2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Economic Platform for Action Model Sharing Based on Blockchain in Mobile Robot Networks

Yu Sakai[1] and Kazuyuki Morioka[1]

***Abstract*— Data sharing among robots will be an important issue in a future robotic society including many kinds of networked robots. This study proposes a model-sharing system for autonomous mobile robot networks. The model in this study means an action model trained by deep reinforcement learning for mobile robot navigation. The proposed system assumes that the mobile robots in the network can share useful and efficient models among robots. The proposed system uses the Ethereum blockchain as a platform to consider the value and ownership of action models. The owners of the models can receive payments from robots that use the shared models for autonomous navigation. The proposed system finally aims to build a future robotic economic system that includes data generation and accumulation, data sharing, and currency transactions among model owners and user robots. This paper introduces the autonomous navigation method of switching action models according to the navigation environment, the data-sharing system using the Ethereum blockchain, and the flow of autonomous navigation using the proposed system. Simulation results shows that the proposed system can achieve navigation of multiple mobile robots through model and currency transactions among Ethereum network participants.**

I. Introduction

In recent years, mobile robots have been expanding for several kinds of applications such as logistics in warehouses and restaurants. Self-driving cars are also becoming more prevalent and more familiar. These robots generally include autonomous navigation systems based on sensor data and environmental maps. In the coming robot society, the number of robots will be increased in our living environments even more than now. Then, the amount of sensor data and the other valuable information for navigation is also increased. Sharing valuable data among many mobile robots will be expected to improve robot system performance. There are several challenges in such a robot network sharing valuable data. For example, the value and ownership of sharing data should be considered. The problems of data tampering and unauthorized use also exist. This study focuses on a blockchain-based data sharing system in autonomous mobile robot network to address the above problems.

At first, sharing data in our mobile robot network is introduced. In our system, action models for mobile robot navigation are shared among mobile robots as valuable data. Generally, the common method for mobile robot navigation involves preparing environment maps based on SLAM for self-position estimation and path planning. This method requires to build environment maps in advance using expensive LiDARs. This increases the difficulty of spreading mobile robots more widely. In contrast to these conventional navigation methods, autonomous navigation methods based on action models acquired through deep reinforcement learning have been also proposed[1]. Reinforcement learning is a method to obtain models for output actions based on input states. In training of models in reinforcement learning, agents iterate actions in environments, receiving states and rewards from environments. The models are updated to output an appropriate action for each state while training. The navigation method based on action models is expected to realize navigation that does not depend on environmental maps.Also, the navigation method using action models has the possibility to achieve simple robot systems with only a monocular camera as external sensors.

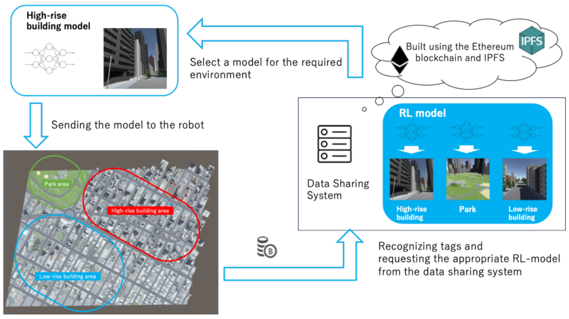

Fig. 1: System Overview

Navigation methods based on action models have the other advantages in the viewpoints of data sharing as described before. The action models can be transferred to the other robots easily, because the models are very simple structure based on image inputs and velocity outputs. Then, a new navigation method based on switching action models with the other robots can be considered according to characteristics of driving environments. Fig. 1 shows an image of this system. Additionally, there has been a concept of storing these action models on the cloud services of the Internet, enabling sharing models among robots participating in the network.

To facilitate the sharing action models among robots in the network, we propose a new type of data-sharing system using blockchain that allows consideration of the value and the ownership of shared data as described above. As an example of blockchain-based robotic systems prior to this study, sharing system of environment maps for navigation on the blockchain was proposed. The system includes a large number of unspecified robots obtained the maps from the blockchain and performed autonomous navigation [2]. In this study, we propose a robot network system that many robots

[1]Meiji Univ, Graduation School of Advanced Mathematical Sciences, Network Design Program, Japan cs243010@meiji.ac.jp

share action models on the blockchain and obtain the models from blockchain according to environments for autonomous navigation. Currency movements to model owners resulting from model transactions during navigation can be recorded in this system. This means that the data ownership can be maintained in the data sharing system.

This paper is organized as follows. II describes related works, and III introduces the details of the action model acquired by deep reinforcement learning and its learning environment. IV describes the configuration of the proposed data sharing system including an overview of blockchain and IPFS. In V, two types of autonomous driving experiments are performed using the proposed system. Finally, VII describes conclusion and future works.

II. Related Works

As mentioned before, the common method of mobile robot navigation is based on pre-prepared occupancy grid maps. In contrast, autonomous navigation methods based on action models trained with deep reinforcement learning have also been proposed. Among these methods, the model input states include traditional LiDARs and images obtained from monocular or RGB-D cameras. Data sharing system in this paper uses action models with inputs of monocular camera image for autonomous navigation of mobile robots. In our previous research, we have demonstrated the potential of deep reinforcement learning-based autonomous navigation. In these studies, inputs of the action models were 2D range data[1]. Additionally, there are several other studies on deep reinforcement learning-based autonomous navigation with action models of monocular camera images as input. Ding et al. proposed an obstacle avoidance system using a action model trained from distance information extracted from depth images obtained from monocular camera images[3]. In contrast, there are studies on creating models that use monocular camera images themselves as input for training[4][5]. Some of camera-based models were implemented to actual robots and robot navigation in real world was achieved. As introduced here, action model-based navigation has been applicable for actual robot systems. Then, the proposed system in the paper considers data sharing of action models for navigation.

In the proposed system for sharing action models, appropriate action models are selected from shared models according to the environment. Similar systems with switching control methods based on the environment have been proposed in the past. They includes changing of the action models or traditional control methods during navigation, and use multiple action models according to the situations to execute tasks[6]. Yokoyama et al. developed a robot system with legs and an arm to perform tasks such as moving objects located at a distance. This system uses several action models for achievement of tasks including navigation, grasping and placing. These three models are switched to execute the tasks[7].

This study proposes a network robot system with not only model sharing and switching but also keeping value and ownership of models based on blockchain. There are a few studies of robotic network systems using blockchain. Shibata et al. developed a robot network system for sharing environmental grid maps using blockchain and conducted a large-scale simulation with 20 mobile robots. The robots obtained the necessary maps from the blockchain according to environments and travelled to their destinations. Additionally, a similar mechanism was implemented for paying to the map owner when obtaining the maps[2]. The other study focuses on Connected and Autonomous Vehicles (CAVs) on open networks. CAVs are vulnerable to spoofing and jamming attacks generally. To address this issue, G. Raja et al. proposed a blockchain-based multi-agent reinforcement learning system to enhance the overall security of the network[8]. Isaac J. et al. utilized blockchain for the control system of UAVs, enhancing the security of swarm robot systems[9]. Thus, in the field of robotics, blockchain is primarily used to ensure the safety of open network systems composed of multiple robots not data sharing such as our proposed system. As the other application of blockchain for robotic systems, efforts have been found to utilize blockchain for coordinated navigation of multiple robots within a warehouse environment[10].

In this study, we focuses on a future robot society with many mobile robot in the realistic city environment. The robots achieve navigation tasks based on action models that shared using blockchain in this system.

III. Action model for navigation

In this chapter, we will discuss an autonomous robot navigation method based on action models obtained by deep reinforcement learning. Reinforcement learning is a method for agents to learn appropriate behavior according to their states. The action models are trained so as to maximize the reward obtained from a series of actions. The appropriate behavior in this study is for the agent to be navigated along the road to the destination without colliding with buildings or obstacles. When training is completed, a weight file of the action model is generated. The file records the weights to output the appropriate behaviors according to input states. The purpose of data sharing system in this study is to share and use the weight files of action models among robots.

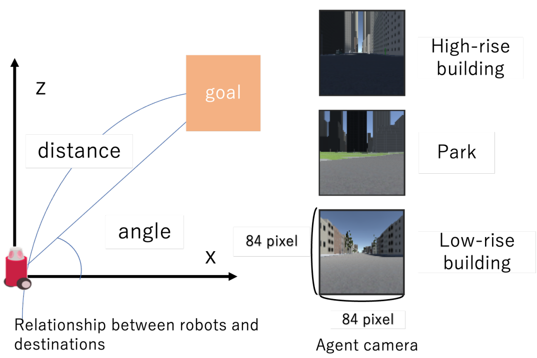

Fig. 2: Input States for Action Models

The input of the action model in this study is shown in

Fig. 2. The agent obtains monocular camera images as the input state. Right images in Fig. 2 show examples of input images according to environments such as explained below. The size of the camera image is 84 x 84 pixels. Also, the model includes the relative distance and angle between the robot and the destination in 2D ground plane. On the other hand, the models output velocity command values, which include translation and angular velocities. An actual robot system was developed with the same model structure in this study. Also, autonomous navigation was achieved in the real environments [4].

The models are trained in the simulation environment representing a part of New York City. The simulation environment is divided into three areas: ”High-rise Building area”, ”High-rise Building area”, and ”Park area”. The action models are trained for each area. PPO[11] proposed by OpenAI is used as the learning algorithm. The model are trained for 500,000 steps in each environment. The navigation experiments using trained actual models are shown in Experiment 1 of section V.

IV. Proposed System

A. Overview of Blockchain

Blockchain is a distributed ledger in which a transaction ledger called a block is shared by all network participants. It was originally developed as a fundamental technology for virtual currencies. It has four features: redundancy, tamper resistance, traceability, and transparency. These features ensure ownership guarantees and enable economic transactions [12]. Utilizing these features, blockchain has been applied to various fields in recent years. The proposed data sharing system among robots is one of such applications.

We use the Ethereum blockchain as the platform for the proposed data-sharing system. Ethereum is generally expected as a appropriate platform for applying blockchain to various services. One of its unique features is ”smart contract”. A smart contract is a program deployed on the Ethereum blockchain that automatically executes specific transactions. The details of the smart contract we have developed is described in subsection IV-C.2.

B. Data storage in Blockchain

Blockchains are unsuitable for sharing large amounts of data by their feature. Therefore, the storage system and the management system of the shared models should be divided. We uses InterPlanetary File System (IPFS) to store models for sharing in the proposed system. The IPFS is a distributed file storage protocol on a peer-to-peer network[13]. Fig. 3 shows an image of storing data on IPFS. When data is stored on IPFS, a unique hash is generated for the data content. In the proposed system, the action models are registered in IPFS and the corresponding hash values are obtained. Then, the environment name and the hash value of each action model are linked and shared to the Ethereum blockchain as the data management system. Robots that want to use the model first obtain the hash value corresponding to the desired model from the blockchain. Then, the robot retrieves the real data of the action model from the IPFS based on the hash value.

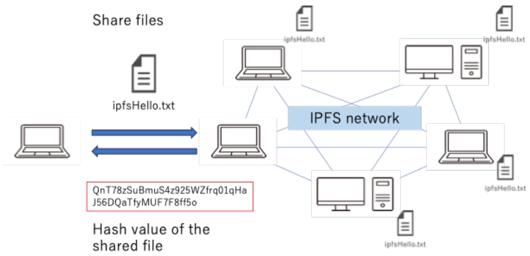

Fig. 3: Data Sharing Method in IPFS

C. Data Sharing System

1) Overview: This system assumes that the ownership of the action model is clearly defined, and transactions of virtual currency are executed between the owner and multiple users when the models are utilized. This chapter describes the details of the smart contracts developed to build this system, the pricing method, and the data sharing flow.

2) Smart Contract for Data Sharing: A smart contract is a digital agreement deployed on the Ethereum blockchain that automatically executes specific transactions. The proposed system includes a smart contract with two following functions :

① Data registration : A function that registers a set of an environment name and a hash value of each action model to the Ethereum blockchain.

② Data transaction : A function that automatically sends the hash value of the desired action model according to the environment name and automatically transfers the currency from the user to the model owner.

3) Determining Price of Action Model: There are several ways to define the price of action models depending on the situation. One method is that the model owner defines the price at registration. Another possible method is dynamic pricing. For example, a price that increases according to the number of model uses can be considered. In this system, the model price is determined by the number of times reached to the destinations using the model. This pricing means that not only the number of model uses but also model performances of navigation are included in the model price. Specifically, the price is defined by the following equations.

$$\text{fee} = \begin{cases} 0.5\,\text{ETH} & (\textit{reachCount} < 5) \\ 0.5\,\text{ETH} + \text{reachCount} \times 0.1\,\text{ETH} & (\textit{otherwise}) \end{cases}$$

reachCount: A variable that records the number of times the model was used to reach the destination.
ETH: Ethereum currency unit.

4) Flow of Data Sharing: Fig. 4 illustrates the flow of data sharing in the proposed system. In the figure, the ”Owner” means the owner of the action model, and the ”Robot” means a robot that requires the action models for autonomous navigation.

① The ”Owner” stores the action model on the IPFS network.
② The ”Owner” obtains a hash value of the registered action model from the IPFS.
③ The ”Owner” registers the set of generated hash value and an environment name to the Ethereum blockchain.
④ ”Robot” requests a hash value of the action model for the environment where the ”Robot” wants to drive. The virtual currency paid by the ”Robot” is automatically transferred to the ”Owner” through a smart contract.
⑤ The ”Robot” obtains the hash value of the desired action model from the Ethereum blockchain,
⑥ The ”Robot” requests the desired action model using the hash value to the IPFS network.
⑦ The ”Robot” downloads the action model (onnx format file) from the IPFS network.

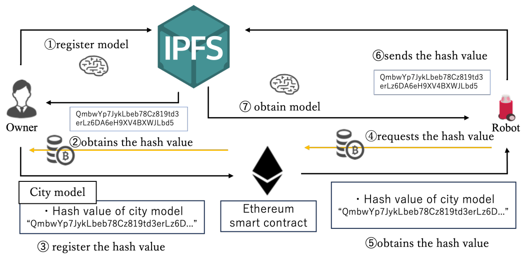

Fig. 4: Data Acquisition Using the Proposed System

5) Autonomous Navigation Using the Proposed System: This subsection describes the flow of autonomous navigation using the proposed system, as illustrated in Fig. 5. The demonstration of autonomous navigation is performed in the simulation environment built in Unity. Also, navigation system is developed in ROS platform. Every changing of current driving environments of the robot, the following process is performed to obtain the suitable action models.

① The robot in Unity recognizes the tag of its current environment.
② The ROS node receives the environment name from Unity.
③ The ROS node accesses Ethereum blockchain to obtain the hash value of the action model by paying the virtual currency.
④ Using the obtained hash value, the ROS node accesses IPFS and downloads the action model.
⑤ The downloaded action model is deployed on the ROS node.
⑥ Model inputs that includes the relative distance, angle to the destination and camera image are subscribed from Unity. Then, velocity commands are output through the action model according to the input.
⑦ The output velocity commands (translation and angular velocities) are published to the robot in Unity.
⑧ The robot in Unity autonomously travels to the destination using the action models downloaded from the proposed data sharing system.

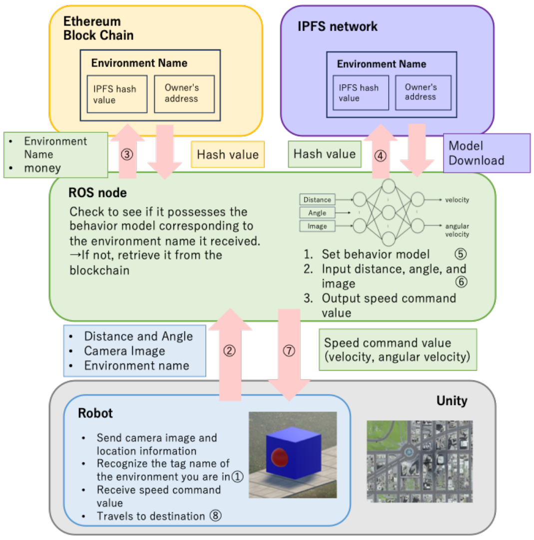

Fig. 5: Navigation Based on the Proposed System

V. Simulation Experiment

A. Experiment Overview

Experiments using a robot network including multiple mobile robots are conducted for performance confirmation of the proposed system. The experiments are performed in a simulator at first because it is difficult to develop actual systems including many mobile robots. The simulator in this experiments was created using the game engine Unity. The asset ”Real New York City Vol. 2[14]” representing realistically part of New York was imported in the simulator. This environment was divided into three areas: ”Park area” ”High-rise Building area” and ”Low-rise Building area”. Each environment is tagged for identification using Unity’s tag function. Action models for each environment were trained in advance. Fig. 6 shows the top-view image of the simulation environment and three divided areas.

Each robot equips a monocular camera as an external sensor. The robot self-position can be obtained using coordinates in Unity. The robot also recognizes the environment by reading the tag information set in the environment.

The experiment was conducted on an Ethereum private chain activated on a PC prepared for this experiment. Each Docker container that controls the robot participates in this private chain and uses the proposed system. Each Ethereum account corresponding to model owners and robots initially holds 100 ETH. The price of each action model for transaction is determined in IV-C.3.

We conducted two types of experiments. The first experiment confirms action model switching in robot navigation via two kinds of environments towards the destination. (VI-A) The robot obtains two different action models according to environments via the blockchain and navigates autonomously to its destination. In the second experiment, we simulated the entire system with the model owners and robots as users of models. (VI-B) The purpose of this experiment is to simulate autonomous navigation along the flow of IV-C.5 and to verify virtual currency transactions by the proposed system. The account balances of each robot during the simulation experiment are recorded to show transactions of models and currencies among robots and owners. In these experiments, the initial positions of the robot and the target destination for navigation are predetermined as the fixed positions in the environment. When the robot reaches its destination, its position is reset to its initial position and it repeats traveling to the same destination again.

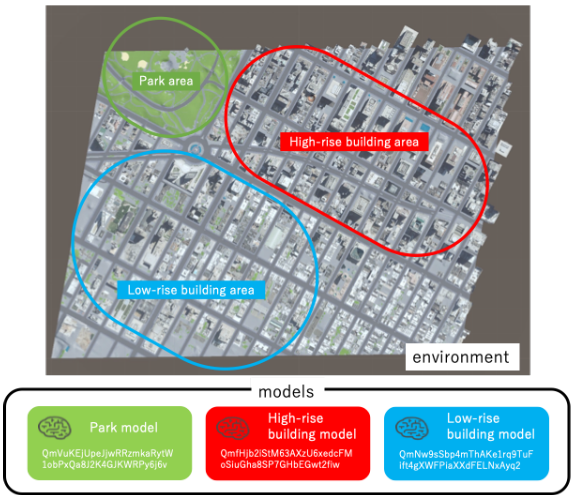

Fig. 6: Overview of the Simulation Environment

B. Control of Each Robot Using Docker

We assume a future robotic society where the number of robots increases explosively and becomes ubiquitous. Because many robots must be included in the simulation, each robot is controlled using a corresponding Docker container. This creates a similar situation with actual robot control, in which one robot in Unity corresponds to one control machine. Each Docker container participates in the Ethereum private chain and launches a ROS node for robot control. The ROS node acquires a behavior model that matches the recognized environment tag from blockchain. Then, camera images and destination information received from Unity are inputted into the model. As the result, the model outputs velocity command values and the robot performs autonomous driving. Fig. 7 shows the system configuration including multiple robots using Docker. When the robot reaches its destination, the acquired action models are deleted.

VI. Experimental Results

A. Experiment 1

Fig.. 8 shows the actual trajectory of the robot in the first experiment. In this experiment, the robot circled around one section of a building while passing through several waypoints. The red line in the figure shows the trajectory when the robot used the model for the ”High-rise building area”. In contrast, the blue line in the figure shows the trajectory when the robot used the model for the ”Low-rise building area”. The green line is the boundary between the

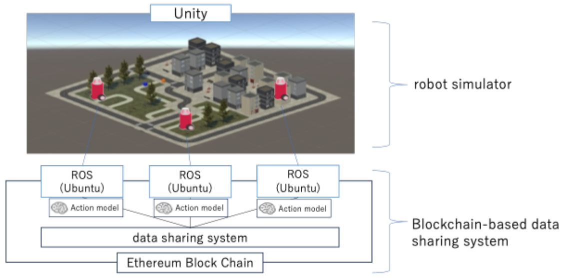

Fig. 7: Model Sharing Network Using Docker

”High-rise building area” and the ”Low-rise building area. In this experiment, the robot acquires two action models according to the environments through the blockchain. The robot could switch models for each environment and then autonomously travel to the destination.

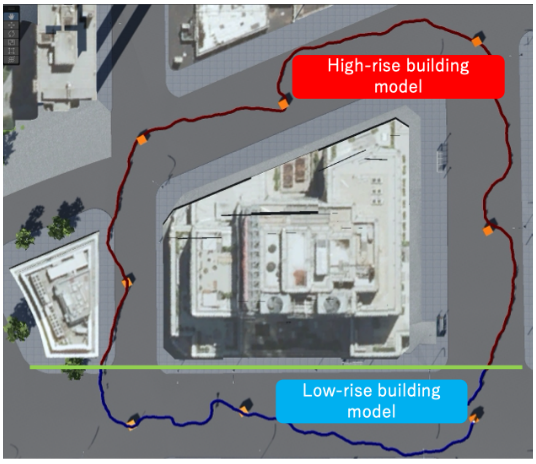

Fig. 8: An Example of Navigation Results

B. Experiment 2

In this experiment, seven robots repeatedly travel to their respective destinations.

The account balance of each robot was recorded during the experiment. Fig. 9 shows the change in the balance of each robot and model owner. Table I shows the number of times each robot downloaded each model. However, Table I does not include information on model ownership. In this experiment, all of the seven robots were able to reach their own destinations.

In this figure, the park model owner, high model owner, and low model owner mean the model owners of the ”Park area” ”High-rise building area” and ”Low-rise building area” respectively. Robot 1 to 7 are the users of the models. Each model owner does not perform navigation but has a role of only holding each action model.

The navigation of each robot is also described in detail. Robot 1,2 and 3 travels fixed routes repeatedly only in the fixed one environment respectively. Robot 1 moves in the High-rise building area. Robot 2 moves in the Low-rise building area. Robot 3 moves in the Park area. Robots 4 to

7 move across the two environments, switching the models they use along their ways, such as traveling from a "Park area" to a "High-rise building area".

Next, we discuss the results from Fig. 9 and Table I. First, the figure shows the owners' balances increased and the robots' balances decreased over time. This simply means that the model transactions among robots and model owners were correctly performed. In combination with Table I, it can also be seen that the amount paid by the robot increases as the number of reaching to the destination. As the example, the blue and red circles in Fig. 9 both represent the amount paid by robot 3, which moved in only the park environment. Comparing amount changes in the two circles, we can see that the red circle (at the end of the experiment) shows a larger change in the balance. The blue circle shows the payment after the park model was totally downloaded for the third times. In contrast, the red circle shows the payment after the 11th times. The difference in the number of downloads of the model affected in the price of the action model.

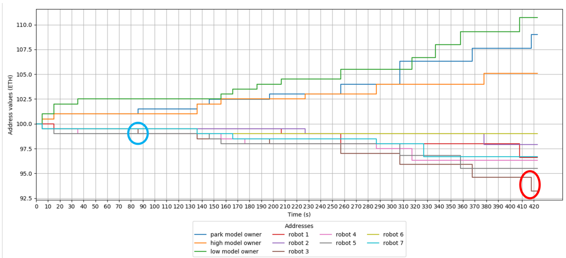

Fig. 9: Changes in the Balance of Robots and Model Owners

TABLE I: Number of Reaching to Destinations

	robot1	robot2	robot3	robot4	robot5	robot6	robot7	total
park	0	0	8	0	3	0	0	11
low	4	0	0	3	3	1	3	14
high	0	3	0	3	0	1	2	9
total	4	3	8	6	6	2	5	

Compare the balances of robots 1, 2, and 3. Table I shows that robot 1, 2, and 3 only downloaded the action models according to the environment in which they exist respectively. The table shows that the "Park" model used by Robot 3 was used 3 times less than the "High" model used by Robot 2. Thus, the price of the "Park" model is lower. However, despite the price difference between these models, the final balance was lower for Robot 3 because of the 5 times difference in the number of downloads for both robots.

Next, we compare the balances of robot 4 and 5. Both robot4 and 5 moved through two environments and downloaded the same number of models. However, it can be confirmed from Fig. 9 that there is a difference in the final balances. This is due to the difference in the total number of downloads by all robots.

VII. Conclusion and Future Work

In this paper, we proposed and implemented a new information-sharing system using the Ethereum blockchain for a future robotic society where robots are ubiquitous in everyday environments. This system guarantees the ownership and value of digital data, enabling an economic system capable of executing currency transactions. Especially, this paper proposed an action model-sharing system among robots for autonomous navigation. The simulation experiments show that multiple robots were able to autonomously travel to their destinations by acquiring suitable action models for their respective environments from the proposed system. At the same time, currency transactions according to the acquisition of action models can be performed in the simulation. This demonstrates the possibility of realizing a robot network with economical aspects for model sharing. Finally, the system will be extended in the future to handle the cases that multiple models with different performances for the same environment are stored.

References

[1] Yuki Kato, Kazuyuki Morioka, "Autonomous Robot Navigation System Without Grid Maps Based on Double Deep Q-Network and RTK-GNSS Localization in Outdoor Environments", Proc. of IEEE/SICE SII, pp.346-351, 2019

[2] Yuki Shibata, Kazuyuki Morioka, "Development of mobile robot network based on data sharing in the Ethereum platform", Proc. of IEEE/SICE SII, 2023

[3] J. Ding et al., "Monocular Camera-Based Complex Obstacle Avoidance via Efficient Deep Reinforcement Learning", IEEE Transactions on Circuits and Systems for Video Technology, Vol.33, No.2, pp.756-770, 2023

[4] Ryuto Tsuruta, Kazuyuki Morioka, "Autonomous Navigation of a Mobile Robot with a Monocular Camera Using Deep Reinforcement Learning and Semantic Image Segmentation", Proc. of IEEE/SICE SII, pp.1107-1112, 2024

[5] Y. Zhu et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning,", Proc. of IEEE ICRA 2017, pp. 3357-3364, 2017

[6] Kanako Amano, Anna Komori, Saki Nakazawa, Yuka Kato, "Adaptive Navigation Method for Mobile Robots in Various Environments using Multiple Control Policies", Proc. of IEEE INDIN, 2023

[7] Naoki Yokoyama, Alex Clegg, Joanne Truong, Eric Undersander, Tsung-Yen Yang, Sergio Arnaud, Sehoon Ha, Dhruv Batra, Akshara Rai. "ASC: Adaptive Skill Coordination for Robotic Mobile Manipulation", IEEE Robotics and Automation Letters, Vol.9, No.1, pp.779-786, 2024

[8] G. Raja, K. Kottursamy, K. Dev, R. Narayanan, A. Raja, K. Karthik, "Blockchain-Integrated Multiagent Deep Reinforcement Learning for Securing Cooperative Adaptive Cruise Control", IEEE Transactions on Intelligent Transportation Systems, Vol.23, No.7, pp.9630–9639, 2022

[9] Isaac J. Jensen, Daisy Flora Selvaraj, Prakash Ranganathan, "Blockchain Technology for Networked Swarms of Unmanned Aerial Vehicles (UAVs)", Proc. of IEEE WoWMoM, 2019

[10] Jonathan Grey, Oshani Seneviratne, Isuru Godage, "Blockchain-Based Mechanism for Robotic Cooperation Through Incentives: Prototype Application in Warehouse Automation", Proc. of IEEE Blockchain, pp.597-604, 2021

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, "Proximal policy optimization algorithms", arXiv preprint, 1707.06347, 2017

[12] Muhammad Nasir Mumtaz Bhutta, Amir A. Khwaja, Adnan Nadeem, Hafiz Farooq Ahmad, Muhammad Khurram Khan, Moataz A. Hanif, Houbing Song, Majed Alshamari, Yue Cao, "A Survey on Blockchain Technology: Evolution, Architecture and Security", IEEE Access, Vol.9, pp.61048-61073, 2021

[13] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System", arXiv preprint, 1407.3561, 2014

[14] "https://assetstore.unity.com/packages/3d/environments/urban/real-new-york-city-vol-2-222827"