Smartphone-Based Adaptive Remote Control Interface for Six-Wheeled Mobile Robot

Mikhail Mironov[1,2], Alexey Kashevnik[1]

[1]SPIIRAS, St.Petersburg, Russia
[2]ITMO University, St.Petersburg, Russia
mmironov@niuitmo.ru, alexey@iias.spb.su

***Abstract*—The paper describes the process of developing an adaptive control interface for a remote control for a six-wheeled mobile robot based on the smart spaces technology. Design approaches of the control interfaces for unmanned ground vehicles have been overviewed. The architecture of a robot control system and Android application have been presented. The user interface for a mobile application adapts to a robot's capabilities, the user needs and task context. The study represents the control process involving the delegating of responsibilities among the group members. It can be applied to a six-wheeled mobile robot.**

I. INTRODUCTION

Modern robots have more computing power and smaller dimensions than their predecessors in virtue of technologically advanced materials and high-performance microchips. In combination with uninterrupted wireless signal receiver robots can perform complex tasks without direct user intervention. All these factors have resulted in the increase of unmanned [1] and cyber-physical [2] systems development.

The scope of such robots implementation can be spread from anti-terrorist operations to the exploration of other planets. The main factor of this universality is the ability to operate under conditions which a human cannot withstand. Often, robots of various sizes and designs are used an unmanned ground vehicle (UGV). However, this kind of robots is often used in everyday life at present, not only in special applications. The widespread use of robots has become possible due to their production cost decrease and the research in the field of robotics engineering. In this project the authors use a mobile robot with a great number of different drives providing its mobility. Such a design allows to simulate various operating scenarios. The robot has an ability to overcome simple obstacles of varying heights with a flat top and different-component complex obstacles, such as stairs.

The inability to automate all processes of the robot functioning requires the presence of the remote control operator. Moreover, the increased number of robot's features and functions transformed control devices in complex and cumbersome sets of levers and buttons. Thereby, to use a control device an operator needs additional training and skills.

Typical radio controls can be replaced by more versatile devices - smartphones. With the progress of microelectronics smartphones have become one of the most popular devices in the world today. The popularity led to billions of sales all over the world each year [3]. These figures suggest they have replaced mobile phones. The mobile technology market growth transformed smartphones into a universal assistant for the interaction with different physical devices and the Internet services [4], [5]. Based on this we infer that modern smartphones are likely to cope with the role of a universal and multipurpose remote control device. Moreover, the fact that each year the computing power increases should be considered. The rate of performance boost over the past several years allows modern smartphones to compete with PCs from the middle of 00's [6]. It is not just about the power of the CPU, but also a graphic component. Modern smartphone screens can reach Full HD (1920×1080) resolution and 6'' size. These characteristics make it possible to display a wide variety of robot controls on the screen. In this way the using of a smartphone as a control remote device can solve the problem with a radio control size and weight. However, control elements redundancy problem is still actual.

Usually, to solve a specific set of tasks the user requires a limited set of tools and features. Therefore, the successful arrangement of these functions control elements increases the operator's speed of task execution. In the case of the control interface design we can arrange the elements next to each other according to the frequency of use or the groups of operations. As a result we still get a controller with all features located on the console. There have been research works on control interface adaptation processes during the last decade [7], [8]. The processes of physical control adaptation may result in the decrease of the number of managed device features or the complication of function using. The way to digitize the control elements enhances the number of options for a control interface adaptation. For example, it is possible to hide the most of functions and display them in case of the user request. First, it makes control interface user friendly excluding excessive diversity of elements. Second, it increases the productivity solving typical problems. Last, it lowers the risk of erroneous keystrokes. This approach can be called the context-based mapping of controls.

As mentioned above, there is a large number of controls associated with increasing functionality of mobile robots. Moreover, a robot configuration can vary greatly depending on use conditions and tasks. In some cases several robots connected with each other may act as UGV multiplying the number of different control tools. On the other hand, mechanisms are strongly dependent on the context in which the robot operates. Situational functional orientation simplifies control processes of multicomponent robots combining their multiple functional elements into a single interface access. For example, the combination of an access to several drives of the robot simplifies the solution of a problem to overcome freestanding obstacles higher than the radius of the robot wheels. In addition, this approach also helps to solve more complex tasks such as stair climbing. While working on the project for an Android smartphone an adaptive interface for multidrive UGV control system has been developed.

The rest of the paper is structured as follows. Related work is presented in Section II. Section III describes the task definition. The robot control system architecture is presented in Section IV. Section V presents an adaptive interface. Architecture of the mobile app showed in Section VI. Evaluation of the work described in Section VII. The results are summarized in Conclusion.

II. RELATED WORK

Recent smartphones have a wide range of built-in sensors: accelerometer, gyro, GPS, etc. This fact provides opportunities to implement various functions of control interfaces. The following is the overview of the articles in which mobile control interfaces were developed.

The most popular approach is the application of directional arrows [9]. Buttons in the form of pointing up and down arrows are responsible for the throttle commands (move forward and backward). The leftward and rightward icons control the steering of the subject. This approach is the easiest to learn for users due to its intuitive and logics. Moreover, it also has a number of drawbacks. The necessity to switch between buttons frequently makes it problematic to turn and maneuver quickly while the robot moves along a complex trajectory. Other control elements can be displayed by arrows or other schematic icons. However, in the first case the presence of several similar function elements can make the user confused and requires additional design in the second case. In addition, this approach shows its imperfection in a situation requiring simultaneous use of a number of functions.

Another embodiment of the control interface is the use of a virtual joystick. The approach increases the number of possible wheel steering angles, and the UGV maneuverability ipso facto grows up. This solution does not affect the intuitive of control as it is used in lighting regulators, motor power regulators, as well as in controlling various devices, machines, mechanisms and even spacecrafts [10]. The movement direction control includes only one user's hand and leaves more freedom for simultaneous manipulation of several functions, in contrast to the previous solution.

Another approach to the maneuvering problem is the use of a virtual steering wheel. In this case, the user can control the rotate direction by turning the steering wheel icon clockwise or counterclockwise. In practice, this approach has too redundant set of possible rotation angles, which is why the rotation of icon to the extreme values takes much time.

Using such elements as the vertical and horizontal sliders is also one of the approaches to the design of the control interface. Initially the sliders are in the middle position. The vertical member deflection is used to perform throttle command, while the horizontal one controls left and right turns. Slider release returns it to the original value. This approach is rarely used because of its non-intuitive.

Controlling by a built-in smartphone accelerometer is one of the most technologically advanced approaches [11]. Control interface software fixes a smartphone turns in both horizontal and vertical planes and gives commands for the robot to change direction or speed respectively. The presence of a gyroscope sensor in the device allows to control other manipulation operations such as raising one or another part of the robot. This approach is often used in first or third-person mobile games (Angry Birds Go, Asphalt). Nevertheless, players can switch to more classic control interface. The downsides of this approach include poor accuracy of the sensors, the need of their calibration and inconvenience of use in motion.

Described in [12] the use of a voice robot control also has advantages and disadvantages. On the one hand, this approach is the most convenient in situations with the operator’s hands occupied. On the other hand, it requires a complex filtering and recognition mechanism and strongly depends on the level of external noise.

Authors of [13] conducted an experiment in which 20 users were to pass a zigzag line by controlling a robot with 4 different combinations of controls. On the task completion time, the number of erroneous actions and feedback from examinees, the authors concluded the arrows approach for giving steering and throttle commands to be the most convenient and simplest for the user than other element combinations.

The most of mentioned control interfaces have their origin in first or third-person video games. The user controls the robot in the same way if the robot had a mounted camera with its video broadcasted on the smartphone. In this project, the video signal is not used, and the robot control interface is depicted in its profile view. This variant of control objects display allows to implement the hybrid approach. A set of arrow icons forms a joystick circle and permits to control each robot element separately. Partial or full functionality hiding is possible depending on the context and the user's commands. It simplifies the visual design of an interface and focuses the user's attention on the task at hand.

III. TASK DEFINITION

In this project the object to control is the prototype of a six-wheeled robot based on the mobile robotic platform MRP-100 (Fig. 1) manufactured in "ROCAD" Scientific and Technical Center [14]. UGV is a two-section platform with wheel-walking-propelled for the creation of mobile robots and mobile robotic systems on its base. The following tasks can be performed:

- visual, electronic, radiation, chemical intelligence of locations, industrial and residential premises, transport objects in any light, electronic mapping of intelligence results, transfer of intelligence data;
- remote radiological study of area;
- cleaning and decontamination of areas;

- assistance in conducting of anti-terrorist operations: production of electronic noise, smoke and special screens, delivery and use of non-lethal weapons, covert penetration to the captured and protected objects, keeping distracting fire, revealing the enemy firing points;
- guard of objects (patrolling the area or perimeter of the object, suppression of intrusion attempts on the object, the neutralization of intruders);
- implementation of complex technical operations;
- work in explosive and flammable environments [14].

Each axis of the platform is a separate unit with a set of drives. The central block is a link to which guiding and closing identical units are attached via tubing. These units have wheeled drives and can move away from the central unit to the length of the connecting tube.

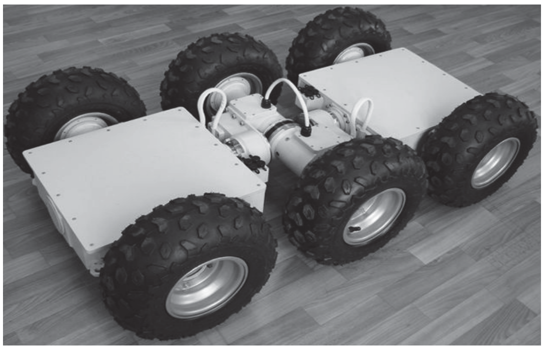

Fig. 1. A robot based on the mobile robotic platform MRP-100

In view of the entire set of drives and mounts, this all-wheel-drive robot can lower, lift and rotate the guide blocks (axes) as well as increase its length.

As a part of the research there has been a task to develop an adaptive interface based on the smartphone to solve the following problem. The operator sends a “move-forward” command to the robot from a smartphone. The robot starts scanning the area in front until it determines an obstacle ahead or is stopped by the user command. UGV stops in front of an obstacle and beeps upon its detection. This event is also displayed on the user interface with a schematic symbol. After that the user can either manually control all the actions of the robot or just give the "overcome the obstacle" command to it. In the last case, the robot attempts to execute a set of commands to perform this action in automatic mode. If the robot fails to overcome the obstacle it stops and beeps. In this case the user should manually notify the command sequence for the script execution. If an obstacle is overcome successfully UGV continues to move forward without stopping. This scenario simulates the robot's control operation in overcoming curbs, stones or other similar obstacles. Obstacle models have height that is more than the radius of the robot wheels. Smaller obstacle climbing does not involve the lifting gear.

Often, more than one users are needed to control complex and multi-component devices, such as a robot-sapper or moonwalker [15]. This project describes such a work scenario. The functionality of the robot can be divided into 2 groups of operations:

- moving (throttle commands and wheel rotation commands);
- manipulation (commands of lifting and lowering units).

As a result, it is possible to distribute a robot control functionality between three groups of users: the driver (only has an access to the first group of transactions); the manipulator (uses only the second group); and the expert (if the robot cannot overcome obstacles in the automatic mode, it sends a signal via the interface of this particular group. In this case, the user switches control to manual mode and helps the robot to perform a task combining groups of operations).

Groups of controls are shared between users based on the following principle: the first operator automatically gets access to all possible operation after connection to the robot. Thereby, he fulfills all the specified roles. If the second operator is connected, the first one can select one of the two control groups. The second user automatically gets the rights of another group. For example, the first connected user selects the driver role, and the last one gets the manipulator role. In a similar way, after the third user connection, the first two users are able to choose in turns which of the three roles to perform. The expert expects the need of manual control of obstacles overcome process. The robot creates a pattern of behavior based on the sequence of the user commands. Further, when an obstacle with similar dimensions is determined, UGV tries to overcome it automatically following this pattern. If the robot gets into that situation being under control of only two operators, the driver gains the access of manipulation and acts as an expert.

IV. THE ROBOT CONTROL SYSTEM ARCHITECTURE

Fig. 2 shows the general architecture of a robot control system. Smart-M3 platform is used in a system for the interaction between system elements. This platform implements the concept of smart spaces (SS) [16]. The system architecture can be divided into physical components and information ones to manage them. The first group includes robots and smartphones used to manage one or more of the UGV. Each physical component in the system corresponds to one information component at least - KP agent, which cooperates with the SS and other system components. Moreover, the information components include Smart-M3 platform and interaction services, which receive data from the SS and transmit it directly to robots through wireless LAN. Control process proceeds as follows: the user or group of users with a mobile application interface through the Internet connection or LAN is linked to the SS. All subsequent user commands are transmitted in the form of RDF triples and stored in the knowledge base. Robot’s KP agents with a subscriptions mechanism transmit commands to the interaction service of the one robot or to the whole group. KP agents independence from device system architecture provides a large variety of ways for the further development of the system and the number of interacting devices increase. As additional services can act stationary video camera, which determines the size and dimensions of obstacles around the robot, or an UGV additional module - quadrocopter or another unmanned aerial vehicle with a camera.

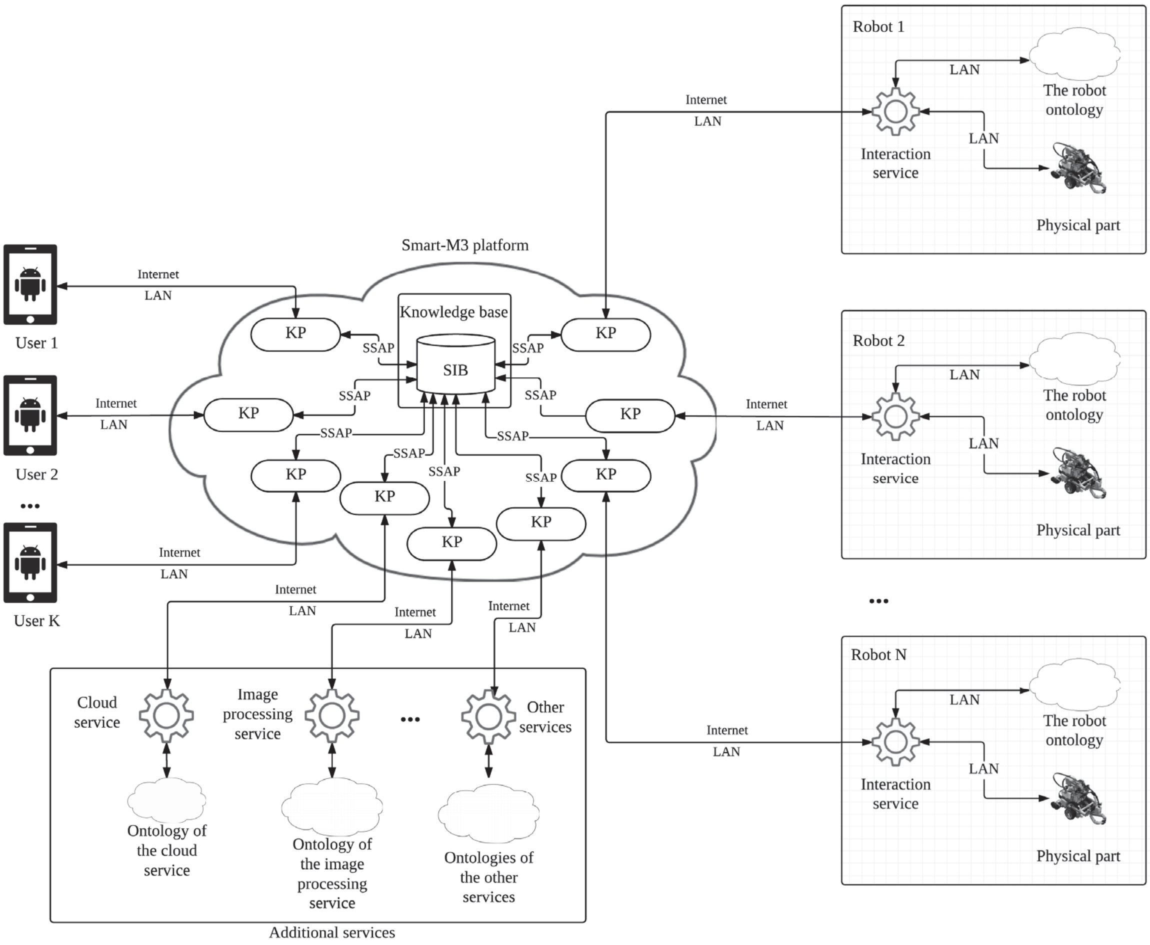

Fig. 2. Robot control system architecture

V. ADAPTIVE INTERFACE

The above-described approaches of control interfaces implementation are mostly common for developing applications for a specific robot with a fixed set of components and modules. That entails a lot of problems in case of adding or disabling various modules. For example, if the optional lifting mechanism is added to block, the robot does not have only to change its scheme of work, but also to implement in an application a new control object functional. This entails updating or reinstalling the application of a control interface. Creating a control interface for multi-axis robots of various modifications have been one of the major point of this project.

The robots may have different components and capabilities depending on the task. While the first SS connection interface subscribes the information about the robot and its set of components. Then if this data is changed, the interface dynamically displays the possible control options. Thus, the axis with no drive is displayed intuitively without direction arrows (forward, backward), as well as a non-lifting block without the up arrow.

Besides, the interface adapts to the user's needs. The lack of clutter arrows, which illustrate all the features of the robot configuration, provides the user friendly control interface displaying. If the user does not want to control all the drives individually and sets in motion the robot in parts, the interface provides the opportunity to budge the robot by pulling the head block in the right direction. In this case, the command is transmitted automatically to all existing drive units.

If the UGV with the ultrasonic sensor faced with an obstacle, it stops and notifies the user with a sound. Then the user can see the obstacles encountered in the conventional image of the interface. Thus, the control program and robot notify the person about the need to bring the situation under manual control and to make a decision. Therefore, it is possible to conclude that the adaptive interface has an ability to change depending on the robot capabilities, the user needs and the context (Fig. 3).

VI. ARCHITECTURE OF THE MOBILE APPLICATION

As an implementation platform of the adaptive control interface the world’s most popular mobile platform - Android OS has been selected. The development process of the mobile app that implements adaptive control interface took place in Android Studio IDE. This tool provides handy functionality to work with code, includes the emulator for project testing and built-in sets of app design.

The activity class is the basis of the graphical interface of Android apps. It displays app's visual activity and determines the actions that user can produce.

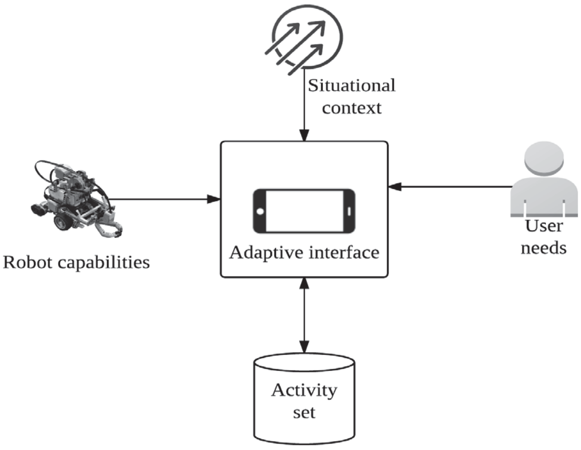

Fig. 3. Adaptive interface structure

IPInsert and Contol classes inherit system AppCompatActivity class in this project. Fig. 4 shows the main app classes, their basic methods and the links between them. IPInsert class implements activity of Smart-M3 server ip-address input screen. Methods of IPAddressValidator class validate the entered data for compliance ipv4 mask. If received ip-address is correct, the application checks the Internet connection on device and starts the process of SS connection. ListOfLastUse component shows a set of recently used addresses. This feature is implemented by using OS's built-in preferences SharedPreferences class. This way allows to save and retrieve data across user sessions, even if the application was killed.

SmartM3 class implements all possible methods to communicate with the SS and providing the interface for all application classes. Moreover, this class is processes all errors that occur during the work with Smart-M3 server, and periodically checks the connection. For the reason of Android OS security policy prohibits data transfer and other network operations through the main thread, this class operating in separate.

After a successful connection to the SS server, the user goes to the main screen of the application which is implemented in the Control class. In first connect paintBlock method draws on the screen Image classes, based on data received from the SS. These entities keep a vector image in the format of the base element container ImageView, and its endpoints coordinates. Every Image class initialization has clickListener with an individual set of operations. In other words, each element of the screen is interactive. An array of current robot blocks forming on the basis of embedded in app templates. Each block has a unique index and set of functions. Image arrow class implements elements of functions control. Methods of this class can hide and show these elements. Thereby, control elements are shown depending on the user's taps on unit.

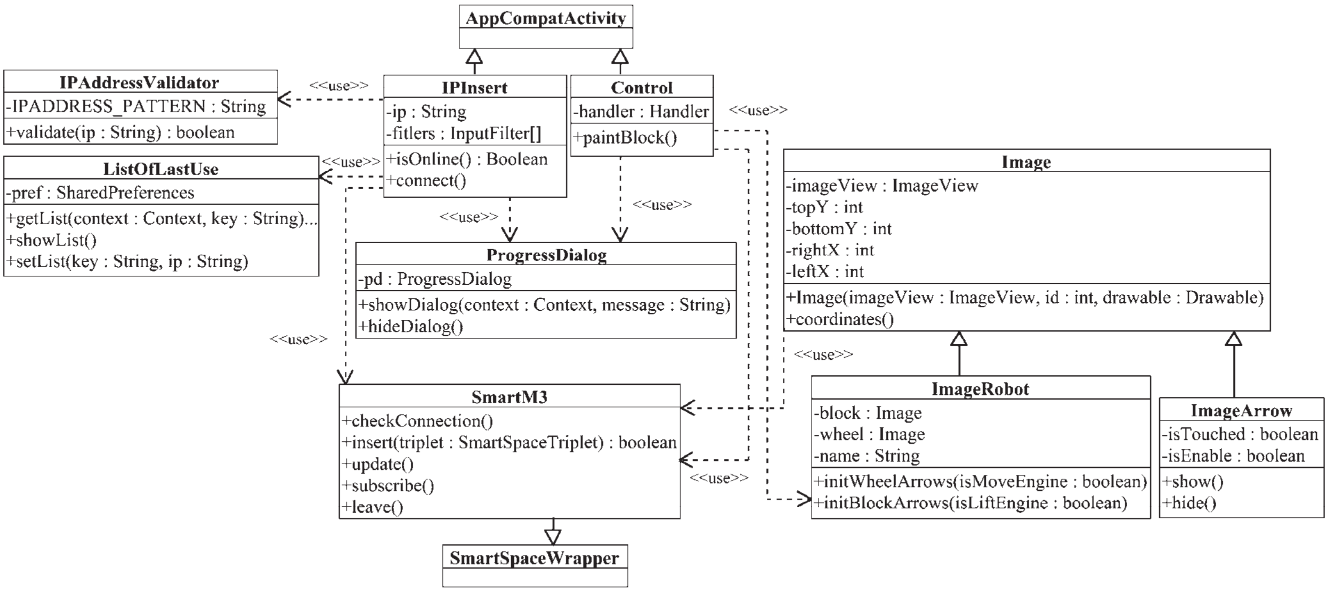

Fig. 4. Application class diagram

After the subscribing on the robot's configuration, the interface displays functional changes of UGV in real time. Because of impossibility to invoke the main thread from the process, works on the network, all subscribes or errors data are transmitted via the system messages processed by an instance of the Handler system class. Besides, on the main screen is shown stop button and the record button which is activated while overcoming obstacles in manual mode.

For notifying user about problems or incorrect data input Activity classes using methods of ProgressDialog class which display toast notifications and loaders.

VII. EVALUATION

Testing of the mobile application with adaptive interface was conducted under the following conditions. The Smart-M3 SIB was installed on the wireless router OS. Prototype of the six-wheeled robot based on the mobile robotic platform MRP-100 was created based on Lego EV3 (Fig. 5). This testing model has fewer features, but able to solve the problems approximated to reality.

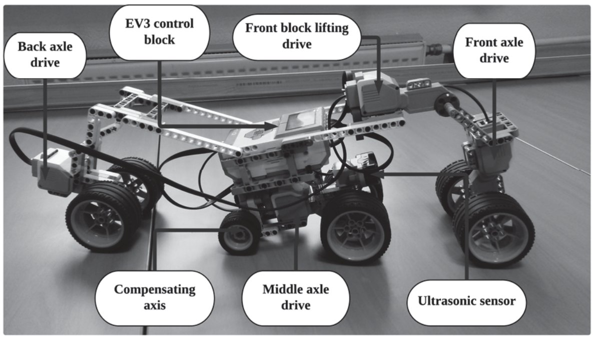

Fig. 5. Prototype of the six-wheeled robot based on Lego EV3

Plastic Lego parts much inferior in strength and mass than metal. For this reason on the central block of Lego-robot mounted an extra pair of wheels of a smaller diameter without a drive unit for the stability of the structure. Also the limited number of ports for connecting motors in the control unit did not allow to implement the lifting gear on closing block and functions of the length increasing. On the positive side the presence of ultrasonic sensor allows detect obstacles accurately and expects the start of maneuver execution more precisely.

The app was tested on Samsung, Asus and ZTE devices. Display size was ranged from 4.3 to 5.5 inches. Application interface has drawn correctly and accurately regardless of the display size and resolution through the vector icons usage. Fig. 6 shows two pictures of the main application screen.

The shown activity is a robot control adaptive interface and all the control elements situated here. The top of the image presents the moment when user tapped on the icon of leading (right) block. The bottom part shows the robot at the time of stopping in front of an obstacle. The user either focused on the guiding block. These examples shows the status of the remote robot control interface with the similar user actions, but in a different situational contexts. Both images were taken in the testing phase and confirm the ability of interface to change depends on the robot capabilities, the user needs and the context.

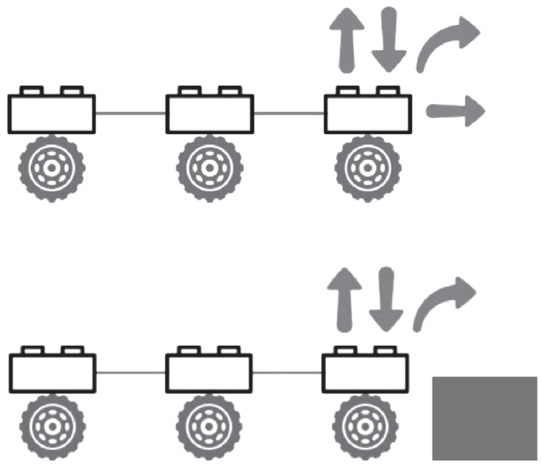

Fig. 6. Screenshot of adaptive interface

VII. CONCLUSION

This paper presents an architecture and mobile application development of a smartphone-based adaptive remote control interface for six-wheeled mobile robot. The related work in the field of mobile control interfaces was reviewed. The working capacity and usability of the interface has been tested on a specially designed multi-functional robot prototype. For future work the extension of the prototype functionality and number of control system services is planned. At the moment authors research possibility and use cases to equip the robot with a webcam. It helps to analyze the image using the Haar cascade and calculate the size of obstacles in front of the robot more accurately.

ACKNOWLEDGMENT

The presented results are part of the research carried out within the project funded by grants ## 16-29-04349 and 16-07-00462 of the Russian Foundation for Basic Research.

REFERENCES

[1] L. Voss, "Unmanned systems market is growing and changing", *Word of computer automation*, vol.1, 209, pp. 44-49.
[2] R.V. Meshcheryakov, O. Y. Osipov, Y. M. Osipov "Active driveline as an element of cyber-physical system", *Journal of instrument engineering*, vol.60, Jan.2017, pp. 934-938.
[3] IDC official website, A. Worldwide Smartphone Forecast Update, Web: https://www.idc.com/getdoc.jsp?containerId=US40734415.
[4] E. Borgia, "The Internet of Things vision: Key features, applica-tions and open issues", *Computer Communications*, vol.54, 2014, pp. 1-31.
[5] A.V. Borodin, "Internet of things – next stage of digital revolution", *Nauka i Obrazovanie.*, vol.5, Feb.2014, pp. 178-182.
[6] AnandTech website, The Great Equalizer 3: How Fast is Your Smartphone/Tablet in PC GPU Terms, Web: http://www.anandtech.com/show/6877/the-great-equalizer-part-3
[7] A.F.Verlan, M.F. Sopel, Yu. O. Furtat, "On adaptive user interface organization in automated systems", *Izvestiya SFedU. Engineering Sciences.*, vol.150,Jan.2014, pp. 100-110.
[8] V.P.Fofanov, E.M. Mehtiev, A.O. Savel'ev, A.A. Alekseev, "The use of permanent geological and technological models for adaptive

management of «intelligent» oilfield development", *Proceedings of TUSUR University.*, vol.34, Dec.2014, pp. 171-175.

[9] Y. Luo, J. Liu, Y. Gao, Z. Lu, "Smartphone-Controlled Robot Snake for Urban Search and Rescue" *Lecture Notes in Computer Science*., vol.8917, 2014 , pp. 352-363.

[10] Y. Seo, H. Jung, C. Lee, T. Yang, "Remote Data Acquisition and Touch-Based Control of a Mobile Robot Using a Smart Phone", *Communication and Networking.*, vol.265, 2012, pp. 219-226.

[11] G. Chen, S.A. King, M. Scherger, "Robot Remote Control Using Bluetooth and a Smartphone Augmented System", *Informatics in Control, Automation and Robotics.* vol.133, 2012, pp. 453-460.

[12] V.A. Egunov, O.A. Abed, "Mobile robot system control by voice commans", *Izvestia VSTU.*, vol.133, Jun.2014, pp. 137-138.

[13] C.W. Kenny, C.H. Foo, Y.S. Lee, "Interactive methods of Tele-operating a single unmanned ground vehicle on a small screen interface", *Proceedings of the 6th international conference on Human-robot interaction.*, 2011, pp. 121-122

[14] ROCAD official website, Science and Technology Center "ROCAD", Web: http://rocad.ru/en/.

[15] O. O. Basov, B. V. Sokolov, R. M. Yusupov, A. L. Ronzhin, "Conceptual and formal models of synthesis of cyber-physical systems and cyber-physical intellectual spaces", *Journal of instrument engineering*, vol.60, Jan.2017, pp. 897-905.

[16] A.M. Kashevnik, D.G. Korzun, S.I. Balandin, *Development of intelligence systems based on Smart-M3 platform*. Petrozavodsk: PetrSU, 2013