Abstract—A decentralized approach for joint frequency and phase synchronization in distributed antenna arrays is presented. The nodes in the array locally broadcast their frequencies and phases to their neighboring nodes, and use consensus averaging to align these parameters across the array. The architecture is fully distributed, requiring no centralization. Each node has a local oscillator and we consider a signal model where intrinsic frequency and phase errors of the local oscillators on each node caused by the frequency drift and phase jitter as well as the frequency and phase estimation errors at the nodes are included and modeled using practical statistics. A decentralized frequency and phase consensus (DFPC) algorithm is proposed which uses an average consensus method in which each node in the array iteratively updates its frequency and phase by computing an average of the frequencies and phases of their neighboring nodes. Simulation results show that upon convergence the DFPC algorithm can align the frequencies and phases of all the nodes up to a residual phase error that is governed by the oscillators and the estimation errors. To reduce this residual phase error and thus improve the synchronization between the nodes, a Kalman filter based decentralized frequency and phase consensus (KF-DFPC) algorithm is presented. The total residual phase error at the convergence of the KF-DFPC and DFPC algorithms is derived theoretically. The synchronization performances of these algorithms are compared to each other and to the diffusion least mean square (DLMS), the diffusion Kalman filter (DKF) algorithm, and the Kalman consensus information filter (KCIF) algorithms, in light of this theoretical residual phase error by varying the duration of the signals, connectivity of the nodes, the number of nodes in the array, and signal to noise ratio of the received signals. Simulation results demonstrate that under certain conditions the proposed KF-DFPC algorithm outperforms and converges in fewer iterations than all the algorithms. Furthermore, for shorter intervals between local information broadcasting, the KF-DFPC algorithm significantly outperforms the DFPC algorithm in reducing the residual total phase error, irrespective of the signal to noise ratio of the received signals.

Index Terms—Distributed phased arrays, frequency and phase synchronization, average consensus, Kalman filtering.

I. INTRODUCTION

Wireless systems evolution from large single-platform architectures to a composite of multiple small spatially distributed systems that make local decisions to achieve a shared global objective has resulted in significant applications in distributed coordinated beamforming [1], [2], network densification in 5G [3], and coherent distributed array (CDA) [4], [5], [6]. In particular, a CDA operates as a network of small, low-cost, and spatially distributed nodes where each node has its own separate transceiver, and the nodes align their transceivers within a fraction of the wavelength of the wireless operation in such a way that their signals add up coherently at the destination. This coordination of the nodes at the level of the wavelength imitates a distributed phased array and results in several advantages at the overall system level that include improved signal-to-noise (SNR) ratio at the destination, enhanced resistance to the system failure, greater spatial diversity, and considerable ease in the scalability of the system [7].

In coherent distributed arrays, each node has its own local oscillator in its transceiver chain. In an unlocked state, these oscillators undergo random frequency and phase drifts, and phase jitter over time due to factors such as temperature variations, noise from the supply voltage, and manufacturing tolerances of its components, etc. [8]. If the nodes generate their carrier signals by using reference signals from their own local oscillators then due to these frequency and phase offsets, there exist a decoherence between their generated signals which distorts the coherent gain at the destination. We can connect these nodes via wires to a common reference signal [9], but that severely limits the mobility of the nodes as well as the application domain of such arrays. Another way is to connect each node to the global positioning system (GPS) signal and use it to synchronize them across the array [10]. However, GPS-based systems are expensive, consume high power, do not work in indoor environments, and are too inaccurate for distributed arrays. Furthermore, wireless locking of the oscillators to a central oscillator have also been investigated using the wireless signals in [11] and [12] and using the optical signals in [13]. However, these methods are incapable of synchronizing the nodes at larger distances and are also not scalable.

To date, several methods have also been proposed for wireless synchronizing of the nodes in a coherent distributed
array that can be broadly classified as either closed-loop or open-loop methods. In the closed-loop methods, such as the 1-bit and 3-bit feedback methods in [14], [15], [16], and [17] and the retrodirective method in [18], the nodes adjust their electrical states based on the feedback from the destination. This feedback is in terms of some useful information, e.g., received signal strengths, preambles, or data throughput, etc., and the nodes use the feedback to adjust their states with the intent to reach a desirable coherent gain at the destination. However, closed-loop methods are only suitable for scenarios where such meaningful feedback is available from the destination, for instance, in communication applications, thus limiting its ability to arbitrarily steer the beam to any destination. On the other hand, in the open-loop methods, such as the synchronization methods in [19], [20], [21], and [22], the nodes coordinate with each other to synchronize their transceivers without using any feedback from the destination. As such, open-loop methods are also suitable for remote sensing [23], [24] and radar applications [25].

For open-loop CDAs, a centralized topology based transceiver synchronization method is proposed in [22], [26], and [27] where a primary node transmits a reference signal to one or more secondary nodes, and the secondary nodes use a frequency locking circuit with phase-locked loops to synchronize their frequencies with the frequency of the primary node. A drawback of this primary-secondary architecture is that it fails whenever the primary node fails. Thus the authors in [19] and [20] proposed a decentralized algorithm for node synchronization in open-loop CDAs. The decentralized topology overcomes the shortcomings of the centralized one, and makes the system easily scalable.

One limitation of [19], [20] is that it only considered frequency synchronization of the oscillators in distributed arrays, however, in practice, the phases of the oscillators also undergo random drifts and jitters over time and thus need to be aligned as well for achieving high coherent gain at the destination [8], [28]. Consequently, we consider the joint frequency and phase synchronization of the nodes in a distributed antenna array, and to this end, the contributions made in this paper are summarized as follows.

- We consider a signal model for the nodes in which the oscillator frequency and phase drifts as well as the phase jitters are included to demonstrate a more practical scenario. Furthermore, due to the addition of these random offsets, we assume that the nodes iteratively estimate their frequencies and phases before updating them, and thus the estimation errors are also included in the signal model. An iterative decentralized frequency and phase consensus (DFPC) algorithm is proposed which synchronizes the nodes across the array in a few iterations. This joint frequency and phase synchronization was also studied in [29] and [30], but their proposed algorithm synchronizes the oscillators to the harmonic means of their initial frequencies [29]. On the contrary, to average out the offset errors and thereby reduce the residual phase errors, it is shown in Section III-A in this paper that the convergence to the arithmetic average value is essential in the presence of the frequency drifts and phase jitters of the oscillators, which is ensured by DFPC.
- Steady-state residual total phase error of the DFPC algorithm is also theoretically derived herein by taking into account the frequency and phase drifts, phase jitters, and frequency and phase estimation errors at the nodes. It is observed that the residual phase error of DFPC decreases with the increase in either the number of nodes in the array or the connectivity between the nodes. This is also illustrated through simulation in Section V where the phase errors are analyzed for different arrays by varying the update intervals and the SNR of the signals.
- In practice, usually the frequencies and phases of the nodes have to be updated periodically with smaller update intervals to avoid the decoherence between the nodes caused by the large oscillators drift in longer time intervals, and to minimize the system impacts due to the platform vibrations [31]. However, Section V shows that the DFPC algorithm results in larger residual phase errors at the smaller update intervals. Similar behavior was also observed for the frequency synchronization algorithms proposed in [19] and [20] which is caused by an increase in the estimation errors. Kalman filtering (KF) is a popular method that has been used in the past in a variety of problems ranging from target tracking to the industrial control [32]. Essentially, it computes the optimal minimum mean squared error estimate of the unknown quantities when the process noise in the state transitioning model and the measurement noise of the observations are normally distributed, and the observations are a linear function of the unknown states [33]. To take advantage of this KF property, we integrate KF with DFPC to reduce the residual phase errors at the shorter update intervals where the measurement errors are dominant and the states are slowly time-varying. The resulting algorithm is referred to as the KF-DFPC algorithm. Simulation results are included where the improvement due to KF at the smaller update intervals is illustrated and compared to DFPC. Furthermore, the synchronization performance and the computational complexities of KF-DFPC and DFPC are compared to the earlier proposed diffusion least mean square (LMS) algorithm [34], the Kalman consensus information filter (K CIF) algorithm [35], and the diffusion KF algorithm [36], [37]. The results show that our proposed KF-DFPC converges faster at lower SNRs and when sparsely connected arrays are used, whereas for the large densely connected arrays, the improvement with using KF comes with no additional increase in the computational complexity of the DFPC algorithm.

The upcoming sections are outlined as follows. Section II introduces the frequency and phase synchronization problem in a distributed antenna array, and describes the modeling of the frequency and phase errors. Section III proposes the DFPC algorithm, derives the steady-state total phase error, and analyzes DFPC performance through simulations. To reduce the residual total phase error and improve the synchronization between nodes, Kalman filtering based KF-DFPC algorithm...
is proposed in Section IV wherein it is also studied through simulations. Section V analyzes the residual phase errors of DFPC and KF-DFPC for different update intervals and connectivity values. Finally, Section VI concludes this work.

**Notations:** Herein, small letters \((x)\) are used to represent scalars or signals, bold small letters \((\mathbf{x})\) are used for matrices. The superscripts \((\cdot)^T\) and \((\cdot)^{-1}\) represent the matrix transpose and inverse operations, respectively. \(N(\mu, \Sigma)\) denotes a normal distribution with mean \(\mu\) and covariance matrix \(\Sigma\). \(\mathbf{I}_N\) denotes the \(N \times N\) identity matrix. In \(\mathbb{E}[\cdot], \mathbb{E}[\cdot]\) denotes the expectation operation with respect to the probability distribution on \(\mathbf{x}\). \(\pi_{1:k}\) is a shorthand form for the set \(\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}\). Finally, \(\mathbf{X} = \text{diag}\{\mathbf{x}\}\) is a diagonal matrix formed by the elements of vector \(\mathbf{x}\) on its main diagonal.

II. FREQUENCY AND PHASE SYNCHRONIZATION IN OPEN-LOOP COHERENT DISTRIBUTED ANTENNA ARRAYS

In open-loop coherent distributed arrays, the nodes synchronize their transceivers in frequency, phase, and time by coordinating with each other (without any feedback from the destination) to perform a shared coherent operation. Each node derives its carrier frequency from its own local oscillator. An oscillator undergoes a random frequency drift over time that offsets its carrier frequency, in the transceiver chain. An oscillator undergoes a random drift and jitter over time that offsets the carrier phase. If the node derives its carrier frequency from its own local oscillator coordinating with each other to form a coherent beam in a joint synchronization.

A. Signal Model With Frequency and Phase Errors

Consider a network of \(N\) nodes spatially separated by multiples of the wavelength of the coherent operation and coordinating with each other to form a coherent beam in a given direction. The signal received in their mutual far-field can be given by

\[
s(t) = C \sum_{n=1}^{N} e^{j(2\pi f_c t + \frac{2\pi}{\lambda} d_n \cos(\theta_n) + \delta \phi_n)},
\]

where \(C\) is a constant amplitude term, \(f_c\) is the carrier frequency of the coherent operation and \(\lambda_c\) is its wavelength, \(d_n\) and \(\theta_n\) are the distance and orientation angle of the \(n\)-th node relative to a reference point or node, and \(\delta \phi_n\) represents the total phase error at the \(n\)-th node. The phase error \(\delta \phi_n\) is defined as \(\delta \phi_n = 2\pi \delta f_n T + 2\pi \varepsilon f T + \theta_n + \delta \theta_n + \varepsilon_n\) in which \(2\pi \delta f_n T\) is the phase error due to the frequency offset \(\delta f_n\) of the oscillator in the \(n\)-th node at the update time \(T\); the term \(2\pi \varepsilon f T\) represents the phase due to frequency estimation error \(\varepsilon f\) at the \(n\)-th node assuming that the nodes need to estimate their oscillator frequency in the update interval to synchronize it with the other nodes; \(\theta_n\) is a constant phase term assumed herein to be uniformly distributed between 0 to \(2\pi\) and accounting for the estimation error in the distance and orientation angle of the \(n\)-th node, any residual phase error due to the timing misalignment, any hardware or channel induced phase delay, and any phase error due to the mismatch of antennas radiation pattern, etc.; \(\delta \theta_n\) represents the phase error due to the time variation of the frequency of the oscillator within the update interval; \(\varepsilon_n\) is the error due to the random phase jitter of the oscillator in the \(n\)-th node; and \(\varepsilon_n\) is the phase estimation error at the \(n\)-th node assuming that the nodes must estimate their oscillator phase in the update interval to synchronize it with the other nodes \([4, 8, 19]\). Note that for an ideal coherent operation, the total phase error \(\delta \phi_n\) must be zero, however, it has been shown in \([4]\) that at least 90% of the ideal coherent gain can be achieved at the destination if this total phase error is below the 18° threshold (see Fig. 4 in \([4]\)).

B. Modeling Frequency and Phase Errors

Herein we discuss the modeling of the errors \(\{\delta f_n, \delta \theta_n, \varepsilon f, \varepsilon_n\}\) for all the nodes, i.e., \(n = 1, 2, \ldots, N\) as follows.

The frequency stability of an oscillator is characterized mainly by its design metrics which includes the manufacturing tolerances of its components and their susceptibility to temperature variations. Allan deviation (ADEV) is a popular metric which is used to quantify the frequency drift \(\delta f_n\) of an oscillator in the \(n\)-th node. It is computed by averaging the fractional frequency error over multiple shifted time intervals and then computing the standard deviation of these measurements. The ADEV value is usually time varying such that in shorter time intervals its value is governed mainly by the noise fluctuations, but as the time passes, these fluctuations equalize resulting in a smaller ADEV and a stable oscillator frequency \([8]\). For a temperature compensated crystal oscillator, the ADEV value usually ranges from \(10^{-7}\) to \(10^{-10}\) at \(T = 1\) sec interval. Following \([19]\), the ADEV can be modeled as

\[
\sigma_f = f_c \sqrt{\frac{\beta_1}{T} + \beta_2 T},
\]

where \(\beta_1\) and \(\beta_2\) depend on the design of an oscillator and \(T\) is the frequency update interval. In this work we set \(\beta_1 = \beta_2 = 5 \times 10^{-19}\) to model a quartz crystal oscillator \([19]\) and model the frequency drift \(\delta f_n\) of an oscillator in the \(n\)-th node as normally distributed with zero mean and standard deviation given by (2) \([19]\).

The time variation of the ADEV metric as discussed above implies that the frequency offset is also time varying. Thus the phase term \(\delta \theta_n\) in (1) denotes the phase adjustment needed at the update time \(T\) by taking into account the
variation of the frequency offset over time. As discussed in [8],
the instantaneous frequency offset can be modeled on average
as varying linearly over the update interval $T$ as a function
$\frac{\delta f_n}{T}$. Hence, the actual phase at time $T$ is calculated as
\[
\delta \theta_{\text{actual}} = 2\pi \int_0^T \left( f_c + \frac{\delta f_n}{T} \right) dt,
\]
and the phase adjustment $\delta \theta_n^f$ at time $T$ is then given by
\[
\delta \theta_n^f = \delta \theta_{\text{actual}} - 2\pi \int_0^T (f_c + \delta f_n) dt,
\]
(4)
The signal generated by an oscillator in practice is also
affected by its phase noise which is a combination of interfer-
ence from the external noise sources that includes noise from
the power supply and bias currents in transistors in the circuit,
and random fluctuation of the phase generated internally in
the oscillator. The phase noise varies randomly in the oscillator
and causes a phase jitter in its generated signal. This phase
jitter can be determined from the phase noise profile of an
oscillator, which can be created by dividing the possible
ranges from
\[
\delta \theta_n^f = \pi \sum_i \delta \phi_i n_i \frac{1}{\pi}.
\]
III. DECENTRALIZED FREQUENCY AND
PHASE SYNCHRONIZATION
We model the network of $N$ nodes in a distributed
antenna array by an undirected graph $G = (V, E)$ where $V = \{1, 2, \ldots, N\}$ represents the set of vertices (nodes)
and $E = \{(i, j) : i, j \in V\}$ is the set of undirected edges
(bidirectional communication links) between the vertices. The
signal generated by the $n$-th node in the network is given by
\[
s_n(t) = e^{j(2\pi f_n t + \theta_n)}
\]
where $f_n$ and $\theta_n$ represent the frequency and
phase of its oscillator, respectively. In the following,
we describe a decentralized frequency and phase consensus
(DFPC) algorithm that iteratively updates the frequency and
phase of each node with the aim to synchronize these para-
eters across the array to the arithmetic averages of the
frequencies and phases of all the nodes in the array. To this
end, we assume that at iteration $k = 0$, each node has an initial
estimate of the frequency and phase of its signal. For the
$n$-th node, the initial frequency is selected as $f_n(0) = f_0 + \mathcal{N}(0, \sigma^2)$
in which $\sigma = 10^{-4} f_c$ represents the crystal clock accuracy
of 100 parts per million (ppm), and the initial phase is assumed
to be uniformly distributed between $0$ to $2\pi$, i.e., $\theta_n(0) \sim \mathcal{U}(0, 2\pi)$. Now let $f(k-1) = [f_1(k-1), f_2(k-1), \ldots, f_N(k-1)]^T$ and $\theta(k-1) = [\theta_1(k-1), \theta_2(k-1), \ldots, \theta_N(k-1)]^T$
represent the combined frequencies and phases of all the nodes
in the $(k-1)$-st iteration of the DFPC algorithm, then in the
$k$-th iteration the algorithm updates the frequencies and phases
of all the nodes by
\[
f(k) = W f(k-1)
\]
\[
\theta(k) = W \theta(k-1),
\]
(8)
in which $W$ represents the $N \times N$ mixing matrix where its
$(i, j)$-th element $w_{i,j}$ denotes the weight corresponding to the
distance between node $i$ and $j$. It is assumed that the matrix
$W$ is symmetric and doubly-stochastic (each row and each
column sums to 1). Furthermore, its element $w_{i,j} = 0$ if nodes
$i$ and $j$ are not connected, i.e., $(i, j) \notin E$. This latter property
enables decentralized (distributed) averaging of the parameters
across the network [44], [45]. In this work, we model $W$ as
a Metropolis-Hastings matrix [19], [46] with $(i, j)$-th element
defined as
\[
w_{i,j} = \begin{cases} 1 & \text{if } (i, j) \in E \\ \frac{\max(n_i, n_j)}{\sum_{i} \frac{1}{n_i}} & \text{if } (i, j) \notin E \text{ and } i \neq j \\ 1 - \sum_{j: j \neq i} w_{i,j} & \text{if } i = j. \end{cases}
\]
where $n_i$ and $n_j$ are the number of edges connected to
node $i$ and $j$ respectively. Note that setting $w_{i,j} = 0$ whenever...
errors are zero. Then the residual frequency error is caused by
error for the DFPC algorithm as follows.

The nodes in the network are declared to have reached a consensus (synchronization in frequency and phase) when the standard deviation of the total phase errors \( \delta \phi_n, s' \) from (1) is less than or equal to some pre-set threshold \( \eta \), i.e.,

\[
\sigma_\phi = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} | \delta \phi_n - \bar{\phi} |^2 \leq \eta},
\]

(10)

where \( \bar{\phi} \) represents the average value of the phase errors.

The proposed DFPC algorithm is described in detail in Algorithm 1 where the frequency and phase errors discussed earlier in Section II-B are also included in the update process.

Algorithm 1 DFPC Algorithm

Input: \( k = 0, W, f(0), \theta(0) \).

/* DFPC run */

while convergence criterion is not met do

\( k = k + 1 \)

1) Define \( f(k-1) = f(k-1) + \delta f \) where frequency drifts are modeled as \( \delta f \sim N(0, \sigma_{\delta f}^2 I_N) \).

2) Set \( \theta(k-1) = \theta(k-1) + \delta \theta \) where phase errors \( \delta \theta \) are computed from (4) and phase jitters are modeled as \( \delta \theta \sim N(0, \sigma_{\delta \theta}^2 I_N) \).

3) Include frequency and phase estimation errors:

\( f(k-1) = f(k-1) + \varepsilon_f \) where \( \varepsilon_f \sim N(0, (\sigma_{\varepsilon_f}^2)^2 I_N) \).

\( \theta(k-1) = \theta(k-1) + \varepsilon_\theta \) where \( \varepsilon_\theta \sim N(0, (\sigma_{\varepsilon_\theta}^2)^2 I_N) \).

4) Run the \( k \)-th iteration of the consensus algorithm as follows.

\[
f(k) = Wf(k-1),
\]

\[
\theta(k) = W\theta(k-1),
\]

end

Output: \( f(k), \theta(k) \)

A. Steady-State Total Phase Error

At the convergence of the proposed DFPC algorithm, the residual frequency and phase errors result from the frequency drifts and phase jitters in the oscillators, and from the errors in estimating the frequencies and phases of their generated signals. Let each \( k \)-th iteration of the proposed algorithm be that of an update interval \( T \). Now if the frequency and phase estimation is performed by observing the signals in this time duration, then in general increasing \( T \) reduces the frequency and phase estimation errors but increases the errors due to the frequency drifts in the oscillators. In this subsection, we theoretically derive the residual steady-state total phase error for the DFPC algorithm as follows.

First let us assume that the frequency and phase estimation errors are zero. Then the residual frequency error is caused by the frequency drift \( \delta f_n \) of the oscillator at the \( n \)-th node. This drift is given by the following standard deviation [19]

\[
\sigma_f = f_c \sqrt{\frac{\beta_1}{T} + \beta_2 T},
\]

(11)

where the above equation combines the frequency deviation in the oscillator due to the white frequency noise which is measured by \( \sigma_{uf} = f_c \sqrt{\frac{\beta_1}{T}} \) and the frequency deviation due to the frequency random walk which is quantified by \( \sigma_{rw} = f_c \sqrt{\beta_2 T} \). The standard deviation of the resulting phase error due to the frequency deviation at time \( T \) is given by

\[
\sigma_\phi = 2 \pi \sigma_f T.
\]

(12)

Now, in practice, the frequency drift of the oscillator varies over time duration \( T \) which causes the phase error \( \delta \theta_n \) in the generated signal as derived in Section II-B. The standard deviation of this phase error be given by

\[
\sigma_\phi = \pi T \sigma_f.
\]

(13)

Furthermore, the phase noise in the oscillator induces a phase jitter in its generated signal which as discussed in Section II-B can be measured by the following standard deviation

\[
\sigma_\phi = \sqrt{2 \times 10^A / 10},
\]

(14)

where as defined earlier \( A \) is the integrated phase noise power computed from the phase noise profile of an oscillator [8].

Next we derive the phase errors resulting from the frequency and phase estimation of the node’s output signal. Let the signal generated by the \( n \)-th node over time duration \( T \) be given by

\[
x_n(t) = e^{j(2\pi f_n t + \theta_n)} + n_o(t),
\]

(15)

where \( f_n \) and \( \theta_n \) represent the frequency and phase of the signal respectively, and \( n_o(t) \) is the transceiver noise which is modeled as a Gaussian process. The variation of the frequency estimation error is lower bounded by the standard deviation [43]

\[
\sigma_f^m \geq \frac{6}{(2\pi)^2 L^3 SNR},
\]

(16)

in which \( L \) denotes the number of samples collected over the time interval \( T \), and SNR is the signal to noise ratio of the received signal. Similarly the variation in the phase estimation error is lower bounded by

\[
\sigma_\phi^m \geq \frac{2}{LSNR}.
\]

(17)

Note that by comparing (16) and (17), it can be seen that the lower bound on the frequency estimation error \( \sigma_f^m \) is larger in value than the bound on the phase estimation error \( \sigma_\phi^m \) by \( \sqrt{(2\pi)^2 L / (2\pi)^2 L^3 SNR} \). This implies that in general accurate frequency estimation is difficult to achieve from the observed signal samples compared to the phase estimation. However, to synchronize the nodes in a distributed phased array system, the collective phase error originated from the combined frequency and phase estimation errors is of importance to ensure that the system operates in a phase-coherent state.
Additionally, the nodes may enable a local broadcast to share their frequencies and phases with the other nodes for synchronization. For a node with \( D \) number of connections on average and synchronized frequencies and phases with its neighbors, the estimation errors usually decrease with the increase in \( D \) due to the increase in SNR. In this case, the lower bound for the frequency estimation is given by

\[
\sigma_f^m \approx \sqrt{\frac{6}{(2\pi)^2 D L^3 \text{SNR}}}, \tag{18}
\]

and the bound for the phase estimation becomes

\[
\sigma_\phi^m \approx \frac{2}{\sqrt{D L \text{SNR}}}, \tag{19}
\]

where the approximation in the above equation is used because \( D \) represents the average number of connections per node in a network. In general, with an increase in the number of connections per node, the resources required to support the wireless connections in a network also must increase proportionally. A solution to cope with the limited resources is to separate the node transmissions using either CDMA or TDMA. While this approach is deemed useful, one major consequence of using TDMA is that the observation time of the signals reduces which in turn reduces the number of samples collected for the frequency and phase estimation. For instance, with \( D \) average number of connections per node, the collected number of samples \( L \) reduces to \( L/D \). Thus, when TDMA is used in the network, the lower bound for the frequency estimation is given by

\[
\sigma_f^m \approx \sqrt{\frac{6D^2}{(2\pi)^2 L^3 \text{SNR}}}, \tag{20}
\]

which in terms of the phase error can be written as \( \sigma_\phi^m = 2\pi f_c \sigma_f^m / T \). Similarly, the lower bound for the phase estimation becomes

\[
\sigma_\phi^m \approx \frac{\sqrt{AD}}{L \text{SNR}}. \tag{21}
\]

Now after one iteration of the proposed DFPC algorithm, the total standard deviation due to frequency errors is \( \sqrt{\sigma_f^2 + \left( \frac{\sigma_f^m}{m} \right)^2} \) and the total standard deviation due to the phase errors is \( \sqrt{(\sigma_\phi^m)^2 + (\sigma_\phi^m)^2 + (\sigma_\phi^0)^2} \). Thus to derive the total phase error after \( I \) iterations, we model the frequency and phase update in the \( k \)-th iteration as

\[
z(k) = W \left( z(k-1) + e_k \right), \tag{22}
\]

in which \( z \in \{ f, \theta \} \) and the error vector \( e_k \) is distributed as \( N \left( 0, \sigma_e^2 I_N \right) \). When \( z = f \), the variance \( \sigma_e^2 \) is given by

\[
\sigma_e^2 = \sigma_f^2 + \left( \frac{\sigma_f^m}{m} \right)^2, \tag{23}
\]

and when \( z = \theta \), the variance of the error is

\[
\sigma_\phi^2 = (\sigma_\phi^m)^2 + (\sigma_\phi^m)^2 + (\sigma_\phi^0)^2. \tag{24}
\]

After \( I \) number of iterations, the above update equation can be written as

\[
z(I) = Wz(I-1) + We_f = W^2z(I-2) + W^2e_{I-1} + We_f = W^I z(0) + \sum_{i=0}^{I-1} W^{I-i} e_{I-i}. \tag{25}
\]

Note that for a symmetric and a doubly stochastic matrix \( W \) as defined in (9), as \( I \to \infty \) then \( W^I \to \frac{1}{N} \). This implies that \( W^I z(0) \) in (23) converges to the average of the initial values, i.e., as \( I \to \infty \), \( W^I z(0) \to \frac{1}{N} z(0) \), which in terms of the phase error can be written as

\[
W^I \theta(0) = \frac{1}{N} \sum_{n=1}^{N} z_n(0) = \left( \frac{1}{N} \right) \theta(0) \]
and integrated in the synchronization process to imitate a more
errors at the nodes as derived in the previous section and
convergence result from the frequency drifts and phase jitters
SNR = 0 residual errors. These residual frequency and phase errors upon
cases converge to an average of their initial values with some
the number of iterations of the DFPC algorithm increases
estimation errors from (6) and (7). It is observed that as
The network is assumed to be sparsely connected with
number of iterations from a single run of the DFPC algorithm.
We consider the case when a network of
performance of the proposed DFPC algorithm through simulations.
B. Simulation Results
In this subsection, we analyze the synchronization perfor-
mance of the proposed DFPC algorithm through simulations.
We consider the case when a network of \( N \) nodes is ran-
domly generated with a known connectivity \( c \). The parameter
\( c \) is defined as the ratio of the number of active edges
in the network to the number of all possible edges given
by \( N(N-1)/2 \). Thus the connectivity \( c \in [0,1] \) where
a smaller value of \( c \) implies a sparsely connected network
and a larger value of \( c \) implies a densely connected network.
Throughout the simulation results included in this paper the
initial frequencies of all the nodes are sampled from the
normal distribution centered on \( f_c = 1 \) GHz and the sampling
frequency is chosen as \( f_s = 10 \) MHz. Since for coherent
distributed arrays, the update interval \( T \) is usually on the order
of ms to sec, we choose \( T = 0.1 \) ms for the results in this
paper unless stated otherwise.

In Figs. 1 and 2 we show the frequency and phase errors
for different number of nodes \( N \) in the network vs. the
number of iterations from a single run of the DFPC algorithm.
The network is assumed to be sparsely connected with \( c = 0.2 \)
and the update interval is set as \( T = 0.1 \) ms. The
SNR = 0 dB is used to compute the frequency and phase
estimation errors from (6) and (7). It is observed that as
the number of iterations of the DFPC algorithm increases
the frequencies and phases of all the nodes in all simulated
cases converge to an average of their initial values with some
residual errors. These residual frequency and phase errors upon
convergence result from the frequency drifts and phase jitters
in the oscillators, and the frequency and phase estimation
errors at the nodes as derived in the previous section and
integrated in the synchronization process to imitate a more
practical scenario. However, it will be shown later that the
variation in the residual total phase error is well below the 18°
threshold which ensures high coherence at the beamforming
destination [4]. From these figures, it is also observed that
the convergence speed of the DFPC algorithm is faster for
larger number of nodes (\( N = 100 \)) vs. smaller number of
nodes (\( N = 20 \)). This is because as \( N \) increases for a given
connectivity \( c \), there are more connections per node on average
given by \( D = c(N-1) \), and thus each node computes a better
local average in the network resulting in a faster convergence.
Therefore, for a given connectivity \( c \), larger number of nodes
in a network makes the system more probable to converge to
the averages of the initial frequency and phase distributions
which is as also visible from these figures. Note that while
we plot the frequency and phase errors relative to the average
values, convergence to a specific frequency and phase is not a
compulsory condition for a coherent operation in distributed
arrays. In fact, consensus at any frequency and phase will
support the coherent operation providing the residual errors
are small as discussed earlier.

Fig. 3 shows the standard deviation of the total phase errors
(\( \delta \phi_n \), for \( n = 1,2, \ldots, N \) as defined in (1) in Section II-A)
of the DFPC algorithm for varying number of nodes \( N \)
in the network when two different SNRs are assumed. The
connectivity in the network is set to \( c = 0.2 \) and the update
interval is \( T = 0.1 \) ms. To generate this figure, we collected
the final converged standard deviation of the total phase errors
from 10^3 independent trials and then computed the average
value and the standard deviation of the samples. In this figure,
the length of the bar determines the value of the standard
deviation whereas its center gives the average value at each
point. It is observed that for each SNR value as the number

residual phase error is when there are no frequency and phase offset errors, the total due to the frequency and phase estimation decreases. Note that improvement in performance because the residual phase errors with different number of nodes when SNR.

In Fig. 3 we compare the convergence speed of DFPC for different number of nodes vs. the connectivity $c$ in the network with different number of nodes when SNR = 0 dB and $T = 0.1$ ms.

In Fig. 4 we compare the convergence speed of DFPC for different number of nodes vs. the connectivity $c$ of the network. To generate this figure, we declare the convergence of DFPC when the standard deviation of the total phase error over the iterations is less than $\eta = 1^\circ$ threshold which is well below the $18^\circ$ threshold needed to achieve at least 90% coherent gain at the destination [4]. In practice, a particular value of the threshold may be application dependent. In this figure, the number of iterations needed for convergence are sampled from $10^3$ independent trials to plot the average value and the standard deviation of the samples. Note that for $N = 5$ and 20 nodes the lowest possible $c$ values that ensure a connected network are 0.4 and 0.11, respectively, and that is why the other values of $c$ are not simulated for these array sizes. It is observed that for all $N$ values, the convergence speed of DFPC improves with the increase in the connectivity $c$ of the network as expected. Furthermore, for each connectivity level $c$ having more nodes in the network result in faster convergence.

In particular, DFPC for $N = 100$ converges faster for a moderately connected networks with $c \in [0.05, 0.8]$. For example, at connectivity $c = 0.05$, DFPC with $N = 100$ nodes requires 16 iterations on average for the convergence whereas when $N = 65$ the convergence happens in 158 iterations. Besides the change in $D$ value with the increase of $c$ or $N$, this improvement in the convergence performance of DFPC with varying $c$ or $N$ can also be attributed to the modulus of the second largest eigenvalue $\lambda_2$ of the mixing matrix $W$. It is shown in Section III-A that $\lambda_2$ controls the total phase error of DFPC. Specifically, $\lambda_2$ is smaller for denser networks and larger for sparser networks, and thus the algorithm converges faster for the former networks than the latter ones.

IV. Kalman Filter Based Decentralized Frequency and Phase Synchronization

In the previous section, we observed that the frequency drifts and phase jitters as well as the frequency and phase estimation errors at the nodes introduce the total phase error upon convergence of the DFPC algorithm that deteriorates the synchronization between the nodes. An increase in SNR reduces the phase error of DFPC, but given the multipath fading channels between the nodes, an improvement in SNR can be achieved by increasing the signal power, which may be limited by the hardware constraints. The Kalman filter is a popular method used for computing the optimal minimum mean square error (MMSE) estimates of the unknown quantities if their state transitioning model follows a first-order Markov process and the observations are a linear function of these quantities [33], [48]. Recently, KF has been used for time synchronization between nodes in [49], [50], and [51] where the oscillator time drifting models are exploited. Thus, in this section, to reduce the residual total phase error at the faster update intervals we propose a Kalman filtering based decentralized consensus algorithm for the frequency and phase synchronization between the nodes in a distributed array. The proposed algorithm is referred to as KF-DFPC and it integrates Kalman filtering with the DFPC algorithm described earlier to improve synchronization between the nodes. Alternatively, the model-free adaptive filtering algorithms such as the diffusion least mean squares algorithm [34], [48], and the KF-based distributed state estimation algorithms in [35], [37], and [36] can be used; however, it is shown through the simulation results later in this section that our proposed KF-DFPC algorithm outperforms and converges faster than these contemporary algorithms at lower SNRs and for the sparsely connected arrays, whereas for the large densely connected array, the use of KF does not increase the computational complexity of DFPC.
In order to use a Kalman filter for frequency and phase estimation at the nodes, we need to define their state transitioning model and the observation model as per its framework [33]. Let at the \( k \)-th time instant, the unknown frequency and phase of the \( n \)-th node be defined as a state vector \( x_n(k) = [f_n(k), \theta_n(k)]^T \). Using the frequency and phase offset models discussed in Section II-B, we can write the state transition model for the \( n \)-th node as

\[
x_n(k) = x_n(k-1) + u_n, \tag{28}
\]

where \( u_n \triangleq [\delta f_n, \delta \theta_n]_T \) and \( u_n \sim \mathcal{N}(0, Q) \) in which the correlation matrix \( Q \) is given by

\[
Q = \mathbb{E}[u_n u_n^T] = \begin{bmatrix}
\sigma_f^2 & -\pi T \sigma_f^2 \\
-\pi T \sigma_f^2 & 2 \pi^2 T^2 \sigma_f^2 + \sigma_\theta^2
\end{bmatrix}. \tag{29}
\]

As seen from Equation (28), the frequency and phase of the oscillator in each node’s transceiver are influenced by random offsets, so their instantaneous values are unknown to the nodes and must be estimated for synchronization. Let the vector \( y_n(k) = [\hat{f}_n(k), \hat{\theta}_n(k)]^T \) define the observation vector that combines the frequency and phase estimates of the \( n \)-th node’s signal obtained from an estimator as discussed in Section II-B, then the observation model for the \( n \)-th node can be written as

\[
y_n(k) = y_n(k) + \nu_n, \tag{30}
\]

where \( \nu_n \triangleq [\varepsilon_f, \varepsilon_\theta]^T \) in which \( \varepsilon_f \) and \( \varepsilon_\theta \) represent the frequency and phase estimation errors respectively. Assuming that the frequency and phase estimation errors are independent, we can model the estimation error vector as \( \nu_n \sim \mathcal{N}(0, \Sigma) \) where the correlation matrix \( \Sigma \) is given by

\[
\Sigma = \mathbb{E}[\nu_n \nu_n^T] = \begin{bmatrix}
\sigma_f^2 m & 0 \\
0 & \sigma_\theta^2 m
\end{bmatrix}. \tag{31}
\]

The parameters \( \sigma_f^m \) and \( \sigma_\theta^m \) represent the standard deviation of the frequency and phase estimation errors respectively. For the purpose of illustrating the synchronization performance of the proposed algorithm, herein both \( \sigma_f^m \) and \( \sigma_\theta^m \) are set equal to the CRLBs given in (6) and (7) respectively. Note that the CRLB can be achieved by an unbiased and efficient estimator that estimates the frequency and phase by collecting a large number of samples over the observation window at the cost of latency.

The KF algorithm is an iterative algorithm that estimates the unknown state vector in each iteration by computing the prediction update step and the time update step. In the prediction update step of the \( k \)-th iteration, it predicts the posterior distribution on the state vector \( x_n(k) \) given the observations up to the time instant \( k-1 \). The posterior distribution is computed as

\[
p(x_n(k) | y_{1:1}^{1:k-1}) = \int \left[ \int \left( p(x_n(k) | x_n(k-1)) \right) p(x_n(k-1) | y_{1:1}^{1:k-1}) \right] dx_n(k-1) \\
\propto \mathcal{N}(m_{n,k-1}(k-1), V_{n,k-1}(k-1)), \tag{32}
\]

where in (32) the state transition distribution is \( p(x_n(k) | x_n(k-1)) = \mathcal{N}(x_n(k-1), Q) \), i.e., it is normally distributed as per the distribution of the offset vector in (28). Assuming that the conditional distribution \( p(x_n(k-1) | y_{n}^{1:k-1}) = \mathcal{N}(m_{n,k-1}(k-1), V_{n,k-1}(k-1)) \), we solve the convolution of the two normal distributions in (32) to get the mean and the covariance of the resulting posterior distribution as

\[
m_{n,k-1}(k) = m_{n,k-1}(k-1) \\
V_{n,k-1}(k) = V_{n,k-1}(k-1) + Q. \tag{33}
\]

Thus, the prediction update step uses the state vector estimate from the previous time instant to produce the estimate of the state vector at the current time instant. In the time update step of the \( k \)-th iteration, the KF algorithm combines the observation from the current time instant to refine this prediction of the state vector and hence obtain a more accurate state vector estimate. To this end, it computes the following posterior distribution

\[
p(x_n(k) | y_{n}^{1:k}) \propto p(x_n(k) | y_{1}^{1:k-1}) p(y_n(k) | x_n(k)) \\
\propto \mathcal{N}(m_{n,k}(k), V_{n,k}(k)), \tag{34}
\]

where in (34) the likelihood function \( p(y_n(k) | x_n(k)) \) is normally distributed as \( \mathcal{N}(x_n(k), \Sigma) \) which is obtained by shifting the distribution of the estimation error vector in (30) by \( x_n(k) \). Inserting (32) in (34), the mean and the covariance of the resulting posterior distribution are computed.
in (35) and (36), shown at the bottom of the page, respectively. This completes the derivation of the Kalman filtering algorithm.

Next we describe the proposed KF-DFPC Algorithm 2 which requires initialization of the KF algorithm in its each iteration. This initialization is explained as follows. To begin, in the \( k = 1 \) iteration of the KF-DFPC algorithm, the prediction update Eqn. (33) of the Kalman filter at the \( n \)-th node can be initialized with the mean and variance of each node’s initial frequency and phase distribution, as stated in Section III, as follows

\[
\mathbf{m}_{n,0}(0) = [f_c, \pi]^T, \quad \mathbf{V}_{n,0}(0) = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 4\pi^2/12 \end{bmatrix},
\]

(37)

Now in the \( k \)-th iteration of KF-DFPC algorithm, let the state vector estimate of the \( n \)-th node after the time update step in (35) can be written as \( \mathbf{m}_{n,k}(k) = [m_{n,k}^f(k), m_{n,k}^\theta(k)]^T \) for all \( n = 1, 2, \ldots, N \). By separating the frequency and phase estimates from all the nodes into vectors \( \mathbf{m}_f(k) \triangleq [m_{1,k}^f, m_{2,k}^f, \ldots, m_{N,k}^f]^T \) and \( \mathbf{m}_\theta(k) \triangleq [m_{1,k}^\theta, m_{2,k}^\theta, \ldots, m_{N,k}^\theta]^T \), the KF-DFPC algorithm updates the frequencies and phases of all the nodes by using

\[
\begin{align*}
\mathbf{f}(k) &= \mathbf{W}_f \mathbf{m}_f(k) \\
\mathbf{\theta}(k) &= \mathbf{W}_\theta \mathbf{m}_\theta(k),
\end{align*}
\]

(38)

where the decentralized mixing matrix \( \mathbf{W} \) is defined in (9). This update implies that in the \( k > 1 \) iterations of the KF-DFPC algorithm, the initialization of Kalman filter at the \( n \)-th node must reflect the local frequency and phase transformation in (38). Since the posterior distribution on the state vector of the \( n \)-th node is a normal distribution, the mean and covariance after a linear transformation can be easily computed \([52]\). Thus the mean vector used for initializing (33) after the linear transformation in (38) and notation adjustment is given by \( \mathbf{m}_{n,k-1}(k-1) = [f_n(k-1), \theta_n(k-1)]^T \) where \( f_n(k-1) \) and \( \theta_n(k-1) \) denote the \( n \)-th element of \( \mathbf{f}(k-1) \) and \( \mathbf{\theta}(k-1) \) vectors, respectively. Similarly, in the \( k > 1 \) iterations, the covariance matrix \( \mathbf{V}_{n,k-1}(k-1) \) in (33) after the linear transformation in (38) and notation adjustment can be given as

\[
\mathbf{V}_{n,k-1}(k-1) = \begin{bmatrix} \mathbf{w}_n^T \mathbf{V}_f \mathbf{w}_n & \mathbf{w}_n^T \mathbf{V}_\theta \mathbf{w}_n \\ \mathbf{w}_n^T \mathbf{V}_\theta \mathbf{w}_n & \mathbf{w}_n^T \mathbf{V}_\theta \mathbf{w}_n \end{bmatrix} \mathbf{V}_{n,k-1}(k-1) \mathbf{w}_n \mathbf{w}_n^T \mathbf{w}_n \mathbf{w}_n^T \mathbf{w}_n \] (39)

where the column vector \( \mathbf{w}_n \) represents the \( n \)-th row of the mixing matrix \( \mathbf{W} \). The diagonal matrix \( \mathbf{V}_f(k-1) = \text{diag}(v_{1,k-1}^f(k-1), v_{2,k-1}^f(k-1), \ldots, v_{N,k-1}^f(k-1)) \) in which each \( v_{n,k-1}^f(k-1) \) for \( n = 1, 2, \ldots, N \) is the (1,1)-th indexed element of the covariance matrix obtained from (36) in the \((k-1)\)-st iteration of KF-DFPC. Similarly, \( \mathbf{V}_\theta(k-1) = \text{diag}(\{v_{1,k-1}^\theta(k-1), v_{2,k-1}^\theta(k-1), \ldots, v_{N,k-1}^\theta(k-1)\}) \) where each \( v_{n,k-1}^\theta(k-1) \) represents the (2,2)-th indexed element of the covariance matrix in (36), and the matrix \( \mathbf{V}_{n,k}(k) = \text{diag}(\{v_{1,k-1}^\theta(k-1), v_{2,k-1}^\theta(k-1), \ldots, v_{N,k-1}^\theta(k-1)\}) \) in which each \( v_{n,k-1}^\theta(k-1) \) denotes the (1,2)-th indexed element of the covariance matrix computed from (36) in the \((k-1)\)-st iteration of the KF-DFPC algorithm. This completes the derivation of the KF-DFPC algorithm which is described in detail in Algorithm 2.

Note that for the notational convenience, we have written (39) using the diagonal matrices \( \mathbf{V}_f(k-1), \mathbf{V}_\theta(k-1) \) and \( \mathbf{V}_{n,k-1}(k-1) \) for the entire array; however, since most of the elements in the weighting vector \( \mathbf{w}_n \) may be zero as per the node’s connectivity, only the diagonal elements corresponding to the connected neighboring nodes are required to evaluate (39) at node \( n \) in the \( k \)-th iteration of our algorithm. To further elaborate, the proposed KF-DFPC algorithm operates in a distributed manner \([37]\), where the nodes run the Kalman filters on their own observations in Step 3(a) and 3(b) in Algorithm 2, in parallel, and then locally broadcast the estimates and their error covariances to their immediate neighboring nodes to update the frequencies and phase across the array in Step 4 and 5. These locally shared values are then used to define the priors for the next iteration of KF using (38) and (39).

The computational complexity of KF-DFPC in each iteration is dominated by equations (8), (35), and (36). Eqn. (8) is part of the KF-DFPC algorithm which has the computational complexity of \( O(\text{card}\{\chi_n\}) \) per node, in which \( \chi_n \) is the set of neighbors of node \( n \) including itself, and the operation \( \text{card}\{\cdot\} \) computes the cardinality of this set. Eqns. (35) and (36) are part of KF which require inverting and then multiplying the \( 2 \times 2 \) matrices and thus has the computational complexity of \( O(8) \). Now since the KFs at all the nodes can be run in parallel, the computational complexity of KF-DFPC for each node per iteration is \( O(\text{card}\{\chi_n\} + 8) \) which is the same as the KCIF and DKF algorithms proposed in \([35]\) and \([36]\). Note that for sparsely connected arrays with \( \text{card}\{\chi_n\} \ll 8 \), the computational complexity of KF-DFPC is \( O(8) \). In contrast, for larger arrays with high connectivity and \( \text{card}\{\chi_n\} \gg 16 \), it becomes \( O(\text{card}\{\chi_n\}) \). Thus, for large, densely connected arrays, the performance improvement by using KF with DFFPC comes at no additional increase in the computational complexity.

### A. Simulation Results

We evaluate the performance of the proposed KF-DFPC algorithm through the simulation results, and compare it to the DFPC algorithm, the diffusion LMS (DLMS) algorithm \([34]\),
Fig. 5. Frequency errors for $N = 100$ nodes in the network vs. the iterations of KF-DFPC for $c = 0.2$, SNR $= 0$ dB, and $T = 0.1$ ms.

Fig. 6. Phase errors for $N = 100$ nodes in the network vs. the iterations of KF-DFPC for $c = 0.2$, SNR $= 0$ dB, and $T = 0.1$ ms.

In Figs. 5 and 6, we show the frequency and phase errors of all the $N = 100$ nodes in the network vs. the number of iteration of KF-DFPC from a single trial when the connectivity between the nodes is $c = 0.2$, SNR $= 0$ dB, and the update interval is set as $T = 0.1$ ms. It is observed that as the number of KF-DFPC iterations increase, both frequency and phase errors converge to the average of their initial values as expected. Furthermore, by comparing these figures with Figs. 1 and 2, we observe that KF-DFPC significantly reduces the residual phase error upon its convergence.

Fig. 7 shows the standard deviation of the total phase errors of the KF-DFPC, DLMS, KCIF, and DKF algorithms upon convergence for different number of nodes $N$ in the network for different connectivity $c$ when SNR $= 0$ dB and $T = 0.1$ ms.

For larger values of $N$ the total phase error reaches an error floor resulting from the frequency and phase offset errors added per iteration. Furthermore, comparing Figs. 3 and 7 for $c = 0.2$ and SNR $= 0$ dB, it is observed that the KF-based algorithms significantly outperforms the DFPC algorithm for all $N$ values. Thus performing an MMSE estimation of the frequencies and phases at the nodes via Kalman filtering and then computing the averages certainly ensures a higher level of synchronization among the nodes.

In Fig. 8 we analyze the convergence speed of the above algorithms when networks with different connectivity $c$ and different number of nodes are considered. In this figure, the convergence is declared when the standard deviation of the total phase error falls below $\eta = 0.1^\circ$ threshold which guarantees high coherence at the destination [4]. The observed
convergence iterations are averaged over $10^3$ trials and we plot the average value and the standard deviation of the samples as before. As expected, it is observed that for all $N$ number of nodes in the network, all algorithms converge faster with the increase in $c$, and for each $c$ having larger number of nodes $N$ results in a faster convergence. The DKF algorithm shows the worst convergence speed for all the $N$ and $c$ values as compared to other algorithms because it only fuses the MMSE estimates from the neighboring nodes in each iteration but does not fuse the error covariance matrices. On the other hand, our proposed KF-DFPC fuses the error covariance matrices as well, as described in (39), and thus converges significantly faster than DKF. For instance, for $c = 0.05$ and $N = 100$, DKF takes 30 iterations on average whereas KF-DFPC takes just 2 iterations. Thus, the synchronization algorithm with faster convergence speed implies that a fewer number of messages needs to be exchanged between the nodes to reach synchronization, which reduces the energy consumption of the nodes and increases the lifetime of the system. This figure shows that our KF-DFPC algorithm also converges faster than the DLMS and KCIF algorithms for sparsely connected arrays. For e.g., for $c = 0.1$ and $N = 65$ nodes, KF-DFPC takes 8 iterations, whereas KCIF takes 12 iterations and DLMS takes 22 iterations. By comparing Figs. 4 and 8, it is observed that the KF-DFPC algorithm converges faster for moderately connected networks with $c \in [0.05, 0.8]$ for all $N$ values than the DFPC algorithm. For instance, for $N = 100$ nodes with network connectivity $c = 0.05$, the KF-DFPC converges in 2 iterations whereas DFPC required 16 iterations, and when $N = 65$ then the KF-DFPC converged in 4 iterations whereas DFPC needed 158 iterations. For higher $c$ values, the required number of iterations for these algorithms are the same, however, a highly connected network can be challenging to implement in practice if the array is large. Thus, given the convergence characteristics and the smaller total phase error, the proposed KF-DFPC algorithm is preferable over the other algorithms for the synchronization in distributed arrays.

Fig. 9 shows the convergence iterations of the KF-DFPC and KCIF algorithm as a function of SNR when the network connectivity is $c = 0.2$ or 0.5. The number of nodes in the network is $N = 100$ and we set the update interval as $T = 0.1$ ms. The threshold on the standard deviation of the total phase error ($\sigma_\phi$) for each $c$ is selected based on the KF-DFPC and KCIF performances shown in Fig. 7. It is observed that for each $c$ value, both algorithms require more iterations to converge at the lower SNR values, and as expected the required convergence iterations decreases with the increase in $c$. The increase in convergence iterations at the lower SNRs is because the Kalman filter used in these algorithms computes the MMSE estimates in each iteration by using the observation up to the present time instant (iteration), and thus with the decrease in SNR more observations (iterations) need to be collected by KF to reduce the estimation error and reach the required total phase error. This implies that the KF-DFPC algorithm can achieve the total phase error irrespective of the SNR in the system. This figure also shows that our KF-DFPC algorithm takes fewer iterations than KCIF at the lower SNRs for the both $c$ values, and while for the sparsely connected array, KCIF shows faster convergence speed at the higher SNRs due to the fusion of the neighboring nodes measurements at each node; nonetheless, as the connectivity between the nodes increases, the exchange of more measurements between the nodes in KCIF slow downs its convergence speed due to the independently varying states of the nodes and its use of a suboptimal Kalman gain [35]. On the other hand, our KF-DFPC algorithm shows significant improvement than KCIF at the higher $c$ values as well.

V. RESIDUAL PHASE ERROR EVALUATION

We compare the synchronization performance of the KF-DFPC and DFPC algorithms by analyzing the total phase error achieved upon convergence when varying the update interval $T$ and the signal to noise ratio SNR of the signals. We assume that the nodes use a local broadcast to share their
frequencies and phases with their neighboring nodes. Each node estimates the frequencies and phases of the signals by sampling them with sampling frequency $f_s = 10 \text{ MHz}$ over the time duration $T$. The frequency drifts and phase jitters in the oscillators are modeled as described earlier in Section II-B. The standard deviation of the total phase errors are averaged over 200 trials in the following figures.

In Fig. 10, we compare the standard deviation of the total phase error of the KF-DFPC and DFPC algorithm as a function of the update interval $T$ for different connectivity $c$ and different number of nodes $N$ in the network when SNR = 0 dB is assumed. It is observed that for $T \in [0.1\text{ms}, 3\text{sec}]$, both DFPC and KF-DFPC algorithms yield total phase error below the 18° threshold for the sparsely connected arrays with $c = 0.021$ and the densely connected ones with $c \geq 0.5$. In particular, for each connectivity $c$, the DFPC algorithm gives a minimum total phase error at $T = 20 \text{ ms}$, but its total phase error increases for the update intervals above and below this value of $T$. This is because for the update intervals above $T = 20 \text{ ms}$, the errors due to the oscillators frequency drift dominate, and for the update intervals below this time duration, the errors due to the frequency and phase estimation are higher. Thus as expected the KF-DFPC algorithm outperforms the DFPC algorithm by large margins when $T \leq 20 \text{ ms}$ for each connectivity $c$ and every number of nodes $N$ in the network. For the purpose of analysis, in this figure we also plot the standard deviation of the phase error $\sigma_{\phi,\text{total}}$ as derived in (27). As explained at the end of Section III-A, for the sparsely connected arrays, for instance, the arrays with $c = 0.021$ as considered in this figure, since the second eigenvalue modulus $\lambda_2$ is close to 1 and $\sum_{m=1}^I \frac{\lambda_2^m}{2^m} \gg 1$, the total phase error upon the convergence of KF-DFPC for $T > 7 \text{ ms}$ and DFPC for all the $T$ values is higher than $\sigma_{\phi,\text{total}}$ bound. For connectivity $c = 0.04$, we have $\sum_{m=1}^I \frac{\lambda_2^m}{2^m} \approx 1$ and thus the total phase error from the DFPC follows the theoretical bound $\sigma_{\phi,\text{total}}$ for all the $T$ values, whereas this holds for the KF-DFPC algorithm only for $T > 20 \text{ ms}$ as expected. As the connectivity $c$ increases further, the second eigenvalue modulus $\lambda_2$ approaches 0 and thus the total phase error from both KF-DFPC and DFPC algorithms decreases below the $\sigma_{\phi,\text{total}}$ bound as seen from this figure.

Finally, in Fig. 11 we show the standard deviation of the total phase error $\sigma_\phi$ of the KF-DFPC and DFPC algorithms when varying the SNR of the signals for the arrays with different connectivity $c$ and different number of nodes $N$. It is assumed that the nodes in the array use TDMA for communication, and thus the frequency and phase estimation errors are modeled using (20) and (21). The update interval is set as $T = 0.1 \text{ ms}$. It is observed that for each connectivity $c$, the total phase error of the DFPC algorithm is higher at lower SNR values but it decreases with the increase in the SNR and reaches the phase error of the KF-DFPC algorithm at the higher SNR values. This performance degradation of DFPC at lower SNRs is because the residual error due to frequency and phase estimation is higher which as a result increases its total phase error upon convergence. This estimation error is also higher for connectivity $c = 0.9$ when $N = 400$ vs. when $N = 100$ due to the decrease in the time duration of sampling. However, higher SNR is difficult to achieve in practice because it requires increasing the transmitted signal power which is limited due to the hardware constraints. Since the KF-DFPC algorithm computes the MMSE estimates of the frequencies and phases in each iteration, it maintains the same total phase error irrespective of the SNR of the signals. Note that this consistent phase error of KF-DFPC is realized at the cost of a few additional convergence iterations at the lower SNR values as shown earlier in Fig. 9.

\section{VI. Conclusion}

A decentralized approach to jointly synchronizing the frequencies and phases of separate nodes in a distributed antenna array was presented. Based on local broadcast of the electrical states of each node, consensus averaging supports convergence to within a residual phase error commensurate with high coherent beamforming gain. We independently modeled the frequency drifts and phase jitters of the oscillators...
and the frequency and phase estimation errors at the nodes using practical statistics. A decentralized algorithm (DFPC) computing a weighted average of the frequencies and phases of its neighboring nodes was analyzed, where the phases and frequencies of the shared signals were modeled with estimation errors. Simulation results showed that the DFPC algorithm synchronizes the nodes up to a non-negligible residual phase error that results from the frequency and phase offset errors at the nodes, but that under certain conditions this residual phase error is below that needed to support high coherent beamforming gain. Furthermore, DFPC takes a large number of iterations to converge for moderately connected arrays with fewer number of nodes, which increases the energy consumption of the array and introduces delay in achieving synchronization. Although its synchronization performance improves at the larger SNRs, in practice the change in SNR is limited by the hardware constraints and is also dependent on multipath fading in communication channels between the nodes. Thus to reduce the residual phase error irrespective of the SNR and improve the synchronization between the nodes, a Kalman filtering based decentralized algorithm (KF-DFPC) was also proposed. The KF-DFPC algorithm reduces the residual phase error by computing the MMSE estimates of the frequencies and phases at the nodes before computing the weighted averages. The synchronization performance of KF-DFPC was compared to the DFPC, DLMS, KCF, and DKF algorithms. Simulation results demonstrate that the KF-DFPC algorithm significantly reduces the residual phase error upon convergence at short update intervals as compared to the DFPC and DLMS algorithms. In addition, under certain conditions, KF-DFPC converges in fewer iterations than all these algorithms, and its synchronization performance is independent of the SNR of the received signals.

Acknowledgment
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research.

References
[1] F. Quitin, M. M. U. Rahman, R. Mudumbai, and U. Madhow, “A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept,” IEEE Trans. Wireless Commun., vol. 12, no. 3, pp. 1418–1428, Mar. 2013.
[2] C. Bocanegra, K. Alemdar, S. Garcia, C. Singhal, and K. R. Chowdhury, “NetBeam: Networked and distributed D-Beamforming for multi-user heterogeneous traffic,” in Proc. IEEE Int. Symp. Dyn. Spec. Access Netw. (DySPAN), Nov. 2019, pp. 1–10.
[3] N. Bhushan et al., “Network densification: The dominant theme for wireless evolution into 5G,” IEEE Commun. Mag., vol. 52, no. 2, pp. 82–89, Feb. 2014.
[4] J. A. Nanzer, R. L. Schmid, T. M. Comberiate, and J. E. Hodkin, “Open-loop coherent distributed arrays,” IEEE Trans. Microw. Theory Techn., vol. 65, no. 5, pp. 1662–1672, May 2017.
[5] S. M. Ellison and J. A. Nanzer, “High-accuracy multinode ranging for coherent distributed antenna arrays,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 5, pp. 4056–4066, Oct. 2020.
[6] S. M. Ellison, S. Mghabghab, J. J. Doroshewitz, and J. A. Nanzer, “Combined wireless ranging and frequency transfer for intermode coordination in open-loop coherent distributed antenna arrays,” IEEE Trans. Microw. Theory Techn., vol. 68, no. 1, pp. 277–287, Jan. 2020.
[7] 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.
[8] 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.
[9] I.-C. Chao, K. B. Lee, R. Candell, F. Proctor, C.-C. Shen, and S.-Y. Lin, “Software-defined radio based measurement platform for wireless networks,” in Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun. (ISPCS), Oct. 2015, pp. 7–12.
[10] K.-Y. Tu and C.-S. Liao, “Application of ANFS for frequency synchronization using GPS carrier-phase measurements,” in Proc. IEEE Int. Freq. Control Symp. 21st Eur. Freq. Time Forum, May 2007, pp. 933–936.
[11] M. Pónton and A. Suárez, “Stability analysis of wireless coupled-oscillator circuits,” in IEEE MIT-S Int. Microw. Symp. Dig., Jun. 2017, pp. 83–86.
[12] M. Pónton, 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.
[13] X. Yang, X. Lu, and A. Babakhani, “Picosecond wireless synchronization using an optically locked voltage controlled oscillator (OL-VCO),” in IEEE MIT-S Int. Microw. Symp. Dig., Jun. 2014, pp. 1–4.
[14] R. Mudumbai, B. Wild, U. Madhow, and K. Ramch, “Distributed beamforming using 1 bit feedback: From concept to realization,” in Proc. Allerton Conf. Commun., Control, Comput., 2006, pp. 1020–1027.
[15] 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.
[16] M. Seo, M. Rodwell, and U. Madhow, “A feedback-based distributed phased array technique and its application to 60-GHz wireless sensor network,” in IEEE MIT-S Int. Microw. Symp. Dig., Jun. 2008, pp. 683–686.
[17] F. Quitin, M. M. U. Rahman, R. Mudumbai, and U. Madhow, “A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept,” IEEE Trans. Wireless Commun., vol. 12, no. 3, pp. 1418–1428, Mar. 2013.
[18] 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., Nov. 2016, pp. 442–447.
[19] 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.
[20] 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.
[21] R. L. Schmid, T. M. Comberiate, J. E. Hodkin, and J. A. Nanzer, “A distributed RF transmitter using one-way wireless clock transfer,” IEEE Microw. Wireless Compon. Lett., vol. 27, no. 2, pp. 195–197, Feb. 2017.
[22] 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.
[23] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, “A tutorial on synthetic aperture radar,” IEEE Geosci. Remote Sens. Mag., vol. 1, no. 1, pp. 6–43, Apr. 2013.
[24] Y. Fan, X. Chen, Y. Wei, Y. Wen, T. Zeng, and Z. Ding, “The distributed SAR reconstruction method for line target,” in Proc. IEEE Int. Conf. Signal Inf. Data Process. (ICSIDP), Dec. 2019, pp. 1–6.
[25] R. L. Schmid, S. M. Ellison, T. M. Comberiate, J. E. Hodkin, and J. A. Nanzer, “Microwave wireless coordination technologies for coherent distributed maritime radar,” in IEEE MIT-S Int. Microw. Symp. Dig., Jun. 2017, pp. 884–887.
[26] S. Mghabghab, H. Ouassal, and J. A. Nanzer, “Wireless frequency synchronization for coherent distributed antenna arrays,” in Proc. IEEE Int. Symp. Antennas Propag. USNC-URSI Radio Sci. Meeting, Jul. 2019, pp. 1575–1576.
[27] O. Abari, H. Rahul, D. Katabi, and M. Pant, “AirShare: Distributed coherent transmission made seamless,” in Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), Apr. 2015, pp. 1742–1750.
[28] S. R. Mghabghab, S. M. Ellison, and J. A. Nanzer, “Open-loop distributed beamforming using wireless phase and frequency synchronization,” IEEE Microw. Wireless Compon. Lett., vol. 52, no. 3, pp. 234–237, Mar. 2022.
[29] E. Mallada and A. Tang, “Distributed clock synchronization: Joint frequency and phase consensus,” in Proc. IEEE Conf. Decis. Control Eur. Control Conf., Dec. 2011, pp. 6742–6747.

[30] E. Mallada, R. A. Freeman, and A. K. Tang, “Distributed synchronization of heterogeneous oscillators on networks with arbitrary topology,” IEEE Trans. Control Netw. Syst., vol. 3, no. 1, pp. 12–23, Mar. 2016.

[31] 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, Jul. 2017, pp. 115–116.

[32] C. Urrea and R. Agramonte, “Kalman filter: Historical overview and review of its use in robotics 60 years after its creation,” J. Sensors, vol. 2022, pp. 1–21, Sep. 2021.

[33] B. Anderson and J. Moore, Optimal Filtering. Upper Saddle River, NJ, USA: Prentice-Hall, 1979.

[34] 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.

[35] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in Proc. 48th IEEE Conf. Decis. Control (CDC), Dec. 2009, pp. 7036–7042.

[36] 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.

[37] 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. II, Expr. Briefs, vol. 69, no. 4, pp. 2371–2375, Apr. 2022.

[38] P. Chatterjee and J. A. Nanzer, “Effects of time alignment errors in coherent distributed radar,” in Proc. IEEE Radar Conf. (RadarConf), Apr. 2018, pp. 0727–0731.

[39] D. B. Leeson, “A simple model of feedback oscillator noise spectrum,” Proc. IEEE, vol. 54, no. 2, pp. 329–330, Feb. 1966.

[40] T. H. Lee and A. Hajimiri, “Oscillator phase noise: A tutorial,” IEEE J. Solid-State Circuits, vol. 35, no. 3, pp. 326–336, Mar. 2000.

[41] Y. Liao, “Phase and frequency estimation: High-accuracy and low-complexity techniques,” M.S. thesis, Worcester Polytech. Inst., Worcester, MA, USA, 2011.

[42] S. Zhou and L. Rongfang, “Efficient and accurate frequency estimator under low SNR by phase unwrapping,” Math. Problems Eng., vol. 2019, pp. 1–6, Apr. 2019.

[43] M. A. Richards, Fundamentals of Radar Signal Processing. New York, NY, USA: McGraw-Hill, 2005.

[44] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” Syst. Control Lett., vol. 53, no. 1, pp. 65–78, 2004.

[45] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: An exact first-order algorithm for decentralized consensus optimization,” J. Optim., vol. 25, no. 2, pp. 944–966, May 2015.

[46] L. Xiao, S. Boyd, and S.-J. Kim, “Distributed average consensus with least-mean-square deviation,” J. Parallel Distrib. Comput., vol. 67, no. 1, pp. 33–46, Jan. 2007.

[47] S. Boyd, P. Diaconis, and L. Xiao, “Fastest mixing Markov chain on a graph,” SIAM Rev., vol. 46, no. 4, pp. 667–689, 2003.

[48] R. Claser and V. H. Nascimento, “On the tracking performance of adaptive filters and their combinations,” IEEE Trans. Signal Process., vol. 69, pp. 3104–3116, 2021.

[49] G. Giorgi, “An event-based Kalman filter for clock synchronization,” IEEE Trans. Instrum. Meas., vol. 64, no. 2, pp. 449–457, Feb. 2015.

[50] P. Li, H. Gong, J. Peng, and X. Zhu, “Time synchronization of white rabbit network based on Kalman filter,” in Proc. 3rd Int. Conf. Electron. Inf. Technol. Comput. Eng. (EITC), Oct. 2019, pp. 572–576.

[51] W. Jinyu and Z. Lili, “Research on clock synchronization algorithm based on Kalman filter,” in Proc. Int. Conf. Big Data Anal. Comput. Sci. (BDACS), Jun. 2021, pp. 162–165.

[52] J. A. Gubner, Probability and Random Processes for Electrical and Computer Engineers. Cambridge, U.K.: Cambridge Univ. Press, 2006.