Quaternion-Valued Adaptive Filtering via Nesterov's Extrapolation

Thiernithi Variddhisaï[1], Min Xiang[1], Scott C. Douglas[2] and Danilo P. Mandic[1]

[1]Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, UK

[2]Department of Electrical Engineering, Southern Methodist University, Dallas, Texas 75275, USA

Email: {tt1513,m.xiang13,d.mandic}@ic.ac.uk, douglas@lyle.smu.edu

***Abstract*—A new quaternion-valued adaptive filtering algorithm based on extrapolated weight methods is proposed. The proposed algorithm belongs to the class of conjugate direction algorithms [1]. This class of extrapolation (momentum) based algorithms is preferred to RLS-based algorithms when the matrix inversion should be avoided, e.g. in the case of non-vector signals, sparse signals or non-stationary signals. This paper introduces Nesterov's optimal gradient methods in widely linear quaternion adaptive filtering. The resulting class of algorithm is shown to both have similar computational complexity and comparable performance to WLQRLS; however, the proposed method is more stable and outperforms WLQRLS in the non-stationary case.**

***Keywords*—Least mean square, recursive least squares gradient, quaternions, widely linear model, Nesterov's gradient.**

I. Introduction

Adaptive filters have been widely used in numerous engineering fields. In a stationary environment, the filter can converge to the Wiener solution in a statistical sense in a recursive manner, while in a non-stationary environment, the filter can track time variations in the statistics of the input data [2]. The least mean square (LMS) and the recursive least squares (RLS) are two of the most popular adaptive filtering algorithms. The former minimizes an instantaneous approximation to the mean square error, while the latter minimizes the accumulated squared error. Consequently, RLS is capable of converging in much fewer iterations than LMS and usually achieves better tracking performance. However, RLS suffers from numerical instability and expensive computation due to the use of the matrix inversion [2], and due to the "accumulated error" cost function. Much research effort has been dedicated to improving these two 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 [3]–[6], while the main focus on enhancing RLS has been on effective implementation and numerical robustness [7].

Many methods have been 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 [8]–[10]. Recently, these RLS-like methods have been incorporated into a unifying framework [1], which is based on the work relating LMS to Kalman filters [11]. This paper proposes an alternative algorithm by using extrapolated (or momentum) gradients based on Nesterov's optimal momentum. Expanding on the framework introduced in [1], this paper considers both algorithms in the quaternion domain because: 1) quaternions are generalization of real and complex numbers so that the derived algorithms will straightforwardly account for both these number systems; 2) the adoption rate of quaternions is increasing in a variety of applications, such as computer graphics [12], array signal processing [13]–[15], color image processing [16], [17] and source separation [18]. Hence, deriving the proposed framework in the quaternion domain offers rigor and intuition in terms of physical interpretation, algebraic generality and algorithmic elegance. Compared to standard RLS, the proposed method avoids computationally expensive and often unstable matrix inversions and is shown to exhibit superior performance in the non-stationary case.

II. Background

The quaternion domain $\mathbb{H}$ is a four-dimensional vector space over the real field $\mathbb{R}$, spanned by the basis $\{1, \imath, \jmath, \kappa\}$. A quaternion variable $q \in \mathbb{H}$ is of the form $q = q_a + q_b\imath + q_c\jmath + q_d\kappa$ where q_a, q_b, q_c, q_d are real variables and $\imath, \jmath, \kappa$ are imaginary units with the properties

$$\begin{aligned} \imath^2 = \jmath^2 = \kappa^2 = -1, \qquad \imath\jmath = -\jmath\imath = \kappa \\ \jmath\kappa = -\kappa\jmath = \imath, \qquad \kappa\imath = -\imath\kappa = \jmath \end{aligned}$$

The conjugate of q is defined as $q^* = q_a - q_b\imath - q_c\jmath - q_d\kappa$, while the modulus of q is given by $|q| = \sqrt{q_a^2 + q_b^2 + q_c^2 + q_d^2}$. Note that quaternion algebra is non-commutative. The transformation $q^\mu \triangleq -\mu q \mu$, geometrically describes a three-dimensional rotation of the vector part of q about the vector part of a quaternion μ [20]. We refer to [19], [20] for more detail on basic quaternion algebra.

The generalized $\mathbb{HR}$ (G$\mathbb{HR}$) calculus enables an elegant form of gradient in the quaternion domain which is consistent with its real- and complex-valued counterparts [21]–[23]. For a function $f(\mathbf{q}) : \mathbb{H}^{M\times 1} \rightarrow \mathbb{H}$, where $\mathbf{q} = (q_1, q_2, \ldots, q_M) \in \mathbb{H}^{M\times 1}$, the quaternion gradient and conjugate gradient are respectively given by [21]

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

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 [23]. This makes the conjugate derivatives a natural choice for the optimization of quaternion algorithms. Within the G$\mathbb{HR}$ calculus, the conjugate derivative is defined based on a generalised basis $\{1, \imath^\mu, \jmath^\mu, \kappa^\mu\}$ where μ is a quaternion.

The left derivative operator is defined as [22]–[24]

$$\frac{\partial f}{\partial q^{\mu *}} \triangleq \frac{1}{4}\left(\frac{\partial f}{\partial q_a}+\frac{\partial f}{\partial q_b}\imath^{\mu}+\frac{\partial f}{\partial q_c}\jmath^{\mu}+\frac{\partial f}{\partial q_d}\kappa^{\mu}\right)$$

The key benefits of this calculus are the novel product and chain rule, which accounts for the non-commutative nature of quaternion products. The product rule states that if the functions $f(q), g(q): \mathbb{H} \rightarrow \mathbb{H}$ are real-differentiable, then so too is their product, that is

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

where $q^{g*} = gq^*g^{-1}$ is a quaternion rotation [21]. For the chain rule, if $g: \mathcal{S} \rightarrow \mathbb{H}$ and $f: \mathcal{T} \rightarrow \mathbb{H}$ 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

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

There has been extensive work on theory and applications of quaternion-valued adaptive filters [34]. Traditional strictly linear quaternion filters are based on the strictly linear model $\hat{y} = \mathbf{u}^H\mathbf{x}$ with an input vector $\mathbf{x} \in \mathbb{H}^{M\times 1}$, a coefficient vector $\mathbf{u} \in \mathbb{H}^{M\times 1}$, and a target signal $\hat{y} \in \mathbb{H}$. Advances in quaternion statistics have established that only widely linear quaternion filters based on the widely liner model $\hat{y} = \mathbf{w}^H\mathbf{q}$, where $\mathbf{w} \in \mathbb{H}^{4M\times 1}$ is an augmented coefficient vector, $\mathbf{q} \triangleq \left[\mathbf{x}^T, \mathbf{x}^{\imath T}, \mathbf{x}^{\jmath T}, \mathbf{x}^{\kappa T}\right]^T$, and $\mathbf{x}^{\imath}, \mathbf{x}^{\jmath}, \mathbf{x}^{\kappa}$ are rotations of $\mathbf{x}$ about the $\imath$, $\jmath$ and κ imaginary axes, exploits three complementary covariances in addition to the standard covariance, and thus captures complete second-order statistical information in quaternion signals [29], [35].

Previous works on using extrapolated gradient (or momentum) in quaternion-valued adaptive filters include those involving the conjugacy property of the descent directions [1], showing that the algorithm shares the same fast-convergence behavior as that of RLS. In some circumstances where matrix inversion should be avoided (e.g. sparse inputs), the momentum-base algorithms require a robust and stable alternative. In this paper, we propose such a stable and robust algorithm based on Nesterov's optimal momentum [37]. We then provide an analysis and compare it with other algorithms of the same kind.

III. WLQLMS with Momentum

For completeness, we first introduce the cost function of the widely linear quaternion LMS (WLQLMS) [25] algorithm, briefly describe how the momentum concept has been applied in the WLQLMS in [1], and finally propose a new WLQLMS algorithm with Nesterov's optimal momentum, named *n*-WLQLMS.

A. Problem Setting

We shall denote by $y_n, x_n \in \mathbb{H}$, $n = 1, ..., N$ the output and input signals, respectively; then, the linear mean square error (MSE) estimator of y_n, denoted by $\hat{y}_n$, can be expressed as [27], [28]

$$\hat{y}_n = \langle \hat{\mathbf{w}}, \mathbf{q}_n \rangle \triangleq \hat{\mathbf{w}}^H\mathbf{q}_n \tag{1}$$

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

$$\mathbf{q}_n = \left[\mathbf{x}_n^T,\ \mathbf{x}_n^{iT},\ \mathbf{x}_n^{jT},\ \mathbf{x}_n^{kT}\right]^T \tag{2}$$

and $\mathbf{x}_n = [x_n, x_{n-1}, ..., x_{n-M+1}]^T$ for a filter of order M. For a strictly linear model, $\mathbf{q}_n = \mathbf{x}_n$ (i.e. data is circular [26]). The goal of such an estimator is to minimize the MSE given by

$$\mathcal{J}_n(\hat{\mathbf{w}}) = E\{\|\xi_n(\hat{\mathbf{w}})\|_2^2\} \tag{3a}$$

$$\xi_n(\hat{\mathbf{w}}) = y_n - \hat{y}_n = y_n - \hat{\mathbf{w}}^H\mathbf{q}_n, \tag{3b}$$

recursively, with the weight update expressed as [21]

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{d}_n\alpha_n \tag{4}$$

where $\mathbf{d}_n$ is the descent direction, while the stepsize $\alpha_n \in \mathbb{H}$ is non-commutative and has to post-multiply $\mathbf{d}_n$ so as not to lead to Sylvester's equation which has no closed-form solution [32]. In the case of standard WLQLMS [25], eq. (3a) becomes an instantaneous estimate, that is, $\mathcal{J}_n(\hat{\mathbf{w}}) \approx \|\xi_n(\hat{\mathbf{w}})\|^2$ and the descent direction will be the conjugate gradient descent, i.e. $\mathbf{d}_n = -\mathbf{g}_{n|n-1}$ where $\mathbf{g}_{n|n-1}$ is given by

$$\mathbf{g}_{n|n-1} \triangleq \nabla_{\hat{\mathbf{w}}^*}\mathcal{J}_n(\hat{\mathbf{w}})|_{\hat{\mathbf{w}}=\mathbf{w}_{n-1}} \tag{5}$$

which as a result renders eq. (4) to become $\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{q}_n e_n^*\alpha_n$ where $\alpha_n \in \mathbb{H}$ is an adaptive stepsize and e_n the *a priori* error, defined as

$$e_n \triangleq \xi_n(\mathbf{w}_{n-1}) = y_n - \mathbf{w}_{n-1}^H\mathbf{q}_n. \tag{6}$$

To incorporate momentum into WLQLMS, the estimate of eq. (3a) has to extend beyond the most recent data which was employed in standard WLQLMS. The cost function of interest hence becomes

$$\mathcal{J}_n(\hat{\mathbf{w}}) \approx \sum_{k=1}^{n}\gamma_k\|\xi_k(\hat{\mathbf{w}})\|_2^2 \tag{7}$$

with $\gamma_k \in \mathbb{R}$ being the weight that controls the impact of each data sample from the past. One popular idea, employed in RLS-type adaptive filtering [30], [31], is that the error decreases as time goes by, so that earlier data are gradually forgotten. Mathematically,

$$\gamma_k = \lambda^{n-k} \tag{8}$$

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. With eq. (7) substituted into eq. (6), we have

$$\mathcal{J}_n(\hat{\mathbf{w}}) = \hat{\mathbf{w}}^H\mathbf{R}_n\hat{\mathbf{w}} - 2\Re\{\hat{\mathbf{w}}^H\mathbf{r}_n\} + \sum_{k=1}^{n}\lambda^{n-k}|y_k|^2 \tag{9}$$

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

$$\mathbf{R}_n \triangleq \sum_{k=1}^{n}\lambda^{n-k}\mathbf{q}_n\mathbf{q}_n^H = \lambda\mathbf{R}_{n-1} + \mathbf{q}_n\mathbf{q}_n^H, \tag{10}$$

$$\mathbf{r}_n \triangleq \sum_{k=1}^{n}\lambda^{n-k}\mathbf{q}_n y_n^* = \lambda\mathbf{r}_{n-1} + \mathbf{q}_n y_n^*. \tag{11}$$

B. Extrapolation in WLQLMS

To derive the *n*-WLQLMS algorithm, we define the extrapolated weight vector, $\mathbf{v}_n$, as

$$\mathbf{v}_n \triangleq \mathbf{w}_{n-1} + (\mathbf{w}_{n-1} - \mathbf{w}_{n-2})\beta_n \tag{12}$$

where $\beta_n > 0$ is extrapolation parameter, and take the descent direction $\mathbf{d}_n$ as a gradient descent around $\mathbf{v}_n$, that is

$$\mathbf{d}_n \triangleq \nabla_{\hat{\mathbf{w}}^*} \mathcal{J}_n(\hat{\mathbf{w}})|_{\hat{\mathbf{w}}=\mathbf{v}_n} = \mathbf{R}_n \mathbf{v}_n - \mathbf{r}_n. \tag{13}$$

The goal now is to find β_n which yields optimal performance. In the real-valued case, Nesterov's momentum is shown to give such a result [37]. Although not yet proven to produce the same result for quaternions, we believe that this approach provides a potential alternative to this still-emerging class of algorithms.

To simplify the problem, we consider β_n to be real-valued so that Nesterov's formulae can be straightforwardly applied to quaternions as follows (for more detail, we refer to [37])

$$\beta_n = \frac{\mu_{n-1}(1-\mu_{n-1})}{(\mu_{n-1})^2 + \mu_n} \tag{14}$$

with $\mu_n \in (0,1)$, such that

$$(\mu_n)^2 = (1-\mu_n)(\mu_{n-1})^2 + \left(\frac{\sigma_n}{L_n}\right)\mu_n \tag{15}$$

where

$$\sigma_n = \|\lambda_{\min}(\mathbf{R}_n)\|_2 \tag{16}$$

and

$$L_n = \|\lambda_{\max}(\mathbf{R}_n)\|_2. \tag{17}$$

The $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$, respectively, represent the minimum and maximum eigenvalues.

Remark 1. *The key difference between m-WLQLMS in [1] and n-WLQLMS proposed in this work is where momentum is applied in the algorithm. In m-WLQLMS, the descent direction is extrapolated, while in n-WLQLMS, extrapolation is performed for the weight vector.*

C. Sketch of Convergence

The rationale behind convergence analyses is to ensure that the cost function decreases monotonically with time in a statistical sense. In our problem, while certainly $\mathcal{J}_n(\mathbf{w}_n) < \mathcal{J}_n(\mathbf{v}_n)$, it is not guaranteed that $\mathcal{J}_n(\mathbf{w}_n) < \mathcal{J}_n(\mathbf{w}_{n-1})$ because the effect of extrapolation could lead to an underdamped trajectory [36]. Even for the non-stochastic function, the analysis is still limited, not to mention the stochastic case. One simple way to preclude this underdamping is to reset β_n whenever $\mathcal{J}_n(\mathbf{w}_n) \geq \mathcal{J}_n(\mathbf{w}_{n-1})$. Some heuristic methods in this direction do exist, like the *gradient check* [38]:

$$\langle \mathbf{w}_n - \mathbf{v}_n, \mathbf{w}_n - \mathbf{w}_{n-1} \rangle < 0 \tag{18}$$

and contraction mapping [39]:

$$\|\mathbf{w}_n - \mathbf{v}_n\|_2 < \epsilon_0 \|\mathbf{w}_{n-1} - \mathbf{v}_{n-1}\|_2.$$

The equation above can be rewritten by exploiting the cost in eq. (9), to yield the following Remark.

Remark 2. *For the stepsize $\alpha_n < 1/L_n$, the condition $(\epsilon_0/2)\|\mathbf{w}_n - \mathbf{w}_{n-1}\|^2 < \mathcal{J}_n(\mathbf{w}_{n-1}) - \mathcal{J}_n(\mathbf{w}_n)$ is satisfied if*

$$\beta_n < \sqrt{\frac{1-\alpha_n\epsilon_0}{1-\alpha_n\sigma_n}} \frac{\|\mathbf{w}_n - \mathbf{w}_{n-1}\|_2}{\|\mathbf{w}_{n-1} - \mathbf{w}_{n-2}\|_2} \tag{19}$$

where ϵ_0 is a small real number in the range (σ_n, L_n), $\forall n$.

Hence, when eqs. (18) and (19) are violated, we restart the extrapolation by setting β_n to zero. The proposed algorithm, termed the *n*-WLQLMS, is summarized in Algorithm 1 below ($\delta > 0$ is used as a stopping criterion),

Algorithm 1: the *n*-WLQLMS algorithm

Input : x_n, y_n, M, λ and δ
Output: $\mathbf{w}$
Initialize $\mathbf{w}_0 = \mathbf{p}_0 = \mathbf{0}$ and $\mathbf{R}_0 = \underline{\mathbf{0}}$;
$n = 0$;
do
  $n = n + 1$;
  Update $\mathbf{q}_n$ according to (2);
  $\hat{y}_n - (\mathbf{w}_{n-1})^H \mathbf{q}_n$;
  $e_n = y_n - \hat{y}_n$;
  $\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \omega_n \mathbf{q}_n \mathbf{q}_n^H$;
  $\mathbf{p}_n = \lambda \mathbf{p}_{n-1} + \omega_n \mathbf{q}_n y_n^*$;
  $\sigma_n = \lambda_{\min}(\mathbf{R}_n)$ (smallest eigenvalue);
  $L_n = \lambda_{\max}(\mathbf{R}_n)$ (largest eigenvalue);
  $\alpha_n = \frac{1}{L_n}$;
  Find $\mu_n \in (0,1)$ such that
  $(\mu_n)^2 = (1-\mu_n)(\mu_{n-1})^2 + \left(\frac{\sigma_n}{L_n}\right)\mu_n$;
  $\beta_n = \frac{\mu_{n-1}(1-\mu_{n-1})}{(\mu_{n-1})^2+\mu_n}$;
  $\mathbf{v}_n = \mathbf{w}_{n-1} + \beta_n(\mathbf{w}_{n-1} - \mathbf{w}_{n-2})$;
  $\mathbf{w}_n = \mathbf{v}_n - \frac{1}{L_n}(\mathbf{R}_n \mathbf{v}_n - \mathbf{p}_n)$;
  $\epsilon_0 = \frac{\sigma_n}{10}$;
  if $\langle \mathbf{w}_n - \mathbf{v}_n, \mathbf{w}_n - \mathbf{w}_{n-1} \rangle < 0$ *or*
  $\beta_n \geq \sqrt{\frac{1-\alpha_n\epsilon_0}{1-\alpha_n\sigma_n}} \frac{\|\mathbf{w}_n - \mathbf{w}_{n-1}\|_2}{\|\mathbf{w}_{n-1} - \mathbf{w}_{n-2}\|_2}$ **then**
    $\mu_n = 1$;
    $\mathbf{w}_n = \mathbf{w}_{n-1} - \frac{1}{L_n}(\mathbf{R}_n \mathbf{w}_{n-1} - \mathbf{p}_n)$;
  end
while $\|e_n\| > \delta$ or $n \leq N$;
$\mathbf{w}^{op} = \mathbf{w}_n$.

IV. Numerical Experiments

Simulations were conducted in MATLAB to test the *n*-WLQLMS against other algorithms of akind. In the first experiment, a widely linear quaternion moving average filter of order 3, WLQMA(3), was run over 200 independent trials, where the input x_n was drawn from 400 samples of the distribution $\mathcal{N}(0,1)$ for each component of x_n, with a SNR of 40dB. The second experiment considers $\mathbb{Q}$-improper (noncircular) non-stationary signals generated based on Saito's chaotic signal. Besides the *a priori* error, another metric used to compare the algorithms was the misalignment, η_n, defined as

$$\eta_n \triangleq \frac{\|\mathbf{w}_n - \mathbf{w}^{op}\|_2^2}{\|\mathbf{w}^{op}\|_2^2}. \tag{20}$$

In the first experiment, the algorithm performance was considered for different values of λ. As seen in Fig. 1, both momentum-based algorithms, *m*- and *n*-WLQLMS, converged as fast and achieved steady-state misalignment as low as that of widely linear quaternion RLS (WLQRLS). At $\lambda = 0.99$ and 0.91, the two algorithms behaved on par with WLQRLS, achieving similar convergence rate and steady-state misalignment. However, for $\lambda = 0.91$, WLQRLS could not maintain stability and diverged, while the other two algorithms converged. This divergence was particularly obvious for smaller values of λ. In other words, the *m*- and *n*-WLQLMS were more numerically stable, owing to the absence of matrix inversion.

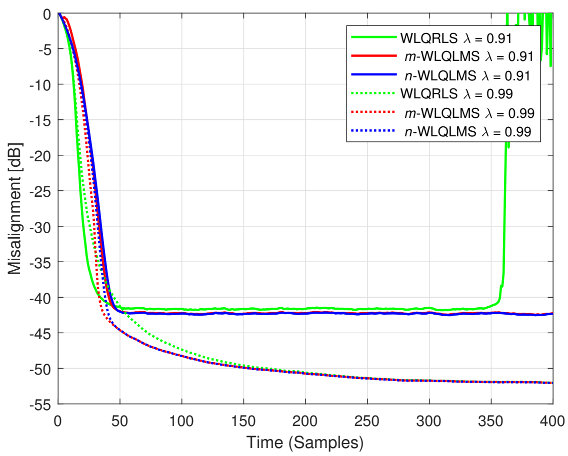

Fig. 1: Misalignment of WLQRLS, *m*-WLQLMS and *n*-WLQLMS algorithms for different λ, averaged over 200 independent trials, when employed for the identification of a WLQMA(3) process, at an SNR of 40dB.

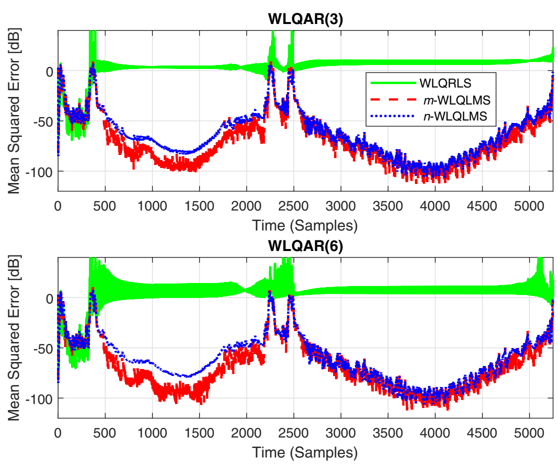

Fig. 2: MSE of WLQRLS, *m*-WLQLMS and *n*-WLQLMS algorithms with $\lambda = 0.95$, when employed for the identification of Saito's circuit through WLQAR(3) and WLQAR(6) models.

The numerical instability of WLQRLS was even more exposed in the second experiment where Saito's signals were modelled by the widely linear quaternion autoregressive (WLQAR) process of different orders in a 3-step prediction setting. Fig. 2 shows that when the system was non-stationary (spikes in MSE), the WLQRLS struggled to re-adjust its weights to converge; intuitively, due to the inherent matrix inversion, the forgetting factor, instead of 'forgetting', kept accumulating past erroneous data, leading to divergence. The momentum-based algorithms, however, converged for both WLQAR(3) and WLQAR(6) models. At the steady state, *m*-WLQLMS performed better than *n*-WLQLMS. Fig. 3 illustrates the performance of WLQAR models of orders from 3 to 6 in the transient state. Observe that an increase in the model order did not significantly increase accuracy of prediction, while *m*-WLQLMS still performed slightly better. Fig. 4 compares the original Saito's signal with its estimates. In the transient state, when compared to *m*-WLQLMS, *n*-WLQLMS fluctuated less at the point of abrupt change but was more prone to overestimation thereafter, hence yielding a larger overall error.

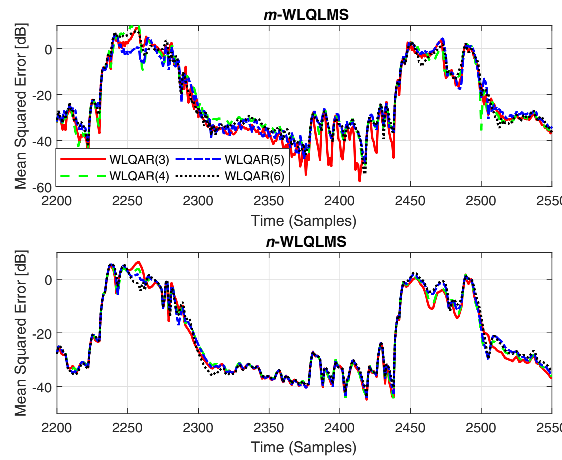

Fig. 3: Transient MSE of *m*-WLQLMS and *n*-WLQLMS with $\lambda = 0.95$, when employed for the identification of Saito's circuit through WLQAR models of orders from 3 to 6.

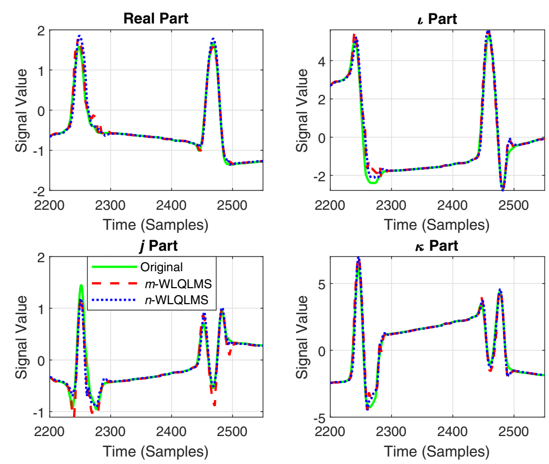

Fig. 4: Values of each component of Saito's signal presented in the original (green) and its estimates through WLQAR(3) in *m*-WLQLMS (red dash) and *m*-WLQLMS (blue dot) algorithms with $\lambda = 0.95$.

V. Conclusion

A new quaternion-valued adaptive filtering algorithm has been proposed which exploits Nesterov's extrapolated gradient methods to accelerate convergence rate of the system in a manner similar to the RLS method but more numerically stable, due to the absence of matrix inversion. This can be useful for non-vector, sparse or non-stationary signals. The experiments have confirmed its advantages over WLQRLS for non-stationary processes where WLQRLS diverges after the state of the system changed but the proposed method converges.

REFERENCES

[1] T. Variddhisaï and D. P. Mandic, "On an RLS-like LMS adaptive filter," in *Proc. 22nd Int. Conf. Digital Signal Process. (DSP)*, London, 2017, pp. 1-5.

[2] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ: Wiley-IEEE Press, 2008.

[3] D. Bismor, K. Czy, Z. Ogonowski, "Review and comparison of variable step-size LMS algorithms," *Int. J. Acoust. Vibra.*, vol. 21, pp. 24-39, 2016.

[4] A. I. Hanna, I. R. Krcmar and D. P. Mandic, "Perlustration of error surfaces for nonlinear stochastic gradient descent algorithms," in *Proc. 6th Seminar on Neural Network Applications in Electrical Engineering*, Belgrade, 2002, pp. 11-16.

[5] S. C. Douglas, "Generalized gradient adaptive step sizes for stochastic gradient adaptive filters," in Proc. *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Detroit, MI, USA, vol.2, pp. 1396-1399.

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

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

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

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

[10] S. K. Zhao, Z. H. Man and S. Y. Khoo, "Conjugate gradient algorithm design with RLS normal equation," in *Proc. 6th Int. Conf. Inf. Comm. Signal Process.*, Singapore, 2007, pp. 1-5.

[11] 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 Process. Mag.*, vol. 32, no. 6, pp. 117-122, Nov. 2015.

[12] T. Rabbani, *Automatic reconstruction of industrial installations using point clouds and images*, Netherlands Geodetic Commission, 2006.

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

[14] J. W. Tao and W. X. Chang, "A novel combined beamformer based on hypercomplex processes," *IEEE Trans. Aero. Electron. Sys.*, vol. 49, no. 2, pp. 1276-1289, Apr. 2013.

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

[16] B. Chen, H. Shu, G. Coatrieux, G. Chen, X. Sun, and J. L. Coatrieux, "Color image analysis by quaternion-type moments." *J. Math. Imag. Vis.*, vol. 51, no. 1 , pp. 124-144, Jan. 2015.

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

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

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

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

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

[22] D. Xu, Y. Xia and D. P. Mandic, "Optimization in quaternion dynamic systems: Gradient, Hessian, and learning algorithms," *IEEE Trans. Neural Net. Learn. Sys.*, vol. 27, no. 2, pp. 249-261, Feb. 2016.

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

[24] T. A. Ell, S. J. Sangwine, "Quaternion involutions and anti-involutions," *Comp. & Math. Appli.*, vol. 53, issue 1, pp. 137-143, 2007.

[25] 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.

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

[27] M. Xiang, S. Enshaeifar, A. E. Stott, C. C. Took, Y. Xia, S. Kanna, D. P. Mandic, "Simultaneous diagonalisation of the covariance and complementary covariance matrices in quaternion widely linear signal processing," *Signal Process.*, vol. 148, pp 193-204, 2018.

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

[29] C. Took and D. P. Mandic, "A quaternion widely linear adaptive filter," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4427-4431, Aug 2010

[30] C. Jahanchahi, C. Cheong Took and D. P. Mandic, "The widely linear quaternion recursive least squares filter," in Proc. *2nd Int. Workshop Cog. Inf. Process.*, pp. 87-92, Elba, 2010.

[31] S. C. Douglas, "Widely-linear recursive least-squares algorithm for adaptive beamforming," in Proc. *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Taipei, pp. 2041-2044, 2009.

[32] V. S Shpakivskyi, "Linear quaternionic equations and their systems," *Adv. Appl. Clifford Alg.*, vol. 21, issue 3, pp. 637-645, 2011.

[33] 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 Trans. Signal Process.*, vol. 64, no. 24, pp. 6428-6442, Dec. 2016.

[34] M. Xiang, S. Kanna, and D. P. Mandic, "Performance analysis of quaternion-valued adaptive filters in nonstationary environments," *IEEE Trans. Signal Process.*, vol. 66, pp. 1566–1579, Mar. 2018.

[35] M. Xiang, S. Kanna, S. C. Douglas, and D. P. Mandic, "Performance advantage of quaternion widely linear estimation: An approximate uncorrelating transform approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 4348–4352, 2016.

[36] Y.E. Nesterov, "Gradient Methods for Minimizing Composite Objective Function," *CORE Discussion Papers*, No. 2007076, Universitè catholique de Louvain, Center for Operations Research and Econometrics (CORE),

[37] Y.E. Nesterov, "A method for solving the convex programming problem with Convergence Rate O(1/k2)," *Soviet Mathematics Doklady*, no. 27, pp. 372-376, 1983.

[38] B. O'Donoghue and E. Candés, "Adaptive restart for accelerated gradient schemes," *Found. Comput. Math.*, vol. 15, no. 3, pp. 715-732, Jun. 2015.

[39] D. P. Mandic and J. Chambers. 2001. *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. John Wiley & Sons, Inc., New York, NY, USA.