Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 56(3), 281–291(2015)

*Maja M. Lutovac Banduka*

# Remote Monitoring and Control of Industrial Robot based on Android Device and Wi-Fi Communication

DOI 10.7305/automatika.2015.10.1057
UDK 621.865.8:[621.398:654.93]; 004.732

**Original scientific paper**

Robot control systems are usually complex systems whose users must be well trained to use them. Also, control process is mainly carried out near the robot or by using wired connections. There is a need for a solution that can provide convenient and intuitive robot control with user's location independence, easy adjustment and simultaneously monitoring of robot motion tasks. Android devices are powerful mobile devices with open architecture and permanently Internet connection that can be applied to resolving those issues. This paper presents a system for remote monitoring and control of industrial robots based on Android device and Wi-Fi communication which provides intuitive robot control at a great distance with simultaneously monitoring of robot motions by observing its 3D model movement or trajectory path. Simple definition of new motion tasks and their redefinition is provided through the so called "speed dial" mode and manual robot guiding is provided through the manual control mode. Proposed solution simplifies an interaction between the human and the industrial robot in the case of robot motion control and tracking at remote location.

**Key words:** Android, remote monitoring, robot control, Wi-Fi

**Daljinski nadzor i upravljanje industrijskim robotima temeljeni na upotrebi Android uređaja i Wi-Fi komunikacije.** Sustavi upravljanja robotima su obično složeni sustavi čiji korisnici moraju biti dobro obučeni da ih koriste. Također, upravljanje procesima se uglavnom provodi u neposrednoj blizini robota ili pomoću žičane veze. Postoji potreba za rješenjem koje može osigurati udobno i intuitivno upravljanje robotima neovisno o mjestu korisnika, koje se lako koristi i istovremeno omogućuje praćenje kretanja robota. Android uređaji su moćni mobilni uređaji sa otvorenom arhitekturom i stalnom internetskom vezom koji se mogu primijeniti na rješavanje tih pitanja. U ovom radu je predstavljen sustav za daljinski nadzor i upravljanje industrijskim robotima temeljen na upotrebi Android uređaja i Wi-Fi komunikacije koji omogućuje intuitivno upravljanje robotima na velikoj udaljenosti uz istovremeno praćenje njihovih kretanja putem 3D modela ili putanje trajektorije. Jednostavno definiranje novih zadataka i njihovo redefiniranje je omogućeno kroz tzv. mod "brzog biranja", dok je ručno vođenje robota omogućeno kroz ručni mod. Predloženo rješenje pojednostavljuje interakciju između čovjeka i industrijskih robota tijekom upravljanja i praćenja kretanja industrijskih robota na udaljenom korisničkom mjestu.

**Ključne riječi:** Android, daljinski nadzor, upravljanje robotima, Wi-Fi

## 1 INTRODUCTION

The level of mobile device use is growing due to portability and efficiency in their use, and possibility to be used anywhere and anytime. There are predictions that their use will continue to rise in the next three years to a point at which mobile application development projects targeting Smart phones and tablets will outnumber native PC projects by a ratio of 4-to-1 [1].

In the last few years mobile devices based on Android operating system found their application in many fields, primarily due to low production price and possibility of continuous adaptation. They are used in television, Internet applications, media players and even automotive industry. The Android OS is open source operating system based on Linux kernel and developed by Google [2, 3]. The Android mobile phones and tablets provide usage of a number of useful services and applications, such as GPS navigation, tools for processing a wide range of text documents (PDF, Word), Web based applications (e-mail, Facebook), or phone calls via the Internet (Viber), in a very easy and intuitive manner.

The use of Android technology in robotics and systems for industrial robots control can give more intuitive and efficient solution in control and monitoring of robot's work, even on a distant location. Compared to the other mobile platforms, such as the iOS, in which case it is necessary to have an Apple device and pay the annual membership fee for access to official documents, Google provides free support and documentation access. Applications development can be performed independently of the operating system which is used for developing and the development environment and support utilities are free [4].

This paper presents the system for remote control and monitoring of industrial robots using Android device and Wi-Fi communication. The system consists of a device with the Android operating system and the real-time control system. The communication between the system for real-time control of robots and the Android device is being performed wireless, which provides user's location independence. The real-time control system is based on real-time Linux platform and OROCOS libraries [5], and it is connected with the robot using MOTENC board. MOTENC board is designed for PC based real-time systems controlling machines that require precision motion control. The board has all the necessary components to implement a precision digital feedback control system. The developed system can operate in two working modes: the control mode and the monitoring mode. In the control mode, the system enables remote control of industrial robot motion through the graphical user interface of the Android mobile phone. The remote control is performed by clicking the desired motion task command within the touch screen and simultaneous monitoring of robot's work using 3D robot model or following trajectory of the robot's end effector. The new motion tasks can be defined or redefined within the Android GUI (Graphical User Interface) and they can be assigned to the desired command button to provide "speed dial" of those motion tasks. In that way proposed solution significantly simplifies interaction between the human and the robot. The manual motion control with simultaneously motion tracking on the Android device is also enabled. The monitoring mode enables remote monitoring of the robot's motion in real time, using 3D industrial robot model or based on robot's end effector trajectory in 3D space, also on Android device.

In Section 2 related work is described. In Section 3, an overview of the proposed system is presented. Implementation of the Android application, 3D modeling of industrial robot Lola 50 [6], remote communication, and the structure and functionalities of the real-time control system are also described in this section. The application of the developed system and obtained results are shown in Section 4. A discussion about the advantages of the proposed solution compared to the existing solutions is given in Section 5. The concluding remarks are given in Section 6.

## 2 RELATED WORK

### 2.1 PC-based Applications in Robotics

The majority of robot manufacturers and many independent software vendors offer graphical environments in which users, namely developers and system integrators, can design and simulate their own manufacturing projects [7]. RobotStudio from ABB Robotics [8] is an offline robot programming and simulation tool that enables robot programming on a PC without the need to interrupt production. RobotStudio provides tools to increase the profitability of the robot system by providing training, programming and optimization without disturbing the manufacturing process. Another example is the KUKA SimPro robot simulation software for creating 3D layouts for systems with KUKA robots [9, 10]. This simulation package includes a program called KUKA OfficeLite that generates KUKA robot commands. Roboguide is a tool for simulation of work environment of the robot on the PC [11]. It is used by the system designers and in the production process. It is capable to check the interference between the robot and other objects, to control different operations by means of simulations, and to monitor the current status of the robot.

### 2.2 Wireless Communication

Android devices, but also the other smart devices [12], provide different communication interfaces, such as Wi-Fi, Bluetooth, USB, and different types of integrated sensors, such as accelerometer, gyroscope, compass and GPS [13]. Currently the wireless scene is held by two standards, the Bluetooth and the IEEE 802.11 (Wi-Fi) protocols, which define the physical layer and the medium access control (MAC) for wireless communications over a short action range (from a few up to several hundred meters) and with low power consumption (from less than 1 mW up to hundreds of mW). Bluetooth is mainly oriented towards connections between close-connected devices, as a substitute for data transfer cables. IEEE 802.11 is devoted to connections among computers, as an extension or substitute for cabled LANs [14]. Bluetooth technology is useful when transferring information between two or more devices that are near each other and speed is not an issue, such as telephones, printers, modems and headsets (from 10 cm up to 10 m) [15]]. Wi-Fi enables a faster connection, better range from the base station, and better security (if configured properly) than the Bluetooth. Comparison characteristics of those two wireless protocols are given in Table 1.

As it was necessary to implement an application that enables control of industrial robot at a great distance, in the case where the real machine cannot be visible and where monitoring is achieved through the 3D model movement or by drawing the trajectory of its end effector in real time, the Wi-Fi IEEE 802.11 protocol was chosen for the implementation.

*Table 1. Comparison of Bluetooth and Wi-Fi protocols*

| | Bluetooth | Wi-Fi |
|---|---|---|
| Frequency band | 2.4 GHz | 2.4 GHz, 5 GHz |
| Nominal range | 10 m | 100 m |
| Maximum one-way data rate | 732 kb/s | 31.4 Mb/s |
| Maximum signal rate | 1 Mb/s | 54 Mb/s |
| Typical current absorbed | 1–35 mA | 100–350 mA |

### 2.3 The Use of Android Devices in Robotics

There are several implementations of robot control over smart phones and using Bluetooth communication [13,16–18]. Reference [16] describes the system for communication with an articulated robot using Android device and Bluetooth. Reference [17] proposes the system for environmental temperature monitoring using the mobile robot, Android device and Bluetooth communication. Methods for connecting Android device with LEGO Mindstorms NXT robotic system are proposed in [13]. The paper describes the use of Android device for remote control in LEGO Mind-storms NXT system via Bluetooth or USB connection. Reference [18] describes how to control a robot using mobile device through Bluetooth communication that includes map based mode of controlling the robot. By this mode, the robot can find the shortest path from defined start point to target point by avoiding all obstacles defined by the user or found while moving. Existing solutions that use Bluetooth for communication [13, 16–18] can provide communication only on shorter distance.

Wi-Fi-based solutions for monitoring and control of robot's or machine's motions are described in [19–22]. Those solutions propose monitoring by using camera [19], by displaying values of the significant parameters as text [20], or by using 3D animation [21], and detecting of robot's position by capturing camera images [19], by using sensors [20], or by using the profile of known environment [22].

Reference [19] proposes an Android OS based robot control and monitoring system that uses 802.11x wireless LAN communication and camera for robot monitoring. The image from the camera mounted on the robot in real time is delivered to the Smartphone through 802.11x wireless LAN. The image size acquired from the camera is around 60-80k bytes, and this large data size makes image transfer by a Smartphone difficult without manipulating YUV image signals. To keep less small size of image data the converting of the YUV image data to JPEG format is proposed in [19]. This yields the image down to 2k bytes level. Still, transferring smaller amount of data in short time intervals that is required for monitoring, can be more useful. It also can be cost-effective in cases when the mobile provider's Internet is used and when the user pays for each transferred byte.

Reference [20] proposes a Smartphone-based human-machine interface for remote control of robot arm. This solution uses the accelerometer and the gyroscope of the Smartphone for command generation and Wi-Fi protocol for communication. The messages that are transferred in the communication are ASCII strings. The monitoring is provided by displaying values of real-time data of all sensors, robot positions and the other significant parameters as text. That kind of preview is not always suitable for obtaining a clear insight into the robot performance.

Another example of remote control applied to the CNC machine is proposed in [21]. This system consists of an Android device that represents a client and the Windows Web server with hybrid application which combines SOAP web service to send and receive control information and Socket communication mechanism to timely update feedback data from the machine via the wireless network. The developed hybrid application enables real-time process data update at rate of 30 ms. Monitoring is obtained by using 3D visualization to animate the machine's real-time operation. The control is performed by filling out the text box or by sending movement directions within the touchscreen of Android device. More intuitive human-robot interface, which would facilitate the programming of robot tasks, would be of significance.

## 3 PROPOSED SYSTEM

### 3.1 System Overview

The system for remote monitoring and control of industrial robot motions based on the usage of Android device and Wi-Fi communication is developed at the Lola Institute. Structure of the proposed system is shown in Fig. 1.

The system can operate in two working modes: the control working mode and the monitoring working mode. In the control working mode the wireless control of industrial robot by using the graphical user interface of Android mobile device is provided. The control can be performed by "speed dial" motion program selection, by standard motion programming or by manual mode control. The "speed dial" motion program selection can be obtained within the Android application through the touch screen of the device. The "speed dial" motion tasks can be define and redefined using desired set of robot motion instructions. After selection of motion tasks, an execution file, in form of object code (run file), is sent to the real-time system. The real-time system interprets instructions written in object code

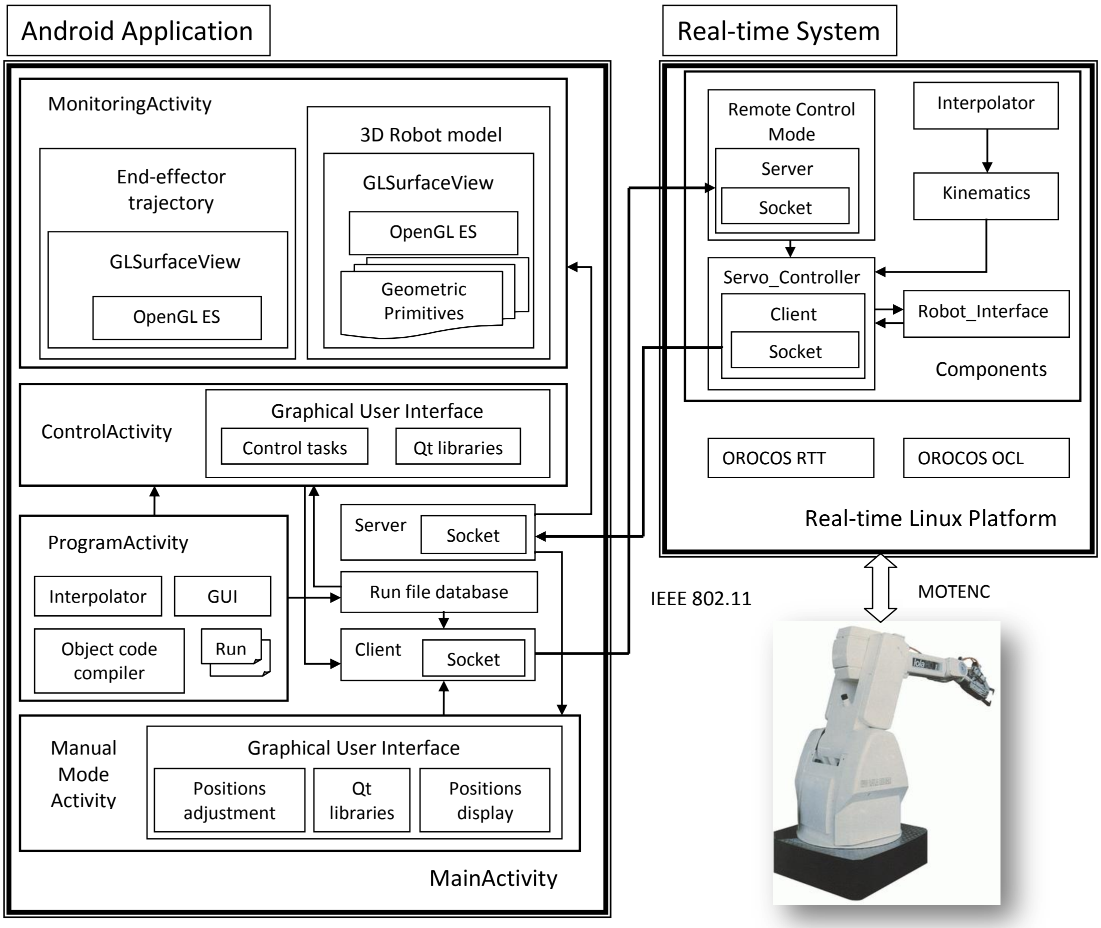


*Fig. 1. Structure of developed system for remote monitoring and control of industrial robots based on Android device and Wi-Fi communication*

and manages the robot motion based on the program content. Android application contains run file database. The run file is a motion program which contains parameters obtained by execution of algorithms for industrial robot path planning developed at the Lola Institute. The run file is generated based on motion instructions that specify motion type, motion speed, motion acceleration and positions represented in internal coordinates, within the robot programming GUI for motion tasks programming and redefining on Android client. Motion types are PTP – point to point, CIRCLE – circle movement, and LIN – linear movement. PTP and LIN movement requires one position to be specified for motion execution and CIRCLE movement requires definition of one more auxiliary position. After motions specification, the user generates an execution file, which can be assigned to the desired "speed dial" button. The proposed implementation enables speed dialing of previously defined motion tasks on Android device. The control can also be applied in manual mode, by adjustment of positions displacements of robot's axes. Displacements can be in positive and in negative direction.

The monitoring working mode enables monitoring of robot's work from the remote user's location within the Android application. It is provided through the 3D industrial robot movement or through the robot's trajectory representation in a 3D coordinate system. It can be applied in combination with the "speed dial" robot control, or manual robot control, but it can also be applied independently.

The Android mobile device and the real-time control system are connected into the network by using Wi-Fi 802.11 technology. Communication is implemented using TCP/IPv4 protocol and Socket communication mechanism. The real-time control system is connected to the robot using MOTENC board. It performs sending of desired positions to the robot and reading achieved position of the robot from encoders.

### 3.2 Android Application Development

For the implementation of Android application Android/Java programming language is used. Eclipse SDK [23] is used as a development environment. The application is developed for 2.3.3 (API level 10) version of Android operating system. For experimental testing the Android mobile phone HUAWEI G300 is used.

In the control working mode Android device represents a client. The client is implemented by using Socket communication mechanism and Java programming language. It represents the part of Android application. The graphical user interface is implemented by using Java programming language and XML files. Events that are generated by clicking on the appropriate button at the graphical user interface are connected to functions for sending commands and data to the server. Message that can be sent to the server can be control message (defined in ControlActivity), and message for manual positioning of robot (defined in ManualMode Activity). Control messages are messages for starting or stopping robot's work and messages for robot motion tasks execution (run files). There are several "speed dial" messages for robot motion tasks execution and each of them is connected to the corresponding run file from the run file database. Run files contain parameters for the specific robot motion task execution and they are executed within the real-time system. Parameters from run files are obtained based on algorithms for industrial robot motion (placed within the Interpolator component) which are developed at the Lola Institute. A new run file with desired instructions, specified motion type, speed, acceleration and positions can be defined within the GUI of ProgramActivity. The object code compiler translates specified motion parameters into the run file and places it into the run file database. An appropriate speed dial control button within the ControlActivity GUI can be associated with generated run file. This implementation enables quick and simple selection and execution of industrial robot motion tasks. Manual positioning messages contain information about axis number and motion direction, and they are defined within the ManualModeActivity. During the manual positioning of robot, the current robot positions are displayed within the ManualModeActivity GUI.

In the monitoring working mode Android device represents a server. The server is implemented by using Socket communication mechanism and Java programming language, and represents the part of Android application. The monitoring can be performed by using 3D model of robot Lola 50 and by using 3D representation of trajectory of the robot's end effector. When monitoring is performed using 3D industrial robot model, commands that server can receive from the client (real time control system) carry information about the current position for each axis of the robot or sign for end of monitoring. When server receives command with position value it adjusts the angle of orientation for each axis of virtual robot model. Since these changes occur in short time intervals of 15 ms, the approximate continuous movement of virtual robot is obtained. In the case when monitoring is carried out using 3D robot's end effector trajectory in 3D coordinated system, commands that server can receive from the client carry information about the position of industrial robot's end-effector in 3D coordinated system. Trajectory drawing is performed in real-time.

### 3.3 3D Model Implementation

The 3D model of industrial robot Lola 50 is implemented by using Java programming language, Android libraries and OpenGL ES libraries. Components of the virtual model are simplified and described using basic geometric primitives (box, cylinder and sphere). Positions of primitives are programmed relative to a specified reference coordinate system and primitives that make a whole are grouped. Movable elements of robot are connected with relevant connections, rotational, or translational. All robot parameters in the virtual robot are set as on the real machine and the robot axes directions are set according to defined kinematic model. After completing construction, axes of 3D robot model are connected to appropriate signals that cause theirs position adjustment in real time monitoring.

### 3.4 Remote Communication

The Android mobile device and the real-time system are connected into the network using Wi-Fi 802.11 technology. In addition to the work described in [24] where basic manual robot control is explained, in this paper the "speed dial" control and the monitoring of robot performance is implemented. During the programming and real-time control of robot manipulators, Android device is used as a client and the real-time system is used as a server. The client can send to the server the following types of messages: task execution messages, messages for the manual manipulator guiding and short control messages. The task execution messages are sent by clicking on the "speed dial" GUI button. They are sent in object code form and executed on the real-time system. Those messages contain parameters obtained by execution of algorithms for robot manipulator path planning based on the manipulator's motion type, motion speed, motion acceleration and positions represented in internal coordinates that are specified within the manipulator's programming GUI. Those parameters are obtained based on algorithms for industrial robot motion which are developed at the Lola Institute [6]. The manual guiding messages are sent as string messages. Each manual guiding message contains information about

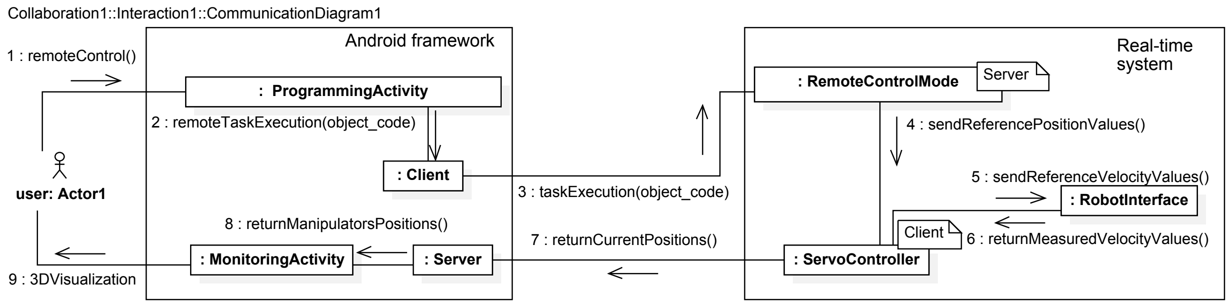


*Fig. 2. Communication diagram for remote control and monitoring*

the axis of which the adjustment of the position is carried out. Short control messages are messages for starting and stopping the manipulator's work.

During the real-time monitoring of robot manipulators the Android device is used as a server and the real-time system is used as a client. The monitoring is performed using the 3D manipulator's model or trajectory path in 3D coordinated system (defined in the 3D Robot model) within the Android application. The real-time system is connected to the manipulator using MOTENC board. It performs sending of desired position to the manipulator and reading the achieved position of the manipulator from the encoders. The position that is read from the encoders is sent to the Android server at 15 ms time intervals. When the server receives a message with the current position value, operation of axes rotation on the 3D model or trajectory path of the manipulator is executed.

The communication diagram for remote task control and monitoring is given in Fig. 2. The communication is described for the case when user performs task execution of the robot manipulator and remote monitoring of its execution. User initiates remote control through the "speed dial" task execution message and the ProgramActivity that is located within the Android framework forwards remoteExecution message to the Client. The Client is connected to the Server whose implementation is located within the real-time system's Remote_Control_Mode component. This message contains the object code which comprising parameters for robot manipulator path planning. The Remote_Control_Mode component uses functionalities of the Interpolator component to calculate the reference position value that is sent to the Servo_Controller component. The Servo_Controller sends reference velocity value to the Robot_Interface component that is connected to the real manipulator by using MOTENC board. This component calculates the velocity output from the real manipulator and forwards it to the Servo_Controller component which contains the Client implementation for the reverse communication. The Servo_Controller calculates and returns the current manipulator's position to the Server within the Android framework in 15 ms intervals. The Server accepts and forwards those values to the MonitoringActivity which performs 3D visualization of the manipulator's task execution in real time.

### 3.5 Real-time Control System

The real-time control system is structured in layers [25]. On the bottom is the real-time Linux operating system which kernel is patched with open source real-time framework Xenomai. Xenomai is a real-time development framework cooperating with the Linux kernel, in order to provide a pervasive hard real-time support to user-space applications. OROCOS Real-Time Toolkit (RTT) is referred as middleware that lying between operating system and application level. The RTT provides infrastructure and functionalities to build robotics real-time applications in C++. It allows setup, distribution and building of real-time components, which are the highest layer of the control system. The client and the server for communication with Android device, both in control and monitoring mode, are implemented by using C++ programming language and Socket communication mechanism.

The application layer consists of components. Component Remote_Control_Mode contains Server implementation and receives messages for remote control or remote manual positioning of robot from Android device in control operating mode. It process received message that contains information about message type. If it is a message for motion tasks execution, server receives the run file. In this case Remote_Control_Mode component sends the run file to the Interpolator component where it has been executed. Interpolator is a trajectory generator. It calculates the reference position values that have to be sent to the Servo_Controller component. New values are written in 5 ms time intervals. The Interpolator component implements algorithms developed at the Lola Institute [6]. The Servo_Controller component represents an abstract system controller. The Servo_Controller sends reference velocity values that have to be reached to the Robot_Interface component every 1 ms. It uses a position feedback law to calculate the velocity output from measured and desired positions. Component Robot_Interface is connected to the real robot by using MOTENC board and it contains implementations of required drivers. If the server receives message for manual positioning, it receives axis number and motion direction. In this case Remote_Control_Mode component sends obtained data directly to the Servo_Controller component. When server receives command for starting or stopping the robot's work, it performs the following operations: preparation of Robot_Interface component, engine breaks releasing and enabling of reading and writing to robot's ports ie. stopping Robot_Interface component, brakes raising and disabling of reading and writing to ports.

In the monitoring working mode, the real-time system works as a client that is implemented within the Servo_Controller component. It connects to the server that runs on Android device and sends the current robot positions in internal coordinates. Positions are sent to the Android server at 15 ms time interval and it is sufficient because the frame rate on Android device is about 60 fps. The 3D model simulation arises as the result of adjustment of those values at the virtual robot model. In that way an approximate continuous virtual robot motions which simulates the motions of the real machine is obtained. In case where the remote monitoring is performed through the end effector's positions, the simulation is executed in similar manner. Only difference is that obtained values for robot positions must be transformed into the external coordinates. This is done by using forward kinematics equations for robot Lola 50 [6].

## 4 USAGE AND RESULTS

The main page of Android application is shown in Fig. 3. The application contains options for robot control (Start Robot, Stop Robot, Remote Robot Control and Manual Robot Control), option for defining and redefining new robot motion tasks (Tasks Programming) and option for monitoring of robot's work remotely and wireless (Remote Robot Monitoring).

By clicking on the option for remote (speed dial) robot control or option for manual mode control, application opens new screen content and the client connects to the server through a specified IP address. Lola 50 [6] movement can be monitored using 3D virtual model or using the representation of motion path of its end effector in 3D space. The execution of defined robot's motion tasks can be achieved by clicking on the one of the defined motion commands within the application page for remote, "speed dial", robot control shown in Fig. 4. The new robot motion

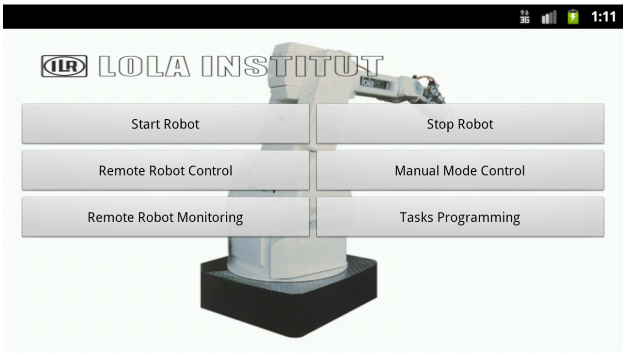


*Fig. 3. Main page of Android application*

tasks can be defined within the application page shown in Fig. 4. There is possibility to define or redefine motion instructions with specified motion type (PTP – point to point, LIN - linear or CIR - circle) within the drop-down menu, and specified speed, acceleration and target positions in external coordinates (position and orientation). Only for CIR motion type the user must specify an auxiliary position. Motion instructions are executed sequentially in the order in which they were listed. After defining or redefining motion task, it can be associated with the desired button and name within the speed dial control interface.

The testing of remote robot control is performed through the combined motion task of industrial robot Lola 50. Robot movement is generated by executing the run file within the real-time control system and the obtained motion path of its end effector in 3D space is monitored in right part of the GUI shown in Fig. 4. Figure 6 presents graph which describes the change of angle values, for six robot axis (ang1-ang6), during the robot's movement that is defined within the motion program shown in Fig. 5. The angle values are measured in radians.

The content of an application page for robot control in manual operating mode with 3D virtual model of robot Lola 50 is given in Fig. 7. The graphical user interface contains options for manual guiding for each axis of an industrial robot. The industrial robot Lola 50 has 6 degree of freedom, which means that it contains the possibility of movement in 6 different axes. The axes are marked with Axisi where i denotes the number of axis. Each axis can be moved in a positive direction (+) and in a negative direction (-). The axes can be moved until they reach the limit positions. Once the command is received from the client and motion information is processed, the real-time system returns notification if the motion can be executed properly or not, and if it can, the real-time system starts returning angles values for each robot axis. In case of that there is a barrier that affects the inability to perform specified motion task, the real-time system also returns an appropriate

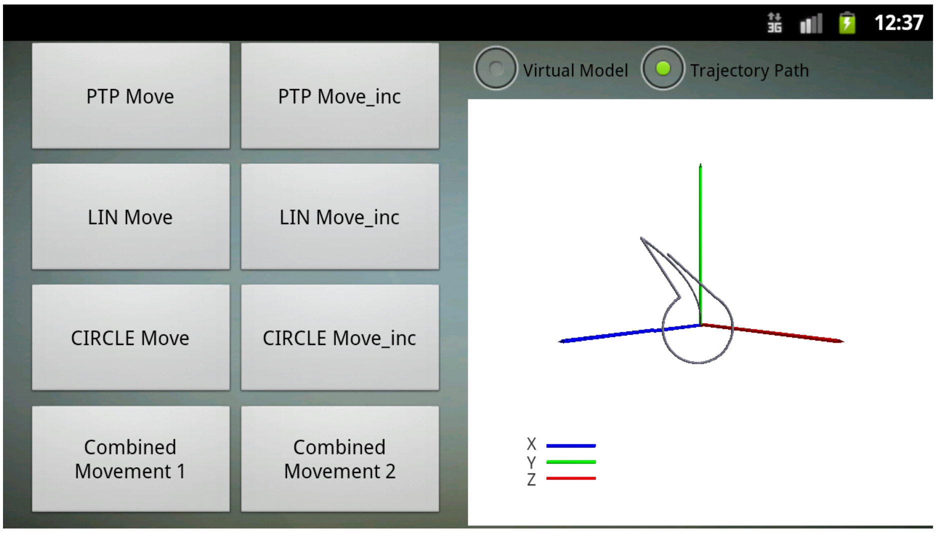


*Fig. 4. Remote robot control in "speed dial" mode with monitoring using robot's end effector trajectory in 3D coordinate system*

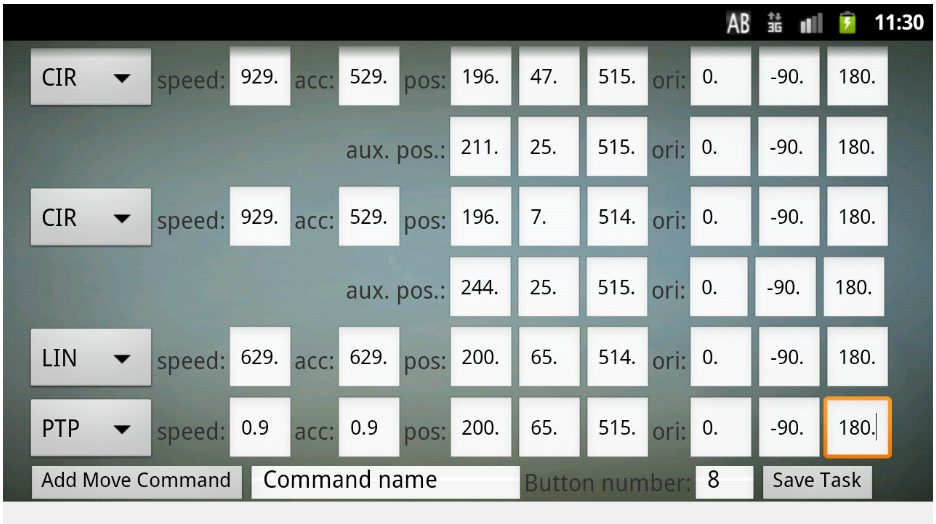


*Fig. 5. Tasks programming page*

notification. The testing of manual robot control is performed and the movement of Lola 50 is monitored using its 3D virtual model shown in Fig. 7.

## 5 DISCUSSION

Compared to the PC-based tools and applications for robot programming and simulation based on virtual representation of the robot [8–11], the approach proposed in this paper gives two major advantages: the simplicity of interaction and the improved access. The simplicity of interaction is provided through the simple, single click execution of complex robot tasks and algorithms using the touch screen interface. Complex motion instructions for defining robot tasks can be defined and appointed within the touch screen programming GUI, and connected with the desired "speed-dial" button. In the control process the robotic tasks execution is initiated with the single user's click, sending the execution message to the robot manipulator and giving the real-time virtual representation of task execution. The anywhere and anytime access is provided with the use of WLAN Internet, and when WLAN connection is not available, Smartphone also gives the possibility to connect to the Internet by using the mobile provider.

The proposed solution overcomes deficiencies and integrates the most effective functionalities of described wire-

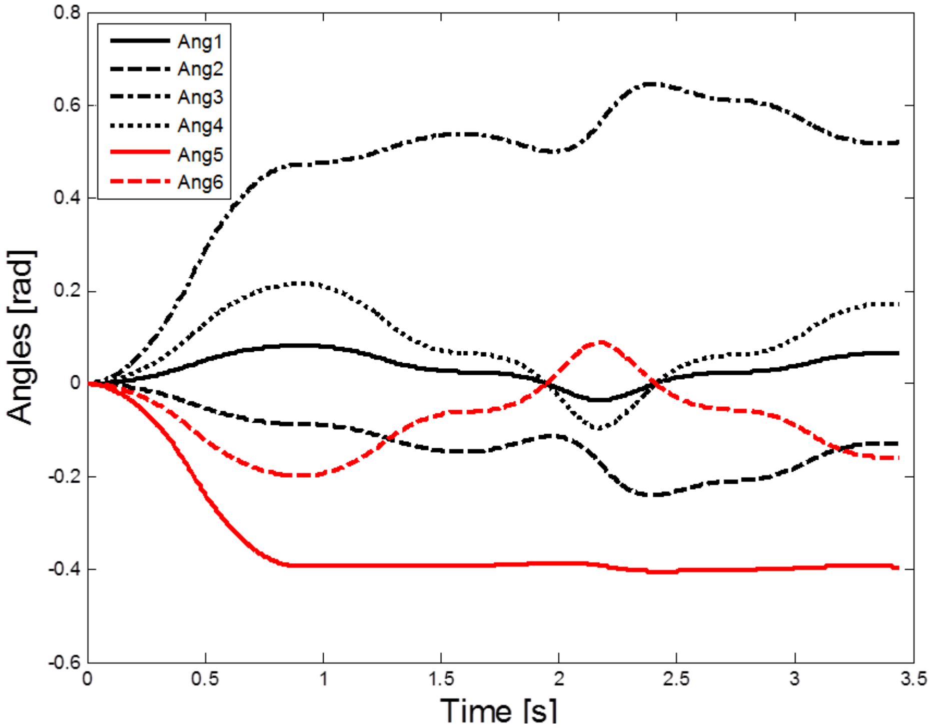


*Fig. 6. Robot axes positions obtained during the execution of Combined Movement 2 task in radians*


*Fig. 7. Remote robot control in manual mode with monitoring using 3D robot model*

less solutions [13, 16–21], providing intuitiveness of interaction in robot control, virtual monitoring of robot performance and communication that uses minimal amounts of data to provide cost-effectiveness. The proposed environment overcomes the issue of large data transfer during the control and monitoring process. This is achieved by sending the object code commands during the control process and by sending the current robot position as a string message during the monitoring process that, for the robot manipulator with 6 DoF, is about several dozen bytes. In this paper, it is proposed to perform the monitoring of the manipulator motion by using 3D robot model and following trajectory of the robot's end effector which gives natural monitoring experience compared to the monitoring by following values of significant motion parameters. The intuitive human-robot interface is achieved by implementing "speed dial" mode that enables the most common robot motion tasks to be executed with a single click on the Android touch screen.

This virtual mobile environment can be also used in the laboratory for monitoring the robot performance from two different perspectives: by observing the real machine's motion and by monitoring virtual motion on the mobile device. This studying from multiple perspectives can give better insight into the robot and industrial algorithms and tasks.

The main advantages of the proposed solution are the simplicity and the intuitiveness of human-robot interaction ("speed dial" - single click execution of most common complex robot algorithms and tasks; touch screen interaction), the user's location independence (providing multi-perspective monitoring of robot performance; usable anywhere and anytime over WLAN or mobile provider's Internet), and the minimal data transfer (object code), which overcomes deficiencies of other solutions [8–11, 13, 16–21].

## 6 CONCLUSION

In this paper, the system for remote monitoring and control of industrial robot Lola 50 is presented. The proposed methods can also be applied on control and monitoring of other industrial robots and devices. The developed system allows intuitive and flexible control of industrial robot by using Android mobile device which today represents a prevalent and inexpensive device. The "speed dial" control mode with tasks programming GUI enables easy robot motion tasks execution, defining and redefining which simplifies an interaction between the human and the industrial robot. The system also contains possibility for manual robot guiding in manual control mode. The user's location independence is obtained by using Wi-Fi communication. Simultaneously remote monitoring of robot's work at a great distance is enabled by using 3D robot model or end effector's trajectory.

The current and future work is oriented toward the consideration of usage of sensors that Android device owns, for the purpose of robot control, such as an accelerometer or microphone.

## ACKNOWLEDGMENT

This work was created within the research project that is supported by the Ministry of Education and Science, Republic of Serbia under Grant TR35023.

## REFERENCES

[1] C. Boja and A. Zamfiroiu, "Input methods in mobile learning environments," *Studies in Informatics and Control*, vol. 22, no. 4, pp. 329–338, 2013.

[2] J. Steele and N. To, *The Android developer's cookbook: building applications with the Android SDK*. USA: Addison-Wesley, 2013.

[3] S. Conder and L. Darcey, *Android wireless application development*. Boston, USA: Addison-Wesley, 2010.

[4] D. Hižak and M. Mikac, "Development of a simple tool for audio analysis on mobile Android platform," *Tehnički glasnik*, vol. 7, no. 2, pp. 177–181, 2013.

[5] G. Ferenc, M. Lutovac, J. Vidaković, Z. Dimić, and V. Kvrgić, "Real-time robot control logic using modular FSM," in *Proceedings book of the 4thInternational Conference Management of Technology - Step to Sustainable Production*, (Zadar, Croatia), pp. 259–265, June 2012.

[6] V. Kvrgić, *Development of Intelligent Systems for Industrial Robots Control and Programming*. PhD thesis, University of Belgrade, Faculty of Mechanical Engineering, 1998.

[7] J. N. Pires, "Robot manipulators and control systems," in *Industrial Robots Programming: Building Applications for the Factories of the Future*, (USA), pp. 35–107, 2007.

[8] C. Connolly, "Technology and applications of ABB RobotStudio," *Industrial Robot: An International Journal*, vol. 36, no. 6, pp. 540–545, 2009.

[9] Y. H. Jen, Z. Taha, and L. J. Vui, "Vr-based robot programming and simulation system for an industrial robot," *International Journal of Industrial Engineering*, vol. 15, no. 3, pp. 314–322, 2008.

[10] A. Sett and K. . Vollmann, "Computer based robot training in a virtual environment," in *Proceedings book of the IEEE International Conference on Industrial Technology (ICIT '02)*, (Bangkok, Thailand), pp. 1185–1189, December 2002.

[11] R. Wolny, "Robots in technological process of painting," in *DAAAM International Scientific Book*, (Austria), pp. 195–204, 2011.

[12] M. Ilyas and S. A. Ahson, "Smartphones," tech. rep., Chicago, IL: Intl. Engineering Consortium, 2006.

[13] S. Goebel, R. Jubeh, S.-L. Raesch, and A. Zuendorf, "Using the Android platform to control robots," in *Proceedings book of the 2th International Conference on Robotics in Education (RiE 2011)*, (Vienna, Austria), pp. 135–142, 2011.

[14] E. Ferro and F. Potorti, "Bluetooth and Wi-Fi wireless protocols: A survey and a comparison," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–16, 2005.

[15] C. M. D. Dominicis, D. Mazzotti, M. Piccinelli, S. Rinaldi, A. Vezzoli, and A. Depari, "Evaluation of Bluetooth hands-free profile for sensors applications in smartphone platforms," in *Sensors Applications Symposium (SAS)*, (Brescia, Italy), pp. 1–6, February 2012.

[16] S. V. D. nd A. Whigham, "A Bluetooth-based architecture for Android communication with an articulated robot," in *Proceedings book of the International Conference on Collaboration Technologies and Systems (CTS)*, (Denver, CO), pp. 104–108, May 2012.

[17] T. M. Jenifer, T. S. V. Priyadharshini, R. Lavanya, and S. R. Pandian, "Mobile robot temperature monitoring system controlled by Android application via Bluetooth," *International Journal on Advanced Computer Theory and Engineering (IJACTE)*, vol. 2, no. 3, pp. 138–142, 2013.

[18] H. Nasereddin and A. Abdelkarim, "Smartphone control robots through Bluetooth," *International Journal of Research and Reviews in Applied Sciences*, vol. 4, no. 4, pp. 399–404, 2010.

[19] S.-W. Moon, Y.-J. Kim, H. J. Myeong, C. S. Kim, N. J. Cha, and D. H. Kim, "Implementation of Smartphone environment remote control and monitoring system for android operating system-based robot platform," in *Proceedings book of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, (Incheon, Korea), pp. 211–214, November 2011.

[20] C. Parga, L. Xiaoou, and Y. Wen, "Tele-manipulation of robot arm with Smartphone," in *Proceedings book of the 6th International Symposium on Resilient Control Systems (ISRCS)*, (San Francisco, USA), pp. 60–65, August 2013.

[21] N.-V. Truong and D.-L. Vu, "Remote monitoring and control of industrial process via wireless network and Android platform," in *Proceedings book of the International Conference on Control, Automation and Information Sciences (ICCAIS)*, (Ho Chi Minh City, Vietnam), pp. 340–343, November 2012.

[22] V. Baranauskas, S. Bartkevicius, O. Fiodorova, and K. Sarkauskas, "Detecting the mobile robot position using the profile of known environment," *Electronics and Electrical Engineering*, vol. 19, no. 7, pp. 7–10, 2013.

[23] T. Mens, J. Fernández-Ramil, and S. Degrandsart, "The evolution of Eclipse," in *Proceedings book of the International Conference on Software Maintenance (ICSM)*, (Beijing, China), pp. 386–395, September 2008.

[24] M. Lutovac, J. Protić, and V. Kvrgić, "Remote control of industrial robot lola 50 using wireless communication and Android device," in *Proceedings book of the 21st Telecommunications Forum (TELFOR)*, (Belgrade, Serbia), pp. 885–888, November 2013.

[25] G. Ferenc, Z. Dimić, M. Lutovac, J. Vidaković, and V. Kvrgić, "Open architecture platforms for the control of robotic systems and a proposed reference architecture model," *Transactions of FAMENA*, vol. 37, no. 1, pp. 89–100, 2013.

**Maja M. Lutovac Bnaduka** was born in Belgrade, Serbia, in 1985. She received her B.Sc. and M.Sc. from the Faculty of Electrical Engineering in Belgrade, Republic of Serbia, in 2009 and 2011 respectively. Lutovac Banduka is Research Associate in Lola Institute, Belgrade, Serbia. Areas of professional and research activities are software development for industrial robot and devices for pilot training and flight simulation, robot programming language, Android application development and 3D graphic programming. She is currently working on her Ph.D. thesis at the Department of Software Engineering, Faculty of Electrical Engineering, Belgrade, Serbia..

**AUTHORS' ADDRESSES**

**Maja M. Lutovac Banduka, M.Sc. EE**
**Department of Software Engineering,**
**Faculty of Electrical Engineering, University of Belgrade,**
**Bulevar kralja Aleksandra 73, RS-11000, Belgrade, Serbia,**
**Department of Robot Controllers,**
**Lola Institute,**
**Kneza Višeslava 70a, RS-11000, Belgrade, Serbia**
**email: maja.lutovac@li.rs**

Received: 2014-10-15
Accepted: 2015-08-26