A virtual vehicle approach to distributed control for formation keeping of underactuated vehicles *

Yann Roth[1], Andrea Alessandretti[12], A. Pedro Aguiar[23], and Colin N. Jones[1]

***Abstract*— This paper addresses the control problem of formation keeping of a fleet of underactuated vehicles under communication constraints, where each agent is allowed to communicate with only a subset of the agents of the fleet. We adopt a virtual vehicle approach where every underactuated agent tracks a virtual vehicle, described by a single integrator model, driven by a consensus-based distributed controller. This approach results in a distributed dynamic controller for formation keeping of underactuated vehicles with exponential convergence guarantee of the formation error to zero. Simulation results are presented for both wheeled-like vehicles (2-D case) and UAV-like vehicles (3-D case).**

I. INTRODUCTION

The noticeable increase of low cost advanced robotic vehicles gives space to a broad range of applications for both academy and industry. Coordination is one of the fundamental tasks of autonomous vehicles, and when the number of agents increases the need of reliable distributed control strategy becomes crucial. In this paper we present a continuous time distributed controller for formation keeping of underactuated vehicles with state and network (information from the neighbor vehicles) feedback.

Consensus algorithms are one of the main tools used to design distributed controllers. Consider a set of systems modeled as single integrators, where each system can only access the state of a subset of the other systems in the network. The goal of a consensus algorithm is to compute a control input that steers the sate of all the systems to a common value. We refer to [1], [2] and the references therein for a survey on the topic. In [3] the authors present an adaptation of the consensus algorithm for the problem of formation keeping, where all the single integrators are steered to a predefined formation around a moving leader. In [4] the formation keeping problem for unicycle-like vehicles is considered.

As main drawback of the methods mentioned above, convergence guarantees are provided only for the case of the single integrator model. Moreover, whenever more complex underactuated vehicles are considered, only the 2-D case is addressed.

Motivated by these observations, in this paper we present a distributed dynamic formation keeping controller with convergence guarantee for a class of 2-D and 3-D underactuated vehicles. This is achieved combining the trajectory-tracking control law adopted in our previous works [5], [6] with the graph-based formation control algorithm for single integrator systems from [3].

The remainder of this paper is organized as follows. Section II presents some results from the literature. The description of the control problem addressed is introduced in Section III. Section IV contains the main results of the paper, followed by Section V with numerical examples. The paper is closed with some conclusion in Section VI.

II. BACKGROUND

In this section we recall some results from the literature used in the problem definition and for the design of the controller.

A. Definition of a Graph

We start with some concepts from graph theory, form [1], [2], [3], used to define the communication structor among the agents.

Let $G = (V, E)$ denote a *graph*, where

- $V = \{v_i,\ i = 1,\ ...,\ N\}$, called *node set*, denotes the set of elements, the nodes, in the graph, and
- $E = \{(v_i,\ v_j)_k,\ k = 1,\ ...,\ M\} \subseteq V \times V$, called *edge set*, denotes the set of directed edges between the nodes of the set V, where if $(v_i,\ v_j) \in E$ then v_i can communicate to v_j.

A method to represent the edge set of a graph consists in using the *adjacency matrix* $A \in \mathbb{R}^{N\times N}$, where the generic element a_{ij} at row i and column j is defined as

$$a_{ij} := \begin{cases} 1 & \text{if} \quad (v_j, v_i) \in E \\ 0 & \text{otherwise} \end{cases} .$$

In general, for the case $(v_j, v_i) \in E$, one can choose a_{ij} to be any positive scalar, for the sake of simplicity we choose it equal to one.

An example of a graph with 5 nodes is given in Fig. 1 with associated adjacency matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} .$$

A *directed path* is a sequence of directed edges that connects a sequence of nodes, where all the edges are directed in the same direction.

* The second author benefited from grant SFRH/BD/51450/2011 of the Foundation for Science and Technology (FCT), Portugal.

[1] École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

[2] Institute for Systems and Robotics (ISR), Instituto Superior Tecnico (IST), Lisbon, Portugal.

[3] Faculty of Engineering, University of Porto (FEUP), Porto, Portugal.

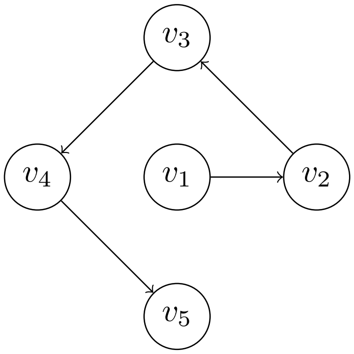

Fig. 1. Example of graph with 5 nodes.

B. Distributed Formation Keeping for Single Integrators

In this subsection we recall a distributed control law, from [3], designed to drive a set of single integrators in formation around a moving leader.

Consider a set of $N+1$ agents consisting of N followers and one leader (the number $N+1$) modeled by single integrators

$$\dot{x}_i(t) = u_i(t), \qquad i = 1,\,\ldots,\,N+1 \tag{1}$$

where $x_i(t) \in \mathbb{R}^n$ and $u_i(t) \in \mathbb{R}^n$ denote the position and velocity, respectively, of the i-th vehicle at time t.

Moreover, let the set of vectors x_{id}, with $i = 1, \ldots,\ N+1$, define a desired formation, where the desired displacement between the agent i and the agent j corresponds to $d_{ij} := x_{id} - x_{jd}$.

The following distributed formation keeping controller for single integrators is from Theorem 3.8 of [3].

Theorem 1 (Formation Keeping of Single Integrators): Consider a set of $N+1$ vehicles modeled by (1) that communicate according to a communication structure (described in Subsection II-A) defined by the adjacency matrix $A \in \mathbb{R}^{n\times n}$. Moreover, let the set of vector x_{id} that define the desired formation be given. If there exist a directed path from v_{N+1} to all the other nodes, then for the closed loop system (1) with

$$u_i(t) = \frac{1}{\eta_i}\sum_{j=1}^{N} a_{ij}[\dot{x}_j(t) - \gamma(x_i(t) - x_j(t) - d_{ij})], \tag{2}$$

with $i = 1,...,N$, $\eta_i := \sum_{j=1}^{N} a_{ij}$, $\gamma > 0$, and $d_{ij} := x_{id} - x_{jd}$, the formation is asymptotically satisfied, i.e., $x_i(t) - x_j(t) \to x_{id} - x_{jd}$ as t goes to infinity. □

It is worth to notice that the input of the leader u_{N+1} can be chosen arbitrarily, thus the vehicles will converge to formation independently from the trajectory of the leader.

C. Trajectory-tracking algorithm

In this section we recall a trajectory-tracking algorithm from [5], [6]. We start by defining the model of an underactuated vehicle moving both in a 2-D plane, e.g., wheeled robot, or a 3-D space, e.g., Unmanned Aerial Vehicle (UAV), where we use the variable $d = 2$ or $d = 3$, respectively, to distinguish between this two cases.

Consider an inertial coordinate frame denoted by I and a body coordinate frame attached to the vehicle denoted by B. The position and orientation of the vehicle is denoted by the pair $(p(t), R(\theta(t))) \in SE(d)$, where for a given $n \in \mathbb{N}$, $SE(n)$ is the Cartesian product of $\mathbb{R}^n$ with the group $SO(n)$ of $n \times n$ rotation matrixes and $R(\theta(t))$ is a rotation matrix, that maps from body to inertial coordinates, associated with the heading vector $\theta(t)$. Next, we denote the twist that defines the velocity of the vehicle, linear and angular, by the pair $(v(t), \Omega(\omega(t))) \in se(d)$ where for a given $n \in \mathbb{N}$, $se(n)$ is the Cartesian product of $\mathbb{R}^n$ with the space $so(n)$ of $n \times n$ skew-symmetric matrices and the matrix $\Omega(\omega(t))$ is a skew-symmetric matrix associated to the angular velocity $\omega(t)$.

2-D	3-D
$p = \begin{pmatrix} x \\ y \end{pmatrix}$	$p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
$v = \begin{pmatrix} v_f \\ 0 \end{pmatrix}$	$v = \begin{pmatrix} v_f \\ 0 \\ 0 \end{pmatrix}$
$\Omega(\omega) = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}$	$\Omega(\omega) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$

Then, we can write

$$\dot{p}(t) = R(\theta(t))v(t) \tag{3a}$$

$$\dot{R}(\theta(t)) = R(\theta(t))\Omega(\omega(t)). \tag{3b}$$

Since we are interested in the class of underactuated vehicles, only the forward velocity and the angular velocity are considered as input of the system, i.e.,

$$u(t) = (v_f(t), \omega(t)')'. \tag{4}$$

Theorem 2 (Trajectory-Tracking controller): Let $p_d(t)$, with $t \in [0, \infty)$, be a differentiable desired trajectory and consider a vector $\epsilon \in \mathbb{R}^d$ such that the matrix Δ from Tab. I is full rank. Then, for the system (3)-(4) in closed-loop with

$$u(t) = \Delta'(\Delta\Delta')^{-1}\left(R(t)'\dot{p}_d(t) - Ke(t)\right) \tag{5}$$

where $e(t) := R(t)'(p(t) - p_d(t)) - \epsilon$ and $K \in \mathbb{R}^{d\times d}$ is any positive-definite matrix, the origin $e = 0$ is a global exponentially stable equilibrium point. □

It is worth to notice that an appropriate choose of the vector ϵ, potentially arbitrarily small in module, that makes the matrix Δ full rank always exists.

Remark 1 (Tracking accuracy): From the exponential stability of the origin $e = 0$ we have that $\|e\| \to 0$ as $t \to \infty$ and consequently, by definition of the vector e, the position of the vehicle converges to an arbitrarily small ball, of size determined by the vector ϵ, around the desired position, i.e., $\|p(t) - p_d(t)\| \to \|\epsilon\|$ as $t \to \infty$. □

2-D	3-D
$\epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix}$	$\epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{pmatrix}$
$\Delta = \begin{pmatrix} 1 & \epsilon_2 \\ 0 & -\epsilon_1 \end{pmatrix}$	$\Delta = \begin{pmatrix} 1 & 0 & -\epsilon_3 & \epsilon_2 \\ 0 & \epsilon_3 & 0 & -\epsilon_1 \\ 0 & -\epsilon_2 & \epsilon_1 & 0 \end{pmatrix}$

TABLE I

DEFINITION OF ϵ AND Δ FOR THE 2-D AND 3-D CASE.

III. PROBLEM STATEMENT

This paper addresses the design of a control strategy to steer a fleet of vehicles modeled by (3) to formation around a moving leader. We use the subscript to distinguish among the different vehicles, i.e., p_i, R_i, and u_i refer to the position vector, rotation matrix, and input vector, respectively, of the generic agent i.

Problem 1 (Formation keeping): Consider a set of $N+1$ vehicles, denoted by v_i, $i = 1, \ldots, N+1$, modeled by (3)-(4) moving in a d-dimential space, with $d = 2$ or $d = 3$. Let the first N be called follower vehicles and the $(N+1)$-th be the leader. Let a communication structure (described in Subsection II-A) be defined by the adjacency matrix $A \in \mathbb{R}^{n \times n}$. Moreover, let a desired formation be given and defined by a set of vectors $d_{ij} \in \mathbb{R}^d$ which represent the desired offset between the agents i and j. Design a control law for the follower vehicles such that the associated positions asymptotically satisfies the formation with an arbitrary small error, i.e., $\|p_i(t) - p_j(t) - d_{ij}\| \to \|\epsilon\|$ as $t \to \infty$ for some vector $\epsilon \in \mathbb{R}^d$ with arbitrarily small module. □

IV. MAIN RESULT

The strategy chosen for the design of the controller consists in introducing a virtual vehicles that allows us to divide the control Problem 1 in the two subproblems presented in the sections II-B and II-C above.

The problem is divided as follows:

- **Formation of virtual vehicles:** We define the internal state of the controller to be the position of a virtual vehicle, modeled by a single integrator, that is driven using the consensus algorithm presented in Section II-B.
- **Extension to real vehicles:** Then, the position of the virtual vehicle, together with its velocity, is used as trajectory that is tracked by the underactuated vehicles using the trajectory-tracking controller of Section II-C.

This approach results in the following dynamic controller.

Proposition 1 (Formation keeping): Consider the control Problem 1 with vector $\epsilon \in \mathbb{R}^d$ such that the matrix Δ from Tab. I is full rank. Moreover, let the graph induced by the adjacency matrix A contain a directed path from the leader to all the followers. Then, the control law

$$\begin{aligned} u_i(t) &= \Delta'(\Delta\Delta')^{-1} \left(R_i'(t) u_{vi}(t) - K e_{vi}(t) \right) \\ \dot{x}_{ci}(t) &= u_{vi}(t) \\ e_{vi}(t) &:= R_i(t)'(p_i(t) - x_{ci}(t)) - \epsilon \\ u_{vi}(t) &:= \frac{1}{\eta_i} \sum_{j=1}^{N} a_{ij} [\dot{x}_{cj}(t) - \gamma(x_{ci}(t) - x_{cj}(t) - d_{ij})] \end{aligned}$$

for any positive definite matrix $K \in \mathbb{R}^{d \times d}$ with $\eta_i := \sum_{j=1}^{N} a_{ij}$, solves Problem 1 for any $\gamma > 0$. □

Proof: The proof follows directly from the Theorems 1-2. In fact by Theorem 1 the virtual vehicles converge to the formation and by Theorem 2 the real vehicles converge to an ϵ-tube around the position of the associated virtual vehicle. ■

Remark 2 (Information excange): Note that the vehicles do not observe the position the (real) neighbor vehicles, as common in other consensus algorithm, but they exchange the internal state of the controllers, which corresponds with the position of the virtual vehicles. □

Remark 3 (Velocity approximation): The information required to compute the input is the position and the velocity of the neighbor virtual vehicles, i.e., x_{jc} and $\dot{x}_{jc}$, respectively, for the generic neighbor vehicle j. Since the velocity cannot be directly read, we use a numerical approximation obtained from the current and previous reading of the position. Thus, considering a model discretization with discretization step $\delta > 0$, we have

$$\dot{x}_{cj}(t) \approx \frac{x_{cj}(t) - x_{cj}(t-\delta)}{\delta}.$$

□

A. The leader

Consider the case where the leader only communicates with agent v_1, then the information about the leader position and velocity are only locally required by v_1 to compute u_1. This implies that the leader position and velocity could be decided by v_1 without the actual existence of a real leader. Motivated by this observation we identify the two following scenarios:

1) **Physical leader:** In this case the agent $N+1$ is a physical vehicles and its position and velocity can either
   - be sent from the leader to some followers
   - or be estimated using relative measurements from sensors on board of the followers.
2) **Virtual leader:** In this case the agent $N+1$ is virtual and its position an velocity are generated and used by an agent of the network that is virtually connected to the leader.

V. SIMULATION RESULTS

The control strategy proposed in Section IV is implemented for the case of vehicles moving in a 2-D or 3-D space. The numerical results are produced using the Matlab Toolbox VirtualArena [7] and the code is available on-line at the toolbox page.

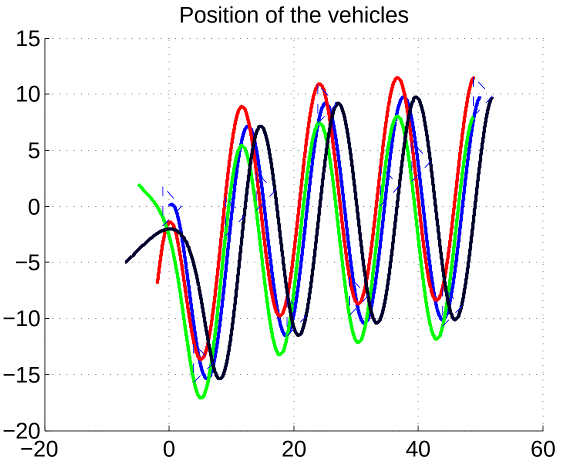

Fig. 2. The closed loop position trajectories of the unicycle-like vehicles are denoted with continuous lines and the dashed lines represent the desired formation.

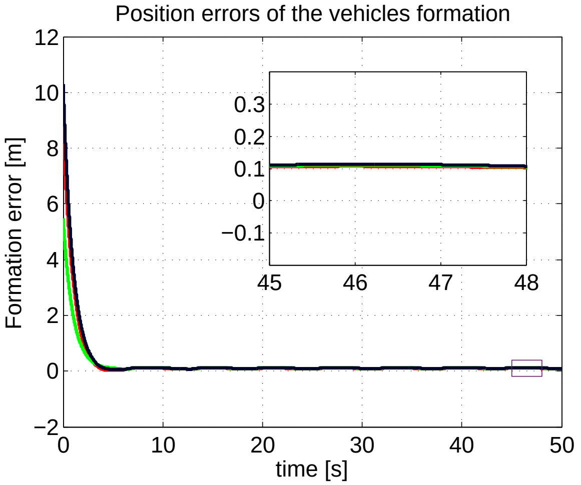

Fig. 3. Time evolution of the norm of the formation error associated to each agent for the 2-D case of unicycle-like vehicles.

A. *Wheeled robot (2-D case)*

In this section we consider four unicycle-like vehicles modeled like (3), for the 2-D case, with initial positions

$$p_1(0) = \begin{pmatrix}0 & 0\end{pmatrix}', \qquad p_2(0) = \begin{pmatrix}-2 & -7\end{pmatrix}',$$
$$p_3(0) = \begin{pmatrix}-5 & 2\end{pmatrix}', \qquad p_4(0) = \begin{pmatrix}-7 & -5\end{pmatrix}'.$$

and heading equal to zero. The leader, agent one, is driven by the controller presented in section II-C to follow the trajectory

$$p_{ld}(t) = \begin{pmatrix}t & 10cos(10^{-3}t)\end{pmatrix}'$$

with controller parameters

$$\epsilon = \begin{pmatrix}-0.1 & 0\end{pmatrix}', \qquad K_e = 0.1 I_{2\times2}$$

where, for a generic positive integer n, the matrix $I_{n\times n}$ denotes the identity matrix of dimension n.

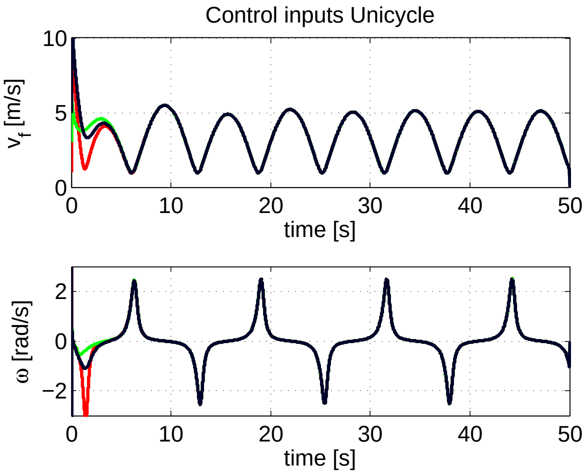

Fig. 4. Inputs signals of each vehicle for the 2-D case.

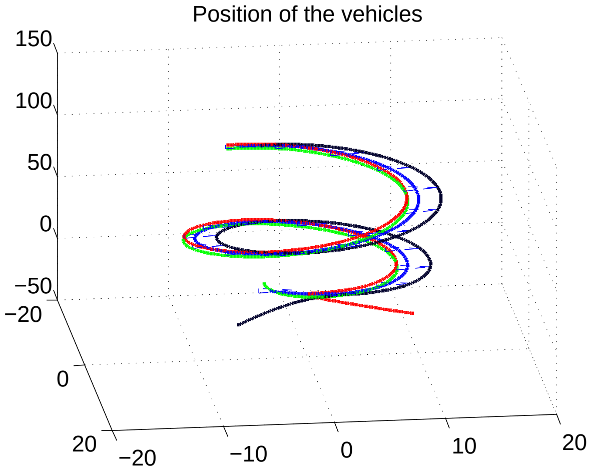

Fig. 5. The closed loop position trajectories of the vehicles are denoted with continuous line and the dashed lines represent the desired formation.

The communication structure is of a chain form where agent i communicates to agent $i+1$, with $i = 1, \dots, 3$.

In the desired formation, the followers are uniformly distributed along a circle of radius 2 meters around the leader.

From the closed loop position trajectories of the vehicles displayed in Fig. 2, with the associated formation error displayed in Fig. 3, it can be seen that the vehicles achieve formation within the desired asymptotic formation error of $\|\epsilon\| = 0.1$. The associated control input signals are displayed in Fig. 4.

B. *Unmanned Aerial Vehicle (3-D case)*

For the 3-D case we consider four vehicles with initial conditions

$$p_1(0) = \begin{pmatrix}0 & -1 & -3\end{pmatrix}', \qquad p_2(0) = \begin{pmatrix}5 & 9 & -4\end{pmatrix}',$$
$$p_3(0) = \begin{pmatrix}-7 & -3 & -7\end{pmatrix}', \quad p_4(0) = \begin{pmatrix}6 & -7 & -6\end{pmatrix}'.$$

and $R_i(0) = I_{3\times3}$, $i = 1, \dots,\ 4$. The agent number one, which is the leader of the fleet, is driven by the controller

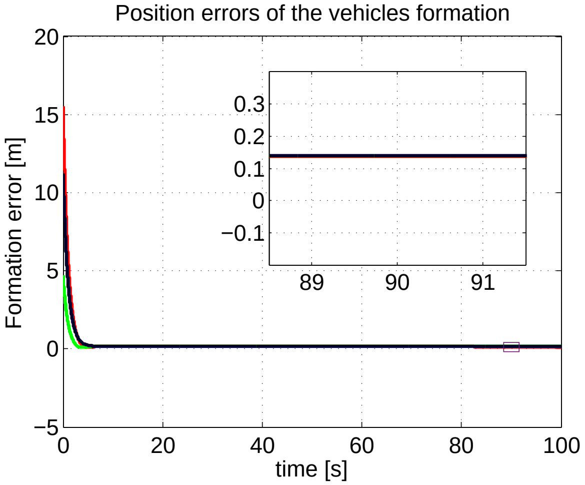

Fig. 6. Time evolution of the norm of the formation error associated to each agent for the 3-D case of UAV-like vehicles.

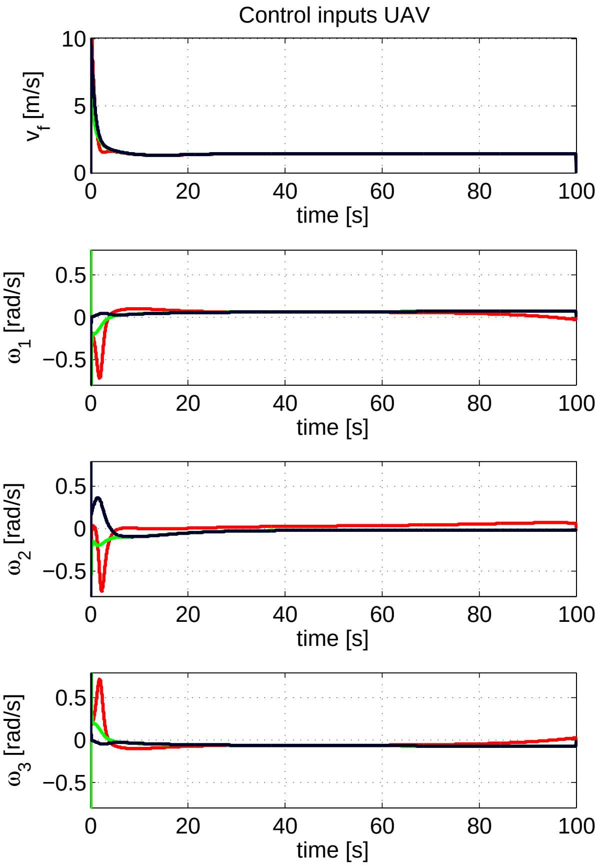

Fig. 7. Inputs signals of each vehicle for the 3-D case.

presented in section II-C to follow the trajectory

$$p_{ld}(t) = \begin{pmatrix} 10\cos(0.1t) & 10\sin(0.1t) & t \end{pmatrix}'$$

with controller parameters

$$\epsilon = \begin{pmatrix} -0.1 & 0 & -0.1 \end{pmatrix}', \qquad K_e = 0.1 I_{3\times 3}.$$

The same communication structure of the 2-D case is adopted. At the desired formation, the followers are uniformly distributed along a circle in the $y-z$ plane of radius 2 meters around the leader.

From the closed loop position trajectories of the vehicles displayed in Fig. 5 with the associated formation error displayed in Fig. 6, it can be seen that the vehicles achieve formation within the desired asymptotic formation error of $\|\epsilon\| \approx 0.14$. The associated control input signals are displayed in Fig. 7.

VI. Conclusions

A distributed controller for formation keeping of multiple underactuated vehicles is presented. The proposed virtual vehicle approach allows to extend the standard consensus based formation keeping algorithm, designed for single integrator systems, to the class of underactuated vehicles resulting in a distributed controller with exponential convergence guarantee of the tracking error to zero. Simulation results are presented for vehicles moving in both 2-D and 3-D space and the code is made available on-line.

References

[1] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.

[2] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[3] Wei Ren and RW Beard. Distributed consensus in multi-vehicle Cooperative control. *Springer-VerlagLondon*.

[4] R Falconi, S Gowal, and A Martinoli. Graph based distributed control of non-holonomic vehicles endowed with local positioning information engaged in escorting missions. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3207–3214, May 2010.

[5] A.P. Aguiar and J.P. Hespanha. Position tracking of underactuated vehicles. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 3, pages 1988–1993. IEEE, 2003.

[6] A. Alessandretti, A.P. Aguiar, and C.N. Jones. Trajectory-tracking and path-following controllers for constrained underactuated vehicles using Model Predictive Control. In *Control Conference (ECC), 2013 European*, pages 1371–1376, 2013.

[7] Andrea Alessandretti, Pedro A. Aguiar, and Colin Jones. Virtual Arena [Software]. `http://wiki.epfl.ch/virtualarena`, 2014.