# Towards coordination algorithms on compact Lie groups

Alain Sarlette
Supervisor: Rodolphe Sepulchre

DEA research report 2007
Department of Electrical Engineering and Computer Science
University of Liège

## Acknowledgments

Hearty thanks are due to Professor Rodolphe Sepulchre, whose enthusiasm, clarifying ideas and support beyond the strict limits of a supervisor, played a major role towards the completion of the present work. Further, Professor Naomi Leonard and the Mechanical and Aerospace Engineering department of Princeton University are acknowledged for an - in all respects - enriching research visit of 3 months. Further thanks go to the many people who have shared insightful discussions. Dr. Luca Scardovi and Professor Pierre-Antoine Absil were the most frequent collaborators. All these people, including colleagues at Liège University and Princeton University, also deserve many thanks for all their friendship. As always, the support of the family was crucial. Finally, it must be mentioned that this work was achieved with the financial support of the FNRS (National Fund for Scientific Research) and the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization) funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

## Abstract

The present work considers the design of control algorithms to coordinate a swarm of identical, autonomous, cooperating agents that evolve on compact Lie groups. The objective is that the agents reach a so-called *consensus state* without using any external reference. In the same line of thought, a *leader-follower* approach where 'follower' agents would track one 'leader' agent is excluded, in favor of a fully cooperative strategy. Moreover, the presence of communication links between agents is explicitly restricted, leading to undirected, directed and/or time-varying communication structures.

Two levels of complexity are considered for the models of the agents. First, they are modeled as simple integrators on Lie groups. This setting is meaningful in a trajectory-planning context for swarms of mechanical vehicles, or to solve algorithmic problems involving multiple agent coordination. In a second step, the model of Newtonian mechanics is used for *Lie group solids*, which correspond to the abstraction of the Euler laws for the rotation of a rigid body to general Lie groups. This setting is relevant for the actual control of mechanical vehicles through torques and forces.

As a common starting point, the consensus problem is formulated in terms of the extrema of a cost function. This cost function is linked to a specific centroid definition on manifolds, which is referred to in this work as the *induced arithmetic mean*, that is easily computable in closed form and hence may be of independent interest. Using the integrator model, this naturally leads to efficient gradient algorithms to synchronize (i.e. maximizing the consensus) or balance (i.e. minimizing the consensus) the agents; the latter however can only implement the corresponding control laws if the communication graph is fixed and undirected. For directed and/or varying communication graphs, a convenient adaptation of the gradient algorithms is obtained using auxiliary *estimator variables* that evolve in an embedding vector space. An extension of these results to homogeneous manifolds is briefly discussed. For the mechanical model, the coordination objective is specialized to *coordinated motion* (i.e. moving such that the relative positions of the agents are conserved) and *synchronization* (i.e. having all the agents at the same position on the Lie group). Control laws are derived using two classical approaches of nonlinear control - tracking and energy shaping. They are both based on the ideas developed in the first part.

For the sake of easier understanding and given its practical importance as representing orientations of rigid bodies in 3-dimensional space, the group $SO(3)$ (or more generally $SO(n)$) is used as a running example throughout this report. Other examples are the circle $SO(2)$ and, for the extension to homogeneous manifolds, the Grassmann manifolds $Grass(p, n)$.

As this report is written in the middle of research activities, it closes with several future research directions that can be explored in the continuity of the present work.

# Contents

**Overview of the contributions** **5**

**1 Relevance of coordination on Lie groups** **7**

1.1 Applications involving coordination on compact Lie groups . . . 7

1.2 Previous work . . . 8

1.2.1 Algorithmic consensus . . . 9

1.2.2 Coordination of mechanical systems . . . 10

**2 Theoretical background** **11**

2.1 Elements of graph theory . . . 11

2.2 Systems on compact Lie groups and homogeneous manifolds . . . 12

2.2.1 Motion on Lie groups and homogeneous manifolds . . . 12

2.2.2 Dynamics on compact Lie groups . . . 14

2.2.3 Examples . . . 14

**3 Induced arithmetic mean and consensus reaching on Lie groups** **17**

3.1 The induced arithmetic mean . . . 17

3.1.1 Definition and discussion . . . 17

3.1.2 Examples . . . 19

3.2 Consensus as an optimization task . . . 20

3.2.1 Definition . . . 20

3.2.2 Examples . . . 22

3.2.3 Consensus optimization strategy . . . 22

3.2.4 Example . . . 24

3.3 Gradient consensus algorithms . . . 24

3.3.1 Gradient algorithms for fixed undirected graphs . . . 24

3.3.2 Extension to directed and time-varying graphs . . . 25

3.3.3 Examples . . . 26

3.4 Consensus algorithms with estimator variables . . . 28

3.4.1 Synchronization algorithm . . . 28

3.4.2 Anti-consensus algorithm . . . 29

3.4.3 About the communication of estimator variables . . . 31

3.4.4 Example . . . 31

3.5 Extension to homogeneous manifolds . . . 32

3.5.1 Example . . . 33

**4 Synchronization of autonomous Lie group solids** **36**

4.1 Problem setting . . . 36

4.2 Coordination strategy . . . 37

4.2.1 From integrators to mechanical systems . . . 37

4.2.2 Example . . . 39

4.3 Consensus tracking . . . 39

4.3.1 Basic considerations . . . 39

4.3.2 The computed torque method . . . 40

4.3.3 The high gain method . . . 42

4.3.4 Example . . . 44

4.4 Energy shaping . . . 44

4.4.1 Dissipation in inertial space . . . 45

4.4.2 Dissipation in shape space . . . 46

4.4.3 Example . . . 47

**Finale** **51**
Recap - Overview of the contributions revisited . . . . . 51
Conclusions . . . . . 52
Perspectives for future research . . . . . 52

**References** **54**

# Overview of the contributions

The work presented in this report is part of a larger research project on coordinated motion control on Lie groups. The present work, as a first step, is restricted to *compact* Lie groups. The starting point is a swarm of identical agents, moving autonomously on a compact Lie group $\mathcal{G}$. The individual agents can share information along communication links as defined by an interconnection graph. The task is to design control laws for the agents as a function of the relative positions of connected agents, such that they move in order to reach a state of coordinated motion. These control laws may imply no external reference, and the various possibilities that must be considered for the interconnection graph exclude a leader-follower approach. The strategy behind the algorithms embeds $\mathcal{G}$ in a Euclidean space $\mathcal{E}$. Distances between agents are measured in $\mathcal{E}$ in order to build cost functions for an optimization-based approach. Two types of models of increasing complexity are considered for the (always fully actuated) agents: simple integrators and controlled mechanical systems (so-called *Lie group solids*).

The simple integrator model is used to focus on the issues related to systems that are *globally* distributed on a Lie group and to reduced interconnections in this setting. The main contribution is to define various qualitative situations for the relative positions of the agents on the Lie group $\mathcal{G}$. In this first part, the degree of freedom concerning possible synchronized motions of the agents is not considered: the algorithms drive the agents towards a specific configuration where they remain at rest. The two extreme such configurations are the situation where all the agents are at the same position (*synchronization*) and the situation where the agents are spread in some way on the whole manifold (*balancing*). Those two problems are formulated as optimizing (i.e., maximizing or minimizing) a simple *consensus function* which depends on the relative states of all the pairs of agents. Using a similar characterization but with a limited set of agent pairs, intermediate situations called *consensus* and *anti-consensus* are defined. Those are all built on a specific definition of the mean of positions on a manifold, which may be interesting on its own account. It is called the *induced arithmetic mean* in the present paper.

Two types of algorithms are proposed, depending on the goal and the available communication links. For fixed undirected interconnection graphs, gradient algorithms can be used to lead the agents to corresponding consensus or anti-consensus states. For time-varying and/or directed interconnection graphs, the two extreme cases of synchronization and balancing can be reached thanks to an adaptation of the previous algorithms using auxiliary *estimator variables.*

These results can be generalized to connected compact homogeneous manifolds. A journal paper version of this first part of the report can be found in [1].

The second part of the report merges the consensus approach considered in the first part with the more realistic model of a mechanical system moving on a Lie group. The main contribution here is the design of algorithms such that the swarm converges towards synchronization of the positions or to synchronization of the velocities (such that the relative positions remain constant). In both cases, the control laws explicitly incorporate the possibility to have a (non-trivial) synchronized motion of the agents. The algorithms are derived using two popular strategies in control of mechanical systems: (consensus) tracking and energy shaping.

For the consensus tracking approach, the trajectories resulting from the consensus algorithms as designed in the first part are considered as desired trajectories that each mechanical agent asymptotically tracks. Though an integrated proof is provided for particular tracking controllers, any controller developed for tracking (or, with some restrictions, even for stabilization) on Lie groups can probably be used instead.

For the energy shaping approach, the consensus function defined in the first part is used as an artificial potential for the mechanical system. This approach is not entirely new, but it seems that the proper design of dissipation in order to obtain asymptotic stability of the synchronized state, without using an external reference for the swarm, had not been explicitly solved yet at the time of the research presented in this report.

The results in this part could not all be formulated for general connected compact Lie groups: the last result is specialized to the classical (and probably most useful) Lie group $SO(3)$, for which the "Lie group solid" becomes the actual 3-dimensional rotating rigid body. A conference paper version of this second part of the report, specialized to $SO(3)$, can be found in [2].

Though the subjects and approaches in this work are popular in the literature, very few authors to date solved relevant problems on Lie groups. Among the many challenges that Lie groups raise with respect to vector spaces, the most significant are the non-convexity of the state space (leading to non-convex problems as soon as the agents are spread over a large portion of the state space - see the contributions in the first part of this report) and the particular relations between position, velocity and acceleration (leading among others to drift terms in the mechanical models - see the contributions in the second part of this report).

The report is organized as follows. The relevance of the present work is briefly discussed in Section 1, including some considerations about previous work. Section 2 provides the necessary background and notations for the mathematical formulations and developments in the main part of the report. Section 3 contains the first part of the main contributions, in which the simple integrator model is considered in order to define the induced arithmetic mean, consensus and anti-consensus configurations, and deduce algorithms to drive the swarm of simple integrators towards these configurations. Section 4, containing the second part of the main contributions, goes one step further by considering mechanical models and synchronized motions of the swarm. After summarizing the results and giving some general conclusions, the report ends with a discussion about future research.

# 1 Relevance of coordination on Lie groups

## 1.1 Applications involving coordination on compact Lie groups

The distributed computation of means/averages/centroids of datasets (in an algorithmic setting) and the *synchronization* of a set of agents (in a control setting) - i.e. driving all the agents to a common position in state space - are ubiquitous tasks in current engineering problems. Likewise, spreading a set of agents in some optimal way in the available state space - linked to *balancing* as defined in the present work - is a classical problem of growing interest. Swarms of autonomous agents are increasingly considered as an advantageous option to carry out complex tasks which would be lengthy or infeasible for a single agent.

For instance, many modern space mission concepts involve the use of multiple satellites flying in formation. Mostly, the objective is to get (virtual) structures in orbit that are substantially larger than what current launch technologies can directly handle. Potential applications arising in current studies include resolution enhancement through multiple-spacecraft SAR (the InSAR concept, or ONERA's Romulus study), interferometry (ESA's Darwin project, NASA's equivalent Terrestrial Planet Finder project or NASA's Constellation-X project) or supersized focal length (ESA's XEUS project, derived from the Symbol-X concept of CNES), sensitivity increasing through screens on secondary spacecraft (the American New World Discoverer concept for the JWST) or large-scale measurements (the ESA-NASA cooperative mission LISA), and autonomous in-orbit assembly of large real structures (projects are still at a draft level; see [3] and [4] for example). Other advantages of spacecraft formations are their robustness with respect to single spacecraft failure, and the reconfigurability of the swarm to fit specific observation requirements.

Similar lists of projects can be made for formations of unmanned aerial vehicles (UAV's), autonomous underwater vehicles (AUV's) or terrestrial platoons - in fact any type of general vehicle formations ([5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27] and references therein). A central problem in formation control is to ensure proper coordination of the agents, i.e. to bring them to and keep them in the desired formation. This requires to design coordination algorithms for mechanical systems, as is done in Section 4. At this point, it must be emphasized that accurate formation control poses several different problems of which the present work just considers one part: focusing on convergence from states that can be far away from the desired equilibrium and inherently including strong robustness considerations, the developed control laws might most probably be useful for initial deployment of the formation or for recovery after strong transient perturbations. In general, most theoretical studies are far from final science operations requirements, where accuracy is a key issue for navigation and control and unmodeled disturbances or constraints are present. An example of a practical GNC implementation for the Darwin mission can be found in [28].

The first part of this work considers coordination algorithms for simple integrators. From this algorithmic point of view, practical applications involving coordination include distributed decision making (e.g. [29, 30]), neural and communication networks (e.g. [31, 32]), clustering and other reduction methods (e.g. [33]), optimal covering or coding (e.g. [34, 35, 36, 37]) and many other fields where "dynamically computing an average" or "optimally distributing a set of points" appear as sub-tasks ([38]). In addition, in a modeling framework, the understanding of synchronization or more generally swarm behavior using simple models has led to many important studies discovering fundamental properties (e.g. [39, 40, 41]). Moreover, as is done in the present work, coordination algorithms for simple integrators can be used - more or less directly - at a task planning level for motion coordination in a swarm of mechanical systems ([13, 18, 16, 19, 42]).

Consensus algorithms are well understood in *Euclidean spaces* (see e.g. the results in [43, 44, 30, 29]). They are based on the natural definition and distributed computation of the centroid in such spaces. The literature about formation control of mechanical systems on vector spaces is even wider; a thorough survey would require a longer discussion and the interested reader is referred to citations in the references of the present paper. However, as evidenced by the above list, many interesting applications involve *manifolds* that are not homeomorphic to a Euclidean space. Even for formations moving in $\mathbb{R}^2$ or $\mathbb{R}^3$, the orientation of the agents is characterized by state variables in a non-Euclidean manifold $SO(2) \cong S^1$ or $SO(3)$. However, much less studies are devoted to consensus and coordination on non-Euclidean spaces. For example, position control of spacecraft formations has attracted many studies, while attitude control has been addressed only in a few papers (see the following review of previous work). Balancing algorithms only make sense on compact state spaces; though many theoretical results concern convex or star-shaped subspaces of a Euclidean space (see e.g. [37]), again most interesting applications involve compact manifolds. The study of global synchronization or balancing in non-Euclidean manifolds is not widely covered in the literature - except for the circle. The definition and computation of centroids on manifolds is also scarcely addressed.

Synchronization on non-Euclidean manifolds raises particular questions that must be solved on the way to particular applications ([45, 46]). The first studies of coordination on manifolds are concerned with the circle ([13, 14, 15, 47]), a basic interest being among others the study of oscillator synchronization through the celebrated Kuramoto model ([39, 48]). The next most important class of examples is probably $SO(n)$, representing the orientations of $n$-dimensional rigid bodies ([7, 10, 11, 49, 50, 51]). On the algorithmic side, data fusion also considers $SO(n)$, as well as the Grassmannian manifolds $Grass(p, n)$ of $p$-dimensional subspaces in an $n$-dimensional space. Optimal packing often considers the sphere and $Grass(p, n)$ as well. For instance, in [36], optimal placement of $N$ laser beams for cancer treatment and the representation of multi-dimensional data on a 2-dimensional computer screen by means of projections on $N$ representative planes are mentioned as practical applications of optimal distribution on $Grass(p, n)$. Clustering algorithms on $Grass(p, n)$ have also received attention recently [33]. The sphere and $Grass(p, n)$ are not Lie groups, but belong to a very close family of perfectly symmetric state spaces called *homogeneous manifolds* (the manifolds corresponding to Lie groups are also termed *principal homogeneous manifolds*) to which the results of the first part can be more or less readily generalized.

To avoid lengthy reformulations, the present paper makes the choice to work exclusively in continuous-time. For algorithmic applications, it is perfectly legitimate to argue that a discrete-time approach would be more appropriate. The adaptation of the present algorithms to discrete-time is expected to cause no fundamental problems. For the circle, equivalent continuous-time and discrete-time algorithms are explicitly established and studied in [47]. See also [52] for discrete-time algorithms on manifolds.

## 1.2 Previous work

A favorite application of coordination on compact Lie groups, which also serves as the running example in the present work, is the group $SO(3)$ characterizing the orientations of rigid bodies in 3-dimensional space. Though the present work takes an inherently geometric viewpoint of $SO(3)$, it must be stressed that the most popular representation of rigid body orientations, particularly for applications to satellite control, uses unitary quaternions ([6, 27, 42]). The manifold corresponding to unitary quaternions is not strictly equivalent to $SO(3)$. For individual satellite control, some easy tricks allow to tackle this problem. However, it is not clear how these tricks could be adaptated to synchronization of autonomous satellites without using an external reference. Therefore, the quaternion representation of rigid body orientation is ignored in the present work.

### 1.2.1 Algorithmic consensus

A first class of previous work in the area of coordination on Lie groups concerns *consensus algorithms.* In this approach, the agents are modeled as first- or second-order integrators whose inputs must be designed under imposed communication constraints. The focus lies on the latter and dynamical issues are ignored and mostly irrelevant because the algorithms operate in a computational or task-planning framework. Most existing consensus results are valid for Euclidean state spaces ([16, 20, 21, 30, 29, 43, 44, 53, 54]), but recent work also considers non-Euclidean spaces ([12, 13, 14, 15, 22, 38, 47, 1],...).

Most of the work related to consensus on manifolds has been done on the circle $S^1$. The most extensive literature on the subject derives from the Kuramoto model (see [48] for a review). Recently however, synchronization on the circle has been considered from a control perspective, where the state variables represent the directions of motion of agents in the plane. Most results concern *local* convergence properties [16, 43]. An interesting set of *globally* convergent algorithms in $SE(2) = S^1 \times \mathbb{R}^2$ is presented in [13], but they require all-to-all communication. Some problems related to global discrete-time synchronization on $S^1$ under different communication constraints are discussed in [55], where connections of the control problem with various existing models are made. Stronger results are presented in [47] for global synchronization and balancing on $S^1$ with varying and directed communication graphs, at the cost of introducing an auxiliary estimator variable that communicating agents must exchange. Finally, [14] presents results on $SE(2)$ similar to those of [13] but under relaxed communication assumptions, using among others the estimator strategy of [47],[38]. In [56], an algorithm similar to the basic algorithm of [13] is proposed for spheres $S^{n-1} \in \mathbb{R}^n$ of arbitrary dimensions; convergence results however are still limited to the case of $S^1$.

Synchronization or balancing on a manifold $\mathcal{M}$ is closely related to the definition and computation of a mean or centroid of points on $\mathcal{M}$. This basic problem has attracted somewhat more attention, as can be seen from [57, 58, 45] among others.

The key element in the strategy developed in the first part of this work is the embedding of $\mathcal{M}$ in a Euclidean space $\mathcal{E}$ and the consequent easy computation of a centroid in $\mathcal{E}$. This idea is not entirely new. It is connected to the "projected arithmetic mean" defined in [46] for $SO(3)$, which similarly uses the metric of the embedding space instead of the inherent Riemannian metric along the manifold. In fact, this simplification process of computing statistics in a larger and simpler embedding manifold (usually Euclidean space) and projecting the result back onto the original manifold, goes back as far as 1972 [59].

A remark about the computation of a "centroid of subspaces" is presented in [60] as a short example, without much theoretical analysis. In fact, one observes that the algorithms of the present work, when written on $Grass(p, n)$, are similar and can eventually be viewed as generalizing the developments in [60] in the framework of consensus and synchronization. More recently, [33] uses the embedding of $Grass(p, n)$ with the projector representation and the associated centroid definition, exactly as is done in Section 3.5 below but without going into theoretical details, to compute the centers of the clusters in a clustering algorithm. The distance measure associated with this centroid on Grassmann is called the *chordal distance* in [35]; the latter notion is in fact introduced in [36] where the projector representation of $Grass(p, n)$ and the associated distance measure in the embedding Euclidean space are used to derive optimal distributions ("packings") of $N$ agents on some specific Grassmann manifolds.

Finally, in general, the optimization-based design of algorithms on manifolds is a topic that has considerably developed over the last decades (see e.g. [61], [62] and the books [63, 52]).

### 1.2.2 Coordination of mechanical systems

A second viewpoint on coordination comes from the field of control of mechanical systems. In this framework, the non-trivial second-order dynamics on Lie groups are always explicitly incorporated. The main example used in the present work - rigid body attitudes on $SO(3)$ - is also the most popular Lie group in the literature about mechanical systems. In a coordination framework, it has attracted some attention for its application to satellite attitude synchronization.

Algorithms that asymptotically stabilize synchronized satellite attitudes are presented in [6] and [27]. Interconnections among satellites are limited, and convergence is proven for a behavioral algorithm combining tracking of a desired attitude, eigenaxis rotations and synchronization of the swarm. However, these results strongly depend on the tracking of a common external reference: when the latter is suppressed, the limited basin of attraction from which synchronization is guaranteed vanishes to the empty set. The work in [26, 27] similarly depends on an external reference. In [49] and [42], attitude synchronization is considered with a leader/follower approach. In that case, the leader spacecraft can be seen as a reference which is tracked by the followers. Control algorithms are presented that globally stabilize attitude synchronization, but the robustness of this approach critically depends on the reliability of the leader spacecraft and on the ability of all the followers to track it.

In these approaches, the use of the convenient but non-unique quaternion representation for rigid-body orientations can produce unwanted artefacts in the satellites' motions: sometimes a satellite that has an attitude very close to the leader moves in the opposite direction to come back from another side. It seems that quaternions are absolutely reliable as long as relative orientations are considered individually, but can run into problems when several orientations are combined without a common external reference. Tracking algorithms working inherently on the relevant manifold ($SO(3)$ in the present case) are developed in [50].

The authors in [7] consider the attitude synchronization problem on $SO(3)$ without external references and quaternion artefacts. In fact, their artificial coupling potentials are the same as in the present work. Using the Method of Controlled Lagrangians, local stability of a synchronized state is studied and achieved in a specific situation (final synchronized rotation around the short principal axis, specifically fixed communication interconnections). In addition to being local, this result is *not asymptotic*, meaning that the satellites remain close to the equilibrium but do not converge towards it. In the same line of work ([5, 8, 9, 10, 11, 64]), asymptotic convergence is achieved by adding an external reference. Thanks to their close link to the present study, these contributions are discussed in more detail in Section 4.2.

It must be mentioned that powerful theoretical tools have been developed to study highly symmetric mechanical systems, though their use is avoided in the present work because of their difficulty of application in practice. The starting point for these developments is a classical tool which is indeed used in the present work: energy shaping[1]. Building on these energy methods, various reduction techniques serve to deal with the symmetries of the systems themselves and those arising from their interconnection. Relevant tools are the Energy-Momentum ([65]) and Energy-Casimir ([66]) methods as well as Semidirect Product Reduction [67]. Application of these concepts to coordination of multiple agents is discussed in [68].

[1] The Method of Controlled Lagrangians is actually a particular tool for energy shaping.

# 2 Theoretical background

The present section briefly reviews two key mathematical tools that are needed to understand the core of the work. First, the representation of agent interconnections by graphs is clarified. Secondly, a technical section about Lie groups and homogeneous manifolds provides an introduction to their basic properties and reviews some identities that are used in the sequel.

## 2.1 Elements of graph theory

Coordination in a group of agents depends on the available communication links. When considering limited agent interconnections, it is customary to represent the communication links by means of a *graph*. The graph $G$ is composed of $N$ *vertices* (denoting the $N$ agents) and contains the *edge* $(j, k)$ if agent $j$ sends information to agent $k$ (i.e. vertex $j$ is an *in-neighbor* of vertex $k$), which is denoted $j \rightsquigarrow k$. A positive *weight* $a_{jk}$ is associated to each edge $(j, k)$ to obtain a weighted graph; the weight is extended to any pair of vertices by imposing $a_{jk} = 0$ iff $(j, k)$ does not belong to the graph edges of $G$. The full notation for the resulting *digraph* (directed graph) is $G(V, E, A)$ where $V$ denotes the set of vertices, $E$ denotes the set of edges and the matrix $A$, composed of the elements $a_{jk}$, is called the *adjacency matrix* of the graph. In agreement with the representation of communication links, it is assumed that $a_{kk} = 0\ \forall k$.

The *out-degree* of a vertex $k$ is defined as the quantity $d_k^{(o)} = \sum_{j=1}^{N} a_{kj}$ of information that leaves $k$ towards other agents and the *in-degree* of $k$ is the quantity $d_k^{(i)} = \sum_{j=1}^{N} a_{jk}$ of information that $k$ receives from other agents. These degrees can be assembled in diagonal matrices $D^{(o)}$ and $D^{(i)}$. A *balanced* graph is a graph for which $D^{(o)} = D^{(i)}$. This is satisfied in particular by *undirected* graphs, for which $A = A^T$. A graph is called *bidirectional* if $(j, k) \in E \Leftrightarrow (k, j) \in E$ (but not necessarily $A = A^T$).

The *Laplacian* $L$ of a graph $G$ is defined as $L = D - A$. Some variations exist on which degree to use for $D$ in the case of directed graphs; to avoid confusion, one can define the in-Laplacian $L^{(i)} = D^{(i)} - A$ and the out-Laplacian $L^{(o)} = D^{(o)} - A$. By construction, $L^{(i)}$ has zero column sums and $L^{(o)}$ has zero row sums. The spectrum of the Laplacian reflects several interesting properties of the associated graph, specially in the case of undirected graphs (see for example [69]). A fundamental property is that the Laplacian of an undirected graph has non-negative eigenvalues, the zero eigenvalues corresponding exactly to the different connected components of $G$.

A digraph $G(V, E, A)$ is *strongly connected* if there is a directed path from any vertex $j$ to any vertex $l$ (i.e. a sequence of vertices starting with $j$ and ending with $l$ such that $(v_k, v_{k+1}) \in E$ for any two consecutive vertices $v_k$ and $v_{k+1}$); if there is such a path in the *associated undirected graph*, derived from the adjacency matrix $A + A^T$, then $G$ is *weakly connected*. A connected component of a disconnect graph $G$ is a subset of nodes such that, together with the edges connecting them in $G$, they form a connected graph.

When considering time-varying interconnections, a time-varying graph $G(t)$ is used and all the previously defined elements simply depend on time. Infinitesimally shortly lasting edges can be avoided by requiring the graph to be piecewise continuous. Another frequent requirement in this line of thought is that the elements of $A(t)$ must be bounded and satisfy some threshold $a_{jk}(t) \geq \delta > 0\ \forall (j, k) \in E(t)$ and $\forall t$. A graph $G(t)$ satisfying these assumptions is called a *bounded $\delta$-digraph*. The present paper always considers piecewise continuous, bounded $\delta$-digraphs.

In a $\delta$-digraph $G(V, E, A)$, a vertex $j$ is said to be *connected* to a vertex $k \neq j$ across $[t_1, t_2]$ if there is a path from $j$ to $k$ for the digraph $\bar{G}(V, \bar{E}, \bar{A})$ defined by

$$\begin{aligned} \bar{a}_{jk} \quad &= \quad \begin{cases} \int_{t_1}^{t_2} a_{jk}(t)dt & \text{if } \int_{t_1}^{t_2} a_{jk}(t)dt \geq \delta \\ 0 & \text{if } \int_{t_1}^{t_2} a_{jk}(t)dt < \delta \end{cases} \\ (j,k) \in \bar{E} \quad &\text{iff} \quad \bar{a}_{jk} \neq 0 \ . \end{aligned}$$

A $\delta$-digraph $G(t)$ is called *uniformly connected* if there exist a vertex $k$ and a time horizon $T > 0$ such that $\forall t$, $k$ is connected to all other vertices across $[t, t+T]$.

## 2.2 Systems on compact Lie groups and homogeneous manifolds

The present section only briefly reviews some elements related to Lie groups and homogeneous manifolds, mainly for the purpose of defining the notation used and recalling some key properties used later in this work. It contains no formal definitions nor proofs, as the reader is expected to be familiar with the subject. References about Lie groups abound and, if necessary, it should be possible to review the basics that are required to understand the following report in any of them.

### 2.2.1 Motion on Lie groups and homogeneous manifolds

**Compact Lie groups** The present report considers a swarm of $N$ agents that evolve on a connected compact Lie group $\mathcal{G}$. The position of agent $k$ on $\mathcal{G}$ is denoted by $y_k$. The basic operation on a Lie group $\mathcal{G}$ is the group multiplication $y_k\, y_j$. The group also contains a special identity element $e$ such that $e\, y_k = y_k\, e = y_k$ and an inverse element $y_k^{-1}$ for each $y_k$ such that $y_k^{-1} y_k = y_k\, y_k^{-1} = e$. However, when considering the manifold associated to the Lie group $\mathcal{G}$, it seems intuitively meaningless to "multiply" (or, thinking as on a vector space, to add) two positions in physical space; multiplication (or addition) of two positions would only make sense if a fixed reference (like the identity $e$) was explicitly present. The present work precisely wants to avoid the use of any fixed reference. Therefore, group multiplications should not involve actual positions but only *relative* positions, i.e. the multiplication $y_k\,(y_k^{-1} y_j)$ is admitted but the multiplication $y_k\, y_j$ is not. In some sense, the agents evolve on "a Lie group on which the reference $e$ has been deleted"[2]. In practice, the conclusion of this paragraph should simply be that, though Lie groups usually contain a reference point $e$ and allow multiplication of 2 absolute positions, in the present work the use of a fixed reference is not admitted and everything is formulated in terms of relative positions. Mathematically, this may be formalized by requiring that the behavior of the system remains unchanged if all the agent positions are translated (i.e. multiplied) by the same arbitrary $g \in \mathcal{G}$. Expressions that are invariant with respect to such a translation will be called *shape entities.*

Elements of a Lie group are usually *represented* as invertible matrices. The basic representation of a Lie group is the *adjoint representation.* An important property of compact Lie groups is that their adjoint representation is always unitary, i.e. the matrices $\overline{y}_k$ representing the elements $y_k$ of $\mathcal{G}$ are unitary. For the ease of notation, the present report always assumes real matrices; the formulation for complex Lie groups is straightforwardly obtained by expliciting the real and imaginary parts of each complex number. The matrices representing positions on $\mathcal{G}$ are thus orthogonal, and as $\mathcal{G}$ must be connected, it can be assumed without loss of generality that they belong to the matrix group $SO(n)$. But one should not be mistaken to simply work on $\mathcal{G} \cong SO(n)$ - indeed, in general $\mathcal{G}$ could be any subgroup of $SO(n)$.

[2] Formally, this actually means that the agents evolve on the *principal homogeneous manifold* associated to the Lie group $\mathcal{G}$. The term "Lie group" has been preferred to "principal homogeneous manifold" in the present report because Lie groups are much more popular and the only difference, explained in this remark, can easily be taken into account explicitly.

The tangent space $T\mathcal{G}_e$ at the origin (and actually at any point) of a group $\mathcal{G}$ is a vector space. This vector space can be turned into a *Lie algebra* $\mathfrak{g}$ by defining a Lie bracket $[\cdot, \cdot] : T\mathcal{G}_e \times T\mathcal{G}_e \rightarrow T\mathcal{G}_e$ which must satisfy the following properties.

- Bilinearity: $[a\xi_1 + b\xi_2, \xi_3] = a[\xi_1, \xi_3] + b[\xi_2, \xi_3]$ and $[\xi_3, a\xi_1 + b\xi_2] = a[\xi_3, \xi_1] + b[\xi_3, \xi_2]$ for all $\xi_1, \xi_2, \xi_3 \in \mathfrak{g}$ and $a, b \in \mathbb{R}$.
- Skew-symmetry: $[\xi_1, \xi_2] = -[\xi_2, \xi_1]$ for all $\xi_1, \xi_2 \in \mathfrak{g}$.
- Jacobi identity: $[\xi_1, [\xi_2, \xi_3]] + [\xi_2, [\xi_3, \xi_1]] + [\xi_3, [\xi_1, \xi_2]] = 0$ for all $\xi_1, \xi_2, \xi_3 \in \mathfrak{g}$.

Another important property of the Lie bracket on compact Lie groups is that $[\xi, \eta]$ is orthogonal to $\xi$ and $\eta$ for all $\xi, \eta$.

**Homogeneous manifolds** Formally, a homogeneous manifold $\mathcal{M}$ is a manifold with a transitive group action by a Lie group $\mathcal{G}$: it is isomorphic to the quotient manifold $\mathcal{G}/\mathcal{H}$, where $\mathcal{H}$ is the isotropy group of any point on $\mathcal{M}$ with respect to $\mathcal{G}$. When $H \neq \{e\}$, $\mathcal{M}$ is not isomorphic to a group, but it is still "perfectly symmetric"; indeed, intuitively, a homogeneous manifold is a manifold on which "all points are equivalent". The most popular example of a homogeneous manifold is the sphere $S^2$ (or in larger dimensions, the spheres $S^n$), which corresponds to the quotient of $SO(3)$ by $SO(2)$. Another useful class of compact connected homogeneous manifolds are the Grassmann manifolds $Grass(p, n)$ (see example section below).

**Moving on a manifold** The kinematic motion law for agent $k$ on any manifold $\mathcal{M}$ is written as

$$\dot{y}_k = \zeta_k \tag{1}$$

where $\zeta_k$ must be an element of the tangent space $T\mathcal{M}_{y_k}$ to $\mathcal{M}$ at $y_k$. On a Lie group $\mathcal{G}$, (1) can be rewritten as

$$\dot{y}_k = y_k \xi_k \tag{2}$$

where $\xi_k$ is an element of the *Lie algebra* $\mathfrak{g}$ of $\mathcal{G}$ (equivalent to the tangent space at the origin $e$) and the multiplication[3] by $y_k$ translates $\xi_k$ from the tangent space at $e$ to the tangent space at $y_k$. In shape entities, this yields

$$y_k^{-1}\dot{y}_k = \xi_k$$

where the control input $\xi_k \in \mathfrak{g}$ is a shape entity. Since $y_k$ is represented as a matrix, $\xi_k$ is a matrix of the same dimension. It is however common practice to convert $\xi_k$ into a vector form $\xi_k^\vee$ in order to simplify notations when working on the vector space corresponding to the Lie algebra $\mathfrak{g}$. When $\xi_k = \xi_0$ is constant, the motion of agent $k$ can be integrated using the *exponential operator* $\exp : \mathfrak{g} \rightarrow \mathcal{G}$, such that $y_k(t) = y_k(0)\exp(t\xi_0)$. When $y_k$ and $\xi_0$ are represented by matrices, the exponential operator is equivalent to the matrix exponential. The fact that the matrices representing elements of $\mathcal{G}$ are orthogonal implies that

$$\exp(\xi_0^T t) = (\exp(\xi_0 t))^T = (\exp(\xi_0 t))^{-1} = \exp(-\xi_0 t)$$

which implies that $\xi_0^T = -\xi_0$, meaning that in matrix form, *the elements of* $\mathfrak{g}$ *are all skew-symmetric.* This is consistent with the fact that the tangent space at the identity to the manifold of orthogonal matrices is the space of skew-symmetric matrices.

A frequently encountered motion is along the *gradient* of some function $f$. When computing gradients along a manifold, the metric with respect to which the gradient is computed must be

[3] Actually, this involves a slight abuse of notation. Indeed, since $\xi_k$ is not an element of the group $\mathcal{G}$, group multiplication of $\xi_k$ by $y_k$ is not defined. To be rigorous, one should speak of an *action* of the group element $y_k$ on $\xi_k$. In practice however, there is no formal difference because matrix multiplication is simply used for the group multiplication as well as for the action on vectors of the tangent spaces.

specified. In the present work, using the matrix representations, the manifolds are embedded in a Euclidean space $\mathbb{R}^{n\times n}$ and the gradient is computed as the projection onto $T\mathcal{M}_{y_k}$ of the gradient in $\mathbb{R}^{n\times n}$. For compact Lie groups with the adjoint representation, this is equivalent to the canonical gradient corresponding to the *bi-invariant metric*.

### 2.2.2 Dynamics on compact Lie groups

A compact Lie group is fundamentally different from a vector space. Therefore, the motion of an unforced mechanical system whose generalized position evolves on a compact Lie group is governed by non-trivial dynamical equations. The best-known example of this kind is probably the Euler equation for the rotation of a 3-dimensional rigid body, whose generalized position evolves on $SO(3)$ (see example section below). By analogy, systems whose generalized position evolves on a compact Lie group $\mathcal{G}$ are sometimes called *Lie group solids* for $\mathcal{G}$. The equivalent of the second-order Newton equation $ma_k = F_k$ for Lie group solids consists of two first-order differential equations. The first one, corresponding to $v_k = \dot{x}_k$ in Euclidean space, is equation (2). The second one defines how $\xi_k$ in (2) evolves as a function of the present state and the generalized input torque $\tau_k$. A discussion of its deduction from general principles can be found in Appendix 2 of [70]; the compactness of the Lie groups in the present work significantly simplifies the discussion.

In order to write the second dynamic equation, one must consider the kinetic energy $T_k$ associated to the Lie group solid. In general, it is a quadratic form on $\xi_k$ involving a generalized moment of inertia $J$ (symmetric positive definite matrix of the quadratic form). Converting the matrix $\xi_k$ into a vector form, this can be written as $T_k = \frac{1}{2}(\xi_k^\vee)^T J \xi_k^\vee$ where it becomes clear that the vector $\xi_k^\vee$ is the *generalized angular velocity* "in body frame" of the Lie group solid. Then the *generalized angular momentum* $M_k$ is defined as $M_k^\vee = J\xi_k^\vee$, where again vector forms are considered. The differential equation corresponding to the generalized Euler equation is

$$\dot{M}_k = [M_k,\ \xi_k] + \tau_k \quad \text{or equivalently} \quad J\dot{\xi}_k^\vee = [J\xi_k^\vee,\ \xi_k^\vee] + \tau_k^\vee \tag{3}$$

where $[\,\cdot\,,\,\cdot\,]$ denotes the Lie bracket of the Lie algebra $\mathfrak{g}$ (for matrix or vector representations according to the arguments) and $\tau_k$ is the generalized input torque. Equations (2) and (3) together represent the mechanical model of a Lie group solid, for which control inputs $\tau_k$ are designed in Section 4. The vectors $M_k^\vee, \xi_k^\vee$ and $\tau_k^\vee$ are all expressed "in body frame".

One of the most important operators on Lie algebras is the *adjoint operator* $\mathrm{Ad}_g$. The adjoint operator satisfies

- $\mathrm{Ad}_g[\xi_1,\xi_2] = [\mathrm{Ad}_g\xi_1, \mathrm{Ad}_g\xi_2]$ for all $g \in \mathcal{G}$ and $\xi_1, \xi_2 \in \mathfrak{g}$ and
- $\mathrm{Ad}_g\mathrm{Ad}_h\xi_1 = \mathrm{Ad}_{gh}\xi_1$ for all $g, h \in \mathcal{G}$ and $\xi_1 \in \mathfrak{g}$.

The interest in the adjoint operator arises from the fact that it converts elements of the Lie algebra like $\xi_k$ or $\tau_k$ that are expressed in one frame into elements that are expressed in another frame. For example, if $\xi_k$ expresses the generalized angular velocity of agent $k$ in a frame attached to $y_k$, then $\mathrm{Ad}_{y_j^{-1}y_k}\xi_k$ expresses this same quantity in a frame attached to $y_j$ (in the present work, the adjoint operator is applied to matrix as well as vectorized forms for $\mathfrak{g}$ with no notational distinction). This fact should be clearer after considering the example $SO(3)$.

### 2.2.3 Examples

**The Grassmann manifolds** The Grassmann manifolds are a special class of compact connected homogeneous manifolds. Since they are discussed as an important example in Section 3.5, a brief review of their characteristics is provided.

On the Grassmann manifold $Grass(p, n)$, each point denotes a $p$-dimensional subspace of $\mathbb{R}^n$. The position of agent $k$ on $Grass(p, n)$ is denoted by $\mathcal{Y}_k$ ($p$-subspace representation). Following the analysis carried out in [60], the canonical geometry of $Grass(p, n)$ arises as the quotient manifold of the *non-compact Stiefel manifold* $ST(p, n)$ - that is, the set of $p$-rank $n \times p$ matrices - by the *general linear group* $GL_p$ - that is, the set of full rank $p \times p$ matrices. Alternatively, the same metric arises for $Grass(p, n)$ as the quotient manifold of the *compact Stiefel manifold* $St(p, n)$ - that is, the set of matrices composed of $p$ orthonormal $n$-vectors - by the *orthogonal group* $O(p)$ - that is, the set of $p \times p$ orthonormal matrices. This expresses the homogeneous manifold structure of $Grass(p, n)$. As a consequence, $\mathcal{Y}_k$ may be represented by an arbitrary $n \times p$ matrix $Y_k \in St(p, n)$ that contains $p$ orthonormal vectors spanning $\mathcal{Y}_k$ (basis representation). In this representation, several elements of $St(p, n)$ characterize the same element of $Grass(p, n)$; therefore the term *quotient manifold.*

Equivalently, a point of $Grass(p, n)$ can be represented as in [71] by one of the projectors $\Pi_k$ or $\Pi_{\perp k}$, which are the orthonormal projectors from $\mathbb{R}^n$ onto $\mathcal{Y}_k$ and onto the space orthogonal to $\mathcal{Y}_k$ respectively (projector representation). Explicitly,

$$\begin{aligned} \Pi_k &= Y_k Y_k^T \\ \Pi_{\perp k} &= I_n - Y_k Y_k^T \end{aligned} \tag{4}$$

where $I_n$ denotes the $n \times n$ identity matrix. The bijection that exists between $Grass(p, n)$ and the orthonormal projectors of rank $p$ is a main advantage of this representation, in contrast to the non-uniqueness of the representation of the points on $Grass(p, n)$ by elements of $ST(p, n)$ or $St(p, n)$. The projector representation makes $Grass(p, n)$ an embedded submanifold of the cone $\mathbb{S}_n^+$ of $n \times n$ symmetric positive semi-definite matrices, while the representations by elements of $ST(p, n)$ or $St(p, n)$ are not embeddings. The disadvantage of the projectors is that the dimension of the representation increases from $np$ or $np - p(p+1)/2$ to $n(n+1)/2$. One could further reduce the dimension of the embedding space by simply leaving out one element of the diagonal, because the trace of $\Pi_k$ is fixed to $p$; using this fact, it is shown in [36] that $Grass(p, n)$ can actually be embedded in the sphere of $\mathbb{R}^{n(n+1)/2-1}$. This embedding is not considered here because it would just complicate notations without really simplifying the algorithms. The dimension of $Grass(p, n)$ itself is $p(n - p)$. Since $Grass(n - p, n)$ is isomorphic to $Grass(p, n)$ by identifying orthogonally complementary subspaces, it is assumed throughout the report that $p \leq \frac{n}{2}$.

The simplest Grassmann manifold $Grass(1, 2)$ is isomorphic to the circle $S^1 \cong SO(2)$. The mapping that achieves this isomorphism is built as follows: fixing a reference $r$ on the unit circle centered at the origin $o$, each element $\mathcal{Y}_k$ of $Grass(1, 2)$ - i.e. each line in the plane - makes an angle $\phi_k$ and $\phi_k + \pi$ with the reference direction $\overrightarrow{or}$. Defining $\theta_k = 2\phi_k$, the mapping from $Grass(1, 2)$ to $\theta_k \in S^1$ is one-to-one and conserves the initial metric of $Grass(1, 2)$. The basis representation of $Grass(1, 2)$ by elements of $St(1, 2)$ is the quotient of $(\cos(\phi_k), \sin(\phi_k))$ by the elements $\pm 1$ of $O(1)$. The corresponding projector representation is

$$\Pi_k = \begin{pmatrix} \cos^2(\phi_k) & \sin(\phi_k)\cos(\phi_k) \\ \sin(\phi_k)\cos(\phi_k) & \sin^2(\phi_k) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 + \cos(\theta_k) & \sin(\theta_k) \\ \sin(\theta_k) & 1 - \cos(\theta_k) \end{pmatrix} \tag{5}$$

from which the correspondence with an element $\begin{pmatrix} \cos(\theta_k) & \sin(\theta_k) \end{pmatrix}^T \in S^1$ is obvious.

**The Lie group $SO(n)$** In its canonical representation, a point of the *special orthogonal Lie group* $SO(n)$ is characterized by a real $n \times n$ orthogonal matrix $Q$ with determinant equal to $+1$. $SO(n)$ can be viewed as the set of positively oriented orthonormal bases of $\mathbb{R}^n$, or equivalently as the set of rotation matrices in $\mathbb{R}^n$; hence in practical applications, it is the natural state space for the orientation of a rigid body in $\mathbb{R}^n$. $SO(n)$ has dimension $n(n - 1)/2$. It is easily understood that $SO(2)$ is isomorphic to the circle $S^1$.

In the present report, $SO(3)$ serves as an important example to clarify the developments of Section 4 because the dynamics of the rotating rigid body in 3 dimensions are both well-known and non-trivial. The Lie algebra $\mathfrak{so}(3)$ corresponding to $SO(3)$ is the set of skew-symmetric matrices $\omega^\wedge$. Defining the vectorized form $\omega = (\omega^\wedge)^\vee$ by

$$\omega^\wedge = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \quad \longleftrightarrow \quad \omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} ,$$

the vector $\omega$ is the usual *angular velocity* of the solid expressed in body coordinates. The Lie bracket on the vectorized form of $\mathfrak{so}(3)$ is simply the vector product of the arguments, i.e. $[\omega_k,\, \omega_j] = \omega_k \times \omega_j$, such that (3) becomes the well-known Euler equation

$$J\dot{\omega}_k = (J\omega_k) \times \omega_k + \tau_k \; . \tag{6}$$

The adjoint representation on $\mathfrak{so}(3)$ is expressed on matrix forms by $\mathrm{Ad}_Q\omega^\wedge = Q\omega^\wedge Q^T$ or on vector forms by $\mathrm{Ad}_Q\omega = Q\omega$. These are simply the expressions for coordinate changes on $\mathfrak{g}$ in matrix and vector forms respectively. From this, one immediately concludes that, for instance, if $\omega_k$ denotes the angular velocity of agent $k$ in body frame $k$, then $Q_k\omega_k$ designs this angular velocity in absolute space, and $Q_j^T Q_k\omega_k$ expresses it in the reference frame of body $j$.

# 3 Induced arithmetic mean and consensus reaching on Lie groups

This first part of the work focuses on a definition of consensus and related concepts on compact Lie groups. Methods to achieve consensus in a swarm of agents with restricted interconnections are presented for simple integrator models. On the way, an easily computable “average position” on compact Lie groups is introduced, for which the denomination *induced arithmetic mean* is chosen in reference to the work of [46]. Most results of the present section can be readily generalized to homogeneous manifolds, as is shown at the end of the section. More details about these results, most notably with a deeper treatment of examples, can be found in the paper [1].

## 3.1 The induced arithmetic mean

### 3.1.1 Definition and discussion

Consider a set of $N$ points on a connected compact (CC) Lie group $\mathcal{G}$; the position of a point $k$ is denoted by $y_k$ and its weight by $w_k$. Furthermore, $\mathcal{G}$ is embedded in a Euclidean space $\mathcal{E}$ of dimension $m$; the element of $\mathcal{E}$ corresponding to $y_k$ is denoted $\overline{y}_k$. Any compact Lie group can be represented in this way using the *adjoint representation.* Moreover, the adjoint representation of a compact Lie group is always *unitary*, such that $\|\overline{y}_k\| = cst$ for any $y_k \in \mathcal{G}$, where $\| \cdot \|$ denotes the usual Euclidean norm in $\mathcal{E}$. For notational convenience, the elements $\overline{y}_k$ will mostly be considered as vectors of $\mathbb{R}^m$, though the actual adjoint representation would rather consider them as square matrices; for instance, the norm $\|\overline{y}_k\|^2$ will mostly be written as $\overline{y}_k^T\overline{y}_k$, while a matrix representation would require to write $\text{trace}(\overline{y}_k^T\overline{y}_k)$.

In this setting, the following definition is introduced in this work for the *induced arithmetic mean* and the *anti-[induced arithmetic mean]* of agents on $\mathcal{G}$; the terminology is derived from [46] where this object is called the *projected arithmetic mean* for the special case of $SO(3)$.

**Definition 1:** *Given a set of $N$ agents positioned at $y_k$, $k = 1...N$ on a CC Lie group $\mathcal{G}$ and a set of associated positive weights $w_k$, the* ***induced arithmetic mean (IAM)*** *$C$ of these agents in $\mathcal{G}$ is the set of points in $\mathcal{G}$ that globally minimize the weighted sum of the squared Euclidean distances* in $\mathcal{E}$ *to each of the agents:*

$$C = \underset{c\in\mathcal{G}}{\text{argmin}} \sum_{k=1}^{N} w_k d^2(\overline{y}_k, \overline{c}) \ . \tag{7}$$

*Similarly, the* ***anti-[induced arithmetic mean] (AIAM)*** *$AC$ of these agents in $\mathcal{G}$ is the set of points in $\mathcal{G}$ that globally maximize the weighted sum of the squared Euclidean distances* in $\mathcal{E}$ *to each of the agents:*

$$AC = \underset{c\in\mathcal{G}}{\text{argmax}} \sum_{k=1}^{N} w_k d^2(\overline{y}_k, \overline{c}) \ . \tag{8}$$

The important fact in this definition is that the distances are measured in the embedding space $\mathcal{E}$ and not along the manifold corresponding to $\mathcal{G}$. Explicitly, if $\overline{y}_k$ is considered as a vector of length $m$, then

$$C = \underset{c\in\mathcal{G}}{\text{argmin}} \sum_{k=1}^{N} w_k (\overline{y}_k - \overline{c})^T (\overline{y}_k - \overline{c}) \qquad \text{and} \qquad AC = \underset{c\in\mathcal{G}}{\text{argmax}} \sum_{k=1}^{N} w_k (\overline{y}_k - \overline{c})^T (\overline{y}_k - \overline{c}) \ . \tag{9}$$

The induced arithmetic mean is *not* the canonical mean of points on a Riemannian manifold corresponding to $\mathcal{G}$. Indeed, the latter is defined as the *Karcher mean*, which is the set of points in $\mathcal{G}$ that minimize the weighted sum of the squared distances to all the points *along the manifold*, that is

$$C_{Karcher} = \underset{c\in\mathcal{G}}{\operatorname{argmin}} \sum_{k=1}^{N} w_k d^2(y_k, c) \tag{10}$$

where $d(b, c)$ denotes the Riemannian distance between points $b$ and $c$ along the manifold - in contrast to $d(\overline{b}, \overline{c})$ which denotes the distance in $\mathcal{E}$. Because the curvature is necessarily bounded, the induced arithmetic mean and the Karcher mean become equivalent when the agents are all located in an infinitesimal subset of $\mathcal{G}$.

The induced arithmetic mean $C$ has the following properties.

1. The induced arithmetic mean of a single point $y_1$ is the point itself.
2. $C$ is invariant with respect to permutations of agents of equal weights.
3. $C$ is compatible with an action of the group $\mathcal{G}$ on itself (typically, a left- or right-multiplication of all the $y_k$ by $g \in \mathcal{G}$ implies that $C$ is also multiplied by $g$).
4. $C$ does not always reduce to a single point; however, this feature seems unavoidable for any mean on a CC Lie group (including the Karcher mean).

The main advantage of Definition 1 is that the IAM and AIAM are easily computed, in contrast to the Karcher mean. The latter gets even more problematic to evaluate as soon as the agents are not located in a convex set of $\mathcal{G}$, which causes no difficulty for the IAM; as an illustration, see the local minima $LM_1$ and $LM_2$ appearing in the computation of the Karcher mean on Figure 1, while the IAM is directly found thanks to the following property.

The IAM and AIAM are closely related to the well-known notion of *centroid* in $\mathcal{E}$.

**Definition 2:** *The (weighted)* ***centroid*** $\overline{C}_e$ *in* $\mathcal{E}$ *of* $N$ *agents located at* $\overline{y}_1 ... \overline{y}_N$ *in* $\overline{\mathcal{G}} \subset \mathcal{E}$ *is defined as*

$$\overline{C}_e = \frac{1}{W} \sum_{k=1}^{N} w_k \overline{y}_k \ , \qquad W = \sum_{k=1}^{N} w_k \ . \tag{11}$$

Since the norm $\|\overline{c}\|^2$ is constant, by considering the $\overline{y}_k$ as $m$-vectors, one easily verifies that alternative definitions for the IAM and AIAM are

$$C = \underset{c\in\mathcal{G}}{\operatorname{argmax}}(\overline{c}^T \overline{C}_e) \qquad \text{and} \qquad AC = \underset{c\in\mathcal{G}}{\operatorname{argmax}}(-\overline{c}^T \overline{C}_e) \ . \tag{12}$$

Hence, the computation of the IAM and AIAM just involves the search for the global maxima of a linear function on $\mathcal{E}$ in a very regular search space (namely, a CC Lie group).

Local maximization methods even suffice provided that the linear function has no maxima on $\mathcal{G}$ other than the global maxima. It turns out that this is the case for many Lie groups and homogeneous manifolds, including $SO(n)$ and $Grass(p, n)$ with the embeddings of the present paper, as well as the canonically embedded $n$-dimensional sphere in $\mathbb{R}^{n+1}$. In absence of a formal proof in the literature, the following blanket assumption is formulated; it must be seen as a condition on the Lie group, maybe including the way it is embedded, which is emphasized with the notation $\overline{\mathcal{G}}$.

**Assumption 1:** *The local maxima of a linear function* $f(\overline{c}) = \overline{c}^T \overline{b}$ *over* $\overline{c} \in \overline{\mathcal{G}}$*, with* $\overline{b}$ *fixed in* $\mathcal{E}$*, are all global maxima as well.*

### 3.1.2 Examples

**The circle:** The canonical embedding of $SO(2) \cong S^1$ in $\mathbb{R}^2$ satisfies Assumption 1. $\overline{C}_e$ is simply the average of the corresponding positions in $\mathbb{R}^2$ and $C$ is its central projection onto the circle. Hence $C$ corresponds to the whole circle when $\overline{C}_e = 0$ (i.e. the centroid in $\mathbb{R}^2$ is located at the center of the circle) and reduces to a single point in other situations; using a polar representation of $\overline{C}_e \in \mathbb{R}^2 \cong \mathbb{C}$, this writes

$$C = \begin{cases} \arg(\overline{C}_e)\,, & \overline{C}_e \neq 0 \\ S^1\,, & \overline{C}_e = 0\,. \end{cases}$$

The Karcher mean may also contain multiple points when $\overline{C}_e = 0$ (for example, when all the agents are uniformly distributed around the circle), but generally not the whole circle. The Karcher mean uses the arc length between two points as their distance, while the IAM considers the “chordal” distance in $\mathbb{R}^2$. The difference between IAM and Karcher mean is illustrated on Figure 1.

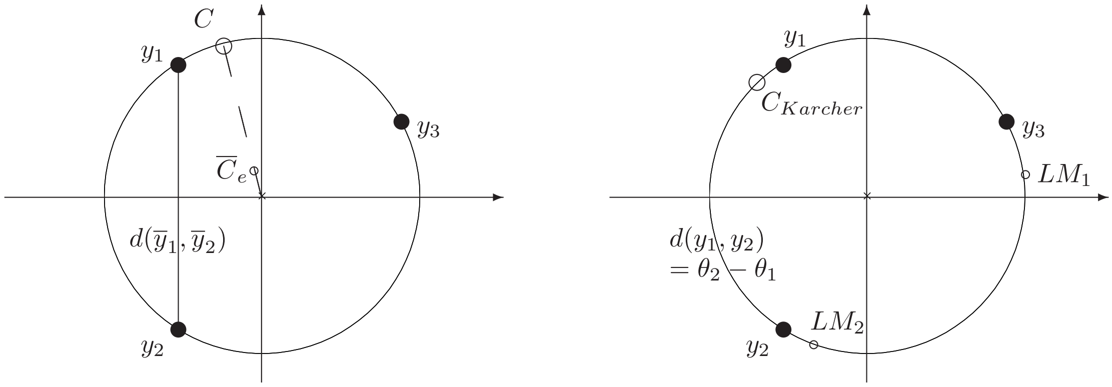

Figure 1: Induced arithmetic mean $C$ and Karcher mean $C_{Karcher}$ of 3 equally-weighted points on the circle. In addition to the different positions of the two means, note the local minima $LM_1$ and $LM_2$ appearing in the computation of the Karcher mean.

**The special orthogonal group:** $SO(n)$ is embedded in $\mathbb{R}^{n\times n}$ with the usual matrix representation. The group $SO(n)$ acts by matrix multiplication on $\mathbb{R}^{n\times n}$ and the Frobenius norm of an orthonormal matrix is $\|Q_k\| = \sqrt{\text{trace}(Q^TQ)} = \sqrt{\text{trace}(I_n)} = \sqrt{n}$.

On $SO(n)$, $\overline{C}_e = \sum_k Q_k$ is a general $n \times n$ matrix. The induced arithmetic mean is linked to the *polar decomposition* of $\overline{C}_e$: any square matrix $B$ can be decomposed into a product $UR$ where $U$ is orthogonal and $R$ is symmetric positive semi-definite; $R$ is always unique, $U$ is unique if $B$ is non-singular [72]. The matrix $U$ obtained is the point in $O(n)$ that is closest to $B$ according to the canonical distance of $\mathbb{R}^{n\times n}$. As a consequence, when $\det(\overline{C}_e) \geq 0$, the induced arithmetic mean contains the matrices $U$ with positive determinant obtained from the polar decomposition of $\overline{C}_e$; this fact has already been noticed and proven in [46]. When $\det(\overline{C}_e) \leq 0$, the result is somewhat more complicated but also has a closed-form solution.

**Proposition 1:** *The induced arithmetic mean $C$ of $N$ points on $SO(n)$ is characterized as follows.*

- *If* $\det(\overline{C}_e) \geq 0$*, $C$ contains the matrices $U$ with positive determinant resulting from the polar decomposition $UR$ of $\overline{C}_e$; it reduces to a single point when the multiplicity of* 0 *as an eigenvalue of $\overline{C}_e$ is less or equal to 1.*

- *If* $\det(\overline{C}_e) \leq 0$, $C = UHJH^T$ *where* $U$ *is an orthogonal matrix of negative determinant resulting from the polar decomposition* $UR$ *of* $\overline{C}_e$, $H$ *contains the orthonormalized eigenvectors of* $R$ *with an eigenvector corresponding to the smallest eigenvalue of* $R$ *in the first column, and*

$$J = \begin{pmatrix} -1 & 0 \\ 0 & I_{n-1} \end{pmatrix} .$$

  *Now* $C$ *reduces to a single point when the smallest eigenvalue of* $R$ *has multiplicity* $1$.

Proof: The proof is postponed because it makes use of calculations presented later in the section. Specifically, it will first be shown that $SO(n)$ satisfies Assumption 1. Then the analytical form of the IAM is established by selecting the local maxima among the critical points of (12).

$\triangle$

These examples exclusively considered the induced arithmetic mean; note that from (12), the conclusions are trivially modified for the anti-[induced arithmetic mean] by replacing $\overline{C}_e$ with $-\overline{C}_e$.

## 3.2 Consensus as an optimization task

### 3.2.1 Definition

Consider a set of $N$ agents on a CC Lie group satisfying Assumption 1 and denote their embedded positions by $\overline{y}_k$. Suppose that the agents are interconnected according to a fixed digraph $G$ of adjacency matrix $A = [a_{jk}]$. In the remainder of this report equal weights $w_k = 1$ are assigned to all agents for notational convenience; the generalization to weighted agents is straightforward. The following definitions are introduced in the present work.

**Definition 3:** *$N$ agents are said to have reached **synchronization** when they are all located at the same position on $\mathcal{G}$.*

**Definition 4:** *$N$ agents are said to have reached a **consensus configuration** with communication graph $G$ if each agent $k$ is located at a point of the induced arithmetic mean of its neighbors $j \rightsquigarrow k$, weighted according to the strength of the communication links:*

$$y_k \in \underset{c \in \mathcal{G}}{\operatorname{argmax}} \left( \overline{c}^T \sum_{j=1}^{N} a_{jk} \overline{y}_j \right) \qquad \forall k \, . \tag{13}$$

*Similarly, $N$ agents are said to have reached an **anti-consensus configuration** when the previous definition is satisfied by replacing the IAM by the AIAM:*

$$y_k \in \underset{c \in \mathcal{G}}{\operatorname{argmin}} \left( \overline{c}^T \sum_{j=1}^{N} a_{jk} \overline{y}_j \right) \qquad \forall k \, . \tag{14}$$

The consensus defined by (13-14) is graph-dependent; this can be interpreted as the fact that each agent considers that it has reached consensus when it is located at the best possible place according to the agents from which it receives information. In the case of a tree or an equally-weighted complete graph $G_c$, consensus means synchronization.

**Proposition 2:** *When $G$ is a tree or $G = G_c$, the only possible consensus configuration is synchronization.*

Proof: For the complete graph, according to Definition 4, at consensus the property

$$\overline{y}_k^T \sum_{j \neq k} \overline{y}_j \geq \overline{c}^T \sum_{j \neq k} \overline{y}_j$$

must be satsified for all $k$ and any $c \in \mathcal{G}$. Furthermore, it is obvious that $\overline{y}_k^T \overline{y}_k > \overline{y}_k^T \overline{c}$ for any $\overline{c} \neq \overline{y}_k \in \mathcal{G}$. As a consequence, at consensus

$$\overline{y}_k^T \sum_{j=1}^{N} \overline{y}_j > \overline{c}^T \sum_{j=1}^{N} \overline{y}_j$$

for any $\overline{c} \neq \overline{y}_k \in \mathcal{G}$. According to (12), this means that $\overline{y}_k$ is located at the IAM of *all* the agents (including itself), and moreover that the latter reduces to a single point. Since this must hold for all $k$, all the agents must be located at this single point (which is then trivially their IAM).

For the tree, start with all agents fixed except a leaf $k$. The obvious unique IAM of the neighbors of $k$ is the position of its parent; therefore all leaves must be synchronized with their parent. Now consider synchronized moves of a parent $j$ and its leaves. As the leaves follow $j$ everywhere, the obvious global minimum in (7) occurs when $j$ is synchronized with its own parent, which brings us back to the previous situation. An inductive argument is then used up to the root.

△

Synchronization is a configuration of complete consensus. It is the only consensus configuration common to all graphs. The opposite of synchronization is more difficult to define because there exists no anti-consensus configuration common to all graphs. A meaningful property to characterize a configuration of complete anti-consensus would be to require that the IAM of the agents is the entire manifold $\mathcal{G}$. This is called a *balanced* configuration in the present work.

**Definition 5:** *$N$ agents are said to be in a* ***balanced*** *configuration when their induced arithmetic mean is the entire manifold $\mathcal{G}$.*

Balancing implies some spreading of the agents on the manifold. It can be a meaningful objective in several applications. Nevertheless, a full characterization of balanced configurations seems complicated. Balanced configurations do not always exist (typically, when the number of agents is too small relative to the manifold dimension) and are mostly not unique (they can appear in qualitatively different forms). Finally, the following link exists between anti-consensus for $G_c$ and balancing.

**Proposition 3:** *All balanced configurations are anti-consensus configurations for $G_c$.*

Proof: Note that for the equally-weighted complete graph, (14) can be written as

$$y_k \in \operatorname*{argmin}_{c \in \mathcal{G}} \left( \overline{c}^T (N C_e - \overline{y}_k) \right) \qquad \forall k \; . \tag{15}$$

Assume that the agents are in a balanced configuration. This means that $f(\overline{c}) = \overline{c}^T \overline{C}_e$ must be constant over $c \in \mathcal{G}$. Therefore condition (15) reduces to $y_k = y_k$ which is trivially satisfied.

△

Note that in contrast to Proposition 2, Proposition 3 does not establish a necessary and sufficient condition: anti-consensus configurations for $G_c$ that are not balanced do exist, though they seem exceptional. The examples below illustrate these considerations.

### 3.2.2 Examples

**The circle:** In [13] and [14], (anti-)consensus configurations are fully characterized for an equally-weighted complete graph on the circle $S^1 \cong SO(2)$. It is shown that (for $N > 1$) the only anti-consensus configurations that are not balanced occur for $N$ odd and correspond to $(N+1)/2$ agents at one position and $(N-1)/2$ agents at the opposite position on the circle. It is not difficult to show that balanced configurations are unique for $N = 2$ and $N = 3$, while there is a continuum of such configurations for $N > 3$.

Another situation where (anti-)consensus configurations are not too difficult to characterize is for the equally-weighted *undirected ring interconnection graph* in which each agent $k$ is connected to two neighbors such that the graph forms a single closed undirected path. In this case, regular configurations are states where consecutive agents in the path are all separated by the same angle $\chi \geq 0$; it turns out that $\chi \leq \pi/2$ corresponds to consensus configurations and that $\chi \geq \pi/2$ corresponds to anti-consensus configurations. In addition, for $N \geq 4$, there are irregular consensus and anti-consensus configurations where non-consecutive angles of the regular configurations are replaced by $(\pi - \chi/2)$. As a consequence:

- There are several possible qualitatively different consensus and anti-consensus configurations.
- There are consensus and anti-consensus states which correspond to equivalent configurations when discarding the underlying graph. For example, the positions occupied by the agents are strictly equivalent for 7 agents separated by $2\pi/7$ (consensus) or separated by $4\pi/7$ (anti-consensus); the only difference, based on *which agent* is located at *which position*, concerns the way the communication links are drawn.
- There may even be degenerate configurations that correspond to consensus and anti-consensus (for example when consecutive agents are separated by $\pi/2$ for $N = 4, 8, ...$); this singularity is specific to the undirected ring graph.
- There is no common anti-consensus state for all possible ring graphs. Indeed, considering an agent $k$, a common anti-consensus state would require that for any two other points selected as neighbors of $k$, either they are separated by $\pi$ or they are at both sides of $k$ at a distance $\chi \geq \pi/2$; one easily verifies that this cannot be satisfied for all $k$.

**The special orthogonal group:** Simulations of the algorithms proposed in this paper suggest that balanced configurations always exist for $N \geq 2$ when $n$ is even and for $N \geq 4$ when $n$ is odd. Furthermore, convergence to an anti-consensus state for $G_c$ that is not balanced is never observed.

### 3.2.3 Consensus optimization strategy

The presence of a maximization condition in the definitions of the previous sections naturally points to the use of optimization methods to compute (anti-)consensus configurations. As a consequence, it is natural to introduce a cost function whose optimization leads to (anti-)consensus configurations.

For a given graph $G$ with adjacency matrix $A = [a_{jk}]$ and associated Laplacian $L^{(i)} = [l^{(i)}_{jk}]$, the cost function $P_L$ is defined as

$$P_L(y) = \frac{1}{2N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} a_{jk} \overline{y}_j^T \overline{y}_k = cst_1 - \frac{1}{4N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} a_{jk} \|\overline{y}_j - \overline{y}_k\|^2 \tag{16}$$

where $y$ symbolizes $(y_1, ...y_N)$ and the constant $cst_1 = \frac{\|\overline{y}_1\|^2}{4N^2} \sum_k \sum_j a_{jk}$. The index $L$ refers to the fact that (16) can also be written as a quadratic form on the graph Laplacian:

$$P_L(y) = cst_2 - \frac{1}{2N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} l_{jk}^{(i)} \overline{y}_j^T \overline{y}_k \tag{17}$$

where the constant $cst_2 = \frac{\|\overline{y}_1\|^2}{2N^2} \sum_k d_k^{(i)}$. In [55] and [14], this form of $P_L$ is studied on the circle for undirected equally-weighted graphs.

For the unit-weighted complete graph, the cost function $P := P_L + \frac{\|\overline{y}_1\|^2}{2N}$ is proportional to the squared norm of the centroid $\overline{C}_e$:

$$P(y) = \frac{1}{2} \|\overline{C}_e\|^2 \; . \tag{18}$$

This is a classical measure of the synchrony of phase variables on the circle $S^1$, that has been used for decades in the literature on coupled oscillators; in the context of the Kuramoto model, $P(y)$ has been called the "complex order parameter"[4]. In [13], $P$ is used to derive gradient algorithms for synchronization (by maximizing (17)) or balancing (by minimizing (17)) on $S^1$.

**Proposition 4:** *Synchronization of the $N$ agents on $\mathcal{G}$ is the unique global maximum of $P_L$ whenever the graph $G$ associated to $L^{(i)}$ is weakly connected.*

<u>Proof:</u> Obvious from the second form of (16).

△

**Proposition 5:** *Given $N$ agents on a CC Lie group $\mathcal{G}$ that satisfies Assumption 1, consider an undirected graph $G$ and the associated cost function $P_L(y)$ as defined by (16). A local maximum of $P_L$ necessarily corresponds to a consensus configuration and a local minimum of $P_L$ necessarily corresponds to an anti-consensus configuration for $G$.*

<u>Proof:</u> Suppose that the system has reached an extremum $y^* = (y_1^*, ...y_N^*)$ of $P_L$. When all $y_j = y_j^*$ are fixed except for a particular agent $k$, since $A = A^T$, the variation of $P_L$ takes the form

$$P_L(y_k) = cst_3 + \frac{1}{N^2} \overline{y}_k^T \sum_{j=1}^{N} a_{jk} \overline{y}_j^*$$

of which $y_k^*$ must be a local maximum (resp. minimum). According to Assumption 1, the extrema of the linear function $P_L(y_k)$ are all global extrema. Hence $y_k^*$ must be an element of the IAM (resp. AIAM) considered in Definition 4. Since this must be true for all $k$, Definition 4 is satisfied at $y^*$.

△

Note that Proposition 5 establishes that a *sufficient* condition for finding (anti-)consensus configurations is to optimize $P_L$. However, nothing guarantees that this condition is also *necessary.* In general, optimizing $P_L$ will thus provide some (anti-)consensus configurations, but not necessarily all of them. An exception to this rule is synchronization: since this is the only consensus configuration for $G_c$, it must correspond to the (unique) maximum of $P$, such that in this particular case, Proposition 5 is also sufficient.

In the remainder of this section, algorithms that drive the swarm to (anti-)consensus configurations are presented. These algorithms are based on the optimization of the cost functions $P_L$ or $P$; as a consequence of the previous remark, they do not target all possible (anti-)consensus configurations.

[4] This terminology stems from the fact that $\mathbb{R}^2$ is usually identified with the complex plane in that context.

### 3.2.4 Example

In practice, it is interesting to write $P_L$ using matrix forms for the embedding. On $SO(n)$, one obtains

$$P_L = \frac{1}{2N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} a_{jk} \text{trace}(Q_j^T Q_k) \ . \tag{19}$$

Observing that the trace is maximal for the identity matrix and considering the particular case of $SO(2)$, one can easily imagine how the trace of $Q_j^T Q_k = Q_j^{-1} Q_k$ can characterize the distance between $Q_j$ and $Q_k$ on $SO(n)$. Indeed, particularizing to the circle, the trace measure of $SO(2)$ yields $2\cos(\theta_j - \theta_k)$, such that the basic element of $P_L$ is a sum of cosines of the phase differences between the agents. This is exactly equal to the traditional measure of synchrony used to study the Kuramoto model (see [13] or [55]).

## 3.3 Gradient consensus algorithms

After the previous definitions, attention is now turned towards algorithms that lead the agents of a swarm towards consensus states *in a distributed way*; in particular, the option to compute a consensus state in closed form and drive each agent towards it is rejected, in favor of a situation where each agent (simple integrator) adjusts its own motions according to the relative positions of its neighbors (defined by the interconnection graph). The previous observations point to the use of ascent and descent algorithms for $P$ in order to achieve synchronization and balancing respectively.

### 3.3.1 Gradient algorithms for fixed undirected graphs

A gradient algorithm for $P_L$ yields the update equation

$$\dot{\overline{y}}_k(t) = 2N^2 \alpha \, D_{k,\overline{\mathcal{G}}}(P_L) \tag{20}$$

where $\alpha > 0$ (resp. $\alpha < 0$) for consensus (resp. anti-consensus), $\dot{\overline{y}}_k$ denotes the time-derivative of the agent's position with respect to a fixed reference frame and $D_{k,\overline{\mathcal{G}}}(s)$ denotes the gradient of $s$ with respect to $\overline{y}_k$ along the group manifold $\mathcal{G}$. In practice, the gradient can be computed according to the standard way for embedded submanifolds of $\mathbb{R}^m$ (projecting $D_{k,\mathcal{E}}$, the gradient in $\mathbb{R}^m$, onto the tangent space to $\overline{\mathcal{G}}$) or directly along the manifold (see [52]). The second approach may be more efficient if the dimension of $\mathcal{G}$ is substantially lower than $m$. The first approach yields

$$D_{k,\mathcal{E}}(P_L) = \frac{1}{2N^2} \sum_j (a_{jk} + a_{kj}) \overline{y}_j$$

which leads to the reformulation of (20) as

$$\dot{\overline{y}}_k(t) = \alpha \, \text{Proj}_{T\overline{\mathcal{G}},k} \left( \sum_j (a_{jk} + a_{kj}) \overline{y}_j \right) = \alpha \, \text{Proj}_{T\overline{\mathcal{G}},k} \left( \sum_j (a_{jk} + a_{kj})(\overline{y}_j - \overline{y}_k) \right) \tag{21}$$

where $\text{Proj}_{T\overline{\mathcal{G}},k}$ denotes the projection from $\mathcal{E}$ onto the tangent space to $\overline{\mathcal{G}}$ at $\overline{y}_k$; the last equality comes from the fact that $\text{Proj}_{T\overline{\mathcal{G}},k}(\overline{y}_k) = 0$. It shows that in order to implement this consensus algorithm, each agent $k$ must know the relative position with respect to itself of all the agents $j$

such that $j \rightsquigarrow k$ or $k \rightsquigarrow j$. Since the information flow is restricted to $j \rightsquigarrow k$, the algorithm can only be implemented for undirected graphs, for which it becomes

$$\dot{\overline{y}}_k(t) = 2\alpha \operatorname{Proj}_{T\overline{\mathcal{G}},k} \left( \sum_j a_{jk}(\overline{y}_j - \overline{y}_k) \right) . \tag{22}$$

In the special case of a complete unit-weighted graph, a formulation equivalent to (22) is

$$\dot{\overline{y}}_k(t) = 2\alpha N \operatorname{Proj}_{T\overline{\mathcal{G}},k} \left( \overline{C_e} - \overline{y}_k \right) . \tag{23}$$

**Proposition 6:** *Consider a group of $N$ agents moving according to the update law (22) on a CC Lie group $\mathcal{G}$ that satisfies Assumption 1. This swarm always converges to a set of equilibrium points. If $\alpha < 0$, the only stable equilibria are anti-consensus configurations for the undirected graph $G$ associated to $A = [a_{jk}]$. If $\alpha > 0$, the only stable equilibria are consensus configurations for $G$; in particular, for the complete graph $G = G_c$, the only stable configuration is synchronization.*

Proof: Obvious from the properties of gradient systems and the previous discussion.

△

### 3.3.2 Extension to directed and time-varying graphs

Formally, Algorithm (22) can be written for directed and even time-varying graphs. However, the gradient descent property is lost for directed graphs; obviously, it has no meaning in the time-varying case, since the form of the cost function $P_L$ would change with time. Nevertheless, some weak results about synchronization can be provided for the general case of (22) with varying and directed graphs.

A positive theoretical fact is that the synchronized state still is a stable equilibrium; it is asymptotically stable if disconnected graph sequences are excluded. Its basin of attraction includes the configurations where all the agents are located in a convex set of $\mathcal{G}$. Indeed, the existing convergence results on Euclidean spaces can be adapted to manifolds when all the agents are located in a convex set (see e.g. [43]). On the other hand, it is already mentioned in [55] that examples where algorithm (22) with $\alpha > 0$ runs into a limit cycle can be built for as simple cases as undirected equally-weighted (but varying) graphs on the circle.

Concerning experimental results, simulations on $SO(n)$ seem to indicate that for randomly generated digraph sequences[5], the swarm eventually converges to synchronization when $\alpha > 0$; this would correspond to *generic* convergence (i.e., probability 1 convergence in the absence of constraints on the graphs).

Finally, it must be pointed out that the present synchronization algorithm can lead to a generalization of Vicsek's phase update law (see [41]) to CC Lie groups. The Vicsek model is a discrete-time algorithm that governs the headings of particles in the plane, and hence operates on the circle $SO(2)$. Using the definitions introduced in the present report, it can be written as

$$y_k(t+1) \in \operatorname{IAM}\left(\{y_j(t) | j \rightsquigarrow k \text{ in } G(t)\} \cup \{y_k(t)\}\right) \tag{24}$$

where the neighborhood relations depend on the positions of the particles in the plane (so-called "proximity graphs"). In this form, the Vicsek law can be readily generalized to any CC Lie group. From the previous discussions, one easily understands why (24) can be viewed as a discrete-time variant of (22). See [55] for a precise relationship in the particular case of algorithms on the circle.

[5] More precisely, the following distribution was examined: initially, the elements $a_{jk}$ of $A$ take a value in $\{0, 1\}$ according to an independent identically distributed probability Prob$(1) = p$. The corresponding graph remains for a time $t_{graph}$ randomly chosen in $[t_{min}, t_{max}]$ with a uniform distribution, after which a new graph is built as initially.

### 3.3.3 Examples

**The circle:** Consensus algorithms on the circle are studied in [13, 55, 47, 14]. The specific form of (22) for $S^1$ is

$$\dot{\theta}_k = \beta \sum_{j \rightsquigarrow k} \sin(\theta_k - \theta_j) \ . \tag{25}$$

For $G = G_c$, this is strictly equivalent to the famous Kuramoto model [39] with identical (zero) natural frequencies.

The present report just adds an illustration that shows how algorithm (22) can run into a limit cycle for varying graphs. Consider an equally-weighted ring graph $G_1$ and assume that the swarm is in a consensus state (local maximum of $P_{L_1}$) with consecutive agents spaced by $\chi \in (0, \pi/2)$. Now consider the graph $G_2$ obtained by connecting each agent to the agents that are located at an angle $\psi > \pi/2$ from itself with $\psi$ properly fixed. $G_2$ is either a ring graph or a collection of disconnected ring graphs. Moreover, the swarm is at a local minimum of $P_{L_2}$. Hence, starting the system in the neighborhood of that state and regularly switching between $G_1$ and $G_2$, the system will oscillate in the neighborhood of this particular state, being driven away by $G_2$ and brought back by $G_1$ if consensus is intended and reversely if anti-consensus is intended.

**The special orthogonal group:** It is a well-known fact of group theory that the tangent space to $SO(n)$ at the identity $I_n$ is the space of skew-symmetric $n \times n$ matrices. Using the group action, the projection of $B \in \mathbb{R}^{n \times n}$ onto the tangent space to $SO(n)$ at $Q_k$ is $Q_k \text{Skew}(Q_k^{-1} B) = Q_k(\frac{Q_k^T B}{2} - \frac{B^T Q_k}{2})$. This leads to the following explicit form of algorithm (22) on $SO(n)$ where the right-hand side only depends on relative positions of the agents with respect to $k$:

$$Q_k^{-1} \dot{Q}_k = \alpha \sum_j a_{jk} \left( Q_k^{-1} Q_j - Q_j^{-1} Q_k \right) \ . \tag{26}$$

Knowing the formula for the gradient of a function defined on $SO(n)$ and using the following Lemma 1, it can now be proven that $SO(n)$ satisfies Assumption 1, such that (26) leads to synchronization for $\alpha > 0$ and $G = G_c$. The proof of this result also includes the proof of Proposition 1.

**Lemma 1:** *The matrix $Q^T B$ with $Q \in SO(n)$ and $B \in \mathbb{R}^{n \times n}$ is symmetric iff $Q = UHJH^T$ where*

$$J = \begin{pmatrix} -I_l & 0 \\ 0 & I_{n-l} \end{pmatrix} ,$$

*$B = UR$ is a polar decomposition of $B$, $l$ is even if $\det(U) > 0$ and odd if $\det(U) < 0$, and the columns of $H$ contain (orthonormalized) eigenvectors of $R$.*

Proof: It is easy to verify that all matrices $Q$ of the given form satisfy the requirement that $Q^T B$ is symmetric. The following constructive proof shows that this is the only possible form.

Obviously, $U^T B = R$ is symmetric with $U \in O(n)$. Define $T = U^T Q \in O(n)$. This reformulates the problem as finding all matrices $T \in O(n)$ such that $S = T^T R$ is symmetric and $\det(T) = \det(U)$. Without loss of generality, work in a basis of eigenvectors $H^*$ diagonalizing $R$ with its eigenvalues placed in decreasing order $\lambda_1 \geq \lambda_2 ... \geq \lambda_n \geq 0$. The $j^{th}$ column of $S$ is simply the $j^{th}$ column of $T$ multiplied by the corresponding eigenvalue $\lambda_j$.

- First consider the case where two or more eigenvalues are equal, $\lambda_i = \lambda_j$. In this case, the corresponding submatrix of $T$ can be arbitrary, but $H^*$ may be chosen such that it is diagonal.

- Now consider that $\lambda_{p+1} = 0$, $\lambda_p \neq 0$. If $S$ is symmetric, this implies that the submatrix $T(n-p : n, 1 : p)$ obtained from the intersection of the last $n - p$ rows and the first $p$

columns of $T$ must be identically 0. Moreover, from the previous argument, the submatrix $T(n-p:n,\, n-p:n)$ is diagonal. Hence the only non-zero element in the last $n-p$ rows of $T$ is on the diagonal. Since the rows and columns of $T$ are normalized, this implies that $T(1:p,\, n-p:n)$ must also be identically zero.

- Finally, consider an index $i_-$ and denote by $i_+$ the index of the first eigenvalue $\lambda_{i_+} < \lambda_{i_-}$. Note that
$$\sum_{j=1}^{n} T_{i_-j}^2 = \sum_{j=1}^{n} T_{ji_-}^2 = 1 \text{ (orthogonality) and } \sum_{j=1}^{n} S_{i_-j}^2 = \sum_{j=1}^{n} S_{ji_-}^2 \text{ (symmetry).} \qquad (27)$$
The following is an inductive argument. Start with $i_- = 1$ and assume that $\lambda_{i_+} > 0$. (27) can only be satisfied if $T_{jk} = T_{kj} = 0$, $\forall j \geq i_+$ and $\forall k- \in [i_, i_+)$; using the first item, this actually means $T_{jk} = T_{kj} = 0 \ \forall j \neq k$ and $\forall k \in [i_-, i_+)$. The present argument can be repeated by defining the new $i_-$ as being the previous $i_+$, until this leads to $\lambda_{i_-} = \lambda_n > 0$ or to a new $i_+$ with $\lambda_{i_+} = 0$; the latter case is covered by the second item.

In every case, $T$ ends up being diagonal. The orthogonality of $T$ only allows values 1 or $-1$ on this diagonal; the number $l$ of $-1$ elements must be compatible with the requirement $\det(T) = \det(U)$.

The final form is simply obtained by returning from the eigenbasis of $R$ to the general basis and reordering the eigenvectors such that those corresponding to $-1$ elements of $T$ appear in the first columns.

△

**Proposition 7:** *The Lie group $SO(n)$ satisfies Assumption 1.*

Proof (+ Prop.1): Consider a linear function $f(Q) = \text{trace}(Q^T B)$ as in Assumption 1, with $Q \in SO(n)$ and $B \in \mathbb{R}^{n\times n}$. The computation $D_{k,\mathbb{R}^{n\times n}}(f) = B$ leads to $D_{k,SO(3)}(f) = \frac{Q}{2}(Q^T B - B^T Q)$. Since $Q$ is invertible, critical points of $f$ appear when $(Q^T B - B^T Q) = 0$, which means that they are of the form described by Lemma 1. Using the notations of Lemma 1, write $R = H \Lambda H^T$ where $\Lambda$ contains the (non-negative) eigenvalues of $R$. This leads to $Q^T B = H J \Lambda H^T$, whose trace is equal to
$$f(Q) = -\sum_{j=1}^{l} \Lambda_{jj} + \sum_{j=l+1}^{n} \Lambda_{jj} \, .$$
One readily verifies that

- if $l \geq 2$, any infinitesimal rotation on the submatrix $J(i,j)$ with $i \in \{1,2\}$ and $j \in \{1,2\}$ increases $f(Q)$ unless $\Lambda_{11} = \Lambda_{22} = 0$.
- if $l = 1$ and there is some $m$ such that $\Lambda_{mm} < \Lambda_{11}$, then any infinitesimal rotation on the submatrix $J(i,j)$ with $i \in \{1,m\}$ and $j \in \{1,m\}$ increases $f(Q)$.

This shows that all maxima of $f(Q)$ are global maxima and characterizes the IAM when selecting $B = \overline{C}_e$. Indeed,

- if $\det(B) \geq 0$, any local maximum requires $l = 0$ such that $Q = U$ and $f(Q)$ is equal to the sum of the eigenvalues of $R$;
- if $\det(B) \leq 0$, any local maximum requires that $U$ takes the form of Lemma 1 with $l = 1$, and $\Lambda_{11} \leq \Lambda_{mm} \forall m$; this implies that the first column-vector of $H$ corresponds to a smallest eigenvalue of $R$ and $f(Q)$ is equal to the sum of the $n-1$ largest eigenvalues of $R$ minus the smallest one.

△

The algorithms obtained on $SO(n)$, and in fact on any CC Lie group, are all strict extensions of the ones of [13, 14] for undirected graphs on the circle $SO(2)$. The reader who rewrites the algorithms for $SO(2)$ in terms of angles $\theta_k$ will obtain update equations of the form (25).

## 3.4 Consensus algorithms with estimator variables

The gradient-type algorithms lead to consensus situations which are linked to the interconnection graph. However, in many applications, the interconnection graph is just a restriction on communication possibilities, under which one actually wants to drive the swarm towards a consensus for the complete graph. Moreover, allowing directed and time-varying communication graphs is desirable for robustness. The following algorithms achieve the same performance as the previous gradient algorithms for $G = G_c$ - that is, driving the swarm to synchronization on one hand, and to a subset of the anti-consensus configurations for $G_c$ (which seems to contain little more than balancing) on the other hand - under very weak conditions on the required communication graph.

However, this reduction of information channels must be compensated for by adding an *estimator variable* $\overline{x}_k \in \mathcal{E}$ to the state space of each agent. It is assumed that interconnected agents are able to communicate their estimator variables. This idea was originally developed for the circle, as published in [47, 38, 14].

### 3.4.1 Synchronization algorithm

For synchronization purposes, the agents implement a consensus algorithm for their estimator variables in $\mathcal{E}$, while the positions $\overline{y}_k$ of the agents on $\mathcal{G}$ independently track (the projection of) $\overline{x}_k$. This leads to the synchronization algorithm

$$\begin{aligned} \dot{\overline{x}}_k &= \beta \sum_{j=1}^{N} a_{jk} (\overline{x}_j - \overline{x}_k) && (28) \\ \dot{\overline{y}}_k &= \gamma_S \, D_{k,\overline{\mathcal{G}}}(\overline{y}_k^T \overline{x}_k) = \gamma_S \operatorname{Proj}_{T\overline{\mathcal{G}},k}(\overline{x}_k) \,, && (29) \end{aligned}$$

where $\overline{y}_k$ and $\overline{x}_k$ are represented as $m$-dimensional vectors and $\beta, \gamma_S$ are positive scalars. Equation (28) is a classical consensus algorithm in the Euclidean space $\mathcal{E} \cong \mathbb{R}^m$, where $\dot{\overline{x}}(t)$ points from $\overline{x}_k(t)$ towards the (appropriately weighted) centroid of the $\overline{x}_j(t)$ for which $j \rightsquigarrow k$ at time $t$. According to [43, 44, 29], if the time-varying communication graph $G(t)$ is piecewise continuous in time and uniformly connected, then the states of all the agents exponentially converge to a common consensus value $\overline{x}_\infty$; moreover, if $G(t)$ is balanced for all $t$, then $\overline{x}_\infty = \frac{1}{N} \sum_{k=1}^N \overline{x}_k(0)$ (in other words, $\overline{x}_\infty$ is the centroid of the initial positions). This implies the following convergence property for (28),(29); the notation $\mathrm{IAM_g}$ generalizes the definition (12) of the IAM to the case where the initial points are not on $\mathcal{G}$.

**Proposition 8:** *Consider a piecewise continuous and uniformly connected bounded time-varying $\delta$-digraph $G(t)$ and a CC Lie group $\mathcal{G}$ satisfying Assumption 1. Further assume that the initial estimators $\overline{x}_k(0)$ are randomly chosen in $\mathcal{E}$. Then the only stable point of algorithm (28),(29) is synchronization at $\overline{y}_\infty$; if $G(t)$ is balanced, $\overline{y}_\infty = \mathrm{IAM_g}\{\overline{x}_k(0) | k = 1...N\}$.*

Proof: The convergence of (28) towards $\overline{x}_k = \overline{x}_\infty \; \forall k$ is ensured by the result of [44]; the property $\overline{x}_\infty = \frac{1}{N} \sum_{k=1}^N \overline{x}_k(0)$ for balanced graphs is easy to check (see [29]). As a consequence, the asymptotic form of (28),(29) is a set of $N$ independent systems

$$\begin{aligned} \overline{x}_k &= \overline{x}_\infty && (30) \\ \dot{\overline{y}}_k &= \gamma_S \operatorname{Proj}_{T\overline{\mathcal{G}},k}(\overline{x}_\infty) && (31) \end{aligned}$$

where $\overline{x}_\infty$ is a constant. According to [73], the $\omega$-limit sets of the original system (28),(29) correspond to the chain recurrent sets of the asymptotic system (30),(31). The first equation is trivial. Since (31) is a gradient ascent algorithm for the linear cost function $f(y_k) = \overline{y}_k^T \overline{x}_\infty$ and smooth (as the gradient of a smooth function along the smooth manifold $\mathcal{G}$), according to [74] its chain recurrent set is equal to its critical points and the stability analysis is reduced to the critical points of (31). Since $\mathcal{G}$ satisfies Assumption 1, all maxima of $f(y_k)$ are global maxima on $\mathcal{G}$, so the only stable points for the gradient ascent algorithm are the points belonging to the projection of $\overline{x}_\infty$ on $\mathcal{G}$. To conclude it remains to show that the projection of $\overline{x}_\infty$ on $\mathcal{G}$ reduces to a single point $\overline{y}_\infty$ with probability 1.

Since (28) and the associated initial conditions are invariant with respect to translations (and rotations) in $\mathcal{E}$, for every particular graph sequence, $\overline{x}_\infty$ has an equal probability to take any value in $\mathcal{E}$. It must be shown that the set $O$ of points in $\mathcal{E}$ whose projection onto $\mathcal{G}$ is not uniquely defined has measure zero in $\mathcal{E}$, such that $\overline{x}_\infty \notin O$ with probability 1. Select any point $x^a \in O$ and denote by $\mathcal{G}^a \subset \mathcal{G}$ its projection on $\mathcal{G}$. Choosing any $x^{*a} \in \mathcal{G}^a$, the set of points $\Omega^a = \{x^a + \sigma x^{*a} | \sigma > 0\}$, has a unique projection on $\mathcal{G}$, namely $x^{*a}$; hence, to the singular point $x^a$ corresponds an infinite set $\Omega^a$ of regular points. Now consider another point $x^b \in O$ and an associated $x^{*b} \in \mathcal{G}^b$. The sets $\Omega^b = \{x^b + \rho x^{*b} | \rho > 0\}$ and $\Omega^a$ do not intersect. Indeed,

- if $x^{*b} \neq x^{*a}$, then the points of $\Omega^b$ and $\Omega^a$ are projected onto different points of $\mathcal{G}$ and so cannot be equal.
- if $x^{*b} = x^{*a}$, it holds that $x^a = x^b + (\rho - \sigma)x^{*a}$. Supposing without loss of generality that $\rho > \sigma$, this leads to $x^a \in \Omega^b$ which contradicts $x^a \in O$.

Thus, to every point of $O$ corresponds an independent infinite set of points in $\mathcal{E}\backslash O$ such that $O$ has measure 0 in $\mathcal{E}$.

△

### 3.4.2 Anti-consensus algorithm

Using the estimator variables $\overline{x}_k$, one can also design algorithms that converge to anti-consensus configurations with limited communication. In analogy with the previous strategy, the position $\overline{y}_k$ of each agent evolves according to a gradient algorithm in order to maximize its distance to $\overline{x}_k(t)$. When $\overline{x}_k(t)$ asymptotically converges to $\overline{C}_e(t)$, this becomes equivalent to the gradient anti-consensus algorithm (23). For $\overline{C}_e(t)$ to be the consensus value of the local estimator variables $\overline{x}_k(t)$, it is necessary to require that $\overline{x}_k(0) = \overline{y}_k(0)$ and that $G(t)$ is balanced. Then the following algorithm is designed, where vector representations are used throughout; $\beta$ is still a positive scalar, but $\gamma_B$ is negative.

$$\dot{\overline{x}}_k = \beta \sum_{j=1}^{N} a_{jk}(\overline{x}_j - \overline{x}_k) + \dot{\overline{y}}_k \tag{32}$$

$$\dot{\overline{y}}_k = \gamma_B \, D_{k,\overline{\mathcal{G}}}(\overline{y}_k^T \overline{x}_k) = \gamma_B \, \mathrm{Proj}_{T\overline{\mathcal{G}},k}(\overline{x}_k)\,, \tag{33}$$

Note that the variables $\overline{x}_k$ and $\overline{y}_k$ are fully coupled; this essential feature of the balancing algorithm must be retained in the form of *implicit* discrete-time update equations in order to ensure convergence in a discrete-time version of this system (see [47] for details).

The following proposition characterizes the convergence properties of (32),(33).

**Proposition 9:** *Assume that $G(t)$ is a piecewise continuous, uniformly connected and balanced bounded time-varying $\delta$-digraph and that $\mathcal{G}$ satisfies Assumption 1. Then, algorithm (32),(33) with initial conditions $\overline{x}_k(0) = \overline{y}_k(0)$ $\forall k$ always converges towards a set of equilibrium configurations; the only stable sets are anti-consensus configurations for $G_c$.*

Proof: Since the interconnection graph is balanced,

$$\sum_{k=1}^{N} \dot{\overline{x}}_k = \beta \sum_{j=1}^{N} \left( \sum_{k=1}^{N} a_{jk} \right) \overline{x}_j - \beta \sum_{k=1}^{N} \left( \sum_{j=1}^{N} a_{jk} \right) \overline{x}_k) + \sum_{k=1}^{N} \dot{\overline{y}}_k(t) = \sum_{k=1}^{N} \dot{\overline{y}}_k(t)$$

and integrating both sides with $\overline{x}_k(0) = \overline{y}_k(0)$ one gets

$$\frac{1}{N}\sum_{k=1}^{N}\overline{x}_k(t) = \frac{1}{N}\sum_{k=1}^{N}\overline{y}_k(t) = \overline{C}_e(t)\ .$$

The Lyapunov function

$$W(t) = \frac{1}{2}\|\overline{x}\|^2$$

is never increasing along the solutions of (32),(33). Indeed, denoting by $(\overline{x})_j$ the vector containing $(\overline{x}_k)_j$ for $k = 1, 2...N$ and $j$ fixed in $\{1...m\}$, and using the Laplacian $L^{(i)}(t)$ of the varying graph associated to the $a_{jk}$, one has

$$\begin{aligned}\dot{W}(t) &= \sum_{k=1}^{N}\overline{x}_k(t)^T\dot{\overline{x}}_k(t) \\ &= \sum_{k=1}^{N}\overline{x}_k(t)^T\dot{\overline{y}}_k(t) - \beta\sum_{j=1}^{n}(\overline{x})_j^T L^{(i)}(\overline{x})_j \\ &= \frac{1}{\gamma_B}\sum_{k=1}^{N}\|\dot{\overline{y}}_k(t)\|^2 - \beta\sum_{j=1}^{n}(\overline{x})_j(t)^T L^{(i)}(t)(\overline{x})_j(t)\ ,\end{aligned}$$

where the last equality is obtained using the fact that

$$\overline{x}_k(t)^T\mathrm{Proj}_{T\overline{\mathcal{G}},k}(\overline{x}_k(t)) = \Big(\mathrm{Proj}_{T\overline{\mathcal{G}},k}(\overline{x}_k(t))\Big)^T\ \mathrm{Proj}_{T\overline{\mathcal{G}},k}(\overline{x}_k(t))\ .$$

The term containing the Laplacian is non-positive because $L(t)$ is positive semi-definite for balanced graphs (see [75]); as $\gamma_B < 0$, this yields $\dot{W} \leq 0$. This implies that $\overline{x}$ is uniformly bounded; as a consequence, $\dot{\overline{y}}_k$ in (33) is uniformly bounded as well. By combining the latter two observations with the fact that the Laplacian $L^{(i)}$ is bounded, it is observed from (32) that $\overline{x}$ is Lipschitz continuous in $t$ with some associated constant $r_1$.

Since $W(t)$ is bounded above by its initial value $W(0) = \frac{N}{2}\|\overline{y}_1\|^2$ and below by 0, it holds that

$$\frac{1}{|\gamma_B|}\sum_{k=1}^{N}\int_0^{+\infty}\|\dot{\overline{y}}_k(t)\|^2 \leq -\int_0^{+\infty}\dot{W}(t)dt \leq \frac{N}{2}\|\overline{y}_1\|^2$$

which means that each $\dot{\overline{y}}_k(t)$ is a function in $L_2(0,+\infty)$. From (33), $\dot{\overline{y}}_k$ is Lipschitz in $y_k$ (as the gradient of a smooth function along the smooth manifold $\mathcal{M}$) and linear in $x_k$. Adding that $\dot{\overline{y}}_k$ is uniformly bounded, one easily verifies that $\dot{\overline{y}}_k$ is Lipschitz in $t$. Then $\dot{\overline{y}}_k(t)$ is a uniformly continuous function in $L_2(0,+\infty)$ and $\dot{\overline{y}}_k \to 0\ \forall k$ from Barbalat's Lemma.

Equation (32) is equivalent to a consensus algorithm in $\mathcal{E}$ with an additive perturbation $\dot{\overline{y}}(t) \in L_2(0,+\infty)$; the addition of this perturbation does not destroy the asymptotic convergence of the exponentially stable consensus algorithm. As a consequence, each $\overline{x}_k$ is driven to some common consensus value $\overline{x}_\infty$, which must be equal to $\lim_{t\to+\infty}\overline{C}_e(t)$ according to the first observation of the proof. The asymptotic dynamics for $\overline{y}_k$ are thus

$$\dot{\overline{y}}_k(t) = \gamma_B\,\mathrm{Proj}_{T\overline{\mathcal{G}},k}(\overline{C}_e(t)) \tag{34}$$

which is exactly equal to the gradient descent algorithm (23) with $\gamma_B = 2\alpha N$. Since $\dot{\overline{y}}_k \to 0$, the solutions must asymptotically converge to a set of equilibria of (34) and it follows from the corresponding considerations about the gradient algorithms that the anti-consensus configurations for $G_c$ are the only stable sets.

△

### 3.4.3 About the communication of estimator variables

In order to implement these synchronization and anti-consensus algorithms, interconnected agents must communicate the values of their estimator variable $\overline{x}_k$. However, the estimators $\overline{x}_k$ evolve in $\mathcal{E}$ while the original system lives on $\mathcal{G}$; the relative position of agents on $\mathcal{G}$ is a meaningful measurement, but nothing ensures that a similar thing can be done in the a priori artificial space $\mathcal{E}$. It is important to note that the variables $\overline{x}_k$ may not just be a set of abstract scalars for each agent $k$: since $\overline{x}_k$ interacts with the geometric $\overline{y}_k$, it must be a geometric quantity too. Hence, the way of meaningfully communicating the estimators must be addressed.

A general solution to this problem is to assume a common (and thus external) reference frame for $\mathcal{E}$ - with a fixed embedding $\overline{\mathcal{G}}$ of $\mathcal{G}$. In this case, the swarm unfortunately loses its full autonomy; however, the external frame is just used for "translation" purposes and does not interfer with the dynamics of the system.

For CC Lie groups, it is possible to reformulate the algorithms such that they work completely autonomously if interconnected agents measure their relative positions. Indeed, the geometric relation between $\overline{y}_k$ and $\overline{x}_k$ in (29) and (33) occurs through the scalar product $\overline{y}_k^T\overline{x}_k$ for vector representations, or equivalently the form trace$(\overline{y}_k^T\overline{x}_k)$ for the matrix representations which are preferred in the remainder of this section. Furthermore, using group operations, the projection of the gradient on the tangent space at $\overline{y}_k$ can be computed as

$$\mathrm{Proj}_{T\overline{\mathcal{G}},k}(\overline{x}_k) = \overline{y}_k \mathrm{Proj}_{\mathfrak{g}}(\overline{y}_k^T\overline{x}_k)$$

where $\mathfrak{g}$ is the Lie algebra (or tangent space at the origin). Since the $\overline{y}_k$ are orthogonal matrices, the change of variables $\overline{z}_k = \overline{y}_k^T\overline{x}_k = \overline{y}_k^{-1}\overline{x}_k \Leftrightarrow \overline{x}_k = \overline{y}_k\overline{z}_k$ (all variables in matrix form) is well-defined. With this new variable $\overline{z}_k$, equations (29) and (33) can be written with shape entities, yielding

$$\overline{y}_k^T\dot{\overline{y}}_k = \gamma\,\mathrm{Proj}_{\mathfrak{g}}(\overline{z}_k) \tag{35}$$

where $\gamma$ is replaced by $\gamma_S$ or $\gamma_B$ for synchronization or balancing respectively, and now $\overline{z}_k$ can be considered as a matrix in the Lie algebra, without any link to a physical reference.

However, by doing this one must ensure that a correct link subsists between $\overline{x}_k$ and $\overline{x}_j$ in equations (28) and (32). This can be done by introducing the relative positions $\overline{y}_k^T\overline{y}_j$ such that $\overline{y}_k^T(\overline{x}_j - \overline{x_k}) = (\overline{y}_k^T\overline{y}_j)\overline{z}_j - \overline{z}_k$. Moreover, a change of variable from $\overline{x}_k$ to $\overline{z}_k$ must be operated on the left hand side of these equations. This leads to an additional term on the right hand side, such that (28) and (32) are respectively replaced by

$$\dot{\overline{z}}_k = (\overline{y}_k^T\dot{\overline{y}}_k)^T\overline{z}_k + \beta\sum_{j=1}^{N} a_{jk}((\overline{y}_k^T\overline{y}_j)\,\overline{z}_j - \overline{z}_k) \tag{36}$$

or

$$\dot{\overline{z}}_k = (\overline{y}_k^T\dot{\overline{y}}_k)^T\overline{z}_k + \beta\sum_{j=1}^{N} a_{jk}((\overline{y}_k^T\overline{y}_j)\,\overline{z}_j - \overline{z}_k) + (\overline{y}_k^T\dot{\overline{y}}_k)\ . \tag{37}$$

Note that the derivative of $\overline{y}_k$, as defined in (35), appears in (36) and (37).

### 3.4.4 Example

Algorithm (28),(29) is easily expressed on the particular manifold $SO(n)$. Introducing the auxiliary $n \times n$-matrices $X_k$ and denoting their vectorized forms by $\overline{x}_k$, equation (28) may be transcribed verbatim. Exploiting the previous discussions, the gradient of (29) is easily expressed by

$\mathrm{Proj}_{T\overline{\mathcal{G}},k}(X_k)$, which leads to

$$\dot{Q}_k = \frac{\gamma_S}{2} Q_k \left( Q_k^T X_k - X_k^T Q_k \right) \tag{38}$$

on $SO(n)$. The balancing algorithm is directly obtained in the same way. The reformulation in shape variables leads to the synchronization algorithm

$$\dot{Z}_k = (Q_k^T \dot{Q}_k)^T Z_k + \beta \sum_{j=1}^{N} a_{jk} \left( (Q_k^T Q_j) Z_j - Z_k \right) \tag{39}$$

$$Q_k^T \dot{Q}_k = \frac{\gamma_S}{2} \left( Z_k - Z_k^T \right) \tag{40}$$

or the balancing algorithm

$$\dot{Z}_k = (Q_k^T \dot{Q}_k)^T Z_k - \beta \sum_{j=1}^{N} a_{jk} \left( (Q_k^T Q_j) Z_j - Z_k \right) + (Q_k^T \dot{Q}_k) \tag{41}$$

$$Q_k^T \dot{Q}_k = \frac{\gamma_B}{2} \left( Z_k - Z_k^T \right) \tag{42}$$

where $Z_k$, $(Q_k^T Q_j) = (Q_j^T Q_k)^T$ and $(Q_k^T \dot{Q}_k)$ are the actual variables used by agent $k$.

Note that the exchange of the entire matrices $Z_k$ may be largely sub-optimal since an $n \times n$ special orthogonal matrix $Q_k$ can be retrieved with reduced information - typically, it is well known that any $d$-dimensional compact manifold can be embedded in a $2d$-dimensional Euclidean space.

## 3.5 Extension to homogeneous manifolds

The special properties of Lie groups are but scarcely used in the algorithms derived so far. In fact, the basis for all the previous developments is just the perfect symmetry of the manifold, i.e. all the points of the manifold must be geometrically strictly equivalent. This symmetry does not require the manifold to be a Lie group. Perfectly symmetric objects like spheres or Grassmann manifolds are not Lie groups, but quotients of Lie groups; these are called *homogeneous manifolds.*

In the previous developments and algorithms, the CC Lie group $\mathcal{G}$ may be replaced everywhere by a CC homogeneous manifold $\mathcal{M}$. All the discussions and results hold for $\mathcal{M}$, except for the two following points.

**Unitary embedding** It is a well-known fact of differential geometry that any smooth manifold of dimension $m$ can be smoothly embedded in $\mathbb{R}^{2m}$ (Whitney's embedding theorem). It seems logical that any proper embedding should respect the symmetry properties of $\mathcal{M}$. Furthermore, given the high symmetry of CC homogeneous manifolds, it seems plausible that it should be possible to embed them such that $\|\overline{y}\| = cst$ for any $y \in \mathcal{M}$; indeed, if some points $\overline{y}_k$ had different norms in $\mathcal{E}$, they could be considered different on that basis. In the absence of further knowledge on this issue, another blanket assumption is formulated, which the CC homogeneous manifolds should satisfy in order to apply the formalism of the present work.

**Assumption 2:** *$\mathcal{M}$ is smoothly embedded in a compact subset of a Euclidean space $\mathcal{E}$ of dimension $m$ such that*

*a) the action of the transitive symmetry group of $\mathcal{M}$ can be expressed as an action of a subgroup of the transitive symmetry group of $\mathcal{E}$;*

*b) the Euclidean norm $\|\overline{y}_k\|$ is constant (or without loss of generality, equals 1) over $\overline{\mathcal{M}}$.*

**Communication of estimator variables** The only point where Lie group properties are extensively used in the preceding developments is in the reformulation of the estimator variable algorithms in terms of local coordinates. Indeed, the group multiplication is a vital tool for the change of variables used in these developments. As a consequence, on CC homogeneous manifolds it seems difficult, if not impossible, to derive a reformulation with estimator variables expressed with respect to the agent positions instead of a common reference frame for $\mathcal{E}$. At the current state of research, the existence and use - for communication purposes only - of such a common frame for $\mathcal{E}$ must thus be assumed in order to implement the algorithms of section 3.4 on CC homogeneous manifolds.

### 3.5.1 Example

The Grassmann manifolds (see section 2.2) are interesting examples of CC homogeneous manifolds. They can be useful to compute means of, reach agreement on, or distribute, a set of subspaces of a larger vector space. The Grassmann manifolds satisfy all the assumptions needed to apply the results of this work and algorithms are easily written using the projector representation[6]. The Frobenius norm $\|\overline{y}_k\|$ of a $p$-rank projector is $\sqrt{p}$. The projection of a matrix $M \in \mathbb{S}_n^+$ onto the tangent space to $Grass(p,n)$ according to the projector representation is given in [71] as $\Pi_k M \Pi_{\perp k} + \Pi_{\perp k} M \Pi_k$.

**Induced arithmetic mean** The IAM on $Grass(p,n)$ can be characterized analytically in the following way.

**Proposition 10:** *The centroid $\overline{C}_e$ of $N$ points in the projector representation of $Grass(p,n)$ is generally a symmetric positive semi-definite matrix of rank $\geq p$. The induced arithmetic mean then contains the eigenspaces $C$ corresponding to $p$ largest eigenvalues of $\overline{C}_e$; it reduces to a single element of $Grass(p,n)$ when the $p$-largest and $(p+1)$-largest eigenvalues of $\overline{C}_e$ are different.*

Proof: The proof of this statement follows the same lines as for $SO(n)$ and is provided somewhat later.

△

In fact, if $\Pi_{\mathcal{B}} = BB^T$ where $B \in St(p,n)$ contains $p$ orthonormal vectors spanning the subspace corresponding to $\Pi_{\mathcal{B}}$, the cost function in (12) becomes

$$f(\Pi_{\mathcal{B}}) = \text{trace}(\Pi_{\mathcal{B}}\overline{C}_e) = \text{trace}(B^T\overline{C}_eB) = p\frac{\|B^T\overline{C}_eB\|}{\|B^TB\|} \tag{43}$$

where the last expression is equal to the generalized Rayleigh quotient for the computation of the dominant $p$-dimensional eigenspace of $\overline{C}_e$. The geometric computation of eigenspaces from the cost function (43) is extensively covered in the work of P.-A. Absil [60, 52]. Furthermore, it is a well-known fact of linear algebra that the $p$ largest eigenvalues (the others being 0) of $\Pi_{\mathcal{B}}\Pi_k$ are the squared cosines of the principal angles $\phi_k^i$, $i = 1...p$ between the subspaces $\mathcal{B}$ and $\mathcal{Y}_k$. From this, one can easily show that the point $C$ is the subspace that minimizes the sum of the squared sines of the principal angles between the set of subspaces $\mathcal{Y}_k$ and a centroid candidate subspace $\mathcal{B}$:

$$C = \underset{\mathcal{B}}{\text{argmin}} \sum_{k=1}^{N}\sum_{i=1}^{p} \sin^2(\phi_k^i)\ . \tag{44}$$

[6] The basis representation of $Grass(p,n)$ with elements $Y_k \in St(p,n)$ respects the two items of Assumption 2, but it is not an embedding of $Grass(p,n)$. The use of the projector representation, embedding $Grass(p,n)$ in $\mathbb{S}_n^+$, is necessary to properly define the induced arithmetic mean. Indeed, when simply summing up arbitrary elements $Y_k \in St(p,n)$ representing $\mathcal{Y}_k$ to get $\overline{C}_e$ and projecting back onto $Grass(p,n)$, the result depends on the particular $Y_k$ chosen.

This provides a geometrical meaning for the induced arithmetic mean of subspaces. Moreover, it illustrates its equivalence with the Karcher mean for infinitesimal angles; indeed, the cost function for the Karcher mean would be equal to the sum of the squared principal angles in the case of $Grass(p,n)$. This observation is also made in a section of [60] on the computation of a "centroid of subspaces" approximation, and in [35, 36] where the measurement of the distance between two subspaces of $Grass(p,n)$ as the Frobenius norm of the difference between the associated projectors is called the *chordal distance*.

**Consensus cost function and balancing** On $Grass(p,n)$, (19) can be rewritten as

$$P = \frac{1}{2N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} a_{jk} \left( \sum_{i=1}^{p} \cos^2(\phi_{jk}^i) \right)$$

where $\phi_{jk}^i$ is the $i^{\text{th}}$ principal angle between the subspaces $j$ and $k$. Similar remarks are made in [36, 35, 60].

The balanced states on $Grass(p,n)$ correspond to the situation where all eigenvalues of $\overline{C}_e$ are equal. Note that $\text{trace}(\overline{C}_e) = \frac{1}{N}\sum_k \text{trace}(\Pi_k) = p$, such that this requirement leads to $\overline{C}_e = \frac{p}{n} I_n$. This is not always possible with $N$ orthonormal projectors of rank $p$ [7]. As for $SO(n)$, simulations tend to indicate that it is possible when $N$ is large enough. However, it seems a bit more tricky to compute the minimal value of $N$ that is needed for a given $n$ and $p$.

**Gradient type algorithms** On $Grass(p,n)$, the gradient consensus algorithm of section 3.3 can be written as

$$\dot{\Pi}_k = 2\alpha \sum_j a_{jk} (\Pi_k \Pi_j \Pi_{\perp k} + \Pi_{\perp k} \Pi_j \Pi_k) \ . \tag{45}$$

In practice, it may be handier to use the basis representation with $Y_k \in St(p,n)$ instead of $\Pi_k$ in the algorithms; indeed, this allows to deal with $n \times p$ and $p \times p$ matrices instead of possibly much larger $n \times n$ matrices. This operation is possible because it is not necessary to embed a manifold in order to compute gradients on it. The calculation of $D_{k,Grass(p,n)}$ on a quotient manifold is explained in [60]; it is based on the projection of the gradient with respect to $Y_k$ onto the so-called *horizontal space*, which is achieved by $\Pi_{\perp k}$ in the present case. This leads to the algorithm

$$\dot{Y}_k = 4\alpha \Pi_{\perp k} \sum_j a_{jk} \Pi_j Y_k = 4\alpha \sum_j a_{jk} \left( Y_j M_{j^T k} - Y_k M_{j^T k}^T M_{j^T k} \right) \tag{46}$$

where the $p \times p$ matrices $M_{j^T k}$ are defined as $M_{j^T k} = Y_j^T Y_k$. Note that, though the gradient in [60] was computed in the non-compact Stiefel manifold, (46) keeps the matrices $Y_k$ on the compact Stiefel manifold (i.e. $Y_k(t)^T Y_k(t) = I_p \ \forall t$ if $Y_k(0)^T Y_k(0) = I_p$). The reader will also notice that the projector representation algorithm obtained from (46) and the identities (4), differs from (45) by a factor 2. This difference can be understood from the fact that components $(\Pi_k)_{ij}$ and $(\Pi_k)_{ji}$ of the symmetric matrix $\Pi_k$ are considered independent in the projector representation, while the basis representation captures their symmetry by construction. For theoretical purposes, the projector representation is a better choice, as for the following proofs.

**Proposition 11:** *The Grassmann manifold $Grass(p,n)$ satisfies Assumption 1.*

Proof (+ Prop.10): Consider a linear function $f(\Pi) = \text{trace}(\Pi^T B)$ as in Assumption 1, where $B \in \mathbb{S}_n^+$ and $\Pi$ represents $\mathcal{Y} \in Grass(p,n)$. From $D_{k,\mathbb{R}^{n\times n}}(f) = B$ one gets $D_{k,Grass(p,n)}(f) =$

[7] In fact, the global minimum of $P$ corresponding to balanced states is a solution of the relaxed convex problem: minimize $\sum_{j=1}^{n}(\lambda_j)^2$ over $\lambda_j \geq 0$ under the constraint $\sum_{j=1}^{n}(\lambda_j) = p$. In order to be a feasible solution for the unrelaxed problem, the $\lambda_j$ should further be the eigenvalues of the matrix $\overline{C}_e$ built from $N$ orthonormal projectors of rank $p$.

$\Pi_k B \Pi_{\perp k} + \Pi_{\perp k} B \Pi_k$. Since the ranges of the first and second terms in this last expression are at most $\mathcal{Y}$ and its orthogonal complement respectively, they must both be equal to zero at a critical point $\mathcal{Y}^*$. This implies that $\mathcal{Y}^*$ must be an invariant subspace of $B$. In an appropriate basis $(e_1 ... e_n)$, one can then write $\Pi = \text{diag}(1, ...1, 0, ...0)$ and $B = \text{diag}(\mu_1, ...\mu_p, \mu_{p+1} ...\mu_n)$. If there exist $d \leq p$ and $l > p$ such that $\mu_l > \mu_d$, then any variation of $\Pi$ rotating $e_d$ towards $e_l$ strictly increases $f(\Pi)$. Therefore, at a local maximum of $f(\Pi)$, the $p$-dimensional space corresponding to $\Pi$ must be an eigenspace of $B$ corresponding to $p$ largest eigenvalues of $B$. This implies that the value of $f(\Pi)$ at any local maximum equals the sum of $p$ largest eigenvalues of $B$ and Assumption 1 is satisfied. Moreover, replacing $B$ by $\overline{C}_e$, Proposition 10 is proven.

△

**Algorithms with estimator variables** First of all, it must be recalled that the issue of communicating the estimator variables in a meaningful way without using a common reference frame in $\mathcal{E}$ has not been solved. Therefore, no reformulation of the algorithms in terms of local variables is available. Assuming the existence of a common reference frame in $\mathcal{E}$ for translation purposes, equation (28) may be transcribed verbatim and (29) becomes

$$\dot{Y}_k = 2\gamma_S (I_n - Y_k Y_k^T) X_k Y_k \quad \text{or} \quad \dot{\Pi}_k = \gamma_S \Pi_k X_k \Pi_{\perp k} + \gamma_S \Pi_{\perp k} X_k \Pi_k \; . \tag{47}$$

The balancing algorithm is directly obtained analogously. Note that here, the full projector representation absolutely must be used in (28) and (32), such that using $n \times n$ matrices becomes unavoidable. Unfortunately, this can be a quite heavy overload with respect to an intrinsic formulation when $p \ll n$.

# 4 Synchronization of autonomous Lie group solids

Section 3 is concerned with the definition of consensus configurations and the design of individual control laws for simple integrator models that drive the swarm towards these configurations. The present Section 4 focuses on the development of coordination algorithms for *mechanical models*, namely the so-called Lie group solids.

In order to focus on the mechanical part, the objective is somewhat simplified to the achievement of *synchronization* (i.e. the agents are all at the same (maybe moving) point on $\mathcal{G}$) and occasionally *arbitrary coordinated motion* (i.e. the relative positions of the agents on $\mathcal{G}$ remain constant, but at arbitrary values, throughout their motion). Mathematically, these 2 situations imply $y_k^{-1} y_j \to e \; \forall j, k$ for synchronization and $y_k^{-1} y_j \to cst \; \forall j, k$ for arbitrary coordinated motion.

The basic strategy to design the algorithms in Section 3 is to formulate consensus as an optimization task. This basic strategy still serves for the design of control algorithms for mechanical models in the present section. The difference lies in the way the optimization problem must be solved. For the consensus tracking point of view (Section 4.3), a consensus algorithm is used at a task planning level to define desired trajectories which are tracked by the individual mechanical agents. A second, more inherently mechanical point of view (Section 4.4) applies the "energy shaping" method: the cost function of the optimization problem is used as an artificial potential and properly designed artificial dissipation asymptotically stabilizes the synchronized state as a minimum of the artificial potential, like in ([5, 7, 10, 11]).

The non-triviality of the mechanical model of the Lie group solid (2),(3) makes the extension with respect to the previous simple integrator models an insightful task. Therefore the problem deserves some discussion before rushing to the control algorithms.

## 4.1 Problem setting

It is worth recalling the basic ingredients before proceeding to the actual discussion.

The task is thus to synchronize (the motion of) $N$ identical Lie group solids, frequently called agents in the following. The mechanical model of a Lie group solid imposes the following equations of motion [70]

$$\begin{aligned} \dot{y}_k &= y_k \xi_k && (48) \\ J \dot{\xi}_k^\vee &= [J\xi_k^\vee ,\; \xi_k^\vee] + \tau_k^\vee && (49) \end{aligned}$$

where the relevant quantities are defined in Section 2.2.2; just recall that $\tau_k^\vee$ is the generalized control torque to be designed. For the special case of a rigid body in 3-dimensional space, $\mathcal{G} = SO(3)$ and the equations of motion become (Euler equations)

$$\begin{aligned} \dot{Q}_k &= Q_k \omega_k^\wedge && (50) \\ J \dot{\omega}_k &= (J\omega_k) \times \omega_k + \tau_k && (51) \end{aligned}$$

where $Q_k$ is a rotation matrix characterizing the orientation of body $k$, $\omega_k$ is the angular velocity vector and $\tau_k$ is the control torque, both expressed in body frame. Without loss of generality, it is assumed in the example sections about $SO(3)$ that $J = \mathrm{diag}(J_1,\, J_2,\, J_3)$ with $J_1 > J_2 > J_3$.

A *synchronized state* is characterized by $y_k = y_j \; \forall k, j$, whatever the absolute position and velocity may be. However, as noted in [5, 64], only the absolute position can be factored out of the state space. Indeed, the dynamics (48),(49) are invariant with respect to a fixed translation (group multiplication) of all the agents by $g \in \mathcal{G}$, but not with respect to any arbitrary synchronized motions since the absolute velocity $\xi_k$ explicitly appears in the dynamics (49). This can be seen both as bad news or as good news: on one hand, it means that the absolute velocity is likely to appear in the control laws, but on the other hand it means that this absolute velocity can somehow be retrieved, without requiring an external reference, by an inertial sensor (in fact, the generalization of the gyroscope in $SO(3)$ to general Lie group solids).

When designing the control inputs, the agents are assumed to be fully actuated, but only have access to information from a subset of the other agents, which are their neighbors according to the interconnection graph. For simplicity, a unit weight $a_{jk} = 1$ is attributed to all the communication links. It is assumed that agent $k$ gets the following information from each of its neighbors $j$:

- its relative position $y_k^{-1} y_j$ ;
- its relative velocity, $\mathrm{Ad}_{y_k^{-1} y_j} \xi_j - \xi_k$ ;
- possibly a set of scalar auxiliary variables $x_k$.

In addition, agent $k$ may have to measure $\xi_k$, its own velocity with respect to an inertial frame. However, it never knows its absolute position $y_k$. This is consistent with the symmetries of the dynamics.

Note that in the present part of the report, a matrix of the adjoint representation of the group element $y_k$ is simply designed by $y_k$ itself; also remember that $y_k^{-1} = y_k^T$.

## 4.2 Coordination strategy

The strategy for Lie group solid coordination strongly builds on the results of Section 3. It turns out that the obtained developments are not entirely new: other authors have previously worked out similar strategies, mainly on $SO(3)$, although without reaching the final results of the present work.

The two approaches considered in the present work to generalize the results of Section 3 to mechanical models - *consensus tracking* and *energy shaping* - are well-known in the control literature. The viewpoint adopted in the present section highlights the similarities of the consensus and energy shaping approaches to Lie group solid synchronization. Therefore the opportunity is taken to refer to related work in more detail when relevant. Of course, this oversimplified review does not capture all the many specificities of both approaches. Most notably, the important work on mechanical symmetries and reduction described among others in [5, 10, 11, 64] is not taken into account. Notations and formulations have also been adapted, for the sake of simplicity and at the cost of generality. The interested reader is kindly asked to consult the corresponding literature.

### 4.2.1 From integrators to mechanical systems

Consider the function $P_L$ defined by (16) as the basis for the optimization strategy. With unit weights and the matrix representation, it becomes $P_L = \frac{1}{2N^2} \sum_{k=1}^{N} \sum_{j \rightsquigarrow k} \mathrm{trace}\left( y_j^T y_k \right)$. The corresponding form for $SO(3)$ (simply replacing $y_k$ by $Q_k$) was itself already introduced in a mechanical/energy-shaping framework in [5, 50, 64, 7].

In the present context, the goal is synchronization of the agents. According to Proposition 2, synchronization is the only maximum of $P_L$ at least when the undirected graph $G$ associated to the agent interconnections is a complete graph $G_c$ or a tree. Therefore most global convergence results in this section assume that $G$ is a tree or $G_c$. However, for arbitrary graphs one can still get local results; in particular, the work of Moreau ([43, 44]) seems to indicate that the basin of attraction of the synchronized state contains all situations which are such that the agents eventually end up in a convex subset of $\mathcal{G}$ for $t$ larger than some $\bar{t}$.

In an *energy-shaping* framework, $P_L$ is used to build an artificial potential $\sigma P_L$, $\sigma < 0$, whose global minimum corresponds to the synchronized state ([5, 7, 10, 11]). This leads to the following basic control torque:

$$\tau_k^{(S)} = -\sigma\, D_{k,\overline{\mathcal{G}}}(P_L)\ . \tag{52}$$

Unsurprisingly, since the torque is equal to minus the gradient of the potential $\sigma P_L$, this control input is (up to a positive multiplication constant) equal to the one for the gradient synchronization algorithm for simple integrators in Section 3.3; as such, it only uses information on relative positions of the agents in a fixed, undirected interconnection graph $G$ ([7, 11, 5, 10]). The difference is that now, the input does not directly govern the motion of the agent on $\mathcal{G}$ but acts as a torque in a second-order mechanical model.

For $SO(3)$, using the Energy-Momentum method [65], it is proven in [7, 10, 11] that the synchronized state where all the rigid bodies rotate together about their aligned short axis, is a stable equilibrium of (50),(51) with the control (52). A weaker form of this result is proven in [5] using Semidirect Product Reduction [67] and the Energy-Casimir method [66]: the synchronized state where the agents are non-rotating is stable. However, the Hamiltonian of the system (consisting of the kinetic energy and the artificial potential, such that (52) is accounted for by the artificial potential) is conserved, such that some form of dissipation must be added to obtain asymptotic stability. This is left as an open question in [7] and solved with the help of an external reference in [10, 11]. In [5], it is suggested to use external dissipation (drag) to obtain asymptotic stability. Dissipation without any external reference is considered in the present report. The design of relative dissipation in mechanical systems in general is also addressed in [68] with a different approach. Due to the derivation of the control torques from the potential $\sigma P_L$, the energy shaping approach is restricted to fixed undirected interconnection graphs $G$ and sensitive to the local minima of $\sigma P_L$ imposed by $G$.

This limitation can be removed in the *consensus tracking* framework. In this approach, a first layer of algorithms defines desired kinematic trajectories to maximize $P_L$, assuming that the generalized angular velocity $\xi_k$ is a direct control input. This is exactly what is done in Section 3, so this first layer actually uses one of the consensus algorithms described earlier in the present report. When $G$ is fixed and undirected, a gradient-type consensus algorithm can be used and when $G$ is directed and/or time-varying, estimator variables must be introduced to ensure convergence. Note that, with the estimator variables, by using a consensus algorithm in Euclidean space, the problem of local maxima is circumvented. This first layer thus provides desired evolutions for $\xi_k = y_k^T \dot{y}_k$, denoted by $\xi_k^{(d)}$. The equations defining $\xi_k^{(d)}$ are directly obained by substitution of $y_k^T \dot{y}_k$ by $\xi_k^{(d)}$ in (22) and (35),(36), leading to

$$\xi_k^{(d)} = \tfrac{\alpha_k}{2N^2}\, \mathrm{Proj}_{\mathfrak{g}} \left( \overline{y}_k^T \sum_{j \rightsquigarrow k} (\overline{y}_j - \overline{y}_k) \right) \tag{53}$$

and

$$\begin{aligned} \dot{\overline{z}}_k &= \xi_k^T \overline{z}_k + \alpha_k^{(1)} \sum_{j \rightsquigarrow k} ((\overline{y}_k^{(-1)} \overline{y}_j)\, \overline{z}_j - \overline{z}_k) & (54) \\ \xi_k^{(d)} &= \tfrac{\alpha_k^{(2)}}{2}\, \mathrm{Proj}_{\mathfrak{g}}(\overline{z}_k) & (55) \end{aligned}$$

respectively where $\alpha_k, \alpha_k^{(1)}, \alpha_k^{(2)} > 0$.

At a second layer, tracking algorithms are used by the mechanical agents in order to individually track their desired trajectories as defined by the consensus algorithm. As is shown in the following Section 4.3, by properly coupling the two layers of algorithms, synchronization of the swarm can be achieved.

### 4.2.2 Example

In fact, the specialization of the mathematical expressions appearing in the preceding discussion to $SO(3)$ was already done in Section 3. Nevertheless, they are repeated hereunder for the sake of easier referencing.

The torque resulting from the gradient of $P_L$ on $SO(n)$ is given in (26). Introducing unit weights and noting that the control torque is related to the vectorized form of the angular velocity $\left(Q_k^T \frac{d}{dt} Q_k\right)^\vee$, equation (52) becomes

$$\tau_k^{(S)} = \tfrac{-\sigma}{2N^2} \left( \sum_{j \rightsquigarrow k} (Q_k^T Q_j - Q_j^T Q_k) \right)^\vee . \tag{56}$$

The right hand side defining the desired trajectories for the gradient consensus algorithm is strictly similar:

$$(\omega_k^{(d)})^\wedge = \tfrac{\alpha_k}{2N^2} \sum_{j \rightsquigarrow k} \left( Q_k^T Q_j - Q_j^T Q_k \right) . \tag{57}$$

If estimator variables are used for consensus, the desired trajectories are defined by (39),(40) which are repeated below.

$$\begin{aligned} \dot{Z}_k &= (Q_k^T \dot{Q}_k)^T Z_k + \beta \sum_{j \rightsquigarrow k} \left( (Q_k^T Q_j) Z_j - Z_k \right) & (58) \\ (\omega_k^{(d)})^\wedge &= \frac{\gamma_S}{2} \left( Z_k - Z_k^T \right) & (59) \end{aligned}$$

## 4.3 Consensus tracking

The consensus algorithms directly assign a desired velocity to each agent $k$ at each time instant $t$. A second step is required to obtain explicit expressions for the control torques $\tau_k$. This is discussed in the following.

### 4.3.1 Basic considerations

The knowledge of an individual's own absolute velocity $\xi_k$ is unavoidable for the present consensus tracking control laws. With this information, it is rather obvious to make the agents individually track a desired velocity field $\xi_k^{(d)}$ defined by a consensus algorithm. In fact, the simplest control strategy based on (54) defines the desired *position* $y_k^{(d)}$ to be the projection of $x_k$ on $\mathcal{G}$; equation (55) is replaced by a dynamical tracking algorithm on $\mathcal{G}$. The projection process may present discontinuities; for example, the projection from $\mathbb{R}^{3\times 3}$ to $SO(3)$ presents a discontinuity when $x_k$ is singular. As a consequence, $y_k^{(d)}$ is not necessarily continuous. However, it is unimportant to track the transient trajectory: the only objective is to synchronize the Lie group solids towards the final consensus value of $x_k$. In this setting, it might even seem useless to move the rigid bodies before the auxiliary variables approach a consensus situation. Moving the rigid bodies into the desired attitude after the agents have reached consensus on $x_k$ would just require a global position *stabilization* controller for each agent. Algorithms for tracking or stabilization on $SO(3)$ ("attitude tracking") or more general Lie groups may be found among others in [23, 49, 50, 51, 42, 76]. Tracking approaches to attitude coordination can also be found in [6, 26, 27], though the strong presence of a common external reference is necessary for their results. The following explicitly considers some control torques for a consensus tracking synchronization strategy for which easy convergence proofs can be provided. It is likely that these rather naive controllers are not the most efficient in practice. For a better performance, other tracking controllers can be selected according to the needs of particular applications.

Both (53) and (55) impose zero velocity when synchronization is achieved. To be more general, a common constant velocity $\xi_0$ (in body frame) is imposed to the synchronized swarm ($\xi_0$ may e.g. result from a consensus algorithm in the vector space $\mathfrak{g}$). For algorithms based on the gradient consensus approach, $\xi_0$ is simply added to the desired velocity for synchronization to get

$$\xi_k^{(d)} = \tfrac{\alpha_k}{2N^2} \operatorname{Proj}_{\mathfrak{g}} \left( \overline{y}_k^T \sum_{j \rightsquigarrow k} (\overline{y}_j - \overline{y}_k) \right) + \xi_0 \ . \tag{60}$$

For algorithms based on the estimator variable consensus algorithm, the desired position for $y_k$ becomes $x_k \text{exp}(t\,\xi_0)$ and the desired velocity becomes

$$\xi_k^{(d)} = \xi_0 + \tfrac{\alpha_k^{(2)}}{2} \ \operatorname{Proj}_{\mathfrak{g}}(z_k \text{exp}(t\,\xi_0)) \ . \tag{61}$$

### 4.3.2 The computed torque method

The so-called *computed torque* method is a rather radical way to deal with mechanical systems: the inherent dynamics (i.e. the dynamics that drive the agents when the input $\tau_k = 0$) are simply countered by an equivalent control torque such that any desired evolution can be imposed to $\xi_k$.

In order to drive $\xi_k$ towards $\xi_k^{(d)}$, an exponential evolution

$$J \tfrac{d}{dt}(\xi_k^\vee - (\xi_k^{(d)})^\vee) = -\beta_k(\xi_k^\vee - (\xi_k^{(d)})^\vee) \tag{62}$$

is imposed, where $\beta_k > 0$ and $\frac{d}{dt}\xi_k^{(d)}$ is deduced from (60) or (61) and (54). Given (49), this leads to the control input

$$\tau_k^\vee = J \tfrac{d}{dt}(\xi_k^{(d)})^\vee - \beta_k(\xi_k^\vee - (\xi_k^{(d)})^\vee) - [J\xi_k^\vee, \xi_k^\vee] \tag{63}$$

which achieves (62) by using available information: $\xi_k^{(d)}$ and $\frac{d}{dt}\xi_k^{(d)}$ involve relative positions and relative velocities of interconnected agents (and possibly auxiliary variables if the estimator consensus strategy is chosen); the knowledge of oneself's absolute velocity $\xi_k$ is unavoidable for the consensus tracking control laws. The following convergence properties hold.

**Proposition 12:** *Consider a swarm of $N$ agents exchanging their relative positions and velocities according to a fixed undirected graph $G$ and applying the control torque (63) where $\omega_k^{(d)}$ is defined by (60). If $\alpha_k = \alpha \forall k$ or $\xi_0 = 0$, then the swarm is asymptotically driven towards an equilibrium set where $\xi_k = \xi_0\ \forall k$ and the $y_k$ are at a critical point of $P_L$. The only stable equilibria are the maxima of $P_L$.*

Proof: Consider the candidate Lyapunov function

$$W = \frac{1}{4N^2} \sum_{k=1}^{N} \sum_{j \rightsquigarrow k} \|y_j - y_k\|^2 + \sum_{k=1}^{N} \tfrac{\delta_k}{2}(\xi_k^\vee - (\xi_k^{(d)})^\vee)^T J (\xi_k^\vee - (\xi_k^{(d)})^\vee)$$

where $\delta_k$ are positive constants to be fixed. Obviously, $W \geq 0$. The first term of (64) is equal to $-P_L$ up to an additive constant. Note that

$$\begin{aligned} \tfrac{d}{dt} P_L &= \sum_k \text{trace}\left( (D_{k,\overline{\mathcal{G}}}(P_L))^T \tfrac{d}{dt} y_k \right) = \sum_k \tfrac{1}{\alpha_k} \text{trace}\left( (y_k(\xi_k^{(d)} - \xi_0))^T (y_k \xi_k) \right) \\ &= \sum_k \tfrac{1}{\alpha_k} \text{trace}\left( (\xi_k^{(d)} - \xi_0)^T \xi_k \right) = \sum_k \tfrac{q}{\alpha_k} ((\xi_k^{(d)})^\vee - \xi_0^\vee)^T \xi_k^\vee \end{aligned}$$

where $q$ is a constant positive factor relating the scalar product of the vectorized forms $\xi_k^\vee$ to the trace of products of the matrix forms $\xi_k$; for $SO(3)$, the factor $q = 2$. The time derivative of the second term in (64) is directly obtained from (62). A few basic calculations lead to

$$\begin{aligned}\frac{d}{dt}W &= \sum_k \tfrac{-q\gamma}{\alpha_k(1-\gamma)}\|\xi_k^\vee - (\xi_k^{(d)})^\vee\|^2 - \tfrac{q}{\alpha_k}(\|\xi_k^\vee - \xi_0^\vee\|^2 + \|(\xi_k^{(d)})^\vee - \xi_0^\vee\|^2) \\ &\qquad + \tfrac{q}{\alpha_k}(\xi_0^\vee)^T((\xi_k^{(d)})^\vee - \xi_0^\vee)\end{aligned} \tag{64}$$

where $(1-\gamma) = \frac{q}{\alpha_k\delta_k\beta_k}$ is chosen in $(0,1)$ through a proper choice of $\delta_k$. One easily checks from (60) that $\sum_k((\xi_k^{(d)})^\vee - \xi_0^\vee) = 0$, such that the last term vanishes when $\xi_0 = 0$ or all the $\alpha_k$ are equal. As a consequence, all the terms in (64) are non-positive. Hence $W$ decreases from its initial value towards a set of points where $\frac{d}{dt}W = 0$. The latter requires $\xi_k = \xi_0 = \xi_k^{(d)}$ $\forall k$. According to the definition (60) of $\xi_k^{(d)}$, the requirement $\xi_k^{(d)} = \xi_0$ characterizes the critical points of $P_L$. Writing $W$ as a function of state variables $(Q_k, \varepsilon_k = \xi_k - \xi_k^{(d)})$ instead of $(Q_k, \xi_k)$, it is obvious that the set of equilibria for $W$ are the critical points of $P_L$ with $\varepsilon_k = 0$ $\forall k$ and the stable points must be maxima of $P_L$ (and minima of $\varepsilon_k^T\varepsilon_k$, which is trivial).

△

**Proposition 13:** *Consider a swarm of $N$ agents exchanging their relative positions and velocities according to a piecewise continuous bounded $\delta$-digraph $G$ and applying the control torque (63) where $\omega_k^{(d)}$ is defined by (61), in combination with the consensus algorithm (54). If $G$ is uniformly connected and the initial values of $z_k$ are randomly chosen in $\mathcal{E}$, then the only stable limit set for $t \to +\infty$ is synchronization of the positions $y_k$ with $\xi_k = \xi_0$ $\forall k$.*

Proof: Similarly to the proof of Proposition 12, consider the candidate Lyapunov function

$$W = -\sum_{k=1}^{N}\sum_{j\rightsquigarrow k}\text{trace}(z_k\text{exp}(\xi_0 t)) + \sum_{k=1}^{N}\tfrac{\delta_k}{2}(\xi_k^\vee - (\xi_k^{(d)})^\vee)^T J(\xi_k^\vee - (\xi_k^{(d)})^\vee)$$

The time derivative of the second term is directly obtained from (62). For the first term, remembering the fact that the estimator variables $x_k = y_k z_k$ evolve independently of the $y_k$, write

$$\begin{aligned}\tfrac{d}{dt}\big(\text{trace}(y_k^T x_k \text{exp}(\xi_0 t)\big) &= \text{trace}\,\big(y_k^T \dot{x}_k \text{exp}(\xi_0 t) + \xi_k^T y_k^T x_k \text{exp}(\xi_0 t) + y_k^T x_k \xi_0 \text{exp}(\xi_0 t)\big) \\ &= \text{trace}\,\big(y_k^T \dot{x}_k \text{exp}(\xi_0 t) \\ &\qquad + \text{trace}\,\big(\xi_k^T (z_k \text{exp}(\xi_0 t))\big) - \text{trace}\,\big(\xi_0^T (z_k \text{exp}(\xi_0 t))\big)\end{aligned} \tag{65}$$

where the facts that $\xi_0$ and $\text{exp}(\xi_0 t)$ commute and that $\xi_0$ must be skew-symmetric have been used. The first term in (65), involving $\dot{x}_k$, will be considered later. The second term in (65) is a scalar product (expressed in matrix form) between $\xi_k \in \mathfrak{g}$ and $z_k\text{exp}(\xi_0 t)$. Therefore, replacing $z_k\text{exp}(\xi_0 t)$ by its projection on $\mathfrak{g}$ changes nothing and according to (61) this second term can be written as

$$\tfrac{2q}{\alpha_k^{(2)}}(\xi_k^\vee)^T((\xi_k^{(d)})^\vee - \xi_0)\ .$$

Similarly, the third term in (65) can be written as

$$\tfrac{2q}{\alpha_k^{(2)}}(\xi_0^\vee)^T((\xi_k^{(d)})^\vee - \xi_0)\ .$$

After a few calculations, analogous to those for Proposition 12, the time derivative of $W$ can be written as

$$\begin{aligned}\frac{d}{dt}W &= \sum_k \tfrac{-q\gamma}{\alpha_k^{(2)}(1-\gamma)}\|\xi_k^\vee - (\xi_k^{(d)})^\vee\|^2 - \tfrac{q}{\alpha_k^{(2)}}(\|\xi_k^\vee - \xi_0^\vee\|^2 + \|(\xi_k^{(d)})^\vee - \xi_0^\vee\|^2) \\ &\qquad - \text{trace}\,\big(y_k^T \dot{x}_k \text{exp}(\xi_0 t)\big)\end{aligned} \tag{66}$$

where, as previously, $(1-\gamma) = \frac{q}{\alpha_k^{(2)}\delta_k\beta_k}$ is chosen in $(0,1)$ through a proper choice of $\delta_k$. From this one observes that $\dot{W}$ is negative definite, except for the last term. However, this last expression asymptotically goes to zero since $\dot{x}_k \to 0$ when the $x_k$ synchronize. The latter fact is ensured by the fact that the conditions for consensus in Euclidean space, as required in [44], are satisfied. Moreover, considering convexity properties for the evolution of the estimator variables ([44]), it is easy to show that $W$ is bounded above as well as below once the initial states have been chosen. As a consequence, if the negative definite terms stay smaller than some $\phi < 0$, after some time they will dominate the behavior of $\dot{W}$, such that $W$ continuously decreases at a rate close to $\phi$. Since this is impossible for $W_{max} > W(t) > W_{min}\ \forall t$, the negative definite terms must asymptotically go to 0. The remaining conclusions are similar to the proof of Proposition 12.

△

The control torque (63) includes a term that exactly cancels the free Lie group solid dynamics. This is characteristic of the computed torque method and requires perfect knowledge of the parameters of the mechanical systems. In order to ensure this fact in real applications, a model adaptation algorithm may be required. Alternatively, a different tracking algorithm can be used, as the following one for instance.

### 4.3.3 The high gain method

A close alternative to the computed torque method is the high gain method, where the free Lie group solid dynamics are dominated instead of cancelled. Indeed, if $\beta_k$ is large enough, it is not necessary to include the last term of (63) cancelling the free Lie group solid dynamics. Thus, the control torque reduces to

$$\tau_k^\vee = J\tfrac{d}{dt}(\xi_k^{(d)})^\vee - \beta_k(\xi_k^\vee - (\xi_k^{(d)})^\vee)\ . \tag{67}$$

The high gain method allows to impose motions $\xi_0$ satisfying $[J\xi_0^\vee,\, \xi_0^\vee] = 0$ only. Thus the imposed final motion is always an equilibrium of the free Lie group solid dynamics. This is coherent with the fact that the free Lie group solid dynamics is not cancelled. Moreover, it is probably the most useful case in practice, as maintaining other motions with whatever algorithm would require persistent control torques.

The following Propositions state the exact convergence results. Note that the bounds were not derived with much care and hence may be quite loose; the main purpose is to show that high gain can be applied. $J_{max}$ denotes the largest eigenvalue of $J$.

**Proposition 14:** *Consider a swarm of $N$ agents exchanging their relative positions and velocities according to a fixed undirected graph $G$ and applying the control torque (67) where $\omega_k^{(d)}$ is defined by (60). Assume that either $\xi_0 = 0$ or $\alpha_k = \alpha \forall k$ and $\xi_0$ satisfies $[J\xi_0,\, \xi_0] = 0$. Moreover, set*

$$\beta_k > 2J_{max}(\|\xi_0^\vee\| + \alpha_k\sqrt{q}\|\overline{y}\|d_k/N^2) \tag{68}$$

*for all agents $k$, where $d_k$ is the number of neighbors of $k$, $q$ is a constant positive factor relating the scalar product of the vectorized forms $\xi_k^\vee$ to the trace of products of the matrix forms $\xi_k$, and $\|\overline{y}\|$ denotes the norm of the elements of $\mathcal{G}$ in the matrix representation. Then the swarm is asymptotically driven towards an equilibrium set where $\xi_k = \xi_0\ \forall k$ and the $y_k$ are at a critical point of $P_L$. The only stable equilibria are the maxima of $P_L$.*

Proof: Consider the same Lyapunov candidate as for Proposition 12. The time derivative of the first part does not change. The time derivative of the second part includes an additional term of the form $\delta_k(\xi_k^\vee - (\xi_k^{(d)})^\vee)^T[J\xi_k^\vee,\, \xi_k^\vee]$. As a consequence, (64) becomes

$$\begin{aligned}\tfrac{d}{dt}W &= \sum_k \tfrac{-q\gamma}{\alpha_k(1-\gamma)}\|\xi_k^\vee - (\xi_k^{(d)})^\vee\|^2 - \tfrac{q}{\alpha_k}(\|\xi_k^\vee - \xi_0^\vee\|^2 + \|(\xi_k^{(d)})^\vee - \xi_0^\vee\|^2) \\ &\qquad + \tfrac{q}{\alpha_k\beta_k(1-\gamma)}(\xi_k^\vee - (\xi_k^{(d)})^\vee)^T[J\xi_k^\vee,\, \xi_k^\vee] \qquad (69)\end{aligned}$$

where the fact that the last term in (64) is equal to zero has already been taken into account. The goal is to bound the indefinite term by the definite terms of $\dot{W}$. First rewrite

$$[J\xi_k^\vee,\, \xi_k^\vee] = [J(\xi_k^\vee - \xi_0^\vee),\, (\xi_k^\vee - \xi_0^\vee)] + [J(\xi_k^\vee - \xi_0^\vee),\, \xi_0^\vee] + [J\xi_0^\vee,\, (\xi_k^\vee - \xi_0^\vee)]$$

where the fact that $[J\xi_0^\vee,\, \xi_0^\vee] = 0$ has been taken into account. Further, the first term in the above expansion is orthogonal to $(\xi_k^\vee - \xi_0^\vee)$ so the scalar product of this term with $(\xi_k^\vee - (\xi_k^{(d)})^\vee)$ is equal to the scalar product with $(\xi_0^\vee - (\xi_k^{(d)})^\vee)$.

Now the worst-case evolution for $W$ is

$$\begin{aligned}\tfrac{d}{dt}W &\leq \sum_k \tfrac{-q\gamma}{\alpha_k(1-\gamma)}\|\xi_k^\vee - (\xi_k^{(d)})^\vee\|^2 - \tfrac{q}{\alpha_k}(\|\xi_k^\vee - \xi_0^\vee\|^2 + \|(\xi_k^{(d)})^\vee - \xi_0^\vee\|^2) \\ &\qquad + \tfrac{qJ_{max}}{\alpha_k\beta_k(1-\gamma)}\left(2\|\xi_k^\vee - (\xi_k^{(d)})^\vee\|\,\|\xi_k^\vee - \xi_0^\vee\|\,\|\xi_0\| + \|\xi_0^\vee - (\xi_k^{(d)})^\vee\|\,\|\xi_k^\vee - \xi_0^\vee\|^2\right)\,. \qquad (70)\end{aligned}$$

The first term in the last line can be rewritten thanks to

$$2\|\xi_k^\vee - (\xi_k^{(d)})^\vee\|\,\|\xi_k^\vee - \xi_0^\vee\| = -\left((\|\xi_k^\vee - \xi_0^\vee\| - \|\xi_k^\vee - (\xi_k^{(d)})^\vee\|)^2 - \|\xi_k^\vee - \xi_0^\vee\|^2 - \|\xi_k^\vee - (\xi_k^{(d)})^\vee\|^2\right)\,.$$

For the second term, note that according to (60), $\|\xi_k^{(d)} - \xi_0\| \leq \frac{\alpha_k d_k}{N^2}\|\overline{y}\|$. This finally leads to

$$\begin{aligned}\tfrac{d}{dt}W &\leq \sum_k \tfrac{-q}{\alpha_k}\|(\xi_k^{(d)})^\vee - \xi_0^\vee\|^2 - \tfrac{qJ_{max}\|\xi_0^\vee\|}{\alpha_k\beta_k(1-\gamma)}\left(\|\xi_k^\vee - \xi_0^\vee\| - \|\xi_k^\vee - (\xi_k^{(d)})^\vee\|\right)^2 \\ &\qquad - \tfrac{q}{\alpha_k}\left(1 - \tfrac{J_{max}(\|\xi_0^\vee\| + \alpha_k\sqrt{q}\|\overline{y}\|d_k/N^2)}{\beta_k(1-\gamma)}\right)\|\xi_k^\vee - \xi_0^\vee\|^2 \\ &\qquad - \tfrac{q\gamma}{\alpha_k(1-\gamma)}\left(1 - \tfrac{J_{max}\|\xi_0^\vee\|}{\beta_k\gamma}\right)\|\xi_k^\vee - (\xi_k^{(d)})^\vee\|^2\end{aligned}$$

where the first line is non-positive definite and the second and third lines are non-positive definite if $\beta_k$ is large enough. Choosing $\gamma = 1/2$, the condition for the coefficient of the third line is automatically satisfied when the condition for the second line is satisfied. The latter provides condition (68). The rest of the proof is the same as for Proposition 12.

△

**Proposition 15:** *Consider a swarm of $N$ agents exchanging their relative positions and velocities according to a piecewise continuous bounded $\delta$-digraph $G$ and applying the control torque (67) where $\omega_k^{(d)}$ is defined by (61), in combination with the consensus algorithm (54). Moreover, assume that $G$ is uniformly connected, the initial values of $z_k$ are randomly chosen in $\mathcal{E}$ and $\xi_0$ satisfies $[J\xi_0,\, \xi_0] = 0$. Moreover, set*

$$\beta_k > J_{max}(2\|\xi_0^\vee\| + \alpha_k^{(2)}\sqrt{q}\|z_k\|) \qquad (71)$$

*for all agents $k$, where $q$ is a constant positive factor relating the scalar product of the vectorized forms $\xi_k^\vee$ to the trace of products of the matrix forms $\xi_k$, and $\|z_k\|$ denotes the norm of $z_k$ in the matrix representation. Then the only stable limit set for $t \to +\infty$ is synchronization of the positions $y_k$ with $\xi_k = \xi_0$ $\forall k$.*

Proof: The proof is simply a mix of those for Propositions 13 and 14 and is therefore not reproduced in full length. The difference in the bound comes from the fact that (61) is used instead of (60) and hence $\|\xi_k^{(d)} - \xi_0\|$ is now bounded by $\alpha_k^{(2)}\|z_k\|/2$.

△

The bound for Proposition 15 involves the values of $\|z_k\|$. As has been hinted in the proof of Proposition 13, those are ensured to be bounded because a general property of the consensus algorithms in Euclidean space ([44]) is that, for whatever interconnection graph and at any time, the values of $x_k(t)$ lie in the convex hull of the initial values $\{x_1(0), x_2(0), ...x_N(0)\}$.

Before proceeding to the energy shaping approach, the flexibility of the consensus tracking approach must be highlighted once more. It has already been mentioned that the 2 tracking methods considered in the present work, along with explicit convergence proofs, are but examples selected for the simplicity of their theoretical analysis and may be replaced by others for better performance or robustness. Even in the present algorithms, the parameters $\alpha_k, \beta_k, ...$ can be chosen according to specific criteria. In fact, there is even no fundamental need for them to be constant. Therefore, their values may be adapted according to other constraints of the system - as long as the hypotheses for convergence are still satisfied (for instance, all $\alpha_k$ equal if $\xi_0 \neq 0$ in Proposition 12). The algorithms may have to be slightly modified - for instance a term in $\frac{d}{dt}\alpha_k$ would appear in the expression of $\frac{d}{dt}\xi_k^{(d)}$.

When $\xi_0 = 0$, the conditions (68) and (71) suggest to maintain comparable values of $\frac{\beta_k}{\alpha_k d_k}$ for the different agents. It can be observed that the choice $\alpha_k = \frac{\alpha}{d_k}$ is a natural choice often encountered in the consensus literature (where $\beta_k$ is absent since only kinematics are considered).

### 4.3.4 Example

It is straightforward to write the particular forms of the control torques on $SO(3)$ using previously given information. Propositions 12 and 13 require no further particularization either.

The constant velocity $\xi_0$ simply becomes a constant angular velocity $\omega_0$ in body frame. Hence at the synchronized equilibrium, the synchronized rigid bodies would rotate with a fixed angular velocity (in body frame and hence also in inertial space). Though this has not been further explored, it may be that for $G \neq G_c$, a rotation of every agent at velocity $\omega_0$ in its own frame turns out to be incompatible with the requirement to stay at a particular critical point of $P_L$ different from synchronization. It could thus be that the introduction of $\omega_0 \neq 0$ favors synchronization of the rigid bodies, in place of other local minima of $P_L$. This remark probably also applies to general Lie group solids.

Regarding Propositions 14 and 15, the requirement $[J\xi_0^\vee, \xi_0^\vee] = 0$ becomes $(J\omega_0) \times \omega_0 = 0$ which implies that $\omega_0$ must be aligned with a principal axis of the rigid body. $J_{max}$ is simply the largest principal moment of inertia, $J_1$. The matrix norm $\|\overline{y}\|$ is equal to $\sqrt{3}$.

## 4.4 Energy shaping

When the interconnection graph is fixed and undirected, an elegant alternative to the consensus tracking approach is the energy shaping approach, where $P_L$ is used in an artificial potential. As previously mentioned, the energy shaping approach to synchronization on $SO(3)$ has been initialized in the work of [5, 7, 10, 11], where the basic control torque (52) was first proposed. The goal in the present section is to obtain *asymptotic* synchronization with control torques satisfying the assumptions about available information.

The energy shaping approach leads to simpler and arguably more robust control laws. Moreover, the basic control torque (52) can be computed without requiring any information about velocities; those will only appear through the dissipation, which is designed in the present work. As a consequence, (52) imposes no restrictions on the final motion of the synchronized agents; the set of possible motions will only be reduced according to the symmetries of the inherent dynamics and the dissipation term. At the end of this section, a control torque that can be implemented *without absolute velocity measurements* is presented and local convergence towards synchronization is proven for $SO(3)$. The most fundamental (current) limitation of energy shaping is that the interconnection graph $G$ must be fixed and undirected (and connected).

The control torque derived from the potential $\sigma P_L$ ensures that the maxima of $P_L$ are stable equilibria of the Lie group solid dynamics. However, without any further control input, they are not asymptotically stable. Indeed, the system obtained from (48),(49) with the control $\tau_k = \tau_k^{(S)}$ defined in (52) $\forall k$ is a conservative mechanical system with the energy function

$$H = T_k + \sigma P_L \ .$$

In order to obtain asymptotic stability and hence actually drive the agents towards consensus configurations, a dissipative torque $\tau_k^{(D)}$ must be added, such that $\tau_k = \tau_k^{(S)} + \tau_k^{(D)}$. The energy function $H$ then evolves as

$$\tfrac{d}{dt} H = \sum_k (\xi_k^\vee)^T (\tau_k^{(D)})^\vee \ . \tag{72}$$

### 4.4.1 Dissipation in inertial space

Simply introducing dissipation on the motion of each individual agent is admissible if each agent measures its own velocity. This was already suggested in [5]. This leads to the control torque

$$\tau_k^{(D)} = -b_k \xi_k \ , \quad b_k > 0 \tag{73}$$

for which the following can be proven.

**Proposition 16:** *Consider a swarm of $N$ agents exchanging their relative positions according to a fixed undirected graph $G$. The control torque defined as the sum of (52) and (73), drives the swarm towards an equilibrium set where $\xi_k = 0$ $\forall k$ and the $y_k$ are at a critical point of $P_L$. The only stable equilibria are the maxima of $P_L$.*

Proof: The proof follows automatically by noting that the controlled swarm is equivalent to a mechanical system with dissipation. Combining (72) and (73), the energy $H$ decreases in time according to

$$\dot{H} = -\sum_k b_k \|\xi_k^\vee\|^2$$

such that the velocities $\xi_k^\vee$ converge to 0. The invariant set with $\xi_k = 0$ $\forall k$ consists of the critical points of the potential, hence of $P_L$. The stable equilibria are the minima of the potential, hence the maxima of $P_L$.

△

Proposition 16 uses a simplified control torque with respect to Propositions 12 and 14, for an equivalent result with $\xi_0 = 0$. In particular, no exchange of relative velocities is needed, the free Lie group solid dynamics are not counteracted and there is no condition on the strength of the control torques.

However, the introduction of dissipation in inertial space stabilizes the agents at rest. Though adaptations similar to those in Section 4.3 can be imagined to stabilize a synchronized motion at velocity $\xi_0$, the reference to absolute velocities will always imply an explicit control on the motion of the agents. In some practical situations, it may be unnecessary to explicitly control the motion of the synchronized agents. The only (or primary) objective could be to synchronize the agents, whatever their synchronized motion, and unnecessary resources would be spent to additionally achieve a specific motion. More importantly, it may be desirable to use a control law that synchronizes the agents without affecting their synchronized motion, in order to couple it with another algorithm that would control the motion of the synchronizd swarm. This can be achieved by replacing the dissipation with respect to inertial space by "inter-agent dissipation" as is considered in the following.

### 4.4.2 Dissipation in shape space

A more elegant way to introduce dissipation in the context of relative motions is through the *relative* velocities. This replaces the measurement of absolute velocities $\xi_k$ by relative velocities of interconnected agents, such that the agents can implement their control torque without any absolute information about their own state. The resulting control torque is

$$\tau_k = \tau_k^{(S)} - b \sum_{j \rightsquigarrow k} (\xi_k - \mathrm{Ad}_{y_k^{-1} y_j} \xi_j) \tag{74}$$

with $b > 0$. A fundamental property of (74) is the fact that the torques $\mathrm{Ad}_{y_k}\tau_k$ in inertial space sum to zero, such that the total generalized angular momentum of the swarm $\sum_k \mathrm{Ad}_{y_k} J \xi_k^\vee$ is conserved.

The control (74) always asymptotically leads to *velocity synchronization*, i.e. $\mathrm{Ad}_{y_k}\xi_k = \mathrm{Ad}_{y_j}\xi_j$ $\forall k, j$. This means that asymptotically, the relative positions $y_k^T y_j$ in the swarm are constant. As a consequence, the control torques $\tau_k$ are all constant as well. However, the control law does not necessarily lead to attitude synchonization. A simple counterexample with just 2 rigid bodies (i.e. Lie group solids on $SO(3)$, see Section 4.4.3) indicates that asymptotic orientation synchronization - even locally - requires at least an additional assumption on the relative strength of the artificial potential with respect to the kinetic energy. The fact that such a condition is also sufficient to ensure local convergence towards synchronization is only proven for the specific case of $SO(3)$ in Section 4.4.3, though it is believed that similar proofs can be made for any specific compact Lie group. In contrast , the following property is easy to prove in general.

**Proposition 17:** *Consider a swarm of $N$ agents exchanging their relative positions according to a connected, fixed undirected graph $G$. The control torque (74) where $\tau_k^{(S)}$ is defined in (52), drives the swarm towards an invariant set under (52) with synchronized velocities $\mathrm{Ad}_{y_k}\xi_k$ (and hence fixed relative positions $y_k^T y_j$).*

Proof: As for Proposition 16, consider the evolution of the energy $H$. From (72) it follows that

$$\begin{aligned} \dot{H} &= -b \sum_k (\xi_k^\vee)^T \sum_{j \rightsquigarrow k} (\xi_k^\vee \mathrm{Ad}_{y_k^T y_j} \xi_j^\vee) \\ &= -b \sum_k \sum_{j \rightsquigarrow k} (\mathrm{Ad}_{y_k}\xi_k^\vee)^T (\mathrm{Ad}_{y_k}\xi_k^\vee - \mathrm{Ad}_{y_j}\xi_j^\vee) \\ &= -b \{\mathrm{Ad}\xi^\vee\} (L \otimes I_m) \{\mathrm{Ad}\xi^\vee\} \end{aligned} \tag{75}$$

where $\{\mathrm{Ad}\xi^\vee\}$ denotes the vector containing all the $\mathrm{Ad}_{y_k}\xi_k^\vee$, $L$ is the Laplacian of the undirected graph $G$ and $(L \otimes I_m)$ denotes the Kronecker product of $L$ with an identity matrix that has the size of $\xi_k^\vee$, such that the Laplacian $L$ operates componentwise on the $\mathrm{Ad}_{y_k}\xi_k^\vee$. According to the properties of the Laplacian of undirected graphs, (75) is strictly negative unless the velocities $\mathrm{Ad}_{y_k}\xi_k$ are all equal within each connected component of $G$. When $G$ is connected, this means that all the velocities must ultimately synchronize. This implies that $\tau_k^{(D)} = 0$ for all $k$ and hence the swarm must end up in a set satisfying (48),(49) with the conservative control (52).

△

Note that though velocity synchronization is achieved, the value taken by the common velocity is not imposed; in fact, it does not even have to be constant and will indeed in general vary with time. When the agents are all synchronized, the control torques $\tau_k$ defined by (74) all vanish; hence any synchronized free Lie group solid motion is an equilibrium trajectory for (48),(49) with control (74). There may however be many other combinations of relative positions and motion that are invariant under (48),(49) with a non-zero conservative control torque (52).

Also note that the sum appearing in $\tau_k^{(D)}$ of (74) need not consider the same interconnection graph $G$ as for $\tau_k^{(S)}$. It is just a natural assumption in the present context.

### 4.4.3 Example

The energy shaping control laws are readily particularized to $SO(3)$, recalling among others that $\mathrm{Ad}_{y_k}\xi_k^\vee$ becomes $Q_k\omega_k$. The main interest of the present example section is to go somewhat further regarding convergence properties of (74) for the particular case of $SO(3)$.

**Counterexample** First, the counterexample mentioned above, showing that (74) does not always lead, even locally, to position synchronization, is briefly developed. Consider 2 identical rigid bodies $A$ and $B$. Imagine them in a synchronized situation where they both have the same orientation and rotate around $e_1$ with angular velocity $\omega$, where $e_1, e_2, e_3$ denote the principal axes corresponding to $J_1, J_2, J_3$ respectively. This is an equilibrium situation for the 2 rigid bodies, with $\tau_A = \tau_B = 0$. Now consider that the real situation is slightly different: with respect to the synchronized situation, body $A$ is tilted by an angle $\phi$ around $e_2$ and body $B$ is tilted by $-\phi$ around the same axis; however, they still rotate around the axis corresponding to the initial orientation of the synchronized $e_1$ axes, which now makes an angle $\phi$ with their actual axes $e_{1\,A}$ and $e_{1\,B}$ (see Figure 2). This situation is not an equilibrium for the free rigid bodies. Indeed, denoting by $\omega$ the angular velocity in inertial space and by $\omega_A$ and $\omega_B$ the angular velocities in body frames,

$$(J\omega_A) \times \omega_A = \begin{pmatrix} J_1 \cos\phi \, \|\omega\| \\ 0 \\ J_3 \sin\phi \, \|\omega\| \end{pmatrix} \times \begin{pmatrix} \cos\phi \, \|\omega\| \\ 0 \\ \sin\phi \, \|\omega\| \end{pmatrix} = \|\omega\|^2 \tfrac{(J_1 - J_3)}{2} \begin{pmatrix} 0 \\ \sin(2\phi) \\ 0 \end{pmatrix}$$

and similarly $(J\omega_B) \times \omega_B = -\|\omega\|^2 \frac{(J_1-J_3)}{2} \sin(2\phi) e_2$. Since $J_1 > J_3$, this indicates that the free rigid body dynamics produce torques around $e_2$ that pull the bodies $A$ and $B$ even further away from the synchronized state. The control torques on $A$ and $B$ must still be added to this effect in order to obtain the total behavior.

Consider that the motion described in Figure 2 is indeed stabilized, such that $\tau_A^{(D)} = \tau_B^{(D)} = 0$ (since both bodies have the same angular velocity $\mathrm{Ad}_{Q_k}\omega_k = \omega$). The effect of $\tau_A^{(S)} = -\tau_B^{(S)}$ is to pull $A$ and $B$ back to the synchronized situation. After a few developments, one gets

$$\tau_A^\vee = -\tau_B^\vee = -\kappa \sin(2\phi) e_2$$

where $\kappa$ is some positive constant. Hence if $\|\omega\|^2 \frac{(J_1-J_3)}{2} = \kappa$, the control torques exactly cancel the torques imposed by the free rigid body dynamics and the motion of Figure 2 is an equilibrium. Local stability of this situation can be verified through simulations or linearization. Further developments determine the angle $\phi$ as a function of the initial configuration of the rigid bodies, making use of angular momentum conservation. The angle $\phi$ can be arbitrarily small, attracting solutions that are arbitrarily close to synchronization, such that the synchronized state is not locally asymptotically stable.

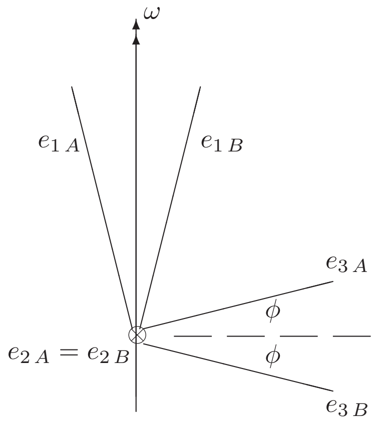

Figure 2: Non-synchronized situation for 2 rigid bodies from which control law (74) does not achieve synchronization if no bound is imposed on $\sigma$. The represented vectors all lie in the same plane, except $e_{2\,A} = e_{2\,B}$ which is perpendicular to the plane. This situation can be achieved arbitrarily close to the synchronized state, i.e. with $\phi$ arbitrarily small.

**Asymptotic synchronization result** Finally, as mentioned in Section 4.4.2, a local convergence result can be proven for (74) on $SO(3)$ when a bound is imposed on the strength of the potential with respect to the kinetic energy. The following proposition states this result.

**Proposition 18:** *Consider a swarm of $N$ agents exchanging their relative positions and velocities according to a connected, fixed undirected graph $G$. The control torque (74) where $\tau_k^{(S)}$ is defined in (52), drives the swarm towards an invariant set under (52) with synchronized angular velocities $Q_k\omega_k$ (and hence fixed relative orientations $Q_k^TQ_j$). Moreover, for every set of initial angular velocities, there exists a constant $\sigma^* < 0$ (actually depending on $N$, $J$, $G$ and the initial* kinetic energy $T(0)$ *only) such that when $|\sigma| > |\sigma^*|$, the orientations $Q_k$ of the agents locally asymptotically synchronize.*

Proof: The synchronization of the angular velocities $Q_k\omega_k$ is simply Proposition 17. For the synchronization of the orientations $Q_k$, the proof is in two steps. First, it is shown that given a neighborhood $W$ of synchronization, there exist a value $|\sigma_1|$ and a neighborhood $U$ of synchronization such that starting in $U$ imposes staying in $W$ for $|\sigma| > |\sigma_1|$. Then it is shown, using linearization, that there exists a value $|\sigma_2|$ such that for $|\sigma| > |\sigma_2|$, synchronization is a locally unique solution of (50),(51),(52) with identical angular velocities $Q_k\omega_k$. Choosing $W$ such that this linear/local result is valid inside $W$ then allows to conclude that solutions starting in $U$ must converge to synchronization for $|\sigma| > \max(|\sigma_1|, |\sigma_2|)$.

For the first part, assume that it is wanted to ensure that $\mathrm{trace}(I - Q_k^TQ_j) < \varepsilon\ \forall k, j$ and for all time. Obviously, this will be satisfied if

$$\tfrac{1}{2}\sum_k \sum_{j \rightsquigarrow k} \mathrm{trace}(I - Q_k^TQ_j) = 3E - N^2 P_L(t) < \varepsilon$$

where $E$ is the number of edges in the graph $G$ (the factor $\frac{1}{2}$ comes from the fact that each distance is counted twice in the sum). Since the energy $H$ is decreasing in time, $H(t) = T(t) + \sigma P_L(t) \leq T(0) + \sigma P_L(0)$ which implies that $\sigma(P_L(t) - P_L(0)) \leq T(0) - T(t) \leq T(0)$. Hence if $|\sigma| > |\sigma_1|$, then $P_L(0) - P_L(t) \leq T(0)/|\sigma_1|$ or

$$3E - N^2 P_L(t) \leq (3E - N^2 P_L(0)) + \tfrac{N^2}{|\sigma_1|} T(0)\ .$$

Choosing the initial neighborhood $U$ of synchronization such that $3E - N^2 P_L(0) < \varepsilon/2$ and $|\sigma_1| > \frac{2N^2}{\varepsilon} T(0)$, one obtains $3E - N^2 P_L < \varepsilon$ as desired.

The second part involves much more calculations, which will not all be detailed. A central element in the proof is the conservation of total angular momentum for the control law (74).

First denote the final common angular velocity by $Q_k \omega_k = \Omega \; \forall k$ and note that $\|\Omega\|^2$ is bounded by $T(0)$. Indeed, denoting by $M_T$ the (conserved) total angular momentum, it is obvious that $\|\Omega\| \leq \frac{M_T}{N J_3}$. Further, applying the inequality $(\sum_k x_k)^2 \leq N \sum_k x_k^2$ to $x_k = J\omega_k(0)$, one obtains $M_T^2 \leq 2NJ_1 T(0)$ such that finally

$$\|\Omega\|^2 \leq \tfrac{2J_1}{NJ_3^2} T(0) \; .$$

Next consider the first derivative of $\Omega = Q_k \omega_k$. One obtains

$$\dot{\Omega} = \tfrac{-\sigma}{N^2} Q_k J^{-1} \sum_{j \rightsquigarrow k} (Q_k^T Q_j - Q_j^T Q_k)^{\vee} + Q_k J^{-1} Q_k^T (Q_k J Q_k^T \Omega) \times \Omega \tag{76}$$

which must be equal $\forall k$. In the following, the values of (76) are compared for two agents $k$ and $j$. Remember that, staying in $W$, trace$(I - Q_k^T Q_j) < \varepsilon \; \forall k, j$ and denote by $\varepsilon_{kj}$ its actual value. With several calculations, one can show that

$$\|Q_k J^{-1} Q_k^T (Q_k J Q_k^T \Omega) \times \Omega - Q_j J^{-1} Q_j^T (Q_j J Q_j^T \Omega) \times \Omega\|^2 \leq \tfrac{16 J_1^2}{J_3^2} \|\Omega\|^4 \varepsilon_{kj}$$

which implies that the same bound must hold for the difference of the first parts of (76),

$$\tfrac{\sigma^2}{N^4} \|Q_k J^{-1} \sum_{l \rightsquigarrow k} (Q_k^T Q_l - Q_l^T Q_k)^{\vee} - Q_j J^{-1} \sum_{m \rightsquigarrow j} (Q_j^T Q_m - Q_m^T Q_j)^{\vee}\|^2 \; .$$

Summing this condition over all $k, j$, after several calculations one obtains

$$\frac{2\lambda_2^3}{J_1^2} (1 - \varepsilon) \; |\varepsilon_{kj}|_{max} \leq \frac{64 J_1^4 E}{J_3^6} \frac{N^2 T(0)^2}{\sigma^2} \; |\varepsilon_{kj}|_{max} + \mathcal{O}\Big( \; (|\varepsilon_{kj}|_{max})^{3/2} \Big)$$

where $E$ is the number of edges in $G$, $\lambda_2 > 0$ is the second-smallest eigenvalue of the Laplacian $L$ of $G$ and $|\varepsilon_{kj}|_{max}$ denotes the maximal value of $\varepsilon_{kj}$ among all pairs of connected agents. It is easy to choose $W$ (and thus $U$ and actually $\sigma_1$) such that the higher order terms represent, for instance, less than 10 percent of the right member. In that case the condition becomes

$$|\varepsilon_{kj}|_{max} \leq \frac{32 J_1^6 E}{0.9 J_3^6 \lambda_2^3 (1 - \varepsilon)} \frac{N^2 T(0)^2}{\sigma^2} \; |\varepsilon_{kj}|_{max} = \frac{\sigma_2^2}{\sigma^2} \; |\varepsilon_{kj}|_{max} \; . \tag{77}$$

Taking $\sigma^2 > \sigma_2^2$, (77) can only be satisfied if $|\varepsilon_{kj}|_{max} = 0$ which means that all the rigid bodies must be synchronized.

△

In the reduced state space $(TSO(3))^N / TSO(3)$ of relative orientations and relative angular velocities, the statement about orientation synchronization is equivalent to local asymptotic stability of the isolated equilibrium $Q_k^T Q_j = I_3$, $Q_k \omega_k - Q_j \omega_j = 0 \; \forall k, j$. The absolute angular velocity - which can not been factored out of the rigid body dynamics but is not part of the state space of relative orientations and velocities - is then considered as an external variable inducing time-varying dynamics. The bound $|\sigma| > |\sigma^*(T(0))|$ for synchronization is non-uniform with respect to the (initial) absolute angular velocity.

As previously mentioned, thanks to the absence of the absolute angular velocity in the control torques (74), the agents, once synchronized, may still freely rotate according to any free rigid body motion. The following Figure 3 illustrates this fact by showing a typical simulation result of control law (74) applied to a swarm of 5 rigid bodies with random initial orientations and angular

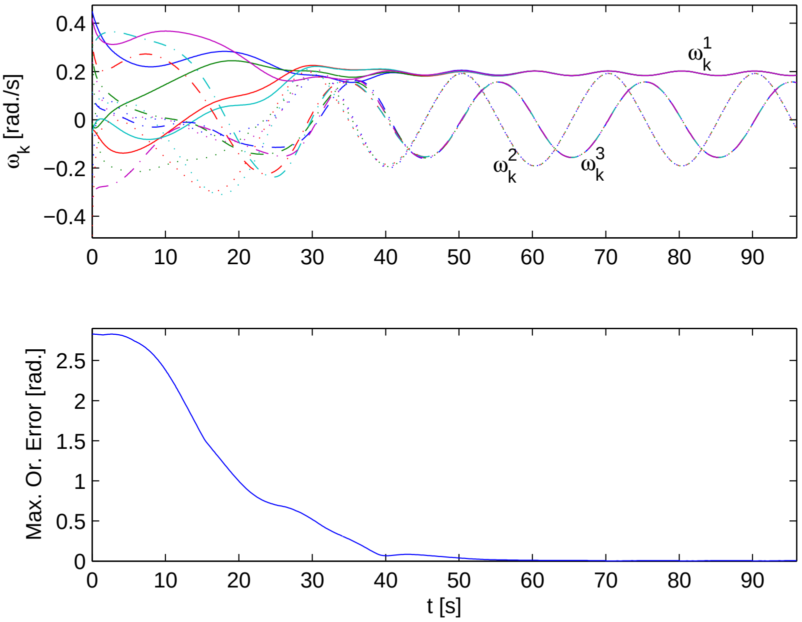

Figure 3: Angular velocities $\omega_k = (\omega_k^1\ \omega_k^2\ \omega_k^3)$ (top) and orientation error (bottom) of five agents applying control law (74). The orientation error is defined as the maximal Euler rotation angle between any couple of agents. The interconnection graph is a ring. Initial orientations and velocities are randomly chosen.

velocities. The interconnection graph is a ring (i.e. 1 is connected to 5 and 2, 2 is connected to 1 and 3,...). On the first graph, one observes the asymptotic synchronization of angular velocities and their ongoing periodic variation after synchronization is reached. The second graph displays the maximal orientation error - defined as the Euler rotation angle - between any two pairs of rigid bodies in the swarm. The convergence of this error towards zero indicates that the orientations also asymptotically synchronize.

# Finale

## Recap - Overview of the contributions revisited

The present work makes several main contributions.

The goal of the first part (Section 3) was to provide *extended definitions* of the mean position of $N$ points and of consensus configurations for $N$ points evolving on a connected compact Lie group, or more generally a connected compact homogeneous manifold. Simple integrator models are assumed for the agents in order to design control algorithms that drive the swarm towards these configurations under constrained communication interconnections and without the use of any external reference.

First, the induced arithmetic mean of $N$ points is defined on an embedded connected compact homogeneous manifold $\mathcal{M}$; though it differs from the traditional Karcher mean, this mean has a clear geometric meaning with the advantage of being easily computable - in fact, analytical solutions are presented for $SO(n)$ and $Grass(p,n)$.

Secondly, an extended definition of consensus is presented on these manifolds, which is directly linked to the induced arithmetic mean. In particular, the notion of balancing introduced in [13] for the circle is extended to CC homogeneous manifolds.

Thirdly, an optimization setting is developed for the consensus problem and related distributed consensus algorithms are designed for $N$ simple integrator agents moving on a CC homogeneous manifold. Gradient algorithms for fixed undirected interconnection graphs are first considered, whose only stable equilibria are consensus and anti-consensus configurations under a specific assumption on the manifold. When the interconnections are allowed to be directed and/or to vary, the convergence properties of the same algorithms are weakened. Consensus for the equally-weighted complete graph $G_c$ is equivalent to synchronization of the swarm. Likewise, it appears that anti-consensus simulations for $G_c$ always converge to balancing of the swarm (when $N$ is large enough), even though that property could not be proven. In a second step, the algorithms are modified by incorporating for each agent an estimator variable for the centroid. In this setting, convergence to the (anti-)consensus states of $G_c$ can be established theoretically for time-varying and directed interconnection graphs. The problem of meaningful communication of the estimators between agents seems to restrict these last results to CC Lie groups.

The goal of the second part was to explore the issues arising when synchronization algorithms on Lie groups are designed at a *dynamical level* with limited inter-agent communications and relative position and velocity measurements only. However, in contrast with the simple integrator description of Section 3 and ([12, 14, 15],...), the Lie group solid mechanics cannot be formulated in shape entities only, because they do not retain full symmetry with respect to any synchronized motion of the agents in state space ([5, 64]). This poses some interesting questions for autonomous synchronization of mechanical systems on Lie groups.

For fixed undirected interconnection graphs, the energy shaping approach can be applied ([7, 10, 11]). The main issue here is to design artificial dissipation without referring to absolute velocities. Though control laws could be designed to ensure convergence to an arbitrary coordinated motion for Lie groups solids, the achievement of synchronization seems more difficult. The present work only provides a local result for $\mathcal{G} = SO(3)$. Another advantage of the energy shaping approach, even when absolute velocities are used, is that it avoids to explicitly counter the free rigid body dynamics as is done with the consensus tracking approach.

When the interconnection graphs are directed and/or varying, the energy shaping approach of the present work cannot be applied and a consensus tracking ([6, 49, 50, 51, 42]) approach is used. In this approach, the consensus algorithms of Section 3 are used in order to define desired trajectories, which are then tracked by the individual mechanical agents. Many tracking controllers can be used in place of the theoretically simple ones considered in Section 4.3 and the real control performance depends on the choice of the tracking controller. However, all of them require each individual to know its own absolute velocity. It may be argued that this is not a problem for autonomy, since a (generalized) gyroscopic sensor should be able to provide this information; however, the presence of the absolute velocity in the control laws biases the swarm's behavior such that only specific synchronized motions can be easily stabilized.

Practical examples on $SO(n)$ and $Grass(p,n)$ illustrate the validity of the results and provide some geometric insight. In [1], the example treatment of Section 3 is further particularized to $\mathrm{Grass}(1,2)$ and $SO(2)$, which are isometric to the circle $S^1$, in order to observe their strict equivalence with existing models and results for $S^1$ (most significantly [13],[14]). This draws a link from the present discussion to the wide literature about synchronization and balancing on the circle.

## Conclusions

The goal of the present work was to explore some issues arising in the study of coordination algorithms, with limited inter-agent communications and for various dynamical models, on *non-Euclidean manifolds.* Due to the strong role played by symmetry in the synchronization problem, Lie groups were considered a good starting point. It appears that many results in the present work strongly rely on the fact that the Lie groups are *compact.* Therefore a major issue for future work will be to consider non-compact Lie groups.

Anyway, already on compact Lie groups, many specific problems related to the non-Euclidean character of the state space have been highlighted. Though many paths towards their solution have been proposed, it would be unrealistic to believe that this first study covers most of them. Therefore the main conclusion of the present work is that consensus and coordination on Lie groups is fundamentally different from coordination on vector spaces. As such, it clearly deserves further research, whatever large the literature about consensus and coordination on vector spaces may be.

Regarding applications, it appears that many practical situations involve problems related to multi-agent systems evolving on Lie groups or homogeneous manifolds. The consensus and coordination approach, as well as its formulation as an optimization problem, seems to be a promising tool for the study of these real-world problems. In particular, it is robust to different interconnections among the agents and leaves sufficient flexibility for practical implementations (choice of parameters, low-level controllers or additional control loops).

## Perspectives for future work

Since the present report is written in the middle of ongoing research, it is important to outline several directions that can be considered in order to explore multi-agent systems and coordination in the continuity of the presented results.

**Taking a rather focused viewpoint,** a few developments close to the present results can be mentioned.

First, it would be nice to try to combine the energy shaping control of Section 4.4 with estimator variables as in Section 4.3. This could potentially achieve *global* position synchronization for *directed and time-varying* interconnection graphs, using only *relative* velocities and maybe simpler algorithms than in Section 4.3.

Another extension of the work about mechanical systems on Lie groups would be to consider more realistic settings for the control of the agents: considering actual actuators (like reaction wheels or magneto-torques on a spacecraft in $SO(3)$) and their dynamics, introducing external forces and realistic disturbances,... In such a framework, questions regarding efficiency in terms of actual performance and cost could be addressed. Additional robustness questions (what happens if the agents have slightly different mechanical properties?) and practical constraints (bounds on achievable controls, fuel balancing among the agents,...) are other issues which would become meaningful in this setting. At that point, it may become relevant to consider in more detail alternative tracking algorithms to those of Section 4.3.

Concerning the first part of the work (definition of consensus and first-order algorithms), the picture seems rather complete. Minor improvements may deal with a more detailed characterization of consensus, anti-consensus and balancing configurations on specific manifolds, like what has been done on the circle by the authors of [14].

**From a broader viewpoint,** it appears that much research work has to be done about the study of coordination on Lie groups.

Though the consensus and anti-consensus configurations described in Section 3 appear to be related to somehow "grouping" and "distributing" a swarm of agents on a homogeneous manifold, the link to actual applications is not fully clarified yet. It is planned to try to establish a link between the consensus algorithms of Section 3 and the classical problems involving multiple agents that are mentioned in Section 1, like packing and clustering tasks. It is hoped that this may provide useful new insights about the solution of these important problems.

Related to the previous point, but maybe closer to motion coordination, it is certainly interesting to look for a method to stabilize some specific isolated configuration of the agents through the use of more complex cost functions instead of $P_L$ alone. One idea, which could be explored in a near future, would be to combine a set of cost functions similar to $P_L$ but involving different powers of the relative positions $y_k^{-1}y_j$; this idea comes from [13] where it is successfully applied on the circle.

Going further towards coordinated motion control, it seems absolutely important to generalize the present results to non-compact Lie groups. Indeed, the work on $SE(2)$ and $SE(3)$ presented in [13, 14] and [12] respectively points towards very rewarding results in this direction. Instead of aiming at specific relative positions of the agents, it seems interesting to consider as a primary goal to design control laws such that the relative positions of the agents remain asymptotically constant - at any value - during the motion of the swarm. This situation is similar to the one described in Proposition 17 of Section 4. Once this is achieved, it is further possible to assign a specific coordinated motion to the swarm (like in 3-dimensional space for instance: parallel translation, rotation on circular paths or helicoidal motion). The definition and stabilization of qualitatively different motions on Lie groups is also part of current research, see also [77, 22].

The previous point becomes even more interesting when considering underactuated or nonholonomic agents. It seems that the combined constraints on the dynamics and for the achievement of a relative equilibrium strongly limit the possible combinations of relative positions of the agents and specific coordinated motion. A clear characterization of this interplay would only be a first step towards the design of control laws in this framework.

The ultimate goal of this research would be to provide a general methodology for the design of distributed control laws that asymptotically stabilize specific coordinated motions associated to general or specific relative equilbria, like in [13, 14] and [12], for possibly underactuated or nonholonomic systems that evolve on general non-compact Lie groups.

# References

[1] A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. *Submitted to SICON special issue on Control and Optimization in Cooperative Networks*, 2006.

[2] A. Sarlette, R. Sepulchre, and N.E. Leonard. Autonomous rigid body attitude synchronization. *Submitted to the 46th IEEE Conf. on Decision and Control*, 2007.

[3] D. Izzo and L. Pettazzi. Equilibrium shaping: distributed motion planning for satellite swarm. *Proc. 8th Intern. Symp. on Artificial Intelligence, Robotics and Automation in space*, 2005.

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

[5] T.R. Smith, H. Hanssmann, and N.E. Leonard. Orientation control of multiple underwater vehicles with symmetry-breaking potentials. *Proc. 40th IEEE Conf. on Decision and Control*, pages 4598–4603, 2001.

[6] J.R. Lawton and R.W. Beard. Synchronized multiple spacecraft rotations. *Automatica*, 38:1359–1364, 2002.

[7] S. Nair and N.E. Leonard. Stabilization of a coordinated network of rotating rigid bodies. *Proc. 43rd IEEE Conf. on Decision and Control*, pages 4690–4695, 2004.

[8] S. Nair, N.E. Leonard, and L. Moreau. Coordinated control of networked mechanical systems with unstable dynamics. *Proc. 42nd IEEE Conf. on Decision and Control*, 2003.

[9] S. Nair and N.E. Leonard. Stable synchronization of mechanical system networks. *Under review for SICON*, 2007.

[10] S. Nair and N.E. Leonard. Stable synchronization of rigid bodies. *In preparation*, 2007.

[11] S. Nair and N.E. Leonard (advisor). Stabilization and synchronization of networked mechanical systems. *PhD Thesis, Princeton University*, 2006.

[12] L. Scardovi, R. Sepulchre, and N.E. Leonard. Consensus based dynamic control laws for the stabilization of collective motion in three dimensional space. *Submitted to the 46th IEEE Conf. on Decision and Control*, 2007.

[13] R. Sepulchre, D. Paley, and N.E. Leonard. Stabilization of planar collective motion: all-to-all communication. *IEEE Transactions on Automatic Control*, 52(5):811–824, 2007.

[14] R. Sepulchre, D. Paley, and N.E. Leonard. Stabilization of planar collective motion: limited communication. *IEEE Transactions on Automatic Control*, to appear.

[15] R. Sepulchre, D. Paley, and N.E. Leonard. Group coordination and cooperative control of steered particles in the plane. In *Group Coordination and Cooperative Control*, volume 336 of *Lecture Notes in Control and Information Sciences*, chapter 13, pages 217–232. Springer-Verlag, 2006.

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

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

[18] N.E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Frantantoni, and R. Davis. Collective motion, sensor networks and ocean sampling. *Proceedings of the IEEE*, 95(1), January 2007.

[19] 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.

[20] J. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.

[21] R. Olfati-Saber and R.M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. *Proc. 41st IEEE Conf. on Decision and Control*, 2002.

[22] E.W. Justh and P.S. Krishnaprasad. Natural frames and interacting particles in three dimensions. *Proc. 44th IEEE Conf. on Decision and Control*, pages 2841–2846, 2005.

[23] C.R. McInnes. Potential function methods for autonomous spacecraft guidance and control. *Proc. AAS/AIAA Astrodynamics Specialist Conf.*, (95-447), 1995.

[24] J.P. Desai, J.P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[25] S.P. Hughes and C.D. Hall. Optimal configurations of rotating spacecraft formations. *Journal of the Astronautical Sciences*, 48(2-3):225–247, 2000.

[26] M.C. VanDyke and C.D. Hall. Decentralized coordinated attitude control of a formation of spacecraft. *Journal of Guidance, Control and Dynamics*, 29(5):1101–1109, 2006.

[27] M.C. VanDyke and C.D. Hall (advisor). Decentralized coordinated attitude control of a formation of spacecraft. *Master Thesis, VirginiaTech*, 2004.

[28] C. Beugnon, E. Buvat, M. Kersten, and S. Boulade. GNC design for the DARWIN spaceborne interferometer. *Proc. 6th ESA Conf. on Guidance, Navigation and Control Systems*, 2005.

[29] 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.

[30] J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[31] J.N. Tsitsiklis and D.P. Bertsekas. Distributed asynchronous optimal routing in data networks. *IEEE Transactions on Automatic Control*, 31(4):325–332, 1986.

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

[33] P. Gruber and F.J. Theis. Grassmann clustering. *Proc. 14th European Signal Processing Conference*, 2006.

[34] A. Barg. Extremal problems of coding theory. In H. Niederreiter, editor, *Coding Theory and Cryptography*, Notes of lectures at the Institute for Math. Sciences of the University of Singapore, pages 1–48. World Scientific, 2002.

[35] A. Barg and D.Y. Nogin. Bounds on packings of spheres in the Grassmann manifold. *IEEE Transactions on Information Theory*, 48(9):2450–2454, 2002.

[36] 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.

[37] J. Cortes, S. Martinez, and F. Bullo. Coordinated deployment of mobile sensing networks with limited-range interactions. *Proc. 43rd IEEE Conf. on Decision and Control*, pages 1944–1949, 2004.

[38] L. Scardovi and R. Sepulchre. Collective optimization over average quantities. *Proc. 45th IEEE Conf. on Decision and Control*, 2006.

[39] Y. Kuramoto. In *International Symposium on Mathematical Problems in Theoretical Physics*, volume 39 of *Lecture Notes in Physics*, page 420. Springer, 1975.

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

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

[42] T.R. Krogstad and J.T. Gravdahl. Coordinated attitude control of satellites in formation. In *Group Coordination and Cooperative Control*, volume 336 of *Lecture Notes in Control and Information Sciences*, chapter 9, pages 153–170. Springer-Verlag, 2006.

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

[44] L. Moreau. Stability of continuous-time distributed consensus algorithms. *Proc. 43rd IEEE Conf. on Decision and Control*, pages 3998–4003, 2004.

[45] G.A. Galperin. A concept of the mass center of a system of material points in the constant curvature spaces. *Comm. Math. Phys.*, 154:63–84, 1993.

[46] M. Moakher. Means and averaging in the group of rotations. *SIAM J. Matrix Anal. Appl.*, 24(1):1–16, 2002.

[47] L. Scardovi, A. Sarlette, and R. Sepulchre. Synchronization and balancing on the $N$-torus. *Systems and Control Letters*, 56(5):335–341, 2007.

[48] S.H. Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled nonlinear oscillators. *Physica D*, 143:1–20, 2000.

[49] A.K. Bondhus, K.Y. Pettersen, and J.T. Gravdahl. Leader/follower synchronization of satellite attitude without angular velocity measurments. *Proc. 44th IEEE Conf. on Decision and Control*, pages 7270–7277, 2005.

[50] F. Bullo and R. Murray (advisor). Nonlinear control of mechanical systems: a Riemannian geometry approach. *PhD Thesis, CalTech*, 1998.

[51] C.D. Hall, P. Tsiotras, and H. Shen. Tracking rigid body motion using thrusters and momentum wheels. *Journal of the Astronautical Sciences*, 50(3):311–323, 2002.

[52] P.A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, to be published, 2007.

[53] A. Olshevsky and J.N. Tsitsiklis. Convergence rates in distributed consensus and averaging. *Proc. 45th IEEE Conf. on Decision and Control*, pages 3387–3392, 2006.

[54] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus and flocking. *Proc. 44th IEEE Conf. on Decision and Control*, 2005.

[55] A. Sarlette, R. Sepulchre, and N.E. Leonard. Discrete-time synchronization on the $N$-torus. *Proc. 17th Intern. Symp. on Mathematical Theory of Networks and Systems*, pages 2408–2414, 2006.

[56] R. Olfati-Saber. Swarms on sphere: a programmable swarm with synchronous behaviors like oscillator networks. *Proc. 45th IEEE Conf. on Decision and Control*, 2006.

[57] K. Hueper and J. Manton. The Karcher mean of points on $SO(n)$. *Talk at Cesame (UCL, Belgium)*, 2004.

[58] S.R. Buss and J.P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics*, 20(2):95–126, 2001.

[59] T.D. Downs. Orientation statistics. *Biometrika*, 59:665–676, 1972.

[60] P.A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220, 2004.

[61] R.W. Brockett. Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and its Applications*, 146:79–91, 1991.

[62] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1999.

[63] U. Helmke and J.B. Moore. *Optimization and dynamical systems.* Springer, 1994.

[64] H. Hanssmann, N.E. Leonard, and T.R. Smith. Symmetry and reduction for coordinated rigid bodies. *Eur. Journal of Control*, (1), 2006.

[65] J.E. Marsden. *Lectures on Mechanics.* Cambridge University Press, 1992.

[66] J.E. Marsden and T.S. Ratiu. *Introduction to mechanics and symmetry.* Springer Verlag, 1994.

[67] J.E. Marsden, T. Ratiu, and A. Weinstein. Semidirect products and reduction in mechanics. *Transactions of the AMS*, 281(1):147–177, 1984.

[68] S.M. Jalnapurkar and J.E. Marsden. Stabilization of relative equilibria II. *Regul.Chaotic Dyn.*, 3(3):161–179, 1998.

[69] F.R.K. Chung. *Spectral Graph Theory.* Number 92 in Regional Conference Series in Mathematics. AMS, 1997.

[70] V. Arnold. *Mathematical methods of classical mechanics.* Springer, 1989.

[71] A. Machado and I. Salavessa. Grassmannian manifolds as subsets of Euclidean spaces. *Res. Notes in Math.*, 131:85–102, 1985.

[72] M. Brookes. Matrix reference manual. *Imperial College London*, 1998-2005.

[73] K. Mischaikow, H. Smith, and H.R. Thieme. Asymptotically autonomous semi-flows: chain recurrence and Lyapunov functions. *Transactions of the AMS*, 347(5):1669–1685, 1995.

[74] M. Hurley. Chain recurrence, semiflows, and gradients. *Journal of Dynamics and Differential Equations*, 7(3):437–456, 1995.

[75] J. C. Willems. Lyapunov functions for diagonally dominant systems. *Automatica*, 12:519–523, 1976.

[76] D.H.S. Maithripala, J.M. Berg, and W.P. Dayawansa. Almost-global tracking of simple mechanical systems on a general class of Lie groups. *IEEE Transactions on Automatic Control*, 51(1):216–225, 2006.

[77] E.W. Justh and P.S. Krishnaprasad. Equilibria and steering laws for planar formations. *Systems and Control Letters*, 52:25–38, 2004.