DECIDE PROJECT SESSION

DISTRIBUTED ON-POLICY ACTOR-CRITIC REINFORCEMENT LEARNING

Miloš S. Stanković[1,2]*,
Marko Beko[3,4],
Miloš Pavlović[2,5],
Ilija Popadić[2],
Srđan S. Stanković[5]

[1]Singidunum University,
Belgrade, Serbia

[2]Vlatacom Institute,
Belgrade, Serbia

[3]Instituto de Telecomunicações,
Instituto Superior Técnico,
Universidade de Lisboa,
Lisbon, Portugal

[4]COPELABS, Universidade Lusófona
de Humanidades e Tecnologias,
Lisbon, Portugal

[5]School of Electrical Engineering,
University of Belgrade,
Belgrade, Serbia

Correspondence:
Miloš S. Stanković

e-mail:
milos.stankovic@singidunum.ac.rs

Abstract:

In this paper, a novel distributed on-policy Actor-Critic algorithm for multi-agent reinforcement learning is proposed. The algorithm consists of the temporal difference scheme with function approximation at the Critic stage, and a policy gradient algorithm at the Actor stage, derived starting from a global objective. At both stages, decentralized agreement among the agents is achieved using the linear dynamic consensus strategy. Compared to the existing schemes, the algorithm has improved convergence rate and noise immunity, and a possibility to achieve multi-task global optimization.

Keywords:

Multi-Agent Systems, Reinforcement Learning, Actor-Critic, Distributed Consensus, Function Approximation.

INTRODUCTION

Under the framework of Markov Decision Processes (MDPs) it is possible to model general decision-making problems in modern complex systems, including Networked Control Systems (NSC), Cyber-Physical Systems (CPS) and Internet of Things (IoT). Reinforcement learning (RL) has been generally accepted as a powerful method for solving MDPs based on online data-based trial-and-error approach, even in the case of very large state and action spaces (see, e.g. [1, 2]). In this case, function approximation represents an important factor, and the problem setup is modified such that the value or policy function is estimated using a limited number of parameters, including the possibility of using (deep) neural network approximators [3, 4, 2]. Three approaches can be, in general, distinguished: a) *value-based methods*, which perform parametrization of the state-value function (in *on-policy* or *off-policy* scenario; see e.g. [5, 6, 4]); b) *policy gradient methods*, which parameterize the policy function typically using a gradient descent algorithm (e.g. [7]); and c) the *Actor-Critic (AC) methods*, which are based on the simultaneous estimation of the parameters of both value function (Critic) and policy function (Actor) [8, 9, 10, 11].

DOI: 10.15308/Sinteza-2022-389-393

In this paper, we deal with multi-agent distributed and decentralized RL methods, which are currently in a strong focus of researchers and practitioners in the modern fields of NSC, CPS and IoT (e.g. [12, 13, 14, 15, 16, 17, 18, 19, 20, 21]). Distributed AC algorithms have been treated in [22, 23, 24, 25, 26, 27, 28] under different settings. In our approach, we assign an independent MDP to each agent and assume the *on-policy setup*, in which in each time step the agents are applying the control policy which is currently estimated as the optimal one. We assume a linear approximation of the state-value function at the Critic stage, and a general nonlinear approximation of the policy function at the Actor stage. The derived Actor stage provides estimates of the policy parameters based on a global objective given in the form of a sum of weighted locally averaged state-value functions, and an exact policy gradient algorithm which we derive from the TD(λ) scheme implemented at the Critic stage. In our multi-agent setup, we propose the agents to collaborate using a linear dynamic consensus scheme aimed at achieving agreement on the policy and value functions between the agents (see, e.g. [29, 19, 16]). The proposed distributed algorithm can be effectively used in multi-task RL problems [24], and as a parallelization tool, significantly improving the rate of convergence, and reducing the overall estimation variance.

The paper is organized as follows. Section 2 contains the problem formulation and the main definitions. In Section 3 we introduce the Critic stage, while in Section 4 we derive the exact policy gradient and the entire AC algorithm in two time-scales. In Section 5 we provide some concluding remarks.

2. PROBLEM FORMULATION

Consider N agents operating in Markov Decision Processes MDP$^{(i)}$, i=1,…,N, defined by the quadruplets $(\mathcal{S},\mathcal{A},\mathcal{P}^i,\mathcal{R}^i)$, where S and A denote finite sets of *states* and *actions*, $\mathcal{P}^i:\mathcal{S}\times\mathcal{S}\times\mathcal{A}\to[0,1]$ is the local *transitional probability* $P^i(s'|s,a)$ of agent i and $\mathcal{R}^i:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathcal{R}$ is the corresponding local real-valued *reward function*, such that the random reward $R^i(s,a,s')$ is characterized by the distribution $p^i(\cdot|s',a,s)$, with the expectation $r^i(s',a,s)$, i=1,…,N.

Communication among the agents is modeled by a *strongly connected digraph* $\mathcal{G}=\{\mathcal{N},\mathcal{E}\}$, where $\mathcal{N}$ is the set of nodes (agents) and $\mathcal{E}$ the set of directed arcs representing inter-node communications. We assume *strict Information Structure Constraints* (ISC), such that node i cannot directly obtain information about the states and actions from MDP$^{(j)}$, $j\neq i$ and such that inter-agent messages can be obtained at node i only from the neighboring nodes [19, 29, 16, 30].

The agents learn from data received by interacting with their *local environments*. In the so-called *on-policy case*, agent i at time t applies an action $a_t^i \sim \pi^i(\cdot|s_t^i)$, where π^i: $\mathcal{S}\times\mathcal{A}^i\to[0,1]$ is a *policy function* (a *conditional probability distribution* on the set of the local state/action pairs). As a consequence, the state of agent i changes to s_{t+1}^i receiving the random reward R_{t+1}^i, i=1,…,N. The *local state value function* at node i, under policy π^i and with the discount factors $\gamma^i\in[0,1]$, is given by Equation 1, where $E_{\pi^i}\{\cdot\}$ denotes the expectation over data generated by the Markov chains induced by π^i, i=1,…,N.

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

Equation 1 – Local state value function.

Introduce the following assumption ensuring that state value functions are well defined:

(A1) $P^{\pi^i,i}=\sum_{a\in\mathcal{A}^i}\pi^i(a|s)P^i(s'|s,a)$ is such that $I-\gamma^i P^{\pi^i,i}$,i is nonsingular, for all π^i, i=1,…,N.

In the context of the *Actor-Critic (AC)* methodology in the single agent case we consider two kinds of *local parametrization*:

a) At the Critic stage, the local value function $V^{\pi^i,i}(s)$ is approximated by $V_{\theta^i}^i(s)=\theta^{i+T}\varphi^i(s)$, where θ^i is the local parameter vector and $\varphi^i(s)\in R^{L_\theta}$ the local *feature vector*, typically satisfying $L_\theta<<M$;

b) At the *Actor stage*, policy π^i is parameterized using the *policy parameter vector* $w^i\in R^{L_w}$, $L_w<<M$, so that $\pi^i=\pi_{w^i}^i$. Agent i is aimed at getting a *locally optimal value* w^{i*} in the sense of a pre-selected criterion using the current estimates θ^i and the local tuples $(s_t^i,a_t^i,R_{t+1}^i,s_{t+1}^i)$.

The *expected linear approximation* of the *local value* function for a given w^i is defined by Equation 2, where $\theta^i=\theta^i(w^i)$ is the local parameter vector.

$$J^i(\theta^i)=\theta^{iT}E_i\{\varphi_t^i\}=\theta^{iT}\sum_s d_b^i(s)\varphi^i(s)$$

Equation 2 – Linear approximation of the local value function.

The locally optimal value is $w^{i*}=Argmax_{w^i}J^i(\theta^i(w^i))$, i=1,…,N.

In the adopted *multi-agent* setting, we are faced with the set of N local criteria $J^i(\theta^i)$ with N possibly different optimal parameter values w^{i*}. We are looking for a solution of a *multi-objective optimization problem* by introducing a convenient *global utility function* denoted as $J(\theta^*(w^1,\ldots,w^N);c)$. This function depends on the global vector $\theta^*(w^1,\ldots,w^N)$, $|\theta^*|=L_\theta$, obtained at the collec-

tive critic stage characterizing the value function of the whole multi-agent system, and on some parameter vector c, $\dim(c)=N$, $0\le c^i\le 1$, $\sum_i c^i=1$ defined *a priori*, giving different importance to the agents. Hence, we introduce the *global criterion* in Equation 3 which enables getting N local optimal policies.

$$J(\theta^*(w^1,...,w^N);\ c)=\theta^{*T}(w^1,...,w^N)\sum_{i=1}^{N}c^i E_i\left\{\varphi_t^i\right\}$$

Equation 3 – Global criterion

However, our goal is to learn a *single policy* that performs optimally for the averaged tasks, so that the goal is to learn a vector w^* characterizing the common policy function $\pi^1_{w^*}=...=\pi^N_{w^*}=\pi_{w^*}$.

3. CRITIC: DISTRIBUTED TD(λ) ALGORITHM

The Critic part of the proposed AC scheme aims at generating recursive estimates θ^i_t, $i=1,\ldots,N$, using local data and communications with the neighboring nodes trying to asymptotically achieve agreement so that $\theta^1=\cdots=\theta^N=\theta^*$. The algorithm consists of two characteristic parts: 1) an update of the local parameter vectors θ^i based on the locally acquired observations, and 2) convexification of the parameter vectors obtained from the neighborhood following a linear consensus scheme. We shall consider in this paper a *distributed version* of the popular temporal difference TD(λ) algorithm, equivalent in the sense of asymptotic behavior under on-policy learning to both the Gradient Temporal Difference GTD(λ) algorithm and the Emphatic Temporal Difference ETD(λ) algorithm [11, 5].

Introducing the bootstrapping parameters λ^i (assumed to be constant, for the sake of simpler notation), we come to the generalized Bellman operators $T(\pi^i,\lambda^i,i)$ $V^i=r^{\pi^i,\lambda^i,i}+P^{\pi^i,\lambda^i,i}V^i$, where $P^{\pi^i,\lambda^i,i}=I-(I-\lambda^i\ P^{\pi^i,i}\Gamma^i)^{-1}(I-P^{\pi^i,i}\Gamma^i)$ and $r^{\pi^i,\lambda^i,i}=(I-\lambda^i\ P^{\pi^i,i}\Gamma^i)^{-1}\ r^{\pi^i,i}$. The gradient TD-algorithms GTD(λ^i) for local linear value function approximation are derived using the following objective function: $J^i_{GTD}(\theta^i)=\frac{1}{2}\left\|\Pi^i\left(T^{(\pi^i,\lambda^i,i)}V^i_{\theta^i}-V^i_{\theta^i}\right)\right\|^2_{d^i_b}$. where Π^i is the projection operator onto the approximation space $\mathcal{L}_{\Phi^i}$ w.r.t. the weighted Euclidian norm $\|\cdot\|_{d^i_b}$) [6].

The value function approximation can formally be expressed as $V^{\theta^i,i}=\Phi^i\,\theta^i$, where $\Phi^i\in R^{M\times L_\theta}$ is a feature matrix with its s-th row equal to the corresponding vector $\varphi^{iT}(s)$. We also adopt the following assumption:

(A2) a) the column vectors of Φ^i are linearly independent;

b) the feature vectors $\varphi^i(s)$ are bounded and with number 1 as their L_θ-th element [11].

The locally optimal parameter vectors θ^{i*} are solutions w.r.t θ^i of equation $E\left\{\delta^i_t e^i_t\right\}=0$,where $\delta^i_t=R^i_{t+1}+\gamma^i\theta^{iT}_t\varphi^i_{t+1}-\theta^{iT}_t\varphi^i_t$ represents the *temporal difference* and $e^i_t=\varphi^i_t+\gamma^i\lambda^i e^i_{t-1}$ the *trace vector* ($e^i_0=0$).

Accordingly, part 1) of the Critic algorithms attached to the nodes is defined in Equation 4, where $\alpha^i_t>0$ is the step size (to be specified later).

$$\tilde{\theta}^i_t=\theta^i_t+\alpha^i_t\rho^i_t\delta^i_t e^i_t$$

Equation 4 – Part 1 of the Critic algorithm

The part 2) is defined in the form given in Equation 5, where α^{ij}_t are elements of an $N\times N$ random matrix $A_t=\left[\alpha^{ij}_t\right],i,j=1,...,N,\alpha^{ij}_t\ge 0$, which is row-stochastic ($\forall t\ge 0$), with $\alpha^{ij}_t=0$ for all (j,i) not belonging to the set of directed arcs $\mathcal{N}$.

$$\theta^i_{t+1}=\sum_{j\in N_i}\alpha^{ij}_t\tilde{\theta}^j_t$$

Equation 5 – Part 2 of the Critic algorithm

The complete Critic algorithm will be denoted as AlgC.

4. ACTOR: ALGORITHM DERIVED FROM DISTRIBUTED TD(Λ)

4.1. POLICY GRADIENTS

The starting relation is $\nabla w^i\sum_{j=1}^{N}\overline{\psi}^j E\left\{\delta^i_t e^i_t\right\}=0,i=1,...,N$. Consequently, Equation 6 is obtained.

$$\sum_j\overline{\psi}^j E\left\{\nabla\left(w^i\rho^j_t\right)\delta^j_t\in^j_t+\rho^j_t\nabla\left(w^i\delta^j_t\right)\in^j_t+\rho^j_t\delta^j_t\left(w^i\in^j_t\right)\right\}=0$$

Equation 6

From Equation 6, the following expression for $\nabla w^i\theta^{iT}$ is obtained directly (see [11] for the single agent case), *i.e.*,

$$\frac{\partial\theta^{iT}}{\partial w^i}=\overline{\psi}^j E_i\left\{\rho^i_t\delta^i_t\left(\nabla w^i\log\pi^i_{w^i}\left(a^i_t\middle|s^i_t\right)\right)\in^{iT}_t+\nabla w^i\in^{iT}_t\right\}\left(\sum_j\overline{\psi}^j A^{\lambda^j,j}\right)$$

Equation 7

where $A^{\lambda^j,j}=E_j\left\{\rho^j_t\left(\varphi^j_t-\gamma^j\varphi^j_{t+1}\right)\in^{jT}_t\right\},j=1,...,N$.

Let $\eta=\left(\sum_j\overline{\psi}^j A^{\lambda^j,j}\right)^{-1}\sum_j\overline{\psi}^j E_j\left\{\varphi^j_t\right\}$, $f^{\lambda^j,j}_t=\in^{jT}\eta$ and $f^{\lambda^j,j}(s)=E_j\left\{f^{\lambda^j,j}_t\middle|s^j_t=s\right\}=E_j\left\{\in^j_t\middle|s^j_t=s\right\}\eta$. Accordingly, we have Equation 8:

$$\sum_j \overline{\psi}^j A^{\lambda^j,\alpha^j} = \sum_j \overline{\psi}^j E_j \left\{ \rho_t^j \left(\varphi_t^j - \gamma^j \varphi_{t+1}^j \right) f_t^{\lambda^j,\alpha^j}(s) \right\} \sum_j \overline{\psi}^j E_j \left\{ \varphi_t^j \right\}$$

Equation 8

Therefore, the expression for the gradient of the global criterion is given by the following Equation 9.

$$\nabla w^i J\left(W\overline{\psi}\right) = \frac{\partial \theta^T}{\partial w^i} \sum_j \overline{\psi}^j E_j \left\{ \varphi_t^j \right\} = \overline{\psi}^i E_j \left\{ \rho_t^i \delta_t^i \nabla w^i \log \pi_{w^i}^i \left(a_t^i \middle| s_t^i \right) f^{\lambda^i,\alpha^i}(s) \nabla w^i \in_t^{iT} \eta \right\}$$

Equation 9

In general, the policy gradient defined by Equation 9 can lead to nontrivial implementation problems, especially in relation with the terms $f_t^{\lambda^i,\alpha^i}$ and $(\nabla w^i \in_t^i)\eta$. However, when the value function is estimated in the Critic part by TD(λ), the solutions become simple and computationally attractive [11, 25].

4.2. POLICY GRADIENT IN THE ON-POLICY SCENARIO

It is easy to demonstrate that the algorithms TD(λ), GTD(λ) and ETD(λ) are equivalent in the on-policy scenario. In order to derive an algorithm for the Actor part on the basis of the exact gradients of the criterion function presented in Section 2, it is necessary to reconsider the derivation presented in the preceding subsection.

Theorem 1. *For the problem of on-policy estimation, the following holds:* $(1-\gamma\lambda^i)\nabla w^i J\left(W;\overline{\psi}\right) = \left\{ \delta_t^i \nabla w^i \log \pi_{w^i} \left(a_t^i \middle| s_t^i \right) \right\}$.

Proof. The proof is based on demonstrating that, in the case of on-policy estimation, f^{λ^i,α^i} satisfies Equation 8. We can derive the following expression $E_j \left\{ \rho_t^j \left(\varphi_t^j - \gamma^j \varphi_{t+1}^j \right) f_t^{\lambda^j,\alpha^j}(s) \right\} = \sum_{s',a} \left(d_{w^i,\alpha^i}(s) - \gamma\lambda^{i\,w^i,\alpha^i}(s'|s) \right) \frac{1}{1-\gamma\lambda^i} \left[\sum_s d_\pi(s)\varphi^i(s) - \gamma\lambda^i \sum_{s'} \left(\sum_s d_{w^i,\alpha^i}(s) P^{w^i,\alpha^i}(s'|s) \right) \varphi^i(s') \right] = E_j \left\{ \varphi_t^j \right\}$ where we have exploited the fact that. $\sum d_{w^i,\alpha^i}(s) P^{w^i,\alpha^i}(s'|s) = d_{w^i,\alpha^i}(s')$ Hence, the result follows. □

4.3. ALGORITHM FOR THE ACTOR STAGE

According to the derivation from the previous subsection, we use the basic relation for the on-policy scenario, and obtain for t≥0 the corresponding iterations for the estimation of the local policy parameters in the Actor part, given in Equation 10, where $\tilde{e}_t^i = \nabla w^i \log \pi_{w^i} \left(a_t^i \middle| s_t^i \right) + \gamma^i \tilde{e}_{t-1}^i, \tilde{e}_{-1}^i = 0$,while δ_t^i is defined above.

$$w_{t+1}^i = w_t^i + \beta_t^i \rho_t^i \delta_t^i \tilde{e}_t^i$$

Equation 10 Actor part of the algorithm

To achieve two-time-scale functioning of the whole AC algorithm, we adopt $\beta_t^i \ll a_t^i, i = 1,...,N$.

In the case of the idea exposed in Section 2 to find the optimal policy function common for all the agents, it is, simply, necessary to add to Equation 10 a "consensus" part of the algorithm, identical to the one formulated in Equation 5 obtained after replacing θ for w. Then, under appropriate assumptions the algorithm asymptotically provides consensus, i.e. $w^1=\cdots=w^N=w^*$. The meaning of such a result is intuitively clear: the obtained solution provides an "average" solution maximizing the scalarizing objective function, but not fulfilling, in general, optimality conditions for any of the agents within the scope of the multi-task problem posed.

5. CONCLUSION

In this paper, we proposed a new distributed on-policy Actor-Critic algorithm using the TD(λ) algorithm with function approximation and dynamic consensus at the Critic stage, and the consensus-based policy gradient algorithm at the Actor stage. The gradient has been derived starting from a multi-task problem formulation and a global objective representing a weighted sum of local criteria. The algorithm can be highly effective in practice, achieving improved rate of convergence and general estimation covariance reduction.

Future work will be directed towards rigorous convergence analysis of the proposed scheme, as well as in depth simulation-based verifications.

6. ACKNOWLEDGEMENTS

This research was supported by the Science Fund of the Republic of Serbia, Grant #6524745, AI-DECIDE.

7. REFERENCES

[1] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT press Cambridge, 2017.

[2] V. Mnih., K. Karavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmilelr, A. K. Fiedjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King and D. Kumaran, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, p. 1307, 2015.

[3] H. R. Maei, C. Szepesvari, S. Bhatnagar and R. S. Sutton, "Toward off policy learning control with function approximation," in *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[4] R. S. Sutton, C. Czepesvari and H. R. Maei, "A convergent o(n) algorithm for off-policy temporal difference learning with linear function approximation," *Advances in neural information processing*, vol. 21, p. 1609–1616, 2008.

[5] 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 *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[6] M. Geist and B. Scherrer, "Off-policy Learning With Eligibility Traces: A Survey," *Journal of Machine Learning Research*, vol. 15, pp. 289-333, 2014.

[7] R. S. Sutton, D. A. M. Allester, S. P. Singh and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000.

[8] V. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM Journal on Control and Optimization*, vol. 42, p. 1143–14166, 2003.

[9] S. Bhatnagar, R. S. Sutton, R. Ghavamzadeh and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, p. 2471–2482, 2009.

[10] S. Srinavasan, M. Lanctot, V. Zambaldi, J. Perolat, K. Tuyls, R. Munos and M. Bowling, "Actor-critic policy optimization in partially observable multiagent environments," in *Advances in Neural Information Processing Systems*, 2018.

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

[12] A. OroojlooyJadid and D. Hajinezhad, "A Review of Cooperative Multi-Agent Deep Reinforcement Learning," *arXiv:1908.03963*, 2019.

[13] 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, p. 156–172, 2008.

[14] S. V. Macua, J. Chen, S. Zazo and A. H. Sayed, "Distributed policy evaluation under multiple behavior strategies," *IEEE Transactions on Automatic Control,* vol. 60, p. 1260–1274, 2015.

[15] M. S. Stanković, S. S. Stanković and K. H. Johansson, "Distributed time synchronization for networks with random delays and measurement noise," *Automatica*, vol. 93, pp. 126-137, 2018.

[16] M. S. Stanković, M. Beko and S. S. Stanković, "Distributed Value Function Approximation for Collaborative Multi-Agent Reinforcement Learning," *IEEE Transactions on Control of Network Systems*, vol. 8, pp. 1270-1280, 2021.

[17] T. Doan, S. Maguluri and J. Romberg, "Finite-Time Analysis of Distributed TD(0) with Linear Function Approximation on Multi-Agent Reinforcement Learning," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[18] J. K. Gupta, M. Egorov and M. Kochenderfer, "Cooperative Multi-agent Control Using Deep Reinforcement Learning," in *Autonomous Agents and Multiagent Systems*, Cham, 2017.

[19] M. S. Stanković, N. Ilić and S. S. Stanković, "Distributed Stochastic Approximation: Weak Convergence and Network Design," *IEEE Transactions on Automatic Control*, vol. 61, pp. 4069-4074, 2016.

[20] S. S. Stanković, N. Ilić and M. S. Stanković, "Adaptive Consensus-based Distributed System for Multisensor Multitarget Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, 2021.

[21] M. S. Stankovic, M. Beko and S. S. S., "Distributed Consensus-Based Multi-Agent Off-Policy Temporal-Difference Learning," in *60th IEEE Conference on Decision and Control (CDC)*, 2021.

[22] K. Zhang, Z. Yang, H. Liu, T. Zhang and T. Basar, "Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents," *arXiv:1802.08757*, 2018.

[23] Y. Zhang and M. M. Zavlanos, "Distributed off-Policy Actor-Critic Reinforcement Learning with Policy Consensus," *arXiv:1903.09255*, 2019.

[24] S. V. Macua, A. Tukiainen, D. G.-O. Hernandez, D. Baldazo, E. M. de Cote and S. Zazo, "Diff-DAC: Distributed Actor-Critic for Average Multitask Deep Reinforcement Learning," *arXiv 1710.10363*, 2019.

[25] M. S. Stankovic, M. Beko and S. S. S., "Distributed Actor-Critic Consensus-Based Learning Using Emphatic Weightings," in 8*th International Conference on Control, Decision and Information Technology, CoDIT'22*, Istanbul, 2022.

[26] M. S. Stankovic, M. Beko and S. S. S., "Convergent Consensus-based Off-Policy Actor-Critic Algorithm for Distributed Reinforcement Learning," in *30th European Signal Processing Conference, EUSIPCO 2022*, 2022.

[27] T. Degris, M. White and R. S. Sutton, "Off policy actor critic," in *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[28] 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.

[29] M. S. Stanković, S. S. Stanković and D. M. Stipanović, "Consensus-based decentralized real-time identification of large-scale systems," *Automatica*, vol. 60, p. 219–226, 2015.

[30] M. S. Stanković, S. S. Stanković and K. H. Johansson, "Asynchronous Distributed Blind Calibration of Sensor Networks Under Noisy Measurements," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 571-582, 2018.