On an RLS-Like LMS Adaptive Filter

Thiernithi Variddhisaï, Danilo P Mandic

On an RLS-Like LMS Adaptive Filter

Thiernithi Variddhisaï and Danilo P. Mandic
Department of Electrical and Electronic Engineering
Imperial College London, UK
Email: {tt1513,d.mandic}@ic.ac.uk

***Abstract*—A unified and generalized framework for a recursive least squares (RLS)-like least mean square (LMS) algorithm is proposed, which adopts the cost function of the RLS to minimize the mean square error. This paper aims to explore, in a systematic way, the corresponding ideas scattered and multiple-time re-invented in the literature, and introduces a unified approach in the same spirit as in [1], which relates LMS with the Kalman filter. The proposed alternative to the RLS is favored when the matrix inversion lemma is not useful, such as the case of multivariate or multichannel data where the input is not a vector. Furthermore, all the derivations are conducted in the quaternion domain and are hence generalizable to complex- and real-valued models. The resulting algorithm has a neat form and a similar complexity to the RLS. Through experiments, the method is shown to exhibit performance close to or even better than the RLS algorithm. Other aspects, such as the choice of descent directions and variable stepsizes, are also discussed to support the analysis.**

***Keywords*—Online predictor, least mean square, recursive least squares, quaternions, widely linear model, steepest descent, conjugate gradient, Markov conjugate**

I. Introduction

The least mean square (LMS) algorithm has long been a workhorse of adaptive filtering and has, since its conception in 1959 [2], found a number of successful applications. Based on the minimization of an instantaneous approximation to the mean square error, the LMS converges to Wiener-Hopf solution $\mathbf{w}^{op}$ (i.e. the optimal solution), while its 'instantaneous' nature also makes it an algorithm of choice for non-stationary signal processing. Another popular way to reach the Wiener solution, $\mathbf{w}^{op}$, is by using the method of least squares, through the algorithm called the recursive least squares (RLS) [3]. Owing to no approximations in the derivation, the RLS is capable of converging in much fewer iterations than the LMS, and achieve better tracking performance. It is nevertheless not without drawbacks, such as poorer ability to track non-stationary processes or processes with outliers. Moreover, it is less numerically stable and more computationally expensive, both due to the use of matrix inversion lemma [3]. To date, there has been no clear consensus on whether to use the LMS or the RLS, and the answer depends on the applications at hand, while much research effort has been dedicated to improving both algorithms. For LMS, most of its variants have focused on varying the step size, in an attempt to accelerate the convergence and reduce steady-state misalignment [4], while for RLS, the main focus has been on effective implementation and numerical robustness [5].

Many methods were proposed in an attempt to create *alternative* online linear algorithms which inherit the virtues of both LMS and RLS, while mitigating some of their drawbacks. Albeit introduced under seemingly different concepts(e.g. [6], [7], [8]), these RLS-like methods would benefit from being undertaken within one unifying framework. This paper intends to provide a rigorous and systematic derivation of a unifying algorithm in this class, based on such a unified approach, so that it can be more accessible and more easily utilized. The other important contribution here is that the framework was derived in the quaternion domain which offers twofold advantages; first, since quaternions are generalization of real and complex numbers so that the derived algorithms will also hold true for both these number systems; second, the adoption rate of quaternions is increasing in a variety of applications, such as modeling of 3-D rotational data (computer graphics [9] and array signal processing [10], [11], [12]), color image processing [13], [14] and source separation [15]. Hence, deriving the proposed framework in the quaternion domain offers more rigor and intuition in terms of physical interpretation, algebraic generality and algorithmic elegance.

II. Background

A. Quaternions and GHR Calculus

Since there is no shortage of excellent tutorials on basic quaternion algebra [16], [17], we start from the generalized HR (GHR) calculus, the recent concept which enables our proposed method to have an elegant form in the quaternion domain which is consistent with its real- and complex-valued counterparts. For detailed information, we refer to [18], [19].

Consider a general case of functions $f(\mathbf{q}) : \mathbb{H}^{M\times 1} \rightarrow \mathbb{H}$, where $\mathbf{q} = (q_1, q_2, ..., q_M)^T \in \mathbb{H}^{M\times 1}$. Then, the quaternion gradient and conjugate gradient are respectively given by [19]

$$\nabla_{\mathbf{q}} f \triangleq \left(\frac{\partial f}{\partial \mathbf{q}}\right)^T = \left(\frac{\partial f}{\partial q_1}, ..., \frac{\partial f}{\partial q_M}\right)^T \in \mathbb{H}^{M\times 1},$$
$$\nabla_{\mathbf{q}^*} f \triangleq \left(\frac{\partial f}{\partial \mathbf{q}^*}\right)^T = \left(\frac{\partial f}{\partial q_1^*}, ..., \frac{\partial f}{\partial q_M^*}\right)^T \in \mathbb{H}^{M\times 1}.$$

Similar to the complex domain, it was shown that the conjugate gradient $\nabla_{\mathbf{q}^*} f$ yields the steepest descent direction of the function f [20]. This makes the conjugate GHR derivatives a natural choice for the optimization of our proposed algorithm.

Within the GHR calculus, the conjugate derivative is defined based on a generalized orthogonal basis $\{1, i^{\mu}, j^{\mu}, k^{\mu}\}$ where μ is a rotating quaternion. In our work, a simplified case where $\mu = 1$ suffices and yields a neater form of the HR conjugate derivative [20]. The left and right derivative operators, which arise out of the non-commutative nature of quaternions, are respectively defined as follows

$$\frac{\partial f}{\partial q^*} = \frac{1}{4}\left(\frac{\partial f}{\partial q_a} + \frac{\partial f}{\partial q_b}i + \frac{\partial f}{\partial q_c}j + \frac{\partial f}{\partial q_d}k\right),$$
$$\frac{\partial_r f}{\partial q^*} = \frac{1}{4}\left(\frac{\partial f}{\partial q_a} + i\frac{\partial f}{\partial q_b} + j\frac{\partial f}{\partial q_c} + k\frac{\partial f}{\partial q_d}\right).$$

where $q = q_a + iq_b + jq_c + kq_d,\ q_a, q_b, q_c, q_d \in \mathbb{R}$ and $i^2 = j^2 = k^2 = -1$. Observe that when f is real-valued, both left and right derivatives are identical, which is the case with our analysis, and from now on, we will therefore implicitly refer to the left derivative.

The key benefits of the GHR calculus are the novel product rule and chain rule (not present in the HR calculus). These two rules are integral in the derivation of the proposed framework. For the product rule, if the function $f, g : \mathbb{H} \to \mathbb{H}$ are real-differentiable, then so too is their product, that is,

$$\frac{\partial (fg)}{\partial q^*} = f\frac{\partial g}{\partial q^*} + \frac{\partial f}{\partial q^{g*}}g \tag{1}$$

where $q^{g*} = gq^*g^{-1}$ is a quaternion rotation [18]. For the chain rule, if $g : \mathcal{S} \to \mathbb{H}$ and $f : \mathcal{T} \to \mathbb{H}$ are real-differentiable at the respective interior points, $q \in \mathcal{S} \subseteq \mathbb{H}$ and $g(q) \in \mathcal{T} \subseteq \mathbb{H}$, then the derivative of the composite function $f(g(q))$ is given by

$$\begin{aligned}\frac{\partial f(g)}{\partial q^*} &= \frac{\partial f}{\partial g}\frac{\partial g}{\partial q^*} + \frac{\partial f}{\partial g^i}\frac{\partial g^i}{\partial q^*} + \frac{\partial f}{\partial g^j}\frac{\partial g^j}{\partial q^*} + \frac{\partial f}{\partial g^k}\frac{\partial g^k}{\partial q^*} \\ &= \frac{\partial f}{\partial g^*}\frac{\partial g^*}{\partial q^*} + \frac{\partial f}{\partial g^{i*}}\frac{\partial g^{i*}}{\partial q^*} + \frac{\partial f}{\partial g^{j*}}\frac{\partial g^{j*}}{\partial q^*} + \frac{\partial f}{\partial g^{k*}}\frac{\partial g^{k*}}{\partial q^*}\end{aligned} \tag{2}$$

B. Adaptive Filters

For completeness, we next briefly summarize the LMS and RLS algorithms in their most general form - the widely linear quaternion model (for general non-circular data [21]).

Denoted by $y_n, x_n \in \mathbb{H}$, $n = 1, ..., N$ the desired (output) signal and input signal, respectively, the linear MSE estimator of y_n, denoted by $\hat{y}_n$, can then be expressed as [22]

$$\hat{y}_n = \langle \hat{\mathbf{w}}, \mathbf{q}_n \rangle \triangleq \hat{\mathbf{w}}^H \mathbf{q}_n \tag{3}$$

where $\hat{\mathbf{w}}$ is an estimate of the optimal solution $\mathbf{w}^{op}$, $\mathbf{q}_n$ is defined as

$$\mathbf{q}_n = [\mathbf{x}_n^T\ \mathbf{x}_n^{iT}\ \mathbf{x}_n^{jT}\ \mathbf{x}_n^{kT}]^T \tag{4}$$

and $\mathbf{x}_n = [x_n, x_{n-1}, ..., x_{n-M+1}]^T$ for a filter of order M. For a simple linear model (i.e. data is circular [21]),

$$\mathbf{q}_n = \mathbf{x}_n \tag{5}$$

The goal of an online linear predictor is to minimize the mean square error (MSE)

$$\mathcal{J}_n(\hat{\mathbf{w}}) = E\{\|\xi_n(\hat{\mathbf{w}})\|^2\} \tag{6a}$$

$$\xi_n(\hat{\mathbf{w}}) = y_n - \hat{y}_n = y_n - \hat{\mathbf{w}}^H \mathbf{q}_n, \tag{6b}$$

recursively, with the weight update expressed as [18]

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \alpha_n \nabla_{\hat{\mathbf{w}}^*} \mathcal{J}_n(\hat{\mathbf{w}})|_{\hat{\mathbf{w}}=\mathbf{w}_{n-1}}. \tag{7}$$

In the case of the LMS algorithm, eq. (6a) becomes an instantaneous estimate, that is,

$$\mathcal{J}_n(\hat{\mathbf{w}}) \approx \|\xi_n(\hat{\mathbf{w}})\|^2, \tag{8}$$

which as a result gives the following update [23]

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \alpha_n \mathbf{q}_n e_n^* \tag{9}$$

where e_n is *a priori* error, defined as

$$e_n \triangleq \xi_n(\mathbf{w}_{n-1}) = y_n - \mathbf{w}_{n-1}^H \mathbf{q}_n \tag{10}$$

and $\alpha_n \in \mathbb{R}$ is an adaptive stepsize. For the normalized LMS, α_n can be found via the minimum disturbance principle [3] as

$$\alpha_n = \frac{1}{\|\mathbf{q}_n\|^2 + \epsilon} \tag{11}$$

where $\epsilon > 0$ is a regularizer which prevents numerical instability when $\|\mathbf{q}_n\|^2 \to 0$.

On the other hand, the RLS filter iteratively solves the least square solution, $\hat{\mathbf{w}}$, of the given set of training pairs y_n, x_n via the following deterministic cost function [25],

$$\Phi_n(\hat{\mathbf{w}}) = \sum_{k=1}^{n} \lambda^{n-k} \|\xi_k(\hat{\mathbf{w}})\|^2 \tag{12}$$

where $0 < \lambda < 1$ is a forgetting factor used to suppress the effect of early data which may be no longer relevant to the current estimate. The core idea of the proposed method is to derive an LMS algorithm which approximates MSE in a manner similar to eq. (12). We did not describe the whole RLS algorithm owing to page limits.

III. An RLS-Inspired LMS Algorithm

A. Derivation in Quaternions

Inspired by the RLS cost function in eq. (12), we reformulate the estimation of the MSE in eq. (6a) to

$$\mathcal{J}_n(\hat{\mathbf{w}}) \approx 2\Phi_n(\hat{\mathbf{w}}) \tag{13}$$

Here, the factor 2 is present to normalize the expressions involving GHR derivatives [19] (which actually can be absorbed into the stepsize α_n). Then, we substitute eqs. (3) and (6b) into eq. (13) to yield

$$\begin{aligned}\mathcal{J}_n(\hat{\mathbf{w}}) =& \hat{\mathbf{w}}^H \mathbf{R}_n \hat{\mathbf{w}} - 2\Re\{\hat{\mathbf{w}}^H \mathbf{r}_n\} \\ &+ \sum_{k=1}^{n} \lambda^{n-k} |y_k|^2\end{aligned} \tag{14}$$

where $\Re\{\cdot\}$ is a real-part operator with

$$\mathbf{R}_n = \sum_{k=1}^{n} \lambda^{n-k} \mathbf{q}_n \mathbf{q}_n^H = \lambda \mathbf{R}_{n-1} + \mathbf{q}_n \mathbf{q}_n^H, \tag{15}$$

$$\mathbf{r}_n = \sum_{k=1}^{n} \lambda^{n-k} \mathbf{q}_n y_n^* = \lambda \mathbf{r}_{n-1} + \mathbf{q}_n y_n^*. \tag{16}$$

We now follow the optimization path of LMS with the MSE estimate as in eq. (12). Through eqs. (1) and (2), the *a priori* gradient $\mathbf{g}_{n|n-1}$ is defined as

$$\mathbf{g}_{n|n-1} \triangleq \nabla_{\hat{\mathbf{w}}^*} \mathcal{J}_n(\hat{\mathbf{w}})|_{\hat{\mathbf{w}}=\mathbf{w}_{n-1}} = \mathbf{R}_n \mathbf{w}_{n-1} - \mathbf{r}_n. \tag{17}$$

Then, the expression of the quaternion weight update becomes

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{d}_n \alpha_n \tag{18}$$

where $\mathbf{d}_n$ is the descent direction, while the stepsize $\alpha_n \in \mathbb{H}$, unlike eq. (7), is non-commutative and has to post-multiply $\mathbf{d}_n$ so that it will not lead to Sylvester's equation which has no closed-form solution [26]. One convenient way to find α_n is by using exact line search (equivalent to method of minimum disturbance in the NLMS case [27]), where $\partial \mathcal{J}_n(\mathbf{w}_n)/\partial \alpha_n$ is set to zero. Upon substituting into eq. (7), we obtain

$$\alpha_n = -\frac{\langle \mathbf{d}_n, \mathbf{g}_{n|n-1} \rangle}{\langle \mathbf{d}_n, \mathbf{h}_{n|n} \rangle} \tag{19}$$

where

$$\mathbf{h}_{n|n} = \mathbf{R}_n \mathbf{d}_n. \tag{20}$$

Observe that $\langle \mathbf{d}_n, \mathbf{h}_{n|n} \rangle$ is real-valued. Now, to reuse past calculations efficiently, *a posteriori* gradient $\mathbf{g}_{n|n}$ is defined as

$$\mathbf{g}_{n|n} \triangleq \nabla_{\hat{\mathbf{w}}^*} \mathcal{J}_n(\hat{\mathbf{w}})|_{\hat{\mathbf{w}}=\mathbf{w}_n} = \mathbf{R}_n \mathbf{w}_n - \mathbf{r}_n, \tag{21}$$

and after some manipulation, we arrive at the following dual recursion

$$\mathbf{g}_{n|n-1} = \lambda \mathbf{g}_{n-1|n-1} - \mathbf{q}_n e_n^* \tag{22}$$

$$\mathbf{g}_{n|n} = \mathbf{g}_{n|n-1} + \mathbf{h}_{n|n} \alpha_n. \tag{23}$$

B. Choice of Descent Directions

Most works concerning the RLS-like LMS algorithm use the conjugate gradient (CG) direction because it is the fastest first-order solver of a quadratic-form equation; however, due to the time-varying statistics of $\mathbf{R}_n$, most proposed CG techniques fail to sustain the conjugacy property (see [6] and reference therein). To overcome this issue, some methods were introduced to relax the conjugacy constraint, like control Lyapunov functions [7] and a CG-like technique [8]. All these methods can be considered under one unifying umbrella termed *Markov conjugacy*.

Definition 1. *A set of descent directions* $\{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_n\}$ *is Markov conjugate if, at any iteration i,*

$$\mathbf{d}_i^H \mathbf{R}_i \mathbf{d}_{i-1} = 0 \text{ for } i = 1, 2, ..., n. \tag{24}$$

In the follow-up paper, we show this idea can easily generalize into a more sophisticated technique. With only one degree of freedom in eq. (24), the *Markov* conjugate direction is expressed as

$$\mathbf{d}_n = -\mathbf{g}_{n|n-1} + \mathbf{d}_{n-1} \beta_n. \tag{25}$$

Pre-multiplying with $(\mathbf{d}_{n-1})^H \mathbf{R}_n$ and substituting eq. (24) into eq. (25) yields

$$\beta_n = \frac{\langle \mathbf{d}_{n-1}, \mathbf{v}_n \rangle}{\langle \mathbf{d}_{n-1}, \mathbf{h}_{n|n-1} \rangle} \tag{26}$$

where

$$\mathbf{h}_{n|n-1} = \mathbf{R}_n \mathbf{d}_{n-1} \tag{27}$$

$$\mathbf{v}_n = \mathbf{R}_n \mathbf{g}_{n|n-1}. \tag{28}$$

and after some manipulation, we arrive at the another dual recursion

$$\mathbf{h}_{n|n-1} = \lambda \mathbf{h}_{n-1|n-1} + \mathbf{q}_n(\mathbf{q}_n^H \mathbf{d}_{n-1}) \tag{29}$$

$$\mathbf{h}_{n|n} = -\mathbf{v}_n + \mathbf{h}_{n|n-1} \beta_n. \tag{30}$$

An interesting simplification is the *steepest descent* scheme where $\beta_n = 0$, $\forall n$ so that $\mathbf{d}_n$ in eq. (25) becomes

$$\mathbf{d}_n = -\mathbf{g}_{n|n-1} \tag{31}$$

and it can be shown that α_n will become real-valued, i.e.,

$$\alpha_n = \frac{1}{\pi_n} = \frac{\|\mathbf{g}_{n|n-1}\|^2}{\langle \mathbf{g}_{n|n-1}, \mathbf{v}_n \rangle}. \tag{32}$$

This is very beneficial in that all the techniques of variable stepsize proposed in the real domain [28] are directly applicable in quaternions, i.e.

$$\alpha_n = \frac{1}{\pi_n + \epsilon_n} \tag{33}$$

where $\epsilon_n \in \mathbb{R}$ is an adaptive regularizer [29]. The unified routine of the RLS-like LMS algorithm, which we shall refer to as the *m*-NLMS i.e. the NLMS with *memory*, is summarized in Algorithm 1, where $\delta > 0$ is used as a stopping criterion.

Algorithm 1: The *m*-NLMS Algorithm

```
Input  : $x_n$, $y_n$, $M$, $\lambda$, $\epsilon$, $\delta$ and update scheme
Output: $\hat{y}_n$ and $\mathbf{w}^{op}$
Initialize $\mathbf{w}_0 = \mathbf{d}_0 = \mathbf{g}_{0|0} = \mathbf{h}_{0|0} = \mathbf{0}$ and $\mathbf{R}_0 = \underline{\mathbf{0}}$;
$n = 0$;
do
    $n = n + 1$;
    Update $\mathbf{q}_n$ by $x_n$ according to eqs. (4) and (5);
    $\hat{y}_n - (\mathbf{w}_{n-1})^H \mathbf{q}_n$;
    $e_n = y_n - \hat{y}_n$;
    $\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \mathbf{q}_n \mathbf{q}_n^H$;
    $\mathbf{g}_{n|n-1} = \lambda \mathbf{g}_{n-1|n-1} - \mathbf{q}_n e_n^*$;
    $\mathbf{v}_n = \mathbf{R}_n \mathbf{g}_{n|n-1}$;
    if Markov conjugate scheme then
        $\mathbf{h}_{n|n-1} = \lambda \mathbf{h}_{n-1|n-1} + \mathbf{q}_n(\mathbf{q}_n^H \mathbf{d}_{n-1})$;
        $\beta_n = \frac{\langle \mathbf{d}_{n-1}, \mathbf{v}_n \rangle}{\langle \mathbf{d}_{n-1}, \mathbf{h}_{n|n-1} \rangle}$    $(\beta_1 = 0)$;
        $\mathbf{d}_n = -\mathbf{g}_{n|n-1} + \mathbf{d}_{n-1} \beta_n$;
        $\mathbf{h}_{n|n} = -\mathbf{v}_n + \mathbf{h}_{n|n-1} \beta_n$;
        $\alpha_n = -\frac{\langle \mathbf{d}_n, \mathbf{g}_{n|n-1} \rangle}{\langle \mathbf{d}_n, \mathbf{h}_{n|n} \rangle}$;
    else steepest descent scheme
        $\mathbf{d}_n = -\mathbf{g}_{n|n-1}$;
        $\alpha_n = \frac{1}{\pi_n + \epsilon_n}$ where $\pi_n = \frac{\langle \mathbf{g}_{n|n-1}, \mathbf{v}_n \rangle}{\|\mathbf{g}_{n|n-1}\|^2}$;
    end
    $\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{d}_n \alpha_n$;
    $\mathbf{g}_{n|n} = \mathbf{g}_{n|n-1} + \mathbf{h}_{n|n} \alpha_n$;
while $\|e_n\| > \delta$ or $n = N$;
$\mathbf{w}^{op} = \mathbf{w}_n$.
```

IV. Numerical Experiments

A set of numerical case studies was performed to benchmark the validity of the *m*-NLMS algorithm against the original LMS and RLS algorithms.

For rigor, the simulations were conducted within a widely linear quaternion framework which has found its potential in MIMO channel identification [30]. Here, a widely linear quaternion moving average of order 3 - WLQMA(3) was used, where the input x_n was drawn from 1000 samples of the same distribution $\mathcal{N}(0,1)$ for each component of x_n, with a moderate SNR of 40dB. Every experiment was conducted over 100 independent realizations and the *m*-NLMS used in the experiments, if not specified otherwise, was the *steepest descent* (SD) scheme. For a fair comparison, the stepsizes

α_n for the original LMS routines was calculated using the minimum disturbance principle as in eq. (11) in which the original LMS yields NLMS. For convenience, we shall omit the term ‘quaternion’, as our primary attention is paid to the algorithmic mechanism. The metric used to benchmark the algorithms was the normalized misalignment, η_n, defined as

$$\eta_n \triangleq \frac{\left\|\mathbf{w}_n - \mathbf{w}^{op}\right\|^2}{\left\|\mathbf{w}^{op}\right\|^2}. \tag{34}$$

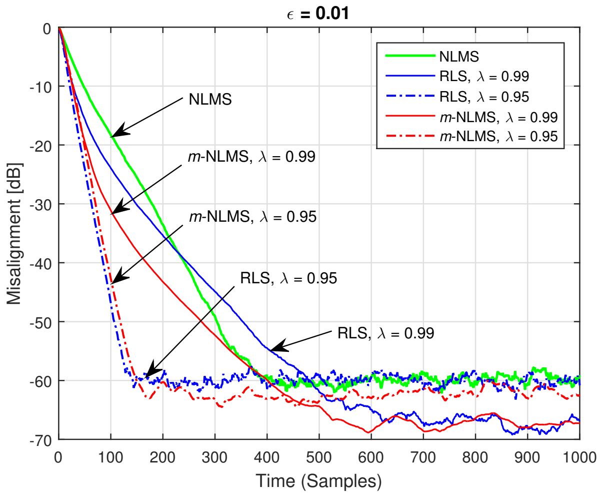

Fig. 1: Misalignment of NLMS, RLS, and *m*-NLMS algorithms with different λ, averaged over 100 trials, for the identification of a WLQMA(3) process, at an SNR of 40dB

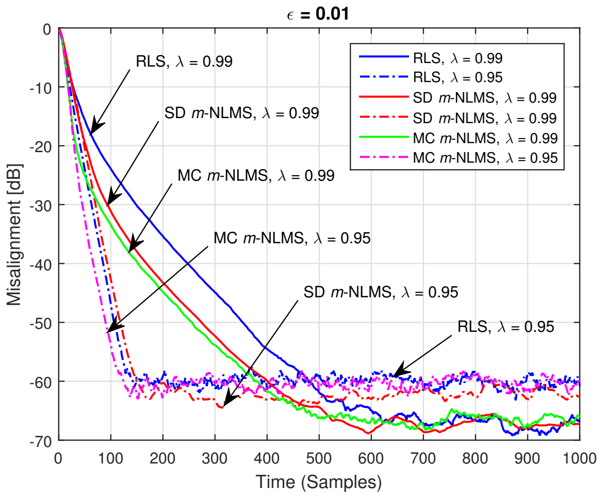

Fig. 2: Misalignment of RLS, SD *m*-NLMS and MC *m*-NLMS algorithms with different λ, averaged over 100 trials, for the identification of a WLQMA(3) process, at an SNR of 40dB

In the first experiment, we tested how our new algorithm performed against the NLMS and RLS with different parameter settings but with a fixed $\epsilon_n = 0.01$ just to ensure numerical stability. As seen in Fig. 1, the *m*-NLMS converged faster and achieved lower steady-state misalignment than the standard NLMS. At $\lambda = 0.99$, the *m*-NLMS also converged faster than RLS while at $\lambda = 0.95$, the *m*-NLMS converged slightly slower but achieved better misalignment than RLS. Fig. 2 demonstrates the performance of the *Markov*-conjugate (MC) *m*-NLMS. At $\lambda = 0.99$, the MC scheme converged slightly faster than the SD scheme and achieved the same misalignment as the RLS while at $\lambda = 0.95$, it converged even faster than the RLS but achieved worse misalignment than the SD scheme. In other words, the *m*-NLMS, regardless of its descent schemes, behaves on par with the RLS.

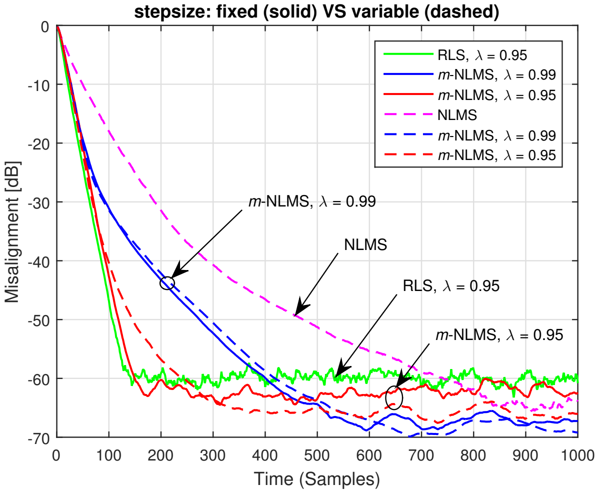

Fig. 3: Misalignment of NLMS, RLS, and *m*-NLMS algorithms with different value of with λ and either fixed or variable stepsizes, averaged over 100 trials, for the identification of a WLQMA(3) process, at an SNR of 40dB

Fig. 3 shows the results of the second experiment which examined the variable-stepsize version of the *m*-NLMS. Three fixed-stepsize routines (RLS and *m*-NLMS) were benchmarked against three variable-stepsize ones (NLLS and *m*-NLMS with GSER scheme [28]). By inspection of *m*-NLMS at $\lambda = 0.99$ and 0.95 as well as comparison of NLMS between Fig. 3 and Fig. 1, it is clear that making ϵ_n variable reduces misalignment but also convergence rate. A closer inspection of *m*-NLMS between $\lambda = 0.99$ and 0.95 reveals that the reduction in

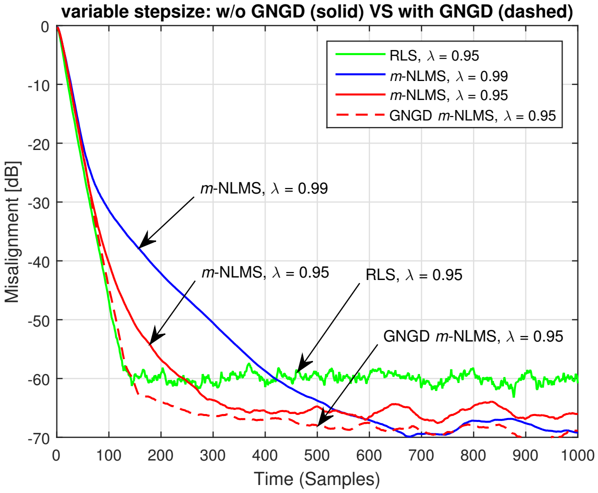

Fig. 4: Misalignment of RLS and *m*-NLMS algorithms with different value of with λ and variable stepsizes both with and w/o GNGD scheme, averaged over 100 trials, for the identification of a WLQMA(3) process, at an SNR of 40dB

misalignment for the case $\lambda = 0.95$ is more significant than that for $\lambda = 0.99$. It is also feasible to make the variable stepsize more effective by tracking whether the learning process is in the transient or steady state. A technique like GNGD [29] could be incorporated to render the algorithm achieving even lower misalignment, without trading off convergence speed, as illustrated by a dashed line in Fig. 4. We did not provide further details on GNGD *m*-NLMS and experiments on real-world signals, like 3-D wind, which will be covered in the follow-up paper and supported with more analysis. Intuitively however, it would behave as consistently as in this paper.

V. Conclusion

In an effort to explore the space of algorithms spanned by the LMS and RLS, we have built on the work in [1], based on the Kalman filter, to provide a unified and generalized approach to a class of RLS-inspired LMS adaptive filters. This has resulted in the proposed *m*-NLMS framework which has been derived in the quaternion domain so that the resulting algorithm, which has the complexity comparable to the RLS, can be straightforwardly simplified to complex- and real-valued cases. Two options of descent direction have also been derived rigorously, under the unifying notion of *Markov* conjugacy proposed in this paper. Since the *m*-NLMS is technically a variant of LMS, it is more numerically stable than RLS, as no matrix inversion is involved in the update. The experiments have demonstrated that the *m*-NLMS has been able to achieve performance which is more or less identical to the RLS, regarding both convergence rate and misalignment. Sharing the virtues of both LMS and RLS, the *m*-NLMS has been shown to admit further improvement through a variable stepsize. The experiments in the system identification have supported the analysis.

References

[1] D. P. Mandic, S. Kanna and A. G. Constantinides, "On the intrinsic relationship between the least mean square and kalman filters [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 117-122, Nov. 2015.

[2] B. Widrow, M.E. Hoff, "Adaptive switching circuits," in *IRE Wescon Conv Rec 4*, pp. 96104.

[3] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New Jersey: Wiley, 2003

[4] D. Bismor, K. Czy, Z. Ogonowski, "Review and comparison of variable step-size LMS algorithms", *International Journal of Acoustics and Vibration*, vol. 21, pp. 24-39, 2016.

[5] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Springer Science & Business Media, 2012.

[6] P. S. Chang and A. N. Willson, "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 48, no. 2, pp. 409-418, Feb 2000.

[7] O. Diene and A. Bhaya, "Adaptive filtering algorithms designed using control Lyapunov functions," *IEEE Signal Processing Letters*, vol. 13, no. 4, pp. 224-227, April 2006.

[8] S. K. Zhao, Z. H. Man and S. Y. Khoo, "Conjugate gradient algorithm design with RLS normal equation," in Proc. *6th International Conference on Information, Communications & Signal Processing*, Singapore, 2007, pp. 1-5.

[9] T. Rabbani, *Automatic reconstruction of industrial installations using point clouds and images*, NCG Nederlandse Commissie voor Geodesie Netherlands Geodetic Commission, 2006.

[10] S. Miron, N. Le Bihan and J. I. Mars, "Quaternion-MUSIC for vector-sensor array processing," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1218-1229, Apr. 2006.

[11] J. W. Tao and W. X. Chang, "A novel combined beamformer based on hypercomplex processes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1276-1289, Apr. 2013.

[12] X. Zhang, W. Liu, Y. Xu, Z. Liu, "Quaternion-valued robust adaptive beamformer for electromagnetic vector-sensor arrays with worst-case constraint," *Signal Processing*, vol. 104, pp. 274-283, Nov. 2014.

[13] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, and J. L. Coatrieux, "Color image analysis by quaternion-type moments." *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1 , pp. 124-144, Jan. 2015.

[14] Y. Xu, L. Yu, H. Xu, H. Zhang and T. Nguyen, "Vector sparse representation of color image using quaternion matrix analysis," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1315-1329, Apr. 2015.

[15] T. S. T. Chan and Y. H. Yang, "Complex and quaternionic principal component pursuit and its application to audio separation," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 287-291, Feb. 2016.

[16] I. L. Kantor, and A. S. Solodovnikov. *Hypercomplex Numbers: An Elementary Introduction to Algebras*. New York: Springer-Verlag, 1989.

[17] J. P. Ward, *Quaternions and Cayley Numbers: Algebra and Applications*. Dordrecht: Kluwer Academic Publishers, 1997.

[18] D. Xu, C. Jahanchahi, C. C. Took, D. P. Mandic, "Enabling quaternion derivatives: The generalized HR calculus," *Royal Society Open Science*, vol. 2, no. 8, pp. 150-255, Aug. 2015.

[19] D. Xu, Y. Xia and D. P. Mandic, "Optimization in quaternion dynamic systems: Gradient, Hessian, and learning algorithms," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 249-261, Feb. 2016.

[20] D. P. Mandic, C. Jahanchahi and C. C. Took, "A quaternion gradient operator and its applications," *IEEE Signal Processing Letters*, vol. 18, no. 1, pp. 47-50, Jan. 2011.

[21] C. Took and D. Mandic, "Augmented second-order statistics of quaternion random signals," *Signal Processing*, vol. 91, no. 2, pp. 214-224, 2011.

[22] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Education; 4^{th} edition 2002.

[23] C. Took and D. Mandic, "The quaternion LMS algorithm for adaptive filtering of hypercomplex processes," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1316-1327, Apr 2009.

[24] C. Took and D. Mandic, "A quaternion widely linear adaptive filter," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4427-4431, Aug 2010

[25] C. Jahanchahi, C. Cheong Took and D. P. Mandic, "The widely linear quaternion recursive least squares filter," in Proc. 2^{nd} *International Workshop on Cognitive Information Processing*, pp. 87-92, Elba, 2010.

[26] V.S Shpakivskyi, "Linear quaternionic equations and their systems," *Advances in Applied Clifford Algebra.*, vol. 21, issue 3, pp. 637-645, 2011.

[27] D. R. Morgan and S. G. Kratzer, "On a class of computationally efficient, rapidly converging, generalized NLMS algorithms," *IEEE Signal Processing Letters*, vol. 3, no. 8, pp. 245-247, Aug. 1996.

[28] J. H. Lee, J. W. Chen and H. C. Huang "Performance comparison of variable step size NLMS algorithm," in *Proceedings Of The World Congress On Engineering and Computer Science USA*, Vol-1, Oct. 2009.

[29] D. P. Mandic, "A generalized normalized gradient descent algorithm," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 115-118, Feb. 2004.

[30] Y. Ding, N. Li, Y. Wang, S. Feng and H. Chen, "Widely linear sphere decoder in MIMO systems by exploiting the conjugate symmetry of linearly modulated signals," *IEEE Transactions on Signal Processing*, vol. 64, no. 24, pp. 6428-6442, Dec. 2016.