

Универзитет у Београду  
Машински факултет

Бошко П. Цветковић

Пројектовање савремених система  
управљања робота применом развојних  
програмабилних система и савремене  
теорије рачуна нецелог реда

Докторска дисертација

Београд, 2021.

University of Belgrade  
Faculty of Mechanical Engineering

Boško P. Cvetković

Development of modern robot control  
systems using development programmable  
systems based on non-integer calculus  
theory

Doctoral dissertation

Belgrade, 2021.

**Комисија за преглед, оцену и одбрану докторске дисертације**

**Ментор:** др Михаило Лазаревић, редовни професор, ментор,  
Универзитет у Београду, Машински факултет

**Чланови комисије:** др Радиша Јовановић, редовни професор,  
Универзитет у Београду, Машински факултет  
др Немања Зорић, ванредни професор,  
Универзитет у Београду, Машински факултет  
др Петар Мандић, доцент,  
Универзитет у Београду, Машински факултет  
др Томислав Шекара, редовни професор,  
Универзитет у Београду, Електротехнички факултет

**Датум одбране:**

## Изрази захвалности

Желим да се захвалим ментору и професору Михаилу Лазаревићу на несебичној подршци током докторских студија као и томе што ме је, када сам неколико пута хтео да одустанем од докторских студија због пословних и приватних разлога, мотивисао да истрајем и довршим докторат и студије.

Хвала мојој супрузи на љубави, разумевању и без чије несебичне подршке, одрицања и мотивисања не бих завршио докторски рад.

Хвала мојим родитељима на љубави коју су ми пружили, на подршци, стрпљењу, на свему што су ме научили и како су ме васпитали.

Захваљујем колеги и доценту Петру Мандићу без чије помоћи и подршке не бих успео да приведем докторат крају.

Захваљујем професору Томиславу Шекари који је увек имао времена да одговори на питања и да идеју како решити одређени проблем.

Хвала професору Радиши Јовановићу на несебичној помоћи, који је увек налазио времена за мене, и захваљујући коме сам заволео аутоматско управљање.

Велику захвалност дугујем пријатељима из Института Михаило Пупин, и то: Владимиру Нешићу, Петру Марићу, Браниславу Шашићу, Милени Манојловић, Огњену Ристићу, Александру Павловићу, Жељку Деспотовићу без чије помоћи управљачки део робота не би био овако напредан.

Али оно што је најбитније од свега је то да захваљујући свима који су ми на неки начин помогли, све време су ми враћали веру у доброту и људе, а мене су учинили бољом особом. На томе вам свима неизмерно захваљујем!

# Пројектовање савремених система управљања работа применом развојних програмабилних система и савремене теорије рачуна нецелог реда.

## Резиме

У овој докторској дисертацији предмет истраживања је пројектовање савремених система управљања применом развојних програмабилних система и применом савремене теорије рачуна нецелог реда. Родригов приступ је примењен у циљу добијања одговарајућег математичког модела роботског система који је моделован као један повезан систем крутих тела који формирају кинематички ланац без гранања. За решавање проблема везаног кретања роботског система у случајевима када је остварен контакт хватаљке са радном површи или кретање по задатој линији, применом Лагранжевих једначина друге врсте у коваријантном облику добијене су диференцијалне једначине везаног кретања роботског система са хватаљком. Други приступ за добијање диференцијалних једначина везаног кретања роботског система је остварен применом Лагранж-Даламберовог принципа у генералисаним координатама. При томе, роботски системи који остварују контакт са радном површи могу се моделирати као сингуларни системи целог и у општем случају нецелог реда.

У оквиру савремене теорије управљања *итеративно управљање путем учења* (ИУУ- *iterative learning control*) представља једну моћну интелигентну методологију управљања који на итеративни начин побољшава понашање репетитивних процеса и динамичких система; а овде је од посебног интереса примена истог на роботске системе. Овде се предлаже један нови НРИУУ (ИУУ нецелог реда) закон  $PD^2D^{\alpha}$  типа у отвореној спрези за класу линеаризованих роботских система. Спроведена је *feedback* линеаризација објекта управљања где се у затвореној спрези добија линеарна зависност улаза и излаза. Даље се разматра проблематика управљања у отворено-затвореној спрези ИУУ нецелог реда сингуларним динамичким системом нецелог реда. Посебна пажња је посвећена и примени закона ИУУ нецелог реда типа  $PD^{\mu}/PD$  NeuroArm роботским системом. Након тога разматран је избор и имплементација хардвера потребног за управљање NeuroArm роботске руке и то тако што за развојну плочу је изабран Beaglebone Black. Пројектује се и израђује плоча за управљање моторима и читање сигнала са енкодера и потенциометара, бира се Linux Debian (Xenoma1 модификација) оперативни систем, што заједно чини комплетан ембедид систем (*embedded system*). Програмирају се напредни алгоритми управљања коришћењем EDICOPT софтверског алата путем FBD-а и ST кода, по први пут у области роботских система. Остварена је комуникација са NeuroArm роботском

руком путем Modbus протокола коришћењем софтвера који је креиран специјално за ту намену (NeuroArm manipulator), и који поред управљања самом роботском руком пружа могућност и логовања података ради анализе кретања делова роботске руке и упоређивања са симулацијама у Matlab-у и Simulink-у. На крају решавана је проблематика даљинског управљања роботом путем Интернета на NeuroArm роботској руци.

**Кључне речи:** роботски системи, Родригов приступ, рачун нецелог реда, фракциони ПИД регулатор, итеративно управљање путем учења, сингуларни системи, развојне плоче, Beaglebone, Modbus, програмирање.

**Научна област:** Машинство

**Ужа научна област:** Механика

# **Development of modern robot control systems using development programmable systems based on non-integer calculus theory.**

## **Abstract**

In this doctoral dissertation the subject of research is development of modern robot control systems using development programmable systems based on non-integer calculus theory. Rodriguez's approach was applied in order to obtain an appropriate mathematical model of a robotic system that was modeled as a single connected system of rigid bodies forming a kinematic chain without branching. To solve the problem of bound motion of the robotic system in cases when the gripper is in contact with the work surface or motion along a given line, by applying Lagrangian equations of second order in covariant form, differential equations of bound motion of the robotic system with the gripper are obtained. Another approach for obtaining differential equations of bound motion of a robotic system was realized by applying the Lagrange-Dalambert principle in generalized coordinates. In doing so, robotic systems that make contact with the work surface can be modeled as singular systems of the integer order and in the general case of the non-integer order. Within modern control theory, iterative learning control (ILC) is a powerful intelligent control methodology that iteratively improves the behavior of repetitive processes and dynamic systems; and here is of special interest to apply it to robotic systems. Here, a new FOILC (fractional order ILC) law of the  $PD^2D^\alpha$  type in open loop is proposed for a class of linear robotic systems. A feedback linearization of the control object was performed, where a linear dependence of inputs and outputs was obtained in a closed loop. The problem of control in open-closed loop of ILC of non-integer order using singular dynamic system using non-integer order is further considered. Special attention is given to the application of the ILC law of the non-integer order of  $PD^\mu/PD$  type for NeuroArm robotic system. After that, the selection and implementation of the hardware needed to control the NeuroArm robotic arm was discussed by selecting Beaglebone Black as the development board. A board for motor control and signal reading from encoders and potentiometers is designed and manufactured, the Linux Debian (Xenomai modification) operating system is selected, which together make a complete embedded system. Advanced control algorithms are programmed using the EDICOPT software tool via FBD and ST code, for the first time in the field of robotic systems. Communication with the NeuroArm robotic arm was achieved via the Modbus protocol using software created specifically for this purpose (NeuroArm manipulator), which in addition to controlling the robotic arm itself provides the ability to log data in order to analyze the movement of parts of the robotic arm and compare with

simulations in Matlab and Simulink. Finally, the problem of remote control of the robot via the Internet on the NeuroArm robot arm was solved.

**Keywords:** robotic systems, Rodriguez's approach, non-integer calculus, fractional PID controller, iterative learning control, singular systems, embedded systems, Beaglebone, Modbus, programming.

**Scientific discipline:** Mechanical engineering

**Scientific subdiscipline:** Mechanics



# Садржај

Увод .....	1
1.1. Преглед литературе .....	1
1.2. Кратак преглед докторске дисертације по поглављима .....	3
2. Везано кретање хватаљке роботског система .....	5
2.1. Математички модел роботског система.....	5
2.2. Диференцијалне једначине везаног кретања роботског система са хватаљком-применом Лагранжевих једначина друге врсте у коваријантном облику.....	7
2.3. Диференцијалне једначине везаног кретања роботског система са хватаљком-применом Лагранж-Даламберовог принципа у генералисаним координатама.....	12
2.4. Одређивање Јакобијана у случају везаног кретања роботске хватаљке .....	14
3. Итеративно управљање путем учења (ИУУ) роботских система .....	17
3.1. Основни појмови итеративног управљања учењем целог реда.....	17
3.2. Закон ИУУ типа $PD^2D^\alpha$ у отвореној спрези.....	20
3.2.1 <i>Feedback</i> линеаризација.....	22
3.2.2 Резултати конвергенције -закона учења типа $PD^2D^\alpha$ у отвореној спрези....	26
3.3. Итеративно управљање учењем у отворено-затвореној спрези типа нецелог реда једним сингуларним динамичким системом нецелог реда.....	32
3.3.1 Итеративно управљање путем учења нецелог реда за линеарне сингуларне системе нецелог реда .....	34
3.3.2 Анализа конвергенције датог алгоритма ИУУ нецелог реда.....	36
3.3.3 Симулациони резултати .....	41
3.4. Итеративно управљање учењем у отворено-затвореној спрези нецелог реда $PD^n/PD$ типа NeuroArm роботским системом.....	43
3.4.1. Математички модел NeuroArm роботског система .....	43
3.4.2 Резултати симулације .....	48
4. Хардверско управљање NeuroArm роботском руком .....	52
4.1. Ембедид системи .....	52
4.2. Избор развојне плоче за управљање NeuroArm роботском руком .....	53
4.3. Плоча за управљање моторима и читање вредности енкодера.....	56
4.4. Оперативни системи за рад у реалном времену са развојним плочама .....	58

4.5. Софтвер за програмирање NeuroArm роботске руке – EDICOPT.....	60
4.6. Имплементација новог контролера и програмирање.....	61
4.7. Тестирање MAXON 142750 мотора.....	63
5. Управљање NeuroArm роботском руком генерисано у програмском окружењу.....	66
5.1. MODBUS.....	66
5.1.1. Комуникациони протокол – основе.....	66
5.1.2. Фрејмови – порука.....	68
5.2. EasyMODBUS TCP библиотека.....	69
5.3. NeuroArm manipulator – софтверско решење за управљање NeuroArm роботском руком.....	69
5.4. Даљинско управљање роботским системом.....	77
5.4.1. Предности и мане даљинског управљања.....	77
5.4.2. Даљинско управљање NeuroArm роботском руком.....	79
5.4.3. Управљање путем мобилног телефона.....	80
5.4.4. Визуелна контрола рада робота.....	81
5.5. Управљање преко контролног панела.....	81
6. Закључак и научни доприноси докторске дисертације.....	82
Литература.....	84
Прилог.....	94
П1. Шематски приказ делова развијене плоче за управљање моторима.....	94
П2. Тест, модел Махон мотора у Matlab-у и Simulink-у и експериментални резултати .....	95
П3. Комуникација са NeuroArm роботском руком.....	103
Биографски подаци кандидата.....	106

# Увод

## 1.1. Преглед литературе

Роботика, као релативно млада и мултидисциплинарна област савремене технологије, захтева познавање електротехничког, машинског и системског инжењерства. Брзи развој роботике током последњих деценија углавном је проистекао из напретка рачунарске и сензорске технологије, као и теоријског напретка у теорији управљања [1]. Робота се највише примењују у индустријским условима, где испуњавају задатке као што су заваривање, сечење, паковање, наношење боје, померање предмета итд. Дакле, индустријски манипулатори морају да испуне високе захтеве у погледу тачности, прецизности и поновљивости.

Дизајнирање таквог контролера је комплексан задатак. Научна заједница известила је о великом броју различитих стратегија управљања роботским манипулаторима [2]. Већина њих је заснована на теорији линеарног управљања из практичних разлога. Линеарни контролери су интуитивни, лако разумљиви и лаки за примену. Овај приступ добија на значају када се динамика робота може сматрати линеарном, што се може постићи употребом мотора са високим преносним односима или употребом нелинеарних алгоритама управљања у циљу добијања линеарних динамичких система. На овај начин нелинеарна динамика роботског система се може занемарити и може се независно управљати сваким роботским сегментом. То је разлог зашто су класични ПИД регулатори и даље неизбежни у индустријском управљању роботских манипулатора [3], [4].

Поред класичних стратегија управљања, побољшање перформанси система се може постићи применом интелигентних техника управљања [5]. Једна од њих је итеративно управљање путем учења (ИУУ), која је недавно привукла пажњу научника као перспективно поље у управљању роботских система [6]. Историјски гледано, по први пут, 1978. године, експлицитну формулацију ИУУ-а је увео *Uchiyama* [7] и представио је на јапанском језику. Затим је, *Arimoto* [6] први предложио ИУУ управљање за прецизно праћење роботских трајекторија. Они су били у могућности да примене концепт ИУУ-а на индустријске роботске манипулаторе у обављању задатих манипулационих задатака који понављају исти задатак из више покушаја. Наиме, ИУУ покушава да опонаша људско учење коришћењем концепта покушаја и грешке, тј. научено из претходног покушаја користи се за прилагођавање управљачке величине за актуелни покушај. На тај начин се може итеративно побољшати одзив система током ограниченог временског интервала [8]. Међутим, ИУУ као једноставна техника преусмеравања не може ефикасно сузбити непознате поремећаје. Да би се то постигло, ИУУ се комбинује са класичним управљачким

системом са негативном повратном спрегом или, као у нашем случају, са ПД регулатором.

Појава и примена теорије фракционог рачуна (*fractional calculus*) (нецелог) реда<sup>1</sup> може се пратити уназад 300 година, а сада игра важну улогу у савременој науци и техници, посебно у области управљања динамичких система [9], [10]. Последњих деценија је све већа пажња посвећена развоју рачуна нецелог реда [11] и њиховој примени у разним научним и инжењерским областима [12], [13], [14], а по први пут примењени су алгоритми ИУУ управљања за одговарајућу класу класичних регуларних сингуларних система, [15], [16] и недавно су добијени резултати у примени алгоритама ИУУ управљања нецелог реда и то за регуларне системе нецелог реда датих у временском домену, [14], [17], [18], [19], [20], [21], [22], [23]. Такође, примена ИУУ-а на динамичке системе нецелог реда је недавно разматрана и уведена [24], [25], [8] при чему ИУУ нецелог реда не задржава само предности класичног ИУУ-а већ може се користити да додатно побољша перформансе контролног система. Интуитивно је јасно да су контролери нецелог реда флексибилнији у поређењу са онима целобројног реда. У основи, контролери нецелог реда (НР) са неколико подесивих параметара могу постићи исту робустност и перформансе као класични контролер високог реда. У литератури су неки од најчешћих НР контролера CRONE контролери [26],  $PI^{\lambda}D^{\mu}$  регулатор [27] и фракциони компензатор са фазним заостајањем или претхођењем [28].

За практичну имплементацију и реализацију сложених алгоритама управљања, потребно је узети у обзир карактеристике хардверског и софтверског дела управљачког система. Са развојем процесора са малом потрошњом енергије али са довољно процесорске снаге да обављају не само једноставније већ и доста захтевне задатке, као и потребом за аутоматизацијом у свим сферама живота појавила се потреба за малим, али довољно моћним развојним плочама [29], [30], [31], [32]. Данас постоји велики број производа за ове намене, али међу најпопуларнијим и најшире прихваћеним у програмерској заједници су Raspberry Pi, Arduino, Beaglebone Black. Са развојем самог хардвера дошло је и до развоја оперативних система који подржавају развојне плоче овог типа, тако да постоји више варијанти Linux оперативног система, као и специјалних верзија RTOS (Real Time Operating System) Linux-а и кернела који подржавају рад у реалном времену [33], [34], [35], [36]. Сваке године знатно напредује развој програмских језика C#, Java, Python, PHP, Javascript ... тако да се у неким програмским језицима као што је нпр. C# [37] могу писати програми за рад под различитим оперативним системима. Повезивање развојне плоче, плоче која управља компонентама типа мотори, чита податке са сензора и софтвера за управљање чини једну функционалну целину коју називамо ембедид систем (embedded system) [33], [34], [38], [39], [40], [41], [42]. Све више, због брзине развоја, смањења трошкова, једноставности употребе, у научним радовима разматра се

---

<sup>1</sup> Термин *рачун нецелог реда* користиће се надаље уместо термина *фракциони рачун*

коришћење ембедид система у роботизи, јер, као што је напоменуто, сваке године развијају се нови процесори који су бржи од претходних, а са њима и оперативни системи, тако да је могуће веома комплексне операције анализе информација из спољашње средине путем сензора примити, обрадити и креирати сложено управљање роботским системом [43], [44], [45], [46], [47], [48], [49].

## 1.2. Кратак преглед докторске дисертације по поглављима

Сада ћемо дати у кратким цртама преглед докторске тезе по поглављима. Дисертација садржи укупно шест поглавља и једно додатно поглавље.

У првом поглављу дат је преглед литературе и кратак преглед по поглављима.

Друго поглавље бави се проблематиком везаног кретања хваталке роботског система. Дат је математички модел роботског система, затим су добијене и приказане диференцијалне једначине везаног кретања роботског система применом Лагранжевих једначина друге врсте у коваријантном облику где су посебно одређене генералисане силе услед реакције веза која намеће радно окружење (површ, линија). Затим су применом Лагранж-Даламберовог принципа у генералисаним координатама такође презентоване и добијене диференцијалне једначине везаног кретања роботског система у облику Лагранжевих једначина друге врсте у коваријантном облику са Лагранжевим множитељима. На крају пажња је посвећена и одређивању Јакобијана при везаном кретању врха хваталке роботског система.

Треће поглавље бави се итеративним управљањем путем учења (ИУУ), и то прво основама рачуна целог реда, применом закона ИУУ  $PD^2D^\alpha$  у отвореној спрези, и ИУУ нецелог реда сингуларним динамичким системом нецелог реда са симулационим резултатима, као и ИУУ у отвореној-затвореној спрези нецелог реда  $PD^\alpha/PD$  типа NeuroArm роботском руком где је дат математички модел NeuroArm роботског система и резултати симулације.

Четврто поглавље бави се разматрањем избора и имплементације хардверских решења за управљање NeuroArm роботске руке. Постављени су задаци које NeuroArm роботска рука мора да извршава да би се на основу тога изабрала развојна плоча која може да испуни све те задатке. Дат је шематски приказ управљачке плоче која је развијена за потребе NeuroArm роботске руке са описом могућности, дато је објашњење зашто је изабран RTOS оперативни систем, приказ програмског алата EDICOPT за програмирање логике рада мотора, и на крају поглавља приказано је лабораторијско тестирање Махон мотора који је у склопу NeuroArm роботске руке.

У петом поглављу акценат је дат на комуникацији са NeuroArm роботском руком путем Modbus протокола, начинима управљања који су примењени и тестирани, предностима даљинске контроле, односно даљинског управљања и новим начинима контроле који тек треба да буду омогућени и примењени.

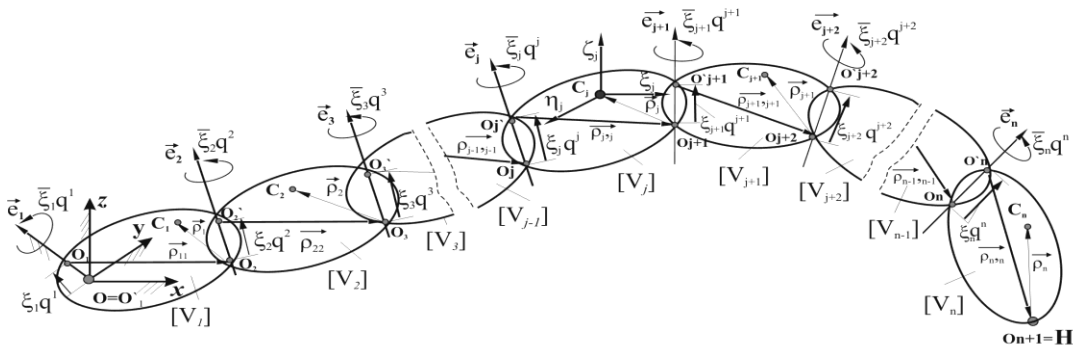
Шесто поглавље садржи закључке и научне доприносе дисертације.

У прилогу могу се наћи шематски приказ преосталих делова управљачке плоче, модел Махон мотора у Matlab-у и Simulink-у, као и један део кода за комуникацију са NeuroArm роботом и програмирањем рада мотора у EDICOPT програмском окружењу.

## 2. Везано кретање хваталке роботског система

### 2.1. Математички модел роботског система

Са механичке тачке гледишта, разматрани роботски систем овде можемо моделовати као један повезан систем крутих тела (сегмената) која формирају одговарајући кинематички ланац без гранања. На слици 2.1 је приказан роботски систем који се састоји од  $n+1$  роботских сегмената (први сегмент је непокретан-непокретно постоље) у виду отвореног кинематичког ланца без гранања који има  $n$  степени слободе. При томе, сегменти су међусобно повезани зглобно тако да два суседна сегмента формирају кинематички пар  $V$  класе, који омогућава само један степен слободе (СС).



Слика 2.1 Роботски систем дат у виду отвореног кинематичког ланца

Даље се уводе параметри  $\bar{\xi}_i = 1 - \xi_i$ ,  $\xi_i$  који се дефинишу на следећи начин. Имају вредности  $\xi_i = 1$ ,  $\bar{\xi}_i = 0$ , за случај да је дати зглоб призматични тј., ако зглоб дозвољава транслацију сегмента ( $V_i$ ) у односу на претходни сегмент ( $V_{i-1}$ ) и  $\xi_i = 0$ ,  $\bar{\xi}_i = 1$ , за случај да је дати зглоб ротациони. Конфигурација роботског система је одређена скупом генерализаним координатама  $q_1, q_2, \dots, q_n$ , при чему се уводи претпоставка да се систем налази у почетном тренутку у референтној конфигурацији, ( $q^i = 0, i = 1, 2, \dots, n$ ) [50]. Уводе се следећи координатни системи: инерцијални координатни систем  $Oxyz$  који је круто везан за непокретно постоље и покретни координатни системи  $C_i \xi_i \eta_i \zeta_i$ , који су смештени у центар инерције  $C_i$ , сегмента ( $V_i$ ) и круто везани за исти. Оријентација свих оса ротације и транслације је дефинисана јединичним векторима  $\bar{e}_i, i = 1, 2, \dots, n$ . Такође, уведени су следећи карактеристични вектори положаја  $\bar{\rho}_{ii}$  и  $\bar{\rho}_i$  који су дефинисани следећим изразима:  $\bar{\rho}_{ii} = \overrightarrow{O'_i O_{i+1}}$ ,  $\bar{\rho}_i = \overrightarrow{O'_{i+1} C_i}$  при чему  $\bar{\rho}_{ii} = \overrightarrow{O'_i O_{i+1}}$  одређује релативни положај зглобова

(између два суседна зглоба) у простору док вектор  $\vec{\rho}_i = \overrightarrow{O'_{i+1}C_i}$  се користи да се одреди положај центра инерције  $C_i, (V_i)$  сегмента. Ако се другачије не нагласи усваја се да се тачка  $O_{n+1}$  поклапа са врхом роботске хваталке  $H$ . Вектор положаја центра масе  $C_i, (V_i)$  сегмента је одређен следећим изразом, (види слику 2.1):

$$\vec{r}_{C_i} = \sum_{i=1}^i (\vec{\rho}_{ii} + \xi_i q^i \vec{e}_i) + \vec{\rho}_i \quad (2.1)$$

или

$$\{\vec{r}_{C_i}(q)\} = \begin{cases} x_i(q) \\ y_i(q) \\ z_i(q) \end{cases} = \sum_{i=1}^n \left( \prod_{k=1}^i [A_{k-1,k}] \right) \left( \{\rho_{ii}^{(i)}\} + \xi_i q^i \{e_i^{(i)}\} \right) + \prod_{k=1}^i [A_{k-1,k}] \{\rho_i^{(i)}\} \quad (2.2)$$

где  $[A_{k-1,k}]$  означава Родригову матрицу трансформације координата између координатних система  $(k-1)$ -тог и  $k$ -тог сегмента. Родригова матрица трансформације је дефинисана следећим изразом:

$$[A_{k-1,k}] = [I] + [e_k^{d(k)}]^2 (1 - \cos q^k) + [e_k^{d(k)}] \sin(q^k) \quad (2.3)$$

где је

$$\{e_k^{(k)}\} = (e_{\xi k}, e_{\eta k}, e_{\zeta k})^T, \quad [e_k^{d(k)}] = \begin{bmatrix} 0 & -e_{\zeta k} & e_{\eta k} \\ e_{\zeta k} & 0 & -e_{\xi k} \\ -e_{\eta k} & e_{\xi k} & 0 \end{bmatrix}. \quad (2.4)$$

Такође, могу се одредити брзина  $\vec{v}_{C_i}$  центра инерције  $C_i (V_i)$  сегмента, као и угаона брзина  $\vec{\omega}_i (V_i)$  сегмента дати са:

$$\vec{v}_{C_i} = \sum_{k=1}^i \frac{\partial \vec{r}_i}{\partial q^k} \dot{q}^k = \sum_{k=1}^n \vec{T}_{k(i)} \dot{q}^k \quad (2.5)$$

$$\vec{\omega}_i = \sum_{k=1}^i \vec{\xi}_k \vec{e}_k \dot{q}^k = \sum_{k=1}^n \vec{\Omega}_{k(i)} \dot{q}^k \quad (2.6)$$

где су са  $\dot{q}^k, k=1,2,\dots,n$  означене генерализане брзине и тзв. квазибазни вектори који су дефинисани са:

$$\frac{\partial \vec{r}_{C_i}}{\partial q^k} = \vec{T}_{k(i)}, \Rightarrow \vec{T}_{k(i)} = \begin{cases} \vec{\xi}_k \vec{e}_k \times \vec{R}_{k(i)} + \xi_k \vec{e}_k, & \forall k \leq i \\ 0, & \forall k > i \end{cases}, \quad (2.7)$$

$$\vec{R}_{k(i)} = \sum_{\alpha=k}^i (\vec{\rho}_{\alpha\alpha} + \xi_\alpha \vec{e}_\alpha q^\alpha) + \vec{\rho}_i, \quad (2.8)$$

као и  $\vec{\Omega}_{k(i)}$  дато са:

$$\frac{\partial \vec{\rho}_i}{\partial q^k} = \vec{\Omega}_{k(i)} \times \vec{\rho}_i, \quad \vec{\Omega}_{k(i)} = \begin{cases} \vec{\xi}_k \vec{e}_k, & \forall k \leq i, \\ 0, & \forall k > i, \end{cases} \quad (2.9)$$



Диференцијалне једначине кретања за дати роботски систем су дате у облику Лагранжевих једначина друге врсте као:

$$\sum_{\alpha=1}^n a_{\gamma\alpha}(q)\ddot{q}_\alpha + \sum_{\alpha=1}^n \sum_{\beta=1}^n \Gamma_{\alpha\beta,\gamma}(q)\dot{q}_\alpha\dot{q}_\beta = Q_\gamma \quad \gamma = 1, 2, \dots, n \quad (2.10)$$

Коефицијенти  $a_{\gamma\alpha} = a_{\alpha\gamma}$  представљају коваријантне координате основног метричког  $[a_{\gamma\alpha}] \in R^{n \times n}$  који су дефинисани са:

$$a_{\alpha\beta} = \sum_{i=1}^n \left( (\bar{\Omega}_{\alpha(i)}) [J_{C_i}] \{ \bar{\Omega}_{\beta(i)} \} + m_i (\bar{T}_{\alpha(i)}) \{ \bar{T}_{\beta(i)} \} \right) \quad (2.11)$$

$$a_{\alpha\beta} = \sum_{i=b}^n m_i (\bar{T}_{\alpha(i)}) \{ \bar{T}_{\beta(i)} \} + \sum_{i=b}^n \bar{\xi}_\alpha \bar{\xi}_\beta (\bar{e}_\alpha^{(i)}) [J_{C_i}] \{ \bar{e}_\beta^{(i)} \}$$

$$a = \inf(\alpha, \beta) = (\underline{\alpha}, \underline{\beta}), b = \sup(\alpha, \beta) = (\underline{\alpha}, \underline{\beta})$$

где  $m_i$  означава масу  $i$ -тог сегмента и тензора инерције  $J_{C_i}$ ,  $i$ -тог сегмента израчунате у односу на локални координатни систем  $C_i \xi_i \eta_i \zeta_i$ . Са  $\Gamma_{\alpha\beta,\gamma}$  је означен Кристофелов симбол прве врсте:

$$\Gamma_{\alpha\beta,\gamma} = \frac{1}{2} \left( \frac{\partial(a_{\beta\gamma})}{\partial q^\alpha} + \frac{\partial(a_{\gamma\alpha})}{\partial q^\beta} - \frac{\partial(a_{\alpha\beta})}{\partial q^\gamma} \right), \quad \alpha, \beta, \gamma = 1, \dots, n. \quad (2.12)$$

или

$$\Gamma_{\alpha\beta,\gamma} = \sum_{i=1}^n m_i \bar{\xi}_\alpha (\bar{e}_\alpha \times \bar{T}_{b(i)}) \{ \bar{T}_{\gamma(i)} \} + \sum_{i=1}^n (\bar{\Omega}_{b(i)} \times \bar{\Omega}_{\gamma(i)}) [\Pi_i] \{ \bar{\Omega}_{\alpha(i)} \} \quad (2.13)$$

$$a = \inf(\alpha, \beta) = (\underline{\alpha}, \underline{\beta}), b = \sup(\alpha, \beta) = (\underline{\alpha}, \underline{\beta}),$$

где  $[\Pi_i]$  означава одговарајући планарни тензор инерције  $i$ -тог сегмента. Генералисане силе  $Q_\gamma$  су представљене са десне стране израза (2.10) и детаљно дате у изразу (2.14) где су са  $Q_\gamma^m, Q_\gamma^g, Q_\gamma^c, Q_\gamma^v, Q_\gamma^f$ ,  $\gamma = 1, 2, \dots, n$  у општем случају означене генералисане управљачке силе, гравитационе силе, вискозне силе, силе у опругама, силе трења респективно:

$$Q_\gamma = Q_\gamma^m + Q_\gamma^g + Q_\gamma^v + Q_\gamma^c + Q_\gamma^f, \quad \gamma = 1, 2, \dots, n \quad (2.14)$$

## 2.2. Диференцијалне једначине везаног кретања роботског система са хватаљком- применом Лагранжевих једначина друге врсте у коваријантном облику

Овде ће бити од интереса размотрити контактни проблем роботског система тј. везаног кретања датог роботског система у случајевима када је остварен контакт хватаљке са радном површи у реализацији захтеваних задатака. Ако координате  $(x_H, y_H, z_H)$  врха  $(H)$  хватаљке роботског система задовољавају релације

$$f^v(x_H, y_H, z_H, t) = 0, \quad v = 1, 2 \quad (2.15)$$

тада хваталка врши ограничено кретање у радном простору роботског система. Такав случај имамо када се врх хваталке креће у општем случају по нестационарној површи (случај  $v=1$ ) и по линији ( $v=2$ , где је линија одређена пресеком двеју површи).

Вектор положаја врха хваталке  $\vec{r}_H$  у односу на непомићну тачку  $O$  може да се изрази у форми (види слику 2.1)

$$\vec{r}_H = \sum_{k=1}^n \xi_k \vec{e}_k q^k + \sum_{k=1}^{n-1} \vec{p}_{kk} + \overline{O'_n H} \quad (2.16)$$

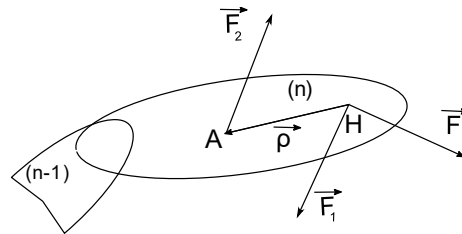
и ако уведемо следеће величине

$$\begin{aligned} \vec{p}_{kk} &= \vec{\rho}_{kk}, \quad (k=1, 2, \dots, n-1) \\ \vec{p}_{nn} &= \overline{O'_n H} \end{aligned} \quad (2.17)$$

можемо да пишемо израз за врх хваталке у следећем облику:

$$\vec{r}_H = \sum_{k=1}^n (\xi_k \vec{e}_k q^k + \vec{p}_{kk}) \quad (2.18)$$

Разматра се случај када на последњи сегмент роботског система делује сила  $\vec{F}$  (са нападном тачком у врху  $H$  хваталке) и спрег сила момента  $\vec{M}$ . Даље је потребно израчунати виртуални рад овог система сила. У том циљу прикажимо спрег силама  $\vec{F}_1$  и  $\vec{F}_2$  (види слику 2.2).



Слика 2.2. Спрег сила  $\vec{F}_1$  и  $\vec{F}_2$

При чему је сада

$$\vec{F}_1 = -\vec{F}_2, \quad \vec{M} = \vec{\rho} \times \vec{F}_2. \quad (2.19)$$

Пошто је,

$$\vec{r}_A = \overline{OA} = \vec{r}_H + \vec{\rho}, \quad (2.20)$$

следи да је

$$\delta \vec{r}_A = \delta \vec{r}_H + \delta \vec{p}, \quad (2.21)$$

Виртуални рад система сила  $(\vec{F}_1, \vec{F}_2, \vec{F})$ , има облик

$$\delta A = \vec{F} \cdot \delta \vec{r}_H + \vec{F}_1 \cdot \delta \vec{r}_H + \vec{F}_2 \cdot \delta \vec{r}_A, \quad (2.22)$$

или, узимајући у обзир (2.21)

$$\delta A = (\vec{F} + \vec{F}_1 + \vec{F}_2) \cdot \delta \vec{r}_H + \vec{F}_2 \cdot \delta \vec{p}. \quad (2.23)$$

Пошто је  $\vec{F}_1 + \vec{F}_2 = 0$  следи

$$\delta A = \vec{F} \cdot \delta \vec{r}_H + \vec{F}_2 \cdot \delta \vec{p}. \quad (2.24)$$

Узимајући у обзир да важи:

$$\vec{r}_H = \vec{r}_H(q^1, q^2, \dots, q^n), \quad \vec{p} = \vec{p}(q^1, q^2, \dots, q^n) \quad (2.25)$$

и

$$\delta \vec{r}_H = \sum_{i=1}^n \frac{\partial \vec{r}_H}{\partial q^i} \delta q^i, \quad \delta \vec{p} = \sum_{i=1}^n \frac{\partial \vec{p}}{\partial q^i} \delta q^i, \quad (2.26)$$

добија се

$$\delta A = \sum_{i=1}^n \left( \vec{F} \cdot \frac{\partial \vec{r}_H}{\partial q^i} + \vec{F}_2 \cdot \frac{\partial \vec{p}}{\partial q^i} \right) \delta q^i, \quad (2.27)$$

Генералисана сила  $Q_i^*$  која одговара овом систему сила износи:

$$Q_i^* = \vec{F} \cdot \frac{\partial \vec{r}_H}{\partial q^i} + \vec{F}_2 \cdot \frac{\partial \vec{p}}{\partial q^i}, \quad i = 1, 2, \dots, n \quad (2.28)$$

Пошто је

$$\begin{aligned} \frac{\partial \vec{r}_H}{\partial q^i} &= \bar{\xi}_i \bar{e}_i \times \left[ \sum_{k=i}^n (\bar{p}_{kk} + \xi_k q^k \bar{e}_k) \right] \\ \frac{\partial \vec{p}}{\partial q^i} &= \bar{\xi}_i \bar{e}_i \times \vec{p} \end{aligned} \quad (2.29)$$

ако уведемо ознаку

$$\bar{P}_{i(n)} = \frac{\partial \vec{r}_H}{\partial q^i}, \quad (2.30)$$

добиајамо

$$Q_i^* = \vec{F} \cdot \vec{P}_{i(n)} + \bar{\xi}_i \bar{e}_i \cdot (\bar{\rho} \times \vec{F}_2), \quad (2.31)$$

и, узимајући у обзир да је  $\vec{M} = \bar{\rho} \times \vec{F}_2$ , (види 2.19), коначно следи

$$Q_i^* = (\vec{F}) \cdot \{\vec{P}_{i(n)}\} + \bar{\xi}_i (\vec{M}) \cdot \{\bar{e}_i\}. \quad (2.32)$$

Ако посматрамо везано кретање хваталке, и ако везе (2.29) уклонимо у тачки  $H$ , делује реакције везе  $\vec{R}$  и пошто је у овом случају  $\vec{M} = 0$  следи:

$$Q_i^* = Q_i^{rh} = (\vec{R}) \cdot \{\vec{P}_{i(n)}\}. \quad (2.33)$$

Диференцијалне једначине кретања овако ослобођеног роботског система имају следећи облик:

$$\sum_{j=1}^n a_{ji} \ddot{q}^j + \sum_{j=1}^n \sum_{k=1}^n \Gamma_{jk,i} \dot{q}^j \dot{q}^k = Q_i^a + Q_i^g + Q_i^c + Q_i^b + Q_i^{rh}, \quad (2.34)$$

$i = 1, 2, \dots, n$

и разликују се од раније добијених једначина само по присуству члана  $Q_i^{rh}$  који је последица везаног кретања хваталке. Познато је, у случају веза (2.29), да реакција идеалне везе има облик:

$$\vec{R} = \lambda_1 \text{grad} f^1 + \lambda_2 \text{grad} f^2 \quad (2.35)$$

или

$$\vec{R} = \left( \lambda_1 \frac{\partial f^1}{\partial x_H} + \lambda_2 \frac{\partial f^2}{\partial x_H} \right) \vec{i} + \left( \lambda_1 \frac{\partial f^1}{\partial y_H} + \lambda_2 \frac{\partial f^2}{\partial y_H} \right) \vec{j} + \left( \lambda_1 \frac{\partial f^1}{\partial z_H} + \lambda_2 \frac{\partial f^2}{\partial z_H} \right) \vec{k} \quad (2.36)$$

Решење директног задатка кинематике има облик (за случај када не постоје везе (2.15))

$$x_H = \varphi_1(q^1, q^2, \dots, q^n), \quad y_H = \varphi_2(q^1, q^2, \dots, q^n), \quad z_H = \varphi_3(q^1, q^2, \dots, q^n) \quad (2.37)$$

тако да је

$$\left(\frac{\partial f^1}{\partial x_H}\right)_{(q^1, q^2, \dots, q^n)} = \left(\frac{\partial f^1}{\partial x_H}\right)_{(x_H=\varphi_1, y_H=\varphi_2, z_H=\varphi_3)} = g_1(q^1, q^2, \dots, q^n)$$

.....

$$\left(\frac{\partial f^2}{\partial z_H}\right)_{(q^1, q^2, \dots, q^n)} = \left(\frac{\partial f^2}{\partial z_H}\right)_{(x_H=\varphi_1, y_H=\varphi_2, z_H=\varphi_3)} = g_6(q^1, q^2, \dots, q^n)$$

и

$$\left\{\bar{R}^{(o)}\right\} = \left\{\begin{matrix} R_x \\ R_y \\ R_z \end{matrix}\right\}, \quad (2.39)$$

где је

$$\begin{aligned} R_x &= \lambda_1 g_1 + \lambda_2 g_2, \\ R_y &= \lambda_1 g_3 + \lambda_2 g_4, \\ R_z &= \lambda_1 g_5 + \lambda_2 g_6, \end{aligned} \quad (2.40)$$

Релација (2.33) сада добија облик

$$Q_i^{rh} = \left(\bar{R}^{(o)}\right) \cdot \left\{\bar{P}_{i(n)}^{(o)}\right\}. \quad (2.41)$$

где је

$$\left\{\bar{P}_{i(n)}^{(o)}\right\} = \left\{\begin{matrix} P_{i(n)x} \\ P_{i(n)y} \\ P_{i(n)z} \end{matrix}\right\}. \quad (2.42)$$

Очигледно је да је

$$Q_i^{rh} = \lambda_1 \Phi_{1i}(q^1, q^2, \dots, q^n) + \lambda_2 \Phi_{2i}(q^1, q^2, \dots, q^n), \quad (2.43)$$

где је

$$\begin{aligned} \Phi_{1i} &= g_1 \cdot P_{i(n)x} + g_3 \cdot P_{i(n)y} + g_5 \cdot P_{i(n)z}, \\ \Phi_{2i} &= g_2 \cdot P_{i(n)x} + g_4 \cdot P_{i(n)y} + g_6 \cdot P_{i(n)z}. \end{aligned} \quad (2.44)$$

Диференцијалне једначине (2.20) сада добијају облик у општем случају

$$\sum_{j=1}^n a_{ij} \ddot{q}^j + \sum_{j=1}^n \sum_{k=1}^n \Gamma_{jk,i} \dot{q}^j \dot{q}^k = Q_i^a(q, \dot{q}, t) + Q_i^s(q) + Q_i^c(q) + Q_i^b(\dot{q}) + \sum_{v=1}^l \lambda_v \Phi_{vi}(q) \quad (2.45)$$

$(i = 1, 2, \dots, n)$

односно за случај  $l = 2$

$$\sum_{j=1}^n a_{ij} \ddot{q}^j + \sum_{j=1}^n \sum_{k=1}^n \Gamma_{jk,i} \dot{q}^j \dot{q}^k = Q_i^a(q, \dot{q}, t) + Q_i^s(q) + Q_i^c(q) + Q_i^b(\dot{q}) + \lambda_1 \Phi_{1i}(q) + \lambda_2 \Phi_{2i}(q) \quad (2.46)$$

$(i = 1, 2, \dots, n)$

и, очигледно садрже  $n + 2$  променљиве величине  $(q^1, q^2, \dots, q^n, \lambda_1, \lambda_2)$ . Једначинама којих има  $n$ , потребно је придружити једначине веза (2.15) у којима је извршена трансформација (2.37). Наиме, након трансформације везе (2.15) добијају облик:

$$f^v(x_H = \varphi_1, y_H = \varphi_2, z_H = \varphi_3, t) = 0, \quad v = 1, 2 \quad (2.47)$$

или

$$f^v(q^1, q^2, \dots, q^n, t) = 0, \quad v = 1, 2 \quad (2.48)$$

Скуп (2.45) и (2.48), потпуно решавају проблем везаног кретања хватаљке јер сачињавају потпун скуп диференцијалних и алгебарских једначина за одређивање непознатих функција  $q^i = q^i(t)$ ,  $i = 1, 2, \dots, n$  и  $\lambda_v = \lambda_v(t)$ ,  $v = 1, 2$ .

### 2.3. Диференцијалне једначине везаног кретања роботског система са хватаљком- применом Лагранж-Даламберовог принципа у генералисаним координатама

Претходно изведене диференцијалне једначине кретања (2.46) могу се добити и применом Лагранж-Даламберовог принципа у генералисаним координатама, [50]:

$$\sum_{i=1}^n \left( \frac{d}{dt} \frac{\partial E_k}{\partial \dot{q}^i} - \frac{\partial E_k}{\partial q^i} - Q_i \right) \delta q^i = 0 \quad (2.49)$$

где је  $E_k$  кинетичка енергија разматраног система дата следећим изразом:

$$E_k = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \dot{q}^i \dot{q}^j, \quad (2.50)$$

и где су са  $\delta q^i$  означене изохроне варијације генералисаних координата  $\delta q^1, \dots, \delta q^n$ . Даље је од интереса користити коваријантни облик израза (2.49) добија се:

$$\sum_{i=1}^n \left( \sum_{j=1}^n a_{ij}(q) \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,i}(q) \dot{q}^j \dot{q}^k - Q_i(q, \dot{q}, t) \right) \delta q^i = 0. \quad (2.51)$$

Варирајмо, надаље, везе (2.48), које су независне тј. важи:

$$\text{rank} \left[ \frac{\partial f^v}{\partial q^i} \right] = l, \quad i = 1, \dots, n \quad (2.52)$$

Добија се у овом случају:

$$\sum_{i=1}^n \frac{\partial f^v}{\partial q^i} \delta q^i = 0, \quad (2.53)$$

где се након множења са скаларним функцијама  $\lambda_\nu = \lambda_\nu(t)$  које су познате под називом Лагранжеви множитељи веза добија следећи израз:

$$\lambda_\nu \sum_{i=1}^n \frac{\partial f^v}{\partial q^i} \delta q^i = 0, \quad \nu = 1, \dots, l \quad (2.54)$$

тј.

$$\sum_{\nu=1}^l \lambda_\nu \sum_{i=1}^n \frac{\partial f^v}{\partial q^i} \delta q^i = 0, \quad \Rightarrow \sum_{i=1}^n \left( \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^v}{\partial q^i} \right) \delta q^i = 0, \quad (2.55)$$

Сабирањем израза (2.55) са (2.51) коначно се добија:

$$\sum_{i=1}^n \left( \sum_{j=1}^n a_{ij}(q) \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,i}(q) \dot{q}^j \dot{q}^k - Q_i(q, \dot{q}, t) + \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^v}{\partial q^i} \right) \delta q^i = 0. \quad (2.56)$$

Како све варијације  $\delta q^1, \dots, \delta q^n$  нису независне, тј. постоји  $l$ -зависних, претходни израз се може написати и у следећем облику:

$$\begin{aligned} & \sum_{\alpha=1}^m \left( \sum_{j=1}^n a_{\alpha j} \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,\alpha} \dot{q}^j \dot{q}^k - Q_\alpha + \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^v}{\partial q^\alpha} \right) \delta q^\alpha + \\ & + \sum_{\pi=m+1}^{m+l=n} \left( \sum_{j=1}^n a_{\pi j} \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,\pi} \dot{q}^j \dot{q}^k - Q_\pi + \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^v}{\partial q^\pi} \right) \delta q^\pi = 0 \end{aligned} \quad (2.57)$$

Наиме, како је испуњен услов (2.52), следи да из (2.53) можемо одредити  $l$  зависних варијација генералисаних координата као линеарну комбинацију преосталих  $m$  независних варијација генералисаних координата. Не умањујући општост разматрања усвојићемо да су независне варијације  $\delta q^1, \dots, \delta q^m$  а зависне  $\delta q^{m+1}, \dots, \delta q^n$  (ако то није испуњено, може се увек прибећи пренумерацији координата до испуњења тога услова). Сада смо у могућности да одаберемо  $\lambda_\nu = \lambda_\nu(t)$  ако израз у другој загради изједначимо са нулом, где је

$$\begin{aligned}\mathcal{L}_\pi &= \sum_{j=1}^n a_{\pi j} \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,\pi} \dot{q}^j \dot{q}^k - Q_\pi \\ \mathcal{L}_\pi + \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^\nu}{\partial q^\pi} &= 0, \quad \pi = 1, \dots, l \Rightarrow \{\mathcal{L}_\pi\} = - \left[ \frac{\partial f^\nu}{\partial q^\pi} \right] \{\lambda_\nu\}\end{aligned}\tag{2.58}$$

одатле се добија, због независности веза, ( $\det \left[ \frac{\partial f^\nu}{\partial q^\pi} \right] \neq 0$ ), следи:

$$\{\lambda_\nu\} = - \left[ \frac{\partial f^\nu}{\partial q^\pi} \right]^{-1} \{\mathcal{L}_\pi\}\tag{2.59}$$

На тај начин у (2.57) остаје:

$$\sum_{\alpha=1}^m \left( \sum_{j=1}^n a_{\alpha j} \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,\alpha} \dot{q}^j \dot{q}^k - Q_\alpha + \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^\nu}{\partial q^\alpha} \right) \delta q^\alpha = 0.\tag{2.60}$$

одакле, због независности  $\delta q^\alpha$  следи:

$$\sum_{j=1}^n a_{\alpha j} \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,\alpha} \dot{q}^j \dot{q}^k - Q_\alpha + \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^\nu}{\partial q^\alpha} = 0, \quad \alpha = 1, 2, \dots, m.\tag{2.61}$$

Претходне једначине и узимајући у обзир и једначине (2.58) доводе до Лагранжевих једначина са множитељима веза у проширеном облику, [51]:

$$\sum_{j=1}^n a_{ij}(q) \ddot{q}^j + \sum_{k=1}^n \sum_{j=1}^n \Gamma_{jk,i}(q) \dot{q}^j \dot{q}^k = Q_i(q, \dot{q}, t) - \sum_{\nu=1}^l \lambda_\nu \frac{\partial f^\nu}{\partial q^i}, \quad i = 1, 2, \dots, n\tag{2.62}$$

где заједно са (2.48) формирају потпун скуп диференцијално-алгебарских једначина које сада нам омогућавају одређивање непознатих функција  $q^i = q^i(t)$  и  $\lambda_\nu = \lambda_\nu(t)$ .

## 2.4. Одређивање Јакобијана у случају везаног кретања роботске хваталке

У неким случајевима потребно је одредити и Јакобијан у случају везаног кретања роботске хваталке. Претпоставимо да је решен директан кинематски задатак у општем случају, (везе (2.15) одсуствују).



$$\begin{aligned}
\bar{q}^1 &= x_H, \bar{q}^2 = y_H, \bar{q}^3 = z_H, \bar{q}^4 = \psi_H, \bar{q}^5 = \theta_H, \bar{q}^6 = \phi_H, \\
\bar{q}^1 &= \varphi^1(q^1, q^2, \dots, q^n), \\
\bar{q}^2 &= \varphi^2(q^1, q^2, \dots, q^n) \quad \Rightarrow \quad \bar{q}^\alpha = \varphi^\alpha(q^1, q^2, \dots, q^n), \quad \alpha = 1, 2, \dots, m \quad (2.63) \\
&\dots\dots\dots \\
\bar{q}^m &= \varphi^m(q^1, q^2, \dots, q^n), \quad \max m = 6
\end{aligned}$$

Диференцирањем претходног израза (2.63) по времену следи:

$$\dot{\bar{q}}^\alpha(t) = \sum_{i=1}^n \frac{\partial \varphi^\alpha}{\partial q^i} \dot{q}^i(t), \quad \alpha = 1, 2, \dots, m \quad (2.64)$$

тј. може се дефинисати Јакобијан матрица  $[J]$  са:

$$\{\dot{\bar{q}}\} = [J]\{\dot{q}\}, \quad [J] = \begin{bmatrix} \frac{\partial \varphi^\alpha}{\partial q^i} \end{bmatrix} \in R^{m \times n} \quad (2.65)$$

Међутим, у случају везаног кретања роботске хваталке, скуп  $(q^1, q^2, \dots, q^n)$ , везан је релацијама (2.48) одакле је:

$$\sum_{i=1}^n \frac{\partial f^\nu}{\partial q^i} \dot{q}^i + \frac{\partial f^\nu}{\partial t} = 0, \quad \nu = 1, 2, \dots, l \quad (2.66)$$

$$\sum_{\beta=1}^m \frac{\partial f^\nu}{\partial q^\beta} \dot{q}^\beta + \sum_{\rho=m+1}^{m+l} \frac{\partial f^\nu}{\partial q^\rho} \dot{q}^\rho + \frac{\partial f^\nu}{\partial t} = 0, \quad \nu = 1, 2, \dots, l \quad (2.67)$$

због независности веза

$$\text{rang} \left[ \frac{\partial f^\nu}{\partial q^\rho} \right] = l \quad (2.68)$$

Множењем са инверзном матрицом (2.68) израз (2.67)

$$\left[ \frac{\partial f^\nu}{\partial q^\beta} \right] \{\dot{q}^\beta\} + \left[ \frac{\partial f^\nu}{\partial q^\rho} \right] \{\dot{q}^\rho\} + \frac{\partial f^\nu}{\partial t} = 0, \quad / \left[ \frac{\partial f^\nu}{\partial q^\rho} \right]^{-1} \quad (2.69)$$

добиа се:

$$\{\dot{q}^\rho\} = - \underbrace{\left[ \frac{\partial f^\nu}{\partial q^\rho} \right]^{-1} \left[ \frac{\partial f^\nu}{\partial q^\beta} \right]}_{[D_\beta^\rho]} \{\dot{q}^\beta\} + \underbrace{\left[ \frac{\partial f^\nu}{\partial q^\rho} \right]^{-1} \frac{\partial f^\nu}{\partial t}}_{\{E^\rho\}} \quad (2.70)$$

тј.

$$\begin{aligned}\{\dot{q}^\rho\} &= [D_\beta^\rho] \{\dot{q}^\beta\} + \{E^\rho\} \\ \dot{q}^\rho &= \sum_{\beta=1}^m D_\beta^\rho \dot{q}^\beta + E^\rho\end{aligned}\quad (2.71)$$

С друге стране, израз (2.65) се може приказати и на следећи начин:

$$\dot{q}^\alpha = \sum_{\beta=1}^m \frac{\partial \varphi^\alpha}{\partial q^\beta} \dot{q}^\beta + \sum_{\rho=m+1}^{m+l} \frac{\partial \varphi^\alpha}{\partial q^\rho} \dot{q}^\rho, \quad \alpha = 1, 2, \dots, m \quad (2.72)$$

Сменом израза (2.71) у (2.72) добија се:

$$\dot{q}^\alpha = \sum_{\beta=1}^m \frac{\partial \varphi^\alpha}{\partial q^\beta} \dot{q}^\beta + \sum_{\rho=m+1}^{m+l} \frac{\partial \varphi^\alpha}{\partial q^\rho} \left( \sum_{\beta=1}^m D_\beta^\rho \dot{q}^\beta + E^\rho \right), \quad \alpha = 1, 2, \dots, m \quad (2.73)$$

$$\dot{q}^\alpha = \sum_{\beta=1}^m \left( \frac{\partial \varphi^\alpha}{\partial q^\beta} + \sum_{\rho=m+1}^{m+l} \frac{\partial \varphi^\alpha}{\partial q^\rho} D_\beta^\rho \right) \dot{q}^\beta + \left( \sum_{\rho=m+1}^{m+l} \frac{\partial \varphi^\alpha}{\partial q^\rho} E^\rho \right), \quad \alpha = 1, 2, \dots, m \quad (2.74)$$

У случају стационарних веза  $E^\rho \equiv 0$  односно (2.74) поприма следећи облик:

$$\{\dot{q}^\alpha\}_{p \times 1} = [J^*]_{p \times m} \{\dot{q}^\beta\}_{m \times 1}, \quad \alpha = 1, 2, \dots, p \quad \beta = 1, 2, \dots, m \quad (2.75)$$

где је сада Јакобијан  $[J^*]$  за везано кретање врха хваталјке

$$[J^*] = \left[ \left( \frac{\partial \varphi^\alpha}{\partial q^\beta} + \sum_{\rho=m+1}^{n=m+l} \frac{\partial \varphi^\alpha}{\partial q^\rho} D_\beta^\rho \right) \right], \quad \alpha = 1, 2, \dots, p \quad \beta = 1, 2, \dots, m \quad (2.76)$$

или у општем случају нестационарних веза једнак

$$[J^*] = \left[ \left( \frac{\partial \varphi^\alpha}{\partial q^\beta} + \sum_{\rho=m+1}^{n=m+l} \left( \frac{\partial \varphi^\alpha}{\partial q^\rho} D_\beta^\rho + \frac{\partial \varphi^\alpha}{\partial q^\rho} E^\rho \right) \right) \right], \quad \alpha = 1, 2, \dots, p \quad \beta = 1, 2, \dots, m \quad (2.77)$$

чиме је решен проблем Јакобијана за везано кретање врха хваталјке.

## 3. Итеративно управљање путем учења (ИУУ) роботских система

### 3.1. Основни појмови итеративног управљања учењем целог реда

У оквиру савремене теорије управљања *итеративно управљање путем учења* (ИУУ- (*ILC*) *iterative learning control*) представља једну моћну интелигентну методологију управљања који на итеративни начин побољшава понашање репетитивних процеса, [52], [53], [54]. ИУУ омогућава да се оствари побољшање непостојаних перформанси репетитивних система и то на једном коначном временском интервалу. Наиме, ИУУ су посебно погодни за примену репетитивних динамичких система, на пример као што су хемијски реактор и роботски систем у производном окружењу или на динамичке системе са периодичним кретањем/ поремећајима, где је периодичност сада функција од времена, стања или трајекторије система.

Историјски гледано, по први пут, 1978. године, експлицитну формулацију ИУУ-а је увео *Uchiyama* [7] и представио је на јапанском језику. Затим је, *Arimoto* [6] први предложио ИУУ управљање за прецизно праћење роботских трајекторија. Они су били у могућности да примене концепт ИУУ-а на индустријске роботске манипулаторе у обављању задатих манипулационих задатака који понављају исти задатак из више покушаја. Аримото је искористио поновљивост извођења манипулационог задатка као и информације из претходних покушаја и перформанси самог задатка у циљу побољшања самог управљања током текућег покушаја и ову методу је он назвао "*betterment process*". Наиме, уочено је да код извођења поновљивих манипулационих задатака који се састоје у понављању једне исте трајекторије, применом ИУУ-а могуће је побољшати динамичке и кинематичке карактеристике разматраног роботског система као што су на пример, прецизност извођења задате трајекторије као и брзина праћења трајекторије.

Са друге стране, познато је да постојећа теорија управљања садржи бројне алате за побољшање перформанси динамичког система. Међутим, ако постоји немоделирана динамика или параметарске неизвесности у динамичком систему није увек могуће остварити захтеване перформансе затвореног система управљања. На пример, коришћењем алгоритама и концепата класичне теорије управљања тешко је оствариво захтевано праћење трајекторија репетитивних динамичких система јер није могуће искористити особину поновљивости у датим алгоритмима управљања. За разлику од класичних алгоритама управљања, ИУУ представља један ефикасни а истовремено и релативно једноставни алгоритам управљања који сада омогућава да се прогресивно смањује грешка праћења и истовремено омогућава побољшање перформансе затвореног система управљања из итерације у итерацију, [55], [56]. Чак шта више, ИУУ може омогућити и савршено праћење и за случајеве

када имамо модел динамичког система са нелинеарношћу, неизвесностима или делимично непознатим параметрима, или непотпуним информацијама о структури самог система, [54]). Уочено је да сам концепт ИУУ представља у суштини један биолошки аналогон процеса учења који карактерише интелигентна бића која имају тенденцију да уче током живота изводећи разна функционална кретања (тј. одабирају управљање) меморишући крајњи исход таквог избора управљања. Такав механизам омогућава предикцију наредних покрета/кретања датог бића која је заснована коришћењем претходног знања и искуства.

Према томе, може се рећи да је основна разлика између ИУУ од других постојећих управљања огледа у чињеници да сам процес учења представља у суштини један *off-line* процес где постоји коначно време између два покушаја. На тај начин итеративно управљање путем учења представља управљање које се генерише у директној ("*feedforward*") грани, мада се користи и управљање у повратној грани. Важно је истакнути да доминантну улогу представља управљање у директној грани јер оно омогућава да се оствари *предикција* будућег управљања, где је предикција последица упамћеног знања стеченог учењем.

Основна одлика алгоритма ИУУ-а је да се коришћењем информације о вектору управљања и о грешци праћења жељене излазне трајекторије из претходне итерације-покушаја одређује вектор управљања у текућој итерацији. На тај начин, процес учења је заснован на информацији о претходном покушају и стеченом искуству. То се може прецизније исказати на следећи начин. Нека је за дати динамички систем позната и задата жељена трајекторија  $y_d(t)$  у простору излаза тако да је грешка праћења у  $i$ -тој итерацији дефинисана са:

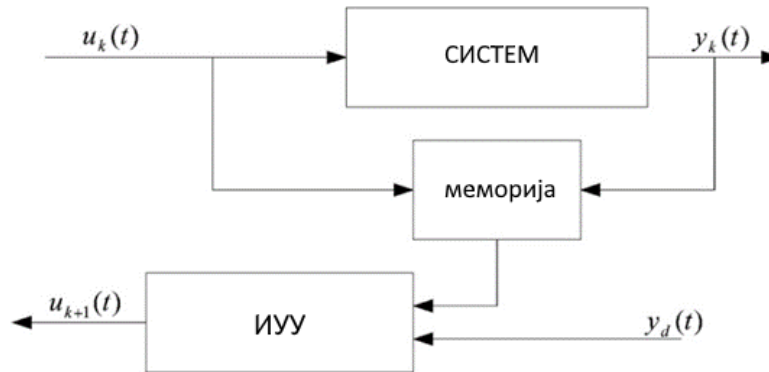
$$e_i(t) = y_d(t) - y_i(t) \quad (3.1)$$

где је са  $y_i(t)$  означен вектор излаза уоченог система и  $t \in [0, T]$  где је време извођења покрета  $T$  коначно и познато. Задатак управљања је да се применом концепта ИУУ за познату жељену излазну трајекторију  $y_d(t)$ , одреди вектор управљања  $u_i(t)$  такав да када број итерација  $i \rightarrow \infty$ ,  $\lim_{i \rightarrow \infty} u_i(t) = u_z(t)$ ,  $\forall t \in [0, T]$ , вектор излаза система  $y_i(t)$  прати жељену излазну трајекторију са задатим степеном тачности, тј.  $\lim_{i \rightarrow \infty} \|e_i(t)\| = \varepsilon_e$ . Тако, [6], је у циљу управљања датим роботским манипулатором предложио следећи тип ИУУ управљања тзв. Д-тип ИУУ у временском домену који захтева познавање првог извода сигнала грешке  $\dot{e}_i(t)$ :

$$\mathbf{u}_{i+1}(t) = \mathbf{u}_i(t) + \Pi \frac{d\mathbf{e}_i(t)}{dt}, \quad (3.2)$$

где  $i$  - означава број итерација, док  $e(t)$  представља грешку праћења (5.1);  $\Pi$  је тзв. појачање учења и одређује се на основу претходног знања о систему.

Основна шема ИУУ је приказана на сл.3.1 где су  $u_k(t)$  односно  $y_k(t)$  улаз и излаз система код  $k$ -те итерације,  $u_{k+1}(t)$  је улаз система код  $k+1$ -тог покушаја, а  $y_d(t)$  је дата жељена трајекторија. Циљ ИУУ је да је  $\lim_{k \rightarrow \infty} y_k(t) = y_d(t)$  за свако  $t \in [0, T]$ , где је  $T$  позната константа.



Сл. 3.1 Основна шема итеративног управљања путем учења

На основу чињенице како се користи сигнал грешке праћења претходне итерације за формирање сигнала управљања у текућој итерацији, ИУУ шеме се могу класификовати као Д-типа, П-типа, ПД-типа, и ПИД типа. На пример, алгоритам ИУУ-а ПИД тип је дат следећим изразом, [52]:

$$u_{i+1} = u_i + \Phi e_i + \Gamma \dot{e}_i + \Psi \int e_i dt \quad (3.3)$$

где су  $\Phi$ ,  $\Gamma$  и  $\Psi$  матрице појачања учења. Исто тако, ИУУ можемо поделити на временски континуалне и дискретне алгоритме ИУУ. Слично томе, временски променљиви алгоритам ИУУ П типа се може написати као

$$u_{k+1}(t) = u_k(t) + \Gamma_k(t)(y_d(t) - y_k(t)) \quad (3.4)$$

где је  $\Gamma_k(t)$  пропорционална матрица појачања учења која је сада временски променљива где се правилним избором матрице  $\Gamma_k(t)$ , ИУУ процес ће за системе релативног нултог степена тежити нултој грешци стационарног стања. Слични резултати се могу очекивати и установити за динамичке системе ненултог релативног степена.

У литератури се могу наћи и комплекснији алгоритми учења (на пример, види [56], [57]). Такође, могу се користити и алгоритми ИУУ-а који користе више изводе и које могу остварити боље перформансе у решавању задатка праћења жељене трајекторије система, [17].

$$u_{i+1}(t) = u_i(t) + \Gamma CB \frac{du_i(t)}{dt} + \Gamma \left( \frac{d^2 e_i(t)}{dt^2} + Q \frac{de_i(t)}{dt} + R e_i(t) \right) \quad (3.5)$$

Такође, комбинацијом контролера у повратној грани са ИУУ-ом у директној грани се добијају добре перформансе за праћење жељене путање и такође се убрзава процес конвергирања. Имајући у виду једноставност и ефикасност учења ИУУ алгоритма, као и његову способност да решава проблеме као што су нелинеарности и неодређености при моделовању овде ће бити од интереса њихова примена за решавање *задатка управљања роботских система*.

Са друге стране, многи класични ИУУ алгоритми имају ограничење које укључује претпоставку да матрица спрезања [CB] линеаризованог роботског система у простору стања има ранг пуне колоне где В означава контролну матрицу, а С означава излазну матрицу у простору стања за дати линеаризовани систем. Да би се превазишао овај недостатак за динамичке линеаризоване системе целог реда, недавно је предложен и уведен алгоритам ИУУ ПИДД2/ПИД тип целог реда у директној и затвореној грани који успешно решава овај недостатак применом другог извода грешке праћења, [58], [59].

### 3.2. Закон ИУУ типа $PD^2D^\alpha$ у отвореној спрези

Појава и примена теорије рачуна нецелог реда може се пратити уназад 300 година, а сада игра важну улогу у савременој науци и техници, посебно у области контролног инжењерства [9], [10]. Последњих деценија је све већа пажња посвећена развоју рачуна нецелог реда и њиховој примени у разним научним и инжењерским областима [9], [12]. [60]. Недавно је пажња усмерена на примени нецелог рачуна у напредном решавању све захтевнијих задатака управљања као и моделирању динамичких система нецелог реда [13], [61], [62], [63], [64], [65].

Такође, примена ИУУ-а на динамичке системе нецелог реда, је недавно разматрана и уведена [8], [18], [24], [25], [66], [67], [68] при чему ИУУ нецелог реда не задржава само предности класичног ИУУ-а, већ такође нуди потенцијал за остваривање бољих перформанси за разне сложене физичке процесе и системе. Већина постојећих алгоритама нецелог реда ИУУ (НРИУУ) су развијене за системе нецелог реда док само неколико њих се бави регуларним динамичким системима целобројног реда и сингуларним динамичким системима нецелог реда.

С друге стране, у многим практичним роботским апликацијама управљања, роботски системи се могу представити као нелинеарни или линеаризовани динамички системи целобројног реда. Дакле, постоји потешкоћа у примени алгоритма ИУУ нецелог реда на динамички систем целобројног реда.

Недавно су аутори [69] предложили алгоритам ИУУ типа  $PD^\alpha$  за већ динамички модел целобројног реда (дат као линеарни механички временски променљив систем).

Овде се предлаже један нови НРИУУ закон  $PD^2D^\alpha$  типа отворене петље за класу линеарних роботских система [70]. Термин  $D^2$  у предложеном НРИУУ служи за превазилажење ограничења матрице спрезања [CB]=0 за линеаризовани систем [58].

Даље се дају основне дефиниције и својства фракционих оператора Риман-Лиувиле (РЛ) и Капута [12]. Нека је  $f(\cdot) \in AC[a, b]$  непрекидна функција на коначном временском интервалу  $[a, b]$  има први извод готово свуда  $[a, b]$  будући интеграбилна, то јест она је у  $L^1[a, b]$ . Такође, простор  $AC^m([a, b])$  је простор функција са непрекидним  $m-1$  изводима на  $[a, b]$ , и  $m$ -тим изводом у  $L^1[a, b]$ . Дефиниција РЛ нецелог интеграла реда  $\mu$  је дата са:

$${}_a I_t^\mu f(t) = {}_a D_t^{-\mu} f(t) = \frac{1}{\Gamma(\mu)} \int_a^t (t-\tau)^{\mu-1} f(\tau) d\tau, \quad \mu \in R^+ \quad (3.6)$$

где је  $\Gamma(\cdot)$  позната Еулерова гама функција, која је дефинисана са  $\Gamma(z) = \int_0^\infty e^{-z} t^{z-1} dt$ . Фракциони РЛ оператор  ${}_a I_t^\mu(\cdot)$  поседује својство полугрупе, тј.  ${}_a I_t^\mu \cdot {}_a I_t^\xi = {}_a I_t^{\mu+\xi}$ . РЛ извод нецелог реда  $\mu$  је дефинисан са:

$${}^{RL} D_t^\mu f(t) = D^n ({}_a I_t^{-(n-\mu)} f(t)) = \frac{d^n}{dt^n} ({}_a I_t^{-(n-\mu)} f(t)) = \frac{1}{\Gamma(n-\mu)} \frac{d^n}{dt^n} \left[ \int_a^t (t-\tau)^{n-\mu-1} f(\tau) d\tau, \right] \quad (3.7)$$

где је  $n-1 \leq \mu < n \in Z^+$  и  $D^n(\cdot) = d^n(\cdot)/dt^n, n \in \mathbb{N}$  класични целобројни извод  $n$ - реда. Ако се уведе ознака  $n = [\mu] \in \mathbb{N}$  који представља целобројни део  $\mu$ , и  $\alpha = (\mu - [\mu]) \in (0, 1)$  следи следећи облик за РЛ фракциони оператор реда:

$${}^{RL} D_t^\mu f(t) = {}^{RL} D_t^{[\mu]+\alpha} f(t) = {}^{RL} D_t^{n+\alpha} f(t) = \frac{d^{n+1}}{dt^{n+1}} ({}_a I_t^{-(1-\alpha)} f(t)) \quad (3.8)$$

Такође, дефиниција Капутовог нецелог извода реда  $\mu$  је дефинисана као:

$${}^C D_t^\mu f(t) = {}_a D_t^{\mu-n} D^n f(t) = {}_a I_t^{n-\mu} (f^{(n)}(t)) = \frac{1}{\Gamma(n-\mu)} \int_a^t (t-\tau)^{n-\mu-1} f^{(n)}(\tau) d\tau, \quad (3.9)$$

где је  $n-1 < \mu < n \in Z^+$ . Такође, има се:

$${}^C D_t^\mu f(t) = {}_a D_t^{\mu-[\mu]-1} D^{[\mu]+1} f(t) = {}_a I_t^{n+1-\mu} (f^{(n+1)}(t)) = {}_a I_t^{1-\alpha} (f^{(n+1)}(t)) \quad (3.10)$$

У случају  $n=1$ , за  $0 \leq \alpha < 1$  добија се:

$${}^{RL} D_t^\alpha f(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \left[ \int_a^t (t-\tau)^{-\alpha} f(\tau) d\tau, \right], \quad {}^C D_t^\alpha f(t) = \frac{1}{\Gamma(1-\alpha)} \int_a^t (t-\tau)^{-\alpha} \frac{df(\tau)}{d\tau} d\tau, \quad (3.11)$$

У следећим одељцима,  $D^\alpha$  означаваће  ${}^C D_t^\alpha, {}^{RL} D_t^\alpha, {}^C D_t^\alpha$  ради сажетог записа оператора.

Ако  $f(0) = 0$  може се лако доказати да у случају  $a = t_0$  следи:

$${}_0 D_t^{-\alpha} \left( {}_0 D_t^{\alpha} f(t) \right) = f(t) \quad (3.12)$$

Лапласова трансформација Риман-Лиувиловог нецелог извода реда  $\mu > 0$  је

$$L\left\{ {}_0 \mathcal{D}_t^{\mu} f(t); s \right\} = s^{\mu} F(s) - \sum_{k=0}^{n-1} s^k \left[ {}_0 \mathcal{D}_t^{\mu-k-1} f(t) \right]_{t=0} \quad (3.13)$$

где је  $n-1 < \mu \leq n$ .

Лапласова трансформација Капутовог нецелог извода реда  $\mu > 0$  је

$$L\left\{ {}_0^C \mathcal{D}_t^{\mu} f(t); s \right\} = s^{\mu} F(s) - \sum_{k=0}^{n-1} s^{\mu-k-1} f^{(k)}(0) \quad (3.14)$$

где је  $n-1 < \mu \leq n$ .

### 3.2.1 Feedback линеаризација

Добро је познато да у литератури постоји много различитих шема управљања роботима. Овде разматрамо управљање роботским системом, где се динамика нашег работа може добити коришћењем поступка за израчунавање комплетне динамике роботског система датог у виду кинематичког ланца применом Родриговог приступа, [50]. Диференцијалне једначине роботског система могу се представити у коваријантном облику Лагранжевих једначина са одговарајућим генералисаним силама. Шта више, динамика роботске руке може се добити у компактном облику као:

$$a(q)\ddot{q} + \left( N(q, \dot{q}) - Q^g \right) = a(q)\ddot{q} + n(q, \dot{q}) = Q^a \quad (3.15)$$

где су  $q(t) \in \mathbb{R}^n, \dot{q}(t) \in \mathbb{R}^n, \ddot{q}(t) \in \mathbb{R}^n$  - вектор генералисаних координата роботског система који одређује конфигурацију истог, вектор генералисаних брзина као и вектор другог извода  $\ddot{q}(t) = d^2 q(t) / dt^2$ , респективно. Са  $Q^g \in \mathbb{R}^n, Q^a = U \in \mathbb{R}^n$  су означене - генералисане силе од сила теже и погонских сила респективно;  $a(q) = [a_{\alpha\beta}] \in R^{n \times n}$  означава основни метрички тензор, док матрица  $N(q, \dot{q})$  садржи елементе који укључују утицај инерцијалних сила (центрифугалних и Кориолисових).

Роботски систем који је предмет проучавања представља један нелинеаран систем што имплицира да је неопходно применити нелинеарне законе при дизајну управљачког система. Међутим, овде је од интереса да се спроведе *feedback* линеаризација објекта управљања где се путем повратне затворене петље добија



линеарна зависност улаза и излаза и затим и пројектовање одговарајућег управљачког система за тако добијени систем. То се постиже тако да све нелинеарне величине које фигуришу у једначинама кретања компензују се одговарајућим законом управљања. *Feedback* линеаризација је општи концепт управљања које је у литератури о роботизи познатија као *Computed-torque u Inverse dynamics* управљање, [71], [72] а суштински се односе на поменућу идеју, где се врши декомпозиција на спољашњу и унутрашњу петљу, тако да унутрашња служи за линеаризацију система, док се спољашњом постиже стабилизација. Може се овде истаћи да *feedback* линеаризација позната као и под називом *егзактна линеаризација* која важи на целом домену од интереса, за разлику од Јакобијан линеаризације где се линеарна апроксимација добија само у малој околини усвојене радне тачке. У неким случајевима *feedback* линеаризација омогућава да се полазни нелинеарни, спрегнути систем, не само линеаризује, већ и у потпуности распрегне, тако да се отвара могућност за независно подешавање сваког канала регулације посебно. Ако се у (3.15) примени следећи закон управљања (3.16) где  $u(t)$  представља ново управљање, следи:

$$Q^a = a(q)u(t) + n(q, \dot{q}) \quad (3.16)$$

У идеалном случају после примене *feedback* линеаризације добија се линеарни распрегнути систем у облику двоструког интегратора:

$$\ddot{q}(t) = u(t), \quad (3.17)$$

док у случају под утицајем неизвесности у моделу  $\eta = \eta(t)$ , има се:

$$\ddot{q}(t) = u(t) + \eta(t) \quad (3.18)$$

Модел *feedback* линеаризованог роботског система може се у простору стања написати као:

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t) + D\eta_i(t), \quad A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}_{2nx2n}, \quad B = D = \begin{bmatrix} 0 \\ I \end{bmatrix}_{2nxn} \quad (3.19)$$

$$y_i(t) = Cx_i(t) = \begin{bmatrix} I & 0 \end{bmatrix}_{n \times 2n} \cdot x_i(t) \quad (3.20)$$

Овде се приступа решавању проблема робустног праћења трајекторије која се понавља у простору стања линеаризованог система (3.19) током временског интервала  $J = [0, T]$ ,  $J \subset R$ , где је  $T$  време трајања итерације,  $i$ - представља број итерације, где су са  $x_i(t), u_i(t), y_i(t)$  означени вектор стања, управљања и излаза система, респективно. Другим речима, роботски систем мора поновљиво да прати жељену трајекторију  $q_d(t) \in R^n$ ,  $t \in J$  уз постојање моделске неизвесности  $\eta_i(t) \in R^n$ .

Даље се разматра и примењује један нови напредни алгоритам робустног ИУУ-а нецелог реда, у отвореној грани, типа  $PD^2D^\alpha$ . Овде се може истаћи чињеница да у литератури постоји мали број алгоритама ИУУ нецелог реда који су примењени за динамичке системе који нису нецелог реда већ целобројног реда. Сама примена ИУУ захтева увођење и одговарајућих претпоставки:

A1: Жељене трајекторије  $y_d(t), x_d(t)$  су непрекидно диференцијабилне на  $J = [0, T]$

A2: Систем (3.19, 3.20) је каузалан, тј. за дат жељени излаз  $y_d(t)$ , постоји јединствени улазни вектор управљања  $u_d(t)$ , као и  $x_d(t)$  који омогућавају да излаз система управо буде  $y_d(t)$ , тј.:

$$\dot{x}_d(t) = Ax_d(t) + Bu_d(t), \quad (3.21)$$

$$y_d(t) = Cx_d(t), \quad (3.22)$$

A3: Понављање почетних услова из итерације у итерацију је испуњено, тј. почетно стање у  $i$ -тој итерацији је увек исто на почетку сваке итерације:

$$x_i(0) = x_d(0), \quad i = 0, 1, 2, \dots, \quad (3.23)$$

A4. Моделска неизвесност  $\eta_i(t)$  је униформно ограничена. Даље, користиће се позитивна константа  $d_\eta$ , да се означи горња граница за  $\eta_i(t)$ , то јест  $\forall t \in [0, T], \forall i \rightarrow \|\eta_i(t)\| \leq d_\eta$ .

У циљу доказа конвергентности предложеног алгоритма управљања учењем ИУУ уводе се следеће норме тј. дефинишу на следећи начин [54] за  $n$ -димензиони Еуклидски простор  $R^n$ :

-max норма  $\|x\| = \max_{0 \leq t \leq T} |x(t)| \quad x = [x_1, x_2, \dots, x_n]^T$ ,

-суп-норма  $\|x\|_\infty = \sup_{1 \leq i \leq n} |x_i| \quad x = [x_1, x_2, \dots, x_n]^T$ ,

-матрична норма  $\|A\|_\infty = \max_{1 \leq i \leq m} \left( \sum_{j=1}^n |g_{i,j}| \right), \quad A = [a_{i,j}]_{m \times n}$

- $\lambda$ -норма која се дефинише за реалну функцију  $h(t)$

$$h(t), \quad (t \in [0, T]), \quad h: [0, T] \rightarrow \mathfrak{R}^n$$

$$\|h(t)\|_\lambda = \sup_{t \in [0, T]} e^{-\lambda t} \|h(t)\|_\infty, \quad \lambda > 0$$

(3.24)

Индукована норма матрице  $A$  се дефинише као:

$$\|A\| = \sup \left\{ \frac{\|Ax\|}{\|x\|} : x \in X \text{ with } \|x\| \neq 0 \right\}, \quad (3.25)$$

где  $\|(\cdot)\|$  означава произвољну векторску норму. У случају  $\|(\cdot)\|_\infty$  имамо  $\|Ax\|_\infty \leq \|A\|_\infty \|x\|_\infty$ , где  $\|A\|_\infty$  означава максималну вредност матрице  $A$ . За претходне норме важи следеће  $\|h(t)\|_\lambda \leq \|h(t)\|_\infty \leq e^{\lambda T} \|h(t)\|_\lambda$ . чиме је доказана еквивалентност  $\lambda$ -норме са  $\infty$ -нормом. Због једноставности када се примењује  $\|(\cdot)\|_\infty$  индекс  $\infty$  ће бити изостављен. Такође, од користи је и следећа особина  $\lambda$ -норме:

$$\sup_{t \in [0, T]} e^{-\lambda t} \int_0^t \|f(\cdot)\| e^{a(t-\tau)} d\tau = \sup_{t \in [0, T]} \int_0^t e^{-\lambda t} \|f(\cdot)\| e^{a(t-\tau)(a-\lambda)} d\tau \leq \frac{1 - e^{(a-\lambda)T}}{\lambda - a} \|f(\cdot)\|_\lambda \quad (3.26)$$

Овде се наводи Лема 1 [54].

**Лема 1.** Претпоставимо да реални позитивни бесконачни ред  $\{a_n\}_1^\infty$  задовољава

$$q_k \leq \rho q_{k-1} + \tilde{\varepsilon} \quad (3.27)$$

где су  $\rho \geq 0$ ,  $\tilde{\varepsilon} > 0$  и  $\rho < 1$ . Онда важи следећи израз:

$$\lim_{k \rightarrow \infty} q_k \leq \tilde{\varepsilon} / (1 - \rho) \quad (3.28)$$

Такође, уочава се да у случају када је  $\tilde{\varepsilon} = 0$ ,  $\lim_{k \rightarrow \infty} q_k \rightarrow 0$ .

**Лема 2.** (Bellman-Gronwall-ова неједнакост) [73]:

Нека су  $f_1(t)$ ,  $g_1(t)$ , и  $h_1(t)$  ненегативне непрекидне функције на временском интервалу  $[0, T]$ . Ако постоји ненегативна константа  $a$  тако да следећа неједнакост важи:

$$f(t) \leq h(t) + \int_0^t a f(\tau) d\tau + \int_0^t g(\tau) d\tau, \quad (3.29)$$

онда, следи:

$$f(t) \leq h(t) + \int_0^t \exp(a \cdot (t - \tau)) [ah(\tau) + g(\tau)] d\tau \quad (3.30)$$

### 3.2.2 Резултати конвергенције -закона учења типа PD<sup>2</sup>D<sup>α</sup> у отвореној спрези

Као што смо раније навели, многи ИУУ закони имају ограничење које укључује претпоставку да матрица спрезања [CB] има пун ранг где В означава контролну матрицу, а С представља излазну матрицу простора стања за линеаризовани систем (3.19), (3.20). Да бисмо решили овај недостатак, као и побољшали перформансе управљања роботом узимајући у обзир својства фракционих оператора, уводи се следећи закон о учењу PD<sup>2</sup>D<sup>α</sup> у отвореној спрези, (види слику 3.2):

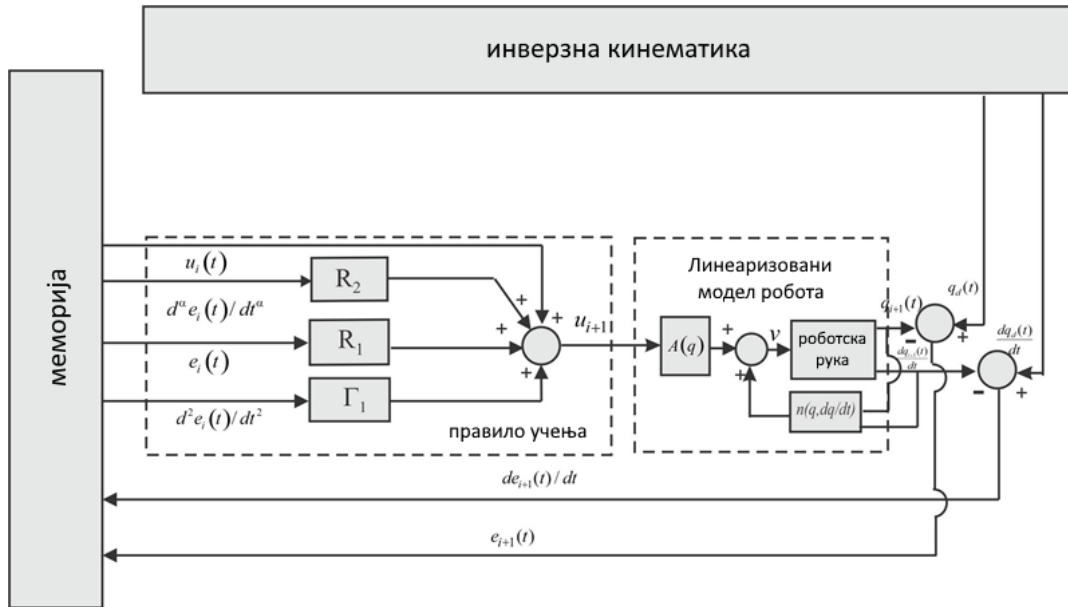
$$u_{i+1}(t) = u_i(t) + \Gamma_1 \ddot{e}_i(t) + R_1 \dot{e}_i(t) + R_2 e_i^{(\alpha)}(t) \quad (3.31)$$

где је  $e_i(t) = y_d(t) - y_i(t)$  грешка праћења путање у  $i$ -тој итерацији,  $e_{i+1}(t) = y_d(t) - y_{i+1}(t)$  је грешка праћења путање у  $i+1$ -тој итерацији,  $\Gamma_1, R_1, R_2 \in R^{n \times n}$ , означавају дијагоналне позитивно-дефинитивне матрице учења у отвореној спрези, тј.  $R_{(.)} = \text{diag}(r_{(.)1}, r_{(.)2}, \dots, r_{(.)n})$ ,  $\Gamma_{(.)} = \text{diag}(g_{(.)1}, g_{(.)2}, \dots, g_{(.)n})$ .

**Лема 3.** За систем (3.19), (3.20) и референтни систем (3.21), (3.22) тада постоји довољно велико  $\lambda$  за које је испуњена следећа релација:

$$\left\| \delta x_i^{(\alpha)}(t) \right\|_{\lambda} \leq \rho'_0 \left\| \delta u_i(t) \right\|_{\lambda} + \varepsilon_{\eta} \quad (3.32)$$

где су  $O(\lambda^{-1}) = \frac{(1 - e^{-\lambda T})}{\lambda} \leq \frac{1}{\lambda}$  и  $\rho'_0 = \rho_0 + h_0 O(\lambda^{-1})$ ,  $\varepsilon_{\eta} = (h_1 + h_2 T) d_{\eta}$ ,  $\rho_0, h_0, h_1, h_2, d_{\eta}$  неке позитивне константе.



Слика 3.2. Блок дијаграм закона учења  $PD^2D^\alpha$  у отвореној спрези за управљање робота

Доказ: Може се добити решење једначине (3.19), тј. одредити одзив у простору стања  $x_i(t)$  који је сада облика:

$$\begin{aligned}
 x_i(t) &= \Phi(t)x_i(0) + \int_0^t \Phi(t-\tau)Bu_i(\tau)d\tau + \int_0^t \Phi(t-\tau)D\eta_i(\tau)d\tau \\
 &= g_i(t) + \int_0^t \Phi(t-\tau)Bu_i(\tau)d\tau + \int_0^t \Phi(t-\tau)D\eta_i(\tau)d\tau
 \end{aligned} \tag{3.33}$$

као и излаз система  $y_i(t)$

$$y_i(t) = Cx_i(t) \tag{3.34}$$

где су  $g_i(t) \in R^r$  и  $\Phi(t-\tau) \in R^{r \times r}$  непрекидне функције по  $t$  и  $\tau$ ,  $x_i(0)$  је почетна вредност  $x_i(t)$  у  $i$ -тој итерацији.

На сличан начин имамо решење за једначину у простору стања (3.21),

$$x_d(t) = g_d(t) + \int_0^t \Phi(t-\tau)Bu_d(\tau)d\tau \tag{3.35}$$

Применом нецелог извода  $\alpha$ -ог реда на (3.35), добијамо:

$$x_d^{(\alpha)}(t) = g_d^{(\alpha)}(t) + \lim_{\tau \rightarrow t-0} \left[ {}_\tau D_t^{\alpha-1} \Phi(t-\tau)Bu_d(\tau) \right] - \int_0^t {}_\tau D_t^\alpha (\Phi(t-\tau)Bu_d(\tau))d\tau \tag{3.36}$$

Нека су

$$\delta x_i = x_d(t) - x_i(t), \quad \delta \dot{x}_i = \dot{x}_d(t) - \dot{x}_i(t) \quad (3.37)$$

$$\delta u_i = u_d(t) - u_i(t), \quad \delta \dot{u}_i = \dot{u}_d(t) - \dot{u}_i(t)$$

$$D^\alpha (\delta x_i(t)) = \delta x_i^{(\alpha)} = x_d^{(\alpha)}(t) - x_i^{(\alpha)}(t)$$

Такође, из (3.37) произлази, [69]:

$$\begin{aligned} \delta x_i^{(\alpha)}(t) &= x_d^{(\alpha)}(t) - x_i^{(\alpha)}(t) = g_d^{(\alpha)}(t) - g_i^{(\alpha)}(t) + \lim_{\tau \rightarrow t-0} \left[ {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) B u_d(\tau)) \right] - \\ &\lim_{\tau \rightarrow t-0} \left[ {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) B u_i(\tau)) \right] + \int_0^t {}_\tau D_t^\alpha (\Phi(t-\tau) B u_d(\tau)) d\tau - \int_0^t {}_\tau D_t^\alpha (\Phi(t-\tau) B u_i(\tau)) d\tau \\ &- \lim_{\tau \rightarrow t-0} \left[ {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) D \eta_i(\tau)) \right] - \int_0^t {}_\tau D_t^\alpha (\Phi(t-\tau) D \eta_i(\tau)) d\tau \end{aligned} \quad (3.38)$$

Узимајући у обзир претпоставку АЗ, следи:

$$g_d^{(\alpha)}(t) - g_i^{(\alpha)}(t) = 0. \quad (3.39)$$

Према томе, израз (3.38) има следећи облик:

$$\begin{aligned} \delta x_i^{(\alpha)}(t) &= {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) B) \Big|_{\tau=t} \delta u_i(t) + \int_0^t {}_\tau D_t^\alpha (\Phi(t-\tau) B \delta u_i(\tau)) d\tau \\ &- \left[ {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) D \eta_i(\tau)) \Big|_{\tau=t} \right] - \int_0^t {}_\tau D_t^\alpha (\Phi(t-\tau) D \eta_i(\tau)) d\tau \end{aligned} \quad (3.40)$$

Примењујући  $\lambda$ -норму на израз (3.40), добија се:

$$\begin{aligned} \left\| \delta x_i^{(\alpha)}(t) \right\|_\lambda &\leq \sup_{0 \leq t \leq T} \left\| {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) B) \Big|_{\tau=t} \right\|_\infty \left\| \delta u_i(t) \right\|_\lambda + \\ &+ \sup_{0 \leq t, \tau \leq T} \left\| {}_\tau D_t^\alpha (\Phi(t-\tau) B) \right\|_\infty \int_0^t e^{-\lambda t} \cdot \left\| \delta u_i(\tau) \right\|_\infty d\tau \\ &+ \sup_{0 \leq t \leq T} \left\| {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) D) \right\|_\infty \left\| \eta_i(t) \right\|_\infty + \sup_{0 \leq t, \tau \leq T} \left\| {}_\tau D_t^\alpha (\Phi(t-\tau) D) \right\|_\infty \int_0^t e^{-\lambda t} \left\| \eta_i(\tau) \right\|_\infty d\tau \\ \left\| \delta x_i^{(\alpha)}(t) \right\|_\lambda &\leq \sup_{0 \leq t \leq T} \left\| {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) B) \Big|_{\tau=t} \right\|_\infty \left\| \delta u_i(t) \right\|_\lambda + \\ &+ \sup_{0 \leq t, \tau \leq T} \left\| {}_\tau D_t^\alpha (\Phi(t-\tau) B) \right\|_\infty \int_0^t e^{-\lambda(t-\tau)} \cdot \sup_{0 \leq \tau \leq T} e^{-\lambda \tau} \left\| \delta u_i(\tau) \right\|_\infty d\tau \\ &+ \sup_{0 \leq t \leq T} \left\| {}_\tau D_t^{\alpha-1} (\Phi(t-\tau) D) \right\|_\infty \cdot d_\eta + \sup_{0 \leq t, \tau \leq T} \left\| {}_\tau D_t^\alpha (\Phi(t-\tau) D) \right\|_\infty \int_0^t e^{-\lambda t} d_\eta d\tau \end{aligned} \quad (3.41)$$

или

$$\begin{aligned} \|\delta x_i^\alpha(t)\|_\lambda &\leq \rho_0 \|\delta u_i(t)\|_\lambda + h_0 \cdot \|\delta u_i(\tau)\|_\lambda \sup_{0 \leq t \leq T} \int_0^t e^{-\lambda(t-\tau)} \cdot d\tau + h_1 d_\eta + \\ &+ h_2 d_\eta \cdot \sup_{0 \leq t \leq T} \int_0^t e^{-\lambda t} d\tau \\ \|\delta x_i^\alpha(t)\|_\lambda &\leq \rho_0 \|\delta u_i(t)\|_\lambda + h_0 \|\delta u_i(\tau)\|_\lambda O(\lambda^{-1}) + h_1 d_\eta + h_2 d_\eta T \end{aligned} \quad (3.42)$$

где су

$$\begin{aligned} \rho_0 &= \sup_{0 \leq t \leq T} \left\| {}_\tau D_t^{\alpha-1} (\Phi(t-\tau)B) \Big|_{\tau=t} \right\|_\infty, h_0 = \sup_{0 \leq t, \tau \leq T} \left\| {}_\tau D_t^\alpha (\Phi(t-\tau)B) \right\|_\infty, \\ h_1 &= \sup_{0 \leq t \leq T} \left\| {}_\tau D_t^{\alpha-1} (\Phi(t-\tau)D) \right\|_\infty, h_2 = \sup_{0 \leq t, \tau \leq T} \left\| {}_\tau D_t^\alpha (\Phi(t-\tau)D) \right\|_\infty \end{aligned} \quad (3.43)$$

$$O(\lambda^{-1}) = \frac{(1-e^{-\lambda T})}{\lambda} \leq \frac{1}{\lambda}, \quad \|\eta_i(t)\|_\infty \leq d_\eta$$

На тај начин, добија се:

$$\|\delta x_i^\alpha(t)\|_\lambda \leq \rho'_0 \|\delta u_i(t)\|_\lambda + \varepsilon_\eta \quad (3.44)$$

где су:

$$\rho'_0 = \rho_0 + h_0 O(\lambda^{-1}), \quad \varepsilon_\eta = (h_1 + h_2 T) d_\eta \quad (3.45)$$

Овим је доказ леме завршен.

Довољан услов за конвергенцију предложеног ИУУ отворене петље дата је Теоремом 1 и са пратећим доказом.

**Теорема 1:** Узимајући у обзир систем (3.19), (3.20) под претпоставкама (A1) - (A4). Ако важи:

$$\rho + \tilde{\rho}_0 < 1 \quad (3.46)$$

где су,  $\rho = \|(I - \Gamma_1 CAB)\|_\infty$ ,  $\tilde{\rho}_0 = \beta_3 \cdot \rho_0 = \|R_2 C\| \cdot \sup_{0 \leq t \leq T} \left\| {}_\tau D_t^{\alpha-1} (\Phi(t-\tau)B) \Big|_{\tau=t} \right\|_\infty$ , онда, закон учења  $PD^2D^\alpha$  у отвореној спрези (3.31) гарантује да када  $i \rightarrow \infty$  границе грешака праћења  $\|x_d(t) - x_i(t)\|$ ,  $\|y_d(t) - y_i(t)\|$ , и  $\|u_d(t) - u_i(t)\|$  асимптотски конвергирају у одређене границе.

Границе  $\tilde{\varepsilon}_x, \tilde{\varepsilon}_y, \tilde{\varepsilon}_u$  су дате следећим изразима:  $\tilde{\varepsilon}_u = \frac{1}{1-\tilde{\rho}} \tilde{\varepsilon}$ ,  $\tilde{\varepsilon}_x = d_1 d_\eta T + b O(\lambda_1^{-1}) \frac{1}{1-\tilde{\rho}} \tilde{\varepsilon}$ ,

$\tilde{\varepsilon}_y = c \cdot \tilde{\varepsilon}_x$  где су  $\tilde{\rho} = \rho + \beta_3 (\rho_0 + h_0 O(\lambda^{-1})) + \beta_1 b O(\lambda_1^{-1})$ ,  $\tilde{\varepsilon} = \beta_1 d_1 d_\eta T + \beta_2 d_\eta + \beta_3 \varepsilon_\eta$ .

Доказ:

Грешка праћења и њени изводи целобројни као и фракциони могу се одредити на следећи начин:

$$e_i = C\delta x_i, \quad e_i^{(\alpha)} = C\delta x_i^{(\alpha)}, \quad \dot{e}_i = C\delta \dot{x}_i = C(A\delta x_i + B\delta u_i - D\eta_i) \quad (3.47)$$

$$\ddot{e}_i = C\delta \ddot{x}_i = CA^2\delta x_i + CAB\delta u_i - CD\dot{\eta}_i - CAD\eta_i$$

Ако се узме у обзир сугерисани закон управљања путем учења ИУУ (3.31) следи:

$$\delta u_{i+1}(t) = u_d(t) - u_{i+1}(t) = \delta u_i(t) - \Gamma_1 \ddot{e}_i(t) - R_1 e_i(t) - R_2 e_i^{(\alpha)}(t) \quad (3.48)$$

Заменом израза (3.47) у (3.48) следи да је

$$\begin{aligned} \delta u_{i+1}(t) = & [I - \Gamma_1 CAB] \delta u_i(t) - (\Gamma_1 CA^2 + R_1 C) \delta x_i(t) - \\ & - R_2 C \delta x_i^{(\alpha)}(t) + \Gamma_1 CAD \eta_i(t) + \Gamma_1 CD \dot{\eta}_i(t) \end{aligned} \quad (3.49)$$

Може се приметити да у овом, предложеном случају матрице  $CAB, CAD$  имају пун ранг, а матрице  $CB, CD$  немају пун ранг, ( $[C][B]=0$ ), ( $[C][D]=0$ ). Из претходне једначине (3.49), добијамо:

$$\delta u_{i+1}(t) = [I - \Gamma_1 CAB] \delta u_i(t) - (\Gamma_1 CA^2 + R_1 C) \delta x_i(t) - R_2 C \delta x_i^{(\alpha)}(t) + \Gamma_1 CAD \eta_i(t) \quad (3.50)$$

Ако се примени норма  $\|(\cdot)\|$  на (3.50) и имајући у виду услов Теореме 1, као и претпоставке А1, повлачи да је сада:

$$\|\delta u_{i+1}(t)\| \leq \rho \|\delta u_i(t)\| + \left\| [\Gamma_1 CA^2 + R_1 C] \right\| \|\delta x_i(t)\| + \|\Gamma_1 CAD\| \|\eta_i(t)\| + \|R_2 C\| \|\delta x_i^{(\alpha)}(t)\| \quad (3.51)$$

После сређивања (3.51) добија се:

$$\|\delta u_{i+1}(t)\| \leq \rho \|\delta u_i(t)\| + \beta_1 \|\delta x_i(t)\| + \beta_2 d_\eta + \beta_3 \|\delta x_i^{(\alpha)}(t)\| \quad (3.52)$$

где су

$$\beta_1 = \left\| [\Gamma_1 CA^2 + R_1 C] \right\|, \quad \beta_2 = \|\Gamma_1 CAD\|, \quad \beta_3 = \|R_2 C\| \quad (3.53)$$

Такође, на основу израза (3.19) и (3.47) има се:

$$\delta x_i = \int_0^t (A\delta x_i(\tau) + B\delta u_i(\tau) - D\eta_i(\tau)) d\tau \quad (3.54)$$

Слично, применом норме на себе стране (3.54), следи:



$$\|\delta x_i\| = \int_0^t (a\|\delta x(\tau)\| + b\|\delta u_i(\tau)\| + d_1\|\eta_i(\tau)\|)d\tau \quad (3.55)$$

где су  $a = \|A\|, b = \|B\|, d_1 = \|D\|$ . Узимајући у обзир претпоставку A4, има се:

$$d_1 \int_0^t (\|\eta_i(\tau)\|)d\tau \leq d_1 \cdot d_\eta T \quad (3.56)$$

што имплицира да је сада у важности

$$\|\delta x_i\| \leq d_1 d_\eta T + \int_0^t (a\|\delta x(\tau)\| + b\|\delta u_i(\tau)\|)d\tau \quad (3.57)$$

Применом сада Белман-Гронвалове леме (*Bellman-Grownwall lemma*) [69], добија се

$$\|\delta x_i\| \leq d_1 d_\eta T e^{at} + \int_0^t e^{a(t-\tau)} b \|\delta u_i(\tau)\| d\tau \quad (3.58)$$

где се после множења претходног израза са  $e^{-\lambda t}, \lambda > a$  и применом  $\lambda$ -норме, добија

$$\|\delta x_i\|_\lambda \leq d_1 d_\eta T + \frac{1 - e^{-(\lambda-a)T}}{\lambda - a} b \|\delta u_i\|_\lambda \leq d_1 d_\eta T + bO(\lambda_1^{-1}) \|\delta u_i\|_\lambda, \quad O(\lambda_1^{-1}) = 1/(\lambda - a) \quad (3.59)$$

Узимајући у обзир (3.52), и применом  $\lambda$ -норме, следи

$$\|\delta u_{i+1}(t)\|_\lambda \leq \rho \|\delta u_i(t)\|_\lambda + \beta_1 \|\delta x_i(t)\|_\lambda + \beta_2 d_\eta + \beta_3 \|\delta x_i^{(\alpha)}(t)\|_\lambda \quad (3.60)$$

Сада, на основу спајања израза (3.32), (3.59) са (3.60), добија се

$$\begin{aligned} \|\delta u_{i+1}(t)\|_\lambda &\leq \rho \|\delta u_i(t)\|_\lambda + \beta_1 (d_1 d_\eta T + bO(\lambda_1^{-1}) \|\delta u_i\|_\lambda) + \beta_2 d_\eta \\ &+ \beta_3 (\rho_0' \|\delta u_i(t)\|_\lambda + \varepsilon_\eta) \end{aligned} \quad (3.61)$$

односно у кондензованој форми:

$$\|\delta u_{i+1}(t)\|_\lambda \leq \tilde{\rho} \|\delta u_i(t)\|_\lambda + \tilde{\varepsilon} \quad (3.62)$$

где су:

$$\begin{aligned} \tilde{\rho} &= \rho + \beta_3 (\rho_0 + h_0 O(\lambda^{-1})) + \beta_1 b O(\lambda_1^{-1}), \\ \tilde{\varepsilon} &= \beta_1 d_1 d_\eta T + \beta_2 d_\eta + \beta_3 \varepsilon_\eta \end{aligned} \quad (3.63)$$

Будући да је  $\rho + \beta_3 \rho_0 < 1$  по претпоставци, могуће изабрати довољно велико  $\lambda$  тако да  $\tilde{\rho} < 1$ . Према томе, на основу Леме 1, произилази:

$$\lim_{i \rightarrow \infty} \|\delta u_i\|_\lambda \leq \frac{1}{1 - \tilde{\rho}} \tilde{\varepsilon} = \tilde{\varepsilon}_u. \quad (3.64)$$

Заменом (3.55) у (3.59), као и у (3.38) следи

$$\begin{aligned} \lim_{i \rightarrow \infty} \|\delta x_i\|_\lambda &\leq d_1 d_\eta T + bO\left(\lambda_1^{-1}\right) \left(\frac{1}{1 - \tilde{\rho}} \tilde{\varepsilon}\right) = \tilde{\varepsilon}_x, \\ \lim_{i \rightarrow \infty} \|e_i\|_\lambda &\leq c \left( d_1 d_\eta T + bO\left(\lambda_1^{-1}\right) \left(\frac{1}{1 - \tilde{\rho}} \tilde{\varepsilon}\right) \right) = \tilde{\varepsilon}_y \end{aligned} \quad (3.65)$$

Овим је доказ Теореме 1 завршен.

*Напомена 1:* У случају неизвесности и несигурности система,  $\xi_i(t) = 0, \eta_i(t) = 0$ , тј.  $d_\xi, d_\eta, d_{\eta\alpha}$  ове границе такође теже нули. Наиме, може се добити када  $i \rightarrow \infty$  границе грешака праћења  $\|x_d(t) - x_i(t)\|, \|y_d(t) - y_i(t)\|, \|u_d(t) - u_i(t)\|$ , асимптотски конвергирају нули. Доказ је сличан доказу Теореме 1, узимајући у обзир, тј.  $\xi_i(t) = 0, \eta_i(t) = 0$ , то јест  $\varepsilon = \delta = 0$ .

$$\lim_{i \rightarrow \infty} \|\delta u_i\|_\lambda = 0 \quad (3.66)$$

На основу теореме о постојању и јединствености диференцијалне једначине целобројног реда, добија се да

$$\lim_{i \rightarrow \infty} y_i(t) = y_d(t) \quad (3.67)$$

### 3.3. Итеративно управљање учењем у отворено-затвореној спрези типа нецелог реда једним сингуларним динамичким системом нецелог реда

У овом поглављу разматра се проблематика управљања једном широм класом динамичких система тј. сингуларним системима нецелог реда и то применом напредних ИУУ који су такође нецелог реда. Уочено је да се сингуларни системи природно појављују у многим инжењерским дисциплинама и проблемима, где се применом рачуна нецелог реда сада могуће добити квалитетније моделе сингуларних система. Такође, овде је од интереса решавање задатка управљање истих. Са друге стране, као врста интелигентног управљања, ИУУ привукао је пажњу због једноставности и ефикасности примене алгоритама ИУУ, односно његових особина да се успешно носе са проблемима неизвесности, временског кашњења,

нелинеарности, као и сложенијим сингуларним системима. Сингуларни системи се добијају као резултат напредног математичког моделирања многих физичких система и процеса, и који се могу описати комбинацијом алгебарских и диференцијалних једначина. Тако, сингуларни системи се појављују у великом броју проблема практичног инжењерства, приликом формирања одговарајућег модела електричних мрежа [74], или у одређивању одговарајућих механичких модела [75] итд. На пример, електричне мреже су најчешће састављене од подсистема мрежних елемената (као што су кондензатори, отпорници, индуктивни елементи) и који су описани различитим врстама диференцијалних једначина а које се сада спрежу са Кирхофовим законима (описани алгебарским једначинама), [76]. У механичким системима диференцијалне једначине најчешће описују динамику подсистема док су алгебарским једначинама одређена ограничења које карактеришу везе тј. ограничења која су наметнута датим везама на пример као што су зглобови [75] или у моделу ограниченог кретања роботског система тј. роботског система у контакту са околином, (види поглавље 2), [77], [78].

Са друге стране, уочено је да се многи теоријски резултати који су добијени за регуларне (несингуларне) системе могу проширити на одговарајуће сингуларне системе. У поређењу са регуларним системима, није тешко закључити да сингуларни системи могу описати и приказати ширу класу динамичких система односно описати више структурних особина физичких система него у случају регуларних система, [79]. У зависности од области примене, ови модели се такође називају дескриптор системи, диференцијално-алгебарским системи, системи са полустањем (*semi-state*), или уопштени системи. Посебно, дескриптивни облик укључује информације о статичким и динамичким ограничењима. Овако формиран модели сингуларних система имају извесне предности, од којих су најзначајније: задржавање битних физичких својстава у самом моделу, блиска повезаност са стварним физичким променљивама стања система, нема потребе за елиминацијом спрегнуто променљивих величина што је код нелинеарних система и практично немогуће, исписивање билансних и других једначина далеко је непосредније, матрице које фигуришу у овим моделима, по правилу су шупље, што знатно поједностављује нумеричке процедуре [80]. Са друге стране, због многих специфичности сингуларних система постоји низ додатих проблема који неминовно прате њихову анализу и следствено, могућу синтезу. Наиме, код оваквих система потребно је размотрити не само стабилност истих, већ и питање постојања и јединствености решења, егзистенцију почетних услова који генеришу тзв. глатка решења и одсуство импулсног понашања у исто време [81] за поједине сингуларне системе, приступ регуларизације сингуларних система на обичне, нормалне системе, итд. Исто тако, уочена сложена природа сингуларних система може да створи многе потешкоће у аналитичком и нумеричком решавању истих. Канонична форма сингуларних система се може приказати у облику:

$$E\dot{x}(t) = f(t, x(t), u(t)), \quad (3.68)$$

$$y(t) = g(t, x(t), u(t)) \quad (3.69)$$

или у посебном, линеарном најчешће сусретаном случају у литератури као

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad (3.70)$$

$$y(t) = Cx(t) + Du(t) \quad (3.71)$$

Такође, роботски системи у решавању контактних проблема могу се моделирати као сингуларни системи, [82], [78].

У последњих неколико деценија посебна пажња је посвећена примени рачуна нецелог реда, тј. фракционим диференцијалним једначинама и њиховим применама у различитим научним и инжењерским пољима [13], [14]. Недавно по први пут примењени су алгоритми ИУУ управљања за одговарајућу класу класичних регуларних сингуларних система, [15], [16].

Са друге стране, недавно су добијени су резултати у примени алгоритама ИУУ управљања нецелог реда и то за регуларне системе нецелог реда дати у временском домену, [14], [17], [18], [22], [23], [83].

Даље, овде ће бити презентовани нови резултати који су добијени у примени алгоритама ИУУ управљања нецелог реда а за одговарајуће класе сингуларних система нецелог реда, који се могу наћи у литератури и који све више заузимају место и у роботици и у решавању контактних проблема [84], [85], [86].

### 3.3.1 Итеративно управљање путем учења нецелог реда за линеарне сингуларне системе нецелог реда

Овде је од интереса разматрање једног линеарног сингуларног система нецелог реда (СНРС) који је дат следећом једначином стања и једначином излаза респективно:

$$ED^\alpha x(t) = Ax(t), \quad n-1 < \alpha < n \quad (3.72)$$

$$y(t) = Cx(t), \quad (3.73)$$

При томе, дозвољени почетни услови за (3.74) дати су са:

$$x^{(k)}(t) = x_{0,k}(t), \quad k = 1, 2, \dots, n-1 \quad (3.74)$$

Овде фракциони оператор  ${}_0^C D^\alpha = D_t^\alpha$  означава  $\alpha$ -ти Капуто извод; односно  $E, A$  и  $C$  су матрице одговарајућих димензија.

**Дефиниција 1.** а) Систем СНРС-а дат са (3.72) је регуларан ако је  $\det(s^\alpha E - A) \neq 0$ .

б) за систем СНРС (3.72) се каже да је импулсно слободан ако је регуларан и ако је испуњено да је  $\deg(\det(s^\alpha E - A)) = \text{rank} E$ .

**Лема 1.** Тројка  $(E, A, \alpha)$  се назива регуларном ако и само ако је  $\det(s^\alpha E - A) \neq 0$  за неке  $s \in \mathbb{C}$ , [87]. Ако је тројка  $(E, A, \alpha)$  регуларна, систем СНРС (3.72) назива се регуларним, па је према томе систем СНРС решив.

Даље се посебно разматра једна класа неаутономног сингуларног линеарног система нецелог реда  $\alpha \in (0, 1)$  који се дат следећом једначином стања као и једначином излаза, следствено [88]:

$$ED^\alpha x(t) = Ax(t) + Bu(t), \quad 0 < \alpha < 1 \quad (3.75)$$

$$y(t) = Cx(t), \quad (3.76)$$

Овде  $A, B$  и  $C$  представљају матрице одговарајућих димензија,  $t$  означава време на одговарајућем временском интервалу рада система  $J = [t_o, t_o + T]$ ,  $J \subset \mathbb{R}$ . Даље се уводи претпоставка да је испуњено  $\det E = 0$ , као и да је СНРС систем регуларан. Такође, уводи се следећа претпоставка која се односи на питање постојања, јединствености и континуитета решења у односу на почетне услове, [87], [89]. Са друге стране, ако постоји параметарска неизвесност код сингуларних система онда се она може приказати на следећи начин:

$$ED^\alpha x(t) = (A + \Delta A)x(t) + Bu(t), \quad 0 < \alpha < 1 \quad (3.77)$$

$$y(t) = Cx(t), \quad (3.78)$$

Уводе се и следеће претпоставке које се односе на систем (3.75), и (3.76):

A1. Жељене трајекторије  $y_d(t), x_d(t)$  су непрекидно диференцијабилне на  $[0, T]$ .

A2. За дату жељену трајекторију излаза  $y_d(t)$ , постоји управљање  $u_d(t)$  тако да

$$ED^\alpha x_d(t) = Ax_d(t) + Bu_d(t), \quad 0 < \alpha < 1 \quad (3.79)$$

$$y_d(t) = Cx_d(t), \quad (3.80)$$

A3. Систем СНРС је обсервабилан и управљив.

A4. За све итерације почетни услови ресетовања важе  $x_k(0) = x_d(0)$ ,  $k = 0, 1, 2, \dots$ ,

### 3.3.2 Анализа конвергенције датог алгоритма ИУУ нецелог реда

У раду [25] је показано да је брзина праћења била најбржа када систем и алгоритам ИУУ нецелог реда имају исти нецели ред, тако да се овде уводи и предлаже следећи тип ИУУ  $PD^\alpha$ -итеративног управљања учењем нецелог реда у затвореној(повратној) спрези (3.81). У алгоритмима ИУУ-у повратна грана, обезбеђује стабилност система и задржава његове грешке вектора стања унутар униформних граница. Према томе, управљање ИУУ у повратној грани је уведено и дато са:

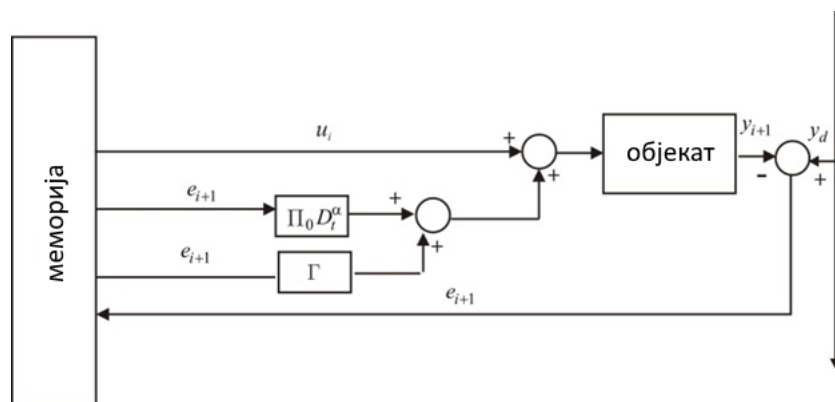
$$u_{fb_{i+1}}(t) = \Pi_0 D_t^\alpha e_{i+1}(t^-) + \Gamma e_{i+1}(t^-) \quad (3.81)$$

где  $e_{i+1}(t^-) = e_{i+1}(t - \varepsilon)$ ,  $\varepsilon \rightarrow 0^+$  означава вектор управо оствареног сигнала грешке праћења у датом временском тренутку. У већини случајева временско кашњење повратне спреге је занемарљиво тако да се може узети:  $e_{i+1}(t^-) = e_{i+1}(t)$ . Према томе, алгоритам  $PD^\alpha$  тип ИУУ нецелог реда у повратној грани је дат следећим изразом:

$$u_{i+1}(t) = u_i(t) + \Gamma e_{i+1}(t) + \Pi_0 D_t^\alpha e_{i+1}(t), \quad (3.82)$$

где  $u(t)$  је сигнал управљања у датом тренутку времена  $t$ , сл.3.3. а са  $\Gamma, \Pi$  су означене матрице појачања одговарајућих димензија. Теоремом 2 су изведени довољни услови за конвергенцију предложеног ИУУ у повратној спрези.

Као што је познато типичан алгоритам управљања ИУУ у временском домену је дат у отвореној спрези (*off-line* ИУУ) и има недостатак да најчешће не може да потисне неочекиване, нерепетитивне поремећаје. За примену ИУУ у реалним применама, за превазилажење поремећаја и постизање добрих перформанси затвореног система управљања најчешће се користи проширени алгоритам ИУУ управљања који садржи управљање у отвореној спрези (*off-line* ИУУ) као и одговарајуће управљање и у повратној спрези (*on-line* ИУУ), [90].



Сл. 3.3. Блок дијаграм итеративног ИУУ управљања  $PD^\alpha$  у повратној спрези

Према томе, за сингуларни систем нецелог реда (3.77) и (3.78) примењује се ИУУ управљање у отворено-затвореној спрези  $P/PD^\alpha$  типа који је дефинисан са, [91]:

$$u_{i+1}(t) = u_i(t) + \Gamma_1 e_i(t) + \Gamma_2 \left( {}_C D_{0,t}^\alpha e_{i+1}(t) + \Pi_2 e_{i+1}(t) \right), \quad (3.83)$$

где су:  $u_i(t)$  -улаз система (вектор управљања);  $y_i(t)$  - излази система у  $i_{th}$  - итерацији;  $e_i(t) = y_d(t) - y_i(t)$  грешка праћења трајекторије у  $i$ -итерацији;  $u_{i+1}(t)$  -је улаз система (вектор управљања) у  $(i+1)_{th}$  итерацији,  $y_d(t) = Cx_d(t)$  представља жељену излазну трајекторију и  $\Gamma_1, \Gamma_2, \Pi_2$  представљају матрице учења одговарајућих димензија у датом алгоритму ИУУ. При томе,  $PD^\alpha$  контролер у затвореној спрези,  $\Gamma_2 \left( {}_C D_{0,t}^\alpha e_{i+1}(t) + \Pi_2 e_{i+1}(t) \right)$  обезбеђује стабилност затвореног система управљања и омогућава задржавање његове грешке стања унутар униформних граница. Следећом Теоремом 2 изведени су довољни услови за конвергенцију предложеног алгоритма ИУУ који је формиран у отвореној-затвореној спрези.

**Теорема 2.** Нека је закон ИУУ (3.83) у отворено-затвореној спрези примењен на посматрани неаутономни сингуларни систем нецелог реда (3.77), (3.78) и нека су испуњене претпоставке  $A, i=1,2,3,4$   $x_i(0) = x_d(0)$ . Ако постоји матрица  $\Gamma_2$  таква да је следећи услов испуњен

$$\| [I - \Gamma_2 C \bar{B}] \| \leq \rho < 1, \quad (3.84)$$

где је  $\bar{B} = (E + B\Gamma_2 C)^{-1} B$ , и матрица  $\Gamma_2$  је таква да за матрицу  $(E + B\Gamma_2 C)$  постоји инверзна матрица, онда када  $i \rightarrow \infty$  границе грешака праћења  $\|x_d(t) - x_i(t)\|$ ,  $\|y_d(t) - y_i(t)\|$ ,  $\|u_d(t) - u_i(t)\|$ , конвергирају асимптотски ка сфери остатка смештеној у координатном почетку.

*Доказ:* Уводе се следеће ознаке

$$\delta h_i = h_d(t) - h_i(t), \quad h = x, x_d, u, u_d, f \quad (3.85)$$

$$D^\alpha (\delta h_i(t)) = \delta h_i^{(\alpha)} = h_d^{(\alpha)}(t) - h_i^{(\alpha)}(t)$$

тако да је грешка праћења одређена са:

$$e_i^{(\alpha)}(t) = \frac{d^{(\alpha)}}{dt^{(\alpha)}} (y_d(t) - y_i(t)) = C \delta x_i^{(\alpha)}(t) \quad (3.86)$$

Са друге стране, ако се узме у обзир предложени закон управљања НРИУУ, произилази да је:

$$\delta u_{i+1} = u_d - u_{i+1} = \delta u_i - \Gamma_1 e_i - \Gamma_2 \left( e_{i+1}^{(\alpha)} + \Pi_2 e_{i+1} \right), \quad (3.87)$$

односно, ако се узме у обзир (3.86) следи

$$\delta u_{i+1} = \delta u_i - \Gamma_1 C \delta x_i - \Gamma_2 C \delta x_{i+1}^{(\alpha)} - \Gamma_2 \Pi_2 C \delta x_{i+1}. \quad (3.88)$$

Такође, на основу (3.77-3.78), као и (3.80) има се:

$$E \delta x_{i+1}^{(\alpha)} = (A + \Delta A) \delta x_{i+1} + B \delta u_{i+1} - \Delta A x_d \quad (3.89)$$

Узимајући у обзир изразе (3.88) и (3.89) следи:

$$(E + B \Gamma_2 C) \delta x_{i+1}^{(\alpha)} = (A + \Delta A - B \Gamma_2 \Pi_2 C) \delta x_{i+1} + B \delta u_i - B \Gamma_1 C \delta x_i - \Delta A x_d. \quad (3.90)$$

Применом погодно одабране матрице појачања  $\Gamma_2$ , и узимајући у обзир раније уведене претпоставке, може се показати да је матрица  $(E + B \Gamma_2 C)$  инвертибилна тј. постоји  $(E + B \Gamma_2 C)^{-1}$ . Даље, множењем на левој страни израза (3.90) инверзном матрицом добија се израз (3.91) као:

$$\delta x_{i+1}^{(\alpha)} = (\bar{A} + \Delta \bar{A}) \delta x_{i+1} + \bar{A}_1 \delta x_i + \bar{B} \delta u_i - \Delta \bar{A} x_d, \quad (3.91)$$

где су уведене и означене следеће величине:

$$\bar{A} = (E + B \Gamma_2 C)^{-1} (A - B \Gamma_2 \Pi_2 C), \bar{B} = (E + B \Gamma_2 C)^{-1} B, \bar{A}_1 = (E + B \Gamma_2 C)^{-1} B \Gamma_1 C. \quad (3.92)$$

$$\Delta \bar{A} = (E + B \Gamma_2 C)^{-1} \Delta A.$$

После замене (3.91) у израз (3.88) добија се:

$$\delta u_{i+1} = [I - \Gamma_2 C \bar{B}] \delta u_i - [\Gamma_2 C (\bar{A} + \Delta \bar{A}) + \Gamma_2 \Pi_2 C] \delta x_{i+1} - [\Gamma_1 C + \Gamma_2 C \bar{A}_1] \delta x_i + \Gamma_2 C \Delta \bar{A} x_d \quad (3.93)$$

Естимацијом претходног израза са одговарајућом нормом  $\|(\cdot)\|$  узимајући у обзир истовремено и услов Теореме 2, следи:

$$\begin{aligned} \|\delta u_{i+1}\| &\leq \rho \|\delta u_i\| + \left\| [\Gamma_2 C (\bar{A} + \Delta \bar{A}) + \Gamma_2 \Pi_2 C] \right\| \|\delta x_{i+1}\| + \left\| [\Gamma_1 C + \Gamma_2 C \bar{A}_1] \right\| \|\delta x_i\| + \|\Gamma_2 C \Delta \bar{A}\| \|x_d\| = \\ &= \rho \|\delta u_i\| + \beta_0 \|\delta x_{i+1}\| + \beta_1 \|\delta x_i\| + \beta_2 \|x_d\| \end{aligned} \quad (3.94)$$

Са друге стране, узимајући у обзир претпоставке А4, решење диференцијалне једначине стања нецелог реда (3.91) се може приказати у облику Волтерове интегралне једначине на следећи начин:

$$\delta x_{i+1}(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \left( [\bar{A} + \Delta \bar{A}] \delta x_{i+1}(s) + \bar{A}_1 \delta x_i(s) + \bar{B} \delta u_i(s) \right) ds - \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \Delta \bar{A} x_d(s) ds \quad (3.95)$$



На сличан начин, применом норме  $\|(\cdot)\|$  на (3.91) и узимајући у обзир услов јединствености решења има се:

$$\begin{aligned} \|\delta x_{i+1}(t)\| &\leq \frac{(a+a_\Delta)}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|\delta x_{i+1}(s)\| ds + \frac{a_1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|\delta x_i(s)\| ds \\ &+ \frac{b}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|\delta u_i(s)\| ds + \frac{a_\Delta}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|x_d(s)\| ds \end{aligned} \quad (3.96)$$

где су  $a = \|\bar{A}\|$ ,  $b = \|\bar{B}\|$ ,  $a_\Delta = \|\Delta \bar{A}\|$ . Са друге стране, у важности је и следећи израз:

$$t \in [0, T], \quad \|\delta x_i(t)\| = \|\delta x_{i+1}(t) + x_{i+1} - x_i(t)\| \leq \|\delta x_{i+1}(t)\| + \|x_{i+1} - x_i(t)\| \quad (3.97)$$

Ако се уведе следећа величина  $\eta_{i+1} = \sup_{t \in [0, T]} \|x_{i+1} - x_i(t)\|$ , онда (3.97) поприма следећи облик:

$$\|\delta x_i(t)\| \leq \|\delta x_{i+1}(t)\| + \eta_{i+1} \quad (3.98)$$

Ако се претходно добијени изрази замене у (3.96) произилази:

$$\begin{aligned} \|\delta x_{i+1}(t)\| &\leq \frac{(a+a_1+a_\Delta)}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|\delta x_{i+1}(s)\| ds + \frac{a_1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \eta_{i+1} ds \\ &+ \frac{b}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|\delta u_i(s)\| ds + \frac{a_\Delta \cdot c}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} ds \\ &\leq \frac{(a+a_1+a_\Delta)}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|\delta x_{i+1}(s)\| ds + \frac{b}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \|\delta u_i(s)\| ds + \\ &+ (a_1 \cdot \eta_{i+1} + a_\Delta \cdot c) \frac{t^\alpha}{\Gamma(\alpha+1)} \end{aligned} \quad (3.99)$$

где је испуњено  $\|x_d(t)\| \leq c$ ,  $\forall t \in [0, T]$ . Даљом применом  $\lambda$  норме на обе стране претходног израза може се добити да је:

$$\begin{aligned} \|\delta x_{i+1}(t)\|_\lambda &\leq \sup_{0 \leq t \leq T} \left\{ \int_0^t e^{-\lambda t} \frac{|(t-s)|^{\alpha-1}}{\Gamma(\alpha)} \left[ (a+a_1+a_\Delta) \|\delta x_{i+1}(s)\| + b \|\delta u_i(s)\| \right] ds \right\} + \sup_{0 \leq t \leq T} \left( e^{-\lambda t} \frac{(a_1 \eta_{i+1} + a_\Delta \cdot c) t^\alpha}{\Gamma(\alpha+1)} \right) \\ &\leq \sup_{0 \leq t \leq T} \int_0^t e^{-\lambda(t-s)} \frac{|(t-s)|^{\alpha-1}}{\Gamma(\alpha)} \sup_{0 \leq s \leq T} e^{-\lambda s} \left[ (a+a_1+a_\Delta) \|\delta x_{i+1}(s)\| + b \|\delta u_i(s)\| \right] ds + \left( \frac{a_1 \eta_{i+1} + a_\Delta \cdot c}{\Gamma(\alpha+1)} \right) e^{-\lambda t^*} \cdot t^{*\alpha}, \end{aligned} \quad (3.100)$$

$$t^* \in [0, T]$$

$$\leq (a_\Sigma \|\delta x_{i+1}(t)\|_\lambda + b \|\delta u_i(t)\|_\lambda) \cdot \sup_{0 \leq t \leq T} \int_0^t e^{-\lambda(t-s)} ds \sup_{0 \leq t \leq T} \int_0^t \frac{|(t-s)|^{\alpha-1}}{\Gamma(\alpha)} ds + \xi.$$

где су  $a_\Sigma = a + a_1 + a_\Delta$ ,  $\xi = (a_1 \eta_{i+1} + a_\Delta \cdot c) / \Gamma(\alpha+1)$ . Даљим сређивањем, добија се

$$\|\delta x_{i+1}(t)\|_{\lambda} \leq (a_{\Sigma} \|\delta x_{i+1}(t)\|_{\lambda} + b \|\delta u_i(t)\|_{\lambda}) \cdot \frac{(1-e^{-\lambda T})}{\lambda} \frac{T^{\alpha}}{\Gamma(\alpha+1)} + \xi. \quad (3.101)$$

Уводи се следећу смену  $O(\lambda^{-1})$  дефинисану са:

$$O(\lambda^{-1}) = \frac{(1-e^{-\lambda T})}{\lambda} \frac{T^{\alpha}}{\Gamma(\alpha+1)}, \quad (3.102)$$

и заменом у израз (3.101) он поприма следећи облик:

$$\|\delta x_{i+1}(t)\|_{\lambda} \leq (a_{\Sigma} \|\delta x_{i+1}(t)\|_{\lambda} + b \|\delta u_i(t)\|_{\lambda}) \cdot O(\lambda^{-1}) + \xi, \quad (3.103)$$

тако да се може закључити да важи:

$$\|\delta x_{i+1}(t)\|_{\lambda} \leq \frac{bO(\lambda^{-1})\|\delta u_i(t)\|_{\lambda} + \xi}{(1-a_{\Sigma}O(\lambda^{-1}))} \leq O_{\gamma}(\lambda^{-1})\|\delta u_i(t)\|_{\lambda} + \xi'(\lambda^{-1}). \quad (3.104)$$

Сада, могуће је да се изабере довољно велика вредност за  $\lambda$ , тако да се лако доказује да је испуњен следећи услов:

$$\lambda \Gamma(\alpha+1) - a_{\Sigma} (1-e^{-\lambda T}) T^{\alpha} > 0. \quad (3.105)$$

С друге стране ако се израз (3.98) замени у израз (3.94) следи:

$$\|\delta u_{i+1}\| \leq \rho \|\delta u_i\| + (\beta_0 + \beta_1) \|\delta x_{i+1}\| + \beta_1 \cdot \eta_{i+1} + \beta_2 \cdot c \quad (3.106)$$

Такође, применом  $\lambda$  норме на претходни израз добија се:

$$\|\delta u_{i+1}\|_{\lambda} \leq \rho \|\delta u_i\|_{\lambda} + \beta' \|\delta x_{i+1}\|_{\lambda} + \beta'' \quad (3.107)$$

Заменом израза (3.104), у (3.107) следи:

$$\|\delta u_{i+1}\|_{\lambda} \leq (\rho + \beta' O_{\gamma}(\lambda^{-1})) \|\delta u_i\|_{\lambda} + \beta' \xi'(\lambda^{-1}) + \beta'' = \rho' \|\delta u_i\|_{\lambda} + \varepsilon. \quad (3.108)$$

Осим тога, ако се узме довољно велика вредност за  $\lambda$  онда је следећи услов испуњен:

$$\rho' = (\rho + \beta' O_{\gamma}(\lambda^{-1})) < 1. \quad (3.109)$$

Применом Леме 2 [92] на (3.108) следи:

$$\lim_{i \rightarrow \infty} \|\delta u_i\|_{\lambda} \leq \frac{1}{1-\rho'} \varepsilon. \quad (3.110)$$

чиме је доказ Теореме 2 завршен.

### 3.3.3 Симулациони резултати

Да би се илустровала успешност и ефикасност предложеног алгоритма управљања- ( $P/PD^\alpha$ ) тип ИУУ у отворено-затвореној спрези разматра се следећи пример сингуларног система нецелог реда који је дат са:

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} D^{0.5} x_1(t) \\ D^{0.5} x_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad (3.111)$$

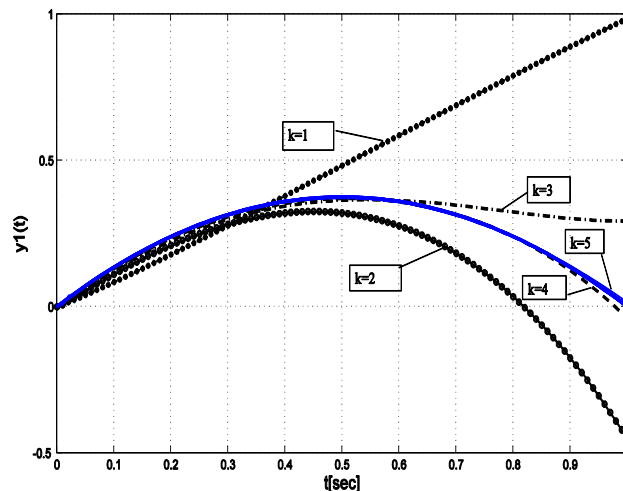
$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad (3.112)$$

где су:  $\Delta A = 0, t \in [0,1], \alpha = 0.5$ . Жељена трајекторија излаза је дата са  $y_{d1}(t) = 1.5t \cdot (1-t), y_{d2}(t) = 0.5t^2, y_{d1,2}(0) = y_{i1,2}(0) = 0$ .

Да би се одредиле вредности матрица појачања, неопходно је испунити услов конвергенције из Теореме 2 уз истовремено испуњења брзине конвергенције истог. Како је добијени довољан услов конвергенције датог алгоритма управљања, онда корисник може матрице појачања одредити применом методе покушаја и грешке:

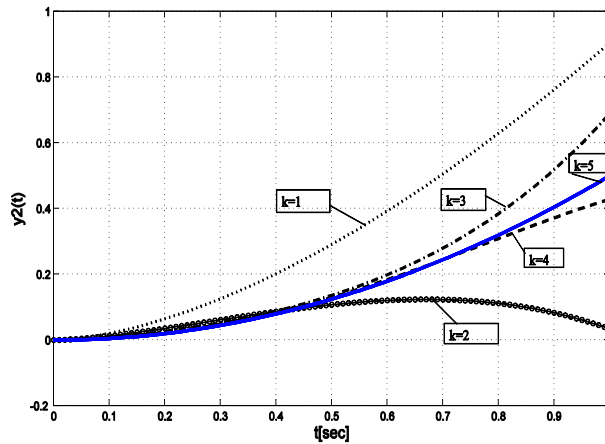
$$\Gamma_1 = \begin{bmatrix} 0.51 & 0.5 \\ 0 & 0.51 \end{bmatrix}, \Gamma_2 = \begin{bmatrix} 0.96 & 0.41 \\ 0 & 0.96 \end{bmatrix}, \Pi_2 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}. \quad (3.113)$$

Лако се показује да је услов конвергенције Теореме T2 испуњен, тј.  $\| [I - \Gamma_2 C \bar{B}] \| = 0.73 < 1$ , односно да је пар (E;A) регуларан.



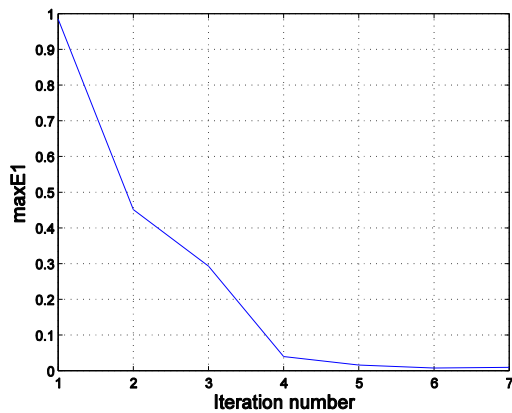
Слика 3.4. Перформанса праћења излаза система

( $y_1(t)$  -испрекидана линија, ( $k=1,2,3,4,5..$ ) број итерација,  $y_{d1}(t)$  - пуна линија)

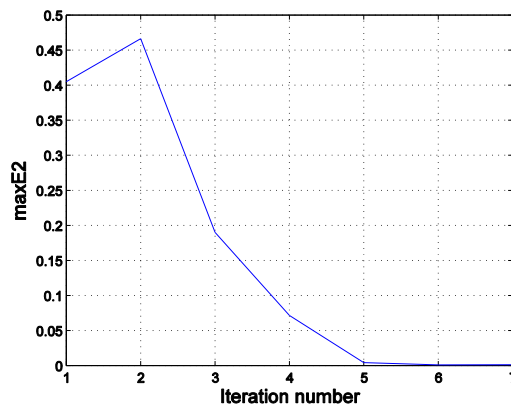


Слика 3.5. Перформанса праћења излаза система

( $y_2(t)$  -испрекидана линија, ( $k = 1, 2, 3, 4, 5..$ ) број итерација,  $y_{d2}(t)$  - пуна линија)



Слика 3.6. Суп-норма грешке праћења  $e_1(t)$  у функцији броја итерација



Слика 3.7. Суп-норма грешке праћења  $e_2(t)$  у функцији броја итерација

Симулациони резултати дати су на сликама 3.4.-3.7. и они веома јасно показују оправданост примене и ефикасност предложеног алгоритма ИУУ управљања нецелог реда у отворено-затвореној спрези. На основу слика 3.6., 3.7. уочава се да се жељена тачност праћења достиже већ после седам итерација.

### 3.4. Итеративно управљање учењем у отворено-затвореној спрези нецелог реда $PD^u/PD$ типа NeuroArm роботским системом

#### 3.4.1. Математички модел NeuroArm роботског система

У роботици се за реализацију кретања роботског система користе најчешће мотори једносмерне струје као погонски мотори. С обзиром на то да они обезбеђују велике угаоне брзине при малом обртном моменту а код кретања роботских сегмената захтевају се мале брзине а велики обртни моменти, убацују се преносници снаге (редуктори) између актуатора и роботских сегмената. За ротационе зглобове најчешће се користе зупчаници као преносници.

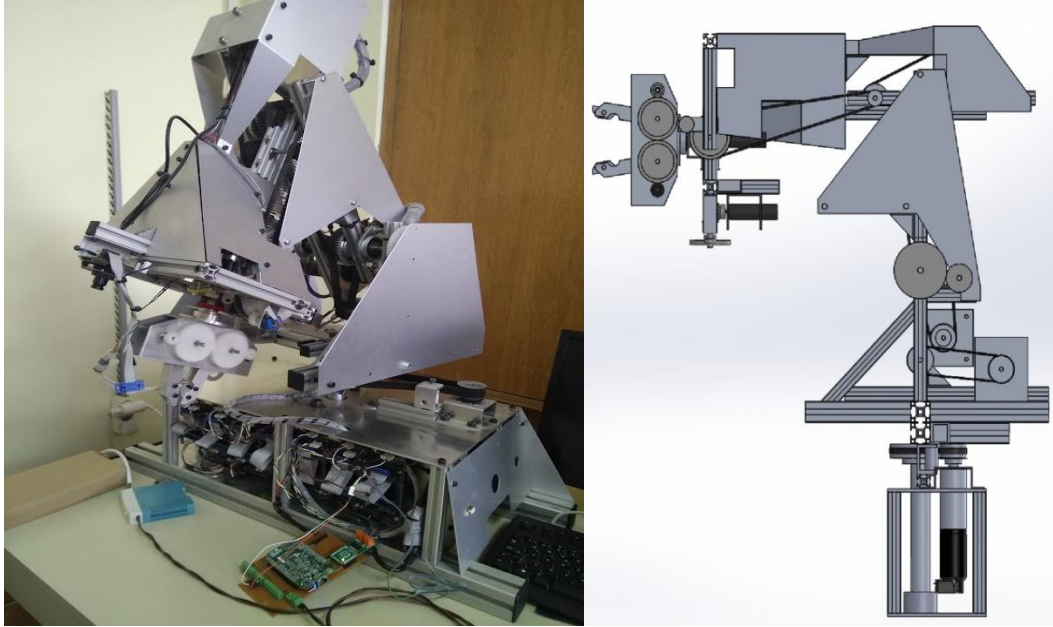
Погонска сила и погонски спрег сила се могу сматрати управљачким величинама и њихове вредности зависе од динамичких особина актуатора и редуктора. Стога, потребно је узети у обзир њихов утицај да би се добио комплетнији и прецизнији математички модел разматраног роботског манипулатора.

NeuroArm роботски систем, који је саставни део Лабораторије за примењену механику Машинског факултета у Београду, поседује седам степени слободе. Прва три ротациона зглоба су одговорна за постављање хваталке у жељени положај (позиционирање), док следећа три зглоба чине сферни зглоб чиме се постиже потпуна оријентација роботске хваталке. Седми степен слободе поседује сама хваталка. У циљу примене и приказа ефикасности предложеног алгорита ИУУ даље се разматра NeuroArm роботска рука са прва три ротациона зглоба, тј. роботски систем који има три степена слободе, без губитка општости.

Механичка структура работа NeuroArm се може моделовати као кинематички ланац без гранања крутих сегмената међусобно повезаних помоћу зглобова. Динамичке једначине роботског система могу се написати у следећем кондензованом облику:

$$A(q)\ddot{q} + C(q, \dot{q})\dot{q} - Q^g = Q^m, \quad (3.114)$$

где су:  $q(t) \in R^3$  вектор генералисаних координата,  $A(q) \in R^{3 \times 3}$  представља основни метрички тензор (или матрицу инерције),  $C(q, \dot{q}) \in R^{3 \times 3}$  је матрица која укључује центрифугалне и Кориолисове ефекте,  $Q^g \in R^3$  и  $Q^m \in R^3$  су вектор генералисаних сила од сила гравитације роботског система и вектор генералисаних сила од погонских момената примењени на зглобове. За детаље прорачуна основног метричког тензора и матрице  $C(q, \dot{q})$  за роботске манипулаторе, погледати [50].



Слика 3.8. а) NeuroArm робот манипулатор са 7 степени слободe б) Механичка структура NeuroArm робота

Такође, потребно је узети у обзир и динамику постојећих мотора једносмерне струје. С обзиром на крутост роботских сегмената, следеће једначине описују преносни однос зупчаника:

$$\mathbf{q}_m = N\mathbf{q}, \quad \mathbf{Q}^m = N\mathbf{M}_l, \quad (3.115)$$

где  $\mathbf{q}_m$  представља положаје осовина актуатора,  $N$  је  $3 \times 3$  дијагонална матрица преносних односа зупчаника,  $\mathbf{M}_l$  је вектор обртних момената који проистиче из робота манипулатора и деловања на осовине мотора. Може се лако показати [93] да је обртни моменат  $M_l$  једнак:

$$\mathbf{M}_l = (N^2)^{-1} (A(\mathbf{q})\ddot{\mathbf{q}}_m + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_m) - N^{-1}\mathbf{Q}^g \quad (3.116)$$

Обртни момент  $M_l$  се може сматрати поремећајем који делује на осовину мотора. Као што видимо из (3.116), утицај овог поремећаја на динамику мотора опада са повећањем коефицијента редуције  $N$ . То значи да присуство високог коефицијента смањења тежи линеаризовању динамичких једначина робота. Другим речима, можемо занемарити нелинеарне спреге између мотора различитих веза и уместо тога користити линеарни модел.

Модел мотора једносмерне струје састоји се од механичког и електричног дела, и може се описати следећим једначинама:

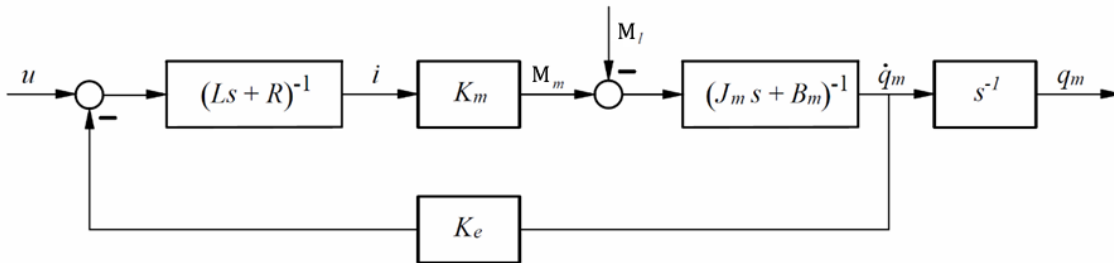
$$\mathbf{J}_m\ddot{\mathbf{q}}_m + \mathbf{B}_m\dot{\mathbf{q}}_m = \mathbf{M}_m - \mathbf{M}_l, \quad (3.117)$$

$$R_m \mathbf{i} + L \frac{d\mathbf{i}}{dt} + K_e \frac{dq_m}{dt} = \mathbf{u}, \quad (3.118)$$

Где је  $J_m$   $3 \times 3$  дијагонална матрица која садржи ефективне инерције мотора,  $B_m$  је  $3 \times 3$  дијагонална матрица која садржи коефицијенте вискозног трења мотора,  $M_m$  је вектор обртних момената актуатора,  $R_m$  је  $3 \times 3$  дијагонална матрица која садржи отпоре арматурних кругова,  $\mathbf{i} \in R^3$  представља вектор струје арматуре,  $L$  је дијагонална матрица индуктивитета арматуре,  $K_e$  је дијагонална матрица која садржи константе контраелектромоторне силе (back EMF),  $\mathbf{u}$  је вектор улазних напона арматуре. Веза између вектора обртног момента  $M_m$  и вектора струје арматуре  $\mathbf{i}$  дата је као:

$$M_m = K_m \mathbf{i}, \quad (3.119)$$

где је  $K_m$   $3 \times 3$  дијагонална матрица која садржи константе обртног момента мотора. Комплетан блок дијаграм динамике мотора приказан на слици 3.9.



Слика 3.9. Блок дијаграм мотора једносмерне струје

Посматрајући горњи дијаграм, однос између управљачког улаза  $u$  и излаза вратила мотора  $q_m$  можемо приказати помоћу следеће преносне функције:

$$G_p(s) = \frac{k_m}{(\ell s + r)(j_m s + b_m) + k_m k_e} \frac{1}{s}. \quad (3.120)$$

NeuroArm робот користи Махон RE36 (70W) моторе једносмерне струје за контролисање позиције прве 3 везе. Из техничке спецификације произвођача:  $j_m = 65.2 \text{ gcm}^2$ ,  $r = 1.71 \Omega$ ,  $\ell = 0.89 \text{ mH}$ ,  $k_m = 44.5 \text{ mNm/A}$ ,  $k_e = 1/215 \text{ V/rpm}$ , преносна функција  $G_p(s)$  постаје:

$$G_m(s) = \frac{K}{s(Ts + 1)}. \quad (3.121)$$

где је  $K=22.515$  и  $T=0.0056409$ . Преносна функција трећег реда (3.120) је сведена на модел другог реда дат у (3.121) јер се један од три пола  $G_p(s)$  налази крајње лево у  $s$ -равни и његов утицај се може занемарити. Коначно, оригинални нелинеарни

роботски систем, због високог коефицијента редукције ( $N = \text{diag}\{185, 230, 74\}$ ), резултирао је у 3 линеарна, раздвојена подсистема, описана преносном функцијом мотора  $G_m(s)$ .

Шема напредног управљачког система робота NeuroArm приказана је на слици 3.10. Као што је претходно речено, управљачки сигнал је подељен у два дела, први који се налази у отвореној (директној) спрузи и други у повратној спрузи. Овде се за управљање у отвореној спрузи предлаже закон ИУУ  $PD^\mu$  типа нецелог реда, док се повратна спруза састоји од класичног ПД алгоритма управљања. Укупни вектор управљања је сада:

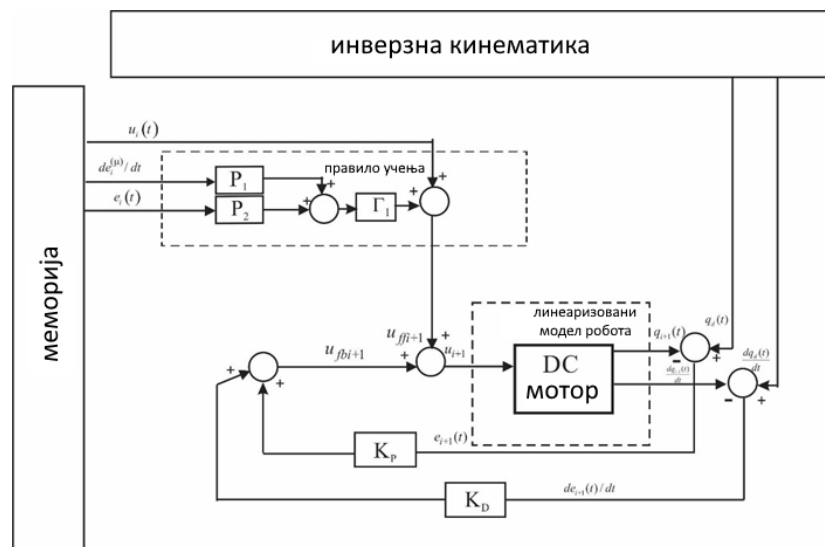
$$u_{i+1}(t) = u_{ff\bar{i}+1}(t) + u_{fb\bar{i}+1}(t), \quad (3.122)$$

где су  $u_{ff\bar{i}+1}(t)$  и  $u_{fb\bar{i}+1}(t)$  два горе поменута дела управљачког сигнала у  $i + 1$  итерацији. Конкретније, можемо написати следеће:

$$u_{ff\bar{i}+1}(t) = u_i(t) + \Gamma_1 \left( P_1 e_i^{(\mu)}(t) + P_2 e_i(t) \right) \quad (3.123)$$

$$u_{fb\bar{i}+1}(t) = K_d \dot{e}_{i+1}(t) + K_p e_{i+1}(t) \quad (3.124)$$

где је  $q_d(t)$  жељена излазна путања,  $e_i(t) = q_d(t) - q_{m,i}(t)$  и  $e_{i+1}(t) = q_d(t) - q_{m,i+1}(t)$  су грешке у праћењу путање у  $i$ -тој итерацији, односно  $i + 1$  итерацији. Такође,  $P_1, P_2, \Gamma_1, K_d, K_p \in R^{3 \times 3}$  су позитивно дефинитне матрице у отвореној и повратној спрузи;  $e_i^{(\mu)}(t)$  у (3.122) представља извод грешке из  $\mu$ -тог реда. Довољан услов конвергенције предложеног закона управљања ИУУ се добија на сличан начин као у поглављу 3.2.





Слика 3.10. Блок дијаграм напредног управљања у директној грани и са повратном спрегом  $PD^u/PD$  контролног система

Матрице за учење  $P_1$ ,  $P_2$  и  $\Gamma_1$  добијају се методом покушаја и грешака, док избор матрица  $K_d$  и  $K_p$  обавља на следећи начин. Закон управљања у повратној спрези за сваки од три мотора који покрећу роботску руку, [94], може се изразити следећом преносном функцијом:

$$C(s) = \frac{k_d s + k_p}{T_f s + 1}, \quad (3.125)$$

где је  $T_f$  временска константа филтра,  $k_d$  и  $k_p$  су коефицијенти диференцијалног и пропорционалног дејства, респективно. Метод подешавања који је изведен у наставку детаљно је дат у [95]. У овом раду, параметри ПД контролера изведени су посебно за  $G_m(s)$  дато у (3.121), с обзиром да је нелинеарна динамика работа еквивалентна том линеарном моделу. За почетак, почињемо са комплементарном преносном функцијом  $T(s)$ , [96]:

$$T(s) = \frac{\eta_1 s + \eta_0}{(\lambda s + 1)^3}, \quad (3.126)$$

где су временска константа  $\lambda > 0$ ,  $\eta_1$  и  $\eta_0$  слободни параметри који ће се одредити ради добијања жељених динамичких карактеристика система затворене петље. Имајући у виду да се  $T(s)$  може формулисати као  $T(s) = C(s)G_m(s)/(1 + C(s)G_m(s))$ , онда се, након неких прорачуна, добија регулатор у следећем облику:

$$C(s) = \frac{\eta_1 s + \eta_0}{(\lambda s + 1)^3 - \eta_1 s - \eta_0} \frac{1}{G_m(s)}, \quad (3.127)$$

Слободни параметри  $\eta_1$  и  $\eta_0$  одређују се како би се поништили полови  $G_m(s)$ , и те вредности су дате као:

$$\eta_0 = 1, \quad \eta_1 = \frac{\lambda}{T^2} [3T^2 - \lambda(3T - \lambda)] \quad (3.128)$$

Коначно, параметри ПД регулатора  $C(s)$  су дефинисани:

$$k_p = \frac{T^2}{(3T - \lambda)K\lambda^2}, \quad k_d = \frac{3T^2 - (3T - \lambda)\lambda}{(3T - \lambda)K\lambda}, \quad T_f = \frac{T\lambda}{3T - \lambda}. \quad (3.129)$$

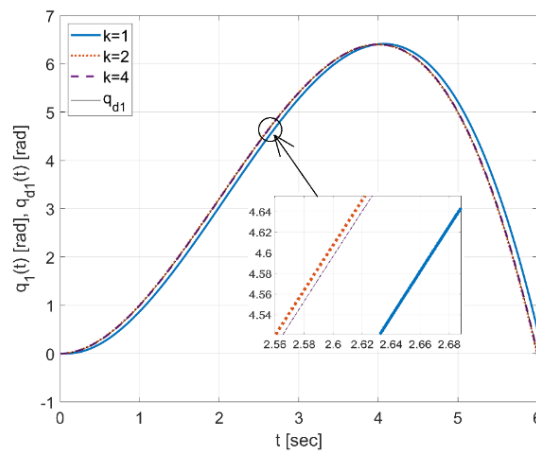
Формуле за подешавање (3.129) садрже један слободан параметар: временску константу  $\lambda$ . Посматрајући горње релације, можемо закључити да је одзив система бржи што је  $\lambda$  мањи. Дакле, подешавањем  $\lambda$  се може постићи врло добар компромис

између робусности и перформанси система, што је кључна ставка у дизајнирању модерних управљачких система.

### 3.4.2 Резултати симулације

Представљени су резултати симулације да би се илустровала ваљаност предложеног напредног ИУУ алгоритма  $PD^\mu/PD$  типа. Жељене излазне трајекторије роботских система су следеће:

$$\begin{aligned} q_{d1} &= 0.2 \cdot t^2 (6-t) [\text{rad}], & q_{d2} &= 0.25 \cdot \cos(t) [\text{rad}], \\ q_{d3} &= 0.2 \cdot \sin(2t) [\text{rad}], & \forall t &\in [0, 6 \text{ sec}]. \end{aligned} \quad (3.130)$$



Слика 3.11. Перформансе праћења прве везе робота кроз 1., 2. и 4. итерацију и  $q_{d1}(t)$

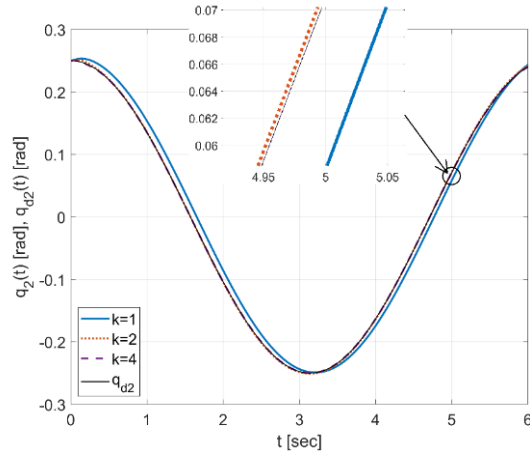
Усвојене су следеће вредности матрица учења.

$$P_1 = \text{diag}(1,1,1), \quad P_2 = \text{diag}(1,1,1), \quad \Gamma_1 = \text{diag}(0.1, 0.01, 0.1). \quad (3.131)$$

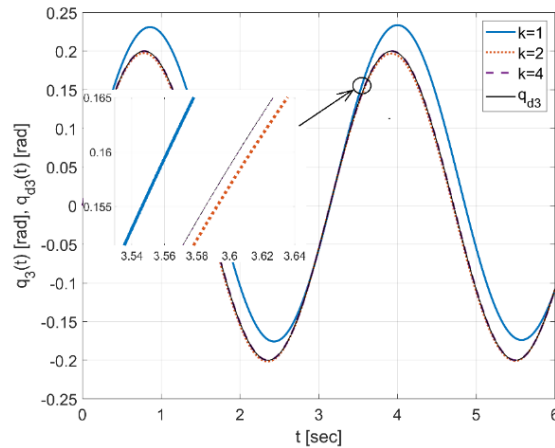
Параметри регулатора одређени су у складу са (3.129) како би се добио најбољи компромис између робусности и перформанси затвореног система аутоматског управљања:

$$K_p = \text{diag}(1,3165; 1,3165; 1,3165), \quad K_d = \text{diag}(0.2255; 0.2255; 0.2255) \quad (3.132)$$

Константа филтра израчунава се као  $T_f = 0.2714$ . За све одабране параметре приказане изнад и за  $\mu = 1.1$ , а за резултате симулације приказани су на сликама 3.11-3.13. Сlike показују да управљачки систем успешно прати излазне путање након само неколико итерација.

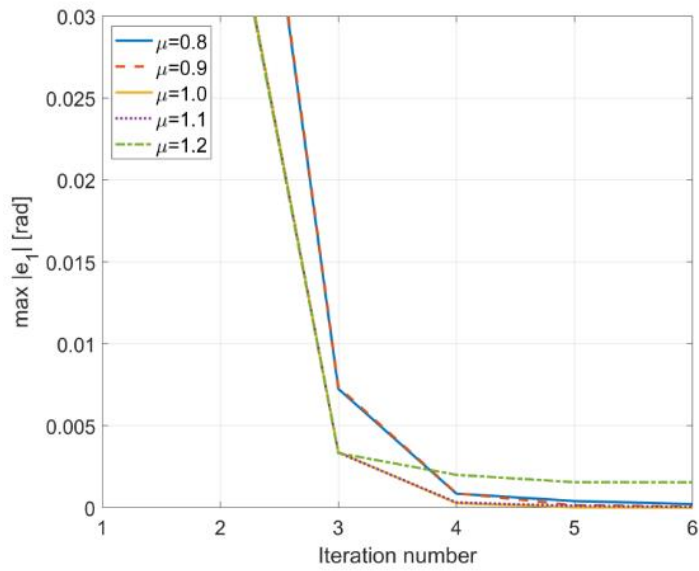


Слика 3.12. Перформансе праћења друге везе робота кроз 1., 2. и 4. итерацију и  $q_{d1}(t)$

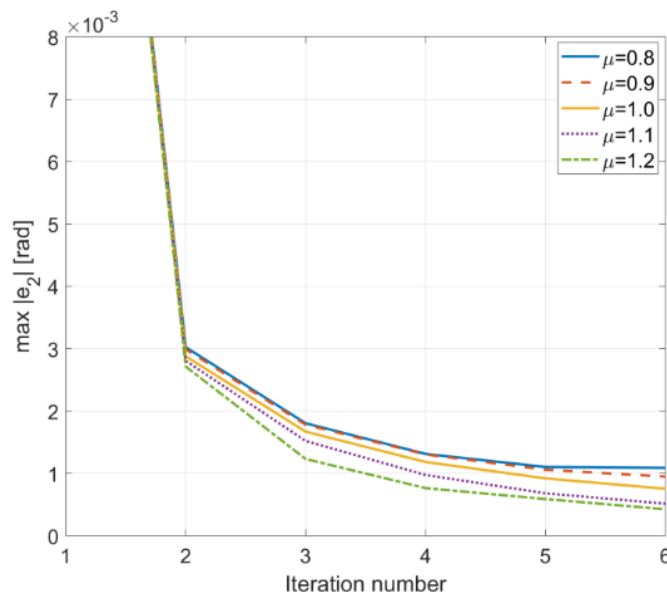


Слика 3.13. Перформансе праћења друге везе робота кроз 1., 2. и 4. итерацију и  $q_{d1}(t)$

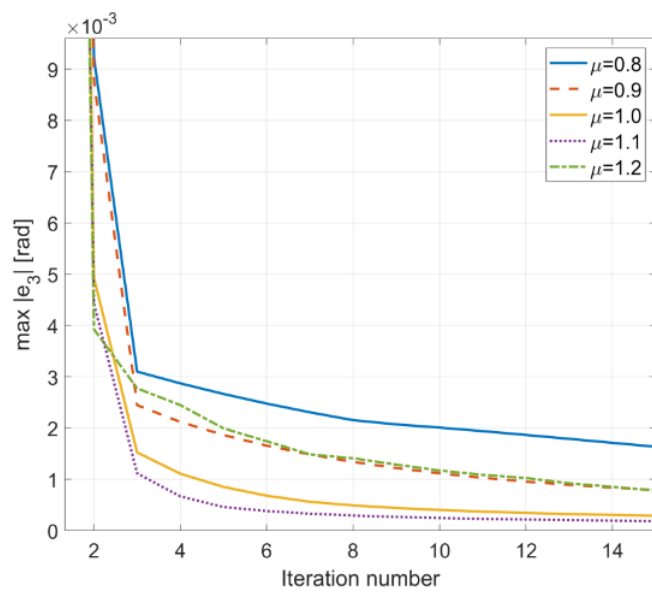
Сада, да бисмо приказали утицај нецелог извода  $\mu$  на грешку праћења, спроведена је следећа симулација. Постепено је мењана вредност параметра  $\mu$  од 0,8 до 1,2 са циљем опажања његовог утицаја на сам одзив система. Резултати су приказани на сликама 3.14.-3.16. у смислу максималне апсолутне грешке праћења за сваки роботски сегмент. Може се видети да се грешке у праћењу смањују са повећањем броја итерација. Такође, најбољи резултати у погледу брзине конвергенције излазног сигнала добијају се за вредност  $\mu=1,1$  за први и трећи роботски сегмент, а  $\mu=1,2$  за други сегмент.



Слика 3.14. Тенденција промене максималне грешке  $e_1(t)$  за први сегмент у функцији промене броја итерација



Слика 3.15. Тенденција промене максималне грешке  $e_2(t)$  за други сегмент у функцији промене броја итерација



Слика 3.16. Тенденција промене максималне грешке  $e_3(t)$  за трећи сегмент у функцији промене броја итерација

## 4. Хардверско управљање NeuroArm роботском руком

### 4.1. Ембедид системи

Ембедид системи (*embedded systems*) данас се могу наћи како у најширој потрошачкој електроници (мобилним телефонима, таблетима, фотографским апаратима, телевизорима, кућним системима за аутоматизацију и сигурност, беле технике), па тако и у аутомобилима, авионима, фабричким постројењима,... Развојем технике и нових захтева које диктира тржиште дошло је до наглог развоја оваквих система који пред собом морају да извршавају веома комплексне задатке и у критичним ситуацијама да извршавају предефинисане задатке у тачно одређеним временским оквирима.

То је условило развој и хардверског и софтверског дела ових система [97], [98], [99]. Код хардверског дела највише је напредовала архитектура процесора који су из године у годину знатно бржи а троше мање енергије од претходних генерација процесора. Архитектура процесора која се користи у оваквим системима је ARM (развио Acorn computers 1980.) и то 32-битна или 64-битна, с тим да се иде у будућности ка напуштању 32-битне архитектуре због ограничења која та архитектура има. Код софтверског дела оперативни систем Linux као основа других напреднијих графичких система (Android и MacOS) знатно напредује сваке године, као и захваљујући *open source* заједници (заједници која подржава развој тиме што је код доступан свима за имплементацију).

Захтеви апликација дефинишу основне карактеристике RTOS-а (Real Time Operating System – оперативни систем са извршавањем у реалном времену)

- *Предвидивост*: Најчешће ембедид системи су системи који раде у реалном времену, неопходно је испуњење временских захтева да би се обезбедио номиналан рад система. Без обзира на број задатака, мора се обезбедити извршење задатака помоћу алгоритма предвидивости.
- *Поузданост*: У зависности од намене, неопходно је да систем без интервенција ради поуздано у дужем временском интервалу без икаквих интервенција споља.
- *Перформансе*: Намена ембедид система одређује које је перформансе неопходно да систем поседује. Перформансе се могу постићи коришћењем бржег процесора или оптимизацијом софтвера и алгоритама програма.
- *Хардверска независност*: Коришћење оперативног система вишег нивоа омогућава, тј. олакшава рад са великим бројем уређаја који се повезују на саму плочу путем неког од интерфејса повезивања на тај начин што се користе вез написани „драјвери“ за рад са таквим уређајима. Пример таквих уређаја су монитори, тастатуре, мишеви, штампачи,...

- *Скалабилност*: У зависности од функционалности која се захтева за дати систем, мора постојати што једноставнија могућност додавања или уклањања одређених модула.
- *Компактност*: Намена система може условити то да систем мора да буде изузетно компактан да би се уклопио у захтевано окружење.

Са друге стране, научна истраживања у примењеној роботизи се све више базирају на употреби различитих ембедид система због горе наведених разлога и појављује се све више научних радова који користе неке од најпознатијих и најраспрострањенијих ембедид система као што су Raspberry Pi и Arduino развојне плоче.

Тако један пример јесте аутономни робот који користи Raspberry Pi развојну плочу за наношење боје и лакова [100] у срединама где су боје и лакови канцерогени чиме се спречава угрожавање здравља људи на тим пословима. Други пример је аутономни робот који користи Raspberry Pi развојну плочу [101] и који се користи за надгледање и контролу граница путем камера и сензора. У овом случају користи се Raspbian оперативни систем и са контролним центром се комуницира путем интернета. Робот не мора бити у потпуности аутономан, већ га оператер може контролисати путем интернета и задавати му неке од предефинисаних радњи или у потпуности преузети управљање њим. Raspberry Pi као развојна плоча коју покреће Linux оперативни систем се лако повезује на интернет (или на локалну мрежу) тако да се штеди време на развоју драјвера и користе се већ постојећи. Arduino развојна плоча се може користити за робота који се користи у спасилачке сврхе који је опремљен разним сензорима за детекцију температуре и избегавање опасних подручја које због високе температуре могу оштетити робота.

## 4.2. Избор развојне плоче за управљање NeuroArm роботском руком

*NeuroArm* роботска рука у својој изворној конфигурацији долази са интегрисаним ATmega 128 процесорима који подржава I2C и SPI индустријске протоколе комуникације. Међутим, процесори су фабрички препрограмирани ПИД алгоритмом који није могуће мењати. Зато је употребна вредност оваквог управљачког система знатно умањена у научне сврхе, јер је практично немогуће имплементисати и тестирати нове алгоритме управљања.

Зато смо се одлучили на промену управљачког система за систем који је модуларан, подржан од стране Open source заједнице и који омогућава једноставнију имплементацију било ког алгоритма управљања. Разлог за такво размишљање је тај што је роботски систем NeuroArm саставни део Лабораторије за примењену механику Машинског факултета у Београду и циљ је да се тај систем у пуном капацитету искористи у настави из предмета „Механика робота“ где ће студентима

бити презентовано како раде индустријски роботи манипулатори у различитим гранама индустрије једноставном променом програма управљачког система. Како се нпр. променом коефицијената ПИД алгоритма мења и само понашање робота и како се прилагођавају параметри за добијање оптималног управљања. Такође, нова верзија NeuroArm роботске руке треба да послужи у научним истраживањима, односно студентима на мастер и докторским студијама за експериментално тестирање нових теорија различитих алгоритама управљања, који се испрограмирају и тестирају у Matlab и Simulink програмским пакетима [102] и тестирају и касније потврђују на NeuroArm роботу.

Могућности које би нови управљачки систем [103] морао да подржава су:

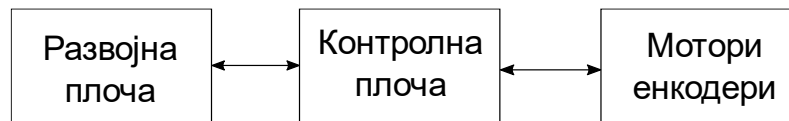
- *Модуларност*: Потребно је да је цео систем модуларног типа да би се нове компоненте могле лако додавати чиме би било омогућено да систем извршава нове задатке.
- *Лака надоградња*: Да поседује најмање 2 USB порта што јер то омогућава повезивање независних компоненти путем овог универзалног прикључка. Оперативни систем такође мора да се лако надограђује на нове верзије без нарушавања постојеће операбилности, с обзиром да нове верзије доносе исправке bug-ова, подршку за нове уређаје који се могу повезати путем USB портова.
- *Компатибилност*: Уколико се користи развојна плоча истог произвођача да она буде pin2pin компатибилна са постојећом, тако да се само заменом плоче омогуће боље перформансе у извршавању задатака.
- *Мрежни приступ*: Са хардверске стране неопходно је да управљачка јединица подржава 2 вида приступа мрежи, односно да поседује Етернет порт RJ-45 (обавезно због једноставности и смањеног одзива приликом управљања) и омогућавања директне даљинске контроле. Такође потребно је да поседује подршку за приступ мрежи бежичним путем преко WiFi-а (што олакшава приступ јер није потребно директно се качити на саму управљачку јединицу и због једноставности повезивања са другим уређајима који подржавају WiFi приступ). Веома битна опција и Bluetooth конекција јер многи сензори омогућавају повезивање искључиво путем ове конекције. Уколико сам контролер поседује USB портове могуће је додавање Bluetooth адаптера тако да није неопходно да поседује уграђену конекцију овог типа.
- *Једноставност управљања*: Оперативни систем треба да буде довољно визуелно познат да би корисник могао лако да се снађе, а програмски алат под којим се врши програмирање управљања треба да буде на начин на који је учено о програмирању индустријских робота на другим предметима.
- *Вишак процесорске снаге*: Неопходно је да се сви задаци извршавају или тренутно или у најкраћем могућем времену, тако да је битно да процесор



има довољно снаге за задатке за које у овом тренутку знамо да ће бити неопходно извршавати, као што је управљање моторима и анализа резултата паралелно са комуникацијом преко мрежног порта, и довољно меморије за чување сакупљених података пре смештање на меморијску картицу или екстерни хард диск.

- Рад у реалном времену: Потребно је изабрати оперативни који омогућава контролу роботског система у реалном времену што укључује управљање путем комплексних алгоритама и читавање и тренутна анализа позиције сваког од мотора.

Да би се интегрисао цео систем у комплетно радно окружење и да се прилагоди за извршавање горе наведених задатака, поред саме развојне плоче неопходно је било осмислити и израдити додатну плочу која омогућава управљање са седам мотора једносмерне струје и читање података са енкодера.



Слика 4.1. Дијаграм повезивања компоненти

У финални избор ушле су следеће развојне плоче:

- Beaglebone Black
- Raspberry Pi model 3B
- Nano Pi
- Arduino Due

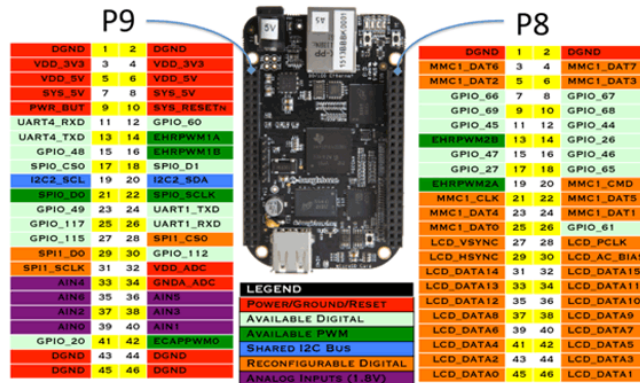
Након првих тестирања, одустало се прво од Arduino DUE развојне плоче. Разлог за то је немогућност обављања вишеструких комплексних операција у исто време које су неопходне како сада, тако и у будућности са даљим развојем роботског система и проширење постојеће функционалности као што је додавање додатних мотора и додатних сензора. Још један од разлога је и тај што Arduino DUE не поседује оперативни систем тзв. Вишег нивоа, што олакшава додавање нових сензора као што је нпр. камера. За финалну имплементацију изабрана је развојна плоча Beaglebone Black [29] јер, за разлику од Raspberry Pi 3B и Nano Pi, поседује два 46-пинска хедера што омогућава укупно 92 могуће конекције, а поседује и 4 тајмера као и 25 PRU I/O најнижег нивоа. Иако Beaglebone Black поседује спорији процесор и мање меморије, предност у односу на преостале две плоче му је дата јер поседује више него довољно процесорске снаге за покретање 7 мотора једносмерне струје са имплементацијом комплексних алгоритама и више додатних сензора. Даље даје се техничка спецификација Beaglebone Black развојне плоче:

- Процесор AM335x 1GHz ARM® Cortex-A8
- 512MB DDR3 RAM

- 4GB 8-bit eMMC flash меморије на плочи
- 3D графички акцелератор
- NEON акцелератор покретног зареза
- 2x PRU 32-bit микроконтролера

Као што је већ наведено, Beaglebone Black поседује 2 46-пинска хедера што омогућава 92 везе, и то: 65 GPIO; 7 аналогних улаза; 8 PWM излаза; 2 SPI буса; 2 I2C буса; 4 тајмера; 4 UART-а; 25 PRU I/O ниске латенције.

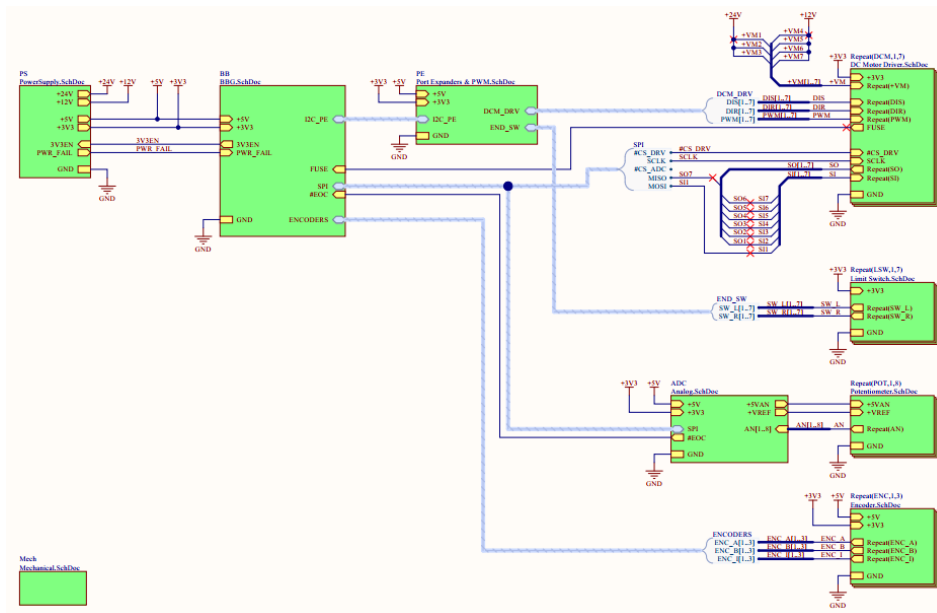
## Cape Expansion Headers



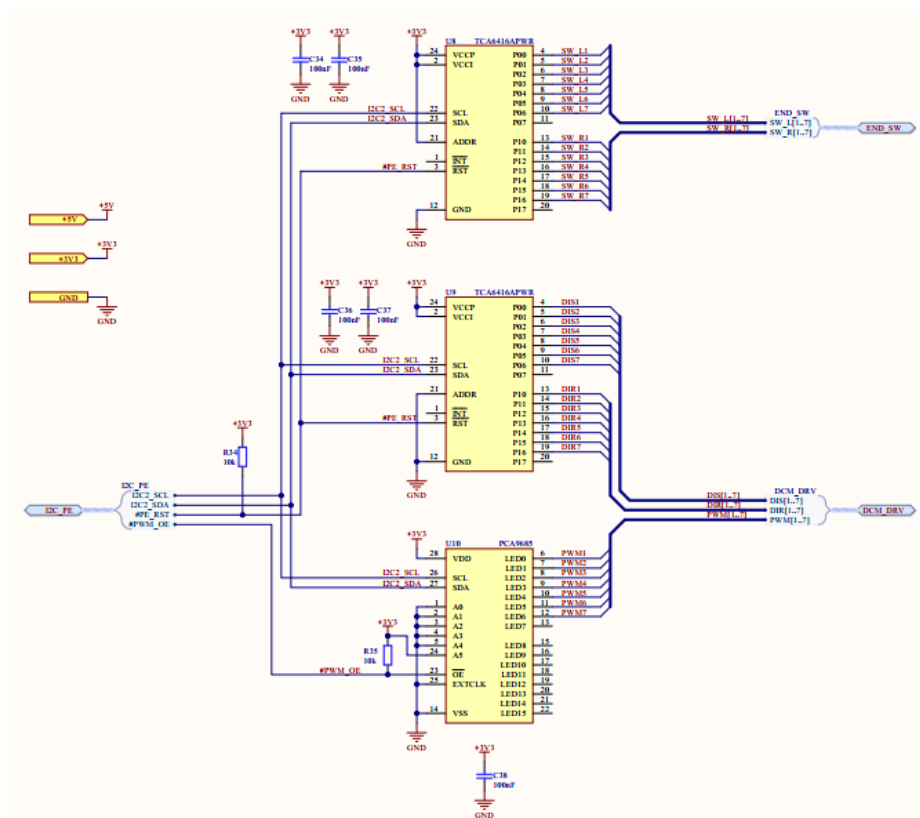
Слика 4.2. Beaglebone Black хедер шема

### 4.3. Плоча за управљање моторима и читање вредности енкодера

Као што је у одељку 4.1 наведено поред избора развојне плоче неопходно је и пројектовати додатну плочу која омогућава управљање са 7 мотора једносмерне струје и читање вредности са енкодера. NeuroArm роботска рука поседује 3 енкодера и 7 потенциометара за добијање тачне позиције мотора, чије је читање вредности и подржано овом новом плочом. Плоча је развијена заједно са инжењерима са Института Михаило Пупин, одељења за Аутоматiku. Шематски приказ делова управљачког дела контролне плоче приказан је на Слици 4.3., на Слици 4.4. приказан је Port Expander и PWM, а шематски приказ драјвера мотора једносмерне струје и аналогног конвертера налазе се у Прилогу П1. Тиме су приказани сви делови од којих се састоји додатна управљачка плоча.



Слика 4.3. Упрости́ена шема управљачке контролне плоче



Слика 4.4. Port expander и PWM

#### 4.4. Оперативни системи за рад у реалном времену са развојним плочама

Такозвани RTOS (Real Time Operating System) системи су оперативни системи намењени плочама где је битан рад и одзив у реалном времену. Апликације које се извршавају на таквим системима имају захтеве за кратким временским одзивом. RTOS мора да одговори на неки екстерни догађај у веома кратком временском року (кашњење интерапта). Такође мора да изврши одређени задатак у одређеном временском року (крајњи рок задатка). Системе можемо поделити на два типа и то системе који се понашају тако да увек реагују и извршавају задатке у задатим временским оквирима, и на системе који већином реагују и извршавају задатке у временским оквирима а мањим делом долази до кашњења. RTOS дефинише следеће карактеристике:

*Минимално кашњење интерапта (прекида):* Кашњење интерапта је време од настанка до тренутка када процесор изврши хендлер интерапта. Да би се смањило ово кашњење, систем мора да подржава угњеждавање интерапта да би осигурало да интерапти ниског нивоа приоритета не одлажу извршавање интерапта високог нивоа приоритета.

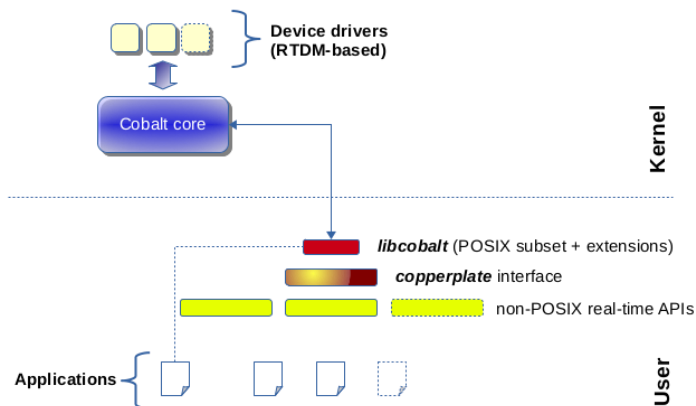
*Управљање извршавањем задатака:* Алгоритам који распоређује задатке и мења стање задатка који је потребно да процесор изврши су контролисане од стране RTOS-а, односно RTOS мора да омогући „приемптив“ динамичко распоређивање задатака.

Као што је већ наведено, RTOS мора да омогући брзи одзив и извршење задатака у строгим временским оквирима. Управљање овим процесима мора да задацима који се морају извршити у најкраћем временском року да приоритет. Постоје два главна алгоритма за овакав тип управљања задацима и то:

- RMS (Rate Monotonic Scheduling) – базиран је на логици да се приоритет даје задацима који имају мањи временски период за извршавање.
- EDF (Earliest Deadline First) – базиран је на логици да се приоритет даје дедлајн вредностима, а не по времену потребном да се задатак изврши. То значи да задатак који има најскорији рок за извршавање има приоритет у односу на све друге задатке.

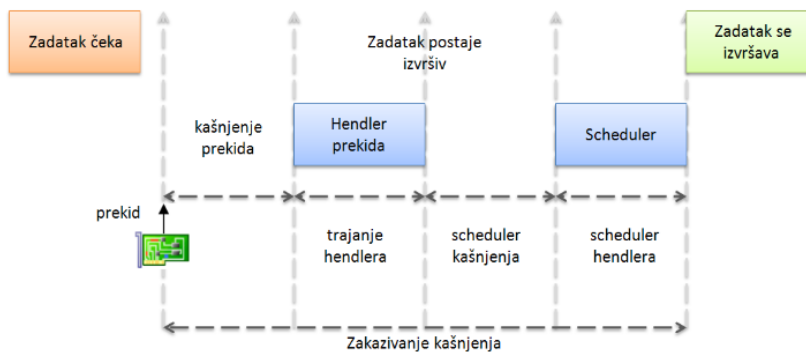
EDF алгоритам је флексибилнији од RMS алгоритма и омогућава боље искоришћење. Када је преоптерећен систем RMS алгоритам је стабилан јер у случају ако задатак није извршен у року, његов приоритет постаје низак што омогућава осталим задацима високог нивоа приоритета да се извршавају у задатим временским оквирима. RMS алгоритам захтева чешће распоређивање задатака.

За Beaglebone Black, односно NeuroArm роботску руку, смо се одлучили за верзију оперативног система која задовољава испуњавање задатака постављених у одељку 4.1 а то је Debian дистрибуција Linux оперативног система која је „печована“ Xenomai 3 керналом са циљем смањења латенције. Xenomai 3 је последња верзија нове архитектуре Xenomai real time фрејмворка. У нашем случају, додато је ново ко-језгро (Cobalt), а кернал Debian Linux-а је „печован“ Xenomai 3 PREEMPT-RT (Mercury језгром). Cobalt језгро поседује RTDM спецификацију за рад драјвера у реалном времену. На слици 3.5. се може види имплементација Cobalt језгра.



Слика 4.5. Xenomai 3

Предност Linux оперативних система у ембедид системима је та што је омогућено коришћење постојећих софтверских решења на различитим хардверским платформама. Скраћује се време развоја контроле нових уређаја због преузимања већ постојећих управљачких алгоритама путем библиотека за контролу различитих USB или мрежних уређаја, као и библиотека за рад са процесорима компатибилних архитектура као што су Intel, AMD или неки од ARM компатибилних процесора. Linux спада у такозвана Open source решења, што значи да је потпуно бесплатан, да је дозвољена модификација и да је повећана безбедност у односу на затворена решења.



Слика 4.6. Приказ рада Xenomai кернала

При развијању апликација у реалном времену, када дође до одређеног догађаја, који год он то био, обавештава се процесор путем тзв. прекида. Тада Хендлер препознаје и покреће се одговор покретањем дела програма који реагује на задати догађај. У Xenoma1 језгру све хендлере прекида извршава директно CPU, јер сви хендлери прекида имају највиши ниво приоритета. То значи да се у том тренутку, док се не обради прекид, стопирају сви остали виши задаци који се тренутно обрађују од стране оперативног система.

#### 4.5. Софтвер за програмирање NeuroArm роботске руке – EDICOPT

Што се тиче софтверског решења за програмирање комплексних алгоритама управљања NeuroArm роботске руке изабран је програмски пакет EDICOPT института Михаило Пупин јер подржава програмирање путем FBD и ST кода, што је један од основних захтева који смо усвојили. Поред програмирања алгоритама рада мотора, могуће је путем online мода програма мењати вредности улазних параметара алгорита сваког од мотора појединачно и тиме управљати радом једног или више мотора истовремено.

EDICOPT је модеран софтвер који је развио Институт Михаило Пупин и омогућава програмирање и конфигурисање индустријских контролера. Апликација је дизајнирана да олакша даљи развој и малих и великих дистрибутивних система. EDICOPT је флексибилни алат који омогућава употребу различитих хардверских платформи и оперативних система. Аутоматизовани системи могу се даље развијати тако да одговарају потребама корисника.

Најбитније карактеристике EDICOPT софтверског алата су:

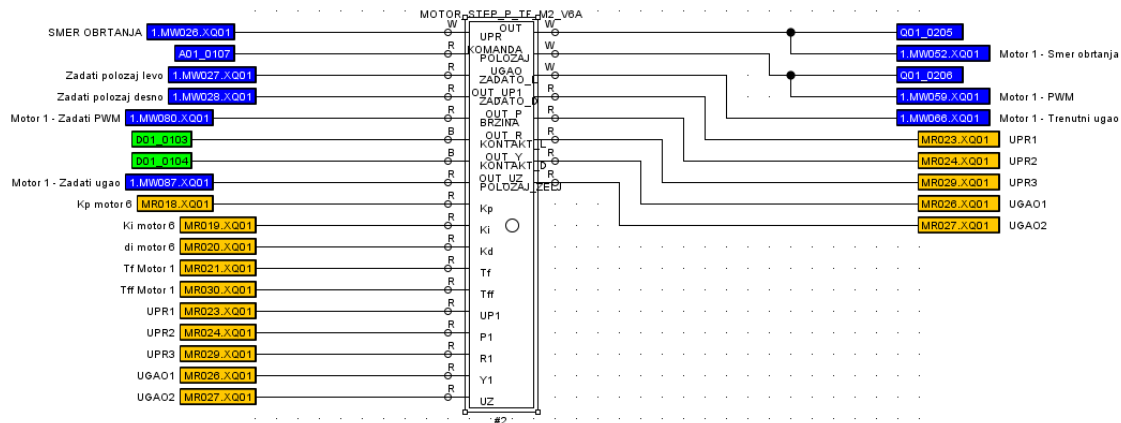
- Подршка за Windows и Linux оперативне системе
- Једноставна инсталација и покретање апликације
- Интуитивно графичко окружење
- Подршка за MySQL базу података
- Подржава рад у локалу без коришћења базе података
- Омогућава конфигурисање, програмирање и праћење рада радних станица
- У потпуности подржава IEC 60870-5-101, IEC 60870-5-104, и MODBUS RTU/TCP протоколе
- У складу је са IEC 61499 и IEC 61131 стандардима

EDICOPT се састоји од две главне компоненте: FBD едитор и Програмски терминал

FBD едитор је део EDICOPT апликације која се користи за цртање функционалних блок дијаграма. Подржава IEC стандарде (IEC 61131). Овај софтвер подржава алате који преводe лидер дијаграме у ST код као и код који је подржан од стране станице (RTU). EDICOPT омогућава двосмерну комуникацију, односно пренос

података између више извора што заузврат омогућава лакшу размену података у реалном времену, без обзира на број контролера без обзира да ли је он један у целом пројекту или неколико дистрибуираних путем локалне мреже.

Алати FBD едитора су: управљање логичким уређивањем; отклањање грешака; генерисање ST кода; документовање верзије и аутора; штампање дијаграма; уређивање базе података; преузимање програма на Atlas Max-RTL® контролере; подешавања путање комуникације; мрежни надзор и симулација.

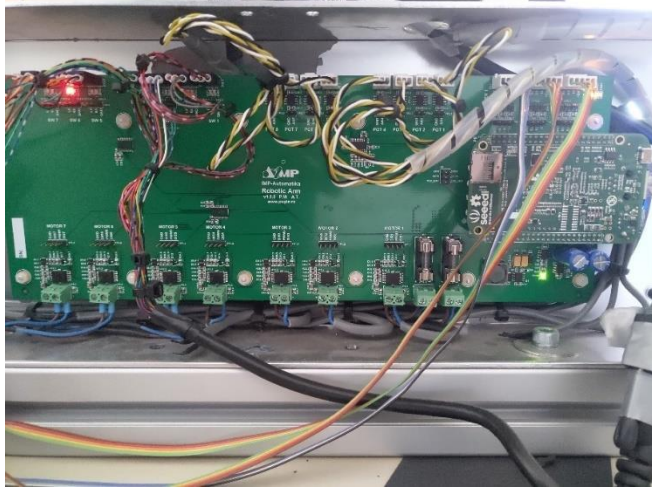


Слика 4.7. FBD едитор (функционални блок дијаграм едитор)

EDICOPT софтверски алат је први пут сада коришћен у оквиру ембедид система који је намењен за покретање једног роботског система. Због велике компатибилности самог програмског алата, и претходног коришћења Beaglebone Black развојне плоче на другим пројектима неvezаним за овај, било је у нашем случају лако све интегрисати.

#### 4.6. Имплементација новог контролера и програмирање

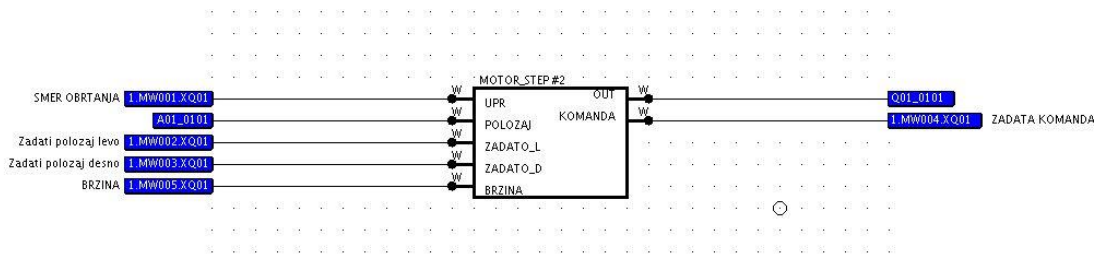
Након повезивања свих претходно наведених компоненти у једну функционалну целину, инсталације оперативног система са пропратним драјверима, на РС рачунар је инсталиран EDICOPT софтверски алат, започиње се тестирање тако што се тестира најједноставнији алгоритам управљања са само једним мотором једносмерне струје. Разлог за то је јер ће се исти начин повезивања и управљања применити и на све остале моторе. Први корак за повезивање са једним једносмерним мотором је провера доступних веза и жица из постојећег система управљања а жице су тестиране и уклоњене из старог система управљања и повезане са новим системом управљања (слика 4.8.).



Слика 4.8. Повезивање развојног система са моторима NeuroArm роботске руке

По окончању подешавања прешло се на тестирање контроле управљања једносмерним мотором. Подаци о положају прикупљају се од отпорних сензора од  $1\text{k}\Omega$  помоћу модула RI8 (отпорни модул) и од оптичких квадратурних енкодера помоћу једног модула DI16 (дигитални улаз).

PLC (програмабилни логички контролер) логика је креирана коришћењем EDICOPT апликације и FBD едитора и то најједноставнијом укључи/искључи контролном логиком. FBD дијаграм је приказан на слици 4.9.



Слика 4.9. укључи/искључи контролна логика

Променљива 1.MW01.XQ01 је контролни сигнал и она дефинише смер ротације, заустављање, или паузирање на основу њене вредности: 0 је искључено; 1 је иди лево; 2 је иди десно; 3 је пауза.

- A01-0101 је променљива која добија вредност у зависности од позиције енкодера
- Променљиве 1.MW002.XQ01 и 1.MW003. дефинишу границе крајње леве и крајње десне позиције.
- Променљива 1.MW005.XQ01 дефинише брзину.
- Излаз из блок дијаграма Q01\_0101 је вредност предефинисане брзине и контролног сигнала променљиве 1.MW004.XQ01.



Блок функција зависи од задатог смера ротације и позиције и онда зауставља мотор.

Иницијално тестирање је прошло у складу са очекивањима и без икаквих проблема. Мотором је управљано у оба смера односно кретањем робота на обе стране. Сваки пут када је положај мотора прешао дозвољену границу, долазило је до заустављања мотора у тој позицији. Тиме смо потврдили да ради исправно и читавање са енкодера и отворили могућност писања комплекснијих алгоритама управљања за даље тестирање.

#### 4.7. Тестирање MAXON 142750 мотора

NeuroArm роботски систем поседује два мотора Maxon 203131, два мотора Maxon 142750, три мотора Maxon gear 110339, три енкодера HEDS 5540 A06, 0740A и седам потенциометра типа 6639s-1-102, res 1kohm. Пре финалног повезивања и пуштања у рад система као једне функционалне целине због разлике између спецификација мотора које је произвођач навео и реалних карактеристика мотора, а због тачности симулације упоређивања са лабораторијским тестовима роботске руке, рађено је тестирање једног од мотора који покреће NeuroArm роботску руку, мотор са ознаком MAXON 142750, у лабораторији.



Слика 4.10. Maxon 142750 мотор

Фабричке карактеристике мотора дате су у Прилогу у Табели П5.1. За потребе испитивања коришћена је следећа опрема: два извора DC напона различитих напонских и струјних опсега, осцилоскоп, унимер, RLC метар.

Један од извора може да да напон од 29V са максималном струјом од 5A, док други извор даје напон од 18V али даје много већу струју. Међутим, ниједан од ова два извора не може да покрије цео опсег напона и струја у којима мотор може да ради. Зато је неопходно уочити правилности по којима се мења карактеристика брзина-струја (момент) променом напона напајања и онда проценити карактеристику мотора за номинални напон напајања.

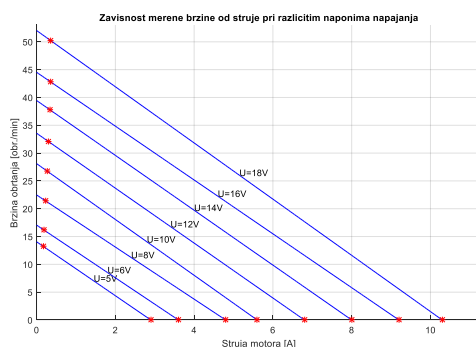
Помоћу DC извора који може да да напон до 18V, снимљене су радне тачке дате у табели 2. Напон напајања је повећаван од 5V до 18V, док је мотор радио без

оптерећења. За сваку од радних тачака снимљен је одзив напона и струје мотора на осцилоскопу. Полазна струја мотора у овом режиму одговара трајној струји која би се јавила при укоченом ротору (струји кратког споја), у табели је означена са  $I_{ks}$ . Са  $I_{ph}$  означена је вредност устаљене струје празног хода која се јавља услед савладавања трења у лежајевима и свих других отпора који се супротстављају обртању ротора. Брзина празног хода ( $n_{ph}$ ) је добијена на основу времена за које осовина на излазу из редуктора направи 10 обртаја ( $t_{10obr.}$ ). Као што је већ речено, не ради се о брзини идеалног празног хода будући да постоји отпор обртању, а самим тим и струја. Брзина идеалног празног хода накнадно је израчуната за сваку радну тачку.

Табела 4.1. Снимљене радне тачке

U [V]	$I_{ks}$ [A]	$I_{ph}$ [A]	$t_{10obr.}$ [s]	$n_{ph}$ [obr./min]
5	2.9	0.17	45.35	13.2304
6	3.6	0.185	37.09	16.1769
8	4.8	0.223	28	21.4286
10	5.6	0.27	22.43	26.7499
12	6.8	0.3	18.7	32.0856
14	8	0.338	15.88	37.7834
16	9.2	0.35	14.01	42.8266
18	10.3	0.36	11.95	50.2092

На Слици 4.10. приказана је фамилија карактеристика брзина-струја за различите напоне напајања мотора добијена на основу мерења. Звездице на апсциси означавају режим кратког споја, односно означене су измерене струје кратког споја, док звездице до ординате представљају радне тачке реалног празног хода, такође добијене мерењем. Линеарном интерполацијом и екстраполацијом, на основу измерених вредности струја и брзина, добијене су приказане карактеристике.



Слика 4.11. Зависност брзине са стране редуктора од струје

Са Слике 4.11. се може уочити да карактеристике имају готово исти нагиб, што је добар показатељ. Мерењем су се потврдила очекивања да се повећањем напона напајања карактеристика “транслира” на више. Укључењем другог извора напајања,

измерена је струја празног хода  $I_{ph} = 0.56A$  и брзина празног хода  $n_{ph} = 78.95 \text{obr./min}$  при напону напајања од  $29.3V$ . Прорачуном средње вредности коефицијента правца за карактеристике приказане на претходном графику, кроз снимљену тачку празног хода ( $0.56A$ ,  $78.95 \text{obr./min}$ ) провучена је права као што је приказано на слици испод. На истој слици је приказана карактеристика која одговара номиналном напону напајања мотора  $U = 32V$ , добијена прорачуном средње вредности коефицијента напона  $k_e = U / n_{idealno}$  за првих 8 радних тачака. На основу  $k_{eSr}$  добијено је  $n_{idealno} = 32V / k_{eSr}$ .

Више информација о комплетном тестирању је дато у Прилогу П5.

## 5. Управљање NeuroArt роботском руком генерисано у програмском окружењу

### 5.1. MODBUS

#### 5.1.1. Комуникациони протокол – основе

MODBUS је комуникациони протокол [104] настао 1979. године од стране фирме Modicon који је намењен комуникацији између програмабилних логичких контролера (PLC-ова). Прихваћеност у индустрији може да захвали једноставности и томе што је протокол отвореног типа (бесплатан). До 2004. развој протокола је контролисао Modicon, када је његов даљи развој преузела заједница корисника и снабдевача опреме за аутоматско управљање под именом Modbus-ИДА. Modbus се користи у опреми почевши од интеграције у најмање сензоре па до комплексних контролера и СКАДА система.

MODBUS је мастер/слејв полу дуплекс протокол [104], а подаци се преносе или путем серијског (RS232, RS485) порта или путем мрежног (Ethernet) порта. Постоје три врсте протокола: RTU, ASCII и TCP/IP. Modbus комуникациони протокол омогућава уређају да приступи другим уређајима и пише и чита у податке из предефинисаних меморијских локација. Пример комуникације је када уређај који нема (или има неку мању) могућност обраде података приступа серверу. Клијент PLC приступа серверу RTU и он је тај који чита податке који су записани у меморији сервера. Клијент може да комуницира са више сервера, исто као што и један сервер може да комуницира са више клијената. Комуникација се одвија тако што се креира ред чекања на захтев клијента, а могуће је и паралелно обрађивање захтева уколико сервер то подржава.

Постоје четири меморијске локације којима се може приступати, и то:

Табела 5.1. Меморијске локације резервисане за сегменте

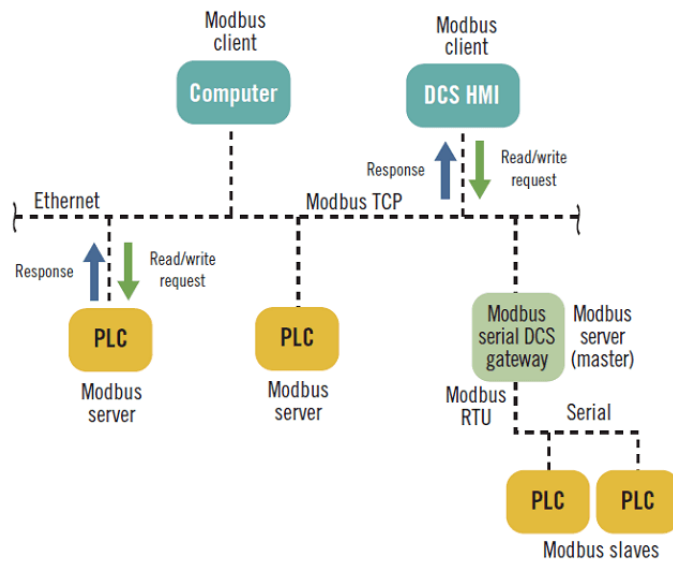
Меморијска локација	Назив	Величина (бит)	Приступ
0-10.000	Коилс	1	Читање/писање
10.000-20.000	Дискретни улази	1	Читање
30.001-40.000	Улазни регистри	16	Читање
40.001-50.000	Холдинг регистри	16	Читање/писање

Најчешће интерна организација је да RTU који је Modbus сервер омогућава читање својих физичких дигиталних улаза кроз меморијске локације за дискретне

улазе, док се дигитални излази могу читати и писати кроз меморијске локације „коилс“. Статус аналогних улаза се чита кроз меморијске локације улазних регистара, а вредности аналогних излаза се читају кроз меморијске локације холдинг регистара. Али се исто тако омогућава самом произвођачу сервера да у одређене меморијске локације смешта оне информације које су му потребне, јер сам Modbus протокол препоручује али не и ограничава.

Један од примера би био HMI интерфејс где је визуелно приказано тренутно стање индустријских процеса и омогућава оператеру да мења одређене параметре путем терминала. HMI терминал који је Modbus клијент чита податке са PLC-а који је Modbus сервер о стању процеса. PLC може користити меморијске локације холдинг регистара да чува податке о тренутном стању процеса који контролише (нпр. Количине воде у резервоару) а у меморијској локацији коилс информације о тастерима које оператер притиска.

Изабрали смо Ethernet TCP верзију овог протокола јер нам она омогућава и даљинско управљање роботом путем интернета. Modbus TCP мрежа се састоји од више уређаја који су повезани путем мреже пратећи структуру клијент-сервер модела. Прво је потребно успоставити везу између клијента и сервера. Једном када је конекција успостављена клијент може послати серверу захтев. Успостављање везе се одвија путем TCP/IP протокола и није у директној вези са Modbus протоколом. Modbus сервер на резервисаној адреси 502 слуша захтеве клијената.



Слика 5.1. Дијаграм Modbus логике рада

MODBUS порука има своју структуру која не зависи од варијанте преноса, већ се убацује у стандардну TCP/IP поруку. Свака порука се састоји од четири елемента који се шаљу по идентичном распореду: адреса примаоца; код функције коју је потребно извршити; параметара (података) функције; провера грешке.

Modbus протокол дефинише велики број функција. Најчешће коришћене функције су оне за приступ меморијским локацијама.

Све функције у Табели 5.2. осим функција 0x14, 0x15, 0x16, 0x18 су захтеви да неки податак буде прочитан или уписан у одређену меморијску локацију. Кодови 0x05, 0x06 се користе за уписивање једног податка. Остале функције омогућавају клијенту да чита или пише више података са меморијских локација које су повезане.

Табела 5.2. Кодови функција

Мем. Локација	Име функције	Код	Адреса	Код грешке
Дискретни улаз	Читање дис. улаза	0x02	1-2000	01, 02, 03, 04
Коил	Читање коилс	0x01	1-2000	01, 02, 03, 04
Коил	Писање једног коила	0x05	1	01, 02, 03, 04
Коил	Писање више коила	0x0F	1-1976	01, 02, 03, 04
Улазни регистри	Читање улазних рег.	0x04	1-125	01, 02, 03, 04
Холдинг регистри	Читање холдинг рег.	0x03	1-125	01, 02, 03, 04
Холдинг регистри	Писање једног рег.	0x06	1	01, 02, 03, 04
Холдинг регистри	Писање више рег.	0x10	1-123	01, 02, 03, 04
Холдинг регистри	Читање/писање више регистара	0x17	1-121 читање 1-125 писање	01, 02, 03, 04
Холдинг регистри	Маскирај пиши рег.	0x16	1	01, 02, 03, 04
Холдинг регистри	Читање ФИФО низа	0x18	1-32	01, 02, 03, 04
Фајл	Прочитај фајл	0x14		01, 02, 03, 04, 08
Фајл	Упиши фајл	0x15		01, 02, 03, 04, 08

### 5.1.2. Фрејмови – порука

Модбус TCP порука (захтев и одговор) састоји се од МВАР заглавља и апликационог нивоа APDU.

Идентификација захтева: Како клијент може послати више узастопних захтева преко исте конекције, није гарантовано да ће одговори стићи истим редом како су захтеви послати. Зато је неопходно идентификовати сваки захтев да би се знао одговор на тачно одређени захтев. Клијент креира идентификатор захтева, а сервер приликом одговора у заглавље поруке такође уноси идентични идентификатор.

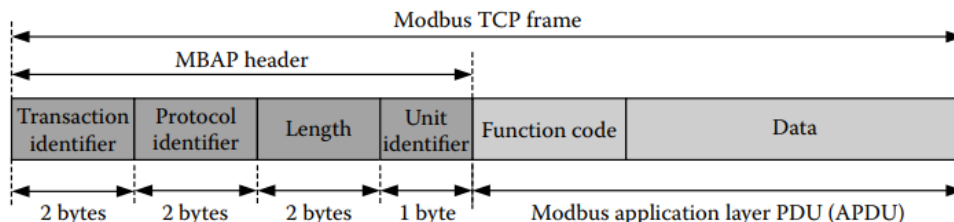
Идентификација протокола: Користи се ради идентификације протокола. По садашњем стандарду, та вредност је увек 0.

Дужина: Она је у бајтовима и односи се на идентификатор уређаја и на APDU. Пренос података у TCP/IP је низ бајтова и у баферу се може наћи више спојених

низова. Да би се идентификовали почетак и крај сваког низа (поруке) мора се знати тачна дужина.

Идентификација уређаја: Ово поље се користи да би се идентификовао уређај (слејв).

TCP поруке немају поље за детекцију грешака, за разлику од серијске комуникације, с обзиром на то да сам TCP/IP протокол има механизам за детекцију грешака у преносу.



Слика 5.3. Modbus фрејм

## 5.2. EasyMODBUS TCP библиотека

За комуникацију између рачунара и NeuroArm роботске руке, тачније Beaglebone Black развојне плоче користи се решење отвореног кода EasyMODBUS TCP библиотека. Оно се може наћи на сајту чија је адреса <http://easymodbustcp.net/en/> или на следећој адреси <https://sourceforge.net/projects/easymodbustcp/>

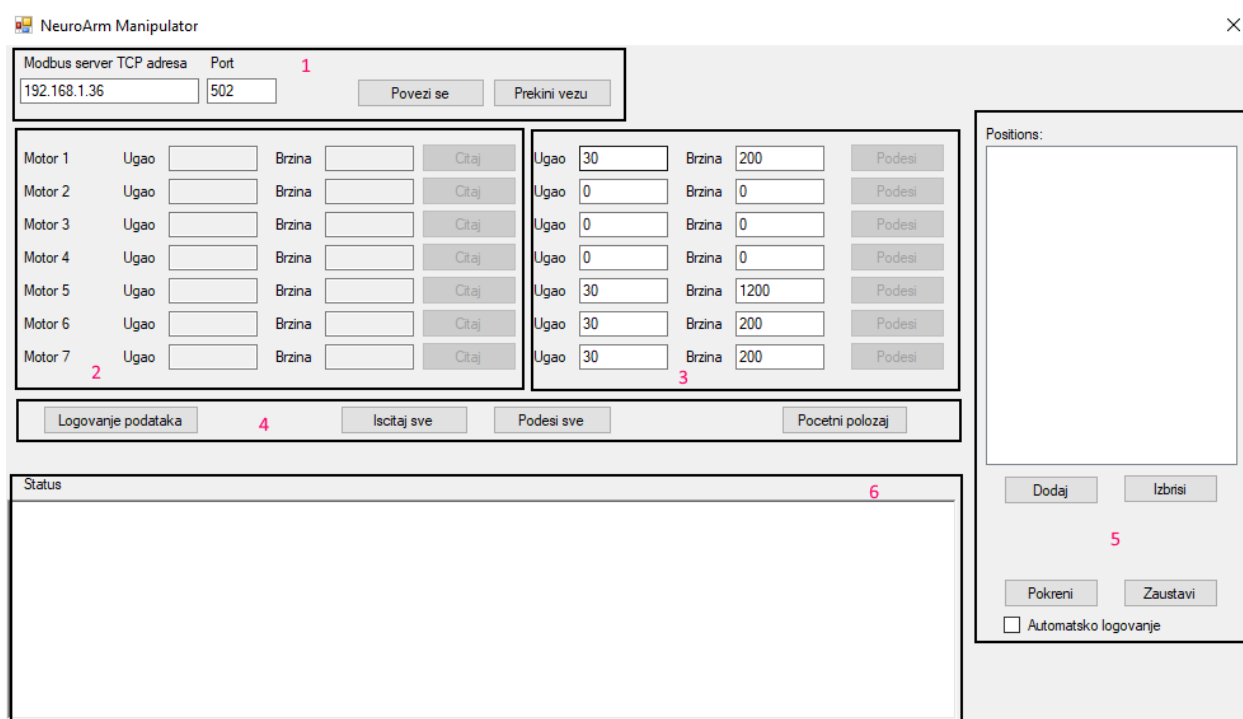
Ова библиотека у потпуности подржава Ethernet TCP/IP и следеће кодове функције:

Read Coils (FC1); Read Discrete Inputs (FC2); Read Holding Registers (FC3); Read Input Registers (FC4); Write Single Coil (FC5); Write Single Register (FC6); Write Multiple Coils (FC15); Write Multiple Registers (FC16).

## 5.3. NeuroArm manipulator – софтверско решење за управљање NeuroArm роботском руком

Да би могли да задајемо команде роботској руци, управљамо њом, програмирамо сложене алгоритме кретања више зглобова у исто време, радимо логовање података мотора током кретања (брзина, тренутни угао) било је неопходно да напишемо програм који ће нам то омогућити. Тако је и настао програм NeuroArm manipulator за манипулацију роботском руком који смо ми написали у програмском језику C# [37], [105], [106] и програмерском окружењу Visual Studio 2019 Community Edition (IDE). Програм омогућава покретање сваког мотора (зглоба)

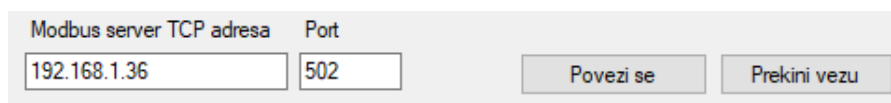
роботске руке, тиме што је могуће подесити максималну жељену брзину мотора као и угао у коме се мотор зауставља. Програм омогућава снимање (логовање) података током процеса рада на сваких 16ms, обраду тих резултата и снимање у фајл који је могуће учитати у нпр. Microsoft Excel где се сачуване вредности могу лакше анализирати због прегледности, као и графички представити. 16ms је време семпловања које нам је у овом тренутку ограничавајући фактор и у будућности је у плану да се убрза ово време тиме што ће се оптимизацијом кода смањити додатно брзина одзива. На следећој слици 4.4. може се видети главни екран апликације који је подељен у више логичких целина.



Слика 5.4. Главни екран апликације NeuroArm manipulator

Елементи главног екрана подељени по логичким целинама:

Уношење TCP/IP адресе и порта NeuroArm роботске руке, као и два дугмета са којим остварујемо или прекидамо везу.



Слика 5.5. Подешавање адресе и порта за повезивање са NeuroArm роботском руком

Део 2 приказује исчитане брзине мотора и углове појединачних делова руке (7 мотора). Могуће је и појединачно читање података за мотор или за све моторе.



Motor 1	Ugao	<input type="text"/>	Brzina	<input type="text"/>	Citaj
Motor 2	Ugao	<input type="text"/>	Brzina	<input type="text"/>	Citaj
Motor 3	Ugao	<input type="text"/>	Brzina	<input type="text"/>	Citaj
Motor 4	Ugao	<input type="text"/>	Brzina	<input type="text"/>	Citaj
Motor 5	Ugao	<input type="text"/>	Brzina	<input type="text"/>	Citaj
Motor 6	Ugao	<input type="text"/>	Brzina	<input type="text"/>	Citaj
Motor 7	Ugao	<input type="text"/>	Brzina	<input type="text"/>	Citaj

Слика 5.6. Приказ исчитаних вредности брзина и углова мотора

Део 3 омогућава уношење жељених брзина мотора и жељених углова делова руке (7 мотора). Могуће је и појединачно подешавање података за мотор или за све моторе.

Ugao	<input type="text" value="30"/>	Brzina	<input type="text" value="200"/>	Podesi
Ugao	<input type="text" value="0"/>	Brzina	<input type="text" value="0"/>	Podesi
Ugao	<input type="text" value="0"/>	Brzina	<input type="text" value="0"/>	Podesi
Ugao	<input type="text" value="0"/>	Brzina	<input type="text" value="0"/>	Podesi
Ugao	<input type="text" value="30"/>	Brzina	<input type="text" value="1200"/>	Podesi
Ugao	<input type="text" value="30"/>	Brzina	<input type="text" value="200"/>	Podesi
Ugao	<input type="text" value="30"/>	Brzina	<input type="text" value="200"/>	Podesi

Слика 5.7. Подешавање жељених вредности улова и брзина мотора

Притиском на тастер „Исчитај све“ шаље се захтев NeuroArm-у за тренутним вредностима брзина мотора и угловима, и те информације су приказане у делу 2.

Притиском на тастер „Подеси све“ шаље се захтев NeuroArm-у да подеси нове вредности брзина мотора и углове које су претходно подешене у делу 3.

Притиском на тастер „Почетни положај“ шаље се захтев NeuroArm-у да постави све вредности за брзине и моторе које су унапред дефинисане као почетне вредности.

Притиском на тастер „Логовање података“ започињемо комуникацију са NeuroArm-ом где добијамо информације о тренутном положају и брзини мотора и те вредности се складиште привремено у RAM. По притиску на тастер „Заустави логовање“ прекида се преузимање вредности и до тада преузете вредности се снимају у „.csv“ (comma separated values) фајл који се може учитати из Microsoft Excel-а или неког другог програма који омогућава табеларни приказ вредности.

Logovanje podataka	Iscitaj sve	Podesi sve	Pocetni položaj
--------------------	-------------	------------	-----------------

Слика 5.8. Логовање података

У делу 5 подешава се трајекторија кретања роботске руке. Могуће је по једном у једном циклусу да се креће произвољан број зглобова (од 1 до 7). Након што се подесе брзине мотора и жељени углови, кликне се на дугме „Додај“. Може се додати произвољан број циклуса. Уколико је дошло до грешке у подешавањима неког циклуса, тада је тај циклус могуће обрисати кликом на дугме „Избриши“. Након што су додате све позиције, пуштање у рад се обавља кликом на дугме „Покрени“. Процес се извршава све док се не заврше сви унети циклуси. Уколико је потребно прекинути процес, то се постиже кликом на дугме „Заустави“. Уколико је селекција „Аутоматско логовање“ изабрано, тада ће се снимити сви подаци о брзини мотора и позицији (углу) у фајл. Име фајла се аутоматски креира и додаје му се датум и време почетка рада процеса.

У делу 6 се налази статусно поље. У њему се исписују сви процеси који се тренутно извршавају, као и одговор NeuroArm-а на одређени захтев који је послат од стране програма.



Слика 5.9. Статусни прозор

Што се тиче програмског дела, прво је потребно дефинисати променљиве које се користе.

Адресе за читање података

```
public int StartingAddressRead = 10;  
public int EndingAddressRead = 30;  
public int BrojRegistaraRead = 21;
```

Адресе за уписивање података

```
public int StartingAddressWrite = 100;  
public int EndingAddressWrite = 30;  
public int BrojRegistaraWrite = 14;
```

Променљиве које се користе су

```
public int StartingAddressWrite = 100;  
public int EndingAddressWrite = 30;  
public int BrojRegistaraWrite = 14;  
  
public int[] Uglovi_Slanje = new int[7];  
public int[] Brzine_Slanje = new int[7];
```

Тачне адресе података који се читају и уписују могу наћи у Прилогу П4.

Као што се може видети, све ове променљиве су низови, односно Array типа које садрже целобројне вредности (Integer type). Сваки низ садржи 7 вредности, јер имамо 7 мотора, односно 7 енкодера са којих читамо вредности.

Код за успостављање и прекидање комуникације је у Прилогу П5.

Преузимање података:

```
private void Iscitaj()
{
    if (modbusClient.Connected)
    {
        try
        {
            int[] serverResponse = modbusClient.ReadHoldingRegisters(StartingAddressRead, BrojRegistaraRead);
            AnswerFromServer.Clear();
            for (int i = 0; i < serverResponse.Length; i++)
            {
                AnswerFromServer.Add(serverResponse[i]);
            }
            Array.Copy(serverResponse, 0, SmerObrtanja_Iscitan, 0, 7);
            Array.Copy(serverResponse, 0, Brzine_Iscitane, 0, 7);
            Array.Copy(serverResponse, 14, Uglovi_iscitani, 0, 7);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Problem sa citanjem vrednosti sa servera", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("Morate se prvo povezati sa serverom");
    }
}
```

Слање података:

```
private void Posalji()
{
    if (modbusClient.Connected)
    {
        try
        {
            int[] registersToSend = new int[BrojRegistaraWrite];

            Array.Copy(Brzine_Slanje, 0, registersToSend, 0, 7);
            Array.Copy(Uglovi_Slanje, 0, registersToSend, 7, 7);

            modbusClient.WriteMultipleRegisters(StartingAddressWrite, registersToSend);
            rtb_History.Text += "Poslati holding registri" + Environment.NewLine;
        }
        catch (Exception exc)
        {
        }
    }
}
```

```

        MessageBox.Show(exc.Message, "Problem sa slanjem vrednosti serveru", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
else
{
    MessageBox.Show("Morate se prvo povezati sa serverom");
}
}
}

```

Периода прикупљања података је 16ms као што је већ напоменуто.

```

public void OnSystemTimerEvent(Object source, System.Timers.ElapsedEventArgs e)
{
    if (modbusClient.Connected)
    {
        //int[] serverResponse = modbusClient.ReadHoldingRegisters(int.Parse(txtStartingAddressInput.Text) - 1,
        int.Parse(txtNumberOfValuesInput.Text));

        try
        {
            int[] serverResponse = modbusClient.ReadHoldingRegisters(StartingAddressRead, BrojRegistaraRead);
            AnswerFromServer.Clear();
            for (int i = 0; i < serverResponse.Length; i++)
            {
                AnswerFromServer.Add(serverResponse[i]);
            }
            LoggedData ld = new LoggedData();
            ld.OrderNumber = LoggOrderNumber;
            //ld.time = DateTime.Now.ToString();
            ld.time = e.SignalTime.Millisecond.ToString();
            ld.data = serverResponse;
            LoggedDataList.Add(ld);
            LoggOrderNumber++;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Problem sa citanjem vrednosti sa servera", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("Morate se prvo povezati sa serverom");
    }
}
}
}

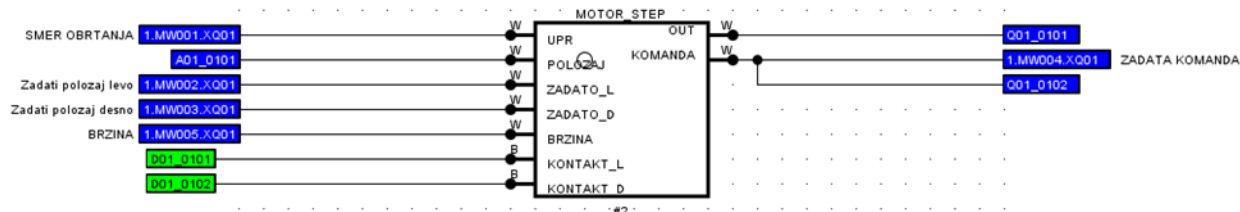
```

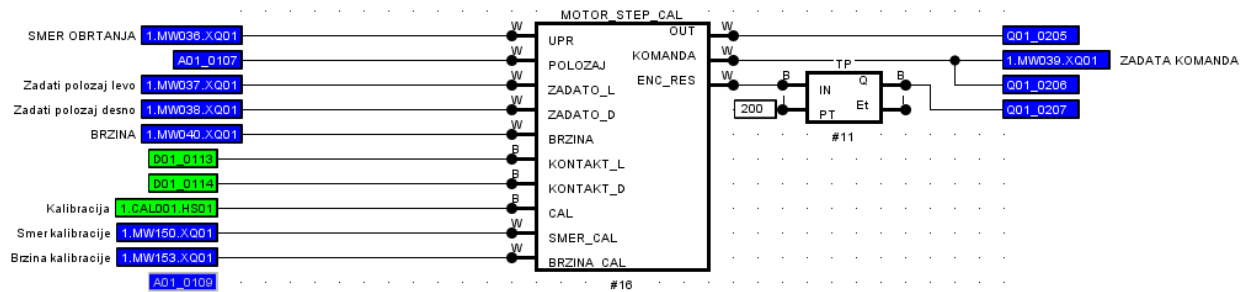
Приказ како изгледају подаци када се фајл учита у Microsoft Excel ради даље анализе

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
130	130	767	0	0	0	0	0	0	715	2	90	0	0	2	2	1	6	0	0	0	46	56	29
131	131	783	0	0	0	0	0	0	715	2	90	0	0	2	2	1	6	0	0	0	46	56	29
132	132	799	0	0	0	0	0	0	713	2	90	0	0	2	2	1	6	0	0	0	46	56	30
133	133	815	0	0	0	0	0	0	712	2	90	0	0	2	2	1	6	0	0	0	46	56	30
134	134	831	0	0	0	0	0	0	702	2	90	0	0	2	2	1	6	0	0	0	46	56	30
135	135	847	0	0	0	0	0	0	688	2	90	0	0	2	2	1	6	0	0	0	46	56	31
136	136	863	0	0	0	0	0	0	681	2	90	0	0	2	2	1	6	0	0	0	47	56	31
137	137	879	0	0	0	0	0	0	672	2	90	0	0	2	2	1	6	0	0	0	47	56	31
138	138	895	0	0	0	0	0	0	660	2	90	0	0	2	2	1	6	0	0	0	47	56	32
139	139	911	0	0	0	0	0	0	648	2	90	0	0	2	2	1	6	0	0	0	47	56	32
140	140	927	0	0	0	0	0	0	633	2	90	0	0	2	2	1	6	0	0	0	46	56	33
141	141	943	0	0	0	0	0	0	616	2	90	0	0	2	2	1	6	0	0	0	47	56	34
142	142	959	0	0	0	0	0	0	600	2	90	0	0	2	2	1	6	0	0	0	47	56	34
143	143	975	0	0	0	0	0	0	587	2	90	0	0	2	2	1	6	0	0	0	47	56	35
144	144	992	0	0	0	0	0	0	567	2	90	0	0	2	2	1	6	0	0	0	47	56	36
145	145	13	0	0	0	0	0	0	543	2	90	0	0	2	2	1	6	0	0	0	47	56	37
146	146	29	0	0	0	0	0	0	526	2	90	0	0	2	2	1	6	0	0	0	47	56	37
147	147	45	0	0	0	0	0	0	515	2	90	0	0	2	2	1	6	0	0	0	47	56	38
148	148	66	0	0	0	0	0	0	491	2	90	0	0	2	2	1	6	0	0	0	46	56	39
149	149	98	0	0	0	0	0	0	430	2	90	0	0	2	2	1	6	0	0	0	47	56	41
150	150	120	0	0	0	0	0	0	426	2	90	0	0	2	2	1	6	0	0	0	47	56	42
151	151	136	0	0	0	0	0	0	405	2	90	0	0	2	2	1	6	0	0	0	47	56	42
152	152	154	0	0	0	0	0	0	379	2	90	0	0	2	2	1	6	0	0	0	47	56	44
153	153	171	0	0	0	0	0	0	366	2	90	0	0	2	2	1	6	0	0	0	47	56	44
154	154	188	0	0	0	0	0	0	345	2	90	0	0	2	2	1	6	0	0	0	47	56	45
155	155	209	0	0	0	0	0	0	325	2	90	0	0	2	2	1	6	0	0	0	47	56	46
156	156	225	0	0	0	0	0	0	304	2	90	0	0	2	2	1	6	0	0	0	47	56	47
157	157	241	0	0	0	0	0	0	232	2	90	0	0	2	2	1	6	0	0	0	47	56	47
158	158	257	0	0	0	0	0	0	269	2	90	0	0	2	2	1	6	0	0	0	47	56	48
159	159	275	0	0	0	0	0	0	257	2	90	0	0	2	2	1	6	0	0	0	46	56	49
160	160	300	0	0	0	0	0	0	231	2	90	0	0	2	2	1	6	0	0	0	47	56	50

Слика 5.10. Приказ података унутар пакета Microsoft Excel.

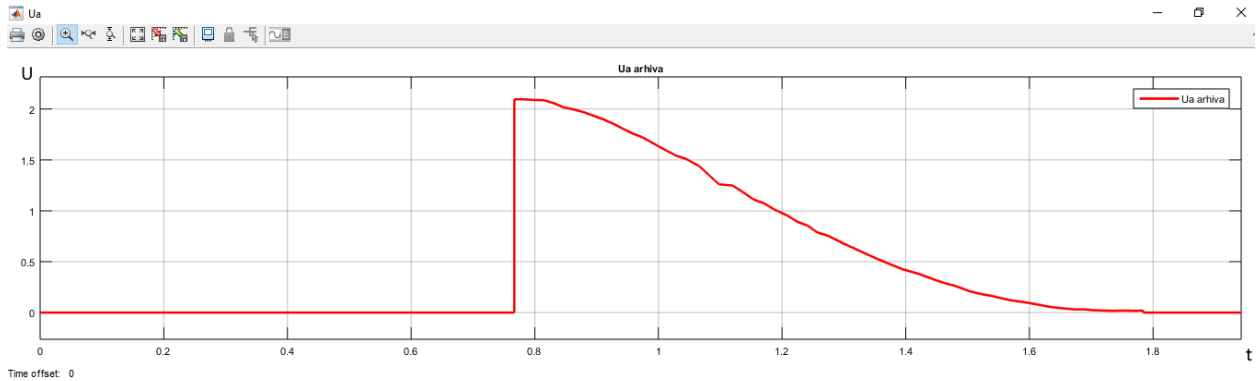
На слици 5.11. је Edicopt FBD за управљање моторима. Сваки мотор је засебна целина која има своје улазне и излазне величине. Приказан је код само 2 мотора из разлога што је идентичан код, али са другачијим улазним и излазним величинама које су дате у Прилогу П4. На слици први FBD код је за моторе 4, 5, 6 и 7 а други је за моторе 1, 2 и 3. Разлика је та што мотори 1, 2, 3 имају читавање позиције поред потенциометара и путем енкодера. За сваки FBD појединачног мотора дефинисан је ST код који анализира улазне величине (леви део), креира управљање које прослеђује на излаз (десни део). Овај код је спуштен у развојну плочу тако да се овим новим предефинисаним алгоритмима сваког мотора понаособ управља путем NeuroArm manipulator програма. Комплетан ST код којим се управља мотором 1 налази се у прилогу П3. У коду се може видети да је прво неопходно дефинисати улазне и излазне променљиве, које су у директној корелацији са улазним и излазним величинама које се налазе на FBD-у (слика 5.11.). У првој итерацији, због једноставности, сигурности, а при томе и без губљења општости, примењен је само пропорционални алгоритам. У коду се, такође, може видети да је интегрисан ПИД алгоритам, али унутар EDICOPT програма коришћено је само  $k_p$ , односно пропорционални део је различит од нуле, док су вредности за  $k_i$  и  $k_d$  нуле.



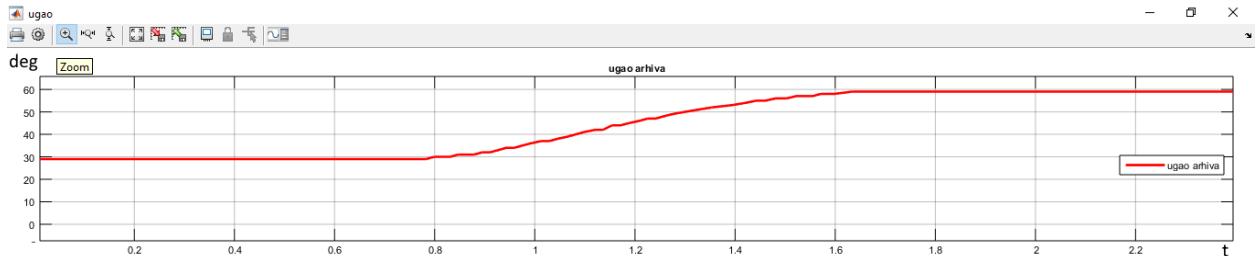


Слика 5.11. FBD контроле свих 7 мотора NeuroArm роботске руке

Након што је путем NeuroArm манипулатор програма задато управљање, где је задато  $k_p = 4$ , почетни угао је  $30^0$  и крајњи  $60^0$ . Снимљени подаци о вредностима углова и брзинама мотора и резултати експеримента могу се видети на сликама 5.12. и 5.13.



Слика 5.12. Промена напона у зависности од времена за  $k_p = 4$



Слика 5.13. Промена угла у зависности од времена од  $30^0$  до  $60^0$  за  $k_p = 4$

## 5.4. Даљинско управљање роботским системом

### 5.4.1. Предности и мане даљинског управљања

Данас даљинско управљање везујемо највећим делом за управљање неким уређајем путем интернета [107], [108], [109], [110]. Али почеци даљинског управљања датирају у 19. век, тачније у 1989. годину када је Никола Тесла први изумео бежично управљање. У Првом и Другом Светском рату коришћењем радио таласа су се контролисали уређаји, као што је 21. 03. 1917. Џофри Де Хавиланд управљао авионом. Данас се радио таласима на сличан начин управља ровером на Марсу.

Постоји више начина за даљинско управљање:

- *Инфра црвена технологија*: (има доста ограничења, као што су ограничена удаљеност од неколико метара до 10 метара, у зависности од квалитета ИР бластера, мора постојати оптичка видљивост између емитера и пријемника)
- *Радио таласи*: распрострањена је оваква контрола код модела РЦ модела авиона, аутомобила, али и код НАСА ровера који се налази на Марсу. Команда је број пулсева који је рисивер примио узастопно у кратком временском интервалу.
- *Препознавањем гласа*: Уређај којим се управља ради константно обраду звука и када препозна гласовну команду тада извршава операције са којим је она логички повезана.
- *Звуком*: Слично као са гласом, ради се стални мониторинг звука који се прима, али за разлику од гласа, у овом случају актуатор функције може бити одређена фреквенција звука одређене дужине трајања, пуцкетање прстима, плесак длановима,... Код овог и претходног начина управљања иако би се рекло да је потребно да особа која командује буде у близини, преко звучника који се налази у близини је могуће издавати команде, а да оператер буде произвољно далеко.
- *Покретом*: Путем камере се препознају покрети нпр. руке и на основу препознатог покрета активира се одређена команда.
- *Мрежна технологија* (путем кабла, ви-фи, блутут): Даљинско управљање путем кабла се може поделити на директно управљање или управљање путем мреже које се додатно дели на управљање путем локалне мреже (интранет) или управљање ван локалне мреже (интернет). Када је блутут даљинско управљање даљина на коју је могуће управљати зависи од блутут стандарда који је подржан као и од препрека које се налазе између. Ви-фи управљање [111], [112] се такође дели на интранет и интернет.

У наставку наше интересовање и пажња биће усмерена на управљање преко интернета, [113].

Развој интернета омогућио је и развој и интеграцију даљинског управљања у све сфере живота. Његова интеграција прво је почела у испоставама у индустријске сврхе [114], војне сврхе [115], [116], да би се проширила употреба на свакодневне активности као што је управљање различитим уређајима у кући.

Постоје три главна проблема код даљинског управљања преко интернета.

*Кашњење* и зависности од примене даљинског управљања је изузетно битно, као што је то случај код операција на даљину. Зато се гледа да у зависности од критичности реаговања подаци који се преносе буду оптималне величине. Најчешће су подаци који се преносе у облику XML или JSON структуре података. Уколико је количина података велика а кашњење критично тада се ради компресија података пре слања и декомпресија података након пријема података.

*Безбедност* је можда и највећи изазов данас зато што се свако може повезати на интернет и приступити свакој интернет адреси [117]. Колико год даљинска контрола има предности јер омогућава удаљеном оператеру да преузме контролу над нпр. нуклеарном електраном и интервенише у случају када је особље морало бити евакуисано, исто тако је исто то постројење изложено нападу хакера. Уколико „погоди“ шифру за приступ, тада хакер може преузети контролу над системом, што може довести до несагледивих последица. За свакодневну употребу примењен је следећи сет заштита од неовлашћеног упада:

- постављање заштитног зида (firewall),
- логовање са шифром одговарајуће тежине (комбинација слова и бројева, слова морају да имају комбинацију малих и великих, обавезно уношење неког од специјалних карактера – „!%&/(-.“,
- логовање свих приступа или ограничење приступа са само одређених IP адреса,
- енкрипција порука које се шаљу (најчешће AES256 [118] или AES512).

Са повећањем нивоа битности, нпр. код нуклеарне електране, знатно се повећава ниво сигурности коришћењем најнапреднијих безбедносних мера и програма за тестирање и проналажење тзв. рупа да би се могућност неовлашћеног упада свела на минимум, јер је, ипак, немогуће такву могућност у потпуности елиминисати. Једна од данас најнапреднијих мера је коришћење вештачке интелигенције за анализу понашања сваког од корисника који су приступили мрежи и уколико се детектује проблем кориснику се укида приступ и обавештава се супервизор са детаљним логом понашања корисника, а након неколико упада долази се до привременог искључења могућности приступа са удаљених локација на одређено време и отвара се само специјални сигурносни канал за приступ. Касперски, иноватор и оснивач фирме за антивирусни софтвер је истакао да за најбитнија војна постројења се креира засебна мрежа потпуно одвојена од интернета, али и такве мреже су исто изложене нападима, у знатно мањој мери, тако да су принципи заштите исти.



Приступ више корисника је такође један од проблема и треба обезбедити да само један корисник има контролу или над целим системом или над једним делом система који је потпуно независан. Потребно је да се корисник излогује након завршеног даљинског приступа, или уколико се корисник не излогује, након одређеног периода неактивности сам систем сматра да корисник више није повезан на систем и аутоматски излогује корисника и омогућава логовање новом кориснику.

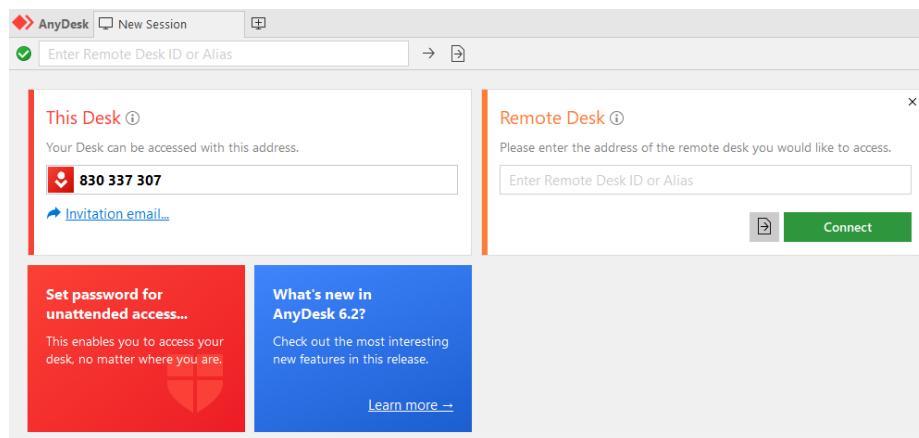
#### 5.4.2. Даљинско управљање NeuroArm роботском руком

NeuroArm роботска рука је саставни део Лабораторије за примењену механику Машинског факултета у Београду и повезана је преко мрежне конекције на локални рутер који је повезан на интернет унутар факултета. Сви уређаји који се налазе унутар мреже Машинског факултета се налазе иза NAT-а. Зато смо изабрали два бесплатна решења која омогућавају повезивање са NeuroArm роботском руком директно и то са компјутером са којим се приступа ван факултетске мреже.

Ова два решења се разликују по могућности програмирања ПИД алгоритама мотора робота, односно само управљањем и манипулацијом моторима роботске руке са препрограмираним PID алгоритмима.

*AnyDesk VPN (<http://www.anydesk.com>)*

Ово решење је свеобухватно јер нам омогућава директан приступ рачунару на коме се налази EDICOPT програмско окружење, које као што је у претходном делу већ објашњено, омогућава програмирање алгоритама управљања мотора.



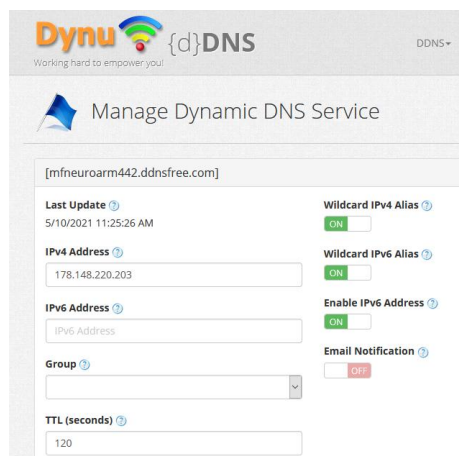
Слика 5.12. AnyDesk VPN

На рачунару у лабораторији потребно је кликом на дугме „Set password for unattended access...” подесити шифру за даљински приступ, односно сваки клијент рачунар који покушава да приступи овом тзв. сервер рачунару мора унети ту шифру за даљински приступ.

Обзиром на то да је након успостављања конекције између два рачунара на клијентовом рачунару омогућен рад као да се налази у лабораторији, NeuroArm је могуће програмирати као да се ради за рачунаром у лабораторији без икаквих ограничења.

*DD-WRT решење (<http://dd-wrt.com>)*

DD-WRT је фирмвер који је могуће инсталирати на велики број рутера који су компатибилни. Разлог за коришћење овог фирмвера је што омогућава знатно бољу сигурност, али и подршку за већи број DDNS, као што се може видети на слици. Подржан је већи број како бесплатних тако и DNS сервера који се плаћају. За наше потребе даљинског приступа роботу изабрали смо Dynu (<https://www.dynu.com/>) решење које у свом бесплатном облику потпуно задовољава све захтеве. Са овим решењем наш програм NeuroArm manipulator може директно приступити роботу и у потпуности га контролисати, односно манипулисати роботском руком, без промене PID алгоритама мотора.



Слика 5.13. Повезивање преко рутера

У NeuroArm manipulator програму као адресу „mfneuoarm422.ddnsfree.com“ којој се приступа потребно је откуцати адресу хоста. Рад надаље је потпуно идентичан као да се ради за самим рачунаром који је директно повезан на NeuroArm роботском руком.

### 5.4.3. Управљање путем мобилног телефона

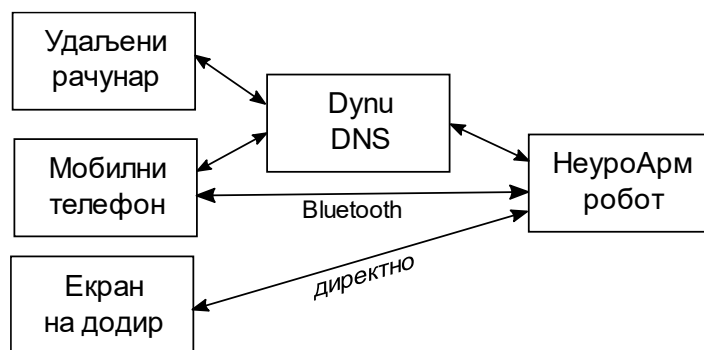
Један од могућих начина управљања је и управљање путем апликације са мобилног телефона [119]. Један од начина конекције мобилног телефона и развојне плоче је путем Bluetooth конекције, а друга је путем мрежног приступа. Апликација за мобилни телефон би користила највећи део кода који је већ написан за апликацију NeuroArm manipulator.

#### 5.4.4. Визуелна контрола рада робота

Иако путем повратних информација са NeuroArm manipulator програма добијамо повратне информације о положају сваког дела роботске руке (угла), брзини рада мотора потребно је имати и визуелни преглед рада руке, односно визуелну контролу рада. Један пример је да се то омогући тако што ће се две ИП камере поставе на две позиције тако да добијамо комплетну слику робота из два угла. Камере треба да имају добру резолуцију, на пример (камере Xiaomi произвођача) које подржавају 1080p (1920x1080) резолуцију. Камерама је могуће приступити преко програма произвођача који приказује истовремено слике са обе камере, или увећану слику преко целог екрана само једне камере.

#### 5.5. Управљање преко контролног панела

Управљање преко контролног панела, у овом случају екрана осетљивог на додир, је у плану имплементисати ради заокруживања целине система управљања. На екрану би се налазиле све опције као у програму NeuroArm manipulator, а извођење је планирано на два начина. Први начин је повезивање екрана директно на Beaglebone Black развојну плочу, што би значило да се комплетна логика управљања врши директно преко развојне плоче, док је друга варијанта да се Arduino развојна плоча постави као интерфејс између Beaglebone Black развојне плоче и екрана на додир, а на Arduino плочи би се испрограмирала комуникација путем Modbus протокола слично као и на РС рачунару.



Слика 5.14. Управљање NeuroArm роботском руком

## 6. Закључак и научни доприноси докторске дисертације

Предмет истраживања ове докторске дисертације је пројектовање савремених система управљања робота применом развојних програмабилних система и савремене теорије рачуна нецелог реда. При томе, разматрани су роботске системи који имају структуру отвореног кинематичког ланца (са три и више степени слободе).

Приликом формирања одговарајућег математичког модела разматраног роботског система, као и за управљање кретањем роботског система, примењена је савремена теорија који представља генерализацију класичног интегро-диференцијалног рачуна која је позната у научној јавности под називом теорија фракционог рачуна (теорија нецелог реда).

Са друге стране, имајући у виду значај практичне реализације и имплементације претходно развијених алгоритама управљања и унапређењу постојећег роботског система, значајна пажња је овде посвећена избору и имплементацији напредних хардверских решења за управљање постојеће NeuroArm роботске руке. За кључне научне доприносе ове докторске дисертације може се издвојити следеће:

- Формирање математичког модела роботског система са хватаљком у случају везаног кретања истог када је остварен контакт хватаљке са радном површи или кретање по задатој линији, применом Лагранжевих једначина друге врсте у коваријантном облику, где је посебно примењен Родригов приступ (Родригова матрица трансформације).
- Одређивање Јакобијан матрице при везаном кретању врха хватаљке роботског система. При томе, роботски системи који остварују контакт са радном површи представљају једну ширу класу динамичких система и могу се моделирати као сингуларни системи целог реда и у општем случају нецелог реда.
- Решавање задатка управљања датом класом динамичких линеаризованих роботских система применом интелигентног робусног ИУУ нецелог реда, у отвореној грани, типа  $PD^2D^\alpha$ . Роботски систем који је нелинеаран је линеаризован применом поступка *feedback* линеаризације. Овде се може нагласити и истаћи чињеница да у научно-стручној литератури постоји изузетно мали број алгоритама ИУУ нецелог реда који су примењени за динамичке системе који нису нецелог реда већ целог реда чиме горе наведен резултат добија на још већем значају. Формулисана је и доказана теорема која се односи на конвергенцију предложеног ИУУ нецелог реда типа  $PD^2D^\alpha$  чиме су добијени довољни услови конвергенције.
- Управљање датим сингуларним системом нецелог реда применом ИУУ нецелог реда. Посебно се разматра класа сингуларних система нецелог реда где постоји

параметарска неизвесност. Само робусно управљање ИУУ је пројектовано у отворено-затвореној спрези типа  $P/PD^\alpha$ . И овде је формулисана и доказана одговарајућа теорема где су изведени су довољни услови за конвергенцију предложеног алгоритма ИУУ нецелог реда. Симулациони резултати су показали оправданост и ефикасност предложеног интелигентног робусног ИУУ нецелог реда предложеног типа  $P/PD^\alpha$ .

- Решавање проблема управљања постојећим NeuroArm роботским системом применом итеративног управљања учењем у отворено-затвореној спрези нецелог реда  $PD^\mu/PD$  типа. Уочено је да на датом роботском систему постоји велики степен преноса тј. редуције чиме је омогућено занемаривање нелинеарне спреге што је за пројектовање управљачког система даље искоришћен линеарни модел истог. Предложен је и примењен напредни  $PD^\mu$  тип ИУУ поставља у директној грани, док је класични ПД контролер примењен у повратној грани. Дати су резултати симулације која потврђује ефикасност напредног ИУУ алгоритма  $PD^\mu/PD$  типа са посебним освртом на утицај нецелог извода  $\mu$  као параметра.
- Пројектовање новог управљачког система за рад у реалном времену користећи развојну плочу и оперативни систем који подржава Open source приступ што даје нове могућности, на пример додавање нових сензора и прилагодљивост роботског система новим захтевима. Тако је направљена и имплементирана плоча за управљање свих седам мотора једносмерне струје који се налазе на роботу и која омогућава читавање са три постојећа енкодера и седам потенциометара ради утврђивања тачне позиције сваког зглоба NeuroArm роботске руке. Инсталиран је оперативни систем Linux Debian а кернел Debian Linux-а је „печован“ Xenomai 3 PREEMPT-RT (Mercury језгром) ради управљања у реалном времену. Развојна плоча Beaglebone Black је изабрана и овде по први пут коришћена за управљање роботским системом са седам степени слободе. Такође, је по први пут коришћен програмски пакет EDICOPT, Института Михаило Пупин, за потребе роботског управљања, за програмирање сложених алгоритама управљања индустријски прихваћеним FBD-ом и ST кодом.
- Развој новог програма за управљање NeuroArm роботском руком (до свих седам мотора у исто време) као и аквизицијом података о тачном положају сваког зглоба ради упоређивања експерименталних података са подацима добијеним симулацијом путем Matlab-а и Simulink-а. Овакав приступ избора хардвера и испрограмираних могућности омогућава и даљински рад са NeuroArm роботском руком, што отвара потпуно нове могућности примене оваквог роботског система.

## Литература

- [1] B. Siciliano, L. Sciavicco, L. Villani, Oriolo G. Robotics, London: Springer-Verlag, 2009.
- [2] H. G. Sage, M. F. De Mathelin, Robust control of robot manipulators: A survey, *Inter J Control*, 76 (1999) 1498-1522.
- [3] G. K. McMillan, Industrial applications of PID control In: Vilanova R, Visioli A, eds. *PID control in the third millennium*, London: Springer, 2012.
- [4] T. B. Šekara, M. R. Mataušek, Optimization of PID Controller Based on Maximization of the Proportional Gain Under Constraints on Robustness and Sensitivity to Measurement Noise, *IEEE Trans Automat Contr*, 54 (2009) 184-189.
- [5] C. Yung, X. Chengying, *Intelligent Systems: Modeling, Optimization, and Control*, 1st ed, CRC Press, 2017.
- [6] S. Arimoto, S. Kawamura, Bettering operation of robots by learning, *J Robotic Syst*, 1 (1984) 123-140.
- [7] M. Uchiyama, Formulation of high-speed motion pattern of a mechanical arm by trial, *Trans Soc Inst Cont Eng.*, 14 (1978) 706-712.
- [8] M. Lazarević, T. Panagiotis, Robust second-order  $PD\alpha$  type iterative learning control for a class of uncertain fractional order singular systems, *Journal of Vibration and Control*, Sage Journals, (2016) 2004 - 2018.
- [9] I. Podlubny, *Fractional Differential Equations*, San Diego: Academic Press, 1999.
- [10] T. M. Atanacković, S. Pilipović, B. Stanković, *Fractional calculus with applications in mechanics: wave propagation, impact and variational principles.*, Wiley, 2014.
- [11] V. Lakshmikantham, Theory of fractional functional differential equations, *Nonlinear Analysis:TMA*, 69 (2008) 3337-3343.
- [12] A. Kilbas, H. Srivastava, J. Trujillo, *Theory and Applications of Fractional Differential Equations*, North-Holland Mathematics Studies, Elsevier, Amsterdam, NEDERLAND, 204

(2006).

- [13] C. A. Monje, Q. Y. Chen, M. B. Vinagre, Y. D. Xue, V. Feliu-Batlle, *Fractional-order Systems and Controls: Fundamentals and Applications*, Springer-Verlag London, 2010.
- [14] M. P. Lazarević, Iterative Learning Control of Integer and Noninteger Order: an Overview, *Scientific Technical Review*, 64:1 (2014) 35-47.
- [15] P. Feng-Xian, Z. Qing-Ling, W. Zhe-Feng, Iterative Learning Control for a Class of Singular Systems, *Acta Automatica Sinica*, 33:6 (June 2007) 658–659.
- [16] T. Senping, H. Gang, Z. Lun, Iterative Learning Control for the Singular Systems with Delay, *Proc. the 26th Chinese Control Conf.*, (July 2007) 26-31.
- [17] M. P. Lazarević, PD $\alpha$  type iterative learning control for fractional LTI system, *Proc. 4th Int. Carpath. Control Conf.*, High Tatras, Tatr. Lomnica, Slovak Republic, (2003) 869–872.
- [18] M. P. Lazarević, PD $\alpha$  type iterative learning control for fractional LTI system, 16th *Int.Cong. CHISA 2004*, Praha, Czech Republic, (2004) 22-26.
- [19] M. Lazarević, N. Đurović, B. Cvetković, P. Mandić, M. Cajić, Type Iterative Learning Control for Fractional-order Singular Time-delay System, 29th *Chinese Control And Decision Conference (CCDC)*, 28 – 30 May 2017, Chongqing, China, (2017) 1933-1938. ISBN: 978-1-5090-4656-0.
- [20] M. Lazarević, N. Đurović, B. Cvetković, P. Mandić, L. Bučanović, Fractional-order Iterative Learning Control for Singular Fractional order System: (P)- PD $\alpha$  Type, *Scientific Technical Review*, 6:3 (2016) 40-49.
- [21] M. P. Lazarević, Iterative Learning Feedback Control for Nonlinear Fractional Order System- PD $\alpha$  Type, *The 4th IFAC Workshop Fractional Differentiation and its Applications*, *Proceedings of FDA'10*, Badajoz, Spain, (October 18-20 2010).
- [22] M. Lazarević, Some New Results on Iterative Learning Control of Noninteger Order, *Scientific Technical Review*, 64:2 (2014c) 31-39.
- [23] M. Lazarević, P. Tzekis, Iterative Learning Feedback Control for Singular Fractional Order System- PD $\alpha$  Type, *IcETRAN 2014*, Vrnjačka Banja, Serbia, (June 2 - 5 2014) AUI2.3, 1-6.
- [24] Z. Q. Chen, K. L. Moore, On type iterative learning control, In *Proc. 40th IEEE Conference*

- on Decision and Control, (2001) 4451-4456.
- [25] Y. Li, Y. Q. Chen, H. S. Ahn, Fractional-order iterative learning control for fractional-order linear systems, *Asian J. Control*, (2011) 54-63.
  - [26] A. Oustaloup, X. Moreau, M. Nouillant, The CRONE suspension, *Control Eng. Practice*, 4 (1996) 1101-1108.
  - [27] I. Podlubny, Fractional-order systems, *IEEE Trans. Autom. Cont*, 44 (1999) 208–214.
  - [28] M. Safaei, S. Tavakoli, Tuning of robust fractional-order phase-lead compensators using pole placement and pole-zero ratio minimization, *Journal of Vibration and Control*, (2018) 1-12.
  - [29] D. Molloy, *Exploring BeagleBone, Tools and Techniques for Building with Embedded Linux®*, Wiley, 2019.
  - [30] B. Cvetković, M. Lazarević, T. Maneski, P. Mandić, B. Lutovac, T. Šekara, Akvizicija podataka korišćenjem razvojne ploče RASPBERRI PI model B, *Informacione tehnologije*, Univerzitet Crne Gore Elektrotehnički fakultet, (2016) 69-72.
  - [31] J. Blum, *Exploring Arduino: Tools and Techniques for Engineering Wizardry*, Wiley, 2019.
  - [32] D. Molloy, *Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux*, Wiley, 2016.
  - [33] B. Amos, *Hands-On RTOS with Microcontrollers: Building real-time embedded systems using FreeRTOS, STM32 MCUs, and SEGGER debug tools*, Packt Publishing, 2015.
  - [34] J. Pratt, S. Siewert, *Real-time embedded components and systems : with Linux and RTOS*, Mercury Learning and Information, 2016.
  - [35] C. Simmonds, *Mastering Embedded Linux Programming, Second Edition*, Packt Publishing, 2017.
  - [36] D. Barać, *Real-Time operativni sistemi za male embedded sisteme*, Elektronski fakultet, Univerzitet u Nišu, 2010.
  - [37] J. Albahari, *C# 9.0 in a Nutshell: The Definitive Reference*, O'Reilly Media.



- [38] K. C. Wang, *Embedded and Real-Time Operating Systems*, Springer, 2017.
- [39] W. Bohm и M. Broy, *Model-Based Engineering of Collaborative Embedded Systems*, Springer, 2021.
- [40] D. Lacamera, *Embedded Systems Architecture: Explore architectural concepts, pragmatic design patterns, and best practices to produce robust systems*, Packt Publishing, 2018.
- [41] J. Catsoulis, *Designing Embedded Hardware: Create New Computers and Devices Second Edition*, O'Reilly Media, 2005.
- [42] J. Peckol, *Embedded Systems: A Contemporary Design Tool*, Wiley, 2019.
- [43] A. Bonci, P. Massimilian, L. Sauro, *Embedded solutions for a class of highly unstable, underactuated and self-balancing robotic systems*, *EURASIP Journal on Embedded Systems* volume, 2017.
- [44] F. Englberger, T. Latzel, P. Sotiriadis, *An Autonomous Robot for Embedded Systems and Robotics*, 2018 12th European Workshop on Microelectronics Education (EWME).
- [45] W. M. L. W. Zainudin, R. L. A. Shauri, M. I. Roslan, M. A. Rosli, M. F. M. Ariffin, *New Interface using Beaglebone Black for 4-DOF Robot Arm System*, 2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA).
- [46] M. Pilatásig, F. Silva, G. Chacón, V. Tapia, J. Espinoza, E. X. Castellanos, L. Guerrero, J. Espinosa, *Interactive System Using Beaglebone Black with LINUX Debian for Its Application in Industrial Processes*, *ICITS 2018: Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)* , 573-582.
- [47] G. K. Kalyani, Z. Yang, V. Gandhi, T. Geng, *Using Robot Operating System (ROS) and Single Board Computer to Control Bioloid Robot Motion*, *TAROS 2017: Towards Autonomous Robotic Systems*, (2017) 41-50.
- [48] S. Venkateswaran, D. Chablat, R. Ramachandran, *Prototyping a piping inspection robot using a BeagleBone black board*, *Conference: Congrès Français de Mécanique (CFM) 2019*.
- [49] A. Andreychuk, A. Bokovoy, K. Yakovlev, *An Empirical Evaluation of Grid-based Path Planning Algorithms on Widely Used in Robotics Raspberry Pi Platform*, *ICAROB 2018: PROCEEDINGS OF THE 2018 INTERNATIONAL CONFERENCE ON ARTIFICIAL LIFE AND ROBOTICS*, 2018.

- [50] V. Čović и M. Lazarević, *Mehanika robota*, Beograd: Mašinski fakultet, 2009.
- [51] A. I. Lurje, *Analytical Mechanics*, Moscow, 1961.
- [52] K. Moore, *Iterative learning control for deterministic systems*, London: pringer-Verlag, 1993.
- [53] Z. Bien, J. X. Xu, *Iterative Learning Control: Analysis, Design, Integration and Application*. Kluwer, Netherlands,, Boston: Kluwer Academic Publishers, 1998.
- [54] H. Ahn, K. Moore, Y. Chen, *Iterative learning control robustness and monotonic convergence for interval systems*. 1st ed., London: Springer-Verlag London Limited, 2007.
- [55] G. Casalino, G. Bartolini, A learning procedure for the control of movements of robotic manipulators, *IASTED Symp. on Robotics and Automation*, (1984) 108–111.
- [56] Z. Bien, K. M. Huh, Higher-Order Iterative Control Algorithm, *IEE Proceedings Part D, Control Theory and Applications*, (1989) 105-112.
- [57] H. Lee, Z. Bien, Study on robustness of iterative learning control with non-zero initial error., *Int. J. of Control*, 64 (1996) 345-359.
- [58] M. P. Lazarević, P. D. Mandić, S. Ostojić, Further results on advanced robust iterative learning control and modeling of robotic systems, *Proc IMechE Part C: J Mechanical Engineering Science* 0(0), (2020) 1-16. doi: 10.1177/0954406220965996.
- [59] M. P. Lazarević, P. D. Mandić, B. P. Cvetković, L. Bučanović, M. Dragović, Advanced open-closed-loop PID2/PID type ILC control of a robot arm, *2018 Innovations in Intelligent Systems and Applications (INISTA)*, (2018) 1-8.
- [60] M. Lazarević, P. Mandić, B. Cvetković, T. Šekara, B. Lutovac, Some electromechanical systems and analogies of mem-systems integer and fractional order, *2016 5th Mediterranean Conference on Embedded Computing (MECO)*, (2016) 230-233.
- [61] I. Petras, *Handbook of Fractional Calculus with Applications*, Vol. 6: Applications in Control, Boston: Walter de Gruyter GmbH, 2019.
- [62] M. Lazarević, *Advanced Topics on Applications of Fractional Calculus on Control Problems, System Stability And Modeling*, WSEAS, ID 9028, (2014a) 202. ISBN: 978-960-474-348-3.

- [63] M. A. Nakhshev, *Fractional Calculus and Its Applications*, Moscow: FIZMATLIT, 2003.
- [64] X. Zhang, Some results of linear fractional order time-delay system, *Applied Mathematics and Computation*, 197:1 (2008) 407–411.
- [65] M. Č. Bošković, M. R. Rapaić, T. B. Šekara, P. D. Mandić, M. P. Lazarević, B. Cvetković, B. Lutovac, M. Daković, On the Rational Representation of Fractional Order Lead Compensator using Padé Approximation, 2018 7th Mediterranean Conference on Embedded Computing (MECO), 2018.
- [66] Y. Li, Q. Y. Chen, S. Ahn, G. Tian, A survey on fractional-order iterative learning control, *J. Optim. Theory Appl.* 1, 15:6 (2013) 127–140.
- [67] X. Liu, L. Y. Zandang, L. Y. Yanmin, The convergence of iterative learning control for some fractional system, *Advances in Difference Equations*, 2017.
- [68] Y. Zhao, F. Zhou, D. Wang, Y. Li, Fractional-order iterative learning control with initial state learning design, *Nonlinear Dyn*, 90 (2017) 1257–1268. doi: 10.1007/s11071-017-3724-6.
- [69] Y. Zhao, F. Zhou, D. Wang, Y. Li, Design and Analysis of Fractional Order PD $\alpha$ -Type Iterative Learning Control, *IFAC PapersOnLine* 50-1, (2017) 8077–8083.
- [70] B. P. Cvetković, M. P. Lazarević, Fractional-order iterative learning control for robotic arm – PD $2D\alpha$  type, *Filomat*, 35:1 (2021), 1-10. <https://doi.org/10.2298/FIL2101001C>
- [71] L. Slotine J.J., *Applied nonlinear control* Vol. 199, NJ: Prentice-Hall Englewood Cliffs, 1991.
- [72] K. I. Khalil, *Nonlinear Systems*, New Jersey: Prentice-Hall, 2002.
- [73] Z. J. Xiaoe, Convergence monotonicity and speed comparison of iterative learning control algorithms for nonlinear systems, *IMA Journal of Mathematical Control and Information* , 30 (2013) 473–486. doi:10.1093/imamci/dns034.
- [74] H. H. Rosenbrock, Structural properties of linear dynamical systems, *Int. J. Control*, 20 (1974) 191-202.
- [75] P. Muller, Modelling and control of mechatronic systems by the descriptor approach, *Journal of theoretical and applied mechanics*, 43:3 (2005) 593-607.
- [76] W. Kampowski, P. Rentrop, W. Schmidt, Classification and numerical simulation of electric

- circuits, *Surv. Math. Ind.*, 2 (1992) 23-65.
- [77] N. Mc-Clamroch, Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems, *Proc. of the 1986 IEEE International Conf. on Robotics and Automations*, 3 (1986) 21-28. doi: 10.1109/ROBOT.1986.1087712
- [78] M. Buzurović, Dynamic Model of Medical Robot Represented as Descriptor System, *Int. Journal of Inf. And System Science*, 2 (2008) 316-333.
- [79] L. Debeljković, D. H. Owens, On practical stability of singular systems, *Proc. Melecon Conf. 85, Madrid, Spain, (1985)* 103-105.
- [80] D. L. Debeljković, S. Milinković, M. B. Jovanović, *Kontinualni singularni sistemi automatskog upravljanja*, Beograd: GIP Kultura, 1996.
- [81] L. Dai, *Singular Control Systems*, Springer-Verlag Berlin Heidelberg, 1989.
- [82] C. R. Hui, A. A. Golenberg, Modelling the Manipulation of Rigid Objects as Singular Systems, *Proc. IEEE Intl. Conf. on Robotics and Automation*, (1988) 240-244.
- [83] M. Lazarević, B. Cvetković и P. Mandić, Closed-loop iterative learning control for fractional-order linear singular time-delay system: PD $\alpha$ -type, *Scientific Technical Review*, 2018.
- [84] A. P. Singh, D. Debb, H. Agarwal, On selection of improved fractional model and control of different systems with experimental validation, *Communications in Nonlinear Science and Numerical Simulation*, 79 (December 2019).
- [85] A. P. Singh, D. Deb, H. Agrawal, K. Bingi, Stepan Ozana, Modeling and Control of Robotic Manipulators: A Fractional Calculus Point of View, *Arabian Journal for Science and Engineering*, (2021). <https://doi.org/10.1007/s13369-020-05138-6>.
- [86] J. E. Lavin-Delgado, K. M. Owolabi, J. F. Gomez-Aguilar, G. Fernandez-Anaya, E. Hernandez-Castillo, Modelling of Chaotic Processes with Caputo Fractional Order Derivative, *Entropy*, 22:9 (2020) 1027.
- [87] Y. Yao, J. Zhuang, S. Chang-Yin, Sufficient and Necessary Condition of Admissibility for Fractional-order Singular System, *Acta automatica sinica*, 39:12 (2013) 2160-2164. [https://doi.org/10.1016/S1874-1029\(14\)60003-3](https://doi.org/10.1016/S1874-1029(14)60003-3).

- [88] Ł. Sajewski, Solution of the State Equation of Descriptor Fractional Continuous-Time Linear Systems with Two Different Fractional. In: Szewczyk R., Zieliński C., Kaliczyńska M. (eds) Progress in Automation, Robotics and Measuring Techniques. ICA 2015., Advances in Intelligent Systems and Computing, 350 (2015).
- [89] M. Ortiguiera, F. Coito, The initial conditions of Riemann-Liouville and Caputo derivatives, 6th EUROMACH Nonlinear Dynamics Conference, 2008.
- [90] M. Lazarević, P. Mandić, Feedback-feedforward iterative learning control for fractional-order uncertain time delay system—PD type, ICFDA'14 Catania, (23-25 June 2014).
- [91] B. Cvetković, M. Lazarević, N. Đurović, P. Mandić, Open-closed loop fractional-order iterative learning control for singular fractional-order system, International Conference on Fractional Differentiation and its Application, ICFDA2016, Novi Sad, (18-20 July 2016) 404-414.
- [92] Z. Chen, Z. Gong, C. Wen, Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays., Automatica, 34:3 (1998) 345–353.
- [93] P. D. Mandić, M. P. Lazarević, T. B. Šekara, M. Č. Bošković, G. Maione, Robust control of robot manipulators using fractional order lag compensator, 7th International Congress of Serbian Society of Mechanics, Sr. Karlovci, Serbia, (June 24-26, 2019).
- [94] J. Samardžić, M. Lazarević, B. Cvetković, Optimal conventional and fractional PID control algorithm for a robotic system with three degrees of freedom driven by DC motors, 2011 IEEE 9th International Symposium on Intelligent Systems and Informatics, 2011.
- [95] M. Č. Bošković, M. R. Rapaić, T. B. Šekara, P. D. Mandić, M. P. Lazarević, A novel method for design of complex compensators in control systems, 18th International Symposium INFOTEH-JAHORINA, 2019.
- [96] K. J. Åström, R. M. Murray, Feedback Systems- An Introduction for Scientists and Engineers, Princeton, USA: Princeton University Press, 2008.
- [97] R. Oshana и M. Kraeling, Software engineering for embedded systems, Elsevier, 2019.
- [98] A. Koubaa, Robot Operating System (ROS), The Complete Reference (Volume 3), Springer, 2018.
- [99] E. White, Making Embedded Systems: Design Patterns for Great Software, O'Reilly Media,

2011.

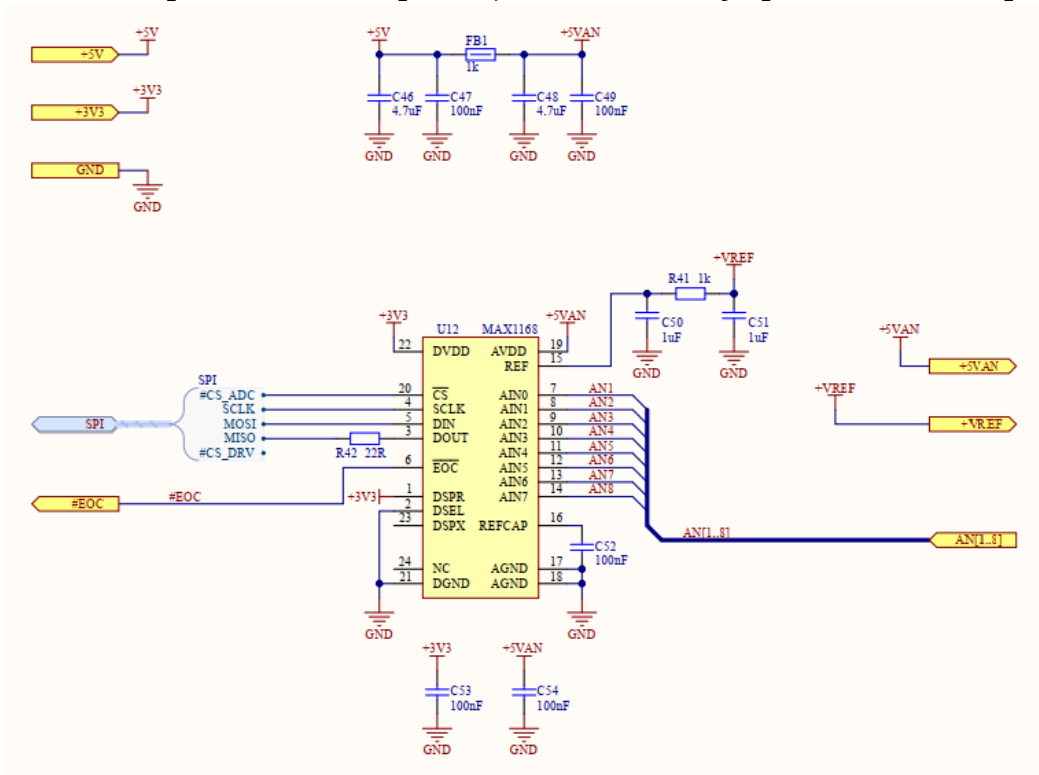
- [100] C. G. Nayana, D. Mamatha, M. Nandini, S. Nandini, Development of Autonomous Wall Painting Robot using Raspberry Pi and Image Processing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IETE – 2020), 8:11 (2020).
- [101] G. O. E. Abdalla, T. Veeramanikandasamy, Implementation of spy robot for a surveillance system using Internet protocol of Raspberry Pi, 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017.
- [102] P. Korke, Robotics, Vision and Control Fundamental Algorithms in MATLAB®, Springer, 2017.
- [103] B. Cvetković, V. Nešić, M. Lazarević, P. Mandić, P. Marić, M. Dragović, Advanced hardware control for seven DOFs robotic arm-neuro arm, IOP Conf. Ser. Mater. Sci. Eng., (2018), doi:10.1088/1757-
- [104] A. Telesis, Modbus guide, Feature Overview and Configuration Guide, 2020.
- [105] M. J. Price, C# 9 and .NET 5 - Modern Cross-Platform Development, Packt Publishing, 2020.
- [106] J. Skeet, C# in Depth, Manning Publications, 2019.
- [107] S. G. Tzafestas, Web-Based Control and Robotics Education, Springer, 2019.
- [108] S.-H. Yang, Internet-Based Controls Systems, Springer, 2011.
- [109] G. Cook, Navigation, Control and Remote Sensing, Wiley, 2011.
- [110] M. Banduka, M. Lutovac, Remote Monitoring and Control of Industrial Robot based on Android Device and Wi-Fi Communication, AUTOMATIKA, 3 (2015) 281–291.
- [111] W. Sun, C. Liu, J. Zhu, A Remote Controlled Mobile Robot Based on Wireless Transmission, 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference(IMCEC 2018).
- [112] M. N. Almali, K. Gürçam и A. Bayram, Wireless Remote Control of a Mobile Robot, International Journal of Scientific Research in Information Systems and Engineering

(IJSRISE) , т. 1 (2), December-2015.

- [113] B. Zhou, W. Chou и S. Wu, Remote control system of mobile robot based on cloud platform, 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), 2017.
- [114] B. M. Wilamowski и J. d. Irwin, Industrial communication systems, SRC Press, 2011.
- [115] K. Kyunghoon, B. Soonil и H. Kwanghak, Intelligent surveillance and security robot systems, 2010 IEEE Workshop on Advanced Robotics and its Social Impacts, 2010.
- [116] J. A. Kurdila и P. Ben-Tzvi, Dynamics and Control of Robotic Systems, NJ, USA,: John Wiley & Sons, 2020.
- [117] P. Nyakomitta, S. Abeka, Security Investigation on Remote Access Methods of Virtual Private Network, Global Journal of Computer Science and Technology, 20:1 (2020).
- [118] M. Abdullah, Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data, Cryptography and Network Security, 2017.
- [119] C. Parga, X. Li, W. Yu, Smartphone-Based Human Machine Interface with Application to Remote Control of Robot Arm, Conference: Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013.

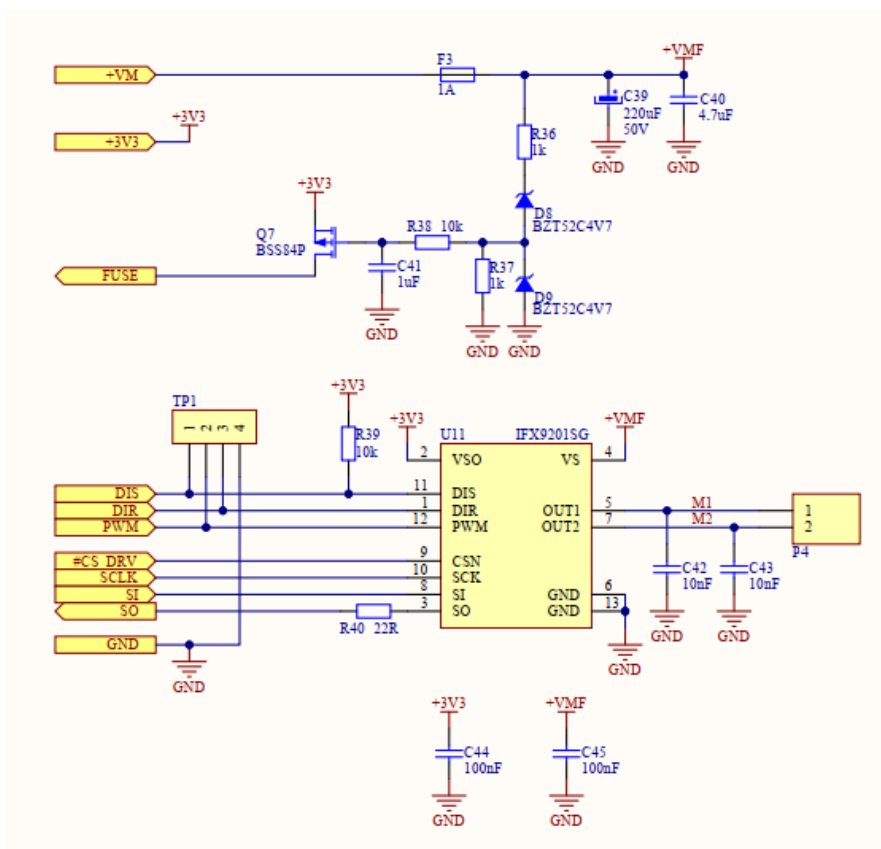
# Прилог

## П1. Шематски приказ делова развијене плоче за управљање моторима



Слика П1. Аналогни конвертер





Слика П2. драјвер мотора једносмерне струје

## П2. Тест, модел Махон мотора у Matlab-у и Simulink-у и експериментални резултати

Табела П1. Фабричке карактеристике мотора

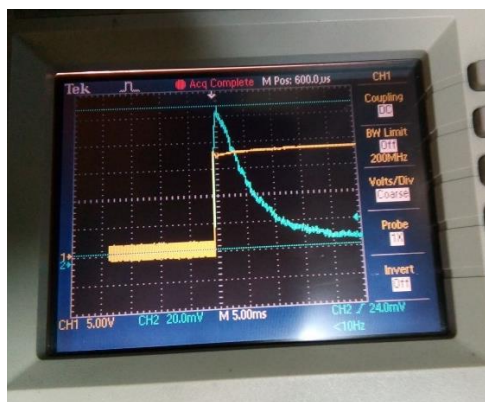
Карактеристика	Вредност
Снага	70 W
Напон	32 V
Брзина у празном ходу	6790 obr./min
Струја празног хода	89 mA
Полазни момент	832 mNm
Полазна струја (кратког споја)	18.7 A
Отпорност намотаја	1.71 $\Omega$
Максимална брзина	8200 obr./min
Максимална трајна струја	1.99 A
Максимални момент у трајном раду	88.5 mNm
Максимални степен искоришћења	86 %
Механичка временска константа	6 ms

Момент инерције ротора	65.2 gcm <sup>2</sup>
Индуктивност намотаја	0.3 mH
Термичка константа намотаја	41 s
Брзина/момент градијент	8.27 min <sup>-1</sup> /mNm
Максимална излазна снага при номиналном напону	146 W
Константа обртног момента	44.5 mNm/A
Преносни однос редуктора	74/1

На Слици П3 дат је приказ одзива струје (плави сигнал) и напона мотора (жути сигнал) снимљених помоћу осцилоскопа за случај када је напон  $U=18V$ .

Вршна вредност струје снимљена на осцилоскопу одговара струји кратког споја, а вредност на којој се струја усталила одговара струји реалног празног хода.

Како би се добила брзина обртања ротора (брзина пре редуктора) потребно је измерену брзину помножити преносним односом редуктора ( $i=74$ ). Израчунате вредности брзина за реалан и идеалан празан ход дате су у табели П2.

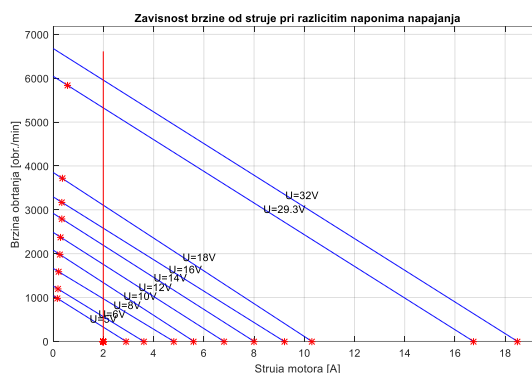
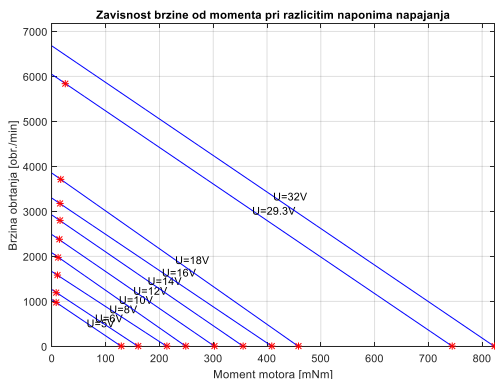


Слика П3. Приказ одзива струје и напона мотора

Табела П2. Израчунате вредности реалног и идеалног хода у зависности од напона

U [V]	n <sub>0realno</sub> [obr./min]	n <sub>0idealno</sub> [obr./min]
5	979.1	1040
6	1197.1	1216.9
8	1585.7	1663
10	1979.5	2079.8
12	2374.3	2483.9
14	2796	2919.3
16	3169.2	3294.5
18	3715.5	3850
29.3	5842.1	6044.3

<b>32</b>	/	<b>6677.2</b>
-----------	---	---------------



Слика П4: Зависност брзине од момента    Слика П5: Ограничење по струји

Процењена брзина идеалног празног хода при номиналном напону  $n_{0idealno} = 6677.2$  obr./min блиска је назначеној вредности брзине празног хода од  $n_{0idealno} = 6790$  obr./min, док је струја кратког споја у овом случају од 18.49A такође веома блиска назначеној вредности од 18.7A.

Пошто је позната константа момента  $k_m = M_{em}/I$ , дата од стране произвођача, може се струја приказана на апсциси превести у момент мотора. Зависност брзина-момент дата је на Слици П4.

Табела П3

U [V]	M <sub>ks</sub> [mNm]	M <sub>ph</sub> [mNm]
<b>5</b>	129	7.6
<b>6</b>	160.2	8.23
<b>8</b>	213.6	9.92
<b>10</b>	249.2	12.01
<b>12</b>	302.6	13.35
<b>14</b>	356	15.04
<b>16</b>	409.4	15.58
<b>18</b>	458.35	16.02
<b>29.3</b>	744.8	24.92
<b>32</b>	822.77	/

Процењена вредност момента кратког споја (полазног момента) при ном. напону  $M_{ks}=822.77mNm$  приближна је назначеној вредности од 832mNm. Момент реалног празног хода при номиналном напону није могуће одредити на овај начин, јер није позната брзина при реалном празном ходу. Уколико би се мотор прикључио на ном. напон, ова брзина би се могла измерити, а на основу ње одредити и момент празног хода при номиналном напону.

Пошто је произвођач дефинисао да је брзинска константа  $k_n=215 \text{ min}^{-1}/V$ , где је  $k_n=n_0/U$ , могуће је проверити да ли се и овај параметар уклапа. Прорачуном је добијено да је  $k_n=208.6611 \text{ min}^{-1}/V$ , што је блиска вредност. Мерењем индуктивности намотаја помоћу RLC метра измерена је вредност од  $0.435 \text{ mH}$ . Грубом проценом индуктивности на основу брзине успостављања струје при прикључењу мотора на различите напоне, процене индуктивности варирају од око  $0.2$  до  $0.4 \text{ mH}$ , што је у прихватљивом опсегу.

Назначени градијент брзина/момент који даје произвођач износи  $8.27 \text{ rpm/mNm}$ , док се прорачуном на основу мерења добија да је овај коефицијент  $8.1154 \text{ rpm/mNm}$ .

Како је произвођач декларисао максималну трајну струју мотора од  $1.99 \text{ A}$ , на Слици Д5.12. су приказане исте зависности брзине од струје уз ограничење по струји. Црвеном вертикалом је означена граница за оптерећење мотора у трајном раду. Трајни рад у области десно од вертикале би изазвао преоптерећење мотора. Преведено у момент, оптерећење мотора би изазвао трајни момент већи од  $88.5 \text{ mNm}$  као што је дато од стране произвођача. Израчунато је да овом ограничењу одговара брзина од  $5958.5 \text{ obr./min}$ .

Прва диференцијална једначина електродинамичке напонске равнотеже:

$$U_a = R_a * i_a + L_a * \frac{di_a}{dt} + E_a$$

где су:

$U_a$  – напон напајања који се доводи на ротор (арматурни напон)

$i_a$  – арматурна, роторска струја

$R_a$  – електрична отпорност роторског намотаја

$L_a$  – индуктивност ротора

$E_a$  – индукована електромоторна сила

Одговарајућа диференцијална једначина кретања ротора је:

$$J \frac{dw_m}{dt} = M_{em} - M_m - k_f * w_m$$

где су:

$J$  – укупан момент инерције (ротор и редуктор)

$w_m$  – механичка угаона брзина ротора

$M_{em}$  – електромагнетски момент

$M_m$  – момент оптерећења

$k_f$  – коефицијент трења (фрикције)

Израз за електромагнетски момент

$$M_{em} = k_m * i_a$$

Израз за електромоторну силу

$$E_a = k_m * \omega_m$$

Применом Лапласове трансформације на претходне једначине добијају се следећи изрази:

$$I_a(s) = \frac{U_a(s) - E_a(s)}{R_a + L_a * s}$$

односно

$$I_a(s) = \frac{1}{R_a} * \frac{1}{1 + T_a * s} * (U_a(s) - E_a(s))$$

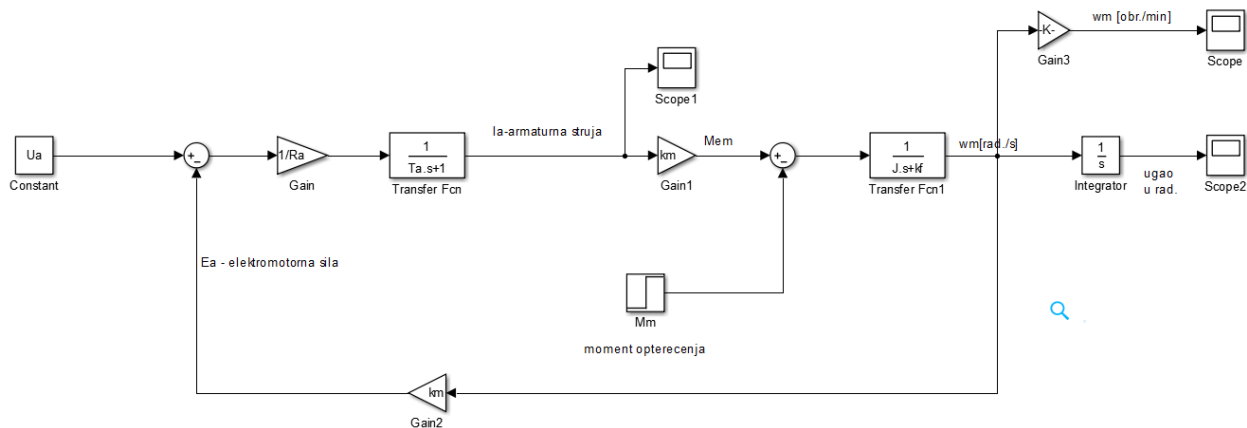
где је  $T_a = L_a / R_a$  електрична временска константа арматурног намотаја.

$$\omega_m(s) = \frac{k_m * I_a(s) - M_m(s)}{k_f + J * s}$$

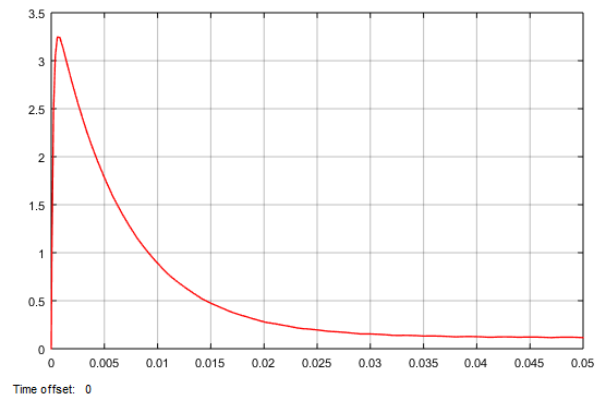
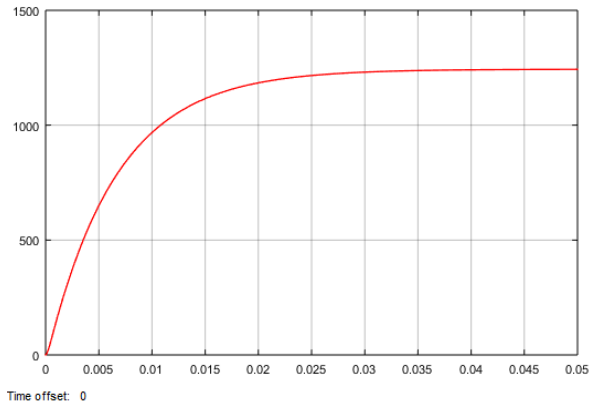
Док се описани угао обртања добија интеграљењем угаоне брзине:

$$\theta(s) = \frac{1}{s} * \omega_m(s)$$

ModelDCMotora.slx



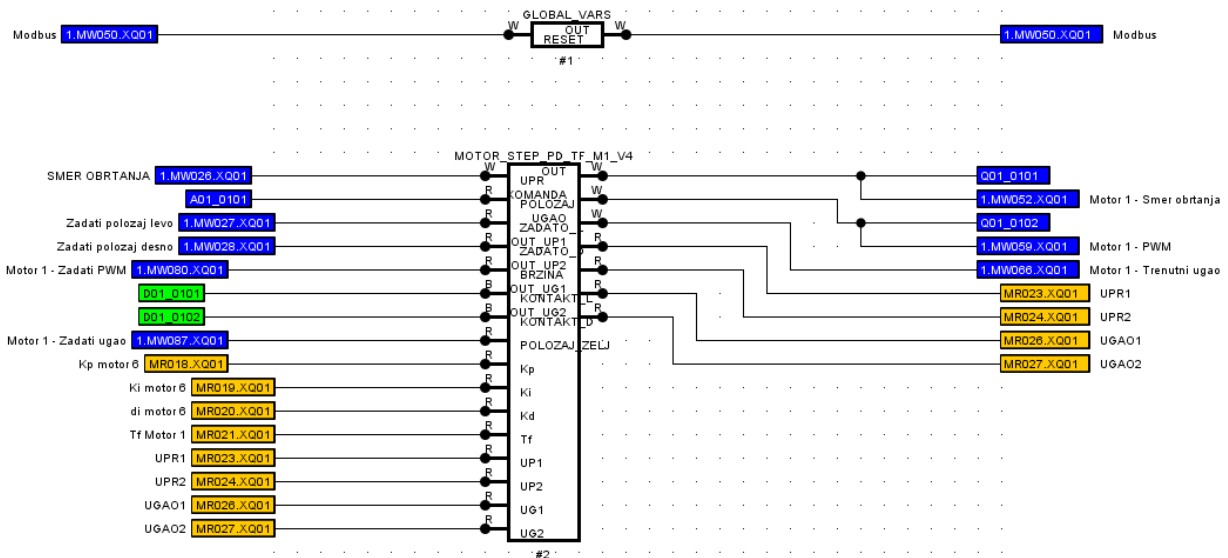
Слика П6: Симулинк модел Махон мотора једносмерне струје



Слика П7: Резултат симулације угаоне брзине Слика П8: Резултат симулације струје

Као што се може видети на Симулинк симулацији, резултат који нам даје модел је тачности од приближно 97%, што нам гарантује и високу тачност Симулинк симулације Максеновог мотора једносмерне струје као саставног дела роботске руке.

У програмском пакету „Edicort“ направљен је FBD (функционални блок дијаграм) управљања мотором



Слика П9: FBD мотора

СТ код:

```
#ifndef _MOTOR_STEP_PID_M7_V2_
#define _MOTOR_STEP_PID_M7_V2_

#include "timer.st"
#include "conv.st"

FUNCTION_BLOCK MOTOR_STEP_PID_M7_V2
```

```

VAR_INPUT
    UPR:WORD;
    POLOZAJ:REAL;
    ZADATO_L:REAL;
    ZADATO_D:REAL;
    BRZINA:REAL;
    KONTAKT_L:BOOL;
    KONTAKT_D:BOOL;
    POLOZAJ_ZELJ:REAL;
    Kp:REAL;
    Ki:REAL;
    Kd:REAL;
    POCUG:REAL;
END_VAR

VAR_OUTPUT
    OUT:WORD;
    KOMANDA:WORD;
    UGAO:WORD;
    STARTUGAO:REAL;
END_VAR

VAR
    CALCULATED:REAL;
    ERROR:REAL;
    LAST_ERROR:REAL;
    CUR_ANGLE:REAL;
    MOTOR_PWM:REAL;
    INT_PART:REAL;
    DIFF_PART:REAL;
    TIME_PER:REAL;
    CUR_ANGLE_RAD:REAL;
END_VAR

VAR_EXTERNAL
    _GlobalTimer: INT;
END_VAR

TIME_PER := 0.005;

CUR_ANGLE := (POLOZAJ - 38000.0) / 195.0;
CUR_ANGLE_RAD := CUR_ANGLE * 3.14 / 180.0;
ERROR := POLOZAJ_ZELJ * 3.14 / 180.0 - CUR_ANGLE_RAD;

IF (ERROR < 0.0) THEN
    ERROR := ABS(ERROR);
    KOMANDA := 2;
ELSIF (ERROR > 0.0) THEN
    KOMANDA := 1;
END_IF;

IF (POLOZAJ > 60000.0) THEN
    KOMANDA := 0;
END_IF;

```

```

IF (POLOZAJ < 38000.0) THEN
    KOMANDA := 0;
END_IF;

DIFF_PART := (ERROR - LAST_ERROR) / TIME_PER;
INT_PART := INT_PART + ERROR * TIME_PER;

CALCULATED := Kp * ERROR + (Ki * INT_PART) + (Kd * DIFF_PART);

MOTOR_PWM := BRZINA * CALCULATED * 4096.0 / 12.0;

IF (MOTOR_PWM > 900.0) THEN
    MOTOR_PWM := 900.0;
END_IF;
IF ((KOMANDA = 1) AND (MOTOR_PWM > 900.0)) THEN
    MOTOR_PWM := 900.0;
END_IF;
(* IF (POLOZAJ < ZADATO_D AND NO(KONTAKT_D)) THEN *)

MOTOR_PWM := BRZINA;
IF (KOMANDA = 2) THEN
    MOTOR_PWM := 0.0;
END_IF;

OUT := MOTOR_PWM;
UGAO := CUR_ANGLE;

END_FUNCTION_BLOCK

#endif

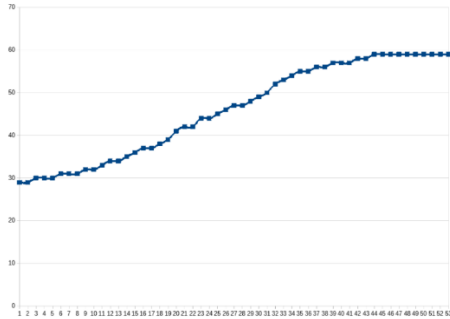
```

По покретању кода, добијени су следећи резултати који су табеларно приказани у Microsoft Excel програму (Слика П10).

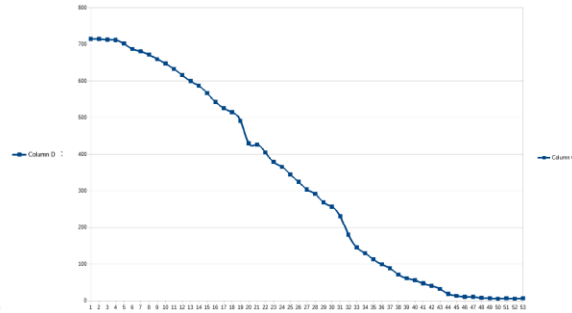
	A	B	C	D	E	F	G	H	I	J
1	1	767	715	29						
2	2	783	715	29				Legenda		
3	3	799	713	30				A: Redni broj iteracije		
4	4	815	712	30				B: Vreme ms kada se desio log		
5	5	831	702	30				C: Vrednost PWM-a		
6	6	847	688	31				D: Vrednost ugla		
7	7	863	681	31						
8	8	879	672	31						
9	9	895	660	32						
10	10	911	648	32						
11	11	927	633	33						
12	12	943	616	34						
13	13	959	600	34						
14	14	975	587	35						
15	15	992	567	36						
16	16	1013	543	37						
17	17	1029	526	37						
18	18	1045	515	38						
19	19	1066	491	39						
20	20	1098	430	41						
21	21	1120	426	42						
22	22	1136	405	42						
23	23	1154	379	44						
24	24	1171	366	44						

Слика П10: Експериментални резултат - табеларни приказ





Слика П11: Промена угла



Слика П12: Промена PWM-a

### П3. Комуникација са NeuroArm роботском руком

Подаци који се читају и пишу у NeuroArm робот

1.MW052.XQ01 0x00000001	Motor 1 - Smer obrtanja	10	HOLREG
1.MW053.XQ01 0x00000001	Motor 2 - Smer obrtanja	11	HOLREG
1.MW054.XQ01 0x00000001	Motor 3 - Smer obrtanja	12	HOLREG
1.MW055.XQ01 0x00000001	Motor 4 - Smer obrtanja	13	HOLREG
1.MW056.XQ01 0x00000001	Motor 5 - Smer obrtanja	14	HOLREG
1.MW057.XQ01 0x00000001	Motor 6 - Smer obrtanja	15	HOLREG
1.MW058.XQ01 0x00000001	Motor 7 - Smer obrtanja	16	HOLREG
1.MW059.XQ01 0x00000001	Motor 1 - PWM	17	HOLREG
1.MW060.XQ01 0x00000001	Motor 2 - PWM	18	HOLREG
1.MW061.XQ01 0x00000001	Motor 3 - PWM	19	HOLREG
1.MW062.XQ01 0x00000001	Motor 4 - PWM	20	HOLREG
1.MW063.XQ01 0x00000001	Motor 5 - PWM	21	HOLREG
1.MW064.XQ01 0x00000001	Motor 6 - PWM	22	HOLREG
1.MW065.XQ01 0x00000001	Motor 7 - PWM	23	HOLREG

1.MW066.XQ01 0x00000001	Motor 1 - Trenutni ugao	24	HOLREG
1.MW067.XQ01 0x00000001	Motor 2 - Trenutni ugao	25	HOLREG
1.MW068.XQ01 0x00000001	Motor 3 - Trenutni ugao	26	HOLREG
1.MW069.XQ01 0x00000001	Motor 4 - Trenutni ugao	27	HOLREG
1.MW070.XQ01 0x00000001	Motor 5 - Trenutni ugao	28	HOLREG
1.MW071.XQ01 0x00000001	Motor 6 - Trenutni ugao	29	HOLREG
1.MW072.XQ01 0x00000001	Motor 7 - Trenutni ugao	30	HOLREG

Подаци који се шаљу

1.MW080.XQ01 0x00000008	Motor 1 - Zadati PWM	100	HOLREG
1.MW081.XQ01 0x00000008	Motor 2 - Zadati PWM	101	HOLREG
1.MW082.XQ01 0x00000008	Motor 3 - Zadati PWM	102	HOLREG
1.MW083.XQ01 0x00000008	Motor 4 - Zadati PWM	103	HOLREG
1.MW084.XQ01 0x00000008	Motor 5 - Zadati PWM	104	HOLREG
1.MW085.XQ01 0x00000008	Motor 6 - Zadati PWM	105	HOLREG
1.MW086.XQ01 0x00000008	Motor 7 - Zadati PWM	106	HOLREG
1.MW087.XQ01	Motor 1 - Zadati ugao 107	HOLREG	0x00000008
1.MW088.XQ01	Motor 2 - Zadati ugao 108	HOLREG	0x00000008
1.MW089.XQ01	Motor 3 - Zadati ugao 109	HOLREG	0x00000008
1.MW090.XQ01	Motor 4 - Zadati ugao 110	HOLREG	0x00000008
1.MW091.XQ01	Motor 5 - Zadati ugao 111	HOLREG	0x00000008
1.MW092.XQ01	Motor 6 - Zadati ugao 112	HOLREG	0x00000008
1.MW093.XQ01	Motor 7 - Zadati ugao 113	HOLREG	0x00000008

Успостављање комуникације:

```
private void btn_Connect_Click(object sender, EventArgs e)
{
    try
    {
        if (modbusClient.Connected)
            modbusClient.Disconnect();
    }
}
```

```

modbusClient.IPAddress = txt_ServerTCPAdresa.Text;
modbusClient.Port = int.Parse(txt_ServerPort.Text);
modbusClient.SerialPort = null;
modbusClient.Connect();
rtb_History.Text += "Uspostavljena komunikacija sa serverom" + Environment.NewLine;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Problem prilikom povezivanja sa serverom!");
}
}

```

Прекидање комуникације:

```

private void btn_Disconnect_Click(object sender, EventArgs e)
{
    try
    {
        if (modbusClient.Connected)
            modbusClient.Disconnect();
        rtb_History.Text += "Prekinuta komunikacija sa serverom" + Environment.NewLine;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Problem komunikacije sa serverom!");
    }
}
}

```

## Биографски подаци кандидата

Кандидат, Бошко П. Цветковић, је дипломирани инжењер машинства и студент Докторских студија на Машинском факултету Универзитета у Београду. Рођен је у Београду, 30. 03. 1976. Завршио је Математичку гимназију у Београду. Основне дипломске студије завршио је 2008. године по старом наставном плану и програму. У току докторских студија све испите положио са оценом 10. У оквиру Докторских студија на предмету „Механика робота“ више пута је са студентима основних и мастер студија изводио лабораторијске вежбе, у оквиру којих је рађено на развојним плочама Arduino и већем броју сензора, као и на LEGO роботу. До 2021. године био је запослен у Немачкој компанији „Stucke Elektronik GmbH“ где је радио на пословима програмирања у више програмских језика (C#, C, C++, PHP, MySQL, HTML, Javascript, Java) под Windows оперативним системом и на развоју PLC уређаја и програмирања истих у програмском језику Keil C. Развијао је софтверска решења за компаније Alstom, Hyundai, Eaton. Од 2012. до 2014. године радио је на позицији консултанта на пројекту ДИЛС Министарства рада и социјалне политике Владе Србије уз подршку Светске банке. Сврха пројекта је била пружање унапређених услуга на локалном новоу. Задужење је било техничка супервизија имплементације решења на пројекту и учествовање на састанцима одбора за имплементацију решења. Од 2021. запослен је у Шведској програмерској компанији ЕРТИ на позицији VPE (Vice President of Engineering). Одговоран је за више пројеката и тимова који раде на тим пројектима. Одлично говори енглески језик и служи се немачким језиком. Члан је Менсе Србија.

**Прилог 1.**

**Изјава о ауторству**

Потписани Бошко П. Цветковић \_\_\_\_\_  
број индекса Д43/10 \_\_\_\_\_

**Изјављујем**

да је докторска дисертација под насловом

**Пројектовање савремених система управљања робота применом развојних програмабилних система и савремене теорије рачуна нецелог реда.**

- 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 ауторска права и користио интелектуалну својину других лица.

**Потпис докторанда**

У Београду, \_\_\_\_\_.2021.

\_\_\_\_\_

## Прилог 2.

### Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора \_\_\_Бошко П. Цветковић\_\_\_\_\_

Број индекса \_\_\_Д43/10\_\_\_\_\_

Студијски програм \_\_\_Докторске академске студије\_\_\_\_\_

Наслов рада **Пројектовање савремених система управљања робота применом  
развијених програмабилних система и савремене теорије рачуна нецелог реда.**

Ментор \_\_\_ проф. др Михаило Лазаревић\_\_\_\_\_

Потписани \_\_\_ Бошко П. Цветковић\_\_\_\_\_

Изјављујем да је штампана верзија мог докторског рада истоветна електронској  
верзији коју сам предао 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, \_\_.\_\_.2021.

---

### Прилог 3.

## Изјава о коришћењу

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

**Пројектовање савремених система управљања робота применом развојних програмабилних система и савремене теорије рачуна нецелог реда.**

која је моје ауторско дело.

Дисертацију са свим прилозима предао 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.

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, \_\_.\_\_.2021.

---

1. Ауторство - Дозвољаваате 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. Дозвољаваате 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. Дозвољаваате 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. Дозвољаваате 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. Дозвољаваате 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. Дозвољаваате 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.