Pulse Coupled Synchronization with Guaranteed Clock Continuity

Timothy Anglea[1] (*IEEE Student Member*) Yongqiang Wang[1] (*IEEE Senior Member*)

***Abstract*—Clock synchronization is a widely discussed topic in the engineering literature. Ensuring that individual clocks are closely aligned is important in network systems, since the correct timing of various events in a network is usually necessary for proper system implementation. However, many existing clock synchronization algorithms update clock values abruptly and instantaneously, resulting in discontinuous clocks, which have been shown to lead to undesirable behavior. In this paper, we explore the generalization of the pulse-coupled oscillator model to guarantee clock continuity, achieving continuous phase evolution in any pulse-coupled oscillator network. We provide rigorous mathematical proof for pulse coupled synchronization under the proposed phase continuity methods, along with analysis of the time to synchronization under phase continuity. Two simple methods to achieve the desired continuity are presented. We further provide simulation and experimental results supporting these proofs, analyzing the effects of the phase continuity methods to convergence. We also investigate the convergence behavior of other pulse-coupled oscillator synchronization algorithms using the proposed methods.**

Index - clock continuity, synchronization, pulse-coupled oscillators, phase jumps

I. Introduction

Ensuring clock synchronization in a distributed system is a very important and well-studied topic in the fields of electrical engineering and computer science. However, many synchronization algorithms require an instantaneous clock value adjustment [1], which results in clocks with discontinuous time stamps. These discontinuous clocks are undesirable, especially if certain events or processes are time-dependent. If time jumps forward, then there is the potential that a scheduled event will never happen, and if time jumps backwards, then one process may be implemented twice, as shown in Fig. 1. It is desirable that a clock's time value evolves continuously while synchronizing with other clocks within the network [2], [3].

Many clock synchronization algorithms use packet-based communication to share local information [4], [5], [6] and can achieve synchronization with a continuous clock [2], [3]. However, as noted in [5], such approaches may require constant adjustment to the clock rate, leading to significant runtime overhead. In this paper, we will show that the pulse-coupled oscillator (PCO) model can be used to achieve clock synchronization with clock continuity.

The PCO model was first introduced by Peskin in 1975 [7]. He used pulse-coupled oscillators (PCOs) to model the synchronization of pacemaker cells in the heart. Mirollo and Strogatz later improved the model, providing a rigorous mathematical formulation [8]. Communication latency, packet loss, signal corruption, and energy consumption are all minimized due to the simplicity of the communication between oscillators in the network (i.e., single pulses).

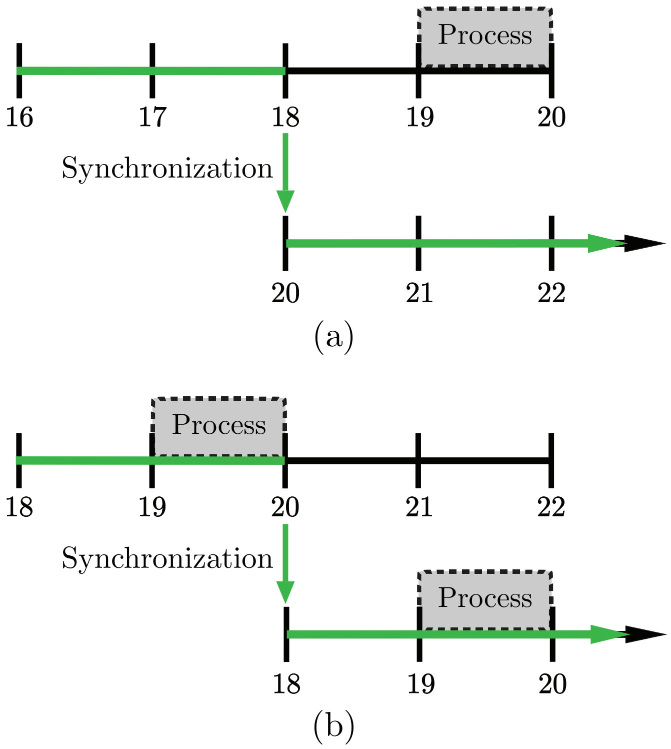


Figure 1. Illustration of disadvantages of discontinuous clock synchronization. (a) Synchronization occurs after a jump forward of two time units, and the scheduled process is not executed. (b) Synchronization occurs after a jump backward of two time units, and the scheduled process is executed twice.

Recently, the PCO model has been widely used to synchronize clocks in wireless sensor networks. However, all of the existing PCO synchronization algorithms achieve synchronization through abrupt jumps in the oscillator phase [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], which, as indicated earlier, may lead to undesired behavior when the phase variable is representative of network time. In this paper, we propose a generalization of the PCO model, which can guarantee continuity in phase evolution for any jump-based PCO algorithm. We will show that the convergence properties of previously proposed jump-based synchronization algorithms in [11], [12] are maintained, even when the phase adjustment

*The work was supported in part by the National Science Foundation under Grant 1738902.

[1] Timothy Anglea and Yongqiang Wang are with the department of Electrical & Computer Engineering, Clemson University, Clemson, SC 29634, USA {tbangle,yongqiw}@clemson.edu

rule is modified to be continuous. To do so, we will prove that the generalization that guarantees phase continuity amounts to a reduction of the coupling strength in the standard PCO model. To our knowledge, this is also the first paper to consider a time-varying coupling strength in a PCO network.

In Section II, we will define our terms and notation for a PCO network. In Section III, we will analyze the behavior of an oscillator under continuous phase evolution in a PCO network, and show that the behavior can be modeled as a bounded, time-varying coupling strength. In Section IV, we will analyze the convergence properties of PCOs under time-varying coupling strengths, and show that a PCO network will synchronize under guaranteed phase continuity. We also determine the time to synchronization for the network. We will then introduce and discuss two simple methods for maintaining the continuity of the oscillator phases in Section V. We use numerical and hardware experimental results to evaluate the proposed approaches in Section VI. We will conclude with final remarks and future work in Section VII.

II. Pulse Coupled Oscillator Preliminaries

A. Oscillator Phase Evolution

Let us consider a network of N identical pulse-coupled oscillators. Let $\theta_i \in [0,1)$ be the associated phase of oscillator $i \in \mathcal{V} = \{1, 2, \cdots, N\}$. For our analysis, each oscillator has an identical fundamental frequency, ω_0, and evolves naturally at that rate on the interval $[0,1)$.

As the network evolves, each oscillator increases its phase at rate ω_0. When an oscillator reaches the threshold value of 1, it fires a pulse and resets its phase to zero. Any connected oscillators then receive that pulse, being notified of the firing instance of an oscillator in the network. Receiving a pulse will cause an oscillator to change its phase in accordance with the chosen PCO synchronization algorithm. Let us denote the amount that oscillator i determines to adjust its phase, given a PCO synchronization algorithm, at a firing instance at time t as

$$\psi_i = \alpha\phi_i = \lim_{\tau \downarrow 0} \big(\theta_i(t+\tau)\big) - \theta_i(t) = \theta_i(t^+) - \theta_i(t) \qquad (1)$$

where α is the coupling strength of the network, and ϕ_i is the phase change amount determined by the PCO synchronization algorithm. The range of the coupling strength allowed for the network is determined by the algorithm, but typical values of coupling strength are within the interval $(0,1]$.

B. Graph Formulations

A PCO network can be described as a directed graph $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_N\}$ is a set of vertices corresponding to the oscillators in the network, and E is the set of directed edges (v_{i_1}, v_{i_2}) corresponding to the interaction between two oscillators [21]. An edge (v_i, v_j) means that oscillator v_j can receive pulses from oscillator v_i. A directed path is a sequence of edges $(v_i, v_j), (v_j, v_k), (v_k, v_\ell), \ldots$ in a graph G. A graph G is considered to be strongly connected if there is a directed path between any pair of vertices in the graph. For a vertex v_i, the outdegree of the vertex, denoted $\delta^+(v_i)$, is the number of edges that leave vertex v_i. Similarly, the indegree of the vertex, denoted $\delta^-(v_i)$, is the number of edges that enter vertex v_i. Alternatively, the outdegree and indegree of a vertex is the number of edges in graph G that have that vertex as the first and second entry, respectively. The value $\delta^+(G) \triangleq \min_{v \in V} \delta^+(v)$ is called the outdegree of the graph G, and the value $\delta^-(G) \triangleq \min_{v \in V} \delta^-(v)$ is called the indegree of the graph G.

C. Phase Continuity

To our knowledge, all existing literature regarding PCO networks has the phase value of each oscillator jump discontinuously at firing instances. In this paper, we will generalize the standard PCO model such that the phase value must evolve continuously at all times (except when it resets its phase to zero when it reaches the threshold). To ensure phase continuity, when oscillator i receives a pulse, it must increase or decrease its individual rate of evolution, ω_i, for a certain amount of time, τ_i, in order to achieve the required phase adjustment, ψ_i, as in (1). If oscillator i receives another pulse before the time needed to achieve ψ_i is completed, then the oscillator will use its current phase θ_i to redetermine a new ψ_i, and thus determine new values for ω_i and τ_i.

III. Oscillator Analysis

We now rigorously analyze the dynamics of oscillators maintaining continuous phase evolution, rather than using phase jumps. All that is required is to take the amount of phase adjustment for oscillator i, i.e. ψ_i, and determine the necessary amount of time, τ_i, and change in frequency ω_i. We then let the oscillator evolve at the new frequency for the required amount of time before it returns to its fundamental frequency, ω_0.

A. Single Oscillator Behavior

Let us analyze the behavior of a single oscillator. Once oscillator i has received a pulse and calculated the necessary change in frequency, ω_i, to achieve the phase adjustment ψ_i in time τ_i, two possibilities can follow: 1) the oscillator receives no new pulses within time τ_i that cause a phase adjustment, and 2) the oscillator receives a new pulse (i.e., the current pulse) within time τ_i that causes a phase adjustment. If the duration of the time interval between the current and previous oscillator firing instances is given as t_0, we can divide these two cases mathematically as 1) $t_0 \geq \tau_i$, and 2) $t_0 < \tau_i$.

1) In the first case, oscillator i finishes adjusting its phase by ψ_i, and returns to evolving at the fundamental frequency ω_0. The same effective change in phase has been achieved as if the oscillator had jumped in phase by ψ_i and evolved normally for a time of length t_0. Thus, no effective change in the phase update rule occurs compared with the standard instantaneous jump-based PCO model.
2) In the second case, the oscillator has not yet achieved its desired amount of phase change. Rather than having adjusted the whole amount ψ_i, it has adjusted only

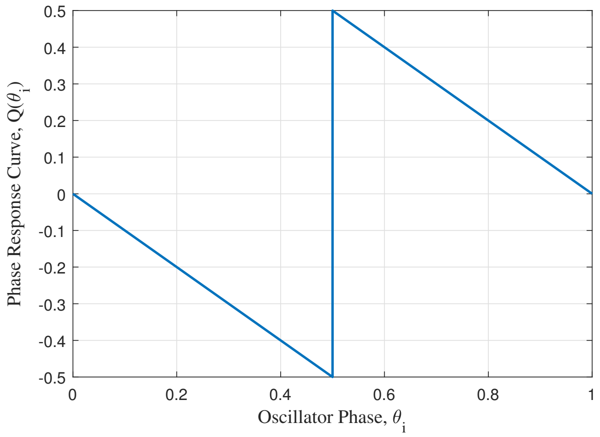


Figure 2. A delay-advance phase response curve (PRC) for PRC synchronization given in (3)

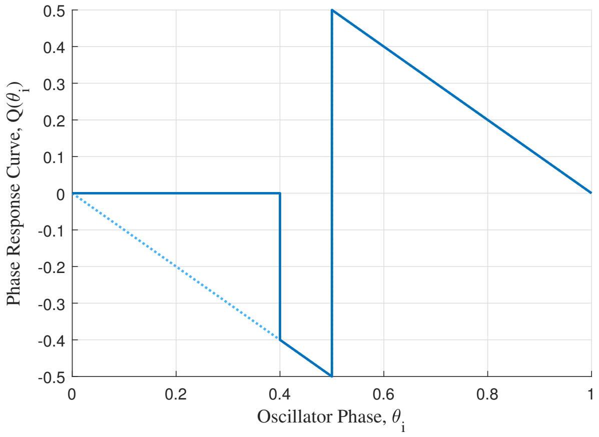


Figure 3. A delay-advance phase response curve (PRC) for PRC synchronization given in (3) with an example refractory period $D = 0.4$.

a portion of that amount, $\frac{t_0}{\tau_i}\psi_i$, in the time interval between received pulses. The oscillator then will use its current phase at the time when the new pulse is received to redetermine new values for ψ_i, τ_i, and ω_i. This truncated phase evolution is equivalent to having the oscillator jump in phase by $\frac{t_0}{\tau_i}\psi_i$, and then evolve normally for a time of duration t_0. This fractional amount of the desired phase change can be seen as a reduction of the coupling strength, α, of oscillator i by the ratio $\frac{t_0}{\tau_i}$.

From (1), both of these cases allow us to write a generalized expression for the effective coupling strength, α_{e_i}, of the oscillator:

$$\alpha_{e_i} = \min\left\{\frac{t_0}{\tau_i}\alpha, \alpha\right\} \tag{2}$$

The quantity $\frac{t_0}{\tau_i}$ is greater than or equal to 1 in the first case, and less than 1 in the second case. Note that this expression is bounded by $(0, \alpha]$ and the value for an individual oscillator may vary over time as the PCO network evolves. This idea of an effective coupling strength leads us to analyze a standard PCO network under the condition of a time-varying coupling strength.

B. Time-Varying Coupling Strength

Let us consider a PCO network under a synchronization algorithm that allows jumps in the phase variable, θ, but has a coupling strength, α, that varies with time.

Proposition 1. *The evolution of an oscillator in a PCO network is dependent upon the value of the coupling strength, α, only at firing instances.*

Proof. The proof for this proposition is straightforward. An oscillator only determines the amount that it needs to jump when it receives a pulse. Thus, the value of the coupling strength is only used at firing instances. Any values the coupling strength takes between firing instances is unused and thus independent of the behavior of the network. Furthermore, if the oscillator receives a pulse, but does not jump (or jumps an amount of zero), then the coupling strength is again independent to the behavior of the network. □

Remark 1. *The phase continuity modification described above does not actually adjust the coupling strength as the network evolves. Only the apparent behavior of an oscillator is being modeled as a reduced coupling strength, α_{e_i}. The actual coupling strength, α, remains unchanged throughout the entire evolution of the network.*

IV. Synchronization Analysis

We will now use the PCO synchronization strategy given in [12] with phase jumps to determine if the convergence properties of the algorithm still hold under a time-varying coupling strength, and thus under the newly proposed phase continuity generalization. The algorithm in [12] uses a delay-advance phase response curve (PRC) to describe the phase update at firing instances.

A. Synchronization Condition

Consider a PCO network with N oscillators in a (strongly) connected graph. Without loss of generality, we can order the oscillators in the network according to their phase, such that oscillator 1 has the smallest phase and oscillator N has the largest phase (i.e., $0 \leq \theta_1 \leq \ldots \leq \theta_N < 1$). When an oscillator receives a pulse, it updates its phase variable according to the phase response curve, or function, Q, as shown in Fig. 2.

$$Q(\theta_i) = \begin{cases} -\theta_i & \text{if } 0 \leq \theta_i \leq \frac{1}{2} \\ (1-\theta_i) & \text{if } \frac{1}{2} < \theta_i < 1 \end{cases} \tag{3}$$

Note that the phase update is independent of the number and relative positions of other oscillators in the network. Thus, the phase of oscillator i after a firing instance can be described as

$$\theta_i(t^+) = \theta_i(t) + \alpha Q(\theta_i(t)) \tag{4}$$

where α is the coupling strength of the network, and, from (1), we have $Q(\theta_i(t)) = \phi_i$.

A refractory period of duration D can be included in the phase response curve, as shown in Fig. 3. An oscillator does not respond to incoming pulses if its phase is within the region $[0, D)$, and continues to freely evolve. Such a refractory period can improve energy-efficiency and robustness to communication latency [12].

Let us define an arc as a connected subset of the interval $[0, 1)$. We can thus define the following set of functions, $v_{i,i+1}$, for all $i \in \mathcal{V}$:

$$v_{i,i+1}(\theta) = \begin{cases} \theta_{i+1} - \theta_i & \text{if } \theta_{i+1} > \theta_i \\ 1 - (\theta_{i+1} - \theta_i) & \text{if } \theta_i > \theta_{i+1} \end{cases} \tag{5}$$

where oscillator $N + 1$ maps to oscillator 1. Notice that these functions do not change between firing instances.

We say that the containing arc of the oscillators is the smallest arc that contains all of the phases in the network. The length of this arc, Λ, is given mathematically as

$$\Lambda = 1 - \max_{i \in \mathcal{V}} \{v_{i,i+1}(\theta)\} \tag{6}$$

As proven in [12], Λ will decrease and converge to zero (i.e., the network will synchronize) under a constant coupling strength. Here, we show that Λ will decrease even when the coupling strength is time-varying.

Theorem 1. *Consider pulse-coupled oscillators with a refractory period D in any delay-advance phase response function (as in* (3) *and shown in Fig. 3). A strongly connected network of such PCOs using phase jumps will synchronize with oscillators having independent and time-varying $\alpha \in (0, 1]$ if the containing arc of the oscillators is less than some $\bar{\Lambda} \in (0, \frac{1}{2}]$, and if the refractory period D is not greater than $1 - \bar{\Lambda}$.*

Proof. Let us consider a PCO network where the initial phases are within some containing arc $\Lambda < \bar{\Lambda}$. Without loss of generality, let us assume that oscillator i has the largest initial phase, $\theta_{\max}$ at time $t = 0$, oscillator j has the smallest initial phase such that $\theta_j = \theta_{\max} - \Lambda$, and all other oscillator phases reside between oscillators i and j.

Since oscillator i has the largest phase, its phase evolves to 1 without perturbation and it reaches the threshold at $t = \frac{1-\theta_{\max}}{\omega_0}$. At this firing instance, all of the other oscillators have phases between $1 - \Lambda$ (which is larger than $\frac{1}{2}$) and 1. In the following time interval of length $\frac{\Lambda}{\omega_0}$, every oscillator will fire once. Since the network is strongly connected, oscillator j receives at least one pulse during its phase evolution from $1 - \Lambda$ to 1, and its phase is increased. (The value of the phase response curve is positive in the interval $(\frac{1}{2}, 1)$.) We denote the phase increase as ψ_j, which is strictly positive and dependent on the time-varying coupling strength, α, and the phase response curve, and hence is time-dependent. Given that the initial phase difference is Λ, and that the phase of oscillator j is increased by ψ_j, the containing arc of the network decreases by at least ψ_j, as oscillator i may have decreased its phase due to the pulse received while in the interval $[D, \Lambda)$, if $D < \Lambda$ holds. (The value of the phase response curve is negative in the interval $(0, \frac{1}{2})$). The network then continues on to the next cycle, and the above analysis repeats.

Therefore, since the containing arc of the network decreases with every cycle, and cannot be negative, then the containing arc converges to zero, and the network synchronizes. □

Remark 2. *The coupling strengths of the oscillators can vary independently from each other, and synchronization will still occur, as long as the coupling is within* $(0, 1]$.

Theorem 1 proves that a PCO network can synchronize for any potentially time-varying coupling strength, α. As shown in Sec. III, phase continuity can be modeled as a reduction in the coupling strength of an oscillator as in (2). Since the bound of this effective coupling strength is $(0, \alpha]$, the effective coupling strength will also be inside the bound $(0, 1]$. Thus, we easily have the following theorem.

Theorem 2. *Consider pulse-coupled oscillators with a refractory period D in any delay-advance phase response function. A strongly connected network of such PCOs with phase continuity as described in Sec. III will synchronize with $\alpha \in (0, 1]$ if the containing arc of the oscillators is less than some $\bar{\Lambda} \in (0, \frac{1}{2}]$, and if the refractory period D is not greater than $1 - \bar{\Lambda}$.*

Proof. This proof follows easily from Theorem 1. Phase continuity results in individual oscillator coupling strengths, α_{e_i}, being independent, time-varying, and bounded within the interval $(0, \alpha]$, as shown in the analysis of Sec. III. Thus, the conditions for Theorem 1 are met, and the network converges to the state of synchronization. □

B. Synchronization Time

As found in [12], the time to synchronization of the oscillator network using phase jumps is proportional to the containing arc of the oscillators and inversely proportional to the coupling strength, average phase response, and the indegree of the network topology. We can show that the time that a PCO network needs to synchronize under phase continuity is proportional to a similar expression.

Theorem 3. *Consider pulse-coupled oscillators having continuous phase using any delay-advance phase response function with a refractory period D (as in* (3) *and shown in Fig. 3) in a strongly connected network such that the containing arc of the oscillators Λ is less than some $\bar{\Lambda} \in (0, \frac{1}{2}]$, and the refractory period D not greater than $1 - \bar{\Lambda}$. Then the time to synchronization is proportional to*

$$\frac{\Lambda}{\alpha_e \Psi \delta^{-}(G)} \tag{7}$$

where $\alpha_e = \frac{1}{\delta^{-}(G)} \sum \alpha_{e_i}$ is the average effective coupling strength at firing instances of a single oscillator during a cycle, $\delta^{-}(G)$ is the indegree of the network represented by graph G, and Ψ denotes the average phase response of oscillators in the advance stage of the phase response function (i.e., $[\frac{1}{2}, 1)$), given by

$$\Psi = 2 \int_{\frac{1}{2}}^{1} Q(\theta) p(\theta) d\theta \tag{8}$$

where $Q(\theta)$ denotes the value of the phase response function when the phase is θ, and $p(\theta)$ denotes the probability that an oscillator receives a pulse at phase θ.

Proof. From Theorem 2, we know that the containing arc Λ decreases by at least amount ψ_j every cycle, so the time to synchronization is proportional to the size of the initial containing arc Λ, and is inversely proportional to the amount that the containing arc decreases, ψ_j. ψ_j represents the increase in the phase of the oscillator that fired last in a cycle, and is proportional to the sum of the product of the effective coupling strength, α_{e_i}, and the average phase increase in the advance stage of the phase response function caused by a single pulse, given by Ψ in (8), for all pulses received in a cycle. The number of pulses received by an oscillator is defined to be the indegree of the vertex corresponding to that oscillator, which is at least $\delta^-(G)$. Both Ψ and $\delta^-(G)$ can be factored from this sum, leaving the expression for the average effective coupling strength, α_e. Thus, we know that the time to synchronization is determined by (7). □

Remark 3. *The result of Theorem 3 is similar to that found in [12]. However, due to the condition of phase continuity, the time to synchronization is now inversely proportional to the average effective coupling strength, rather than the coupling strength directly. When considering phase continuity, the effective coupling strength will be less than or equal to the actual coupling strength of the network, implying that phase continuity will cause the network to converge more slowly. Furthermore, as the network synchronizes, the effective coupling for oscillator j, as in Theorem 2, will approach zero for all but the last pulse that is received. Thus, the sum of the effective coupling strengths will approach the value of the effective coupling from the last pulse received, and the dependence on the indegree of the network will disappear, increasing the total time to synchronization.*

V. Methods for Achieving Continuous Phase Evolution

In this section, we propose two methods for adjusting the phase of an oscillator in a continuous fashion. In the first method, the oscillator, in response to a received pulse, achieves the desired phase adjustment by increasing or decreasing its frequency by a fixed amount. In the second method, the oscillator achieves the desired phase adjustment in a fixed amount of time.

A. Constant Frequency Method

In the constant frequency method, the frequency of oscillator i is increased or decreased by a set amount ω_a for an adjustable duration of time τ_i. The amount of time the oscillator spends at this new frequency is dependent on the phase amount ψ_i that it needs to adjust:

$$\tau_i = \frac{|\psi_i|}{\omega_a} \tag{9}$$

Thus, once an amount of phase adjustment ψ_i is determined, the oscillator will increase its frequency to $\omega_i = \omega_0 + \omega_a$ if ψ_i is positive, or decrease its frequency to $\omega_i = \omega_0 - \omega_a$ if ψ_i is negative, for time τ_i determined in (9). Once the time τ_i has elapsed, the oscillator returns to its fundamental frequency ω_0. If ψ_i is zero, then the oscillator remains at its fundamental frequency, ω_0, and evolves until the next firing instance.

B. Constant Time Method

In the constant time method, oscillator i spends a fixed amount of time τ at an adjustable frequency ω_i. The new frequency at which the oscillator evolves is dependent on the phase amount ψ_i the oscillator needs to adjust:

$$\omega_i = \omega_0 + \omega_a = \omega_0 + \frac{\psi_i}{\tau} \tag{10}$$

Note that ω_a can be positive or negative. Thus, once an amount of phase adjustment ψ_i is determined, the oscillator will update its frequency to $\omega_i = \omega_0 + \omega_a$ for time τ. Once the fixed amount of time τ has elapsed, the oscillator again returns to its fundamental frequency ω_0.

C. Remarks on Methods

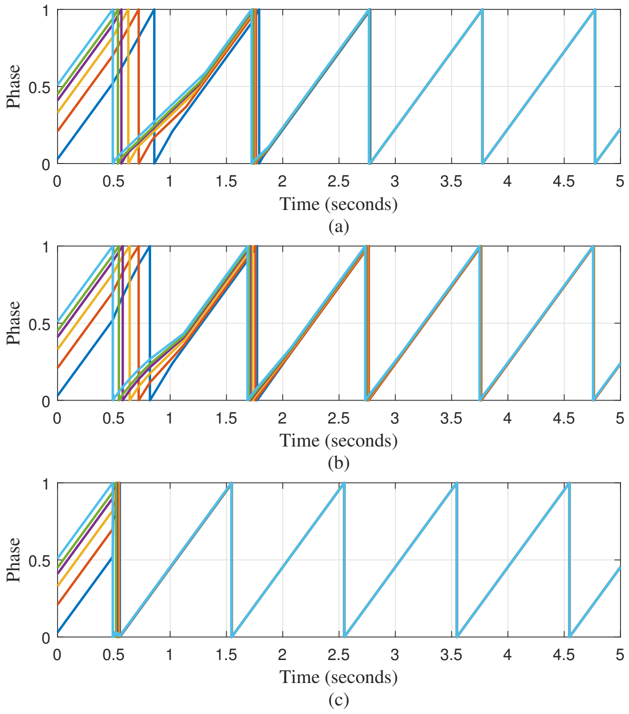
Remark 4. *In the constant frequency method, it is possible to have different amounts of frequency change when increasing or decreasing the oscillator's frequency, i.e., ω_a^+ and ω_a^- respectively. In this paper, we will focus on using the same amount of frequency change $\omega_a = \omega_a^+ = \omega_a^-$ for both increasing and decreasing the frequency of the oscillator.*

Remark 5. *The constant time method described above is a general case of how other algorithms ensure clock continuity. In packet-based synchronization algorithms, the value of τ is set to be the length of the communication period [3], [4].*

Remark 6. *Phase jumps, as is standard in the literature, can be seen as a specific case of either of the above phase continuity methods. These cases can be obtained by either taking the limit as ω_a goes to infinity in the constant frequency method, or by taking the limit as τ goes to zero in the constant time method.*

Remark 7. *Depending on the parameters chosen in each of the above phase continuity methods, the oscillators may evolve backward in phase (i.e., $\omega_i < 0$). Negative frequencies are acceptable in the analysis used in this paper. However, parameters can be chosen to ensure that $\omega_a > -1$ such that the oscillator frequency, ω_i, remains strictly positive.*

Remark 8. *Both methods achieve the same basic result of having the phase evolve continuously. However, each has their desirable characteristics. The constant frequency method only requires the oscillators to evolve at a countable set of frequencies, and the effective coupling between oscillators is maximized as the network approaches synchronization. The constant time method, however, ensures that the phase adjustment occurs in a set amount of time, resulting in more gradual changes in phase as the network synchronizes. The method used should take into consideration the specific application of the PCO network.*



(a)

(b)

(c)

Figure 4. Phase evolution of the PRC synchronization algorithm in [12] for $N = 6$ oscillators in an all-to-all topology, $\alpha = 0.5$, no refractory period, and random initial conditions in a containing arc $\Lambda < \frac{1}{2}$. (a) Continuous phase evolution under the constant frequency method, with $\omega_a = 0.3\omega_0$. (b) Continuous phase evolution under the constant time method, with $\tau = 0.3$ seconds. (c) Phase jumps, for comparison to the phase continuity methods.

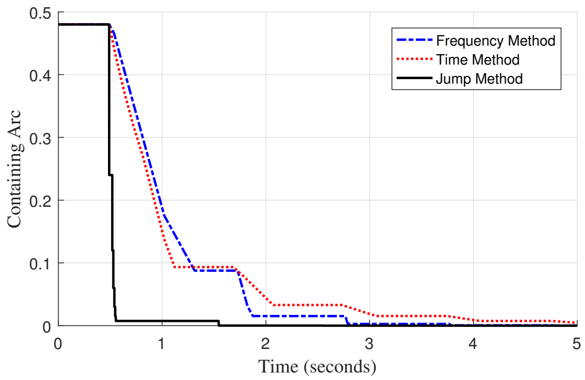
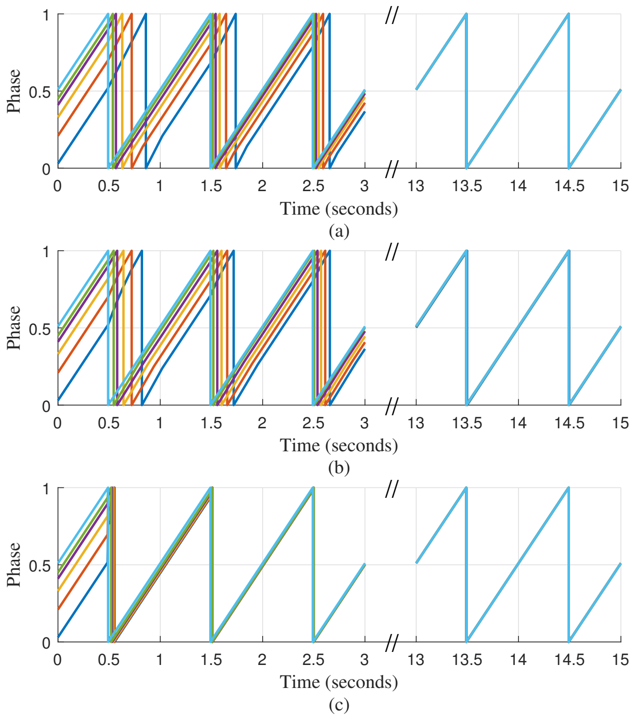


Figure 5. Containing arcs, Λ, as a function of time for the networks in Fig. 4. The convergence speed of the containing arcs under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 0.3$ seconds, is reduced compared with the phase jump case.

Remark 9. *Additional methods besides the ones described above may also be used to achieve continuous oscillator phase evolution. The concept of a time-varying effective coupling strength that is reduced from the network coupling strength is independent of the specific method chosen to ensure phase continuity.*

VI. Simulations and Experimental Results

To test the results concerning phase continuity in pulse-coupled oscillator networks, we perform multiple simulations and physical experiments to observe the behavior of standard PCO networks and synchronization algorithms when phase continuity methods, as in Sec. V, are applied. We first simulate the proposed phase continuity methods in MATLAB, and then test them on a physical PCO network of Raspberry Pi 3 Model B microcomputers.



(a)

(b)

(c)

Figure 6. Phase evolution of the PRC synchronization algorithm in [12] for $N = 6$ oscillators in an all-to-all topology, $\alpha = 0.5$, refractory period $D = 0.5$, and random initial conditions in a containing arc $\Lambda < \frac{1}{2}$. (a) Continuous phase evolution under the constant frequency method, with $\omega_a = 0.3\omega_0$. (b) Continuous phase evolution under the constant time method, with $\tau = 0.3$ seconds. (c) Phase jumps, for comparison to the phase continuity methods.

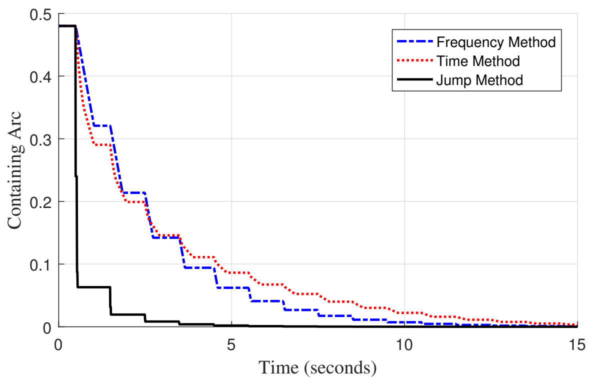


Figure 7. Containing arcs, Λ, as a function of time for the networks in Fig. 6. The convergence speed of the containing arcs under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 0.3$ seconds, is reduced compared with the phase jump case.

A. *PCO Simulations in MATLAB*

We first perform simulations of PCO synchronization algorithms in MATLAB, using the two phase continuity methods as

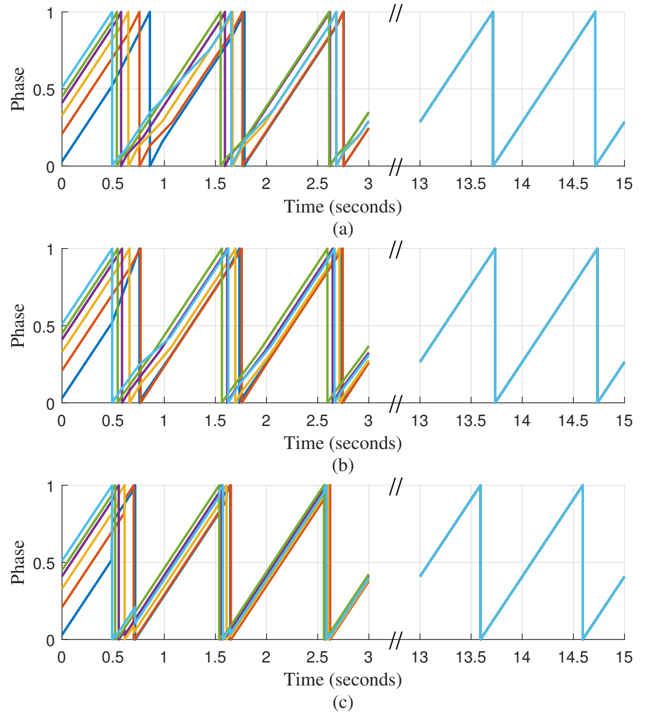


Figure 8. Phase evolution of the PRC synchronization algorithm in [12] for $N = 6$ oscillators in a ring topology, $\alpha = 0.5$, no refractory period, and random initial conditions in a containing arc $\Lambda < \frac{1}{2}$. (a) Continuous phase evolution under the constant frequency method, with $\omega_a = 0.3\omega_0$. (b) Continuous phase evolution under the constant time method, with $\tau = 0.3$ seconds. (c) Phase jumps, for comparison to the phase continuity methods.

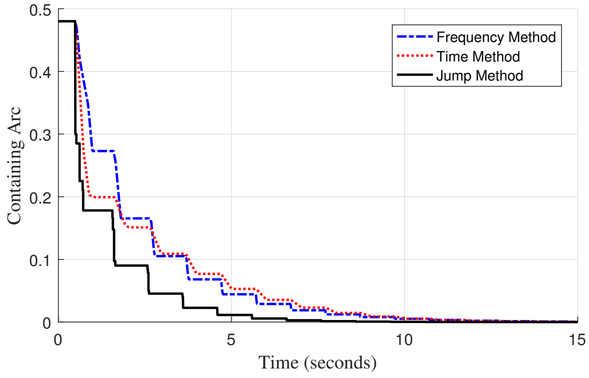


Figure 9. Containing arcs, Λ, as a function of time for the networks in Fig. 8. The convergence speed of the containing arcs under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 0.3$ seconds, is reduced compared with the phase jump case.

discussed in Sec. V. Each oscillators evolves over the interval $[0, 1)$, with fundamental frequency $\omega_0 = 1$, and period of one second.

1) PRC Synchronization: We first use the PRC given in [12] (as shown in Fig. 2 and expressed in (3)) under an all-to-all connection topology. Fig. 4 shows that the network does indeed synchronize under both proposed phase continuity methods from Sec. V. As expected, the network converges more slowly when under the phase continuity methods, due to the reduced effective coupling strengths of the oscillators at most firing instances. The time to synchronization is illustrated in Fig. 5 where the length of the containing arc is plotted as a function of time.

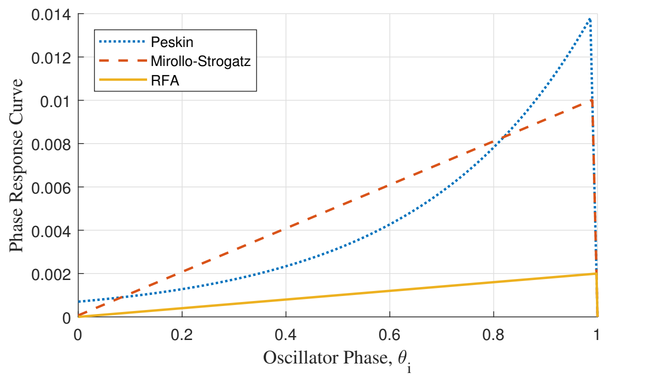


Figure 10. Phase Response Curve (PRC) equivalents for the Peskin and Mirollo-Strogatz synchronization algorithms, and the Reachback Firefly Algorithm (RFA).

As indicated in [12], a refractory period of length D can be incorporated into the PRC, as shown in Fig. 3, and synchronization can still be achieved. We illustrate in Fig. 6 that the network will synchronize using the phase continuity methods in Sec. V. As can be seen in Fig. 7, convergence is slower than when we excluded the refractory period in Fig. 5. Again, the reduced coupling strength of the network due to phase continuity leads to an increased synchronization time compared to the phase jump case.

The PRC in [12] can also achieve synchronization in networks with a more generally connected topology. We illustrate this case in Fig. 8, where we use the general bidirectional ring topology. Again, we see in Fig. 9 that the use of the phase continuity methods in Sec. V still allow the network to synchronize, although the convergence rate is decreased due to the reduced effective coupling strength of the oscillators at most firing instances.

2) Peskin Synchronization Algorithm: We will next consider the original PCO model that was first introduced by Peskin in [7]. He described the oscillators as "integrate-and-fire" oscillators, increasing in phase and firing and resetting their phase when they reached a threshold. The phase of an oscillator is mapped onto a state variable, $x_i(t)$, using the relation $x_i(t) = f(\theta_i(t))$, where $f(\theta)$ is a function that is "smooth, monotonic increasing, and concave down" [8]. When a pulse is received, the oscillator maps its current phase to the state variable, increments the state variable by an amount ϵ, and then maps the state back to the phase using the inverse function $g(x) = f^{-1}(x)$. That is, the new phase of the oscillator can be written as

$$\theta_i^+(t) = g(f(\theta_i(t)) + \epsilon) \tag{11}$$

If the state variable is incremented past the threshold value (i.e., $f(\theta_i(t)) + \epsilon > 1$), then the oscillator immediately fires and resets its phase to zero, and becomes completely synchronized with the oscillator that had fired previously.

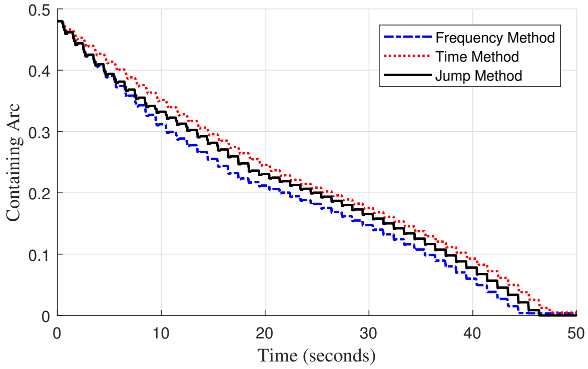


Figure 11. Containing arcs, Λ, as a function of time for the networks under the Peskin synchronization model with parameters $\epsilon = 0.002, \gamma = 3$. The convergence property of the containing arcs under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 0.1$ seconds, is maintained compared to the phase jump case.

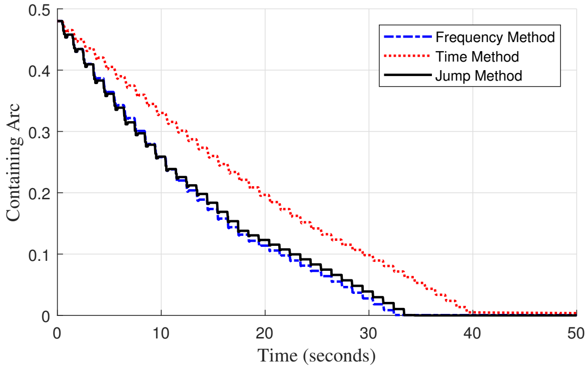


Figure 12. Containing arcs, Λ, as a function of time for the networks under the Mirollo-Strogatz synchronization model with parameters $\epsilon = 0.002, b = 5$. The convergence property of the containing arcs under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 0.1$ seconds, is maintained compared to the phase jump case.

Any function $f(\theta)$ that meets the requirements as above can be used to map the phase into the state variable. Peskin used the following function and its associated inverse:

$$f(\theta) = (1 - e^{-\gamma})(1 - e^{-\gamma\theta}) \tag{12}$$

$$g(x) = \frac{1}{\gamma}\ln(\frac{1 - e^{-\gamma}}{1 - e^{-\gamma} - x}) \tag{13}$$

Mirrolo and Strogatz further improved Peskin's model in [8], and used an alternate function for mapping the phase to the state variable:

$$f(\theta) = \frac{1}{b}\ln(1 + (e^b - 1)\theta) \tag{14}$$

$$g(x) = \frac{e^{bx} - 1}{e^b - 1} \tag{15}$$

Using these functions, an equivalent phase response curve (PRC) can be found by determining ϕ_i from (1). Fig. 10 illustrates these equivalent PRCs.

It is important to note that the Peskin PCO model does not incorporate any kind of coupling strength parameter, α, as in the PRC synchronization given in [12]. This is easily verified by noting that the state variable increment ϵ does not simply scale the equivalent PRC function, but modifies the overall shape of the function. The Peskin model requires that the phases jump the entire amount necessary according to the state mapping function parameters. Equivalently, the Peskin model assumes that the coupling strength α for the network is always 1.

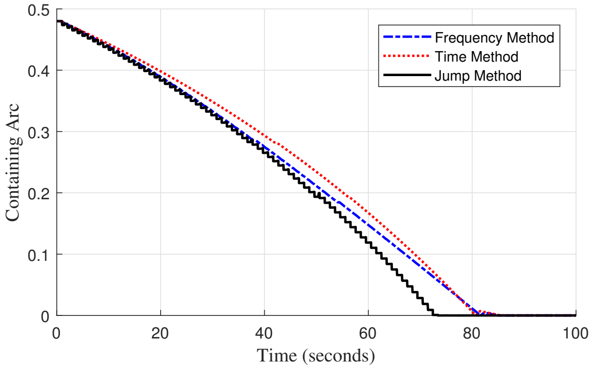


Figure 13. Containing arcs, Λ, as a function of time for the networks under the Reachback Firefly Algorithm (RFA) with parameter $\epsilon = 0.002$. The convergence property of the containing arcs under the constant frequency method, with $\omega_a = 0.007\omega_0$, and the constant time method, with $\tau = 1.1$ seconds, is maintained compared to the phase jump case. The RFA algorithm further allows for phase continuity parameters to be in a broader interval than the Peskin algorithm.

The lack of a coupling strength parameter makes it difficult to extend the analysis of the previous sections to the Peskin model. However, even though there is no coupling strength inherent to the Peskin model, simulations show good synchronization results when the phase continuity methods are applied. For example, using the state variable created from the functions given in (12) and (13), and the equivalent PRC function, we find the amount that the oscillator would jump and apply phase continuity methods from Sec. V. Fig. 11 shows that the phase continuity methods still allow the PCO network to synchronize.

Similarly, using the alternate state variable function introduced by Mirollo and Strogatz given in (14) and (15) also allows the network to synchronize under the phase continuity methods, as shown in Fig. 12.

3) Reachback Firefly Algorithm: Another synchronization algorithm that has been proposed is the Reachback Firefly Algorithm (RFA) by Werner-Allen et. al. in [9]. This algorithm is based on the Peskin PCO model, where the phase is mapped to a state variable. RFA uses a simple mapping function to decrease computational complexity.

$$f(\theta) = \ln(\theta) \tag{16}$$

$$g(x) = e^x \tag{17}$$

Fig. 10 also illustrates the equivalent PRC function for the functions used in the RFA model.

The key difference between the RFA model and the Peskin model is that the oscillators wait to jump until the moment they fire. As the oscillator receives pulses, it records how much it would jump, according to the state variable mapping, at each

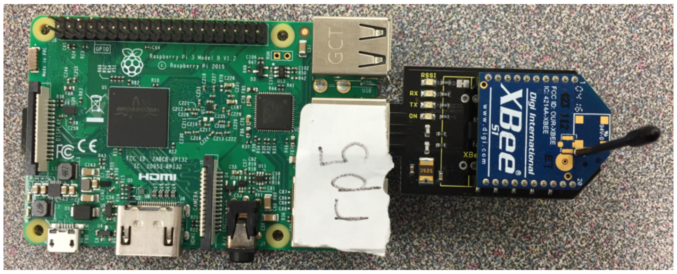


Figure 14. Raspberry Pi 3 Model B and Xbee RF communication module.

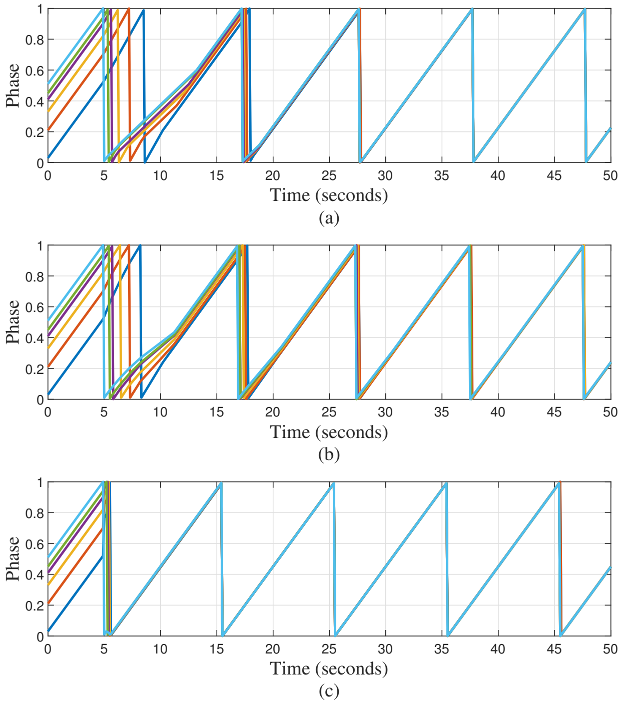
time instance. When the oscillator reaches the threshold and fires, it then adds all of the recorded jump amounts from the previous cycle, and jumps by the total amount. This process is then repeated for each cycle until synchronization is achieved.

As with the Peskin model, there is no inherent coupling strength parameter α in the RFA model. The coupling strength α is assumed to be always 1. This lack of coupling strength incorporated into the RFA model makes it similarly difficult to extend the analysis from the previous sections. However, like with the Peskin model, simulations show good synchronization results when the phase continuity methods are applied. Fig. 13 illustrates that the phase continuity methods in Sec. V allow the PCO network to synchronize. It is also important to note that the phase continuity method parameters can be within a broader interval than for the simpler Peskin model. Since each oscillator does not jump when it receives a pulse, and only when it fires, the effective coupling strength can be maximized more easily than in the Peskin model.

B. PCO Experiments on Raspberry Pi

We next perform physical experiments of PCO synchronization algorithms on a network of Raspberry Pi 3 Model B microcomputers, using the two phase continuity methods discussed in Sec. V. These physical experiments should verify the simulations done in MATLAB, while accounting for non-ideal conditions found in physical PCO networks, such as propagation delay of transmitted pulses and variations in fundamental frequency [10]. Each Raspberry Pi models a single oscillator and sends pulses to other Raspberry Pis in the network using an Xbee RF communication module, as shown in Fig. 14. Each Raspberry Pi runs an identical set of code written in Python, and the phase of each oscillator is based on the internal clock of the Raspberry Pi and evolves over the interval $[0, 1)$.

1) PRC Synchronization: We first use the PRC given in [12] (as shown in Fig. 2 and expressed in (3)) under an all-to-all connection topology. The period of one cycle is set to be ten seconds, which leads to a fundamental frequency $\omega_0 = 0.1$ for each oscillator. We give the oscillators in the network the same initial phase and phase continuity parameters as used in the MATLAB simulations to compare the behavior between the two networks. Fig. 15 shows that the physical network does indeed synchronize. As expected, the network synchronizes more slowly when under the phase continuity methods from Sec. V, due to the reduced effective coupling strengths of the oscillators at most firing instances. The rate of convergence is illustrated in Fig. 16 where the length of the containing arc is plotted as a function of time.



(a)

(b)

(c)

Figure 15. Phase evolution of the PRC synchronization algorithm in [12] for $N = 6$ oscillators in an all-to-all topology, $\alpha = 0.5$, no refractory period, and random initial conditions in a containing arc $\Lambda < \frac{1}{2}$. (a) Continuous phase evolution under the constant frequency method, with $\omega_a = 0.3\omega_0$. (b) Continuous phase evolution under the constant time method, with $\tau = 3$ seconds. (c) Phase jumps, for comparison to the phase continuity methods.

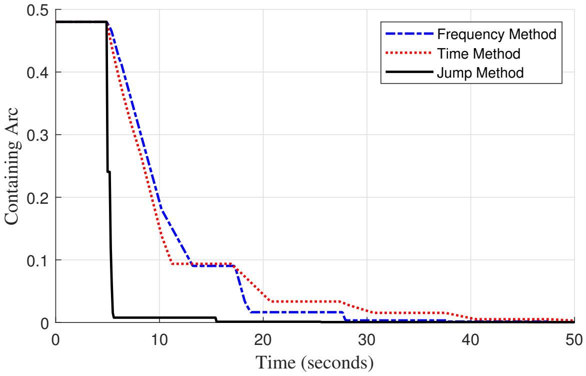
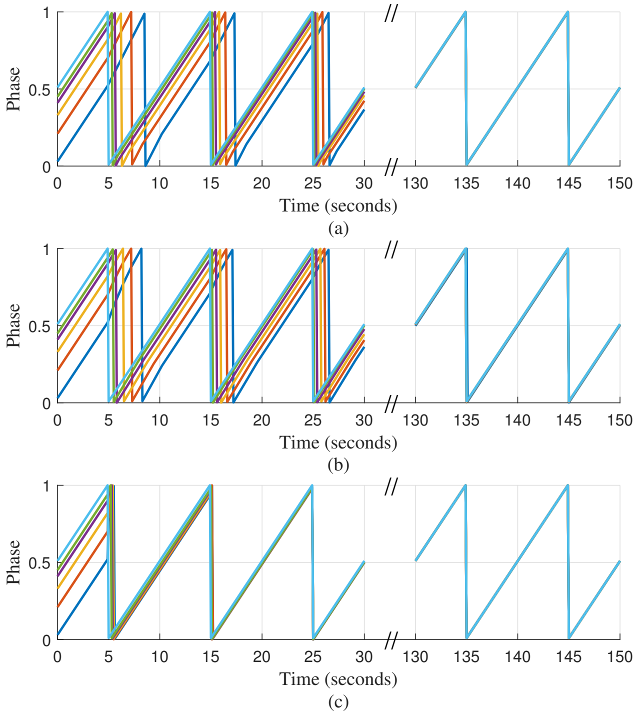


Figure 16. Containing arcs, Λ, as a function of time for the networks in Fig. 15. The convergence speed of the containing arcs under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 3$ seconds, is reduced compared with the phase jump case.

As indicated in [12], a refractory period of length D can be incorporated into the PRC, as shown in Fig. 3, and synchronization can still be achieved. We illustrate in Fig. 17 that the physical network will still synchronize using the phase continuity methods in Sec. V. As can be seen in Fig. 18, convergence is slower than when we excluded the refractory period in Fig. 16. Again, the reduced coupling strength of the network due to phase continuity leads to an increased



(a)

(b)

(c)

Figure 17. Phase evolution of the PRC synchronization algorithm in [12] for $N = 6$ oscillators in an all-to-all topology, $\alpha = 0.5$, refractory period $D = 0.5$, and random initial conditions in a containing arc $\Lambda < \frac{1}{2}$. (a) Continuous phase evolution under the constant frequency method, with $\omega_a = 0.3\omega_0$. (b) Continuous phase evolution under the constant time method, with $\tau = 3$ seconds. (c) Phase jumps, for comparison to the phase continuity methods.

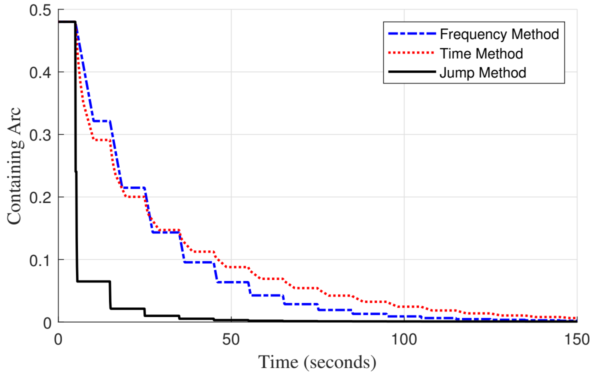


Figure 18. Containing arcs, Λ, as a function of time for the networks in Fig. 17. The convergence speed of the containing arcs under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 3$ seconds, is reduced compared with the phase jump case.

synchronization time compared to the phase jump case.

By comparing the results of the MATLAB simulations with the Raspberry Pi network, we can see that the physical network closely matches the behavior seen in the simulation. The non-ideal conditions inherent in a physical PCO network, such as propagation delay and fundamental frequency variation, do not affect the synchronization behavior of the networks.

2) Peskin Synchronization Algorithm: We next reconsider the Peskin synchronization model, and perform experiments to see how the physical PCO network responds under non-ideal conditions and phase continuity. We use the same initial phase values and parameters for the Peskin state variable and phase update functions as used in the MATLAB simulations to compare the response of both networks. That is, the oscillators evolve over the interval [0,1), with fundamental frequency $\omega_0 = 1$, and period of one second.

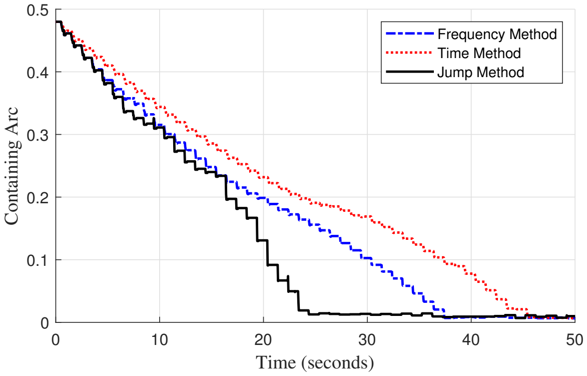


Figure 19. Containing arcs, Λ, as a function of time for the networks under the Peskin synchronization model with parameters $\epsilon = 0.002, \gamma = 3$. The synchronization of the network under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 0.1$ seconds, is achieved more slowly when compared to the phase jump case.

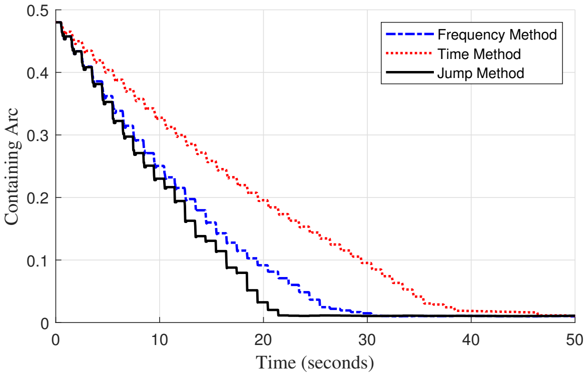


Figure 20. Containing arcs, Λ, as a function of time for the networks under the Mirollo-Strogatz synchronization model with parameters $\epsilon = 0.002, b = 5$. The synchronization of the network under the constant frequency method, with $\omega_a = 0.3\omega_0$, and the constant time method, with $\tau = 0.1$ seconds, is achieved more slowly when compared to the phase jump case.

Again, we see that we get good synchronization results under phase continuity, even though the synchronization algorithm does not incorporate a coupling strength parameter, as discussed previously. Fig. 19 shows that the phase continuity methods still allow the physical PCO network to synchronize. Similarly, using the alternate state variable function introduced by Mirollo and Strogatz also allows the network to synchronize under the phase continuity methods, as shown in Fig. 20.

An important difference between the synchronization of the simulations and the physical PCO network under the Peskin synchronization algorithm should be addressed. Specifically, the jump case in the MATLAB simulation synchronizes much more slowly than in the physical network, although oscillators under the phase continuity methods synchronize in comparable amounts of time. This behavior is due to the propagation delay of the transmitted pulse between oscillators. Ideally, once two oscillators synchronize, they maintain the same phase over time and form a single oscillator group. Any oscillators in an oscillator group will fire together, forming a single transmitted pulse, resulting in fewer pulses sent per cycle until all oscillators are synchronized, eventually resulting in one pulse per cycle. However, in a physical PCO network, these oscillator groups cannot form perfectly, so each oscillator receives the same number of pulses in each cycle, allowing them to synchronize more quickly. For oscillators operating under phase continuity, the increase in the number of pulses received is countered by a decrease in the effective coupling strength.

VII. Conclusions

In this paper, we consider the problem of synchronizing clocks while guaranteeing time continuity. To do so, we utilize and analyze the behavior of pulse-coupled oscillator networks under the constraint of phase continuity. Original PCO network models use discontinuous phase jumps, but sharp discontinuities in the phase variable of the oscillators is usually not desirable. We present a generalization of the standard PCO model in which the phase variable of an oscillator must evolve in a continuous manner.

Using this phase continuity modification, we show that the behavior of the network under these restraints can be modeled as a time-varying coupling strength in the network. Specifically, the coupling strength may be effectively reduced at firing instances. We mathematically prove that a pulse-coupled oscillator network will synchronize with a time-varying coupling strength using a delay-advance phase response curve. Additionally, we find that the reduced coupling results in a decrease in convergence speed. Overall, PCO networks under various synchronization algorithms can still achieve desirable behavior using continuous evolutions in the phase variable.

The Peskin and RFA models for pulse-coupled oscillators do not include a coupling strength parameter. Further research may be desirable to consider the effects of having a reduced, and possibly time-varying, coupling strength in the Peskin and RFA models.

Other phase continuity modifications besides the one proposed in this paper are possible. For example, oscillator frequency adjustments can be aggregated if multiple pulses are heard before the desired phase change is complete, resulting in a combined response to multiple firings, rather than a response based on the most recent pulse. Such a design will require separate analysis to ensure the desired convergence properties. General analysis of the myriad of other PCO algorithms under phase continuity and time-varying coupling strength reveals much research potential.

VIII. Acknowledgment

We would like to thank those who reviewed the initial drafts of this paper.

References

[1] L. Schenato and F. Fiorentin. Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica*, 47(9):1878 – 1886, 2011.

[2] L. Lamport and P. M. Melliar-Smith. Synchronizing clocks in the presence of faults. *Journal of the Association for Computing Machinery*, 32(1):52–78, January 1985.

[3] T. K. Srikanth and S. Toueg. Optimal clock synchronization. *Journal of the Association for Computing Machinery*, 34(3):626–645, July 1987.

[4] M. Mock, R. Frings, E. Nett, and S. Trikaliotis. Continuous clock synchronization in wireless real-time applications. In *Proceedings of the 19th IEEE Symposium on Reliable Distributed Systems*, SRDS '00, pages 125–, Washington, DC, USA, 2000. IEEE Computer Society.

[5] M. Ryu, J. Park, and S. Hong. Timing constraint remapping to achieve time equi-continuity in distributed real-time systems. *IEEE Transactions on Computers*, 50(12):1310–1320, Dec 2001.

[6] B. Sundararaman, U. Buy, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: A survey. *Ad Hoc Networks 3*, pages 281–323, 2005.

[7] C. S. Peskin. *Mathematical aspects of heart physiology*. Courant Institute of Mathematical Science, New York University, New York, 1975.

[8] R. Mirollo and S. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, 1990.

[9] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 142–153, New York, NY, USA, 2005. ACM.

[10] Y.W. Hong and A. Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE Journal on Selected Areas in Communications*, 23(5):1085–1099, May 2005.

[11] Y. Q. Wang and F. J. Doyle III. Optimal phase response functions for fast pulse-coupled synchronization in wireless sensor networks. *IEEE Transactions on Signal Processing*, 60(10):5583–5588, Oct 2012.

[12] Y. Q. Wang, F. Núñez, and F. J. Doyle III. Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening. *IEEE Transactions on Signal Processing*, 60(10):5293–5306, Oct 2012.

[13] F. Dörfler and F. Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539 – 1564, 2014.

[14] J. Nishimura. Frequency adjustment and synchrony in networks of delayed pulse-coupled oscillators. *Physical Review E*, 91:012916, Jan 2015.

[15] F. Núñez, Y. Q. Wang, and F. J. Doyle III. Synchronization of pulse-coupled oscillators on (strongly) connected graphs. *IEEE Transactions on Automatic Control*, 60(6):1710–1715, June 2015.

[16] F. Núñez, Y. Q. Wang, and F. J. Doyle III. Global synchronization of pulse-coupled oscillators interacting on cycle graphs. *Automatica*, 52:202–209, 2015.

[17] R. Gentz, A. Scaglione, L. Ferrari, and Y. W. P. Hong. Pulsess: A pulse-coupled synchronization and scheduling protocol for clustered wireless sensor networks. *IEEE Internet of Things Journal*, PP(99):1–13, 2016.

[18] J. Klinglmayr, C. Bettstetter, M. Timme, and C. Kirst. Convergence of self-organizing pulse-coupled oscillator synchronization in dynamic networks. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2016.

[19] G. Brandner, U. Schilcher, and C. Bettstetter. Firefly synchronization with phase rate equalization and its experimental analysis in wireless systems. *Computer Networks*, 97:74 – 87, 2016.

[20] A. V. Proskurnikov and M. Cao. Synchronization of pulse-coupled oscillators and clocks under minimal connectivity assumptions. *IEEE Transactions on Automatic Control*, 62(11):5873–5879, Nov 2017.

[21] C. Godsil and G. Royle. *Algebraic Graph Theory*, volume 207. New York: Springer, 2001.