Izrada makete prigušenog oscilatora s prikazom na pametnom telefonu

Gegić, Pavle

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:684843

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-14**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

IZRADA MAKETE PRIGUŠENOG OSCILATORA

Završni rad

Pavle Gegić

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 16.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Pavle Gegić
Studij, smjer:	Sveučilišni prijediplomski studij Elektrotehnika i informacijska
Mat. br. Pristupnika, godina upisa:	4368, 25.07.2017.
OIB Pristupnika:	12594268537
Mentor:	izv. prof. dr. sc. Ivan Aleksi
Sumentor:	dr. sc. Željka Mioković
Sumentor iz tvrtke:	
Naslov završnog rada:	Izrada makete prigušenog oscilatora s prikazom na pametnom telefonu
Znanstvena grana rada:	**Arhitektura računalnih sustava (zn. polje računarstvo)**
Zadatak završnog rad:	Zadatak ovog završnog rada je izraditi maketu prigušenog oscilatora primjenom Arduino mikroupravljačkog okruženja. Maketa prigušenog oscilatora predstavlja jednostavno njihalo. Koristi se kuglica koja visi na tankoj niti. Potrebno je mjeriti amplitudu kuta otklona koji se polako smanjuje tijekom vremena. Mjerne podatke koje treba detektirati je amplituda kuta otklona kuglice u određenom trenutku titranja. Mjerne
Prijedlog ocjene završnog rada:	Dobar (3)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 1 razina
Datum prijedloga ocjene od strane mentora:	16.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 25.09.2023.

Ime i prezime studenta:	Pavle Gegić
Studij:	Sveučilišni prijediplomski studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4368, 25.07.2017.
Turnitin podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Izrada makete prigušenog oscilatora s prikazom na pametnom telefonu**

izrađen pod vodstvom mentora izv. prof. dr. sc. Ivan Aleksi

i sumentora dr. sc. Željka Mioković

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD **1**
1.1. Zadatak završnog rada **1**
2. PREGLED PODRUČJA TEME **2**
2.1. Simulacija– Pendulum Lab 4
2.1.1. Rezultati mjerenja 5
2.2 Mjerni instrumenti **8**
2.2.1. XM132 8
2.2.2. HC-SR04 10
2.2.3. TF40 LiDAR 11
2.2.4. R-EVO-MINI-I2C 13
3. IZRADA MAKETE **15**
4. EKSPERIMENTI I REZULTATI **16**
4.1. Arduino mega 2560 REV3 – izvođenje eksperimenta **16**
4.2. Arduino **19**
4.3. Python **20**
4.4. Rezultati mjerenja **21**
5. ZAKLJUČAK **25**
LITERATURA **26**
SAŽETAK **27**
ABSTRACT **27**
ŽIVOTOPIS **28**
Prilog 1 – upute za pokretanje programa **29**
Prilog 2 – Arduino program **30**
Prilog 3 – Python program **38**
Prilog 4 – MS Excel tablica rezultata **42**

1. UVOD

Ovaj rad usmjeren je na praktičnu primjenu teorije prigušenih oscilatora putem izrade makete prigušenog oscilatora koja koristi senzor za očitavanje amplitude kuta otklona metalne kuglice. Prigušeni oscilatori predstavljaju ključnu temu u mnogim granama fizike, inženjeringa i prirodnih znanosti. Ovi sustavi se koriste za dublje razumijevanje dinamičkih procesa i fenomena, često dajući dublji uvid u ponašanje sustava u stvarnim uvjetima. Prigušeni oscilatori su dinamički sustavi koji se sastoje od mase koja je pod utjecajem tangencijalnih sila i prigušujuće sile koja je posljedica trenja, zračnog otpora ili drugih faktora koji usporavaju oscilaciju. Ključna karakteristika prigušenih oscilatora je da gube energiju s vremenom i da usporavaju svoje oscilacije, suprotno od idealnog neprigušenog oscilatora koji bi beskonačno oscilirao bez gubitaka energije.

U drugom poglavlju nalazi se pregled literature, teorijska pozadina prigušenog oscilatora, on-line virtualna simulacija, te opisu i specifikacijama mjernih uređaja. Treće poglavlje posvećeno je izradi makete prigušenog oscilatora. Tema četvrtog poglavlja je eksperimentalni dio i rezultati mjerenja. Kroz ovaj proces, bit će moguće bolje razumjeti složene dinamičke sustave i primijeniti teorijska znanja na konkretnom eksperimentalnom modelu.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je izraditi maketu prigušenog oscilatora primjenom Arduino mikroupravljačkog okruženja. Maketa prigušenog oscilatora predstavlja jednostavno njihalo. Koristi se kuglica koja visi na tankoj niti. Potrebno je mjeriti amplitudu kuta otklona koji se polako smanjuje tijekom vremena. Mjerne podatke koje treba detektirati je amplituda kuta otklona kuglice u određenom trenutku titranja. Mjerne podatke u obliku ovisnosti amplitude kuta otklona kuglice u vremenu potrebno prikazati u tabličnom i grafičkom obliku.

2. PREGLED PODRUČJA TEME

U literaturi [1] autor analizira titrajne sustave i utjecaj vanjskih faktora na gibanje, istražujući njihove kompleksne dinamičke interakcije i pružajući dublje uvide u mehanizme koji leže u osnovi ovih fenomena.

Titrajni sustavi u prirodi obično se neće gibati beskonačno dugo (zbog utjecaja vanjskih sila), već će se s vremenom amplituda titranja smanjivati, a potom će se zaustaviti. U stvarnim titrajnim sustavima dio mehaničke energije titranja troši se na savladavanje vanjskih disipativnih sila. Na primjeru kod njihala, dio energije troši se na svladavanje otpora zraka, dok kod tijela koje je obješeno na opruzi, dio energije troši se na zagrijavanje opruge. Titranje u kojem se amplituda postupno smanjuje tijekom vremena naziva se prigušeno titranje. Prigušeno titranje može se predočiti ako tijelo koje titra na opruzi uronimo u neku viskoznu tekućinu, kao što je ulje. Sila koja djeluje tijekom ovakvog titranja će biti razmjerna brzini gibanja tijela i djelovat će u suprotnom smjeru od smjera brzine, gdje će b konstanta trenja (prigušenja). Predznak "-" će ukazivati da su sila trenja i brzina u suprotnom smjeru, te će biti iskazana formulom:

$$F_{tr} = -bv = -b\frac{dx}{dt} \tag{1-1}$$

Jednadžba gibanja (II. Newtonov zakon) za prigušeno titranje glasi:

$$ma = F_{opr} + F_{tr} \tag{1-2}$$

$$m\frac{d^2x}{dt^2} = -kx - b\frac{dx}{dt} \tag{1-3}$$

Preuređivanjem jednadžbe (1-3) dobiva se jednadžba prigušenog titranja:

$$\frac{d^2x}{dt^2} + 2\delta\frac{dx}{dt} + \omega_0 = 0 \tag{1-4}$$

ω_o je vlastita kružna frekvencija titranja neprigušenog harmonijskog oscilatora, a δ faktor prigušenja.

Bez detaljnijeg matematičkog izvoda, rješenje homogene linearne diferencijalne jednadžbe (1-4), u slučaju vrlo slabih prigušenja ($\delta < \omega_o$), ima oblik:

$$\theta(t) = \theta e^{-\delta t}\sin(\omega t + \varphi_0) \tag{1-5}$$

ω predstavlja kružnu frekvenciju prigušenog oscilatora.

Amplituda titranja:

$$\theta(t) = \theta e^{-\delta t} \tag{1-6}$$

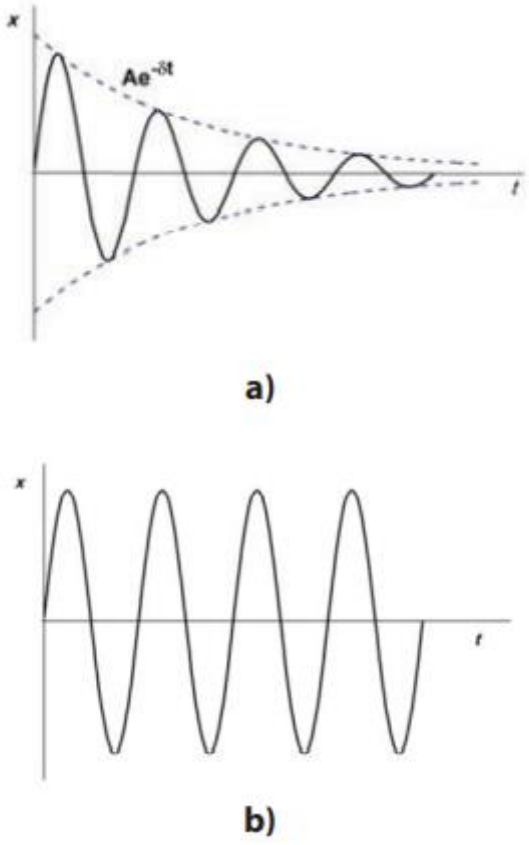
opada eksponencijalno s vremenom. Što je faktor prigušenja veći, to se amplituda brže smanjuje (slika 1.1.).

Omjer dviju susjednih amplituda kod prigušenog titranja, tj. amplituda koje se razlikuju za period T uvijek je stalan i iznosi:

$$\frac{\theta(t)}{\theta(t+T)} = e^{\delta t} \tag{1-7}$$

Logaritamski dekrement titranja glasi:

$$\lambda = \ln\left(\frac{\theta(t)}{\theta(t+T)}\right) = \delta T \tag{1-8}$$



Slika 2.1. Usporedba dvaju titrajnih sustava; ovisnost elongacije x o vremenu t za:

a) prigušeno titranje

b) neprigušeno titranje

Slika 2.1. prikazuje usporedbu dvaju titrajnih sustava i ovisnost elongacije o vremenu za prigušeno i neprigušeno titranje.

Logaritamski dekrement titranja λ i period titranja T nekog titrajnog sustava mogu se lako izmjeriti pa se, pomoću relacije (1-8), može odrediti δ faktora prigušenja, odnosno b=2mδ konstanta trenja. Prigušeni titrajni sustav može se, također, opisati tzv. Q-faktorom, odnosno faktorom dobrote ili kvalitete titrajnog sustava koji je određen relacijom:

$$Q = \frac{\pi}{\lambda} = \frac{\pi}{\delta T} = \frac{\omega_0}{2\delta} \tag{1-9}$$

ω_o je vlastita frekvencija neprigušenog titranja, a λ logaritamski dekrement titranja dan relacijom (1-8). Što je faktor dobrote veći, prigušenje je manje i manji je gubitak energije iz titrajnog sustava. U dobrim mehaničkim titrajnim sustavima Q-faktor je reda veličine 10^3 do 10^5.

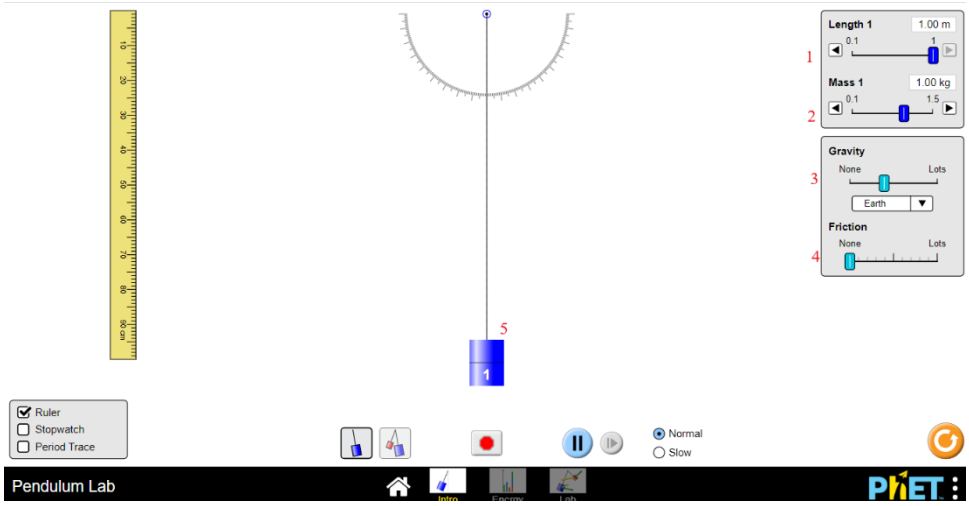
Zakoni titranja harmonijskog oscilatora primjenjuju se na proučavanje titranja laboratorijskih njihala kao stvarnih mehaničkih titrajnih sustava. U svakodnevnom životu njihalo predstavlja tijelo obješeno o uže tako da mu se centar mase nalazi ispod ovjesišta. Kada se takvo tijelo otkloni iz ravnotežnog položaja pojavljuje se moment sile koji ga vraća prema ravnotežnom položaju. Zbog djelovanja ovog momenta sile tijelo se počinje gibati periodično oko ravnotežnog položaja i ovo gibanje ima sve osobine harmonijskog titranja.

Autor u literaturi [2] izveo je eksperiment prigušenog oscilatora velikim plastičnim predmetom koji je ovješen o kratku nit, gdje se senzor udaljenosti nalazi s jedne strane kako bi pratio i zabilježio elongaciju u svakom trenutku. Ovaj konkretni eksperiment pruža dobar temelj za daljnje istraživanje. S obzirom na naglasak na što manji utjecaj vanjskih sila, bit će potrebno koristiti slične postavke, kao što je zamjena velikog predmeta manjom kuglicom i duljom niti.

Autori u literaturi [3] kreirali su on-line virtualni laboratorij koji pruža izvođenje eksperimenta i uvid u promjene kuta otklona kuglice prigušenog oscilatora.

2.1. Simulacija– Pendulum Lab

Prilikom pristupa virtualnom laboratoriju putem reference [3], prva radnja koju je potrebno izvesti jest odabrati opciju "Play", a nakon toga odabrati "Intro". Ovim koracima otvara se mogućnost pristupa simulaciji, što je prikazano na Slici 2.2.



Slika 2.2. Simulacija Pendulum Lab

U virtualnom laboratoriju pruža se širok spektar mogućnosti za kontroliranje različitih parametara koji značajno utječu na elongaciju titranja prigušenog oscilatora. Ti parametri uključuju:

1. Duljinu niti (označena kao 1): Ova opcija omogućava mijenjanje duljine niti na kojoj je ovješen predmet.
2. Masu predmeta (označena kao 2): Kontrola nad masom objekta koji titra omogućuje proučavanje kako različite mase utječu na ponašanje prigušenog oscilatora.
3. Gravitacija (označena kao 3): Promjenom vrijednosti gravitacije moguće je simulirati kako bi se prigušeni oscilator ponašao u različitim okruženjima s različitim gravitacijskim ubrzanjima.
4. 10 razina sile trenja (označeno kao 4): Sile trenja igraju ključnu ulogu u prigušivanju oscilacija. Ova opcija omogućava eksperimentiranje s različitim razinama trenja za bolje razumijevanje utjecaja prigušenja na oscilacije.

Simulacija se pokreće jednostavnim klikom na predmet označen brojem 5 te pomicanjem istog u stranu pod željenim kutem i puštanjem lijevog klika se pokreće simulacija.

2.1.1. Rezultati mjerenja

Za eksperimente korišteni su sljedeći podaci:

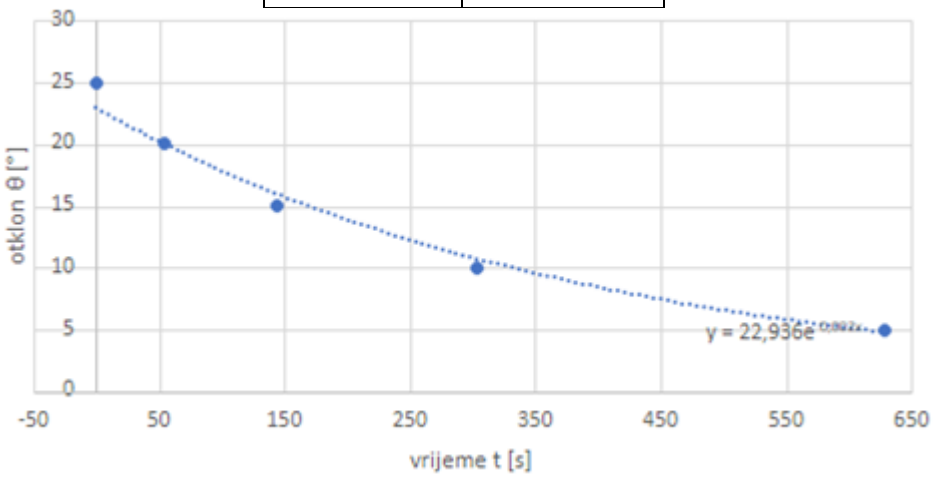
- Duljinu niti = 1m,
- Masa tijela = 1kg,

- Ubrzanje sile teže jednako je ubrzanju sile teže na Zemlji, g ≈ 9,81m/s^2.

Sila trenja se mijenja u eksperimentima i mjerenja se vrše na razinama 3,4 i 5.

Tablica 2.1. Rezultati mjerenja na 3. razini trenja:

vrijeme t[s]	otklon θ[°]
0	25
54	20
145	15
304	10
629	5



Slika 2.3. Dijagram amplitude kuta otklona na 3. razini trenja

U tablici 2.1. prikazani su rezultati mjerenja na 3. razini trenja, dok slika 2.3. prikazuje dijagram promjene amplitude kuta otklona.

Eksponencijalna jednadžba krivulje glasi:

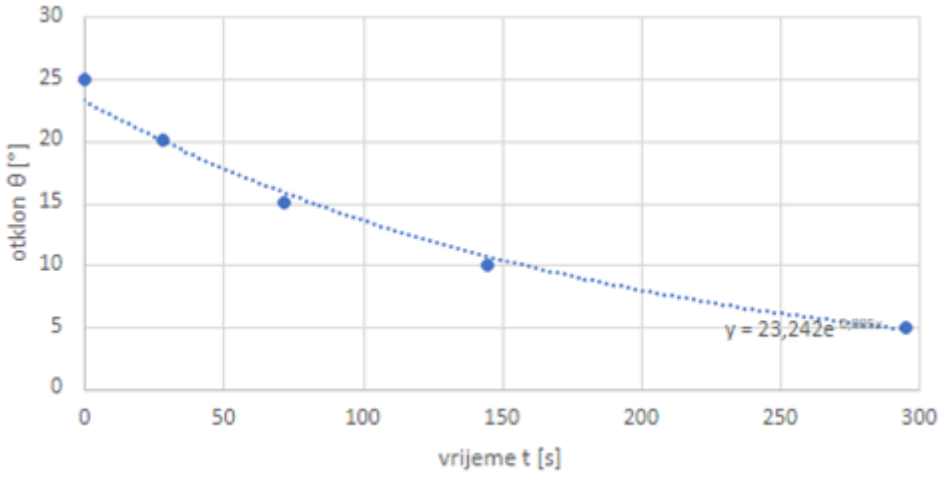
$$\theta(t) = 22{,}936e^{-0{,}002t} \qquad (2\text{-}1)$$

Prema relacijama (1-6) i (2-1) vidljivo je da amplituda iznosi θ=22,936°, dok faktor prigušenja iznosi δ=0,002s^{-1}.

Iz relacije (1-9) izračuna se faktor dobrote koji iznosi Q=785,4.

Tablica 2.2. Rezultati mjerenja na 4. razini trenja:

vrijeme t[s]	otklon θ[°]
0	25
28	20
72	15
145	10
295	5



Slika 2.4. Dijagram amplitude kuta otklona na 4. razini trenja

U tablici 2.2. prikazani su rezultati mjerenja na 4. razini trenja, dok slika 2.4. prikazuje dijagram promjene amplitude kuta otklona.

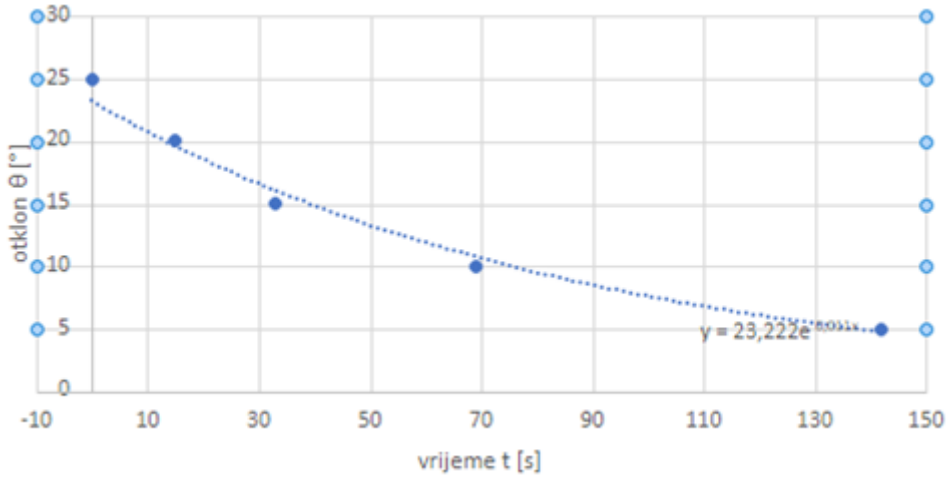
Eksponencijalna jednadžba krivulje glasi:

$$\theta(t) = 23{,}242e^{-0{,}005t} \qquad (2\text{-}2)$$

Prema relaciji (2-2), amplituda iznosi θ=23,242°, dok faktor prigušenja iznosi δ=0,005s^{-1}. Faktor dobrote iznosi Q=314,2.

Tablica 2.3. Rezultati mjerenja na 5. razini trenja:

vrijeme t[s]	otklon θ[°]
0	25
15	20
33	15
69	10
142	5



Slika 2.5. Dijagram amplitude kuta otklona na 5. razini trenja

U tablici 2.3. prikazani su rezultati mjerenja na 5. razini trenja, dok slika 2.5. prikazuje dijagram promjene amplitude kuta otklona.

Eksponencijalna jednadžba krivulje glasi:

$$\theta(t) = 23{,}222e^{-0{,}011t} \qquad (2\text{-}3)$$

Prema relaciji (2-3), amplituda iznosi θ=23,222°, dok faktor prigušenja iznosi δ=0,011s^{-1}. Faktor dobrote iznosi Q=142,8.

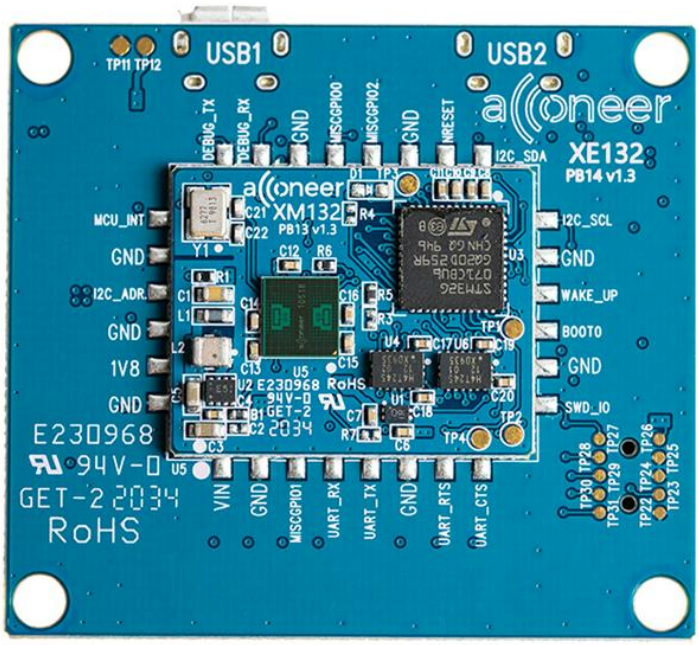
Iz priloženih rezultata vidljivo je kako porastom faktora prigušenja smanjuje se faktor dobrote, odnosno kvalitete titrajnog sustava.

2.2 Mjerni instrumenti

Prije pristupa izradi makete prigušenog oscilatora, istraživanje je provedeno kako bi se identificirali prikladni uređaji za mjerenje udaljenosti. Specifikacije su pažljivo uspoređivane kako bi se osigurala maksimalna preciznost rezultata. U nastavku, bit će predstavljeni mjerni instrumenti koji su odabrani za uži izbor.

2.2.1. XM132

Ulazni modul XM132, prema [4], temelji se na Acconeer A111 pulsnom koherentnom radaru (PCR) i STMicroelectronics M0+ MCU (STM32G071CBU6).



Slika 2.6. Uređaj XM132

Na slici 2.6. prikazan je ulazni modul XM132, čije su značajke:

- A111 60 GHz pulsni koherentni radar (PCR) s integriranim osnovnim pojasom, RF prednjim dijelom i Antena u paketu (AiP),
- 32-bitni ARM® CortexTM M0+ MCU (STM32G071CBU6) 64 MHz clk brzina, 128 kB Flash; 36 kB RAM-a,
- Dimenzije 25x20 mm s komponentama montiranim na gornjoj strani PCB-a i LGA lemnim jastučićima na donjoj strani PCB-a,
- Radni napon: 1,8-3,6V,
- Radna temperatura: -40° do 85°C,
- Uključeno na ploču za procjenu XE132 s podrškom za LH132 komplet objektiva u svrhu evaluacije,
- Eksterna podrška za sučelja - UART, I2C, GPIO, Reset,
- SWD/JTAG za programiranje i rješavanje problema.

Ulazni modul XM132 može se koristiti za precizno mjerenje udaljenosti, mjerenje razine u spremniku, mjerenje razine kante za otpad, detekcija zauzetosti parkirnog mjesta i detekcija pametne prisutnosti:

- Visoko precizna mjerenja udaljenosti s podesivom učestalošću ažuriranja.
- Mjeri apsolutni raspon do 7 m s apsolutnom točnošću u mm.

- Mogućnost prepoznavanja kretanja više objekata.

Jednostavna integracija:

- XM132 ulazni modul može se integrirati iza plastike ili stakla bez potrebe za fizički otvor blende.
- Podrška za LGA jastučiće za lemljenje za jednostavnu integraciju

2.2.2. HC-SR04

Odašiljač HC-SR04 (Slika 2.7.), prema [5], inicira emisiju osam pulsa ultrazvučnih valova frekvencije 40 KHz i istovremeno pokreće mjerač vremena. Ultrazvučni impulsi se šire prema vani sve dok ne naiđu na neki objekt, koji uzrokuje refleksiju vala natrag prema uređaju. Prijamnik za ultrazvuk registrira ovu reflektiranu valnu povratnicu i zaustavlja mjerač vremena. U zraku, ultrazvučni valovi putuju brzinom od 340 m/s. Na temelju broja impulsa zabilježenih od strane mjerača vremena, udaljenost između odašiljača i objekta može se izračunati. Formula za izračun ove udaljenosti se izražava kao:

$$D = C \cdot T \tag{2-4}$$

Relacija (2-4) poznata je kao formula za mjerenje vremena i brzine, gdje D predstavlja izmjerenu udaljenost, C je brzina širenja (brzina zvuka) u zraku, a T predstavlja vrijeme. U ovom specifičnom slučaju, T se dijeli s 2 jer obuhvaća dvostruko vremensko putovanje od odašiljača do objekta i natrag do prijemnika.

Slika 2.7. Uređaj HC-SR04

Tablica 2.4. Specifikacija uređaja HC-SR04

ELEKTRIČNI PARAMETRI	HC-SR04 Ultrazvučni modul
RADNI NAPON	5V DC
RADNA STRUJA	15mA
RADNA FREKVENCIJA	40kHz
MAX. DOMET	4m
MIN. DOMET	2cm
MJERNI KUT	15°
ULAZNI SIGNAL OKIDAČA	10us min. TTL pulse
IZLAZNI EHO SIGNAL	TTL signal razine proporcionalan udaljenosti
DIMENZIJE PLOČE	1-13/16" X 13/16" X 5/8"
SPOJEVI NA PLOČI	4 X 0,1" nagiba pod pravim kutom zaglavlja

2.2.3. TF40 LiDAR

TF40 (Slika 2.8.), prema [6], je LiDAR točnosti na razini mm sa svojim dometom do 40m. LiDAR (engl. Light Detection and Ranging) je tehnologija za daljinsko mjerenje udaljenosti i stvaranje trodimenzionalnih mapa okoline. Ova tehnologija koristi laser za emisiju kratkih svjetlosnih impulsa prema cilju, a zatim registrira vrijeme koje je potrebno da se svjetlosni impuls odbije od cilja i vrati na uređaj. Na temelju vremena putovanja svjetlosnih impulsa, LiDAR sustav može precizno izračunati udaljenost do cilja. LiDAR se često koristi u različitim primjenama, uključujući kartografiju i topografiju, autonomna vozila, urbanističko planiranje, arheologija, industrija, građevina itd. LiDAR tehnologija se kontinuirano razvija, što rezultira sve manjim, jeftinijim i preciznijim uređajima.

Slika 2.8. Uređaj TF40 LiDAR

Značajke:

- Visoka točnost
- Malih dimenzija
- Mala svjetlosna mrlja
- Vidljiv laser, lakši za ciljanje

Tablica 2.5. Specifikacija uređaja TF40 LiDAR

PERFORMANSE PROIZVODA	
DOMET	0,04-40m pri 90% refleksivnosti, 0,04-20m @ 10% refleksivnosti
TOČNOST	±2mm
UDALJENOST RAZLUČIVOSTI	1mm
FREKVENCIJA UZORKOVANJA	5Hz
PONOVLJIVOST	1σ：<2mm
RADNA TEMPERATURA	10~50℃
OPTIČKI PARAMETRI	
IZVOR SVJETLOSTI	LD
VALNA DULJINA	635nm
LASERSKA KLASA	CLASS 2 (EN60825)
KUT DETEKCIJE	<1mrad
ELEKTRIČNI PARAMETRI	
NAPON NAPAJANJA	3,3V
PROSJEČNA STRUJA	≤180mA
POTROŠNJA SNAGE	≤0.6W
MAX. STRUJA	≤180mA
KOMUNIKACIJSKA RAZINA NAPONA	LVTTL (3,3V)
KOMUNIKACIJSKI PROTOKOL	UART
OSTALO	
DIMENZIJE	44mm*64,5mm*23mm（L*W*H）
MATERIJAL KUĆIŠTA	ALUMINIJSKA LEGURA + ABS
SPREMNIK TEMPERATURE	-30~70℃
MASA	57g±3g

2.2.4. R-EVO-MINI-I2C

R-EVO-MINI-I2C (Slika 2.9.), prema [7], je najmanji i najlakši dodatak TeraRanger Evo obitelji senzora. Mjerenje udaljenost, otkrivane pokreta, praćenje razina zaliha i prepoznavanje jednostavnih pokreta samo su neke od stvari koje možete učiniti s Terabee mini senzorom udaljenosti.

Slika 2.9. R-EVO-MINI-I2C

Ključne značajke:

- Time-of-Flight tehnologija
- Odličan omjer cijene i učinka
- Optimizirano za unutarnje mjerenje udaljenosti - 0,03 m do 3,3 m
- Birajte između načina rada s 1, 2 ili 4 piksela
- Lagan dizajn i male veličine - samo 9 grama (uključujući ploču)
- Mala potrošnja energije - prikladno za IoT projekte na baterije
- USB, UART i I2C spojnice, zamjenjive, sučelja
- Kompatibilan s Arduinom, Raspberry Pi i ROS-om

Tablica 2.6. Specifikacija uređaja TR-EVO-MINI-I2C

PERFORMANSE		VRIJEME INICIJALIZACIJE	<1s
PRINCIP DETEKCIJE	Infracrveno vrijeme leta	KOMUNIKACIJA	
VALNA DULJINA IZVORA SVJETLOSTI	940nm	SERIJSKA SUČELJA	USB 2.0 Micro-B UART
PROSTOR NAMJENE	Zatvoreni prostor	VIZUALNA OBAVIJEST	2 x LED diode
PONOVLJIVOST	<5mm	MEHANIČKI PODACI	
REZOLUCIJA IZLAZNE UDALJENOSTI	1mm	DIMENZIJE	42 x 30 x 13 mm (uključujući ploču)
VIDNO POLJE	27°	MASA	9g
PROJEKTIRANO PODRUČJE ZA PRIJEM	48cm x 48cm @ 1m	RADNA TEMPERATURA	-20°C do 75°C
NAČINI RADA PIXEL-a	1px, 2px, 4px (2x2)	MATERIJAL KUĆIŠTA	ABS
ELEKTRONIKA		NAČIN POVEZIVANJA	USB 2.0 Micro-B I2C/UART
NAPON NAPAJANJA DC	5V DC ±5%	SUKLADNOST	
PROSJEČNA STRUJA	50mA	REFERENTNI STANDARD	CE, RoHS

Nakon detaljne provjere ranije navedenih mjernih instrumenata vidljivo je da je najbolji izbor za ovaj eksperiment R-EVO-MINI-I2C. Najveća prednost ovog instrumenta je frekvencija uzorkovanja od 40Hz koja omogućava 40 mjerenja u 1 sekundi. Prednost ovog uređaja su također dimenzije, cijena i uputstva koja korištenje ovog uređaja čine vrlo jednostavnim.

3. IZRADA MAKETE

Prilikom izrade makete najviše pažnje posvećeno je stabilnosti i elastičnosti u smislu da je moguće zamijeniti mjerni instrument kao i duljinu niti i masu kuglice.

Slika 3.1. Maketa prigušenog oscilatora

Slika 3.1. prikazuje maketu prigušenog oscilatora s povezanim mjernim uređajima. Noseći dio makete (stalak, označen kao 1) izrađen je od željeznih četvrtastih cijevi vanjskog promjera 25mm radi stabilnosti ako umjesto metalne kuglice od 50g (označena kao 2) bude ovješeno tijelo veće mase. Na desnoj strani stalka (Slika 3.1.) nalazi se nosač za mjerni uređaj (senzor). Nosač je podesiv po visini i po dubini kako bi bilo moguće regulirati položaj mjernog instrumenta ako je potrebno mjeriti s izmijenjenim parametrima, tj. drugačiji oblik ovješenog predmeta, duljina niti, drugačiji mjerni instrument itd. Ovjesište tijela napravljeno je od željezne šipke duljine 20cm na kojoj je narezan navoj M6 te dvije metalne pločice kojima se steže nit. Na ovjesištu je narezan navoj kako bi bilo moguće regulirati udaljenost ovješenog tijela u odnosu na mjerni instrument.

Bitno je napomenuti da je maketu moguće rastaviti na tri dijela radi transporta i skladištenja te za rastavljanje i sklapanje iste nije potreban alat.

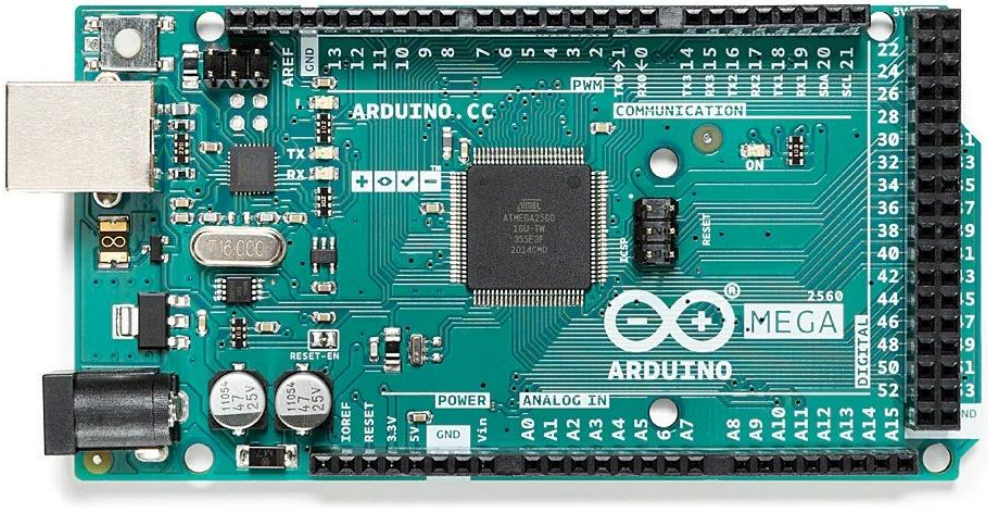
4. EKSPERIMENTI I REZULTATI

Na temelju specifikacija uređaja navedenih u drugom dijelu (2.2. MJERNI INSTRUMENTI), odabran je R-EVO-MINI-I2C (oznaka 4, prikazan na Slici 3.1) kao najkompatibilniji uređaj, koji je povezan s mikroupravljačkom pločom Arduino Mega 2560 (oznaka 3 na Slici 3.1). Prvi korak ovog eksperimentalnog procesa obuhvaća mjerenje udaljenosti kuglice od senzora putem programskog jezika Arduino IDE. Budući da se s ovakvom postavkom makete može mjeriti samo jedna strana elongacije, ključno je izdvajati najmanje udaljenosti kuglice od senzora tijekom svake oscilacije kako bi se postigli relevantni podaci. Drugi korak obuhvaća prelazak na programski jezik Python koji kontinuirano prima izmjerene vrijednosti od prethodnog programa u stvarnom vremenu. Vrijednosti najmanjih udaljenosti kuglice od senzora preračunavaju se u amplitude kuta otklona. Treći korak uključuje ispisivanje rezultata mjerenja i stvaranje grafičkog prikaza u Python-u kako bi se vizualizirali rezultati eksperimenta. Također, vrijeme i amplituda kuta otklona se zapisuju u MS Excel-u kako bi se omogućila bolja analiza i vizualizacija podataka te mogućnost generiranja grafova s pratećim jednadžbama krivulja za daljnju analizu dinamike oscilacija. Navedeni postupak vrši se putem računala (oznaka 5, prikazan na Slici 3.1).

Upute za pokretanje ovog programa nalaze se u **Prilogu 1**.

4.1. Arduino mega 2560 REV3 – izvođenje eksperimenta

Mjerni uređaj R-EVO-MINI-12C povezan je na Arduino Mega 2560 Rev3 (Slika 4.1.)

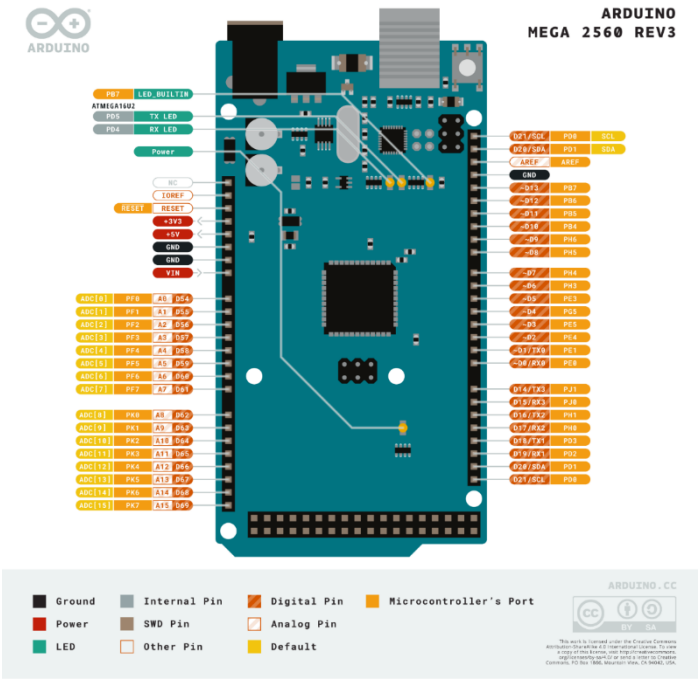


Slika 4.1. Arduino Mega 2560 Rev3

Prema [8], Arduino Mega 2560 predstavlja mikrokontrolersku ploču temeljenu na ATmega2560 mikrokontroleru. Ova ploča (Slika 4.2.) posjeduje 54 digitalna ulazno/izlazna priključka, od kojih je 15 moguće koristiti kao PWM izlazi. Također, nudi 16 analognih ulaznih priključaka, 4 UART (hardverska serijska porta), kristalni oscilator s frekvencijom od 16 MHz, USB priključak, ulaz za napajanje, ICSP zaglavlje te gumb za resetiranje. Ova ploča sadrži sve potrebno za podršku mikrokontroleru. Jako je jednostavno povezati je s računalom putem USB kabela ili ju napajati putem AC-to-DC adaptera ili baterije kako biste započeli s radom. Također, važno je napomenuti da Mega 2560 ploča ostaje kompatibilna s većinom štitova dizajniranih za Uno i prethodne verzije ploča poput Duemilanove ili Diecimila. Mega 2560 predstavlja ažuriranu verziju Arduino Mega, zamjenjujući prethodnu verziju.

Tablica 4.1. Specifikacija uređaja Arduino Mega 2560 Rev3

Radni napon	5V
Ulazni napon (preporučeno)	7-12V
Ulazni napon (limit)	6-20V
Digitalni ulazno/izlazni priključci	54 od kojih 15 daje PWM izlaz
Analogni ulazni priključci	16
DC struja po ulazno/izlaznom priključcima	20mA
DC struja za priključke od 3,3V	50mA
Brza memorija	256KB od čega 8KB koristi bootloader
SRAM	8KB
EEPROM	4KB
Brzina sata	16MHz
LED_BUILTIN	13
Duljina	101,52mm
Širina	53,3mm
Masa	37g



Slika 4.2. Arduino Mega 2560 Rev3 – dijagram priključaka

Upute za korištenje ovog uređaja, kao i detaljne tehničke specifikacije, dostupne su na službenoj internet stranici Arduino platforme. Na toj web lokaciji nalaze se sve potrebne informacije za uspješno upravljanje uređajem te za razumijevanje njegovih tehničkih karakteristika. Preporučuje se konzultiranje ove web stranice kako bi se osiguralo pravilno korištenje uređaja u skladu s najnovijim informacijama i smjernicama dostupnima na Arduino platformi.

4.2. Arduino

U programskom jeziku Arduino IDE razvijen je kod s ciljem povezivanja uređaja R-EVO-MINI-12C s Arduino Mega 2560 Rev3 pločom radi preciznog mjerenja i ispisivanja najmanjih udaljenosti kuglice od senzora tijekom svakog otklona.

U **Prilogu 2 – Arduino program** nalazi se detaljno analiziran i opisan kod, zajedno s komentarima koji prate svaku funkciju. Svaki komentar u kodu označava se s " //" kako je propisano u programskom jeziku Arduino IDE. Ovaj pristup omogućava jasno razumijevanje funkcionalnosti programa i njegove svrhe, čineći ga dostupnim za daljnje prilagodbe i razvoj u budućnosti.

Pokretanjem ovog Arduino programa, sustav će neprekidno mjeriti udaljenost kuglice od senzora u režimu rada jedan piksel s brzinom uzorkovanja od 40 Hz kako bi rezultati mjerenja bili što precizniji. Svaki rezultat mjerenja će biti bilježen u stvarnom vremenu, što znači da će uz svaku zabilježenu udaljenost biti evidentirano točno vrijeme u formatu sati, minuta i sekundi (npr., 19:25:01). Osim toga, dodavanjem kružnog međuspremnika (eng. circular buffer) u kod, eliminirana su nepotrebna mjerenja i osigurano je bilježenje samo amplitude oscilacija, odnosna najmanje udaljenosti kuglice od senzora. Ovaj pristup omogućuje pregledno praćenje i analizu samo relevantnih podataka koji su potrebni za nastavak projekta.

Ključno je naglasiti da ukoliko program generira neočekivane vrijednosti, koji gravitiraju oko granice od približno 630mm, preporučljivo je prilagoditi gornju granicu mjerenja koja se u kodu označava varijablom DISTANCE_MAX. Gornju granicu je potrebno smanjiti na najmanje 530mm. Ova konkretna prilagodba bit će ključna u eliminaciji mogućih grešaka i nesporazuma u funkcioniranju Arduino programa. Smanjenje gornje granice na 530 mm neće bitno narušiti preciznost mjerenja, budući da je ova vrijednost tek nešto manja od kuta od 5°, što znači da neće znatno utjecati na kvalitetu izračuna udaljenosti. Implementacijom ovih

prilagodbi bit će osigurana stabilnost i točnost Arduino koda, čime će se postižu precizna i konzistentna mjerenja udaljenosti kuglice od senzora.

4.3. Python

Sljedeći ključni korak u procesu je pretvaranje udaljenosti kuglice od senzora u kut i stvaranje dijagrama koji prikazuje ovisnost između amplitude kuta otklona o vremenu. Budući da Arduino IDE ne pruža ugrađene alate za matematičke operacije i vizualizaciju podataka, rezultati mjerenja se prenose u programski jezik Python kako bi zadatak bio uspješno riješen.

U **Prilogu 3 – Python program** nalazi se kod s detaljnim komentarima koji opisuju svaku funkciju. Kako bi različiti aspekti koda bili pravilno dokumentirani, svaki komentar u kodu označen je simbolom "#" kako je definirano u programskom jeziku Python. Dodatno, u sklopu **Priloga 3**, nalazi se segment koda koji omogućava prenošenje prikupljenih podataka u MS Excel tablicu. Ova tablica služi kao alat za organizaciju i pohranu rezultata eksperimenta čime postižemo bolju preglednost rezultata mjerenja. Budući da se rezultati mjerenja zapisuju u tablicu, postoji mogućnost crtanja dijagrama te ispis jednadžbe krivulje koja je potrebna za analizu rezultata.

Za bolje razumijevanje linije koda "l_modified = l * 0.75", u nastavku se nalazi objašnjenje. Ova linija koda predstavlja ključnu komponentu u analizi udaljenosti kuglice od središta prigušenog oscilatora, a njezino razumijevanje ključno je za tumačenje rezultata. Varijabla "l" predstavlja udaljenost kuglice od središta prigušenog oscilatora, odnosno toče mirovanja. Udaljenost je definirana matematičkom jednadžbom:

$$l = CENTER_DISTANCE - data \tag{4-1}$$

"CENTER_DISTANCE" je konstanta koja predstavlja udaljenost senzora od središta prigušenog oscilatora. U ovom slučaju, ta udaljenost iznosi 630mm. S druge strane, varijabla "data" predstavlja udaljenost kuglice od senzora, koja se dinamički mijenja tijekom oscilacija. Razlog za primjenu ove linije koda je rješavanje anomalija. Na prigušenom oscilatoru nalazi se kutomjer za mjerenje kuta između kuglice i središta oscilatora. Usporedbom rezultata mjerenja uređaja i vizualnom ocjenom kutova golim okom zamijećena su odstupanja. Provedene su detaljne provjere kutova na uzorcima pod različitim kutovima: 5°, 10°, 15°, 20° i 25°. Analizom rezultata detektirano je linearno odstupanje koje je približno jednako koeficijentu 0,75. Linija koda "l_modified = l * 0.75" koristi se za prilagodbu udaljenosti

kuglice od središta prema otkrivenom odstupanju. Ovo je ključno za korekciju podataka i dobivanje točnih rezultata. Ovaj postupak analize i korekcije odstupanja temeljito je provjeren i dokumentiran radi osiguranja valjanost rezultata mjerenja.

4.4. Rezultati mjerenja

Usklađeno djelovanje između Arduino IDE i Python koda bitno je za precizno prikazivanje i analizu mjerenja. Za postizanje ovog cilja potrebno je pravilno konfigurirati Arduino IDE kako bi se osiguralo ispravno izvršavanje Python koda i prikazivanje rezultata mjerenja.

Prvi važan korak je pokretanje Arduino IDE-a bez "serial monitora". Ova konfiguracija omogućuje Python kodu da čita podatke s Arduina i koristi ih za izradu grafičkih prikaza i analize. Bez otvorenog "serial monitora", komunikacija između Arduino i Python koda je neometana, što je od suštinskog značaja za našu analizu.

Python kod ispisuje mjerenja u posebnom formatu radi lakšeg čitanja i analize. Format {"Sensor measure" "l" "l_modified" "angle" "vrijeme"} jasno definira redoslijed i prikaz podataka (Slika 4.3.). Vrijednosti duljina "Sensor measure" i "l" zaokružene su na jednu decimalu, dok je "l_modified" zaokružen na dvije decimale kako bi se osigurala preciznost.

Sensor measure: 68.0 l = 562.0 l_modified = 421.5 angle = 24.929325033155735 vrijeme = 12.983996

Sensor measure: 171.0 l = 459.0 l_modified = 344.25 angle = 20.1360197093923 vrijeme = 55.775942

Sensor measure: 283.0 l = 347.0 l_modified = 260.25 angle = 15.084896772614655 vrijeme = 110.476076

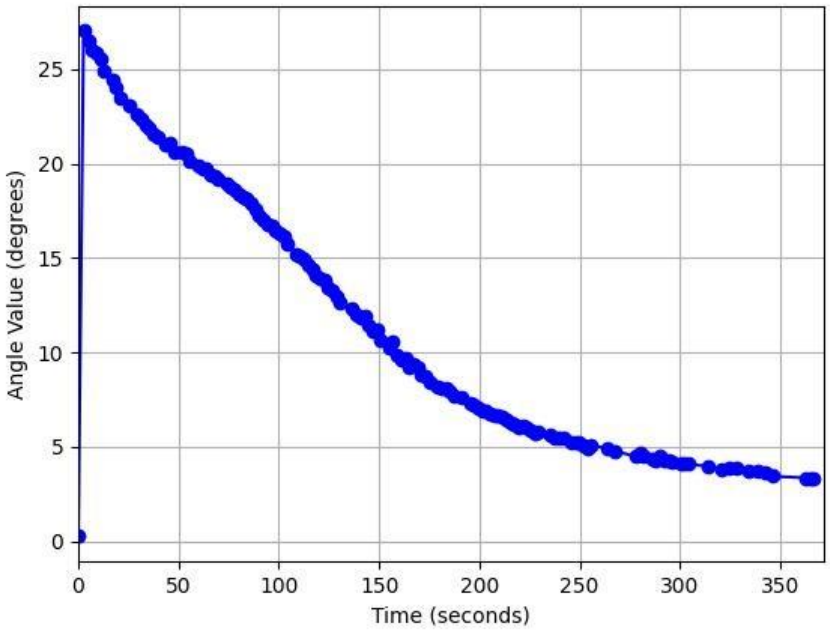
Sensor measure: 386.0 l = 244.0 l_modified = 183.0 angle = 10.544550322861383 vrijeme = 156.979844

Sensor measure: 512.0 l = 118.0 l_modified = 88.5 angle = 5.077319059793709 vrijeme = 251.821799

Slika 4.3. Ispis rezultata mjerenja kutova približnih 25°, 20°, 15°, 10° i 5° - Python

Kutovi su prikazani s čak četrnaest decimala, a vrijeme sa šest decimala. Ovaj detaljan format pruža pregledan i precizan uvid u dinamiku promjene vrijednosti tijekom eksperimenta.

Koristeći prikupljene podatke, Python izrađuje trenutačne grafičke prikaze rezultata mjerenja, što je jasno vidljivo na Slici 4.4.



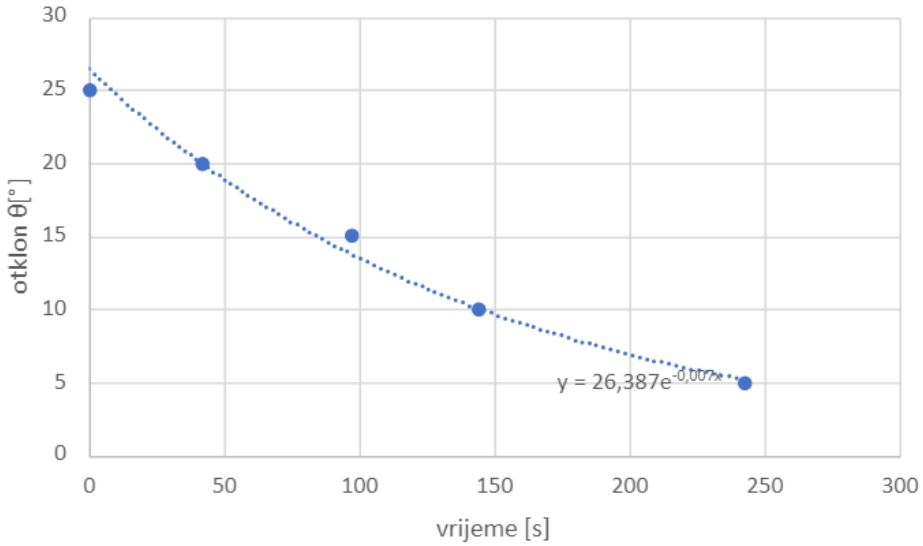
Slika 4.4. Dijagram kuta otklona u ovisnosti u vremenu – Python

Slika 4.4. prikazuje promjene amplitude kuta otklona kuglice u odnosu na vrijeme. Ova slika pruža vizualni prikaz rezultata i omogućuje brzu interpretaciju dinamike oscilacija. Rezultati prikazani na slici jasno ilustriraju kako se kut otklona mijenja tijekom vremena.

Tablica 4.2. Rezultati mjerenja kutova približnih 25°, 20°, 15°, 10° i 5° - MS Excel

vrijeme t[s]	otklon θ[°]
12,983996	24,929325033155735
55.775942	20,1360197093923
110.476076	15,084896772614655
156.979844	10, 544550322861383
251.821799	5, 077319059793709

Rezultati svih mjerenja ispisanih u MS Excel tablici nalaze se u **Prilogu 4**.



Slika 4.5. Dijagram kuta otklona u ovisnosti u vremenu – rezultati mjerenja

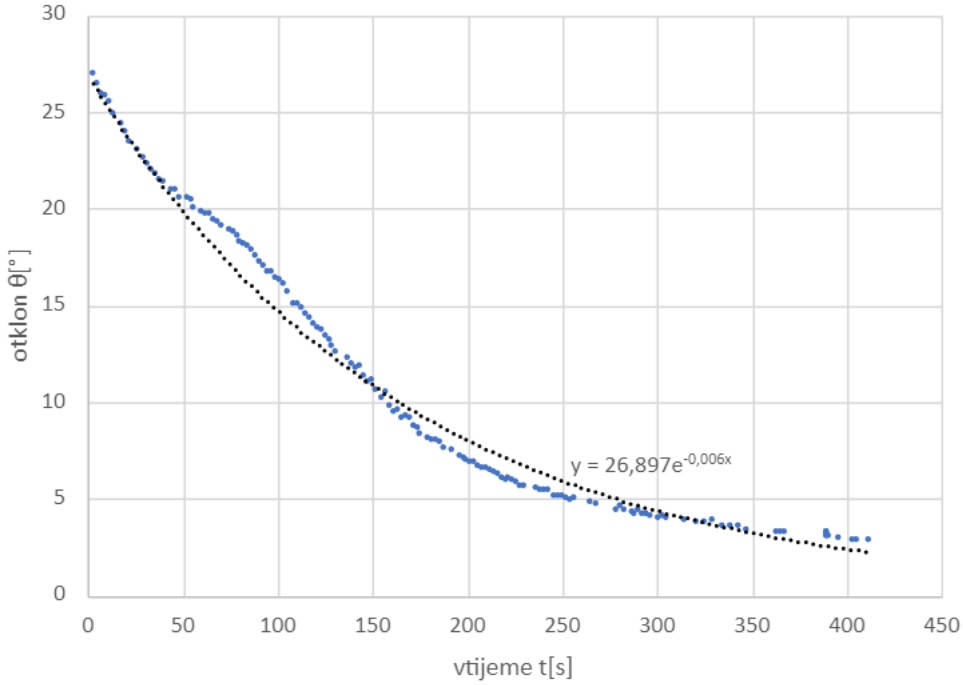
Slika 4.5. sadrži sve relevantne podatke iz Tablice 4.2. Proučavajući dijagram na Slici 4.5., vidljivi su trendovi i obrasci koji sugeriraju postojanje određene korelacije ili zavisnosti između različitih varijabli. Povlačenjem linije koja najbolje odgovara raspodjeli podataka moguće je izvesti pretpostavke o linearnoj zavisnosti između promatranih varijabli. Ovaj korak je ključan u procesu kvantifikacije i modeliranja odnosa među varijablama.

Eksponencijalna jednadžba krivulje glasi:

$$\theta(t) = 26{,}387e^{-0{,}007t} \tag{4-2}$$

Prema relacijama (1-6) i (4-2) vidljivo je da amplituda iznosi θ=26,387°, dok faktor prigušenja iznosi δ=0,007s^{-1}.

Prema relaciji (1-9), faktor dobrote koji iznosi Q=224,4.



Slika 4.6. Grafički prikaz ovisnosti kuta otklona o vremenu

Rezultati mjerenja prvih 400 sekundi eksperimenta ispisani su u MS Excel tablici i nalaze se u **Prilogu 4**. Slika 4.6. grafički prikazuje promjenu amplitude kuta otklona u vremenu. Eksponencijalna jednadžba krivulje glasi:

$$\theta(t) = 26{,}897e^{-0{,}006t} \qquad (4\text{-}3)$$

Prema relaciji (4-3), amplituda iznosi θ=26,897°, dok faktor prigušenja iznosi $\delta=0{,}006s^{-1}$. Prema relaciji (1-9), faktor dobrote koji iznosi Q=261,8.

5. ZAKLJUČAK

Na temelju dobivenih rezultata, može se zaključiti da maketa prigušenog oscilatora ne zadovoljava u potpunost faktor dobrote. Iako ovaj eksperiment nije u potpunosti uspješan, i dalje pruža dublje razumijevanje prigušenih oscilatora. Nadalje, rezultati ovog eksperimenta mogu biti od velike koristi za studente koji sudjeluju u laboratorijskim vježbama iz fizike. Ova maketa i prateći eksperimentalni pristup mogu pružiti praktično iskustvo u analizi i razumijevanju fenomena prigušenog oscilatora te ostavlja prostor za poboljšanje i daljnji razvoj.

LITERATURA

[1] Ž. Mioković, Fizika 1 priručnik za laboratorijske vježbe, str. 309-405, Đakovo, 1996.

[2] Data Harvest: Wireless Data Logging Solutions, dostupno na https://www.youtube.com/watch?v=97CfLu3627w&ab_channel=DataHarvest%3AWirelessDataLoggingSolutions [pristup: 14.9.2023.]

[3] PHET, Pendulum Lab, dostupno na: https://phet.colorado.edu/en/simulations/pendulum-lab [pristup: 14.9.2023.]

[4] Mouser Electronics, dostupno na: https://hr.mouser.com/pdfDocs/XM132Datasheet.pdf

[5] ALLDATASHEET.COM, List of Unclassifed Manufacturers, dostupno na: https://www.alldatasheet.com/datasheet-pdf/pdf/1132205/ETC2/SR04.html [pristup 14.9.2023.]

[6] Mouser Electronics, TF40 Product Specification V1.0 , dostupno na: https://hr.mouser.com/datasheet/2/1099/Benewake_10152020_TF40-1954048.pdf [pristup 14.9.2023.]

[7] Mouser Electronics, Specification sheet Terabee, dostupno na: https://hr.mouser.com/datasheet/2/944/evoMini-specsheet-1670051.pdf [pristup 14.9.2023.]

[8] Arduino, dostupno na: https://store.arduino.cc/products/arduino-mega-2560-rev3 [pristup 14.9.2023.]

SAŽETAK

U okviru ovog završnog rada izrađena je maketa prigušenog oscilatora koristeći Arduino mikroupravljač. Maketa se sastoji od jednostavnog njihala s kuglicom koja visi na tankoj niti. Glavni cilj istraživanja je mjerenje amplitude kuta otklona kuglice tijekom vremena kako bi se proučavala postupna promjena amplitude oscilacija. Kroz korištenje senzora i Arduino platforme, prikupljeni su traženi podaci. Ovi podaci su prikazani i analizirani u obliku tabličnih podataka i grafičkog prikaza. Ovaj rad pruža uvid u primjenu Arduino tehnologije za istraživanje prigušenih oscilacija i njihovu vizualizaciju, čime doprinosi boljem razumijevanju ovog fenomena.

Ključne riječi: Arduino, matematičko njihalo, prigušeni oscilator, Python

ABSTRACT

Making a model of a dumped oscillator

As part of this final work, a model of a damped oscillator was made using an Arduino microcontroller. The model consists of a simple pendulum with a ball hanging on a thin thread. The main goal of the research is to measure the amplitude of the deflection angle of the ball over time in order to study the gradual change in the amplitude of the oscillations. Through the use of sensors and the Arduino platform, the required data was collected. These data are presented and analyzed in the form of tabular data and graphical representation. This paper provides an insight into the application of Arduino technology for the research of damped oscillations and their visualization, thereby contributing to a better understanding of this phenomenon.

Keywords: Arduino, damped oscillator, mathematical pendulum, Python

ŽIVOTOPIS

Pavle Gegić rođen je 19. veljače 1999. godine u Vukovaru. 2005. godine počinje pohađati osnovnu školu u Područnoj školi Vera, 2009. godine nastavlja školovanje u Osnovnoj školi Bobota te 2013. godine nastavlja srednjoškolsko obrazovanje u Gimnaziji Vukovar, smjer Prirodoslovno-matematička gimnazija. Nakon četverogodišnjeg srednjoškolskog obrazovanja i položene državne mature, 2017. godine upisuje Preddiplomski studij Elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer Elektrotehnika te se na drugoj godini opredjeljuje za Elektroenergetiku kao izborni blok.

Prilog 1 – upute za pokretanje programa

Da biste pokrenuli Python skriptu na sustavu Windows, slijedite ove korake:

1. Otvorite naredbeni redak:
    a. Pritisnite tipku Windows + R za otvaranje Run dijaloga.
    b. Upišite "cmd" i pritisnite Enter. Otvorit će se naredbeni redak.
2. Promijenite trenutni direktorij na direktorij u kojem se nalazi vaša skripta.
    a. Upišite naredbu cd C:\Putanja\Do\Vašeg\Direktorija i pritisnite Enter
    b. npr. cd C:\Users\PAVLE\OneDrive\Dokumenti\Zavrsni\Python
3. 3Pokrenite Python skriptu:
    a. Nakon što ste promijenili direktorij, upišite sljedeću naredbu za pokretanje Python skripte "njihalo.py":
    b. python njihalo.py
4. 4. Pritisnite Enter. Skripta će se izvršiti, a rezultati će biti prikazani.

Važno je napomenuti da za pokretanje Python skripte trebate imati Python instaliran na vašem računalu. Preporučljivo je koristiti Python 3.x verziju jer su starije verzije Pythona dostigle kraj životnog ciklusa i ne primaju više ažuriranja.

Ako Python nije instaliran na vašem računalu, možete preuzeti i instalirati najnoviju verziju s Python web stranice (https://www.python.org/).

Prilog 2 – Arduino program

```
// definiranje minimalnog i maksimalnog otklona

#define DISTANCE_MIN 20u

#define DISTANCE_MAX 630u

// kreiranje tablice cikličke provjere redundantnosti koja se koristi u funkciji “crc8”

static const uint8_t crc_table[] = {

  0x00, 0x07, 0x0e, 0x09, 0x1c, 0x1b, 0x12, 0x15, 0x38, 0x3f, 0x36, 0x31,

  0x24, 0x23, 0x2a, 0x2d, 0x70, 0x77, 0x7e, 0x79, 0x6c, 0x6b, 0x62, 0x65,

  0x48, 0x4f, 0x46, 0x41, 0x54, 0x53, 0x5a, 0x5d, 0xe0, 0xe7, 0xee, 0xe9,

  0xfc, 0xfb, 0xf2, 0xf5, 0xd8, 0xdf, 0xd6, 0xd1, 0xc4, 0xc3, 0xca, 0xcd,

  0x90, 0x97, 0x9e, 0x99, 0x8c, 0x8b, 0x82, 0x85, 0xa8, 0xaf, 0xa6, 0xa1,

  0xb4, 0xb3, 0xba, 0xbd, 0xc7, 0xc0, 0xc9, 0xce, 0xdb, 0xdc, 0xd5, 0xd2,

  0xff, 0xf8, 0xf1, 0xf6, 0xe3, 0xe4, 0xed, 0xea, 0xb7, 0xb0, 0xb9, 0xbe,

  0xab, 0xac, 0xa5, 0xa2, 0x8f, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9d, 0x9a,

  0x27, 0x20, 0x29, 0x2e, 0x3b, 0x3c, 0x35, 0x32, 0x1f, 0x18, 0x11, 0x16,

  0x03, 0x04, 0x0d, 0x0a, 0x57, 0x50, 0x59, 0x5e, 0x4b, 0x4c, 0x45, 0x42,

  0x6f, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7d, 0x7a, 0x89, 0x8e, 0x87, 0x80,

  0x95, 0x92, 0x9b, 0x9c, 0xb1, 0xb6, 0xbf, 0xb8, 0xad, 0xaa, 0xa3, 0xa4,

  0xf9, 0xfe, 0xf7, 0xf0, 0xe5, 0xe2, 0xeb, 0xec, 0xc1, 0xc6, 0xcf, 0xc8,

  0xdd, 0xda, 0xd3, 0xd4, 0x69, 0x6e, 0x67, 0x60, 0x75, 0x72, 0x7b, 0x7c,

  0x51, 0x56, 0x5f, 0x58, 0x4d, 0x4a, 0x43, 0x44, 0x19, 0x1e, 0x17, 0x10,

  0x05, 0x02, 0x0b, 0x0c, 0x21, 0x26, 0x2f, 0x28, 0x3d, 0x3a, 0x33, 0x34,

  0x4e, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5c, 0x5b, 0x76, 0x71, 0x78, 0x7f,

  0x6a, 0x6d, 0x64, 0x63, 0x3e, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2c, 0x2b,

  0x06, 0x01, 0x08, 0x0f, 0x1a, 0x1d, 0x14, 0x13, 0xae, 0xa9, 0xa0, 0xa7,
```

```
  0xb2, 0xb5, 0xbc, 0xbb, 0x96, 0x91, 0x98, 0x9f, 0x8a, 0x8d, 0x84, 0x83,
  0xde, 0xd9, 0xd0, 0xd7, 0xc2, 0xc5, 0xcc, 0xcb, 0xe6, 0xe1, 0xe8, 0xef,
  0xfa, 0xfd, 0xf4, 0xf3
};

// funkcija za izračunavanje CRC-8 checksuma
uint8_t crc8(uint8_t *p, uint8_t len) {
  uint8_t i;
  uint8_t crc = 0x0;
  while (len--) {
    i = (crc ^ *p++) & 0xFF;  // XOR trenutnog bajta s CRC-om
    crc = (crc_table[i] ^ (crc << 8)) & 0xFF;
  }
  return crc & 0xFF;
}

// popis komandi
const byte PRINTOUT_BINARY[4] = { 0x00, 0x11, 0x02, 0x4C };
const byte PRINTOUT_TEXT[4] = { 0x00, 0x11, 0x01, 0x45 };
const byte RUNMODE_SINGLE_PIXEL[4] = { 0x00, 0x21, 0x01, 0xBC };
const byte RUNMODE_TWO_PIXEL[4] = { 0x00, 0x21, 0x03, 0xB2 };
const byte RUNMODE_TWO_BY_TWO_PIXEL[4] = { 0x00, 0x21, 0x02, 0xB5 };
const byte RANGEMODE_SHORT[4] = { 0x00, 0x61, 0x01, 0xE7 };
const byte RANGEMODE_LONG[4] = { 0x00, 0x61, 0x03, 0xE9 };
// inicijalizacija varijabli
```

```
const int BUFFER_LENGTH = 10;

uint8_t Framereceived[BUFFER_LENGTH];

uint8_t index;

uint16_t distance;

uint16_t distance1;

uint16_t distance2;

uint16_t distance3;

//definiranje kružnog buffera

#define bufferSize (5u)

uint16_t circularBuffer[bufferSize];

uint8_t bufferHead = 0u;

uint16_t oscilationCount = 0u;

bool isBallReverseDetected = false;

uint8_t indexOfMinValue = 0u;

bool check_is_ball_reversing_direction(void);

void setup() {

  pinMode(13, OUTPUT);                    //inicijalizirati digitalni pin 13 kao izlaz (izbrisiv ako komunikacija radi)

  Serial.begin(115200);                   //otvori serijski priključak 0 (koji odgovara USB priključku i pinovima TX0 i RX0), postavi brzinu prijenosa podataka na 115200 bps

  Serial1.begin(115200);                  //otvori serijski priključak 1 (koji odgovara pinovima TX1 i RX1), postavi brzinu prijenosa podataka na 115200 bps

  Serial1.write(RUNMODE_SINGLE_PIXEL, 4);  //postavi TeraRanger u način rada s jednim pikselom

  Serial1.write(RANGEMODE_SHORT, 4);        //postavi Teraranger u način rada kratkog dometa
```

```
  Serial1.write(PRINTOUT_BINARY, 4);       //postavi TeraRanger u binarni način, tekstualni način nije podržan
  index = 0;
}

//glavna petlja počinje ovdje
void loop() {
  if (Serial1.available() > 0) {
    // šalje podatke samo kada zaprimi podatke
    uint8_t inChar = Serial1.read();
    if (index == 0) {
      if (inChar == 'T') {
        //traži početak okvira “T”
        Framereceived[index++] = inChar;
      } else return;
    } else if ((index > 0) && (index < 10)) {
      //prikupljanje podataka
      Framereceived[index++] = inChar;
    }
    //provjera je li okvir od  jednog piksela
    if (index == 4) {
      if (crc8(Framereceived, 3) == Framereceived[3]) {
        //pretvaranje bajta u udaljenost
        distance = (Framereceived[1] << 8) + Framereceived[2];
        //filtriranje netočno detektiranih podataka
```

```
    if (distance >= DISTANCE_MIN && distance <= DISTANCE_MAX) {
      circularBuffer[bufferHead] = distance;
      //provjera je li najmanja vrijednost oscilacije detektirana
      isBallReverseDetected = check_is_ball_reversing_direction();
      (++bufferHead) %= bufferSize;
      if (true == isBallReverseDetected) {
        //kuglica se odmiče od senzora
        Serial.println(circularBuffer[indexOfMinValue]);
        //reset radi sigurnosti
        isBallReverseDetected = false;
      }
    }
    index = 0;
    Framereceived[0] = 0;
  }
}
//provjera je li okvir od  dva piksela
else if (index == 6) {
  if (crc8(Framereceived, 5) == Framereceived[5]) {
    //pretvaranje bajta u udaljenost
    distance = (Framereceived[1] << 8) + Framereceived[2];
    distance1 = (Framereceived[3] << 8) + Framereceived[4];
    Serial.print("Distances in mm : ");
    Serial.print(distance);
```

```
    Serial.print(" | ");
    Serial.println(distance1);
    index = 0;
    Framereceived[0] = 0;
  }
}
//provjera je li okvir od  dva puta dva piksela
else if (index == 10) {
  if (crc8(Framereceived, 9) == Framereceived[9]) {
    //pretvaranje bajta u udaljenost
    distance = (Framereceived[1] << 8) + Framereceived[2];
    distance1 = (Framereceived[3] << 8) + Framereceived[4];
    distance2 = (Framereceived[5] << 8) + Framereceived[6];
    distance3 = (Framereceived[7] << 8) + Framereceived[8];
    Serial.print("Distances in mm : ");
    Serial.print(distance);
    Serial.print(" | ");
    Serial.print(distance1);
    Serial.print(" | ");
    Serial.print(distance2);
    Serial.print(" | ");
    Serial.println(distance3);
    index = 0;
    Framereceived[0] = 0;
```

```
    } else {
      index = 0;
      Framereceived[0] = 0;
      Serial.println("CRC checks failed. Couldn't find valid frame in buffer length");
    }
   }
  }
}

bool check_is_ball_reversing_direction() {
  bool retVal = false;
  uint8_t indexTestMinus2 = (bufferHead + 1u) % bufferSize;
  uint8_t indexTestMinus1 = (indexTestMinus2 + 1u) % bufferSize;
  uint8_t indexTest = (indexTestMinus1 + 1u) % bufferSize;
  uint8_t indexTestPlus1 = (indexTest + 1u) % bufferSize;
  uint8_t indexTestPlus2 = (indexTestPlus1 + 1u) % bufferSize;
  //provjera je li vrijednost koja se trenutno testira najniža od "susjednih"
  if ((circularBuffer[indexTestMinus1] > circularBuffer[indexTest]) &&
(circularBuffer[indexTest] < circularBuffer[indexTestPlus1])) {
   if ((circularBuffer[indexTestMinus2] > circularBuffer[indexTestMinus1]) &&
(circularBuffer[indexTestPlus1] < circularBuffer[indexTestPlus2])) {
    //vraća true kako bi označi da je vrijednost pronađena
    retVal = true;
    //pohrani indeks koji će se koristiti za ispis vrijednosti
    indexOfMinValue = indexTest;
```

```
    }
  }
  return retVal;
}
```

Prilog 3 – Python program

```
import serial #biblioteka koja omogućava komunikaciju s Arduinom putem serijske veze

import matplotlib.pyplot as plt #biblioteka koja omogućava stvaranje grafova i dijagrama

from datetime import datetime #biblioteka koja omogućava praćenje vremenskih informacija

import numpy as np # biblioteka koja omogućava znanstveno računanje (nizovi i matrice)

import math #biblioteka koja omogućava izvođenje matematičkih operacija

import pandas as pd #biblioteka za izvoz u Excel


COM_PORT = 'COM3'  #naziv USB priključka na računalu

BAUD_RATE = 115200  #brzina prijenosa podataka u baud-ima

#pokretanje serijske komunikacije

ser = serial.Serial(COM_PORT, BAUD_RATE)

CENTER_DISTANCE = 630 #udaljenost senzora od kuglice u stanju mirovanja (0°)

#pokretanje listi za spremanje podataka

timestamps = []

data_values = []

angle_values = []

start_time = None  #vremenska varijabla se pokreće zaprimanjem prve izmjerene vrijednosti

#inicijalizacija za crtanje

plt.ion()  #pokretanje interaktivnog načina rada

fig, ax = plt.subplots()

ax.set_xlabel('vrijeme t[s]')

ax.set_ylabel('otklon θ[°] ')

ax.grid(True)

#određivanje stila linije s oznakama za sve točke
```

```
line, = ax.plot([], [], 'bo-')  # 'bo-' znači plavi krugovi povezani linijama
#funkcija za ažuriranje crtanja
def update_plot():
    elapsed_time = (datetime.now() - start_time).total_seconds()
    line.set_xdata(timestamps)
    line.set_ydata(angle_values)
    ax.relim()
    ax.autoscale_view()
    ax.set_xlim([0, elapsed_time + 5])  #podešavanje granica x-osi
    fig.canvas.flush_events()

try:
    while True:
        #čitanje podataka sa serijskog porta red po red
        data_line = ser.readline().decode('utf-8').strip()
        #trenutno vrijeme proteklo u sekundama
        elapsed_time = (datetime.now() - start_time).total_seconds() if start_time else 0
        data = float(data_line) #podatak udaljenosti kuglice od senzora
        l = CENTER_DISTANCE - data #udaljenost kuglice od centra
        l_modified = l * 0.75 #pogreška senzora
        angle_in_radians = math.acos((l_modified/1000)) #računanje kuta u radijanima
        angle_in_degrees = math.degrees(angle_in_radians) #pretvaranje radijana u stupnjeve
        angle_alpha = 90 - angle_in_degrees #izračun treženog kuta
        #postavljanje vremena početka ako već nije postavljeno
```

```
        if start_time is None:
            start_time = datetime.now()
        #filtriranje rezultata s prevelikim odstupanjem
        skip_sample = False
        if len(data_values) > 1:
            if data_values[-1] > data:
                if data_values[-1] - data > 50:
                    skip_sample = True
            else:
                if data - data_values[-1] > 50:
                    skip_sample = True
        if skip_sample == False:
            #dodavanje podataka u listu
            timestamps.append(elapsed_time)
            data_values.append(data)
            angle_values.append(angle_alpha)
            print("Sensor measure: {}  l = {}  l_modified = {}  angle = {}  vrijeme = {}".format(data, l, l_modified, angle_alpha, elapsed_time))
            #crtanje dijagrama
            update_plot()

#zaustavljanje crtanja na naredbu s tipkovnice (CTRL+C)
except KeyboardInterrupt:
    print("Plotting stopped.")
    ser.close()
```

```
except KeyboardInterrupt:

print("Plotting stopped.")

ser.close()

# Stvaranje DataFrame-a iz dobivenih podataka

data_df = pd.DataFrame({'vrijeme [s]': timestamps, 'otklon [°]': angle_values})

# Ime Excel datoteke za izvoz

excel_file_name = 'podaci.xlsx'

# Izvoz podataka u Excel

data_df.to_excel(excel_file_name, index=False)

# Ispis poruke o uspješnom izvozu

print(f"Podaci su izvezeni u '{excel_file_name}'")
```

Prilog 4 – MS Excel tablica rezultata

vrijeme [s]	otklon θ[°]
45.079	27.032
45.173	26.503
45.175	26.024
45.177	25.833
45.179	25.546
45.181	24.929
17.0	24.409
44.945	24.032
44.947	23.468
44.982	23.094
45.014	22.628
45.016	22.349
33.4	22.024
35.4	21.885
37.4	21.561
39.5	21.422
43.5	21.008
45.6	21.054
47.6	20.640
51.7	20.594
53.7	20.502
55.7	20.136
59.8	19.861
61.8	19.770
63.8	19.770
65.8	19.450
67.9	19.359
69.9	19.177
73.9	18.950
76.0	18.814
78.0	18.632

80.0	18.360
82.1	18.179
84.1	18.134
86.1	17.908
88.2	17.592
90.2	17.277
92.2	17.052
94.2	16.783
96.3	16.738
98.3	16.469
100.3	16.334
102.3	16.155
104.4	15.753
108.3	15.173
110.4	15.084
112.4	14.951
114.5	14.640
116.5	14.418
118.5	14.063
120.5	13.886
122.6	13.798
124.5	13.444
126.6	13.267
128.6	12.958
130.7	12.606
136.7	12.298
138.8	11.990
140.8	11.814
142.8	11.946
144.8	11.420
146.9	11.113
148.8	11.200
150.9	10.675

155.0	10.238
156.9	10.544
158.9	9.889
160.9	9.584
163.0	9.671
165.0	9.235
167.0	9.366
169.0	9.192
171.0	8.800
173.1	8.757
175.1	8.409
179.1	8.192
181.2	8.062
183.2	8.062
185.2	7.975
187.2	7.671
191.2	7.585
195.3	7.281
197.3	7.195
199.3	7.065
201.3	6.935
203.3	6.935
205.5	6.718
207.4	6.675
209.4	6.675
211.4	6.545
213.5	6.416
215.5	6.286
217.5	6.156
219.5	6.027
221.5	6.113
223.5	6.027
225.6	5.897

227.6	5.724
229.6	5.767
235.6	5.595
237.7	5.508
239.7	5.508
241.7	5.465
245.7	5.206
247.7	5.249
249.8	5.206
251.8	5.077
253.8	4.947
255.8	5.077
263.9	4.904
267.9	4.775
278.0	4.473
280.0	4.646
282.0	4.516
286.1	4.387
288.1	4.301
290.0	4.473
292.1	4.301
294.1	4.258
296.2	4.215
300.2	4.085
302.2	4.128
304.3	4.085
314.2	3.956
320.3	3.827
324.3	3.870
328.3	3.913
334.4	3.698
338.4	3.698
342.5	3.612

346.5	3.439
362.6	3.353
364.6	3.310
366.7	3.310
389.1	3.353
389.2	3.310
389.3	3.267
389.4	3.181
389.5	3.095
389.6	3.095
394.9	3.009
402.8	2.923
404.9	2.923
410.9	2.966