

Consensus on Nonlinear Spaces [★]

R. Sepulchre

*Dept. of Electrical and Computer Engineering, Institut Montefiore,
B28, Université de Liège, 4000 Liège Sart-Tilman, Belgium
(e-mail: r.sepulchre@ulg.ac.be).*

Abstract: Consensus problems have attracted significant attention in the control community over the last decade. They act as a rich source of new mathematical problems pertaining to the growing field of cooperative and distributed control. This paper is an introduction to consensus problems whose underlying state-space is not a linear space, but instead a highly symmetric nonlinear space such as the circle and other relevant generalizations. A geometric approach is shown to highlight the connection between several fundamental models of consensus, synchronization, and coordination, to raise significant global convergence issues not present in linear models, and to be relevant for a number of engineering applications, including the design of planar or spatial coordinated motions.

1. INTRODUCTION

The present paper is an introduction to recent work by the author and several collaborators on a geometric consensus theory aimed at unifying consensus algorithms defined on linear and nonlinear spaces. The work started in 2003 with the practical question of designing control laws in order to coordinate the motion of rigid bodies moving at constant speed in the plane. The question arised in the context of designing autonomous oceanographic sensor networks (see Leonard et al. [2007]). The latest published work returns to this question by deriving the same control laws from a general coordination theory on Lie Groups (Sarlette et al. [2010b]). The basic investigation that sustained the underlying research over the seven years interval is the extension of the consensus problem, in which agents dynamically seek a value of common agreement, from values on the *real line* to values on the *circle* (Sepulchre et al. [2004]). A main reference on this topic is Sarlette and Sepulchre [2009b], strongly supported by the results of the thesis of Sarlette [2009]. This question is thought to clarify important connections and differences between the three related questions of *consensus*, *(phase) synchronization*, and *coordination*.

Because the results of the paper have been published elsewhere, the present exposition is conceptual rather than technical, referring the interested reader to the appropriate references for the technical content and a precise statement of the results.

The starting point is a brief summary of consensus theory in linear spaces in Section 2. The linear consensus model is viewed as a basic model of distributed computation. The emphasis is on its convergence properties and on its symmetries. The goal is to extend this model of distributed computation to nonlinear spaces, retaining its symmetry

properties and, to a lesser extent, its convergence properties.

Section 3 presents the geometric framework that underlies the proposed generalization. While a Riemannian structure is sufficient for an abstract generalization of the concept of *updating the current local consensus estimate in the direction of a local average*, an additional extrinsic geometry is assumed to turn the abstract update into a practical algorithm for particular configuration spaces encountered in applications: the circle, embedded in the complex plane, the orthogonal group, embedded in a linear matrix space, and the Grassmann manifold, embedded in the space of linear projectors. As a result of this additional extrinsic geometric structure, the update of the proposed generalized consensus algorithms reduces to a linear update in the ambient Euclidean space followed by a natural projection in the tangent space (for continuous-time algorithms) or on the manifold (for discrete-time algorithms). The underlying distance is a *chordal distance* (i.e. a distance computed in the ambient space in contrast to a *geodesic* distance computed on the manifold), previously used in a number of computational problems. This simple geometric framework is sufficient to connect linear consensus algorithms to a number of important nonlinear models of distributed computation, including the phase model of Kuramoto [1975], the particle model of Vicsek et al. [1995], the grassmann packing of Conway et al. [1996], and even the neural network model of Hopfield [1982].

Section 4 summarizes the current understanding of convergence properties of the proposed generalized consensus algorithms. The fundamental difference between linear consensus algorithms and the novel algorithms considered in this paper is the non-convex nature of configuration spaces like the circle. As a result, a global convergence analysis is quite untractable and at least very dependent on the communication graph of the consensus algorithm. On the positive side, we consider three modifications of consensus algorithms defined on the circle that enable a global convergence analysis. Those solutions are of definite

[★] This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author.

interest in design applications and have been applied to the design of (almost) globally convergent coordination control laws (Sepulchre et al. [2008]).

Section 5 shortly addresses the problem of *coordination* on nonlinear spaces and its relationship to consensus. While synchronization is about consensus in the configuration space, coordination is about consensus in the velocity space: a motion is coordinated when all velocity vectors are the same. In linear spaces, the velocity space can be identified to the configuration space, which makes the problems of synchronization and coordination mathematically equivalent. In contrast, when the configuration space is nonlinear, velocity vectors at different points belong to different tangent spaces, which raises the issue of comparing them in a consensus algorithm. If the configuration space is a Lie group, the natural approach is to employ the left and right invariant translation mappings to compare velocities in the Lie algebra. The concept of left (and right) relative positions can be defined accordingly, leading to a systematic approach for the design of left (and right) invariant coordination. When applied to the configuration space of rigid motions in the plane or in space, this theory recovers Lyapunov functions that were derived in an ad-hoc manner in our initial work (Sepulchre et al. [2007]).

The paper ends with a recent “afterthought” about the nonlinear nature of the configuration space of linear consensus algorithms (Sepulchre et al. [2010]). Consensus algorithms were originally considered in the framework of Markov chains. The configuration space of probability vectors is the positive orthant and consensus algorithms iterate stochastic maps on the positive orthant. A general result of Birkhoff [1957] relates the Lyapunov function introduced by Tsitsiklis for consensus to a projective distance intrinsically adapted to the nonlinear nature of the positive orthant. This connection underlies the value of a geometric approach to recognize the nonlinear nature of a seemingly linear problem and to adapt accordingly analysis tools such as Lyapunov functions. This claim is supported by the failure of quadratic Lyapunov functions to provide a tractable convergence analysis of (not so?) linear consensus algorithms (Olshevsky and Tsitsiklis [2008]).

The paper concentrates on contributions by the author and its collaborators but those contributions of course build upon an impressive body of work pertaining to the geometric control of mechanical systems. Key references relevant for the contributions presented here include the thesis of Bullo [1998], which proposes a geometric approach to the tracking control problem of mechanical systems, the thesis of Nair [2006], which proposes a geometric approach to the coordination of (dynamical models) of mechanical systems, and the work of Justh and Krishnaprasad [2004, 2005], which revealed the importance of the Lie group setting for a geometric theory of coordination. A more exhaustive account of relevant references for geometric consensus theory is to be found in the different papers providing the material of the present survey.

2. LINEAR CONSENSUS

The starting point of this paper is the by now classical linear consensus algorithm describing the behavior of N agents exchanging *locally* information about their state

$x_k \in \mathbb{R}^n$, $k \in \mathcal{V} = \{1, 2, \dots, N\}$, in order to asymptotically reach a *global* consensus, i.e. a common value of agreement. In continuous-time, the update is

$$\frac{d}{dt}x_k(t) = \sum_{j=1}^N a_{jk}(t)(x_j(t) - x_k(t)), \quad k = 1, 2, \dots, N \quad (1)$$

where a_{jk} is the weight of link $j \rightsquigarrow k$: the state of agent k evolves towards to the (positively weighted) arithmetic mean of its neighbors, $\frac{1}{d_k^{(i)}} \sum_{j \rightsquigarrow k} a_{jk}x_j$, where in-degree $d_k^{(i)} = \sum_{j \rightsquigarrow k} a_{jk}$. The corresponding update in discrete-time is

$$x_k(t+1) = \frac{1}{\beta_k(t) + d_k^{(i)}(t)} \left(\sum_{j \rightsquigarrow k} a_{jk}(t)x_j(t) + \beta_k(t)x_k(t) \right), \quad (2)$$

with non-vanishing weight $\beta_k(t) \geq \beta_0 > 0$. The weights a_{jk} induce a communication graph between the agents, see Appendix. They can be asymmetric (leading to a *directed* communication graph) and/or depend on time (leading to a *time-varying* communication graph).

In matrix form, the continuous-time algorithm is the linear time-varying system

$$\frac{d}{dt}x(t) = -L(t)x(t), \quad (3)$$

where $L(t) = D^{(i)}(t) - A(t)$ (see Appendix), while the discrete-time algorithms is the linear time-varying system

$$x(t+1) = M(t)x(t), \quad (4)$$

where $M(t) = (D^{(i)}(t) + B(t))^{-1}(A(t) + B(t))$ and $B(t)$ is a diagonal matrix with elements $B_{kk} = \beta_k$. The matrices $M(t)$ are row-stochastic, i.e. the elements of each row sum to one.

Consensus algorithms have been studied by several authors, including Tsitsiklis [1984], Jadbabaie et al. [2003], Moreau [2004, 2005], Blondel et al. [2005], Olfati-Saber and Murray [2004], Olshevsky and Tsitsiklis [2006]; see Olfati-Saber et al. [2007] for a review and Moreau [2005] for some examples of applications. The main result is that the update guarantees asymptotic consensus under minimal connectivity assumptions of the underlying communication graph.

Proposition 1. (adapted from Moreau [2004, 2005], Olfati-Saber and Murray [2004], Tsitsiklis [1984]) Consider a set of N agents evolving on \mathbb{R}^n according to (continuous-time) (1) or according to (discrete-time) (2). Then the agents globally and exponentially converge to a consensus value $\bar{x} \in \mathbb{R}^n$ if and only if the communication among agents is characterized by a (piecewise continuous) δ -digraph which is uniformly connected (see Appendix for a definition).

If in addition, \mathbb{G} is balanced for all times, then the consensus value is the arithmetic mean of the initial values: $\bar{x} = \frac{1}{N} \sum_{k=1}^N x_k(0)$.

Consensus algorithms provide a basic model of *distributed* computation: a given agent at a given time only performs *local average* computations but the spread of information over time eventually enables the computation of a centralized quantity (the arithmetic average of initial states under a balancing assumption). This diffusion mechanism is central to distributed dynamical systems and its proper

understanding in instrumental to a distributed system theory.

An essential feature of consensus algorithms is their symmetry, i.e. their invariance properties to certain transformations: the dynamics are invariant under reordering of the agents (discrete permutation symmetry) and under uniform translation of the state (continuous symmetry): for any $a \in \mathbb{R}^n$, a shifted initial condition $y_k(0) = x_k(0) + a$ $\forall k \in \mathcal{V}$ yields the shifted solution $y_k(t) = x_k(t) + a$ $\forall k \in \mathcal{V}$ and $\forall t \geq 0$. This is because the exchange of information only involves *relative* quantities ($x_j - x_k$). As a consequence, the distributed computation of a centralized quantity does not require that all agents share a (centralized) common reference frame.

In the present paper, the symmetries resulting from a fully decentralized setting are considered as constitutive of the consensus algorithm. This is not always the case in the literature. However, one should be aware of the invariance properties of a given distributed system because they are a source of fundamental performance limitations. Whether the exchange of information in a distributed setting only involves *relative* quantities or can also include *absolute* quantities (access to a centralized reference frame or distinction between the different agents) is of course problem dependent. Our recent study of distributed control of large segmented mirrors is an example of application where such a symmetry is constitutive of the design problem: the (centralized) objective of controlling the global shape of the mirror must be achieved from (distributed) sensors that can only measure the *relative* displacement of neighboring mirror cells (Sarlette et al. [2010a]). Even in contexts where partial or intermittent access to a centralized controller is possible, the design of invariant consensus algorithms is of interest to decouple the centralized reference tracking from the coordination, analogously to the classical decoupling between a tracking controller and a stabilizing controller (see Sepulchre et al. [2007] for an extended discussion).

The convergence analysis of consensus algorithms is the convergence analysis of a time-varying linear system. Remarkably, it can be completely characterized. For consensus algorithms defined in \mathbb{R} , the early analysis of Tsitsiklis [1984] rests on the basic but fundamental observation that the (time-invariant) Lyapunov function

$$V(x) = \max_{1 \leq i \leq N} x_i - \min_{1 \leq i \leq N} x_i \quad (5)$$

is non-increasing along the solutions. The proof that it decreases *strictly* over a uniform horizon under appropriate assumptions only involves elementary calculations (see e.g. Blondel et al. [2005] for details). Jadbabaie et al. [2003] exploited the graph interpretation of consensus algorithms. In discrete-time, the flow of the algorithm involves products of row-stochastic matrices and its convergence properties is related to ergodicity theorems in the theory of Markov chains. The subsequent analysis of Moreau [2004, 2005] emphasizes that the convergence result relies in an essential way on the *convexity* rather than the *linearity* of the update law: the position of each agent k for $t > \tau$ always lies in the convex hull of the $x_j(\tau)$, $j = 1, 2, \dots, N$. The permanent contraction of this convex hull, at some nonzero minimal rate because weights are non-vanishing, allows to conclude that the agents end up at a consensus

value. This approach extends the Lyapunov function (5) to vector-valued algorithms and allows for the convergence analysis of nonlinear consensus algorithms provided that the convexity property is retained.

To complete this brief summary, it is worth observing that when interconnections are not only balanced, but also undirected and fixed, then the linear consensus algorithm is a gradient descent algorithm for the disagreement cost function

$$V_{\text{vect}}(x) = \frac{1}{2} \sum_{k=1}^N \sum_{j=1}^N a_{jk} \|x_j - x_k\|^2 = x^T (L \otimes I_n) x \quad (6)$$

where $\|z\|$ denotes the Euclidean norm $\sqrt{z^T z}$ of $z \in \mathbb{R}^m$, and L is the Laplacian matrix of \mathbb{G} (see Appendix), $x \in \mathbb{R}^{Nn}$ denotes the vector whose elements $(k-1)n+1$ to kn contain x_k , and $\otimes I_n$ is the Kronecker product by the $n \times n$ identity matrix.

3. CONSENSUS IN NONLINEAR SPACES

3.1 Generalized consensus algorithms

A geometric interpretation of the linear consensus algorithms (1) and (2) is to view the state $x_k(t)$ as the estimate at time t by agent k of the consensus value. At each time step, each agent updates its current estimate of the consensus value towards a (weighted) average of its neighbors estimates. Moreover, the weighted arithmetic average can be given the geometric interpretation of the point that minimizes the sum of the (weighted) squared distances :

$$\sum_{j=1}^N a_{ij} x_j = \min_{z \in \mathbb{R}^n} \sum_{j=1}^N a_{ij} \|z - x_j\|^2 \quad (7)$$

With this geometric interpretation, consensus algorithms can be defined on arbitrary Riemannian manifolds. The Riemannian (or Karcher) mean on a manifold \mathcal{M} is defined by substituting the Riemannian (geodesic) distance to the Euclidean distance in (7):

$$\text{mean}(x_1, \dots, x_N) = \min_{z \in \mathcal{M}} \sum_{j=1}^N a_{ij} d_{\mathcal{M}}(z, x_j)^2 \quad (8)$$

Furthermore, on a Riemannian manifold, “updating a point towards a new point” simply translates as “moving along the geodesic path connecting the two points”. This approach yields an intrinsic definition of consensus algorithms on Riemannian manifolds but suffers both a fundamental and a practical limitations. The fundamental limitation is that the uniqueness of a geodesic is only ensured locally. If several geodesics connect two points, both the concepts of mean and the concepts of “moving along shortest paths connecting two points” become non unique. A practical limitation is that the computation of geodesics at each time step in a distributed algorithm might represent a formidable computational task. Those limitations can be overcome to a large extent if the manifold is embedded in an Euclidean space and if mean and distance calculations are carried out in the Euclidean geometry of the ambient space.

An additional desired property of our generalized consensus algorithms is to retain the symmetry properties. In essence, distributed algorithms should be defined on spaces where all points “look alike”. This is the case for Lie groups, and, more generally, homogeneous spaces.

Those considerations led us to consider in Sarlette and Sepulchre [2009c] state spaces that satisfy the following assumption :

Assumption 1. \mathcal{M} is a connected compact homogeneous manifold smoothly embedded in \mathcal{E} with the Euclidean norm $\|x\| = r_{\mathcal{M}}$ constant over $x \in \mathcal{M}$.

The embedding space \mathcal{E} denotes the linear vector space \mathbb{R}^n or the linear matrix space $\mathbb{R}^{n \times n}$. It is a well-known fact of differential geometry that any smooth m -dimensional Riemannian manifold can be smoothly and isometrically embedded in \mathbb{R}^{2m} . The additional condition $\|y\| = r_{\mathcal{M}}$ is in agreement with the fact that all points on \mathcal{M} should be equivalent.

On a manifold that satisfies Assumption 1, a convenient alternative to the intrinsic generalization of consensus algorithms is to base the calculations on the distance of the ambient Euclidean space. Replacing the distance on \mathcal{M} by the Euclidean distance in \mathcal{E} in the definition (8) leads to the **induced arithmetic mean** $IAM \subseteq \mathcal{M}$ of N agents of weights $a_{ij} > 0$ and positions $x_j \in \mathcal{M}$, $j = 1 \dots N$:

$$IAM(x_1, \dots, x_N) = \operatorname{argmin}_{z \in \mathcal{M}} \sum_{j=1}^N a_{ij} \|\hat{x}_j - \hat{z}\|^2 \quad (9)$$

The notation \hat{z} denotes the (vector) embedding of z in the linear space \mathcal{E} .

As a global minimizer, the induced arithmetic mean is not necessarily a singleton. But it is continuously differentiable and it includes the reassuring properties that the IAM of a single point x_1 is the point itself, that it is invariant under permutations of agents of equal weights, and that it commutes with the symmetry group of the homogeneous manifold. Although less intrinsic, its main advantage over the Karcher mean is computational. The IAM is closely related to the weighted centroid in the ambient space, defined as

$$C_e(\hat{x}_1, \dots, \hat{x}_N) = \frac{1}{a_i} \sum_{j=1}^N a_{ij} \hat{x}_j, \quad a_i = \sum_{j=1}^N a_{ij}.$$

Since $\|c\| = r_{\mathcal{M}}$ by Assumption 1, an equivalent definition for the IAM is

$$IAM = \operatorname{argmax}_{z \in \mathcal{M}} (\hat{z}^T C_e) \quad (10)$$

Hence, computing the IAM just involves a search for the global maximizers of a linear function on \mathcal{E} in a very regular search space \mathcal{M} . Local maximization methods even suffice if the linear function has no maxima on \mathcal{M} , which is formulated as a blanket assumption.

Assumption 2. The local maxima of a linear function $f(\hat{z}) = \hat{z}^T b$ over $z \in \mathcal{M}$, with b fixed in \mathcal{E} , are all global maxima.

Assumptions 1 and 2 are satisfied for a number of nonlinear spaces encountered in applications. Meaningful examples

discussed in Sarlette and Sepulchre [2009c] include the circle S^1 , the unit sphere of \mathcal{E} , the orthogonal group $SO(n)$, and the Grassmann manifold $Gr(p, n)$.

The circle is a fundamental and representative example. The circle embedded in the (complex) plane with its center at the origin satisfies Assumptions 1 and 2. The IAM is simply the central projection of C_e onto the circle. It corresponds to the entire circle if $C_e = 0$ and reduces to a single point in other situations. The IAM uses the chordal length between points, while the Karcher mean would use arclength distance.

The generalization of linear consensus algorithms to state spaces that satisfy Assumptions 1 and 2 is straightforward: the update is simply taken as a linear update towards the centroid in the ambient space, and then projected to the closest point of the manifold M . In continuous-time, this amounts to the differential equation

$$\dot{x}_k(t) = \operatorname{Proj}_{T_{\mathcal{M}_{x_k}}} \left(\sum_j a_{jk} (\hat{x}_j - \hat{x}_k) \right), \quad k = 1 \dots N. \quad (11)$$

Similarly, the discrete-time update is

$$x_k(t+1) = \operatorname{Proj}_M \left(\frac{1}{\beta_k + d_k^{(i)}} \left(\sum_{j \rightsquigarrow k} a_{jk} \hat{x}_j + \beta_k \hat{x}_k \right) \right) \quad (12)$$

which can also be written as

$$x_k(t+1) \in IAM(\{x_j(t) | j \rightsquigarrow k \text{ in } G(t)\} \cup \{x_k(t)\}) \quad (13)$$

These updates admit an analytic expression for several manifolds of interest, see Sarlette and Sepulchre [2009c]. For the circle, the embedding of a point $\theta_k \in S^1$ in \mathbb{C} is the vector $e^{i\theta_k}$. The continuous-time update is

$$\dot{\theta}_k = \operatorname{Im} \left(\sum_j a_{jk} (e^{i(\theta_j - \theta_k)} - 1) \right) = \sum_j a_{jk} \sin(\theta_j - \theta_k) \quad (14)$$

while the discrete-time update is

$$\theta_k(t+1) = \arg \left(\frac{1}{\beta_k + d_k^{(i)}} \left(\sum_{j \rightsquigarrow k} a_{jk} e^{i\theta_j} + \beta_k e^{i\theta_k} \right) \right) \quad (15)$$

Those expressions establish a clear connection between consensus algorithms on the circle and well-known phase synchronization models. For the equally-weighted complete graph, the continuous-time update (14) is the celebrated model of Kuramoto [1975] with identical (zero) natural frequencies. For the equally-weighted complete graph, the norm of the centroid is

$$\|C_e\|^2 = \left\| \frac{1}{N} \sum_{j=1}^N e^{i\theta_j} \right\|^2 \quad (16)$$

and (14) is a gradient system for (16). It is interesting to observe that this cost function was used as a measure of synchrony in the original paper of Kuramoto under the name *complex order parameter*. The discrete-time update (15) is Vicsek’s phase update law (see Vicsek et al. [1995]) governing the headings of a set of particles in the plane. The communication graph $G(t)$ is in this case a (state-dependent) proximity graph, with weight $a_{jk} = 1$ if particle j and k lie within a given radius of visibility and $a_{jk} = 0$ otherwise. The precise relationship between the continuous-time and discrete-time consensus algorithms on the circle is further discussed in see Sarlette et al. [2006]. Consensus on the circle is studied in Sepulchre et al. [2007],

Sarlette et al. [2006], Scardovi et al. [2007a], Sepulchre et al. [2008]. In Sarlette and Sepulchre [2009c], corresponding algorithms are also derived on the orthogonal group and on the Grassmann manifold.

3.2 Consensus and anti-consensus configurations

Consider a set of N agents of positions x_k on a manifold \mathcal{M} satisfying Assumption 1.

For a given (fixed) graph G , a stable fixed point of consensus algorithms is when the state of each agent is the IAM of its neighbors neighbors $j \rightsquigarrow k$, weighted according to the strength of the corresponding edge. Such a configuration is called a *consensus configuration*. **Synchronization** is the particular consensus configuration where $x_j = x_k \forall j, k$. In linear consensus algorithms, synchronization is the only consensus configuration of a strongly connected graph. But this is no longer true in nonlinear spaces, where consensus configurations become graph-dependent. Only very particular graphs - such as equally-weighted complete graphs or trees - admit synchronization as the sole consensus configuration. Of course, synchronization is a consensus state for all graphs. As a consequence, switching among different graphs increases the probability of eventually reaching a synchronized configuration.

On compact spaces such as those considered in this paper, it also makes sense to consider *anti-consensus* algorithms, that do the opposite of consensus : each state is updated to move *away* from the mean of its neighbors. Stable fixed points of those algorithms are called anti-consensus configurations. The opposite of synchronization is called *balancing*. Balanced configurations correspond to anti-consensus configurations in which the IAM of the agents includes the entire manifold. In balanced configurations, the agent states spread over the entire manifold. Computational algorithms that aim at maximally spreading a set of points over a compact manifold are not uncommon. They include the Grassmann packing problem, that has attracted significant attention in the recent years.

On the circle, synchronization maximizes the centroid norm while balancing states minimize it. Anti-consensus configurations for the equally-weighted complete graph are fully characterized in Sepulchre et al. [2007]. It is shown that the only anti-consensus configurations that are not balanced correspond to $(N+1)/2$ agents at one position and $(N-1)/2$ agents at the opposite position on the circle, for N odd. Balanced configurations are unique for $N=2$ and $N=3$ and form a continuum for $N>3$.

(Anti)-consensus algorithms on compact manifolds embedded on a unit sphere can be thought as distributed ways to optimize the norm of the global centroid. Each agent updates its state by optimizing a local estimate of the centroid. Eventually, one expects the entire swarm to evolve either to synchronization (maximization) or to balancing (minimization). Of course, spurious local minima and maxima will exist for a given graph. But switching among graphs that share a restricted set of (anti)-consensus states will favor convergence towards them. In particular, synchronization is a consensus state shared by all graphs, which explains why synchronization is the

most frequently asymptotic state observed in simulations of consensus algorithms.

4. CONVERGENCE ANALYSIS

The consensus algorithms (11) and (12) are meaningful extensions of synchronization algorithms for compact homogeneous spaces like the circle. However, those configuration spaces are not convex. As a consequence, their convergence properties are fundamentally different from those of (1) and (2) on vector spaces.

The convergence analysis of consensus algorithms on nonlinear spaces is extensively studied in the thesis of Sarlette [2009]. The expository paper by Sarlette et al. [2008] focuses on the circle but contains most important convergence results. We briefly summarize them in the present section.

4.1 General (but not global) convergence results

Moreau's convergence analysis of consensus algorithms is based on the contraction of the convex hull of agent's states. This analysis is not restricted to linear updates and it applies to the nonlinear consensus algorithms of this paper over arbitrary convex sets of the manifold M . In the case of the circle, for instance, the largest convex subset is a semicircle. As a consequence, uniform convergence to consensus under a uniform connectedness assumption holds provided that the states are initially contained in a semicircle. Because none of the considered manifolds is globally convex, the *linear* convergence result never provides a global convergence analysis.

Another general observation is that the continuous-time algorithm (11) defines a gradient system for the disagreement cost function

$$V(x) = \frac{1}{2} \sum_{k=1}^N \sum_{j=1}^N a_{jk} \|\hat{x}_j - \hat{x}_k\|^2 = \hat{x}^T (L \otimes I_n) \hat{x} \quad (17)$$

whenever the communication graph is fixed and undirected. As a consequence, the general theory of gradient systems applies in this situation: all solutions converge to the set of critical points and all strict minima of $V(x)$ are stable equilibria. Synchronization corresponds to the global minimum of the disagreement cost function. The descent property extends to discrete-time algorithms for sufficiently small step sizes (i.e. for β_k close to one) or for an asynchronous version of (12) in which agents are updated asynchronously. Those results are reminiscent of convergence results for the Hopfield neural network model Hopfield [1982] in which N neurons with states $x_k \in \{-1, 1\}$ are updated according to

$$x_k(t+1) = \text{sign} \left(\sum_{j=1}^N a_{jk} x_j(t) + \xi_k \right), \quad 1, 2, \dots, N \quad (18)$$

where ξ_k is a firing threshold. Defining the sphere S^n of dimension n as $\{x_k \in \mathbb{R}^{n+1} : \|x_k\| = 1\}$, the set $\{-1, 1\}$ can be seen as " S^0 ", while the circle is S^1 . It is a - perhaps not so irrelevant - curiosity to observe that, for $\xi_k = 0$, (18) is in fact the strict analog of (12) for "the sphere of dimension 0" - namely moving towards the neighbors in the embedding vector space and projecting

back to the state space. The absence of inertia in (18) corresponds to $\beta = 0$ in (12). Both (15) and (18) can be viewed as projections of descent algorithms for a symmetric quadratic potential, which remain descent algorithms under locally asynchronous update such that convergence is ensured. Both algorithms can fail to converge and run into a limit cycle in synchronous operation.

4.2 Global convergence analysis is graph dependent

The global properties of consensus algorithms defined on nonlinear spaces are graph dependent. Some graphs – including the complete (directed or undirected) graph and any root-connected tree – have the property that almost all solutions converge to a synchronized solution. But it seems quite hard to characterize the graphs that have that property (see Section 4.3. in Sarlette et al. [2008]). Graphs that have some (discrete) symmetry are prone to (locally) stable consensus configurations different from sync. In fact, any sufficiently spread agent configuration on the circle can be made a stable consensus configuration of a well-chosen directed graph.

It is shown in (Sarlette and Sepulchre [2009b]) that stable limit sets of consensus algorithms on the circle do not restrict to equilibria. Fixed directed graphs can be constructed that lead to stable periodic and quasi-periodic behaviors. The situation is even more complex for time-varying graphs.

4.3 Three solutions to recover (almost) global convergence

The negative fact that a global analysis of the proposed consensus algorithms seems elusive on nonlinear spaces, even on the circle, is compensated for by the fact that the algorithms can be modified in such a way that convergence is guaranteed under a mere uniform connectedness assumption, such as in linear spaces. Three solutions have been investigated (on the circle) to recover global convergence.

Potential shaping. A first solution only concerns the case of a fixed undirected graph. The continuous-time algorithm (11) is then a gradient system and, for almost all initial conditions, convergence is guaranteed to a local minimum of the disagreement cost function. In other words, the sinusoidal coupling in (14) derives from a potential. It is shown in (Sarlette and Sepulchre [2009b]) how to shape the potential in such a way that sync is the only stable equilibrium. The only graph information needed to construct this potential is an upper bound on the number of nodes. A descent algorithm for this shaped potential guarantees almost global convergence to sync if the (fixed undirected) graph is connected.

Gossip Algorithm. A second solution, first proposed in Sarlette et al. [2008], is to introduce randomness in the link selection of the consensus algorithm, following the idea of a *gossip* consensus algorithm, see Boyd et al. [2006]. At each time instant, a given agent selects randomly one (or none) of its neighbors. The update is then taken as if this neighbor was the only one, disregarding the information from others. Such a gossip algorithm can be shown to

uniformly converge to sync with probability one under a uniform connectedness of the graph, even if the graph is directed and/or time-varying. An extreme version of this gossip algorithm is when the update is chosen with no inertia ($\beta = 0$): each agent selects one neighbor randomly and replaces its current value by the neighbor's value with a certain probability. In this case, the consensus value is the initial condition of one of the agents. The convergence property of this gossip algorithm is very general and does not require any geometric structure on the underlying configuration space. The convergence property of the algorithm only relies on the probabilistic time evolution of N states switching among at most N different symbols. The probability that the number of possible symbols decreases over a finite time horizon (defined by the uniform connectedness assumption) is positive, leading eventually to consensus. Favoring the probability of convergence to synchronization through randomness comes with a price: convergence can be arbitrarily slow. The expected synchronization time can be shown to be independent of the initial conditions but it is not easy to characterize which graph property favors faster convergence. Designing the “right” amount of randomness for a given graph would be a sensible design question.

Dynamic consensus. A third solution, first proposed in (Scardovi et al. [2007a]), is to increase the amount of information exchanged by the agents. The non-convexity of S^1 can be circumvented if the agents are able to communicate *auxiliary variables* in addition to their positions on the circle. An interpretation is that the limited number of communication links for information flow is compensated by sending larger communication packets along existing links. Such strategies allow to recover the synchronization properties of vector spaces for almost all initial conditions. Their potential interest lies more in engineering applications than in physical modeling, where communication of auxiliary variables is questionable. The use of auxiliary variables was initially motivated by the design of coordinated motions in the plane: the assumption of complete communication in Sepulchre et al. [2007] was relaxed in this way to an assumption of uniform connectedness Sepulchre et al. [2008].

The proposed dynamic consensus algorithm exploits the embedding of the algorithm in a linear space. Algorithms (11) and (12) compute an update in the embedding space and project the result onto the manifold. If the embedding variables can be exchanged, the consensus algorithm can instead be run entirely on the embedding variables, while each agent tracks its own embedding variable. Consensus among the embedding variables relies on convergence of linear consensus algorithms, leading to asymptotic consensus on the manifold as well. When the manifold is a Lie group such as the circle, the communication of auxiliary variables can be implemented in such a way that it respects the symmetry constraints of the algorithm, i.e. without the need of a common reference frame. See Scardovi et al. [2007a] and Sarlette [2009] for details and a formal proof.

5. COORDINATION ON NONLINEAR SPACES

The design and analysis of control laws that coordinate swarms of identical autonomous agents has attracted sig-

nificant attention in the recent years: a non-exhaustive list includes flocking mechanisms (Jadbabaie et al. [2003], Blondel et al. [2005]), vehicle formations (Fax and Murray [2004], Desai et al. [2001], Olfati-Saber and Murray [2002], Justh and Krishnaprasad [2004, 2005]), spacecraft formations (Mesbahi and Hadaegh [2001], VanDyke and Hall [2006], Lawton and Beard [2002], Krogstad and Gravdahl [2006], McInnes [1996], Ren [2006]), mechanical system networks (Smith et al. [2001], Hansmann et al. [2006], Nair [2006]), and mobile sensor networks (Scardovi et al. [2007b], Sepulchre et al. [2007, 2008], Leonard et al. [2007], Swain et al. [2007]).

Coordination is closely related to consensus and synchronization: while synchronization refers to consensus among positions (or configuration variables), coordination refers to consensus among velocities. For instance, on the circle, phase synchronization refers to consensus on S^1 , while coordination would refer to a situation where all phase variables evolve at the same speed, a situation commonly referred to as phase locking. Both synchronization and coordination can be considered as consensus problems, but in different spaces: the *configuration space* for the former, and the *tangent bundle* for the latter. When the configuration space is linear, there is a natural identification between the configuration space and the tangent space. Velocity vectors are treated as vectors of the configuration space by translating them to the origin. Coordination and synchronization are then equivalent in the sense that they both reduce to a linear consensus problem. The situation is different when the configuration space is nonlinear. Velocity vectors of different agents then belong to different tangent spaces that can no longer be identified to the configuration space. This raises the issue of comparing velocities and to ensure the analog *invariance* properties of a coordination control law.

Lie groups offer a convenient setting for coordination on nonlinear spaces because the group operation provides an invertible translation mapping that generalizes the translation operation in linear spaces. Because the group operation is in general non commutative, one distinguishes between two translation operations: the product $g_k^{-1}g_j$ defines a left-invariant relative position between g_k and g_j while the product $g_jg_k^{-1}$ provides a right-invariant relative position. Both generalize the relative position $r_k - r_j$ in a linear space. The recent work by Sarlette et al. [2010b] builds upon this generalization to investigate a coordination theory on Lie groups. Left-invariant (resp. right-invariant) coordination is defined as designing a control law that stabilizes left-invariant (resp. right-invariant) relative positions. Bi-invariant coordination refers to simultaneous left and right invariant coordination.

Intuitively, constant relative positions correspond to equal velocities. On Lie groups, velocities can be compared in the Lie algebra, i.e. the tangent space at identity, which is a linear space. And indeed, constant left-invariant relative positions are shown to be equivalent to identical right-invariant velocities. Likewise, constant right-invariant relative positions are shown to be equivalent to identical left-invariant velocities. This characterization allows to classify all possible coordinated motions on a particular Lie group. Lie groups of special interest in mechanics include the

group of rotations $SO(n)$, the group of rigid motions in the plane $SE(2)$ and the group of rigid motions in space $SE(3)$.

The characterization of coordination in terms of equal velocities suggests a systematic Lyapunov design of coordinating control laws. The Lyapunov functions measure the distance to coordination by comparing the velocity vectors in the Lie algebra. The feedback control laws are then selected to decrease those distances along the closed-loop trajectories.

The illustration of the theory on the Lie group $SE(2)$ is instructive. The left-invariant velocity of an agent with position r_k and orientation θ_k is the velocity in body frame (i.e. the linear velocity v_k^k and the angular velocity ω_k). Right-invariant coordination thus corresponds to equal velocities in body frame. The trajectory of a given agent in the plane is arbitrary, but any two agents trajectories are related by a rigid rotation and translation in the plane.

On $SE(2)$, the right-invariant linear velocity of an agent is the center of curvature

$$v_r^k = e^{i\theta_k} v_t^k - i\omega_k r_k$$

where the complex notation is a compact notation for planar rotations. Note that v_r^k differs from the linear velocity in reference frame when $\omega_k \neq 0$. Left-invariant coordination thus corresponds to equal centers of curvatures. Left-invariant coordinated motions correspond to parallel motion of the agents (if the angular velocity is zero) or to concentric circular motions around a fixed center of curvature (if the angular velocity is different from zero).

Right-invariant coordination on $SE(2)$ is achieved by running a consensus algorithms among the linear and angular velocities. Left-invariant coordination is achieved by synchronizing the centers of curvature. The naive Lyapunov function for left-invariant coordination is the disagreement cost function

$$V(x) = \frac{1}{2} \sum_{k=1}^N \sum_{j=1}^N a_{jk} \|v_r^k - v_r^j\|^2 \quad (19)$$

between right-invariant velocities expressed in the Lie algebra. This formula connects a general coordination theory on Lie groups to the ad-hoc Lyapunov design proposed in Sepulchre et al. [2007] for coordination in a model of steered particles moving at unit speed in the plane. In a model of steered particles, the linear velocity v_t^k is fixed to the constant vector e_1 . The steering control is a consensus angular velocity ω corrected by a feedback term aimed at achieving left-invariant coordination. If $\omega = 0$, the expression (19) reduces to $\frac{1}{2} \sum_{k=1}^N \sum_{j=1}^N a_{jk} \|e^{i\theta_k} - e^{i\theta_j}\|^2$ and the associated Lyapunov control stabilizes parallel motion. The task happens to be equivalent to synchronization of the agents orientations, which is a consensus problem on the circle. If $\omega \neq 0$, the Lyapunov function becomes $\frac{1}{2} \sum_{k=1}^N \sum_{j=1}^N a_{jk} \|e^{i\theta_k} - e^{i\theta_j} - i\omega(r_k - r_j)\|^2$ and the associated Lyapunov control stabilizes circular motion. Those two types of coordinated motion are the only two possible coordinated motions in a model of steered particles.

The geometric theory of coordination on Lie groups facilitates extensions to other nonlinear configuration spaces

such as $SE(3)$, see Scardovi et al. [2008], Sarlette et al. [2010b] for details.

6. A NONLINEAR GLIMPSE AT LINEAR CONSENSUS

A remarkable feature of the convergence analysis of the time-varying linear system associated to consensus algorithms (3) and (4) is the non-quadratic nature of the Lyapunov function (5). It is an essential feature of the convergence analysis: the recent paper Olshevsky and Tsitsiklis [2008] provides the explicit construction of a sequence of eight matrices $M(t)$ that satisfies the assumptions of a linear consensus algorithm but that does not admit a common time-invariant quadratic Lyapunov function. We recently proposed in (Sepulchre et al. [2010]) a geometric interpretation of this non-quadratic Lyapunov function, as a distance measure intrinsically related to the nonlinear nature of the underlying state-space. Since the consensus algorithm (1) is invariant by translation, it can be assumed without loss of generality that it is defined in the positive orthant $\mathcal{K} = \{(x_1, \dots, x_N) : x_i \geq 0, 1 \leq i \leq N\}$. In other words, each row-stochastic matrix $M(t)$ defines a positive linear map in \mathcal{K} , i.e. an application that maps the interior of the cone on itself. (The assumption $\beta_k > 0$ in (2) guarantees that points in the interior of the cone are mapped in the interior of the cone). As a linear map, it is also monotone, i.e. it preserves the natural order of the cone \mathcal{K} .

A general result by Birkhoff [1957] shows that positive linear maps on cones are nonexpanding. On the positive orthant, the contraction measure is the Hilbert metric

$$d_H(x, y) = \max_{1 \leq i \leq N} \log(x_i/y_i) - \min_{1 \leq i \leq N} \log(x_i/y_i)$$

and a consequence of Birkoff theorem is that

$$d_H(M(t)x, M(t)y) \leq d_H(x, y)$$

for any x and y in the (interior of) the positive orthant. Row-stochasticity of $M(t)$ means that $M(t)\mathbf{1} = \mathbf{1}$. The distance to $\mathbf{1}$ is

$$V_B(x) = d_H(x, \mathbf{1}) = \max_{1 \leq i \leq N} \log(x_i) - \min_{1 \leq i \leq N} \log(x_i)$$

which is precisely the Lyapunov function (5) in log coordinates. Both Lyapunov functions provide a distance to consensus. Tsitsiklis Lyapunov function is translation-invariant ($V(x + \lambda\mathbf{1}) = V(x)$) while Birkhoff Lyapunov function is scaling-invariant ($V_B(\lambda x) = V(x)$ for $\lambda > 0$). They both provide a measure of the diameter of the convex hull of (x_1, \dots, x_N) , which is the Lyapunov function proposed by Moreau [2005].

This interpretation of the non-quadratic Lyapunov function (5) underlines the perhaps not so linear nature of the underlying state space of linear consensus algorithms. Because Birkhoff theorem holds in arbitrary cones, this geometric interpretation allows to consider consensus algorithms in arbitrary cones. The paper by Sepulchre et al. [2010] uses this framework to study consensus algorithms on the cone of positive definite matrices, which are closely related to stochastic quantum maps encountered in non-commutative probability spaces.

7. CONCLUSION

The paper is a survey of the work by the author and collaborators on a geometric consensus theory. While elementary, the proposed theory enables precise connections between several popular models of distributed computation, by viewing them as realizations of the same geometric update on different nonlinear configuration spaces. Phase synchronization is viewed as a consensus algorithm on the circle, while coordination on Lie groups is viewed as consensus of invariant velocities in the Lie algebra.

An essential difference between linear consensus algorithms and their nonlinear extensions is the non-convex nature of symmetric spaces like the circle. This property is what makes the convergence analysis graph dependent on nonlinear spaces. From a design viewpoint, it is of interest to reformulate consensus algorithms on nonlinear spaces in such a way that they converge (almost) globally under the same assumptions as linear consensus algorithms, and several solutions were proposed on the circle.

The proposed theory was initially motivated by coordination problems whose configuration spaces are those encountered in mechanics, such as $SE(2)$ or $SE(3)$. But a particular emphasis of the geometric approach is also on the invariance properties of distributed computation models. Much work remains to be done in this direction, but recognizing the invariance properties of distributed systems and characterizing the associated fundamental design limitations may end up being the most important added value of a geometric consensus theory in applications, see Sarlette et al. [2010a], Sarlette and Sepulchre [2009a] for preliminary steps in that direction.

ACKNOWLEDGEMENTS

The author wishes to express sincere gratitude to long-time collaborators who made this contribution possible. The work was initiated thanks to Naomi Leonard during a fruitful sabbatical stay at Princeton University. Many results presented here were obtained with Alain Sarlette during the course of his PhD. Other precious collaborators include Pierre-Antoine Absil, Vincent Blondel, Silvère Bonnabel, Julien Hendrickx, Derek Paley, Pierre Rouchon, Luca Scardovi, and Emre Tuna. They cannot be all listed as co-authors of the present paper but their contributions are well identified in the different papers that provided the material of the present survey.

REFERENCES

- G Birkhoff. Extensions of Jentzsch's theorem. *Trans. Amer. Math. Soc.*, 85:219–227, 1957.
- V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus and flocking. *Proc. 44th IEEE Conf. Decision and Control*, 2005.
- S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Information Theory (Special issue)*, 52(6):2508–2530, 2006.
- F. Bullo. Nonlinear control of mechanical systems: a Riemannian geometry approach. *PhD Thesis, CalTech*, 1998.

- F.R.K. Chung. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. AMS, 1997.
- J.H. Conway, R.H. Hardin, and N.J.A. Sloane. Packing lines, planes, etc.: packings in Grassmannian spaces. *Exper. Math.*, 5(2):139–159, 1996.
- J.P. Desai, J.P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Trans. Robotics and Automation*, 17(6):905–908, 2001.
- R. Diestel. *Graph Theory*. Springer, 1997.
- J. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Trans. Automatic Control*, 49(9):1465–1476, 2004.
- H. Hansmann, N.E. Leonard, and T.R. Smith. Symmetry and reduction for coordinated rigid bodies. *Eur. J. Control*, 12(2):176–194, 2006.
- J.J. Hopfield. Neural networks and physical systems with emergent collective computational capabilities. *Proc. Nat. Acad. Sci.*, 79:2554–2558, 1982.
- A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Automatic Control*, 48(6):988–1001, 2003.
- E.W. Justh and P.S. Krishnaprasad. Equilibria and steering laws for planar formations. *Systems and Control Letters*, 52:25–38, 2004.
- E.W. Justh and P.S. Krishnaprasad. Natural frames and interacting particles in three dimensions. *Proc. 44th IEEE Conf. Decision and Control*, pages 2841–2846, 2005.
- T.R. Krogstad and J.T. Gravdahl. Coordinated attitude control of satellites in formation. In *Group Coordination and Cooperative Control*, volume 336 of *Lect. N. Control and Inform. Sci.*, pages 153–170. Springer, 2006.
- Y. Kuramoto. Self-entrainment of population of coupled nonlinear oscillators. In *Internat. Symp. on Math. Problems in Theoretical Physics*, volume 39 of *Lect. N. Physics*, page 420. Springer, 1975.
- J.R. Lawton and R.W. Beard. Synchronized multiple spacecraft rotations. *Automatica*, 38:1359–1364, 2002.
- N.E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Frantantoni, and R. Davis. Collective motion, sensor networks and ocean sampling. *Proc. IEEE*, 95(1):48–74, 2007.
- C.R. McInnes. Distributed control for on-orbit assembly. *Adv. Astronautical Sci.*, 90:2079–2092, 1996.
- M. Mesbahi and F.Y. Hadaegh. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching. *J. Guidance, Control and Dynamics*, 24(2):369–377, 2001.
- L. Moreau. Stability of continuous-time distributed consensus algorithms. *Proc. 43rd IEEE Conf. Decision and Control*, pages 3998–4003, 2004.
- L. Moreau. Stability of multi-agent systems with time-dependent communication links. *IEEE Trans. Automatic Control*, 50(2):169–182, 2005.
- S. Nair. Stabilization and synchronization of networked mechanical systems. *PhD Thesis, Princeton University*, 2006.
- R. Olfati-Saber and R.M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. *Proc. 41st IEEE Conf. Decision and Control*, 2002.
- R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time delays. *IEEE Trans. Automatic Control*, 49(9):1520–1533, 2004.
- R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE*, 95(1):215–233, 2007.
- A. Olshevsky and J. Tsitsiklis. On the nonexistence of quadratic Lyapunov functions for consensus algorithms. *IEEE Trans. Autom. Control*, 53(11):2642–2645, 2008.
- A. Olshevsky and J.N. Tsitsiklis. Convergence rates in distributed consensus and averaging. *Proc. 45th IEEE Conf. Decision and Control*, pages 3387–3392, 2006.
- W. Ren. Distributed attitude consensus among multiple networked spacecraft. *Proc. American Control Conference*, 2006.
- A. Sarlette. Geometry and symmetries in coordination control. *PhD Thesis, University of Liège*, 2009.
- A. Sarlette and R. Sepulchre. A PDE viewpoint on basic properties of coordination algorithms with symmetries. *Proc. 48th IEEE Conference on Control and Decision, Shanghai (China)*, pages 1696–1701, 2009a.
- A. Sarlette and R. Sepulchre. Synchronization on the circle. *arxiv:0901.2408*, 2009b.
- A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. *SIAM J. Control and Optimization*, 48(1):56–76, 2009c.
- A. Sarlette, R. Sepulchre, and N.E. Leonard. Discrete-time synchronization on the N -torus. *Proc. 17th Intern. Symp. Math. Theory of Networks and Systems*, pages 2408–2414, 2006.
- A. Sarlette, S.E. Tuna, V.D. Blondel, and R. Sepulchre. Global synchronization on the circle. *Proc. 17th IFAC World Congress*, 2008.
- A. Sarlette, Ch. Bastin, M. Dimmler, B. Sedgi, T. Erm, B. Bauvir, and R. Sepulchre. Integral control from distributed sensing: an integral control from distributed sensins: an extremely large telescope case study. *submitted to IEEE Trans. on Control Systems Technology*, 2010a.
- A. Sarlette, S. Bonnabel, and R. Sepulchre. Coordinated motion design on Lie groups. *IEEE Trans. Automatic Control*, 55(5):1047–1058, May 2010b.
- L. Scardovi, A. Sarlette, and R. Sepulchre. Synchronization and balancing on the N -torus. *Systems and Control Letters*, 56(5):335–341, 2007a.
- L. Scardovi, R. Sepulchre, and N.E. Leonard. Stabilization of collective motion in three dimensions: a consensus approach. *Proc. 46th IEEE Conf. Decision and Control*, pages 2931–2936, 2007b.
- L. Scardovi, N.E. Leonard, and R. Sepulchre. Stabilization of three-dimensional collective motion. *submitted to the Brockett Legacy Special Issue of Communications in Information and Systems*, 2008.
- R. Sepulchre, D. Paley, and N.E. Leonard. Collective motion and oscillator synchronization. In A. Morse V. Kumar, N. Leonard, editor, *Cooperative control*, volume 309 of *Lect. N. Control and Information Sci.*, pages 189–205. Springer, 2004.
- R. Sepulchre, D. Paley, and N.E. Leonard. Stabilization of planar collective motion with all-to-all communication. *IEEE Trans. Automatic Control*, 52(5):811–824, 2007.

- R. Sepulchre, D. Paley, and N.E. Leonard. Stabilization of planar collective motion with limited communication. *IEEE Trans. Automatic Control*, 53(3):706–719, 2008.
- R. Sepulchre, A. Sarlette, and P. Rouchon. Consensus on non-commutative spaces. *Submitted to IEEE Conference on Control and Decision, Atlanta.*, 2010.
- T.R. Smith, H. Hansmann, and N.E. Leonard. Orientation control of multiple underwater vehicles with symmetry-breaking potentials. *Proc. 40th IEEE Conf. Decision and Control*, pages 4598–4603, 2001.
- D.T. Swain, N.E. Leonard, I.D. Couzin, A. Kao, and R. Sepulchre. Alternating spatial patterns for coordinated group motion. *Proc. 46th IEEE Conf. Decision and Control*, pages 2925–2930, 2007.
- J.N. Tsitsiklis. Problems in decentralized decision making and computation. *PhD Thesis, MIT*, 1984.
- M.C. VanDyke and C.D. Hall. Decentralized coordinated attitude control of a formation of spacecraft. *J. Guidance, Control and Dynamics*, 29(5):1101–1109, 2006.
- T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75(6):1226–1229, 1995.

APPENDIX: NOTIONS OF GRAPH THEORY

In the framework of coordination with limited interconnections between agents, it is customary to represent communication links by means of a *graph* (see for instance Diestel [1997], Chung [1997]).

Definition 1. A directed graph $\mathbb{G}(\mathcal{V}, \mathcal{E})$ (short digraph \mathbb{G}) is composed of a finite set \mathcal{V} of vertices, and a set \mathcal{E} of edges which represent interconnections among the vertices as ordered pairs (j, k) with j and $k \in \mathcal{V}$.

A weighted digraph $\mathbb{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$ is a digraph associated with a set \mathcal{A} that assigns a positive weight $a_{jk} \in \mathbb{R}_{>0}$ to each edge $(j, k) \in \mathcal{E}$.

An unweighted graph is often considered as a weighted graph with unit weights. A weighted graph can be defined by its vertices and weights only, by extending the weight set to all pairs of vertices and imposing $a_{jk} = 0$ if and only if (j, k) does not belong to the edges of \mathbb{G} . A digraph is said to be *undirected* if $a_{jk} = a_{kj} \forall j, k \in \mathcal{V}$. It may happen that $(j, k) \in \mathcal{E}$ whenever $(k, j) \in \mathcal{E} \forall j, k \in \mathcal{V}$, but $a_{jk} \neq a_{kj}$ for some $j, k \in \mathcal{V}$; in this case the graph is called *bidirectional*. Equivalently, an unweighted undirected graph can be defined as a set of vertices and a set of *unordered* pairs of vertices.

In the present paper, each agent is identified with a vertex of a graph; the N agents = vertices are designed by positive integers $1, 2, \dots, N$, so $\mathcal{V} = \{1, 2, \dots, N\}$. The presence of edge (j, k) has the meaning that agent j sends information to agent k , or equivalently, agent k measures quantities concerning agent j . It is assumed that no “communication link” is needed for an agent k to get information about itself, so \mathbb{G} contains no self-loops: $(k, k) \notin \mathcal{E} \forall k \in \mathcal{V}$. In visual representations of a graph, a vertex is depicted by a point, and edge (j, k) by an arrow from j to k . Therefore a frequent alternative notation for $(j, k) \in \mathcal{E}$ is $j \rightsquigarrow k$.

One also says that j is an *in-neighbor* of k and k is an *out-neighbor* of j . In the visual representation of an undirected graph, all arrows are bidirectional; therefore arrowheads are usually dropped. One simply says that j and k are *neighbors* and writes $j \sim k$ instead of $j \rightsquigarrow k$ and $k \rightsquigarrow j$. The *in-degree* of vertex k is $d_k^{(i)} = \sum_{j=1}^N a_{jk}$. The *out-degree* of vertex k is $d_k^{(o)} = \sum_{j=1}^N a_{kj}$. A digraph is said to be *balanced* if $d_k^{(i)} = d_k^{(o)} \forall k \in \mathcal{V}$; in particular, undirected graphs are balanced.

The *adjacency matrix* $A \in \mathbb{R}^{N \times N}$ of a graph \mathbb{G} contains a_{jk} in row j , column k ; it is symmetric if and only if \mathbb{G} is undirected. Denote by $|\mathcal{E}|$ the number of edges in \mathbb{G} .

The in- and out-degrees of vertices $1, 2, \dots, N$ can be assembled in diagonal matrices $D^{(o)}$ and $D^{(i)}$. The *in-Laplacian* of \mathbb{G} is $L^{(i)} = D^{(i)} - A$. Similarly, the associated *out-Laplacian* is $L^{(o)} = D^{(o)} - A$. For a balanced graph \mathbb{G} , the *Laplacian* $L = L^{(i)} = L^{(o)}$. The standard definition of Laplacian L is for undirected graphs. For the latter, L is symmetric. For general digraphs, by construction, $(\mathbf{1}_N)^T L^{(i)} = 0$ and $L^{(o)} \mathbf{1}_N = 0$ where $\mathbf{1}_N$ is the column vector of N ones. The spectrum of the Laplacian reflects several interesting properties of the associated graph, specially in the case of undirected graphs, see for example Chung [1997]. In particular, it reflects its *connectivity* properties.

A *directed path* of length l from vertex j to vertex k is a sequence of vertices v_0, v_1, \dots, v_l with $v_0 = j$ and $v_l = k$ and such that $(v_m, v_{m+1}) \in \mathcal{E}$ for $m = 0, 1, \dots, l-1$. An *undirected path* between vertices j and k is a sequence of vertices v_0, v_1, \dots, v_l with $v_0 = j$ and $v_l = k$ and such that $(v_m, v_{m+1}) \in \mathcal{E}$ or $(v_{m+1}, v_m) \in \mathcal{E}$, for $m = 0, 1, \dots, l-1$. A digraph \mathbb{G} is *strongly connected* if it contains a directed path from every vertex to every other vertex (and thus also back to itself). A digraph \mathbb{G} is *root-connected* if it contains a node k , called the *root*, from which there is a path to every other vertex (but not necessarily back to itself). A digraph \mathbb{G} is *weakly connected* if it contains an undirected path between any two of its vertices. For an undirected graph \mathbb{G} , all these notions become equivalent and are simply summarized by the term *connected*. For \mathbb{G} representing interconnections in a network of agents, clearly coordination can only take place if \mathbb{G} is connected. If this is not the case, coordination will only be achievable separately in each connected component of \mathbb{G} . A more interesting discussion of connectivity arises when the graph \mathbb{G} can vary with time. In this case, the communication links are represented by a *time-varying graph* $\mathbb{G}(t)$ in which the vertex set \mathcal{V} is fixed (by convention), but edges \mathcal{E} and weights \mathcal{A} can depend on time. All the previous definitions carry over to time-varying graphs; simply, each quantity depends on time. To prevent edges from vanishing or growing indefinitely, the present paper considers *δ -digraphs*, for which the elements of $A(t)$ are bounded and satisfy the threshold $a_{jk}(t) \geq \delta > 0 \forall (j, k) \in \mathcal{E}(t)$, for all t . In addition, in continuous-time \mathbb{G} is assumed to be piecewise continuous. For δ -digraphs $\mathbb{G}(t)$, it is intuitively clear that coordination may be achieved if information exchange is “sufficiently frequent”, without requiring it to take place all the time. The following definition of “integrated con-

nectivity over time” can be found in Blondel et al. [2005], Moreau [2004, 2005], Tsitsiklis [1984].

Definition 2. (from Moreau [2004, 2005]) *In discrete-time, for a δ -digraph $\mathbb{G}(\mathcal{V}, \mathcal{E}(t), \mathcal{A}(t))$ and some constant $T \in \mathbb{Z}_{\geq 0}$, define the graph $\bar{\mathbb{G}}(\mathcal{V}, \bar{\mathcal{E}}(t), \bar{\mathcal{A}}(t))$ where $\bar{\mathcal{E}}(t)$ contains all edges that appear in $\mathbb{G}(\tau)$ for $\tau \in [t, t + T]$ and $\bar{a}_{jk}(t) = \sum_{\tau=t}^{t+T} a_{jk}(\tau)$. Similarly, in continuous-time, for a δ -digraph $\mathbb{G}(\mathcal{V}, \mathcal{E}(t), \mathcal{A}(t))$ and some constant $T \in \mathbb{R}_{>0}$, define the graph $\bar{\mathbb{G}}(\mathcal{V}, \bar{\mathcal{E}}(t), \bar{\mathcal{A}}(t))$ by*

$$\bar{a}_{jk}(t) = \begin{cases} \int_t^{t+T} a_{jk}(\tau) d\tau & \text{if } \int_t^{t+T} a_{jk}(\tau) d\tau \geq \delta \\ 0 & \text{if } \int_t^{t+T} a_{jk}(\tau) d\tau < \delta \end{cases}$$

$(j, k) \in \bar{\mathcal{E}}(t)$ if and only if $\bar{a}_{jk}(t) \neq 0$.

Then $\mathbb{G}(t)$ is said to be uniformly connected over T if there exists a time horizon T and a vertex $k \in \mathcal{V}$ such that $\bar{\mathbb{G}}(t)$ is root-connected with root k for all t .

The (equally weighted) *complete graph* is an unweighted, undirected graph that contains an edge between any pair of vertices. An *undirected tree* is a connected undirected graph in which it is impossible to select a subset of at least 3 vertices and a subset of edges among them to form an undirected cycle. A *directed tree* of root k is a root-connected digraph of root k , in which every vertex can be reached from k by following one and only one directed path. In a directed tree \mathbb{G} , the (unique) in-neighbor of a vertex j is called its *parent* and its out-neighbors are its *children*. The root has no parent, and the vertices with no children are called the *leaves*. This can be carried over to an undirected graph after selecting an arbitrary root.