APPLICATION OF DISTRIBUTED CONTAINMENT CONTROL TO MULTI-ROBOT SYSTEMS

Stefan Tauchnitz
University of Rhode Island, stauchnitz@uri.edu

Follow this and additional works at: https://digitalcommons.uri.edu/theses

Recommended Citation
Tauchnitz, Stefan, "APPLICATION OF DISTRIBUTED CONTAINMENT CONTROL TO MULTI-ROBOT SYSTEMS" (2021). Open Access Master's Theses. Paper 2087.
https://digitalcommons.uri.edu/theses/2087

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons@etal.uri.edu.
APPLICATION OF DISTRIBUTED CONTAINMENT CONTROL TO MULTI-ROBOT SYSTEMS

BY

STEFAN TAUCHNITZ

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2021
MASTER OF SCIENCE THESIS

OF

STEFAN TAUCHNITZ

APPROVED:

Thesis Committee:

Major Professor  Paolo Stegagno
Richard J. Vaccaro
Chengzhi Yuan
Brenton DeBoef

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2021
This thesis proposes the application of a distributed containment control algorithm to a team of mobile robots. The containment controller this thesis builds on [1] was developed for generic linear multi-agent system and tested in simulation only. In this thesis, I 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. Additionally the containment controller is tested in a multi-layer control scheme in which the leaders perform encircling control and the followers perform the proposed containment control.

Parts of this thesis were submitted to the IEEE Robotics and Automation Letters with option of presenting at the IEEE International Conference on Robotics and Automation 2022.
ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my thesis supervisor, Paolo Stegagno, whose encouragement and guidance were a constant source of inspiration and motivation. His patience and willingness to explain and question were indispensable.

I would also like to thank my committee members Richard Vaccaro and Chengzhi Yuan. Their unhesitant willingness to participate in the defense of this thesis on such a short notice and their insightful suggestions. For her immediate agreement to chair my committee, I would like to thank Sigrid Berka.

I would like to thank Thivanka Perera for the robots I used, as well as his support for computer, programming, and the other problems that had come up throughout the duration of my work.

For their constant companionship and their heartfelt welcome, I would like to thank my friends and housemates in the TI and IEP Houses. During months of online classes and long hours simulating and programming, they provided welcomed distractions and diversions.

Finally, I want to express my gratitude to my family, friends, and girlfriend for their warmhearted and consistent support and encouragement. From the fledgling idea of studying in Rhode Island to this day, they supported me with good advice, friendship, and their company.
# TABLE OF CONTENTS

ABSTRACT ................................................................. ii

ACKNOWLEDGMENTS ................................................... iii

TABLE OF CONTENTS ................................................... iv

LIST OF FIGURES ...................................................... vi

CHAPTER

1 Introduction ......................................................... 1

2 Literature Review ................................................... 3

3 Problem Setting ...................................................... 6

4 Methodology .......................................................... 8
   4.1 Background Containment Control ................................. 8
       4.1.1 System Descriptions ....................................... 8
       4.1.2 Assumptions on Agent Dynamics and Communication
            Topology .................................................. 9
       4.1.3 Control Algorithm ....................................... 9
       4.1.4 LMI Condition to Determine the Controller Coefficients . 10
   4.2 Containment Control Particularization ......................... 10

5 Simulation ............................................................ 14
   5.1 Containment Controller ........................................ 14
       5.1.1 Setup in Gazebo .......................................... 14
       5.1.2 Simulation of Ten Followers and Five Leaders .......... 14
5.2 Encircling Controller .......................................................... 19
  5.2.1 Problem Setting Encircling Controller ............................... 19
  5.2.2 Background Encircling Control ...................................... 20
  5.2.3 Particularization Encircling Control ................................. 21
  5.2.4 Simulation of Three Robots with a Moving Center ............... 22
  5.3 Multi-Layer Formation Control .......................................... 24

6 Experiment .............................................................................. 28
  6.1 Simulation Setup ............................................................... 28
    6.1.1 Hardware ................................................................... 28
    6.1.2 Nonholonomic Constraints ......................................... 28
  6.2 Experiment of Containment Control with Three Followers and Three Virtual Leaders .................................................. 30

7 Conclusion and Future Work .................................................... 36

LIST OF REFERENCES ................................................................. 38

BIBLIOGRAPHY ........................................................................... 41
| Figure | Description |
|--------|-------------|
| 1      | Control system for the containment controller running on robot $i$. |
| 2      | Simulated robot used in Gazebo with two controlled wheels (dark gray) and an omnidirectional caster wheel (blue). |
| 3      | Communication graph in the simulation with the leaders $l_k$ and the followers $f_i$. |
| 4      | 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 $\xi_i$ (colored, solid). The convex hull at time $t = 0s$ (gray, dotted) and at time $t = 38s$ (gray, dash-dotted) are also shown. |
| 5      | Simulation results: distance $d_i$ from followers to the convex hull; negative distances indicate that the followers are within the hull. |
| 6      | Simulation results: errors of the IOC during the simulation. |
| 7      | Control system for the encircling controller running on robot $i$. |
| 8      | Simulation results: trajectories of the virtual center and the encircling robots. |
| 9      | Simulation results: error of the radius $\rho_i$ of the output of the encircling controller. |
| 10     | Simulation results: error of the phase $\phi_i$ of the output of the encircling controller. |
| 11     | Communication graph in the simulation with the leaders $l_k$ controlled by an encircling controller and the followers $f_i$. The node $b$ is the base station. The communication between the leaders for the encircling controller is shown by the dotted arrows, the communication between leaders and followers for the containment controller by the normal arrows. |
| Figure | Page |
|--------|------|
| 12     | Simulation results: trajectories of the virtual center, the leaders performing encircling control and followers performing containment control during the beginning of the simulation where all robot’s converge to their desired trajectories. .................. 27 |
| 13     | Simulation results: trajectories of the virtual center, the followers performing encircling control and followers performing containment control during the later part of the simulation. ... 27 |
| 14     | Robot used for all experiments with two actuated wheels and a caster. ................................................................. 29 |
| 15     | Communication graph in the experiments with the leaders $l_k$ and the followers $f_i$. .................................................. 30 |
| 16     | 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 $\xi_i$ (colored, solid). The convex hull at time $t = 0s$ (gray, dotted) and at time $t = 186s$ (gray, dash-dotted) are also shown. ......... 32 |
| 17     | Experimental results: distance $d_i$ from followers to the convex hull; negative distances indicate that the followers are within the hull. ................................................................. 32 |
| 18     | Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 0s$ where all robots have not reached their desired location. ......................... 33 |
| 19     | Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 4s$ where all robots have reached their desired location but not a useful orientation. 33 |
| 20     | Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 20s$ where all robots have reached their desired location and oriented themselves along their trajectories. .......................... 34 |
| 21     | Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 52s$ where all robots start performing the first turn. ................................. 34 |
| Figure | Page |
|--------|------|
| 22 | Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 84s$ where all robots are in the tightest part of the turn. | 35 |
| 23 | Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 116s$ where all robots have left the tight turn. | 35 |
CHAPTER 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-robot systems. The multi-robot systems discussed in this thesis consist of several mobile robots communicating with each other while moving in space. A variety of coordination problems have been proposed over time which can be solved by a group of mobile robots [2]. In formation control the goal is that different robots stay at a fixed relative position to its neighbors. Coverage control is concerned with the goal of optimizing the coverage of an area with the lowest number of robots or cover the largest area possible with a given number or robots. Another common problem is consensus control where the goal is for all robots to reach the same state. In most papers this is discussed under constrains like high communication latency or changing communication networks. The main focus of this thesis 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 [3].

One exemplary application of containment control is given in [4] 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.

In this thesis I will place an emphasis on distributed multi-robot containment control. Here I will show a widely applicable controller introduced in [1] and particularize it for the problem of mobile robots. Additionally I will give a brief
introduction into an encircling controller proposed in [5]. Afterwards I will show several simulations and experiments to validate the function of the containment controller as well as the interaction of the two in a multi layer formation controller. This thesis will be concluded by a discussion of open questions and interesting avenues for future work.
CHAPTER 2

Literature Review

Since the introduction of the first containment control problem in [6], 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, in [7] a distributed containment controller using
only the location of agents and not their velocity or acceleration is proposed. The
authors of [8] introduce dispersion behavior into the distributed containment con-
troller. Similarly, group dispersion is used to avoid collisions in [9]. In [10] several
distributed containment control algorithms for multiple stationary leaders as well
as leaders with identical and different velocities are introduced. A special emphasis
on function under disturbances is placed in [4] where a distributed observer is used
as part of the distributed containment controller to estimate the weighted average
of the leaders’ speed.

In [11] 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. The authors of [12] consider a distributed con-
tainment 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 [13], only relative states and relative state estimates are
used for the computation of the control input. In [14] the system dynamics are
limited to the first and second order. Second order linear systems are discussed
also in [15] 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 delays for the communication are considered in [16]. The authors of [17] 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 in [18] to enable distributed containment control for nonidentical networks with external disturbances. In [1], the authors develop a distributed containment controller for generic systems with heterogeneous and unknown linear dynamics.

Few papers deal with nonlinear dynamics. The authors of [19] use a neural network approximator to estimate the non linear system dynamics. [20] reduces communication between agents with a distributed event-triggered containment control algorithm. In [3], the authors 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. In [21], a two-level cooperative control architecture is proposed to achieve a containment formation, and paired with a lyapunov analysis.

In most of the works mentioned above [1, 3, 4, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 19, 20] the proposed controllers are validated only in simulation, while only few papers provide experimental result on a robotic system. In [21] a team of two ground robots is used with bidirectional communication between the robots and virtual leaders. Ground robots are also used in [9] and [10]. Multirotor aerial vehicles are used in [11], 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 in [8], [10], and [14], while leaders with identical input are required in [9] and
[10]. A paper that assumes little limitations both on the motion of the leaders and on the system model is [1] which is not only applicable to generic linear systems but also does not require knowledge of the followers system matrix. Moreover, in general the model of the different follower robots can be different between robots.
CHAPTER 3
Problem Setting

The multi-robot system discussed in this thesis 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 [22]:

\[
\begin{bmatrix}
\dot{\xi}_{1,i} \\
\dot{\xi}_{2,i} \\
\dot{\phi}_i
\end{bmatrix} =
\begin{bmatrix}
\cos \phi_i \\
\sin \phi_i \\
0
\end{bmatrix} v_i
+ \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} \omega_i
\]

(1)

where \( \xi_i = [\xi_{1,i}, \xi_{2,i}]^T \in \mathbb{R}^2 \) and \( \phi_i \in SO(2) \) are respectively the global position and orientation of the \( i \)-th robot in a world frame of reference.

The virtual leaders for containment control are a group of \( M \) virtual agents that exist in the same Cartesian space as the robots. I will indicate their state as \( s_k \in \mathbb{R}^2, k \in \mathcal{R} \). The symbol \( \text{Co}(\mathcal{R}) = \text{Co}(\{s_{N+1}, \ldots, s_{N+M}\}) \) describes the convex 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. 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 \( \delta^k_i = 1 \) in \( \Delta_k = \text{diag}\{\delta^1_i, \ldots, \delta^N_i\} \forall k \in \mathcal{R} \) with \( \Delta_k \in \mathbb{R}^{N \times N} \). Conversely, \( \delta^k_i = 0 \) if there is no communication between leader \( k \in \mathcal{R} \) and follower \( i \in \mathcal{F} \).

The goal of the containment control algorithm is to achieve containment con-
trol as defined in [17] as:

$$\lim_{t \to \infty} \text{dist} \left( x_i(t), \text{Co}(\mathcal{R}) \right) = 0, \ \forall i \in \mathcal{F}. \quad (2)$$

My solution to this problem is an application of the containment controller for multi-agent systems presented in [1].

In the rest of this document I will use the following symbols. $I$ 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.
CHAPTER 4

Methodology

4.1 Background Containment Control

In this section I provide for completeness an overview on the setup and main findings of [1].

4.1.1 System Descriptions

The authors of [1] 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:

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

(3)

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:

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

(4)

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 $\Delta_k$ matrix.

The following linear system generates the leaders input:

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

(5)

where $A_r \in \mathbb{R}^{n_r \times n_r}$ is constant.
4.1.2 Assumptions on Agent Dynamics and Communication Topology

The authors of [1] state several assumptions regarding the system dynamics as well as 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 $A_0 = A_i + B_iK_{1i}^T$ and $B_0 = B_iK_{2i}^T \forall i \in \mathcal{F}$.

**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. Moreover, there is at least one follower that each leader has a directed path to it.

4.1.3 Control Algorithm

The error signal used in the controller is defined as:

$$e_i = \sum_{j=1}^{N} a_{ij} (x_j - x_i) + \sum_{k=N+1}^{N+M} \delta_k^i (s_k - x_i) \quad \forall i \in \mathcal{F}.$$  \hspace{1cm} (6)

This distributed observer based adaptive control protocol is proposed by [1]:

$$\dot{\hat{r}}_i = A_i \hat{r}_i$$  \hspace{1cm} (7)

$$+ L \left[ \sum_{j=1}^{N} a_{ij} (\hat{r}_j - \hat{r}_i) + \sum_{k=N+1}^{N+M} \delta_k^i (r_k - \hat{r}_i) \right]$$

$$\dot{\hat{K}}_{1i} = \gamma x_i e_i^T P B_i$$  \hspace{1cm} (8)

$$u_i = \hat{K}_{1i}^T x_i + K_{2i}^T \hat{r}_i + K_{3i}^T e_i.$$  \hspace{1cm} (9)

$\hat{r}_i \in \mathbb{R}^{n_r}$ is the distributed observer state of the leader’s input signals $r_k$. $L \in \mathbb{R}^{n_r \times n_r}$ is a controller coefficient. $\hat{K}_{1i} \in \mathbb{R}^{n \times n_{u,i}}$ is the estimation of $K_{1i} \forall i \in \mathcal{F}$ and is influenced by the controller 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 controller coefficient. These four coefficients are designed according to the following equation (10).

4.1.4 LMI Condition to Determine the Controller Coefficients

Equations (7)-(9) 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 [1]:

$$\begin{bmatrix} A_0 \hat{P} + \hat{P} A_0^T - \lambda_i(\mathcal{H})(B_0 \hat{K}_3 + \hat{K}_3^T B_0^T) & B_0 \\ B_0^T & QA_r + A_r^T Q - \lambda_i(\mathcal{H})(\hat{L} + \hat{L}^T) \end{bmatrix} < 0$$ (10)

where $\lambda_i(\mathcal{H})$ are the eigen-values of the matrix

$$\mathcal{H} = \sum_{k=N+1}^{N+M} \frac{1}{M} \mathcal{L} + \Delta_k,$$ (11)

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:

$$P = \hat{P}^{-1}$$ (12)

$$K_3 = \hat{K}_3 \hat{P}^{-1}$$ (13)

$$L = Q^{-1} \hat{L}.$$ (14)

A proof for the stability of this controller can be found in [1].

4.2 Containment Control 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 Figure 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 \(\omega_i\) for the robot as described in [23]. 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 \(\bar{\varphi}_i\):

\[
\begin{bmatrix}
    v_i \\
    \omega_i
\end{bmatrix} =
\begin{bmatrix}
    \cos(\bar{\varphi}_i) & \sin(\bar{\varphi}_i) \\
    -\sin(\bar{\varphi}_i)/b & \cos(\bar{\varphi}_i)/b
\end{bmatrix} \times \mu_i
\]  

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 (10). For both the leaders and followers, I picked a simple integrator dynamics by selecting \(A_0 = O, B_0 = I, A_r = ...
$O, A_i = O, B_i = I$. 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 it 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 chapter 7 I 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 (10) is reduced to:

$$\begin{bmatrix}
-\lambda_i(H)(B_0\hat{K}_3 + \hat{K}_3^T B_0^T) & B_0 \\
B_0^T & -\lambda_i(H)(\hat{L} + \hat{L}^T)
\end{bmatrix} < 0$$  \tag{18}$$

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 [1] but caused by the chosen system dynamics. In section 6.2 I 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. I start with the discrete system dynamics at time step $m$ of length $\Delta t$:

$$x_{i,m} = A_{i,d}x_{i,m-1} + \Delta t B_{i,d}u_{i,m-1}$$  \tag{19}$$

$$s_{k,m} = A_{0,d}s_{k,m-1} + \Delta t B_{0,d}r_{k,m-1}$$  \tag{20}$$

with the discrete input matrices $B_{i,d} \in \mathbb{R}^{n \times n_{u,i}} : B_{i,d} = B_i = I$ and $B_{0,d} \in \mathbb{R}^{n \times n_c} : B_{0,d} = B_0 = I$, 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

$$r_{k,m} = A_{r,d}v_{k,m}.$$  \tag{21}$$
with a separate algorithm supplying the input $u_{k,m} \in \mathbb{R}^{n_r}$. As above, $A_{r,d} = I + \Delta t A_r = I$. The discrete error signal is defined similar as in (6):

$$e_{i,m} = e_{i,m-1} + \sum_{j=1}^{N} a_{ij} (x_{j,m} - x_{i,m})$$

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

The continuous control algorithm described in equations (7)-(9) is discretized as:

$$\hat{r}_{i,m} = \hat{r}_{i,m-1} + \frac{1}{\Delta t} L \left[ \sum_{j=1}^{N} a_{ij} (\hat{r}_{j,m-1} - \hat{r}_{i,m-1}) ight. \right.$$  

$$+ \left. \sum_{k=N+1}^{N+M} \delta_i^k (r_{k,m-1} - \hat{r}_{i,m-1}) \right]$$

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

$$u_{i,m} = \hat{K}_{1i,m} x_{i,m} + K_{2i}^{T} \hat{r}_{i,m} + K_{3}^{T} K_{3} e_{i,m},$$

where the controller coefficients $P$, $K_3$, and $L$ are the same as the continuous case.
CHAPTER 5

Simulation

5.1 Containment Controller
5.1.1 Setup in Gazebo

To have a realistic simulation of the controller I used the open-source 3D robotics simulator Gazebo. As the controllers discussed in this thesis do not use any exeroceptive sensors, many functions like the simulation of a lidar or the measurements of a camera are not used. Important for this thesis is the simulation of communication between different robots with ROS. In so called topics, robots can publish data which can then be received by neighboring robots subscribing to the topics. Other topics are used to communicate the ground truth location of the robots. The simulated robots are differential drive robots with two independently actuated wheels and an omnidirectional caster wheel as shown in Figure 2. The simulated environment consists of a simple infinite plane.

5.1.2 Simulation of Ten Followers and Five Leaders

To validate the proposed controller I 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 selected communication graph shown in Figure 3 has the following Laplacian matrix $\mathcal{L}$:

$$
\mathcal{L} = \begin{bmatrix}
2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & -1 & 2 & 0 & 0 & 0 & -1 & 0 & 0 \\
-1 & -1 & 0 & 0 & 4 & 0 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 3 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 & -1 & 0 & 0 & 3 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 2
\end{bmatrix}
$$

(26)
Figure 2: Simulated robot used in Gazebo with two controlled wheels (dark gray) and an omnidirectional caster wheel (blue).

with the leader-follower weights

\[ \Delta_1 = \text{diag}\{1, 0, 0, 0, 0, 0, 0, 0, 0\} \]
\[ \Delta_2 = \text{diag}\{0, 0, 1, 0, 0, 0, 0, 0, 0\} \]
\[ \Delta_3 = \text{diag}\{0, 0, 0, 0, 0, 0, 0, 0, 1\} \]
\[ \Delta_4 = \text{diag}\{0, 0, 0, 0, 0, 0, 0, 0, 1\} \]
\[ \Delta_5 = \text{diag}\{0, 0, 0, 0, 0, 1, 0, 0, 0\}. \] (27)

Note that the selected communication graph respects Assumption 3. The controller coefficients \( P, K_3 \) and \( L \) are obtained as

\[ P = \begin{bmatrix} 0.1818 & 0 \\ 0 & 0.1818 \end{bmatrix} \] (28)
\[ K_3 = \begin{bmatrix} 1.6120 & 0 \\ 0 & 1.6120 \end{bmatrix} \] (29)
\[ L = \begin{bmatrix} 1.6545 & 0 \\ 0 & 1.6545 \end{bmatrix} \] (30)

by solving the LMI condition in (18).

The five virtual leaders move each with a different velocity of \( v_{1,m} = \)
Figure 3: Communication graph in the simulation with the leaders $l_k$ and the followers $f_i$.

\[
\begin{align*}
[0.0225 \ 0.07]^T \text{ m/s}, & \quad v_{2,m} = [-0.0075 \ 0.05]^T \text{ m/s}, & \quad v_{3,m} = [-0.0125 \ 0.01]^T \text{ m/s}, \\
v_{4,m} = [0.004 \ 0.03]^T \text{ m/s}, & \quad v_{5,m} = [-0.015 \ 0.025]^T \text{ m/s}
\end{align*}
\]

Figure 4 shows the simulated trajectories for leaders, and followers. Colored solid lines represent the actual trajectories $\xi_i$ 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 setpoint trajectory. This is corroborated also by the plot of the distance $d_i$ between each robot and the convex hull $\text{Co}(\mathcal{R})$ spanned by the leaders reported in Figure 5. Since a negative value of $d_i$ indicates that robot $i$ is within $\text{Co}(\mathcal{R})$, from the plot
it is possible to observe that all robots eventually converge to and remain in the convex hull.

It must be noted that at the beginning of the simulation the reference signals moves fast compared to maximum velocity that the robots can exert, causing a significant error in the IOC. This is visible in Figure 6 that shows the absolute 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.
Figure 5: Simulation results: distance $d_i$ from followers to the convex hull; negative distances indicate that the followers are within the hull.

Figure 6: Simulation results: errors of the IOC during the simulation.
5.2 Encircling Controller

Controlling the leaders manually or with open loop control is possible as seen before but not ideal. To show how the leaders could be controlled by a different controller I will use a formation control algorithm. I chose an encircling controller as an example proposed in [5]. This formation controller has the goal of moving all robots on a circle with a given radius and a fixed velocity around a specified center. The full integration of the encircling controller, the containment controller and the underlying proportional controller of the IOC leads to a multi-layer formation control and communication network, with a set of robots (leaders) that are able to communicate long range to a base station to receive global commands on the trajectory of the center, radius and speed of the encircling, while another set (followers) needs only local communication to perform containment control.

In the following I will give a short introduction into the encircling controller used before showing the full multi-layer formation controller in section 5.3.

5.2.1 Problem Setting Encircling Controller

Each of the $N$ robots encircling the target is represented by a kinematic point $R_1, \ldots, R_n$ with first-order dynamics. The absolute number $N$ is not used in the controller design and is not known to the robots. Due to the circular nature of the control task cylindrical coordinates will be used to denote the location of the $i$–th robot $q_i = [\rho_i \ \phi_i \ z_i]^T$ where $\rho_i = \sqrt{\xi_{1,i}^2 + \xi_{2,i}^2}$ denotes the radius, $\phi_i = \text{atan2}(\xi_{1,i}, \xi_{2,i})$ the phase.

The encirclement task is described by the following conditions:

$$\lim_{t \to \infty} \rho_i(t) = \rho^*$$ (31)

$$\lim_{t \to \infty} \phi_i(t) = \bar{\phi}_i(t)$$ (32)

$$\lim_{t \to \infty} \omega_i(t) = \omega^*$$ (33)
where $\rho^*$ is the desired radius and $\bar{\phi}_i(t)$ the average phase of the neighbors of robot $i$ at time $t$. $\omega_i(t)$ denotes the angular velocity at robot $i$ at time $t$ and is supposed to reach the desired angular velocity $\omega^*$.

The encircling controller requires, like the containment controller, a connected communication graph. Due to the low number of robots used in this thesis I will always use a fully connected communication graph. For details on the communication requirements consult the original publication proposing this controller [5].

5.2.2 Background Encircling Control

The goal of the encircling controller proposed in [5] is for every robot to have a phase equal to the average to it’s neighbors phase as shown in (32) and on a set radius. This average is defined as

\[
\bar{\phi}_1 = \frac{\phi_2 + \phi_n - 2\pi}{2} \quad (34)
\]
\[
\bar{\phi}_i = \frac{\phi_{i+1} + \phi_{i-1}}{2} \quad \forall i = 2, \ldots n - 1 \quad (35)
\]
\[
\bar{\phi}_n = \frac{\phi_1 + \phi_{n-1} + 2\pi}{2} \quad (36)
\]

which can be simplified to

\[
\vec{\phi} = [\bar{\phi}_1 \ldots \bar{\phi}_n]^T = C\phi + b \quad (37)
\]

where $C$ is the circular matrix with the first row $[0 \ 1 \ 0 \ldots 0 \ 1/2]$ and $b = [-\pi \ 0 \ldots 0 \ \pi]^T$. $\phi = [\phi_1 \ldots \phi_n]^T$ is the aggregate of the robots’ phases, $\vec{\phi} = [\bar{\phi}_1 \ldots \bar{\phi}_n]^T$ the aggregate of the average phases.

The cylindrical coordinates of the robots’ location and orientation can be considered linear and decoupled. Therefore a separate controller for each component can be designed.
The controller for each of the components can be given as

\[ \dot{\rho}_i = k_\rho (\rho^* - \rho_i) \]  
\[ \dot{\phi}_i = \omega^* + k_\phi (\bar{\phi}_i - \phi_i) \]

with the positive gains \( k_\rho, k_\phi, k_z \). It converges exponentially to the values given in (31) to (33) for any initial condition.

### 5.2.3 Particularization Encircling Control

As the containment controller earlier, the encircling controller is discretized. The three linear decoupled proportional controller equations (38)-(39) become

\[ \rho_{i,m} = \rho_{i,m-1} + \Delta t k_{\rho,d} (\rho^* - \rho_{i,m-1}) \]  
\[ \phi_{i,m} = \phi_{i,m-1} + \Delta t (\omega^* + k_{\phi,d} (\bar{\phi}_{i,m-1} - \phi_{i,m-1})) \]  
\[ z_{i,m} = z_{i,m-1} - \Delta t k_{z,d} z_{i,m-1} \]

with the controller gains \( k_\rho, k_z, k_\phi \) and all variables at time step \( m \) or the previous time step \( m - 1 \).

As before the output of the encircling controller needs a control system architecture consisting of the controller that generates a reference trajectory and an Input/Output Controller (IOC) which has the linear and rotational velocity for that specific robot as an output. The architecture is shown in Figure 7.
The controller output in (40)-(42) gives the output in cylindrical coordinates and not in the cartesian coordinates needed by the rest of the control system. The cylindrical coordinates are converted to cartesian coordinates using

\[ x_{i,m} = \begin{bmatrix} \rho_{i,m} \cos (\phi_{i,m}) \\ \rho_{i,m} \sin (\phi_{i,m}) \end{bmatrix} \]  

(43)

where \( x_{i,m} \) is the set point passed on to the IOC.

5.2.4 Simulation of Three Robots with a Moving Center

To test the implemented encircling controller, I set up a simulation using three robots circling around a moving point. The center point \( c \) was controlled using open loop control.

As I am using only three circling robots in this test, the communication graph is fully connected. The center point \( c \) moves with a velocity of

\[ v_c = \begin{bmatrix} 0.005 \cdot 1.5 + 2 \sin(0.05t + \pi) \\ 0.005 \cdot 1.5 + 2 \sin(0.05t) \end{bmatrix} \, \text{m/s} \]

in a sinusoidal motion, where \( t \) is the time since the start of the simulation in seconds. The desired radius is set at \( \rho^* = 4 \text{ m} \) and the desired rotational speed at \( \omega^* = 0.05 \text{ rad/s} \). The gains are chosen as \( k_\rho = 0.25 \) and \( k_\omega = 0.05 \). The resulting trajectories of the simulation can be seen in Figure 8. The error calculated within the controller between the desired radius \( \rho^* \) and the radius of the desired location \( x_i \) given by the controller is shown in Figure 9. Figure 10 shows the error between the phase of \( x_i \) and the desired phase \( \phi_i \). We can observe that the errors don’t converge towards zero but oscillates around zero. This is mainly caused by the changing velocity of the center. The time constant of the center’s movement is too small for the encircling controller. Therefore, the encircling controller can not follow the fast oscillations of the center point. While the phase error plot shows a smooth line for the first 500 seconds a jittery motion is observable after this time. This is presumably caused by the simulation itself and can not be explained by the controller’s behavior.
Figure 8: Simulation results: trajectories of the virtual center and the encircling robots.

Figure 9: Simulation results: error of the radius $\rho_i$ of the output of the encircling controller.
5.3 Multi-Layer Formation Control

After confirming the performance of the encircling controller in section 5.2 I will show in this section the full multi-layer formation control algorithm made by using an encircling controller to control the leaders of the containment controller.

The connected communication graph among the followers is shown in Figure 11, and it has the following Laplacian matrix $\mathcal{L}$ for the communication between the followers:

$$
\mathcal{L} = \begin{bmatrix}
1 & -1 & 0 & 0 & 0 \\
-1 & 3 & 0 & -1 & -1 \\
0 & 0 & 1 & -1 & 0 \\
0 & -1 & -1 & 3 & -1 \\
0 & -1 & 0 & -1 & 2
\end{bmatrix}
$$

(44)
with the leader-follower weights

\[ \Delta_1 = \text{diag}\{1, 0, 0, 0, 0\} \]
\[ \Delta_2 = \text{diag}\{0, 0, 1, 0, 0\} \]
\[ \Delta_3 = \text{diag}\{0, 0, 0, 0, 1\}. \]

Note that the selected communication graph respects Assumption 3. The controller coefficients \( P, K_3 \) and \( L \) are obtained as

\[
P = \begin{bmatrix} 0.1818 & 0 \\ 0 & 0.1818 \end{bmatrix}
\] (46)

\[
K_3 = \begin{bmatrix} 1.5597 & 0 \\ 0 & 1.5597 \end{bmatrix}
\] (47)

\[
L = \begin{bmatrix} 1.5848 & 0 \\ 0 & 1.5848 \end{bmatrix}
\] (48)

by solving the LMI condition in (18). The center \( c \) of the encircling controller of the leaders is communicated by a base station and moves with a velocity of \( v_c = [0.005 \ 0]^T \text{ m/s} \). The encircling set points are chosen as \( \rho^* = 3.5 \text{ m}, \omega^* = 0.05 \text{ rad/s} \) and the gains as \( k_\rho = 0.25 \) and \( k_\phi = 0.05 \). Figure 12 and 13 show the simulated trajectories for followers, leaders and the center of the encircling controller. While figure 12 shows the trajectories from the start until each robot is on their desired trajectory, figure 13 shows a later point in time at which both controllers reach their final state after which the behavior is repeating itself periodically. Colored solid lines represent the actual trajectories \( \xi \) of the robots while colored dashed lines represent the desired trajectories \( x \). The black dotted line represents the trajectory \( c \) of the encircling controllers’ center. Even though the followers start outside the convex hull spanned by the leaders they end within the hull. The plots confirm that the interaction of the two controllers leads to a successful outcome of the task.
Figure 11: Communication graph in the simulation with the leaders $l_k$ controlled by an encircling controller and the followers $f_i$. The node $b$ is the base station. The communication between the leaders for the encircling controller is shown by the dotted arrows, the communication between leaders and followers for the containment controller by the normal arrows.
Figure 12: Simulation results: trajectories of the virtual center, the leaders performing encircling control and followers performing containment control during the beginning of the simulation where all robot’s converge to their desired trajectories.

Figure 13: Simulation results: trajectories of the virtual center, the followers performing encircling control and followers performing containment control during the later part of the simulation.
CHAPTER 6

Experiment

6.1 Simulation Setup

6.1.1 Hardware

For an experimental implementation a system of three virtual leaders and three $\sim 20\text{cm}$ differential drive mobile robots was used. The robots shown in Figure 14 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 original robots were built as part of [24].

6.1.2 Nonholonomic Constraints

The robot’s are limited in their movement by nonholonomic constraints. These don’t apply to a pure unicycle robot, but to the robot’s used in all simulations as well as experiment’s done in this thesis. If I would use robot’s with four ideal unsteerable wheels, the robot’s only degree of freedom would be to move on the line of their current orientation. Ideal wheels can only rotate around their horizontal axle and can not slip sideways or rotate around the vertical axis. Steering a robot with four fixed wheels can only be done like a tank with tracks which requires slipping. By using two interdependently controllable wheels and an omnidirectional uncontrolled caster wheel the robot’s can be steered without slipping and are closer to the theoretical unicycle model. Therefore the robots were adapted by removing two of the original wheels and substituting them with a caster.

The robots move in a $10m \times 10m$ area equipped with an Optitrack motion capture system that provides the position of the robots needed in the control law.
Figure 14: Robot used for all experiments with two actuated wheels and a caster.
Figure 15: Communication graph in the experiments with the leaders $l_k$ and the followers $f_i$.

6.2 Experiment of Containment Control with Three Followers and Three Virtual Leaders

The communication graph, depicted in Figure 15, is described by the matrices:

$$
L = \begin{bmatrix}
2 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2 \\
\end{bmatrix} \quad \Delta_1 = \text{diag}\{1, 0, 0\} \\
\Delta_2 = \text{diag}\{0, 1, 0\} \\
\Delta_3 = \text{diag}\{0, 0, 1\}.
$$

The corresponding controller coefficients computed by solving the LMI in (18) are:

$$
P = \begin{bmatrix}
0.1818 & 0 \\
0 & 0.1818 \\
\end{bmatrix} \quad (50)
$$

$$
K_3 = \begin{bmatrix}
1.5597 & 0 \\
0 & 1.5597 \\
\end{bmatrix} \quad (51)
$$

$$
L = \begin{bmatrix}
1.5848 & 0 \\
0 & 1.5848 \\
\end{bmatrix} \quad (52)
$$

The virtual leaders move with a velocity of

$$
v_{1,m} = \begin{bmatrix}
0.075 \cos(t/20 + 3/2\pi) + 0.0125 \\
0.075 \sin(t/20 + 3/2\pi) + 0.0125 \\
\end{bmatrix} \text{ m/s } (53)
$$

$$
v_{2,m} = \begin{bmatrix}
0.075 \cos(t/20 + \pi) + 0.0125 \\
0.075 \sin(t/20 + \pi) + 0.0125 \\
\end{bmatrix} \text{ m/s } (54)
$$

$$
v_{3,m} = \begin{bmatrix}
0.075 \cos(t/20) + 0.0125 \\
0.075 \sin(t/20) + 0.0125 \\
\end{bmatrix} \text{ m/s. } (55)
$$
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 16. 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 (49).

Figure 16 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 17 with negative values indicating the robot being inside the convex hull. Screenshots of the video showing the moving robots are in Figure 18-Figure 23.
Figure 16: 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.

Figure 17: Experimental results: distance $d_i$ from followers to the convex hull; negative distances indicate that the followers are within the hull.
Figure 18: Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 0$ s where all robots have not reached their desired location.

Figure 19: Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 4$ s where all robots have reached their desired location but not a useful orientation.
Figure 20: Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 20s$ where all robots have reached their desired location and oriented themselves along their trajectories.

Figure 21: Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 52s$ where all robots start performing the first turn.
Figure 22: Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 84s$ where all robots are in the tightest part of the turn.

Figure 23: Experimental results: location of the robot’s performing containment control using virtual leaders at $t = 116s$ where all robots have left the tight turn.
CHAPTER 7
Conclusion and Future Work

This thesis shows a mobile robot application of the distributed containment controller introduced in [1] and its validation in simulation with ten robots and experiments with three robots. The containment controller has also been tested in a multi-layer control scheme and communication network in which the leaders perform an encircling control and the followers perform the proposed containment control.

In general, the controller showed the expected behavior being able to solve the containment control problem 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 6 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 could be to try to "slow down" the dynamics of the reference trajectories by changing the selected $A_i$ matrix.

Note that the computational requirements grow linearly with the number of neighbors each robot is connected to. This could lead to problems in larger systems with many followers and densely connected communication networks. However one potential solution to this problem would be the artificial elimination of redundant connections within the communication graph as long as Assumption 3 is still
fulfilled.

In the future, on the one hand, we plan to investigate improvements to the control scheme presented in this paper by studying the effects of different values of $A_i$ on the closed loop system, and by studying the problem of reduction of the connections in the communication graph. On the other hand, we plan to implement the 3D equivalent version of this controller on drones.

The multi-layer control scheme simulated could be expanded to different types of controllers for the leaders. Additionally an experimental implementation would be interesting.
LIST OF REFERENCES

[1] C. Yuan, W. Zeng, and S.-L. Dai, “Distributed model reference adaptive containment control of heterogeneous uncertain multi-agent systems,” *ISA Transactions*, vol. 86, pp. 73–86, Mar. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0019057818304294

[2] B. Zhu, L. Xie, and D. Han, “Recent developments in control and optimization of swarm systems: A brief survey,” in *2016 12th IEEE International Conference on Control and Automation (ICCA)*. Kathmandu, Nepal: IEEE, June 2016, pp. 19–24. [Online]. Available: http://ieeexplore.ieee.org/document/7505246/

[3] S. Xiao and J. Dong, “Distributed Fault-Tolerant Containment Control for Nonlinear Multi-Agent Systems Under Directed Network Topology via Hierarchical Approach,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 806–816, Apr. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9374809/

[4] X. Wang, S. Li, and P. Shi, “Distributed Finite-Time Containment Control for Double-Integrator Multiagent Systems,” *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1518–1528, Sept. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6670105/

[5] A. Franchi, P. Stegagno, and G. Oriolo, “Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance,” *Autonomous Robots*, vol. 40, no. 2, pp. 245–265, Feb. 2016. [Online]. Available: http://link.springer.com/10.1007/s10514-015-9450-3

[6] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, “Containment Control in Mobile Networks,” *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1972–1975, Sept. 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4631516/

[7] J. Li, W. Ren, and S. Xu, “Distributed Containment Control with Multiple Dynamic Leaders for Double-Integrator Dynamics Using Only Position Measurements,” *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1553–1559, June 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6069542/

[8] W. Zhang and Y. Tang, “Consensus control for agent networks with stationary leaders,” in *2016 Chinese Control and Decision Conference (CCDC)*. Yinchuan, China: IEEE, May 2016, pp. 241–245. [Online]. Available: http://ieeexplore.ieee.org/document/7530988/
[9] H. Zhang, Z. Zhao, Z. Meng, and Z. Lin, “Experimental Verification of a Multi-robot Distributed Control Algorithm with Containment and Group Dispersion Behaviors: The Case of Dynamic Leaders * *This work was supported in part by the National Natural Science Foundation of China under Grant No. 61273105.” IFAC Proceedings Volumes, vol. 46, no. 13, pp. 153–158, 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S147466701630249X

[10] Y. Cao, D. Stuart, W. Ren, and Z. Meng, “Distributed containment control for double-integrator dynamics: Algorithms and experiments,” in Proceedings of the 2010 American Control Conference. Baltimore, MD: IEEE, June 2010, pp. 3830–3835. [Online]. Available: http://ieeexplore.ieee.org/document/5531204/

[11] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, “Theory and Experiment on Formation-Containment Control of Multiple Multirotor Unmanned Aerial Vehicle Systems,” IEEE Transactions on Automation Science and Engineering, vol. 16, no. 1, pp. 229–240, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8295262/

[12] M. Santilli, M. Franceschelli, and A. Gasparri, “Dynamic Resilient Containment Control in Multirobot Systems,” IEEE Transactions on Robotics, pp. 1–14, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9361756/

[13] Z. Li, Z. Duan, W. Ren, and G. Feng, “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, vol. 25, no. 13, pp. 2101–2121, Sept. 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/rnc.3195

[14] L. Bo, C. Zeng-qiang, L. Zhong-xin, and Z. Qing, “Containment control of discrete-time multi-agent systems with multiple stationary leaders and time-delays,” in 2015 34th Chinese Control Conference (CCC). Hangzhou, China: IEEE, July 2015, pp. 7062–7066. [Online]. Available: http://ieeexplore.ieee.org/document/7260756/

[15] J. Fu, Y. Wan, G. Wen, and T. Huang, “Distributed Robust Global Containment Control of Second-Order Multiagent Systems With Input Saturation,” IEEE Transactions on Control of Network Systems, vol. 6, no. 4, pp. 1426–1437, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8616812/

[16] B. Li, H.-Y. Yang, Z.-Q. Chen, and Z.-X. Liu, “Distributed Containment Control of Multi-agent Systems with General Linear Dynamics and
[17] J. Qin, Q. Ma, X. Yu, and Y. Kang, “Output Containment Control for Heterogeneous Linear Multiagent Systems With Fixed and Switching Topologies,” *IEEE Transactions on Cybernetics*, vol. 49, no. 12, pp. 4117–4128, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8457489/

[18] H. Liang, Y. Zhou, H. Ma, and Q. Zhou, “Adaptive Distributed Observer Approach for Cooperative Containment Control of Nonidentical Networks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 299–307, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8275040/

[19] G. Wen, P. Wang, T. Huang, W. Yu, and J. Sun, “Robust Neuro-Adaptive Containment of Multileader Multiagent Systems With Uncertain Dynamics,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 406–417, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/7984856/

[20] T. Xu, G. Lv, Z. Duan, Z. Sun, and J. Yu, “Distributed Fixed-Time Triggering-Based Containment Control for Networked Nonlinear Agents Under Directed Graphs,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 10, pp. 3541–3552, Oct. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9090308/

[21] N. Gu, Z. Peng, D. Wang, and F. Zhang, “Path-Guided Containment Maneuvering of Mobile Robots: Theory and Experiments,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7178–7187, Aug. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9113689/

[22] K. Do, “Global output-feedback path-following control of unicycle-type mobile robots: A level curve approach,” *Robotics and Autonomous Systems*, vol. 74, pp. 229–242, Dec. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889015001669

[23] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*, ser. Advanced Textbooks in Control and Signal Processing, M. J. Grimble and M. A. Johnson, Eds. London: Springer London, 2009. [Online]. Available: http://link.springer.com/10.1007/978-1-84628-642-1

[24] R. Perera, “A PHD FILTER BASED RELATIVE LOCALIZATION FILTER FOR ROBOTIC SWARMS,” Ph.D. dissertation, University of Rhode Island, Kingston, RI, 2021. [Online]. Available: https://digitalcommons.uri.edu/theses/1927
BIBLIOGRAPHY

Bo, L., Zeng-qiang, C., Zhong-xin, L., and Qing, Z., “Containment control of discrete-time multi-agent systems with multiple stationary leaders and time-delays,” in 2015 34th Chinese Control Conference (CCC). Hangzhou, China: IEEE, July 2015, pp. 7062–7066. [Online]. Available: http://ieeexplore.ieee.org/document/7260756/

Cao, Y., Stuart, D., Ren, W., and Meng, Z., “Distributed containment control for double-integrator dynamics: Algorithms and experiments,” in Proceedings of the 2010 American Control Conference. Baltimore, MD: IEEE, June 2010, pp. 3830–3835. [Online]. Available: http://ieeexplore.ieee.org/document/5531204/

Do, K., “Global output-feedback path-following control of unicycle-type mobile robots: A level curve approach,” Robotics and Autonomous Systems, vol. 74, pp. 229–242, Dec. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889015001669

Dong, X., Hua, Y., Zhou, Y., Ren, Z., and Zhong, Y., “Theory and Experiment on Formation-Containment Control of Multiple Multirotor Unmanned Aerial Vehicle Systems,” IEEE Transactions on Automation Science and Engineering, vol. 16, no. 1, pp. 229–240, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8295262/

Franchi, A., Stegagno, P., and Oriolo, G., “Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance.” Autonomous Robots, vol. 40, no. 2, pp. 245–265, Feb. 2016. [Online]. Available: http://link.springer.com/10.1007/s10514-015-9450-3

Fu, J., Wan, Y., Wen, G., and Huang, T., “Distributed Robust Global Containment Control of Second-Order Multiagent Systems With Input Saturation,” IEEE Transactions on Control of Network Systems, vol. 6, no. 4, pp. 1426–1437, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8616812/

Gu, N., Peng, Z., Wang, D., and Zhang, F., “Path-Guided Containment Maneuvering of Mobile Robots: Theory and Experiments,” IEEE Transactions on Industrial Electronics, vol. 68, no. 8, pp. 7178–7187, Aug. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9113689/

Ji, M., Ferrari-Trecate, G., Egerstedt, M., and Buffa, A., “Containment Control in Mobile Networks,” IEEE Transactions on Automatic Control, vol. 53, no. 8, pp. 1972–1975, Sept. 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4631516/
Li, B., Yang, H.-Y., Chen, Z.-Q., and Liu, Z.-X., “Distributed Containment Control of Multi-agent Systems with General Linear Dynamics and Time-delays,” International Journal of Control, Automation and Systems, vol. 16, no. 6, pp. 2718–2726, Dec. 2018. [Online]. Available: http://link.springer.com/10.1007/s12555-017-0696-8

Li, J., Ren, W., and Xu, S., “Distributed Containment Control with Multiple Dynamic Leaders for Double-Integrator Dynamics Using Only Position Measurements,” IEEE Transactions on Automatic Control, vol. 57, no. 6, pp. 1553–1559, June 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6069542/

Li, Z., Duan, Z., Ren, W., and Feng, G., “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, vol. 25, no. 13, pp. 2101–2121, Sept. 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/rnc.3195

Liang, H., Zhou, Y., Ma, H., and Zhou, Q., “Adaptive Distributed Observer Approach for Cooperative Containment Control of Nonidentical Networks,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 2, pp. 299–307, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8275040/

Perera, R., “A PHD FILTER BASED RELATIVE LOCALIZATION FILTER FOR ROBOTIC SWARMS,” Ph.D. dissertation, University of Rhode Island, Kingston, RI, 2021. [Online]. Available: https://digitalcommons.uri.edu/theses/1927

Qin, J., Ma, Q., Yu, X., and Kang, Y., “Output Containment Control for Heterogeneous Linear Multiagent Systems With Fixed and Switching Topologies,” IEEE Transactions on Cybernetics, vol. 49, no. 12, pp. 4117–4128, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8457489/

Santilli, M., Franceschelli, M., and Gasparri, A., “Dynamic Resilient Containment Control in Multirobot Systems,” IEEE Transactions on Robotics, pp. 1–14, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9361756/

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., Robotics, ser. Advanced Textbooks in Control and Signal Processing, Grimble, M. J. and Johnson, M. A., Eds. London: Springer London, 2009. [Online]. Available: http://link.springer.com/10.1007/978-1-84628-642-1

Wang, X., Li, S., and Shi, P., “Distributed Finite-Time Containment Control for Double-Integrator Multiagent Systems,” IEEE Transactions on
Wen, G., Wang, P., Huang, T., Yu, W., and Sun, J., “Robust Neuro-Adaptive Containment of Multileader Multiagent Systems With Uncertain Dynamics,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 406–417, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/7984856/

Xiao, S. and Dong, J., “Distributed Fault-Tolerant Containment Control for Nonlinear Multi-Agent Systems Under Directed Network Topology via Hierarchical Approach,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 806–816, Apr. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9374809/

Xu, T., Lv, G., Duan, Z., Sun, Z., and Yu, J., “Distributed Fixed-Time Triggering-Based Containment Control for Networked Nonlinear Agents Under Directed Graphs,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 10, pp. 3541–3552, Oct. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9090308/

Yuan, C., Zeng, W., and Dai, S.-L., “Distributed model reference adaptive containment control of heterogeneous uncertain multi-agent systems,” *ISA Transactions*, vol. 86, pp. 73–86, Mar. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0019057818304294

Zhang, H., Zhao, Z., Meng, Z., and Lin, Z., “Experimental Verification of a Multi-robot Distributed Control Algorithm with Containment and Group Dispersion Behaviors: The Case of Dynamic Leaders *This work was supported in part by the National Natural Science Foundation of China under Grant No. 61273105.*” *IFAC Proceedings Volumes*, vol. 46, no. 13, pp. 153–158, 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S147466701630249X

Zhang, W. and Tang, Y., “Consensus control for agent networks with stationary leaders,” in *2016 Chinese Control and Decision Conference (CCDC)*. Yinchuan, China: IEEE, May 2016, pp. 241–245. [Online]. Available: http://ieeexplore.ieee.org/document/7530988/

Zhu, B., Xie, L., and Han, D., “Recent developments in control and optimization of swarm systems: A brief survey,” in *2016 12th IEEE International Conference on Control and Automation (ICCA)*. Kathmandu, Nepal: IEEE, June 2016, pp. 19–24. [Online]. Available: http://ieeexplore.ieee.org/document/7505246/