



Distributed Consensus Problem of Multiple Non-holonomic Mobile Robots

Nour Jaoura¹  · Lama Hassan¹  · Adel Alkafri¹

Received: 20 March 2021 / Revised: 4 July 2021 / Accepted: 28 July 2021 / Published online: 2 September 2021
© Brazilian Society for Automatics–SBA 2021

Abstract

In this paper, we propose a new consensus protocol for networks of multiple mobile robots with fixed communication topology using a strategy based on an invariant manifold technique. The agents are subjected to non-holonomic constraints and transformed into Brockett integrator form. The objective is to asymptotically stabilize the non-holonomic model for each agent about a group decision value using distributed protocol. This approach solves the consensus problem in two cases: leader–follower consensus problem when the leader is static and leaderless consensus problem. Then, a design of test-bed is fully described for testing distributed protocols. The setup consists of a group of non-holonomic mobile robots moving on a platform with different ArUco markers on their tops, an overhead camera to determine the poses (positions and orientations) of these markers, and a desktop computer to provide the interaction between the robots. The design of this test-bed has relied on ROS framework as a software platform for offering ROS network architecture a solution for distributed communication. Finally, results were presented to show the performance of this design by applying the proposed protocol.

Keywords ArUco · Consensus · Distributed · Graph · Mobile robot · Multi-agent · Non-holonomic · ROS · Test-bed

1 Introduction

Multi-autonomous agents capable of working cooperatively can be used to accomplish many tasks more efficiently and robustly than with a single agent. For this reason, they have been intensively studied and received considerable attention in the last few decades for different applications, such as search and rescue mission (Davids 2002), formation control (Lei et al 2012), and rendezvous (Cortés et al 2006). One of the most important problems in cooperative control is consensus problem, which refers to a group of agents interacting with each other via a sensing or communication network to reach an agreement on a common value (Li and Duan 2017), this value can be the position or speed of agents. Existing consensus algorithms can be roughly classified into two

branches: the first one is leaderless consensus, which means that the agents aim to reach a common value between them. The second is leader–follower consensus which aims to track or converge to the leader’s state (Li and Duan 2017).

Many works have been carried out to make a team of agents cooperate with each other and reach consensus. Most of them use the distributed approach to achieve this goal. The first who solved consensus problem for networks of dynamic agents were Olfati-Saber and Murray (Saber and Murray 2003; Olfati-Saber and Murray 2004), who modeled each agent in the network as a first integrator. Later, consensus of multi-agent systems with general linear dynamics appeared in Li et al (2009). On the other hand, recent years have seen a lot of works related to consensus problem with nonlinear systems (Li et al 2012; Rehan et al 2017; ur Rehman et al 2018) to name a few. However, these results cannot be applied on multi-agent systems with non-holonomic restrictions since such systems are not stabilizable with continuous pure state feedback laws as indicated in Khennouf and De Wit (1995). As solution to this problem, Dimarogonas (2007) introduced a decentralized feedback control strategy, which is time invariant and discontinuous to drive a system of multiple non-holonomic unicycles to a rendezvous point in terms of both position and orientation. Consensus problem

✉ Nour Jaoura
nour.jaoura@hiast.edu.sy

Lama Hassan
lama.alhassan@hiast.edu.sy

Adel Alkafri
adel.alkafri@hiast.edu.sy

¹ Higher Institute for Applied Sciences and Technology,
Damascus, Syria

of network consisted of non-holonomic systems described by chained form was studied in Zhai et al (2010), which proposed a time-invariant continuous state feedback controller. Yet the authors have not been able to determine the relationship between the consensus value and the initial states. In addition, consensus problems on different kinds of reference signals have been realized using the protocol devised by Cao et al (2014). Recently, a solution to the full consensus problem of non-holonomic vehicles based on a complete kinematics dynamics model and a smooth time-varying δ -persistently exciting controller presented in Maghenem et al (2018).

According to the importance of experimental verification of new theories, different platforms were designed in multi-agent systems field. For example, In Cremean et al (2002), a platform for testing decentralized control methodologies for multiple vehicle coordination and formation stabilization has been introduced. Another test-bed for experimental validation of multi-robot systems is represented in Antonelli et al (2009) and Jácome et al (2019), where in Jácome et al (2019), the test-bed was designed based on the position control and trajectory tracking of mini-sized ground robots. In addition, a basic structure of swarming UAV system using ROS (Robot Operating System) framework was created and tested using software in the loop (SITL) simulator in Lamping et al (2018) and Indriyanto et al (2020).

In this paper, we concentrate on non-holonomic systems because they provide an important class of mechanical control systems. This importance arises from the fact that there are a lot of mechanical systems with non-integrable constraints, which make them controllable but not stabilizable by smooth time-invariant state feedback control laws (Brockett 1983), such as wheeled mobile robots, robots manipulator, and underwater or aerial vehicles. In this paper, we choose wheeled mobile robots as an example of non-holonomic systems.

The main contributions of this paper are summarized in two points as follows. Firstly, a new control law for multiple non-holonomic systems is proposed to achieve stabilization of all the states and reach consensus on the uncontrollable state of all agents to solve rendezvous problem in a specific formation with the same orientation. Secondly, an experimental multi-agent test-bed is presented for supporting and validating distributed multi-agent protocols.

The rest of this paper is organized as follows. Section 2 presents concepts in graph theory and the mathematical equations describing the kinematics of a unicycle-type wheeled mobile robot. Then, the proposed distributed controller is presented. Section 4 details the test-bed architecture with its main components and the distributed structure of ROS's application. Section 5 illustrates the real results of the proposed controller. Finally, conclusions are given in Sect. 6.

2 Preliminaries and Problem Formulation

2.1 Preliminaries

Consider a weighted graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ of order N , where \mathcal{V} is a set of nodes $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$, \mathcal{E} is a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and \mathcal{A} is a weighted adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ with positive adjacency elements a_{ij} . Each element in \mathcal{E} is denoted by $e_{ij} = (v_i, v_j)$ which is termed an edge from parent node v_i to child node v_j , and it means that v_j can obtain information from v_i . A graph is said to be undirected if all edges in the graph are bidirectional, otherwise the graph is directed. The set of neighbors of v_i is $\mathcal{N}_i = \{v_j : (v_j, v_i) \in \mathcal{E}\}$ (Olfati-Saber and Murray 2004).

A path on G from node v_{i_1} to node v_{i_l} is a sequence of ordered edges of the form $(v_{i_k}, v_{i_{k+1}}), k = 1, \dots, l - 1$. A/directed/undirected graph G is called strongly connected/connected if there is a/directed/undirected path from every node to every other nodes. A spanning tree is a graph where every node can get information from only one node, called the root, which has no parent and has directed paths to all other nodes. If a graph is strongly connected, it contains at least one spanning tree (Li and Duan 2017).

The adjacency matrix $\mathcal{A} = [a_{ij}]$ associated with the graph G is defined by $a_{ii} = 0, a_{ij} > 0$ if $(v_j, v_i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The Laplacian matrix $\mathcal{L}_N = [l_{ij}] \in \mathbb{R}^{N \times N}$ is defined as (Li and Duan 2017):

$$l_{ij} = \begin{cases} \sum_{j=1, j \neq i}^N a_{ij} & i = j \\ -a_{ij} & i \neq j \end{cases}$$

Lemma 1 *The matrix \mathcal{L}_N has the following properties: (Li and Duan 2017)*

- Zero is an eigenvalue of \mathcal{L}_N with $\mathbf{1}_N$ as a corresponding right eigenvector, i.e., $\mathcal{L}_N \mathbf{1}_N = 0$, and it is a simple eigenvalue if and only if the graph has a directed spanning tree.
- The rest eigenvalues of \mathcal{L}_N have positive real parts. In particular, if the graph is undirected, then they are all positive and real.

2.2 Problem Formulation

Consider a team of N differential driven mobile robots moving on a plane, as shown in Fig. 1. Each agent is described by the following equations in the global coordinates $\{X, Y\}$ (Anvari 2013; De Luca and Oriolo 1995):

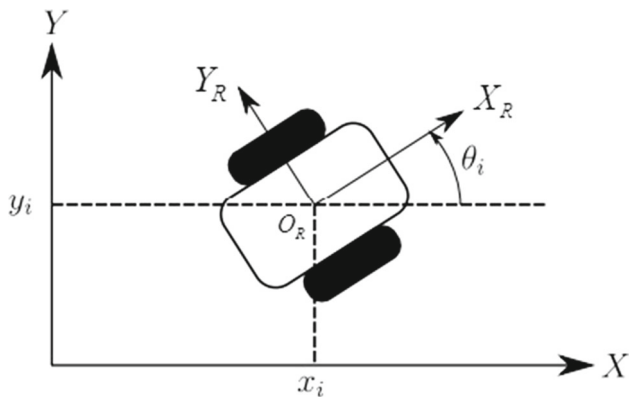


Fig. 1 Definition of configuration variables

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \\ 0 \end{bmatrix} v_i + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_i \tag{1}$$

where x_i, y_i are the coordinates of the midpoint of the mobile robot i in the global coordinate frame, θ_i is its orientation with respect to positive X-axis, v_i, w_i are the linear and angular velocities, respectively, where the linear velocity is defined as the average of the linear velocities of the two wheels: $v_i = \frac{v_r + v_l}{2}$, and the angular velocity is: $w_i = \frac{v_r - v_l}{L}$ where L is the distance between the wheels, and v_r, v_l are the velocities of right and left wheels, respectively (De Luca and Oriolo 1995). Therefore, the kinematics of a differential robot can be described by unicycle kinematics as in (1).

The non-holonomic constraint for the model (1) is given in Anvari (2013) and De Luca and Oriolo (1995) by:

$$-\dot{x}_i \sin \theta_i + \dot{y}_i \cos \theta_i = 0 \tag{2}$$

By applying the change of coordinates (DeVon and Bretl 2007) with adding a displacement to the coordinates (x_i, y_i):

$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{bmatrix} = \begin{bmatrix} \theta_i \\ \tilde{p}_{ix} \cos(\theta_i) + \tilde{p}_{iy} \sin(\theta_i) \\ -2(\tilde{p}_{ix} \sin(\theta_i) - \tilde{p}_{iy} \cos(\theta_i)) + x_{i1}x_{i2} \end{bmatrix} \tag{3}$$

where

$$\begin{aligned} \tilde{p}_{ix} &= x_i - p_{x_i} \\ \tilde{p}_{iy} &= y_i - p_{y_i} \end{aligned}$$

Knowing that p_{x_i}, p_{y_i} are the displacements on the X, Y axes, respectively.

This transformation (3) preserves the origin and transforms the system (1) into non-holonomic integrator (or

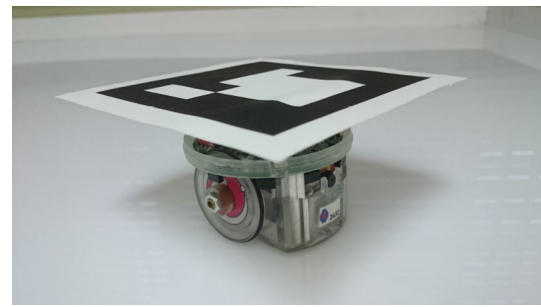


Fig. 2 An e-puck robot with its mounted ArUco marker

Table 1 Computer hardware specifications

Item	Details
Processor	Intel core i7-7700,3.6 GHz, 8 MB Cache
Motherboard	ASUS PRIME Z270-P
GPU card	Geforce GTX 1050Ti, 4GB GDDR5
RAM	16 GB DDR4 ZEPPELIN
Storage HDD	1 TB SEGEAT
Storage SSD	240 GB

Brockett integrator) form:

$$\begin{bmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \\ \dot{x}_{i3} \end{bmatrix} = \begin{bmatrix} u_{i1} \\ u_{i2} \\ x_{i1}u_{i2} - x_{i2}u_{i1} \end{bmatrix} \tag{4}$$

For N differential mobile robots, Eq. (4) can be rewritten in matrix form:

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ \mathbf{X}_1 U_2 - \mathbf{X}_2 U_1 \end{bmatrix} \tag{5}$$

where $X_j = [x_{1j}, x_{2j}, \dots, x_{Nj}]^T$ for $j = 1, 2, 3, U_k = [u_{1k}, u_{2k}, \dots, u_{Nk}]^T$ for $k = 1, 2, \mathbf{X}_1$ and \mathbf{X}_2 are diagonal matrices in which diagonal entries are the elements of the vectors X_1 and X_2 , respectively.

3 Consensus and Protocol

Proposition 1 (Ren and Beard 2008; Li and Duan 2017) For system (4), protocol u_{i1}, u_{i2} solves the consensus problem, if and only if the states of all agents converge to the same vector, i.e.,

$$\lim_{t \rightarrow \infty} \|x_{ik} - x_{jk}\| = 0, \quad \forall k = 1, 2, 3, \quad \forall i, j = 1, \dots, N, \quad i \neq j$$

Fig. 3 The implemented test-bed

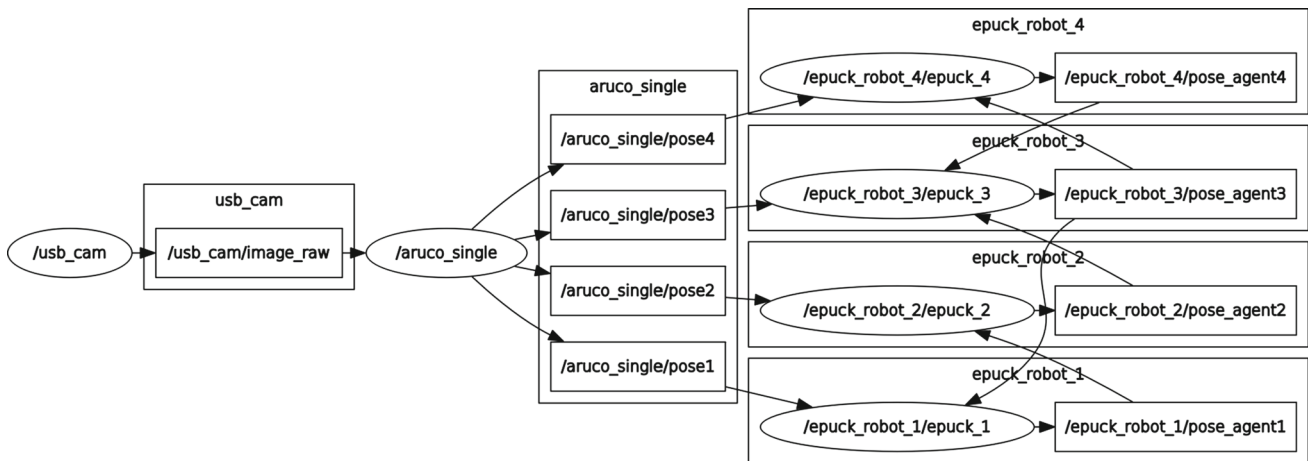
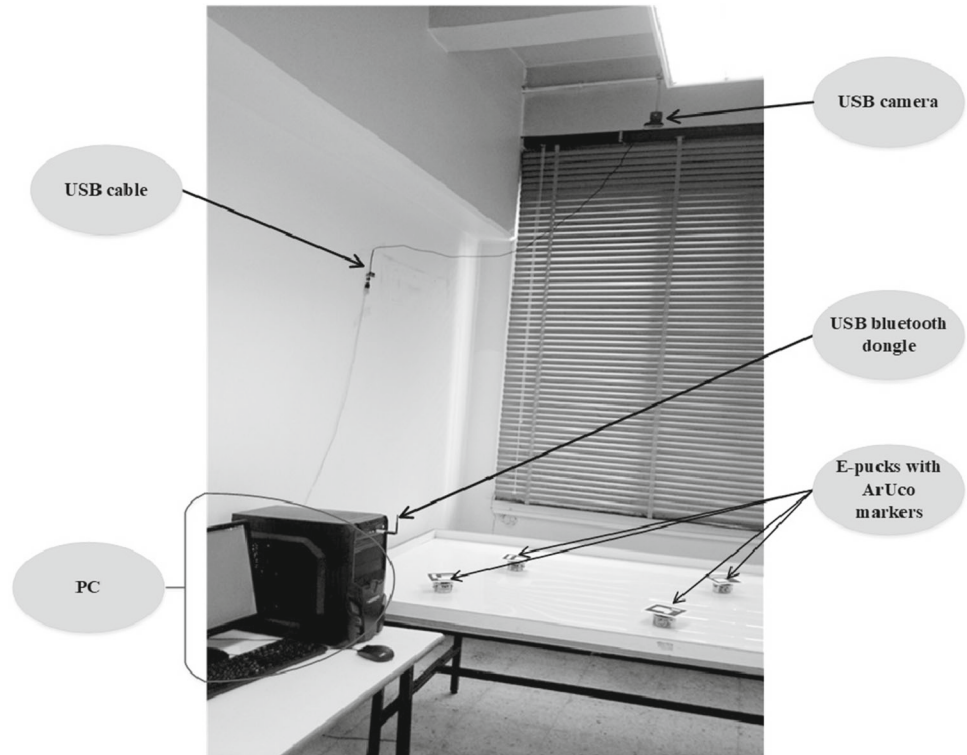


Fig. 4 ROS graph for the application

This leads to the following:

$$\lim_{t \rightarrow \infty} \left(\begin{bmatrix} \tilde{p}_{ix} - \tilde{p}_{jx} \\ \tilde{p}_{iy} - \tilde{p}_{jy} \end{bmatrix} \right) = 0, \quad \lim_{t \rightarrow \infty} (\theta_i - \theta_j) = 0 \quad (6)$$

In Cao et al (2014), it has been confirmed that by applying the linear protocol (7) on system (5), the consensus of X_1 and X_2 is obtained, while consensus of the third state cannot be reached due to the uncontrollability of subsystem X_3 when U_1 reaches consensus.

$$\begin{aligned} U_1 &= -k_1 \mathcal{L}_N X_1 \\ U_2 &= -k_2 \mathcal{L}_N X_2 + k_3 U_1 \mathcal{L}_N X_3 \end{aligned} \quad (7)$$

In order to get (6) and solve consensus problem, we propose an extension of the controller presented in Astolfi (1998), Khennouf and De Wit (1995), Tsiotras (1997) and DeVon and Bretl (2007). In our approach, an invariant manifold is chosen based on relative state information of neighboring agents. Before discussing in detail our proposed controller, let us present the one developed in DeVon and Bretl (2007). Assuming that the initial state of the i th agent $x_i(0) = (x_{i1}(0), x_{i2}(0), x_{i3}(0)) \in \mathbb{R}^3$ verifies $x_{i1}^2(0) + x_{i2}^2(0) > 0$, then the kinematic control law is given by the following equations:

$$\begin{aligned} u_{i1} &= -k_1 x_{i1} + k_2 \frac{-s_i(x_i)}{x_{i1}^2 + x_{i2}^2} x_{i2} \\ u_{i2} &= -k_1 x_{i2} + k_2 \frac{s_i(x_i)}{x_{i1}^2 + x_{i2}^2} x_{i1} \end{aligned} \tag{8}$$

where $s_i(x_i) = x_{i3}$ is a smooth function achieving the condition $\dot{x}_{i3} = -k_2 x_{i3}$ for any x_{i3} and for any (x_{i1}, x_{i2}) achieving $x_{i1}^2 + x_{i2}^2 > 0$ (Astolfi 1998). k_1, k_2 are positive constants with the condition $k_2 > 2k_1$ to ensure the boundedness of the control law (8) (Khennouf and De Wit 1995; DeVon and Bretl 2007).

The objective of control law (8) is to force $s_i(x_i) = x_{i3}$ to converge to an invariant manifold $\mathcal{M} = \{x_i \in \mathbb{R}^3 \mid x_{i3} = 0\}$, then the remaining states x_{i1}, x_{i2} converge to the origin. In the following, we will demonstrate how to extend this control law to solve the leaderless consensus problem and the leader–follower problem in multi-agent network.

3.1 Leaderless Problem

Based on the aforementioned control law, the proposed distributed controller to solve the leaderless consensus problem is defined as follows:

$$\begin{aligned} u_{i1} &= -k_1 x_{i1} - k_2 \frac{\sum_{j \in \mathcal{N}_i} a_{ij} (x_{j3} - x_{i3})}{x_{i1}^2 + x_{i2}^2} x_{i2} \\ u_{i2} &= -k_1 x_{i2} + k_2 \frac{\sum_{j \in \mathcal{N}_i} a_{ij} (x_{j3} - x_{i3})}{x_{i1}^2 + x_{i2}^2} x_{i1} \end{aligned} \tag{9}$$

where a_{ij} is the edge weight constant between two agents i, j . k_1 and k_2 are positive constants verifying the condition $k_2 > 2k_1$.

We substitute $s(x)$ in (8) by the state information exchanged between mobile robots, i.e., $s_i(x) = \sum_{j \in \mathcal{N}_i} a_{ij} (x_{j3} - x_{i3})$ and we assume that $x_i = (x_{i1}, x_{i2}, x_{i3}) \notin \mathbb{D}$, where the set \mathbb{D} is defined as:

$$\mathbb{D} = \{x_i \in \mathbb{R}^3 \mid x_{i1}^2 + x_{i2}^2 > 0, s_i(x) \neq 0 \ \forall \ i = 1, \dots, N\} \tag{10}$$

Substituting (9) into (4) and rewriting the closed-loop system for each mobile robot gives:

$$\begin{bmatrix} \dot{x}_{i1} \\ \dot{x}_{i2} \\ \dot{x}_{i3} \end{bmatrix} = \begin{bmatrix} u_{i1} \\ u_{i2} \\ k_2 \sum_{j \in \mathcal{N}_i} a_{ij} (x_{j3} - x_{i3}) \end{bmatrix} \tag{11}$$

We can rewrite (11) in matrix form as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ k_2 S(x) \end{bmatrix} \tag{12}$$

where U_1, U_2 are defined by:

$$\begin{aligned} U_1 &= -k_1 X_1 - k_2 \mathbf{X}_2 \mathbf{X}_{12}^{-1} S(x) \\ U_2 &= -k_1 X_2 + k_2 \mathbf{X}_1 \mathbf{X}_{12}^{-1} S(x) \end{aligned} \tag{13}$$

And $S(x) = [s_1(x), \dots, s_N(x)]^T = -\mathcal{L}_N X_3$, \mathcal{L}_N is the Laplacian matrix associated with the communication graph between mobile robots, and \mathbf{X}_{12} is a diagonal matrix its entries are the elements of the vector $[x_{11}^2 + x_{12}^2, \dots, x_{N1}^2 + x_{N2}^2]^T$.

Substituting (13) into (12) and writing the closed-loop system for N mobile robots gives:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -k_1 X_1 + k_2 \mathbf{X}_2 \mathbf{X}_{12}^{-1} \mathcal{L}_N X_3 \\ -k_1 X_2 - k_2 \mathbf{X}_1 \mathbf{X}_{12}^{-1} \mathcal{L}_N X_3 \\ -k_2 \mathcal{L}_N X_3 \end{bmatrix} \tag{14}$$

Proposition 2 Consider the system (5) in a directed topology and the control law (13) with the condition on the initial state of each mobile robot, i.e., $x_i \notin \mathbb{D}$ (10). If the directed graph of this topology has a spanning tree, then the state feedback control law (13) exponentially stabilizes system (5) and reaches consensus on the state X_3 .

Proof Consider the candidate Lyapunov function:

$$V(X(t)) = X_1^T(t) X_1(t) + X_2^T(t) X_2(t) + X_3^T(t) \mathcal{L}_N^T X_3(t)$$

which is positive definite while $x_i \notin \mathbb{D}$ and \mathcal{L}_N has a spanning tree. Differentiating with respect to time yields:

$$\begin{aligned} \dot{V} &= 2X_1^T \dot{x}_1 + 2X_2^T \dot{x}_2 + 2X_3^T \mathcal{L}_N^T \dot{x}_3 \\ &= -2k_1 (X_1^T X_1 + k_2 X_1^T \mathbf{X}_2 \mathbf{X}_{12}^{-1} \mathcal{L}_N X_3 + X_2^T X_2 \\ &\quad + k_2 X_2^T \mathbf{X}_1 \mathbf{X}_{12}^{-1} \mathcal{L}_N X_3) - 2k_2 X_3^T \mathcal{L}_N^T \mathcal{L}_N X_3 \end{aligned}$$

Since $\mathbf{X}_1, \mathbf{X}_2$ are diagonal matrices, then the terms $k_2 X_1^T \mathbf{X}_2 \mathbf{X}_{12}^{-1} \mathcal{L}_N X_3, k_2 X_2^T \mathbf{X}_1 \mathbf{X}_{12}^{-1} \mathcal{L}_N X_3$ are equal. For this reason, we have:

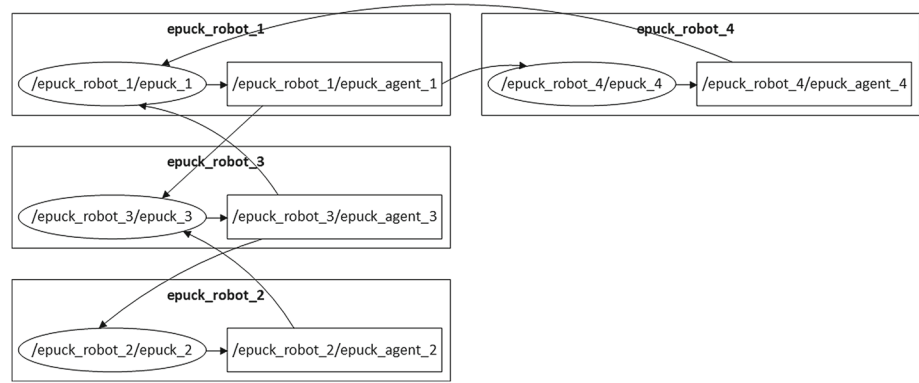
$$\dot{V} = -2k_1 X_1^T X_1 - 2k_1 X_2^T X_2 - 2k_2 X_3^T \mathcal{L}_N^T \mathcal{L}_N X_3$$

which is negative definite. According to LaSalle invariant principle, the largest invariant set verifying $\dot{V} = 0$, to which the system converge, is:

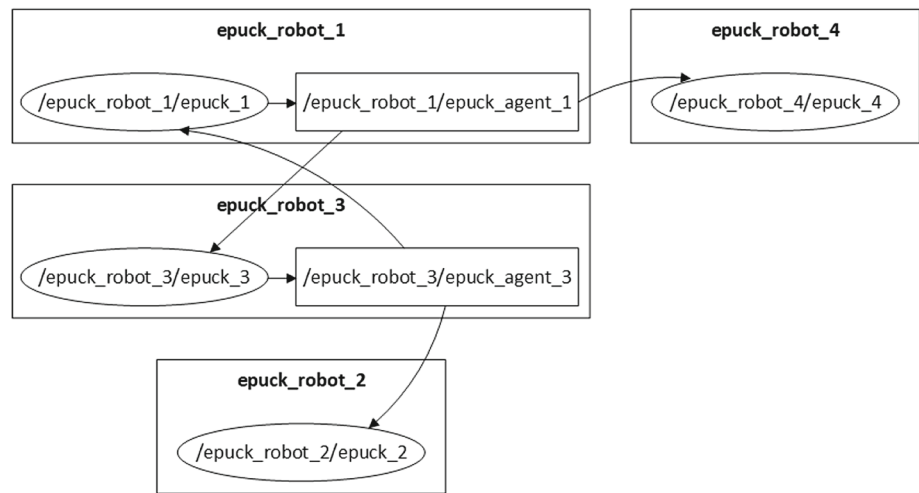
$$\Omega = \{\mathbf{X}_1 = \mathbf{X}_2 = \mathbf{0}, \mathcal{L}_N \mathbf{X}_3 = \mathbf{0}\}$$

Since the graph has a spanning tree, then consensus problem is solved with convergence value $(\mathbf{0}, \mathbf{0}, \alpha)$, where α is the convergence value for X_3 . The final consensus value α

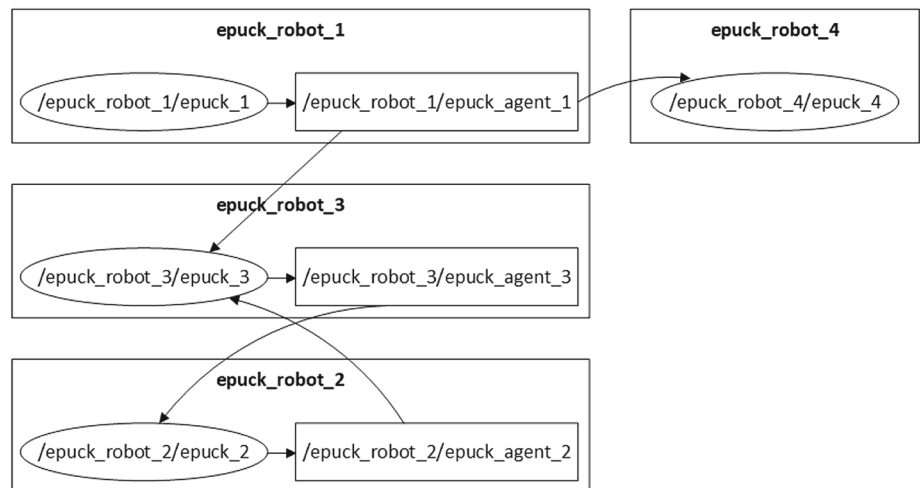
Fig. 5 Communication topology between e-pucks on ROS



(a) Leaderless case - An undirected graph



(b) Leaderless case - A directed graph



(c) Leader-follower case - A directed graph with leader

Fig. 6 The real states of e-pucks-Leaderless case for undirected graph

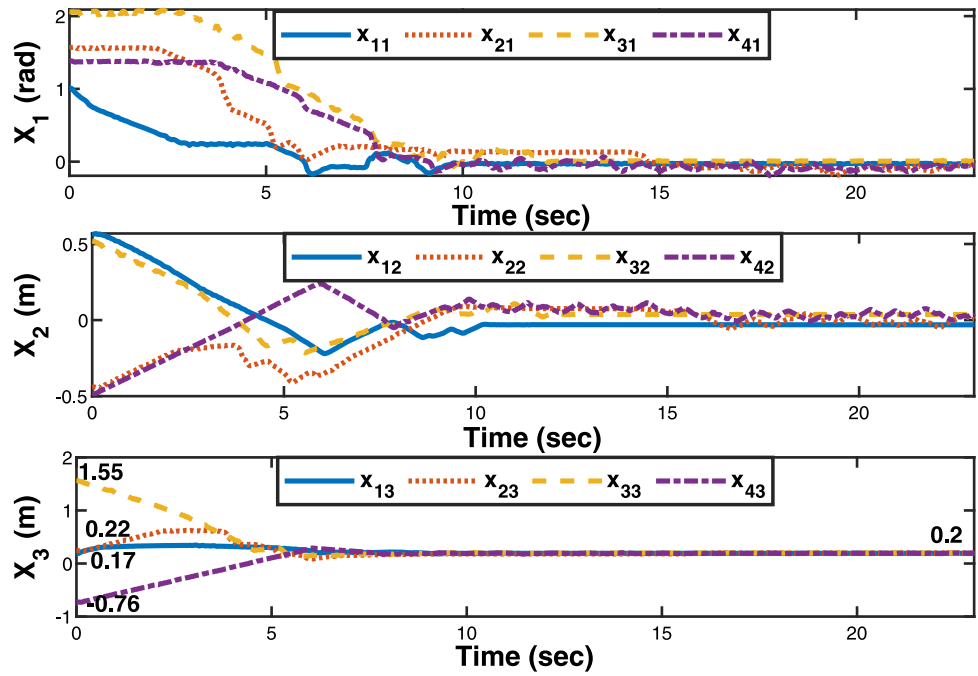
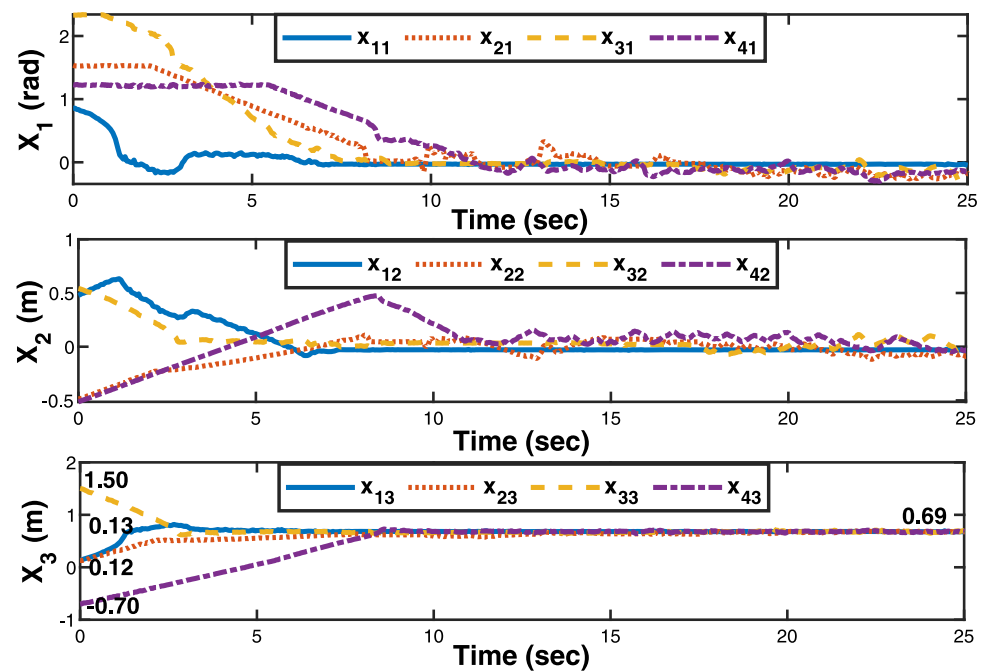


Fig. 7 The real states of e-pucks-Leaderless case for directed graph



depends on the initial states $X_3(0)$ and can be calculated by the following equation (Ren and Beard 2008):

$$\alpha = \frac{\gamma^T X_3(0)}{\gamma^T \mathbf{1}_N} \tag{15}$$

where γ is the left eigenvector of the Laplacian matrix associated with the zero eigenvalue. As a special case, when the graph is balanced or undirected and connected then the con-

vergence value of the state X_3 is the average of the initial state $X_3(0)$. \square

3.2 Leader-Follower Problem

Let us suppose that a network of N agents consists of a Leader and $N - 1$ followers, and the kinematic model for the leader is equal to the following:

Fig. 8 The real states of e-pucks-leader-follower case

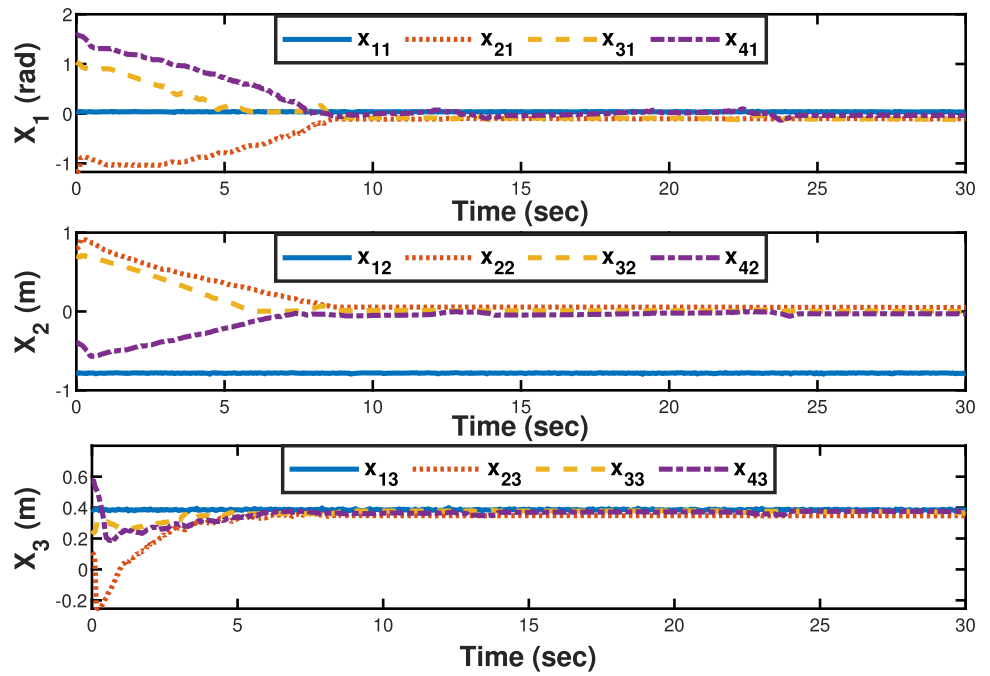
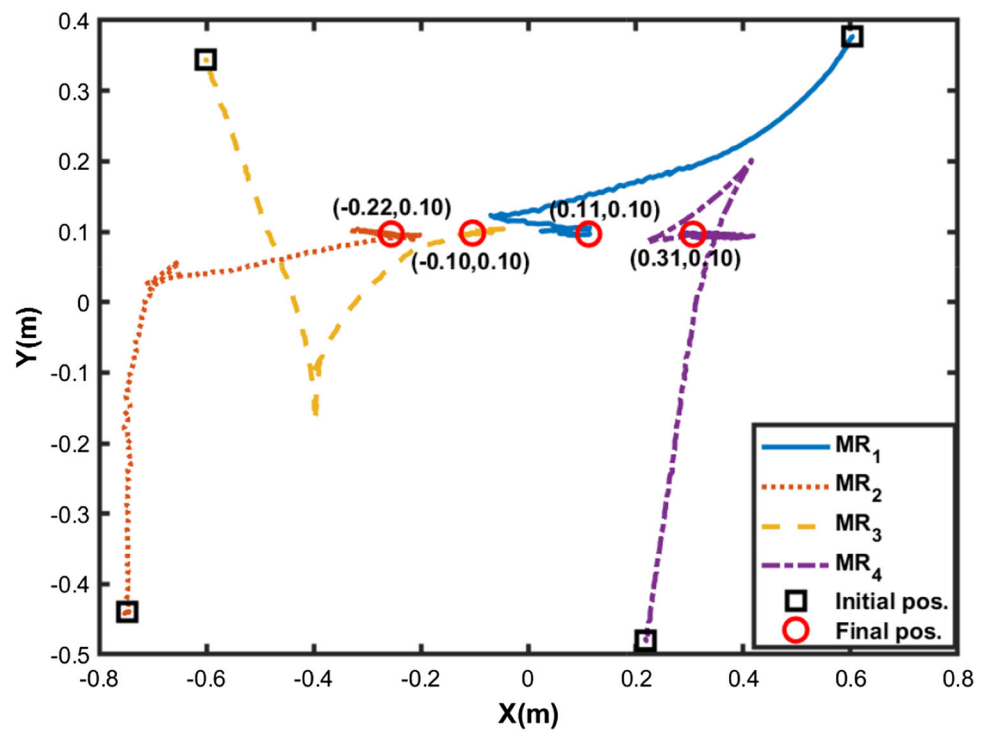


Fig. 9 Trajectories of the four e-puck robots-Leaderless case-undirected graph



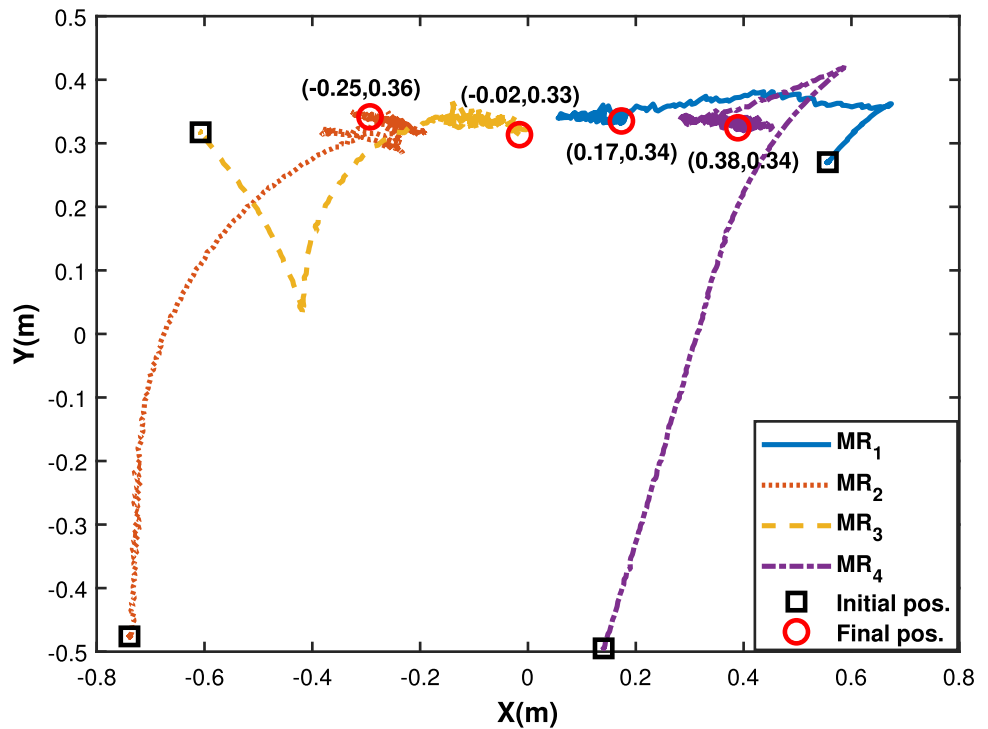
$$\begin{bmatrix} \dot{x}_{11} \\ \dot{x}_{12} \\ \dot{x}_{13} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$(16) \quad \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} [0; U_1] \\ [0; U_2] \\ [0; X_1 U_2 - X_2 U_1] \end{bmatrix} \quad (17)$$

With this consideration, equation (5) can be rewritten as:

where $X_1, X_2, X_3 \in \mathbb{R}^{N \times 1}$, $U_1, U_2 \in \mathbb{R}^{(N-1) \times 1}$ and $X_1, X_2 \in \mathbb{R}^{(N-1) \times (N-1)}$.

Fig. 10 Trajectories of the four e-puck robots-Leaderless case-directed graph



For leader–follower consensus problem, we propose the following distributed controller:

$$\begin{aligned}
 u_{i1} &= -k_1 x_{i1} - \frac{k_2 \sum_{j \in \mathcal{N}_i} a_{ij}(x_{j3} - x_{i3}) - a_{i1}(x_{13} - x_{i3})}{x_{i1}^2 + x_{i2}^2} x_{i2} \\
 u_{i2} &= -k_1 x_{i2} + \frac{k_2 \sum_{j \in \mathcal{N}_i} a_{ij}(x_{j3} - x_{i3}) - a_{i1}(x_{13} - x_{i3})}{x_{i1}^2 + x_{i2}^2} x_{i1}
 \end{aligned}
 \tag{18}$$

where a_{i1} is the connection weight constant between the leader and agent i (this weight is zero when the leader does not interact with it), whereas a_{ij}, k_1, k_2 are as mentioned in the leaderless case. In addition we assume that $x_i = (x_{i1}, x_{i2}, x_{i3}) \notin \mathbb{D}$.

We can write (18) in matrix form as in (13), where we define $S(x)$ in this case:

$$S(x) = -\mathcal{L}_{(N-1)} X_3 - \Gamma_{a_{i1}} (X_{13} - X_3)
 \tag{19}$$

where $\mathcal{L}_{(N-1)} \in \mathbb{R}^{(N-1) \times (N-1)}$ is the Laplacian matrix for the followers, $\Gamma_{a_{i1}} = \text{diag}\{a_{11}, \dots, a_{(N-1)1}\}$ and $X_{13} = [x_{13_i}] \in \mathbb{R}^{(N-1) \times 1}$ where

$$x_{13_i} = \begin{cases} x_{13} & (v_1, v_i) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

where \mathcal{E} is the set of edges, and v_1, v_i are the nodes which represent the leader and agent i in the communication graph, respectively.

By substituting (13) in (17) and taking into consideration (19), we get the closed-loop system:

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} -\mathbf{K}_1 X_1 + k_2 X_2 X_{12}^{-1} \mathcal{L}'_N X_3 \\ -\mathbf{K}_1 X_2 - k_2 X_1 X_{12}^{-1} \mathcal{L}'_N X_3 \\ -k_2 \mathcal{L}'_N X_3 \end{bmatrix}
 \tag{20}$$

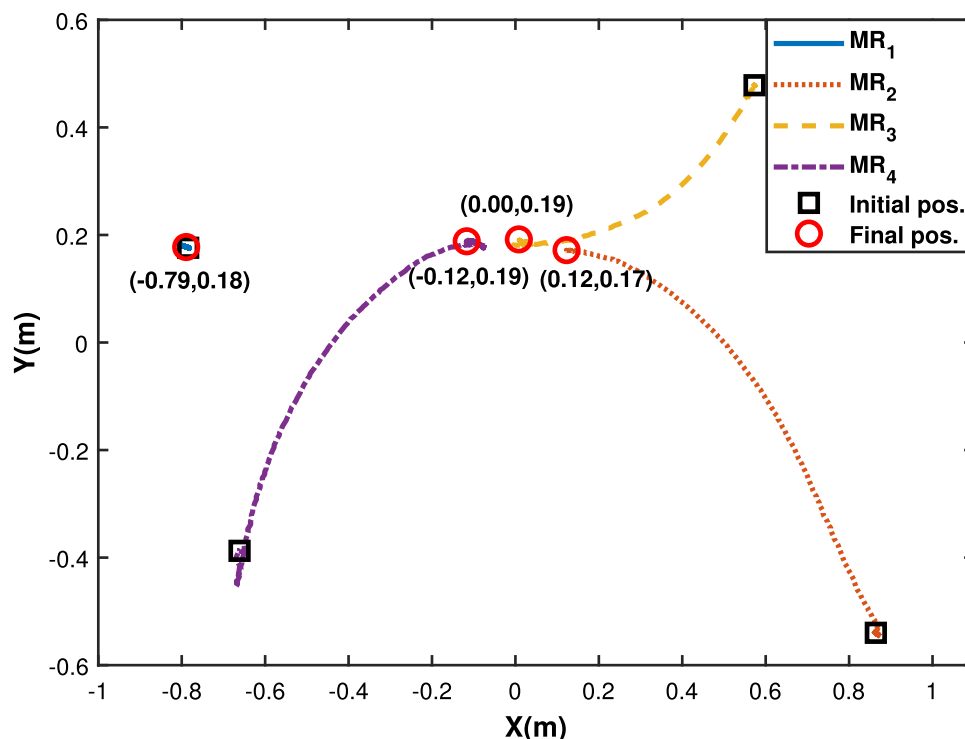
where \mathcal{L}'_N is the Laplacian matrix which describe the topology between the leader with the followers and between the followers, $\mathbf{K}_1 = \text{diag}\{0, k_1, \dots, k_1\} \in \mathbb{R}^{N \times N}$ where N is the number of agents.

Proposition Consider the system (17) in a directed topology which is consisted of a leader with $(N - 1)$ followers, and the control law (18) with the condition on the initial state of each follower $x_i \notin \mathbb{D}$. If the directed graph of this topology has a spanning tree and the root of this tree is the leader, then the state feedback control law (18) which is applied on each follower exponentially stabilizes system (17) and the agents reach consensus on the state X_3 to be equal to the leader's third state x_{13} .

Proof The proof of this proposition is almost the same of the proof of Proposition 2. By taking the same Lyapunov function which is positive definite while $x_i \notin \mathbb{D}$ and \mathcal{L}'_N has a spanning tree with the leader as the root, differentiating this function respect to time yields:

$$\dot{V} = -2X_1^T \mathbf{K}_1 X_1 - 2X_2^T \mathbf{K}_1 X_2 - 2k_2 X_3^T \mathcal{L}'_N \mathcal{L}'_N X_3$$

Fig. 11 Trajectories of the three followers–leader–follower case



which is negative definite. According to LaSalle invariant principle, the largest invariant set verifying $\dot{V} = 0$, to which the system converge, is:

$$\Omega = \{\mathbf{X}_1 = \mathbf{X}_2 = \mathbf{0}, \mathcal{L}'_N \mathbf{X}_3 = \mathbf{0}\}$$

Since the graph has a spanning tree and its root is the leader, then consensus problem is solved with convergence value equals to the leader's third state x_{13} .

In the next section, a design of test-bed is presented in order to test the proposed protocols which discussed earlier in this paper. \square

4 Design Experimental Test-Bed

4.1 Experimental Test-Bed Components

To build a distributed multi-agent system, a certain number of mobile robots, representing the agents, are needed, a localization device, in our case, a digital camera, to obtain the positions and orientations of the robots with respect to a reference frame in a specific area, and a personal computer to process the obtained images and send the poses to the mobile robots according to the predefined graph topology. In the following, these components are presented in detail.

4.1.1 Mobile Robot

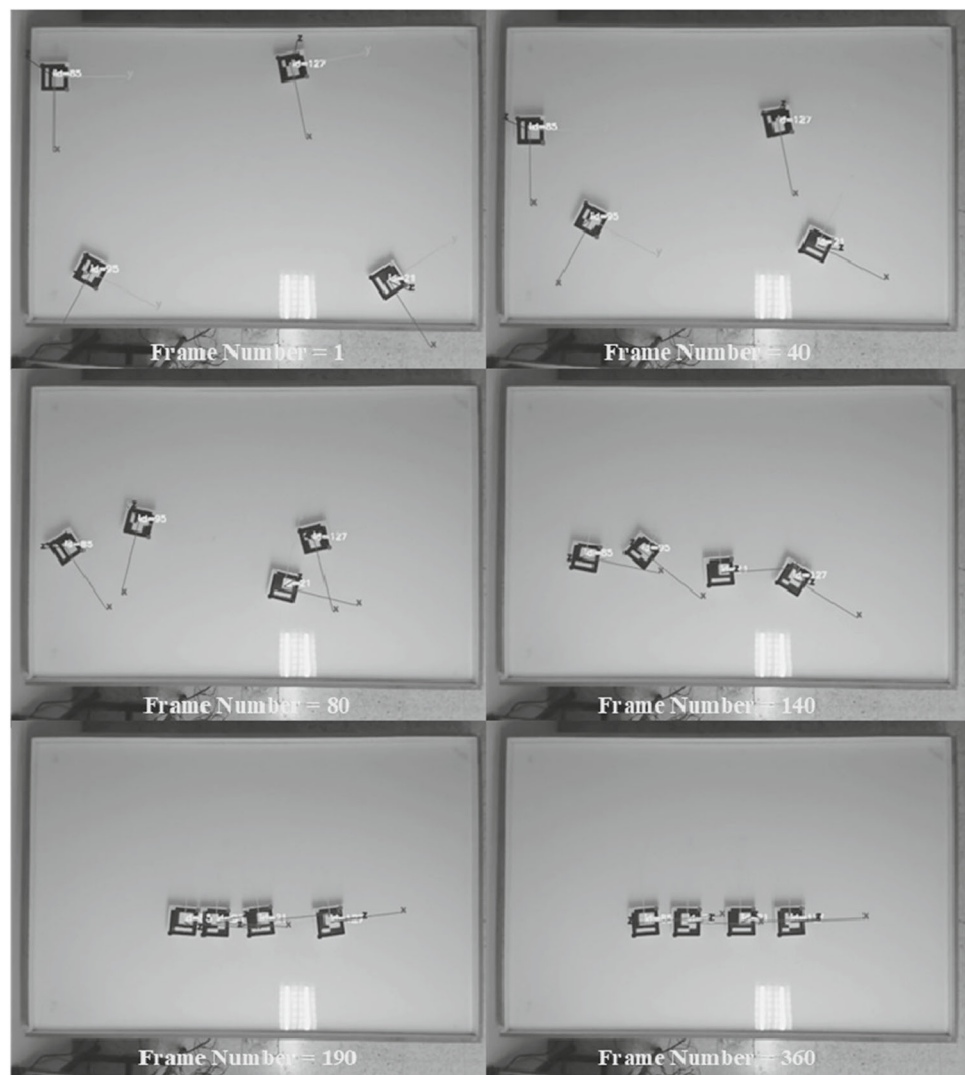
The robot which will be used in our test-bed should meet the following criteria: It has a small size with respect to environment bounds, can communicate with a computer and other robots wirelessly, has an open source hardware and software development model, and is available in a suitable number.

For accommodating the points aforementioned, e-puck mobile robot is used (Mondada et al 2009). E-puck is a differential mobile robot with two stepper motors, so it can be modeled as in (4). It has a diameter of 75 mm with a height which depends on the connected extensions. Its hardware and software are fully open source. It has a Bluetooth radio link to connect to a desktop computer or to communicate with up to seven e-pucks, and it can work in swarms (Mondada et al 2009). Using four e-pucks, a team of robots communicating with each other is formed, to be used in the validation of distributed consensus protocols.

4.1.2 Localization Device

The robots are moving in a rectangular arena of dimension $1.8 \text{ m} \times 1.2 \text{ m}$. In order to obtain robots' positions and orientations with respect to a reference frame, an ArUco marker is mounted on top of each robot. In addition, a fixed overhead camera hanging on the ceiling is used. The camera captures image frames at a specific rate. Those images are processed to get the poses of each marker in the image which will be sent to the computer via a USB cable.

Fig. 12 Different frames for moving robots-undirected graph



The camera chosen for this mission is a Logitech C270 camera fixed at a height of 2.15 m above the arena with a frame rate of 20 FPS and a resolution of 800×600 pixel. So, the working space captured by the camera covers the wanted area. An ArUco marker is a synthetic square marker composed by a wide black border and an inner binary matrix which defines a unique identifier that may include error detection and correction bits. There are several types of markers, each of them belongs to a dictionary which consists of a set of markers (Salinas 2019). In this work, the original ArUco dictionary with a marker size of 10 cm is used for a successful detection. Figure 2 shows an e-puck robot with an ArUco marker mounted on top.

4.1.3 Personal Computer and Operating System

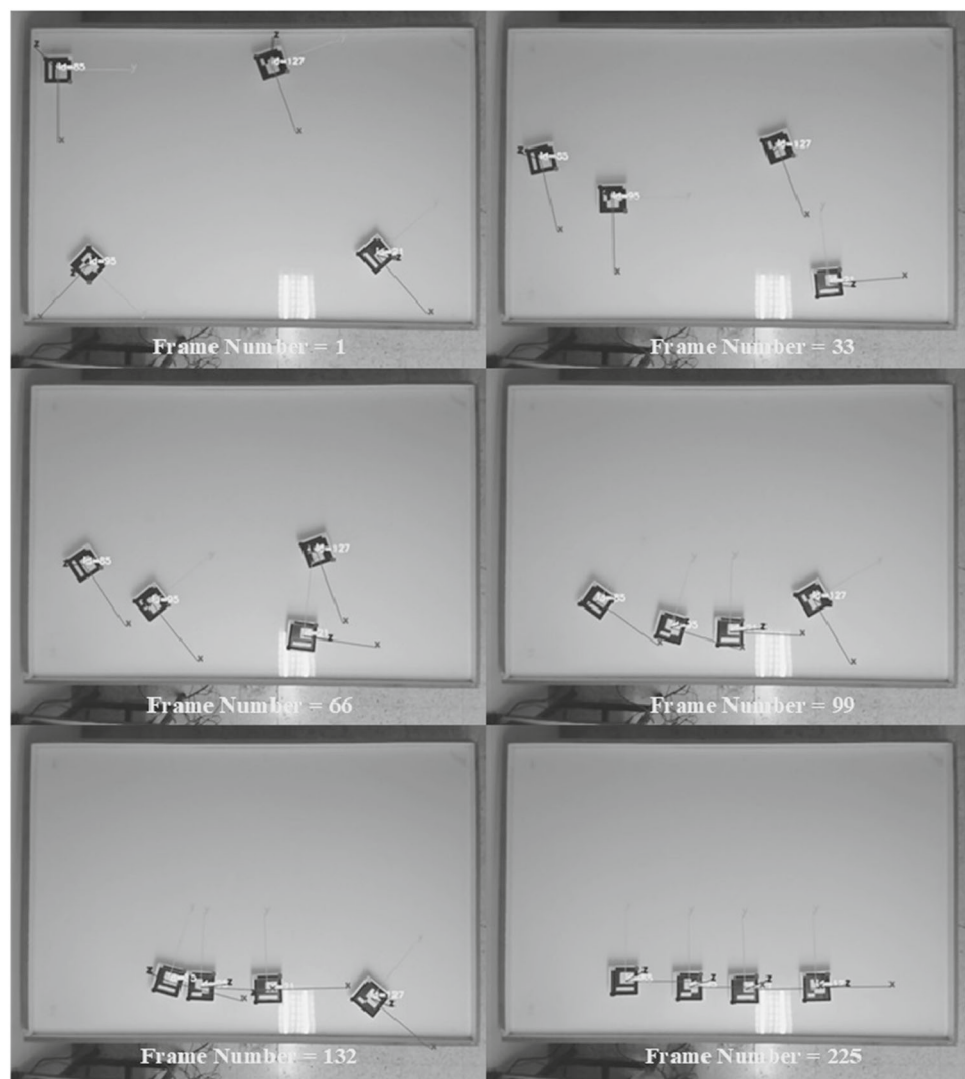
A personal computer (PC), acting as a bridge between the camera and mobile robots, extracts poses of the robots from

captured images to send them to the mobile robots. Table 1 shows the basic technical specifications of the computer which is used for this test-bed. ROS, which is an open-source meta-operating system was chosen as a software platform to achieve this task. The main supported operating system for ROS is Ubuntu (Joseph and Cacace 2018).

ROS is a collection of software packages that aim to reduce software complexity, and save developers time by supporting code reuse in robotics research and development. It is designed to be a distributed computing environment, where a number of components like robots and computers are networked to communicate with each other by passing messages, using a publisher and subscriber model (Joseph and Cacace 2018).

The ROS architecture has been designed and divided into three levels of concepts (Joseph and Cacace 2018):

Fig. 13 Different frames for moving robots-directed graph



- The filesystem level: In this level, a group of concepts is used to explain how ROS files are organized on the hard disk. The most basic unit of ROS are the ROS packages. They contain one or more programs (nodes), libraries, messages, and so on, which are organized together as a single unit.
- The computation graph level: is the peer-to-peer network of ROS processes that are processing data together. The main concepts in this level are ROS nodes, master, parameter server, messages, topics, services, and bags. Any node in the system can access this network, communicate with other nodes using messages which are transported using named buses called topics. The ROS master provides naming and registration services to the nodes in the ROS system. It tracks publishers and subscribers to topics. Without the master, nodes would not be able to find each other and exchange messages.

- The community level: which comprises a set of tools and concepts to share knowledge, algorithms, and code between developers.

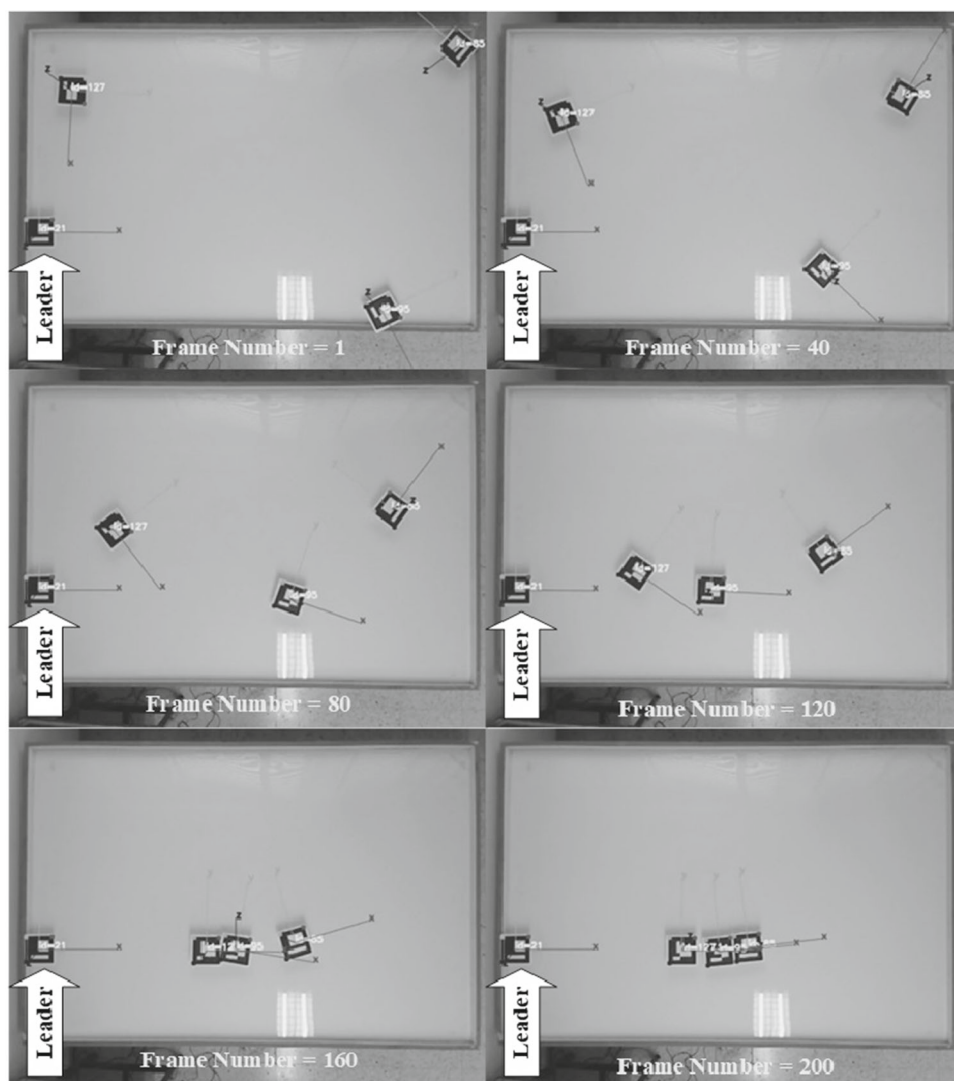
Figure 3 shows all the previous components together.

4.2 Application Structure on ROS

A ROS application structure resembles a graph with nodes as vertices, due to the possibility of breaking down the application into nodes, that can run independently, connect and communicate with each other by passing messages or making services calls.

Figure 4 depicts the ROS graph for the application which is designed in this paper. It shows the running nodes and the topics they use to communicate which are represented in ellipses and rectangles, respectively, and are detailed in the following:

Fig. 14 Different frames for moving robots–leader–follower case



- `usb_cam` node: It interfaces with standard USB cameras (e.g., the Logitech Quickcam) using `libusb_cam` and publishes images in a topic called `usb_cam/image_raw`. This node is contained in the package `usb_cam` which is sited in http://wiki.ros.org/usb_cam. The camera which is used in this application is calibrated using this node with ROS, and the resulting files are placed so ROS can find it and publish it.
- `aruco_single` node: It exists in the package `aruco_ros` which is a software package and a ROS wrapper of the ArUco Augmented Reality marker detector library, and it is placed in https://github.com/pal-robotics/aruco_ros. The `aruco_single` node subscribes the previous topic `usb_cam/image_raw` to process the obtained images for estimating the poses of markers in these images, and publishes a topic called `aruco_single/pose` linked to a specific marker. Consequently, this node has been edited to publish four topics with specific identification number

to be `aruco_single/pose i` , for $i = 1, 2, 3, 4$. Each one of these topics related to the pose of one marker that has been placed on a top of a specific e-puck robot.

- `epuck_robot` node: In this application, this node has been repeated four times to represent the four e-pucks in our configuration. In Fig. 4, there are four nodes with the name `epuck_robot i` , for $i = 1, 2, 3, 4$, each one of them has the same i number compared with the i number of the topic `aruco_single/pose i` , and is set to subscribe its pose and its neighbors' poses according to the predefined topology. Then, this information is sent from PC to the real e-puck via Bluetooth. The `epuck_robot` node is contained in the package `epuck_driver` which is sited in https://github.com/gctronic/epuck_driver_cpp.

Now, each robot receives its pose and its neighbors' pose based on Baudrate 115.2 kbit/s, and calculate its control signal to apply it on its motors and move toward achieving the

desired mission with frequency rate 20 Hz (depending on the frame rate of the used camera).

5 Experimental Validation

Along this section, a set of experiments and results are presented in order to evaluate the proposed architecture (see Sect. 4) by applying the proposed protocol (13) to solve rendezvous problem in two cases: leaderless case and leader–follower case.

Let the interaction graphs between four e-pucks as shown in Fig. 5, which models the information flow between the robots in two types for leaderless case, one of them is undirected connected graph (Fig. 5a) and the second is directed graph that has a spanning tree (Fig. 5b), while Fig. 5c depicts a directed graph for leader–follower case. Each one of the robots has a random initial state vector except the leader that has a static state with zero orientation. The protocol (13) is applied on the four robots to line up by a separation distance equals to 14 cm on X-axis with control parameters k_1 and k_2 are taken as $k_1 = 1$, $k_2 = 8$.

Figures 6 and 7 present the real results for the four e-pucks' states while applying the protocol (13) for leaderless case. Both Figs. 6 and 7 show that the system is stabilizable, and the state X_3 converge to an agreed-upon value which can be calculated using (15). It should be noted that there is a difference between the theoretical value and the real convergence value, and that could be leading to an error on robots' position on Y-axis in the plane. In this experiment, the error is equal to 5 cm for the undirected communication topology, and 7 cm for the directed communication topology.

Figure 8 presents the real states for leader–follower case (Fig. 5c). Also in this case, the system is stabilizable, and the convergence value of X_3 is the leader's state x_{13} . Figures 9, 10, and 11 show the robots' trajectories with clarifying the final position of them. Figures 12, 13, and 14 illustrate different frames taken from the aruco_single node. It shows that e-pucks move to have the zero orientation with the same position on Y-axis in order to form a line.

6 Conclusion

In this paper, a new control law based on an invariant manifold in order to solve consensus problem in a network of non-holonomic system is proposed. In this approach, agents share information about their state through a communication network such that to reach consensus on the state which is uncontrollable. Moreover, a design of experimental platform for studying and developing the distributed consensus protocols of multi-agent systems is presented. The implemented platform is composed of four e-pucks distinguished

by different ArUco markers, an overhead usb camera to determine the markers poses, and a PC with ROS framework to reduce software complexity, development time, and integration hurdles. Our proposed protocol is tested using the implemented platform on undirected and directed topology and for both cases: Leaderless and leader–follower. The protocol has shown a good performance when applied using the test-bed. The multi-agent system was stabilized and got rendezvous on a known point on the plane.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Antonelli, G., Arrichiello, F., & Chiaverini, S. (2009). Experiments of formation control with multirobot systems using the null-space-based behavioral control. *IEEE Transactions on Control Systems Technology*, 17(5), 1173–1182.
- Anvari, I. (2013). *Non-holonomic differential drive mobile robot control & design: Critical dynamics and coupling constraints* (Doctoral dissertation). Arizona State University.
- Astolfi, A. (1998). Discontinuous control of the Brockett integrator. *European Journal of Control*, 4(1), 49–63.
- Brockett, R. W. (1983). Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, 27(1), 181–191.
- Cao, K., Jiang, B., & Yue, D. (2014). Distributed consensus of multiple nonholonomic mobile robots. *IEEE/CAA Journal of Automatica Sinica*, 1(2), 162–170.
- Cortés, J., Martínez, S., & Bullo, F. (2006). Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8), 1289–1298.
- Cremean, L., Dunbar, W. B., van Gogh, D., Hickey, J., Klavins, E., Meltzer, J., & Murray, R. M. (2002). The Caltech multi-vehicle wireless testbed. In *Proceedings of the 41st IEEE conference on decision and control* (Vol. 1, pp. 86–88).
- Davids, A. (2002). Urban search and rescue robots: From tragedy to technology. *IEEE Intelligent Systems*, 17(2), 81–83.
- De Luca, A., & Oriolo, G. (1995). Modelling and control of non-holonomic mechanical systems. In *Kinematics and dynamics of multi-body systems* (pp. 277–342). Vienna: Springer.
- DeVon, D., & Bretl, T. (2007). Kinematic and dynamic control of a wheeled mobile robot. In *2007 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4065–4070). IEEE.
- Dimarogonas, D. V., & Kyriakopoulos, K. J. (2007). On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52(5), 916–922.
- Indriyanto, T., Rizki, A. R., Hariyadin, M. L., Akbar, M. F., & Syafi, A. A. (2020). Centralized swarming UAV using ROS for collaborative missions. In *AIP conference proceedings* (Vol. 2226, No. 1, p. 030012). AIP Publishing LLC.
- Jácóme, G., Sierra, M., & Cruz, P. J. (2019). A mini-sized agent testbed for applications in mobile robotics. In *2019 IEEE 4th Colombian conference on automatic control (CCAC)* (pp. 1–6). IEEE.

- Joseph, L., & Cacace, J. (2018). *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd.
- Khenouf, H., & De Wit, C. C. (1995). On the construction of stabilizing discontinuous controllers for nonholonomic systems. *IFAC Proceedings Volumes*, 28(14), 667–672.
- Lamping, A. P., Ouwerkerk, J. N., & Cohen, K. (2018). Multi-UAV control and supervision with ROS. In *2018 aviation technology, integration, and operations conference* (p. 4245).
- Lei, M., Zhou, S. L., Yang, X. X., & Yin, G. Y. (2012). Complex formation control of large-scale intelligent autonomous vehicles. *Mathematical Problems in Engineering*, 2012.
- Li, Z., & Duan, Z. (2017). *Cooperative control of multi-agent systems: A consensus region approach*. CRC Press.
- Li, Z., Duan, Z., Chen, G., & Huang, L. (2009). Consensus of multi-agent systems and synchronization of complex networks: A unified viewpoint. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(1), 213–224.
- Li, Z., Liu, X., Fu, M., & Xie, L. (2012). Global H consensus of multi-agent systems with Lipschitz non-linear dynamics. *IET Control Theory and Applications*, 6(13), 2041–2048.
- Maghenem, M., Bautista, A., Nuno, E., Loria, A., & Panteley, E. (2018). Consensus of multi-agent systems with nonholonomic restrictions via Lyapunov's direct method. *IEEE Control Systems Letters*, 3(2), 344–349.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., & Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions* (Vol. 1, No. CONF, pp. 59–65). IPCB: Instituto Politécnico de Castelo Branco.
- Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.
- Pitzer, B. *usb_cam*, http://wiki.ros.org/usb_cam
- Rehan, M., Jameel, A., & Ahn, C. K. (2017). Distributed consensus control of one-sided Lipschitz nonlinear multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(8), 1297–1308.
- Rehman, A., Rehan, M., Iqbal, N., & Ahn, C. K. (2018). Toward the LPV approach for adaptive distributed consensus of Lipschitz multi-agents. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(1), 91–95.
- Ren, W., & Beard, R. W. (2008). Distributed consensus in multi-vehicle cooperative control (Vol. 27, No. 2, pp. 71–82). London: Springer.
- Ren, W., & Atkins, E. (2007). Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 17(10–11), 1002–1033.
- Saber, R. O., & Murray, R. M. (2003). Consensus protocols for networks of dynamic agents.
- Salinas, R. M. (2019). *ArUco: An efficient library for detection of planar markers and camera pose estimation*.
- Salinas, R. M. *aruco_ros*, https://github.com/pal-robotics/aruco_ros.
- Tsiotras, P. (1997). Invariant manifold techniques for control of under-actuated mechanical systems. In *Modelling and control of mechanical systems* (pp. 277–292).
- Verlab Laboratory at Universidade Federal de Minas Gerais. https://github.com/gctronic/epuck_driver_cpp
- Zhai, G., Takeda, J., Imae, J., & Kobayashi, T. (2010). Towards consensus in networked non-holonomic systems [brief paper]. *IET Control Theory and Applications*, 4(10), 2212–2218.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.