A family of quaternion-valued pipelined second-order Volterra adaptive filters for nonlinear system identification

Qianqian Liu · Yigang He

Received: 8 June 2021 / Accepted: 3 April 2022 / Published online: 18 April 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract This paper primarily proposes a family of quaternion Volterra filters based on the feedforward pipelined structure (QPSOVAFs) for nonlinear quaternion system identification to reduce the computational complexity. Then, the strictly nonlinear QPSOVAF (SNL-QPSOVAF), semi-widely nonlinear QPSOVAF (SWNL-QPSOVAF) and widely nonlinear QPSOVAF (WNL-QPSOVAF) are proposed. This architecture consists of several quaternion-valued second-order Volterra (SOV) modules. The structure’s nonlinear subsection executes a nonlinear mapping from the input space to an intermediate space using the feedforward SOV; the linear combiner subsection performs a linear mapping from the intermediate space to the output space. Moreover, the theoretical analysis expresses the effectiveness of the proposed QPSOVAFs in a specific condition. Finally, simulation results further prove that the proposed QPSOVAFs have good performance in identifying the quaternion-valued nonlinear system.

Keywords Nonlinear quaternion system · Pipelined structure · Quaternion-valued SOV · Widely nonlinear · Strictly nonlinear · Semi-widely nonlinear

1 Introduction

The Volterra filter is widely applied to process nonlinear problems [1–6]. It was usually used in many nonlinear applications such as system identification [7,8], acoustic echo cancellation [9], speech prediction [10], biomedical engineering [11], channel equalization [12] and digital communication channels [13]. However, the computational complexity of the Volterra filter (VF) will exponentially increase with the order and memory (or delay) of the filter increasing [14], which significantly limits the practical applications of the VF. Then, the complex-valued SOV proposed to characterize the complex-valued nonlinearity, which has used to many applications such as nonlinear amplifier modelling [15], beam-forming [16,17], Nyquist subcarrier modulation [18] and transmitter modelling [19]. However, neither the real model nor the complex-valued model could be effectively used to model nonlinear 3-D and 4-D signals.

Since quaternion algebra can effectively avert some important attributes of numerical issues for vector algebra, it has been markedly applied in image processing [20], body motion tracking [21,22], land classification [23], computer graphing [24], adaptive filtering [25–32] and other application areas [28,33–36]. The Hamilton-real (HR) calculus [29], the augmented statistics [25] and generalized Hamilton-real (GHR) calculus [31], which provide a great model for the interrelation between observations expressed by a quater-
nion, have paved the way for recently researching the theory and application of quaternion-valued adaptive filter [21,27–30,32]. For instance, the classical strictly linear (SL) quaternion-valued adaptive filter model [21,27–29,32] was proposed to process the quaternion-valued signals, which are second-order circular, by using the standard covariance information. On the other hand, the widely linear (WL) quaternion-valued adaptive filter model can process and model the quaternion-valued signals, which are second-order non-circular, by using three pseudo-covariance matrices [21,27–29,32]. Moreover, it is well known that the SWL model is the simplified version of the WL model, which can capture the unique second-order statistical characteristics of quaternion-valued signals [32,37,38].

Recently, the quaternion-valued SOV adaptive filters (QSOVAFs) were provided [39], which can efficiently process the nonlinear signals. However, it has a high computational burden and a low convergence speed in the identification system. To solve the high computational cost of quaternion-valued SOV, we introduce a novel pipelined architecture. This idea is inspired by the efficient pipelined structure [40,41] and later was improved by other researcher [42] leading to the novel adaptive joint process filter using pipelined feedforward second-order Volterra architecture (JPPSOV). The merits of the pipelined architecture are that the pipelined realization modularity and such realizations use less computational cost to approximate the nonlinear system effectively. However, the traditional pipelined structure ignores the importance of the signals closer to the current moment, leading to the poor performance. Therefore, we present a novel pipelined structure and then propose a quaternion-valued second-order Volterra filter family based on the novel pipelined structure and later was improved by other researcher [42] leading to better performance.

In this study, we mainly have the following contributions:

1. To overcome the drawback of the architecture of the JPPSOV [42], we proposed a novel structure of QPSOVAF by maximizing use of the information of the signals closer to the current moment, which leads to better performance.

2. Adaptive algorithms for QPSOV are derived via using the generalized Hamilton-real (GHR) calculus. For the strictly nonlinear (SNL), semi-widely nonlinear (SWNL) and widely nonlinear (WNL) QPSOVAFs, we propose novel weight update equations of the SNL-QPSOVAF, SWNL-QPSOVAF and WNL-QPSOVAF, respectively, and the simulation proves that QPSOVAFs could better deal with nonlinear identification system. Compared with the conventional JPPSOV which cannot be applied to the modelling nonlinearity of 3-D and 4-D signals, the proposed QSOVAFs enable the modelling of nonlinear 3-D and 4-D signals in the quaternion domain and have merit pipelined architecture.

3. This study provides the convergence analysis and the mean square steady-state analysis to obtain the stable value range of the step size and the steady-state performance of the proposed QPSOVAFs.

In a word, this paper tries to implement low computational complexity SNL-QPSOV algorithm (will be emphatically studied), SWNL-QPSOV algorithm and WNL-QPSOV algorithm in the quaternion domain. The rest of this article is listed as follows. Section 2 reviews the quaternion theory. Sections 3 and 4 derive the proposed QPSOVAFs. In sect. 5, we give the performance analysis. Section 6 introduces the simulations. Finally, Sect. 7 summarizes this work.

2 The quaternion theory

2.1 Basic properties of quaternion algebra

The quaternion is a non-commutative extension of the complex number, i.e. it does not conform to the commutative law. For example, define the quaternion variable $s$ as follows

\[ s = s_a + i s_b + j s_c + k s_d \]

where let $\Re(s) = s_a$ represents the real part and $\Im(s) = i s_b + j s_c + k s_d$ denotes the imaginary part with $s_a, s_b, s_c$ and $s_d$ being belong to the real-valued. \{1, i, j, k\} denotes a basis of the quaternion variable $s$ and satisfies $i^2 = j^2 = k^2 = i j k = -1$, where $j k = i = - k j, i j = k = - j i, k i = j = - i k$ and $j i \neq i j = - k$. Define the conjugate of $s$ as $s^\ast = s_a - i s_b - j s_c - k s_d$ and satisfies $(q s)^\ast = s^\ast q^\ast$. The quaternion variable $s$ has the following properties: (1) Defining its modulus and inverse operations as $|s| = \sqrt{s s^\ast}, |q s| = |q| |s|, s^{-1} = s^\ast / |s|^2$, and $(q s)^{-1} = s^{-1} q^{-1}$, where $s \neq 0$; (2) The polar form: $s = |s| (\cos \theta + \hat{s} \sin \theta)$, where $\theta = \arccos (\Re(s) / |s|) \in \mathbb{R}$.
\( R \) is a angle and \( s = \Im / |\Im| \) is a pure unit quaternion; (3) \( s \) has rotation and involution properties. For the property (3), we define the 3-D rotation of the vector part of \( s \) by an angle \( 2\theta \) about the vector part of \( \mu \) as \([31]\)

\[
s^\mu = \mu s s^* s^{-1} \tag{2}\]

where \( \mu = |\mu| (\cos \theta + \tilde{\mu} \sin \theta) \) represents a nonzero quaternion. (2) becomes a quaternion involution for a pure unit quaternion \([43,44]\). Similarly, \( s^i, s^j \) and \( s^k \) can be defined by

\[
s^i = -is_i = sa + is_b - js_c - ks_d \]

\[
s^j = js_j = sa - is_b + js_c + ks_d \]

\[
s^k = -ks_k = sa - is_b - js_c + ks_d \tag{3}\]

The other properties of quaternion rotation is listed as follows

\[
s^\mu = s^{(\overline{\mu})}, (qs)^\mu = q^{*} s^\mu, qs = qs \]

\[
\mu^\mu = (s^*)^\mu, s^{\mu*} \triangleq (s^*)^\mu = (s^\mu)^* \triangleq s^{\mu*} \tag{4}\]

where \( \mu \) is not necessarily a unit quaternion. However, \( \mu / |\mu| = 1 \) is an unit quaternion. For more properties and details about the quaternion algebra can be looked for \([29,43]\).

### 2.2 GHR calculus

Recently, \([29]\) proposed HR calculus to decrease the high computational complexity of classical quaternion pseudo-derivatives. However, it does not include the traditional product and chain rules. Therefore, \([31]\) provided GHR with novel product and chain rules. Then we briefly introduce some necessary attributes of the GHR calculus, as shown below. Firstly, let \( Y : \mathbb{H} \to \mathbb{H} \), then, respectively, defining the left GHR derivatives of \( Y(s) \) with respect to \( s^\mu \) and \( s^{\mu*} \) as

\[
\frac{\partial Y}{\partial s^\mu} = \frac{1}{4} \left( \frac{\partial Y}{\partial s_a} i^\mu + \frac{\partial Y}{\partial s_b} j^\mu + \frac{\partial Y}{\partial s_c} k^\mu - \frac{\partial Y}{\partial s_d} j^\mu \right) \in \mathbb{H} \tag{5}\]

\[
\frac{\partial Y}{\partial s^{\mu*}} = \frac{1}{4} \left( \frac{\partial Y}{\partial s_a} i^{\mu*} + \frac{\partial Y}{\partial s_b} j^{\mu*} + \frac{\partial Y}{\partial s_c} k^{\mu*} + \frac{\partial Y}{\partial s_d} k^{\mu*} \right) \in \mathbb{H} \]

where \( \partial Y / \partial s_\varphi \in \mathbb{H} \) represents the partial derivatives of \( Y(s) \) with respect to \( s_\varphi \), \( (Q = a, b, c, d) \). Moreover, other some attributes of the left GHR derivatives are defined as \([31]\)

Conjugate rules:

\[
\frac{\partial Y^*}{\partial s^\mu} = \frac{\partial Y^*}{\partial s^{\mu*}}, \quad \frac{\partial Y^*}{\partial s^{\mu*}} = \frac{\partial Y^*}{\partial s^\mu} \tag{6}\]

If \( Y \) is a real function, then

\[
\frac{\partial Y^*}{\partial s^\mu} = \frac{\partial Y^*}{\partial s^{\mu*}} = \frac{\partial Y}{\partial s^\mu} \tag{7}\]

Product rules:

\[
\frac{\partial (YZ)}{\partial s^\mu} = Y \frac{\partial Z}{\partial s^\mu} + \frac{\partial Y}{\partial s^\mu} Z \tag{8}\]

\[
\frac{\partial (YZ)}{\partial s^{\mu*}} = Y \frac{\partial Z}{\partial s^{\mu*}} + \frac{\partial Y}{\partial s^{\mu*}} Z.
\]

Since the proposed series of QPSOVAF algorithms mainly use the left GHR calculus attributes above in this paper, for more details about GHR calculus, please refer to \([29,31]\).

### 2.3 WL model and SWL model

Usually, \( \mathbf{R} = E\{xx^H\} \) is defined as the classical covariance matrix to evaluate the second-order circular signals, where \( \mathbf{x} \) is a random vector, and \( (\cdot)^H \) denotes the Hermitian transpose. However, in practical application, we often encounter second-order non-circular signals, which are no longer sufficient to process by the rules mentioned above of circular signals. Thus, we also need to evaluate the pseudo-covariance matrices, i.e., \( \mathbf{P} = E\{xx^H\}, \mathbf{S} = E\{xx^H\} \) and \( \mathbf{T} = E\{xx^H\} \).

To this end, the quaternion-valued WL modelling is given by \([26]\)

\[
y = h^T x^a = w^T x + g^T x^i + u^T x^j + v^T x^k \tag{9}\]

where \( y \) denotes the filter output, \( h^a = \{w^T, g^T, u^T, v^T\}^T \) and \( x^a = \{x^T, x^iT, x^jT, x^kT\}^T \) are the augmented weight vector and the augmented input vector, respectively. Therefore, combining the three pseudo-covariance matrices and covariance matrix, the augmented covariance matrix \( \mathbf{R}^a = E\{x^a x^a^H\} \) can be expressed as \([26]\)

\[
\mathbf{R}^a = \begin{bmatrix} \mathbf{R} & \mathbf{P} & \mathbf{S} & \mathbf{T} \\ \mathbf{P}^i & \mathbf{R}^i & \mathbf{T}^i & \mathbf{S}^i \\ \mathbf{S}^j & \mathbf{T}^j & \mathbf{R}^j & \mathbf{P}^j \\ \mathbf{T}^k & \mathbf{S}^k & \mathbf{P}^k & \mathbf{R}^k \end{bmatrix} \tag{10}\]
Note that the augmented covariance matrix applies to second-order non-circular signals. If the quaternion-valued signal is a circular signal, then $P$, $S$ and $T$ will disappear. In addition, simplifying WL modelling to SWL modelling to obtain the special quaternion-valued signals’ second-order statistical characteristics, Eq. (9) will become

$$y = h^T x^a = w^T x + g^T x^i$$

(11)

where the augmented weight vector of semi-widely linear modelling is $h^a = [w^T, g^T]^T$, and the augmented input vector is $x^a = [x^T, x^i]^T$. In addition, the augmented covariance matrix of semi-widely linear modelling is \[37, 38\]

$$R^a = \begin{bmatrix} R & P \\ P^i & R^i \end{bmatrix}$$

(12)

Note that the pseudo-covariance matrix $P$ will disappear for a circular second-order quaternion-valued signal, just like widely linear modelling.

3 The quaternion-valued pipelined SOV adaptive filters

3.1 Review of the quaternion-valued SOV

In the quaternion-valued domain, the classic Volterra series can be written as

$$y(m) = h_0 + \sum_{n_1=0}^{N-1} h_1(n_1) x(m-n_1)$$

\[N-1\]

$$+ \sum_{n_1=0}^{N-1} \sum_{n_2=n_1}^{N-1} h_2(n_1, n_2) x(m-n_1) x(m-n_2)$$

\[N-1\]

$$+ \cdots + \sum_{n_1=0}^{N-1} \cdots \sum_{n_p=n_{p-1}}^{N-1} h_p(n_1, \ldots, n_p)$$

\[N-1\]

$$x(m-n_1) \cdots x(m-n_p)$$

\( (13) \)

where $y(m) \in \mathbb{H}$ and $h_p(n_1, \ldots, n_p) \in \mathbb{H}$ are the output and the set of $p$th-order kernel coefficients of the quaternion-valued Volterra, respectively. $h_0$ denotes the zeroth-order quaternion-valued kernel constant, and $N$ denotes the system memory size. Considering the high computational complexity of the Volterra series model in (13), we mainly focus on the SOV architecture for identifying the nonlinear system in this paper. Therefore, define the output of quaternion-valued SOV as

$$y(m) = h_0 + \sum_{n_1=0}^{N-1} h_1(n_1) x(m-n_1)$$

\[N-1\]

$$+ \sum_{n_1=0}^{N-1} \sum_{n_2=n_1}^{N-1} h_2(n_1, n_2)$$

\[N-1\]

$$x(m-n_1) x(m-n_2)$$

\( (14) \)

where $h_1(n_1)$ is the first-order quaternion-valued kernel coefficients and $h_2(n_1, n_2)$ is the second-order kernel coefficients. Equation (14) can also be rewritten as a vector form, as follows

$$y(m) = h^T(m)x(m)$$

\( (15) \)

where $h(m)$ and $x(m)$ are the expanded weight vector and the input vector of QSOV, respectively. They are defined as

$$h(m) = [h_0, h_1(0), \ldots, h_2(0, 0), h_2(0, 1),$$

\( \ldots, h_2(N-1, N-1)]^T$$

\( (16) \)

$$x(m) = [1, x(m), \ldots, x(m-N+1),$$

\( x^2(m), \ldots, x^2(m-N+1)]^T$$

\( (17) \)

where the length of $x(m)$ is expressed as \[45\]

$$L = 1 + N + \frac{N(N+1)}{2}$$

\( (18) \)

3.2 The novel QPSOVAF structure

Though the quaternion-valued SOV have good performance, its implementation to treat nonlinear cases would be unfeasible, as it would require an additional exponential increase in the number of coefficients. Therefore, we study the novel QPSOV filter structure in this subsection.

As demonstrated in Fig. 1, the nonlinear subsection of the proposed novel QPSOV filter consists of a certain number of $M$ identical feedforward SOV modules. Moreover, the output of the nonlinear subsection constitutes the linear subsection input and use the transversal filter to represent the linear filter in this paper. This combination structure of nonlinear and linear processes can effectively extract the nonlinear and linear relationships. More details about the proposed novel QPSOV filter architecture will be discussed in the following.
3.2.1 Nonlinear subsection

We can also clearly see from Fig. 1 that each module has two input signals: One is an external input signal, and the other is the output signal of the previous module. Since applying the same modules in this structure, they have the same number of input signals and similar operations. Besides, the synaptic weight matrix of all the modules of the proposed QPSOVAFs are set the same. Figure 2 shows the details of the $r$th module.

First, define the input vector $X_r(m)$ of the $r$th module as

$$X_r(m) = [x(m - r - 1), y_{r-1}(m)]^T$$

where $y_{r-1}(m)$ represents the previous module output, and $1 \leq r < M$. However, if $r = 1$, then $X_r(m)$ is expressed as

$$X_1(m) = [x(m - 1), x(m - 2)]^T$$

Then, the $r$th module input vector $X_r(m)$ is extended to $U_r(m)$ by the SOV series and is written as

$$U_r(m) = [1, X_{r,1}(m), X_{r,2}(m), X_{r,3}(m), X_{r+1,1}(m), X_{r+1,2}(m)]^T = [1, x(m - r - 1), y_{r-1}(m), x^2(m - r - 1), x(m - r - 1)y_{r-1}(m), y_{r-1}^2(m)]^T$$

![Fig. 1 Novel architecture of QPSOVAFs](image1)

![Fig. 2 Detailed construction of the $r$th module of QPSOVAFs](image2)

![Fig. 3 Architecture of JPPSOV](image3)
where the length of $U_r(m)$ is given by Eq. (18), i.e. $L = 1 + 2 + 2(2 + 1)/2 = 6$. Since the weight and rules of all modules are identical, the synaptic weight vector of the SOV series of each module is expressed by $H(m)$, whose length is also $L$, and written as

$$H(m) = [h_1(m), h_2(m), \ldots, h_L(m)]^T.$$  

Therefore, the $r$th module output can be represented by

$$y_r(m) = H^T(m)U_r(m).$$  \hspace{1cm} (23)

3.2.2 Linear subsection

In Fig. 1, each module’s output connects a conventional transversal filter and its weight vector can be expressed as

$$W(m) = [w_1(m), w_2(m), \ldots, w_M(m)]^T$$  \hspace{1cm} (24)

where $M$ represents the number of SOV modules in structure. The linear filter input consists of the present output $y_r(m)$ by each module, as follows

$$\hat{y}(m) = [y_1(m), y_2(m), \ldots, y_M(m)]^T.$$  \hspace{1cm} (25)

The output of the $M$ cascaded modules forms an $M \times 1$ output vector $\hat{y}(m)$ in (25). Therefore, the linear filter output $y(m)$ can be written as

$$y(m) = W^T(m)\hat{y}(m)$$  \hspace{1cm} (26)

where $y(m)$ is a estimated value of the practical desired sample $d(m - 1)$ or a predicted value of the practical desired sample $d(m)$. Compared with the nonlinear part, the storage space of the linear filter is limited.

3.2.3 Comparison with the traditional structure of JPPSOV

The block diagram of the JPPSOV [42] is depicted in Fig. 3. First, Fig. 3 shows that the JPPSOV uses the modular output feedback as input in final modular, i.e. final modular in essence is bilinear modular, which results in slow convergence speed and latent instability as same as adaptive bilinear filters even if the measurement noise is the Gaussian process. The proposed novel PSOV use previous modular output as input in final modular, which improve the stability of nonlinear adaptive filter. Second, the number of the sub-module of the proposed novel pipelined Volterra structure in Fig. 1 can easily vary while the conventional pipelined Volterra structure cannot. Therefore, the proposed novel pipelined Volterra structure is more flexible than the pipelined Volterra structure in [42], which is important for the practice. Lastly, from Fig. 3, the output of module $M$, $y_M(m)$, is related to the previous output of module $M$, $y_M(m - 1)$, and the input $x(m - M)$, i.e. $y_M(m) = f_M(y_M(m - 1), x(m - M))$. Then, the output of module $M - 1$, $y_{M-1}(m)$, can be expressed as $y_{M-1}(m) = f_{M-1}(y_M(m), x(m - M + 1)) = f_{M-1}(f_M(y_M(m), x(m - M - 1), x(m - M + 1))$. In other words, $y_{M-1}(m)$ is related to $y_M(m - 1)$, $x(m - M)$ and $x(m - M + 1)$ for the JPPSOV filter. Similarly, we can easily get that $y_1(m)$ is related to $y_M(m - 1)$, $x(m - M)$, $x(m - M + 1)$, and $x(m - 2)$ and $x(m - 1)$. In turn, $x(m - 1)$ only impacts $y_1(m)$ while $x(m - M)$ impacts $y_1(m), \ldots, y_M(m)$, which slows down the convergence speed because signals closer to the current moment tend to be more important. The detailed relationships between $\{y_1(m), y_2(m), \ldots, y_M(m)\}$ and $\{x(m - 1), x(m - 2), \ldots, x(m - M)\}$ in the JPPSOV is given in Fig. 4. Using the same methods of analysis, for the novel pipelined structure of QPSOVAFs, $x(m - M - 1)$ only impacts $y_M(m)$ while $x(m - 1)$ impacts $y_1(m), \ldots, y_M(m)$. Therefore, the novel pipelined structure can improve the performance of QPSOVAFs. Figure 5 depicts the detailed relationships between $\{y_1(m), y_2(m), \ldots, y_M(m)\}$ and $\{x(m - 1), x(m - 2), \ldots, x(m - M)\}$ in the QPSOVAFs.

4 Adaptive algorithms of the proposed QPSOVAFs

This section uses the pipelined filter structure to effectively extract the nonlinear and linear information. To be more specific, the input signal is linearized in the nonlinear subsection. These linearized data then feed into a linear combiner used the corresponding adaptive algorithm to generate the estimated value of the original desired signal.

Here, the cost function of the algorithm is defined by

$$J(m) = |e(m)|^2 = e^*(m)e(m)$$  \hspace{1cm} (27)

where the quaternion-valued error signal $e(m)$ is given by

$$e(m) = d(m - 1) - y(m) = d(m - 1) - W^T(m)\hat{y}(m)$$  \hspace{1cm} (28)
4.1 Derivation of the SNL-QPSOVAF

4.1.1 Linear subsection

We use the product law of the left GHR calculus given in (8) to obtain the derivation of \( J(m) \) with respect to the weight vector \( W(m) \), as shown below

\[
\frac{\partial J(m)}{\partial W(m)} = e^*(m) \frac{\partial W(m)}{\partial W(m)} + \frac{\partial e^*(m)}{\partial W(m)e(m)}e(m) \quad (29)
\]

where the above two partial derivations are obtained by

\[
\frac{\partial e(m)}{\partial W(m)} = \partial \left( d(m - 1) - W^T(m)\hat{y}(m) \right) = -\Re(\hat{y}^T(m)) \quad (30)
\]

\[
\frac{\partial e^*(m)}{\partial W(m)e(m)}e(m) = \frac{\partial (d^*(m - 1) - \hat{y}^H(m)W^*(m))}{\partial W(m)e(m)}e(m) = \frac{1}{2} \hat{y}^H(m)e^*(m). \quad (31)
\]

Then, substituting (30) and (31) into (29), we get

\[
\frac{\partial J(m)}{\partial W(m)} = -e^*(m)\Re(\hat{y}^T(m)) + \frac{1}{2} \hat{y}^H(m)e^*(m) = -\frac{1}{2} \hat{y}^T(m)e^*(m) \quad (32)
\]

According to Eq. (32) and LMS adaptive algorithm [25, 46], we get the weight update formula of the proposed SNL-QPSOVAF, as follows

\[
W(m + 1) = W(m) - \lambda_1 \nabla_W J(m) = W(m) - \lambda_1 \left( \frac{\partial J(m)}{\partial W(m)} \right)^H \quad (33)
\]

where \( \lambda_1 \) denotes a linear step size and the constant 1/2 in (32) is absorbed into the step size.
4.1.2 Nonlinear subsection

We still use the minimize the cost function \( J(m) \) to update the nonlinear weight vector \( \mathbf{H}(m) \) of the proposed SNL-QPSOV AF as follows

\[
\mathbf{H}(m+1) - \mathbf{H}(m) = -\lambda_2 \nabla_{\mathbf{H}} J(m) \\
= -\lambda_2 \left( \frac{\partial J(m)}{\partial \mathbf{H}(m)} \right)^H
\]

(34)

where \( \lambda_2 \) represents the nonlinear step size, and

\[
\mathbf{H}(m) = [h_1(m), h_2(m), \cdots, h_L(m)]^T.
\]

(35)

In order to obtain the above \( \nabla_{\mathbf{H}} J(m) \), the rule given in (8) should also be used as follows

\[
\frac{\partial J(m)}{\partial \mathbf{H}(m)} = e^*(m) \frac{\partial e(m)}{\partial \mathbf{H}(m)} + \frac{\partial e^*(m)}{\partial \mathbf{H}(m)} e(m). \tag{36}
\]

Then, similar to (29), process two partial derivations in (36) as follows

\[
\frac{\partial e(m)}{\partial \mathbf{H}(m)} = \frac{\partial \left( d(m-1) - \mathbf{W}^T(m)\hat{\mathbf{y}}(m) \right)}{\partial \mathbf{H}(m)} = -\mathbf{R}(\mathbf{W}^T(m)) \left[ \mathbf{R}(\mathbf{U}_1^T(m)), \ldots, \mathbf{R}(\mathbf{U}_M^T(m)) \right]^T \tag{37}
\]

\[
= -\sum_{r=1}^M \mathbf{R}(w_r(m)) \mathbf{R}(\mathbf{U}_r^T(m))
\]

and

\[
\frac{\partial e^*(m)}{\partial \mathbf{H}(m) e(m)} e(m) = -\frac{\partial (\hat{\mathbf{y}}^H(m)\mathbf{W}^*(m))}{\partial \mathbf{H}(m) e(m)} e(m)
\]

\[
= -\sum_{r=1}^M \frac{\partial (\hat{\mathbf{y}}^*_r(m)w^*_r(m))}{\partial \mathbf{H}(m) e(m)} e(m)
\]

\[
\approx -\sum_{r=1}^M w^*_r(m) \frac{\partial (\mathbf{U}_r^H(m)\mathbf{H}^*(m))}{\partial \mathbf{H}(m) e(m)} e(m)
\]

\[
= \frac{1}{2} \sum_{r=1}^M w^*_r(m) \mathbf{U}_r^H(m)e^*(m)
\]

(38)

Then, plugging (37) and (38) into (36), we obtain

\[
\frac{\partial J(m)}{\partial \mathbf{H}(m)} = -e^*(m) \sum_{r=1}^M \mathbf{R}(w_r(m)) \mathbf{R}(\mathbf{U}_r^T(m))
\]

\[
+ \frac{1}{2} \sum_{r=1}^M w^*_r(m) \mathbf{U}_r^H(m)e^*(m) \tag{39}
\]

\[
= -\frac{1}{2} \sum_{r=1}^M w_r(m) \mathbf{U}_r^T(m)e^*(m)
\]

Using the equation (39), (34) can be rewritten as

\[
\mathbf{H}(m+1) = \mathbf{H}(m) - \lambda_2 \left( \frac{\partial J(m)}{\partial \mathbf{H}(m)} \right)^H
\]

(40)

\[
= \mathbf{H}(m) + \lambda_2 e(m) \sum_{r=1}^M \mathbf{U}_r^* (m) w^*_r (m)
\]

4.2 Derivation of the SWNL-QPSOVAF

4.2.1 Nonlinear subsection

According to SWL modelling in (11), similar to (23), the \( r \)th augmented model output of the SWNL-QPSOVAF is given by

\[
y_r(m) = \mathbf{H}^a(m) \mathbf{U}_r(m) + \mathbf{G}(m) \mathbf{U}_r^a(m)
\]

\[
= \mathbf{H}^aT(m) \mathbf{U}_r^a(m) \tag{41}
\]

where \( \mathbf{H}^a(m) \) and \( \mathbf{U}_r^a(m) \) are the augmented weight and input vector of the \( r \)th module of SOV series, respectively. \( \mathbf{H}^a(m) \) and \( \mathbf{U}_r^a(m) \) are given by

\[
\mathbf{H}^a(m) = \left[ \mathbf{H}^T(m), \mathbf{G}^T(m) \right]^T \tag{42}
\]

\[
\mathbf{U}_r^a(m) = \left[ \mathbf{U}_r^T(m), \mathbf{U}_r^{aT}(m) \right]^T \tag{43}
\]

where

\[
\mathbf{G}(m) = [g_1(m), g_2(m), \cdots, g_L(m)]^T. \tag{44}
\]

Similar to Eq. (34), minimizing the cost function \( J(m) \), the weight update function of the proposed SWNL-QPSOVAF is obtained as

\[
\mathbf{H}^a(m+1) - \mathbf{H}^a(m) = -\lambda_2 \nabla_{\mathbf{H}^a} J(m)
\]

\[
= -\lambda_2 \left( \frac{\partial J(m)}{\partial \mathbf{H}^a(m)} \right)^H \tag{45}
\]
Processing $\nabla_{H^a} J(m)$ by using the same manner in section A, we get

$$H^a(m + 1) = H^a(m) + \lambda_2 e(m) \sum_{r=1}^{M} U^r s^r(m) w^r_s(m)$$

(46)

### 4.2.2 Linear subsection

Since the weight update of the linear subsection of the proposed model is the same as the previous part, it is not derived here. Please refer to the previous section IV-A for details.

### 4.3 Derivation of the WNL-QPSOVAF

#### 4.3.1 Nonlinear subsection

In this section, considering the WNL model given in (9), we define the $r$th augmented model output of the proposed WNL-QPSOVAF in the vector form as

$y_r(m) = H^T(m)U_r(m) + G^T(m)U^r(m)
+ V^T(m)U^j(m) + Z^T(m)U^k(m)$

(47)

$$= H^aT(m)U^a_r(m)$$

where $V(m) = [v_1(m), v_2(m), \cdots, v_L(m)]^T$ and $Z(m) = [z_1(m), z_2(m), \cdots, z_L(m)]^T$ are the $j$ and $k$ basis quaternion coefficients of the WNL-QPSOVAF. The input $U^a_r(m)$ and the augmented weight $H^a(m)$ are given by as follows

$$U^a_r(m) = \left[ U_r(m), U^j_r(m), U^k_r(m), U^r(m) \right]^T$$

(48)

$$H^a(m) = \left[ H^T(m), G^T(m), V^T(m), Z^T(m) \right]^T$$

(49)

Similar to Eqs. (45) and (46), we can get the update weight of the WNL-QPSOVAF

$$H^a(m + 1) = H^a(m) + \lambda_2 e(m) \sum_{r=1}^{M} U^r s^r(m) w^r_s(m).$$

(50)

#### 4.3.2 Linear subsection

Note that the update of the parameters of the linear subsection is unchanged with the last part. So please refer to the previous section A for details.

### 5 Performance analysis

#### 5.1 Convergence analysis

To analyse the proposed algorithms, we primarily process the following assumptions.

**Assumption 1** The input signal $x(m)$ is statistically independent of the desire signal $d(m)$, the weight $H(m)$ and $W(m)$.

**Assumption 2** The system noise $v(m)$ is a zero-mean white Gaussian noise with variance $\delta^2_v$ and is independent of $x(m)$, $H(m)$, $W(m)$, $e(m)$, $\epsilon_W(m)$ and $\epsilon_H(m)$.

**Assumption 3** The auto-correlation positive definite matrix of the output $\hat{y}(m)$ and its conjugate form are $R_{\hat{y}}$ and $R_{\hat{y}^*}$, respectively; $\hat{y}(m)$ and the desired signal $d(n)$ have a covariance matrix $R_{d\hat{y}}$

**Assumption 4** The a prior error $\epsilon(m)$ and $e(m)$ are independent of $\|\hat{y}^*(m)\|^2$.

#### 5.1.1 Linear subsection

Substituting (28) into (33), we can obtain

$$W(m + 1) = W(m) - \lambda_1 (d(m - 1)) \hat{y}^*(m)$$

(51)

Rearranging yields

$$W(m + 1) = W(m) - \lambda_1 (d(m - 1)) \hat{y}^*(m)$$

(52)

Since $W^T(m)\hat{y}(m)\hat{y}^*(m) = (W^T(m)\hat{y}(m)\hat{y}^H(m))^T$, we have

$$W(m + 1) = W(m) - \lambda_1 (W^T(m)\hat{y}(m)\hat{y}^H(m))^T + \lambda_1 (d(m - 1)) \hat{y}^*(m)$$

(53)

Taking the transpose of both sides of (53) yields

$$W^T(m + 1) = W^T(m) - \lambda_1 (W^T(m)\hat{y}(m)\hat{y}^H(m))^T + \lambda_1 (d(m - 1)) \hat{y}^H(m)$$

(54)

Using (54) yields

$$W^T(m + 1) = W^T(m) \left( I_M - \lambda_1 (\hat{y}(m)\hat{y}^H(m))^T \right) + \lambda_1 (d(m - 1)) \hat{y}^H(m)$$

(55)

where $I_M$ represents the $M \times M$ unit matrix.

© Springer
Using Assumptions 1–4, the expectation on both sides of (55) can be obtained by

\[
E\left\{ W(m + 1) \right\} = E\left\{ W(m) \right\} \left[ I_M - \lambda_1 R_{\hat{y}\hat{y}} \right] + \lambda_1 R_{d\hat{y}} \tag{56}
\]

where

\[
R_{\hat{y}\hat{y}} = E\left\{ \hat{y}(m)\hat{y}^T(m) \right\}
\]

\[
R_{d\hat{y}} = E\left\{ d(m - 1)\hat{y}^T(m) \right\}.
\]

Simplifying the above formula according to the Wiener filter theory [47] yields

\[
E\left\{ W(m + 1) \right\} = W(0) \left[ I_M - \lambda_1 R_{\hat{y}\hat{y}} \right]^{m+1} + \lambda_1 \sum_{r=0}^{m} \left[ I_M - \lambda_1 R_{\hat{y}\hat{y}} \right]^r R_{d\hat{y}}. \tag{58}
\]

Finally, we can gain the step size range

\[
0 < \lambda_1 < \frac{2}{\zeta_{\text{max}1}} \tag{59}
\]

where \( \zeta_{\text{max}1} \) is the max eigenvalue of matrix \( R_{\hat{y}\hat{y}} \).

5.1.2 Nonlinear subsection

Similar to the linear subsection, the derivation of \( J(m) \) with respect to \( H(m) \) can be rewritten as

\[
\nabla_{H^*} J(m) = -\nabla_{H^*} \left[ d^*(m - 1)W^T(m)\hat{y}(m) + \hat{y}^*(m)W^T(m) d(m - 1) \right] + \nabla_{H^*} \left[ \left[ W^T(m)\hat{y}(m) \right]^* W^T(m)\hat{y}(m) \right]. \tag{60}
\]

The first derivative on the right side of (60) is derived as follows

\[
\nabla_{H^*} \left[ d^*(m - 1)W^T(m)\hat{y}(m) + \hat{y}^*(m)W^T(m) d(m - 1) \right]
= \nabla_{H^*} \left[ d^*(m - 1) \sum_{r=1}^{M} w_r(m)y_r(m) \right] + \nabla_{H^*} \left[ \left[ \sum_{r=1}^{M} w_r(m)y_r(m) \right]^* W^T(m)\hat{y}(m) \right]
= \nabla_{H^*} \left[ d^*(m - 1) \sum_{r=1}^{M} w_r(m)H^T(m)U_r(m) \right] + \nabla_{H^*} \left[ \left[ \sum_{r=1}^{M} w_r(m)H^T(m)U_r(m) \right]^* d(m - 1) \right]
= d^*(m - 1)\tilde{\omega}(m) + \tilde{\omega}^*(m)d(m - 1)
\]

where

\[
\tilde{\omega}(m) = \sum_{r=1}^{M} w_r(m)U_r(m) \tag{62}
\]

The second derivative on the right side of (60) is derived as shown below

\[
\nabla_{H^*} \left[ \left[ W^T(m)\hat{y}(m) \right]^* W^T(m)\hat{y}(m) \right]
= \nabla_{H^*} \left[ \sum_{r=1}^{M} w_r(m)y_r(m) \right]^* \sum_{r=1}^{M} w_r(m)y_r(m)
= \tilde{\omega}^*(m)H(m)\tilde{\omega}(m) \tag{63}
\]

Therefore, according to Eqs. (61) and (63), (60) can be rewritten as

\[
\nabla_{H^*} J(m) = -d^*(m - 1)\tilde{\omega}(m) - \tilde{\omega}^*(m)d(m - 1) + \tilde{\omega}^*(m)H(m)\tilde{\omega}(m) \tag{64}
\]

Substituting (64) in (40) can be obtained as shown below

\[
H(m + 1) = H(m) + \lambda_2 \left[ d^*(m - 1)\tilde{\omega}(m) + \tilde{\omega}^*(m)H(m)\tilde{\omega}(m) \right]. \tag{65}
\]

Then, taking the expectations on both sides of (65) and using Assumptions 1–4, we can obtain

\[
E \left\{ H(m + 1) \right\} = \left[ I_M - \lambda_1 R_{t13} \right] E \left\{ H(m) \right\} + \lambda_2 R_{d13} \tag{66}
\]

where

\[
R_{t13} = E \left\{ \tilde{\omega}^*(m)\tilde{\omega}(m) \right\} \tag{67}
\]

\[
R_{d13} = E \left\{ d^*(m - 1)\tilde{\omega}(m) + \tilde{\omega}^*(m)d(m - 1) \right\}.
\]

Considering the Wiener filter theory [47], Eq. (66) can be simplified as

\[
E \left\{ H(m + 1) \right\} = \left[ I_M - \lambda_2 R_{t13} \right] E \left\{ H(m) \right\} + \lambda_2 R_{d13} \tag{68}
\]

From the above equation, we can get the range of step size \( \lambda_2 \) as follows

\[
0 < \lambda_2 < \frac{2}{\zeta_{\text{max}2}} \tag{69}
\]

where \( \zeta_{\text{max}2} \) denotes the max eigenvalue of matrix \( R_{t13} \).
Remark 1 Combine (41), (45) and (68), we can obtain the weight expectation update function of the SWNL-QPSOV algorithm as follows

$$E\left\{ \hat{H}^a(m + 1) \right\} = \left[ I_m - \lambda_2 R^a_{\Omega,j3} \right]^{m+1} \hat{H}^a(0)$$

$$+ \lambda_2 \sum_{r=0}^{m} (I_m - \lambda_2 R^a_{\Omega,j3})^r R^a_{\Omega,j3}. \quad (70)$$

Then, by analysing (70), which is similar to the convergence of the SNL-QPSOV algorithm, the step size range of the SWNL-QPSOV algorithm can be finally obtained as

$$0 < \lambda_2 < \frac{2}{\zeta^a_{\max 2}} \quad (71)$$

where $\zeta^a_{\max 2}$ denotes the max eigenvalue of matrix $R^a_{\Omega,j3}$, which only include $R$ and $P$ defined in (12).

Remark 2 Combining (47), (50), similar to the above proposed algorithm analysis, we can get the step size range of the WNL-QPSOV algorithm, as follows

$$0 < \lambda_2 < \frac{2}{\zeta^a_{\max 2}} \quad (72)$$

where $\zeta^a_{\max 2}$ is the max eigenvalue of $R^a_{\Omega,j3}$, which include $R$, $P$, $S$ and $T$ defined in (10).

Note that the linear subsection analysis of the SWNL-QPSOVAF algorithm and the WNL-QPSOVAF algorithm are the same as the SNL-QPSOVAF algorithm. Therefore, there is no need to go into details here, and please refer to the linear subsection analysis of the SNL-QPSOV algorithm.

5.2 Mean square steady-state analysis

In this section, the mean square steady-state theoretical performance of SNL-QPSOVAF algorithm under the Gaussian input assumption is discussed.

5.2.1 Steady-state analysis of $W(m)$

Let $\tilde{W}(m) = W_o - W(m)$, where $W_o$ is the optimal weight vector. Then, subtracting both sides of (33) from $W_o$ yields

$$\tilde{W}(m + 1) = \tilde{W}(m) - \lambda_1 e(m) \hat{y}^*(m). \quad (73)$$

Then, calculating the 2-Norm of both sides of (73) yields

$$\|\tilde{W}(m + 1)\|^2 = \|\tilde{W}(m)\|^2 - 2\lambda_1 \tilde{W}(m) \hat{y}^*(m) e(m)$$

$$+ \lambda_1^2 \|\hat{y}^*(m)\|^2 e(m)^2. \quad (74)$$

Considering $E\|\tilde{W}(m + 1)\|^2 = E\|\tilde{W}(m)\|^2$ for $m \to \infty$ and using assumptions, so the expectation of (74) can be expressed as

$$2\lambda_1 E\{\tilde{W}(m) \hat{y}^*(m) e(m)\} = \lambda_1^2 E\{\|\hat{y}^*(m)\|^2 e(m)^2\}. \quad (75)$$

The *a priori* error $e(m)$ of the whole system is consist of the linear error $\epsilon_w(m)$ and the nonlinear error $\epsilon_H(m)$, i.e. $\epsilon(m) = \epsilon_w(m) + \epsilon_H(m)$, and it is related to $e(m)$ via $e(m) = \epsilon(m) + \nu(m)$. According to Assumptions 1–4 and the equation $\epsilon_w(m) = \tilde{W}(m) \hat{y}^*(m)$ when $\epsilon_H(m) \approx 0$, (75) can be rewritten as

$$2E\{\tilde{W}(m) \hat{y}^*(m) \epsilon_w(m) + \nu(m)\}$$

$$= \lambda_1 E\{\|\hat{y}^*(m)\|^2 [\epsilon_w(m) + \nu(m)]^2\}$$

$$= \lambda_1 E\{\|\hat{y}^*(m)\|^2 [\epsilon_w(m)]^2\}$$

$$+ \lambda_1 E\{\|\hat{y}^*(m)\|^2 \delta_w^2\}, \quad (76)$$

$$2E\{\epsilon_w^2(m)\} = \lambda_1 E\{\|\hat{y}^*(m)\|^2 \epsilon_w^2(m)\}$$

$$+ \lambda_1 E\{\|\hat{y}^*(m)\|^2 \delta_w^2\},$$

where $E\{\|\hat{y}^*(m)\|^2\}$ can be calculated by using the approximation $\hat{H}(m) = H_o$. Then, rearranging (76), we have

$$E\{\epsilon_w^2(m)\} = \frac{\lambda_1 E\{\|\hat{y}^*(m)\|^2\} \delta_w^2}{2 - \lambda_1 E\{\|\hat{y}^*(m)\|^2\}} \quad (77)$$

which leads to the EMSE of the linear section as $m \to \infty$, i.e.

$$\epsilon_w = \lim_{m \to \infty} E\{\epsilon_w^2(m)\} = \frac{\lambda_1 E\{\|\hat{y}^*(m)\|^2\} \delta_w^2}{2 - \lambda_1 E\{\|\hat{y}^*(m)\|^2\}}. \quad (78)$$

5.2.2 Steady-state analysis of $H(m)$

Similar to the method of steady-state analysis of the linear weight vector $W(m)$, we estimate the EMSE of the nonlinear weight vector $H(m)$ in this section. Letting $\hat{H}(m) = H_o - H(m)$, and substituting it into (40),
we have
\[ \tilde{\mathbf{H}}(m + 1) = \tilde{\mathbf{H}}(m) - \lambda_2 e(m) \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m). \] (79)

Calculating the 2-Norm on both sides of (79), using assumptions and taking its expectation, we get
\[
E \left\{ \left\| \tilde{\mathbf{H}}(m + 1) \right\|^2 \right\} = E \left\{ \left\| \tilde{\mathbf{H}}(m) \right\|^2 \right\} - 2\lambda_2 E \left\{ \tilde{\mathbf{H}}(m) \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) e(m) \right\} \nonumber
+ \lambda_2^2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 e(m)^2 \right\}. \] (80)

When \( m \to \infty \), we have \( E \{ \| \tilde{\mathbf{H}}(m + 1) \|^2 \} = E \{ \| \tilde{\mathbf{H}}(m) \|^2 \} \). Using the same method in subsection 5.2.1, we get \( e_H(m) = \tilde{\mathbf{H}}(m) \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \).

Thus, (80) can be rewritten as
\[
2\lambda_2 E\{ e_H(m) e(m) \} = \lambda_2^2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 e(m)^2 \right\}. \] (81)

Under the condition that Assumptions 1–3, using \( e(m) = e_H(m) + \nu(m) \) for \( e_W(m) \approx 0 \), (81) can be rearranged as
\[
2 E \{ e_H(m) [ e_H(m) + \nu(m) ] \} = \lambda_2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 \right\} E \{ [ e_H(m) + \nu(m) ]^2 \} \] \[ = \lambda_2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 \right\} E \{ e_H^2(m) \} + \delta^2 \] \[
(82)
\]

where \( E \{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 \} \) can be calculated by using the approximation \( \mathbf{W}(m) = \mathbf{W}_o \). Then rearranging (82), we have
\[
E \{ e_H^2(m) \} = \frac{\lambda_2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 \right\} \delta^2}{2 - \lambda_2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 \right\}}. \] (83)

Similarly, \( \xi_H = \lim_{m \to \infty} E \{ e_H^2(m) \} \) is specified as the EMSE of the nonlinear section parameter. Substituting (83) into \( \xi_H = \lim_{m \to \infty} E \{ e_H^2(m) \} \), we get
\[
\xi_H = \frac{\lambda_2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 \right\} \delta^2}{2 - \lambda_2 E \left\{ \left( \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \right)^2 \right\}}. \] (84)

5.2.3 Steady-state EMSE for SNL-QPSOVAF

The whole EMSE of the SNL-QPSOVAF algorithm is evaluated by
\[
\xi = \lim_{m \to \infty} E \{ e^2(m) \} = \lim_{m \to \infty} E \{ e_W(m) + e_H(m) \} \nonumber
= \lim_{m \to \infty} E \{ e_W^2(m) \} + \lim_{m \to \infty} E \{ e_H^2(m) \} \nonumber
+ 2 \lim_{m \to \infty} E \{ e_W(m) e_H(m) \} \] \[ = \xi_w + \xi_H + 2 \xi_{WH}. \] (85)

On the condition of system at steady state, \( W(m) \approx W_o \) \( H(m) \approx H_o \), we have
\[
\xi_{WH} = \lim_{m \to \infty} E \{ e_W(m) e_H(m) \} \nonumber
= \lim_{m \to \infty} E \{ [ W_o - W(m) ]^T \hat{\mathbf{v}}^*(m) \} \nonumber
\times \left[ H_o - H(m) \right]^T \sum_{r=1}^{M} \mathbf{U}^*_{r}(m) w^*_r(m) \} \approx 0 \] (86)

Therefore, (85) becomes
\[
\xi = \xi_w + \xi_H. \] (87)

5.3 Complexity analysis

For the convenience of observation, we show the complexity evaluation in table I. As can be seen from the Table 1, if \( N, M \) and \( K \) are, respectively, set to 10, 5 and 2, the SNL-QPSOVAF requires 66 additions and 131 multiplications while SNL-QPSOV needs only 25 additions and 41 multiplications. Therefore, we can obtain that the computational complexity of the proposed QPSOVAFs, is significantly lower than that of the QSOVAFs.

6 Simulation

In this section, simulated experiments are carried out to illustrate the performance of the proposed filters and verify the correctness of the steady-state analysis.

6.1 Case study 1

We use the following model of the unknown system as
\[
d(m) = \frac{d(m - 1)}{1 + d^2(m - 1)} + x^2(m) \] (88)
where \( d(m) \) represents the output of the nonlinear system, and \( x(m) \) denotes the input of the unknown system. \( d(m) \) is also interfered with the independent Gaussian white noise \( v(m) \). In the following experiments, the input \( x(m) \) is assumed as a white Gaussian noise with zero-mean and unit variance. We use the mean square error (MSE) defined as \( 10 \log_{10} (e(m))^2 \) to evaluate the algorithm performance. The simulated results are gained by averaging the results over 30 independent runs. The modules number \( M \) is set to 5, and the number of an external signal of each module is set to 1.

In Fig. 6, the step sizes \( \lambda_1 \) and \( \lambda_2 \) are set to 0.003. Figure 6 clearly shows that the convergence speed of the proposed WNL-QPSOV is faster than those of other proposed algorithms under the same steady-state error condition.

Figure 7 compares the performance of the proposed SNL-QPSOV algorithm with that of the conventional SNL-QSOV algorithm [39], where the step sizes \( \lambda_1 \) and \( \lambda_2 \) of the SNL-QSOVAF and the proposed SNL-QPSOV are all set to 0.002. The signal-to-noise ratio (SNR) for Figs. 7a and 7b is set to 30dB and 20dB, respectively. The simulation result shows that the proposed SNL-QPSOV has a better performance than the SNL-QSOVAF.

### 6.2 Case study 2

In this experiment, we compare the simulated EMSE with the derived theoretical EMSE, where the unknown nonlinear system is given by

\[
y_r(m) = H_o^T U_r(m)
\]

\[
\hat{y}(m) = [y_1(m), y_2(m), \ldots, y_M(m)]^T
\]

\[
y(m) = W_o^T \hat{y}(m) + v(m)
\]

The optimal weight vectors \( H_o \) and \( W_o \) are given by

\[
H_o = [0.5; -0.08; -0.3 + 0.4i + 0.3j - 0.5k; 0.06; 0.6; -0.3]
\]

\[
W_o = [0.03; 0.2 - 0.5i + 0.1j - 0.1k; 0.22; -0.34; 0.4].
\]

(90)

In order to calculate the theoretical EMSEs given in (78) and (84), \( E(\|\hat{y}_r(m)\|^2) \) and \( E(\left\{ \sum_{i=1}^{M} U_r(m) w_r^*(m) \right\}^2) \) were first estimated by a numerical simulation, which is obtained by averaging of last 40000 samples averaged across 30 independent trails. Figure 8 gives the simulated and theoretical EMSE curves of the SNL-QPSOV adaptive filter where \( H_o \) is available. Figure 9 compares the simulated and theoretical EMSE versus step size. Figures 8 and 9 show the proximity between the theoretical and simulated EMSEs for different step sizes, where the variance \( \delta_r^2 \) is set to 0.001. Figure 10 depicts the simulated and theoretical EMSE curves of the SNL-QPSOV adaptive filter where \( W_o \) is available. Figure 11 compares the simulated and theoretical EMSE versus step size, where the variance \( \delta_r^2 \) is set to 0.002. Figures 10 and 11 clearly shows that the simulated EMSE is very close to the theoretical one.

### 6.3 Case study 3

In this experiment, we verify the advantage of the proposed pipelined architecture. To get a fair comparison, we assume that the imaginary part of all signals in the proposed SNL-QPSOVAF are zero, because the JPPSOV is only for real-valued signals. In the experiment, the model of the unknown system is

\[
d(m) = \frac{d(m-1)}{1 + d^2(m-1)} + x^2(m) + x^2(m-1)
\]

The measurement noise belonged to zero-mean and white Gaussian sequence, and the input signal-to-measurement noise ratio is chosen to be 30 dB. Figure 12 presents the simulated results. It is clearly seen

| Type of filter | Addition | Multiplication |
|---------------|----------|----------------|
| SNL-QSOVAF    | \( N^2/2 + 3N/2 + 1 \) | \( N^2 + 3N + 1 \) |
| SWNL-QSOVAF   | \( 2N^2 + 3N + 1 \) | \( 4N^2 + 6N + 1 \) |
| WNL-QSOVAF    | \( 8N^2 + 6N + 1 \) | \( 16N^2 + 12N + 1 \) |
| SNL-QPSOV     | \( K^2 + 2K + 4M - 3 \) | \( 3K^2/2 + 9K/2 + 5M + 1 \) |
| SWNL-QPSOV    | \( 4K^2 + 4K + 4M - 3 \) | \( 6K^2 + 9K + 5M + 1 \) |
| WNL-QPSOV     | \( 16K^2 + 8K + 4M - 3 \) | \( 24K^2 + 18K + 5M + 1 \) |
that the proposed SNL-QPSOVAF obtain better performance than the JPPSOV, which demonstrates the superiority of the novel pipelined structure.

In this experiment, we also test the effect of the number of modules. Figure 13 depicts the steady-state MSE of the SNL-QPSOVAF with $M$ increasing from 1 to
A family of quaternion-valued pipelined second-order

Fig. 11 Comparison between theoretical EMSE versus simulated EMSE for different values of step size

Fig. 12 MSE curves of SNL-QPSOVAF and JPPSOV

Fig. 13 Relation between the MSE and the number of modules

10. Note that there is an optimal value of the number of modules.

7 Conclusion

This paper first proposes the QPSOVs and gives detailed derivations. Since the pipelined structure is used, these algorithms could significantly reduce the computational complexities of conventional adaptive quaternion-valued SOV filters. Then, the convergence analysis, the complexity analysis and the mean square steady-state analysis of the proposed QPSOVs are provided. Moreover, this paper focuses on the study of the SNL-QPSOV algorithm. Considering that the analysis methods of the other two proposed algorithms are similar to the SNL-QPSOV algorithm and the length of the paper is limited, we do not conduct the detailed studies on the WNL-QPSOV and SWNL-QPSOV algorithms here. The results of the SNL-QPSOVAF algorithm can be easily expanded to its SWNL and WNL versions. Finally, we further verify effectiveness of the proposed algorithms and the correctness of the theoretical analysis through the simulations.

Some potential future work contains: (1) the extension to constrained minimization problems; (2) the robust QPSOVAFs against the impulsive noise in quaternion domain; and (3) the variable step size version of QPSOVAFs to improve the convergence rate.

Funding This work was supported by the National Natural Science Foundation of China under Grant No. 51977153, 51577046, the State Key Program of National Natural Science Foundation of China under Grant No. 51637004, the national key research and development plan “important scientific instruments and equipment development” Grant No. 2016YFF0102200, Equipment research project in advance Grant No. 41402040301.

Data availability statements The data that support the findings of this study are available from the corresponding author [Y. He] upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Lu, L., Yang, X., Wang, W., Yu, Y.: Recursive second-order volterra filter based on dawson function for chaotic memristor system identification. Nonlinear Dyn. 99(4), 3123–3142 (2020)
2. Wouw, N., Nijmeijer, H., van Campen, D.H.: A volterra series approach to the approximation of stochastic nonlinear dynamics. Nonlinear Dyn. 27(4), 397–409 (2002)
3. Annabestani, M., Naghavi, N.: Practical realization of discrete-time volterra series for high-order nonlinearities. Nonlinear Dyn. 98(3), 2309–2325 (2019)
4. de Paula, N.C.G., Marques, F.D.: Multi-variable volterra kernels identification using time-delay neural networks:

 Springer
application to unsteady aerodynamic loading. Nonlinear Dyn. 97(1), 767–780 (2019)
5. Lu, L., Yin, K., de Lamare, R.C., Zheng, Z., Yu, Y., Yang, X., Chen, B.: A survey on active noise control in the past decade-part ii: Nonlinear systems. Signal Process. 181, 107929 (2021)
6. Lu, L., Zhao, H., Chen, B.: Time series prediction using kernel adaptive filter with least mean absolute third loss function. Nonlinear Dyn. 90(2), 999–1013 (2017)
7. Burton, T., Beaucoup, F., Goubran, R.: Nonlinear system identification using a subband adaptive volterra filter. In: IEEE Instrumentation and Measurement Technology Conference 2008, 939–944 (2008)
8. Sicuranza, G.L., Mathews, V.J.: Polynomial signal processing. Wiley, New York (2001)
9. Burton, T.G., Goubran, R.A.: A generalized proportionate subband adaptive second-order volterra filter for acoustic echo cancellation in changing environments. IEEE Trans. Audio Speech Lang. Process. 19(8), 2364–2373 (2011)
10. Despotovic, V., Goertz, N., Peric, Z.: Nonlinear long-term prediction of speech based on truncated volterra series. IEEE Trans. Audio Speech Lang. Process. 20(3), 1069–1073 (2012)
11. Franz, M.O., Schölkopf, B.: A unifying view of wiener and volterra theory and polynomial kernel regression. Neural Comput. 18(12), 3097–3118 (2006)
12. Lainiotis, D.G., Papararshkeva, P.: Adaptive approach to nonlinear channel equalization. IEEE Trans. Commun. 46(10), 1325–1336 (1998)
13. Korenberg, M. J., Witke, P. H.: Adaptive identification of dispersive nonlinear data transmission channels. In: IEEE International Conference on Communications, - Spanning the Universe, pp. 939–945 vol. 2 (1988)
14. Zhao, H., Zhang, J.: A novel adaptive nonlinear filter-based pipelined feedforward second-order volterra architecture. IEEE Trans. Signal Process. 57(1), 237–246 (2009)
15. Crespo-Cadenas, C., Aguilera-Bonet, P., Becerra-González, J.A., Cruces, S.: Subband adaptive model for complex-valued systems. Signal Process. 96, 401–405 (2014)
16. Chevalier, P., Duvaut, P., Piccinbono, B.: Complex transversal volterra filters optimal for detection and estimation. In Acoustics, Speech, and Signal Processing, IEEE International Conference on. IEEE Computer Society, pp. 3537–3538 (1991)
17. Chevalier, P., Oukaci, A., Delmas, J.: Third order widely non linear volterra mvdr beamforming. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 2648–2651 (2011)
18. Liu, N., Ju, C.: Blind i/q imbalance and nonlinear is mitigation in nyquist-scm direct detection system with cascaded widely linear and volterra equalizer. Optics Commun. 409, 94–98 (2018)
19. Crespo-Cadenas, C., Madero-Ayora, M.J., Reina-Tosina, J., Becerra-González, J.: Formal deduction of a volterra series model for complex-valued systems. Signal Process. 131, 245–248 (2017)
20. Pei, Soo-Chang., Chen, Ching-Min.: Color image processing by using binary quaternion-moment-preserving thresholding technique. IEEE Trans. Image Process. 8(5), 614–628 (1999)
21. Xia, Y., Jahanchahi, C., Mandic, D.P.: Quaternion-valued echo state networks. IEEE Trans. Neural Netw. Learn. Syst. 26(4), 663–673 (2015)
22. Xia, Y., Jahanchahi, C., Nitta, T., Mandic, D.P.: Performance bounds of quaternion estimators. IEEE Trans. Neural Netw. Learn. Syst. 26(12), 3287–3292 (2015)
23. Shang, F., Hirose, A.: Quaternion neural-network-based polar land classification in poincare-sphere-parameter space. IEEE Trans. Geosci. Remote Sens. 52(9), 5693–5703 (2014)
24. Hanson, A. J.: Visualizing quaternions. In ACM SIGGRAPH 2005 Courses, pp. 1–es. (2005)
25. Took, C.C., Mandic, D.P.: The quaternion lms algorithm for adaptive filtering of hypercomplex processes. IEEE Trans. Signal Process. 57(4), 1316–1327 (2009)
26. Cheong Took, C., Mandic, D.P.: A quaternion widely linear adaptive filter. IEEE Trans. Signal Process. 58(8), 4427–4431 (2010)
27. Ujang, B.C., Took, C.C., Mandic, D.P.: Quaternion-valued nonlinear adaptive filtering. IEEE Trans. Neural Netw. 22(8), 1193–1206 (2011)
28. Jahanchahi, C., Mandic, D.P.: A class of quaternion kalman filters. IEEE Trans. Neural Netw. Learn. Syst. 25(3), 533–544 (2013)
29. Xu, D., Jahanchahi, C., Took, C.C., Mandic, D.P.: Enabling quaternion derivatives: the generalized hr calculus. Royal Soci. Open. Sci. 2(8), 150255 (2015)
30. Paul, T.K., Ogunfunmi, T.: A kernel adaptive algorithm for quaternion-valued inputs. IEEE Trans. Neural Netw. Learn. Syst. 26(10), 2422–2439 (2015)
31. Xu, D., Xia, Y., Mandic, D.P.: Optimization in quaternion dynamic systems: gradient, hessian, and learning algorithms. IEEE Trans. Neural Netw. Learn. Syst. 27(2), 249–261 (2015)
32. Xiang, M., Kanna, S., Mandic, D.P.: Performance analysis of quaternion-valued adaptive filters in nonstationary environments. IEEE Trans. Signal Process. 66(6), 1566–1579 (2017)
33. Buchholz, S., Le Bihan, N.: Polarized signal classification by complex and quaternionic multi-layer perceptrons. Int. J. Neural Syst. 18(02), 75–85 (2008)
34. Chen, X., Song, Q., Li, Z.: Design and analysis of quaternion-valued neural networks for associative memories. IEEE Trans. Syst. Man, Cybern: Syst. 48(12), 2305–2314 (2018)
35. Chen, X., Song, Q.: State estimation for quaternion-valued neural networks with multiple time delays. IEEE Trans. Syst. Man, Cybern: Syst. 49(11), 2278–2287 (2019)
36. Wei, R., Cao, J., Abdel-Aty, M.: Fixed-time synchronization of second-order mnns in quaternion field, IEEE Trans. Syst. Man, Cybern: Syst. 51(6), 3587–3598 (2019)
37. Vía, J., Ramírez, D., Santamaría, I.: Properness and widely linear processing of quaternion random vectors. IEEE Trans. Inf. Theory 56(7), 3502–3515 (2010)
38. Navarro-Moreno, J., Ruiz-Molina, J.C.: Semi-widely linear estimation of c$q$-proper quaternion random signal vectors under gaussian and stationary conditions. Signal Process. 119, 56–66 (2016)
39. Mengiç, E. C.: Design of quaternion-valued second-order volterra adaptive filters for nonlinear 3-d and 4-d signals, Signal Processing, p. 107619, (2020)
40. Li, L., Haykin, S.: A cascaded recurrent neural network for real-time nonlinear adaptive filtering. In IEEE International Conference on Neural Networks. IEEE, pp. 857–862 (1993)
41. Mandic, D., Chambers, J.: Recurrent neural networks for prediction: learning algorithms, architectures and stability, Wiley, (2001)
42. Zhao, H., Zhang, J.: A novel adaptive nonlinear filter-based pipelined feedforward second-order volterra architecture. IEEE Trans. Signal Process. 57(1), 237–246 (2008)
43. Ward, J. P.: Quaternions and Cayley numbers: Algebra and applications, vol. 403, Springer Science & Business Media, (2012)
44. Ell, T.A., Sangwine, S.J.: Quaternion involutions and anti-involutions. Comput. Math. Appl. 53(1), 137–143 (2007)
45. Singh, T.S.D., Chatterjee, A.: A comparative study of adaptation algorithms for nonlinear system identification based on second order volterra and bilinear polynomial filters. Measurement 44(10), 1915–1923 (2011)
46. Clarkson, P.M., Dokic, M.V.: Stability and convergence behaviour of second-order lms volterra filter. Electron. Lett. 27(5), 441–443 (1991)
47. Franz, M.O., Schölkopf, B.: A unifying view of wiener and volterra theory and polynomial kernel regression. Neural Comput. 18(12), 3097–3118 (2006)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.