

УНИВЕРЗИТЕТ У БЕОГРАДУ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

Маја М. Лутовац Бандука

**СОФТВЕРСКИ СИСТЕМ ЗА
ДАЉИНСКО УПРАВЉАЊЕ И НАДЗОР
РОБОТА БАЗИРАН НА АНДРОИД
ОПЕРАТИВНОМ СИСТЕМУ И
БЕЖИЧНОЈ КОМУНИКАЦИЈИ**

докторска дисертација

Београд, 2016

UNIVERSITY OF BELGRADE
SCHOOL OF ELECTRICAL ENGINEERING

Maja M. Lutovac Banduka

**SOFTWARE SYSTEM FOR REMOTE
CONTROL AND MONITORING OF
ROBOTS BASED ON ANDROID
OPERATING SYSTEM AND WIRELESS
COMMUNICATION**

Doctoral Dissertation

Belgrade, 2016

Ментор:

др Бошко Николић, редовни професор
Универзитет у Београду, Електротехнички факултет

Чланови комисије:

др Милош Цветановић, доцент
Универзитет у Београду, Електротехнички факултет

др Владимир Квргић, виши научни сарадник
Лола институт

др Коста Јовановић, доцент
Универзитет у Београду, Електротехнички факултет

др Вељко Поткоњак, редовни професор
Метрополитан универзитет у Београду, Факултет информатичких
технологија

Датум одбране:

Изјава захвалности

Захваљујем се свом ментору, професору др Бошку Николићу, на пуној подршци током израде ове докторске дисертације и досадашњег научног рада. Желела бих да се захвалим и професору др Милошу Цветановићу, др Владимиру Квргићу, професору др Вељку Поткоњаку и професору др Кости Јовановићу на корисним саветима и сугестијама који су допринели коначном изгледу дисертације.

Овај рад не би био могућ без подршке колега из Лола института на челу са директором др Владимиром Квргићем, којима дугујем велику захвалност на увек инспиративној и пријатној радној атмосфери.

Посебну захвалност дугујем својој породици, родитељима Мирославу и Анити, супругу Марку и сестри Александри на безрезервној подршци коју су ми пружили током рада на овој дисертацији. Својој ћерки Еми, која ми је била највећа мотивација у току рада на овој теми, посвећујем овај рад.

Наслов докторске дисертације: Софтверски систем за даљинско управљање и надзор робота базиран на Андроид оперативном систему и бежичној комуникацији

Резиме: Роботски системи су сложени системи чији корисници морају бити добро обучени. Управљање се најчешће спроводи близу самог робота или употребом жичне конекције. Данашње брзе промене и напредак у развоју технологије захтевају имплементацију нових технолошких решења у области програмирања и управљања индустријских робота и других робота специјалне намене. У савременој роботској индустрији постоји стална потреба за унапређењем приступачности и интеракције у раду са роботским системима, па је све већи значај развоја корисничких интерфејса који су интуитивни и лаки за коришћење. У складу са тим, приликом пројектовања графичког корисничког интерфејса за програмирање и управљање робота није довољно направити фокус само на обезбеђивању детаљних и комплетних функционалности управљања, већ се ергономија употребе такође треба узети у обзир. Поред тога, масивни и недоступни работи све више захтевају управљање независно од локације оператера.

Ресурси везани за роботске лабораторије често нису доступни корисницима које тек треба обучити за рад са оваквим системима, а рад са роботским системима често захтева детаљно познавање софтверских парадигми за програмирање робота. Због наведених разлога студенти који тек уче о основним принципима роботике углавном нису у могућности да се упознају са програмирањем и управљањем комплексних роботских система на практичан начин. У области едукације у роботици, бежична виртуелна окружења, која дају реалну слику о раду робота, би била од великог значаја. Посебно би се издвојио њихов значај у случају примене таквог решења за едукацију у области управљања слабо доступних робота као што су, на пример, уређаји за адаптацију и тренинг пилота. Такође, поред едукативне сврхе, употреба лако доступних, широко распрострањених и економичних уређаја би била јако исплатива. Истраживање о генералној употреби и знању студената о технологији паметних телефона, које је спроведено у оквиру ове

докторске дисертације, потврђује да данашњи брз развој технологије захтева унапређење метода за учење и едукацију.

Постоји мноштво научно-истраживачких радова и извештаја из праксе који описују софтверска решења која произвођачи робота и независни произвођачи софтвера нуде програмерима и систем интеграторима за пројектовање и симулацију производних пројеката. Један број научно-истраживачких радова описује имплементацију и употребу паметних телефона за управљање и надзор робота. Даљински надзор се спроводи уз помоћ камере, приказивањем најзначајнијих вредности у текстуалном формату или уз помоћ 3Д анимације, док се програмирање углавном врши уз помоћ акцелерометра или жirosкопа, или попуњавањем текстуалних поља жељеним вредностима. Такве методе програмирања су временски захтевне за подешавање програма и генерисање жељеног извршавања, што посебно долази до изражаја када се често користи исти или сличан програм за извршавање. Такође, такве методе нису довољно интуитивне, поготово за потребе едукације у роботизи где студенти тек уче о основним елементима роботике.

У овом раду је представљена нова метода за даљинско програмирање, управљање и надзор робота независно од локације корисника на паметним телефонима која се фокусира на једноставност и интуитивност употребе окружења и минимизирање когнитивног оптерећења корисника уз одржавање способности за прецизно и ефикасно програмирање и управљање. Описана је имплементација и примена новог оригиналног софтверског система који примењује осмишљену методу. Употребом предложеног софтверског система је значајно унапређена интеракција између корисника и роботског система, као и приступ роботским лабораторијама. Верификација је извршена поређењем укупног броја акција корисника потребних за програмирање, даљинско управљање и симулацију пет тест програма применом предложеног решења и применом програмског језика и окружења код кога се интеракција са корисником базира на употреби *PC*-ја. Резултати поређења су показали да је предложено софтверско окружење утицало на значајно редуковање броја акција

корисника потребних за програмирање, даљинско управљање и симулацију програма за кретање робота.

Применом предложеног решења (које је названо у публикованим радовима *Robotics First*) у едукацији добила се могућност да се приступи иначе недоступним роботским системима, кроз интуитивне виртуелне лабораторије, независно од локације корисника, и да се тиме значајно унапреди квалитет преноса знања. Употребом широко распрострањених уређаја ниске цене процес опремања студената одговарајућом опремом је учињен економичним. Испитивањем искустава ученика трећег и четвртог разреда средње електротехничке школе, смер аутоматика, донети су закључци о употребљивости и корисности окружења *Robotics First*. Са становишта извршене евалуације закључује се да је окружење *Robotics First* интуитивно, кориснички оријентисано и лакше за употребу од окружења код кога се програмирање кретања робота обавља употребом програмског језика и PC-ја, као и да окружење *Robotics First* елиминише потребу за детаљним познавањем роботско-специфичних програмских парадигми, синтаксе и семантике програмског језика.

Кључне речи: даљинско управљање, надзор, интуитивно програмирање, бежична комуникација, индустријски роботи, Андроид, едукација, роботика, паметни телефони, мобилне платформе.

Научна област: Електротехника и рачунарство

Ужа научна област: Софтверско инжењерство

УДК број: 621.3

Title: Software system for remote control and monitoring of robots based on Android operating system and wireless communication

Abstract: Robot control systems are complex systems whose users must be well trained to use them. Control process is mainly carried out near the robot or by using wired connections. Because of rapid advances in technology, it is of great importance to implement new technology solutions in training environments, in robotics and aerospace industry. In contemporary robotics industry there is a constant need to improve accessibility of robotic systems and human-robot interaction, so the significance of developing new user interfaces that are intuitive and easy to use is growing. Therefore, for designing robot control user interfaces it is not sufficient to simply focus on providing detailed and complete control capability, but ergonomics of use should also be taken into account. Besides, massive and unreachable robots require control that is independent of the user's location.

Resources in the robotics field are limited for a common user to access and demand deep knowledge of robot-specific programming paradigms. For these reasons, it is not often possible for inexperienced students to become familiar with programming and control of complex robotic systems in a practical manner. Virtual wireless environments that provide realistic experience in robot control would be of great significance in the robotics education. It would be especially important in the case of application of such a solution for education in control of poorly accessible robots, such as devices for pilots' training and flight simulations. Also, in addition to educational purposes, the use of easily accessible, widespread and cost-effective device would be very profitable. The evaluation of the general engineering students' use and knowledge of smart phone technology, which was conducted within this dissertation, confirms that today's rapid development of technology requires improved methods for learning and education.

There are a number of scientific and research papers and reports that describe software solutions that the majority of robot manufacturers and independent software vendors offer to developers and system integrators for design and simulation of manufacturing projects. A certain number of science and research papers report the implementation and the use of smart phones for robot

monitoring and control. The remote monitoring of robot and machine performance is implemented by using camera, in textual form, or by using 3D animation, and robot programming is performed by using accelerometer, gyroscope or by filling out the text fields with desired values. Such programming methods are time-consuming for setting up and generating the desired robot performance, that is especially pronounced when frequently use the same or a similar program. Also, these methods are not sufficiently intuitive, especially for education in robotics and students that only begin to learn about the basic elements of robotics.

This dissertation presents a new method for remote robot programming, control and monitoring independently of user's location by using smart phones, which focuses on ease-of-use, intuitiveness and minimizing the users' cognitive load while maintains abilities for accurately and efficiently programming and control. It describes the implementation and the application of a new original software system that applies the proposed method. Human-robot interaction and access to the robotics laboratories is significantly improved and facilitated with the use of the proposed software system. The proposed solution was verified by comparing the total number of user actions required for the programming, remote control and simulation of five different robot motion test programs by using the proposed software system with the total number of user actions required for performing the same tasks by using the robot programming language and environment with the *PC*-based interaction. The results show that the proposed system significantly reduces the number of user actions required for programming, remote control and simulation of robot motion programs.

With the application of the proposed solution (in the published papers named Robotics First) in education, the ability to work with inaccessible robotics systems and equipment through the intuitive virtual laboratories, independent on the user's location, is provided and thereby the quality of gaining knowledge is significantly improved. With the use of widespread low-cost devices the process of equipping students with suitable equipment is made more economical. The usability of Robotics First was evaluated from the point of view of electrical engineering high school students, concerning its usability and usefulness. From this evaluation, it can be concluded that from the students' perspective Robotics

First environment is intuitive, user-friendly and easier to use than the environment in which the movement of robots is programmed by using programming language and PC, and that it eliminates the need of knowing robot-specific programming syntax and semantic paradigms.

Keywords: remote control, monitoring, intuitive programming, wireless communication, industrial robots, Android, education, robotics, smart phones, mobile platforms.

Scientific area: Electrical and Computer Engineering

Scientific subarea: Software Engineering

UDC number: 621.3

1.	УВОД	1
2.	ОСНОВНИ ПОЈМОВИ И ТЕРМИНОЛОГИЈА.....	6
2.1	Роботика и индустријски роботи.....	6
2.2	Кинематика манипулатора	8
2.3	Програмирање робота	9
2.4	Мобилно учење (<i>M-Learning</i>).....	10
3.	ЛОЛА РОБОТИ И ЛОЛА УПРАВЉАЧКИ СИСТЕМ БАЗИРАН НА УПОТРЕБИ <i>PC</i> -ЈА.....	12
3.1	Лола роботи.....	12
3.2	Лола управљачки систем базиран на употреби <i>PC</i> -ја за интеракцију са корисником.....	16
3.2.1	Окружење за програмирање и управљање Лола роботима базирано на употреби <i>PC</i> -ја за интеракцију са корисником.....	16
3.2.2	<i>L-IRL</i> језик за програмирање робота.....	19
3.2.3	Имплементација <i>L-IRL</i> окружења за програмирање.....	22
3.2.4	Тестирање парсера <i>L-IRL</i> језика.....	24
4.	ПРЕГЛЕД РЕЛЕВАНТНЕ ЛИТЕРАТУРЕ.....	27
4.1	Преглед постојећих софтверских решења за програмирање и управљање робота базираних на <i>PC</i> -ју.....	27
4.2	Преглед постојећих софтверских решења за програмирање и управљање робота уз употребу мобилних уређаја	28
4.3	Преглед постојећих софтверских решења за програмирање и управљање робота путем интернет претраживача.....	32
4.4	Преглед окружења за едукацију у роботизи	33
5.	ДЕФИНИЦИЈА ПРОБЛЕМА И ПРЕДЛОГ РЕШЕЊА	37

5.1	Дефиниција проблема.....	37
5.2	Класификација постојећих софтверских решења из области програмирања, управљања и надзора робота употребом мобилних уређаја	46
5.3	Класификација софтверских окружења за едукацију у роботизици.....	50
5.4	Критичка анализа постојећих решења	51
5.5	Полазне хипотезе	55
5.6	Предлог нове методе за даљинско програмирање, управљање и надзор робота.....	56
5.6.1	Прилагођење садржаја окружења за програмирање, управљање и надзор робота према карактеристикама и функционалностима мобилних уређаја.....	61
5.7	Предлог методе за едукацију у роботизици.....	64
6.	АНАЛИЗА ТЕХНОЛОГИЈА ЗА РАЗВОЈ СОФТВЕРСКОГ СИСТЕМА	67
6.1	Мобилне платформе	67
6.2	Андроид ОС.....	69
6.3	Технологије за бежичну и интернет комуникацију	72
6.3.1	<i>Bluetooth (IEEE 802.15.1)</i>	72
6.3.2	<i>Wi-Fi (IEEE 802.11)</i>	73
6.3.3	<i>ZigBee (IEEE 802.15.4)</i>	74
6.3.4	<i>3G и 4G</i> технологије	75
6.3.5	Протоколи транспортног слоја <i>TCP/IP</i> модела.....	76
6.3.6	Web технологије	77
6.4	Избор технологија за имплементацију софтверског система за даљинско програмирање, управљање и надзор робота.....	78
7.	ИМПЛЕМЕНТАЦИЈА СОФТВЕРСКОГ СИСТЕМА ЗА ДАЉИНСКО ПРОГРАМИРАЊЕ, УПРАВЉАЊЕ И НАДЗОР РОБОТА	81

7.1	Андроид окружење	81
7.1.1	Класе Андроид окружења	85
7.1.2	Преводалац роботских програма	88
7.1.3	ЗД анимација	92
7.2	Систем за рад у реалном времену (системски рачунар)	94
7.3	Комуникација између Андроид уређаја и системског рачунара	96
8.	ОПИС СОФТВЕРСКОГ СИСТЕМА ЗА ДАЉИНСКО ПРОГРАМИРАЊЕ, УПРАВЉАЊЕ И НАДЗОР РОБОТА	100
8.1	Програмирање робота	102
8.1.1	Програмирање индустријских робота	102
8.1.2	Програмирање уређаја за тренинг пилота и симулацију лета.....	104
8.2	Управљање и надзор робота у ручном режиму рада	106
8.3	Управљање и надзор робота у режиму рада „брзог бирања“ (“ <i>speed dial</i> ”)	107
8.4	Симулација и тестирање програмираних задатака.....	109
8.5	Даљински надзор робота.....	111
8.6	<i>Robotics First</i> – окружење за едукацију у роботици.....	112
9.	ПРИМЕНА И ВЕРИФИКАЦИЈА СОФТВЕРСКОГ СИСТЕМА ЗА ДАЉИНСКО ПРОГРАМИРАЊЕ, УПРАВЉАЊЕ И НАДЗОР РОБОТА	115
9.1	Даљинско програмирање, управљање и надзор индустријских робота.....	116
9.2	Симулација кретања уређаја за тренинг пилота и симулацију лета	135
10.	ПРИМЕНА И ЕВАЛУАЦИЈА ОКРУЖЕЊА <i>ROBOTICS FIRST</i> У ОБРАЗОВНОМ ПРОЦЕСУ	147
10.1	Примена окружења <i>Robotics First</i>	148
10.2	Евалуација окружења <i>Robotics First</i>	151

10.3	Дискусија.....	157
11.	ЗАКЉУЧАК	161
	ЛИТЕРАТУРА.....	169
	БИОГРАФИЈА АУТОРА	184

1. УВОД

У роботској индустрији постоји стална потреба за унапређењем приступачности и интеракције у раду са роботским системима. Брзе промене и напредак у развоју технологије у данашње време захтевају имплементацију нових технолошких решења у области програмирања и управљања индустријских робота и других робота специјалне намене. Поред тога масивни и слабо доступни роботи све више захтевају управљање независно од локације оператера.

У последњих неколико година паметни мобилни уређаји који су базирани на Андроид оперативном систему су нашли своју примену у великом броју области. Због могућности да се користе било где и било када, али и због своје портабилности и ефикасности при обављању различитих функционалности, употреба мобилних уређаја све више расте. Томе доприноси њихова ниска цена, али и могућности сталне адаптације. Они се користе у телевизији, за интернет апликације, али и у медицини [26], [70], [119], [124], хемијској индустрији [51], аутомобилској индустрији [50], [125], локализацији [65], туризму [98] и другим областима [80]. Постоје предвиђања да ће њихова употреба наставити да расте и да ће у наредним годинама број апликација које су намењене паметним телефонима и таблетима достићи четвороструко већи број у односу на број апликација намењених за *PC* (персоналне рачунаре) [8].

Андроид ОС (оперативни систем) је оперативни систем отвореног кода који је базиран на *Linux* кернелу и који је развијен од стране компаније *Google* [19], [126]. Андроид мобилни телефони и таблети пружају могућност употребе великог броја корисних сервиса и апликација као што је GPS навигација, алате за обраду различитих формата текстуалних докумената (*PDF*, *Word*), употреба апликација путем интернета (*e-mail*, *Facebook*), или телефонирање путем интернета (*Viber*). Употреба Андроид технологије у роботици и системима за програмирање и управљање робота може да

омогући интуитивно и ефикасно решење при управљању и програмирању оваквих система, чак и на удаљеној локацији.

У току 2014. године покривеност становништва мобилним телефонима је достигла 93 процената на глобалном нивоу [55]. Од укупног броја свих мобилних уређаја 85 процената чине паметни телефони, а предвиђа се да ће до 2017. године 70,5 процената свих уређаја за комуникацију бити паметни телефони [41]. Оваква експанзија у употреби паметних мобилних уређаја у комбинацији са интеракцијом уз помоћ екрана осетљивих на додир, сталним приступом интернету и великим бројем сервиса, доносе нове могућности за унапређење едукације увођењем мобилног учења (*Mobile Learning* или *m-Learning*) у различитим областима. У оквиру ове докторске дисертације спроведено је истраживање о генералној употреби и знању ученика средњих школа и студената о технологији паметних телефона. Истраживање је потврдило да данашњи брз развој технологије захтева унапређење метода за учење и едукацију.

Иако је употреба паметних телефона у роботици релативно нови концепт [111], постоји одређени број научно-истраживачких радова који описују софтверска решења која користе паметне телефоне за програмирање, симулацију и надзор индустријских процеса и рада роботских система [2], [4], [44], [58], [62], [66], [99], [102], [104], [111], [123], [137], [138]. Таква решења предлажу даљински надзор уз помоћ камере, приказивањем најзначајнијих вредности у текстуалном формату или уз помоћ 3Д анимације. Предложене методе програмирања роботских задатака у постојећим решењима се базирају на употреби акцелерометра или жirosкопа или на попуњавању текстуалних поља жељеним вредностима. Такве методе програмирања су временски захтевне за подешавање и генерисање жељеног извршавања, што посебно долази до изражаја када се често користи исти или сличан програм за извршавање. Такође, такве методе нису довољно интуитивне, поготово за потребе едукације у роботици где студенти тек уче о основним елементима роботике.

У овом раду је представљена нова метода за даљинско програмирање, управљање и надзор робота независно од локације корисника на паметним

телефонима која се фокусира на једноставност и интуитивност коришћења окружења и минимизирање когнитивног оптерећења корисника уз одржавање способности за прецизно и ефикасно програмирање и управљање роботима. Циљ истраживања и предложене методе је да унапреди интеракцију између оператера и роботског система приликом програмирања, управљања и надзора рада робота и да унапреди приступ роботским лабораторијама уз употребу широко распрострањене и приступачне технологије. Предложен је нови оригинални софтверски систем који примењује осмишљену методу [83]. Како би се верификовао рад предложеног решења, софтверски систем је примењен у индустријским условима и извршено је поређење броја акција оператера потребних за програмирање, даљинско извршавање и симулацију програма за кретање робота у односу на број акција оператера потребних за обављање истих задатака применом *L-IRL (Lola Industrial Robot Language)* језика за програмирање робота [112] и Лола система базираном на употреби *PC*-ја за интеракцију са корисником [90].

Дат је предлог примене предложене методе и софтверског система у области едукације у роботизици. Предложено решење је названо *Robotics First* [84]. Циљ предложеног решења је да се редукује време потребно за савладавање градива из области програмирања и управљања робота, као и да се редукује потреба за асистенцијом професионалаца, али и да се омогући флексибилна обука, као и анализа индустријских процеса из више перспектива. У оквиру ове дисертације извршена је евалуација предложеног решења са аспекта интуитивности и употребљивости решења приликом примене на практичној настави у Лола институту. Систем је испробало 24 ученика средње електротехничке школе, смера аутоматика, и евалуација је извршена попуњавањем унапред припремљених упитника.

Докторска дисертација садржи једанаест поглавља, скуп неопходних прилога и преглед коришћене литературе. У другом поглављу објашњени су основни појмови и терминологија, укључујући термине из роботике, али и из области едукације. У трећем поглављу описани су Лола роботи који су коришћени приликом примене и верификације предложеног софтверског

система и чији су модели коришћени у имплементацији симулатора и виртуелних лабораторија. У овом поглављу је такође описан Лола управљачки систем базиран на употреби *PC*-ја и *L-IRL* језика за програмирање кретања робота у оквиру кога је реализован део доприноса ове докторске дисертације и који је коришћен приликом верификације предложеног софтверског система за даљинско програмирање, управљање и надзор робота. У четвртом поглављу је направљен преглед постојећих решења из области програмирања и управљања и надзора робота употребом решења код којих је интеракција са корисником базирана на употреби *PC*-ја, затим решења код којих се интеракција са корисником базира на употреби паметних телефона, као и решења код којих се програмирање, управљање и надзор робота обавља кроз интернет претраживаче мобилних уређаја. У петом поглављу најпре је дата дефиниција проблема, а затим су дате класификација постојећих софтверских решења из области програмирања, управљања и надзора робота употребом мобилних уређаја и класификација постојећих софтверских решења из области едукације у роботизици. Након детаљне анализе релевантне литературе и критичке анализе постојећих решења дате су полазне хипотезе и предлог нове методе за даљинско програмирање, управљање и надзор робота. У овом поглављу обухваћено је истраживање услова за прилагођење садржаја окружења екранима мобилних уређаја и интеракцији употребом екрана осетљивих на додир. Затим је дат предлог методе за едукацију у роботизици. У шестом поглављу дата је анализа технологија за развој предложеног софтверског система, укључујући мобилне платформе и технологије за бежичну и интернет комуникацију. У седмом поглављу описана је имплементација и структура предложеног софтверског система. У осмом поглављу дат је детаљан опис предложеног софтверског система за даљинско програмирање, управљање и надзор робота. У деветом поглављу описана је примена предложеног софтверског система у индустријским условима. Предложено решење је примењено на даљинско програмирање, управљање и надзор индустријског робота Лола 15 и на програмирање и симулацију програма за кретање индустријског робота Лола 50, центрифуге моделоване као манипулатор са три степена слободе и

уређаја за просторну дезоријентацију пилота моделованог као манипулатор са четири степена слободе. У овом поглављу су дати резултати верификације предложеног софтверског система поређењем укупног броја акција корисника потребних за програмирање и извршавање, односно симулацију пет различитих тест програма за кретање робота применом предложеног софтверског система базираног на Андроид оперативном систему и применом *L-IRL* програмског језика и Лола окружења код кога се интеракција са корисником базира на употреби *PC*-ја. У десетом поглављу је описана примена предложеног софтверског система (*Robotics First*) у оквиру практичне наставе у Лола институту и дата је евалуација са аспекта ефикасности и интуитивности употребе предложеног решења при програмирању, управљању и симулацији кретања индустријских робота. У једанаестом поглављу дат је закључак, изложени су резултати и доприноси рада, као и правци могућег даљег истраживања. На крају је дат списак коришћене литературе.

2. ОСНОВНИ ПОЈМОВИ И ТЕРМИНОЛОГИЈА

2.1 Роботика и индустријски работи

Роботика је мултидисциплинарна и интердисциплинарна наука и технологија која се бави истраживањем, развојем, пројектовањем и применом робота. То је област која обухвата примењене инжењерске науке (машинство, производно машинство, електротехника, електроника), рачунарске науке као и математику и механику [96].

Порекло речи *робот* потиче од чешке речи "робота" што значи присилан рад. Ова реч је први пут уведена од стране чешког писца научне фантастике Карела Чапека. Према дефиницији која се може наћи у Вебстеровом (*Webster*) речнику, робот је "*аутоматизовани уређај који обавља функције које се обично приписују човеку*". Званична и прецизнија дефиниција дата од стране RIA (*Robotic Industries Association*), која се односи на индустријске роботе, у слободнијем преводу гласи [116]:

"Индустријски робот је вишефункционални манипулатор који се може репрограмирати и који је намењен да помера радни материјал, предмете, алат и специјалне уређаје на разне задате начине у циљу извршавања различитих задатака".

Механички систем робота назива се још и механичка структура, механизам робота или манипулатор [96]. Овај систем има основну функцију успостављања просторних односа између енд-ефектора (завршног уређаја) и радног објекта. Сваки механизам се састоји из покретних чланова (сегмената) и непокретних чланова (постоља). Кинематички пар подразумева два међусобно повезана тела. Низ крутих тела или сегмената спојених кинематичким паровима образују кинематички ланац.

Тело у простору може имати највише шест степени слободe, три трансляције и три ротације, тако да и робот обично има до 6 степени слободe (чланова), мада може имати и додатне (редундантне) степене слободe. На крајњем члану робота се налази енд-ефектор.

Управљачки систем робота је систем који омогућава меморисање, одвијање тока програма, везу са периферним уређајима, управљање и надгледање извршавања појединих функција робота. На данашњем нивоу развоја, управљачки системи су готово увек засновани на примени рачунара [96].

Два основна типа кретања робота који су тесно везани и за типове постављених задатака су [73]:

- кретање тачка по тачка, и
- кретање по програмираној путањи.

Кретање тачка по тачка (*Point To Point - PTP*) је временски оптимално кретање између две тачке у простору. Чланови робота крећу се синхроно од почетних до крајњих положаја, што резултује закривљеном путањом врха енд-ефектора у простору, која зависи од врсте, величине и положаја чланова робота. Сви чланови истовремено започињу и завршавају кретања. *PTP* кретања служе за брзо долажење робота у циљну тачку, где треба да се изврши или да отпочне нека роботска операција.

Код кретања по програмираној путањи (*Controlled Path* или *Continuous Path - CP*) врх енд-ефектора се креће по математички дефинисаној путањи која може бити права линија, кружница, парабола, елипса или слично. Код ових кретања се у планеру путање најпре одреди једначина програмиране путање, која се затим интерполира, тако да се оствари задата промена брзине енд-ефектора по њој. Овде се програмира и промена оријентације енд-ефектора од почетне до крајње тачке. Затим се инверзном кинематиком одређују узастопни положаји чланова, односно мотора робота. Овакав начин кретања неопходан је за извршавање задатака као што су електролучно заваривање, фарбање или машинска обрада.

2.2 Кинематика манипулатора

Манипулатор се може моделирати као прост отворен кинематички ланац који се састоји од неколико сегмената повезаних ротационим или транслаторним зглобовима [96]. Први сегмент се везује за непокретан члан (постоље), док се за последњи везује енд-ефектор. Дакле, постоји n покретних сегмената и n зглобова са по једним степеном слободе.

Унутрашње или генерисане координате: $q_i(t)$, $i=1, \dots, n(6)$, дефинишу релативна померања сегмената, односно зглобова. Унутрашње координате могу бити углови или дужине. То су непрекидне, два пута диференцијабилне, временске функције у опсезима који су одређени механичким ограничењима:

- $q_i(t)=\theta_i$ – када су зглобови ротациони,
- $q_i(t)=a_i$ – када су зглобови транслаторни.

Ове скаларне величине могу бити организоване у векторском облику тако да можемо успоставити вектор унутрашњих координата:

$$\mathbf{q}=[q_1(t) \ q_2(t) \ \dots \ q_n(t)], \ i = 1, 2, \dots, 6; \quad (1)$$

Вектор спољашњих координата дефинише позицију и оријентацију енд-ефектора. Позиција енд-ефектора се најчешће описује Декартовим координатама (X, Y, Z) , а оријентација помоћу углова закретања $(A, B$ и $C)$ између оса координатног система везаног за завршни уређај и непокретног координатног система. Углови којима се задаје оријентација су најчешће RPY (*roll, pitch* и *yaw*) или Ојлерови углови.

Директном кинематиком се израчунава позиција и оријентација енд-ефектора (спољашње координате) на основу позиција чланова робота (унутрашњих координата). Много сложенији је инверзни кинематички проблем који представља одређивање вектора унутрашњих координата за задати вектор спољашњих координата. Код управљања CP кретањима се израчунавање инверзне кинематике врши у сваком периоду интерполације, док се код управљања PTP кретањима то не врши, већ се ту интерполира путања сваког члана робота [73].

2.3 Програмирање робота

Програмабилност је једна од основних карактеристика индустријског робота [96]. Програмабилност или могућност репрограмирања значи да се програмирана кретања и помоћне функције могу мењати без физичке промене односно интервенције на самом роботу.

Методe програмирања обучавањем или “*on-line*” методe су методe у којима се сам робот користи при програмирању и чији се жељени покрети меморишу и касније репродукују. То су једноставне методe и од програмера не захтевају познавање програмских језика већ вештину за извршавање програмираног задатка. Ово су данас веома заступљене методe али нису погодне за комплексне програме, а уз то задржавају робот док се врши програмирање.

У индустријама и технологијама као што су аутомобилска и авиоиндустрија где је потребно програмирати на стотине и хиљаде тачака уз задржавање целе линије, програмирање обучавањем није повољно. Са обзиром на то да сви подаци (координате) тачака већ постоје на цртежима или геометријском моделу, логично је користити текстуалне језике (*off-line* методу програмирања).

Језици за програмирање и управљање робота представљају скупове инструкција за кретање и помоћних функција, којима се дефинише задатак роботског система. Они се могу поделити на следеће нивое [73]:

1. језици опште намене,
2. роботски језици на нивоу управљања кретањем тачка по тачка,
3. роботски језици којима је омогућено управљање не само кретањима тачка по тачка, већ и кретањима по правој линији или кружном луку,
4. структурирани, односно виши програмски језици за програмирање робота,
5. имплицитни језици, код којих се програмирање врши на нивоу задатка.

Уобичајено је да се језици за роботе по начину реализације деле на:

- интерпретерске и
- компајлерске.

У првом приступу интерпретер прихвата програм директно у изворном коду, а затим га интерпретира. У другом приступу се изворни код преводи у међу-код који се затим машински интерпретира. Први приступ је повољнији са аспекта приступа програма, док је други погоднији са аспекта брзине, због бржег интерпретирања.

2.4 Мобилно учење (*M-Learning*)

Мобилно учење или М-учење (*Mobile Learning, m-Learning*) представља надоградњу електронског учења (*e-Learning*), у коме се за учење користе мобилни уређаји и бежичне комуникационе технологије [13]. Мобилно учење може да унапреди електронско учење померајући границе изван затвореног простора и омогућавајући ученицима да унапреде своје вештине у било ком тренутку и независно од њихове локације. Због тога мобилни уређаји могу бити моћан алат за студенте и ученике у ситуацијама када желе да прикупе информације и унапреде већ стечено знање [16]. Рад [18] говори о томе да употреба мобилних уређаја може да охрабри студенте да учествују у различитим курсевима и да им помогне у учењу. Кључне карактеристике мобилног учења су свеприсутност, портабилност, мала величина уређаја, приватност, интерактивност, могућност сарадње, као и способност достављања инстант порука. Мобилно учење се базира на интересима ученика, њиховом искуству и потребама, и омогућава им да буду на правом месту у право време, тамо где им највише одговара да уче [110]. Мобилно учење се заснива на предавању и учењу кроз било који тип мобилног уређаја, било да је то паметан телефон, *PDA (Personal Digital Assistant)*, конзола, таблет *PC, iPad* [27] или било који други уређај који обезбеђује бежичну комуникацију и који се може користити у покрету. Све већа употреба ове технологије је довела до развоја све већег броја система за мобилно учење на свим образовним нивоима [39], [127], као допунско средство које унапређује интеракцију између студента и предавача, мотивацију студената и сам

процес учења. Радови [101] и [149] говоре о томе да студенти 21. века виде мобилно учење као важну допуну електронском учењу.

3. ЛОЛА РОБОТИ И ЛОЛА УПРАВЉАЧКИ СИСТЕМ БАЗИРАН НА УПОТРЕБИ РС-ЈА

3.1 Лола роботи

Лола 15 је индустријски робот зглобне конфигурације са 6 степени слободе, носивости 15 kg (слика 3.1). Развијен је у Лола фабрици робота и Лола институту ради извршавања експеримената везаних за развој система за управљање кретањима робота. Намењен је за апликације заваривања (електролучно, гасно), гасног сечења, метализације, манипулације и монтаже.



Слика 3.1 Индустријски робот Лола 15

Лола 50 је индустријски робот зглобне конфигурације са 6 степени слободе, носивости 50 kg (слика 3.2). Развијен је у Лола институту, а

произведен у Лола ФАМ-у. Наменен је за апликације тачкастог заваривања, фарбања, манипулације материјалом, чишћењу одливака, брушења, полирања и за разне друге примене.



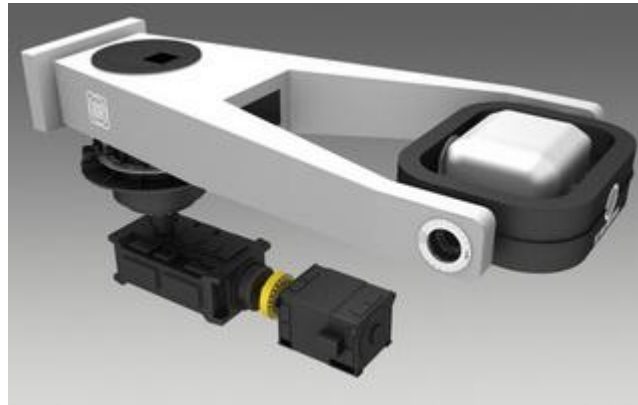
Слика 3.2 Индустијски робот Лола 50

Лола институт се такође бави развојем следећих врста уређаја за тренинг пилота борбених авиона:

- Центрифуга за симулацију лета борбених авиона (Слика 3.3),
- Уређај за просторну дезоријентацију пилота (Слика 3.4).

Модерни борбени авиони могу да врше убрзања крећући се у разним правцима у односу на координатни систем пилота, посебно током такозваних суперманевара. Ови "агилни" авиони врше неконвенцијалне летове са великим угловима напада и кретањима око све 3 осе авиона и убрзањима до 9 G, а са променама убрзања до 9 G/s. Из овог разлога је потребно испитати утицај ефеката дејства великих убрзања и великих промена ових убрзања на психологију пилота и његову способност да управља авионом у тим условима. Центрифуга се користи за поуздано

генерисање велике почетне промене силе убрзања G и постизање и одржавање велике силе убрзања G , што је потребно ради реакција тестирања пилота на те услове.



Слика 3.3 Центрифуга за симулацију лета борбених авиона



Слика 3.4 Уређај за просторну дезоријентацију пилота

Центрифуга има облик 3-осног манипулатора са обртним осама, код кога се глава (или груди код неких тренинга) посматра као енд-ефектор. Обртање руке центрифуге око вертикалне осе (планетарно кретање) је главно кретање које остварује жељену силу убрзања. Центрифуга мора да оствари брзину, убрзање и промену убрзања одговарајућим обртањем руке

око вертикалне осе. Рука носи кардански улежиштену гондолу која има 2 осе које обезбеђују могућности пропињања (*pitch*) и ваљања (*roll*). Оса ваљања лежи у равни обртања руке, управно на осу главне ротације, односно у правцу x -осе. Оса пропињања (y) је управна на осу ваљања. Задатак оса ваљања и пропињања центрифуге је да усмере силу убрзања у жељени правац и смер.

Модерни борбени авиони могу заузети врло неуобичајене оријентације. Због тога је потребно испитати способност пилота да препозна ту оријентацију и да се прилагоди најразличитијим положајима у којима може да се нађе. Потребно га је уверити да не треба да се ослања на свој осећај већ да треба да доноси закључак о тренутној оријентацији авиона на основу информација са инструмената авиона.

Уређај за просторну дезоријентацију пилота или дезоријентатор је дизајниран као манипулатор са 4 ротационе осе. Закретањима тих оса постижу се различите оријентације пилота. Осим тога, на уређају се могу симулирати и различите силе убрзања које делују на пилота. Глава или груди пилота се посматрају као енд-ефектор. Ротација руке око вертикалне (планетарне) осе је главно кретање. Рука носи жироскопски улежиштену гондолу са 3 ротационе осе које остварују закретање (*yaw*), пропињање (*pitch*) и ваљање (*roll*). Оса закретања (z) је паралелна са руком пилота. Оса ваљања је у равни ротације руке, управна на главну обртну осу, односно у правцу x -осе. Оса пропињања (y) је управна на осу ваљања.

3.2 Лола управљачки систем базиран на употреби РС-ја за интеракцију са корисником

3.2.1 Окружење за програмирање и управљање Лола роботима базирано на употреби РС-ја за интеракцију са корисником

Систем за програмирање и управљање Лола роботима и уређајима за тренинг пилота и симулацију лета код кога је интеракција са корисником базирана на употреби РС-ја имплементиран је за *Linux* оперативни систем. Структура развијеног система је приказана на слици 3.5. Лола управљачки систем (Лола УС) се састоји од:

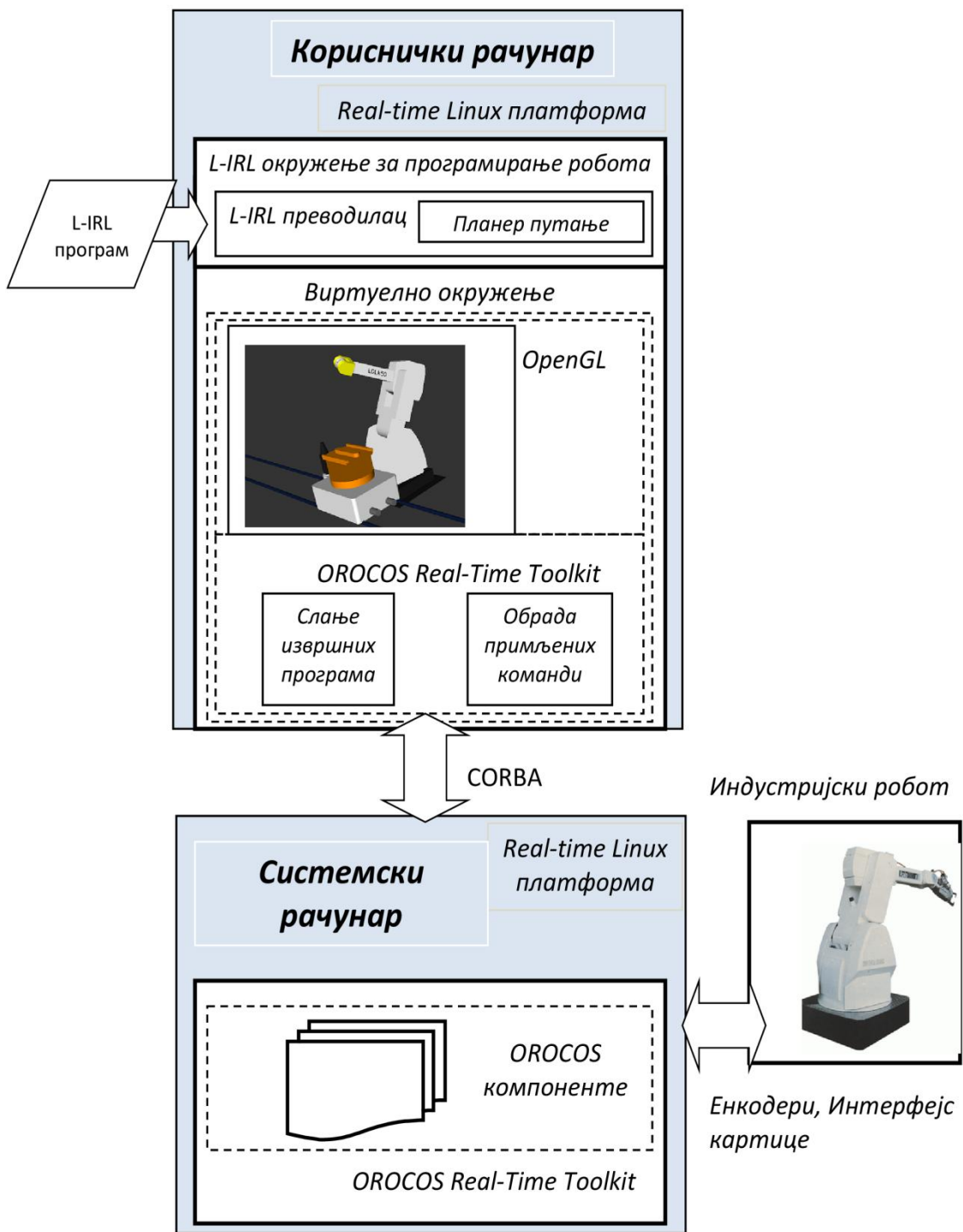
- Корисничког рачунара и
- Системског рачунара.

Кориснички систем за програмирање индустријских робота и уређаја за тренинг пилота и симулацију лета се састоји из *L-IRL* окружења за програмирање кретања робота и виртуелног окружења. *L-IRL* окружење за програмирање кретања робота садржи преводаца који генерише извршне програме у форми објектног кода. Улаз *L-IRL* преводиоца представља програм написан на *L-IRL (Lola Industrial Robot Language)* програмском језику. *L-IRL* преводац, на основу датог *L-IRL* програма, генерише датотеку која садржи објектни код, односно бинарни приказ параметара којима се описује жељено кретање, са екстензијом *“.run”*. *Run* датотека са објектним кодом се генерише на основу алгоритама кретања и кинематике робота који су укључени у сам преводац и који су дефинисани у оквиру планера путање [72], [73], [74]. Он се даље шаље на системски рачунар, где се извршава.

Кориснички рачунар омогућава проверу да ли, према написаном програму на *L-IRL* језику, робот излази из свог радног простора, да ли су осе прекорачиле ограничења, омогућава приказивање планираних путања енд-

ефектора и чланова робота и приказивање дијаграма њихових брзина и убрзања, као и 3Д симулацију кретања робота. У току превођења програма генеришу се табеле које садрже вредности углова у току кретања, за сваку осу робота, на основу којих се извршава симулација у оквиру виртуелног окружења корисничког система.

Поред симулације виртуелно окружење корисничког система обавља слање команди за извршавање роботских програма у форми објектног кода и прима податке о тренутним позицијама робота. Пренос података се обавља путем *CORBA* протокола [88], [97]. *CORBA* је акроним за *Common Object Request Broker Architecture*, архитектуру која је дефинисана од стране *OMG (Object Management Group)* са циљем да се премости јаз између програма писаних на различитим програмским језицима, који се извршавају на различитим рачунарима са различитим оперативним системима и уз то повезаних различитим мрежним протоколима. На системском рачунару се дати програм интерпретира и на основу добијених података извршава се кретање робота. Кориснички рачунар путем *CORBA* протокола добија повратне информације о положају робота, односно уређаја за тренинг пилота и симулацију лета, у реалном времену од системског рачунара.



Слика 3.5 Структура Лола управљачког система базираног на употреби PC-ја за интеракцију са корисником

Приликом имплементације окружења на корисничком рачунару коришћено је:

- *Qt* развојно окружење [60] за креирање графичког корисничког интерфејса *L-IRL* едитора и симулатора;
- *OpenGL* библиотеке [145] за цртање и повезивање геометријских примитива које формирају виртуелне моделе робота, као и за транслацију и ротацију делова робота;
- *OROCOS* софтвер и библиотеке, за процесирање пристиглих команди од стране системског рачунара [109].

Системски рачунар је управљачки систем за рад у реалном времену (*real-time*) и састављен је од неколико нивоа [33], [35]. Најнижи ниво чини *real-time Linux* оперативни систем чији је кернел надграђен са *real-time* окружењем отвореног кода *Xenomai*. *OROCOS Real-Time Toolkit (RTT)* представља средњи слој који се налази између оперативног система и апликативног слоја који обезбеђује инфраструктуру и функционалности за имплементацију роботских *real-time* апликација на C++ програмском језику. Самим тим омогућено је подешавање и имплементација компонената за рад у реалном времену које уједно представљају највиши ниво управљачког система. Окружење на системском рачунару се састоји од компонената које укључују имплементацију интерпретера извршних датотека за кретање робота, као и алгоритама за кретање робота развијених у Лола институту [72], [73], [74].

3.2.2 *L-IRL* језик за програмирање робота

L-IRL (Lola Industrial Robot Language) је језик структурираног типа и припада класи проблемски оријентисаних језика [112]. Програмске целине се састоје из декларативног дела и дела са наредбама. Један програм представља целину која се може засебно преводити и извршавати. Програми се могу повезивати у главни програм. Одређене програмске целине је могуће увозити у друге програме одговарајућим декларацијама. Садржи наредбе за програмирање кретања робота и наредбе за програмирање логичког тока

програма. *L-IRL* омогућава рад са променљивама, рад са аритметичким, логичким и геометријским изразима, као и рад са сигнаlima из окружења.

L-IRL је пројектован тако да буде употребљив у индустријским условима. Прилагођен је проблему програмирања роботских операција, које могу укључивати рад једног робота са пратећим сензорима, али и кооперативни рад више робота и других уређаја.

Структура језика *L-IRL* одговара општем програмском језику PASCAL са одређеним модификацијама. Као узор приликом реализације модуларности и роботских специфичних програмских конструкција коришћен је предлог стандарда за роботске језике IRL (DIN 66312).

L-IRL садржи наредбе за синхронизацију са догађајима из окружења, као и наредбе за контролу програмског тока, позиве процедура и функција, петље и наредбе за задавање кретања. Типови података могу бити једноставни, структурирани или геометријски. Описивање положаја робота и објеката је омогућено геометријским типовима података и геометријским низовима. Геометријским операторима је омогућено сабирање и одузимање вектора, релативна translација и ротација координатних система и слично. Исказима кретања је омогућено програмирање путања робота у координатама основе робота (*MOVE*) и координатама алата, односно енд-ефектора (*MOVE_INC*). Кретање робота може бити: тачка по тачка (*Point-to-Point - PTP*) и кретање по математички дефинисаним путањама (*CP*). Математички дефинисане путање по којима је могуће задавање кретања робота су линија (*LIN*) и круг у простору (*CIRCLE*). Повезивање два неколинеарна праволинијска кретања могуће је извршити апроксимативним *CP* кретањем по параболи. За везу са окружењем користе се променљиве које се називају сигнали, а за везу са сензорима користе се логички канали.

L-IRL језик садржи стандардне, геометријске и структуриране типове података. Подржани су сви стандардни типови података који се срећу у општим језицима (*int, real, string, bool, char, string*). Структурирани типови података су омогућени у облику записа и низова. Дефинисани су геометријски типови података који служе за описивање просторних величина:

- *Position* – описује позицију у милиметрима координатног система у простору релативном на референтни координатни систем. Представља се са три реална броја која описују вредности X , Y и Z координата.
- *Orientation* – се представља са три реална броја која описују углове оријентације A , B и C у степенима координатног система у простору релативном на референтни координатни систем.
- *Pose* – се састоји од компонената *pos* и *ori*, које су типа *position* и *orientation*, респективно и описује позицију робота у спољашњим координатама.
- *Mainjoint* – описује позицију и оријентацију робота у унутрашњим координатама.
- *Addjoint* – дефинише додатне осе робота.
- *Joint* – се састоји од компонената *mjoint* и *add_joint*, које су *mainjoint* и *addjoint* типа, респективно
- *Robtarget* – описује позицију и оријентацију енд-ефектора и статус оса робота.

Брзина и убрзање кретања се специфицира параметрима *speed_ptp* и *acc_ptp* за *PTP* кретања, односно *speed* и *acc* за *CP* кретања, док се оријентација, која може бити променљива или фиксна, специфицира параметром *c_ori*. Уколико се овај параметар не наведе, подразумева се променљива оријентација.

PTP и *CP* инструкцијама кретања је могуће програмирати кретања индустријских робота различитих конфигурација. *PTP* инструкцијама је могуће програмирати кретање уређаја за просторну дезоријентацију пилота.

За потребе програмирања и управљања центрифуге која је моделирана као троосни манипулатор са ротационим зглобовима, у *L-IRL* језик су уграђене специфичне инструкције кретања тзв. *GMOVE* инструкције. Овим инструкцијама се задаје жељено убрзање које делује на пилота, прираштај овог убрзања или време одржавања константног убрзања у режиму отворене петље. Прираштај убрзања може бити позитиван или негативан. Жељено убрзање које делује на пилота се специфицира

параметром *bl_g*, прираштај убрзања параметром *acc_g*, а време одржавања константног убрзања *time* параметром.

Када се програмирају кретања потребно је навести и назив робота на кога се кретање односи дефинисањем параметра *act_rob* или навођењем системске спецификације робота у заглављу програма (*system_specification*).

Пример програма за кретање робота Лола 50 написаног на *L-IRL* језику који садржи неколико различитих типова инструкција кретања и типова података је дат у следећем тексту.

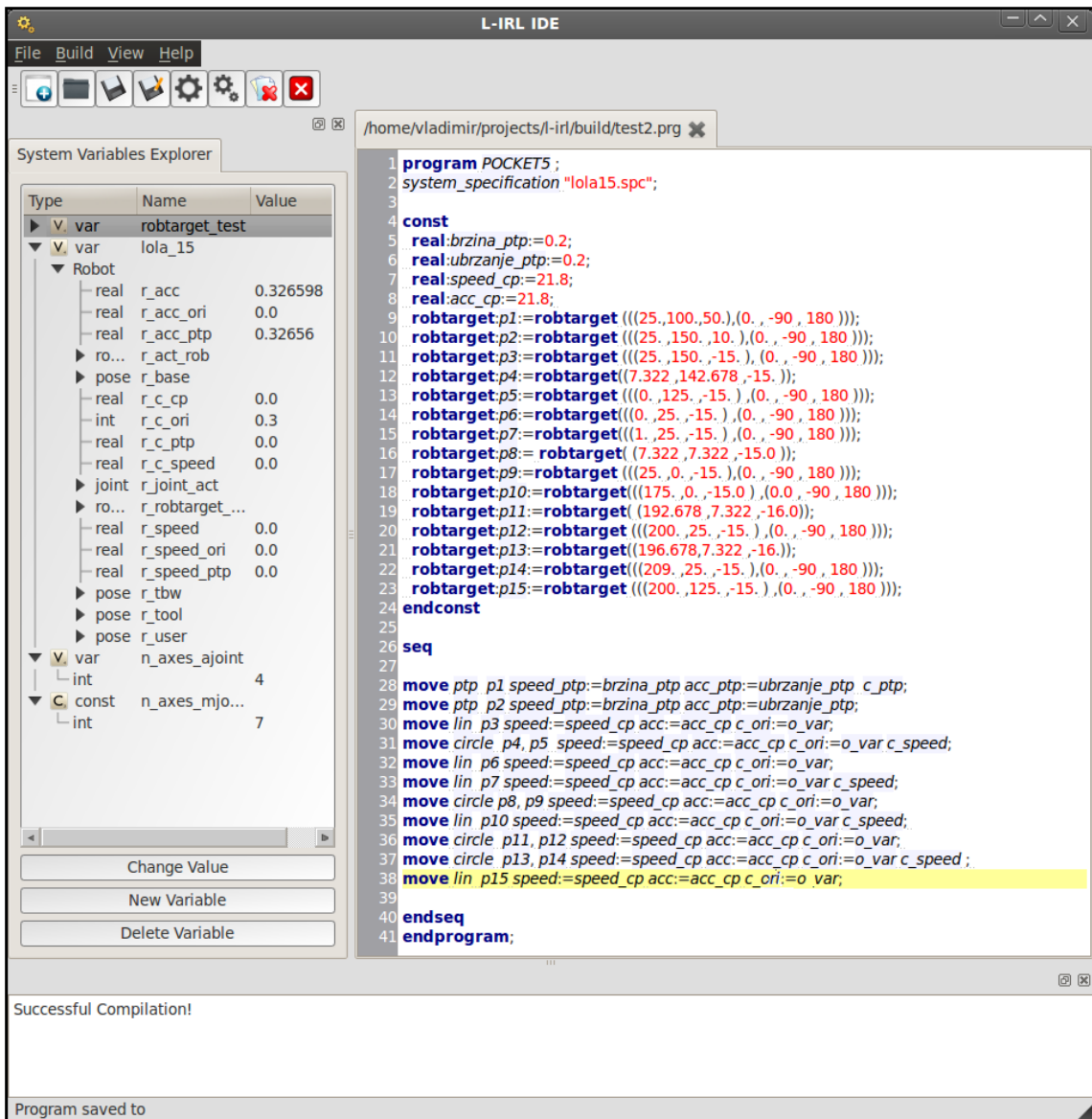
```
program lolatest;
seq
  move ptp joint(mainjoint(90, 0, 45, 90, 45, 90)) speed_ptp :=
0.8 acc_ptp := 0.1 act_rob:"lola_50";
  move ptp joint(mainjoint(95, 20, 45, 90, 20, 90)) speed_ptp :=
0.9 acc_ptp := 0.5 act_rob:"lola_50";
  move lin robtaraget (pose(position (0.1, 0.1, 0.1), orientation(-
19.8, -67.6, -163))) speed:= 1.4 acc := 10.6 c_ori:=o_fix
act_rob:"lola_50";
  move_inc circle robtaraget(position(-0.1, 0.1, 0.5)), robtaraget
(pose(position (0.1, 0.1, 0.1), orientation(-19.8, -67.6, -163)))
speed:=0.4 acc:=0.6 c_ori:=o_fix act_rob:"lola_50";
  move ptp joint(mainjoint(90, 0, 45, 90, 45, 90)) speed_ptp :=0.8
acc_ptp := 0.1 act_rob:"lola_50";
endseq
endprogram;
```

3.2.3 Имплементација *L-IRL* окружења за програмирање

Графички кориснички интерфејс едитора за програмирање робота на *L-IRL* програмском језику је приказан на слици 3.6. Графички кориснички интерфејс је развијен употребом C++ програмског језика и *Qt* библиотека [60]. Парсер језика је имплементиран употребом алата *Bison* и *Lex. Flex (fast lexical analyzer generator)* је алат за генерисање скенера, односно програма који препознаје лексичке обрасце у тексту. *Flex* чита улазне датотеке и даје

опис скенера који генерише [132]. Опис је представљен регуларним изразима у С коду, названим правилима. Он се често користи са генератором парсера *Bison* [5]. *Bison* представља генератор парсера који претвара опис *LALR (Look Ahead Left to Right)* бесконтекстне граматике у С програм за парсирање. Улаз у *Bison* представља секвенца токена која припада језику описаним том граматиком. Описом граматике се специфицирају синтаксне групе и дају правила за њихово конструисање из делова.

Семантичка анализа и табела симбола, генерисање објектног кода и његова оптимизација је имплементирана на С++ програмском језику. У претходној верзији овог окружења, парсер и преводацац су били развијени употребом методе рекурзивног спуста. У новој верзији парсирање је имплементирано употребом *LALR (Look Ahead Left to Right)* алгоритма, који је директно инкорпориран у алат *Bison*. Употребом овог софтверског алата заједно са методологијом објектно-оријентисаног програмирања постигло се ефикасно структурално и логичко рефакторисање кода, а поступак проширивања језика новим инструкцијама је значајно поједностављен у односу на претходну верзију окружења за програмирање базираног на *PC*-ју [112].



Слика 3.6 Едитор за програмирање робота на *L-IRL* језику

3.2.4 Тестирање парсера *L-IRL* језика

Технике и алати за генерисање тест реченица који се могу користити за аутоматизовано тестирање парсера са применом на тестирање парсера *L-IRL* језика су описани у радовима [85] и [86]. Обављено је тестирање парсера језика *L-IRL* за програмирање робота и то на основу тест реченица генерисаних употребом *Grow* алгоритма и употребом *Purdum* алгоритма [118]. Тестирање је обављено за исправну граматiku језика, као и за намерно

модификовану (покварену) граматику *L-IRL* језика, коришћењем оба алгорита.

Главна идеја код методологије генерисања позитивних и негативних тестова за тестирање парсера лежи у томе да се модификује (мутира) оригинална грамика како би се добила грамика која описује језик који је веома сличан оригиналном, али није идентичан. Такве измењене грамике представљају улазни тест генератора, који затим производи потенцијално негативне тестове [147].

За тестирање је коришћен *Forson* алат [38]. Улазна датотека у току генерисања тест примера садржала је *Bison* граматику *L-IRL* језика. У току тестирања коришћена је датотека *l-irl.lexicon* са листом лексичких вредности које су повезане са токенима циљног језика. Квантитативни подаци о граматици језика *L-IRL* (број смена и токена), као и подаци о примењеним мутацијама и генерисаним позитивним и негативним тестовима за *Purdom* и *Grow* алгоритам су приказани у табели 3.1.

Табела 3.1. Подаци о *L-IRL* граматици и поступку тестирања

Број смена грамике	83
Број токена грамике	110
Број примењених мутација	2
Број позитивних тест примера генерисаних <i>Purdom</i> алгоритмом	15
Број позитивних тест примера генерисаних <i>Grow</i> алгоритмом	10
Број негативних тест примера генерисаних <i>Purdom</i> алгоритмом	15
Број негативних тест примера генерисаних <i>Grow</i> алгоритмом	10

Генерисање негативних тест примера је обављено коришћењем *Grow* и затим коришћењем *Purdom* алгоритма. Грамика је намерно “покварена” (мутирана) са циљем да се генеришу синтаксно неисправне реченице.

Негативни тест примери су добијени једноставним мутацијама оригиналне граматике *L-IRL* језика, јер према [108] ако су детектоване једноставне грешке, онда ће и компликованије бити детектоване.

Позитивни и негативни тест примери, добијени коришћењем *Grow* алгоритма, као и коришћењем *Purdom* алгоритма, су извршени у оквиру окружења за програмирање робота у *L-IRL* језику и резултати су упоређени са очекиваним. Добијени резултат за све позитивне тест примере је успешно парсирање, без јављања било каквих синтаксних грешака. За потенцијално негативне тест примере преводилац је пријављивао очекиване синтаксне грешке.

У случају примене *Grow* алгоритма на граматiku која има већи број рекурзивних правила, као што је случај са *L-IRL* граматиком, може доћи до генерисања јако великих, као и бесконачних, тест реченица. Алат *Forson* додељује исту учестаност избора свим правилима нетерминалног симбола. Једини начин да се ово избегне је експлицитно додавање копија правила за неки нетерминални симбол у *Bison* улазну граматiku. Ово није елегантно решење, па даљи рад на овом алату има за циљ имплементацију сета контроле, као што је на пример провера заузећа стека, са циљем да се ограничи или чак специфицира жељена величина генерисаних тест реченица [38].

Тестирање употребом *Purdom* алгоритма омогућена је потпуна покривеност свих правила граматике. Овај алгоритам се може употребити за тестирање и мањих и већих граматика, али ипак гарантује свеобухватност само у случају његове примене на мање граматике неког језика, као што је *L-IRL* граматика.

4. ПРЕГЛЕД РЕЛЕВАНТНЕ ЛИТЕРАТУРЕ

4.1 Преглед постојећих софтверских решења за програмирање и управљање работа базираних на PC-ју

Већина произвођача работа и независних произвођача софтвера нуде графичке корисничке интерфејсе базиране на PC—ју у којима је могуће пројектовати и симулирати производне процесе [115]. Софтвер за симулацију и *off-line* програмирање работа, *RobotStudio* [20], омогућава програмирање работа на PC-ју без потребе прекидања производње. *RobotStudio* пружа могућност програмирања кретања работа, едитовања и дебаговања, као и симулације и оптимизације постојећих роботских програма. Често се користи на универзитетима за едукацију инжењера. Један пример симулатора представља *KUKA SimPro* – софтвер за симулацију базиран на *KUKA* роботу, који укључује *KUKA OfficeLite* софтвер за генерисање команди за *KUKA* робот [61], [122]. *Roboguide* је алат за симулацију рада и радног окружења работа базиран на PC-ју [144]. Има могућност провере интерференције између работа и других објеката, могућност симулације различитих управљачких операција и могућност надзора тренутног стања работа. У лабораторији *Human-Machine Interfaces and Intelligent Systems Laboratory* на Универзитету Отава, Канада развијен је систем за даљинско управљање работа са шест степени слободе базиран на употреби стерео камера повезаних на PC [67]. У систему се реконструисано кретање руке оператора користи за управљање енд-ефектором работа у реалном времену, док се надзор врши на основу садржаја прикупљеног са камера у роботској лабораторији, а комуникација обавља жичном везом. У раду [142] описана је платформа за управљање и надзор мобилног работа са пет степени слободе, где се даљинско управљање обавља путем интернета, употребом PC-ја, а даљински надзор посматрањем видео снимка у реалном времену и

приказивањем информација са сензора. Систем за даљинско управљање мобилног робота под називом *ROBUTER_ULM* је описан у раду [9]. Код *ROBUTER_ULM*-а интеракција са корисником се обавља употребом хост рачунара, а комуникација са *embedded* рачунаром - сервером се врши бежично, у оквиру локалне рачунарске мреже. Један предлог роботског система за учествовање у спасилачким мисијама је описан у раду [131]. Систем се састоји од *PC*-ја и програмабилног логичког контролера који представљају клијент, односно сервер, и обезбеђује бежичну интеракцију између корисника и машине, укључујући и надзор. Рад [59] описује интерактивни алат за моделовање и симулацију рада индустријских робота, алгоритама базираних на употреби рачунарског вида (*Computer Vision*) и даљинских операција. Описани алат базиран је на употреби Јава програмског језика и намењен је имплементацији напредних роботских апликација и апликација базираних на рачунарском виду. Алат се састоји од два главна дела: (1) Јава библиотека под називом *EjsRL*, која обезбеђује окружење за моделовање и симулацију роботских и система базираних на рачунарском виду и (2) *EJS*, алата отвореног кода који обезбеђује подршку за графички интерфејс. Даљинске операције су обезбеђене кроз комуникацију између клијента и сервера употребом *HTTPS* протокола који обезбеђује безбедну комуникацију.

4.2 Преглед постојећих софтверских решења за програмирање и управљање робота уз употребу мобилних уређаја

Постоји одређени број решења за програмирање и управљање робота употребом паметних телефона и *Bluetooth* технологије [44], [62], [102], [104], [138]. У раду [138] описан је систем за комуникацију са артикулисаним (зглобним) роботом употребом Андроид уређаја и *Bluetooth* комуникације. Употребом овог система корисник може да пошаље жељене команде роботу попуњавањем одговарајућих текстуалних поља и кликом на дугме за слање

на екрану осетљивом на додир. Рад [62] предлаже систем за надзор температуре окружења употребом мобилног робота, Андроид уређаја и *Bluetooth* комуникације. Методе за повезивање Андроид уређаја са *LEGO Mindstorms NXT* роботским системом са циљем да се омогући даљинско управљање употребом *Bluetooth* или *USB* конекције су описане у [44]. У раду [102] описано је решење за управљање робота употребом мобилног уређаја, *Bluetooth* комуникације и режима управљања базираног на мапи (*map based mode of controlling*). Уз овај режим рада робот проналази најкраће могуће растојање од стартне до циљне позиције уз избегавање препрека које је дефинисао корисник или које су се појавиле у току кретања. Предлог апликације за Андроид мобилни уређај, намењене за управљање мобилног робота у ручном режиму рада путем бежичне *Bluetooth* везе, дат је у раду [104]. Ручни режим рада дате апликације подразумева задавање жељеног правца и смера кретања у оквиру екрана осетљивог на додир, или гласовним командама.

У раду [146] описана је прва фаза пројекта имплементације управљачког система за управљање индустријске роботске руке *KUKA KR6* путем Андроид мобилног уређаја. Првенствена намена овог решења би била у медицини. Комуникација са роботском руком се не обавља бежично, већ путем *USB* серијске везе, док се програмирање врши употребом дугмади у оквиру екрана осетљивог на додир и акцелерометра или жироскопа.

Постојећа решења која користе *Bluetooth* протокол за комуникацију [44], [62], [102], [104], [138] могу да обезбеде комуникацију само на краћим растојањима, док комуникација путем *USB* везе [146] не може да обезбеди локацијску независност оператера.

У постојећим решењима код којих се интеракција са корисником базира на употреби паметних мобилних уређаја и бежичне комуникације (*Wi-Fi* или *Bluetooth*), програмирање кретања се најчешће врши уз помоћ акцелерометра или жироскопа [66], [111], попуњавањем текстуалних поља жељеним вредностима [4], [62], [137], [138] или бирањем правца и смера кретања кликом на одговарајућу дугмад [2], [4], [44], [58], [137], односно превлачењем прстом по екрану осетљивом на додир [104], [123].

Решења која користе *Wi-Fi (IEEE 802.11x)* протокол за програмирање, управљање и надзор робота и машина су описана у радовима [2], [4], [58], [66], [99], [111], [123], [137]. Оваква решења обављају надзор уз помоћ камере [99], [123], приказивањем вредности најзначајнијих параметара у текстуалном формату [111], или употребом 3Д анимације [58], [137], док се детектовање позиција робота обавља анализом слика добијених кроз камеру [99] или употребом сензора [111].

У раду [66] представљен је систем за управљање робота у коме се користе акцелерометар мобилног телефона и покрети руке како би се програмирало кретање робота. Систем користи два рачунара од којих први обавља задатке управљања робота, праћења објеката, препознавање команди и генерисање аудио-визуалних повратних информација, док други обавља задатке трансформације акција, односно покрета мобилног телефона које произведе корисник у скице кретања и пројектује их у окружење у коме робот треба да се креће. Мобилни уређај је са другим рачунаром повезан употребом *Wi-Fi* протокола, док је робот повезан са првим рачунаром употребом *Bluetooth* протокола комуникације.

У раду [99] предложен је систем за управљање и надзор робота базиран на Андроид ОС-у који за комуникацију користи *Wi-Fi* протокол, а за надзор кретања робота користи камеру. Сlike добијене са камере у реалном времену се испоручују паметном телефону кроз локалну мрежу. Величина једне слике која је добијена са камере износи 60-80 kB. Да би се смањила количина података која се преноси у току надзора извршена је конверзија YUV података у JPEG формат [99]. Овакво решење смањује величину података за пренос на ниво од око 2kB. Ипак, ефективнији би био пренос мање количине података, са обзором на то да се пренос обавља у кратким временским интервалима. Поред тога, овакав поступак би био и економичнији, поготово у случајевима када се користи интернет путем мобилног оператера и када корисник плаћа сваки пренесени бајт.

У раду [111] предложен је интерфејс за комуникацију између човека и машине базиран на употреби паметног телефона. Ово решење користи акцелерометар и жirosкоп паметног телефона са циљем да генерише

команде за управљање робота, док за комуникацију користи *Wi-Fi* протокол. Поруке које се преносе у току комуникације су ASCII стрингови. Надзор се спроводи приказивањем вредности података са сензора у реалном времену, приказивањем позиција робота, као и других значајних параметара у текстуалном формату. Оваква метода надзора није увек прикладна за стицање јасног увида у рад робота.

Још један пример система за даљинско управљање који је примењен на CNC машину је предложен у раду [137]. Овај систем се састоји од Андроид уређаја који ради као клијент и *Windows web* сервера са хибридном апликацијом која комбинује SOAP *web* сервисе за слање и примање контролних информација и комуникацију употребом сокета за периодично ажурирање података добијених са машине путем бежичне мреже. Даљински надзор се обавља употребом 3Д визуелизације како би се анимирале операције машине у реалном времену. Програмирање и управљање се спроводи попуњавањем текстуалних поља жељеним вредностима или слањем жељеног правца и смера кретања у оквиру екрана осетљивог на додир.

Опис апликације и комуникације за бежично управљање робота-рибе употребом паметног телефона је дат у раду [2]. У оквиру апликације омогућене су и показне анимације са циљем да се корисници, који се први пут сусрећу са једном оваквом апликацијом, упознају са њеним функционалностима. Управљање се врши употребом клизача на екрану мобилног уређаја. Комуникација између мобилног уређаја и робота-рибе путем *Wi-Fi* протокола је имплементирана употребом *Arduino Duemillanove* микроконтролера.

Систем за управљање хуманоидног робота Нао развијеном у *Aldebaran Robotics*, који користи паметни телефон, предложен је у раду [4]. У овом решењу се управљање говором робота спроводи уношењем текста у текстуална поља или снимањем говора, а окретање главе робота у жељеном смеру се спроводи кликом на дугмад у оквиру екрана осетљивог на додир мобилног уређаја. За надзор тренутног стања робота у овом решењу је предложена употреба икона.

У раду [58] је предложена употреба мобилног паметног телефона као преносног панела за обучавање у процесу управљања индустријских робота. Обучавање се врши у оквиру екрана осетљивог на додир мобилног уређаја, задавањем помераја у спољашњим или у унутрашњим координатама. Симулација се обавља на захтев који се добија са паметног телефона, а у оквиру апликације на *PC*-ју. Симулација се у току извршавања може прекинути и наставити слањем одговарајућих команди са паметног телефона на *PC*. Комуникација између паметног телефона и *PC*-ја се обавља уз употребу *Wi-Fi* протокола и сокета. Повратне информације о раду робота су обезбеђене у облику графичке 3Д анимације.

У раду [123] је описана имплементација интерфејса за управљање робота код кога се управљање врши путем екрана осетљивог на додир Андроид таблета. Управљање се врши тако што оператор додирује екран, а затим превлачењем по екрану задаје правац кретања робота. Надзор је омогућен употребом две камере, а комуникација између таблета и рачунара који је повезан са роботом се обавља бежично употребом *Wi-Fi* протокола.

4.3 Преглед постојећих софтверских решења за програмирање и управљање робота путем интернет претраживача

Један од првих успешних пројеката у оквиру кога су имплементиране функционалности управљања робота путем интернета, користећи интернет претраживач, је био *Mercury Project* [43]. Такав систем је био једноставан, лоше прецизности и имао је свега неколико функционалности, али је први који је испунио задатак даљинског управљања путем интернета.

Рад [129] описује систем за даљинско управљање и надзор мобилног робота имплементираног употребом *ARM7* микроконтролера. Комуникација је имплементирана употребом бежичних технологија и интеграцијом комуникационих и технологија за навигацију. За даљински надзор коришћена је *IP web* камера, а за чување снимљених видео снимака и слика

коришћен је *cloud* сервер - *Google Drive*. У оквиру овог решења корисник може да посматра позицију робота на *Google* мапи, а може се обављати и даљински надзор робота кроз интернет претраживач рачунара или мобилног телефона, приказивањем садржаја снимљеног *IP web* камером.

У раду [12] је описан алат за даљинско управљање мобилних робота код кога је кориснички интерфејс имплементиран као *HTML* страница са *Java Applet*-ом. Комуникација са робот контролером се обавља путем интернета, употребом *TCP/IP (Transmission Control Protocol/Internet Protocol)* протокола, кроз интернет претраживач. За надзор се користи виртуелно окружење.

У раду [105] су дати резултати испитивања перцепције објеката и робота у виртуелном и реалном индустријском окружењу. Приликом испитивања коришћено је виртуелно окружење коме се приступа употребом интернет претраживача *Internet Explorer* са *Cosmo player* додатком и у коме се комуникација са роботом обавља путем интернета.

У научним радовима описан је одређени број окружења за даљинско програмирање, управљање и надзор робота путем интернет претраживача који су намењени за учење и едукацију у роботизи [15], [30], [42], [46], [71], [130]. Описи ових решења су дати у наредном поглављу.

4.4 Преглед окружења за едукацију у роботизи

Постоји одређени број научно-истраживачких радова везаних за софтверске системе и окружења за едукацију у роботизи [7], [14], [15], [28], [29], [30], [42], [46], [71], [76], [90], [117], [130].

У раду [76] предложен је конструктивистички роботски симулатор базиран на игри (*G-IRSTS*). Овај симулатор развијен је са циљем да покаже ефективност употребе игре у едукацији и истражи ставове студената приликом употребе једног таквог окружења на часовима из предмета везаних за роботизу.

У раду [30] је предложен 3Д роботски симулатор за едукацију који као циљ има упознавање млађих генерација студената са роботиком и вештачком интелигенцијом. Ово окружење омогућава студентима да креирају и подешавају сопствено виртуелно окружење са могућношћу неограниченог приступа реалистичним роботским симулацијама.

Интерактивни алат за едукацију у роботици – *ReAct!* је представљен у раду [28]. *ReAct!* је алат који помаже студентима да науче главне концепте техника планирања употребом вештачке интелигенције, као и техника хибридног планирања.

Окружење за програмирање кретања робота базирано на *Python* језику за програмирање – *Pyro* је описано у раду [7]. Циљ окружења *Pyro* је да пружи јединствене концепте независне од специфичне роботске платформе и да их укључи у већ добро познате програмске парадигме.

Још један алат за едукацију у роботици – *Urban Search and Rescue Simulatuon (USARSim)* је предложен у раду [14]. То је мулти-роботски симулатор отвореног кода који се може покретати на *Windows, Linux* и *Mac* оперативним системима.

У раду [29] описан је систем за даљинско програмирање и управљање робота *SCARA*, као и даљинско планирање путање и извршавање програма. Систем је намењен студентима који желе да стекну практично искуство у области програмирања и управљања робота и да унапреде знање из те области. Систем користи софтверски алат *Leonardo* који се покреће на клијентском *PC*-ју и који путем *TCP/IP* протокола комуницира са сервером. *Leonardo* пакет обезбеђује све потребне функционалности за планирање путање, верификацију на виртуелном моделу робота, као и извршавање роботских програма.

Виртуелна лабораторија за роботiku (*Virtual Laboratory for Robotics - VLR*) која је развијена на Електротехничком факултету Универзитета у Београду је описана у раду [117]. *VLR* је намењена употреби на *PC*-ју, а укључује симулационе моделе којима се емулира динамика робота, *user-friendly* графички кориснички интерфејс и визуелизацију уз помоћ напредне и детаљне графике која укључује приказ свих унутрашњих механизма

робота. На тај начин студентима се пружа адекватан утисак рада као са правом опремом. *VLR* се може користити за експерименте и вежбе, али и за учење теорије робота. Приказ у овој мери детаљне графике која укључује све унутрашње механизме на екранима мобилних уређаја не би дао исти ефекат као приказ на екранима монитора рачунара већих димензија.

Рад [90] описује окружење базирано на *PC*-ју за програмирање и управљање робота са применом у едукацији у роботизи који је развијено у Лола институту. Детаљан опис овог окружења је дат у поглављу 3.2. Графички кориснички интерфејс са инструкцијама за програмирање кретања индустријских робота је приказан на слици 3.6.

Окружења којима се може приступити путем интернета и интернет претраживача за едукацију у роботизи су описана у радовима [15], [30], [42], [46], [71], [130].

Систем који укључује лабораторију са даљинским приступом - *Distance Laboratory System (DLS)* и који омогућава даљинско тестирање управљачких алгоритама робота је предложен у раду [15]. *DLS* помаже студентима да боље разумеју како дизајн контролера утиче на рад робота и помаже им да боље савладају кинематичке и динамичке моделе робота. Ово решење је намењено *Windows* оперативном систему, а комуникација се спроводи путем интернета. Корисници могу да приступе роботском систему кроз интернет претраживач (*Netscape Navigator* или *Microsoft Internet Explorer*).

ЗД роботски симулатор за едукацију који омогућава студентима да креирају и подешавају сопствено виртуелно окружење са могућношћу неограниченог приступа реалистичним роботским симулацијама [30] има могућност да користи *Macromedia Shockwave* додатак за претраживач са циљем да се приступ предложеној апликацији омогући кроз интернет.

Платформа за пројектовање и развој лабораторијских експеримената за роботске системе базирана на *web*-у је предложена у раду [71]. Ова платформа омогућава управљање, програмирање и надзор роботских станица у реалном времену за потребе експерименталних истраживања и омогућава студентима да обаве своја истраживања користећи стандардни интернет претраживач.

Решење за даљинско управљање у роботској индустрији и у едукацији - *Remote Rapid Control Prototyping (RRCP)* је предложено у раду [130]. Ово решење је базирано на употреби алата *MATLAB/Simulink*, *DSP-2* контролера и *LabVIEW* виртуелног окружења. Корисник може да извршава даљинско управљање и надзор кроз стандардни интернет претраживач.

У раду [46] је описано окружење које омогућава корисницима са различитих локација да интерагују (комуницирају, сарађују и координирају процесима) кроз дељено интерактивно 3Д интелигентно окружење. Корисници приступају окружењу путем интернет претраживача у оквиру кога се приказује интерактивни графички кориснички интерфејс. Интернет претраживач покреће клијентски софтвер као *Java Applet* у оквиру кога се уноси идентификационо име и лозинка. Сваки корисник је идентификован својим 3Д аватаром. Након пријављивања, корисник приступа виртуелном центру за обуку који је имплементиран као дељени простор. Приказ корисника у оквиру виртуелног окружења је дефинисан кроз његов аватар, а кориснику се садржај окружења приказује у првом лицу.

Алат под називом *LABEL* који је намењен за учење и едукацију у области паралелне роботике је описан у раду [42]. *LABEL* је намењен проучавању робота са паралелном кинематиком. Код овог алата интеракција са корисником је такође имплементирана кроз интернет претраживач.

5. ДЕФИНИЦИЈА ПРОБЛЕМА И ПРЕДЛОГ РЕШЕЊА

5.1 Дефиниција проблема

Роботски системи су сложени системи чији корисници морају бити добро обучени. Управљачки процес се најчешће спроводи у непосредној близини самог робота или употребом жичне конекције. Ресурси везани за роботске лабораторије нису често доступни корисницима које тек треба обучити за рад са оваквим системима, док рад са роботским системима често захтева детаљно познавање софтверских парадигми за програмирање робота. Због наведених разлога студенти који тек уче о основним принципима роботике углавном нису у могућности да буду упознати са програмирањем и управљањем комплексних роботских система и робота на практичан начин. Поред тога масивни и недоступни роботи све више захтевају програмирање и управљање независно од локације оператера.

У савременој роботизици постоји потреба за корисничким интерфејсима и уређајима који су прецизни и лаки за употребу, широко распрострањени, али и приступачни. Како роботика постаје све више софистициранија, а корисници уређаја за програмирање и управљање робота све мање обучени, постаје очигледан све већи значај развоја корисничких интерфејса и алата који су лаки и интуитивни за коришћење [66]. У складу са тим, приликом пројектовања графичког корисничког интерфејса за управљање робота није довољно направити фокус само на обезбеђивању детаљних и комплетних функционалности управљања, већ се лакоћа употребе такође треба узети у обзир [123]. У неким системима употреба робота представља само спорадичну функционалност и у таквим случајевима графички кориснички интерфејси за њихово управљање не могу увек да подразумевају екстензивну обуку и стручност оператера. Чак и када се подразумева експертиза, дизајн графичких корисничких интерфејса и даље треба да захтева што мање

когнитивног оптерећења оператера и обима посла, са циљем да се редукује стрес и умор оператера, али и могућност грешке, као и да се унапреди безбедност оператера [49].

Интуитивно програмирање индустријских робота подразумева интеракцију корисника са роботом на што лакши и природнији начин, без потребе познавања правила програмирања [53]. Један пример интерфејса софтвера који омогућава лако и интуитивно програмирање је описан у раду [63]. Интуитивно окружење за програмирање робота мора бити лако за употребу чак и када корисник није претходно упознат са његовим коришћењем, али у исто време мора да одржи комплексност и стабилност како би оператери били у могућности да креирају поуздане задатке. У идеалном интуитивном програмском језику састављање програма не би било потребно, већ би корисник могао да изрази шта жели на природан начин, а писање програма би се одвијало аутоматски [6]. Такав интерфејс би требао бити прилагођен навикама и потребама корисника, а не обрнуто.

Поред тога, у индустрији постоји потреба за решавањем проблема безбедности особља и потреба за обезбеђивањем бежичне опреме за рад и комуникацију [53]. Главна погодност коју доноси бежично програмирање и надзор су боља безбедност радника, омогућавајући им да програмирају робота у оквиру заштићеног и безбедног окружења. Остале предности овакве комуникације су слобода кретања корисника (елиминисањем употребе каблова за повезивање са робот контролером), али и могућност програмирања роботских задатака са неке друге локације, без физичке присутности.

Робот манипулатори се углавном користе за реализацију софистицираних задатака као што су брушење, обрада, брисање, и монтаже [79], али нису заступљени само у индустријским окружењима. Они се такође користе у нуклеарној индустрији [47], подводним истраживањима [120], аутомобилској [11] и авио индустрији [128]. Због све бржег развоја технологије, од великог је значаја укључити нова технолошка решења и у имплементацију окружења за обуку у авио индустрији [100], тако да будући авио инжењери буду у могућности да у будућности, у радном окружењу,

користе исте оне уређаје које користе данас у свакодневној употреби [135]. Према раду [3], да би се обезбедило да авијатичари (пилоти, навигатори, инжењери летачи и оператори на разним системима на авиону) у будућности могу да испуне радне задатке, мора да се промени систем њиховог образовања и тренинга (*"to ensure the future Airman are prepared to meet future challenges, the Air Force must transform how it educates and trains"*). Паметни телефони и други портабилни бежични уређаји могу дати могућност за стварање нових решења у приступу и дељењу знања међу авио инжењерима.

У едукацији инжењера, практичан рад у лабораторијама представља саму срж учења и има велики утицај на стечено знање студената [48]. Практична настава у лабораторијама се користи често и то са циљем да се обезбеди практичан увид у теоријске принципе и да се стекну практичне вештине. Међутим, практична настава у лабораторијама захтева посвећеност у времену, обезбеђивање простора за одвијање овакве наставе, као и обезбеђивање новчаних средстава за прикупљање и одржавање одговарајуће опреме. Поред тога, универзитети се данас суочавају за озбиљним ограничењима у погледу организације практичне наставе у роботским лабораторијама.

Подршка за учење на даљину (*Distance Learning*) и приступ материјалима за учење путем интернета су постали опште прихваћени на модерним универзитетским курсевима [106]. Даљински приступ роботским лабораторијама би могао да омогући бољу организацију лабораторијских активности и да унапреди квалитет учења [29]. Поред тога, овакав приступ би такође могао да обезбеди значајне предности у односу на постојеће методе практичне наставе, као што су приступачнија окружења за учење, повезивање теоријског са практичним радом или окружење за учење прилагођено појединцу. Имплементација и примена даљинских лабораторија у производном погону би донеле следеће предности: омогућавање студентима да приступе садржајима из специфичних области из којих није могуће обавити едукацију на самом универзитету, пружање прилике за рад са софистицираном опремом која је иначе превише скупа или неприступачна, унапређење радног искуства студената омогућавањем рада на окружењима

која већ постоје и на којима се већ ради у различитим компанијама, омогућавање компанијама да имају бољи увид о наставном програму на факултетима и самим тим и бољи увид у стечено као и потребно искуство студената, омогућавање студентима да имају бољи увид о њиховој будућој професионалној каријери, и на крају постојањем даљинских лабораторија у индустрији направила би се веза са академским институцијама и олакшао би се развој заједничких образовних и истраживачких програма [21]. Суштинска функционалност која даје предност лабораторијама са даљинским приступом је та да се не постављају никаква ограничења корисницима у погледу приступа већ јој се може приступити било где, било када.

Приступ са било које хардверске и софтверске платформе даје предност даљинским лабораторијама којима се може приступити путем интернет претраживача у односу на самосталне апликације које захтевају инсталацију, али и у одређеном броју случајева и додатни софтвер. Међутим, решење које би било могуће увек имати на свом персоналном уређају (као што је мобилни телефон) и које би увек било доступно (чак и у случају недоступности интернета) би могло чак и превазићи потребе за приступом преко различитих софтверских или хардверских платформи, а дало би и предност у могућности коришћења апликације и без сталног приступа интернету, где би се програм могао припремити *offline*, а када се приступ интернету омогући, покренути *online*. Таква функционалност код лабораторија са даљинским приступом путем интернет претраживача није обезбеђена. Решење које би било могуће увек имати на свом персоналном уређају превазилази поменуте недостатке тако што је једном инсталирана апликација доступна на корисниковом мобилном телефону, а самим тим и доступна било где и било када, и не захтева инсталацију на појединачним хардверским и софтверским платформама.

У области едукације у роботизи, бежична виртуелна окружења која дају реалну слику о раду робота би могла бити од великог значаја. У раду [117] забележене су следеће предности и унапређене функционалности виртуелних лабораторија у односу на “праве” лабораторије: могућност масивне реконфигурације, извршавање екстремних задатака, посматрање

унутрашњости робота, итд, и истакнуте су добро познате предности у вези са имплементацијом симулатора у процесу обуке: ниски трошкови, доступност, флексибилан распоред употребе, уштеда енергије, уштеда времена и за студенте и за особље, елиминисање могућности оштећења робота или окружења и настанка кварова, итд. Посебно би се издвојио значај примене оваквих лабораторија за едукацију у програмирању и управљању слабо доступних робота, као што су уређаји за адаптацију и тренинг пилота. Поред тога, примена виртуелних и лабораторија са даљинским приступом које су имплементирани за платформе и уређаје који су лако доступни, широко распрострањени и економични, као што су паметни телефони, би била јако исплатива.

Експанзија паметних телефона који обезбеђују интеракцију са корисником путем екрана осетљивог на додир, као и сталну интернет конекцију и велики број других сервиса, дају нове могућности за мобилно учење (*m-Learning*). Они пружају портабилност, ефикасност и могућност да се користе било где и било када, управо због ниске цене производње и могућности за сталном адаптацијом. Мобилно учење представља електронско учење са додатним функционалностима, у којима се мобилни уређаји и бежичне технологије користе за учење [13]. Мобилно учење може да унапреди електронско учење померајући границе изван затвореног простора и дајући ученицима могућност да унапреде своје вештине када год желе и где год желе. Због тога, мобилно учење, чије спровођење није везано за одређену локацију, представља следећу генерацију електронског учења [23]. Поред тога што би студенти овакве уређаје могли да користе за сакупљање и приказивање информација потребних за учење, они би оваква окружења могли и да персонализују у складу са својим потребама [143].

Истраживање о генералној употреби и знању студената о технологији као што су паметни телефони је спроведено у оквиру ове докторске дисертације. У истраживању је учествовало 64 ученика и студената. Двадесет и седам учесника су похађали електротехничку средњу школу “Никола Тесла”, смер електротехничар рачунара, а 37 учесника су били студенти Универзитета Сингидунум различитих смерова, укључујући електротехнику

и рачунарску технику, као и информатику и рачунарство. Просечан број година ученика средњих школа је био 18 година (од тога две особе женског пола и 25 особа мушког пола), а просечан број година студената је био 20.02 године (од тога 15 особа женског пола и 22 особе мушког пола). У истраживању се разматрала генерална употреба и знање студената о технологији као што су паметни телефони. Учесницима су постављена питања о томе да ли користе паметне мобилне телефоне, које паметне телефоне користе, да ли имају сталан приступ интернету и колико времена на дан проводе при коришћењу свог паметног телефона за забаву, али и за учење. Такође, учесницима су постављена следећа питања о организацији на предавањима и настави: да ли постоје курсеви који нуде окружења за мобилно учење и да ли су по њиховом мишљењу таква окружења корисна. Затим је од њих затражено да дају податак о томе колико често користе апликације на мобилним уређајима у односу на апликације на рачунарима као и у односу на апликације којима се приступа путем интернет претраживача. У оквиру истраживања учесници су имали могућност да бирају одговор од неколико понуђених код свих питања осим код питања о учестаности коришћења паметних телефона где су наводили време у часовима и минутима. Резултати истраживања су приказани у табелама 5.1-5.7.

Табела 5.1: Питање Q1: Да ли поседујеш паметан мобилни телефон?

Да ли поседујеш паметан мобилни телефон?	Ученици средњих школа	Студенти
Да	85,2%	94,6%
Не	14,8%	5,4%
Не знам	0%	0%

Табела 5.2: Питање Q2: Који оперативни систем има твој телефон?

Који оперативни систем има твој телефон?	Ученици средњих школа	Студенти
Андроид	85,2%	75,7%
iOS	3,7%	8,1%
Windows 8	0%	5,4%
Не знам	11,1%	10,8%

Табела 5.3: Питање Q3: Да ли имаш стални приступ интернету преко паметног телефона?

Да ли имаш стални приступ интернету преко паметног телефона?	Ученици средњих школа	Студенти
Да	70,4%	94,6%
Не	29,6%	5,4%
Не знам	0%	0%

Табела 5.4: Питање Q4: Да ли користиш телефон за учење?

Да ли користиш телефон за учење?	Ученици средњих школа	Студенти
Да	55,6%	70,3%
Не	44,4%	29,7%
Не знам	0%	0%

Табела 5.5: Питање Q5: Колико често (у сатима) користиш телефон за забаву и за учење?

	Ученици средњих школа	Студенти
Колико (у сатима) користиш телефон у току дана?	6,42 сати	7,15 сати
Колико (у сатима) користиш телефон за учење у току дана?	1,37 сати	2,08 сати

Табела 5.6: Питање Q6: Да ли похађаш курсеве који нуде мобилна окружења за учење?

Да ли похађаш курсеве који нуде мобилна окружења за учење?	Ученици средњих школа	Студенти
Да	15%	59,5%
Не	85%	37,8%
Не знам	0%	2,7%

Табела 5.7: Питање Q7: Да је корисно имати мобилна окружења за учење?

Да ли је корисно имати мобилна окружења за учење?	Ученици средњих школа	Студенти
Да	81,5%	81,1%
Не	18,5%	18,9%
Не знам	0%	0%

Одговори на питања Q1 (Табела 5.1) и Q2 (Табела 5.2) указују на широко распрострањену употребу паметних телефона међу студентима и ученицима средњих школа, али и то да студенти више користе овакве уређаје у односу на ученике средњих школа. Одговори на питање Q2 такође указују на то да су уређаји са Андроид оперативним системом најчешће коришћени уређаји међу студентима и ученицима средњих школа. Одговори

на питање Q3 (Табела 5.3) показују да скоро сваки студент и велики број ученика средњих школа има могућност сталног приступа интернету у оквиру њихових мобилних уређаја. Одговори на питања Q4 (Табела 5.4) и Q5 (Табела 5.5) указују на честу употребу паметних телефона за потребе учења, посебно међу студентима. Просечно време употребе паметних телефона за учење код студената износи 2,08 часова на дан, док за ученике средњих школа тај податак износи 1,37 часова на дан. Те вредности су једнаке износу од приближно једне петине, односно једне трећине укупног броја сати проведеног у коришћењу паметних телефона на дан за ученике средњих школа, односно за студенте, респективно. Одговори дати на питање Q6 (Табела 5.6) указују на то да ученици средњих школа похађају веома мали број курсева у оквиру својих предмета који им даје могућност коришћења окружења за мобилно учење, док студенти похађају одређени број таквих курсева. Одговори које су студенти и ученици средњих школа дали на питање Q7 (Табела 5.7) показују да већина њих из обе групе сматра да је могућност употребе окружења за мобилно учење јако корисна за њих.

Резултати истраживања о генералној употреби технологије паметних телефона међу ученицима и студентима, које је спроведено учешћем 27 ученика електротехничке средње школе и 37 студената технике потврђују да данашњи брз развој технологије захтева унапређење метода за учење и едукацију. Резултати истраживања такође показују да је потреба за једним оваквим решењем већа међу студентима, него међу ученицима средњих школа.

5.2 Класификација постојећих софтверских решења из области програмирања, управљања и надзора работа употребом мобилних уређаја

Класификација постојећих софтверских решења која користе мобилне телефоне и бежичну комуникацију за програмирање, управљање и даљински надзор работа дата је у табелама 5.8 (а) и 5.8 (б). Садржај је подељен у две табеле како би се постигла прегледност приликом анализирања већег броја постојећих решења. У оквиру класификације укључено је и решење предложено у оквиру ове докторске дисертације. Знаком “+” је означено испуњење функционалности за постојећа решења из литературе са одговарајућом референцом или за предложено решење, док је знаком “-” означено да решење не испуњава дату функционалност. Знак “/” означава да информација о томе да ли неко решење испуњава дату функционалност није позната из литературе. Знаком “*” је означена функционалност предложеног решења која је планирана за имплементацију у оквиру будућег рада. Најпре је извршена класификација постојећих решења према начину програмирања кретања робота и роботских задатака, са циљем формирања нове методе која је интуитивнија и једноставнија од постојећих, а што ће се показати резултатима истраживања кроз упитнике намењене ученицима који ће предложено решење користити у оквиру практичне наставе. Класификација према мобилности решења је разматрана са становишта врсте коришћеног протокола за бежичну комуникацију (*Wi-Fi, Bluetooth*) од којих зависи колика је могућа удаљеност оператера од робота. Затим је извршена класификација према врсти надзора и могућности симулације.

Очекује се да главне предности предложеног решења буду унапређење интеракције између корисника и роботског система увођењем режима рада брзог бирања (*speed dial*) и новог режима рада за програмирање робота који применом одговарајућих елемената графичког корисничког интерфејса

(дугмади, падајућих менија и текстуалних поља) елиминише потребу за детаљним познавањем парадигми роботског програмског језика, као и унапређење приступа роботским лабораторијама омогућавањем бежичне комуникације са роботским лабораторијама на већим растојањима (употребом *Wi-Fi* протокола) уз надзор базиран на 3Д визуелизацији. Укључен би био и рад у ручном режиму у оквиру кога се управљање робота одвија бирањем правца и смера кретања оса робота. У односу на већи број постојећих решења базираних на употреби мобилних уређаја, предложено решење би укључивало и симулатор за потребе тестирања роботских програма пре примене у роботској лабораторији. На тај начин би предложено решење превазишло недостатке и интегрисало најефективније функционалности постојећих софтверских решења у једну целину.

Табела 5.8 (а) Класификација постојећих софтверских решења која користе мобилне телефоне и бежичну комуникацију за програмирање, управљање и даљински надзор робота на основу протокола бежичне комуникације, начина програмирања робота, врсте надзора и могућности симулације (први део)

		[2]	[4]	[44]	[58]	[62]	[66]	[99]	[102]	[104]
Бежична комуникација	<i>Wi-Fi</i>	+	+	-	+	-	+	+	-	-
	<i>Bluetooth</i>	-	-	+	-	+	-	-	+	+
Програмирање	<i>Touch screen</i> (попуњавање текстуалних поља)	-	+	-	-	+	-	/	-	-
	<i>Touch screen</i> (бирање правца и смера кретања)	+	+	+	+	-	-	/	-	+ ¹
	<i>Touch screen</i> (режим рада брзог бирања)	-	-	-	-	-	-	/	-	-
	Акцелерометар или жироскоп	-	-	-	-	-	+	-	-	-
Надзор	Камера	-	-	-	-	-	-	+	-	-
	ЗД визуелизација	-	-	-	+	-	-	-	-	-
Симулација		-	-	-	+ ²	-	-	-	-	-

Табела 5.8 (б) Класификација постојећих софтверских решења која користе мобилне телефоне и бежичну комуникацију за програмирање, управљање и даљински надзор робота на основу протокола бежичне комуникације, начина програмирања робота, врсте надзора и могућности симулације (други део)

		[111]	[123]	[137]	[138]	[146]	Предложено решење
Бежична комуникација	<i>Wi-Fi</i>	+	+	+	-	-	+
	<i>Bluetooth</i>	-	-	-	+	-	-
Програмирање	<i>Touch screen</i> (попуњавање текстуалних поља)	-	-	+	+	-	+
	<i>Touch screen</i> (бирање правца и смера кретања)	-	+ ¹	+	-	+	+
	<i>Touch screen</i> (режим рада брзог бирања)	-	-	-	-	-	+
	Акцелерометар или жирокоп	+	-	-	-	+	*
Надзор	Камера	-	+	-	-	-	-
	ЗД визуелизација	-	-	+	-	-	+
Симулација		-	-	-	-	-	+

¹ правац кретања се задаје превлачењем прстом по екрану осетљивом на додир (*Touch screen*-у)

² на РС-ју

5.3 Класификација софтверских окружења за едукацију у роботници

Извршена је класификација постојећих софтверских решења за едукацију у роботници на основу следећих карактеристика:

- да ли је окружење пројектовано за употребу на *PC*-ју,
- да ли се окружењу може приступити путем интернет претраживача *PC*-ја или мобилних уређаја, или
- да ли је садржај окружења прилагођен и имплементиран у складу са функционалностима и карактеристикама паметних мобилних уређаја.

Класификација је дата у табели 5.9. У оквиру класификације укључено је и решење за едукацију у роботници које је предложено у оквиру ове докторске дисертације (*Robotics First*). Знаком “+” је означено коју од датих карактеристика поседује свако од постојећих решења из литературе са одговарајућом референцом, као и предложено решење. На основу дате табеле класификације може се видети које су очекиване предности предложеног решења у односу на постојећа решења. Главна предност предложеног решења би била прилагођење садржаја окружења за едукацију у роботници функционалностима и особинама мобилних уређаја и интеракцији базираној на употреби екрана осетљивог на додир, чиме би се допринело интуитивности и мобилности у области едукације у роботници .

Табела 5.9 Класификација постојећих софтверских решења за едукацију у
роботици

	Употреба на PC-ју	Пристап путем интернет претраживача	Садржај прилагођен мобилним платформама
[7]	+	-	-
[14]	+	-	-
[15]	+	+	-
[28]	+	-	-
[29]	+	-	-
[30]	+	+	-
[42]	+	+	-
[46]	+	+	-
[71]	+	+	-
[76]	+	-	-
[90]	+	-	-
[130]	+	+	-
<i>Robotics First</i>	-	-	+

5.4 Критичка анализа постојећих решења

Приликом директног визуелног надзора робота у лабораторији, окружење у коме се обавља извршавање роботских програма из више разлога може имати слабу видљивост [58]. У таквим случајевима оператор не би могао да буде сигуран да ли робот исправно извршава задати задатак и да ли су позиције робота коректне. Такође, посматрање извршавања роботских програма на овај начин није увек безбедно. У неким случајевима је и обавезно имати могућност програмирања, управљања и надзора са удаљене локације, као што је случај са уређајима за тренинг пилота и симулацију лета.

Описана решења која користе *Bluetooth* протокол за комуникацију [44], [62], [102], [104], [138] су оријентисана ка употреби уређаја на краћим растојањима, код којих је ова технологија коришћена као замена за жични

пренос података. *IEEE 802.11 (Wi-Fi)* технологија је коришћена као замена за жичну локалну мрежу или за комуникацију путем интернета [2], [4], [58], [66], [99], [111], [123], [137]. Ова технологија обезбеђује бржу конекцију, већу удаљеност од базне станице и бољу сигурност него *Bluetooth*. Метода за даљинско управљање и надзор робота која је предложена у овој докторској дисертацији захтева имплементацију комуникације на већи растојањима, у случајевима када робот није видљив и када се даљински надзор обавља кроз 3Д визуелизацију. Због наведених разлога, за даљинско програмирање, управљање и надзор предлаже се употреба *Wi-Fi* технологије.

Методe за добијање повратних информација код даљинског надзора рада робота се могу поделити на [107]:

- визуелне повратне информације,
- звучне повратне информације и
- повратне информације у виду повратних сила - *force feedback*.

Метода надзора уз помоћ визуелних повратних информација може да пружи знатно већу количину података и информација у односу на преостале две методе. Такође, интерпретирање нумеричких вредности како би се стекао увид у рад робота није једноставно. Даљински надзор који се обавља кроз 3Д виртуелне моделе робота или праћењем путање енд-ефектора дао би далеко природнији и јаснији увид у кретање робота у односу на праћење кретања робота на основу текстуалног приказа параметара [111].

Имплементација надзора кретања робота кроз виртуелни модел робота или праћењем путање енд-ефектора робота у 3Д координатном систему, који се обавља на основу тренутне позиције робота у лабораторији, захтевала би пренос знатно мање количине података у односу на решења која примењују надзор помоћу камере [99], [123]. Употреба бинарног кода за представу података који ће се преносити путем бежичне мреже, а који ће се користити у интерпретирању роботских програма, такође би захтевала пренос веома мале количине података.

Употреба симулатора у оквиру окружења за даљинско програмирање, управљање и надзор би донела могућност верификације и тестирања роботских задатака пре примене у роботској лабораторији. Од описаних

решења која користе мобилне уређаје за програмирање, управљање и надзор, једино решење предложено у раду [58] садржи могућност употребе симулатора, али је он имплементиран за *PC* и не представља интегрални део апликације која се покреће на мобилном уређају.

Редуковањем броја акција корисника потребних за програмирање и покретање роботског програма допринело би се једноставнијој и интуитивнијој интеракцији између човека и машине, односно робота, у односу да претходно описана решења [2], [4], [58], [66], [90], [99], [111], [123], [137], [146]. Дефинисање роботског програма приликом сваког извршавања је временски веома захтеван процес за подешавање жељеног кретања робота, што посебно долази до изражаја када се често користи исти или сличан програм за извршавање. Методе за програмирање и управљање робота уз помоћ акцелерометра или жirosкопа [66], [111], [146] нису довољно прецизне приликом специфицирања путање кретања индустријских робота. Програмирање и управљање робота бирањем правца и смера кретања робота кликом на одговарајуће дугме интерфејса онемогућава извршавање сложених кретања робота, али и код једноставних задатака захтева велики утрошак времена оператера, али и велику прецизност током дефинисања потребног правца и смера кретања. Због наведених разлога постоји потреба за једноставнијим и интуитивнијим интерфејсом за интеракцију са корисником (оператером) којим би се омогућило извршавање комплексних роботских задатака уз минимизирање когнитивног оптерећења оператера, али и уз очување прецизности у извршавању таквих задатака. Уз овакву функционалност знатно би се олакшало програмирање и покретање роботских задатака и редуковало би се време потребно за подешавање и генерисање жељеног роботског програма. Метода управљања која је предложена у раду [123] смањује когнитивно оптерећење оператера, међутим, ова метода није погодна за извршавање репетативних комплексних роботских програма, са обзиром на то да сваки пут захтева поновно специфицирање кретања. У циљу редуковања времена потребног за дефинисање и извршавање репетативних роботских задатака било би корисно имати могућност да се једном дефинисани задатак робота

меморише. Такође, метода предложена у раду [123] не омогућава довољно прецизно задавање путање кретања робота.

Анализирана решења за едукацију у роботизи [7], [14], [15], [28], [29], [30], [42], [46], [71], [76], [90], [130] су пројектована за употребу на *PC*-ју или им се приступа кроз интернет претраживач, и њихов садржај није прилагођен и имплементиран у складу са функционалностима и карактеристикама паметних мобилних уређаја. Поред тога, у неким решењима комплексне инструкције кретања робота које су базиране на употреби кода [90] (инструкције приказане на слици 3.6) захтевају дубоко познавање парадигми роботског програмског језика, али и репетитивно уношење података и команди, што није погодно за обучавање.

Са обзиром на велику распрострањеност употребе мобилних уређаја у данашње време, окружења за едукацију у роботизи би се требала прилагодити мобилним платформама са циљем да се учење у областима везаним за роботизу унапреди. Ово унапређење би се постигло обезбеђивањем једноставнијег и интуитивнијег интерфејса за интеракцију између човека и машине, обезбеђивањем могућности употребе окружења за едукацију било где и било када, али и употребом економичне опреме. Окружење намењено употреби на паметним телефонима које се предлаже у овој докторској дисертацији поседује дате функционалности.

Окружењима намењеним едукацији у роботизи, описаним у радовима [15], [30], [42], [46], [71], [130], се може приступити кроз интернет претраживач *PC*-ја, али и користећи интернет претраживач на таблет-у или неком другом мобилном уређају. Међутим, садржај интернет страница је у највећем броју случајева имплементиран тако да одговара *PC* рачунарима са великим екранима и високом резолуцијом. Такође, одређени број интернет претраживача намењених мобилним платформама не поседује одговарајуће софтверске додатке (*plug-in*), али и не поседује довољно слободне меморије за приказивање страница и графике намењених за извршавање на *PC*-ју [95]. Када се апликација намењена за *PC* користи кроз интернет претраживач мобилног уређаја, садржај апликације се компресује са циљем да се уклопи у величину екрана таблета или паметног телефона [25]. Ово чини приказ

садржаја апликације естетски неприхватљивим, са отежаном навигацијом, а у неким случајевима и потпуно нечитким приказом (*“aesthetically unpleasant, unnavigable, and in the worst case, completely illegible”*) [136]. Уколико је апликација превише компликована или недовољно интуитивна за коришћење корисник јој неће посветити много времена. Током имплементације окружења или апликације за мобилно учење потребно је посебну пажњу посветити особинама и ограничењима мобилних платформи, са циљем да се садржај који је потребан за учење прилагоди специфичном типу уређаја којем је намењен. Апликације и окружења за мобилно учење се морају прилагодити екранима мобилних уређаја и интеракцији базираној на екранима осетљивим на додир, али и морају бити пројектовани тако да буду интуитивни и лаки за употребу, са циљем да повећају мотивацију и жељу корисника за учењем. У радовима [4] и [37] су пројектантима мобилних апликација понуђене сугестије и принципи за пројектовање интерфејса апликација за мобилне уређаје којима би се постигло прилагођење садржаја окружења према особинама екрана мобилних уређаја и интеракцији базираној на екранима осетљивим на додир. У раду [148] описане су функционалности које је потребно остварити како би се постигла успешна телеманипулација – даљинско вођење робота [114]. Ови радови ће детаљније бити анализирани у поглављу 5.6.1.

5.5 Полазне хипотезе

Основне претпоставке су да је могуће унапредити и олакшати процес програмирања, управљања и надзора робота у индустријским условима, али и у образовном процесу:

- Интегрисањем најефективнијих појединачних функционалности постојећих софтверских решења из области програмирања, управљања и надзора робота која се користе у индустријским условима и у едукацији, са посебним освртом на решењима која користе паметне телефоне и мобилне платформе.

- Превазилажењем уочених недостатака постојећих софтверских решења базираних на *PC*-ју као и решења којима се може приступити путем интернет претраживача мобилних уређаја уз поштовање услова за прилагођење функционалности система мобилним платформама и паметним телефонима.
- Обезбеђивањем једноставности и интуитивности у интеракцији, мобилности и економичности.

Тиме би се значајно унапредио и олакшао процес програмирања, управљања и надзора робота. Предложено решење би дало могућност млађој популацији, а будућим инжењерима у области аеро и роботске индустрије, да користе исте уређаје у радном окружењу, као што их користе данас у свакодневној употреби. Применом оваквог решења у едукацији поправио би се квалитет преноса знања, тако што би се добила могућност да се приступи иначе недоступним роботским системима, кроз интуитивне виртуелне лабораторије и независно од локације корисника. Употребом широко распрострањених уређаја ниске цене процес опремања студената одговарајућом опремом би се учинио економичним.

5.6 Предлог нове методе за даљинско програмирање, управљање и надзор робота

Формирање нове методе за даљинско програмирање, управљање и надзор робота је извршено на основу анализа постојећих решења чији су преглед и класификација дати у овој докторској дисертацији, са циљем унапређења интеракције између корисника и роботског система приликом програмирања и управљања робота, као и унапређења приступа роботским лабораторијама. Нова метода за даљинско програмирање, управљање и надзор робота се базира на употреби паметног мобилног уређаја и бежичне комуникације на већим растојањима, уз виртуелно окружење за даљински надзор и симулацију.

Разлике у односу на решења из постојеће литературе и унапређења које даје предложено решење су дати у табелама 5.8 а), 5.8 б) и 5.9. У односу на претходно описана и класификована постојећа решења, предложена метода представља интегрисано решење које обухвата:

- Могућност употребе било где и било када употребом бежичне комуникације са роботским системом у оквиру локалне рачунарске мреже или путем интернета,
- Даљински надзор рада робота кроз 3Д визуелизацију који не захтева пренос велике количине података,
- Интуитивно програмирање без потребе детаљног познавања парадигми језика за програмирање робота,
- Значајно редуковање броја акција корисника током програмирања и управљања робота уз очување прецизности при извршавању сложених роботских програма и
- Симулатор за тестирање и верификацију роботских програма.

Имплементацијом бежичне комуникације са роботом у оквиру локалне рачунарске мреже или путем интернета на мобилном уређају, обезбеђује се локацијска независност корисника и повећава се њихова безбедност. Поред тога, омогућава се надзор из више перспектива истовремено, посматрањем кретања робота уживо у лабораторији и истовременим праћењем његовог кретања у оквиру 3Д приказа на мобилном уређају са различитих позиција. Овакав начин праћења извршавања роботских задатака може да пружи бољи увид у извршавање роботских програма, алгоритама, као и сам рад робота. Применом предложеног решења у областима у којима се користе уређаји за тренинг пилота и симулацију лета обезбеђује се могућност да се ови слабо доступни уређаји програмирају са удаљене локације и на тај начин омогућава се операторима са slabим искуством у програмирању и управљању оваквих уређаја, да се на флексибилан начин упознају са њиховим радом.

Имплементацијом скупа кључних команди за извршавање сложених роботских програма са само једном акцијом у оквиру екрана мобилног уређаја обезбеђује се значајно редуковање броја акција корисника потребних

за извршавање и симулацију роботских програма. Имплементацијом окружења за програмирање кретања робота којим се уводи употреба одређених елемената графичког корисничког интерфејса екрана осетљивог на додир (дугмад, падајући мениј и текстуална поља) приликом дефинисања кретања робота и тиме елиминише потреба употребе програмског језика и детаљно познавање његове синтаксе и семантике, доприноси се редуковању броја акција корисника при програмирању роботских задатака, као и једноставности и интуитивности употребе предложеног решења. У односу на традиционалне и стандардне методе програмирања робота програмским језиком, предложена метода обезбеђује значајно редуковање корака потребних за писање и припрему програма за кретање робота и за њихово извршавање. На тај начин се значајно поједноставља програмирање и постиже се уштеда времена потребног за обуку оператера за програмирање кретања робота комплексним роботским инструкцијама. Очување прецизности се обезбеђује интеграцијом алгоритама за кретање робота [72], [73], [74] у оквиру окружења на мобилној платформи и њиховим повезивањем са елементима графичког корисничког интерфејса за програмирање и управљање робота. У раду [82] су дати резултати процене интуитивности четири елемента графичког корисничког интерфејса таблета (дугмад, џојстик, превлачење прстом по екрану осетљивом на додир и нагињање, односно окретање таблета) при управљању мобилним манипулатором. Према датим подацима најуспешнији резултати при управљању мобилним манипулатором су добијени када су корисници користили дугмад, док су најлошији резултати добијени када су корисници окретањем, односно нагињањем таблета, управљали кретањем робота. Имајући то у виду изабрано је да се графички кориснички интерфејс предложеног решења заснива на употреби дугмади, уз додатак падајућих менија и текстуалних поља за избор, односно унос, вредности жељених параметара.

Употребом симулатора у оквиру предложеног окружења омогућава се дефинисање сложених кретања робота и верификацију комплексних роботских програма пре примене на даљинско програмирање и управљање

робота који се налазе у роботским лабораторијама. Такође, ова функционалност је од значаја и за потребе примене у оквиру различитих презентација рада робота у едукацији у роботици.

У складу са описаним функционалностима предлаже се да окружење садржи пет различитих режима рада: режим рада за програмирање, режим рада брзог бирања, ручни режим рада, режим рада за надзор и режим рада за тестирање. Режим рада за програмирање служи за дефинисање роботских програма и њихово повезивање са акцијама и елементима у оквиру графичког корисничког интерфејса режима рада брзог бирања. Режим рада за програмирање је пројектован тако да не захтева детаљно познавање синтаксе и семантике језика за програмирање робота, већ омогућава бирање жељених инструкција из понуђеног скупа већ дефинисаних инструкција и унос жељених вредности потребних параметара који описују кретање робота у одговарајућа текстуална поља. Навођење назива параметара и променљивих, као и оператора доделе вредности, од стране оператера није потребно. Режим рада брзог бирања укључује могућност извршавања команди које дефинишу претходно програмиране сложене роботске програме са само једном акцијом (један клик на дугме) у оквиру екрана осетљивог на додир мобилног уређаја, уз очување прецизности у извршавању. Ручни режим рада омогућава ручно вођење оса робота са дефинисаним коракком. Режим рада за надзор омогућава даљински надзор кретања робота кроз 3Д окружење и може се користити у комбинацији са режимом рада брзог бирања или ручним режимом рада. Подаци потребни за извршавање роботских задатака, који се преносе путем интернета или у локалној мрежи, се представљају у бинарном формату. Надзор се спроводи на основу позиција робота у унутрашњим координатама, које представљају повратне информације. Оваква имплементација елиминише потребу за преносом велике количине података током управљања и надзора. Режим рада за тестирање укључује употребу симулатора за тестирање и верификацију сложених роботских програма пре примене на даљинско програмирање и управљање робота у роботским лабораторијама.

Применом предложене методе доприноси се интуитивности, локацијској независности и економичности у процесу програмирања, управљања и надзора рада робота у индустријским условима, али и у едукацији. Применом предложеног режима рада брзог бирања и режима рада за програмирање унапређује се интеракција између корисника и роботског система приликом програмирања и управљања рада робота у односу на окружења код којих се програмирање кретања робота обавља употребом програмског језика [90]. Применом предложеног виртуелног окружења и бежичне комуникације независне од локације корисника унапређује се приступ роботским лабораторијама. Употребом паметних мобилних телефона или таблета за интеракцију са корисником превазилазе се недостаци постојећих софтверских решења базираних на *PC*-ју за интеракцију са корисником [9], [20], [59], [61], [67], [115], [122], [131], [142], [144]. Економичност се остварује употребом ових широко распрострањених уређаја ниске цене приступачних широј популацији, а посебно студентима и ученицима који у великом броју већ поседују овакве уређаје. Метода за превазилажење недостатака постојећих софтверских решења којима се може приступити путем интернет претраживача мобилних уређаја [12], [15], [30], [42], [43], [46], [71], [105], [129], [130], којом се извршава прилагођење садржаја окружења за даљинско програмирање, управљање и надзор робота према функционалностима и карактеристикама мобилних уређаја и интеракцији која се базира на употреби екрана осетљивог на додир, је описана у следећем поглављу. Да предложена метода обезбеђује интуитивност биће показано у поглављу 10, у коме је описано квалитативно истраживање са аспекта ефикасности и интуитивности употребе предложеног решења. У истраживању је учествовало 24 ученика електротехничке средње школе “Стари град” из Београда.

5.6.1 Прилагођење садржаја окружења за програмирање, управљање и надзор работа према карактеристикама и функционалностима мобилних уређаја

Због потребе да се превазиђу недостаци постојећих софтверских решења којима се може приступити путем интернет претраживача мобилних уређаја [12], [15], [30], [42], [43], [46], [71], [105], [129], [130], садржај окружења за даљинско програмирање, управљање и надзор работа које је намењено мобилним платформама мора бити прилагођен функционалностима и карактеристикама мобилних уређаја и интеракцији која се базира на употреби екрана осетљивог на додир.

У раду [37] су понуђене следеће сугестије пројектантима мобилних апликација за учење (*m-learning* апликација):

1. Број акција на тастатури потребних да се изврши неки догађај треба ограничити,
2. Садржај у текстуалном облику треба ограничити и
3. Треба укључити само онај садржај који је потребан или који је најбитнији за учење.

Како би предложено решење било намењено за употребу у индустријским условима, али и у едукацији у роботици, током пројектовања тежило се да се испуне дате сугестије. Омогућавањем извршавања роботског програма са само једним кликом у оквиру екрана мобилног уређаја максимално се ограничава број акција корисника потребних да се изврши неки догађај и на тај начин се испуњава прва сугестија дата у раду [37]. Овај услов се такође задовољава применом предложеног режима рада за програмирање, где се у потпуности изоставља потреба за дефинисањем назива параметара и променљивих, као и потреба за специфицирањем оператора доделе вредности од стране оператера. Оваква функционалност такође задовољава и другу сугестију за имплементацију мобилних апликација за учење описану у раду [37] којом се сугерише ограничавање

количине текстуалног садржаја. Како виртуелизација представља један од најважнијих алата за учење, поготово у области роботике због тога што даје могућност да се добије јасан увид у извршавање роботских процеса, већи део функционалности предложеног решења је базиран на 3Д визуелизацији. Такав облик приказа информација најпогоднији је за примену у едукацији у роботици, па оваква имплементација испуњава и трећу сугестију која је предложена у раду [37].

У раду [148] је наведено да је могућност ограничене обуке оператера једна од битних функционалности за успешну телеманипулацију. Роботи се често употребљавају за веома специјализоване задатке као што су дифузија бомби или истраживање нивоа мора. Управљање таквих робота захтева огромну стручност, искуство и вештину. Такви роботи би могли да се користе у много већем броју области када би поседовали једноставнији кориснички интерфејс чиме би обука била сведена на минимум и самим тим већи број оператера би се за краће време оспособио за рад са оваквим системима. Обезбеђивањем могућности програмирања кретања робота кроз предложени режим рада за програмирање елиминише се потреба за дубоким познавањем парадигми језика за програмирање робота, као и потреба за познавањем синтаксе и семантике језика. Тиме се постиже једноставност графичког корисничког интерфејса за програмирање и испуњавају се дати услови потребни за спровођење ограничене обуке [148].

У раду [4] је одређено неколико принципа у пројектовању *user-friendly* корисничког интерфејса за програмирање робота:

1. Важне информације би требало да заузимају велики део интерфејса и да се налазе у центру,
2. За пренос информација би требало користити графику уместо текста са циљем да се помогне корисницима да боље разумеју тренутно стање робота на интуитиван начин и
3. На главном екрану би требало приказивати само неопходне информације; превише информација би само збуњивало и ометало кориснике.

Виртуализација, као један од најважнијих алата за спровођење даљинског надзора у области роботике, у оквиру предложеног графичког корисничког интерфејса заузима већи део. Уз то, коришћење ове функционалности може да се комбинује са различитим режимима рада, али може да се обавља и независно. На овај начин се задовољавају први и други принцип дати у раду [4]. Трећи принцип је испуњен предложеним режимом рада за програмирање којим се ограничава количина информација која се приказује на екрану, као и количина информација коју је потребно да корисник унесе, приказивањем и захтевањем само неопходних кључних речи и вредности параметара од значаја.

Развојем окружења за програмирање, управљање и надзор робота наменски за мобилну платформу добија се могућност његове употребе као самосталне *offline* апликације. На тај начин се једном сачувана и инсталирана апликација може поново покретати и радити у режиму рада за тестирање, односно симулацију и за њен рад није потребно имати приступ интернету. Оваква функционалност може бити од користи у случајевима када не постоји могућност да се користи интернет (*Wi-Fi* или мобилни интернет). Након спровођења тестирања и верификације роботских програма у режиму рада за тестирање, а након што се добије могућност приступа интернету, корисник може да пређе у неки од режима рада који подразумевају даљинску комуникацију са роботском лабораторијом. Окружење које је развијено наменски за мобилну платформу има могућност да приступа подацима са сензора мобилног уређаја, као што су жироскоп или *GPS* [81]. Апликације које за интеракцију са корисником користе интернет претраживач мобилних уређаја немају ову могућност. Ове функционалности могу бити од значаја у даљем развоју окружења за едукацију у роботизици - *Robotics First*, са циљем да се функционалности окружења прошире омогућавањем даљинског програмирања, управљања и симулације роботских задатака на основу покрета мобилног уређаја који корисник држи у руци и комбиновањем таквих команди (покрета) са режимом рада за програмирање и режимом рада брзог бирања, тако да се на тај начин поступак програмирања и управљања робота додатно унапреди.

5.7 Предлог методе за едукацију у роботици

Са циљем мотивације ученика и студената да уз примену популарних и широко распрострањених уређаја ниске цене науче основне концепте роботике, где је доступност ресурса ограничена и у већини случајева је потребно темељно познавање парадигми роботских програмских језика, предлаже се примена описане методе и окружења за даљинско програмирање, управљање и надзор робота у едукацији, у роботичи. Решење је названо *Robotics First* [84]. Циљ овог приступа је заинтересовати средњошколце за програмирање и управљање робота и едуковати студенте са одговарајућим предзнањем из ове области о томе како да програмирају и управљају комплексне роботске системе употребом паметних мобилних телефона или таблета, интуитивног програмирања и даљинског приступа роботској лабораторији.

У научној литератури [64] едукатори из области роботике су предложили стратегије које би биле од значаја за едукацију студената на курсевима из роботике, које укључују извођење вежби у оквиру лабораторија као додатак уз теоријску наставу, а са циљем да се надомести недостатак практичног искуства. Такође, досадашња истраживања су показала да студенти имају већу мотивацију да уче уколико се суоче са реалним сценаријима и ситуацијама из стварног живота у оквиру самог курса [54]. Такве функционалности би се могле омогућити употребом виртуелних лабораторија, али и лабораторија са даљинским приступом, тамо где је то могуће. У данашње време виртуелне и/или лабораторије са даљинским приступом су постале обавезан део модерних и успешних курсева у едукацији. Виртуелне лабораторије су засноване на употреби софтвера којима се симулира извршавање експеримената, док се у лабораторијама са даљинским приступом експерименти изводе даљински, путем интернета, уз употребу правих робота и машина [23]. Једна од најважнијих функционалности, као и предност виртуелних лабораторија у односу на

лабораторије са даљинским приступом, је омогућавање студентима да науче и из својих грешака и неуспеха, без изазивања икакве штете или хаварије. Ипак, лабораторије са даљинским приступом имају предности у односу на виртуелне лабораторије: оне дају информације о резултатима извршавања роботских програма на правим роботским системима и могу да пруже реалистичнији увид у извршавање таквих процеса [54]. Због наведених карактеристика виртуелних и лабораторија са даљинским приступом, од значаја би било имати могућност употребе обе врсте лабораторија које би се користиле у одговарајућим нивоима учења у оквиру истог виртуелног окружења. У складу са тим предлаже се метода за едукацију у роботици која се састоји од два нивоа:

1. Основни ниво, којим би се студенти уводили у области програмирања и управљања робота употребом симулација роботских апликација и виртуелних лабораторија.
2. Напредни ниво, креиран са циљем да студенти са одређеним предзнањем из области математике и програмирања науче како да програмирају сложене роботске програме и алгоритме, извршавају тестирања различитих управљачких стратегија, као и да примењују ново-развијене програме и алгоритме кретања у роботским лабораторијама употребом даљинског приступа.

Основни и напредни ниво учења се лако могу прилагодити нивоу курса и знању студената, почевши од најједноставнијих окружења која укључују једног робота, па до комплексних окружења са више робота. Окружење за програмирање и даљинско извршавање роботских програма не захтева детаљно познавање синтаксе и семантике роботског језика за програмирање, већ се суштина окружења базира на разумевању извршавања роботских програма и алгоритама. Недостатак опреме и прилика да се приступи роботским лабораторијама се превазилазе омогућавањем виртуелног присуства, независно од локације ученика. Бежична комуникација између мобилног уређаја и робота даје могућност посматрања рада робота из више перспектива истовремено: уживо у лабораторији и истовремено на мобилном уређају. Симулација употребом 3Д визуелизације

омогућава стицање искуства у раду са роботским системима без непотребног ризика од хаварија у роботским лабораторијама.

Роботи специјалне намене, као што су уређаји за тренинг пилота и симулацију лета, су масивни и комплексни уређаји и управљати таквим уређајима није једноставно ни безбедно за неискусне кориснике, а нарочито за кориснике који тек треба да се едукују о раду оваквих уређаја. Међутим, такви уређаји често нису доступни за потребе едукације. Употребом предложеног окружења *Robotics First* које би омогућило симулацију и даљински приступ овим уређајима, ученицима и студентима би се дала прилика да науче како да програмирају њихова кретања на безбедан начин. Тиме би им се омогућило стицање искуства у тој области, али би се они и мотивисали да се упознају и да даље наставе рад са таквим уређајима.

Употребом интуитивног графичког корисничког интерфејса за програмирање и управљање роботима, уз бежичну, даљинску комуникацију са роботским лабораторијама и могућношћу употребе симулатора, предложено решење би значајно олакшало ученицима и студентима учење у области роботике, а неприступачне уређаје и роботе би учинило далеко приступачнијим. Применом предложеног окружења за програмирање робота, које се базира на разумевању извршавања роботских програма и алгоритама и које не захтева детаљно познавање парадигми роботских програмских језика, редуковало би време потребно за савладавање градива из области управљања робота. Стицањем искустава у управљању робота употребом виртуелне лабораторије и симулација редуковала би се потреба за асистенцијом професионалаца. Анализа индустријских процеса и рада робота из више перспектива, која је омогућена у оквиру предложеног окружења, се предлаже са циљем да се омогући бољи увид у сам рад роботских система.

6. АНАЛИЗА ТЕХНОЛОГИЈА ЗА РАЗВОЈ СОФТВЕРСКОГ СИСТЕМА

Поставља се питање зашто развијати апликације за мобилне телефоне и уређаје? [94] Појава модерних мобилних паметних телефона — мулти-функционалних уређаја који укључују функционалности и телефона али и функционалности интернет претраживача, камера, бежичне комуникације, као и сервиса за одређивање локације, је фундаментално променила начин на који људи интерагују са својим мобилним телефонима и приступају интернету. Употреба мобилних телефона надмашује употребу рачунара у многим земљама, са више од три милијарде корисника широм света. Година 2009. представља годину у којој је први пут забележено да је већи број људи приступио интернету путем мобилног телефона него путем *PC*-ја. Све већа популарност модерних паметних телефона у комбинацији са бежичним преносом података ствара могућност за имплементацију напредних мобилних апликација. Свеprisутност паметних мобилних телефона, као и свакодневна повезаност са њима, чини их потпуно другачијом платформом за развој апликација у односу на *PC*. Са сензорима као што су микрофон, камера, екран који омогућава интеракцију додиром и детекција локације, ови уређаји представљају уређаје са додатном перцепцијом.

6.1 Мобилне платформе

Осамдесетпет процената свих мобилних телефона у свету су паметни телефони и предвиђа се да ће до 2017. године 70,5 процената од укупног броја свих уређаја који имају приступ интернету чинити управо паметни телефони [41]. Број уређаја који поседују Андроид оперативни систем из године у годину све више расте. Више од 850 000 нових Андроид уређаја се активира свакога дана, а најновије студије показују да највећи број продатих нових паметних телефона чине Андроид уређаји. На крају 2015. године број

продатих Андроид паметних телефона је износио 80,7 процената од продаје свих паметних телефона, док је друго место заузео *iOS* са 17,7 процената. Свој максимум је продаја Андроид мобилних уређаја достигла у трећем кварталу 2015. године и износила је 84,7 процената [133].

За развој Андроид апликација користи се једноставан, моћан алат на бази отвореног кода, који може да се користи без плаћања лиценце и који садржи обимну документацију [94]. Како би се добио приступ за *iOS SDK (Software Development Kit)* потребно би било да се плати годишња чланарина и право да се објаве апликације. Такође, *iOS SDK* је доступан само за *Mac* платформу. Андроид представља платформу отвореног кода, што значи да свако може бесплатно да приступи Андроид оперативном систему и бесплатно користи Андроид *SDK*. Платформе као што су *Windows Phone* и *iPhone* такође пружају функционална и једноставна развојна окружења за мобилне апликације. Ипак, за разлику од Андроид платформе, у неким случајевима они дају предност изворним (*native*) апликацијама у односу на апликације које су имплементирание као *third parties* апликације (апликације чији аутори нису произвођачи оперативних система, већ појединци или друге компаније), ограничавајући комуникацију између апликација и употребу уграђених функционалности телефона, и ограничавајући или контролишући дистрибуцију *third parties* апликација на својим платформама [94]. Код Андроид платформе, све апликације се третирају на исти начин. *Third parties* апликације и изворне апликације су написане уз употребу истог АПИ-ја (Апликациони Програмски Интерфејс). Корисници могу да уклоне или да замене било коју изворну апликацију, укључујући чак и *Dialer* и *Home Screen*.

Андроид ОС није имплементиран само за једну хардверску платформу, већ је дизајниран тако да подржи широк дијапазон хардверских платформи од паметних телефона, преко таблета, до телевизора. Приступ хардверу је омогућен за све апликације кроз скуп АПИ библиотека. Интеракција између апликација је у потпуности подржана, иако је у потпуности контролисана. Бесплатан и отворен приступ развоју мобилних апликација са пуним

приступом свим ресурсима уређаја представља највећу предност Андроид ОС.

6.2 Андроид ОС

Први Андроид мобилни уређај који је пуштен у продају је *T-Mobile G1* у САД, у октобру 2008. године. Само до почетка 2012. године продато је више од 300 милиона Андроид уређаја од 39 различитих произвођача, у више од 123 земље.

Андроид је софтверски стек за мобилне уређаје који укључује оперативни систем, средњи слој (*middleware*) и кључне апликације [24]. Андроид користи *Linux* ОС за основне системске сервисе: управљање меморијом, управљање процесима, управљање потрошњом, управљање хардвером и сигурност. Андроид ОС се ослања на *Linux* кернел, али на пример, стандардне *Linux* функционалности као што су *X-windows* и *GNU C* библиотеке нису подржане [126]. Код писања апликација за Андроид користи се Јава програмски језик, али у суштини то није Јава. Стандардне Јава библиотеке, као на пример *Swing*, нису подржане. Употреба неких од библиотека, као што је *Timer*, се не преферирају већ су замењене са другим библиотекама, које су оптимизоване за употребу у Андроид окружењу са ограниченим ресурсима. Андроид ОС је ОС отвореног кода, што значи да програмери могу да виде и користе било који део системског изворног кода. Јава код који је написан за Андроид ОС се уместо у *Java byte code (.class)* преводи у *Dalvik byte code (.dex)* [24]. На тај начин се извршава оптимизација за мобилне уређаје (боље управљање меморијом, коришћење батерије, итд). Уместо Јава виртуелне машине, код Андроид ОС-а се користи *Dalvik* виртуелна машина која извршава *.dex* датотеке.

Окружење за развој апликација за Андроид ОС укључује скуп емулятора виртуелних мобилних уређаја. На овај начин је омогућен развој апликација на рачунару без коришћења правих уређаја. *ADB (Android Debug Bridge)* је клијент/сервер програм који омогућава повезивање

клијентске/развојне машине са уређајем/емулатором како би се олакшао развој.

Постоје четири основне компоненте унутар Андроид апликација [24]:

- Активности (*Activity*)
- Сервиси (*Service*)
- Дохватање садржаја (*Content Provider*)
- Пријем обавештења (*Broadcast Receiver*)

Апликације се развијају користећи једну или више ових компоненти.

Активност је једна од основних компоненти апликације, која обезбеђује радну површ преко које корисник може да интерагује. Свакој активности се додељује прозор у коме се исцртава кориснички интерфејс. Активности која се покреће могу се послати аргументи користећи објекат намере (*Intent*). Намере представљају објекте за комуникацију између активности (иницирање и добијање одговора), између активности и сервиса и између активности и објекта који пружа садржај.

Сервиси су компоненте апликација које могу да обављају дуготрајне операције у позадини без коришћења корисничког интерфејса.

Компоненте за дохватање садржаја - *Content Providers* су објекти који пружају приступ до структурираних података. Ови објекти имају за циљ да повежу податке који постоје у једном процесу са програмом који се извршава у другом процесу.

Компонента *Broadcast Receiver* је задужена за пријем обавештења која су намењена целом систему, не обавезно унутар једне апликације.

Сваки Андроид пројекат мора да садржи своју манифест датотеку, *AndroidManifest.xml*. Ова датотека дефинише структуру и мета-податке апликације, њене компоненте и потребне дозволе [94]. Андроид подржава екстернализацију ресурса, почевши од једноставних вредности као што су стрингови или боје па све до комплексних као што су слике (*Drawables*), анимације, теме, и мениј. Најчешће коришћени ресурси су распореди компонената (*layout*). Употреба распореда компонената за описивање графичког корисничког интерфејса у *XML* формату је најчешћа пракса у Андроид апликацијама. Дефиниција сваког распореда компонената се

смешта у посебну *XML* датотеку у *res/layout* директоријуму. Само име датотеке након тога постаје идентификатор ресурса, а то се повезује уз помоћ манифест датотеке.

Приступ ресурсима у коду се обавља кроз статичку *R* класу. *R* је класа која се генерише у зависности од спољних ресурса и креира се при превођењу пројекта. *R* класа садржи статичке под-класе за сваки од типова ресурса за који је дефинисан најмање један ресурс.

У оквиру програмирања графичких корисничких интерфејса, Андроид уводи нову терминологију:

- Компонента *View* је објекат који исцртава нешто по екрану и са којим корисник може да интерагује,
- Компонента *ViewGroup* је објекат који може да садржи већи број компонената за исцртавање (*View* и *ViewGroup*) како би дефинисао распоред на корисничком интерфејсу.

Декларација распореда компонената се може обавити на два различита начина [24]:

- Декларација корисничког интерфејса користећи *XML*;
Као помоћ за развој корисничког интерфејса Андроид пружа праволинијско пресликавања атрибута у *XML* опису у одговарајуће *View* класе и поткласе, као што су оне за рад са распоредима компонената.
- Декларација корисничког интерфејса у време извршавања;
Поред статичке декларације корисничког интерфејса, могуће га је декларисати и у време извршавања постављањем компонената корисничког интерфејса програмским путем.

6.3 Технологије за бежичну и интернет комуникацију

6.3.1 *Bluetooth (IEEE 802.15.1)*

Bluetooth (IEEE 802.15.1) дефинише физички слој и MAC (*Medium Access Control*) подслој за бежичну комуникацију и најчешће се користи за повезивање и пренос података између уређаја на краћем растојању. *Bluetooth* технологија је од значаја када се преносе информације између два или више уређаја који су близу један другом, у случају када брзина преносе није од велике важности, као што је случај са телефонима, штампачима или слушалицама (на растојањима од 10 cm до 10 m) [36].

Првобитна идеја за развој *Bluetooth* технологије је конципирана 1994. године када је компанија *Ericsson Mobile Communications* почела да проучава системе са ниском потрошњом енергије који могу заменити каблове при повезивању мобилних уређаја и додатне опреме на краћим растојањима. Мобилна телефонија је једно од првих тржишта где је почела да се употребљава *Bluetooth* технологија, највише због тога што се *Bluetooth* чипови на лак начин могу додати у мобилни уређај [75]. Међутим, *Bluetooth* технологија је нашла своју примену и код других уређаја, као што су миш, штампачи, *PC*, *Hi-Fi* системи, али и код апликација развијених за аутомобиле.

Уређај који комуницира путем *Bluetooth* технологије може да ради или у *master* или у *slave* режиму рада. Максималан број уређаја који може бити повезан овом технологијом је осам – седам активних *slave* уређаја и један *master* уређај, и они образују тзв. *Piconet*. *Piconet* представља најједноставнију конфигурацију *Bluetooth* мреже. Овакве конфигурације се такође могу међусобно повезати образујући тзв. *Scatternet*.

Два *slave* уређаја не могу да комуницирају директно један са другим у било ком тренутку осим у фази откривања. Расподела канала и успостављање комуникације су у надлежности *master* уређаја. *Master* и *slave*

уређаји могу да замене улоге у случају када један уређај треба да се повеже у више од једне *Piconet* мреже.

Комуникација употребом *Bluetooth* технологија нема никакву зависност од *IP* адресе корисника. Оваквим дизајном се олакшава распоређивање уређаја због тога што не мора да се води рачуна о расподели адреса, подразумеваном рутеру, итд.

6.3.2 Wi-Fi (IEEE 802.11)

Wi-Fi (IEEE 802.11) протокол дефинише физички слој и *MAC (Medium Access Control)* подслој за бежичну комуникацију и углавном се користи за повезивање више рачунара, као проширење или замена за жичну мрежу - *LAN (Local Area Network)* [36]. Стандард *IEEE 802.11* је познатији под називом *Wi-Fi*. *Wi-Fi* омогућава бржу конекцију, већу удаљеност од базне станице и бољу сигурност у односу на *Bluetooth*. Мобилност уређаја је једна од битнијих ставки у бежичним мрежама. Клијент може да мења своју локацију слободно, било у оквиру једног сегмента мреже, било између различитих сегмената мреже.

IEEE је 1997. године одобрио стандард за бежичне локалне мреже - *WLAN (Wireless Local Area Network)* под називом *802.11* којим се специфицирају карактеристике уређаја са брзином преноса података од 1 до 2 Mb/s [36]. Након задовољавајућих резултата добијених од стране компанија *Lucent Technologies* и *Harris Semiconductors*, *IEEE* је направио проширење стандарда са бољим перформансама под називом *IEEE 802.11b*, који додатно подржава брзину протока од 5,5 и 11 Mb/s: највећи број тренутно доступних уређаја на тржишту су базирани на употреби ове технологије. Стандардом *802.11b* специфициране су неке од модификација у кодирању. Током 1997. године *IEEE* је такође објавио спецификацију стандарда *802.11a*. Таква спецификација се и даље односила на *MAC* и физички слој са радом у опсегу од 5 GHz. Током 2003. године *IEEE* је одобрио употребу стандарда *802.11g*, најновије верзије *802.11* стандарда. Стандард *802.11g* обезбеђује исте перформансе као и *802.11a*, са том разликом што

користи опсег од 2,4 GHz. Компатибилност са *802.11b* уређајима је загарантована.

Wi-Fi бежична мрежа је заснована на ћелијској архитектури, а основна ћелија се назива основни сервисни скуп (*BSS - Basic Service Set*). Основни сервисни скуп представља скуп мобилних или фиксних станица [77]. Уколико се станица помери тако да се налази изван свог основног сервисног скупа, она више није у могућности да директно комуницира са осталим члановима тог скупа. *BSS* је основа за независни основни сервисни скуп (*IBSS - Independent Basic Service Set*) и за проширени сервисни скуп (*ESS - Extended Service Set*). *IBSS* представља најједноставнију конфигурацију у којој станице могу да комуницирају директно, без употребе приступне тачке - *AP (Access Point)*. Више основних сервисних скупова може да образује проширену форму мреже. Компонента која се користи за повезивање више основних сервисних скупова у том случају се назива дистрибутивни систем (*DS*). Дистрибутивни систем са приступном тачком омогућава креирање мреже проширеног сервисног скупа произвољне величине и сложености.

Расположиви пропусни опсег је подељен на 14 делимично преклапајућих канала, а ширина сваког од њих износи 22 MHz [36]. Једанаест ових канала је доступно у Сједињеним Америчким Државама, 13 је доступно у Европи, а само један је доступан у Јапану. Сви уређаји у оквиру истог основног сервисног скупа користе исти канал.

6.3.3 ZigBee (IEEE 802.15.4)

ZigBee технологија је технологија ниске цене, са малом потрошњом енергије, за рад на малим растојањима у бежичним комуникационим мрежама. *ZigBee* дефинише физички слој и *MAC (Medium Access Control)* подслој за бежичну комуникацију. Првенствено је намењена за бежичне приватне мреже (*Wireless Personal Area Network – WPAN*) и представља једну од новијих технологија. *ZigBee* је синоним за *IEEE 802.15.4* протокол [52].

Брзина преноса података је мања него код *Wi-Fi* технологије. и износи 250 Kb/s. Због тога, *ZigBee* се највише користи код једноставнијих апликација

које преносе мале количине података, као што је употреба код кућних сензора за светло. *ZigBee* технологија има малу потрошњу енергије и намењена је за комуникацију на малим растојањима од 20 до 50 метара. *ZigBee* користи опсег од 900 MHz у САД, 868 MHz у Европи и 2,4 GHz у свету [134].

Овај стандард пружа спецификацију у зависности од способности уређаја и укључује два различита типа рада: *FFD - Full-Function Device* и *RFD - Reduced-Function Device*. Основна разлика је та што *RFD* уређаји имају редуковани стек и због тога захтевају мање меморијских ресурса, знатно су једноставнији и јефтинији.

ZigBee подржава топологију звезде, стабла и мреже. Основна карактеристика топологије звезде је да има један централни чвор који је повезан са свим чворовима у мрежи и да све поруке пролазе кроз централни чвор. Топологија стабла има један чвор на врху, који се назива корен, на који су повезани остали чворови, по принципу гранања. Порука унутар ове мреже путује уз стабло или низ стабло. Топологија мреже је слична топологији стабла, са том разликом што су неки чворови међусобно повезани.

Стандардни Андроид телефон или уређај, иако има подршку за изванредан број технологија за пренос података као што су *2G*, *3G*, *Wi-Fi*, *Bluetooth*, *FM*, итд, још увек нема подршку за *ZigBee*. Да би се омогућило коришћење *ZigBee* протокола на оваквим платформама, потребан је *ZigBee* чип [52].

6.3.4 3G и 4G технологије

3G (third generation) је трећа генерација мобилне телекомуникационе технологије [56]. То је скуп стандарда за мобилне уређаје и мобилне сервисе прописане спецификацијама од стране *International Mobile Telecommunications-2000 (IMT-2000)* у оквиру Међународне Телекомуникационе уније (*International Telecommunication Union*) [17]. *3G* технологија је нашла примену у бежичној телефонији, мобилном интернет приступу, видео позивима и мобилној телевизији.

Одређени број телекомуникационих компанија промовишу бежични мобилни интернет под називом *3G*, што значи да су дате услуге обезбеђене кроз *3G* бежичну мрежу. Такве услуге доступне су корисницима најчешће на нивоу месечне претплате. *3G* технологија подржава брзине преноса података од 384 Kb/s до 2 Mb/s, иако се у комерцијалној употреби најчешће користи брзина преноса података од око 100 Kb/s [78].

4G (fourth generation) је четврта генерација мобилне комуникационе технологије и представља наследницу *3G* технологије. Ова технологија обезбеђује веће брзине протока података у односу на *3G* [1]. Почетак употребе *4G* технологије сеже у 2007. годину када је у Јужној Кореји представљен *WiMAX* стандард ове технологије. У марту 2008. године, сектор за комуникације *International Telecommunications Union-Radio (ITU-R)* је дао спецификацију прописа које мора да задовољава *4G* стандард под називом *International Mobile Telecommunications Advanced (IMT-Advanced)*. Прописи у погледу брзине преноса података постављају се на око 100 Mb/s за комуникацију у стању високе мобилности (на пример из воза или аутомобила) и 1 Gb/s за стања ниске мобилности (на пример у току шетње или мировања) [57].

Покривеност сигналом код *4G* мреже је један од значајнијих проблема, чак и у развијеним земљама. Због тога су корисници ове технологије често ограничени на урбане средине и главне путне правце. Базне станице *4G* мреже су углавном лоциране у густо насељеним подручјима, а изван тих области пренос података се може остварити употребом *3G* мреже.

6.3.5 *Протоколи транспортног слоја TCP/IP* ***модела***

Транспортни слој у *TCP/IP (Transmission Control Protocol/Internet Protocol)* моделу представља везу између апликација и мреже [40]. Задатак транспортног слоја је одржавање вишеструких комуникација између апликација. Са гледишта протокола транспортног слоја, комуникација се

изводи између портова. Порт се може описати као тачка изворишта или одредишта података на рачунару.

TCP протокол је протокол транспортног слоја *TCP/IP* модела који омогућава поуздан пренос података. Овим протоколом преноси се неструктуриран континуалан ток октета (*Unstructured stream transfer*), а успостављање везе се обавља пре трансфера тока податка. *TCP* конекција је веза између два порта на истим или различитим хостовима. Идентификатор *TCP* конекције се представља са четири податка: изворишна *IP* адреса, изворишни *TCP* порт, одредишна *IP* адреса, одредишни *TCP* порт. *TCP* протокол се користи када је потребно обезбедити поуздан пренос података и управо због тога се многи протоколи апликативног нивоа (*HTTP*, *SMTP*...) ослањају на њега.

UDP протокол је протокол транспортног слоја *TCP/IP* модела који омогућава једноставну комуникацију између процеса. Овај протокол не омогућава поуздан пренос података, а трансфер података се обавља без претходног успостављања конекције. Овај протокол се користи тамо где апликације не захтевају поуздан пренос података и где су *UDP* функције довољне. Таквим апликацијама би употреба *TCP* протокола представљала проблем, пре свега због употребе великог заглавља и континуалног преноса података у малим количинама [40].

Основна дефиниција сокета (*Socket*) је крајња тачка комуникације. Овај механизам омогућава слање и примање тока података (*TCP*) или порука (*UDP*) између процеса на истој машини или преко мреже [10]. Сокет се описује са три податка: *IP* адреса, транспортни протокол (*TCP* или *UDP*) и број порта.

6.3.6 Web технологије

HTTP (*Hyper Text Transfer Protocol*) је мрежни протокол који представља главни и најчешћи коришћени метод преноса информација на интернету. *HTTP* је протокол за комуникацију између сервера и клијента и његов рад се базиран на употреби *TCP/IP*.

REST (Representational State Transfer) је протокол који је заснован на *HTTP* протоколу [68]. *REST* сервиси су независни од коришћене платформе и језика. *REST* може да користи *XML* формат порука, али и друге формате, као што су *CSV (Comma Separated Values)* или *JSON (JavaScript Object Notation)*. Модерни *web* сервиси су засновани на употреби *REST* технологије. Старији *web* сервиси углавном користе технологију *SOAP*.

SOAP (Simple Object Access Protocol) дефинише и прописује формат порука, и омогућава комуникацију између различитих апликација путем *HTTP* [69]. Независтан је од платформе и програмског језика. *SOAP* порука је *XML* документ.

6.4 Избор технологија за имплементацију софтверског система за даљинско програмирање, управљање и надзор работа

Популарност и доступност Андроид оперативног система и знатно већи број корисника у односу на друге мобилне платформе представља главни разлог избора ове технологије за имплементацију окружења за даљинско програмирање, управљање и надзор работа на паметним телефонима које је представљено у овој дисертацији. Распрострањеност Андроид уређаја и чињеница да је већина студената али и инжењера упозната са употребом ових уређаја даје Андроид ОС-у велику предност при избору оперативног система за имплементацију окружења за мобилно учење у роботизици. Ниска цена ових уређаја у односу на друге мобилне уређаје (као што су *Apple* уређаји) је такође једна од предности ове технологије, нарочито приликом опремања студената са одговарајућом опремом за учење у области роботике, али и опремању оператора у индустријским условима.

Андроид уређаји пружају различите комуникационе интерфејсе, као што су *Wi-Fi*, *Bluetooth*, или *USB*. Тренутно се за бежични пренос најчешће користе *Wi-Fi* и *Bluetooth* протокол. Поред њих разматрана је још једна, новија технологија за бежични пренос података са малом потрошњом

енергије, *ZigBee* технологија. У табели 6.1 дате су упоредне карактеристике *Bluetooth*, *Wi-Fi* и *ZigBee* технологије за бежичну комуникацију [36], [77].

Табела 6.1 Упоредне карактеристике *Bluetooth*, *Wi-Fi* и *ZigBee* технологије

	<i>Bluetooth</i>	<i>Wi-Fi</i>	<i>ZigBee</i>
Фреквенцијски опсег	2,4 GHz	2,4 GHz, 5 GHz	868/915 MHz, 2,4 GHz
Номинални распон	10 m	100 m	10 - 100 m
Максимална брзина преноса података	1 Mb/s	54 Mb/s	250 Kb/s
Номинална предајна снага	0 - 10 dBm	15 - 20 dBm	(-25) - 0 dBm

Како би у случају избора *ZigBee* технологије било потребно имати додатни чип, због тога што стандардни Андроид уређаји још увек немају подршку за њену примену, искључена је могућност избора ове технологије. Како *Wi-Fi* технологија омогућава бржу конекцију, већу удаљеност од базне станице, и бољу сигурност у односу на *Bluetooth* и како метода предложена у овој докторској дисертацији дефинише програмирање, управљање и надзор робота на већим растојањима, у случају када робот не може бити видљив, већ се надзор обавља даљински уз помоћ 3Д виртуелног окружења, за имплементацију је изабран *Wi-Fi (IEEE 802.11)* протокол. *3G* или *4G* технологија се може користити за спровођење даљинске комуникације са роботском лабораторијом у оквиру Андроид мобилног уређаја у случајевима када употреба *Wi-Fi* технологије на Андроид мобилном уређају није доступна.

TCP је протокол за поуздано испоручивање података где се успостављање везе обавља пре трансфера тока податка. Са друге стране, *UDP* не омогућава поуздан пренос података, а трансфер података се обавља без претходног успостављања конекције. Због потребе поузданог преноса података у току даљинског програмирања, управљања и надзора рада робота и наведених особина датих протокола, за имплементацију софтверског система који примењује методу за даљинско програмирање, управљање и

надзор робота предложену у оквиру ове докторске дисертације изабран је *TCP* протокол.

За *TCP/IP* слој подаци који се шаљу су бајтови који морају бити пренети са изворишта на одредиште. Када се користе *web* сервиси, подаци који се преносе морају бити у одговарајућем формату. Уколико се при комуникацији контролишу обе стране (и пошиљалац и прималац), као што је случај код предложеног решења, може се дефинисати посебан формат поруке, што може бити ефикасније од употребе готових решења. У зависности од тога који се подаци шаљу, структура порука може бити веома једноставна. *Web* сервиси су најпогоднији за употребу код апликација које су доступне на интернету, а решење које користи сокете и токове података (*stream*) је најбоље за комуникацију у локалним мрежама [45]. Ипак, када се користе уређаји са малим меморијским простором, као што су паметни телефони, *PDA*, итд, *stream* сокети могу бити бољи избор, зато што таква имплементација генерише програме који захтевају мањи меморијски простор.

7. ИМПЛЕМЕНТАЦИЈА СОФТВЕРСКОГ СИСТЕМА ЗА ДАЉИНСКО ПРОГРАМИРАЊЕ, УПРАВЉАЊЕ И НАДЗОР РОБОТА

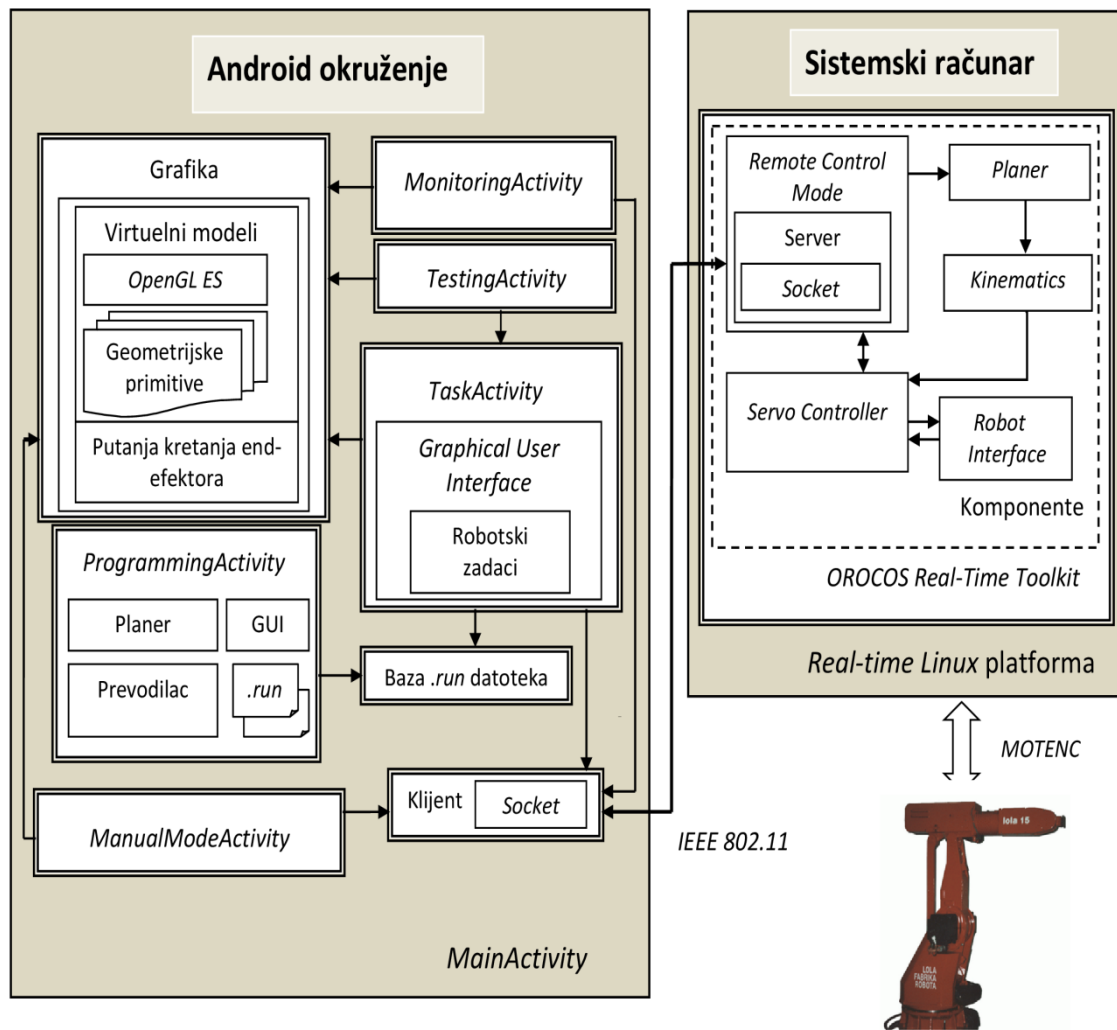
На основу методе за даљинско програмирање, управљање и надзор робота која је предложена у овој докторској дисертацији (поглавље 5.6), развијен је софтверски систем за даљинско програмирање, управљање и надзор индустријских робота [83] и уређаја за тренинг пилота и симулацију лета. Структура софтверског система дата је на слици 7.1. Систем се састоји од окружења на Андроид мобилном уређају и окружења на системском рачунару, а комуникација се обавља употребом *Wi-Fi* протокола.

7.1 Андроид окружење

За имплементацију Андроид окружења коришћен је Андроид/Јава програмски језик. Окружење *Eclipse SDK* [31] је коришћено као развојно окружење. Прва верзија је била намењена за Андроид ОС верзије 2.3.3 (*API level 10*), а последња верзија је намењена за Андроид ОС 4.3.

Као што је дефинисано предложеном методом, окружење може да ради у пет различитих режима рада:

- режим рада за програмирање,
- режим рада брзог бирања,
- ручни режим рада,
- режим рада за надзор и
- режим рада за тестирање.



Слика 7.1 Структура софтверског система за даљинско програмирање, управљање и надзор робота

У режиму рада за програмирање обавља се програмирање кретања робота, превођење програма на објектни код и повезивање дефинисаних програма са елементима графичког корисничког интерфејса режима рада брзог бирања. Графички кориснички интерфејс је имплементиран употребом Јава и XML програмског језика. Преводилац који преводи програме за кретање робота на објектни код имплементиран је употребом Јава програмског језика. У оквиру преводиоца укључени су алгоритми за различите врсте кретања робота, као и кинематика индустријских робота и

уређаја за тренинг пилота и симулацију лета (центрифуге и уређаја за просторну дезоријентацију пилота), који су развијени у Лола институту [72], [73], [74]. Подаци добијени у току процеса превођења се у формату објектног кода смештају у датотеке са екстензијом *.rnp* и представљају податке потребне за извршавање роботског програма/задатка на системском рачунару. Датотека са екстензијом *.rnp* садржи податке о жељеном кретању, брзини кретања, убрзању, као и позицијама. Такве датотеке се, након креирања, смештају у базу *.rnp* датотека у оквиру Андроид окружења. Због одређеног броја случајева у којима жељено кретање није могуће (превелика брзина осе, кретање изван опсега, итд), дефинисана су одговарајућа обавештења. Након успешног превођења, роботски програм се повезује са одговарајућим елементом графичког корисничког интерфејса за брзо бирање жељеног задатка и додељује му се жељено име. Поред *.rnp* датотека које су намењене за извршавање програма на системском рачунару, у току поступка превођења креирају се и табеле за симулацију. Такве табеле садрже податке о позицијама сваке од оса робота на сваких 15 ms током извршавања задатка.

У режиму рада брзог бирања и ручном режиму рада Андроид уређај ради као клијент. Клијент је имплементиран коришћењем сокета и Јава програмског језика. Догађаји који се генеришу кликом на одговарајуће елементе графичког корисничког интерфејса су повезани са функцијама за слање команди и потребних пратећих података на сервер. Дефинисане су следеће поруке које клијент може да пошаље серверу:

- управљачке поруке,
- *"speed dial"* поруке и
- поруке за ручно вођење оса робота.

Управљачке поруке обухватају поруке за покретање и заустављање рада робота и поруке за препознавање режима рада. *"Speed dial"* поруке су поруке за извршавање роботских програма/задатака у режиму рада брзог бирања. У оквиру графичког корисничког интерфејса, као и у бази Андроид окружења, постоји дефинисан одређен број тзв. *"speed dial"* програма за кретање робота (Роботски задаци). Могу се дефинисати нови *"speed dial"*

програми и постојећи се могу репрограмирати. “*Speed dial*” програми су повезани са својим “*speed dial*” порукама, односно порукама брзог бирања. Свака порука брзог бирања је повезана са одговарајућом датотеком која садржи податке за извршавање одговарајућег програма, односно задатка робота (*.run* датотека). Након првобитног повезивања програма за кретање робота са одговарајућим елементом графичког корисничког интерфејса режима рада брзог бирања, односно “*speed dial*” дугметом, једним кликом на “*speed dial*” дугме може се послати порука роботу који се налази на удаљеном месту путем бежичне везе. Порука садржи опис задатка који робот треба да изврши. Као повратне информације добијају се достигнуте позиције робота у унутрашњим координатама на основу којих се извршава 3Д приказ кретања виртуелног модела робота или путање енд-ефектора робота у оквиру Андроид окружења. Једном дефинисани програм, који је повезан са одговарајућим “*speed dial*” дугметом, се сваки наредни пут може покренути са само једним кликом на његово “*speed dial*” дугме, али се такође може и репрограмирати у оквиру режима рада за програмирање. Оваква имплементација омогућава ефикасно и једноставно бирање и извршавање програма и задатака намењених индустријским роботима, али и другим уређајима специјалне намене.

У ручном режиму рада се обавља ручно вођење оса робота. Поруке ручног режима рада које се размењују између Андроид уређаја и системског рачунара садрже информације о броју осе којом се управља и смеру кретања. У току ручног вођења робота, тренутне достигнуте позиције робота се могу пратити на 3Д моделу робота, али и у текстуалном формату на графичком корисничком интерфејсу корисника. Овакво управљање је погодно за тестирање ново-дефинисаних кретања и задатака робота пре извршавања задатака у режиму рада брзог бирања.

У оквиру Андроид уређаја, надзор робота се може извршавати у комбинацији са режимом рада брзог бирања или ручним режимом рада, али се може извршавати и независно, у посебном режиму рада за надзор. Надзор се врши кроз кретање 3Д модела робота или праћењем путање врха енд-ефектора робота у 3Д координатном систему. Команде које прима Андроид

уређај носе информацију о тренутној позицији робота у унутрашњим координатама. Када се повратна информација о раду робота добије, обрађује се са циљем да се добију вредности углова за сваку осу робота, а затим се добијене вредности подешавају на 3Д моделу робота. Ове промене се дешавају у интервалу од 15 ms и визуелно дају континуално кретање виртуелног модела робота, зато што се ради о веома малим временским интервалима у којима људско око не види прекид. У случају када се надзор врши посматрањем путање енд-ефектора робота, након пријема повратне информације и обраде добијеног садржаја, добијене вредности се морају додатно трансформисати у спољашње координате употребом директне кинематике робота.

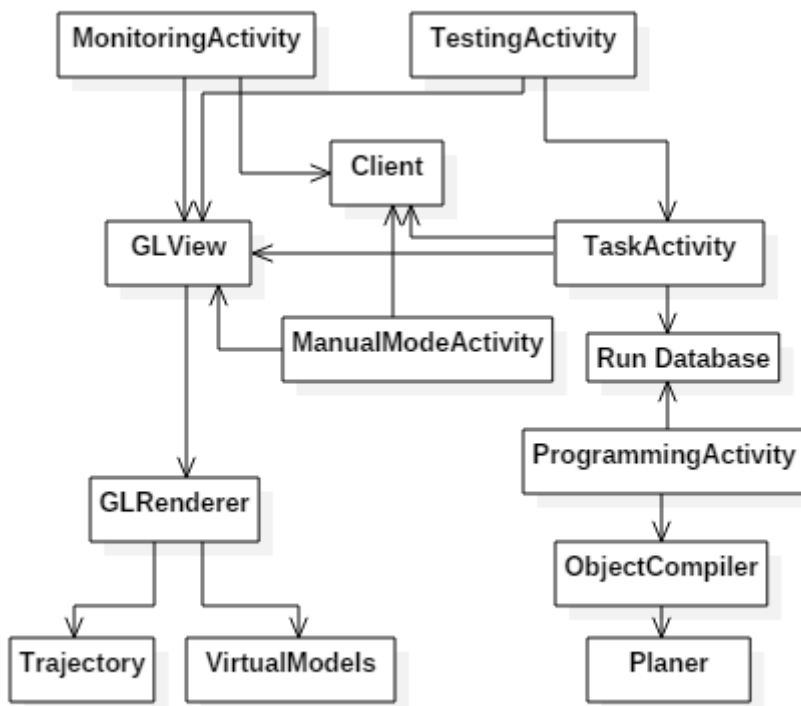
У режиму рада за тестирање, симулација извршавања роботских програма, односно задатака, се обавља као независна апликација. Симулација кретања робота употребом његовог виртуелног модела или кроз праћење путање енд-ефектора се одвија на основу података из табела за симулацију које су креиране током превођења програма за кретање робота. Функционалности као што су покретање и извршавање програма за кретање робота у режиму рада брзог бирања на нивоу симулације, као и повезивање сложеног програма за кретање робота са одговарајућим “*speed dial*” елементом графичког интерфејса, су такође укључене у режим рада тестирања.

7.1.1 Класе Андроид окружења

Упрошћени класни дијаграм Андроид окружења је дат на слици 7.2.

Класа *ProgrammingActivity* садржи функционалности везане за програмирање и репрограмирање роботских задатака. *ProgrammingActivity* обухвата имплементацију графичког корисничког интерфејса за програмирање врсте кретања, брзине, убрзања и позиција робота, односно инструкција које описују програм за жељено кретање робота. У оквиру ове активности се користе функционалности преводиоца објектног кода (класа *ObjectCompiler*) за превођење инструкција роботских задатака на бинарни

формат. Такође, у оквиру ове класе се врши генерисање табела са параметрима потребним за извршавање симулације кретања робота употребом његовог виртуелног модела или праћењем путање енд-ефектора. Функционалности за повезивање дефинисаног роботског задатка са одговарајућим називом и елементом графичког корисничког интерфејса режима рада брзог бирања су такође дефинисане у оквиру ове класе.



Слика 7.2 Класни дијаграм Андроид окружења

Класа *ManualModeActivity* омогућава управљање робота у ручном режиму рада, односно ручно вођење оса робота. Ова класа обухвата имплементацију графичког корисничког интерфејса ручног режима рада и функционалности за креирање одговарајућих порука на основу којих се извршава ручно вођење сваке од оса робота, а које се шаљу на системски рачунар.

Класа *TaskActivity* садржи имплементацију графичког корисничког интерфејса режима рада брзог бирања и функционалности за креирање

порука режима рада брзог бирања. Ова класа користи функционалности класе *Client* са циљем да иницира комуникацију са сервером (системским рачунаром) и изврши слање програма за кретање робота у формату објектног кода. У оквиру ове класе дефинисане су функционалности за приступ бази *.run* програма, односно програма за кретање робота у формату објектног кода, на основу задатог имена и позиције елемента графичког корисничког интерфејса.

Класа *MonitoringActivity* садржи имплементацију функционалности за даљински надзор. У оквиру ове активности се информације о тренутном положају робота у роботској лабораторији, добијене са системског рачунара, обрађују и постављају на виртуелним моделима робота или 3Д приказу путање енд-ефектора робота.

Класа *TestingActivity* садржи функционалности везане за симулацију кретања робота. Симулацију је могуће извршавати у комбинацији са режимом рада брзог бирања са циљем да се жељени програм изврши на виртуелном моделу робота.

Класа *VirtualModels* садржи имплементације 3Д модела робота. Имплементације 3Д модела робота су реализоване употребом елементарних геометријских примитива, што је детаљније објашњено у поглављу 7.1.3.

Класа *Trajectory* садржи имплементацију приказа кретања енд-ефектора робота у 3Д координатном систему.

Класа *ObjectCompiler* садржи имплементацију преводиоца који дефинисане инструкције кретања робота преводи у бинарни формат (по унапред дефинисаним правилима) и припрема за слање на системски рачунар. Ова класа користи функционалности класе *Planner* која укључује алгоритме за различите врсте кретања, као и кинематику индустријских робота, али и уређаја за тренинг пилота и симулацију лета (центрифуге и уређаја за просторну дезоријентацију пилота). Рад преводиоца је детаљније објашњен у поглављу 7.1.2.

Класа *Client* садржи функционалности за комуникацију мобилног уређаја са системском рачунаром путем *TCP* протокола и сокета. *Client*

обавља слање управљачких команди, “*speed dial*” команди, као и команди за ручни режим рада и прима информације о тренутном стању робота.

7.1.2 **Преводилац роботских програма**

Класа *ObjectCompiler* садржи имплементацију преводиоца који роботске програме дефинисане у оквиру графичког корисничког интерфејса за програмирање робота преводи у бинарне датотеке са екстензијом *.gip*. Такве датотеке садрже бинарни запис свих параметара који су од значаја и који су потребни за интерпретирање и извршавање програма за кретање робота на системском рачунару.

Садржај бинарне датотеке се формира по тачно дефинисаној структури која у потпуности усклађена са форматом циљног кода преводиоца *L-IRL* језика за *MOVE* кодну наредбу [112]. Таква имплементација је изабрана како би се интерпретирање наредби кретања које су добијене као резултат превођења у оквиру предложеног окружења намењеног мобилним уређајима обављало употребом постојећих функционалности интерпретера системског рачунара које се користе за интерпретирање излазног бинарног кода *L-IRL* преводиоца. Структура записа бинарних датотека са називима параметара потребних за интерпретирање и извршавање наредби *PTP* кретања индустријских Лола робота дата је на слици 7.3. Објашњења ознака параметара су описана у [73] и дата испод Сlike 7.3. Код *CP* кретања робота користе се слични интерполациони параметри који се односе на кретање врха енд-ефектора [73]. Структура записа бинарних датотека са називима параметара потребних за интерпретирање и извршавање наредби кретања центрифуге дата је на слици 7.4. Објашњења ознака параметара су дата испод Сlike 7.4. Структура записа бинарних датотека за програме кретања уређаја за просторну дезоријентацију пилота је слична структури бинарних датотека индустријских робота, са разликом у броју унутрашњих координата (код индустријских робота Лола 15 и Лола 50 број унутрашњих координата је 6, а код уређаја за просторну дезоријентацију пилота је 4). Параметри коју су садржани у оквиру бинарног записа су добијени на основу алгоритама за

различите врсте кретања робота, као и на основу кинематике робота [72], [73], [74]. Класа *Planer* садржи имплементацију тих алгоритама.

Move
Pos [6][Na+Nc+Nd]
Na
Nc
Nd
du_prev [6]
du_a [6]
du_d [6]
du_c [6]
tin2
tin4
tin5

Слика 7.3 Структура записа *.lup* датотека са називима параметара наредби *PTP* кретања индустријских Лола робота

Move - Ознака за *MOVE* инструкцију.

Pos [6] [Na+Nc+Nd] - Позиције робота у унутрашњим координатама (у сваком периоду интерполације, укупно *Na+Nc+Nd* интерполација); 6 степени слободе.

Na - Број интерполација при равномерно променљивом кретању на почетку сегмента кретања.

Nc - Број интерполација кретања константном брзином.

Nd - Број интерполација при равномерно променљивом кретању на крају сегмента кретања.

du_prev[6] - Кориговане вредности пређених путева чланова робота у последњем периоду интерполације претходног кретања.

du_a[6] - Разлике пређених путева чланова робота за време два узастопна периода интерполације при равномерно променљивом кретању на почетку сегмента кретања.

du_d [6] - Разлике пређених путева чланова робота за време два узастопна периода интерполације при равномерно променљивом кретању на крају сегмента кретања.

du_c[6] - Пређени путеви чланова робота за један период интерполације кретања константном брзином.

tin2 - Параметар купловања за другу осу робота (за робот Лола 15).

tin4 - Параметар купловања за четврту осу робота.

tin5 - Параметар купловања за пету осу робота.

GMove
Pos [3][Na+Nc+Nd]
Na
Nc
Nd
bl_g
acc_g
time

Слика 7.4 Структура записа *.rip* датотека са називима параметара наредби кретања центрифуге

GMove - Ознака за *GMOVE* инструкцију.

Pos [3] [Na+Nc+Nd] - Позиције центрифуге са 3 степена слободе у унутрашњим координатама (у сваком периоду интерполације, укупно $Na+Nc+Nd$ интерполација).

Na - Број интерполација при равномерно променљивом кретању на почетку сегмента кретања.

Nc - Број интерполација кретања константном брзином.

Nd - Број интерполација при равномерно променљивом кретању на крају сегмента кретања.

bl_g - Жељено убрзање које делује на пилота.

acc_g - Прираштај убрзања.

time - Време одржавања константног убрзања.

Програм за кретање робота не може бити сасвим произвољан, већ мора бити дефинисан у складу са одређеним ограничењима. Дефинисана су ограничења у погледу вредности брзина, убрзања, као и циљних и помоћних тачака кретања које се могу задати. У случају задавања неодговарајуће вредности која проузрокује немогућност превођења роботског програма корисник добија одговарајуће обавештење. Обавештење које корисник може добити као упозорење да програм за кретање индустријског робота није дефинисан у складу са постојећим ограничењима може бити једно од следећих:

- Погрешна брзина,
- Погрешно убрзање,
- Немогуће кретање,
- Кратка путања,
- Погрешна помоћна тачка,
- Немогућа почетна тачка,
- Велика брзина осе,
- Кретање изван опсега.

Уколико је програм дефинисан у складу са свим постојећим ограничењима, преводилац извршава превођење на бинарни код који је даље

могуће послати на системски рачунар на извршавање. Извршавање програма се покреће кликом на одговарајуће дугме графичког корисничког интерфејса мобилног уређаја у оквиру интерфејса режима рада брзог бирања.

Током превођења генеришу се и табеле које садрже вредности унутрашњих координата робота у току кретања на сваких 15 ms. Вредности унутрашњих координата робота током кретања се израчунавају на основу алгоритама садржаних у класи *Planer*. Добијене табеле се користе током процеса симулације кретања робота.

7.1.3 3Д анимација

Андроид садржи подршку за 3Д графику у виду *OpenGL ES* библиотека [94]. Модели индустријских робота Лола 15 и Лола 50, као и модели уређаја за тренинг пилота и симулацију лета, односно центрифуге и дезоријентатора, су имплементирани употребом Јава програмског језика, Андроид библиотека и *OpenGL ES* библиотека.

Имајући у виду брзину процесирања паметних мобилних телефона, као и то да се на њима углавном извршава више процеса истовремено, имплементација 3Д визуелизације мора бити једноставна, али опет у довољној мери подударна за спецификацијом и изгледом робота. Због тога се уместо употребе комплексних облика препоручује употреба једноставних геометријских примитива [58]. Компоненте које чине сваки од робота су поједностављене и описане употребом елементарних геометријских примитива: коцке, цилиндра и сфере [87], [92]. Позиције примитива су одређене релативно у односу на специфицирани координатни систем, а затим су примитиве које чине целине груписане у одговарајуће елементе. Покретни елементи робота су повезани одговарајућим везама, обртним или трансаторним. Сви параметри за сваки од 3Д модела робота су постављени у складу са дефинисаним кинематичким моделом робота [72], [73], [74]. Након комплетирања 3Д модела, осама виртуелног робота су додељене одговарајуће функције које узрокују подешавање њихових позиција у реалном времену на захтев системског рачунара или током симулације.

За имплементацију је коришћена Андроид класа *SurfaceView* која обезбеђује објекат *Surface* као подршку за цртање објеката из позадинске нити и опционо користи функционалности *OpenGL*-а за имплементацију различитог графичког приказа [94]. Ово је добра опција за захтевне графичке функционалности које се често ажурирају или које приказују графичке информације (3Д визуелизација).

Код којим се извршава анимација кретања 3Д модела робота са три степена слободе је дат у следећем тексту:

```
robot.setAngleAxis0(ang1);  
robot.setAngleAxis1(ang2);  
robot.setAngleAxis2(ang3);  
robot.draw(gl);
```

Променљива *robot* представља објекат истоименог типа који се састоји од покретних делова робота и базе робота, где је сваки од њих састављен од геометријских примитива и повезан са одговарајућом осом. Функција *setAngleAxisi (int ang)* обавља ротацију покретног дела робота који је повезан са осом под редним бројем *i*. Функција *draw(GL10 gl)* извршава цртање делова робота на задатим позицијама. Ова функција користи параметар *gl* који се односи на *Graphics Library – OpenGL ES* библиотеку.

У току даљинског надзора системски рачунар на Андроид уређај шаље достигнуте позиције робота за сваку његову осу (променљива *ang*). Информације о достигнутим позицијама се добијају у интервалима од 15 ms, чиме се иницира подешавање вредности углова оса и ажурирање садржаја на екрану мобилног уређаја. У току симулације подешавање вредности углова оса и ажурирање садржаја на екрану мобилног уређаја се извршава у интервалима од 15 ms, а вредности променљиве *ang* се добијају из табела генерисаних у поступку превођења роботског програма. Приликом 3Д приказа путање енд-ефектора робота, пре подешавања позиција у 3Д координатном систему, мора се обавити трансформација унутрашњих у

спољашње координате употребом директне кинематике робота. Као резултат се, у оба случаја (даљински надзор и симулација), добија визуелно континуално кретање 3Д модела робота, односно континуално исцртавање путање енд-ефектора робота, којима се описује кретање робота у лабораторији.

7.2 Систем за рад у реалном времену (системски рачунар)

Систем за рад у реалном времену имплементиран је употребом C++ програмског језика и *OROCOS Real-Time Toolkit* алата (*RTT*) [32]. *OROCOS Real-Time Toolkit (RTT)* обезбеђује инфраструктуру и функционалности за имплементацију роботских апликација које раде у реалном времену. Самим тим омогућено је подешавање и имплементација компонената за рад у реалном времену које уједно представљају највиши ниво овог дела управљачког система. Првобитна верзија окружења на системском рачунару је укључивала интерпретер роботских програма у бинарном формату [91], а комуникација са корисничким рачунаром је у овој верзији била имплементирана употребом *CORBA* протокола [33]. Функционалности овог система су проширене имплементацијом комуникације са Андроид мобилним уређајима употребом *Wi-Fi* или *3G/4G* технологија [83], [93], као и имплементацијом интерпретера “*speed dial*” порука и порука за ручни режим рада које се могу добити са мобилног уређаја. Сервер за комуникацију са Андроид мобилним уређајем и функционалности интерпретера су имплементирани употребом C++ програмског језика.

Апликативни слој овог система се састоји од компонената [34]. Дијаграм компонената је приказан на слици 7.5. Компонента *Server* прима поруке за даљинско управљање, ручно вођење, покретање или заустављање робота од стране Андроид мобилног клијента. Након пријема, такве поруке се шаљу компоненти *Remote_Control_Mode* која садржи интерпретер порука и обрађује информације о жељеном кретању робота.

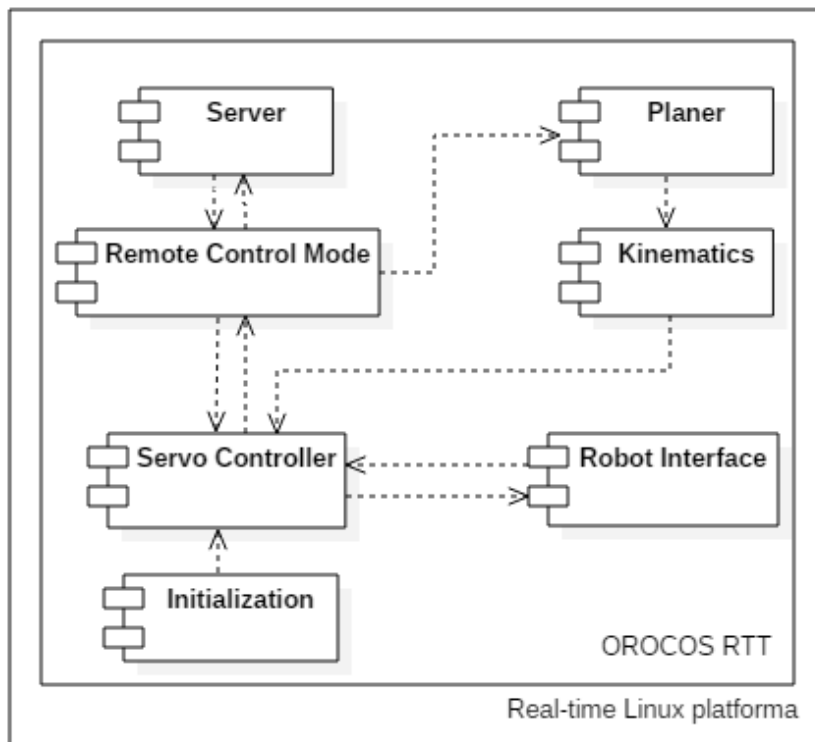
Уколико порука иницира извршавање одређеног задатка, она укључује и *.run* датотеку са параметрима потребним за извршавање програмираног задатка. Садржај *.run* датотеке се обрађује у оквиру интерпретера објектног кода који је саставни део *Remote_Control_Mode* компоненте. *Remote_Control_Mode* компонента затим прослеђује добијене податке компоненти *Planer*. Компонента *Planer* користи функционалности компоненте *Kinematics* која садржи кинематику робота. Компонента *Planer* представља генератор путање. Она садржи функције и алгоритме за рачунање референтне вредности позиције робота која треба да буде послата компоненти *Servo_Controller*. Вредности позиција се подешавају на сваких 5 ms. Сви алгоритми кретања робота су развијени у Лола институту [72], [73], [74]. Компонента *Initialization* генерише вредности позиција робота у току процеса иницијализације и шаље их *Servo_Controller* компоненти. Компонента *Servo_Controller* представља апстрактни систем контролер. На сваких 1 ms, *Servo_Controller* компоненти *Robot_Interface* шаље референтну вредност брзине која треба да буде достигнута. *Robot_Interface* компонента комуницира са погонима и мерним системима робота.

У току управљања у ручном режиму рада сервер прима поруке које садрже информације о жељеном смеру кретања и вредности помераја оса робота. На основу тих података се извршава подешавање позиција оса робота, израчунавањем и слањем референтне вредности позиције робота *Servo_Controller* компоненти.

Када сервер прими поруку за покретање или заустављање рада робота извршавају се следеће операције: иницијализација *Robot_Interface* компоненте, отпуштање кочница и омогућавање читања и уписивања на портове робота, односно, заустављање *Robot_Interface* компоненте, подизање кочница и онемогућавање читања и уписивања на портове робота.

Као резултат извршавања роботског програма или управљања у ручном режиму рада, достигнуте позиције робота се шаљу назад на Андроид мобилни уређај. Позиције које се шаљу представљају унутрашње координате робота. Слање позиција на Андроид уређај се обавља на сваких 15 ms. Поруке представљају скуп од неколико реалних вредности у текстуалном формату.

Свака вредност представља тренутно достигнуту позицију одговарајуће осе робота.



Слика 7.5 Дијаграм компонената система за рад у реалном времену

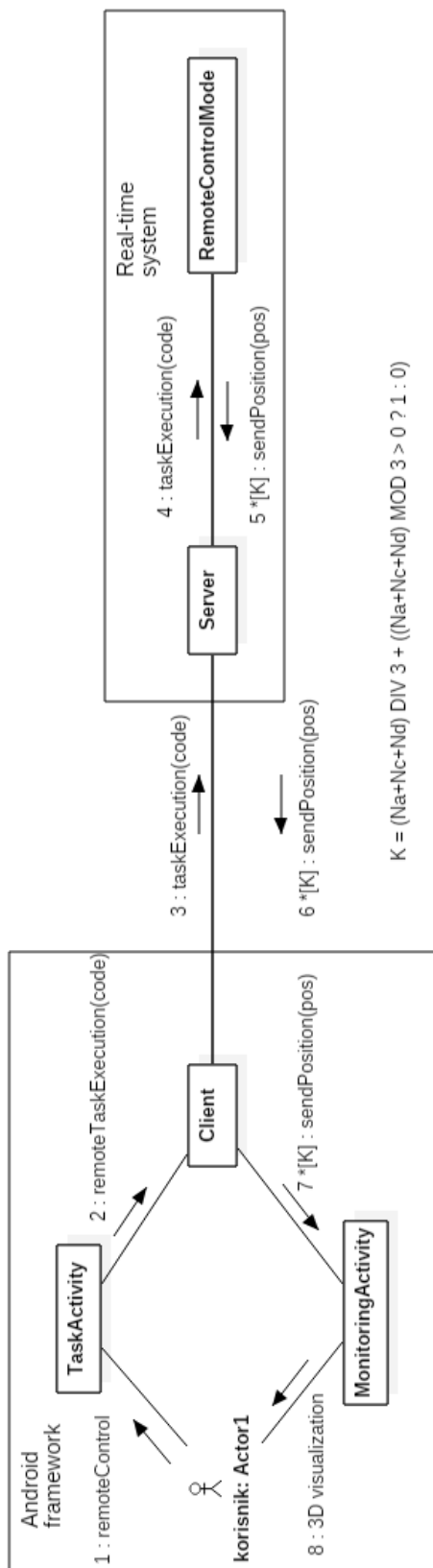
7.3 Комуникација између Андроид уређаја и системског рачунара

Комуникација између Андроид мобилног уређаја и системског рачунара у оквиру *Wi-Fi* мреже је имплементирана коришћењем *TCP/IP* (*Transmission Control Protocol/Internet Protocol*) сокета. Системски рачунар је повезан са роботом употребом *MOTENC* картица. Он извршава слање жељених позиција робота и читавање достигнутих позиција робота са енкодера.

Комуникациони дијаграм за случај даљинског управљања и надзора робота је дат на слици 7.6. Комуникација је описана за случај када корисник покреће извршавање жељеног задатка робота у режиму рада брзог бирања и

обавља даљински надзор тог извршавања, након што је робот покренут. Корисник даљински иницира извршавање жељеног програма кроз “*speed dial*” графички кориснички интерфејс, односно графички кориснички интерфејс режима рада брзог бирања (порука *remoteControl*). Након тога активност *TaskActivity* прослеђује поруку са садржајем *.run* датотеке клијенту (*remoteTaskExecution(code)*). Андроид клијент (*Client*) поруку којом се даљински покреће извршавање роботског програма (*taskExecution(code)*) шаље серверу (*Server*) чија се имплементација налази на системском рачунару. Сервер прослеђује поруку компоненти *Remote_Control_Mode* (*taskExecution(code)*) која обрађује добијени захтев са циљем да на основу садржаја поруке одреди да ли је потребно покренути извршавање *.run* датотеке или је потребно извршити ручно вођење неке од оса робота. Како је у овом случају потребно извршити *.run* програм, ова компонента прослеђује податке добијене интерпретирањем садржаја *.run* програма компоненти *Planner*, чија је улога објашњена у претходном поглављу. Након извршеног дела програма кретања за једну итерацију од 15 ms, односно три периода интерполације од 5 ms, а на основу података добијених са мерних система робота, компонента *Remote_Control_Mode* обавља формирање поруке са повратном информацијом о раду робота. Затим се порука (*sendPosition(pos)*) прослеђује серверу (*Server*). Сервер враћа повратну информацију Андроид уређају (*sendPosition(pos)*). Повратне поруке се примају на свака три периода интерполације (укупно $N_a + N_c + N_d \text{ DIV } 3$), где један период интерполације износи 5 ms, уз још једну додатну поруку која садржи позиције у последњем интерполационом периоду и која се прима уколико укупан број интерполација ($N_a + N_c + N_d$) није дељив са три. Клијент прихвата поруку и добијене вредности о достигнутим позицијама робота прослеђује активности *MonitoringActivity* (*sendPosition(pos)*) која обавља функционалности 3Д визуелизације кретања робота у реалном времену (*3D visualization*). 3Д визуелизација настаје као резултат постављања добијених достигнутих вредности позиција робота на виртуелном моделу робота у кратким временским интервалима (15 ms). На тај начин се добија визуелно континуално кретање виртуелног робота које приказује кретање робота у

лабораторији. У случају када се даљински надзор врши кроз праћење кретања енд-ефектора робота, добијене вредности у унутрашњим координатама морају се трансформисати у спољашње координате. Таква трансформација се врши употребом директне кинематике [72], [73], [74].



Слика 7.6 Дијаграм комуникације за даљинско управљање и надзор робота

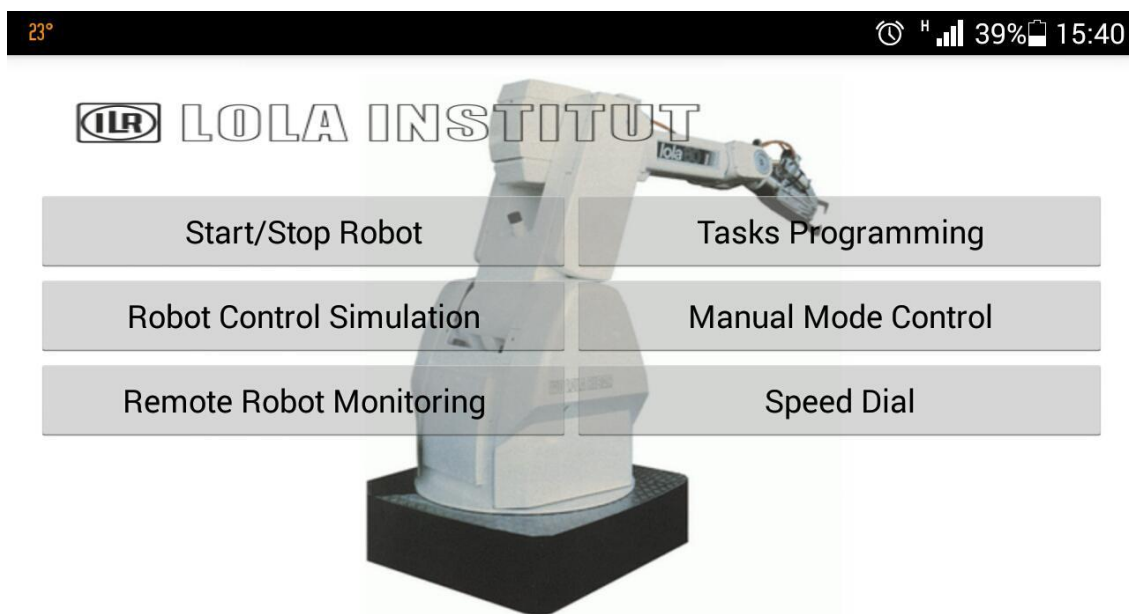
8. ОПИС СОФТВЕРСКОГ СИСТЕМА ЗА ДАЉИНСКО ПРОГРАМИРАЊЕ, УПРАВЉАЊЕ И НАДЗОР РОБОТА

Софтверски систем за даљинско програмирање, управљање и надзор робота садржи пет режима рада: режим рада за програмирање, режим рада брзог бирања, ручни режим рада, режим рада за надзор и режим рада за тестирање. Режим рада за програмирање омогућава програмирање робота без потребе детаљног познавања парадигми роботских језика и омогућава повезивање роботских програма са елементима графичког корисничког интерфејса режима рада брзог бирања. Режим рада брзог бирања омогућава извршавање сложених кретања робота уз значајно редуковање броја акција корисника у односу на корисничко окружење Лола УС-а код кога се интеракција са корисником заснива на употреби РС-ја и језика за програмирање робота. Ручно вођење оса робота у унутрашњим координатама је омогућено кроз ручни режим рада. Режим рада за надзор се може користити у комбинацији са режимом рада брзог бирања или ручним режимом рада, али се може користити и независно. Режим рада за тестирање омогућава тестирање и верификацију сложених роботских програма применом симулације, пре њихове примене на даљинско програмирање и управљање робота у роботским лабораторијама.

У окружење су укључени алгоритми којима се описују врсте кретања индустријских робота и уређаја за тренинг пилота и симулацију лета, као и алгоритми који описују њихову кинематику. Функционалности софтверског система се могу проширити додавањем нових робота и машина, укључивањем одговарајућих алгоритама у *Planer* класу Андроид окружења и подешавањем графичког корисничког интерфејса за програмирање. Подешавање графичког корисничког интерфејса брзог бирања није

потребно, јер се такав интерфејс може повезати са различитим програмима или роботима, и независан је од појединачних роботских инструкција.

Главна страна Андроид окружења за даљинско програмирање, управљање и надзор индустријског робота Лола 50 је дата на слици 8.1. У оквиру графичког корисничког интерфејса постоје опције за покретање, односно заустављање рада робота (*Start/Stop Robot*), дефинисање и редефинисање новог роботског задатка (*Tasks Programming*), симулацију рада робота (*Robot Control Simulation*), даљинско управљање у ручном режиму рада (*Manual Robot Control*), даљински бежични надзор (*Remote Robot Monitoring*), и опција за режим рада брзог бирања (*Speed Dial*).



Слика 8.1 Графички кориснички интерфејс окружења за даљинско програмирање, управљање и надзор индустријског робота Лола 50

Кликом на опцију за даљинско управљање (у режиму рада брзог бирања или ручном режиму рада) или даљински надзор приказује се садржај екрана за одговарајући режим рада и обавља се повезивање са сервером кроз задату *IP* адресу.

8.1 Програмирање робота

Програмирање индустријских робота је омогућено кроз једноставан и интуитиван графички кориснички интерфејс, који елиминише потребу за детаљним познавањем синтаксе и семантике роботских језика. Оваква функционалност је постигнута имплементацијом графичког корисничког интерфејса где се врсте кретања робота бирају у оквиру падајућег менија са дефинисаним кретањима робота, док се остали параметри, укључујући брзину кретања, убрзање, као и вредност циљне и помоћне тачке у спољашњим или унутрашњим координатама, дефинишу уношењем жељених вредности у текстуална поља. Ново кретање се додаје кликом на одговарајуће дугме графичког корисничког интерфејса. Након што се дефинишу све инструкције кретања у оквиру једног програма који описује жељени задатак робота и након што се зада жељени назив и редни број елемента графичког корисничког интерфејса за који ће се везати задати програм, кликом на дугме *Save Task* покреће се превођење програма. Након тога се извршавање програма за кретање робота иницира једним кликом на одговарајуће дугме у оквиру режима рада брзог бирања. На сличан начин се покреће и симулација програма за кретање робота.

8.1.1 Програмирање индустријских робота

У оквиру графичког корисничког интерфејса за програмирање индустријских робота, подржана су следећа кретања:

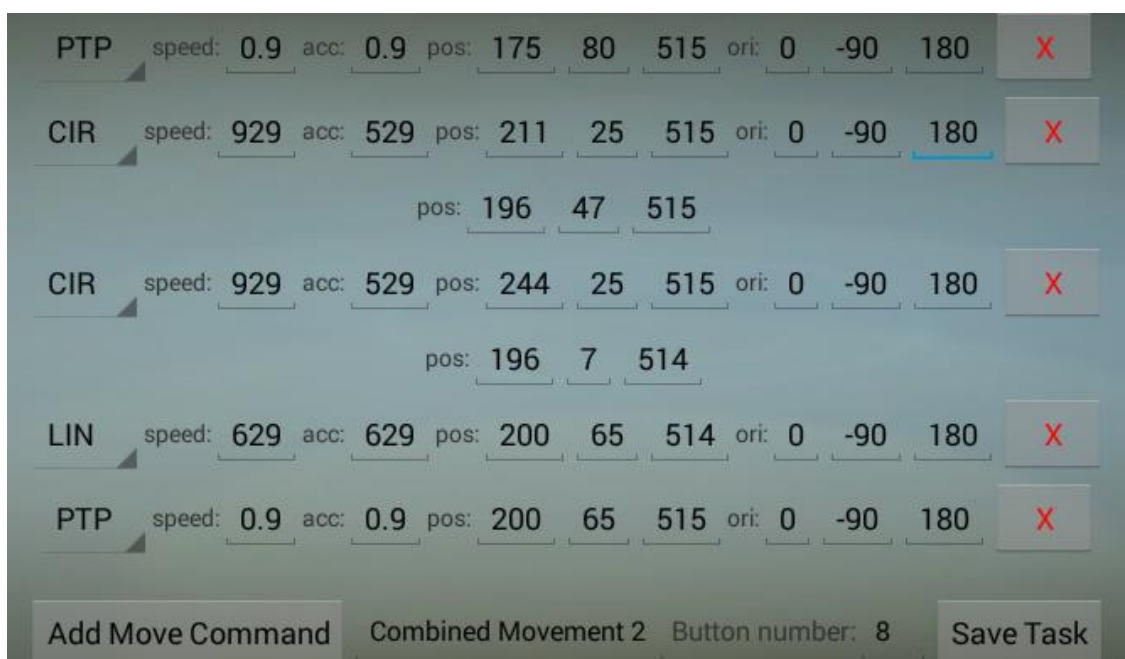
- тачка по тачка (*Point-to-Point - PTP*),
- кретање по математички дефинисаној путањи, кретање по линији (*LIN*),
- кретање по математички дефинисаној путањи, кретање по кругу у простору (*CIRCLE*);

Овакве врсте кретања су подржане за индустријске роботе Лола 15 и Лола 50. Таква кретања укључују могућност специфицирања следећих параметара:

- За кретање тачка-по-тачка и линеарно кретање је потребно дефинисати брзину кретања, убрзање и циљну тачку. Циљну тачку је могуће дефинисати у спољашњим координатама (дефинисањем позиције у Декартовом систему и оријентације) и у унутрашњим координатама, дефинисањем вредности угла за сваку осу.
- За кружно кретање је потребно дефинисати, брзину кретања, убрзање и циљну тачку у спољашњим или унутрашњим координатама, али и помоћну тачку. Помоћна тачка је потребна приликом дефинисања овог кретања и она се дефинише позицијом у Декартовом систему.

Графички кориснички интерфејс Андроид окружења за програмирање роботских задатака и њихово повезивање са режимом рада брзог бирања, на таблет уређају, приказан је на слици 8.2. Кретања робота се дефинишу бирањем жељене скраћенице у оквиру падајућег менија (*PTP* – кретање тачка-по тачка, *LIN* – линеарно кретање, *CIR* – кружно кретање). Вредности брзина и убрзања, као и вредности којима се дефинише циљна тачка (за све три врсте кретања) и помоћна тачка (за кружно кретање) се дефинишу уношењем жељених вредности у текстуална поља. Након дефинисања целокупног задатка робота, односно свих жељених инструкција кретања, роботски програм се повезује са одговарајућим елементом у оквиру графичког корисничког интерфејса режима рада брзог бирања. Пре тога је потребно унети редни број елемента интерфејса у поље поред ознаке *Button number:*, као и жељено име тог елемента у поље *Command name*. Овако дефинисани задатак се памти и преводи кликом на дугме *Save Task*, и повезује се са задатим елементом графичког корисничког интерфејса режима рада брзог бирања. У примеру који је дат на слици 8.2. дефинисана су два кружна кретања (*CIR*) са одговарајућим вредностима брзина и убрзања, и циљним и помоћним тачкама у спољашњим координатама. Затим је дефинисано једно линеарно кретање (*LIN*), и на крају кретање тачка-по-тачка, укључујући и одговарајуће вредности брзина и убрзања кретања и циљне тачке у спољашњим координатама. Све инструкције дефинишу

програмирање путање робота у координатама основе робота (*MOVE* наредба) [112], са тим да се у овој верзији окружења подразумева променљива оријентација. Извршавање инструкција кретања се обавља секвенцијално, а додавање нове инструкције кретања се врши кликом на дугме *Add Move Command*.



Слика 8.2 Графички кориснички интерфејс за програмирање кретања индустријских робота у спољашњим координатама

8.1.2 Програмирање уређаја за тренинг пилота и симулацију лета

Програмирање центрифуге се обавља употребом *GMOVE* инструкција. Овим инструкцијама се задаје жељено убрзање које делује на пилота - *bl_g*, и прираштај убрзања - *acc_g* или време одржавања константног убрзања - *time* у режиму отворене петље.

Графички кориснички интерфејс за програмирање центрифуге, на мобилном уређају, дат је на слици 8.3. Дефинисано је пет *GMOVE* инструкција. Како је у оквиру ових инструкција могуће направити избор између

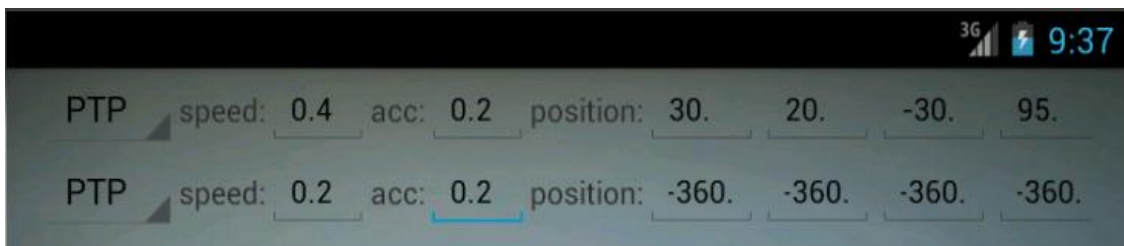
дефинисања прираштаја убрзања или времена одржавања константног убрзања, у оквиру датог интерфејса је омогућен избор жељеног параметра из падајућег менија. У првој, трећој и петој инструкцији је дефинисано време, а друга и четврта инструкција користе прираштај убрзања за специфицирање жељеног кретања. Параметар који дефинише вредност жељеног убрзања је обавезан. Овако дефинисан задатак се повезује са елементом графичког корисничког интерфејса режима рада брзог бирања на исти начин као и код графичког корисничког интерфејса за програмирање индустријских робота. Роботски задатак на слици 8.3 је повезан са елементом интерфејса режима рада брзог бирања под редним бројем 4 и именом *Gmove Program 2*.



Слика 8.3 Графички кориснички интерфејс за програмирање кретања центрифуге

Програмирање кретања уређаја за просторну дезоријентацију пилота се дефинише употребом наредби кретања тачка-по-тачка - *PTP* и следећих параметара кретања: брзина кретања, убрзање и циљна тачка. Циљна тачка се дефинише у унутрашњим координатама са четири вредности које представљају позиције сваке од четири осе робота.

Пример дела графичког корисничког интерфејса за програмирање кретања уређаја за дезоријентацију пилота је дат на слици 8.4. Програм се састоји из два кретања тачка-по-тачка.



Слика 8.4 Графички кориснички интерфејс за програмирање уређаја за просторну дезоријентацију пилота

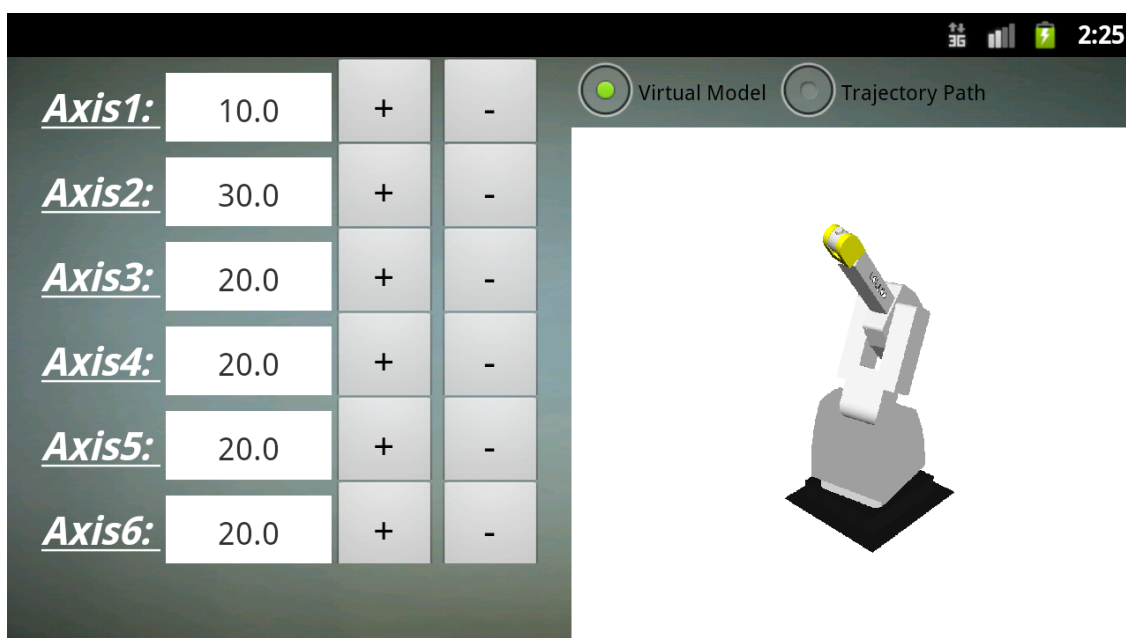
8.2 Управљање и надзор работа у ручном режиму рада

За сваки од индустријских робота као и уређаја за тренинг пилота и симулацију лета дефинисано је управљање у ручном режиму рада. Овако управљање представља ручно вођење оса робота са дефинисаним кораком помераја. Кретање се може обављати у позитивном и у негативном смеру. Управљање у ручном режиму рада је омогућено са удаљене локације уз праћење кретања робота кроз његов виртуелни модел, или уз праћење путање енд-ефектора робота у 3Д координатном систему.

Овакав начин управљања је погодан за случајеве тестирања и верификације ново-развијених алгоритама кретања робота. Управљање у ручном режиму рада може бити од користи и приликом примене у роботској лабораторији, где би се посматрањем кретања робота и посматрањем кретања његовог виртуелног модела истовремено, уз бежичну комуникацију, добила могућност посматрања малих помераја робота из више перспектива, што би допринело бољем увиду у сам рад и алгоритме кретања робота.

Изглед графичког корисничког интерфејса Андроид окружења за управљање индустријским роботом Лола 50 у ручном режиму рада, у

унутрашњим координатама, дат је на слици 8.5. У оквиру графичког корисничког интерфејса се приказују тренутне вредности углова за сваку осу робота. За индустријски робот Лола 50 са 6 степени слободе приказује се 6 поља са вредностима углова за сваку осу. Ручно вођење оса се може вршити у позитивном (+) и негативном смеру (-). Осе се могу померати у жељеном смеру све док се не достигну граничне вредности. Даљински надзор који се спроводи у оквиру датог графичког корисничког интерфејса, у комбинацији за управљањем у ручном режиму рада, се обавља употребом виртуелног 3Д модела индустријског робота Лола 50.

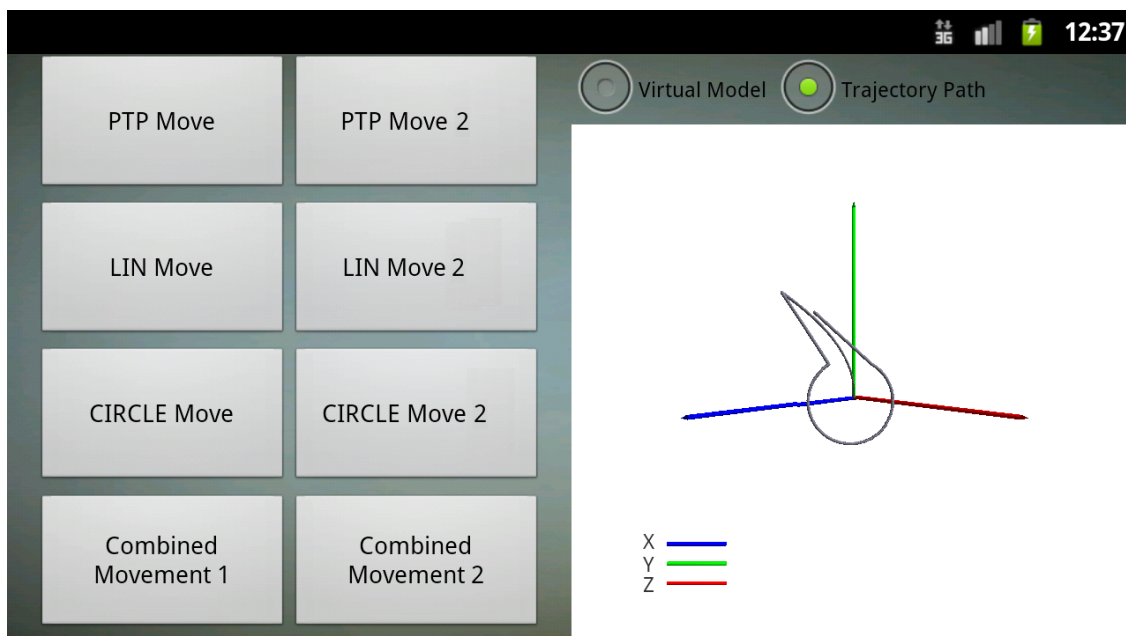


Слика 8.5 Графички кориснички интерфејс за управљање индустријског робота Лола 50 у ручном режиму рада, у унутрашњим координатама

8.3 Управљање и надзор робота у режиму рада „брзог бирања“ (*“speed dial”*)

Управљање индустријских робота и уређаја за тренинг пилота и симулацију лета је омогућено у режиму рада брзог бирања (*“speed dial”*). Режим рада брзог бирања даје могућност даљинског иницирања извршавања

роботског програма са само једном акцијом корисника, односно једним кликом на екрану мобилног уређаја, независно од његове локације. У оквиру овог режима рада се такође може обављати даљинско праћење кретања робота кроз његов 3Д модел или кроз кретање енд-ефектора у 3Д координатном систему. Роботски задаци, који се могу покренути кроз графички кориснички интерфејс режима рада брзог бирања, се могу дефинисати и касније редефинисати у оквиру графичког корисничког интерфејса за програмирање кретања робота. Дефинисани програм кретања се приликом програмирања повезује са жељеним именом и елементом графичког корисничког интерфејса режима рада брзог бирања. Покретање роботских задатака у режиму рада брзог бирања је независно од врсте робота и кретања која се могу програмирати и не мора се подешавати према специфичном роботу и инструкцијама кретања, већ се само повезује са жељеним дефинисаним низом инструкција кретања робота у оквиру графичког корисничког интерфејса за програмирање кретања робота. У случају да постоји неко ограничење које узрокује немогућност извршавања "speed dial" програма, кориснику се враћа одговарајуће обавештење.



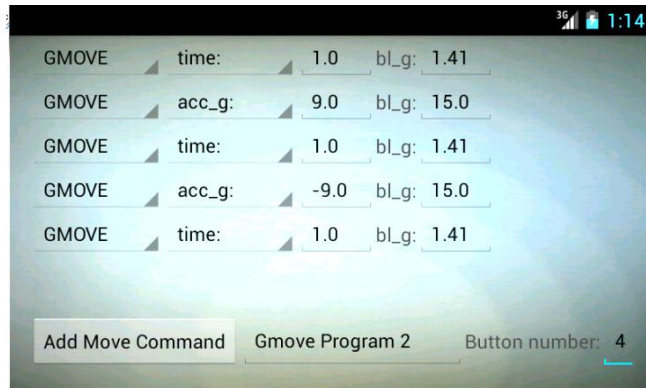
Слика 8.6 Графички кориснички интерфејс режима рада брзог бирања уз праћење путање енд-ефектора робота у 3Д координатном систему

Графички кориснички интерфејс режима рада брзог бирања са осам дефинисаних задатака индустријског робота Лола 50 је приказан на слици 8.6. У оквиру датог графичког корисничког интерфејса даљински надзор робота се обавља праћењем путање његовог енд-ефектора у 3Д координатном систему.

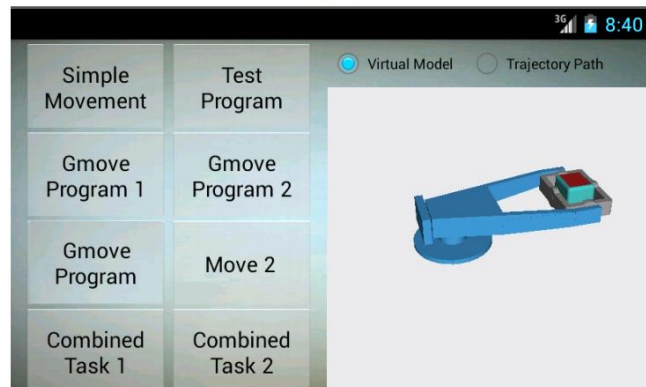
8.4 Симулација и тестирање програмираних задатака

Предложени софтверски систем укључује имплементацију симулатора који кроз 3Д моделе робота или 3Д приказ кретања енд-ефектора робота омогућава симулацију дефинисаних програма за кретање робота. Симулатор је имплементиран за индустријске роботе Лола 15 и Лола 50 и за уређаје за тренинг пилота и симулацију лета, односно центрифугу и уређај за просторну дезоријентацију пилота.

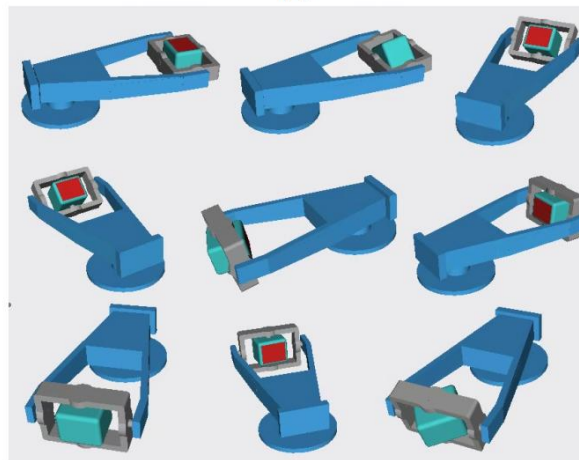
Симулација се покреће кликом на дугме у оквиру графичког корисничког интерфејса режима рада брзог бирања, а резултати симулације се прате на 3Д моделу робота или посматрањем резултујуће путање енд-ефектора робота у 3Д координатном систему. Поступак извршавања симулације кретања центрифуге по корацима је приказан на слици 8.7. Након дефинисања програма у оквиру графичког корисничког интерфејса за програмирање (Слика 8.7 (а)), он се преводи у одговарајући формат и повезује се са одговарајућим елементом графичког корисничког интерфејса под редним бројем 4 и именом *Move Program 2* (Слика 8.7 (б)). Кликом на дугме графичког корисничког интерфејса брзог бирања под називом *Move Program 2* покреће се симулација кретања центрифуге. Неколико положаја током симулације кретања дато на слици 8.7 (в).



(a)



(б)

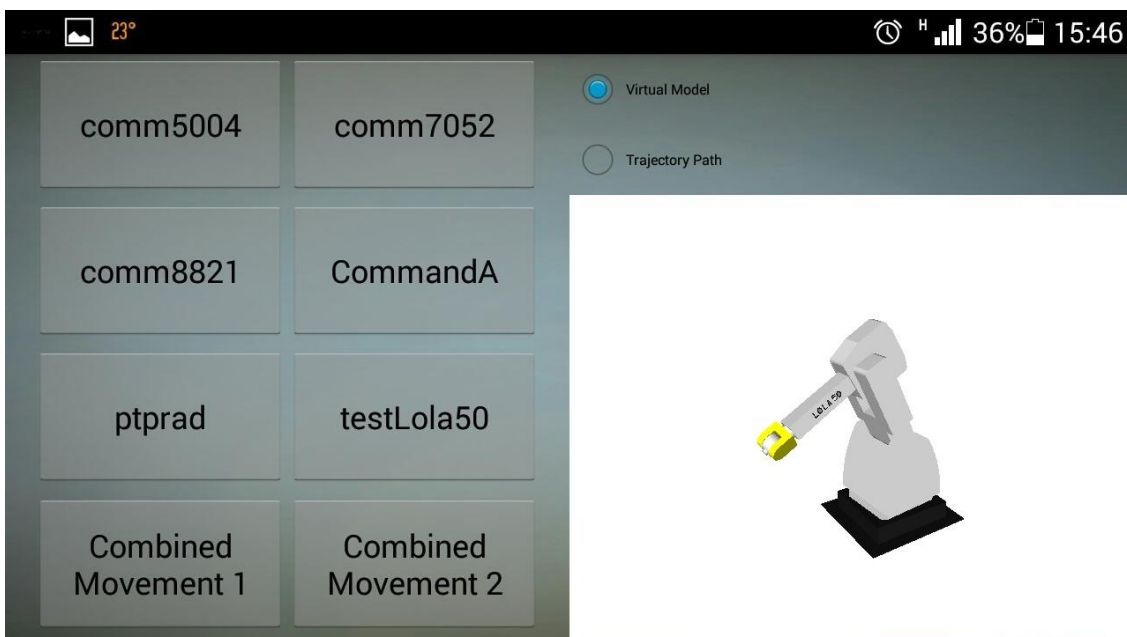


(в)

Слика 8.7 Поступак извршавања симулације кретања центрифуге
 (а) Дефинисање програма за симулацију кретања центрифуге у режиму рада
 за програмирање
 (б) Графички кориснички интерфејс режима рада брзог бирања за
 симулацију кретања центрифуге
 (в) Неколико положаја центрифуге током извршавања симулације програма
Gmove Program 2

8.5 Даљински надзор робота

Даљински надзор робота се обавља бежично и независно од локације корисника на Андроид мобилним уређајима. Даљински надзор је омогућен кроз праћење кретања 3Д модела индустријских робота и уређаја за тренинг пилота и симулацију лета или кроз праћење путања њихових енд-ефектора у 3Д координатном систему. Даљински надзор се може вршити у комбинацији са управљањем у режиму рада брзог бирања, у комбинацији са управљањем у ручном режиму рада или самостално. Оваква имплементација омогућава слободно кретање корисника и његову локацијску независност, унапређује безбедност оператера, али и отвара нове могућности при обучавању студената у области едукације у роботизици. Графички кориснички интерфејс Андроид окружења за управљање у режиму рада брзог бирања и даљински надзор употребом виртуелног модела индустријског робота Лола 50 је приказан на слици 8.8.



Слика 8.8 Графички кориснички интерфејс за даљинско управљање и надзор индустријског робота Лола 50

8.6 *Robotics First* – окружење за едукацију у роботици

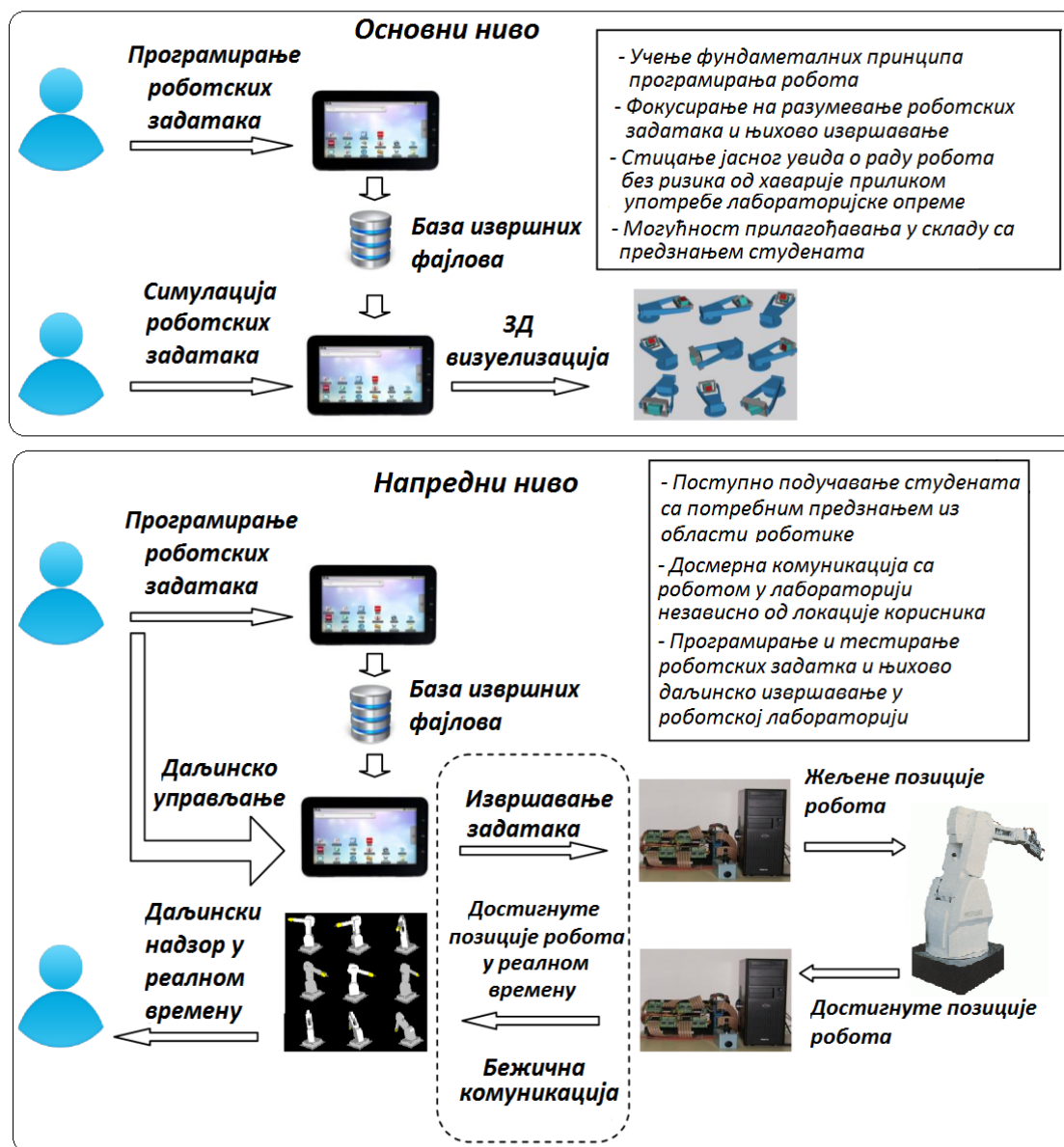
У оквиру ове докторске дисертације предложено је да се описана метода за даљинско програмирање, управљање и надзор робота и софтверски систем који примењује осмишљену методу примене за потребе едукације у роботици. Предложено окружење је погодно за едукацију студената због тога што нуди неколико различитих режима рада који се могу користити у зависности од предзнања ученика и студената. Окружење је у публикованим радовима названо *Robotics First*. *Robotics First* укључује могућност програмирања, управљања и надзора неколико робота са различитим бројем степени слободе, а укључује и могућност проширења функционалности додавањем нових робота и алгоритама.

Функционалности предложеног софтверског система су груписане у одговарајуће нивое учења, у складу са потребама ученика без искуства у раду са роботским системима, који се тек упознају са радом оваквих система, као и у складу са потребама студената са одговарајућим предзнањем из области математике и програмирања. Главне функционалности основног и напредног нивоа учења окружења *Robotics First* су приказане на слици 8.9.

Две главне функционалности основног нивоа учења окружења *Robotics First* су програмирање роботских задатака и симулација роботских задатака. Интерфејс за програмирање који не захтева детаљно познавање парадигми за програмирање робота омогућава фокусирање на разумевање роботских задатака и њихово извршавање. Симулација употребом 3Д визуелизације (праћењем кретања виртуелног робота и праћењем кретања енд-ефектора робота) омогућава ученицима да стекну јасан увид у резултате извршавања дефинисаних роботских програма. Таква имплементација такође омогућава ученицима да науче да програмирају и управљају роботима без непотребног ризика од хаварија које се могу десити у случају различитих грешака при програмирању робота у роботским лабораторијама.

Главне функционалности напредног нивоа учења су даљинско програмирање, управљање и надзор робота у реалном времену, путем локалне рачунарске мреже или путем интернета. Функционалности напредног нивоа учења дају могућност студентима да стекну практично искуство у управљању роботима, независно од њихове локације и без потребе за обезбеђивањем роботске лабораторије и додатне скупе опреме, али и уз смањену потребу за асистенцијом професионалаца. Употреба бежичне комуникације за даљински надзор пружа могућност надзора рада робота посматрањем кретања робота у лабораторији и истовременим посматрањем кретања виртуелног робота на мобилном уређају, али из друге перспективе. На овај начин се може обављати анализа индустријских процеса и проучавање понашања робота из више перспектива са циљем да се добије бољи увид у сам рад система.

Једноставно подешавање сложености роботских програма и избор одговарајућег нивоа учења, у складу са предзнањем ученика и студената, пружа могућност да они поступно науче како да програмирају и тестирају роботске програме и како да обављају управљање роботским и индустријским процесима.



Слика 8.9 Функционалности окружења *Robotics First*

9. ПРИМЕНА И ВЕРИФИКАЦИЈА СОФТВЕРСКОГ СИСТЕМА ЗА ДАЉИНСКО ПРОГРАМИРАЊЕ, УПРАВЉАЊЕ И НАДЗОР РОБОТА

Софтверски систем који је предложен и описан у овој докторској дисертацији, примењен је за даљинско програмирање, управљање и надзор индустријског робота Лола 15 у лабораторијским условима, и за програмирање и симулацију кретања индустријског робота Лола 50, као и уређаја за тренинг пилота и симулацију лета, односно центрифуге моделоване као манипулатор са 3 степена слободе и уређаја за просторну дезоријентацију пилота који је моделован као манипулатор са 4 степена слободе.

Како би се верификовало да се програмирање у оквиру предложеног решења врши у складу са првом сугестијом пројектантима мобилних апликација датом у раду [37], по којој број акција на тастатури потребних да се изврши неки догађај треба ограничити, у току програмирања мерено је колико је акција оператера потребно како би се написао, односно унео, комплетан програм у оквиру окружења на мобилном Андроид уређају. Након тога је извршено програмирање истих програма у оквиру *L-IRL* окружења код кога се интеракција са корисником, односно оператером, базира на употреби *PC*-ја и такође је измерено колико је акција корисника потребно како би се написао комплетан програм. Добијени резултати су упоређени и приказани графички. Узето је у обзир да акција корисника у оквиру екрана осетљивог на додир може бити “клик” на неки од елемената графичког корисничког интерфејса или унос једног карактера у текстуална поља [121]. Акција корисника у оквиру *L-IRL* окружења може бити “клик” или “двоклик” на неки од елемената графичког корисничког интерфејса употребом миша, или унос

једног карактера употребом тастатуре у оквиру *L-IRL* едитора за програмирање.

Верификација предложене методе за програмирање и управљање робота је извршена поређењем укупног броја акција корисника, односно оператера, потребних за програмирање и извршавање применом предложеног софтверског система базираног на Андроид ОС-у и применом *L-IRL* програмског језика и Лола УС-а код кога се интеракција са корисником базира на употреби *PC* –ја, за пет тест програма датих у следећем поглављу.

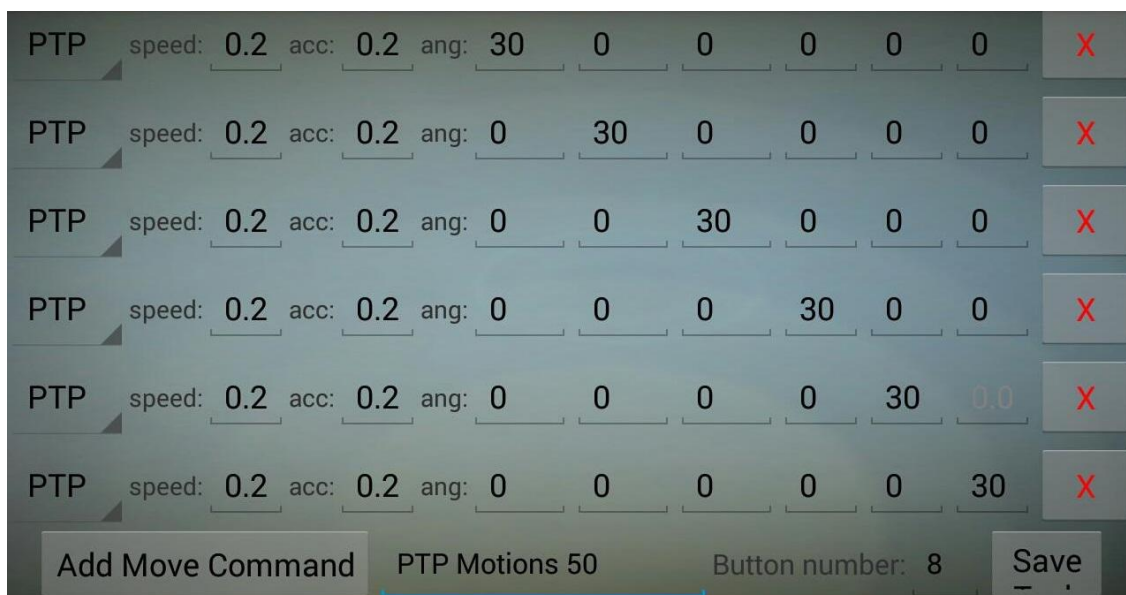
9.1 Даљинско програмирање, управљање и надзор индустријских робота

Софтверски систем за даљинско програмирање, управљање и надзор робота је примењен за програмирање и симулацију индустријског робота Лола 50 и за даљинско програмирање, управљање и надзор индустријског робота Лола 15.

Први дефинисани тест програм је програм за кретање робота Лола 50 који садржи шест *PTP* инструкција. Сваком инструкцијом је задато померање једне од оса робота за 30 степени почевши од прве осе. Циљне позиције су програмиране у унутрашњим координатама. На слици 9.1 је дат први тест програм у оквиру графичког корисничког интерфејса мобилног уређаја.

Након успешног превођења, програм је повезан са елементом графичког корисничког интерфејса за режим рада брзог бирања под редним бројем 8 и додељено му је име *PTP Motions 50*. Кликом на елемент графичког корисничког интерфејса режима рада брзог бирања под редним бројем 8 покренута је симулација програма. Извршавање је праћено кроз 3Д виртуелни модел робота. Вредности позиција робота су добијене као резултат извршавања програма и оне су записане у одговарајућу текстуалну датотеку. Те вредности представљају позиције робота у унутрашњим координатама на сваких 15 ms током симулације дефинисаног програма за кретање робота. Дијаграм промене вредности унутрашњих координата за

сваку осу робота је приказан на слици 9.2. На слици 9.3 приказане су позиције виртуелног модела робота Лола 50 након сваке *PTP* инструкције кретања.



Слика 9.1 Програм *PTP Motions 50*

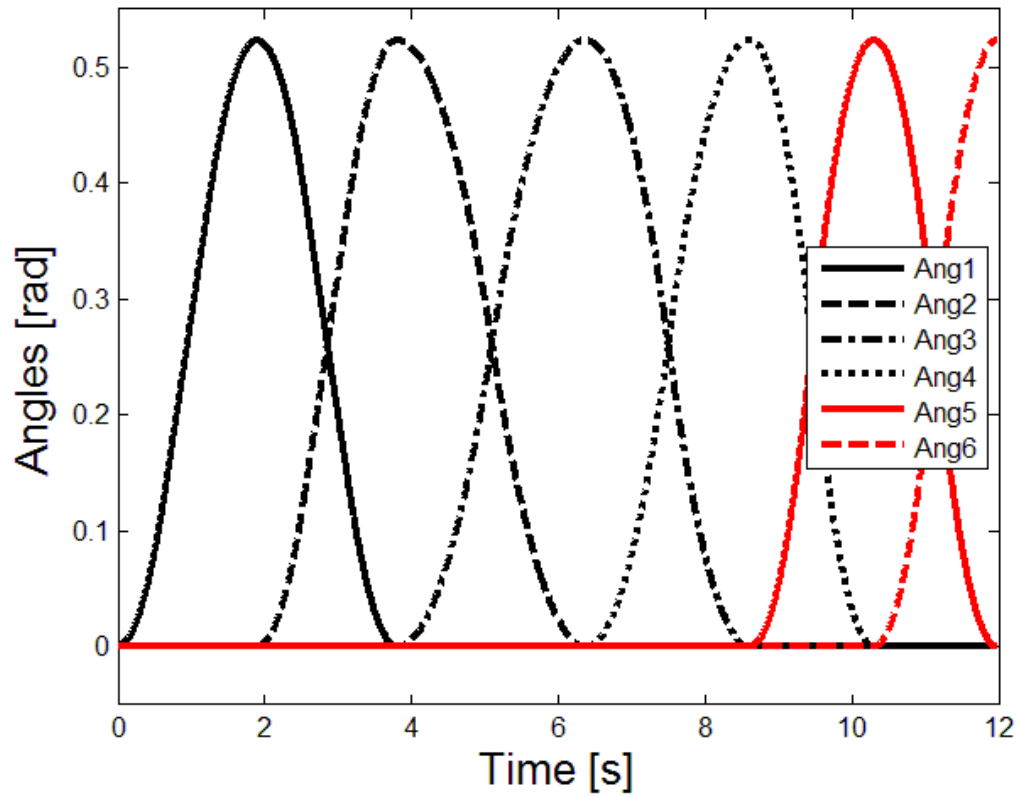
Програм *PTP Motions 50* је затим програмиран *L-IRL* програмским језиком и извршен применом Лола окружења базираног на употреби *PC*-ја за интеракцију са корисником које је описано у поглављу 3.2. Програм *PTP Motions 50* написан на *L-IRL* језику је дат у следећем тексту.

```

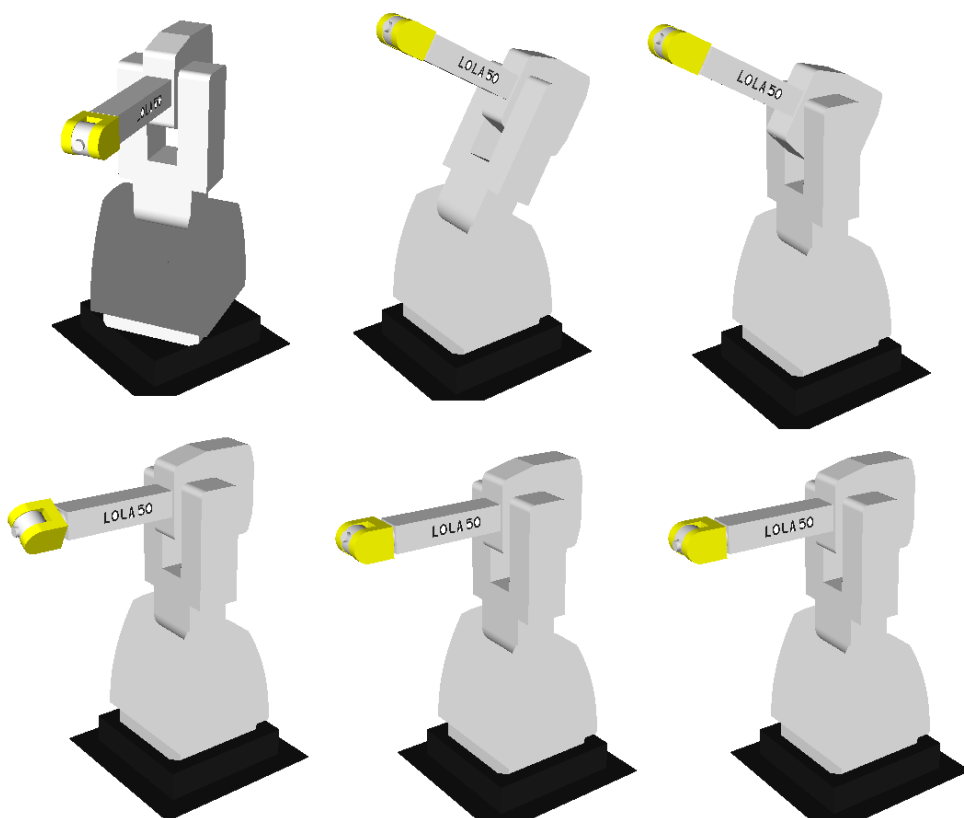
program PTP_Motions_50;
system_specification "lola50.spc";
seq
  move ptp joint(mainjoint(30,0,0,0,0,0)) speed_ptp :=0.2
acc_ptp := 0.2;
  move ptp joint(mainjoint(0,30,0,0,0,0)) speed_ptp :=0.2
acc_ptp := 0.2;
  move ptp joint(mainjoint(0,0,30,0,0,0)) speed_ptp :=0.2
acc_ptp := 0.2;
  move ptp joint(mainjoint(0,0,0,30,0,0)) speed_ptp :=0.2
acc_ptp := 0.2;
  move ptp joint(mainjoint(0,0,0,0,30,0)) speed_ptp :=0.2
acc_ptp := 0.2;
  move ptp joint(mainjoint(0,0,0,0,0,30)) speed_ptp :=0.2

```

```
acc_ptp := 0.2;  
endseq  
endprogram;
```

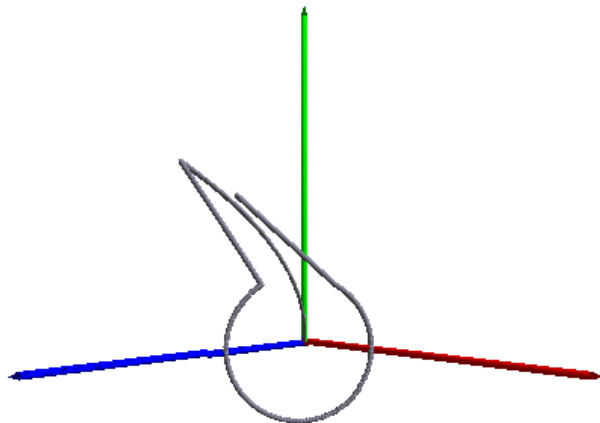


Слика 9.2 Позиције робота Лола 50 у унутрашњим координатама током симулације програма *PTP Motions 50*



Слика 9.3 Позиције виртуелног модела робота Лола 50 након сваке *PTP* инструкције кретања у току симулације програма *PTP Motions 50*

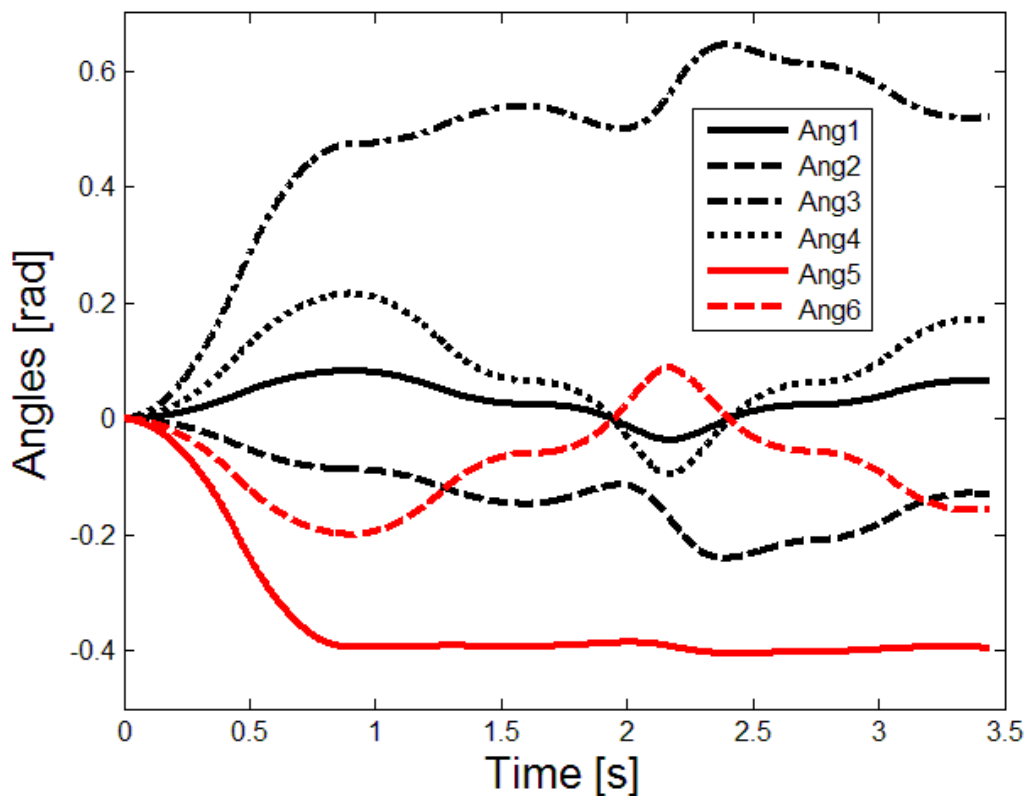
Други тест програм под називом *Combined Movement 2* дефинише кретање робота Лола 15 и укључује две *PTP*, две *CIRCLE* и једну *LIN* инструкцију кретања. Циљне позиције су програмиране у спољашњим координатама. Програм *Combined Movement 2* у оквиру графичког корисничког интерфејса таблет уређаја је приказан на слици 8.2.



X ———
 Y ———
 Z ———

Слика 9.4 Путања енд-ефектора робота у току извршавања програма *Combined Movement 2* приказана у оквиру окружења на Андроид уређаја

Превођење програма је извршено на исти начин као и у претходном случају. Кликком на елемент графичког корисничког интерфејса режима рада брзог бирања под редним бројем 8 извршавање програма је иницирано даљински. Извршавање је праћено кроз кретање енд-ефектора робота у 3Д координатном систему на мобилном уређају. Вредности позиција робота које су достигнуте, достављане су као повратне информације са системског рачунара путем бежичне мреже на сваких 15 ms. Те вредности представљају позиције робота у унутрашњим координатама, а на основу њих су применом директе кинематике добијене вредности позиција робота у спољашњим координатама, потребне за приказ путање енд-ефектора робота. Путања енд-ефектора робота у току извршавања програма *Combined Movement 2*, која је праћена у оквиру окружења мобилног Андроид уређаја, је приказана на слици 9.4. Путања енд-ефектора је формирана на основу добијених вредности достигнутих позиција робота у унутрашњим координатама које су приказане на слици 9.5.



Слика 9.5 Позиције робота у унутрашњим координатама током извршавања програма *Combined Movement 2*

Програм *Combined Movement 2* је затим програмиран *L-IRL* програмским језиком и извршен применом Лола УС-а који је описан у поглављу 3.2. Програм *Combined Movement 2* написан на *L-IRL* језику је дат у следећем тексту. Вредности брзине и убрзања *PTP* кретања у овој верзији програма су дефинисане као константе.

```

program Combined_Movement_2;
system_specification "lolal15.spc";
const
  real:brzina_ptp:=0.9;
  real:ubrzanje_ptp:=0.9;
endconst
seq
  move ptp robtarget (pose(position(175,80,515),
orientation(0,-90,180))) speed_ptp:=brzina_ptp acc_ptp:=ubrzanje_ptp;

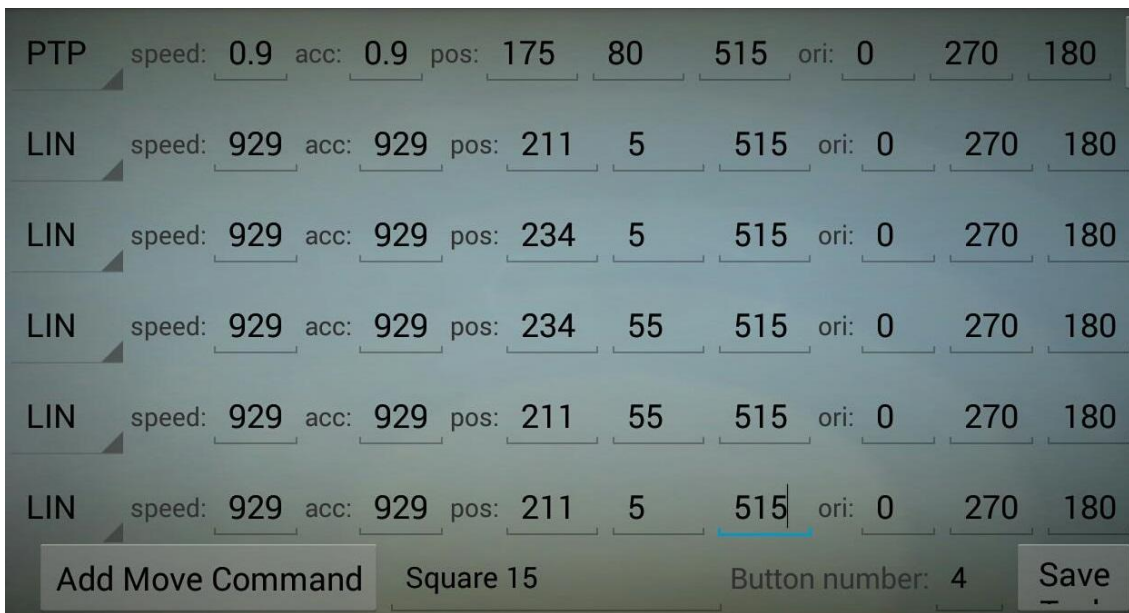
```

```

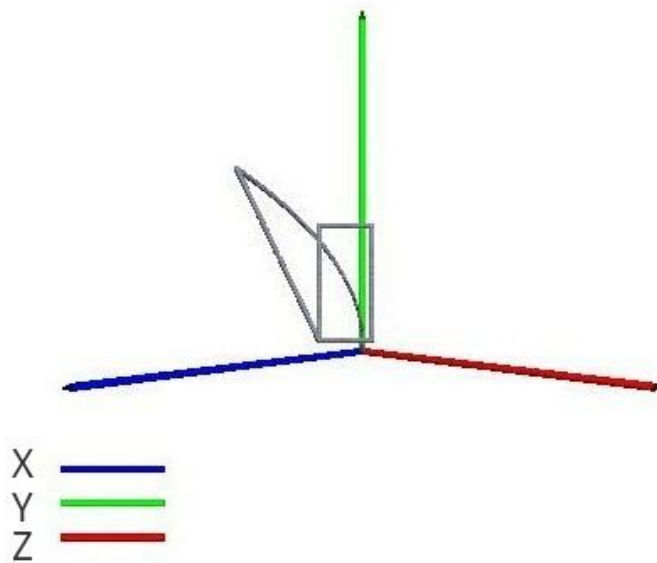
        move          circle          robtarget (position(196,47,515)),
robtarget (pose (position (211,25,515), orientation (0,-90,180)))
speed:=929 acc:=529;
        move          circle          robtarget (position(196,7,514)),
robtarget (pose (position (244,25,515), orientation (0,-90,180
        )))
speed:=929 acc:=529;
        move lin robtarget (pose (position (200,65,514),
orientation (0,-90,180))) speed:=629 acc:=629;
        move ptp robtarget (pose (position (200,65,515),
orientation (0,-90,180))) speed_ptp:=brzina_ptp acc_ptp:=ubrzanje_ptp;
        endseq
        endprogram;

```

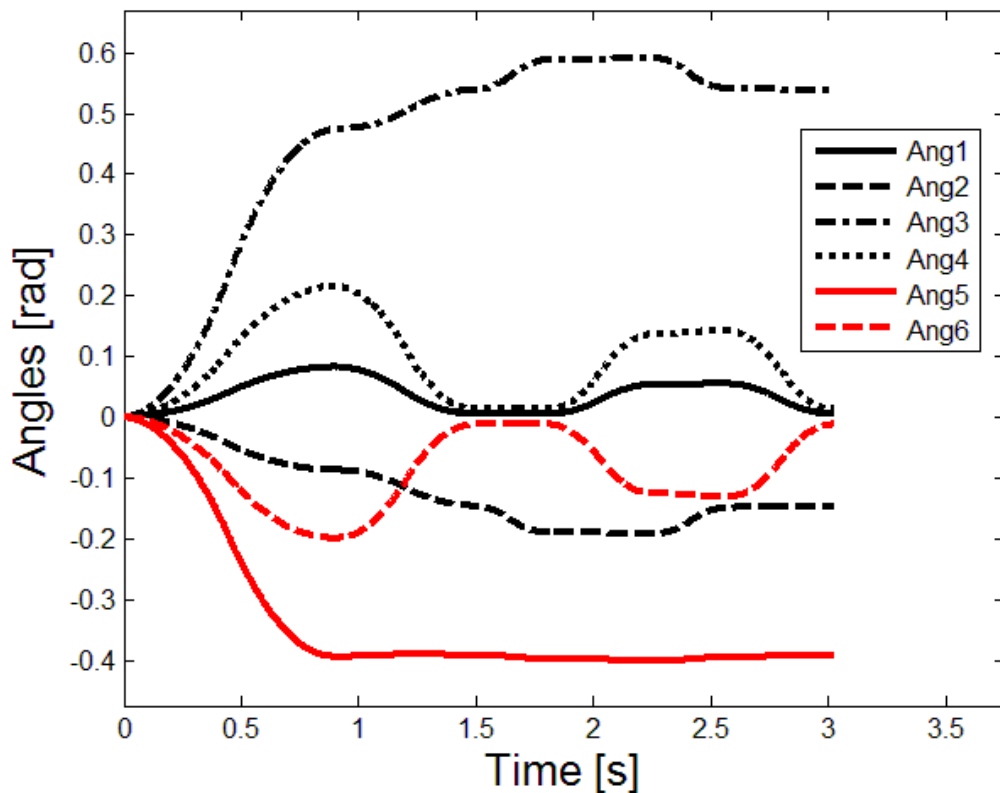
У оквиру трећег тестирања, програм са једном *PTP* и пет *LIN* инструкција под називом *Square 15* је примењен на даљинско програмирање, управљање и надзор кретања индустријског робота Лола 15. Програм је дат на слици 9.6. Извршавање програма *Square 15* је покренуто и праћено у оквиру Андроид окружења на мобилном уређају. На слици 9.7 је приказана путања енд-ефектора робота у 3Д координатном систему у оквиру окружења на мобилном Андроид уређају. Путања енд-ефектора је формирана на основу добијених вредности достигнутих позиција робота у унутрашњим координатама које су приказане на слици 9.8.



Слика 9.6 Програм *Square 15*



Слика 9.7 Путања енд-ефектора робота у току извршавања програма *Square 15* приказана у оквиру окружења на Андроид уређају



Слика 9.8 Позиције робота у унутрашњим координатама током извршавања програма *Square 15*

Програм *Square 15* је затим програмиран *L-IRL* програмским језиком и извршен применом Лола УС-а. Програм *Square 15* написан на *L-IRL* језику је дат у следећем тексту.

```

program Square_15;
system_specification "lola15.spc";
seq
  move ptp robtarget (pose(position(175,80,515),
orientation(0,-90,180))) speed_ptp:= 0.9 acc_ptp:= 0.9;
  move lin robtarget (pose(position(211,5,515),
orientation(0,-90,180))) speed:=929 acc:=929;
  move lin robtarget (pose(position(234,5,515),
orientation(0,-90,180))) speed:=929 acc:=929;
  move lin robtarget (pose(position(234,55,515),
orientation(0,-90,180))) speed:=929 acc:=929;

```

```

    move lin robtarget (pose(position(211,55,515),
orientation(0,-90,180))) speed:=929 acc:=929;
    move lin robtarget (pose(position(211,5,515),
orientation(0,-90,180))) speed:=929 acc:=929;
endseq
endprogram;

```

За сваки од тест програма (*PTP Motions 50, Combined Movement 2 и Square 15*), у току програмирања кретања мерено је колики је број акција (БА) корисника било потребно како би се написао, односно унео, комплетан програм. Мерење је обављено посебно за софтверски систем који је предложен у овој докторској дисертацији и посебно за *L-IRL* окружење. Узето је у обзир да акција корисника (А) у оквиру екрана осетљивог на додир може бити “клик” на неки од елемената графичког корисничког интерфејса (К) или унос једног карактера у текстуална поља (Т). Акција корисника код *L-IRL* окружења, код кога је интеракција са корисником базирана на употреби *PC*-ја и миша или тастатуре као периферних уређаја, може бити један клик мишем (КМ), два клика мишем (2КМ) или притиснуто дугме на тастатури (КТ).

За програмирање кретања робота са n инструкција кретања, употребом предложеног решења, укупан број акција корисника представља збир броја акција корисника потребних за програмирање појединачне инструкције кретања (уз акцију за додавање нове инструкције за сваку инструкцију осим прве), затим броја акција корисника потребних за уношење назива програм, за уношење редног броја програма, као и броја акција корисника потребних за превођење и снимање програма.

$$\text{УБАМ}(\text{Program}) = \sum_{i=0}^n \text{БА}(\text{инструкција кретања } (i)) + (n-1) * \text{БА}(\text{Add Move Command}) + \text{БА}(\text{назив}) + \text{БА}(\text{редни број}) + \text{БА}(\text{Save}) \quad (2)$$

При програмирању првог тест програма под називом *PTP Motions 50*, употребом предложеног решења, потребно је дефинисати шест *PTP* инструкција кретања. За сваку инструкцију потребно је изабрати врсту кретања и унети жељене вредности параметара. Назив робота није потребно

наводити у самој инструкцији, већ се он бира при самом уласку у режим рада за програмирање. Врста кретања се бира уз помоћ једног клика за отварање падајућег менија са понуђеним врстама кретања и уз помоћ другог клика за избор жељене врсте кретања, па су за овај део програмирања потребне две акције корисника K ($BA(\text{врста кретања})=2*K$). Иако је врста кретања *PTP* подразумевана вредност падајућег менија, узета је претпоставка да корисник увек жели да погледа све врсте кретања пре коначног избора. Након избора врсте кретања курсор се аутоматски помера на следећи елемент графичког корисничког интерфејса за програмирање чиме се омогућава специфицирање параметара изабране инструкције кретања робота. Остали параметри се специфицирају уношењем жељених вредности у одговарајућа текстуална поља и самим тим број акција корисника зависи од вредности за сваки од параметара. Пре уноса жељене вредности у свако текстуално поље, осим првог текстуалног поља инструкције, потребна је акција корисника којом се селекује жељени елемент графичког корисничког интерфејса ($BA(\text{сел}) = K$). За прво текстуално поље инструкције то није потребно због аутоматског преласка на наредни елемент графичког корисничког интерфејса након бирања врсте кретања из падајућег менија. Укупан број акција корисника за програмирање *PTP* инструкције кретања се рачуна по формули:

$$BA(PTP) = BA(\text{врста кретања}) + BA(speed) + BA(acc) + BA(target) \quad (3)$$

За прву инструкцију тест програма *PTP Motions 50* број акција корисника за специфицирање параметара кретања износи: три карактера за брзину ($BA(speed) = BA(0.2) = 3T = 3A$), три карактера и један клик за убрзање ($BA(acc) = BA(0.2) + BA(\text{сел}) = 3T + K = 4A$) и седам карактера и шест кликова за циљну тачку ($BA(target) = BA(30\ 0\ 0\ 0\ 0\ 0) + 6*BA(\text{сел}) = 7T + 6K = 13A$). Свака наредна инструкција програма *PTP Motions 50* дефинише кретање једне од оса за исту вредност у степенима, истом брзином и убрзањем, па самим тим захтева исти број акција корисника. Свака инструкција после

подразумеване прве захтева још једну додатну акцију корисника којом се кликом на дугме (*Add Move Command*) додаје инструкција кретања. Након програмирања свих инструкција кретања потребно је дефинисати назив програма и његов редни број у графичком корисничком интерфејсу режима рада брзог бирања. У овом случају број акција корисника за дефинисање назива програма износи $BA(PTP\ Motions\ 50) + BA(сел) = 14T + K$, док за дефинисање његовог редног броја износи $BA(8) + BA(сел) = T + K$. Превођење и снимање програма, како би се он касније могао извршавати у оквиру режима рада брзог бирања, захтевају додатно још једну акцију корисника, односно клик на дугме *Save* ($BA(Save) = K$). На основу претходне анализе и примене формула (2) и (3), израчунат је укупан број акција корисника УБАМ за програмирање програма *PTP Motions 50* са шест *PTP* инструкција у оквиру режима рада за програмирање предложеног софтверског система:

$$\begin{aligned} \text{УБАМ (PTP Motions 50)} &= 6*BA(PTP) + 5*BA(Add\ Move\ Command) + BA(PTP \\ &Motions\ 50) + BA(сел) + BA(8) + BA(сел) + BA(Save) = 6* (BA(врста\ кретања) + \\ &BA(0.2) + BA(0.2) + BA(сел) + BA(30\ 0\ 0\ 0\ 0\ 0)) + 6*BA(сел) + 5*BA(Add\ Move \\ &Command) + BA(PTP\ Motions\ 50) + BA(сел) + BA(8) + BA(сел) + BA(Save) = 6 * \\ &(2K + 3T + 3T + K + 7T + 6K) + 5*K + 14T + K + T + K + K = 6 * (2A + 3A + 3A + A + \\ &7A + 6A) + 5A + 14A + A + A + A + A = 155A \end{aligned}$$

При програмирању роботских програма употребом *L-IRL* окружења укупан број акција корисника УБАЛ одговара броју унетих карактера m путем тастатуре, односно укупном броју карактера програма укључујући и размак, уз додатну акцију корисника потребну за снимање програма, што је дефинисано следећом формулом:

$$\text{УБАЛ (Program)} = \sum_{i=0}^m BA(\text{број карактера}) + BA(Save) \quad (4)$$

Снимање програма се обавља кликом на одговарајућу иконицу графичког корисничког интерфејса *L-IRL* едитора и износи $BA(Save) = K$.

У случају употребе предложеног софтверског система, као и у случају употребе *L-IRL* окружења, укупан број акција се рачуна од тренутка започињања писања програма, док се број акција корисника потребних да се покрене одговарајуће окружење занемарује.

За први тест програм под називом *PTP Motions 50* укупан број акција корисника УБАЛ износи:

$$\text{УБАЛ (PTP Motions 50)} = 509\text{КТ} + \text{К} = 510\text{А}$$

При програмирању другог тест програма под називом *Combined Movement 2*, употребом софтверског система предложеног у овој докторској дисертацији, потребно је дефинисати две *PTP* инструкција кретања, две *CIRCLE* и једну *LIN* инструкцију кретања. Број акција корисника за *LIN* инструкцију кретања се рачуна на исти начин као и за *PTP* инструкцију кретања. *CIRCLE* инструкција кретања захтева већи број акција корисника због тога што ова инструкција захтева дефинисање помоћне тачке кретања (потребно је задати позицију помоћне тачке, а оријентација није потребна), па број акција корисника за програмирање ове врсте кретања износи:

$$\text{БА (CIRCLE)} = \text{БА(врста кретања)} + \text{БА(speed)} + \text{БА(acc)} + \text{БА(target)} + \text{БА(auxiliary)} \quad (5)$$

Применом формула (2), (3) и (5) израчунат је укупан број акција корисника УБАМ потребних за програмирање програма *Combined Movement 2* у режиму рада за програмирање предложеног софтверског система:

$$\begin{aligned} \text{УБАМ (Combined Movement 2)} &= \text{БА(PTP1)} + \text{БА(CIRCLE1)} + \text{БА (CIRCLE2)} \\ &+ \text{БА(LIN)} + \text{БА(PTP2)} + 4*\text{БА(Add Move Command)} + \text{БА(Combined Movement 2)} + \\ &\text{БА(сел)} + \text{БА(8)} + \text{БА(сел)} + \text{БА(Save)} = \\ &\text{БА(врста кретања)} + \text{БА(0.9)} + \text{БА(0.9)} + \text{БА(сел)} + \text{БА(175 80 515 0 270} \\ &180) + 6*\text{БА(сел)} + //\text{PTP1} \end{aligned}$$

БА(врста кретања) + БА(929) + БА(529) + БА(сел) + БА(211 25 515 0 270 180) + 6*БА(сел) + БА(196 47 515) + 3*БА(сел) + //CIRCLE1

БА(врста кретања) + БА(929) + БА(529) + БА(сел) + БА(244 25 515 0 270 180) + 6*БА(сел) + БА(196 7 514) + 3*БА(сел) + // CIRCLE2

БА(врста кретања) + БА(629) + БА(629) + БА(сел) + БА(200 65 514 0 270 180) + 6*БА(сел) + //LIN

БА(врста кретања) + БА(0.9) + БА(0.9) + БА(сел) + БА(200 65 515 0 270 180) + 6*БА(сел) + //PTP2

4*БА(*Add Move Command*) + БА(*Combined Movement 2*) + БА(сел) + БА(8) + БА(сел) + БА(*Save*)=

2К+ 3Т + 3Т + К + 15Т + 6К + //PTP1

2К+ 3Т + 3Т + К + 23Т + 9К + //CIRCLE1

2К+ 3Т + 3Т + К + 22Т + 9К + // CIRCLE2

2К+ 3Т + 3Т + К + 15Т + 6К + //LIN

2К+ 3Т + 3Т + К + 15Т + 6К + //PTP2

+ 4*К + 19Т + К + Т + К + К =

2А+ 3А + 3А + А + 15А + 6А + //PTP1

2А+ 3А + 3А + А + 23А + 9А + //CIRCLE1

2А+ 3А + 3А + А + 22А + 9А + // CIRCLE2

2А+ 3А + 3А + А + 15А + 6А + //LIN

2А+ 3А + 3А + А + 15А + 6А + //PTP2

+ 4*А + 19А + А + А + А + А = 198А

При програмирању другог тест програма под називом *Combined Movement 2* употребом *L-IRL* окружења укупан број акција корисника УБАЛ је израчунат применом формуле (4) и износи:

УБАЛ (*Combined Movement 2*)= 718КТ + К = 719А

Применом формула (2) и (3) израчунат је укупан број акција корисника УБАМ потребних за програмирање трећег тест програма *Square 15* у режиму рада за програмирање предложеног софтверског система:

УБАМ (*Square 15*) = БА(*PTP*) + БА(*LIN1*) + БА(*LIN2*) + БА(*LIN3*) +
 БА(*LIN4*) + БА(*LIN5*) + 5*БА(*Add Move Command*) + БА (*Square 15*) + БА(сел) +
 БА(4) + БА(сел) + БА(*Save*)=

БА(врста кретања) + БА(0.9) + БА(0.9) + БА(сел) + БА(175 80 515 0 270
 180) + 6*БА(сел) + //PTP

БА(врста кретања) + БА(929) + БА(929) + БА(сел) + БА(211 5 515 0 270
 180) + 6*БА(сел) + //LIN1

БА(врста кретања) + БА(929) + БА(929) + БА(сел) + БА(234 5 515 0 270
 180) + 6*БА(сел) + //LIN2

БА(врста кретања) + БА(929) + БА(929) + БА(сел) + БА(234 55 515 0
 270 180) + 6*БА(сел) + //LIN3

БА(врста кретања) + БА(929) + БА(929) + БА(сел) + БА(211 55 515 0
 270 180) + 6*БА(сел) + //LIN4

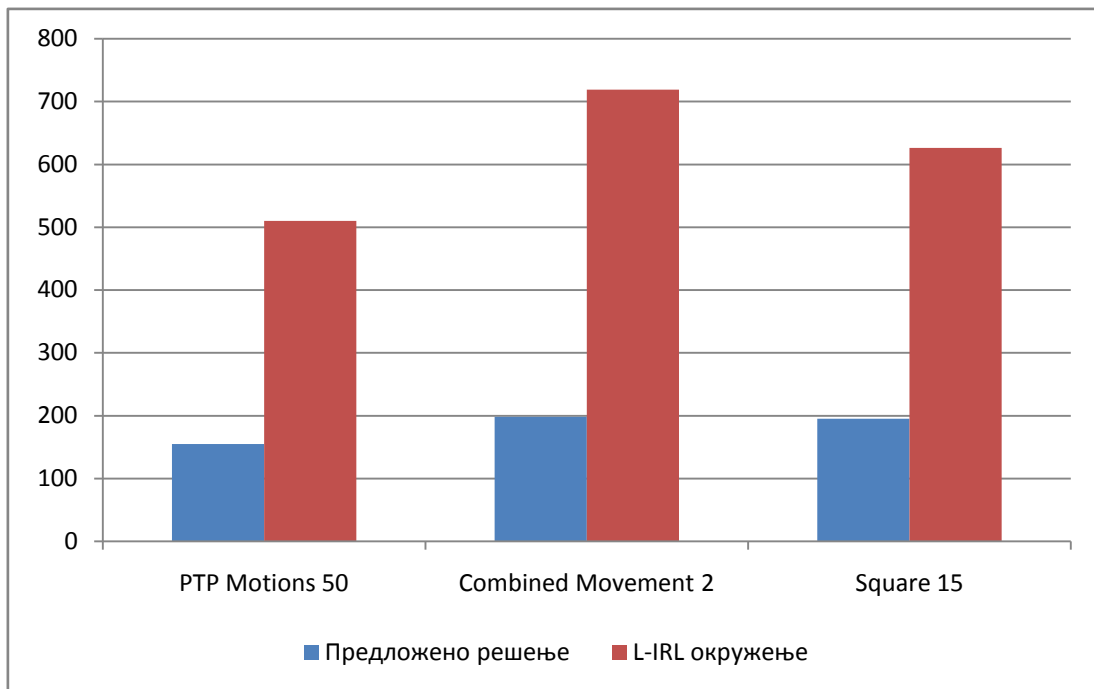
БА(врста кретања) + БА(929) + БА(929) + БА(сел) + БА(211 5 515 0 270
 180) + 6*БА(сел) + //LIN5

+ 5*БА(*Add Move Command*) + БА(*Square 15*) + БА(сел) + БА(4) + БА(сел)
 + БА(*Save*)=

$$\begin{aligned}
 &2K + 3T + 3T + K + 15T + 6K + \\
 &2K + 3T + 3T + K + 14T + 6K + \\
 &2K + 3T + 3T + K + 14T + 6K + \\
 &2K + 3T + 3T + K + 15T + 6K + \\
 &2K + 3T + 3T + K + 15T + 6K + \\
 &2K + 3T + 3T + K + 14T + 6K + \\
 &+ 5*K + 9T + K + T + K + K = 195A
 \end{aligned}$$

При програмирању *Square 15* програма употребом *L-IRL* окружења
 укупан број акција корисника УБАЛ који је израчунат применом формуле (4)
 износи:

$$\text{УБАЛ (Square 15)} = 625KT + K = 626A$$



Слика 9.9 Укупан број акција корисника (A) приликом програмирања тест програма *PTP Motions 50*, *Combined Movement 2* и *Square 15* применом предложеног решења (плава боја) и *L-IRL* окружења (црвена боја)

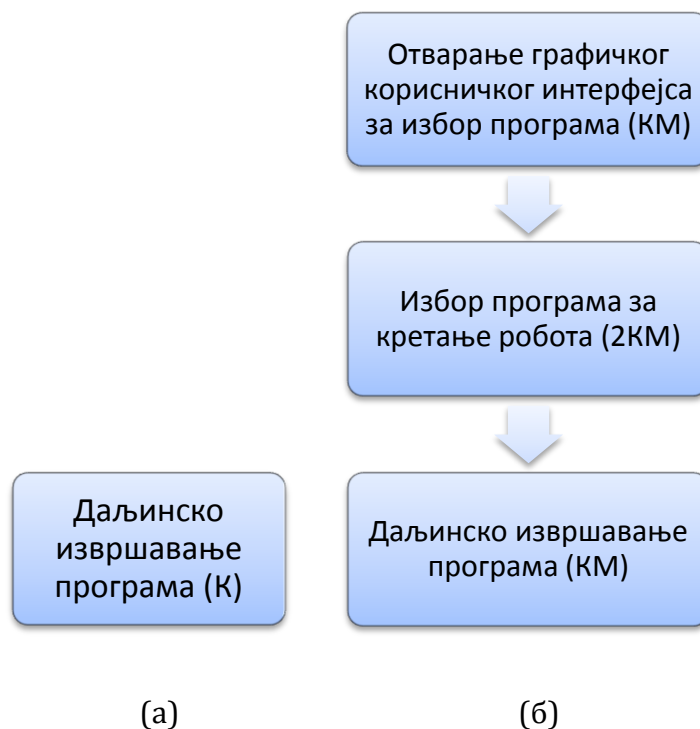
На слици 9.9 је упоредо приказано колико је акција потребно спровести приликом програмирања сваког од тест програма (*PTP Motions 50*, *Combined Movement 2* и *Square 15*) применом софтверског система предложеног у овој докторској дисертацији (плава боја) и *L-IRL* окружења (црвена боја).



Слика 9.10 Кораци које је потребно извршити приликом даљинског покретања претходно дефинисаног програма (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола УС-а базираног на употреби РС-ја за интеракцију са корисником, почевши од покретања окружења

Затим је измерен број акција корисника потребних за покретање претходно дефинисаног програма *PTP Motions 50* употребом режима рада брзог бирања предложеног софтверског система и употребом Лола УС-а базираног на употреби РС-ја за интеракцију са корисником. Слика 9.10 описује кораке које је потребно извршити приликом даљинског покретања претходно дефинисаног програма (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола УС-а базираног на употреби РС-ја за интеракцију са корисником, почевши од покретања окружења. Број корака

одговара броју акција корисника. Претпоставља се да се и у једном и у другом случају иконица за покретање окружења налази на главном екрану. Такође се у оба случаја повезивање на сервер обавља аутоматски приликом избора робота. За сваки од корака означена је врста акције корисника.

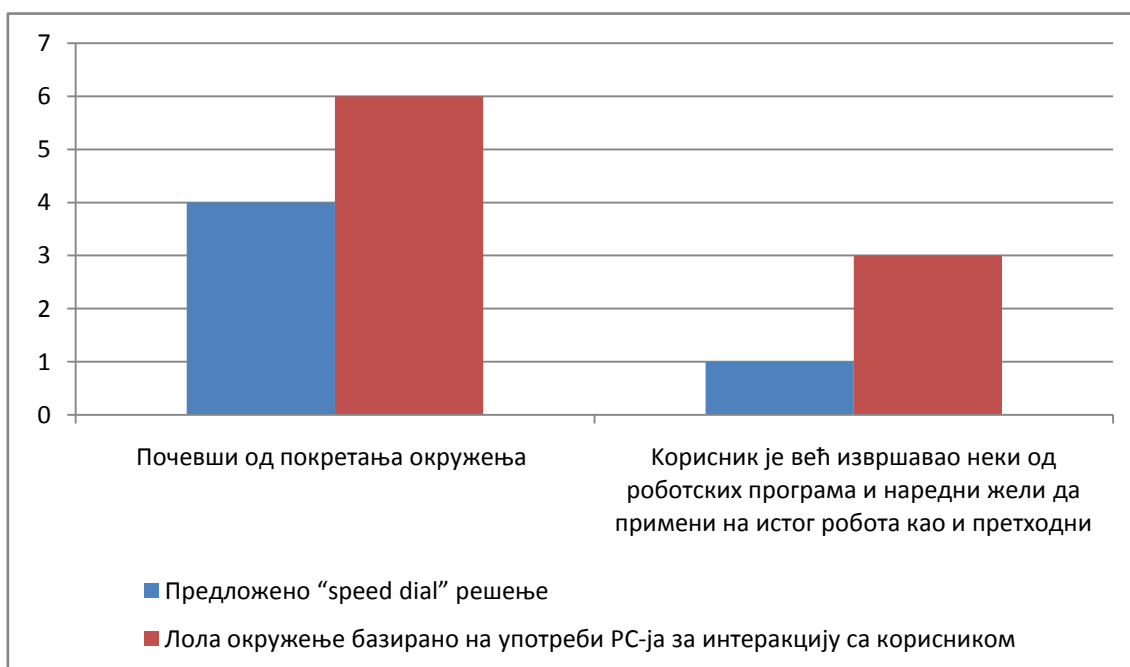


Слика 9.11 Кораци које је потребно извршити приликом даљинског покретања претходно дефинисаног програма (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола окружења базираног на употреби РС -ја за интеракцију са корисником, у случају када је окружење већ покренуто и када је корисник претходно већ извршавао неки од програма за кретање оног робота за кога жели да изврши и наредни програм

Слика 9.11 описује кораке које је потребно извршити приликом покретања претходно дефинисаног програма (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола окружења базираног на употреби РС -ја за интеракцију са корисником, у случају када је окружење већ покренуто и када је корисник претходно већ извршавао неки од програма за кретање оног робота за кога жели да изврши и наредни програм. Број корака

одговара броју акција корисника. За сваки од корака означена је врста акције корисника.

На слици 9.12 је графички је приказано колико је акција потребно спровести приликом даљинског покретања претходно дефинисаног програма применом предложеног “*speed dial*” решења (плава боја) и Лола окружења базираног на употреби *PC*-ја за интеракцију са корисником (црвена боја) почевши од покретања окружења, као и за случај када је корисник већ извршавао неки од роботских програма и наредни жели да примени на истог робота као и претходни.



Слика 9.12 Број акција корисника (А) приликом даљинског покретања претходно дефинисаног програма применом предложеног “*speed dial*” решења (плава боја) и Лола окружења базираног на употреби *PC*-ја за интеракцију са корисником (црвена боја) почевши од покретања окружења (лево) и за случај када је корисник већ извршавао неки од роботских програма и наредни жели да примени на истог робота као и претходни (десно)

Резултати поређења су показали да је предложено софтверско окружење утицало на редуковање броја акција корисника потребних за

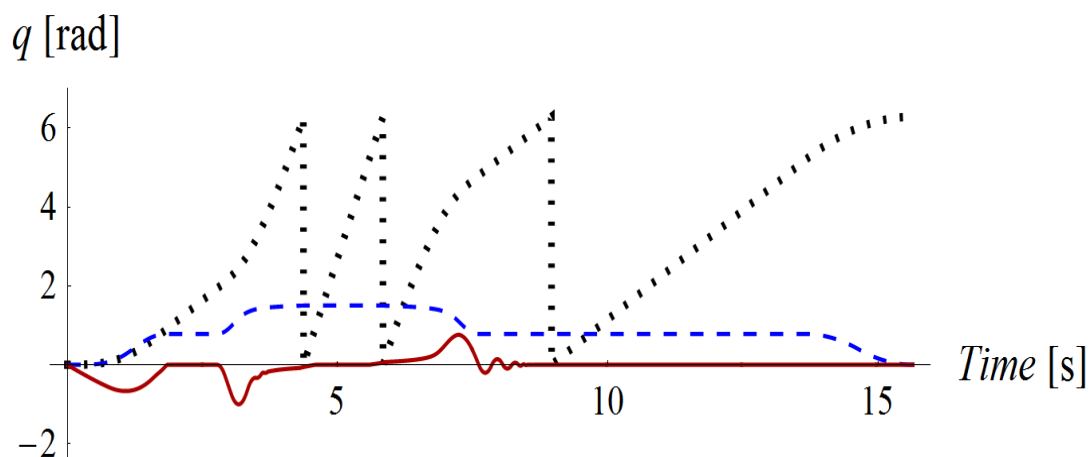
програмирање новог и покретање постојећег програма за кретање индустријских робота. Значајно редуковање броја акција корисника приликом програмирања новог програма за кретање индустријског робота је постигнуто употребом предложеног режима рада за програмирање и та вредност износи нешто више од 30 процената од вредности броја акција које је потребно спровести употребом *L-IRL* окружења за први тест програм (30,39 %) и трећи тест програм (31,15 %), а нешто испод 30 процената за други тест програм (27,54 %). Максимално редуковање броја акција корисника потребних за покретање извршавања претходно дефинисаног програма у случају када је окружење већ покренуто и када је корисник претходно извршавао неки од роботских програма за истог робота за кога жели да изврши и наредни програм, постигнуто је предложеним режимом рада брзог бирања ("*speed dial*"), чиме је број акција корисника сведен на само једну потребну акцију.

9.2 Симулација кретања уређаја за тренинг пилота и симулацију лета

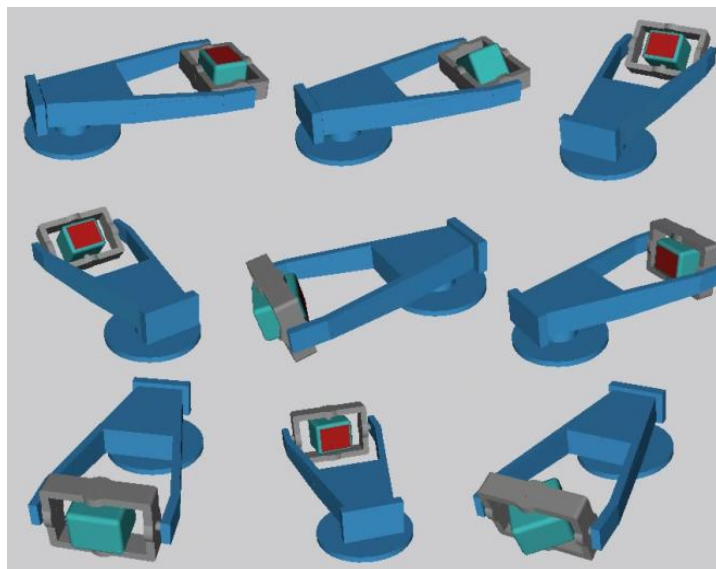
Софтверски систем који је предложен у овој докторској дисертацији примењен је за програмирање и симулацију кретања центрифуге моделиране као манипулатор са три степена слободе и уређаја за просторну дезоријентацију са четири степена слободе.

Програмирање новог програма за кретање центрифуге је извршено употребом графичког корисничког интерфејса режима рада за програмирање. Програм је дат на слици 8.3. Након успешног превођења, програм је повезан са елементом под редним бројем 4 графичког корисничког интерфејса за режим рада брзог бирања и именован *Gmove Program 2*. У току превођења програма, применом алгоритама којима се дефинише кинематика и алгоритми кретања робота [72], [139], [140], [141], генерисане су вредности углова за сваку од три осе робота у току кретања (Слика 9.13). Те вредности се користе за симулацију кретања виртуелног

модела центрифуге. Кликом на елемент графичког корисничког интерфејса режима рада брзог бирања под називом *Gmove Program 2* програм је покренут. Извршавање је праћено кроз 3Д виртуелни модел центрифуге. Неколико позиција током кретања виртуелног модела центрифуге при симулацији је приказано на слици 9.14.



Слика 9.13 Вредности углова у радијанима за сваку од три осе робота у току симулације програма *Gmove Program 2* (тачкаста линија - прва оса, испрекидана линија - друга оса, пуна линија - трећа оса)

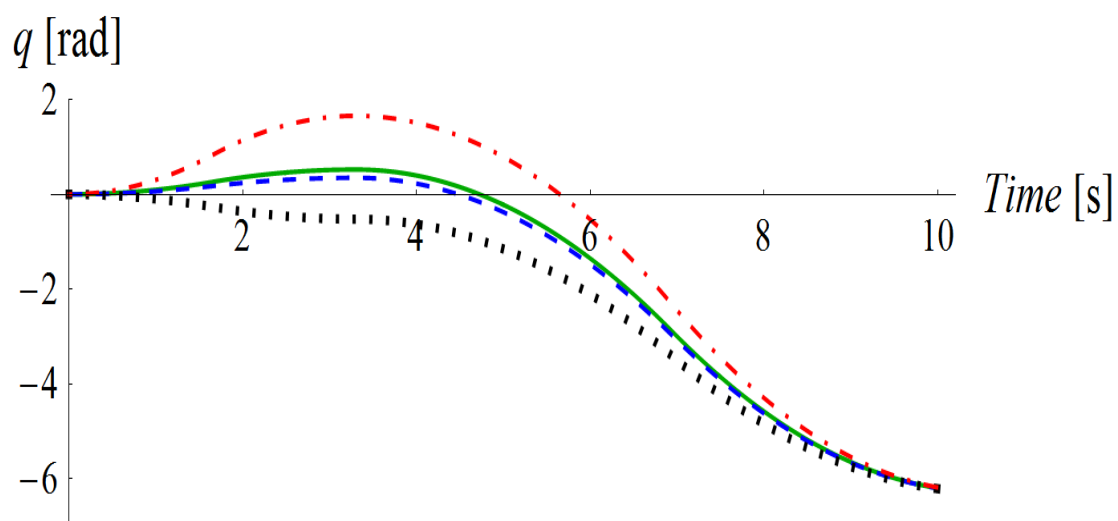


Слика 9.14 Неколико позиција виртуелног модела центрифуге током симулације програма *Gmove Program 2*

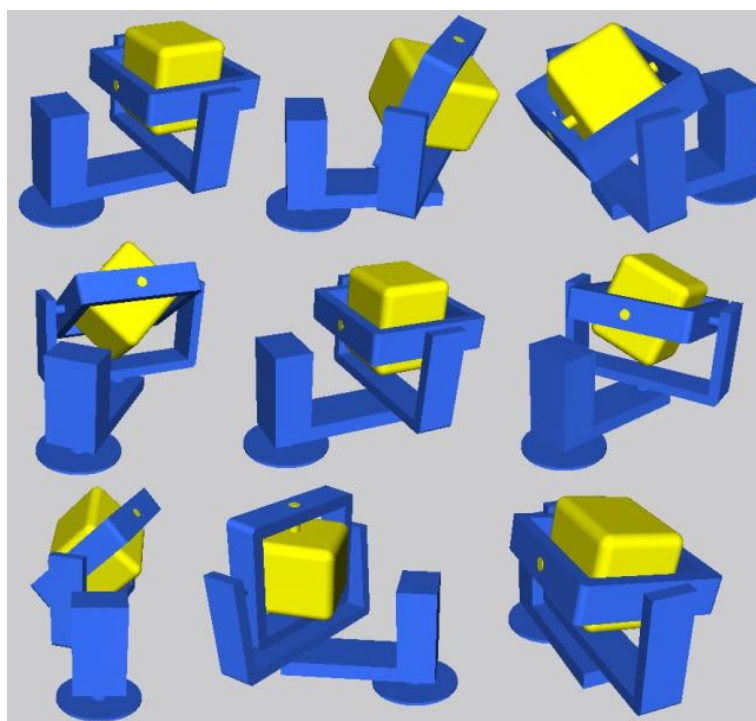
Програм *Gmove Program 2* је затим програмиран *L-IRL* програмским језиком и његово извршавање је симулирано применом Лола УС-а који је описан у поглављу 3.2. Програм *Gmove Program 2* написан на *L-IRL* језику је дат у следећем тексту.

```
program Gmove_program_2;
system_specification "centrifuge";
seq
gmove time:= 1.0 bl_g:= 1.41;
gmove acc_g:= 9.0 bl_g:= 15.0;
gmove time:= 1.0 bl_g:= 15.0;
gmove acc_g:= -9.0 bl_g:= 1.41;
gmove time:= 1.0 bl_g:= 1.41;
endseq
endprogram;
```

Затим је дефинисан програм за кретање уређаја за просторну дезоријентацију пилота који је моделован као манипулатор са четири степена слободе. Програм је дат на слици 8.4. Након успешног превођења програм је повезан са одговарајућим елементом графичког корисничког интерфејса режима рада брзог бирања и именом *DoTest*. У току превођења програма, применом одговарајућих алгоритама којима се дефинише кретање робота [74], генерисане су вредности углова за сваку од четири осе робота у току кретања (Слика 9.15). Неколико позиција током кретања виртуелног модела уређаја за просторну дезоријентацију пилота при симулацији је приказано на слици 9.16.



Слика 9.15. Вредности углова у радијанима за сваку од четири осе дезоринетатора у току симулације програма *DoTest 2* (пуна линија - прва оса, испрекидана линија - друга оса, тачкаста линија - трећа оса, линија црта-тачка - четврта оса)



Слика 9.16 Неколико позиција виртуелног модела уређаја за просторну дезоријентацију пилота током симулације програма *DoTest*

Програм *DoTest* је затим написан употребом *L-IRL* програмског језика и симулиран применом Лола УС-а базираног на употреби РС-ја за интеракцију са корисником. Програм *DoTest* написан на *L-IRL* језику је дат у следећем тексту.

```
program DoTest;
system_specification "DO.spc";
seq
move ptp joint(mainjoint(30., 20., -30., 95.)) speed_ptp :=0.4
acc_ptp := 0.2;
move ptp joint(mainjoint(-360., -360., -360., -360.)) speed_ptp
:=0.2 acc_ptp := 0.2;
endseq
endprogram;
```

За тест програме *Gmove Program 2* и *DoTest* мерено је колико је акција корисника потребно како би се написао, односно унео, комплетан програм употребом софтверског решења предложеног у овој докторској дисертацији и *L-IRL* окружења. Параметри кретања су у оба случаја дефинисани као реални бројеви. Узето је у обзир да акција корисника (А) у оквиру екрана осетљивог на додир може бити “клик” на неки од елемената графичког корисничког интерфејса (К) или унос једног карактера у текстуална поља (Т). Акција корисника код *L-IRL* окружења, код кога је интеракција са корисником базирана на употреби РС-ја и миша или тастатуре као периферних уређаја, може бити један клик мишем (КМ), два клика мишем (2КМ) или притиснуто дугме на тастатури (КТ).

Број акција корисника за програмирање *GMOVE* инструкције кретања се рачуна по формули:

$$\begin{aligned} \text{БА (GMOVE)} &= \text{БА(избор параметра)} + (\text{БА(time)} \mid \text{БА(acc_g)}) \\ &+ \text{БА(bl_g)} \end{aligned} \tag{6}$$

У овом случају постоји само једна врста кретања, па избор врсте кретања није потребан. Како је поред обавезног параметра *blg* потребно навести параметар *time* или параметар *acc_g*, у оквиру падајућег менија врши се избор једног од ова два параметра. Подразумевана вредност првог параметра *GMOVE* инструкције кретања је *time*, па је додатна акција корисника за избор параметра потребна само у случају избора параметра *acc_g*. На почетку дефинисања програма курсор је постављен на прво текстуално поље инструкције кретања. У случају избора параметра курсор се аутоматски помера на следећи елемент графичког корисничког интерфејса, па је додатна акција корисника за селектовање текстуалног поља потребна само за параметар *blg*.

Применом формула (2) и (6) израчунат је укупан број акција корисника УБАМ потребних за дефинисање програма *Gmove Program 2*:

$$\begin{aligned}
 \text{УБАМ (Gmove Program 2)} &= \text{БА(Gmove1)} + \text{БА(Gmove2)} + \text{БА(Gmove3)} + \\
 &\text{БА(Gmove4)} + \text{БА(Gmove5)} + 4*\text{БА(Add Move Command)} + \text{БА(Gmove Program 2)} + \\
 &\text{БА(сел)} + \text{БА(4)} + \text{БА(сел)} + \text{БА(Save)} = \\
 &\text{БА(time)} + \text{БА(bl_g)} + \\
 &\text{БА(избор параметра)} + \text{БА(acc_g)} + \text{БА(bl_g)} + \\
 &\text{БА(time)} + \text{БА(bl_g)} + \\
 &\text{БА(избор параметра)} + \text{БА(acc_g)} + \text{БА(bl_g)} + \\
 &\text{БА(time)} + \text{БА(bl_g)} + \\
 &4*\text{БА(Add Move Command)} + \text{БА (Gmove Program 2)} + \text{БА(сел)} + \text{БА(4)} + \\
 &\text{БА(сел)} + \text{БА(Save)} = \\
 &\text{БА(1.0)} + \text{БА(1.41)} + \text{БА(сел)} + // \text{Gmove1} \\
 &2*\text{К} + \text{БА(9.0)} + \text{БА(15.0)} + \text{БА(сел)} + // \text{Gmove2} \\
 &\text{БА(1.0)} + \text{БА(15.0)} + \text{БА(сел)} + // \text{Gmove3} \\
 &2*\text{К} + \text{БА(-9.0)} + \text{БА(1.41)} + \text{БА(сел)} + // \text{Gmove4} \\
 &\text{БА(1.0)} + \text{БА(1.41)} + \text{БА(сел)} + // \text{Gmove5} \\
 &4*\text{БА(Add Move Command)} + \text{БА (Gmove Program 2)} + \text{БА(сел)} + \text{БА(4)} + \\
 &\text{БА(сел)} + \text{БА(Save)} =
 \end{aligned}$$

$$3T + 4T + K + 2K + 3T + 4T + K + 3T + 4T + K + 2K + 4T + 4T + K + 3T + 4T + K + 4K + 15T + K + T + K + K = 68A$$

Применом формуле (4) израчунат је укупан број акција корисника при програмирању програма *Gmove Program 2* употребом *L-IRL* окружења (УБАЛ) који износи:

$$\text{УБАЛ (Gmove Program 2)} = 226KT + K = 227A$$

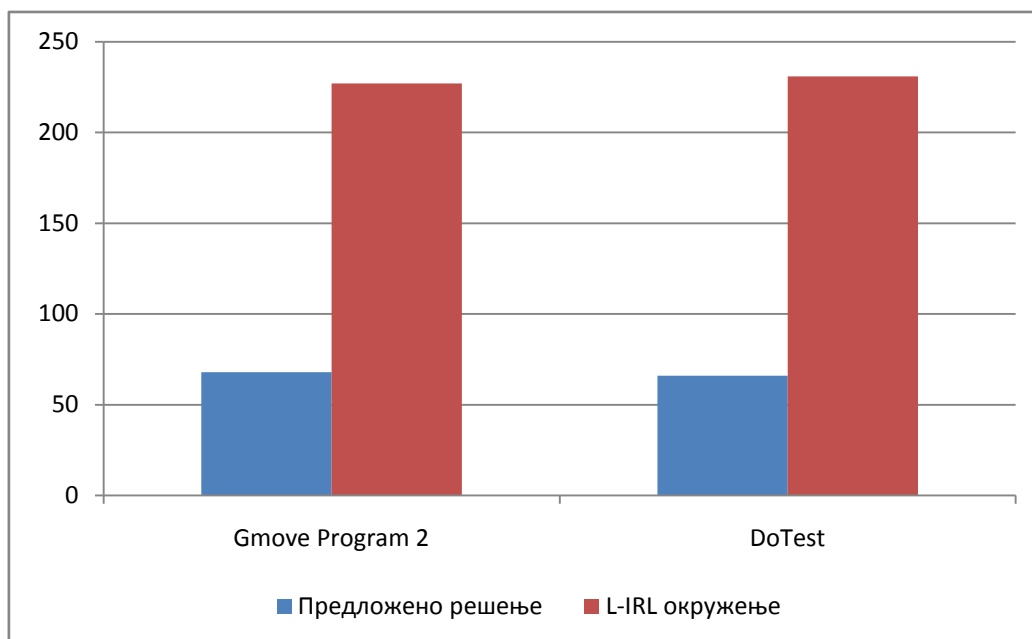
Затим је применом формула (2) и (3) израчунат укупан број акција корисника УБАМ потребних за програмирање тест програма *DoTest* у режиму рада за програмирање предложеног софтверског система. Како је уређај за просторну дезоријентацију пилота моделован као манипулатор са четири степена слободе, циљна позиција у обе инструкције кретања се програмира са четири реалне вредности. *PTP* врста кретања је једина дефинисана врста кретања овог робота, па се при програмирању не мора специфицирати већ представља подразумевану вредност.

$$\begin{aligned} \text{УБАМ (DoTest)} &= \text{БА(PTP1)} + \text{БА(PTP2)} + \text{БА(Add Move Command)} + \text{БА} \\ &(\text{DoTest}) + \text{БА(сел)} + \text{БА(4)} + \text{БА(сел)} + \text{БА(Save)} = \\ &\text{БА(0.4)} + \text{БА(0.2)} + \text{БА(сел)} + \text{БА(30. 20. -30. 95.)} + 4 * \text{БА(сел)} + //PTP1 \\ &\text{БА(0.2)} + \text{БА(0.2)} + \text{БА(сел)} + \text{БА(-360. -360. -360. -360.)} + 4 * \text{БА(сел)} + \\ &//PTP2 \\ &\text{БА(Add Move Command)} + \text{БА (DoTest)} + \text{БА(сел)} + \text{БА(4)} + \text{БА(сел)} + \\ &\text{БА(Save)} = \\ &3T + 3T + K + 13T + 4K + 3T + 3T + K + 20T + 4K + \\ &+ K + 6T + K + T + K + K = 66A \end{aligned}$$

Укупан број акција корисника при програмирању програма *DoTest* употребом *L-IRL* окружења (УБАЛ) износи:

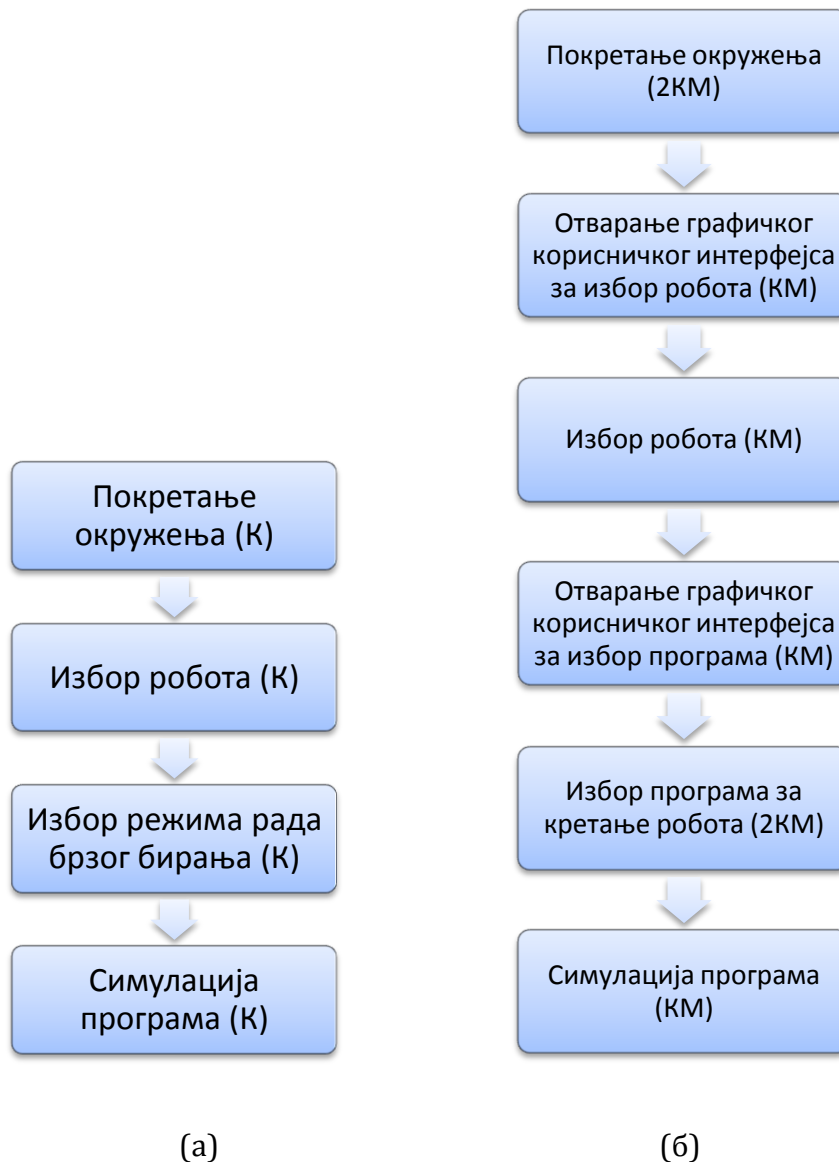
$$\text{УБАЛ (DoTest)} = 230\text{КТ} + \text{К} = 231\text{А}$$

На слици 9.17 упоредо је приказано колико је акција корисника потребно приликом програмирања четвртог и петог тест програма за симулацију (*Gmove Program 2* и *DoTest*) применом софтверског решења предложеног у овој докторској дисертацији (плава боја) и *L-IRL* окружења (црвена боја).



Слика 9.17 Број акција корисника потребних за програмирање тест програма за симулацију (*Gmove Program 2* и *DoTest*) применом предложеног решења (плава боја) и *L-IRL* окружења (црвена боја)

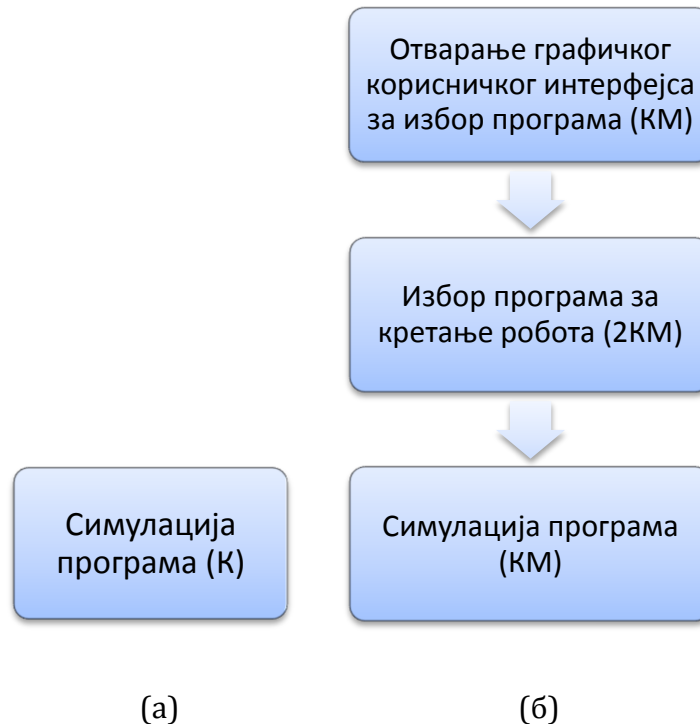
Затим је измерено колико је акција потребно спровести како би се симулирао програм за кретање робота (Слика 9.18) (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола УС-а базираног на употреби РС-ја за интеракцију са корисником, почевши од покретања окружења. Број корака одговара броју акција корисника. Претпоставља се да се и у једном и у другом случају иконица за покретање окружења налази на главном екрану. За сваки од корака означена је врста акције корисника.



Слика 9.18 Кораци које је потребно извршити приликом симулације програма за кретање робота (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола УС-а базираног на употреби РС -ја за интеракцију са корисником, почевши од покретања окружења

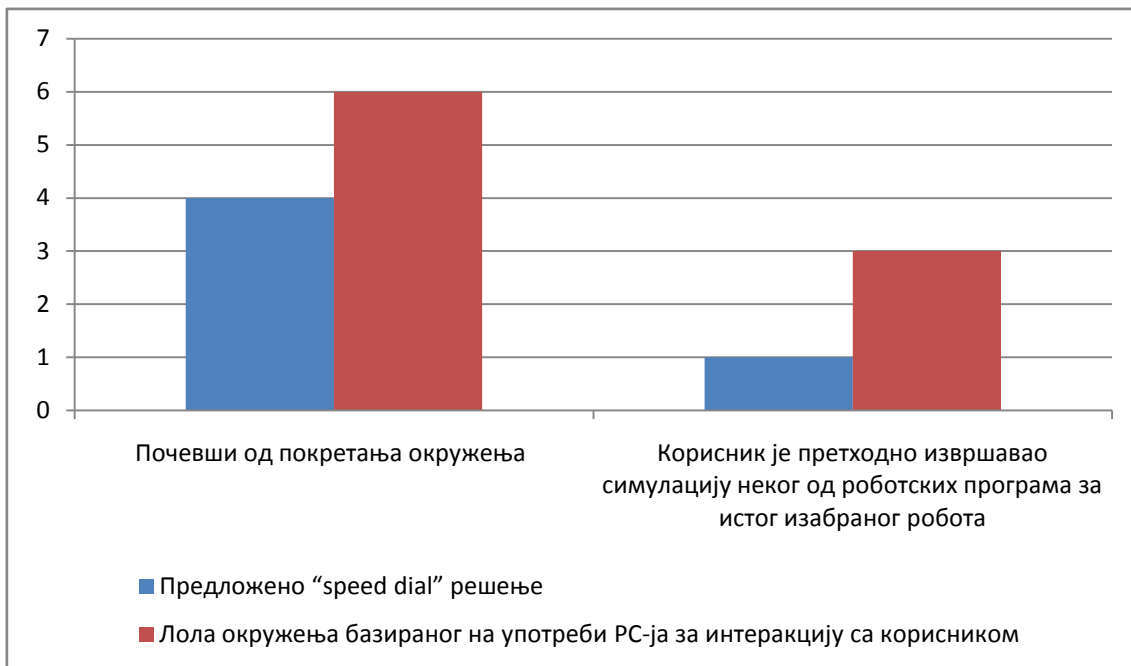
Слика 9.19 описује кораке које је потребно извршити приликом симулације дефинисаног програма (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола УС-а базираног на употреби РС-ја за интеракцију са корисником, у случају када је окружење већ покренуто и када је корисник претходно извршавао симулацију за неки од роботских

програма, а наредни програм жели да симулира користећи истог робота као и у претходном случају.



Слика 9.19 Кораци које је потребно извршити приликом симулације програма за кретање робота (а) употребом предложеног “*speed dial*” решења и (б) употребом Лола УС-а базираног на употреби РС -ја за интеракцију са корисником за случај када је окружење већ покренуто и када је корисник извршавао симулацију за неки од роботских програма, а наредни програм жели да симулира користећи истог робота као и у претходном случају

На слици 9.20 је графички је приказано колико је акција потребно спровести приликом симулације дефинисаног програма за кретање робота применом предложеног “*speed dial*” решења (плава боја) и Лола УС-а базираног на употреби РС-ја за интеракцију са корисником (црвена боја) почевши од покретања окружења, као и за случај када је корисник претходно симулирао извршавање неког од роботских програма за истог изабраног робота.



Слика 9.20 Број акција корисника (А) потребних за извршавање симулације роботског програма применом предложеног "speed dial" решења (плава боја) и Лола УС-а базираног на употреби РС-ја за интеракцију са корисником (црвена боја) почевши од покретања окружења и за случај када је корисник претходно симулирао извршавање неког од роботских програма за истог изабраног робота

Подаци са Слика 9.17 и 9.20 су показали да је предложено софтверско окружење утицало на редуковање броја акција корисника потребних за програмирање новог и симулацију постојећег програма за кретање уређаја за тренинг пилота и симулацију лета. Значајно редуковање броја акција корисника приликом програмирања новог програма за кретање уређаја за тренинг пилота и симулацију лета је постигнуто употребом предложеног режима рада за програмирање. Број акција корисника потребних за програмирање кретања центрифуге применом предложеног решења износи нешто испод 30 процената (29,95 %) од броја акција потребних за програмирање кретања центрифуге применом *L-IRL* окружења и програмског језика, док за уређај за просторну дезоријентацију пилота износи нешто више од 28 процената (28,57 %). Максимално редуковање

броја акција корисника потребних за покретање симулације претходно дефинисаног програма у случају када је окружење већ покренуто и када је корисник претходно извршавао симулацију за неки од роботских програма користећи истог робота, постигнуто је предложеним режимом рада брзог бирања “*speed dial*”, чиме је број акција корисника сведен на једну потребну акцију.

10. ПРИМЕНА И ЕВАЛУАЦИЈА ОКРУЖЕЊА *ROBOTICS FIRST* У ОБРАЗОВНОМ ПРОЦЕСУ

Предложени софтверски систем је примењен на практичној настави у Лола институту и у оквиру тога је обављено истраживање које оцењује искуство ученика средњих школа при употреби окружења *Robotics First* у режиму рада за програмирање робота и режиму рада брзог бирања при симулацији рада индустријских робота. У овом делу истраживања учествовало је 24 ученика електротехничке средње школе “Стари град”, из Београда, смер аутоматика. Десет ученика су били ученици трећег разреда средње школе, а 14 ученика су били ученици четвртог разреда, од тога једна особа женског пола и 23 особе мушког пола. Квалитативно истраживање у оквиру практичне наставе је обављено у трајању од две недеље, у мају и јуну 2016. године. На почетку истраживања ученицима је приказана демонстрација рада *L-IRL* окружења, као и демонстрација рада окружења *Robotics First* у режиму рада за програмирање, режиму рада за симулацију и режиму рада брзог бирања. Након тога обављено је истраживање о ефикасности и употребљивости предложеног окружења *Robotics First* у едукацији у роботизици. Ученицима је дато да реше шест различитих тест задатака у оквиру којих је било потребно да програмирају *PTP* кретања индустријских робота. У прва три задатка било је потребно користити програмски језик, а у друга три методу програмирања предложену у овој докторској дисертацији. Циљ овог дела истраживања је упоређивање успешности ученика у писању програма за кретање робота при употреби програмског језика и при употреби методе и окружења предложених у овој докторској дисертацији. Након решавања задатака ученици су имали могућност да примене дефинисане програме у режиму рада за симулацију. На

крају је ученицима дата могућност да оцене функционалности *Robotics First* окружења, као и да га упореде са решењима код којих се интеракција са корисником базира на употреби *PC*-ја и решењима код којих се интеракција са корисником заснива на употреби интернет претраживача. У овом делу истраживања, које укључује седам тврдњи, коришћена је Ликертова скала са вредностима од “1”, што означава одговор “уопште се не слажем”, до “5”, што означава одговор “потпуно се слажем”. Ликертова скала је коришћена како би се измерио ниво слагања учесника са наведеним тврдњама. За обављена истраживања израчунат је *Cronbach*-ов коефицијент поузданости (*Cronbach α*) [22] са циљем да се тестира поузданост добијених резултата. Резултати истраживања описани су у поглављу 10.2.

10.1 Примена окружења *Robotics First*

Прва три тест задатка, која су дата ученицима на решавање, су захтевала употребу *L-IRL* програмског језика за програмирање *PTP* кретања индустријских робота. Пре почетка теста ученицима је одржана презентација у којој су им објашњени типови података и врсте инструкција кретања *L-IRL* језика и у којој им је демонстриран рад *L-IRL* окружења. Текстови прва три тест задатка су дати у наставку:

Задатак 1.

Написати *PTP* инструкцију на *L-IRL* језику којом се прва оса робота Лола 50 помера за 30 степени (користити унутрашње координате).

*водити рачуна о ограничењима брзине и убрзања

Задатак 2.

Написати *PTP* инструкцију на *L-IRL* језику којом се прва оса робота Лола 50 помера за 30 степени (користити унутрашње координате).

*водити рачуна о ограничењима брзине и убрзања

*по угледу на приказани пример

Задатак 3.

Написати три *PTP* инструкције на *L-IRL* језику којима се прва оса робота Лола 50 помера за 30 степени, затим друга оса робота помера за 30 степени и на крају трећа оса робота помера за 60 степени (користити унутрашње координате).

*водити рачуна о ограничењима брзине и убрзања

*по угледу на приказани пример

У првом и другом задатку се захтевало од ученика да напишу исту *PTP* инструкцију кретања са том разликом што при решавању првог задатка нису имали увид у пример *PTP* инструкције кретања написане на *L-IRL* језику, док им је при решавању другог задатка дата могућност увида у један пример такве инструкције. Пример инструкције која је дата на увид ученицима је дата у следећем тексту:

```
move ptp joint (mainjoint(0, 0, 0, 90, 0, 0)) speed_ptp := 0.1
acc_ptp := 0.1 act_rob:="lola_50";
```

Трећи задатак се састојао од три *PTP* инструкције кретања којима је било потребно реализовати кретање за три различите осе робота. При решавању трећег задатка ученици су имали могућност увида у пример *PTP* инструкције кретања.

Наредна три тест задатка, која су дата ученицима на решавање, су захтевала употребу предложеног окружења *Robotics First* за програмирање *PTP* кретања индустријских робота. Пре почетка теста ученицима је одржана презентација у којој им је демонстриран рад *Robotics First* окружења. Текстови четвртог, петог и шестог тест задатака су дати у наставку:

Задатак 4.

Употребом *Robotics First* окружења формирати *PTP* инструкцију којом се прва оса робота Лола 50 помера за 30 степени.

*водити рачуна о ограничењима брзине и убрзања

Задатак 5.

Употребом *Robotics First* окружења формирати *PTP* инструкцију којом се прва оса робота Лола 50 помера за 30 степени.

*водити рачуна о ограничењима брзине и убрзања

*по угледу на приказани пример

Задатак 6.

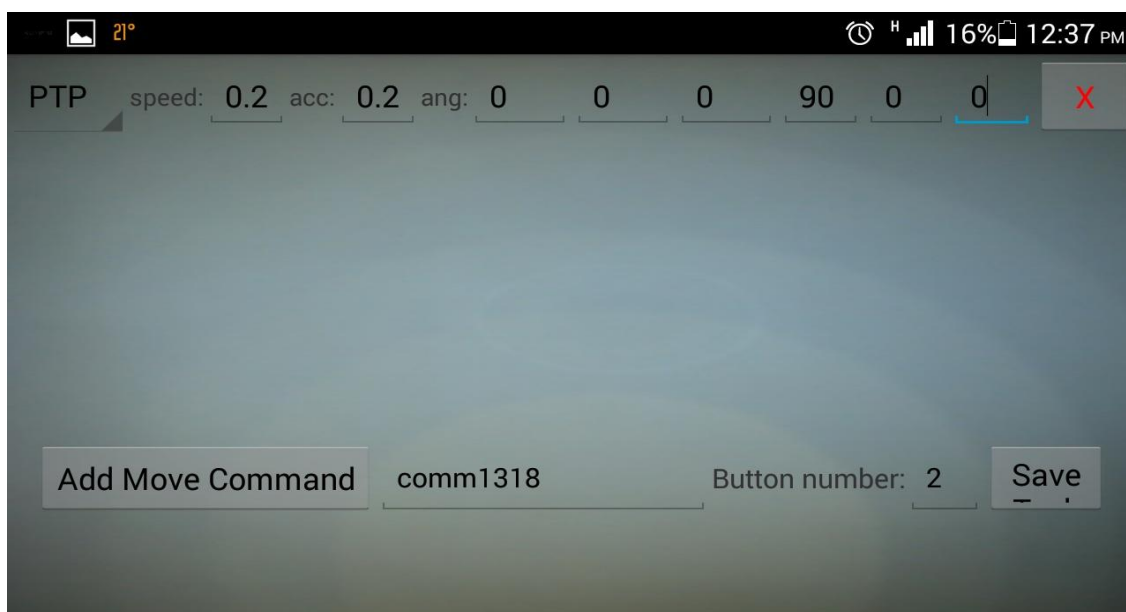
Употребом *Robotics First* окружења формирати три *PTP* инструкције којима се прва оса робота Лола 50 помера за 30 степени, затим друга оса робота помера за 30 степени и на крају трећа оса робота помера за 60 степени.

*водити рачуна о ограничењима брзине и убрзања

*по угледу на приказани пример

У четвртном и петом задатку се захтевало од ученика да напишу исту *PTP* инструкцију кретања са том разликом што при решавању четвртог задатка нису имали увид у пример *PTP* инструкције дефинисане у оквиру *Robotics First* окружења, док им је при решавању петог задатка дата могућност увида у један пример такве инструкције. Пример *PTP* инструкције дефинисане у оквиру *Robotics First* окружења, која је дата на увид ученицима, приказан је на слици 10.1.

Шести задатак се састојао од три *PTP* инструкције кретања којима је било потребно реализовати кретање за три различите осе робота. При решавању шестог задатка ученици су имали могућност увида у пример *PTP* инструкције кретања дефинисане у оквиру *Robotics First* окружења.



Слика 10.1 Пример *PTP* инструкције у оквиру *Robotics First* окружења

10.2 Евалуација окружења *Robotics First*

Евалуација окружења *Robotics First* са аспекта ефикасности и интуитивности при програмирању и симулацији кретања индустријских робота је извршена анализом резултата тест задатака описаних у претходном поглављу и попуњавањем унапред припремљених упитника у којима су ученици имали прилику да искажу своје мишљење о предложеном решењу.

Табела 10.1 Резултати тест задатака за ученике треће и четврте године средње електротехничке школе, смер аутоматика (број ученика који су успешно решили задатак/укупан број ученика)

		<i>L-IRL</i> окружење и програмски језик	<i>Robotics First</i> окружење
Кретање једне осе индустријског робота (без увида у пример <i>PTP</i> инструкције кретања)	Трећа година	0/10	9/10
	Четврта година	4/14	12/14
Кретање једне осе индустријског робота (по угледу на пример <i>PTP</i> инструкције кретања)	Трећа година	6/10	9/10
	Четврта година	5/14	12/14
Кретање три осе индустријског робота – три засебне <i>PTP</i> инструкције кретања (по угледу на пример <i>PTP</i> инструкције кретања)	Трећа година	2/10	9/10
	Четврта година	5/14	12/14

Добијени резултати тест задатака су приказани у табели 10.1. Прва вредност представља број ученика који су успешно решили одговарајући задатак, док друга вредност представља укупан број ученика који су решавали дати задатак. Први задатак, у коме је било потребно написати једну *PTP* инструкцију којом се врши померање једне осе робота на *L-IRL* језику без увида у пример инструкције, решило је свега 4 од 24 ученика. Са могућим увидом у пример написане *PTP* инструкције на *L-IRL* језику исти задатак је решио нешто већи број ученика, укупно 11 од 24. Трећи задатак, који се састоји од три засебне *PTP* инструкције на *L-IRL* језику са увидом у дати пример, решило је укупно 7 од 24 ученика. Добијени резултати указују

на то да кратка презентација о инструкцијама *L-IRL* језика и раду *L-IRL* окружења није довољна да већина ученика успешно реши једноставан задатак са *L-IRL* инструкцијама кретања. Од укупног броја ученика трећег разреда ниједан није успео успешно да реши задатак без увида у дати пример. Увид у пример *L-IRL* програма је донекле повећао број ученика који су успешно решили други задатак, али је и даље тај број износио испод 50 процената од укупног броја ученика трећег и четвртог разреда. Са увидом у пример инструкције кретања, трећи, сложенији задатак је решио нешто мањи број ученика у односу на други, једноставнији задатак.

Резултати добијени након употребе *Robotics First* окружења указују на то да је кратка презентација о раду овог окружења сасвим довољна да већина ученика реши дате задатке без потребе увида у пример. Број ученика који су успешно решили задатак са увидом у пример је исти као и број ученика који су успешно решили задатак без увида у пример инструкције кретања, што значи да увид у пример инструкције кретања при решавању задатка употребом овог окружења не утиче на успешност решавања. У оба случаја број ученика који су успешно решили задатак је висок и износи 21 од 24 ученика или 87,5 процената. Вредност од 87,5 процената указује на лакоћу употребе и интуитивност предложеног решења. Шести, сложенији задатак је решио исти број ученика (87,5 процената) као и четврти, односно пети задатак, што указује на то да је употребом предложеног *Robotics First* окружења подједнако лако програмирати и једноставније и сложеније програме кретања.

Ученицима су затим дати унапред припремљени упитници са седам тврдњи у оквиру којих су могли да оцене употребљивост предложеног окружења *Robotics First*, његове функционалности и особине (даљински приступ лабораторијама у оквиру курсева, интуитивност употребе), као и да упореде предложено решење са решењима код којих се интеракција корисника базира на употреби *PC*-ја или интернет претраживача. У оквиру анкете ученицима је постављено питање да ли по њиховом мишљењу паметни телефони имају потенцијал да постану широко прихваћени и

стандардни алати за учење. На крају анкете ученици су могли да упореде предложено *Robotics First* окружење са *L-IRL* окружењем, код кога се програмирање спроводи употребом програмског језика, у погледу интуитивности. За мерење нивоа сагласности учесника са тврдњама датим у анкети коришћена је Ликертова скала са вредностима од “1”, што означава одговор “уопште се не слажем”, до “5”, што означава одговор “ потпуно се слажем”. Након завршетка анкете учесницима је усмено постављено питање како би се по њиховом мишљењу окружење *Robotics First* могло унапредити.

Табела 10.2: Q: Оцени окружење *Robotics First* и потенцијал употребе паметних мобилних уређаја

	Средња вредност оцне (ученици треће године средње школе)	s (ученици треће године средње школе)	Средња вредност оцне (ученици четврте године средње школе)	s (ученици четврте године средње школе)
<i>Robotics First</i> окружење је корисно за учење у области роботике.	4.4	0.24	4.50	0.96
Даљински приступ лабораторијама са било ког места у било које време имао би велики значај у стицању практичног знања а посебно у ситуацијама када не постоји могућност употребе лабораторијске опреме.	4.7	0.21	4.43	0.96

Мобилне апликације и окружења (<i>Robotics First</i>) користио бих за учење чешће него <i>PC</i> апликације и окружења.	3.4	1.84	4.50	0.54
Мобилне апликације и окружења (<i>Robotics First</i>) користио бих за учење чешће него интернет апликације (апликације којима се може приступити употребом интернет претраживача).	3.7	1.01	4.71	0.35
Паметни телефони имају потенцијал да постану стандардно усвојен алат за учење.	4.5	0.25	4.71	0.49
<i>Robotics First</i> окружење је интуитивно и <i>user-friendly</i> .	4.3	0.41	4.86	0.26
<i>Robotics First</i> окружење је интуитивније и лакше за употребу од <i>L-IRL</i> окружења.	4.6	0.24	4.71	0.49

Резултати овог истраживања дати су у табели 10.2. Резултати су приказани посебно за ученике трећег разреда средње школе и посебно за ученике четвртог разреда средње школе. За сваку од тврдњи израчуната је и стандардна девијација узорка s , односно одступање од средње вредности.

Средње оцене за први аспект у оквиру упитника Q показују да су ученици мишљења да је *Robotics First* окружење корисно за учење програмирања кретања робота. Средње оцене дате за други аспект у оквиру датих тврдњи указују на то да ученици сматрају да би могућност даљинског приступа лабораторијама, независног од локације корисника, била јако

корисна за стицање практичног знања на пољу роботике. Трећи аспект који је дат у оквиру упитника Q процењује колико ученика за учење чешће користи паметни телефон у односу на PC. Иако је овај аспект добио нешто нижу оцену од стране ученика трећег разреда, оцене дате од стране ученика четвртог разреда су доста високе, док су се у петом аспекту ученици оба разреда сложили да паметни телефони имају потенцијал да постану стандардно усвојени алат за учење. Средње оцене за четврти аспект у оквиру анкете Q указују на то да ученици трећег разреда дају благу предност мобилним апликацијама у односу на апликације којима се може приступити путем интернет претраживача у оквиру мобилног уређаја или PC-ја, док ученици четвртог разреда дају значајну предност мобилним апликацијама и окружењима. Средње оцене ученика четвртог разреда су за трећи и четврти аспект упитника Q веће у односу на средње оцене које су дали ученици трећег разреда. То указује да ученици који имају веће предзнање дају већи значај употреби паметних мобилних уређаја. Средње оцене које су ученици дали за шести аспект анкете показују да се они слажу у чињеници да је предложено окружење интуитивно и кориснички оријентисано (*user-friendly*). За ову тврдњу ученици четвртог разреда су дали високу просечну оцену (4,86). Оцене дате за седми аспект у оквиру упитника Q доказују да је *Robotics First* окружење интуитивније и лакше за употребу од *L-IRL* окружења код кога је интеракција са корисником базирана на употреби PC-ја и код кога се програмирање извршава користећи програмски језик специјалне намене. Ученици су у коментарима на питање о могућем унапређењу окружења *Robotics First* предлагали употребу светлијих боја у дизајну графичког корисничког интерфејса.

За обављена истраживања израчунат је *Cronbach*-ов коефицијент поузданости (*Cronbach α*) [22]. *Cronbach*-ов коефицијент поузданости је израчунат посебно за резултате истраживања у коме су учествовали ученици трећег разреда средње школе и посебно за резултате истраживања у коме су учествовали ученици четвртог разреда средње школе. Ово израчунавање је извршено са циљем да се тестира поузданост добијених резултата

истраживања. Добијена вредност *Cronbach*-овог коефицијента α износи 0.731 за ученике трећег разреда, односно 0.802 за ученике четвртог разреда средње школе. Добијене вредности указују на прихватљиву поузданост, односно на добру поузданост истраживања.

10.3 Дискусија

Резултати истраживања које је описано у поглављу 5.1 указују на то да постоји потреба за мобилним окружењем за едукацију у роботизи које даје могућност даљинског приступа лабораторијама независно од локације корисника уз интуитивну интеракцију, а са циљем да се олакша разумевање роботских програма и алгоритама, али и превазиђе недостатак и неприступачност лабораторијске опреме.

Поређењем са описаним решењима за едукацију у роботизи која су базирана на употреби *PC*-ја за интеракцију са корисником [7], [14], [15], [28], [29], [30], [42], [46], [71], [76], [90], [130], предложено окружење *Robotics First* унапређује поступак едукације у роботизи доносећи два главна побољшања: интуитивност и једноставност у интеракцији и могућност употребе било где и било када. Предложени начин интеракције приликом управљања робота је далеко једноставнији и интуитивнији у односу на интеракцију путем графичког корисничког интерфејса базираног на употреби програмског језика и *PC*-ја који описан у поглављу 3.2. У раду [103] је забележено да су генерално мања новчана средства потребна за опремање студената са мобилним уређајима него са десктоп или лаптоп рачунарима. У раду [113] је наведено да је опремање мобилним уређајима најекономичнији начин за опремање студената са рачунарским уређајем који они могу користити и од куће и који се може повезати на интернет. Ова економичност у опремању одговарајућом опремом представља треће побољшање које уводи предложено окружење *Robotics First* у односу на решења базирана на употреби *PC*-ја за едукацију у роботизи.

Robotics First окружење је развијено поштујући сугестије које су дате у раду [37] са циљем да се превазиђу недостаци решења која се могу користити употребом интернет претраживача на мобилним уређајима [15], [30], [42], [46], [71], [130], као и са циљем да се садржај за учење прилагоди екранима мобилних уређаја и интеракцији базираној на употреби екрана осетљивих на додир. Окружење *Robotics First* захтева унос минималне количине података приликом програмирања роботских задатака а инструкције за програмирање кретања су концизне. Предложено решење даје могућност извршавања симулације и даљинског управљања са једном акцијом корисника у оквиру екрана мобилних уређаја. Оваква имплементација редукује количину садржаја графичког корисничког интерфејса у текстуалном формату и број акција корисника потребних да се изврши неки догађај и на тај начин испуњава прву и другу сугестију за имплементацију мобилних апликација за учење описану у раду [37]. Визуелизација представља један од најважнијих алата за учење, поготово у области роботике. Она даје могућност да се добије јасан увид у извршавање роботских процеса и олакшава разумевање роботских програма. Функционалности надзора код предложеног решења *Robotics First* су базирани на 3Д визуелизацији. Такав облик приказа информација је најпогоднији за окружења намењена едукацији у роботици. Оваква имплементација испуњава трећу сугестију која је предложена пројектантима мобилних апликација за учење, дату у раду [37]. Могућност рада као самостална *offline* апликација је још једна предност предложеног решења у односу на апликације за мобилно учење којима се приступа путем интернет претраживача мобилних уређаја. Након почетне инсталације, све потребне функционалности остају на уређају и након тога за рад у симулационом режиму није потребна интернет конекција. Оваква имплементација може бити корисна у случајевима када не постоји могућност да се користи интернет, било путем *Wi-Fi* или мобилне мреже, или када корисник нема довољно кредита за употребу интернета, што може бити доста значајно за млађе генерације [95]. Симулација и тестирање роботских задатака могу

бити извршени *offline* а касније, када се добије могућност приступа интернету, искоришћени за даљинско управљање робота. Мобилне апликације које су инсталиране у оквиру мобилних уређаја су уочљивије у односу на апликације којима се приступа путем интернет претраживача [81]. *Robotics First* смешта свој лого у оквиру листе апликација, омогућавајући приказ и видљивост на дневној бази и подсећајући кориснике на чешћу употребу, што може допринети томе да ученици и студенти унапреде постојеће и стекну ново знање. Апликације којима се може приступити путем интернет претраживача мобилних уређаја немају могућност да користе податке са сензора мобилног уређаја, као што су жirosкоп или GPS [81], док апликације базиране на употреби *PC*-ја захтевају примену додатне опреме за прикупљање података о локацији или оријентацији. Функционалности базиране на употреби сензора, као што је жirosкоп или акцелерометар, су укључене у план за даљи развој *Robotics First* окружења са циљем да се омогући програмирање роботских задатака и даљинско управљање на основу команди добијених покретима мобилног уређаја који се држи у руци, чиме би се процес програмирања и управљања учини још интуитивнијим и природнијим.

Истраживање о употребљивости и успешности коришћења предложеног окружења *Robotics First* из перспективе ученика средње електротехничке школе “Стари град”, које је извршено у оквиру ове докторске дисертације и које је описано у претходном поглављу, потврђује да је окружење за мобилно учење *Robotics First* интуитивно, кориснички оријентисано и лакше за употребу од окружења код кога је интеракција са корисником заснована на употреби *PC*-ја и код кога се програмирање кретања робота обавља употребом програмског језика специјалне намене. Резултати истраживања доказују да окружење *Robotics First* даје могућност ученицима да науче да пишу програме за кретање робота уз краткотрајну обуку без потребе дубоког познавања специфичних парадигми роботског програмског језика, односно његове синтаксе и семантике. На тај начин

ученицима се оставља могућност да се сконцентришу на битне делове предмета обуке, као што су разумевање роботских задатака и алгоритама.

Robotics First окружење има циљ да мотивише ученике средњих школа који немају много искуства из области роботике и који немају приступ роботским лабораторијама, да се боље упознају са овом облашћу на један економичан и приступачан начин. Ово окружење даје могућности ученицима да упознају и истраже различите управљачке стратегије и да тестирају различите роботске програме пре њихове примене у роботским лабораторијама и у индустријским условима, са циљем да се значајно редукује могућност настанка оштећења или хаварије у току тестирања. Поред тога, даљински приступ роботској лабораторији, као једна од главних функционалности предложеног решења, оставља ученицима више времена за интеракцију са роботима у оквиру лабораторије и самим тим даје могућност за боље разумевање роботских задатака и анализу теоријских концепата. Употребом предложеног решења сваки ученик може да користи лабораторијску опрему индивидуално, што може бити знатно ефикасније од извршавања групних задатака.

Чињеница да паметан мобилни уређај још увек не поседује баш сваки студент или ученик се може сматрати недостатком предложеног решења. Ипак, број студената и ученика који не поседују паметне мобилне телефоне у данашње време је веома мали и постоје предвиђања да ће тај број наставити да се смањује са обзиром на то да се употреба паметних мобилних уређаја веома брзо шири. Другим недостатком се може сматрати то што се предложено решење односи на само једну мобилну платформу, Андроид платформу. Овај недостатак се може превазићи развојем одговарајућих окружења и за друге популарне мобилне платформе, као што је *iOS*, иако је, као што је приказано у овој дисертацији, употреба Андроид уређаја далеко више распрострањена.

11. ЗАКЉУЧАК

Предмет истраживања ове дисертације је бежично и интуитивно програмирање, управљање и надзор робота на мобилним уређајима, са циљем да се унапреде интеракција између корисника и роботског система и приступ роботским лабораторијама, уз потребно прилагођење садржаја окружења према карактеристикама и функционалностима мобилних уређаја.

Методологија која је примењена приликом израде докторске дисертације је систематизовано прикупљање и преглед постојећих научних резултата и достигнућа из области управљања робота, са акцентом на решењима која користе паметне телефоне и мобилне платформе. Метода евалуације и класификације је коришћена у делу истраживања постојећих софтверских решења за даљинско програмирање, управљање и надзор рада робота и других процеса, у индустријским условима и у едукацији. Класификација је извршена према начину програмирања кретања робота и роботских задатака, мобилности и економичности како би се истражиле најефективније појединачне функционалности постојећих софтверских решења и како би се дали предлози за превазилажење њихових недостатака. Метода анализе је коришћена у делу истраживања услова за прилагођење садржаја окружења према карактеристикама и функционалностима мобилних платформи. На основу донетих закључака извршено је формирање нове методе за даљинско програмирање, управљање и надзор робота и реализација циљног софтверског система који примењује предложену методу. За верификацију предложеног решења користиле су се експерименталне методе, на реалним софтверским системима и пројектима. Статистичке методе су коришћене у делу истраживања колико ће предложено решење допринети једноставности, интуитивности и мобилности у области едукације у роботизици.

У односу на постојећа решења за програмирање и управљање робота која су базирана на употреби *PC*-ја за интеракцију са корисником [9], [20],

[59], [61], [67], [115], [122], [131], [142], [144] и употреби програмског језика за програмирање кретања робота [90], решење предложено у овој докторској дисертацији даје две главне предности: унапређење интеракције између корисника и робота и унапређен приступ роботским лабораторијама. Унапређење интеракције са роботом је постигнуто обезбеђивањем режима рада брзог бирања за извршавање колекције најчешће коришћених сложених роботских задатака са минималним бројем акција корисника у оквиру екрана мобилног уређаја и режима рада за програмирање који елиминише потребу за детаљним познавањем парадигми роботских програмских језика, као и потребу за познавањем његове синтаксе и семантике. Употребом предложених режима рада брзог бирања и режима рада за програмирање редукује се време и број акција корисника потребних за припрему и извршавање роботских програма. Оваквом имплементацијом омогућено је и мање искусним корисницима да програмирају и управљају роботе, па је самим тим редукована и потреба за обуком од стране професионалаца. Приступ роботским лабораторијама је унапређен омогућавањем бежичне комуникације са роботском лабораторијом путем интернета или у локалној рачунарској мрежи. Предложено софтверско окружење превазилази недостатке и интегрише најефективније функционалности постојећих софтверских решења из области даљинског програмирања и управљања робота [2], [4], [44], [58], [62], [66], [99], [102], [104], [111], [123], [137], [138], обезбеђујући интуитивност у интеракцији при управљању и програмирању робота, даљински надзор робота кроз виртуелна окружења и пренос минималне количине података потребних за управљање и надзор робота. Решење које је описано у овој докторској дисертацији предлаже надзор кретања робота уз помоћ 3Д виртуелног модела робота или праћењем путање кретања енд-ефектора робота. Оваква имплементација даје природнији и вернији приказ у односу на надзор који се обавља праћењем промена вредности параметара од значаја у текстуалном формату. Због своје мобилности, односно локацијске независности, са могућношћу бежичне комуникације, предложено решење се може користити за даљински надзор,

али и у оквиру роботских лабораторија, са циљем да се добије могућност надзора рада робота из више перспектива истовремено: посматрањем кретања робота у лабораторији и истовременим посматрањем кретања виртуелног робота на мобилном уређају из друге перспективе. Овакво посматрање и проучавање понашање робота из више перспектива може да да бољи увид у сам рад робота. Поред тога, бежична комуникација унапређује и безбедност корисника. Тестирање и верификација роботских програма је омогућена кроз употребу симулатора кретања робота и ручног режима рада. Симулатор омогућава верификацију програма за кретање робота пре њихове примене у роботској лабораторији са циљем да се нежељено понашање у току симулације идентификује, да се програм редефинише и да се постигне жељено понашање. Управљање у ручном режиму рада омогућава тестирање програма и алгоритама за кретање робота корак по корак. Предложено решење решава питање преношења велике количине података приликом слања инструкција и програма за кретање робота и приликом даљинског надзора кретања робота. Подаци потребни за извршавање роботских задатака, који се преносе путем интернета или у локалној мрежи, се представљају у бинарном формату. Подаци који се користе за надзор се шаљу у текстуалном формату и представљају позиције робота у унутрашњим координатама. Оваква имплементација елиминише потребу за преносом велике количине података током управљања и надзора, као што је случај са имплементацијом надзора кретања робота слањем слика са камере [99].

Предложено решење је имплементирано користећи уређаје са Андроид оперативним системом који у су данашње време популарни, широко распрострањени и приступачни, што доприноси употребљивости и економичности решења. Тиме се даје могућност млађој популацији, а будућим инжењерима у области аеро и роботске индустрије, да користе исте уређаје у радном окружењу, као што их користе данас у свакодневној употреби.

У оквиру ове докторске дисертације предложено решење је примењено на програмирање, управљање и даљински надзор индустријског

робота Лола 15 и на програмирање и симулацију кретања индустријског робота Лола 50, центрифуге моделоване као манипулатор са три степена слободе и уређаја за дезоријентацију пилота моделованог као манипулатор са четири степена слободе. Верификација је извршена поређењем укупног броја акција корисника, односно оператера, потребних за програмирање, даљинско извршавање и симулацију пет тест програма применом предложеног софтверског система и применом *L-IRL* програмског језика и Лола УС-а код кога се интеракција са корисником базира на употреби РС-ја. Резултати поређења су показали да је предложено софтверско окружење утицало на значајно редуковање броја акција корисника потребних за програмирање новог програма за кретање робота (и до преко 70 процената), као и да је постигнуто максимално редуковање броја акција корисника потребних за даљинско извршавање или симулацију претходно дефинисаног програма у случају када је окружење већ покренуто и када је корисник претходно извршавао, односно симулирао извршавање програма за истог изабраног робота.

Предложена метода и софтверски систем се могу применити на програмирање, управљање и надзор и других робота, проширивањем скупа инструкција за програмирање кретања у оквиру графичког корисничког интерфејса режима рада за програмирање, укључивањем одговарајућих алгоритама кретања и имплементацијом одговарајућег 3Д модела робота. Подешавање графичког корисничког интерфејса режима рада брзог бирања не би било потребно, већ би се само жељени елементи тог интерфејса повезали са одговарајућим програмима за кретање робота. Бежична комуникација би се могла имплементирати и реализовати и на другим робот контролерима отворене архитектуре.

Са обзиром на ограниченост ресурса у области роботике и на чињеницу да такви ресурси обично нису доступни за организовање практичне наставе, студенти и ученици средњих школа често нису у прилици да научне да програмирају и управљају роботе у оквиру практичне наставе. Такође, за програмирање и управљање робота потребно је имати одређено

предзнање, а потребно је и познавати синтаксу и семантику роботских програмских језика. Истраживање које је потврдило потребу за унапређењем решења за едукацију у роботизи је обављено у оквиру ове докторске дисертације и у њему је учествовало 27 ученика средњих школа и 37 студената. Постоји одређени број решења за едукацију у роботизи која су базирана на употреби PC-ја за интеракцију са корисником или којима се може приступити путем интернет претраживача мобилних уређаја. Један број таквих решења је описан у оквиру ове докторске дисертације [7], [14], [15], [28], [29], [30], [42], [46], [71], [76], [90], [130]. Циљ примене предложеног решења у образовном процесу у области роботике је да се унапреди знање ученика у овој области и да се они мотивишу да раде са роботским системима.

Као део докторске дисертације обављено је квалитативно истраживање које разматра ефикасност и употребљивост решења *Robotics First* из перспективе ученика трећег и четвртог разреда средње електротехничке школе, смер аутоматика. Са становишта извршене евалуације може се дати закључак да је окружење *Robotics First* интуитивно, кориснички оријентисано и лакше за употребу од окружења код кога је интеракција са корисником заснована на употреби PC-ја и код кога се програмирање кретања робота обавља употребом програмског језика специјалне намене. Резултати евалуације доказују да окружење *Robotics First* омогућава ученицима да науче да пишу програме за кретање робота уз краткотрајну обуку без потребе дубоког познавања специфичних парадигми роботског програмског језика, односно његове синтаксе и семантике, чиме им се оставља могућност да се сконцентришу на битне делове предмета обуке, као што су разумевање роботских задата и алгоритама. Даљинским приступом роботским лабораторијама превазилази се недостатак ресурса у области роботике и омогућава се ученицима да унапреде своје знање из ове области. Предложено решење унапређује интеракцију са роботом, унапређује приступ роботским лабораторијама и обезбеђује приступачност у смислу цене, чиме се превазилазе недостаци постојећих решења базираних на

употреби PC-ја и интернет претраживача мобилних уређаја за интеракцију са роботом. Употребом окружења *Robotics First* ученици средњих школа се мотивишу да науче фундаменталне концепте роботике, а студенти са одговарајућим предзнањем из сродних области могу да науче како да програмирају сложене роботске програме и да тестирају различите управљачке стратегије на интуитиван и једноставан начин. Паметни мобилни телефони и таблети, који се користе као једина потребна опрема, немају високу цену и лако су доступни, а ученици и студенти су већ упознати са радом оваквих уређаја. Због наведених разлога ово решење је погодно за употребу на практичној настави са великим групама ученика. *Robotics First* се лако може проширити новим модулима за различите типове робота и може се прилагодити различитим нивоима курсева, у складу са предзнањем сваког ученика. Након спроведених истраживања и евалуације може се дати закључак да окружење *Robotics First* може бити успешно коришћено за едукацију ученика средњих школа, као и да је мобилно учење (*m-learning*) изазовна техника за едукацију у области роботике.

У оквиру дисертације постигнути су следећи научни доприноси:

- Систематизација и преглед постојећих софтверских решења у области програмирања и управљања робота са акцентом на решењима која користе паметне мобилне уређаје, у индустрији, научним и академским институцијама, као и у едукацији.
- Класификација и анализа са критичким освртом на погодности коришћења постојећих софтверских решења у области употребе паметних телефона и мобилних платформи за програмирање, управљање и надзор различитих процеса на основу начина програмирања роботских програма и једноставности коришћења, мобилности и економичности, а у циљу постизања што интуитивнијег новог софтверског решења.
- Анализа услова за прилагођење садржаја окружења за програмирање, управљање и надзор робота мобилним платформама и паметним телефонима.

- Формирање нове методе за даљинско програмирање, управљање и надзор робота на основу детаљне анализе постојећих софтверских решења у области програмирања, управљања и надзора робота са акцентом на решењима која користе мобилне уређаје, уз испуњење услова за прилагођење садржаја окружења према карактеристикама мобилних уређаја, којом се:
 - Унапређује интеракција између корисника и роботског система приликом програмирања, управљања и надзора рада робота.
 - Унапређује приступ роботским лабораторијама кроз виртуелно окружење независно од локације корисника.
 - Превазилазе идентификовани недостаци и интегришу најефективније функционалности постојећих софтверских решења у једну целину, обезбеђујући интуитивност, локацијску независност и економичност у управљању робота у индустријским условима, али и у едукацији.
- Формирање и предлог нове методе за едукацију у роботизи којом се редукује време потребно за савладавање градива из области управљања и програмирања робота, као и потреба за асистенцијом професионалаца, уз омогућавање анализе индустријских процеса из више перспектива.
- Анализа могућих технологија за развој новог оригиналног софтверског система који примењује предложену методу за даљинско програмирање, управљање и надзор робота.
- Предлог и имплементација новог оригиналног софтверског система који примењује предложену методу за даљинско програмирање, управљање и надзор робота.
- Предлог примене новог оригиналног софтверског окружења у образовном процесу (*Robotics First*).

- Верификација предложене методе и имплементираног софтверског система применом у лабораторијским условима.
- Евалуација ефикасности и употребљивости предложене методе и софтверског система приликом примене на практичној настави.

Будуће активности које би могле бити продужетак овог рада се могу поделити на неколико праваца. Први правац би могао да укључује проширење функционалности програмирања роботских задатака уз помоћ сензора које поседују Андроид уређаји, као што су акцелерометар или жироскоп. Међутим, како ове методе не би биле довољно прецизне приликом специфицирања путање кретања енд-ефектора робота, предлог би био да се одговарајућим покретима руке при држању телефона специфицира жељени роботски задатак у оквиру режима рада брзог бирања. Други правац би могао да укључује могућност програмирања и управљања робота употребом микрофона, као једног од сензора Андроид мобилних уређаја. Тачно дефинисаним говорним командама би се иницирало слање извршног програма за кретање робота на системски рачунар. На тај начин би се прошириле функционалности интуитивног програмирања и управљања у оквиру предложеног решења. Трећи могући правац будућих активности би могао да буде прилагођење и развој софтверског окружења за друге мобилне платформе. На тај начин би се доступност предложеног окружења још више проширила. Четврти правац истраживања би се односио на развој реконфигурабилног окружења за симулацију и надзор робота [89], прилагођеног мобилним платформама.

ЛИТЕРАТУРА

- [1] 4G Wireless Technology, albert-learning.com, 2015, Available at: <http://www.albert-learning.com>, Accessed June 2016.
- [2] Abaid N, Bernhardt J, Fran, J A, Kapila V, Kimani D, Porfiri M, *Controlling a robotic fish with a smart phone*, Mechatronics, vol. 23, no. 5, pp. 491-496, 2013.
- [3] AETC White Paper, *On learning: The future of Air Force education and training*, Air Education and Training Command Randolph AFB TX, US, 2008.
- [4] Ahn H, Kim H, Oh Y, Oh S, *Smartphone-Controlled Telerobotic Systems*, Proceedings of the IEEE International Conference on Cyber-Physical Systems, Networks, and Applications CPSNA, IEEE, pp. 77-80, 2014.
- [5] Bison - GNU parser generator, Available at: www.gnu.org/software/bison/, Accessed October 2013.
- [6] Blackwell A F, *Metacognitive theories of visual programming: What do we think we are doing?*, Proceedings of the IEEE Symposium on Visual Languages, IEEE, pp. 240-246, 1996.
- [7] Blank D, Kumar D, Meeden L, Yanco H, *Pyro: An integrated environment for robotics education*, Proceedings of the 20th national conference on Artificial intelligence, AAAI Press, pp. 1718–1719, 2005.
- [8] Boja C, Zamfiroiu A, *Input methods in mobile learning environments*, Studies in Informatics and Control, vol. 22, no. 4, pp. 329–338, 2013.
- [9] Bouzouia B, Rahiche A, *Teleoperation system of the mobile Manipulator Robot ROBUTER_ULM: Implementation issues*, Proceedings of the XXII International Symposium on Information, Communication and Automation Technologies ICAT, IEEE, pp. 1-7, 2009.
- [10] Binkley J, *TCP/IP - Socket Programming*, Available at: <http://web.cecs.pdx.edu/~jrb/tcpip/lectures/pdfs/>, Accessed June 2016.
- [11] Brogårdh, T, *Present and future robot control development—An industrial perspective*, Annual Reviews in Control, vol. 31, no. 1, pp. 69-79, 2007.

- [12] Brscic D, *Tool for remote control of mobile robots with visualization by means of virtual reality*, Proceedings of the 7th International Conference on Telecommunications ConTEL 2003, vol. 2, IEEE, pp. 673-675, 2003.
- [13] Cabot A G, Garcia-Lopez E, de-Marcos L, Abraham-Curto J, *Adapting learning contents to mobile devices and context to improve students' learning performance: A case study*, Journal of Universal Computer Science, vol. 20, no. 15, pp. 2032-2042, 2014.
- [14] Carpin S, Lewis M, Wang J, Balakirsky S, Scrapper C, *USARSim: A robot simulator for research and education*, Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, pp. 1400-1405, 2007.
- [15] Castellanos A R S, Santana L H, Rubio E, Ching I S, Santonja R A, *Virtual and remote laboratory for robot manipulator control study*, International Journal of Engineering Education, vol. 22, no. 4, pp. 702-710, 2006.
- [16] Chorfi H, Sevkli A, Bousbahi F, *From e-learning to m-learning: Context-aware CBR system*, Neural Information Processing, Springer Berlin/Heidelberg, pp. 528-534, 2012.
- [17] Clint S, Daniel C, *3G wireless networks*, McGraw-Hill Professional, 2002.
- [18] Conde M Á, García-Peñalvo F J, Alier M, Piguillem J, *The implementation, deployment and evaluation of a mobile personal learning environment*, Journal of Universal Computer Science, vol. 19, no. 7, pp. 854-872, 2013.
- [19] Conder S, Darcey L, *Android Wireless Application Development*, Pearson Education, 2010.
- [20] Connolly C, *Technology and applications of ABB Robot-Studio*, Industrial Robot: An International Journal, vol. 36, no. 6, pp. 540-545, 2009.
- [21] Cooper M, Ferreira J M, *Remote laboratories extending access to science and engineering curricular*, IEEE Transactions on learning technologies, vol. 2, no. 4, pp. 342-353, 2009.
- [22] Cronbach L, *Coefficient alpha and the internal structure of tests*, Psychometrika, vol. 16, no. 3, pp. 297-334, 1951.

- [23] Cui L, Tso F P, Yao D, Jia W, *WeFiLab: A web-based wifi laboratory platform for wireless networking education*, IEEE Transactions on Learning Technologies, vol. 5, no. 4, pp. 291-303, 2012.
- [24] Cvetanović M, Radivojević Z, *Programiranje mobilnih uređaja*, School of Electrical Engineering, University of Belgrade, Serbia, Available at: <http://rti.etf.bg.ac.rs/rti/si4pmu/>, Accessed May 2016..
- [25] Dankasa J, *Examining the progression of mobile technologies and their applications to learning environment: Implications for m-learning in Africa*, International Journal of Research Studies in Educational Technology, vol. 3, no. 2, pp.51-64, 2014.
- [26] Dantu V, Vempati J, Srivilliputhur S, *Non-invasive blood glucose monitor based on spectroscopy using a smartphone*, Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, pp. 3695-3698, 2014.
- [27] Dhir A, Gahwaji N M, Nyman G, *The role of the iPad in the hands of the learner*, Journal of Universal Computer Science, vol. 19, no. 5, pp. 706-727, 2013.
- [28] Dogmus Z, Erdem E, Patoglu V, *ReAct!: An interactive educational tool for AI planning for robotics*, IEEE Transactions on Education, vol. 58, no. 1, pp. 15-24, 2015.
- [29] Draganjac I, Sesar V, Bogdan S, Kovacic Z, *An internet-based system for remote planning and execution of SCARA robot trajectories*, Proceedings of the 34th Annual Conference of Industrial Electronics IECON, IEEE, pp. 3485-3490, 2008.
- [30] Dunn T L, Wardhani A, *A 3D robot simulation for education*, Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia, ACM, pp. 277-278, 2003.
- [31] Eclipse Project, Available at: <https://eclipse.org/eclipse/>, Accessed June 2013.

- [32] Ferenc G, Dimić Z, Lutovac M, Kvrđić V, Cvijanović V, *Distributed robot control system based on the real-time linux platform*, Journal of Mechanics Engineering and Automation, vol. 2, no. 3, 2012.
- [33] Ferenc G, Dimić Z, Lutovac M, Vidaković J, Kvrđić V, *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.
- [34] Ferenc G, Lutovac M, Dimić Z, Vidaković J, Kvrđić V, *Development of a real-time system based on the modular FSM in distributed system for robot control*, Annals of the Faculty of Engineering Hunedoara, vol. 11, no. 2, pp. 221-226, 2013.
- [35] Ferenc G, Lutovac M, Vidaković J, Dimić Z, Kvrđić V, *Real-time robot control logic using modular FSM*, Proceedings book of the 4th International Conference Management of Technology-Step to Sustainable Production, pp. 259-265, 2012.
- [36] Ferro E, Potorti F, *Bluetooth and Wi-Fi wireless protocols: A survey and a comparison*, IEEE Wireless Communications, vol. 12, no. 1, pp. 12-26, 2005.
- [37] Fisher M, Baird D E, *Making mLearning work: Utilizing mobile technology for active exploration, collaboration, assessment, and reflection in higher education*, Journal of Educational Technology Systems, vol. 35, no. 1, pp. 3-30, 2006.
- [38] Forson, a sentence generation tool, Available at: <http://forson.sourceforge.net/>, Accessed June 2013.
- [39] Frohberg D, Goth C, Schwabe G, *Mobile Learning Projects - A critical analysis of the state of the art*, Journal of Computer Assisted Learning, vol. 25, no. 4, pp. 307-331, 2009.
- [40] Gajin S, *Transportni sloj*, Računarske mreže 1, School of Electrical Engineering, University of Belgrade, Belgrade, 2008.
- [41] Galletta D F, Dunn B K, *Assessing smartphone ease of use and learning from the perspective of novice and expert users: Development and illustration of mobile benchmark tasks*, AIS Transactions on Human-Computer Interaction, vol. 6, no. 4, pp. 74-91, 2014.

- [42] Gil A, Peidro A, Reinoso Ó, Marín J M, *Implementation and assessment of a virtual laboratory of parallel robots developed for engineering students*, IEEE Transactions on Education, vol. 57, no. 2, pp. 92-98, 2014.
- [43] Goldberg K, Mascha M, Gentner S, Rothenberg N, Sutter C, Wiegley J, *Desktop teleoperation via the world wide web*, Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, IEEE, pp. 654-659, 1995.
- [44] Göbel S, Jubeh R, Raesch S L, Zündorf A, *Using the Android platform to control robots*, Proceedings of the 2th International Conference on Robotics in Education RiE, pp. 135-142, 2011.
- [45] Gracia J, Bayo E, *Stream sockets versus web services for high-performance and secure structural analysis in internet environments*, Journal of Computing in Civil Engineering, vol. 3, no. 1, pp. 47-56, 2009.
- [46] Habib M K, *Distributed teleoperation and collaborative environment for robotics e-learning and cooperation*, Proceedings of the 6th IEEE International Conference on Industrial Informatics, IEEE, pp. 1358-1363, 2008.
- [47] Hamel W R, Martin H L, *Robotics-related technology in the nuclear industry*, Robotics and robot sensing systems, vol. 442, pp. 97-107, 1983.
- [48] Hanson B, Culmer P, Gallagher J, Page K, Read E, Weightman A, Levesley M, *ReLOAD: Real laboratories operated at a distance*, IEEE Transactions on Learning Technologies, vol. 2, no. 4, pp. 331-341, 2009.
- [49] Hart S G, Staveland L E, *Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research*, Advances in psychology, vol. 52, pp. 139-183, 1988.
- [50] Händel P, Ohlsson J, Ohlsson M, Skog I, Nygren E, *Smartphone-based measurement systems for road vehicle traffic monitoring and usage-based insurance*, IEEE Systems Journal, vol. 8, no. 4, pp. 1238-1248, 2014.
- [51] Hossain A, Canning J, Ast S, Rutledge P J, Yen T L, Jamalipour A, *Lab-in-a-phone: smartphone-based portable fluorometer for pH measurements of environmental water*, IEEE Sensors Journal, vol. 15, no. 9, pp. 5095-5102, 2015.

- [52] Hu S Q, Fu Y, She C D, Yao H, *Enabling ZigBee communications in Android devices*, Advanced Materials Research, vol. 756, Trans Tech Publications, pp. 2125-2130, 2013.
- [53] Ioaneş C, Chioreanu A, *Current trends regarding the intuitive programming of industrial robots*, Acta Technica Napocensis-Series: Applied Mathematics, Mechanics, And Engineering, vol. 55, no. 1, pp. 187-192, 2012.
- [54] Ionescu C M, Fabregas E, Cristescu S M, Dormido S, De Keyser R, *A remote laboratory as an innovative educational tool for practicing control engineering concepts*, IEEE Transactions on Education, vol. 56, no. 4, pp. 436-442, 2013.
- [55] ITU - The International Telecommunication Union, *IMT-2000 Project – ITU*, Geneva, Switzerland, 2001-2002.
- [56] ITU - The International Telecommunication Union, *Measuring the Information Society*, International Telecommunication Union, Geneva, Switzerland, 2014.
- [57] ITU-R, Report M.2134, *Requirements related to technical performance for IMT-Advanced radio interface(s)*, November 2008.
- [58] Jan Y, Hassan S, Pyo S, Yoon J, *Smartphone based control architecture of teaching pendant for industrial manipulators*, Proceedings of the 4th International Conference on Intelligent Systems, Modelling and Simulation, IEEE, pp. 370-375, 2013.
- [59] Jara C A, Candelas F A, Gil P, Torres F, Esquembre F, Dormido S, *EJS+ EjsRL: An interactive tool for industrial robots simulation*, Computer Vision and remote operation, Robotics and Autonomous systems, vol. 59, no. 6, pp. 389-401, 2011.
- [60] Jasmin B, *C++ GUI Programming with Qt4*, Pearson Education India, 2008.
- [61] Jen Y H, Taha Z, Vui L J, *Vr-based robot programming and simulation system for an industrial robot*, International Journal of Industrial Engineering: Theory, Applications and Practice, vol. 15, no. 3, pp. 314-322, 2008.
- [62] Jenifer T M, Priyadharshini T V, Lavanya R, Pandian S R, *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.

- [63] Jerome J, *Virtual Instrumentation Using LabVIEW*, PHI Learning Pvt, 2010.
- [64] Jung S, *Experiences in developing an experimental robotics course program for undergraduate education*, IEEE Transactions on Education, vol. 56, no. 1, pp. 129-136, 2013.
- [65] Kim Y, Chon Y, Cha H, *Smartphone-based collaborative and autonomous radio fingerprinting*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 1, pp. 112-122, 2012.
- [66] Koceski S, Koceska N, Kocev I, *Design and evaluation of cell phone pointing interface for robot control*, International Journal of Advanced Robotic Systems, vol. 9, pp. 1-9, 2012.
- [67] Kofman J, Wu X, Luu T J, Verma S, *Teleoperation of a robot manipulator using a vision-based human-robot interface*, IEEE Transactions on Industrial Electronics, vol. 52, no. 5, pp. 1206-1219, 2005.
- [68] Kojić N, *REST web services*, School of Electrical Engineering, University of Belgrade, Belgrade, 2014.
- [69] Kojić N, *SOAP - Simple Object Access Protocol*, School of Electrical Engineering, University of Belgrade, Belgrade, 2010.
- [70] Kostikis N, Hristu-Varsakelis D, Arnaoutoglou M, Kotsavasiloglou C A, *Smartphone-based tool for assessing Parkinsonian hand tremor*, IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 6, pp. 1835-1842, 2015.
- [71] Kuzucuoglu A E, Erdemir G, *Development of a web-based control and robotic applications laboratory for control engineering education*, Information Technology and Control, vol. 40, no. 4, pp. 352-358, 2011.
- [72] Kvrđić V, Vidaković J, Lutovac M, Ferenc G, Cvijanović V, *A control algorithm for a centrifuge motion simulator*, Robotics and Computer-Integrated Manufacturing, vol. 30, no. 4, pp. 399-412, 2014.
- [73] Kvrđić V, *Development of intelligent systems for industrial robots control and programming*, PhD thesis, Faculty of Mechanical Engineering University of Belgrade, Belgrade, Serbia, 1998.

- [74] Kvrđić V, Višnjić Z, Cvijanović V, Divnić D, *Dynamics and control of a spatial disorientation trainer*, *Robotics and Computer-Integrated Manufacturing*, vol. 35, pp. 104–125, 2015.
- [75] Labiod H, Afifi H, De Santis C, *Wi-Fi, Bluetooth, Zigbee and WiMAX*, Springer, 2007.
- [76] Lee J, Taha Z, Yap H J, Kinsheel A, *Constructivist game-based robotics simulator in engineering education*, *International Journal of Engineering Education*, vol. 29, no. 4, pp. 1024-1036, 2013.
- [77] Lee J S, Su Y W, Shen C C, *A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*, *Proceedings of the 33rd Annual Conference of the Industrial Electronics Society IECON, IEEE*, pp. 46-51, 2007.
- [78] Lehr W, McKnight L W, *Wireless internet access: 3G vs. WiFi?*, *Telecommunications Policy*, vol. 27, no. 5, pp. 351-370, 2003.
- [79] Leondes C T, *Intelligent Systems: Technology and Applications*, CRC Press, 2002.
- [80] Li H, Chan E C, Guo X, Xiao J, Wu K, Ni L M, *Wi-counter: Smartphone-based people counter using crowdsourced Wi-Fi signal data*, *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 442-452, 2015.
- [81] Lionbridge, *Mobile web apps vs mobile native apps: How to make the right choice*, Available at: <http://www.lionbridge.com/files/2012/11/Lionbridge-WP-MobileApps2.pdf>, Accessed July 2015.
- [82] Lopez D A, Frank J A, Kapila V, *Comparing interface elements on a tablet for intuitive teleoperation of a mobile manipulator*, *Proceedings of the 12th Biennial Conference on Engineering Systems Design and Analysis, American Society of Mechanical Engineers*, pp. 1-10, 2014.
- [83] Lutovac Banduka M, *Remote Monitoring and Control of Industrial Robot based on Android Device and Wi-Fi Communication*, *Automatika–Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 56, no. 3, pp. 281-291, 2015.

- [84] Lutovac Banduka M, *Robotics First-A Mobile Environment for Robotics Education*, International Journal of Engineering Education, vol. 32, no. 2, pp. 818-829, 2016.
- [85] Lutovac M, Bojić D, Kvrđić V, *Automated testing of L-IRL robot programming language parser*, Proceedings of the 21st Telecommunications Forum TELFOR, IEEE, pp. 825-828, 2013.
- [86] Lutovac M, Bojić D, *Techniques for automated testing of Lola industrial robot language parser*, Telfor Journal, vol. 6, no. 1, pp. 69-74, 2014.
- [87] Lutovac M, Dimić Z, Ferenc G, Vidaković J, Bućan M, *Virtual robot in distributed control system*, Proceedings of the 20th Telecommunications Forum TELFOR, IEEE, pp. 1401-1404, 2012.
- [88] Lutovac M, Dimić Z, Ferenc G, Vidaković J, Kvrđić V, *Distribuirani sistem za kontrolu robota korišćenjem CORBA protokola*, Zbornik radova 56. konferencije ETRAN, str. RO1.3-1-4, 2012.
- [89] Lutovac M, Dimić Z, Mitrović S, Stepanović A, *Reconfigurable multi-robot virtual environment*, Proceedings of 23rd Telecommunications Forum TELFOR, pp. 954-957, 2015.
- [90] Lutovac M, Ferenc G, Kvrđić V, Vidaković J, Dimić Z, *Robot programming system based on L-IRL programming language*, Acta Technica Corviniensis-Bulletin of Engineering, vol. 5, no. 2, pp. 27-30, 2012.
- [91] Lutovac M, Ferenc G, Vidaković J, Dimić Z, Kvrđić V, *Usage of XML and P code for robot motion control*, Proceedings of the Mediterranean Conference on Embedded Computing MECO, IEEE, pp. 162-165, 2012.
- [92] Lutovac M, Kvrđić V, Ferenc G, Dimić Z, Vidaković J, *3D simulator for human centrifuge motion testing and verification*, Proceedings of the 2nd Mediterranean Conference on Embedded Computing MECO, IEEE, pp. 160-163, 2013.
- [93] Lutovac M, Protić J, Kvrđić V, *Remote control of industrial robot Lola 50 using wireless communication and Android device*, Proceedings of the 21st Telecommunications Forum TELFOR, IEEE, pp. 885-888, 2013.

- [94] Meier R, *Professional Android 4 application development*, John Wiley & Sons, 2012.
- [95] Mierau G, Straßer M, *Mobile native vs. web-based learning applications*, Mobile Learning Seminar, Learning Technologies Research Group, Computer Science Department, RWTH Aachen University, 2012.
- [96] Milutinović D, *Industrijski roboti*, Faculty of Mechanical Engineering University of Belgrade, 2015.
- [97] Mitrović S, Dimić Z, Vidaković J, Lutovac M, Kvirgić V, *System for simulation and supervision of robotic cells*, Proceedings of the 12th International Scientific Conference MMA, pp. 51-54, 2015.
- [98] Mladenović V, Lutovac M M, Lutovac M D, *Electronic tour guide for Android mobile platform with multimedia travel book*, Proceedings of the 20th Telecommunications Forum TELFOR, IEEE, pp. 1460-1463, 2012.
- [99] Moon S W, Kim Y J, Myeong H J, Kim C S, Cha N J, Kim D H, *Implementation of smartphone environment remote control and monitoring system for Android operating system-based robot platform*, Proceedings of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence URAI, IEEE, pp. 211-214, 2011.
- [100] Moore B R, *Evaluating and blending multimedia mobile applications into technical training*. PhD thesis, University of North Texas, USA, 2011.
- [101] Motiwalla L F, *Mobile learning: A framework and evaluation*, Computers & Education, vol. 49, pp. 581-596, 2007.
- [102] Nasereddin H H, Abdelkarim A A, *Smartphone control robots through Bluetooth*, International Journal of Research and Reviews in Applied Sciences, vol. 4, no. 4, pp. 399-404, 2010.
- [103] Naismith L, Lonsdale P, Vavoula G, Sharples M, *Literature review in mobile technologies and learning*, NESTA (National Endowment for Science Technology and the Arts), Bristol, UK, 2004.
- [104] Nádvorník J, Smutný P, *Remote control robot using Android mobile device*, Proceedings of the 15th International Carpathian Control Conference ICC, IEEE, pp. 373-378, 2014.

- [105] Ng P P, Duffy V G, Yucel G, *Impact of dynamic virtual and real robots on perceived safe waiting time and maximum reach of robot arms*, International Journal of Production Research, vol. 50, no. 1, pp. 161-176, 2012.
- [106] Nikolić B, Radivojević Z, Djordjević J, Milutinović V, *A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization*, IEEE Transactions on Education, vol. 52, no. 4, pp. 449-458, 2009.
- [107] Niwa M, Tsuda N, Kato N, Nomura Y, Matsui H, *Automatic control of a monitoring camera for remote manipulator operation*, Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication ROMAN, IEEE, pp. 151-156, 2006.
- [108] Offutt A J, Untch R H, *Mutation 2000: Uniting the orthogonal*, Mutation testing for the new century, pp. 34-44, Springer US, 2001.
- [109] OROCOS – Open Robot Control Software, Available at: <http://www.orocos.org>, Accessed May 2013.
- [110] Ozdamli F, Cavus N, *Basic elements and characteristics of mobile learning*, Procedia - Social and Behavioral Sciences, vol. 28, pp. 937-942, 2011.
- [111] Parga C, Li X, Yu W, *Tele-manipulation of robot arm with Smartphone*, Proceedings of the 6th International Symposium on Resilient Control Systems ISRCS, IEEE, pp. 60-65, 2013.
- [112] Pavlović M, *High level programming language for multi-robotic operations*, M.S. thesis, Scholl of Electrical Engineering, University of Belgrade, Belgrade, Serbia, 1994.
- [113] Perry D, *Hand-held computers (PDAs) in schools*, Coventry, UK, pp. 1-30, 2003.
- [114] Petrović I, *Upravljanje robotskim sustavima*, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, 2015/2016.
- [115] Pires J N, *Robot manipulators and control systems*, Industrial Robots Programming: Building Applications for the Factories of the Future, pp. 35-107, 2007.

- [116] Potkonjak V, *Robotika i automatizacija*, School of Electrical Engineering, University of Belgrade, Serbia, Available at: <http://automatizacija1.etf.rs/udzbenici.html>, Accessed May 2016.
- [117] Potkonjak V, Vukobratović M, Jovanović K, Medenica M, *Virtual Mechatronic/Robotic laboratory–A step further in distance learning*, Computers & Education, vol. 55, no. 2, pp. 465-475, 2010.
- [118] Purdom P, *A sentence generator for testing parsers*, BIT Numerical Mathematics, vol. 12, no. 3, pp. 366-375, 1972.
- [119] Sabani A H, Jailani R, *Android based control and monitoring system for leg orthosis*, Proceedings of the 11th International Colloquium on Signal Processing & Its Applications CSPA, IEEE, pp. 40-45, 2015.
- [120] Saltaren R, Aracil R, Alvarez C, Yime E, Sabater J M, *Field and service applications-exploring deep sea by teleoperated robot-an underwater parallel robot with high navigation capabilities*, IEEE Robotics & Automation Magazine, vol. 14, no. 3, pp. 65-75, 2007.
- [121] Salvesen K, Galeotti J P, Gross F, Fraser G, Zeller A, *Using dynamic symbolic execution to generate inputs in search-based GUI testing*, Proceedings of the 8th International Workshop on Search-Based Software Testing, IEEE Press, pp. 32-35, 2015.
- [122] Sett A, Vollmann K, *Computer based robot training in a virtual environment*, Proceedings of the IEEE International Conference on Industrial Technology, vol. 2, IEEE, pp. 1185-1189, 2002.
- [123] Singh A, Seo S H, Hashish Y, Nakane M, Young J E, Bunt A, *An interface for remote robotic manipulator control that reduces task load and fatigue*, Proceedings of the IEEE RO-MAN, pp. 738-743, IEEE, 2013.
- [124] Song W, Yu H, Liang C, Wang Q, Shi Y, *Body monitoring system design based on Android smartphone*, Proceedings of the World Congress on Information and Communication Technologies WICT, IEEE, pp. 1147-1151, 2012.
- [125] Spelta C, Manzoni V, Corti A, Goggi A, Savaresi S M, *Smartphone-based vehicle-to-driver/environment interaction system for motorcycles*, IEEE Embedded Systems Letters, vol. 2, no. 2, pp. 39-42, 2010.

- [126] Steele J, To N, *The Android developer's cookbook: building applications with the Android SDK*, Pearson Education, 2010.
- [127] Suki N M, Suki N M, *Mobile phone usage for m-learning: Comparing heavy and light mobile phone users*, *Campus-Wide Information Systems*, vol. 24, no. 5, pp. 355–365, 2007.
- [128] Summers M, *Robot capability test and development of industrial robot positioning system for the aerospace industry*, *SAE transactions*, vol. 114, no. 1, pp. 1108-1118, 2005.
- [129] Sunehra D, Bano A, Yandrathi S, *Remote monitoring and control of a mobile robot system with obstacle avoidance capability*, *Proceedings of the International Conference on Advances in Computing, Communications and Informatics ICACCI*, IEEE, pp. 1803-1809, 2015.
- [130] Šafarič R, Truntič M, Hercog D, Pačnik G, *Control and robotics remote laboratory for engineering education*, *iJOE International Journal on Online Engineering*, pp. 1–8, 2005.
- [131] Tang H, Xie X, Xiao J, *Design and implement of rescue robot control system based on wireless network*, *Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification*, IEEE, pp. 339-342, 2010.
- [132] The Fast Lexical Analyzer, Available at: <http://flex.sourceforge.net/>, Accessed October 2013.
- [133] The Statistics Portal Statista, Operating systems' market share of new smartphone sales worldwide from 1st quarter 2011 to 1st quarter 2016, Available at: <http://www.statista.com/statistics/236035/market-share-of-global-smartphone-os-shipments-by-mobile-operating-system-per-quarter/>, Accessed March 2016.
- [134] The difference between Wi-Fi, ZigBee, and Bluetooth, Available at: <https://blog.ecoventsystems.com/2015/03/wirelessprotocols/>, Accessed March 2016.
- [135] Thompson J, *The future of Air Force education and training an advance planning briefing to industry*, Air Education and Training Command Randolph AFB TX, US, 2009.

- [136] Trifonova A, Ronchetti M, *Where is mobile learning going?* Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, AACE, pp. 1794-1801, 2003.
- [137] Truong N V, Vu D L, *Remote monitoring and control of industrial process via wireless network and Android platform.* Proceedings of the International Conference on Control, Automation and Information Sciences ICCAIS, IEEE, pp. 340-343, 2012.
- [138] Van Delden S, Whigham A, *A Bluetooth-based architecture for Android communication with an articulated robot,* Proceedings of the International Conference on Collaboration Technologies and Systems CTS, IEEE, pp. 104-108, 2012.
- [139] Vidaković J, Ferenc G, Lutovac M, Kvrđić V, *Development and implementation of an algorithm for calculating angular velocity of main arm of human centrifuge,* Proceedings of 15th International in Power Electronics and Motion Control EPE/PEMC, IEEE, pp. DS2a-17, 2012.
- [140] Vidaković J, Lazarević M, Kvrđić V, Dančuo Z, Lutovac M, *Comparison of Numerical Simulation Models for Open Loop Flight Simulations in the Human Centrifuge,* Proceedings in Applied Mathematics and Mechanics PAMM, vol. 13, no. 1, pp. 485-486, 2013.
- [141] Vidaković J, Kvrđić V, Ferenc G, Lutovac M, *Kinematički model humane centrifuge,* Zbornik radova 56. konferencije ETRAN, Zlatibor, str. R01.4-1-4, 2012.
- [142] Wang D, Yi J, Zhao D, Yang G, *Teleoperation system of the internet-based omnidirectional mobile robot with a mounted manipulator,* Proceedings of the International Conference on Mechatronics and Automation, IEEE, pp. 1799-1804, 2007.
- [143] Wen C, Zhang J, *Design of a microlecture mobile learning system based on smartphone and web platforms,* IEEE Transactions on Education, vol. 58, no. 3, pp. 203-207, 2015.
- [144] Wolny R, *Robots in technological process of painting,* DAAAM International Scientific Book, pp. 195-204, 2011.

- [145] Woo M, Neider J, Davis T, *OpenGL programming guide*, Addison-Wesley, 1999.
- [146] Yepes J C, Yepes J J, Martinez J R, Pérez V Z, *Implementation of an Android based teleoperation application for controlling a KUKA-KR6 robot by using sensor fusion*, Health Care Exchanges PAHCE, Pan American, IEEE, pp. 1-5, 2013.
- [147] Zelenov S V, Zelenova S A, *Generation of positive and negative tests for parsers*, Programming and Computer Software, vol. 31, no. 6, pp. 310-320, 2005.
- [148] Zubrycki I, Granosik G, *Intuitive user interfaces for mobile manipulation tasks*, Journal of Automation Mobile Robotics & Intelligent Systems, vol. 9, no. 1, pp. 41-52, 2015.
- [149] Zurita G, Nussbaum M, *A conceptual framework based on activity theory for mobile CSCL*, British Journal of Educational Technology, vol. 38, no. 2, 2007, pp. 211-235.

БИОГРАФИЈА АУТОРА

Маја Лутовац Бандука (девојачко Лутовац), мастер инжењер електротехнике, рођена је 06.08.1985. године у Београду, Република Србија, од оца Мирослава и мајке Аните. Основну и средњу школу “Никола Тесла” завршила је у Београду као један од најбољих ученика.

Електротехнички факултет у Београду уписала је 2004/2005. школске године. Дипломирала је 2009. године са оценом 10 на дипломском. Школске 2009/2010. године уписала је мастер студије на Електротехничком факултету у Београду. Мастер студије је завршила 2011. године са просечном оценом 8.66 и оценом на мастер раду 10. Исте године, засновала је радни однос у Лола институту, у сектору Робот контролери, као истраживач приправник, где је запослена до данас, као истраживач сарадник.

Школске 2011/2012. године уписала је докторске студије на Електротехничком факултету у Београду, смер Софтверско инжењерство. На докторским студијама положила је све испите са просечном оценом 10. Област научног истраживања кандидаткиње обухвата развој софтверских система за индустријске роботе и уређаје за тренинг пилота и симулацију лета, развој језика за програмирање робота, развој Андроид апликација и 3Д графичких апликација. Рецензент је на међународним конференцијама *TELFOR (Telecommunications Forum)* и *MECO (Mediterranean Conference on Embedded Computing)*, као и међународног часописа *Turkish Journal of Electrical Engineering & Computer Sciences*.

Аутор је и коаутор преко 30 радова у научним часописима и на домаћим и страним конференцијама, од којих је преко 20 везано за тему докторске дисертације. Аутор је и коаутор четири рада са *impact* фактором (*SCI* листа) који су у директној вези са темом докторске дисертације, од тога два као самостални аутор.

Прилог 1.

Изјава о ауторству

Потписани-а МАЈА ЛУТОВАЦ БАНДУКА

број уписа 11/5067

Изјављујем

да је докторска дисертација под насловом

СОФТВЕРСКИ СИСТЕМ ЗА ДАВИНСКО УПРАВЉАЊЕ И
НАДЗОР РОБОТА БАЗИРАН НА АНДРОИД ОДЕРАТИВНОМ СИСТЕМУ
И БЕЖИЧНОЈ КОМУНИКАЦИЈИ

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 29. 12. 2016.

Маја Лутовац Бандука

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора МАСА ЛУТОВАЦ БАНДУКА

Број уписа 11/5067

Студијски програм СОФТВЕРСКО ИНЖИЊЕРСТВО

Наслов рада СОФТВЕРСКИ СИСТЕМ ЗА ДИГИТАЛНО УПРАВЉАЊЕ И НАДЗОР РОБОТА
БАЗИРАН НА АНДРОИД ОПЕРАТИВНИМ СИСТЕМУ И БЕЖИЧНОЈ КОМУНИКАЦИЈИ

Ментор ДР БОШКО НИКОЛИЋ

Потписани МАСА ЛУТОВАЦ БАНДУКА

изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу Дигиталног репозиторијума Универзитета у Београду.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 29.12.2016.

Маса Лутовац Бандука

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

СОФТВЕРСКИ СИСТЕМ ЗА ДАЉИНСКО УПРАВЉАЊЕ И НАДЗОР РОБОТА
БАЗИРАН НА АНДРОИД ОПЕРАТИВНОМ СИСТЕМУ И БЕЖИЧНОЈ КОМУНИКАЦИЈИ

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, 29.12.2016.

Дејан Милошевић

1. Ауторство - Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољавање умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.