



Design of an Android application for autonomous control of mobile robot

Mobil robotun otonom kontrolü için android uygulama tasarımı

Ahmet Top^{1,*} , Muammer Gökbulut² 

^{1,2} Department of Electrical and Electronic Engineering, Faculty of Technology, Fırat University, Elazığ, Türkiye

Abstract

In this study, a mobile application (app) for Android devices (ADs) is designed to gather robot data and supply the reference values needed for the autonomous operation of mobile robots intended for various uses. Robot control settings can be supplied with any AD using this application; no further hardware or computer is needed for robot control. Beyond providing manual control, the application also has an interface for autonomous control. It allows the robot to be directed to certain locations by sending it three coordinate points and the approach distance. Furthermore, as the robot moves, real-time data can be received by Bluetooth (BT) on the AD, and the robot's current position is shown on the device's screen. To demonstrate the efficacy of the study, control of the robot with Proportional-Integral-Derivative (PID) and Fuzzy logic controller (FLC), whose prototype are carried out, was performed per the references received from the AD. Meanwhile, real-time data about the robot was gathered on the same AD with the Excel file extension, mailed to the PC, and data graphs were created. The study has demonstrated that ADs may be used for both manual and autonomous control of robots.

Keywords: Android application, MIT App Inventor, Robot control

1 Introduction

Robotics technology has advanced significantly in recent years, along with every other technology. They provide the necessary workforce in sectors where human labor is insufficient or poses harm to human health [1]. While building robots, it's crucial to consider usability and functionality. Because of this, using a smartphone for robot control would be more practical in terms of both cost savings and usability. These days, smartphones are practically an extension of the human body. As a result of this, new smartphones are manufactured daily that are stronger, faster, better at performing, and more durable. Additionally, they greatly simplify human life by including capabilities like Wireless Fidelity, Bluetooth (BT), Global Positioning System, cameras, and different detectors [2]. Applications (app) created using these features provide various functions by interacting with things and taking care of users' individual needs. These devices, made by various firms, run on a variety

Öz

Bu çalışmada, farklı alanlarda kullanılmak üzere tasarlanan mobil robotların otonom kontrolü için gerekli referans değerleri sağlamak ve robot bilgilerini toplamak amacıyla Android tabanlı bir uygulama tasarlanmıştır. Bu uygulama ile robot kontrolü için herhangi bir ek donanıma veya bilgisayara ihtiyaç duymadan herhangi bir Android cihaz ile robot kontrol parametreleri sağlanabilmektedir. Uygulama, manuel kontrolün yanı sıra otonom kontrol için bir arayüz sunmakta ve robotun istenilen hedeflere gidebilmesi için robota üç koordinat noktası ve yaklaşım mesafesini gönderebilmektedir. Ayrıca robot hareket etmeye başladığında Bluetooth üzerinden Android cihaza anlık veri alınabilmekte ve robotun anlık konumu cihaz ekranında koordinat düzleminde görüntülenmektedir. Çalışmanın etkinliğini ortaya koymak amacıyla, prototipi gerçekleştirilen robotun PID ve FL kontrolör ile otonom kontrolü, Android cihazdan alınan referanslar doğrultusunda gerçekleştirilmiştir. Aynı zamanda robot hakkında anlık bilgiler Excel dosya uzantısında Android cihaz üzerinde toplanmış, mail yoluyla bilgisayara aktarılmış ve verilerin grafikleri çizdirilmiştir. Çalışma, robotların yalnızca manuel olarak değil, aynı zamanda otonom kontrolünün de Android cihazlar yardımı ile yapılabileceğini göstermiştir.

Anahtar kelimeler: Android uygulama, MIT App Inventor, Robot kontrol

of operating systems, including Android, iPhone Operating System and Windows Operating System. Among them, the most widely used operating system is Android. Apps built for Android, therefore, appeal to an expanded audience.

These days, one of the places where Android apps are made is managed by the Massachusetts Institute of Technology (MIT), and it is called MIT App Inventor, which was originally made available via Google Business [3]. A web-based, open-source development tool, MIT App Inventor allows users to design and develop fully working Android apps while giving them a graphical user experience [4, 5]. Google Blockly was the foundation [6] and employs a block-based programming language derived from languages like StarLogo TNG [7] and Scratch [8, 9]. In the last decade, as well as to develop apps such as solutions for automation [10-13], air quality monitoring devices [14], apps for smartphones [15], training [16, 17], wellness [18, 19], monitoring and management of machinery [20-22],

* Sorumlu yazar / Corresponding author, e-posta / e-mail: atop@firat.edu.tr (A.Top)

Geliş / Received: 13.05.2023 Kabul / Accepted: 10.01.2024 Yayınlanma / Published: 15.04.2024

doi: 10.28948/ngumuh.1296747

navigation [23] communication [24], agriculture [25], etc., Android apps have been used also for robotic systems [26-42].

A three-degrees-of-freedom reconfigurable articulated robot has been developed by Rajpar et al. They used Arduino for controlling the robot's motions, and their own Android platform to send manual control commands [26]. Utilizing BT and a mobile Android app that enables control of an interactive robot using image or voice, to manually control the mobile robot (MR), Nádvrnk and Smutny did so. Through the app, they were able to observe the robot's proximity to obstacles using the data they collected from the sensors positioned in front of the robot [27]. Aldhalemi et al. used a smartphone and a gyroscope connected via BT to execute control of a two-wheeled self-balancing robot in real-time [28]. By sending voice commands via the Android app to the Arduino through BT, Harhipriya et al. were able to drive a robot vehicle [29]. Erşahin and Sedef used wireless technologies to control a robot remotely. On a tablet, they put the Android app they created. Through Wi-Fi, they transmitted the images to the tablet they were taking with the robot's camera, as well as the values they were setting for the robot's direction and speed to the robot [30]. For environmental safety and security, A robot with a camera, PIR, and gas sensor was remotely controlled via the Internet by Ramya and Palaniappan. In the Microsoft Visual Basic interface they created, they can take camera pictures and manually operate the robot to detect gas leaks and unlawful live entry [31]. An Android smartphone app was used by Meteab et al. to drive a robot vehicle [32]. To gauge temperature, humidity, and carbon monoxide levels in mines with hazardous and challenging working circumstances, Quadri et al. utilized a 3D printer to make a robot. To control this robot, an Android app has been created. When the measurement results were seen on a smartphone, they showed 100% accuracy up to a distance of 10 m [33]. Papcun et al. used BT and a Wi-Fi interface to connect the mobile robot Khepera III to a mobile smartphone. They performed face recognition by transferring the photo they captured with the robot's camera to the mobile device [34]. A powerful, affordable Robotic vacuum that can be operated by mobile devices was designed and produced by Eren and Doğan. They created an Android app so that the robot could be controlled remotely. With the app, the brush motor and vacuum may be operated while the robot is moving [35]. Using the Robot operating system (ROS), Kırılı et al. created mobile robot operations and ADs after mapping the environment with a six-wheeled mobile robot. The camera image and the computed map data are sent back to the device by the guided robot using the AD [36]. In a study conducted by Molnar et al., a mobile robot that stands on two wheels to give balance was designed and controlled. They used the ESP8266 microcontroller to carry out PID control. They gave users control over the mobile device and the PC via the web interface by creating a mobile app with MIT App Inventor [37]. For the Robosoccer game, Varga et al. created a mobile app that uses an accelerometer to control movement and direction by pairing players 1 and 2 via BT [38]. In the study in [39], the robot was operated by BT by projecting the

Android phone's screen onto the computer using the Mobizen program.

When the studies in the literature are analyzed, it is clear that simple apps have only been created for manual control or that ready-made apps are utilized to operate robots. Any app such as sending reference values to the robot for autonomous control, collecting real-time data, and displaying a graph of the robot's position on an AD, has not yet been created. Due to this, a mobile robot with four wheels whose prototype has been realized has been both autonomously and manually controlled in this study using an Android app that transfers data between the Android phone and the robot. MIT App Inventor's creator and blocking editor are used to create the interface and code of the app. Three separate reference points and approach distances can be given for the robot to travel to more than one target point. Additionally, information regarding how close to these locations the robot should get can be given over the same screen. Following the transmission of these references to the robot, data on the robot's linear speed, angular speed, and instantaneous wheel speed are transferred to the AD, where an instant graph is drawn and the data is gathered in an Excel file for transfer to another environment. To demonstrate the effectiveness of the app, the requested data was sent via the app to the mobile robot, whose position and speed control were carried out with PID (proportional-integral-derivative) and fuzzy logic controllers (FLC). The data received from the robot was collected as stated and stored on the device. It was discovered that the data transmission was carried out in both directions without any issues and that the robot control was available for the desired references after these data were sent to the MATLAB software, performance graphs of the robot were generated, and results were reviewed.

2 Material and method

2.1 MIT App Inventor

MIT App Inventor was originally made available by the Google firm and is now managed by MIT [43]. The design interface and the block editor are its two components. The area where screens and content are developed is the design interface seen in Figure 1.

Below are the sections of the design interface and their respective tasks [43]:

- In Palette: This is where the viewer area's components are inserted using the drag-and-drop approach.
- Designer: It's utilized to access the design space.
- Properties: This section allows you to change the component properties.
- Viewer: This section demonstrates how the components will be shown.

The block editor, depicted in Figure 2, is a workspace where programmers may graphically change their app's logic using blocks to describe the software [44].

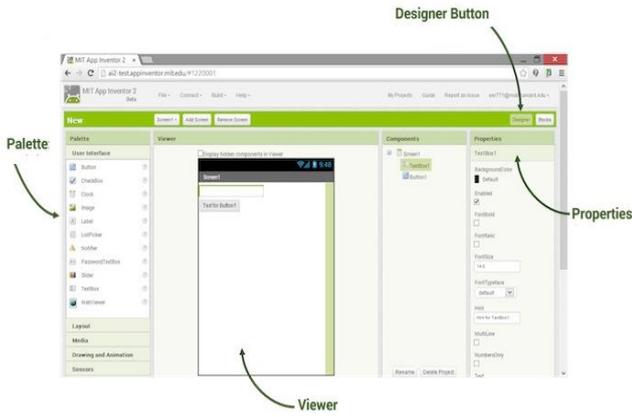


Figure 1. MIT App Inventor design interface

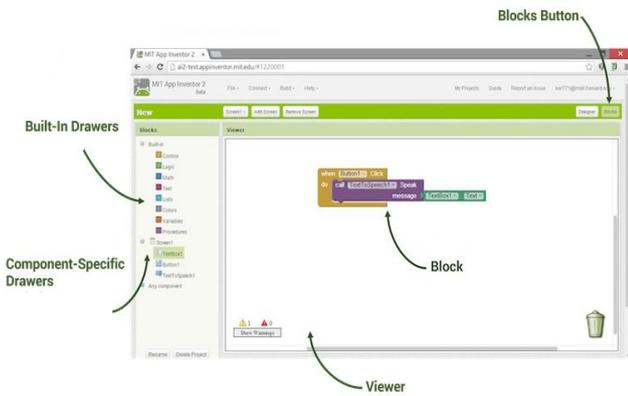


Figure 2. MIT App Inventor block editor

The following are the sections and their functions in the block editor interface:

- **Component-Specific Drawers:** This is the section that contains the code blocks that can be applied to the components on the design screen.
- **Block-In Drawers:** This is the section where the general code blocks are located.
- **BlocksButton:** It is the button to open the block editor.
- **Blocks Viewer:** It is the area where the code blocks can be dragged and dropped.
- **Block:** Code blocks used for software [45].

MIT App Inventor has been applied to numerous apps in recent years, including systems for automation [46], intelligent toggles [47], games and tests [48], smart home management [49-52], and teaching [53].

2.2 Mobile robot

The mobile robot, which will be controlled with the Android app, has a 4-wheel construction with a differential driving mechanism, each wheel powered by a 12V DC motor. Figure 3 shows the prototype of this mobile robot as well as its 3D designs.

Two Sparkfun Monster Moto Shields with PWM control are used to control the motors' speed. The Arduino Due microcontroller is sending these PWM values. Reference and approach distance values sent from the designed Android app are sent to Arduino via BT 4.0 HM-10 module. According to these transferred values, the autonomous

control of the robot is performed. In addition, while the robot is in motion, the robot's data is transferred to the AD via the same BT module, and the instant motion graph of the robot is drawn on the AD screen and the data is collected in the Excel file at the same time. Figure 4 shows the design of the architecture of these procedures.



Figure 3. Mobile robot (a) 3D drawing view and (b) Realized view

2.3 Control software

While manual control of the robot is carried out according to PWM values without the need for any algorithm, PID and fuzzy logic algorithms are used for autonomous control.

PID controllers, sometimes referred to as proportional, integral, and derivative (PID) controllers, are widely used to control DC motor speed. In nonlinear control systems, the PID controller is thought to be the technique that is used the most [54]. It provides the most straightforward and effective solution for a variety of control issues, addressing both steady-state and transient reactions. Equation (1) "gain notation" is typically used to write the transfer function.

$$T(s) = K_p + K_i \frac{1}{s} + K_d s \quad (1)$$

Where K_i is the integral gain, K_d is the derivative gain, and K_p is the proportional gain. In this study, it is used to control the reference speed values given to the motors.

As shown in Figure 5, the fuzzy logic controller (FLC) is a control algorithm that generates outputs that can be utilized in the physical world after processing input data that is extracted from the physical world and whose value is known.

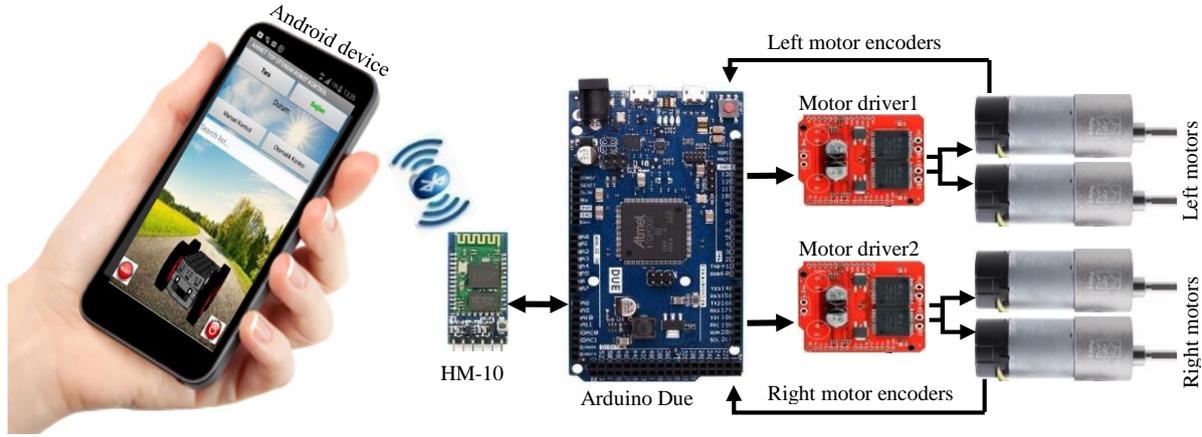


Figure 4. Robot control architecture

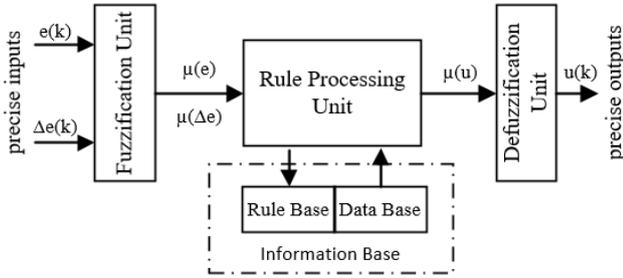


Figure 5. The general structure of the FLC

This control scheme consists of four key parts. these: the rule basis, the inference, the defuzzification units, and the fuzzification unit. Mamdani-type fuzzy logic was employed in this study to execute position control and path planning following the destination coordinates given to the robot by the Android app. The controller has two input variables and two output variables. As input variables, the distance error and angle error to the coordinate point where the robot should move are given. The output variable is the linear velocity and angular velocity values that the robot must achieve instantaneously.

When the general workflow diagram in Figure 6 (a) is examined, the position and angle error values found by first taking the difference between the reference position information coming from the AD and the current position information are applied as input to FLC. Reference motor speeds are calculated by substituting the reference linear and angular velocities of the robot obtained from the output of this controller into the inverse kinematic equations given in Equations (2-3). In line with these speeds, instantaneous robot speeds are calculated again with the forward kinematic equations in Equations (4-5) by utilizing the instantaneous speed information of the PID-controlled DC motors and these are converted into position information.

$$w_L = \frac{1}{R} \cdot (V - W \frac{L}{2}) \quad (2)$$

$$w_R = \frac{1}{R} \cdot (V + W \frac{L}{2}) \quad (3)$$

$$V = \frac{R}{2} \cdot (w_R + w_L) \quad (4)$$

$$W = \frac{R}{L} \cdot (w_R - w_L) \quad (5)$$

3 Android application

The MR is designed to be controlled in two different ways: manually and autonomously, so the Android app interface is divided into three sections to reduce complexity and provide visibility. The initial screen, the autonomous control screen, and the control screen for manual operations are these components. Figure 6 (b) shows a general workflow diagram for the app.

When the app is first launched, the Home Screen appears shown in Figure 7 (a). The AD and the robot's BT connection are shown on this screen. In addition, there are buttons for the selection of manual or autonomous control screen and choice of reset and shutdown

3.1 Manual control screen

The screen where manual control of the MR is offered is known as the "Manual Control Screen". There are buttons for direction, speed change and indicators, return, fan, and lights on the manual control screen shown in Figure 7 (b). One can control the robot's movement and alter its pace, and its fan and headlamps can be turned on and off using these buttons.

It is based on sending the string data type information sent from the AD to Arduino when the relevant button is pressed. This information was read in Arduino and a command was written for the desired task. Motor speed values are sent in PWM and these values are sent to the motor driver to control the robot.

It can be used when the user wants to use the robot manually, or it can be used to bring the robot to its starting position with autonomous control.

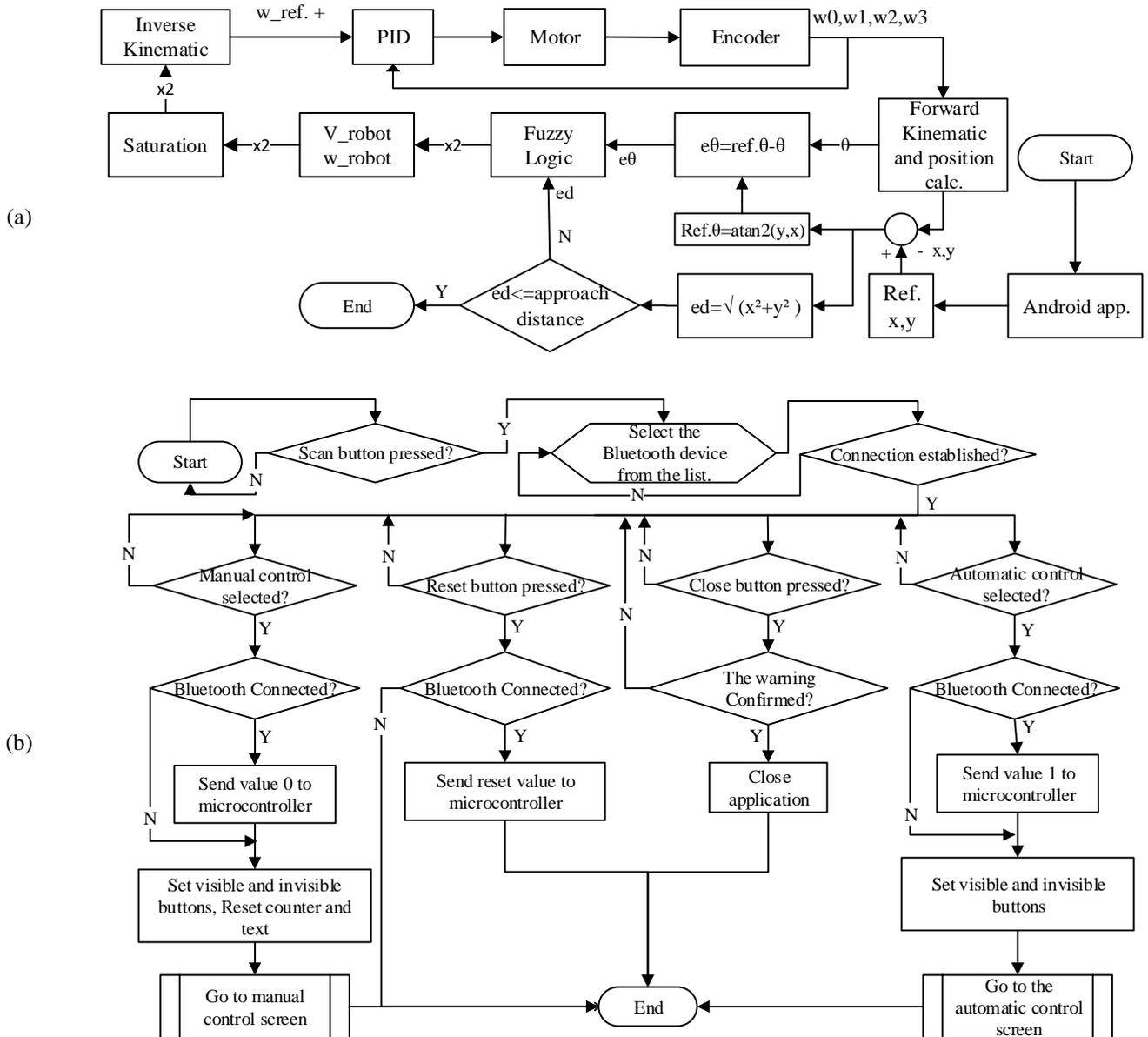


Figure 6. General workflow diagram (a) autonomously control software and (b) Android application software

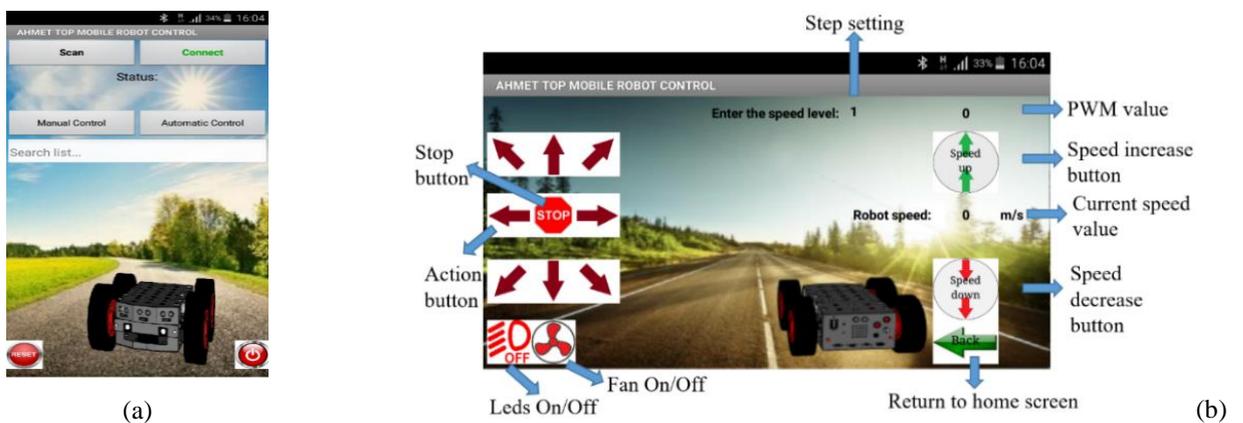


Figure 7. Android application (a) Home screen and (b) Manual control screen

3.1 Autonomous control screen

Neural Networks, Fuzzy Logic, and so forth. Algorithms can be used to control the robot's location and speed. Sending the desired position and approach distance information for the robot to the robot as a reference value is necessary once the control algorithms have been loaded into the robot processor. This procedure is likewise carried out on the Android application's autonomous control panel. Three input, deletion, and confirmation buttons for reference coordinate points and a back button are present on this screen, as shown in Figure 8 (a). Unlike manual control, no direction or speed information is given to the robot, only the desired coordinate point and lookahead are given. Control algorithms determine the speed and direction of the robot.

The user can enter the reference x and y axes and the robot's desired approach distance to the aim point in meters. In addition, he/she can enter three reference points $x_1, y_1, x_2, y_2, x_3,$ and $y_3,$ as well as specify two targets by giving the zero value to the first reference point or a single target by giving zero values to the two reference point. The approach distance also defines the robot's distance from the goal in meters. If at least one of the values to be entered on the screen is missing, the missing parts are marked with red as in Figure 9 and the parts where the value is entered are green ticks. Then, an error message is displayed stating that the values must be completed and the send button must be pressed. When the send button is pressed after all the values are entered, this data is transmitted to the Arduino, which processes these values as a reference. In addition, the coordinate axis in Figure 8 (b) is displayed, and the instant position of the robot and whether it reaches the given targets can be seen from here.

For example, by entering $x_1:2, y_1:2, x_2:4, y_2:2, x_3:6, y_3:0$ m reference values and 0.05 m approach distance in Figure 10 (a), these reference values are sent to the robot when the

send button is pressed. While the robot is performing its autonomous movement, the data collected on the AD via BT and the graphic in Figure 10 (b) are drawn on the screen.

The share files button at the bottom of Figure 11 (a) turns on when the robot has finished its motion. In addition to the instant position of the robot, information such as the robot's linear speed, angular speed, angle, angle error, reference, and actual speeds of the motors are collected in Excel format, and with this share button, it can be shared on platforms such as mail, WhatsApp, drive, etc., as seen in Figure 11 (b). In addition, as in Figure 11 (c), this information is saved to the AD. xls extension and the instant graphic as .jpg extension.

When sharing via mail, the Excel file containing the data and the explanation is automatically added to the mail as seen in Figure 12. The workflow diagram of the above processes is given in Figure 13.

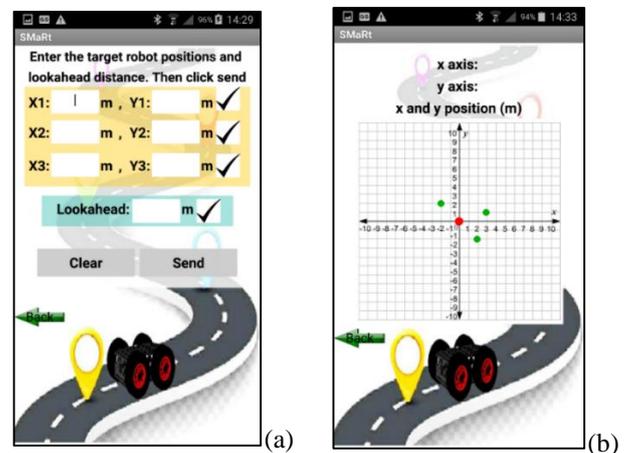


Figure 8. Application's (a) Autonomous control screen and (b) Instant location drawing screen

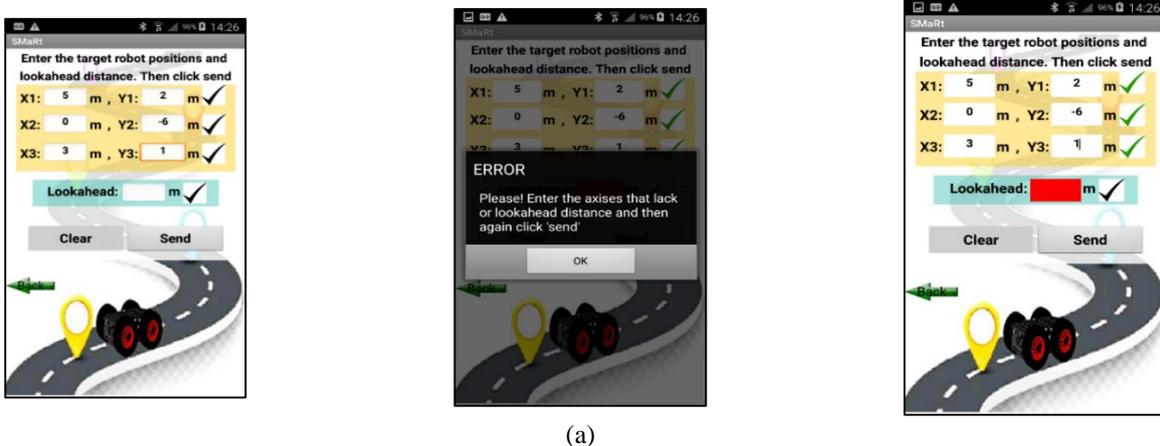


Figure 9. Error messages received in case of missing reference values (a) Only approach distance is missing, (b) Missing two coordinates and approach distance (c) All values are missing

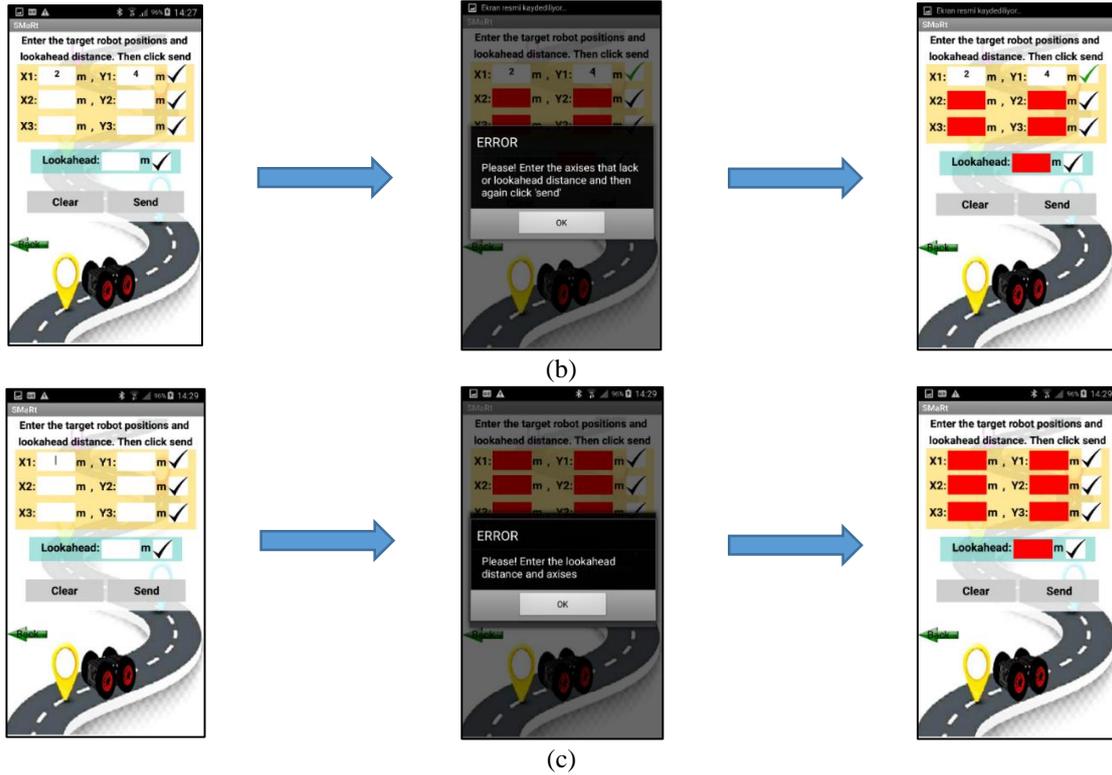


Figure 9 (continue). Error messages received in case of missing reference values (a) Only approach distance is missing, (b) Missing two coordinates and approach distance (c) All values are missing

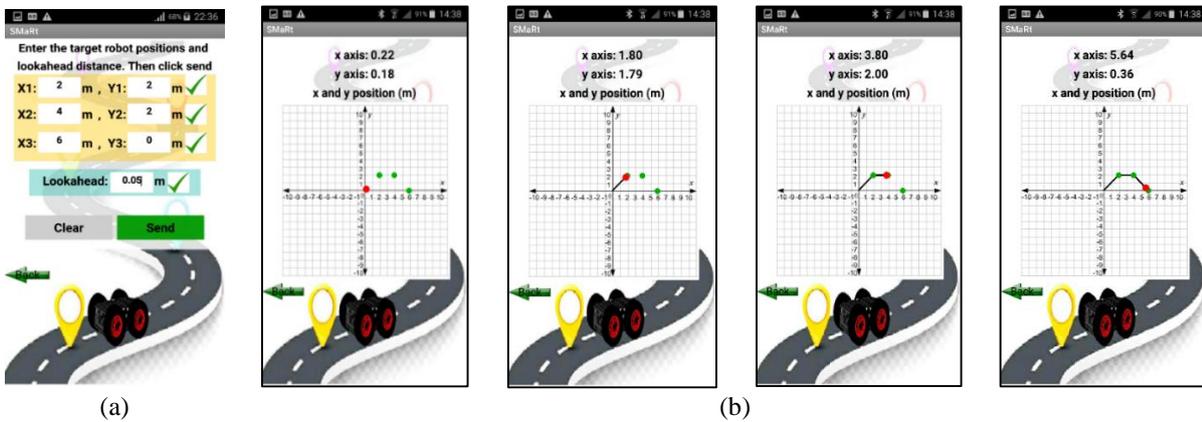


Figure 10. (a) Reference values sent to the robot and (b) Instant motion graph of the robot

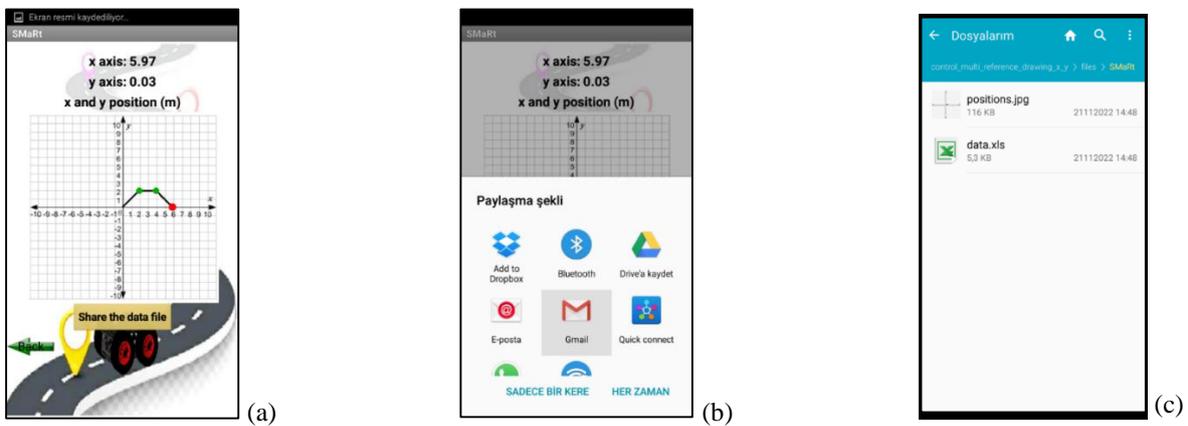


Figure 11. (a) File sharing button, (b) File sending options and (c) Saving files

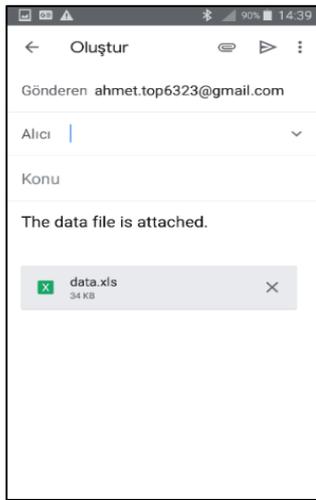


Figure 12. Sending the data file by e-mail

4 Results and discussion

The coordinate points (1,0), (1,2), (3,2) m, and the approach distance of 0.05 m were sent as references from the AD to the robot, respectively. During the autonomous movement of the robot, the data transferred to the AD were collected in Excel and transmitted to the computer by mail at the end of the movement. By transferring these collected data

to the MATLAB program, robot performances in Figure 14 and motor speed and errors in Figure 15 were obtained. Since the first target point is on the x-axis, the robot moved straight as in Figure 14 (a) without any rotational movement. It is also understood from the angle graph in Figure 14 (b) and the angular speed graph in Figure 14 (c) that there is no angular motion. Looking at the motor speeds in Figure 14 and the linear speed graphs in Figure 14 (d), the robot first started to move at 25 rpm and 0.275 m/s.

The robot changed direction twice during the whole motion, once to the right and once to the left, and during this movement, which in the experimental investigation lasted 22.61 seconds, the motors and the robot made the speed errors indicated in Table 1.

Looking at the target points, both right and left turns are expected to be 90°. However, since the given approach distance is 0.05 m when the first target is reached, it does not go exactly perpendicularly to the second target, as the turn was taken from behind the point. Therefore, the first angle to rotate was 1.54 radians, while the second angle was 1.63 radians. Likewise, it turned very close to the second target and when it completed its movement, the robot stopped at an angle of 0.093 radians. The instant location information drawn on the AD screen during the movement of the robot to these reference points is shown in Figure 16.

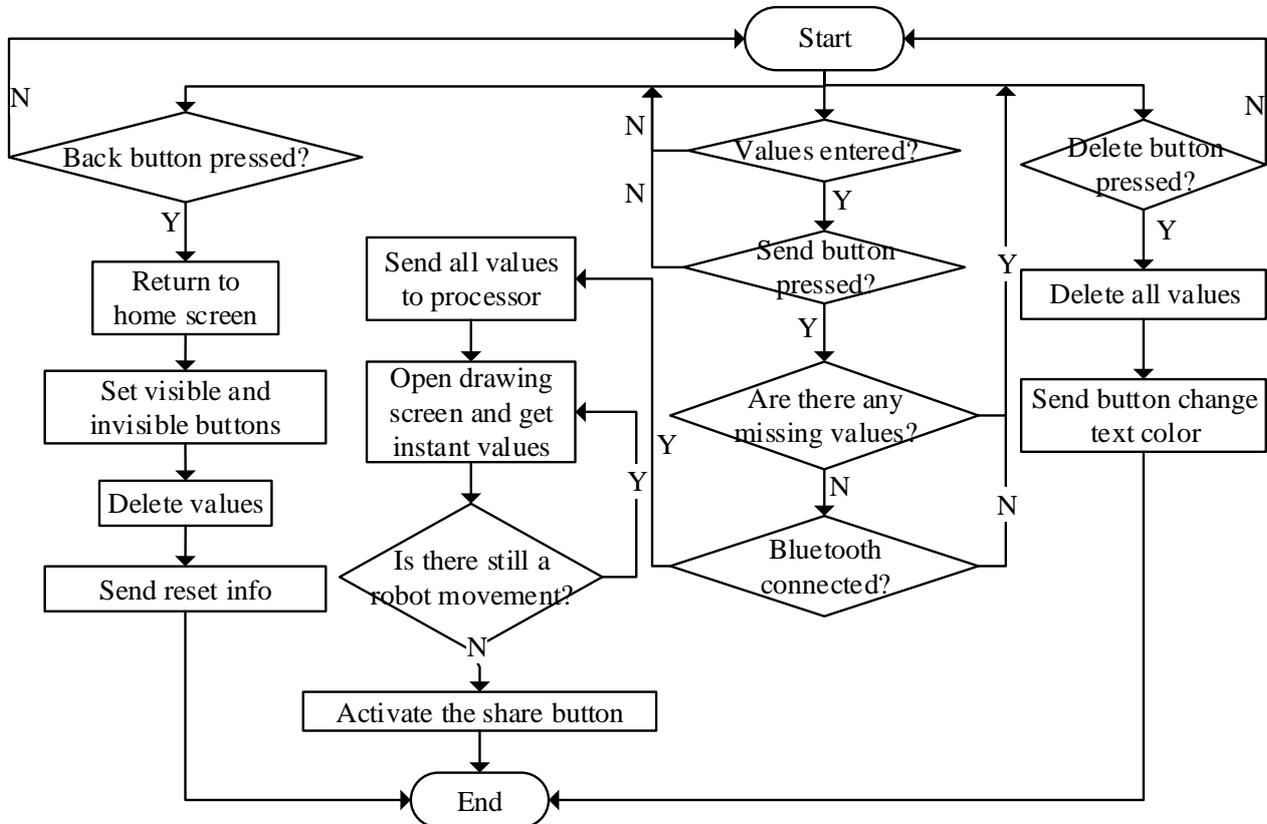


Figure 13. The autonomous control screen's workflow diagram

Table 1. Speed errors of motors and robot for $x_1:1, y_1:0, x_2:1, y_2:2, x_3:3, y_3:2$ m reference values and 0.05 m approach distance

	Motor1	Motor2	Motor3	Motor4	Robot
Mean absolute speed error (rpm)	0.311	0.262	0.269	0.305	---
Max. speed error (rpm)	12.648	12.467	12.393	12.691	---
Mean absolute angular speed error (rad/s)	---	---	---	---	0.0083
Max. angular speed error (rad/s)	---	---	---	---	0.355
Mean absolute linear speed error (m/s)	---	---	---	---	0.0025
Max. linear speed error (m/s)	---	---	---	---	0.073

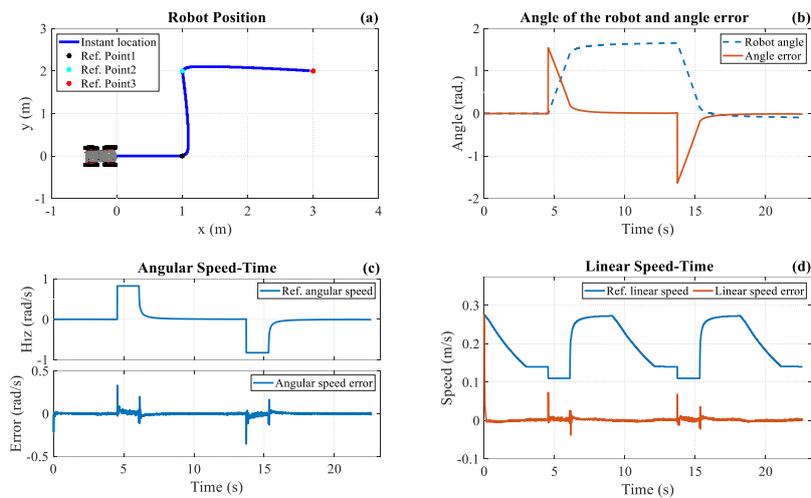


Figure 14. Experimental results for $x_1:1, y_1:0, x_2:1, y_2:2, x_3:3, y_3:2$ m reference and 0.05 m approach distance. Robot: (a) Location, (b) Angle and angle error, (c) Angular speed and error, (d) Linear speed and error

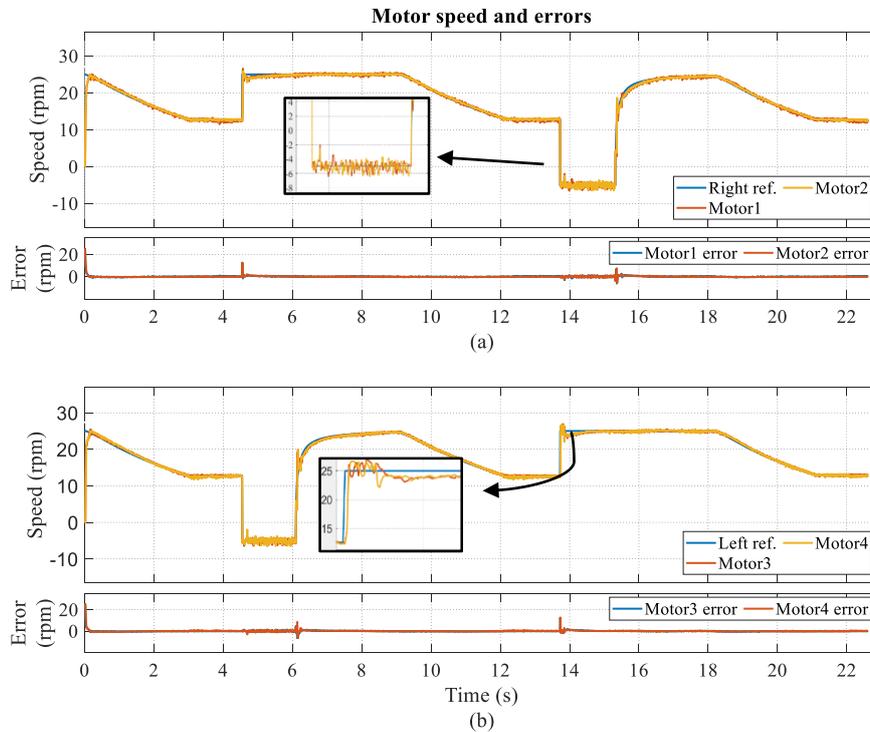


Figure 15. Experimental results for $x_1:1, y_1:0, x_2:1, y_2:2, x_3:3, y_3:2$ m reference values and 0.05 m approach distance. Robot: (a) Right motor speed and errors and (b) Left motor speed and errors

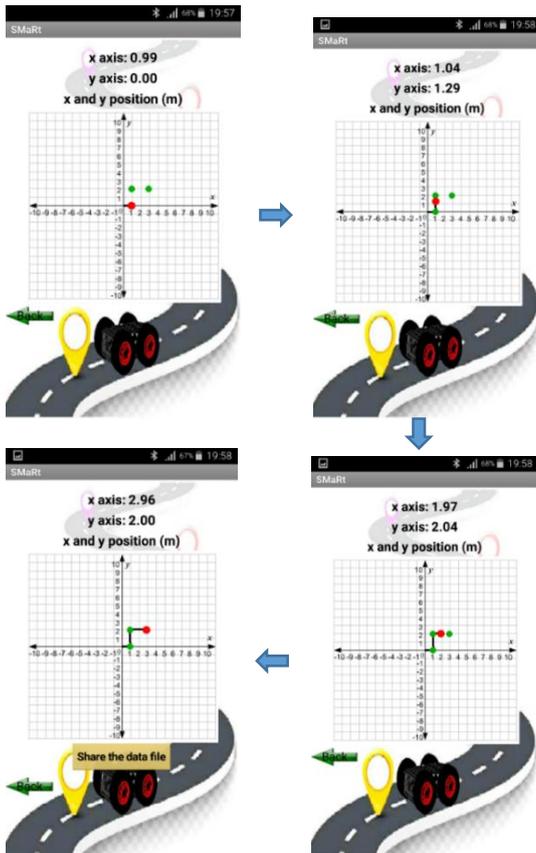


Figure 16. For the reference values $x_1:1, y_1:0, x_2:1, y_2:2, x_3:3, y_3:2$ m, an instant position graph

5 Conclusion

In this study, a new interface has been developed on the AD to send the reference and approach distance values required for autonomous control of the mobile robot that can be controlled manually with the AD. Thanks to this Android app, three different reference points where the robot is desired to go are sent to the robot processor via BT and autonomous control of mobile robots can be realized thanks to the algorithms written in the robot processor. After sending the reference values, the values such as linear speed, angular speed, angle of the robot, and the speed of the motors are collected by the processor and sent to the AD via BT when the robot starts to move. This data transferred to the AD is collected in Excel file format, as well as the instant location information of the robot is drawn on the screen. After the robot movement is completed, the Excel file where the data is collected can be shared on a desired platform. In the example carried out for this study, three different reference points are sent to the robot, and its control is ensured. The data collected during the movement were transferred to the computer by mail and the graphics were drawn in the MATLAB program. When these graphs are compared with the instantaneous position graphs drawn on the AD screen, it is seen that the robot data is transferred without error. Thanks to this app, both manual and autonomous control of robots can be realized. In manual

control, the robot can be moved to the desired position by the user at the desired speed, while in autonomous control, a PID controller and FLC are used for this work. In addition, since communication is provided via BT, With the module used, strong communication can be established up to a distance of approximately 60 meters. If a location farther than this distance is used as a reference, connection interruption may occur. Furthermore, if the speed limits of the robot are increased, the sampling time must also be increased to receive information more synchronously.

Funding

This study was supported by the Firat University Scientific Research Projects Unit with the project number TEKF.19.07.

Code availability

It can be provided if requested personally through the corresponding author's mail.

Compliance with ethical standards

Conflicts of interest

The authors declare no conflicts of interest.

Data availability

All data generated or analyzed during this study are included in this published article

Similarity rate (iThenticate): 6%

References

- [1] T. Akilan, S. Chaudhary, P. Kumari, and U. Pandey, Surveillance Robot in Hazardous Place Using IoT Technology. 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 775-780, December 2020.
- [2] T. Mikolajczyk, H. Fuwen, L. Moldovan, A. Bustillo, M. Matuszewski, and K. Nowicki, Selection of machining parameters with Android application made using MIT App Inventor bookmarks. Procedia Manufacturing, 22, 172-179, 2018. <https://doi.org/10.1016/j.promfg.2018.03.027>.
- [3] S. Hong, and Y. Hwang, design and implementation for iort-based remote control robot using block-based programming. Issues in Information Systems, 21 (4), 317-330, 2020. https://doi.org/10.48009/4_iis_2020_317-330.
- [4] S. C. Pokress, and J. J. D. Veiga, MIT App Inventor: Enabling personal mobile computing, 2013. arXiv preprint arXiv:1310.2830.
- [5] A. J. Colter, Evaluating and improving the usability of MIT App Inventor. Doctoral dissertation, Massachusetts Institute of Technology, USA, 2016.
- [6] N. Fraser, Blockly: A visual programming editor. URL: <https://code.google.com/p/blockly>, 42. Accessed: 10 May 2023.

- [7] A. Begel, and E. Klopfer, Starlogo TNG: An introduction to game development. *Journal of E-Learning*, 53, 146, 2007.
- [8] M. Resnick, J. Maloney, A. Monroy-Hernández, N.Rusk, E. Eastmond, K. Brennan, and Y. Kafai, Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67, 2009.
- [9] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, The Scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 1-15, 2010.
- [10] A.E. Amoran, A.S. Oluwole, E.O. Fagorola, and R.S. Diarah, Home automated system using Bluetooth and an android application. *Scientific African*, 11, e00711, 2021, <https://doi.org/10.1016/j.sciaf.2021.e00711>.
- [11] A. K. K. A. Raheem, Bluetooth-Based Smart Home Automation System using Arduino UNO Microcontroller. *Al-Mansour Journal*, 27, 2017.
- [12] Y. Liu, and D. R. A. Uthra, Bluetooth-Based Smart Home Control, and Air Monitoring System. *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 11(5), 2020.
- [13] T.Adiono, S.F. Anindya, S. Fuada, K. Afifah, and I. G. Purwanda, Efficient Android software development using mit app inventor 2 for Bluetooth-based smart home. *Wireless Personal Communications*, 105(1), 233-256, 2019, <https://doi.org/10.1007/s11277-018-6110-x>.
- [14] K.D. Purkayastha, R.K. Mishra, A.Shil, and S.N. Pradhan, IoT-Based Design of Air Quality Monitoring System Web Server for Android Platform. *Wireless Personal Communications*, 1-20, 2021, <https://doi.org/10.1007/s11277-021-08162-3>.
- [15] R. Robianto, Analysis Of Android-Based Mobile Blocking Application Design Using MIT App Inventor. *Jurnal Ipteks Terapan*, 14(1), 1-11, 2020, <http://doi.org/10.22216/jit.2020.v14i1.5082>.
- [16] M.F. Areed, M.A. Amasha, R.A. Abougalala, S. Alkhalaf, & D. Khairy, Developing gamification quizzes based on an android app: the impact of asynchronous form. *Education and Information Technologies*, 1-22, 2021 <https://doi.org/10.1007/s10639-021-10469-4>.
- [17] M. Z. Asghar, I. Sana, K. Nasir, H. Iqbal, F.M. Kundi, & S. Ismail, Quizzes: Quiz application development using Android-based MIT APP Inventor platform. *International Journal of Advanced Computer Science and Applications*, 7 (5), 2016 <https://doi.org/10.14569/IJACSA.2016.070508>.
- [18] M. Singh, G. Singh, J. Singh, & Y. Kumar, Design and Validation of Wearable Smartphone-Based Wireless Cardiac Activity Monitoring Sensor. *Wireless Personal Communications*, 1-17 2021 <https://doi.org/10.1007/s11277-021-08219-3>.
- [19] A. Verma, S.B. Amin, M. Naeem, & M. Saha, Detecting COVID-19 from chest computed tomography scans using AI-driven android application. *Computers in biology and medicine*, 143, 105298, 2022.
- [20] M. S. Ahsan, S. Das, & H. Mobarak, Android App-based Bluetooth controlled Low-cost Cloth Folding Machine. In 2020 IEEE Region 10 Symposium (TENSYP), pp. 170-173, IEEE, 2020, <https://doi.org/10.1109/TENSYP50017.2020.9231012>
- [21] T. Mikolajczyk, H. Fuwen, L. Moldovan, A. Bustillo, M. Matuszewski, & K. Nowicki, Selection of machining parameters with Android application made using MIT App Inventor bookmarks. *Procedia Manufacturing*, 22, 172-179, 2018, <https://doi.org/10.1016/j.promfg.2018.03.027>.
- [22] R. Kleinwort, T. Semm, P.M. Falger, & M.F. Zaeh, Integration of an Android Application into the Learning Factory for Optimized Machining. *Procedia Manufacturing*, 23, 9-14, 2018.
- [23] M. Samuel, N. Nazeem, P. Sreevals, R. Ramachandran, & P. Careena, Smart indoor navigation and proximity advertising with Android application using BLE technology. *Materials Today: Proceedings*, 43, 3799-3803, 2021.
- [24] P. Rajeswari, & P. Anbalagan, Design and deployment of Android-based mobile application for performance analysis of microstrip patch antenna. *Microprocessors and Microsystems*, 77, 103111, 2020.
- [25] A. Sherafati, K. Mollazade, M. K. Saba, & F. Vesali, TomatoScan: An Android-based application for quality evaluation and ripening determination of tomato fruit. *Computers and Electronics in Agriculture*, 200, 107214, 2022.
- [26] A. H. Rajpar, A. Eladwi, I. Ali, & M.B. Ali Bashir, Reconfigurable Articulated Robot Using Android Mobile Device. *Journal of Robotics*, 2021. <https://doi.org/10.1155/2021/6695198>.
- [27] J. Nádorník, & P. Smutný, Remote control robot using Android mobile device. In *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*, pp. 373-378, 2014, <https://doi.org/10.1109/CarpathianCC.2014.6843630>.
- [28] A.A. Aldhalemi, A.A.Chlaihawi, and A. Al-Ghanimi, Design and Implementation of a Remotely Controlled Two-Wheel Self-Balancing Robot. In *IOP Conference Series: Materials Science and Engineering*, Vol. 1067, No. 1, p. 012132, 2021.
- [29] G. Haripriya, Y. Divyavani, A.R. Devi, M. Sravan, & K. Reddy, Arduino Based Voice Controlled Robot. *Complexity International*, 25(01), 2021.
- [30] G. Ersahin, & H. Sedef, Wireless mobile robot control with tablet computer. *Procedia-Social and Behavioral Sciences*, 195, 2874-2882, 2015 <https://doi.org/10.1016/j.sbspro.2015.06.411>.
- [31] V. Ramya, & B. Palaniappan, Web-based embedded Robot for safety and security applications using Zigbee. *International Journal of Wireless & Mobile Networks*, 4(6), 155, 2012.
- [32] W. K. Meteab, H. T. S. ALRikabi, S. A. H. Al Sultani, & I. A. Aljazeera, Controlling and Monitoring a Robot-Car Based on Smart Phone Applications. In *IOP Conference Series: Materials Science and Engineering*

- 1094 (1), p. 012096, 2021 <https://doi.org/10.1088/1757-899X/1094/1/012096>.
- [33] I. Qadri, A. Muneer, & S.M. Fati, Automatic robotic scanning and inspection mechanism for mines using IoT. In IOP Conference Series: Materials Science and Engineering, 1045 (1), p. 012001, 2021 <https://doi.org/10.1088/1757-899X/1045/1/012001>.
- [34] P. Papcun, I. Zolotova, & K. Tafsi, Control and teleoperation of robot Khepera via Android mobile device through Bluetooth and wifi. IFAC-PapersOnLine, 49 (25), 188-193, 2016 <https://doi.org/10.1016/j.ifacol.2016.12.032>.
- [35] A. Eren, H. Doğan, Design and implementation of a cost-effective vacuum cleaner robot. Turkish Journal of Engineering, 6 (2), 166-177, 2022 <https://doi.org/10.31127/tuje.830282>.
- [36] A. Kırılı, M. Dilaver, F. Çakmak, Mobile Robot Control With Android Device Sensors By Using Ros. Mugla Journal of Science and Technology, Vol 3, No 1, Pages 31-34, 2017, <https://doi.org/10.22531/muglajsci.272475>.
- [37] J. Molnár, Š. Gans and O. Slavko, Design and Implementation Self-Balancing Robot. 2020 IEEE Problems of Automated Electrodrive. Theory and Practice (PAEP), Kremenchuk, Ukraine, pp. 1-4, 2020, <https://doi.org/10.1109/PAEP49887.2020.9240815>.
- [38] J. Varga, L. Vargovčík, & V. Baláž, Design of Mobile Application for Controlling Robosoccer via Bluetooth. Journal of Automation and Control, 5 (2), 46-49, 2017, <https://doi.org/10.12691/automation-5-2-3>.
- [39] R. K. Fahmidur, H.M. Munaim, S.M. Tanvir, & A.S. Sayem, Internet controlled robot: A simple approach. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 1190-1194, 2016.
- [40] J. Azeta, C.A. Bolu, D. Hinvı, A.A.Abioye, H. Boyo, P. Anakhu, & P. Onwordi, An android-based mobile robot for monitoring and surveillance. Procedia Manufacturing, 35, 1129-1134, 2019.
- [41] E. Vasey, M.S. FakhrHosseini, Z. Zheng, C.H. Park, A. Howard, & M. Jeon, Development and usability testing of a remote control app for an interactive robot. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 61, No. 1, pp. 808-812, Sage CA: Los Angeles, CA: SAGE Publications, 2017.
- [42] S. Ullah, Z. Mumtaz, S. Liu, M. Abubaqr, A. Mahboob, & H.A. Madni, Single equipment with multiple applications for an automated robot-car control system. Sensors, 19(3), 662, 2019.
- [43] Massachusetts Institute of Technology, Designer and Blocks Editor, <https://Appinventor.mit.edu/explore/designer-blocks>. 10 May 2023.
- [44] E.W. Patton, M. Tissenbaum, & F. Harunani, MIT app inventor: Objectives, design, and development. In Computational thinking education, pp. 31-49, Springer, Singapore, 2019.
- [45] M.F. Pekyürek, Z. Sağlam, & A.B. Ustun, MIT App Inventor ve Android Studio kullanılarak tasarlanmış mobil uygulamanın performans karşılaştırması. Bilgi ve İletişim Teknolojileri Dergisi, 2 (2), 161-181, 2020.
- [46] P.B. De Moura Oliveira, Teaching automation and control with App Inventor applications. In 2015 IEEE Global Engineering Education Conference (EDUCON), pp. 879-884, 2015.
- [47] A. Howedi, & A. Jwaid, Design and implementation prototype of a smart house system at low cost and multi-functional. In 2016 Future Technologies Conference (FTC), pp. 876-884, 2016.
- [48] M.Z. Asghar, I. Sana, K. Nasir, H. Iqbal, F.M. Kundi, & S. Ismail, Quizzes: Quiz application development using the Android-based MIT APP Inventor platform. International Journal of Advanced Computer Science and Applications, 7 (5), 2016.
- [49] D. Sullivan, W. Chen, & A. Pandya, Design of remote control of home appliances via Bluetooth and Android smartphones. In 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 371-372, IEEE, 2017.
- [50] S. Kannapiran, & A. Chakrapani, A novel home automation system using Bluetooth and Arduino. International Journal of advances in computer and Electronics Engineering, 2 (2), 41-44, 2017.
- [51] S. S. Prayogo, T. Saptariani, & N.S. Salahuddin, Rancang Aplikasi Android Pengendali Mobil dan Kamera Menggunakan APP inventor. Seminar Nasional Aplikasi Teknologi Informasi (SNATI), Vol. 1, No. 1, 2015.
- [52] T. Adiono, S.F. Anindya, S. Fuada, K. Afifah, & I.G. Purwanda, Efficient Android software development using MIT App Inventor 2 for Bluetooth-based smart home. Wireless Personal Communications, 105(1), 233-256, 2019.
- [53] M. Karakus, S. Uludag, E. Guler, S.W. Turner, & A. Ugur, Teaching computing and programming fundamentals via App Inventor for Android. International Conference on Information Technology Based Higher Education and Training (ITHET) pp. 1-8. 2012.
- [54] M. Kushwah, & A. Patra, Tuning PID controller for speed control of DC motor using soft computing techniques-A review. Advance in Electronic and Electric Engineering, 4(2), 141-148, 2014.

