

# Automatsko upravljanje položaja loptice na gredi

---

Škrobar, David

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:218423>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-22**



Repository / Repozitorij:

[University North Digital Repository](#)





# Sveučilište Sjever

Završni rad br. 465/EL/2020

## Automatsko upravljanje položaja loptice na gredi

David Škrobar, 1446/336

Varaždin, rujan 2020. godine





# Sveučilište Sjever

Odjel za elektrotehniku

Završni rad br. 465/EL/2020

## Automatsko upravljanje položaja loptice na gredi

### Student

David Škrobar, 1446/336

### Mentor

Miroslav Horvatić, dipl.ing.

Varaždin, rujan 2020. godine

# Prijava završnog rada

## Definiranje teme završnog rada i povjerenstva

ODJEL Odjel za elektrotehniku

STUDIJ preddiplomski stručni studij Elektrotehnika

PRISTUPNIK David Škrobar

MATIČNI BROJ 1446/336

DATUM 29.06.2020.

KOLEGIJ Automatsko upravljanje

NASLOV RADA Automatsko upravljanje položaja loptice na gredi

NASLOV RADA NA ENGL. JEZIKU Automatic control of the Ball & Beam system

MENTOR Miroslav Horvatić, dipl. ing.

ZVANJE predavač

ČLANOVI POVJERENSTVA

1. mr. sc. Ivan Šumiga, dipl. ing., viši predavač
2. mr. sc. Matija Mikac, dipl. ing., viši predavač
3. Miroslav Horvatić, dipl. ing., predavač
4. dr. sc. Dunja Srpak, docent, rezervni član
- 5.

## Zadatak završnog rada

BROJ 465/EL/2020

OPIS

Potrebno je realizirati maketu sustava za automatsko upravljanje položaja loptice na gredi. Upravljanje realizirati korištenjem PID regulatora i Arduino platforme.

U radu je potrebno:

- osmisлити sklopovski i programski dio sustava
- realizirati sklopovski i programski dio sustava
- prikazati matematički model sustava i regulatora
- realizirati automatsko upravljanje položaja loptice na gredi
- ispitati i analizirati ponašanje sustava.

ZADATAK URUČEN

29.07.2020.

POTPIS MENTORA



## Sažetak

Prilikom izrade ovog završnog rada bilo je nužno poznavanje raznih cjelina iz područja automatskog upravljanja. Glavna ideja ovog rada je modeliranje sustava loptice na gredi, te programska implementacija PID regulatora kako bi sustav pozicionirao lopticu na referentnom zadanom položaju. Prikazana je izrada makete sustava, te su opisani drugi načini izrade makete korištenjem raznih drugih mogućih elemenata.

U radu je objašnjen matematički model sustava. Matematički model sustava nije linearan, stoga je u radu objašnjena linearizacija modela, a objašnjena je i linearizacija nelinearne statičke karakteristike senzora.

Objašnjeni su i svi dijelovi cjelokupnog sustava. Najvažniji dio sustava čini Arduino platforma. Arduino mikrokontroler prima informaciju o položaju loptice u obliku napona koji šalje infracrveni senzor, te ga uspoređuje s referentnim položajem. PID algoritam u programskom kodu zatim koristi razliku tih signala, kako bi na servomotor dao upravljački signal za regulaciju položaja loptice na gredi. U radu se objašnjava programski kod te utjecaj pojedinih dijelova koda na regulaciju sustava. Programski paket MATLAB korišten je za simulaciju i analizu ponašanja sustava. Na kraju rada rezultati se objašnjavaju i analiziraju, te se donosi zaključak.

**Ključne riječi:** automatsko upravljanje, PID regulacija, Arduino, loptica na gredi, senzor

## **Abstract**

During the creation of this final thesis, it was necessary to know various topics in the field of automatic control. The main idea of this thesis is to model the ball and beam system, and the implementation of a PID controller through Arduino program, so that the system positions the ball at the reference set point. The creation of a physical model of the system is described, as well as other ways of making a model using various other possible elements.

This thesis explains the mathematical model of the system. The mathematical model of the system is nonlinear, therefore the linearization of the model is explained.

All parts of the whole system are also explained. The most important part of the system is the Arduino platform. The Arduino microcontroller receives the information about the position of the ball from the IR sensor in the form of voltage, and compares it with the reference position. The PID algorithm in the Arduino program code then uses the difference of these signals, to give the servo motor a control signal to control the position of the ball on the beam. The thesis explains the program code and the influence of individual parts of the code on the regulation of the system. The MATLAB programming environment was used for simulation and analysis of system behavior. At the end of the thesis, the results are explained and analyzed, and a conclusion is made.

**Keywords:** automatic control, PID controller, Arduino, ball and beam, sensor

## **Popis korištenih kratica**

|            |                                    |
|------------|------------------------------------|
| <b>IR</b>  | Infrared                           |
| <b>LED</b> | Light emitting diode               |
| <b>IDE</b> | Integrated Development environment |
| <b>PWM</b> | Pulse width modulation             |
| <b>PID</b> | Proportional-Integral-Derivative   |



# Sadržaj

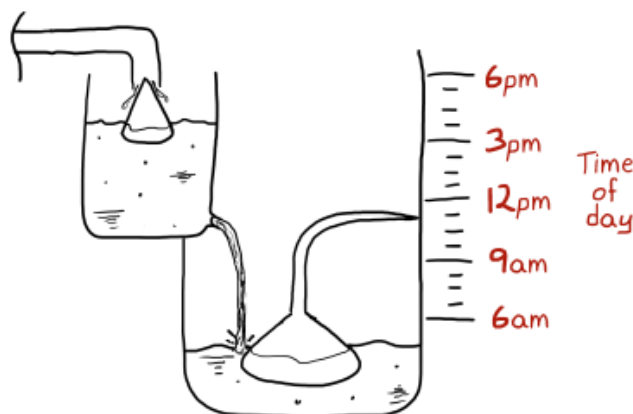
|       |  |    |
|-------|--|----|
| 1.    | Uvod .....   | 1  |
| 2.    | Teoretska analiza i dijelovi sustava upravljanja .....         | 3  |
| 2.1   | Struktura sustava automatskog upravljanja .....                | 3  |
| 2.1.1 | Upravljanje bez povratne veze .....                            | 4  |
| 2.1.2 | Upravljanje s povratnom vezom .....                            | 5  |
| 2.2   | PID regulacija i regulatori .....                              | 6  |
| 2.3   | Arduino platforma .....  | 8  |
| 2.3.1 | Arduino Uno .....  | 9  |
| 3.    | Matematički model .....  | 10 |
| 3.1   | Parametri sustava .....  | 11 |
| 3.2   | Modeliranje sustava .....                                      | 12 |
| 4.    | Praktični dio rada .....                                       | 15 |
| 4.1   | Izrada makete sustava .....                                    | 16 |
| 4.2   | Senzor .....   | 19 |
| 4.2.1 | Linearizacija nelinearne statičke karakteristike senzora ..... | 22 |
| 4.3   | Servomotor .....   | 24 |
| 4.4   | Arduino IDE i programski kod .....                             | 27 |
| 5.    | Analiza rezultata .....  | 32 |
| 6.    | Zaključak .....  | 39 |
| 7.    | Literatura .....   | 41 |
|       | Popis slika .....  | 42 |
|       | Popis tablica .....  | 43 |

# 1. Uvod

Današnji svijet nezamisliv je bez automatskog upravljanja. Razvojem čovječanstva automatsko upravljanje postaje temelj mnogim tehničkim i netehničkim znanostima, pa se stoga u modernim vremenima koriste razni dinamički sustavi automatskog upravljanja kako bi se realiziralo upravljanje nekim objektom ili procesom bez prisutnosti čovjeka. Automatski upravljani sustavi doprinose brzini, kvaliteti, sigurnosti i smanjenom utrošku energije.

Jedna od najvažnijih podjela sustava automatskog upravljanja je podjela na sustave s i bez povratne veze. U praksi su najčešći i najvažniji sustavi s povratnom vezom koji izlaznu veličinu uspoređuju s referentnom. Najveća prednost sustava s povratnom vezom je da imaju mogućnost automatskog smanjenja regulacijske pogreške. Mogućnost da se sustavi automatski prilagode promjenama u okolini odbacila je svaku potrebu da ljudi budu dio upravljanog procesa. Stoga automatsko upravljanje i sustave regulacije nalazimo u raznim situacijama, stvarima i sustavima za koje nismo ni svjesni da su automatski upravljane.

Jedan od prvih primjera sustava s povratnom vezom datira čak dvije tisuće godina prije prve industrijske revolucije, kada je starogrčki izumitelj i otac pneumatike Ktesibije osmislio vodeni sat, čiji je princip mjerenja vremena ostao u upotrebi do 17. stoljeća, dok se taj princip regulacije stalnog protoka vode čak i danas slično koristi u vodokotlićima. [1]



*Slika 1.1 Pojednostavljeni prikaz Ktesibijevog vodenog sata [1]*

Na slici 1.1 prikazan je Ktesibijev vodeni sat, koji funkcionira na način da se u prvoj posudi korištenjem plutajućeg čepa regulira konstanta razina vode, što zatim rezultira konstantnim protokom vode iz prve posude u drugu.

Kako bi se istražile i prikazale neke mogućnosti automatskog upravljanja, za predmet ovog rada odabran je sustav loptice na gredi koji predstavlja jedan izrazito popularan model, na kojem su izvršena brojna istraživanja i testiranja raznih metoda regulacije. Cilj sustava upravljanja položaja loptice na gredi je pozicionirati lopticu na željenoj točki na gredi. Sustav je kompleksan te ga karakterizira dinamičko ponašanje na koje nailazimo u raznim modernim sustavima automatskog upravljanja, poput regulacije položaja krila aviona prilikom turbulencija. Nadalje, sustav je izrazito nelinearan i nestabilan u otvorenom krugu i sadrži relativno složeni matematički model, no s druge strane princip rada sustava lako je razumljiv pa je stoga idealan primjer za edukativne svrhe.

## 2. Teoretska analiza i dijelovi sustava upravljanja

Kako bi pravilno razumjeli tematiku ovog rada, nužno je poznavati osnove automatskog upravljanja, stoga će se u ovom poglavlju opisati osnovna struktura i osnovni elementi sustava automatskog upravljanja, prikazivanje sustava blok dijagramima, te sustavi s povratnom vezom i bez povratne veze. Opisati će se i Arduino platforma te će se objasniti princip rada PID regulatora.

### 2.1 Struktura sustava automatskog upravljanja

Osnovna struktura sustava upravljanja sastoji se od četiri glavna elementa:

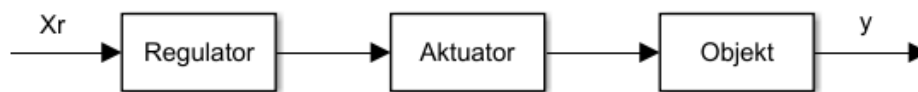
- Proces, odnosno objekt upravljanja
- Regulator
- Aktuator, odnosno izvršni element
- Detektor, odnosno mjerni element

Sustavi se prikazuju blok dijagramima. Blok dijagrami su način grafičkog predstavljanja sustava s fokusom na koncept, jednostavnost i razumijevanje sustava, bez dodavanja nepotrebnih informacija o sustavu. Blok dijagrami prikazuju međupovezanost elemenata sustava i tok signala među njima. U blok dijagramu se za prikaz nekog sustava ili elementa koriste kvadrati ili češće pravokutnici, dok se za prikaz toka signala koriste linije sa strelicom koje se nazivaju signalnim linijama. Ako signalna linija sa strelicom ulazi u neki blok, tada predstavlja ulaznu veličinu, dok signalna linija koja izlazi iz bloka predstavlja izlaznu veličinu. Također, blok dijagrami se koriste za prikazivanje kompleksnih arhitektura sustava i njihovo pojednostavljivanje kako bi se dobila prijenosna funkcija sustava.

Sustavi automatskog upravljanja mogu se podijeliti na sustave upravljane bez povratne veze i sustave upravljane s povratnom vezom, a oni s povratnom vezom dijele se na sustave s pozitivnom povratnom vezom i s negativnom povratnom vezom.

### 2.1.1 Upravljanje bez povratne veze

Sustav bez povratne veze je sustav koji nema mjerni element, stoga ne uspoređuje izlaznu veličinu i referencu. Procesom upravlja regulator koji ima svoj zadani program, u kojem se pojedine operacije odvijaju u određenom vremenskom slijedu. Upravljanje bez povratne veze naziva se još i upravljanje u otvorenoj petlji (eng. open loop). Upravljanje je jednostavno, sustavi imaju jednostavni dizajn, no nemaju mogućnost potpunog uklanjanja djelovanja poremećaja.

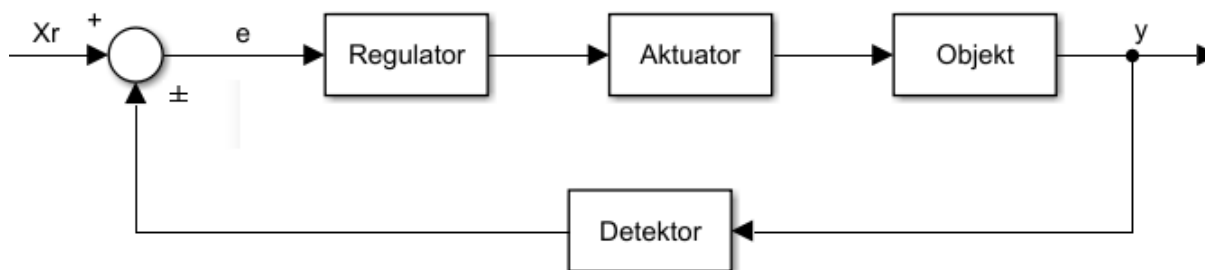


*Slika 2.1 Blok dijagram upravljanja bez povratne veze:  $X_r$  – Referentna veličina;  
 $y$  – Izlazna veličina*

Kao primjer upravljanja bez povratne veze može se uzeti mikrovalna pećnica kojoj je cilj zagrijati neko jelo. Korisnik namjesti željeno vrijeme grijanja jela, no mikrovalna pećnica radi sve do isteka zadanog vremena, bez obzira je li jelo dovoljno zagrijano ili ne.

## 2.1.2 Upravljanje s povratnom vezom

Ovakav sustav upravljanja bazira se na usporedbi izlazne veličine s referentnom. Nužno je mjeriti izlaznu veličinu kako bi se usporedila s referentnom veličinom na ulazu. Takvo upravljanje još se naziva i upravljanje u zatvorenoj petlji (eng. closed loop). Izlazni signal može se zbrojiti ili oduzeti od referentnog. Ako se signali zbrajaju, takva veza naziva se pozitivnom povratnom vezom, a ako se signali oduzimaju, takva veza naziva se negativnom povratnom vezom. Pozitivna povratna veza je nepoželjna za mnoge procese, jer se povratnom vezom pojačava pobuda, koja zatim može stvoriti nestabilnosti i čak dovesti do uništenja procesa. Negativna povratna veza ključna je za sustave automatskog upravljanja, budući da je mnoge procese moguće upravljati samo s negativnom povratnom vezom, stoga je mnogo češća u praksi.



Slika 2.2 Blok dijagram upravljanja s povratnom vezom:  $X_r$  – Referentna veličina;  
 $y$  – Izlazna veličina;  $e$  – Zbroj ili razlika signala

Ako se signali oduzimaju, tada se varijabla  $e$  često naziva i regulacijskim odstupanjem (eng. error). Princip rada sustava s negativnom povratnom vezom je da regulacijsko odstupanje teži nuli, odnosno da ulazna i izlazna veličina budu jednake.

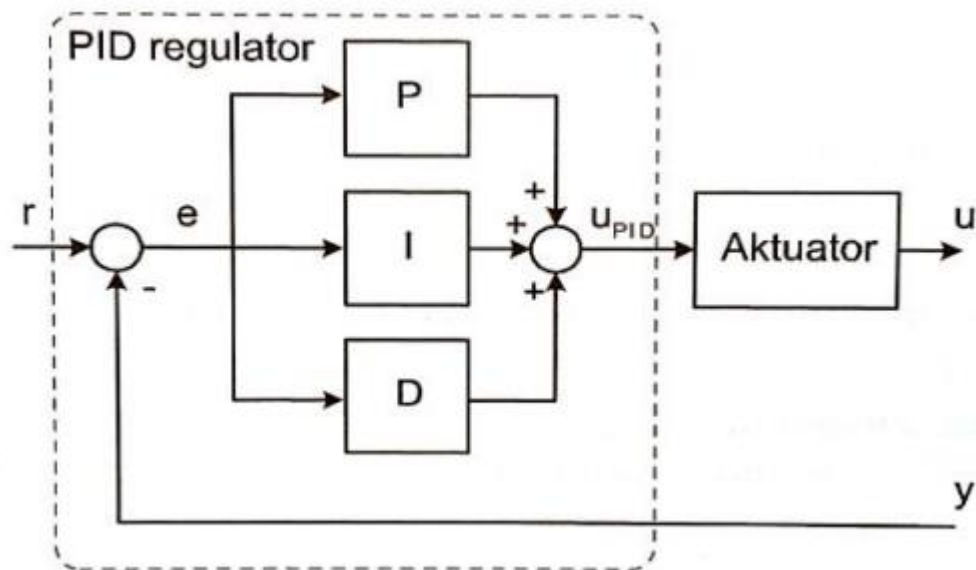
Primjer automatiziranog sustava s povratnom vezom je i sam čovjek, kod kojeg također možemo pronaći sve osnovne elemente sustava regulacije. Objekt upravljanja predstavlja sam čovjek, mozak predstavlja regulator, mišići predstavljaju izvršni element, odnosno aktuator, dok oči ili uši predstavljaju mjerni element, odnosno senzor koji prikuplja informacije i šalje ih mozgu na obradu.

## 2.2 PID regulacija i regulatori

Regulatori su iznimno česta pojava u industrijskim postrojenjima, gdje se koriste kako bi u sustavima s povratnom vezom regulirali neki proces. Najčešće se koriste takozvani PID regulatori. U industriji, PID regulatori se koriste za regulaciju raznih procesnih veličina, kao što su temperatura, protok, tlak, svjetlost, i razne druge veličine. Budući da je ulaz u PID regulator regulacijsko odstupanje, odnosno razlika između željene i stvarne vrijednosti, glavna funkcija tog regulatora je da se regulacijsko odstupanje svede na nulu tijekom nekog vremena, odnosno da izlaz iz sustava bude jednak željenoj vrijednosti. Sustav PID regulatora se sastoji od 3 glavna člana: proporcionalni član (P), integralni član (I) i derivacijski član (D). Svaki od tih članova na različit način pridonosi regulaciji regulatora. Svaki od tih članova može više ili manje pridonositi regulaciji pomoću namještanja PID konstanti. Te konstante se nazivaju  $k_p$ ,  $k_i$  i  $k_d$  konstante. Vrijednost svake od tih konstanti govori koji utjecaj na regulaciju ima svaki od pripadajućih članova. Ako je vrijednost neke konstante 0, tada taj član nema utjecaj na rad regulatora. Primjerice, ako je vrijednost  $k_d$  konstante 0, tada utjecaj D člana ne postoji pa se taj regulator naziva PI regulatorom.

Prijenosna funkcija sustava je Laplaceova transformacija impulsnog odziva linearnog, kontinuiranog i vremenski nepromjenjivog sustava, kada su početni uvjeti postavljeni na nulu. Prijenosna funkcija predstavlja omjer između Laplaceovih transformacija ulaza u sustav  $U$  i izlaza iz sustava  $Y$ . Prijenosna funkcija PID regulatora ima sljedeći oblik:

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



Slika 2.3 Struktura PID regulatora prikazana blok dijagramom [2]

Proporcionalni član je relativno jednostavan, jer izlaznu veličinu sustava regulira proporcionalno regulacijskom odstupanju, odnosno pojačava razliku između ulaza i izlaza iz sustava.

Integracijski član se koristi kako bi se uklonile vrlo male vrijednosti regulacijskog odstupanja prilikom stacionarnog stanja (eng. steady-state error). Ako je regulacijom postignuto stacionarno stanje, no još uvijek postoji regulacijsko odstupanje, tada integralni član integrira regulacijsko odstupanje kroz vrijeme. Budući da je integral broja većeg od 0 uvijek pozitivan, to znači da se izlaz iz integracijskog člana povećava kroz vrijeme, te tako nastoji smanjiti regulacijsko odstupanje.

Derivacijski član predstavlja mjeru promjene regulacijskog odstupanja, odnosno koliko brzo se regulacijsko odstupanje smanjuje ili povećava. Ako se objekt upravljanja brzo približava željenoj vrijednosti, to znači da se regulacijsko odstupanje brzo smanjuje.



## 2.3 Arduino platforma

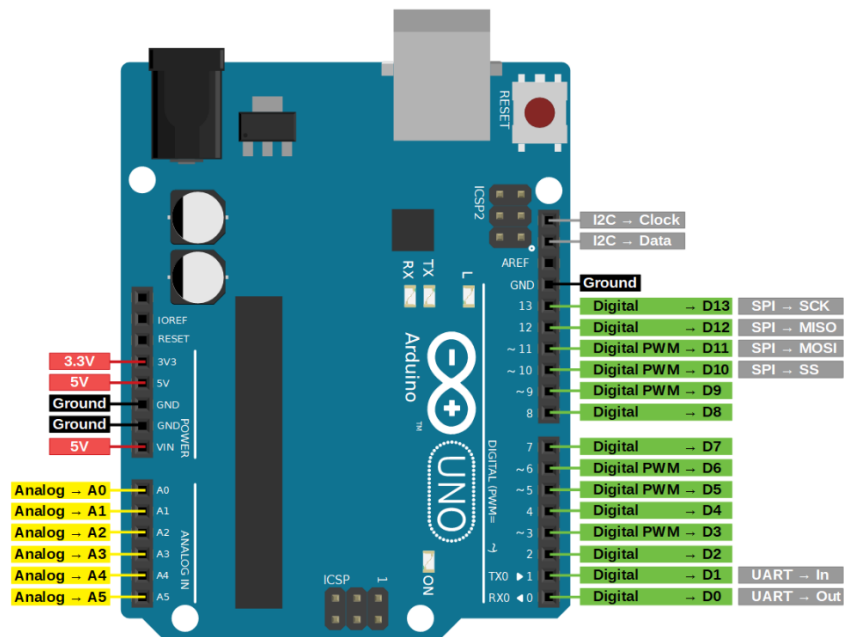
Arduino je mikrokontrolerska platforma otvorenog koda (eng. open source) koja se temelji na jednostavnosti uporabe hardvera i softvera, a služi za izradu raznih elektroničkih projekata. Sastoji se od programabilnog mikrokontrolera (na razvojnoj pločici) i od softverskog razvojnog okruženja. Mikrokontroler izvršava procesorske zadatke, dok se programiranje odvija na računalu korištenjem pojednostavljenog C++ jezika. Iznimnu popularnost temelji na relativno niskoj cijeni programibilnih pločica, jednostavnosti uporabe za početnike, te velikoj podršci na internetu. Arduino razvojno okruženje je iznimno jednostavno za početnike, no dovoljno fleksibilno za iskusnije korisnike. Također, Arduino se može koristiti na Windows, Mac i Linux operacijskim sustavima. Arduino pločice postoje u raznim veličinama, izvedbama i vrstama, a koriste se ovisno o namjeni projekta koji se izrađuje. Jedna od najpopularnijih, najjednostavnijih i najkorištenijih pločica je Arduino Uno, koja je korištena i u ovom radu.



*Slika 2.4 Prikaz Arduino Uno pločice*

### 2.3.1 Arduino Uno

Arduino Uno bazira se na ATmega328P procesoru. Na pločici se nalazi 14 digitalnih ulazno/izlaznih pinova, od kojih se 6 može koristiti kao PWM izlazni pinovi. Također, na pločici se nalazi i 6 analognih ulaza. Mikrokontroler sadrži 32 KB flash memorije, 1 KB EEPROM memorije, te 2 KB SRAM-a. Na slici 2.5 prikazan je popis pinova (eng. pinout) i njihove funkcije.



Slika 2.5 Prikaz pinova i njihovih funkcija na Arduino Uno pločici [3]

U radu je korišten ulazni pin A0 za primanje signala sa senzora, pin 9 kao izlazni za upravljački signal servomotoru, dok je pin reguliranog napajanja od 5V korišten kako bi se napajao senzor.

### 3. Matematički model

Kako bi se realizirala automatizacija za neki proces ili sustav, najprije je potrebno taj isti proces ili sustav opisati efikasno i korisno, tako da se opisivanjem nastoji olakšati razumijevanje problema. No, efikasnost i korisnost opisa mijenja se u ovisnosti o kontekstu problema. U svijetu automatizacije, ključno je proces ili sustav opisati matematički, kako bi se izrazili odnosi između pojedinih komponenti.

U sustavu loptice na gredi, loptica se kotrlja po gredi čiji položaj se korigira servomotorom. Kako bi se loptica pozicionirala u željenoj točki, greda se mora pravilno i pravovremeno okretati oko svoje centralne osi. To nadalje znači da i servomotor mora pravilno i pravovremeno dobivati električne impulse kojima rotira gredu. Stoga je iznimno važno izraditi matematički model sustava, koji matematički prikazuje vladanje sustava. Sustav se matematički može opisati na više načina. Najčešći način matematičkog opisivanja je pomoću diferencijalnih jednadžbi i prijenosnih funkcija. Ako se modelira neki nelinearni sustav, poput sustava loptice na gredi, ponekad je potrebno napraviti aproksimativni opis korištenjem linearnih modela. Također, nužno je napomenuti da nije moguće napraviti savršeni model sustava, neovisno o korištenoj metodi.

### 3.1 Parametri sustava

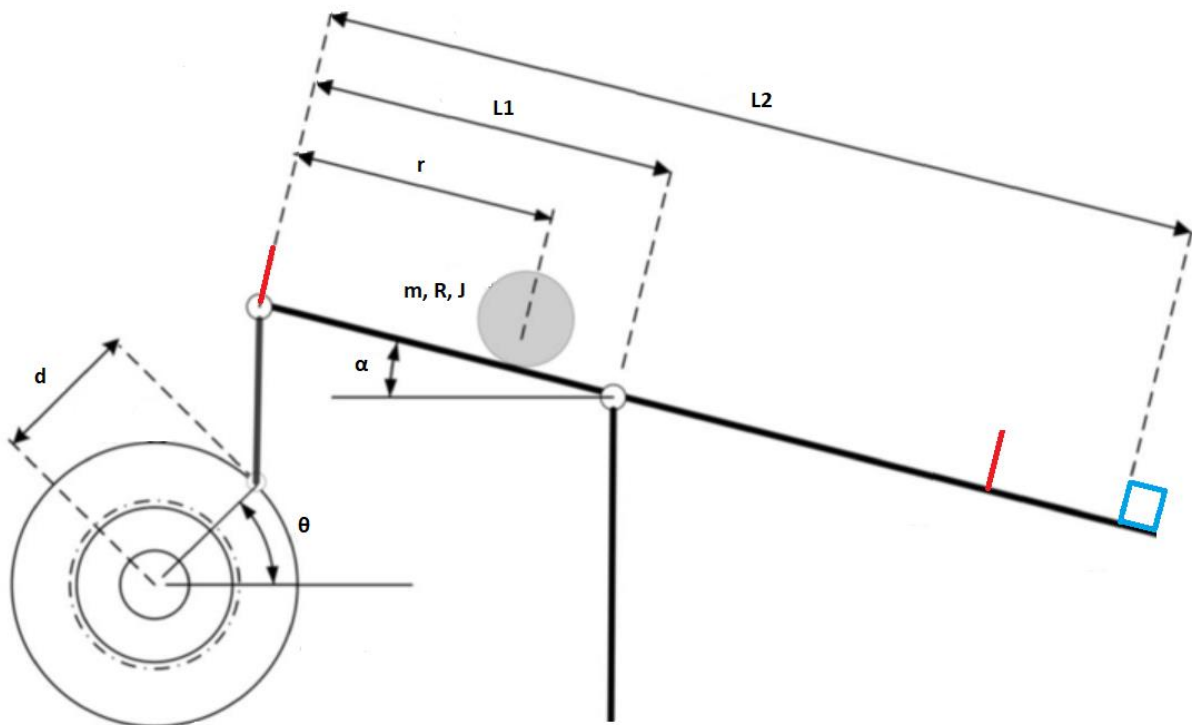
Neki parametri sustava loptice na gredi dobivaju se direktnim mjerenjem ili eksperimentiranjem, dok su neki već određeni kao konstante. Parametri sustava loptice na gredi korišteni u radu, prikazani su u tablici 3.1.

| Redni broj | Parametar                                       | Simbol         | Jedinica          | Vrijednost             |
|------------|---|----------------|-------------------|------------------------|
| 1          | Masa loptice                                    | m              | kg                | 0.0027                 |
| 2          | Radijus loptice                                 | R              | m                 | 0.02                   |
| 3          | Pomak poluge                                    | d              | m                 | 0.03                   |
| 4          | Gravitacijsko ubrzanje                          | g              | m/s <sup>2</sup>  | 9.81                   |
| 5          | Duljina grede od oslonca do krajnjeg graničnika | L <sub>1</sub> | m                 | 0.175                  |
| 6          | Duljina grede od senzora do krajnjeg graničnika | L <sub>2</sub> | m                 | 0.45                   |
| 7          | Moment inercije loptice                         | J              | kg·m <sup>2</sup> | 7.2 x 10 <sup>-7</sup> |
| 8          | Pozicija loptice                                | r              | m                 | -                      |
| 9          | Kut zakreta grede                               | $\alpha$       | rad               | -                      |
| 10         | Kut zakreta servomotora                         | $\theta$       | rad               | -                      |

Tablica 3.1 Prikaz parametara korištenih u matematičkom modelu

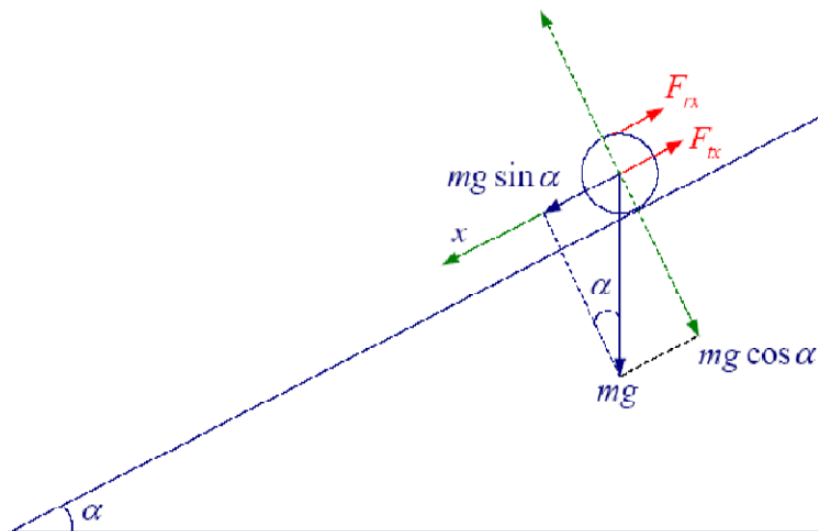
### 3.2 Modeliranje sustava

Prilikom modeliranja sustava loptice na gredi, nužno je koristiti zakone fizike kako bi se dobile jednačbe koje opisuju sustav. Slika 3.1 prikazuje skicu sustava na temelju koje je napravljen matematički model. Graničnici su na slici označeni crvenom bojom, dok je senzor prikazan plavom bojom. Parametri sustava iz tablice 3.1 su također označeni na slici.



Slika 3.1 Skica sustava loptice na gredi

Dinamika loptice podložna je gravitaciji, inerciji, translaciji i centrifugalnim silama. Klizanje i trenje loptice, te kašnjenje motora i senzora su zanemareni.



Slika 3.2 Prikaz loptice na kosini i pripadajućih sila

Sila koja nastaje uslijed translacije loptice je:

$$F_{trl} = m \cdot \frac{d^2r}{dt^2} \quad (3-1)$$

Sila koja nastaje uslijed rotacije loptice je:

$$F_{rot} = \frac{J}{R^2} \cdot \frac{d^2r}{dt^2} \quad (3-2)$$

Formula za moment inercije šuplje sfere iznosi:

$$J = \frac{2}{3}m \cdot R^2 \quad (3-3)$$

Početna nelinearna diferencijalna jednačba koja opisuje ravnotežu sila koje djeluju na lopticu:  
[4]

$$0 = \left(\frac{J}{R^2} + m\right) \cdot \frac{d^2r}{dt^2} + mgsin\alpha - mr \cdot \left(\frac{d\alpha}{dt}\right)^2 \quad (3-4)$$

Posljednji član u jednačbi (3-4) predstavlja centrifugalnu silu koja ovisi o promjenama kuta grede. Uz pretpostavku da su promjene kuta  $\alpha$  jako male, gornja diferencijalna jednačba može se linearizirati oko iznosa kuta  $\alpha = 0$  ( $sin\alpha \approx \alpha$ ), te se tada dobiva:

$$\left(\frac{J}{R^2} + m\right) \cdot \frac{d^2r}{dt^2} = -mg\alpha \quad (3-5)$$

Jednačba koja povezuje kut grede  $\alpha$  i kut motora  $\theta$  može se aproksimirati sljedećom jednačbom:

$$\alpha = \frac{d}{L_1} \cdot \theta \quad (3-6)$$

Uvrštavanjem (3-6) u (3-5) dobiva se:

$$\left(\frac{J}{R^2} + m\right) \cdot \frac{d^2r}{dt^2} = \frac{-mgd\theta}{L_1} \quad (3-7)$$

Prijenosna funkcija se dobiva Laplaceovom transformacijom gornje jednadžbe:

$$\left(\frac{J}{R^2} + m\right) \cdot s^2 R(s) = \frac{-mgd \cdot \theta(s)}{L_1} \quad (3-8)$$

$$P(s) = \frac{R(s)}{\theta(s)} = \frac{-mgd}{L_1 \cdot \left(\frac{J}{R^2} + m\right) \cdot s^2} \quad (3-9)$$

Uvrštavanjem (3-3) u (3-9) dobiva se

$$P(s) = \frac{R(s)}{\theta(s)} = \frac{-mgd}{L_1 \cdot \left(\frac{\frac{2}{3}m \cdot R^2}{R^2} + m\right) \cdot s^2} = \frac{-mgd}{L_1 \cdot \left(\frac{5}{3}m\right) \cdot s^2} \quad (3-10)$$

Završna prijenosna funkcija iznosi:

$$P(s) = \frac{R(s)}{\theta(s)} = \frac{-3gd}{5L_1} \cdot \frac{1}{s^2} \quad \left[ \frac{m}{rad} \right] \quad (3-11)$$

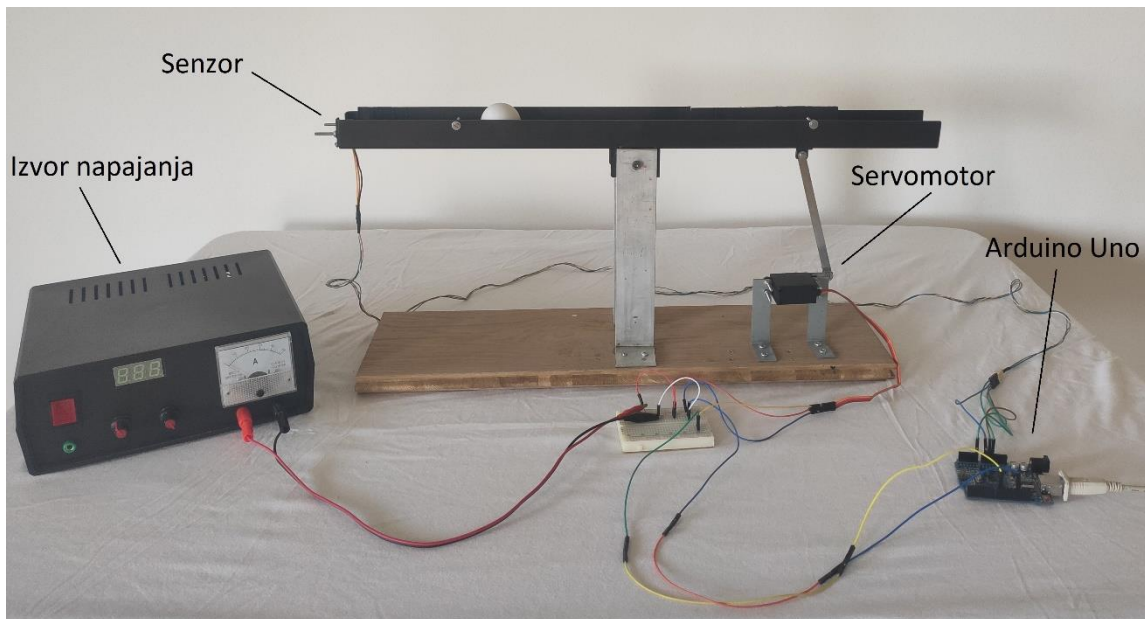
## 4. Praktični dio rada

Sustav loptice na gredi predstavlja jednostavni mehanički sustav, no istovremeno složen za upravljanje. Mehanički sustav se sastoji od grede koja slobodno rotira oko vertikalne osi, oslonca, te loptice koja se pravocrtno kreće po gredi. Sustav se može izvesti u više različitih popularnih konfiguracija. Konfiguracije se najčešće razlikuju u položaju oslonca ili položaju motora. Konfiguracija sustava loptice na gredi korištena u radu sastoji se od grede kojoj je oslonac u njenom središtu, dok je servomotor stavljen sa strane, te pričvršćen na jednom kraju grede. U ovome poglavlju prikazana je montaža svih dijelova makete. Svaki dio makete je zasebno prikazan i objašnjen. Servomotor, senzor, Arduino i izvor napajanja spojeni su pomoću male testne pločice.



## 4.1 Izrada makete sustava

Na slici 4.1 prikazana je maketa sustava korištena u radu. Na slici su prikazani svi elementi sustava loptice na gredi.



*Slika 4.1 Maketa sustava i njeni elementi*

Kao podloga makete korištena je drvena daska dimenzija 54 x 16 centimetara, dok je debljina daske bila 2 centimetara. Nosači oslonca napravljeni su od aluminijskog L profila debljine 2 milimetra, gdje podnožje L profila čini kvadrat veličine 4 x 4 centimetara, a visina L profila do oslonca iznosila je 19 centimetara.

Greda je napravljena od tankog lima debljine 1 milimetar. Lim je savijen u U profil, čije su stranice visine 3 centimetara, a podnožje U profila širine 4 centimetara. Duljina grede iznosi 60 centimetara. Kako bi greda mogla rotirati oko oslonca, 5 centimetara istog tog U profila je odrezano, okrenuto naglavačke, te potom zavareno s donje strane grede, kao što je prikazano na slici 4.2.



*Slika 4.2 Zavareni komadi U profila s donje strane grede*

Aluminijski nosači i zavareni komad U profila s donje strane grede su potom probušeni svrdlom, te je kroz njih povučen običan vijak koji je pritegnut maticom. Vijak u ovom slučaju predstavlja oslonac, odnosno os oko koje greda rotira.

Kako bi se omogućila montaža senzora, bilo je nužno pri kraju grede zadnji centimetar podnožja odrezati i malo modificirati, a stranice savinuti prema unutrašnjosti, te ih probušiti odgovarajućim svrdlom, kako je prikazano na slici 4.3. Cijeli modificirani dio je potom izgladen kako bi se odstranili oštri dijelovi lima grede.



*Slika 4.3 Prilagodba lima grede montaži senzora*

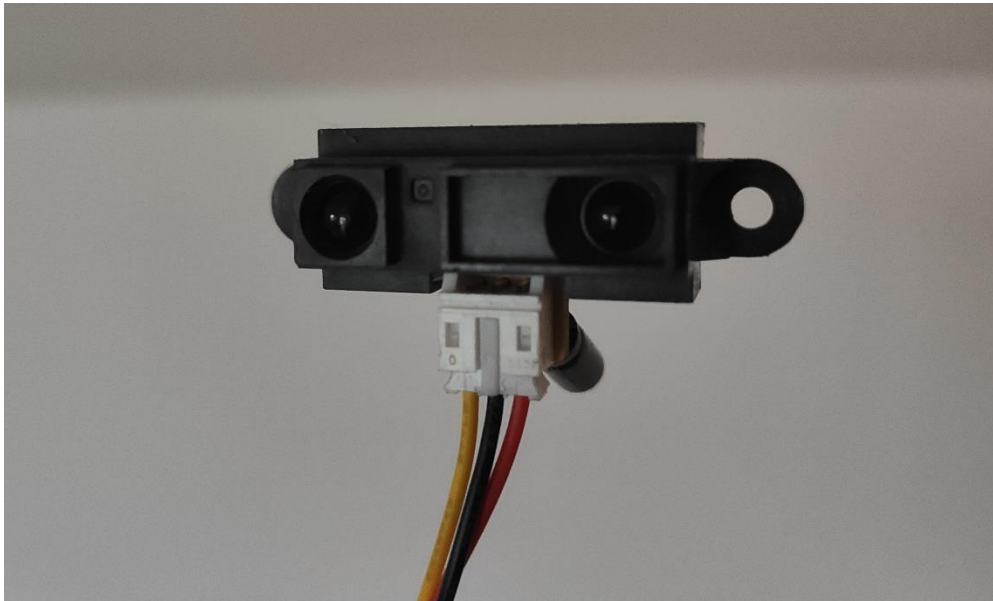
Nadalje, dodana su dva graničnika koja ograničavaju kretanje loptice. Prvi graničnik nalazi se na udaljenosti od 10 centimetara od senzora kako bi se omogućio ispravan rad senzora. Drugi graničnik nalazi se na 45 centimetara od senzora kako bi se postigla simetrija mogućeg kretanja loptice s obje strane oslonca. Izbušene su rupe, te su graničnici postavljeni pri vrhu jedne od stranica grede, kako ne bi ometali rad senzora. Graničnike predstavljaju dva vijka pritegnuta maticom, svaki dug 2 centimetara, kao što je prikazano na slici 4.4. Cijela greda je potom obojana u crnu mat boju radi lakšeg uočavanja položaja loptice, te kako bi se uklonila svaka mogućnost utjecaja reflektivne podloge na performanse infracrvenog senzora.



*Slika 4.4 Prikaz montiranog graničnika*

## 4.2 Senzor

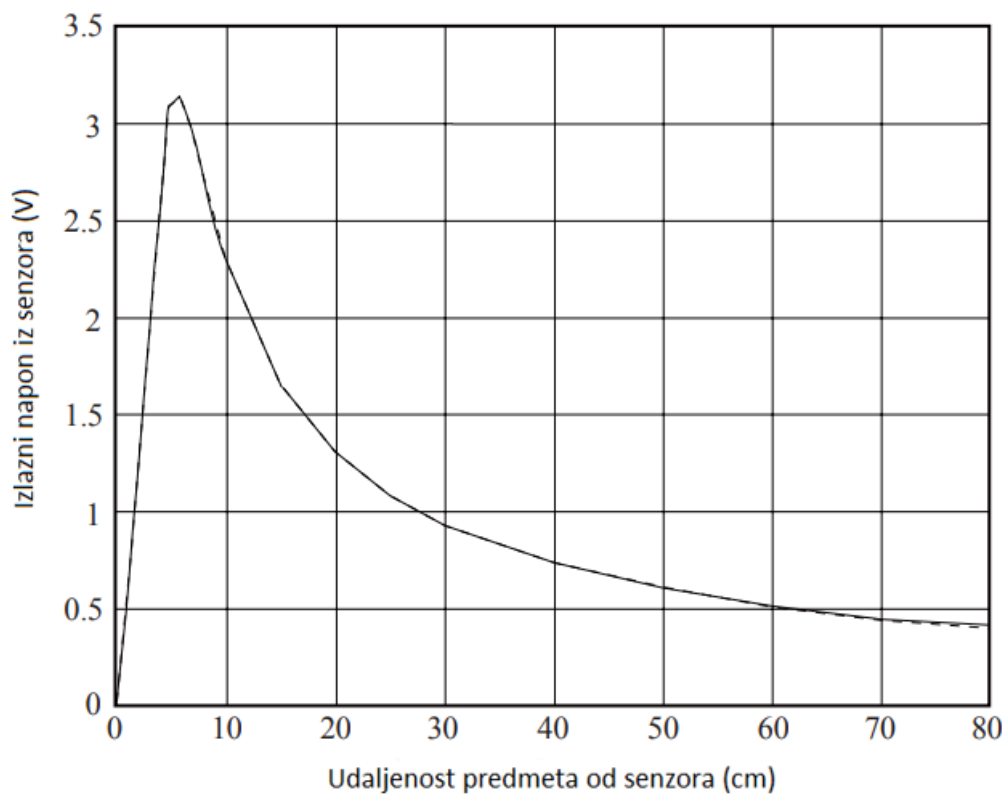
U radu je korišten infracrveni senzor Sharp IR GP2Y0A21YK0F. Senzoru je za pravilan rad nužno dovesti napon u rasponu od 4,5V do 5V, te je iz tog razloga senzor napajan putem Arduino pločice. Princip rada senzora temelji se na emitiranju nevidljivih infracrvenih zraka i mjerenju intenziteta odbijenih zraka od objekta. Jedna infracrvena LED dioda (eng. IR Transmitter) odašilje infracrvene zrake, dok druga fotodioda (eng. IR Receiver) prima te zrake, te im mjeri intenzitet.



*Slika 4.5 Prikaz prednje strane Sharp IR GP2Y0A21YK0F senzora*

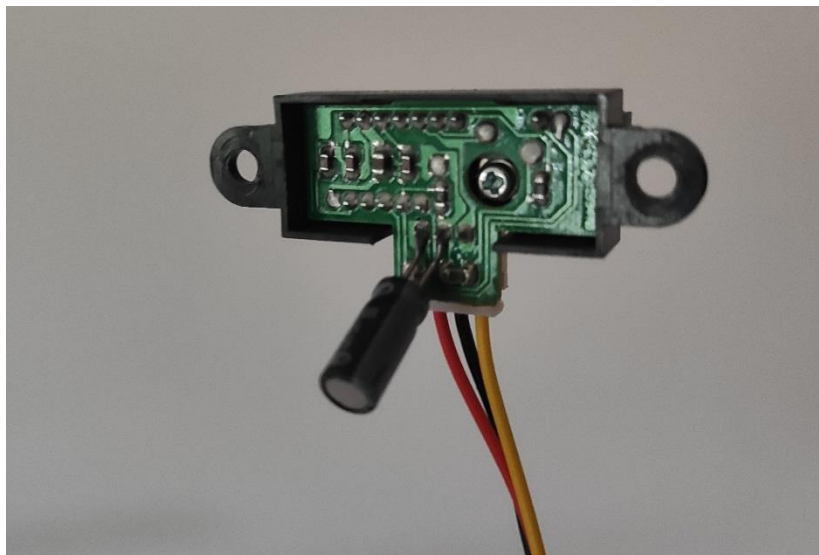
U ovisnosti o izmjerenom intenzitetu odbijenih infracrvenih zraka, senzor na izlaz daje određeni napon. Izlazni napon senzora mjeri Arduino, koji korištenjem algoritma u programskom kodu računa točan položaj loptice na gredi. Između senzora i njemu najbližeg graničnika potrebno je ostaviti 10 centimetara razmaka, kako bi senzor mogao ispravno raditi. Nužno je napomenuti da karakteristika senzora nije linearna, već ju je potrebno linearizirati.

Ovisnost izlaznog napona o udaljenosti objekta može se pronaći u službenim specifikacijama senzora, te je prikazana na slici 4.6. [5]



Slika 4.6 Prikaz ovisnosti izlaznog napona senzora o udaljenosti objekta od senzora [5]

U specifikacijama senzora naveden je naputak kojim se preporuča dodavanje kondenzatora od 10  $\mu\text{F}$ , kako bi se uklonili šumovi i padovi napona, te stabiliziralo napajanje senzora. Kako bi kondenzator stabilizirao napon napajanja, kondenzator je nužno montirati što bliže senzoru. Montaža kondenzatora na senzor prikazana je na slici 4.7.



*Slika 4.7 Prikaz montiranog kondenzatora na stražnju stranu senzora*

Mjerenje udaljenosti u sustavu loptice na gredi moguće je postići na razne druge načine. Često se koriste i ultrazvučni senzori. Također, postoje i izvedbe s otpornom žicom kod koje loptica leži na dvije žice od kojih je jedna vodljiva, a druga otporna. Ako se žice spoje na napon, tada su žice kratko spojene na mjestu trenutnog položaja loptice, te se izlazni napon mijenja u ovisnosti o otporu žice. Otpor žice ovisi o položaju loptice.

#### 4.2.1 Linearizacija nelinearne statičke karakteristike senzora

Senzor koji mjeri udaljenost loptice na izlazu daje određeni napon, koji ovisi o statičkoj karakteristici senzora definiranoj u specifikacijama senzora, a prikazanoj na slici 4.8. Budući da ta karakteristika nije linearna, kako bi se mogao projektirati PID regulator korištenjem teorije upravljanja linearnim sustavima, potrebno je linearizirati statičku karakteristiku senzora. Linearizacija funkcije predstavlja nadomještanje nelinearne funkcije linearnom, u okolišu odabrane radne točke. Linearni oblik predstavlja linearan odnos između dviju veličina, u ovom slučaju napona kojeg mjeri senzor i udaljenosti loptice od senzora. Linearizacija karakteristike se provodi kroz točke (0.2, 1.3) i (0.5, 0.6).

$$y_1 = k \cdot x_1 + l \quad \rightarrow \quad 1.3 = k \cdot 0.2 + l \quad (4-1)$$

$$y_2 = k \cdot x_2 + l \quad \rightarrow \quad 0.6 = k \cdot 0.5 + l \quad (4-2)$$

Množenjem (4-2) s (-1), a zatim zbrajanjem s (4-1), dobiva se sljedeći izraz:

$$1.3 - 0.6 = k \cdot (0.2 - 0.5) \quad (4-3)$$

$$k = -\frac{0.7}{0.3} = -\frac{7}{3} \quad (4-4)$$

Konstanta l dobiva se uvrštavanjem konstante k u (4-1):

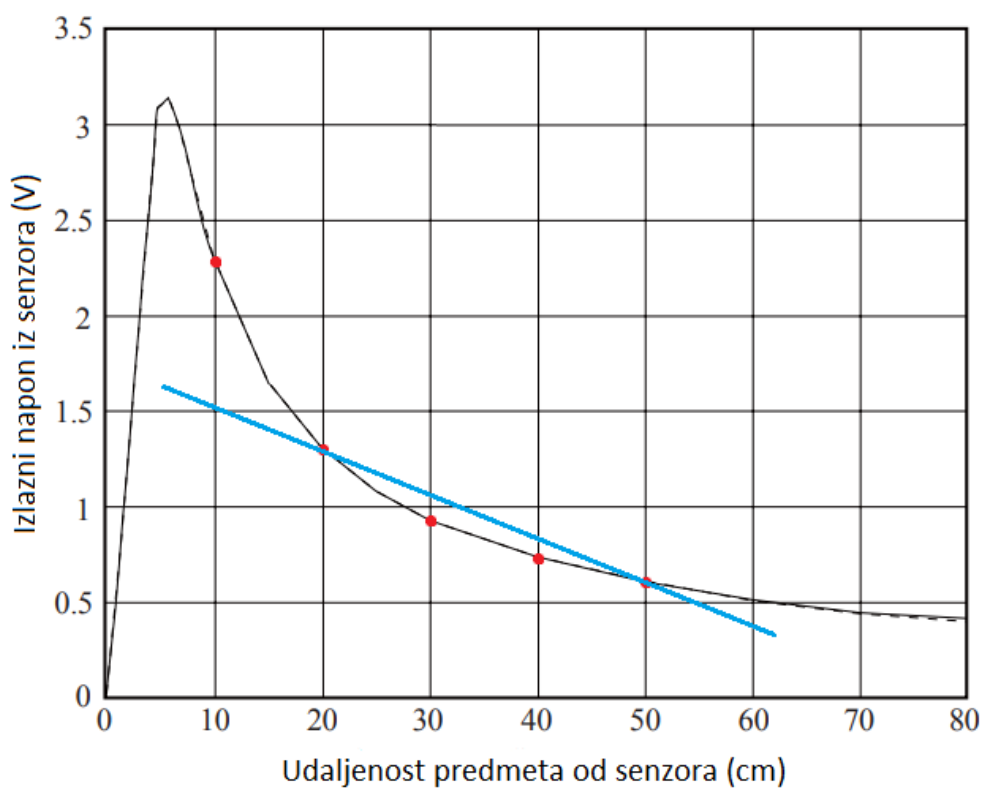
$$l = y_1 - k \cdot x_1 = 1.3 - \left(-\frac{7}{3}\right) \cdot \frac{2}{10} = \frac{13}{10} + \frac{14}{30} \quad (4-5)$$

$$l = \frac{3 \cdot 13 + 14}{30} = \frac{39 + 14}{30} = \frac{53}{30} \quad (4-6)$$

Jednadžba pravca dobivenog linearizacijom iznosi:

$$y = -\frac{7}{3}x + \frac{53}{30} \quad (4-7)$$

Pravac dobiven linearizacijom statičke karakteristike prikazan je na slici 4.8, a označen je plavom bojom.



Slika 4.8 Prikaz pravca dobivenog linearizacijom statičke karakteristike.



### 4.3 Servomotor

Servomotor korišten u radu je Power HD 1501MG servomotor tvrtke Pololu. Za potrebe rada, bilo je nužno imati dovoljno jak i brz motor koji može odgovoriti zahtjevima regulacije. Motor je napajan vanjskim izvorom napajanja na kojem je moguće podesiti napon i maksimalnu struju. Motor se nipošto ne preporuča napajati iz Arduino pločice, jer zbog relativno velike struje koju uzima može doći do oštećenja Arduino mikrokontrolera. Na slici 4.9 prikazan je servomotor korišten u ovom radu.



*Slika 4.9 Prikaz servomotora korištenog u radu*

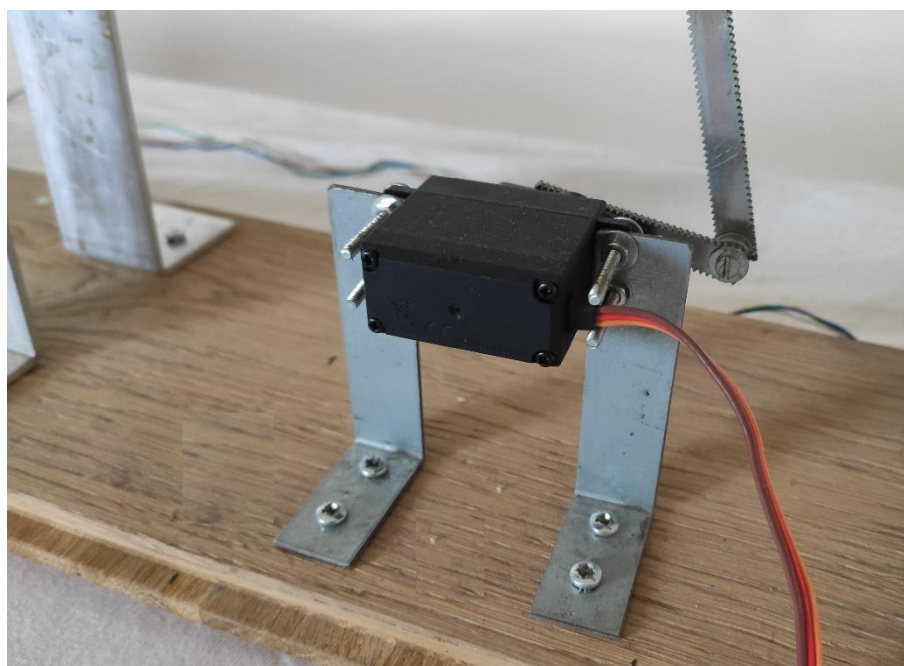
U sustavu loptice na gredi, u svrhu aktuatora najčešće se koriste servomotori, dok su moguće i izvedbe sa step motorima te većim istosmjernim motorima. Veličinu i snagu motora potrebno je odabrati tako da zadovoljavaju potrebne uvjete za realizaciju projekta, a pri odabiru motora važni parametri su zasigurno zakretni moment, brzina okretanja te maksimalni kut zakreta.

Relevantne specifikacije servomotora prikazane su u tablici 4.1. [6]

|  |                       |        |
|--|-----------------------|--------|
| Dimenzije                              | 40,7 x 20,5 x 39,5 mm |        |
| Masa                                   | 63 ± 1g               |        |
| Raspon radnog napona                   | 4,8V – 6V             |        |
| Zakretni moment pri napajanju od 6V    | 17 kg/cm              |        |
| Brzina okretanja pri napajanju od 6V   | 0,14 sec/60°          |        |
| Zakretni moment pri napajanju od 4.8V  | 15,5 kg/cm            |        |
| Brzina okretanja pri napajanju od 4.8V | 0,16 sec/60°          |        |
| Maksimalni kut zakreta                 | 180° ± 10°            |        |
| Raspon radne temperature               | -10°C – 50°C          |        |
| Potrošnja struje u stanju praznog hoda | Pri 4.8V              | Pri 6V |
|  | 400 mA                | 500 mA |
| Potrošnja struje u stacionarnom stanju | 4 mA                  | 5 mA   |

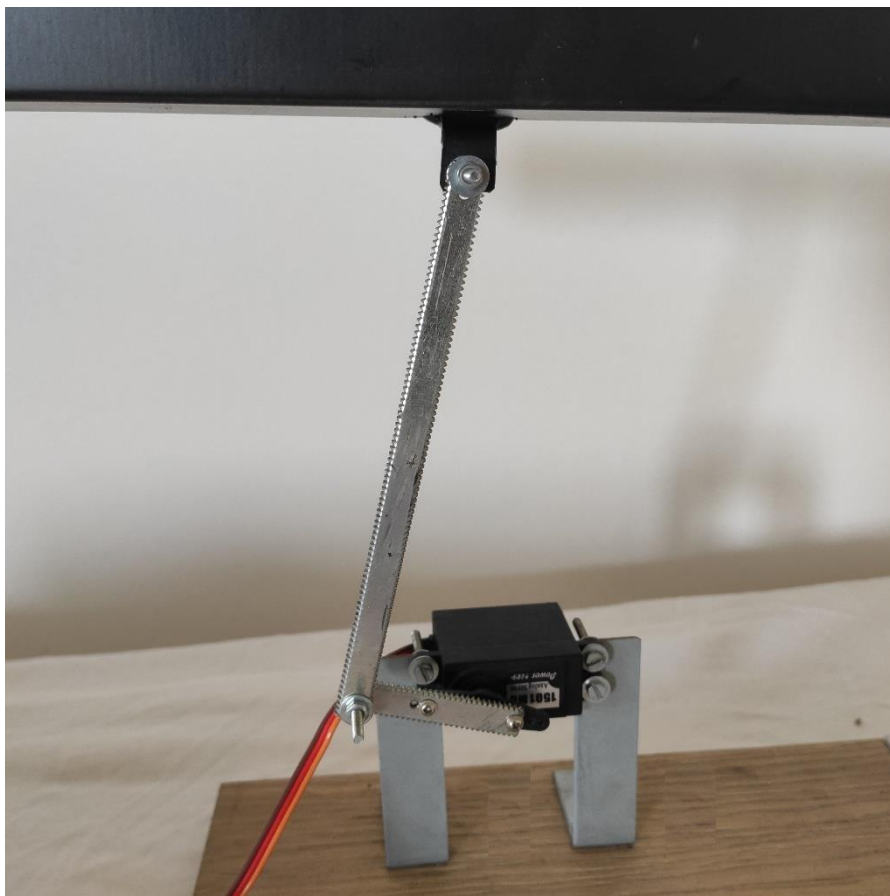
*Tablica 4.1 Prikaz specifikacija servomotora [6]*

Servomotor je montiran na dva mala limena L profila debljine 1 milimetar i pričvršćen vijcima, a L profili su također vijcima pričvršćeni za drvenu podlogu. Podnožje L profila čini pravokutnik dimenzija 2 x 4 centimetara, dok je visina L profila 7 centimetara. Osovina servomotora nalazi se na 6,5 centimetara visine od drvene podloge. Slika 4.10 prikazuje stražnju stranu montaže servomotora.



*Slika 4.10 Prikaz stražnje strane montaže servomotora*

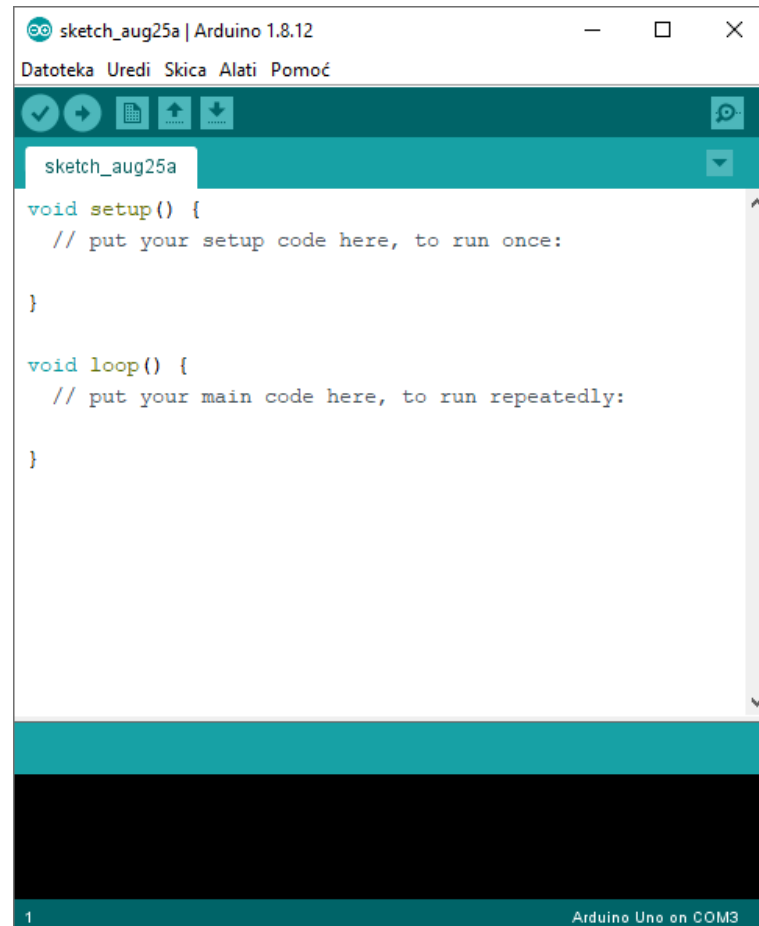
Kako bi matematički model bio ispravan, nužno je osigurati da je osovina servomotora montirana točno ispod mjesta gdje je poluga motora pričvršćena za gredu. Poluga motora pričvršćena je vijkom za mali komadić lima probušen odgovarajućim svrdlom, te zavaren za gredu točno ispod krajnjeg graničnika nasuprot senzora. Za ispravnost formule za odnos kuta položaja grede  $\alpha$  i kuta servomotora  $\theta$ , nužno je da je poluga okomita na drvenu podlogu u krajnje gornjem i krajnje donjem položaju servomotora. Budući da se servomotor može okretati od  $0^\circ$  do  $180^\circ$ , servomotor je potrebno spojiti na Arduino, te ga kalibrirati tako da je greda u vodoravnom položaju kad je servomotoru zadan položaj od  $90^\circ$ . Slika 4.11 prikazuje prednju stranu montaže servomotora.



*Slika 4.11 Prikaz prednje strane montaže servomotora*

## 4.4 Arduino IDE i programski kod

Arduino IDE je programsko razvojno okruženje u kojem se izrađuje program za Arduino mikrokontroler. Program se u razvojnom okruženju piše pojednostavljenim C++ jezikom, što ga čini jednostavnim za razumijevanje. Također, unutar razvojnog okruženja nalaze se brojni primjeri programa pogodnih za početnike.



Slika 4.12 Prikaz početnog sučelja Arduino razvojnog okruženja

Na slici 4.12 prikazano je početno sučelje Arduino razvojnog okruženja. Početni programski kod se sastoji od dvije funkcije. Programski kod napisan unutar funkcije `void setup()` izvršava se samo jednom, prilikom pokretanja programa. S druge strane, programski kod unutar funkcije `void loop()` izvršava se nakon što se izvrši funkcija `void setup()`, no za razliku od `void setup()` funkcije, izvršavanje programskog koda unutar „`void loop()`“ funkcije ponavlja se beskonačno mnogo puta.

Kako bi sustav upravljanja položaja loptice na gredi pravilno funkcionirao, bilo je potrebno napisati programski kod za regulaciju. Programski kod se između ostaloga sastoji i od gotove biblioteke (eng. library) za PID regulaciju. [7]

```
#include <PID_v1.h>
#include <Servo.h>

Servo servomotor;
int Sensor_in = A0;
double distance = 0;

double Setpoint, Input, Output;

double Kp = 2.5, Ki = 0.15, Kd = 1.5;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
```

*Slika 4.13 Prikaz deklaracije globalnih varijabli i korištenja PID biblioteke [7]*

Na slici 4.13 prikazan je početak programskog koda. Korištenjem `#include` naredbe obuhvaćaju se željene vanjske biblioteke, te im se omogućava korištenje u programskom kodu. Osim PID biblioteke, nužno je koristiti i biblioteku za servomotor, kako bi se lakše upravljalo servomotorom. Nadalje, u kodu se deklariraju razne globalne varijable, te im se po potrebi postavljaju inicijalne vrijednosti.

Funkcija „PID myPID“ stvara PID regulator. Funkcija za ulaz uzima željeni referentni položaj loptice (u kodu označen sa `Setpoint`), stvarni položaj loptice kojeg dobiva sa senzora (`Input`), te konstante PID regulatora (`Kp`, `Ki`, `Kd`). U varijablu `Output` sprema se izračunata vrijednost PID algoritma, koju se zatim prosljeđuje na servomotor. U PID funkciji se također može zadati i smjer (eng. `direction`) kretanja outputa, odnosno treba li se output smanjiti ili povećati ovisno o regulacijskom odstupanju. Smjer se može zadati kao `DIRECT` ili `REVERSE`, a primjena ovisi o procesu koji se regulira.

Konstante PID regulatora su dobivene eksperimentalno.

```
void setup()
{
  Serial.begin(9600);
  servomotor.attach(9);
  servomotor.write(90);
  pinMode(Sensor_in, INPUT);
  distance = get_dist(50);
  Input = distance;
  Setpoint = 20;
  myPID.SetTunings(Kp, Ki, Kd);
  myPID.SetSampleTime(50);
  myPID.SetOutputLimits(-70, 70);
  myPID.SetMode(AUTOMATIC);
}
```

*Slika 4.14 Prikaz koda unutar „void setup()“ funkcije*

Na slici 4.14 prikazana je `void setup()` funkcija. Na početku se započinje serijska komunikacija, te se prikazuje postavljanje servomotora na Arduinov pin 9 i postavljanje u početni stacionarni položaj, takav da je greda vodoravno postavljena. Zatim se pin senzora postavlja kao ulazni, te se inicijalno poziva funkcija za dobivanje udaljenosti. Dobivena vrijednost se postavlja kao input, te se postavlja željeni setpoint.

Nadalje, postavljaju se konstante PID regulatora, unutarnje odgađanje (eng. delay) između petlji PID algoritma, te automatski način rada PID regulatora. PID algoritam kao output daje vrijednost od 0 do 255, no budući da servomotor na ulaz ne prima PWM signal već vrijednost kuta, tu vrijednost potrebno je skalirati. Vrijednost je skalirana na raspon od -70 do 70.

```

void loop()
{
  distance = get_dist(50);
  Input = distance;
  if (myPID.Compute()){
    servomotor.write(Output+90);
  }

  Serial.print(Setpoint);
  Serial.print(" ");
  Serial.println(distance);
  delay(75);
}

```

*Slika 4.15 Prikaz koda unutar „void loop()“ funkcije*

Na slici 4.15 prikazana je `void loop()` funkcija. U funkciji se periodički poziva funkcija za dobivanje udaljenosti loptice i prosljeđuje se u PID algoritam koji zatim računa vrijednost outputa te ju skalira u navedenom rasponu od -70 do 70. Ta vrijednost se zatim zbraja s brojem 90 koji predstavlja onaj početni kut kod kojeg je greda u vodoravnom položaju. Time se postiže ograničavanje mogućeg okretanja servomotora u rasponu od 20° do 160°, što zadovoljava potrebe regulacije.

`Serial.print` naredbe korištene su kako bi se unutar Arduino razvojnog okruženja pomoću Serial Plotter alata mogao iscrtati graf koji uspoređuje trenutni položaj loptice i referentnu vrijednost. Željene vrijednosti šalju se putem serijske komunikacije na računalo, te se iscrtavaju na grafu. Na kraju funkcije dodan je `delay` od 75 milisekundi.

```

double get_dist(int n)
{
    long sum=0;
    for(int i=0;i<n;i++)
    {
        sum=sum+analogRead(Sensor_in);
    }
    double adc=sum/n;

    double distance_cm = 17569.7 * pow(adc, -1.2062);
    return(distance_cm);
}

```

*Slika 4.16 Prikaz `get_dist` funkcije za dobivanje trenutne udaljenosti loptice od senzora [8]*

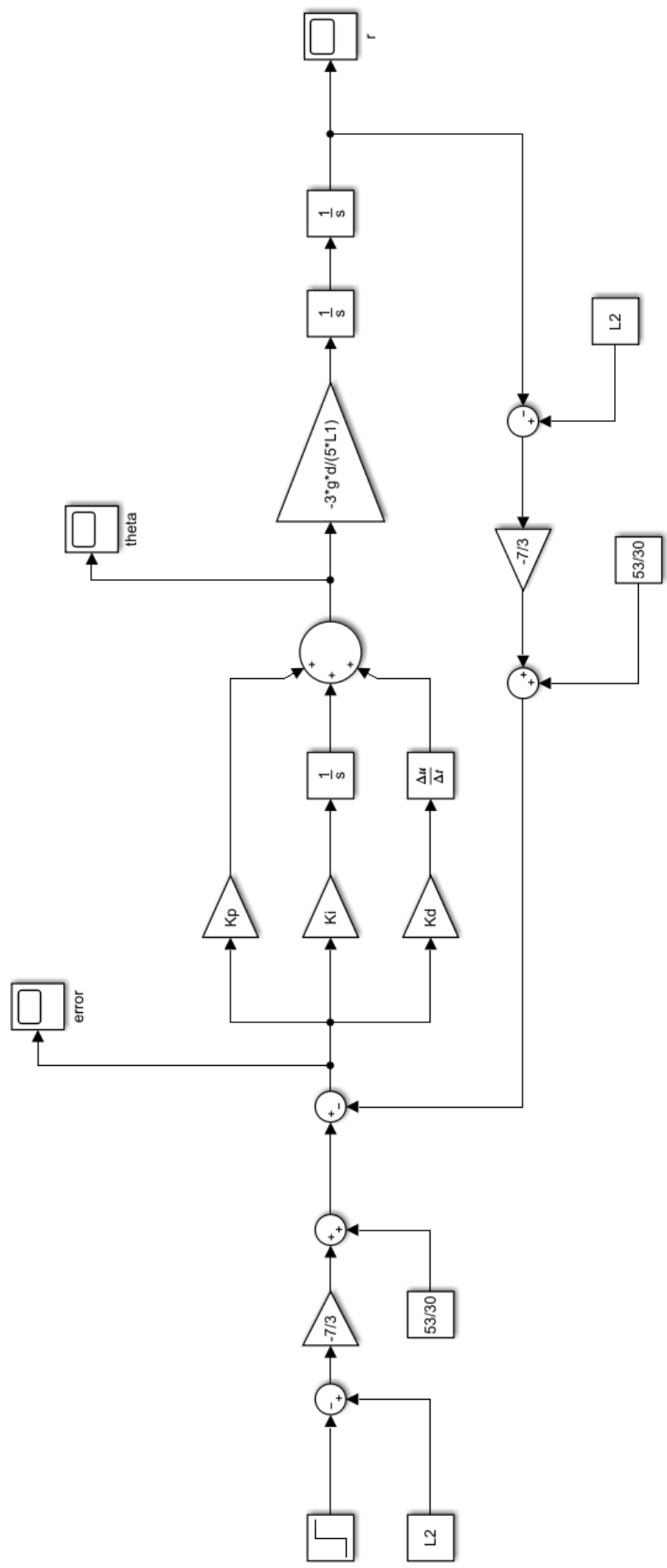
Na slici 4.16 prikazan je funkcija za dobivanje udaljenosti loptice od senzora. Funkcija za dobivanje udaljenosti loptice od senzora za ulaz uzima parametar `n`, koji označava početni broj mjerenja iz kojeg se zatim računa prosječna vrijednost udaljenosti. Funkcija `analogRead(Sensor_in)` daje vrijednost od 0 do 1024 koja se dobiva ovisno o izlaznom naponu senzora, odnosno ovisno o položaju loptice. Prosječna vrijednost tih očitavanja se zatim šalje u formulu kojom se računa udaljenost loptice, a rezultat se dobiva na dvije decimale. Izračunata vrijednost se zatim šalje u PID algoritam na dalju obradu.



## 5. Analiza rezultata

Analiza rezultata obavljena je korištenjem programskog paketa MATLAB. MATLAB je programski paket koji služi za rad s matricama, grafičko prikazivanje funkcija i podataka, implementaciju algoritama i za rješavanje raznih matematičkih problema koji su preteški za analitičko rješavanje. MATLAB-ov program Simulink koristi se za simulaciju raznih složenih sustava. U Simulink radnom prozoru moguće je izgraditi model i blok dijagram sustava koristeći biblioteku blokova. Modeliranje je proces u kojem se neki stvarni sustav predstavlja pojednostavljenim modelom, no uz zadržavanje svih relevantnih karakteristika. Simulaciju nekog sustava korištenjem modela moguće je napraviti iz više razloga. Ako se dinamika nekog sustava opisuje diferencijalnim jednažbama, ponekad su te diferencijalne jednažbe previše složene za analitičko rješavanje, pa se u tom slučaju može napraviti model sustava na kojem se vrše mjerenja. Također, ako se radi o složenom i nestabilnom sustavu ili procesu, ponekad je potrebno simulirati rezultate korištenjem modela, kako ne bi došlo do uništenja sustava usljed nestabilnosti. [9]

Korištenjem MATLAB-ovog Simulink programa, napravljen je blok dijagram cijelokupnog sustava loptice na gredi kako bi se ispitalo da li matematički model ispravno opisuje fizikalne karakteristike sustava. Na slici 5.1 prikazan je blok dijagram sustava loptice na gredi napravljen u Simulinku.

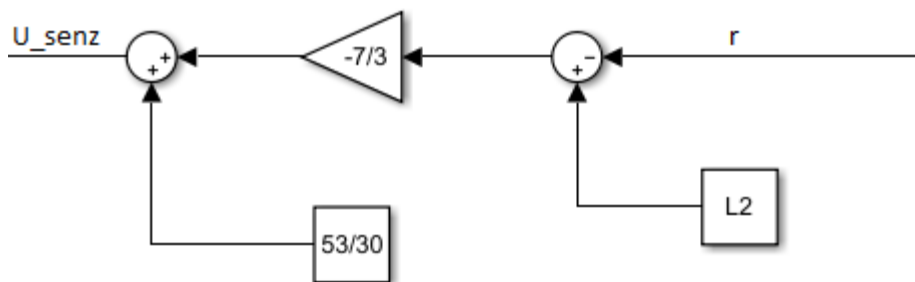


Slika 5.1 Blok dijagram simulacijske sheme sustava loptice na gredi napravljen u Simulinku

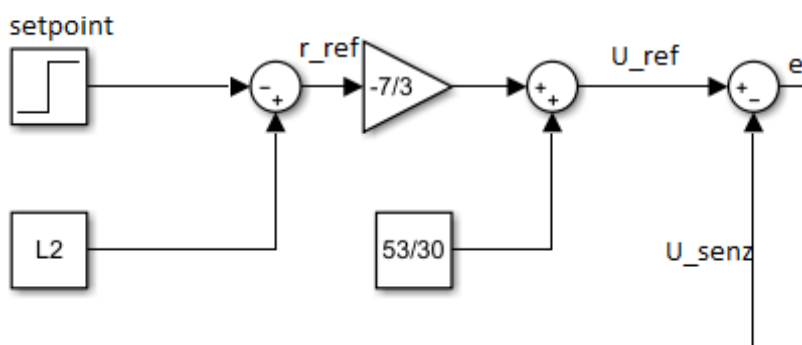
Ako se pogleda slika 3.1 koja je korištena za izradu matematičkog modela, vidi se da se položaj loptice  $r$  mjeri od suprotne strane senzora. To pak znači da stvarni položaj loptice odgovara razlici parametra  $L_2$  i vrijednosti koju mjeri senzor.

$$r = L_2 - r_{senz} \quad (5-1)$$

Iz tog razloga je prilikom izrade simulacijske sheme u povratnoj vezi potrebno oduzeti spomenute parametre, a zatim tu vrijednost pretvoriti u napon koji odgovara tom položaju prema linearnom pravcu statičke karakteristike senzora (slika 4.8). U sumacijskoj točki iz koje na izlazu dobivamo regulacijsko odstupanje ne uspoređuje se referentni položaj loptice s trenutnim, već se uspoređuje referentni napon s naponom dobivenim linearizacijom nelinearne karakteristike senzora. Na slici 5.2 prikazana je povratna veza blok dijagrama simulacijske sheme u kojoj se događa pretvorba, dok je na slici 5.3 prikazan dio blok dijagrama simulacijske sheme gdje se uspoređuju referentni napon i napon senzora.



Slika 5.2 Prikaz pretvorbe u povratnoj vezi



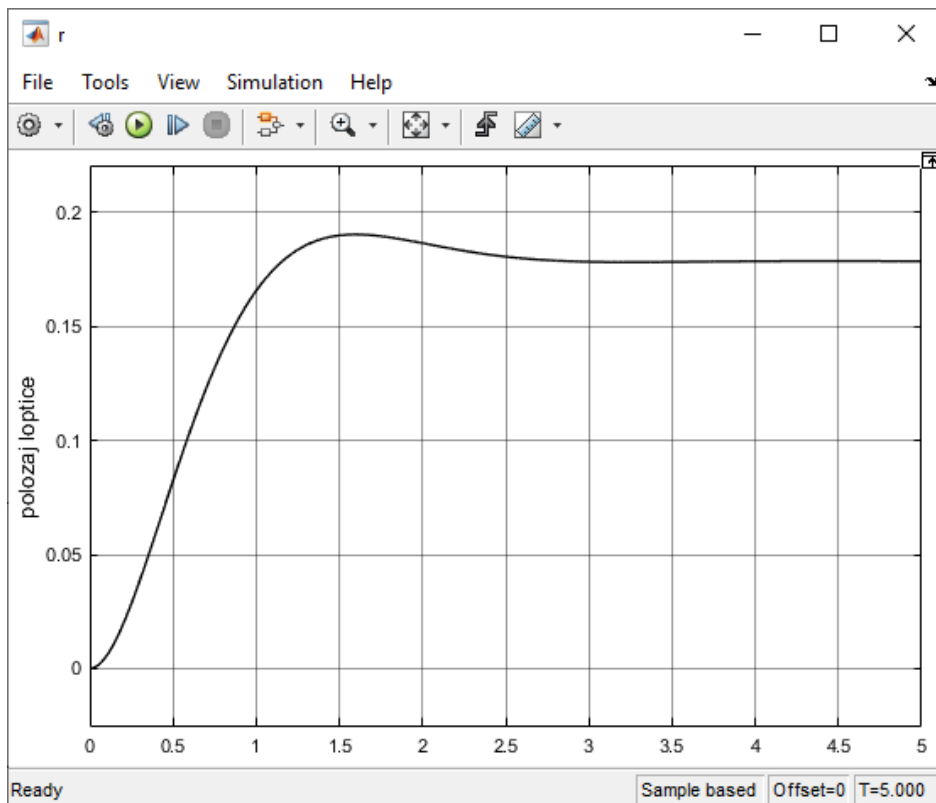
Slika 5.3 Prikaz dobivanja regulacijskog odstupanja usporedbom napona

Prije pokretanja simulacije, nužno je upisati korištene parametre u početni MATLAB-ov prozor. Vrijeme trajanja simulacije postavljeno je na 5 sekundi. Na slici 5.4 prikazano je definiranje parametara prije pokretanja.

```
Command Window
fx >> m = 0.0027
R = 0.02
g = -9.8
L1 = 0.175
L2 = 0.45
d = 0.03
J = 7.2e-7
Kp = 2.5
Ki = 0.15
Kd = 1.5
```

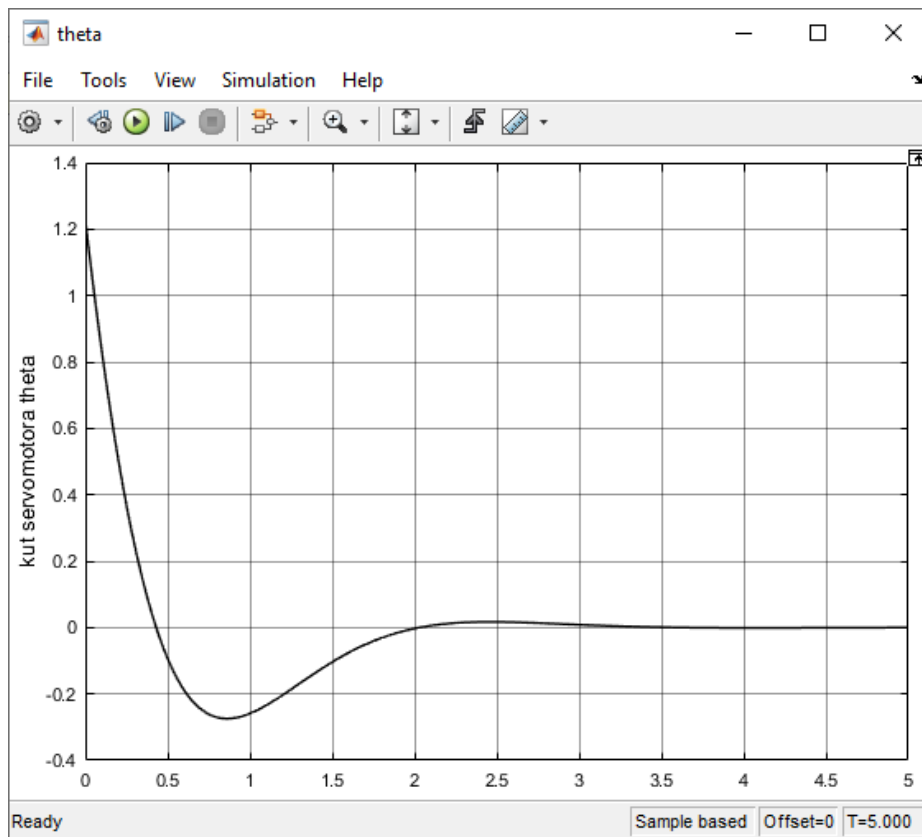
Slika 5.4 Unos parametara prije simulacije

Pokretanjem simulacije dobiveni su rezultati prikazani na sljedećim slikama. Na slici 5.5 prikazana je promjena položaja loptice u vremenu.

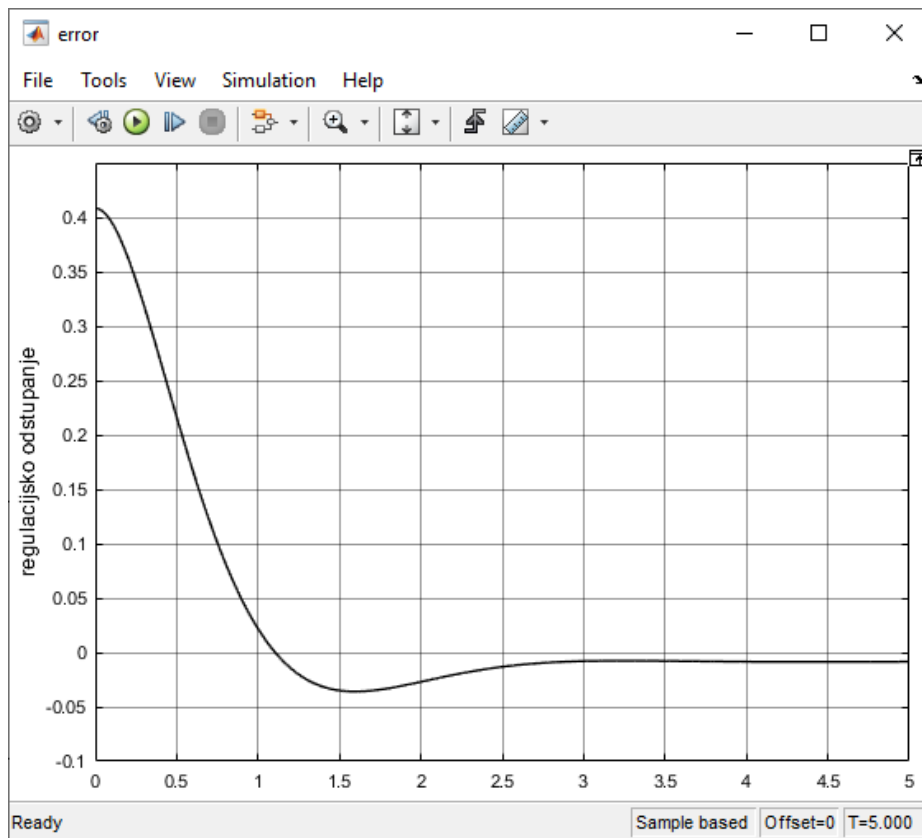


Slika 5.5 Promjena položaja loptice u vremenu

Na slici 5.6 prikazana je promjena kuta motora theta u vremenu, dok je na slici 5.7 prikazana promjena regulacijskog odstupanja u vremenu.

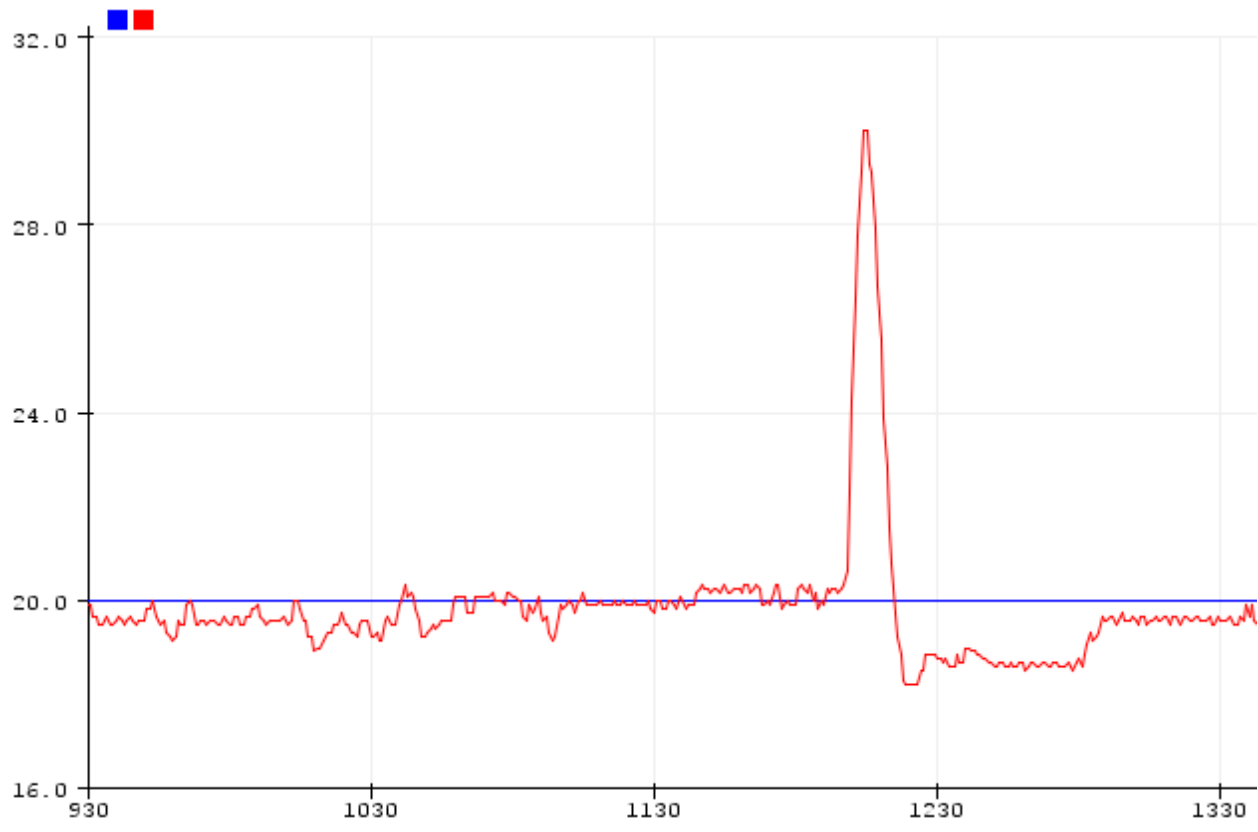


*Slika 5.6 Promjena kuta motora theta u vremenu*



*Slika 5.7 Promjena regulacijskog odstupanja u vremenu*

Slika 5.8 prikazuje stvarno ponašanje sustava. Graf je snimljen korištenjem Serial plotter alata unutar Arduino razvojnog okruženja. Plavi pravac na grafu prikazuje željeni referentni položaj, dok crvena krivulja prikazuje promjenu stvarnog položaja loptice u vremenu.



*Slika 5.8 Prikaz stvarnog ponašanja sustava*

## 6. Zaključak

U ovome radu prikazane su mogućnosti automatskog upravljanja i regulacijske tehnike, koje čine strukturu raznih industrijskih grana i modernih pogona. Sustav upravljanja položaja loptice na gredi predstavlja odličan sustav za razmatranje, te može poslužiti kao ogledan primjer nestabilnog složenog sustava upravljanog PID regulatorom.

U radu je osmišljena i realizirana izrada makete sustava, te su objašnjeni njeni elementi. Prikazan je matematički model sustava i regulatora. Prikazano je i osmišljavanje i realizacija programskog dijela sustava, te je realizirano automatsko upravljanje položaja loptice na gredi. Na kraju rada ispitano je i analizirano ponašanje sustava korištenjem programskog paketa MATLAB, te programa Simulink. Simulacijom je potvrđeno ispravno modeliranje sustava.

Prikazani primjer upravljanja također može dobro poslužiti za razumijevanje PID regulatora i razne druge edukativne svrhe.

---

(vlastoručni potpis)

U Varaždinu, 24. rujna 2020. godine



IZJAVA O AUTORSTVU  
I  
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, David Škrobar (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Automatsko upravljanje položaja loptice na gredi (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:  
(upisati ime i prezime)

David Škrobar,  
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, David Škrobar (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Automatsko upravljanje položaja loptice na gredi (upisati naslov) čiji sam autor/ica.

Student/ica:  
(upisati ime i prezime)

David Škrobar,  
(vlastoručni potpis)

## 7. Literatura

- [1] <https://drive.google.com/file/d/1LAjaDDViFG4H7dQ6PQVHo8XSQHS59GJf/view>, dostupno 13.5.2020
- [2] Z. Vukić, Lj. Kuljača: Automatsko upravljanje - analiza linearnih sustava, Kigen, 2004
- [3] <https://diyi0t.com/arduino-uno-tutorial/>, dostupno 4.8.2020
- [4] <http://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam&section=SimulinkModeling>, dostupno 17.5.2020
- [5] [https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk\\_e.pdf](https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf), dostupno 3.5.2020
- [6] <https://www.pololu.com/file/0J729/HD-1501MG.pdf>, dostupno 3.5.2020
- [7] <https://playground.arduino.cc/Code/PIDLibrary/>, dostupno 10.4.2020
- [8] [http://electronoobs.com/eng\\_arduino\\_tut100\\_code1.php](http://electronoobs.com/eng_arduino_tut100_code1.php), dostupno 13.4.2020
- [9] I. Vujović: Modeliranje i simuliranje u elektrotehnici Matlab/Simulink, Pomorski fakultet u Splitu, Split, 2014.

## Popis slika

Slika 1.1 Pojednostavljeni prikaz Ktesibijevog vodenog sata

Slika 2.1 Blok dijagram upravljanja bez povratne veze:  $X_r$  – Referentna veličina;  
 $y$  – Izlazna veličina

Slika 2.2 Blok dijagram upravljanja s povratnom vezom:  $X_r$  – Referentna veličina;  
 $y$  – Izlazna veličina;  $e$  – Zbroj ili razlika signala

Slika 2.3 Struktura PID regulatora prikazana blok dijagramom

Slika 2.4 Prikaz Arduino Uno pločice

Slika 2.5 Prikaz pinova i njihovih funkcija na Arduino Uno pločici

Slika 3.1 Skica sustava loptice na gredi

Slika 3.2 Prikaz loptice na kosini i pripadajućih sila

Slika 4.1 Maketa sustava i njeni elementi

Slika 4.2 Zavareni komadi U profila s donje strane grede

Slika 4.3 Prilagodba lima grede montaži senzora

Slika 4.4 Prikaz montiranog graničnika

Slika 4.5 Prikaz prednje strane Sharp IR GP2Y0A21YK0F senzora

Slika 4.6 Prikaz ovisnosti izlaznog napona senzora o udaljenosti objekta od senzora

Slika 4.7 Prikaz montiranog kondenzatora na stražnju stranu senzora

Slika 4.8 Prikaz pravca dobivenog linearizacijom statičke karakteristike

Slika 4.9 Prikaz servomotora korištenog u radu

Slika 4.10 Prikaz stražnje strane montaže servomotora

Slika 4.11 Prikaz prednje strane montaže servomotora

Slika 4.12 Prikaz početnog sučelja Arduino razvojnog okruženja

Slika 4.13 Prikaz deklaracije globalnih varijabli i korištenja PID biblioteke

Slika 4.14 Prikaz koda unutar „void setup()“ funkcije

Slika 4.15 Prikaz koda unutar „void loop()“ funkcije

Slika 4.16 Prikaz `get_dist` funkcije za dobivanje trenutne udaljenosti loptice od senzora

Slika 5.1 Blok dijagram simulacijske sheme sustava loptice na gredi napravljen u Simulinku

Slika 5.2 Prikaz pretvorbe u povratnoj vezi

Slika 5.3 Prikaz dobivanja regulacijskog odstupanja usporedbom napona

Slika 5.4 Unos parametara prije simulacije

Slika 5.5 Promjena položaja loptice u vremenu

Slika 5.6 Promjena kuta motora theta u vremenu

Slika 5.7 Promjena regulacijskog odstupanja u vremenu

Slika 5.8 Prikaz stvarnog ponašanja sustava

## **Popis tablica**

Tablica 3.1 Prikaz parametara korištenih u matematičkom modelu

Tablica 4.1 Prikaz specifikacija servomotora