Connectivity Control on Lie Groups

Aykut C Satici and Mark W Spong

***Abstract*—The preservation of connectivity in networks is critical to the success of existing algorithms designed to achieve various goals. Control laws increasing the connectivity of a given state-dependent graph have been formulated for agents evolving on the Euclidean space. In this paper, we show that similar control laws can be adapted to more general spaces. In particular, we consider agents whose configuration space is represented by a Lie group. In addition, we will require the resulting control law to be symmetric with respect to the natural group action, with the intention that additional controllers for the group action may be implemented on top of the connectivity controllers.**

I. Introduction

The problem of maintaining connectivity in mobile robot networks has been receiving increasing attention in recent years. Mobile robot networks afford an inexpensive and robust method for achieving certain coverage tasks or cooperative missions. Many of the algorithms employed to achieve those tasks depend on communication between any two robots. This is possible only when the network is connected.

For many applications, the edges or links in the mobile robot network are functions of the relative positions of the nodes in the network. Thus, the connectivity of the network is affected by the motion of the robots, and the motion controllers must be tasked with maintaining connectivity, in addition to achieving other goals.

A good review of different methods to control and maintain connectivity can be found in [1]. The connectivity can be maintained in a centralized or decentralized manner. One of the simplest methods of ensuring connectivity, assuming that the network is initially connected, is to guarantee that existing edges in the network are maintained for all time [2], [3]. This is especially true of many decentralized connectivity preserving methods. A notable exception is found in [4], where the authors propose algorithms to decide if edges may be deleted while still ensuring that a spanning subgraph exists, based on local estimates of the network topology. Methods that use potential functions to maintain edges suffer from the phenomenon of the overall potential blowing up (and large control effort due to the gradient) whenever a new edge is created. This can be overcome using a hysteresis protocol when adding the potential corresponding to new edges.

Aykut C Satici is with the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, Texas, USA acsatici@utdallas.edu

Mark W Spong is the Dean of the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, Texas, USA mspong@utdallas.edu

This research was partially supported by the National Science Foundation Grants ECCS 07-25433 and CMMI-0856368, by the University of Texas STARs program, and by DGIST R&D Program of the Ministry of Education, Science and Technology of Korea(12-BD-0101).

A more centralized method for maintaining connectivity among a group of mobile robots is to maximize the second smallest eigenvalue of the graph Laplacian [5], when the edge strengths are non-increasing functions of the distance between robots. The resulting graph is fully connected, as seen in the simulation results in [5]. This method is effective for solving rendezvous problems, and can be extended to other applications [1].

The controllers presented in nearly all of these works in the literature exclusively consider agents evolving on vector spaces with first or second order dynamics. In this work, we design analogous controllers for agents evolving on Lie groups, a generalization of Euclidean space. Not only do the controllers we present increase the connectivity of a state-dependent connected set of agents, but they also are shown to preserve the symmetry of the configuration space. This feature allows the introduction of further control action that is aimed at imposing desired motion along the symmetry directions, although this latter opportunity is not exploited in this paper.

II. Background

In this section we give a recount of concepts from graph theory used to model the connectivity of a robotic network and recall briefly the basic definitions of Lie groups and Riemannian manifolds. We also provide the dynamic model of the systems that we are going to consider.

A. Graph Theory

A weighted graph $\mathbb{G}$ is a tuple consisting of a set of vertices $\mathcal{V}$ (also called nodes) and a function $\mathcal{W}$, that is,

$$\mathbb{G} = (\mathcal{V}, \mathcal{W})$$

where $\mathcal{V} = \{1, ..., N\}$ denoted the set of nodes. The function $W : \mathcal{V} \times \mathcal{V} \times \mathbb{R}_+ \to \mathbb{R}_+$ is used to compute the weights of the edges in $\mathbb{G}$, such that

$$w_{ij}(t) = W(i, j, t); \tag{1}$$

If $w_{ij}(t) = 0$, then there is no connection between nodes i and j. The edge weights give rise to the graph Laplacian $\mathcal{L} \in \mathbb{R}^{N\times N}$ defined as

$$\mathcal{L}_{ij}(t) = \begin{cases} -w_{ij}(t) & \text{if } i \neq j \\ \sum_{k\neq i} w_{ik}(t) & \text{if } i = j \end{cases}$$

The Laplacian gives us a measure of the connectivity of the graph $\mathbb{G}$ since the number of connected components in the graph is equal to the number of zero eigenvalues of $\mathcal{L}$. Thus, for the graph to be connected, only one eigenvalue of $\mathcal{L}$ will be zero. The second smallest eigenvalue $\lambda_2(\mathcal{L})$ thus becomes an indicator of connectivity in the graph.

978-1-4673-5769-2/13/$31.00 ©2013 IEEE

The Laplacian $\mathcal{L}$ can be converted to a matrix $\mathcal{M} \in \mathbb{R}^{N-1\times N-1}$, whose eigenvalues are the largest $N-1$ eigenvalues of $\mathcal{L}$. The matrix $\mathcal{M}$ is given by

$$\mathcal{M} = \mathcal{P}^T \mathcal{L} \mathcal{P} \tag{2}$$

where $\mathcal{P} \in \mathbb{R}^{N\times N-1}$ satisfies $\mathcal{P}^T \mathbf{1} = 0$ and $\mathcal{P}^T\mathcal{P} = I_{N-1}$. Thus, the determinant of $\mathcal{M}$ vanishes if and only if $\lambda_2(L)$ vanishes.

In this work, we are going to assume that the weights w_{ij} are dependent on time through the states; that is, $w_{ij} = \mathcal{W}(g_i, g_j) : G \times G \to \mathbb{R}_+$, where G is the configuration space of each agent.

B. Lie Groups

Following the exposition in [6], a *Lie group* is a group G with a differentiable structure such that the mapping $G\times G \to G$ given by $(g, h) \mapsto gh^{-1}$, $g, h \in G$, is differentiable. It follows then that *translations* from the left L_g and *translations* from the right R_g given by $L_g : G \to G$, $L_g(h) = gh$; $R_g : G \to G$, $R_g(h) = hg$ are diffeomorphisms.

A Riemannian metric on a differentiable manifold M is a correspondence that associates to each point p of M an inner product $\langle\cdot,\cdot\rangle$ on the tangent space T_pM which is a symmetric, bilinear, positive definite and differentiable two covariant tensor.

We say that a Riemannian metric on a Lie group G is *left invariant* if $\langle u, v\rangle_y = \langle d(L_x)_y u, d(L_x)_y v\rangle_{L_x(y)}$ for all $x, y \in G$, $u, v \in T_yG$, that is, if L_x is an isometry. Analogously, we define a *right invariant Riemannian metric* and call a metric *bi-invariant* if it is both left and right invariant. We can always endow a Lie group with a left or right invariant metric. On the other hand, not all Lie groups admit bi-invariant metrics. One sufficient condition for the existence of a bi-invariant metric is the Lie group be compact.

We say that a differentiable vector field X on a Lie group G is *left-invariant* if $dL_xX = X$, for all $x \in G$; thus the left-invariant vector fields are completely determined by their values at a single point of G. It can be shown that the bracket of two left-invariant vector fields is again a left-invariant vector field and with this bracket operation T_eG is termed the *Lie algebra*, $\mathfrak{g}$ of G. With this understanding, the elements in $\mathfrak{g}$ can be thought of either as vectors in T_eG or as left-invariant vector fields on G.

C. System Dynamics

In the present work, the dynamics of each of the agents are assumed to be determined by the geodesics of the space that they live in. This is not a restriction for a large class of rigid-body systems because it is well-known that a strong connection between Lagrangian dynamics and geodesics of the configuration space exists, and in fact, in most of the cases they coincide.

The geodesics of a space, of course, requires the existence of a metric. In particular, we are interested in endowing our configuration space with a Riemannian metric. To begin with, suppose that the configuration space of each of the agents is a Lie group, G, with a left-invariant metric $\langle\cdot,\cdot\rangle$,

We assume that the left-invariant kinematics of each system is given by

$$\frac{d}{dt} g_k = dL_{g_k} \xi_k \tag{3}$$

where $\xi_k \in \mathfrak{g}$ is the left-invariant (body-frame) velocity of the agent. The dynamics of the k^{th} agent may also be found by the geodesic equation

$$\nabla_{\dot g_k} \dot g_k = B u_k \tag{4}$$

where ∇ represents the Riemannian (Levi-Civita) connection on the tangent bundle, $B \in \mathbb{R}^{n\times m}$ is a constant matrix with full column rank and specifies the range of the control term $u_k \in \mathbb{R}^m$. Note that (3)-(4) form a second-order differential equation and the control term u_k affect the acceleration of each agent.

A very familiar example of the presently described dynamical system is the rigid-body orientation, whose configuration space is $G = SO(3)$. The equations of motion governing the rigid-body orientation is

$$R_k^T \frac{d}{dt} R_k = \Omega_k \tag{5a}$$

$$J\dot\omega = J\omega \times \omega + \tau \tag{5b}$$

We immediately recognize the first of these equations to be of the form (3), as the tangent map of the left translation in SO(3) is given by multiplication by the rotation matrix. The left-invariant velocity vector Ω_k is a skew-symmetric matrix, the Lie algebra $\mathfrak{so}(3)$ of $SO(3)$. Similarly, the second equation is an example of the geodesic equation (4), with fully-actuated dynamics. We have also identified $\mathfrak{so}(3)$ with $\mathbb{R}^3$, where appropriate.

III. Control Design

We consider a group of n dynamical systems each one with n-dimensional configuration space G that is a Lie group. Denote the space in which all of the agents live $Q = G^N$. The extent at which the agents are connected is measured by the determinant of the matrix $\mathcal{M}$ defined in equation (2).

We are primarily interested in the problem of increasing the connectivity of a given system of agents, under the constraint that the resulting dynamical system still maintain the symmetry of the Lie group. In other words, whenever $q = (g_1(t), g_2(t), \ldots, g_N(t))$ is a solution to (3) and (4), then so is $g \cdot q := (gg_1(t), gg_2(t), \ldots, gg_N(t))$ for any constant $g \in G$. Here, the dot represents the action of the Lie group G onto itself by left translation.

Furthermore, we would like the connectivity control to have the flexibility of reaching a desired value of connectivity and turning off at that point while disallowing losing connectivity at all costs.

A. Velocity Level Connectivity Controller

We start the design of the connectivity controller by parametrizing the space over which the connectivity controller is active. This is achieved by sandwiching the function $\det(\mathcal{M})$ by two scalars $\underline{\alpha}$ and $\bar\alpha$.

$$\underline{\alpha} \le \det(\mathcal{M}) = \det(\mathcal{P}^T \mathcal{L}(q) \mathcal{P}) \le \bar\alpha$$

This defines a submanifold S of the total configuration space Q where the connectivity controller is active. The connectivity controller will ensure the lower bound $\underline{\alpha}$ is never reached, once the robots are started in $\{q \in Q : \det(\mathcal{M}(q)) > \underline{\alpha}\}$. On the other hand, in the complement of this subset S, the connectivity controller will not be active.

These bounds immediately correspond to bounds on $\underline{\lambda}(\mathcal{M})$ once the determinant is seen as the product of the eigenvalues. In other words, a lower bound $\underline{\alpha} > 0$ on the determinant also bounds the second smallest eigenvalue of the graph Laplacian away from zero. Similarly the number $\bar{\alpha} \le N^{N-1}$ corresponds to an upper threshold for the second smallest eigenvalue. The absolute upper bound N^{N-1} on $\bar{\alpha}$ exists because of the very nature of the graph Laplacian.

The aim of the controller will be to never let the robots reach a configuration where $\det \mathcal{M} \le \underline{\alpha}$ and be unresponsive whenever $\det \mathcal{M} \ge \bar{\alpha}$. We address this problem through the following potential function $V_c(q)$ [7]:

$$D(q) := \det(\mathcal{M}(q)) \tag{6a}$$

$$V_c(D) = \left(\min\left\{0, \frac{D^2 - \bar{\alpha}^2}{D^2 - \underline{\alpha}^2}\right\}\right)^2 \tag{6b}$$

The range of this function is $[0, \infty]$. It blows up whenever the determinant approaches the lower bound and is zero whenever the determinant is greater than the upper bound. We would like to prescribe a potential-energy shaping control law defined as the gradient of this function V_c. We evaluate this gradient as follows

$$\text{grad } V_c = \begin{cases} 0 & \text{if } D \le \underline{\alpha} \\ \beta(q)\text{tr}\left(\mathcal{M}^{-1}\text{grad } \mathcal{M}\right) & \text{if } \underline{\alpha} < D < \bar{\alpha} \\ 0 & \text{if } \bar{\alpha} \le D \end{cases} \tag{7}$$

where

$$\beta(q) = 4\frac{\left(\bar{\alpha}^2 - \underline{\alpha}^2\right)\left(D^2 - \bar{\alpha}^2\right)}{\left(D^2 - \underline{\alpha}^2\right)^3} D^2 < 0.$$

It is important to note that in (7), the multiplication $\mathcal{M}^{-1}$grad $\mathcal{M}$ is carried out componentwise. For instance, in the simulation section IV, we shall have an example where each element of $\mathcal{M}$ is a function of a matrix, so its gradient with respect to this matrix shall be a matrix. In this case, the term $\mathcal{M}^{-1}$grad $\mathcal{M}$ will generate n^2 matrices, one for each element of $\mathcal{M}$. Once these each of these matrices are contracted with the trace function, a matrix is achieved as the final form of the gradient.

The final form of the controller will be achieved with a backstepping argument. Therefore, we first regard ξ_k in equation (3) as an input to the system and come up with a desired value ξ_k^d for this quantity for each $k \in \mathcal{V}$

Theorem III.1. *Under the assignment*

$$\xi^d = -k_p dL_{g_k^{-1}} \text{ grad } V_c \tag{8}$$

with $k_p > 0$, the first-order agents with dynamics (3) *converges to the set $E = \{q \in Q : \det(\mathcal{M}(q)) \ge \bar{\alpha}\}$ and the graph G whose nodes the robots represent stays connected for all time.*

Proof: We take the time derivative of the potential function V_c along the trajectory of the system (3), yielding

$$\begin{aligned}\dot{V}_c &= dV_c(q) \cdot \dot{q} \\ &= -k_p dV_c(q) \cdot \text{ grad } V_c \\ &= -k_p \langle \text{grad } V_c, \text{grad } V_c \rangle\end{aligned}$$

which is smaller than zero whenever $\det(\mathcal{M}(q)) < \bar{\alpha}$.

The second statement follows from the fact that the level sets $V_c^{-1}([0,\gamma]) = \{q \in Q : V_c(q) \le \gamma\}$ of $V_c(q)$ are positively invariant. ∎

The next proposition identifies the condition for the control law (8) to render (3) invariant under action of G onto itself by group translation.

Proposition III.1. *If the potential function V_c of equation* (6b) *is invariant under the group translation, then so is the kinematics* (3).

Proof: By the hypothesis, we have $V_c(L_g q) = V_c(q)$, which implies upon differentiating

$$d(V_c \circ L_g)(q) = dV_c(q)$$

Thus, for any left-invariant vector field Y of G,

$$\langle \text{grad } V_c \circ L_g, Y \rangle_{L_g q} = \left\langle \text{grad } V_c, dL_g^{-1} Y \right\rangle_q$$

and so, by the left-invariance of the Riemannian metric, we deduce that

$$dL_g(q) \cdot \text{grad } V_c(q) = \text{grad } (V_c \circ L_g)(q) \tag{9}$$

Next, consider the differential equation that $L_g q$ satisfies, given (3)

$$\begin{aligned}\frac{d}{dt}(L_g q(t)) &= dL_g(q(t)) \cdot \dot{q}(t) \\ &= -k_p dL_g(q(t)) \cdot \text{grad } V_c(q(t)) \\ &= -k_p \text{ grad } (V_c \circ L_g)(q(t))\end{aligned} \tag{10}$$

where the last step follows from (9). As desired, (10) says that the differential equations (3) possesses group translation symmetry provided that V_c is invariant under this action. ∎

Finally, we analyze when $V_c(q)$ of equation (6b) is invariant under the group action. We have

$$\begin{aligned} V_c(g \cdot q) &= \left(\min\left\{0, \frac{\det \mathcal{M}(g \cdot q)^2 - \bar{\alpha}^2}{\det \mathcal{M}(g \cdot q)^2 - \underline{\alpha}^2}\right\}\right)^2 \\ &= \left(\min\left\{0, \frac{\det \mathcal{M}(q)^2 - \bar{\alpha}^2}{\det \mathcal{M}(q)^2 - \underline{\alpha}^2}\right\}\right)^2 \\ &= V_c(q) \end{aligned} \tag{11}$$

whenever $\mathcal{M}(g \cdot q) = \mathcal{M}(q)$, which, in turn, means that the weight functions has to satisfy $w_{ij}(g \cdot q) = w_{ij}(q)$. In this work, we are going to consider $w_{ij}(q) = \psi(d(g_i, g_j))$, a function of the distance between agent i and agent j. The distance is equal to the length of the curve $\exp(\xi_{ij})$ such that $\exp(\xi_{ij}) = g_i^{-1} g_j$. This length is equal to $\sqrt{\langle \xi_{ij}, \xi_{ij} \rangle}$, but since $g_i^{-1} g_j = g_i^{-1} g^{-1} g g_j$, we have $d(g_i, g_j) = d(g g_i, g g_j)$. The outer function ψ, on the other hand, maps the real numbers to the real numbers in a smooth fashion. It is included to

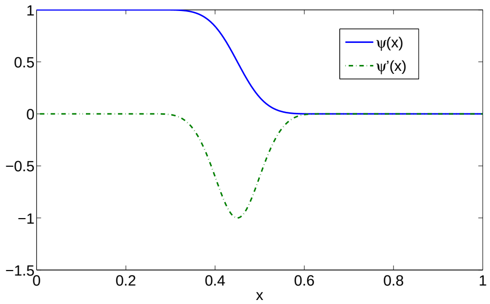

Fig. 1: Bump function

normalize the weights to the range $[0, 1]$, with the value 1 when the distance is zero and 0 when they are a specified distance away from each other. The formula of this function can be written as

$$\psi(x) = \begin{cases} 1 & \text{if } x \leq \rho_1 \\ \frac{\exp\left(-\frac{1}{\rho_2 - x}\right)}{\exp\left(-\frac{1}{\rho_2 - x}\right) + \exp\left(\frac{1}{\rho_1 - x}\right)} & \text{if } \rho_1 \leq x \leq \rho_2 \\ 0 & \text{if } \rho_2 \leq x \end{cases} \quad (12)$$

which can also be recognized as the definition of a bump function from differential geometry. Its graphical representation is given in figure 1. The values of ρ_1 and ρ_2 determine the values at which two agents are as connected as it gets and start getting connected, respectively. These are assigned by the user, according to the particular application. Note that under these conditions, we have that $w_{ij}(g \cdot q) = w_{ij}(q)$, so that $\mathcal{M}(g \cdot q) = \mathcal{M}(q)$ and so $V_c(g \cdot q) = V_c(q)$.

B. Backstepping the controller to acceleration level

With the desired velocity signal presented in equation (8), we need to design a tracking algorithm that makes ξ_k converge to ξ_k^d, for all $k \in \mathcal{V}$. In this section, we are going to assume that the control input matrix B of equation (4) is square $n \times n$ so that the system is fully actuated. There are quite a few approaches to make this possible in the control literature; two of which are *computed torque* and *high gain*. In both cases, we would like to achieve exponential convergence of ξ_k to ξ_k^d, i.e., we would like to have

$$\frac{d}{dt}\left(\xi_k - \xi_k^d\right) = -\mu\left(\xi - \xi_k^d\right), \quad \mu > 0 \quad (13)$$

The computed torque method achieves (13) by cancelling the free dynamics of each agent. Put another way, one computes the Christoffel symbols associated with the connection (4) and includes this term in the controller cancelling its nonlinear effect yielding the desired form. Explicitly, this amounts to the following process:

- Express all of the left-invariant vector fields in equation (4) in the Lie algebra,
- Choose a basis $\{X_i\}_1^n$ for the Lie algebra and write out the geodesic equation,
- Include control action to cancel out the nonlinear coupling.

For agent k, the geodesic equation (4) on the Lie algebra looks like

$$\dot{\xi}_k^l + \sum_{i,j=1}^{n} \xi_k^i \xi_k^j \Gamma_{ij}^l = b_l u_k^l \quad (14)$$

for all $l \in \{1, \ldots, n\}$, where ξ_k^l is the coordinates of ξ_k and b_l is the rows of the input matrix B in the selected basis $\{X_i\}_1^n$. To implement the computed torque controller which yields exponential convergence of the Lie algebra element ξ_k to $\xi_k^{(d)}$, we let

$$u_k = B^{-1}\left(-h(g_k, \xi_k) + \dot{\xi}_k^{(d)} + \mu(\xi_k^{(d)} - \xi_k)\right) \quad (15)$$

for each $k \in \mathcal{V}$, with $h(g_k, \xi_k)$ capturing the nonlinear terms in the preceding equation. On the other hand, if we do not want to cancel the natural dynamics on the grounds of robustness, for instance, we may choose to dominate them with a sufficiently high gain on the velocity term in equation (15) and omit the term $h(g_k, \xi_k)$. In this case,

$$u_k = B^{-1}\left(\dot{\xi}_k^{(d)} + \mu(\xi_k^{(d)} - \xi_k)\right) \quad (16)$$

for each $k \in \mathcal{V}$, and $\mu > \mu^*$. In both cases one achieves exponential convergence of ξ_k to $\xi_k^{(d)}$, because the linearization of the resulting equations at any point yields exactly (13).

IV. Simulation Results

We will be validating the controllers presented in section IV on two Lie groups, namely, $S^1 \cong SO(2)$, the space of rotations of the Euclidean 2-space and $SO(3)$, the space of rotations of Euclidean 3-space.

A. The circle: S^1

The space $S^1 = \{z = \rho e^{i\theta} \in \mathbb{C} : \rho = 1\}$ is a Lie group where the left translation operation is given by addition on the power of the exponential, i.e., $z_1 z_2 = e^{i\theta_1} e^{i\theta_2} = e^{i(\theta_1 + \theta_2)}$. This is the space on which the rotation of a planar rigid body is naturally modeled. Its Lie algebra is simply the real line. From now on, we are going to identify the circle with the space $\mathbb{R}/\sim$, where $x_1 \sim x_2$ if and only if $x_1 - x_2 = 2m\pi$, $m \in \mathbb{Z}$.

We start constructing the elements of the matrix $\mathcal{M}$ by determining the weight functions w_{ij}. A convenient formula for the distance in S^1, that is invariant under translation by a constant angle ϕ is given by

$$d(\theta_i, \theta_j) = 2\sin\left(\frac{\theta_i - \theta_j}{2}\right) \quad (17)$$

This distance takes values between 0 and 2. We construct the matrix $\mathcal{M}$ and form the potential function V_c as in equation (6b). Taking its gradient with respect to agent k, we find

$$\frac{\partial}{\partial \theta_k} d(\theta_k, \theta_j) = \cos\left(\frac{\theta_k - \theta_j}{2}\right) \quad (18)$$

Once we have the gradients of the distance functions between each agent $i, j \in \mathcal{V}$, we construct the matrix grad $\mathcal{M} = \mathcal{P}^T(\text{grad } \mathcal{L})\mathcal{P}$. Each of the elements of $\mathcal{L}$ is the bump function ψ of equation (12) applied on the distances $d(\theta_i, \theta_j)$. After this construction, we can apply the controls (8) and (15) to the second-order dynamics given by

$$\ddot{\theta}_k = u_k \quad (19)$$

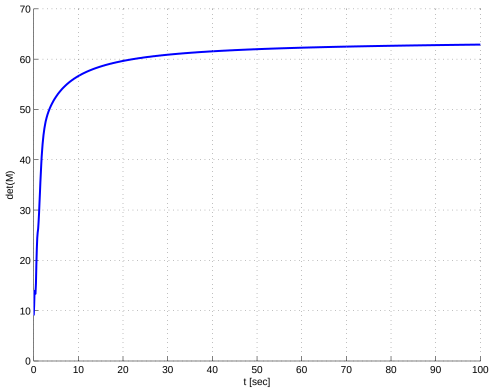

Fig. 2: Determinant of $\mathcal{M}$

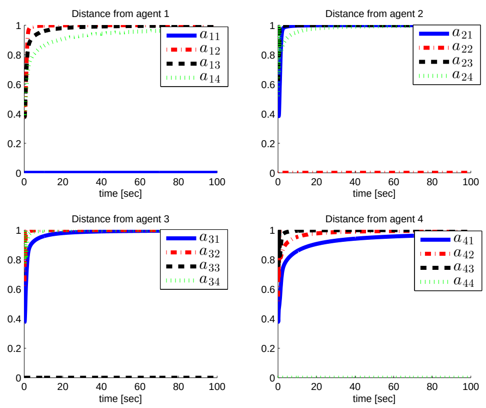

Fig. 3: The adjacency matrix elements

In the simulation, we have $N = 4$ robots and have set the desired value for the determinant to be its maximum achievable, N^{N-1}. The gains, k_p and μ of equations (8) and (15), respectively, are chosen to be 2.0×10^{-2} and $\mu = 100.0$ The figure 2, shows this value is actually where the system is headed asymptotically. The corresponding evolution of the individual components of the adjacency matrix is provided in figure 3. The initial and final configurations of the agents are depicted in figure 4.

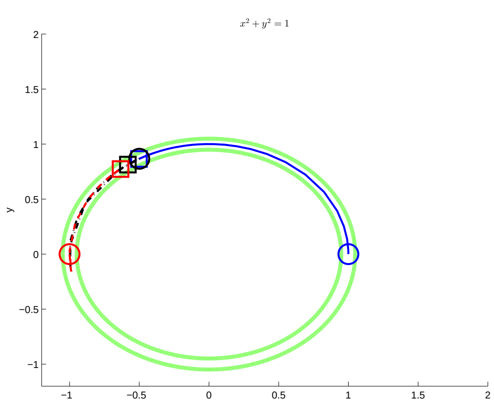

Fig. 4: Initial and final orientation

B. Euclidean 3-space rotations: $SO(3)$

The space $SO(3) = \{R \in GL(3, \mathbb{R}) : R^T R = I_3, \det R = 1\}$ is a Lie group where the left translation operation is given by multiplication on the left and the inversion is given by the usual matrix inversion. This is the space on which the rotation of a rigid body is naturally modeled. Its Lie algebra $\mathfrak{so}(3) = \{\Omega \in GL(3, \mathbb{R}) : \Omega^T = -\Omega, \mathrm{tr}(\Omega) = 0\}$ is isomorphic to $\mathbb{R}^3$. We are going to use the lower case ω for the representation of an element of $\mathfrak{so}(3)$ in $\mathbb{R}^3$, and the capital Ω for its skew-symmetric matrix representation.

We start constructing the elements of the matrix $\mathcal{M}$ by determining the weight functions w_{ij}. Suppose $w_{ij} \in \mathfrak{so}(3)$ is such that $\exp(\omega_{ij}) = R_i^T R_j$. A convenient formula for the distance in $SO(3)$ is given by

$$d(R_i, R_j) = \|\omega_{ij}\| = \arccos\left(\frac{\mathrm{tr}\left(R_i^T R_j\right) - 1}{2}\right) \tag{20}$$

Note its invariance with respect to translation via a constant group element $R \in SO(3)$. This distance takes values between 0 and π. Once we construct the matrix $\mathcal{M}$, we form the potential function V_c as given in equation (6b), and take its gradient with respect to agent k. The only nontrivial process is that of taking the gradient. To do that, note that we can take this derivative in the ambient space $\mathbb{R}^{3\times 3}$ and project the result onto the tangent space $T_R SO(3)$ of $SO(3)$ at R to arrive at the result. Thus, on $\mathbb{R}^{3\times 3}$,

$$\begin{aligned} \frac{\partial}{\partial R_k} d(R_k, R_j) &= \frac{-\frac{1}{2}}{\sqrt{1 - \left(\frac{\mathrm{tr}\left(R_k^T R_j\right) - 1}{2}\right)^2}} \frac{\partial \mathrm{tr}\left(R_k^T R_j\right)}{\partial R_k} \\ &= \frac{-\frac{1}{2}}{\sqrt{1 - \left(\frac{\mathrm{tr}\left(R_k^T R_j\right) - 1}{2}\right)^2}} R_j \end{aligned} \tag{21}$$

Projecting this expression onto $T_{R_k} SO(3)$, we get the desired gradient of the distance function on $SO(3)$

$$\begin{aligned} &\mathrm{Proj}_{T_{R_k} SO(3)} \frac{\partial}{\partial R_k} d(R_k, R_j) \\ &= \frac{-\frac{1}{2}}{\sqrt{1 - \left(\frac{\mathrm{tr}\left(R_k^T R_j\right) - 1}{2}\right)^2}} R_k \left(\frac{R_k^T R_j - R_j^T R_k}{2}\right) \end{aligned} \tag{22}$$

Once we have the gradients of the distance functions between each agent $i, j \in \mathcal{V}$, we construct the 4-tensor grad $\mathcal{M}$ by noticing that for each element of $\mathcal{M}$, we have another matrix, the gradient of w_{ij} with respect to one of its arguments R_k. After going carefully through this involved procedure, we can apply the controls (8) and (15) to the second-order dynamics given by the usual rigid-body attitude dynamics

$$R_k^T \frac{d}{dt} R_k = \Omega_k \tag{23a}$$

$$J\dot{\omega} = J\omega \times \omega + \tau \tag{23b}$$

In the simulation, we have $N = 4$ robots and have set the desired value for the determinant to be its maximum

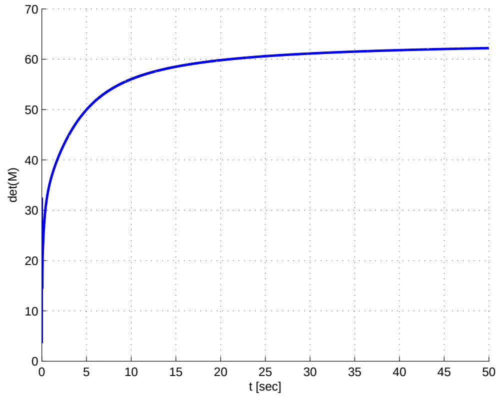

Fig. 5: Determinant of $\mathcal{M}$

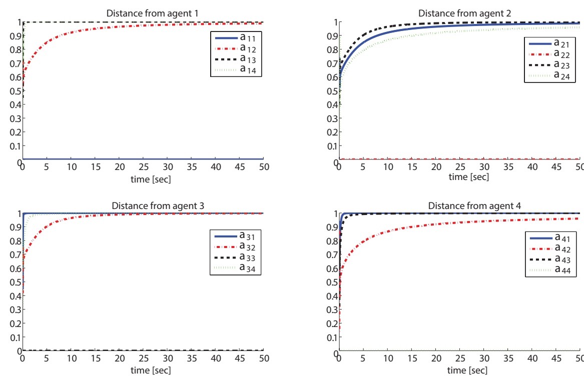

Fig. 6: The adjacency matrix elements

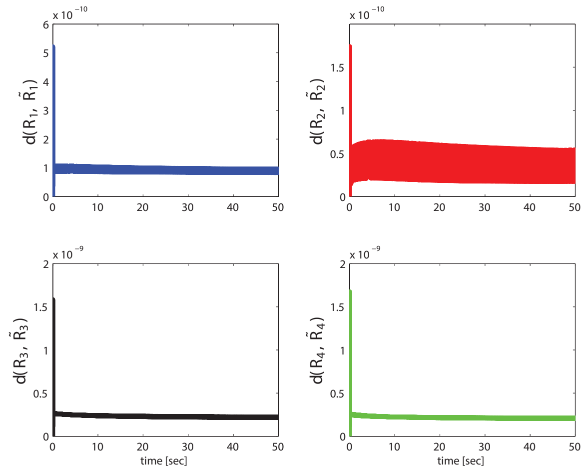

Fig. 7: Distances between two solutions whose initial conditions are separated by a group symmetric action

achievable, N^{N-1}. We choose the inertia matrix J, for each of the agents as the identity matrix, although, it may be any constant positive definite matrix. The gains, k_p and μ of equations (8) and (15), respectively, are chosen to be 5.0×10^{-3} and $\mu = 100.0$. The figure 5, shows this value is actually where the system is headed asymptotically. The corresponding evolution of the individual components of the adjacency matrix is provided in figure 6.

A second simulation is carried out, where, the agents are started at their previous locations translated by a constant rotation matrix Q. What we expect to see is that solution of the system, when multiplied by the inverse of Q yield exactly the previous solution. This, indeed turns out to be the case as figure 7 illustrates. This figure plots the distances between two solutions R and R' of the rigid-body attitude dynamics with $R'(0) = QR(0)$. We observe that to the precision of the integration routine (ode45) we have $Q^T R'(t) = R(t)$, $\forall t \in [t_i, t_f]$. Thus the desired symmetry has indeed been preserved. The distance function used to generate this plot is the same as the one used to generate the adjacency matrix.

V. Conclusion

In this paper, we have presented a connectivity control method for agents evolving on Lie groups based on maximization of the product of nonzero eigenvalues of the graph Laplacian $\mathcal{L}$. We show that the connectivity control maintains connectivity by increasing the connectivity away from zero whenever it is below a certain threshold. The method also respects the symmetry of the group in the sense that translating the agents by the constant amount does not change the behavior of the agents in any way. This is made precise by showing that if the system's differential equations are satisfied by a particular solution, then translating the solution by the action yields another solution. This fact, in particular, yields the possibility of designing an additional controller that acts on the directions in which the group acts. This might, for instance, be used to drive the agents as a whole in the configuration space while the connectivity is maintained at a desired value.

A decentralized version of the connectivity control may be derived using the estimation method of $\lambda_2(\mathcal{L})$ as in [8].

Simulations demonstrate the properties of the controllers: the convergence of the connectivity measure of the agents to its desired value and the preservation of the group symmetry.

References

[1] M. Zavlanos, M. Egerstedt, and G. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525 –1540, sept. 2011.

[2] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *Robotics, IEEE Transactions on*, vol. 23, no. 4, pp. 693 –703, aug. 2007.

[3] M. Zavlanos and G. Pappas, "Controlling connectivity of dynamic graphs," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, dec. 2005, pp. 6388 – 6393.

[4] ——, "Distributed connectivity control of mobile networks," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1416 –1428, dec. 2008.

[5] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *Automatic Control, IEEE Transactions on*, vol. 51, no. 1, pp. 116 – 120, jan. 2006.

[6] M. do Carmo, *Riemannian Geometry*, ser. Mathematics (Boston, Mass.). Birkhäuser, 1992. [Online]. Available: http://books.google.com/books?id=uXJQQgAACAAJ

[7] S. Mastellone, D. Stipanovic, and M. Spong, "Remote formation control and collision avoidance for multi-agent nonholonomic systems," in *Robotics and Automation, 2007 IEEE International Conference on*, april 2007, pp. 1062 –1067.

[8] P. Yang, R. Freeman, G. Gordon, K. Lynch, S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity in mobile sensor networks," in *American Control Conference, 2008*, june 2008, pp. 2678 –2683.