2020

COMPOSITE COOPERATIVE ADAPTIVE LEARNING AND CONTROL FOR MULTI-ROBOT COORDINATION

Xiaonan Dong
University of Rhode Island, dong_xn@outlook.com

COMPOSITE COOPERATIVE ADAPTIVE LEARNING AND CONTROL FOR MULTI-ROBOT COORDINATION

BY

XIAONAN DONG

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

MECHANICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2020

DOCTOR OF PHILOSOPHY DISSERTATION

OF

XIAONAN DONG

APPROVED:

Dissertation Committee:

Major Professor	Chengzhi Yuan
	David Chelidze
	Paolo Stegagno
	Nasser H. Zawia DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2020

ABSTRACT

A cooperative adaptive learning-based control (CALC) method and a corresponding experience-based controller for a group of identical unicycle-type ground vehicles are proposed in this research, through both state feedback and output feedback. Specifically, consider the generalized dynamic model of the unicycle-type vehicle with unstructured system uncertainties, the proposed CALC method is able to drive all vehicle agents in the multi-agent system (MAS) to their respective desired reference trajectories and accurately approximate the unmodeled vehicle dynamics with radial basis function (RBF) neural network (NN) at the same time. Furthermore, it is shown that the approximation of the unknown dynamics, presented by the NN weights, will reach consensus by converging to the optimal value of approximation along the union of reference trajectories, for all vehicle agents in the MAS.

In addition, a high-gain observer is also developed to estimate the generalized velocities of the vehicles, in case that only the vehicle's generalized coordinates are measured and accessible for the proposed control methods. It is shown that the proposed state feedback controllers can be modified using output feedback with the estimated velocities, and the objectives of trajectory tracking and accurate learning can still be achieved.

An important novelty of the proposed adaptive learning algorithm is that it grants every vehicle in the group the ability of locally accurately identifying the vehicle dynamics not only along the trajectory experienced by itself, but also along the union trajectories experienced by all other vehicles as well, for both state feedback and output feedback controllers. Another novelty of this research is that the control methods proposed in this research can be applied to a more generalized vehicle model with fewer constraints and assumptions, compared to other research

results for controlling the unicycle-type vehicles using adaptive learning.

Theoretical analysis, as well as simulations, are provided to show the tracking convergence, learning consensus, and accurate approximation of the proposed control methods.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Chengzhi Yuan, for all the help he provided during my Ph.D. program. During the three years program, I had numbers of new experiences. Under his guide, I published my first peer-reviewed paper and presented our research in the academic conference for the first time. I learned the whole process of doing research, from analyzing exsisting researches to forming our results into papers. Not to mention that we went through all the way from setting up equipment and experiments in the temporary lab in Pastore hall and URI Engineering at Schneider Electric, to moving to the new home for the ICRobots lab in the new Engineering building. All the difficulties we overcame and the progress we made during the time will become useful experiences and joyful memories for me in the future.

I also want to thank my committee member Dr. Paolo Stegagno for helping me on coding and the courses I took from him. Countless of problems on running the simulation on Gazebo have been overcome with his help, from installing Ubuntu and setting up ROS, to every single details of writing C++ plugins and solving problems of the Gazebo. I still remember the shock and gratefulness when I followed his suggestion of moving the simulation to a more powerful computer and solved the explainable tracking deviation.

I would like to thank my committee member Dr. David Chelidze for the helping me on the discussion of vehicle dynamic model as a professor, and the advice of the Ph.D. program as a graduate director.

I would like to give my gratitude to Dr. Richard Vaccaro for being the defense chair. I would like to thank him again for all the help he provided during the master's program and the thesis.

At last, I would like to thank my friends, Xiaotian Chen and Jingting Zhang,

for the academic discussions during work time, and all the laughter we have in off days.

TABLE OF CONTENTS

CHAPTER

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Backgrounds

The two wheel-driven, unicycle-type ground vehicle is one of the most widely used mobile robot systems due to its simplicity and practicality. With two actuated wheels installed on each side of the vehicle's body and caster wheel(s) preventing the vehicle from falling down, this unmanned ground vehicle (UGV) has the same kinematic model as a unicycle travelling on a flat plane. Numerous results have been published based on this type of unmanned ground vehicle (UGV), focusing on various fields of research such as trajectory tracking [1, 2, 3, 4, 5], formation control [6, 7, 8, 9, 10], path planning [11, 12, 13, 14, 15], localization [16, 17, 18, 19, 20], etc.

Independently driven by two actuated wheels, a common approach of controlling this unicycle-type vehicle is to design a controller at the kinematic level and generate the desired linear and angular velocities of the vehicle, then use a lower level controller to drive the actuated wheels to the speed calculated from the desired linear and angular velocities with the differential drive method [21]. With the lower level controller integrated into hardware (e.g. Arduino motor shield [22]) with the desired velocity input (e.g. PWM signals), researchers and developers can thereby focus on developing control methods on the kinematic level [8, 14, 16, 23, 24] and developing associated velocity commands for simulations and applications.

1.2 Challenges and Objectives

Despite the simplicity of the two level control setup mentioned above, however, the lower level controller is usually required to be much faster than the differential drive controller at the kinematic level, i.e., the settling time of the lower level controller needs to be significantly shorter than the kinematic controller. In case of tracking a fast time-varying reference trajectory, the lower level controller must be fast and powerful enough to overcome the force generated by inertia and friction almost instantly, which will significantly increase the cost of the hardware and actuators. Therefore, a controller considering the second-order model of the vehicle, including both kinematics and dynamics, should be developed for quick response and accurate tracking.

Following standard back-stepping control method [25] and the virtual velocity tracking controller at the kinematic level [1], a force/torque controller can be easily developed to control the second-order model of the vehicle [26]. However, this full state feedback controller relies on the measurement of all state variables and the accurate modelling of the vehicle, which may raise some challenges in applications.

The first challenge is to have access to all state variables of the system. For the unicycle-type vehicle used in this research, the vehicle position and velocity are typically both required for the force/torque controller. With exteroceptive sensors such as cameras and GPS signals, obtaining the vehicle's position is usually not an issue, however, direct measurement of the vehicle's velocity is more challenging. State observer has been proposed to estimate the full state of the system using the measurable signals [27, 28], however, traditional observers require the knowledge of the system model for accurate state estimations. High-gain observer has been proposed to estimate the unmeasured state variables in case that the system model is not fully known to the observer, and the estimated states can be used for control purposes [29, 30, 31, 32]. In this research, we follow the standard high-gain observer design method [30] to obtain the estimation of vehicle velocities using the measured vehicle position.

The second challenge is the accurate modeling of the vehicle. For traditional controller design, the state-space model is required by the controller and built based on the knowledge of physical model and measurement of the system parameters. For the ground vehicle used in this research, the measurement of kinematic parameters, such as wheel separation and diameter, are usually quite straightforward. However, the dynamic parameters (e.g. moment of inertia, position of the mass center, friction applied on the wheels) can be difficult to measure or model. Different methods have been developed for controlling this vehicle without fully knowing the system model, such as robust control [33, 34, 35, 36, 10] and adaptive control [37, 38, 39, 9, 40]. Particularly, for systems whose model is known yet inaccurate, robust control methods ensure the system stability if the modeling error is bounded [41]. For systems whose model is partially unknown, adaptive control methods are able to adapt/update the parameters of the controller (direct adaptive control) or the system model (indirect adaptive control) to stabilize the system if the system dynamics is time invariant or slowly time-varying [42].

A drawback for the traditional adaptive control methods is that the algorithm is only able to approximate parameters of the structured system uncertainties. For systems with unstructured uncertainties, the deterministic learning (DL) theory was recently proposed to approximate the unmodeled system dynamics with radial basis function (RBF) neural network (NN), use the approximation to design a controller tracking the reference trajectory, and update the NN weights at the same time [43]. Moreover, under the condition of partial persistency of excitation (PE), the system uncertainties can be locally accurately represented by the converged NN weights [44], and the learned knowledge can be directly applied to an experience-based controller without repeating the learning process for future control tasks with similar reference trajectories. Based on this DL method, the problem of composite trajectory tracking control and accurate adaptive learning (i.e., convergence of associated NN weights to their true values) was solved for single unicycle-type vehicles, and the learned knowledge (experience) could be reused for latter controls facing same or similar tasks [31].

One major feature of the DL methods is that the RBFNN approximation is locally accurate along the PE trajectory [44, 31], i.e., the converged NN weight will accurately present the unmodeled system dynamics only along the trajectory experienced by itself. As a result, for vehicles required to cope with multiple tracking tasks with different reference trajectories, traditional DL methods requires the vehicle to run its own learning process and store the NN weight separately for every task, even when the parameters of the vehicles are identical and remain unchanged.

To solve this problem, we take the idea from distributed control of nonholonomic systems in MAS literature [9, 23, 24, 45, 46, 47, 48, 49, 50, 51] to extend the DL theory from controlling single dynamical systems to MASs by sharing the learning information/knowledge among multiple agents. To be more specific, we will propose a cooperative deterministic learning (CDL) method by deploying a group of identical vehicles running different tasks and communicating inside the MAS in real-time. By sharing the learning information of NNs through a MAS consensus control approach [52, 53, 33], our method enables all vehicle agents to learn the unmodeled dynamics (presented by a common RBFNN weight) of the vehicle agent not only along its own trajectory, but also along the union of trajectories experienced by all vehicles in the MAS. Moreover, this cooperative DL method enables all vehicles to achieve tracking and learning simultaneously within one learning process, and the associated weights of all neural networks will reach consensus by converging to the common optimal RBFNN approximation of the vehicle dynamics. In addition, the learned knowledge presented by the converged

NN weight can be re-utilized for controlling the vehicle following the trajectories not only experienced by itself, but also by all other vehicle agents in the MAS as well.

1.3 Dissertation Outline

The rest of this dissertation is organized as follows.

In Chapter 2, notations used in this dissertation and some preliminaries are introduced, including graph theory, RBFNNs based DL method, and the physical model of the unicycle-type vehicles.

In Chapter 3, a cooperative adaptive learning-based controller (CALC) is proposed to drive a group of unicycle-type ground vehicles tracking desired reference trajectories and approximating system uncertainties with RBFNN at the same time. With the learned knowledge presented by the NN weights, an experience-based controller is also proposed for vehicles in the MAS to track the learned trajectories. Analytical analysis and MATLAB simulation results are provided to show the tracking convergence and accurate approximation of the proposed controllers.

In Chapter 4, we further propose the output-feedback controllers based on the results from Chapter 3, with the linear and angular velocities of the vehicles estimated by the observer. A high-gain observer is introduced to estimate the generalized velocities of all vehicle agents in the MAS using the measurement of generalized coordinates. The cooperative adaptive learning-based controller and experience-based controller proposed in Chapter 3 are modified, such that the trajectory tracking and accurate learning can still be achieved with the observer estimation. Analytical analysis and MATLAB simulation results are also provided to show the effectiveness of the proposed controllers.

In Chapter 5, a simulation is run on Gazebo with the proposed controllers.

The model is built in Gazebo simulator for a group of unicycle-type ground vehicles, the proposed controllers are transferred into Python codes, and the simulation is run through the Robot Operating System (ROS).

In Chapter 6, we conclude this work and highlight the contributions of our research, as well as discuss the potential future work.

List of References

[1] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on.* IEEE, 1990, pp. 384–389.

[2] Z.-P. Jiang, E. Lefeber, and H. Nijmeijer, "Saturated stabilization and tracking of a nonholonomic mobile robot," *Systems & Control Letters*, vol. 42, no. 5, pp. 327–332, 2001.

[3] G. M. Atınç, D. M. Stipanović, P. G. Voulgaris, and M. Karkoub, "Collision-free trajectory tracking while preserving connectivity in unicycle multi-agent systems," in *2013 American control conference.* IEEE, 2013, pp. 5392–5397.

[4] Y. Zhang, G. Liu, and B. Luo, "Finite-time cascaded tracking control approach for mobile robots," *Information Sciences*, vol. 284, pp. 31–43, 2014.

[5] X. Yu, L. Liu, and G. Feng, "Trajectory tracking for nonholonomic vehicles with velocity constraints," *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 918–923, 2015.

[6] K. Do and J. Pan, "Nonlinear formation control of unicycle-type mobile robots," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 191–204, 2007.

[7] K. D. Do, "Formation tracking control of unicycle-type mobile robots with limited sensing ranges," *IEEE transactions on control systems technology*, vol. 16, no. 3, pp. 527–538, 2008.

[8] J. Ghommam, H. Mehrjerdi, M. Saad, and F. Mnif, "Formation path following control of unicycle-type mobile robots," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 727–736, 2010.

[9] X. Cai and M. de Queiroz, "Adaptive rigidity-based formation control for multirobotic vehicles with dynamics," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 389–396, 2015.

[10] X. Chu, Z. Peng, G. Wen, and A. Rahmani, "Robust fixed-time consensus tracking with application to formation control of unicycles," *IET Control Theory & Applications*, vol. 12, no. 1, pp. 53–59, 2017.

[11] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[12] K. Pathak and S. K. Agrawal, "An integrated path-planning and control approach for nonholonomic unicycles using switched local potentials," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1201–1208, 2005.

[13] M. Haddad, W. Khalil, and H. Lehtihet, "Trajectory planning of unicycle mobile robots with a trapezoidal-velocity constraint," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 954–962, 2010.

[14] D. Panagou, "A distributed feedback motion planning protocol for multiple unicycle agents of different classes," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1178–1193, 2016.

[15] K. Li, C. Yuan, J. Wang, and X. Dong, "Four-direction search scheme of path planning for mobile agents," *Robotica*, vol. 38, no. 3, pp. 531–540, 2020.

[16] G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, N. Michael, G. Pappas, and K. Daniilidis, "Vision-based localization for leader–follower formation control," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1431–1438, 2009.

[17] T. H. Dinh, M. D. Phung, T. H. Tran, and Q. V. Tran, "Localization of a unicycle-like mobile robot using lrf and omni-directional camera," in *2012 IEEE International Conference on Control System, Computing and Engineering*. IEEE, 2012, pp. 477–482.

[18] H. Sert, W. Perruquetti, A. Kokosy, X. Jin, and J. Palos, "Localizability of unicycle mobiles robots: An algebraic point of view," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 223–228.

[19] V. Vasilopoulos, O. Arslan, A. De, and D. E. Koditschek, "Sensor-based legged robot homing using range-only target localization," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 2630–2637.

[20] B. Araki, I. Gilitschenski, T. Ogata, A. Wallar, W. Schwarting, Z. Choudhury, S. Karaman, and D. Rus, "Range-based cooperative localization with nonlinear observability analysis," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1864–1870.

[21] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control.* Springer Science & Business Media, 2010.

[22] *Arduino motor shield.* [Online]. Available: https://store.arduino.cc/usa/arduino-motor-shield-rev3

[23] Z. Peng, G. Wen, A. Rahmani, and Y. Yu, "Distributed consensus-based formation control for multiple nonholonomic mobile robots with a specified reference trajectory," *International Journal of Systems Science*, vol. 46, no. 8, pp. 1447–1457, 2015.

[24] Z. Liu, L. Wang, J. Wang, D. Dong, and X. Hu, "Distributed sampled-data control of nonholonomic multi-robot systems with proximity networks," *Automatica*, vol. 77, pp. 170–179, 2017.

[25] H. J. Marquez, *Nonlinear Control Systems: Analysis and Design.* Hoboken: Wiley-Interscience, 2003, vol. 1.

[26] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 4. IEEE, 1995, pp. 3805–3810.

[27] D. G. Luenberger, "Observing the state of a linear system," *IEEE transactions on military electronics*, vol. 8, no. 2, pp. 74–80, 1964.

[28] D. Luenberger, "An introduction to observers," *IEEE Transactions on automatic control*, vol. 16, no. 6, pp. 596–602, 1971.

[29] K. W. Lee and H. K. Khalil, "Adaptive output feedback control of robot manipulators using high-gain observer," *International Journal of Control*, vol. 67, no. 6, pp. 869–886, 1997.

[30] H. K. Khalil, "High-gain observers in nonlinear feedback control," in *2008 International Conference on Control, Automation and Systems.* IEEE, 2008, pp. xlvii–lvii.

[31] W. Zeng, Q. Wang, F. Liu, and Y. Wang, "Learning from adaptive neural network output feedback control of a unicycle-type mobile robot," *ISA transactions*, vol. 61, pp. 337–347, 2016.

[32] A. Boker and C. Yuan, "High-gain observer-based distributed tracking control of heterogeneous nonlinear multi-agent systems," in *2018 37th Chinese Control Conference (CCC).* IEEE, 2018, pp. 6639–6644.

[33] S. Khoo, L. Xie, and Z. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE/ASME transactions on mechatronics*, vol. 14, no. 2, pp. 219–228, 2009.

[34] S. Shi, X. Yu, and S. Khoo, "Robust finite-time tracking control of nonholonomic mobile robots without velocity measurements," *International Journal of Control*, vol. 89, no. 2, pp. 411–423, 2016.

[35] S. Roy, S. Nandy, R. Ray, and S. N. Shome, "Robust path tracking control of nonholonomic wheeled mobile robot: Experimental validation," *International Journal of Control, Automation and Systems*, vol. 13, no. 4, pp. 897–905, 2015.

[36] S. Roy, S. Nandy, I. N. Kar, R. Ray, and S. N. Shome, "Robust control of nonholonomic wheeled mobile robot with past information: Theory and experiment," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 231, no. 3, pp. 178–188, 2017.

[37] W. Dong and K.-D. Kuhnert, "Robust adaptive control of nonholonomic mobile robot with parameter and nonparameter uncertainties," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 261–266, 2005.

[38] K. Shojaei, A. M. Shahri, and A. Tarakameh, "Adaptive feedback linearizing control of nonholonomic wheeled mobile robots in presence of parametric and nonparametric uncertainties," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 194–204, 2011.

[39] F. G. Rossomando and C. M. Soria, "Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural pid," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1179–1191, 2015.

[40] Z. Miao and Y. Wang, "Adaptive control for simultaneous stabilization and tracking of unicycle mobile robots," *Asian Journal of Control*, vol. 17, no. 6, pp. 2277–2288, 2015.

[41] L. Eugene, W. Kevin, and D. Howe, *Robust and adaptive control with aerospace applications.* Springer London, 2013.

[42] P. A. Ioannou and J. Sun, *Robust adaptive control.* Courier Corporation, 2012.

[43] C. Wang and D. J. Hill, "Learning from neural control," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 130–146, 2006.

[44] C. Wang and D. J. Hill, *Deterministic learning theory for identification, recognition, and control.* CRC Press, 2009, vol. 32.

[45] X. Chu, Z. Peng, G. Wen, and A. Rahmani, "Distributed formation tracking of multi-robot systems with nonholonomic constraint via event-triggered approach," *Neurocomputing*, vol. 275, pp. 121–131, 2018.

[46] X. Chu, Z. Peng, G. Wen, and A. Rahmani, “Distributed fixed-time formation tracking of multi-robot systems with nonholonomic constraints,” *Neurocomputing*, vol. 313, pp. 167–174, 2018.

[47] C. Yuan, “Distributed adaptive switching consensus control of heterogeneous multi-agent systems with switched leader dynamics,” *Nonlinear Analysis: Hybrid Systems*, vol. 26, pp. 274–283, 2017.

[48] C. Yuan, S. Licht, and H. He, “Formation learning control of multiple autonomous underwater vehicles with heterogeneous nonlinear uncertain dynamics,” *IEEE transactions on cybernetics*, no. 99, pp. 1–15, 2017.

[49] C. Yuan, H. He, X. Dong, and S. Ilbeigi, “Robust distributed adaptive containment control of heterogeneous linear uncertain multi-agent systems,” in *2018 Annual American Control Conference (ACC).* IEEE, 2018, pp. 3660–3665.

[50] C. Yuan, H. He, and C. Wang, “Cooperative deterministic learning-based formation control for a group of nonlinear uncertain mechanical systems,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 319–333, 2019.

[51] P. Stegagno and C. Yuan, “Distributed cooperative adaptive state estimation and system identification for multi-agent systems,” *IET Control Theory & Applications*, vol. 13, no. 1, pp. 815–822, 2019.

[52] N. A. Lynch, *Distributed algorithms.* Elsevier, 1996.

[53] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

CHAPTER 2

Preliminaries

Notations: $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{Z}_+$ denote, respectively, the set of real numbers, the set of positive real numbers and the set of positive integers; $\mathbb{R}^{m\times n}$ denotes the set of $m \times n$ real matrices; $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors; $O_{m\times n}$ denotes the zero matrix with dimension of $m \times n$; $\mathcal{N}(A)$ denotes the nullspace of matrix A; I_n denotes the $n \times n$ identity matrix; Subscript $(\cdot)_k$ denotes the k^{th} column vector of a matrix; $|\cdot|$ is the absolute value of a real number, and $||\cdot||$ is the 2-norm of a vector, i.e. $||x|| = (x^T x)^{\frac{1}{2}}$; $\dot{z}$ denotes the total derivative of z with respect to the time; $\partial/\partial\mathbf{z}$ denotes the Jacobian matrix as $\frac{\partial}{\partial\mathbf{z}} = \begin{bmatrix}\frac{\partial}{\partial z_1} & \cdots & \frac{\partial}{\partial z_n}\end{bmatrix}$, with $\mathbf{z} = \begin{bmatrix}z_1 & z_2 & \cdots & z_n\end{bmatrix}^T$.

2.1 Graph Theory

In a graph defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, the elements in set $\mathcal{V} = \{1, 2, \ldots, n\}$ are called vertices, the elements of $\mathcal{E}$ are pairs (i, j) with $i, j \in \mathcal{V}, i \neq j$ called edges, and the matrix $\mathcal{A}$ is called the adjacency matrix. If $(i, j) \in \mathcal{E}$, then agent i is able to receive information from agent j, and agent i and j are called adjacent. The adjacency matrix is thus defined as $\mathcal{A} = [a_{ij}]_{n\times n}$, in which

$$\begin{cases} a_{ij} > 0 & (i, j) \in \mathcal{E}, \\ a_{ij} = 0 & (i, j) \notin \mathcal{E}. \end{cases} \tag{1}$$

For any two nodes $v_i, v_j \in \mathcal{V}$, if there exists a path between them, then the graph $\mathcal{G}$ is called connected. Furthermore, the graph $\mathcal{G}$ is called fixed if $\mathcal{E}$ and $\mathcal{A}$ do not change over time, and called undirected if $\forall(i, j) \in \mathcal{E}$, pair (j, i) is also in $\mathcal{E}$.

Lemma 1. *[1] For the Laplacian matrix $L = [l_{ij}]_{n\times n}$ associated with the undirected* *graph $\mathcal{G}$, in which*

$$l_{ij} = \begin{cases} \sum_{j=1,j\neq i}^{n} a_{ij} & i = j, \\ -a_{ij} & i \neq j. \end{cases} \tag{2}$$

If the graph is connected, then L is a positive semi-definite symmetric matrix, with one zero eigenvalue and all other eigenvalues to be positive and hence, $\operatorname{rank}(L) \leq n-1$.

2.2 RBFNN Approximation and Deterministic Learning

The RBF neural network output function can be described by $f_{nn}(Z) = \sum_{i=1}^{N_n} w_i s_i(Z) = W^T S(Z)$ [2], where $Z \in \Omega_Z \subset \mathbb{R}^q$ is the input vector, $W = \begin{bmatrix} w_1 & \cdots & w_{N_n} \end{bmatrix}^T \in \mathbb{R}^{N_n}$ is a vector of NN weights, N_n is the NN node number, and $S(Z) = \begin{bmatrix} s_1(||Z-\mu_1||) & \cdots & s_{N_n}(||Z-\mu_{N_n}||) \end{bmatrix}^T$, with $s_i(\cdot)$ being a radial basis function, and μ_i $(i = 1, 2, \cdots, N_n)$ being the coordinate vectors of distinct points in state space. The Gaussian function $s_i(||Z-\mu_i||) = \exp[-\frac{(Z-\mu_i)^T(Z-\mu_i)}{\eta_i^2}]$ is one of the most commonly used radial basis functions, where $\mu_i = \begin{bmatrix} \mu_{i1} & \cdots & \mu_{iq} \end{bmatrix}^T \in \mathbb{R}^q$ is the center of the receptive field and η_i is the width of the receptive field. The Gaussian function belongs to the class of localized RBFs in the sense that $s_i(||Z-\mu_i||) \to 0$ as $||Z-\mu_i|| \to \infty$. It is easily seen that $S(Z)$ is bounded and there exists a real constant $S_M \in \mathbb{R}_+$ such that $||S(Z)|| \leq S_M$ [3].

It has been shown in [2, 4] that for any continuous function $f(Z) : \Omega_Z \to \mathbb{R}$ where $\Omega_Z \subset \mathbb{R}^q$ is a compact set, and for the NN approximator, where the node number N_n is sufficiently large, there exists an ideal constant weight vector W^*, such that for any $\epsilon^* > 0$, $f(Z) = W^{*T} S(Z) + \epsilon$, $\forall Z \in \Omega_Z$, where $|\epsilon| < \epsilon^*$ is the ideal approximation error. The ideal weight vector W^* is an 'artificial' quantity required for analysis, and is defined as the value of W that minimizes $|\epsilon|$ for all $Z \in \Omega_Z \subset \mathbb{R}^q$, i.e. $W^* := \arg\min_{W \in \mathbb{R}^{N_n}} \{\sup_{Z \in \Omega_Z} |f(Z) - W^T S(Z)|\}$. Moreover, based on the localization property of RBF NNs [3], for any bounded trajectory $Z(t)$ within the compact set Ω_Z, $f(Z)$ can be approximated by a limited number of neurons located in a local region along the trajectory: $f(Z) = W_\zeta^{*T} S_\zeta(Z) + \epsilon_\zeta$, where ϵ_ζ is the approximation error, with $\epsilon_\zeta = O(\epsilon) = O(\epsilon^*)$, $S_\zeta(Z) = \begin{bmatrix} s_{j1}(Z) & \cdots & s_{j\zeta}(Z) \end{bmatrix}^T \in \mathbb{R}^{N_\zeta}$, $W_\zeta^* = \begin{bmatrix} w_{j1}^* & \cdots & w_{j\zeta}^* \end{bmatrix}^T \in \mathbb{R}^{N_\zeta}$, $N_\zeta < N_n$, and the integers $j_i = j_1, \cdots, j_\zeta$ are defined by $|s_{j_i}(Z_p)| > \theta$ ($\theta > 0$ is a small positive constant) for some $Z_p \in Z(k)$.

It is shown in [3] that for a localized RBF network $W^T S(Z)$ whose centers are placed on a regular lattice, almost any recurrent trajectory [1] $Z(k)$ can lead to the satisfaction of the PE condition of the regressor subvector $S_\zeta(Z)$. This result can be formally summarized in the following lemma.

Lemma 2 ([3, 5])**.** *Consider any recurrent trajectory $Z(k)$: $\mathbb{Z}_+ \to \mathbb{R}^q$. $Z(k)$ remains in a bounded compact set $\Omega_Z \subset \mathbb{R}^q$, then for RBF network $W^T S(Z)$ with centers placed on a regular lattice (large enough to cover compact set Ω_Z), the regressor subvector $S_\zeta(Z)$ consisting of RBFs with centers located in a small neighborhood of $Z(k)$ is persistently exciting.*

2.3 Vehicle Model

The physical model of the unicycle-type vehicle is shown in Figure 1. The body-fixed frame of the vehicle is defined using the following forward-left-up right-handed Cartesian coordinate system:

- The origin (reference point) is the center point between two actuated wheels.
- The X_V (forward) axis is pointing towards the center of the front caster.
- The Y_V (left) axis is pointing towards the center of the left actuated wheel, along the joints of the actuated wheels.

[1] A recurrent trajectory represents a large set of periodic and quasiperiodic trajectories generated from linear/nonlinear dynamics systems. A detailed characterization of recurrent trajectories can be found in [3].

- The Z_V (up) axis is pointing upwards and perpendicular to the $X_V - Y_V$ plane

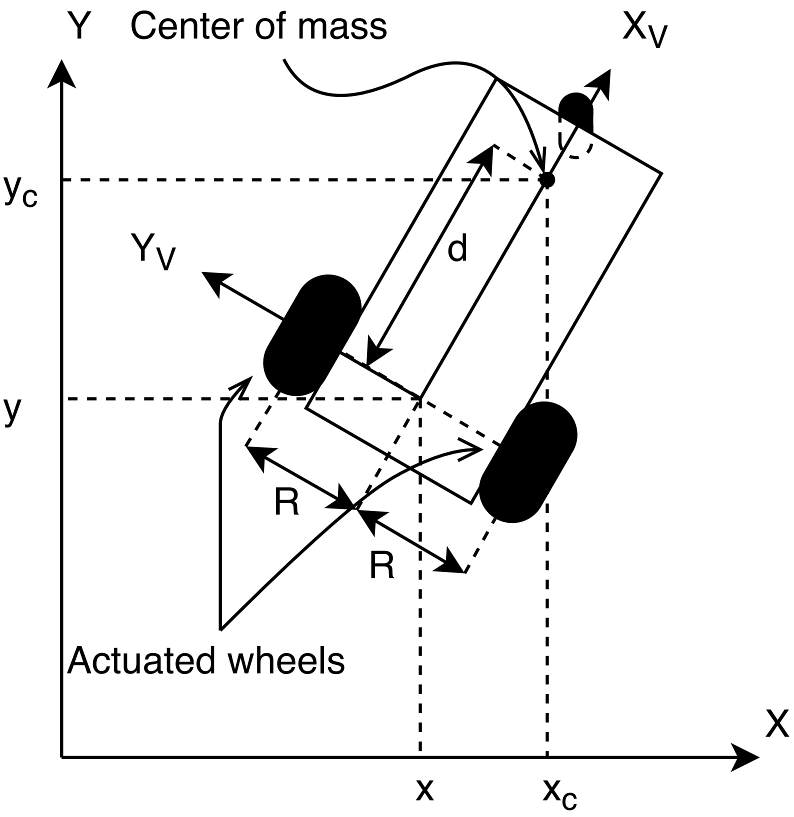


Figure 1: Unicycle-type vehicle

By defining the coordinates of the reference point in the ground frame as (x, y) and the angle between X_V and the X axis of the ground frame as θ, the vehicle's position and orientation can be thereby combined into a generalized coordinates vector $\mathbf{q} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$. Therefore, the unicycle-type ground vehicle is a 3-DOF system.

The mass center of the vehicle is on the X_V axis of the body-fixed frame, whose distance from the reference point equals to d and hence, then the generalized coordinates of the mass center in the ground frame is

$$\mathbf{q}_c = \begin{bmatrix} x_c \\ y_c \\ \theta_c \end{bmatrix} = \begin{bmatrix} x + d\cos\theta \\ y + d\sin\theta \\ \theta \end{bmatrix}. \tag{3}$$

As is shown in Figure 1, this unicycle-type vehicle is a nonholonomic system, with the constraint force preventing the vehicle from sliding along the axis of the actuated wheels (Y_V). The nonholonomic constraint, also known as the kinematic constraint, can be expressed in the Pfaffian form as [6]

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0, \tag{4}$$

or equivalently, in the matrix form

$$A^T(\mathbf{q})\dot{\mathbf{q}} = 0, \tag{5}$$

where $A^T(\mathbf{q}) = \begin{bmatrix} \sin\theta & -\cos\theta & 0 \end{bmatrix}$. With this constraint, the mobility of the vehicle is limited and hence, $\dot{\mathbf{q}}$ must be in the null space of $A^T(\mathbf{q})$. Therefore, the time derivative of the generalized coordinates $\mathbf{q}$ can be constructed as a linear combination of two independent generalized velocities u_1 and u_2 as

$$\dot{\mathbf{q}} = j_1 u_1 + j_2 u_2, \tag{6}$$

where j_1 and j_2 are basis of $\mathcal{N}(A^T(\mathbf{q}))$. A natural choice of the generalized velocities is the linear velocity v along X_V axis, and the angular velocity ω with respect to Z_V axis, then the non-slippery kinematics of the vehicle can be presented as

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \overset{\text{def}}{=} J(\mathbf{q})\mathbf{u}, \tag{7}$$

where $J(\mathbf{q}) = \begin{bmatrix} j_1 & j_2 \end{bmatrix}$, and $\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T = \begin{bmatrix} v & \omega \end{bmatrix}^T$. Notice that j_1, j_2, and A compose a set of orthonormal basis of the linear space $\mathbb{R}^3$

For the dynamics of the vehicle, the state-space model can be described by a 2nd order ordinary differential equation (ODE) of the generalized coordinates $\mathbf{q}$ [7]

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + F(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) = B(\mathbf{q})\tau + A(\mathbf{q})\lambda, \tag{8}$$

in which $M \in \mathbb{R}^{3\times3}$ is a positive definite matrix that denotes the inertia, $C \in \mathbb{R}^{3\times3}$ is the centripetal and Coriolis matrix, $F \in \mathbb{R}^{3\times1}$ is the friction vector, $G \in \mathbb{R}^{3\times1}$ is the gravity vector. $\tau \in \mathbb{R}^{2\times1}$ is a vector of system input, i.e. the torque applied on each actuation wheel, $B = \frac{1}{r}\begin{bmatrix}\cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ R & -R\end{bmatrix} \in \mathbb{R}^{3\times2}$ is the input transformation matrix, projecting the system input τ onto the space spanned by the generalized coordinates $\mathbf{q}$, where R is the distance from the actuation wheel to the origin of the body-fixed frame, and r is the radius of the wheel. λ is a Lagrange multiplier, and $A(\mathbf{q})\lambda \in \mathbb{R}^{3\times1}$ denotes the constraint force.

Matrices M and C in equation (8) can be derived using the Lagrangian equation with the following steps. First we derive the kinetic energy of the vehicle using the linear and angular velocities with respect to the mass center

$$T = \frac{m(\dot{x}_c^2 + \dot{y}_c^2)}{2} + \frac{I_c\dot{\theta}_c^2}{2}, \tag{9}$$

where m is the mass of the vehicle, I_c is the moment of inertia measured at the center of mass. From equation (3), we can derive the time derivative of $\mathbf{q}_c$

$$\begin{aligned}\dot{x}_c &= \dot{x} - d\dot{\theta}\sin\theta, \\ \dot{y}_c &= \dot{y} + d\dot{\theta}\cos\theta, \\ \dot{\theta}_c &= \dot{\theta},\end{aligned} \tag{10}$$

then the kinetic energy (9) can be rewritten into

$$\begin{aligned}T(\mathbf{q},\dot{\mathbf{q}}) &= \frac{m[(\dot{x} - d\dot{\theta}\sin\theta)^2 + (\dot{y} + d\dot{\theta}\cos\theta)^2]}{2} + \frac{I_c\dot{\theta}^2}{2} \\ &= \frac{1}{2}[m\dot{x}^2 + m\dot{y}^2 + (md^2 + I_c)\dot{\theta}^2 - 2md\sin\theta\dot{x}\dot{\theta} + 2md\cos\theta\dot{y}\dot{\theta}] \\ &= \frac{\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}}}{2},\end{aligned} \tag{11}$$

in which

$$M = \begin{bmatrix} m & 0 & -md\sin\theta \\ 0 & m & md\cos\theta \\ -md\sin\theta & md\cos\theta & md^2 + I_c \end{bmatrix}. \tag{12}$$

It will be shown later that the inertia matrix M given above is identical to that in equation (8). The dynamics equation (8) can be derived from the following Lagrangian equation [6]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)^T - \left(\frac{\partial L}{\partial \mathbf{q}}\right)^T = A(\mathbf{q})\lambda + \mathbf{Q}, \tag{13}$$

in which $L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q})$ is the Lagrangian of the vehicle with $U(\mathbf{q})$ presenting the potential energy, λ is the Lagrangian multiplier, and $A(\mathbf{q})^T\lambda$ denotes the constraint force. $\mathbf{Q} = B(\mathbf{q})[\tau - \mathbf{f}(\mathbf{u})]$ denotes the external force, where τ is the force generated by the actuator, and $\mathbf{f}(\mathbf{u})$ is the friction on actuators, joints, and the wheel surface. Then equation (13) can be rewritten into

$$M(\mathbf{q})\ddot{\mathbf{q}} + \dot{M}\dot{\mathbf{q}} - \left(\frac{\partial T}{\partial \mathbf{q}}\right)^T + \left(\frac{\partial U}{\partial \mathbf{q}}\right)^T + B(\mathbf{q})\mathbf{f}(\mathbf{u}) = A(\mathbf{q})\lambda + B(\mathbf{q})\tau. \tag{14}$$

By setting $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \dot{M}\dot{\mathbf{q}} - \left(\frac{\partial T}{\partial \mathbf{q}}\right)^T$, $F(\mathbf{q}, \dot{\mathbf{q}}) = B(\mathbf{q})\mathbf{f}(\dot{\mathbf{q}})$, and $G(\mathbf{q}) = \left(\frac{\partial U}{\partial \mathbf{q}}\right)^T$, equation (14) can be thereby transferred into (8), with the inertia matrix M identical to that in (11). Notice that the form of $C_{n\times n}$ is not unique, however, with a proper definition of the matrix C, we will have $\dot{M} - 2C$ to be skew-symmetric [6]. The $(i, j)^{\text{th}}$ entry of C is defined as $c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k$, where $\dot{q}_k$ is the k^{th} entry of $\dot{\mathbf{q}}$, and $c_{ijk} = \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{jk}}{\partial q_i}\right)$ is defined using the Christoffel symbols of the first kind [6]. Then we have the centripetal and Coriolis matrix calculated as

$$C = \begin{bmatrix} 0 & 0 & -md\dot{\theta}\cos\theta \\ 0 & 0 & -md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

With the vehicle operating on the ground, the potential energy U is a constant value, and we thereby have the gravity vector $G = \left(\frac{\partial U}{\partial \mathbf{q}}\right)^T$ equals to zero. The friction vector F is assumed to be a nonlinear function of the generalized velocity $\mathbf{u}$ and unknown to the controller.

To reduce the degree of freedor (DOF) of the system into two and eliminate the nonholonomic constraint force $A(\mathbf{q})\lambda$ from equation (8), we left multiplying

$J^T(\mathbf{q})$ to the equation, it yields:

$$J^T M J \dot{\mathbf{u}} + J^T(M\dot{J} + CJ)\mathbf{u} + J^T F + J^T G = J^T B\tau + J^T A\lambda. \tag{16}$$

Since columns of J and A are orthonormal basis of $\mathbb{R}^3$, then we have $J^T A = \mathbf{0}_{2\times 1}$, and the dynamic equation of $\mathbf{u}$ can be simplified as

$$\bar{M}(\mathbf{q})\dot{\mathbf{u}} + \bar{C}(\mathbf{u})\mathbf{u} + \bar{F}(\mathbf{u}) + \bar{G}(\mathbf{q}) = \bar{\tau}, \tag{17}$$

where

$$\begin{aligned}
\bar{M} &= J^T M J = \begin{bmatrix} m & 0 \\ 0 & md^2 + I_c \end{bmatrix}, \\
\bar{C} &= J^T(M\dot{J} + CJ) = \begin{bmatrix} 0 & -md\dot{\theta} \\ md\dot{\theta} & 0 \end{bmatrix}, \\
\bar{F} &= J^T F, \\
\bar{G} &= J^T G = \mathbf{0}_{2\times 1}, \\
\bar{\tau} &= \begin{bmatrix} \bar{\tau}_v \\ \bar{\tau}_\omega \end{bmatrix} = J^T B\tau = \begin{bmatrix} 1/r & 1/r \\ R/r & -R/r \end{bmatrix}\tau.
\end{aligned} \tag{18}$$

Notice that $md^2 + I_c = I$ is the moment of inertia with respect to the reference point, which can be derived using the parallel axis theorem [8].

The DOF of the vehicle dynamics is now reduced to two with the constraint force taken out of the equation. Since $J^T B$ is of full rank, then for any transformed torque input $\bar{\tau}$, there exists a unique corresponding actual torque input $\tau \in \mathbb{R}^2$ that applied on each wheel. With the state-space model fully described by the kinematic equation (7) and dynamic equation (17), we can now proceed to discuss the problem statement and controller design in the following chapters.

List of References

[1] R. Agaev and P. Chebotarev, “The matrix of maximum out forests of a digraph and its applications,” *arXiv preprint math/0602059*, 2006.

[2] J. Park and I. W. Sandberg, “Universal approximation using radial-basis-function networks,” *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.

[3] C. Wang and D. J. Hill, *Deterministic learning theory for identification, recognition, and control.* CRC Press, 2009, vol. 32.

[4] M. D. Buhmann, *Radial basis functions: theory and implementations.* Cambridge university press, 2003, vol. 12.

[5] C. Wang and D. J. Hill, "Learning from neural control," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 130–146, 2006.

[6] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control.* Springer Science & Business Media, 2010.

[7] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 4. IEEE, 1995, pp. 3805–3810.

[8] R. C. Hibbeler, *Engineering mechanics. Statics and dynamics. 11th.* Upper Saddle River, NJ: Pearson/Prentice-Hall. xvi, 1997.

CHAPTER 3

The cooperative adaptive learning and control (CALC) framework via state feedback

3.1 Problem Statement

Consider a group of unicycle-type ground vehicles with identical mechanical model, the objective of this research is to design a trajectory tracking controller. For traditional control methods such as pole placement, optimal control, and input-output linearization, the accurate description of the system model is required for controller design. Recall the parameters in the vehicle dynamics equation (17)

$$
\begin{aligned}
&\bar{M} = J^T M J = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}, \quad \bar{C} = J^T(M\dot{J} + CJ) = \begin{bmatrix} 0 & -md\dot{\theta} \\ md\dot{\theta} & 0 \end{bmatrix}, \\
&\bar{F} = J^T F, \quad \bar{G} = J^T G = \mathbf{0}_{2\times 1}, \quad \bar{\tau} = \begin{bmatrix} \bar{\tau}_v \\ \bar{\tau}_\omega \end{bmatrix} = J^T B \tau = \begin{bmatrix} 1/r & 1/r \\ R/r & -R/r \end{bmatrix} \tau.
\end{aligned} \tag{19}
$$

For the input transformation matrix $J^T B$, the measurement of dimensional parameters (wheel offset R and radius r) are quite straightforward. However, measuring the moment of inertia I and the position of mass center d usually takes much more effort. In addition, the linear model of Coulomb friction may not accurately present the actual friction $\bar{F}$ applied on the vehicle. To control the vehicles with unknown dynamic parameters, we introduce a cooperative adaptive learning-based controller and an experience-based controller for the trajectory tracking task, using the RBFNN approximation of the unstructured system uncertainties. In order to apply the cooperative adaptive learning algorithms, we have the following assumptions for our multi-agent system (MAS)

Assumption 1. *The communication graph $\mathcal{G}$ associated to the MAS is undirected and connected.*

Assumption 2. *The reference trajectories for all vehicle agents in the MAS are bounded continuous functions.*

Assumption 3. *The state variables, i.e., generalized coordinates and velocities, associated to all reference trajectories are recurrent.*

Based on the vehicle model in Section 2.3 and the assumptions shown above, the objective of this dissertation can be concluded as follows. Consider an MAS containing n identical unicycle-type vehicles with the state-space model of each vehicle in the MAS described by the kinematics equation (7) and the dynamics equation (17), under the Assumptions 1, 2, and 3, the proposed control algorithms will achieve the following objectives

i) *Trajectory tracking:* Each vehicle in the MAS will track its desired reference trajectory $\mathbf{q}_{r_i}(t)$, i.e., $\lim_{t\to\infty}(\mathbf{q}_{r_i}(t) - \mathbf{q}_i(t)) = 0$, $\forall i \in \{1, \cdots, n\}$.

ii) *Consensus learning:* The homogeneous unmodeled dynamics of all the vehicles can be locally accurately identified, i.e., the NN weights for all vehicle agents in the MAS reach their common optimal value, along the union of the trajectories experienced by all vehicle agents in the MAS.

iii) *Experience-based control:* The learned knowledge from the cooperative learning phase can be re-utilized by each local vehicle to perform stable trajectory tracking without adaptive updating opearations.

3.2 CALC with Tracking Stability and Weight Convergence Analysis

For better analyzing the tracking convergence, we first define the tracking error $\tilde{\mathbf{q}}_i$ of the i^{th} vehicle by projecting the tracking error measured in the ground frame onto the body-fixed frame of the vehicle, as shown in Figure 2:

$$\tilde{\mathbf{q}}_i = \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \\ \tilde{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{r_i} - x_i \\ y_{r_i} - y_i \\ \theta_{r_i} - \theta_i \end{bmatrix}, \tag{20}$$

in which the subscript r denotes the variables of the reference trajectories. Using the constraint (5) and the kinematics (7), we then have the derivative of the

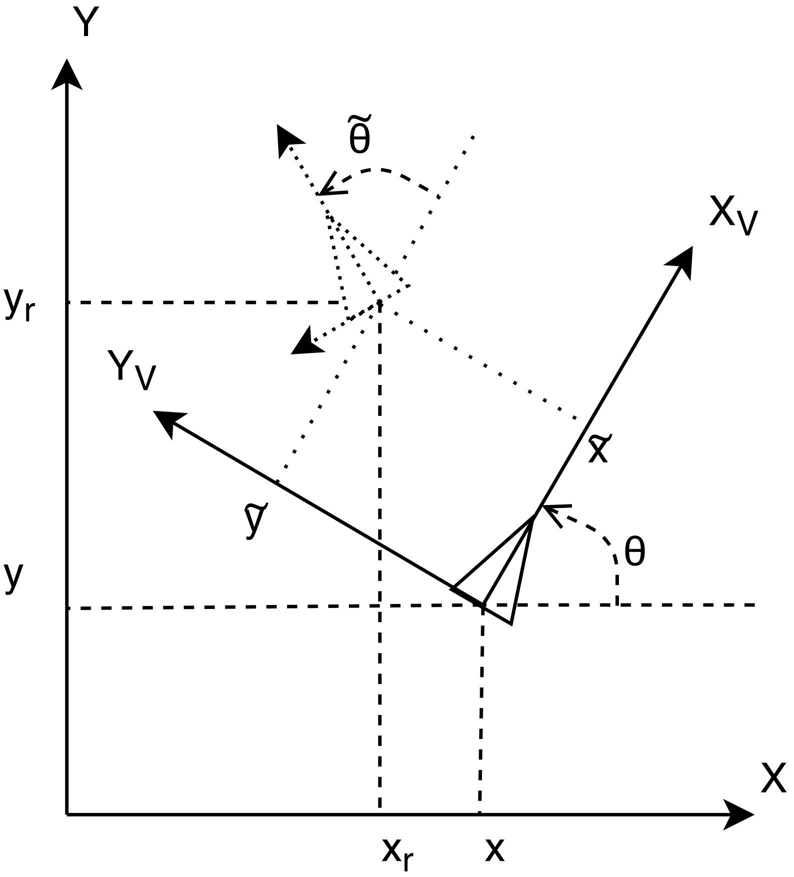


Figure 2: Projecting tracking error onto the body-fixed frame

tracking error as

$$\begin{aligned}
\dot{\tilde{x}}_i &= (\dot{x}_{r_i} - \dot{x}_i)\cos\theta_i - (x_{r_i} - x_i)\sin\theta_i\dot{\theta}_i + (\dot{y}_{r_i} - \dot{y}_i)\sin\theta_i + (y_{r_i} - y_i)\cos\theta_i\dot{\theta}_i \\
&= v_{r_i}\cos\theta_{r_i}\cos\theta_i + v_{r_i}\sin\theta_{r_i}\sin\theta_i + \tilde{y}\omega - v_i \\
&= v_{r_i}\cos\tilde{\theta}_i + \omega_i\tilde{y}_i - v_i \\
\dot{\tilde{y}}_i &= -(\dot{x}_{r_i} - \dot{x}_i)\sin\theta_i - (x_{r_i} - x_i)\cos\theta_i\dot{\theta}_i + (\dot{y}_{r_i} - \dot{y}_i)\cos\theta_i - (y_{r_i} - y_i)\sin\theta_i\dot{\theta}_i \\
&= -v_{r_i}\cos\theta_{r_i}\sin\theta_i + v_{r_i}\sin\theta_{r_i}\cos\theta_i - \tilde{x}\omega \\
&= v_{r_i}\sin\tilde{\theta}_i - \omega_i\tilde{x}_i \\
\dot{\tilde{\theta}}_i &= \omega_{r_i} - \omega_i
\end{aligned} \tag{21}$$

in which v_i and ω_i are the linear and angular velocities of the i^{th} vehicle, respectively.

In order to utilize the backstepping control theory [1], we treat v_i and ω_i in equation (21) as virtual inputs, then following the methodology from [2], we can design a stabilizing virtual controller as

$$\mathbf{u}_{c_i} = \begin{bmatrix} v_{c_i} \\ \omega_{c_i} \end{bmatrix} = \begin{bmatrix} v_{r_i}\cos\tilde{\theta}_i + K_x\tilde{x}_i \\ \omega_{r_i} + v_{r_i}K_y\tilde{y}_i + K_\theta\sin\tilde{\theta}_i \end{bmatrix}, \tag{22}$$

in which K_x, K_y, and K_θ are all positive constants. It can be shown that this virtual velocity controller is able to make the tracking error $\mathbf{q}_i$ converge to zero asymptotically on the kinematic level. Specific proofs can be done by replacing the v_i and ω_i in equation (21) with the virtual controller v_{c_i} and ω_{c_i}, respectively. To this end, we define the following Lyapunov function for the i^{th} vehicle

$$V_{1_i} = \frac{\tilde{x}_i^2}{2} + \frac{\tilde{y}_i^2}{2} + \frac{(1 - \cos\tilde{\theta}_i)}{K_y} \tag{23}$$

and the derivative of V_{1_i} is

$$\begin{aligned} \dot{V}_{1_i} &= \tilde{x}_i\dot{\tilde{x}}_i + \tilde{y}_i\dot{\tilde{y}}_i + \frac{\sin\tilde{\theta}_i}{K_y}\dot{\tilde{\theta}}_i \\ &= \tilde{x}_i(v_{r_i}\cos\tilde{\theta}_i + \omega_i\tilde{y}_i - v_{c_i}) + \tilde{y}_i(v_{r_i}\sin\tilde{\theta}_i - \omega_i\tilde{x}_i) + \frac{\sin\tilde{\theta}_i}{K_y}(\omega_{r_i} - \omega_{c_i}) \\ &= \tilde{x}_i(\omega_i\tilde{y}_i - K_x\tilde{x}_i) + \tilde{y}_i(v_{r_i}\sin\tilde{\theta}_i - \omega_i\tilde{x}_i) + \frac{\sin\tilde{\theta}_i}{K_y}(-v_{r_i}K_y\tilde{y}_i - K_\theta\sin\tilde{\theta}_i) \\ &= -K_x\tilde{x}_i^2 - \frac{K_\theta}{K_y}\sin^2\tilde{\theta}_i \leq 0 \end{aligned} \tag{24}$$

Since $\dot{V}_{1_i}$ is negative semi-definite, we can conclude that the system is stable, with the tracking error $\tilde{\mathbf{q}}_i$ to be bounded for all vehicle agents in the MAS.

Remark 1. *In addition to the stable conclusion above, we could also conclude the asymptotic stability by finding the invariant set of* $\dot{V}_{1_i} = 0$ *[3]. By setting* $\dot{V}_{1_i} = 0$*, we have* $\tilde{x}_i = 0$ *and* $\sin\tilde{\theta}_i = 0$*. Applying this result into equation (21) and (22), we have the invariant set equals to* $\{\tilde{x}_i = 0, \tilde{y}_i = 0, \tilde{\theta}_i = 0\} \cup \{\tilde{x}_i = 0, \tilde{y}_i = 0, \tilde{\theta}_i = -\pi, v_{r_i} \neq 0, \omega_{r_i} = 0\} \cup \{\tilde{x}_i = 0, \tilde{y}_i \neq 0, \sin\tilde{\theta}_i = 0, v_{r_i} = 0, \omega_{r_i} = 0\}$*. With the assumption 3, the generalized velocities of the reference trajectory cannot be constant over time, then we can conclude that the only invariant subset of* $\dot{V}_{1_i} = 0$ *is the origin* $\tilde{\mathbf{q}}_i = \mathbf{0}$*. Therefore, we can conclude that virtual controller (22) will make the tracking system asymptotically stable.*

With the idea of backstepping control, we then derive the transformed torque input $\bar{\tau}_i$ for the i^{th} vehicle with the following steps. By defining the error between the virtual controller $\mathbf{u}_{c_i}$ and the actual velocity $\mathbf{u}_i$ as $\tilde{\mathbf{u}}_i = \begin{bmatrix}\tilde{v}_i & \tilde{\omega}_i\end{bmatrix}^T = \mathbf{u}_{c_i} - \mathbf{u}_i$, we can rewrite equation (21) in terms of $\tilde{v}_i$ and $\tilde{\omega}_i$ as

$$\begin{aligned} \dot{\tilde{x}}_i &= v_{r_i} \cos \tilde{\theta}_i + \omega_i \tilde{y}_i - v_{c_i} + \tilde{v}_i \\ &= -K_x \tilde{x}_i + \omega_i \tilde{y}_i + \tilde{v}_i \\ \dot{\tilde{y}}_i &= -\omega_i \tilde{x}_i + v_{r_i} \sin \tilde{\theta}_i \\ \dot{\tilde{\theta}}_i &= \omega_{r_i} - \omega_{c_i} + \tilde{\omega}_i \\ &= -v_{r_i} K_y \tilde{y}_i - K_\theta \sin \tilde{\theta}_i + \tilde{\omega}_i \end{aligned} \tag{25}$$

Then we define a new Lyapunov function $V_{2_i} = V_{1_i} + \frac{\tilde{\mathbf{u}}_i^T \bar{M} \tilde{\mathbf{u}}_i}{2}$ for the closed-loop system (25), whose derivative can be written as

$$\begin{aligned} \dot{V}_{2_i} &= \tilde{x}_i \dot{\tilde{x}}_i + \tilde{y}_i \dot{\tilde{y}}_i + \frac{\sin \tilde{\theta}_i}{K_y} \dot{\tilde{\theta}}_i + \tilde{\mathbf{u}}_i^T \bar{M} \dot{\tilde{\mathbf{u}}}_i \\ &= \tilde{x}_i(-K_x \tilde{x}_i + \omega_i \tilde{y}_i + \tilde{v}_i) + \tilde{y}_i(-\omega_i \tilde{x}_i + v_{r_i} \sin \tilde{\theta}_i) \\ &\quad + \frac{\sin \tilde{\theta}_i}{K_y}(-v_{r_i} K_y \tilde{y}_i - K_\theta \sin \tilde{\theta}_i + \tilde{\omega}_i) + \tilde{\mathbf{u}}_i^T \bar{M} \dot{\tilde{\mathbf{u}}}_i \\ &= -K_x \tilde{x}_i^2 - \frac{K_\theta}{K_y} \sin^2 \tilde{\theta}_i + \tilde{\mathbf{u}}_i^T \left(\begin{bmatrix} \tilde{x}_i \\ \frac{\sin \tilde{\theta}_i}{K_y} \end{bmatrix} + \bar{M} \dot{\tilde{\mathbf{u}}}_i \right) \end{aligned} \tag{26}$$

To make the system stable, the term $\tilde{\mathbf{u}}_i^T \left(\begin{bmatrix} \tilde{x}_i \\ \frac{\sin \tilde{\theta}_i}{K_y} \end{bmatrix} + \bar{M} \dot{\tilde{\mathbf{u}}}_i \right)$ needs to be negative definite. From the definition of $\tilde{\mathbf{u}}_i$ and equation (17), we have

$$\bar{M} \dot{\tilde{\mathbf{u}}}_i = \bar{M} \dot{\mathbf{u}}_{c_i} - \bar{M} \dot{\mathbf{u}}_i = \bar{M} \dot{\mathbf{u}}_{c_i} + \bar{C} \mathbf{u}_i + \bar{F} - \bar{\tau}_i \tag{27}$$

Motivated from the results of [4], it is easy to show that this term is negative definite if $\bar{\tau}_i$ is designed to be

$$\bar{\tau}_i = \bar{M} \dot{\mathbf{u}}_{c_i} + \bar{C} \mathbf{u}_i + \bar{F} + K_u \tilde{\mathbf{u}}_i + \begin{bmatrix} \tilde{x}_i \\ \frac{\sin \tilde{\theta}_i}{K_y} \end{bmatrix}, \tag{28}$$

where $K_u \in \mathbb{R}^+$, and

$$\dot{\mathbf{u}}_{c_i} = \begin{bmatrix} \dot{v}_{r_i} - v_{r_i} \sin \tilde{\theta}_i \dot{\tilde{\theta}}_i + K_x \dot{\tilde{x}}_i \\ \dot{\omega}_{r_i} + \dot{v}_{r_i} K_y \tilde{y}_i + v_{r_i} K_y \dot{\tilde{y}}_i + K_\theta \cos \tilde{\theta}_i \dot{\tilde{\theta}}_i \end{bmatrix}, \tag{29}$$

with $\dot{\tilde{x}}_i = v_{r_i}\cos\tilde{\theta}_i + \omega_i\tilde{y}_i - v_i$, $\dot{\tilde{y}}_i = v_{r_i}\sin\tilde{\theta}_i - \omega_i\tilde{x}_i$, and $\dot{\tilde{\theta}}_i = \omega_{r_i} - \omega_i$ given by (21).

To control the vehicles following their reference trajectories, system parameter matrices $\bar{M}$, $\bar{C}$, and $\bar{F}$ need to be approximated by the RBFNN, i.e.

$$H(X_i) = \bar{M}\dot{\mathbf{u}}_{c_i} + \bar{C}(\mathbf{u}_i)\mathbf{u}_i + \bar{F}(\mathbf{u}_i) = W^{*T}S(X_i) + \epsilon_i, \tag{30}$$

in which $S(X_i) \in \mathbb{R}^N$ is a vector of RBFs, where $X_i = \begin{bmatrix}\dot{\mathbf{u}}_{c_i}^T & \mathbf{u}_i^T\end{bmatrix}^T$ is the input vector of the NN, and N is the number of neurons in the network, $W^* \in \mathbb{R}^{N\times 2}$ is the common ideal approximation weight of this RBFNN, and ϵ_i is the ideal approximation error, which can be made arbitrarily small given sufficiently large number of neurons. Consequently, we propose the implementable controller for the i^{th} vehicle by replacing the unstructured system uncertainties with the RBFNN approximation

$$\bar{\tau}_i = \hat{W}_i^T S(X_i) + K_u\tilde{\mathbf{u}}_i + \begin{bmatrix}\tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y}\end{bmatrix}, \tag{31}$$

where $\hat{W}_i$ is the approximation of the ideal NN weight by the i^{th} vehicle agent. To drive this approximation $\hat{W}_i$ converging to the common ideal value W^*, we propose an online NN weight updating law as follows

$$\dot{\hat{W}}_i = \Gamma S(X_i)\tilde{\mathbf{u}}_i^T - \gamma\hat{W}_i - \beta\sum_{j=1, j\neq i}^{n} a_{ij}(\hat{W}_i - \hat{W}_j), \tag{32}$$

in which $\Gamma S(X_i)\tilde{\mathbf{u}}_i$ is the local updating term using the tracking error $\tilde{\mathbf{u}}_i$, $-\gamma\hat{W}_i$ is the leakage term inspired from the robust adaptive control to stabilize the system in case of disturbance [3], and $-\beta\sum_{j=1}^{n} a_{ij}(\hat{W}_i - \hat{W}_j)$ is the cooperative updating term to share information/knowledge inside the MAS. Γ and β are positive constants, γ is a positive constant close to zero, $a_{ij} = 1$ if agent i and j are connected, and $a_{ij} = 0$ otherwise.

Theorem 1. *Consider the closed-loop system including n unicycle-type vehicles described by equation (7) and (17), the desired reference trajectory $\mathbf{q}_r(t)$, adaptive*

NN controller (31) with the virtual velocity (22), and the online weight updating law (32), under the assumptions 1, 2, and 3, then for any bounded initial condition of all the vehicles and $\hat{W}_i = 0$, the tracking error $\tilde{\mathbf{q}}_i$ converges asymptotically to a small neighborhood around zero for all vehicle agents in the MAS.

Proof. From equation (27) and (31), we have the error dynamics of the velocity error $\tilde{\mathbf{u}}_i$ as

$$\begin{aligned} \dot{\tilde{\mathbf{u}}}_i &= \bar{M}^{-1}(\bar{M}\dot{\mathbf{u}}_{c_i} + \bar{C}\mathbf{u}_i + \bar{F} - \bar{\tau}_i) \\ &= \bar{M}^{-1}\left\{W^{*T}S(X_i) + \epsilon_i - \hat{W}_i^T S(X_i) - K_u\tilde{\mathbf{u}}_i - \begin{bmatrix} \tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y} \end{bmatrix}\right\} \\ &= \bar{M}^{-1}\left\{\tilde{W}_i^T S(X_i) + \epsilon_i - K_u\tilde{\mathbf{u}}_i - \begin{bmatrix} \tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y} \end{bmatrix}\right\} \end{aligned} \tag{33}$$

By defining the error between the ideal and actual weight as $\tilde{W}_i = W^* - \hat{W}_i$, the error dynamics of the NN weight is

$$\begin{aligned} \dot{\tilde{W}}_i = -\dot{\hat{W}}_i &= -\Gamma S(X_i)\tilde{\mathbf{u}}_i^T + \gamma\hat{W}_i + \beta\sum_{j=1,j\neq i}^{n} a_{ij}(\hat{W}_i - \hat{W}_j) \\ &= -\Gamma S(X_i)\tilde{\mathbf{u}}_i^T + \gamma\hat{W}_i + \beta\sum_{j=1,j\neq i}^{n} a_{ij}(-(W^* - \hat{W}_i) + (W^* - \hat{W}_j)) \\ &= -\Gamma S(X_i)\tilde{\mathbf{u}}_i^T + \gamma\hat{W}_i - \beta\sum_{j=1,j\neq i}^{n} a_{ij}(\tilde{W}_i - \tilde{W}_j) \end{aligned} \tag{34}$$

or equivalently,

$$\dot{\tilde{W}}_i = -\Gamma S(X_i)\tilde{\mathbf{u}}_i^T + \gamma\hat{W}_i - \beta\sum_{j=1}^{n} l_{ij}\tilde{W}_j \tag{35}$$

where l_{ij} is the $(i,j)^{\text{th}}$ entry of the Laplacian matrix L associated with the undirected graph $\mathcal{G}$.

For the closed-loop system given by equation (25), (33), and (35), we can build a new Lyapunov function of the whole MAS as

$$V = \sum_{i=1}^{n}\left[\frac{\tilde{x}_i^2}{2} + \frac{\tilde{y}_i^2}{2} + \frac{(1-\cos\tilde{\theta}_i)}{K_y} + \frac{\tilde{\mathbf{u}}_i^T\bar{M}\tilde{\mathbf{u}}_i}{2} + \frac{\text{trace}(\tilde{W}_i^T\tilde{W}_i)}{2\Gamma}\right] \tag{36}$$

whose derivative is

$$\dot{V} = \sum_{i=1}^{n} \left[\tilde{x}_i \dot{\tilde{x}}_i + \tilde{y}_i \dot{\tilde{y}}_i + \frac{\sin \tilde{\theta}_i}{K_y} \dot{\tilde{\theta}}_i + \tilde{\mathbf{u}}_i^T \bar{M} \dot{\tilde{\mathbf{u}}}_i + \frac{\text{trace}(\tilde{W}_i^T \dot{\hat{W}}_i)}{\Gamma} \right] \tag{37}$$

Apply equation (25), (33), and (35) to (37), we have

$$\begin{aligned}
\dot{V} &= \sum_{i=1}^{n} \Bigg\{ \tilde{x}_i(\tilde{v}_i + \omega_i \tilde{y}_i - K_x \tilde{x}_i) + \tilde{y}_i(v_{r_i} \sin \tilde{\theta}_i - \omega_i \tilde{x}_i) \\
&\quad + \frac{\sin \tilde{\theta}_i}{K_y}(\tilde{\omega}_i - v_{r_i} K_y \tilde{y}_i - K_\theta \sin \tilde{\theta}_i) + \tilde{\mathbf{u}}_i^T \left[\tilde{W}_i^T S(X_i) + \epsilon_i - K_u \tilde{\mathbf{u}}_i - \begin{bmatrix} \tilde{x}_i \\ \frac{\sin \tilde{\theta}_i}{K_y} \end{bmatrix} \right] \\
&\quad + \text{trace} \left(\tilde{W}_i^T \left[-S(X_i) \tilde{\mathbf{u}}_i^T + \frac{\gamma \hat{W}_i}{\Gamma} - \frac{\beta}{\Gamma} \sum_{j=1}^{n} l_{ij} \tilde{W}_j \right] \right) \Bigg\} \\
&= \sum_{i=1}^{n} \Bigg\{ -K_x \tilde{x}_i^2 - \frac{K_\theta}{K_y} \sin^2 \tilde{\theta}_i - K_u \tilde{\mathbf{u}}_i^T \tilde{\mathbf{u}}_i + \tilde{\mathbf{u}}_i^T \epsilon_i + \tilde{\mathbf{u}}_i^T [\tilde{W}_i^T S(X_i)] \\
&\quad - \text{trace} \left([\tilde{W}_i^T S(X_i)] \tilde{\mathbf{u}}_i^T \right) + \text{trace} \left(\frac{\gamma \tilde{W}_i^T \hat{W}_i}{\Gamma} \right) \Bigg\} \\
&\quad - \text{trace} \left(\sum_{i=1}^{n} \frac{\beta}{\Gamma} \tilde{W}_i^T \sum_{j=1}^{n} l_{ij} \tilde{W}_j \right) \\
&= \sum_{i=1}^{n} \left\{ -K_x \tilde{x}_i^2 - \frac{K_\theta}{K_y} \sin^2 \tilde{\theta}_i - K_u \tilde{\mathbf{u}}_i^T \tilde{\mathbf{u}}_i + \tilde{\mathbf{u}}_i^T \epsilon_i + \frac{\gamma}{\Gamma} \text{trace} \left(\tilde{W}_i^T \hat{W}_i \right) \right\} \\
&\quad - \frac{\beta}{\Gamma} \text{trace} \left(\tilde{W}^T (L \otimes I) \tilde{W} \right)
\end{aligned} \tag{38}$$

where $\tilde{W} = \begin{bmatrix} \tilde{W}_1^T & \cdots & \tilde{W}_n^T \end{bmatrix}^T$, and the operator $\otimes$ denotes the Kronecker product of two matrices. Since β and Γ are all positive, and L is positive semi-definite, then we have $\frac{\beta}{\Gamma} \text{trace} \left(\tilde{W}^T (L \otimes I) \tilde{W} \right) \geq 0$. Notice that the approximation error can be made arbitrarily small with a sufficient large number of neurons, and γ is a small positive constant. Therefore, we can conclude that the closed-loop system (25), (33), and (35) is stable, i.e. $\dot{V} \leq 0$, if the following condition stands

$$K_x \tilde{x}_i^2 + \frac{K_\theta}{K_y} \sin^2 \tilde{\theta}_i + K_u \tilde{\mathbf{u}}_i^T \tilde{\mathbf{u}}_i \geq \tilde{\mathbf{u}}_i^T \epsilon_i + \frac{\gamma}{\Gamma} \text{trace} \left(\tilde{W}_i^T \hat{W}_i \right) \tag{39}$$

Hence, the closed-loop system is stable, and all tracking error are bounded. Since all variables in (37) are continuous and bounded (i.e. $\ddot{V}$ is bounded), then with the application of Barbalat's lemma [5], we have $\lim_{t\to\infty} \dot{V} = 0$, which implies that the tracking error $\tilde{\mathbf{q}}_i$ for all agents will converge to a small neighborhood of zero, whose size depends on the norm of $\tilde{\mathbf{u}}_i^T \epsilon_i + \frac{\gamma}{\Gamma} \operatorname{trace}\left(\tilde{W}_i^T \hat{W}_i\right)$. □

To this point, we have fulfilled the first objective of trajectory tracking raised in Section 3.1. In the following part, we will further show that convergence of NN weight approximation by each vehicle agent is guaranteed not only along its own trajectory, but along trajectories experienced by other vehicle agents as well, i.e., $\hat{W}_i$ for all vehicle agents will converge to the common ideal approximation weight W^* along the union trajectory (denoted as $\cup_{i=1}^{n} \zeta_i[X_i(t)]$) experienced by all vehicles in the MAS.

By defining $\tilde{v} = \begin{bmatrix}\tilde{v}_1 & \dots & \tilde{v}_n\end{bmatrix}^T$, $\tilde{\omega} = \begin{bmatrix}\tilde{\omega}_1 & \dots & \tilde{\omega}_n\end{bmatrix}^T$, $\tilde{W}_v = \begin{bmatrix}\tilde{W}_{1,1} & \dots & \tilde{W}_{n,1}\end{bmatrix}^T$, $\tilde{W}_\omega = \begin{bmatrix}\tilde{W}_{1,2} & \dots & \tilde{W}_{n,2}\end{bmatrix}^T$, and $\mathbf{S} = \operatorname{diag}(S(X_1), S(X_2), \dots, S(X_n))$, we combine the error dynamics in equations (33) and (35) for all vehicles into the following form:

$$\begin{bmatrix}\dot{\tilde{v}} \\ \dot{\tilde{\omega}} \\ \dot{\tilde{W}}_v \\ \dot{\tilde{W}}_\omega\end{bmatrix} = \begin{bmatrix} A & B \\ -C & -D \end{bmatrix} \begin{bmatrix}\tilde{v} \\ \tilde{\omega} \\ \tilde{W}_v \\ \tilde{W}_\omega\end{bmatrix} + E \tag{40}$$

in which

$$
\begin{aligned}
A_{2n\times 2n} &= -K_u(\bar{M}^{-1}\otimes I_n),\\
B_{2nN\times 2n} &= (\bar{M}^{-1}\otimes I_n)(I_2\otimes \mathbf{S}) = \bar{M}^{-1}\otimes \mathbf{S}^T,\\
C_{2n\times 2nN} &= \Gamma(I_2\otimes \mathbf{S}),\\
D_{2nN\times 2nN} &= \beta I_2\otimes (L\otimes I_N),\\
E_{(2nN+2nN)\times 1} &= \begin{bmatrix} E_1\\ E_2\\ E_3\\ E_4 \end{bmatrix},\\
E_1 &= \frac{1}{m}\begin{bmatrix} \epsilon_{v_1}-\tilde{x}_1\\ \vdots\\ \epsilon_{v_n}-\tilde{x}_n \end{bmatrix},\\
E_2 &= \frac{1}{I}\begin{bmatrix} \epsilon_{\omega_1}-\frac{\sin\tilde{\theta}_1}{K_y}\\ \vdots\\ \epsilon_{\omega_n}-\frac{\sin\tilde{\theta}_n}{K_y} \end{bmatrix},\\
E_3 &= \gamma\begin{bmatrix} \hat{W}_{1,1}\\ \vdots\\ \hat{W}_{n,1} \end{bmatrix},\\
E_4 &= \gamma\begin{bmatrix} \hat{W}_{1,2}\\ \vdots\\ \hat{W}_{n,2} \end{bmatrix}.
\end{aligned}
$$

As is shown in Theorem 1, the tracking errors $\tilde{v}_i$ and $\tilde{\omega}_i$ will converge to a small neighborhood around zero, $\forall i = 1, \ldots, n$. Furthermore, the ideal approximation errors ϵ_{v_i} and ϵ_{ω_i} can be made arbitrarily small given sufficiently large number of RBF neurons, and γ is chosen to be a small positive constant, therefore, we can conclude that the norm of E in equation (40) has a small value. In the following theorem, we will show that $W_i = \begin{bmatrix} W_{i,1} & W_{i,2} \end{bmatrix}$ converges to a small neighborhood of the common ideal weight W^* for all $i = 1, \ldots, n$ under assumptions 1 and 3.

Before proceeding further, we denote the parts of the NN weight W related to the region close to and away from the trajectory ζ as W_ζ and $W_{\bar{\zeta}}$, respectively

[6].

Theorem 2. *Consider the error dynamics (40), under the assumptions 1, 2, and 3, then for any bounded initial condition of all the vehicles and* $\hat{W}_i = 0$*, along the union of the system trajectories* $\cup_{i=1}^{n} \zeta_i[X_i(t)]$*, all locally approximated neural weights* $\hat{W}_{\zeta_i}$ $(i = 1, \cdots, n)$ *used in (31) converge to a small neighborhood of their common ideal value* W_{ζ}^{*}*, and locally accurate identification of nonlinear uncertain dynamics* $H(X(t))$ *can be achieved by* $\hat{W}_i^T S(X)$ *as well as* $\bar{W}_i^T S(X)$ *obtained from any vehicle agent in the MAS, for all* $X \in \cup_{i=1}^{n} \zeta_i[X_i(t)]$*, where*

$$\bar{W}_i = \underset{t_{a_i} \leq t \leq t_{b_i}}{\text{mean}} \hat{W}_i(t) \tag{41}$$

with $[t_{a_i}, t_{b_i}]$ $(t_{b_i} > t_{a_i} > T_i)$ *being a time segment after the transient period of tracking control.*

Proof. According to [6], if the nominal part of closed loop system shown in (40) is uniformly locally exponentially stable (ULES), then $\tilde{v}$, $\tilde{\omega}$, $\tilde{W}_v$, and $\tilde{W}_\omega$ will converge to a small neighborhood of the origin, whose size depends on the value of $||E||$.

Now the problem boils down to proving ULES of the nominal part of system (40). To this end, we need to resort to the results of Lemma $5'$ in [7]. It is stated that if the Assumptions $1'$ and $2'$ therein are satisfied, then the nominal part of (40) is ULES if there exist two positive constants T_0 and α, such that

$$\int_{t}^{t+T_0} [(B^T B) + D] d\tau \geq \alpha I_{2nN} \tag{42}$$

holds for all $t \geq 0$.

The assumption $1'$ therein is automatically verified since $\mathbf{S}$ is bounded, and Assumption $2'$ therein also holds if we set the counterparts $P = \Gamma(\bar{M} \otimes I_n)$ and $Q = 2\Gamma K_u I_{2n}$. Furthermore, the PE condition presented by equation (42) is met if the RBFNN input signal X_i is recurrent for all $i = 1, \cdots, n$ [6], which is guaranteed by our Assumption 3 and results from Theorem 1. Therefore, we can obtain the conclusion that $\tilde{v}$, $\tilde{\omega}$, $\tilde{W}_v$, and $\tilde{W}_\omega$ will converge to the neighborhood of the origin, whose size depends on the small value of $||E||$.

Similar to [8], the convergence of $\hat{W}_{\zeta i}$ to a small neighborhood of W_ζ^* implies that for all $X \in \cup_{i=1}^n \zeta_i[X_i(t)]$, we have

$$H(X) = W_\zeta^{*T} + \epsilon_\zeta = \hat{W}_{\zeta_i}^T S_\zeta(X) + \tilde{W}_{\zeta_i}^T S_\zeta(X) + \epsilon_{\zeta i} = \hat{W}_{\zeta_i}^T S_\zeta(X) + \epsilon_{1\zeta_i} \tag{43}$$

where $\epsilon_{1\zeta_i} = \tilde{W}_{\zeta_i}^T S_\zeta(X) + \epsilon_{\zeta_i}$ is close to ϵ_{ζ_i} due to the convergence of $\tilde{W}_{\zeta_i}$. With the $\bar{W}_i$ defined in (41), then equation (43) can be rewritten into

$$\begin{aligned} H(X) &= \hat{W}_{\zeta_i}^T S_\zeta(X) + \epsilon_{1\zeta_i} \\ &= \bar{W}_{\zeta_i}^T S_\zeta(X) + \epsilon_{2\zeta_i} \end{aligned} \tag{44}$$

where $\bar{W}_{\zeta_i}^T = \begin{bmatrix} w_{1_\zeta} & \cdots & w_{k_\zeta} \end{bmatrix}^T$ is a subvector of $\bar{W}_i$ and $\epsilon_{2\zeta_i}$ is the error using $\bar{W}_{\zeta_i}^T S_\zeta(X)$ as the system approximation. After the transient process, $||\epsilon_{1\zeta_i}||$ and $||\epsilon_{2\zeta_i}||$ are small for all $i = 1, \cdots, n$.

On the other hand, due to the localization property of Gaussian RBFs, both $S_{\bar{\zeta}}$ and $\bar{W}_{\bar{\zeta}} S_{\bar{\zeta}}(X)$ are very small. Hence, along the union trajectory $\cup_{i=1}^n \zeta_i[X_i(t)]$, the entire constant RBF network $\bar{W}^T S(X)$ can be used to approximate the nonlinear uncertain dynamics

$$\begin{aligned} H(X) &= W_\zeta^{*T} S_\zeta(X) + \epsilon_\zeta \\ &= \hat{W}_{\zeta_i}^T S_\zeta(X) + \hat{W}_{\bar{\zeta}_i}^T S_{\bar{\zeta}}(X) + \epsilon_{1_i} \\ &= \hat{W}_i^T S(X) + \epsilon_{1_i} \\ &= \bar{W}_{\zeta_i}^T S_\zeta(X) + \bar{W}_{\bar{\zeta}_i}^T S_{\bar{\zeta}}(X) + \epsilon_{2_i} \\ &= \bar{W}_i^T S(X) + \epsilon_{2_i} \end{aligned} \tag{45}$$

where $||\epsilon_{1_i}||$, $||\epsilon_{1\zeta_i}||$, $||\epsilon_{2_i}||$, and $||\epsilon_{2\zeta_i}||$ are all small for all $i = 1, \cdots, n$. Therefore, the conclusion of Theorem 2 can be drawn. □

3.3 Experience-based Controller Design and Stability Analysis

After showing that the NN weight for any vehicle agent in the MAS locally converged to the common ideal value W^* along the union trajectory $\cup_{i=1}^{n} \zeta_i[X_i(t)]$, we further propose an experience-based trajectory tracking controller without the adaptive/updating operations for decreasing the computational complexity. In addition, we will show that with any NN weight $\bar{W}_i$ obtained from equation (41), all vehicle agents in the MAS are able to track the trajectory not only experienced by vehicle agent i, but all trajectories experienced in the learning phase as well.

To this end, we replace the NN weight $\hat{W}_i$ in equation (31) by the converged constant NN weight $\bar{W}_j$ taken from a random vehicle agent j in the MAS. Then for the i^{th} vehicle, the experience-based controller is thereby constructed

$$\bar{\tau}_i = \bar{W}_j^T S(X_i) + K_u \tilde{\mathbf{u}}_i + \begin{bmatrix} \tilde{x}_i \\ \frac{\sin \tilde{\theta}_i}{K_y} \end{bmatrix}, \tag{46}$$

Theorem 3. *Consider the closed-loop system including n unicycle-type vehicles described by equation (7) and (17), the desired reference trajectory $\mathbf{q}_{ri} \in \cup_{j=1}^{n} \mathbf{q}_j(t)$, and the experience-based controller (46) with the virtual velocity (22), for any bounded initial condition, the tracking error $\tilde{\mathbf{q}}_i$ converges asymptotically to a small neighborhood around zero, $\forall i = 1, \cdots, n$.*

Proof. Similar to the proof of Theorem 1, we first derive the error dynamics of the velocity error $\mathbf{u}_i$ with the proposed experience-based controller (46) as

$$\begin{aligned} \dot{\tilde{\mathbf{u}}}_i &= \bar{M}^{-1}(\bar{M}\dot{\mathbf{u}}_{c_i} + \bar{C}\mathbf{u}_i + \bar{F} - \bar{\tau}_i) \\ &= \bar{M}^{-1} \left\{ \epsilon_{2_j} - K_u \tilde{\mathbf{u}}_i - \begin{bmatrix} \tilde{x}_i \\ \frac{\sin \tilde{\theta}_i}{K_y} \end{bmatrix} \right\} \end{aligned} \tag{47}$$

where $\epsilon_{2_j} = H(X_j) - \bar{W}_j^T S(X_j)$ is small along the union of reference trajecotries. Similarly, we can build a positive definite function of this closed-loop system of the i^{th} vehicle as

$$V_i = \frac{\tilde{x}_i^2}{2} + \frac{\tilde{y}_i^2}{2} + \frac{1 - \cos \tilde{\theta}_i}{K_y} + \frac{\tilde{\mathbf{u}}_i^T \bar{M} \tilde{\mathbf{u}}_i}{2} \tag{48}$$

whose derivative is

$$\begin{aligned}\dot{V}_i &= \tilde{x}_i\dot{\tilde{x}}_i + \tilde{y}_i\dot{\tilde{y}}_i + \frac{\sin\tilde{\theta}_i}{K_y}\dot{\tilde{\theta}}_i + \tilde{\mathbf{u}}_i^T\bar{M}\dot{\tilde{\mathbf{u}}}_i \\ &= -K_x\tilde{x}_i^2 - \frac{K_\theta}{K_y}\sin^2\tilde{\theta}_i - K_u\tilde{\mathbf{u}}_i^T\tilde{\mathbf{u}}_i + \tilde{\mathbf{u}}_i^T\epsilon_{2_j}\end{aligned} \tag{49}$$

Since the Lyapunov function V_i is positive definite and $\dot{V}_i$ is negative definite in the region $K_x\tilde{x}_i^2 + \frac{K_\theta}{K_y}\sin^2\tilde{\theta}_i + K_u\tilde{\mathbf{u}}_i^T\tilde{\mathbf{u}}_i \geq \tilde{\mathbf{u}}_i^T\epsilon_{2_j}$, under the condition that K_x, K_y, K_θ, and K_u are all positive, we can thereby conclude that the closed-loop system for trajectory tracking with the experience-based controller (46) is asymptotically stable, and the tracking error for any vehicle will converge to a small neighborhood of zero, whose size depends on the norm of $\tilde{\mathbf{u}}_i^T\epsilon_{2_j}$. □

3.4 Simulation Study (MATLAB Results)

In this section, we conduct a simulation using the MATLAB to demonstrate the effectiveness of the proposed controllers. Specifically, we consider four identical unicycle-type ground vehicles tracking four different trajectories while communicating inside the MAS. For each vehicle, the state-space model is described by the kinematics (7) and dynamics (17), with matrices of the dynamic model given by (18) and friction model $\bar{F} = \begin{bmatrix} 0.1mv_i + 0.05mv_i^2 \\ 0.2I\omega_i + 0.1I\omega_i^2 \end{bmatrix}$, all unknown to the controller. The physical parameters of the vehicles are shown in Table 1.

Table 1: Parameters of the vehicle model in MATLAB

vehicle model parameter	value
wheel radius (r)	0.06 m
wheel separation ($D = 2R$)	0.22 m
location of mass center (d)	0.1 m
mass (m)	2 kg
moment of inertia ($I = md^2 + I_c$)	0.2 kg $\cdot$ m^2

The the reference trajectories of the three vehicles are given by

$$\begin{cases} x_{r_1} = -\sin t \\ y_{r_1} = 2\cos t \end{cases} \quad \begin{cases} x_{r_2} = 2\cos t \\ y_{r_2} = \sin t \end{cases} \quad \begin{cases} x_{r_3} = -2\sin t \\ y_{r_3} = 3\cos t \end{cases} \quad \begin{cases} x_{r_4} = 3\cos t \\ y_{r_4} = 2\sin t \end{cases} \tag{50}$$

and for all vehicles, the orientations and velocities of the reference trajectories satisfy the following equations

$$\begin{aligned} \tan\theta_{ri} &= \frac{\dot{y}_{ri}}{\dot{x}_{ri}}, \\ v_{ri} &= \sqrt{\dot{x}_{ri}^2 + \dot{y}_{ri}^2}, \\ \omega_{ri} &= \frac{\dot{x}_{ri}\ddot{y}_{ri} - \ddot{x}_{ri}\dot{y}_{ri}}{\dot{x}_{ri}^2 + \dot{y}_{ri}^2}. \end{aligned} \tag{51}$$

The parameters of the cooperative adaptive-based controller (31) with (22) and weight updating law (32), as well as the experience-based controller (46), are given by Table 2

Table 2: Parameters of controllers and continuous-time weight updating law

controller parameter	value
K_x	1
K_y	1
K_θ	1
K_u	2
Γ	100
γ	0.001
β	10

For each $i = 1, 2, 3, 4$, we normalize $X_i = \begin{bmatrix} \dot{\mathbf{u}}_{c_i}^T & \mathbf{u}_i^T \end{bmatrix}^T \in \mathbb{R}^{4\times 1}$ from $[-2, 2] \times [-2, 2] \times [0, 4] \times [0, 4]$ to the space $[-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1]$, then we construct the Gaussian RBFNN $\hat{W}_i S(X_i)$ using $N = 5 \times 5 \times 5 \times 5 = 625$ neuron nodes with the centers evenly placed over that space and the respective field η of the Gaussian function equal to 0.5. The connection between vehicle agents is shown in Figure 3, and the Laplacian matrix L associated with the graph $\mathcal{G}$ is

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

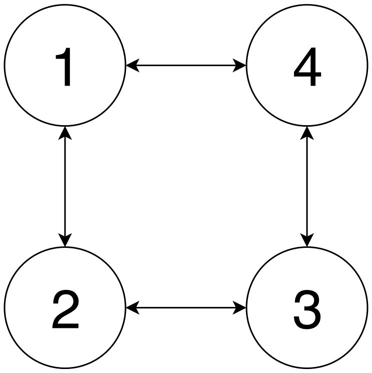
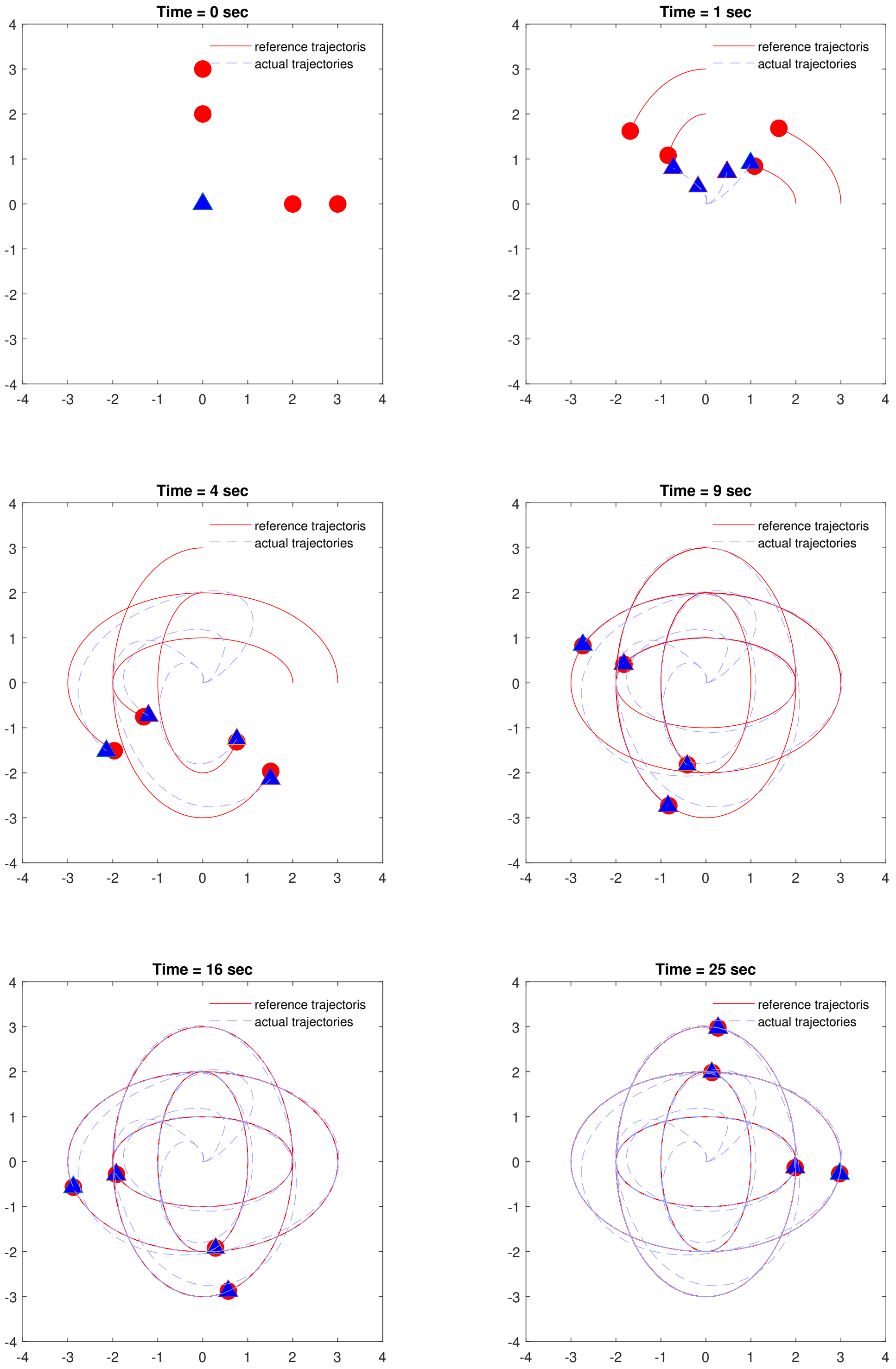


Figure 3: Connection between four vehicles

For the simulation on MATLAB, the function `ode45` is used to solve a group of ordinary differential equations (ODEs), including the vehicle kinematics (7), dynamics (17), and the weight updating law (32). The vehicles are simulated on the time period from 0 to 300 seconds, with the initial position of the vehicles set at the origin of the ground frame, the velocities set to be zero, and the initial weights of RBFNNs set to be zero as well. Simulation results are shown as follows.

Figure 4 shows that all vehicles (blue triangles) will track their own reference trajectories (red solid circles), and Figure 5 shows that all tracking errors $\tilde{x}_i$ and $\tilde{y}_i$ will converge to zero, within 30 seconds of the simulation in the learning phase. Figure 6 shows that the norm of the NN weights $\hat{W}_i$ for all vehicle agents converge to the same constant, and Figure 7 shows that the approximation errors of all vehicle agents converg to a small neighborhood around zero, suggesting that all vehicle agents in the MAS are able to accurately approximate the unstructured system uncertainties $\bar{M}\dot{\mathbf{u}}_{c_i} + \bar{C}_i\mathbf{u}_i + \bar{F}_i$ along the union of trajectories.


Time = 0 sec

Time = 1 sec

Time = 4 sec

Time = 9 sec

Time = 16 sec

Time = 25 sec

Figure 4: Snapshot of trajectory tracking using CALC.

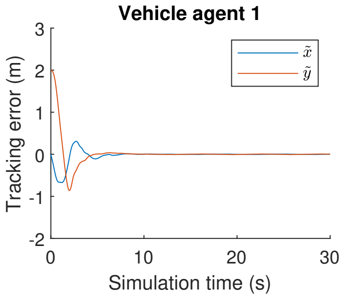


Figure 5: Tracking errors using CALC.

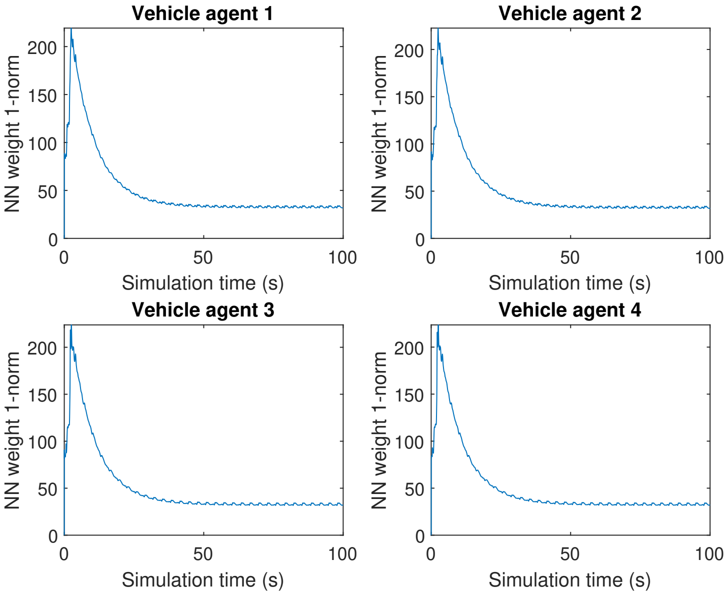


Figure 6: Weight vector 1-norm of $\hat{W}_i$.

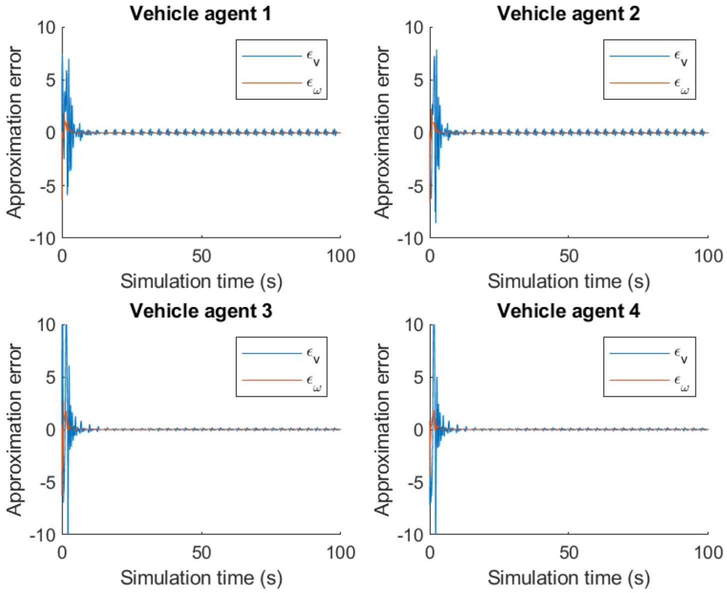


Figure 7: Approximation errors using CALC.

After the learning phase shown above, we calculate and store the average of NN weights over the last 3 seconds using equation (41). With this learned knowledge, we also show the tracking convergence using the experience-based controller (46). Figures 8 and 9 show that the tracking errors converge to zero at a similar rate as the CDL-based controller did, with all vehicle agents using the learned knowledge $\bar{W}$ randomly selected from vehicle agent 3.

3.5 Summary

In this chapter, we proposed a cooperative adaptive learning-based controller (CALC) and an experience-based controller for the trajectory tracking problem of a group of identical unicycle-type ground vehicles. Theoretical analysis are provided to show that the adaptive tracking convergence (Theorem 1), cooperative learning consensus and accurate approximation (Theorem 2), as well as trajectory tracking using learned knowledge (Theorem 3), can all be achieved by the proposed control algorithms. MATLAB simulation results show that the objectives set in section 3.1

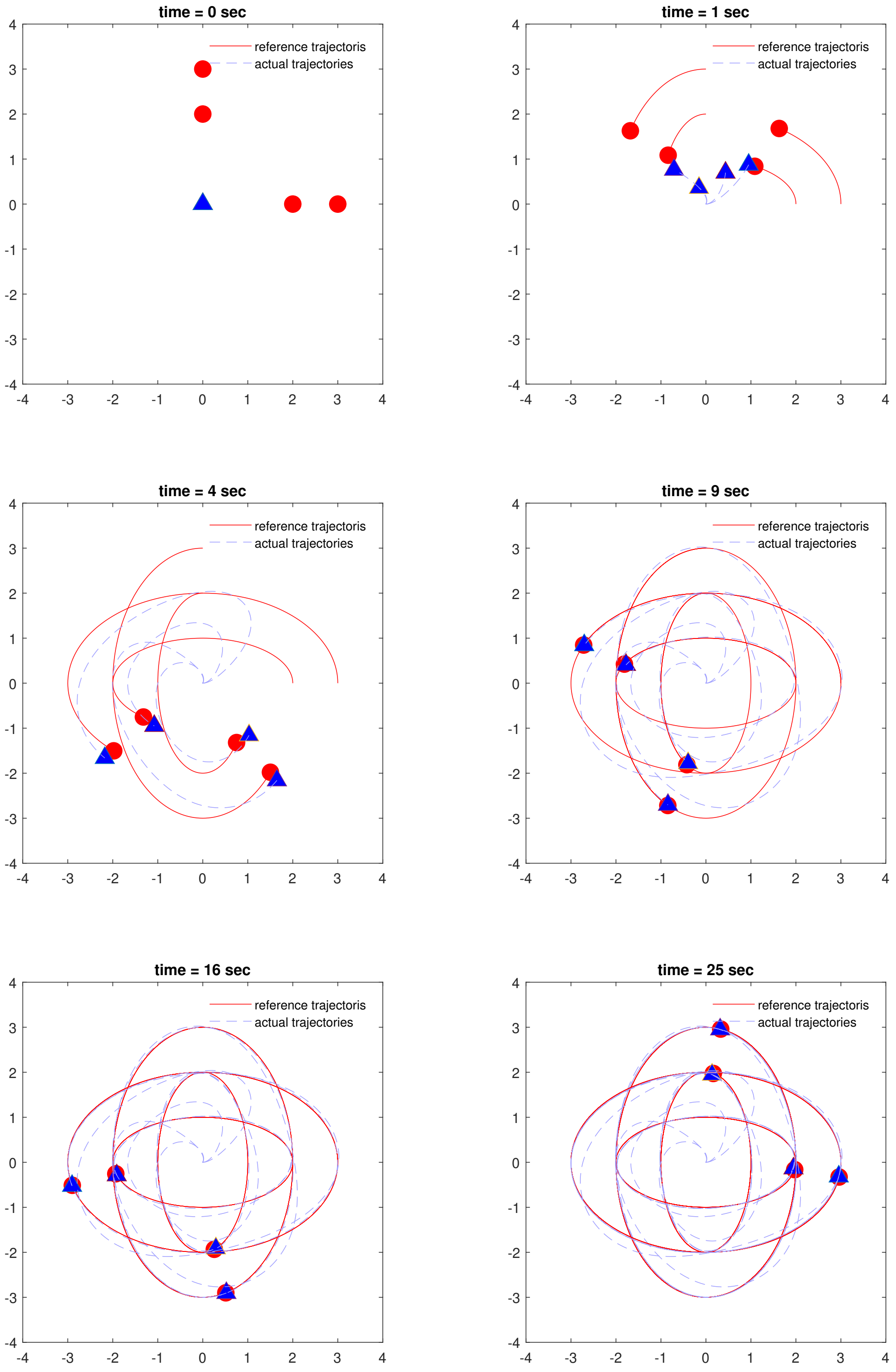


Figure 8: Snapshot of trajectory tracking using experience-based controller.

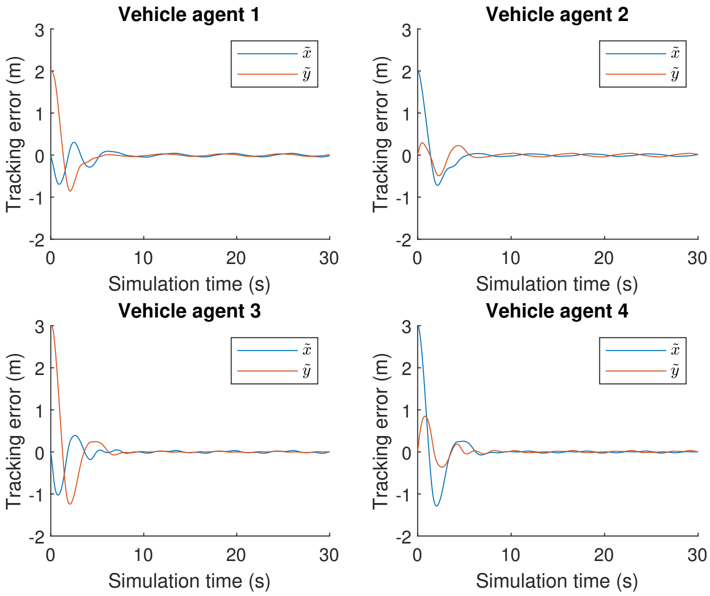


Figure 9: Tracking errors using experience-based controller.

are reached by the proposed CALC and experience-based controller.

List of References

[1] H. J. Marquez, *Nonlinear Control Systems: Analysis and Design*. Hoboken: Wiley-Interscience, 2003, vol. 1.

[2] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. IEEE, 1990, pp. 384–389.

[3] L. Eugene, W. Kevin, and D. Howe, *Robust and adaptive control with aerospace applications*. Springer London, 2013.

[4] W. Zeng, Q. Wang, F. Liu, and Y. Wang, "Learning from adaptive neural network output feedback control of a unicycle-type mobile robot," *ISA transactions*, vol. 61, pp. 337–347, 2016.

[5] I. Barbalat, "Systemes d'équations différentielles d'oscillations non linéaires," *Rev. Math. Pures Appl*, vol. 4, no. 2, pp. 267–270, 1959.

[6] C. Wang and D. J. Hill, *Deterministic learning theory for identification, recognition, and control*. CRC Press, 2009, vol. 32.

[7] W. Chen, C. Wen, S. Hua, and C. Sun, "Distributed cooperative adaptive identification and control for a group of continuous-time systems with a cooperative

pe condition via consensus," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 91–106, 2014.

[8] C. Wang and D. J. Hill, "Learning from neural control," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 130–146, 2006.

CHAPTER 4

High-gain observer-based CALC via output feedback

4.1 Problem Statement

The state feedback controllers proposed in Chapter 3, including both learning-based (31) and experience-based (46), require the generalized velocities $\mathbf{u}_i$ to calculate the velocity error $\tilde{\mathbf{u}}_i$ and the RBFNN input X_i. Directly measuring the velocity of an objective (e.g. Doppler effect method [1]) is usually difficult to be applied on small UGVs. For indirect velocity measurements, the most commonly used method is to apply an encoder or a rev-counter to the actuator and count the steps in a certain time period, then calculate the speed of the wheel using the gear reduction rate. Despite the simplicity of hardware and calculation required by this method, encoder/rev-counter based measurements also bring new issues. One issue is that the calculation of vehicle's velocities from the speed of the wheels is based on the assumption that no slippery exists between the wheel and the ground, which may not be applicable on the real vehicle control. Another potential issue is the sampling interval of the hardware for encoders might not be short enough when the wheel is rotating at low speed.

With the issues brought by proprioceptive sensors (e.g. encoder, rev-counter) and the incapability of the exteroceptive sensors (e.g. Doppler radar), we use the high-gain observer in this research to estimate the vehicle's velocities from the measurement of the coordinates. To be specific, consider the same MAS used in Chapter 3 with only vehicle's generalized coordinates measured, a high-gain observer will be developed to accurately estimate the vehicle's generalized velocities, and the state feedback controllers proposed in the previous chapter are modified to control the multi-vehicle system with the estimated velocities, such that all objectives raised in Chapter 3, including tracking convergence, learning consensus, and experience-based trajectory tracking control can also be achieved via output feedback.

4.2 High-gain Observer-based Trajectory Tracking with CALC

In this section, we follow the high-gain observer design method in [2, 3] and introduce a high-gain observer to estimate the velocities using the measurement of robot positions. First, we define two new variables for the i^{th} vehicle agent as follows

$$\begin{aligned} p_{x_i} &= x_i \cos\theta_i + y_i \sin\theta_i, \\ p_{y_i} &= y_i \cos\theta_i - x_i \sin\theta_i. \end{aligned} \tag{52}$$

Notice that the operation above can be considered as a projecting the vehicle position onto the a rotational frame, whose origin is fixed to the origin of ground frame and the axes are parallel to the body-fixed frame of the vehicle. The coordinates of the vehicle in this rotational frame is (p_{x_i}, p_{y_i}), which can be calculated based on the measurement of the vehicle's generalized coordinates $\mathbf{q}_i$. The rotation rate of this frame equals to the vehicle's angular velocity ω_i. Based on these definitions, we first design the high-gain observer for the angular velocity ω_i of the i^{th} vehicle as

$$\begin{aligned} \dot{\hat{\theta}}_i &= \hat{\omega}_i + \frac{l_1}{\delta}(\theta_i - \hat{\theta}_i) \\ \dot{\hat{\omega}}_i &= \frac{l_2}{\delta^2}(\theta_i - \hat{\theta}_i) \end{aligned} \tag{53}$$

in which δ is a small positive scalar to be designed, and l_1 and l_2 are parameters to be chosen, such that $\begin{bmatrix} -l_1 & 1 \\ -l_2 & 0 \end{bmatrix}$ is Hurwitz stable, i.e., $l_1 > 0$ and $l_2 > 0$. The time derivative of this coordinates defined in (52) is given by

$$\begin{aligned} \dot{p}_{x_i} &= v_i + p_{y_i}\omega_i \\ \dot{p}_{y_i} &= -p_{x_i}\omega_i \end{aligned} \tag{54}$$

then we design the high-gain observer for the linear velocity v_i of the i^{th} vehicle based on equation (54) as

$$\begin{aligned}\dot{\hat{p}}_{x_i} &= \hat{v}_i + p_{y_i}\hat{\omega}_i + \frac{l_1}{\delta}(p_{x_i} - \hat{p}_{x_i}) \\ \dot{\hat{v}}_i &= \frac{l_2}{\delta^2}(p_{x_i} - \hat{p}_{x_i})\end{aligned} \tag{55}$$

Theorem 4. *Consider the closed-loop system for the i^{th} unicycle-type vehicles with the dynamics described by equation (17), and the proposed high-gain observer (53) and (55) with $l_1 > 0$, $l_2 > 0$, and δ being a small positive constant, for any bounded initial condition, the estimation error between the actual and estimated velocities $\mathbf{z}_i = \mathbf{u}_i - \hat{\mathbf{u}}_i$ converge asymptotically to a small neighborhood around zero, $\forall i = 1, \cdots, n$.*

Proof. To show the convergence of estimation error of the generalized velocity vector $\mathbf{z}_i = \mathbf{u}_i - \hat{\mathbf{u}}_i$ for the i^{th} vehicle, we further define the error between the actual and estimated coordinates as

$$\mathbf{e}_i = \begin{bmatrix} p_{x_i} - \hat{p}_{x_i} \\ \theta_i - \hat{\theta}_i \end{bmatrix}, \tag{56}$$

then the time derivative of $\mathbf{e}_i$ and $\mathbf{z}_i$ can be derived using (17), (53), and (55) as

$$\begin{aligned}\dot{\mathbf{e}}_i &= \begin{bmatrix} \dot{p}_{x_i} - \dot{\hat{p}}_{x_i} \\ \dot{\theta}_i - \dot{\hat{\theta}}_i \end{bmatrix} \\ &= \begin{bmatrix} v_i + p_{y_i}\omega_i - \hat{v}_i - p_{y_i}\hat{\omega}_i - \frac{l_1}{\delta}(p_{x_i} - \hat{p}_{x_i}) \\ \omega_i - \hat{\omega}_i - \frac{l_1}{\delta}(\theta_i - \hat{\theta}_i) \end{bmatrix} \\ &= -\frac{l_1}{\delta}\mathbf{e}_i + \begin{bmatrix} 1 & p_{y_i} \\ 0 & 1 \end{bmatrix}\mathbf{z}_i, \\ \dot{\mathbf{z}}_i &= \dot{\mathbf{u}}_i - \dot{\hat{\mathbf{u}}}_i \\ &= -\frac{l_2}{\delta^2}\mathbf{e}_i + \bar{M}^{-1}(\bar{\tau}_i - \bar{C}\mathbf{u}_i - \bar{F}_i),\end{aligned} \tag{57}$$

or in the equivalent matrix form of

$$\begin{bmatrix} \dot{\mathbf{e}}_i \\ \dot{\mathbf{z}}_i \end{bmatrix} = \begin{bmatrix} -(l_1/\delta)I_2 & P \\ -(l_2/\delta^2)I_2 & O_{2\times 2} \end{bmatrix} \begin{bmatrix} \mathbf{e}_i \\ \mathbf{z}_i \end{bmatrix} + \begin{bmatrix} O_{2\times 1} \\ \bar{M}^{-1}(\bar{\tau}_i - \bar{C}\mathbf{u}_i - \bar{F}_i) \end{bmatrix}, \tag{58}$$

where $P_{2\times 2} = \begin{bmatrix} 1 & p_{y_i} \\ 0 & 1 \end{bmatrix}$.

To show that the nominal part of (58) is stable, we analyze the eigenvalue of $\begin{bmatrix} -(l_1/\delta)I_2 & P \\ -(l_2/\delta^2)I_2 & O_{2\times 2} \end{bmatrix}$ by calculating the determinant of the following matrix

$$\lambda I_4 - \begin{bmatrix} -(l_1/\delta)I_2 & P \\ -(l_2/\delta^2)I_2 & O_{2\times 2} \end{bmatrix} = \begin{bmatrix} (\lambda + l_1/\delta)I_2 & -P \\ (l_2/\delta^2)I_2 & \lambda I_2 \end{bmatrix}. \tag{59}$$

For block square matrix, we have $\det(\begin{bmatrix} A & B \\ C & D \end{bmatrix}) = \det(D) \cdot \det(A - BD^{-1}C)$ [4], then the determinant of (59) equals to

$$\begin{aligned} &\det(\lambda I_2) \cdot \det((\lambda + \frac{l_1}{\delta})I_2 - (\frac{l_2}{\delta^2})I_2 \cdot (\lambda I_2)^{-1} \cdot P) \\ =&\lambda^2 \cdot \det(\begin{bmatrix} \lambda + \frac{l_1}{\delta} + \frac{l_2}{\lambda\delta^2} & \frac{l_2 p_{y_i}}{\lambda\delta^2} \\ 0 & \lambda + \frac{l_1}{\delta} + \frac{l_2}{\lambda\delta^2} \end{bmatrix}) \\ =&(\lambda^2 + \frac{l_1}{\delta}\lambda + \frac{l_2}{\delta^2})^2, \end{aligned} \tag{60}$$

Then the eigenvalues of the nominal part of (58) are the values of λ that make the determinant (60) singular. It can be easily shown that the real parts of all eigenvalues λ are negative, given the condition that l_1, l_2, and δ are all positive. In addition, the real parts of λ are proportional to $1/\delta$, then we can conclude that the settling time of the observer is smaller if the δ is chosen to be smaller.

Therefore, we have proved that the closed-loop system of the estimation error (58) is stable, and the estimation error $\mathbf{z}_i^T$ is bounded and converges to a constant vector, whose norm depends on the product of δ and the norm of $\bar{M}^{-1}(\bar{\tau}_i - \bar{C}\mathbf{u}_i - \bar{F}_i)$. Since δ is a small positive constant, and the norm of $\bar{M}^{-1}(\bar{\tau}_i - \bar{C}\mathbf{u}_i - \bar{F}_i)$ is bounded, then we can conclude that the estimation error $\mathbf{z}_i$ for the i^{th} vehicle will converge to a small neighborhood around zero. □

Remark 2. *To prevent peaking while using this high-gain observer and in turn improving the transient response, parameter δ cannot be too small [3]. Due to the use of a globally bounded control, decreasing δ does not induce peaking phenomenon* *of the state variables of the system, while the ability to decrease δ will be limited by practical factors such as measurement noise and sampling rates [5, 6].*

With the generalized velocities accurately estimated by the high-gain observer, we can now proceed to the trajectory tracking with this estimation. In the proposed state feedback controller (31) with (29) and the online weight updating law (32), the RBFNN input X_i contains the generalized velocity vector $\mathbf{u}_i$. In addition, $\dot{\mathbf{u}}_c$, $\mathbf{u}_c$, and $\tilde{\mathbf{u}}_i$ are all calculated using $\mathbf{u}_i$. By replacing $\mathbf{u}_i$ in (29), (31) and (32) with the estimated velocity vector $\hat{\mathbf{u}}_i$, we modify the state feedback controller (31) and the online weight updating law (32) as follows

$$\bar{\tau}_i = \hat{W}_i^T S(\hat{X}_i) + K_u(\mathbf{u}_{c_i} - \hat{\mathbf{u}}_i) + \begin{bmatrix} \tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y} \end{bmatrix}, \tag{61}$$

$$\dot{\hat{W}}_i = \Gamma S(\hat{X}_i)(\mathbf{u}_{c_i}^T - \hat{\mathbf{u}}_i^T) - \gamma\hat{W}_i - \beta\sum_{j=1,j\neq i}^{n} a_{ij}(\hat{W}_i - \hat{W}_j), \tag{62}$$

with $\hat{X}_i = \begin{bmatrix}\dot{\mathbf{u}}_{c_i}^T & \hat{\mathbf{u}}_i^T\end{bmatrix}^T$ and

$$\begin{aligned}\dot{\mathbf{u}}_{c_i} &= \begin{bmatrix} \dot{v}_{r_i} - v_{r_i}\sin\tilde{\theta}_i\dot{\tilde{\theta}}_i + K_x\dot{\tilde{x}}_i \\ \dot{\omega}_{r_i} + \dot{v}_{r_i}K_y\tilde{y}_i + v_{r_i}K_y\dot{\tilde{y}}_i + K_\theta\cos\tilde{\theta}_i\dot{\tilde{\theta}}_i \end{bmatrix} \\ &= \begin{bmatrix} \dot{v}_{r_i} - v_{r_i}\sin\tilde{\theta}_i(\omega_{r_i} - \hat{\omega}_i) + K_x(v_{r_i}\cos\tilde{\theta}_i + \hat{\omega}_i\tilde{y}_i - \hat{v}_i) \\ \dot{\omega}_{r_i} + \dot{v}_{r_i}K_y\tilde{y}_i + v_{r_i}K_y(v_{r_i}\sin\tilde{\theta}_i - \hat{\omega}_i\tilde{x}_i) + K_\theta\cos\tilde{\theta}_i(\omega_{r_i} - \hat{\omega}_i) \end{bmatrix},\end{aligned} \tag{63}$$

Theorem 5. *Consider the closed-loop system including n unicycle-type vehicles described by equation (7) and (17), the desired reference trajectory $\mathbf{q}_r(t)$, high-gain observer (53) and (55), adaptive NN controller (61) with the virtual velocity (22), and the online weight updating law (62), under the assumptions 1, 2, and 3, for any bounded initial condition of all the vehicles and $\hat{W}_i = 0$, both tracking control and learning objectives can be achieved at the same time for all vehicle agents in the MAS, i.e.,*

- *The tracking error $\tilde{\mathbf{q}}_i$ converges asymptotically to a small neighborhood around zero for all vehicle agents in the MAS.*

- *The NN weights $\hat{W}$ for all vehicles locally converge to a small neighborhood of the common optimal NN weights W^* along the union of trajectories experienced by all vehicles.*

Proof. With the output feedback controller (61), the error dynamics of $\tilde{\mathbf{u}}_i$ and $\tilde{W}_i$ can be rewritten into

$$\begin{aligned}\dot{\tilde{\mathbf{u}}}_i &= \bar{M}^{-1}(\bar{M}\dot{\mathbf{u}}_{c_i} + \bar{C}\mathbf{u}_i + \bar{F} - \bar{\tau}_i) \\ &= \bar{M}^{-1}\left\{W^{*T}S(X_i) + \epsilon_i - \hat{W}_i^T S(X_i) - K_u(\mathbf{u}_{c_i} - \mathbf{u}_i + \mathbf{u}_i - \hat{\mathbf{u}}_i) - \begin{bmatrix}\tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y}\end{bmatrix}\right\} \\ &= \bar{M}^{-1}\left\{\tilde{W}_i^T S(X_i) + \epsilon_i - K_u(\tilde{\mathbf{u}}_i + \mathbf{z}_i) - \begin{bmatrix}\tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y}\end{bmatrix}\right\},\end{aligned} \tag{64}$$

and

$$\dot{\tilde{W}}_i = -\Gamma S(X_i)(\tilde{\mathbf{u}}_i^T + \mathbf{z}_i^T) + \gamma\hat{W}_i - \beta\sum_{j=1}^{n} l_{ij}\tilde{W}_j, \tag{65}$$

respectively. Then with the same Lyapunov function used in Theorem 1, we apply equation (64) and (65) to (37), and the time derivative of the Lyapunov function is

$$
\begin{aligned}
\dot{V} &= \sum_{i=1}^{n}\left[\tilde{x}_i\dot{\tilde{x}}_i + \tilde{y}_i\dot{\tilde{y}}_i + \frac{\sin\tilde{\theta}_i}{K_y}\dot{\tilde{\theta}}_i + \tilde{\mathbf{u}}_i^T\bar{M}\dot{\tilde{\mathbf{u}}}_i + \frac{\mathrm{trace}(\tilde{W}_i^T\dot{\tilde{W}}_i)}{\Gamma}\right] \\
&= \sum_{i=1}^{n}\Big\{\tilde{x}_i(\tilde{v}_i + \omega_i\tilde{y}_i - K_x\tilde{x}_i) + \tilde{y}_i(v_{r_i}\sin\tilde{\theta}_i - \omega_i\tilde{x}_i) \\
&\quad + \frac{\sin\tilde{\theta}_i}{K_y}(\tilde{\omega}_i - v_{r_i}K_y\tilde{y}_i - K_\theta\sin\tilde{\theta}_i) \\
&\quad + \tilde{\mathbf{u}}_i^T\left[\tilde{W}_i^T S(X_i) + \epsilon_i - K_u(\tilde{\mathbf{u}}_i + \mathbf{z}_i) - \begin{bmatrix}\tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y}\end{bmatrix}\right] \\
&\quad + \mathrm{trace}\left(\tilde{W}_i^T\left[-S(X_i)(\tilde{\mathbf{u}}_i^T + \mathbf{z}_i^T) + \frac{\gamma\hat{W}_i}{\Gamma} - \frac{\beta}{\Gamma}\sum_{j=1}^{n} l_{ij}\tilde{W}_j\right]\right)\Big\} \qquad (66) \\
&= \sum_{i=1}^{n}\Big\{-K_x\tilde{x}_i^2 - \frac{K_\theta}{K_y}\sin^2\tilde{\theta}_i - K_u\tilde{\mathbf{u}}_i^T\tilde{\mathbf{u}}_i - K_u\tilde{\mathbf{u}}_i^T\mathbf{z}_i + \tilde{\mathbf{u}}_i^T\epsilon_i + \tilde{\mathbf{u}}_i^T[\tilde{W}_i^T S(X_i)] \\
&\quad - \mathrm{trace}\left([\tilde{W}_i^T S(X_i)]\tilde{\mathbf{u}}_i^T + [\tilde{W}_i^T S(X_i)]\mathbf{z}_i^T\right) + \mathrm{trace}\left(\frac{\gamma\tilde{W}_i^T\hat{W}_i}{\Gamma}\right)\Big\} \\
&\quad - \mathrm{trace}\left(\sum_{i=1}^{n}\frac{\beta}{\Gamma}\tilde{W}_i^T\sum_{j=1}^{n} l_{ij}\tilde{W}_j\right) \\
&= \sum_{i=1}^{n}\Big\{-K_x\tilde{x}_i^2 - \frac{K_\theta}{K_y}\sin^2\tilde{\theta}_i - K_u\tilde{\mathbf{u}}_i^T\tilde{\mathbf{u}}_i + \tilde{\mathbf{u}}_i^T\epsilon_i + \frac{\gamma}{\Gamma}\mathrm{trace}\left(\tilde{W}_i^T\hat{W}_i\right) \\
&\quad - \mathbf{z}_i^T\left[K_u\tilde{\mathbf{u}}_i + \tilde{W}_i^T S(X_i)\right]\Big\} - \frac{\beta}{\Gamma}\mathrm{trace}\left(\tilde{W}^T(L\otimes I)\tilde{W}\right).
\end{aligned}
$$

From Theorem 4, we have $\mathbf{z}_i$ converging to a small neighborhood around zero, $\forall i = 1, \cdots, n$, then following the same process in the proof of Theorem 1, we can conclude that the tracking error $\tilde{\mathbf{q}}_i$ for all agents will converge to a small neighborhood of zero, whose size depends on the norm of $\tilde{\mathbf{u}}_i^T\epsilon_i + \frac{\gamma}{\Gamma}\,\mathrm{trace}\left(\tilde{W}_i^T\hat{W}_i\right) - \mathbf{z}_i^T\left[K_u\tilde{\mathbf{u}}_i + \tilde{W}_i^T S(X_i)\right]$.

In addition, we can also show the NN weights of all vehicle agents converging to their optimal values by following a similar line of the proof of Theorem 2. By combining (64) and (65) for all vehicles, We have the matrix form:

$$\begin{bmatrix} \dot{\tilde{v}} \\ \dot{\tilde{\omega}} \\ \dot{\tilde{W}}_v \\ \dot{\tilde{W}}_\omega \end{bmatrix} = \begin{bmatrix} A & B \\ -C & -D \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \\ \tilde{W}_v \\ \tilde{W}_\omega \end{bmatrix} + E, \tag{67}$$

in which

$$\begin{aligned} A_{2n\times 2n} &= -K_u(\bar{M}^{-1} \otimes I_n), \\ B_{2nN\times 2n} &= (\bar{M}^{-1} \otimes I_n)(I_2 \otimes \mathbf{S}) = \bar{M}^{-1} \otimes \mathbf{S}^T, \\ C_{2n\times 2nN} &= \Gamma(I_2 \otimes \mathbf{S}), \\ D_{2nN\times 2nN} &= \beta I_2 \otimes (L \otimes I_N), \\ E_{(2nN+2nN)\times 1} &= \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix}, \\ E_1 &= \frac{1}{m}\begin{bmatrix} \epsilon_{v_1} - (v_1 - \hat{v}_1) - \tilde{x}_1 \\ \vdots \\ \epsilon_{v_n} - (v_n - \hat{v}_n) - \tilde{x}_n \end{bmatrix}, \\ E_2 &= \frac{1}{I}\begin{bmatrix} \epsilon_{\omega_1} - (\omega_1 - \hat{\omega}_1) - \frac{\sin\tilde{\theta}_1}{K_y} \\ \vdots \\ \epsilon_{\omega_n} - (\omega_n - \hat{\omega}_n) - \frac{\sin\tilde{\theta}_n}{K_y} \end{bmatrix}, \\ E_3 &= \begin{bmatrix} -(v_1 - \hat{v}_1) - \gamma\hat{W}_{1,1} \\ \vdots \\ -(v_n - \hat{v}_n) - \gamma\hat{W}_{n,1} \end{bmatrix}, \\ E_4 &= \begin{bmatrix} -(\omega_1 - \hat{\omega}_1) - \gamma\hat{W}_{1,2} \\ \vdots \\ -(\omega_n - \hat{\omega}_n) - \gamma\hat{W}_{n,2} \end{bmatrix}. \end{aligned}$$

As is shown in Theorem 4, the estimation error $(v_i - \hat{v}_i)$ and $(\omega_i - \hat{\omega}_i)$ converge quickly to a small neighborhood around zero, $\forall i = 1, \ldots, n$. In addition, it is shown in the tracking part of Theorem 5 that tracking errors $\tilde{v}_i$ and $\tilde{\omega}_i$ for all vehicles also converge to a small neighborhood around zero. Furthermore, the ideal approximation errors ϵ_{v_i} and ϵ_{ω_i} can be made arbitrarily small given sufficiently large number of RBF neurons, and γ is chosen to ba a small positive constant, therefore, we can conclude that the norm of E in (40) is a small value. Then $\tilde{v}$, $\tilde{\omega}$, $\tilde{W}_v$, and $\tilde{W}_\omega$ will converge to a small neighborhood of the origin, whose size depends on the value of $||E||$, if the nominal part of closed loop system shown in (40) is uniformly locally exponentially stable (ULES).

Similar to the proof of Theorem 2, we can easily reach the conclusion that under the assumptions 1, 2, and 3, $\tilde{v}$, $\tilde{\omega}$, $\tilde{W}_v$, and $\tilde{W}_\omega$ will converge to the neighborhood of the origin, whose size depends on the small value of $||E||$, i.e., the unmodeled vehicle dynamics can be locally accurately approximated by the NN weights of all vehicle agents along the union of the reference trajectories. □

4.3 High-gain Observer-based Trajectory Tracking with Experience

After showing the the convergence of NN weights to their common optimal value W^* alone the union of reference trajectories, we now further propose an experience-based output feedback controller based on the high-gain observer (53) and (55), using the learned knowledge presented by the time average of the NN weights given by equation (41). For the i^{th} vehicle, we choose the converged NN weight from a random vehicle agent, marked as j, and the experience-based controller is thereby constructed as

$$\bar{\tau}_i = \bar{W}_j^T S(\hat{X}_i) + K_u(\mathbf{u}_{c_i} - \hat{\mathbf{u}}_i) + \begin{bmatrix} \tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y} \end{bmatrix}, \tag{68}$$

Theorem 6. *Consider the closed-loop system including n unicycle-type vehicles described by equation (7) and (17), the desired reference trajectory $\mathbf{q}_{ri} \in \cup_{j=1}^{n}\mathbf{q}_j(t)$, high-gain observer (53) and (55), and the experience-based controller (68) with the virtual velocity (22), then for any bounded initial condition, the tracking error $\tilde{\mathbf{q}}_i$ converges asymptotically to a small neighborhood around zero.*

Proof. Similar to the proof of Theorem 5, we first derive the error dynamics of the velocity error $\mathbf{u}_i$ with the proposed experience-based controller (46) as

$$\begin{aligned}\dot{\tilde{\mathbf{u}}}_i &= \bar{M}^{-1}(\bar{M}\dot{\mathbf{u}}_{c_i} + \bar{C}\mathbf{u}_i + \bar{F} - \bar{\tau}_i) \\ &= \bar{M}^{-1}\left\{\epsilon_{2_j} - K_u(\tilde{\mathbf{u}}_i + \mathbf{z}_i) - \begin{bmatrix}\tilde{x}_i \\ \frac{\sin\tilde{\theta}_i}{K_y}\end{bmatrix}\right\}\end{aligned} \tag{69}$$

where $\epsilon_{2_j} = H(X_j) - \bar{W}_j^T S(X_j)$. Similarly, we can build a positive definite function of this closed-loop system of the i^{th} vehicle as

$$V_i = \frac{\tilde{x}_i^2}{2} + \frac{\tilde{y}_i^2}{2} + \frac{1 - \cos\tilde{\theta}_i}{K_y} + \frac{\tilde{\mathbf{u}}_i^T \bar{M}\tilde{\mathbf{u}}_i}{2} \tag{70}$$

whose derivative is

$$\begin{aligned}\dot{V}_i &= \tilde{x}_i\dot{\tilde{x}}_i + \tilde{y}_i\dot{\tilde{y}}_i + \frac{\sin\tilde{\theta}_i}{K_y}\dot{\tilde{\theta}}_i + \tilde{\mathbf{u}}_i^T \bar{M}\dot{\tilde{\mathbf{u}}}_i \\ &= -K_x\tilde{x}_i^2 - \frac{K_\theta}{K_y}\sin^2\tilde{\theta}_i - K_u\tilde{\mathbf{u}}_i^T\tilde{\mathbf{u}}_i + \tilde{\mathbf{u}}_i^T(\epsilon_{2_j} - K_u\mathbf{z}_i)\end{aligned} \tag{71}$$

Then following the similar arguments in the proof of Theorem 5, given positive K_x, K_y, K_θ, and K_u, we can conclude that the Lyapunov function V_i is positive definite and $\dot{V}_i$ is negative semi-definite in the region $K_x\tilde{x}_i^2 + \frac{K_\theta}{K_y}\sin^2\tilde{\theta}_i + K_u\tilde{\mathbf{u}}_i^T\tilde{\mathbf{u}}_i \geq \tilde{\mathbf{u}}_i^T(\epsilon_{2_j} - K_u\mathbf{z}_i)$. Since the approximation error ϵ_{2_j} and the observer estimation error $\mathbf{z}_i$ are both close to zero, then follow the proof of Theorem 3, it can be shown that the tracking errors will converge to a small neighborhood around zero for any vehicle agent in the MAS, given any learned experience $\bar{W}_j$ obtained from the learning phase. □

4.4 Simulation Study (MATLAB Results)

In this section, we conduct the MATLAB simulation following the similar setup in Chapter 3, i.e., using four unicycle-type vehicle with the same parameters given in Table 1, and following the same reference trajectories (50). Parameters for the controller and weight updating law using output feedback remain unchanged from the state feedback case as shown in Table 2. The RBFNN setup also remains unchanged from Chapter 3, while the parameters for the high-gain observer is given in Table 3.

Table 3: Parameters of the high-gain observer

observer parameter	value
l_1	1
l_2	2
δ	0.01

Using the same `ode45` function in MATLAB, we perform the simulation on solving a group of ODEs, including the vehicle kinematics (7), dynamics (17), high-gain observer (53) and (55), as well as the weight updating law (32), on the time period from 0 to 300 seconds. The initial conditions of the ODEs are the same as those in Chapter 3, with the initial state of the observer set to zero. Simulation results are shown as follows.

Figure 10 shows that all vehicles (blue triangles) will track their own reference trajectories (red solid circles), and Figure 11 shows that all tracking errors $\tilde{x}_i$ and $\tilde{y}_i$ will converge to zero, within 30 seconds of the simulation in the learning phase. Figure 12 shows that the observer error will converge to a close neighborhood around zero in a very short time period. Figure 14 shows that the norms of the NN weights $\hat{W}_i$ for all vehicle agents converge to the same constant, and Figure 13 shows that the approximation errors of all vehicle agents converge to a small neighborhood around zero, suggesting that all vehicle agents in the MAS are able to accurately approximate the unstructured system uncertainties $\bar{M}\dot{\mathbf{u}}_{c_i} + \bar{C}_i\mathbf{u}_i + \bar{F}_i$ along the union of trajectories.

After the learning phase shown above, we calculate and store the average of NN weights over the last 3 seconds using equation (41). With this learned knowledge, we also show the tracking convergence using the experience-based controller (68).

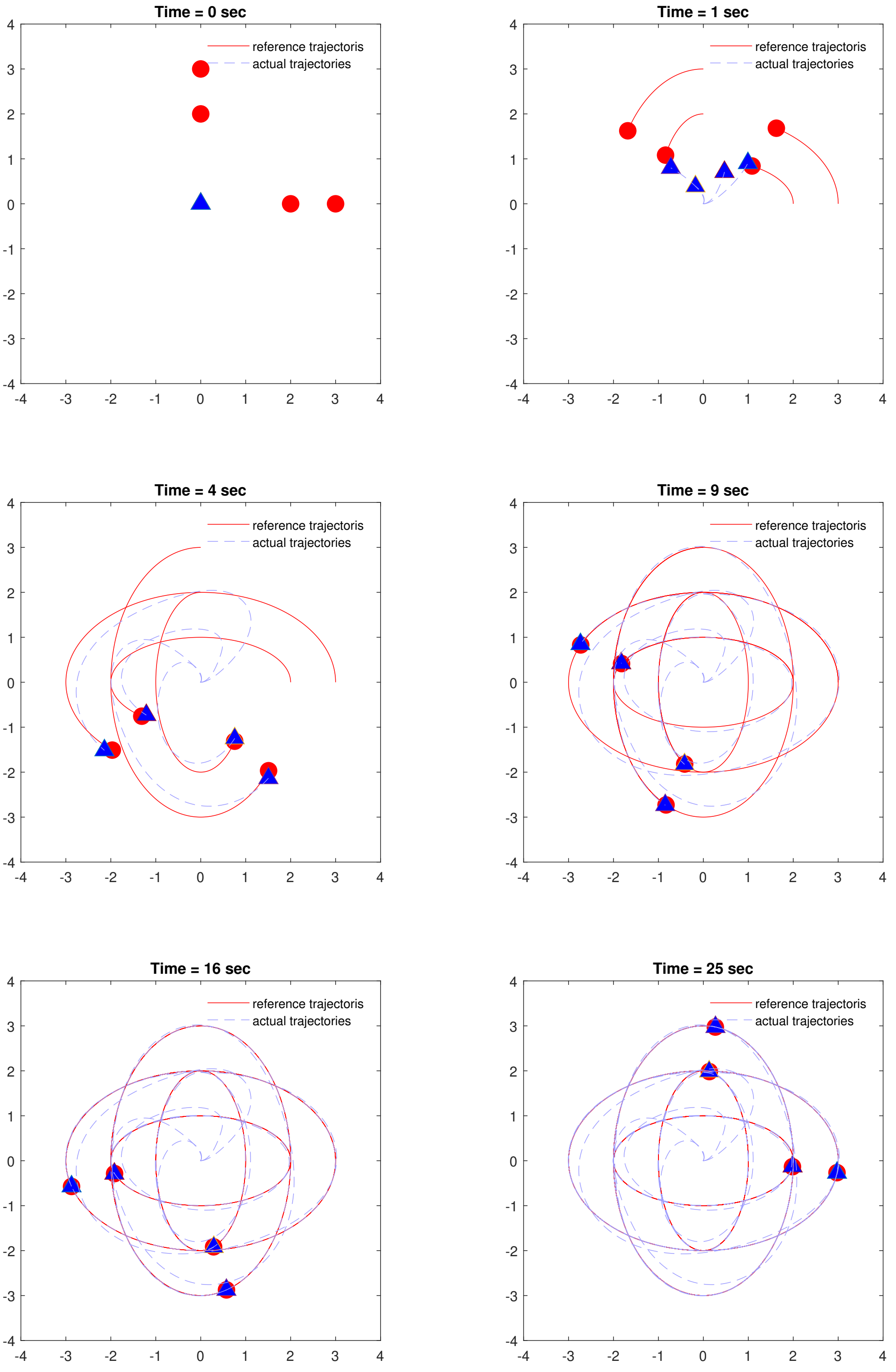


Figure 10: Snapshot of trajectory tracking using CALC via output feedback.

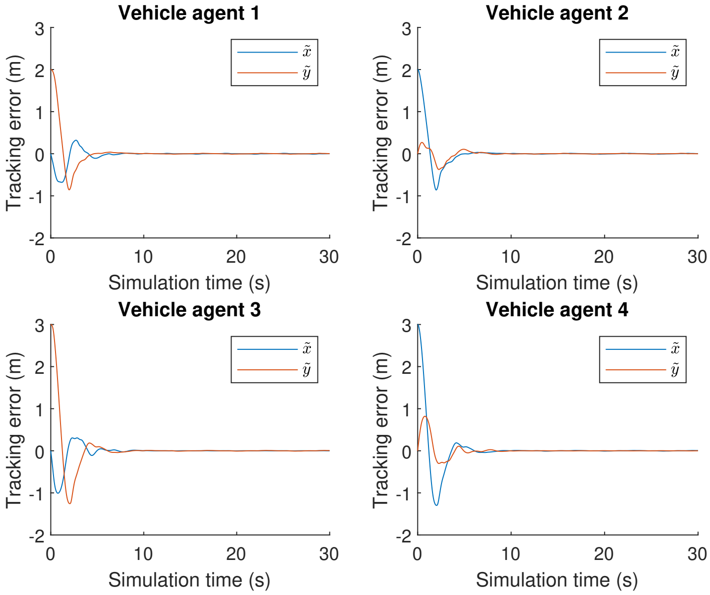


Figure 11: Tracking errors using CALC via output feedback.

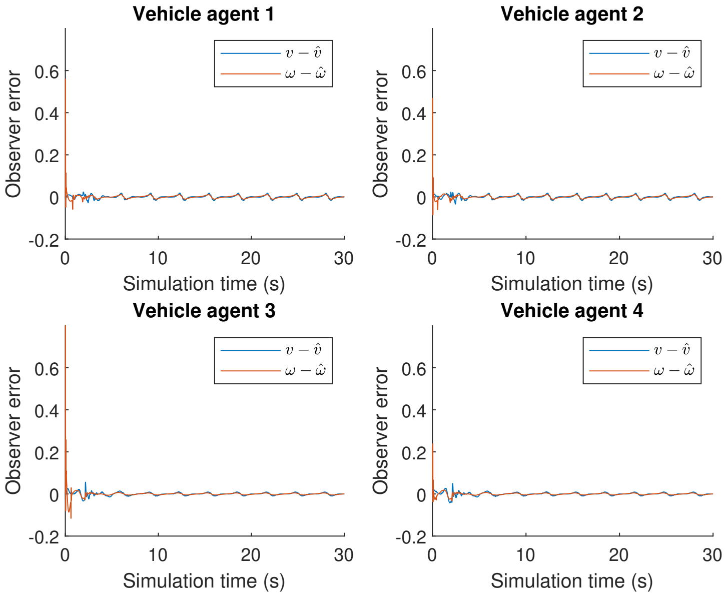


Figure 12: Observer error with CALC via output feedback.

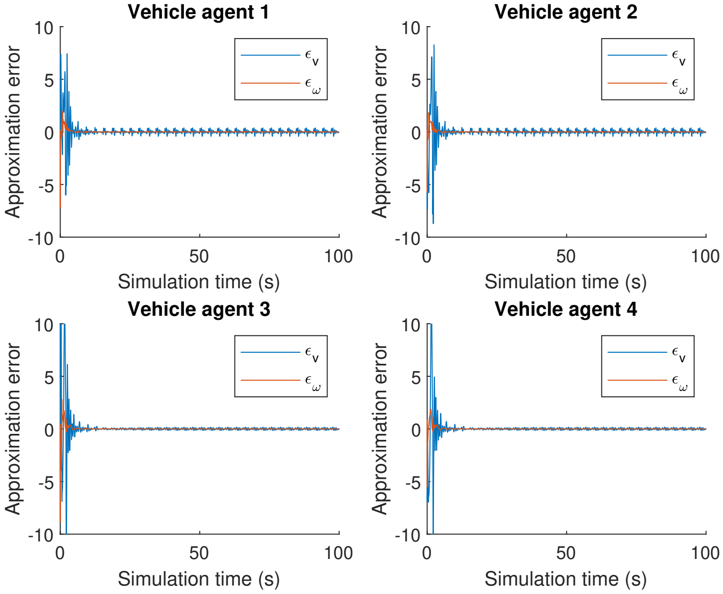


Figure 13: Approximation errors using CALC via output feedback.

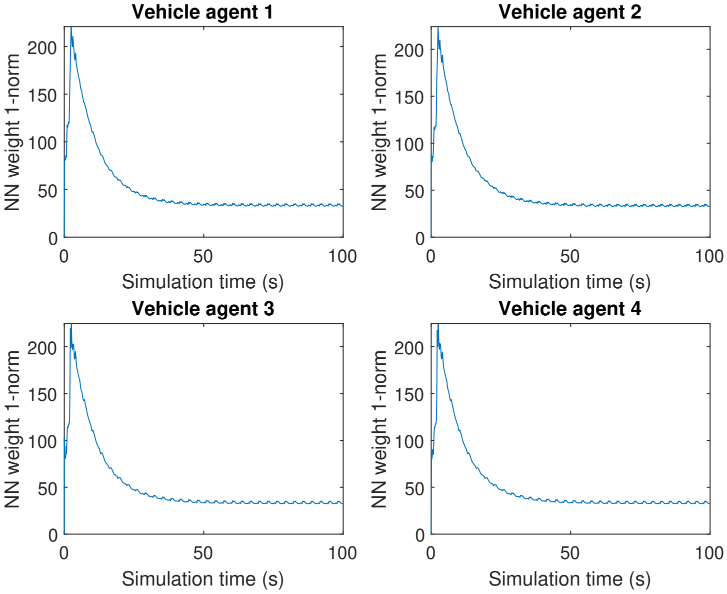
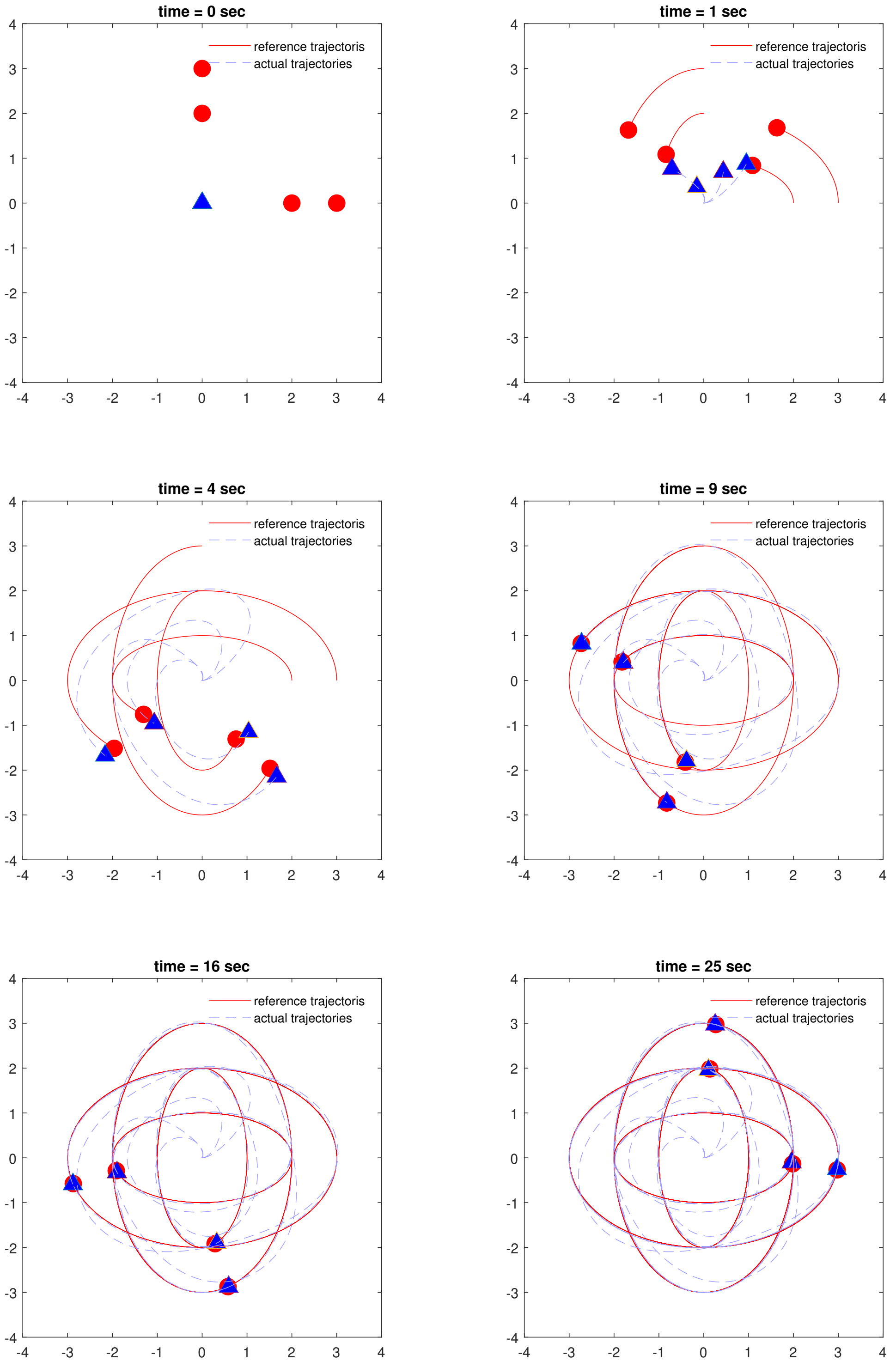


Figure 14: Weight vector 1-norm of $\hat{W}_i$


time = 0 sec

time = 1 sec

time = 4 sec

time = 9 sec

time = 16 sec

time = 25 sec

Figure 15: Snapshot of trajectory tracking using experience-based controller via output feedback.

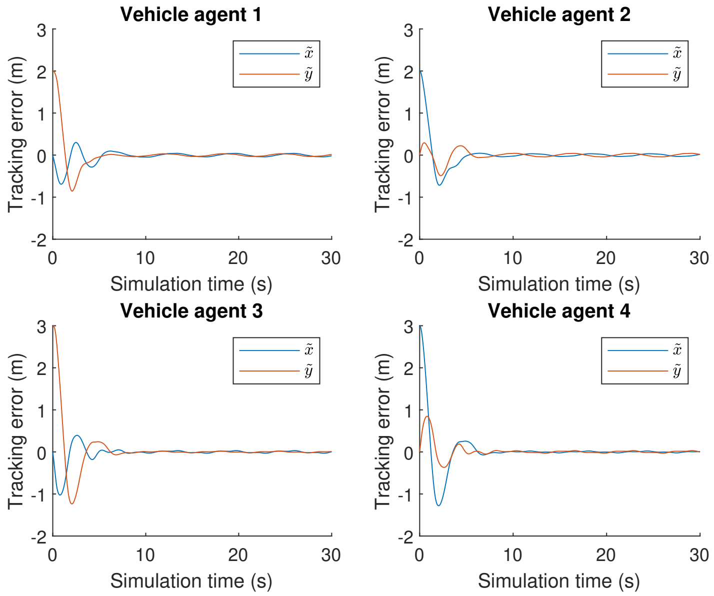


Figure 16: Tracking errors using experience-based controller via output feedback.

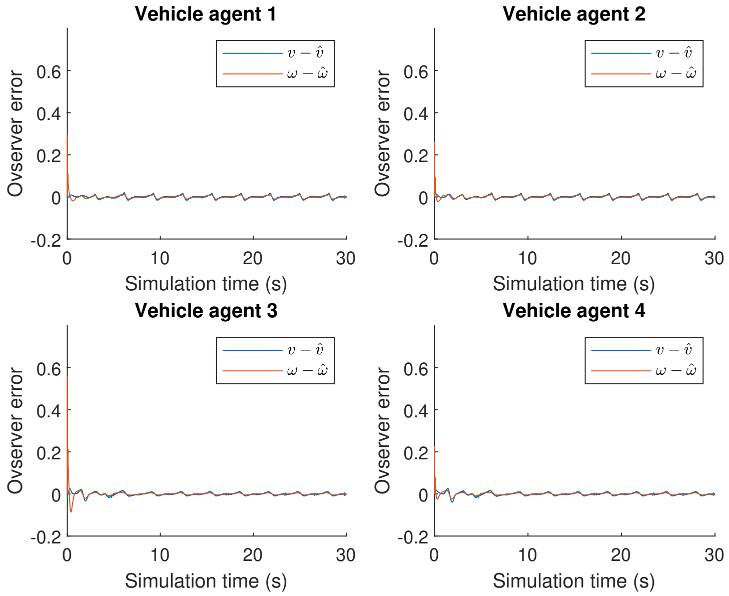


Figure 17: Observer error with experience-based controller via output feedback.

Figures 15 and 16 show that the tracking errors converge to zero at a similar rate as the CDL-based controller did, with all vehicle agents using the learned knowledge $\bar{W}$ randomly selected from vehicle agent 4. Figure 17 shows that the observer error will converge to a close neighborhood around zero in a very short time period for the experience-based control simulation.

4.5 Summary

In this chapter, we designed a high-gain observer to estimate the generalized velocities of the vehicles using measured generalized coordinates, and modified the controllers proposed in the previous chapter for the output feedback control problems. Theoretical analysis has been provided to show that the adaptive tracking convergence using CALC and learned experience, as well as learning consensus and accurate approximation achieved by the state feedback controller shown in Chapter 3, can still be achieved by the controller proposed in this chapter via output feedback. MATLAB simulation results show that the objectives raised in section 4.1 are reached by the proposed output feedback controllers. Simulation results also show effectiveness of the proposed output-feedback controllers.

List of References

[1] P. Waldteufel and H. Corbin, "On the analysis of single-doppler radar data," *Journal of Applied Meteorology*, vol. 18, no. 4, pp. 532–542, 1979.

[2] H. K. Khalil, "High-gain observers in nonlinear feedback control," in *2008 International Conference on Control, Automation and Systems.* IEEE, 2008, pp. xlvii–lvii.

[3] W. Zeng, Q. Wang, F. Liu, and Y. Wang, "Learning from adaptive neural network output feedback control of a unicycle-type mobile robot," *ISA transactions*, vol. 61, pp. 337–347, 2016.

[4] C. D. Meyer, *Matrix analysis and applied linear algebra.* Siam, 2000, vol. 71.

[5] K. W. Lee and H. K. Khalil, "Adaptive output feedback control of robot manipulators using high-gain observer," *International Journal of Control*, vol. 67, no. 6, pp. 869–886, 1997.

[6] S. Oh and H. K. Khalil, “Nonlinear output-feedback tracking using high-gain observer and variable structure control,” *Automatica*, vol. 33, no. 10, pp. 1845–1856, 1997.

CHAPTER 5

Gazebo simulator development and validation

5.1 Problem Statement

In Chapters 3 and 4, we performed simulations on the proposed controllers using MATLAB. To simulate the state variables of the MAS, a group of ODEs, including the vehicle model and the control algorithms, are solved using the `ode45` function. To this end, the physical model of the unicycle-type vehicle is translated into the differential equations (8) and (17), under the assumption that no relative slipping occurs between the wheels and the ground plane, which can be presented by the nonholonomic constraint (5). Although these ODEs for the vehicles satisfy the non-slippery assumptions, it may not be applicable for actual vehicles, considering the thickness of the wheels and the maximum friction that can be applied on the vehicle.

To analyze capability of the proposed controllers for the trajectory tracking tasks on real vehicles, a simulation tool that is more powerful and closer to real world is required. The Gazebo simulator [1] is one of the most widely used simulators in robotics for designing, testing, and AI training. Based on the Open Dynamics Engine [2], Gazebo is able to simulate the dynamics and sensoring of robot models, as well as the interaction between models and the environment. Models simulated in the Gazebo can be defined using Unified Robot Description Format (URDF) [3] or Simulation Description Format (SDF) [4]. Properties of the model, including but not limited to, dimension, inertia, friction, and stiffness, can be defined in the `.urdf` or `.sdf` of the model to make the model closer to the actual robot in the real world. Control inputs of the model, such as the torque applied on the joints, can be specified by either directly defining them in the simulator or through a Gazebo plugin. Model states during the simulation can be viewed in the Gazebo's graphical user interface (GUI), or exported through its application programming interface (API).

For better organizing the simulation, from launching Gazebo to recording data, we use the Robot Operating System (ROS) [5] on Linux. ROS is a set of software libraries and tools that help researchers and developers to build robotic applications [5], on which information from different programming languages/applications can be shared. `roslaunch` can be used to launch vehicle models into Gazebo and create ROS nodes, which can be treated as publishers and subscribers for information. `rostopic` is a package to transfer and display messages between ROS nodes. `rosbag` is a tool for recording ROS topics. Other applications of ROS can be found on the ROS documentation page [6].

5.2 Simulation Design

To demonstrate the effectiveness of the proposed controllers on models closer to the real world, we performe a simulation using the Gazebo simulator in this Chapter. Specifically, a group of unicycle-type ground vehicle models are built in the Gazebo, with a C++ Gazebo plug-in receiving control inputs and applying driving torque on the joints of the wheels. The control algorithm, both learning-based and experience-based, are coded in Python. Robot Operating System (ROS) is used as a platform to coordinate the Gazebo simulator and the Python controller: vehicle models are loaded into Gazebo using the `roslaunch` command with a `.launch` file, vehicle states and control inputs are transferred between the simulator and controller through `rostopic`, and simulation data are recorded into a `.bag` file through the `rosbag` command. The structure and the data flow of the simulation is shown in Figure 18.

As is shown in Figure 19, the unicycle-type ground vehicle model contains a body, a front caster, and two actuated wheels. The ball-shaped front caster

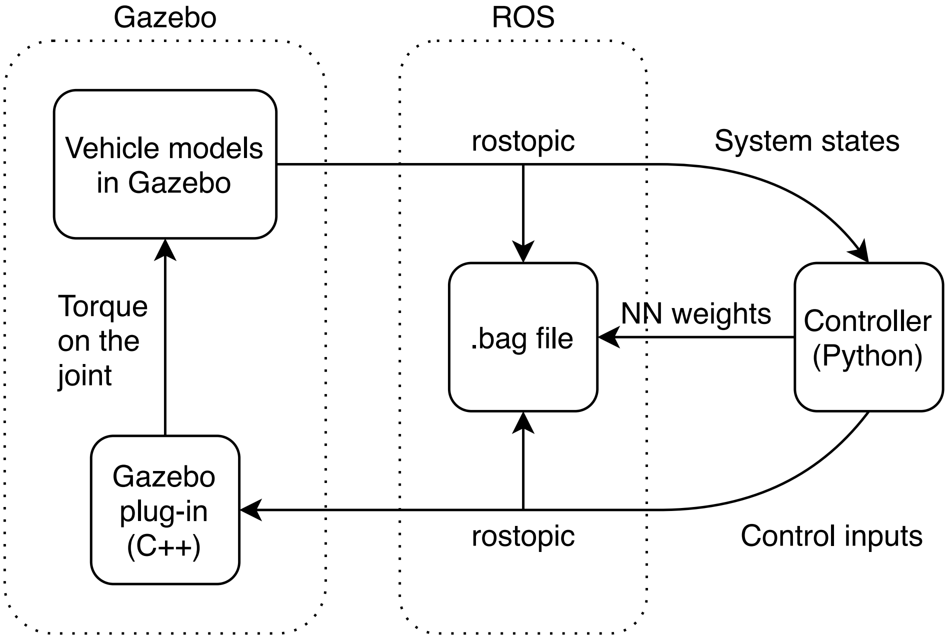


Figure 18: Structure and data flow of the simulation

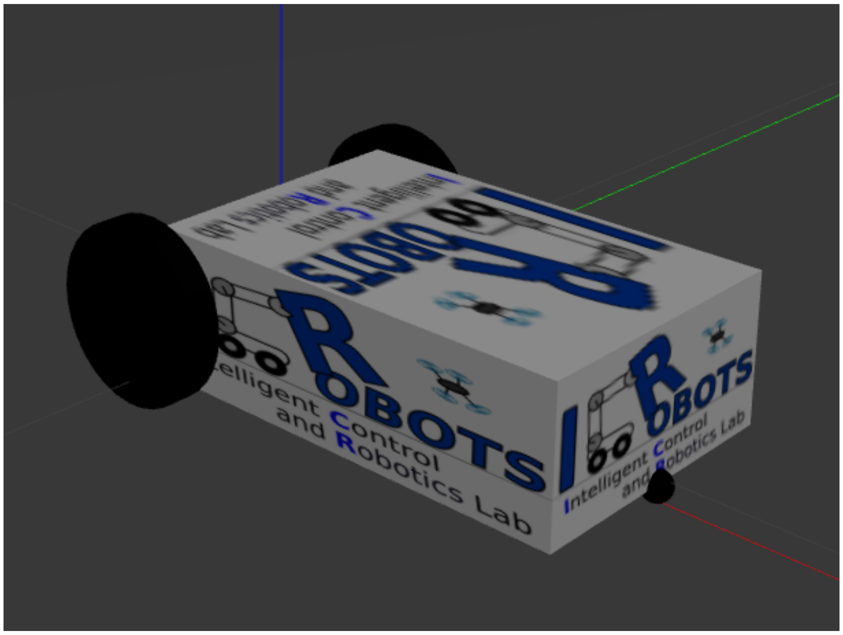


Figure 19: Vehicle model in the Gazebo simulator

is mounted at center front of the body's bottom, and two actuated wheels are connected to the body through driving joints located at the center rear of the body's sides. The parameters of the vehicle are listed in table 4.

Table 4: Parameters of the vehicle model in Gazebo

	distance from the reference point	shape		mass	friction coefficient
body	—	l	0.3 m	1 kg	—
		w	0.2 m		
		h	0.1 m		
front caster	0.3 m	r	0.01 m	0.1 kg	0.2
actuated wheels	R = 0.11 m	r	0.06 m	0.1 kg	1
		h	0.02 m		

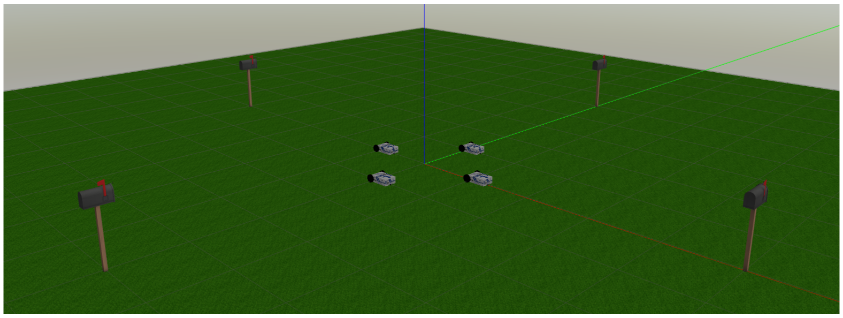


Figure 20: Testing field in the Gazebo simulator

The testing field in Gazebo is shown in Figure 20. The initial positions of vehicles are 1 m from the origin, along (x, y) axis of the test field. After the simulation starts, all vehicles will track the following reference trajectories

$$\begin{cases} x_{r_1} = 5 - \sin t \\ y_{r_1} = 2\cos t \end{cases} \begin{cases} x_{r_2} = 2\cos t \\ y_{r_2} = 5 + \sin t \end{cases} \begin{cases} x_{r_3} = -5 - 2\sin t \\ y_{r_3} = 3\cos t \end{cases} \begin{cases} x_{r_4} = 3\cos t \\ y_{r_4} = -5 + 2\sin t \end{cases}$$

Compared to the reference trajectories used for the MATLAB simulation, the centers of the trajectories, marked by four mailboxes, are moved from the center of the ground frame to avoid collisions between vehicles.

Different from the continuous-time simulation run on MATLAB in previous chapters, the Python control algorithms we have for the Gazebo simulator is designed for discrete-time control, which is more common for control tasks in the real world. To this end, we transform the weight updating law (32) from the differential equation into the following iteration form using the zero-order hold approach

$$\hat{W}_i[k+1] = \hat{W}_i[k] + \left[\Gamma S(X_i)\tilde{\mathbf{u}}_i^T - \gamma \hat{W}_i - \beta \sum_{j=1, j\neq i}^{n} a_{ij}(\hat{W}_i - \hat{W}_j) \right] \Delta t, \tag{72}$$

where Δt is the time interval for each iteration step. Apart from this change, other parts of the control algorithm, i.e., the cooperative adaptive-based controller (31) and the experience-based controller (46), remain unchanged. The parameters of the cooperative adaptive-based controller (31) with (22) and weight updating law (72), as well as the experience-based controller (46), are given by Table 5

Table 5: Parameters of controllers and discrete-time weight updating law

controller parameter	value
K_x	1
K_y	1
K_θ	0.1
K_u	2
Γ	0.1
γ	0.005
β	0.01

For each $i = 1, 2, 3, 4$, we normalize $X_i = \begin{bmatrix} \dot{\mathbf{u}}_{c_i}^T & \mathbf{u}_i^T \end{bmatrix}^T \in \mathbb{R}^{4\times 1}$ from $[-4, 4] \times [-4, 4] \times [0, 4] \times [0, 4]$ to the space $[-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1]$, then we construct the Gaussian RBFNN $\hat{W}_i S(X_i)$ using $N = 5 \times 5 \times 5 \times 5 = 625$ neuron nodes with the centers evenly placed over that space and the respective field η of the Gaussian function equal to 0.5.

The connection between vehicle agents remains unchanged from the simulations in previous chapters, which can be seen in Figure 3, and the Laplacian matrix

L associated with the graph $\mathcal{G}$ is still given by

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

After the NN weight converged in the learning-based control simulation, the average value of the NN weight over the last 3 seconds is stored for the experience-based control simulation. Following the experience-based control method shown in the previous section, we randomly choose the average weight $\bar{W}$ from one of the vehicle agents and apply it on all vehicles to track their reference trajectories. The parameters of the experience-based controller are the same as those in the CALC.

5.3 Simulation Results and Discussions

Simulation results are shown as follows.

Figure 21 shows that tracking errors for all vehicles converge to a small neighborhood of zero, and Figure 22 shows that the norms of the NN weights $\hat{W}_i$ for all vehicle agents converge as the same constant did, suggesting that the consensus of NN weights is reached for all vehicle agents in the MAS.

Figure 23 to 26 show that the tracking error for all vehicles converge to zero at a similar rate to the CDL-based controller, with all vehicle agents using the learned knowledge $\bar{W}$ from vehicle agent 1 to 4, respectively. Therefore, we can conclude that both learning-based and experience-based controllers are able to drive the vehicles to their reference trajectories, and the NN weights for all vehicle agents reach consensus with the discrete-time weight updating law (72). The approximation accuracy cannot be directly examined since the exact model of $\bar{C}(\mathbf{u}_i)\mathbf{u}_i + \bar{F}(\mathbf{u}_i)$ is unknown, however, we can still conclude that the NN weight $\hat{W}_i$ of any vehicle agent in the MAS converges to the common ideal value W^* along the union of reference trajectories, because the tracking error will converge to zero only if the RBFNN approximation $\bar{W}S(X)$ is accurate.

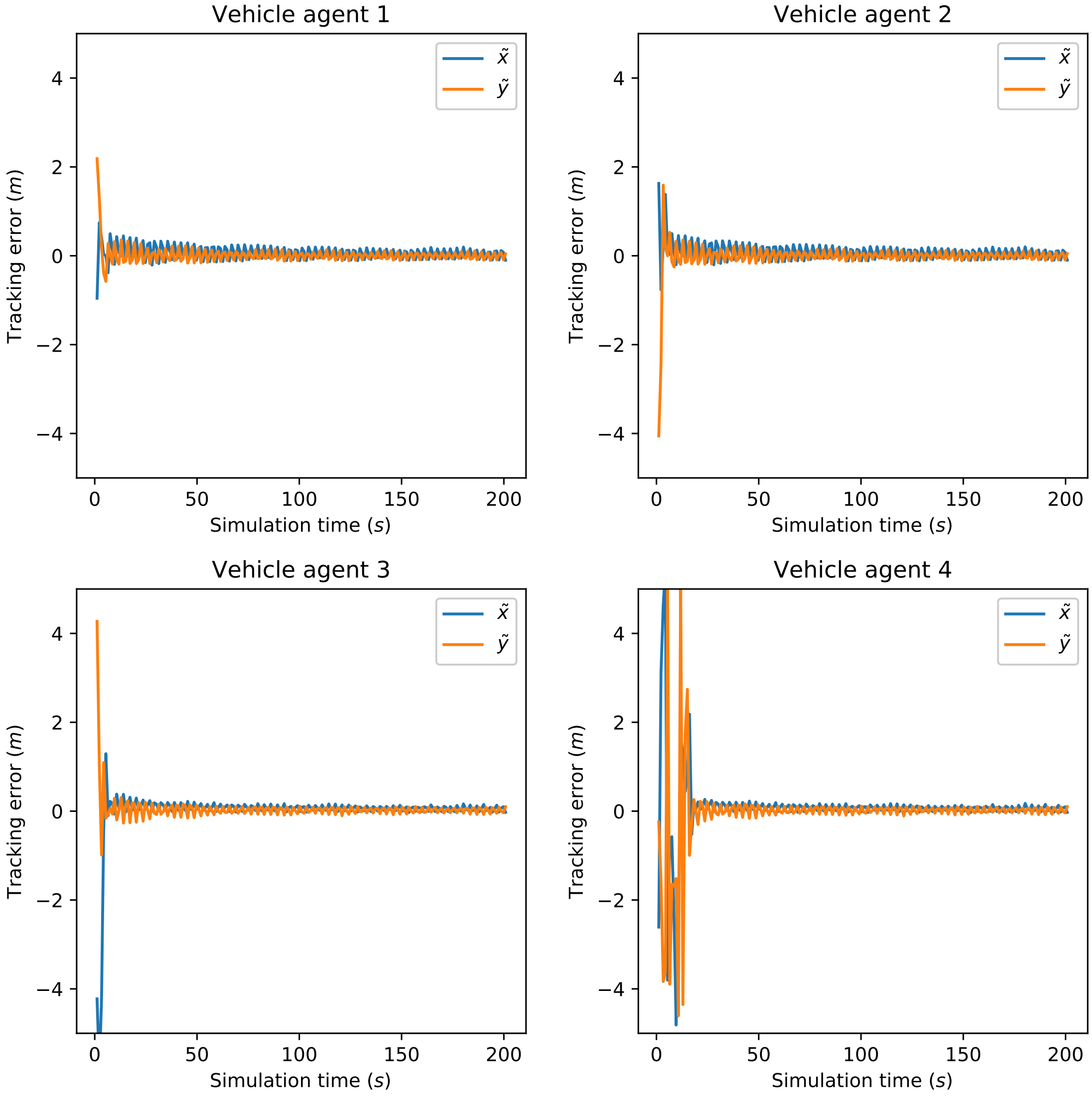


Figure 21: Tracking errors using CALC with discrete-time weight updating law.

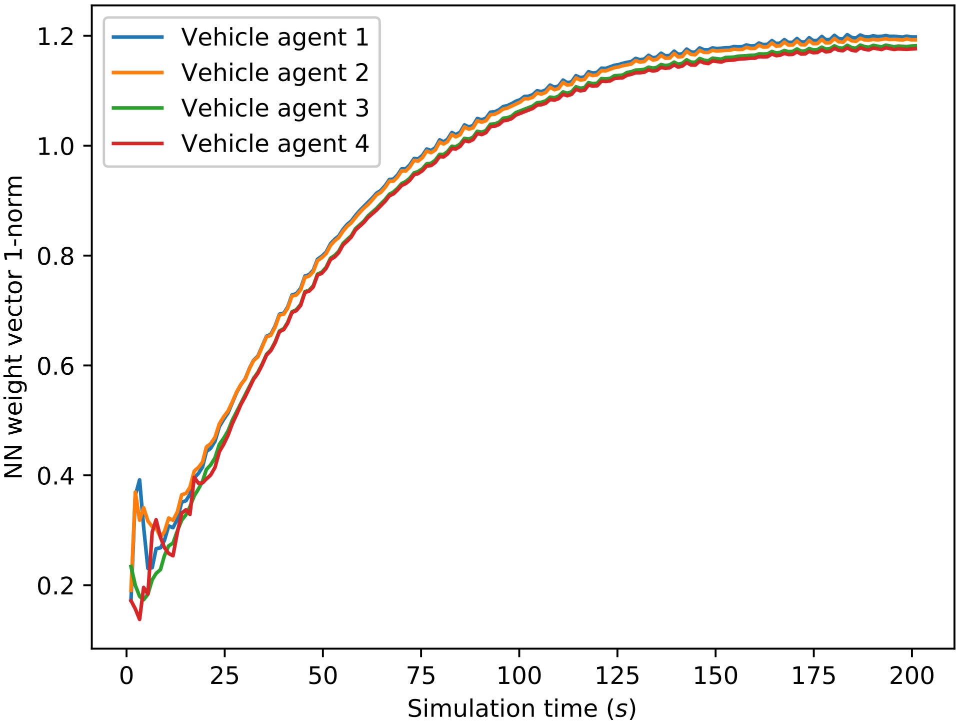


Figure 22: Weight vector 1-norm of $\hat{W}_i$.

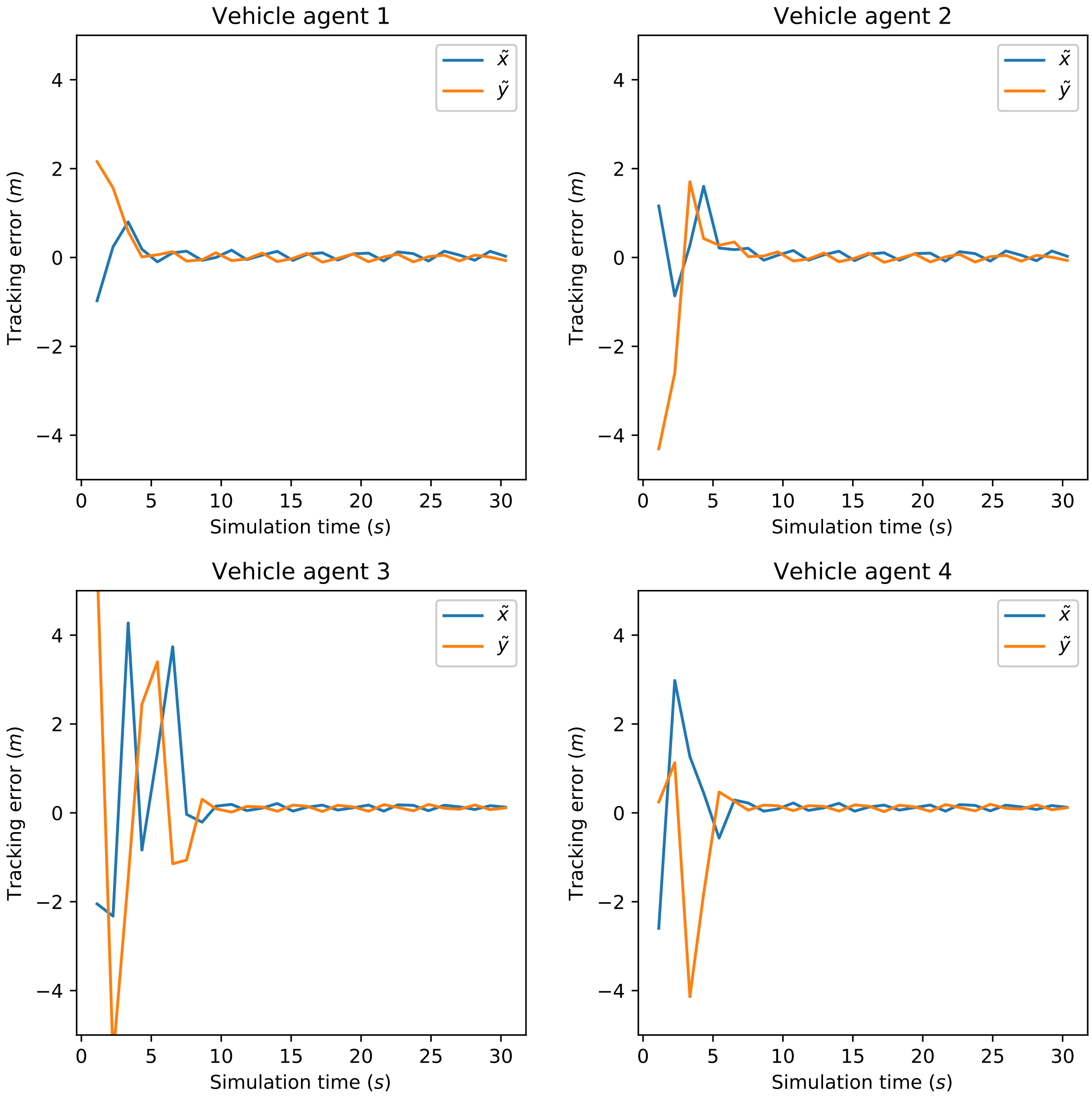


Figure 23: Tracking errors using experience-based controller with NN weight from vehicle agent 1.

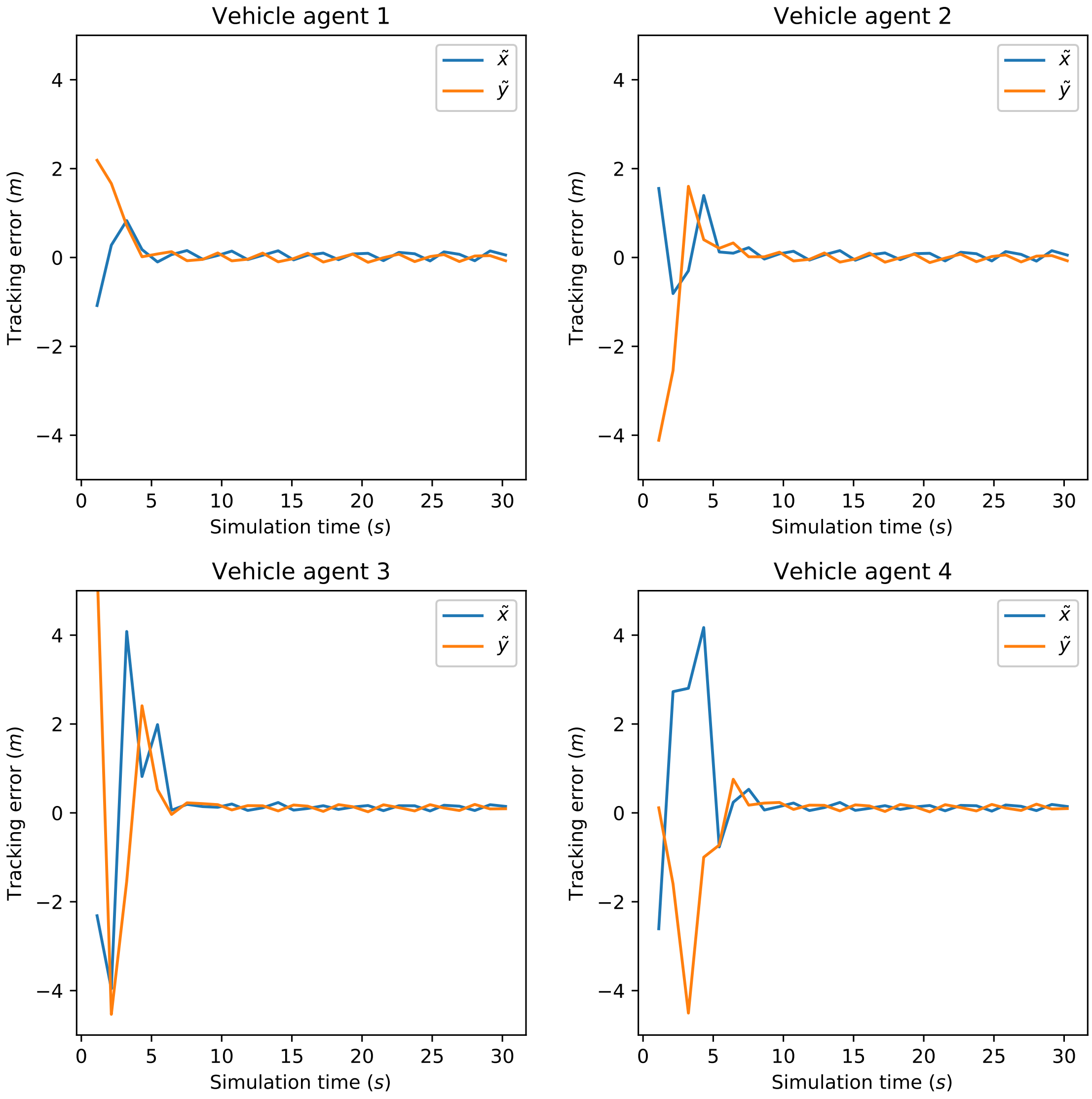


Figure 24: Tracking errors using experience-based controller with NN weight from vehicle agent 2.

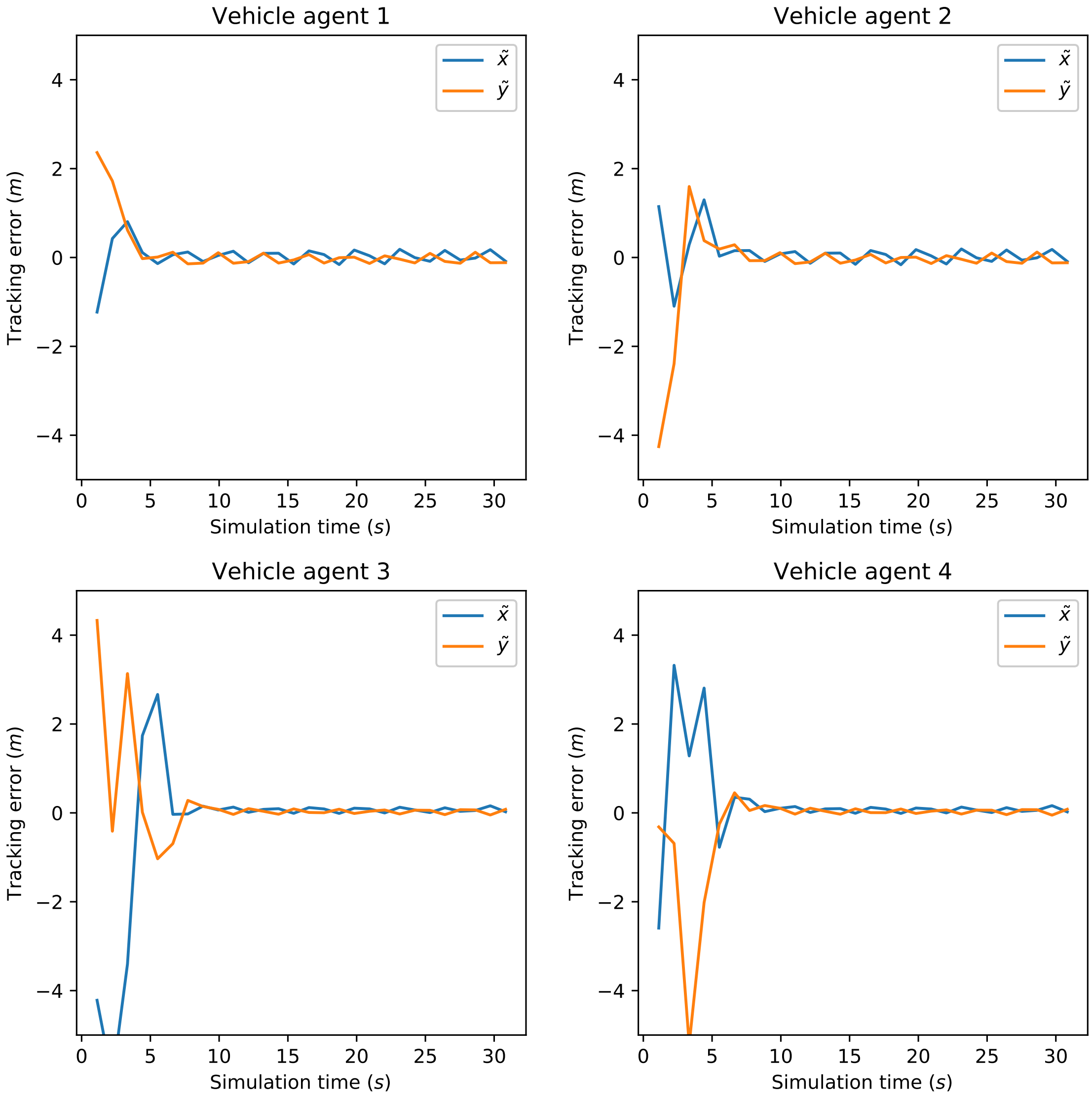


Figure 25: Tracking errors using experience-based controller with NN weight from vehicle agent 3.

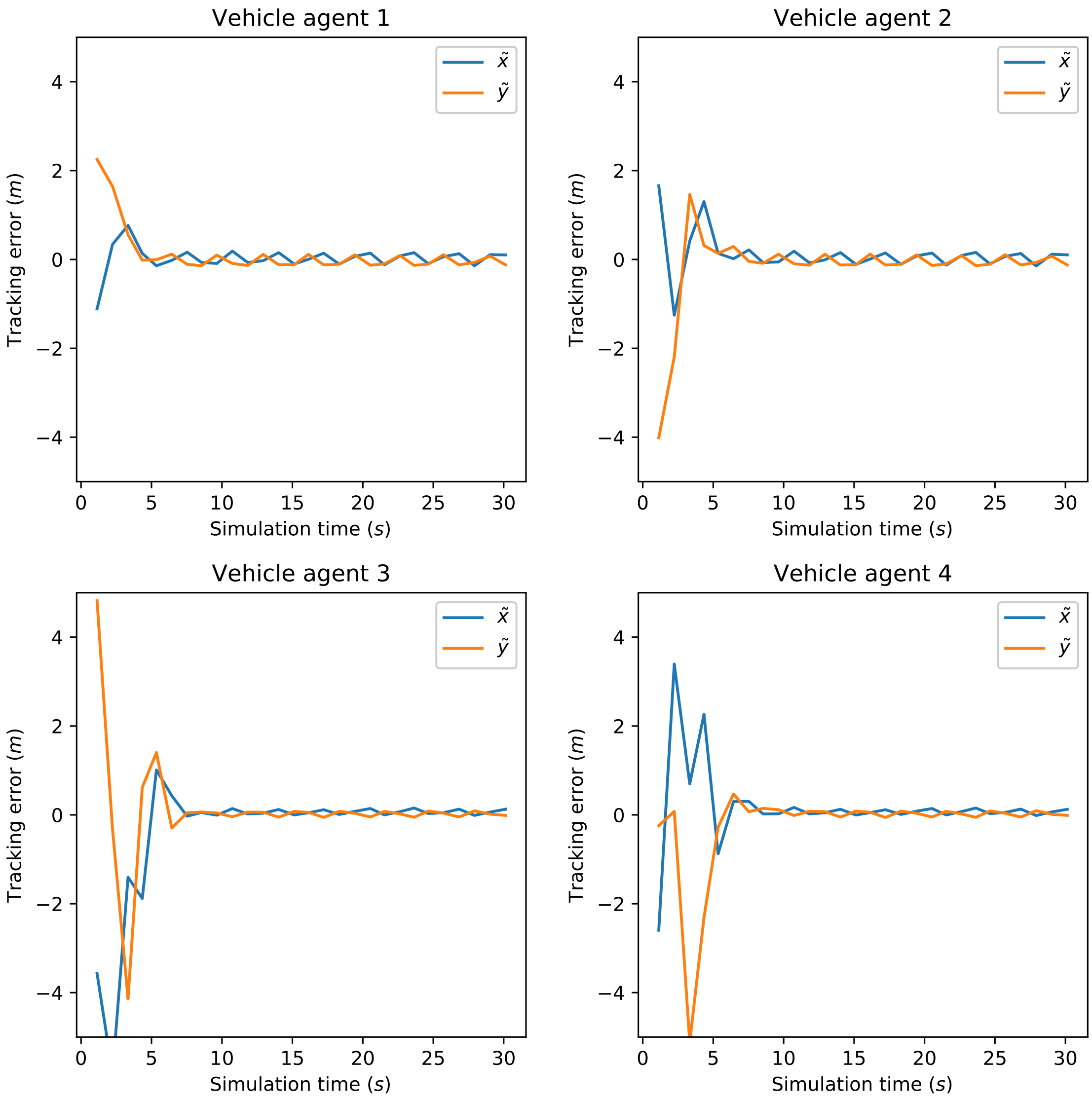


Figure 26: Tracking errors using experience-based controller with NN weight from vehicle agent 4.

To show the advantages of learning consensus with the proposed CALC, we also preformed a comparing simulation using traditional DL-based controller designed for single agent systems, with the same vehicles running same trajectories used in the CALC simulation. The discrete-time weight updating law is shown in the following iteration form

$$\hat{W}_i[k+1] = \hat{W}_i[k] + \left[\Gamma S(X_i)\tilde{\mathbf{u}}_i^T - \gamma \hat{W}_i\right] \Delta t. \tag{73}$$

Compared to our cooperative weight updating law (72), the traditional DL weight updating law does not have the communication term $\beta \sum_{j=1, j\neq i}^{n} a_{ij}(\hat{W}_i - \hat{W}_j)$, which only guarantees the accurate approximation along the trajectory experienced by the vehicle itself. The learning-based and experience-based controller, as well as the controller parameters, remain unchanged in this comparing simulation.

Figure 27 shows that tracking errors for all vehicles converge to a small neighborhood of zero, while Figure 28 shows that the norm of the NN weights $\hat{W}_i$ for all vehicle agents do not converge to the same constant, suggesting that the RBFNN weight for a specific vehicle agent will only locally represent the unmodeled dynamics along the trajectory experienced by itself.

With all vehicles running on the NN weight taken from vehicle agent 4 using traditional DL algorithm without communication, it is shown in Figure 29 that the tracking errors for vehicle agents 1 and 2 are much larger than those of vehicle agents 3 and 4, suggesting that the NN weight taken from vehicle agent 4 does not accurately approximate the unmodeled dynamics along the trajectories of agents 1 and 2, when communication among vehicles is not available. Compared to the results shown in Figures 23 to 26, it is clear that the proposed CALC improved the local approximation capability of RBFNNs over traditional DL algorithms by reaching learning consensus and accurately approximating the unstructured system uncertainties along the union trajectories experienced by all vehicle agents in the

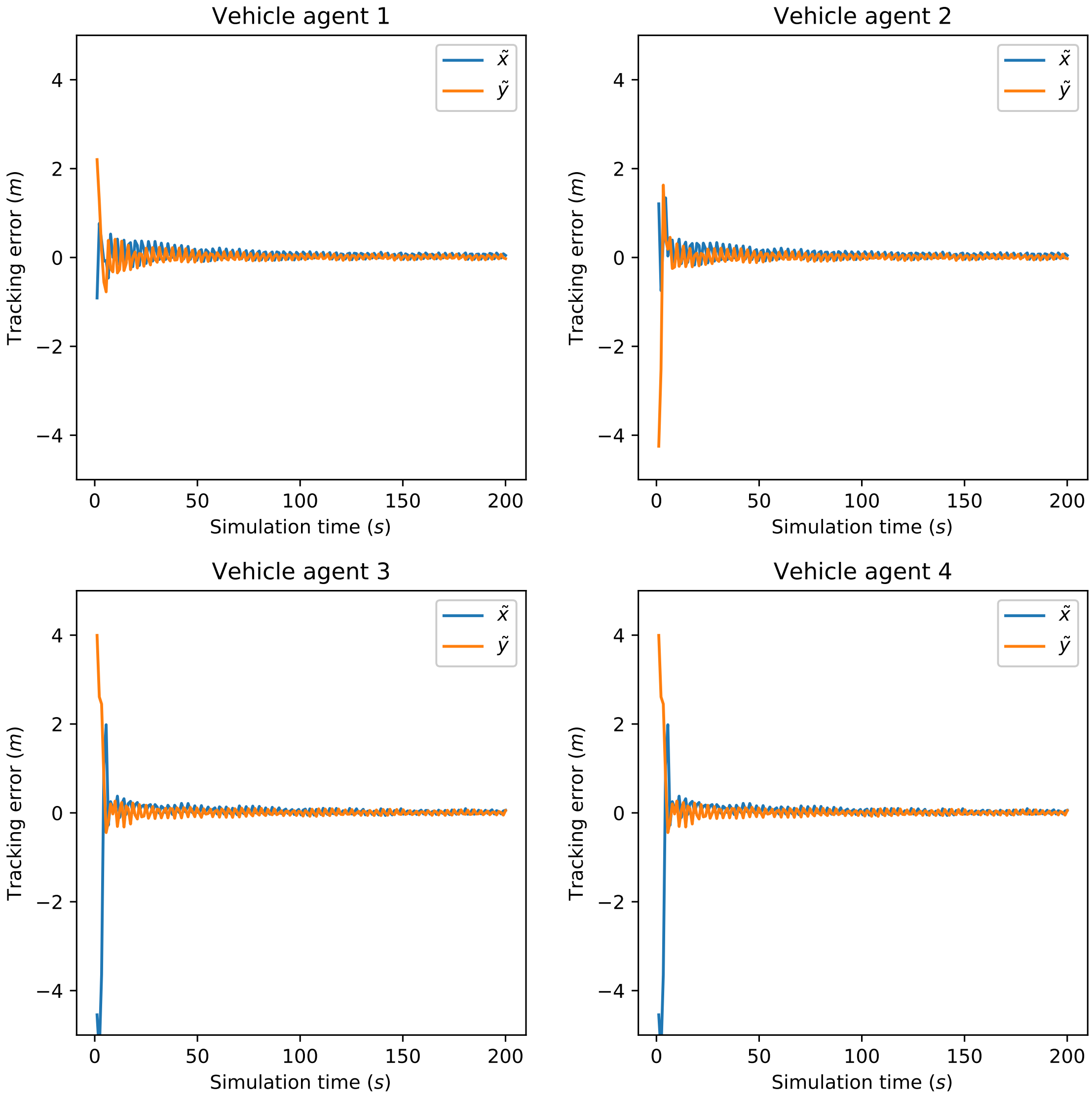


Figure 27: Tracking errors using traditional DL-based controller without communication.

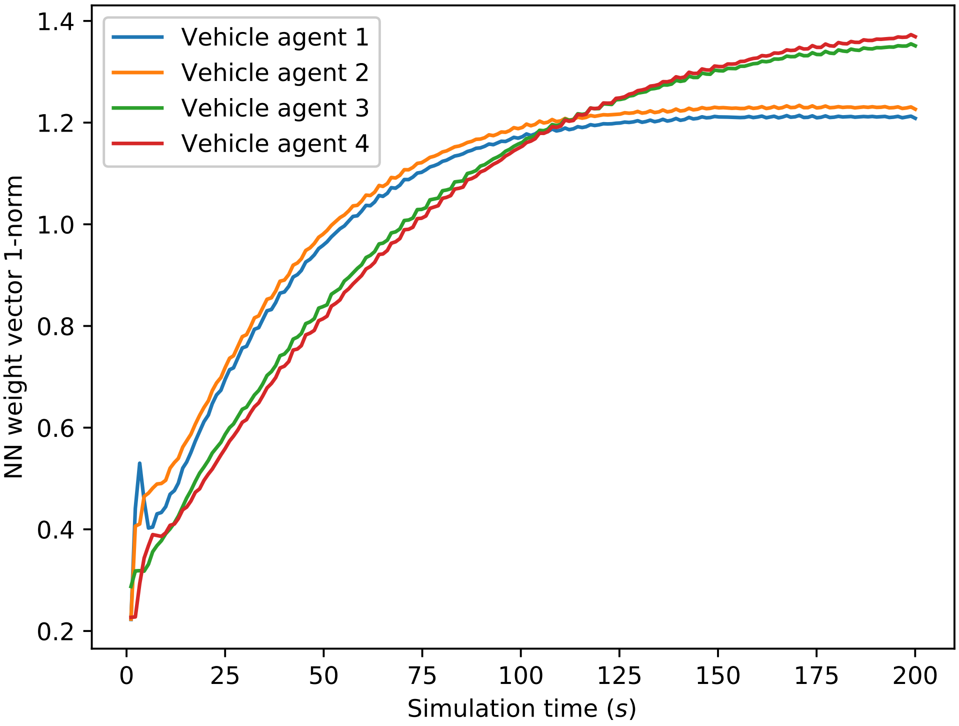


Figure 28: Weight vector 1-norm of $\hat{W}_i$.

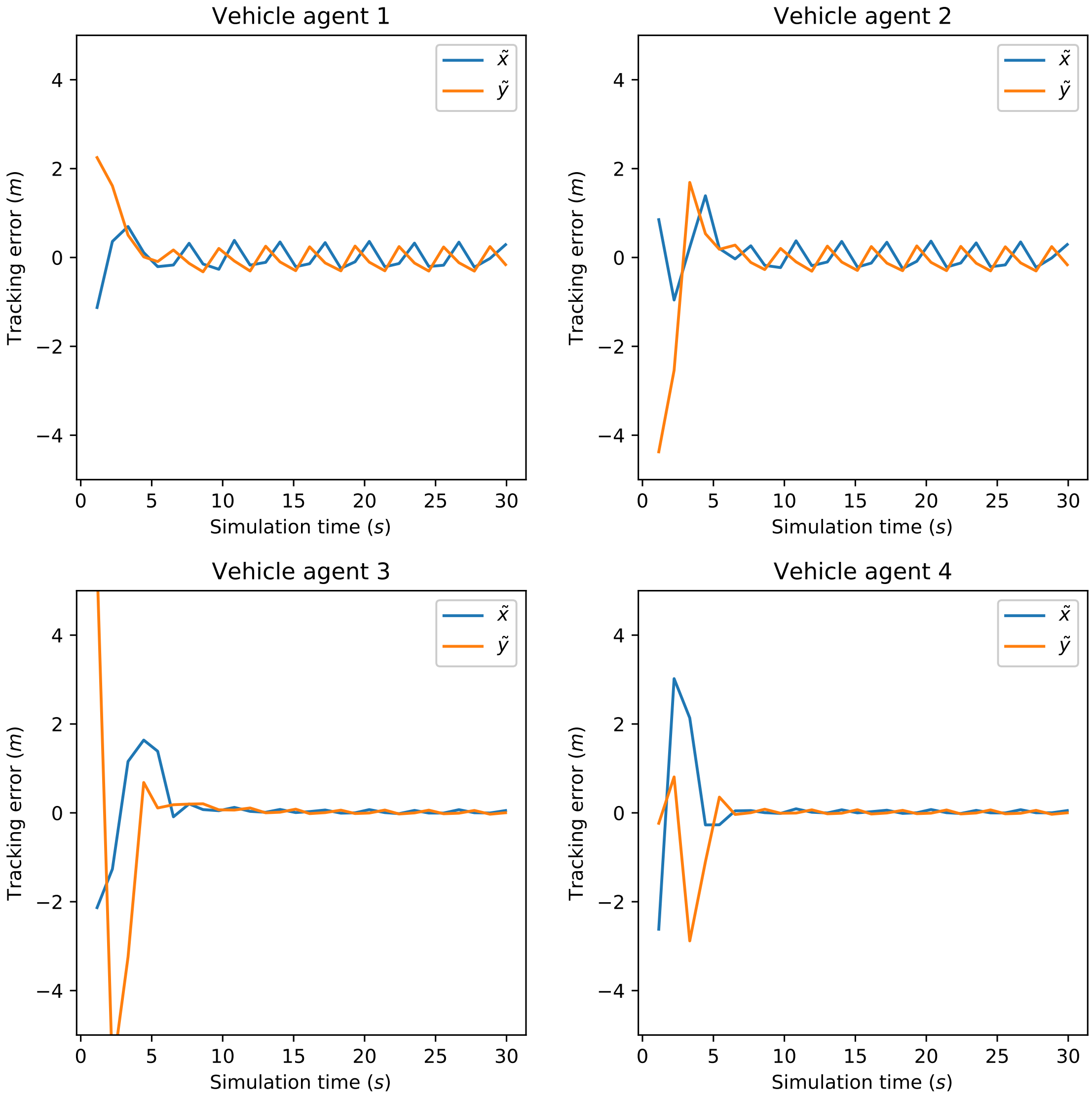


Figure 29: Tracking errors using experience-based controller with NN weight from vehicle agent 4 with traditional DL algorithm.

MAS.

5.4 Summary

In this chapter, we performed a physical simulation of the proposed controllers using the Gazebo simulator. Models of the unicycle-type vehicle are built in the Gazebo, and the proposed controllers are coded into Python, with the original weight updating law transferred into the discrete-time form. Gazebo simulation results show that tracking convergence is achieved using the proposed CALC, and the NN weights for all vehicle agents in the MAS reached consensus with the discrete-time weight updating law. It is also shown that the proposed experience-based controller is also able to drive the vehicles to their desired reference trajectories using any converged NN weight from the learning phase. In addition, the comparison of simulation results show the superiority of the proposed CALC over traditional DL algorithms.

List of References

[1] *Gazebo*. [Online]. Available: http://gazebosim.org/

[2] *Open Dynamics Engine*. [Online]. Available: http://www.ode.org/

[3] *Unified Robot Description Format*. [Online]. Available: http://wiki.ros.org/urdf

[4] *Simulation Description Format*. [Online]. Available: http://sdformat.org/

[5] *Robot Operating System*. [Online]. Available: https://www.ros.org/

[6] *ROS Documentation*. [Online]. Available: http://wiki.ros.org/

CHAPTER 6

Conclusions

6.1 Dissertation Contributions and Concluding Remarks

In this research, we consider a group of unicycle-type vehicles with generalized model assumptions and proposed a cooperative adaptive learning-based controller (CALC) and an experience-based controller for the trajectory tracking tasks in Chapter 3. Theoretical analysis and MATLAB simulations are provided to demonstrate the effectiveness of the proposed control methods, including adaptive tracking convergence, cooperative learning consensus and accurate approximation, as well as trajectory tracking using learned knowledge. Compared to results of traditional learning-based control methods designed for single agent systems, where NN weight convergence is only guaranteed locally along the PE trajectory of its own, it is shown that the proposed cooperative learning algorithm is able to accurately approximate the unstructured system uncertainties along the union of reference trajectories experienced by all vehicle agents in the MAS.

In Chapter 4, we extend the results of state feedback controllers proposed in Chapter 3 to output feedback cases with the estimated generalized velocities obtained from a high-gain observer. It is shown that the objectives raised for state feedback controllers can also be reached by the proposed output feedback controllers. Simulation results show effectiveness of the proposed output-feedback control algorithms.

In Chapter 5, we performed a simulation of the proposed controllers using the Gazebo simulator. To save the computational power for simulating the solutions of ODEs, the proposed continuous-time weight updating law is transformed into the discrete-time iteration form. Gazebo simulation results show that tracking convergence is achieved using the proposed CALC, and the NN weights for all vehicle agents in the MAS reached consensus with the discrete-time weight updating law. It is also shown that the proposed experience-based controller is also able to drive the vehicles to their desired reference trajectories using any converged NN weight from the learning phase.

Therefore, we can conclude that the objectives of this research, including tracking convergence, cooperative learning consensus, and accurate approximation along all trajectories, are achieved by the proposed CALC and experience-based control methods, through both state feedback and output feedback.

6.2 Future Works

Despite the contributions made in this research, there are still improvements and potential follow-up work to be done in the future. One potential improvement is the design of discrete-time observers under the condition that exact system model is not available.

Another potential improvement is to integrate the proposed control methods with collision avoidance. Different from the MATLAB simulation, where collision between vehicles is not considered, vehicle models could run into each other in the Gazebo simulator. Although the reference trajectories are moved apart in the Gazebo simulation to avoid potential collision, vehicles may still collide when running wider than the reference trajectories, especially at the beginning of the simulation when the vehicle dynamics is not yet accurately approximated by the RBFNN.

At last, further simulation studies should be performed to analyze the robustness of the proposed controller before moving to the applications on real vehicles, by introducing measuring noise in the simulation and minor difference on the vehicle models.

For further researches and applications based on the results in this study, the unicycle-type vehicle can serve as a mobile platform and be extended by mounting robotic manipulators. Applications for such mobile manipulators may include warehouse or home delivery, as well as operations on assembly lines where products can be assembled while moving along the line.

BIBLIOGRAPHY

Arduino motor shield. [Online]. Available: https://store.arduino.cc/usa/arduino-motor-shield-rev3

Gazebo. [Online]. Available: http://gazebosim.org/

Open Dynamics Engine. [Online]. Available: http://www.ode.org/

Robot Operating System. [Online]. Available: https://www.ros.org/

ROS Documentation. [Online]. Available: http://wiki.ros.org/

Simulation Description Format. [Online]. Available: http://sdformat.org/

Unified Robot Description Format. [Online]. Available: http://wiki.ros.org/urdf

Agaev, R. and Chebotarev, P., "The matrix of maximum out forests of a digraph and its applications," *arXiv preprint math/0602059*, 2006.

Araki, B., Gilitschenski, I., Ogata, T., Wallar, A., Schwarting, W., Choudhury, Z., Karaman, S., and Rus, D., "Range-based cooperative localization with nonlinear observability analysis," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1864–1870.

Atınç, G. M., Stipanović, D. M., Voulgaris, P. G., and Karkoub, M., "Collision-free trajectory tracking while preserving connectivity in unicycle multi-agent systems," in *2013 American control conference*. IEEE, 2013, pp. 5392–5397.

Barbalat, I., "Systemes d'équations différentielles d'oscillations non linéaires," *Rev. Math. Pures Appl*, vol. 4, no. 2, pp. 267–270, 1959.

Boker, A. and Yuan, C., "High-gain observer-based distributed tracking control of heterogeneous nonlinear multi-agent systems," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 6639–6644.

Buhmann, M. D., *Radial basis functions: theory and implementations*. Cambridge university press, 2003, vol. 12.

Cai, X. and de Queiroz, M., "Adaptive rigidity-based formation control for multirobotic vehicles with dynamics," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 389–396, 2015.

Chen, W., Wen, C., Hua, S., and Sun, C., "Distributed cooperative adaptive identification and control for a group of continuous-time systems with a cooperative pe condition via consensus," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 91–106, 2014.

Chu, X., Peng, Z., Wen, G., and Rahmani, A., "Robust fixed-time consensus tracking with application to formation control of unicycles," *IET Control Theory & Applications*, vol. 12, no. 1, pp. 53–59, 2017.

Chu, X., Peng, Z., Wen, G., and Rahmani, A., "Distributed fixed-time formation tracking of multi-robot systems with nonholonomic constraints," *Neurocomputing*, vol. 313, pp. 167–174, 2018.

Chu, X., Peng, Z., Wen, G., and Rahmani, A., "Distributed formation tracking of multi-robot systems with nonholonomic constraint via event-triggered approach," *Neurocomputing*, vol. 275, pp. 121–131, 2018.

Dinh, T. H., Phung, M. D., Tran, T. H., and Tran, Q. V., "Localization of a unicycle-like mobile robot using lrf and omni-directional camera," in *2012 IEEE International Conference on Control System, Computing and Engineering.* IEEE, 2012, pp. 477–482.

Do, K. and Pan, J., "Nonlinear formation control of unicycle-type mobile robots," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 191–204, 2007.

Do, K. D., "Formation tracking control of unicycle-type mobile robots with limited sensing ranges," *IEEE transactions on control systems technology*, vol. 16, no. 3, pp. 527–538, 2008.

Dong, W. and Kuhnert, K.-D., "Robust adaptive control of nonholonomic mobile robot with parameter and nonparameter uncertainties," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 261–266, 2005.

Eugene, L., Kevin, W., and Howe, D., *Robust and adaptive control with aerospace applications.* Springer London, 2013.

Fierro, R. and Lewis, F. L., "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 4. IEEE, 1995, pp. 3805–3810.

Ghommam, J., Mehrjerdi, H., Saad, M., and Mnif, F., "Formation path following control of unicycle-type mobile robots," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 727–736, 2010.

Haddad, M., Khalil, W., and Lehtihet, H., "Trajectory planning of unicycle mobile robots with a trapezoidal-velocity constraint," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 954–962, 2010.

Hibbeler, R. C., *Engineering mechanics. Statics and dynamics. 11th.* Upper Saddle River, NJ: Pearson/Prentice-Hall. xvi, 1997.

Ioannou, P. A. and Sun, J., *Robust adaptive control.* Courier Corporation, 2012.

Jiang, Z.-P., Lefeber, E., and Nijmeijer, H., “Saturated stabilization and tracking of a nonholonomic mobile robot,” *Systems & Control Letters*, vol. 42, no. 5, pp. 327–332, 2001.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T., “A stable tracking control method for an autonomous mobile robot,” in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on.* IEEE, 1990, pp. 384–389.

Khalil, H. K., “High-gain observers in nonlinear feedback control,” in *2008 International Conference on Control, Automation and Systems.* IEEE, 2008, pp. xlvii–lvii.

Khoo, S., Xie, L., and Man, Z., “Robust finite-time consensus tracking algorithm for multirobot systems,” *IEEE/ASME transactions on mechatronics*, vol. 14, no. 2, pp. 219–228, 2009.

Lee, K. W. and Khalil, H. K., “Adaptive output feedback control of robot manipulators using high-gain observer,” *International Journal of Control*, vol. 67, no. 6, pp. 869–886, 1997.

Li, K., Yuan, C., Wang, J., and Dong, X., “Four-direction search scheme of path planning for mobile agents,” *Robotica*, vol. 38, no. 3, pp. 531–540, 2020.

Liu, Z., Wang, L., Wang, J., Dong, D., and Hu, X., “Distributed sampled-data control of nonholonomic multi-robot systems with proximity networks,” *Automatica*, vol. 77, pp. 170–179, 2017.

Luenberger, D., “An introduction to observers,” *IEEE Transactions on automatic control*, vol. 16, no. 6, pp. 596–602, 1971.

Luenberger, D. G., “Observing the state of a linear system,” *IEEE transactions on military electronics*, vol. 8, no. 2, pp. 74–80, 1964.

Lynch, N. A., *Distributed algorithms.* Elsevier, 1996.

Mariottini, G. L., Morbidi, F., Prattichizzo, D., Vander Valk, N., Michael, N., Pappas, G., and Daniilidis, K., “Vision-based localization for leader–follower formation control,” *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1431–1438, 2009.

Marquez, H. J., *Nonlinear Control Systems: Analysis and Design.* Hoboken: Wiley-Interscience, 2003, vol. 1.

Meyer, C. D., *Matrix analysis and applied linear algebra.* Siam, 2000, vol. 71.

Miao, Z. and Wang, Y., “Adaptive control for simultaneous stabilization and tracking of unicycle mobile robots,” *Asian Journal of Control*, vol. 17, no. 6, pp. 2277–2288, 2015.

Oh, S. and Khalil, H. K., "Nonlinear output-feedback tracking using high-gain observer and variable structure control," *Automatica*, vol. 33, no. 10, pp. 1845–1856, 1997.

Olfati-Saber, R. and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

Panagou, D., "A distributed feedback motion planning protocol for multiple unicycle agents of different classes," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1178–1193, 2016.

Park, J. and Sandberg, I. W., "Universal approximation using radial-basis-function networks," *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.

Pathak, K. and Agrawal, S. K., "An integrated path-planning and control approach for nonholonomic unicycles using switched local potentials," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1201–1208, 2005.

Peng, Z., Wen, G., Rahmani, A., and Yu, Y., "Distributed consensus-based formation control for multiple nonholonomic mobile robots with a specified reference trajectory," *International Journal of Systems Science*, vol. 46, no. 8, pp. 1447–1457, 2015.

Reeds, J. and Shepp, L., "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

Rossomando, F. G. and Soria, C. M., "Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural pid," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1179–1191, 2015.

Roy, S., Nandy, S., Kar, I. N., Ray, R., and Shome, S. N., "Robust control of nonholonomic wheeled mobile robot with past information: Theory and experiment," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 231, no. 3, pp. 178–188, 2017.

Roy, S., Nandy, S., Ray, R., and Shome, S. N., "Robust path tracking control of nonholonomic wheeled mobile robot: Experimental validation," *International Journal of Control, Automation and Systems*, vol. 13, no. 4, pp. 897–905, 2015.

Sert, H., Perruquetti, W., Kokosy, A., Jin, X., and Palos, J., "Localizability of unicycle mobiles robots: An algebraic point of view," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2012, pp. 223–228.

Shi, S., Yu, X., and Khoo, S., "Robust finite-time tracking control of nonholonomic mobile robots without velocity measurements," *International Journal of Control*, vol. 89, no. 2, pp. 411–423, 2016.

Shojaei, K., Shahri, A. M., and Tarakameh, A., "Adaptive feedback linearizing control of nonholonomic wheeled mobile robots in presence of parametric and nonparametric uncertainties," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 194–204, 2011.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., *Robotics: modelling, planning and control.* Springer Science & Business Media, 2010.

Stegagno, P. and Yuan, C., "Distributed cooperative adaptive state estimation and system identification for multi-agent systems," *IET Control Theory & Applications*, vol. 13, no. 1, pp. 815–822, 2019.

Vasilopoulos, V., Arslan, O., De, A., and Koditschek, D. E., "Sensor-based legged robot homing using range-only target localization," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO).* IEEE, 2017, pp. 2630–2637.

Waldteufel, P. and Corbin, H., "On the analysis of single-doppler radar data," *Journal of Applied Meteorology*, vol. 18, no. 4, pp. 532–542, 1979.

Wang, C. and Hill, D. J., "Learning from neural control," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 130–146, 2006.

Wang, C. and Hill, D. J., *Deterministic learning theory for identification, recognition, and control.* CRC Press, 2009, vol. 32.

Yu, X., Liu, L., and Feng, G., "Trajectory tracking for nonholonomic vehicles with velocity constraints," *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 918–923, 2015.

Yuan, C., "Distributed adaptive switching consensus control of heterogeneous multi-agent systems with switched leader dynamics," *Nonlinear Analysis: Hybrid Systems*, vol. 26, pp. 274–283, 2017.

Yuan, C., He, H., Dong, X., and Ilbeigi, S., "Robust distributed adaptive containment control of heterogeneous linear uncertain multi-agent systems," in *2018 Annual American Control Conference (ACC).* IEEE, 2018, pp. 3660–3665.

Yuan, C., He, H., and Wang, C., "Cooperative deterministic learning-based formation control for a group of nonlinear uncertain mechanical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 319–333, 2019.

Yuan, C., Licht, S., and He, H., "Formation learning control of multiple autonomous underwater vehicles with heterogeneous nonlinear uncertain dynamics," *IEEE transactions on cybernetics*, no. 99, pp. 1–15, 2017.

Zeng, W., Wang, Q., Liu, F., and Wang, Y., "Learning from adaptive neural network output feedback control of a unicycle-type mobile robot," *ISA transactions*, vol. 61, pp. 337–347, 2016.

Zhang, Y., Liu, G., and Luo, B., "Finite-time cascaded tracking control approach for mobile robots," *Information Sciences*, vol. 284, pp. 31–43, 2014.