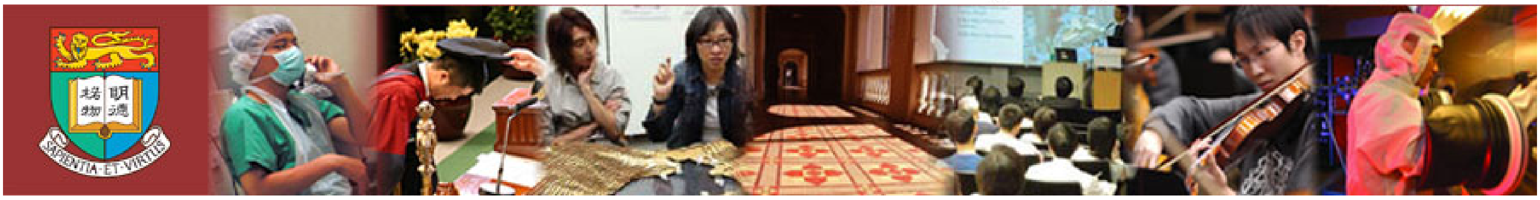

Title	A New State-Regularized QRRLS Algorithm with Variable Forgetting Factor
Author(s)	Chan, SC; Chu, YJ
Citation	IEEE Transactions on Circuits and Systems Part 2: Express Briefs, 2012, v. 59 n. 3, p. 183-187
Issued Date	2012
URL	http://hdl.handle.net/10722/189091
Rights	IEEE Transactions on Circuits and Systems Part 2: Express Briefs. Copyright © IEEE

A New State-Regularized QRRLS Algorithm With a Variable Forgetting Factor

S. C. Chan and Y. J. Chu

***Abstract*—This brief proposes a new state-regularized (SR) and QR-decomposition-based (QRD) recursive least squares (RLS) adaptive filtering algorithm with a variable forgetting factor (VFF). It employs the estimated coefficients as prior information to minimize the exponentially weighted observation error, which leads to reduced variance over a conventional RLS algorithm and reduced bias over an L_2-regularized RLS algorithm. To improve the tracking performance, a new measure of convergence status is introduced in controlling the forgetting factor. Consequently, the resultant SR-VFF-RLS algorithm stabilizes the update and adaptively selects the number of measurements by means of the VFF. Improved tracking performance, steady-state mean-square error, and robustness to power-varying inputs over conventional RLS algorithms can be achieved. Furthermore, the proposed algorithm can be implemented using QRD, which leads to a lower roundoff error and more efficient hardware realization than the direct implementation. The effectiveness of the proposed algorithm is demonstrated by computer simulations.**

***Index Terms*—Adaptive filters, QR decomposition (QRD), recursive least squares (RLS), variable regularization, variable forgetting factor (VFF).**

I. Introduction

THE recursive least squares (RLS) algorithm [1] is an effective adaptive filtering algorithm that has been widely used in various applications, such as system identification, interference suppression, and adaptive echo cancelation (AEC). Compared with other adaptive filtering algorithms such as the least-mean-square-based (LMS) algorithms [1], [2], the RLS algorithms usually have a faster convergence speed. However, an inherent problem of RLS-like algorithms is that the covariance matrix of input signals may become poorly conditioned or even singular if the input signal is not persistently exciting [2]. This is often encountered in acoustic applications such as AEC, where the level of the excitation signal usually varies significantly over time. In such situations, the estimation error variance of the RLS algorithm will increase considerably or even become unstable. This is even more critical if a small forgetting factor (FF) is used in variable FF (VFF)-RLS algorithms at a nonstationary environment. To address this problem, a regularization technique [3]–[8], which is a useful tool for reducing the estimation error variance, is usually incorporated into these algorithms. L_2 regularization is a widely used technique in RLS algorithms. Recently, the performance of the weighted L_2-based RLS algorithm has been analyzed theoretically in [8]. L_1 regularization, on the other hand, tends to produce sparse solutions as in [9] and [10].

For both L_1 and L_2 regularizations, the estimated parameters are penalized by the regularization term, leading to an estimate that is biased toward zero. The undesirable bias can be asymptotically suppressed to zero by using the smoothly clipped absolute deviation (SCAD) [11] regularization. However, these regularization methods usually assume that the parameters to be estimated possess sparsity, which may not be applied directly and require additional sparsity-enhancing transformations [10]. In this brief, a new state-based regularization method for RLS algorithms is proposed. The proposed state regularization differs from these previous studies in that it employs the current estimated coefficients as prior information and minimizes the observation errors as in the conventional RLS algorithm. The concept is intimately connected to the Kalman filter and the LMS algorithm, except that an infinite number of measurements, as in the RLS algorithm, with variable weighting is employed. Simulation results show that it has a better steady-state MSE than L_2 regularization due to the reduced bias. For a sparse coefficient vector or sparse coefficient changes, sparsity-enhancing transformation and SCAD proposed in [11] can also be incorporated to achieve better performance. Another contribution of this brief is to improve the tracking performance of this state-regularized (SR) RLS algorithm by introducing a new VFF scheme using a new measure of convergence status and the approach in [17] to vary the FF. This resultant SR-VFF-RLS algorithm stabilizes the update using the previous estimated filter coefficients and adaptively selects the number of measurements used by means of the VFF. Therefore, improved tracking performance, steady-state MSE, and robustness to power-varying inputs over the conventional RLS can be achieved. Furthermore, the proposed SR-VFF-RLS algorithm can be implemented using the QR decomposition (QRD) structure that leads to a lower roundoff error and more efficient hardware realization than the direct implementation. The effectiveness of the proposed algorithm is demonstrated by computer simulations and comparison with conventional RLS and gradient-based VFF-RLS (GVFF-RLS) algorithms [12].

II. System Model and the QRRLS Algorithm

Consider the adaptive system identification problem where an input signal $x(n)$ is applied simultaneously to an L-order adaptive transversal filter with weight vector $\boldsymbol{w}(n) = [w_1(n), w_2(n), \ldots, w_L(n)]^T$ and an unknown system to be identified with an impulse response $\boldsymbol{w}_0(n) = [w_{0\_1}(n), w_{0\_2}(n), \ldots, w_{0\_L}(n)]^T$. Let $\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T$

Manuscript received April 15, 2011; revised July 19, 2011, September 20, 2011, and November 14, 2011; accepted January 9, 2012. Date of publication February 3, 2012; date of current version March 16, 2012. This paper was recommended by Associate Editor P. K. Meher.

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong (e-mail: scchan@eee.hku.hk; yjchu@eee.hku.hk.).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2012.2184374

be the input vector. Then, the output of the adaptive filter is $y(n) = \boldsymbol{x}^T(n)\boldsymbol{w}(n)$. The measured output is used as the desired signal $d(n)$ of the adaptive filter as follows:

$$d(n) = \boldsymbol{x}^T(n)\boldsymbol{w}_0(n) + \eta(n) \tag{1}$$

where $\eta(n)$ denotes the additive noise or modeling error and the superscript ^{T} denotes matrix transposition. The adaptive filter aims to minimize error measurement of the estimation error $e(n) = d(n) - y(n)$. In RLS algorithms, the cost function is minimized as follows:

$$J(n) = \sum_{i=0}^{n} \lambda_{n-i}(n)e^2(i) \tag{2}$$

where $\lambda_{n-i}(n) = \lambda(n)\lambda_{n-i-1}(n-1),\ 0 \le i < n$ serves the purpose of an exponential window that puts less weight to errors at distant past. Here, $\lambda_0(n) = 1$ and $\lambda(n)$ is the FF used at the time index n, which usually satisfies $0 < \lambda(n) < 1$. For example, $\lambda_{n-i}(n)$ can be chosen as λ^{n-i}, where λ is a constant in a conventional RLS algorithm or is updated adaptively, as in VFF-RLS algorithms. We only discuss RLS algorithms with real input in this brief, although it can be extended to complex systems. By setting the first partial derivative of $J(n)$ with respect to $\boldsymbol{w}(n)$ to zero, one finds that the optimal weight vector satisfies the following normal equation:

$$\boldsymbol{R}_{XX}(n)\boldsymbol{w}_{\text{opt}}(n) = \boldsymbol{p}_X(n) \tag{3}$$

where $\boldsymbol{R}_{XX}(n) = \sum_{i=0}^{n} \lambda_{n-i}(n)\boldsymbol{x}(i)\boldsymbol{x}^T(i)$ and $\boldsymbol{p}_X(n) = \sum_{i=0}^{n} \lambda_{n-i}(n)d(i)\boldsymbol{x}(i)$ are the estimated covariance matrix of $\boldsymbol{x}(n)$, which is of zero mean, and the estimated cross-correlation vector of $d(n)$ and $\boldsymbol{x}(n)$, respectively. Applying the matrix inversion lemma to (3), the following RLS algorithm can be obtained [1], [8]:

$$\boldsymbol{P}(n) = \lambda^{-1}(n)\left(\boldsymbol{I} - \boldsymbol{k}(n)\boldsymbol{x}^T(n)\right)\boldsymbol{P}(n-1) \tag{4a}$$

$$\boldsymbol{k}(n) = \frac{\boldsymbol{P}(n-1)\boldsymbol{x}(n)}{\lambda(n) + \boldsymbol{x}^T(n)\boldsymbol{P}(n-1)\boldsymbol{x}(n)} \tag{4b}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \left(d(n) - \boldsymbol{w}^T(n)\boldsymbol{x}(n)\right)\boldsymbol{k}(n) \tag{4c}$$

where $\boldsymbol{P}(n)$ is the recursive update of $\boldsymbol{R}_{XX}^{-1}(n)$ and $\boldsymbol{I}$ is an identity matrix with an appropriate dimension. Equation (4) can also be efficiently implemented using a QR-based algorithm [8], as summarized in Table I, with the first update only. This QRRLS algorithm is mathematically equivalent to but has higher numerical stability than the direct implementation. The arithmetic complexity is of order $O(L^2)$.

III. SR-VFF-QRRLS Algorithm

A. The SR-QRRLS Algorithm

In some acoustic and related applications, the input to the adaptive filter, such as speech signals, may not be persistently exciting when the input signal level is very low. Consequently, the covariance matrix $\boldsymbol{R}_{XX}(n)$ may be ill-conditioned, and a large estimation error variance will result. To address this problem, a regularization term on the adaptive filter coefficients, i.e., $\kappa(n)\|\boldsymbol{w}(n)\|_2^2$, is usually imposed on the objective function (2) to limit the variation in $\boldsymbol{w}(n)$. The solution, instead of (3), will be modified to

$$\left(\boldsymbol{R}_{XX}(n) + \kappa(n)\boldsymbol{D}\right)\boldsymbol{w}(n) = \boldsymbol{p}_X(n) \tag{5}$$

TABLE I
The R-QRRLS/VR-QRRLS Algorithm

Initialization:

$\boldsymbol{R}(0) = \sqrt{\delta}\boldsymbol{I}$, δ is a small positive constant;

$\boldsymbol{u}(0) = \boldsymbol{0}$, $\boldsymbol{w}(0) = \boldsymbol{0}$ are null vectors.

Recursion:

Given $\boldsymbol{R}(n-1)$, $\boldsymbol{u}(n-1)$, $\boldsymbol{w}(n-1)$, $\boldsymbol{x}(n)$ and $d(n)$, compute at time n:

(i). The first update:

$$\begin{bmatrix} \boldsymbol{R}^{(1)}(n) & \boldsymbol{u}^{(1)}(n) \\ \boldsymbol{0}^T & c^{(1)}(n) \end{bmatrix} = \boldsymbol{Q}^{(1)}(n) \begin{bmatrix} \sqrt{\lambda(n)}\boldsymbol{R}(n-1) & \sqrt{\lambda(n)}\boldsymbol{u}(n-1) \\ \boldsymbol{x}^T(n) & d(n) \end{bmatrix}$$

The second update (regularized RLS algorithms only):

a) For the L_2 regularization

$$\begin{bmatrix} \boldsymbol{R}(n) & \boldsymbol{u}(n) \\ \boldsymbol{0}^T & c(n) \end{bmatrix} = \boldsymbol{Q}(n) \begin{bmatrix} \boldsymbol{R}^{(1)}(n) & \boldsymbol{u}^{(1)}(n) \\ \sqrt{\mu(n)}\boldsymbol{d}_l & 0 \end{bmatrix},$$

b) For the state-based regularization

$$\begin{bmatrix} \boldsymbol{R}(n) & \boldsymbol{u}(n) \\ \boldsymbol{0}^T & c(n) \end{bmatrix} = \boldsymbol{Q}(n) \begin{bmatrix} \boldsymbol{R}^{(1)}(n) & \boldsymbol{u}^{(1)}(n) \\ \sqrt{\mu(n)}\boldsymbol{d}_l & \sqrt{\mu(n)}w_l(n-1) \end{bmatrix}$$

where $\boldsymbol{Q}^{(1)}(n)$ and $\boldsymbol{Q}(n)$ are calculated by Givens rotation to obtain the left hand side of each equation above and $\mu(n)$=$\kappa(n)L$. For the QRRLS algorithm, $\boldsymbol{R}^{(1)}(n)$= $\boldsymbol{R}(n)$.

(ii). $\boldsymbol{w}(n) = \boldsymbol{R}^{-1}(n)\boldsymbol{u}(n)$ (back-substitution).

where $\kappa(n)$ is a possibly variable regularization parameter and $\boldsymbol{D}$ is a positive definite matrix. We note that the rank-1 update of $\boldsymbol{R}_{XX}(n) = \boldsymbol{R}_{XX}(n-1) + \boldsymbol{x}(n)\boldsymbol{x}^T(n)$ in the conventional QRRLS algorithm can be efficiently implemented by updating the Cholesky factor $\boldsymbol{R}(n)$ of $\boldsymbol{R}_{XX}(n)$ recursively using the QRD, as shown in recursion (i) of Table I. The update of the term $\kappa(n)\boldsymbol{D}$ in (5) is however complicated since it is of full rank. In [8], an L_2-regularized QR recursive least M-estimate algorithm was proposed. The idea is to apply the regularization successively using QRD. For the LS case considered here, the robust weighting in [8] can be chosen as unity, and it gives a QRD implementation, as shown in Table I [i.(a)], for approximating the L_2 regularization. It can be seen that the QRD is executed once for the data vector $[\boldsymbol{x}^T(n), d(n)]$ and once for the regularization vector $[\sqrt{\mu(n)}\boldsymbol{d}_l, 0]$ at each time instant, where $\boldsymbol{d}_l$ is the lth row of the regularization matrix $\boldsymbol{D}$ and $\mu(n) = \kappa(n)L$. If the vector is sequentially applied, then $l = (n \bmod L) + 1$. Therefore, the complexity is twice that of the RLS algorithm. Another solution for introducing regularization to the least squares lattice algorithm was also proposed in [7], where a vector composed of a shift-invariant signal was introduced as additional inputs. If the regularization parameter $\kappa(n)$ is made variable at each iteration, this yields the variable L_2-regularized QRRLS algorithm. In [8], the following regularization parameter $\kappa(n)$ was proposed to balance between bias and variance errors for white input:

$$\kappa(n) = \overline{\sigma}_x^2\lambda^{-1}(n)\sqrt{1-\lambda(n)}\sqrt{\gamma\left(\sigma_\eta^2/\sigma_x^2(n)\right)/\|\boldsymbol{w}_0\|_2^2} \tag{6}$$

where $\gamma = 1/((1/L)(2 + ((1-\lambda(n))L/\lambda(n))) + (1-\lambda(n))L/\lambda^2(n))$, $\overline{\sigma}_x^2$ is the averaged input power over the whole duration, whereas $\sigma_x^2(n)$ is the short-term averaged input power, which can be estimated by using a FF; σ_η^2 is the noise variance; and $\|\boldsymbol{w}_0\|_2^2$ is the squared norm of the system channel, which is usually assumed to be known *a priori* [3]. For instance, in acoustic applications, $\|\boldsymbol{w}_0\|_2^2$ can be estimated by, for example, Sabin's reverberation formulas [3]. By using L_2 regularization, the ill-conditioned problem can be improved significantly. However, it is shown from (5) that L_2 regularization introduces a bias to the true solution, particularly when a large regularization parameter is used. To solve this problem, let us rewrite (3) as

$$\left(\boldsymbol{R}_{XX}(n) + \kappa(n)\boldsymbol{I}\right)\boldsymbol{w}(n) = \boldsymbol{p}_X(n) + \kappa(n)\boldsymbol{w}(n) \tag{7}$$

where $\boldsymbol{D}$ has been chosen as an identity matrix. First, it can be seen that the optimal solution to (7) is identical to that of (3). Second, the matrix $\boldsymbol{R}_{XX}(n) + \kappa(n)\boldsymbol{I}$ for a sufficiently large $\kappa(n)$ is positive definite and, hence, invertible. Therefore, the regularization in (7) is unbiased and depends on the state $\boldsymbol{w}(n)$. To iteratively solve (7), the weight vector $\boldsymbol{w}(n)$ on the right-hand side is approximated by its values in the previous iteration, i.e., $\boldsymbol{w}(n-1)$. Hence, the algorithm is asymptotically unbiased. The relationship between this simplified version of (7) and the LMS algorithm can be seen by considering a single measurement $\boldsymbol{x}(n)$ at time instant n, i.e., $\boldsymbol{R}_{XX}(n) \approx \boldsymbol{x}(n)\boldsymbol{x}^T(n)$, $\boldsymbol{p}_X(n) \approx \boldsymbol{x}(n)d(n)$; hence, it is unable to obtain a unique solution to (7). However, the relaxation $\boldsymbol{w}(n) \approx \boldsymbol{w}(n-1)$ allows us to utilize the prior information obtained up to the $(n-1)$th iteration and the current information. Equation (7), after some manipulation, can be rewritten as

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \frac{\boldsymbol{x}(n)e_p(n)}{\kappa(n) + \boldsymbol{x}^T(n)\boldsymbol{x}(n)} \tag{8}$$

where $e_p(n) = d(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n-1)$. One immediately recognizes that (8) is a variable step-size LMS algorithm with the step size $\mu(n) = (\kappa(n) + \boldsymbol{x}^T(n)\boldsymbol{x}(n))^{-1}$, which can also be viewed as a normalized LMS algorithm with unity step size. Thus, the relaxed form of (7) will reduce to the LMS algorithm when one measurement is used. This partially explains the improved performance of the LMS algorithm when the input changes considerably. On the other hand, to avoid an excessive bias, the number of measurements used in the RLS algorithm should be reduced by using a small FF. In this case, the RLS algorithm, without using the prior state information $\boldsymbol{w}(n-1)$, may become unstable. Sufficient L_2 regularization may help to avoid the instability, but the convergence speed has to be compromised. By using the relaxed form of (7) and a small FF, we shall show later that such relaxation can be realized using a QRD implementation. Moreover, in a stationary environment, a large FF can be used together with the QRD to utilize all the available measurements. Since the regularization in (7) involves changing the covariance matrix to $\boldsymbol{R}_{XX}(n) + \kappa(n)\boldsymbol{I}$ and the cross-correlation vector to $\boldsymbol{p}_X(n) + \kappa(n)\boldsymbol{w}(n-1)$, which is a function of the previous state vector, we shall call it "state regularization." In fact, after removing the first term on both sides of (7), the relaxation now becomes

$$\boldsymbol{I} \cdot \boldsymbol{w}(n) \approx \boldsymbol{w}(n-1) \tag{9}$$

which is closely related to the state equation in the Kalman filter. In fact, one can view (9) as a state equation, i.e., $\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \boldsymbol{u}(n)$ with $\boldsymbol{u}(n)$ being the state noise, which requires the current weight vector to stay close to the previous weight vector in case the number of relevant measurements is limited at a nonstationary environment. If $\boldsymbol{u}(n)$ is Gaussian distributed, an L_2 regularization on the deviation from the previous weight vector is imposed. Therefore, this algorithm can be viewed as a special case of Kalman filter, except that an infinite number of measurements with variable weighting is employed in the QRD-RLS algorithm. Further, if $\boldsymbol{u}(n)$ is assumed to be Laplacian distributed, an L_1 regularization on the deviation from previous weight vector should be imposed. Since L_1 and SCAD can be realized as weighted L_2, we only consider L_2 regularization. Moreover, sparsity-enhancing transformation, as proposed in [10], can also be used to improve its performance for sparse coefficient vectors. Finally, we note that the variable regularization parameter described in (6) can be used together with the proposed SR QRD to overcome the problem due to nonpersistent excitation. To solve the relaxed form of (7) recursively, we need to find the QRD of the matrix $\boldsymbol{R}_{XX}(n) + \kappa(n)\boldsymbol{I}$. This can be implemented in a similar way as the L_2-regularized QRRLS. To implement the last term at the right-hand side of (7), we found that, instead of appending $[\sqrt{\mu(n)}\boldsymbol{d}_l, 0]$ to the second QRD, the state regularization can be approximately implemented by appending $[\sqrt{\mu(n)}\boldsymbol{d}_l, \sqrt{\mu(n)}w_l(n-1)]$ to the second QRD successively, where $w_l(n-1)$ is the lth element of $\boldsymbol{w}(n-1)$. The regularization parameter $\kappa(n)$ can also be updated, as in (6). This yields the proposed SR-QRRLS algorithm, as shown in Table I [i.(b)]. This algorithm corrects asymptotically the bias introduced by L_2 regularization. By incorporating the VFF scheme proposed later, it shows in simulation results that the proposed SR-VFF-QRRLS algorithm combines advantages of RLS and LMS while avoiding their disadvantages.

B. SR-QRRLS Algorithm With VFF

As mentioned earlier, to achieve a low steady-state excess MSE (EMSE) in a stationary environment, a large FF has to be used. On the other hand, a relatively small FF has to be used in a nonstationary environment to facilitate tracking. Consequently, the FF plays an important role in such RLS algorithms, and much effort has been devoted to the selection of the FF to obtain good performance in terms of convergence speed, tracking capability, and steady-state EMSE in stationary and nonstationary environments [12]–[15], [18]. In [13] and [14], the FF was updated by applying an appropriate weighting window to the input data sequence. However, it is not easy to determine the parameters for adjusting the FF. In [12], the FF is adaptively adjusted according to the gradient of the estimated MSE so as to minimize the MSE. However, the tracking speed for sudden-change parameters may be degraded since FF may converge slowly. Recently, a robust VFF-RLS algorithm has been proposed in [15] with a similar performance to that in [12]. Intuitively, the FF controls how the measurements are used in estimation. A less number of measurements should be used if the estimation variance increases rapidly due to nonstationary inputs or systems. Here, we propose a measure of the estimation variance of $\boldsymbol{w}(n)$ to determine the number of measurements and, hence, FF to be used. It is known from the classical performance analysis of the LMS algorithm for Gaussian inputs [16] $E[\boldsymbol{x}(n)e(n)] = E[\boldsymbol{x}(n)(\boldsymbol{x}^T(n)(\boldsymbol{w}_0(n) - \boldsymbol{w}(n)) + \eta(n))] = \boldsymbol{R}_{XX}E[\boldsymbol{v}(n)]$, where $\boldsymbol{v}(n) = \boldsymbol{w}_0(n) - \boldsymbol{w}(n)$ is the weight error vector. Therefore, a good measure of the convergence status is the estimated norm of its time average as follows:

$$\sigma_{xe}^2(n) = \lambda_e \sigma_{xe}^2(n-1) + (1-\lambda_e)\boldsymbol{x}_e^T(n)\boldsymbol{x}_e(n) \tag{10}$$

where λ_e is an FF and $\boldsymbol{x}_e(n)$ is averaged from $\boldsymbol{x}(n)e(n)$ over a time window of length T_s so as to suppress the effect of background noise on $\sigma_{xe}^2(n)$. By adopting the approach in [17], we propose to estimate the exponential window size of the algorithm $L(n)$ at each time instant from the measure in (10) as follows:

$$L(n) = \mathrm{round}\left\{L_L + \left[1 - g\left(\overline{G}_N(n)\right)\right](L_U - L_L)\right\} \tag{11}$$

where $\overline{G}_N(n) = \sigma_{xe}^2(n)/\overline{\sigma}_0^2$ with $\overline{\sigma}_0^2$ is the average of the first T_{s0} estimates of $\sigma_{xe}^2(n)$ at the beginning of adaptation; the operator $round\{.\}$ rounds its argument to the nearest integer; L_L and L_U are the lower and upper bounds of $L(n)$, respectively; and $g(x) = \min\{x, 1\}$ is a clipping function that keeps its positive argument x within the interval [0, 1]. From (11), the factor then can be estimated as

$$\lambda(n) = 1 - 1/L(n). \tag{12}$$

Equations (10)–(12) yield the proposed VFF scheme for SR-VFF-QRRLS. It can be seen that if a system change with $\sigma_{xe}^2(n)$ comparable to or larger than $\overline{\sigma}_0^2$ is encountered, a small FF will be chosen to obtain fast tracking speed. At the steady state, $\overline{G}_N(n)$ is usually small, and a larger FF will be employed to obtain a smaller EMSE. Therefore, $\overline{\sigma}_0^2$ serves as a reference for measuring the magnitude of $\sigma_{xe}^2(n)$ to control the FF through (11) and (12). To reduce the effect of input power on $\overline{\sigma}_0^2$, we assume that an approximate nominal signal level $\hat{\sigma}_x^2$ is available, e.g., from users' experience or experiments. Let $\hat{\sigma}_{x0}^2$ be the signal level recorded during the computation of $\overline{\sigma}_0^2$. Then, $\overline{\sigma}_0^2$ can be scaled by $\hat{\sigma}_x^2/\hat{\sigma}_{x0}^2$ for later computation to account for possible dependence on the input power. Alternatively, $\overline{\sigma}_0^2$ can be obtained from Monte Carlo simulations using typical signal levels and system impulse response, if this information is available.

IV. Simulation Results

To evaluate the performance of the proposed algorithms, computer simulations of the system identification problem are carried out. Their performances are compared with the conventional RLS algorithm and the GVFF-RLS algorithm proposed in [12], which usually outperforms other VFF algorithms in convergence and tracking speed [15]. Three different situations are considered. First, the convergence and tracking behavior of various algorithms is examined in a sudden-change model. Second, the tracking capability of these algorithms is studied in a random-walk channel model. Finally, a commonly encountered situation in acoustics, where the system is of long impulse response and the input power is time-varying, is studied. Unless specified otherwise, the simulation results are averaged over 100 Monte Carlo runs.

A. Performance Comparison in a Sudden-Change Channel

In this experiment, the impulse response of the system to be identified changes from $\boldsymbol{w}_0(0) = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]^T$ to $\boldsymbol{w}_0(1500) = [0.1, -0.2, -0.3, -0.4, -0.5, -0.6, 0.7, 0.8, 0.9, 1]^T$ at the 1500th iteration. Both white and colored inputs are used to excite the system. The white Gaussian input is of zero mean and unit variance, whereas the colored input is simulated by a first-order autoregressive (AR) process: $x(n) = 0.9x(n-1) + g(n)$, where $g(n)$ is a zero-mean Gaussian sequence with unit variance. The additive noise power at the output $\boldsymbol{w}_0$ is set to achieve an SNR of 10 dB, where $\mathrm{SNR} = 10\log_{10}(((1/N)\sum_{n=0}^{N} y^2(n))/\sigma_\eta^2)$, and N is the signal length. The proposed SR-VFF-QRRLS are compared with the conventional RLS and the GVFF-RLS algorithm [12]. For a fair comparison, the VFF scheme is also applied to the L_2-QRRLS algorithm with a constant FF in [8]. The resultant

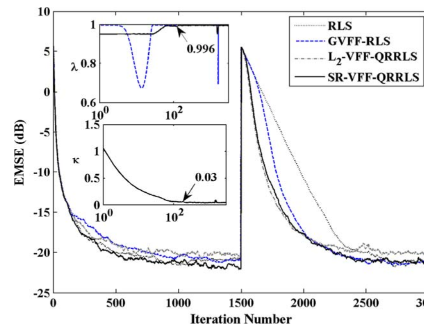

Fig. 1. Learning curves for EMSE in the sudden-change channel model with white Gaussian input at SNR = 10 dB.

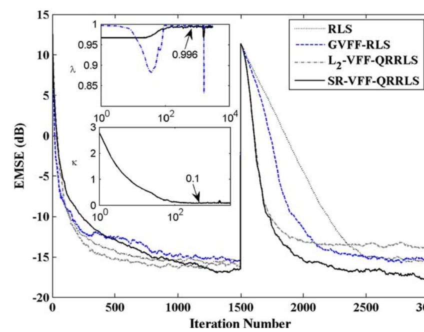

Fig. 2. Learning curves for EMSE in the sudden-change channel model with first-order AR input at SNR = 10 dB.

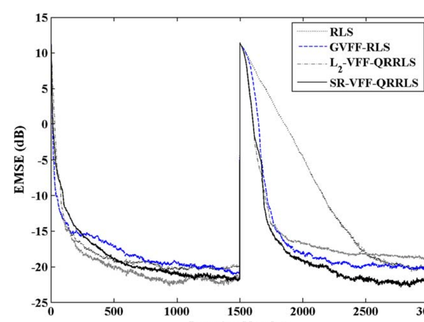

Fig. 3. Learning curves for EMSE in the random-walk channel model with first-order AR input at SNR = 15 dB.

algorithm is denoted as L_2-VFF-QRRLS. The FF of the RLS algorithm is set to be 0.996. The parameters of the GVFF-RLS algorithm are $\alpha = 0.3$, $\beta = 0.99$, $\mu = 0.04$, as used in [12], which also give best performance under this setting. ω^* is chosen as 0.996 to achieve a similar EMSE to the RLS algorithm. After extensive testing, the following parameters for the VFF scheme are recommended to achieve a satisfactory performance under various conditions: a short window length $T_s = 20$ and $\lambda_e = 0.9$ are used to achieve a quick response when the system changes rapidly, and a longer window $T_{s0} = 100$ is used to estimate a more reliable reference for the convergence status $\overline{\sigma}_0^2$. L_L and L_U are, respectively, chosen as 20 and 600 so that the minimum and maximum FFs are 0.95 and 0.998. The EMSE curves of all the algorithms are shown in Figs. 1 and 2. The FF and the regularization parameter $\kappa(n)$ are also shown in the subplots of the figures. The two VFF-QRRLS algorithms have similar convergence speed with that of GVFF-RLS. However, they have faster tracking speeds when the system changes at the 1500th iteration due to the fast response of the proposed

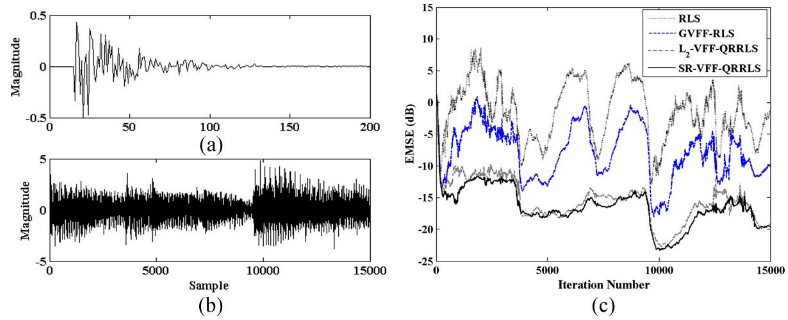

Fig. 4. Learning curves for EMSE (c) for the impulse response in (a) with a segment of music shown in (b) as input at SNR $= 15$ dB.

measure $\overline{G}_N(n)$. The initial convergence speed of SR-VFF-QRRLS over L_2-VFF-QRRLS is not significant because the SR tries to constrain $\boldsymbol{w}(n)$ around its initial value $\mathbf{0}$. Comparing the results for white and colored inputs, the SR-VFF-QRRLS converges to a lower steady-state EMSE than the other algorithms with colored inputs, although they use similar FFs. This is because the regularization further normalizes eigenvalues of colored input covariance matrix as [8]. The high steady-state EMSE of L_2-VFF-QRRLS is due to the bias of L_2. Similar observations were obtained at different SNRs, and they were omitted due to page limitation.

B. Performance Comparison in a Random-Walk Channel

Identification of a time-varying system is considered in this experiment. The random-walk model for channel coefficients is $\boldsymbol{w}_0(n+1) = \boldsymbol{w}_0(n) + \boldsymbol{v}(n)$, where $\boldsymbol{w}_0(0)$ is the same, as in Section IV-A, and $\boldsymbol{v}(n)$ is a white Gaussian vector sequence with zero-mean and variance matrix $\sigma_v^2\boldsymbol{I}$. σ_v^2, and the SNRs are chosen as 5×10^{-5} and 15 dB, respectively. The input signal is the first-order AR process mentioned before. All the algorithm parameters are identical to that in the sudden system change model. The EMSE curves are shown in Fig. 3. As can be seen, the SR-VFF-QRRLS algorithm has the best tracking performance due to the effectiveness of the state regularization imposed by (9). There is a bias in the L_2-VFF-QRRLS algorithm caused by L_2 regularization, particularly after system changes.

C. Performance Comparison With Time-Varying Input Power

In this experiment, the performance of the algorithms is examined in an environment with time-varying input power. The sampling frequency is 16 kHz. A longer channel is used to simulate the acoustic impulse response inside a vehicle as shown in Fig. 4(a), and its length is 200. The input signal is a segment of music, as shown in Fig. 4(b). The SNR is set to be 15 dB. The FF of the RLS algorithm is set to be 0.99. The other algorithmic parameters are the same as that in the previous experiment. The performances of various algorithms are compared in Fig. 4(c). It shows that RLS is very sensitive to input power. The estimation variance becomes very large when the exciting signal is low due to the ill-conditioning problem of the RLS algorithm. GVFF-RLS cannot alleviate this problem by using the VFF. The two regularized VFF-RLS algorithms, however, adaptively select the regularization parameters and offer high immunity to variation in input power. Compared with L_2-VFF-QRRLS, SR-VFF-QRRLS obtains even faster convergence speed, better stability, and lower EMSE values.

V. Conclusion

A new SR-VFF-QRRLS algorithm, which employs previous estimated filter coefficients to stabilize the update and a VFF, has been presented. Improved tracking performance, steady-state MSE, and robustness to power-varying inputs over conventional RLS can be achieved.

References

[1] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2001.

[2] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[3] T. V. Waterschoot, G. Rombouts, and M. Moonen, “Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement,” *Signal Process.*, vol. 88, no. 3, pp. 594–611, Mar. 2008.

[4] A. Houacine, “Regularized fast recursive least squares algorithms for adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 860–871, Apr. 1991.

[5] Z. Tian, K. L. Bell, and H. L. Van Trees, “A recursive least squares implementation for LCMP beamforming under quadratic constraint,” *IEEE Trans. Signal Process.*, vol. 49, no. 6, pp. 1138–1145, Jun. 2001.

[6] J. W. Stokes and J. C. Platt, “Robust RLS with round robin regularization including application to stereo acoustic echo cancellation,” in *Proc. ICASSP*, Toulouse, France, May 14–19, 2006, pp. III376–III379.

[7] S. L. Gay, “Dynamically regularized fast RLS with application to echo cancellation,” in *Proc. ICASSP*, 1996, vol. 2, pp. 957–960.

[8] S. C. Chan, Y. J. Chu, and Z. G. Zhang, “A new regularized QRD recursive least M-estimate algorithm: performance analysis and applications,” in *Proc. ICGCS*, Shanghai, China, Jun. 21–23, 2010, pp. 190–195.

[9] Y. Chen, Y. Gu, and A. O. Hero, “Sparse LMS for system identification,” in *Proc. ICASSP*, Taipei, Taiwan, Apr. 19–24, 2009, pp. 3125–3128.

[10] S. C. Chan, Z. G. Zhang, and Y. J. Chu, “A new transform-domain regularized recursive least M-estimate algorithm for robust linear estimation,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 2, pp. 120–124, Feb. 2011.

[11] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[12] S. H. Leung and C. F. So, “Gradient-based variable forgetting factor RLS algorithm in time-varying environments,” *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3141–3150, Aug. 2005.

[13] D. T. M. Slock and T. Kailath, “Fast transversal filters with data sequence weighting,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 3, pp. 346–359, Mar. 1989.

[14] B. Toplis and S. Pasupathy, “Tracking improvements in fast RLS algorithms using a variable forgetting factor,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 2, pp. 206–227, Feb. 1988.

[15] C. Paleologu, J. Benesty, and S. Ciochina, “A robust variable forgetting factor recursive least-squares algorithm for system identification,” *IEEE Signal Process. Lett.*, vol. 15, no. 3, pp. 597–600, 2008.

[16] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, “Stationary and nonstationary learning characteritic of the LMS adaptive filter,” *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.

[17] B. Liao, Z. G. Zhang, and S. C. Chan, “A new robust Kalman filter-based subspace tracking algorithm in an impulsive noise environment,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 9, pp. 740–744, Sep. 2010.

[18] Y. Zhou, S. C. Chan, and K. L. Ho, “A new variable forgetting factor QR-based recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise environment,” in *Proc. EUSIPCO*, Florence, Italy, Sep. 4–8, 2006.