Beyond Sync: Distributed Temporal Coordination and Its Implementation in a Multi-Robot System

Agata Barciś
Karl Popper Kolleg on
Networked Autonomous Aerial Vehicles
University of Klagenfurt
Klagenfurt, Austria
Email: agata.barcis@aau.at

Christian Bettstetter
Institute of Networked and Embedded Systems
University of Klagenfurt
Klagenfurt, Austria
Email: christian.bettstetter@aau.at

***Abstract*—We propose a technique for adaptive temporal coordination in multi-agent systems where tasks have to be scheduled in a decentralized way. It provides three states: synchronized, splay, and clustered. Due to discretization in time and phase, the model operates with low signaling effort and is robust in the presence of delays. We analyze the model through simulations and demonstrate its feasibility through experiments with small robots.**

***Index Terms*—Synchronization, de-synchronization, temporal coordination, discrete coupled oscillators, self-organization, swarm robotics.**

I. Introduction

A. Motivation

Multi-agent systems often require agents to perform actions in a simultaneous way. For instance, a swarm of robots performing cooperative stereo vision needs to be synchronized to achieve good depth resolution. Centralized approaches are not always applicable, which is why distributed algorithms for mutual synchronization have been developed. Examples include extensions and modifications of well-known models for self-synchronizing coupled oscillators, such as the Kuramoto model [1] and the Mirollo-Strogatz model [2]. However, in some cases, synchronization to the same oscillator phase is not required and even undesirable, but the agents should be in *anti-phase*. For example, during a surveillance mission, aerial robots should fly to a charging station at different times to avoid congestion. This problem is often solved by planning algorithms, although self-organizing solutions offer advantages in terms of robustness, flexibility, and adaptability (e.g., agents can join and leave the mission at any time).

In some applications, an even more complex temporal behavior than anti-phase holds interest. For example, two aerial robots cooperatively delivering a package should simultaneously take off but avoid flying into the warehouse at the time when another group is inside. Time coordination can also be used indirectly to manage the behavior of agents. For instance, splitting of an aerial swarm flying in formation into phase-dependent clusters could facilitate obstacle avoidance [3].

This work was supported by the Karl Popper Kolleg on Networked Autonomous Aerial Vehicles at the University of Klagenfurt and by the Austrian Science Fund (FWF) under grant P30012-NBL.

In conclusion, there is a need for coordination models that enable the distributed control of timing and the creation of temporal patterns, and we propose and evaluate such a model in this paper. Our approach is based on the well-known concept of coupled oscillators but also takes into account the practical particularities for implementation in technical systems, such as signaling overhead and delays.

Robots are often connected with a wireless network that is used by multiple programs running on each robot. Temporal coordination is only one of these programs. An efficient temporal coordination method should therefore lead to a low signaling overhead. Models based on permanent information exchange (e.g., utilizing continuous phase coupling) are not applicable in practice for such systems.

Wireless communication always introduces delay between the instants when the message was sent and received. Whenever a robot receives a phase transmitted by another oscillator, it will actually receive an outdated value from an unknown moment in the past. In addition, the more devices that share the same medium, the higher the communication delays and usually jitter (variation of delays). Hence, a temporal coordination model robust against such variable delays would facilitate the operation of large groups of robots.

B. Related work

Coupled oscillators are extensively studied from the perspective of theory (e.g., [4], [5]) and engineering (e.g., sensor networks [6], radio communications [7], and robotics [8]). The ultimate goal in these papers is always to achieve synchronization, i.e., to align the phases of all oscillators. The opposite — repulsive coupling of oscillators — is rarely studied. There is some work on this topic, albeit it is mostly theoretical and focuses on the coexistence of repulsive and attractive coupling (see [9], [10]). There is also work on applications, e.g., repulsive coupling between clusters of nodes in a network [11].

A generalization of oscillator coupling allowing the formation of splay or cluster states has been analyzed in theory (see [12], [13]) and sometimes with potential applications in technology (e.g., multi-agent systems [14] and networks with different topologies [15]). Cluster and splay states are often considered as unwanted equilibria; therefore their stability is extensively examined for both continuous coupling [4] and pulse coupling [16]–[18].

The task of phase discretization has also been considered in the literature, although the models are either limited to an arbitrary number of phase levels [19] or they focus on stochastic interactions between oscillators [20]. Additionally, these approaches only regard continuous coupling.

Coupled oscillators are also employed in distributed control algorithms, where not only synchronization but also splay and cluster states hold interest [21]–[23]. Many approaches focus on multi-agent systems, they take into account connectivity (see [24], [25]) or propose discrete time models to permit the exchange of messages [26]. There is also some work undertaken to cope with delayed messages [27].

C. Contributions and paper structure

To the best of our knowledge, there is no coupled oscillator model that enables the formation of synchronized, splay, and cluster states while also being robust to delays. We contribute to this topic by introducing a unified temporal coordination model that is discrete in both time and phase and enables us to obtain synchronized, splay, and cluster states. Furthermore, we present a way to put the system off unwanted equilibria, we verify the model in simulation, and finally demonstrate its feasibility in real-world experiments using small mobile robots.

The remainder of this paper is structured as follows. Section II introduces the temporal coordination model. Section III contains the simulation-based analysis. Section IV presents experiments with robots, and finally Section V concludes.

II. Temporal Coordination Model

A. Phase representation

The phase of the j-th oscillator is a sum of two parts:

$$\Phi_j = \phi_j + \hat{\theta}_j \; , \tag{1}$$

where $\Phi_j \in [0, 2\pi)$. The term $\phi_j \in [0, \frac{2\pi}{L})$ is the oscillatory part of the phase, which we call the *internal clock*. The term $\hat{\theta}_j = \frac{\theta_j}{L} 2\pi$ is a discrete phase value, with *phase level* $\theta_j \in \{0, 1, \dots, L-1\}$, where L is the number of phase levels. The proposed phase model is presented in Figure 1.

Whenever ϕ_j resets, the phase level is incremented by 1 and a signal communicating the value of θ_j is sent. The signals are marked with arrows in Figure 1. Different ways to encode θ_j are possible: for example, it can be a single number in a data message or be represented by the frequency of a simple radio, audio, or light signal that otherwise contains no data.

This paper focuses on how the oscillators mutually influence their phase levels, rather than how their internal clocks are synchronized. It is assumed that the oscillators have equal natural frequencies ω. Their internal clocks need to be synchronized, which can be achieved by established techniques. In this work we incorporate firefly synchronization to provide common time steps $\Delta T = \frac{1}{L\omega}$ for updating the phase levels. We use this method for synchronization because it is distributed and works well on simple devices. In principle, other synchronization techniques can also be used.

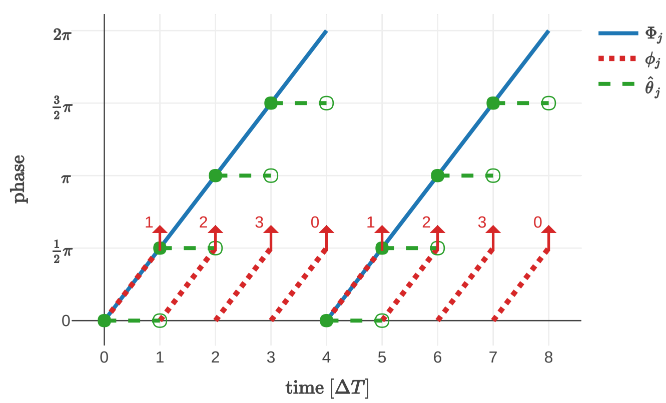


Fig. 1: Phase of an oscillator Φ_j with its components $\hat{\theta}_j$ and ϕ_j over normalized time. Arrows show the moment a signal is emitted; the numbers next to them represent the phase level.

B. Phase patterns and their applications

Different applications require different temporal arrangements. We are interested in three of them: the *synchronized state*, with all oscillators having the same phase level; the *splay state*, with oscillators evenly spaced in the phase domain; and the *cluster state*, with oscillators forming evenly-spaced clusters of equal size in the phase domain. Note that even though phase levels differ in some states, internal clocks are not influenced and remain synchronized.

For a given number of oscillators N, the three states can be described in a unified way by defining the number of clusters M that oscillators should form: $M = 1$ for the synchronized state, $M = N$ for the splay state, and $1 < M < N$ for the cluster state. We consider only states with clusters of equal size, i.e., $M \mid N$ (M divides N). Furthermore, we assume that $M \,|\, L$, which allows maintaining equal distances between clusters. We call these states *M-cluster states*.

The proposed model can be generalized, whereby each oscillator can maintain multiple phase levels with one internal clock. The multiple phase levels of a single oscillator should not influence each other but the corresponding levels of other oscillators. With this approach, a single oscillator can be a part of multiple patterns at the same time. There should always be only one internal clock, so all signals will be sent at the same time, and thus they can be merged in one message.

Let us outline potential applications, one for each state:

1) *Synchronized states* can be used for agents that act simultaneously. For example, robots take pictures at the same moment to assure that the objects are at the same place in all of the pictures, even if the scene is dynamic. Later these pictures can be used for image stitching or reconstruction of three-dimensional scenes.
2) *Splay states* can be used for agents that perform asynchronous actions. For example, robots send measurements in bundles to a base station in different time slots to avoid interference. Each robot transmits when its phase level is equal to 0.

3) *Cluster states* can be used to split agents into groups performing actions asynchronously with other groups. For example, based on the phase levels, robots split into M groups (those with the same level are in one group). They can only exchange messages (e.g. pictures or measurements) when their phase level is 0 to avoid intergroup interference.

C. Order parameters

A measure of synchrony is the complex *order parameter* [1]:

$$re^{\mathrm{i}\Psi} = \frac{1}{N}\sum_{j=1}^{N} e^{\mathrm{i}\hat{\theta}_j}, \tag{2}$$

where r is the radius of the centroid of the phases and Ψ is its angle. The radius reaches its maximum value ($r = 1$) in a synchronized state, i.e., when all oscillators have equal phases. When oscillators achieve a *balanced state*, their phase centroid coincides with the point $(0, 0)$ on the complex plane, and thus the radius reaches its minimal value ($r = 0$).

Splay and cluster states are special cases of a balanced state. In order to distinguish them, we use the order parameters of higher phase harmonics [23]:

$$r_m e^{\mathrm{i}\Psi_m} = \frac{1}{Nm}\sum_{j=1}^{N} e^{\mathrm{i}m\hat{\theta}_j}, \tag{3}$$

with $m \in \mathbb{N}$ and $m \leq M$, where the term r_m is the radius of the centroid of the m-th harmonic of phases and Ψ_m is its angle. Similar to the first order parameter, if $r_m = \frac{1}{m}$, the m-th phase harmonic is synchronized. In the balanced state, its centroid is $r_m = 0$. In order to achieve the desired number of clusters, the first $M-1$ harmonics must be in a balanced state and the M-th harmonic must be in synchrony ($r_M = \frac{1}{M}$).

D. State potential

Based on the order parameter, we define the potential U_m of the m-th phase harmonic similarly as in [23]:

$$U_m = \frac{N}{2} K_m r_m^2, \tag{4}$$

where K_m is the coupling strength of the m-th harmonic.

The potential has its minimum in different states: if $K_m < 0$, the potential is minimized if the m-th phase harmonic is synchronized ($r_m = \frac{1}{m}$). For $K_m > 0$, it reaches its minimum if the m-th phase harmonic is in a balanced state ($r_m = 0$).

The sum of potentials of M harmonics describes the potential of the M-cluster state [23]:

$$U^{(M)} = \sum_{m=1}^{M} U_m. \tag{5}$$

The first $M - 1$ phase harmonics must be in a balanced state and the M-th one must be synchronized. This leads to the following conditions for the coupling strength: $K_m > 0$ for $m < M$ and $K_M < 0$. In order to achieve the desired M-cluster state, $U^{(M)}$ needs to be minimized. It reaches its global minimum if each U_m is minimized, and this state corresponds to the M-cluster state, as proven in [23].

E. Phase control

Gradient control can be used to minimize $U^{(M)}$ of the system of oscillators with continuous phase [23]:

$$\dot{\Phi}_j = \omega - \frac{1}{N}\frac{\partial U^{(M)}}{\partial \Phi_j}. \tag{6}$$

It results in the coupling function Γ, being a linear combination of continuous coupling similar to the Kuramoto model for the first M phase harmonics [23]:

$$\frac{\partial U^{(M)}}{\partial \Phi_j} = \sum_{k=1}^{N} \Gamma(\Phi_{kj}) = \sum_{k=1}^{N} \frac{K_m}{m}\sin(m\Phi_{kj}), \tag{7}$$

where the phase difference between two oscillators is represented by $\Phi_{kj} = \Phi_k - \Phi_j$. In a similar way, differences of other variables are represented, e.g., $\theta_{kj} = \theta_k - \theta_j$.

For a system with discrete phase and discrete time, we propose the following model based on the same coupling function (evaluated for discrete values of $\hat{\theta}_{kj}$): the phase correction of the j-th oscillator at time instant t is

$$\delta\theta_j[t] = \delta\theta_j[t-1] + \frac{1}{N}\sum_{j=1}^{N} \Gamma(\hat{\theta}_{kj}[t]). \tag{8}$$

F. Common time frame

In order to enable the desired coordination, we need to assure a common time frame for all agents. To achieve this, each agent maintains an internal clock. Each time that the phase ϕ_j reaches the threshold $\frac{2\pi}{L}$, the j-th agent executes three actions: first, it updates its phase level $\theta_j \mod L$, by incrementing it by 1 and applying phase correction rounded towards zero:

$$\theta_j[t+1] = (\theta_j[t] + 1 + \mathrm{sgn}(\delta\theta_j[t]) \cdot \lfloor|\delta\theta_j[t]|\rfloor) \mod L; \tag{9}$$

second, the phase correction is reset $\delta\theta_j[t] = 0$, if the rounded correction was different than 0 (phase level was changed); and third, the oscillator emits the signal containing its updated phase level $\theta_j[t + 1]$.

In order to assure that all updates take place simultaneously and each oscillator receives the data about others in time (before the update), internal clocks ϕ are synchronized using the firefly synchronization algorithm. In systems of pulse coupled oscillators, two types of coupling can be used, namely excitatory or inhibitory. If excitatory coupling is applied, the oscillator's phase is slightly increased each time that it receives a pulse. Inhibitory coupling triggers the opposite reaction, whereby the phase is slightly decreased.

In case of temporal coordination, if excitatory coupling were used for each received signal, the oscillator that received an even slightly delayed message would jump forward. Although eventually its internal clock would catch up with the rest, it would cause the change in its phase level (relative to the other oscillators). This might break the desired pattern. In order to avoid this, each oscillator employs inhibitory coupling to adjust its internal clock in the first half of the oscillation cycle and excitatory in the second half.

In order to make the synchronization of the internal clocks more resistant to delays, we use a refractory period [28]. During this time, the oscillator does not react to signals; the phase of the internal clock will not change even if the signal is received. Despite this, the received value of the phase level is still stored and used for temporal coordination.

The synchronized signals provide a common time frame to update and apply phase corrections, assuring that all oscillators perform it simultaneously. Since the internal clocks might slightly differ and delays might occur, the signals containing current phase levels of other oscillators can be received in a time frame before or after it emits its own signal. In order to compensate for this, the states of other oscillators are gathered in the time window $[T-0.5\Delta T, T+0.5\Delta T)$, where T is the moment when the oscillator reaches the k-th phase level and emits its signal. The reason for choosing this interval is that in this window the oscillator is attracted by the k-th phase level (received signals bring it closer to level k while synchronizing the internal clock). Once all of the signals are gathered, in the time instant $T+0.5\Delta T$, the new phase correction $\delta\theta$ is calculated and can be used to update the phase level at the end of the oscillation cycle.

This approach makes the time coordination resistant to delays up to $\frac{\Delta T}{2} - \tau$, where τ is the precision of synchronization. The delays exceeding this threshold might have negative influence on time coordination, as the data contained in the message might be outdated. Furthermore, the model requires that the communication is reliable in such a way that there are only a few message losses.

G. Equilibrium states

A continuous system with the coupling function Γ defined in (7) has multiple equilibrium points. Only a subset of them is stable and corresponds to a desired M-cluster state [23]. Other equilibria are unstable in systems with a continuous phase. For instance, a synchronized state is an equilibrium for splay and cluster states, as in this state all phase differences are 0, implying $\Gamma(\hat{\theta}_{kj}) = 0 \ \forall j, k$. Note that stable equilibria points coinciding with discrete phase levels exist if $M \mid L$. Hence, it is possible to achieve an M-cluster state in a discretized system.

The phase discretization strongly increases the probability of the system being in one of the equilibria. For example, consider a system with $L = 4$ phase levels and $N = 2$ designated to reach synchrony. If the oscillators start with random phase levels, there is a 25 % chance that they will start exactly in anti-phase ($\theta_j = \theta_k + \frac{L}{2}$), which is an equilibrium for the synchronized state.

The goal of the discretization of the temporal coordination is to prevent the influence of disruptions (e.g., communication delays, oscillators imperfections or noise). If the system is in equilibrium, all interactions cancel out. As a result, there is no influence that can unbalance the system. Thus, in order to break the unwanted clusters and equilibria, we introduce noise η and *energy* of state E.

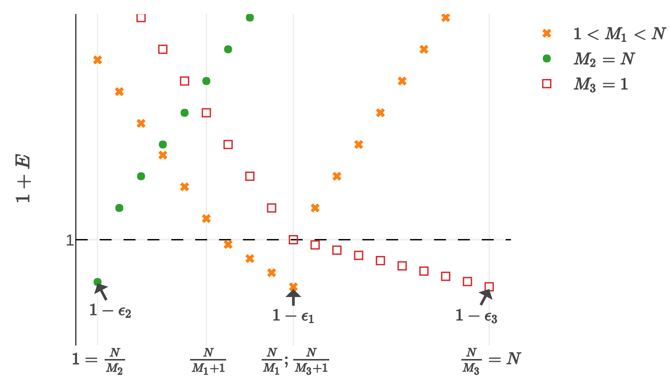


Fig. 2: Shape of energy function for different sizes of clusters. $-\epsilon_j$ denotes minimal energy.

If two oscillators have the same phase level, they are equally influenced by others and will thus always change their phase levels in the same moment, even if they should be separated to achieve the desired state. The role of noise is to slightly variate interactions between oscillators. In the described example, noise can cause the oscillators to change their phase levels in different time steps.

The energy of the state has two tasks, namely stabilizing the expected state and breaking unwanted clusters or equilibria. First, it suppresses phase correction (originating from both noise and interactions between oscillators) of the oscillators that create clusters of size N_C meeting the condition $\frac{N}{M+1} < N_C \leq \frac{N}{M}$. Accordingly, oscillators that form clusters of proper size (or almost proper if there are still too few oscillators in the cluster) are more reluctant to change their phase levels. Therefore, noise and some small deviations of the system's state from the desired one will not cause them to change their phase levels. Second, if the cluster is too large, the energy amplifies the phase correction, causing the cluster to break. If the system is in an unstable equilibrium, only noise influences phase correction. In this case, energy amplifies the noise, making it easier for oscillators to leave this state. Additionally, oscillators in significantly too small clusters ($N_C \leq \frac{N}{M+1}$) are more "eager" to change their phase level and "look for" a larger cluster. This property is not crucial for achieving the desired state but speeds up this process. Examples of energy functions for different cluster sizes are presented in Figure 2.

The energy is used to either suppress or amplify the previous phase correction. The modified equation of the phase update considers noise and energy:

$$\delta\theta_j[t] = (1+E)\cdot\delta\theta_j[t-1] + \eta + \frac{1}{N}\sum_{k=1}^{N}\Gamma(\hat{\theta}_{kj}[t]). \quad (10)$$

H. Constraints

In order to assure that energy and noise have only a limited impact on the system state, we introduce three constraints:

1) *Energy*: Phase correction suppression should never be too strong. Even minimal interactions with other oscillators should eventually lead to changing the phase level. We assume that the oscillator's phase level is incorrect if it is influenced by at least minimal non-zero force of at least $0.5N$ other agents. In other words, the largest correction loss, caused by energy, occurs when correction approaches 1. Even at this point, the loss should be lower than half of the minimal non-zero influence. Thus, we can formulate the energy constraint as:

$$(1-\epsilon)+\frac{1}{2}\min_{\substack{\hat{\theta}_{kj},\\ \Gamma(\hat{\theta}_{kj})\neq 0}}\left|\Gamma(\hat{\theta}_{kj})\right|>1, \tag{11}$$

where ϵ is *energy margin*, and $-\epsilon$ is the minimum value of energy, as marked in Figure 2.

2) *Noise*: If the oscillator is in the desired state (size of its cluster is proper), the energy should suppress phase correction sufficiently strongly to prevail noise. As a result, noise cannot cause an oscillator to change its phase level if it is not subject to other influences and the cluster is of proper size. We can formulate it as follows:

$$(1-\epsilon)+\eta<1. \tag{12}$$

3) *Number of phase levels*: We consider only the number of phase levels, such that $M \mid L$. Moreover, the condition $L \geq 3M$ needs to be fulfilled. Otherwise, for $L = M$ and $L = 2M$, the coupling of the M-th harmonic is only sampled at the points where its value is 0.

Based on the constraints (11) and (12) we can calculate the minimal energy ϵ and the maximal value of a random noise. Moreover, we can determine the number of phase levels needed to form the desired M-cluster state.

III. Simulation-Based Analysis

A. Setup

The temporal coordination model is tested with a simulation implemented in *Python*. We focus on the evaluation of temporal pattern formation with a discretized model.

All oscillators start with uniformly distributed random phases Φ (both components ϕ and $\hat{\theta}$ are random) and without any phase correction $\delta\theta = 0$. During our tests, the coupling strength $K = 0.25$ proved to be the most versatile for different parameter sets; therefore this value is used in all presented results. Based on this value, we calculate $K_m = \frac{L}{2\pi} \cdot K$ for $m < M$ and $K_M = -0.1 \cdot \frac{L}{2\pi} \cdot K$.

A scaling factor $\frac{L}{2\pi}$ is used to speed up the convergence. It assures that if the influence is very high (e.g. $\sin(\theta_{kj}) = 1$), the attraction will be sufficiently high for the oscillator to change its level by more than 1. It is especially useful if the number of levels is high. For example, if $L = 1000$, $M = 1$, $N = 2$ and in the initial state $\theta_1 = \theta_2 + 250$, each oscillator will have to change its phase level by 125. If the $\max_{\theta_{kj}} \Gamma(\theta_{kj}) = 1$, they would need at least 125 oscillation cycles to meet.

Phenomena related to communication and imperfections of oscillators are not taken into account. Agents have full connectivity without delays or packet losses. This leads to the full knowledge about phase levels of other oscillators and perfect synchronization of the internal clocks.

B. Results

For each experiment we present a plot showing how the potential $U^{(M)}$ converges over time. We shift its value by $\frac{K_M N}{2M^2}$ and normalize it, whereby it thus always converges to 0 and has a maximum value of 1, independent of the parameters. Additionally, we show the phase plots at the beginning, in the middle, and at the end of the simulation. The values of the potential in these moments are marked with squares in the potential plot. In each phase plot, the possible phase levels are numbered on the angular axis and marked with the lines of corresponding color. The clusters are marked with circles filled with the color of their level. The size of the circles and the numbers next to them indicate the size of clusters.

Figures 3 to 6 show the simulation results for different parameters, presenting M-cluster states for $M \in \{1, 2, 4, 8\}$. We notice that for $M = 1$ it takes a long time until the first change in the potential appears. This is caused by the initial distribution of phase levels, which is close to the balanced state. In this case, the interactions between the oscillators are weak, therefore more oscillation cycles are needed to change this state. This behavior cannot be observed for $M > 1$ because higher harmonics of phase levels are not balanced in the initial state.

For all patterns, the closer the system gets to the desired state, the less dynamic the changes, given that the interactions between oscillators become weaker when the system approaches the stable state.

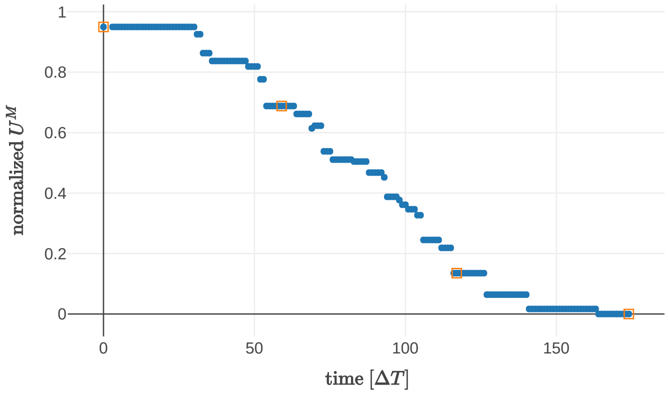
The gradient control used in the continuous model would cause the potential to decrease. In some of the potential plots (Figures 4a and 5a), occasional growth of potential can be observed. This can be caused by energy and noise that breaks an unwanted cluster or equilibrium state. In this case, the change in one oscillator's phase level is caused entirely by noise. Another reason for a potential growth can be the discretization of the model. For example, a local minimum of the j-th dimension of continuous potential can be located between two phase levels. In this case, the j-th oscillator changes its phase level, moving in the direction of the local minimum, which in fact causes it to jump to the other side of the minimum.

In both scenarios, the temporary increase of the potential can either cause the system to find an optimal way toward the global minimum of potential, otherwise the oscillator returns to the previous state if the last change was not desired.

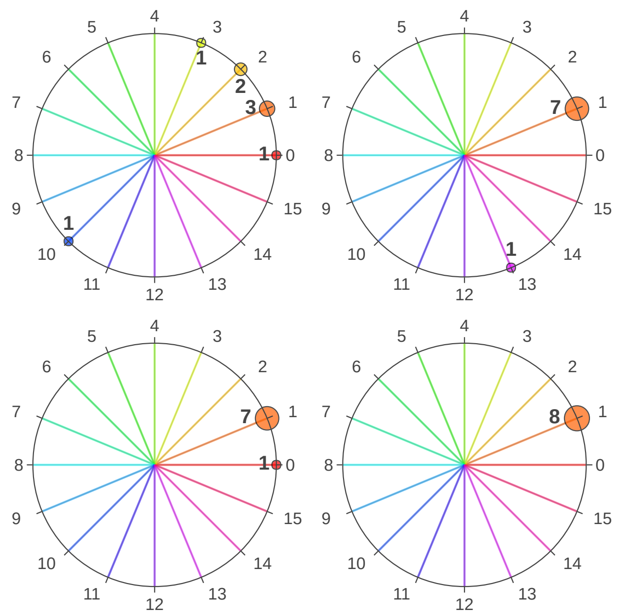
IV. Experiments with Robots

A. Setup

For a real-world evaluation of the temporal coordination model, we use a robot swarm platform based on Pololu Balboa robots. The main computer is a Raspberry Pi 3B+. It exchanges messages with other robots through a wireless network, computes the results of the model, and visualizes its

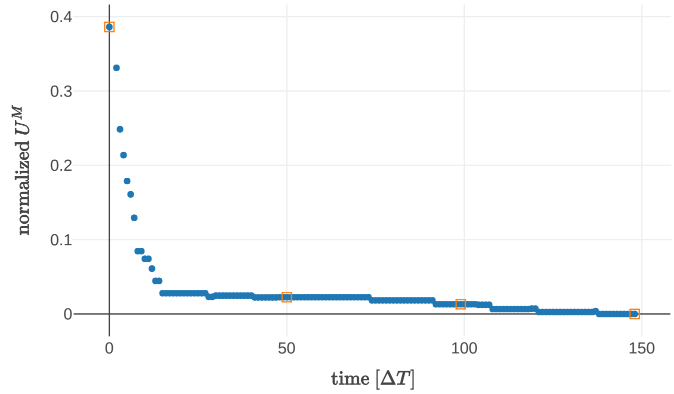


(a) Potential convergence

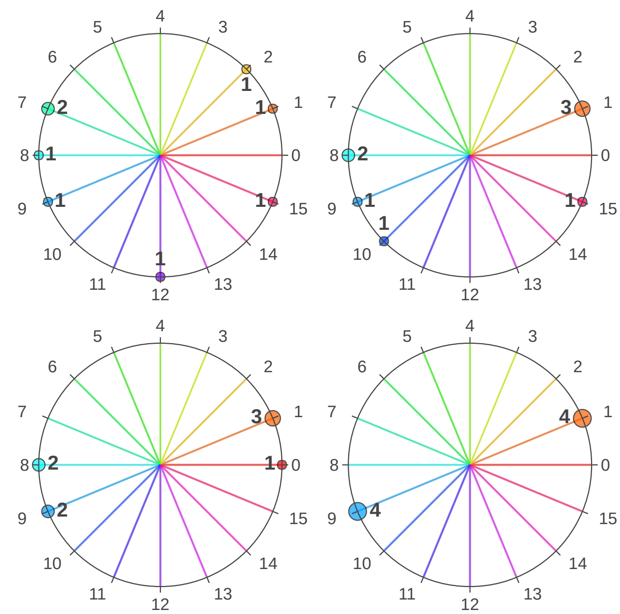


(b) History of phase level changes. The plots are ordered as follows: left top, right top, left bottom, right bottom.

Fig. 3: M-cluster state for $N = 8$, $L = 16$, $M = 1$.



(a) Potential convergence



(b) History of phase level changes. The plots are ordered as follows: left top, right top, left bottom, right bottom.

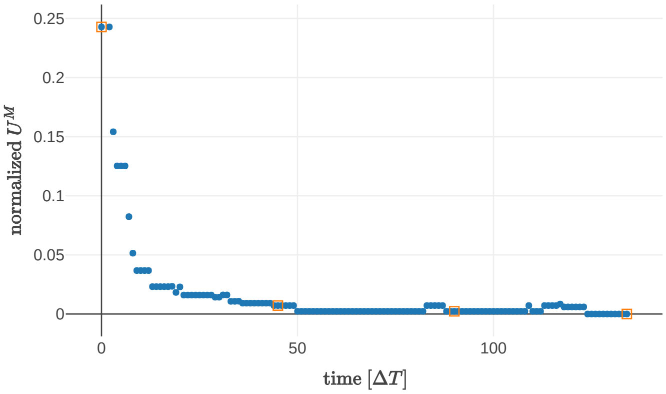
Fig. 4: M-cluster state for $N = 8$, $L = 16$, $M = 2$.

phase levels using an RGBW light-emitting diode (LED) strip attached to the bumpers of the robot. A framework based on Ansible is used for deployment, starting the experiments, and downloading the logged data. The robot swarm platform and deployment framework designed for multi-robot systems were introduced in [29].

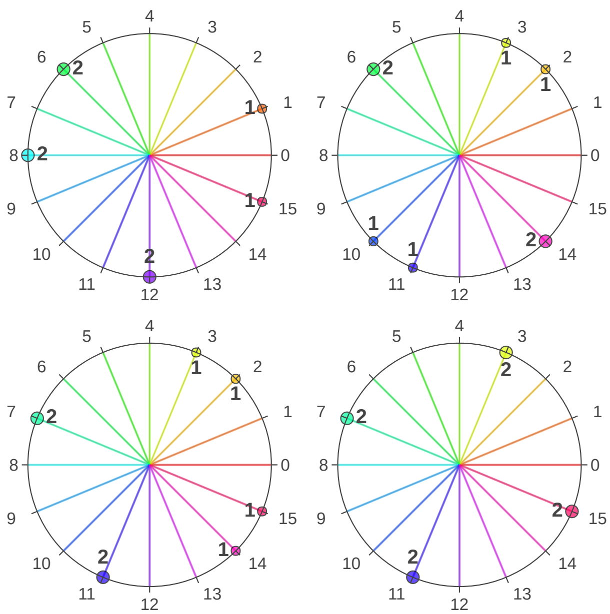
Robots communicate via an IEEE 802.11g network operating in ad-hoc mode, which enables them to join and leave the group without any infrastructure. The network is set up using the built-in wireless card of the Raspberry Pi 3B+. From the software perspective, the communication is realized in the ROS 2 framework (Bouncy Bolson release) using the Data Distribution Service (DDS) communication standard, which applies a Real-Time Publish Subscribe (RTPS) protocol. In this work, we use eProsima Fast RTPS. The robots are configured with the best-effort mode with multicast enabled to reduce communication load.

The robots use the described communication only to exchange their phase levels. A robot receives messages from all robots (its own messages are filtered out, thus not taken into account, neither for internal clock synchronization nor for time coordination). The signaling effort depends on the natural frequency of oscillations. In our experiments, we use $\omega = \frac{1}{L}\mathrm{s}^{-1}$. This means that each robot sends only one message per second.

Moreover, the robots know neither the total number of agents nor the number of agents with the same phase level. For an update of phase correction and energy, the number of the messages received during the last oscillation cycle was assumed to be the number of oscillators and the data contained

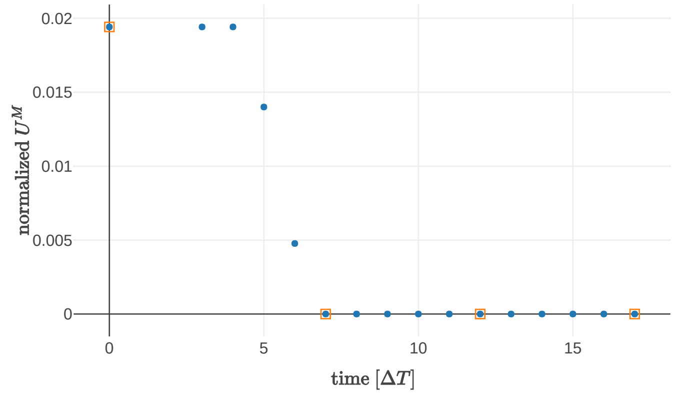


(a) Potential convergence

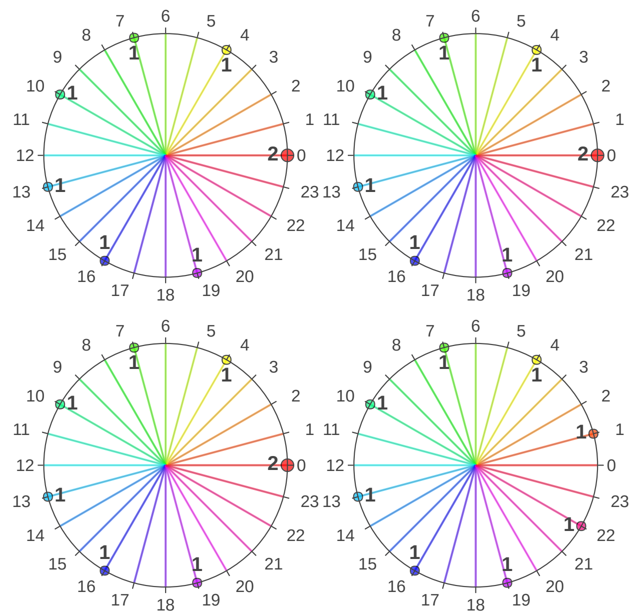


(b) History of phase level changes. The plots are ordered as follows: left top, right top, left bottom, right bottom.

Fig. 5: M-cluster state for $N = 12$, $L = 16$, $M = 4$.



(a) Potential convergence



(b) History of phase level changes. The plots are ordered as follows: left top, right top, left bottom, right bottom.

Fig. 6: M-cluster state for $N = 8$, $L = 24$, $M = 8$.

in these messages was used to count the number of agents with the same phase level. Therefore, if many messages are delayed or missing, it can have an impact on the convergence and stability of the desired pattern.

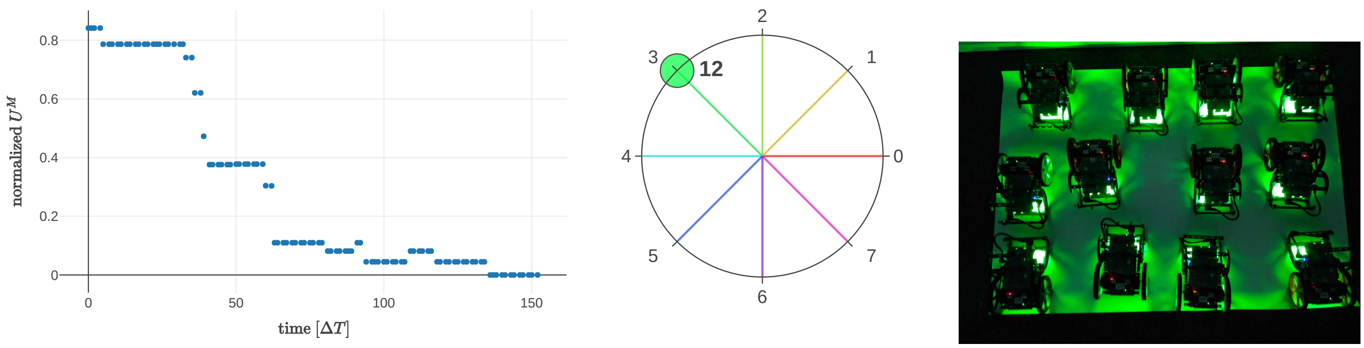
Although mobility is not exploited in this work, the use of a robot swarm is motivated by the possibility of testing temporal coordination in an environment similar to the one that it will be applied in, i.e., with the same main computer and communication capabilities. Additionally, the robots will allow us to make demonstration of the model visually appealing, for example, by representing internal clock synchronization with a swinging movement of robots, similar to our video about firefly synchronization of a robot swarm [30].

B. Results

The experimental results are shown in Figures 7 to 9. They are presented in a similar way as the simulation results aside from the fact that the phase plot is only shown for the final state, and a snapshot of the robots with lights representing their phase level is given for each pattern.

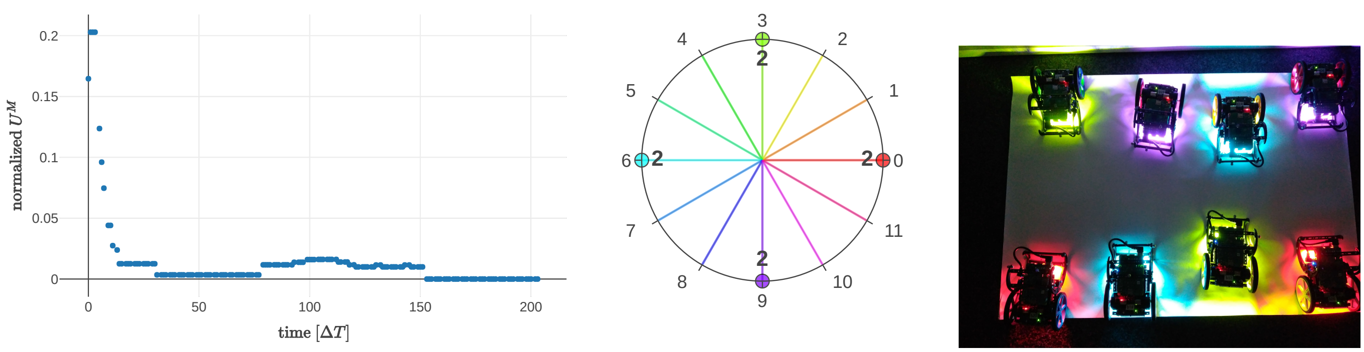
Again, we observe an occasional increase of the potential, now due to packet losses in addition to previously described reasons. Although short disruptions of potential occur, the system recovers and converges to the desired state.

During the experiment, we measure the accuracy of internal clock synchronization, defined as a maximum difference between their phases. Although synchronization of the internal clocks is not the focus of this work, we use its poor precision to present the link quality and show that the time coordination



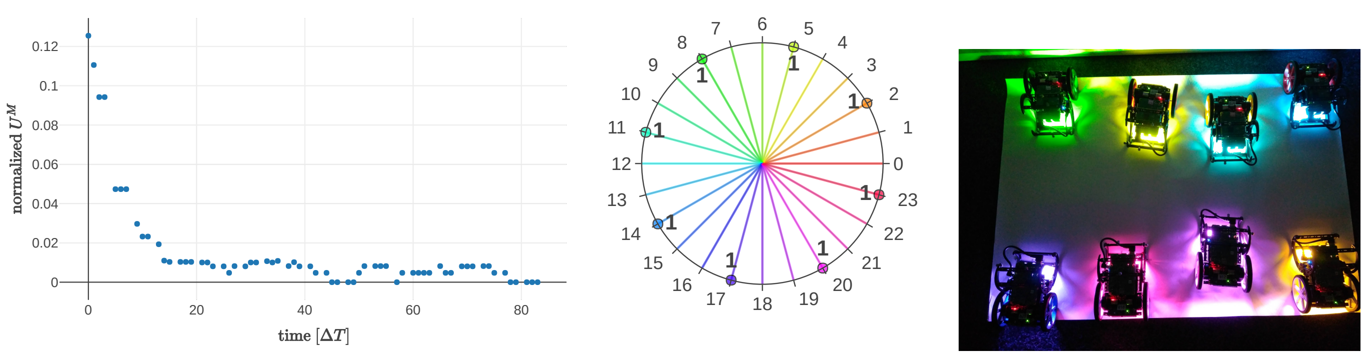
(a) Potential convergence (b) Phase plot (c) Picture of robots

Fig. 7: Experiment results: M-cluster state for $N = 12$, $L = 8$, $M = 1$.



(a) Potential convergence (b) Phase plot (c) Picture of robots

Fig. 8: Experiment results: M-cluster state for $N = 8$, $L = 12$, $M = 4$.



(a) Potential convergence (b) Phase plot (c) Picture of robots

Fig. 9: Experiment results: M-cluster state for $N = 8$, $L = 24$, $M = 8$.

model does not require precise synchronization. The results show that the accuracy sometimes drops to $40\,\mathrm{ms}$, which suggests that the link quality varies over time and significant delays occur. Despite these disturbances, our temporal coordination model leads to the desired states.

V. Conclusions and Outlook

We have introduced a distributed and adaptive model for temporal coordination in multi-agent systems and demonstrated that three intended states emerge: synchronized, splay, and cluster states. There are many uses for these patterns in multi-robot systems, e.g., to share a communication medium or split robots into groups for task assignment. The model was tested with mobile robots, showing that time and phase discretization enables the system to converge to a desired state even in the presence of delays.

Future work may involve optimizing of the convergence time for specific patterns and further reducing the number of messages that need to be exchanged to establish and maintain temporal coordination. The model will be used on aerial robots to support the coordination required in 3D reconstruction tasks.

References

[1] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*. Mineola, N.Y: Dover Publications, Aug. 2003.

[2] R. Mirollo and S. Strogatz, “Synchronization of pulse-coupled biological oscillators,” *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, Dec. 1990.

[3] B. D. O. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx, “Rigid graph control architectures for autonomous formations,” *IEEE Control Systems Magazine*, vol. 28, no. 6, pp. 48–63, Dec. 2008.

[4] S. H. Strogatz, “From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators,” *Physica D: Nonlinear Phenomena*, vol. 143, no. 1, pp. 1–20, Sep. 2000.

[5] F. Dörfler and F. Bullo, “Synchronization in complex networks of phase oscillators: A survey,” *Automatica*, vol. 50, no. 6, pp. 1539–1564, Jun. 2014.

[6] 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*, vol. 23, no. 5, pp. 1085–1099, 2005.

[7] G. Brandner, U. Schilcher, and C. Bettstetter, “Firefly synchronization with phase rate correction and its experimental analysis in wireless systems,” *Computer Networks*, vol. 97, pp. 74–87, Mar. 2016.

[8] F. Perez-Diaz, R. Zillmer, and R. Groß, “Firefly-inspired synchronization in swarms of mobile agents,” in *Proc. International Conference on Autonomous Agents and Multiagent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 279–286.

[9] H. Hong and S. H. Strogatz, “Kuramoto model of coupled oscillators with positive and negative coupling parameters: An example of conformist and contrarian oscillators,” *Physical Review Letters*, vol. 106, no. 5, Feb. 2011.

[10] ——, “Mean-field behavior in coupled oscillators with attractive and repulsive interactions,” *Physical Review E*, vol. 85, no. 5, May 2012.

[11] L.-x. Yang, X.-l. Lin, and J. Jiang, “Influences of adding negative couplings between cliques of Kuramoto-like oscillators,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 6, Jun. 2018.

[12] K. Okuda, “Variety and generality of clustering in globally coupled oscillators,” *Physica D: Nonlinear Phenomena*, vol. 63, no. 3, pp. 424–436, Mar. 1993.

[13] F. Ferrante and Y. Wang, “A hybrid systems approach to splay state stabilization of pulse coupled oscillators,” in *Proc. IEEE Conference on Decision and Control*, 2016, pp. 1763–1768.

[14] W. Xia and M. Cao, “Cluster synchronization algorithms,” in *Proc. American Control Conference*, 2010, pp. 6513–6518.

[15] C.-U. Choe, T. Dahms, P. Hövel, and E. Schöll, “Controlling synchrony by delay coupling in networks: From in-phase to splay and cluster states,” *Physical Review E*, vol. 81, no. 2, Feb. 2010.

[16] B. Chen, J. R. Engelbrecht, and R. Mirollo, “Cluster synchronization in networks of identical oscillators with α-function pulse coupling,” *Physical Review E*, vol. 95, no. 2, Feb. 2017.

[17] R. Zillmer, R. Livi, A. Politi, and A. Torcini, “Stability of the splay state in pulse-coupled networks,” *Physical Review E*, vol. 76, no. 4, Oct. 2007.

[18] F. Ferrante and Y. Wang, “Robust almost global splay state stabilization of pulse coupled oscillators,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3083–3090, Jun. 2017.

[19] T. Prager, B. Naundorf, and L. Schimansky-Geier, “Coupled three-state oscillators,” *Physica A: Statistical Mechanics and its Applications*, vol. 325, no. 1, pp. 176–185, Jul. 2003.

[20] D. J. Jörg, “Stochastic Kuramoto oscillators with discrete phase states,” *Physical Review E*, vol. 96, no. 3, Sep. 2017.

[21] D. A. Paley, N. E. Leonard, and R. Sepulchre, “Oscillator models and collective motion: Splay state stabilization of self-propelled particles,” in *Proc. IEEE Conference on Decision and Control*, Dec. 2005, pp. 3935–3940.

[22] D. A. Paley, N. E. Leonard, R. Sepulchre, D. Grunbaum, and J. K. Parrish, “Oscillator models and collective motion,” *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 89–105, Aug. 2007.

[23] R. Sepulchre, D. A. Paley, and N. E. Leonard, “Stabilization of planar collective motion: All-to-all communication,” *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 811–824, May 2007.

[24] D. J. Klein, P. Lee, K. A. Morgansen, and T. Javidi, “Integration of communication and control using discrete time Kuramoto models for multivehicle coordination over broadcast networks,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 695–705, May 2008.

[25] R. Sepulchre, D. A. Paley, and N. E. Leonard, “Stabilization of planar collective motion with limited communication,” *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 706–719, Apr. 2008.

[26] D. J. Klein and K. A. Morgansen, “Set stability of phase-coupled agents in discrete time,” in *Proc. American Control Conference*, Jun. 2008, pp. 2285–2290.

[27] B. I. Triplett, D. J. Klein, and K. A. Morgansen, “Discrete time Kuramoto models with delay,” in *Proc. Networked Embedded Sensing and Control: Workshop*, ser. Lecture Notes in Control and Information Science, P. J. Antsaklis and P. Tabuada, Eds. Springer Berlin Heidelberg, 2006, pp. 9–23.

[28] Y. Kuramoto, “Collective synchronization of pulse-coupled oscillators and excitable units,” *Physica D: Nonlinear Phenomena*, vol. 50, no. 1, pp. 15–30, May 1991.

[29] A. Gniewek, M. Barciś, and C. Bettstetter, “Robots that sync and swarm: A proof of concept in ROS 2,” *arXiv:1903.06440*, Mar. 2019.

[30] ——, “Video: Firefly synchronization of a robot swarm,” https://www.youtube.com/watch?v=BfPEWH4ASXE, 2018.