2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009



# Constrained Invariant Motions for Networked Multi-Agent Systems

Mauro Franceschelli*, Magnus Egerstedt†, Alessandro Giua* and Cristian Mahulea ‡

*Electrical and Electronic Engineering
University of Cagliari
09123 Cagliari, Italy
*mauro.franceschelli@diee.unica.it*
*giua@diee.unica.it*

†Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
*magnus@ece.gatech.edu*

‡ Computer Science and Systems Engineering
University of Zaragoza
50018 Zaragoza, Spain
*cmahulea@unizar.es*

***Abstract*— In this paper we propose a methodology to solve the constrained consensus problem, i.e., the consensus problem for multi-agent systems with constrained dynamics. We propose a decentralized one-step horizon optimization problem to be solved iteratively by the agents to achieve rendezvous at the centroid of the network while ensuring the connectivity of the network and the feasibility of the agents motion respect to their constrained kinematics. We also provide simulations of the algorithm behavior.**

## I. Introduction

The consensus problem, i.e., the problem of having a collection of agents' states reach a common value in the presence of network and information sharing constraints, has recently received considerable attention. For a representative sample, see [1], [2], [4], [6], [9], [8], [11], [16]. A common approach to this problem is to use a linear, nearest neighbor control strategy, resulting in a linear dynamic system driven by the *graph Laplacian* associated with the underlying network topology.

In [14] was proposed a method to probe a network of agents executing a Laplacian-based control strategy with an application to fault detection. Furthermore a decentralized algorithm was proposed to recover the initial network centroid for a network of agents after a failure detection dragged the network away from the desired location.

In this paper we propose a framework to address the Constrained Consensus Problem, i.e., the consensus problem with constrained agents' trajectories and constrained mission objectives, based on the results of [14].

The proposed methodology follows decentralized model predictive control applied to multi-agent systems [12], [13]. Our approach differs form the ones in the literature in that it allows to perform consensus on the average in the presence of constraints. In particular we show that the iterative solution of a one step-horizon optimization problem, involving only information locally available by the agents without any communication, can efficiently solve complex constrained consensus problem. We focus on the application of our algorithm to rendezvous in multi-agent systems assuming single integrator agents with bounded speed. Such assumption is taken to decouple the motion coordination problem from the low level control of the single agent. The motion coordination algorithm basically specifies the set point that the low level controller need to track within a specified time horizon. The constraints that we consider are kinematic constraints of the agents, network constraints such as network connectivity preservation and mission constraints such rendezvous at the initial network centroid while ensuring all the other constraints along their motion.

## II. Network Model

We will be considering networks of agents, whose nominal state evolution is governed by a discrete time consensus equation that can be written quite generally as

$$x(k+1) = Px(k), \tag{1}$$

where $P$ is a stochastic, indecomposable, aperiodic matrix, as discussed in [7]. Moreover, $x \in \mathbb{R}^n$ is an aggregated state vector, with each component $x_i$ representing a scalar state associated with agent $i = 1, \ldots, n$.

We model the network as an undirected graph $\mathcal{G} = V \times E$, with $V$ being a set of vertices $V = \{1, \ldots, n\}$ that represent the agents, and where the edge set $E \subseteq V \times V$ encodes the network topology in that $(i, j) \in E$ if and only if agents $i$ and $j$ can share information. The graph can be encoded through its adjacency matrix $A$ , i.e., a $n \times n$ matrix such that $a_{i,j} = 1$ if and only if $(i, j) \in E$ and is 0 elsewhere.

Let $\mathcal{N}_i \subset V$ be the set of vertices adjacent to vertex $i$, and let $|\mathcal{N}_i|$ denote its cardinality. We can then define the degree matrix $\Delta$ as the diagonal matrix whose diagonal entries are $\Delta_{i,i} = |\mathcal{N}_i|$. Using these matrices, a standard, discrete time model of consensus networks is the one defined by $x(k+1) = (I - \epsilon\mathcal{L})x(k)$, where $I$ is the identity matrix, $\mathcal{L} = \Delta - A$ is the graph Laplacian of the graph $\mathcal{G}$, and $\epsilon > 0$ the sampling time. Under this dynamics, the matrix $P$ in Equation (1) becomes

$$P = I - \epsilon\mathcal{L}. \tag{2}$$

Following the notation in [7], we will refer to $P$ as a Perron matrix, and this is the particular choice of $P$-matrix that will be used throughout the paper.

For the developments in this paper, we will not necessarily assume that the network is static, i.e. edges may be removed or added, thus resulting in a change in network topology. As such, we will in fact let the network be represented by a time varying, undirected graph $\mathcal{G}(t) = (V, E(t))$, where the edge set is time dependent. This could be caused by communication failures, or by the movements of the

individual agents as they enter and leave each others' sensory ranges.

Now, as the adjacency and degree matrices are time dependent, the Laplacian will depend on time as well and rather than explicitly computing $\mathcal{L}(t)$, we assume that we have an enumeration of all possible graphs over $n$ agents. Relative to this enumeration, we can define $T$ as the index set of all connected graphs $T = \{i \mid \mathcal{G}_i = (V, E_i)\ is\ connected\}$.

In fact, we will assume that the graph that is currently encoding the network topology is connected, i.e., its index belongs to $T$, and the linear consensus dynamics that we will employ can thus be given for $k \geq 0$ by $x(k+1) = P_{i(k)}x(k)$ with $x(0) = x_0$, $i(k) \in T$, where $x_0$ is the initial state of the system.

Finally, note that this can be generalized to

$$P_{i(k)} = I - \epsilon D\mathcal{L}_{i(k)}, \tag{3}$$

where $D$ is a positive definite, diagonal matrix representing the speed (or gain) of each agent. The introduction of gains does not change anything significant in that $P_i$ in Equation (3) is still a stochastic, indecomposable, aperiodic matrix for any sufficiently small, positive $\epsilon$. In fact, any $0 \leq \epsilon < \min_j 1/(d_j(n-1))$, where $d_j$ is the $j^{th}$ diagonal entry of $D$, is a valid choice.

## III. Invariant Motions

The consensus equation has been successfully applied in a number of applications where networked agents have to agree on some state value. Notable examples of this include the rendezvous problem, in which a collection of mobile agents are to meet at a common location, and the distributed average problem in sensor networks. However, in a number of situations, the motion that the agents can execute may be constrained, rendering a pure consensus-based control law infeasible. In this paper, we address this problem by endowing the agents with additional freedom in that their controllers need not necessarily be pure consensus-based. This additional control freedom must not, however, interfere with the global objective, which in this paper is taken to mean to reach agreement.

In order to ensure that agreement is still achieved, it is necessary that the new control signals are chosen in such a way that, after the execution, they do not corrupt the information needed to solve the original agreement problem. In this section, we will make these rather informal observations concrete by proposing a class of such control signals that we will refer to as *Invariant Motions*.

To study the evolution of such systems we need to point out the connection between this formulation of the consensus dynamics and that of discrete time Markov chains. In fact, one can think of $P_i$ as being the transition matrix in a Markov chain, with a corresponding, unique stationary distribution $\pi_i$ such that $\lim_{k\to\infty} P_i^k = \mathbf{1}\pi_i^T$, where $\mathbf{1}$ is the vector with ones in each entry.

The following result can then be directly obtained:

***Lemma 1:*** [14] The stationary distribution of

$$\mathcal{P}(t) = \prod_{k=0}^{t-1} P_{i(k)},$$

for any $t \geq 1$, where $P_{i(k)} = I - \epsilon D\mathcal{L}_{i(k)}$, $i(k) \in T$, is

$$\pi = D^{-1}\mathbf{1}\alpha,$$

with $\alpha$ being a normalizing scalar such that $\sum_{j=1}^{n} \pi_j = 1$. ■

In order to allow for the agents to exert a control action different from the consensus-based maneuver, we assume that the network behavior can be described by:

$$x(k+1) = P_{i(k)}x(k) + Bu(k) \qquad x(0) = x_0,\ i(k) \in T. \tag{4}$$

Here $B$ is an $n \times n$ matrix (typically the identity matrix), $u \in \mathbb{R}^n$ is a vector of inputs whose $i^{th}$ component $u_i$ is the scalar input exerted by agent $i$.

***Lemma 2:*** [14] Given the network dynamics in Equation (4). If $\sum_{k=0}^{t-1} u(k) = 0$, then $\pi^T x(t) = \pi^T x(0)$, where $\pi$ is the stationary distribution in Lemma 1.[1] ■

The previous lemma can be understood in the context of the partial difference equation analogy with the heat equation [17]. Any agent applying an input can be seen as an agent that is "warming up" or "cooling down" the network, depending on the sign of the input. Since the system is conservative (no heat can flow away), in order to recover the initial thermal equilibrium point the only information needed is how much heat has flown in or out from the network. This quantity corresponds to the integral of the applied input. Hence, if the integral is zero, the total heat present in the network has been preserved, and the initial, thermal equilibrium point will be reached under the regular evolution of the heat equation.

## IV. Constraints and Motion Feasibility

We have now seen that the definition of an invariant motion is one in which the sum over all the inputs (as a function of time) is zero. An alternative way in which one can think about this is through a storage state variable $z_i,\ i = 1, \ldots, n$, given by $z_i(t+1) = z_i(t) - u_i(t)$ with $z_i(0) = 0$. Clearly $z_i(t) = -\sum_{k=0}^{t-1} u_i(k)$.

The introduction of storage state variables implies that for a control strategy to correspond to an invariant motion, it has to drive $z$ to the origin. (Here $z = (z_1, \ldots, z_n)^T$.) The overall dynamics now becomes

$$\begin{cases} x(t+1) = Px(t) + u(t), \\ z(t+1) = z(t) - u(t), \\ x(0) = x_0,\ z(0) = 0. \end{cases} \tag{5}$$

[1] If the $n$ agents are moving in a $m-$dimensional space, the extension to $\mathbb{R}^{n\times m}$ requires the presence of a relative inertial reference for each agent. This means that the assumption that $\sum_{k=0}^{t-1} u(k) = 0$ needs to hold for $u \in \mathbb{R}^{n\times m}$.

***Theorem** 3:* A system governed by the dynamics in Equation (5) evolves on the hyperplane

$$\{(x,z) \in \mathbb{R}^{2n} \mid \pi^T x + \pi^T z = \pi^T x(0)\}.$$

*Proof:*

Applying lemma 1, at time $t$ the system (5) evolves from $x(0) = x_0$ and $z(0) = 0$ to

$$\begin{cases} x(t) = \prod_{k=0}^{t-1} P_{i(k)} x(0) + \sum_{k=0}^{t-1} \prod_{j=0}^{k} P_{i(j)} u(k) \\ z(t) = -\sum_{k=0}^{t-1} u(k). \end{cases}$$

Multiplying by $\pi^T$ on both sides of the above equation and observing that $\pi$ is the stationary distribution of $P_i \ \forall i \in T$, we get $\pi^T x(t) = \pi^T x(0) + \pi^T \sum_{k=0}^{t-1} u(k)$. Hence

$$\pi^T x(t) + \pi^T z(t) = \pi^T x(0)$$

■

As a consequence of Theorem 3, we can in fact think of the average of $z(k)$ as specifying how far from the initial average, the average of $x$ has drifted in $k$ steps. As such, one could for instance be interested in the problem of first selecting $u$ in such a way that a particular task is executed, and then driving $z$ back to 0 to preserve the initial centroid of the system. The reason why this strategy might be of importance is, for example, if additional constraints are imposed on the system dynamics, such as preserving network connectivity.

To address this type of problem formulation, we classify the possible constraints into three types. The structure of these constraints is *decentralized*, they involve only a sets of neighbors directly connected to each other:

*1) Kinematic constraints:* Each agent being a dynamical system has several constraint such as finite maximum acceleration, maximum speed and so on. In our simplified model for the agents dynamics we give an example of kinematic constraints by requiring that the maximum displacement in one time step for any agent $i$ is limited, i.e.

$$\|x_i(k+1) - x_i(k)\| \leq \beta. \tag{6}$$

This bounds the speed of the agents.

*2) Network constraints:* These constraints can be thought as constraints involving the graph representing the network. We give an example of network constraints by asking the network to stay always connected. To ensure such property we apply a stronger constraint by requiring that no couple of agents sharing an edge may lose connection by increasing their distance above a certain threshold, i.e

$$\|x_i(k+1) - x_j(k+1)\| \leq \alpha$$

where $\alpha$ is the sensing radius.

Here we stated a constraint that requires information about the future state of the neighbors. Such constraint can be ensured in a more conservative way asking that:

$$\begin{aligned} \|x_i(k+1) - x_j(k)\| &\leq \|x_i(k) - x_j(k)\| \\ \|x_i(k) - x_j(k+1)\| &\leq \|x_i(k) - x_j(k)\|, \end{aligned} \tag{7}$$

In other words, if both agents do not move in directions that may increase their relative distance regarding the neighbor as standing still then we ensure that at the next step their distance will surely be less than the threshold. This constraint does not need to be always active, in the case study of the next section we will design a rule that will render this constraint active only when strictly necessary.

*3) Mission constraints:* Mission constraints are the more various and hard to model since are most of the times qualitative in nature. Simple examples of mission constraints are:

1) Rendezvous at the network centroid.
2) Move the centroid of the network to a pre-specified location.
3) Surveillance over an area while preserving connectivity of the network and the initial centroid.
4) Avoid moving obstacles while preserving network properties as connectivity and initial centroid.

All of these mission constraints can be addressed within our framework of constrained invariant motions. In particular since our system evolves on the hyperplane $\pi^T x(t) + \pi^T z(t) = \pi^T x(0)$, we can put constraints on the increased state space to achieve for instance that the centroid at a particular instant of time is equal to the initial centroid by requiring that $\pi^T z(t) = 0$.

Following this, we will set up a general constrained optimization problem that will take the form of a decentralized one step-horizon optimization. The purpose of such control scheme is to keep the best features of the Laplacian feedback such as its average preserving property or its robustness to network topology changes while being able to address kinematic limitations of the agents and network constraints such as connectivity. In particular, if the particular decentralized model predictive control algorithm used ensures that the dynamics of the storage variables $z(k)$ is asymptotically stable, then we ensure the average preserving property of the Laplacian feedback regardless of the constraints.

## V. Constrained Consensus

We present here an algorithm for connectivity preserving rendezvous of a network of agents with centroid preserving motions and kinematic constraints. Our working assumptions are:

(A1) Each agent can sense all the neighboring agents up to a distance $k_r$, i.e., if the distance between an agent $i$ and an agent $j$ is less than or equal to $k_r$, there exists an edge in the associated proximity graph;

(A2) Proximity graph representing the network is connected at time $t = 0$;

(A3) All agents have the same speed, i.e., in equation (3) we assume $D = I$. This implies that the stationary distribution in $Lemma\ 1$ is $\pi = \mathbf{1}$, where $\mathbf{1}$ is a vector of ones.

In our model the agents know the following parameters:

- $\epsilon$ - the sampling time.
- $\beta$ - the agents' maximum displacement allowed in one time step.
- $\gamma$ - a bound on $|u_i(k)|$, in our simulations we choose $\gamma = \frac{\beta}{2}$.
- $k_r$ - the sensing radius.
- $\alpha$ - the maximum distance at which a neighbor is considered "sufficiently close" such that no connectivity preserving constraint is applied. $\alpha$ is constrained to be at least $\alpha \leq k_r - 2\beta$.
- $B = \frac{\beta\sqrt{n}}{4\epsilon(n-1)}$, the dimensions of a ball where the agents' are considered sufficiently close.

***Algorithm** 1 (Constrained Consensus Algorithm):*

1) Let $t = 0$.
2) Until all agents performed rendezvous somewhere (within a certain ball $B$), do:
   Each agent solves the following optimization problem:

$$\begin{aligned} &\min \|u_i(t)\|_2^2 \\ &\text{s.t.} \\ (a)\quad & x_i(t+1) = P[i,\cdot](t)\cdot x(t) + u_i(t) \\ (b)\quad & z_i(t+1) = z(t) - u_i(t) \\ (c)\quad & \|x_i(t+1) - x_i(t)\|_2 \leq \beta \\ & \quad \text{if } P(i,j) \neq 0 \text{ and } \|x_i(t) - x_j(t)\|_2 \leq \alpha \\ (d)\quad & \quad \|x_i(t+1) - x_j(t)\|_2 \leq \alpha \\ & \quad \text{elseif } P(i,j) \neq 0 \text{ and } \|x_i(t) - x_j(t)\|_2 \geq \alpha \\ (e)\quad & \quad \|x_i(t+1) - x_j(t)\|_2 \leq \|x_i(t) - x_j(t)\|_2 \end{aligned} \tag{8}$$

   Let $t = t + 1$.
3) Each agent solves the following optimization problem:

$$\begin{aligned} &\min \|z_i(t+1)\|_2^2 \\ &\text{s.t.} \\ (a)\quad & x_i(t+1) = P[i,\cdot](t)\cdot x(t) + u_i(t) \\ (b)\quad & z_i(t+1) = z(t) - u_i(t) \\ (c)\quad & \|u_i(t)\| \leq \gamma. \end{aligned} \tag{9}$$

   Let $t = t + 1$.

To solve this application we proposed an optimization problem solved by each agent using only local information (the distances between it and the neighbors) without the use of any kind of communication. The proposed optimization problem first minimizes the control input $u_i(k)$ such that the motions are feasible (Step 2 of algorithm 1), once all the constraints are no longer active it minimizes the 2-norm of the corresponding $z_i$ variable trying to bring it to zero (Step 3 of algorithm 1). According to Theorem 3 if all $z's$ are null then network centroid is equal to the initial centroid. Therefore, we chose as performance index of the optimization problem to solve at Step 3

$$\min \|z_i(t+1)\|_2^2$$

Regarding the constraints, using the classification in previous section we have:

1) *Kinematic constraints:* constraint (8.c). The motion distance of the agents in one step is bounded, this is ensured by (6). Without loosing of generality, we assume the same $\beta$ for all agents.
2) *Network constraints:* constraints (8.d), (8.e). The graph should remain connected, consequently some constraints (7) are be imposed for each agent. The number of such constraints depends on the number of neighbors and on the relative distance between them. We identify two types of neighbors: (a) agents with a relative distance less than $\alpha$, where $\alpha < k_r$ and, (b) agents with the relative distance between $\alpha$ and $k_r$. We say that the agents (b) are in the *critical region* and the relative distance cannot be increased at the next step to avoid the losing of connectivity. Using a constraint of type (7) we ensure that this distance is not increased. For agents of type (a) such a constrained is not active because $\alpha$ is chosen such that any input we apply to both agents, at the next step the connection between them is not loosed, i.e., the relative distance will be less than $k_r$.
3) *Mission constraints:*. We want to rendezvous at the initial network centroid.

We now prove that algorithm 1 solves the Constrained Consensus Problem.

***Theorem** 4:* There exists a suitable choice of parameters $\alpha, \beta, \epsilon, \gamma, B, k_r$ such that algorithm 1 makes the agents rendezvous at the initial centroid of the network.

*Proof:*

The proof is constructive, i.e., we show how to determine suitable values of the parameters that ensures convergence of algorithm 1. We divide it into two steps: in the first step the agents perform the first optimization problem, as a result we prove convergence to a point inside the convex hull defined by the initial position of the agents. At the end of this step the storage variables $z(k)$ have an arbitrary value function of the trajectories adopted by the agents. In the second step we prove that performing the second optimization problem the agents are able to bring the value of their storage variables to zero monotonically while being sure to follow feasible trajectories, in such a way the centroid of the network is brought again to the initial position. From this point on the network will evolve as a standard consensus network, thus asymptotically converge to its centroid.

- *First step*: We now show that during the first part of the algorithm $V(k) = \|\delta(k)\|_\infty$ with

$$\delta(k) = x(k) - \frac{\mathbf{1}\mathbf{1}^T}{n} x(k) \tag{10}$$

  is a monotonically decreasing function.
  The agents whose $\delta_i(k)$ is maximum are at the border of the polytope spanned by the agents. If an agent sees the neighbors in a convex hull, then, given the nature of the constraints, it will move in a direction inside such convex hull thus reducing $\delta_i(k)$. In particular since at each instant of time we have that each bordering agent will move in a direction inside the convex hull of his neighbors we have that

$$V(k+1) = \|\delta(k+1)\|_\infty < \|\delta(k)\|_\infty = V(k)$$

Thus proving the statement.

- *Second step*: When the agents solve the second optimization problem, i.e., step 3 of Algorithm 1, we prove that defining an appropriate constant $\gamma$ such that $\|u(k)\|_\infty < \gamma$ and a constant $B$, we have that if $\|\delta(k)\|_2 < B$, then $\|\delta(k+1)\|_2 < B$, where $\delta(k)$ is given in (10). In other words if the agents are sufficiently close and their inputs are sufficiently small, the agents are able to apply whatever input they need to bring their storage variables to zero while staying bounded in a region where no constraint is ever active. We start by working out the dynamics of $\delta(k)$. We have:

$$\delta(k+1) = Px(k) + u(k) - \frac{\mathbf{1}\mathbf{1}^T}{n}(Px(k) + u(k))$$

Since $P$ is doubly stochastic, $\mathbf{1}^T\delta(k) = 0$ and $\frac{\mathbf{1}\mathbf{1}^T}{n}^2 = \frac{\mathbf{1}\mathbf{1}^T}{n}$:

$$\delta(k+1) = (P - \frac{\mathbf{1}\mathbf{1}^T}{n})\delta(k) + (I - \frac{\mathbf{1}\mathbf{1}^T}{n})u(k)).$$

Taking the 2-norm of both arguments, for the triangular inequality:

$$\|\delta(k+1)\|_2 \leq \|(P - \frac{\mathbf{1}\mathbf{1}^T}{n})\delta(k)\|_2 + \|(I - \frac{\mathbf{1}\mathbf{1}^T}{n})u(k)\|_2.$$

The 2-norm is an induced norm, for a symmetric matrix it is equivalent to its spectral radius, therefore:

$$\|(I - \frac{\mathbf{1}\mathbf{1}^T}{n})\|_2 = \rho(I - \frac{\mathbf{1}\mathbf{1}^T}{n}) = 1,$$

thus $\|\delta(k+1)\|_2 \leq \|(P - \frac{\mathbf{1}\mathbf{1}^T}{n})\delta(k)\|_2 + \|u(k)\|_2$, and finally

$$\|\delta(k+1)\|_2 \leq \rho(P - \frac{\mathbf{1}\mathbf{1}^T}{n})\|\delta(k)\|_2 + \|u(k)\|_2.$$

The matrix $A = P - \frac{\mathbf{1}\mathbf{1}^T}{n}$ is contractive since its spectral radius $\rho(A) < 1$, [15]. Then

$$\|\delta(k+1)\|_2 \leq \rho(A)\|\delta(k)\|_2 + \|u(k)\|_2.$$

We can bound $\|u(k)\|_2$ with $\sqrt{n}\|u(k)\|_\infty$ where $n$ is the number of agents. Therefore:

$$\|\delta(k+1)\|_2 \leq \rho(A)\|\delta(k)\|_2 + \sqrt{n}\|u(k)\|_\infty,$$

Multiplying and dividing by $\|\delta(k)\|_2$:

$$\|\delta(k+1)\|_2 \leq (\rho(A) + \frac{\sqrt{n}\|u(k)\|_\infty}{\|\delta(k)\|_2})\|\delta(k)\|_2.$$

So for $\rho(A) + \frac{\sqrt{n}\|u(k)\|_\infty}{\|\delta(k)\|_2} < 1$,

$$\|u(k)\|_\infty < (1 - \rho(A))\frac{\|\delta(k)\|_2}{\sqrt{n}},$$

we have that $\|\delta(k+1)\|_2 < \|\delta(k)\|_2$.
This proves that if $\gamma$ is small respect to $\|\delta(k)\|_2$ we have a contraction of $\|\delta(k)\|_2$.

We now give an upper bound to $\|\delta(k)\|_2$ such that if $\|\delta(k)\|_2 < B$ and $\|u(k)\|_\infty < \gamma$ no constraint is active. The first constraint that we consider is the connectivity preserving constraint. It is straightforward to see that if each agent is in a ball of diameter $\frac{K_r}{2}$ each agent will be able to see each other and no connectivity constraint will be active. The constraint that we need to take care of is the kinematic constraint, namely:

$$\|x_i(k+1) - x_i(k)\|_2 < \beta,$$

because we assume that $\beta << K_r$. We recall the update rule for the generic agent:

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in N_i} (x_j(k) - x_i(k) + u_i(k),$$

Adding the constraint on the maximum speed of the agents we have that

$$\|\epsilon \sum_{j \in N_i} (x_j(k) - x_i(k)) + u_i(k)\|_2 < \beta,$$

since $\|u_i(k)\|_\infty < \gamma$ and defining

$$\Gamma = \max_{j,i} \|x_j - x_i\|_\infty \geq 2\|\delta(k)\|_\infty \geq \frac{2\|\delta(k)\|_2}{\sqrt{n}}$$

we obtain $\gamma < \beta - \epsilon(n-1)\Gamma$. Giving a reasonable bound on $\Gamma$ such that $\gamma$ can be comparable with $\beta$, we choose $\Gamma = \frac{\beta}{2\epsilon(n-1)}$. In such way choosing $\gamma = \frac{\beta}{2}$ we have that

$$\|\delta(k)\|_2 < \frac{\Gamma\sqrt{n}}{2} = \frac{\beta\sqrt{n}}{4\epsilon(n-1)} = B$$

So if $\|\delta(k)\|_2 < B$ also $\|\delta(k+1)\|_2 < B$.
In such a way the agents will apply the appropriate $\|u_i(k)\|_\infty < \gamma$ to bring $z(k)$ monotonically to zero while keeping the network bounded in a ball where no constraint is ever active (the agents are sufficiently close between each other). When $z(k)$ is equal to zero the system will evolve with $u(k) = 0$ as a normal consensus network (and we proved that the consensus dynamic will not violate any constraint) and so

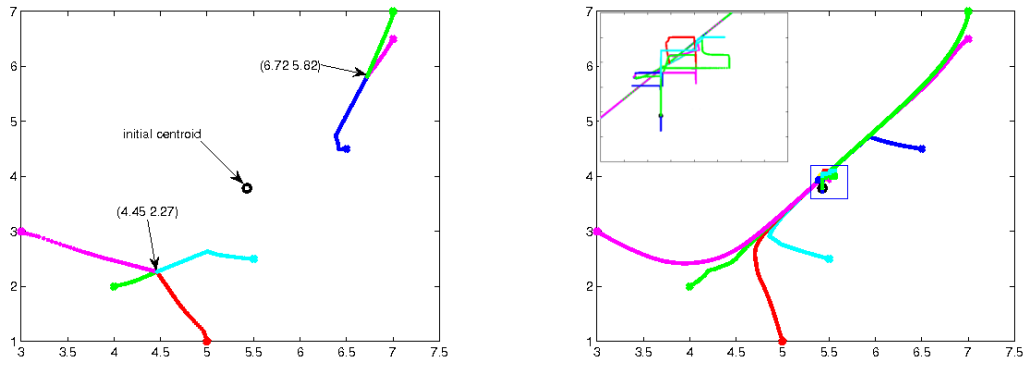
$$\lim_{k \to \infty} x(k) = \frac{\mathbf{1}\mathbf{1}^T}{n}x(0).$$

■

## VI. Simulations

We initialize a network of seven agents and assign them the task of rendezvous at the initial network centroid while ensuring connectivity of the network and satisfying the kinematic constraints of the agents that are assumed to have a single integrator dynamic with a finite speed. For each agent we solve the online optimization problems of Algorithm 1 with the following parameters:

- Sampling time: $0.001\ sec$
- Finite speed: $\|x_i(t+1) - x_i(t)\|_2 \leq \beta = 0.001\ m\ (1\ m/sec)$.




(a) Network of seven agents performing rendezvous using the standard consensus dynamic (1)

(b) Network of seven agents performing rendezvous using Alg. 1 after step (3)

Fig. 1. A comparison between the basic consensus dynamic for coordination of multi-agent systems and the constraint invariant motions.

- Sensing radius: $\|x_i(t+1) - x_j(t)\|_2 \leq \alpha = 2.5\ m$.
- Sensing radius critical margin: $k_c = 0.5\ m$.

In Figure 1(a) is shown the evolution of the network of seven agents denoted $a_1$ to $a_7$ starting from the initial positions $\{(3\ 3),\ (4\ 2),\ (5\ 1),\ (6.5\ 4.5),\ (5.5\ 2.5),\ (7\ 6.5),\ (7\ 7)\}$ with the standard consensus dynamic using a proximity graph. As can be observed, the initial centroid shown as a small circle at $(5.43\ 3.79)$ is not reached. Figure 1(b) illustrates the evolution of the same network of agents, starting from the same initial positions, but now applying Alg. 1: in figure 1(b) it can be seen that the initial centroid is reached after step (3) regardless of the constraints.

From the comparison between Figure 1(a) and Figure 1(b) we show how the use of invariant constraint motions allows the network of agents to exploit the properties of the consensus dynamic while retaining the necessary freedom to perform real world task. It is relevant that such approach can be used to split up in two levels the controllers needed to perform coordination of multi-agent systems, i.e. the high level controllers make use of the constraint invariant motions to perform tasks while being sure to provide feasible problems to solve to the lower level controllers.

## VII. Conclusion

In this paper we presented a general tool, the Invariant Motions, to deal with constrained consensus problems while ensuring the average preserving properties of Laplacian-based controllers. In particular we have presented an algorithm for multi-agent networked systems that, using such Invariant Motions, achieves consensus (rendezvous) on the average of the initial states while ensuring the connectivity of the network and feasible trajectories for the agents.

## References

[1] J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in $d$ dimensions. *IEEE Trans. Robot. Automat.*, 51(8): 1289-1298, 2006.

[2] G. Ferrari-Trecate, A. Buffa, and M. Gati. Analysis of coordination in multi-agent systems through partial difference equations. Part I: The Laplacian control. *16th IFAC World Congress on Automatic Control*, 2005.

[3] G. Ferrari-Trecate, M. Egerstedt, A. Buffa, and M. Ji. Laplacian Sheep: A Hybrid, Stop-Go Policy for Leader-Based Containment Control. *Hybrid Systems: Computation and Control*, Springer-Verlag, pp. 212-226, Santa Barbara, CA, March 2006.

[4] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Automat. Contr.*, 48(6):988-1001, June 2003.

[5] Z. Lin, M. Broucke, and B. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Trans. Automat. Contr.*, 49(4):622-629, 2004.

[6] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Automat. Contr.*, 51(3):401-420, 2006.

[7] R. Olfati-Saber. Consensus and cooperation in networked multi-agent systems. *IEEE Proceedings.*, 95(1):215, Jannuary 2007.

[8] W. Ren, R.W. Beard, and E. Atkins. A Survey of Consensus Problems in Multi-agent Coordination. In Proceedings of *American Control Conference*, Portland, OR, 2005.

[9] K. Sugihara and I. Suzuki. Distributed motion coordination of multiple robots. In Proceedings of *IEEE Int. Symp. on Intelligent Control*, pages 138-143, 1990.

[10] H. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents, part II : Dynamic topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2016-2021, 2003.

[11] R. Olfati-Saber, R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, volume 49, pages 1520–1533, 2004.

[12] G. Ferrari-Trecate, L. Galbusera, M.P.E. Marciandi, and R. Scattolini. Contractive distributed mpc for consensus in networks of single- and double-integrators. 17th IFAC World Congress on Automatic Control, 2008

[13] Dimos V. Dimarogonas and Karl H. Johansson, "Decentralized Connectivity Maintenance in Mobile Networks with Bounded Inputs", 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, pp. 1507-1512, May 2008

[14] M. Franceschelli, M. Egerstedt, and A. Giua. Motion Probes for Fault Detection and Recovery in Networked Control Systems. In Proceedings of *American Control Conference*, Seattle, WA, June 2008.

[15] H. Liu, M. Lu, F. TianOn. "On the Laplacian spectral radius of a graph". Linear Algebra and Its Applications, 2004.

[16] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," Systems and Control Letters, vol. 53, pp. 65-78, 2004.

[17] Q. Hui and W. M. Haddad, "Distributed nonlinear control algorithms for network consensus," Automatica, vol. 44, no. 9, pp. 2375-2381, 2008.