Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd

ISSN,e-ISSN: 2548-1304
Volume:IDAP-2023, pp:191-197, 2023
https://doi.org/10.53070/bbd.1377537
Research Paper

Blockchain Powered Task Management for Robots in Healthcare: Webots Simulation

Mehmed Oğuz ŞEN[*1], Fatih OKUMUŞ[2], Adnan Fatih KOCAMAZ[1]

[1] Bilgisayar Mühendisliği Bölümü, İnönü Üniversitesi, Malatya, Türkiye

[2] Yazılım Mühendisliği Bölümü, İnönü Üniversitesi, Malatya, Türkiye

(oguz.sen@inonu.edu.tr, fatih.okumus@inonu.edu.tr, fatih.kocamaz@inonu.edu.tr)

Received:Oct.17,2023 Accepted:Oct.17,2023 Published:Oct.18,2023

Abstract— In recent years, the adoption of robotic systems has witnessed remarkable growth across diverse industries, revolutionizing various aspects of automation and human-robot interaction. Healthcare is one of the major professions where replacement of human labor with well-functioning service robots would be heavily desired. Mobile service robots in a multi robot system can work collaborately and cooperatively to assist medical personnel by performing work routines as simple or complex tasks. As these robots carry out complex tasks, there arises a crucial need for secure, efficient, and immutable storage mechanisms to manage the vast amount of task-related information generated during their operation. In this study, we propose a simulation where a system of multiple TIAGo++ robots can request and complete simple tasks while using Hyperledger Fabric (HLF) blockchain platform to address the requirements of storing and accessing task information. Communication with the blockchain network and controlling it in Webots environment for a single TIAGo++ robot are integrated into same Java application in this simulation. Experimental results show that HLF and Webots can successfully run concurrently for task allocation and to store data acquired from mobile robots in a multi robot simulation.

Keywords : *Hyperledger Fabric, blockchain, health robots, decentralized management, Webots, TIAGo++.*

1. Introduction

In the aftermath of COVID-19 outbreak it was well experienced that deployment of service robots in healthcare would have been playing a critical role not only in isolation of infected patients and medical personnel but also in diminishing the number of possible infections. With the rapid advancements in robotics and artificial intelligence, mobile service robots have emerged as integral tools in various industries, transforming traditional workflows and offering novel solutions to complex challenges. As an ever improving sub branch of robotics, mobile robots and their behavioral planning stands as a hot research area in robotics. In recent years, healthcare has emerged as a promising domain for integrating mobile robots to enhance patient care, improve medical procedures, and alleviate the burden on healthcare professionals. As an alternative to the cutting-edge healthcare robots, the TIAGo++ robot, with its human-like appearance and versatile capabilities, can be utilized in revolutionizing patient assistance, rehabilitation, and healthcare services. By leveraging its sensing, perception, and mobility capabilities; it can perform a range of tasks, such as patient monitoring, medication and other essential delivery, physical therapy assistance, and data collection, with precision and efficiency.

TIAGo++, developed by PAL Robotics, is introduced as a new version of TIAGo and is a ROS compliant robot. With its 7-DoF arms to perform coordinated dual-arm actions and expandable base structure, TIAGo++ can carry out complex tasks with enhanced dexterity and adaptability. Since obtaining several TIAGo++ robots can be quite costly, experiments with multiple TIAGo++ robots are often conducted in simulated environments such as Webots, where it can be tested and fine-tuned in a controlled and safe setting.

As the tasks to be executed by the TIAGo++ robot become increasingly intricate and the volume of data generated during its operations continues to surge, managing tasks and storing task-related information poses a significant challenge. In traditional robotic applications, task data management has been accomplished using conventional centralized databases, leading to concerns about data integrity, security, and the risk of single points of failure. To overcome these limitations, there is a compelling need for a robust, transparent, and decentralized approach to store and manage task information efficiently.

In recent years, blockchain technology has gained considerable attention as a potential solution for addressing data management challenges across various domains. The underlying principles of immutability, transparency, and decentralized consensus make blockchain an attractive candidate for enhancing data integrity and security. Hyperledger Fabric, a leading open-source blockchain framework, offers a unique set of features ideally suited for enterprise-grade applications, making it an appealing choice for use in robotics.

Hyperledger Fabric (HLF) is an enterprise-grade, permissioned blockchain framework that operates as an open-source project under the Linux Foundation's Hyperledger initiative. Designed to cater to the unique requirements of business applications, HLF stands out as a prominent blockchain platform within the industry. One of its key advantages lies in its modular architecture, which permits pluggable consensus protocols, security modules and membership services, thereby fostering flexibility and scalability (Androulaki et al., 2018). In HLF, peers of the blockchain network can communicate in a channel-based system, which provides distinct private channels for secure and confidential transactions between certain peers and organizations. Moreover, HLF utilizes an endorsement policy framework which enables fine-grained control over transaction validation, enhancing governance and regulatory compliance. Its support for chaincode development (terminology used for smart contracts in HLF) in various programming languages provides versatility to decentralized application developers (Hewa et al., 2021). Its enterprise-level security mechanisms, such as access controls and identity management, assures heightened protection against malicious activities such as Sybil attacks.

In this study, we present an application to address the task information storage needs in a system of multiple TIAGo++ robots operating within the Webots simulation environment. Leveraging the capabilities of HLF, we aim to achieve an immutable and decentralized repository for task data, facilitating efficient information sharing, retrieval, and validation in a secure and transparent manner. By adopting a blockchain-based solution, we introduce a Webots simulation as an extension of our previous study on using blockchain powered task distribution for robots in healthcare for enhanced collaboration and reliable operation in a system of multiple mobile robots.

The primary objectives of this study are as follows:

1. To investigate the challenges associated with conventional data storage mechanisms for the Tiago++ robot's tasks in Webots and identify their limitations in terms of scalability, integrity, and security.

2. To explore the potential benefits of employing blockchain technology, particularly Hyperledger Fabric, as a robust and decentralized data management system for the TIAGo++ robot's task information.

3. To design and implement a proof-of-concept simulation system that integrates TIAGo++ as a ROS compliant robot in Webots with HLF.

4. To demonstrate the proper execution of task management in a system of multiple Tiago++ robots in a simulation environment.

The remainder of this article is organized as follows: Section 2 presents a review of related work about applications of blockchain powered systems of mobile robots. In Section 3, we delve into the methodology and architectural design of our proposed system, discuss the implementation details and evaluate the experiments in the simulation platform. Finally in Section 4, we conclude the study, highlighting the contributions and outlining future research directions.

2. Related Work

Systems of multiple mobile robots is an ongoing hot research field where applications on several research domains like indoor logistics (Okumuş, Dönmez and Kocamaz, 2020), search and rescue (Dadgar, Couceiro and Hamzeh, 2020), molecular robotics (Trotta et al. 2020) (Kabir, Inoue and Kakugo, 2020) (Keya et al. 2018) and health (Holland et al. 2021) (Farkh et al. 2021) (Fang 2021) have been proposed. Among these research domains, healthcare related studies gained much attention and popularity since the beginning of ongoing COVID-19 pandemic. Common applications of mobile robots in healthcare include medical examination, patient surveillance (Holland et al. 2021) (Farkh et al. 2021) and medical transportation (Fang 2021) but these applications mostly deal with single robots. Multi robot systems can provide an efficient way to perform healthcare services fluently.

Specific adaptations of blockchain systems to multi robot systems for multi robot collaboration in fighting COVID-19 pandemic are stated in (Alsamhi and Lee, 2021) and (Alsamhi et al. 2021), but an application for general healthcare services hasn't been proposed yet. In this study we implement a simulation to demonstrate a proof-of-concept for our study (Şen, Okumuş and Kocamaz. 2022) which introduces and describes a system of multiple healthcare robots using a permissioned blockchain platform for decentralized management with use case scenarios.

3. Methodology

3.1. Technical Background:

3.1.1. Webots:

Webots is a widely adopted and versatile robot simulation software that gained popularity among robotics researchers and developers. As a cross-platform and open-source tool, Webots provides researchers and engineers with a comprehensive environment for designing, simulating, and evaluating various robotic systems, thereby facilitating both virtual prototyping and algorithm validation (Michel, 2004). Notably, Webots supports a diverse range of robot models, sensors, actuators, and environments, enabling users to emulate complex real-world scenarios accurately. Moreover, Webots possesses a high degree of extensibility through its scripting and plugin capabilities, thereby fostering the integration of external libraries and frameworks. This flexibility allows for the seamless adaptation of Webots to specific research objectives and experimental requirements.

3.1.2. Hyperledger Fabric:

As described above HLF gives users to setup a permissioned blockchain network which can be tailored for custom requirements. In the process of setting up a HLF network; various types of nodes, including peers, ordering service nodes, and clients are deployed by organizations. Chaincodes are implemented to encapsulate business logic and are installed on peers in order to commit transactions. Transactions are initiated by clients and undergo a series of steps. Firstly, transactions are proposed to endorsing peers, which simulate the transaction's impact on the ledger using the associated chaincode. Endorsement policies determine the required peer endorsements. Subsequently, endorsed transactions are sent to ordering nodes for consensus and sequencing into blocks. Ordering service nodes broadcast these blocks to endorsing peers for validation against endorsement policies and ledger consistency. The endorsed and validated transactions are then committed to the shared ledger, reflecting the agreed-upon state changes. In our case, this process is simplified by using default endorsement policies since defining complicated policies are not required for this simulation.

3.2. Setup of Simulation Environment:

Our system consists two main components, first one is the HLF blockchain network, where each robot is also a peer node, running task and robot chaincodes and second one is the Webots simulation environment and HLF Java gateway appplications (as separate Java processes) attached to it as external controllers. We launched a HLF network with 5 peer and 3 orderer nodes without using Docker Compose. In order to simulate a system of 5 TIAGo++ robots, we created a Webots world with 5 TIAGO++ PROTO nodes using external controllers. Since Webots provides a Java APl for programming robot controllers, we were able to implement the code for both connecting to HLF gateway and controlling every TIAGo++ robot externally in the same Java project. Therefore we managed to integrate Webots and HLF into a single Java application. Multiple instances of this Java application was needed for each TIAGo++ robot, therefore we opened separate IntelliJ IDEA IDE instances for it.

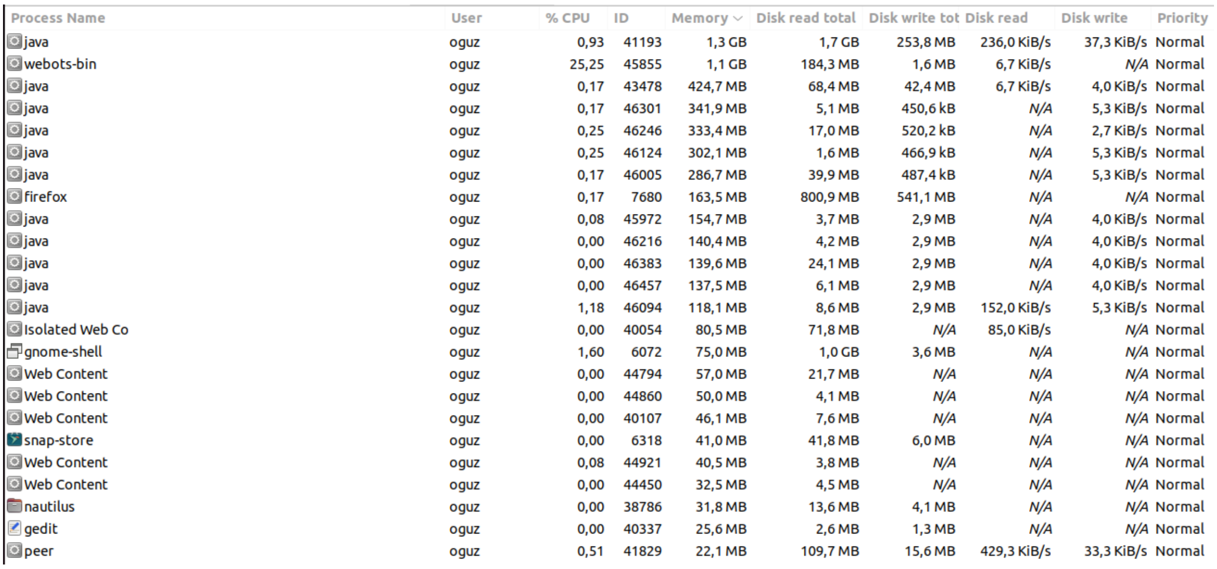
For the sake of simplicity, the task for a robot is defined such as to reach a goal position. When the simulation starts, each robot obtains available tasks in the task ledger and requests the task with least cost for it. If this request is accepted, the robot starts the task and ignores other available tasks until its task is completed. After completing a task, the robot sends a request to the blockchain network to update ledger data and checks available tasks to select a new one. If no task is available, the robot stays idle until new tasks are generated.

We experienced problems which required technical matters to overcome during the implementation of the simulation, therefore in order to execute our simulation properly these technical issues should be considered.

1. Although Webots supports controllers written in Python; Java was preferred to implement the source code which integrates HLF Gateway SDK and Webots API, since no official HLF Python SDK is released by the time of this study. This choice yields the Webots Java controllers to run in JVM processes which consumes significant amount of memory. In a case running 10 Java controllers costs up to 4 GB of RAM which can be seen in Figure 1.

2. Setting Webots to use GPU as graphics renderer is essential, since Webots process can crash suddenly while running the simulation when software graphics renderer with CPU is selected. This happens occasionally with Linux versions (e.g. Ubuntu 22.04) of Webots.

3. Using external controllers in Webots gives the flexibility both to connect to HLF network and Webots in the same Java process. While using the configurations described in the official tutorial, we faced a runtime error about missing Java libraries although correct path to Webots Java controller package was included as an environment variable and we solved this problem by adding the path to Webots C++ library to the same environment variable.



Process Name	User	% CPU	ID	Memory	Disk read total	Disk write tot	Disk read	Disk write	Priority
java	oguz	0,93	41193	1,3 GB	1,7 GB	253,8 MB	236,0 KiB/s	37,3 KiB/s	Normal
webots-bin	oguz	25,25	45855	1,1 GB	184,3 MB	1,6 MB	6,7 KiB/s	N/A	Normal
java	oguz	0,17	43478	424,7 MB	68,4 MB	42,4 MB	6,7 KiB/s	4,0 KiB/s	Normal
java	oguz	0,17	46301	341,9 MB	5,1 MB	450,6 kB	N/A	5,3 KiB/s	Normal
java	oguz	0,25	46246	333,4 MB	17,0 MB	520,2 kB	N/A	2,7 KiB/s	Normal
java	oguz	0,25	46124	302,1 MB	1,6 MB	466,9 kB	N/A	5,3 KiB/s	Normal
java	oguz	0,17	46005	286,7 MB	39,9 MB	487,4 kB	N/A	5,3 KiB/s	Normal
firefox	oguz	0,17	7680	163,5 MB	800,9 MB	541,1 MB	N/A	N/A	Normal
java	oguz	0,08	45972	154,7 MB	3,7 MB	2,9 MB	N/A	4,0 KiB/s	Normal
java	oguz	0,00	46216	140,4 MB	4,2 MB	2,9 MB	N/A	4,0 KiB/s	Normal
java	oguz	0,00	46383	139,6 MB	24,1 MB	2,9 MB	N/A	4,0 KiB/s	Normal
java	oguz	0,00	46457	137,5 MB	6,1 MB	2,9 MB	N/A	4,0 KiB/s	Normal
java	oguz	1,18	46094	118,1 MB	8,6 MB	2,9 MB	152,0 KiB/s	5,3 KiB/s	Normal
Isolated Web Co	oguz	0,00	40054	80,5 MB	71,8 MB	N/A	85,0 KiB/s	N/A	Normal
gnome-shell	oguz	1,60	6072	75,0 MB	1,0 GB	3,6 MB	N/A	N/A	Normal
Web Content	oguz	0,00	44794	57,0 MB	21,7 MB	N/A	N/A	N/A	Normal
Web Content	oguz	0,00	44860	50,0 MB	4,1 MB	N/A	N/A	N/A	Normal
Web Content	oguz	0,00	40107	46,1 MB	7,6 MB	N/A	N/A	N/A	Normal
snap-store	oguz	0,00	6318	41,0 MB	41,8 MB	6,0 MB	N/A	N/A	Normal
Web Content	oguz	0,08	44921	40,5 MB	3,8 MB	N/A	N/A	N/A	Normal
Web Content	oguz	0,00	44450	32,5 MB	4,5 MB	N/A	N/A	N/A	Normal
nautilus	oguz	0,00	38786	31,8 MB	13,6 MB	4,1 MB	N/A	N/A	Normal
gedit	oguz	0,00	40337	25,6 MB	2,6 MB	1,3 MB	N/A	N/A	Normal
peer	oguz	0,51	41829	22,1 MB	109,7 MB	15,6 MB	429,3 KiB/s	33,3 KiB/s	Normal

Figure 1. Memory consumption of Java processes for controlling TIAGo++ robots in Webots.

3.3. Application:

We executed our simulation on a PC with Intel Core i5 CPU with 4 cores running at 2.3 GHz, NVIDIA GeForce 920MX GPU and 8 GB of RAM. More robots can also be tested on a computer system with enough memory but we decided to run a system of 5 TIAGo++ robots under this memory constraint. We launched our HLF network in separate terminals and installed robot and task chaincodes. After successful launch of HLF network; the task ledger is populated with initial tasks shown in Table 1, since robots are expected to select a task from the list of available tasks. Next, we started Webots and ran the Java applications described in Section 3.2. An overview of the Webots simulation world and 5 TIAGo++ robots at initial positions is shown in Figure 2.

Table 1. List of initially generated tasks on the task ledger. Since all tasks have its initial value "PENDING", status of the tasks are omitted.

Task ID	Goal Position
1	(2, 2)
2	(1, 4)
3	(-2, -2)
4	(4, 3)
5	(-1, -3)
6	(3, -2)
7	(-2, 3)
8	(4, -4)
9	(5, -1)
10	(-5, 1)

TIAGo++ PROTO in Webots does not include sensor suite for navigation, so that GPS and compass modules in Webots is added as components to each robot in the simulation. Since precise localization and mapping is beyond the scope of this study, we assumed navigation data acquired from GPS and compass modules in Webots would be sufficient.

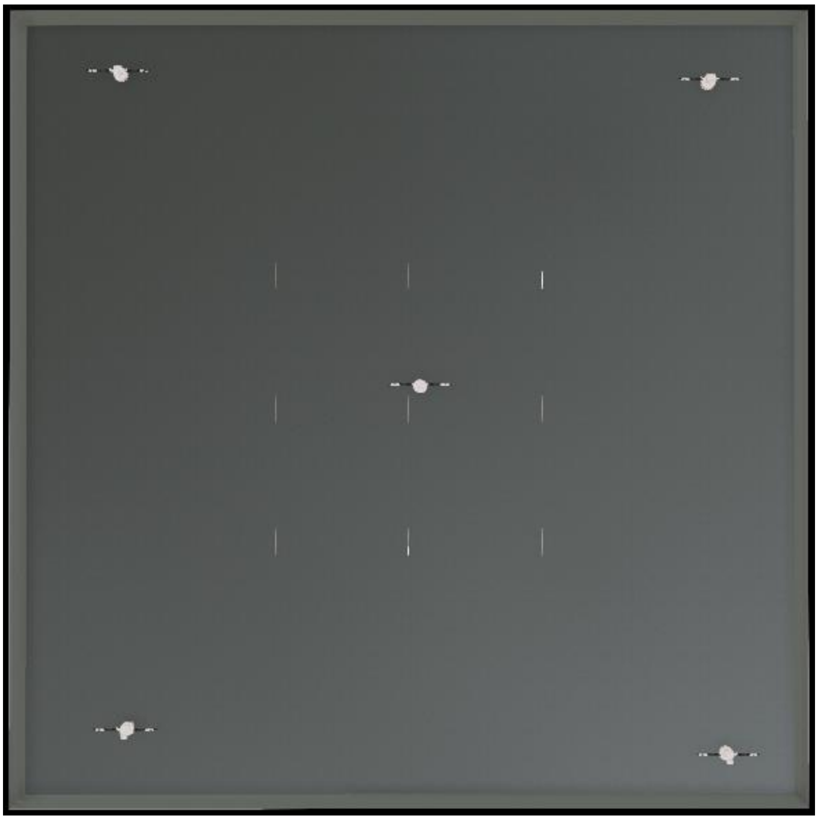


Figure 2. Initial positions of 5 TIAGo++ robots in Webots simulation

3.4. Experimental Results:

We tested our simulation with 10 tasks and 5 robots. After all tasks are completed, task data acquired from the robots are shown in Table 2. Final positions of the robots are obtained from GPS modules attached to them and hence can contain sensor noise and errors defined in Webots simulation. However, this study does not focus on precise positioning as we stated in Section 3.3 therefore values obtained from simulated sensors are used to check for task completion.

Table 2. Status of the robots after all tasks are completed.

Robot ID	Assigned Tasks	Final Position
Robot 1	2, 6	(-4.11, 2.92)
Robot 2	4, 10	(6.08, -3.95)
Robot 3	5, 9	(-9.91, 5.08)
Robot 4	1, 7	(4.99, 7.01)
Robot 5	3, 8	(-9.94, -4.92)

Experimental results we stated above shows that a system of multiple TIAGo++ robots in Webots simulation successfully interacts with the HLF blockchain network for efficient task allocation and storing task data. This simulation can be extended with additional task generations and more robots, it is going to run independent from the total number of robots and tasks as more computer system resources are available. We actually tried and managed to run 10 TIAGo++ robots at the same time but had to disable real time graphics rendering in Webots, because of the system constraints described in Section 3.3.

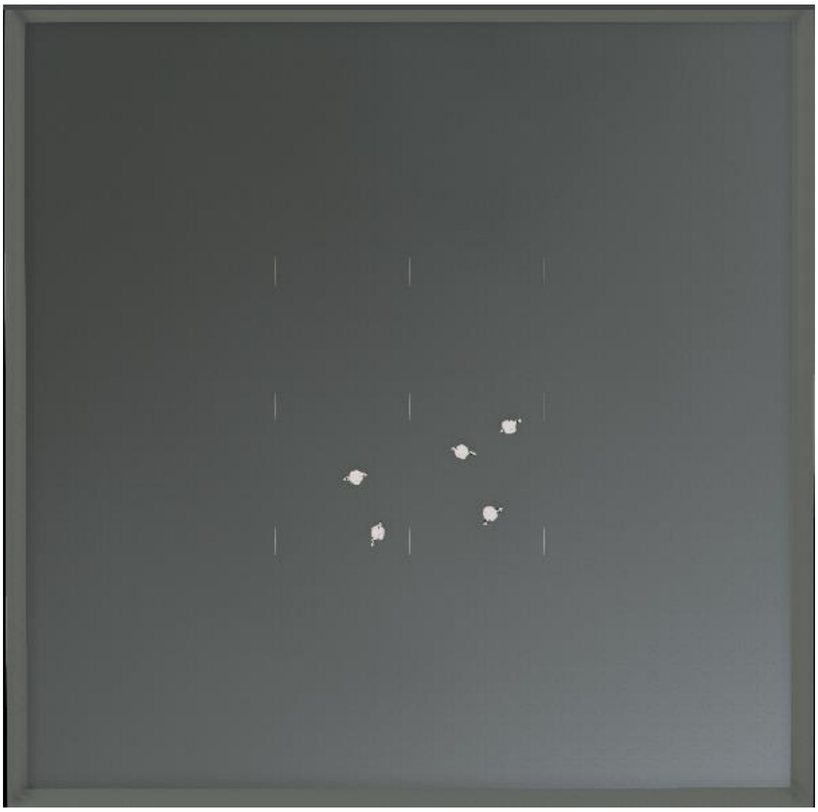


Figure 3. Final positions of 5 TIAGo++ robots after all tasks are completed in Webots simulation

4. Discussion and Future Work

In this study we proposed a simulation in Webots where a system of multiple TIAGo++ robots can select and execute tasks which are distributed among a HLF blockchain network. Each robot selected most appropriate task for it among the tasks previously generated and stored in task ledger. After stating related studies in the literature, we gave the implementation steps of the proposed system and indicated technical aspects related with performance which can be critical for running the simulation. Next, we presented the execution of the simulation in Webots with 5 TIAGo++ robots. After completion of tasks by robots, task ledger is updated with final task data gathered from them. Successful execution of the simulation demonstrates a proof-of-concept for our previous study (Şen, Okumuş and Kocamaz, 2022) and shows that real TIAGo++ robots can be used for healthcare duties in real world scenarios.

Although TIAGo++ is a ROS enabled robot, ROS features are not integrated in the application and Webots Java API is used for controlling TIAGo++ instead. We are planning to extend the scope of this study by fully integrating ROS into the simulation. Introducing message transmission among robots as an enhancement of task management and distribution would also be an improvement of the simulation we propose as future work.

Acknowledgements

While preparing this study, support was provided from Malatya İnönü University Scientific Research Projects (BAP) fund with the project code ID 3136 FBG-2023-3136.

References

Androulaki, E. et al (2018) Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the Thirteenth EuroSys Conference (EuroSys '18), Porto, Portugal. https://doi.org/10.1145/3190508.3190538

Michel O. Cyberbotics Ltd. WebotsTM: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems. 2004;1(1). doi:10.5772/5618

T. M. Hewa, Y. Hu, M. Liyanage, S. S. Kanhare and M. Ylianttila, "Survey on Blockchain-Based Smart Contracts: Technical Aspects and Future Research," in IEEE Access, vol. 9, pp. 87643-87662, 2021, doi: 10.1109/ACCESS.2021.3068178.

TIAGo++, the bi-manual robot with two arms for your research! (2023) https://blog.pal-robotics.com/tiago-bi-manual-robot-research. Accessed 01 Aug 2023

Şen M. O, Okumuş F, Kocamaz A. F. (2022) Application of Blockchain Powered Mobile Robots In Healthcare: Use Cases, Research Challenges and Future Trends.

Okumuş F, Dönmez E, Kocamaz AF. A cloudware architecture for collaboration of multiple agvs in indoor logistics: Case study in fabric manufacturing enterprises. Electronics. 2020;9(12): 2023.

Dadgar M, Couceiro MS, Hamzeh A. RbRDPSO: Repulsion-Based RDPSO for Robotic Target Searching. Iranian Journal of Science and Technology - Transactions of Electrical Engineering. 2020;44(1): 551–563.

Trotta A, Montecchiari L, Felice MD, Bononi L. A GPS-Free Flocking Model for Aerial Mesh Deployments in Disaster-Recovery Scenarios. IEEE Access. 2020;8: 91558–91573.

Kabir AMR, Inoue D, Kakugo A. Molecular swarm robots: recent progress and future challenges. Science and Technology of Advanced Materials. 2020;21(1): 323–332.

Keya JJ, Kabir AMR, Inoue D, Sada K, Hess H, Kuzuya A, Kakugo A. Control of swarming of molecular robots. Scientific Reports. 2018;8(1): 1–10.

Holland J, Kingston L, McCarthy C, Armstrong E, O'Dwyer P, Merz F, et al. Service Robots in the Healthcare Sector. Robotics. 2021;10(1): 47.

Farkh R, Marouani H, Al Jaloud K, Alhuwaimel S, Quasim MT, Fouad Y. Intelligent autonomous-robot control for medical applications. Computers, Materials and Continua. 2021;68(2): 2189-2203.

Fang B, Mei G, Yuan X, Wang L, Wang Z, Wang J. Visual SLAM for robot navigation in healthcare facility. Pattern Recognition. 2021;113: 107822.

Alsamhi SH, Lee B. Blockchain-Empowered Multi-Robot Collaboration to Fight COVID-19 and Future Pandemics. IEEE Access. 2021;9: 44173–44197.

Alsamhi SH, Lee B, Guizani M, Kumar N, Qiao Y, Liu X. Blockchain for decentralized multi-drone to combat COVID-19 and future pandemics: Framework and proposed solutions. Transactions on Emerging Telecommunications Technologies. 2021;32: e4255.