

An Adaptive Fuzzy Reinforcement Learning Cooperative Approach for the Autonomous Control of Flock Systems

Shuzheng Qu, Mohammed Abouheaf, Wail Gueaieb, and Davide Spinello

Abstract—The flock-guidance problem enjoys a challenging structure where multiple optimization objectives are solved simultaneously. This usually necessitates different control approaches to tackle various objectives, such as guidance, collision avoidance, and cohesion. The guidance schemes, in particular, have long suffered from complex tracking-error dynamics. Furthermore, techniques that are based on linear feedback strategies obtained at equilibrium conditions either may not hold or degrade when applied to uncertain dynamic environments. Pre-tuned fuzzy inference architectures lack robustness under such unmodeled conditions. This work introduces an adaptive distributed technique for the autonomous control of flock systems. Its relatively flexible structure is based on online fuzzy reinforcement learning schemes which simultaneously target a number of objectives; namely, following a leader, avoiding collision, and reaching a flock velocity consensus. In addition to its resilience in the face of dynamic disturbances, the algorithm does not require more than the agent position as a feedback signal. The effectiveness of the proposed method is validated with two simulation scenarios and benchmarked against a similar technique from the literature.

I. INTRODUCTION

The increasing complexity and diversity of modern robotic fields have raised new challenges and imposed limitations on various multi-agent robotic applications. For instance, many of such systems lack the autonomy required to control the collective behavior of a fleet of self-driving vehicles on a highway. Distributed control paradigms thrive in this domain [1], [2]. They provide an alternative solution based on relaxing the inter-dependency requirements among the agents, such as the communication range of each agent, the amount of information required about its surrounding agents and the environment, etc. A key aspect about any distributed controller is its ability to optimize a set of simultaneously conflicting local and team objectives. The underlying control methods are often based on theories of feedback mechanisms and communication graphs [3], [4].

The behavior of large numbers of interacting agents, such as flocks of flying birds and fish schools, is used in the study the cooperative dynamical systems [5], [6]. Sliding mode methods are employed to achieve coordination of multi-agent systems in [7], [8]. A neighbor-based tracking control scheme with distributed estimators is applied to advise distributed tracking strategies in [9]. Relevant control approaches relied on the existence of a virtual leader or

a moving target in mobile sensor networks [10], [11]. A common shortcoming to all these techniques is their dependence on the prior knowledge of a system model that is free of unstructured uncertainties. A distributed adaptive strategy that is based on a backstepping scheme is adopted to solve a consensus control problem in [12]. The tracking problem of fractional-order multi-agent systems is addressed in [13] using adaptive strategies with neural network approximators.

Reinforcement Learning (RL) is a machine learning tool that allows agents to learn the best strategies to solve certain problems upon interacting with unknown environments [14], [15]. They do so by tracing the usefulness of being at a certain state while following some action in order maximize a cumulative reward criterion to eventually reach a target state. RL techniques offer flexible adaptations while interacting within complex environments [16]. They are usually implemented using either a Value Iteration or a Policy Iteration algorithm. A distributed fusion-based search strategy is developed to expedite the locomotion learning process for robots in [17]. RL has helped with various robotic applications, such as unmanned flexible wing aircraft [18], [19], autonomous helicopters [20], crowd aware robot navigation [21], multi-robot predator avoidance [22], guidance for biped humanoid robots [23], multi-robot collision avoidance [24], and efficient driving systems [25]. RL-based path-following control mechanisms for unmanned surface vehicles and unmanned aerial vehicles are presented in [26], [27]. To help approximate the unknown value function and the associated optimal strategy, actor-critic neural networks are usually utilized [14]. The actor-critic structures have been adopted for cooperative control problems, such as graphical games and mobile sensor networks [28]–[30]. Despite their high potential in learning how to interact with ill-defined environments, RL schemes suffer from a major shortcoming stemming from the discrete (non-smooth) nature of their action space, which may lead to a “jerky” behavior when they are applied to control a real-world system. However, this is possible to solve by combining them with a fuzzy logic engine, taking advantage of its nonlinear approximation ability, to get a fuzzy RL algorithm with a continuous output space.

Fuzzy logic owes its popularity to its ability to incorporate human-like expertise in controlling complex systems in a model-free fashion. It does so by defining the acquired meta knowledge about the system’s functionality in terms of linguistic relations [31]. Fuzzy logic is employed to identify and control nonlinear dynamical systems in [32]. Relevant fuzzy schemes have been applied in many processes, like

This work was partially supported by NSERC Grant EGP 537568-2018. Shuzheng Qu, Mohammed Abouheaf, and Wail Gueaieb are with the School of Electrical Engineering & Computer Science, while Davide Spinello is with the Department of Mechanical Engineering, University of Ottawa, Ottawa, Canada. E-mail: {fqu096,mabouhea,wgueaieb,dspinell}@uottawa.ca.

temperature control [33], autonomous wheeled robot navigation [34], and vehicle guidance [35]. They can be augmented with neural networks to automate their parameter-tuning process. For example, a neurofuzzy algorithm is designed to control a robotic manipulator in [36]. Fuzzy schemes are used to guide mobile robots to follow a leader and to avoid obstacles in [37], [38]. A collision avoidance approach based on an extended Takagi-Sugeno-Kang (TSK) inference system is proposed for a flock system in [4].

This work contributes an online fuzzy RL process that guides the motion of a flock. This is done while simultaneously compromising between competing local and team objectives. This architecture benefits novel model-free collision-avoidance and tracking mechanisms. It addresses concerns related to the dependence of the guidance control laws on prior knowledge about the dynamical models of the agents or the environment [3], [39], [40]. Further, it enables online tuning of the fuzzy inference system without relying on a computationally exhaustive fuzzy Q-Learning search process. The rest of the paper is organized as follows: Section II formulates the problem and introduces a high level of the proposed control structure. The tracking, collision avoidance, and consensus control policies are detailed in Sections III, IV, and V, respectively. Section VI presents and discusses the simulation results. Finally, Section VII offers some concluding remarks.

II. PROBLEM FORMULATION AND CONTROL STRUCTURE

We now formally define the cooperative control problem of a flock of N agents communicating over a fully connected undirected graph. The goal is to make the flock follow a leader while satisfying a set of constraints. The leader does not have to be a physical agent. It may represent a virtual time-dependent signal or trajectory that would command the agents behavior. The problem is casted in a trajectory-tracking or pursuer-evader framework. The mobile agents are set to navigate in a 2D plane. The motion of each agent i is governed by

$$\mathbf{p}_{k+1}^i = \mathbf{p}_k^i + T \mathbf{q}_k^i \quad \mathbf{q}_{k+1}^i = \mathbf{q}_k^i + T \mathbf{u}_k^i$$

where $\mathbf{p}_k^i = [x_k^i \ y_k^i]^T \in \mathbb{R}^2$ and $\mathbf{q}_k^i = [v_k^{i,x} \ v_k^{i,y}]^T \in \mathbb{R}^2$ denote the position and linear velocity, respectively, of agent $i \in \{1, 2, \dots, N\}$ at discrete-time index $k \in \mathbb{N}$, T is the sampling time, and $\mathbf{u}_k^i = [u_k^{i,x} \ u_k^{i,y}]^T \in \mathbb{R}^2$ is the control signal commanding the acceleration in the x- and y-directions. The control signals of all the agents are calculated by the proposed control algorithm, except the one for the leader $\ell \in \{1, \dots, N\}$, which is generated by an independent command generator. The location-measurements of the flock are set to be available for each agent.

The aim of the flock is to satisfy the following objectives simultaneously: 1) track the leader; 2) keep a safe distance among the agents; and 3) reach a velocity consensus. The latter requirement involves an information layer among the agents. The agents have to manage the occasionally conflicting nature of such goals to reach the best possible compromise. They collectively search for a decision making

policy that balances the local and team objectives. The local objective for each agent is to avoid colliding with its neighbors while staying close to the flock's focus area; while the team objectives involve reaching consensus on velocity for the flock and successfully following the leader. Thus, the above three objectives can be formally described by the following respective relations for each pursuer $i \in \{1, \dots, N\} \setminus \{\ell\}$:

$$\lim_{k \rightarrow \infty} \|\mathbf{p}_k^i - \mathbf{p}_k^\ell\| = 0 \quad (1a)$$

$$\lim_{k \rightarrow \infty} |\zeta_k^i - \zeta_k^j| \geq d, \quad \forall \zeta \in \{x, y\}, \quad \forall j \in \mathcal{N}_i \quad (1b)$$

$$\lim_{k \rightarrow \infty} v_k^{i,\zeta} = v_k^{i,*} \quad (1c)$$

where \mathcal{N}_i denotes the set of neighboring agents to agent i , d is a safety distance, $v_k^{i,*}$ is a consensus speed, and $\zeta \in \{x, y\}$ refers to either the x or y component.

To satisfy these objectives, the control command of each pursuer $i \in \{1, \dots, N\} \setminus \{\ell\}$ is formulated as an aggregate of three auxiliary signals

$$\mathbf{u}_k^{i,\zeta} = \mathbf{u}_{t,k}^{i,\zeta} + \mathbf{u}_{s,k}^{i,\zeta} + \mathbf{u}_{c,k}^{i,\zeta}. \quad (2)$$

The first term $\mathbf{u}_{t,k}^{i,\zeta}$ is the tracking or navigation control signal. It is defined as a function of the agent's and leader's positions, $\mathbf{u}_{t,k}^{i,\zeta} = f_{t,\zeta}^i(\zeta_k^i, \zeta_k^\ell)$, which handles the conflicts pertaining to these positions. The separation or collision avoidance control of agent i , $\mathbf{u}_{s,k}^{i,\zeta}$, is applied to avoid colliding with agents j in its neighborhood \mathcal{N}_i in the ζ direction. It depends on the agent's position relative to its neighbors and is set as

$$\mathbf{u}_{s,k}^{i,\zeta} = f_{s,\zeta}^i(\zeta_k^i, \zeta_k^j) = \frac{\sum_{j \in \mathcal{N}_i} \mathbf{u}_{s,k}^{ij,\zeta}}{|\mathcal{N}_i|} \quad (3)$$

where $|\mathcal{N}_i|$ denotes the cardinality of \mathcal{N}_i and $\mathbf{u}_{s,k}^{ij,\zeta}$ refers to the opinion or partial separation control signal taken by agent i due to each agent $j \in \mathcal{N}_i$. Finally, the synchronization goal is to guide all the pursuers to reach consensus on a common velocity. This is achieved by a consensus protocol $\mathbf{u}_{c,k}^{i,\zeta} = f_{c,\zeta}^i(v_k^{i,\zeta}, v_k^{j,\zeta})$, $\forall j \in \mathcal{N}_i$, which is defined in terms of the velocities of each agent and its neighbors.

This framework can be employed in many useful applications, including UAV monitoring and surveillance, area coverage, and mobile sensor networks, to name a few. This work does not consider the limitations imposed by the time-varying graph topologies.

III. TRACKING CONTROL MECHANISM

The section addresses the computation of the tracking control signal $\mathbf{u}_{t,k}^{i,\zeta}$.

A. Optimization Framework

Let $\mathbf{E}_k^{i,\zeta} = [\zeta_k^i - \zeta_k^\ell, \zeta_{k-1}^i - \zeta_{k-1}^\ell, \zeta_{k-2}^i - \zeta_{k-2}^\ell]^T \in \mathbb{R}^3$ be an error signal vector corresponding to agent i at time index k , representing a time window of the recent tracking error measurements between the agent's position and that of the leader. The choice of the number of tracking error instances depends on the complexity of the problem to solve.

In our case, a time window of three components provided a satisfactory compromise between accuracy and computational complexity. In order for each agent i to measure the quality of its tracking control strategy $u_{t,k}^{i,\zeta}$, a performance index $\chi_0^{i,\zeta} = \sum_{k=0}^{\infty} U_k^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta})$ is proposed, where $U_k^{i,\zeta}$ is an objective function that is designed to be quadratic and convex

$$U_k^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta}) = \frac{1}{2} \left[\mathbf{E}_k^{i,\zeta T} \mathbf{Q}^i \mathbf{E}_k^{i,\zeta} + R^i (u_{t,k}^{i,\zeta})^2 \right] \quad (4)$$

where $\mathbf{0} < \mathbf{Q}^i \in \mathbb{R}^{3 \times 3}$ and $0 < R^i \in \mathbb{R}$ are some weighting factors corresponding to the tracking error vector and the control signal, respectively. When applied to matrices, the notations “ $> \mathbf{0}$ ” and “ $\geq \mathbf{0}$ ” denote positive definite and positive semi-definite matrices, respectively.

Motivated by the structure of the performance index $\chi^{i,\zeta}$ and cost function (4), a solving value function $V^{i,\zeta}$ is formulated as

$$V^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta}) = \frac{1}{2} \begin{bmatrix} \mathbf{E}_k^{i,\zeta T} & u_{t,k}^{i,\zeta} \end{bmatrix} \mathbf{H}^{i,\zeta} \begin{bmatrix} \mathbf{E}_k^{i,\zeta} \\ u_{t,k}^{i,\zeta} \end{bmatrix}$$

such that, $\mathbf{H}^{i,\zeta} \equiv \begin{bmatrix} \mathbf{H}_{\mathbf{E}_k^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} & \mathbf{H}_{\mathbf{E}_k^{i,\zeta} u_{t,k}^{i,\zeta}}^{i,\zeta} \\ \mathbf{H}_{u_{t,k}^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} & \mathbf{H}_{u_{t,k}^{i,\zeta} u_{t,k}^{i,\zeta}}^{i,\zeta} \end{bmatrix} \in \mathbb{R}^{4 \times 4}$

where $\mathbf{H}^{i,\zeta} > \mathbf{0}$, $\mathbf{H}_{u_{t,k}^{i,\zeta} u_{t,k}^{i,\zeta}}^{i,\zeta} \in \mathbb{R}$, and $\mathbf{H}_{u_{t,k}^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} \in \mathbb{R}^{1 \times 3}$. This results in the following temporal difference (Bellman) equation

$$V^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta}) = U_k^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta}) + V^{i,\zeta}(\mathbf{E}_{k+1}^{i,\zeta}, u_{t,k+1}^{i,\zeta}).$$

Applying the optimality principle by taking $\arg \min_{u_{t,k}^{i,\zeta}} (V^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta}))$ yields the optimal strategy-to-follow that is given by

$$u_{t,k}^{i,\zeta(o)} = - \left(\mathbf{H}_{u_{t,k}^{i,\zeta} u_{t,k}^{i,\zeta}}^{i,\zeta} \right)^{-1} \mathbf{H}_{u_{t,k}^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} \mathbf{E}_k^{i,\zeta}. \quad (5)$$

Employing the optimal policy in Bellman equation yields the Bellman optimality relation

$$V^{i,\zeta(o)}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta(o)}) = U_k^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta(o)}) + V^{i,\zeta(o)}(\mathbf{E}_{k+1}^{i,\zeta}, u_{t,k+1}^{i,\zeta(o)}) \quad (6)$$

This equation is solved simultaneously by each agent i so that the agents can eventually converge to optimized tracking strategies.

In this work, a two-step technique, known as Value Iteration, is applied to concurrently solve Bellman optimality (6) using optimal policy (5) as follows:

$$V^{i,\zeta(r+1)}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta}) = U_k^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, u_{t,k}^{i,\zeta(r)}) + V^{i,\zeta(r)}(\mathbf{E}_{k+1}^{i,\zeta}, u_{t,k+1}^{i,\zeta(r)})$$

$$u_{t,k}^{i,\zeta(r+1)} = - \left(\mathbf{H}_{u_{t,k}^{i,\zeta} u_{t,k}^{i,\zeta}}^{i,\zeta(r)} \right)^{-1} \mathbf{H}_{u_{t,k}^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta(r)} \mathbf{E}_k^{i,\zeta}$$

till convergence [14]. This procedure leads to a converging nondecreasing sequence of solving value functions $0 \leq V^{i,\zeta(0)} \leq V^{i,\zeta(1)} \leq \dots \leq V^{i,\zeta(r)} \leq \dots \leq V^{i,\zeta(o)}$ [41].

B. Neural Network Approximation

An actor-critic approach is employed by each agent to execute the Value Iteration process. The actor is realized by a neural network approximator to estimate the optimal tracking strategy (5). Another neural network is designed to implement the critic which approximates the optimal value function defined in (6).

The structures of the actor and critic approximators are motivated by the those of the optimal policy $u_{t,k(o)}^{i,\zeta}$ and optimal value function $V^{i,\zeta(o)}$. To this end, the optimal policy is approximated as $\hat{u}_{t,k}^{i,\zeta} = \omega^{i,\zeta} \mathbf{E}_k^{i,\zeta}$, where $\omega^{i,\zeta} \in \mathbb{R}^{1 \times 3}$ is a row vector of the actor approximation weights. Similarly, the optimal value function $V^{i,\zeta(o)}$ is estimated by

$$\hat{V}^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, \hat{u}_{t,k}^{i,\zeta}) = \frac{1}{2} \begin{bmatrix} \mathbf{E}_k^{i,\zeta T} & \hat{u}_{t,k}^{i,\zeta} \end{bmatrix} \Omega^{i,\zeta} \begin{bmatrix} \mathbf{E}_k^{i,\zeta} \\ \hat{u}_{t,k}^{i,\zeta} \end{bmatrix}$$

such that, $\Omega^{i,\zeta} \equiv \begin{bmatrix} \Omega_{\mathbf{E}_k^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} & \Omega_{\mathbf{E}_k^{i,\zeta} \hat{u}_{t,k}^{i,\zeta}}^{i,\zeta} \\ \Omega_{\hat{u}_{t,k}^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} & \Omega_{\hat{u}_{t,k}^{i,\zeta} \hat{u}_{t,k}^{i,\zeta}}^{i,\zeta} \end{bmatrix} \in \mathbb{R}^{4 \times 4}$

where $\Omega^{i,\zeta} > \mathbf{0}$, $\Omega_{\hat{u}_{t,k}^{i,\zeta} \hat{u}_{t,k}^{i,\zeta}}^{i,\zeta} \in \mathbb{R}$, and $\Omega_{\hat{u}_{t,k}^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} \in \mathbb{R}^{1 \times 3}$.

A gradient descent approach is applied for the online training of the neural networks. The target approximation values of the optimal strategy and the optimal value function can be expressed as $\tilde{u}_{t,k}^{i,\zeta} = - \left(\Omega_{\hat{u}_{t,k}^{i,\zeta} \hat{u}_{t,k}^{i,\zeta}}^{i,\zeta} \right)^{-1} \Omega_{\hat{u}_{t,k}^{i,\zeta} \mathbf{E}_k^{i,\zeta}}^{i,\zeta} \mathbf{E}_k^{i,\zeta}$ and $\tilde{V}_k^{i,\zeta} = U_k^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, \hat{u}_{t,k}^{i,\zeta}) + \hat{V}^{i,\zeta}(\mathbf{E}_{k+1}^{i,\zeta}, \hat{u}_{t,k+1}^{i,\zeta})$, respectively. As a result, the actor and critic approximation errors are $\varepsilon_{t,k}^{i,\zeta(actor)} = \frac{1}{2} \left(\hat{u}_{t,k}^{i,\zeta} - \tilde{u}_{t,k}^{i,\zeta} \right)^2$ and $\varepsilon_{t,k}^{i,\zeta(critic)} = \frac{1}{2} \left(\hat{V}^{i,\zeta}(\mathbf{E}_k^{i,\zeta}, \hat{u}_{t,k}^{i,\zeta}) - \tilde{V}_k^{i,\zeta} \right)^2$, respectively. Applying the gradient along the direction minimizing the errors yields the following weight update laws

$$\omega^{i,\zeta(r+1)} = \omega^{i,\zeta(r)} - \rho_a \left(\varepsilon_{t,k}^{i,\zeta(actor)} \mathbf{E}_k^{i,\zeta} \right)^{(r)} \quad (7)$$

$$\Omega^{i,\zeta(r+1)} = \Omega^{i,\zeta(r)} - \rho_c \left(\varepsilon_{t,k}^{i,\zeta(critic)} Z_{t,k}^{i,\zeta T} Z_{t,k}^{i,\zeta} \right)^{(r)} \quad (8)$$

where $Z_{t,k}^{i,\zeta} = [\mathbf{E}_k^{i,\zeta T} \hat{u}_{t,k}^{i,\zeta}]$, r is the iteration index of the weight update loop, and $0 < \rho_a, \rho_c < 1$ are the learning rates of the actor and critic, respectively.

IV. SEPARATION CONTROL MECHANISM

The separation control objective aims to prevent agents from colliding by imposing a safety distance between the agents. The RL protocol is designed such that it penalizes agents that are closer or further to each other than they should. Each agent i aggregates its decision based on its relative distance from its neighboring agents $j \in \mathcal{N}_i$, as described by (3). The mechanism is realized through a zero-order Tagaki-Sugeno (TS) fuzzy logic inference engine with an online adaptation capability based on an actor-critic scheme. The synergistic integration of fuzzy logic and connectionist modeling theories has been exploited in the past to provide an approximate dynamic programming solution of the separation control problem [15], [42]. The merit of such adaptive fuzzy-RL scheme is its ability to approximate

the highly nonlinear system's input-output relationship, using attractive-repulsive potential functions, on which it builds its optimized policy. Furthermore, it solves the granularity issue associated to the output space of classical RL approaches by converting it from a discrete action space to a continuous one.

A. Zero-Order TS Fuzzy Inference System

One of the most salient features of fuzzy logic is its ability to control complex systems characterized by dynamic uncertainties without the need for their precise mathematical models. It does so by incorporating human-like expertise, in the form of if-then rules, for the control of such ill-defined systems. An A -input single-output zero-order TS fuzzy system employs \mathcal{P} rules. Each rule $p \in \{1, \dots, \mathcal{P}\}$ is of the form

If θ_1 is $\Gamma_{m_1}^{(p)}, \dots$, and θ_A is $\Gamma_{m_A}^{(p)}$ Then $u^{(p)} = \phi^{(p)}$,

where input θ_a , $a \in \{1, \dots, A\}$, is an antecedent fuzzy variable, $\Gamma_{m_a}^{(p)}$ is a linguistic label associated to fuzzy set m_a , $u^{(p)}$ is a consequent fuzzy variable, and $\phi^{(p)} \in \mathbb{R}$ is a singleton. The firing strength of rule p is $\Psi^{(p)} = \left[\prod_{a=1}^A \eta^{\Gamma_{m_a}^{(p)}}(\theta_a) \right] / \sum_{p=1}^{\mathcal{P}} \left(\prod_{a=1}^A \eta^{\Gamma_{m_a}^{(p)}}(\theta_a) \right)$ where $\eta^{\Gamma_{m_a}^{(p)}}$ is a membership function associated to $\Gamma_{m_a}^{(p)}$. The fuzzy engine's inferred output is computed through a defuzzification process as $u^{\text{Fuzzy}} = \sum_{p=1}^{\mathcal{P}} \Psi^{(p)} \phi^{(p)}$.

B. Adaptive Critics Fuzzy System

A zero-order TS fuzzy system is adopted by each agent i to compute a separation signal $u_{s,k}^{ij,\zeta}$ to control its proximity to agent j , $\forall j \in \mathcal{N}_i$. The fuzzy engine takes a single input fuzzy variable $\tilde{\zeta}_k^{ij} = |\zeta_k^i - \zeta_k^j| - d$, where d is the desired safety distance. With such a single-input single-output fuzzy system, rule p becomes of the form

If $\tilde{\zeta}_k^{ij}$ is $\Gamma_{m_1}^{(p)}$ Then $u_{s,k}^{ij,\zeta(p)} = \phi^{ij,\zeta(p)}$.

The firing strength $\Psi_k^{ij,\zeta(p)}$ of rule p and the defuzzified output $u_{s,k}^{ij,\zeta}$ of the fuzzy system are computed as described above (with $A = 1$).

In order to design an adaptive law to tune the consequent membership function $\phi^{ij,\zeta(p)}$ of each rule p , the quality of the taken actions are assessed using a value function, which is approximated by a critic neural network $S_k^{ij,\zeta} = \sum_{p=1}^{\mathcal{P}} \Psi_k^{ij,\zeta(p)} \Phi^{ij,\zeta(p)}$ with $\Phi^{ij,\zeta(p)}$ being the critic weight corresponding to rule p . A temporal difference equation can be formed in terms of the value function $S_k^{ij,\zeta}(\tilde{\zeta}_k^{ij})$ and a reward function $\mathcal{R}_k^{ij,\zeta}$ as $S_k^{ij,\zeta}(\tilde{\zeta}_k^{ij}) = \mathcal{R}_k^{ij,\zeta} + S_{k+1}^{ij,\zeta}(\tilde{\zeta}_{k+1}^{ij})$. The learning process of the actor and critic weights relies on such temporal difference equations. The control objective here is to maximize instant rewards $\mathcal{R}_k^{ij,\zeta}$ based on the distance $\tilde{\zeta}_k^{ij}$, $\forall k$. Hence, the temporal difference error is represented by $\mathcal{T}_k^{ij,\zeta} = S_k^{ij,\zeta}(\tilde{\zeta}_k^{ij}) - (\mathcal{R}_k^{ij,\zeta} + S_{k+1}^{ij,\zeta}(\tilde{\zeta}_{k+1}^{ij}))$. Applying a gradient-based descent approach, the update law

of the actor weights is then

$$\begin{aligned} \phi^{ij,\zeta(p)(r+1)} &= \phi^{ij,\zeta(p)(r)} - \alpha_a \left(\text{sign} \left(\mathcal{T}_k^{ij,\zeta} \right) \frac{\partial u_{s,k}^{ij,\zeta}}{\partial \phi^{ij,\zeta(p)}} \right)^{(r)} \\ &= \phi^{ij,\zeta(p)(r)} - \alpha_a \left(\text{sign} \left(\mathcal{T}_k^{ij,\zeta} \right) \Psi^{ij,\zeta(p)} \right)^{(r)} \end{aligned} \quad (9)$$

where $0 < \alpha_a < 1$ is a learning rate. Similarly, the adaptative law of the critic weights is derived from the temporal difference evaluation $\mathcal{T}_k^{ij,\zeta}$ as

$$\begin{aligned} \Phi^{ij,\zeta(p)(r+1)} &= \Phi^{ij,\zeta(p)(r)} - \alpha_c \left(\left(\mathcal{T}_k^{ij,\zeta} \right) \frac{\partial S_k^{ij,\zeta}}{\partial \Phi^{ij,\zeta(p)}} \right)^{(r)} \\ &= \Phi^{ij,\zeta(p)(r)} - \alpha_c \left(\left(\mathcal{T}_k^{ij,\zeta} \right) \Psi^{ij,\zeta(p)} \right)^{(r)} \end{aligned} \quad (10)$$

for a learning rate $0 < \alpha_c < 1$. This adaptive critics structure is implemented by agent i in each direction ζ to provide an aggregate separation policy $u_{s,k}^{i,\zeta}$ defined by

$$u_{s,k}^{i,\zeta} = \frac{\sum_{j \in \mathcal{N}_i} \sum_{p=1}^{\mathcal{P}} \Psi_k^{ij,\zeta(p)} \phi^{ij,\zeta(p)}}{|\mathcal{N}_i|}. \quad (11)$$

The resulting control process provides each agent with an online collision avoidance mechanism without the need to undergo many offline training episodes before applying the right decision [15].

V. CONSENSUS CONTROL MECHANISM

The flock of agents is guided to reach a consensus on a common flock velocity using a communication layer designated by a graph topology. An undirected fully connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ is adopted for this purpose, where $\mathcal{N} = \{\delta_i\}_{i=1, \dots, |\mathcal{N}|}$ is the set of nodes of cardinality $|\mathcal{N}|$ and $\mathcal{E} = \{(\delta_i, \delta_j) \in \mathcal{N}^2\}$ is the set of edges representing the communication links between the agents [3]. We will denote the connectivity weights associated to every edge $(\delta_i, \delta_j) \in \mathcal{E}$ by $c_{ij} = c_{ji}$, $j \neq i$, with $c_{ii} = 0$. The local consensus protocol followed by each agent i is set to

$$u_{c,k}^{i,\zeta} = - \sum_{j \in \mathcal{N}_i} c_{ij} (v_k^{i,\zeta} - v_k^{j,\zeta}). \quad (12)$$

Hence, the consensus control decisions for all agents can be presented collectively by $\mathbf{u}_{c,k}^\zeta = -\mathbf{L}\mathbf{v}_k^\zeta$, where $\mathbf{u}_{c,k}^\zeta = [u_{c,k}^{1,\zeta} \ u_{c,k}^{2,\zeta} \ \dots \ u_{c,k}^{N,\zeta}]^T$, $\mathbf{v}_k^\zeta = [v_k^{1,\zeta} \ v_k^{2,\zeta} \ \dots \ v_k^{N,\zeta}]^T$, and \mathbf{L} is the graph Laplacian. Let $\mathbf{C} = [c_{ij}] \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$ be the graph's adjacency matrix and $\mathbf{C}^d = [c_{ii}^d] \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$ be a square diagonal matrix where $c_{ii}^d = \sum_{j \in \mathcal{N}_i} c_{ij}$. Then, $\mathbf{L} = \mathbf{C}^d - \mathbf{C}$. The convergence speed of this process is governed by the second eigenvalue associated with Fiedler Eigenvector of the graph Laplacian \mathbf{L} [43]. As a result, the consensus relies on the graph topology as well as the cohesion features implicitly imposed by the tracking control law.

VI. SIMULATION RESULTS AND DISCUSSION

Two simulation scenarios are set up to validate the performance of the proposed adaptive fuzzy-RL algorithm. The

first assigns a mobile agent as a leader and commands it to navigate in a circular trajectory defined by $x_k^\ell = 5 \cos(0.03k)$, $y_k^\ell = 5 \sin(0.03k)$. 20 other agents (followers) are then controlled by the fuzzy-RL algorithm to achieve the goals casted by (1). The agents' initial positions and velocities are randomly selected within the ranges of $[-5 \ 5]$ m and $[0 \ 1]$ m/s, respectively. The linear velocities in the x- and y-directions are bound to $[-10 \ 10]$ m/s. A reward function $\mathcal{R}_k^{ij,\zeta}$ is designed to give the highest value when the distance between the agents is equal to the target value d .

$$\mathcal{R}_k^{ij,\zeta} = \begin{cases} -\tilde{\zeta}_k^{ij}, & \text{if } \tilde{\zeta}_k^{ij} > 0 \\ 3, & \text{if } \tilde{\zeta}_k^{ij} = 0 \\ \frac{3\tilde{\zeta}_k^{ij}}{d}, & \text{if } \tilde{\zeta}_k^{ij} < 0 \end{cases}$$

The fuzzy logic engine is designed with five symmetric triangular membership functions as follows:

$$\eta^{\Gamma_{m1}}(\tilde{\zeta}_k^{ij}) = \begin{cases} 0, & \text{if } \tilde{\zeta}_k^{ij} < t_l \\ \frac{\tilde{\zeta}_k^{ij} - t_l}{t_c - t_l}, & \text{if } t_l \leq \tilde{\zeta}_k^{ij} \leq t_c \\ \frac{t_h - \tilde{\zeta}_k^{ij}}{t_h - t_c}, & \text{if } t_c \leq \tilde{\zeta}_k^{ij} \leq t_h \\ 0, & \text{if } \tilde{\zeta}_k^{ij} > t_h \end{cases}$$

The membership functions are centered around $t_c \in \{-6, -3, 0, 3, 6\}$ with $t_l = t_h = 1.5$. The input universe of the fuzzy logic controller is set to $(-7.5, 7.5)$. These parameters reflect the action and state spaces for each agent, while the consequences of the fuzzy rules are decided online using the adaptive actor-critic structures. The remaining simulation parameters are taken as: $\mathbf{Q}^i = \mathbf{I}_{3 \times 3}$, $\mathbf{R}^i = 1$, $\rho_c = 10^{-7}$, $\alpha_c = 0.05$, $\rho_a = 10^{-2}$, $\alpha_a = 0.1$, $d = 2$ m, $T = 0.1$ s. The neighborhood \mathcal{N}_i , for each follower i , is defined to be $\mathcal{N}_i = \{\delta_j\}_{j=1, \dots, N} \setminus \{\delta_i, \delta_\ell\}$. This makes $|\mathcal{N}_i| = 19$, since $N = 21$. In order to quantitatively assess the performance of the proposed algorithm, the following objective measures are adopted: The average separation error for each follower i at time index k is measured as $O_{s,k}^i = \left[\sum_{j \in \mathcal{N}_i} (\|\mathbf{p}_k^j - \mathbf{p}_k^i\| - d) \right] / |\mathcal{N}_i|$. The overall average tracking error, separation error, and velocity are measured as $O_{t,k} = (\sum_{i \neq \ell} \|\mathbf{p}_k^i - \mathbf{p}_k^\ell\|) / (N - 1)$, $O_{s,k} = (\sum_{i \neq \ell} O_{s,k}^i) / (N - 1)$, and $O_{v,k} = (\sum_{i \neq \ell} \|\mathbf{q}_k^i\|) / (N - 1)$, respectively.

The results are shown in Fig. 1. The performance is compared with a similar technique proposed by Gu et al. in [3]. Fig. 1a reveals the circular paths of the agents, where the start and end locations are indicated by a triangle and a circle, respectively. It is interesting to notice that with the adaptive fuzzy-RL method, not only do all the signals reach their respective steady states faster than with Gu's method, they do so within less than 2 s. Although this comes at the expense of a larger variation (standard deviation) for the average tracking and average separation errors, the agents reach a consensus on almost the exact same velocity with practically a nil standard deviation compared to Gu's method (Fig. 1d). The proposed method led to a significantly lower

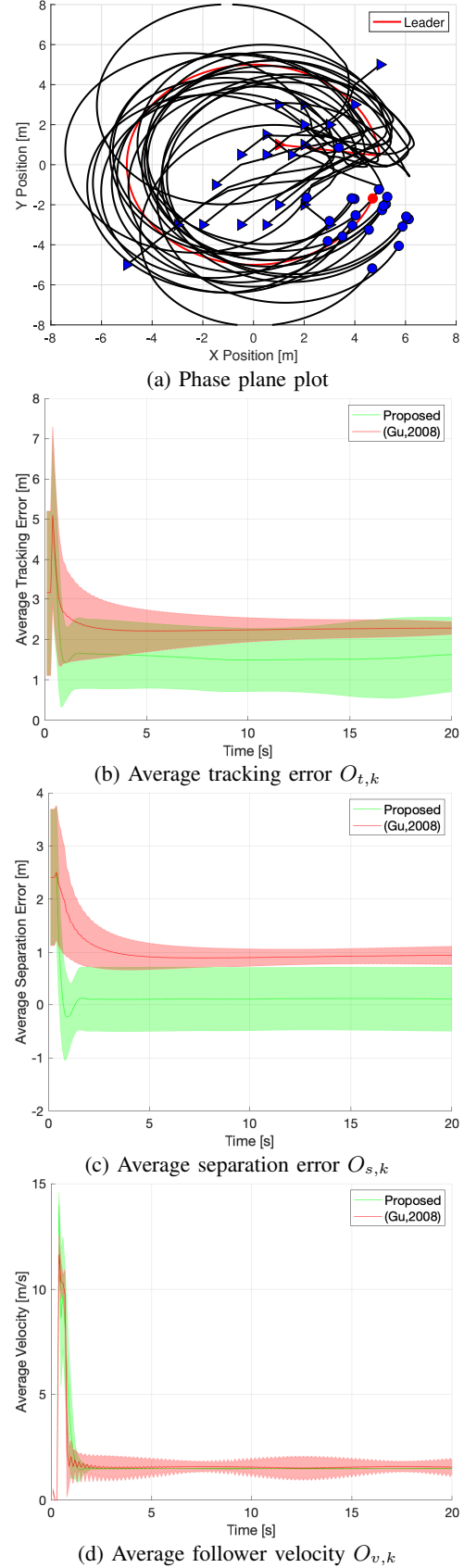


Fig. 1: Simulation results of Scenario 1

average tracking and average separation errors than Gu's controller. The average separation error with the adaptive fuzzy-RL algorithm rapidly converged to practically zero, outperforming Gu's algorithm whose error converged to 1 m. This means that while maintaining the competing objectives, the desired overall separation is achieved among the agents.

The second simulation is designed to test the online ability of the fuzzy-RL controller to adapt to dynamic disturbances while the agents are in action. To that end, the flock starts off as in the first simulation, but with the following changes: 1) at time $t = 10$ s, 4 of the 20 followers are decommissioned; 2) at $t = 20$ s, the leader changes its trajectory from a circular to a linear trajectory defined by $v_k^{\ell,x} = 1.7321$, $v_k^{\ell,y} = 1$ m/s (a linear motion with a heading of 30°); and 3) at $t = 30$ s, the safety distance d is changed from 2 to 2.5 m.

The results of this scenario are depicted in Fig. 2. They are not compared with Gu's algorithm this time because it was not designed to be adaptive to dynamic variations. Its feedback policies are based on fixed control gains. The 4 agents decommissioned at $t = 10$ s are identified in green in Fig. 2a. The figure demonstrates how the followers are able to globally track the leader regardless of the sudden change in trajectory or the number of active agents. The average tracking error, separation error, and velocity, rapidly converged right after each disturbance. Their standard deviations were not much affected by the changes in the simulation conditions. Nevertheless, the average separation error was the signal that is most affected by varying the safety distance from 2 to 2.5 m at $t = 30$ s. It goes from almost zero right before the change to about -0.75 m after it. On the other hand, Fig. 2d shows that the standard deviation of the consensus velocity converges to an insignificant value after each disturbance.

VII. CONCLUSION

An adaptive distributed online fuzzy reinforcement learning approach is proposed for the autonomous control of flock systems. It tries to simultaneously compromise three objectives: 1) tracking a leader; 2) avoiding collisions by maintaining a predefined safety distance between the neighboring agents; and 3) reaching a velocity consensus among the flock. The algorithm guides the flock towards a leader by applying a reinforcement learning scheme that is only dependent on the agent position feedback signals. The collision avoidance is realized by means of an adaptive fuzzy inference system that is based on its own reinforcement learning process. Finally, a graph-based protocol is employed to enable consensus on a common flock velocity. In addition to its flexible and simple structure, the proposed algorithm is shown to possess a number of other salient features, such as fast convergence and online adaptability to dynamic disturbances. More specifically, it is proved to be relatively insensitive to the sudden changes in the leader's trajectory, number of active agents, and the target separation distance between the agents. The algorithm's superiority was demonstrated against a similar technique proposed in the literature.

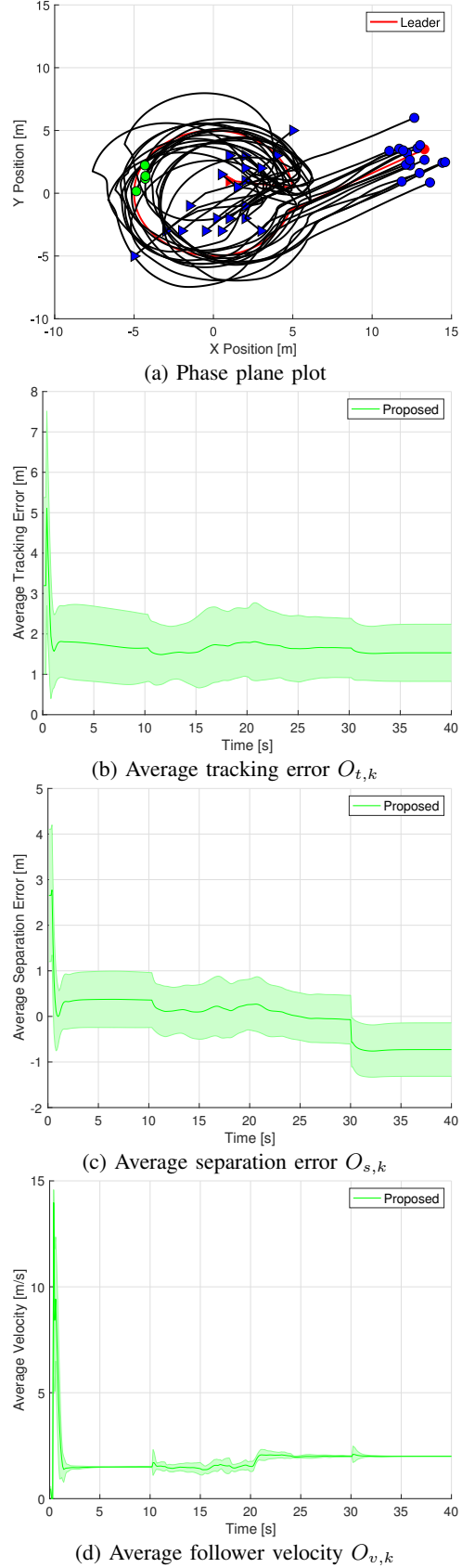


Fig. 2: Simulation results of Scenario 2

REFERENCES

- [1] U. Halder and B. Dey, "Biomimetic algorithms for coordinated motion: Theory and implementation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5426–5432.
- [2] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "A distributed formation-based odor source localization algorithm - design, implementation, and wind tunnel evaluation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1830–1836.
- [3] D. Gu and H. Hu, "Using fuzzy logic to design separation function in flocking algorithms," *IEEE Transactions on fuzzy Systems*, vol. 16, no. 4, pp. 826–838, 2008.
- [4] M. Abouheaf and W. Gueaieb, "Flocking motion control for a system of nonholonomic vehicles," in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. IEEE, 2017, pp. 32–37.
- [5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.
- [6] A. A. Paranjape, S. Chung, K. Kim, and D. H. Shim, "Robotic herding of a flock of birds using an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 901–915, 2018.
- [7] M. Ghasemi and S. G. Nersesov, "Finite-time coordination in multi-agent systems using sliding mode control approach," *Automatica*, vol. 50, no. 4, pp. 1209–1216, 2014.
- [8] T. Li, R. Zhao, C. P. Chen, L. Fang, and C. Liu, "Finite-time formation control of under-actuated ships using nonlinear sliding mode control," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3243–3253, 2018.
- [9] J. Hu and G. Feng, "Distributed tracking control of leader-follower multi-agent systems under noisy measurement," *Automatica*, vol. 46, no. 8, pp. 1382–1387, 2010.
- [10] H. M. La, T. H. Nguyen, C. H. Nguyen, and H. N. Nguyen, "Optimal flocking control for a mobile sensor network based a moving target tracking," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 4801–4806.
- [11] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 293–307, 2009.
- [12] Y. Yang and D. Yue, "Distributed tracking control of a class of multi-agent systems in non-affine pure-feedback form under a directed topology," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 169–180, 2018.
- [13] Z. Li, L. Gao, W. Chen, and Y. Xu, "Distributed adaptive cooperative tracking of uncertain nonlinear fractional-order multi-agent systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 292–300, 2020.
- [14] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19–22, 1992.
- [15] T. Akinbulire, H. Schwartz, R. Falcon, and R. Abielmona, "A reinforcement learning approach to tackle illegal, unreported and unregulated fishing," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–8.
- [16] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.
- [17] Z. Cao, Q. Xiao, and M. Zhou, "Distributed fusion-based policy search for fast robot locomotion learning," *IEEE Computational Intelligence Magazine*, vol. 14, no. 3, pp. 19–28, 2019.
- [18] M. Abouheaf, W. Gueaieb, and F. Lewis, "Online model-free reinforcement learning for the automatic control of a flexible wing aircraft," *IET Control Theory & Applications*, vol. 14, no. 1, pp. 73–84, 2020.
- [19] M. Abouheaf, N. Q. Mailhot, W. Gueaieb, and D. Spinello, "Guidance mechanism for flexible-wing aircraft using measurement-interfaced machine-learning platform," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4637–4648, 2020.
- [20] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in neural information processing systems*, 2007, pp. 1–8.
- [21] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6015–6022.
- [22] R. Konda, H. M. La, and J. Zhang, "Decentralized function approximated q-learning in multi-robot systems for predator avoidance," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6342–6349, 2020.
- [23] K. Lobos-Tsunekawa, F. Leiva, and J. Ruiz-del-Solar, "Visual navigation for biped humanoid robots using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3247–3254, 2018.
- [24] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6252–6259.
- [25] X. Qi, Y. Luo, G. Wu, K. Boriboonsomsin, and M. Barth, "Deep reinforcement learning enabled self-learning control for energy efficient driving," *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 67–81, 02 2019.
- [26] J. Woo, C. Yu, and N. Kim, "Deep reinforcement learning-based controller for path following of an unmanned surface vehicle," *Ocean Engineering*, vol. 183, pp. 155–166, 2019.
- [27] J. Valasek, J. Doebbler, M. D. Tandale, and A. J. Meade, "Improved adaptive-reinforcement learning control for morphing unmanned air vehicles," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 1014–1020, 2008.
- [28] M. Abouheaf and W. Gueaieb, "Multi-agent reinforcement learning approach based on reduced value function approximations," in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, 2017, pp. 111–116.
- [29] P. Pennesi and I. C. Paschalidis, "A distributed actor-critic algorithm and applications to mobile sensor network coordination problems," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 492–497, 2010.
- [30] M. Abouheaf, M. S. Mahmoud, and W. Gueaieb, "Integral reinforcement learning solutions for a synchronisation system with constrained policies," *IET Control Theory & Applications*, vol. 14, no. 12, pp. 1599–1611, 2020.
- [31] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.
- [32] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on fuzzy systems*, vol. 8, no. 4, pp. 349–366, 2000.
- [33] P. Singhala, D. Shah, and B. Patel, "Temperature control using fuzzy logic," *arXiv preprint arXiv:1402.3654*, 2014.
- [34] W.-S. Lin, C.-L. Huang, and M.-K. Chuang, "Hierarchical fuzzy control for autonomous navigation of wheeled robots," *IEEE Proceedings-Control Theory and Applications*, vol. 152, no. 5, pp. 598–606, 2005.
- [35] D. Driankov and A. Saffiotti, *Fuzzy logic techniques for autonomous vehicle navigation*. Physica, 2013, vol. 61.
- [36] M. J. Er and Y. Gao, "Robust adaptive control of robot manipulators using generalized fuzzy neural networks," *IEEE Transactions on industrial electronics*, vol. 50, no. 3, pp. 620–628, 2003.
- [37] B. Innocenti, B. López, and J. Salvi, "A multi-agent architecture with cooperative fuzzy control for a mobile robot," *Robotics and Autonomous Systems*, vol. 55, no. 12, pp. 881–891, 2007.
- [38] X. Li and B.-J. Choi, "Design of obstacle avoidance system for mobile robot using fuzzy logic systems," *International Journal of Smart Home*, vol. 7, no. 3, pp. 321–328, 2013.
- [39] 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.
- [40] M. I. Abouheaf and F. L. Lewis, "Approximate dynamic programming solutions of multi-agent graphical games using actor-critic network structures," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–8.
- [41] M. I. Abouheaf, F. L. Lewis, M. S. Mahmoud, and D. G. Mikulski, "Discrete-time dynamic graphical games: model-free reinforcement learning solution," *Control Theory and Technology*, vol. 13, no. 1, pp. 55–69, 2015.
- [42] L. Jouffe, "Fuzzy inference system learning by reinforcement methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 3, pp. 338–355, 1998.
- [43] 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.