Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-37 (2022) 548–553

Application of Distributed Linear Multi-Agent Containment Control to Robotic Systems ⋆

Stefan Tauchnitz * **Chengzhi Yuan** * **Paolo Stegagno** *

* *ICRobots lab, University of Rhode Island, Kingston, RI, USA (e-mail: stauchnitz@uri.edu; cyuan@uri.edu; pstegagno@uri.edu).*

Abstract: This article proposes the application of a distributed containment control algorithm to a team of mobile robots. This paper builds on the containment controller developed by Yuan et al. (2019) for generic linear multi-agent system and tested in simulation only. In this article, we particularize the controller for the case of multiple mobile robots by including it into a two-layer control scheme. The high-level controller computes a desired position for the mobile robots, that is then used as reference trajectory for the low-level controller. The resulting control system is implemented as a fully distributed system on a team of mobile robots and validated in simulations and experiments.

Keywords: Multi-agent systems, containment control, mobile robotics

1. INTRODUCTION

In recent years, advances in small computers as well as a growing field of potential applications lead to an increased research interest in distributed control algorithms for multi-agent systems. An overview on such problems, that include leaderless consensus control, formation control and coverage control, is provided by Zhu et al. (2016). The focus of this paper is on a specific problem for distributed control in multi-agent systems called containment control. It describes a control problem where a group of follower agents converges into the convex hull defined by the state of several leader agents (Xiao and Dong, 2021). One application of containment control is given by Wang et al. (2014) with a group of vehicles crossing a hazardous area where only some agents have the sensor ability to detect the hazards. The latter take the role of leaders and mark the safe area in which the followers must remain.

Since the introduction of the first containment control problem by Ji et al. (2008), many distributed solutions have been proposed for many different types of systems and operative conditions. A popular approach is to focus the study to the control of single and double integrator systems. For example, Li et al. (2012) propose a distributed containment controller using only the location of agents and not their velocity or acceleration. Zhang and Tang (2016) introduce dispersion behavior into the distributed containment controller. Similarly, group dispersion is used to avoid collisions in Zhang et al. (2013). Cao et al. (2010) introduce several distributed containment control algorithms for multiple stationary leaders as well as leaders with identical and different velocities. A special emphasis on function under disturbances is placed by Wang et al. (2014) that use a distributed observer as part of the distributed containment controller to estimate the weighted average of the leaders speed. In Dong et al. (2019) a group of robot leaders performs distributed formation control while the followers use a distributed containment control algorithm to stay within the convex hull spanned by the leaders. Santilli et al. (2021) consider a distributed containment control algorithm functioning in the presence of anonymous adversarial agents using time-varying graphs.

Other authors have proposed more general control laws designed to work on generic linear systems. In Li et al. (2015), only relative states and state estimates are used for the computation of the control input. In Bo et al. (2015) the system dynamics are limited to to the first and second order. Second order linear systems are discussed also by Fu et al. (2019) where the problem of input saturation is addressed by using sliding mode control and relative pose measurements. Z-Transforms are used to give sufficient conditions for distributed containment control. Fixed time communication delays are considered by Li et al. (2018). Qin et al. (2019) introduce a distributed containment controller for heterogeneous linear systems where even the dimension of the state can vary from agent to agent. An adaptive distributed observer is used by Liang et al. (2019) to enable distributed containment control for nonidentical networks with external disturbances. Yuan et al. (2019) develop a distributed containment controller for generic systems with heterogeneous and unknown linear dynamics.

Few papers deal with nonlinear dynamics. Wen et al. (2019) use a neural network approximator to estimate the non linear system dynamics. Xu et al. (2020) reduce communication between agents with a distributed event-triggered containment control algorithm. Xiao and Dong (2021) tackle the distributed containment control problem with a two layer approach in which the top layer does the containment control while the lower level performs fault-tolerant tracking control. Gu et al. (2021) propose a two-level cooperative control architecture paired with a Lyapunov analysis to achieve a containment formation.

⋆ This work was supported by the National Science Foundation under Grant CMMI 1952862.

10.1016/j.ifacol.2022.11.240

In most of the works mentioned above (Yuan et al., 2019; Xiao and Dong, 2021; Wang et al., 2014; Ji et al., 2008; Li et al., 2012; Zhang and Tang, 2016; Santilli et al., 2021; Li et al., 2015; Bo et al., 2015; Fu et al., 2019; Li et al., 2018; Qin et al., 2019; Liang et al., 2019; Wen et al., 2019; Xu et al., 2020) the proposed controllers are validated only in simulation, while only few papers provide experimental result on a robotic system. Gu et al. (2021) presented a team of two ground robots with bidirectional communication between the robots and virtual leaders. Ground robots are also used by Zhang et al. (2013) and Cao et al. (2010). Multirotor aerial vehicles are used in Dong et al. (2019), but the authors do not take advantage of the increased dimensionality and limit the problem to a bi-dimensional plane.

Another interesting aspect is the limitations on the leaders' movements assumed in several papers. For example, the leaders are required to remain stationary by Zhang and Tang (2016), Cao et al. (2010), and Bo et al. (2015), while leaders with identical input are required in Zhang et al. (2013) and Cao et al. (2010). A paper that assumes little limitations both on the motion of the leaders and on the system model is Yuan et al. (2019) which is not only applicable to generic linear systems but also does not require knowledge of the followers system matrices. Moreover, in general the model of the different follower robots can be different between robots.

In this paper, we present a distributed multi-robot implementation of the multi-agent containment control developed by Yuan et al. (2019). This will involve particularizing the controller for a suitable dynamics while verifying its assumptions, including it in a proper control scheme, and testing the resulting control system both in simulation and in real experiments. The validation of Yuan et al. (2019) on a real system, as well as its particularization for a multi-robot system, are the main contribution of this paper.

2. PROBLEM SETTINGS

The multi-robot system discussed in this paper consists of a group $\mathcal{F} = \{1, \ldots, N\}$ of N followers and a group $\mathcal{R} = \{N+1, \ldots, N+M\}$ of M virtual leaders. Each follower is a unicycle-type mobile robot with non-linear dynamics (Do, 2015):

$$\begin{bmatrix} \dot{\xi}_{1,i} \\ \dot{\xi}_{2,i} \\ \dot{\varphi}_i \end{bmatrix} = \begin{bmatrix} \sin\varphi_i \\ \cos\varphi_i \\ 0 \end{bmatrix} v_i + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_i, \tag{1}$$

where $\xi_i = [\xi_{1,i}\ \xi_{2,i}]^T \in \mathbb{R}^2$ and $\varphi_i \in SO(2)$ are respectively the global position and orientation of the i-th robot in a world frame of reference. The virtual leaders are a group of M virtual agents that exist in the same Cartesian space as the robots. We will indicate their state as $s_k \in \mathbb{R}^2, k \in \mathcal{R}$. The symbol $\mathrm{Co}(\mathcal{R}) = \mathrm{Co}(\{s_{N+1}, \ldots, s_{N+M}\})$ describes the complex hull spanned by the leaders $k \in \mathcal{R}$. An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is used to describe the communication among the followers. This graph consists in a set of vertices $\mathcal{V} = \{1, 2, \ldots, N\}$ representing the followers, and a set of edges $\mathcal{E} = \{(i,j)\}$, where $(i,j) \in \mathcal{E}$ if robot i can communicate with robot j. The adjacency matrix $\mathcal{A} \in \mathbb{R}^{N\times N}$ provides a matrix representation of $\mathcal{G}$ and is defined as $\mathcal{A} = [a_{ij}]$ with $a_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise, including if $i = j$. The Laplacian matrix $\mathcal{L} \in \mathbb{R}^{N\times N}$ of $\mathcal{G}$ is defined as $\mathcal{L} = [I_{ij}]$ with $I_{ii} = \sum_{j\neq i} a_{ij}$ and $I_{ij} = -a_{ij}$ if $i \neq j$. The communication between leader $k \in \mathcal{R}$ and follower $i \in \mathcal{F}$ is marked with a weight of $\delta_i^k = 1$ in $\Delta_k = diag\{\delta_1^k, \ldots, \delta_N^k\}\ \forall k \in \mathcal{R}$ with $\Delta_k \in \mathbb{R}^{N\times N}$. Conversely, $\delta_i^k = 0$ if there is no communication between leader $k \in \mathcal{R}$ and follower $i \in \mathcal{F}$. In general, the communication graph $\mathcal{G}$ and the communication matrices Δ_k are assumed constant for the whole duration of the containment control task. The goal of this control algorithm is to achieve containment control as defined in Qin et al. (2019) as: $\lim_{t\to\infty} \mathrm{dist}\,(x_i(t), \mathrm{Co}(\mathcal{R})) = 0,\ \forall i \in \mathcal{F}$. Our solution to this problem is an application of the containment controller for multi-agent systems presented by Yuan et al. (2019). In the following, I describes the identity matrix of arbitrary dimension, O describes the zero matrix of arbitrary dimension, and $\mathbb{S}_+^n$ denotes the sets of symmetrical and positive definite $n \times n$ matrices.

3. BACKGROUND

3.1 System Descriptions

Yuan et al. (2019) propose a containment control algorithm applicable to linear multi-agent systems with the following characteristics. N heterogeneous followers are each described with the following linear uncertain system model:

$$\dot{x}_i = A_i x_i + B_i u_i \quad \forall i \in \mathcal{F}, \tag{2}$$

where the unknown system matrix $A_i \in \mathbb{R}^{n\times n}$ and the known input matrix $B_i \in \mathbb{R}^{n\times n_{u,i}}$ are constant. $x_i \in \mathbb{R}^n$ is the state of the i-th follower, and $u_i \in \mathbb{R}^{n_{u,i}}$ the control input. There are M homogeneous leaders with the generic k-th leader described as:

$$\dot{s}_k = A_0 s_k + B_0 r_k \quad \forall k \in \mathcal{R}, \tag{3}$$

with constant and known system matrices $A_0 \in \mathbb{R}^{n\times n}$ and $B_0 \in \mathbb{R}^{n\times n_{u,r}}$. $s_k \in \mathbb{R}^n$ is the k-th leaders' state and $r_k \in \mathbb{R}^{n_r}$ the bounded input signal. The input signal r_k is measurable for followers neighboring the leader as indicated in the Δ_k matrix. The following linear system generates the leaders input:

$$\dot{r}_k = A_r r_k \quad \forall k \in \mathcal{R} \tag{4}$$

where $A_r \in \mathbb{R}^{n_r\times n_r}$ is constant.

3.2 Assumptions

Yuan et al. (2019) state several assumptions regarding the system dynamics and communication among the agents.

Assumption 1: There exist constant matrices $K_{1i} \in \mathbb{R}^{n\times n_{u,i}}$ and $K_{2i} \in \mathbb{R}^{n_r\times n_{u,i}}$, such that $A_0 = A_i + B_i K_{1i}^T$ and $B_0 = B_i K_{2i}^T\ \forall i \in \mathcal{F}$.

Assumption 2: (A_0, B_0) is stabilizable, and the leaders' input signals r_k are bounded, i.e., $||r_k|| \leq r_k^*\ \forall k \in \mathbb{R}$, where r_k^* are positive constants.

Assumption 3: The interaction graph $\mathcal{G}$ among the follower agents is undirected and connected. There is at least one follower that each leader has a directed path to.

3.3 Control Algorithm

The error signal used in the controller is defined as:

$$e_i = \sum_{j=1}^{N} a_{ij}\,(x_j - x_i) + \sum_{k=N+1}^{N+M} \delta_i^k\,(s_k - s_i) \quad \forall i \in \mathcal{F}. \quad (5)$$

This distributed observer based adaptive control protocol is proposed by Yuan et al. (2019):

$$\dot{\hat{r}}_i = A_r \hat{r}_i + L\left[\sum_{j=1}^{N} a_{ij}\,(\hat{r}_j - \hat{r}_i) + \sum_{k=N+1}^{N+M} \delta_j^k\,(r_k - \hat{r}_i)\right] \quad (6)$$

$$\dot{\hat{K}}_{1i} = \gamma x_i e_i^T P B_i \quad (7)$$

$$u_i = \hat{K}_{1i}^T x_i + K_{2i}^T \hat{r}_i + K_{2i}^T K_3 e_i, \quad (8)$$

where $\hat{r}_i \in \mathbb{R}^{n_r}$ is the distributed observer state of the leaders' input signals r_k, $L \in \mathbb{R}^{n_r \times n_r}$ is a control coefficient, and $\hat{K}_{1i} \in \mathbb{R}^{n \times n_{u,i}}$ is the estimation of K_{1i} $\forall i \in \mathcal{F}$ and is influenced by the control coefficients $\gamma \in \mathbb{R}_+$ and $P \in \mathbb{S}_+^n$. The followers' control input u_i is dependent on K_{2i} which can be derived from **Assumption 1**, and on $K_3 \in \mathbb{R}^{n_r \times n}$ which is another coefficient. These four coefficients are designed according to the following equation (9).

3.4 LMI Condition to Determine the Controller Coefficients

Equations (6)-(8) implement an observer for the leader's input as well as a control algorithm for the follower robots that depends on several parameters. These must be properly selected to achieve the desired containment control behavior. In particular any $\gamma > 0$ can be selected, while the following Linear Matrix Inequality (LMI) needs to be fulfilled for all followers in order to obtain $\lim_{t\to\infty} e_i(t) = 0$ according to Yuan et al. (2019):

$$\begin{bmatrix} A_0\hat{P} + \hat{P}A_0^T - \lambda_i(\mathcal{H})(B_0\hat{K}_3 + \hat{K}_3^T B_0^T) & B_0 \\ B_0^T & QA_r + A_r^T Q - \lambda_i(\mathcal{H})(\hat{L} + \hat{L}^T) \end{bmatrix} < 0 \quad (9)$$

where $\lambda_i(\mathcal{H})$ are the eigen-values of the matrix $\mathcal{H} = \sum_{k=N+1}^{N+M} \frac{1}{M}\mathcal{L} + \Delta_k$, and the variables are the positive definite matrices $\hat{P} \in \mathbb{S}_+^n$ and $Q \in \mathbb{S}_+^{n_r}$ as well as the rectangular matrices $\hat{K}_3 \in \mathbb{R}^{n_r \times n}$ and $\hat{L} \in \mathbb{R}^{n_r \times n_r}$. The controller coefficients are calculated as: $P = \hat{P}^{-1}$, $K_3 = \hat{K}_3\hat{P}^{-1}$, $L = Q^{-1}\hat{L}$. A proof for the stability of this controller is provided by Yuan et al. (2019).

4. PARTICULARIZATION

The controller presented above has been developed for linear systems, however our robots have nonlinear dynamics. Its application to our system can be done through the control system architecture presented in Section 1, that presents the control system running on each robot. It is structured as a two layer system, in which a reference trajectory is generated for the robots through the developed controller. At this aim, the i-th robot communicates with its communication neighbors to obtain their state and their control inputs (for the leaders only). The generated trajectory is then used as a reference signal for an Input/Output Controller (IOC) that generates the linear velocity v_i and angular velocity ω_i for the robot as described by Siciliano (2009). Using the error $\varepsilon_i = x_i - \bar{\xi}_i$, the desired velocity $\mu_i = k_i \cdot \varepsilon_i$ is calculated, where k_i is a positive proportional gain. The linear and angular velocities for the robots are calculated using their orientation

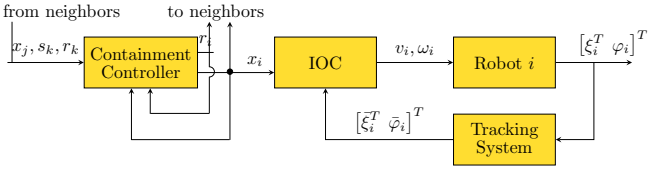

Fig. 1. Control system running on robot i

$$\bar{\varphi}_i\text{:} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos(\bar{\varphi}_i) & \sin(\bar{\varphi}_i) \\ -\sin(\bar{\varphi}_i)/b & \cos(\bar{\varphi}_i)/b \end{bmatrix} \times \mu_i,$$

where $b > 0$ is a parameter.

The implementation of the containment controller to generate the reference signals first required the selection of a linear system that would respect **Assumptions 1-3**, and would lead to a solvable LMI (9). For both the leaders and followers, we picked a simple integrator dynamics by selecting $A_0 = O, B_0 = I, A_r = O, A_i = O, B_i = b_i I$, where $b_i > 0$ is a parameter that can be selected to improve the system performance. The resulting system fulfills **Assumption 1** with $K_{1i}^T = O$ and $K_{2i}^T = I$. **Assumption 2** is fulfilled with B_0 being the identity matrix and the controllability matrix therefore having full rank i.e., the system (A_0, B_0) is stabilizable. **Assumption 3** is a condition on the communication graph, therefore is not affected by the system matrices. It will be fulfilled later in the simulations and experiments sections. Note that choice of A_i is only limited to linear systems, but is not limited to an integrator dynamics. In Section 7 we will discuss how it can be used as a parameter of the system to improve the closed loop behavior.

With these choices the LMI from (9) is reduced to:

$$\begin{bmatrix} -\lambda_i(\mathcal{H})(B_0\hat{K}_3 + \hat{K}_3^T B_0^T) & B_0 \\ B_0^T & -\lambda_i(\mathcal{H})(\hat{L} + \hat{L}^T) \end{bmatrix} < 0 \quad (10)$$

Choosing $A_r = O$ leads to $\dot{r}_k = 0$ which means that the leaders have a constant velocity. This limitation in general is not given in Yuan et al. (2019) but caused by the chosen system dynamics. In Section 6 we will show that slow changes in velocity do not affect the stability and behavior of the controller.

The implementation of the controller on our mobile robots required a time discrete version of it. We start with the discrete system dynamics at time step m of length Δt:

$$x_{i,m} = A_{i,d}x_{i,m-1} + \Delta t B_{i,d} u_{i,m-1} \quad (11)$$

$$s_{k,m} = A_{0,d}s_{k,m-1} + \Delta t B_{0,d} r_{k,m-1} \quad (12)$$

with the discrete input matrices $B_{i,d} \in \mathbb{R}^{n \times n_{u,i}} : B_{i,d} = B_i = I$ and $B_{0,d} \in \mathbb{R}^{n \times n_r} : B_{0,d} = B_0 = I$, the discrete follower's system matrix $A_{i,d} \in \mathbb{R}^{n \times n} : A_{i,d} = I + \Delta t A_i = I$, and the discrete leader's system matrix $A_{0,d} \in \mathbb{R}^{n \times n} : A_{0,d} = I + \Delta t A_0 = I$. The k-th leader input is given as $r_{k,m} = A_{r,d} v_{k,m}$. with a separate algorithm supplying the input $v_{k,m} \in \mathbb{R}^{n_r}$. As above, $A_{r,d} = I + \Delta t A_r = I$. The discrete error signal is defined similar as in (5):

$$e_{i,m} = e_{i,m-1} + \sum_{j=1}^{N} a_{ij}\,(x_{j,m} - x_{i,m}) + \sum_{k=N+1}^{N+M} \delta_i^k\,(s_{k,m} - s_{i,m}) \quad \forall i \in \mathcal{F}. \quad (13)$$

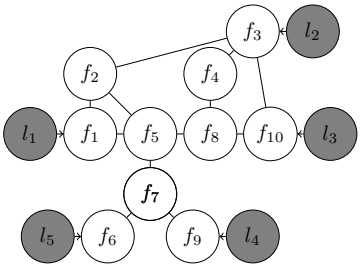

Fig. 2. Communication graph in the simulation with the leaders l_k and the followers f_i.

The continuous control algorithm described in equations (6)-(8) is discretized as:

$$\hat{r}_{i,m} = \hat{r}_{i,m-1} + \frac{1}{\Delta t} L \left[\sum_{j=1}^{N} a_{ij} \left(\hat{r}_{j,m-1} - \hat{r}_{i,m-1} \right) + \sum_{k=N+1}^{N+M} \delta_i^k \left(r_{k,m-1} - \hat{r}_{i,m-1} \right) \right] \tag{14}$$

$$\hat{K}_{1i,m} = \hat{K}_{1i,m-1} + \frac{1}{\Delta t} \gamma x_{i,m} e_{i,m-1}^T P B_i \tag{15}$$

$$u_{i,m} = \hat{K}_{1i,m}^T x_{i,m} + K_{2i}^T \hat{r}_{i,m} + K_{2i}^T K_3 e_{i,m}, \tag{16}$$

where the controller coefficients P, K_3, and L are the same as the continuous case.

5. VALIDATION IN SIMULATION

To validate the proposed controller we implemented a distributed simulation in Gazebo using five virtual leaders and ten simulated unicycle-style robots with the dynamics described in (1) as followers. The communication graph is shown in Figure 2, which respects **Assumption 3**. The controller coefficients P, K_3 and L are obtained as $P = \begin{bmatrix} 0.1818 & 0 \\ 0 & 0.1818 \end{bmatrix}$, $K_3 = \begin{bmatrix} 1.6120 & 0 \\ 0 & 1.6120 \end{bmatrix}$, $L = \begin{bmatrix} 1.6545 & 0 \\ 0 & 1.6545 \end{bmatrix}$ by solving the LMI condition in (10). The simulation was performed with the parameter $b_i = 1$, so that the matrix $B_i = I$. The five virtual leaders move each with a different velocity of $v_{1,m} = [0.0225\ 0.07]^T\ m/s$, $v_{2,m} = [-0.0075\ 0.05]^T\ m/s$, $v_{3,m} = [-0.0125\ 0.01]^T\ m/s$, $v_{4,m} = [0.004\ 0.03]^T\ m/s$, $v_{5,m} = [-0.015\ 0.025]^T\ m/s$ Figure 3 shows the simulated trajectories for leaders, and followers. Colored solid lines represent the actual trajectories ξ_i of the robots while the colored dashed lines represent the desired trajectories x_i. Black dashed lines represent the trajectory s_k of the five virtual leaders. The gray dotted and dash dotted lines mark the convex hull spanned by the leaders at the start and end of the simulation at $t = 0s$ and $t = 38s$. From the plot, we can observe that all followers starting outside the convex hull spanned by the leaders ended within it and all followers end up following their set point trajectory. This is corroborated also by the plot of the distance d_i between each robot and the convex hull $\mathrm{Co}(\mathcal{R})$ spanned by the leaders reported in Figure 4. Since a negative value of d_i indicates that robot i is within $\mathrm{Co}(\mathcal{R})$, from the plot it is possible to observe that all robots eventually converge to and remain in the convex hull.

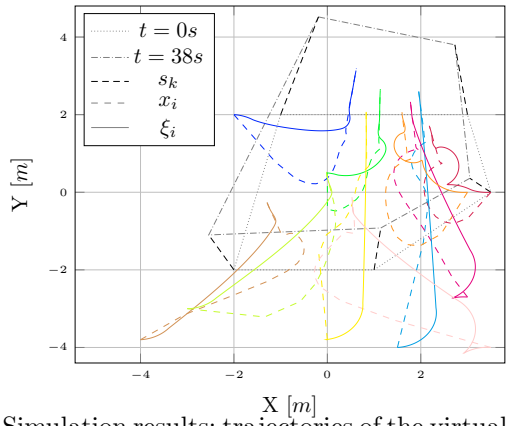

Fig. 3. Simulation results: trajectories of the virtual leaders s_k (black, dashed), the reference trajectories of the followers x_i (colored, dashed) and the robots trajectories ξ_i (colored, solid). The convex hull at time $t = 0s$ (gray, dotted) and at time $t = 38s$ (gray, dash-dotted) are also shown.

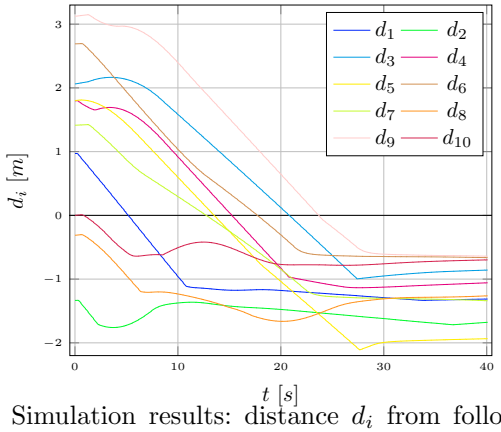

Fig. 4. Simulation results: distance d_i from followers to the convex hull; negative distances indicate that the followers within the hull.

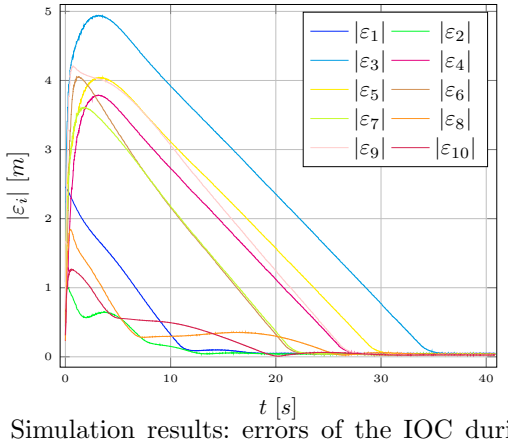

Fig. 5. Simulation results: errors of the IOC during the simulation with $b_i = 1$.

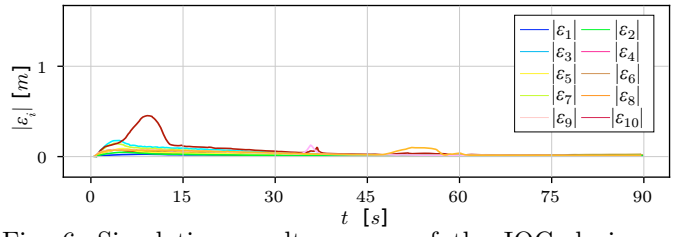

Fig. 6. Simulation results: errors of the IOC during a simulation with $b_i = 0.015$.

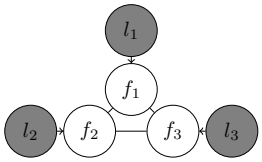

Fig. 7. Communication graph in the experiments with the leaders l_k and the followers f_i.

It must be noted that at the beginning of the simulation the reference signals move fast compared to maximum velocity that the robots can exert, causing a significant error in the IOC. This is visible in Figure 5 that shows the tracking error $|\varepsilon_i|$ for each follower over time t, with a sharp increase of the errors at the beginning followed by a steady decrease. This is due to the fact that most robots starts relatively far from the convex hull $(1-3m)$. Therefore the containment controller must recover a large initial error that is subsequently translated into error for the IOC. It is possible to reduce the parameter b_i so that the reference points will not move too fast for the robots. We performed the same simulation with $b_i = 0.015$, and the corresponding tracking errors, provided in Figure 6, are much smaller and mostly due to the initial difference between the orientation of the robots and the direction of motion of the reference points. It must be noted that in this case the Cartesian trajectories of the reference points for the followers (omitted for brevity) are very similar to the simulation with $b_i = 1$, however, the robots take longer to converge to the convex hull as they are moving slower.

6. VALIDATION IN EXPERIMENT

For an experimental implementation a system of three virtual leaders and three $\sim 20cm$ differential drive mobile robots was used. The robots are equipped with an arduino Romeo board to perform the low-level control tasks and compute the odometry, and an ODROID-XU4 for high-level control tasks and communication through a Wifi module. The robots move in a $10m \times 10m$ area equipped with an Optitrack motion capture system that provides the ground truth.

The communication graph is depicted in Figure 7. The corresponding controller coefficients computed by solving the LMI in (10) are: $P = \begin{bmatrix} 0.1818 & 0 \\ 0 & 0.1818 \end{bmatrix}$, $K_3 = \begin{bmatrix} 1.5597 & 0 \\ 0 & 1.5597 \end{bmatrix}$, $L = \begin{bmatrix} 1.5848 & 0 \\ 0 & 1.5848 \end{bmatrix}$. The virtual leaders move with a velocity of

$$v_{1,m} = \begin{bmatrix} 0.075\cos(t/20 + 3/2\pi) + 0.0125 \\ 0.075\sin(t/20 + 3/2\pi) + 0.0125 \end{bmatrix} m/s \quad (17)$$

$$v_{2,m} = \begin{bmatrix} 0.075\cos(t/20 + \pi) + 0.0125 \\ 0.075\sin(t/20 + \pi) + 0.0125 \end{bmatrix} m/s \quad (18)$$

$$v_{3,m} = \begin{bmatrix} 0.075\cos(t/20) + 0.0125 \\ 0.075\sin(t/20) + 0.0125 \end{bmatrix} m/s. \quad (19)$$

This choice of velocities creates a triangular convex hull that rotates about its center of mass and slowly drifts with a linear motion, as visible in Figure 8. The fully distributed controller was executed by the ODROID-XU4 on each robot. Communication between the robots and the calculation of the virtual leader's position was managed by a ground station computer hosting a shared ROS master node. Each follower only subscribed to and received information from neighboring robots as described by the communication graph in Figure 7.

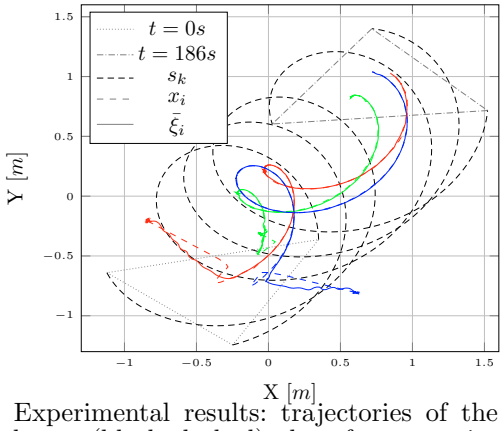

Fig. 8. Experimental results: trajectories of the virtual leaders s_k (black, dashed), the reference trajectories of the followers x_i (colored, dashed) and the real robots trajectories $\bar{\xi}_i$ (colored, solid). The convex hull at time $t = 0s$ (gray, dotted) and at time $t = 186s$ (gray, dash-dotted) are also shown.

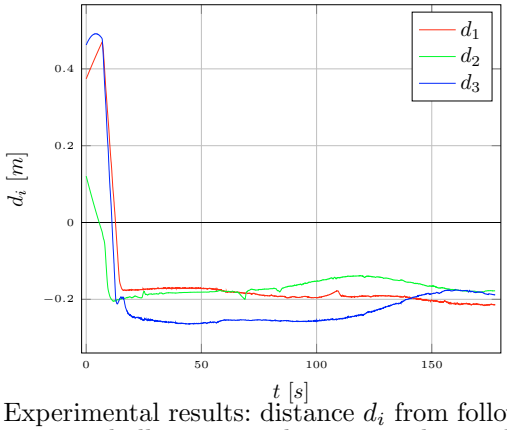

Fig. 9. Experimental results: distance d_i from followers to the convex hull; negative distances indicate that the followers are within the hull.

Figure 8 shows the measured trajectory $\bar{\xi}_i$ (colored, solid lines), the reference trajectories computed by the containment controller x_i (colored, dashed) and the position of the virtual leaders s_k (black, dashed). As in simulation, the plot shows that the followers reach and stay within the convex hull spanned by the virtual leaders plotted for $t = 0s$ and $t = 186s$ (gray, dotted and dot-dashed respectively). Notably, the tracking error of the IOC in this case is lower with respect to the simulation. This happens because the robots start from a configuration that is closer to the convex hull. Therefore the reference trajectories move slower with respect to the simulation case. The distance d_i of robot i to the convex hull is plotted in Figure 9 with negative values indicating the robot being inside the convex hull. A clip of the experiment can be seen in the accompanying video.

7. CONCLUSION

This paper shows a mobile robot application of the distributed containment controller introduced in Yuan et al. (2019) and its validation in simulation with ten robots and experiments with three robots. In general, the controller showed the expected behavior being able to solve the

containment control and drive the robots inside the convex hull. This applies both in case that the virtual leaders move according to **Assumption 1**, that allows for each leader to move with a different constant velocity (simulation), and also when that assumption is violated as shown in the experiment where a circular motion component is added to the velocity of the leaders. However, it is also evident from the plots in Figure 5 that a limited robot velocity can cause the tracking error in the lower level IOC to increase significantly at the beginning of the control task if the robots start in a configuration that is far from the convex hull. One possible solution to this problem is to "slow down" the dynamics of the reference trajectories by changing the b_i parameter with a proper adaptive law.

REFERENCES

Bo, L., Zeng-qiang, C., Zhong-xin, L., and Qing, Z. (2015). Containment control of discrete-time multi-agent systems with multiple stationary leaders and time-delays. In *2015 34th Chinese Control Conference (CCC)*, 7062–7066. IEEE, Hangzhou, China. doi:10.1109/ChiCC.2015.7260756.

Cao, Y., Stuart, D., Ren, W., and Meng, Z. (2010). Distributed containment control for double-integrator dynamics: Algorithms and experiments. In *Proceedings of the 2010 American Control Conference*, 3830–3835. IEEE, Baltimore, MD. doi:10.1109/ACC.2010.5531204.

Do, K. (2015). Global output-feedback path-following control of unicycle-type mobile robots: A level curve approach. *Robotics and Autonomous Systems*, 74, 229–242. doi:10.1016/j.robot.2015.07.019.

Dong, X., Hua, Y., Zhou, Y., Ren, Z., and Zhong, Y. (2019). Theory and Experiment on Formation-Containment Control of Multiple Multirotor Unmanned Aerial Vehicle Systems. *IEEE Transactions on Automation Science and Engineering*, 16(1), 229–240. doi:10.1109/TASE.2018.2792327.

Fu, J., Wan, Y., Wen, G., and Huang, T. (2019). Distributed Robust Global Containment Control of Second-Order Multiagent Systems With Input Saturation. *IEEE Transactions on Control of Network Systems*, 6(4), 1426–1437. doi:10.1109/TCNS.2019.2893665.

Gu, N., Peng, Z., Wang, D., and Zhang, F. (2021). Path-Guided Containment Maneuvering of Mobile Robots: Theory and Experiments. *IEEE Transactions on Industrial Electronics*, 68(8), 7178–7187. doi:10.1109/TIE.2020.3000120.

Ji, M., Ferrari-Trecate, G., Egerstedt, M., and Buffa, A. (2008). Containment Control in Mobile Networks. *IEEE Transactions on Automatic Control*, 53(8), 1972–1975. doi:10.1109/TAC.2008.930098.

Li, B., Yang, H.Y., Chen, Z.Q., and Liu, Z.X. (2018). Distributed Containment Control of Multi-agent Systems with General Linear Dynamics and Time-delays. *International Journal of Control, Automation and Systems*, 16(6), 2718–2726. doi:10.1007/s12555-017-0696-8.

Li, J., Ren, W., and Xu, S. (2012). Distributed Containment Control with Multiple Dynamic Leaders for Double-Integrator Dynamics Using Only Position Measurements. *IEEE Transactions on Automatic Control*, 57(6), 1553–1559. doi:10.1109/TAC.2011.2174680.

Li, Z., Duan, Z., Ren, W., and Feng, G. (2015). Containment control of linear multi-agent systems with multiple leaders of bounded inputs using distributed continuous controllers: Containment of multi-agent systems. *International Journal of Robust and Nonlinear Control*, 25(13), 2101–2121. doi:10.1002/rnc.3195.

Liang, H., Zhou, Y., Ma, H., and Zhou, Q. (2019). Adaptive Distributed Observer Approach for Cooperative Containment Control of Nonidentical Networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(2), 299–307. doi:10.1109/TSMC.2018.2791513.

Qin, J., Ma, Q., Yu, X., and Kang, Y. (2019). Output Containment Control for Heterogeneous Linear Multi-agent Systems With Fixed and Switching Topologies. *IEEE Transactions on Cybernetics*, 49(12), 4117–4128. doi:10.1109/TCYB.2018.2859159.

Santilli, M., Franceschelli, M., and Gasparri, A. (2021). Dynamic Resilient Containment Control in Multirobot Systems. *IEEE Transactions on Robotics*, 1–14. doi:10.1109/TRO.2021.3057220.

Siciliano, B. (ed.) (2009). *Robotics: modelling, planning and control*. Advanced textbooks in control and signal processing. Springer, London.

Wang, X., Li, S., and Shi, P. (2014). Distributed Finite-Time Containment Control for Double-Integrator Multiagent Systems. *IEEE Transactions on Cybernetics*, 44(9), 1518–1528. doi:10.1109/TCYB.2013.2288980.

Wen, G., Wang, P., Huang, T., Yu, W., and Sun, J. (2019). Robust Neuro-Adaptive Containment of Multi-leader Multiagent Systems With Uncertain Dynamics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(2), 406–417. doi:10.1109/TSMC.2017.2722042.

Xiao, S. and Dong, J. (2021). Distributed Fault-Tolerant Containment Control for Nonlinear Multi-Agent Systems Under Directed Network Topology via Hierarchical Approach. *IEEE/CAA Journal of Automatica Sinica*, 8(4), 806–816. doi:10.1109/JAS.2021.1003928.

Xu, T., Lv, G., Duan, Z., Sun, Z., and Yu, J. (2020). Distributed Fixed-Time Triggering-Based Containment Control for Networked Nonlinear Agents Under Directed Graphs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(10), 3541–3552. doi:10.1109/TCSI.2020.2991101.

Yuan, C., Zeng, W., and Dai, S.L. (2019). Distributed model reference adaptive containment control of heterogeneous uncertain multi-agent systems. *ISA Transactions*, 86, 73–86. doi:10.1016/j.isatra.2018.11.003.

Zhang, H., Zhao, Z., Meng, Z., and Lin, Z. (2013). Experimental Verification of a Multi-robot Distributed Control Algorithm with Containment and Group Dispersion Behaviors: The Case of Dynamic Leaders. *IFAC Proceedings Volumes*, 46(13), 153–158. doi:10.3182/20130708-3-CN-2036.00077.

Zhang, W. and Tang, Y. (2016). Consensus control for agent networks with stationary leaders. In *2016 Chinese Control and Decision Conference (CCDC)*, 241–245. IEEE, Yinchuan, China. doi:10.1109/CCDC.2016.7530988.

Zhu, B., Xie, L., and Han, D. (2016). Recent developments in control and optimization of swarm systems: A brief survey. In *2016 12th IEEE International Conference on Control and Automation (ICCA)*, 19–24. IEEE, Kathmandu, Nepal. doi:10.1109/ICCA.2016.7505246.