Online Expectation-Maximization Based Frequency and Phase Consensus in Distributed Phased Arrays

Mohammed Rashid\textsuperscript{\oe}, Member, IEEE, and Jeffrey A. Nanzer\textsuperscript{\oe}, Senior Member, IEEE

Abstract—A distributed phased array is comprised of separate, smaller antenna systems (nodes) that coordinate with each other to support coherent beamforming. However, due to the frequency drift and phase jitter of the oscillators on the nodes, as well as the frequency and phase estimation errors induced in the synchronization process, there exists a decoherence between the nodes that degrades their beamforming gain. A decentralized frequency and phase consensus (DFPC) algorithm was proposed in a prior work for undirected array networks in which the nodes locally share their frequencies and phases with their neighboring nodes to reach a synchronized state. Kalman filtering (KF) was also integrated with DFPC, and the resulting KF-DFPC showed significant reduction in the total residual phase error upon convergence. One limitation of the DFPC-based algorithms is that, due to relying on the average consensus protocol which assumes undirected networks, they do not generally converge for directed array networks with column stochastic weighting matrix. For directed networks, a push-sum protocol based frequency alignment algorithm was recently proposed, but it does not take into account both frequency and phase variations in-between the update intervals. In this paper, we model the variations using a first-order Markov process and propose a push-sum based frequency and phase consensus (P-FPC) algorithm which is accordingly a modified version of the push-sum protocol. The residual phase error of P-FPC upon convergence is theoretically derived as well. Kalman filtering is also integrated with P-FPC and the resulting KF-P-FPC algorithm shows significant reduction in the residual error of the array. A limitation of KF-P-FPC is that the model parameters, i.e., the measurement noise and the process noise covariance matrices, are assumed to be known a priori. Since they may not be known in practice, we develop an online expectation maximization (EM) based algorithm that iteratively computes the maximum likelihood (ML) estimates of these matrices in an online manner. EM is integrated with KF-P-FPC and the resulting algorithm is referred to as the EM-KF-P-FPC algorithm. Simulation results are included where the performances of the P-FPC-based algorithms are analyzed for different distributed array networks, and are compared to the DFPC-based algorithms as well as to the earlier proposed hybrid consensus on measurement and consensus on information (HCMCI) algorithm.

Index Terms—Distributed beamforming, distributed phased arrays, frequency and phase consensus, Kalman filtering, online expectation maximization, push-sum algorithm.

I. INTRODUCTION

Modern advancements in wireless technologies have enabled more reliable communications between separate wireless systems, and as a result, several new applications involving distributed systems have emerged over the recent years. These include distributed beamforming systems [1], distributed automotive radars [2], [3], and distributed massive MIMO systems [4], among others. The underlying antenna array technology in all these applications is a phased array (a.k.a. distributed antenna arrays, virtual antenna arrays, or coherent distributed arrays). A distributed phased array is essentially a network of multiple spatially distributed and mutually independent antenna systems that are wirelessly coordinated at the wavelength level to coherently transmit and receive radio frequency signals [5]. When compared to the classical analog or digital phased arrays where all the antennas are mounted together on a single-platform, coherent distributed phased array architectures acquire significantly better spatial diversity, higher directivity, improved signal-to-noise ratio (SNR) at the destination, greater resistance to system failures and interference, capability to easily reconfigure for the dynamic environments, and ease in scalability.

In a single-platform phased array, usually all antennas and their transceiver chains are driven by a universal local oscillator to assure the synchronized state of the array, whereas in a distributed phased array, each antenna system has its own local oscillator in the transceiver chain. Thus, in a free-running state in which the oscillator is not locked to any reference signal, it undergoes a random frequency drift and phase jitter over time which results in a decoherence between the signals emitted by the different nodes. While a reference signals from the systems, such as global positioning system (GPS) can be used to synchronize these oscillators provided that the nodes are equipped with the GPS receivers [6], it is not always the case that the GPS signal is available or reliable. Alternatively, these nodes can also be synchronized by wirelessly transmitting the reference signals from the destination system, for instance, using a single or multiple bit feedback methods [7], [8], or retrodirective synchronization [9]; however, since these approaches rely on the feedback from the target destination, they are essentially closed-loop, and cannot steer beams to arbitrary directions. These closed-loop
methods are thus only suitable for the wireless communication applications where such feedback from the target destination is feasible. Recently, open-loop synchronization methods have also been proposed [1], [10], [11], [12], in which the nodes communicate with each other to synchronize their oscillators without using any feedback from the destination. Since the destination need not be an active system, open-loop distributed phased arrays can also be used for radar and remote sensing applications.

In an open-loop distributed phased array, the electrical states (i.e., the frequencies and phases) of the nodes can be synchronized by either using a centralized array or a decentralized one. In [1], [12], [13], [14], a centralized open-loop distributed phased array was used in which a node is classified as either a primary node or a secondary node. The primary node transmits a reference signal to the secondary ones that is used to synchronize the electrical states across the array. However, this architecture is susceptible to the primary node’s failure, and due to limited communication resources, it is also not scalable. On the other hand, in a decentralized (i.e., distributed) open-loop distributed phased array, there is no such classification of the nodes as either the primary or secondary nodes. Therein, the nodes communicate with their neighboring nodes to iteratively share their measurements and update their electrical state, until synchronization is achieved across the array. Based on this approach, a decentralized frequency and phase consensus (DFPC) algorithm was proposed in [11] wherein the nodes update these parameters iteratively by computing a weighted average of the shared values using the average consensus algorithm [15]. Kalman filtering was also integrated with the DFPC algorithm that further reduced the residual phase error upon convergence. Both synchronization as well as convergence performances of the KF-DFPC algorithm were compared to the DFPC algorithm, the diffusion least mean square (DLMS) algorithm [16], the Kalman consensus information filter (KCIF) algorithm [17], and the diffusion Kalman filtering (DKF) algorithm [18], [19], and it was shown that KF-DFPC outperforms DFPC and DLMS in terms of reducing the residual phase error, and that for the same residual phase error, KF-DFPC converges faster than the DFPC, KCIF, and DKF algorithms. Thus, KF-DFPC is more favorable in the case where the latency for synchronization is undesirable or when low-powered nodes are used in the distributed phased array.

However, there are two major limitations of the above algorithms that are as follows. First, these algorithms assume bidirectional (undirected) communication links between the nodes, whereas in practice due to the large spatial separation between the nodes, the dynamic changes in their environments, and their limited communication ranges, there may exist unidirectional (directed) communication links between them. Since these algorithms use the traditional average consensus algorithm [15] for updating the frequency and phase at each node, they do not generally converge for directed networks. This is because the weighting matrix of an undirected network is a doubly-stochastic matrix (i.e., the elements in both its rows and columns sum to one), whereas for the directed networks, it is usually a column stochastic matrix. Secondly, the above algorithms, in particular, the KF-DFPC algorithm of [11], assume that the statistical knowledge of the frequency drifts and phase jitters of the oscillators as well as that of the frequency and phase estimation errors are known a priori. Specifically, it is assumed that the process noise and the measurement noise covariance matrices in the state transition model and the observation model for the Kalman filtering algorithm are known to the nodes; however in practice, this is not always the case since the operational dynamics of the oscillators are influenced by several factors, such as the changes in their operating temperatures and the power supply voltages [20], aging of the oscillators [21], [22], and noise induced by their electronic components [23]. Furthermore, the statistics of the frequency and phase estimation errors are estimated from the observed estimates at the nodes, where their accuracies are influenced by the SNR of the observed signals and the numbers of samples collected per observation period. Thus, the noise covariance matrices are generally unknown to the nodes and must be estimated for Kalman filtering.

A push-sum method [24] based decentralized frequency synchronization algorithm was proposed in [25] for directed networks of distributed phased array. The authors did not consider the frequency and phase variations between the iterations and assumed a specific class of the nodes in which the frequencies are only incremented in discrete frequency steps, that limits the frequency resolution of the nodes and increases the residual phase error of the algorithm. In comparison to [25], in this paper, we consider a more advanced class of nodes, for e.g., USRP-based nodes as used in [1], for which such frequency quantization errors are negligible. In addition, we consider both frequency and phase synchronization of the nodes for directed networks of distributed phased arrays, both of which are necessary for distributed beamforming. While [25] directly applies the push-sum algorithm [24] for the frequency alignment, in this paper, we model frequency and phase variation at the nodes using a first-order Markov process, and then taking into account the states transitioning in between the iterations (update intervals) we propose a modified push-sum consensus algorithm for the joint frequency and phase alignment of the array. This paper represents a significant expansion of the concepts briefly described in [26]. The major contributions of this paper are as follows.

- We consider the same signal model for the nodes as assumed in [11] in which the oscillators induced offset errors, such as the frequency drifts, the phase drifts, and the phase jitters are included and modeled using practical statistics. Due to these added offsets, the exact frequencies and phases are unknown to the nodes and must be estimated from the observed signals. Thus, we also include the frequency and phase estimation errors in our signal model. The array network in [11] is assumed to be undirected (bidirectional) which is less likely to exist in practice. Therefore, in this paper, we consider directed communication links between the nodes, i.e., if a node transmits information to another node, it may not hear back from it. For such networks, as discussed before,
the algorithms in [11] are not applicable whereas [25] only considered the frequency alignment, so we propose a push-sum algorithm based decentralized joint frequency and phase consensus algorithm which is referred to herein as the P,FPC algorithm. As stated above, our proposed P,FPC algorithm is not a direct application of the push-sum algorithm of [24], [25], in fact it is a modified version where the estimated frequencies and phases of the nodes are included and must be multiplied by the weighting factor from the previous iteration in order to compute the temporary variables (see Eqn. (3) below). This step is required to recover the temporary variables from the previous iteration taking into account the fact that the updated frequencies and phases from each iteration of P,FPC undergo a state transition due to the random drifts and jitter in between the update intervals. Compared to [26], Section II-C of this present paper includes new results with different scenarios.

- The residual phase error of the P,FPC algorithm is also theoretically derived to analyze the contributing factors to the residual decoherence between the nodes upon its convergence. It is observed that the phase error depends on the network algebraic connectivity (i.e., the modulus of the second largest eigenvalue $\lambda_2$ of the weighting matrix) which decreases with the increase in the number of nodes in the array or with the increase in the connectivity between the nodes. Simulation results are also included where the residual phase errors of P,FPC are analyzed in the context of these theoretical results for different array networks.

- In distributed phased arrays, the electrical states of the nodes are updated with smaller update intervals, usually on the order of millisecond (ms), to avoid large oscillator drifts [27] and to correct for the residual phase errors due to the platform vibrations [28]. However, it was shown in [11] that at such update intervals, the residual phase error of a consensus algorithm is usually higher when no filtering is used at the nodes, which deters an accurate coherent operation. Kalman filtering (KF) is a popular algorithm that provides the optimal minimum mean squared error (MMSE) estimates of the unknown quantities when their states transitioning model follows the first-order Markov process, and the observations are a linear function of the states. Thus, we integrate KF with P,FPC to improve the residual phase errors for the directed networks of distributed phased array. The resulting algorithm is referred to as the KF-P,FPC algorithm.

- In contrast to [11], we assume in this work that the measurement noise and process noise covariance matrices are unknown to KF, and thus we derive an online expectation-maximization (EM) algorithm [29] to compute their maximum likelihood (ML) estimates from the instantaneously observed measurements. The online EM algorithm is integrated with KF and P,FPC, and the resulting distributed consensus based KF algorithm is referred to as the EM-KF-P,FPC algorithm. In [30], an EM algorithm was derived for the Kalman filtering, but it assumes a batch mode technique in which a batch of measurements is recorded a priori, and then the EM algorithm is run iteratively (until convergence) on the entire batch to estimate the unknown parameters. This batch-mode EM is not applicable in our considered synchronization problem because we want to update the nodes in an online manner to avoid larger oscillators drifts and to simultaneously perform the coherent operation during the synchronized interval. Recently, along these lines, a variational Bayes (VB) method based hybrid consensus on measurement and consensus on information (VB-HCMCI) algorithm was proposed in [31], wherein the unknown noise matrices are estimated in an online manner using VB; however, it has the following drawbacks. Firstly, in contrast to our proposed EM-KF-P,FPC algorithm, VB-HCMCI additionally requires instantaneously fusing the neighboring nodes’ measurements and the KF predicted information at each node that demands more communication resources. Secondly, the nodes using VB-HCMCI also need to perform multiple iterations at each time instant for the convergence of the VB algorithm. We note that, in our considered synchronization problem, performing multiple iterations at each time instant increases the update interval of the nodes in a practical distributed array, which is not desirable to avoid the strong decoherence between the nodes caused due to the larger oscillator drifts. Finally, VB-HCMCI is developed for the undirected networks using the average consensus algorithm of [15], and as discussed before, it does not generally converge for directed array networks. In contrast, our EM-KF-P,FPC algorithm requires only a single iteration per time instant, which makes it computationally less expensive, and yet computes the ML estimates of the covariance matrices in an online manner. In addition, it requires less communication resources per node (see Section III-D), and is feasible for both directed as well as undirected networks.

- The computational complexity analysis of EM-KF-P,FPC is also included at the end of Section III where it is shown that our proposed online EM algorithm has the same computational complexity as that of KF, and notably for the moderately connected large arrays, using the EM and KF algorithms does not increase the overall computational complexity of the P,FPC algorithm. Simulations results are also included where the residual phase errors and the convergence speeds of the proposed P,FPC and EM-KF-P,FPC algorithms are analyzed by varying the number of nodes in the array, the SNR of the observed signals, and the connectivity between the nodes. Furthermore, our online EM algorithm is also integrated with the KF-DFPC algorithm of [11] and the performance of the resulting EM-KF-DFPC is also compared to the EM-KF-P,FPC algorithm and the HCMCI algorithm of [32].

Rest of this article is organized as follows. Section II describes the signal model of the nodes, the proposed P,FPC algorithm, and theoretically analyzes the residual phase error of P,FPC upon its convergence. Simulation results are also included therein to analyze the synchronization performance...
of P,FPC. The KF algorithm and the online EM algorithm are derived in Section III where these algorithms are integrated with the P,FPC algorithm to improve the synchronization between the nodes. The performance of EM-KF-P,FPC is examined through simulations and is also compared to EM-KF-DFPC for the undirected networks. To this end, the initialization of both the EM and KF algorithms are also described in Section III and the computational complexity analysis of the overall algorithm is also performed.

Notations: A lower case small letter \( x \) is used to represent a signal or scalar, and a lower case bold small letter \( \mathbf{x} \) is used to denote a vector. An upper case bold capital letter \( \mathbf{X} \) represents a matrix. The transpose and inverse operations on a matrix are indicated by the superscripts \(^T\) and \(^{-1}\), respectively. The notations \( |X| \) and \( \text{tr}[X] \) are used for the determinant and trace of a matrix \( X \), respectively. A normal probability distribution on a random vector \( x \) is denoted by \( \mathcal{N}(\mu, \Sigma) \) in which \( \mu \) indicates the mean and \( \Sigma \) represents the covariance matrix of the distribution. The expectation of \( x \) with respect to the probability distribution on \( x \) is represented by \( \mathbb{E}[x] \). For ease of notation, a set \{x\(_n\)(1), x\(_n\)(2), …, x\(_n\)(k)\} is written in compact form as \( \mathbf{x}_n^{k} \). 1 is a column vector of all 1s. Finally, a diagonal matrix created with the elements in vector \( x \) is denoted by \( \mathbf{X} = \text{diag}(\mathbf{x}) \), and an \( N \times N \) identity matrix is represented by \( \mathbf{I}_N \).

II. FREQUENCY AND PHASE SYNCHRONIZATION IN DISTRIBUTED PHASED ARRAY WITH DIRECTED COMMUNICATION LINKS

Consider a distributed phased array made up of \( N \) spatially distributed antenna nodes that are coordinating with each other to perform a coherent operation toward the destination. We assume that due to the large spatial separation between the nodes and their limited communication ranges, the communication links between the nodes are unidirectional across the entire array. Thus, we model this array network by a directed graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) in which \( \mathcal{V} = \{1, 2, \ldots, N\} \) represents the set of vertices (antenna nodes), and \( \mathcal{E} = \{(n, m): n, m \in \mathcal{V}\} \) is the set of all directed edges (unidirectional communication links) in the array. Let the signal generated by the \( n \)-th node in the \( k \)-th iteration be given by \( s_n(t) = e^{j(2\pi f_n(k)t + \theta_n(k))} \) for \( t \in [(k-1)T, kT], \) where \( T \) is the signal duration, and \( k \in \{1, 2, \ldots\} \). The parameters \( f_n(k) \) and \( \theta_n(k) \) represent the frequency and phase of the \( n \)-th node in the \( k \)-th iteration, respectively. In a decentralized consensus averaging algorithm, the nodes iteratively exchange their frequencies and phases with their neighboring nodes and update these parameters by computing a weighted average of the shared values until convergence is achieved (i.e., these parameters are synchronized across the array). Due to the frequency drift and phase jitter of the oscillator, the frequency and phase of the \( n \)-th node in the \( k \)-th iteration can be written as

\[
\begin{align*}
    f_n(k) &= f_n(k-1) + \delta f_n \\
    \theta_n(k) &= \theta_n(k-1) + \delta \theta_n + \delta \theta_n,
\end{align*}
\]

in which \( \delta f_n \) is the frequency drift of the oscillator at the \( n \)-th node which we assume is normally distributed as \( \mathcal{N}(0, \sigma_f^2) \) with \( \sigma_f \) representing the Allan deviation (ADEV) of the oscillator. The ADEV is modeled as \( \sigma_f = f_c \sqrt{\frac{2}{T}} + \beta f \) in which the oscillator’s design parameters \( \beta_1 \) and \( \beta_2 \) are set as \( \beta_1 = \beta_2 = 5 \times 10^{-19} \) which represent a quartz crystal oscillator [11], [33]. The phase error \( \delta \theta_n \) in (1) represents the phase offset due to the temporal variation of the frequency drift \( \delta f_n \) between the update intervals which is \( \delta \theta_n = -\pi T \delta f_n \) as shown in [11]. Finally, \( \delta \theta_n \) in (1) represents the phase offset due to the phase jitter of the oscillator at the \( n \)-th node. It is modeled as \( \delta \theta_n \sim \mathcal{N}(0, \sigma_\theta) \) in which \( \sigma_\theta = \sqrt{2 \times 10^3/10} \) and \( A \) is the integrated phase noise power of the oscillator which is obtained from its phase noise profile. Herein, we set \( A = -53.46 \) dB that models a typical high phase noise voltage controlled oscillator [11], [27]. We assume that the frequency transitioning process in (1) begins with the initial value \( f_n(0) = \mathcal{N}(f_c, \sigma^2) \) in which \( f_c \) is the carrier frequency and \( \sigma = 10^{-4} f_c \) denotes a crystal clock accuracy of 100 parts per million (ppm), whereas the phase transitioning process starts with \( \theta_n(0) \sim \mathcal{U}(0, 2\pi) \) which represents the hardware induced initial phase offset of the oscillator.

Due to the above dynamics of the oscillators, the nodes need to estimate their frequencies and phases in each iteration before updating them, and thus the estimated frequency and phase of the \( n \)-th node in the \( k \)-th iteration are given by

\[
\begin{align*}
    \hat{f}_n(k) &= f_n(k) + \varepsilon_f \\
    \hat{\theta}_n(k) &= \theta_n(k) + \varepsilon_\theta,
\end{align*}
\]

in which the frequency estimation error \( \varepsilon_f \) and the phase estimation error \( \varepsilon_\theta \) are both normally distributed with zero mean and standard deviation \( \sigma_f^m \) and \( \sigma_\theta^m \), respectively.

A. PUSH-SUM FREQUENCY AND PHASE CONSENSUS ALGORITHM

We consider a directed array network of \( N \) nodes described by a directed graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) as before, whereas the \((n, m)\)-th edge in \( \mathcal{E} \) is assigned herein a weight \( w_{m,n} \). We assume that \( \mathcal{G} \) represents a strongly connected directed graph, i.e., there exists at least one directed path from any node \( n \) to any node \( m \) in the graph with \( n \neq m \). Furthermore, it is assumed that the \( n \)-th node in the network knows its in-neighbor set \( \chi_n^{\text{in}} \triangleq \{v \in \mathcal{V} : w_{v,n} > 0\} \) and its out-neighbor set \( \chi_n^{\text{out}} \triangleq \{v \in \mathcal{V} : w_{n,v} > 0\} \). The proposed algorithm is based on the push-sum consensus algorithm [24], [25], and is referred to herein as the push-sum frequency and phase consensus (PSFPC) algorithm. In the PSFPC algorithm, the \( n \)-th node maintains three temporary variables in the \( k \)-th iteration, i.e., \((k, \hat{x}^f_n(k), \hat{x}_n^\theta(k), s_n(k)\), which are updated as follows.

\[
\hat{x}^f_n(k) = \sum_{m \in \chi_n^{\text{in}}} w_{n,m} \hat{f}_m(k) s_m(k-1)
\]
Algorithm 1 PₜFPC Algorithm

Input: \( k = 0 \), define \( s_n(0) = 1 \) for all \( n = 1, 2, \ldots, N \).

while convergence criterion is not met do

\( k = k + 1 \)

For each node \( n \):

a) Use the observed \( \hat{f}_n(k) \) and \( \hat{\theta}_m(k) \) for all \( m \in \chi_n \) and update \( x_n^\theta(k) \), \( x_n^\phi(k) \), and \( s_n(k) \) using (3).

b) Update \( f_n(k) \) and \( \theta_n(k) \) using (4).

end

Output: \( f_n(k) \) and \( \theta_n(k) \) for all \( n = 1, 2, \ldots, N \)

\[
x_n^\theta(k) = \sum_{m \in \chi_n} w_{n,m} \hat{\theta}_m(k) s_m(k-1)
\]

\[
s_n(k) = \sum_{m \in \chi_n} w_{n,m} s_m(k-1),
\]

(3)

where \( s_n(0) = 1 \) and the weight \( w_{n,m} = \frac{1}{\lambda_n \pi_n \sigma_m^2} \) if \( m \in \chi_n \) and is 0 otherwise. The parameter \( \theta_{out}^m \) is the out-degree of node \( m \) which can be found by computing the cardinality of the set \( \chi_n \). These weights make the \( N \times N \) weighting matrix \( W = [w_{n,m}] \) a column stochastic matrix that supports an average consensus [35]. The parameters \( f_n(k) \) and \( \theta_n(k) \) represent the estimated frequency and phase of the \( m \)-th node in the \( k \)-th iteration, and they are multiplied by the weighting factor \( s_m(k-1) \) from the previous iteration. This multiplication is required to recover the temporary variables \( x_n^\phi(k-1) \) and \( x_n^\theta(k-1) \) taking into account the fact that at the beginning of each iteration, the nodes observe only the estimated frequencies and phases after their updated values in the previous iteration are influenced by the oscillator drifts and the phase jitters. The updated frequency and phase of the \( m \)-th node in the \( k \)-th iteration are computed by

\[
f_n(k) = \frac{x_n^\phi(k)}{s_n(k)}
\]

\[
\theta_n(k) = \frac{x_n^\theta(k)}{s_n(k)}
\]

(4)

The above steps are repeated iteratively at each node until the algorithm converges. The pseudo-code of the PₜFPC algorithm is provided in Algorithm 1.

B. Residual Phase Error Analysis

Herein, we theoretically analyze the residual phase errors of the nodes upon the convergence of the PₜFPC algorithm. To this end, let at the \( I \)-th iteration, the estimated frequencies and phases of the nodes be written in the short-hand vector form as \( \hat{z}(I) = z(I-1) + e_I \) in which \( z \in \{f, \theta\} \) and \( e_I \sim \mathcal{N}(0, \sigma^2 I_N) \). Note that by using (1) and (2), it can be seen that when \( z = f \) then the error variance is \( \sigma_e^2 = \sigma_f^2 + \langle \sigma_f^m \rangle^2 \), and when \( z = \theta \) then the variance is \( \sigma_\theta^2 = \pi^2 T^2 \sigma_f^2 + \langle \sigma_f^m \rangle^2 + \sigma_\theta^2 \). Thus, at the \( I \)-th iteration, the temporary variables in (3) can be alternatively written as

\[
x_n^\phi(I) = [W (\hat{z}(I) \circ s(I-1))]_n
\]

\[
x_n^\theta(I) = \frac{\sigma^2_e}{\sigma_n^2} \sum_{i=1}^I \lambda^2_i |s_n(I-i)|^2
\]

where \( \circ \) is the Hadamard product between the two vectors, and the operation \( [\cdot]_n \) selects the \( n \)-th element of the resulting vector. Inserting \( \hat{z}(I) \) in (5) and using backward recursion, it can be reduced as follows

\[
x_n^\phi(I) = \frac{\sigma^2_e}{\sigma_n^2} \sum_{i=1}^I \lambda^2_i |s_n(I-i)|^2
\]

Similarly, we have \( s_n(I) = [W^T s(0)]_n \). For a column stochastic matrix \( W \), it is shown in [36] that as \( I \rightarrow \infty, W^T = \pi^T I \) where \( \pi > 0 \) is the stationary distribution of the random process with transition matrix \( W \) that satisfies \( W \pi = \pi \) with \( I^T \pi = 1 \). Thus from (4), the updated frequency or phase of node \( n \) at the \( I \)-th iteration can be written in the shorthand form as

\[
z_n(I) = \frac{1}{\pi_n N} \left[ W^T (z(0) \circ s(0)) + \sum_{i=1}^I W^T (e_I(i-1) \circ s(I-i)) \right]
\]

where to get (7) we set \( s(0) = 1 \) (as in Section II-A) that gives for a large \( I \), \( s_n(I) = W^T s(0) / \pi_n N \) in which \( \pi_n \) is the \( n \)-th element of \( \pi \). Similarly, for larger \( I \), the first summand in (7) reduces to \( \frac{1}{N} \sum_{n=1}^N z_n(0) \), i.e., it gives the average of the initial frequencies and phases of the nodes, whereas the second summand in (7) represents the residual total phase error due to the frequency and phase offset errors induced at the nodes in every iteration. To quantify this residual phase error for node \( n \) after \( I \) iterations, we find the variance of the following term

\[
\frac{1}{\pi_n N} \left[ \sum_{i=1}^I (W^T - \pi^T I^T) (e_I(i-1) \circ s(I-i)) \right]
\]

\[
= \frac{1}{\pi_n N} \left[ \sum_{i=1}^I (W^T - \pi^T I^T)^i (e_I(i-1) \circ s(I-i)) \right]
\]

(8)

where to get the second equality in the above equation, we used the following two facts. Firstly, a column stochastic \( W \) implies that \( I^T W^T = I^T \), and secondly, we have \( (I_N - \pi^T I^T)^i = (I_N - \pi^T I^T)^{i-1} \) as it is a projection matrix. Now let \( \lambda_2 \) be the modulus of the second largest eigenvalue of the mixing matrix \( W \). Then it can be conveniently shown in a few steps that \( \lambda_2^2 I_N - (W - \pi^T I^T)^i \) is a positive semi-definite matrix. Thus, using this identity along with assuming that the frequency and phase offset errors induced at the nodes in every iteration are mutually independent across the array, the variance of the residual phase error for node \( n \) can be solved as

\[
\sigma^2_{n,\text{residual}} = \sigma_e^2 \frac{1}{\pi_n N^2} \sum_{i=1}^I \lambda^2_i |s_n(I-i)|^2
\]

(9)

Since for large \( I \), the scalars \( s_n(I) \to \pi_n N \), it implies that we can write the phase error in (9) as \( \sigma^2_{n,\text{residual}} = \sigma_e^2 \sum_{i=1}^I \lambda^2_i + C \)
Fig. 1. Frequency errors in ppm for all the $N$ nodes in the array vs. P,FPC iterations for $c = 0.2$ and when (a) $N = 20$ and SNR = 0 dB, and (c) $N = 100$ and SNR = 30 dB.

Fig. 2. Phase errors in degrees for all the $N$ nodes in the array vs. P,FPC iterations for $c = 0.2$ and when (a) $N = 20$ and SNR = 0 dB, and (c) $N = 100$ and SNR = 30 dB.

where the constant $C$ is close to zero for the larger arrays. Consequently, for sparsely connected arrays, $\lambda_2$ is close to 1 and $\sum_{i=1}^{N} \lambda_2^i \gg 1$ which results in a higher residual phase error, but as the connectivity increases, $\lambda_2 \to 0$ and the residual phase error decreases. This variation in the total phase error as a function of the change in the array connectivity is illustrated through simulations in Section III-E.

C. Simulation Results

For validation purposes, we examine the frequency and phase synchronization performances of the proposed P,FPC algorithm through simulations. To this end, we consider an array network of $N$ nodes that is randomly generated with connectivity $c$ between the nodes. The parameter $c$ is defined as the total number of existing connection in the generated network divided by the total number of all possible connections $\frac{N}{2} (N - 1)$. Thus, $c \in [0, 1]$ where a smaller value of $c$ implies a sparsely connected array network with a few connections per node, and a larger value of $c$ defines a highly connected array network with many connections per node. The average number of connections per node in a network is given by $D = c(N - 1)$. The carrier frequency of the nodes is chosen in this analysis to be $f_c = 1$ GHz, and the sampling frequency is set to $f_s = 10$ MHz throughout this paper. As the update interval for distributed phased arrays is usually on the order of ms, we set $T = 0.1$ ms for all the simulated cases in this paper.

Figs. 1 and 2 show the frequencies and phases for all the $N$ nodes in the array relative to their average values vs. the number of iteration of the P,FPC algorithm from a single trial, when a moderately connected array network with connectivity $c = 0.2$ is generated for the different number of nodes $N$ in the array. The SNR of the observed signals is either set as 0 dB or 30 dB. These figures show that with the increase in the number of iterations, both frequencies and phases of the nodes converge to the average of their initial values for all the considered cases. The residual error upon convergence results from the oscillators’ frequency drifts and phase jitters, as well as the frequency and phase estimation errors induced at the nodes. Thus, when the SNR increases to 30 dB for $N = 100$ nodes, it is observed that the residual error also decreases significantly upon the convergence due to the decrease in the estimation errors.

Fig. 3. Standard deviation of the total phase errors of the P,FPC and DFPC algorithms vs. the connectivity $c$ in the array for different $N$ values when SNR = 30 dB is assumed.

Now let the total residual phase error of node $n$ be denoted by $\delta \phi_n = 2\pi \delta f_n T + 2\pi \epsilon f T + \delta \theta_n + \epsilon \theta$. Thus, in Fig. 3, we plot the standard deviation of the total phase errors $\delta \phi_n$
for all the $N$ nodes in the array for the P,FPC algorithm by varying the connectivity $c$ between the nodes and the number of nodes $N$ in the array, and when SNR $= 30$ dB is assumed unless otherwise stated. For the comparison, we generate the bidirectional array networks for each $c$ and $N$ values and show the standard deviation of the total phase error of the DFPC algorithm proposed in [11] for such array networks. Note that the standard deviation of the DFPC-based algorithms in this work provides the lower bounds on the standard deviation of the total phase error of the P,FPC-based algorithms, and the slight increase in the phase error of the P,FPC-based algorithms as compared to the DFPC-based algorithms is due to the decrease in the number of connection per node in the directed networks which effects the accuracy of the local averages computed per node throughout the network as expected. For this figure, both P,FPC and DFPC algorithms were run for 100 iterations and the final standard deviations of the total phase errors were averaged over $10^5$ trials. The center of each point in the error bar plot is the average value of the samples and the length of the bar defines the standard deviation around the average value. It is observed that for each $N$ value, as the connectivity $c$ between the nodes increases then the total phase error of the P,FPC and DFPC algorithms decreases proportionately. Furthermore, an increase in the number of nodes $N$ in the array, for a given $c$ value, also decreases the total phase errors of both algorithms. As discussed in the previous subsection, the decrease in the residual phase error of P,FPC with the increase in the $c$ or $N$ values is because the network algebraic connectivity $\lambda_2$ decreases which in turn reduces the residual phase of P,FPC. The residual phase error of DFPC is also a function of $\lambda_2$ as shown in [11] and thus the same trend is observed for the DFPC algorithm as well.

In Fig. 3, we also illustrate the performances of the P,FPC and DFPC algorithms when $N = 100$ and SNR $= 25$ dB are assumed. In this case, an increase in the residual phase error is observed for both algorithms at all the $c$ values when compared to the phase errors of these algorithms in the other case where SNR $= 30$ dB. This degradation in performances is due to the increase in the frequency and phase estimation errors at the nodes which contribute to the total residual phase error of the algorithms; however, the phase errors decreases with the increase in $c$ due to the reasons explained earlier. A more thorough analysis of the change in the phase error of both algorithms vs. SNR is included later in Section III-E.

Finally, in Fig. 4, we compare the convergence speeds of the P,FPC and DFPC algorithms by plotting the standard deviation of the total phase error vs. the number of iteration of the algorithms when the two different $N$ and $c$ values are assumed and the SNR is set to 30 dB. It is observed that either when $N$ increases from 20 to 100 for a given $c$ value, or when $c$ increases from 0.2 to 0.5 for a given $N$ value, the P,FPC algorithm converges faster. Specifically, for $N = 20$ and $c = 0.2$, P,FPC converges in 32 iterations, but for this $N$ value, when $c$ increases to 0.5, P,FPC converges in about 11 iterations. Moreover, for $c = 0.5$ and $N = 100$ nodes, P,FPC converges in about 5 iterations. This is because as either $c$ or $N$ increases then the average number of connections per node in the network, i.e., the $D$ value, increases which in turn quickly stabilizes the local averages computed at the nodes. Although the same trend is also observed for the DFPC algorithm in all the considered cases in this figure, particularly it is observed that the P,FPC algorithm converges faster than the DFPC algorithm in all cases. This is because the network is bidirectional for the DFPC algorithm, and as such there are more connections per node in the network which in turn delays the convergence of the local averages computed at the nodes. Note that the P,FPC algorithm can be deployed for a bidirectional network as well, in which case it results in the same synchronization and convergence performance as that of the DFPC algorithm; however, the DFPC algorithm does not converge in case of directed networks due to its dependence on the average consensus algorithm [15] as discussed before. In practice, since the failure of a link may not be easily detectable, this means that the P,FPC algorithm is a preferred choice over the DFPC algorithm.

### III. Mitigating Residual Phase Errors With Online Expectation-Maximization and Kalman Filtering

In order to reduce the residual phase error of the P,FPC algorithm, we propose to use Kalman filtering (KF) at the nodes that computes the MMSE estimates of the frequencies and phases in each iteration to mitigate the offset errors. KF is applicable in scenarios where the state transitioning model follows a first-order Markov process, and the observations are a linear function of the unobserved state [37]. KF has been used for the time synchronization between the nodes in [38], [39], and for the frequency and phase synchronizations between the nodes in a distributed antenna arrays in [11]. However, an implicit assumption in using KF with the synchronization algorithms in [11], [38], [39] is that the process noise and the measurement noise covariance matrices are known to the Kalman filter. As discussed in Section I, these matrices are usually unknown and must be estimated for the KF algorithm. Thus, in this section, we propose to use an online expectation-maximization (EM) algorithm [29] that iteratively computes the maximum likelihood estimates of these noise matrices for

---

**Figure 4.** Standard deviation of the total phase errors of the P,FPC and DFPC algorithms vs. the number of iterations for different $N$ and $c$ values when SNR $= 30$ dB is assumed.
Kalman filtering. The EM algorithm is integrated with the KF and P_{f}FPC algorithms, and the resulting algorithm is referred to herein as EM-KF-P_{f}FPC. Now, to develop the EM and KF algorithms, we start with describing the temporal variation of the frequencies and phases of the nodes via a state-space model as follows.

A. State-Space Model

To write the state-space equation for the n-th node, let at the k-th time instant its state vector be given by \( x_{n}(k) = [f_{n}(k), \theta_{n}(k)]^{T} \) in which \( f_{n}(k) \) and \( \theta_{n}(k) \) represent its frequency and phase, respectively. Using the frequency drift and phase jitter models as described in Section II, the state-space equation for the n-th node is written as

\[
x_{n}(k) = x_{n}(k - 1) + u_{n},
\]

in which the process noise vector is defined as \( u_{n} = [\delta f_{n}, \delta \theta_{n}]^{T} \), and we assume that \( u_{n} \) is normally distributed with zero mean and the correlation matrix \( Q_{n} \) which is given by

\[
Q_{n} = \mathbb{E}[u_{n}u_{n}^{T}] = \begin{bmatrix} \sigma_{f}^{2} & -\pi T \sigma_{f}^{2} \\ -\pi T \sigma_{f}^{2} & \pi^{2} T^{2} \sigma_{f}^{2} + \sigma_{\theta}^{2} \end{bmatrix},
\]

Next to write the observation equation, let the frequency and phase estimates of the signal for the n-th node be written in the vector form as \( y_{n}(k) = [\hat{f}_{n}(k), \hat{\theta}_{n}(k)]^{T} \), then in terms of the estimation errors, this observation vector is defined as

\[
y_{n}(k) = x_{n}(k) + v_{n},
\]

in which the measurement noise vector \( v_{n} \) is given by \( v_{n} = [\varepsilon_{f}, \varepsilon_{\theta}]^{T} \) in which \( \varepsilon_{f} \) and \( \varepsilon_{\theta} \) denote the frequency and phase estimation errors, respectively. We assume that \( v_{n} \) is also normally distributed with zero mean and the correlation matrix \( \Sigma_{n} \) which is given by

\[
\Sigma_{n} = \mathbb{E}[v_{n}v_{n}^{T}] = \begin{bmatrix} (\sigma_{f}^{n})^{2} & 0 \\ 0 & (\sigma_{\theta}^{n})^{2} \end{bmatrix},
\]

in which the standard deviations \( \sigma_{f}^{n} \) and \( \sigma_{\theta}^{n} \) define the marginal distributions on the frequency and phase estimation errors, respectively.

In the above state-space model in (10) and (12), the measurement noise vector \( v_{n} \) is independent of the state vector \( x_{n}(k) \) and the process noise vector \( u_{n} \), thus the state vector estimate at each time instant can be easily obtained by using Kalman filtering as described below.

B. Kalman Filtering

Kalman filtering is an online estimation algorithm that estimates the state vector \( x_{n}(k) \) of the n-th node at the current time instant \( k \) by using all the observations up to the present time. This process involves sequentially computing the prediction-update step and the time-update step at each time instant. To this end, let the state vector estimate of the n-th node at time instant \( k - 1 \) be given by the vector \( m_{n,k-1}(k-1) \) whereas its error covariance matrix is defined by the matrix \( V_{n,k-1}(k-1) \). As annotated in the subscripts, these parameters in KF are computed by using the observations up to time \( k - 1 \). Thus, in the prediction-update step at time instant \( k \), we use the linear transformation in (10) and predict the new state vector estimate and the corresponding error covariance matrix as follows.

\[
m_{n,k}(k) = m_{n,k-1}(k-1)
\]

\[
V_{n,k}(k) = V_{n,k-1}(k-1) + Q_{n},
\]

In the time-update step, an a priori normal distribution is assumed on the state vector \( x_{n}(k) \) whose mean and covariance are set equal to the predicted estimate vector and the error covariance matrix in (14), respectively. This a priori distribution is then used to compute the a posteriori mean and covariance of the state vector given the current observation \( y_{n}(k) \) by using

\[
m_{n,k+1}(k) = m_{n,k-1}(k) + K_{n}(k) (y_{n}(k) - m_{n,k-1}(k))
\]

\[
V_{n,k+1}(k) = V_{n,k-1}(k) - K_{n}(k) V_{n,k-1}(k),
\]

where the Kalman gain matrix \( K_{n}(k) \) is given by

\[
K_{n}(k) = V_{n,k-1}(k) (V_{n,k-1}(k) + \Sigma_{n})^{-1},
\]

The mean vector \( m_{n,k}(k) \) in (15) gives the MMSE estimate of the state vector at time instant \( k \) and the matrix \( V_{n,k}(k) \) defines its error covariance matrix. These means and covariances of the state vectors are used in the EM algorithm derived below to compute the maximization step in every iteration.

C. Online Expectation-Maximization Algorithm

The prediction update and time update steps of KF in (14) and (15) assume that the process noise covariance matrix \( Q_{n} \) and the measurement noise covariance matrix \( \Sigma_{n} \) are known a priori. As discussed earlier, these model parameters are unknown in practice and must be estimated for Kalman filtering. Let \( \Theta = \{Q_{n}, \Sigma_{n}\} \) denote the set of these model parameters, then given all the instantaneous observations of the node \( n \) up to the present time \( K \) as denoted by \( y_{n}^{K} \), the marginal log-likelihood of \( \Theta \) is written as

\[
\ell(\Theta) = \ln \sum_{x_{n}^{K}} p (y_{n}^{K}, x_{n}^{K}; \Theta)
\]

\[
= \ln \prod_{k=1}^{K} \sum \frac{p (y_{n}(k)|x_{n}(k); \Sigma_{n}) p (x_{n}(k)|x_{n}(k-1); Q_{n})}{Q_{n}},
\]

where the conditional distributions in (17) can be easily found from (10) and (12) by shifting the distributions of the noise vectors to the given state vectors. Note that due to the normally distributed conditional distributions and due to the log of the summation in (17), the above objective function is a non-concave function of \( \Theta \), and directly maximizing it with respect to \( \Theta \) does not provide a closed-form solution for estimating \( Q_{n} \) and \( \Sigma_{n} \). An expectation-maximization (EM) algorithm [29], [40] is an iterative algorithm that often results in the closed-form update equations for the unknown parameters and maximizes the marginal log-likelihood function in
every iteration until convergence to its local maximum or saddle point [41]. An EM algorithm starts with an initial estimate of the unknown parameters, thus if $\Theta^{(l-1)}$ is the estimate at its $(l - 1)$-st iteration, in the $l$-th iteration it computes an expectation step (E-step) and a maximization step (M-step). In the E-step, it computes the expectation of the complete data log-likelihood function using the estimate of $\Theta$ from the previous iteration as follows.

$$L_n\left(\Theta; \Theta^{(l-1)}\right) = \mathbb{E}\left[\ln p\left(y_{1:n}^{1:K}; x_{1:n}^{1:K}; \Theta\right) | y_{1:n}^{1:K}, \Theta^{(l-1)}\right],$$  

(18)

where the expectation in (18) is a conditional expectation given the observations $y_{1:n}^{1:K}$ and the estimate $\Theta^{(l-1)}$. In the M-step, it maximizes this objective function with respect to $\Theta$ to get its new estimate by solving

$$\Theta^{(l)} = \arg\max_{\Theta} L_n\left(\Theta; \Theta^{(l-1)}\right).$$  

(19)

The above E-step and M-step are repeated iteratively until the convergence is achieved. The EM algorithm in (18) and (19) describes a batch-mode EM algorithm that is proposed in [30], [40] where first the entire data set up to the time instant $K$ is collected, and then the algorithm is run iteratively on this data set until convergence is achieved. This batch-mode EM algorithm is applicable in cases where the cost of collecting the entire data set at once is affordable, whereas in our synchronization task due to the oscillators instantaneous frequency drifts and phase jitters, we aim to instantaneously estimate $\Theta$ and update the frequencies and phases of the nodes in every iteration to synchronize these parameters across the array.

To instantaneously estimate $\Theta$ from the observed data, an online version of the EM algorithm is proposed in [29], [42] where the E-step in (18) is replaced by a stochastic approximation step in the $k$-th iteration as follows.

$$L_{n,k}\left(\Theta; \Theta^{(k-1)}\right) = L_{n,k-1}\left(\Theta; \Theta^{(k-1)}\right) + \gamma_k \left\{ \ell_{n,k}\left(\Theta; \Theta^{(k-1)}\right) \right\} - L_{n,k-1}\left(\Theta; \Theta^{(k-1)}\right),$$  

(20)

where $\gamma_k$ is a smoothing factor, and we define

$$\ell_{n,k}\left(\Theta; \Theta^{(k-1)}\right) = \mathbb{E}\left[\ln p(y_{1:n}(k), x_{1:n}(k-1); \Theta) | y_{1:n}^{1:k}, \Theta^{(k-1)}\right].$$  

(22)

Note that the iteration index in the E-step in (20) is the same as the time index due to the online nature of the algorithm. Assuming a constant smoothing factor with $\alpha_k = 1 - \alpha$, where $\alpha$ can be optimized on an initial dataset [42], the E-step in (20) can be written in compact form as

$$L_{n,k}\left(\Theta; \Theta^{(k-1)}\right) = \alpha L_{n,k-1}\left(\Theta; \Theta^{(k-1)}\right) + (1 - \alpha) \ell_{n,k}\left(\Theta; \Theta^{(k-1)}\right)$$  

$$= (1 - \alpha) \sum_{j=1}^{k} \alpha^{k-j} \ell_{n,j}\left(\Theta; \Theta^{(j-1)}\right).$$  

(23)

In the M-step of the online EM algorithm, we maximize the objective function in (23) with respect to $\Theta$ as in (19).

Now we describe the computation of the auxiliary function in (23) as follows. By using the conditional distributions from (10), (12), and (17), this function can be computed as shown in (21), shown at the bottom of the page, in which $m_{n,k}(k)$ and $m_{n,k}^\theta(k)$ are the first and second element of the vector $m_{n,k}(k)$, respectively, and the scalars $\gamma_{k,1}$ and $\gamma_{k,2}$ define the $(1,1)$-th and $(2,2)$-th indexed elements of the matrix $\Gamma_{k,k}$, respectively. The correlation matrices $\Gamma_{k-k,k-1}$, $\Gamma_{k,k}$, $\Gamma_{k,k-1}$, and the mean vector $m_{n,k}(k)$ in (21), in iteration $k$, are defined in the following. To begin, the matrix $\Gamma_{k-k,k-1}^{n}$ is defined by

$$\Gamma_{k-k,k-1}^{n-1} = \mathbb{E}\left[x_{n}(k-1)x_{n}^T(k-1) | y_{n}^{1:k}, \Theta^{(k-1)}\right] = V_{n,k}(k-1) + m_{n,k}(k-1)m_{n,k}^T(k-1),$$  

(24)

where by using the fixed-point Kalman smoothing equations from [37], we get the covariance matrix $V_{n,k}(k-1)$ as

$$V_{n,k}(k-1) = V_{n,k-1}(k-1) + U_{n,k-1}^{n} (V_{n,k-1}(k-1)) (U_{n,k-1}^{n})^T,$$  

(25)

in which

$$U_{n,k-1}^{n} = V_{n,k-1}(k-1) (V_{n,k-1}(k-1))^{-1},$$  

(26)

and we get the mean $m_{n,k}(k-1)$ in (24) as

$$m_{n,k}(k-1) = m_{n,k-1}(k-1) + U_{n,k-1}^{n} (m_{n,k}(k) - m_{n,k-1}(k)).$$  

(27)

Similarly, the correlation matrix $\Gamma_{k,k-1}^{n}$ in (21) is defined as

$$\Gamma_{k,k-1}^{n-1} = \mathbb{E}\left[x_{n}(k)x_{n}^T(k) | y_{1:n}^{1:k}, \Theta^{(k-1)}\right] = m_{n,k}(k)m_{n,k}^T(k-1) + V_{n,k}(k) (U_{n,k-1}^{n})^T,$$  

(28)

and the correlation matrix $\Gamma_{k,k}^{n}$ in (21) is given by using the Kalman filtering time-update equations as

$$\Gamma_{k,k}^{n} = \mathbb{E}\left[x_{n}(k)x_{n}^T(k) | y_{1:n}^{1:k}, \Theta^{(k-1)}\right] = V_{n,k}(k) + m_{n,k}(k)m_{n,k}^T(k),$$  

(29)

$$L_{n,k}\left(\Theta; \Theta^{(k-1)}\right) = -\frac{(1 - \alpha)}{2} \sum_{j=1}^{k} \alpha^{k-j} \left[ \ln |Q_n| + \ln \left(\sigma_{\theta}^2 + \sigma_{\sigma}^2\right) \right] + \text{tr} \left\{ Q_n^{-1} \left( \hat{\theta}_{n,j} - \Gamma_{j-j-1}^{n} \right)^T + \Gamma_{j-j-1}^{n} \right\}$$  

$$+ (\sigma_{\theta}^2)^2 \left( \frac{1}{|f_n(j)|^2} - 2f_n(j)m_{n,j}(j) + \gamma_{n}^{j,j}\right) \left(\sigma_{\theta}^2\right)^{-2} \left(\frac{1}{|\hat{\theta}_{n,j}|^2} - 2\hat{\theta}_{n,j}m_{n,j}(j) + \gamma_{n}^{j,j}\right) + \text{const.}$$  

(21)
where \( V_{n,k}(k) \) and \( m_{n,k}(k) \) are defined in (15). In fact, note that all the matrices in (24)-(29) are computed using the KF prediction-update and time-update equations defined in (14) and (15) in Section III-B.

Next we compute the M-step of the online EM algorithm in which we maximize the auxiliary function in (23) to recursively estimate \( \Theta = \{ Q_n, \Sigma_n \} \). To this end, to estimate \( Q_n \), we take the partial derivative of \( L_{n,k}(\Theta; \Theta^{(k-1)}) \) with respect to the matrix \( Q_n \) and set it equal to zero. We get the estimate of \( Q_n \) in the \( k \)-th iteration as

\[
Q_n(k) = \frac{\sum_{j=1}^{k} \alpha^{k-j} \left( \Gamma_{n,j} - \Gamma_{n,j-1} - (\Gamma_{n,j-1}^T + \Gamma_{n,j-1}) \right) \Gamma_{n,j-1}^T}{\sum_{j=1}^{k} \alpha^{k-j}},
\]

where to get the recursive form for updating \( Q_n \), we write the numerator in (30) as

\[
\xi(k) = \sum_{j=1}^{k} \alpha^{k-j} \left( \Gamma_{n,j} - \Gamma_{n,j-1} - (\Gamma_{n,j-1}^T + \Gamma_{n,j-1}) \right),
\]

\[
= \alpha \xi(k-1) + \Gamma_{n,k} - \Gamma_{n,k-1} - (\Gamma_{n,k-1}^T + \Gamma_{n,k-1-1}),
\]

thus the update equation for \( Q_n \) is written as

\[
Q_n(k) = \frac{\alpha \xi(k-1) + \Gamma_{n,k} - \Gamma_{n,k-1} - (\Gamma_{n,k-1}^T + \Gamma_{n,k-1-1})}{\lambda_k},
\]

(32)

in which \( \lambda_k \triangleq \frac{1}{1-\alpha^k} \) and we set \( \xi(0) = 0 \). Similarly the matrix \( \Sigma_n \) can also be estimated in the M-step through computing the partial derivatives. Since it is a diagonal matrix as shown in (13), it can be easily estimated by recursively estimating its diagonal elements \( \sigma^m_{f} \) and \( \sigma^m_{\theta} \) as follows. To estimate \( \sigma^m_{f} \), we compute

\[
\left( \sigma^m_{f}(k) \right)^2 = \frac{\alpha \xi(k-1) + (\hat{f}_n(k))^2 - 2\hat{f}_n(k)m_{n,k}(k) + \gamma^m_{n,k}}{\lambda_k},
\]

(33)

where \( \xi(0) = 0 \), and to estimate \( \sigma^m_{\theta} \), we use

\[
\left( \sigma^m_{\theta}(k) \right)^2 = \frac{\alpha \xi(k-1) + (\hat{\theta}_n(k))^2 - 2\hat{\theta}_n(k)m_{n,k}(k) + \gamma^m_{n,k}}{\lambda_k},
\]

(34)

where we set \( \xi(k) = 0 \). Thus, \( \sigma^m_{f}(k) \) and \( \sigma^m_{\theta}(k) \) together define the correlation matrix \( \Sigma^m_{(k)} \) in the \( k \)-th iteration.

This completes the derivation of the online EM algorithm. For further detail, the pseudo-code of the resulting algorithm which combines EM, KF, and P-FPC is given in Algorithm 2.

D. Discussions

In this subsection, we begin with describing the initialization of the KF algorithm in each iteration, and then compute the computational complexity of the proposed EM-KF-P,FPC algorithm. Note that the initialization of the EM algorithm is included in Section III-E where the simulation results are discussed.

**Algorithm 2** EM-KF-P,FPC Algorithm

**Input:** \( k = 0 \), define \( s_n(0) = 1 \) for each node \( n \), initialize \( m_{n,0}(0) \) and \( V_{n,0}(0) \).

**while** convergence criterion is not met do

\( k = k + 1 \)

For each node \( n \):

a) Obtain the observation vector \( y_n(k) = [\hat{f}_n(k), \hat{\theta}_n(k)]^T \).
b) Compute the prediction update step of KF.
c) Run the time update step of KF by computing \( m_{n,k-1}(k) \) and \( V_{n,k-1}(k) \) from (14).
d) Estimate \( Q_n(k) \) and \( \Sigma^m_{n}(k) \) matrices using (32)-(34).
e) For each \( m \in x_n \), let \( m_{n,m}(k) \triangleq [m_{f,n,k}(k), m_{\theta,n,k}(k)]^T \), then update each node \( n \) temporary variables by:

\[
x_{f,n}(k) = \sum_{m \in x_n} w_{n,m} m_{f,n,k}(k) s_m(k-1),
\]

\[
x_{\theta,n}(k) = \sum_{m \in x_n} w_{n,m} m_{\theta,n,k}(k) s_m(k-1),
\]

\[
s_n(k) = \sum_{m \in x_n} w_{n,m} s_m(k-1),
\]

Next update \( f_n(k) \) and \( \theta_n(k) \) using

\[
f_n(k) = \frac{x_{f,n}(k)}{s_n(k)},
\]

\[
\theta_n(k) = \frac{x_{\theta,n}(k)}{s_n(k)}.
\]

**end**

**Output:** \( f_n(k) \) and \( \theta_n(k) \) for all \( n = 1, 2, \ldots, N \)

\( \text{a) Initialization: To perform the prediction update step of KF in the iteration } k = 1 \text{ of the EM-KF-P,FPC algorithm, we define the mean vector } m_{n,0}(0) \text{ and the covariance matrix } V_{n,0}(0) \text{ of the } n \text{-th node as } m_{n,0}(0) = [f_c, \pi]^T \text{ and } V_{n,0}(0) = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 4\pi^2/12 \end{bmatrix}. \) Next since the posterior distribution on the state vector of node \( n \) is a normal distribution and the means undergo a linear transformation in Step (c) of EM-KF-P,FPC algorithm, we initialize the prediction-update step of KF in the \( k > 1 \) iteration by defining the mean vector from the previous iteration as \( m_{n,k-1}(k-1) = [f_n(k-1), \theta_n(k-1)]^T \) and the
covariance matrix $\mathbf{V}_{n,k-1}(k-1)$ as

$$
\mathbf{V}_{n,k-1}(k-1) = \begin{bmatrix} \sum_{m \in \chi_n^1} \nu_{m,n}^{-2} v_{m,n}^1(1) & \sum_{m \in \chi_n^1} \nu_{m,n}^{-2} v_{m,n}^2(1) \\ \sum_{m \in \chi_n^1} |s_n(1)|^2 & \sum_{m \in \chi_n^2} \nu_{m,n}^{-2} v_{m,n}^2(1) \\ \sum_{m \in \chi_n^2} |s_n(1)|^2 & \sum_{m \in \chi_n^2} \nu_{m,n}^{-2} v_{m,n}^2(1) \end{bmatrix},
$$

(35)

where $\nu_{m,n}^{-2} \triangleq |w_{m,n} s_n(2)|^2$ and the components $v_{m}^1(1)$, $v_{m}^2(1)$, and $v_{m}^2(1)$ are the $(1,1)$-th, $(1,2)$-th, and $(2,2)$-th elements, respectively, of the covariance matrix computed at node $m$ in the $(k-1)$-st iteration of EM using (15). Noticeably the proposed EM-KF-P,FPC algorithm is a distributed algorithm in which the nodes run the EM and KF algorithms on their own observations, in parallel, and then locally share their MMSE estimates and the error covariances with their neighboring nodes to update the electrical states in step (e) across the array. The shared covariances are used to define the priors at each node for Kalman filtering in step (b).

b) Computational Complexity: For the $n$-th node in the array, the computational complexity of EM-KF-P,FPC per iteration is dominated by (3), (16), and (26). Equation (3) is part of the P,FPC algorithm which has the computational complexity of $O(\text{card}(\chi_n^1))$, where the notation $\text{card}(\chi_n^1)$ defines the cardinality of the set $\chi_n^1$. Equations (16) and (26) are part of the EM and KF algorithms, respectively, which require inverting the $2 \times 2$ matrices and then multiplying the $2 \times 2$ matrices. These two equations have the computational complexity of $O(8)$. Now since the KF and EM step in the synchronization algorithm for each node $n$ can be run in parallel, the computational complexity of the overall EM-KF-P,FPC algorithm is $O(\text{card}(\chi_n^1) + 8)$. This shows that for the sparsely connected arrays with $\text{card}(\chi_n^1) \ll 8$, the computational complexity is $O(8)$, whereas for the moderately connected large arrays with $\text{card}(\chi_n^1) \gg 8$, it can be approximated as $O(\text{card}(\chi_n^1))$. Thus, for the moderately connected large arrays, using the EM and KF algorithms result in an improved synchronization performance of P,FPC without any additional increase in the computational complexity. This result is the same as was observed for the KF-DPCF algorithm in [11] which was proposed for the bidirectional array networks. Furthermore, we also note that the computational complexity of EM-KF-P,FPC is significantly lower than that of the VB-HCMCI algorithm in [31] which not only additionally requires consensus on measurements and consensus on the predicted information at each node that comes at the cost of increase in the bandwidth requirement between the nodes and the increase in the latency for encoding the information, but they also need to perform multiple iterations at each time instant $k$ and at each node $n$ for the VB’s convergence, before progressing on to the next time instant $k + 1$.

Fig. 5. Frequency errors of the EM-KF-P,FPC vs. the number of iterations for $N = 100$ nodes, $c = 0.2$, $T = 0.1$ ms, and SNR = 0 dB.

E. Simulation Results

In this subsection, we evaluate the frequency and phase synchronization performances of the proposed EM-KF-P,FPC algorithm by varying the number of nodes $N$ in the array, the connectivity $c$ between the nodes, and the SNR of the observed signals. For the demonstration purposes, we consider two cases of the initializations for the EM algorithm in EM-KF-P,FPC, i.e., case (a) represents a poor initialization of EM with $Q_n^{(0)} = \left[ \begin{array}{cc} \hat{\sigma}_f^2 & 0 \\ 0 & \pi^2 T \hat{\sigma}_f^2 \end{array} \right]$ in which $\hat{\sigma}_f^2 = \frac{1}{\sqrt{N}}$, and with $\Sigma_n^{(0)} = \left[ \begin{array}{cc} 10^{3} & 0 \\ 0 & 10^{-12} \end{array} \right]$, whereas case (b) represents a good initialization of EM with $Q_n^{(0)} = Q_n$ and $\Sigma_n^{(0)} = \left[ \begin{array}{cc} 10^{3} & 0 \\ 0 & 10^{-12} \end{array} \right]$. Note that the initialization in case (a) uses a poor estimate of $Q_n$ that ignores the cross-correlation terms in $Q_n$, uses an estimate of $\sigma_f^2$, and assumes that there is no phase jitter at the nodes with $\hat{\sigma}_\theta = 0$. In contrast, the initialization in case (b) uses the true $Q_n$ matrix. Furthermore, both cases use a poor estimate of $\Sigma_n$, where the true diagonal elements can be computed using the considered simulation parameters in (13) as $(\sigma_f^2)$ and $(\sigma_\theta^2) = 4 \times 10^{-12}$.

For the comparison purposes, we plot the performance of the KF-P,FPC with known process noise and measurement noise covariance matrices and refer to it as the Genie-aided-KF-P,FPC. In addition, we also show the performance of KF-P,FPC when these noise covariance matrices are not updated over the iterations and the algorithm uses the initially estimated covariance matrices which is referred to herein as the Naive-KF-P,FPC algorithm. The carrier frequency of the nodes is chosen as $f_c = 1$ GHz, the sampling frequency is set as $f_s = 10$ MHz, and the update interval is $T = 0.1$ ms. The results in this section were averaged over $10^3$ independent trials.

Figs. 5 and 6 show the frequency and phase errors of all the $N$ nodes in the array vs. the number of iterations of the EM-KF-P,FPC algorithm (initialized with the case (b) setting) when connectivity $c = 0.2$ and SNR = 0 dB are assumed. It is observed that as the number of iterations increases both

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The EM-KF-P algorithm is initialized with the case (b) setting, and the SNR is assumed. It is observed that for each connectivity \( c \) setting, and the SNR is 30 dB, and for different initialization of the noise covariance matrices (\( Q_n \) and \( \Sigma_n \)) in the online EM algorithm. For both initializations and for both \( N = 20 \) and 40 nodes, we evaluate the residual phase error of EM-KF-P,FPC vs. the number of iterations and compare it against Naive-KF-P,FPC and the Genie-aided-KF-P,FPC as shown in the figure. It is observed that with the case (a) initialization and for both \( N \) values, the Naive-KF-P,FPC algorithm although converges faster but results in a higher residual phase error as compared to EM-KF-P,FPC. Our EM-KF-P,FPC algorithm takes about 50 to 80 more iteration to converge when \( N \) varies from 20 to 40 nodes, respectively, than the Naive-KF-P,FPC algorithm but significantly reduces the residual phase error upon convergence. Note that, in this case, the residual phase error of EM-KF-P,FPC is far from the Genie-aided-KF-P,FPC for both \( N \) values due to the convergence of EM to the local maximum near the initial estimate as discussed in Section III-C. In contrast, in case (b), where a better initial estimate is available, the Naive-KF-P,FPC continues to show a higher residual phase error as compared to EM-KF-P,FPC, whereas the EM-KF-P,FPC algorithm follows both the convergence speed and the residual phase error of the Genie-aided-KF-P,FPC. To summarize, in both cases, EM-KF-P,FPC performs better than Naive-KF-P,FPC, and in particular with a good initialization of EM, EM-KF-P,FPC converges in residual phase error to the Genie-aided-KF-P,FPC as expected. Thus, from here onwards, we use the case (b) initialization for the EM-based algorithms to analyze their performances.

In Fig. 8, we analyze the convergence properties of the EM-KF-P,FPC algorithm by evaluating the standard deviation of the residual phase error vs. the number of iterations for \( N = 20 \) and 40 nodes in the array with connectivity \( c = 0.2 \) and SNR = 30 dB, and for different initialization of the Naive-KF-P,FPC and EM-KF-DPFC algorithms vs. the number of iterations for different \( N \) values and for different initialization of EM when \( c = 0.2 \), and for SNR = 30 dB.

The frequencies and phases of all the nodes converge to the average of their initial values as expected. Furthermore, comparing these figures with Figs. 1(b) and 2(b), it is apparent that the use of Kalman filtering with a consensus averaging algorithm significantly reduces the residual errors upon convergence.

Fig. 7 compares the standard deviation of the total phase errors of the P,FPC and EM-KF-P,FPC algorithms by varying the number of nodes \( N \) in the array and the connectivity \( c \) between the nodes. The EM-KF-P,FPC algorithm is initialized with the case (b) setting, and the SNR is 30 dB is assumed. It is observed that for each \( N \) and \( c \) values, the EM-KF-P,FPC algorithms significantly reduces the residual phase error, and the phase errors continue to decrease with the increase in either \( N \) or \( c \) due to the reasons explained earlier in Section II-C. Specifically, for \( N = 100 \) and \( c = 0.2 \), the phase errors of P,FPC and EM-KF-P,FPC are 1.4 × 10^{-2} degs. and 2.7 × 10^{-3} degs., respectively, whereas when \( c \) increases to 0.5, the phase errors decrease to 7.2 × 10^{-3} degs. and 1.4 × 10^{-3} degs., respectively.
Residual phase errors of the EM-KF-P Kalman filtering algorithm (for details, see Fig. 9 in [11]). Observations (iterations) to accurately estimate the states in the consensus averaging; whereas, in particular, the increase in the phase errors of the genie-aided schemes at the lower SNRs thus apparently do not suffer from the convergence errors as the EM-based algorithms at the lower SNR consensus averaging; whereas, in particular, the increase in the phase errors of the genie-aided schemes at the lower SNRs thus apparently do not suffer from the convergence errors as the EM-KF-P, FPC algorithms by varying the number of nodes for SNR = 30 dB. It is observed that as the value of c or N increases, the performance of all the algorithms improves due to the increase in the average number of connections per node (D) and a decrease in the network algebraic connectivity (λ2) as discussed before. For all the c and N values, the EM-KF-P, FPC algorithm performs better than the Naive-KF-P, FPC algorithm and similar to the Genie-aided-KF-P, FPC algorithm. In this figure, we also extended the KF-DPC algorithm of [11] to integrate the online EM algorithm with it, and thus evaluated the performance of the EM-KF-DPC algorithm and the Genie-aided-KF-DPC algorithms. It is observed that for both c = 0.2 and 0.5, and all N values, EM-KF-DPC performs similar to Genie-aided-KF-DPC but better than the EM-KF-P, FPC algorithm. This is expected because the DPC-based algorithms were evaluated using undirected array networks which entail more exchange of information between the nodes as compared to the directed networks which were used for the P, FPC-based algorithms. Note that, as mentioned before, the DPC-based algorithms are only applicable to the undirected array networks with bidirectional links between the nodes, whereas for the directed array networks, they do not converge due to the use of average consensus [15] as the underlying algorithm. However, our proposed P, FPC-based algorithms are applicable for both types of array networks, whereas in the case of undirected networks P, FPC-based algorithms results in the same synchronization and convergence performance as the DPC-based algorithms. Thus, the performances of the DPC-based algorithms in this work can also be interpreted as the performances of the P, FPC-based algorithms for the directed array networks to draw a comparison of directed vs. undirected array network.

Finally, Fig. 10 also shows the residual phase error of the HCMCI algorithm of [32] for the undirected array networks with L = 1 consensus steps, which is the base algorithm of VB-HCMCI (see Table I in [31]). To this end, we assume here that the noise covariance matrices are known to the HCMCI algorithm, and thus refer to it as the Genie-aided-HCMCI algorithm. It is observed that the Genie-aided-HCMCI performs slightly better than the EM-KF-DPC (EM-KF-P, FPC) algorithm for all the c and N values. As discussed earlier, this is because HCMCI-based algorithms additionally need to perform a consensus on the neighboring nodes’ measurements and a consensus on their predicted information matrices in each iteration as compared to the KF-DPC and KF-P, FPC based algorithms which only perform consensus on the MMSE estimates and the error covariances at the neighboring nodes. Hence, the latter ones are computationally less complex and have lower latency in encoding the information and the lower bandwidth requirement for sharing that information between the nodes as compared to the HCMCI-based algorithms.

IV. CONCLUSION

We considered the problem of joint frequency and phase synchronization of the nodes in distributed phased arrays. To this end, we included the frequency drift and phase jitter of the oscillator, as well as the frequency and phase estimator errors in our signal model. We considered a more

![Fig. 9. Standard deviation of the total phase errors of the P, FPC, EM-KF-P, FPC, DPC, and EM-KF-DPC algorithms vs. SNR (dB) for different c values when N = 100 and T = 0.1 ms.](image1)

![Fig. 10. Standard deviation of the total phase errors of the EM-KF-P, FPC and EM-KF-DPC algorithms vs. the number of nodes N for different c values when SNR = 30 dB and T = 0.1 ms.](image2)
practical case, that is when there exist directed communication links between the nodes in an array. A push-sum protocol based frequency and phase consensus algorithm referred to herein as the P\textsubscript{FPC} algorithm was proposed, which is a modified version of the push-sum protocol as it takes into account the states transitioning in between the update intervals. Kalman filtering was also integrated with P\textsubscript{FPC} to propose the KF-P\textsubscript{FPC} algorithm. KF assumes that the process noise and the measurement noise covariance matrices are known which is an impractical assumption. An online EM based algorithm is developed that iteratively computes the ML estimates of these unknown matrices which are then used for KF. The residual phase error of EM upon convergence is defined by the initialization which is expected due to the non-concave structure of the objective function. However, given a good initialization, the proposed EM-KF-P\textsubscript{FPC} algorithm significantly improves the residual phase error at the moderate SNR values as compared to the P\textsubscript{FPC} and the Naive-KF-P\textsubscript{FPC} algorithms, and also shows similar synchronization and convergence performance as that of the Genie-aided-KF-P\textsubscript{FPC}. The residual phase error and the convergence speed of EM-KF-P\textsubscript{FPC} improves by increasing the number of nodes in the array, the connectivity between the nodes, and the SNR of the observed signals. Furthermore, the computational complexity of EM-KF-P\textsubscript{FPC} is lower than that of the HCMCI-based algorithms and is the same as the computational complexity of the KF-P\textsubscript{FPC} and DF-P\textsubscript{FPC} algorithms. In particular, for the moderately connected large array, the use of EM and KF doesn’t increase the computational complexity of P\textsubscript{FPC}.

REFERENCES

[1] S. M. Ellison, S. R. Mghabghab, and J. A. Nanzer, “Multi-node openloop distributed beamforming based on scalable, high-accuracy ranging,” IEEE Sensors J., vol. 22, no. 2, pp. 1629–1637, Jan. 2022.
[2] S. Schieler et al., “OFDM waveform for distributed radar sensing in automotive scenarios,” in Proc. 16th Eur. Radar Conf. (EurRAD), 2019, pp. 225–228.
[3] B. Pardhasaradhi and L. R. Cenkeramaddi, “GPS spoofing detection and mitigation for drones using distributed radar tracking and fusion,” IEEE Sensors J., vol. 22, no. 11, pp. 11122–11134, Jun. 2022.
[4] U. Madhow, D. R. Brown, S. Dasgupta, and R. Mudumbai, “Distributed massive MIMO: Algorithms, architectures and concept systems,” in Proc. Inf. Theory Appl. Workshop (ITA), Feb. 2014, pp. 1–7.
[5] J. A. Nanzer, S. R. Mghabghab, S. M. Ellison, and A. Schlegel, “Distributed phased arrays: Challenges and recent advances,” IEEE Trans. Microw. Theory Techn., vol. 69, no. 11, pp. 4893–4907, Nov. 2021.
[6] K.-Y. Tu and C.-S. Liao, “Application of AFNI for frequency synchronization using GPS carrier-phase measurements,” in Proc. IEEE Int. Freq. Control Symp. Joint 21st Eur. Freq. Time Forum, May 2007, pp. 933–936.
[7] R. Mudumbai, B. Wild, U. Madhow, and K. Ramch, “Distributed beamforming using 1-bit feedback: From concept to realization,” in Proc. 44th Allerton Conf. Commun., Control Comput., 2006, pp. 1020–1027.
[8] W. Tushar and D. B. Smith, “Distributed transmit beamforming based on a 3-bit feedback system,” in Proc. IEEE 11th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC), Jun. 2010, pp. 1–5.
[9] B. Peiffer, R. Mudumbai, S. Goguri, A. Kruger, and S. Dasgupta, “Experimental demonstration of retrodirective beamforming from a fully wireless distributed array,” in Proc. IEEE MIL. Commun. Conf. (MILCOM), Nov. 2016, pp. 442–447.
[10] J. A. Nanzer, R. J. Schmid, T. M. Comberiate, and J. E. Hodkin, “Openloop coherent distributed arrays,” IEEE Trans. Microw. Theory Techn., vol. 65, no. 5, pp. 1662–1672, May 2017.
[11] M. Rashid and J. A. Nanzer, “Frequency and phase synchronization in distributed antenna arrays based on consensus averaging and Kalman filtering,” IEEE Trans. Wireless Commun., early access, Oct. 18, 2022, doi: 10.1109/TWC.2022.3213788.
[12] S. R. Mghabghab and J. A. Nanzer, “Open-loop distributed beamforming using wireless frequency synchronization,” IEEE Trans. Microw. Theory Techn., vol. 69, no. 1, pp. 896–905, Jan. 2021.
[13] M. Ponton and A. Suárez, “Stability analysis of wireless coupled-oscillator circuits,” in IEEE MTT-S Int. Microw. Symp. Dig., Jun. 2017, pp. 83–86.
[14] M. Pontón, A. Herrera, and A. Suárez, “Wireless-coupled oscillator systems with an injection-locking signal,” IEEE Trans. Microw. Theory Techn., vol. 67, no. 2, pp. 642–658, Feb. 2019.
[15] S. Boyd, P. Diaconis, and L. Xiao, “Fastest mixing Markov chain on a graph,” SIAM Rev., vol. 46, no. 4, pp. 667–689, 2003.
[16] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” IEEE Trans. Signal Process., vol. 58, no. 3, pp. 1035–1048, Mar. 2010.
[17] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in Proc. 48th IEEE Int. Conf. Decis. Control (CDC) Held Jointly 28th Chin. Control Conf., Dec. 2009, pp. 7036–7042.
[18] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” IEEE Trans. Autom. Control, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.
[19] D.-J. Xin, L.-F. Shi, and X. Yu, “Distributed Kalman filter with faulty/reliable sensors based on Wasserstein average consensus,” IEEE Trans. Circuits Syst. I, Exp. Briefs, vol. 69, no. 4, pp. 2371–2375, Apr. 2022.
[20] F. L. Walls and J. Gagnepain, “Environmental sensitivities of quartz oscillators,” IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 39, no. 2, pp. 241–249, Mar. 1992.
[21] R. L. Filler and J. R. Vig, “Long-term aging of oscillators,” IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 40, no. 4, pp. 387–394, Jul. 1993.
[22] S.-Y. Wang, B. Neubig, J.-H. Wu, T.-F. Ma, J.-K. Du, and J. Wang, “Extension of the frequency aging model of crystal resonators and oscillators by the Arhenius factor,” in Proc. Symp. Piezoelectricity, Acoustic Waves, Device Appl. (SPAWD4), Oct. 2016, pp. 269–272.
[23] M. Goryachev, S. Galliou, P. Abbe, and V. Komine, “Oscillator frequency stability improvement by means of negative feedback,” IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 58, no. 11, pp. 2297–2304, Nov. 2011.
[24] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, “Push-sum distributed dual averaging for convex optimization,” in Proc. IEEE 51st IEEE Int. Conf. Decis. Control (CDC), Dec. 2012, pp. 5453–5458.
[25] H. Ouassal, T. Rocco, M. Yan, and J. A. Nanzer, “Decentralized frequency synchronization in distributed antenna arrays with quantized frequency states and directed communications,” IEEE Trans. Antennas Propag., vol. 68, no. 7, pp. 5280–5288, Jul. 2020.
[26] M. Rashid and J. A. Nanzer, “Push-sum protocol based frequency and phase synchronization in distributed phased arrays,” in Proc. IEEE Int. Symp. Phased Array Syst. Technol. (PAST), Oct. 2022, pp. 1–4.
[27] S. R. Mghabghab and J. A. Nanzer, “Impact of VCO and PLL phase noise on distributed beamforming arrays with periodic synchronization,” IEEE Access, vol. 9, pp. 56578–56588, 2021.
[28] P. Chatterjee and J. A. Nanzer, “A study of coherent gain degradation due to node vibrations in open loop coherent distributed arrays,” in Proc. USNC-URSI Radio Sci. Meeting (Joint AP-S Symposium), Jul. 2017, pp. 115–116.
[29] O. Cappé and E. Moulines, “On-line expectation-maximization algorithm for latent data models,” J. Roy. Stat. Soc., B (Stat. Methodol.), vol. 71, no. 3, pp. 393–413, Jun. 2009.
[30] D.-J. Xin and L.-F. Shi, “Kalman filter for linear systems with unknown structural parameters,” IEEE Trans. Circuits Syst. I, Exp. Briefs, vol. 69, no. 3, pp. 1852–1856, Mar. 2022.
[31] X. Dong, G. Battistelli, L. Chisci, and Y. Cai, “An adaptive consensus filter for distributed state estimation with unknown noise statistics,” IEEE Signal Process. Lett., vol. 28, pp. 1595–1599, 2021.
[32] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, “Consensus-based linear and nonlinear filtering,” IEEE Trans. Autom. Control, vol. 60, no. 5, pp. 1410–1415, May 2015.
[33] H. Ouassal, M. Yan, and J. A. Nanzer, “Decentralized frequency alignment for collaborative beamforming in distributed phased arrays,” IEEE Trans. Wireless Commun., vol. 20, no. 10, pp. 6269–6281, Oct. 2021.
Mohammed Rashid (Member, IEEE) received the B.Sc. degree (Hons.) in electrical engineering from the NWFP University of Engineering and Technology, Peshawar, Pakistan, in 2012, and the M.Sc. and Ph.D. degrees in electrical engineering from Louisiana State University (LSU), Baton Rouge, LA, USA, in 2018 and 2021, respectively.

During his graduate studies, he was a Graduate Research and Teaching Assistant with LSU, where he was also nominated for membership in the Honor Society of Phi Kappa Phi. Since August 2021, he has been a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA, where he is working on developing distributed consensus algorithms for synchronizing antenna nodes in distributed phased arrays. His research interests include Bayesian machine learning, Bayesian compressive sensing, algorithms for approximate inference in probabilistic graphical models, distributed phased arrays, radar-based sensing, and statistical information processing in systems. For his B.Sc. degree, he was awarded the University Gold Medal for distinction from the NWFP University of Engineering and Technology.

Jeffrey A. Nanzer (Senior Member, IEEE) received the B.S. degree in electrical engineering and the B.S. degree in computer engineering from Michigan State University, East Lansing, MI, USA, in 2003, and the M.S. and Ph.D. degrees in electrical engineering from The University of Texas at Austin, Austin, TX, USA, in 2005 and 2008, respectively.

From 2008 to 2009, he was with the University of Texas Applied Research Laboratories, Austin, TX, as a Post-Doctoral Fellow of designing electrically small HF antennas and communications systems. From 2009 to 2016, he was with the Johns Hopkins University Applied Physics Laboratory, where he created and led the Advanced Microwave and Millimeter-Wave Technology Section. In 2016, he joined the Department of Electrical and Computer Engineering, Michigan State University, where he held the Dennis P. Nyquist Assistant Professorship from 2016 to 2021. He is currently an Associate Professor with Michigan State University. He directs the Electromagnetics Laboratory, which consists of the Antenna Laboratory, the Radar Laboratory, and the Wireless Laboratory. He has published more than 200 refereed journal and conference papers, two book chapters, and the book Microwave and Millimeter-Wave Remote Sensing for Security Applications (Artech House, 2012). His research interests include distributed phased arrays, dynamic antenna arrays, millimeter-wave imaging, remote sensing, millimeter-wave photonics, and electromagnetics.

Dr. Nanzer was a Founding Member and the First Treasurer of the IEEE APS/MTT-S Central Texas Chapter. He is a member of the IEEE Antennas and Propagation Society Education Committee and the URSI/USNC Commission B. He is a Distinguished Microwave Lecturer for the IEEE Microwave Theory and Techniques Society (Tatsuo Itoh Class of 2022–2024). He was a recipient of the Google Research Scholar Award in 2022, the IEEE MTT-S Outstanding Young Engineer Award in 2019, the DARPA Director’s Fellowship in 2019, the National Science Foundation (NSF) CAREER Award in 2018, the DARPA Young Faculty Award in 2017, and the JHU/APL Outstanding Professional Book Award in 2012. He served as the Vice Chair for the IEEE Antenna Standards Committee from 2013 to 2015. He served as the Chair for the Microwave Systems Technical Committee (MTT-16), IEEE Microwave Theory and Techniques Society, from 2016 to 2018. He was a Guest Editor for the Special Issue on Radar and Microwave Sensor Systems of the IEEE MICROWAVE AND WIRELESS COMPONENTS LETTERS in 2022. He is currently an Associate Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION.