UWB Role Allocation with Distributed Ledger Technologies for Scalable Relative Localization in Multi-Robot Systems

Paola Torrico Morón†, Salma Salimi†, Jorge Peña Queraltat, Tomi Westerlund†

†Turku Intelligent Embedded and Robotic Systems (TIERS) Lab, University of Turku, Finland.

Emails: †{ptcorno, salmas, jopequ, tovewe}@utu.fi

Abstract—Systems for relative localization in multi-robot systems based on ultra-wideband (UWB) ranging have recently emerged as robust solutions for GNSS-denied environments. Scalability remains one of the key challenges, particularly in ad-hoc deployments. Recent solutions include dynamic allocation of active and passive localization modes for different robots or nodes in the system, with larger-scale systems becoming more distributed. Key research questions arise in the areas of security and trustability of such localization systems. This paper studies the potential integration of collaborative-decision making processes with distributed ledger technologies. Specifically, we investigate the design and implementation of a methodology for running an UWB role allocation algorithm within smart contracts in a blockchain. In previous works, we have separately studied the integration of ROS2 with the Hyperledger Fabric blockchain, and introduced a new algorithm for scalable UWB-based localization. In this paper, we extend these works by (i) running experiments with larger number of mobile robots switching between different spatial configurations and (ii) integrating the dynamic UWB role allocation algorithm into Fabric smart contracts for distributed decision-making in a system of mobile mobile robots. This enables us to deliver the same functionality within a secure and trustable process, with enhanced identity and data access management. Our results show the effectiveness of the UWB role allocation for continuously varying spatial formations of six autonomous mobile robots, while demonstrating a low impact on latency and computational resources of adding the blockchain layer that does not affect the localization process.

Index Terms—Ultra-wideband (UWB); Localization; Mobile robots; Distributed ledger technologies; Blockchain; Hyperledger Fabric; Scalable positioning; Cooperative localization

I. INTRODUCTION

Autonomous mobile robots have been penetrating multiple industries and domains of our society. At the same time, their connectivity has been increasing and networked collaborative systems have gained importance within the field [1]. From construction [2] and industrial warehouses [3] to mining, the ability for robots to operate without a user input has become more important [4]. In order to achieve a high level of autonomy and situational awareness, localization is one of the first problems to be solved [5]. Outdoors, GNSS sensors are widely used [6], but multipath propagation and other inherent limitations make this solution unreliable in indoors locations [7], [8]. Ultra-wideband (UWB)-based localization has emerged as a robust solution for localization of mobile robots in GNSS-denied environments [9], [10]. Positioning methods based on UWB have potential also for relative localization and state estimation in multi-robot systems [11], [12], [13]. However, most systems rely on active UWB ranging for localization, which presents inherent scalability limitations owing to the required time allocation of the ranging messages. In a recent work, we have presented an approach to scalable localization in mobile multi-robot systems through dynamic role allocation, combining both active and passive positioning [14]. In this paper, our objective is to design and implement a methodology for integrating the UWB role allocation algorithm as a secure and distributed collaborative decision-making problem with blockchain technology and smart contracts.

The increasing research interest towards robot swarms and multi-robot systems has brought an increasing awareness in areas such as security of robotic systems [15], [16], or the explainability and auditability of their behaviour [17]. At the same time, distributed ledger technologies (DLTs) are penetrating the design and development of networked systems and the next-generation internet. At the intersection of these two trends, there is significant potential for the design of more secure and trustable distributed robotic systems, from swarms of robots [18] to autonomous vehicles [19]. Surveillance or industrial application areas are an area of particular potential [20].

In summary, this manuscript delves into the integration of the Hyperledger Fabric Blockchain and smart contracts
with distributed role allocation methods for cooperative UWB-based localization in multi-robot systems. The main contributions of this work are:

i. the design and implementation of smart contracts for managing dynamic UWB localization roles in multi-robot systems, and

ii. the deployment and experimentation of the proposed solution, extending the experimental setup of the previous work where the scalable UWB role allocation algorithm is introduced to a larger number of mobile robots [14].

The remainder of the paper is structured as follows. Section II introduces related works in scalable UWB positioning in addition to spatial coordination approaches incorporating UWB-based localization and DLTs for decision-making. Section III then presents the system architecture, with the implementation details given in Section IV. Section V reports experimental results while Section VI concludes the work and lists future research directions.

II. BACKGROUND AND RELATED WORKS

Localization in mobile robotics utilizing radio-based technology is not a new approach. A recent trend, however, is the adoption of UWB-based solutions, specially for GNSS-denied environments [9], driven by unprecedented accuracy in the field of radio-based indoor positioning at low cost. UWB provides a higher accuracy, along with lower interference in the bands it operates and better robustness against multipath when comparing with Wi-Fi or Bluetooth, allowing for a wider adoption [8].

Autonomous mobile robots are part of multiple industries such as warehouse management [21], mining [22], packet deliveries. All these applications might not be able to provide controlled environments for the robots to operate in. Approaches based on light based sensors like cameras, LiDARS and VIO cameras depend on the quality of the environment, being susceptible to dust and low level visibility conditions [10]. Wireless technologies such as UWB are not influenced by those conditions. In addition, onboard odometry approaches using LiDar and VIO cameras might drift over time, and integrating them with UWB measurements aid it reducing and correcting long-term drift [10], [23].

In the case of UAVs, the centimeter-level accuracy that can be provided by UWB makes a cost effective replacement for a MOCAP system [24], [25], and has also the advantage that it does not need to be tied to such as a constraint environment as MOCAP systems usually are. Typical applications for UWB already in use for UAVs include ground to air ranging [26] and level of swarm-wide state estimation with a UWB in each robot [11].

A. UWB localization

UWB localization can be performed utilizing different ranging modalities, which are common to wireless ranging systems [10]. Two widely use rangings are Time of Flight (ToF) and Time Difference of Arrival (TDoA) also known as hyperbolic positioning [27]. Both are widely used in both research and industry. Angle of arrival (AoA), can also be used though it is more rare [28]. Received Signal Strength Indicator (RSSI) is commonly used in Bluetooth technology [8], [28]. However, UWB localization systems are able to produce, out-of-the-box, centimeter-level accuracy while RSSI is more dependent on the environment and is typically at least an order of magnitude less accurate.

Time of flight ranging measures the amount of time it takes for the signal to propagate from the initiator, that start the transmission, to the receiver [7]. ToF measurements are often performed one to one between two devices, in a process called Two Way Ranging (TWR). It can either consist on two transmission, or single sided two-way ranging (SS-TWR), or three, with a double sided two-way ranging (DS-TWR) [10]. In SS-TWR, a device A initiates the transmission, having device B receive it. Then B responds to the polling message, returning the message after an internal delay for processing. When device A receives the response message, it is able to compute the time of flight of the signal. On the other hand, with DS-TWR, the devices take turns initiating the communication, making it equivalent to two SD-TWR ranging [9]. Given the time of flight of the signal and knowing the speed of the signal in the transmission medium, the distance between devices A and B is obtained. To obtain the position of a UWB node using ToF ranging, the distance to at least three nodes is needed [29]: once the distances are obtained, with anchors normally fixed in place, the actual position of the node can be calculated using trilateration or multilateration.

Time Difference of Arrival measures the difference of propagation time between the transmission point of a UWB signal and two or more receivers [9]. Then it is possible to obtain the difference of distance between the transmitter and each of the receivers. In passive TDoA scheme, a number of UWB nodes engage in constant communication and a passive UWB just receives all the messages, enabling the computation of all the differences of distances to pairs of active nodes. From these, the position can be calculated at the intersection of a series of hyperbolas.

When deploying a system in larger areas for a larger number of nodes, the scalability of the system can be considered as a key parameter [10]. The number of nodes which are actively communicating will limit the communication frequency [30], which affects the localization frequency. With ToF, the more nodes on the system the lower the frequency [31]. Passive TDoA does not present this problem, since part of the nodes just have to receive and not transmit any message [32]. However, the main drawback is the requirement of clock synchronization [33] and the positioning error increases when leaving the convex envelope defined by the active nodes [34].

B. Hyperledger Fabric

In distributed multi-robot systems, the implementation of blockchain technology can bring some key properties, such as a built-in security while data sharing in the network [35], enabling auditability through the immutability of the data [36], and collaborative decision-making through smart contracts
enabled by consensus protocols [37], [38]. Blockchain platforms handle user credentials differently, creating with both permissionless and permissioned solutions [39].

Hyperledger Fabric is a private permissioned blockchain developed in Go, Java and Javascript [40]. It is able to achieve data isolation by enabling different channels of communication between subsets of peers, allowing for private communication [17], [41]. Hyperledger fabric provides high modularity, configurability and pluggable features, that enable developers to implement different technologies, such as consensus protocols [42].

Having being designed for enterprise use since the beginning, Hyperledger Fabric has a set of characteristics that can be implemented in distributed robotic systems. All the participants in the blockchain are identified, which enables identity and data access management. Fabric contains tools for certificate generation and identity management. The network is permissioned, adding a layer of security to the system. It is also capable of meeting the needs of real time robotic system due to its high transaction throughput performance. Transaction confirmation can be configured so consensus can happen with low-latency but in a non deterministic manner, depending on connectivity [40].

III. SYSTEM OVERVIEW

Through this section we describe the system overview. It is divided in two main sections, the UWB part for the relative localization of the robots on the system and the implementation of smart contract in the Hyperledger Fabric for the positions and roles in the system.

A. UWB Relative Localization

As mentioned before, UWB technology can be used for localization with centimeter level accuracy. It has also several ranging methods, two of the most used being ToF and TDoA. ToF localization is based on the exchange of messages between a pair of nodes, where the distance between the nodes can be measured based on the time of flight of the message. With a minimum of three distances (towards anchors), multilateration can be performed and relative localization with reference to the anchors can be obtained. The disadvantage here is that the position of some of the nodes has to be known, and more importantly, there are a high number of messages needed to be exchanged. With each new node in the system decreasing the frequency of ranging for all the UWB nodes.

In case of TDoA, and more specifically in passive TDoA, less messages can be exchanged. Normally fixed nodes transmit ranging between them, in our particular case message for TWR, and passive nodes are able to listen to those. This time instead of computing the distance, it is the difference of distance to a pair of nodes, so TDoA is a more scalable solution than ToF, because adding more nodes does not necessarily decrease the localization frequency of the whole system. The disadvantage however, is that the localization accuracy decreases outside the convex envelope of the fixed anchors. If we consider that most or all of the nodes of the system are mobile, the probability of some passive node leaving the convex envelope is very high.

With the aim of using the strength of both ToF and TDoA, and minimize their disadvantages, we proposed a combined solution in a previous work [14]. A subset of nodes, active nodes, use ToF for constant two way ranging and localization. A second subset of nodes, in contrast, use TDoA, receiving all the messages from the active nodes and using them for their own localization. These nodes are thus called listener nodes. In addition, the listener nodes need to be placed towards the center of the convex envelope created by the active nodes to improve accuracy. Assuming all nodes are mobile, the role of the nodes will therefore change based on the position of the nodes on the system.

B. Role allocation

The algorithm for role allocation that we integrate within a smart contract in Hyperledger Fabric is shown in Algorithm 1 (please see our previous work [14] for reference on the notation). The core idea behind this role allocation process is to keep listener nodes towards the centroid of the system’s convex envelope, and active nodes towards the outside of the system. The implemented smart contract analyzes all the possible combinations for a fixed number of nodes and a given number of active nodes. The number of active nodes is directly derived from the desired localization frequency, based on the ranging speed (maximum ranging frequency between two nodes $f_{UWB,max}$) and the minimum localization frequency ($f_{LOC,min}$): $k = \text{floor}(f_{UWB,max}/f_{LOC,min})$. From this, the centroid of all node combinations is computed and a minimization function is used to look up which combination of
Algorithm 1: UWB role allocation and relative positioning

Input:
Maximum number of active nodes: \( k \)
Node positions at \( t - 1 \): \( \{p_i[t - 1]\} \in \mathbb{R}^{3n} \);

Output:
Set of active nodes : \( A[t + 1] \in \mathbb{P}_k(n) \);
Set of passive listeners : \( L[t + 1] = [n \setminus A[t + 1]] \);
Active node positions : \( \{p_a[t]\}_{a \in A} \in \mathbb{R}^{3k} \);
Passive listener positions : \( \{p_p[t]\}_{L \in L[t]} \in \mathbb{R}^{2(n-k)} \);

Function tooa_ls_estimator \((A[t], \{p_i[t]\}) :\)
\[
\begin{align*}
\text{return} & \quad \argmin_{p \in \mathbb{R}^3} \sum_{i \in A[t], j \in A[t]} \left( d_{ij}^2 - \|p_j - p_i\| - \|p_j - p_l\| \right)^2; \\
\end{align*}
\]

Function localization_cost_f \((n, k, \{p_i[t]\}) :\)
\[
\begin{align*}
\text{return} & \quad \sum_{i \in A, j \in A \setminus j} \left( d_{ij}^2 - \|p_j - p_i\| - \|p_j - p_k\| \right)^2; \\
\end{align*}
\]
if \( t = 0 \) then
\[
\begin{align*}
\{d_{ij}\}_{j \in [n], i \neq j} & \leftarrow \text{get_tof_ranges}([n]); \\
\{p_i[0]\}_{i \in [n]} & \leftarrow \text{multilateration} \{d_{ij}\}; \\
\end{align*}
\]
else
\[
\begin{align*}
\{d_{ij}\}_{j \in A, i \neq j} & \leftarrow \text{get_tof_ranges} \(A[t]\); \\
\{d_{ij}\}_{L, j \in A, i \neq j} & \leftarrow \text{tdoa_ranges} \(L[t]\); \\
\end{align*}
\]
// Active node positions
\[
\{p_a[t]\}_{a \in A} \leftarrow \text{multilateration} \{d_{ij}\};
\]
// Passive listener positions
\[
\{p_p[t]\}_{L \in L[t]} \leftarrow \text{tdoa_ls_estimator} \{d_{ij}\};
\]
// Role allocation
\[
\begin{align*}
A[t + 1] & \leftarrow \argmin_{A \in \mathbb{P}_k(n)} \text{localization_cost_f} \(n, k, \{p_i[t]\}\); \\
L[t + 1] & \leftarrow [n \setminus A[t + 1] \\
\end{align*}
\]

nodes creates the least amount of distance error squared of the remaining nodes. The combination of nodes for the minimum are then the active nodes and the rest the passive nodes.

C. Hyperledger Fabric

The aim of using Hyperledger Fabric in this paper is for secure data management and robots control. On the other hand, smart contracts in Fabric blockchain have been used for storing path history and also role allocation of the robots.

Two smart contracts have been implemented in this framework: one for storing the path tracking of six Jetbots and one for implementing the role allocation algorithm and also storing the roles. The calculated roles are then published for the UWB nodes to switch roles accordingly.

IV. EXPERIMENTAL SETUP

For the experiment, we used six mobile robots and two additional fixed nodes. Each of the robots moved with a predefined trajectory and a series of waypoints to reach. The motion planning is implemented in a way that robots only navigate towards a waypoint when all of them have reached their current objective. The different waypoints were selected to enforce that the robots would need to switch between inner positions and outer ones, effectively requiring continuous reallocation of roles. This allows us to analyze different metrics on the role allocation process. All six robots are equipped with a DWM1001-Dev UWB node, while two extra static UWB nodes define the frame of reference for the relative localization. The path history and roles of the nodes is stored in Hyperledger fabric, to analyze the overhead that it adds to the system, and thus study the potential of this approach for enhanced security and trustability. The robots are deployed in an area of approximately \( 20 m^2 \).

A. Hardware

The platforms used in the experiment are six Jetbot robots from Waveshare. Each Jetbot incorporates a Jeson Nano Module with 4GB of RAM, 64-bit quad-core ARM A57 processor, and a Dual Band Wireless-AC 8265 for WiFi and Bluetooth connectivity. We program the Decawave’s DWM1001 UWB modules with a custom firmware to enable both ToF and TDoA localization, as well as scheduling of the ToF transmissions in time. The modules are able to act as active or passive nodes. Active nodes broadcast their own ranging or position estimations, while passive nodes do so through ROS topics. Therefore, all information is available to all nodes. While this requires slightly more communication, the overhead is mostly negligible and it allows for the position of all nodes to be known through the multi-robot system.

For ground reference a MOCAP system is used, consisting of six Optitrack cameras connected to the backbone Wifi network. We use as a controller for the entire system a computer with an Intel i7-10750H processor, to coordinate the actions of the six Jetbots, run the different location algorithms, as well as participate in the Hyperledger Fabric blockchain. The robots do not directly run the role allocation smart contract but instead only an interfacing ROS 2-Fabric application.

B. Software

The Jetbots run ROS Melodic under Ubuntu 18.04. The computer that serves as controller coordinates the waypoints for the Jetbots, publishing the next waypoint for each robot in ROS topics. The path controller, run directly in the Jetbot, subscribes to the topic and moves the it to it, waiting for the next one. The computer waits for all nodes to reach their next waypoint and then publishes a new set of objectives. The Decawave DWM1001 modules are connected to each Jetbot through a serial interface, and run custom firmware developed in C. Each Jetbot runs a python interface to publish the information provided by the UWB to different topics. The main controller can then either compute the position of all the nodes and also the roles directly, or use the ROS 1-ROS 2 bridge to pass the necessary information to the Fabric smart contract and other ROS 2 nodes.

We utilize the ros1_bridge to forward bidirectional data between the ROS 1 nodes that interface with the Jetbot motion planner, UWB nodes and global controller, and the ROS 2 nodes in Go that interface with the Fabric application. To simplify the setup, all topics are made available to both ROS 1 and ROS 2 nodes. In order to analyze the impact of the Fabric integration, we implement the role allocation algorithm in three different process: (i) a ROS 1 Python node; (ii) a ROS 2
TABLE I: The table shows the percentage of overlap in the roles when using the MOCAP localization vs the ToF ranging localization

|         | MOCAP [%] | ToF [%] | Total % |
|---------|-----------|---------|---------|
|         | Active    | Passive | Active  | Passive | Overlap |
| Node 3  | 25.00     | 75.00   | 27.50   | 72.50   | 94.17   |
| Node 4  | 61.25     | 38.75   | 55.00   | 45.00   | 87.92   |
| Node 5  | 0.83      | 99.17   | 36.67   | 63.33   | 64.17   |
| Node 6  | 49.71     | 50.83   | 60.42   | 39.58   | 82.08   |
| Node 7  | 25.83     | 74.17   | 24.17   | 75.83   | 93.33   |
| Node 8  | 37.92     | 62.08   | 33.33   | 66.67   | 95.42   |

Go node using `rclgo`; and (iii) a Fabric smart contract also using Go. All three implementations run at a fixed rate of 5 Hz. The role allocation algorithm does not need to run at high frequencies as it only affects the higher-frequency motion planning of the robots indirectly. The ROS 2-Fabric application subscribes to all position data, which is stored in the blockchain through a smart contract. The role allocation results from a separate smart contract are also saved in the blockchain.

From the perspective of the Fabric blockchain setup, we deploy a private Hyperledger Fabric network. The network setup process consists of the following steps: (1) **Identities**: we define six organizations, one for each of the mobile robots (representing potentially different but collaborating parties or owners), and one orderer node. Each organization is defined with a single peer and a certificate authority (CA) for simplicity. (2) **Deployment**: Docker containers for each of the peer nodes and the ordered are deployed across the network, and we initialize the genesis block. (3) **Channels**: for the purpose of demonstrating a proof-of-concept in integrating Fabric smart contracts for collaborative decision making, we create a single channel. With the channel genesis block created, all peer nodes join the same channel. (4) **Smart contracts**: we deploy the two different chaincodes for data recording and role allocation (smart contracts) through one of the peers. The rest of the peer nodes then proceed to approve the chaincodes for their respective organizations. (5) **Invocation**: once the chaincode is approved by enough members in the channel, it can be invoked by any of the peers for data storage. A ROS 2-Fabric Go application invokes the role allocation chaincode at the predefined 5 Hz frequency. The application provides to the chaincode the latest known location to all nodes, received from the different ROS 2 topics.

V. EXPERIMENTAL RESULTS

This section discusses the results obtained through our experiments. The results report the role allocation results based on data calculated with the UWB nodes mounted on the six mobile Jetbots, as well as the reference MOCAP system. We also report the latency of the three different role allocation processes that we implemented (ROS 1-Python, ROS 2-Go and Fabric smart contract in Go). Figure 3 illustrates the different paths of the Jetbots during one of the experimental runs. Four of the mobile robots, namely nodes 3, 4, 7 and 8, share the area in which they operate and their positions continuously switch between inner and outer positions between from the perspective of the robot’s convex envelope. Taking into account that the node 1 of the system is in the origin and node 2 defines the x-axis at a distance of 3.4 m, the four nodes are passive when located towards the bottom left of the graph, the inner part of the system. Node 6 is, most of the time, an passive node, only changing to active node when located towards the upper left part of its trajectory. Node 5 is also mostly passive, tending to be in the inner part of the envelope, behind the line defined by node 2 and nodes 4 or 8.

Figure 3 shows, importantly, a comparison between the results obtained with reference localization obtained by the MOCAP system and the localization obtained by using ToF ranging. For both localization schemes, the role allocation algorithm remains the same. The difference in the roles in both systems is due to the error from the UWB localization. However, this does not result in poorer system performance. The core takeaway is that the roles effectively change with both the reference MOCAP and the UWB localization. A system failure would occur when a passive node remains passive even when located outside the convex envelope of the system and far from other inner nodes. In such case, the TDoA localization method would easily diverge and unexpected positioning information be estimated by the algorithm, Table I shows the percentage of overlap of the MOCAP-based role allocation and the UWB-based role allocation results. We can see that five out of six robots show an overlap larger than 80%, with three of them showing over 90% overlap. Node 5, which has a trajectory that is inherently defined around the convex envelope, shows a lower overlap of roles with only 64.17%. This can be explained by the uncertainty in the UWB localization being bigger than the distance between the node and the actual convex envelope in most of its trajectory. While the results are satisfactory in this case, the planning module could integrate knowledge of such a limitation in the role allocation process. Therefore, the controller can define waypoints that minimize the potential for errors in the role allocation process accounting for error estimation of the UWB localization.

Finally, we analyze the impact of integrating Hyperledger Fabric smart contracts for the role allocation algorithm as a distributed and cooperative decision-making process by measuring the role allocation latency. The results are shown in Figure 4. The role allocation algorithm is running at 5 Hz for all three cases. The ROS 2 Go implementation is naturally faster than the interpreted ROS 1 Python node. However, both implementations are fast enough for meeting the frequency defined for the role allocation. Even though the addition of a Hyperledger Fabric interface to the baseline Go implementation increments its latency, it still remains fast enough. From these results, we can conclude that it is effectively possible to run this type of distributed decision-making algorithms within the Fabric blockchain without a significant impact on the performance of the system. The benefits of Fabric for this use case include the possibility of managing identities, and the inherent...
Fig. 3: Plots with the trajectories of each of the six moving Jetbots. The orange paths show the predefined trajectory of the Jetbot. The red and blue marks show the roles that the Jetbot would be allocated to, if we were using the positioning data from the MOCAP system as a reference. The red color represents an active role while blue represents a passive role. The purple and cyan marks show, respectively, the same active and passive roles calculated using ToF information from the UWB transceivers and without the external reference.

Fig. 4: The latency of the role allocation algorithm for Python, Go and Fabric+Go implementation when running the algorithm at 5Hz.

Immutability of data stored in the blockchain. In addition, the chaincode implementation of the role allocation algorithm enhances the trust on the results and within the system. There is potential for enabling trustable relative collaboration between multiple third parties or different robots. While different Fabric organizations are created for the experiments, we have focused on a proof-of-concept implementation where the actual control was not distributed. Our results, nonetheless, open the door to more secure and trustable multi-robot collaboration and identity management with a permissioned blockchain, and to more scalable localization and distributed decision-making.

VI. CONCLUSION

In this paper we have proposed, designed, and implemented a role allocation methodology for UWB-based localization relying on the Hyperledger Fabric blockchain. We analyze the performance of the role allocation algorithm with multiple moving nodes, with larger-scale experiments than in previous works. Our results show that the role allocation algorithm performs well with a higher number of mobile robots. In addition, we analyze the impact on the performance when implementing the algorithm through Fabric smart contracts, concluding that the additional latency is not significant.

In future works we seek to extend the uses of Fabric smart contracts to other collaborative decision-making problems, while extending the experiments to include aerial robots.

ACKNOWLEDGMENT

This research work is supported by the R3Swarms project funded by the Secure Systems Research Center (SSRC), Technology Innovation Institute (TII).
REFERENCES

[1] Samira Hayat, Evgen Yanmaz, and Raheeb Muzzaffar. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. IEEE Communications Surveys & Tutorials, 18(4):2624–2661, 2016.

[2] Gilles Albeaino and Masoud Gheisari. Trends, benefits, and barriers of unmanned aerial systems in the construction industry: a survey study in the united states. J. Inf. Technol. Constr., 26:84–111, 2021.

[3] IG Plaksina, GI Chistoshkina, and DV Topolskiy. Development of a transport robot for automated warehouses. In 2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), pages 1–4. IEEE, 2018.

[4] Guang-Zhong Yang, Jim Bellingham, Pierre E Dupont, Peer Fischer, Luciano Floridi, Robert Full, Neil Jacobson, Vijay Kumar, Marica McNutt, Robert Merrifield, et al. The grand challenges of science robotics. Science robotics, 3(14):eaaar7650, 2018.

[5] Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pulllinen, Victor Kathan Sarker, Tian Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. IEEE Access, 8:191617–191643, 2020.

[6] W Stempfhuber and M Buchholz. A precise, low-cost rtk gnss system for uav applications. Proc. of Unmanned Aerial Vehicle in Geomatics, SNPR, 2011.

[7] Yang Qi, Yisheng Zhong, and Zongying Shi. Cooperative 3-d relative localization for uav swarm by fusing uwb with imu and gps. In Journal of Physics: Conference Series. IOP Publishing, 2020.

[8] Xiaosi Chen, Chong Shen, Qian Gao, Qun Zhou, and Gaogao Feng. Network scalability with weight analysis based on uwb indoor positioning system. In 2016 IEEE Conference on Wireless Sensors (ICWISE), pages 95–99. IEEE, 2016.

[9] Wang Shule, Carmen Martinez Almansa, Jorge Peña Queralta, Zhuo Xiaosi Chen, Chong Shen, Qian Gao, Qun Zhou, and Gaoang Feng. Network scalability with weight analysis based on uwb indoor positioning system. In 2016 IEEE Conference on Wireless Sensors (ICWISE), pages 95–99. IEEE, 2016.

[10] Yu Xinjia, Li Qingqing, Jorge Pena Queralta, Jukka Heikkonen, and Tomi Westerlund. Cooperative uwb-local based localization for outdoors positioning and navigation of uavs aided by ground robots. In 2021 IEEE International Conference on Autonomous Systems (ICAS), pages 1–5. IEEE, 2021.

[11] Kexin Guo, Zhirong Qiu, Cunxiao Mao, Abdul Hanif Zaini, Chin-Lin Chen, Wei Meng, and Lihua Xie. Ultra-wideband-based localization for quadcopter navigation. Unmanned Systems, 4(01):23–34, 2016.

[12] Jorge Peña Queralta, Carmen Martínez Almansa, Fabrizio Schiano, Dario Floreano, and Tomi Westerlund. Uwb-based system for uav localization in gns-denied environments: Characterization and dataset. In IROS. IEEE, 2020.

[13] Wahab Khawaja, Ozgur Ozdemir, Fatih Erden, Ismail Guvenç, and David W Matolak. Uwb air-to-ground propagation channel measurements and modeling using uavs. In 2019 IEEE Aerospace Conference, pages 1–10. IEEE, 2019.

[14] Michael Hamer and Raffaello D’Andrea. Self-calibrating ultra-wideband network supporting multi-robot localization. IEEE Access, 6:22292–22304, 2018.

[15] Weihua Gao, Ganapathi Kamath, Kalyan Veeramachaneni, and Lisa Osadiw. A particle swarm optimization based multilateration algorithm for uwb sensor network. In 2009 Canadian Conference on Electrical and Computer Engineering, pages 950–953. IEEE, 2009.

[16] Kexin Guo, Zhirong Qiu, Wei Meng, Lihua Xie, and Rodney Teo. Ultra-wideband-based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments. International Journal of Micro Air Vehicles, 9(3):169–186, 2017.

[17] Pablo Corbalán and Gian Pietro Picco. Concurrent ranging in ultra-wideband radios: Experimental evidence, challenges, and opportunities. In EWSN, pages 55–66, 2018.

[18] Milad Heydarian, Hossein Dahiran, and Omprakash Gavareli. Aguloc: Concurrent angle of arrival estimation for indoor localization with uwb radios. In 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), pages 112–119. IEEE, 2020.

[19] Davide Vecchia, Pablo Corbalán, Timoefi Istomin, and Gian Pietro Picco. Talla: Large-scale tdoa localization with ultra-wideband radios. In 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–8. IEEE, 2019.

[20] Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. Snaploc: An ultra-fast uwb-based indoor localization system for an unlimited number of tags. In Proceedings of the 18th International Conference on Information Processing in Sensor Networks, pages 61–72, 2019.

[21] Joao P Jansch-Porto and Geir E Dullerud. Robust decentralized switching control of uavs using uwb-based localization and cooperation. IFAC-PapersOnLine, 53(2):7418–7423, 2020.

[22] Lakshmi Siva Sankar, M Sindhuv, and M Suthumadhavan. Survey of consensus protocols on blockchain applications. In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pages 1–5. IEEE, 2017.

[23] Ruffin White, Gianluca Ciazzia, Agostino Cortesi, Young Im Cho, and Henrik I Christensen. Black block recorder: Immutable black box logging for robots via blockchain. IEEE Robotics and Automation Letters, 4(4):3812–3819, 2019.

[24] Wenbo Wang, Dinh T Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, and Dong In Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access, 7:222326–222370, 2019.

[25] Truong T Nguyen, Amartya Hatua, and Andrew H Sung. Blockchain approach to solve collective decision making problems for swarm...
robotics. In *International Congress on Blockchain and Applications*, pages 118–125. Springer, 2019.

[39] Jorge Peña Queralta and Tomi Westerlund. Blockchain for mobile edge computing: Consensus mechanisms and scalability. In *Mobile Edge Computing*, pages 333–357. Springer, 2021.

[40] Salma Salimi, Jorge Peña Queralta, and Tomi Westerlund. Towards managing industrial robot fleets with hyperledger fabric blockchain and ros 2. *arXiv e-prints*, 2022.

[41] USP Srinivas Aditya, Roshan Singh, Pranav Kumar Singh, and Anshuman Kalla. A survey on blockchain in robotics: Issues, opportunities, challenges and future directions. *Journal of Network and Computer Applications*, 196, 2021.

[42] Paulo Ávares, Lion Silva, and Naercio Magaia. Blockchain-based solutions for uav-assisted connected vehicle networks in smart cities: A review, open issues, and future perspectives. In *Telecom*, volume 2. Multidisciplinary Digital Publishing Institute, 2021.