2021 29th Mediterranean Conference on Control and Automation (MED)
June 22-25, 2021. Bari, Puglia, Italy

2021 29th Mediterranean Conference on Control and Automation (MED) | 978-1-6654-2258-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/MED51440.2021.9480176

Distributed safe reinforcement learning for multi-robot motion planning

Yang Lu, Yaohua Guo, Guoxiang Zhao, and Minghui Zhu

***Abstract*— This paper studies optimal motion planning of multiple mobile robots with collision avoidance. We develop a distributed reinforcement learning algorithm which ensures suboptimal goal reaching and anytime collision avoidance simultaneously. Theoretical results on the convergence of neural network weights, the uniform and ultimate boundedness of system states of the closed-loop system, and anytime collision avoidance are established. Numerical simulations for single integrator and unicycle robots illustrate the effectiveness of our theoretical results.**

***Index Terms*— Safety, reinforcement learning, motion planning**

I. Introduction

The rapid advances in embedded processors, mobile sensing and high-speed communications in the last decades stimulate the emergency of multi-robot systems. please see, e.g., [1]. Compared with single-robot systems, multi-robot systems exhibit greater flexibility, robustness and adaptability [2]. Consequently, multi-robot systems have a wide range of applications, e.g., traffic coordination [3] and sensor deployment [4].

Distributed control is desired due to large scale of multi-robot systems. Most existing distributed control algorithms of mobile robots are myopic. In particular, the control law of each robot is driven by the partial gradient of a local objective function. This class of distributed control algorithms are scalable to robot number and robust to failures of individual robots. Their asymptotic convergence is ensured using, e.g., Lyapunov analysis, but their optimality, e.g., total energy consumption, is not. Recently, differential game theory has been adopted to synthesize distributed optimal controllers [5]–[9]. Using Nash equilibrium strategies, each agent is able to optimize its own performance objective. As for linear multi-agent systems, both open-loop and feedback Nash equilibrium strategies have been well studied [5], [7]. As for nonlinear multi-agent systems, game solutions are characterized by coupled Hamilton-Jacobi-Bellman (HJB) equations. Dynamic programming (DP) is a typical way to solve the HJB equations. However, DP suffers from the well-known curse of dimensionality. Reinforcement learning (RL) or adaptive dynamic programming (ADP) has been recognized as an effective technique to mitigate the curse of dimensionality via iteratively performing policy evaluation and policy improvement until a suboptimal solution is found [10], [11]. Several efforts have been made to solve multi-agent coordination problems via RL/ADP. In [9], a value iteration Heuristic DP algorithm is proposed to solve the dynamic graphical games of discrete-time multi-agent systems. For continuous-time differential graphical games, [8] proposes a cooperative policy iteration algorithm to compute the optimal control solutions, where the value function and control policy are approximated by critic and actor neural networks (NNs), respectively. The weight errors of NNs are proven to be uniformly ultimately bounded (see Definition 4 in [8]). Different from [8], paper [12] utilizes generalized fuzzy hyperbolic models to approximate the value functions of the coupled HJB equations associated with optimal consensus control. Nevertheless, system safety (collision avoidance) has not been addressed in the papers aforementioned.

Stability of closed-loop systems under RL/ADP has been extensively studied; please refer to the survey paper [13]. The adaptive controllers implement policy iteration by an actor/critic NN structure and can simultaneously guarantee optimality and closed-loop dynamical stability. Recently, the work [14] develops a framework of robust RL/ADP to address both parametric and dynamic uncertainties. The synthesized controllers ensure robust stability against the system uncertainties and become suboptimal when the system uncertainties are absent. However, system safety is still not taken into account in these papers.

Contributions. This paper studies cooperative optimal motion planning of a group of robots. We aim to identify distributed algorithms to solve the problem online and meanwhile ensure robot safety all the time. The goal reaching objective is first formulated as an optimal control problem. Then a repulsive potential function is incorporated into the value function to facilitate collision avoidance. A distributed RL algorithm is developed to solve the induced HJB equations. In particular, the algorithm adopts a single critic NN in contrast to classic actor-critic NN schemes. The simplified NN relaxes the need of an initial admissible controller and improves computational efficiency. Theoretical results on the convergence of NN weights, the stability of the overall closed-loop system, and anytime collision avoidance are established. The efficacy of the theoretical results is verified by simulations for single integrator and unicycle robots.

Notations. The following notations are adopted throughout the paper. Let $\mathbb{R}^n$ be the set of real numbers of size n and $\mathbb{R}^{n\times m}$ be the set of $n \times m$ real matrices. Denote by I_n the $n \times n$ identity matrix. Let $\sigma_{\min}(A)$ denote the minimum singular value of a matrix A. Denote by $\|A\|$ the 2-norm of a vector or matrix A.

Y. Lu (*Corresponding author*), G. Zhao and M. Zhu are with School of Electrical Engineering and Computer Science, Pennsylvania State University, University Park, PA 16802 (Email: {yml5046, gfz5014, muz16}@psu.edu); Y. Guo is with School of Aerospace Engineering, Xi'an Jiaotong University, Xi'an, China, 710049 (Email: gyhua123@xjtu.edu.cn).
This work was partially supported by the grants NSF ECCS-1710859 and CNS 1830390.

II. Problem formulation

This section presents the system model and formulates the motion planning problem.

A. System model

Consider a group of N mobile robots, denoted by $\mathcal{V} \triangleq \{1, \cdots, N\}$. Each robot i has state $z_i \in \mathbb{R}^n$ and control input $u_i \in \mathbb{R}^m$. The dynamics of z_i is governed by the following control-affine system:

$$\dot{z}_i(t) = f_i(z_i(t)) + g_i(z_i(t))u_i(t). \tag{1}$$

In system (1), $z_i(t) \triangleq [p_i^T(t), q_i^T(t)]^T$ is robot i's state, where $p_i(t) \triangleq [x_i(t), y_i(t)]^T \in \mathbb{R}^2$ is its position in the Cartesian coordinate frame and $q_i(t) \in \mathbb{R}^{n-2}$ is the sub-state of robot i other than its position. System (1) includes, e.g., single integrator, double integrator and unicycle, as special cases. Each robot has a detection region with radius R. Denote $z \triangleq [z_1^T, \cdots, z_N^T]^T$ and $p \triangleq [p_1^T, \cdots, p_N^T]^T$. Define the detection set as $D \triangleq \{p \in \mathbb{R}^{2N} : \exists i, j \in \mathcal{V}, \text{ s.t. } \|p_i - p_j\| \leq R\}$. Define the safety set as $\Theta \triangleq \{p \in \mathbb{R}^{2N} : \|p_i - p_j\| > r, \ \forall i, j \in \mathcal{V}\}$, where $0 < r < R$ is the safety distance between two robots.

B. Motion planning

Each robot $i \in \mathcal{V}$ has a desired destination $z_i^d = [p_i^{dT}, q_i^{dT}]^T \in \mathbb{R}^n$. Let $\tilde{z}_i \triangleq z_i - z_i^d$, $\tilde{p}_i \triangleq p_i - p_i^d$, $\tilde{q}_i \triangleq q_i - q_i^d$, $z^d = [z_1^{dT}, \cdots, z_N^{dT}]^T$, $\tilde{z} \triangleq [\tilde{z}_1^T, \cdots, \tilde{z}_N^T]^T$, $p^d = [p_1^{dT}, \cdots, p_N^{dT}]^T$, and $\tilde{p} \triangleq [\tilde{p}_1^T, \cdots, \tilde{p}_N^T]^T$. The safety set in terms of $\tilde{p}$ is then $\tilde{\Theta} \triangleq \{\tilde{p} \in \mathbb{R}^{2N} : \tilde{p} + p^d \in \Theta\}$. The dynamics of $\tilde{z}_i$ is

$$\dot{\tilde{z}}_i(t) = \dot{z}_i(t) = \tilde{f}_i(\tilde{z}_i(t)) + \tilde{g}_i(\tilde{z}_i(t))u_i(t) \tag{2}$$

where $\tilde{f}_i(\tilde{z}_i) \triangleq f_i(\tilde{z}_i + z_i^d)$ and $\tilde{g}_i(\tilde{z}_i) \triangleq g_i(\tilde{z}_i + z_i^d)$. The control objective of this paper is formulated as follows.

Problem 1: Design distributed optimal feedback control strategies such that the robots can eventually reach the desired points while avoiding collisions. Mathematically, the objectives are formulated as:

$$\text{Goal reaching}: \lim_{t\to\infty} \|\tilde{z}(t)\| = 0. \tag{3}$$

$$\text{Safety}: \ p(t) \in \Theta, \ \forall t \geq 0. \tag{4}$$

The following assumption on the robots' initial and desired states is necessary to ensure the feasibility of the collision avoidance objective.

Assumption 2.1: It holds that $p(0) \in \Theta$ and $p^d \in \Theta$.

A centralized solution to solve Problem 1 is to formulate it as a constrained optimal control problem, where the objective function deals with the goal reaching objective and the hard constraint deals with the safety objective [15]. However, the centralized solution is offline and not scalable with respect to the robot number. This paper adopts a novel approach to address the challenge. The roadmap of the overall approach is summarized here. First, we only consider the individual goal reaching problem and formulate it as an optimal control problem. After that, we further adopt repulsive potential function (RPF) to facilitate the safety objective and incorporate it as a soft constraint of the optimal control problem. Finally, we use neural networks (NNs) to approximate optimal value functions and the associated controllers.

Individual goal reaching. Only considering the goal reaching objective (3), the following is the infinite-horizon cost of robot i when system (2) starts from state $\tilde{z}_i(t)$ at time t and is driven by a feedback control policy u_i:

$$J_i(\tilde{z}_i(t), u_i) = \int_t^\infty (\|\tilde{z}_i(\tau)\|^2 + \|u_i(\tilde{z}_i(\tau))\|^2) d\tau. \tag{5}$$

We note that (5) is decoupled since it only considers the individual goal reaching problem without taking into account the safety issue. The optimal value function (without considering the safety issue) $V_i^*(\tilde{z}_i(t))$ is defined as

$$V_i^*(\tilde{z}_i(t)) = \min_{u_i \in \mathcal{U}} \int_t^\infty (\|\tilde{z}_i(\tau)\|^2 + \|u_i(\tilde{z}_i(\tau))\|^2) d\tau \tag{6}$$

where $\mathcal{U}$ is the set of feedback controllers from $\mathbb{R}^{nN} \to \mathbb{R}^m$. Notice that $u_i(\cdot)$ in general depends on the overall state $\tilde{z} \in \mathbb{R}^{nN}$ to accommodate the later analysis that will take into account collision avoidance of the robots. The following continuous differentiability assumption on V_i^* is standard in the RL literature [16]–[18]. Under this assumption, the image sets of V_i^* and $\frac{\partial V_i^*}{\partial \tilde{z}_i}$ are both compact on a compact set of $\tilde{z}_i$. Moreover, the continuous differentiability property is also needed to rationalize using NNs to approximate V_i^*'s; please see the next section.

Assumption 2.2: For all $i \in \mathcal{V}$, $V_i^*(\tilde{z}_i)$ is continuously differentiable over $\mathbb{R}^n$.

Under Assumption 2.2, given a feedback control policy $u_i(\cdot)$ and a value function $V_i(\cdot)$, the robots' Hamiltonian functions are defined as:

$$H_i(\tilde{z}_i, \frac{\partial V_i}{\partial \tilde{z}_i}, u_i) \triangleq \|\tilde{z}_i\|^2 + \|u_i(\tilde{z}_i)\|^2 + \frac{\partial V_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}(\tilde{f}_i(\tilde{z}_i) + \tilde{g}_i(\tilde{z}_i)u_i(\tilde{z}_i)). \tag{7}$$

The robots' optimal controllers are obtained by minimizing the Hamiltonian functions associated with the optimal value functions, as follows:

$$u_i^*(\tilde{z}_i) = \operatorname*{argmin}_{u_i \in \mathcal{U}} H_i(\tilde{z}_i, \frac{\partial V_i^*}{\partial \tilde{z}_i}, u_i) = -\frac{1}{2}\tilde{g}_i^T(\tilde{z}_i)\frac{\partial V_i^*(\tilde{z}_i)}{\partial \tilde{z}_i}. \tag{8}$$

The optimal value function V_i^* and the optimal controller u_i^* satisfy the following HJB equation

$$H_i(\tilde{z}_i, \frac{\partial V_i^*}{\partial \tilde{z}_i}, u_i^*) = 0. \tag{9}$$

Integration of individual goal reaching and collision avoidance. Next, we further introduce repulsive potential functions to facilitate the safety objective (4). In particular, to avoid inter-robot collisions, for each pair of robots $i, j \in \mathcal{V}$, we construct the repulsive potential function as

$$V_{ij}(\tilde{p}_i, \tilde{p}_j) \triangleq \Big(\min\Big\{0, \frac{\|\tilde{p}_i + p_i^d - \tilde{p}_j - p_j^d\|^2 - R^2}{\|\tilde{p}_i + p_i^d - \tilde{p}_j - p_j^d\|^2 - r^2}\Big\}\Big)^2, \quad \forall i, j \in \mathcal{V}, \ i \neq j. \tag{10}$$

When $\|p_i - p_j\|$ is beyond the detection range R, the repulsive potential function is not activated and outputs zero. When two robots detect each other, the value of V_{ij} monotonically increases as the robots are getting closer, and it grows to infinity when they collide, i.e., $\|p_i - p_j\| = r$.

Let $V_{ii}(\tilde{p}) \triangleq \sum_{j\in\mathcal{V}\setminus\{i\}} V_{ij}(\tilde{p}_i, \tilde{p}_j)$. To achieve both the goal reaching objective (3) and the safety objective (4), for each $i \in \mathcal{V}$, we consider the following value function:

$$\tilde{V}_i^*(\tilde{z}) \triangleq V_i^*(\tilde{z}_i) + V_{ii}(\tilde{p}). \tag{11}$$

The Hamiltonian associated with (11) is defined as:

$$H_i(\tilde{z}_i, \frac{\partial \tilde{V}_i^*}{\partial \tilde{z}_i}, \tilde{u}_i) \triangleq \|\tilde{z}_i\|^2 + \|\tilde{u}_i(\tilde{z})\|^2 + \frac{\partial \tilde{V}_i^{*T}(\tilde{z})}{\partial \tilde{z}_i}(\tilde{f}_i(\tilde{z}_i) + \tilde{g}_i(\tilde{z}_i)\tilde{u}_i(\tilde{z})),\ \forall i \in \mathcal{V}. \tag{12}$$

The robots' optimal controllers are obtained by minimizing (12) as follows:

$$\tilde{u}_i^*(\tilde{z}) = \underset{\tilde{u}_i\in\mathcal{U}}{\operatorname{argmin}} H_i(\tilde{z}_i, \frac{\partial \tilde{V}_i^*}{\partial \tilde{z}_i}, \tilde{u}_i) = -\frac{1}{2}\tilde{g}_i^T(\tilde{z}_i)\frac{\partial \tilde{V}_i^*(\tilde{z})}{\partial \tilde{z}_i} = -\frac{1}{2}\tilde{g}_i^T(\tilde{z}_i)(\frac{\partial V_i^*(\tilde{z}_i)}{\partial \tilde{z}_i} + \frac{\partial V_{ii}(\tilde{p})}{\partial \tilde{z}_i}),\ \forall i \in \mathcal{V}. \tag{13}$$

The following assumptions are needed to ensure the goal reaching objective.

Assumption 2.3: It holds that $\sum_{i\in\mathcal{V}} H_i(\tilde{z}_i, \frac{\partial \tilde{V}_i^*}{\partial \tilde{z}_i}, \tilde{u}_i^*) < \sum_{i\in\mathcal{V}}(\|\tilde{z}_i\|^2 + \|\tilde{u}_i^*(\tilde{z})\|^2)$ for any $\tilde{z} \in \tilde{\Theta} \times \mathbb{R}^{(n-2)N}$.

Assumption 2.4: It holds that $p^d \notin D$.

Remark 2.1: By (12), we have $\sum_{i\in\mathcal{V}} H_i(\tilde{z}_i, \frac{\partial \tilde{V}_i^*}{\partial \tilde{z}_i}, \tilde{u}_i^*) - \sum_{i\in\mathcal{V}}(\|\tilde{z}_i\|^2 + \|\tilde{u}_i^*(\tilde{z})\|^2) = \sum_{i\in\mathcal{V}} \dot{\tilde{V}}_i^*$. Assumption 2.3 implies that $\sum_{i\in\mathcal{V}} \dot{\tilde{V}}_i^* < 0$ and hence $\sum_{i\in\mathcal{V}} \tilde{V}_i^*$ is a Lyapunov function. Under Assumption 2.4, for any $i, j \in \mathcal{V}$, $V_{ij}(\tilde{p}_i, \tilde{p}_j) = 0$ when $\tilde{p}_i = \tilde{p}_j = 0$. Notice that $V_i^*(\tilde{z}_i) \geq 0$ and $V_i^*(\tilde{z}_i) = 0$ if and only if $\tilde{z}_i = 0$. Assumption 2.4 then implies that $\sum_{i\in\mathcal{V}} \tilde{V}_i^*(\tilde{z}) = 0$ if and only if $\tilde{z} = 0$ and hence $\tilde{z} = 0$ is the only limit point.

Theorem 2.1: Suppose that Assumptions 2.1–2.4 hold. Then objectives (3) and (4) are achieved simultaneously by the controllers (13).

Remark 2.2: Another possible RPF-based approach is to include the RPF $V_{ii}(\tilde{p})$ into the integrand of the overall value function $\tilde{V}_i^*(\tilde{z})$ as, e.g., $\tilde{V}_i^*(\tilde{z}(t)) = \min_{u_i\in\mathcal{U}} \int_t^\infty (\|\tilde{z}_i(\tau)\|^2 + \|u_i(\tilde{z}_i(\tau))\|^2 + V_{ii}(\tilde{p}(\tau)))d\tau$. This approach may only guarantee collision avoidance in the almost everywhere sense. Specifically, we can derive that $\tilde{V}^*(\tilde{z}(t))$ is decreasing. Hence, if $\tilde{V}^*(\tilde{z}(0)) < \infty$, we have $\tilde{V}^*(\tilde{z}(t)) < \infty$ for any $t \geq 0$. However, since $V_{ii}(\tilde{p}(t))$ is inside the integrand, even if $\tilde{V}^*(\tilde{z}(t)) < \infty$ for any $t \geq 0$, $V_{ii}(\tilde{p}(t))$ could be infinite at isolated time instants with zero measure. In contrast, in our approach, since $V_{ii}(\tilde{p}(t))$ is outside the integrand, $\tilde{V}^*(\tilde{z}(t)) < \infty$ for any $t \geq 0$ ensures that $V_{ii}(\tilde{p}(t)) < \infty$ for any $t \geq 0$, which implies anytime collision avoidance.

III. Controller Design and Analysis

With Theorem 2.1, the problem becomes to solve the HJB equations (9). However, it is in general difficult or even impossible to get the closed-form expression of V_i^*. In the field of RL, the solutions of HJB equations are often obtained through the Policy Iteration (PI) algorithm. Nevertheless, the PI algorithm needs an initial admissible controller, which can stabilize the system and introduce a finite function value and often runs off-line (See [12] for example). To relax this requirement, we will develop a novel RL algorithm by using a single critic NN to approximate the optimal value function, and also adding an additional term to ensure stability and collision avoidance of the multi-robot system. Most existing RL schemes have two NNs, one for actor and the other for critic. Our scheme only has a single critic NN and hence has a lower computational complexity.

A. Approximator of local optimal value function

This subsection presents a distributed single NN-based RL technique to identify an optimal solution of the optimal control problem. According to the universal approximation property of NNs [19], the continuously differentiable optimal value function V_i^* can be approximated by

$$V_i^*(\tilde{z}_i) = W_i^T\phi_i(\tilde{z}_i) + \varepsilon_i(\tilde{z}_i) \tag{14}$$

where $\phi_i : \mathbb{R}^n \to \mathbb{R}^\eta$ is the continuously differentiable activation function with η the number of neurons in the hidden layer, $\varepsilon_i : \mathbb{R}^n \to \mathbb{R}$ is the continuously differentiable approximation error function, and $W_i \in \mathbb{R}^\eta$ is the least-square weight vector, i.e., $W_i = \operatorname{argmin}_{\bar{W}_i\in\mathbb{R}^\eta}\|V_i^*(\tilde{z}_i) - \bar{W}_i^T\phi_i(\tilde{z}_i)\|$. It is well known that as $\eta \to \infty$, $\varepsilon_i(\tilde{z}_i) \to 0$ and $\frac{\partial \varepsilon_i(\tilde{z}_i)}{\partial \tilde{z}_i} \to 0$ [20]. The derivative of $V_i^*(\tilde{z}_i)$ is

$$\frac{\partial V_i^*(\tilde{z}_i)}{\partial \tilde{z}_i} = \frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}W_i + \frac{\partial \varepsilon_i(\tilde{z}_i)}{\partial \tilde{z}_i}. \tag{15}$$

Substituting (15) into (8) and (13), respectively, renders the following

$$u_i^*(\tilde{z}_i, W_i) = -\frac{1}{2}\tilde{g}_i^T(\tilde{z}_i)(\frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}W_i + \frac{\partial \varepsilon_i(\tilde{z}_i)}{\partial \tilde{z}_i}), \tag{16}$$

$$\tilde{u}_i^*(\tilde{z}, W_i) = -\frac{1}{2}\tilde{g}_i^T(\frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}W_i + \frac{\partial \varepsilon_i(\tilde{z}_i)}{\partial \tilde{z}_i} + \frac{\partial V_{ii}(\tilde{p})}{\partial \tilde{z}_i}). \tag{17}$$

By substituting (16) into (9), the HJB equation can be rewritten as

$$0 = H_i(\tilde{z}_i, W_i) = \|\tilde{z}_i\|^2 - \frac{1}{4}W_i^T\frac{\partial \phi_i(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{g}_i(\tilde{z}_i)\tilde{g}_i^T(\tilde{z}_i)\frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}W_i + W_i^T\frac{\partial \phi_i(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{f}_i(\tilde{z}_i) - d_i(\tilde{z}_i, W_i) \tag{18}$$

where all the terms of $\varepsilon_i(\tilde{z}_i)$ are included in $d_i(\tilde{z}_i, W_i)$:

$$d_i(\tilde{z}_i, W_i) \triangleq \frac{1}{4}\frac{\partial \varepsilon_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{g}_i(\tilde{z}_i)\tilde{g}_i^T(\tilde{z}_i)\frac{\partial \varepsilon_i(\tilde{z}_i)}{\partial \tilde{z}_i} + \frac{1}{2}W_i^T\frac{\partial \phi_i(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{g}_i(\tilde{z}_i)\tilde{g}_i^T(\tilde{z}_i)\frac{\partial \varepsilon_i(\tilde{z}_i)}{\partial \tilde{z}_i} - \frac{\partial \varepsilon_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{f}_i(\tilde{z}_i).$$

Since the least-square weight vector W_i and the approximation error function $\varepsilon_i(\cdot)$ are unknown, estimated weight vector $\hat{W}_i$ is used to build the NN, i.e.,

$$\hat{V}_i^*(\tilde{z}_i) = \hat{W}_i^T \phi_i(\tilde{z}_i). \tag{19}$$

Substituting (19) into (8) and (13), respectively, renders

$$\hat{u}_i(\tilde{z}_i, \hat{W}_i) = -\frac{1}{2}\tilde{g}_i^T(\tilde{z}_i)\frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}\hat{W}_i, \tag{20}$$

$$\hat{\tilde{u}}_i(\tilde{z}, \hat{W}_i) = -\frac{1}{2}\tilde{g}_i^T(\tilde{z}_i)\left(\frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}\hat{W}_i + \frac{\partial V_{ii}(\tilde{p})}{\partial \tilde{z}_i}\right). \tag{21}$$

Based on (19) and (20), the HJB equation (18) is approximated by

$$\hat{H}_i(\tilde{z}_i, \hat{W}_i) = \|\tilde{z}_i\|^2 - \frac{\hat{W}_i^T}{4}\frac{\partial \phi_i(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{g}_i(\tilde{z}_i)\tilde{g}_i^T(\tilde{z}_i)\frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}\hat{W}_i + \hat{W}_i^T\frac{\partial \phi_i(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{f}_i(\tilde{z}_i) \triangleq e_i(\tilde{z}_i, \hat{W}_i). \tag{22}$$

The term $e_i(\tilde{z}_i, \hat{W}_i)$ is the Bellman residual error and given by $e_i(\tilde{z}_i, \hat{W}_i) = \hat{H}_i(\tilde{z}_i, \hat{W}_i) - H_i(\tilde{z}_i, W_i)$. This error is caused by $\varepsilon_i(\cdot)$ and the difference between W_i and $\hat{W}_i$. We are to select $\hat{W}_i$ for $i \in \mathcal{V}$ such that the squared residual error $E(\tilde{z}, \hat{W}) \triangleq \sum_{i\in\mathcal{V}} \frac{1}{2}e_i^2(\tilde{z}_i, \hat{W}_i)$ is minimized. For each $i \in \mathcal{V}$, design the learning law of $\hat{W}_i$ as follows

$$\dot{\hat{W}}_i = -\frac{\alpha_i \bar{\sigma}_i(\tilde{z}_i, \hat{W}_i)}{m_i(\tilde{z}_i, \hat{W}_i)}e_i(\tilde{z}_i, \hat{W}_i) - \alpha_i F_i \hat{W}_i \tag{23}$$

where $\bar{\sigma}_i(\tilde{z}_i, \hat{W}_i) \triangleq \sigma_i(\tilde{z}_i, \hat{W}_i)/m_i(\tilde{z}_i, \hat{W}_i)$, with $\sigma_i(\tilde{z}_i, \hat{W}_i) \triangleq \frac{\partial \phi_i(\tilde{z}_i)}{\partial \tilde{z}_i}(\tilde{f}_i(\tilde{z}_i) - \frac{1}{2}\tilde{g}_i(\tilde{z}_i)\tilde{g}_i^T(\tilde{z}_i)\frac{\partial \phi_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}\hat{W}_i)$ and $m_i(\tilde{z}_i, \hat{W}_i) \triangleq \|\sigma_i(\tilde{z}_i, \hat{W}_i)\|^2 + 1$, $\alpha_i > 0$ is the learning rate, and $F_i \in \mathbb{R}$ is a design parameter. The first term of (23) is based on the normalized gradient descent algorithm, which minimizes the squared residual error $\frac{1}{2}e_i^2(\tilde{z}_i, \hat{W}_i)$. The second term of (23) stems from Lyapunov stability analysis and is needed to establish stability of the closed-loop system. In particular, it introduces a parameter F_i which is to be chosen such that the second-order term associated with the weight estimation error $W - \hat{W}$ in the time derivative of a Lyapunov function is negative definite. This property will establish stability of the closed-loop system. The implementation of the distributed RL algorithm is given by Algorithm 1.

Algorithm 1: Distributed safe RL algorithm

while $t \geq 0$ **do**
- Each robot $i \in \mathcal{V}$ measures $z_i(t)$ and $z_j(t)$ for all j such that $\|z_i(t) - z_j(t)\| \leq R$;
- Each robot $i \in \mathcal{V}$ updates $\hat{W}_i$ by (23);
- Each robot $i \in \mathcal{V}$ executes (21) to system (1).

We need the following assumptions to guarantee the closed-loop stability and collision avoidance.

Assumption 3.1: For each $i \in \mathcal{V}$, $f_i(z_i)$ and $g_i(z_i)$ are continuous in z_i.

Assumption 3.2: For each $i \in \mathcal{V}$, $\text{rank}(g_i(z_i)) = n$ for any $z_i \in \mathbb{R}^n$.

Remark 3.1: Assumption 3.1 is very mild and is satisfied by many standard physical systems, including single integrator, double integrator and unicycle. Under this assumption, the image sets of f_i and g_i are compact on a compact set of z_i. Assumption 3.2 states that $g_i(z_i)$ has full row rank for any $z_i \in \mathbb{R}^n$. This is a sufficient assumption to ensure collision avoidance for Algorithm 1. Assumption 3.2 is satisfied by single integrator, but not satisfied by double integrator or unicycle. However, as will be shown in the next section, collision avoidance is indeed achieved for unicycle under Algorithm 1. This illustrates that our algorithm could be successfully applied to a wider range of problems in practice.

The next theorem establishes the convergence, stability and collision avoidance of the closed-loop multi-robot system under Algorithm 1.

Theorem 3.1: Suppose Assumptions 2.1, 2.2, 3.1, and 3.2 are satisfied. Consider the multi-robot system (1). Let the control inputs be provided by (21) and the NN weight tuning laws be given by (23). Then, there exists a positive integer η_0 such that for any $\eta > \eta_0$, $p(t) \in \Theta$ for all $t \geq 0$, and moreover, the tracking error $\|z - z^d\|$, the weight estimation error $\|W - \hat{W}\|$ and control policy estimation error $\|\hat{\tilde{u}} - \tilde{u}^*\|$ are all uniformly ultimately bounded.

IV. Simulation

This section verifies the efficacy of Algorithm 1 by simulations for single integrator and unicycle robots.

A. Single integrator robots

Consider a group of 10 robots, where each robot i's state $z_i(t)$ is governed by the single integrator dynamics $\dot{z}_i(t) = u_i(t)$. Here, $n = m = 2$, $z_i = p_i$, q_i is null, and $u_i = [v_{ix}, v_{iy}]^T$ is robot i's linear velocities along the x-axis and y-axis in the Cartesian coordinate frame. We choose $R = 1$ and $r = 0.1$. The goal states are equally distributed around a circle while the initial state of the robots are random. Following the successful practice in [8], the basis functions choose the second order polynomials of the state variables. The learning law (23) ensures UUB of the tracking error $\tilde{z}$ and weight estimation error $\tilde{W}$. However, it says nothing about the optimality of the learning result. In this example, $\hat{W} = 0$ is a trivial equilibrium. If starting from $\hat{W}(0) = 0$, the trajectory of $\hat{W}(t)$ will remain unchanged and there is actually no learning at all. To avoid converging to such trivial equilibrium points, we add an additional term to (23) so that (23) is modified as $\dot{\hat{W}}_i = -\frac{\alpha_i \bar{\sigma}_i(\tilde{z}_i, \hat{W}_i)}{m_i(\tilde{z}_i, \hat{W}_i)}e_i(\tilde{z}_i, \hat{W}_i) - \alpha_i F_i \hat{W}_i + \frac{\alpha_i}{2}\Upsilon_i(\tilde{z}_i, \hat{\tilde{u}}_i)\frac{\partial \phi_i(\tilde{z}_i)}{\partial \tilde{z}_i}\tilde{g}_i(\tilde{z}_i)\tilde{g}_i^T(\tilde{z}_i)\frac{\partial K_i(\tilde{z}_i)}{\partial \tilde{z}_i}$, where $K_i(\tilde{z}_i)$ is radially unbounded non-negative function such that $K_i(\tilde{z}_i) = 0$ if and only if $\tilde{z}_i = 0$, and $\Upsilon_i(\tilde{z}_i, \hat{\tilde{u}}_i)$ is defined as $\Upsilon_i(\tilde{z}_i, \hat{\tilde{u}}_i) = 0$ if $\frac{\partial K_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}(\tilde{f}_i(\tilde{z}_i) + \tilde{g}_i(\tilde{z}_i)\tilde{u}_i^*) < 0$ and $\Upsilon_i(\tilde{z}_i, \hat{\tilde{u}}_i) = 1$ if otherwise. Roughly speaking, the newly added term is active in the learning update law if and only if system (2) is unstable under the controller $\tilde{u}_i^*$. This guarantees that the learning takes effect as long as

system (2) is not yet stable and effectively avoids the situation of converging to trivial equilibrium points. Under the assumption that there exists a positive constant c_i such that $\frac{\partial K_i^T(\tilde{z}_i)}{\partial \tilde{z}_i}(\tilde{f}_i(\tilde{z}_i) + \tilde{g}_i(\tilde{z}_i)\tilde{u}_i^*) \leq -c_i \| \frac{\partial K_i(\tilde{z}_i)}{\partial \tilde{z}_i} \|^2$ for any $\tilde{z}_i \in \mathbb{R}^n$, all the claims of Theorem 3.1 still hold [21]. In the simulation, for each $i \in \mathcal{V}$, function K_i is chosen as $K_i(\tilde{z}_i) = \tilde{z}_i^T \tilde{z}_i$. Other learning parameters are set as $\alpha = 0.001$ and $F = 0$. Figure 1 shows the 10 single integrators can exponentially converge to their goal states and no collision occurs during the convergence as indicated in Figure 2. The associated trajectories are shown in Figure 3. The evolution of each robot's neural network weights are illustrated in Figure 4, where within 4 seconds all neural networks converge to non-zero stationary points.

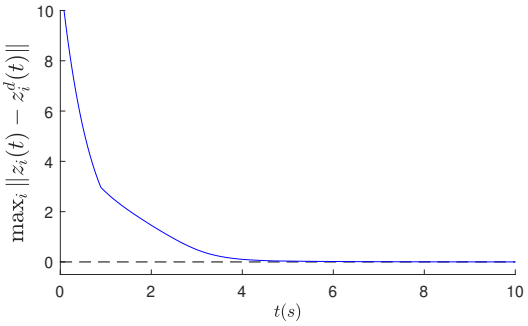

Fig. 1. Maximum formation errors of 10 single integrators over time.

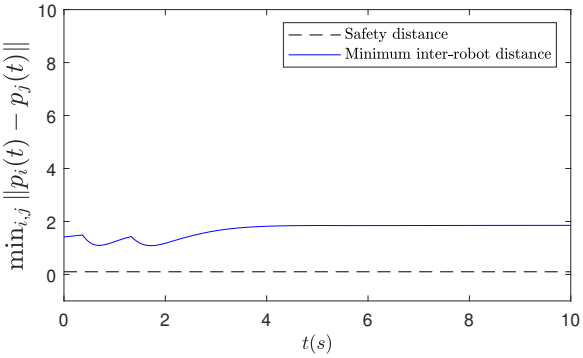

Fig. 2. Minimum inter-robot distances of 10 single integrators over time.

B. Unicycle robots

We examine Algorithm 1 on unicycle robots. All settings are consistent with those on the single integrator robots except $F = 0.1$. The goal region for each robot is a ball of radius 1 centered at their goal states. Robots arriving at their goal regions are immediately removed from the scene. The commonly-used unicycle dynamic $\begin{bmatrix} \dot{x}_i & \dot{y}_i & \dot{\theta}_i \end{bmatrix}^T = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix}$ includes the orientation θ_i as a periodic state variable while the positions x_i and y_i are not. To address such a difference, we introduce $\theta_i^x = \cos\theta_i$ and $\theta_i^y = \sin\theta_i$ to remove the periodic property and an

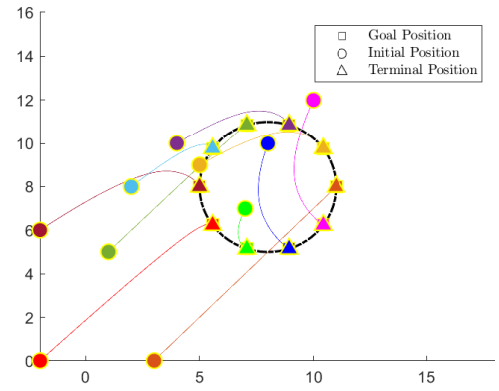

Fig. 3. Trajectories of 10 single integrators

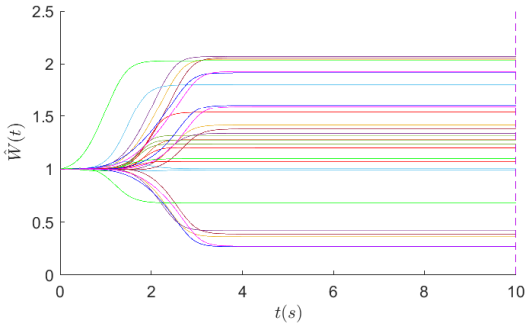

Fig. 4. Neural network weights of 10 single integrators over time.

alternative form of the unicycle dynamic is as follows: $[\dot{x}_i \quad \dot{y}_i \quad \dot{\theta}_i^x \quad \dot{\theta}_i^y]^T = \begin{bmatrix} \theta_i^x & \theta_i^y & 0 & 0 \\ 0 & 0 & -\theta_i^y & \theta_i^x \end{bmatrix}^T \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix}$. The neural network weights are initialized following a zero-mean Gaussian distribution with standard deviation 0.1. Figure 5 shows that all unicycle robots converge to their respective goal positions within finite time. Around 2100s and 2300s, the last robot was close to its goal region but deviates from it because its neural network is not optimal yet and requires more information for training. Figure 6 indicates that no collision occurs in the movement and the resulting trajectories are shown in Figure 7. Figure 8 shows the evolution of each robot's neural network over time, where curves of same colors are the weights of the same robots and the vertical dash lines indicate the arrival time. Figure 8 show that all weights remain finite until the robots arrive at their goal regions. The oscillatory curves in Figure 8 are caused by insufficient exploration, where the robots are exploring new environment and the neural networks have not converged yet.

V. Conclusions

This paper has studied optimal motion planning for multi-robot systems. A distributed safe reinforcement learning algorithm is presented to ensure both goal reaching and collision avoidance. Simulation results for single integrator and unicycle robots illustrate the feasibility and effectiveness

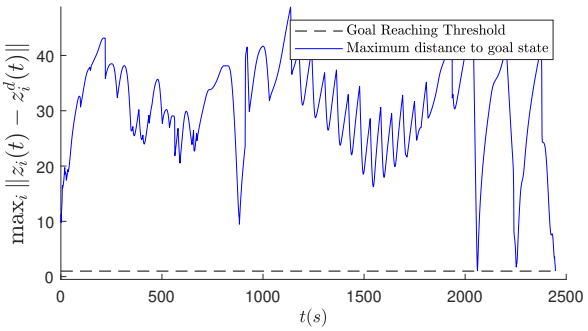

Fig. 5. Maximum formation errors of 10 unicycle robots over time.

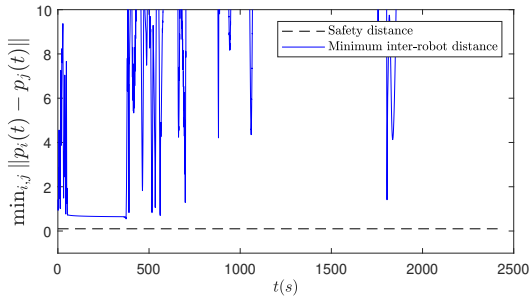

Fig. 6. Minimum inter-robot distances of 10 unicycle robots over time.

of the developed algorithm.

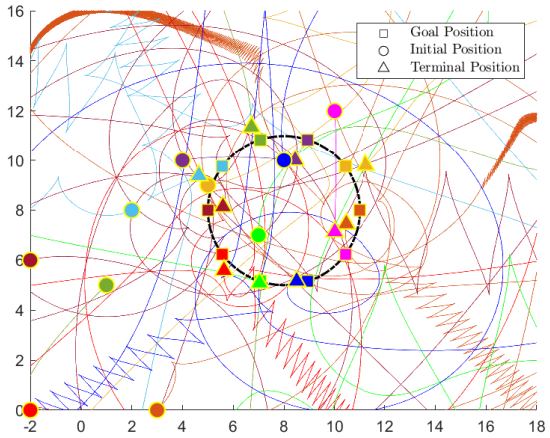

Fig. 7. Trajectories of 10 unicycle robots.

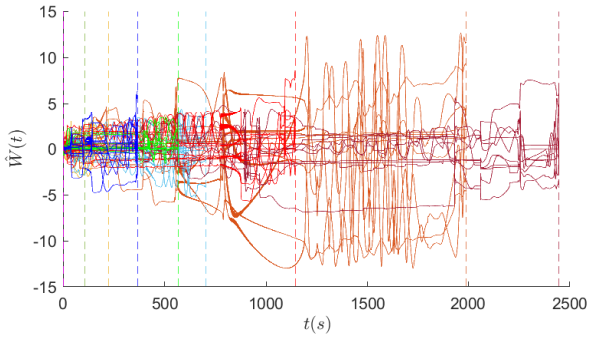

Fig. 8. Neural network weights of 10 unicycle robots over time.

REFERENCES

[1] A.Khan, B.Rinner, and A.Cavallaro, "Cooperative robots to observe moving targets: Review," *IEEE Trans. Cybern.*, vol. 48, pp. 187–198, January 2018.

[2] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Auto. Robots*, vol. 4, pp. 7–27, March 1997.

[3] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Trans. Robot.*, vol. 23, pp. 1170–1183, December 2007.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, pp. 243–255, April 2004.

[5] D. Gu, "A differential game approach to formation control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 1, pp. 85–94, 2008.

[6] P. Reddy and G. Zaccour, "Feedback Nash equilibria in linear-quadratic difference games with constraints," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 590–604, 2017.

[7] T. Mylvaganam, M. Sassano, and A. Astolfi, "A differential game approach to multi-agent collision avoidance," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4229–4235, 2017.

[8] K. G. Vamvoudakis and F. L. Lewis, "Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton-Jacobi equations," *Automatica*, vol. 47, pp. 1556–1569, August 2011.

[9] 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, pp. 3038–3053, December 2014.

[10] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol. 12, pp. 219–245, January 2000.

[11] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*. New York: Wiley, 2002.

[12] H. Zhang, J. Zhang, G.-H. Yang, and Y. Luo, "Leader-based optimal coordination contorol for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 1, pp. 152–163, 2015.

[13] S. N. Balakrishnan, J. Ding, and F. L. Lewis, "Issues on stability of ADP feedback controllers for dynamical systems," *IEEE Trans. Syst. Man Cybern. B*, vol. 38, pp. 913–917, August 2008.

[14] Z. Jiang and Y. Jiang, "Robust adaptive dynamic programming for linear and nonlinear systems: An overview," *Eur. J. Control*, vol. 19, pp. 417–425, September 2013.

[15] G. Zhao and M. Zhu, "Pareto optimal multi-robot motion planning," *IEEE Transactions on Automatic Control*, 2020. To appear.

[16] D. Liu, Y. Huang, D. Wang, and Q. Wei, "Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming," *Int. J. Control*, vol. 86, no. 9, pp. 1554–1566, 2013.

[17] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online adaptive algorithm for optimal control with integral reinforcement learning," *Int. J. Robust Nonlin.*, vol. 24, pp. 2686–2710, November 2014.

[18] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to slove the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, pp. 878–888, May 2010.

[19] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, 1990.

[20] B. A. Finlayson, *The method of weighted residuals and variational principles*. New York: Academic Press, 1990.

[21] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 206–216, 2013.