times n}$ and has the form

$$S \triangleq \begin{bmatrix} \Gamma_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & 0_{(n-m)\times(n-m)} \end{bmatrix},$$

where $\Gamma \in \mathbb{R}^{m\times m}$ contains the m non-zero singular values of $\sum_{i=0}^{k-1} A_i$. Note that the singular value decomposition has the form $USU^{\rm T}$ because $\sum_{i=0}^{k-1} A_i$ is symmetric [9, Corollary 5.4.5]. Next, define

$$\hat{S} \triangleq \begin{bmatrix} 0_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & \epsilon I_{(n-m)\times(n-m)} \end{bmatrix},$$

where $\epsilon \geq 0$. Finally,

$$R_k \triangleq \begin{cases} R_0, & k = 0 \\ U\hat{S}U^{\rm T}, & \text{if rank } \sum_{i=0}^{k-1} A_i < n, \\ 0, & \text{if rank } \sum_{i=0}^{k-1} A_i = n, \end{cases} \tag{24}$$

In the following example we compare GW-VR-RLS with $R_k = I_{3\times3}$ and R_k given by (24) with $\epsilon = 1$. In both cases, for all $k \geq 0$, A_k and b_k are the same. For this example, Figure 3 shows that setting R_k given by (24) with $\epsilon = 1$ yields faster convergence of ε_k to zero than setting $R_k = I_{7\times7}$.

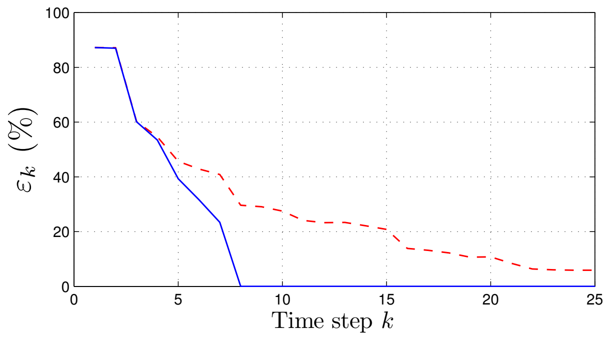

Fig. 3. Effect of R_k on convergence of x_k to $x_{k,\rm opt}$. The solid line denotes ε_k with R_k given by (24) and the dashed line denotes ε_k with $R_k = I_{7\times7}$. For this example, setting R_k given by (24) with $\epsilon = 1$ yields faster convergence of ε_k to zero than setting $R_k = I_{7\times7}$.

B. Effect of α_k

Figure 4 compares GW-VR-RLS with $\alpha_k = x_{k-1}$ and $\alpha_k = x_0$, where, for all $k \geq 0$, $R_k = I_{7\times7}$. For this example, setting $\alpha_k = x_{k-1}$ yields faster convergence of ε_k to zero than setting $\alpha_k = x_0$.

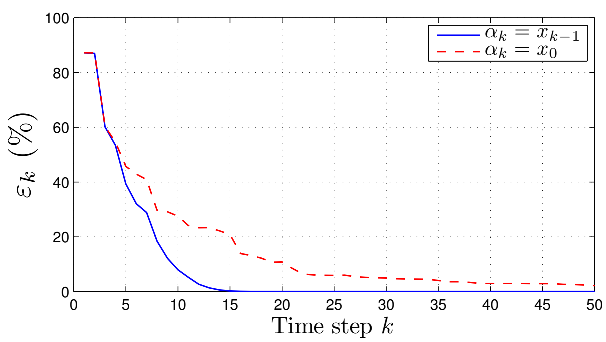

Fig. 4. Effect of one step regularization on convergence of x_k to $x_{k,\rm opt}$. For this example, setting $\alpha_k = x_{k-1}$ yields faster convergence of ε_k to zero than setting $\alpha_k = x_0$

C. Effect of Forgetting Factor

In this section, we examine standard RLS with forgetting factor (as described in Section III-C). In the following example, we consider three values of λ, specifically λ=1, λ=0.995 or λ=0.9. For all $k \geq 0$, $R_k = 0.1I_{7\times7}$ and

$$x_{k,\rm opt} = \begin{cases} z_1, & 0 \leq k \leq 200 \\ z_2, & k > 200 \end{cases}$$

For this example, Figure 5 shows that, for $k \leq 200$, the forgetting factor has negligible impact on the behavior of ε_k. For $k > 200$, smaller values of λ yield faster convergence of ε_k to zero.

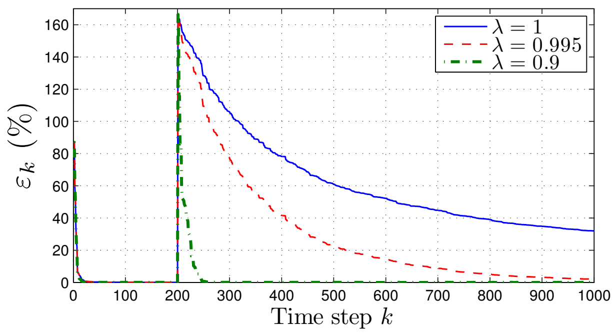

Fig. 5. Effect of forgetting factor on convergence of x_k to $x_{k,\rm opt}$. For $k \le 200$, the forgetting factor has negligible impact on the behavior of ε_k. For $k > 200$, $x_{k,\rm opt} \ne x_{200,\rm opt}$, and a smaller value of λ yields faster convergence of x_k to $x_{k,\rm opt}$.

VI. Numerical Simulations with Noisy Data

We now investigate the effect of R_k, α_k, and λ on GW-VR-RLS when the data have noise. More specifically, for all $k \ge 0$, M_k and $N_{k,i}$ are generated from zero mean Gaussian distributions with variances depending on $\mathrm{SNR}_{\psi,\rm i}$ and SNR_β, respectively. Figure 6 shows the effect of noise on standard RLS for different SNR values. In this example, a smaller value of SNR yields a larger asymptotic value of ε_k.

In the next example, we examine the convergence of ε_k for standard RLS when ψ_k and β_k have constant bias. We consider three cases of constant bias, specifically, for all $k \ge 0$, $N_k = (0.2)1_{7\times1}$ and $M_k = 0.2$, $N_k = (0.2)1_{7\times1}$ and $M_k = 0$ or $N_k = 0_{7\times1}$ and $M_k = 0.2$. For this example, Figure 7 shows that bias increases the asymptotic value of ε_k. Furthermore, bias in β_k yields a higher asymptotic value of ε_k than an equal percent of bias in ψ_k.

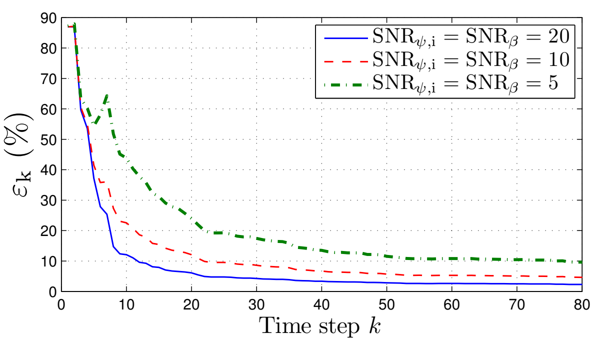

Fig. 6. Effect of noise on standard RLS. In this example, smaller values of SNR yield larger asymptotic values of ε_k.

A. Effect of R_k

In this section, we examine the effect of R_k where R_k is constant. In the following example, we test GW-VR-RLS for three different values of R_k. Specifically, for all $k \ge 0$, $R_k = I_{7\times7}$, $R_k = 0.1I_{7\times7}$ or $R_k = 0.01I_{7\times7}$. Furthermore, $\mathrm{SNR}_{\psi,\rm i} = \mathrm{SNR}_\beta = 5$ and, for all $k \ge 0$, A_k and b_k are the same. For this example, Figure 8 shows that smaller values of R_k can result in larger peak values of ε_k.

Recall that, Figure 1 showed that smaller values of R_k can yield faster convergence of ε_k to zero. However, if the data have noise, then Figure 8 shows that the transient response

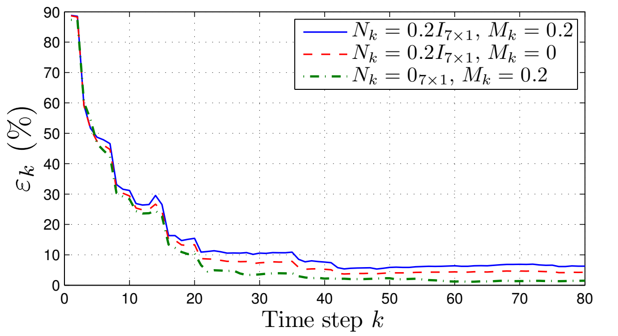

Fig. 7. Effect of bias on standard RLS. For this example, bias increases the asymptotic value of ε_k. Furthermore, bias in β_k yields a higher asymptotic value of ε_k than an equal percent of bias in ψ_k.

of ε_k can be worse for smaller values of R_k. As the SNR increases, Figure 8 converges to Figure 1.

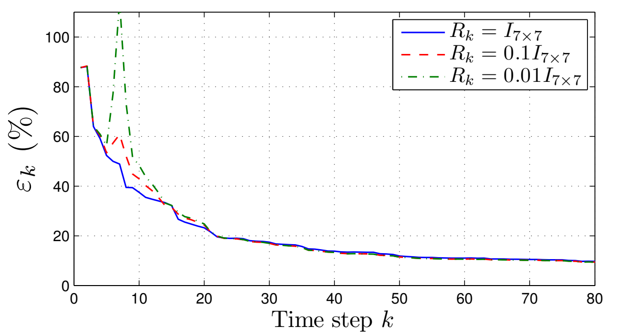

Fig. 8. Effect of R_k on convergence of x_k to $x_{k,\rm opt}$. For this example, smaller values of R_k can result in larger peak values of ε_k.

B. Effect of α_k

Figure 9 compares GW-VR-RLS with $\alpha_k = x_{k-1}$ and $\alpha_k = x_0$, where, $\mathrm{SNR}_{\psi,\rm i} = \mathrm{SNR}_\beta = 5$ and for all $k \ge 0$ $R_k = 0.1I_{7\times7}$. For this example, Figure 9 shows that the transient response of ε_k can be worse for $\alpha_k = x_{k-1}$ than it is for $\alpha_k = x_0$.

Figure 4 showed that setting $\alpha_k = x_{k-1}$ can yield faster convergence of ε_k to zero than setting $\alpha_k = x_0$. However, if the data have noise, then Figure 9 shows that the transient response of ε_k can be worse with $\alpha_k = x_{k-1}$ than it is with $\alpha_k = x_0$. As the SNR increases, Figure 9 converges to Figure 4.

Next, we compare GW-VR-RLS for different choices of α_k. More specifically, we let $\alpha_k = L_\nu(k)$ where

$$L_\nu(k) \triangleq \begin{cases} x_{k-1}, & 0 < k \le \nu, \\ x_{k-\nu}, & k > \nu, \end{cases}$$

where ν is a positive integer. In the following example, we test GW-VR-RLS for three different ν. Specifically, $\nu = 1$, $\nu = 5$, $\nu = 10$. In all cases, for all $k \ge 0$, A_k and b_k are the same, $R_k = I_{7\times7}$ and $\mathrm{SNR}_\beta = \mathrm{SNR}_{\psi,\rm i} = 5$. For this example, Figure 10 shows that larger values of ν can yield better transient performance of ε_k.

Next, we let $\alpha_k = W_\rho(k)$ where

$$W_\rho(k) \triangleq \begin{cases} x_0, & k = 1, \\ \frac{1}{k-1}\sum_{i=1}^{k-1} x_{k-i}, & 1 < k \le \rho, \\ \frac{1}{\rho}\sum_{i=1}^{\rho} x_{k-i}, & k > \rho, \end{cases}$$

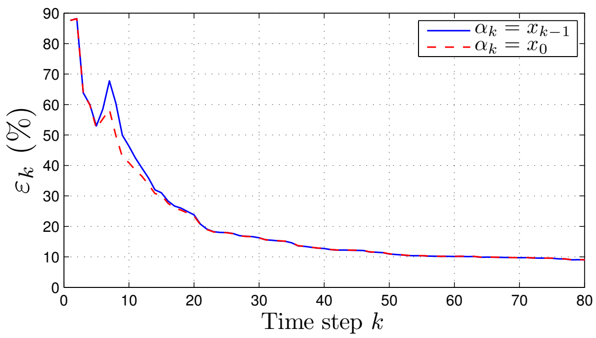

Fig. 9. Effect of α_k on convergence x_k to $x_{k,\rm opt}$. For this example, this figure shows that the transient response of ε_k can be worse for $\alpha_k = x_{k-1}$ than it is for $\alpha_k = x_0$.

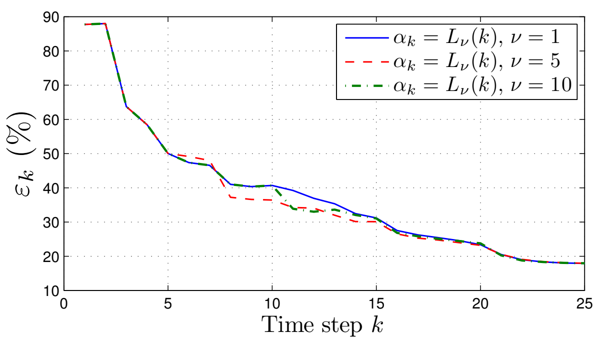

Fig. 10. Convergence of x_k to $x_{k,\rm opt}$. For this example, larger values of ν yield better transient performance of ε_k.

where ρ is a positive integer. In the following example, we test GW-VR-RLS for three different values of ρ. Specifically, $\rho = 1$, $\rho = 5$, $\rho = 10$. In all cases, for all $k \geq 0$, A_k and b_k are the same, $R_k = I_{7\times 7}$ and $\text{SNR}_\beta = \text{SNR}_{\psi,\rm i} = 5$. For this example, Figure 11 shows that larger values of ρ can yield better transient performance of ε_k than smaller values of ρ.

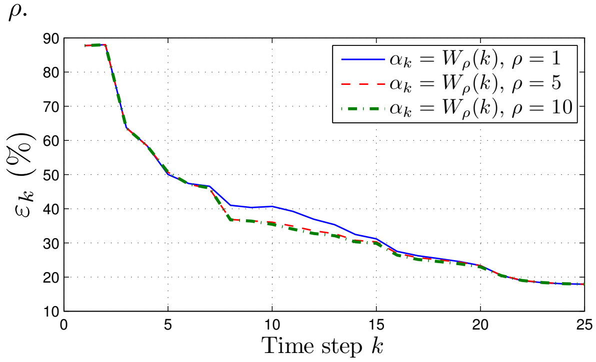

Fig. 11. Convergence of x_k to $x_{k,\rm opt}$. In this example, larger values of ρ yield better transient performance of ε_k than smaller values of ρ.

C. Effect of Forgetting Factor

In this section, we examine standard RLS with forgetting factor. In the following example, we test RLS for three values of λ, specifically λ=1, λ=0.95 or λ=0.9. Let $\text{SNR}_{\psi,\rm i} = \text{SNR}_\beta = 5$, and, for all $k \geq 0$, $R_k = 0.1 I_{7\times 7}$. For this example, Figure 12 shows that smaller values of λ yield larger asymptotic value of ε_k.

Next, we let

$$x_{k,\rm opt} = \begin{cases} z_1, & 0 \leq k \leq 500, \\ z_2, & k > 500, \end{cases}$$

For this example, Figure 13 shows that, for $k \leq 500$, smaller values of λ yield larger asymptotic values of ε_k. For $k > 500$, $x_{k,\rm opt} \neq x_{500,\rm opt}$, and a smaller value of λ yields faster convergence of ε_k to its asymptotic value.

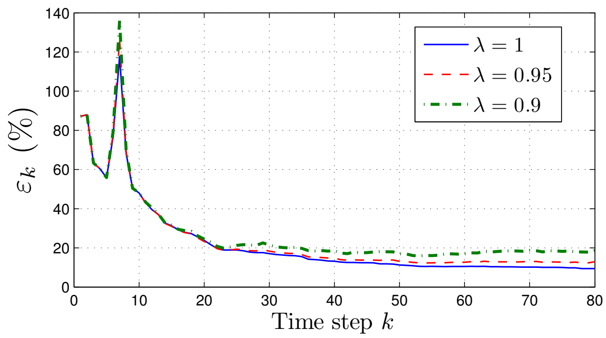

Fig. 12. Effect of forgetting factor on convergence of x_k to $x_{k,\rm opt}$. For this example smaller values of λ yield larger asymptotic values of ε_k.

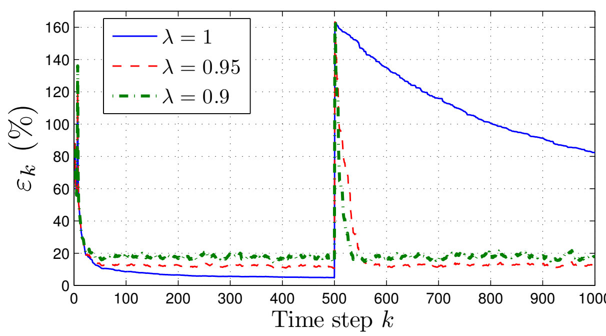

Fig. 13. Effect of forgetting factor on convergence of x_k to $x_{k,\rm opt}$. For $k \leq 500$, smaller values of λ yield larger asymptotic values of ε_k. For $k > 500$, $x_{k,\rm opt} \neq x_{500,\rm opt}$, and a smaller value of λ yields faster convergence of ε_k to its asymptotic value.

VII. Conclusions

In this paper, we presented a growing-window variable-regularization recursive least squares (GW-VR-RLS) algorithm. This algorithm allows for a time-varying regularization term in the RLS cost function. More specifically, GW-VR-RLS allows us to vary both the weighting in the regularization as well as what is being weighted, that is, the regularization term can weight the difference between the next state estimate and a time-varying vector of parameters rather than the initial state estimate. Future work will include an investigation of the convergence properties.

References

[1] A. H. Sayed, *Adaptive Filters*. Hoboken, New Jersey: John Wiley and Sons, Inc., 2008.

[2] J. N. Juang, *Applied System Identification*. Prentice-Hall, 1993.

[3] L. Ljung and T. Söderström, *Theory and practice of Recursive Identification*. Cambridge, MA: The MIT Press, 1983.

[4] L. Ljung, *System Identification: Theory for the User, 2nd ed.* Upper Saddle River, NJ: Prentice-Hall, 1999.

[5] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Addison-Wesley, 1995.

[6] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control*. Prentice Hall, 1984.

[7] G. Tao, *Adaptive Control Design and Analysis*. Wiley, 2003.

[8] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[9] D. S. Bernstein, *Matrix Mathematics*, 2nd ed. Princeton University Press, 2009.