

# Decentralized Frequency Alignment for Collaborative Beamforming in Distributed Phased Arrays

Hassna Ouassal, Ming Yan, and Jeffrey A. Nanzer, *Senior Member, IEEE*

**Abstract**—A new approach to distributed syntonization (frequency alignment) for the coordination of nodes in open loop coherent distributed antenna arrays to enable distributed beamforming is presented. This approach makes use of the concept of consensus optimization among nodes without requiring a centralized control. Decentralized frequency consensus can be achieved through iterative frequency exchange among nodes. We derive a model of the signal received from a coherent distributed array and analyze the effects on beamforming of phase errors induced by oscillator frequency drift. We introduce and discuss the average consensus protocol for frequency transfer in undirected networks where each node transmits and receives frequency information from other nodes. We analyze the following cases: 1) undirected networks with a static topology; 2) undirected networks with dynamic topology, where connections between nodes are made and lost dynamically; and 3) undirected networks with oscillator frequency drift. We show that all the nodes in a given network achieve average consensus and the number of iterations needed to achieve consensus can be minimized for a given cluster of nodes. Numerical simulations demonstrate that the consensus algorithm enables tolerable errors to obtain high coherent gain of greater than 90% of the ideal gain in an error-free distributed phased array.

**Index Terms**—Syntonization, distributed arrays, distributed beamforming, collaborative beamforming, consensus averaging.

## I. INTRODUCTION

COHERENT distributed arrays (CDAs) are clusters of cooperating wireless nodes whose radiated signals add up constructively at a designated destination. For instance, distributed beamforming [1], [2], virtual antenna arrays [3], distributed MIMO arrays [4] are all implicit applications of CDAs. These distributed schemes have been proven to be technically feasible and bring improvements on three fronts:

- 1) *Enhanced energy efficiency*: A transmit cluster focuses the emitted energy spatially toward the intended direction, yielding high directivity and thus better signal-to-noise ratio at the destination.

This material is based upon work supported The Defense Advanced Research Projects Agency (grant number N66001-17-1-4045) and by the National Science Foundation (grant number DMS-1621798). The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

H. Ouassal and J. Nanzer are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (email: nanzer@msu.edu).

M. Yan is with the Department of Computational Mathematics, Science and Engineering and the Department of Mathematics, Michigan State University, East Lansing, MI 48824 (email: myan@msu.edu).

- 2) *Increased reliability*: Having many transmitters allows for more signal path diversity and helps mitigate the multipath fading and shadowing.
- 3) *Increased range*: Aggregate transmit power growth proportional to the square (and cube) of  $N$  transmitters (i.e., radars) enables longer communication range.

The major technical challenge in realizing these benefits consists of synchronizing the radio frequency (RF) signals on the array transmitters, so that they are coherently combined at the destination. This implies synchronization of the frequency and phase of these signals. In a distributed array, each RF signal is generated from an independent local oscillator. These oscillators have intrinsic frequency offsets and undergo stochastic frequency drifts over time (i.e., because of manufacturing tolerances and temperature variations) inducing phase offset variations. Therefore, it is necessary to compensate these clock offsets to frequency lock the transmitters and then implement phase calibration and channel estimation to achieve coherent transmission. CDAs generally require precise alignment of the phases of the transmitted signals to ensure energy-efficient transmissions to the destination [2], with total phase errors of less than  $18^\circ$  required to achieve 90% ideal coherent gain, or a degradation of less than 0.5 dB from the ideal case [1].

Many techniques have been developed to achieve distributed transmitter synchronization, including receiver-coordinated explicit-feedback [5], one-bit feedback [6], master-slave synchronization [2], reciprocity (i.e., channel state estimation by a transmit cluster from signals emitted by the target receiver) [7], [8], round-trip synchronization [9], and two-way synchronization [10]. Most of these techniques use explicit/implicit feedback from the destination, and are thus *closed loop CDAs* [1]. These feed-back schemes undergo latency and overhead, and most importantly are unsuitable for remote sensing and radar applications, where feedback is generally not available.

In [1], the *open loop CDA* (OL-CDA) concept was introduced, where the array self-aligns without the use of feedback from the destination. Although this array uses zero feedback, it does involve inter-node coordination and phase alignment to achieve coherent beam steering. Indeed, it operates as distributed phased array, but the array geometry and element locations must be known. OL-CDA enables the development of future (fixed and mobile) wireless communication networks, which are energy efficient, secure, and can be built with low synchronization overheads. While closed loop CDA approaches have been heavily studied, the OL-CDA

concept is still in its infancy, and more effort needs to be devoted to explore its potential. Recent theoretical studies and experiments have begun demonstrating the feasibility and potential applications of OL-CDAs (e.g. [11]–[13]). OL-CDAs are inherently sparse antenna arrays, and while there has been considerable work done in the design and analysis of the radiation patterns of distributed antenna arrays in terms of the antenna locations and sidelobe levels (e.g. [14]–[21]), there has not been a focused effort on the design of inter-node frequency alignment approaches and the effects on distributed beamforming of frequency mismatch in the nodes in distributed antenna arrays in a decentralized sense.

A well-known frequency synchronization technique is the aforementioned master-slave architecture, where slave transmitters lock to a reference signal broadcast by the master transmitter. Despite its simplicity, this approach has the following limitations: (1) the technique is fragile to failures of the master node (if the master node fails, then the whole RF synchronization fails); (2) the architecture requires a dedicated master node, and thus limits the flexibility of the array. While GPS has been considered in some works as a master node, GPS is only adequate for coherent distributed transmissions at very low frequencies [22] ( $\sim 10$  MHz); (3) the master-slave architecture is not scalable.

This paper presents a new approach to distributed frequency alignment (syntonization) using decentralized (distributed) consensus averaging. Consensus, i.e., group agreement, has been considered in contexts such as flocking [23], formation control [24], flight of unmanned air vehicles (UAVs), and clusters of satellites. The proposed consensus scheme involves iterative frequency exchange (e.g., via local broadcast) among cooperating nodes to reach (asymptotically) a common agreed frequency, i.e., the average of all nodes' initial frequencies. In other words, the frequency of each node converges to the global average. Note that unlike centralized master-slave schemes, the distributed scheme is robust to node failures and is scalable for large networks.

This paper is organized as follows. Section II presents an analysis of the effect on coherent array gain of the phase error resulting from frequency misalignment. Section III models the network and introduces a distributed approach for frequency synchronization. For a static network topology, numerical results showing the effectiveness of the proposed algorithm are presented. Then Section IV employs the algorithm to deal with dynamic networks, where the connectivity of the network changes and the frequencies change due to oscillator drift. Numerical results show that a consensus can be reached in the presence of oscillator drift on low-cost oscillators to achieve above 90% of the ideal level of coherent gain.

## II. SYNTONIZATION (FREQUENCY ALIGNMENT) IN COHERENT DISTRIBUTED ARRAYS

This work focuses on OL-CDAs, in which individual nodes self-align to achieve distributed coherence. As noted earlier, open-loop operation necessitates more information than closed-loop systems (where feedback from the destination is available). There are a few basic drivers of coordination

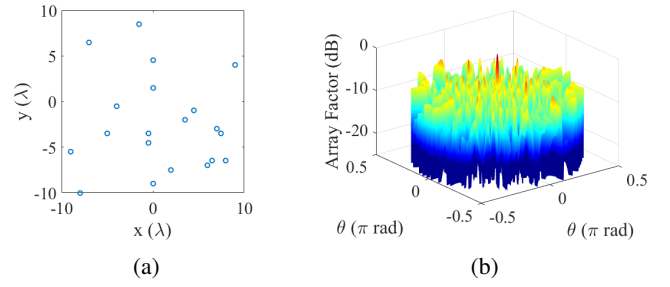


Fig. 1: (a) Sparse array layout for  $N = 20$  elements and (b) broadside radiation pattern.

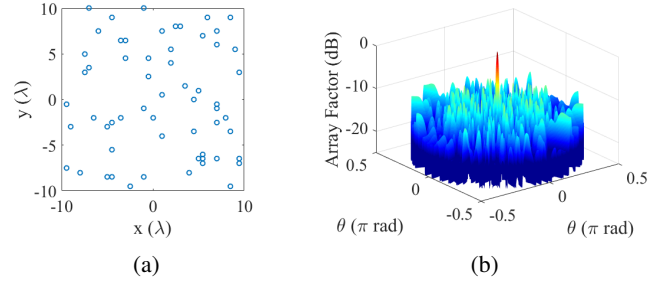


Fig. 2: (a) Sparse array layout for  $N = 60$  elements and (b) broadside radiation pattern.

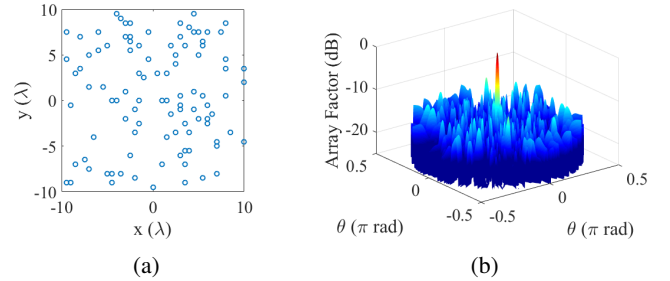


Fig. 3: (a) Sparse array layout for  $N = 100$  elements and (b) broadside radiation pattern.

errors in open-loop distributed arrays, which are errors in the measurement of the distances between node pairs; in aligning the phases of the nodes; in aligning the timing of the nodes; and in aligning the frequencies of the nodes. In previous work, we analyzed the effects on achieving coherent gain of errors in distance, phase, and time [1], [25]. In this work, we focus on the effects of errors in frequency alignment and present an approach for decentralized consensus optimization of the frequency. As such it is assumed that the errors due to distance, relative phasing, and timing are accounted for and thus negligible; because such errors are generally independent, their contributions can be analyzed individually and combined for a total error budget.

In a coherent distributed transmit operation, the set of nodes in the array transmit waveforms with the appropriate relative phasing and timing such that the transmitted waveforms arrive at the target in-phase, ensuring that the waveforms add constructively at the destination node. We consider sparse planar arrays of transmitters, placed randomly within a  $10\lambda \times 10\lambda$  domain on a  $\lambda/2$  grid. While the application space is general,

notional use cases are communications from a set of compact radios (e.g. [2]) or remote sensing from an array of UAVs (e.g. [26]–[29]); operating at a frequency of 100 MHz, the domain represents a 30m×30m space, within which it is certainly feasible to locate nodes within half of a wavelength (1.5m). Figs. 1-3 show random array layouts for arrays with 20, 60, and 100 nodes, and the corresponding array radiation patterns. The randomness in the array layout ensures that no significant sidelobes are present. When implementing frequency alignment between nodes, it is desired that the mainbeam gain remains as high as possible in the presence of errors. In the following we assess the degradation of the mainbeam gain in the presence of phase errors due to frequency misalignment.

#### A. Distributed Array Model

To model the relative effects of errors in a coherent distributed radar operation, we assess the relative coherent summation of the signals that is achieved in the presence of errors and determine the resulting signal gain at a destination point relative to an ideal summation of signals without errors. We consider the array to comprise vertically polarized small dipoles (antenna element factors could be included later in a straightforward manner), whose current density is

$$\mathbf{J} = \hat{\mathbf{z}} e^{j\omega t} \sum_n I_n l_n \delta(x - dx_n) \delta(y - dy_n) \delta(z - dz_n), \quad (1)$$

where  $I_n$  is the current on element  $n$ ,  $l_n$  is the length of the  $n^{\text{th}}$  dipole, and  $(dx_n, dy_n, dz_n)$  is the displacement from the origin. The electric field intensity is found in terms of the far-field magnetic vector potential

$$\mathbf{A} \approx \frac{e^{-jkr}}{4\pi r} \tilde{\mathbf{J}}, \quad (2)$$

through

$$\mathbf{E} = -j\omega\mu\mathbf{A} + \frac{1}{j\omega\epsilon} \nabla (\nabla \cdot \mathbf{A}), \quad (3)$$

where

$$\tilde{\mathbf{J}} = \hat{\mathbf{z}} e^{j\omega t} \sum_n I_n l_n e^{-j(k_x dx_n + k_y dy_n + k_z dz_n)} \quad (4)$$

is the Fourier transform of the current density.

Beamsteering in a distributed array requires relative phase coordination between all elements in the array. Separate from aligning the frequency of the elements, phase coordination is what enables transmitted signals to arrive in-phase at the desired destination. The beamsteering phase of each element can be implemented in relative to a global array origin, or relative to other nodes in the array. When implementing a beamsteering phase to one element relative to the reference point, the wavenumber vector can be given by [1]

$$\mathbf{k}_n = \hat{\mathbf{z}}_n \frac{2\pi}{\lambda} d_n \cos\theta_n \quad (5)$$

and the far-field electric field intensity is thus

$$\begin{aligned} E_\theta &= jk\eta \sin\theta \sum_{n=1}^N \frac{I_n l_n}{4\pi r_n} e^{j\omega t} e^{-j\frac{2\pi}{\lambda} d_n \cos\theta_n} \\ &= \sum_{n=1}^N h_n \alpha_n e^{j(\omega t - \frac{2\pi}{\lambda} d_n \cos\theta_n)}, \end{aligned} \quad (6)$$

where  $N$  is the number of elements in the array,  $h_n$  contains the amplitude terms relating to the channel propagation effects and constant scalars, and  $\alpha_n$  accounts for the amplitude of the current elements. The signal  $s(t)$  (V) received at a point in space is proportional to the electric field intensity  $E$  (V/m). If the receiving point is far-field to the array, and the array emits continuous-wave signals such that relative timing effects can be neglected, the total ideal signal can be written

$$s_i(t) = C \sum_{n=1}^N e^{j(2\pi f t - \frac{2\pi}{\lambda} d_n \cos\theta_n)}. \quad (7)$$

If errors are present in the coordination, the total received signal is given by

$$s_r(t) = C \sum_{n=1}^N e^{j[2\pi(f + \delta f_n)(t - \delta t_n) + \phi_n + \phi_{s,n}]}, \quad (8)$$

where  $\delta t_n$  is the time alignment error,  $\delta f_n$  is the frequency error,  $\phi_{s,n}$  is the phase added for beamsteering (from (5)), and  $\phi_n$  is the phase error, given by

$$\phi_n = \frac{2\pi}{\lambda} (d_n + \delta d_n) \cos(\theta_n + \delta\theta_n) + \phi_c, \quad (9)$$

where  $\delta d_n$  is the error in the antenna separation measurement,  $\delta\theta_n$  is the error in estimating the desired beamsteering angle relative to the platform attitude, and  $\phi_c$  is the relative phase error of the oscillator.

#### B. Signal Model with Frequency Errors

While it is ultimately the differences in phase that cause degradation of the coherent gain, in this work the focus is on aligning the frequencies of the systems, or syntonization, between the separate platforms. In a system where continuous syntonization can be implemented, phase-locked loops (PLLs) can be used. In theory, the phase error of a PLL is zero, thus with continuous syntonization it can be expected that the phase error resulting from frequency differences will be negligible. In practice, however, syntonization may have to be implemented periodically, due to the nature of information transfer between the nodes. In this case, the frequencies of the oscillators on each node will drift between updates. The total phase error due to the evolution of the frequency drift in a time  $T$  is given by

$$\delta\phi_n = 2\pi\delta f_n T. \quad (10)$$

With other error terms corrected, the received signal in the presence of frequency drift is thus

$$s_r(t) = C \sum_{n=1}^N e^{j(2\pi f t + \delta\phi_n + \phi_{s,n})}. \quad (11)$$

If the frequency error is constant, the total phase error will be the frequency error multiplied by the time interval. In practice, the frequency will drift from zero error to some maximum value, thus estimating the frequency error as a constant represents an upper bound on the total error.

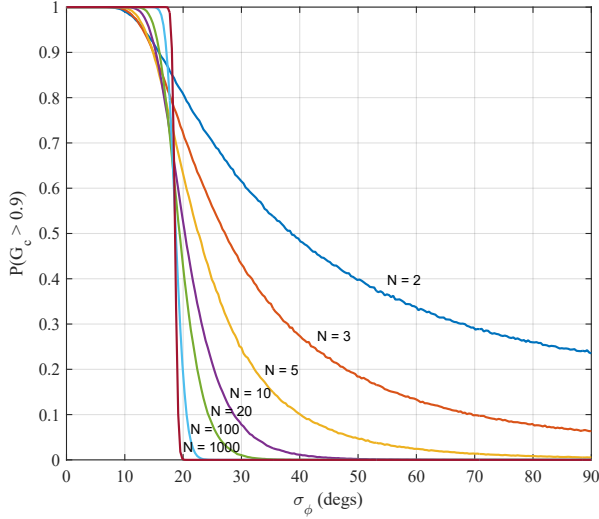


Fig. 4: Probability of the coherent gain exceeding 0.9 versus clock phase error (100,000 Monte Carlo simulations). The threshold for achieving  $P(G_c \geq 0.9) = 1$  approaches  $18^\circ$ .

### C. Inter-Node Phase Error Tolerance Resulting From Frequency Drift

To evaluate the effect of the error terms, the coherent gain  $G_c$ , which is the received signal power of (11) relative to (7), and is given by

$$G_c = \frac{|s_r s_r^*|}{|s_i s_i^*|}, \quad (12)$$

is evaluated. The standard deviations of the frequency-induced phase error is varied, and the probability that the signal power exceeds a given threshold

$$P(G_c \geq X), \quad (13)$$

where  $0 \leq X \leq 1$  is the fraction of the ideal coherent signal power gain, is determined through Monte Carlo simulations. In this work, we evaluate the probability that the received signal will exceed 0.9 of the ideal gain ( $X = 0.9$ ), corresponding to a coherent gain degradation of 0.5 dB.

In Fig. 4, the probability of the received signal power exceeding 0.9 is shown for errors in the oscillator phase  $\sigma_\phi$  for coherent distributed arrays consisting of  $N = 2, 3, 5, 10, 20, 100$ , and 1000 nodes. As the number of nodes in the system increases, the line marking the area where  $P(G_c \geq 0.9)$  becomes sharper, converging to a point below which  $P(G_c \geq 0.9)$  is approximately 1, and above which is zero. The area where  $P(G_c \geq 0.9) \approx 1$  also increases as the number of platforms increases. This is due to the decreasing variance of the received signal power as the number of elements in the network increases. As  $N \rightarrow \infty$ , this cutoff error below which  $P(G_c \geq 0.9) \approx 1$  approaches  $18^\circ$ .

The effect of phase errors on the array beampattern is shown in Figs. 5-7. For relatively small arrays ( $N = 20$ ), the main-beam degradation relative to the sidelobe levels is significant when the phase error is above  $36^\circ$ , and the mainbeam gain is in the order of the sidelobe gain. At lower errors, the mainbeam increases, and at  $18^\circ$  the mainbeam gain is degraded by only

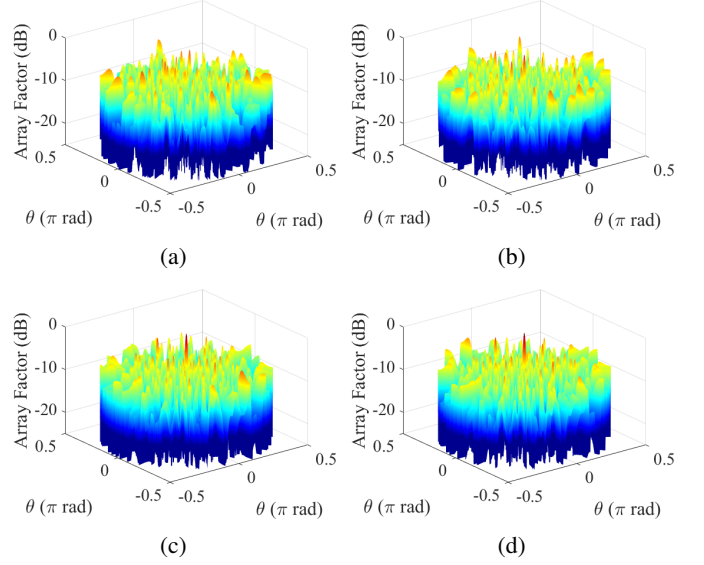


Fig. 5: Broadside beamforming for the  $N = 20$  element array with phase standard deviations of (a)  $180^\circ$ , (b)  $90^\circ$ , (c)  $36^\circ$ , (d)  $18^\circ$ .

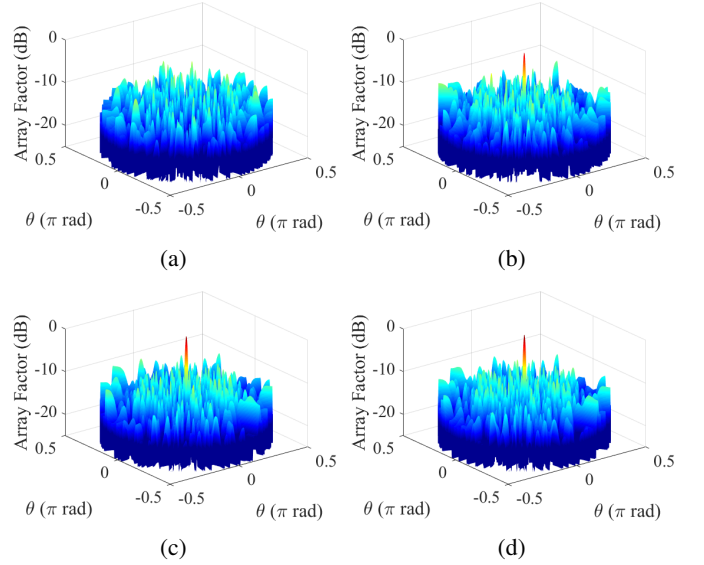


Fig. 6: Broadside beamforming for a  $N = 60$  element array with phase standard deviations of (a)  $180^\circ$ , (b)  $90^\circ$ , (c)  $36^\circ$ , (d)  $18^\circ$ .

0.5 dB. The sidelobe levels are low for larger arrays of  $N = 60$  and  $N = 100$ , thus the beam can still be formed with errors up to  $90^\circ$ , albeit with significant gain degradation.

## III. DECENTRALIZED FREQUENCY ALIGNMENT

### A. System Model

The distributed antenna array is modeled as a network (connected undirected graph)  $G = \{\mathcal{N}, \mathcal{E}\}$  consisting of a set of nodes  $\mathcal{N} = \{1, \dots, n\}$  and a set of undirected edges  $\mathcal{E}$ . An edge  $(i, j) \in \mathcal{E}$  means that there is a connection between nodes  $i$  and  $j$  and the communication is bidirectional. We assume that the nodes are equipped with hardware clocks subject to clock drift.



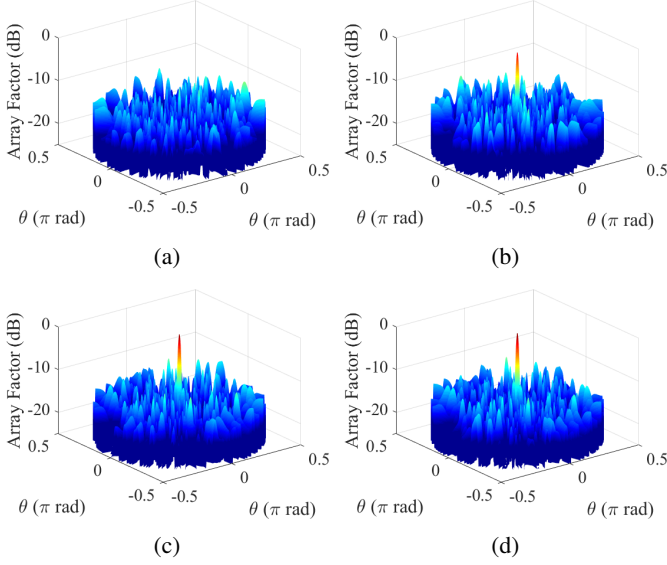


Fig. 7: Broadside beamforming for the  $N = 100$  element array with phase standard deviations of (a)  $180^\circ$ , (b)  $90^\circ$ , (c)  $36^\circ$ , (d)  $18^\circ$ .

Each node  $i$  has an associated RF signal  $x_i(t) = \cos(2\pi f_i t)$ , the frequency of which the network must reach a consensus. It has an initial estimate for the true frequency  $f_c$ , which is denoted as  $f_i(0)$ . Then decentralized averaging algorithms update the estimates at all nodes during each iteration. We say that the nodes reach a consensus if

$$f_i = \bar{f} = \frac{1}{n} \sum_{i=1}^n f_i(0), \quad \forall i = 1, \dots, n. \quad (14)$$

We assume that  $f_i(0) = f_c + \sigma_i X$  where  $\sigma_i = 1 \times 10^{-4} f_c$  is the error in frequency:  $1 \times 10^{-4}$  i.e., 100 ppm (parts per millions) represents crystal clock accuracy, and  $X$  is a random variable with normal distribution  $N(0, 1)$ .

### B. Average Consensus Model

The proposed decentralized frequency alignment consensus (DFAC) scheme with a fixed graph is described in Algorithm 1. We denote  $\mathbf{f}(k) = \{f_1(k), \dots, f_n(k)\}$  as the estimates from all nodes at iteration  $k$ .

#### Algorithm 1 Decentralized frequency alignment consensus

---

**Input:**  $k = 0$ , the initial estimates  $\mathbf{f}(0)$ , mixing matrix  $\mathbf{W}$   
**while** stopping criteria is not satisfied **do**  
     $k = k + 1$   
     $\mathbf{f}(k) = \mathbf{W}\mathbf{f}(k-1)$   
**end while**  
**return**  $\mathbf{f}(k)$

---

The mixing matrix  $\mathbf{W} = [w_{ij}]$ , where  $w_{ij}$  is the  $(i, j)$  element of the matrix, satisfies the following assumptions:

- (Symmetry)  $\mathbf{W} = \mathbf{W}^\top$ .
- (Doubly stochastic)  $\mathbf{W}\mathbf{1} = \mathbf{1}$ ,  $\mathbf{1}^\top \mathbf{W} = \mathbf{1}^\top$ , where  $\mathbf{1}$  is the  $n \times 1$  vector of all ones. That is, each row sums to 1 and each column sums to 1.

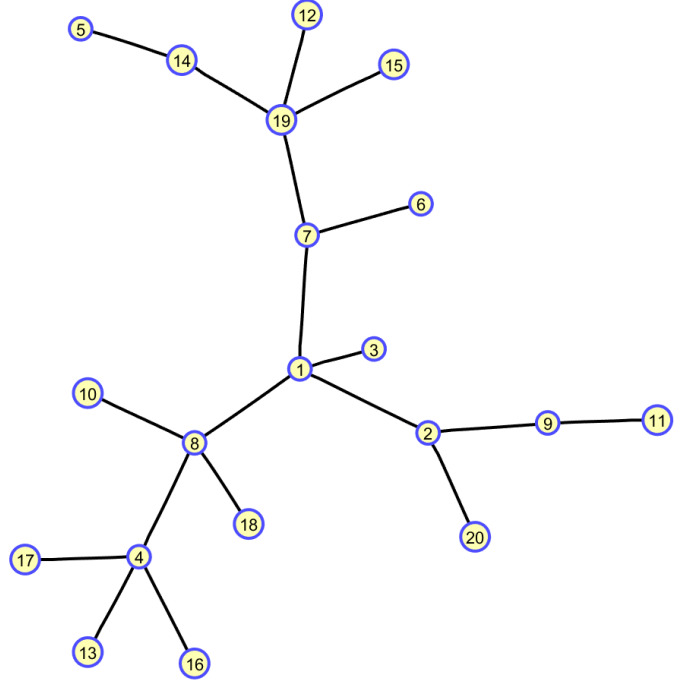


Fig. 8: A sparsely connected network of 20 nodes and  $r = 0.1$ .

- (Decentralized property)  $w_{ij} = 0$  if  $i \neq j$  and nodes  $i$  and  $j$  are not connected, i.e.,  $(i, j) \notin \mathcal{E}$ .

### C. Numerical Results

In this section, we evaluate the performance of the proposed DFAC algorithm in a simulated environment with a fixed network. The algorithm is implemented in MATLAB; the initial states of the nodes are selected at  $f_c = 1$  GHz. We evaluate the quantitative performance of the decentralized algorithm by computing the normalized error  $(\mathbf{f} - f_c)/(f_c)$  along the iteration number. We vary the number of nodes  $N$  and the network connectivity.

1) *Varying number of nodes:* The network is randomly generated with connectivity ratio  $r$ , where  $r$  is defined as the number of edges divided by the total number of all possible edges  $N(N-1)/2$ . When  $r$  is small, the network presents a cluster of sparsely connected nodes. As  $r$  increases, the network becomes more and more connected. Fig. 8 is a sparse network of 20 nodes. The total number of possible edges is 190, and this network has 19 edges. We fix  $r = 0.1$  and change the number of nodes  $n$  to 5, 20, and 100. The result is shown in Fig. 9. This figure shows that as  $n$  increases, the final absolute value of error decreases. In addition, the algorithm converges faster with more nodes (100 nodes and 20 nodes). It is worth noting that the  $r$  value for 5 nodes is 0.4 (the minimum number of edges is 4) resulting in faster convergence than 20 nodes.

Fig. 10 shows the RMS Error over 10,000 simulations for different numbers of nodes  $n$  between 5 and 100. i.e., we averaged the normalized error at all 10,000 simulations for all  $n$  values. The case  $n = 5$  gives the highest error with the largest variability. As the number of nodes increases, the error decreases and its variability lessens. Large values of  $N$

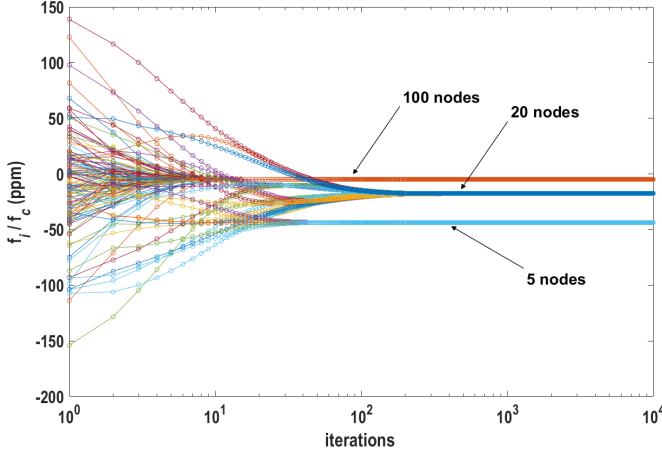


Fig. 9: Normalized error comparison under varying number of nodes of sparse networks with  $r = 0.1$  (20 and 100 nodes) and  $r = 0.4$  (5 nodes).

achieve similar performance; in particular, they have small errors. Indeed, the RMS Error for a small size network with 5 nodes is 12.8 ppm, it reaches about 2.9 ppm and 4.1 ppm for large networks with 100 and 60 nodes, respectively, while 6.5 ppm error is obtained by a medium size network (20 nodes). Thus the large 60 and 100 nodes networks save 43% and 55% of the error, respectively comparing to a 20 node network.

It is important to note that the error holds irrespective of the connectivity  $r$  of the network, as will be demonstrated later.

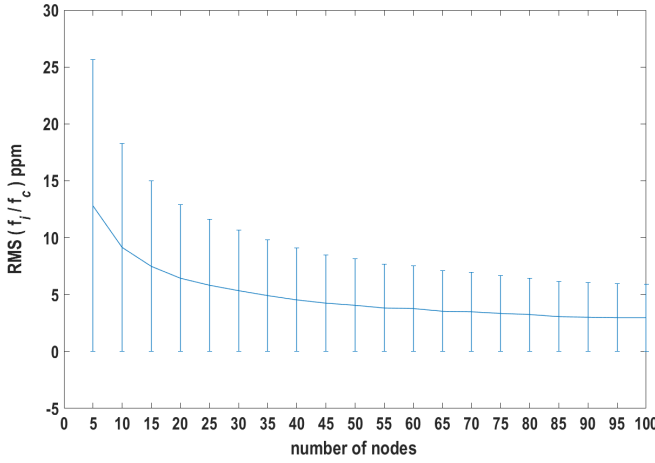


Fig. 10: RMS error for varying networks.

Then we compare the convergence speed for different connectivity. We measure the number of iterations required for consensus. We evaluate the stopping criteria for DFAC algorithm as  $\max |f(k) - \bar{f}| < 2 \times 10^{-3}$ . The results for  $n = 5, 20, 60$ , and 100 are shown in Fig. 11. We plotted the means and variances for 10,000 simulations. For all values of  $r \geq 0.1$ , the two large clusters (60 and 100 nodes) perform better than the other two (5 and 20 nodes). For example, with  $r = 0.1$ , the number of iterations for 100 nodes is about 57. It reaches 819 for 20 nodes, while the second smallest number is attained by 60 nodes that requires

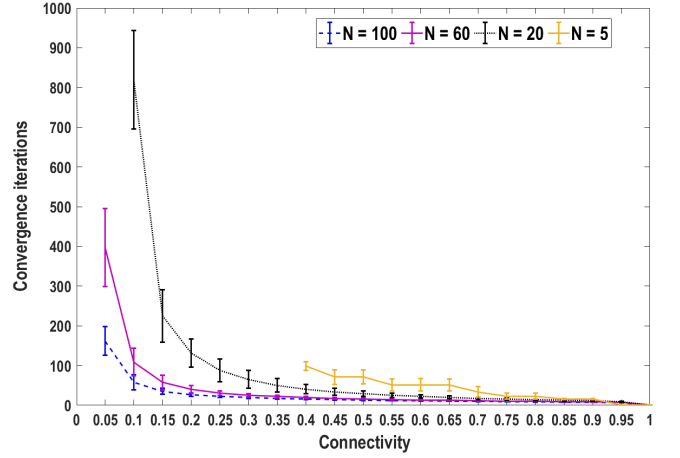


Fig. 11: Convergence iterations for varying networks.

around 108 iterations. Hence the 100 nodes network saves about 47% of the iterations comparing to 60 nodes, and this induces 47% less transmissions among the nodes. Though the number of iterations is smaller for 100 nodes, the number of communications is larger for each iteration, compared to 60 nodes. This is because the number of edges is bigger (on average there are 6 nodes connected to one node for a 60 nodes network, while the number increases to 10 for a 100 node network). As the connectivity  $r$  increases, the number of iterations decreases for all networks. A large connectivity (high density of links) is probably unrealistic as it would promote interference among the nodes. In fact, reducing the convergence time is important, as it leads to a smaller number of messages exchanged among the nodes and to a reduced energy cost of the algorithm.

2) *Varying connectivity:* For  $n = 100$ , the sparsity of the network is changed from a sparse connectivity ( $r = 0.05$ ) to a dense connectivity ( $r = 0.9$ ). The state trajectories are shown in Fig. 12. It is clear that as the number of edges increases, connectivity increases, and the settling time of the state trajectories decreases. An explanation of this trend can be inferred by looking at  $\lambda_2(\mathbf{W})$  (the second largest eigenvalue in magnitude) of the undirected graph that is tightly linked to the convergence speed of the algorithm. Indeed,  $\lambda_2$  is relatively small for dense networks and is relatively large for sparse networks. Therefore, a dense network solves a consensus problem faster than a sparse network.

It can be seen that the error remains unchanged for changing connectivity  $r$ . This is solely due to the rational of the average consensus, i.e., the average consensus algorithm guarantees that the iterative procedure  $\mathbf{f}(k) = \mathbf{W}\mathbf{f}(k-1)$  leads all the estimates to converge to  $\bar{f}$ , irrespective of the connectivity of the network as long as this network is connected.

#### IV. DYNAMIC FREQUENCY ALIGNMENT IN COHERENT DISTRIBUTED ARRAYS

To investigate the performance of the average consensus algorithm in a realistic environment, we consider two scenarios: changing mixing matrix and changing frequencies due to oscillator drift.

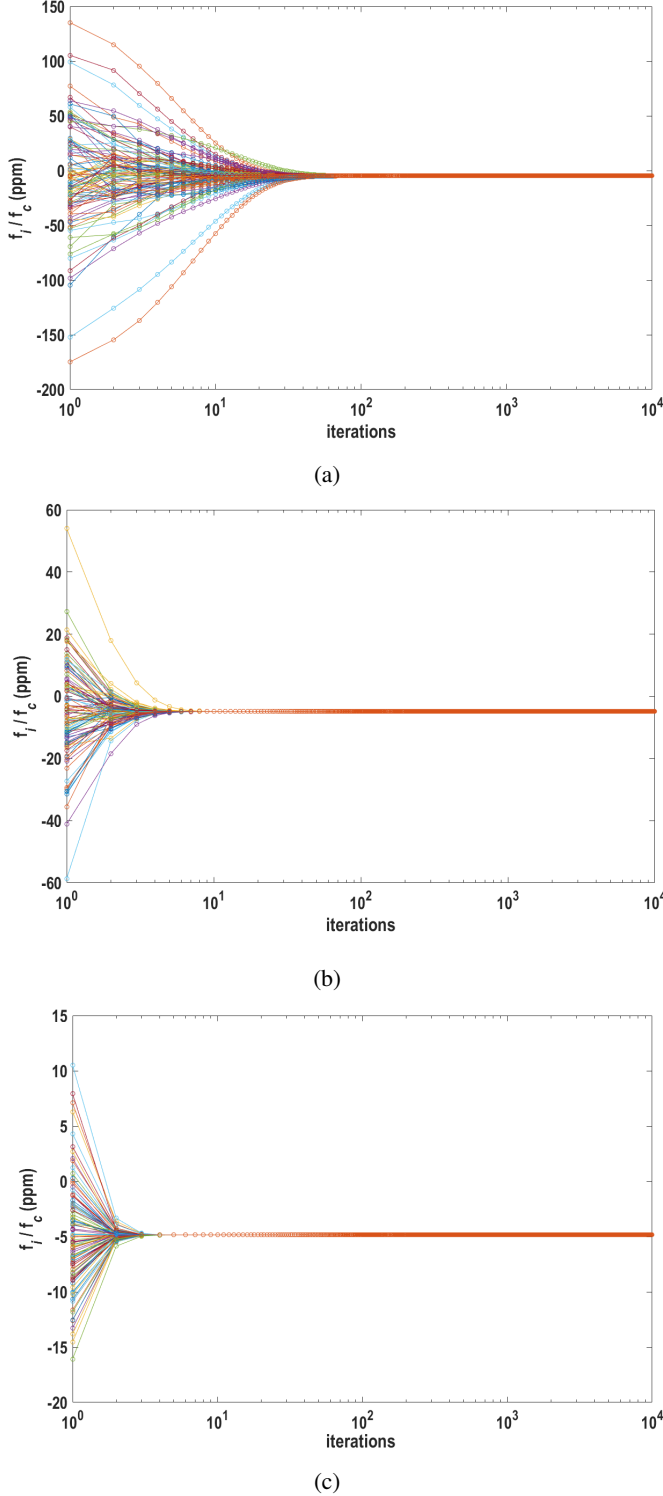


Fig. 12: State trajectories of all nodes corresponding to network with  $n = 100$ . (a)  $r = 0.05$ . (b)  $r = 0.3$ . (c)  $r = 0.9$ .

#### A. Dynamic Array Connectivity

In Section III, the network topology is assumed stationary. But wireless networks are intrinsically dynamic, i.e., due to the presence of obstructions, or fluctuating detection range. In this regard, the effects on the consensus of a dynamic network,

i.e., a network with a time-varying topology, should be taken into consideration. Herein we are interested to investigate the convergence time of a dynamic network. We assume a varying interconnection topology in which the number of nodes is fixed but the communication links are changing; links are restored and failed, i.e., edges are added and removed from the graph. Algorithm 2 describes the detailed operation of changing the mixing matrix.

---

#### Algorithm 2 DFAC with dynamic networks

---

**Input:**  $k = 0$ , the initial estimates  $\mathbf{f}(0)$ , mixing matrix  $\mathbf{W}$ , probabilities  $P_1 \leq 1$   
**while** stopping criteria is not satisfied **do**  
      $k = k + 1$   
     Generate a random number in  $\alpha \in [0, 1]$  from the uniform distribution  
     **if**  $\alpha \in (P_1, P_1 + (1 - P_1)/2]$  **then**  
         Randomly choose one node  $i$   
         Randomly choose another node  $j$  in its neighbor  
         Change the submatrix of  $\mathbf{W}$  with  $i, j$  rows and columns:  $[w_{ii}, w_{ij}, w_{ji}, w_{jj}] \rightarrow [w_{ii} + w_{ij}, 0, 0, w_{ji} + w_{jj}]$   
     **else if**  $\alpha > P_1 + (1 - P_1)/2$  **then**  
         Randomly choose one node  $i$   
         Randomly choose another not connected node  $j$   
         Change the submatrix of  $\mathbf{W}$  with  $i, j$  rows and columns:  $[w_{ii}, 0, 0, w_{jj}] \rightarrow [w_{ii} - w, w, w, w_{jj} - w]$ , with  $w = 0.2 \times \min\{w_{ii}, w_{jj}\}$   
     **end if**  
      $\mathbf{f}(k) = \mathbf{W}\mathbf{f}(k - 1)$   
**end while**  
**return**  $\mathbf{f}(k)$

---

We consider a network with  $n = 100$  nodes and connectivity  $r = 0.03$ . At every iteration  $k$ , the topology of the network is reconfigured. A number  $\alpha$  is randomly generated from a uniform distribution from  $[0, 1]$ . If  $\alpha < 1$ , then we do not change the mixing matrix. Otherwise, depending on the value of  $\alpha$ , we either add or remove one edge from the network. When we remove one edge  $(i, j)$ , we move the weight  $w_{ij}$  and  $w_{ji}$  onto the nodes  $i$  and  $j$ . That is  $[w_{ii}, w_{ij}, w_{ji}, w_{jj}] \rightarrow [w_{ii} + w_{ij}, 0, 0, w_{ji} + w_{jj}]$ . When we add a new edge  $(i, j)$ , we move the weight  $w$  from nodes  $i$  and  $j$  to the edge, i.e.,  $[w_{ii}, 0, 0, w_{jj}] \rightarrow [w_{ii} - w, w, w, w_{jj} - w]$ , with  $w = 0.2 \times \min\{w_{ii}, w_{jj}\}$ . These operations keep the properties of the mixing matrix. When  $P_1 = 1$ , no changes occur and the matrix is not reconfigured. Fig. 13(a) shows the state trajectories of this dynamic network. One can observe that a consensus is asymptotically reached. The results of the original case (static network with  $n = 100$  and  $r = 0.03$  are shown in Fig. 13(b) for comparison). As can be seen from both graphs, the reconfiguration decreases the settling time of the state trajectories. This character is consistent with the above results that explain the decrease in convergence time by the relatively unpronounced eigenvalue  $\lambda_2$  of the network. Assume that the matrix  $W1$  is changed to  $W$ . Then  $\lambda_2(W \cdot W1)$  of the product of changing mixing matrix from  $W1$  to  $W$  is consistently less than  $\lambda_2(W \cdot W)$  of the product of the matrix

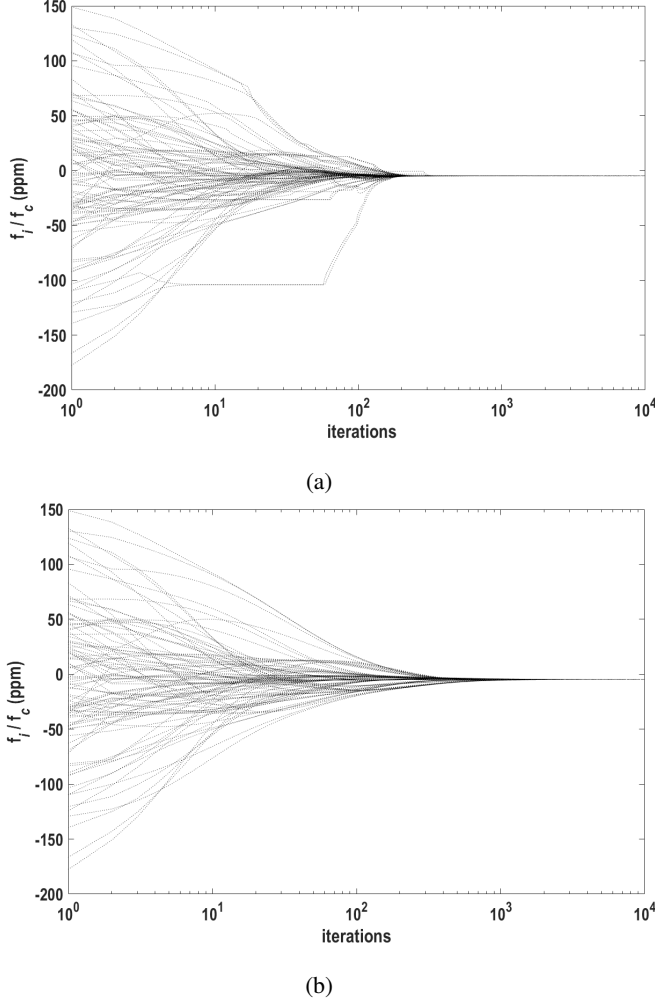


Fig. 13: State trajectories of all nodes corresponding to network with  $N = 100$  with  $r = 0.03$ . (a) Dynamic network. (b) Static network.

by itself (no changes in the matrix). To support this argument, Fig. 14 compares the normal distribution curves of 10,000 sampled  $\lambda_2$  of changing and unchanging mixing matrices. It can be seen that  $\lambda_2(W \cdot W1) < \lambda_2(W \cdot W)$  with a high probability.

We extended Algorithm 2's performance in relation to different values of network connectivity for different values of  $P_1$ . The results presented in Fig. 15 are achieved in a network consisting of 100 nodes. Results shown on the figure are illustrating the algorithm's performance for  $P \in \langle \text{zero } W \text{ change, add edge, remove edge} \rangle$ . In the same case with connectivity  $r = 0.03$ , the average convergence iterations is 6400 for the static network, 5312 for  $P = \langle 0.9, 0.05, 0.05 \rangle$ , 884 for  $P = \langle 0.3, 0.35, 0.35 \rangle$ , and 821 for  $P = \langle 0.0, 0.5, 0.5 \rangle$ . With  $r = 0.08$ , the average is 159 for  $P = \langle 0.0, 0.5, 0.5 \rangle$ , 154 for  $P = \langle 0.3, 0.35, 0.35 \rangle$ , 140 for  $P = \langle 0.9, 0.05, 0.05 \rangle$ , and 126 for static case. For  $r \geq 0.13$ , the difference between average convergence of dynamic and static networks narrows down. A key observation of these results is that the convergence depends strongly on the connectivity and the network dynamic. As the probability of dynamic connectivity decreases, and

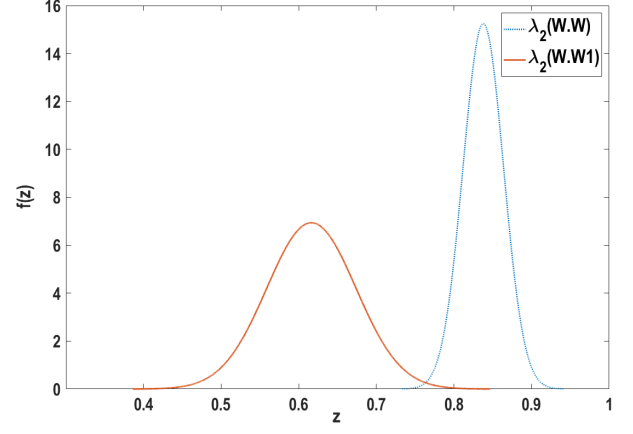


Fig. 14: Normal distribution of the second largest eigenvalue  $\lambda_2$  samples of changing and unchanging mixing matrices.  $N = 100$ ,  $r = 0.05$ .

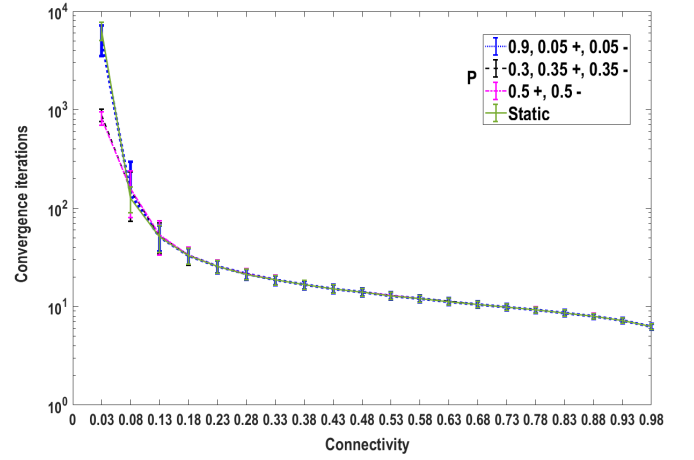


Fig. 15: Convergence iterations for dynamic networks.

the network connectivity increases, the convergence iterations decreases.

### B. Oscillator drift

The previous sections established the ability to syntonize the oscillators in the absence of oscillator frequency drift. In practice, the wireless syntonization signals may be implemented periodically, thus the frequencies will be aligned and then drift for some period of time before being aligned again. In this section, the convergence of the decentralized consensus averaging algorithm in the presence of such oscillator drift is analyzed.

The rate at which the frequency of an oscillator drifts is strongly dependent on the design of the oscillator. A common metric is the Allan deviation (ADEV), which is the square-root of the two-sample variance of the oscillator frequency drift, thus a lower ADEV indicates a more stable oscillator. The ADEV of common temperature-controlled crystal oscillators (TCXOs) is on the order of  $1 \times 10^{-9}$  to  $1 \times 10^{-10}$  over a 1s interval. At short intervals, the frequency drift is heavily



influenced by noise fluctuations, and as the interval increases these fluctuations tend to average out, resulting in a decreased ADEV and a more stable frequency. Eventually, long-term drift causes the ADEV to increase. In synchronizing systems in a coherent distributed array, it is necessary to perform coordination updates relatively frequently since the nodes in the array may be in motion. Algorithm 3 describes the detailed operation of the consensus algorithm with oscillator drift.

---

**Algorithm 3** DFAC with oscillator drift

---

**Input:**  $k = 0$ , the initial estimates  $\mathbf{f}(0)$ , mixing matrix  $\mathbf{W}$   
**while** stopping criteria is not satisfied **do**  
     $k = k + 1$   
    Generate random vector in  $\boldsymbol{\alpha} \in [0, \sigma_\phi]$  from uniform distribution, where  $\sigma_\phi$  is the phase error due to random frequency drift  
     $\mathbf{f}(k-1) = \boldsymbol{\alpha} + \mathbf{f}(k-1)$   
     $\mathbf{f}(k) = \mathbf{W}\mathbf{f}(k-1)$   
**end while**  
**return**  $\mathbf{f}(k)$

---

For this work, the decentralized consensus algorithm is investigated using a 1s interval time, over which the ADEV is taken to be  $1 \times 10^{-9}$ . The frequencies of each node in the array are updated at each processing interval with a frequency drift randomly drawn from a distribution matching the ADEV at a carrier frequency of 1 GHz. The array was set up with a connectivity ratio of 0.15 (or greater if a connectivity of 0.15 was not possible due to the array size being too small). The convergence, standard deviation of the total phase error, and the coherent gain (12) were evaluated for array sizes of 20, 60, and 100 nodes.

Fig. 16 shows the evolution of the phases for the cases with  $N = 20$ ,  $N = 60$ , and  $N = 100$ , respectively. The phase differences are quite large initially for both, but converge to a tight cluster around 130, 55, and 25 iterations, respectively. The standard deviation of these phases determines the level of coherent gain achieved. Fig. 17 shows the standard deviation of the phases of all the oscillators in the array as a function of iteration. It can be seen that as the array size increases the convergence time decreases. This is because the connectivity ratio was fixed to be 0.15, meaning that larger arrays enable the information to propagate more quickly through the array. The dotted line indicates the  $18^\circ$  requirement discussed in Section II, and clearly the decentralized algorithm enables phase errors below this level even with the relatively poor frequency stability of  $1 \times 10^{-9}$ . Fig. 18 shows the level of coherent gain supported by the array in the presence of phase errors, showing that in each case the algorithm enables coherent gain levels well above 90% of the ideal coherent gain, indicated by the dotted line.

### C. Discussion

The numerical results demonstrate that the decentralized consensus protocol can be applied to other topologies that realistically model arrays dynamics, and should allow coherent gain of the distributed beamformer when (a) there is a high

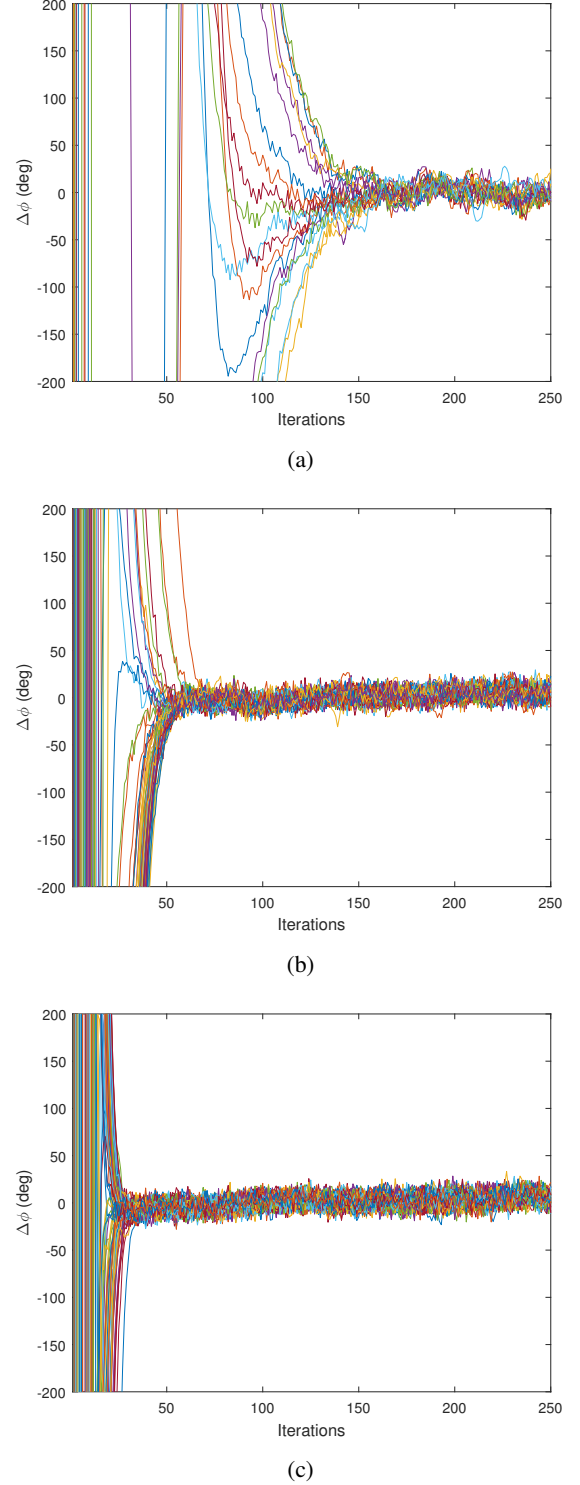


Fig. 16: Phases of the oscillators per iteration. (a)  $N = 20$ . (b)  $N = 60$ . (c)  $N = 100$ .

probability of link failure and restoration (b) low frequency stability of the oscillators is present. From the above results, it can be concluded that oscillator drift can be considered as the main source of errors in frequency. In systems with mobility, minimizing the number of convergence iterations is

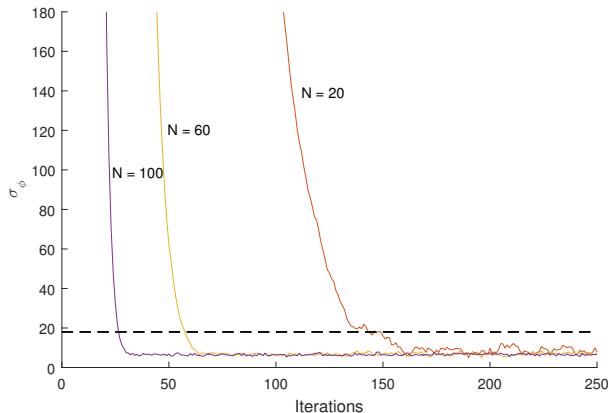


Fig. 17: Standard deviation of the phases in the arrays. The dotted line indicates the  $18^\circ$  requirement to achieve 90% of the ideal coherent gain.

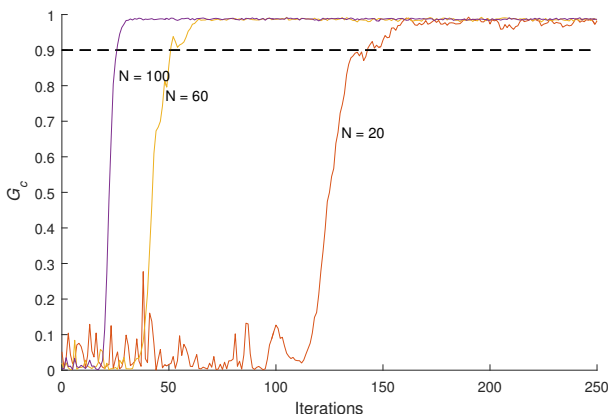


Fig. 18: Coherent beamforming gain in the antenna mainbeam enabled with the total phase errors given in Fig. 17. The dotted line indicates 90% of the ideal coherent gain.

necessary to reduce energy consumption. This can be achieved by insuring that sufficient communication links are restored.

## V. CONCLUSION

Herein, we have proposed a decentralized average consensus method for distributed achieving frequency synchronization, which is vital for distributed beamforming in open loop coherent distributed arrays, without relying on a centralized approach. We have analyzed the performance of the algorithm in undirected (full duplex) networks with static and dynamic topologies. The algorithm is scalable, in that convergence time decreases with the increasing number of participating nodes. The algorithm is found to provide good performance even under adverse conditions such as link failure or oscillator drift. Our ongoing work regards hardware implementation of the promising properties of the suggested protocol and exploration of the decentralized consensus in directed (half-duplex) networks.

## ACKNOWLEDGMENT

The authors gratefully acknowledge enlightening discussions with Prof. Sandeep Kulkarni, Department of Computer Science and Engineering, Michigan State University.

## REFERENCES

- [1] J. A. Nanzer, R. L. Schmid, T. M. Comberiate, and J. E. Hodkin, "Open-loop coherent distributed arrays," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 5, pp. 1662–1672, May 2017.
- [2] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1754–1763, May 2007.
- [3] M. Dohler, J. Dominguez, and H. Aghvami, "Link capacity analysis for virtual antenna arrays," in *Proceedings IEEE 56th Vehicular Technology Conference*, vol. 1, Sep. 2002, pp. 440–443 vol.1.
- [4] S. Xi and M. D. Zoltowski, "Multiuser transmit beamforming design for sinr maximization in cooperative mimo systems," in *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, Aug 2007, pp. 408–412.
- [5] P. Bidigare, M. Oyarzyn, D. Raeman, D. Chang, D. Cousins, R. O'Donnell, C. Obranovich, and D. R. Brown, "Implementation and demonstration of receiver-coordinated distributed transmit beamforming across an ad-hoc radio network," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov 2012, pp. 222–226.
- [6] R. Mudumbai, B. J. Wild, U. Madhow, and K. Ramchandran, "Distributed beamforming using 1 bit feedback : from concept to realization," in *Proceedings of the 44th Allerton conference on communication, control and computation*, 2006.
- [7] P. Bidigare, D. Brown, U. Madhow, R. Mudumbai, and A. Kumar, "Wideband distributed transmit beamforming using reciprocity with endogenous relative calibration," in *Asilomar Conference on Signals, Systems, and Computers, 2015 IEEE*, 2015.
- [8] B. Peiffer, R. Mudumbai, A. Kruger, A. Kumar, and S. Dasgupta, "Experimental demonstration of a distributed antenna array pre-synchronized for retrodirective transmission," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 460–465.
- [9] D. R. Brown III and H. V. Poor, "Time-slotted round-trip carrier synchronization for distributed beamforming," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5630–5643, Nov 2008.
- [10] R. D. Preuss and D. R. Brown, III, "Two-way synchronization for coordinated multicell retrodirective downlink beamforming," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5415–5427, Nov 2011.
- [11] 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 *2017 IEEE MTT-S International Microwave Symposium (IMS)*, June 2017, pp. 884–887.
- [12] R. L. Schmid, T. M. Comberiate, J. E. Hodkin, and J. A. Nanzer, "A distributed rf transmitter using one-way wireless clock transfer," *IEEE Microwave and Wireless Components Letters*, vol. 27, no. 2, pp. 195–197, Feb 2017.
- [13] M. D. Sharp and J. A. Nanzer, "Accuracy of angle rate measurements using a distributed radar with a correlation receiver," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 2, pp. 209–212, Feb 2018.
- [14] R. M. Leahy and B. D. Jeffs, "On the design of maximally sparse beamforming arrays," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 8, pp. 1178–1187, Aug 1991.
- [15] K. Chen, X. Yun, Z. He, and C. Han, "Synthesis of sparse planar arrays using modified real genetic algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 4, pp. 1067–1073, April 2007.
- [16] F. Namin, J. S. Petko, and D. H. Werner, "Analysis and design optimization of robust aperiodic micro-uav swarm-based antenna arrays," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 5, pp. 2295–2308, May 2012.
- [17] B. Fuchs, "Synthesis of sparse arrays with focused or shaped beam-pattern via sequential convex optimizations," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 7, pp. 3499–3503, July 2012.
- [18] D. J. Ludick, R. Maaskant, D. B. Davidson, U. Jakobus, R. Mittra, and D. de Villiers, "Efficient analysis of large aperiodic antenna arrays using the domain green's function method," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 4, pp. 1579–1588, April 2014.
- [19] S. M. Ellison and J. A. Nanzer, "Comparison of crossover recombination operators in ga-optimized sparse linear array design," in *2017 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, July 2017, pp. 145–146.

- [20] —, “Two-step optimization of linear distributed arrays,” in *2018 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, July 2018, pp. 2111–2112.
- [21] T. M. Comberiate, R. L. Schmid, J. E. Hodkin, and J. A. Nanzer, “Platform placement for sidelobe mitigation in mobile sparse arrays,” in *2016 USNC-URSI Radio Science Meeting*, June 2016, pp. 7–8.
- [22] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, “Distributed transmit beamforming using feedback control,” *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 411–426, Jan 2010.
- [23] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: algorithms and theory,” *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.
- [24] D. P. Scharf, J. A. Keim, and F. Y. Hadaegh, “Flight-like ground demonstrations of precision maneuvers for spacecraft formations—part i,” *IEEE Systems Journal*, vol. 4, no. 1, pp. 84–95, March 2010.
- [25] P. Chatterjee and J. A. Nanzer, “Effects of time alignment errors in coherent distributed radar,” in *2018 IEEE Radar Conference (Radar-Conf18)*, April 2018, pp. 0727–0731.
- [26] J. S. Petko and D. H. Werner, “Positional tolerance analysis and error correction of micro-uav swarm based antenna arrays,” in *2009 IEEE Antennas and Propagation Society International Symposium*, June 2009, pp. 1–4.
- [27] Z. Chen, J. Chamberland, and G. H. Huff, “Impact of uav swarm density and heterogeneity on synthetic aperture doa convergence,” in *2017 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium)*, July 2017, pp. 3–4.
- [28] S. Yeh, J. Chamberland, and G. H. Huff, “An investigation of geolocation-aware beamforming algorithms for swarming uavs,” in *2017 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, July 2017, pp. 641–642.
- [29] P. Chatterjee and J. A. Nanzer, “A study of coherent gain degradation due to node vibrations in open loop coherent distributed arrays,” in *2017 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium)*, July 2017, pp. 115–116.