



June 2024, Special Volume 1, Issue 2

A Master-Slave Approach for Simultaneously Controlling Two Drones when Carrying an Object

Seyyed Mohammad Ali Ardehali, Code Orkid: 0009-0006-7526-511X

*Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran,
seyyedmohammad.ardehali@gmail.com*

Monireh Abdoos, Code Orkid: 0000-0002-3106-503X

Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran, m_abdoos@sbu.ac.ir

Amin Faraji, Code Orkid: 0000-0001-6139-6190

Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran, e.aminfaraji@gmail.com

Armin Salimi-Badr[✉], Code Orkid: 0000-0001-6613-7921

Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran, a_salimibadr@sbu.ac.ir

Abstract- This paper proposes a master-slave approach to simultaneously control two drones with the aim of carrying an object toward a goal. The proposed method utilizes the Double Deep Q-Learning (DDQN) technique to train a master agent to be able to carry an object toward a goal with the help of a slave agent. This procedure is implemented such that the master agent gathers the observations and specifies the actions to be made by itself and the slave agent. Indeed, the slave agent just applies a predefined action and does not process any input for producing the output. This manner of learning, leads to a unified convergence to an optimal solution compared to the situation in which each agent is trained separately. To verify the functionality of the proposed method, the algorithm is examined in the webots simulation environment. The simulations show that the introduced method has a good performance when controlling the drones to reach to the goal. The introduced method, other than algorithmic benefits which leads to a faster convergence of the model, suggests some reduction in the processing demand. The reason is that the learning procedure is guided by one of the agents and consequently only one of the agents is responsible for doing the calculations that lead to choosing the action. In this scenario, the slave agent does not require any processing resources for choosing the action and just simply applies a predefined action dictated by the master agent.



Keywords—Index Terms— Reinforcement Learning, Double Deep Q-Learning, master-slave approach.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have significantly brought attentions in recent decades. Although the UAVs are preliminarily designed to be used in military missions that are too dull, dirty or hazardous for humans, their applications have immediately expanded to commercial, recreational, scientific, and others [1]. They have been successful in many commercial applications, such as aerial photography [2], agricultural service [3], search and rescue, and so on.

One of the ongoing topics in UAVs is carrying an object by them [4]. Recent researches have suggested control solutions for the manipulation and transportation of the objects using quadrotors working together. For instance, in [5], the position and attitude of the objects are controlled in a cable-driven parallel robot fashion. Another approach without communication has been introduced in [6] that utilizes force feedback for controlling the pose of the object. In [7], the manipulation task is done by a quadrotor formation, which benefits from easy configuration and trajectory/task planning for the robots.

Some other works can be also mentioned as successful examples of controlling aerial vehicles that have been well performed and shown notable functionality, such as planning them to be able to navigate between small openings [8] and perform aerobatics [9]. In addition, the mentioned approaches have enabled aerial vehicles to beneficially control the objects [10].

A good option for enabling the UAVs for carrying the objects is training them with the Reinforcement Learning (RL) algorithms. The RL algorithms have the ability to teach the agents to make good decisions when they encounter different situations. The RL algorithms enable the agent to this capability through a training procedure in which the agent receives feedback from the environment according to its performance. Indeed, the agent will receive positive rewards when it performs the actions that result in going toward solving the problem, such as getting close to the goal. On the other hand, it will receive negative rewards (punishment) when doing actions that lead to getting away from solving the problem, for example going away from the goal.

A Master-Slave Approach for Simultaneously Controlling Two Drones when Carrying an Object

In [11], reinforcement learning was utilized for reaching end-to-end (i.e., from load pick-up to delivery) object transportation, in which a meta-learning method is used for updating the dynamic model of the system as soon as facing the variations in the object.

In another work, reinforcement learning was used to transport objects with the help of more than one quadrotor, in such a way that learning was used to plan smooth and swing-free trajectories [12].

In [13] some applications were regarded considering one vehicle and cooperative transportation.

In this paper, a master-slave approach is introduced to plan carrying an object using two agents in a unified procedure, preventing the agents from stochastic learning. In this approach, one of the agents is considered the master agent that is responsible for determining the appropriate action (i.e., the movements of the robots) and also performing a part of that. Besides, the other agent is considered the slave agent that just simply receives the dictated action and performs it.

The rest of the paper is organized as follows: the methodology, including the prerequisites and the proposed method, is presented in section II. In section III, the results of applying the proposed are shown. Finally, section IV concludes this paper.

II. METHODOLOGY

In this section, the proposed method is introduced and detailed for simultaneously controlling two drones in order to deliver an object to a target. The idea behind the proposed method is managing the agents by using an RL algorithm in a master-slave approach. In this manner, there should be a planner agent that decides which actions should be made according to the different states. In addition, another agent must perform the actions that are dictated to it. The former agent is called the master and the latter is called the slave.

Utilizing the master-slave approach leads to a stable RL learning for the problem, resulting in a more purposeful procedure compared to the situation in which each agent is trained separately.

A. Double DQN

The algorithm used in the proposed method is the Double Deep Q-Network (DDQN) which is one of the useful RL algorithms. This model, same as DQN, aims at updating its weights based on the gathered experience during the RL procedure. The mentioned experiences affect the network updating through the following formulation:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \lambda \max_{a'} Q(s', a') - Q(s, a)) \quad (1)$$

where $Q(s, a)$, α , r , λ are the value of the state-action value function, learning rate, reward, and discount factor, respectively. Also, s , a , s' , a' demonstrate current state, current action, next state, and next action, respectively.

The DDQN algorithm, same as DQN, collects some experiences after performing each action of an episode. After finishing the episode, these experiences are used to update the network.

Indeed, leveraging (1) in the DDQN training procedure leads the network to update its target toward a more beneficial procedure resulting in more rewards. Doing this causes the learning of the network to be performed such that the agent applies an action that returns more reward.

The difference between DQN and DDQN is in the period of time that the network is updated. In the former, the network is updated per episode of learning while in the latter, the update is performed every n episode. Doing this prevents the network from changing repeatedly and gives more chances for the trajectories to act in the problem space.

The DDQN consists of two networks, one to be updated per episode called policy network and one to be updated per n episode called target network. Also, it should be mentioned that during the RL procedure, the actions are chosen according to the policy network and the target network is used to cooperate in calculating (1).

B. Problem Space

In this section, the space of the problem is described to intricate what actions, states, and rewards have been considered for the problem.

B.1 Actions

In this paper, with the aim of abstracting the modeling, the possible primary actions of the drones, i.e., moving forward, and backward, going up and down, and turning left and right have been shortened to three actions.

To control the agents to move together, three actions have been considered to be performed. These actions are moving forward, turning right, and turning left. Thus, two agents can be considered as the wheels of a vehicle that is able to do the mentioned actions. To do these actions, each agent should go forward or backward. For example, if the action is going forward, both agents should go forward but when the action is turning left, the left agent should go backward and the right agent should go forward and vice-versa.

B.2 States

The state considered for the problem, includes the distance of the agents to the target and their coordinates in the space along with the changing values of the roll, pitch, and yaw axis. Therefore, the state of each agent will be a vector of size 7.

The mentioned type of the state causes the agents to be aware of their situation in the environment as the state reflects the distance to the goal, the current coordinate of the agent in the environment and the amount of the changes that the agent undergoes with respect to the roll, pitch and yaw axis.

B. 3. Rewards

To specify the reward for each action, the distance to the goal is considered to encourage or punish the agents. This distance is the Euclidian distance of the agent from the goal in the 3-dimensional space of the problem.

The rewards have been specified hierarchically, the closer to the target, the more positive reward is gained and vice-versa. Table I demonstrates the rewards considered for the problem. The rewards of the table have been calculated according to the following formula:

TABLE I. THE REWARDS ALLOCATED TO THE AGENTS ACCORDING TO PROXIMITY TO THE GOAL

Situation	Growth Factor	Base Reward	Total Reward
$d < 0.4$	-	-	1000
$d < 5$	15	12	11.4
$d < 10$	12	10	9
$d < 15$	10	9	7.7
$d < 20$	9	8.5	7
$d < 25$	8	8	6.3
$d < 30$	7	7.5	5.6
$d < 35$	4.5	6	3.5
$d > 35$	-	-	$-d/100$
Getting close to the goal in two consequent actions	-	-	0.3
getting away from the goal in two consequent actions	-	-	-2
having collusion	-	-	-10

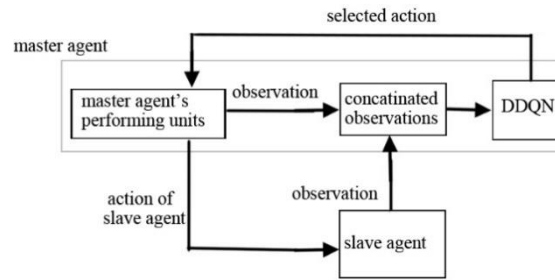


Fig. 1. The structure of the proposed method

$$total\ reward = b \times (1 - e^{(-g \times (1/d))}) \quad (2)$$

where b , g , and d stand for base reward, growth factor, and distance to the goal, respectively.

B. 4. Equipment

To gather the observations from the environment, the agents utilize the GPS and Inertial Unit sensors to correctly be informed about the states that they are in. The former is used to specify the coordinate that the agents are acting in and the latter is used to measure the amount of the changes in the pitch, roll, and yaw axis of the agents. Also, a touch sensor is used when training the agents to check if there is a collision in the environment or not.

Since a part of the proposed method, provided later, includes communication between two agents, there should be some devices to perform the sending and receiving operations. To do these, the emitter and the receiver devices are embedded on the drones.

C. The Proposed Method

In this section, the structure of the proposed method is described in which the master-slave approach controls the actions to be performed by each agent. Also, the advantages of the proposed method are mentioned here.

C. 1. The Structure of the Proposed Method

The general structure of the proposed method is illustrated in Fig. 1. As can be seen in the figure, the proposed method consists of a master agent that generally controls the learning procedure and a slave agent that just performs a predefined action. Also, a detailed scheme of the proposed method has been provided in Fig. 2. The master agent receives the observations and passes them to a deep network in order to receive the corresponding action. The concatenation presented in Fig. 2, comes from combining two vectors of the observations provided by two agents. Each vector consists of 7 values, as discussed in section II. B. 2. Thus, after concatenation, a vector of size 14 is produced to be fed into the DDQN to the corresponding action be determined by this network.

A Master-Slave Approach for Simultaneously Controlling Two Drones when Carrying an Object

After choosing the action, the master agent performs the part of the action related to itself and dictates another part to the slave agent. It means that when an action is going to be done, each agent should act such that the resulted action be the same one determined by the deep network. For instance, if the action is turning right, the left agent should go forward and the right agent should go backward.

When the actions are made by the agents, their observations are concatenated to form the new observation to be used in the next step of the RL learning.

As can be seen in Fig. 1, the slave agent just performs the predefined action and does not process any input for producing the output.

C. 2. Advantages of the Proposed method

The aforementioned procedure for the proposed method causes a unified policy for controlling two agents when going toward the goal. It can be said that the master-slave approach, makes the problem notably get rid of the stochastic behavior during the training. When two agents are trained individually, it is more likely that a time-consuming convergence accrue. The reason is that when each agent is going to perform its desired actions, the other agent's action will affect its action and the overall action will be ruined. In this situation, it is expected that the RL training faces an unstable training procedure.

Other than the mentioned algorithmic advantages of the proposed method, the master-slave approach leads to some reduction in the processing required for choosing the action. The reason for this reduction is the fact that in the master-slave manner, only one of the agents (master) is responsible for doing the mathematical calculations for determining the actions. These pros, along with the algorithmic advantages, make the proposed method a nice option for the problem of simultaneously carrying the object.

In addition to the discussed pros of the proposed method, also there exists another aspect that can be noticed. The proposed method follows a master-slave approach in which the agents cooperate together to carry an object toward a goal. The actions that are made by the agents are chosen based on a deep neural network used in the DDQN algorithm. The procedure of the proposed method does not insist on the usage of a particular kind of algorithm to be used in the RL and allows the other RL algorithms to be used in the proposed master-slave approach.

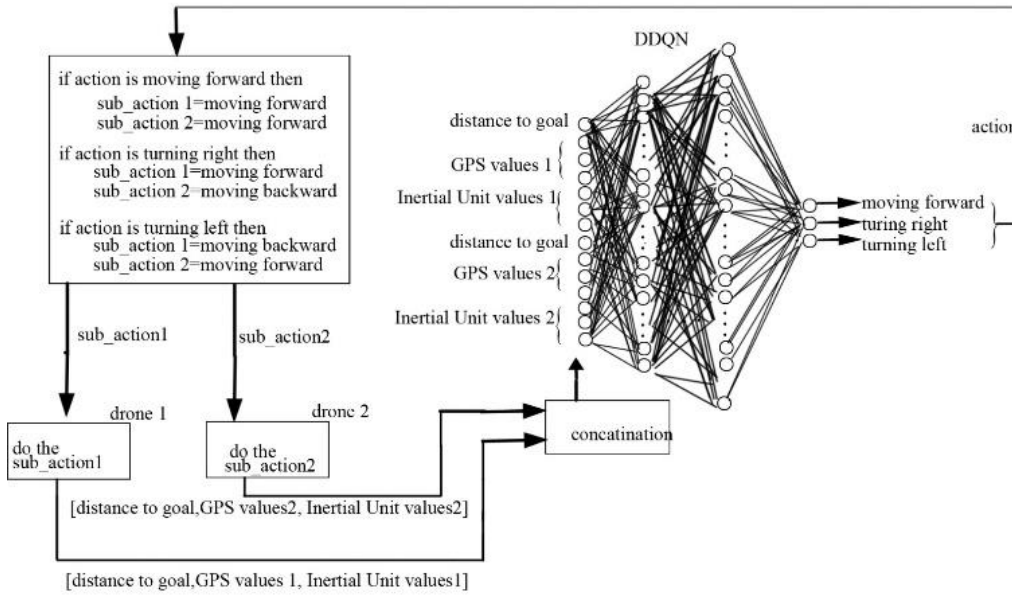


Fig. 2. A detailed structure of the proposed method

III. EXPERIMENTS

In this section, the results of implementing the proposed method are presented to validate the functionality of that. To do this, at first, a description of the simulation environment is provided and afterward, the simulation results are presented and discussed.

A. Simulation Environment Setting

In order to examine the performance of the proposed method, two MAVIC robots, which is a quadcopter drone, have been launched in the webots simulation software and the algorithm of the proposed method was implemented as their controller.

The general view of the environment has been demonstrated in Fig. 3 showing the drones and the target which is positioned at the end of the road. Indeed, at first, the drones launch from the ground and afterward, discover the world in the sky regarding the designed actions. The purpose of the problem is to control the drones in such a way that they are capable of carrying the object toward the target while preserving the balance to avoid dropping it.

To reach the mentioned purpose, the following strategy is applied; one of the robots is controlled by the master controller that determines the action (performing its action and sending the corresponding action of the slave robot) and concatenates the observations of two robots. The other robot is controlled by a simple controller that just receives and performs the desired action.

The aforementioned send/receive pulses are performed using emitter and receiver sensors. Also, the agents' movements are designed to apply an almost 10-degree rotation to the left or right for the corresponding actions.

To be aware of the states that robots are moving in, the GPS sensor is utilized to prepare the coordinates of the

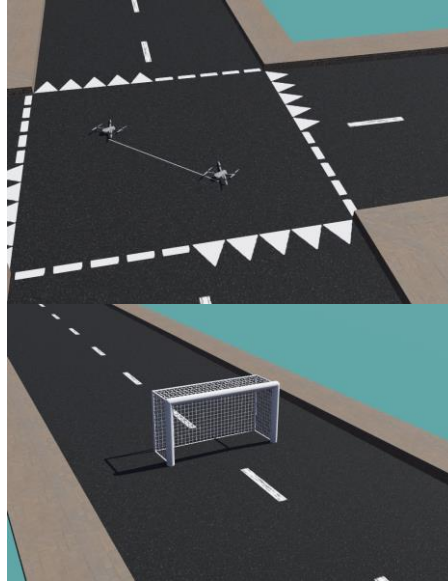


Fig. 3. The simulation environment including the position of the drones (up) and the target (down).

agents and the Inertial Unit sensor is used to measure the roll, pitch, and yaw changes. In addition, a touch sensor is embedded on the agents to correctly terminate the unsuitable episodes.

B. Numerical Results

In this section, the rewards that have been achieved during the training, are considered as a criterion to evaluate the functionality of the proposed method. Fig. 4 demonstrates the rewards achieved during training the proposed method for carrying the object toward the goal. As can be seen in the figure, in the initial episodes, the rewards

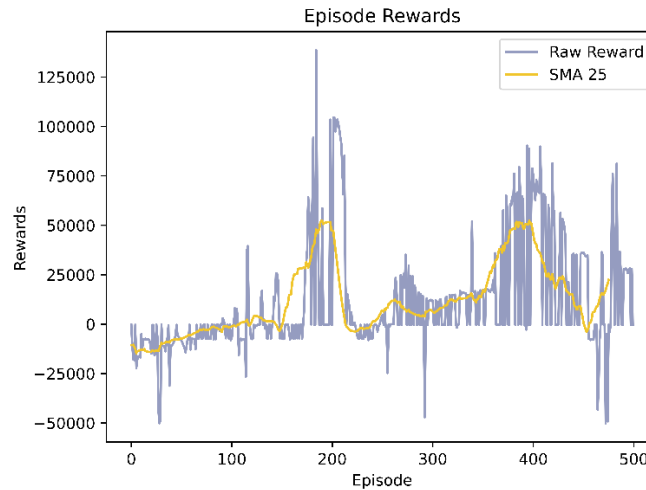


Fig. 4. The rewards achieved per episode of training when using the proposed method for carrying the object toward the goal

are mostly large negative values. As the training goes on, the agents get closer to the goal and some larger positive rewards are achieved and after a while, the agents mostly gain positive results. This means that the proposed method is able to force the agents to move around the goal instead of the preliminary dummy moves. Also, when it came to the inference, the agents showed a good performance as they reached exactly to the goal.

IV. CONCLUSION

In this paper, a master-slave approach was introduced to be utilized for controlling two drones simultaneously aiming at carrying an object toward a goal. The proposed master-slave method is designed such that one of the drones is specified as the master agent responsible for gathering the observations and choosing the corresponding action. As the final action of the system should be performed cooperating two drones, the master agent applies the part of the action related to itself and sends the other part of the action to the slave agent. The proposed master-slave approach led to a homogenous training as both agents collaborate for doing a unified action. This type of action is opposite to the one in which each agent is trained and acts separately which leads to a time-consuming training procedure.

The experiments, showed successful performance of the algorithm as the agents were able to drive toward the goal when preserving the object. Other than the mentioned algorithmically benefits, the proposed method required less processing abilities. The reason is that only one of the agents is responsible for doing the calculation for choosing the action, so there is no need for another agent to do the calculation.

The aforementioned advantages can suggest the proposed method as a suitable option for controlling two drones with the aim of carrying an object toward a goal. The proposed method applies a master-slave technique through the DDQN algorithm and can be also expanded to the RL algorithms other than DDQN reflecting the generic aspect of the proposed method

REFERENCES

- [1] Y. Wang, J. Sun, H. He and C. Sun, "Deterministic policy gradient with integral compensator for robust quadrotor control", *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 10, pp. 3713-3725, Oct. 2020.
- [2] R. G. Valenti, Y.-D. Jian, K. Ni, and J. Xiao, "An autonomous flyer photographer," in *Proc. IEEE Int. Conf. Cyber Technol. Autom. Control Intell. Syst. (CYBER)*, 2016, pp. 273–278.
- [3] J. Valente, J. Del Cerro, A. Barrientos, and D. Sanz, "Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach," *Comput. Electron. Agricult.*, vol. 99, pp. 153–159, Nov. 2013.
- [4] S. Belkhale, R. Li, G. Kahn, R. McAllister, R. Calandra and S. Levine, "Model-based meta-reinforcement learning for flight with suspended payloads", *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1471-1478, Apr. 2021.
- [5] D. Sanalidro, H. J. Savino, M. Tognon, J. Cortes and A. Franchi, "Full-pose manipulation control of a cable-suspended load with multiple UAVs under uncertainties", *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2185-2191, Apr. 2020.
- [6] C. Gabellieri, M. Tognon, D. Sanalidro, L. Pallottino and A. Franchi, "A study on force-based collaboration in swarms", *Swarm Intell.*, vol. 14, no. 1, pp. 57-82, Mar. 2020.
- [7] K. Gkountas and A. Tzes, "Leader/follower force control of aerial manipulators", *IEEE Access*, vol. 9, pp. 17584-17595, 2021.
- [8] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 664–674, 2012.
- [9] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadcopter multi-flips," in *Proc. Int. Conf. Robot. Automat.*, 2010, pp. 1642–1648.
- [10] S. Tang, V. Wüest, and V. Kumar, "Aggressive flight with suspended payloads using vision-based control," *Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1152–1159, 2018.
- [11] S. Belkhale, R. Li, G. Kahn, R. McAllister, R. Calandra and S. Levine, "Model-based meta-reinforcement learning for flight with suspended payloads", *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1471-1478, Apr. 2021.
- [12] X. Li, J. Zhang, and J. Han, "Trajectory planning of load transportation with multi-quadrotors based on reinforcement learning algorithm," *Aerosp. Sci. Technol.*, vol. 116, Sep. 2021, Art. no. 106887.
- [13] Y. Liu, F. Zhang, P. Huang, and X. Zhang, "Analysis, planning and control for cooperative transportation of tethered multi-rotor UAVs," *Aerosp. Sci. Technol.*, vol. 113, Jun. 2021, Art. no. 106673.



Mohammad Ali Ardehali received the B.S. degree in computer engineering from **K. N. Toosi University of Technology, Tehran, Iran, in 2021**. He is currently pursuing a master's degree in **artificial intelligence at Shahid Beheshti University, Tehran, Iran**. His research interests include **machine learning, deep learning, and deep reinforcement learning**.
<https://orcid.org/0009-0006-7526-511X>




Amin Faraji received the B.S. degree in computer engineering from University of Mohaghegh Ardabili, Ardabil, Iran, in 2019 and the master's degree in artificial intelligence at Yazd University, Computer Engineering Department, Yazd, Iran, in 2022. He is now pursuing Ph.D. in artificial intelligence at Shahid Beheshti University, Tehran, Iran. His research interests include deep learning and neural networks and their applications.
<https://orcid.org/0000-0001-6139-6190>



Monireh Abdoos received her Ph.D. in Computer-Engineering-Artificial Intelligence from Iran University of Science and Technology, Tehran, Iran in 2013. She is currently an assistant professor at Faculty of Computer Science and Engineering in Shahid Beheshti University, Tehran, Iran. Her research interests include: Multi-agent Systems, Soft Computing, Machine learning and Intelligent Transportation System
<https://orcid.org/0000-0002-3106-503X>



Armin Salimi-Badr  received the B.Sc., M.Sc. and PhD degrees in Computer Engineering, all from Amirkabir University of Technology, Tehran, Iran in 2010, 2012, and 2018 respectively. He also obtained a PhD degree in Neuroscience from University of Burgundy, Dijon, France in 2019, where he was researching on presenting a computational model of brain motor control in the Laboratory 1093 CAPS (Cognition, Action, et Plasticité Sensorimotrice) of the Institut National de la Santé et de la Recherche Médicale (INSERM). He was a Postdoctoral Research Fellow at Biocomputing lab of Amirkabir University of Technology from October 2019 to September 2020. Currently, he is an Assistant Professor at Faculty of Computer Science and Engineering of Shahid Beheshti University, Tehran, Iran and also the Head of

Artificial Intelligence & Robotics & Cognitive Computing group in this faculty. He is also the founder and Chair of Robotics & Intelligent Autonomous Agents (RoIAA) Lab in Shahid Beheshti University. He is IEEE Senior Member and currently the Chair of Professional Activities Committee and a Board Member of Computer Society of IEEE Iran Section. He was the recipient of the IEEE Young Investigator Award from the IEEE Iran Section in 2024. He is also the Distinguished Researcher of Shahid Beheshti University in the field of Computer Science and Engineering in 2025. His research interests include Computational Intelligence, Computational Neuroscience, and Robotics.
0000-0001-6613-7921