This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3407632

Cooperative Multi-Agent Q-Learning using Distributed MPC

Hossein Nejatbakhsh Esfahani and Javad Mohammadpour Velni

***Abstract*— In this paper, we propose a cooperative Multi-Agent Reinforcement Learning (MARL) approach based on Distributed Model Predictive Control (DMPC). In the proposed framework, the local MPC schemes are formulated based on the dual decomposition method in the context of DMPC and will be used to derive the local state (and action) value functions required in a cooperative Q-learning algorithm. We further show that the DMPC scheme can yield a framework based on the Value Function Decomposition (VFD) principle so that the global state (and action) value functions can be decomposed into several local state (and action) value functions captured from the local MPCs. In the proposed cooperative MARL, the coordination between individual agents is then achieved based on the multiplier-sharing step, a.k.a inter-agent negotiation in the DMPC scheme.**

***Index Terms*— Multi-Agent Q-Learning, Distributed MPC, Cooperative Control**

I. Introduction

REINFORCEMENT Learning (RL) has become a popular machine learning method for Markov Decision Processes (MDPs) and seeks to improve the closed-loop performance of the control policy deployed on the MDPs as observations are collected [1]. Most RL methods use Deep Neural Networks (DNNs) to approximate the optimal policy or state (and action) value functions underlying the MDP. Motivated by the success of RL for single-agent systems, the RL community has recently started investigating Multi-Agent RL (MARL) and developed new algorithms for the control of networks of systems [2].

Authors in [3] adopted a value factorization based on the IGM (Individual-Global-Max) principle to implement a cooperative multi-agent Q-learning, where the individual value functions were parameterized by the DNN. To coordinate multiple agents, a cooperative Q-learning was developed based on the cooperative repeated games in [4]. A distributed actor-critic algorithm using DNNs as function approximators was proposed in [5]. A decentralized/distributed and collaborative MARL was proposed in [6] where a network of multiple agents aims to maximize the global reward function cooperatively. However, the structure of the proposed algorithms above are based on the DNN where these DNN-based MARL algorithms cannot provide a safe learning framework in order to handle the local agent constraints and ensure closed-loop robust stability. Furthermore, there is no systematic or physically meaningful way to choose the initial values and the number of hidden layers of a DNN.

In the context of single agent RL, [7] proposed the idea of using Model Predictive Control (MPC) schemes as value function/policy approximators and formally justified that an MPC scheme can generate the optimal state (and action) value functions and optimal policy underlying an MDP even if the MPC model does not capture the true system dynamics accurately. Successful applications built on this result include [8]–[10]. In this paper, we present an approach to provide a cooperative multi-agent framework for the MPC-based RL algorithms above. In the context of multi-agent systems, Distributed MPC (DMPC) is a well-known control approach to cope with challenges associated with interconnected systems. Dual decomposition is a method for solving DMPC problems, where a coupled constraint between the agents can be formulated as a dual problem [11].

The main contributions of this work are as follows: In the context of multi-agent Q-learning, we propose an approach to develop a distributed MPC-based multi-agent Q-learning, which allows us to decompose the global state (and action) value functions. Moreover, we use a DMPC scheme to approximate the local state (and action) value functions with lower complexity and computational efforts for learning the local MPCs in a cooperative manner. It is noted that the proposed cooperative MARL based on DMPC scheme can be formulated in both the Q-learning and Policy Gradient (PG) frameworks. More precisely, we show that the structure of the DMPC can be leveraged to introduce some local MPC-based value function approximators required in a cooperative Q-learning. In the context of PG, one can also use the same structure to capture the local optimal policies, which are delivered from the local MPC schemes.

This paper is organized as follows. In Section II, a distributed MPC scheme based on dual decomposition method is described. The DMPC parameterization and the idea of using DMPC as state (and action) value function approximators in the context of MARL are described in Section III. In Section IV, the performance of the proposed algorithm is illustrated using a numerical example.

II. Preliminaries and Problem Statement

In this section, we provide formulations for centralized control problem and the use of dual decomposition method to break the centralized problem into several local optimization problems.

This work was supported by the US National Science Foundation under award #2302219.

Authors are with the Department of Mechanical Engineering, Clemson University, Clemson, SC, USA (e-mail: hnejatb@clemson.edu; javadm@clemson.edu).

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3407632

A. Centralized Control

Let the state and input of the agent i be denoted by $\boldsymbol{x}_i$ and $\boldsymbol{u}_i$, respectively. Suppose a deterministic model of the system dynamics is available as $\boldsymbol{x}_i^{k+1} = \boldsymbol{f}_i(\boldsymbol{x}_i^k, \boldsymbol{u}_i^k)$. A cooperative DMPC scheme can be based on a centralized optimization problem or set of decentralized problems that need to be solved at time instant k. Let us consider a cooperative multi-agent system with m agents and inequality coupling constraints as $C\boldsymbol{x}^k \leq \boldsymbol{c}$, where $x^k = \left[\boldsymbol{x}_1^k, \cdots, \boldsymbol{x}_m^k\right]^\top$, where $\boldsymbol{x}^k \in \mathbb{R}^{n_x}$, $n_x = \sum_{i=1}^m n_{x_i}$. The matrix $C \in \mathbb{R}^{n_c \times n_x}$, and $\boldsymbol{c} \in \mathbb{R}^{n_c}$ describes n_c coupling constraints. The centralized optimal control problem can be cast as [12]

$$\min_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \sum_{i=1}^{m} T_i\left(\hat{\boldsymbol{x}}_i^{k+N}\right) + \sum_{\ell=k}^{k+N-1} l_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right) \tag{1a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{x}}_i^{\ell+1} = \boldsymbol{f}_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right), \quad \hat{\boldsymbol{x}}_i^{k} = \boldsymbol{x}_i^{k}, \tag{1b}$$

$$h_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right) \leq 0, h_i\left(\hat{\boldsymbol{x}}_i^{k+N}\right) \leq 0, g_i\left(\hat{\boldsymbol{u}}_i^{\ell}\right) \leq 0, \tag{1c}$$

$$C\hat{\boldsymbol{x}}^{\ell} \leq \boldsymbol{c} \tag{1d}$$

where N is the prediction horizon, and T_i, l_i, h_i and g_i denote the respective terminal cost, stage cost, mixed inequality constraint and input inequality constraint for agent i. Solving (1) yields a sequence of optimal input predictions and corresponding state predictions as

$$\begin{aligned} \hat{\boldsymbol{u}}^\star &= \{(\hat{\boldsymbol{u}}_1^{k:k+N-1})^\star, \ldots, (\hat{\boldsymbol{u}}_m^{k:k+N-1})^\star\} \\ \hat{\boldsymbol{x}}^\star &= \{(\hat{\boldsymbol{x}}_1^{k:k+N})^\star, \ldots, (\hat{\boldsymbol{x}}_m^{k:k+N})^\star\} \end{aligned} \tag{2}$$

where the first element $(\hat{\boldsymbol{u}}_i^k)^\star$ of the input sequence $\hat{\boldsymbol{u}}_i^\star$ is applied to each agent. At each physical time instant k, a new state $\boldsymbol{x}_i^k$ is received, and problem (1) is solved again, producing a new $\hat{\boldsymbol{u}}_i^\star$ and $(\hat{\boldsymbol{u}}_i^k)^\star$ for each agent. However, repeatedly solving the centralized MPC problem (1) can fail for large-scale systems where the communication bandwidth is restricted. To address this issue, we use a decentralized DMPC scheme based on dual decomposition. We show next how to modify (1) to arrive at a decentralized problem.

B. Dual Decomposition

We can relax the coupling constraints (1d) of the centralized optimization problem (1) by introducing the associated Lagrange multipliers $\boldsymbol{\mu}^\ell \in \mathbb{R}^{n_c}$. Let $\boldsymbol{\Omega} \in \mathbb{R}^{Nn_c}$ be a compact representation of the Lagrange multipliers such that $\boldsymbol{\Omega} = \left[(\boldsymbol{\mu}^k)^\top, \cdots, (\boldsymbol{\mu}^{k+N})^\top\right]^\top$. We then rewrite the original problem (1) as

$$V_\Omega(\boldsymbol{\Omega}, \boldsymbol{s}) = \tag{3a}$$

$$\min_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \sum_{i=1}^{m} \left(T_i\left(\hat{\boldsymbol{x}}_i^{k+N}\right) + \sum_{\ell=k}^{k+N-1} l_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right) \right) + \sum_{\ell=k}^{k+N} (\boldsymbol{\mu}^\ell)^\top \left(C\hat{\boldsymbol{x}}^{\ell} - \boldsymbol{c}\right)$$

$$\text{s.t.} \quad \hat{\boldsymbol{x}}_i^{\ell+1} = \boldsymbol{f}_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right), \quad \hat{\boldsymbol{x}}_i^{k} = \boldsymbol{s}_i^{k}, \tag{3b}$$

$$h_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right) \leq 0, h_i\left(\hat{\boldsymbol{x}}_i^{k+N}\right) \leq 0, g_i\left(\hat{\boldsymbol{u}}_i^{\ell}\right) \leq 0 \tag{3c}$$

where $\boldsymbol{s}_i^k, i = 1, \ldots, m$ are the local states of the cooperative agents at every time instant k and $V_\Omega(\boldsymbol{\Omega}, \boldsymbol{s})$ reads as an optimal value of the problem (3). Then, the problem (3) is completely separable as both objective and constraints can be distributed among the m agents. Indeed, the objective function is separable as the matrix C can be split into blocks $C = [C_1, \cdots, C_m]$ such that $C\hat{\boldsymbol{x}}^k = \sum_{i=1}^m C_i \hat{\boldsymbol{x}}_i^k$, where $C_i \in \mathbb{R}^{n_c \times n_{x_i}}$. To evaluate a subgradient of $V_\Omega(\boldsymbol{\Omega}, \boldsymbol{s})$, the problem (3) is solved for a given realization of $\boldsymbol{\Omega}$. We then take the derivative of the objective function of problem (3) with respect to $\boldsymbol{\Omega}$ such that a subgradient of V_Ω is obtained as

$$\left[(C\hat{\boldsymbol{x}}^{\star,k} - \boldsymbol{c})^\top, \cdots, (C\hat{\boldsymbol{x}}^{\star,k+N} - \boldsymbol{c})^\top\right]^\top \in \partial\left(V_\Omega\right)\left(\Omega^k\right) \tag{4}$$

where $\partial\left(V_\Omega\right)\left(\boldsymbol{\Omega}^k, \boldsymbol{s}\right)$ denotes the subdifferential of V_Ω at $\boldsymbol{\Omega}$. The dual of the original centralized problem (1) can be formulated as

$$\max_{\boldsymbol{\Omega} \geq 0} \quad V_\Omega(\boldsymbol{\Omega}, \boldsymbol{s}). \tag{5}$$

Hence, the original problem (1) can be solved in a distributed manner by solving its dual using a subgradient method. In a subgradient approach, steps of appropriate length are taken in the direction of a subgradient of the dual problem which corresponds to iteratively updating the Lagrange multipliers Ω^k. This method can then be implemented in a distributed manner since a subgradient of the dual problem (5) is given by (4), which is separable as $C\hat{\boldsymbol{x}}^{\star,k:k+N} = \sum_{i=1}^m C_i \hat{\boldsymbol{x}}_i^{\star,k:k+N}$. The local optimization problem can then be expressed as

$$\min_{\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i} T_i\left(\hat{\boldsymbol{x}}_i^{k+N}, \boldsymbol{\mu}^{k+N}\right) + \sum_{\ell=k}^{k+N-1} l_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right) + (\boldsymbol{\mu}^\ell)^\top C_i \hat{\boldsymbol{x}}_i^{\ell} \tag{6a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{x}}_i^{\ell+1} = \boldsymbol{f}_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right), \quad \hat{\boldsymbol{x}}_i^{k} = \boldsymbol{s}_i^{k}, \tag{6b}$$

$$h_i\left(\hat{\boldsymbol{x}}_i^{\ell}, \hat{\boldsymbol{u}}_i^{\ell}\right) \leq 0, h_i\left(\hat{\boldsymbol{x}}_i^{k+N}\right) \leq 0, g_i\left(\hat{\boldsymbol{u}}_i^{\ell}\right) \leq 0 \tag{6c}$$

where the multipliers at every inter-agent negotiation can be updated using a projected subgradient step as

$$\boldsymbol{\Omega} \leftarrow \max\left(0, \boldsymbol{\Omega} + \beta \boldsymbol{Z}\right), \tag{7a}$$

$$\boldsymbol{Z} = \sum_{i=1}^{m} C_i \hat{\boldsymbol{x}}_i^{\star,k:k+N} - \boldsymbol{c}, \tag{7b}$$

where $\beta > 0$ is the step size and $\boldsymbol{Z} \in \mathbb{R}^{Nn_c}$. Note that the projected subgradient above is a method to solve the optimization problem (5) by iteratively updating the multipliers. More specifically, the equations above describe how a consensus on the coupling constraints can be reached when the corresponding Lagrangian multipliers converge to their optimal values $\boldsymbol{\Omega}^\star$ after some iterations, e.g., the inter-agent negotiation between the agents is terminated, e.g., $\max(\boldsymbol{Z}) \leq \epsilon$.

It is worth mentioning that one can adopt Newton's methods to speed up the inter-agent negotiation by exploiting curvature information in addition to the gradient in the dual decomposition framework [13]. Moreover, an Alternative Direction Method of Multipliers (ADMM) may be leveraged in the DMPC scheme to improve convergence properties in terms of

speed and stability [14]. However, the basic concept behind the proposed DMPC-based cooperative Q-learning is still valid using the methods above.

III. Proposed DMPC-based Q-Learning Method

In this section, we describe our developed multi-agent Q-learning method based on the DMPC scheme to approximate the state (and action) value functions with local MPC schemes.

A. Local Approximators based on DMPC

To capture the local value functions in a distributed and cooperative manner, we formulate a parameterized version of (6) at $\boldsymbol{\Omega}^\star$ as

$$V_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{\Omega}^\star\right) = \min_{\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i, \boldsymbol{\sigma}_i} \gamma^N T_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^{k+N}, \boldsymbol{\sigma}_i^{k+N}, \boldsymbol{\mu}^{\star,k+N}\right) \tag{8a}$$

$$+ \sum_{\ell=k}^{k+N-1} \gamma^{\ell-k} L_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^\ell, \hat{\boldsymbol{u}}_i^\ell, \boldsymbol{\mu}^{\star,\ell}, \boldsymbol{\sigma}_i^\ell\right)$$

$$\text{s.t.} \quad \hat{\boldsymbol{x}}_i^{\ell+1} = \boldsymbol{f}_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^\ell, \hat{\boldsymbol{u}}_i^\ell\right), \quad \hat{\boldsymbol{x}}_i^k = \boldsymbol{s}_i^k, \tag{8b}$$

$$h_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^\ell, \hat{\boldsymbol{u}}_i^\ell\right) \le \boldsymbol{\sigma}_i^\ell, \quad h_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^{k+N}\right) \le \boldsymbol{\sigma}_i^{k+N}, \tag{8c}$$

$$g_i\left(\hat{\boldsymbol{u}}_i^{k+\ell-1}\right) \le 0, \quad \boldsymbol{\sigma}_i^\ell \ge 0, \quad \ell = k, \ldots, k+N \tag{8d}$$

where

$$L_i^{\theta_i} = l_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^\ell, \hat{\boldsymbol{u}}_i^\ell\right) + \left(\boldsymbol{\mu}^{\star,\ell}\right)^\top C_i^{\theta_i} \hat{\boldsymbol{x}}_i^\ell + \boldsymbol{p}^\top \boldsymbol{\sigma}_i^\ell \tag{9a}$$

$$T_i^{\theta_i} = T_{c,i}^{\theta_i}\left(\hat{\boldsymbol{x}}_i^N\right) + \left(\boldsymbol{\mu}^{\star,N}\right)^\top C_i^{\theta_i} \hat{\boldsymbol{x}}_i^N + \boldsymbol{p}^\top \boldsymbol{\sigma}_i^N \tag{9b}$$

where θ_i is a set of parameters assigned to the terminal cost, the stage cost, the model and the inequality constraints.

Remark 1. *Assume that constraints* $h_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^\ell, \hat{\boldsymbol{u}}_i^\ell\right) \le 0$ *and* $h_i^{\theta_i}\left(\hat{\boldsymbol{x}}_i^{k+N}\right) \le 0$ *are relaxed as per* (8c) *by introducing some slack variables* $\boldsymbol{\sigma}_i^\ell$*, and the term* $\sum_{\ell=1}^N \boldsymbol{p}^\top \boldsymbol{\sigma}_i^\ell$ *is added to the cost. Then, if* $\boldsymbol{p} < \infty$ *is large enough, the solution of* (8) *is unchanged whenever feasible and recursive feasibility is guaranteed [15].*

The solution of (8) yields the sequences of optimal input predictions $(\hat{\boldsymbol{u}}_i)^\star = \{(\hat{\boldsymbol{u}}_i^k)^\star, \ldots, (\hat{\boldsymbol{u}}_i^{k+N-1})^\star\}$ at $\boldsymbol{\Omega}^\star$. The first element defines the policy for each agent as

$$\boldsymbol{\pi}_{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{\Omega}^\star\right) = (\hat{\boldsymbol{u}}_i^k)^\star\left(\boldsymbol{s}_i^k, \boldsymbol{\Omega}^\star, \boldsymbol{\theta}_i\right). \tag{10}$$

We next consider this optimal policy delivered by the DMPC scheme as an action $\boldsymbol{a}_i^k$ in the context of reinforcement learning, which is selected according to the above policy with the possible addition of exploratory moves. Then, an action value function approximation can be formulated as

$$Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right) = \min_{\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i, \boldsymbol{\sigma}_i} \quad (8a) \tag{11a}$$

$$\text{s.t.} \quad (8b) - (8d), \quad \hat{\boldsymbol{u}}_i^k = \boldsymbol{a}_i^k \tag{11b}$$

Note that the proposed approximators (8) and (11) satisfy the fundamental equalities underlying the Bellman equations as

$$\boldsymbol{\pi}_{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{\Omega}^\star\right) = \arg\min_{\boldsymbol{a}_i^k} Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right), \tag{12a}$$

$$V_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{\Omega}^\star\right) = \min_{\boldsymbol{a}_i^k} Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right). \tag{12b}$$

B. Cooperative Q-Learning based on DMPC

In this section, we propose a cooperative classical off-policy Q-learning algorithm, which is based on the on-the-fly temporal-difference learning method. Let $\boldsymbol{s} = \operatorname{col}\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_m\} \in \mathcal{S}$ be defined as the state of the multi-agent system, where $\boldsymbol{s}_i \in \mathcal{S}_i$ denotes the state of agent i. Similarly, the action of the whole system is defined as $\boldsymbol{a} = \operatorname{col}\{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_m\} \in \mathcal{A}$, where $\boldsymbol{a}_i \in \mathcal{A}_i$ denotes the action of agent i. Let $L_i(\boldsymbol{s}, \boldsymbol{a})$ be the local cost of agent i. In a cooperative learning scenario, all agents aim to minimize the global cost function as

$$L\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) = \sum_{i=1}^m L_i\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right). \tag{13}$$

Let $V^N\left(\boldsymbol{x}^k\right)$ and $V_i^{N,I}\left(\boldsymbol{x}_i^k\right)$ be the value functions associated with the centralized and decentralized problems (1) and (6), respectively. Considering $\boldsymbol{a}^k, \boldsymbol{a}_i^k$ as additional constraints for these problems, the corresponding action value functions are $Q^N\left(\boldsymbol{x}^k, \boldsymbol{a}^k\right)$ and $Q_i^{N,I}\left(\boldsymbol{x}_i^k, \boldsymbol{a}_i^k\right)$. Note that I denotes the iteration number for the inter-agent negotiation stage. According to the duality theorem, we then have that

$$\sum_{i=1}^m V_i^{N,I}\left(\boldsymbol{x}_i^k\right) \le V^N\left(\boldsymbol{x}^k\right), \tag{14a}$$

$$\sum_{i=1}^m Q_i^{N,I}\left(\boldsymbol{x}_i^k, \boldsymbol{a}_i^k\right) \le Q^N\left(\boldsymbol{x}^k, \boldsymbol{a}^k\right), \tag{14b}$$

for any I. However, we propose to use the parameterized DMPC schemes to approximate the optimal state (and action) value functions above. Hence, the global state (and action) value functions delivered by a centralized MPC parameterized in θ can be decomposed into the sum of local state (and action) value functions for each agent delivered by (8) and (11) at $\boldsymbol{\Omega}^\star$ as

$$V^\star\left(\boldsymbol{s}^k\right) \approx V^\theta\left(\boldsymbol{s}^k\right) = \sum_{i=1}^m V_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{\Omega}^\star\right), \tag{15a}$$

$$Q^\star\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) \approx Q^\theta\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) = \sum_{i=1}^m Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right) \tag{15b}$$

Let $Q^\theta\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right)$ and $Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right)$ be the approximated global and local action value functions associated to the centralized and distributed MPC schemes, respectively. Proposition 1 then shows consistency between the centralized policy and the distributed policy. According to (15), the proposed parameterized state (and action) value functions captured from the DMPC scheme satisfy the additive decomposition required in the Individual-Global-Max (IGM) principle [16]. We next show that the IGM principle holds in the proposed DMPC-based cooperative Q-learning. It is noted that the IGM condition is

introduced as Individual-Global-Min (IGMi) in our framework, in which the aim is to minimize the global stage cost function.

Proposition 1. *Considering the state (and action) value decomposition* (15)*, the IGMi principle holds such that*

$$\arg\min_{\boldsymbol{a}^k} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) = \begin{pmatrix} \arg\min_{\boldsymbol{a}_1^k} Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right) \\ \vdots \\ \arg\min_{\boldsymbol{a}_m^k} Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right) \end{pmatrix} \tag{16}$$

Proof. Let us consider the global action value function associated with the centralized multi-agent MPC with m agents as

$$\begin{aligned} &Q^{\theta}\left(Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right), \cdots, Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right)\right) \\ &\geq Q^{\theta}\left(\min_{\boldsymbol{a}_1^k} Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right), \cdots, Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right)\right) \\ &\cdots \geq \\ &Q^{\theta}\left(\min_{\boldsymbol{a}_1^k} Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right), \cdots, \min_{\boldsymbol{a}_m^k} Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right)\right). \end{aligned} \tag{17}$$

Hence, the mixing action value function constructed based on the additive decomposition has a unified minimizer as

$$\begin{aligned} &\min_{\boldsymbol{a}^k} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) := \\ &\min_{\boldsymbol{a}^k} Q^{\theta}\left(Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right), \cdots, Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right)\right) = \\ &Q^{\theta}\left(\min_{\boldsymbol{a}_1^k} Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right), \cdots, \min_{\boldsymbol{a}_m^k} Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right)\right) \end{aligned} \tag{18}$$

Adopting the multi-agent policy as

$$\boldsymbol{\pi}_{\theta} = \left(\boldsymbol{\pi}_{\theta_1}, \cdots, \boldsymbol{\pi}_{\theta_m}\right) = \begin{pmatrix} \arg\min_{\boldsymbol{a}_1^k} Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right) \\ \vdots \\ \arg\min_{\boldsymbol{a}_m^k} Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right) \end{pmatrix} \tag{19}$$

and considering the Bellman equalities (12), the following result then holds

$$\begin{aligned} &Q^{\theta}\left(Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{\pi}_{\theta_1}, \boldsymbol{\Omega}^\star\right), \cdots, Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{\pi}_{\theta_m}, \boldsymbol{\Omega}^\star\right)\right) = \\ &Q^{\theta}\left(V_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{\Omega}^\star\right), \cdots, V_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{\Omega}^\star\right)\right) = \\ &Q^{\theta}\left(\min_{\boldsymbol{a}_1^k} Q_1^{\theta_1}\left(\boldsymbol{s}_1^k, \boldsymbol{a}_1^k, \boldsymbol{\Omega}^\star\right), \cdots, \min_{\boldsymbol{a}_m^k} Q_m^{\theta_m}\left(\boldsymbol{s}_m^k, \boldsymbol{a}_m^k, \boldsymbol{\Omega}^\star\right)\right) \\ &\qquad = \min_{\boldsymbol{a}^k} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) \end{aligned} \tag{20}$$

Thus, $\boldsymbol{\pi}_{\theta} = \arg\min_{\boldsymbol{a}^k} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right)$, which proves the proposition. ■

Proposition 2. *Let us define the global Temporal Difference (TD) error at the time instant k as*

$$\delta^k = L\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) + \gamma V^{\theta}\left(\boldsymbol{s}^{k+1}\right) - Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right), \tag{21}$$

where $0 < \gamma \leq 1$ is the discount factor. The local parameter update is then obtained as

$$\theta_i \leftarrow \theta_i + \alpha \delta^k \nabla_{\boldsymbol{\theta}_i} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right), \tag{22}$$

where the scalar $\alpha > 0$ is the learning rate and

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_i} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) = &\frac{\partial Q_i^{\theta_i}}{\partial \boldsymbol{\theta}_i} + \frac{\partial Q_i^{\theta_i}}{\partial \boldsymbol{\Omega}^\star} \frac{\partial \boldsymbol{\Omega}^\star}{\partial \boldsymbol{\theta}_i} \\ &+ \sum_{j=1, j\neq i}^{m} \frac{\partial Q_j^{\theta_j}}{\partial \boldsymbol{\Omega}^\star} \frac{\partial \boldsymbol{\Omega}^\star}{\partial \boldsymbol{\theta}_i}. \end{aligned} \tag{23}$$

Proof. Let us define the centralized gradient step as

$$\Delta\theta = \alpha \delta^k \nabla_{\boldsymbol{\theta}} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right). \tag{24}$$

The DMPC-based state (and action) value function approximation then provides an additive decomposition in (15) so that the gradient step above can be split as

$$\begin{aligned} \Delta\boldsymbol{\theta} = \begin{bmatrix} \Delta\boldsymbol{\theta}_1 \\ \vdots \\ \Delta\boldsymbol{\theta}_m \end{bmatrix} = &\begin{bmatrix} \alpha\delta^k \nabla_{\boldsymbol{\theta}_1} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) \\ \vdots \\ \alpha\delta^k \nabla_{\boldsymbol{\theta}_m} Q^{\theta}\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right) \end{bmatrix} = \\ &\begin{bmatrix} \sum_{i=1}^{m} \alpha\delta^k \nabla_{\boldsymbol{\theta}_1} Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right) \\ \vdots \\ \sum_{i=1}^{m} \alpha\delta^k \nabla_{\boldsymbol{\theta}_m} Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right) \end{bmatrix}. \end{aligned} \tag{25}$$

It is obvious that the DMPC scheme provides a connection line between the agents via multiplier-sharing. Therefore, the local sensitivities above are delivered using the chain rule as

$$\nabla_{\boldsymbol{\theta}_i} Q_j^{\theta_j}\left(\boldsymbol{s}_j^k, \boldsymbol{a}_j^k, \boldsymbol{\Omega}^\star\right) = \frac{\partial Q_j^{\theta_j}}{\partial \boldsymbol{\Omega}^\star} \frac{\partial \boldsymbol{\Omega}^\star}{\partial \boldsymbol{\theta}_i}, \tag{26a}$$

$$\nabla_{\boldsymbol{\theta}_i} Q_i^{\theta_i}\left(\boldsymbol{s}_i^k, \boldsymbol{a}_i^k, \boldsymbol{\Omega}^\star\right) = \frac{\partial Q_i^{\theta_i}}{\partial \boldsymbol{\theta}_i} + \frac{\partial Q_i^{\theta_i}}{\partial \boldsymbol{\Omega}^\star} \frac{\partial \boldsymbol{\Omega}^\star}{\partial \boldsymbol{\theta}_i}. \tag{26b}$$

Considering the equations (25) and (26), the local gradient step (23) holds. ■

In the above cooperative TD learning algorithm, the baseline stage cost $L\left(\boldsymbol{s}^k, \boldsymbol{a}^k\right)$ (total reward in the context of multi-agent RL) is defined as a function of state-action pair in order to provide an evaluation signal upon the cooperative RL. More specifically, the baseline cost affects the multi-agent system behavior and control policy via RL parameter updating, where the global TD error is appeared. We next provide details on how to compute the above sensitivities.

C. Sensitivity Computation

To compute the sensitivities needed in the proposed DMPC-based multi-agent Q-learning, let $\mathcal{L}_i^{\theta_i}$ be the local Lagrange function associated with the parameterized DMPC scheme (11) as

$$\mathcal{L}_i^{\theta_i} = \phi_i + \boldsymbol{\lambda}_{\text{eq}_i}^{\top} \boldsymbol{G}_i^{\theta_i} + \boldsymbol{\lambda}_{\text{ineq}_i}^{\top} \boldsymbol{H}_i^{\theta_i} \tag{27}$$

where ϕ_i, $\boldsymbol{\lambda}_{\text{eq}_i}$ and $\boldsymbol{\lambda}_{\text{ineq}_i}$ denote the total cost in (8a), the dual variables of the equality constraints and the dual variables of the inequality constraints, respectively. The vectors $G_i^{\theta_i}$ and $H_i^{\theta_i}$ collect the equality and inequality constraints, respectively. We then label $\boldsymbol{\Gamma}_i = \{\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{u}}_i, \boldsymbol{\sigma}_i\}$ the primal variables for the DMPC. The associated primal-dual variables then read as $\boldsymbol{z}_i = \left\{\boldsymbol{\Gamma}_i, \boldsymbol{\lambda}_{\text{eq}_i}, \boldsymbol{\lambda}_{\text{ineq}_i}\right\}$. The sensitivities of the action value function $Q_i^{\theta_i}$ captured from the DMPC scheme (11) w.r.t. the

parameters θ_i, and the optimal multipliers are then obtained by the sensitivity analysis as

$$\frac{\partial Q_i^{\theta_i}}{\partial \boldsymbol{\theta}_i} = \frac{\partial \mathcal{L}_i^{\theta_i}}{\partial \boldsymbol{\theta}_i}, \quad \frac{\partial Q_i^{\theta_i}}{\partial \boldsymbol{\Omega}^\star} = \frac{\partial \mathcal{L}_i^{\theta_i}}{\partial \boldsymbol{\Omega}^\star}, \tag{28}$$

where the optimal multipliers $\boldsymbol{\Omega}^\star$ are considered as additional parameters in the parametric Nonlinear Programming (NLP) so that the total parameters assigned to each parameterized DMPC scheme are $\{\boldsymbol{\theta}_i, \boldsymbol{\Omega}^\star\}$. To compute the sensitivity of the optimal multipliers $\boldsymbol{\Omega}^\star$ w.r.t. the parameters $\boldsymbol{\theta}_i$, one can use the Implicit Function Theorem (IFT) on the Karush Kuhn Tucker (KKT) conditions underlying the parametric NLP. It is noted that this sensitivity is globally obtained from a centralized sensitivity analysis upon the parameterized version of the centralized problem (1). Let $\boldsymbol{z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m, \boldsymbol{\Omega}^\star\}$ be the total primal-dual variables of (1). Assuming that Linear Independence Constraint Qualification (LICQ) and Second Order Sufficient Condition (SOSC) hold at $\boldsymbol{z}^\star$, then, the following holds:

$$\frac{\partial \boldsymbol{z}^\star}{\partial \boldsymbol{\theta}} = -\frac{\partial \kappa_\theta}{\partial \boldsymbol{z}}^{-1} \frac{\partial \kappa_\theta}{\partial \boldsymbol{\theta}}, \tag{29}$$

where $\kappa_\theta = \left[\nabla_{\boldsymbol{\Gamma}} \mathcal{L}^\theta, \boldsymbol{G}^\theta, \operatorname{diag}\left(\boldsymbol{\Lambda}_{\text{ineq}}\right) \boldsymbol{H}^\theta, (\boldsymbol{\Omega}^\star)^\top \boldsymbol{M}^\theta\right]^\top$ is the KKT matrix associated with the parameterized version of (1). Note that $\boldsymbol{M}^\theta$ gathers the coupling constraints (1d). We denote the global Lagrange function, the total equality constraints, the total inequality constraints and the dual variables of the inequality constraints by $\mathcal{L}^\theta$, $\boldsymbol{G}^\theta$, $\boldsymbol{H}^\theta$ and $\boldsymbol{\Lambda}_{\text{ineq}}$, respectively. As $\boldsymbol{\Omega}^\star$ is part of $\boldsymbol{z}^\star$, the corresponding sensitivity $\frac{\partial \boldsymbol{\Omega}^\star}{\partial \boldsymbol{\theta}_i}$ can be extracted from the gradient $\frac{\partial \boldsymbol{z}^\star}{\partial \boldsymbol{\theta}}$.

IV. Simulation Results

To examine the viability of the proposed DMPC-based cooperative Q-learning approach, we consider a heterogeneous multi-agent scenario where three linear systems with different dynamics must satisfy their local constraints and the coupling equality constraints (a desired distance between their first states) in a cooperative manner. We then consider only the equality part of the coupling inequality constraint (1d) as $C\hat{\boldsymbol{x}}^\ell = \boldsymbol{c}$. However, the local state constraints and coupling constraints will be violated due to model inaccuracies and disturbances. Let us consider three agents with the following dynamics

$$\boldsymbol{x}_1^{k+1} = \begin{bmatrix} 0.9 & 0.35 \\ 0 & 1.1 \end{bmatrix} \boldsymbol{x}_1^k + \begin{bmatrix} 0.0813 \\ 0.2 \end{bmatrix} u_1^k + \begin{bmatrix} e_1^k \\ 0 \end{bmatrix} \tag{30a}$$

$$\boldsymbol{x}_2^{k+1} = \begin{bmatrix} 0.91 & 0.33 \\ 0 & 0.98 \end{bmatrix} \boldsymbol{x}_2^k + \begin{bmatrix} 0.0611 \\ 0.23 \end{bmatrix} u_2^k \tag{30b}$$

$$\boldsymbol{x}_3^{k+1} = \begin{bmatrix} 0.88 & 0.3 \\ 0 & 1.1 \end{bmatrix} \boldsymbol{x}_3^k + \begin{bmatrix} 0.0837 \\ 0.21 \end{bmatrix} u_3^k \tag{30c}$$

and choose an imperfect model for three local MPC schemes as

$$\boldsymbol{x}^{k+1} = \begin{bmatrix} 1 & 0.25 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}^k + \begin{bmatrix} 0.0312 \\ 0.25 \end{bmatrix} u^k, \tag{31}$$

where the disturbance e_1^k is random, uncorrelated and uniformly distributed in the interval $[-0.1, 0]$. Let us label the states of the agents as $\boldsymbol{x}_1 = [x_{1,1}, x_{1,2}]^\top$, $\boldsymbol{x}_2 = [x_{2,1}, x_{2,2}]^\top$ and $\boldsymbol{x}_3 = [x_{3,1}, x_{3,2}]^\top$. We then consider the local constraints $0 \leq x_{1,1} \leq 0.5$, $0 \leq x_{2,1} \leq 2.5$ and $-2.5 \leq x_{3,1} \leq 0$ on the first agent, second agent and third agent, respectively. The control input constraint $-1 \leq u_{1,2,3} \leq 1$ is considered for all agents, and coupling constraints are defined as relative distances $d_{12} = 2$, $d_{13} = 2$ and $d_{23} = 4$. The global MARL cost function is then defined as

$$\begin{aligned} L(\boldsymbol{x}^k, \boldsymbol{u}^k) =& 10 \left\|\boldsymbol{x}_c\right\|_2^2 \\ &+ \sum_{i=1}^{m} L_i\left(\boldsymbol{x}_i^k, \boldsymbol{u}_i^k\right) + \boldsymbol{p}^\top \cdot \max\left(0, h_i\left(\boldsymbol{x}_i^k, \boldsymbol{u}_i^k\right)\right) \end{aligned} \tag{32}$$

where $\boldsymbol{x}_c = [x_{11} - x_{21} - d_{12}, x_{11} - x_{31} - d_{13}, x_{21} - x_{31} - d_{23}]^\top$. The local cost functions L_i can be, for instance, a quadratic function, and the penalty vector is set to $\boldsymbol{p} = [100, 100]$. To show the performance of the proposed cooperative Q-learning algorithm, we first run a learning stage shown in figures 1 and 2, where we use a total of 3×10^4 samples to adjust the local MPCs in a cooperative manner. As observed, the local constraint of the first agent $0 \leq x_{1,1}$ and the coupling equality constraints are violated at the beginning of the learning stage. We then observe that these violations are decreased during the learning, and finally constraint satisfaction is achieved at the end of the learning process. Hence, the global TD error and the global stage cost move towards zero as shown in Fig. 2. To demonstrate the performance of the parameterized MPC schemes after learning stage, we then consider different initial conditions for three agents and run 100 time instants as shown in figures 3 and 4. It is observed that the coupling constraints (solid lines) in Fig. 3 are satisfied while the agents with MPC schemes without learning (dashed lines) cannot satisfy these equality constraints at 2 and 4. As illustrated in Fig. 4, the first agent without learning has a constraint violation (dashed line) on $0 \leq x_{1,1}$ while this violation is disappeared (solid line) when the learned MPC is used. It is noted that a cooperative multi-agent system cannot be learned using the existing MPC-based RL approach as it has not been formulated to deal with cooperative MDPs. However, the proposed DMPC-based MARL addresses this issue so that the local MPCs are learned in a cooperative manner. As observed, the proposed DMPC-based MARL can improve the closed-loop performance, i.e., minimizing the total cost function L, as shown in Fig. 2.

V. Concluding Remarks

This work developed a multi-agent RL approach based on a DMPC scheme leading to a cooperative Q-learning algorithm. The dual decomposition method was adopted to make consensus between the agents for minimizing a global cost function in a cooperative manner. In the proposed DMPC-based Q-learning, the local state (and action) value functions were approximated by the local MPCs dedicated to each agent, and all agents were shown to agree on accomplishing a global task while the local constraints were satisfied as well.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

This article has been accepted for publication in IEEE Control Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2024.3407632

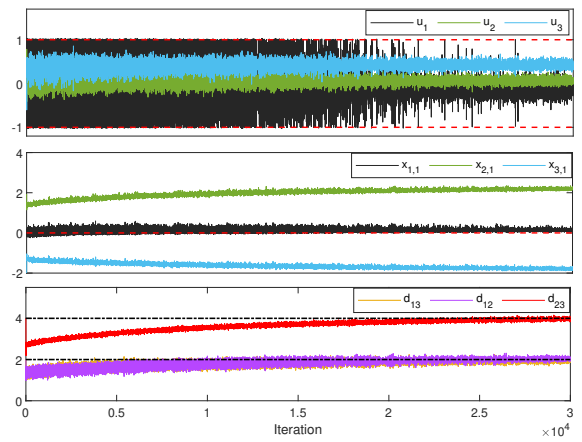

Fig. 1. In the first subplot, the constraint on control inputs $\boldsymbol{u_1, u_2, u_3}$ are illustrated. The evolution of the coupling states during the learning process is depicted in the second subplot. The constraint violation on the first state of the first agent $\boldsymbol{x_{1,1}}$ is decreased with more learning iteration. The third subplot shows the coupling constraints as the desired relative distance between the first states of all three agents.

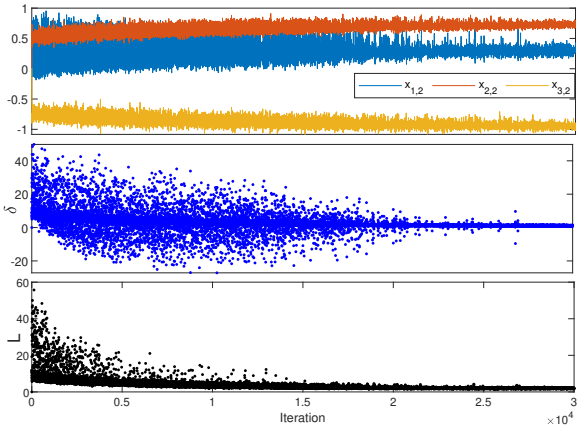

Fig. 2. The first subplot depicts the second states of all three agents, which are coupling-free. The evolution of the global temporal difference error $\boldsymbol{\delta}$ is shown in the second subplot. In the third subplot, it is observed that the total stage cost of multi-agent system is decreased with more learning iterations.

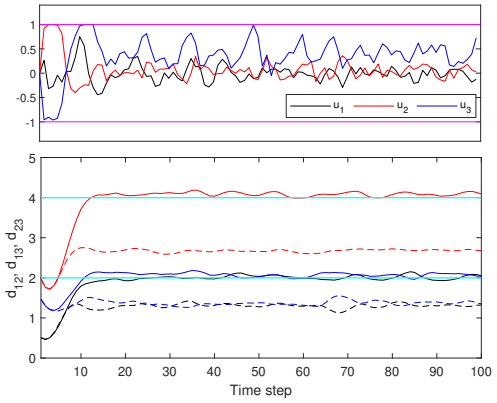

Fig. 3. Control inputs and coupling equality constraints showing that the coupling equality constraints (solid lines) are satisfied in the learned DMPC scheme while they cannot achieve their desired values without learning (dashed lines).

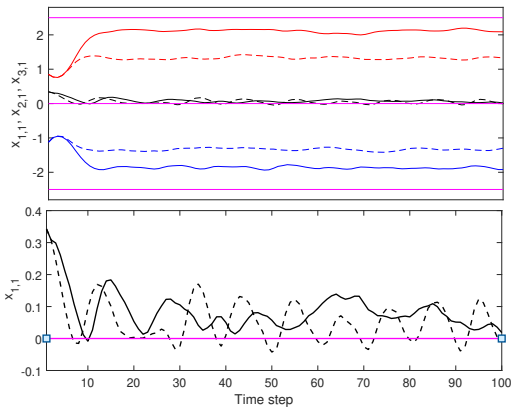

Fig. 4. Coupling states and constraint satisfaction/violation showing that the coupling states (solid lines) can approach the values required to satisfy the coupling constraints in the learned DMPC. Moreover, the constraint violation on the first state of the agent $\mathbf{1}$ is significantly decreased in the learned DMPC (solid line in the second subplot).

[2] L. Buşoniu, R. Babuška, and B. De Schutter, *Multi-agent Reinforcement Learning: An Overview*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 183–221.

[3] J. Wang, Z. Ren, B. Han, J. Ye, and C. Zhang, "Towards understanding cooperative multi-agent q-learning with value factorization," in *Neural Information Processing Systems*, 2020.

[4] Z. Zhang and D. Wang, "A multiagent q-learning algorithm for coordination of multiple agents," *Complexity, Hindawi*, vol. 2018, 2018.

[5] S. V. Macua, A. Tukiainen, D. G. Hernández, D. Baldazo, E. M. de Cote, and S. Zazo, "Diff-dac: Distributed actor-critic for multitask deep reinforcement learning," *CoRR*, vol. abs/1710.10363, 2017.

[6] Y. Lin, K. Zhang, Z. Yang, Z. Wang, T. Başar, R. Sandhu, and J. Liu, "A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 5562–5567.

[7] S. Gros and M. Zanon, "Data-driven economic nmpc using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2020.

[8] H. Nejatbakhsh Esfahani, A. Bahari Kordabad, and S. Gros, "Approximate robust nmpc using reinforcement learning," in *2021 European Control Conference (ECC)*, 2021, pp. 132–137.

[9] H. Nejatbakhsh Esfahani, A. Bahari Kordabad, W. Cai, and S. Gros, "Learning-based state estimation and control using mhe and mpc schemes with imperfect models," *European Journal of Control*, vol. 73, p. 100880, 2023.

[10] H. N. Esfahani and S. Gros, "Policy gradient reinforcement learning for uncertain polytopic lpv systems based on mhe-mpc," *IFAC-PapersOnLine*, vol. 55, no. 15, pp. 1–6, 2022, 6th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2022.

[11] D. Hammami, S. Maraoui, and K. Bouzrara, "Nonlinear distributed model predictive control with dual decomposition and event-based communication approach," *Transactions of the Institute of Measurement and Control*, vol. 42, no. 15, pp. 2929–2940, 2020.

[12] R. Negenborn, *Distributed model predictive control made easy*. Springer, 2014, vol. Intelligent Systems, Control and Automation: Science and Engineering, harvest.

[13] A. Kozma, E. Klintberg, S. Gros, and M. Diehl, "An improved distributed dual newton-cg method for convex quadratic programming problems," in *2014 American Control Conference*, 2014, pp. 2324–2329.

[14] Farokhi, F. and Shames, I. and Johansson, K. H., *Distributed MPC Via Dual Decomposition and Alternative Direction Method of Multipliers*, Maestre, José M. and Negenborn, Rudy R., Ed. Springer Netherlands, 2014.

[15] P. Scokaert and J. Rawlings, "Feasibility issues in linear model predictive control," *AIChE J.*, vol. 45, no. 8, p. 1649–1659, 1999.

[16] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," *ArXiv*, vol. abs/1905.05408, 2019.