Navigation Function Based Decentralized Control of A Multi-Agent System with Network Connectivity Constraints

Zhen Kan, John M. Shea, and Warren E. Dixon   [1,2]

Abstract. A wide range of applications require or can benefit from collaborative behavior of a group of agents. The technical challenge addressed in this chapter is the development of a decentralized control strategy that enables each agent to independently navigate to ensure agents achieve a collective goal while the system maintains network connectivity. Specifically, cooperative controllers are developed for networked agents that have both limited sensing and network connectivity constraints. By modeling the interaction among the agents as a graph, several different approaches to address the problems of preserving network connectivity are presented, with the focus on a method that utilizes navigation function frameworks. By modeling network connectivity constraints as artificial obstacles in navigation functions, a decentralized control strategy is presented in two particular applications, formation control and rendezvous for a system of autonomous agents, which ensures global convergence to the unique minimum of the potential field (i.e., desired formation or desired destination) while preserving network connectivity. Simulation results are provided to demonstrate the developed strategy.

Keywords. Network Connectivity, Navigation Function, Formation Control, Rendezvous, Decentralized Control

1. Introduction

Multi-agent systems under cooperative control provide versatile platforms for various commercial and military applications, such as formation flight and cooperative attack in military systems [1], environmental sampling and distributed aperture observing for mobile sensor networks [2], and intelligent highways and air traffic control in transportation systems [3]. These types of tasks usually require or can benefit from collaborative motion of a group of agents, and thus the agents must be able to exchange information over some form of communications network. For most applications, communications will be over a wireless network, in which the communication links between agents are dependent on

the propagation of electromagnetic signals between the agents, and the electromagnetic power density decreases with distance. When performing desired tasks, the underlying wireless communication can be impacted due to the motion of agents. If the network is partitioned and the inter-agent communication is disconnected, the agents can no longer coordinate their movements, and the mission may fail. Hence, control algorithms must be designed under the constraint of preserving network connectivity when performing desired tasks.

1.1. Overview of Research on Maintenance of Network Connectivity

Network connectivity is a mainstream research focus. The interaction of agents is typically modeled using constructs from graph theory, and the graph determines which agents can exchange and share information and how robust the group can behave in a dynamic environment. Proximity-based graphs are generally used to capture the inter-agent communication. In particular, a time varying graph $\mathscr{G}(t)$ is used to model the dynamic graph, where $\mathscr{V}$ is the set of vertices (representing the agents) and $\mathscr{E}(t)$ is the set of edges connecting the vertices in $\mathscr{V}$. Each edge connecting node x and y in $\mathscr{E}(t)$ specifies an available communication link.

A metric that is typically used to capture network connectivity is the second smallest eigenvalue $\lambda_2(\mathscr{L})$ of the Laplacian matrix $\mathscr{L}$ of the graph $\mathscr{G}$, which is also known as the Fiedler value [4]. A positive $\lambda_2(\mathscr{L})$ indicates a connected graph, and the associated eigenvector can be used to determine a set of links that if removed will cause the network to partition [5]. To ensure network connectivity, optimization based approaches are developed in the works of [6] and [7] to maximize the Fiedler value. However, the computation of $\lambda_2(\mathscr{L})$ is generally centralized due to the requirement of the knowledge of the entire network structure. Moreover, $\lambda_2(\mathscr{L})$ is a non-differentiable function of the Laplacian matrix $\mathscr{L}$, which presents an obstacle for designing continuous feedback controllers. Alternative ways to overcome this constraint is to use the determinant of $\mathscr{L}$ [8], which is a differentiable function of $\mathscr{L}$, or achieving consensus on Laplacian eigenvectors [9].

Since the edge connection information is collected into the adjacency matrix $A(\mathscr{G})$, network connectivity can be captured by the sum of powers of the adjacency matrix $\sum_{k=0}^{K} A^k$, which represents the number of paths up to length K connecting two nodes in the graph $\mathscr{G}$ [4]. If every entry in $\sum_{k=0}^{K} A^k$ is positive, any two nodes in the graph $\mathscr{G}$ are connected with a path of maximum length K. Following this idea, centralized optimization-based controllers are developed in the work of [10] and [11] to maintain the positiveness of all entries in $\sum_{k=0}^{K} A^k$. Discrete-time approaches are discussed in [12–15] which rely on local gradients and switching of graphs in the case of edge addition. This class of approaches are typically hybrid, since both continuous edge preservation and discrete topology control are considered.

Artificial potential fields based approaches that use attractive and repulsive potentials are also widely utilized to guide the movement of autonomous systems while preserving network connectivity. Particularly, attractive potential fields are centered at the goal locations, and repulsive potential fields are generated around obstacles. Driven by the negative gradient of the potential field, each mobile robot will converge to a minimum of the potential field, which is typically the desired final position. Modeling network connectivity as an artificial constraint, results such as [8, 15–22] are motivated by

the need to prevent the graph from partitioning using artificial potential fields. A potential field based centralized control approach is developed in [8] and [20] to ensure the connectivity of a group of agents using the graph Laplacian matrix. In [16], connectivity control is performed in the discrete space of graphs to verify link deletions with respect to connectivity, and motion control is performed in the continuous configuration space using a potential field. A potential field-based neighbor control law is designed in [17] to achieve velocity alignment and network connectivity among different topologies. In [15] and [19], a repulsive potential is used for a collision avoidance objective, and an attractive potential field is used to drive agents together. Distributed control laws are investigated to ensure edge maintenance in [22] by allowing unbounded potential force whenever pairs of agents are about to break existing links. In [21], a potential field is designed for a group of mobile agents to perform desired tasks while maintaining network connectivity; however, it is unclear how the potential field method in [21] can be extended to include static obstacles. Other results that use artificial potential fields for networked agents to perform formation control, rendezvous, flocking and containment control while preserving network connectivity include [23–30].

1.2. Main Contributions

A common problem with the aforementioned artificial potential field-based control algorithms is the existence of local minima when attractive and repulsive force are combined. When trapped by local minima, the system will no longer converge to the desired minimum (i.e., control objective) and result in mission failure. To avoid local minima, a specific type of artificial potential, called a navigation function, achieves a unique minimum (cf. [31, 32]) and has been widely used in motion control for multi-agent systems. The navigation function developed in [32] is a real-valued function that is designed so that the negated gradient field does not have a local minima. The negated gradient of the navigation function is attracted towards the goal and repulsed by obstacles for almost all initial states. As such, closed-loop navigation function approaches guarantee convergence to a desired destination.

The development in this chapter utilizes ideas from navigation function frameworks to control a group of agents that have constraints on their sensing range and a network connectivity constraint. Each agent is assumed to have limited sensing capabilities or knowledge about the environment and limited communication capabilities with nearby agents. To show the effectiveness of the navigation function based approaches, two example applications, formation control and rendezvous, developed in our previous works of [33] and [34] are introduced. In comparison to the above artificial potential field-based results, the method developed in [33] achieves convergence to a desired configuration and maintenance of network connectivity using a decentralized navigation function approach that uses only local feedback information. By using a local range sensor, an advantageous feature of the developed decentralized controller is that no inter-agent communication is required (i.e., communication-free global decentralized group behavior). That is, the goal is to maintain connectivity so that radio communication is available when required for various task/mission scenarios, but communication is not required to navigate, enabling stealth modes of operation. In [34], a group of wheeled robots with nonholonomic constraints is tasked with the objective of rendezvousing at a common specified setpoint with a desired orientation while maintaining network connectivity. Only a subset of the

robots are assumed to be aware of the global destination, and the remaining robots must move with the constraint of ensuring network connectivity so that the informed robots can guide the group to the goal. Since the classical navigation function based approach in [34] is not applicable to robots with nonholonomic constraints, a decentralized time-varying continuous controller is developed to reach the desired destination with a desired orientation while preserving network connectivity based on a dipolar navigation function framework. Only local sensing feedback (i.e., relative position) from neighboring robots is used to navigate the group. Simulation results demonstrate the performance of the developed approaches.

2. Navigation Function Framework

A navigation function is a particular category of potential functions where the potential field does not have local minima and the negative gradient vector field of the potential field guarantees almost global convergence to a desired destination, along with (guaranteed) collision avoidance, if the initial conditions do not lie within the sets of measure zero. Formally, a navigation function is defined as:

Definition 1. [31] [32]Let $\mathscr{F} \subset E^n$ be a compact connected analytic manifold with a boundary. A map $\varphi : \mathscr{F} \rightarrow [0,1]$ is a Navigation Function, if it is: 1) smooth on $\mathscr{F}$ (at least a $\mathscr{C}^2$ function); 2) admissible on $\mathscr{F}$, (uniformly maximal on $\partial \mathscr{F}$ and constraint boundary); 3) polar on $\mathscr{F}$, (q_d is a unique minimum); and 4) a Morse function, (critical points of the navigation function are non-degenerate).

The second condition in Definition 1 establishes that the generated trajectories are collision-free, since the resulting vector field is transverse to the boundary of $\mathscr{F}$, while the third point indicates that, using a polar function on a compact connected manifold with a boundary, all initial conditions are either brought to a saddle point or to the unique minimum q_d. The requirement that the navigation function is a Morse function ensures that the initial conditions that bring the system to saddle points are sets of measure zero [32]. Given this property, all initial conditions not within sets of measure zero are brought to the unique minimum. An example of the generated artificial potential field is shown in Fig. 1 in which the destination is assigned a minimum potential value, and the obstacle is assigned a maximum potential value.

3. Applications

In this section, two results that are developed in our previous works of [33] and [34] are discussed. Based on the navigation function framework, a group of agents are controlled to perform cooperative tasks, such as formation control in [33] and rendezvous in [34], while preserving network connectivity.

3.1. Formation Control

3.1.1. Problem Formulation

Consider a network composed of N agents in the workspace $\mathscr{F}$, where agent i moves according to the following kinematics:
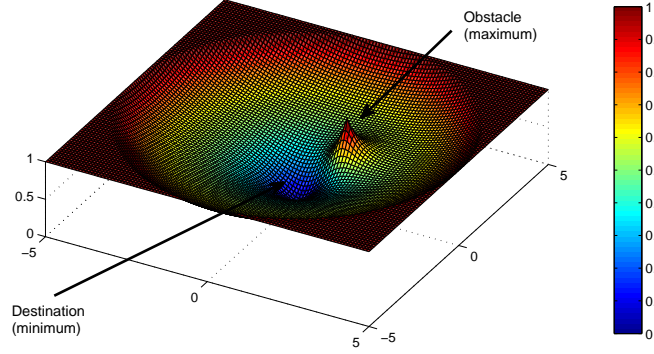
Figure 1. An example of the artificial potential field generated for a disk-shaped workspace with destination at the origin and an obstacle (i.e., artificial constraints) located at $[1,1]^T$.

$$\dot{q}_i = u_i, \ i = 1, \cdots, N \tag{1}$$

where $q_i \in \mathbb{R}^2$ denotes the position of agent i in a two dimensional (2D) plane, and $u_i \in \mathbb{R}^2$ denotes the velocity of agent i (i.e., the control input). The workspace $\mathscr{F}$ is assumed to be circular and bounded with radius R, and $\partial\mathscr{F}$ denotes the boundary of $\mathscr{F}$. Each agent in $\mathscr{F}$ is represented by a point-mass with a limited communication and sensing capability encoded by a disk area. Two moving agents can communicate with each other if they are within a distance R_c, while the agent can sense stationary obstacles or other agents within a distance R_s. For simplicity and without loss of generality, assume that the sensing area coincides with the communication area, i.e., $R_c = R_s$. A set of fixed points, $p_1, \cdots, p_M$, are defined to represent M stationary obstacles in the workspace $\mathscr{F}$, and the index set of obstacles is denoted as $\mathscr{M} = \{1, \cdots, M\}$.

The interaction of the system is modeled as a dynamic graph, in the sense that the system evolves in time governed by the agent kinematics in (1). The dynamic graph is denoted as $\mathscr{G}(t) = (\mathscr{V}, \mathscr{E}(t))$, where $\mathscr{V} = \{1, \cdots, N\}$ denotes the set of nodes, and $\mathscr{E}(t) = \{(i, j) \in \mathscr{V} \times \mathscr{V} | d_{ij} \leq R_c\}$ denotes the set of time varying edges, where node i and node j are located at a position q_i and q_j, and $d_{ij} \in \mathbb{R}^+$ is the relative distance defined as $d_{ij} = \|q_i - q_j\|$. In graph $\mathscr{G}(t)$, each node i represents an agent, and the edge (i, j) denotes a link between agent i and j when they stay within a distance R_c. The set of neighbors of node i (i.e., all the agents within the sensing zone of agent i) is given by $\mathscr{N}_i = \{j, j \neq i | j \in \mathscr{V}, (i, j) \in \mathscr{E}\}$. One objective in this work is to have the multi-agent system converge to a desired configuration, determined by a formation matrix $c_{ij} \in \mathbb{R}^2$ representing the desired relative position of node i with an adjacent node $j \in \mathscr{N}_i^f$, where $\mathscr{N}_i^f \subset \mathscr{N}_i$ denotes the set of nodes required to form a prespecified relative position with node i in the desired configuration. The neighborhood $\mathscr{N}_i$ is a time varying set since nodes may enter or leave the communication region of node i at any time instant, while $\mathscr{N}_i^f$ is a static set which is specified by the desired configuration. The desired position of node i, denoted by q_{di}, is defined as $q_{di} = \left\{ q_i | \|q_i - q_j - c_{ij}\|^2 = 0, j \in \mathscr{N}_i^f \right\}$. An edge (i, j) is only established between nodes i and j if $j \in \mathscr{N}_i^f$.

A collision region [3] is defined for each agent i as a small disk with radius $\delta_1 < R_c$ around the agent i, such that any other agent $j \in \mathcal{N}_i$, or obstacle p_k, $k \in \mathcal{M}$, inside this region is considered as a potential collision with agent i. To ensure connectivity, an escape region for each agent i is defined as the outer ring of the communication area with radius r, $R_c - \delta_2 < r < R_c$, where $\delta_2 \in \mathbb{R}$ is a predetermined buffer distance. Edges formed with any node $j \in \mathcal{N}_i^f$ in the escape region are in danger of breaking. The objective is to develop a decentralized controller u_i that uses relative position information from the range sensor to regulate a connected initial graph to a desired configuration while maintaining network connectivity and avoiding collisions with other agents and obstacles.

3.1.2. Control Design

A decentralized controller is developed using only local sensing to navigate the agents to a desired formation while maintaining network connectivity. Consider a decentralized navigation function candidate $\varphi_i : \mathscr{F}_i \to [0, 1]$ for each node i as

$$\varphi_i = \frac{\gamma_i}{(\gamma_i^\alpha + \beta_i)^{1/\alpha}}, \tag{2}$$

where $\alpha \in \mathbb{R}^+$ is a tuning parameter, $\gamma_i : \mathbb{R}^2 \to \mathbb{R}^+$ is the goal function, and $\beta_i : \mathbb{R}^2 \to [0, 1]$ is a constraint function for node i. The goal function γ_i in (2) encodes the control objective of node i, specified in terms of the desired relative position with respect to the adjacent nodes $\left\{ j \in \mathcal{N}_i^f \right\}$, and drives the system to a desired configuration[4]. The goal function is designed as

$$\gamma_i(q_i, q_j) = \sum_{j \in \mathcal{N}_i^f} \left\| q_i - q_j - c_{ij} \right\|^2. \tag{3}$$

The constraint function β_i in (2) is designed as

$$\beta_i = B_{i0} \prod_{j \in \mathcal{N}_i^f} b_{ij} \prod_{k \in \mathcal{N}_i \cup \mathcal{M}_i} B_{ik}, \tag{4}$$

to ensure collision avoidance and network connectivity by only accounting for nodes and obstacles located within its sensing area during each time instant. Specifically, the constraint function in (4) is designed to vanish whenever node i intersects with one of the constraints in the environment, (i.e., if node i touches a fixed obstacle, the workspace boundary, other nodes, or departs away from its adjacent nodes $\left\{ j \in \mathcal{N}_i^f \right\}$ to a distance of R_c).

In (4), $b_{ij} \triangleq b(q_i, q_j) : \mathbb{R}^2 \to [0, 1]$ ensures connectivity of the network graph (i.e., guarantees that nodes $\left\{ j \in \mathcal{N}_i^f \right\}$ will never leave the communication zone of node i if node j is initially connected to node i) and is designed as

---

[3]The potential collision for node i in this work not only refers to the fixed obstacles, but also other moving nodes or the workspace boundary, which are currently located in its collision region.

[4]The formation objective γ_i is developed based on the desire to control the distance and relative bearings between nodes. For some applications, only the relative distance between nodes is important, and the objective could be rewritten as $\gamma_i = \sum_{j \in \mathcal{N}_i^f} \left(\|q_i - q_j\| - \|c_{ij}\| \right)^2$; however, this objective can introduce redundant desired configurations. Future efforts could consider this alternative objective, where an approach such as [18] may be explored to address the multiple desired minima.

$$b_{ij} = \begin{cases} 1 & d_{ij} \leq R_c - \delta_2 \\ -\frac{1}{\delta_2^2}(d_{ij} + 2\delta_2 - R_c)^2 + \frac{2}{\delta_2}(d_{ij} + 2\delta_2 - R_c) & R_c - \delta_2 < d_{ij} < R_c \\ 0 & d_{ij} \geq R_c. \end{cases} \quad (5)$$

Also in (4), $B_{ik} \triangleq B(q_i, q_k) : \mathbb{R}^2 \to [0, 1]$, for point $k \in \mathcal{N}_i \cup \mathcal{M}_i$, where $\mathcal{M}_i$ indicates the set of obstacles within the sensing area of node i at each time instant, ensures that node i is repulsed from other nodes or obstacles to prevent a collision, and is designed as

$$B_{ik} = \begin{cases} -\frac{1}{\delta_1^2}d_{ik}^2 + \frac{2}{\delta_1}d_{ik} & d_{ik} < \delta_1 \\ 1 & d_{ik} \geq \delta_1. \end{cases} \quad (6)$$

Similarly, the function B_{i0} in (4) is used to model the potential collision of node i with the workspace boundary, where the positive scalar $B_{i0} \in \mathbb{R}$ is designed similar to B_{ik} by replacing d_{ik} with d_{i0}, where $d_{i0} \in \mathbb{R}^+$ is the relative distance of the node i to the workspace boundary defined as $d_{i0} = R - \|q_i\|$.

Based on the definition of the navigation function candidate, the decentralized controller for each node is designed as

$$u_i = -K\nabla_{q_i}\varphi_i, \quad (7)$$

where K is a positive gain, and $\nabla_{q_i}\varphi_i$ is the gradient of φ_i with respect to q_i. Hence, the controller in (7) is bounded and yields the desired performance by steering node i along the direction of the negative gradient of φ_i if (2) is a navigation function. Due to space limitation, the proof that (2) is a qualified navigation function is not included and can be referred to the work of [33].

3.1.3. Simulation Results

Simulation results illustrate the performance of the proposed control strategy. As shown in Fig. 2, a connected initial graph of 20 nodes with kinematics in (1) are randomly deployed with desired neighborhood in a workspace of $R = 10$ m with static obstacles. Each node is assumed to have a limited communication and sensing zone of $R_c = 2$ m. The squares and dots denote the moving agents and the static obstacles respectively, while the solid line connecting two nodes represents a communication link, indicating that the two agents are located within each other's communication and sensing zone. The desired configuration is characterized by a shape of "UF". The system is simulated for $50s$ with the step size of 0.1. The tuning parameter α in (2) is set as $\alpha = 1.5$, and $\delta_1 = \delta_2 = 0.4$ m in (5) and (6). Results in Fig. 3 indicate that the system finally converges to the desired configuration. Fig. 4 shows that the inter-node distance between nodes converges to the desired value. To show the connectivity of the network during the evolution, the Fiedler eigenvalue of the graph Laplacian matrix is plotted in Fig. 5. Since the Fiedler eigenvalue is always positive, the graph is connected [4].

3.2. Rendezvous for Mobile Agents with Nonholonomic Constraints

3.2.1. Problem Formulation

Consider N networked mobile robots operating in a workspace $\mathcal{F}$, where $\mathcal{F}$ is a bounded disk area with radius R_w. Each robot in $\mathcal{F}$ moves according to the following nonholonomic kinematics:
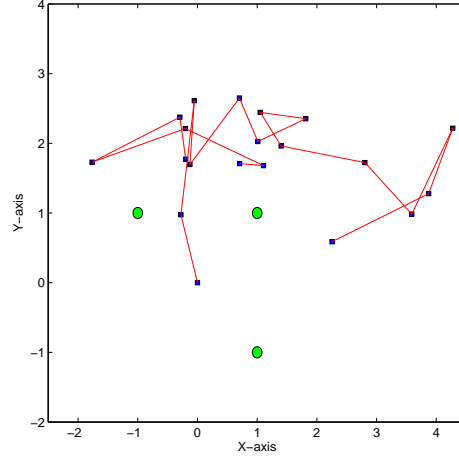
Figure 2. A connected initial graph with desired neighborhood in the workspace with static obstacles, where dots represent the static obstacles, squares represent agents, and the line connecting the nodes indicate the available communication between nodes.
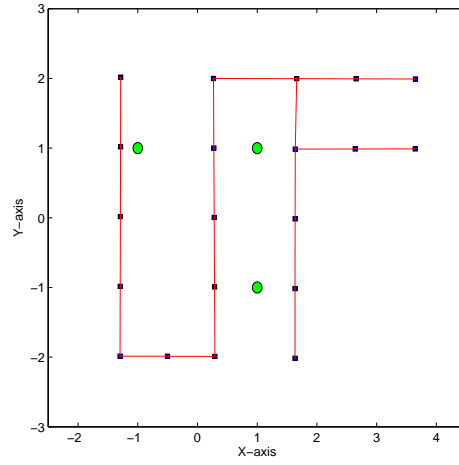

Figure 3. The achieved final configuration.

$$\dot{q}_i = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i(t) \\ \omega_i(t) \end{bmatrix}, \ i = 1, \cdots, N \tag{8}$$

where $q_i(t) \triangleq \begin{bmatrix} p_i^T(t) \ \theta_i(t) \end{bmatrix}^T \in \mathbb{R}^3$ denotes the states of robot i, with $p_i(t) \triangleq \begin{bmatrix} x_i(t) \ y_i(t) \end{bmatrix}^T \in \mathbb{R}^2$ denoting the position of robot i, and $\theta_i(t) \in (-\pi, \pi]$ denoting the robot orientation with respect to the global coordinate frame in $\mathscr{F}$. In (8), $v_i(t)$, $\omega_i(t) \in \mathbb{R}$ are the control inputs that represent the linear and angular velocity of robot i,
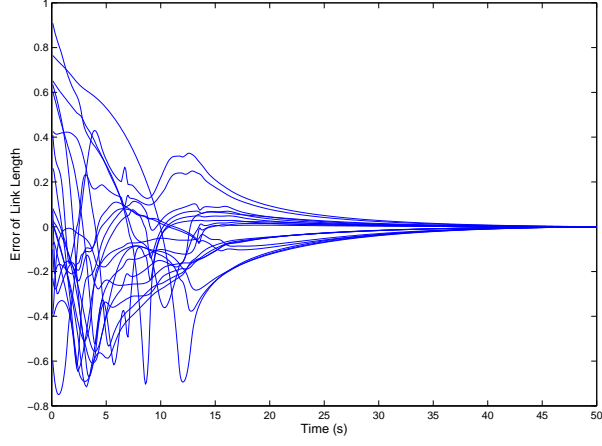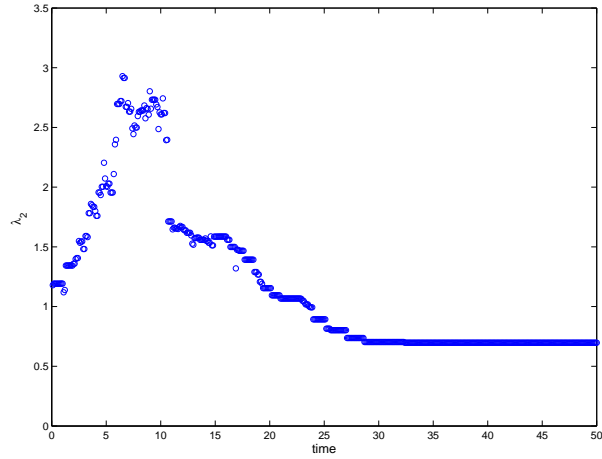
Figure 4. The error plot.

Figure 5. The plot of the Fiedler eigenvalue of the Laplacian matrix during the evolution. The circle indicates the Fiedler eigenvalue of the graph at each time instance.

respectively.

Assume that each robot has sensing and communication limitations encoded by a disk area with radius R, which indicates that two moving robots can sense and communicate with each other as long as they stay within a distance of R. We also assume that only a subset of the robots, called informed robots, are provided with knowledge of the destination, while the other robots can only use local state feedback (i.e., position feedback from immediate neighbors and absolute orientation measurement). Furthermore, while multiple informed robots may be used for rendezvous, the analysis and results of this work are focused on a single informed robot. The techniques proposed in this

work could be extended to the case of multiple informed robots by using containment control [27, 35, 36]. The interaction among the robots is modeled as a directed graph $\mathscr{G}(t) = (\mathscr{V}, \mathscr{E}(t))$, where the node set $\mathscr{V} = \{1, \cdots, N\}$ represents the group of robots, and the edge set $\mathscr{E}(t)$ denotes time-varying edges. The set of informed robots and followers are denoted as $\mathscr{V}_L$ and $\mathscr{V}_F$, respectively, such that $\mathscr{V}_L \cup \mathscr{V}_F = \mathscr{V}$ and $\mathscr{V}_L \cap \mathscr{V}_F = \emptyset$. Let $\mathscr{V}_L = \{1\}$ and $\mathscr{V}_F = \{2, \cdots, N\}$. A directed edge $(i, j) \in \mathscr{E}$ in $\mathscr{G}(t)$ exists between node i and j if their relative distance $d_{ij} \triangleq \|p_i - p_j\| \in \mathbb{R}^+$ is less than R. The directed edge (i, j) indicates that node i is able to access the states (i.e., position and orientation) of node j through local sensing, but not vice versa. Accordingly, node j is a neighbor of node i (also called the parent of node i), and the neighbor set of node i is denoted as $\mathscr{N}_i = \{j \mid (i, j) \in \mathscr{E}\}$, which includes the nodes that can be sensed. A directed spanning tree is a directed graph, where every node has one parent except for one node, called the root, and the root node has directed paths to every other node in the graph. Since the follower robots are not aware of the destination, they have to stay connected with the informed robot either directly or indirectly through concatenated paths, such that the knowledge of the destination can be delivered to all the nodes through the connected network. Hence, to complete the desired tasks, maintaining connectivity of the underlying graph is necessary.

The main objectives are to derive a set of distributed controllers using only local information (i.e., the position feedback from other robots within a sensing area) to lead the robots to rendezvous at a common destination p^* with a desired orientation θ^*, i.e., $q_i^* = \left[(p^*)^T \ \theta^* \right]^T \ \forall i$ in the workspace $\mathscr{F}$, while guaranteeing the underlying graph $\mathscr{G}(t)$ remains connected during the system evolution, provided the given initial graph has a directed spanning tree.

Assumption 1. The initial graph $\mathscr{G}(0)$ has a directed spanning tree with the informed node as the root.

3.2.2. Control Design

In contrast to the fully actuated dynamics in (1), mobile agents with nonholonomic constraints in (8) are considered. The navigation function introduced in [31] and [32] ensures global convergence of the closed-loop system; however, the approach is not suitable for nonholonomic systems, since the feedback law generated from the gradient of the navigation function can lead to undesirable behavior. To overcome the undesirable behaviors, the original navigation function was extended to a Dipolar Navigation Function in [37] and [38], where the flow lines created in the potential field resemble a dipole, so that the flow lines are all tangent to the desired orientation at the origin and the vehicle can achieve the desired orientation. One example of the dipolar navigation is shown in Fig. 6, where the potential field has a unique minimum at the destination (i.e., $p^* = [0, 0]^T$ and $\theta^* = 0$), and achieves the maximums at the workspace boundary of $R_w = 5$. Note that the surface $x = 0$ divides the workspace into two parts and forces all the flow lines to approach the destination parallel to the y-axis.

The control strategy is to develop a dipolar navigation function for the informed robot, which creates a feasible nonholonomic trajectory for the nonholonomic robot and guarantees the achievement of the specified destination with a desired orientation, while other follower robots aim to achieve consensus with the informed robot and maintain network connectivity by using only local interaction with neighboring robots. Following
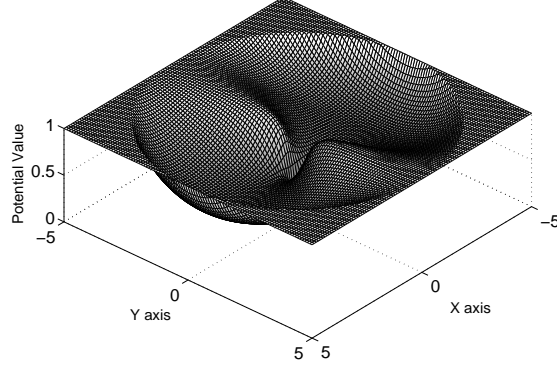
Figure 6. An example of a dipolar navigation function with a workspace of $R_w = 5$ and destination located at the origin with a desired orientation $\theta^* = 0$.

this idea, the dipolar navigation function is designed for the informed node $i \in \mathcal{V}_L$ as $\varphi_i^d(t) : \mathcal{F} \to [0,1]$,

$$\varphi_i^d = \frac{\gamma_d}{\left(\gamma_d^\alpha + H_d \cdot \beta_d\right)^{1/\alpha}}, \tag{9}$$

where $\alpha \in \mathbb{R}^+$ is a tuning parameter. The goal function $\gamma_d(t) : \mathbb{R}^2 \to \mathbb{R}^+$ in (9) encodes the control objective of achieving the desired destination, specified by the distance from $p_i(t) \in \mathbb{R}^2$ to the destination $p^* \in \mathbb{R}^2$, which is designed as

$$\gamma_d = \|p_i(t) - p^*\|^2.$$

The factor $H_d(t) \in \mathbb{R}^+$ in (9) creates a repulsive potential to align the trajectory of node i at the destination with the desired orientation. The repulsive potential factor is designed as

$$H_d = \varepsilon_{nh} + \left(\left(p_i - p^*\right)^T \cdot n_d\right)^2, \tag{10}$$

where ε_{nh} is a small positive constant, and $n_d = \left[\cos(\theta^*) \sin(\theta^*)\right]^T \in \mathbb{R}^2$. A small disk area with radius $\delta_1 < R$ centered at node i is denoted as a collision region. To prevent a potential collision between node i and the workspace boundary $\partial \mathcal{F}$, the function $\beta_d : \mathbb{R}^2 \to [0,1]$ in (9) is designed as

$$\beta_d = \begin{cases} -\frac{1}{\delta_1^2} d_{i0}^2 + \frac{2}{\delta_1} d_{i0}, & d_{i0} < \delta_1 \\ 1, & d_{i0} \geq \delta_1, \end{cases} \tag{11}$$

where $d_{i0} \triangleq R_w - \|p_i\| \in \mathbb{R}$ is the relative distance of node i to the workspace boundary.

Since γ_d and β_d in (9) are guaranteed to not be zero simultaneously, the navigation function candidate in (9) achieves its minimum of 0 when $\gamma_d = 0$ and achieves its maximum of 1 when $\beta_d = 0$. Our previous work in [33] proves that the original navigation function with the form of $\varphi_i = \frac{\gamma_i}{\left(\gamma_i^\alpha + \beta_i\right)^{1/\alpha}}$ is a qualified navigation function. It is also

shown in [39] that the navigation properties are not affected by the modification to a dipolar navigation with the design of (10), as long as the workspace is bounded, H_d in (9) can be bounded in the workspace, and ε_{nh} is a small positive constant. As a result, the decentralized navigation function φ_i^d proposed in (9) can be proven to be a qualified navigation function by following a similar procedure in [39] and [33]. From the properties of the navigation function, it is known that almost all initial positions (except for a set of measure zero points) asymptotically approach the desired destination.

To track the informed node while maintaining network connectivity, a local interaction rule is designed for each follower node $i \in \mathscr{V}_F$ as $\varphi_i^f(t) : \mathscr{F} \to [0,1]$,

$$\varphi_i^f = \frac{\gamma_i}{(\gamma_i^\alpha + \beta_i)^{1/\alpha}}, \tag{12}$$

where $\alpha \in \mathbb{R}^+$ is a tuning parameter. The goal function $\gamma_i(t) : \mathbb{R}^2 \to \mathbb{R}^+$ in (12) encodes the control objective of achieving consensus on the position between node i and neighboring nodes $j \in \mathscr{N}_i$, which is designed as

$$\gamma_i = \sum_{j \in \mathscr{N}_i} \left\| p_i(t) - p_j(t) \right\|^2. \tag{13}$$

To ensure connectivity of the existing links between nodes i and its neighboring nodes $j \in \mathscr{N}_i$, an escape region for each node is defined as the outer ring of the sensing area with radius r, $R - \delta_2 < r < R$, where $\delta_2 \in \mathbb{R}^+$ is a predetermined buffer distance. Each edge formed by node i and the adjacent node $j \in \mathscr{N}_i$ in the escape region have the potential to break connectivity. Hence, the constraint function $\beta_i : \mathbb{R}^{2N} \to [0,1]$ in (12) is designed as

$$\beta_i = \prod_{j \in \mathscr{N}_i} b_{ij}, \tag{14}$$

where $b_{ij} \triangleq b(p_i, p_j) : \mathbb{R}^2 \to [0,1]$ ensures connectivity of the existing links between nodes i and its neighboring nodes $j \in \mathscr{N}_i$ (i.e., guarantees that nodes $j \in \mathscr{N}_i$ will never leave the sensing and communication zone of node i if node j is initially connected to node i) and is designed as

$$b_{ij} = \begin{cases} 1, & d_{ij} \leq R - \delta_2 \\ -\frac{1}{\delta_2^2}(d_{ij} + 2\delta_2 - R)^2 \\ +\frac{2}{\delta_2}(d_{ij} + 2\delta_2 - R), & R - \delta_2 < d_{ij} < R \\ 0, & d_{ij} \geq R. \end{cases} \tag{15}$$

The constraint function in (14) is designed to vanish whenever node i meets the constraints of network connectivity in the workspace, (i.e., if node i departs from its neighbor nodes $j \in \mathscr{N}_i$ to a distance of R). Since γ_i and β_i in (12) will not be zero simultaneously from their definitions, it is clear that φ_i^f achieves its minimum of 0 if $\gamma_i = 0$ (i.e., the consensus is reached between node i and its immediate neighbors), and φ_i^f achieves its maximum of 1 if $\beta_i = 0$ (i.e., the constraint of network connectivity is met).

For brevity, φ_i is used to represent the potential function designed for each node i, where particularly $\varphi_i = \varphi_i^d$ in (9) if $i \in \mathscr{V}_L$, and $\varphi_i = \varphi_i^f$ in (12) if $i \in \mathscr{V}_F$. The desired orientation for any robot $i \in \mathscr{V}$, denoted by $\theta_{di}(t)$, is defined as a function of the negative gradient of the decentralized function φ_i as,

$$\theta_{di} \triangleq \arctan 2\left(-\frac{\partial \varphi_i}{\partial y_i}, -\frac{\partial \varphi_i}{\partial x_i}\right), \tag{16}$$

where $\arctan 2\left(\cdot\right): \mathbb{R}^2 \to \mathbb{R}$ denotes the four quadrant inverse tangent function, and $\theta_{di}\left(t\right)$ is confined to the region of $\left(-\pi, \pi\right]$. By defining $\theta_{di}\big|_{p^*} = \arctan 2\left(0, 0\right) = \theta_i\big|_{p^*}$, θ_{di} remains continuous along any approaching direction to the goal position. Based on the definition of θ_{di} in (16)

$$\nabla_i \varphi_i = -\|\nabla_i \varphi_i\|\left[\cos\left(\theta_{di}\right) \sin\left(\theta_{di}\right)\right]^T, \tag{17}$$

where $\nabla_i \varphi_i = \left[\frac{\partial \varphi_i}{\partial x_i} \frac{\partial \varphi_i}{\partial y_i}\right]^T$ denotes the partial derivative of φ_i with respect to p_i, and $\|\nabla_i \varphi_i\|$ denotes the Euclidean norm of $\nabla_i \varphi_i$. The difference between the current orientation and the desired orientation for robot i at each time instant is defined as

$$\tilde{\theta}_i\left(t\right) = \theta_i\left(t\right) - \theta_{di}\left(t\right), \tag{18}$$

where $\theta_{di}\left(t\right)$ is generated from the decentralized navigation function φ_i and (16).

Based on the open-loop system in (8), the controller for each robot (i.e., the linear and angular velocity of robot i) is designed as

$$v_i = k_{v,i}\|\nabla_i \varphi_i\| \cos \tilde{\theta}_i, \tag{19}$$

$$\omega_i = -k_{w,i}\tilde{\theta}_i + \dot{\theta}_{di}, \tag{20}$$

where $k_{v,i}, k_{w,i} \in \mathbb{R}^+$ denote the control gains for robot i.

3.2.3. Simulation Results

The following numerical simulation demonstrates the performance of the controller developed in (19) and (20) in a scenario in which a group of six mobile robots with the kinematics in (8) are navigated to the common destination $p^* = \left[0\ 0\right]^T$ with the desired orientation $\theta^* = 0$. The limited communication and sensing zone for each robot is assumed as $R = 2$ m and $\delta_1 = \delta_2 = 0.4$ m. The tuning parameter α in (9) is selected as $\alpha = 1.2$. The group of mobile robots is arbitrarily deployed in the workspace and forms a connected graph. The informed node is randomly selected from the group, and is the only node aware of the desired destination p^* and orientation θ^*. The control laws in (19) and (20) yield the simulation results shown in Fig. 7-9. Fig. 7 shows the trajectory for each robot, where the associated arrows indicate the initial or final orientation. In Fig. 8, the position and orientation error plot indicates that each robot achieves the common destination with the desired orientation. The evolution of the inter-robot distance is shown in Fig. 9, which implies that the connectivity of the underlying graph is maintained, since the inter-robot distance is less than the radius $R = 2$ m during the motion.

4. Conclusion and Future Work

An overview of the current research in preserving network connectivity for networked agents is provided, with a focus on the use of navigation function framework in two particular applications, formation control and rendezvous, for agents with limited sensing
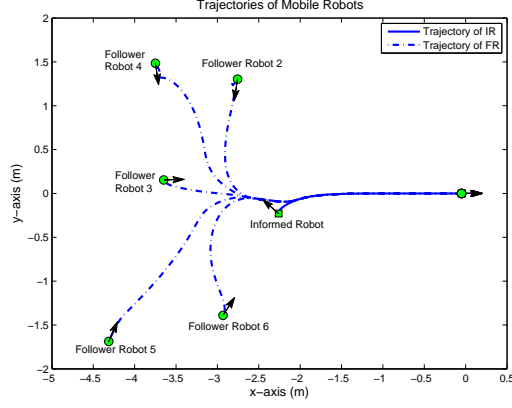
Figure 7. Plot of robot trajectories with solid line and dot-dash line indicating the trajectory of the informed robot (IR) and the follower robot (FR), respectively.
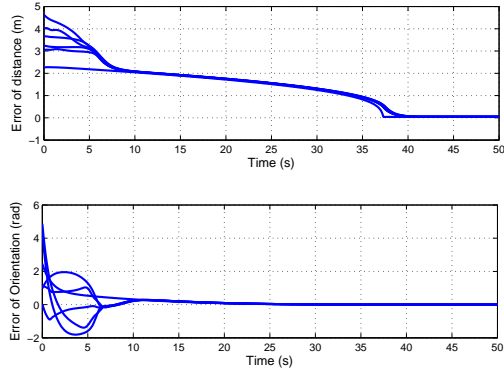

Figure 8. Error plot of the distance to the destination and the error plot of the orientation $\theta_i - \theta^*$.

and network connectivity constraints. Prior works based on artificial potential fields can cause the system to be trapped by local minima, and thus result in mission failure. By modeling the network connectivity as an artificial obstacle in navigation functions, the developed control strategy ensures global convergence to the unique minimum of the potential field (i.e., control objective) while maintaining network connectivity.

In the formation control result, the initial topology is assumed to be a supergraph of the desired topology, which ensures that the agents are originally in a feasible interconnected state. Additional efforts could consider formation control from an arbitrary initial graph to a desired graph. Additional efforts could also incorporate more realism into the physical and communications models, by accounting for the dynamics of the robots and the effects of those on communications, and incorporating more realistic channel models. In the rendezvous result, although robots are guaranteed to converge to the desired destination, the rate of convergence is not considered. Generally, the rate of convergence depends on the network topology, which is a function of the roles of nodes (i.e., informed
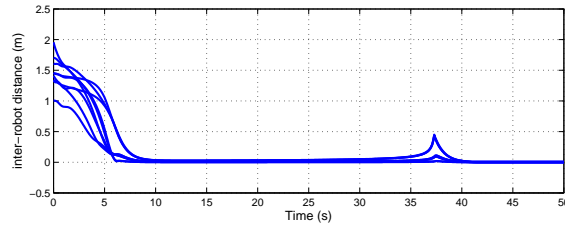
Figure 9. The evolution of inter-robot distance.

nodes or followers) and their interactions. A different set of informed nodes may lead to different convergence rates. Extension of this work could seek to optimize performance metrics such as the degree of connectivity and the convergence rate of the network in scenarios where the set of informed nodes can be determined and/or positioned a priori.

References

[1] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *IEEE Proc.*, vol. 94, no. 7, pp. 1306–1324, 2006.

[2] F. Zhang and N. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 650–663, 2010.

[3] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multia-gent hybrid systems," *IEEE Trans. on Autom. Control*, vol. 43, no. 4, pp. 509–521, 1998.

[4] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2001.

[5] R. Merris, "Laplacian matrices of graphs: A survey," *Lin. Algebra. Appl.*, vol. 197-198, pp. 143–176, 1994.

[6] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Trans. Automat. Control*, vol. 51, no. 1, pp. 116–120, Jan. 2006.

[7] M. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. IEEE Conf. Decis. Control*, Dec. 2006, pp. 3628–3633.

[8] M. Zavlanos and G. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 812–816, Aug. 2007.

[9] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[10] M. Zavlanos and G. Pappas, "Controlling connectivity of dynamic graphs," in *Proc. IEEE Conf. Decis. Control*, Dec. 2005, pp. 6388–6393.

[11] K. Srivastava and M. Spong, "Multi-agent coordination under connectivity constraints," in *Proc. Am. Control Conf.*, June 2008, pp. 2648–2653.

[12] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie, "Maintaining limited-range connectivity among second-order agents," in *Proc. Am. Control Conf.*, 2006.

[13] F. Bullo, J. Cortés, and S. Martınez, "Distributed control of robotic networks," *Applied Mathematics Series. Princeton University Press*, 2009.

[14] D. Spanos and R. Murray, "Robust connectivity of networked vehicles," in *Proc. IEEE Conf. Decis. Control*, vol. 3, Dec. 2004, pp. 2893–2898.

[15] D. Dimarogonas and K. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Trans. Robot*, vol. 24, no. 5, pp. 1213 –1223, 2008.

[16] M. Zavlanos and G. Pappas, "Distributed connectivity control of mobile networks," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1416–1428, Dec. 2008.

[17] M. Zavlanos, H. Tanner, A. Jadbabaie, and G. Pappas, "Hybrid control for connectivity preserving flock-ing," *IEEE Trans. Automat. Control*, vol. 54, no. 12, pp. 2869 –2875, 2009.

[18] A. Ghaffarkhah and Y. Mostofi, "Communication-aware target tracking using navigation functions - centralized case," in *Int. Conf. Robot Commun. Co-ord.*, March 31 - April 2 2009, pp. 1 –8.

[19] D. V. Dimarogonas and K. J. Kyriakopoulos, "On the rendezvous problem for multiple nonholonomic agents," *IEEE Trans. Automat. Control*, vol. 52, no. 5, pp. 916–922, May 2007.

[20] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.

[21] D. Dimarogonas and K. Johansson, "Bounded control of network connectivity in multi-agent systems," *Control Theory Appl.*, vol. 4, no. 8, pp. 1330 –1338, Aug. 2010.

[22] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, 2007.

[23] Z. Kan, S. Subramanian, J. Shea, and W. E. Dixon, "Vision based connectivity maintenance of a network with switching topology," in *IEEE Multi-Conf. Syst. and Contr.*, Yokohama, Japan, September 2010, pp. 1493–1498.

[24] Z. Kan, A. Dani, J. Shea, and W. E. Dixon, "Information flow based connectivity maintenance of a multi-agent system during formation control," in *Proc. IEEE Conf. Decis. Control*, Orlando, FL, 2011, pp. 2375–2380.

[25] Z. Kan, E. L. Pasiliao, J. W. Curtis, and W. E. Dixon, "Particle filter based average consensus target tracking with preservation of network connectivity," in *Proc. IEEE Mil. Commun. Conf.*, 2012, pp. 760–765.

[26] Z. Kan, J. M. Shea, and W. E. Dixon, "Influencing emotional behavior in a social network," in *Proc. Am. Control Conf.*, Montréal, Canada, June 2012, pp. 4072–4077.

[27] Z. Kan, J. Klotz, E. L. Pasiliao, and W. E. Dixon, "Containment control for a directed social network with state-dependent connectivity," in *Proc. Am. Control Conf.*, Washington DC, June 2013, pp. 1953–1958.

[28] H. Su, X. Wang, and G. Chen, "Rendezvous of multiple mobile agents with preserved network connectivity," *Syst. Control Lett.*, vol. 59, no. 5, pp. 313–322, 2010.

[29] F. Xiao, L. Wang, and T. Chen, "Connectivity preservation for multi-agent rendezvous with link failure," *Automatica*, vol. 48, pp. 25–35, Jan. 2012.

[30] L. Navaravong, Z. Kan, J. M. Shea, and W. E. Dixon, "Formation reconfiguration for mobile robots with network connectivity constraints," *IEEE Network*, vol. 26, no. 4, pp. 18–24, 2012.

[31] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct 1992.

[32] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, pp. 412–442, Dec 1990.

[33] Z. Kan, A. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Trans. Automat. Control*, vol. 57, no. 7, pp. 1827– 1832, 2012.

[34] Z. Kan, J. Klotz, E. L. P. Jr, J. M. Shea, and W. E. Dixon. (2014) Decentralized rendezvous of nonholonomic robots with sensing and connectivity constraints. arXiv:1402.5639.

[35] Z. Kan, S. Mehta, E. Pasiliao, J. W. Curtis, and W. E. Dixon, "Balanced containment control and cooperative timing of a multi-agent system," in *Proc. Am. Control Conf.*, 2014.

[36] Y. Cao and W. Ren, "Containment control with multiple stationary or dynamic leaders under a directed interaction graph," in *Proc. IEEE Conf. Decis. Control*, 2009, pp. 3014–3019.

[37] H. Tanner and K. Kyriakopoulos, "Nonholonomic motion planning for mobile manipulators," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2000, pp. 1233 –1238 vol.2.

[38] H. Tanner, S. Loizou, and K. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 53–64, Feb 2003.

[39] S. Loizou and K. Kyriakopoulos, "Navigation of multiple kinematically constrained robots," *IEEE Trans. Robot*, vol. 24, no. 1, pp. 221 –231, 2008.