Development of a robust Wi-Fi/4G-based ROS communication platform for an assembly and repair mobile robot with reliable behavior under unstable network or connection failure

Efim Poberezkin1 · Hamid Roozbahani1 · Marjan Alizadeh1 · Heikki Handroos1

Received: 26 November 2021 / Accepted: 10 July 2022 / Published online: 5 September 2022
© The Author(s) 2022

Abstract
This paper presents various perspectives on designing and implementing a communication platform for a teleoperated mobile robot. The deployment of a communication network for a mobile robot and the integration of robot components in the developed communication platform are discussed in this article. A Wi-Fi-based communication network has been established, and to secure remote control over long distances, Internet-based communication via the 4G protocol has been launched using a virtual private network setup. Since an unstable network can damage the robot or the surrounding environment, an algorithm has been developed to monitor the state of the network connection through different protocols. The developed algorithm is able to detect network failure independently of the wireless communication technology used and notify the system of any disruptive communication. The robot’s primary reactions to connection failure have been programmed to keep the robot under control until the communication with the control station is restored. Various experiments were carried out to validate the performance of the designed algorithm and statistical analysis was performed for each experiment. One of the main contributions of this study is the development of an algorithm for communication between the mobile robot and the control station based on both Wi-Fi and 4G, which is capable to keep the robot behavior safe and reliable in the presence of an unstable network or connection failure.

Keywords Communication platform · Mobile robot · ROS · Robot network failure · Wireless network · 4G

1 Introduction

Removing humans from dangerous areas and using mobile robots to perform specific operations in hazardous zones has existed for decades. The mobile robots should be able to be controlled remotely by an operator to perform the required tasks [1–3]. The establishment of an effective communication system is indispensable for this purpose. Communication plays a crucial role in the performance of a robot. The processes of sending or receiving data by various components of the robot imply communication in robotics. It includes communication between the operator in the control station and the robot, between control devices and specific hardware of the robot, between different running nodes that are executed by the processor of robot, and so on [4, 5]. Different communication methods can be used by mobile robots, including wired and wireless communications. Although wired communications can provide higher communication quality, wired communications for a teleoperated mobile robot are not desired due to the various physical limitations caused by cables [6–8]. Generally, wireless communications, such as Bluetooth, Wi-Fi and cellular network technology are applied to mobile robots due to the valuable benefits they offer [6, 7, 9, 10]. Each communication method has unique advantages; however, the best communication method for each mobile robot is determined based on the application of the robot and its specific requirements. Radio waves are useful for long-distance communication but are not suitable if there are obstacles in the movement area of the mobile robot due to the possible obstruction of the waves [8, 11]. Wi-Fi technology is widely used as a communication method for mobile robots. However, Wi-Fi technology suffers from limited communication distance and the possibility...
of network failure [7, 12]. Internet-based technology substantially contributes to the establishment of communication systems for mobile robots. However, problems such as internet transmission delays or insecure bandwidth are inevitable in the development of internet-based communication systems for mobile robots [9, 13, 14].

Poor communication quality or loss of communication can lead to damage to the mobile robot or the operational area of the robot. Various studies have been performed to enhance the quality of wireless networks of tele-operated mobile robots by improving hardware configuration and software deployment [5–7, 13–15]. Nevertheless, the development of an algorithm to verify the connection status and notify the system of delayed or loss of communication, as well as the programming of actions to be taken by the mobile robot in the event of a connection disruption, has not been sufficiently studied. In addition, although different types of communication systems and methods have been studied for mobile robots, in most cases only one specific method was considered for this purpose.

This study focuses on creating a dual communication system for a teleoperated mobile robot described in Sect. 1.1. It is worth mentioning that the system studied in this research is not among the multimodality systems. Two types of communication methods, including Wi-Fi and 4G, have been established for the mobile robot in question to strengthen the possibility of establishment of communication considering the possible existence of electromagnetic interferences in the robot operating environment. The Internet-based communication method (4G) has been created due to the limited communication distance that can be covered by Wi-Fi communication. In addition, Internet-based communication enables the operator to control the robot remotely from anywhere else, for instance, if the knowledge and skills of a particular operator are required for performing a special task. Since the computing devices and other components of a mobile robot are able to operate without the control of the robot by the operator, therefore, an algorithm was developed for communication between the mobile robot and the control station to determine the actions that should be taken by robot and keep the robot’s behavior safe in case of an unstable network or lost connection. Although the developed communication platform can be applied to any mobile robot using different implementation mechanisms, the case study of this paper is LUT mobile assembly and repair robot, which conducts assembly and repair operations in hazardous areas.

### 1.1 Overview of LUT mobile assembly and repair robot

The research was conducted at Lappeenranta-Lahti University of Technology (LUT) as part of the Mobile Assembly and Repair Robot project. The main goal of this project was to develop a mobile robot that replaces humans in hazardous areas to perform assembly and repair operations. The robot can be controlled remotely by an operator, which allows operators to be kept at remote distances from hazardous areas. Toxicity, radiation, explosion, corrosion, high voltage, and excessive temperatures are a few samples of such hazards in human operating environments [16–18]. The developed robot is presented in Fig. 1. Meanwhile, the application of the robot will not be limited to repair and assembly in hazardous areas. The versatility and modularity of the concept allow it to be utilized in a wide range of industrial applications and other areas of human activity [16–18].

Robot operating system (ROS) was chosen as the main communication platform between most of the robot components and controllers. The chief robot control device is Advantech ARK-3440 A2, which is a fanless embedded computer [19]. This insensitivity of embedded computer to dust, cold and heat, contamination, shock and vibration, EMI, and other harmful factors, makes it a suitable option for machine automation. All the robot systems are under control via this embedded computer, which is in contact with the remote-control station. In this study, Linux Operating System (OS) Ubuntu 14.04 has been used, which is supported by the embedded computer mentioned above. The robot locomotion’s main components are four aluminum omnidirectional Mecanum wheels 254 mm. The wheels are powered by Maxon Motor motors and the controlling is performed by easy positioning system (EPOS) controllers. Mecanum wheels have the capability to move freely in closed environments with static and dynamic obstacles [20]. The front Omni-directional wheels of the studied robot are presented in Fig. 2.

A separate EPOS2 digital positioning controller controls each motor and eventually results in the independence of each wheel’s velocity and orientation. This feature enables the robot to move in various trajectories [21]. RS-232 cable connects the computer to the EPOS2 controllers, and epos_hardware of ROS package is used to control the motors. The robot has been equipped with two industrial robot arms Universal Robot 10 (UR10) installed in the

---

**Fig. 1** The developed LUT mobile robot
robot chassis to perform tasks with high accuracy [22]. An operator controls each UR10 robot via a Geomagic Touch haptic joystick, enabling the operator to employ both robot arms simultaneously for assembly and repair tasks. Also, to facilitate the operator’s performance, Robotiq 3-Finger Adaptive Robot Grippers have been installed on both arms [23]. UR10 robotic arms, grippers, and haptic joysticks are presented in Fig. 3. Arms and grippers have been connected to the computer via Ethernet. Universal Robot and ROS Robotiq packages have been utilized for control purposes.

A set of four cameras have been installed on the robot to provide the operator with video feedback. Four SWIT S-4914 transmitter–receiver sets provide a wireless connection to transmit camera data. Two paired Marshall cameras have been located on the robot’s head. The operator can control the neck actuators via a joystick through ROS to drive the robot head in motion. Moreover, robot grippers have been equipped with two separate cameras to provide additional feedback. The robot has been equipped with two human-shaped surrounding ears attached to the head with voice command capability via ROS. The robot head is presented in Fig. 4.

2 Means and methods of communication

Robot operating system (ROS) has been utilized as a framework for developing the robot application of this study. The developed robot in this work combines various hardware pieces into a unified ROS network and utilizes existing control codes and drivers [24, 25]. For this project, ROS distribution Indigo Igloo package was used, as some beneficial control packages are available with this version. For the mobile robot under study, ROS-based computing is performed by the distributed system, including the robot-embedded Advantech computer and the computer-based control station. Two separate virtual machines have been used in the control station to control driver nodes from the operator side including Xbox 360 joystick for the locomotion system and Geomagic Touch for arms control. Robot embedded computer controls all driver nodes related to the robot hardware such as UR10 robotic arms, EPOS controllers, Robotiq grippers and more. The embedded computer of robot was selected as the master machine in this study. In this way, if the robot disconnects from the control station and new nodes have to be set up, they can register in the master machine.

2.1 Communication over Wi-Fi

A wireless medium is required for communication between the operator and mobile robot. For this purpose, Wi-Fi technology has been used for the communication system of the developed robot [26, 27]. The communication network with network switches for robot onboard devices and two Robustel R3000-Q4LB industrial cellular routers has been launched. Omnidirectional antennas have been externally mounted on both routers to extend the Wi-Fi signal range [28, 29]. The robot computer and the control station are the minimum devices that must be included in the teleoperated mobile robot network. For stable communication and management simplification, static IP addresses have been assigned to the other devices communicating directly
through the network. The computer network diagram of this study is presented in Fig. 5.

The communication network has been divided into two major subnetworks, which are 192.168.0.0 assigned for the control station and 192.168.2.0 assigned for the embedded computer of the robot and its onboard components. Both subnetworks are governed by Robustel routers. The router in the subnet 192.168.0.0 operates in Wi-Fi access point mode and allows the router in the subnet 192.168.2.0 to connect to the control station subnet over Wi-Fi [28]. The onboard robot router operates in a client mode and serves as a gateway from one network to another, as shown in Fig. 5. ICMP primary detection server determines whether the router remains online and connected securely to the access point [29]. Ethernet has been used to connect both computers to the routers. The control room virtual machine interfaces connected to the 192.168.0.0 subnet are configured as bridge networking [30]. In order for each network node to reach other nodes in a different subnet, permanent static routes were appended to each machine's routing table. This process is automated by dynamic host configuration protocol (DHCP) [28, 31].

2.2 Communication over Internet

Due to the limited communication distance of Wi-Fi, Internet-based communication was also established for the mobile robot under study. 4G cellular technology was selected for this purpose. The control station is also connected to the Internet. All nodes of the robot have been joined to the private network and a virtual private networking (VPN) has been organized for suitable communication via the Internet [32, 33]. The scheme of the virtual private network developed for the LUT mobile assembly and repair robot is presented in Fig. 6. It indicates the feasibility of ROS communication via the Internet between far nodes of the system. The robot and the operator have been connected to a dedicated VPN server. Creating another client of the virtual private network for the operator provides the option of controlling the robot by any remote operator.

3 Development of robot communication system

The operator, who controls the mobile teleoperated robot remotely, sends control data to the robot through a communication channel. Particular control devices such as pedals or joysticks or direct commands from the terminal create the control inputs. In the latter case, the controller input conversion into the data that the robot can interpret is bypassed. Basically, robot control is performed through ROS. Special ROS packages include nodes with driver code for the corresponding robot components. ROS topics are used to communicate between nodes [24]. For instance, ur_modern_driver package has been designed to control UR10 robotic arms. The connection to the right UR10 arm is established by running roslaunch command for this package inside the Ubuntu terminal and the arm is enabled to move. The driver node of the Geomagic Touch device sends the positional data to a particular topic, that the arm node listens to. Control systems for different components of the robot were developed and integrated into a united ROS-based communication system. Some challenges were raised during the
accomplishment of this task. Since the work was performed across several computers asynchronously, integration of those systems created conflicts when they were transferred to the robot computer and the control station. The conflicts were resolved, and proper communication was established between different components of the system. Also, conflicts between control systems adapted from ROS-ready packages were resolved. For example, simultaneous use of two UR10 arms and two Geomagic Touch controllers is not supported by the ROS package. New packages were created to activate both arms at the same time, and communication problems that occurred between them were solved. Another example is the conflict made by joysticks, which control both wheels and the neck. Joysticks for both the wheels and the neck were powered by \textit{joy_node}, topic /joy package, thus creating a naming conflict. This problem was resolved by changing one of the nodes and one of the topics.

3.1 Robot control and communication system under unstable connection

Communication failure between the robot and the operator can result in unintended consequences, like damage to the robot or harm to the surrounding environment [34, 35]. Therefore, it is crucial that the robot be able to recognize connection failure and be programmed beforehand for essential reactions to such events [36, 37]. Network failure can be identified using different approaches [36, 38, 39]. Continuous monitoring of connection status is an effective solution. The initial method of checking the connection state is to monitor signal strength. Because Wi-Fi communication was initially implemented in this study, Wi-Fi signal strength of $-70$ dBm was monitored for reliable data delivery. However, this method is only applicable to a particular wireless communication technology. Hence, implementation of that was refused. Instead, a general method independent of wireless communication technology used was investigated. An effective monitoring software tool monitors different open system interconnection (OSI) layers through which communication is conveyed [40, 41]. Protocols of the transport layer, UDP, and TCP are utilized by ROS for transmitting messages [24]. One solution for evaluating a TCP connection is to track the number of retransmitted segments and compare the number of segments sent. Monitoring of statistics given by netstat on Linux systems can be performed by continuously running it. The robot reacts to the loss or weakening of the connection by receiving a message about that event from the control system. The message is passed via a topic, which notifies other ROS nodes. A code for analyzing the connection state was developed during the study. The analysis is carried out based on the data generated by \textit{rosping}, which is a modified ROS package, and values of ping delay. Depending on the various levels of delay or complete connection loss, notifying messages are created. In contrast with the connection loss that was easy to obtain, various experiments were conducted to achieve different levels of delay. The \textit{rosping} package node constantly pings certain IP address sent to it as a parameter and publishes delay data into /ping/delay topic and the alerting node subscribes to it. In case that node fails to reach the destination interface, it publishes a value of “$-1$” into the /ping/delay topic. Also, the alerting node advertises its own topic – communication state. The target is to notify other
nodes about the connection status by transmitting integer values, which determines the connection status, from the alerting node. If a value “−1” is received from the /ping/delay topic by the alerting node, it republishes the value “−1” into the communication_state topic in order to notify other nodes to react to it. If the alerting node receives a value other than “−1”, it publishes a value “0” into the communication_state topic to notify other nodes that the connection is restored. The proposed algorithm has the ability to inform the system about various connection quality levels. To fulfill this purpose, values for three levels of ping have been set by the algorithm. The working concept is that the last few values of the delay received from the /ping/delay topic are compared to the set values, and in case of compliance with the defined criteria, it publishes the corresponding integer value into the communication_state topic. The alerting node also publishes messages into the /rosout topic—system-wide ROS logging tool, in addition to publishing into the communication_state topic [24]. The alerting node tracks the current state of the connection level with the help of several Boolean variables to prevent identical messages from being published [28]. The simplified block diagram of the developed algorithm is shown in Fig. 7.

The diagram has no end block because this code runs continuously while the robot is working. rosping nodes and alerting nodes run on the robot computer as well as in the operator control station. Other nodes analyze the data published by alerting node for appropriate response and action, such as stopping the robot wheels. In case of a connection failure, wheels can cause a crash since their control system has been programmed in such a way that they continue spinning with a defined velocity, and the operator is unable to control the robot. To avoid this problem, if the driver node of the wheels receives a value “−1”, it must subscribe to the communication_state topic and set the velocity of each wheel to zero. For other driver nodes, similar actions were implemented; for instance, in case of a connection failure, the control system of the UR10 robotic arms commands the arms to return to the initial position. A separate alerting node runs in the operator control station to warn the operator about the connection failure and triggers an alarm to alert the operator.

3.2 Statistical analysis of delay

To be able to evaluate the connection status more effectively and possibly even predict it for a short time, statistical analysis of delay experiments was performed. The analyzed sample was delay values that have occurred over the last 5 min. The sample contains three hundred values after the initial 5 min because the target interface is pinged every second. Mathematical expectation, variance, and frequency histogram were analyzed. The analysis of expectation and delay variance was performed in the first step. Since the sample of delay values is constantly changing, expectation and variance can be calculated at each step, and the dynamic of their change can be analyzed [28]. The expectation is presented as an unbiased sample estimate of expectation, or sample mean, denoted as $\overline{X}$ due to the random values of the sample. The expectation of delay is calculated by Eq. (1) [42]:

$$E[X] = \overline{X} = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

where $X$ is the random variable, $n$ is the size of the sample and $x_i$ is the value of the random variable $X$. A value called unbiased sample variance $S^2$ is employed to estimate the variance of sample values $x_1,...,x_n$ of a random variable $X$ (Eq. 2). This value is analyzed at each step [42]:

$$S^2 = \frac{1}{n-1} \left[ \sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n} \right] = \frac{\sum_{i=1}^{n} (x_i - \overline{X})^2}{n}.$$  

Several experiments were conducted to evaluate the performance of the developed algorithm. The experiments were performed in an area of 40 $m^2$ and a height of 2.8 m with concrete walls and metal doors without glass coverage. Statistical analysis was performed for each experiment. The initial idea of the code was to recognize a connection failure and generate messages notifying about several levels of delay and complete loss of connection. The robot was tested both in stationary and motion modes. The experiment of complete loss of connection was simply performed by
switching off the robot onboard router. The algorithm succeeded in recognizing the connection loss and creating the correct notification message. In addition, as programmed by the algorithm, the robot was kept in frozen mode by immediately stopping the wheels and arms of the robot.

Unlike the connection loss experiment that was simple to achieve, several methods were examined to weaken the connection, including placing the robot router behind a concrete wall or covering it with lead ware, detaching the router antenna or using a kind of jammer device. However, all attempts to create a connection with high latency by imitating consistently high ping were not succeeded. During all the experiments, the ping showed very low values meaning good connection with very moderate sporadic individual increases, which could not qualify as a persistently high delay. Eventually, the connection weakening experiment was performed by connecting the laptop to the network and gradually increasing the distance from the test environment. Although it was not possible to achieve persistently high ping during the experiments, the developed algorithm successfully identified the weak connection and notified the system about the level of delay.

The value of delay, expectation, variance, and frequency histogram were analyzed for each experiment. The waiting time to respond was set to one second, and the sample was planned to reset whenever a destination interface was inaccessible. Samples of expectation and variance during such an experiment are presented in Figs. 8 and 9. As shown by these diagrams, when there is no ping or the value cannot be calculated, the results are skewed to $-100$. It can strongly be concluded that both expectation and variance rise significantly before the connection failure. Variance remarkably follows this fact, in the sense that when the response delay exceeds the waiting time and ping increases gradually, the variance reacts too sharply. The experiments revealed that whenever the response does not receive during the waiting time, it does not necessarily mean that the connection is completely lost, although it means a weakening of the connection status. Only a few subsequent drops in responses occurred, leading to unjustified reset of the sample. As a result, it was decided that if the destination interface could not be reached, the sample would not be reset immediately, and a special counter would be utilized instead. Each time that the response is not received, the counter is incremented, and resets if the delay is normalized. If the counter reaches a certain value, set to 30, the sample is reset. Similar to previous tests, experiments demonstrate an increase in expectation and variance before the connection failure. Also, the results show that these values remain high after the first drop, which can be described by occasional too high delay values. These periods are related to the times when the connection becomes unstable. Indeed, the most significant data corresponded to the period before the first connection failure, which declares the transition from a steady state with confident reception to a disconnection state. Figures 10, 11, and 12 demonstrate the output of various experiments before the first connection failure. Graphs display values of delay, expectation, variance, and variance trend before the first connection failure based on data of experiment no. 1.
expectation, and variance at the edge of a steady state just before the transition to an unstable connection state.

Another statistical entity studied was the frequency histogram. Initially, the detection of minimum and maximum values in the sample at each stage during the experiments dynamically determines the histogram range. This domain was then divided into intervals according to an adjustable variable that represents the range of such intervals. Afterwards, a special cycle defined in the program checks each value of the sample and increments an element of the array corresponding to the interval of value. However, analysis of the histogram is overcomplicated since the range of the histogram changes each time a new minimum or maximum delay value emerges, which may generate different values for new intervals of the same number. Therefore, a fixed histogram consisting of fifty intervals of twenty milliseconds was considered. The dynamic of histogram change can be represented as a sequence of histograms in a three-dimensional space due to instant updating, as illustrated in Fig. 13.

In Fig. 13, the height of the histogram columns correlates to the amount of sample delay values at a given time associated with the relevant intervals. From the figure, it can be concluded that by increasing the distance to the target interface and decreasing the connection stability, the histogram dome moves slightly to the right. However, no specific information is provided by this observation that can be utilized to predict connection status; hence, the frequency histogram analysis was excluded from the software developed to analyze the connection status. Therefore, expectation and variance are the basis of statistical analysis software output. Before connection drops, a significant increment is presented by the variance graphs. During the experiments, it was noticed that variance rises at a much-accelerated rate than expected in the case of an unstable connection. Due to the fact that variance provides an absolute estimate of the deviation measurement, a relative indicator called the coefficient of variation $c_v$ is employed according to Eq. (3).

$$c_v = \frac{S}{\bar{X}}$$

(3)

$S$ is equal to the square root of the unbiased sample variance $S^2$ and is the standard deviation. Using this indicator makes it possible to compare the homogeneity of the phenomena regardless of the scale and units of measurement. During the conducted experiments, it was also concluded that the mentioned indicator could be engaged as a sign of connection status. The coefficient of variance $c_v$ was fluctuating between values 0.5–0.8 when the system was in the stationary state, while it was significantly increasing to values 2–3 each time the system was entering an unstable connection. An increase in the value of the mentioned indicator can be utilized as a signal for the system when approaching a nonstationary mode and an unstable connection.
4 Conclusion

The main aim of this research was to design and implement a communication system for a teleoperated mobile robot. The robot under study is a versatile robot that can perform assembly and repair operations in hazardous areas. During the study, diverse aspects communication system design for a mobile robot were investigated. The research work includes the establishment of a mobile robot communication network and the development of an algorithm to monitor the connection status and determine the appropriate response of the robot to connection failure. As ROS was chosen as the main communication platform, ROS requirements were considered for a distributed computing and networking system. A wireless network based on Wi-Fi was modeled to join robot hardware and controlling devices to unified ROS-based communication infrastructure. Wireless communication based on the Internet 4G was also investigated due to the limited distance range of a Wi-Fi network. Various methods were studied to monitor the connection status for different protocols used by ROS. A program was developed to notify the system about the connection status, and in case of complete communication loss between the robot and control station, it keeps the robot in a frozen mode until the communication with the control station is restored. Several experiments were conducted to evaluate the performance of the developed algorithm and statistical analysis was performed for each experiment. Statistical analysis of delays provided a tool to evaluate the state of connection more effectively and possibly even anticipate it for a short time.

The future research direction will be developing an autonomous navigation system to return the robot to the base in case of network failure instead of freezing the robot movement.

Funding Open Access funding provided by LUT University (previously Lappeenranta University of Technology (LUT)).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Yarad Jácome MJ, Carguachi JB (2021) Mobile robot teleoperated through wireless communication and video acquisition in real-time for hazard applications. In: Botto-Tobera M, Cruz H, Díaz Cadena A (eds) Artificial intelligence, computer and software engineering advances. CIT 2020. Advances in intelligent systems and computing 1326, Springer, Cham. https://doi.org/10.1007/978-3-030-68080-0_33
2. Rubio F, Valero F, Llopis-Albert C (2019) A review of mobile robots: concepts, methods, theoretical framework, and applications. Int J Adv Robot Syst. https://doi.org/10.1177/1729881419839596
3. Popescu D, Roibu H, Manta FL et al (2020) Design of a mobile robot to work in special environments. In: 2020 international conference and exposition on electrical and power engineering (EPE), Iasi, Romania, Oct 22–23, 2020, pp 226–231. https://doi.org/10.1109/EPE45022.2020.9305611
4. Shneier MO, Bostelman RV (2015) Literature review of mobile robots for manufacturing. National Institute of Standards and Technology, US Department of Commerce, Gaithersburg
5. Moniruzzaman MD, Rassaua A, Chaia D, Shamsulislam SM (2022) Teleoperation methods and enhancement techniques for mobile robots: a comprehensive survey. Robot Auton Syst. https://doi.org/10.1016/j.robot.2021.103973
6. Haxhibeqiri J, Alizadeh E, Moerman I, Hoebeke J (2018) Flexible Wi-Fi communication among mobile robots in indoor industrial environments. Mob Inf Syst 2018:1–19. https://doi.org/10.1155/2018/391302
7. Parasuraman R (2014) Wireless communication enhancement methods for mobile robots in radiation environments. Ph.D. thesis, Universidad Politécnica de Madrid. https://doi.org/10.13140/RG.2.2.27878.93769
8. Sawai K, Aoyama S, Motoyoshi T et al (2019) A mobile robot teleoperation system with a wireless communication infrastructure using a leaky coaxial cable based on TCP/IP. Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2019.0100713
9. Kahar S, Sulaiman R, Prabuwono AS et al (2012) A review of wireless technology usage for mobile robot controller. In: 2012 international conference on system engineering and modeling (ICSEM 2012)
10. Sawai K, Yoshii Y, Oshima T et al (2018) A study of wireless mobile robot tele-operation method using zigbee networks as reserve communication infrastructure. In: The proceedings of JSME annual conference on robotics and mechatronics (Robomec) 2018, p 1P2-D08. https://doi.org/10.1299/jsmermd.2018.1P2-D08
11. Parasuraman R, Kershaw K, Ferre M (2013) Experimental investigation of radio signal propagation in scientific facilities for telerobotic applications. Int J Adv Robot Syst. https://doi.org/10.5772/56847
12. Sweatt M, Ayoade A, Han Q et al (2015) WiFi based communication and localization of an autonomous mobile robot for refinery inspection. In: Proceedings of the IEEE international conference on robotics and automation, Seattle, USA, May 26–30, 2015, pp 4490–4495. https://doi.org/10.1109/ICRA.2015.7139821
13. Phung MD, Hoang TT, Van NTTT et al (2012) A novel platform for internet-based mobile robot systems. In: 2012 7th IEEE conference on industrial electronics and applications (ICIEA), Singapore, July 18–20, 2012, pp 1972–1977. https://doi.org/10.1109/ICIEA.2012.6361052
14. Tsuzuiki R, Yamauchi G, Nagatani K, Yoshida K (2014) Teleoperation of mobile robots using hybrid communication system in unreliable radio communication environments. In: 2014 IEEE international symposium on safety, security, and rescue robotics,
Hokkaido, Japan, October 27–30, 2014, pp 1–5. https://doi.org/10.1109/SSRR.2014.7017649

15. Lindhorst T, Lukas G, Nett E (2013). Wireless mesh network infrastructure for industrial applications—a case study of tele-operated mobile robots. In: 2013 IEEE 18th conference on emerging technologies and factory automation (ETFA), Cagliari, Italy, September 10–13, 2013, pp 1–8. https://doi.org/10.1109/ETFA.2013.6647955

16. Roozbahani H, Handroos H (2017) Developing a mobile assembly robot for hazardous environments with CompactRIO and ROS—a case study. LUT Mob Assem Robot. https://doi.org/10.13140/RG.2.2.33155.60968

17. Roozbahani H, Handroos H (2018) TIERA—Mobile assemblyblagerobot voor Gevaarlijke omgevingen. Vis Robot 10(2):12–15

18. Roozbahani H (2018) Mobile assembly and repair robot for hazardous areas. In: ATEX workshop, Aarhus Denmark. https://doi.org/10.13140/RG.2.2.21455.18082

19. Advantech ARK-3440 A2. (2015) https://www.ipc-systems.com/documents/ARK-3440_A2.pdf. Accessed 10 June 2022

20. Diegel O, Badve A, Bright G, Potgieter J, Tlale S (2002) Improved mecanum wheel design for omni-directional robots. In: Australasian conference on robotics and automation, Auckland, pp 117–121

21. Maxon EPOS. http://www.maxonmotor.com/maxon/view/content/EPOS-Detailsite. Accessed 10 June 2022

22. Universal Robots. UR10 robot. https://www.universal-robots.com/products/ur10-robot/. Accessed 10 June 2022

23. Robotiq. 3-Finger Adaptive Robot Gripper. http://robotiq.com/products/industrial-robot-hand/. Accessed 10 June 2022

24. ROS documentation. http://wiki.ros.org. Accessed 10 June 2022

25. Araújo A, Portugal D, Couceiro MS, Rocha RP (2013) Integrating arduino-based educational mobile robots in ROS. In: 13th international conference on autonomous robot systems, Lisbon, pp 1–6. https://doi.org/10.1109/Robotica.2013.6623520

26. Labiod H, Afifi H, De Santis C (2007) WI-FI TM, BLUE TOOTH TM, ZIGBEE TM AND WIMAX TM. Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-5397-9

27. Xu S, Chou W (2017) An improved indoor localization method for mobile robot based on WiFi fingerprint and AMCL. In: 10th international symposium on computational intelligence and design (ISCID), Hangzhou, China, pp 9–10. https://doi.org/10.1109/ISCID.2017.25

28. Poberezkin E (2017) Design and development of communication system for mobile robot. Master thesis. https://lupub.ltu.fi/handle/10024/135060

29. Robustel. Robustel GoRugged R3000 Quad user guide. https://www.robustel.com/product/r3000-quad-industrial-lot-gateway/. Accessed 10 June 2022

30. VirtualBox. Bridged networking. https://www.virtualbox.org/manual/ch06.html#network_bridged. Accessed 10 June 2022

31. Microsoft Corporation. What Is DHCP? https://technet.microsoft.com/en-us/library/dd145320(v=ws.10).aspx. Accessed 10 June 2022

32. Microsoft Corporation (2003) Virtual private networking with Windows Server 2003

33. Seid HA, Lespagnon A (1998) Virtual private network. Alcatel Data Networks Inc. Appl. US 08/632168

34. Nagatani K, Kiribayashi S, Okada Y et al (2013) Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. J Field Robot 30(1):44–63. https://doi.org/10.1002/rob.21439

35. Magid E, Pashkin A, Simakov N et al (2020) Artificial intelligence based framework for robotic search and rescue operations conducted jointly by international teams. In: Proceedings of 14th international conference on electromechanics and robotics “Zavalishin’s Readings”, pp 15–26. https://doi.org/10.1007/978-981-13-9267-2_2

36. Bereznikov D, Zakiev A (2020) Network failure detection and autonomous return for PMB-2 mobile robot. Proc Int Conf Artif Life Robot 25:444–447. https://doi.org/10.5954/ICAROB.2020.OS18-5

37. Alishiev N, Lavrenov R, Hsia K-H et al (2018) Network failure detection and autonomous return algorithms for a crawler mobile robot navigation. In: 2018 11th international conference on developments in eSystems engineering (DeSE), Cambridge, UK, September 2–5, 2018, pp 169–174. https://doi.org/10.1109/DeSE.2018.00040

38. Tardioli D, Mosteo AR, Riazuelo L et al (2010) Enforcing network connectivity in robot team missions. Int J Robot Res 29(4):460–480. https://doi.org/10.1177/0278364909358274

39. Derbakova A, Correll N, Rus D (2011) Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems. In: 2011 IEEE international conference on robotics and automation, Shanghai, China, May 9–13, 2011, pp 3863–3868. https://doi.org/10.1109/ICRA.2011.5980367

40. Fernandes A, Couceiro MS, Portugal D, Santos JM, Rocha RP (2015) Ad hoc communication in teams of mobile robots using zigbee technology. Comput Appl Eng Educ 23(5):733–745. https://doi.org/10.1002/cae.21646

41. Alani MM (2014) Guide to OSI and TCP/IP models. Springer International Publishing, New York. https://doi.org/10.1007/978-3-319-05152-9

42. Olive DJ (2014) Statistical theory and inference. Springer International Publishing, New York. https://doi.org/10.1007/978-3-319-04972-4

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.