2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Barrier Function-based Safe Reinforcement Learning for Formation Control of Mobile Robots

Xinglong Zhang[1], Yaoqian Peng[1], Wei Pan[2], Xin Xu[1], Haibin Xie[1]

Abstract— **Distributed model predictive control (DMPC) concerns how to online control multiple robotic systems with constraints effectively. However, the nonlinearity, nonconvexity, and strong interconnections of dynamic system models and constraints can make the real-time and real-world DMPC implementations nontrivial. Reinforcement learning (RL) algorithms are promising for control policy design. However, how to ensure safety in terms of state constraints in RL remains a significant issue. This paper proposes a barrier function-based safe reinforcement learning algorithm for DMPC of nonlinear multi-robot systems under state constraints. The proposed approach is composed of several local learning-based MPC regulators. Each regulator, associated with a local system, learns and deploys the local control policy using a safe reinforcement learning algorithm in a distributed manner, i.e., with state information only among the neighbor agents. As a prominent feature of the proposed algorithm, we present a novel barrier-based policy structure to ensure safety, which has a clear mechanistic interpretation. Both simulated and real-world experiments on the formation control of mobile robots with collision avoidance show the effectiveness of the proposed safe reinforcement learning algorithm for DMPC.**

I. Introduction

Distributed model predictive controls (DMPC) of multi-robot systems have received significant attention in recent years [1]–[3]. Compared with centralized control solutions, the distributed control structure is more efficient, scalable, and friendly to maintain [4]. A well-studied solution to multi-robot control with state constraints relies on a distributed structure based on model predictive control (MPC) [5], [6]. In this setting, many DMPC approaches have been proposed [7]–[11] where local MPC problems are transformed into several optimization problems which can be solved online in parallel with information exchanges from their neighbors. Such a procedure requires periodic access to the local onboard computational resources. To reduce the computational burden, an explicit distributed MPC algorithm was proposed [12], in which a set of piece-wise explicit control policies, associated with marked separate constraint regions, are firstly computed offline. Then, the control policies are searched online according to the region where the state lies. However, the approach is designed only for linear systems and relies on a separability assumption of systems.

In a different setting, distributed solutions based on reinforcement learning (RL) and adaptive dynamic programming (ADP) have also received great attention in the past decades [13]–[19]. In the RL-based distributed approaches, RL and ADP usually rely upon an actor-critic structure to learn a near-optimal control policy of the infinite horizon optimal control problem in a forward-in-time manner. Compared with DMPC, distributed RL methods [13]–[19] can solve the optimization problem more efficiently. Moreover, the resulting control policy can be learned offline and deployed online, dramatically reducing the computational load. Given the key feature, actor-critic reinforcement learning has been used in a receding horizon manner to learn an explicit control policy of a centralized MPC in [20]–[22]. In this paper, we extend the work [20]–[22] to the scenario of distributed control for multiple agents.

Crucially, the satisfaction of state constraint in actor-critic learning is a difficult task and an indispensable requirement in many safety-critical distributed control applications. To the best of our knowledge, this issue has not been addressed in the above-mentioned learning-based MPC [20], [21] and RL algorithms [13]–[19], [23], [24]. In [25], a safe reinforcement learning algorithm was proposed for multi-robot motion planning. An impulsive potential field was utilized to reconstruct the cost function, which was inspired by the Lyapunov barrier-based cost function shaping approach for the centralized safe RL [26]–[28]. However, such a cost function-shaping treatment could introduce abrupt changes in the cost function gradient, prone to diverging behavior of the weights of both actor and critic networks.

This paper proposes a barrier function-based safe reinforcement learning (BSRL) approach for distributed MPC (DMPC), termed as SL-DMPC, of nonlinear multi-robot systems with state constraints. The proposed approach is composed of several local BSRL-based MPC regulators. Each regulator, associated with a local system, learns and deploys the local MPC policy with the BSRL algorithm in a distributed manner, i.e., with state information exchanges only among the neighbor agents. Note that how to guarantee to learn safety under state constraints is still challenging in the reinforcement learning community. In the proposed BSRL algorithm, different from [25], [27], [28], we present a novel barrier-based control policy structure to guarantee learning safety under state constraints, which has a clear mechanistic interpretation for a safety certificate.

Unlike DMPC with numerical solutions, the implemen-

The work was supported by the National Natural Science Foundation of China under Grant 62003361, 61825305, U21A20518 and Postdoctoral Science Foundation under Grant 47680. Wei Pan is supported by Huawei.
[1]Xinglong Zhang, Yaoqian Peng, Xin Xu, and Haibin Xie are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, 410073, China. {zhangxinglong18,pengyq20,xinxu}@nudt.edu.cn, xhb2575_sx@sina.com
[2]Wei Pan is with the Department of Cognitive Robotics, Delft University of Technology, the Netherlands. wei.pan@tudelft.nl
Xinglong Zhang and Yaoqian Peng contributed equally to this work.

tation of SL-DMPC relies upon several barrier-based actor-critic RL algorithms, generating for each agent a local explicit state-feedback control policy rather than an open-loop control sequence using policy gradient. These local control policies can be learned and deployed online synchronously or learned offline and deployed online. Moreover, the safety guarantee of the barrier-based reinforcement learning algorithm is proven in each prediction horizon. Both simulated and real-world experiments on formation control of mobile robots with collision avoidance have been demonstrated, which shows the effectiveness of the proposed SL-DMPC.

The rest of the paper is organized as follows. Section II introduces the problem formulation and a preliminary on DMPC for linear dynamical systems. In Section III, SL-DMPC algorithm is proposed for nonlinear systems. The theoretical analysis is presented. In Section IV, both the simulation and real-world experimental results for nonlinear formation control of mobile robots are demonstrated. Conclusions are drawn in Section V.

Notation: We use $\mathbb{N}_{l_1}^{l_2}$ to denote the set of integers $l_1, l_1 + 1, \cdots, l_2$. For a group of vectors $z_i \in \mathbb{R}^{n_i}$, $i \in \mathbb{N}_1^M$, we use $\mathrm{col}_{i\in\mathbb{N}_1^M}(z_i)$ to denote $[z_1^\top, \cdots, z_M^\top]^\top$, where M is an integer. We use $\boldsymbol{u}(k)$ to represent a control policy formed by the control sequence $u(k), \cdots, u(k+N-1)$, where N is the prediction horizon of MPC, k is the discrete-time index.

II. Problem formulation and preliminary on DMPC

A. Problem formulation

The overall system under control is composed of M discrete-time, nonlinear, interacting subsystems, which are described by

$$\Sigma_i: \; x_i(k+1) = f_i(x(k)) + g_i(x_i(k))u_i(k) \qquad (1)$$

$\forall i \in \mathbb{N}_{i=1}^M$, where $x_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ and $u_i \in \mathbb{R}^{m_i}$ are the state and input variables associated with the subsystem Σ_i, while k is the discrete-time index, $f_i \in \mathbb{R}^{n_i}$ and $g_i \in \mathbb{R}^{n_i \times m_i}$ are smooth state transition and input mapping functions respectively and $f_i(0) = 0$. The local state sets are defined as $\mathcal{X}_i = \{x_i \in \mathbb{R}^{n_i} | B_i^t(x_i) \leq 0, t \in \mathbb{N}_1^{q_i}\}$, $i \in \mathbb{N}_1^M$, B_i^t is a C^1 function, and $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_M$.

Collecting all the subsystems from (1), the overall centralized dynamical model, denoted as Σ, can be written as

$$\Sigma: \; x(k+1) = F(x(k)) + G(x(k))u(k) \qquad (2)$$

where $x = \mathrm{col}_{i\in\mathbb{N}_1^M}(x_i) \in \mathbb{R}^n$, $n = \sum_{i=1}^M n_i$, $u = \mathrm{col}_{i\in\mathbb{N}_1^M}(u_i) \in \mathbb{R}^m$, $m = \sum_{i=1}^M m_i$, $F = (f_1, \cdots, f_M)$, the diagonal blocks of G are input matrices g_i, $i \in \mathbb{N}_1^M$. Now we give a definition of neighbor subsystem in a multi-robot system.

Definition 1 (Neighbor subsystem): For a subsystem Σ_i, the state x_j is a neighbor state of Σ_i if x_j^a, x_j^b with $x_j^a \neq x_j^b$ implies different values of f_i. The state $x_{\mathcal{N}_i} \in \mathbb{R}^{n_{\mathcal{N}_i}}$ is the collection of all the neighbor states, i.e., $x_{\mathcal{N}_i} = \mathrm{col}_{j\in\mathcal{N}_i} x_j$, where $\mathcal{N}_i$ is the set of all the indices j associated with the neighbor states. ◀

With Definition 1, model (1) can be written as

$$\Sigma_i: \; x_i(k+1) = f_i(x_{\mathcal{N}_i}(k)) + g_i(x_i(k))u_i(k) \; . \qquad (3)$$

Here we give the assumption on the communication protocol among the robots below:

Assumption 1 (Communication protocol): The communications among the neighbor subsystems are allowed, i.e., the state information can be exchanged in a bidirectional way between Σ_i and Σ_j if $i \in \mathcal{N}_j$ or $j \in \mathcal{N}_i$. ◀

The control objective considered in this paper is to drive $x(k) \to 0$ and $u(k) \to 0$ as $k \to +\infty$, meanwhile enforcing local state constraints $x_i(k) \in \mathcal{X}_i$, for all $i \in \mathbb{N}_1^M$.

B. Preliminary on DMPC for linear systems

Given the capability of MPC in dealing with state constraints explicitly, DMPC can be naturally used to solve the considered multi-robot control problem. First, let us recall a DMPC formulation [29] for linear systems. At each time-step k the following finite horizon cooperative cost function is to be minimized:

$$\min_{\boldsymbol{u}_1(k), \cdots, \boldsymbol{u}_M(k)} \quad J(x(k)) \qquad (4)$$

where the global cost J is defined as

$$J(x(k)) = \begin{array}{l} \sum_{i=1}^M \sum_{j=0}^{N-1} r_i(x_{\mathcal{N}_i}(k+j), u_i(k+j)) \\ + \|x_i(k+N)\|_{P_i}^2, \end{array} \qquad (5)$$

the stage cost $r_i(x_{\mathcal{N}_i}(k), u_i(k)) = \|x_{\mathcal{N}_i}(k)\|_{Q_i}^2 + \|u_i(k)\|_{R_i}^2$, $N > 0$ is the prediction horizon, $Q_i = Q_i^\top \in \mathbb{R}^{n_{\mathcal{N}_i} \times n_{\mathcal{N}_i}}$, $Q_i \succ 0$, $R_i = R_i^\top \in \mathbb{R}^{m_i \times m_i}$, $R_i \succ 0$, $i \in \mathbb{N}_1^M$; $P_i = P_i^\top \succ 0$.

The optimization problem in (4) with cost (5) can be solved numerically subject to a linear version of model (2), local state constraints $x_i(k+j) \in \mathcal{X}_i$, $\forall\, j \in \mathbb{N}_1^N$, $i \in \mathbb{N}_1^M$. This problem can be solved in a distributed manner via numerical optimization tools, see [29], [30], since the local state constraints are decoupled. As such, the local open-loop control sequence can be computed for each agent in each prediction horizon. As in the procedure of dynamic programming, the first control action is applied, and (4) is solved repeatedly at the next time instant.

C. Definitions on barrier function and safe control

We introduce a definition of barrier function on state constraint, which will be used for state constraint satisfaction in the barrier function-based reinforcement learning algorithm.

Definition 2 (Barrier function): For any set $\mathcal{X}_i$, $i \in \mathbb{N}_1^M$, a barrier function is defined as

$$\mathcal{B}_i^o(x_i) = \begin{cases} -\sum_{t=1}^{q_i} \log\big(-B_i^t(x_i)\big) & x_i \in \mathrm{Int}(\mathcal{X}_i) \\ +\infty \;\; \text{otherwise.} \end{cases} \qquad (6)$$

A recentered transformation of $\mathcal{B}_i^o(x_i)$ centered at $z_i \in \mathbb{R}^{n_i}$ is defined as $\mathcal{B}_i(x_i) = \mathcal{B}_i^o(x_i) - \mathcal{B}_i^o(z_i) - \nabla\mathcal{B}_i^o(z_i)^\top x_i$, where $z_i = 0$ if $\{0\} \subseteq \mathcal{X}_i$ or z_i is selected such that $z_i \in \mathcal{X}_i$ otherwise. ◀

Definition 3 (Safe control in the prediction horizon): At a generic time k, a control policy denoted as $\boldsymbol{u}(k)$

is safe for (2) if the resulting state evolutions satisfy $x(k+1), \cdots, x(k+N) \in \mathcal{X}^N$. ◀

III. Barrier Function-based safe reinforcement learning for DMPC

Since the dynamic system model (1) is nonlinear even nonconvex, solving the considered multi-robot control problem using DMPC like (4) is nontrivial. This section presents a barrier function-based safe reinforcement learning algorithm to solve DMPC, i.e., SL-DMPC, to learn each agent's local state feedback control policy in a distributed manner. In Section III-A, we first present the main idea of SL-DMPC, in which a barrier function-based control policy is utilized to deal with state constraints. The associated safety guarantee and convergence analysis of the proposed approach are given in each prediction horizon. In Section III-B, the barrier function-based reinforcement learning algorithm is presented to implement SL-DMPC.

A. Design of SL-DMPC

The main difficulty of designing an RL algorithm to solve DMPC lies in guaranteeing safety under local state constraints. To address this issue, in SL-DMPC we propose a barrier function-based control policy structure to guarantee safety. To this end, we first reconstruct the performance index J with barrier functions, i.e.,

$$\begin{aligned}\bar{J}(x(k)) = \textstyle\sum_{i=1}^{M}\sum_{j=0}^{N-1} \bar{r}_i(x_{\mathcal{N}_i}(k+j), u_i(k+j)) + \\ \bar{J}_i(x_i(k+N))\end{aligned} \tag{7}$$

where the stage cost is

$$\bar{r}_i(x_{\mathcal{N}_i}(\tau), u_i(\tau)) = r_i(x_{\mathcal{N}_i}(\tau), u_i(\tau)) + \mu\mathcal{B}_i(x_i(\tau)),$$

$\bar{J}_i(x_i(k+N)) = \|x_i(k+N)\|^2_{P_i} + \mu\mathcal{B}_i(x_i(k+N))$, $\mu > 0$ is a tuning parameter.

In a collective form, for any $\tau \in [k, k+N-1]$, one can write

$$\bar{J}\big(x(\tau)\big) = \bar{r}(x(\tau), u(\tau)) + \bar{J}\big(x(\tau+1)\big) \tag{8}$$

where $\bar{r}(x(\tau), u(\tau)) = \sum_{i=1}^{M} \bar{r}_i(x_{\mathcal{N}_i}(\tau), u_i(\tau))$, $\bar{J}\big(x(k+N)\big) = \sum_{i=1}^{M} \bar{J}_i(x_i(k+N))$.

To optimize (8) in a distributed manner, we construct a local barrier-based control policy for subsystem Σ_i:

$$u_i(x_{\mathcal{N}_i}) = v_i(x_{\mathcal{N}_i}) + L_i \nabla\mathcal{B}_i(x_i) \tag{9}$$

where $v_i \in \mathbb{R}^{m_i}$ is a new virtual control policy, $L_i \in \mathbb{R}^{m_i \times n_i}$ is a decision variable to be further optimized (deferred in Section III-B).

Remark 1: The second term in (9) is to generate a local repulsive force associated with the i-th agent, which grows as the state x_i moves toward the boundary of $\mathcal{X}_i$. In doing so, the local state x_i can be restricted in their interiors, i.e., safety is guaranteed by the gradients of the barrier functions of state constraints in (9). ▲

At any time instant $\tau \in [k, k+N-1]$, let $\bar{J}^*(x(\tau))$ be an optimal value function under the control structure (9), then one can write the discrete-time HJB equation as

$$\bar{J}^*(x(\tau)) = \min_{u_i(x_{\mathcal{N}_i}(\tau)), i \in \mathbb{N}_1^M} \bar{r}(\tau) + \bar{J}^*\big(x(\tau+1)\big), \tag{10}$$

for each local optimal control policy:

$$u_i^*(x_{\mathcal{N}_i}(\tau)) = \underset{u_i(\tau)}{\operatorname{argmin}} \{\bar{r}_i(\tau) + \sum_{j \in \bar{\mathcal{N}}_i} \bar{J}_j^*\big(x_{\mathcal{N}_j}(\tau+1)\big)\} \tag{11}$$

for $i \in \mathbb{N}_1^M$, where $\bar{\mathcal{N}}_i$ is the collection of subsystems with the i-th subsystem its neighbor.

Different from the classical numerical optimization methods, we present in the following the SL-DMPC algorithm to solve (10) and (11), see Algorithm 1.

Algorithm 1: Pseudocode of SL-DMPC

```
Require: $\bar{J}_i^0(\tau)$, $\bar{\epsilon} > 0$, safe policy $\boldsymbol{u}_i^0$, $i \in \mathbb{N}_1^M$, $t = 0$.
    for $k = 1, 2, \cdots$ do loop
 1: while $\sum_{\tau=k}^{k+N-1} \bar{J}_i^t(\tau) - \bar{J}_i^{t-1}(\tau) \geq \bar{\epsilon}$ do
 2:   for $\tau = k, \cdots, k+N-1$ do
 3:     Receive the neighbor state $x_{\mathcal{N}_i}(\tau)$.
 4:     Compute $x_i(\tau+1)$ with $u_i^t(x_{\mathcal{N}_i}(\tau))$ using (3);
 5:     Distributed policy evaluation:
          $\bar{J}_i^{t+1}(x_{\mathcal{N}_i}(\tau)) = \bar{r}_i(\tau) + \bar{J}_i^t(x_{\mathcal{N}_i}(\tau+1)).$   (12a)
 6:     Receive the cost $\bar{J}_j^{t+1}\big(x_{\mathcal{N}_j}(\tau+1)\big)$ for $j \in \bar{\mathcal{N}}_i$.
 7:     Distributed barrier-based policy update:
          $(v_i, L_i)^{t+1} = \underset{v_i, L_i}{\operatorname{argmin}} \{\bar{r}_i(\tau) + \sum_{j \in \bar{\mathcal{N}}_i} \bar{J}_j^{t+1}\big(x_{\mathcal{N}_j}(\tau+1)\big)\}$
          $u_i^{t+1}(x_{\mathcal{N}_i}) = v_i^{t+1}(x_{\mathcal{N}_i}) + L_i^{t+1}\nabla\mathcal{B}_i(x_i(\tau)).$   (12b)
 8:   end for
 9:   $t \leftarrow t+1$.
10: end while
11: Update $x_i(k+1)$ by applying $u_i^t(x_{\mathcal{N}_i}(k))$ to (1).
```

In the following theorem, we prove that in each prediction horizon, the control policy in (12) is safe (see Definition 3), also the control policy and value function will eventually converge to their optimal values respectively, i.e., $u_i^t(x_{\mathcal{N}_i}(\tau)) \to u_i^*(x_{\mathcal{N}_i}(\tau))$ and $\bar{J}_i^t(\tau) \to \bar{J}_i^*(\tau)$ as $t \to +\infty$, $\forall i \in \mathbb{N}_1^M$.

Theorem 1 (Safety guarantee and convergence): Let $\boldsymbol{u}^0(k)$ be a safe policy and the initial value function $\bar{J}^0(\tau) \geq \bar{r}(x(\tau), u^0(\tau)) + \bar{J}^0(x(\tau+1))$, $\tau \in [k, k+N-1]$; then under iteration (12), it holds that

1) $\bar{J}^{t+1}(\tau) \leq V^t(\tau) \leq \bar{J}^t(\tau)$, where $V^t(\tau) = \bar{r}(x(\tau), u^t(\tau)) + \bar{J}^t(x(\tau+1))$;
2) $\boldsymbol{u}^t(k)$ is a safe control policy;
3) $\bar{J}^t(\tau) \to \bar{J}^*(\tau)$ for all $\tau \in [k, k+N]$, as $t \to +\infty$. ■

Proof. 1): First, by collecting the iterative step (12a) for all $i \in \mathbb{N}_1^M$ results in the following centralized form

$$\bar{J}^{t+1}(x(\tau)) = \bar{r}(\tau) + \bar{J}^t(x(\tau+1)) \tag{13a}$$

Moreover, since u_i is only related to $x_{\mathcal{N}_i}$, (12b) is equivalent to

$$u_i^{t+1}(x_{\mathcal{N}_i}(\tau)) = \underset{u_i(x_{\mathcal{N}_i}(\tau))}{\operatorname{argmin}} \left\{\bar{r}_i(\tau) + \bar{J}^{t+1}\big(x(\tau+1)\big)\right\} \quad (13b)$$

and equivalent to the centralized form of the policy update, i.e.,

$$u^{t+1}(x(\tau)) = \underset{u_i(x_{\mathcal{N}_i}(\tau)),\, i\in\mathbb{N}_1^M}{\operatorname{argmin}} \left\{\bar{r}(\tau) + \bar{J}^{t+1}\big(x(\tau+1)\big)\right\} \quad (13c)$$

Then, one can apply the proof arguments in [31] for the centralized system, which proves that $\bar{J}^{t+1}(x(\tau)) \leq V^t(x(\tau)) \leq \bar{J}^t(x(\tau))$, for all $\tau \in [k, k+N-1]$.

Moreover, in line with [31], the second and third points can be naturally proven. Interested readers may refer to [31] for more details. □

B. Distributed Barrier Function-based RL for DMPC

In the following, Algorithm 1 is implemented with a distributed barrier function-based RL algorithm using actor-critic framework [16]. Compared with numerical solution of DMPC, such an implementation has the merit of generating a local explicit state-feedback control policy for each agent, rather than an open-loop control sequence. This allows the direct deployment of the learned control policy offline. The proposed distributed barrier-based RL consists of M actor-critic network pairs, each one designed for each local agent to learn the local control policy and the value function of information exchange among its neighbor agents. For any subsystem $i \in \mathbb{N}_1^M$, the critic network is constructed as

$$\hat{\bar{J}}_i(x_{\mathcal{N}_i}(\tau)) = (W_{c,i}^1)^\top \sigma_{c,i}(x_{\mathcal{N}_i}(\tau), \tau) + W_{c,i}^2 \mathcal{B}_i(x_i(\tau)), \quad (14)$$

where $W_{c,i}^1 \in \mathbb{R}^{N_{c,i}\times n_i}$ is the weighting matrix, $\sigma_{c,i} \in \mathbb{R}^{N_{c,i}}$ is a vector composed of activation functions, $W_{c,i}^2 \in \mathbb{R}$.

Let $h_{c,i} = (\sigma_{c,i}(x_{\mathcal{N}_i}(\tau),\tau), \mathcal{B}_i(x_i(\tau)))$, $(W_{c,i})^\top = [(W_{c,i}^1)^\top \; W_{c,i}^2]$, one can write

$$\hat{\bar{J}}_i(x_{\mathcal{N}_i}(\tau)) = (W_{c,i})^\top h_{c,i}(x_{\mathcal{N}_i}(\tau), \tau). \quad (15)$$

To minimize the deviation between $\bar{J}_i^*$ and $\hat{\bar{J}}_i$, in view of (12a), we define the following $\bar{J}_i^d$ as the desired value of $\hat{\bar{J}}_i$, such that

$$\bar{J}_i^d(x_{\mathcal{N}_i}(\tau)) = \; \bar{r}_i(\tau) + \hat{\bar{J}}_i(x_{\mathcal{N}_i}(\tau+1)), \quad (16)$$

for $\tau \in [k, k+N-1]$, where $\hat{\bar{J}}_i(x_{\mathcal{N}_i}(k+N)) = \|x_i(k+N)\|_{P_i}^2 + \mu\mathcal{B}_i(x_i(k+N))$.

Let $\epsilon_{c,i} = \bar{J}_i^d(x_{\mathcal{N}_i}(\tau)) - \hat{\bar{J}}_i(x_{\mathcal{N}_i}(\tau))$, $\forall i \in \mathbb{N}_1^M$ be the local approximation error. The following quadratic cost is to be minimized for each $i \in \mathbb{N}_1^M$, i.e.,

$$\delta_{c,i}(\tau) = q_{c,1}\epsilon_{c,i}(\tau)^2 + q_{c,2}\epsilon_{c,i}(k+N)^2, \quad (17)$$

where $q_{c,1}$, $q_{c,2} > 0$ are tuning parameters. Minimizing (17) leads to the update rule of the weighting matrix $W_{c,i}$ as

$$W_{c,i}(\tau+1) = W_{c,i}(\tau) - \gamma_{c,i}\frac{\partial \delta_{c,i}(\tau)}{\partial W_{c,i}(\tau)} \quad (18)$$

where $\gamma_{c,i}$ is the local learning rate.

Likewise, we construct the local barrier-based actor for each agent, i.e.,

$$u_i(x_{\mathcal{N}_i}(\tau)) = W_{a,i}^\top \sigma_{a,i}(x_{\mathcal{N}_i}(\tau), \tau) + \hat{L}_i \nabla\mathcal{B}_i(x_i(\tau)), \quad (19)$$

$i \in \mathbb{N}_1^M$, where $W_{a,i} \in \mathbb{R}^{N_{u,i}\times m_i}$ is the weighting matrix, $\sigma_{a,i} \in \mathbb{R}^{N_{u,i}}$ is a vector composed of activation functions, $\hat{L}_i \in \mathbb{R}^{m_i\times n_i}$ is the approximation of L_i. In view of (12b), we design a desired value of $u_i(\tau)$, i.e., $u_i^d(\tau)$ as

$$u_i^d(\tau) := -\frac{1}{2}R^{-1}\sum_{j\in\bar{\mathcal{N}}_i} g_i^\top(x_i)\frac{\partial \hat{\bar{J}}_j(x_{\mathcal{N}_j}(\tau+1))}{\partial x_i(\tau+1)} \quad (20)$$

for $i \in \mathbb{N}_1^M$, $\tau \in [k, k+N-1]$.

Let $\epsilon_{a,i}(\tau) = u_i^d(\tau) - u_i(\tau)$, where $\epsilon_{a,i}$ is the approximation error. At each time instant $\tau \in [k, k+N-1]$, each agent minimizes $\epsilon_{a,i}(\tau)$ with a cost of quadratic type, i.e., $\delta_{a,i}(\tau) = \|\epsilon_{a,i}(\tau)\|^2$ leading to the update rule of $W_{a,i}$ and $\hat{L}_i$ as

$$W_{a,i}(\tau+1) = W_{a,i}(\tau) - \gamma_{a,i}\frac{\partial \delta_{a,i}(\tau)}{\partial W_{a,i}(\tau)} \quad (21a)$$

$$\hat{L}_i(\tau+1) = \hat{L}_i(\tau) - \gamma_{a,i}\frac{\partial \delta_{a,i}(\tau)}{\partial \hat{L}_i(\tau)} \quad (21b)$$

where $\gamma_{a,i}$ is the local learning rate.

The learning steps of the proposed distributed barrier function-based safe RL algorithm in each prediction horizon $[k, k+N]$ can be summarized as follows. For $\tau = k, k+1, \cdots, k+N-1$, the following steps are repetitively performed:

- Generate $x_i(\tau+1)$ using (1), $\forall i \in \mathbb{N}_1^M$.
- Calculate $\bar{J}_i^d(x_{\mathcal{N}_i}(\tau))$ with (16) and $u_i^d(\tau)$ with (20) using $x_{\mathcal{N}_i}(\tau+1)$ and $x_j(\tau)$, for all $j \in \bar{\mathcal{N}}_i$.
- Update in parallel $W_{c,i}(\tau+1)$ with (18) and $W_{a,i}(\tau+1)$, $\hat{L}_i(\tau+1)$ with (21).

After the learning process in the prediction horizon $[k, k+N]$, the first control action $u_i(x_{\mathcal{N}_i}(k))$ computed with (20) is to be applied to (1). Then at the subsequent prediction horizon $[k+1, k+N+1]$, the above learning process is repeated.

Remark 2: Note that the training process (18) and (21) of the RL algorithm only uses the partial model information $G(x)$. In a slight different way, one can change the role of the local critic for estimating the local state $\lambda_i = \partial\bar{J}_i(x_{\mathcal{N}_i})/\partial x_{\mathcal{N}_i}$ according to the dual heuristic programming [32], which can speed up the convergence process by also utilization of the model information $F(x)$. ▲

Remark 3: It should be highlighted that the distributed barrier function-based RL algorithm is trained and deployed in a completely distributed manner, i.e., the state information is communicated among neighbor agents in each training time instant for weight update of the actor-critic and in each deployment time instant for realizing distributed control. ▲

Now we give some discussions on the computational complexity issue. The computational complexity of the classic distributed linear MPC is roughly $O(MN(n_{\mathcal{N}_i}+m_i)n_{\mathcal{N}_i}^2)$ if a fast local MPC implementation like [33] is used. The main computational complexity of our approach in the training process is the update of (18), (21), and the forward model prediction, which is roughly $O(MN((n_{c,i}+n_{u,i}+n_{\mathcal{N}_i})n_{\mathcal{N}_i}))$ for linear systems. When directly deploying the learned control policy, the overall online computational complexity is reduced to $O(M(n_{u,i}n_{\mathcal{N}_i}))$ even for nonlinear multi-robot systems. In summary, the overall computational complexity grows linearly with the number of agents to go. ▲

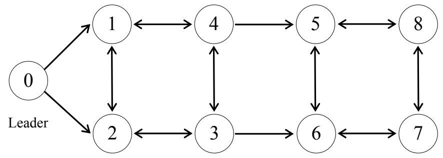

Fig. 1. Communication graph of mobile robots, where the arrows indicate the direction for information transmission of the mobile robots, and the robot 0 is the leading robot.

IV. Simulation and experimental results on formation control of mobile robots

A. Simulated experiments

Consider the formation control of M ($M = 8$) mobile robots with collision avoidance. The communication directed graph between the local mobile robots is shown in Fig. 1. The kinematic model of the i-th robot is given as

$$\dot{q}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_i \\ v_i \sin\theta_i \\ \omega_i \\ a_i \end{bmatrix} \tag{22}$$

where (x_i, y_i) is the coordinate of the i-th robot in Cartesian frame, θ_i and v_i are the yaw angle and the linear velocity respectively, $u_i = [a_i, \omega_i]^\top$ is the control input, where a_i and ω_i are the acceleration and yaw rate, respectively.

Let us define the formation error of the i-th robot in the local coordinate frame as

$$\begin{aligned} e_i = \ & T_i(\textstyle\sum_{j=1}^{M} c_{ij}\left(\Lambda_1(q_j - q_i) + \Delta h_{ji}\right) + \\ & s_i(q_r - q_i + \Delta h_{ri})) + \Lambda_2(q_r - q_i), \end{aligned} \tag{23}$$

where $a_{ij} = 0, 1$ represents the connection status, $a_{ij} = 1$ if robot i receives state information from robot j and $a_{ij} = 0$ otherwise; s_i represents the pinning gain, $s_i = 1$ if the robot i can receive the information of the leader, $\Lambda_1 = \text{diag}\{1, 1, 0, 0\}$, $\Lambda_2 = \text{diag}\{0, 0, 1, 1\}$. Δh_{ji} and Δh_{ri} are coordinate correction variable, which is determined by the formation shape and size; the coordinate transformation matrix is

$$T_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & I_2 \end{bmatrix}. \tag{24}$$

By discretizing (22) under the transformation (23), one can write the discrete-time local formation error model as

$$\begin{cases} e_{i,x}(k+1) = e_{i,x}(k) + \Delta t\big(\omega_i(k)e_{i,y}(k) - (h_i + s_i)v_i(k) \\ \qquad + \sum_{j\in\mathcal{N}_i, j\neq i} c_{ij}v_j(k)\cos\theta_{ji}(k) \\ \qquad + s_i v_r(k)\cos\theta_{ri}(k)\big) \\ e_{i,y}(k+1) = e_{i,y}(k) + \Delta t\big(-\omega_i(k)e_{i,y}(k) + \\ \qquad \sum_{j\in\mathcal{N}_i, j\neq i} c_{ij}v_j(k)\sin\theta_{ji}(k) + s_i v_r(k)\sin\theta_{ri}(k)\big) \\ e_{i,\theta}(k+1) = e_{i,\theta}(k) + \Delta t\left(\omega_r(k) - \omega_i(k)\right) \\ e_{i,v}(k+1) = e_{i,v}(k) + \Delta t\left(a_r(k) - a_i(k)\right), \end{cases} \tag{25}$$

where $(e_{i,x}, e_{i,y}, e_{i,\theta}, e_{i,v}) =: e_i$, $v_r = 1m/s$ and $\omega_r = 0rad/s$ are the reference linear velocity and angular velocity respectively, $h_i = \sum_{j\in\mathcal{N}_i, j\neq i} c_{ij}, \theta_{ji} = \theta_j - \theta_i, \theta_{ri} = \theta_r - \theta_i, \Delta t = 0.05s$ is the adopted sampling interval.

In the simulation process, the control objective is that the mobile robots can follow a straight line with a rectangular formation shape of four rows and two columns, as well as avoiding the obstacles on the path (see Fig. 2), where the desired distances between neighbor mobile robots in the same row and column were 2m and 1m respectively. All the obstacles to be avoided are circular objects with a diameter of 0.5m (see also Fig. 2). The coordinates of two obstacles were $(0, 2)$ and $(25, 0)$. In the proposed SL-DMPC, the penalty matrices were selected as $Q_{n_{\mathcal{N}_i}} = I_{n_{\mathcal{N}_i}}, R_i = I_2$ and the prediction horizon was set as $N = 20$. In the training process, the weighting matrices of the actor and critic were set as uniformly distributed random values and were updated at each time instant according to (18) and (21). The simulation was performed using MATLAB on a Laptop with Intel Core i7@2.30 GHz.

The simulation results in Fig. 2 show that the mobile robots can achieve a predefined formation shape from the disordered initial state condition, while avoiding all encountered obstacles on the path and recovering the formation shape after collision avoidance. Moreover, it is also observed in Fig. 2 that the formation error of each local robot converges to the origin.

B. Real-world experiments

We also tested our proposed algorithm on two real-world mobile robots for formation control with collision avoidance (see Fig. 3). In the experiment, we directly deployed the offline learned local policies to control the local mobile robots. At each sampling instant, the onboard satellite inertial guidance integrated positioning system measured each local state q_i, $i = 1, 2$. The measured q_1 and the corresponding reference q_r were transmitted from the first robot to the second by radio. Furthermore, the measured q_2 was transmitted from the second robot to the first one. Each robot was equipped with a Laptop in which the Unbuntu operating system was installed. In each laptop, the control input was computed in real-time by using the measured state information. The sampling interval was set as $\Delta t = 0.1$s. In the experiment, we added a static obstacle on the reference path. The first robot is expected to avoid the obstacle. To simplify the experimental setup, we assume the location information of the obstacle was pre-detected using some computer vision algorithms. Experimental results under local

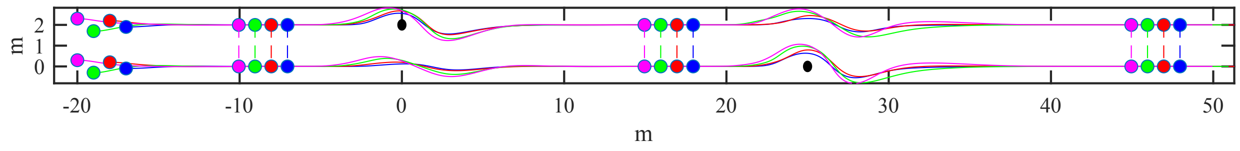

Fig. 2. Formation control of 8 mobile robots with collision avoidance in the simulation tests, where the black circular areas (0.5m in diameter) represent the obstacles, the colored lines represent the trajectories of mobile robots, meanwhile the mobile robots in the same column are marked with the same colored dots.

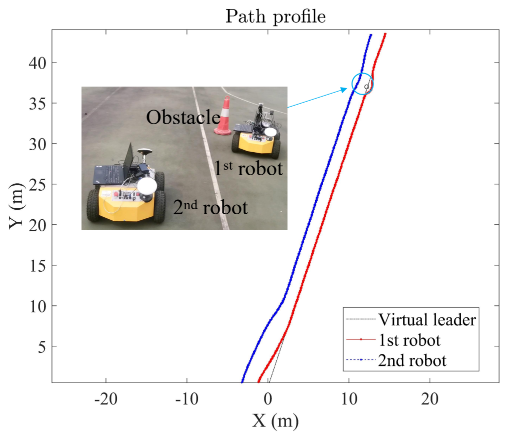

Fig. 3. Path profile of the mobile robots.

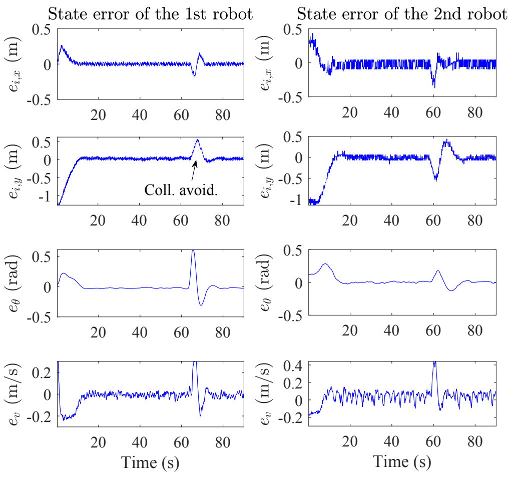

Fig. 4. State errors of the mobile robots.

collision avoidance were illustrated in Figs. 3 and 4, which show that the mobile robots can successfully avoid the obstacle. In the meanwhile, the formation shape was kept even in the collision avoidance process. Note that there was an evident asynchronous issue in the built software between the measured state and the one received from the radio, which could cause non-negligible uncertainties. Even so, in this case, our algorithm shows strong robustness (see also Fig. 4).

V. Conclusion

This paper proposed a barrier function-based safe reinforcement learning for distributed model predictive control (i.e., SL-DMPC) of nonlinear multi-robot systems with state constraints. As opposed to classic DMPC algorithms, the implementation of SL-DMPC relies on a distributed safe reinforcement learning algorithm based on the actor-critic framework, resulting for each agent a local explicit state-feedback control policy rather than an open-loop control sequence. These local control policies can be learned and deployed online synchronously or learned offline and deployed online, leading to significant computational load reduction concerning DMPC. As another prominent feature, we present a novel barrier-based control policy structure to guarantee safety in the distributed reinforcement learning algorithm. Moreover, the learning safety and convergence of the proposed barrier function-based safe reinforcement learning algorithm are proven in each prediction horizon. Both simulated and real-world experimental results on formation control of mobile robots with collision avoidance have validated the effectiveness of the proposed approach. The future direction will focus on the closed-loop recursive feasibility and stability guarantees of SL-DMPC and the application to formation transformation of mobile robots.

References

[1] Yun Ho Choi and Sung Jin Yoo. Minimal-approximation-based distributed consensus tracking of a class of uncertain nonlinear multiagent systems with unknown control directions. *IEEE Transactions on Cybernetics*, 47(8):1994–2007, 2017.

[2] Mariagrazia Dotoli, Hayfa Zgaya, Carmine Russo, and Slim Hammadi. A multi-agent advanced traveler information system for optimal trip planning in a co-modal framework. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2397–2412, 2017.

[3] Lara Brinón-Arranz, Alexandre Seuret, and Carlos Canudas-de Wit. Cooperative control design for time-varying formations of multi-agent systems. *IEEE Transactions on Automatic Control*, 59(8):2283–2288, 2014.

[4] Xinglong Zhang. Hierarchical and multilayer control structures based on MPC for large-scale systems. 2018.

[5] Alphan Ulusoy, Ozgur Gurbuz, and Ahmet Onat. Wireless model-based predictive networked control system over cooperative wireless network. *IEEE Transactions on Industrial Informatics*, 7(1):41–51, 2010.

[6] Jingyuan Zhan and Xiang Li. Consensus of sampled-data multi-agent networking systems via model predictive control. *Automatica*, 49(8):2502–2507, 2013.

[7] Bei Hou, Shaoyuan Li, and Yi Zheng. Distributed model predictive control for reconfigurable systems with network connection. *IEEE Transactions on Automation Science and Engineering*, 2021.

[8] Uglješa Todorović, José Ramón D Frejo, and Bart De Schutter. Distributed MPC for large freeway networks using alternating optimization. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[9] Qishao Wang, Zhisheng Duan, Yuezu Lv, Qingyun Wang, and Guanrong Chen. Distributed model predictive control for linear–Quadratic performance and consensus state optimization of multiagent systems. *IEEE Transactions on Cybernetics*, 51(6):2905–2915, 2020.

[10] Li Dai, Zhiwen Qiang, Zhongqi Sun, Tianyi Zhou, and Yuanqing Xia. Distributed economic MPC for dynamically coupled linear systems with uncertainties. *IEEE Transactions on Cybernetics*, 2020.

[11] Chao Shen and Yang Shi. Distributed implementation of nonlinear model predictive control for AUV trajectory tracking. *Automatica*, 115:108863, 2020.

[12] Carmen Amo Alonso, Nikolai Matni, and James Anderson. Explicit distributed and localized model predictive control via system level synthesis. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5606–5613. IEEE, 2020.

[13] Adedapo Odekunle, Weinan Gao, Masoud Davari, and Zhong-Ping Jiang. Reinforcement learning and non-zero-sum game output regulation for multi-player linear uncertain systems. *Automatica*, 112:108672, 2020.

[14] Xiangnan Zhong and Haibo He. GrHDP solution for optimal consensus control of multiagent discrete-time systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(7):2362–2374, 2018.

[15] Wei Wang, Xin Chen, Hao Fu, and Min Wu. Model-free distributed consensus control based on actor–critic framework for discrete-time nonlinear multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(11):4123–4134, 2018.

[16] Yi Jiang, Jialu Fan, Weinan Gao, Tianyou Chai, and Frank L Lewis. Cooperative adaptive optimal output regulation of nonlinear discrete-time multi-agent systems. *Automatica*, 121:109149, 2020.

[17] Hao Fu, Xin Chen, Wei Wang, and Min Wu. Data-based optimal synchronization control for discrete-time nonlinear heterogeneous multiagent systems. *IEEE Transactions on Cybernetics*, 2020.

[18] Huaipin Zhang, Dong Yue, Chunxia Dou, Wei Zhao, and Xiangpeng Xie. Data-driven distributed optimal consensus control for unknown multiagent systems with input-delay. *IEEE Transactions on Cybernetics*, 49(6):2095–2105, 2018.

[19] Qinglai Wei, Hongyang Li, Xiong Yang, and Haibo He. Continuous-time distributed policy iteration for multicontroller nonlinear systems. *IEEE Transactions on Cybernetics*, 51(5):2372–2383, 2020.

[20] Lu Dong, Jun Yan, Xin Yuan, Haibo He, and Changyin Sun. Functional nonlinear model predictive control based on adaptive dynamic programming. *IEEE Transactions on Cybernetics*, 2018.

[21] Xin Xu, Hong Chen, Chuanqiang Lian, and Dazi Li. Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances. *IEEE Transactions on neural networks and learning systems*, (99):1–12, 2018.

[30] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

[22] Xinglong Zhang, Jiahang Liu, Xin Xu, Shuyou Yu, and Hong Chen. Robust learning-based predictive control for discrete-time nonlinear systems with unknown dynamics and state constraints. *arXiv preprint arXiv:1911.09827*, 2019.

[23] Huaguang Zhang, Hongjing Liang, Zhanshan Wang, and Tao Feng. Optimal output regulation for heterogeneous multiagent systems via adaptive dynamic programming. *IEEE Transactions on neural networks and learning systems*, 28(1):18–29, 2015.

[24] Majid Mazouchi, Mohammad Bagher Naghibi-Sistani, and Seyed Kamal Hosseini Sani. A novel distributed optimal adaptive control algorithm for nonlinear multi-agent differential graphical games. *IEEE/CAA Journal of Automatica Sinica*, 5(1):331–341, 2017.

[25] Yang Lu, Yaohua Guo, Guoxiang Zhao, and Minghui Zhu. Distributed safe reinforcement learning for multi-robot motion planning. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pages 1209–1214. IEEE, 2021.

[26] Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(12):803–832, 2002.

[27] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. *arXiv preprint arXiv:1805.07708*, 2018.

[28] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

[29] Christian Conte, Colin N Jones, Manfred Morari, and Melanie N Zeilinger. Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. *Automatica*, 69:117–125, 2016.

[31] Biao Luo, Derong Liu, Tingwen Huang, and Jiangjiang Liu. Output tracking control based on adaptive dynamic programming with multistep policy evaluation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(10):2155–2165, 2017.

[32] Ganesh K Venayagamoorthy, Ronald G Harley, and Donald C Wunsch. Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator. *IEEE Transactions on Neural Networks*, 13(3):764–773, 2002.

[33] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2009.