Distributed Actor-Critic Multi-Task Learning with Shared and Task-Specific Parameters

Miloš S. Stanković, Marko Beko, Srdjan S. Stanković and Karl Henrik Johansson

***Abstract*—For Markov Decision Processes with finite state and action spaces a new two-time-scale multi-agent Actor-Critic algorithm for distributed consensus-based off-policy multi-task reinforcement learning (RL) is proposed. The algorithm is designed starting from the formulation of local criterion functions as the local value function approximations, and the global criterion function as a weighted sum of the local criteria. The Critic algorithm is in the form of the Gradient Temporal Difference GTD(1) algorithm working at a fast time-scale, while the new Actor algorithm, working at a slow time scale, is a consensus-based policy gradient algorithm obtained exactly from the global criterion function. The adopted information structure constraints allow direct access only to the local states, actions, and rewards. The Actor parameters include those shared between the agents, and those that are locally task-specific. Weak convergence of the algorithm to the limit set of an attached mean ordinary differential equation (ODE) is proved under mild conditions. Stability analysis of the obtained ODE based on vector Lyapunov functions and aggregate modeling is provided. The proposed algorithm can be considered as a tool for RL parallelization and fusion of complementary state explorations, increasing considerably the sample efficiency of single-agent schemes. Asymptotic convergence rate and variance reduction are studied by using stochastic differential equations. An experimental verification of the algorithm's properties is presented using an example from transportation engineering.**

I. Introduction

Reinforcement learning (RL) has become a widely accepted sample-based tool for solving hard decision making problems in unknown and stochastic environments [1]. Numerous RL algorithms applicable to Markov Decision Processes (MDPs) with large state and action spaces are based on function approximation aimed at estimating the value and/or the policy functions using a limited number of parameters [2]–[4]. The so-called *Actor-Critic* (AC) methods have appeared as RL analogs of the dynamic programming policy iterations [5]–[8]. They are characterized by a structure consisting of two main parts: a) the Critic, aimed at the state (or state-action) value function estimation, b) the Actor, aimed at the policy function improvement by using current results given by the Critic [5], [6]. *Multi-task reinforcement learning* (MTRL) problems are oriented towards learning a common policy from multiple tasks [9]–[11]. There are many successful results related to MTRL, based on different inductive bias sharing between tasks, including deep neural networks [12]–[14]. In general, *decentralized and distributed multi-agent* RL algorithms are currently in the focus of researchers due to promising practical results [10], [15]–[20]. Particular attention has been paid to *consensus-based methods* which provide a useful framework for diverse collaboration-based problems but with many theoretical challenges [3], [17], [21], [22].

A. Main Contributions

We propose in this paper a synergy between the methodologies of multi-task learning (MTL) and the multi-agent AC off-policy RL, resulting into a provably convergent distributed MTRL algorithm. Our starting point is the definition of the local criteria connected to the local tasks, in the form of a linear approximation of the local MDP state-value functions. Then we define a global (proxy) criterion to be maximized in a distributed way by the RL policy parameters under specific constraints required by the so-called *hard parameter sharing*, in which a subset of the policy parameters is shared between the tasks (agents), while the remaining parameters are task-specific [23]. We derive our AC algorithm for MTRL, generalizing the concept of Maei [8] proposed for the design of single-agent AC algorithms, in such a way that the Critic algorithm is chosen to be the GTD(1) temporal-difference algorithm working at a fast time scale, while the Actor algorithm is obtained from the Critic algorithm as a *gradient-type recursive procedure* utilizing exactly the proposed global (proxy) criterion and working at a slow time scale. Connecting the general MTL problem to the structure of the AC RL leads to nice convergence properties of the entire MTRL algorithm. The policy parameters sharing is achieved by implementing a *dynamic consensus scheme* [17], [22]s. The adopted strict *information structure constraints* (sISCs) imply, in accordance with the MTL problem formulation, that the states, actions, and rewards of the MDPs are accessible only by the locally attached agents [10], [24]–[28]. The sISCs are essentially different from those adopted in typical multi-agent reinforcement learning (MARL) algorithms, since they assume accessibility of all the states, actions, and rewards by all the agents.

Besides the main concept, our contributions encompass several theoretical and practical aspects of the proposed algorithm. The fundamental definition and property of the Actor algorithm is given by Theorem 1. An analysis of the trace

M. S. Stanković is with Universidade Lusófona, Lisboa, Portugal; and Singidunum University, Belgrade, Serbia (e-mail: milos.stankovic@ulusofona.pt).

M. Beko is with Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal; and Universidade Lusófona, Lisboa, Portugal (e-mail: beko.marko@ulusofona.pt).

S. S. Stanković is with School of Electrical Engineering, University of Belgrade, Serbia (e-mail: stankovic@etf.rs).

K. H. Johansson is with School of Electrical Engineering and Computer Science, and Digital Futures, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: kallej@kth.se).

This research was supported by the Fundação para a Ciência e a Tecnologia under Grants 2023.08205.CEECIND and 2022.07530.CEECIND.

variables of the Actor algorithm represents a prerequisite of the subsequent convergence analysis (Lemma 2). The weak convergence of the algorithm to an attached mean *ordinary differential equation* (ODE) is proved separately for the fast time scale (Theorem 2) and the slow time scale (Theorem 3) [17], [22], [29]–[34]. The proof of Theorem 3 is based on weak assumptions and provides a firm basis for the applications of the algorithm. The stability analysis of the derived ODE given in Subsection V-D is based on the methodology of *vector Lyapunov functions* and *aggregation models* developed initially for large-scale systems [35]. It is shown that a specific feature vector introduced in the Critic algorithm enables asymptotic stability of the algorithm while keeping the required dynamics of the task-specific parameters (Theorem 4, Corollary 1). By a convergence rate analysis based on a stochastic differential equation (SDE), we demonstrate possible covariance reduction achievable by the algorithm and its utilization as a parallelization and data fusion tool (compare with [36]).

B. Related Work

To our knowledge, the general methodology of the algorithm design proposed in this paper has not been described in the literature. There are, however, numerous related works. The paper [37] can be considered to be complementary to the present paper in the sense of starting from the general AC design methodology proposed in [8]. However, the Critic and Actor algorithms in [37] are quite different, with different convergence properties. Moreover, the structure of the overall algorithm in [37] is simplified and does not envisage the task-specific parameters important for the MTL context; also, the attention has been paid neither to the stability problem, nor to the convergence rate. The distributed AC algorithm proposed in [26] is based on the coordination of multiple agents solving a class of MDPs by using a consensus-like algorithm for updating the policy parameters and assuming the same sISCs as in the present paper; sufficient attention is not paid to the convergence issues. In [10], a distributed multi-agent AC system for MTRL is proposed starting from the duality theory (Diff-DAC), assuming sISCs. Convenient approximations, together with neural networks and the consensus-like (diffusion) algorithms are introduced, but without studying convergence issues. We mention also the recent papers [11], [13], [14], [38]–[41] devoted to different aspects of distributed learning with consensus in the context of MTRL. A significant contribution is presented in [39] based on learning of the whole set of policies by all the agents; in the present paper, we propose a simpler solution based on one common policy. The transfer learning adopted in [11] and the definition of the finite-time convergence rate in [13] are complementary to the line of thought of this paper. The paper [14] (and references therein) is devoted to the behavior sharing and propose a Q-switch mixture of policies. In [40] the authors treat the problem of learning what to share, while [41] proposes a context-based representation. Valuable general results related to the convergence of RL algorithms are presented in [31], [33], [42], [43]. Fundamentals of the stability analysis by the vector Lyapunov functions and aggregation modeling are presented in [35] and the references therein.

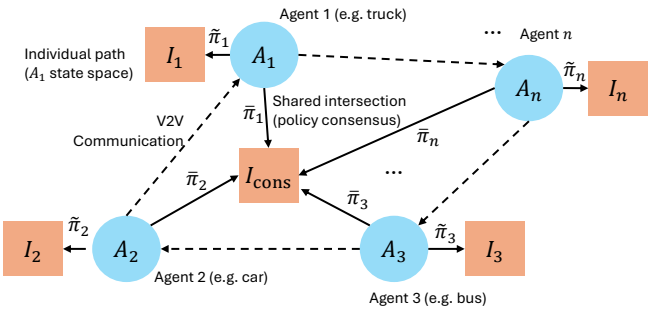

Fig. 1. Illustration of a multi-agent multi-task vehicle routing scenario in an intelligent traffic system. Agents/vehicles A_1,...,A_n navigate a shared intersection I_{cons}, requiring policy consensus via V2V communication (dashed lines). Individual intersections (I_1,...,I_n) allow independent policies $\tilde{\pi}_1$,...,$\tilde{\pi}_n$.

Preliminary results related to the here proposed scheme and its weak convergence in single-task and multi-task cases (with all parameters sharing) have been presented in [34], [44].

C. Illustrative Example

We provide a motivating example of our MTRL approach inspired by autonomous vehicle routing [45]–[47]. We consider the scenario illustrated in Fig. 1 where multiple vehicles (agents) operate within a traffic network represented as a MDP. The task of each agent is to estimate (on the basis of their local environment models) the optimal policy for navigation through intersections, defined by actions such as turning left, right, or proceeding straight. The models encapsulate the state spaces (e.g., current intersection, traffic signals, nearby vehicles) and reward functions (e.g., travel time, fuel consumption). Due to variations in the vehicle types (e.g., cars vs. trucks) or road conditions, the agents' local models may differ.

Agents operate in a decentralized and distributed manner, communicating over a network to share the locally computed optimal policy estimates. The goal is to reach a consensus on the optimal actions for shared segments of the environment (such as major intersections or the frequently used arterial roads). Conversely, for parts of the environment used exclusively by the individual agents, such as residential streets or peripheral routes, any consensus is unnecessary, allowing the agents to retain individualized policies tailored to their unique conditions.

A simplified MDP model of the described traffic problem is studied and evaluated in Section VII.

D. Paper Organization

The paper is organized as follows. Section II contains the problem formulation and the definition of the performance criteria for the MTRL problem. Section III is devoted to the formulation of the Critic algorithm and the derivation of the policy gradient utilized for the definition of the Actor algorithm. Section IV deals with the weak convergence of the Critic, while in Section V Actor weak convergence is provided together with a stability analysis. Section VI is devoted to the rate of convergence of the algorithm. In Section VII simulation results are given. Concluding remarks together with some suggestions for future work are provided in Section VIII.

II. Problem Formulation

A. General Setting

Consider a general *RL setting* in which N *learning agents* acquire data from their *local environments* having MDP properties. Each MDP$^{(i)}$ (attached to agent i), $i = 1, \ldots, N$, is characterized by the finite sets of states $\mathcal{S}$ and actions $\mathcal{A}$, the probability $P^i(\hat{s}^i|s^i, a^i)$ (to move to state $\hat{s}^i \in \mathcal{S}$ from state $s^i \in \mathcal{S}$ by applying action $a^i \in \mathcal{A}$) and the random reward $R^i(s^i, a^i, \hat{s}^i)$, with the probability distribution $p^i(\cdot|\hat{s}^i, a^i, s^i)$ characterized by the expectation $r^i(\hat{s}^i, a^i, s^i)$ [1]. We define the *local target policy* as $\pi^i(s^i, a^i) : \mathcal{S} \times \mathcal{A} \to [0, 1]$, specified by a corresponding probability distribution $p^i(\cdot|s^i)$. In the *on-policy setting*, agent i in state s_t^i at discrete time t executes action $a_t^i \sim p^i(\cdot|s_t^i)$; as a consequence, MDP$^{(i)}$ transitions to a new state s_{t+1}^i and produces a random reward R_{t+1}^i [1]. The *local value function* of state s^i in MDP$^{(i)}$ is defined under the target policy π^i as

$$V^{\pi^i,i}(s^i) = E_{\pi^i}\left\{\sum_{j=0}^{\infty}\prod_{k=1}^{j}(\gamma_{t+k}^i)^j R_{t+1+j}^i|s_t^i = s^i\right\}, \quad (1)$$

where $\gamma_t^i = \gamma^i(s_t^i) \in (0, 1]$ is the local *discount factor*, depending, in general, on i and s_t^i, while $E_{\pi^i}\{\cdot\}$ denotes the expectation over data generated by the Markov chain induced by π^i in MDP$^{(i)}$. In this paper, we assume the *off-policy* scenario, i.e., each MDP$^{(i)}$ generates data using a local *stationary behavior policy* $\pi_b^i(s^i, a^i) : \mathcal{S} \times \mathcal{A} \to [0, 1]$ [48].

We assume that the agents exchange data over a network. The inter-agent communications are formally represented by a *digraph* $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ is the set of *nodes* attached to the agents and $\mathcal{E}$ the set of directed edges. Let $A_{\mathcal{G}} = [A_{\mathcal{G}}^{ij}]$ be the *adjacency matrix* and $\mathcal{N}_i \subset \mathcal{N}$ the *in-neighborhood* of node i [17], [22]. In addition, we assume sISCs, under which agent i, $i = 1, \ldots, N$, has direct access for each t only to the action, state, and reward from the local MDP$^{(i)}$ [37].

We assume function approximation, so that for each MDP$^{(i)}$ the Critic generates an *approximation of the value function* $V^{\pi^i,i}(s^i)$ in the form of $V_{\theta^i}^i(s) = \theta^{iT}\varphi^i(s)$, where θ^i is a local *parameter vector* and $\varphi^i(s) \in \mathcal{R}^{p_i}$ a *feature vector*, $p_i \ll |\mathcal{S}|$ [1], [2]. The local target policies π^i are also parameterized: $\pi^i(s^i, a^i) = \pi^i(s^i, a^i; w^i) = \pi_{w^i}^i(s^i, a^i)$, where $w^i \in \mathcal{R}^{q_i}$ is the local *policy parameter vector*, $q_i \ll |\mathcal{S} \times \mathcal{A}|$.

We adopt the following basic assumptions [37]:

(A1) a) matrix $I - P^{\pi^i,i}\Gamma^i$ is nonsingular, where $P^{\pi^i,i}$ is the matrix of local state transition probabilities under π^i, and Γ^i is a diagonal matrix with $\gamma^i(s^i)$ at the main diagonal, $s^i = s_1, \ldots, s_{|\mathcal{S}|}$; b) $P^{\pi^i,i}$ is irreducible and $\forall s^i, \hat{s}^i \in \mathcal{S}: \; P_{s^i,\hat{s}^i}^{\pi_b^i,i} = 0 \Rightarrow P_{s^i,\hat{s}^i}^{\pi^i,i} = 0$ [1], [31].

(A2) Let $V_{\theta^i}^i = \Phi^i\theta^i$, where $V_{\theta^i}^i = [V_{\theta^i}(s_1^i) \cdots V_{\theta^i}(s_{|\mathcal{S}|}^i)]^T$ and $\Phi^i \in R^{|\mathcal{S}|\times p_i}$ is a feature matrix with the vector $\varphi^{iT}(s)$ as its s-th row. Then: a) the column vectors of Φ^i are *linearly independent*, b) all the vectors $\varphi^i(s)$ are almost surely (a.s.) bounded and have a *unit value* of 1 as their p_i-th element [8].

Remark 1: Assumptions (A1) and (A2a) are standard; assumption (A2b) is specific, related to the way we are defining policy gradients (see Subsection III-C and Section V below, as well as [8] for more details).

B. Multi-Task Learning: Performance Criteria

The idea of MTRL is to learn a *common policy* for multiple tasks, so that it generalizes well across all of them, e.g. [9], [10], [49]. A common methodology is based on the design of a *parameterized hypothesis class* that shares some policy parameters between tasks, found by optimizing an additional *proxy objective* [23]. In such a way, we shall distinguish below two subsets of the policy parameters: a) those explicitly shared between the tasks b) those that remain locally task-specific. Such a diversification of the character of parameters adds a flexibility to the whole MTRL system, but implies theoretical challenges in the analysis of the convergence and stability. Most of the existing MTRL methods are based on the assumption that all the local data are available to all the tasks, which is not realistic in many applications.

We propose a completely distributed AC off-policy algorithm for MTRL, under the presented sISCs. We assume that

$$J^i(w^i) = \sum_{s^i \in \mathcal{S}} p_b^i(s^i)V_{\theta^i}^i(s^i) = \theta^i(w^i)^T E_i\{\varphi_t^i\}, \quad (2)$$

where $E_i\{\cdot\}$ denotes the expectation w.r.t. to $p_b^i(s)$, the stationary distribution induced in MDP$^{(i)}$ by π_b^i, while $\varphi_t^i = \varphi^i(s_t^i)$, $i = 1, \ldots, N$. Notice that (2) is a linear function of θ^i, but, in general, a nonlinear function of w^i. According to the adopted strategy of parameter sharing, we introduce $w^i = [\bar{w}^{iT}\vdots\tilde{w}^{iT}]^T$, where $\bar{w}^i = \bar{w}$ is the shared parameter vector (equal for all the agents) and $\tilde{w}^i$ are the local *task-specific* vectors ($i = 1, \ldots, N$). We shall also adopt w.l.o.g. the simplifying assumption that $\dim(\bar{w}^i) = \bar{q}$, $\dim(\tilde{w}^i) = \tilde{q}$, $q^i = q = \bar{q} + \tilde{q}$, in order to avoid too many indices (in general, we may have $\bar{q} = const$, but $q^i = \bar{q} + \tilde{q}^i$).

At the level of the entire network, we introduce the following *global (proxy) criterion*

$$J(W) = \sum_{i=1}^{N} c^i J^i(w^i), \quad (3)$$

where $W^T = [\bar{W}^T \; \tilde{W}^T]^T$, $\bar{W} = [\bar{w}^{1T} \cdots \bar{w}^{NT}]^T$, $\tilde{W} = [\tilde{w}^{1T} \cdots \tilde{w}^{NT}]^T$, and $c^i \in \mathcal{R}^+$ are *a priori* chosen weights. Consequently, our MTRL problem can be formulated as the following *optimization problem*:

$$\max_W J(W) \quad \text{subject to}: \quad \bar{w}^1 = \cdots = \bar{w}^N, \quad (4)$$

The optimal parameter vector $W^* = \arg\max_W J(W)$ is given by $W^* = [\bar{W}^{*T}\vdots\tilde{W}^{*T}]^T$, $\bar{W}^* = [\bar{w}^{*T} \cdots \bar{w}^{*T}]^T$, $\tilde{W}^* = [\tilde{w}^{1*T} \cdots \tilde{w}^{N*T}]^T$.

III. Algorithm

In this section, we derive the proposed overall two-time-scale distributed AC MTRL algorithm, whose pseudo-code is represented below as Algorithm 1.

A. Critic

In principle, any temporal difference algorithm estimating the value function approximation parameters θ_t^i from (2) in real time on the basis of current MDP observations could be

used as a local Critic algorithm, e.g. [1], [4], [17], [31], [37], [48]. In this paper, we adopt GTD(1) (GTD(λ)$|_{\lambda=1}$) as the local Critic algorithm, bearing in mind its salient properties presented in [8], [44] (see also Subsection V-B). The *statistical form* of GTD(1) given by

$$\lim_{t\to\infty} E_i\{\rho_t^i\delta_t^i e_t^i\} = 0 \tag{5}$$

generates the following stochastic approximation iterates

$$\theta_{t+1}^i = \theta_t^i + \alpha_t^i\rho_t^i\delta_t^i e_t^i, \tag{6}$$

where $\rho_t^i = \pi_{w^i}^i(a_t^i|s_t^i)/\pi_b^i(a_t^i|s_t^i)$ is the *importance ratio*,

$$\delta_t^i = R_{t+1}^i + \gamma_{t+1}^i\theta_t^{iT}\varphi_{t+1}^i - \theta_t^{iT}\varphi_t^i \tag{7}$$

the *temporal difference*,

$$e_t^i = \varphi_t^i + \gamma_t^i\rho_{t-1}^i e_{t-1}^i, \quad e_{-1}^i = 0, \tag{8}$$

the *eligibility trace vector* and $\alpha_t^i > 0$ the *step size* (see e.g. [8], [30], [50]). For the special reasons to be exposed below, we shall introduce a possibility to have $\varphi^i = [\bar{\varphi}^{iT}\vdots\tilde{\varphi}^{iT}]^T$, $\theta^i = [\bar{\theta}^{iT}\vdots\tilde{\theta}^{iT}]^T$ and $e^i = [\bar{e}^{iT}\vdots\tilde{e}^{iT}]^T$ (with w.l.o.g. $\dim\bar{\varphi}^i = \dim\bar{\theta}^i = \dim\bar{e}^i = \bar{p}$, $\dim\tilde{\varphi}^i = \dim\tilde{\theta}^i = \dim\tilde{e}^i = \tilde{p}$), so that $\theta^{iT}\varphi^i = \bar{\theta}^{iT}\bar{\varphi}^i + \tilde{\theta}^{iT}\tilde{\varphi}^i$ can be explicitly incorporated in the definition δ_t^i in (7) (if necessary, see Subsection V-D). In general, the iterates (6) for θ_t^i can be decomposed into the following iterates for $\bar{\theta}_t^i$ and $\tilde{\theta}_t^i$:

$$\bar{\theta}_{t+1}^i = \bar{\theta}_t^i + \alpha_t^i\rho_t^i\delta_t^i\bar{e}_t^i; \quad \tilde{\theta}_{t+1}^i = \tilde{\theta}_t^i + \alpha_t^i\rho_t^i\delta_t^i\tilde{e}_t^i. \tag{9}$$

Notice the following two important facts related to the GTD(1) algorithm:

a) GTD(1) generates asymptotically the optimal mean-squared-error (MSE) solution [4], and

b) the following linear equation for the asymptotically optimal parameter vector $\theta^{i*}(w^i)$ is obtained from (5) for any given w^i

$$b^i + C^i\theta^{i*}(w^i) = 0, \tag{10}$$

where $b^i = \lim_t E\{R_{t+1}^i e_t^i\}$ and $C^i = \lim_t E\{e_t^i\psi_t^{iT}\}$, with $\psi_t^i = \gamma_{t+1}^i\varphi_{t+1}^i - \varphi_t^i$ ($E\{\cdot\}$ denotes the expectation w.r.t. the target policy) [1], [8], [31].

B. Local Policy Gradients

The local policy gradients

$$\nabla_{w^i} J^i(w^i) = \nabla_{w^i}\theta^i(w^i)^T E_i\{\varphi_t^i\} \tag{11}$$

are elaborated in this subsection, extending the results of [8] to the multi-agent multi-task problems, as formulated in Section II-B.

For GTD(1) as the Critic algorithm, we start from (5) and obtain that $\lim_{t\to\infty}\nabla_{w^i}E_i\{\rho_t^i\delta_t^i e_t^i\} = 0$, wherefrom it follows that

$$\begin{aligned}\nabla_{w^i}\theta^{iT}(w^i) = [E_i\{\rho_t^i\delta_t^i\nabla_{w^i}\log\pi_{w^i}^i(s_t^i,a_t^i)\} \\ + E_i\{\rho_t^i\delta_t^i\nabla_{w^i}e_t^{iT}\}][E_i\{\rho_t^i(\varphi_t^i - \gamma_{t+1}^i\varphi_{t+1}^i)e_t^{iT}\}]^{-1},\end{aligned} \tag{12}$$

where $\nabla_{w^i}\log\pi_{w^i}^i(s_t^i,a_t^i)\} = [\nabla_{\bar{w}^i}\log\pi_{w^i}^i(s_t^i,a_t^i)\}^T\vdots\nabla_{\tilde{w}^i}\log\pi_{w^i}^i(s_t^i,a_t^i)\}^T]^T$ and $\nabla_{w^i}e_t^{iT} = [(\nabla_{\bar{w}^i}e_t^{iT})^T\vdots(\nabla_{\tilde{w}^i}e_t^{iT})^T]^T$. Therefore, the local policy gradient derived from (4) is given by

$$\nabla_{w^i}J^i = E_i\{\rho_t^i\delta_t^i[f_t^i\nabla_{w^i}\log\pi_{w^i}^i(s_t^i,a_t^i)] + \nabla_{w^i}e_t^{iT}\eta^i]\}, \tag{13}$$

where $\eta^i = (H^{iT})^{-1}E_i\{\varphi_t^i\}$, $H^i = E_i\{\rho_t^i(\varphi_t^i - \gamma^i\varphi_{t+1}^i)e_t^{iT}\}$ and $f_t^i = e_t^{iT}\eta^i$, bearing in mind that H^i is nonsingular due to (A2a).

We also have the following important relation utilized in Lemma 1 and Theorem 1 below:

$$E_i\{\rho_t^i(\varphi_t^i - \gamma_{t+1}^i\varphi_{t+1}^i)f^i(s_t^i)\} = E_i\{\varphi_t^i\}, \tag{14}$$

where $f^i(s_t^i) = E\{f_t^i|s_t^i\} = E\{e_t^i|s_t^i\}^T\eta^i$ [8].

C. Actor

In general, implementations of the general expression (13) may be faced with serious problems in the case of off-policy learning [8]. However, starting from the following lemma from [8], we are going to demonstrate that GTD(1) used as the Critic algorithm generates a new Actor algorithm for real-time AC RL using *exact gradients* of the *global criterion* (4).

Lemma 1 ([8]): For GTD(1) as the Critic algorithm, the following holds for the corresponding policy gradient formulated in Subsection III-B: a) The sequence $\{f_t^i\}$, $t \geq 0$, defined by

$$f_t^i = 1 + \gamma_t^i\rho_{t-1}^i f_{t-1}^i, \quad f_{-1}^i = 0, \tag{15}$$

$i = 1,\dots,N$, satisfies (14); b) $\eta^i = [0\ \cdots\ 0\ 1]^T$.

Proof: The proof can be derived following [8]. One should bear in mind that $f^i(s^i)$ is the unique solution to (14) for a given s^i. ■

Using Lemma 1 and the definition of the global criterion (4), we obtain our main result for the Actor algorithm design:

Theorem 1: For GTD(1) as the Critic algorithm, (13) gives rise to the following statistical form of the corresponding distributed policy gradient Actor algorithm solving the multi-agent MTL problem formulated by (4):

$$\begin{aligned}\nabla_{\bar{w}}J(W) &= \sum_{i=1}^N c^i\nabla_{\bar{w}^i}J^i(\bar{w}^i,\tilde{w}^i)|_{\bar{w}^i=\bar{w}} \\ &= \lim_{t\to\infty}\sum_{i=1}^N E_i\{c^i\rho_t^i\delta_t^i\bar{\varepsilon}_t^i\} = 0,\end{aligned} \tag{16}$$

$$\nabla_{\tilde{w}^i}J(W) = \nabla_{\tilde{w}^i}J^i(w^i) = \lim_{t\to\infty}E_i\{\rho_t^i\delta_t^i\tilde{\varepsilon}_t^i\} = 0, \tag{17}$$

$i = 1,\dots N$, where, for $t \geq 0$,

$$\bar{\varepsilon}_t^i = f_t^i\nabla_{\bar{w}^i}\log\pi_{w^i}^i(s_t^i,a_t^i) + \gamma_t^i\rho_{t-1}^i\bar{\varepsilon}_{t-1}^i, \ \ \bar{\varepsilon}_{-1}^i = 0, \tag{18}$$

$$\tilde{\varepsilon}_t^i = f_t^i\nabla_{\tilde{w}^i}\log\pi_{w^i}^i(s_t^i,a_t^i) + \gamma_t^i\rho_{t-1}^i\tilde{\varepsilon}_{t-1}^i, \ \ \tilde{\varepsilon}_{-1}^i = 0. \tag{19}$$

Proof: The first term at the right hand side of (13) can be written as $\sum_{s^i}p_b^i(s^i)E\{f_t^i|s_t^i = s^i\}$ $E\{\rho_t^i\delta_t^i\nabla_{w^i}\log\pi_{w^i}^i(s_t^i,a_t^i)\ |s_t^i = s^i\}$, since conditioning on $s_t^i = s^i$ makes f_t^i independent of s_{t+1}^i, as it depends on s_{t-1}^i and the past [1], [4], [8]. According to Lemma 1, part a), this term becomes equal to $E_i\{f_t^i\rho_t^i\delta_t^i\nabla_{w^i}\log\pi_{w^i}^i(s_t^i,a_t^i)\}$, where f_t^i is generated by (15). Applying Lemma 1, part b), we conclude that the second term at the right hand

side of (13) satisfies $\nabla_{w^i} e_t^{iT}\eta^i = \nabla_{w^i} f_t^i$. Defining $\varepsilon_t^i = f_t^i \nabla_{w^i} \log \pi_{w^i}^i(s_t^i, a_t^i) + \nabla_{w^i} f_t^i$, validity of (18) and (19) follows after appropriate changes of variables, taking into account that $\varepsilon^i = [\bar{\varepsilon}^{iT} \vdots \tilde{\varepsilon}^{iT}]^T$, $\dim(\bar{\varepsilon}^i) = \bar{q}$, $\dim(\tilde{\varepsilon}^i) = \tilde{q}$. ■

The statistical forms (16) and (17) generate directly the following distributed AC (DAC) MTRL algorithm consisting of two interconnected iterates of stochastic approximation type:

$$\hat{w}_t^i = \bar{w}_t^i + \beta_t^i c^i \rho_t^i \delta_t^i \bar{\varepsilon}_t^i, \quad \bar{w}_{t+1}^i = \textstyle\sum_{j\in\mathcal{N}_i} a_t^{ij} \hat{w}_t^j, \tag{20}$$

$$\tilde{w}_{t+1}^i = \tilde{w}_t^i + \beta_t^i \rho_t^i \delta_t^i \tilde{\varepsilon}_t^i \tag{21}$$

$\forall t \geq 0$, $i = 1, \ldots, N$, where β_t^i is a step-size sequence satisfying $\beta_t^i \ll \alpha_t^i$, $\forall t \geq 0$, in order to ensure the *two-time-scale property* of the whole AC algorithm, while $\bar{\varepsilon}_t^i$ and $\tilde{\varepsilon}_t^i$ are defined by (18) and (19). Matrix $A_t = [a_t^{ij}]$, $i, j = 1, \ldots, N$, $a_t^{ij} \geq 0$, is a row-stochastic matrix with random elements attached to the graph $\mathcal{G}$ (the so-called consensus matrix) representing a part of a dynamic consensus algorithm [22]. Notice that the consensus from (20) is applied only to $\bar{w}^i$, the part of w^i which is shared by all the agents. Formally, (20) is composed of two main parts: 1) update of $\bar{w}_t^i$ using the currently observed local trajectory tuples, and 2) convexification of the estimates obtained from the node neighborhoods, aimed at achieving convergence to consensus, providing $\bar{w}^{1*} = \ldots = \bar{w}^{N*} = \bar{w}^*$. The iterates (21) generating the task-specific parts $\tilde{w}^i$, $i = 1, \ldots, N$, are composed of N interconnected gradient-based recursions of stochastic approximation type. Obviously, (20) and (21) are interrelated, influencing the stability properties of the whole algorithm in a nontrivial way (see Section V below).

The proposed DAC MTRL algorithm is represented by a pseudo code as the Algorithm 1 below.

Algorithm 1 DAC MTRL Algorithm

for All the nodes $i \in \mathcal{N}$ **do**
  Initialize $\theta_0^i = \theta^{0i}$, $w_0^i = [\bar{w}_0^{iT} \vdots \tilde{w}_0^{iT}]^T = w^{0i}$
  loop
    In each time step t:
    Observe state s_t^i
    Execute $a_t^i \sim \pi_b^i(a_t^i | s_t^i)$
    Observe reward R_{t+1}^i
    Critic:
    Compute ρ_t^i, δ_t^i and e_t^i using (7) and (8)
    Compute θ_{t+1}^i by (6) (with fast time-scale step size α_t^i)
    Actor:
    Compute $\bar{\varepsilon}_t^i$ and $\tilde{\varepsilon}_t^i$ using (18) and (19)
    Compute $\tilde{w}_t^i$ by (21) (with slow time-scale step β_t^i)
    Compute $\hat{w}_t^i$ by (20) (with slow time-scale step β_t^i)
    Broadcast $\hat{w}_t^i$ to out-neighbors
    Receive $\hat{w}_t^j$ from in-neighbors $j \in \mathcal{N}_i$
    Compute $\bar{w}_t^i$ using (20)
    Until convergence
  end loop
end for

D. Global Algorithm Model

For the Critic algorithm, we introduce functions $g^i(\theta^i, w^i, \xi_g^i) = \rho^i(s^i, a^i)\bar{\delta}^i(\theta^i, s^i, a^i, \hat{s}^i)e^i$, where $\xi_g^i = (s^i, a^i, \hat{s}^i, e^i)$, $\bar{\delta}^i(\theta^i, s^i, a^i, \hat{s}^i) = r^i(s^i, a^i, \hat{s}^i) + \gamma^i(\hat{s}^i)\varphi^{iT}(\hat{s}^i)\theta^i - \varphi^{iT}(s^i)\theta^i$ and $r^i(s^i, a^i, \hat{s}^i)$ is the one-step expected reward (while following π_b^i, $i = 1, \ldots, N$). Notice that $\delta_t^i = \bar{\delta}^i(\theta_t^i, s_t^i, a_t^i, s_{t+1}^i) + \omega_{t+1}^i$, where ω_{t+1}^i is a zero mean random sequence, modeling randomness in R_{t+1}^i. Also, we have $g_t^i = g^i(\theta_t^i, w_t^i, \xi_{g,t}^i)$.

For the Actor algorithm, we define $h_{[1]}^i(\theta^i, w^i, \bar{\xi}_h^i) = c^i\rho^i(s^i, a^i)\bar{\delta}^i(\theta^i, s^i, a^i, \hat{s}^i)\bar{\varepsilon}^i$ and $\bar{\xi}_h^i = (s^i, a^i, \hat{s}^i, \bar{\varepsilon}^i, \hat{f}^i)$, while $h_{[2]}^i(\theta^i, w^i, \tilde{\xi}_h^i) = \rho^i(s^i, a^i)\bar{\delta}^i(\theta^i, s^i, a^i, \hat{s}^i)\tilde{\varepsilon}^i$, where $\tilde{\xi}_h^i = (s^i, a^i, \hat{s}^i, \tilde{\varepsilon}^i, \hat{f}^i)$; also, $h_{[1],t}^i = h_{[1]}^i(\theta_t^i, w_t^i, \bar{\xi}_t^i)$ and $h_{[2],t}^i = h_{[2]}^i(\theta_t^i, w_t^i, \tilde{\xi}_t^i)$.

Therefore, the proposed algorithm can be represented, at the network level, by the following vector-matrix equations:

$$\Theta_{t+1} = \Theta_t + \alpha_t[G(\Theta_t, W_t, \Xi_{g,t}) + \Omega_{g,t}] \tag{22}$$

$$W_{t+1} = \text{blockdiag}\{(A_t \otimes I_{\bar{q}}), I_{N\tilde{q}}\} \times \{W_t + \beta_t[H(\Theta_t, W_t, \hat{\Xi}_{h,t}) + \Omega_{h,t}]\} \tag{23}$$

where we assumed that $\alpha_t^i = \alpha_t$ and $\beta_t^i = \beta_t$, for all $i = 1, ..., N$, and where $\Theta_t = [\theta_t^{1T} \cdots \theta_t^{NT}]^T$, $W_t = [\bar{W}_t^T \vdots \tilde{W}_t^T]^T$, $\bar{W}_t = [\bar{w}_t^{1T} \cdots \bar{w}_t^{NT}]^T$, $\tilde{W}_t = [\tilde{w}_t^{1T} \cdots \tilde{w}_t^{NT}]^T$, $G(\Theta_t, W_t, \Xi_{g,t}) = [g^1(\cdot,\cdot,\cdot)^T \cdots g^N(\cdot,\cdot,\cdot)^T]^T$, $H(\Theta_t, W_t, \Xi_{h,t}) = [H_{[1]}(\Theta_t, W_t, \bar{\Xi}_{h,t})^T \vdots H_{[2]}(\Theta_t, W_t, \tilde{\Xi}_{h,t})^T]^T$, $H_{[1]}(\Theta_t, W_t, \bar{\Xi}_{h,t}) = [h_{[1]}^1(\cdot,\cdot,\cdot)^T \cdots h_{[1]}^N(\cdot,\cdot,\cdot)^T]^T$, $H_{[2]}(\Theta_t, W_t, \tilde{\Xi}_{h,t}) = [h_{[2]}^1(\cdot,\cdot,\cdot)^T \cdots h_{[2]}^N(\cdot,\cdot,\cdot)^T]^T$, $\Xi_{g,t} = [\xi_{g,t}^{1T} \cdots \xi_{g,t}^{NT}]^T$, $\Xi_{h,t} = [\bar{\Xi}_{h,t}^T \vdots \tilde{\Xi}_{h,t}^T]^T$, $\bar{\Xi}_{h,t} = [\bar{\xi}_{h,t}^{1T} \cdots \bar{\xi}_{h,t}^{NT}]^T$, $\tilde{\Xi}_{ht} = [\tilde{\xi}_{h,t}^{1T} \cdots \tilde{\xi}_{h,t}^{NT}]^T$, $\Omega_{g,t} = [\rho_t^1\omega_{t+1}^1 e_t^{1T} \cdots \rho_t^N\omega_{t+1}^N e_t^{NT}]^T$, $\Omega_{h,t} = [\bar{\Omega}_{h,t}^T \vdots \tilde{\Omega}_{h,t}^T]^T$, $\bar{\Omega}_{h,t} = [\rho_t^1\omega_{t+1}^1\bar{\varepsilon}_t^{1T} \cdots \rho_t^N\omega_{t+1}^N\bar{\varepsilon}_t^{NT}]^T$, $\tilde{\Omega}_{h,t} = [\rho_t^1\omega_{t+1}^1\tilde{\varepsilon}_t^{1T} \cdots \rho_t^N\omega_{t+1}^N\tilde{\varepsilon}_t^{NT}]^T$.

IV. Convergence Analysis: Critic, Fast Time-Scale

Convergence of the GTD(λ) algorithms ($\lambda \in [0, 1]$) has been thoroughly analyzed in the literature, e.g. [4], [31]. We shall recapitulate here a theorem dealing with the *weak convergence* of the GTD(1) algorithm, assuming that $W_t = W$.

(A3) Matrix C^i is nonsingular, $i = 1, \ldots, N$.

(A4a) Θ_t is tight [30].

Theorem 2 ([31], [33]): Let (A1), (A2), (A3) and (A4a) hold. Let $\{\Theta_t\}$ be generated by (22) with $\alpha_t = \alpha > 0$ and $W_t = W$, where W is an arbitrary constant vector. Let $\{t_\alpha\}$ be a sequence of nonnegative integers such that $\alpha t_\alpha \to \infty$ as $\alpha \to 0$. Then, there exists a sequence T_α with $T_\alpha \to \infty$ as $\alpha \to 0$, such that for any $\delta^i > 0$

$$\lim_{\alpha\to 0} \sup P\{\theta_t^i \notin \mathcal{N}_{\delta^i}(\theta^{i*}), \text{some } t \in [t_\alpha, t_\alpha + T_\alpha/\alpha]\} = 0, \tag{24}$$

where $\mathcal{N}_{\delta^i}(\cdot)$ denotes the δ^i-neighborhood of an indicated set and $\theta^{i*} = (C^i)^{-1}b^i$ by (10), so that $\Theta^* = [\theta^{1*T} \cdots \theta^{N*T}]^T$.

V. Convergence Analysis: Actor, Slow Time-Scale

Within this subsection we shall focus on: 1) general assumptions related to the consensus scheme incorporated in the algorithm, 2) an analysis of the Feller-Markov properties of the *trace variables* characterizing the Actor algorithm, as prerequisites for the convergence proof, $i = 1, \dots, N$, 3) *weak convergence proof* of the Actor algorithm to a limit ODE at the slow time-scale, with Θ_t replaced by $\Theta^*(W_t)$, according to Theorem 2, 4) a stability analysis of the derived limit ODE based on the vector Lyapunov functions and the aggregation models.

Similarly with (A4a), we assume:

(A4b) W_t is tight.

Remark 2: Assumptions (A4a) and (A4b) are unnecessary when some kind of projection or truncation is incorporated in the AC algorithms [29], [30].

A. Consensus Part

The general network setting and the related assumptions are presented very shortly; details can be found in [17], [22], [29].

Let $Y(t|k) = A_t \cdots A_k$ for $t \geq k$, $Y(t|t+1) = I_N$, and let $\tilde{\mathcal{F}}_t$ be an increasing sequence of σ-algebras such that $\tilde{\mathcal{F}}_t$ measures $\{W_k, k \leq t, A_k, k < t\}$.

(A5) Graph $\mathcal{G}$ is strongly connected.

(A6) There is a scalar $\varsigma_0 > 0$ such that $a_t^{ii} \geq \varsigma_0$, and, for $i \neq j$, either $a_t^{ij} = 0$ or $a_t^{ij} \geq \varsigma_0$.

(A7) There are a scalar $p_0 > 0$ and an integer t_0 such that $P_{\tilde{\mathcal{F}}_t}\{\text{agent } j \text{ communicates to agent } i \text{ on the interval } [t, t+t_0]\} \geq p_0$, for all t and i, j for which $A_{\mathcal{G}}^{ij} \neq 0$.

(A8) Let $Y_k = \lim_{t\to\infty} Y(t|k)$ exists w.p.1. Then, there is an $N \times N$ matrix $\bar{Y}$ such that $E\{|E_{\tilde{\mathcal{F}}_k}\{Y_k\} - \bar{Y}|\} \to 0$ as $|t-k| \to \infty$, which has the form $\bar{Y} = [\hat{Y} \cdots \hat{Y}]^T$, where $\hat{Y} = [\bar{y}^1 \cdots \bar{y}^N]^T$; also, $Y_t = \begin{bmatrix} y_t^1 & \cdots & y_t^N \\ & \cdots & \\ y_t^1 & \cdots & y_t^N \end{bmatrix}$ [17], [22], [29].

(A9) Sequence $\{A_t\}$ is independent of the processes in MDP$^{(i)}$, $i = 1, \dots, N$.

Remark 3: Assumptions (A5)–(A9), formulated according to [29], are, essentially, very mild. They allow, for example, considering different types of asynchronous broadcast gossip algorithms, random communication dropouts, etc. [22].

B. Actor Trace Variables

In this subsection, we shall be concerned with several important properties of the Actor *trace variables*, defined as $\{\hat{Z}_t^i\} = \{s_t^i, a_t^i, \varepsilon_t^i, \hat{f}_t^i\}$, $i = 1, \dots, N$. We shall include the following assumptions:

(A10) For every $(s^i, a^i) \in \mathcal{S} \times \mathcal{A}$ the mappings $w^i \longmapsto \pi_{w^i}^i$ are twice differentiable, $i = 1, \dots, N$.

(A11) $\sup_{w^i, s^i, a^i} \|\nabla_{w^i} \log \pi_{w^i}^i(s^i, a^i)\| < \infty$ and $\nabla_{w^i} \log \pi_{w^i}^i(s^i, a^i)$ has a bounded derivative $\forall (s^i, a^i) \in \mathcal{S} \times \mathcal{A}$.

The following lemma deals with the properties of $\{\hat{Z}_t^i\}$ related closely to the properties presented under (i) and (ii) in [32, Subsection 3.1, Appendix A] and [33]. The derivation of the proof of the lemma is, however, new and different from the related proofs in [32].

Lemma 2: Let $\chi_t^i = [\bar{\varepsilon}_t^{iT} \hat{f}_t^i]^T$. Then, under (A1)–(A11) the following holds for any bounded χ_{-1}^i: a) $\sup_{t\geq 0} E_i\{\|\chi_t^i\|\} < \infty$, and b) the zero-input response of χ_t^i tends to zero a.s.

Proof: Recursions (18), (19) and (15) give rise to the following discrete-time dynamic model:

$$\chi_t^i = \gamma_t^i \rho_{t-1}^i \hat{S}_t^i \chi_{t-1}^i + \hat{T}_t^i, \tag{25}$$

where $\hat{S}_t^i = \left[\begin{array}{c:c} I & \nabla_{\bar{w}^i} \log \pi_{w^i}^i(s_t^i, a_t^i) \\ \hdashline 0 & 1 \end{array}\right]$, $\hat{T}_t^i = \left[\begin{array}{c} \nabla_{\bar{w}^i} \log \pi_{w^i}^i(s_t^i, a_t^i) \\ \hdashline 1 \end{array}\right]$, wherefrom

$$\chi_t^i = \prod_{k=1}^t \gamma_k^i \rho_{k-1}^i \hat{S}_k^i \chi_{-1}^i + \sum_{k=1}^t \prod_{j=k}^{t-1} \gamma_{j+1}^i \rho_j^i \hat{S}_{j+1}^i \hat{T}_k^i. \tag{26}$$

a) It is straightforward to derive (like in [37]) that $E_i\{\prod_{j=k}^{t-1} \|\hat{S}_j^i\|\} \leq L_1^i (t-k)$, $0 < L_1^i < \infty$. Also, using [32, Lemmas A.1 and A.2, Proposition A.1], we find out that for $|t-k|$ large enough $E_i\{E\{\prod_{j=k}^{t-1} \gamma_{j+1}^i \rho_j^i | \mathcal{F}_k\}\} \leq L_2^i (\sigma^i)^{t-k}$, where $\mathcal{F}_k$ is a σ-algebra generated by χ_t^i in (25) up to time k and $\hat{T}_t^i$ up to time $k-1$, $0 < L_2^i < \infty$, with $|\sigma^i| < 1$. Therefore,

$$E_i\{\|\chi_t^i\|\} \leq L_3 t (\sigma^i)^t + L_4 \sum_{k=1}^t (t-k)(\sigma^i)^{t-k} < \infty \tag{27}$$

($0 < L_3, L_4 < \infty$).

b) The form of (25) implies that $\|\sum_j \hat{S}_j^i\| \leq L_5 t$, $0 < L_5 < \infty$. Therefore, we define for $t \geq 0$ a nonnegative sequence $\{\Delta_t^i\}$ by

$$\Delta_t^i = t \prod_{j=1}^t \gamma_j^i \rho_{j-1}^i = \frac{t}{t-\mu} \prod_{j=t-\mu+1}^t (\gamma_j^i \rho_{j-1}^i) \Delta_{t-\mu}^i, \tag{28}$$

(with a.s. bounded initial condition), where $\mu \in \mathcal{I}^+$. It is evident that $\exists \mu$ such that $E\{\prod_{j=t-\mu}^t (\gamma_j^i \rho_{j-1}^i) | \mathcal{F}_{t-\mu}\} < 1$ (a.s.). Then, $\exists t_0 > 0$ such that $\frac{t}{t-\mu} E\{\prod_{j=t-\mu}^t (\gamma_j^i \rho_{j-1}^i) | \mathcal{F}_{t-\mu}\} \leq 1$ (a.s.), $\forall t > t_0$. Therefore, the convergence theorem for nonnegative supermartingales can be applied, i.e. $\Delta_t^i \to \Delta_\infty^i$ (a.s.). As $E_i\{\Delta_t^i\} \to_{t\to\infty} 0$, $\Delta_\infty^i = 0$ a.s. Hence, the result follows. ∎

The Actor trace properties analogous to the properties (iii) and (iv) from [32, Subsection 3.1] can be demonstrated using Lemma 2 and [32, Theorem 3.2].

Based on [32] and the above derived results, it follows that: a) $\{\hat{Z}_t^i\}$ is a Feller-Markov chain bounded in probability and having a unique probability measure $\hat{\zeta}^i$, and b) for each $\hat{Z}_0^i$ the sequence $\frac{1}{t}\sum_{k=0}^{t-1} f(\hat{Z}_k^i)$ converges in mean and a.s. to $E_{\hat{\zeta}^i}\{f(\hat{Z}_0^i)\}$ for any Lipschitz continuous function f.

Let $\bar{h}_{[1]}^i(\theta^{i*}(w^i), w^i) = E_{\zeta_h^i}\{h_{[1]}^i \ (\theta^{i*}(w^i), w^i, \xi_{h,0}^i)\}$ and $\bar{h}_{[2]}^i(\theta^{i*}(w^i), w^i) = E_{\zeta_h^i}\{h_{[2]}^i \ (\theta^{i*}(w^i), w^i, \xi_{h,0}^i)\}$, where ζ_h^i is the probability measure corresponding to ξ_h^i. We can also write $\bar{h}_{[1]}^i = \bar{b}^i + D_{11}^i \bar{\theta}^{i*}(w^i) + D_{12} \tilde{\theta}^{i*}(w^i)$ and $\bar{h}_{[2]}^i = \tilde{b}^i + D_{21}^i \bar{\theta}^{i*}(w^i) + D_{22} \tilde{\theta}^{i*}(w^i)$, where $\bar{b}^i = E\{R_{t+1}^i \bar{\varepsilon}_t^i\}$, $\tilde{b}^i = E\{R_{t+1}^i \tilde{\varepsilon}_t^i\}$, $D_{11}^i = E\{\bar{\varepsilon}_t^i \bar{\psi}_t^{iT}\}$, $D_{12}^i = E\{\bar{\varepsilon}_t^i \tilde{\psi}_t^{iT}\}$, $D_{21}^i = E\{\tilde{\varepsilon}_t^i \bar{\psi}_t^{iT}\}$ and $D_{22} = E\{\tilde{\varepsilon}_t^i \tilde{\psi}_t^{iT}\}$ (the expectations are obtained using

ζ_h^i). Also, let $\bar{H}_{[1]}(\Theta^*(W),W) = [\bar{h}_{[1]}^1(\theta^{1*}(w^1),w^1)^T \cdots \bar{h}_{[1]}^N (\theta^{N*}(w^N),w^N)^T]^T$ and $\bar{H}_{[2]}(\Theta^*(W),W) = [\bar{h}_{[2]}^1 (\theta^{1*}(w^1),w^1)^T \cdots \bar{h}_{[2]}^N(\theta^{N*}(w^N),w^N)^T]^T$.

C. Weak Convergence Proof

Assume that $\beta_t = \beta > 0$. Let t_β be a sequence tending to ∞ when $\beta \to 0$, satisfying $\beta^{\frac{1}{2}} t_\beta \to 0$ as $\beta \to 0$ and $\sup_k P\{|Y(k+t_\beta|k) - Y_k| \geq \beta^2\} \leq \beta^2$ (see [29]). Define $\bar{W}_0^\beta = (Y(t_\beta|0) \otimes I_{\bar{q}})\bar{W}_0 + \beta \sum_{k=0}^{t_\beta - 1}(Y_k \otimes I_{\bar{q}})\bar{H}(\Theta^*(W_k),W_k,\bar{\Xi}_k)$. For $\tau \in \mathcal{R}^+$, let $W^\beta(\tau) = W_t$ for $\tau \in [(t-t_\beta)\beta,(t-t_\beta+1)\beta)$ ($W^\beta(\cdot)$ starts slightly away from the origin due to the existence of the consensus scheme, see [29], [30] for details).

Theorem 3: Let W_t be generated by (23) in which Θ_t is replaced by $\Theta^*(W_t)$ and let (A1)–(A11) be satisfied. Then, $W^\beta(\tau)$ is tight and converges weakly when $\beta \to 0$ to $W^*(\tau) = [\bar{w}(\tau)^T \cdots \bar{w}(\tau)^T, \tilde{w}^{1T}(\tau) \cdots \tilde{w}^{NT}(\tau)]^T$, $\tau \in \mathcal{R}^+$, where $\bar{w}(\tau)$ and $\tilde{w}^i(\tau)$ satisfy

$$\dot{\bar{w}} = \sum_{i=1}^N c^i \bar{y}^i \bar{h}_{[1]}^i(\theta^{i*}(w^i),w^i), \quad \bar{w}(0) = \bar{w}_0, \tag{29}$$

$$\dot{\tilde{w}}^i = \bar{h}_{[2]}(\theta^{i*}(w^i),w^i), \quad \tilde{w}^i(0) = \tilde{w}_0^i, \tag{30}$$

$i = 1,\ldots,N$, with $\theta^{i*}(w^i)$ given by (10).

Also, there exists a sequence T_β with $T_\beta \to \infty$ as $\beta \to 0$ such that $\forall i \in \{1,\ldots,N\}$ and any $\delta > 0$

$$\lim_{\beta \to 0} \sup P\{W_t \notin \mathcal{N}_\delta(\mathcal{W}^*), \text{some } t \in [t_\beta, t_\beta + T_\beta/\beta]\} = 0, \tag{31}$$

where $\mathcal{W}^*$ is a bounded invariant set of (29), (30).

Proof: At the first step it is essential to verify the basic assumptions from [29], [30, Theorem 3.1]. Using the results from [33, Paragraph 4.1.1] and the Subsection V-B devoted to the Actor trace variables, we readily conclude that the assumptions C(3.2) and C(3.3') from [29, Section 3] are satisfied for the proposed algorithm. More specifically, the results of Subsection V-B imply C(3.2), while Lemma 2, together with [33, Paragraph 4.1.1], can be used in a straightforward way to verify C(3.3').

Iterating both $\bar{W}_t$ and $\tilde{W}_t$ backwards, we obtain

$$\begin{aligned}\bar{W}_{t+1} =& \bar{W}_0^\beta + \beta \sum_{k=t_\beta}^t (Y_k \otimes I_{\bar{q}}) H_{[1]}(\Theta^*(W_k),W_k,\bar{\Xi}_{h,k}) \\ &+ \beta \bar{\varrho}_t^\beta + [(Y(t|0) - Y(t_\beta|0)) \otimes I_{\bar{q}}]\bar{W}_0,\end{aligned} \tag{32}$$

$$\tilde{W}_{t+1} = \tilde{W}_{t_\beta} + \beta \sum_{k=t_\beta}^t H_{[2]}(\Theta^*(W_k),W_k,\tilde{\Xi}_{h,k}), \tag{33}$$

where $\bar{\varrho}_t^\beta = \sum_{k=0}^t \{[Y(t|k) - Y_k] \otimes I_{\bar{q}}\} H_{[1]}(\Theta^*(W_k),\bar{W}_k,\bar{\Xi}_{h,k})$. Notice that the iterations for $\bar{W}_t$ and $\tilde{W}_t$ are interrelated, However, it is possible to find out that the results from [29, Theorem 3.1, Part 1] can be extended to (32) with minor technical modifications implied by the adopted specific parametrization of the target policy. Namely, using (32) and (A8) we obtain that $\sup_{\beta, t \geq t_\beta} \frac{1}{\beta^2} E_i\{\|W_{t+1} - W_t\|^2\} < \infty$ and $\{\frac{1}{\beta}\|W_{t+1} - W_t\|, t \geq t_\beta\}$ is uniformly integrable, implying that $\{W^\beta(\cdot)\}$ is tight and that the limit paths are Lipschitz continuous in τ.

The asymptotic mean ODEs from (29) and (30) are derived extending the methodology from [17], [29], [30]. Namely, following [29], we introduce

$$\begin{aligned}M_f(\tau) =& f(W(\tau)) - f(W(0)) \\ &- \int_0^\tau \nabla_W f(W(s))^T \left[\begin{array}{c}(\bar{Y} \otimes I_{\bar{q}})\bar{H}_{[1]}(\Theta^*(W),W) \\ \hdashline \bar{H}_{[2]}(\Theta^*(W),W)\end{array}\right] ds.\end{aligned} \tag{34}$$

Using [29] we can show that $M_f(\tau)$ is a continuous-time martingale and that $M_f(\tau) = 0$, where $f(\cdot)$ is any real valued function with compact support and continuous second derivatives. Consequently, one obtains on the basis of Theorem 3.1, Part 3 of [29] that

$$\dot{\bar{W}} = (\bar{Y} \otimes I_{\bar{q}})\bar{H}_{[1]}(\Theta^*(W),W), \quad \dot{\tilde{W}} = \bar{H}_{[2]}(\Theta^*(W),W). \tag{35}$$

By (A5)–(A9) all the rows of $\bar{Y}$ are equal, so that it follows from (35) that $\bar{W}(\cdot) = [\bar{w}(\cdot)^T \cdots \bar{w}(\cdot)^T]^T$, with $\bar{w}(\cdot)$ satisfying (29). The set of ODEs in (30) is equivalent to the second relation in (35).

The last part of the theorem asserts that the process spends nearly all of its time in a small neighborhood of an invariant set of the derived system of ODEs (29), (30). It is important to notice that $\nabla_{\bar{w},\tilde{w}^1,\ldots,\tilde{w}^N} J(W) = 0$ for $\bar{y}^i = 1/N$, according to (3). Consequently, the chain recurrent points of an invariant set consist of the stationary points of the derived ODEs. The theorem does not rule out the convergence to unstable or saddle points with some probability. In general, the choice of a convenient Lyapunov function for the stability analysis depends on the form of the target policy; then, the analysis of the fixed points can follow the general results from, e.g. [51]. Notice only that the stochastic nature of the problem implies that the probability is zero that the algorithm converges to possibly unstable stationary points in $\mathcal{W}^*$ [51]. ■

Remark 4: The weights c^i and the coefficients $\bar{y}^i$ appear only within the products $c^i\bar{y}^i$. Practically, there are two main possibilities: a) to adopt $c^i = 1$ and to choose the appropriate values of $\bar{y}^i$ by adequate network design (details concerning the network design can be found in [22]) or b) to adopt $\bar{y}^i = 1/N$ and to retain the freedom of selecting c^i *a priori*. Notice here only that there is always (under unrestrictive conditions) such a matrix $A_t = A$ that provides any desired set of coefficients $\bar{y}^i$, $i = 1,\ldots,N$.

Remark 5: The above analysis is based on the assumption that the step sizes of the algorithm are *constant*. It is technically straightforward to formulate a similar convergence proof for *tapering step sizes* (tending to zero more slowly than $1/t$), by applying the general methodology from [30], [31].

D. Limit ODE: Stability

Stability analysis of the stationary points of the obtained limit ODEs (29), (30) will be done in this subsection starting from the methodology of the vector Lyapunov functions, which offers significant flexibilities in the case of distributed and large-scale systems exposed to structural perturbations,

e.g. [35], [52]–[54]. The main focus will be placed on the interconnections between the estimation of the shared parameter vector, on one side, and of the local parameter vectors, on the other. We shall analyze two characteristic cases, depending on the a priori choice of the feature vector $\tilde{\varphi}_t^i$ in the Critic algorithm.

1) Case (1): Using (35) and assuming $\tilde{\varphi}_t^i = 0$, we obtain the following nonlinear vector-matrix ODE

$$\dot{W} = \begin{bmatrix} \dot{\bar{W}} \\ \dot{\tilde{W}} \end{bmatrix} = \begin{bmatrix} \bar{b} \\ \tilde{b} \end{bmatrix} + \begin{bmatrix} \check{A} \\ 0 \end{bmatrix} \bar{\Theta}(W) + \begin{bmatrix} 0 \\ \check{B} \end{bmatrix} \bar{\Theta}(W) = F_{(1)}^w(W) + K_{(1)}^w(W), \quad (36)$$

where $F_{(1)}^w(W) = \begin{bmatrix} \bar{b} \\ \tilde{b} \end{bmatrix} + \begin{bmatrix} \check{A} \\ 0 \end{bmatrix} \bar{\Theta}$, $\check{A} = \begin{bmatrix} D_{11}^1 & \cdots & D_{11}^N \\ \vdots & & \vdots \\ D_{11}^1 & \cdots & D_{11}^N \end{bmatrix} \in \mathcal{R}^{N\bar{q} \times N\bar{p}}$, $K_{(1)}^w(W) = \begin{bmatrix} 0 \\ \check{B} \end{bmatrix} \bar{\Theta}$ and $\check{B} = \mathrm{diag}\{D_{21}^1, \ldots, D_{21}^N\} \in \mathcal{R}^{N\tilde{q} \times N\bar{p}}$; the vector function $\bar{\Theta}(W) = [\bar{\theta}^{1T} \cdots \bar{\theta}^{NT}]^T$ results from the Critic algorithm.

In order to analyze stability of the stationary points of (36), we define $W^{*,(1)} = \mathrm{Arg}_W\{F_{(1)}^w(W) = 0\}$. After introducing $X = W - W^{*,(1)}$, (36) gives $\dot{X} = F_{(1)}^x(X) = F_{(1)}^w(X + W^{*,(1)})$, satisfying $F_{(1)}^x(0) = 0$; similarly, we have $K_{(1)}^x(X) = K_{(1)}^w(X + W^{*,(1)})$, so that (36) is equivalent to $\dot{X} = F_{(1)}^x(X) + K_{(1)}^x(X)$, $X^T = [x^{1T} \cdots x^{NT}]^T$, $x^i = w^i - w^{i*,(1)}$, $x^i = [\bar{x}^T \vdots \tilde{x}^{iT}]^T$.

Theorem 4: Let a function $v^{(1)} : \mathcal{R}_\rho^{N\bar{q}} \to \mathcal{R}_+$ be continuously differentiable w.r.t. X, where $\mathcal{R}_\rho^{N\bar{q}} = \{X \in R^{N\bar{q}} : \|X\| < \rho\}$, and let $v^{(1)}(0) = 0$,

$$\eta_1^{(1)}\|X\| \le v^{(1)}(X) \le \eta_2^{(1)}\|X\|$$
$$\dot{v}^{(1)}(X)_{[F]} \le -\eta_3^{(1)}\|X\|,$$
$$\|\mathrm{grad}_X[v^{(1)}(X)]\| \le \eta_4^{(1)}, \quad (37)$$

where $\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}$ and $\eta_4^{(1)}$ are positive numbers and the derivative $v^{(1)}(X)_{[F]}$ is obtained along the trajectories of $\dot{X} = F_{(1)}^x(X)$. Suppose that the interconnection function $K_{(1)}^x(X)$ satisfies the following inequality

$$\mathrm{grad}_X[v^{(1)}(X)]^T K_{(1)}^x(X) \le \nu^{(1)}\|X\|, \quad (38)$$

where $\nu^{(1)} \ge 0$ is bounded (see [35, Definitions 2.17, 2.18], [55]). Let the aggregation matrix $M^{H,(1)}$ satisfy the following Hicks stability condition for the Metzler matrices [55]–[57]

$$M^{H,(1)} = -(\eta_2^{(1)})^{-1}\eta_3^{(1)} + \nu^{(1)}\eta_4^{(1)}(\eta_1^{(1)})^{-1} > 0. \quad (39)$$

Then the equilibrium $X^* = 0$ is exponentially stable.

Remark 6: The theoretical background of the vector Lyapunov functions and the aggregate models applied to large-scale systems can be found in [35]. If s is the dimension of a selected vector Lyapunov function, $s \times s$ aggregation matrices appear as parts of aggregate system models [35, Theorems 2.7, 2.11]. Under appropriate conditions, the *quasi-dominant block-diagonal property* of an aggregation matrix implies stability of the entire system (in our case, $s = 1$, so that the Hicks conditions reduces to the scalar inequality (39)) [35], [55]–[57].

Proof: We shall follow the general methodology of [35, Theorem 2.16]. The total time derivative $\dot{v}^{(1)}(X)_{[F+K]}$ along the solutions of the equation $\dot{X} = F_{(1)}^x(X) + K_{(1)}^x(X)$ is obtained as

$$\dot{v}^{(1)}(X)_{[F+K]} = \dot{v}^{(1)}(X)_{[F]} + [\mathrm{grad}_X v^{(1)}(X)]^T K_{(1)}^x(X)$$
$$\le -\eta_3^{(1)}\|X\| + \nu^{(1)}\eta_4^{(1)}\|X\| \le -(\eta_2^{(1)})^{-1}\eta_3^{(1)}v^{(1)}(X)$$
$$+\nu^{(1)}\eta_4^{(1)}(\eta_1^{(1)})^{-1}v^{(1)}(X) \le -\alpha^{(1)}v^{(1)}(X), \quad (40)$$

where $\alpha^{(1)} > 0$. From the last inequality we get

$$v^{(1)}(X(\tau)) = v^{(1)}(X_0)\exp[-\alpha^{(1)}(\tau - \tau_0)], \quad (41)$$

for some X_0 and τ_0. The assertion follows after using [35, Theorems 2.15, 2.16]. ∎

The model $\dot{X} = F_{(1)}^x(X)$ is stable at $X^* = 0$ under (37) according to the classical Lyapunov methodology, while Theorem 4 deals with the stability of the entire model (36) under different interconnection functions $K_{(1)}^x(X)$. Analyzing (36) term by term, we immediately find out that matrix $\check{A}$ in $F_{(1)}^x(X)$ is composed of identical block-rows composed of D_{11}^i, $i = 1, \ldots, N$, and that matrix $\check{B}$ in $K_{(1)}^x(X)$ is block-diagonal, having D_{21}^i at the block-diagonal, $i = 1, \ldots, N$. Therefore, $F_{(1)}^x(X)$ gives rise to N identical $\bar{q}$-th order ODEs defined by $\dot{\tilde{x}} = \sum_{i=1}^N c^i \bar{y}^i D_{11}^i \theta^i(x^i)$, while $K_{(1)}^x(X)$ is composed of N $\tilde{q}$-th order ODEs $\dot{\tilde{x}}^i = D_{21}^i \theta^i(x^i)$, $i = 1, \ldots, N$. Analysis of these lower order ODEs can be done using the methodology of Theorem 4.

Elaborating practical implications of Theorem 4, we recall that $D_{21}^i = E\{\tilde{\varepsilon}_t^i \bar{\psi}_t^{iT}\}$ by definition and conclude that a convenient choice of the trace vectors $\tilde{\varepsilon}_t^i$ and the feature vectors in $\bar{\psi}_t^{iT}$ influences directly the term $\nu^{(1)}$ from (38) and can make it as small as necessary, in accordance with (39). This means to select, during the design phase of the algorithm, appropriate feature vectors $\bar{\varphi}_t^i$ and vectors $\nabla_{w^i} \log \pi_{w^i}^i(s_t^i, a_t^i) = [\nabla_{\bar{w}^i} \log \pi_{w^i}^i(s_t^i, a_t^i)^T \vdots \nabla_{\tilde{w}^i} \log \pi_{w^i}^i(s_t^i, a_t^i)^T]^T$ (due to the properties of the linear dynamic models generating the trace variables for both Critic and Actor, presented in [31], [32]). Definition of these vectors and their fine tuning are to be done in parallel with the final choice of the Lyapunov function itself.

Corollary 1: Let the assumptions of Theorem 4 hold. Then, the equilibrium $X^* = 0$ can be made exponentially asymptotically stable by an adequate choice of the feature vectors $\tilde{\varphi}_t^i$ for the Critic, as well as of the vectors $\nabla_{\bar{w}^i}^i \log \pi_{w^i}^i(s_t^i, a_t^i)$ and $\nabla_{\tilde{w}^i} \log \pi_{w^i}^i(s_t^i, a_t^i)$ for the Actor.

Remark 7: A possible drawback of adopting the algorithm structure given in (36) lies in a possibility to have a low convergence rate of the local task-specific parameters at the expense of achieving stability by choosing too low values of the spectral norm of the cross-correlation matrix D_{21}^i.

2) Case (2): Assuming $\tilde{\varphi}_t^i \neq 0$, we obtain the following model

$$\dot{W} = \begin{bmatrix} \dot{\bar{W}} \\ \dot{\tilde{W}} \end{bmatrix} = \begin{bmatrix} \bar{b} \\ \tilde{b} \end{bmatrix} + \begin{bmatrix} \check{A} & 0 \\ 0 & \tilde{D} \end{bmatrix} \Theta(W) + \begin{bmatrix} 0 & \check{C} \\ \tilde{B} & 0 \end{bmatrix} \Theta(W) = F_{(2)}^w(W) + K_{(2)}^w(W), \quad (42)$$

where $F^w_{(2)}(W) = \begin{bmatrix}\bar{b}\\ \tilde{b}\end{bmatrix} + \begin{bmatrix}\check{A} & 0\\ 0 & \check{D}\end{bmatrix}\Theta$, $\check{D} = \text{diag}\{D^1_{22},\cdots,D^N_{22}\} \in \mathcal{R}^{N\tilde{q}\times N\tilde{p}}$, $K^w_{(2)}(W) = \begin{bmatrix}0 & \check{C}\\ \check{B} & 0\end{bmatrix}\Theta$, $\check{C} = \begin{bmatrix}D^1_{12}\ \cdots\ D^N_{12}\\ \vdots\quad\ \vdots\\ D^1_{12}\ \cdots\ D^N_{12}\end{bmatrix} \in \mathcal{R}^{N\bar{q}\times N\tilde{p}}$, $\Theta = \Theta(W) = [\bar{\Theta}^T\vdots\tilde{\Theta}^T]^T$, $\tilde{\Theta} = [\tilde{\theta}^{1T}\dots\tilde{\theta}^{NT}]^T$. Also, we have $F^x_{(2)}(X) = F^w_{(2)}(X + W^{*,(2)})$ and $K^x_{(2)}(X) = K^w_{(2)}(X + W^{*,(2)})$, where $W^{*,(2)} = \text{Arg}_W\{F^w_{(2)}(W) = 0\}$, in accordance with Theorem 4. Notice that the model (42) is obtained by incorporating $\psi^i = [\bar{\psi}^{iT}\vdots\tilde{\psi}^{iT}]^T$ in the definition of the temporal difference in (7).

Stability analysis of (42) can be carried out similarly as in Theorem 4. Moreover, all the analogous assumptions and assertions can be obtained directly from those given in Theorem 4 after replacing all the subscripts and superscripts "(1)" (indicating Case (1)) by the subscripts and superscripts "(2)" (indicating Case (2)).

The starting point is now the introduction of a new Lyapunov function candidate $v^{(2)} : \mathcal{R}^{N\bar{p}}_\rho \to \mathcal{R}_+$. Therefore, the new stability condition replacing (39) is now in the form

$$M^{H,(2)} = -(\eta^{(2)}_2)^{-1}\eta^{(2)}_3 + \nu^{(2)}\eta^{(2)}_4(\eta^{(2)}_1)^{-1} > 0, \qquad (43)$$

where $\nu^{(2)}$ follows formally from (38) after inserting the appropriate subscripts and superscripts.

The essential difference between the Cases (1) and (2) does not lie in the theoretical formalities, but in the interpretation of the adopted model structure and the related practical consequences. According to (42), we immediately find out that the matrix $\check{B}$ is composed of D^i_{12} and the matrix $\check{C}$ of D^i_{21} as their block-sub-matrices, respectively, $i = 1,\dots,N$. According to the above given definitions, $D^i_{12} = E\{\tilde{\varepsilon}^i_t\tilde{\psi}^{iT}_t\}$ and $D^i_{21} = E\{\tilde{\varepsilon}^i_t\bar{\psi}^{iT}_t\}$, indicating that the choice of the feature and trace vectors $\bar{\varphi}^{iT}_t$, $\tilde{\varphi}^{iT}_t$, $\bar{\varepsilon}^i_t$ and $\tilde{\varepsilon}^i_t$, can directly make $\nu^{(2)}$ from (43) arbitrarily smaller, leading to the asymptotic stability. In other words, by decreasing cross-correlation between $\bar{\varepsilon}^i_t$ and $\tilde{\psi}^{iT}_t$, as well as between $\tilde{\varepsilon}^i_t$ and $\bar{\psi}^{iT}_t$, we can ensure stability of the whole system, preserving, at the same time, their satisfactory dynamics. Naturally, this conclusion holds under the general assumption that the cross-correlation between $\bar{\varepsilon}^i_t$ and $\bar{\psi}^{iT}_t$, as well between $\tilde{\varepsilon}^i_t$ and $\tilde{\psi}^{iT}_t$, is kept above a certain predefined level. Procedurally, this means to select appropriately the feature vectors $\varphi^i_t = [\bar{\varphi}^{iT}_t\vdots\tilde{\varphi}^{iT}_t]^T$ and the vectors $\nabla_{w^i}\log\pi^i_{w^i}(s^i_t,a^i_t) = [\nabla_{\bar{w}^i}\log\pi^i_{w^i}(s^i_t,a^i_t)^T\vdots\nabla_{\tilde{w}^i}\log\pi^i_{w^i}(s^i_t,a^i_t)^T]^T$.

In conclusion, in the Case (2) the introduction of a specific feature vector $\tilde{\varphi}^i_t$ with the required correlation properties enables obtaining stability of the algorithm without sacrificing quality of the $\tilde{w}^i$ estimation, overcoming in such a way the main drawback of the Case (1).

Remark 8: The state-action value-function parametrization in the Actor based on the Gibbs distribution can be taken as an illustration of the above line of thought [1], [2], [5]. Consider a policy function that is a Gibbs distribution in a linear combination of features represented by

$$\pi^i_{w^i}(s^i,a^i) = \frac{\exp\{w^{iT}\phi^i(s^i,a^i)/\tau_g\}}{\sum_{b^i}\exp\{w^{iT}\phi^i(s^i,b^i)/\tau_g\}} \qquad (44)$$

where $\phi^i(s^i,a^i)$ is a q^i-dimensional feature vector characterizing the state-action pairs (s^i,a^i) in MDP^i and τ_g the temperature parameter. Thus, we have

$$\nabla_{w^i}\log\pi^i_{w^i}(s^i,a^i) = \phi^i(s^i,a^i) - \sum_{b^i}\pi^i(s^i,b^i)\phi^i(s^i,b^i). \qquad (45)$$

After adopting $\phi^i(s^i,a^i) = [\bar{\phi}^i(s^i,a^i)^T\vdots\tilde{\phi}^i(s^i,a^i)^T]^T$, it is possible to implement directly either Case (1) or Case (2).

Remark 9: The extension of Theorems 2 and 3 to the entire two-time-scale algorithm originally defined in (22) and (23) can be done by applying the methodology from [37] (proposed in [31] for GTD(λ) algorithms). Let $W^{\alpha,\beta}_t$ be generated by the *two-time-scale algorithm* (22), (23) with $\alpha_t = \alpha$ and $\beta_t = \beta$, satisfying $\beta \ll \alpha$, and let assumptions (A1)–A(11) hold. Then, $W^{\alpha,\beta}(\tau)$, which is obtained from $W^{\alpha,\beta}_t$ in the same way as $W^\beta(\tau)$ is obtained from W^β_t in Theorem 3, converges weakly to $W(\tau) = [\bar{w}(\tau)^T\cdots\bar{w}(\tau)^T\vdots\tilde{w}^1(\tau)^T\cdots\tilde{w}^N(\tau)^T]^T$, $\tau\in\mathcal{R}^+$, where $\bar{w}(\tau)$ and $\tilde{w}^i(\tau)$ satisfy (29) and (30). The proof of this statement can be derived following the results from [37].

VI. Convergence Rate; Covariance Reduction

Besides providing a solution to the MTRL problem, the proposed distributed AC algorithm can be also considered as an *efficient, low variance parallelization* tool. The structure of the whole algorithm is in this case similar to the structure of the popular A3C and A2C algorithms [36], but in our case based of *completely decentralized computations*. We shall provide an insight into this situation by analyzing the *asymptotic convergence rate* of the Actor algorithm, according to the methodology from [29, Section 6].

Let (A1)–(A11) hold and let $\tilde{w}^i = 0$, so that $w^i = \bar{w}^i = \bar{w}$, $i = 1,\dots,N$. According to Theorem 3, W_t (or $\bar{W}_t$) weakly converges to some $W^* = \bar{W}^* = [\bar{w}^{*T}\cdots\bar{w}^{*T}]^T$. Let

$$U^\beta_t = (W_t - W^*)/\sqrt{\beta}, \qquad (46)$$
$$\Sigma^\beta_t = \sqrt{\beta}\textstyle\sum^t_{T+t_\beta+1}(Y_{t+1}\otimes I_{\bar{q}})\bar{H}_{[1]}(\Theta^*(\bar{W}),\bar{W},\bar{\Xi}_{h,t}),$$

$t > t_\beta + T$, $T > 0$. It can be proved using [29] that the processes $U^\beta(\tau)$ and $\Sigma^\beta(\tau)$, $\tau\in\mathcal{R}^+$, where $U^\beta(\tau) = U^\beta_t$ and $\Sigma^\beta(\tau) = \Sigma^\beta_t$ for $\tau\in[\beta(t-T-t_\beta),\beta(t-T-t_\beta+1))$, weakly converge (under mild assumptions) to $U(\cdot) = [u(\cdot)^T\cdots u(\cdot)^T]^T$ and $\Sigma(\cdot) = [\sigma(\cdot)^T\cdots\sigma(\cdot)^T]^T$, respectively, where $u(\cdot)$ satisfies the following Itô stochastic differential equation (SDE)

$$du = Mud\tau + d\sigma, \qquad (47)$$

where $M = [\sum^N_{i=1}c^i\bar{y}^i\bar{h}^i_{[1]}(\theta^{i*}(\bar{w}),\bar{w})]'_{\bar{w}}$ is the Jacobian matrix, while $\sigma(\cdot)$ is a $\bar{q}$-dimensional Wiener process, such that $\text{cov}\{\sigma\}(1) = \bar{R}$, where

$$\bar{R} = \textstyle\sum^\infty_{t=-\infty}E\{[\sum^N_{i=1}c^iy^i_{t+1}\bar{h}^i_{[1]}(\theta^{i*}(\bar{w}),\bar{w},\bar{\xi}^i_{h,t})]$$
$$\times[\textstyle\sum^N_{i=1}c^iy^i_{t+1}\bar{h}^i_{[1]}(\theta^{i*}(\bar{w}),\bar{w},\bar{\xi}^i_{h,t})]^T\}. \qquad (48)$$

Fig. 2. Diagram of the simulated MDP.

The stationary covariance of $u(\cdot)$ is defined by

$$\bar{R}_u = \int_0^\infty e^{M\tau} \bar{R} e^{M^T \tau} d\tau \tag{49}$$

and can be taken as a measure of the *asymptotic rate of convergence* of the algorithm.

Assume that $P^i(\hat{s}^i|s^i,a^i) = P(\hat{s}^i|s^i,a^i)$, $p^i(\cdot|\hat{s}^i,s^i,a^i) = p(\cdot|\hat{s}^i,s^i,a^i)$ and $r^i(\hat{s}^i,s^i,a^i) = r(\hat{s}^i,s^i,a^i)$ and that $\{\bar{\xi}^i_{h,t}\}$ are mutually independent and $\text{cov}\{\bar{h}^i_{[1]}(\theta^{i*}(\bar{w}),\bar{w},\xi^i_h)\} = R^i$. When, moreover, $\bar{h}^i_{[1]}(\cdot,\cdot,\cdot) = \bar{h}_{[1]}(\cdot,\cdot,\cdot)$ and $R^i = R$, we obtain the SDE from (47) with $\sigma = \sigma^d$ where

$$\text{cov}\{\sigma^d\}(1) = R\sum_{i=1}^N E\{y_t^{i2}\}. \tag{50}$$

In order to compare the proposed RL algorithm with a standard single-agent alternative, consider a single-agent algorithm defined by the iterates

$$z^i_{t+1} = z^i_t + \beta h_{[1]}(\theta^{i*}(z^i_t), z^i_t, \xi^i_{h,t}), \tag{51}$$

where $z^i(\cdot) \in \mathcal{R}^{\bar{q}}$, $i = 1,\dots,N$. Introducing $u^{c,i}_t = (z^i_t - \bar{w}^*)/\sqrt{\beta}$, we obtain

$$du^{c,i} = Mu^{c,i}d\tau + d\sigma^c, \tag{52}$$

where σ^c is a Wiener process with $\text{cov}\{\sigma^c\}(1) = R$. As $\sum_{i=1}^N E\{y_t^{i2}\} < 1$ in (50), our *distributed multi-agent algorithm* yields a potentially significant improvement in the sense of covariance reduction w.r.t. the *standard single-agent algorithms* due to the averaging over the network. The infimum of $\sum_{i=1}^N E\{y_t^{i2}\}$ occurs when all $E\{y_t^{i2}\}$ are equal for all i, giving the improvement factor of $1/N$. In general, the design of a network providing a desired covariance reduction w.r.t. the single agent case can be done following the methodology from [22].

VII. Simulation Results

The simulated MDP environment is a variant of the Boyan's chain, designed to represent a simplified travel decision-making problem as presented in the introduction, e.g. [4], [17], [18], [44]. The diagram of this specific MDP is depicted in Fig. 2.

It is assumed that there are 15 states that represent routing points where a decision should be made by each agent on whether to take the action $a^{i,h}$ (remain on the current main route/road) or $a^{i,\text{exit}}$ (exit the current main route/road). State 15 is the goal/absorbing state. The discount factor γ is set to 0.9. Ten agents are aimed at optimizing the (stationary) policy that determines the probability of selecting $a^{i,\text{exit}}$, at a given state s^i, denoted as $\pi^i_{w^i}, (s^i, a^{i,\text{exit}})$. If $a^{i,\text{exit}}$ is chosen, the probability of encountering a traffic jam $p^{i,\text{exit}}_{\text{stuck}}$ and the corresponding reward $R^i(s^i,a^{i,\text{exit}},\hat{s}^i)$ (for all s^i and $\hat{s}^i$) may be different for each agent i, $i = 1,...,10$, depending on the experimental setup as described below. On the other hand, if action $a^{i,h}$ is chosen, the reward is $R(s^i,a^{ih},\hat{s}^i) = -1$ for any s^i and $\hat{s}^i$, and the probability of getting stuck in a traffic jam increases as $1 - \frac{1}{s^i}$, where s^i represents the local state. A sparse time-invariant communication graph is used where each agent communicates with 2-3 randomly selected agents. The Critic employs the linear function approximation based on a Gaussian radial basis model with 7 features $\varphi_j(s^i) = \exp\{-\frac{(s^i-z^j)^2}{2\sigma^2}\}$, $j = 1,...,7$, where $z^j \in \{1,3,5,7,9,11,13\}$ and σ^2 is set to 2. In the case of the Actor, the policy parametrization defined by $\pi^i_{w^i}(s^i,a^i)$ is implemented using the Gibbs distribution given by (44). Since the chain has an absorbing state, the algorithms are executed over multiple episodes.

In the initial experiment, we demonstrate the algorithm's effectiveness in the multi-task case, where the agents share only a subset of the Actor parameters trying to reach consensus w.r.t. them. The probabilities $p^{i,\text{exit}}_{\text{stuck}}$ and rewards $R^i(s^i,a^{i,\text{exit}},\hat{s}^i)$ are different for each agent (different fuel consumption, road conditions, driving abilities) and set to the following values for different agents: $p^{i,\text{exit}}_{\text{stuck}} \in \{0.9,0.8,0.7,0.6,0.65,\ 0.85,0.75,0.65,0.8,0.9\}$, $R^i(s^i,a^{i,\text{exit}},\hat{s}^i) \in \{-2.5,-3.5,-1.5,-5.5,-4,\ -6,-8,-1,-2.5,-3\}$. The agents' behavior policies are also different (off-policy configuration), so that $\pi^i_b(s^i,a^{i,\text{exit}}) \in \{0.15,0.24,0.13,0.38,0.55,0.89,0.64,0.97,0.75,0.69\}$. In this experiment, tabular policy features are utilized with $q^i = 15 \times 2$, ensuring that no "information" is lost and allowing the local algorithms to converge to the optimal policy; also, $\tau_g = 1/20$. Figs. 3 and 4 show the true optimal value function and the value function of the average MDP corresponding to the final policies estimated by each agent, for the case when the agents share the parameters corresponding to middle states and final states, respectively. The effectiveness of the algorithm in reaching the optimal consensus w.r.t. the shared parameters is obvious.

In the second experiment we assume a single-task setup, adequate for a comparison and demonstration of the behavior (time-evolution) and the benefits of the proposed algorithm. The agents share all the Actor parameters and we set $p^{i,\text{exit}}_{\text{stuck}}$ to 0.2 and rewards $R^i(s^i,a^{i,\text{exit}},\hat{s}^i)$ to 2.5 for each agent, $i = 1,...,10$ (single task setup). We compare our algorithm with a) centralized cases (all-to-all communication) and with b) the algorithm proposed in [37]. In order to convincingly demonstrate the multi-agent benefit, we also assume that the individual agents are unable to find the optimal policy individually due to the imposed restrictions on their behavior. Each agent is allowed to visit only specific subsets of states, as defined by their starting and stopping state pairs: $[(7,13),(8,14),(10,15),(1,8),(1,10),(3,14),(8,13),(7,15),(6,11),(10,15)]$. The behavior policies of the agents and the policy parametrization are the same as in the previous experiment. Figure 5 displays the evolution of the exact value function, calculated precisely at each time step corresponding to the agents' policy estimates

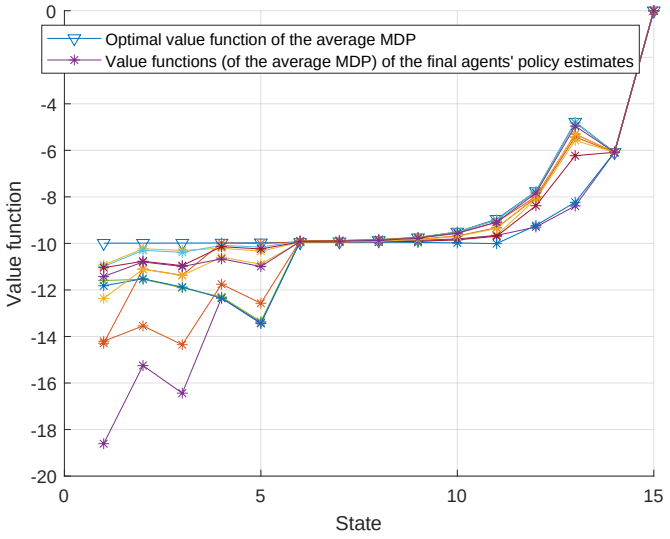

Fig. 3. Experiment 1. The value functions corresponding to the final optimal policy estimates obtained by the agents, together with the true optimal value function, when the agents share the parameters corresponding to the middle states.

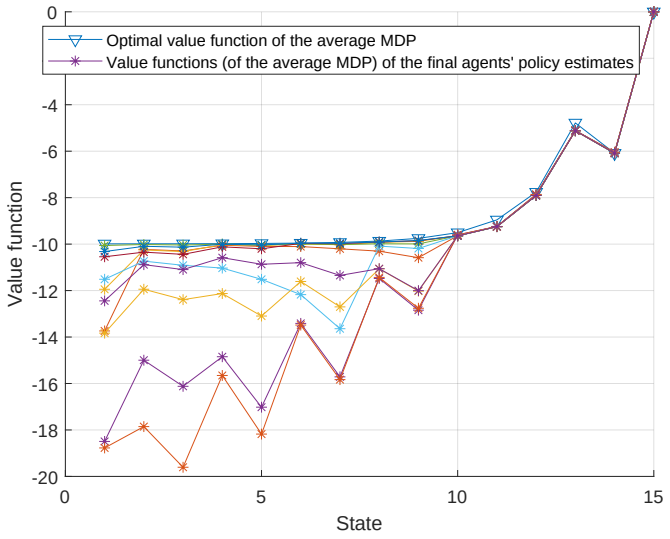

Fig. 4. Experiment 1. The value functions corresponding to the final optimal policy estimates obtained by the agents, together with the true optimal value function, when the agents share the parameters corresponding to the final states.

and averaged over all the agents and all the states. Step sizes $\alpha = 0.02$ and $\beta = 0.0002$ are used. The red horizontal line represents the optimal value function. It can be observed that despite the individual restrictions on the state visiting and the approximations used by the Critic, the agents provide collective convergence close to the optimal policy. The rate of convergence of our scheme is close to the centralized case, and better than in [37].

VIII. Conclusion

In this paper we have proposed a distributed multi-agent off-policy MTRL algorithm based on the AC learning methodology. Concisely, the paper contains the following main contributions:

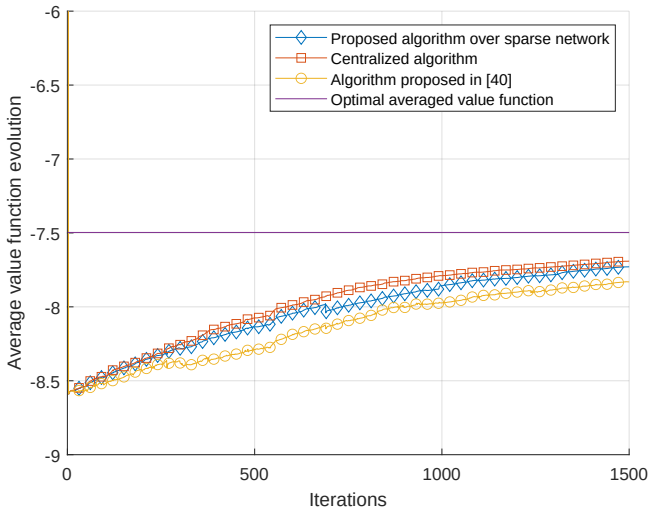

Fig. 5. Experiment 2. Comparison of the evolutions of the averaged (over all the agents and states) value functions for three specified algorithms. The horizontal line is the optimal value.

- proposal of a novel distributed multi-agent AC algorithm for MTRL derived as a gradient scheme from the adopted local and global criterion functions, with a specific policy parametrization including the shared parameters, in parallel with the task-specific local ones,
- weak convergence proof of the proposed two-time-scale algorithm to the set of the limit points of the derived asymptotic mean ODE,
- proof of the stability of the limit ODE using the vector Lyapunov functions and the aggregation modeling,
- derivation of the Feller-Markov properties for the state-trace variables characterizing the novel Actor algorithm,
- analysis of the asymptotic convergence rate using SDEs, including the covariance reduction,
- simulation-based illustration of the applicability of the proposed algorithm.

Immediate continuation of the work can be oriented towards diverse forms of criteria and approximating functions, including deep learning based on consensus in the Actor. The application of the developed methodology to partially observed MDPs represents a special theoretical and practical challenge.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, Cambridge MA, 2017.
[2] H. R. Maei, C. Szepesvari, S. Bhatnagar, and R. S. Sutton, "Toward off policy learning control with function approximation," in *Proc. Intern. Conf. Machine Learning*, 2010, pp. 719–726.
[3] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
[4] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proc. 26th Int. Conf. on Machine Learning*, 2009, pp. 993–1000.
[5] S. Bhatnagar, R. S. Sutton, R. Ghavamzadeh, and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, pp. 2471–2482, 2009.
[6] V. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, pp. 1143–14 166, 2003.
[7] T. Degris, M. White, and R. S. Sutton, "Off policy actor critic," in *Proc. Int. Conf. Machine Learning*, 2012, pp. 179–186.

[8] H. R. Maei, "Convergent actor-critic algorithms under off-policy training and function approximation," *arXiv:1802.07842*, 2018.

[9] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: a survey," *J. of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

[10] S. Valcarcel Macua, A. Tukiainen, D. Garcia-Ocana Hernandez, D. Baldazo, E. Munoz de Cote, and S. Zazo, "Diff-DAC: Distributed actor-critic for average multitask deep reinforcement learning," *arXiv 1710.10363*, 2019.

[11] Y. W. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, "Distral: Robust multitask reinforcement learning," *arXiv:1707.04175*, 2017.

[12] Y. Zhang. and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, pp. 30–43, 2017.

[13] M. A. A. S. Zeng, T. Doan, A. Raychowdhury, and J. Romberg, "A decentralized policy gradient approach to multi-task reinforcement learning," in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, PMLR*, 2021, pp. 1002–1012.

[14] G. Zhang, A. Jain, I. Hwang, S.-H. Sun, and J. Lim, "QMP: Q-switch mixture of policies for multi-task behavior sharing," *arXiv:2302.00671v3*, 2025.

[15] T. Chen, K. Zhang, G. B. Giannakis, and T. Başar, "Communication-efficient policy gradient methods for distributed reinforcement learning," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 2, pp. 917–929, 2022.

[16] D. Lee, H. Yoon, and N. Hovakimyan, "Primal-dual algorithm for distributed reinforcement learning: Distributed GTD," in *IEEE Conf. Decision and Control*, 2018, pp. 1967–1972.

[17] M. S. Stanković, M. Beko, and S. S. Stanković, "Distributed value function approximation for collaborative multiagent reinforcement learning," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 3, pp. 1270–1280, 2021.

[18] M. S. Stanković, M. Beko, N. Ilić, and S. S. Stanković, "Distributed consensus-based multi-agent temporal-difference learning," *Automatica*, vol. 151, p. 110922, 2023.

[19] C. Fiscko, S. Kar, and B. Sinopoli, "Model-free learning and optimal policy design in multiagent MDPs under probabilistic agent dropout," *IEEE Transactions on Control of Network Systems*, vol. 12, no. 1, pp. 361–373, 2025.

[20] T. Sadamoto, A. Kikuya, and A. Chakrabortty, "Distributed reinforcement learning for networked dynamical systems," *IEEE Transactions on Control of Network Systems*, vol. 11, no. 2, pp. 1103–1115, 2024.

[21] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, 2008.

[22] M. S. Stanković, N. Ilić, and S. S. Stanković, "Distributed stochastic approximation: Weak convergence and network design," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4069–4074, 2016.

[23] O. Sener and V. Koltun, "Multi task learning as multi objective optimization," *arXiv: 1810.04650*, 2019.

[24] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," *arXiv:1802.08757*, 2018.

[25] W. Suttle, Z. Yang, K. Zhang, Z. Wang, T. Basar, and J. Liu, "A multi-agent off-policy actor-critic algorithm for distributed reinforcement learning," *arXiv:1908.03963*, 2019.

[26] P. Pennesi and I. Paschalidis, "A distributed actor-critic algorithm and applications to mobile sensor network coordination problems," *IEEE Trans. Autom. Control*, vol. 55, pp. 492–497, 2010.

[27] Y. Zhang and M. M. Zavlanos, "Distributed off-policy actor-critic reinforcement learning with policy consensus," *arXiv:1903.09255*, 2019.

[28] A. OroojlooyJadid and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *arXiv:1908.03963*, 2019.

[29] H. J. Kushner and G. Yin, "Asymptotic properties of distributed and communicating stochastic approximation algorithms," *SIAM J. Control Optim.*, vol. 25, pp. 1266–1290, 1987.

[30] ——, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

[31] H. Yu, "On convergence of some gradient-based temporal-differences algorithms for off-policy learning," *arXiv:1712.09652*, 2017.

[32] ——, "On convergence of emphatic temporal-difference learning," in *Proceedings of The 28th Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 40, 2015, pp. 1724–1751.

[33] ——, "Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize," *Journal of Machine Learning Research*, vol. 17, pp. 1–58, 2016.

[34] M. S. Stanković, M. Beko, N. Ilić, and S. S. Stanković, "Multi-agent actor-critic multitask reinforcement learning based on GTD(1) with consensus," in *IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 4591–4596.

[35] D. D. Šiljak, *Large scale dynamic systems - stability and structure*. North Holland, New York, 1978.

[36] V. Mnih, A. Badia, M. Mirza, A. Graves, T. Harley, T. Lillicrap, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. of 33rd Intern. Conf. on Machine Learning*, 2016.

[37] M. S. Stanković, M. Beko, N. Ilić, and S. S. Stanković, "Multi-agent off-policy actor-critic algorithm for distributed multi-task reinforcement learning," *European Journal of Control*, vol. 74, p. 100853, 2023.

[38] H. Bou-Ammar, E. Eaton, P. Ruvolo, and M. E. Taylor, "Online multi-task learning for policy gradient methods," in *Proc. Intern. Conf. on Machine Learning*, 2014, pp. 1206–1214.

[39] S. El-Bsat, H. Bou-Ammar, and M. E. Taylor, "Scalable multitask policy gradient reinforcement learning," in *Proc. AAAI Conf. on Artificial Intelligence*, 2017, pp. 1847–1853.

[40] X. Sun, R. Panda, R. Feris, and K. Saenko, "Adashare: Learning what to share for efficient deep multi-task learning," *arXiv:1911.12323*, 2020.

[41] S. Sodhani, A. Zhang, and J. Pineau, "Multi-task reinforcement learning with context-based representations," in *Proceedings of the 38-th International Conference on Machine Learning, PMLR*, 2021, p. 139.

[42] H. Yu, "On convergence of emphatic temporal-difference learning," *arXiv:1506.02582v2*, 2015.

[43] H. Yu, A. Mahmood, and R. Sutton, "On generalized Bellman equations and temporal-difference learning," *Journal of Machine Learning Research*, vol. 19, pp. 1–49, 2019.

[44] M. S. Stanković, M. Beko, and S. S. Stanković, "Convergent distributed actor-critic algorithm based on gradient temporal difference," in *Proc. 30th European Signal Processing Conference*, 2022, pp. 2066–2070.

[45] Y. Wu, H. Chen, and F. Zhu, "DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 103, pp. 246–260, 2019.

[46] M. Hua, X. Qi, D. Chen, K. Jiang, Z. E. Liu, H. Sun, Q. Zhou, and H. Xu, "Multi-agent reinforcement learning for connected and automated vehicles control: Recent advancements and future prospects," *IEEE Transactions on Automation Science and Engineering*, 2025.

[47] A. Haydari and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 11–32, 2022.

[48] M. Geist and B. Scherrer, "Off-policy learning with eligibility traces: A survey," *Journal of Machine Learning Research*, vol. 15, pp. 289–333, 2014.

[49] S. Ruder, "An overview of multitask learning in deep neural networks," *arXiv:1706.05098*, 2017.

[50] H. F. Chen, *Stochastic Approximation and its Applications*. Dordrecht, the Netherlands: Kluwer Academic, 2002.

[51] M. B. Nevel'son and R. Z. Has' minskii, *Stochastic approximation and recursive estimation*. American Mathematical Soc., 1976, vol. 47.

[52] R. Bellman, "Vector Lyapunov functions," *SIAM Journal on Control*, vol. 1, pp. 32–34, 1962.

[53] J. P. LaSalle, "Vector Lyapunov functions," *Bull. Inst. Math. Academia Sinica*, vol. 3, pp. 139–150, 1975.

[54] V. Lakshmikantham, "On the method of vector Lyapunov functions," in *Proc. Twelfth Allerton Conf. Circ. Systems*, 1974, pp. 71–74.

[55] L. Grujić and D. D. Siljak, "Asymptotic stability and instability of large-scale systems," *IEEE Trans. Autom. Contr.*, vol. 18, pp. 636–645, 1973.

[56] P. K. Newman, "Some notes on stability conditions," *Review of economic studies*, vol. 72, pp. 1–9, 1959.

[57] F. R. Gantmacher, *The theory of matrices*. Chelsea, New York, 1960.