Discrete-Time Synchronization on the N-Torus

Alain Sarlette, Rodolphe Sepulchre, and Naomi Ehrich Leonard

***Abstract*— In this paper, we study the behavior of a discrete-time network of N agents, each evolving on the circle. The global convergence analysis on the N-torus is a distinctive feature of the present work with respect to previous synchronization results that have focused on convergence in the Euclidean space $(\mathbb{R}^n)^N$. We address the question from a control perspective, but make several connections with existing models, including the Hopfield network, the Vicsek model and the (continuous-time) Kuramoto model.**

We propose two different distributed algorithms. The first one achieves convergence to equilibria in shape space that are the local extrema of a potential U_L built on the graph Laplacian associated to a fixed, undirected interconnection topology; it can be implemented with sensor-based interaction only, since each agent just relies on the relative position of its neighbors. The second one achieves synchronization under varying and/or directed communication topology using local estimates of a consensus variable that are communicated between interacting agents. Both algorithms are based on the notion of centroid and can be interpreted as descent algorithms. The proposed approach can be extended to other embedded compact manifolds.

***Keywords*— Coordinated control, Synchronization**

I. INTRODUCTION

Swarm control is a topic of growing interest among scientists and engineers. Indeed, several current applications involve the cooperation of many identical agents in order to perform a global task; among others, these include formation flight [1], ocean exploration [2] and some space applications [3],[4],[5]. In addition, many complex natural systems are composed of many similar individuals that are interacting [6],[7],[8]; our understanding of the phenomena that are observed in those systems should benefit from the study of the basic mechanisms involved in distributed systems.

In the present paper, we focus on the (almost-)global distributed stabilization of orientation variables in discrete-time. The global convergence analysis on the N-torus is a distinctive feature of the present work with respect to previous synchronization results [9],[10],[11],[12],[13] that have focused on convergence in the Euclidean space $(\mathbb{R}^n)^N$, considering the present problem either by local linearization or by restriction of the initial conditions to a convex subspace.

We start by making some specific observations on a class of consensus algorithms in the plane ($\mathbb{C}^N$), which can then be extended to the embedded torus ($\mathbb{T}^N$). This provides a first class of discrete-time descent algorithms (*Algorithm A*) that lead to analogies with several existing models, explicitly the Vicsek model (discrete-time on $SE(2)^N/SE(2)$), the Hopfield model (discrete-time on the N-hypercube) and the Kuramoto model (continuous-time on $\mathbb{T}^N/\mathbb{T}^1$). Convergence is established when the communication topology is fixed and undirected. The algorithm is based on a local centroid computation and asynchronous operation, extending to synchronous operation for "sufficiently small moves".

A second control strategy (*Algorithm B*) for which global convergence properties can be established is presented on the same basis of local centroid computation. Using an idea recently proposed in [14], Algorithm B is similar to Algorithm A but uses a somewhat increased exchange of information among communicating agents: in addition to its relative position, each agent communicates its current estimate of a consensus variable.

The paper is organized as follows. In Section II, we make some observations about consensus algorithms in the Euclidean space $\mathbb{C}$. This summary of existing results paves the way for the corresponding algorithms that we develop on the torus. Algorithm A is presented in Section III, along with a discussion of the various connections with existing models. Section IV presents Algorithm B, achieving global synchronization under directed and switching communication with the help of an estimator variable. We conclude with some perspectives for future research in Section V.

II. CONSENSUS ALGORITHMS IN EUCLIDEAN SPACE

We consider a swarm of N identical agents evolving in the (complex) plane; the state of each agent k is the number $z_k \in \mathbb{C}$. The agents interact through links that may be undirected or directed, static or varying as a function of time or system state. The interconnection topology is specified by means of a directed graph G on N vertices, containing the edge (l, k) if agent l sends information to ($\equiv$ is a neighbor of) agent k, which is denoted $l \sim k$; the number $d_k^{(o)}$ of agents l such that $k \sim l$ is called the out-degree of k and the number $d_k^{(i)}$ of agents l such that

This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. The scientific responsibility rests with its authors. A. Sarlette holds a FNRS fellowship (Belgian Fund for Scientific Research).

A. Sarlette and R. Sepulchre are with the Department of Electrical Engineering and Computer Science (Montefiore Institute), University of Liege, 4000 Liege, Belgium. alain.sarlette@ulg.ac.be, r.sepulchre@ulg.ac.be

N.E. Leonard is with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton NJ 08544, USA. naomi@princeton.edu

$l \sim k$ is called the in-degree of k. The swarm has no leader and no external reference. The goal is to reach and stabilize certain particular formations (*i.e.* configurations that are invariant modulo a rigid-body transformation) using a control law

$$z_k[t+1] = z_k[t] + u_k(\{z_l[t] - z_k[t] : l \sim k\}) , \quad k = 1, 2, ..., N . \tag{1}$$

A particular form of (1) considers

$$p_k = \operatorname*{av}_{l\sim k}(z_l) = \frac{\sum_{l\sim k} z_l}{d_k^{(i)}} ,$$

the centroid of the neighbors of agent k, as the point to reach. This is motivated by the objective to gather all the agents at a single point (the so-called "consensus value"). The fraction of the distance towards p_k that is covered over a single iteration can be imposed by including the position of agent k itself with a certain weight in the average. This leads to the particular family of control laws

$$z_k[t+1] = \varepsilon_k \operatorname*{av}_{l\sim k}(z_l[t]) + (1 - \varepsilon_k) z_k[t] , \quad \varepsilon_k \in (0, 1),\ k = 1, 2, ..., N . \tag{2}$$

A. MAIN CONVERGENCE PROPERTY

A number of authors have recently studied the properties of systems comprising (2) as a particular case. Convergence results are available that allow time-varying and directed interconnection graphs [10],[11],[13] and may be summarized by the following proposition adapted from [10].

Proposition 1: *Considering a sequence of directed graphs $G[t]$ with edge set $\mathcal{E}[t]$, we say that agent l is connected to agent k across a time interval I if there is a directed path from l to k in the graph containing the edges appearing in $\bigcup_{t\in I} \mathcal{E}[t]$.*

- *If there is $T \geq 0$ such that for all $t_0 \in \mathbb{N}$ there is an agent connected to all other agents across $[t_0,\ t_0{+}T]$, or*
- *if all communication graphs are bidirectional and for all $t_0 \in \mathbb{N}$ there is an agent connected to all other agents across $[t_0,\ +\infty)$,*

then any solution of (2) converges to $z_k = c_0\ \forall k$ when $t \to +\infty$.

B. FIXED UNDIRECTED GRAPH

We further examine the situation of a fixed, undirected interconnection topology; in that case, $d_k^{(i)} = d_k^{(o)} = d_k$. Following [15], we consider an artificial potential U_L built on the Laplacian of the graph G,

$$U_L(z) = \frac{1}{2N}\langle z|Lz\rangle \tag{3}$$

$$= \frac{1}{2N}\|B^T z\|^2 . \tag{4}$$

In (3), $\langle v_1|v_2\rangle$ denotes the scalar product $[\bar{v}_1]^T[v_2]$ and z denotes the vector containing all z_k 's. The equivalent expression (4) uses the incidence matrix B of G. This matrix has one row per agent and one column per communication link and is defined as follows: if link e connects agent k to agent l, then $B_{ke} = -1$ and $B_{le} = 1$ (for undirected graphs, which one of the two agents takes the minus sign is irrelevant); $B_{me} = 0\ \forall m \notin \{k, l\}$. The incidence matrix is related to the Laplacian by $BB^T = L$. The use of the potential (3) is motivated by the following properties.

(i) For any connected communication topology, $U_L = 0$ if and only if $z_k = z_1\ \forall k$. The global minimum of U_L thus selects the synchronized state.

(ii) According to (4), U_L may be interpreted as the sum of the squared lengths of the links between connected agents. In that sense, U_L will be small for a "grouped" swarm and large for a "dispersed" swarm.

(iii) Expression (4) shows that the potential U_L can be computed from relative positions that are available through the communication topology defined by G.

(iv) The symmetry of L for undirected graphs ensures that the k-th component of the gradient $\nabla_z U_L$ can be computed from the relative positions between agent k and its neighbors.

We observe that (2) is strictly equivalent to

$$z_k[t+1] = z_k[t] - \frac{N\varepsilon_k}{d_k}\frac{\partial U_L}{\partial z_k} . \tag{5}$$

This observation has the following implications.

Descent property: For any $0 < \varepsilon_k < 1$, (2) is a descent algorithm for U_L (*i.e.* $U_L(z[t+1]) \leq U_L(z[t])$); this is true for synchronous operation (all agents are updated at time t), as well as any type of partially asynchronous operation (only the agents belonging to some subset of $\{1, 2, ...N\}$ are updated at time t).

Continuous-time limit: When $\varepsilon_k \to 0$, (2) turns out to be an Euler discretization of the continuous-time system

$$\dot{z}(t) = -KLz(t) \tag{6}$$

where K is a positive diagonal gain matrix. System (6) is a gradient descent algorithm for $\langle z|Lz\rangle$ with respect to the (diagonal) metric $\langle z|K^{-1}z\rangle$, as studied in *e.g.* [9].

III. CONSENSUS ALGORITHMS ON THE N-TORUS

When studying *e.g.* the synchronization of headings in the plane, the system evolves on the N-torus, which can be embedded in $\mathbb{C}^N$. Our goal is to present some properties of (2) that can still be considered when the state space is reduced to the circle $|z_k| = 1$. The state of each agent k then reduces to the angle $\theta_k \in S^1$, and the analog of (1) is the state equation

$$\theta_k[t+1] = \theta_k[t] + u_k(\{\theta_l[t] - \theta_k[t] : l \sim k\}) , \quad k = 1, 2, ..., N . \tag{7}$$

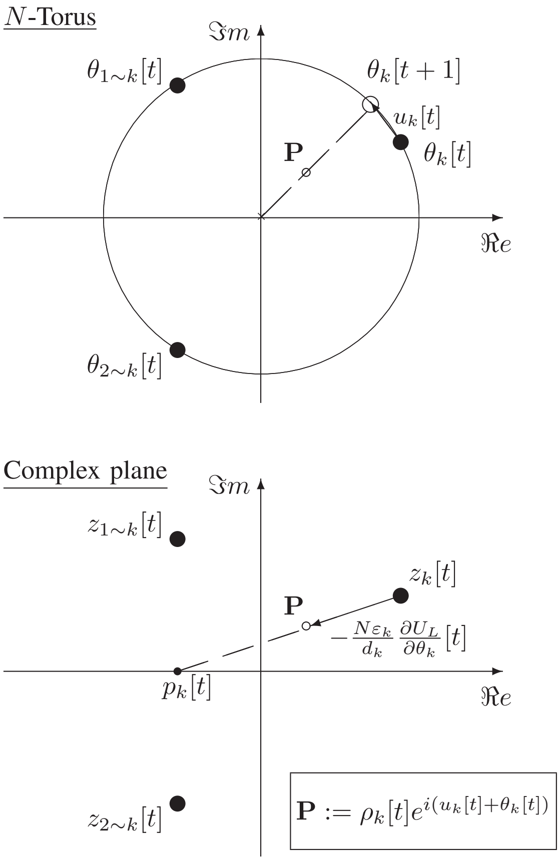

Fig. 1. Interpretation of (9) as the projection of (5) onto the state space of agent k.

For the rest we replace z_k by $e^{i\theta_k}$. That is, for a fixed, undirected communication topology, we first build the potential

$$\begin{aligned} U_L(\theta) &= \frac{1}{2N}\langle e^{i\theta}|Le^{i\theta}\rangle \\ &= \frac{1}{2N}\left(\sum_{l=1}^{N} d_l - \sum_{l=1}^{N}\sum_{m\sim l}\cos(\theta_m-\theta_l)\right) . \quad (8) \end{aligned}$$

According to (8), the symmetry of L ensures that the k-th component of the gradient $\nabla_\theta U_L$ can still be computed from the relative positions between agent k and its neighbors. Then we consider the same centroid p_k and the update law

$$\begin{aligned} \theta_k[t+1] &= \arg\left(\varepsilon_k \operatorname*{av}_{l\sim k}(z_l[t]) + (1-\varepsilon_k)z_k[t]\right) \quad (9) \\ &=: \arg\left(\rho_k[t]e^{i(u_k[t]+\theta_k[t])}\right) , \\ &\qquad \varepsilon_k \in (0,1),\ k = 1,2,...,N . \end{aligned}$$

This amounts to projecting the point reached after the gradient descent (5) onto the unit circle, as depicted on Figure 1. In case $\rho_k = 0$, the angle u_k is not defined and we allow agent k to choose any position on the circle.

Locally, the dynamics (9) can be mapped onto the real line, leading to an Euclidean algorithm of the type (2), with $z_k \in \mathbb{R}$. This has been suggested *e.g.* in [10] as a way to use the results of the previous section on the circle, allowing for convergence analysis in the case of directed and varying communication topologies. The obtained convergence results however are not global: when working out more than a local stability analysis, they restrict the set of initial conditions to half a circle.

Connection with the Vicsek model: We note that the control law (9) is directly related to the Vicsek model. This model was first proposed in [16] to describe the discrete-time evolution of interacting particles that move with unit velocity in the plane. In the absence of noise, the update law of the Vicsek model is

$$\begin{aligned} r_k[t+1] &= r_k[t] + e^{i\theta_k[t]} \\ \theta_k[t+1] &= \operatorname{atan}\left(\frac{\sum_{m\in\{k,\{l:l\sim k\}\}}\sin(\theta_m[t])}{\sum_{m\in\{k,\{l:l\sim k\}\}}\cos(\theta_m[t])}\right) \quad (10) \end{aligned}$$

where θ_k denotes the heading angles and r_k the positions in the plane. The latter influence the dynamics of the headings through the interconnection topology that varies as a function of the proximity of the agents; this part is beyond the scope of the present work. It is easy to verify that (10) actually corresponds to (9) with $\varepsilon_k = \frac{d_k}{1+d_k}$.

A. AN ASYNCHRONOUS DESCENT ALGORITHM

We still consider a fixed, undirected communication topology. The descent property of (2) in Section II cannot be extended to (9) on the torus in its full generality. However, randomly choosing one single agent to update at each time instant decouples the dynamics of the individual agents, allowing to minimize U_L by consecutive (and hence independent) displacement of each agent. This leads to an asynchronous descent algorithm for U_L on the torus (Algorithm A), which trivially extends to the situation where only disconnected agents do update their state at the same time (we refer to this operation mode as "locally asynchronous"). Synchronous operation is investigated in the next subsection.

According to (8) and (7), if only agent k is updated at time t, the variation of the potential reduces to

$$\begin{aligned} \Delta U_L &= -\frac{2d_k}{N}\sin\left(\tfrac{u_k}{2}\right) \\ &\qquad \Im m\left(e^{-iu_k/2}\frac{\sum_{l\sim k}e^{i(\theta_l-\theta_k)}}{d_k}\right) \\ &= -\frac{2(d_k+b_k)}{N}\sin\left(\tfrac{u_k}{2}\right) \\ &\qquad \Im m\left(e^{-iu_k/2}\frac{\sum_{l\sim k}e^{i(\theta_l-\theta_k)}+b_k}{d_k+b_k}\right) \\ &\quad -\frac{b_k}{N}\left(1-\cos(u_k)\right) , \qquad b_k > 0 . \quad (11) \end{aligned}$$

In fact, writing $\varepsilon_k = \frac{d_k}{d_k+b_k}$ and introducing (9), we have

$$\frac{\sum_{l\sim k} e^{i(\theta_l-\theta_k)} + b_k}{d_k+b_k} = (\varepsilon_k p_k + (1-\varepsilon_k) z_k) e^{-i\theta_k} = \rho_k e^{iu_k} .$$

For $\rho_k \neq 0$, (11) becomes

$$\Delta U_L = \frac{(d_k+b_k)\,\rho_k + b_k}{N} \left(\cos(u_k) - 1 \right) \leq 0 . \quad (12)$$

In case $\rho_k = 0$, ΔU_L is negative for any $u_k \neq 0$ and consequently, we get a descent algorithm for U_L.

In order to allow global convergence with asynchronous operation, we must ensure that every agent is eventually updated over a uniform time horizon. This leads to the following assumption, which is generically satisfied for a randomly chosen update sequence.

Assumption 1: *The sequence of indices $\{I[t]\}_{t_0 \leq t < \infty}$, $I[t] \in \{1, 2, ...N\}$ chosen for the asynchronous update of the agents has the property that there exist a finite time span T and a partition of the discrete-time space $[t_0, t_1)$, $[t_1, t_2)$,... with $(t_{n+1} - t_n) < T$ $\forall n \in \mathbb{N}$, such that $k \in \{I[t] : t_n \leq t < t_{n+1}\}$ for every agent $k \in \{1, 2, ...N\}$ and for every interval $[t_n, t_{n+1})$.*

In fact, Assumption 1 ensures that every agent k is updated at an infinite number of time instants $t_{1k}, t_{2k}, ... t_{nk}, ...$. As a consequence, since $U_L \geq 0$ and $\Delta U_L \leq 0$, it is necessary that all sequences $\{\Delta U_L[t_{nk}]\}$ converge to 0 when $n \rightarrow +\infty$. Considering (12), this implies that $u_k \rightarrow 0$ and agent k asymptotically reaches a point θ^*. Note that if $\rho_k|_{\theta^*} = 0$ for any agent k, then θ^* is not a fixed point (since any move of agent k would be allowed at θ^*). However, convergence towards such a point is highly unstable since it corresponds to a maximum of U_L with respect to θ_k; in fact, this instability is reflected by the discontinuity of the control law (9) when $\rho_k = 0$. In the vicinity of any other equilibrium point, the control law (9) is continuous so that the stable equilibria of the descent algorithm are the local minima of U. As $\frac{\partial^2 U_L}{\partial \theta_k^2} = \frac{1}{N} \sum_{l\sim k} \cos(\theta_l - \theta_k) \geq 0$ at these points, one easily cross-checks indeed that they correspond to situations with $\rho_k > 0$. This leads to the following proposition.

Proposition 2 (Algorithm A): *Consider a fixed, undirected graph G and the artificial potential U_L built on its Laplacian according to (8). Then, any (locally) asynchronous descent algorithm resulting from the application of (9) and satisfying* Assumption 1 *drives the system towards a local minimum of U_L for almost all initial situations. The position vector of agent k is then aligned with the centroid p_k of its neighbors as defined by the graph G.*

Remarks:

1) The minima of U_L correspond to "grouped states" in the sense of (ii) in Section II.B. In particular, for all-to-all communication (G being the complete graph), the synchronized state $\theta_k = \theta_1$ $\forall k$ is the only local minimum of U_L along the N-torus.
2) Using (11) with $b_k < 0$, one can develop a similar asynchronous ascent algorithm for $U_L(\theta)$, leading to the local maxima which correspond to "distributed states". For all-to-all communication, they correspond to "balanced states" where the centroid of the agents is located at the center of the circle.
3) It is not difficult to show that these algorithms remain valid when ε_k varies with time.

B. CONTINUOUS-TIME LIMIT and SYNCHRONOUS OPERATION

When $\varepsilon_k \rightarrow 0$, the movements of the agents are infinitesimal. In this limit case, moving agent k by $-\frac{N\varepsilon_k}{d_k}\frac{\partial U_L}{\partial z_k}$ and then projecting the new position onto the circle (Algorithm A) is strictly equivalent to taking the component $\frac{\partial U_L}{\partial \theta_k}$ of the gradient that is tangent to the circle and moving by $-\frac{N\varepsilon_k}{d_k}\frac{\partial U_L}{\partial \theta_k}$ along the circle.

As a consequence, for small ε_k, (9) may be seen as an Euler-discretization of the continuous-time gradient system

$$\dot{\theta}_k = K \frac{\partial U_L}{\partial \theta_k} = -\frac{K}{N} \sum_{l\sim k} \sin(\theta_l - \theta_k) , \quad K < 0 \quad (13)$$

described in [17] and [15] which inspired the present work. In equation (13), for all-to-all communication we recover the celebrated Kuramoto model for the evolution of agents on the circle. This shows how the sine function of the Kuramoto model is actually linked to a centroid strategy.

In [15], the continuous-time gradient algorithm (13) is shown to converge towards the local minima of U_L; this means that Algorithm A may be used in synchronous operation when $\varepsilon_k \rightarrow 0$. On the other hand, the global convergence properties of Algorithm A are lost in synchronous operation when $\varepsilon_k \rightarrow 1$. Indeed, in that case the distinction between the proposed descent algorithm and the corresponding ascent algorithm (see Remark 2) is lost in shape space and, depending on initial state, the system may converge to local maxima or local minima of U_L; further illustration, not presented here, shows that at least for some topologies, the synchronous algorithm is even prone to run into a limit cycle. A striking question at the present point is thus to know to what extent the asynchronous setting is required to ensure convergence of Algorithm A. As could be imagined, the answer involves a bound on ε_k; the following (very conservative) bound is derived in the appendix:

$$\frac{\varepsilon_k}{1 - 2\varepsilon_k} \leq \frac{M^* d_k}{2 d_{max}} \quad \forall k$$

$$\text{where} \quad \frac{e^{M^*} - 1}{M^*} = 1 + \frac{d_{max}}{N d_{mean}} .$$

However, a major shortcoming is that distributed knowledge of the global information d_{max}, d_{mean} and N (through M^*) is needed for every agent to compute this bound.

Connection with the Hopfield model: The property of asynchronous convergence that fails to extend to synchronous operation was already observed on the discrete-time network proposed by Hopfield in [18]. The Hopfield network considers N neurons with states $x_k \in \{-1, 1\}$. The discrete-time update law for the states of the neurons is

$$x[t+1] = \mathrm{sign}\left(Wx[t] + \xi\right) \qquad (14)$$

where ξ_k is a firing threshold for neuron k and W is a symmetric weight matrix with $w_{kk} = 0$. Considering the potential

$$U_L = -\frac{1}{2}\langle x|Wx\rangle - \langle x|\xi\rangle \ ,$$

Hopfield showed that when (14) is applied asynchronously with a random update sequence, the property $U_L[t+1] \leq U_L[t]$ always holds and the network eventually reaches a fixed point that corresponds to a local minimum of U_L. This is not true anymore for synchronous operation. In fact, it is shown in [19] that the system can go into a limit cycle in that case.

A comparison with Algorithm A is straightforward, though the systems themselves are completely different. Indeed, both laws are descent algorithms for a symmetric quadratic potential, with states restricted to a subset of an Euclidean state space: the N-torus in the present paper, the N-hypercube for Hopfield networks. In both situations, convergence is achieved by asynchronously moving agent k towards the point in its state space which is closest to some point p_k that lies in the associated Euclidean space ($\mathbb{C}$ or $\mathbb{R}$); this point is defined as the (possibly weighted) centroid of the neighbors of agent k in the Euclidean space. Both algorithms can alternatively be viewed as moving along the gradient of U_L in the associated Euclidean space and projecting the reached point on the state space manifold of agent k. The assumption $w_{kk} = 0$ in the Hopfield network may be linked to $\varepsilon_k \to 1$ in Algorithm A; in this case, both algorithms fail to converge in synchronous operation.

IV. A GLOBALLY CONVERGENT ALGORITHM ON THE N-TORUS

Algorithm A in the previous section has two shortcomings:

(i) the convergence analysis requires a fixed communication graph;

(ii) depending on the communication graph, the synchronized state is usually not the unique minimum of the potential.

Simulations indicate that the convergence properties of the algorithm are generically[1] retained with switching communication graphs, and in fact improved because the synchronized state is the only persistent minimum of the family of potentials $U_L(\theta)$. Nevertheless, the convergence analysis remains elusive in this more general framework.

In this section, we present an alternative synchronization algorithm (*Algorithm B*), for which global convergence properties can be established. Based on an idea recently proposed in [14], this algorithm involves a different exchange of information between communicating agents: in addition to communicating their relative position, agents are required to communicate their current estimate of a consensus orientation vector $p_k[t] \in \mathbb{C}$.

The estimate p_k of agent k is updated according to the consensus algorithm (2), that is,

$$p_k[t+1] = \varepsilon_k \operatorname*{av}_{l\sim k}(p_l[t]) + (1-\varepsilon_k)p_k[t] \, . \qquad (15)$$

The (normalized) estimate is then used in the update law (9) as a substitute to the local centroid $\mathrm{av}_{l\sim k}(z_l)$, that is,

$$\begin{aligned}\theta_k[t+1] \quad &= \quad \arg\left(\zeta_k \frac{p_k[t]}{\|p_k[t]\|} + (1-\zeta_k)z_k[t]\right) \ , \\ &\qquad \zeta_k \in (0,1], \ k = 1, 2, ..., N \ . \quad (16)\end{aligned}$$

Convergence of the consensus algorithm (15) is guaranteed under the assumptions of Proposition 1. This means that the state dynamics (16) are asymptotically governed by the update law

$$\theta_k[t+1] = \arg\left(\zeta_k \frac{p_\infty[t]}{\|p_\infty[t]\|} + (1-\zeta_k)z_k[t]\right) \ . \qquad (17)$$

The only stable fixed point of the (decentralized) dynamics (17) is $\theta_k = \arg(p_\infty)$; for $\zeta_k > 1/2$, it is the only fixed point. This means that the synchronous state $\theta_k = \arg(p_\infty)$ is the only limit set of the dynamic algorithm (15),(16). This result is summarized in the following proposition.

Proposition 3 (Algorithm B): *Considering a sequence of graphs satisfying one of the assumptions of* Proposition 1*, any solution of (15),(16) with $\zeta_k > 1/2$ and such that $p_\infty \neq 0$ globally synchronizes the agents at $\theta_k = \arg(p_\infty)$ $\forall k$ when $t \to +\infty$.*

Remarks:

1) A particular weighting of each agent's contribution in the centroid computation, leading to

$$p_k[t+1] = \sum_{l\sim k} \frac{p_l[t]}{d_l^{(o)}[t]+1} \ , \qquad k = 1, 2, ..., N \ , \qquad (18)$$

allows to control p_∞ by ensuring that $\sum_{k=1}^{N} p_k = c_0$ is invariant; the value of p_∞ is then fixed by the initial conditions which will generically be such that $c_0 \neq 0$.

2) As written down in (15) and (16), Algorithm B is not rotationally invariant; this is in contradiction with our goal to design distributed algorithms that need no external reference. The following expressions are equivalent to (15),(16), but they only involve the relative variables $\theta_l - \theta_k$, $p_k e^{-i\theta_k} =: \left(pe^{-i\theta}\right)_k$ and $\theta_k[t+1] - \theta_k[t]$; the variables $\theta_l - \theta_k$ and $\left(pe^{-i\theta}\right)_l$ are

[1]This means with unit probability for a randomly chosen graph sequence. In fact, one can easily construct counterexamples for some particular graph sequences.

the values actually exchanged among communicating agents.

$$\theta_k[t+1] - \theta_k[t] = \arg\left(\zeta_k \frac{\left(pe^{-i\theta}\right)_k[t]}{\|p_k[t]\|} + (1-\zeta_k)\right)$$

$$\begin{aligned}\left(pe^{-i\theta}\right)_k[t+1] = &\left(e^{-i(\theta_k[t+1]-\theta_k[t])}\right)\\ &\Bigg((1-\varepsilon_k)\left(pe^{-i\theta}\right)_k[t]\\ &+\varepsilon_k \operatorname*{av}_{l\sim k}\left(\left(pe^{-i\theta}\right)_l[t]\, e^{-i(\theta_k[t]-\theta_l[t])}\right)\Bigg).\end{aligned}$$

V. CONCLUSIONS AND FUTURE RESEARCH

In the present paper, we presented two strategies based on a local centroid computation to achieve global synchronization of a network of agents evolving on the circle in discrete-time.

The first strategy drives the system towards the local minima of an artificial potential based on the fixed, undirected communication topology. The control law requires knowledge of the relative positions of an agent's neighbors; it always converges when implemented asynchronously, but synchronous operation requires an appropriate step size.

The second strategy requires communication of a consensus variable which evolves in the embedding Euclidean space rather than simply the position of the agents on the circle. However, with this additional information and under some mild assumptions, the system is driven towards the synchronized state when the communication topology is varying and possibly directed.

In future work, we plan to consider similar synchronization algorithms on more general manifolds.

REFERENCES

[1] E. Justh and P. Krishnaprasad. A simple control law for UAV formation flying. Technical report, TR 2002-38, Institute for Systems Research, University of Maryland, 2002.

[2] N.E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Frantantoni, and R.Davis. Collective motion, sensor networks and ocean sampling. *To appear, Proceedings of the IEEE*, 2006.

[3] M.G. Hinchey et al. Swarm-based space exploration. *ERCIM News*, (64):26–27, January 2006.

[4] D. Izzo and L. Pettazzi. Equilibrium shaping: distributed motion planning for satellite swarm. *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in space (iSAIRAS Munich)*, 2005.

[5] C.R. McInnes. Distributed control for on-orbit assembly. *Advances in the Astronautical Sciences*, 90:2079–2092, 1996.

[6] C.W. Reynolds. Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics (ACM SIGGRAPH Conference Proceedings)*, 21:25–34, July 1987.

[7] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[8] S.H. Strogatz. *Sync: The emerging science of spontaneous order.* Hyperion, 2003.

[9] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[10] L. Moreau. Stability of multi-agent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[11] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbour rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[12] N. Moshtagh, A. Jadbabaie, and K. Daniilidis. Distributed geodesic control laws for flocking of nonholonomic agents. *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, pages 2835–2840, 2005.

[13] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus and flocking. *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, 2005.

[14] L. Scardovi and R. Sepulchre. Collective optimization over average quantities. *Submitted to the 45th IEEE Conference on Decision and Control*, 2006.

[15] R. Sepulchre, D. Paley, and N.E. Leonard. Stabilization of planar collective motion. *To appear, IEEE Transactions on Automatic Control*, 2006.

[16] T. Vicsek et al. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229, 1995.

[17] A. Jadbabaie, N. Motee, and M. Barahona. On the stability of the Kuramoto model of coupled nonlinear oscillators. *Proceedings of the American Control Conference*, 2004.

[18] J.J. Hopfield. Neural networks and physical systems with emergent collective computational capabilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 1982.

[19] E. Goles-Chacc, F. Fogelman-Soulie, and D. Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discrete Applied Mathematics*, 12(3):261–277, 1985.

APPENDIX

In this appendix, we derive a sufficient bound on the value of ε_k in order to ensure convergence for synchronous operation of Algorithm A. We proceed in two steps.

Step 1: Consider a first-order Euler-discretization of (13) in the form

$$\theta_k[t+1] = \theta_k[t] + KT\frac{\partial U_L}{\partial \theta_k}, \quad K<0\,. \tag{19}$$

The variation of the potential between two time steps is

$$\begin{aligned}\Delta U_L[t] \;=\; & KT\left\|\nabla_\theta U_L\right\|^2\\ &+\tfrac{1}{2!}\nabla_\theta^2 U_L \prod_{m=1}^{2} \times_m (KT\nabla_\theta U_L)\\ &+\tfrac{1}{3!}\nabla_\theta^3 U_L \prod_{m=1}^{3} \times_m (KT\nabla_\theta U_L)\\ &+...\end{aligned} \tag{20}$$

where $\times_m$ denotes tensor multiplication along dimension m. According to (20), U_L will be non-increasing as long as the first term remains dominant. To satisfy this condition, a bound has to be imposed on the value of KT. For this purpose, we consider the potential in the form (8) and its derivatives with respect to θ. By giving the maximal value 1 to all appearing sines and cosines, one observes that the sum of the absolute values of all elements in the tensor $\nabla_\theta^n U_L$ is smaller than $d_{mean}2^{n-1}$:

$$\operatorname{abs}\left(\nabla_\theta^n U_L\right) \prod_{m=1}^{n} \times_m \left([11...1]^T\right) \le d_{mean}2^{n-1}$$

where $d_{mean} = \frac{1}{N}\sum_{k=1}^{N} d_k$ denotes the average degree in the communication graph, $\mathrm{abs}(A)$ denotes the tensor containing the absolute values of the elements of tensor A and $n \geq 2$.

For any vector x and any tensor A_n of degree $n \geq 2$, one has

$$\begin{aligned}
&\left| A_n \prod_{m=1}^{n} \times_m x \right| \\
&\quad \leq \quad \mathrm{abs}(A_n) \prod_{m=1}^{2} \times_m [11...1]^T x^T x \prod_{m=3}^{n} \times_m \mathrm{abs}(x) \\
&\quad \leq \quad \mathrm{abs}(A_n) \prod_{m=1}^{n} \times_m [11...1]^T x^T x \left(\max_k |x_k| \right)^{n-2} .
\end{aligned}$$

In particular, replacing x by $(KT\nabla_\theta U_L)$ and A_n by $\nabla_\theta^n U_L$, we see that the higher order terms in (20) are bounded by

$$\begin{aligned}
&\frac{\left| \Delta U_L - KT \left\| \nabla_\theta U_L \right\|^2 \right|}{\left| KT \left\| \nabla_\theta U_L \right\|^2 \right|} \\
&\quad \leq \quad \frac{\sum_{n=2}^{+\infty} \frac{1}{n!} \left| A_n \prod_{m=1}^{n} \times_m x \right|}{\left| KT \left\| \nabla_\theta U_L \right\|^2 \right|} \\
&\quad \leq \quad |K| T d_{mean} \sum_{n=2}^{+\infty} \frac{1}{n!} 2^{n-1} \left(\frac{|K| T d_{max}}{N} \right)^{n-2} \\
&\quad \leq \quad \frac{N}{M} \frac{d_{mean}}{d_{max}} \sum_{n=2}^{+\infty} \frac{M^n}{n!} ,
\end{aligned}$$

$$M = \frac{2|K| T d_{max}}{N} .$$

This implies that the first term of (20) will be dominant if

$$\frac{e^M - 1}{M} \leq 1 + \frac{d_{max}}{N d_{mean}} \tag{21}$$

which may be solved numerically to produce a sufficient[2] convergence condition $M \leq M^*$. This in turn fixes a sufficient bound on KT for a given topology, which through (19) leads to the following equivalent requirement on the move of agent k between two time steps:

$$|\Delta\theta_k| \leq \frac{M^* N}{2 d_{max}} \left| \frac{\partial U_L}{\partial \theta_k} \right| . \tag{22}$$

Step 2: The second step is to connect (19) to Algorithm A. After some elementary geometrical observations, one verifies that for $\varepsilon_k < 1/2$, the distance $\Delta\theta_k$ travelled along the circle when applying control law (9) satisfies

$$|\Delta\theta_k| \leq \frac{N}{d_k} \left| \frac{\partial U_L}{\partial \theta_k} \right| \frac{\varepsilon_k}{1 - 2\varepsilon_k} .$$

Comparing with (22), we finally obtain the condition on ε_k:

$$\frac{\varepsilon_k}{1 - 2\varepsilon_k} \leq \frac{M^* d_k}{2 d_{max}} .$$

[2]Note that this bound is very conservative; solutions of (21) reduce to $M = 0$ when $\frac{d_{max}}{N d_{mean}}$ approaches 0.