

**МЕРНО-УПРАВЉАЧКИ PID СИСТЕМ СА ЈЕДНОСМЕРНИМ МОТОРОМ****MEASUREMENT AND CONTROL PID SYSTEM WITH DC MOTOR**

Милица Митровић, Факултет техничких наука, Нови Сад

**Област— ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО**  
**Кратак садржај** – Тема овог рада јесте контрола брзине једносмерног мотора применом управљачког алгоритма заснованог на ПИД регулатору. У експерименталном делу је као објекат управљања коришћен Graupner Speed 400 мотор. Анализом перформанси објекта управљања при задавању брзине путем микроконтролера, примећена је нестабилност очитане вредности брзине због чега је и било потребно обезбедити PID регулацију представљену у раду. Како се једносмерни мотор чијом се брзином управља понаша као систем првог реда, у раду је представљен поступак пројектовања ПИД регулатора. Сви резултати су потврђени првобитно на нивоу симулације, а потом и експериментално.

**Кључне речи:** PID регулација, Labview, Arduino, DC motor, mikrokontroler

**Abstract** – In this paper the problem of DC motor speed regulation is discussed through usage of PID system regulation. In the experimental part, the Graupner Speed 400 engine was used as the control object. By analyzing the performance of the control object when setting the speed via the microcontroller, the instability of the read speed value was observed, which is why it was necessary to provide the PID regulation presented in the paper. Since the DC motor, the speed of which is controlled, behaves as a first-order system, the design procedure of the PI controller is presented in the paper. All results were confirmed initially at the level of simulation, and then experimentally.

**Keywords:** PID regulation, Labview, Arduino, DC motor, microcontroller

**1. УВОД**

Пројектовање управљачког алгоритма представља важну инжењерску дисциплину која има примену у свим гранама индустрије и технологије. Концепт управљања у затвореној повратној спрези представља процес у коме се посматрани систем од интереса (објекат управљања) доводи у жељено стање посредством сензора, регулатора и актуатора који образују управљачку петљу чији је задатак да надзире понашање система поређењем задате и измерене вредности управљачке величине и на основу тога врши корективне акције уколико има потребе. Оваква структура се у теорији аутоматског управљања назива основно коло система аутоматског управ-

љања [1]. Услед комплексности и широке примене управљачких алгоритама, данас су развијени многобројни алгоритми за контролу па је овај рад посвећен само једном приступу у процесу пројектовања управљачког алгоритма. Посматрани објекат управљања је једносмерни мотор, а управљачки алгоритам, заснован на ПИД регулатору, треба да омогући контролу брзине поменутог система. Поред теоријске анализе система, описана је и спрега софтверских окружења који омогућавају примену целокупне математичке анализе на конкретну хардверску поставку. На тај начин креирано је окружење које омогућава комуникацију између рачунара, микроконтролера и једносмерног мотора.

**2. PID РЕГУЛАЦИЈА**

ПИД регулатори представљају најзаступљенију врсту регулатора у индустријским погонима где је преко 90% управљачких петљи реализовано на основу њих [3][4]. Оваквој свеопштој популарности поменуте технике управљања доприноси једноставна структура, солидан квалитет регулације и могућност подешавања мањег броја параметара. Сам назив ПИД регулатора потиче од три дејства која овај регулатор поседује, а могу се комбиновати на различите начине. Прво дејство је пропорционално дејство (П), друго дејство је интегрално дејство (И) и треће дејство је диференцијално дејство (Д). Закон управљања ПИД регулатора се дефинише следећим изразом

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (1)$$

где је  $t$  временска променљива,  $u$  управљачки сигнал,  $e$  сигнал грешке, а параметри регулатора

$K_p$ ,  $K_i$  и  $K_d$  представљају константу појачања пропорционалног дејства, константу појачања интегралног дејства и константу појачања диференцијалног дејства, респективно [2][5]. Услед комплексности анализе у временском домену, примењује се Лапласова трансформација као математички апарат који омогућава прелазак из континуалног временског домена у комплексни домен у коме се систем описује функцијом преноса. У складу са тим, функција преноса ПИД регулатора се дефинише као

$$G(s) = \frac{K_p s + K_i + K_d s^2}{s}, \quad (2)$$

где је  $s$  Лапласова комплексна променљива [1][6].

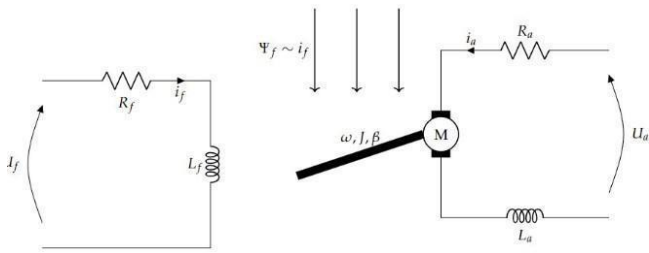
**3. МОТОРИ ЈЕДНОСМЕРНЕ СТРУЈЕ**

На слици 1 приказан је шематски приказ мотора једносмерне струје на основу кога се изводи математички модел система који се састоји од

**Напомена:**

Овај рад проистекао је из мастер рада чији ментор је био др Платон Совиљ, ред. проф.

математичких релација (једначина) које описују понашање система и омогућавају лакшу анализу његових перформанси.



Слика 1. Шематски приказ једносмерног мотора

Основни делови машине су ротор (покретни део) и статор (непокретни део). Величине које описују статорско коло у индексу имају  $f$ , док величине које описују роторско коло имају у индексу  $a$ . Принцип рада мотора једносмерне струје се заснива на електромагнетној индукцији, а комбиновањем основних једначина напонске и механичке равнотеже може се доћи до математичког модела система

$$\omega: -\frac{1}{T_s} \omega + \frac{K_{sm}}{T_s} U_a, \quad y = \omega, \quad (3)$$

где је  $\omega$  угаона брзина вратила машине,  $U_a$  је напон ротора, док су  $K_{sm}$  и  $T_s$  константе мотора. Прва једначина у (3) се назива једначина динамике стања, док се друга једначина у (3) назива једначина излаза. Сходно томе добијен је математички модел система првог реда. Еквивалентно као у претходном делу, применом Лапласове трансформације могуће је одредити функцију преноса једносмерног мотора у случају управљања његовом брзином

$$G(s) = \frac{K}{sT_d + 1}, \quad (4)$$

где је  $K$  статичко појачање система, а  $T_d$  је временска константа система [1][4].

### 3.1. ИДЕНТИФИКАЦИЈА ПАРАМЕТАРА ФУНКЦИЈЕ ПРЕНОСА ЈЕДНОСМЕРНОГ МОТОРА

Иницијални корак у поступку пројектовања управљачког алгоритма је идентификација објекта управљања, односно одређивање вредности непознатих параметара  $K$  и  $T_d$  који фигуришу у функцији преноса система (4). Данас постоје многобројни напредни алгоритми за идентификацију параметара система [7], али услед једноставности посматраног система првог реда, естимација је извршена директним задавањем улазног сигнала и мерењем одређених параметара на излазу. Поменути параметри су време успона система и вредност излазног сигнала у устаљеном стању, а на основу њих су, преко унапред познатих релација из теорије управљања, одређене вредности параметара из функције преноса система:

Табела 1. Естимирани вредности параметара функције преноса

$K$	$T_d$
2	0,37

## 4. ХАРДВЕРСКО ОКРУЖЕЊЕ

За имплементацију целокупног система коришћене су следеће хардверске компоненте: једносмерни мотор Graupner Speed 400, развојна плоча Arduino Uno, интегрисано коло L298N и енкодер.

### 4.1. МОТОР GRAUPNER SPEED 400

Мотори једносмерне струје су уређаји који претварају електричну енергију у механичку. Углавном раде на принципу електромагнетне индукције.

Главна предност једносмерних мотора у односу на оне који се напајају наизменичним величинама је лакше управљање брзином, моментом и смером обртања. DC мотори се углавном праве за мање снаге, јер би за веће снаге они морали да имају веће и јаче изворе једносмерне струје.

У овом раду коришћен је Graupner Speed 400 мотор који има веома ниску брзину обртања, али са друге стране поседује велики обртни моменат. Троши много мање струје и има знатне предности у погледу загревања па представља одличну замену за моторе са четкицама.

### 4.2. РАЗВОЈНА ПЛОЧА ARDUINO UNO

На Arduino Uno развојној плочи налази се микроконтролер Atmega328P. Поред тога, Arduino Uno има 14 дигиталних улазно/излазних (I/O) пинова, од којих 6 има могућност да дају PWM (енгл. *Pulse Width Modulation*) сигнал. Развојну плочу је могуће проширити додатним хардверским модулима који се непрестано развијају, како од стране компаније која их прави, тако и од самих корисника. Ако се узме у обзир да је плоча развијена као *open source* платформа, њен додатни развој од стране корисника је практично неограничен, па се и налази имплементирана у много апликација.

### 4.3. ИНТЕГРИСАНО КОЛО L298N

За покретање мотора једносмерне струје није довољно да се његови крајеви доведу на један од излазних пинова Arduino Uno система. Наиме, мотори једносмерне струје имају захтеве за струјом и напоном који су изван могућности микроконтролера или микрорачунара. Због тога је неопходно користити неку спољну електронику за погон и контролу мотора. Постоји неколико начина за покретање мотора једносмерне струје са излаза рачунарског уређаја. Један од њих је да применом транзистора и ово решење добро функционише под условом да не постоји потреба мењања смера обртања мотора. Други начин контроле мотора једносмерне струје је коришћење кола које се зове Х-мост. Х-мост представља структуру са таквим распоредом транзистора који омогућава контролу смера обртања и брзине мотора.

У овом раду ће бити представљен Х-мост модул заснован на L298N интегрисаном колу.

### 4.4. ЕНКОДЕР

Енкодер је уређај који претвара податке из једног формата у други.

У случају детекције положаја, енкодер је уређај који може детектовати и претворити механичко кретање у

аналогни или дигитални кодирани излазни сигнал. Тачније, он мери положај, док се брзина, убрзање и смер могу извести из положаја линеарним или ротационим кретањем. Класификација енкодера се врши према врсти кретања, те у том случају разликују две врсте: линеарни и ротациони. Код магнетних енкодера, линеарни енкодер је инсталиран изнад магнетне траке, и када се трака помери, енкодер чита пређени пут. Уколико се мери ротација, ротациони енкодер се окреће око своје осе. Оба се користе у електричним моторима јер прате положај ротирајуће основе и мере њен угао.

Према принципу кодирања, енкодери се деле на инкременталне и апсолутне. Мерење помоћу инкременталних енкодера се изводи бројањем импулса са енкодера, док апсолутни енкодери дају бинарни код тренутне позиције [8].

У овом раду кориштен је двофазни оптички ротациони енкодер LPD3806.

## 5. СОФТВЕРСКО ОКРУЖЕЊЕ

### 5.1. ARDUINO IDE

Arduino IDE (enl. *Integrated Development Environment*) представља софтверско окружење које је 'open-source' и намењено је за писање софтвера који се спушта на микроконтролер (Arduino Uno). Ова апликација је доступна за различите оперативне системе као што су Windows, Mac OS X и Linux, а подржава различите програмске језике као што су C и C++. Друга важна чињеница приликом адаптације софтверског са хардверским окружењем је и фреквенција импулса који се доводе на улаз микроконтролера. Наиме, ради се о сигналу велике брзине, а како је брзина којом ради микроконтролер физички ограничена његовим осцилатором, процесорско време потребно за извршавање читавог кода мора да буде што је мање могуће. Због тога се имплементира у коду интерапт рутина (enl. *Interrupt Service Routine*) која ће само узимати узлазне ивице импулсног сигнала доведеног на улаз микроконтролера. Рутина итерира бројач и то се дешава док вредност бројача не постане 100, након чега програм израчунава потребно време. На основу тог времена може се доћи до информације о угаоној брзини, а такође и до линеарне брзине којом би мотор радио. Ова брзина зависи од пречника 'точка'. У овом раду се мери пречник осовине мотора. Програм даље серијском комуникацијом шаље податке на COM (enl. *Computer serial port*) порт рачунара брзином од 115 200 bps. Описани поступак је имплементиран у Arduino IDE софтверском окружењу и кориштен је за мерење брзине једносмерног мотора.

### 5.2. LabVIEW

Даља обрада података извршебна је у Labview програмском окружењу.

Labview програм је написан тако да омогући једноставан приказ података примљених од микроконтролера путем серијске комуникације. Због начина на који хардвер комуницира са Labview фронтендом програм је потребно написати каскадно из два дела, тако да први део програма иницијализује комуникацију, док други у бесконачној петљи

константно врши жељену обраду података, све док га корисник не заустави.

### 5.3. ИНИЦИЈАЛИЗАЦИЈА КОМУНИКАЦИЈЕ

Како је претходно напоменуто потребно је користити каскадну структуру унутар Labview-a, односно *flat sequence*. Реч је о структури која се понаша као слика унутар филма (што и објашњава њен изглед). Свака нова слика (део кода) не може да се прикаже (изврши) пре оне која јој је претходила.

Унутар прве структуре програм почиње са постављањем параметара за Visa серијску комуникацију. Прво што је потребно подесити је избор порта на коме се врши комуникација. Уколико су драјвери за уређај инсталирани на исправан начин и постоји подршка за Visa комуникацију истих, они не морају да се поново подешавају јер ће програм сам изабрати отворен Visa порт. У програму је ипак додата опција за ручно бирање порта. Следећа битна опција коју је потребно наместити је брзина комуникације на одабраном порту.

Како је претходно напоменуто да се подаци са микроконтролера (Arduino) шаљу брзином од 115.200 bps исту брзину потребно је поставити и у Labview програмском окружењу.

Последња опција која је остављена да се ручно унесе је број битова податка. У овом случају то је 8 битова што одговара броју битова које микроконтролер шаље.

Следеће на шта би се требало обратити пажња унутар кода је иницијализација рада са датотекама коју омогућава Labview. Пошто ће се прочитани подаци о брзини тренутно приказивати (у виду графика, и нумеричког индикатора) потребно их је и сачувати да би се њима касније приступило.

У ове сврхе користи се опција *Open/create/replace file* која омогућава креирање датотеке у локалној меморији рачунара. Да би се то успешно одрадило потребно је предефинисати неколико опција. Прва је путања на којој ће се чувати датотека, као и подразумевани назив саме датотеке. Поред тога, кориснику је остављена могућност додавања имена и екстензије датотеке. Због своје универзалности, *CSV (comma-separated values file)* је одабрана као екстензија.

Након креирања датотеке иницијално се испишују први подаци у њу. Исписивање се врши следећим редоследом: број мерења, брзина, време.

Ови стрингови су раздвојени зарезом, док се на последњем месту налази карактер за нови ред. Овако се постиже формирање потребно за исправно чување CSV датотеке у којој ће бити уписана информација редног броја читавања, времена читавања као и брзине која је прочитана.

### 5.4. КОНТРОЛА И ОБРАДА ПОДАТАКА

Након што су извршена претходна иницијална подешавања неопходна за реализацију замишљеног програма, прелази се на следећу каскаду.

Први део унутар ње је отварање претходно креиране датотеке. Након тога, отвара се бесконачна петља која се извршава све док корисник не кликне на дугме *stop* постављено у фронт панелу. Први део програма унутар петље је приступ битовима података који се

шаљу серијсом комуникацијом. Да би се учитали, користи се *Visa read* функција. Она преузима податке као стринг након чега се они конвертују у нумерички тип података. Подаци се конвертују у нумерички тип да би се могли приказивати као нумерички индикатор и због лакшег приказа на графику.

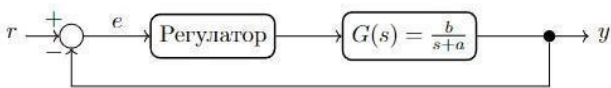
Када пристигне податак читава се и тренутно време као и број итерације понављајуће петље. Ови подаци се прослеђују путем функције *Concatenate string* по обрасцу који је претходно успостављен приликом уписа иницијалних података.

Након тога се задаје брзина мотору путем *Visa write* функције при чему се на скали одређује брзина од 0 cm/sec до 360 cm/sec и иста се конвертује у од 0 до 255 што представља фактор испуне PWM-а на излазном пину (израчунат као  $2^8$ ).

Као што је претходно напоменуто овај поступак се понавља све док га корисник не заустави притиском на дугме након чега се излази из бесконачне петље, датотека се затвара и програм се зауставља.

## 6. ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Као што је раније објашњено у поглављу 6.4 иницијални корак је идентификација објекта управљања, односно одређивање непознатих параметара у функцији преноса. Након завршетка идентификације, могуће је прећи на пројектовање управљачког алгорита, односно избор регулатора и одређивање његових параметара. За почетак, на слици 2 је приказана поједностављена структура основног кола система аутоматског управљања.



Слика 2. Поједностављени приказ управљачке петље

Приликом избора регулатора постоје два основна услова која треба да буду испуњена - да је систем стабилан у затвореној спреси и да има нулту грешку у устаљеном стању. Како посматрани објекат управљања представља систем првог реда без астатизма, закључује да је потребно имати бар један астатизам у директној грани регулационе петље како би се постигла нулта грешка у устаљеном стању. Добијени закључак имплицира да је неопходно изабрати регулатор са интегралним дејством (које уноси астатизам) па је сходно томе изабран најједноствнији регулатор који поседује интегрално дејство - ПИ регулатор.

Што се тиче другог услова везаног за стабилност система у негативној затвореној спреси он је уско повезан са вредношћу параметара регулатора, односно стабилност је експлицитно одређена вредношћу параметара регулатора. За рачунање вредности параметара регулатора се користе многобројне методе, а једна од њих, која ће бити приказана у наставку, је методе подешавања полова. Првобитно је потребно одредити карактеристични полином система у негативној затвореној спреси као именилац функције спрегнутог преноса (функције затвореног кола)

$$f(s) = s^2 + s(a + bKp) + bKi, \quad (5)$$

Непознати параметри регулатора  $Kp$  и  $Ki$  се могу одредити методом подешавања полова тако што се коефицијенти карактеричног полинома система изједначе са коефицијентима ткз. жељеног карактеристичног полинома који ће директно одредити понашање система у затвореној спреси. У овом раду изабрано је да сви полови система буду у тачки  $-2$  што имплицира да је систем стабилан у затвореној спреси чиме је испуњен и други услов при пројектовању регулатора. Следи да је жељени карактеристични полином

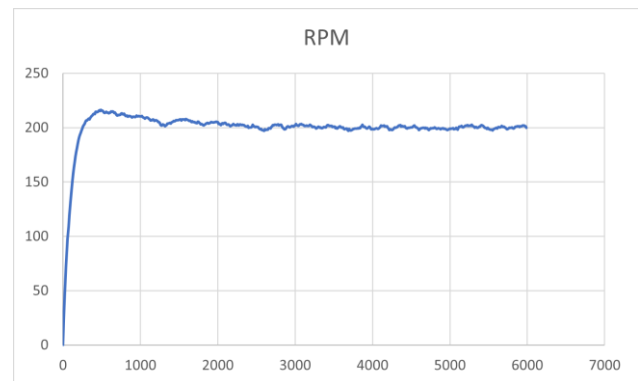
$$f_z(s) = (s + 2)^2 = s^2 + 4s + 4 \quad (6)$$

па се вредности параметара ПИ регулатора добијају изједначавањем коефицијената полинома из (5) и (6).

$Kp$	$Ki$
0,241	0,741

Табела 2. Вредности параметара регулатора

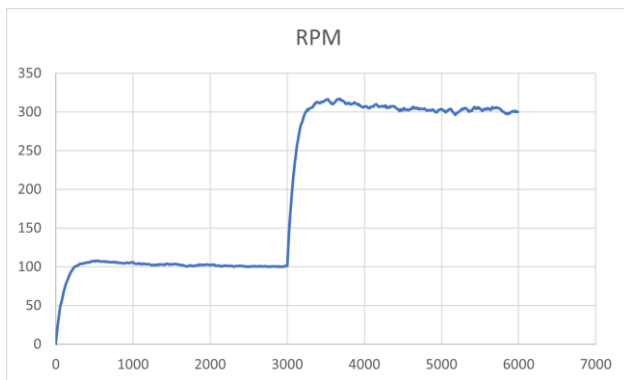
Добијене вредности су примењене на конкретан систем користећи претходно описано софтверско окружење. Извршено је више мерења, а резултати су приказани графички. За сваки експериментални поступак задата је референта вредност брзине, док је снимана излазна вредност, односно постигнута брзина приказана на графицима у наставку. Постигнута брзина једносмерног мотора, приказана на ординатама графика у наставку, је изражена у јединицама обртаји у минути - rpm, док је време, приказано на апсисама, изражено у милисекундама- ms.



Слика 3. Графички приказ остварене брзине, референта вредност брзине је 200 rpm

Приказани су примери одзива система на различите вредности побудног сигнала, са скоковитим променама вредности како би се анализирао брзина одзива система која је на задовољавајућем нивоу. Такође, постигнута је успешна регулација, односно праћење задате вредности брзине.

У самом одзивном сигналу су приметне благе осцилације које су последица несавршености лабораторијске хардверске поставке.



Слика 4. Графички приказ остварене брзине при скоковитој промени референтне вредности брзине са 100 rpm на 300 rpm

## 7. ЗАКЉУЧАК

У овом раду је приказан поступак пројектовања управљачког алгорита чији је крајњи циљ контрола брзине обртања мотора једносмерне струје. Наиме, за посматрани објекат управљања је прво изведен математички модел у простору стања на основу чега је закључено да се приликом управљања брзином једносмерног мотора добија систем првог реда. На основу математичког модела је изведена функција преноса система у којој фигуришу два непозната параметра, статичко појачање и доминантна временска константа. Ови параметри имају и своју физичку интерпретацију, показују колико систем појачава сигнал са улаза и колико брзо систем улази у устаљено стање, респективно.

Да би се одредиле вредности ових параметара, односно да би се извршила идентификација функције преноса, спроведен је експеримент у коме је на улаз система доведен константан сигнал, а на излазу је снимљен одзив, односно његова вредност у устаљеном стању. Такође, на основу излазног сигнала је одређено и време успона на основу кога је одређена доминантна временска константа система која је даље једнозначно одредила и статичко појачање система. На тај начин је објекат управљања у потпуности идентификован.

Након потпуног одређивања функције преноса објекта управљања, уследило је пројектовање управљачке петље, односно избор регулатора и одређивање вредности параметара регулатора применом методе подешавања полова.

Структура регулатора је изабрана тако да вредност грешке у устаљеном стању на константан улазни сигнал, а са објектом управљања првог реда без астатизма, буде једнака нули. Најједноставнија структура регулатора која задовољава наведени критеријум је ПИИ регулатор. Оваква врста регулатора се састоји из пропорционалног и интегралног дејства, односно у структури регулатора фигуришу појачање пропорционалног и појачање интегралног дејства које је потребно одредити.

Непознати параметри регулатора су израчунати на основу методе подешавања полова у којој се врши директно изједначавање карактеристичног полинома система и жељеног карактеристичног полинома чији облик одређује понашање система у затвореној спреси (стабилност, брзина одзива итд.).

Добијени параметри регулатора, односно пројектовани управљачки алгоритам је примењен на објекат управљања при чему су добијени резултати снимљени и графички приказани. Упркос несавршеностима хардверске поставке, добијени резултати су задовољавајући што је потврђено и графичким резултатима.

## 8. ЛИТЕРАТУРА

- [1] Милан Р. Рапаић, Зоран Д. Јеличић. *Пројектовање линеарних регулатора и естиматора у простору стања*. Универзитет у Новом Саду, Факултет техничких наука, Нови Сад, 2014.
- [2] Karl J. Astrom, Bjorn Wittenmark. *Computer-Controlled Systems Theory and Design*. Dover Publications Inc. 3rd edition, New York, USA, 2011.
- [3] Милић Р. Стојић. *Системи аутоматског управљања*. 2006.
- [4] Зоран Д. Јеличић, Мирна Н. Капетина, Милан Р. Рапаић. *Дискретни управљачки системи*. Универзитет у Новом Саду, Факултет техничких наука, Нови Сад, 2023.
- [5] Karl J. Astrom, Tore Hagglund. *PID Controllers*. Instrumentation Society of America, 1995.
- [6] Милић Р. Стојић. *Дигитални системи управљања*. Грађевинска књига, Београд, 2006.
- [7] Александар Ердџан, Дарко Чапко. *Моделовање и симулација са примерима*. Универзитет у Новом Саду, Факултет техничких наука, Нови Сад, 2015.
- [8] W.Bolton, System models, Editor(s): W.Bolton, *Control Systems*, Newnes, 2002, Pages 37-70, ISBN9780750654616, <https://doi.org/10.1016/B978-075065461-6/50002-7>.

### Кратка биографија:



**Милица Митровић** рођена је у Зворнику 1997. год. Мастер рад на Факултету техничких наука из области Електротехнике и рачунарства - Мерно-управљачки PID систем са једносмерним мотором одбранила је 2024. год.

Контакт: milica.mitroviczv@gmail.com