

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Toni Bešenić

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

dr. sc. Mladen Crneković

Student:

Toni Bešenić

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem mentoru dr.sc. Mladenu Crnekoviću na savjetima pri izradi rada te uopće za samu ideju rada. Zahvaljujem svojoj obitelji koja mi je uvijek stalna podrška prije i tijekom studija.

Toni Bešenić



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 24 – 06 / 1	
Ur.broj: 15 – 24 –	

ZAVRŠNI ZADATAK

Student: **Toni Bešenić**

JMBAG: **0035235903**

Naslov rada na hrvatskom jeziku: **Regulacija položaja slabo prigušenih inercijskih sustava**

Naslov rada na engleskom jeziku: **Position control of low damped inertial systems**

Opis zadatka:

Slabo prigušeni inercijski sustavi posebno su izazovni za regulaciju jer su slabo intuitivni. Zbog male prigušenosti vrijeme oscilacija je veliko i lako mogu postati nestabilni. Takvi sustavi su svemirske letjelice i dronovi, ali i ostali nemehanički sustavi s malim otporom. Ponašanje sustava treba pokazati na modelu balansiranja kuglice na kosini koji bi se mogao koristiti na vježbama iz automatike.

U radu je potrebno:

- Konstruirati i izraditi model aktivne kosine s vodilicom za kuglicu.
- Odabrati pogonski element za nagnjanje kosine, inklinometar i daljinomjer.
- Izvesti matematički model procesa.
- Predložiti upravljački algoritam i prikazati karakteristične odzive sustava.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

24. 4. 2024.

Datum predaje rada:

2. rok (izvanredni): 11. 7. 2024.
3. rok: 19. i 20. 9. 2024.

Predviđeni datumi obrane:

2. rok (izvanredni): 15. 7. 2024.
3. rok: 23. 9. – 27. 9. 2024.

Zadatak zadao:

prof. dr. sc. Mladen Crneković

Predsjednik Povjerenstva:

izv. prof. dr. sc. Petar Ćurković

Sadržaj

Sadržaj	I
Popis slika	III
Popis tablica	IV
Popis oznaka	V
Sažetak	1
Summary	2
1 Uvod	3
1.1 Općenito o slabo prigušenim inercijskim sustavima	3
1.2 Povijesni kontekst	4
2 Matematičko modeliranje	5
2.1 Euler-Lagrange jednadžbe	5
2.2 Momenti tromosti	6
2.2.1 Moment inercije grede i utega	6
2.2.2 Moment inercije kuglice	7
2.3 Greda i uteg	8
2.4 Vektor položaja i brzina	9
2.5 Kuglica	9
2.6 Raspis Euler-Lagrange jednadžba	10
2.6.1 Moment prve upravljane koordinate	10
2.6.2 Moment druge upravljane koordinate	11
2.7 Linearizacija sustava	11
2.8 Prostor stanja	11
3 Odabir komponenti	14
3.1 Daljinomjer DAOKAI VL53L0X	14
3.2 Inklinometar MPU6050	15
3.3 NEMA 17 koračni motor	16

4	Konstruktivska izvedba	18
4.1	Ideja iza konstrukcije	18
4.2	Sklopni crtež	20
5	Elektronika	21
6	Program	23
6.1	Diskretni PID algoritam	23
6.2	PID regulator u diskretnoj računalnoj izvedbi	24
6.3	Komplementarni filter	24
6.4	Testiranje VL53L0X	26
6.5	Odziv sustava	28
6.6	Arduino kod	29
7	Zaključak	39
	Literatura	40
	Prilozi	41

Popis slika

1	Satelit u svemiru kao primjer slabo prigušenog inercijskog sustava	3
2	Dron kao primjer slabo prigušenog inercijskog sustava	4
3	Skica sustava	5
4	Skice osi grede	6
5	Skice osi utega	7
6	Skica kuteva i veličina pri zakretanju grede	8
7	VL53LOX TOF senzor	14
8	MPU6050	15
9	NEMA 17 Koračni motor	16
10	Momentna karakteristika NEMA17 motora	17
11	Uležištenje modela	18
12	Presjek postolja modela	19
13	Render sklopa u Solidworksu	20
14	Okretni enkoder	21
15	A4988 driver za koračni motor	22
16	Princip komplementarnog filtra	25
17	Odziv komplementarnog filtra	26
18	Neprecizna mjerenja VL53L0X	27
19	Mjerenja kod pozicije kuglice oko 200 mm od VL53L0X senzora	27
20	Odziv prave makete	28
21	Sklop modela aktivne kosine	42
22	Električna shema elektronike modela	43
23	Slika izvedene makete	44

Popis tablica

2	Tablica s podacima o NEMA17 koračnom motoru [1]	16
---	---	----

Popis oznaka

Oznaka	Mjerna jedinica	Objašnjenje
T_{ij}	N, Nm	sila/moment u i-toj upravljanoj koordinati
K_j	J	kinetička energija j-te koordinate
P_j	J	potencijalna energija j-te koordinate
q_i	-	i-ta upravljana koordinata
I	$\text{kg} \cdot \text{m}^2$	moment inercije tijela
I_{cm}	$\text{kg} \cdot \text{m}^2$	moment inercije oko osi kroz centar mase
m	kg	masa tijela
x	m	položaj kuglice
I_L	$\text{kg} \cdot \text{m}^2$	moment inercije grede
I_{cmL}	$\text{kg} \cdot \text{m}^2$	moment inercije grede kroz centar mase
m_L	kg	masa grede
d	m	udaljenost osi rotacije od centra mase grede
L	m	duljina grede
I_U	$\text{kg} \cdot \text{m}^2$	moment inercije utega
I_{cmU}	$\text{kg} \cdot \text{m}^2$	moment inercije utega kroz centar mase
m_U	kg	masa utega
R	m	radijus utega
k	m	udaljenost utega od osi rotacije
J	$\text{kg} \cdot \text{m}^2$	moment inercije kuglice
r	m	radijus kuglice
θ	rad	kut nagiba grede
g	m/s^2	gravitacijsko ubrzanje
$\dot{\theta}$	rad/s	kutna brzina
$\ddot{\theta}$	rad/s^2	kutno ubrzanje
v_2	m/s	brzina
ω	rad/s	kutna brzina
p_2	m	položaj kuglice
\dot{x}	m/s	brzina kuglice u smjeru x
\ddot{x}	m/s^2	ubrzanje kuglice u smjeru x
T	Nm	moment grede

x_1	m	položaj u smjeru x
x_2	m/s	brzina u smjeru x
x_3	rad	kut nagiba grede
x_4	rad/s	kutna brzina

Sažetak

Cilj ovog rada je proučavanje načina reguliranja položaja slabo prigušenih inercijskih sustava. Na početku navodimo par primjera takvih sustava korištenih u praksi. Algoritam regulacije jednog takvog sustava ćemo prikazati pomoću makete kuglice na kosini čiji se položaj može regulirati pomoću senzora daljine kuglice, senzora nagiba, mikroupravljača te motora koji zakreće kosinu. Tako se može prikazati djelovanje digitalnog PID regulatora na slabo prigušeni, podupravljeni, nelinearni sustav poput ove makete kuglice na kosini.[2]

Za ostvarenje makete je potrebno izvesti matematički model koji opisuje dinamiku sustava. Izvedeni model ćemo zatim pojednostaviti i linearizirati kako bi ga lakše simulirali. Zatim slijedi odabir komponenti koje će se koristiti u sustavu. Kada imamo opisanu dinamiku i specifikacije odabranih komponenti možemo krenuti u konstrukcijsku razradu i izvedbu. Nakon fizičke izvedbe dodajemo ostatak komponenti te programiramo mikroupravljač.

Ključne riječi: Slabo prigušeni inercijski sustav, regulacija, matematičko modeliranje, simulacija

Summary

The aim of this work is to study the methods of controlling the position of lightly damped inertial systems. At the beginning, we provide a few examples of such systems used in practice. We will demonstrate the control algorithm of one such system using a model of a ball on an incline, where the position of the ball can be controlled using a distance sensor, an inclination sensor, a microcontroller, and a motor that tilts the incline. In this way, we can demonstrate the operation of a digital PID controller on a lightly damped, underactuated, nonlinear system like this ball-on-an-incline model.

To construct the model, it is necessary to develop a mathematical model that describes the system's dynamics. The developed model will then be simplified and linearized to facilitate simulation. Next comes the selection of the components that will be used in the system. Once the dynamics are described and the specifications of the selected components are defined, we can proceed with the design and implementation. After the physical implementation, we add the remaining components and program the microcontroller.

Keywords: Lightly damped inertial system, control, mathematical modeling, simulation

1 Uvod

1.1 Općenito o slabo prigušenim inercijskim sustavima

Slabo prigušeni inercijski sustavi su dinamički sustavi sa relativno niskim stupom prigušenja, zbog čega njihova regulacija može biti izazovna i slabo intuitivna. Relativno niski stupanj prigušenosti u odnosu na inerciju sustava često uzrokuje nepoželjne oscilatorne odzive.



Slika 1: Satelit u svemiru kao primjer slabo prigušenog inercijskog sustava

Primjer takvog sustava je satelit u orbiti. Kako bi se pravilno orijentirao mora stvoriti moment rotacije, ali nema značajnog trenja ili vanjske sile koja bi mu ispravila rotaciju.

Prigušenje je smanjenje vrijednosti neke fizikalne veličine pri prijenosu mehaničkih ili elektromagnetskih valova kroz različite medije ili optička sredstva ili prolaskom kroz uređaje zbog pretvaranja dijela energije valova u druge oblike energije [3].

Slabo prigušeni inercijski sustavi su česti u današnjoj mehatronici i robotici. Roboti i dronovi trebaju imati dobro modelirane metode regulacije jer nepoželjne oscilacije lako izbacuju sustave iz stabilnosti. Nisko prigušeni sustavi ne moraju biti mehaničke prirode, nego se pojavljuju i u elektronici u kontekstu rezonantnih krugova.

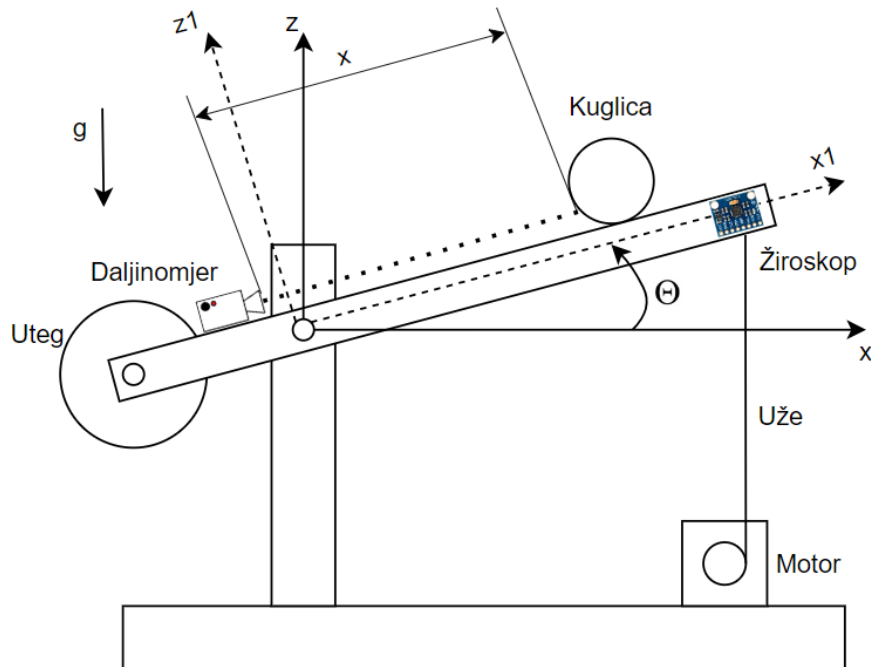


Slika 2: Dron kao primjer slabo prigušenog inercijskog sustava

1.2 Povijesni kontekst

Isaac Newton (1643–1727) je postavio temelje za koncept inercije u svojoj knjizi *Philosophiæ Naturalis Principia Mathematica* (1687). Leonhard Euler je razvio jednadžbe gibanja koje su pak generalizirale Newtonove zakone za kruta tijela i rotacijsko gibanje. Joseph Fourier (1768–1830) je prvi predložio da se periodične funkcije mogu izraziti kao suma funkcija sinusa i kosinusa, što je uvelike pridonijelo analizi oscilacijskih sustava. Hermann von Helmholtz (1821–1894) je donio značajan doprinos u proučavanju rezonancije, a Arnold Sommerfeld (1868–1951) je značajno doprinio u polju prisilnih oscilacija. Poslije 2. Svjetskog rata došlo je do naglog razvoja teorije upravljanja, zahvaljujući doprinosu znanstvenika kao Richarda Bellmana i Rudolfa Kalmana. Krajem 20. stoljeća puno je truda uloženo u proučavanje nisko prigušenih sustava sa ciljem smanjenja oscilacija i vibracija u njima. U današnje vrijeme su ovakvi sustavi i dalje tema aktualnih istraživanja, npr. u nanomehaničkim rezonatorima, zrakoplovstvu i satelitskoj tehnologiji.

2 Matematičko modeliranje



Slika 3: Skica sustava

2.1 Euler-Lagrange jednadžbe

Model prikazan na slici ćemo modelirati pomoću Euler-Lagrange jednadžba koje glase:

$$\frac{d}{dt} \left(\frac{\partial K_j}{\partial \dot{q}_i} \right) - \frac{\partial K_j}{\partial q_i} + \frac{\partial P_j}{\partial q_i} = T_{ij}$$

- T_{ij} - sila/moment u i-toj upravljanoj koordinati
- K_j - kinetička energija j-te mase
- P_j - potencijalna energija j-te mase
- q_i - i-ta upravljana koordinata

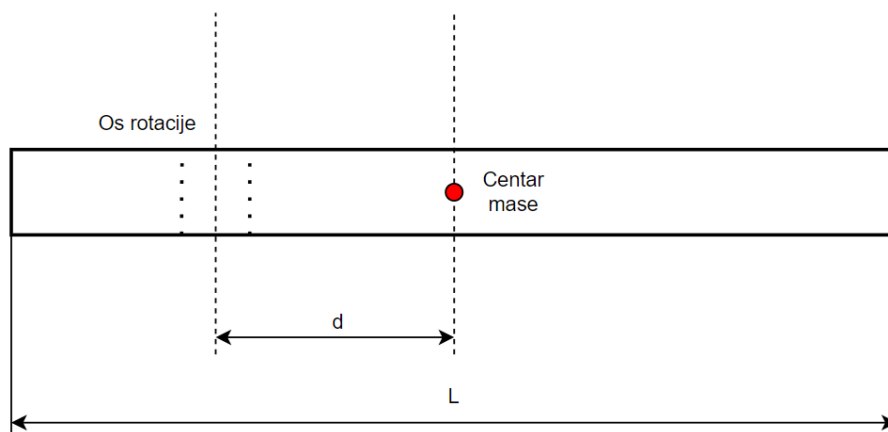
2.2 Momenti tromosti

Kako bismo dobili kinetičke energije potrebno je znati momente tromosti pojedinih tijela. Pošto tijela ne rotiraju kroz središte svoje mase potrebno je upotrijebiti teorem paralelnih osi, poznat kao i Steinerovo pravilo:

$$I = I_{\text{cm}} + mx^2$$

- I - Moment inercije oko odabrane osi
- I_{cm} - Moment inercije oko centra mase
- m - Masa tijela
- x - udaljenost između osi rotacije i paralelne osi kroz centar mase

2.2.1 Moment inercije grede i utega

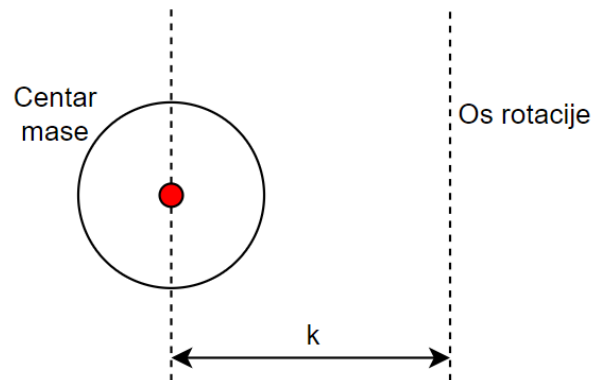


Slika 4: Skice osi grede

Na skici iznad je prikazana os rotacije grede duljine L na udaljenosti d od centra mase grede, pa za njezin moment tromosti možemo pisati:

$$I_L = I_{\text{cm}L} + m_L d^2$$

$$I_L = \frac{m_L L^2}{12} + m_L d^2$$



Slika 5: Skice osi utega

Za uteg oblika diska, mase m_U i radijusa R koji rotira na udaljenosti k od osi rotacije možemo pisati:

$$I_U = I_{cmU} + m_U k^2$$

$$I_U = \frac{1}{2} m_U R^2 + m_U k^2$$

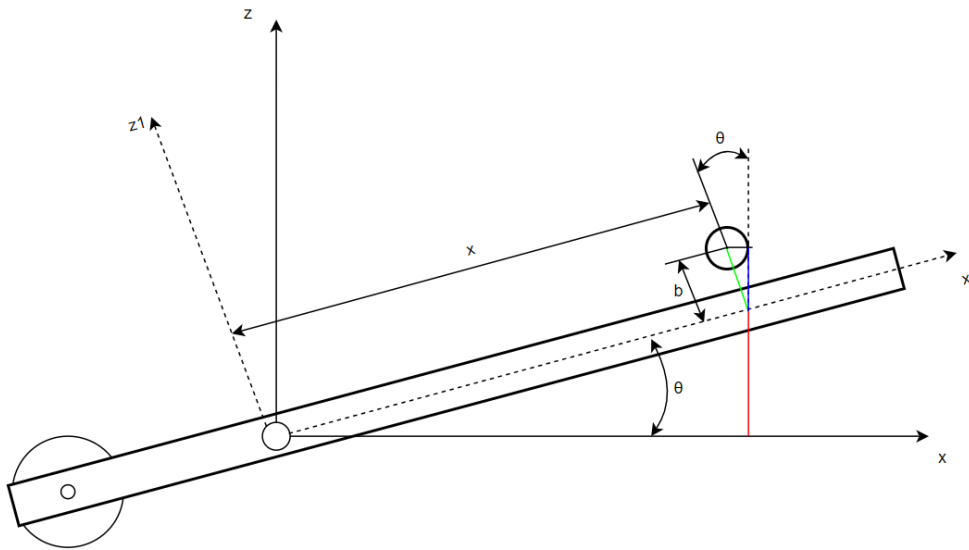
2.2.2 Moment inercije kuglice

Moment inercije kuglice mase m , radijusa r i udaljene R od osi rotacije glasi:

$$J = \frac{2}{5} m r^2 + m R^2$$

Iz skice na idućoj stranici je vidljivo da se R^2 može zapisati kao:

$$J = \frac{2}{5} m r^2 + m(x^2 + b^2)$$



Slika 6: Skica kuteva i veličina pri zakretanju grede

2.3 Greda i uteg

Pošto se greda i uteg kruto kreću zajedno, odnosno samo rotiraju oko osi rotacije, možemo ih promatrati kao kruto tijelo i zbrojiti njihove kinetičke i potencijalne energije.

Kinetička energija grede:

$$K_L = \frac{1}{2} I_L \dot{\theta}^2 = \frac{1}{2} \dot{\theta}^2 \left(\frac{m_L L^2}{12} + m_L d^2 \right)$$

Potencijalna energija grede:

$$P_L = -m_L g d \sin \theta$$

Kinetička energija utega:

$$K_U = \frac{1}{2} I_U \dot{\theta}^2 = \frac{1}{2} \dot{\theta}^2 \left(\frac{m_U R^2}{12} + m_U k^2 \right)$$

Potencijalna energija utega:

$$P_U = m_U g k \sin \theta$$

Zbrajajući ove izraze dobivamo:

$$K_1 = \frac{1}{2} \dot{\theta}^2 \left(\frac{m_L L^2}{12} + m_L d^2 \right) + \frac{1}{2} \dot{\theta}^2 \left(\frac{m_U R^2}{12} + m_U k^2 \right)$$

$$K_1 = \frac{1}{2}\dot{\theta}^2\left(\frac{m_L L^2}{12} + m_L d^2 + \frac{m_U R^2}{12} + m_U k^2\right)$$

$$P_1 = -m_L g d \sin \theta + m_U g k \sin \theta$$

$$P_1 = \sin \theta (m_U g k - m_L g d)$$

2.4 Vektor položaja i brzina

Pošto se koordinatni sustav zakreće možemo prilagoditi nove koordinate pomoću transformacije rotacije:

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ z_1 \end{bmatrix}$$

Ako uvrstimo $x_1 = x$ i $z_1 = b$ (b je udaljenost od središta grede prikazana na stranici 8) dobijemo novi položaj p_2 :

$$p_2 = \begin{bmatrix} x \cos \theta & -b \sin \theta \\ x \sin \theta & b \cos \theta \end{bmatrix}$$

Sada kada znamo položaj kuglice ovisno o zakretu i njezinom promjeru, možemo izračunati njezinu brzinu kao derivaciju njezinog položaja:

$$\frac{dp_2}{dt} = v_2 = \dot{p}_2 = \begin{bmatrix} \dot{x} \cos \theta - x \dot{\theta} \sin \theta & -b \dot{\theta} \cos \theta \\ \dot{x} \sin \theta + x \dot{\theta} \cos \theta & -b \dot{\theta} \sin \theta \end{bmatrix}$$

$$v_2^2 = [(x - b\dot{\theta}) \cos \theta - x\dot{\theta} \sin \theta]^2 + [(x - b\dot{\theta}) \sin \theta + x\dot{\theta} \cos \theta]^2$$

$$= [(\dot{x} - b\dot{\theta})^2 \cos^2 \theta - 2(\dot{x} - b\dot{\theta}) \cos \theta x \dot{\theta} \sin \theta + x^2 \dot{\theta}^2 \sin^2 \theta]$$

$$+ [(\dot{x} - b\dot{\theta})^2 \sin^2 \theta + 2(\dot{x} - b\dot{\theta}) \sin \theta x \dot{\theta} \cos \theta + x^2 \dot{\theta}^2 \cos^2 \theta]$$

$$= [(\dot{x} - b\dot{\theta})^2 + x^2 \dot{\theta}^2]$$

$$= \dot{x}^2 - 2b\dot{x}\dot{\theta} + b^2\dot{\theta}^2 + x^2\dot{\theta}^2$$

2.5 Kuglica

Kinetička energija kuglice:

$$K_2 = \frac{1}{2} m v_2^2 + \frac{1}{2} J \omega^2$$

$$K_2 = \frac{1}{2} m v_2^2 + \frac{1}{2} J \dot{\theta}^2$$

Odnosno zbog $\omega = \frac{v}{R}$ i $v = \dot{x}$:

$$K_2 = \frac{1}{2} m v_2^2 + \frac{1}{2} J \frac{\dot{x}^2}{r^2}$$

- m - masa kuglice
- v^2 - translacijska brzina kuglice
- J - rotacijski moment tromosti
- $\omega = \frac{v}{R}$ - kutna brzina kuglice
- R - udaljenost od centra rotacije

Potencijalna energija kuglice:

$$P_2 = -mgx \sin \theta - mgb \cos \theta$$

2.6 Raspis Euler-Lagrange jednadžba

Sada kada imamo sve potrebne podatke možemo raspisati Euler-Lagrange jednadžbe:

$$\frac{d}{dt} \left(\frac{\partial K_j}{\partial \dot{q}_i} \right) - \frac{\partial K_j}{\partial q_i} + \frac{\partial P_j}{\partial q_i} = T_{ij}$$

$$q_1 = x \quad , \quad q_2 = \theta$$

$$T_1 = 0 \quad , \quad T_2 = T$$

Zbrajanjem kinetičke i potencijalne energije utega, grede i kuglice dobivamo ukupnu kinetičku i potencijalnu energiju sustava.

Ukupna kinetička energija:

$$K = K_1 + K_2 = \frac{1}{2} \dot{\theta}^2 \left(\frac{m_L L^2}{12} + m_L d^2 + \frac{m_U R^2}{12} + m_U k^2 \right) + \frac{1}{2} m (\dot{x}^2 - 2b\dot{x}\dot{\theta} + b^2\dot{\theta}^2 + x^2\dot{\theta}^2) + \frac{1}{2} J \frac{\dot{x}^2}{r^2}$$

Ukupna potencijalna energija:

$$P = P_1 + P_2 = \sin \theta (m_U g k - m_L g d) - mgx \sin \theta - mgb \cos \theta$$

2.6.1 Moment prve upravljane koordinate

$$\frac{\partial K}{\partial \dot{x}} = m\dot{x} - mb\dot{\theta} + \frac{J}{r^2} \dot{x}$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{x}} \right) = m\ddot{x} - mb\ddot{\theta} + \frac{J}{r^2} \ddot{x}$$

$$\frac{\partial K}{\partial x} = m\dot{x}\dot{\theta}^2$$

$$\frac{\partial P}{\partial x} = -mg \sin \theta$$

$$m\ddot{x} - mb\ddot{\theta} + \frac{J}{r^2} \ddot{x} - m\dot{x}\dot{\theta}^2 - mg \sin \theta = 0 \quad (1)$$

2.6.2 Moment druge upravljane koordinate

$$\begin{aligned}\frac{\partial K}{\partial \dot{\theta}} &= \dot{\theta} \left(\overbrace{\frac{m_L L^2}{12} + m_L d^2 + \frac{m_U R^2}{12} + m_U k^2}^H \right) + m(-b\dot{x} + b^2\dot{\theta} + x^2\dot{\theta}) \\ \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\theta}} \right) &= H\ddot{\theta} - mb\ddot{x} + mb^2\ddot{\theta} + mx^2\ddot{\theta} \\ \frac{\partial K}{\partial \dot{\theta}} &= 0 \\ \frac{\partial P}{\partial \dot{\theta}} &= \cos \theta (m_U gk - m_L gd) - mgx \cos \theta + mgb \sin \theta \\ H\ddot{\theta} - mb\ddot{x} + mb^2\ddot{\theta} + mx^2\ddot{\theta} + \cos \theta (m_U gk - m_L gd) - mgx \cos \theta + mgb \sin \theta &= T\end{aligned}\quad (2)$$

2.7 Linearizacija sustava

Jednadžbe (1) i (2) predstavljaju nelinearni model sustava, no taj model možemo linearizirati oko radne točke i pojednostaviti sa par pretpostavka:

$$\cos \theta \cong 1 \quad , \quad \sin \theta \cong \theta \quad , \quad \dot{\theta}^2 \cong x^2 \cong \dot{\theta}\dot{x} = 0$$

Linearizirani oblik onda glasi:

$$\frac{J}{r^2} \ddot{x} + m\ddot{x} - mb\ddot{\theta} - mg\theta = 0 \quad (3)$$

$$H\ddot{\theta} + mb^2\ddot{\theta} - mb\ddot{x} + m_U gk - m_L gd - mgx\theta + mgb\theta = T \quad (4)$$

Dodatno pojednostavljenje koje možemo uvesti je da masa kuglice m ne utječe značajno na moment grede:

$$\ddot{x} \left(\frac{J}{r^2} + m \right) - mb\ddot{\theta} - mg\theta = 0 \quad (5)$$

$$H\ddot{\theta} + m_U gk - m_L gd = T \quad (6)$$

2.8 Prostor stanja

Da bi smo naš sustav od dvije diferencijalne jednadžbe drugog reda pretvorili u prostor stanja, potrebno ih je izraziti u obliku:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Prvo definiramo varijable stanja:

- $x_1 = x$
- $x_2 = \dot{x}$
- $x_3 = \theta$
- $x_4 = \dot{\theta}$

Sada možemo izraziti derivacije drugog stupnja pomoću varijabli stanja:

$$\ddot{x} = \dot{x}_2 \quad , \quad \ddot{\theta} = \dot{x}_4$$

Sada možemo raspisati jednačbe (5) i (6) pomoću varijabli stanja:

Jednačba (5) :

$$\begin{aligned} \ddot{x} \left(\frac{J}{r^2} + m \right) - mb\ddot{\theta} - mg\theta &= 0 \\ \left(\frac{J}{r^2} + m \right) \dot{x}_2 - mb\dot{x}_4 - mgx_3 &= 0 \\ \dot{x}_2 &= \frac{mb\dot{x}_4 + mgx_3}{\frac{J}{r^2} + m} \end{aligned}$$

Jednačba (6) :

$$\begin{aligned} H\ddot{\theta} + m_{\text{U}}gk - m_{\text{L}}gd &= T \\ H\dot{x}_4 &= T - m_{\text{U}}gk + m_{\text{L}}gd \\ \dot{x}_4 &= \frac{T - m_{\text{U}}gk + m_{\text{L}}gd}{H} \end{aligned}$$

Prikaz u prostoru stanja

Sada možemo zapisati jednačbe prostora stanja za $\dot{x}_1\dot{x}_2\dot{x}_3\dot{x}_4$:

- $\dot{x}_1 = x_2$
- $\dot{x}_2 = \frac{mbx_4 + mgx_3}{\frac{J}{r^2} + m}$
- $\dot{x}_3 = x_4$
- $\dot{x}_4 = \frac{T - m_{\text{U}}gk + m_{\text{L}}gd}{H}$

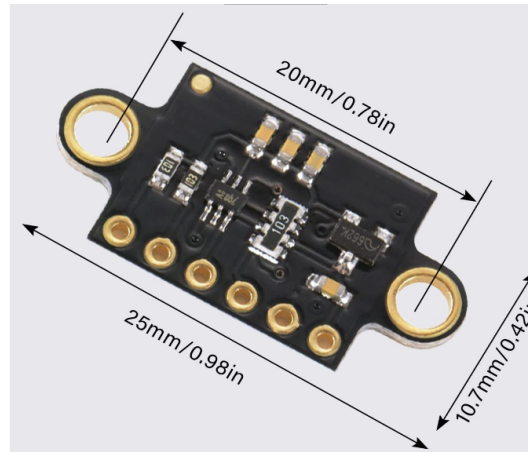
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{mg}{r^2+m} & \frac{mb}{r^2+m} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{H} \end{bmatrix} T \quad (7)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} T \quad (8)$$

3 Odabir komponenti

U okviru ovog završnog rada je potrebno odabrati daljinomjer, inklinometar i motor za naginjanje kosine.

3.1 Daljinomjer DAOKAI VL53L0X



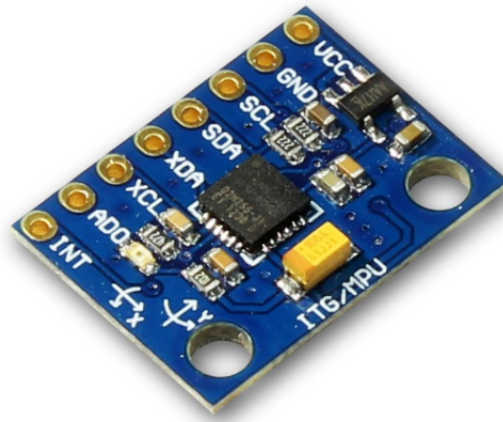
Slika 7: VL53LOX TOF senzor

Daljinomjer DAOKAI VL53L0X je time-of-flight (TOF) senzor daljine koji radi na principu infracrvenog odašiljača valne duljine 940 nm.

Odabrao sam ovaj senzor jer njegov domet, brzina mjerenja i točnost odgovaraju grubim početnim specifikacijama ovog modela. Njegove glavne karakteristike su mu [4] :

- I^2C adresa 0x52
- Mjerna udaljenost od 10 mm do 2000 mm.
- I^2C komunikacija sa mikroupravljačem.
- Razlučivost od 1 mm.
- Brzina mjerenja do 50 Hz.
- Radna temperatura od 20°C do +70°C.

3.2 Inklinometar MPU6050



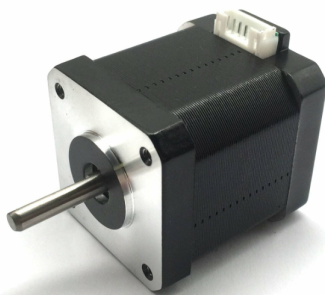
Slika 8: MPU6050

MPU6050 je šesteroosni (6-DOF) senzor inercijalnog mjerenja (IMU) koji kombinira troosni žiroskop i troosni akcelerometar u jednom modulu. Ovaj senzor je vrlo popularan u elektronici i robotici zbog svoje male veličine, niske cijene i jednostavnosti integracije s mikrokontrolerima.

Glavne karakteristike su mu [5] :

- I^2C adresa 0x68
- Mjeri ubrzanje po X, Y i Z osi, koje se mogu podešavati u rasponu mjerenja $\pm 2g$, $\pm 4g$, $\pm 8g$ i $\pm 16g$.
- Mjeri kutnu brzinu po X, Y i Z osi, u rasponu mjerenja ± 250 , ± 500 , ± 1000 i ± 2000 stupnjeva po sekundi (dps).
- Koristi I^2C komunikacijski protokol.
- Ugrađeni DMP (Digital Motion Processor) omogućuje predobradu podataka, filtriranje i spajanje podataka iz akcelerometra i žiroskopa, što omogućuje precizniji izračun orijentacije, rotacije i kretanja.
- MPU6050 se lako može povezati s mikrokontrolerima kao što su Arduino, ESP8266, ESP32 i drugi. Na internetu je dostupno mnogo biblioteka i primjera koji olakšavaju korištenje ovog senzora u projektima.

3.3 NEMA 17 koračni motor

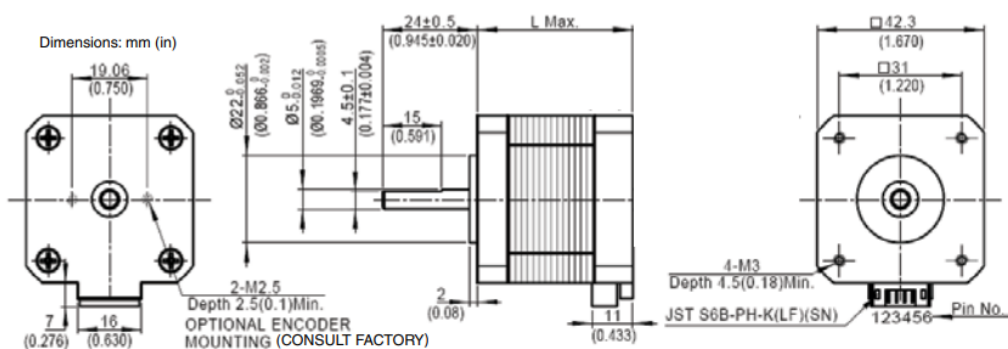


Slika 9: NEMA 17 Koračni motor

NEMA 17 koračni motor sam odabrao zbog njegovih dimezija, momentne karakteristike i načina upravljanja pomoću drivera. Naziv NEMA 17 je zbog činjenice da su mu dimenzije prednje strane motora odgovaraju NEMA 17 standardu od 1,7 inča x 1,7 inča [1].

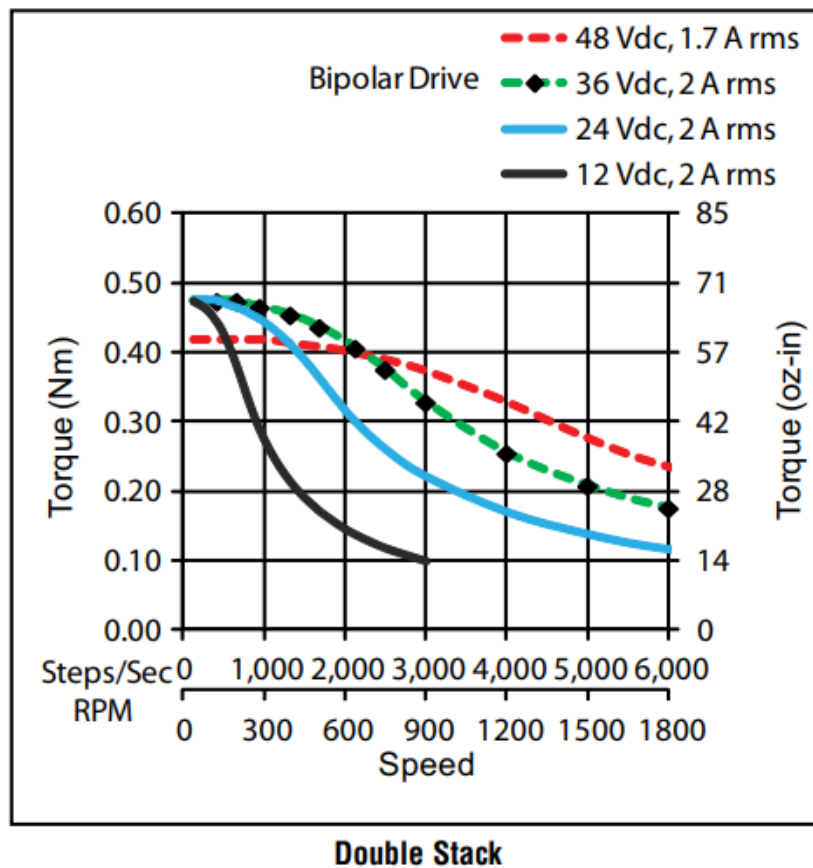
Description	Length	Mounted Rated Current	Mounted Holding Torque		Winding		Detent Torque	Rotor Inertia		Motor Weight	
			Nm Typ.	oz-in Typ.	Ohms	mH		mNm oz-in	g cm ² oz-in ²	kg	lbs
(Stack)	"L" Max	Amps			±10% @ 20°C	Typ.					
Single	39.8 mm (1.57 in)	2	0.48	68	1.04	2.2	15	2.1	57	0.31	0.28 0.62
Double	48.3 mm (1.90 in)	2	0.63	89	1.3	2.9	25	3.5	82	0.45	0.36 0.79
Triple	62.8 mm (2.47 in)	2	0.83	120	1.49	3.8	30	4.2	123	0.67	0.6 1.3

*All standard motors have plug connector. Consult factory for other options.



Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.

Tablica 2: Tablica s podacima o NEMA17 koračnom motoru [1]



Slika 10: Momentna karakteristika NEMA17 motora

Slika iznad prikazuje momentnu karakteristiku NEMA17 koračnog motora prema [1]. Iz slike je vidljivo da za niske brzine motor postiže oko 0,48 Nm.

4 Konstrukcijska izvedba

4.1 Ideja iza konstrukcije

Konstrukcijsku izradu započeo sam skiciranjem i približnim mjerenjem pomoću olovke, papira i grubih mjerenja. Nakon toga sam modeliranje započeo u Solidworks 2020. programu.

Glavna motivacija pri konstruiranju ove makete bila je *rastavljivost*. Pošto je maketa poprilično masivna kad je potpuno sklopljena vjerujem da je najpraktičnije dizajnirati na način da se može jednostavno i brzo rastaviti bez kompromisa sa za njezin mehanički rad.

Tako naprimjer kod osovine koja spaja kosinu sa postoljem se nalazi dio kroz koji ide polukružna osovina sa šipkom zavarenom na njezin kraj. Osovina je polukružna jer tako jednostavno osigurava kruto okretanje dijelova spojenih na nju. Na postolju se također nalaze kuglični ležajevi 6205 koji minimiziraju trenje pri zakretanju kosine. Makar su oba ležaja aksijalno čvrsta (što je ozbiljni propust kod čeličnih konstrukcija) osovine na kojima se nalaze su napravljene sa dovoljno zračnosti da se kompenziraju aksijalni pomaci, pa je primjena 2 aksijalno kruta ležaja u ovom modelu opravdana.



Slika 11: Uležištenje modela

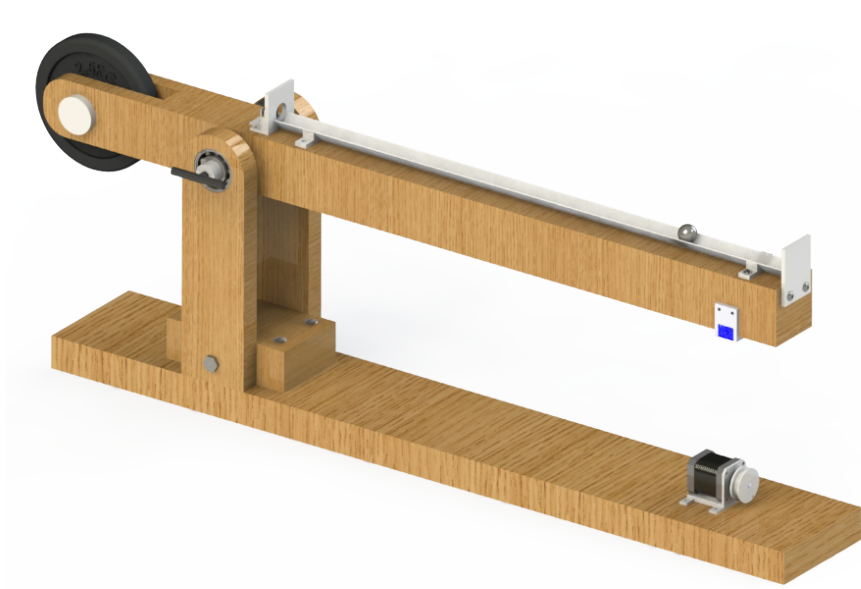
Samo postolje spaja se na dasku pomoću dugog vijka koji osigurava njegov položaj. Kroz ovakav pristup se cijeli model može lako rastaviti, što značajno povećava njegovu eventualnu

praktičnu primjenjivost pošto se onda lakše prenosi. Uteg je spojen na gredu pomoću vijka koji služi kao mirna osovina.



Slika 12: Presjek postolja modela

Za vodilicu kuglice odabrao sam aluminijski kutni profil jer odgovara dimenzijski, dovoljno je čvrst i minimizira trenje pri kotrljanju kuglice u odnosu na npr. utor u drvu. Svi PLA plastični dijelovi su izrađeni pomoću Ender 3 Pro 3D printera, korištenje 3D printanih dijelova je značajno ubrzalo prototipiranje ovog modela.



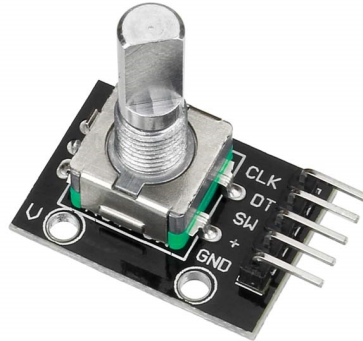
Slika 13: Render sklopa u Solidworksu

4.2 Sklopni crtež

Opisani dijelovi prikazani su u sklopnom tehničkom crtežu u prilogu.

5 Elektronika

Osnovna ideja elektronike modela je da se pomoću mehaničkog enkodera i LCD ekrana mogu proizvoljno namjestiti proporcionalna, intergirajuća i derivirajuća pojačanja modela.

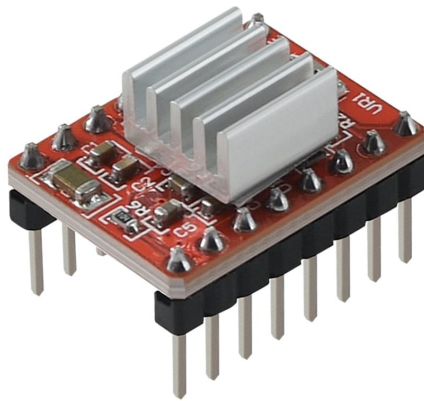


Slika 14: Okretni enkoder

Na shemi na sljedećoj strani je prikazan način spajanja svih komponenti. Pošto LCD ekran 20x4 ima mnogo izlaza/ulaza koristio sam I^2C adapter koji se onda sa Arduino MEGA spaja pomoću samo 4 žice (5V, GND, SDA i SCL).

Daljinomjer VL53L0X, MPU6050 i LCD Adapter PCF85574 koriste I^2C protokol za razmjenu informacija sa mikroupravljačem te se spajaju na pripadajuće SDA i SCL pinove na mikroupravljaču. I^2C adresa od PCF85574 je 0x27.

Za upravljanje NEMA 17 koračnim motorom se koristi A4988 driver. A4988 se spaja na vanjsko napajanje od 12V istosmjerne struje, te se paralelno na njegov ulaz i uzemljenje spaja kondenzator od 100 μF kako bi se driver zaštitio od promjena kod napona napajanja. Prije upotrebe je potrebno kalibrirati driver na željenu vrijednost postavljajući odgovarajući V_{ref} pomoću voltmetra i potencijometra na samom driveru. Enable i Sleep pinovi se moraju kratko spojiti kako bi bio omogućen rad drivera. Kod spajanja zavojnica koračnog motora na A4988 nije bitan polaritet već to da izlazi jedne zavojnice se spoje na 1A i 1B, a druge na 2A i 2B. Jednostavan način za provjeriti koji kontakti pripadaju istoj zavojnici je to da se spoje i onda ručno okrene osovina motora, ako se osjeti povećan otpor okretanja to znači da su ti kontakti sa iste zavojnice bipolarnog motora.



Slika 15: A4988 driver za koračni motor

Nakon što se paralelno spoji voltmetar na potencijometar od drivera namještava se V_{ref} prema [6] :

$$\text{Maksimalna struja} = \frac{V_{ref}}{8 \cdot R_{cs}}$$

gdje je R_{cs} otpornik ugrađen na A4988 driveru i iznosi $0,068 \Omega$. Pomoću ove jednadžbe vidimo da za faznu struju od 1 A trebamo imati V_{ref} od 0,540 V.

Električna shema spajanja svih navedenih komponenti se nalazi u prilogu.

6 Program

6.1 Diskretni PID algoritam

Kontinuirani PID regulator paralelne strukture je oblika:

$$y(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

Prelazimo na diskretni oblik kako bismo implementirali regulator na mikroupravljaču:

$$t = kT \quad , \quad k = 1, 2, 3, 4 \dots$$

$$\int_0^t e(t) dt \quad \rightarrow \quad T \sum_{i=0}^{k-1} e(i)$$

$$\frac{de(t)}{dt} \quad \rightarrow \quad \frac{e(k) - e(k-1)}{T}$$

Pa diskretni oblik ispada:

$$y(k) = K_p e(k) + K_i T \sum_{i=0}^{k-1} e(i) dt + K_d \frac{e(k) - e(k-1)}{T}$$

Sada kada imamo diskretni oblik možemo prijeći na njegov rekurzivni oblik:

$$y(k) = K_p e(k-1) + K_i T \sum_{i=0}^{k-2} e(i) dt + K_d \frac{e(k-1) - e(k-2)}{T}$$

$$y(k) - y(k-1) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$$

Ukoliko rekurzivni oblik svedemo na minimalni broj računskih operacija (3):

$$y(k) = y(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$$

$$q_0 = K_p + \frac{K_d}{T}$$

$$q_1 = K_p + K_i T - 2 \frac{K_d}{T}$$

$$q_2 = \frac{K_d}{T}$$

Nedostatak ovog pristupa je gubljenje utjecaja pojedinih članova PID regulatora.

6.2 PID regulator u diskretnoj računalnoj izvedbi

- Regulacijska pogreška $\rightarrow e = w - x$
- P djelovanje $\rightarrow y_p = K_p \cdot e$
- I djelovanje $\rightarrow y_i = y_i + K_i \cdot T \cdot e$
- D djelovanje $\rightarrow y_d = \frac{K_d}{T} \cdot (e - e_b)$
- Početni uvjeti $\rightarrow y_i = 0$, $e_b = e$
- PID djelovanje $\rightarrow y = y_p + y_i + y_d$
- Period izvršavanja algoritma $\rightarrow T$

6.3 Komplementarni filter

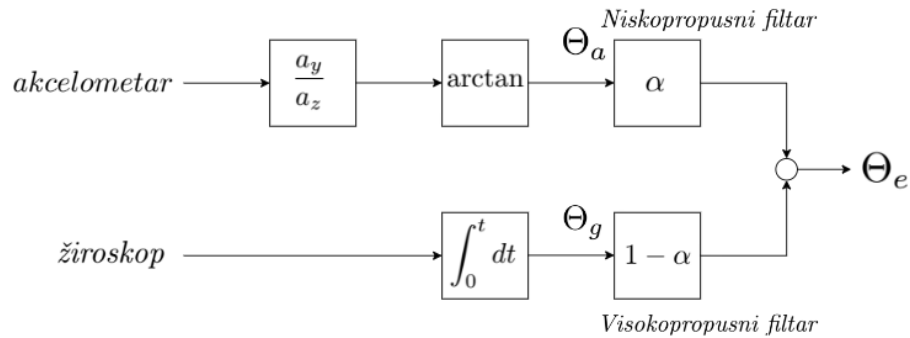
MPU6050 daje očitavanja žiroskopa i akcelometra. Kako bismo dobili poziciju iz žiroskopa integriramo njegova očitavanja u vremenu:

$$nagib_{\text{žiroskop}} = \int_0^t \omega dt$$

Dok za određivanje nagiba pomoću akcelometra možemo koristiti kotangens ubrzanja u smjeru osi grede i smjeru z, odnosno smjeru gravitacije Zemlje:

$$nagib_{\text{akcelometar}} = \arctan \frac{a_y}{a_z}$$

Problem za integriranjem očitavanja žiroskopa je gomilanje greške integracije kroz vrijeme. Problem za očitanjima akcelometra je što su jako osjetljiva na vibracije pri kojima naglo mijenjaju vrijednosti. Zbog tih problema nije prikladno koristiti niti jednu od ovih metoda samostalno za određivanje nagiba. Rješenje je implementacija nekog oblika kombinacije ovih podataka. U teoriji kontrole postoje razne metode za ovakve probleme, npr. razne varijacije Kalmanovog filtra, Mahony filter, Fuzzy logika itd. U ovom radu se koristi komplementarni filter za kombinaciju podataka sa žiroskopa i akcelometra. Komplementarni filter primjenjuje niskopropusni filter na signal sa visokofrekventnim smetnjama (akcelometar) te visokopropusni filter na signal sa niskofrekventnim smetnjama (žiroskop). Na taj način se otklanja spora greška žiroskopa i nagla promjena očitavanja akcelometra.



Slika 16: Princip komplementarnog filtra

Niskopropusni filter 1. reda u Laplaceovoj domeni je oblika:

$$\alpha = \frac{1}{\tau s + 1}$$

Visokopropusni filter 1. reda u Laplaceovoj domeni je oblika:

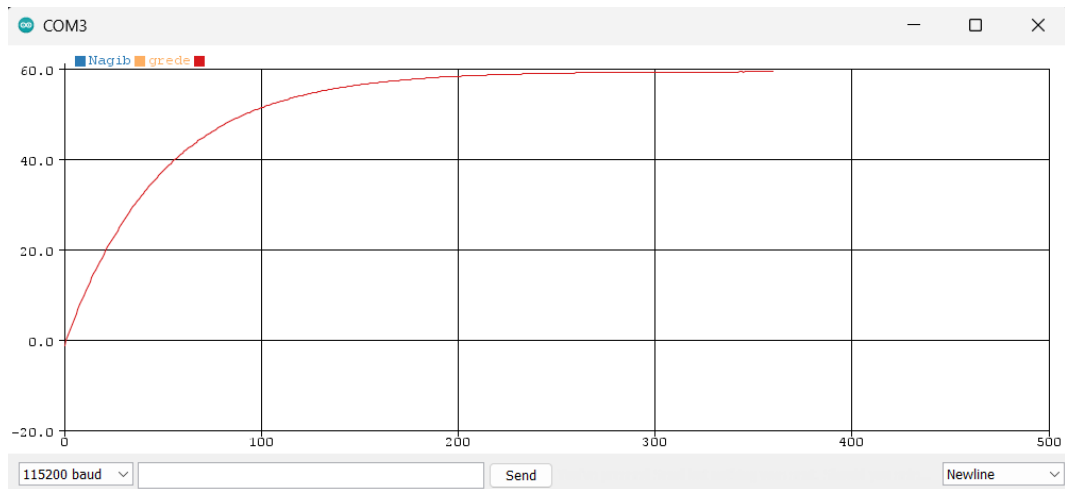
$$1 - \alpha = \frac{\tau s}{\tau s + 1}$$

Zbrajanjem izlaza filtara dobivamo izraz za izlaz komplementarnog filtra:

$$\begin{aligned}\Theta_e &= \alpha \Theta_a + (1 - \alpha) \Theta_g \\ \Theta_e &= \frac{1}{\tau s + 1} \arctan \frac{a_y}{a_z} + \frac{\tau s}{\tau s + 1} \omega \frac{1}{s}\end{aligned}$$

Za rad na mikroupravljaču prebacujemo izraz za komplementarni filter u njegov diskretni oblik:

$$\begin{aligned}\Theta_e &= \frac{\Delta t}{\tau + \Delta t} \arctan \frac{a_y}{a_z} + \frac{\tau}{\tau + \Delta t} (\Theta_e + \omega \Delta t) \\ \frac{\Theta_a}{\Theta_g} &= \frac{\Delta t}{\tau} = \alpha\end{aligned}$$

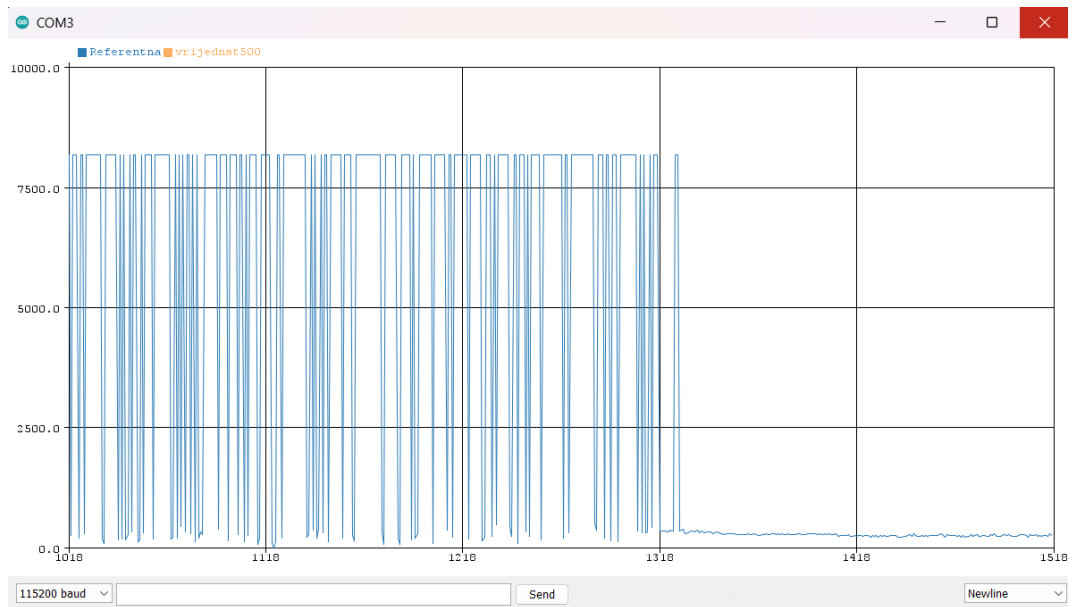


Slika 17: Odziv komplementarnog filtra

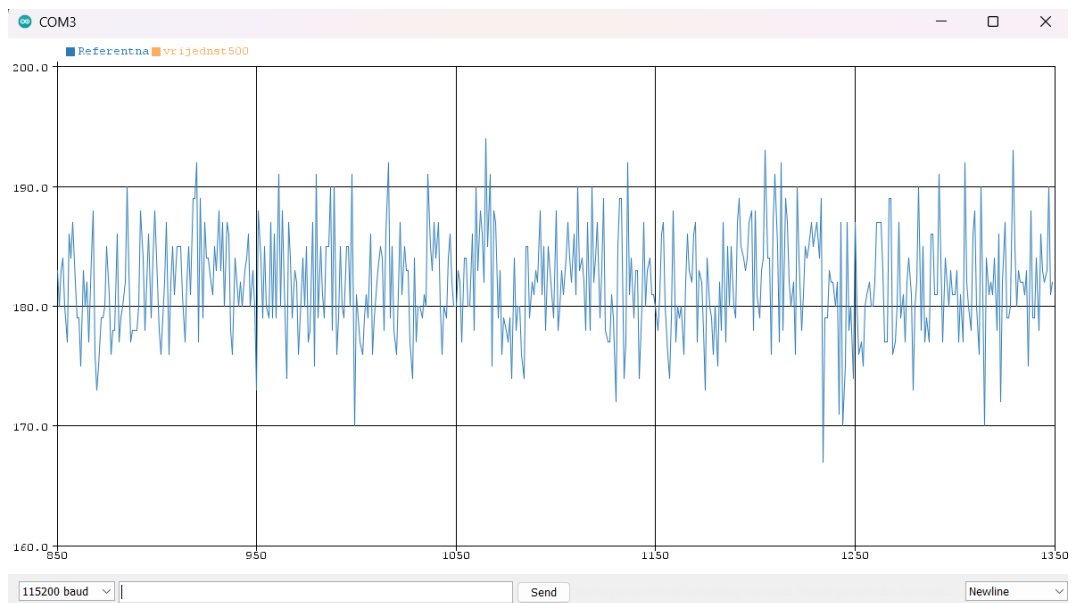
Na slici iznad se vidi početni odziv komplementarnog filtra. Na ordinati su stupnjevi a na apscisi sekunde pomnožene faktorom 100. U početnom položaju je greda nagnuta oko 59 stupnjeva te se za dobivanje ovog odziva ne mijenja. Iz slike se vidi da filtru pri početku rada treba oko 4 sekunde da dostigne stacionarno očitavanje, nakon čega relativno brzo i točno prati pravu vrijednost nagiba.

6.4 Testiranje VL53L0X

Prema [4] VL53L0X ima raspon mjerenja od 10 do 2000 mm. Kroz više izvora na internetu sam pročitao da nije pouzdan za mjerenja iznad 1000 mm, ali testiranjem senzora ispada da dobivam pouzdane rezultate samo za duljine do 500 mm. Na slici 18 se može vidjeti da na većim udaljenostima senzor naglo daje rezultat mjerenja 8000 mm. Pri mjerenju izlaza kod 200mm udaljenosti kuglice dobivaju se mjerenja kao na slici 19. Vidi se da u tom slučaju izlaz varira oko ± 10 mm premda kuglica miruje.



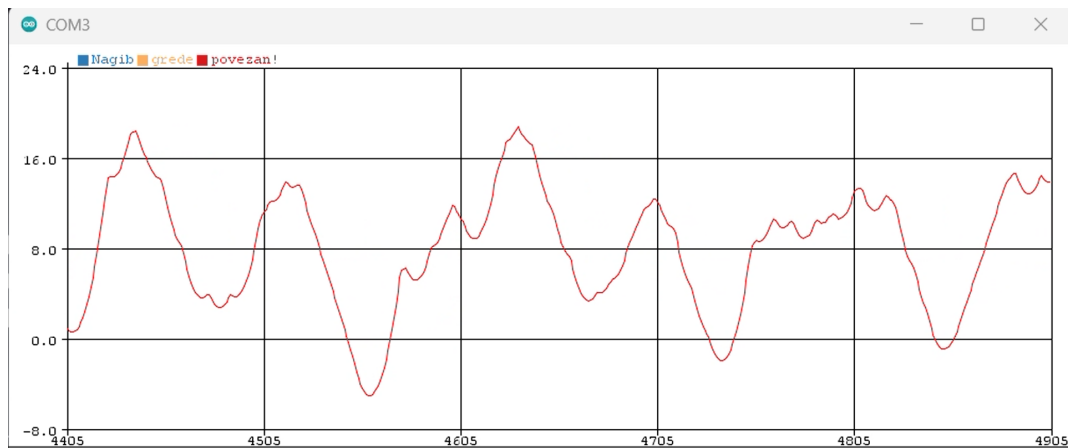
Slika 18: Neprecizna mjerenja VL53L0X



Slika 19: Mjerenja kod pozicije kuglice oko 200 mm od VL53L0X senzora

6.5 Odziv sustava

Na fizičkoj maketi je teško postignuti stabilni odziv zbog nesavršenosti mjerenja pozicije te elastičnosti užeta koja spaja motor i gredu. Pošto je sam proces već inercijski i samim time integralan (ima određeno kašnjenje) prikladno je koristiti PD regulator. Prikazan odziv je dobiven za pojačanja $K_p = 50$, $K_i = 0$, $K_d = 10000$. Uvođenje integralnog pojačanja bi previše usporilo odziv sustava i učinio ga potpuno nestabilnim. Na ordinati je nagib grede u stupnjevima, a na apscisi vrijeme u sekundama pomnoženim sa faktorom 100.



Slika 20: Odziv prave makete

6.6 Arduino kod

```
1 /*
2  * Program koji na temelju podataka sa VL53L0X daljinomjera
3  * i MPU6050 žiroskopa/akcelometra te diskretnog PID algoritma
4  * mijenja brzinu okretanja koračnog motora. Kut sa MPU6050 se
5  * očitava koristeći komplementarni filter. Koračni motor
6  * koristi 1/16 microstepping. Kako bi precizno implementirali
7  * korake u mikrosekundama koračnog motora koriti se prekidna
8  * rutina bazirana na Timer1 timeru.
9  *
10 * Autor: Toni Bešenić
11 * Datum: 15.09.2024.
12 */
13
14 //Biblioteke
15 #include <LiquidCrystal_I2C.h>
16 #include <Wire.h>
17 #include <VL53L0X.h>
18 #include <MPU6050.h>
19
20 //Pinovi enkodera
21 #define outputA 49
22 #define outputB 47
23 #define outputC 39
24 #define outputD 37
25 #define outputE 29
26 #define outputF 27
27
28 LiquidCrystal_I2C lcd(0x27, 20, 4);
29 VL53L0X sensor;
30 MPU6050 mpu;
31
32 //Varijable enkodera
33 int counter1 = 0;
34 int counter2 = 0;
```

```
35 int counter3 = 0;
36 int aState1;
37 int aState2;
38 int aState3;
39 int aLastState1;
40 int aLastState2;
41 int aLastState3;
42
43 //Varijable za MPU6050 i komplementarni filter
44 const float alpha = 0.98;
45 float accAngleX, accAngleY;
46 float gyroAngleX, gyroAngleY;
47 float roll;
48 float nagib = 0;
49 unsigned long pt, tt; // Prošlo vrijeme i trenutno vrijeme
50 float dt;
51
52 //Varijabla za VL53L0X
53 volatile unsigned long lastSensorTriggerTime = 0;
54 volatile bool sensorDataReady = false;
55 unsigned long sensorInterval = 300;
56 int udaljenost = 0; // distance variable
57
58 // Pinovi NEMA 17 koračnog motora
59 #define stepPin 3
60 #define dirPin 2
61 int korekcija = 3000; //Početna vrijednost brzine motora
62 volatile int pinState = LOW; //Varijabla za ISR
63
64 // Varijable za PID algoritam
65 double e, y, y_p, y_d;
66 double y_i = 0;
67 double eb = 0;
68 const int T = 5; //Period izvršenja PID algoritma u mikrosekundama
69 //Pojačanja PID regulatora
70 int K_p = 20;
```



```
71 int K_i = 0;
72 int K_d = 6500;
73
74 //Postavljanje tražene pozicije
75 int trazena_pozicija = 300;
76
77 //ZA millis() umijesto delaya
78 unsigned long previousMillisPID = 0;
79 unsigned long previousMicrosNEMA = 0;
80
81 void setup() {
82   pinMode(outputA, INPUT);
83   pinMode(outputB, INPUT);
84
85   pinMode(stepPin, OUTPUT);
86   pinMode(dirPin, OUTPUT);
87
88   // Postavlja Timer1 za prekidnu rutinu
89   cli(); // Onemogućava globalne prekide
90
91   // Stavljaja Timer1 u CTC način rada (Clear Timer on Compare Match)
92   TCCR1A = 0; // Normalni rad, bez PWM
93   TCCR1B = 0;
94
95   // Pretvara vrijeme u mikrosekundama iz varijable korekcija u tikove od
   Timer1
96   OCR1A = (16 * korekcija) - 1;
97
98   // Počinje Timer1 bez prescalera
99   TCCR1B |= (1 << WGM12) | (1 << CS10);
100
101   TIMSK1 |= (1 << OCIE1A);
102
103   sei(); // Ponovo omogućava globalni prekid
104
105   //Podešavanje početnog izgleda LCD ekrana
```

```
106  lcd.init();
107  lcd.backlight();
108  lcd.setCursor(0, 0);
109  lcd.print("Odabir_pojacanja:");
110  lcd.setCursor(0, 1);
111  lcd.print("P_");
112  lcd.setCursor(0, 2);
113  lcd.print("I_");
114  lcd.setCursor(0, 3);
115  lcd.print("D_");
116
117  Serial.begin(115200);
118  //Inicijalizacija VL53L0X daljinomjera.
119  Wire.begin();
120  sensor.init();
121  sensor.setTimeout(500);
122  sensor.startContinuous(); //Unutar zagrada se može zadati vrijeme očitavanja
    u ms.
123
124  //Inicijalizacija i provjera veze MPU6050
125  mpu.initialize();
126  if (!mpu.testConnection()) {
127      Serial.println("MPU6050_nije_povezan!");
128      while (1);
129  } else {
130      Serial.println("MPU6050_je_povezan!");
131  }
132  pt = millis();
133
134  // Očitavanje početnih vrijednosti enkodera
135  aLastState1 = digitalRead(outputA);
136  aLastState2 = digitalRead(outputC);
137  aLastState3 = digitalRead(outputE);
138 }
139
140 void loop() {
```

```
141 //citajEnkoder();
142 MPU6050_kut();
143 ocitanjeDaljine();
144 PID();
145 // Ovaj dio koda ograničava vrijednosti izlaza regulatora na
146 // +-1000, što služi za ograničavanje brzine motora.
147 const int ogranicenje = 1000;
148 if (y > ogranicenje){
149     y = ogranicenje;
150 }
151 if (y < -ogranicenje){
152     y = -ogranicenje;
153 }
154 if (y >= 0){
155     digitalWrite(dirPin,LOW);
156 }
157 if (y < 0){
158     digitalWrite(dirPin,HIGH);
159 }
160 // Ažuriranje brzine motora
161 // Pretvaranje izlaza regulatora u brzinu koračnog motora pomoću
162 // map() funkcije.
163 korekcija = map(abs(y), 0, 1000, 10000, 80);
164 //Postavlja novi iznos korekcije za Timer1
165 OCR1A = (16 * korekcija) - 1;
166
167 // Ovaj dio koda ograničava brzinu motora ako se previše naginje.
168 // Služi za zaštitu u slučaju prevelikog prebačaja makete.
169 if ((nagib <= -2) && (digitalRead(dirPin)==LOW)){
170     //Serial.println("PREVELIK NAGIB !!!!!!!!!!!");
171     korekcija = 300;
172     OCR1A = (16 * korekcija) - 1;
173 }
174 Serial.print("IZLAZ_REGULATORA:_");
175 Serial.println(y);
176 Serial.print("Brzina_koračnog_motora:_");
```

```
177 Serial.println(korekcija);
178 Serial.print("Referenca_udaljenosti");
179 Serial.println(trazena_pozicija);
180 Serial.println("Udaljenost_kuglice");
181 Serial.print(udaljenost);
182 }
183
184 void citajEnkoder() {
185 // Funkcija koja očitava promjene enkodera i prikazuje ih na LCD ekranu
186 // Na početku se definira maksimalna i minimalna vrijednost koju pojačanja
    mogu
187 // poprimiti i korak koji se izvršni pri jednom impulsu.
188 int min_ = 0;
189 int max_ = 1200;
190 int korak = 5; //Zbog 2 impulsa po jednom okretaju se izvršavaju dva koraka
    .
191 aState1 = digitalRead(outputA);
192 aState2 = digitalRead(outputC);
193 aState3 = digitalRead(outputE);
194 //Ovaj dio koda provjerava je li došlo do promjene izlaza od prošlog stanja
    te smanjuje
195 //ili povećava iznos ovisno o smjeru rotacije. Također uspoređuje
    vrijednost sa minimalnom
196 //i maksimalnom te ih ograničava.
197 if (aState1 != aLastState1) {
198     if (digitalRead(outputB) != aState1) {
199         counter1 += korak;
200         if (counter1 > max_) {
201             counter1 = max_;
202         }
203     } else {
204         counter1 -= korak;
205         if (counter1 < min_) {
206             counter1 = min_;
207         }
208     }
```

```
209     updateLCD(4, 1, counter1);
210 }
211 aLastState1 = aState1;
212 if (aState2 != aLastState2) {
213     if (digitalRead(outputD) != aState2) {
214         counter2 += korak;
215         if (counter2 > max_) {
216             counter2 = max_;
217         }
218     } else {
219         counter2 -= korak;
220         if (counter2 < min_) {
221             counter2 = min_;
222         }
223     }
224     updateLCD(4, 2, counter2);
225 }
226 aLastState2 = aState2;
227 if (aState3 != aLastState3) {
228     if (digitalRead(outputF) != aState3) {
229         counter3 += korak;
230         if (counter3 > max_) {
231             counter3 = max_;
232         }
233     } else {
234         counter3 -= korak;
235         if (counter3 < min_) {
236             counter3 = min_;
237         }
238     }
239     updateLCD(4, 3, counter3);
240 }
241 aLastState3 = aState3;
242 }
243
244 void updateLCD(int col, int row, int value) {
```

```
245 //Funkcija koja briše dio ekrana sa vrijednostima pojačanja svaki put
246 //kad se one promijene. Bez ove funkcije nastaju greške na ekranu kada se
247 //mijenja broj znameki broja pojačanja.
248 lcd.setCursor(col, row);
249 lcd.print("    ");
250 lcd.setCursor(col, row);
251 lcd.print(value);
252 }
253
254 void ocitanjeDaljine(){
255 //Funkcija za očitavanje daljine sa VL53L0X daljinomjera,
256 //u komentarima je potreban kod ako se u void setup() odabere
257 //konkretno vrijeme očitavanja podatka sa senzora.
258 // unsigned long asd = micros();
259 udaljenost = sensor.readRangeContinuousMillimeters();
260 //int distance =sensor.startContinuous(100);
261 //distance = distance;
262 // Serial.print("Pozicija kuglice: ");
263 // Serial.print(udaljenost);
264 // Serial.print("mm");
265 // if (sensor.timeoutOccurred()) {
266 //   Serial.print(" TIMEOUT");
267 // }
268 // Serial.println();
269 // unsigned long dsa = micros() - asd;
270 // asd = dsa;
271 // Serial.print("Vrijeme potrebno za očitavanje daljine: ");
272 // Serial.println(dsa);
273 }
274
275 void MPU6050_kut() {
276 //unsigned long prd = micros();
277 //Funkcija koja daje kut nagiba grede koristeći komplementarni
278 //filter koji kombinira podatke sa žiroskopa i akcelometra.
279 tt = millis();
280 dt = (tt - pt) / 1000.0; // Pretvorba u sekunde
```

```
281 pt = tt;
282
283 // Direktna očitavanja žiroskopa i akcelometra
284 int16_t ax, ay, az;
285 int16_t gx, gy, gz;
286 mpu.getAcceleration(&ax, &ay, &az);
287 mpu.getRotation(&gx, &gy, &gz);
288
289 // Prilagodba čistih očitavanja akcelometra
290 float accY = ay / 16384.0; // 16384 je faktor za osjetljivost +-2g
291 float accZ = az / 16384.0;
292
293 // Prilagodba čistih očitavanja žiroskopa
294 float gyroX = gx / 131.0; // 131 faktor za osjetljivost +-250 stupnjeva/
    sekundi
295
296 // Izračun kuteva akcelometra
297 accAngleX = atan2(accY, accZ) * 180 / PI;
298
299 // Integracija očitavanja žiroskopa
300 gyroAngleX += gyroX * dt;
301
302 // Komplementarni filter
303 roll = alpha * (roll + gyroX * dt) + (1 - alpha) * accAngleX;
304 nagib = -1 * roll - 2.3; //2.3 je offset
305 // Serial.print("Nagib grede: ");
306 // Serial.println(nagib);
307 // unsigned long drt = micros() - prd;
308 // Serial.print("Vrijeme za očitavanje nagiba: ");
309 // Serial.println(drt);
310 // drt = prd;
311 }
312
313 void PID() {
314 //PID algoritam
315 unsigned long currentMillisPID = millis();
```

```
316  if (currentMillisPID - previousMillisPID >= T){
317      previousMillisPID = currentMillisPID;
318      e = trazena_pozicija - udaljenost;
319      y_p = K_p * e;
320      y_i = y_i + K_i * T * e;
321      y_d = (K_d/T) * (e-eb);
322      y = y_p + y_i + y_d;
323      eb = e;
324  }
325 }
326
327 ISR(TIMER1_COMPA_vect) {
328     // Mijenja stanje stepPin-a
329     pinState = !pinState;
330     digitalWrite(stepPin, pinState);
331 }
```


7 Zaključak

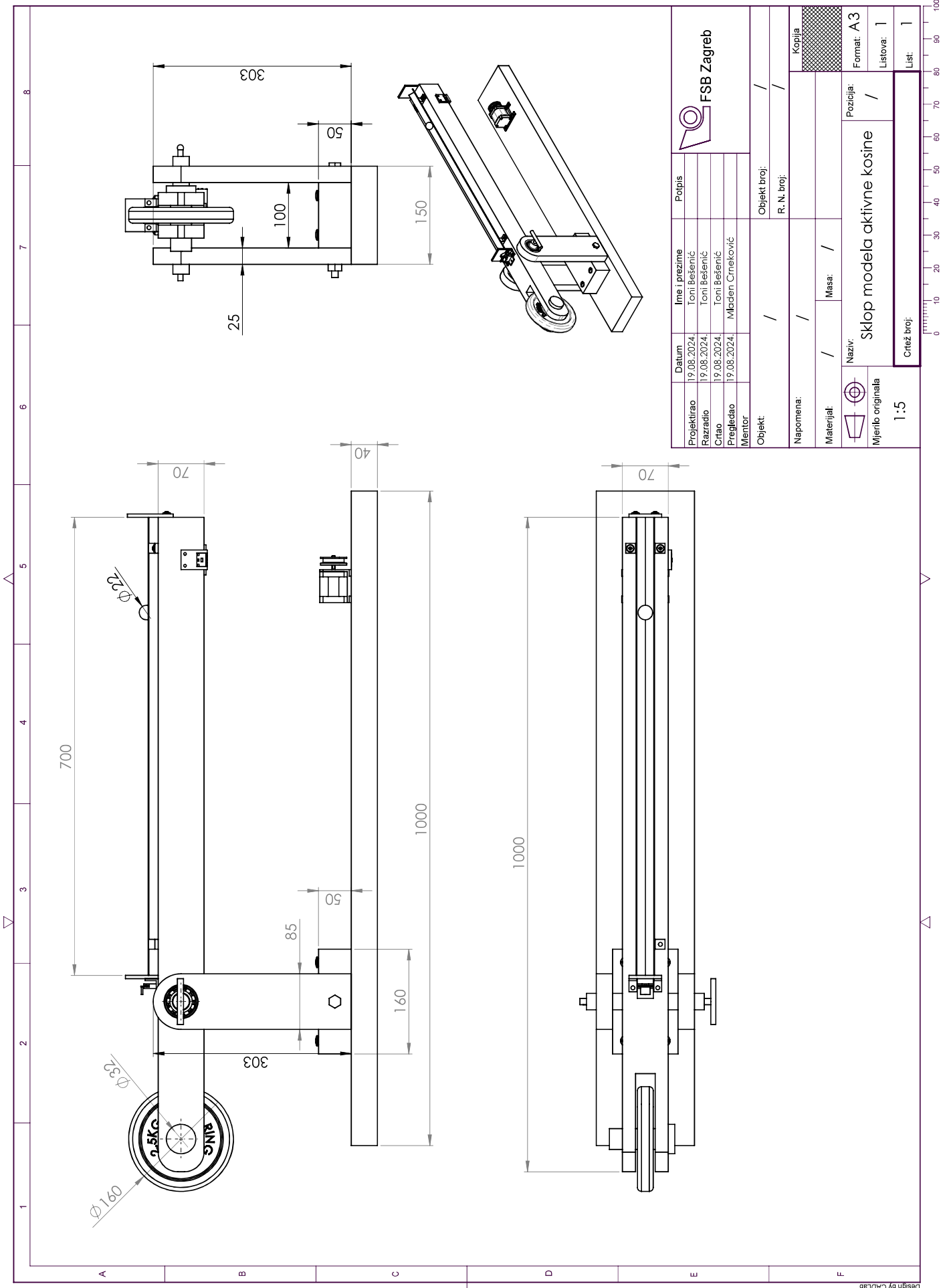
U radu smo opisali primjere i probleme kod regulacija slabo prigušenih inercijskih sustava. Izveli smo matematički model za model kuglice na kosini te uspješno konstruirali i programirali sam model. Na modelu nije postignuta potpuna stabilizacija, uglavnom zbog netočnosti samog senzora mjerenja položaja te pojednostavljenja kod upravljanja. Pri mijenjanju pojačanja modela se jasno vide utjecaji pojedinih članova na odziv modela. Maketa je funkcionalna te bi se mogla koristiti za demonstracije na vježbama iz automatike. Model bi se mogao doraditi nabavom boljeg senzora udaljenosti. Dodatno bi se na temelju matematičkog modela i simulacije u nekom programu mogla dobiti optimalna pojačanja regulatora koristeći jednu od metoda optimizacije, pa onda usporediti eksperimentalni i simulacijski odziv za ta pojačanja. Maketa je napravljena da bude modularna te se sa već fizički izvedenom maketom se može relativno jednostavno mijenjati upravljački algoritam ili uređaj.

Literatura

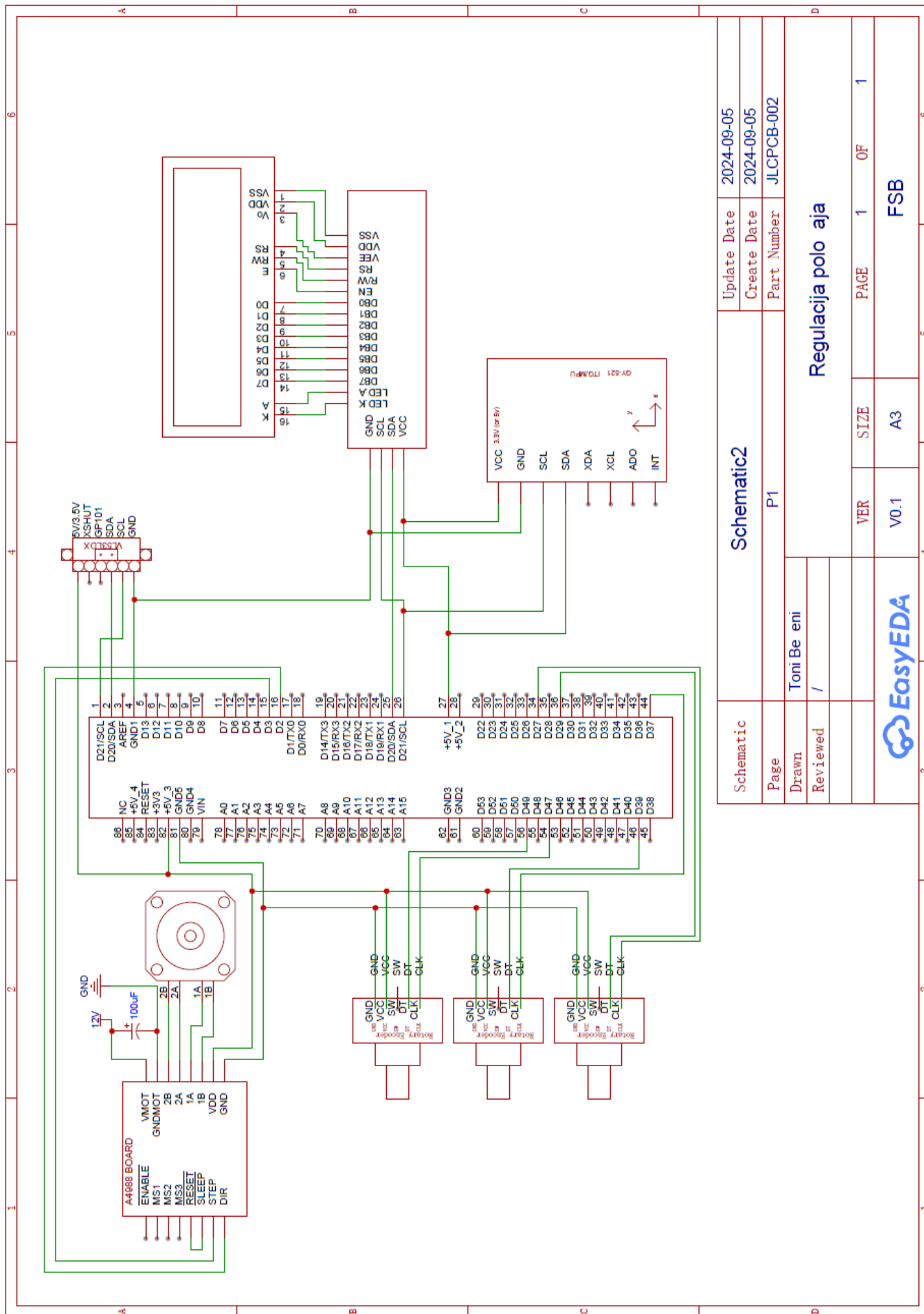
- [1] <https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf>.
- [2] T. Šurina and Ž. Cvirn. *Automatska regulacija*. Udžbenici Sveučilišta u Zagrebu. Školska knjiga, 1991.
- [3] D. Brozović. *Hrvatska enciklopedija*. Number v. 4 in Hrvatska enciklopedija. Leksikografski zavod "Miroslav Krleža", 1999.
- [4] <https://www.st.com/en/imaging-and-photonics-solutions/v15310x>.
- [5] <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [6] https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf.

Prilozi

- Sklopni crtež modela
- Električna shema modela
- Slika makete



Slika 21: Sklop modela aktivne kosine



Slika 22: Električna shema elektronike modela



Slika 23: Slika izvedene makete