applied sciences

MDPI

Article

Artificial Neural Networks in Coordinated Control of Multiple Hovercrafts with Unmodeled Terms

Kairong Duan [1,*], Simon Fong [1,*], Yan Zhuang [1] and Wei Song [2]

[1] Department of Computer and Information Science, University of Macau, Taipa 999078, Macau, China; syz@umac.mo

[2] School of Computer Science, North China University of Technology, Beijing 100144, China; sw@ncut.edu.cn

* Correspondence: yb67408@umac.mo (K.D.); ccfong@umac.mo (S.F.)

Received: 17 April 2018; Accepted: 22 May 2018; Published: 24 May 2018

Abstract: In this paper, the problem of coordinated control of multiple hovercrafts is addressed. For a single hovercraft, by using the backstepping technique, a nonlinear controller is proposed, where Radial Basis Function Neural Networks (RBFNNs) are adopted to approximate unmodeled terms. Despite the application of RBFNNs, integral terms are introduced, improving the robustness of controller. As a result, global uniformly ultimate boundedness is achieved. Regarding the communication topology, two different directed graphs are chosen under the assumption that there are no delays when they communicate with each other. In order to testify the performance of the proposed strategy, simulation results are presented, showing that vehicles can move forward in a specific formation pattern and RBFNNs are able to approximate unmodeled terms.

Keywords: surface vehicle; underactuated vehicle; RBFNNs; directed graph; coordinated control

1. Introduction

In recent years, Wireless Sensor Networks (WSNs) have attracted growing interests from researchers, because they have merits, compared with traditional networking solutions, such as reliability, flexibility, and an ease of deployment, that enable their use in a wide range of varied application scenarios [1]. They can be applied to track moving objects, to monitor special areas so as to trigger alarm systems when some dangerous signals are detected, etc. As the eyes and ears of the IoT, WSNs can work as bridges to build connections between the real-world and the digital-world. In light of this promising application scenario, this paper mainly focuses on a case study of mobile WSNs, where a group of hovercrafts equipped with specific sensors are chosen as test platforms. The objective is to enable them to move around and interact with the physical environment [2] and thus execute a mission of mapping, searching, and monitoring in a specific area.

Coordinated control of a fleet of hovercrafts is challenging, especially when we take into account their complex dynamic models. Until now, for a single surface vehicle, many research results have been reported. For example, a linear fuzzy-PID controller was proposed in [3]. Compared with the ordinary PID controller, the proposed controller therein performs better in term of improving settling time and reducing overshoot of the control signal. However, their works just consider the kinematic models of the vehicle without considering the dynamic models, which is not realistic in real operation scenarios. Another weakness of the linear controller is that it usually achieves local stability, e.g., [4], where velocity and position controllers were developed based on a linearized system, which is controllable only when the angular velocity is nonzero. Considering the limitations of linear controllers, in [5,6], nonlinear controllers for underactuated ships were designed, and global asymptotic stability is achieved. In [7], a nonlinear Lyapunov-based tracking controller was presented, and it was able to exponentially stabilize the position tracking error to a neighborhood of the origin that can be made arbitrarily small. A method of incorporating multiobjective controller selection into a closed-loop control system was presented in [8], where the authors designed three controllers so as to capture three "behaviors" representative of typical maneuvers that would be performed in a port environment. However, none of the works mentioned above consider disturbances and unmodeled terms of the vehicle. In order to ensure that the vehicle is robust to external disturbances, in [9], two controllers with application to a surface vehicle (named Qboat) and a hovercraft were proposed, and the authors designed disturbance estimators to estimate external constant disturbances. The disadvantage of this control strategy is that they did not consider unmodeled terms involved in the dynamic model of the vehicle. Considering this constraint, an estimator was developed in [10], where a fuzzy system was used to approximate unknown kinetics. A fault tolerant tracking controller was designed in [11] for a surface vessel. In addition, a self-constructing adaptive robust fuzzy neural control scheme for tracking surface vessels was proposed in [12], where simulation results were shown to testify the efficiency of the proposed method therein.

With respect to coordinated control strategy for multiple vehicles, many authors have presented their own approaches. In [13], a cooperative path following methodology was proposed under the assumption that the communication among a group of fully-actuated surface vehicles is undirected and continuous. A coordinated path following with a switching communication topology was designed in [14], while a null-space-based behavioral control technique was proposed in [15,16]. In [17–19], a leader–follower control strategy was presented. In [20], an adaptive coordinate tracking control problem for a fleet of nonholonomic chained systems was discussed under the assumption that the desired trajectory is available only to part of the neighbors. The reader is also referred to [21] for more results about multi-vehicle control approaches.

Inspired and motived by those works mentioned above, in this paper, we first develop a controller that is able to drive a single hovercraft to the neighborhood of a desired smooth path, where a Radial Basis Function Neural Network (RBFNN) is applied to approximate unmodeled dynamic terms of the vehicle while integral error terms are introduced, thus improving the robust performance of the controller. It is relevant to point out that all elements of the estimation weight matrix are always bounded through the use of a smooth projection function. We also derive a consensus strategy to make sure the desire paths progress in a specific formation. In order to validate the effectiveness of the proposed strategy, simulation results are presented.

The rest of the paper is organized as follows: Section 2 presents robot modeling, graph theory, RBFNNs, and coordinated control problem. A single controller is proposed in Section 3, while Section 4 devises a consensus strategy. Simulation results are given in Section 5 to validate the performance of the proposed approach herein. At last, Section 6 summarizes our work and describes the future work.

2. Problem Formulation

2.1. Vehicle Modeling

We first define a global coordinate frame $\{U\}$ and a body frame $\{B\}$ as shown in Figure 1. The kinematic equations of the vehicle are written as

$$\dot{\mathbf{p}} = \mathbf{R}\mathbf{v} \tag{1}$$

$$\dot{\psi} = \omega \tag{2}$$

where $\dot{\mathbf{p}} = [x, y]^T$ denotes the coordinates of its center of mass, $\mathbf{v} = [u, v]^T$ represents linear velocity, ψ is the orientation of the vehicle, and its angular velocity is represented by ω. Moreover, the rotation matrix $\mathbf{R}(\psi)$ is given by

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}. \tag{3}$$

Its dynamic equations are

$$\dot{\mathbf{v}} = -\mathbf{S}(\omega)\mathbf{v} + m^{-1}T\mathbf{n}_1 + \mathbf{\Delta}_v \tag{4}$$

$$\dot{\omega} = -J^{-1}\omega + J^{-1}\tau + \Delta_\omega \tag{5}$$

where $\mathbf{S}(\omega)$ is a skew symmetric matrix, given by

$$\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}. \tag{6}$$

$\mathbf{n}_1 = [1,\ 0]^T$, m and J denote the car's mass and rotational inertial, respectively. The force used to make the car move forward is denoted by T, and τ represents the torque that can steer the vehicle. Unmodeled dynamic terms are represented by $\mathbf{\Delta}_v$ and Δ_ω. For more details about modeling surface vehicles, the reader is referred to [22].

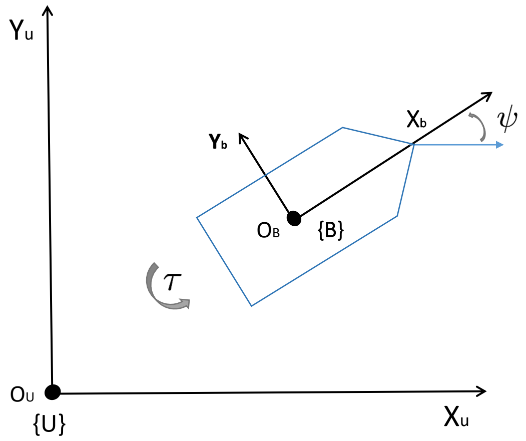


Figure 1. Simple Model of The Vehicle.

2.2. Graph Theory

In this paper, $\mathcal{G} = \mathcal{G}(\mathcal{V}, \mathcal{E})$ denotes a directed graph that can be used to model the interaction communication topology among mobile robots. The graph $\mathcal{G}$ consists of a finite set $\mathcal{V} = \{1, 2, ..., n\}$ of n vehicles and a finite set $\mathcal{E}$ of m pairs of vertices $\mathcal{V}_{ij} = \{i, j\} \in \mathcal{E}$. If $\mathcal{V}_{ij}$ belongs to $\mathcal{E}$, then i and j are said to be adjacent. A graph from i to j is a sequence of distinct vertices starting with i and ending with j such that consecutive vertices are adjacent. In this case, $\mathcal{V}_{ij}$ also represents a directional communication link from agent i to agent j. The adjacency matrix of the graph $\mathcal{G}$ is denoted by $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$, which is a square matrix where a_{ij} equals to one if $\{j,\ i\} \in \mathscr{E}$ and zero otherwise. Moreover, the Laplacian matrix $\mathcal{L}$ is defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$, where the degree matrix $\mathcal{D} = [d_{ij}] \in \mathbb{R}^{n\times n}$ of the graph $\mathcal{G}$ is a diagonal matrix and d_{ij} equals the number of adjacent vertices of vertex i.

2.3. Radial Basis Function Neural Networks

Radial Basis Function Neural Networks (RBFNNs) can be used to approximate the unmodeled nonlinear dynamic terms due to their universal approximation capability [23]. For any unknown smooth function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ can be approximated by RBFNNs in the following form, given by

$$\hat{f}(\mathbf{x}) = \mathbf{W}^T\sigma(\mathbf{x}) \tag{7}$$

where $\mathbf{x} \in \Omega \subset \mathbb{R}^n$, Ω is a compact set. The adjustable weight matrix with n neurons is denoted by $\mathbf{W} \in \mathbb{R}^{n\times m}$ under the assumption that it is a bounded matrix, that is

$$\mathbf{W} \leq \mathbf{W}_{\max}. \tag{8}$$

It is important to point out that here when we say matrix $\mathbf{x} \in \mathbb{R}^{m\times n}$ is smaller than or equal to $\mathbf{x}_{\max} \in \mathbb{R}^{m\times n}$, we mean all elements of $\mathbf{x}$ are smaller than or equal to their corresponding elements of $\mathbf{x}_{\max} \in \mathbb{R}^{m\times n}$. Moreover, $\boldsymbol{\sigma}(\mathbf{x})$ is the basis function vector and $\sigma_i(x_i) = \exp(-(x_i-\mu_i)^T(x_i-\mu_i)/c_i^2)$, $i = 1, 2, \ldots, n$ denotes its component, μ_i is the center of the receptive field, and c_i represents the width of the Gaussian function. Moreover, it is relevant to point out that, in order to achieve better approximate results, we should make the neuron number n large enough and choose the parameters properly. Going back to the smooth function $f(\mathbf{x})$ mentioned above, there is an ideal weight $\mathbf{W}_d$ such that

$$f(\mathbf{x}) = \mathbf{W}_d^T\boldsymbol{\sigma}(\mathbf{x}) + \boldsymbol{\epsilon}(\mathbf{x}) \tag{9}$$

where $\boldsymbol{\epsilon}(\mathbf{x})$ denotes the approximation error and satisfies $||\boldsymbol{\epsilon}(\mathbf{x})|| \leq \epsilon_{\max}$, where $\epsilon_{\max}$ is a positive number. It is noted that $\mathbf{W}_d$ is an "artificial" quantity for the purpose of mathematic analysis, in the process of controller design, we need to estimate it [24]. A simple RBFNNs is given by Figure 2.

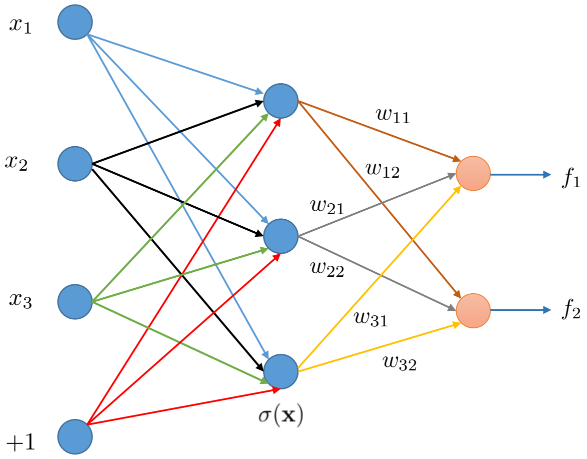


Figure 2. Simple example of RBFNNs.

2.4. Problem Statement

Now we can state our problem: Through designing a controller for each robot and proposing consensus strategies for their corresponding desired paths, we want a group of mobile robots to move forward in a specific formation pattern; that is,

(1) for an individual vehicle, $||\mathbf{p} - \mathbf{p}_d|| \to \delta$, where δ is an arbitrarily small constant value;

(2) for a group of n desired paths, $\gamma_i - \gamma_j \to 0$ and $\dot{\gamma}_i - \dot{\gamma}_d \to 0$, where agent j is the neighbor of agent i, $\dot{\gamma}_d$ represents the desired value of $\dot{\gamma}_i$, which is a known value.

3. Controller Design

Following the works of [7] and [9], we define the position error in the body frame as

$$\mathbf{e}_1 = \mathbf{R}^T(\mathbf{p} - \mathbf{p}_d) \tag{10}$$

where $\mathbf{R} = \mathbf{R}(\psi)$ denotes the rotation matrix. The time derivative of $\mathbf{e}_1$ yields

$$\dot{\mathbf{e}}_1 = -\mathbf{S}(\omega)\mathbf{e}_1 + \mathbf{v} - \mathbf{R}^T \dot{\mathbf{p}}_d. \tag{11}$$

Define our first Lyapunov function as

$$V_1 = \frac{1}{2}\mathbf{e}_1^T \mathbf{e}_1 \tag{12}$$

and compute its time derivative, we have

$$\dot{V}_1 = -\alpha_1 + \mathbf{e}_1^T(\mathbf{v} - \mathbf{R}^T \dot{\mathbf{p}}_d + k_1 \mathbf{e}_1) \tag{13}$$

where $\alpha_1 = k_1 \mathbf{e}_1^T \mathbf{e}_1$ is positive definite, and k_1 denotes gain, which is a positive value.

In order to continue to use the backstepping technique, a second error term is defined as

$$\mathbf{e}_2 = \mathbf{v} - \mathbf{R}^T \dot{\mathbf{p}}_d + k_1 \mathbf{e}_1 - \boldsymbol{\eta}, \tag{14}$$

and its integral term

$$\mathbf{e}_{2n} = \int_0^t \mathbf{e}_2 dt \tag{15}$$

where $\boldsymbol{\eta} = [\eta_1, \eta_2]^T$, $\eta_1 \neq 0$ is a constant vector. Define our second Lyapunov function as

$$V_2 = V_1 + \frac{1}{2}\mathbf{e}_2^T \mathbf{e}_2 + \frac{1}{2}\mathbf{e}_{2n}^T \mathbf{e}_{2n}, \tag{16}$$

and its time derivative yields

$$\begin{aligned} \dot{V}_2 = -\alpha_2 + \mathbf{e}_1^T \boldsymbol{\eta} + \mathbf{e}_2^T \Big(& -\mathbf{S}(\omega)\boldsymbol{\eta} + m^{-1} T \mathbf{n}_1 - \mathbf{R}^T \ddot{\mathbf{p}}_d + k_1(\mathbf{e}_2 - k_1 \mathbf{e}_1 + \boldsymbol{\eta}) \\ & + k_2 \mathbf{e}_2 + \boldsymbol{\Delta}_v + \mathbf{e}_{2n} \Big) \end{aligned} \tag{17}$$

where $\alpha_2 = k_1 \mathbf{e}_1^T \mathbf{e}_1 + k_2 \mathbf{e}^T \mathbf{e}_2$, which is positive definite, and k_2 is a positive number. It is relevant to point out that $\mathbf{e}_{2n}$ is introduced to eliminate the external slow-varying disturbances that act on the dynamic of the linear velocity $\mathbf{v}$. Moreover, notice that we do not know $\boldsymbol{\Delta}_v$, thereby we use RBFNNs mentioned before to approximate it, given by

$$\boldsymbol{\Delta}_v = \mathbf{W}_{d1}^T \sigma(\mathbf{x}_1) + \epsilon_1(\mathbf{x}_1) \tag{18}$$

where $\mathbf{x}_1 = [1, \mathbf{v}^T]^T \in \mathbb{R}^3$. Moreover, notice that

$$-\mathbf{S}(\omega)\boldsymbol{\eta} + m^{-1} T \mathbf{n}_1 = \mathbf{N}\mathbf{I} \tag{19}$$

where

$$\mathbf{N} = \begin{bmatrix} m^{-1} & \eta_2 \\ 0 & -\eta_1 \end{bmatrix}. \quad \mathbf{I} = [T, \omega]^T.$$

Rewrite Equation (17), and we have

$$\dot{V}_2 = -\alpha_2 + \mathbf{e}_1^T \boldsymbol{\eta} + \mathbf{e}_2^T \Big(\mathbf{N}\mathbf{I} + \boldsymbol{\beta} + \mathbf{W}_{d1}^T \sigma(\mathbf{x}_1) \Big) + \mathbf{e}_2^T \epsilon_1(\mathbf{x}_1) \tag{20}$$

where $\boldsymbol{\beta} = -\mathbf{R}^T \ddot{\mathbf{p}}_d + k_1(\mathbf{e}_2 - k_1 \mathbf{e}_1 + \boldsymbol{\eta}) + k_2 \mathbf{e}_2 + \mathbf{e}_{2n}$.

Now, we can define our third Lyapunov function as

$$V_3 = V_2 + \frac{1}{2}\,\mathrm{tr}\left(\widetilde{\mathbf{W}}_{d1}^T \Gamma_1^{-1} \widetilde{\mathbf{W}}_{d1}\right) \tag{21}$$

where $\widetilde{\mathbf{W}}_{d1} = \mathbf{W}_d - \widehat{\mathbf{W}}_{d1}$ denotes the estimation error, $\Gamma_1 = \mathrm{diag}(\lambda_{11}, and\lambda_{12})$ is a matrix, where λ_{11} and λ_{12} are positive values. Compute the time derivative of V_3, and one obtains

$$\begin{aligned} \dot{V}_3 = &-\alpha_2 + \mathbf{e}_1^T \boldsymbol{\eta} + \mathbf{e}_2^T \epsilon_1(\mathbf{x}_1) + \mathbf{e}_2^T \left(\mathbf{NI} + \boldsymbol{\beta} + \widehat{\mathbf{W}}_{d1}^T \sigma(\mathbf{x}_1)\right) \\ &+ \mathbf{e}_2^T \widetilde{\mathbf{W}}_{d1}^T \sigma(\mathbf{x}_1) - \mathrm{tr}\left(\widetilde{\mathbf{W}}_{d1}^T \Gamma_1^{-1} \dot{\widehat{\mathbf{W}}}_{d1}\right). \end{aligned} \tag{22}$$

Therefore, we choose our desired input $\mathbf{I}_d$ as

$$\mathbf{I}_d = -\mathbf{N}^{-1}\left(\boldsymbol{\beta} + \widehat{\mathbf{W}}_{d1}^T \sigma(\mathbf{x}_1)\right), \tag{23}$$

and thereby our first controller T is chosen as

$$T = \mathbf{n}_1^T \mathbf{I}_d. \tag{24}$$

Correspondingly, the desired angular velocity is

$$\omega_d = \mathbf{n}_2^T \mathbf{I}_d \tag{25}$$

where $\mathbf{n}_2 = [0, 1]^T$.

Notice that, if the updated law for $\dot{\widehat{\mathbf{W}}}_{d1}$ is set as

$$\dot{\widehat{\mathbf{W}}}_{d1} = \Gamma_1 \sigma(\mathbf{x}_1) \mathbf{e}_2^T, \tag{26}$$

we cannot ensure that it is bounded by $\mathbf{W}_{\max}$. To solve this problem, a projection operator, which is Lipschitz continuous [25], is applied in our case, which is given by

$$\mathbf{proj}(\rho, \hat{\ell}) = \begin{cases} \rho & \text{if } \Theta(\hat{\ell}) \le 0 \\ \rho & \text{if } \Theta(\hat{\ell}) \ge 0 \text{ and } \Theta_{\hat{\ell}}(\hat{\ell})\rho \le 0 \\ (1 - \Theta(\hat{\ell}))\rho & \text{if } \Theta(\hat{\ell}) > 0 \text{ and } \Theta_{\hat{\ell}}(\hat{\ell})\rho > 0 \end{cases} \tag{27}$$

where

$$\Theta(\hat{\ell}) = \frac{\hat{\ell}^2 - \ell_{\max}^2}{\epsilon^2 + 2\epsilon\ell_{\max}^2}, \quad \Theta_{\hat{\ell}}(\hat{\ell}) = \frac{\partial\Theta(\hat{\ell})}{\partial\hat{\ell}}, \tag{28}$$

with the following condition: if $\dot{\hat{\ell}} = proj(\rho, \hat{\ell})$ and $\hat{\ell}(t_0) \le \ell_{\max}$, then

(1) $\hat{\ell} \le \ell_{\max} + \epsilon, \forall\, 0 \le t < \infty$;
(2) $\mathbf{proj}(\rho, \hat{\ell})$ is Lipschitz continuous;
(3) $|\mathbf{proj}(\rho, \hat{\ell})| \le |\rho|$;
(4) $\tilde{\ell}\mathbf{proj}(\rho, \hat{\ell}) \ge \tilde{\ell}\rho$, where $\tilde{\ell} = \ell - \hat{\ell}$.

Therefore, to make sure all elements of $\hat{\mathbf{W}}_{d1}$ are upper-bounded, the update law for $\dot{\hat{\mathbf{W}}}_{d1}$ is finally set as

$$\dot{\widehat{\mathbf{W}}}_{d1} = \Gamma_1 \mathbf{proj}\left(\sigma(\mathbf{x}_1)\mathbf{e}_2, \widehat{\mathbf{W}}_{d1}\right). \tag{29}$$

To keep using the backstepping technique, we define a new error term

$$\mathbf{e}_3 = \omega - \omega_d, \tag{30}$$

and its corresponding integral term

$$\mathbf{e}_{3n} = \int_0^t \mathbf{e}_3 dt. \tag{31}$$

Then, we define a new Lyapunov function as

$$V_4 = V_3 + \frac{1}{2}\mathbf{e}_3^T\mathbf{e}_3 + \frac{1}{2}\mathbf{e}_{3n}^T\mathbf{e}_{3n}. \tag{32}$$

Compute its time derivatives, substitute Equation (29) into Equation (24), and combine the 4th property of projector ($\tilde{\ell}\mathbf{proj}(\rho, \hat{\ell}) \geq \tilde{\ell}\rho$), and one obtains

$$\begin{aligned} \dot{V}_4 \leq & -\alpha_3 + \mathbf{e}_1^T\boldsymbol{\eta} + \mathbf{e}_2^T\boldsymbol{\epsilon}_1(\mathbf{x}_1) + \mathbf{e}_3^T\Big(\mathbf{n}_2^T\mathbf{N}^T\mathbf{e}_2 - \frac{1}{J}\tau + \mathbf{n}_2^T\mathbf{N}^{-1}(\dot{\boldsymbol{\beta}} + \dot{\widehat{\mathbf{W}}}_{d1}\boldsymbol{\sigma}(\mathbf{x}_1) \\ & + \widehat{\mathbf{W}}_{d1}\dot{\boldsymbol{\sigma}}(\mathbf{x}_1)) + k_3\mathbf{e}_3 + \mathbf{e}_{3n} + \Delta_\omega\Big) \end{aligned} \tag{33}$$

where $\alpha_3 = \alpha_2 + k_3\mathbf{e}_3^T\mathbf{e}_3 \geq 0, k_3 > 0$. However, it is noted that both $\dot{\boldsymbol{\beta}}$ and $\dot{\boldsymbol{\sigma}}(\mathbf{x}_1)$ contain unmodeled term $\boldsymbol{\Delta}_v$, so we need to separate $\boldsymbol{\Delta}_v$ out from $\dot{\boldsymbol{\beta}}$ and $\dot{\boldsymbol{\sigma}}(\mathbf{x}_1)$. After that, we can use RBFNNs to estimate it. Similar to Δ_ω, we also need to approximate it. This is given by,

$$\boldsymbol{\Delta}_v = \mathbf{W}_{d2}^T\boldsymbol{\sigma}(\mathbf{x}_2) + \boldsymbol{\epsilon}_2(\mathbf{x}_2) \tag{34}$$

$$\boldsymbol{\Delta}_\omega = \mathbf{W}_{d3}^T\boldsymbol{\sigma}(\mathbf{x}_3) + \boldsymbol{\epsilon}_3(\mathbf{x}_3) \tag{35}$$

where $\mathbf{x}_2 = [1, \mathbf{v}^T]^T \in \mathbb{R}^3$ and $\mathbf{x}_3 = [1, \mathbf{v}^T, \omega]^T \in \mathbb{R}^4$.

Now we define our last Lyapunov function as

$$V_5 = V_4 + \frac{1}{2}\operatorname{tr}\left(\widetilde{\mathbf{W}}_{d2}^T\Gamma_2^{-1}\widetilde{\mathbf{W}}_{d2}\right) + \frac{1}{2}\operatorname{tr}\left(\widetilde{\mathbf{W}}_{d3}^T\Gamma_3^{-1}\widetilde{\mathbf{W}}_{d3}\right) \tag{36}$$

where $\widetilde{\mathbf{W}}_{d2} = \mathbf{W}_d - \widehat{\mathbf{W}}_{d2}, \widetilde{\mathbf{W}}_{d3} = \mathbf{W}_d - \widehat{\mathbf{W}}_{d3},\ \Gamma_2 = \operatorname{diag}(\lambda_{21},\lambda_{22})$, and $\Gamma_3 = \operatorname{diag}(\lambda_{31}, \lambda_{32})$ are positive definite gain matrices. Then, we compute the time derivative of V_5 as

$$\begin{aligned} \dot{V}_5 \leq & -\alpha_3 + \mathbf{e}_1^T\boldsymbol{\eta} + \mathbf{e}_2^T\boldsymbol{\epsilon}_1(\mathbf{x}_1) + \mathbf{e}_3^T(\mathbf{n}_2^T\mathbf{M})\boldsymbol{\epsilon}_2(\mathbf{x}_2) + \mathbf{e}_3^T\boldsymbol{\epsilon}_3(\mathbf{x}_3) + \mathbf{e}_3^T\Big(\mathbf{n}_2^T\mathbf{N}^T\mathbf{e}_2 + k_3\mathbf{e}_3 \\ & - \frac{\tau}{J} - \mathbf{n}_2^T\hat{\dot{\mathbf{I}}}_d - \mathbf{n}_2^T\mathbf{M}\widehat{\mathbf{W}}_{2d}^T\boldsymbol{\sigma}(\mathbf{x}_2) + \widehat{\mathbf{W}}_{3d}^T\boldsymbol{\sigma}(\mathbf{x}_3) + \mathbf{e}_{3n}\Big) + \mathbf{e}_3^T\mathbf{n}_2^T\mathbf{M}\widetilde{\mathbf{W}}_{2d}^T\boldsymbol{\sigma}(\mathbf{x}_2) \\ & + \mathbf{e}_3^T\widetilde{\mathbf{W}}_3^T\boldsymbol{\sigma}(\mathbf{x}_3) - \operatorname{tr}\left(\widetilde{\mathbf{W}}_{d2}^T\Gamma_2^{-1}\dot{\widehat{\mathbf{W}}}_{d2}\right) - \operatorname{tr}\left(\widetilde{\mathbf{W}}_{d3}^T\Gamma_3^{-1}\dot{\widehat{\mathbf{W}}}_{d3}\right) \end{aligned} \tag{37}$$

where

$$\hat{\dot{\mathbf{I}}}_d = \hat{\dot{\boldsymbol{\beta}}} + \dot{\widehat{\mathbf{W}}}_{d1}\boldsymbol{\sigma}(\mathbf{x}_1) + \widehat{\mathbf{W}}_{d1}\hat{\dot{\boldsymbol{\sigma}}}(\mathbf{x}_1) \tag{38}$$

and

$$\mathbf{M} = \mathbf{N}^{-1}\widehat{\mathbf{W}}_{d1}^T\mathbf{G} + (k_1 + k_2)\mathbf{N}^{-1} \tag{39}$$

with $\mathbf{G} = [\mathbf{0}_{2\times 1}^T, \mathbf{n}_1\sigma'(\mathbf{n}_1^T\mathbf{v}), and \mathbf{n}_1\sigma'(\mathbf{n}_2^T\mathbf{v})]^T$. Then, we define our second control law, torque, as

$$\tau = J(-\mathbf{n}_2^T\hat{\dot{\mathbf{I}}}_d + \mathbf{n}_2^T\mathbf{N}^T\mathbf{e}_2 + k_3\mathbf{e}_3 + \mathbf{n}_2^T\mathbf{M}\widehat{\mathbf{W}}_{d2}^T\boldsymbol{\sigma}(\mathbf{x}_2) + \widehat{\mathbf{W}}_{d3}^T\boldsymbol{\sigma}(\mathbf{x}_3) + \mathbf{e}_{3n}), \tag{40}$$

and estimate laws for $\dot{\widehat{\mathbf{W}}}_{d2}$ and $\dot{\widehat{\mathbf{W}}}_{d3}$

$$\dot{\widehat{\mathbf{W}}}_{d2} = \Gamma_2 \mathbf{proj}\Big(\sigma(\mathbf{x}_2)(\mathbf{n}_2^T\mathbf{M})\mathbf{e}_3^T\,\widehat{\mathbf{W}}_{d2}\Big), \tag{41}$$

$$\dot{\widehat{\mathbf{W}}}_{d3} = \Gamma_3 \mathbf{proj}\Big(\sigma(\mathbf{x}_3)\mathbf{e}_3^T,\,\widehat{\mathbf{W}}_{d3}\Big). \tag{42}$$

Substitute Equation (40)–(42) into Equation (37), one obtains

$$\dot{V}_5 \leq -\alpha_3 + \mathbf{e}_1^T\boldsymbol{\eta} + \mathbf{e}_2^T\boldsymbol{\epsilon}_1(\mathbf{x}_1) + \mathbf{e}_3^T(\mathbf{n}_2^T\mathbf{M})\boldsymbol{\epsilon}_2(\mathbf{x}_2) + \mathbf{e}_3^T\boldsymbol{\epsilon}_3(\mathbf{x}_3). \tag{43}$$

where $\alpha_3 = k_1\mathbf{e}_1^T\mathbf{e}_1 + k_2\mathbf{e}_2^T\mathbf{e}_2 + k_3\mathbf{e}_3^T\mathbf{e}_3$.

3.1. Stability Analysis

Theorem 1. *For a single mobile robot, by applying control laws, Equations (24) and (40), and updated laws, Equation (29), Equation (41), and Equation (42), for any large initial position, the robot will converge to the neighborhood of its corresponding desired path $\mathbf{p}_d(\gamma)$, whose partial derivatives with respect to γ are all bounded. As a consequence, global uniformly ultimately boundedness is achieved.*

Proof. Let's go back to Equation (43). Rewrite it, and we obtain

$$\begin{aligned}\dot{V}_5 &\leq -\mathbf{X}^T\mathbf{K}\mathbf{X} + \mathbf{X}^T\boldsymbol{\rho}\\ &\leq |\mathbf{X}|^T\Big(-k_{\min}|\mathbf{X}| + |\boldsymbol{\rho}|\Big)\end{aligned} \tag{44}$$

where $\mathbf{X} = [\mathbf{e}_1^T, \mathbf{e}_2^T, \mathbf{e}_3^T]$, $k_{\min}$ is the smallest eigenvalue of $\mathbf{K} = \text{diag}(k_1, k_2, k_3)$, and $\boldsymbol{\rho} = [\boldsymbol{\eta}^T, \boldsymbol{\epsilon}_1(\mathbf{x}_1)^T, (\mathbf{n}_2^T\mathbf{M})\boldsymbol{\epsilon}_2(\mathbf{x}_2), \boldsymbol{\epsilon}_3(\mathbf{x}_3)]^T$, which is bounded due to the fact that $||\boldsymbol{\epsilon}_i(\mathbf{x})|| \leq \epsilon_{\max}$, and the upper bound of $\boldsymbol{\rho}$ is

$$\boldsymbol{\rho}_{\max} = [\boldsymbol{\eta}^T, \boldsymbol{\epsilon}_{1\max}(\mathbf{x}_1)^T, (\mathbf{n}_2^T\mathbf{M})\boldsymbol{\epsilon}_{2\max}(\mathbf{x}_2), \boldsymbol{\epsilon}_{3\max}(\mathbf{x}_3)]^T. \tag{45}$$

Thereby, we can obtain that $\dot{V}_5$ is negative for $||\mathbf{X}|| \geq ||\boldsymbol{\rho}_{\max}/k_{\min}||$, which can be made as small as possible by tuning the value of $k_{\min}$. As a result, the system is uniformly ultimate bounded, global uniformly ultimate boundedness is achieved. □

4. Consensus Strategy

Building upon the work of [14], the proposed solution is given by

$$\dot{\gamma}_i = v_d\mathbf{1} - a_1\sum_{j\in N_i}(\gamma_i - \gamma_j) + z_i \tag{46}$$

$$\dot{z}_i = -a_2 z_i + a_3\sum_{j\in N_i}(\gamma_i - \gamma_j) \tag{47}$$

where $a_1, a_2, and a_3$ are positive numbers, and v_d denotes the desired value of $\dot{\gamma}_i$. It is relevant to point out that z_i can be viewed as an auxiliary state that helps n paths to reach consensus.

4.1. Stability Analysis

Theorem 2. *For a group of n desired paths $\mathbf{p}_d(\gamma_i)$ $(i = 1, 2, \ldots, n)$, by applying Equations (46) and (47), $\gamma_i - \gamma_j$ and $\dot{\gamma}_i - \dot{\gamma}_d$ converge to zero.*

Proof. We first choose Laplacian matrix $\mathcal{L}$, and define the coordinate error as

$$\Gamma_e = \mathcal{L}\boldsymbol{\Lambda} \tag{48}$$

where $\mathbf{\Lambda} = [\gamma_i]_{n\times 1}$. Rewrite Equations (46) and (47), and one obtains

$$\dot{\Gamma}_e = -\mathbf{A}_1\mathcal{L}\Gamma_e + \mathcal{L}\mathbf{Z}, \tag{49}$$

$$\dot{\mathbf{Z}} = -\mathbf{A}_2\mathbf{Z} + \mathbf{A}_3\Gamma_e, \tag{50}$$

where $\mathbf{A}_i = \mathrm{diag}(a_i), (i = 1,2,3)$ is a positive definite matrix. Define $\mathbf{x} = [\Gamma_e, \mathbf{Z}]^T$, and rewrite Equations (49) and (50), and we have

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{51}$$

where

$$\mathbf{A} = \begin{bmatrix} -\mathbf{A}_1\mathcal{L} & \mathcal{L} \\ \mathbf{A}_3 & -\mathbf{A}_2 \end{bmatrix}. \tag{52}$$

In order to ensure that Equation (51) is stable, we need to guarantee that all the eigenvalues of **A** have negative or zero real parts and all the Jordan blocks corresponding to eigenvalues with zero real parts are 1 × 1. We consider n agents, where

$$\mathcal{L} = \begin{vmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{vmatrix}_{n\times n}. \tag{53}$$

For the sake of saving space, here we just present the eigenvalues of **A** directly, they are λ_1, λ_2, λ_3, and λ_4, and their corresponding multiplicities are 1, 1, $n-1$, and $n-1$, with

$$\lambda_1 = 0 \tag{54}$$

$$\lambda_2 = -a_2 \tag{55}$$

$$\lambda_3 = -(a_1 + a_2)/2 + \sqrt{a_1^2 + a_2^2 - 2a_1a_2 + 4a_3}/2 \tag{56}$$

$$\lambda_4 = -(a_1 + a_2)/2 - \sqrt{a_1^2 + a_2^2 - 2a_1a_2 + 4a_3}/2. \tag{57}$$

Choosing $a_1, a_2, and a_3$ properly, we can guarantee that λ_2, λ_3, and λ_4 are negative-definite. Moreover, we also have that the Jordan block corresponding to λ_1 is 1 × 1. As a consequence, Equation (51) is stable [26]. □

To summarize, a fleet of n desired paths can progress in a specific formation, while each individual mobile robot converges to the neighborhood of its corresponding desired path. As a result, all those robots can move forward in a specific formation.

5. Simulation Results

In this section, we present simulation results with two different communication graphs including a cascade-directed communication graph (CDCG) and a parallel-directed communication graph (PDCG). Figures 3 and 4 show the sketches of the control blocks in Simulink/Matlab.

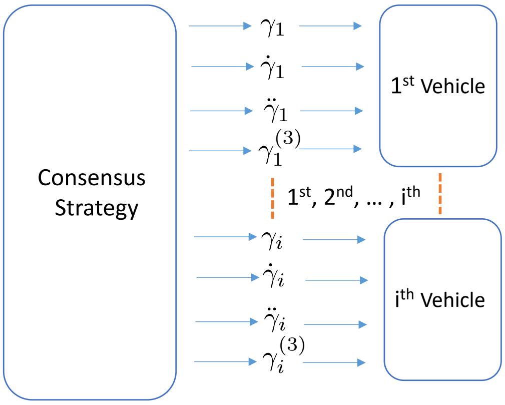


Figure 3. Coordinated control block in Simulink/Matlab.

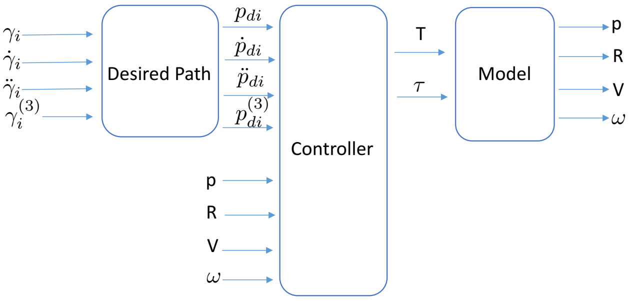


Figure 4. Control block in Simulink/Matlab for the *i*-th vehicle.

5.1. The Cascade-Directed Communication Graph

The CDCG used in this study is shown in Figure 5, where agent 1 can be viewed as the leader. Its corresponding Laplacian matrix $\mathscr{L}_1$ is

$$\mathscr{L}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}. \tag{58}$$

The desired paths are defined as

$$\mathbf{p}_{di}(\gamma_i) = R_i \begin{bmatrix} \cos(\gamma_i) \\ \sin(\gamma_i) \end{bmatrix} \text{ (m)} \tag{59}$$

where $R_i = 6 - i$, $(i = 1, 2, \ldots, 5)$ denotes the radius of the circle, and unmodeled terms $\boldsymbol{\Delta}_V$ and Δ_ω are denoted by $[0.1u^2 + 0.01uv, 0.01uv + 0.1v^2]^T$ and $0.01uv + 0.05u\omega + 0.01v\omega$, respectively. The parameters used herein are as follows: $m = 0.6$, $J = 0.1$, $c_i = 2$, $a_1 = 1$, $a_2 = 2$, $a_3 = 1$, $k_1 = 6$, $k_2 = 3$, $k_3 = 2$, $\Gamma_1 = \text{diag}(0.6, 0.6)$, $\Gamma_2 = \text{diag}(0.01, 0.01)$, $\Gamma_3 = \text{diag}(0.022, 0.022)$, and $\boldsymbol{\eta} = [0.2, 0]^T$.

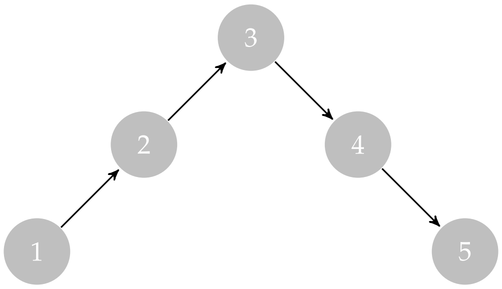
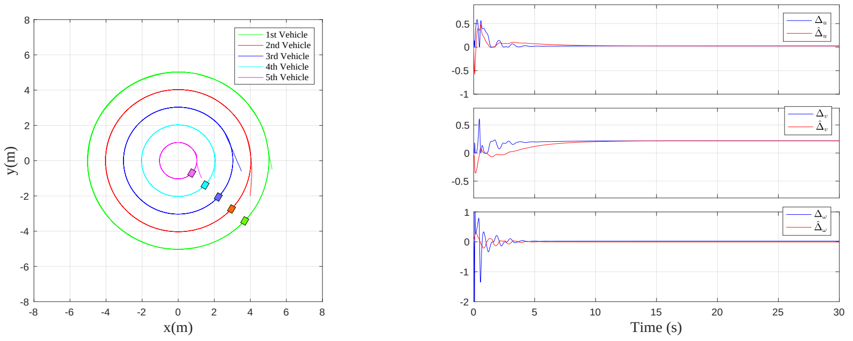


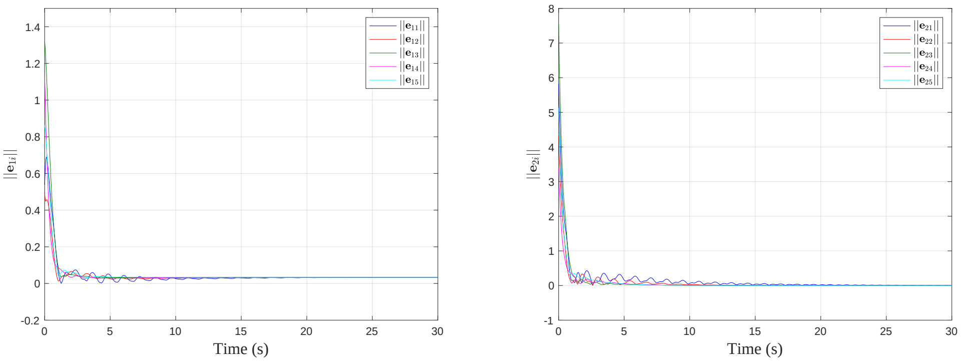
Figure 5. A cascade-directed communication graph (CDCG).

Figure 6a shows the actual trajectory of the mobile robots, and we can see that they move forward in a line formation. Moreover, Figures 7a,b display the convergence of $||\mathbf{e}_{1i}||$ and $||\mathbf{e}_{2i}||$, respectively, showing that all of them converge to the neighborhood of zero. The consensus performances are shown in Figures 8a,b, showing that $\gamma_{ij} = \gamma_i - \gamma_j$ converges to zero, where agent j is the neighbor of agent i. Moreover, we can also see that $\dot{\gamma}_i$ converges to the desired value $\dot{\gamma}_d = 0.5$. It is important to point out that, in this work, we chosen agent 1 as our leader, and this satisfies $\dot{\gamma}_1 = \dot{\gamma}_d$. The approximate performance of RBFNNs can be found in Figure 6b, where the blue lines denote the real values of $\mathbf{\Delta}_v$ and Δ_ω, while the red lines represent their estimates $\hat{\mathbf{\Delta}}_v$ and $\hat{\Delta}_\omega$. Thus, both estimates converge to their corresponding real values.



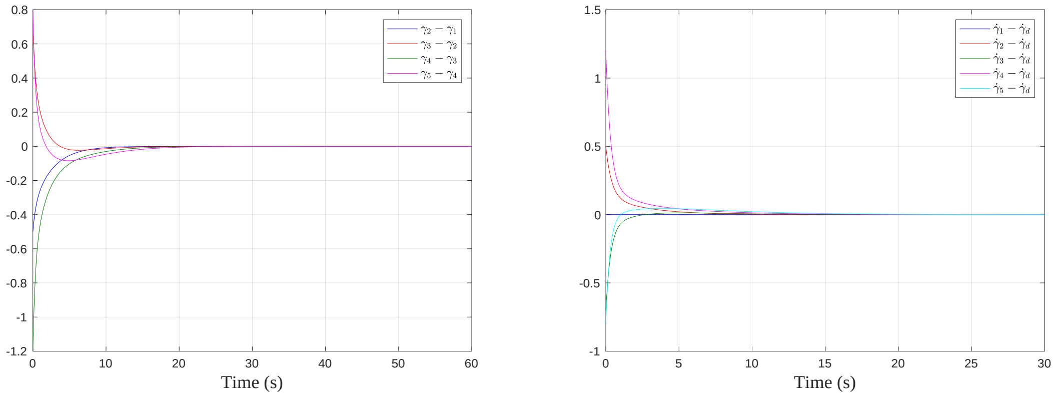
(**a**) (**b**)

Figure 6. Norm of the position errors and the performance of unmodeled term approximation (CDCG). (**a**) Norm of the position errors. (**b**) Performance of unmodeled term approximation.



(**a**) (**b**)

Figure 7. Norm of the position and linear velocity errors (CDCG). (**a**) Norm of the position errors. (**b**) Norm of the linear velocity errors.



(**a**) (**b**)

Figure 8. Performance of $\gamma_i - \gamma_j$ and $\dot{\gamma}_i - \dot{\gamma}_d$, where γ_j is the neighbor of γ_i (CDCG). (**a**) Performance of $\gamma_i - \gamma_j$. (**b**) Performance of $\dot{\gamma}_i - \dot{\gamma}_d$.

5.2. Parallel Communication Graph

The parallel communication graph is depicted in Figure 9.

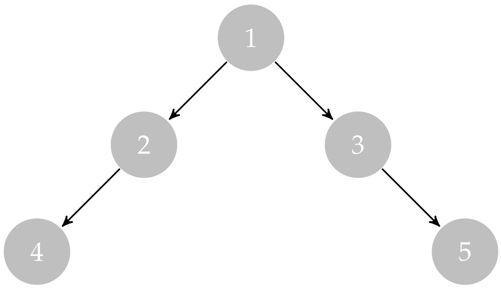


Figure 9. A parallel-directed communication graph (PDCG).

In this case, the Laplacian matrix $\mathscr{L}_2$ is

$$\mathscr{L}_2 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{60}$$

Moreover, it is noted that the graph presented in Figure 9 can be viewed as the combination of two cascade-directed graphs, where **1** → **2** → **4** and **1** → **3** → **5** and where agent **1** is the leader whose state is known. The desired paths are as follows:

(1) 1st vehicle:

$$\mathbf{p}_{d1}(\gamma_1) = R_1 \begin{bmatrix} \cos(\gamma_1) \\ \sin(\gamma_1) \end{bmatrix} \text{ (m)} \tag{61}$$

(2) 2nd vehicle:

$$\mathbf{p}_{d2}(\gamma_2) = R_2 \begin{bmatrix} \cos(\gamma_2 - \pi/24) \\ \sin(\gamma_2) - \pi/24 \end{bmatrix} \text{ (m)} \tag{62}$$

(3) 3rd vehicle:

$$\mathbf{p}_{d3}(\gamma_3) = R_3 \begin{bmatrix} \cos(\gamma_3 - \pi/12) \\ \sin(\gamma_3 - \pi/12) \end{bmatrix} \text{ (m)} \tag{63}$$

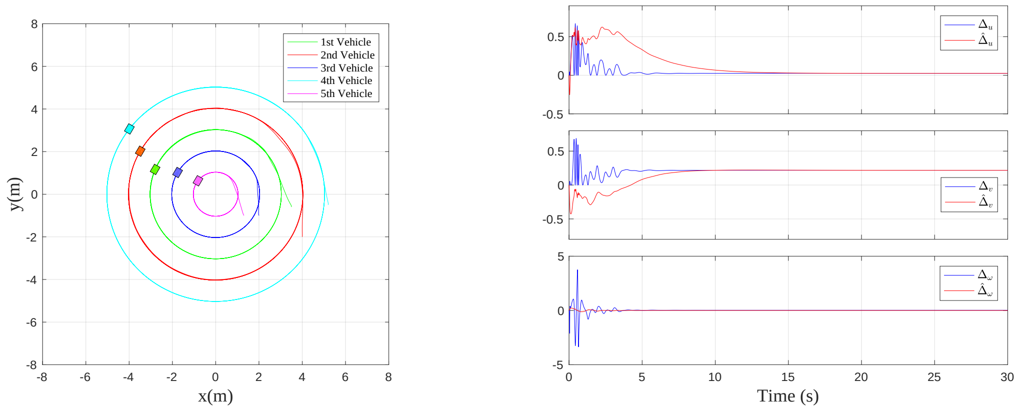
(4) 4th vehicle:

$$\mathbf{p}_{d4}(\gamma_4) = R_4 \begin{bmatrix} \cos(\gamma_4 - \pi/24) \\ \sin(\gamma_4 - \pi/24) \end{bmatrix} \text{ (m)} \tag{64}$$

(5) 5th vehicle:

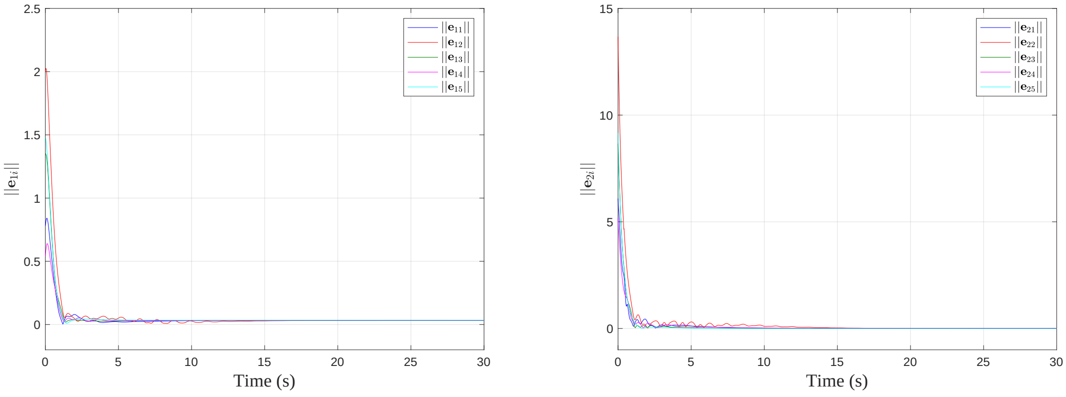
$$\mathbf{p}_{d5}(\gamma_5) = R_5 \begin{bmatrix} \cos(\gamma_5 - \pi/12) \\ \sin(\gamma_5 - \pi/12) \end{bmatrix} \text{ (m)} \tag{65}$$

where $R_1 = 3$, $R_2 = 4$, $R_3 = 2$, $R_4 = 5$, and $R_5 = 1$. Figure 10a displays the actual paths of the robots. From Figure 11a,b, we can obtain that the norm of position errors and linear velocity errors converge to a ball centered at the origin. Moreover, Figure 12a,b show the performance of the consensus strategy introduced herein.



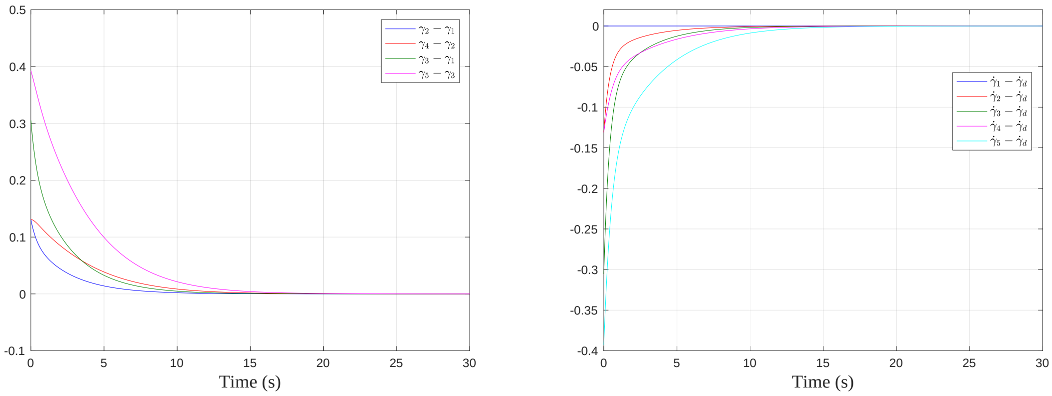
(**a**) (**b**)

Figure 10. Norm of the position errors and performance of unmodeled term approximation (PDCG). (**a**) Norm of the position errors. (**b**) Performance of unmodeled term approximation.



(**a**) (**b**)

Figure 11. Norm of the position and linear velocity errors (PDCG). (**a**) Norm of the position errors. (**b**) Norm of the linear velocity errors.



(**a**) (**b**)

Figure 12. Performance of $\gamma_i - \gamma_j$ and $\dot{\gamma}_i - \dot{\gamma}_d$, where γ_j is the neighbor of γ_i (PDCG). (**a**) Performance of $\gamma_i - \gamma_j$. (**b**) Performance of $\dot{\gamma}_i - \dot{\gamma}_d$.

It is interesting to remark that, by using the proposed consensus strategy—Equations (46) and (47)—a consensus is researched if and only if the directed graph has a directed spanning tree [21]. However, in our case, the root must be the leader whose states are known beforehand.

6. Conclusions

In this paper, we mainly focus on designing coordinated control algorithms for multiple agents, where a group of underactuated hovercrafts were chosen as test platforms. In order to testify the efficiency of the devised control strategy, we implemented it by using Simulink/Matlab. Moreover, it is necessary to point out that agents can also be mobile robots, unmanned air vehicles, etc. For a single vehicle, we used RBFNNs to approximate unmodeled terms and introduce integral terms, which can improve the robustness of the controller. For multiple vehicles, we consider directed topology under the assumption that the communication among vehicles are continuous.

With respect to our future works, we plan to (i) use deep neural networks to estimate unmodeled terms so as to enhance the performance of approximation, (ii) build a mathematical model for external disturbance, such as winds, waves, or currents, (iii) take into account time-delays when we develop communication strategy for vehicles, and (iv) propose collision–avoidance algorithms so that we can ensure the operation is safe.

Author Contributions: K.D. and S.F. conceived and designed the experiments; K.D. performed the experiments and wrote the paper; Y.Z. and W.S. provided guidance for this paper.

Acknowledgments: The authors are grateful for financial support from the following research grants: (1) the 'Nature-Inspired Computing and Metaheuristics Algorithms for Optimizing Data Mining Performance' grant from the University of Macau (grant no. MYRG2016-00069-FST); the (2) 'Improving the Protein–Ligand Scoring Function for Molecular Docking by Fuzzy Rule-Based Machine Learning Approaches' grant from the University of Macau (grant no. MYRG2016-00217-FST), and (3) the 'A Scalable Data Stream Mining Methodology: Stream-based Holistic Analytics and Reasoning in Parallel' grant from the FDCT, the Macau Government (grant no. FDCT/126/2014/A3).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rawat, P.; Singh, K.D.; Chaouchi, H.; Bonnin, J.M. Wireless sensor networks: a survey on recent developments and potential synergies. *J. Supercomput.* **2014**, *68*, 1–48. [CrossRef]
2. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* **2008**, *52*, 2292–2330. [CrossRef]
3. Majid, M.H.A.; Arshad, M.R. A Fuzzy Self-Adaptive PID Tracking Control of Autonomous Surface Vehicle. In Proceedings of the IEEE International Conference on Control System, Computing and Engineering, George Town, Malaysia, 27–29 November 2015.
4. Fantoni, I.; Lozano, R.; Mazenc, F.; Pettersen, K.Y. Stabilization of a nonlinear underactuated hovercraft. In Proceedings of the 38th Conference on Decision and Control, Phoenix, AZ, USA, 7–10 December 1999.
5. Do, K.D.; Jiang, Z.P.; Pan, J. Underactuated Ship Global Tracking Under Relaxed Conditions. *IEEE Trans. Autom. Control* **1999**, *47*, 1529–1536. [CrossRef]
6. Do, K.D.; Jiang, Z.P.; Pan, J. Universal controllers for stabilization and tracking of underactuated ships. *Syst. Control Lett.* **2002**, *47*, 299–317. [CrossRef]
7. Aguiar, A.P.; Cremean, L.; Hespanha, J.P. Position tracking for a nonlinear underactuated hovercraft: Controller design and experimental results. In Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, HI, USA, 9–12 December 2003; Volume 4, pp. 3858–3863.
8. Bertaska, I.R.; Ellenrieder, K.D. Experimental Evaluation of Supervisory Switching Control for Unmanned Surface Vehicle. *IEEE J. Ocean. Eng.* **2017**. [CrossRef]
9. Xie, W. Robust Motion Control of Underactuated Autonomous Surface Craft. Master's Thesis, University of Macau, Macau, China, 2016.
10. Peng, Z.H.; Wang, J.; Wang, D. Distributed Maneuvering of Autonomous Surface Vehicles Based on Neurodynamic Optimization and Fuzzy Approximation. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1083–1090. [CrossRef]
11. Chen, X.; Tan, W.W. Tracking control of surface vessel via fault tolerant adaptive backstepping interval type-2 fuzzy control. *Ocean Eng.* **2013**, *26*, 97–109. [CrossRef]
12. Wang, N.; Meng, J.E. Self-Constructing Adaptive Robust Fuzzy Neural Tracking Control of Surface Vehicles With Uncertainties and Unknown Disturbances. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 991–1002. [CrossRef]
13. Almeida, J.; Silvestre, C.; Pascoal, A. Cooperative control of multiple surface vessels in the presence of ocean currents and parametric model uncertainty. *Int. J. Robust Nonlinear Control* **2010**, *20*, 1549–1565. [CrossRef]
14. Ghabcheloo, R.; Aguiar, A.P.; Pascoal, A.; Silvestre, C.; Kaminer, I.; Hespanha, J.P. Coordinated path-following control of multiple underactuated autonomous vehicles in the presence of communication failures. In Proceedings of the 45th IEEE Conference on Decision Control, San Diego, CA, USA, 13–15 December 2006.
15. Arrichiello, F.; Chiaverini, S.; Fossen, T.I. Formation control of marine surface vessels using the null-space-based behavioral control. In *Group Coordination and Cooperative Control. Lecture Notes in Control and Information Science*; Springer: Berlin, Germany, 2006; Volume 336.
16. Balch, T.; Arkin, R.C. Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **1999**, *14*, 926–939. [CrossRef]
17. Shojaei, K. Leader–follower formation control of underactuated autonomous marine surface vehicles with limited torque. *Ocean Eng.* **2015**, *105*, 196–205. [CrossRef]
18. Peng, Z.H.; Wang, D.; Chen, Z.Y.; Hu, X.J.; Lan, W.Y. Adaptive Dynamic Surface Control for Formations of Autonomous Surface Vehicles with Uncertain Dynamics. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 513–520. [CrossRef]
19. Luca, C.; Fabio, M.; Domenico, P.; Mario, T. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica* **2008**, *44*, 1343–1349.
20. Wang, Q.; Chen, Z.; Yi, Y. Adaptive coordinated tracking control for multi-robot system with directed communication topology. *Int. J. Adv. Rob. Syst.* **2017**, *14*. [CrossRef]
21. Ren, W.; Beard, R. *Distributed Consensus in Multi-Vehicle Cooperative Control-Theory and Applications*; Springer: Berlin, Germany, 2007.
22. Fossen, T.I. Modeling of Marine Vehicles. In *Guidance and Control of Ocean Vehicles*; John Wiley & Sons Ltd.: Chichester, UK, 1994; pp. 6–54, ISBN 0-471-94113-1.

23. Wen, G.X.; Chen, P.C.L.; Liu, Y.J.; Liu, Z. Neural-network-based adaptive leader-following consensus control of second-order nonlinear multi-agent systems. *IET Control Theory Appl.* **2015**, *9*, 1927–1934. [CrossRef]
24. Wen, G.X.; Chen, P.C.L.; Liu, Y.J.; Liu, Z. Neural Network-Based Adaptive Leader-Following Consensus Control for a Class of Nonlinear Multiagent State-Delay Systems. *IEEE Trans. Cybern.* **2017**, *47*, 2151–2160. [CrossRef] [PubMed]
25. Do, K.D. Robust adaptive path following of underactuated ships. *Automatic* **2004**, *40*, 929–944. [CrossRef]
26. Hespanha, J.P. Internal or Lyapunov Stability. In *Linear Systems Theory*; Princeton University Press: Princeton, NJ, USA, 2009; pp. 63–78, ISBN 978-0-691-14021-6.