01 Jan 2023

Extended Kalman Filter based Resilient Formation Tracking Control of Multiple Unmanned Vehicles Via Game-Theoretical Reinforcement Learning

Lei Xue

Bei Ma

Jian Liu

Chaoxu Mu

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/ele_comeng_facwork/4718

Recommended Citation

L. Xue et al., "Extended Kalman Filter based Resilient Formation Tracking Control of Multiple Unmanned Vehicles Via Game-Theoretical Reinforcement Learning," *IEEE Transactions on Intelligent Vehicles*, Institute of Electrical and Electronics Engineers, Jan 2023.
The definitive version is available at https://doi.org/10.1109/TIV.2023.3237790

Extended Kalman Filter Based Resilient Formation Tracking Control of Multiple Unmanned Vehicles via Game-Theoretical Reinforcement Learning

Lei Xue, *Member, IEEE,* Bei Ma, Jian Liu, *Member, IEEE,* Chaoxu Mu, *Senior Member, IEEE,* Donald C. Wunsch, *Fellow, IEEE,*

***Abstract*—In this paper, we discuss the resilient formation tracking control problem of multiple unmanned vehicles (MUV). A dynamic leader-follower distributed control structure is utilized to optimize the performance of the formation tracking. For the follower of the MUV, the leader is a cooperative unmanned vehicle, and the target of formation tracking is a non-cooperative unmanned vehicle with a nonlinear trajectory. Therefore, an extended Kalman filter (EKF) observer is designed to estimate the state of the target. Then the leader of the MUV is adjusted dynamically according to the state of the target. In order to describe the interactions between the follower and dynamic leader, a Stackelberg game model is constructed to handle the hierarchical decision problems. At the lower layer, each follower responds by observing the leader's strategy, and the potential game is used to prove a Nash equilibrium among all followers. At the upper layer, the dynamic leader makes decisions depending on the response of all followers to reaching the Stackelberg equilibrium. Moreover, the Stackelberg-Nash equilibrium of the designed game theoretical model is proven. A novel reinforcement learning-based algorithm is designed to achieve the Stackelberg-Nash equilibrium of the game. Finally, the effectiveness of the method is verified by a variety of formation tracking simulation experiments.**

***Index Terms*—extended Kalman filter; leader-switching; formation tracking; Stackelberg-Nash equilibrium; reinforcement learning**

I. Introduction

In recent years, unmanned vehicles are widely used in military and civil fields [1]–[5]. Therefore, the cooperative control of the MUV becomes one of the most active research problems, such as formation control, coordination control, and target tracking. The formation tracking problem can be stated as MUV needs to track a given target trajectory while maintaining the desired formation shape to accomplish a specific task. Research on this problem can be applied to many fields, such as cooperative reconnaissance, search and rescue missions, forest fire detection, and environmental monitoring [6]–[9].

Most of the formation tracking problems of the MUV rely upon the leader-follower structure [10]–[13], which assumes the presence of a special vehicle, referred to as the leader. The leader has access to the target trajectory to be transmitted to the whole system. Therefore, the leader needs to estimate the state of the target at the next time step by the historical information and observations of the target. One of the most classical methods for target trajectory prediction is the Kalman filter (KF) algorithm [14], which has excellent estimation results for linear systems. The KF was used to estimate the parameters during real-time formation tracking in [15] and [16]. The KF can estimate the parameters of linear system models. As to a nonlinear trajectory, the extended Kalman filter (EKF) was designed to linearize the nonlinear function for linear approximation [17]. In addition, the convergence of the EKF algorithm for nonlinear discrete systems was discussed in [18], and sufficient conditions to ensure local asymptotic convergence were obtained. Inspired by the above work, an EKF algorithm is designed to estimate the nonlinear trajectory of the target, which provides a guarantee for the implementation of the formation tracking problem.

Leader selection is crucial in the leader-follower formation tracking problem because it significantly affects the performance of formation tracking for the multi-agent system (MAS) [19]–[24]. Under some circumstances, the leader is considered to be a particular member chosen by the systems at the beginning of the task. However, both [25] and [26] demonstrated that the system performance can be optimized by online leader selection. A suitable error function was defined to represent the tracking performance of MAS. Subsequently, a distributed adaptive algorithm was designed for selecting the optimal leader among the current neighbors. In [27], an emotion-based leader selection model was proposed, in which robots can choose the leader according to their affection state. In [28], a supermodel game was introduced to solve the problem of selecting k leaders in a second-order MAS. Therefore, an online leader selection algorithm is designed to optimize the performance of formation tracking by online leader selection.

Game theory is an effective tool to deal with cooperative control of MAS. In [29], the relationship between game theory and cooperative control was elaborated, as well as how game theory can be combined to solve cooperative control problems. In [30], distributed multi-agent control, optimal control theory,

L. Xue, B. Ma, J. Liu are with the Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education; School of Automation, Southeast University, Nanjing 210096, China (e-mail: leixue, 220211907@seu.edu.cn; bkliujian@163.com).

C. Mu is with the School of of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: cxmu@tju.edu.cn).

D. C. Wunsch is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA (e-mail: dwunsch@mst.edu).

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB1714700, the National Natural Science Foundation of China under Grant 62022061 and Grant 62273094, and the "Zhishan" Scholars Programs of Southeast University. (Corresponding author: Chaoxu Mu.)

and game theory were combined to address multi-agent dynamic graph games. For a dynamic leader-follower formation tracking distributed control structure in our study, there are two kinds of goals, which are cooperative and non-cooperative goals. The cooperative goal is the leader. The follower follows the leader in the MUV system. The non-cooperative goals are targets of the system, i.e., nonlinear trajectories that need to be estimated and predicted by the leader. In [31], a hierarchical formation control structure was constructed by dividing large-scale agents into three categories. The Stackelberg game [32]–[34] is an effective tool for describing the hierarchical decision-making process of MAS with cooperative and non-cooperative goals. Therefore, a Stackelberg game is designed to describe the interactions between the vehicles of the MUV system.

Due to the computational complexity of the game equilibrium, many studies have solved the game model by designing reinforcement learning-based game strategies. In [32], a two-level value iteration-based integral reinforcement learning (VI-IRL) algorithm was developed to overcome the difficulty of computing equilibrium points. In [35], a novel actor-critic reinforcement learning approach was designed to solve two coupled equations of the mean-field game. In [36]–[38], game theory and reinforcement learning were combined to design reinforcement learning-based strategies. Therefore, a reinforcement learning-based algorithm is developed to achieve the Stackelberg-Nash equilibrium of the game model.

In this paper, we study the formation tracking problem of a MUV system in which the trajectory of the target is nonlinear. A dynamic leader-follower distributed control structure is constructed to optimize the formation tracking process. Subsequently, the leader is switched online to improve the tracking performance. Moreover, a Stackelberg game is designed to illustrate the interactions between the dynamic leaders and the followers, and the existence of Stackelberg-Nash equilibrium is proven. Finally, a reinforcement learning-based algorithm is developed to optimize the decision making process of unmanned vehicles.

The main contributions of this paper can be summarized as follows:

1) A real-time leader-switching algorithm is designed to construct a dynamic leader-follower structure, which improves the performance of dynamic target formation tracking. Compared with [25] and [26], our nonlinear target is estimated by the designed EKF observer.
2) A hierarchical decision problem is illustrated by constructing a Stackelberg game with a dynamic leader and followers. Compared with [32], the existence of the Stackelberg-Nash equilibrium is proven by introducing an ordinal potential game.
3) A novel Q-learning algorithm is developed to reach the Stackelberg-Nash equilibrium by introducing leader switching and quadrant pre-determination in the traditional Q-learning algorithm.

The rest of our paper is organized as follows. Section II introduces the formation tracking problem for the MUV system and states the problem. In Section III, the EKF observer is designed for state estimation of a nonlinear target. Section IV designs a real-time leader-switching algorithm and a Stackelberg game model. Section V develops the novel reinforcement learning-based algorithm for solving the game model. Section VI gives some simulation examples to illustrate the effectiveness of the game theoretical methods. Finally, Section VII gives the conclusion.

II. Problem Formulation

In this section, the model of the MUV system is introduced. Then the objective of formation tracking is established. Finally, the formation tracking problem is transformed into an optimization problem by combining the desired objectives.

A. System Model

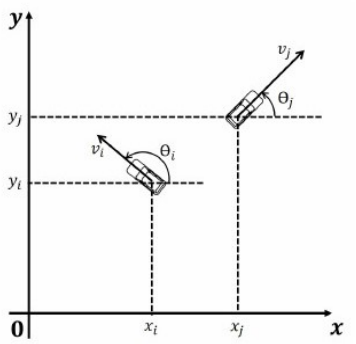


Fig. 1. The absolute coordiignate system of the MUV system.

All vehicles in the system are considered as a set $\mathbf{N} = \{1, 2, ..., n\}$ that can be divided into leaders and followers, denoted by $\mathbf{L}$ and $\mathbf{F}$, respectively. The topology of the system is an undirected graph. The connectivity matrix is defined as $\mathbf{E} = \{e_{ij} \mid i, j \in \mathbf{N}\}$ with $e_{ij} = 1$ if ith vehicle can exchange information with the jth vehicle and $e_{ij} = 0$ otherwise. $\mathbf{N}_i = \{j \mid e_{ij} = 1, j \in \mathbf{N}\}$ denotes the set of all neighbors of the ith vehicle. The absolute coordinate system of this MUV system is shown in Fig. 1.

According to the coordinate system shown in Fig. 1, the dynamics model of the ith vehicle can be written as:

$$\begin{cases} \dot{x}_i = v_i \cos\theta_i \\ \dot{y}_i = v_i \sin\theta_i \\ \dot{\theta}_i = \phi_i \end{cases} \tag{1}$$

where x_i and y_i represent the horizontal and vertical coordinates of the ith vehicle, respectively. v_i represents the velocity of the ith vehicle. The θ_i and ϕ_i represent the heading angle and turning rate of the ith vehicle, respectively. Therefore, the position of the ith vehicle can be expressed as $p_i = (x_i, y_i)$.

Based on the dynamics of the ith vehicle, the motion constraint is defined as:

$$\begin{cases} 0 \le |v_i| \le |v_m| \\ 0 \le |\phi_i| \le |\phi_m| \end{cases} \tag{2}$$

where $|v_m|$ and $|\phi_m|$ denote the modulus of the maximum speed and maximum turning rate of the ith vehicle, respectively.

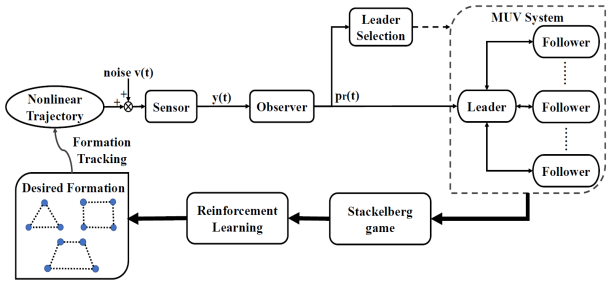


Fig. 2. Schematic of the nonlinear target formation tracking problem.

B. Problem Statement

As shown in Fig. 2, an observer is designed to estimate the nonlinear trajectory. Then the vehicles of the MUV system are classified into leaders and followers by the leader selection algorithm. In addition, a Stackelberg game model of the formation tracking problem is designed. Finally, a reinforcement learning-based algorithm is designed to solve the game model.

For the formation tracking problem of the MUV system, to improve the tracking performance and maintain the desired formation shape, the following three objectives should be satisfied:

Objective 1: Use the leader as a time variable to optimize the performance of tracking the target and achieving the desired formation.

Objective 2: The leader tracks the target with a nonlinear trajectory, then the distance between the leader and the target should converge to zero.

Objective 3: All vehicles should maintain the desired formation, which means that the relative distance between vehicles should be given by $d = \{d_{ij}\}$, where d_{ij} denotes the distance between the ith vehicle and the jth vehicle. Then $|\|p_i - p_j\|_2 - d_{ij}|$ should converge to zero.

Combining the above three objectives, the objective function f_i of the ith vehicle can be defined as:

$$f_i = \begin{cases} \|p_i - p_r\|_2 + \\ \sum_{j\in\mathbf{N}_i} e_{ij} \left|\|p_i - p_j\|_2 - d_{ij}\right|, & i \in \mathbf{L} \\ \\ \sum_{j\in\mathbf{N}_i} e_{ij} \left|\|p_i - p_j\|_2 - d_{ij}\right|, & i \in \mathbf{F} \end{cases} \tag{3}$$

where p_r is the target trajectory, $\mathbf{L}$ and $\mathbf{F}$ denote the set of leaders and the set of followers, respectively.

An objective function $f_{\mathbf{L}}$ for the set of leaders is designed as follows:

$$f_{\mathbf{L}} = \sum_{i\in\mathbf{L}} f_i, \quad i \in \mathbf{L} \tag{4}$$

Therefore, the choice of the leader affects the speed of convergence of the whole system to the desired formation. Note that our paper studies the case where there is only one leader in the set of leaders.

Based on the above analysis, the formation tracking problem is converted into an optimization problem:

Problem: For each vehicle in the MUV system, the problem is to find its strategy for achieving:

$$\begin{array}{l} min\ f_{\mathbf{L}} \\ min_{j\in\mathbf{F}}\ f_j \end{array} \tag{5}$$

The observation of nonlinear trajectories is a prerequisite for solving the optimization problem, which is exactly addressed in Section III.

III. Design of EKF-Based Observer for Tracking the Nonlinear Trajectory

For estimating the nonlinear trajectory of the target and eliminating the noise of the environment, a suitable filtering algorithm is designed to improve the performance of the formation tracking. Therefore, a modified EKF-based observer is designed and the convergence of the algorithm is proven theoretically.

A. Linearization of Nonlinear Trajectory

Due to the turning rate constraint, the vehicle can only make circular turns. Therefore, the nonlinear trajectory of the target point is assumed to be a circle. The dynamic equation of the nonlinear trajectory is shown as follows:

$$\begin{cases} \dot{p}_{rx}(t) = -wl \cdot \sin(wt) \\ \dot{p}_{ry}(t) = wl \cdot \cos(wt) \end{cases} \tag{6}$$

where $p_{rx}(t)$ and $p_{ry}(t)$ denote the horizontal and vertical coordinates, respectively. $p_r(t) = (p_{rx}(t), p_{ry}(t))$ is the target coordinates. Both w and l are constants that affect the size of the circular trajectory.

When using the sensor to measure the position, there exists a measurement noise $v(t)$ with a variance of σ_1, where $v(t) = [v_1(t), v_2(t)]^T$. There is also a process noise $z(t)$ with a variance of σ_2, where $z(t) = [z_1(t), z_2(t)]^T$. Then (6) can be rewritten as a nonlinear state-space expression:

$$\begin{cases} \dot{p}_r(t) = \begin{bmatrix} \dot{p}_{rx}(t) \\ \dot{p}_{ry}(t) \end{bmatrix} = wl \begin{bmatrix} -\sin(wt) \\ \cos(wt) \end{bmatrix} + \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} \\ y(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{rx}(t) \\ p_{ry}(t) \end{bmatrix} + \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \end{cases} \tag{7}$$

where $y(t)$ is the measured signal with noise.

To design a discrete-time Kalman filter, the discrete-time model with the sampling interval Δt of equation (7) by the approximation of the first-order derivative can be defined as follows:

$$\begin{cases} \dot{p}_{rx}(t) \approx \frac{p_{rx}(k) - p_{rx}(k-1)}{\Delta t} \mid t = (k-1) \\ \dot{p}_{ry}(t) \approx \frac{p_{ry}(k) - p_{ry}(k-1)}{\Delta t} \mid t = (k-1) \end{cases} \tag{8}$$

where Δt is the sampling time; $p_{rx}(k)$ and $p_{rx}(k-1)$ denote the horizontal coordinates of the target point at k moments and $k-1$ moments; $p_{ry}(k)$ and $p_{ry}(k-1)$ denote the vertical coordinates of the target point at k and $k-1$ moments, respectively.

Substituting (8) into (7), the discrete nonlinear model of the target point can be obtained:

$$\begin{array}{l} p_r(k) = \begin{bmatrix} p_{rx}(k) \\ p_{ry}(k) \end{bmatrix} = \begin{bmatrix} p_{rx}(k-1) \\ p_{ry}(k-1) \end{bmatrix} \\ +\Delta t \begin{bmatrix} -wl \cdot \sin(w(k-1)) \\ wl \cdot \cos(w(k-1)) \end{bmatrix} + \Delta t \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} \end{array} \tag{9}$$

The nonlinear function f_1 and f_2 can be defined as:

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} wl \cdot \sin(w(k-1))\Delta t \\ wl \cdot \cos(w(k-1))\Delta t \end{bmatrix} \tag{10}$$

where $f\left(\hat{p}_r(k-1)^+\right) = [f_1, f_2]^T$, $\hat{p}_{rx}(k-1)^+$ and $\hat{p}_{ry}(k-1)^+$ denote the posterior estimates of the horizontal and vertical coordinates of the target at time step $k-1$, respectively.

The first-order Taylor expansion of the nonlinear function is performed by choosing $\hat{p}_r(k-1)^+$ as the working point. Thus, the linearized Jacobian matrix A_p is obtained:

$$A_p = \begin{bmatrix} \frac{\partial f_1}{\partial p_{rx}} & \frac{\partial f_1}{\partial p_{ry}} \\ \frac{\partial f_2}{\partial p_{rx}} & \frac{\partial f_2}{\partial p_{ry}} \end{bmatrix} \mid \hat{p}_r(k-1)^+ \tag{11}$$

In conclusion, a linear time-varying model is obtained. During the next subsection, a modified EKF based observer is designed.

B. The Design of Modified EKF-Based Observer

The process noise $z(k)$ and the measurement noise $v(k)$ are white noise with mean 0, whose covariance matrices are $Q(k)$ and $R(k)$, respectively.

They are expressed in the following forms:

$$\begin{aligned} \mathbb{E}\left[z(k)z(l)^T\right] &= Q(k)\delta(k-l) \\ \mathbb{E}\left[v(k)v(l)^T\right] &= R(k)\delta(k-l) \\ \mathbb{E}\left[z(k)v(l)^T\right] &= 0 \end{aligned} \tag{12}$$

where $\delta(k-l)$ is Kronecker delta function. If $k = l$, $\delta(k-l) = 1$; if $k \neq l$, $\delta(k-l) = 0$.

The nonlinear trajectory can be expressed as a discrete nonlinear dynamic model in the general case:

$$\begin{cases} p_r(k) = p_r(k-1) + \Delta t \cdot f(\hat{p}_r(k-1)) + \Delta t \cdot z(k-1) \\ y(k-1) = C \cdot p_r(k-1) + v(k-1) \end{cases} \tag{13}$$

where $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, the same matrix as in equation (7).

Then, depending on the EKF, the local filter process can be presented by the following recursion:

$$\begin{aligned} \hat{p}_r(k)^- &= \hat{p}_r(k-1)^+ + f(\hat{p}_r(k-1)^+)\Delta t \\ P(k)^- &= A(\hat{p}_r(k-1)^+)P(k-1)^+A(\hat{p}_r(k-1)^+)^T + Q(k-1) \\ K(k) &= P(k)^-C^T((CP(k)^-C)^T + R(k))^{-1} \\ \hat{p}_r(k)^+ &= \hat{p}_r(k)^- + K(k)(y(k) - C\hat{p}_r(k-1)) \\ P(k)^+ &= (I - K(k)C)P(k)^-(I - K(k)C)^T + K(k)R(k)K(k)^T \end{aligned}$$

The modified EKF algorithm for observing the nonlinear trajectory is designed as Algorithm 1.

For proving the convergence of the modified EKF algorithm, two lemmas were introduced as follows.

Lemma 1. *[18]*

Designing the Lyapunov function:

$$V(k) = (\tilde{p}_r(k)^+)^T(P(k)^+)^{-1}\tilde{p}_r(k)^+$$

where $\tilde{p}_r(k)^- = p_r(k)^- - \hat{p}_r(k)^-$, $\tilde{p}_r(k-1)^+ = p_r(k-1)^+ - \hat{p}_r(k-1)^+$.

Therefore, by the constraints of $Q(k)$ and $R(k)$, $V(k)$ decreases and converges to a positive scalar V.

Lemma 2. *[18]*

The following three relationships hold:

$$\lim_{k\to\infty} \lambda_{min}((P(k)^+)^{-1}) = \infty \tag{14}$$

$$\frac{\lambda_{min}((P(k)^+)^{-1}(\tilde{p}_r(k)^+)^T\tilde{p}_r(k)^+)}{n\lambda_{max}((P(k)^+)^{-1})} \leq \frac{V(k)}{tr((P(k)^+)^{-1})} \tag{15}$$

$$\lim_{k\to\infty} \sup \frac{\lambda_{max}((P(k)^+)^{-1})}{\lambda_{min}((P(k)^+)^{-1})} < \infty \tag{16}$$

Theorem 1. *By adjusting $Q(k)$ and $R(k)$, the modified EKF algorithm can converge to 0, which means the error between the estimated value and the true value converges to zero.*

Proof. See Appendix A. □

Algorithm 1 Modified EKF(n) Algorithm.

Input: w, l: Parameters of determine the radius of the circle;
$\hat{p}_{rx}^+, \hat{p}_{ry}^+$: Initial estimated state;
Δt: Sampling time;
Q, R: The matrices affecting the effect of the algorithm;
B: Input Matrix; C: Output Matrix;
P^+: Unit matrix with the same number of dimensions as the input matrix B;

Output: estimated value $p_r(t_{n-1})$.

for kk = 2 to n do
$\hat{p}_{rx}^- = \hat{p}_{rx}^+ - wl \cdot \sin\left((kk-1)\,w\right)\Delta t$;
$\hat{p}_{ry}^- = \hat{p}_{ry}^+ + wl \cdot \cos\left((kk-1)\,w\right)\Delta t$;
$\hat{p}_r^- = \left[\hat{p}_{rx}^-, \hat{p}_{ry}^-\right]^T$;
$A = I + A_p\Delta t$;
$P^- = AP^+A^T + Q$;
$K = P^-C^T\left(\left(CP^-C\right)^T + R\right)^{-1}$;
$\hat{p}_r^+ = \hat{p}_r^- + K\left(y - C\hat{p}_r^-\right)$;
$p_r(t_{kk-1}) \leftarrow \hat{p}_r^+$;
$P^+ = \left(I - KC\right)P^-\left(I - KC\right)^T + KRK^T$;
end for
return $p_r(t_{n-1})$**; exit**

In this section, the leader of the MUV can observe the trajectory of the target by using the modified EKF algorithm. Therefore, how to dynamically adjust the leader in the formation tracking problem is the core issue, and this issue is addressed in Section IV.

IV. Formation Tracking Stackelberg Game

A. Design of Leader Switching Algorithm

For the formation tracking of the MUV system, the leader during the tracking process is dynamically adjusted for optimizing the performance. Therefore, a leader switching algorithm is designed.

The leader is set as a variable with a variation period of T. Suppose the leader is l_k at the time step t_1, the leader will be updated from the $\mathbf{N}_{l_k}$ at time step t_{1+T} by the state of the

target. Referring to the form of the equation (4) in Section II, the error function $f_{\mathbf{L}}(i)$ is defined as follows:

$$f_{\mathbf{L}}(i) = \|p_i - p_r\|_2 + \sum_{j\in\mathbf{N}_i} e_{ij} \left| \|p_i - p_j\|_2 - d_{ij} \right| \quad (17)$$

where $i \in i \cup \mathbf{N}_i$.

It is known from (17) that the choice of the leader affects the speed of convergence to the desired target state and the position and velocity errors with the followers. Therefore, it can be used as an error function for leader selection. The leader-switching algorithm is given as follows.

Algorithm 2 LEADER-SWITCHING(l_0, T)

```
The initial leader is denoted by l_0;
k ← 0;
while ture do
Every time T, switch leader;(Note that T is the same as Δt in Algorithm 1)
k ← k + 1;
p_r(t_k) ← EKF(k + 1)
if the leader receives a new value of target point at time k p_r(t_k) then
    The current leader l_{k-1} sends p_r(t_k) to its neighbors in N_{l_{k-1}} and calculate the error function f_L;
    if min_{m∈N_{l_{k-1}}} f_L(m) < f_L(l_{k-1}) then
        l_k = argmin_{m∈N_{l_{k-1}}} f_L(m);
    else
        l_k = l_{k-1};
    end if
end if
end
```

Therefore, accurate estimation of the target trajectory using EKF is a prerequisite for leader selection and for establishing a game theory model. By forming a new set of leaders and followers, the problem can be described as a hierarchical decision problem. The Stackelberg game is established to illustrate the interactions between the vehicles.

B. Design of Stackelberg Game Model

The hierarchical decision problem can be modeled as a Stackelberg game when there is a leader and multiple followers, and all followers make decisions simultaneously [32]. As shown in Fig. 3, the optimal leader is first selected by a leader-switching algorithm, and then a dynamic leader-follower structure is obtained. In a Stackelberg game, the leader makes a decision first, and all followers make decisions simultaneously according to the leader's decision. All followers arrive at the corresponding positions and thus from the desired shape. Thereafter, the leader adjusts the decision according to the decision of the followers. The iteration will be ended until a Stackelberg-Nash equilibrium is reached.

Therefore, the Stackelberg game model for the formation tracking problem is established. The elements of Stackelberg game model $\mathbf{G} = (\mathbf{N}, \mathbf{A}, \mathbf{U})$ are defined as follows.

- $\mathbf{N} = \{\mathbf{L}, \mathbf{F}\}$ is the set of vehicles playing the game, where $\mathbf{L}$ is the set of leaders, $\mathbf{F}$ is the set of followers;

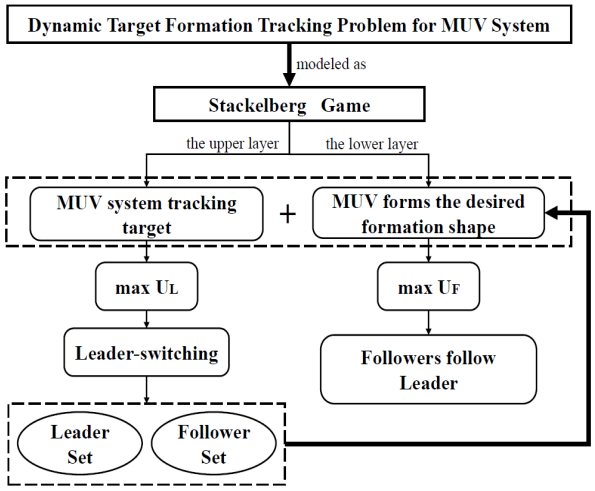


Fig. 3. The relationship between leader-switching and stackelberg game.

- $\mathbf{A} = \{\mathbf{A_L}, \mathbf{A_F}\}$ is the action set, $\mathbf{A_L} = \{a_i \mid i \in \mathbf{L}\}$ is the action set of the leader, $\mathbf{A_F} = \{a_j \mid j \in \mathbf{F}\}$ is the action set of the follower;
- $\mathbf{U} = \{\mathbf{U_L}, \mathbf{U_F}\}$ is the utility function, $\mathbf{U_L} = \{\mathbf{U}_i \mid i \in \mathbf{L}\}$ is the utility function of the leader, $\mathbf{U_F} = \{\mathbf{U}_j \mid j \in \mathbf{F}\}$ is the utility function of the follower.

The utility function is given as follows:

$$\mathbf{U}_i = -\|p_i - p_r\|_2 - \sum_{j\in\mathbf{N}_i} e_{ij} \left| \|p_i - p_j\|_2 - d_{ij} \right|, \quad i \in \mathbf{L}$$

$$\mathbf{U}_j = -\sum_{k\in\mathbf{N}_j} e_{jk} \left| \|p_j - p_k\|_2 - d_{jk} \right|, \quad j \in \mathbf{F}$$

where $\mathbf{N}_j$ denotes the set of neighbors of the jth vehicle; d_{jk} denotes the distance between the jth vehicle and the kth vehicle.

Throughout the Starkberg game, the leader's goal is to find an action a_i^* that maximizes the utility function $\mathbf{U}_i$. The follower's goal is to find an action a_j^* that maximizes the utility function $\mathbf{U}_j$. Thus (a_i^*, a_j^*) is the equilibrium point of the Starkberg game. Therefore, it is important to prove the existence of the equilibrium point.

C. Proof of Starkberg-Nash Equilibrium

According to the definition of equilibrium in [24], the Stackelberg-Nash equilibrium is given to describe the feature of the designed game model.

Definition 1. *(Starkberg-Nash Equilibrium): If there exists a mapping $T_j : \boldsymbol{A_L} \to \boldsymbol{A_F}$ for each $j \in \boldsymbol{F}$, such that, for any fixed $a_i \in \boldsymbol{A_L}$*

$$\boldsymbol{U}_j\left(a_i, T_j\left(a_i\right), T_{-j}\left(a_i\right)\right) \geq \boldsymbol{U}_j\left(a_i, a_j, T_{-j}\left(a_i\right)\right) \quad (18)$$

where $\boldsymbol{U}_i = -f_i$, $i \in \boldsymbol{L}$; $\boldsymbol{U}_j = -f_j$, $j \in \boldsymbol{F}$. For all $a_j \in \boldsymbol{A_F}$, $T_{-j}\left(a_i\right) = \{T_m\left(a_i\right) \mid m \in \boldsymbol{F}, m \neq j\}$

If there exists a_i^* such that

$$\boldsymbol{U}_i\left(a_i^*, T_j\left(a_i^*\right), T_{-j}\left(a_i^*\right)\right) \geq \boldsymbol{U}_i\left(a_i, T_j\left(a_i\right), T_{-j}\left(a_i\right)\right) \quad (19)$$

$\left(a_i^*, a_j^*\right)$ *is called the Stackelberg-Nash equilibrium, where* $a_j^* = T_j\left(a_i^*\right), i \in \boldsymbol{L}, j \in \boldsymbol{F}$.

Equation (18) states that for a fixed action $a_i \in \mathbf{A_L}$, when other followers take the action $T_{-j}\left(a_i\right)$, the jth follower chooses its response action $T_j\left(a_i\right)$ with the maximum value of utility function $\mathbf{U}_j$. Then all the followers form a Nash equilibrium, i.e., $\left\{T_1\left(a_i\right), T_2\left(a_i\right), \ldots, T_p\left(a_i\right)\right\}, j = 1, 2, \ldots, p$. In this case, no follower j can benefit by unilaterally deviating from its best response $T_j\left(a_i\right)$.

Equation (19) represents a Stackelberg equilibrium of the leader, which depends on the response action of the Nash equilibrium composed of all followers. Since the leader knows the response action of the followers, the leader can choose an action a_i^* with the maximum value of $\mathbf{U}_i$. Finally, the Stackelberg-Nash equilibrium can be achieved.

Unlike the traditional one-follower model, the interaction between multiple followers affects the Stackelberg equilibrium. Therefore, the ordinal potential game is introduced to prove the existence of Nash equilibrium among followers. The definition of the ordinal potential game is given as follows.

Definition 2. *(Ordinal Potential Game):* $\boldsymbol{G_F} = (\boldsymbol{F}, \boldsymbol{A_F}, \boldsymbol{U_F})$ *is an ordinal potential game if there exists a function* φ

$$\boldsymbol{sgn}\left(\Delta\varphi\right) = \boldsymbol{sgn}\left(\Delta \boldsymbol{U}_j\right) \tag{20}$$

For any $j \in \mathbf{F}$, the leader action changes from a_i to a_i'. Then the jth follower action changes from $T_j\left(a_i\right)$ to $T_j\left(a_i'\right)$ with the actions of the other followers $T_{-j}\left(a_i\right)$ remain unchanged. $\Delta\varphi$ and $\Delta\mathbf{U}_j$ can be written as

$$\begin{aligned}\Delta\varphi = &\varphi\left(a_i, T_j\left(a_i\right), T_{-j}\left(a_i\right)\right) \\ &- \varphi\left(a_i', T_j\left(a_i'\right), T_{-j}\left(a_i\right)\right),\end{aligned} \tag{21}$$

$$\begin{aligned}\Delta\mathbf{U}_j = &\mathbf{U}_j\left(a_i, T_j\left(a_i\right), T_{-j}\left(a_i\right)\right) \\ &- \mathbf{U}_j\left(a_i', T_j\left(a_i'\right), T_{-j}\left(a_i\right)\right),\end{aligned} \tag{22}$$

where φ is an ordinal potential function. Therefore, for an ordinal potential game, the trend of the utility function is the same as the trend of the ordinal potential function.

The existence of the Stackelberg-Nash equilibrium is proven in Theorem 2.

Theorem 2. *There exists a Stackelberg-Nash equilibrium for the designed Stackelberg game model* $\boldsymbol{G} = (\boldsymbol{N}, \boldsymbol{A}, \boldsymbol{U})$.

Proof. See Appendix B. □

V. Reinforcement Learning-based Algorithm for Solving the Stackelberg game Model

In multi-agent reinforcement learning, each vehicle is considered as an agent, which takes action in the current state to get the next state and reward. The agent takes the next action with the new state and reward. However, the environment is often unpredictable in the real world, so model-free reinforcement learning is widely used in various fields. Q-learning is the most effective model-free reinforcement learning algorithm, which can be converged if a proper ϵ-greedy strategy is chosen [39].

A reinforcement learning-based algorithm with leader switching is designed to achieve the equilibrium point of the Stackelberg game model for the formation tracking problem.

For Q-learning reinforcement learning algorithms, the setting of the action set is important. For the formation tracking problem of the MUV system, the state transition equation of the ith vehicle can be written as

$$S_i(t+1) = S_i(t) + v_i(t)\eta$$

where $S_i(t)$ is the position vector of the ith vehicle at moment t, $v(t)$ is the velocity vector of the ith vehicle at moment t, η is the step size, and $S_i(t+1)$ is the position vector of the ith vehicle at moment $t+1$.

Thus, the velocity vector v_i is the action that the ith vehicle needs to select, which allows the vehicle to move to the next state. The velocity vector v_i can be represented in two ways, one can be decomposed into two velocity components:

$$v_i(t) = v_{ix}(t) + v_{iy}(t)$$

where $v_{ix}(t)$ is the velocity component of $v_i(t)$ in the x-coordinate, $v_{iy}(t)$ is the velocity component of $v_i(t)$ in the y-coordinate.

However, this approach will make the subsequent algorithm design complex. Therefore, we choose the second approach, a fixed velocity norm, by changing the directional angle and thus forming different motion vectors:

$$\begin{aligned}v_{ix}(t) &= \left|v_i(t)\right| cos\theta_i \\ v_{iy}(t) &= \left|v_i(t)\right| sin\theta_i\end{aligned}$$

where θ_i is the directional angle of the vehicle.

Therefore, the environment, state set, action set and reward set of the MUV system can be set as follows:

- Environment: A canvas is used to display the environment for multi-vehicle motion trajectories, as shown in Section VI.
- State set: The position of the vehicle in the environment, where the state set of the ith vehicle is S_i.
- Action set: Fixing the norm of the velocity, the directional angle is taken as an action set. Each of the four quadrants is divided into 15 actions, where A_1, A_2, A_3, and A_4 represent the set of actions in the first, second, third, and fourth quadrants, respectively. Note that the vehicle determines the quadrant after the quadrant predetermination, and then can only select the action in that quadrant. Thus limiting the steering rate of the vehicle.
- Reward set: Set the reward set $\mathbf{R}$ to the utility function $\mathbf{U}$.

A Q-learning algorithm with leader-switching is developed for solving the designed game model.

Theorem 3. *The leader-switching Q-learning algorithm can achieve equilibrium of the designed Stackelberg game.*

Proof. See Appendix C. □

Note that when the leader makes a decision, the followers will reach the position corresponding to the leader according to the desired formation. With this strategy, all followers will reach equilibrium, so the Stackelberg-Nash equilibrium can be reached.

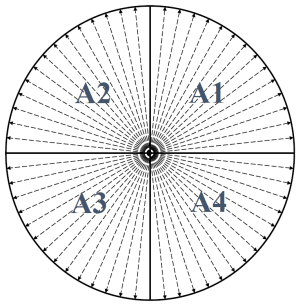


Fig. 4. Action set.

Algorithm 3 Leader-Switching Q-learning Algorithm.

Input:

γ: Discount factor;
α: Learning rate;
ε -greedy: Probabilistic action selection policy, where the action is selected by the optimal of the Q table with probability of ε and the probability of 1-ε is selected randomly;
A_1, A_2, A_3, A_4: Action set of the first, second, third, and fourth quadrants

Output: Q: Final updated Q table.

Initialize $Q(s, a)$
repeat (for each epsoid)
    optimal leader $\leftarrow$ LEADER-SWITCHING(l_0, T).
    Determine the leader in m quadrant, where m=1,2,3,4.
    Based on the current state and the Q table, the leader selects the action $a_i \in A_m$ by the ε-greedy policy.
    The leader announces the state action pair (s_i, a_i), and receives the state information from the followers.
    The jth follower makes the response action $T_j(a_i)$ by observing the environment and the leader's strategy and then arrives at the corresponding position in the desired formation.
    for $i \in$ **L do**
        Leader i makes the action a_i based on the reference follower response, get reward r
        $Q_i(s_i, a_i) \leftarrow Q_i(s_i, a_i)$
            $+\alpha\left[r + \gamma \max_{a_i'} Q_i(s_i', a_i') - Q_i(s_i, a_i)\right]$
    end for
    $s_i \leftarrow s_i^{'}$
end
until The leader's state s_i converges to the goal state p_r

VI. Simulation Experiments

In this section, the simulation experiments are given based on the PyCharm platform. The effectiveness of the EKF algorithm is first verified, then the effectiveness of the leader switching method is verified by comparative simulations. Finally, formation tracking of the MUV system with various shapes [40]–[42](e.g., triangle, quadrilateral, and trapezoid) is verified.

A. Simulation of EKF Algorithm for Target Trajectory Estimation

In this experiment, we set $l = 300$ and $w = 0.01\pi$ to get a circle with (-3, 0) as the center and 3 as the radius. So by changing the constants w and l, any circular trajectory can be obtained.

The parameters are set as $\Delta t = 0.01s$, $\sigma_1 = 0.1$, $\sigma_2 = 0.1$, $B = 0$, $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, thus obtaining the discrete state space expression. The measurement noise $v(k)$, which is white noise with mean 0, is then input to test the validity of Algorithm 1.

A larger Q means less confidence in the predicted values and more confidence in the measured values. A larger R implies that the EKF response will be slower. Better results are obtained by adjusting the Q and R matrices, as shown in Fig. 5 and Fig. 6. Fig. 5(a) shows the target point trajectory measured by the sensor. Fig. 5(b) shows the estimated target point trajectory after the EKF algorithm. Fig. 6(a) shows the measured and estimated values of the p_{rx}. Fig. 6(b) shows the measured and estimated values of the p_{ry}. It can be seen that the EKF algorithm accurately estimates the position of the target, which ensures the accuracy of the formation tracking problem.

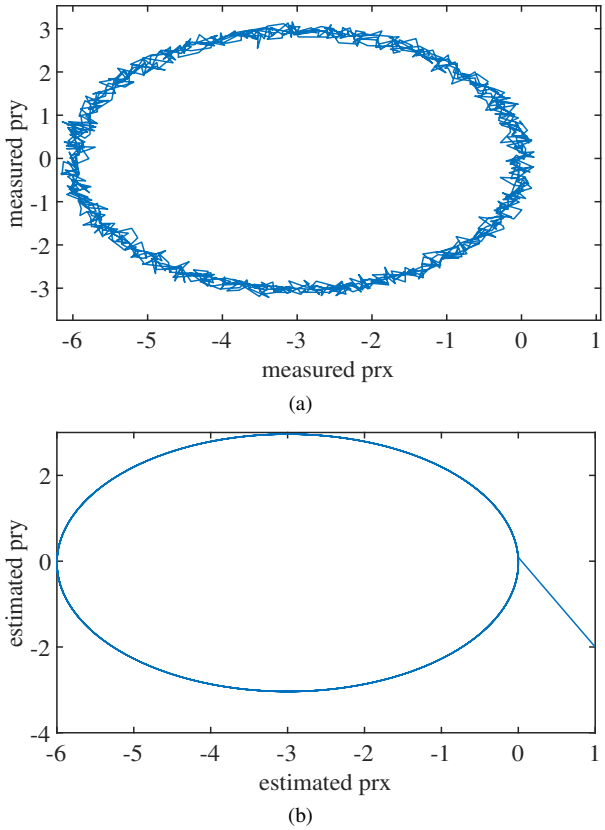


Fig. 5. Target point p_r. (a) Measurement trajectory of the target point. (b) Estimated trajectory of the target point.

B. Simulation of Leader-Switching Q-Learning Algorithm

Fig. 7 shows the trajectory of the formation tracking using the novel developed reinforcement learning-based algorithm. The blue diamond represents the target point. The green

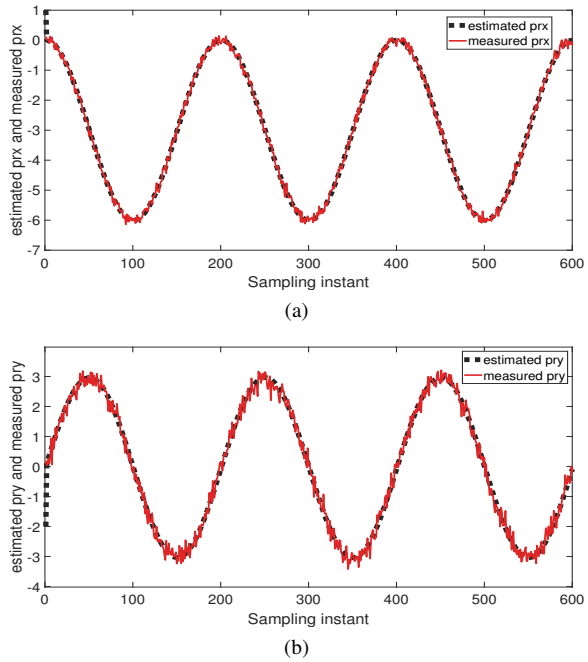


Fig. 6. Horizontal and vertical position of the target point. (a) Measurement trajectory and estimated trajectory of p_{rx}. (b) Measurement trajectory and estimated trajectory of p_{ry}.

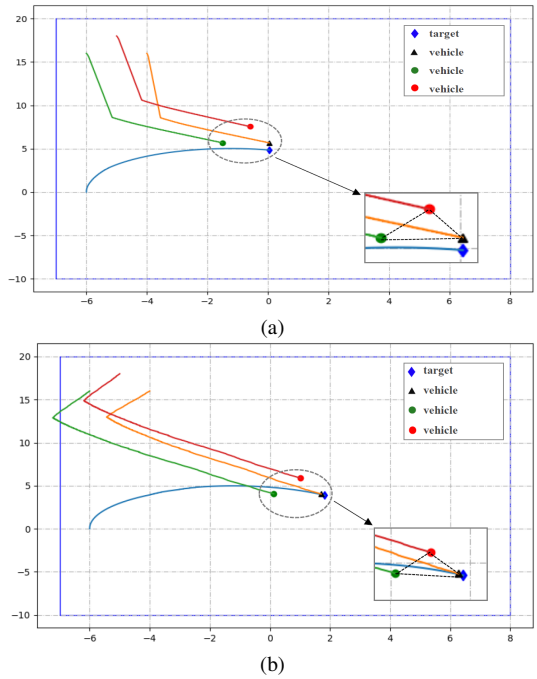


Fig. 8. Tracking trajectory under two algorithms. (a) Q-learning algorithm with leader switching. (b) Traditional Q-learning algorithm.

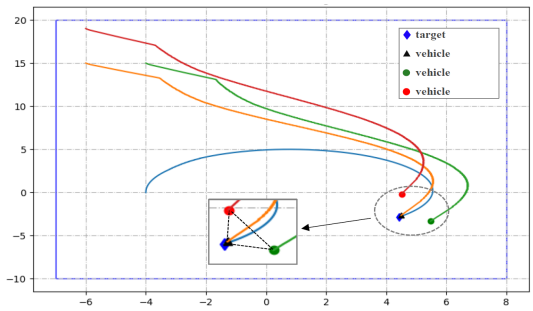


Fig. 7. Trajectory of MUV using novel developed algorithm for formation tracking

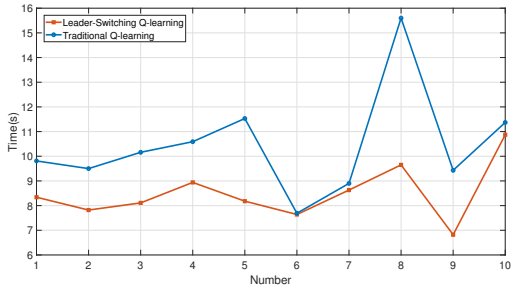
TABLE I
TRACKING TIME IN TWO CASES

Leader-Switching Q-learning	Traditional Q-learning
8.34s	9.81s
7.82s	9.50s
8.11s	10.16s
8.94s	10.59s
8.18s	11.53s
7.64s	7.69s
8.63s	8.90s
9.65s	15.60s
6.82s	9.43s
10.87s	11.37s

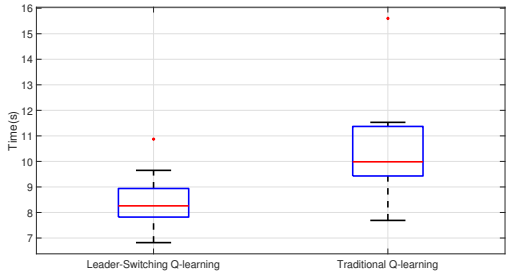
dot represents the initial leader. The black triangle and red dot represent the initial two followers. During the formation tracking, the vehicles in the followers appear more favorable to be the leader than the green dot, so the leader is switched. After the last leader switching, the black triangle represents the new leader, and the red and green dots represent the new two followers.

By comparing (a) and (b) in Fig. 8, it can be found that the target point can be tracked faster by the novel developed reinforcement learning-based algorithm, i.e., the leader-switching Q-learning algorithm. In Fig .8, the initial position of the target point is (-6,0); the initial position of the initial leader is (-6,16); the initial positions of the two initial followers are (-5,18) and (-4,16). Fig. 8(a) shows the simulation generated by the leader-switching Q-learning algorithm, tracking the target point at the position (0,5), and switching the leader once in total during the tracking process. Fig. 8(b) shows the simulation generated by the traditional Q-learning algorithm, where the black triangle is the fixed leader, and finally tracks the target point at position (2,4). It can be seen that Fig. 8(a) tracks the target point faster than Fig. 8(b).

The initial position of the target point is changed and several comparative simulations are performed. The formation tracking times of leader-switching Q-learning and traditional Q-learning are compared, as shown in Table I. Fig. 9 shows more specifically the effect of the two algorithms. Fig. 9(a) shows the tracking time of the leader-switching Q-learning algorithm and the traditional Q-learning algorithm under 10 experiments. The statistical comparison of the tracking times of the leader-switching Q-learning algorithm and the traditional Q-learning algorithm can be seen in Fig. 9(b). The red line represents the median. The upper blue and black lines represent the upper quartile and the upper limit of the tracking time. The lower blue and black lines represent the lower quartile and the lower limit of the tracking time, respectively. The red dots can be interpreted as outliers. Therefore, the leader-switching Q-learning algorithm works better than the traditional Q-learning algorithm.



(a)



(b)

Fig. 9. Comparison of the two algorithms. (a) The tracking times of the two algorithms are compared separately for 10 experiments. (b) Compare the two algorithms with the statistics of 10 experiments.

C. Simulation of Multiple Formation Shapes

In Fig. 10 and Fig. 11, we also implement a variety of shapes, such as quadrilateral and trapezoid formations, which are capable of tracking up dynamic nonlinear targets while switching leaders.

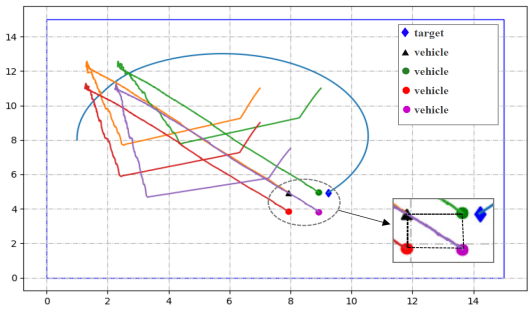


Fig. 10. Trajectories of quadrilateral formation tracking.

In Fig. 10, the blue diamond represents the target point. The red dot represents the initial leader. The black triangles, the green and purple dots represent the initial followers. During the formation tracking, other followers are in positions more conducive to tracking the target, so the leader was switched. After the last leader switching, the green dot represents the new leader. The black triangles, the red and purple dots represent the new followers.

In Fig. 11, the blue diamond represents the target point. The black triangle represents the initial leader. The red, green, and purple dots represent the initial followers. During the formation tracking, choosing the vehicle among the followers as the leader is more conducive to tracking the target, so the leader was switched. After leader switching, the green dot

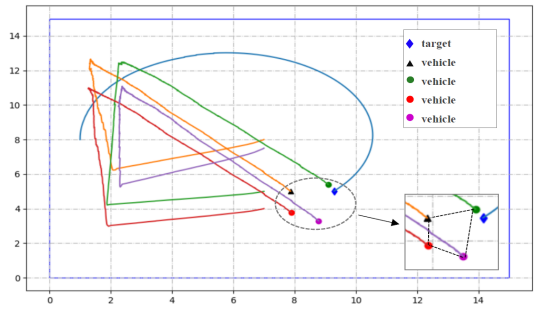


Fig. 11. Trajectories of trapezoidal formation tracking.

represents the new leader. The black triangles, the red and purple dots represent the new followers.

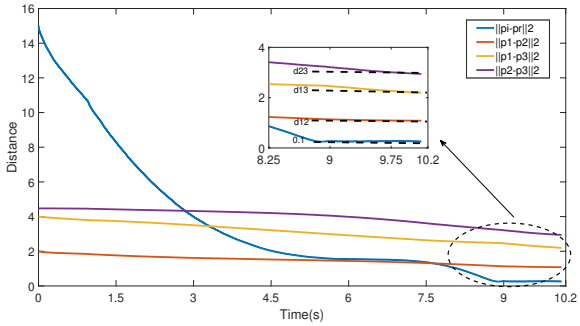


Fig. 12. Convergence of the distance between MUV.

The formation tracking process of the MUV system in Fig. 7 is selected for analysis. This includes the distance between the leader and the target as well as the distance between the vehicles of the MUV system. The blue line in Fig. 12 represents the distance between the leader and the target of the MUV system during the formation track. The red line in Fig. 12 represents the distance between the black triangle and the green dot in Fig. 7. The yellow line in Fig. 12 represents the distance between the black triangle and the red dot in Fig. 7. The purple line in Fig. 12 represents the distance between the green and red dots in Fig. 7.

The change of the blue line in Fig. 12 shows that the distance between the leader of the MUV system and the target converges to zero. Thus satisfying our desired Objective 2. The variations of the red, yellow, and purple lines in Fig. 12 converge to the given values $d12$, $d13$, and $d23$, respectively. Thus satisfying our desired Objective 3.

VII. Conclusions

In this paper, the formation tracking problem of the MUV system is discussed. An EKF algorithm is introduced to accurately estimate the state of the target. We establish a hierarchical Stackelberg game model for the formation tracking problem, which consists of a leader and multiple followers. The existence of the Stackelberg-Nash equilibrium of the designed game model is proven. A reinforcement learning-based algorithm for solving the game model is developed. The effectiveness of the designed method is verified by realizing

the formation of triangles, quadrilaterals, and trapezoids in four quadrants. By comparing the traditional Q-learning and designed leader-switching Q-learning algorithm, it can be found that the designed method can make the multi-vehicle system can earn a shorter formation tracking time of MUV.

APPENDIX A

Proof. The linear property of the model is evaluated by introducing the unknown matrices $\alpha(k)$ and $\beta(k)$ and constructing the equation:

$$\begin{aligned}\alpha(k)e(k) &= C\tilde{p}_r(k)^- \\ \tilde{p}_r(k)^- &= \beta(k-1)A(\hat{p}_r(k-1)^+)\tilde{p}_r(k-1)^+\end{aligned}$$

Use it instead of the approximation in the linear convergence analysis:

$$\begin{aligned}e(k) &\approx C\tilde{p}_r(k)^- \\ \tilde{p}_r(k)^- &\approx A(\hat{p}_r(k-1)^+)\tilde{p}_r(k-1)^+\end{aligned}$$

where $\alpha(k)$ and $\beta(k-1)$ are the introduced diagonal matrices, $e(k) = y(k) - C\hat{p}_r(k)^-$.

As described in Lemma 1,

$$\lim_{k\to\infty} V(k) = V$$

By means of (14), (15) and (16) in Lemma 2, it can be proven as follows:

$$\begin{aligned}\lim_{k\to\infty}\lambda_{min}((P(k)^+)^{-1}) = \infty &\overset{(14)}{\Longrightarrow} \lim_{k\to\infty} tr((P(k)^+)^{-1}) = \infty \\ &\overset{(15)}{\Longrightarrow} \lim_{k\to\infty}\frac{\lambda_{min}((P(k)^+)^{-1}(\tilde{p}_r(k)^+)^T\tilde{p}_r(k)^+)}{n\lambda_{max}((P(k)^+)^{-1})} = 0 \\ &\overset{(16)}{\Longrightarrow} \lim_{k\to\infty}\tilde{p}_r(k)^+ = 0 \Longrightarrow p_r(k) = \hat{p}_r(k)^+\end{aligned}$$

□

APPENDIX B

Proof. For leader i, if it changes its current behavior a_i to a_i', then follower j will change it's behavior $T_j(a_i)$ to $T_j(a_i')$, while other followers keep their behavior $T_{-j}(a_i)$ unchanged.

Design a potential function φ

$$\varphi = -\textstyle\sum_{j\in\mathbf{F}} = -\left(f_j + \sum_{p\in\mathbf{N}_j} e_{jp}f_p + \sum_{z\in\mathbf{N}_{j,p}} e_{jz}f_z\right)$$

Where $\mathbf{N}_j = \{p \in \mathbf{F} \mid p \neq j\}$, $\mathbf{N}_{j,p} = \{z \in \mathbf{F} \mid z \neq j, z \notin \mathbf{N}_j\}$

If the action of the jth follower changes from a_j to a_j', then the objective function changes from f_j to f_j', and from f_p to f_p'. f_z remains the same.

$$\Delta\varphi = -\left((f_j - f_j') + \textstyle\sum_{p\in N_j} e_{jp}f_p - \sum_{p\in N_j} e_{jp}f_p'\right) \tag{23}$$

Since information is interactive, the change in the objective function of jth follower is equal to the change in the sum of the objective functions of the neighbors with whom jth follower can interact with information, the details are as follows.

$$f_j - f_j' = \textstyle\sum_{p\in N_j} e_{jp}f_p - \sum_{p\in N_j} e_{jp}f_p' \tag{24}$$

Substitute (24) into (23) to get (25)

$$\Delta\varphi = -2\left(f_j - f_j'\right) \tag{25}$$

In Definition 1, $\mathbf{U}_j = -f_j, j \in \mathbf{F}$, (25) can be rewritten as (26)

$$\Delta\mathbf{U}_j = -\left(f_j - f_j'\right) \tag{26}$$

Combining (25) and (26)

$$2\Delta\mathbf{U}_j = \Delta\varphi \tag{27}$$

Equation (20) is satisfied, so the designed game is an ordinal potential game.

An ordinal potential game has a Nash equilibrium [33], [43]. Therefore, the problem can be established as an ordinal potential game between followers, and there exists at least one Nash equilibrium. If there is a Nash equilibrium for the followers, then the Stackelberg-Nash equilibrium of the system is also guaranteed. When all followers reach the Nash equilibrium, no follower can improve its utility by deviating its strategies. The leader makes a strategy to maximize its utility in the presence of Nash equilibrium. Therefore, in the Stackelberg-Nash equilibrium, neither the leader nor the followers can improve their utility by deviating their strategies. □

APPENDIX C

Proof. The convergence of Algorithm 3 can proven by using the three assumptions in Theorem 2 in [44].

It is known by Algorithm 3

$$\begin{aligned}&Q_i(s_i,a_i) \leftarrow \\ &(1-\alpha)\,Q_i(s_i,a_i) + \alpha\left[r + \gamma\max_{a_i'} Q_i(s_i',a_i')\right]\end{aligned} \tag{28}$$

Subtracting $Q_i^*(s_i,a_i)$ from both sides of (28) and making $\Delta t = Q_i(s_i,a_i) - Q_i^*(s_i,a_i)$, then (28) can be rewritten as

$$\begin{aligned}&\Delta t \leftarrow \\ &(1-\alpha)\,\Delta t + \alpha\left[r + \gamma\max_{a_i'} Q_i(s_i',a_i') - Q_i^*(s_i,a_i)\right]\end{aligned}$$

Firstly, α is the learning rate and satisfies $0 \le \alpha \le 1$, so that Assumption 1 is satisfied.

Secondly, Assumption 2 can be proven. Set F as follows

$$F(s_i,a_i) = r + \gamma\max_{a_i'} Q_i(s_i',a_i') - Q_i^*(s_i,a_i)$$

The probability of the ith leader will convert from s_i to s_i' as $P_{a_i}(s_i,s_i')$

$$\begin{aligned}&\mathbb{E}(F) \\ &= \textstyle\sum_{s_i'\in S_i} P_{a_i^*}(s_i,s_i')\left[r + \gamma\max_{a_i'} Q_i(s_i',a_i') - Q_i^*(s_i,a_i)\right] \\ &= (\mathbf{H}Q_i)(s_i,a_i) - Q_i^*(s_i,a_i) \\ &= (\mathbf{H}Q_i)(s_i,a_i) - (\mathbf{H}Q_i^*)(s_i,a_i)\end{aligned} \tag{29}$$

where $\mathbf{H}$ is the operator.

For the simplicity of the subsequent proof, simplify (29) to

$$\mathbb{E}(F) = \mathbf{H}Q_i - \mathbf{H}Q_i^*$$

Then

$$\begin{aligned}
&\|\mathbf{H}Q_i - \mathbf{H}Q_i^*\|_\infty \\
&= \max \left| \textstyle\sum_{s_i' \in S_i} P_{a_i} \left[r + \gamma \max_{a_i'} Q_i - (r + \gamma \max_{a_i'} Q_i^*) \right] \right| \\
&= \max \quad \gamma \left| \textstyle\sum_{s_i' \in S_i} P_{a_i} \left[\max_{a_i'} Q_i - \max_{a_i'} Q_i^* \right] \right| \\
&\leq \max \quad \gamma \textstyle\sum_{s_i' \in S_i} P_{a_i} \left| \max_{a_i'} Q_i - \max_{a_i'} Q_i^* \right| \\
&\leq \max \quad \gamma \textstyle\sum_{s_i' \in S_i} P_{a_i} \quad \max_{a_i'} |Q_i - Q_i^*| \\
&= \max \quad \gamma \textstyle\sum_{s_i' \in S_i} P_{a_i} \|Q_i - Q_i^*\|_\infty \\
&= \gamma \|Q_i - Q_i^*\|_\infty = \gamma \|\Delta t\|_\infty
\end{aligned}$$

where γ is the decay rate and satisfies $0 < \gamma < 1$, so that Assumption 2 is satisfied.

Finally, Assumption 3 can be proven. Set the variance of F as follows

$$\begin{aligned}
&\mathrm{Var}(F) \\
&= \mathbb{E}\left[(F - \mathbb{E}(F))^2\right] \\
&= \mathbb{E}\left[(r + \gamma \max_{a_i'} Q_i - Q_i^* - (\mathbf{H}Q_i - Q_i^*))^2\right] \\
&= \mathbb{E}\left[(r + \gamma \max_{a_i'} Q_i - \mathbf{H}Q_i)^2\right] \\
&= \mathrm{Var}(r + \gamma \max_{a_i'} Q_i)
\end{aligned}$$

where the reward r is bounded and $0 < \gamma < 1$, so $Var(F)$ is bounded and therefore satisfies Assumption 3.

By Theorem 2 in [44], Δt converges to 0 if three assumptions are satisfied, i.e., Q_i converges to Q_i^*. □

REFERENCES

[1] Z. Qu, *Cooperative control of dynamical systems: applications to autonomous vehicles*. Springer Science & Business Media, 2009.

[2] T. Suzuki, R. Usami, and T. Maekawa, "Automatic two-lane path generation for autonomous vehicles using quartic b-spline curves," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 547–558, 2018.

[3] T. Z. Muslimov and R. A. Munasypov, "Adaptive decentralized flocking control of multi-uav circular formations based on vector fields and backstepping," *ISA Transactions*, vol. 107, pp. 143–159, 2020.

[4] J. Yu, X. Dong, Q. Li, J. Lv, and Z. Ren, "Distributed adaptive cooperative time-varying formation tracking guidance for multiple aerial vehicles system," *Aerospace Science and Technology*, vol. 117, p. 106925, 2021.

[5] C. Zu, C. Yang, J. Wang, W. Gao, D. Cao, and F.-Y. Wang, "Simulation and field testing of multiple vehicles collision avoidance algorithms," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1045–1063, 2020.

[6] M. A. Kamel, X. Yu, and Y. Zhang, "Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review," *Annual Reviews in Control*, vol. 49, pp. 128–144, 2020.

[7] M. Khaledyan, T. Liu, V. Fernandez-Kim, and M. de Queiroz, "Flocking and target interception control for formations of nonholonomic kinematic agents," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 4, pp. 1603–1610, 2020.

[8] D. Meltz and H. Guterman, "Functional safety verification for autonomous ugvs—methodology presentation and implementation on a full-scale system," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 472–485, 2019.

[9] J. Lu, Q. Wei, and F.-Y. Wang, "Parallel control for optimal tracking via adaptive dynamic programming," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 6, pp. 1662–1674, 2020.

[10] X. Jin, "Nonrepetitive leader–follower formation tracking for multiagent systems with los range and angle constraints using iterative learning control," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1748–1758, 2019.

[11] H. Liu, Q. Meng, F. Peng, and F. L. Lewis, "Heterogeneous formation control of multiple UAVs with limited-input leader via reinforcement learning," *Neurocomputing*, vol. 412, pp. 63–71, 2020.

[12] H. Liu, F. Peng, H. Modares, and B. Kiumarsi, "Heterogeneous formation control of multiple rotorcrafts with unknown dynamics by reinforcement learning," *Information Sciences*, vol. 558, pp. 194–207, 2021.

[13] G. Franzè, W. Lucia, and A. Venturino, "A distributed model predictive control strategy for constrained multi-vehicle systems moving in unknown environments," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 343–352, 2021.

[14] M. Baradaran Khalkhali, A. Vahedian, and H. Sadoghi Yazdi, "Multi-target state estimation using interactive kalman filter for multi-vehicle tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1131–1144, 2020.

[15] F. Jiang, J. Chen, and A. L. Swindlehurst, "Optimal power allocation for parameter tracking in a distributed amplify-and-forward sensor network," *IEEE Transactions on Signal Processing*, vol. 62, no. 9, pp. 2200–2211, 2014.

[16] S. R. Jondhale and R. S. Deshpande, "GRNN and KF framework based real time target tracking using psoc ble and smartphone," *Ad Hoc Networks*, vol. 84, pp. 19–28, 2019.

[17] X. Wang, M. Fu, and H. Zhang, "Target tracking in wireless sensor networks based on the combination of KF and MLE using distance measurements," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 567–576, 2012.

[18] Z. Song, "Convergence analysis of the extended kalman filter for nonlinear random discrete time systems," *Control Theory Applications*, vol. 17, no. 2, pp. 264–269, 2000.

[19] S. Patterson, "In-network leader selection for acyclic graphs," in *2015 American Control Conference (ACC)*, 2015, pp. 329–334.

[20] Y. Cao, L. Zhang, C. Li, and M. Z. Q. Chen, "Observer-based consensus tracking of nonlinear agents in hybrid varying directed topology," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 2212–2222, 2017.

[21] Y. Wu, S. Zhuang, C. K. Ahn, and W. Li, "Aperiodically intermittent discrete-time state observation noise for consensus of multiagent systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[22] E. Mackin and S. Patterson, "Submodular optimization for consensus networks with noise-corrupted leaders," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 3054–3059, 2019.

[23] J. Liu, Y. Zhang, Y. Yu, and C. Sun, "Fixed-time leader–follower consensus of networked nonlinear systems via event/self-triggered control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 5029–5037, 2020.

[24] J. Liu, Y. Zhang, C. Sun, and Y. Yu, "Fixed-time consensus of multi-agent systems with input delay and uncertain disturbances via event-triggered control," *Information Sciences*, vol. 480, pp. 261–272, 2019.

[25] A. Franchi and P. Robuffo Giordano, "Online leader selection for improved collective tracking and formation maintenance," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 3–13, 2018.

[26] A. Franchi, P. R. Giordano, and G. Michieletto, "Online leader selection for collective tracking and formation control: The second-order case," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 4, pp. 1415–1425, 2019.

[27] F. Li, Y. Ding, M. Zhou, K. Hao, and L. Chen, "An affection-based dynamic leader selection model for formation control in multirobot systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1217–1228, 2017.

[28] L. Xue and X. Cao, "Leader selection via supermodular game for formation control in multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3656–3664, 2019.

[29] J. Shamma, *Cooperative control of distributed multi-agent systems*. John Wiley & Sons, 2008.

[30] M. I. Abouheaf, F. L. Lewis, K. G. Vamvoudakis, S. Haesaert, and R. Babuska, "Multi-agent discrete-time graphical games and reinforcement learning solutions," *Automatica*, vol. 50, no. 12, pp. 3038–3053, 2014.

[31] C. Mu, J. Peng, and C. Sun, "Hierarchical multiagent formation control scheme via actor-critic learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.

[32] M. Li, J. Qin, N. M. Freris, and D. W. C. Ho, "Multiplayer stackelberg-nash game for nonlinear system via value iteration-based integral reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2020.

[33] A. A. Kulkarni and U. V. Shanbhag, "An existence result for hierarchical stackelberg v/s stackelberg games," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3379–3384, 2015.

[34] Y. Zhang, Y. Xu, Y. Xu, Y. Yang, Y. Luo, Q. Wu, and X. Liu, "A multi-leader one-follower stackelberg game approach for cooperative anti-jamming: No pains, no gains," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1680–1683, 2018.

[35] Z. Zhou and H. Xu, "Decentralized optimal large scale multi-player pursuit-evasion strategies: A mean field game approach with reinforcement learning," *Neurocomputing*, 2021.
[36] L. Xue, C. Sun, D. Wunsch, Y. Zhou, and F. Yu, "An adaptive strategy via reinforcement learning for the prisoner dilemma game," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 301–310, 2018.
[37] C. Mu, K. Wang, and C. Sun, "Policy-iteration-based learning for nonlinear player game systems with constrained inputs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 6488–6502, 2021.
[38] X. Yuan, Y. Wang, J. Liu, and C. Sun, "Action mapping: A reinforcement learning method for constrained-input systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
[39] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *Robotica*, vol. 17, no. 2, pp. 229–235, 1999.
[40] B. Das, B. Subudhi, and B. B. Pati, "Cooperative formation control of autonomous underwater vehicles: An overview," *International Journal of Automation and computing*, vol. 13, no. 3, pp. 199–225, 2016.
[41] J. Zhang, J. Yan, and P. Zhang, "Multi-UAV formation control based on a novel back-stepping approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2437–2448, 2020.
[42] Z. Peng, W. Wu, D. Wang, and L. Liu, "Coordinated control of multiple unmanned surface vehicles: recent advances and future trends," *Chinese Journal of Ship Research*, vol. 16, no. 1, pp. 51–64, 2021.
[43] N. Li and J. R. Marden, "Designing games for distributed optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 230–242, 2013.
[44] F. S. Melo, "Convergence of q-learning: A simple proof," *Institute of Systems and Robotics*, pp. 1–4, 2001.

Lei Xue (Member, IEEE) received his Ph.D. degree in control science and engineering from Southeast University, Nanjing, China, in 2017.

From September 2013 to September 2014, he was a Visiting Ph.D. student with Applied Computational Intelligence Laboratory, Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO, USA. Currently he is an Associate Professor with School of Automation, Southeast University, Nanjing, China. His research interests include game theory, multi-agent system, and optimization control.

Bei Ma was born in 1999. She is currently pursuing her M.S. degree with the School of Automation, Southeast University, Nanjing, China. Her current research interests include game theory, reinforcement learning, and multiagent systems.

Jian Liu (Member, IEEE) received his B.S. and Ph.D. degree from the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China, in 2015 and 2020, respectively.

From September 2017 to September 2018, he was a joint training student with the Department of Mathematics, Dartmouth College, Hanover, NH, USA. From 2020 to 2021, he was a Postdoctoral Fellow with the School of Automation, Southeast University, Nanjing, China, where he is currently an Associate Professor. His current research interests include multi-agent systems, nonlinear control, event-triggered control, fixed-time control, intermittent control.

Chaoxu Mu (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the School of Automation, Southeast University, Nanjing, China, in 2012.

She was a Visiting Ph.D. Student with the Royal Melbourne Institute of Technology University, Melbourne, VIC, Australia, from 2010 to 2011. She was a Post-Doctoral Fellow with the Department of Electrical, Computer and Biomedical Engineering, The University of Rhode Island, Kingston, RI, USA, from 2014 to 2016. She is currently a Professor with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. Her current research interests include nonlinear system control and optimization, and adaptive and learning systems.

Donald C. Wunsch (Fellow, IEEE) received the B.S. degree in applied mathematics from the University of New Mexico, Albuquerque, NM, USA, in 1984, the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, WA, USA, in 1987 and 1991, respectively, the M.B.A. degree in executive from Washington University in St. Louis, St. Louis, MO, USA, and the Jesuit Core Honors Program from Seattle University, Seattle, in 2006.

He is the Mary K. Finley Missouri Distinguished Professor with the Missouri University of Science and Technology (Missouri S&T), Rolla, MO, USA. He is the Director of the Applied Computational Intelligence Laboratory, a multidisciplinary research group. He was with Texas Tech University, Lubbock, TX, USA; Boeing, Seattle, WA, USA; Rockwell International, Kirtland AFB, Albuquerque, NM, USA; and the International Laser Systems, Kirtland AFB. He has produced 20 Ph.D. recipients in Computer Engineering, Electrical Engineering, Systems Engineering, and Computer Science. He has attracted over $10 million in sponsored research and has over 450 publications including nine books. He has over 15000 citations. His current research interests include clustering/unsupervised learning, biclustering, adaptive resonance and reinforcement learning architectures, hardware and applications, neurofuzzy regression, traveling salesman problem heuristics, games, robotic swarms, and bioinformatics.

Dr. Wunsch was a recipient of the NSF CAREER Award and the 2015 INNS Gabor Award. He served as the IJCNN General Chair, and on several boards, including the St. Patrick's School Board, IEEE Neural Networks Council, International Neural Networks Society, and the University of Missouri Bioinformatics Consortium, Chaired the Missouri S&T Information Technology and Computing Committee as well as the Student Design and Experiential Learning Center Board. He was an INNS President and an INNS Fellow.