Quaternion-Valued Distributed Filtering and Control

Sayed Pouria Talebi, Stefan Werner, and Danilo P. Mandic

***Abstract*—This work presents a unified framework for filtering and control of quaternion-valued state vector processes through multi-agent networked systems. To achieve this goal, the filtering problem in sensor networks is revisited, where a distributed Kalman filtering algorithm for filtering/tracking quaternion-valued state vector processes is developed. The distributed quaternion Kalman filter is formulated to mirror the operations of an optimal centralized approach in a fashion that will allow each agent to retain a Kalman style filtering operation and an intermediate estimate of the state vector. The work includes a comprehensive performance analysis of the developed distributed quaternion Kalman filtering algorithm, resulting in a closed-form expression for the second-order error moment. More importantly, due to the comprehensive framework for fusion of the covariance information and drawing upon concepts from the conducted performance analysis, a duality between the developed distributed Kalman filter and decentralized control is established. This essentially extends the duality between Kalman filtering and linear quadrature regulators to the quaternion domain and distributed setting. The theoretical concepts in this work are verified via simulations.**

***Index Terms*—Qauternion-valued filtering and control, sensor networks, distributed Kalman filtering, decentralized control.**

I. Introduction

In recent years, multi-agent networked systems have emerged as a suitable solution in a variety of complex engineering applications [1]–[10]. Owing to the versatility of the state-space model for representing real-world dynamic systems and optimality of the framework derived in [11], developing Kalman filtering techniques that are scalable with the size of the network has become the focal point of research into distributed filtering and learning [12]–[16].

Distributed solutions for filtering and control applications in networked multi-agent systems started to emerge in the late 1970s with the decentralized linear quadratic regulator proposed in [17]. In turn, the work in [17], inspired the initial distributed Kalman filtering solutions put forth in [18,19]. However, the filtering and control techniques in [17]–[19] are tantamount to replicating well-known centralized filtering and control algorithms at each agent in the network. This replication procedure is not only computationally inefficient, as all agents in the network are performing the same computations, but also results in a large amount of communication traffic, as each agent must have access to a vast array of information from all of its peers in the network.

In the Kalman filtering arena, in order to limit communication traffic and eliminate overlapping computations, two classes of distributed Kalman filtering frameworks have emerged. Referred to as *consensus Kalman filters*, this class of distributed Kalman filters use consensus information fusion techniques to fuse state vector estimates and observation data from agents across the network in order to improve the performance of local Kalman filtering operations implemented at each agent [1,2,12,20]–[23]. Following the introduction of consensus Kalman filtering frameworks and capitalizing upon the more flexible framework of diffusion techniques, so-called *diffusion Kalman filters* were introduced in the literature [15, 24,25]. This class of distributed Kalman filters operate akin to their consensus duals while replacing consensus information fusion techniques with diffusion. The flexibility with which diffusion weighting coefficients can be selected as compared to consensus weighting coefficients provides for a more effective approach for fusion of state vector estimates [14,15,24,26,27]. On the other hand, the consensus frameworks provide for a more comprehensive fusion of covariance information [12,23]. In the control arena, decentralized control techniques are seen as solutions that simultaneously accommodate for both scalability with the size of the network, as the work-load is spread among the agents, and improved performance in comparison to their non-cooperative duals, as each agent remains reasonably aware of the network status [1,2,17,28].

Traditionally, quaternions have been used for modeling three-dimensional rotations and orientation in a compact and computationally efficient manner [29]–[31]. The major advantages of modeling rotations using quaternions as compared to rotation matrices can be summarized as follows:

- The division algebra of quaternions provides for concise and mathematically tractable solutions with fewer restrictions than those obtainable in the real domain. This division algebra also allows to avoid problems associated with gimbal lock [29,30].
- Smooth interpolations of rotations can be produced when using quaternions, allowing for higher quality computer graphics and smooth trajectory control [32].
- Applying the roll, pitch, and yaw angles to express rotations, a degree of freedom is lost when one of the angles reaches $\pi/2$. This is not the case for quaternions.

In recent years, the introduction of the $\mathbb{HR}$-calculus [33,34], a framework for calculating derivatives of quaternion-valued functions, and the augmented quaternion statistics [35]–[37], a framework for exploiting the full second-order statistical information of quaternion-valued random signals, have led to the development of rigorous quaternion-valued signal process-

Sayed Pouria Talebi is with the Department of Signal Processing and Acoustics, School of Electrical Engineering, Aalto University, Espoo FI-02150 Finland, E-mail: pouria.talebi@aalto.fi.

Stefan Werner is with the Department of Electronic Systems, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Trondheim NO-7491 Norway, E-mail: stefan.werner@ntnu.no.

Danilo P. Mandic is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. E-mail: d.mandic@imperial.ac.uk.

The work of Sayed Pouria Talebi and Stefan Werner was supported in part by Academy of Finland under Grant 296849 and Research Council of Norway.

ing algorithms that are shown to outperform their real and complex-valued counterparts in various applications including three-phase power system monitoring [6,38]–[40], bearings-only tracking [6,41], color image processing [42], kernel learning [43], and avionic system [29,30]. This is largely owed to two main factors [29,30,32,38]–[41]:

- The more elegant way for expressing rotations compared with those obtainable with matrix algebras in $\mathbb{R}^3$ and $\mathbb{R}^4$; thus, resulting in mathematically tractable functional expressions with fewer constraints and with closed-form solutions.
- Allowing for estimation and control techniques to be derived directly in the quaternion domain using their division algebra; thus, resulting in information processing techniques with rigorous physical interpretation and compact representations leading to more accurate and efficient learning and control techniques.

Although distributed filtering and control techniques form the core of most modern networked multi-agent systems, the literature in this area is largely concerned with real-valued signals. However, many real-world applications deal with signals that are inherently three-dimensional, a scenario ideally suited for quaternions in terms of both convenience of representation and mathematical tractability. This presents the requirement for an all-inclusive quaternion-valued distributed filtering and control framework. To this end, the problem of filtering/tracking quaternion-valued state vector sequences via sensor networks is considered and a versatile quaternion-valued distributed Kalman filtering technique is derived. This is achieved through the decomposition of the optimal centralized Kalman filtering operations. This is conducted in a manner that allows each agent to retain a Kalman style filtering operation and an estimate of the state vector. The performance of the derived distributed quaternion-valued Kalman filter is analyzed establishing that it provides unbiased estimates and formulating a closed-form expression for the second-order error moment, referred to as the mean square deviation (MSD).

Importantly, the dual quaternion-valued regulator to the considered filtering problem is formulated. Then, drawing upon ideas from the performance analysis of the derived quaternion-valued distributed Kalman filter, it is demonstrated that optimal solution to the formulated quaternion-valued regulator problem can be approximated with any desirable degree of accuracy, using the decomposition approach conceived for the distributed quaternion-valued Kalman filter. This presents a quaternion-valued extension to the principle established by R. E. Kalman in [11]. Thus, introducing a rigorous framework for quaternion-valued distributed filtering and control.

***Mathematical Notations*:** Scalars, column vectors, and matrices are denoted by lowercase, bold lowercase, and bold uppercase letters, respectively. The augmented state vector at time instant n is denoted by $\mathbf{x}_n^a$, while $\mathbf{I}$ denotes an identity matrix of appropriate size. The transpose and Hermitian transpose operators are denoted as $(\cdot)^{\mathsf{T}}$ and $(\cdot)^{\mathsf{H}}$. The trace operator is represented by $\mathsf{Tr}\{\cdot\}$, while $\mathsf{E}\{\cdot\}$ represents the statistical expectation operator. The operator $\mathsf{vec}\{\cdot\}$ transforms a matrix into a column vector by stacking its columns. The real and quaternion domains are denoted by $\mathbb{R}$ and $\mathbb{H}$. Finally, the Kronecker product is denoted by $\otimes$.

II. Quaternion Algebra and Statistics

The skew-field of quaternions is a four-dimensional, non-commutative, associative, division algebra. A quaternion variable $q \in \mathbb{H}$ consists of a real part, $\Re(q)$, and a three-dimensional imaginary part or pure quaternion, $\Im(q)$, which comprises three components $\{\Im_i(q),\Im_j(q),\Im_k(q)\}$. A variable $q \in \mathbb{H}$ can be expressed as

$$\begin{aligned} q = \Re(q) + \Im(q) =& \Re(q) + \Im_i(q) + \Im_j(q) + \Im_k(g) \\ =& q_r + iq_i + jq_j + kq_k \end{aligned}$$

where $\{q_r, q_i, q_j, q_k\} \in \mathbb{R}$, while the orthonormal vectors $\{i, j, k\}$ form the basis of the quaternion imaginary subspace. The orthonormal vectors $\{i, j, k\}$ obey the multiplication rules

$$ij = -ji = k, jk = -kj = i, ki = -ik = j, ijk = -1$$

resulting in the non-commutativity property. The quaternion conjugate and norm are defined as

$$q^* = \Re(q) - \Im(q) \text{ and } |q| = \sqrt{qq^*} = \sqrt{q_r^2 + q_i^2 + q_j^2 + q_k^2}$$

respectively, whereas the multiplicative inverse of $q \in \mathbb{H}$ and $q \neq 0$ can be found from $q^{-1} = q^*/|q|^2$.

As an associative hypercomplex algebra, quaternions have an isomorphic matrix algebra [44]. In this context, we have

$$\forall q \in \mathbb{H} \leftrightarrow \mathbf{Q} = \begin{bmatrix} q_r & -q_i & -q_j & q_k \\ q_i & q_r & -q_k & -q_j \\ q_j & q_k & q_r & q_i \\ -q_k & q_j & -q_i & q_r \end{bmatrix} \in \mathbb{R}^4 \times \mathbb{R}^4 \quad (1)$$

where $\mathbf{Q}$ is the representation of q in the isomorphic matrix algebra [44,45]. The isomorphism in (1) is a useful analytical tool, as it allows for the use of well-known concepts in matrix algebra to be applied to quaternions with minor modifications.

The involution of $q \in \mathbb{H}$ around $\mu \in \mathbb{H}$ is defined as $q^\mu \triangleq \mu q \mu^{-1}$ [46]. Involution is seen as the quaternion equivalent of the conjugate operator in the complex domain and are used to introduce a mapping between $\mathbb{H}^4$ and $\mathbb{R}^4$ as [33,35]

$$\mathbf{q}^a = \begin{bmatrix} q \\ q^i \\ q^j \\ q^k \end{bmatrix} = \begin{bmatrix} 1 & i & j & k \\ 1 & i & -j & -k \\ 1 & -i & j & -k \\ 1 & -i & -j & k \end{bmatrix} \begin{bmatrix} q_r \\ q_i \\ q_j \\ q_k \end{bmatrix} \quad (2)$$

where $\mathbf{q}^a$ is referred to as the augmented quaternion vector. The quaternion conjugate can also be defined through involution as [35,36,46]

$$q^* = \Re(q) - \Im(q) = \frac{1}{2}\left(q^i + q^j + q^k - q\right).$$

Augmenting a quaternion-valued variable with its involution around the i, j, and k axes forms the basis of the $\mathbb{HR}$-calculus which is elaborated upon in the sequel.

The quaternion-valued function $f(\cdot) : \mathbb{H}^M \to \mathbb{H}$ with parameter vector $\mathbf{q} \in \mathbb{H}^M$ is differentiable if and only if it satisfies the Cauchy-Riemann-Fueter condition [33,47]. This imposes a severe restriction on admissible functions in filtering/control applications, as only linear functions can meet this condition [33,41]. The $\mathbb{HR}$-calculus [33,34,48] presents a solution to this problem using the mapping in (2). The $\mathbb{HR}$-calculus, considers the function $f(\mathbf{q}=\mathbf{q}_r+i\mathbf{q}_i+j\mathbf{q}_j+k\mathbf{q}_k):\mathbb{H}^M\rightarrow\mathbb{H}$ in terms of the orthogonal quaternion basis $\{\mathbf{q},\mathbf{q}^i,\mathbf{q}^j,\mathbf{q}^k\}$, so that $f(\mathbf{q}^a=[\mathbf{q}^{\mathsf{T}},\mathbf{q}^{i\mathsf{T}},\mathbf{q}^{j\mathsf{T}},\mathbf{q}^{k\mathsf{T}}]^{\mathsf{T}}):\mathbb{H}^{4M}\rightarrow\mathbb{H}$. Then, formulating the quaternion-valued function in terms of its real-valued components as $f(\mathbf{q}^a)=f_r(\mathbf{q}^a)+if_i(\mathbf{q}^a)+jf_j(\mathbf{q}^a)+kf_k(\mathbf{q}^a)$ and using the mapping in (2), a relation is established between the derivatives taken in $\mathbb{R}^4$ and those taken directly in $\mathbb{H}$. This, forms a unified framework for calculating the derivatives and establishing the gradients of quaternion-valued functions directly in the quaternion domain.

The relation between $\mathbf{q}^a=[\mathbf{q}^{\mathsf{T}},\mathbf{q}^{i\mathsf{T}},\mathbf{q}^{j\mathsf{T}},\mathbf{q}^{k\mathsf{T}}]^{\mathsf{T}}\in\mathbb{H}^{4M}$ and $[\mathbf{q}_r^{\mathsf{T}},\mathbf{q}_i^{\mathsf{T}},\mathbf{q}_j^{\mathsf{T}},\mathbf{q}_k^{\mathsf{T}}]^{\mathsf{T}}\in\mathbb{R}^{4M}$ also forms the basis of the augmented quaternion statistics. In this framework, the full second-order statistical description of quaternion random variables is given by the augmented covariance matrix [35,37]

$$\mathbf{C}_{\mathbf{q}^a}=\mathsf{E}\left\{\mathbf{q}^a\mathbf{q}^{a\mathsf{H}}\right\}=\begin{bmatrix}\mathbf{C}_{\mathbf{q}\mathbf{q}} & \mathbf{C}_{\mathbf{q}\mathbf{q}^i} & \mathbf{C}_{\mathbf{q}\mathbf{q}^j} & \mathbf{C}_{\mathbf{q}\mathbf{q}^k}\\ \mathbf{C}_{\mathbf{q}^i\mathbf{q}} & \mathbf{C}_{\mathbf{q}^i\mathbf{q}^i} & \mathbf{C}_{\mathbf{q}^i\mathbf{q}^j} & \mathbf{C}_{\mathbf{q}^i\mathbf{q}^k}\\ \mathbf{C}_{\mathbf{q}^j\mathbf{q}} & \mathbf{C}_{\mathbf{q}^j\mathbf{q}^i} & \mathbf{C}_{\mathbf{q}^j\mathbf{q}^j} & \mathbf{C}_{\mathbf{q}^j\mathbf{q}^k}\\ \mathbf{C}_{\mathbf{q}^k\mathbf{q}} & \mathbf{C}_{\mathbf{q}^k\mathbf{q}^i} & \mathbf{C}_{\mathbf{q}^k\mathbf{q}^j} & \mathbf{C}_{\mathbf{q}^k\mathbf{q}^k}\end{bmatrix}$$

where $\forall\zeta,\zeta'\in\{1,i,j,k\},\mathbf{C}_{\mathbf{q}^\zeta\mathbf{q}^{\zeta'}}=\mathsf{E}\left\{\mathbf{q}^\zeta\mathbf{q}^{\zeta'\mathsf{H}}\right\}$.

To further illustrate the need for a so-called augmented approach, consider the minimum mean square error (MMSE) estimator of a variable, y, conditioned on the observation, z, that is $\hat{y}=\mathsf{E}\{y|z\}$. Following the complex domain analogy [49], when y and z are quaternion-valued zero-mean and jointly Gaussian random variables, the MMSE estimator has to be expressed according to the individual components of the quaternion random variables; thus, we have

$$\begin{aligned}\hat{y}=&\mathsf{E}\{y_r|z_r,z_i,z_j,z_k\}+i\mathsf{E}\{y_i|z_r,z_i,z_j,z_k\}+\\&+j\mathsf{E}\{y_j|z_r,z_i,z_j,z_k\}+k\mathsf{E}\{y_k|z_r,z_i,z_j,z_k\}.\end{aligned}$$

Now, the mapping in (2) is used to replace $\{z_r,z_i,z_j,z_k\}$ with $\{z,z^i,z^j,z^k\}$ resulting in

$$\begin{aligned}\hat{y}=&\mathsf{E}\left\{y_r|z,z^i,z^j,z^k\right\}+i\mathsf{E}\left\{y_i|z,z^i,z^j,z^k\right\}\\&+j\mathsf{E}\left\{y_j|z,z^i,z^j,z^k\right\}+k\mathsf{E}\left\{y_k|z,z^i,z^j,z^k\right\}.\end{aligned}$$

Therefore, for quaternion-valued, zero-mean, and jointly Gaussian z and y, the MMSE solution is in the form of a widely-linear estimator given by [35,50]

$$\hat{y}=\mathbf{g}^{\mathsf{T}}\mathbf{z}+\mathbf{h}^{\mathsf{T}}\mathbf{z}^i+\mathbf{u}^{\mathsf{T}}\mathbf{z}^j+\mathbf{v}^{\mathsf{T}}\mathbf{z}^k=\left[\mathbf{g}^{\mathsf{T}},\mathbf{h}^{\mathsf{T}},\mathbf{u}^{\mathsf{T}},\mathbf{v}^{\mathsf{T}}\right]\mathbf{z}^a \quad (3)$$

where $\{\mathbf{g},\mathbf{h},\mathbf{u},\mathbf{v}\}$ are quaternion-valued coefficient vectors and $\mathbf{z}$ is the regressor vector[1]. The estimator in (3) can also be expressed as

$$\hat{\mathbf{y}}^a=\mathbf{W}^a\mathbf{z}^a \quad (4)$$

where $\mathbf{W}^a$ is matrix constructed appropriately from $\{\mathbf{g},\mathbf{h},\mathbf{u},\mathbf{v}\}$ and their involutions [41].

[1]For more detail regarding the MMSE solution in (3) the keen reader is referred to [35,50], while an inclusive treatment of statistical estimation in the quaternion domain can be found in [35]–[37,43,50].

III. Problem Statement

A. The Network

The network is modeled as a connected undirected graph $\mathcal{G}=\{\mathcal{N},\mathcal{E}\}$ so that node set $\mathcal{N}$ denotes the agents in the network and edge set $\mathcal{E}$ denotes bidirectional communication links between the agents. The cardinality of node set $\mathcal{N}$ is expressed as $|\mathcal{N}|$ and denotes the number of nodes in $\mathcal{N}$. The neighborhood of node l is defined as the set of nodes that can communicate with it, including self-communication, which is represented by the set $\mathcal{N}_l$ whose cardinality is denoted as $|\mathcal{N}_l|$.

B. Distributed Filtering Problem

Consider a dynamic physical system that is expressed in terms of the quaternion-valued augmented discrete-time state-space model

$$\mathbf{x}_n^a=\mathbf{A}_n^a\mathbf{x}_{n-1}^a+\boldsymbol{\nu}_n^a \quad (5)$$

where $\mathbf{A}_n^a$ is the state transition matrix at time n, while $\boldsymbol{\nu}_n$ is a zero-mean white Gaussian random vector with augmented covariance matrix $\mathsf{E}\left\{\boldsymbol{\nu}_n^a\boldsymbol{\nu}_n^{a\mathsf{H}}\right\}=\mathbf{C}_{\boldsymbol{\nu}_n^a}$. The goal is to enable each agent of the network to track the state vector sequence given available observations throughout the network, which are modeled as

$$\mathbf{y}_{l,n}^a=\mathbf{H}_{l,n}^a\mathbf{x}_n^a+\boldsymbol{\omega}_{l,n}^a \quad (6)$$

where $\mathbf{y}_{l,n}^a$ and $\mathbf{H}_{l,n}^a$ are the augmented observation vector and observation matrix at time instant n at node l, while $\boldsymbol{\omega}_{l,n}$ is the observation noise at node l and time instant n, which is independent from $\boldsymbol{\nu}_n$, and has augmented covariance matrix

$$\mathsf{E}\left\{\boldsymbol{\omega}_{l,n}^a\boldsymbol{\omega}_{m,\iota}^{a\mathsf{H}}\right\}=\mathbf{C}_{\boldsymbol{\omega}_{l,n}^a}\delta(l-m)\delta(n-\iota) \quad (7)$$

with $\delta(\cdot)$ denoting the Kronecker delta function.

C. Distributed Control Problem

The dual regulator to the filtering problem in Section III-B is now formulated. To this end, consider the noise-free multi-agent controlled dynamic system modeled as

$$\mathbf{x}_{n+1}^a=\mathbf{A}^a\mathbf{x}_n^a+\sum_{\forall l\in\mathcal{N}}\mathbf{B}_l^a\mathbf{u}_{l,n}^a \quad (8)$$

where $\mathbf{B}_l^a$ represents the actuator dynamic of the agent implementing control input $\mathbf{u}_{l,n}^a$. The goal is to find optimal control inputs $\{\mathbf{u}_{l,n}^a:\forall l\in\mathcal{N},n=1,2,\ldots,N\}$ that minimize the cost function

$$\mathcal{J}=\mathbf{x}_N^{a\mathsf{H}}\mathbf{T}^a\mathbf{x}_N^a+\sum_{n=1}^{N-1}\mathbf{x}_n^{a\mathsf{H}}\mathbf{Q}^a\mathbf{x}_n^a+\sum_{n=1}^{N-1}\sum_{\forall l\in\mathcal{N}}\mathbf{u}_{l,n}^{a\mathsf{H}}\mathbf{R}_l^a\mathbf{u}_{l,n}^a \quad (9)$$

subject to the widely-linear control law $\forall l,n:\mathbf{u}_{l,n}^a=\mathbf{L}_{l,n}^a\mathbf{x}_n^a$, where N is referred to as the control horizon, while $\mathbf{T}^a$, $\mathbf{Q}^a$, and $\{\mathbf{R}_l^a:\forall l\in\mathcal{N}\}$ are positive definite Hermitian symmetric weighting matrices.

Remark 1. In order to simplify the derivation, the models in (5) and (6) are assumed widely-linear, that is, according to the $\mathbb{HR}$-calculus framework, they are linear in $\mathbf{x}_n^i$, $\mathbf{x}_n^j$, $\mathbf{x}_n^k$, and $\mathbf{x}_n$. However, owing to the widely-linear setting, nonlinear functions can be accommodated using their Taylor expansions [41]. The same statement follows for the control problem in Section III-C.

IV. Quaternion-Valued Distributed Kalman Filter

A. Filter Derivation

In this section, the optimal solution to the filtering problem in Section III-B is constructed; then, a framework for approximating this optimal solution at each agent in a distributed fashion is derived. Initially, the filtering problem in Section III-B is considered from a network-wide perspective, where observation vectors are organized in a column vector so that

$$\mathbf{y}_{col,n} = [\mathbf{y}^{a\mathsf{T}}_{1,n}, \dots, \mathbf{y}^{a\mathsf{T}}_{|\mathcal{N}|,n}]^{\mathsf{T}}. \tag{10}$$

From (6) and (10) it can be interpreted that

$$\mathbf{y}_{col,n} = \mathbf{H}_{col,n}\mathbf{x}^a_n + \boldsymbol{\omega}_{col,n} \tag{11}$$

where

$$\begin{aligned}\mathbf{H}_{col,n} =& [\mathbf{H}^{a\mathsf{T}}_{1,n}, \dots, \mathbf{H}^{a\mathsf{T}}_{|\mathcal{N}|,n}]^{\mathsf{T}}\\ \boldsymbol{\omega}_{col,n} =& [\boldsymbol{\omega}^{a\mathsf{T}}_{1,n}, \dots, \boldsymbol{\omega}^{a\mathsf{T}}_{|\mathcal{N}|,n}]^{\mathsf{T}}.\end{aligned}$$

Now, the filtering problem becomes that of tracking the state vector modeled via the dynamic system in (5), given the observation vector in (10) and observation function in (11). The optimal solution to this problem comes in the form of centralized quaternion Kalman filter (CQKF) the operation of which are summarized in the following [6,41]:
Initialize with:

$$\hat{\mathbf{x}}^a_{0|0} = \mathsf{E}\left\{\mathbf{x}^a_0\right\} \tag{12a}$$

$$\mathbf{M}^a_{0|0} = \mathsf{E}\left\{(\mathbf{x}^a_0 - \mathsf{E}\left\{\mathbf{x}^a_0\right\})(\mathbf{x}^a_0 - \mathsf{E}\left\{\mathbf{x}^a_0\right\})^{\mathsf{H}}\right\} \tag{12b}$$

Model update:

$$\hat{\mathbf{x}}^a_{n|n-1} = \mathbf{A}^a_n \hat{\mathbf{x}}^a_{n-1|n-1} \tag{12c}$$

$$\mathbf{M}^a_{n|n-1} = \mathbf{A}^a_n \mathbf{M}^a_{n-1|n-1}\mathbf{A}^{a\mathsf{H}}_n + \mathbf{C}_{\boldsymbol{\nu}^a_n} \tag{12d}$$

Measurement update:

$$\mathbf{M}^{a^{-1}}_{n|n} = \mathbf{M}^{a^{-1}}_{n|n-1} + \mathbf{H}^{\mathsf{H}}_{col,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}}\mathbf{H}_{col,n} \tag{12e}$$

$$\mathbf{G}_n = \mathbf{M}^a_{n|n}\mathbf{H}^{\mathsf{H}}_{col,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}} \tag{12f}$$

$$\hat{\mathbf{x}}^a_{n|n} = \hat{\mathbf{x}}^a_{n|n-1} + \mathbf{G}_n\big(\mathbf{y}_{col,n} - \mathbf{H}_{col,n}\hat{\mathbf{x}}^a_{n|n-1}\big) \tag{12g}$$

where $\hat{\mathbf{x}}^a_{n|n-1}$ and $\hat{\mathbf{x}}^a_{n|n}$ denote respectively the *a priori* and *a posteriori* estimates of $\mathbf{x}^a_n$, while $\mathsf{E}\left\{\boldsymbol{\omega}_{col,n}\boldsymbol{\omega}^{\mathsf{H}}_{col,n}\right\} = \mathbf{C}_{\boldsymbol{\omega}_{col,n}}$.

Although the CQKF is optimal in the mean square error sense, as it can incorporate all the available observation information in the network, its operation requires the transfer of all observation vectors to the central processing unit. This burdens the network with complex communication protocols and makes the algorithm vulnerable to the failure of the central processing unit. Taking into account that in agent networks communication is restricted to the neighborhood of each agent, next we aim to mirror the operations of the CQKF within a distributed setting.

Considering the expression in (7), the covariance matrix $\mathbf{C}_{\boldsymbol{\omega}_{col,n}}$ has a block diagonal structure given by

$$\begin{aligned}\mathbf{C}_{\boldsymbol{\omega}_{col,n}} =& \mathsf{E}\left\{\boldsymbol{\omega}_{col,n}\boldsymbol{\omega}^{\mathsf{H}}_{col,n}\right\}\\ =& \text{block-diag}\{\mathbf{C}_{\boldsymbol{\omega}^a_{l,n}} : \forall l \in \mathcal{N}\}\end{aligned} \tag{13}$$

Therefore, from (12e), we have

$$\begin{aligned}\mathbf{M}^{a^{-1}}_{n|n} =& \mathbf{M}^{a^{-1}}_{n|n-1} + \sum_{\forall l \in \mathcal{N}} \mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\mathbf{H}^a_{l,n}\\ =& \frac{1}{|\mathcal{N}|}\sum_{\forall l \in \mathcal{N}} \boldsymbol{\Gamma}^a_{l,n}\end{aligned} \tag{14}$$

where

$$\boldsymbol{\Gamma}^a_{l,n} = \mathbf{M}^{a^{-1}}_{n|n-1} + |\mathcal{N}|\mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\mathbf{H}^a_{l,n}. \tag{15}$$

Replacing (12f) into (12g) and after some mathematical manipulation, it follows that

$$\begin{aligned}\hat{\mathbf{x}}^a_{n|n} =& \hat{\mathbf{x}}^a_{n|n-1} + \mathbf{M}^a_{n|n}\mathbf{H}^{\mathsf{H}}_{col,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}}\mathbf{y}_{col,n}\\ &- \mathbf{M}^a_{n|n}\mathbf{H}^{\mathsf{H}}_{col,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}}\mathbf{H}_{col,n}\hat{\mathbf{x}}^a_{n|n-1}.\end{aligned} \tag{16}$$

Once more, taking into account the block-diagonal structure of $\mathbf{C}_{\boldsymbol{\omega}_{col,n}}$ given in (13) we have

$$\begin{aligned}\mathbf{H}^{\mathsf{H}}_{col,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}}\mathbf{y}_{col,n} =& \sum_{\forall l \in \mathcal{N}} \mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\mathbf{y}^a_{l,n}\\ \mathbf{H}^{\mathsf{H}}_{col,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}}\mathbf{H}_{col,n} =& \sum_{\forall l \in \mathcal{N}} \mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\mathbf{H}^a_{l,n}.\end{aligned} \tag{17}$$

Now, substituting the expression in (17) into (16) allows the *a posteriori* estimate of the state vector to be calculated by

$$\begin{aligned}\hat{\mathbf{x}}^a_{n|n} =& \hat{\mathbf{x}}^a_{n|n-1} + \mathbf{M}^a_{n|n}\left(\sum_{\forall l \in \mathcal{N}} \mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\mathbf{y}^a_{l,n}\right)\\ &- \mathbf{M}^a_{n|n}\left(\sum_{\forall l \in \mathcal{N}} \mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\mathbf{H}^a_{l,n}\right)\hat{\mathbf{x}}^a_{n|n-1}\\ =& \hat{\mathbf{x}}^a_{n|n-1} + \sum_{\forall l \in \mathcal{N}} \mathbf{M}^a_{n|n}\mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\big(\mathbf{y}^a_{l,n} - \mathbf{H}^a_{l,n}\hat{\mathbf{x}}^a_{n|n-1}\big).\end{aligned} \tag{18}$$

The expression in (18) can be formulated in a more elegant fashion as

$$\hat{\mathbf{x}}^a_{n|n} = \frac{1}{|\mathcal{N}|}\sum_{\forall l \in \mathcal{N}} \boldsymbol{\psi}^a_{l,n} \tag{19}$$

where

$$\begin{aligned}\boldsymbol{\psi}^a_{l,n} =& \hat{\mathbf{x}}^a_{n|n-1}\\ &+ |\mathcal{N}|\mathbf{M}^a_{n|n}\mathbf{H}^{a\mathsf{H}}_{l,n}\mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}}\big(\mathbf{y}^a_{l,n} - \mathbf{H}^a_{l,n}\hat{\mathbf{x}}^a_{n|n-1}\big).\end{aligned} \tag{20}$$

Remark 2. In (20) and (15) the updating parameters are scaled by a factor of $|\mathcal{N}|$ in order to preserve the equivalence between expressions in (19) and (14) with those in (12e) and (12g).

In essence, the expressions in (14)-(20) show that the operations of the CQKF can be mirrored in a distributed fashion through the averaging procedures in (14) and (19). Thus, in order to present a distributed formulation of the CQKF and based on the work in [51,52], a framework for approximating the averages in (14) and (19) is established. To this end, effects of consensus filters in [51,52] on quaternion-valued matrices with an augmented composition is next investigated. This also provides the format with which the consensus filters will be implemented in this work and forms the basis of our performance analysis.

Given the set of augmented quaternion-valued matrices $\{\mathbf{F}^a_{l_0} : l = 1, 2, \ldots, |\mathcal{N}|\}$ as inputs of the following iterative consensus filter

$$\mathbf{F}^a_{l_\iota} = \mathbf{F}^a_{l_{\iota-1}} + \sum_{\forall m \in \mathcal{N}_l} w_{l,m}\big(\mathbf{F}^a_{m_{\iota-1}} - \mathbf{F}^a_{l_{\iota-1}}\big) \tag{21}$$

where $\mathbf{F}^a_{l_\iota}$ is the output of the iterative consensus filter at node l after ι iterations and $w_{l,m}$ is a positive real-valued weight. The iterations of the consensus filter in (21) can alternatively be expressed in a network-wide formulation as

$$\boldsymbol{\mathcal{F}}_\iota = (\boldsymbol{\mathcal{W}} \otimes \mathbf{I})\boldsymbol{\mathcal{F}}_{\iota-1} = (\boldsymbol{\mathcal{W}} \otimes \mathbf{I})^\iota \boldsymbol{\mathcal{F}}_0 = (\boldsymbol{\mathcal{M}} \otimes \mathbf{I})\boldsymbol{\mathcal{F}}_0 \tag{22}$$

where $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{W}}^\iota$, while $\boldsymbol{\mathcal{F}}_\iota = [\mathbf{F}^{a\mathsf{T}}_{1_\iota}, \mathbf{F}^{a\mathsf{T}}_{2_\iota}, \ldots, \mathbf{F}^{a\mathsf{T}}_{|\mathcal{N}|_\iota}]^\mathsf{T}$ and the element on the l^{th} row and m^{th} column of $\boldsymbol{\mathcal{W}}$ is

$$\boldsymbol{\mathcal{W}}^{\{l,m\}} = \begin{cases} 1 - \sum\limits_{\forall \kappa \in \mathcal{N}_l \setminus l} w_{l,\kappa} & \text{if } m = l, \\ w_{l,m} & \text{if } m \in \mathcal{N}_l \setminus l, \\ 0 & \text{otherwise.} \end{cases}$$

In addition, if the weights are selected in a manner to also make $\boldsymbol{\mathcal{W}}$ doubly stochastic, from the work in [51], we have

$$\forall l, m \in \mathcal{N} : \boldsymbol{\mathcal{M}}^{\{l,m\}} \to 1/|\mathcal{N}| \;\; \text{as} \;\; \iota \to \infty. \tag{23}$$

where $\boldsymbol{\mathcal{M}}^{\{l,m\}}$ is the element on the l^{th} row and m^{th} column of $\boldsymbol{\mathcal{M}}$. Hence, the expressions in (21)-(23) indicate that

$$\lim_{\iota \to \infty} \mathbf{F}^a_{l_\iota} = \frac{1}{|\mathcal{N}|} \sum_{\forall m \in \mathcal{N}} \mathbf{F}^a_{m_0}$$

resulting in the average consensus filter (ACF) required to approximate the averages in (14) and (19). For brevity, the network-wide operation of the ACF after ι iterations is demonstrated through the schematic

$$\mathbf{F}^a_{l_\iota} \leftarrow \boxed{\text{ACF}} \leftarrow \{\mathbf{F}^a_{m_0} : \forall m \in \mathcal{N}\}$$

where $\{\mathbf{F}^a_{m_0} : \forall m \in \mathcal{N}\}$ are the network-wide inputs to the ACF and $\mathbf{F}^a_{l_\iota}$ is the output at node l after ι iterations of the expression in (21) and its required data exchanges.

Finally, the operations of the derived distributed Kalman filter are summarized in Algorithm 1, where $\hat{\mathbf{x}}^a_{l,n|n-1}$ and $\hat{\mathbf{x}}^a_{l,n|n}$ denote the *a priori* and *a posteriori* estimates of $\mathbf{x}^a_n$ at node l, while the ACF is iterated a predefined number of times in order to approximate the averages in (14) and (19) with sufficient accuracy.

Algorithm 1. Distributed Quaternion Kalman Filter (DQKF)

For nodes $l = \{1, \ldots, |\mathcal{N}|\}$:

Initialize with:

$$\hat{\mathbf{x}}^a_{l,0|0} = \mathsf{E}\{\mathbf{x}^a_0\} \tag{24a}$$

$$\mathbf{M}^a_{l,0|0} = \mathsf{E}\Big\{(\mathbf{x}^a_0 - \mathsf{E}\{\mathbf{x}^a_0\})(\mathbf{x}^a_0 - \mathsf{E}\{\mathbf{x}^a_0\})^\mathsf{H}\Big\} \tag{24b}$$

Model update:

$$\hat{\mathbf{x}}^a_{l,n|n-1} = \mathbf{A}^a_n \hat{\mathbf{x}}^a_{l,n-1|n-1} \tag{24c}$$

$$\mathbf{M}^a_{l,n|n-1} = \mathbf{A}^a_n \mathbf{M}^a_{l,n-1|n-1} \mathbf{A}^{a\mathsf{H}}_n + \mathbf{C}_{\boldsymbol{\nu}^a_n} \tag{24d}$$

Measurement update:

$$\boldsymbol{\Gamma}^a_{l,n} = \mathbf{M}^{a^{-1}}_{l,n|n-1} + |\mathcal{N}|\mathbf{H}^{a\mathsf{H}}_{l,n} \mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}} \mathbf{H}^a_{l,n} \tag{24e}$$

$$\mathbf{M}^{a^{-1}}_{l,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\boldsymbol{\Gamma}^a_{m,n} : \forall m \in \mathcal{N}\} \tag{24f}$$

$$\mathbf{G}^a_{l,n} = |\mathcal{N}|\mathbf{M}^a_{l,n|n} \mathbf{H}^{a\mathsf{H}}_{l,n} \mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}} \tag{24g}$$

$$\boldsymbol{\psi}^a_{l,n} = \hat{\mathbf{x}}^a_{l,n|n-1} + \mathbf{G}^a_{l,n}\big(\mathbf{y}^a_{l,n} - \mathbf{H}^a_{l,n}\hat{\mathbf{x}}^a_{l,n|n-1}\big) \tag{24h}$$

$$\hat{\mathbf{x}}^a_{l,n|n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\boldsymbol{\psi}^a_{m,n} : \forall m \in \mathcal{N}\} \tag{24i}$$

Remark 3. As the iterations of the ACF increase resulting in $\forall l, m \in \mathcal{N} : \boldsymbol{\mathcal{M}}^{\{l,m\}} \to 1/|\mathcal{N}|$; then, (24f) and (24i) become convergent to the summation in (14) and (19). Thus, it follows that $\forall l \in \mathcal{N} : \mathbf{M}^a_{l,n|n} \to \mathbf{M}^a_{n|n}$ and as a result $\hat{\mathbf{x}}^a_{l,n|n} \to \hat{\mathbf{x}}^a_{n|n}$. Indicating that performance levels comparable to that of the CQKF are attainable. This is also demonstrated through simulations in Section VI-A.

B. Stability and Performance Analysis

The aim of this section is to provide a better insight into derived DQKF operations and formulate a closed-form expression for the second-order error moment, commonly referred to in the literature as the MSD, which for node l at time instant n is given by

$$\text{MSD}_{l,n} = \mathsf{E}\Big\{\big(\mathbf{x}^a_n - \hat{\mathbf{x}}^a_{l,n|n}\big)^\mathsf{H}\big(\mathbf{x}^a_n - \hat{\mathbf{x}}^a_{l,n|n}\big)\Big\}. \tag{25}$$

In keeping with classical performance analysis and for simplicity of derivation, we consider the time-invariant case. This is the essence of *Assumptions* 1 and 2. To achieve this goal, the following standard conditions in Kalman filtering analysis are held to be true [6,15,53,54]:

Assumption 1: The state transition matrix becomes time invariant and state evolution noise becomes stationary,

$$\lim_{n\to\infty} \mathbf{A}^a_n = \mathbf{A}^a \quad \text{and} \quad \lim_{n\to\infty} \mathbf{C}_{\boldsymbol{\nu}^a_n} = \mathbf{C}_{\boldsymbol{\nu}^a}.$$

Assumption 2: The observation function of all agents become time invariant and the observation noise at all nodes in the network become stationary,

$$\forall l \in \mathcal{N} : \lim_{n\to\infty} \mathbf{H}^a_{l,n} = \mathbf{H}^a_l \quad \text{and} \quad \lim_{n\to\infty} \mathbf{C}_{\boldsymbol{\omega}^a_{l,n}} = \mathbf{C}_{\boldsymbol{\omega}^a_l}$$

resulting in $\lim\limits_{n\to\infty} \mathbf{C}_{\boldsymbol{\omega}_{col,n}} = \mathbf{C}_{\boldsymbol{\omega}_{col}}$.

Assumption 3: The matrix pairs $\forall l \in \mathcal{N} : \{\mathbf{A}^a_n, \mathbf{H}^a_{l,n}\}$ are jointly observable over the communication matrix $\boldsymbol{\mathcal{M}}$ and the matrix pair $\{\mathbf{A}^a_n, \mathbf{C}^{\frac{1}{2}}_{\boldsymbol{\nu}^a_n}\}$ is controllable.

Assumption 4: The matrices $\forall l \in \mathcal{N} : \{\mathbf{\Gamma}^a_{l,n}, \mathbf{M}^a_{l,n}\}$ remain positive definite and retain their augmented composition at all time instances.[2]

Consider the fusion of covariance information in (24f). Substituting the ACF operation from (22) into (24f) yields

$$\begin{bmatrix} \mathbf{M}^{a^{-1}}_{1,n|n} \\ \vdots \\ \mathbf{M}^{a^{-1}}_{|\mathcal{N}|,n|n} \end{bmatrix} = (\boldsymbol{\mathcal{M}} \otimes \mathbf{I}) \begin{bmatrix} \mathbf{\Gamma}^a_{1,n} \\ \vdots \\ \mathbf{\Gamma}^a_{|\mathcal{N}|,n} \end{bmatrix}. \tag{26}$$

Formulating (26) from the perspective of agent $l \in \mathcal{N}$ gives

$$\mathbf{M}^{a^{-1}}_{l,n|n} = \sum_{\forall m \in \mathcal{N}} \boldsymbol{\mathcal{M}}^{\{l,m\}} \mathbf{\Gamma}^a_{m,n}. \tag{27}$$

The expression in (27) can be rearranged as

$$\begin{aligned}\mathbf{M}^{a^{-1}}_{l,n|n} =& \boldsymbol{\mathcal{M}}^{\{l,l\}} \mathbf{M}^{a^{-1}}_{l,n|n-1} + \boldsymbol{\mathcal{M}}^{\{l,l\}} |\mathcal{N}| \mathbf{H}^{a\mathsf{H}}_{l,n} \mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{l,n}} \mathbf{H}^a_{l,n} \\ &+ \sum_{\substack{\forall m \in \mathcal{N} \\ m \neq l}} \boldsymbol{\mathcal{M}}^{\{l,m\}} \mathbf{\Gamma}^a_{m,n}. \end{aligned} \tag{28}$$

Alternatively, from replacing $\{\mathbf{\Gamma}^a_{m,n} : \forall m \in \mathcal{N}\}$, as given in (24e), into (28), we have

$$\begin{aligned}\mathbf{M}^{a^{-1}}_{l,n|n} =& \mathbf{M}^{a^{-1}}_{l,n|n-1} + \sum_{\forall m \in \mathcal{N}} \boldsymbol{\mathcal{M}}^{\{l,m\}} |\mathcal{N}| \mathbf{H}^{a\mathsf{H}}_{m,n} \mathbf{C}^{-1}_{\boldsymbol{\omega}^a_{m,n}} \mathbf{H}^a_{m,n} \\ &+ \sum_{\forall m \in \mathcal{N}} \boldsymbol{\mathcal{M}}^{\{l,m\}} \left(\mathbf{M}^{a^{-1}}_{m,n|n-1} - \mathbf{M}^{a^{-1}}_{l,n|n-1} \right). \end{aligned} \tag{29}$$

Substituting (24d) into (29) allows evolution of the matrix set $\{\mathbf{M}^a_{l,n|n}, \mathbf{M}^a_{l,n|n-1} : \forall l \in \mathcal{N}\}$ to be expressed as

$$\begin{aligned} &\mathbf{M}^a_{l,n|n-1} = f_{l,n}\left(\mathbf{\Phi}^a_{l,n-1}\right) \\ &\mathbf{\Phi}^a_{l,n} \leftarrow \boxed{\text{ACF}} \leftarrow \left\{ \mathbf{M}^{a^{-1}}_{m,n|n-1} : \forall m \in \mathcal{N} \right\} \end{aligned} \tag{30}$$

where

$$f_{l,n}(\mathbf{\Theta}^a) = \mathbf{A}^a_n \left(\mathbf{\Theta}^{a^{-1}}_{l,n} + \boldsymbol{\mathcal{H}}^{\mathsf{H}}_{l,n} \mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}} \boldsymbol{\mathcal{H}}_{l,n} \right)^{-1} \mathbf{A}^{a\mathsf{H}}_n + \mathbf{C}_{\boldsymbol{\nu}^a_n}$$

$$\boldsymbol{\mathcal{H}}_{l,n} = \sqrt{|\mathcal{N}|} \left[\sqrt{\boldsymbol{\mathcal{M}}^{\{l,1\}}} \mathbf{H}^{a\mathsf{T}}_{1,n}, \ldots, \sqrt{\boldsymbol{\mathcal{M}}^{\{l,|\mathcal{N}|\}}} \mathbf{H}^{a\mathsf{T}}_{|\mathcal{N}|,n} \right]^{\mathsf{T}}.$$

Now, consider the class of recursive functions

$$\forall l \in \mathcal{N} : \mathbf{\Upsilon}^a_{l,n} = f_{l,n}\left(\mathbf{\Upsilon}^a_{l,n-1}\right). \tag{31}$$

From the made assumptions, it follows that matrices $\{\boldsymbol{\mathcal{H}}_{l,n}, \mathbf{C}^{-1}_{\boldsymbol{\omega}_{col,n}}, \mathbf{C}_{\boldsymbol{\nu}^a_n}\}$, and by extension the function $f_{l,n}(\cdot)$, become time invariant. Thus, (31) constitutes an algebraic Riccati equation that converge to a unique stabilizing solution given that $\{\mathbf{A}^a_n, \boldsymbol{\mathcal{H}}_{l,n}\}$ are detectable and the matrices $\{\mathbf{A}_n, \mathbf{C}^{\frac{1}{2}}_{\boldsymbol{\nu}^a}\}$ are stabilizable, assured under *Assumption* 3.

Proposition 1. *If the Reccati recursions in (31) converge to unique stabilizing solutions; then, the recursion in (30) is also convergent to a set of stabilizing matrices.*

Proof of Proposition 1: If *Assumption* 3 holds; then, from previous results [6], (31) converges to a unique stabilizing solution, i.e., as $n \to \infty$, $\forall l \in \mathcal{N} : \mathbf{\Upsilon}^a_{l,n} - \mathbf{\Upsilon}^a_{l,n-1} \to \mathbf{0}$.

Given that the recursive equations in (31) are convergent to unique stabling solutions; then, from (30), it follows that the spectral radius of $\mathbf{\Phi}^a_{l,n} - \mathbf{\Phi}^a_{l,n-1}$ must be less than that of $\mathbf{M}^a_{l,n+1|n} - \mathbf{M}^a_{l,n|n-1}$. Furthermore, in the context of the ACF, we have

$$\mathbf{\Phi}^a_{l,n} - \mathbf{\Phi}^a_{l,n-1} \leftarrow \boxed{\text{ACF}} \leftarrow \{\mathbf{\Delta}_{m,n} : \forall m \in \mathcal{N}\} \tag{32}$$

where $\mathbf{\Delta}_{m,n} = \mathbf{M}^{a^{-1}}_{m,n|n-1} - \mathbf{M}^{a^{-1}}_{m,n-1|n-2}$. Since the ACF weights were selected to make $\boldsymbol{\mathcal{W}}$ doubly stochastic; then, from (22) it follows that the ACF is a non-expanding operator with regards to input set $\{\mathbf{\Delta}_{m,m} : \forall m \in \mathcal{N}\}$. Therefore

$$\forall l \in \mathcal{N} : \quad \mathbf{M}^a_{l,n+1|n} \to \mathbf{M}^a_{l,n|n-1} \ \text{ as } \ n \to \infty \tag{33}$$

indicating convergence to a stabilizing solution. ■

Proposition 2. *The stabilizing solutions in Proposition 1 are unique.*

Proof of Proposition 2: Assume that both matrix sequences $\{\mathbf{M}^a_{l,n|n-1}, \mathbf{\Phi}^a_{l,n} : \forall l \in \mathcal{N}\}$ and $\{\mathbf{M}^{a'}_{l,n|n-1}, \mathbf{\Phi}^{a'}_{l,n} : \forall l \in \mathcal{N}\}$ stabilize (30). Then, replacing matrices $\{\mathbf{M}^a_{l,n+1|n}, \mathbf{\Phi}^a_{l,n}\}$ in (32)-(33) with $\{\mathbf{M}^{a'}_{l,n|n-1}, \mathbf{\Phi}^{a'}_{l,n-1}\}$ and following the same line of reasoning used in the proof of Proposition 1, it can be concluded that

$$\forall l \in \mathcal{N} : \quad \mathbf{M}^{a'}_{l,n|n-1} \to \mathbf{M}^a_{l,n|n-1} \ \text{ as } \ n \to \infty$$

indicating uniqueness of the stabilizing solutions. ■

Remark 4. In the steady-state case, where the matrix set $\{\mathbf{M}^a_{l,n|n-1} : \forall l \in \mathcal{N}\}$ have converged to their stabilizing values, the need for implementing ACF operations with regards to $\{\mathbf{\Gamma}^a_{l,n} : \forall l \in \mathcal{N}\}$ is negated. This reduces communication traffic over the network, as the local state vector estimates, $\boldsymbol{\psi}_{l,n}$, are the only variables that are shared. Note that the complexity analysis and implementation lemmas introduced in [41] for single agent Kalman filtering is applicable in the setting of Algorithm 1. The extension from the single agent case to the distributed case is straightforward, and hence, has been omitted.

In order to formulate the MSD of state vector estimates at each node, we consider the intermediate state vector estimation error at node l at time instant n given by

$$\boldsymbol{\epsilon}^a_{l,n} = \mathbf{x}^a_n - \boldsymbol{\psi}^a_{l,n}. \tag{34}$$

Substituting (24h) into the expression in (34) yields

$$\boldsymbol{\epsilon}^a_{l,n} = \mathbf{x}^a_n - \hat{\mathbf{x}}^a_{l,n|n-1} - \mathbf{G}^a_{l,n} \left(\mathbf{y}^a_{l,n} - \mathbf{H}^a_{l,n} \hat{\mathbf{x}}^a_{l,n|n-1} \right) \tag{35}$$

which given the observation model in (6) is rearranged to give

$$\boldsymbol{\epsilon}^a_{l,n} = \left(\mathbf{I} - \mathbf{G}^a_{l,n} \mathbf{H}^a_{l,n} \right) \boldsymbol{\epsilon}^a_{l,n|n-1} - \mathbf{G}^a_{l,n} \boldsymbol{\omega}^a_{l,n} \tag{36}$$

where $\boldsymbol{\epsilon}^a_{l,n|n-1} = \mathbf{x}^a_n - \hat{\mathbf{x}}^a_{l,n|n-1}$. Furthermore, considering the state evolution model in (5) it becomes apparent that

$$\boldsymbol{\epsilon}^a_{l,n|n-1} = \mathbf{A}^a_n \boldsymbol{\epsilon}^a_{l,n-1|n-1} + \boldsymbol{\nu}^a_n. \tag{37}$$

[2]This holds true as the matrices in question are calculated from the summation of other positive definite and positive semi-definite matrices with augmented compositions. In practice, if need be, at each time instance these matrices can be replaced with their nearest augmented positive definite approximations.

Now, replacing (37) into (36) yields

$$\begin{aligned}\boldsymbol{\epsilon}^a_{l,n} =& \left(\mathbf{I}-\mathbf{G}^a_{l,n}\mathbf{H}^a_{l,n}\right)\mathbf{A}^a_n\boldsymbol{\epsilon}^a_{l,n-1|n-1}\\ &+\left(\mathbf{I}-\mathbf{G}^a_{l,n}\mathbf{H}^a_{l,n}\right)\boldsymbol{\nu}^a_n-\mathbf{G}^a_{l,n}\boldsymbol{\omega}^a_{l,n}.\end{aligned} \tag{38}$$

Fully capturing the evolution of state vector estimation error terms requires close scrutiny of the ACF impact on state vector estimation errors. This is achieved through formulating (38) in a network-wide format as

$$\boldsymbol{\mathcal{E}}_n = \mathcal{P}_n\boldsymbol{\mathcal{E}}_{n-1|n-1}+\mathcal{Q}_n\boldsymbol{\nu}_{col,n}-\mathcal{G}_n\boldsymbol{\omega}_{col,n} \tag{39}$$

where $\{\mathcal{P}_n,\mathcal{Q}_n,\mathcal{G}_n\}$ are the block-diagonal matrices

$$\begin{aligned}\mathcal{P}_n =& \text{block-diag}\{\left(\mathbf{I}-\mathbf{G}^a_{l,n}\mathbf{H}^a_{l,n}\right)\mathbf{A}^a_n : \forall l\in\mathcal{N}\}\\ \mathcal{Q}_n =& \text{block-diag}\{\left(\mathbf{I}-\mathbf{G}^a_{l,n}\mathbf{H}^a_{l,n}\right) : \forall l\in\mathcal{N}\}\\ \mathcal{G}_n =& \text{block-diag}\{\mathbf{G}^a_{l,n} : \forall l\in\mathcal{N}\}\end{aligned}$$

while $\boldsymbol{\nu}_{col,n} = [\boldsymbol{\nu}^{a\mathsf{T}}_n,\ldots,\boldsymbol{\nu}^{a\mathsf{T}}_n]^{\mathsf{T}}$, whereas

$$\boldsymbol{\mathcal{E}}_n = \begin{bmatrix}\boldsymbol{\epsilon}^a_{1,n}\\ \vdots\\ \boldsymbol{\epsilon}^a_{|\mathcal{N}|,n}\end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{E}}_{n-1|n-1} = \begin{bmatrix}\boldsymbol{\epsilon}^a_{1,n-1|n-1}\\ \vdots\\ \boldsymbol{\epsilon}^a_{|\mathcal{N}|,n-1|n-1}\end{bmatrix}.$$

Impact of the ACF on state vector estimation error terms can now be rigorously formulated using the expressions in (22) and (39) culminating in the network-wide regressive expression for state vector estimation error given by

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n} =& (\boldsymbol{\mathcal{M}}\otimes\mathbf{I})\boldsymbol{\mathcal{E}}_n\\ =& \boldsymbol{\mathcal{P}}_n\boldsymbol{\mathcal{E}}_{n-1|n-1}+\boldsymbol{\mathcal{Q}}_n\boldsymbol{\nu}_{col,n}-\boldsymbol{\mathcal{G}}_n\boldsymbol{\omega}_{col,n}\end{aligned} \tag{40}$$

where

$$\boldsymbol{\mathcal{P}}_n = (\boldsymbol{\mathcal{M}}\otimes\mathbf{I})\mathcal{P}_n,\ \boldsymbol{\mathcal{Q}}_n = (\boldsymbol{\mathcal{M}}\otimes\mathbf{I})\mathcal{Q}_n, \text{ and } \boldsymbol{\mathcal{G}}_n = (\boldsymbol{\mathcal{M}}\otimes\mathbf{I})\mathcal{G}_n.$$

Remark 5. Taking the statistical expectation of (40) and accounting for the fact that

$$\mathsf{E}\left\{\boldsymbol{\nu}_n\right\} = 0 \quad \text{and} \quad \forall l\in\mathcal{N} : \mathsf{E}\left\{\boldsymbol{\omega}_{l,n}\right\} = 0$$

we have

$$\mathsf{E}\left\{\boldsymbol{\mathcal{E}}_{n|n}\right\} = \boldsymbol{\mathcal{P}}_n\mathsf{E}\left\{\boldsymbol{\mathcal{E}}_{n-1|n-1}\right\} = \left(\prod_{i=1}^{n}\boldsymbol{\mathcal{P}}_i\right)\mathsf{E}\left\{\boldsymbol{\mathcal{E}}_{0|0}\right\}. \tag{41}$$

Thus, given the initialization condition in Algorithm 1, that is $\forall l\in\mathcal{N} : \hat{\mathbf{x}}^a_{l,0|0} = \mathsf{E}\{\mathbf{x}^a_0\}$, the expression in (41) proves that the developed DQKF operates in an unbiased fashion.

The expression in (40) is now used to formulate the network-wide square deviation term as

$$\begin{aligned}\boldsymbol{\mathcal{E}}_{n|n}\boldsymbol{\mathcal{E}}^{\mathsf{H}}_{n|n} =& \boldsymbol{\mathcal{P}}_n\boldsymbol{\mathcal{E}}_{n-1|n-1}\boldsymbol{\mathcal{E}}^{\mathsf{H}}_{n-1|n-1}\boldsymbol{\mathcal{P}}^{\mathsf{H}}_n\\ &+\boldsymbol{\mathcal{Q}}_n\boldsymbol{\nu}_{col,n}\boldsymbol{\nu}^{\mathsf{H}}_{col,n}\boldsymbol{\mathcal{Q}}^{\mathsf{T}}_n+\boldsymbol{\mathcal{G}}_n\boldsymbol{\omega}_{col,n}\boldsymbol{\omega}^{\mathsf{H}}_{col,n}\boldsymbol{\mathcal{G}}^{\mathsf{H}}_n\\ &-\boldsymbol{\mathcal{Q}}_n\boldsymbol{\nu}_{col,n}\boldsymbol{\omega}^{\mathsf{H}}_{col,n}\boldsymbol{\mathcal{G}}^{\mathsf{H}}_n-\boldsymbol{\mathcal{G}}_n\boldsymbol{\omega}_{col,n}\boldsymbol{\nu}^{\mathsf{H}}_{col,n}\boldsymbol{\mathcal{Q}}^{\mathsf{H}}_n\\ &-\boldsymbol{\mathcal{G}}_n\boldsymbol{\omega}_{col,n}\boldsymbol{\mathcal{E}}^{\mathsf{H}}_{n-1|n-1}\boldsymbol{\mathcal{P}}^{\mathsf{H}}_n-\boldsymbol{\mathcal{P}}_n\boldsymbol{\mathcal{E}}_{n-1|n-1}\boldsymbol{\omega}^{\mathsf{H}}_{col,n}\boldsymbol{\mathcal{G}}^{\mathsf{H}}_n\\ &+\boldsymbol{\mathcal{Q}}_n\boldsymbol{\nu}_{col,n}\boldsymbol{\mathcal{E}}^{\mathsf{H}}_{n-1|n-1}\boldsymbol{\mathcal{P}}^{\mathsf{H}}_n+\boldsymbol{\mathcal{P}}_n\boldsymbol{\mathcal{E}}_{n-1|n-1}\boldsymbol{\nu}^{\mathsf{H}}_{col,n}\boldsymbol{\mathcal{Q}}^{\mathsf{H}}_n.\end{aligned} \tag{42}$$

Given that $\boldsymbol{\nu}^a_n$ and $\{\boldsymbol{\omega}^a_{l,n} : \forall l\in\mathcal{N}\}$ are zero-mean white Gaussian random variables that are also independent from $\{\boldsymbol{\epsilon}^a_{l,n-1|n-1} : \forall l\in\mathcal{N}\}$; then, taking the statistical expectation of the expression in (42) yields

$$\boldsymbol{\Sigma}_n = \boldsymbol{\mathcal{P}}_n\boldsymbol{\Sigma}_{n-1}\boldsymbol{\mathcal{P}}^{\mathsf{H}}_n+\boldsymbol{\mathcal{Q}}_n\mathbf{C}_{\boldsymbol{\nu}_{col,n}}\boldsymbol{\mathcal{Q}}^{\mathsf{H}}_n+\boldsymbol{\mathcal{G}}_n\mathbf{C}_{\boldsymbol{\omega}_{col,n}}\boldsymbol{\mathcal{G}}^{\mathsf{H}}_n \tag{43}$$

where $\boldsymbol{\Sigma}_n = \mathsf{E}\left\{\boldsymbol{\mathcal{E}}_{n|n}\boldsymbol{\mathcal{E}}^{\mathsf{H}}_{n|n}\right\}$. Furthermore, from Proposition 1 and Proposition 2, at steady-state the matrices $\{\mathbf{M}^a_{l,n|n} : \forall l\in\mathcal{N}\}$ converge to the stabilizing solution of (30) and become time invariant. From (24g) it follows that if $\{\mathbf{M}^a_{l,n|n} : \forall l\in\mathcal{N}\}$ converge and the assumptions made in this section hold true; then, the matrices $\{\mathbf{G}^a_{l,n} : \forall l\in\mathcal{N}\}$ become time invariant. Furthermore, the expressions in (38)-(40) and (42)-(43) indicate that if the matrices $\{\mathbf{G}^a_{l,n}, l\in\mathcal{N}\}$ are time invariant; then, the matrices $\{\boldsymbol{\mathcal{P}}_n,\boldsymbol{\mathcal{G}}_n,\boldsymbol{\mathcal{Q}}_n\}$ also become time invariant, that is

$$\lim_{n\to\infty}\boldsymbol{\mathcal{P}}_n = \boldsymbol{\mathcal{P}},\ \lim_{n\to\infty}\mathcal{Q}_n = \boldsymbol{\mathcal{Q}}, \text{ and } \lim_{n\to\infty}\boldsymbol{\mathcal{G}}_n = \boldsymbol{\mathcal{G}}.$$

Thus, if the made assumption hold true, as $n\to\infty$, the expression in (43) turns into the quaternion-valued discrete-time Lyapunov equation

$$\boldsymbol{\Sigma} = \boldsymbol{\mathcal{P}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{P}}^{\mathsf{H}}+\boldsymbol{\mathcal{Q}}\mathbf{C}_{\boldsymbol{\nu}_{col}}\boldsymbol{\mathcal{Q}}^{\mathsf{H}}+\boldsymbol{\mathcal{G}}\mathbf{C}_{\boldsymbol{\omega}_{col}}\boldsymbol{\mathcal{G}}^{\mathsf{H}}. \tag{44}$$

Now, invoking the framework put forth in [6], the expression in (44) converges to the unique stabling solution

$$\mathsf{vec}\left\{\boldsymbol{\Sigma}^{\mathbb{HR}}\right\} = \left(\mathbf{I}-\boldsymbol{\mathcal{P}}^{\mathbb{HR}}\otimes\boldsymbol{\mathcal{P}}^{\mathbb{HR}}\right)^{-1}\mathsf{vec}\left\{\boldsymbol{\Upsilon}^{\mathbb{HR}}\right\} \tag{45}$$

where $\boldsymbol{\Upsilon} = \boldsymbol{\mathcal{Q}}\mathbf{C}_{\boldsymbol{\nu}_{col}}\boldsymbol{\mathcal{Q}}^{\mathsf{H}}+\boldsymbol{\mathcal{G}}\mathbf{C}_{\boldsymbol{\omega}_{col}}\boldsymbol{\mathcal{G}}^{H}$ and $\{\boldsymbol{\Sigma}^{\mathbb{HR}},\boldsymbol{\mathcal{P}}^{\mathbb{HR}},\boldsymbol{\Upsilon}^{\mathbb{HR}}\}$ are duals of $\{\boldsymbol{\Sigma},\boldsymbol{\mathcal{P}},\boldsymbol{\Upsilon}\}$ in the quaternion isomorphic matrix algebra as defined in (1).

From the expression in (25) the MSD at each node can be formulated as

$$\text{MSD}_{l,n} = \mathsf{Tr}\left\{\mathsf{E}\left\{\boldsymbol{\epsilon}^a_{l,n|n}\boldsymbol{\epsilon}^{a\mathsf{H}}_{l,n|n}\right\}\right\}$$

where $\mathsf{E}\left\{\boldsymbol{\epsilon}^a_{l,n|n}\boldsymbol{\epsilon}^{a\mathsf{H}}_{l,n|n}\right\}$ is the l^{th} block-diagonal element of $\boldsymbol{\Sigma}_n$ as given in (43). In addition, from (40)-(45), in a steady-state mode, the MSD of each node converges to the block-diagonal elements of $\boldsymbol{\Sigma}$ as given in (45).

V. Quaternion-Valued Distributed Control

In this section, a distributed solution to the control problem formulated in Section III-C is derived. This is enabled via the comprehensive fusion of covariance information envisioned in the derived DQKF and essentially extends the duality between filtering and control (introduced in [11]) to quaternion-valued processes.

Given equations (8) and (9), the control problem in Section III-C is rearranged into the following network-wide constrained optimization problem:
Minimize:

$$\mathcal{J} = \mathbf{x}^{a\mathsf{H}}_N\mathbf{T}^a\mathbf{x}^a_N+\sum_{n=1}^{N-1}\left(\mathbf{x}^{a\mathsf{H}}_n\mathbf{Q}^a\mathbf{x}^a_n+\mathbf{u}^{\mathsf{H}}_{col,n}\boldsymbol{\mathcal{R}}\mathbf{u}_{col,n}\right) \tag{46a}$$

Through the widely-linear control law:

$$\mathbf{u}_{col,n} = \underbrace{[\mathbf{L}^{a\mathsf{T}}_{1,n},\ldots,\mathbf{L}^{a\mathsf{T}}_{|\mathcal{N}|,n}]^{\mathsf{T}}}_{\boldsymbol{\mathcal{L}}_n}\mathbf{x}^a_n \tag{46b}$$

Subject to dynamic constraint:

$$\mathbf{x}^a_{n+1} = \mathbf{A}^a\mathbf{x}^a_n+\boldsymbol{\mathcal{B}}\mathbf{u}_{col,n} \tag{46c}$$

where

$$\mathbf{u}_{col,n} = [\mathbf{u}_{1,n}^{a\mathsf{T}}, \ldots, \mathbf{u}_{|\mathcal{N}|,n}^{a\mathsf{T}}]^{\mathsf{T}} \quad \text{and} \quad \boldsymbol{\mathcal{B}} = [\mathbf{B}_1^a, \ldots, \mathbf{B}_{|\mathcal{N}|}^a]$$

while $\boldsymbol{\mathcal{R}} = \text{block-diag}\{\mathbf{R}_l^a : \forall l \in \mathcal{N}\}$.

The defined optimization problem is tackled by dividing the cost function in (46a) into time-local cost functions given by

$$J_n = \begin{cases} \mathbf{x}_N^{a\mathsf{H}}\mathbf{T}^a\mathbf{x}_N^a & \text{if } n = N, \\ \mathbf{x}_n^{a\mathsf{H}}\mathbf{Q}^a\mathbf{x}_n^a + \mathbf{u}_{col,n}^{\mathsf{H}}\boldsymbol{\mathcal{R}}\mathbf{u}_{col,n} & \text{otherwise.} \end{cases} \tag{47}$$

where for $n \neq N$ replacing (46b) into (47) gives

$$J_n = \mathbf{x}_n^{a\mathsf{H}}\mathbf{Q}^a\mathbf{x}_n^a + \mathbf{u}_{col,n}^{\mathsf{H}}\boldsymbol{\mathcal{R}}\mathbf{u}_{col,n} = \mathbf{x}_n^{a\mathsf{H}}\boldsymbol{\Psi}_n^a\mathbf{x}_n^a$$

with

$$\boldsymbol{\Psi}_n^a = \mathbf{Q}^a + \boldsymbol{\mathcal{L}}_n^{\mathsf{H}}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{L}}_n = \mathbf{Q}^a + \sum_{\forall l \in \mathcal{N}} \mathbf{L}_{l,n}^{a\mathsf{H}}\mathbf{R}_l^a\mathbf{L}_{l,n}^a.$$

In turn, this allows a time-accumulative cost function to be defined as

$$\mathcal{J}_\iota = \sum_{m=\iota}^{N} J_m \tag{48}$$

where for $\iota = N$, we have $\mathcal{J}_N = \mathbf{x}_N^{a\mathsf{H}}\mathbf{T}^a\mathbf{x}_N^a$.

Proposition 3. *There exists an augmented positive definite Hermitian symmetric matrix such that* $\mathcal{J}_\iota = \mathbf{x}_\iota^{a\mathsf{H}}\boldsymbol{\mathcal{V}}_\iota^a\mathbf{x}_\iota^a$.

Proof of Proposition 3: For the case of $\iota = N$, from (48) it follows that $\boldsymbol{\mathcal{V}}_N^a = \mathbf{T}^a$. It is now assumed that the augmented positive definite Hermitian symmetric matrix $\boldsymbol{\mathcal{V}}_\iota^a$ exist so that $\mathcal{J}_\iota = \mathbf{x}_\iota^{a\mathsf{H}}\boldsymbol{\mathcal{V}}_\iota^a\mathbf{x}_\iota^a$. Then, for the case of $\mathcal{J}_{\iota-1}$, we have

$$\begin{aligned} \mathcal{J}_{\iota-1} = &\sum_{m=\iota-1}^{N} J_m = J_{\iota-1} + \sum_{m=\iota}^{N} J_m \\ = &\mathbf{x}_{\iota-1}^{a\mathsf{H}}\boldsymbol{\Psi}_{\iota-1}^a\mathbf{x}_{\iota-1}^a + \mathbf{x}_\iota^{a\mathsf{H}}\boldsymbol{\mathcal{V}}_\iota^a\mathbf{x}_\iota^a \end{aligned} \tag{49}$$

where substituting (46b) and (46c) into (49) yields

$$\mathcal{J}_{\iota-1} = \mathbf{x}_{\iota-1}^{a\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota-1}^a\mathbf{x}_{\iota-1}^a \tag{50}$$

with

$$\boldsymbol{\mathcal{V}}_{\iota-1}^a = \boldsymbol{\Psi}_{\iota-1}^a + \left(\mathbf{A}^{a\mathsf{H}} + \boldsymbol{\mathcal{L}}_{\iota-1}^{\mathsf{H}}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\right)\boldsymbol{\mathcal{V}}_\iota^a\left(\mathbf{A}^a + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{L}}_{\iota-1}\right).$$

Thus, through induction in reverse time, the expressions (49) and (50) show that there exists a positive definite matrix $\boldsymbol{\mathcal{V}}_{\iota-1}^a$ so that $\mathcal{J}_{\iota-1} = \mathbf{x}_{\iota-1}^{a\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota-1}^a\mathbf{x}_{\iota-1}^a$, proving Proposition 3. ■

Proposition 3 allows the introduction of a discrete-time quaternion-valued Hamilton-Jacobi-Bellman[3] type framework for solving the formulated optimization problem. In this setting, moving backwards in time and assuming that control inputs for future time instances have been obtained, the task becomes that of calculating the control input at the previous time instant resulting in the following minimization task

$$2\mathcal{J}_\iota = \min_{\mathbf{u}_{col,\iota}} \left\{2\mathbf{x}_\iota^{a\mathsf{H}}\mathbf{Q}^a\mathbf{x}_\iota^a + 2\mathbf{u}_{col,\iota}^{\mathsf{H}}\boldsymbol{\mathcal{R}}\mathbf{u}_{col,\iota} + 2\mathcal{J}_{\iota+1}\right\} \tag{51}$$

where all elements have been scaled by 2 to compensate for the scales generated after differentiation. The minimum of (51) is achievable at

$$\mathbf{u}_{col,\iota} = -\left(\boldsymbol{\mathcal{R}} + \boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\boldsymbol{\mathcal{B}}\right)^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\mathbf{A}^a\mathbf{x}_\iota^a \tag{52}$$

[3]More information regarding the Hamilton-Jacobi-Bellman equation used in optimization and dynamic programming applications is available in [54].

where comparing (52) to (46b) makes it apparent that

$$\boldsymbol{\mathcal{L}}_\iota = -\left(\boldsymbol{\mathcal{R}} + \boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\boldsymbol{\mathcal{B}}\right)^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\mathbf{A}^a.$$

In addition, substituting (52) into (51) gives

$$\begin{aligned} \boldsymbol{\mathcal{V}}_\iota^a = &\mathbf{Q}^a + \mathbf{A}^{a\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\mathbf{A}^a \\ &- \mathbf{A}^{a\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\boldsymbol{\mathcal{B}}\left(\boldsymbol{\mathcal{R}} + \boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\boldsymbol{\mathcal{B}}\right)^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\mathbf{A}^a. \end{aligned} \tag{53}$$

The expression in (53) can be rearranged using the matrix inversion lemma to yield

$$\boldsymbol{\Lambda}_\iota^a = \left(\left(\mathbf{A}^{a\mathsf{H}}\boldsymbol{\Lambda}_{\iota+1}^a\mathbf{A}^a + \mathbf{Q}^a\right)^{-1} + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\right)^{-1} \tag{54}$$

where $\boldsymbol{\Lambda}_\iota^a$ is introduced as an intermediate variable so that $\boldsymbol{\mathcal{V}}_\iota^a = \mathbf{A}^{a\mathsf{H}}\boldsymbol{\Lambda}_{\iota+1}^a\mathbf{A}^a + \mathbf{Q}^a$. Given that $\boldsymbol{\mathcal{R}}$ is block diagonal the expression in (54) is simplified into

$$\boldsymbol{\Lambda}_\iota^{a^{-1}} = \left(\mathbf{A}^{a\mathsf{H}}\boldsymbol{\Lambda}_{\iota+1}^a\mathbf{A}^a + \mathbf{Q}^a\right)^{-1} + \sum_{\forall l \in \mathcal{N}} \mathbf{B}_l^a\mathbf{R}_l^{a^{-1}}\mathbf{B}_l^{a\mathsf{H}}. \tag{55}$$

Note the similarity between (55) and the Riccati equation governing the behaviour of the CQKF

$$\mathbf{M}_{n|n}^{a^{-1}} = \left(\mathbf{A}_n^a\mathbf{M}_{n-1|n-1}^{a^{-1}}\mathbf{A}_n^a\right)^{-1} + \mathbf{H}_{col,n}^{\mathsf{H}}\mathbf{C}_{\boldsymbol{\omega}_{col,n}}^{-1}\mathbf{H}_{col,n} \tag{56}$$

obtainable via substituting (12d) into (12e). Building upon the duality between (56) and (55), it follows that the values of $\{\boldsymbol{\Lambda}_\iota^a, \boldsymbol{\mathcal{V}}_\iota^a\}$ can be approximated through the framework derived for fusion of covariance information in the DQKF.

The control input vectors in (52) can alternatively be calculated as

$$\begin{aligned} \mathbf{u}_{col,\iota} &= -\left(\boldsymbol{\mathcal{R}} + \boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\boldsymbol{\mathcal{B}}\right)^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\mathbf{A}^a\mathbf{x}_\iota^a \\ &= \left(\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{R}}^{-1} - \boldsymbol{\mathcal{R}}^{-1}\right)\boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\mathcal{V}}_{\iota+1}^a\mathbf{A}^a\mathbf{x}_\iota^a \\ &= -\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\left(\boldsymbol{\mathcal{V}}_{\iota+1}^{a^{-1}} + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\right)^{-1}\mathbf{A}^a\mathbf{x}_\iota^a \end{aligned} \tag{57}$$

where replacing (54) and (55) into (57) yields

$$\mathbf{u}_{col,\iota} = -\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{B}}^{\mathsf{H}}\boldsymbol{\Lambda}_{\iota+1}^a\mathbf{A}^a\mathbf{x}_\iota^a. \tag{58}$$

The block-diagonal structure of $\boldsymbol{\mathcal{R}}$ allows the control input of each agent to be expressed through rearranging (58) as

$$\mathbf{u}_{l,\iota} = -\mathbf{R}_l^{a^{-1}}\mathbf{B}_l^{a\mathsf{H}}\boldsymbol{\Lambda}_{\iota+1}^a\mathbf{A}^a\mathbf{x}_\iota^a. \tag{59}$$

Applying the same approach used for filtering, the expression in (55) can be formulated in a distributed fashion as

$$\boldsymbol{\Lambda}_\iota^{a^{-1}} = \frac{1}{|\mathcal{N}|}\sum_{\forall l \in \mathcal{N}} \boldsymbol{\Theta}_{l,\iota}^a \tag{60}$$

with

$$\boldsymbol{\Theta}_{l,\iota}^a = \boldsymbol{\mathcal{V}}_\iota^{a^{-1}} + |\mathcal{N}|\mathbf{B}_l^a\mathbf{R}_l^{a^{-1}}\mathbf{B}_l^{a\mathsf{H}}. \tag{61}$$

Now, the operations of the quaternion-valued widely-linear quadrature regulator can be approximated in a decentralized manner through the distributed implementation of the summation in (60). The operations of such a decentralized quaternion-valued regulator are summarized in Algorithm 2, where $\{\hat{\boldsymbol{\mathcal{V}}}_{l,n}^a, \hat{\boldsymbol{\Lambda}}_{l,n}^a, \hat{\boldsymbol{\Theta}}_{l,n}^a\}$ are local estimates of $\{\boldsymbol{\mathcal{V}}_n^a, \boldsymbol{\Lambda}_n^a, \boldsymbol{\Theta}_n^a\}$ at node l.

Algorithm 2. Decentralized Quaternion-Valued Widely-Linear Quadrature Regulator

For nodes $l = \{1, \ldots, |\mathcal{N}|\}$:
Initialize with:

$$\forall l \in \mathcal{N} : \hat{\boldsymbol{\mathcal{V}}}^a_{l,N} = \mathbf{T}^a \tag{62a}$$

Estimate output of the Riccati equation:

$$\hat{\boldsymbol{\Theta}}^a_{l,n} = \hat{\boldsymbol{\mathcal{V}}}^{a^{-1}}_{l,n} + |\mathcal{N}|\mathbf{B}^a_l \mathbf{R}^{a^{-1}}_l \mathbf{B}^{a\mathsf{H}}_l \tag{62b}$$

$$\hat{\boldsymbol{\Lambda}}^{a^{-1}}_{l,n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\hat{\boldsymbol{\Theta}}^a_{m,n} : \forall m \in \mathcal{N}\} \tag{62c}$$

$$\hat{\boldsymbol{\mathcal{V}}}^a_{l,n-1} = \mathbf{A}^{a\mathsf{H}} \hat{\boldsymbol{\Lambda}}^a_{l,n} \mathbf{A}^a + \mathbf{Q}^a \tag{62d}$$

Calculate control vector sequences:

$$\mathbf{u}_{l,n} = -\mathbf{R}^{a^{-1}}_l \mathbf{B}^{a\mathsf{H}}_l \hat{\boldsymbol{\Lambda}}^a_{n+1} \mathbf{A}^a \mathbf{x}^a_n \tag{62e}$$

Remark 6. As the iterations of the ACF increase resulting in $\forall l, m \in \mathcal{N} : \boldsymbol{\mathcal{M}}^{\{l,m\}} \rightarrow 1/|\mathcal{N}|$; then,

$$\forall l \in \mathcal{N} : \{\hat{\boldsymbol{\mathcal{V}}}^a_{l,n}, \hat{\boldsymbol{\Lambda}}^a_{l,n}, \hat{\boldsymbol{\Theta}}^a_{l,n}\} \rightarrow \{\boldsymbol{\mathcal{V}}^a_n, \boldsymbol{\Lambda}^a_n, \boldsymbol{\Theta}^a_n\}.$$

Indicating that the optimal solution in (52) can be approximated with any desirable degree of accuracy. This is also demonstrated through simulations in Section VI-B.

VI. Simulations and Discussions

In this section, performance of the derived framework is illustrated using a general setting that is intended to be adoptable for use in a variety of applications. Furthermore, filtering performance of the derived framework is also demonstrated for tacking three-dimensional rotations induced via a chaotic system. In all simulations, the network of 28 nodes with the topology shown in Figure 1 was used.

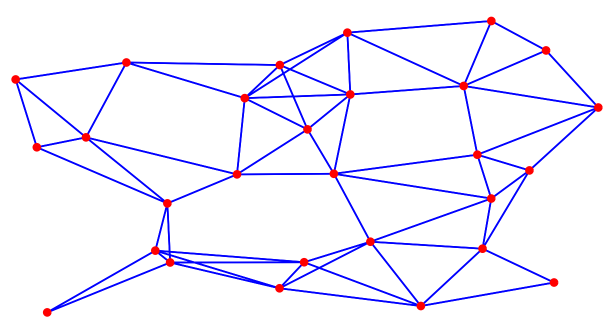

Fig. 1. The network of 28 nodes and 64 edges used for simulations.

A. Quaternion-Valued Filtering

The general equation of motion, applicable in a large number of scenarios, such as bearings-only and rotation tracking were considered. In this case, we have

$$\frac{\partial \varphi}{\partial t} = \phi \quad \text{and} \quad \frac{\partial \phi}{\partial t} = \nu \tag{63}$$

where ν represents the input to the system[4], φ represents the variable of interest (target location or orientation) with ϕ indicating its rate of change. Transforming the equations in (63) into its discrete-time mode yields the state-space model

$$\begin{bmatrix} \varphi_{n+1} \\ \phi_{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \varphi_n \\ \phi_n \end{bmatrix} + \underbrace{\begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix}}_{\mathbf{B}} \nu_n \tag{64}$$

where $\Delta T = 0.04\,$s denotes the sampling interval.

The input, ν_n, was considered to be a zero-mean white Gaussian random sequence with covariance $\mathsf{E}\{\nu_n \nu_n^{\mathsf{H}}\} = 0.8125$ and pseudo-covariances

$$\forall \zeta \in \{i, j, k\}, \mathsf{E}\{\nu_n \nu_n^{\zeta\mathsf{H}}\} = -0.1875$$

while the observation at each node was modeled as

$$\forall l \in \mathcal{N} : y_{l,n} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \varphi_{n+1} \\ \phi_{n+1} \end{bmatrix} + \omega_{l,n}$$

where the $\omega_{l,n}$ was considered to be a zero-mean white Gaussian random sequence with covariance $\mathsf{E}\{\omega_n \omega_n^{\mathsf{H}}\} = 0.5$ and vanishing pseudo-covariances.

The MSD achieved by each agent in the network for different number of iterations of the ACF is shown in Figure 2. Furthermore, as benchmarks, Figure 2 also includes MSD values predicted via the theoretical analysis in Section IV-B, the MSD achieved by the optimal CQKF, and the MSD achieved by each agent using the legacy filtering technique introduced in [6]. Note that the derived DQKF performed as predicted through the theoretical analysis in Section IV-B and was able to achieve smaller MSD levels than its predecessor. More importantly, as the number of ACF iterations grow, the MSD achieved by each agent becomes closer to that of the optimal CQKF. This provides a trade-off between performance (smaller MSD) and implementation complexity (ACF iterations), which is not available with the diffusion-based framework in [6].

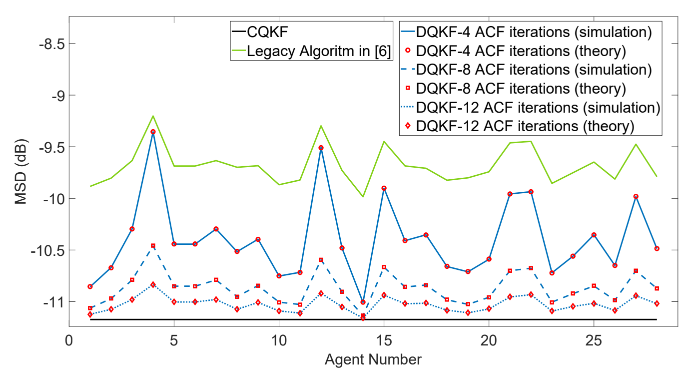

Fig. 2. The MSD performance of the derived DQKF across all 28 nodes of the network, obtained with different number of ACF iterations. Performance of the CQKF and the legacy algorithm in [6] are provided as benchmarks.

B. Quaternion-Valued Control

The control dual of the filtering problem in Section VI-A is now considered. The state vector and state transition matrix remain the same as in (64), whereas for odd numbered agents

$$\mathbf{B}^a_l = \left(\mathbf{I} \otimes \begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix}\right) \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

[4]In the case of bearing-only (*cf.* rotation) tracking applications, the input typically represents force (*cf.* torque) or acceleration (*cf.* angular acceleration).

and for even numbered agents

$$\mathbf{B}_l^a = \left(\mathbf{I} \otimes \begin{bmatrix} \Delta T^2/2 \\ \Delta T \end{bmatrix}\right) \begin{bmatrix} 0.5 & 0 & 0 & -0.5 \\ 0 & 0.5 & -0.5 & 0 \\ 0 & -0.5 & 0.5 & 0 \\ -0.5 & 0 & 0 & 0.5 \end{bmatrix}.$$

In essence, this allows odd (*cf.* even) numbered agents to implement control inputs in the Real-k (*cf.* i-j) plane only. Other weighting matrices of the cost function in (9) were considered to be $\mathbf{Q}^a = \mathbf{I}$, $\mathbf{T}^a = 10^2 \times \mathbf{I}$, and

$$\mathbf{R}_l^a = l^2 \left(10\mathbf{I} - 1.875(\mathbf{1}\mathbf{1}^\mathsf{T})\right)$$

where $\mathbf{1} = [1, 1, 1, 1]^\mathsf{T}$.

The goal was to bring the object to a stand-still at the center of the coordinate system from an initial position and speed using a model predictive control framework. To this, end the proposed decentralized quaternion-valued widely-linear quadrature regulator control procedure in Algorithm 2 and its centralized counterpart were implemented over the network in Figure 1. Control vector sequences were estimated for $1.6\,\mathrm{s}$ long segments. The first $0.8\,\mathrm{s}$ portion of the estimated control vector sequences were implemented; then, control vector sequences were re-estimated using the new state-vector information. This procedure was repeated to achieve the desired goal. The trajectory of φ_n is shown in Figure 3. Note that the developed decentralized control framework operated correctly. Furthermore, the object followed similar trajectories when using the centralized and developed decentralized control frameworks.

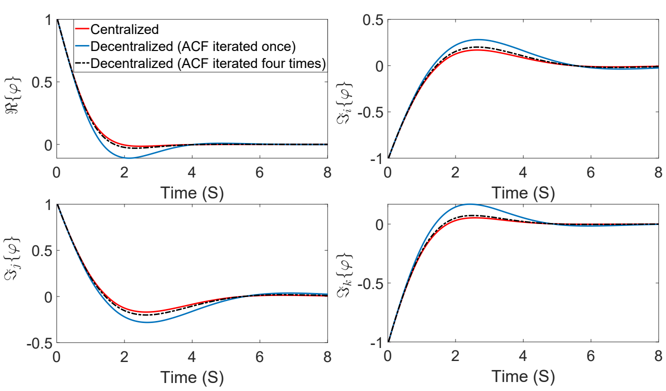

Fig. 3. Performance of the developed decentralized quaternion-valued widely-linear quadrature regulator with different ACF iterations. Performance of the centralized approach is provided as a benchmark.

C. Tracking Chaotic Rotations

In the final set of simulations, the signal generated by the Lorenz attractor

$$\begin{aligned} \chi_{n+1} &= \chi_n + 10^{-2}\left(\xi_n - \chi_n\right) \\ \xi_{n+1} &= \xi_n + 10^{-3}\left(-\chi_n\zeta_n + 28\chi_n - \xi_n\right) \\ \zeta_{n+1} &= \zeta_n + 10^{-3}\left(\chi_n\xi_n - \frac{8}{3}\zeta_n\right) \end{aligned}$$

was used to generate roll, α_n, pitch, β_n, and yaw ,γ_n, rotation signals so that $\cos(\chi_n) + i\sin(\chi_n) \to e^{i\alpha_n}$, $\cos(\xi_n) + j\sin(\xi_n) \to e^{j\beta_n}$, and $\cos(\zeta_n) + k\sin(\zeta_n) \to e^{k\gamma_n}$.

Each agent could observe these rotations as a change in its orientation. In this setting, if the orientation of agent l at time instant n is modeled as $q_{l,n}$, where $\forall l, n : \Re\{q_{l,n}\} = 0$ while $\Im_i\{q_{l,n}\}$, $\Im_j\{q_{l,n}\}$, and $\Im_k\{q_{l,n}\}$ are elements of the unit vector describing the orientation of agent l in a three-dimensional space. Thus, we have

$$q_{l,n+1} = \mu_n q_{l,n} \mu_n^{-1} \quad \text{with} \quad \mu_n = e^{i\alpha_n} e^{j\beta_n} e^{k\gamma_n}. \tag{65}$$

The model in (65) can be reformulated in an augmented fashion as

$$\mathbf{q}_{l,n+1}^{a\mathsf{T}} = \mathbf{q}_{l,n}^{a\mathsf{T}} \begin{bmatrix} \mu_n^* \Re\{\mu_n\} \\ \mu_n^{*i} \Im_i\{\mu_n\} \\ \mu_n^{*j} \Im_j\{\mu_n\} \\ \mu_n^{*k} \Im_k\{\mu_n\} \end{bmatrix}.$$

The aim is for the agents to cooperatively track the induced rotations. To this end, the derived DQKF was employed. The parameters in the state-space and observation model in (5)-(6) were selected as follows:

$$\begin{aligned} \mathbf{x}_n &= [\mu_n^* \Re\{\mu_n\}, \mu_n^{*i} \Im_i\{\mu_n\}, \mu_n^{*j} \Im_j\{\mu_n\}, \mu_n^{*k} \Im_k\{\mu_n\}]^\mathsf{T} \\ \mathbf{A}_n &= \mathbf{I}, \quad \mathbf{H}_{l,n} = \mathbf{q}_{l,n}^{a\mathsf{T}} \end{aligned}$$

while the observation of each node was its post rotation orientation, i.e., $\mathbf{y}_{l,n} = \mathbf{q}_{l,n+1}^{a\mathsf{T}}$. The state noise covariance was assumed to be diagonal with unit power. Although noise was not added to the observations, the observation covariance values were considered to be diagonal with 10^2 power. It is worthy to note that the ACF was iterated only once at each time instant.

Figure 4, shows the error between the predicted post rotation orientation and its actual value for each agent in the network. In addition, the performance of a conventional real-valued approach using rotation matrices and implementing the real-valued dual to the derived DQKF is provided as a benchmark, where in order to provide a balanced performance comparison, filtering parameters were set so that both techniques achieve a similar upper bound on their steady-state prediction errors. Note that the quaternion-valued approach achieved a lower median error, while converging faster than its real-valued dual. This simulation mainly demonstrates the generality of derived framework for use in a wide range of applications and the vantage point of quaternions when it comes to modeling three-dimensional signals.

VII. Conclusion

A unified framework for distributed filtering and control of quaternion-valued state-space processes has been developed. This framework has been realized through derivation of a distributed Kalman filtering technique for sequential tracking of quaternion-valued signals via networked multi-agent systems. The derived Kalman filtering algorithm mirrors the operations of an optimal centralized approach using embedded average consensus filters while allowing each agent to retain a Kalman style filtering operation and an intermediate estimate of the state vector. The distributed filtering framework has been expanded to solve the quaternion-valued widely-linear regulator problem, extending the duality between filtering and control to the quaternion domain and distributed setting. The

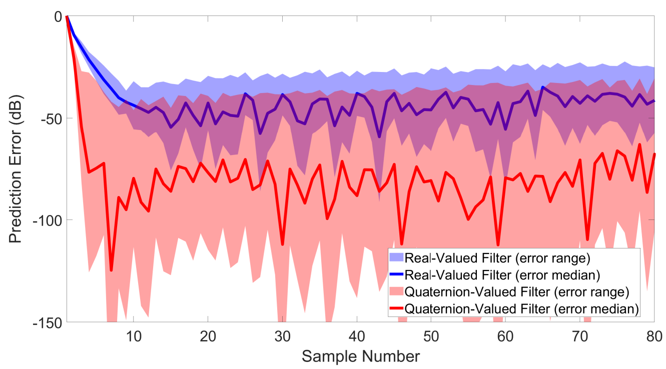

Fig. 4. Prediction error for tracking rotations generated by a Lorenz attractor. For the quaternion-valued (*cf.* real-valued) approach, prediction error of all agents lie within the region in red (*cf.* blue). The red (*cf.* blue) line indicate the median point between highest and lowest achieved error in dB scale.

concepts have been verified in a number simulations indicating that the developed framework can reach performance levels comparable to that of its centralized counterpart.

References

[1] J. S. Shamma, *Cooperative Control of Distributed Multi-Agent Systems*. John Wiley & Sons, 2007.
[2] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.
[3] S. Xu, K. Doğançay, and H. Hmam, "Distributed pseudolinear estimation and UAV path optimization for 3D AOA target tracking," *Signal Processing*, vol. 133, no. Supplement C, pp. 64–78, 2017.
[4] E. Franco, R. Olfati-Saber, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis using sensor networks and consensus-based filters," *In Proceedings of IEEE International Conference on Decision and Control*, pp. 386–391, December 2006.
[5] M. S. Mahmoud and H. M. Khalid, "Distributed Kalman filtering: A bibliographic review," *IET Control Theory Applications*, vol. 7, no. 4, pp. 483–501, March 2013.
[6] S. P. Talebi, S. Kanna, and D. P. Mandic, "A distributed quaternion Kalman filter with applications to smart grid and target tracking," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 477–488, December 2016.
[7] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Systems*, vol. 34, no. 4, pp. 87–97, August 2014.
[8] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, March 2014.
[9] R. Arablouei, S. Werner, Y. Huang, and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, January 2014.
[10] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, January 2013.
[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME Journal of Basic Engineering*, vol. 82, Series D, pp. 35–45, 1960.
[12] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," *In Proceedings of IEEE Conference on Decision and Control*, pp. 8179–8184, December 2005.
[13] Y. Zhang, C. Wang, N. Li, and J. Chambers, "Diffusion Kalman filter based on local estimate exchanges," *In Proceedings of IEEE International Conference on Digital Signal Processing*, pp. 828–832, July 2015.
[14] S. P. Talebi, S. Kanna, Y. Xia, and D. P. Mandic, "Cost-effective diffusion Kalman filtering with implicit measurement exchanges," *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4411–4415, 2017.
[15] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, September 2010.
[16] S. P. Talebi and S. Werner, "Distributed Kalman filtering in presence of unknown outer network actuations," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 186–191, January 2019.
[17] J. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Transactions on Automatic Control*, vol. 24, no. 2, pp. 266–269, April 1979.
[18] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, 1988.
[19] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *International Journal of Robotics Research*, vol. 12, no. 1, pp. 20–44, February 1993.
[20] W. Li, G. Wei, F. Han, and Y. Liu, "Weighted average consensus-based unscented Kalman filtering," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 558–567, February 2016.
[21] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," *In Proceedings of IEEE Conference on Decision and Control*, pp. 7036–7042, December 2009.
[22] M. Alighanbari and J. P. How, "An unbiased Kalman consensus algorithm," *In Proceedings of American Control Conference*, pp. 3519–3524, June 2006.
[23] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," *In Proceedings of IEEE Conference on Decision and Control*, pp. 5492–5498, December 2007.
[24] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.
[25] J. Hu, L. Xie, and C. Zhang, "Diffusion Kalman filtering based on covariance intersection," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 891–902, February 2012.
[26] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.
[27] J. T. Kong, J. W. Lee, S.-E. Kim, S. Shin, and W.-J. Song, "Diffusion LMS algorithms with multi combination for distributed estimation: Formulation and performance analysis," *Digital Signal Processing*, vol. 71, no. Supplement C, pp. 117–130, December 2017.
[28] S. P. Talebi and S. Werner, "Distributed Kalman filtering and control through embedded average consensus information fusion," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4396–4403, October 2019.
[29] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, NJ USA, 2007.
[30] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, January 2007.
[31] K. W. Spring, "Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: A review," *Mechanism and Machine Theory*, vol. 21, no. 5, pp. 365–373, 1986.
[32] K. Shoemake, "Animating rotation with quaternion curves," *SIGGRAPH Computer Graphics*, vol. 19, no. 3, pp. 245–254, July 1985.
[33] D. Xu, C. Jahanchahi, C. C. Took, and D. P. Mandic, "Enabling quaternion derivatives: The generalized HR calculus," *Royal Society Open Science*, vol. 2, no. 8, 2015.
[34] 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, February 2016.
[35] C. C. Took and D. P. Mandic, "Augmented second-order statistics of quaternion random signals," *Signal Processing*, vol. 91, no. 2, pp. 214–224, February 2011.
[36] J. Via, D. Ramirez, and I. Santamaria, "Properness and widely linear processing of quaternion random vectors," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3502–3515, July 2010.
[37] J. Via, D. P. Palomar, L. Vielva, and I. Santamaria, "Quaternion ICA from second-order statistics," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1586–1600, April 2011.
[38] S. P. Talebi and D. P. Mandic, "A quaternion frequency estimator for three-phase power systems," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3956–3960, April 2015.
[39] V. d. P. Brasil, A. de Leles Ferreira Filho, and J. Y. Ishihara, "Electrical three phase circuit analysis using quaternions," *In Proccedings of International Conference on Harmonics and Quality of Power*, pp. 1–6, May 2018.

[40] J. L. Contreras Hernandez, D. L. Almanza-Ojeda, S. Ledesma, A. Garcia-Perez, R. d. J. Romero-Troncoso, and M. Ibarra-Manzano, "Quaternion signal analysis algorithm for induction motor fault detection," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8843–8850, November 2019.

[41] C. Jahanchahi and D. P. Mandic, "A class of quaternion Kalman filters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 533–544, March 2014.

[42] C. E. Moxey, S. J. Sangwine, and T. A. Ell, "Hypercomplex correlation techniques for vector images," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1941–1953, July 2003.

[43] T. K. Paul and T. Ogunfunmi, "A kernel adaptive algorithm for quaternion-valued inputs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2422–2439, October 2015.

[44] D. Alfsmann, H. G. Gockler, S. J. Sangwine, and T. A. Ell, "Hypercomplex algebras in digital signal processing: Benefits and drawbacks," *In Proceedings of European Signal Processing Conference*, pp. 1322–1326, September 2007.

[45] D. Alfsmann, "On families of 2^N-dimensional hypercomplex algebras suitable for digital signal processing," *In Proceedings of European Signal Processing Conference*, pp. 1–4, September 2006.

[46] T. A. Ell and S. J. Sangwine, "Quaternion involutions and antiinvolutions," *Computes & Mathematics with Applications*, vol. 53, no. 1, pp. 137–143, January 2007.

[47] K. Abdel-Khalek, "Quaternion analysis," *Dipartimento di Fisica, Universita de Lecce, Lecce, Italy, Technical Report*, 1996. [Online]. Available: http://arxiv.org/abs/hep-th/9607152v2

[48] D. Xu and D. P. Mandic, "The theory of quaternion matrix derivatives," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1543–1556, March 2015.

[49] P. J. Schreier and L. L. Scharf, *Statistical Signal Processing of Complex-Valued Data: The Theory of Improper and Noncircular Signals*. Cambridge University Press, Cambridge, U.K., 2010.

[50] Y. Xia, C. Jahanchahi, T. Nitta, and D. P. Mandic, "Performance bounds of quaternion estimators," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3287–3292, December 2015.

[51] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *In Proceedings of Fourth International Symposium on Information Processing in Sensor Networks*, pp. 63–70, 2005.

[52] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[53] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.

[54] R. F. Stengel, *Optimal Estimation and Control*. Dover Publications, 1994.

Sayed Pouria Talebi received his PhD degree in statistical signal processing from Imperial College London, London, U.K., where his main focus of research was on the development of quaternion-valued distributed signal processing techniques. He is currently working as a post-doctoral researcher at Aalto University in Finland, where his main research interest is distributed optimization, estimation, and control methods for networked multi-agent systems.

Stefan Werner (SM'07) received the M.Sc. degree in electrical engineering from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 1998 and the D.Sc. degree with honors in electrical engineering from the Signal Processing Laboratory, Helsinki University of Technology (TKK), Espoo, Finland, in 2002. He is currently Professor at the Department of Electronic Systems at the Norwegian University of Science and Technology (NTNU). He is also an Adjunct Professor with Aalto University in Finland, and an Adjunct Senior Research Fellow with the Institute for Telecommunications Research, University of South Australia. He was holding an Academy Research Fellowship funded by the Academy of Finland from 2009 to 2014. His research interests include adaptive and statistical signal processing, wireless communications, and smart electric power grids. Dr. Werner is a member of the editorial board for the EURASIP journal of Signal Processing.

Danilo P. Mandic (M'99–SM'03–F'12) is a Professor of Signal Processing with Imperial College London, London, U.K., where he has been involved in nonlinear adaptive signal processing and nonlinear dynamics. He has been a Guest Professor with Katholieke Universiteit Leuven, Leuven, Belgium, the Tokyo University of Agriculture and Technology, Tokyo, Japan, and Westminster University, London, U.K., and a Frontier Researcher with RIKEN, Wako, Japan. He has two research monographs titled Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability (West Sussex, U.K.: Wiley, 2001) and Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models (West Sussex, U.K.: Wiley, 2009), an edited book titled Signal Processing Techniques for Knowledge Extraction and Information Fusion (New York, NY, USA: Springer, 2008), and more than 200 publications on signal and image processing. Prof. Mandic has been a member of the IEEE Technical Committee on Signal Processing Theory and Methods, and an Associate Editor of the IEEE SIGNAL PROCESSING MAGAZINE, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON NEURAL NETWORKS. He has produced award winning papers and products resulting from his collaboration with the industry.