Obrada signala mjerenja kružnosti

Kožuh, Vilim

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Vilim Kožuh

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:
Prof. dr. sc. Vedran Šimunović

Student:
Vilim Kožuh

Zagreb, 2024.

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

Zahvaljujem mentoru Prof. dr. sc. Vedranu Šimunoviću na korisnim savjetima, dostupnosti za pomoć i prenesenom znanju tokom pisanja ovog rada. Također bi se htio zahvaliti prijatelju Petru sa FER-a, koji je uvelike pomogao sa razumijevanjem koncepata vezanih uz obradu signala i programiranje.

Vilim Kožuh

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment, mehatronika i robotika, autonomni sustavi i računalna inteligencija

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

DIPLOMSKI ZADATAK

Student: **Vilim Kožuh** JMBAG: 0035225520

Naslov rada na hrvatskom jeziku: **Obrada signala mjerenja kružnosti**

Naslov rada na engleskom jeziku: **Signal processing for roundness measurement**

Opis zadatka:

Mjerenje i kvaliteta nezaobilazan su dio proizvodnog procesa. Sve veći zahtjevi na automatizaciju procesa postavljaju u fokus obradu digitalnog signala mjerenja primjenom numeričkih metoda. U radu će biti potrebno obraditi signal mjerenja kružnosti (diskontinuirani) kako bi se mogli odrediti parametri kružnosti *RONt*, i *RONq* metodom najmanjih kvadrata nakon primjene Gauss digitalnog filtra. U zadatku je potrebno:

- Opisati mjerenje kružnosti metodom s vanjskom mjernom referencom.
- Opisati signal mjerenja kružnosti.
- U programskom paketu Pyhton napisati algoritam za digitalnu filtraciju signala (Gauss filtar).
- U programskom paketu Pyhton napisati algoritam za izračun *RONt*, i *RONq* parametara kružnosti.
- Provesti mjerenje kružnosti i analizirati dobivene rezultate.

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

Zadatak zadan:	Datum predaje rada:	Predviđeni datumi obrane:
9. svibnja 2024.	11. srpnja 2024.	15. – 19. srpnja 2024.

Zadatak zadao: Predsjednik Povjerenstva:

Doc.dr.sc. Vedran Šimunović Prof. dr. sc. Ivica Garašić

SADRŽAJ

POPIS SLIKA III
POPIS TABLICA V
POPIS OZNAKA VI
POPIS KRATICA VII
SAŽETAK VIII
SUMMARY IX
1. UVOD 1
2. KRUŽNOST 2
2.1. Pojmovi, definicije, greške 2
2.2. Mjerenje kružnosti 6
2.2.1. Komponente stroja za mjerenje kružnosti sa rotirajućim stolom i princip rada 12
2.2.2. Moguće greške prilikom mjerenja 14
2.2.3. Procedura mjerenja kružnosti 16
3. NAČELA OBRADE PODATAKA 19
3.1. Prikupljeni podatci i njihov prikaz 19
3.2. Uklanjanje ekscentričnosti 22
3.2.1. Uklanjanje ekscentričnosti limakon aproksimacijom 22
3.2.2. Uklanjanje ekscentričnosti Fourierovom transformacijom 24
3.3. Fourierova transformacija 25
3.3.1. Diskretna Fourierova transformacija 29
3.3.2. Brza Fourierova transformacija 31
3.4. Gaussov filter 32
4. PROGRAMIRANJE BIBLIOTEKE FUNKCIJA 34
4.1. Program za analizu odstupanja od kružnosti 35
4.1.1. Importovi glavnog programa 36
4.1.2. Odabir datoteke 36
4.1.3. Uklanjanje ekscentra aproksimacijom limakona 39
4.1.4. Crtanje profila 41
4.1.5. Računanje RONp, RONv, RONq i RONt 42
4.1.6. Fourierova transformacija 43
4.1.7. Gaussova filtracija 45

4.2. Generiranje signala 48
4.2.1. Importovi programa za generiranje signala 49
4.2.2. Generiranje signala 49
4.2.3. Prikaz signala i profila podataka....51
4.2.4. Generiranje datoteka 51
5. ANALIZA REZULTATA MJERENJA....53
5.1. Provjera učitavanja podataka....53
5.2. Provjera uklanjanja ekscentra limakon aproksimacijom i Fourierovom transformacijom....54
5.3. Provjera Gaussove filtracije podataka 55
5.4. Provjera proračuna parametara kružnosti....56
5.5. Provjera ukupnog rada programa sa nasumično zadanim parametrima 57
5.6. Obrada realnih mjernih rezultata 60
6. ZAKLJUČAK....63
LITERATURA....64
PRILOZI 65

POPIS SLIKA

Slika 1. Primjeri kružnica koje definiraju kružnost predmeta 2
Slika 2. Referentne kružnice i parametri, a) MZCI referentna kružnica, b) LSCI referentna kružnica, *RONp* i *RONv*, c) MICII referentna kružnica, d) MCCI referentna kružnica [4] 4
Slika 3. Cilindričnost [4] 5
Slika 4. Greške kružnosti nastale tokom strojne obrade, a) ovalnost, b) izbočenost [6] 6
Slika 5. Mjerenje promjera na različitim mjestima [4] 7
Slika 6. Oblici profila koji pri mjerenju pomičnim mjerilom daju istu vrijednost [4] 8
Slika 7. Ispitivanje kružnosti primjenom V-prizmi [7] 8
Slika 8. Greške prilikom mjerenja pomoću V-prizmi [4] 9
Slika 9. Ispitivanje kružnosti primjenom mjernih šiljaka [7] 10
Slika 10. Greške prilikom ispitivanja kružnosti primjenom mjernih šiljaka [2] 10
Slika 11. Mjerenje sa vanjskom mjernom referencom, a) tip rotirajućeg stola, b) tip rotirajućeg ticala [2] 11
Slika 12. Shema uređaja za ispitivanje kružnosti primjenom okretnog stola [2] 12
Slika 13. Uređaj za ispitivanje kružnosti primjenom okretnog stola 12
Slika 14. Građa ticala [4] 13
Slika 15. Koncentričnost i ekscentričnost 15
Slika 16. Moguće greške geometrije uređaja [4] 15
Slika 17. Stavljanje predmeta na uređaj 16
Slika 18. Pozicioniranje ispitnog uzorka 17
Slika 19. Uklanjanje eliptičnosti 18
Slika 20. Podatci rezultata mjerenja: a) Izgled podataka nakon mjerenja, b) isti podatci prikazani u obliku grafa 19
Slika 21. Podatci rezultata mjerenja prikazani pomoću polarnih koordinata 20
Slika 22. Utjecaj ekscentričnosti na mjerne rezultate 21
Slika 23. Limakon aproksimacija [9] 22
Slika 24. Fourierova transformacija nasumične funkcije [10] 25
Slika 25. Mogući izgledi Fourierove transformacije [11] 26
Slika 26. Reprezentacija točaka grafa frekvencijskog područja 27

Slika 27. Razdvajanje funkcije na podfunkcije [12] 28
Slika 28. Načelo obrade podataka pomoću Fourierove transformacije i Gaussove filtracije 28
Slika 29. Primjer izgleda rješenja diskretne Fourierove transformacije [12] 30
Slika 30. „Long wave pass filter“ [13] 32
Slika 31. Prigušivanje harmonika 33
Slika 32. Funkcija u pythonu 34
Slika 33. Izgled prozora za odabir datoteke 37
Slika 34. Generirani podatci za provjeru učitavanja podataka, a) početni parametri, b) prikaz podataka 53
Slika 35. Prikaz učitanih podataka 53
Slika 36. Generirani podatci za provjeru uklanjanja ekscentra, a) početni parametri, b) prikaz podataka 54
Slika 37. Prikaz uklanjanja ekscentra, a) limakon aproksimacijom, b) Fourierovom transformacijom 54
Slika 38. Generirani podatci za provjeru prigušivanja podataka, a) početni parametri, b) prikaz podataka 55
Slika 39. Prikaz prigušivanja podataka Gaussovim filterom 55
Slika 40. Generirani podatci za provjeru računanja parametara kružnosti, a) početni parametri, b) prikaz podataka 56
Slika 41. Generirani podatci za sveukupnu provjeru programa, a) početni parametri, b) prikaz podataka 57
Slika 42. Uklanjanje ekscentra limakon aproksimacijom 57
Slika 43. Uklanjanje ekscentra Fourierovom transformacijom 58
Slika 44. Prigušivanje podataka Gaussovim filterom 58
Slika 45. Usporedba filtriranih i nefiltriranih generiranih podataka 58
Slika 46. Neobrađeni podatci referentnog prstena 60
Slika 47. Podatci referentnog prstena nakon uklanjanja ekscentra limakon aproksimacijom 60
Slika 48. Podatci referentnog prstena nakon uklanjanja ekscentra Fourierovom transformacijom 61
Slika 49. U potpunosti obrađeni podatci referentnog prstena 61
Slika 50. Usporedba filtriranih i nefiltriranih podataka referentnog prstena 62

POPIS TABLICA

Tablica 1. Metode sa unutrašnjom mjernom referencom 7
Tablica 2. Elementi ticala.....13
Tablica 3. Parametri vezani uz limakon.....23
Tablica 4. Izračunati parametri kružnosti za postavljene parametre 56
Tablica 5. Parametri kružnosti tokom rada programa pri obradi generiranih podataka, a) nakon uklanjanja ekscentra limakon aproksimacijom, b) nakon uklanjanja ekscentra Fourierovom transformacijom, c) nakon prigušivanja Gaussovim filterom, d) promjena parametara, e) promjena parametara, % 59
Tablica 6. Parametri kružnosti tokom rada programa pri obradi izmjerenih podataka, a) nakon uklanjanja ekscentra limakon aproksimacijom, b) nakon uklanjanja ekscentra Fourierovom transformacijom, c) nakon prigušivanja Gaussovim filterom, d) promjena parametara, e)promjena parametara, % 62

POPIS OZNAKA

Oznaka	Mjerna jedinica	Opis oznake
a	µm	udaljenost središta aproksimirane kružnice od ekscentra po x osi
ak	—	koeficijent
a_0	—	amplituda sinusoide prije filtriranja
a_1	—	amplituda sinusoide poslije filtriranja
b	µm	udaljenost središta aproksimirane kružnice od ekscentra po y osi
DEV	µm	razlika između vrijednosti podatka i polumjera kružnice oblika uzorka
$F(f)$	—	funkcija u frekvencijskoj domeni
$F(x_c, y_c, R_c)$	—	„Least squares reference circle“ - konstruirana kružnica
$F(\omega)$	—	funkcija u frekvencijskoj domeni, kružna frekvencija
f	Hz	frekvencija
f_c	Hz	frekvencija od koje se počinje filtrirati
$f(t)$	—	funkcija u vremenskoj domeni
$f\,[k]$	—	očitana vrijednost
k	—	oznaka za broj
N	—	broj očitanih vrijednosti
n	—	oznaka za broj
R_c	µm	polumjer referentne kružnice
$\boldsymbol{R_i}$	µm	udaljenost između točke referentne kružnice i središta kružnice
$R_i(\theta_i)$	µm	niz polumjera kružnice oblika ispitnog uzorka
$RONp$	µm	najviši vrh profila u odnosu na referentnu kružnicu
$RONq$	µm	korijen prosječnog odstupanja profila od referentne kružnice
$RONt$	µm	razlika u radijusu kružnica koje obuhvaćaju predmet
$RONv$	µm	najdublji dol profila u odnosu na referentnu kružnicu
r	µm	polumjer aproksimirane kružnice
t	s	vrijeme
x_c	µm	x vrijednost koordinate središta
x_i	µm	vrijednost podatka
y_c	µm	y vrijednost koordinate središta
ΔR_l	µm	lokalno odstupanje od kružnosti
θ	°	oznaka za kut
θ_i	°	niz kutova
ω	rad/s	kružna frekvencija

POPIS KRATICA

Kratica	**Opis**
ISO	*International organization for standardization* – Međunarodna organizacija za standardizaciju
MZCI	*Minimim zone circles*
LSCII	*Least squares reference circle*
MICII	*Maximum inscribed circle*
MCCI	*Minimum circumscribed circle*
UPR	*Undulatiations per revolution* – broj izbočenja na predmetu

SAŽETAK

Cilj ovog diplomskog rada je napraviti biblioteku funkcija koja se može koristiti za određivanje odstupanja od kružnosti na temelju izmjerenih podataka. Teorijski dio diplomskog rada objašnjava što je kružnost i odstupanje od kružnosti, parametre kružnosti, kojim metodama se mjeri kružnost uz njihove prednosti i mane, te zašto je i koja je metoda odabrana za mjerenje kružnosti ispitnog uzorka. Odabrana metoda detaljno je opisana sa naglaskom na greške koje se mogu pojaviti prilikom mjerenja ispitnog uzorka. Ujedno je opisano kako se te greške uklanjaju obradom signala i pozicioniranjem uzorka. Eksperimentalni dio vezan je uz obradu i filtraciju prikupljenih podataka; napravljena je biblioteka funkcija koja pomoću Fourierove transformacije obrađuje i filtrira podatke, te računa parametre kružnosti. Programski kod biblioteke detaljno je opisan, uz ponuđena objašnjenja korištenih funkcija. Izrađen je dodatni program za generiranje simuliranih podataka mjerenja kružnost u svrhu provjere točnosti i verifikacije izrađene biblioteke. Također su izvršena mjerenja kružnosti u Laboratoriju za precizna mjerenja dužina Fakulteta strojarstva i brodogradnje, na uređaju Mahr MMQ3. Podaci su analizirani i filtrirani korištenjem izrađene biblioteke za mjerenje kružnosti. Na samom kraju rada dana zaključak temeljem analize dobivenih rezultata.

Ključne riječi: kružnost, odstupanje od kružnosti, obrada signala, Fourierova transformacija, filtracija podataka

SUMMARY

The aim of this master's thesis is to create a function library that can be used to determine deviations from roundness based on measured data. The theoretical part of the thesis explains what roundness and deviations from roundness are, the parameters of roundness, the methods used to measure roundness along with their advantages and disadvantages, and why and which method was chosen for measuring the roundness of the test sample. The selected method is described in detail with an emphasis on the errors that may occur during measurements of the test sample. It also describes how these errors are eliminated through signal processing and sample positioning. The experimental part is related to the processing and filtration of the collected data; a function library was created that uses the Fourier transform to process and filter the data and calculates the roundness parameters. The program code of the library is described in detail, with explanations of the functions used. An additional program was created to generate simulated roundness measurement data to verify the accuracy and validation of the developed library. Additionally, roundness measurements were performed in the Laboratory for Precision Length Measurements at the Faculty of Mechanical Engineering and Naval Architecture, on the Mahr MMQ3 device. The data was analyzed and filtered using the developed roundness measurement library. At the very end of the thesis, a conclusion is provided based on the analysis of the obtained results.

Key words: roundness, roundness deviation, signal processing, Fourier transform, data filtering

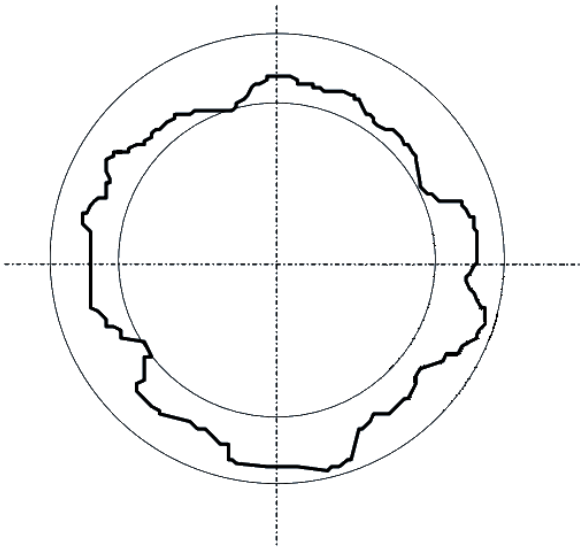
1. UVOD

Izum kotača je jedno od najznačajnijih tehnoloških postignuća čovječanstva, te predstavlja ključnu točku u povijesti civilizacije. Izmišljen oko 3500. godine prije Krista, kotač je revolucionirao transport, poljoprivredu i industriju, te osigurao ubrzani razvoj i širenje ljudske rase po cijelom svijetu. Glavno svojstvo kotača je njegova geometrija, kružnost. Geometrijsko svojstvo kružnosti predmeta vrlo je bitno; omogućuje jednaku raspodjelu sile, pretvorbu linearnog gibanja u kružno gibanje, stezne spojeve itd. U modernom društvu, sav oblik transporta bazira se na kružnosti; kretanje klipova motora koje rotiraju vratilo koje nadalje daje pogon kotačima, sami kotači koji uvelike smanjuju utjecaj trenja i značajno olakšavaju kretanje, brodske i avionske turbine koje pokreću strojeve i omogućavaju prijenos ljudi i dobara preko cijelog svijeta, svi rade na temelju kružnog gibanja komponenata motora i različitih kružnih dosjeda. Kružnost uz transportnu industriju utječe i na građevinu (vijci i matice), sva područja strojarstva (gotovo svaki stroj čije se komponente kreću), brojnih oblike montaže, industriju pakiranja dobara (od limenki za piće do boca za plin), konstrukciju cjevovoda, proizvodnju energije pomoću turbina (sva energija se proizvodi na temelju rotacije turbina) itd. Uistinu, kružnost je osnovno svojstvo gotovo svih strojeva i naprava, te moderno društvo ne bi moglo postojati bez kružnih komponenata strojeva. Međutim, osigurati mala odstupanja od kružnosti nekog predmeta nije jednostavno; odstupanja od kružnosti u strojevima mogu uzrokovati prijevremeni kvar ili kompletan otkaz sustava. Kako bi se osigurala tolerancija kružnosti, razvijene su su mjerne metode pomoću kojih se može pouzdano odrediti odstupanje od kružnosti mjerenog predmeta. Uz mjerne metode, konstruirani su uređaji koji vrlo precizno mogu mjeriti pomake okruglih/rotacionih predmeta. Ovaj diplomski rad detaljno će opisati što je odstupanje od kružnosti, kako se mjeri, kako se obrađuju mjereni podatci, te što se može zaključiti o mjerenom predmetu nakon obrade podataka.

2. KRUŽNOST

2.1. Pojmovi, definicije, greške

Odstupanje kružnosti možemo promatrati kao mjeru koliko neki oblik odstupa od idealne, matematički savršene kružnice. Takva definicija je možda intuitivna, međutim u inženjerskom kontekstu je neprimjenjiva; takvo izražavanje kružnosti predstavlja niz problema: gdje se nalazi centar predmeta u odnosu na referentnu kružnicu, koliki je referentni radijus, kako bi se izražavao stupanj odstupanja od kružnosti i koji bi se parametri koristili za računanje takvog stupnja itd. Kružnost (tj. odstupanje od kružnosti) je prema normi ISO 1101 [1] definirana kao razlika u radijusu dvije koncentrične kružnice koje u potpunosti obuhvaćaju predmet, tj. njegov profil (iz perspektive tlocrta ili na nekom određenom presjeku predmeta), [Slika 1], [2].



Slika 1. Primjeri kružnica koje definiraju kružnost predmeta

Parametri pomoću kojih je kružnost definirana koriste matematički idealnu, referentnu kružnicu, koja je izračunata matematičkim metodama; prije računala referentna kružnica se određivala vizualnom metodom, koja je neprecizna i varira ovisno o osobi koja provodi postupak, te o njihovom iskustvu i vještini. Postoji četiri različite vrste referentnih kružnica [3]:

1. „Minimum zone circles“ (MZCI) - sastoji se od dvije kružnice, jedne upisane u izmjereni profil i jedne opisane mjerenom profilu na takav način da je razlika u njihovom radijusu minimalna. [Slika 2] a).
2. „Least squares reference circle“ (LSCII) - Matematički definirano, to je kružnica čija suma kvadrata udaljenosti od vrhova/dolova profila do referentne kružnice daje minimalan iznos (jednadžba (1)). Ovakva kružnica predstavlja prosjek svih vrhova i dolina, [Slika 2] b).

$$F(x_c, y_{c,} R_c) = Min\left(\sum_{i=1}^{n} R_i^2\right) \quad \textbf{(1)}$$

Gdje je:

$F(x_c, y_c, R_c)$	—	Kružnica koja se konstruira
x_c	µm	x vrijednost koordinate središta
y_c	µm	y vrijednost koordinate središta
R_c	µm	Polumjer referentne kružnice
R_i	µm	Udaljenost između točke referentne kružnice i središta kružnice

3. „Maximum inscribed circle“ (MICII) - Najveća kružnica koja može biti nacrtana u potpunosti unutar profila (jednadžba (2)). Moguće više rješenja, centar se može pomaknuti, [Slika 2] c).

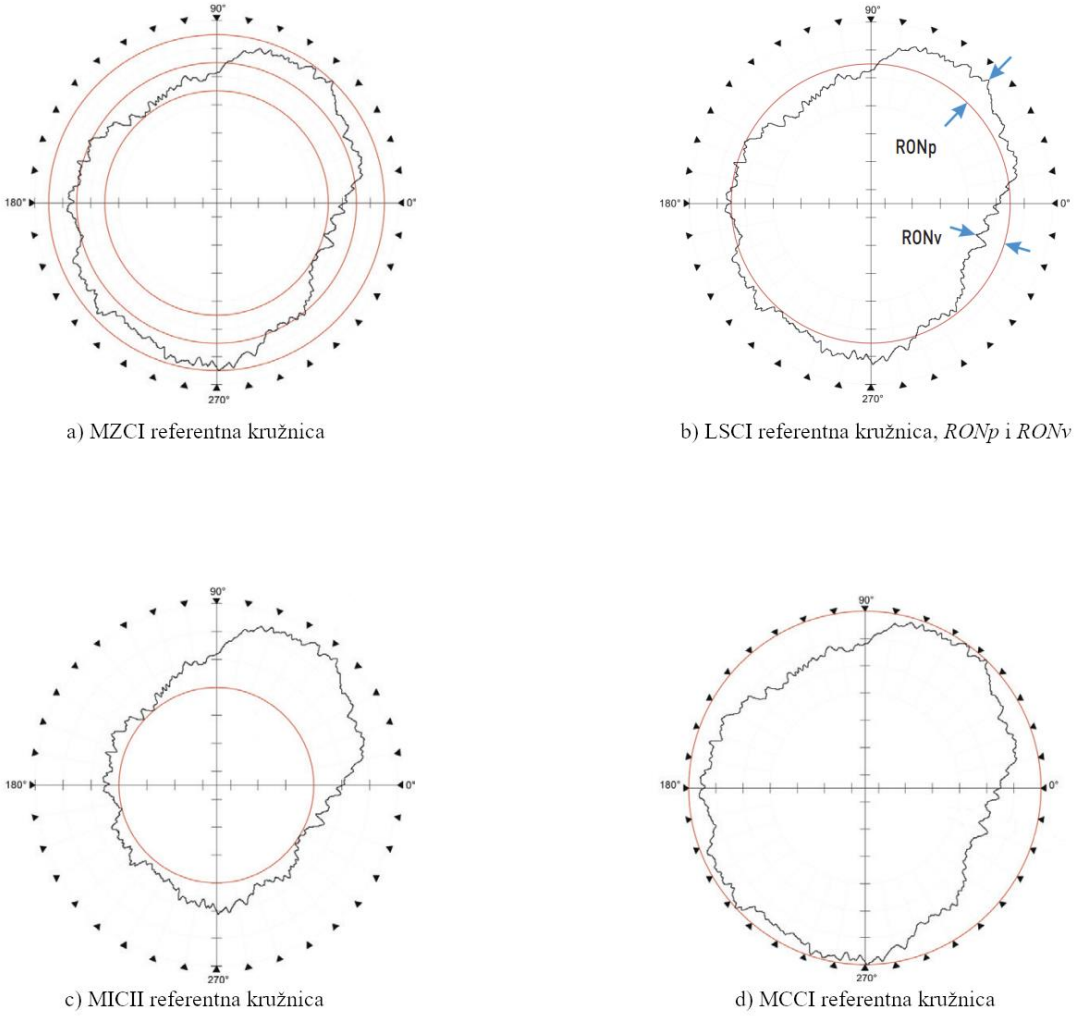
$$F(x_c, y_c) = Max\,\{Min\,[R_i]\} \quad \textbf{(2)}$$

4. „Minimum circumscribed circle“ (MCCI) - Najmanja kružnica koja u potpunosti obuhvaća profil bez da ga dodiruje (jednadžba (3)), [Slika 2] d).

$$F(x_c, y_c) = Min\,\{Max\,[R_i]\} \quad \textbf{(3)}$$

S ovakvom definicijom moguće je pripisati još nekoliko parametara koji se koriste pri određivanju kružnosti [4] :

- *RONt* (roundness total, „peak to valley“) - razlika u radijusu kružnica koje obuhvaćaju predmet, u slučaju sa referentnim kružnicama (LSCII) može se vizualizirati kao visina iz najdublje doline do najvišeg vrha profila, zbroj *RONp* i *RONv*.
- *RONp* - najviši vrh profila u odnosu na referentnu kružnicu (LSCII), [Slika 2] b).
- *RONv* – najdublji dol profila u odnosu na referentnu kružnicu (LSCII), [Slika 2] b).



a) MZCI referentna kružnica

b) LSCI referentna kružnica, *RONp* i *RONv*

c) MICII referentna kružnica

d) MCCI referentna kružnica

Slika 2. Referentne kružnice i parametri, a) MZCI referentna kružnica, b) LSCI referentna kružnica, *RONp* i *RONv*, c) MICII referentna kružnica, d) MCCI referentna kružnica [4]

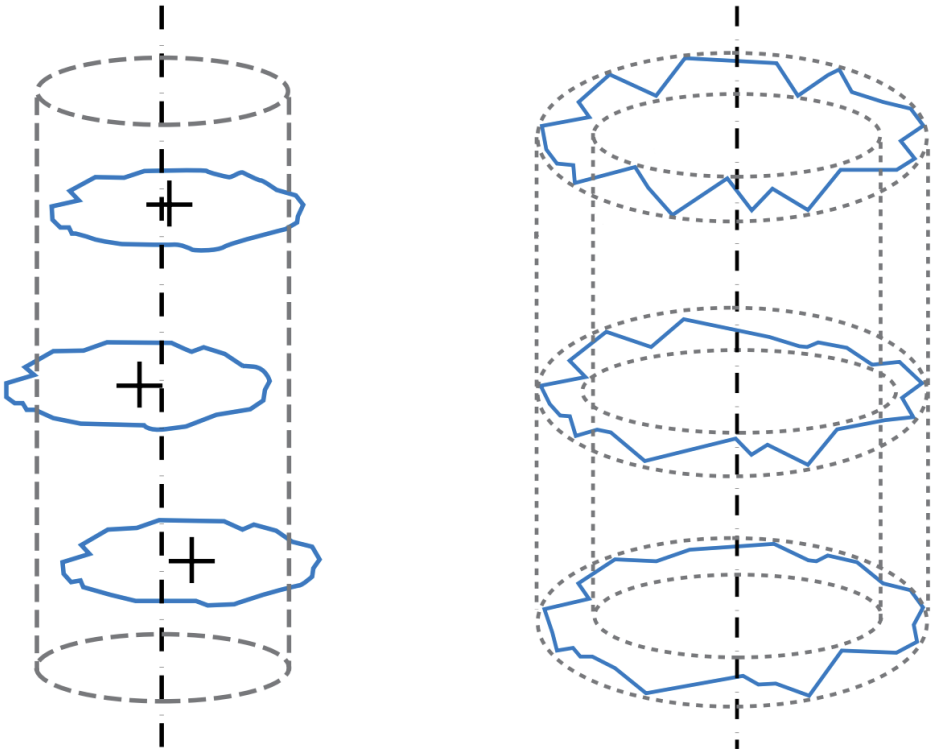
- *RONq* – korijen prosječnog odstupanja profila od referentne kružnice (LSCII), opisan formulom (4), prema normi ISO 12181 [5]:

$$RONq = \sqrt{\frac{1}{2\pi}\int_0^{2\pi} \Delta R_l \, d\theta} \qquad \textbf{(4)}$$

Gdje je:

$RONq$	µm	korijen prosječnog odstupanja profila od referentne kružnice (LSCII)
ΔR_l	µm	lokalno odstupanje od kružnosti
θ	°	kut

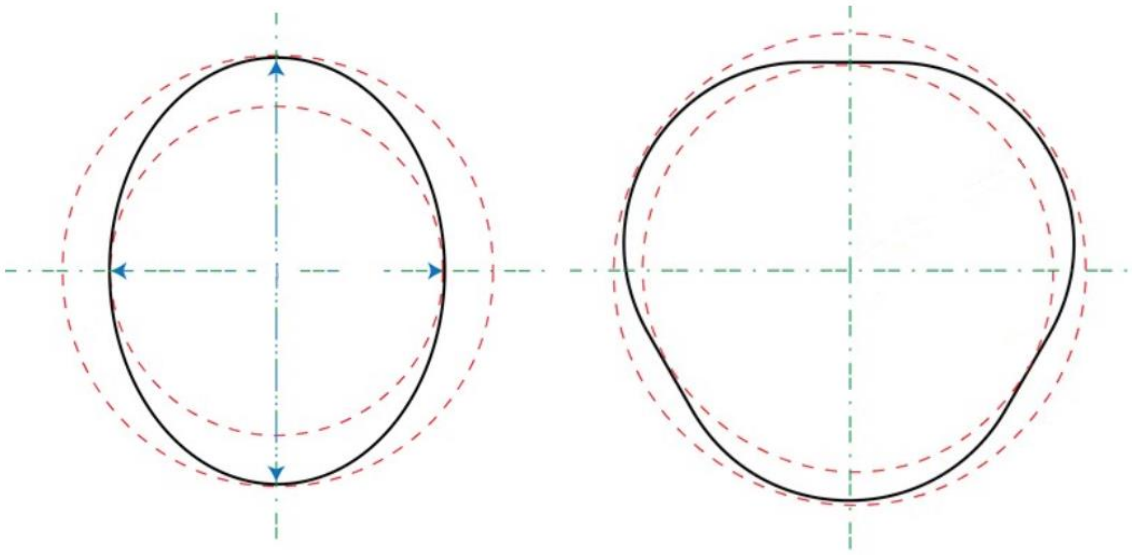
Za razumijevanje pojma kružnosti nadalje je potrebno objasniti razliku između kružnosti, hrapavosti i cilindričnosti. Cilindričnost je svojstvo predmeta koje je slično i usko vezano uz kružnost, te se promatra kao višestruko mjerenje kružnosti duž treće osi, [Slika 3], te posjeduje niz svojih parametara, međutim, cilindričnost je izvan opsega ovog diplomskog rada i neće biti razmatrana.



Slika 3. Cilindričnost [4]

Sljedeće svojstvo koje je blisko kružnosti je hrapavost; ključna razlika je što je kružnost svojstvo oblika, dok je hrapavost svojstvo površine. Hrapavost se promatra na značajno manjem redu veličina (od desetaka mikrometara do manje od mikrometra) i svrha mjerenja hrapavosti je definirati svojstva površine, te kako će se ta površina ponašati u kontaktu sa drugom površinom. Kružnost, s druge strane, promatra oblik predmeta i utjecaj njegove geometrije u nekom sustavu. Greške kod hrapavosti najčešće uzrokuju proklizavanje ili skraćenje životnog vijeka proizvoda, dok greške kod kružnosti mogu uzrokovati lom ili

otkazivanje sustava, nemogućnost montaže/sastavljanja dijelova i prijevremene kvarove strojeva. Odstupanje od kružnosti dijeli se na simetrično i asimetrično (valovitost); najčešća značajna odstupanja su simetrična pošto su ona uzrokovana procesima strojne obrade, te mogu uzrokovati paran ili neparan broj izobličenja. Paran broj izbočenja (primjerice ovalnost profila) uglavnom uzrokuju procesi bušenja i obrade provrta kada je korišten alat istrošen, dok neparan broj izobličenja (izbočenost profila) uzrokuje stezna glava koja drži obradak tokom tokarenja ili drugih postupaka strojne obrade cilindričnih dijelova [Slika 4].



Slika 4. Greške kružnosti nastale tokom strojne obrade, a) ovalnost, b) izbočenost [6]

Značajan parametar za opisivanje oblika pri mjerenju kružnosti je broj izbočenja na predmetu, UPR- „undulatiations per revolution“. Općenito, što je UPR veći, to je svako pojedinačno izbočenje manje; UPR-ovi između 2 i 50 odnose se na izbočenost, preko 50 na valovitost, a UPR-ovi veći od 500 odnose se na hrapavost.

Ovo poglavlje fokusirano je na definiranje što je kružnost i o kojim parametrima ovisi, međutim mjerenje kružnosti predstavlja zaseban problem. Sa temeljnim razumijevanjem što se želi izmjeriti i koji parametri utječu na kružnost, u sljedećem poglavlju bit će objašnjene metode mjerenja kružnosti, te njihove prednosti i mane.

2.2. Mjerenje kružnosti

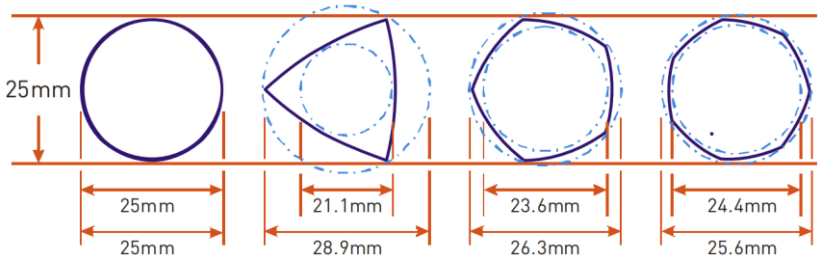
Primarna podjela mjerenja kružnosti je na metode ispitivanja kružnosti sa unutrašnjom mjernom referencom i vanjskom mjernom referencom [7]. Metode unutrašnje mjerne

reference za mjernu referencu koriste jednu ili više točaka s površine kontroliranog izratka, tj. referenca mjerenja se nalazi na predmetu mjerenja. [Tablica 1] opisuje metode unutrašnje mjerne reference i njihove mane.

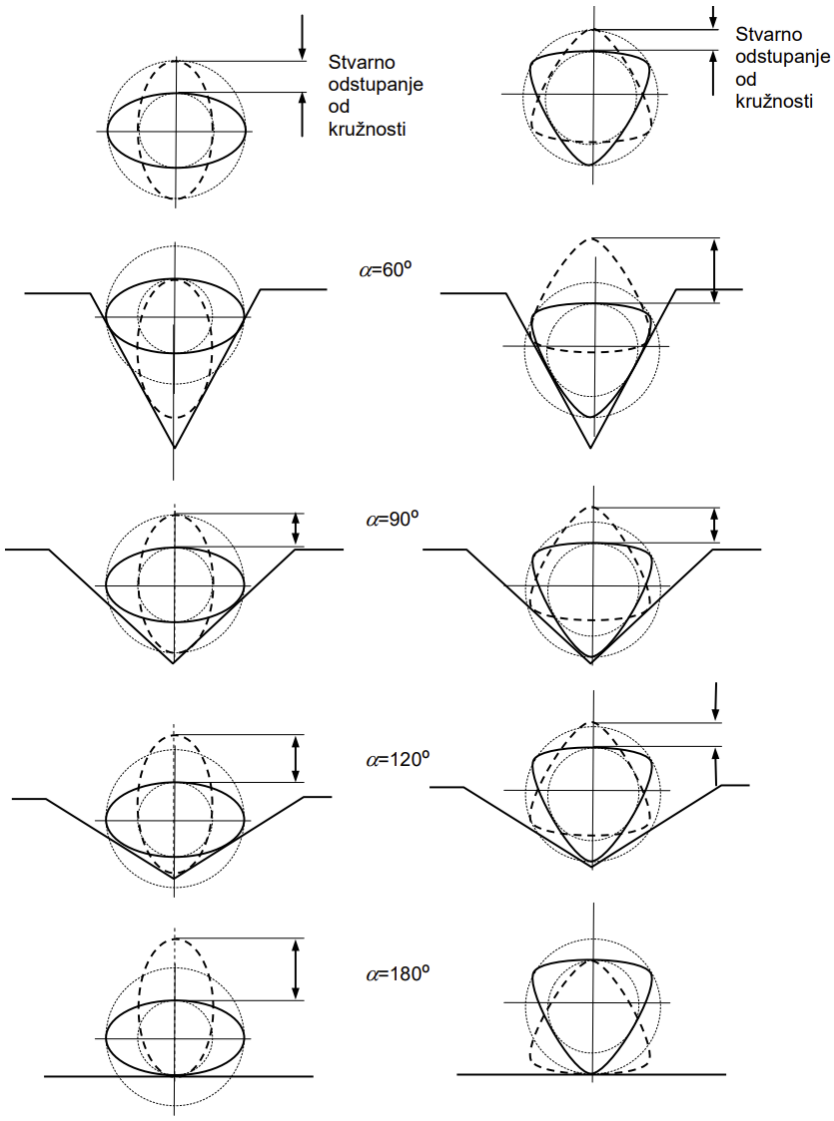
Tablica 1. Metode sa unutrašnjom mjernom referencom

Metoda	Opis	Mane
Dijametralno ispitivanje kružnosti	Mjeri se promjer objekta, koji se nakon mjerenja zarotira za neki kut, te se opet ponavlja mjerenje, [Slika 5].	Postoje oblici koji prilikom mjerenja imaju konstantan promjer, a značajno odstupaju od kružnosti (Reuleauxov mnogokut, [Slika 6]).
Ispitivanje kružnosti primjenom V-prizmi	Predmet se stavlja u prizme V oblika nazivnih kutova 60º, 90º i 120º, te se rotira i bilježi izmjereno odstupanje. S porastom kuta prizme izmjereno odstupanje od kružnosti ovalnih uzoraka raste, a izbočenih uzoraka pada, [Slika 7].	Na rezultat mjerenja utječu i razmaci između izbočenja i udubljenja i njihova interakcija sa nasuprotnih strana predmeta, a ne samo njihova visina, te sam predmet se miče po visini, [Slika 8].
Ispitivanje kružnosti primjenom mjernih šiljaka	Cilindričan predmet natakne se na mjerne šiljke, rotira, te se mjere pomaci na odabranoj poziciji, [Slika 9].	Mogućnost nepravilnog pozicioniranja šiljaka na objekt (nisu u centru), šiljci nisu na istoj osi, predmet se deformira zbog dužine, [Slika 10].

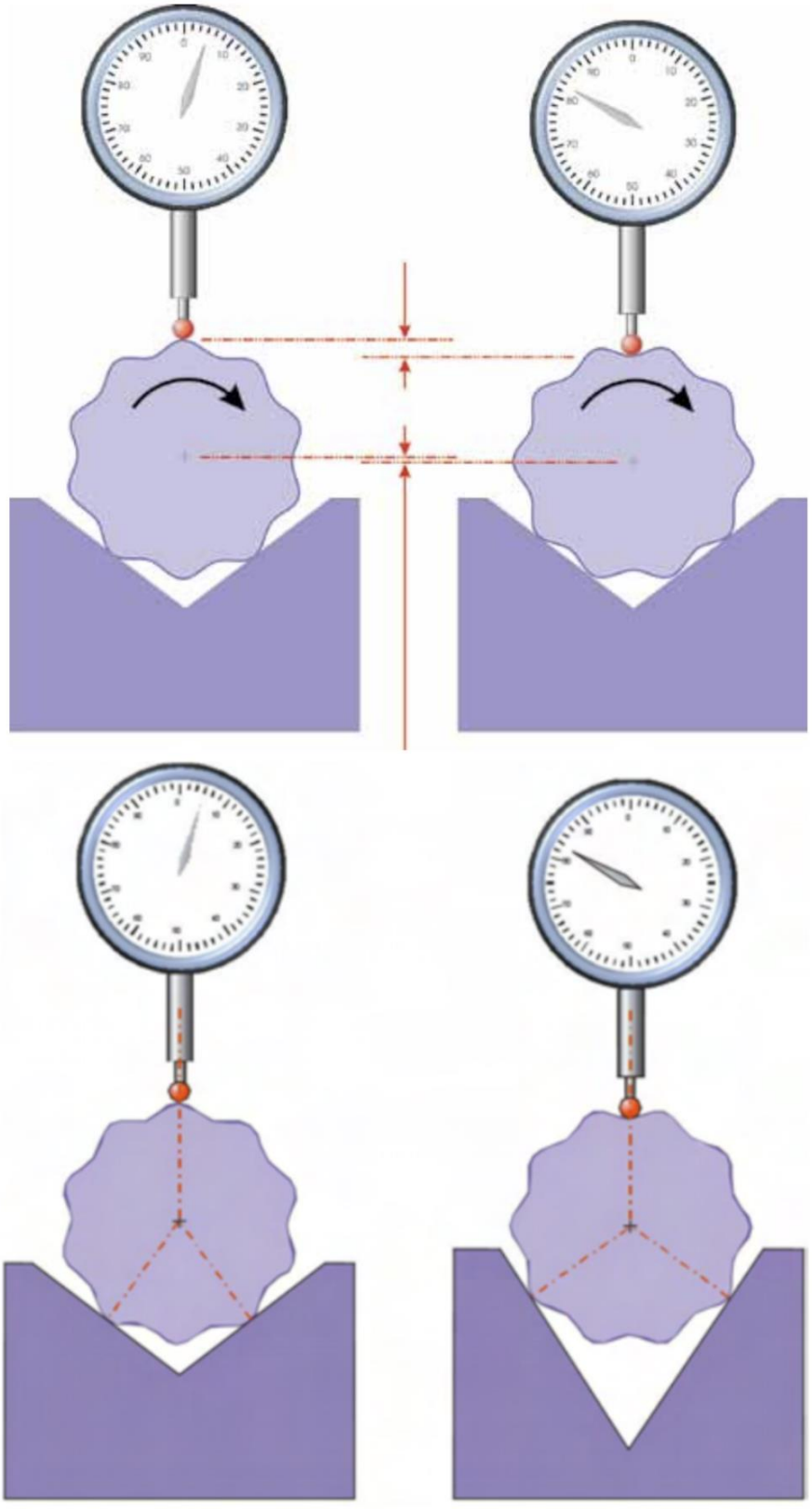
Slika 5. Mjerenje promjera na različitim mjestima [4]



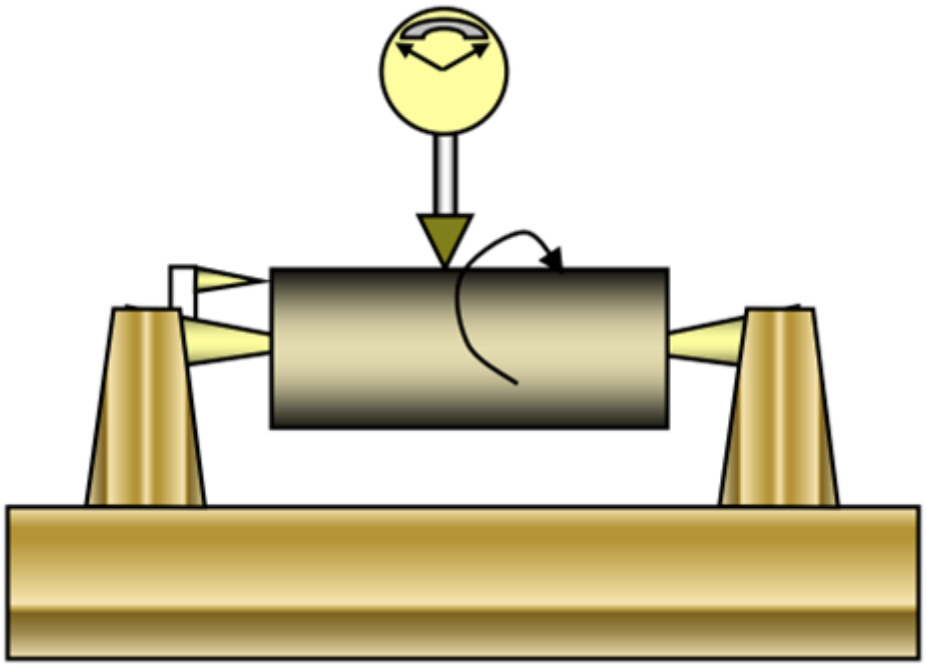
Slika 6. Oblici profila koji pri mjerenju pomičnim mjerilom daju istu vrijednost [4]



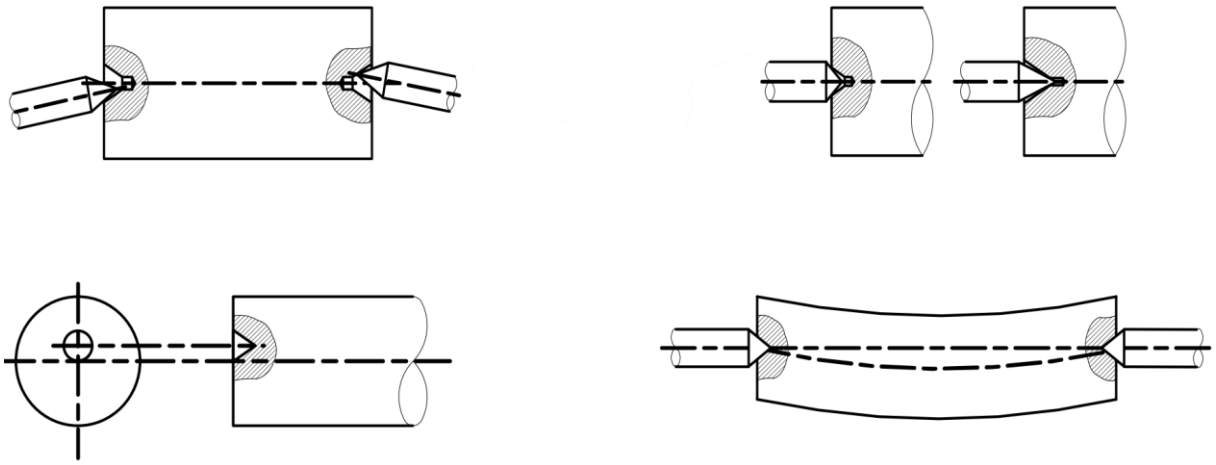
Slika 7. Ispitivanje kružnosti primjenom V-prizmi [7]



Slika 8. Greške prilikom mjerenja pomoću V-prizmi [4]



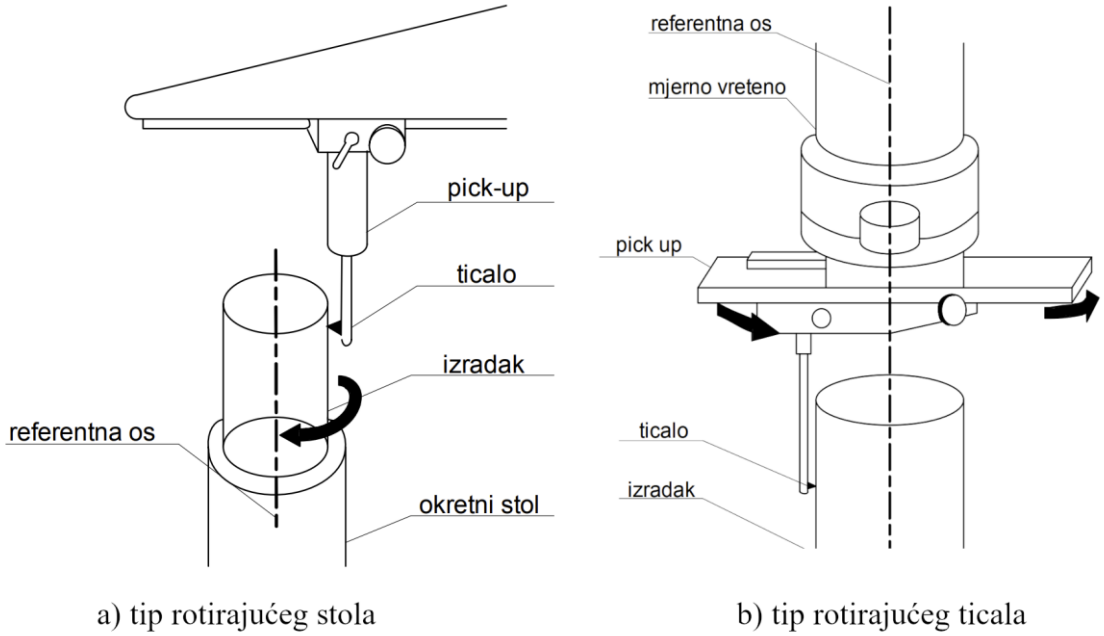
Slika 9. Ispitivanje kružnosti primjenom mjernih šiljaka [7]



Slika 10. Greške prilikom ispitivanja kružnosti primjenom mjernih šiljaka [2]

Općenito metode ispitivanja kružnosti koriste se za ispitivanje je li profil pravilan, ali se ne može dobiti veličina odstupanja od kružnosti, stoga se ove metode ne koriste kod preciznog mjerenja kružnosti, već za brzu provjeru oblika, često unutar samog postrojenja.

Kod metoda sa vanjskom mjernom referencom za ispitivanje kružnosti za mjernu referencu se uzima os rotacije stroja, tj. vrlo precizno izrađenog vretena [2]. Ove metode za ispitivanje kružnosti daju stvarnu sliku geometrijskog stanja ispitane površine (uz potrebnu računalnu obradu podataka). Postoje dva tipa uređaja za ispitivanje kružnosti sa vanjskom mjernom referencom; tip rotirajućeg stola [Slika 11] a) i tip rotirajućeg ticala [Slika 11] b).

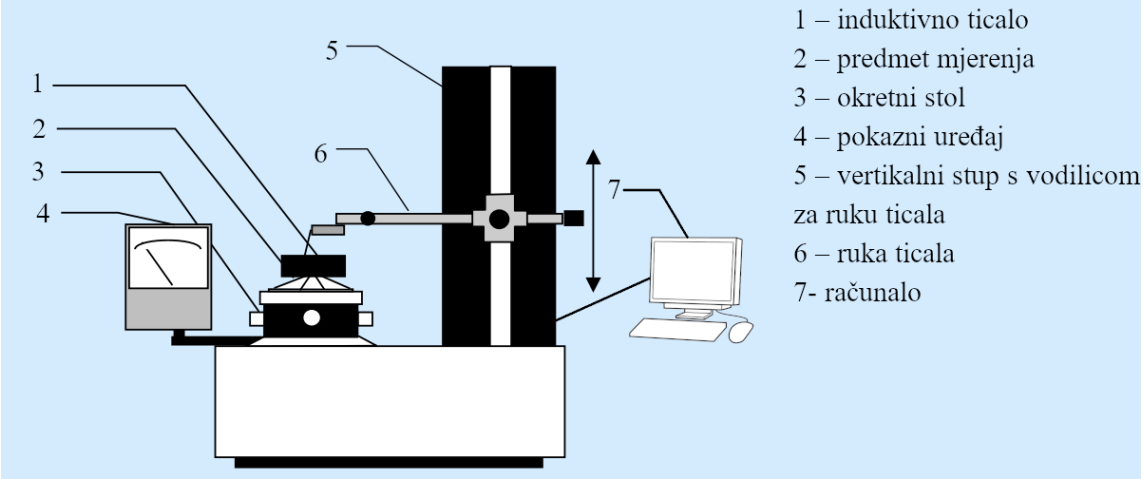


Slika 11. Mjerenje sa vanjskom mjernom referencom, a) tip rotirajućeg stola, b) tip rotirajućeg ticala [2]

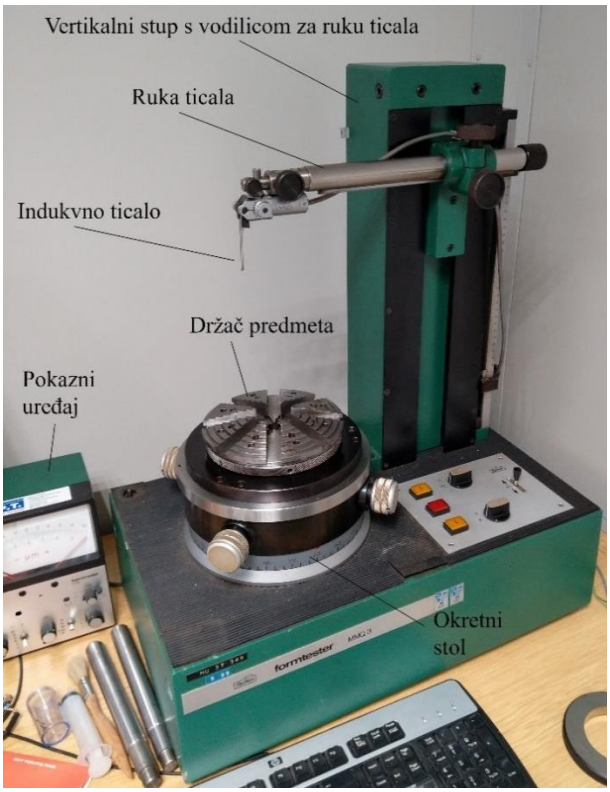
Kod tipa rotirajućeg stola, predmet se rotira dok je ticalo statično, a kod tipa rotirajućeg ticala predmet je statičan dok se ticalo okreće. Za ovaj diplomski rad koristi se uređaj sa rotirajućim stolom, te će u sljedećem poglavlju biti detaljno opisan.

2.2.1. Komponente stroja za mjerenje kružnosti sa rotirajućim stolom i princip rada

Sa razumijevanjem kako se mjeri kružnost i koja je metoda odabrana i zašto, potrebno je detaljnije opisati princip rada stroja. [Slika 12] shematski prikazuje sve osnovne komponente potrebne za provedbu mjerenja, dok [Slika 13] prikazuje isti uređaj u Laboratoriju za precizna mjerenja dužina na Fakultetu strojarstva i brodogradnje.

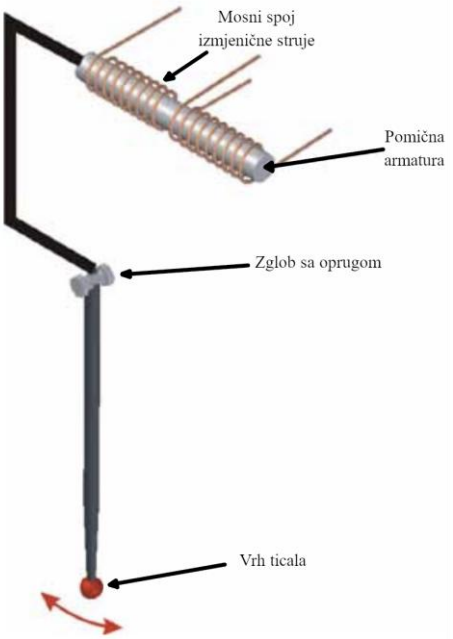


Slika 12. Shema uređaja za ispitivanje kružnosti primjenom okretnog stola [2]



Slika 13. Uređaj za ispitivanje kružnosti primjenom okretnog stola

Ključne komponente stroja su ticalo i okretni stol. [Tablica 2] opisuje građu ticala i svrhu/zadatak pojedinačnog elementa, te [Slika 14] daje izgled ticala.



Slika 14. Građa ticala [4]

Tablica 2. Elementi ticala

Element	Svrha
Vrh ticala	Naslonjen na ispitni uzorak, miče se u jednoj dimenziji, naprijed ili natrag.
Zglob sa oprugom	Drži vrh ticala u kontaktu na ispitnim uzorkom i prenosi gibanje vrha ticala na pomičnu armaturu. Početna pozicija vrha ticala je takva da stalno djeluje sa malom silom na uzorak, kako bi se osiguralo da se vrh ticala miče kada tokom ispitivanja dođe do izbočenja ili udubljenja na predmetu.
Pomična armatura	Zajedno rade kako bi generirali struju. U početnoj poziciji vrha ticala mosni spoj je u balansu i ne generira struju. Pomicanjem vrha ticala tj. armature generira se električni signal koji se šalje u računalo i pretvara u podatke.
Mosni spoj izmjenične struje	

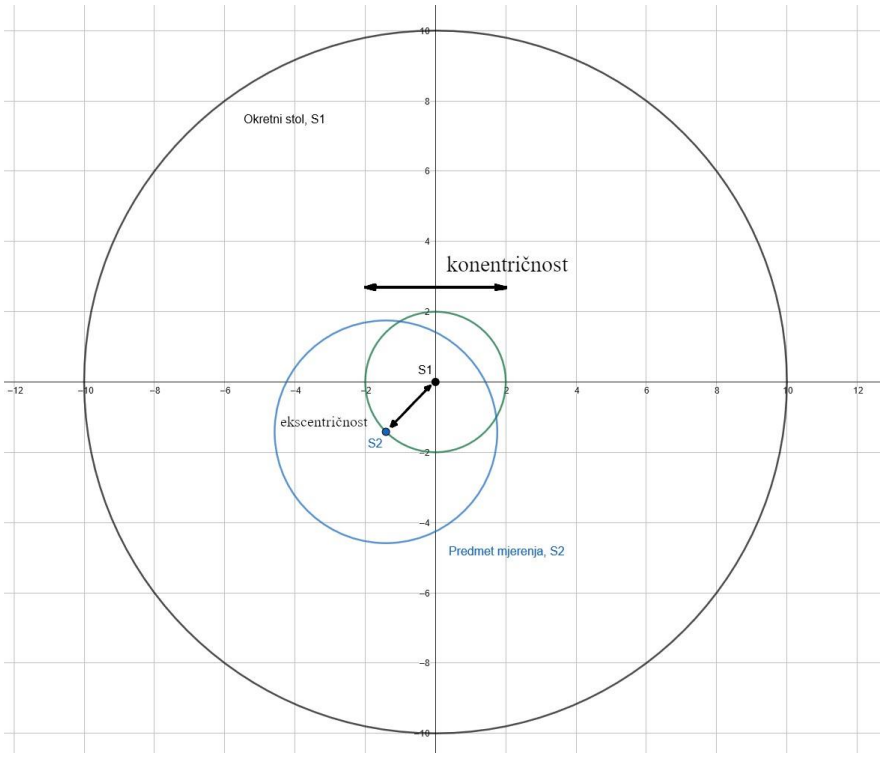
Ticalo je naslonjeno na ispitni uzorak koji stoji na okretnom stolu i rotira se. Samo ticalo postavljeno je tako da uvijek sa malom silom djeluje na ispitni uzorak, kako bi uvijek bilo u kontaktu sa njime. Kako se uzorak okreće, nepravilnosti njegove površine pomiču ticalo; ako ticalo dođe do izbočenja na predmetu pomaknut će se u jednom smjeru, a ako naleti na udubljenje na predmetu pomaknuti će se u suprotnom smjeru. Ovisno o smjeru gibanja vrha ticala, generira se odgovarajući električni signal koji se šalje uračunalo koje taj signal interpretira i pretvara u podatke s kojima se dalje ispituje odstupanje od kružnosti.

Općenite karakteristike metode mjerenja kružnosti pomoću stroja s rotirajućim stolom uključuju [8]:

- Os rotacije je mjerna referenca te se tako osigurava ponovljivost rezultata ispitivanja
- Mala osjetljivost sustava na trenutne promjene temperature.
- Može se jednostavno mjeriti i cilindričnost.
- Referentna os rotacije mora biti iznimno točna kako bi se dobili dobri rezultati.

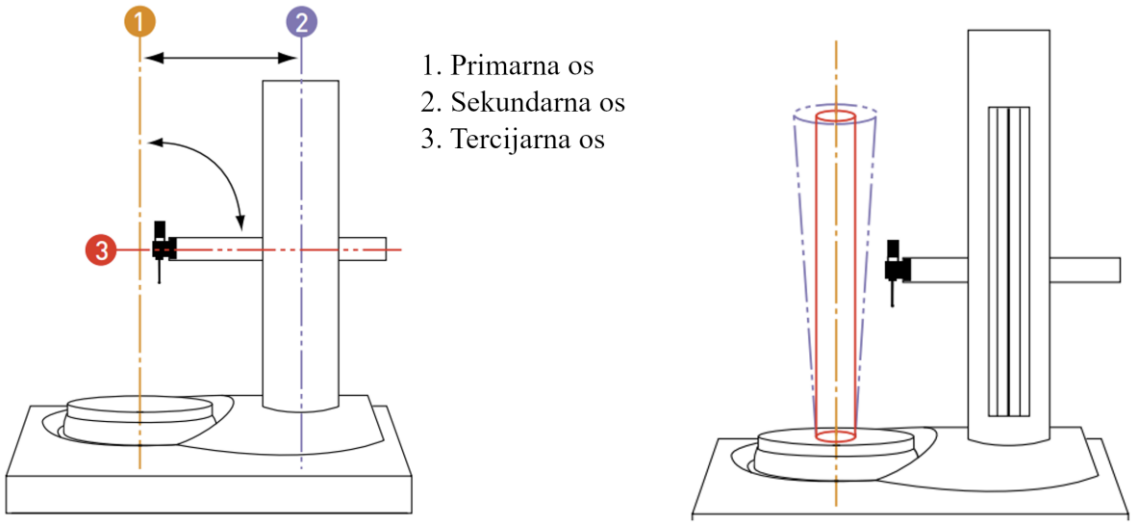
2.2.2. Moguće greške prilikom mjerenja

Greške prilikom mjerenja mogu se podijeliti u dvije skupine: greške pozicioniranja predmeta i greške uzrokovane strojem. Greške pozicioniranja predmeta su neizbježne, te se kasnije korigiraju obradom dobivenih podataka. Za bolje razumijevanje ovakve greške potrebno je objasniti koncentričnost, parametar kružnosti opisan kao mjera razmaka između središta referentne kružnice (matematičkog središta ispitnog uzorka) i referentne mjerne osi, tj. centra okretnog stola. Prema ISO 1101 [1] koncentričnost je definirana kao promjer kružnice čije se središte preklapa sa točkom interesa, u ovom slučaju središte okretnog stola, a rubom obuhvaća središte referentne kružnice. Usko vezano uz koncentričnost je ekscentričnost, koji nije službeni ISO parametar, ali je koristan kod obrade podataka. Ekscentričnost je udaljenost između središta referentne kružnice i središta okretnog stola, te iznosi pola vrijednosti koncentričnosti, [Slika 15].



Slika 15. Koncentričnost i ekscentričnost

Greške uzrokovane strojem sprječavaju se umjeravanjem uređaja, koje je provedeno prije mjerenja, te nije u sklopu ovog diplomskog rada. Unatoč tome, dobro je znati koje se greške mogu pojaviti; greške uzrokovane odstupanjem geometrije stroja, [Slika 16], i greške pri generiranju i slanju signala.

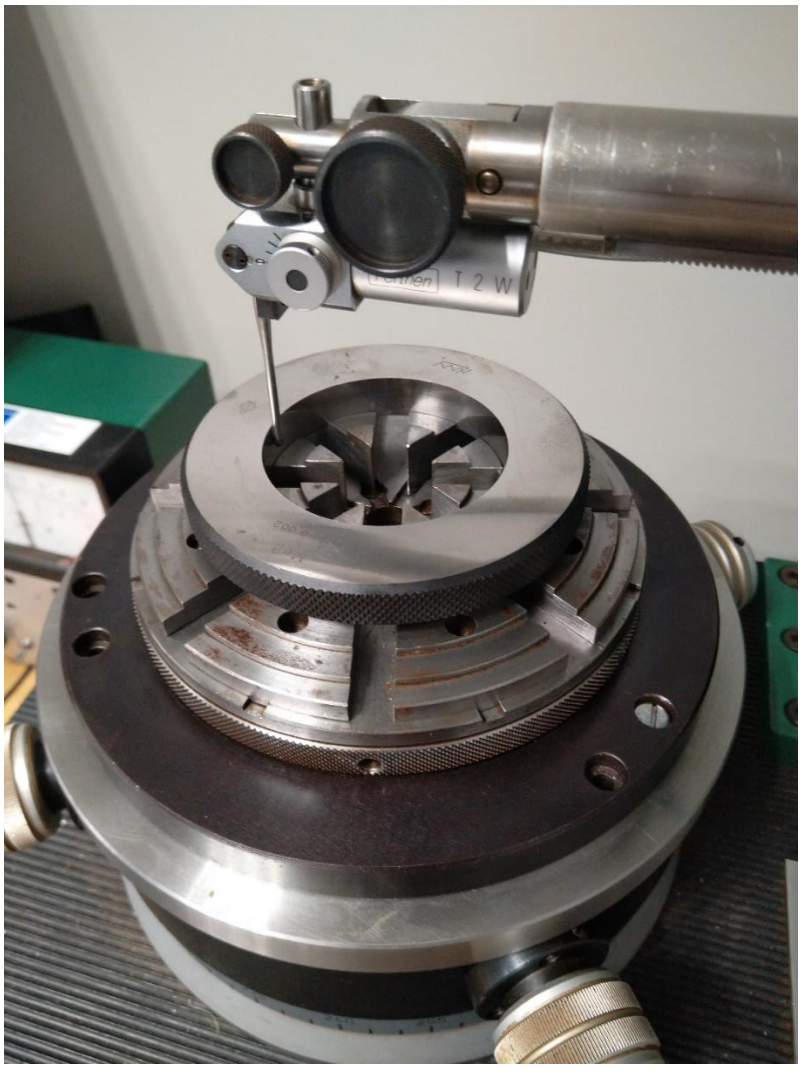


Slika 16. Moguće greške geometrije uređaja [4]

[Slika 16] će služiti kao referenca za opis grešaka; greške prvog, drugog i trećeg reda, na temelju toga na koju se os odnose. Greške prvog reda uzrokovane su kada referentna mjerna os nije vertikalna, te se stvara dodatna ekscentričnost na mjernim podatcima. Greške drugog reda uzrokuju grešku konusa što je ispitni uzorak viši. Greške trećeg reda promijeniti će kut pod kojim ticalo dodiruje predmet, i tako promijeniti podatke koji dolaze do računala.

2.2.3. Postupak mjerenja kružnosti

Uređaj se koristi na sljedeći način: predmet kojemu se mjeru kružnost stavlja se na okretni stol, te se pozicionira da je na centru okretnog stola, te se ticalo stavlja tako da je krak ticala paralelan sa ravninom predmeta na koju se vrh ticala naslanja, [Slika 17];

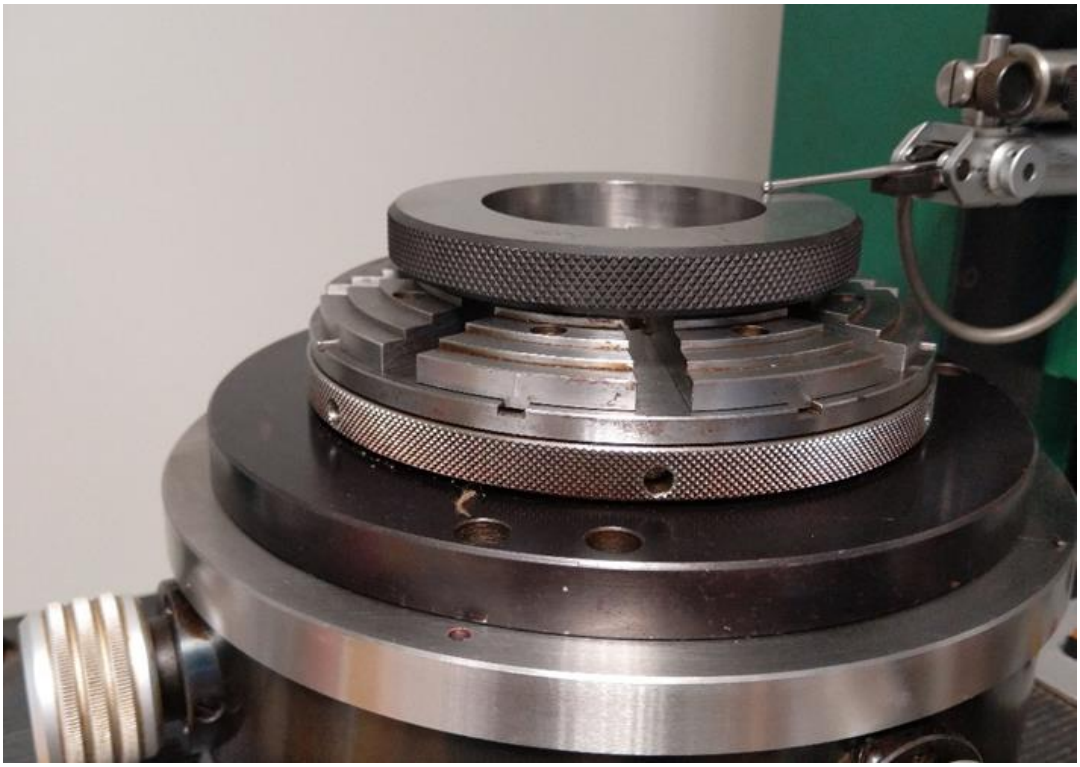


Slika 17. Stavljanje predmeta na uređaj

Predmet se stavlja na okretni stol, i od oka se centrira, te se na njega nasloni ticalo. Potom se učvrsti sa četiri strane pomoću držača, te se rotira. Istovremeno se prati očitanje (koji je postavljan na minimalnu osjetljivost) tj. kretanje kazaljke na pokaznom uređaju. Sa okretanjem predmeta kazaljka će se micati i obraća se pozornost na njeno kretanje, te se traži sljedeći obrazac ponašanja; kada kazaljka napravi veliki pomak obraća se pozornost hoće li napraviti pomak u suprotnu stranu kada se predmet zarotira za 180° [Slika 18], drške označene sa 1 i 2 kako bi se lakše pratila rotacija okretnog stola. To znači da je prisutan ekscentar, tj. da se os rotacije predmeta i os rotacije stroja ne preklapaju. Potom se stezanje držača i pozicija predmeta adekvatno prilagodi te se postupak ponavlja.

Slika 18. Pozicioniranje ispitnog uzorka

Kada se pomaci više ne mogu uočiti povećava se osjetljivost pokaznog uređaja te se procedura ponavlja dok konačno više nema velikih pomaka pokazne kazaljke. Tada je predmet centriran, međutim i dalje je prisutna ekscentričnost malog iznosa koju je potrebno ukloniti (više u sljedećem poglavlju). Prije nego što mjerenje započne, potrebno je osigurati i horizontalnost uzorka; da je gornja ploha uzorka paralelna sa površinom okretnog stola. u slučaju da uzorak nije horizontalno postavljen, prilikom rotacije u podatcima će biti prisutna eliptičnost. Postupak osiguranja horizontalnosti uzorka analogan je postupku uklanjanja ekscentra, samo se ticalo stavlja na aksijalnu površinu, [Slika 19]. Prati se micanje kazaljke na pokaznom uređaju, pozicija uzorka se prilagodi, poveća se osjetljivost pokaznog uređaja i postupak se ponavlja.



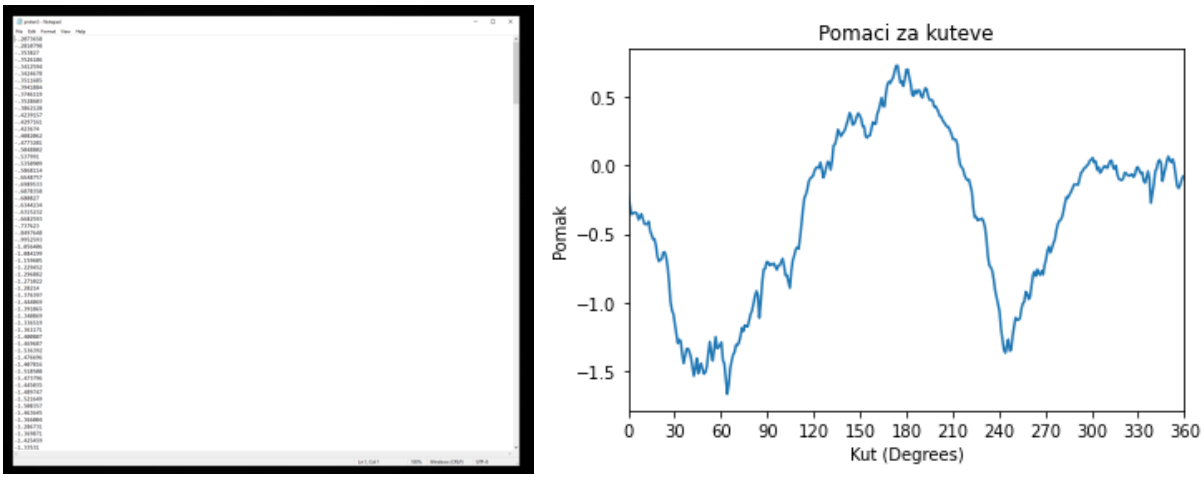
Slika 19. Uklanjanje eliptičnosti

.

3. NAČELA OBRADE PODATAKA

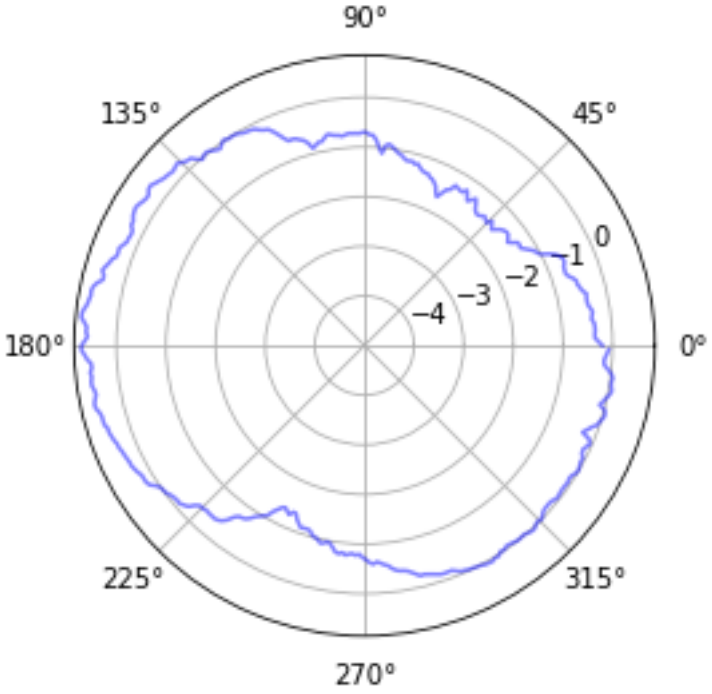
3.1. Prikupljeni podatci i njihov prikaz

Prije obrade podataka potrebno je znati u kojem formatu se nalaze mjerni podatci, što predstavljaju i kako ih tumačiti za daljnju interpretaciju i korištenje. Nakon provedenog mjerenja računalo podatke ispisuje u .txt ili .dat formatu, kao niz brojeva. Svaki broj predstavlja pomak vrha ticala od njegove početne pozicije za određeni kut rotacije okretnog stola; kada bi bio set podataka od 36 brojeva, svaki broj bi predstavljao pomak ticala za 10° rotacije okretnog stola. Uobičajeni broj prikupljenih podataka iznosi između 300 i 1500. [Slika 20] prikazuje sirovu .txt/.dat datoteku koju računalo generira nakon mjerenja, te su radi bolje preglednosti ti podatci prikazani kao graf sa vrijednostima brojeva i odgovarajućim kutom.



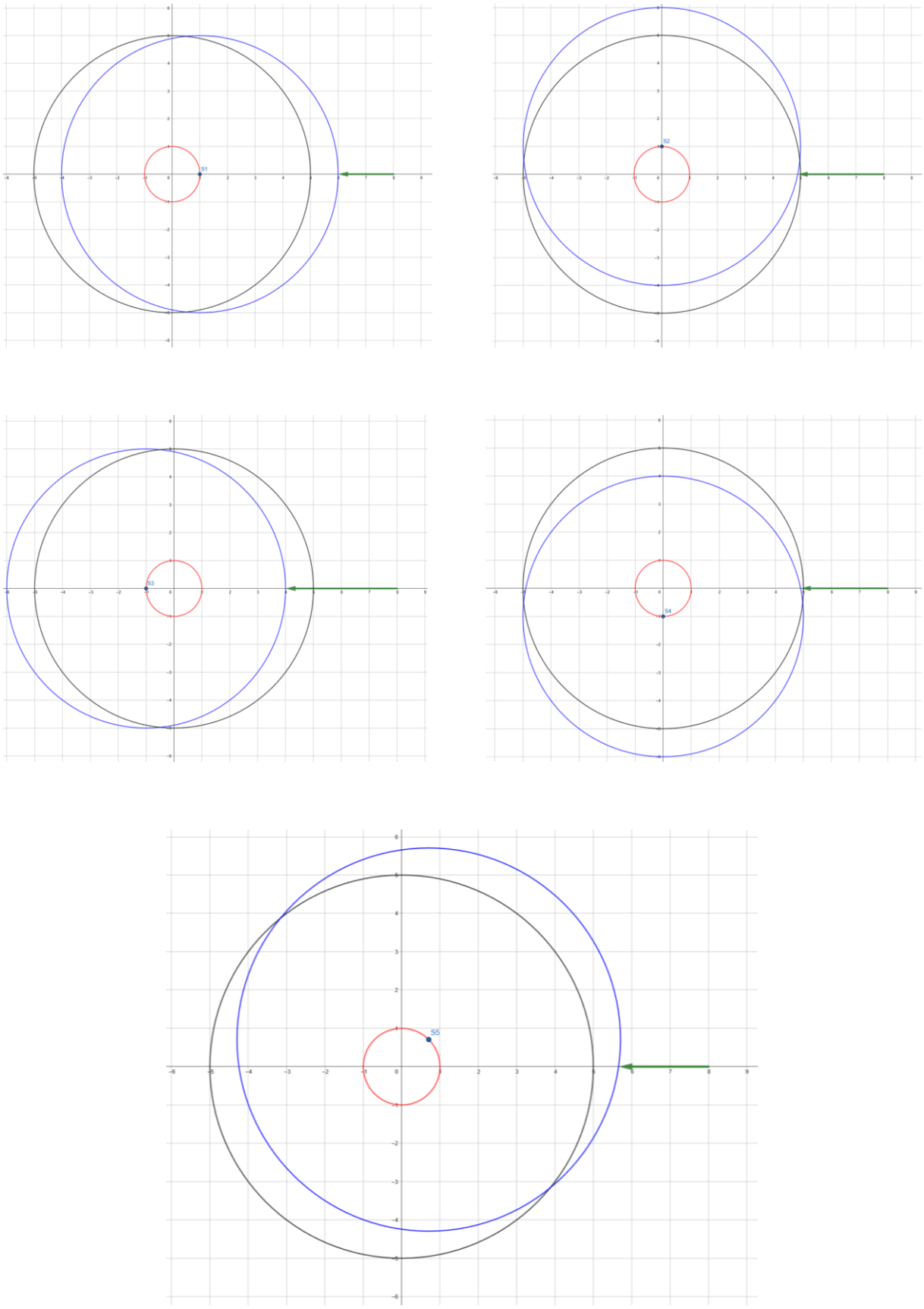
Slika 20. Podatci rezultata mjerenja: a) Izgled podataka nakon mjerenja, b) isti podatci prikazani u obliku grafa

Još prikladniji prikaz podataka bi bio u polarnim koordinatama, kao da je takav graf „omotan“ oko kružnice; takva reprezentacija podataka se približava ciljanom izgledu grafa koji se koristi za predstavljanja odstupanja od kružnosti [Slika 21].



Slika 21. Podatci rezultata mjerenja prikazani pomoću polarnih koordinata

Međutim, prije početka analize kružnosti potrebno je ispraviti grešku ekscentričnosti koja se nalazi u podatcima. U prošlom poglavlju naglašeno je da prilikom postavljanja ispitnog uzorka na stroj uvijek postoji greška ekscentričnosti koja je uključena u rezultate mjerenja i mjerne podatke; pomak ticala nije reprezentativan svojstvu oblika ispitnog uzorka, već predstavlja njegovo kretanje na okretnom stolu. [Slika 22] shematski prikazuje kako se ponaša ticalo u slučaju prisustva ekscentričnosti ispitnog uzorka. Crno je označena kružnica sa središtem u ishodištu koordinatnog sustava (0,0), polumjera 5. Crveno je označena ekscentričnost: kružnica po kojoj se kreće središte referentne kružnice, označena plavo (kružnica ekscentra polumjera 1 i referentna kružnica polumjera 5). Zelena strelica predstavlja ticalo. Promatra se pet pozicija: kada je središte referentne kružnice u (1,0), u (0,1), u (-1,0), u (0, -1) i kada se središte nalazi na 45°, približno (0,707, 0,707). U slučaju da je referentna kružnica savršeno pozicionirana na (0,0) ticalo se uopće ne bi pomaklo, s obzirom da je u ovom primjeru referentna kružnica matematički savršena.



Slika 22. Utjecaj ekscentričnosti na mjerne rezultate

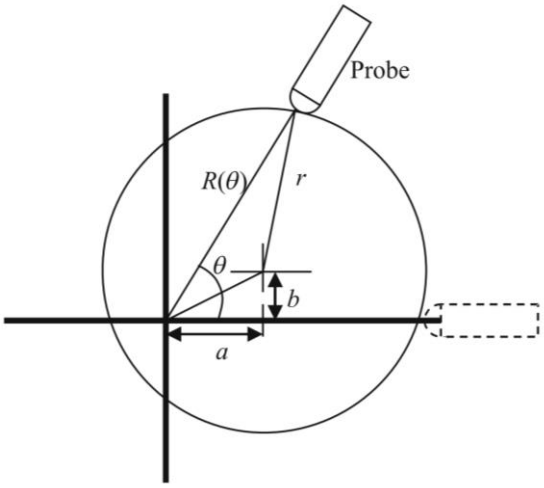
Kao što je vidljivo, vrh ticala značajno se pomiče, unatoč tome što je kružnica matematički savršena. Kada bi se pratila putanja jedne odabrane točke na referentnoj kružnici dok se vrti sama oko sebe, te joj se središte vrti po nekoj drugoj kružnici, putanja bi iscrtala limakon. [Slika 21] koja prikazuje prikupljene podatke u polarnim koordinatama upravo predstavlja limakon, a ne željenu kružnicu koja se dalje koristi za ispitivanje kružnosti. Stoga prije računanja ikakvih parametara, potrebno je ukloniti ekscentričnost iz podataka, kako bi se osiguralo da prikupljeni brojevi uistinu predstavljaju samo pomake uzrokovane oblikom mjerenog ispitnog uzorka.

3.2. Uklanjanje ekscentričnosti

Postoje dva načina na koja se može ukloniti ekscentričnost; limakon aproksimacijom i korištenjem Fourierove transformacije. Oba pristupa daju iste rezultate, koji se razlikuju jedino zbog broja decimala i zaokruživanja brojeva. U radu koristit će se obje metode te će se međusobno usporediti dobiveni rezultati.

3.2.1. Uklanjanje ekscentričnosti limakon aproksimacijom

Uklanjanje ekscentričnosti limakon aproksimacijom svodi se na matematičke jednadžbe koje se koriste kako bi se iz podataka koji čine limakon izračunali parametri koji se mogu koristiti u daljnjem određivanju kružnosti. Nadalje, iz limakona se računa aproksimirana kružnica koja se vrti oko ekscentra, njen promjer, i pozicija njenog središta u odnosu na globalno središte. [Slika 23] prikazuje sve parametre koji se računaju.



Slika 23. Limakon aproksimacija [9]

Osim parametara koji su prikazani na slici, za računanje su potrebne varijable vezane uz samu datoteku mjerenja, [Tablica 3] prikazuje sve potrebne veličine vezane uz aproksimaciju limakona:

Tablica 3. Parametri vezani uz limakon

a	µm	udaljenost središta aproksimirane kružnice od ekscentra po x osi
b	µm	udaljenost središta aproksimirane kružnice od ekscentra po y osi
r	µm	polumjer aproksimirane kružnice
θ_i	°	niz kutova
$R_i(\theta_i)$	µm	niz polumjera kružnice oblika ispitnog uzorka
n	—	ukupan broj podataka
x_i	µm	vrijednost podatka
DEV	µm	niz razlika između vrijednosti podatka i polumjera kružnice oblika uzorka (odstupanje svake točke profila od referentne kružnice)

Najprije je potrebno izračunati sve kutove θ i r. Njihovo računanje je jednostavno; θ se računa na temelju ukupnog broja podataka prema jednadžbi (5):

$$\theta = \frac{360°}{n} \cdot \frac{\pi}{180°} \tag{5}$$

r se računa kao prosječna vrijednost svih prikupljenih podataka, jednadžba (6):

$$r = \left(\sum_{i=1}^{n} x_i \right) / n \tag{6}$$

a se računa po formuli (7):

$$a = 2 \cdot \left(\left(\sum_{i=1}^{n} (x_i \cdot cos\,\theta) \right) / n \right) \tag{7}$$

dok se b računa prema formuli (8):

$$b = 2 \cdot \left(\left(\sum_{i=1}^{n} (x_i \cdot sin\,\theta) \right) / n \right) \tag{8}$$

Sa svim potrebnim podatcima, može se izračunati $R_i(\theta_i)$ prema formuli (9):

$$R_i(\theta_i) = r + a \cdot cos(\theta_i) + b \cdot sin(\theta_i) \quad \textbf{(9)}$$

I konačno, računa se uvedena veličina *DEV* koja će se dalje koristiti kod određivanja kružnosti, jednadžba (10):

$$DEV = x_i - R_i(\theta_i) \quad \textbf{(10)}$$

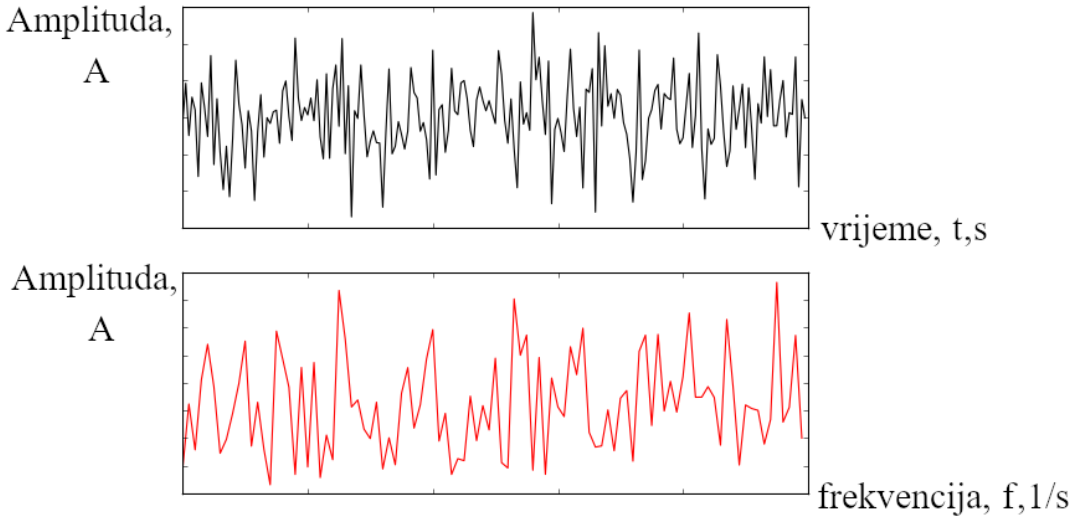
Nakon računanja, određen je ekscentar i aproksimirana je kružnica koja se vrti oko njega, te je izračunat polumjer kružnice oblika ispitnog uzorka za pojedinačne kutove. Ovim pristupom nije potrebno crtati referentnu kružnicu, pošto sama referentna kružnica služi kako bi se uklonio utjecaj ekscentra. Parametri koji opisuju kružnost (*RONp* i *RONv*) sada su izraženi kao najmanja i najveća vrijednost *DEV*. Ova metoda je preciznija i podatke koji se koriste za određivanje odstupanja od kružnosti.

3.2.2. Uklanjanje ekscentričnosti Fourierovom transformacijom

Drugi, značajno elegantniji pristup kojim se može ukloniti ekscentričnost je primjenom Fourierove transformacije koja će nadalje poslužiti kao moćan alat u obradi signala. U sljedećem poglavlju biti će detaljnije objašnjena Fourierova transformacija, te nakon što su objašnjeni temeljni koncepti, biti će opisan način uklanjanja ekscentričnosti. Ukoliko je pomoću Fourierove transformacije uklonjen nulti i prvi harmonik iz signala kružnosti dobije se centrirani profil s korekcijom limakona, više u sljedećem poglavlju.

3.3. Fourierova transformacija

Fourierova transformacija je je matematički postupak koji uzima vremensku funkciju i transformira ju u frekvencijsku funkciju. Konkretno, uzima funkciju u vremenskom području (signal, koji je definiran sa amplitudom i vremenom) i transformira ju u novu funkciju, ali u frekvencijskom području (isti signal, ali predstavljen kao funkcija frekvencije i amplitude). Svrha takve transformacije je mogućnost matematičke obrade podataka, poput filtracije, prigušenja, uklanjanje šuma i sl. [Slika 24] prikazuje Fourierovu transformaciju nasumične funkcije i njezinu Fourierovu transformaciju u frekvencijskom području.



Slika 24. Fourierova transformacija nasumične funkcije [10]

Fourierova transformacija provodi se pomoću izraza (11):

$$F(f) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi j f t} dt \qquad \textbf{(11)}$$

Gdje je:

f(t)		funkcija u vremenskoj domeni
F(f)		funkcija u frekvencijskoj domeni
t	s	vrijeme
f	1/s	frekvencija

Ako se uvede definicija za kružnu frekvenciju (12), izraz (11) može se zapisat kao jednadžba (13);

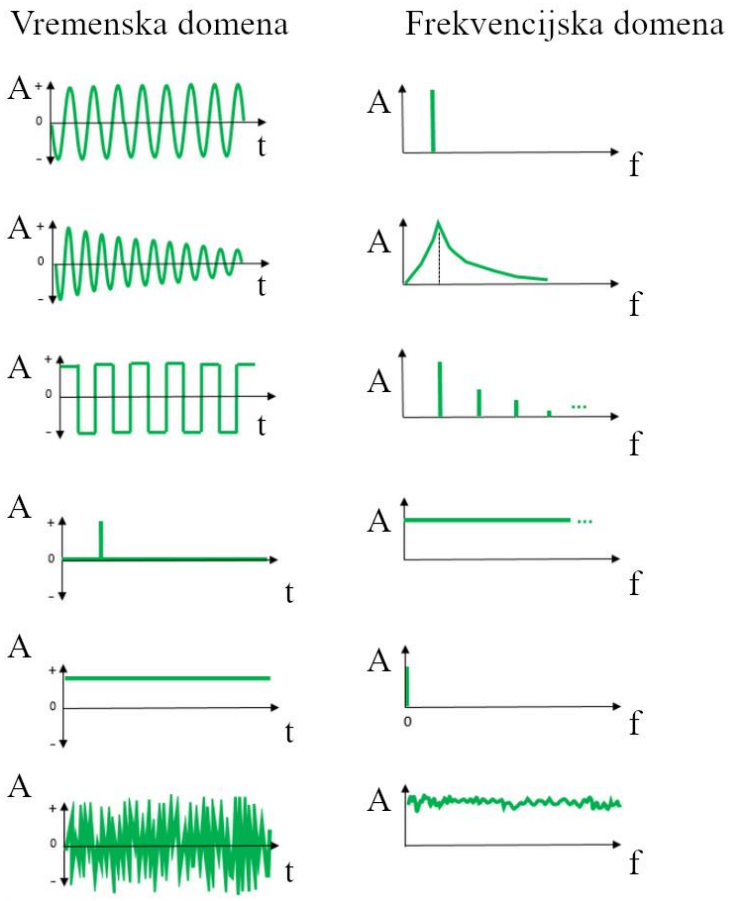
$$\omega = 2\pi f \tag{12}$$

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \tag{13}$$

Gdje je:

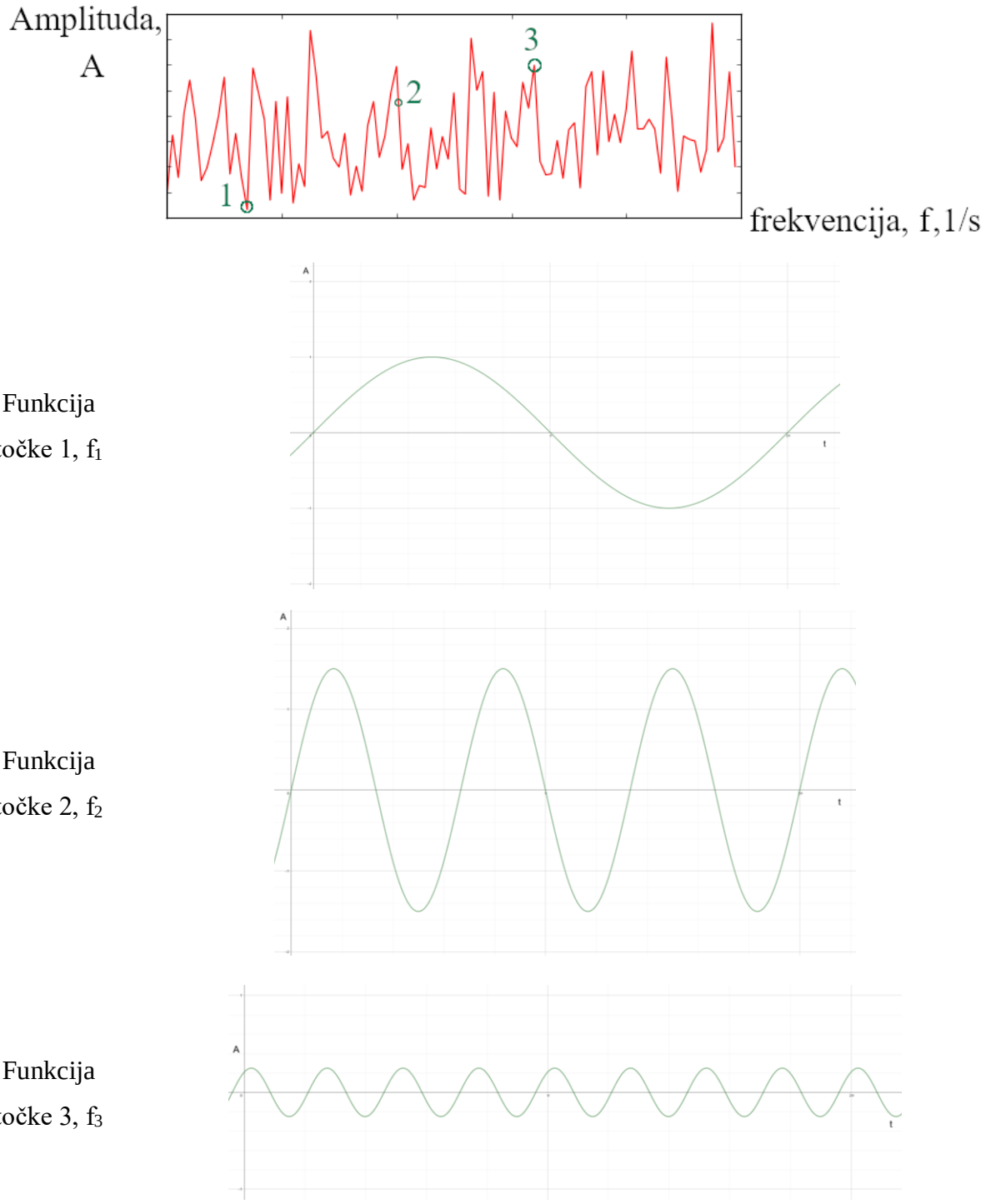
f(t)		funkcija u vremenskoj domeni
F(ω)		funkcija u frekvencijskoj domeni
t	s	vrijeme
ω	rad/s	kružna frekvencija

Potrebno je naglasiti da nakon transformacije izlaz neće nužno biti funkcija, već može biti točka, beskonačan niz točaka (nekontinuirana funkcija), kontinuirana funkcija itd., [Slika 25].



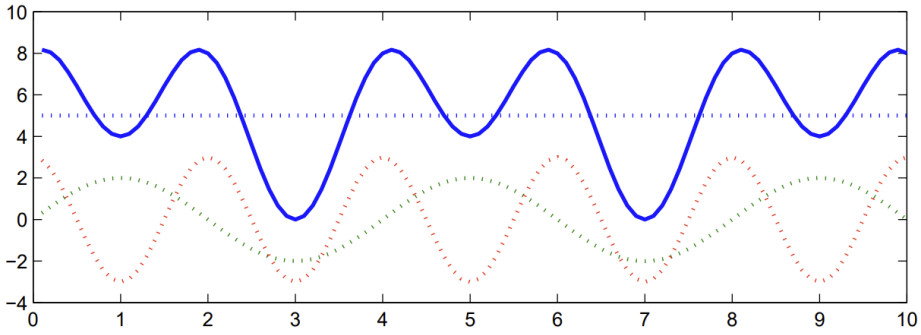
Slika 25. Mogući izgledi Fourierove transformacije [11]

Nakon prelaska iz vremenske domene u frekvencijsku domenu, moguća je obrada signala. Ako se iz grafa u frekvencijskoj domeni odabere bilo koja točka, ona predstavlja sinusoidnu funkciju (ili pomak) u vremenskoj domeni, sa svojom amplitudom, frekvencijom i fazom, [Slika 26]. (nisu točne vrijednosti, već grafovi služe samo kao vizualan primjer).



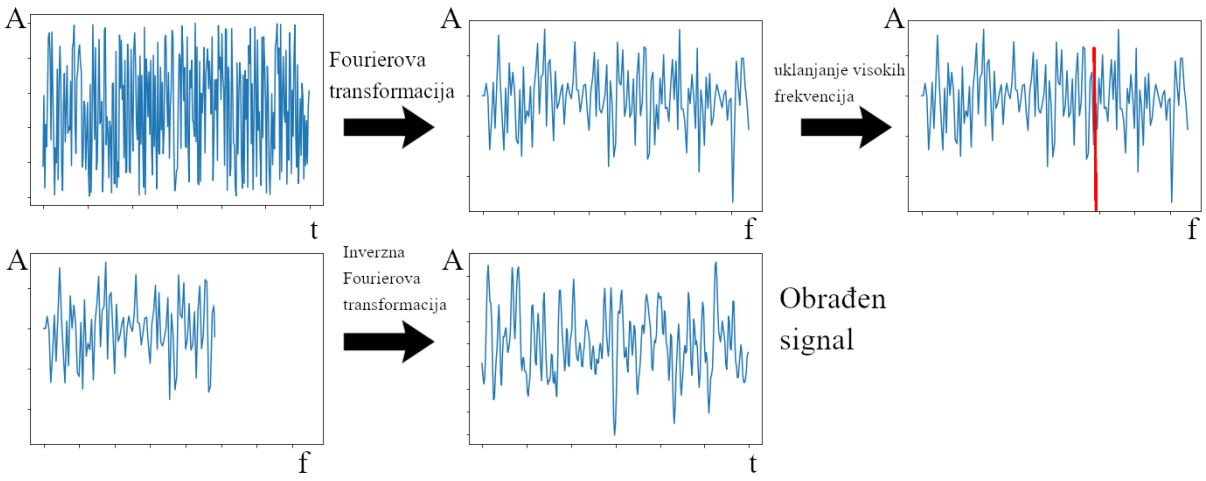
Slika 26. Reprezentacija točaka grafa frekvencijskog područja

Prateći ovu logiku dolazi se do zaključka da je početna funkcija u vremenskoj domeni zapravo suma sinusoidnih podfunkcija, koje se jasno mogu reprezentirati u frekvencijskoj domeni, i upravo je to svrha Fourierove transformacije; ona omogućava razlaganje početne funkcije (signala) na sinusoidne podfunkcije određenih frekvencija, s kojima se može jednostavno manipulirati, [Slika 27]. Svaka podfunkcija sa svojom frekvencijom zove se harmonik.



Slika 27. Razdvajanje funkcije na podfunkcije [12]

Primjena Fourierove transformacije može se jednostavno demonstrirati na primjeru filtracije zvuka. Ako za vremena snimanja zvuka u blizini prođe vozilo sa sirenom, snimka zvuka će biti „zagađena“ sa visokim frekvencijama sirene. Signal od zvuka, prvo se transformira iz vremenskog u frekvencijsko područje, te se u frekvencijskom području jednostavno ukloni sav zvuk frekvencije više od odabrane, potom se inverznom Fourierovom transformacijom iz frekvencijskog područja signal vrati u vremensko područje, te je sada iz snimljenog zvuka uspješno uklonjen zvuk sirene, [Slika 28].



Slika 28. Načelo obrade podataka pomoću Fourierove transformacije i Gaussove filtracije

Inverzna Fourierova transformacija definirana je izrazom (14) ili izrazom (15):

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(\omega)e^{j\omega t}d\omega \quad \textbf{(14)}$$

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(f)e^{-2\pi jft}df \quad \textbf{(15)}$$

3.3.1. Diskretna Fourierova transformacija

Sa temeljnim razumijevanjem što radi Fourierova transformacija moguće je objasniti kako se koristi u realnom svijetu, gdje se ne raspolaže sa funkcijama već sa setom podataka. Svi uređaji za snimanje signala rade na istom principu; uređaj za svaki vremenski interval t zabilježi iznos signala. Što je kraći vremenski interval t između kojeg se uzima vrijednost signala to je set podataka veći i precizniji. Međutim, set podataka nikada ne može u potpunosti opisati funkciju; bilo bi potrebno uzeti beskonačno mnogo podataka u beskonačno kratkom vremenskom intervalu. U takvim, realnim slučajevima koristi se diskretna Fourierova transformacija. Ona ne uzima funkciju u vremenskom području koju transformira u neku novu funkciju u frekvencijskom području, već uzima diskretni set podataka koji su opisani sa trenutkom uzimanja podatka i iznosom, te transformira taj set podataka u novi set koji je definiran sa frekvencijom i amplitudom. Ključna razlika je što se ne transformiraju matematičke funkcije, sastavljene od beskonačno točaka, već se transformira jedan niz brojeva u drugi. Diskretna Fourierova transformacija računa se pomoću izraza (16):

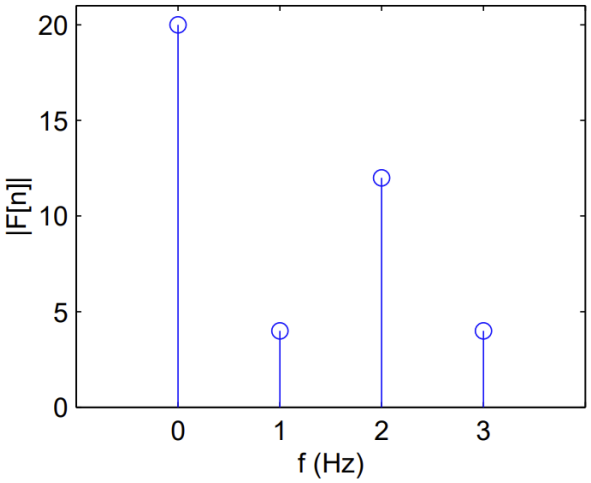
$$F[n] = \sum_{k=0}^{N-1} f[k]e^{-j\frac{2\pi}{N}nk} \quad \textbf{(16)}$$

Gdje je:

f [k] — očitana vrijednost
N — broj očitanih vrijednosti
k — redni broj očitane vrijednosti u nizu, minus jedan
n — redni broj izračunate vrijednosti u nizu, minus jedan

Rješenje se zapisuje u obliku umnoška matrica, tj. vektora, [Slika 29].

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ F[3] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ f[3] \end{pmatrix} = \begin{pmatrix} 20 \\ -j4 \\ 12 \\ j4 \end{pmatrix}$$



Slika 29. Primjer izgleda rješenja diskretne Fourierove transformacije [12]

Inverzna diskretna Fourierova transformacija definirana je izrazom (17) i služi za povratak u vremensko područje nakon obrade signala:

$$f[k] = \frac{1}{N} \sum_{n=0}^{N-1} F[n] e^{+j\frac{2\pi}{N}nk} \qquad \textbf{(17)}$$

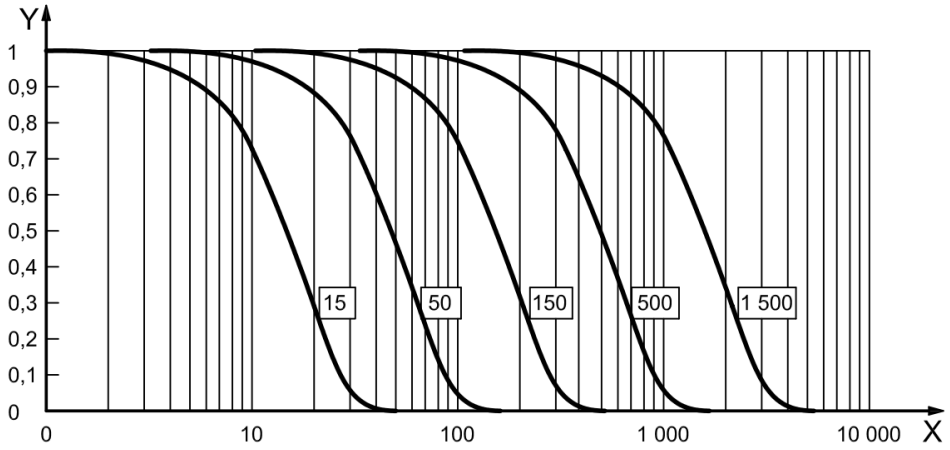
U diplomskom radu koristi se varijanta diskretne Fourierove transformacije, brza Fourierova transformacija. Računalo pomoću nje puno brže računa nego sa diskretnom, ali radi na istom principu, više u sljedećem potpoglavlju.

3.3.2. Brza Fourierova transformacija

Brza Fourierova transformacija je jedan od najkorisnijih i najšire primijenjenih algoritama na cijelome svijetu. Internet, pozivi, obrada zvuka, uklanjanje šuma, slanje i filtracija signala itd., sve se provodi pomoću brze Fourierove transformacije. Razvijena kako bi zamijenila diskretnu Fourierovu transformaciju i omogućila brz i efikasan prijenos podataka, koji danas predstavlja temelj modernog društva. Temeljna razlika između diskretne i brze Fourierove transformacije je u brzini obrade podataka. Kao što je objašnjeno u prošlom potpoglavlju, diskretna Fourierova transformacija koristi matricu kako bi diskretan skup podataka transformirala iz vremenskog u frekvencijsko područje. Računanje sa matricama iznimno je neefikasno, pošto broj potrebnih računa iznosi n^2, gdje je n predstavlja broj podataka. Kada se obrađuju milijuni podataka, broj potrebnih računa vrlo brzo raste, te procesiranje signala postaje vrlo sporo. Brza Fourierova transformacija ne mora provodit n^2 računa, već provodi svega $n \cdot \log n$ računa. Takav broj računa raste vrlo sporo sa povećanjem broja podataka; sa svega 10^5 podataka, što je u realnom svijetu mal iznos, diskretna Fourierova transformacija mora provesti 10^{10} računa, dok brza Fourierova transformacija provodi svega $5 \cdot 10^5$ računa; razlika od 20 000 puta. Broj računa se smanjuje manipulacijom članova matrice podataka; koristeći svojstva preklapanja parnih i neparnih sinusa i kosinusa, za mnogo članova matrice postavi se da su jednaki nuli, te se tako izbjegava računanje svih članova koji su jednaki nuli. Bitno je naglasiti da diskretna i brza Fourierova transformacija daju identične rezultate, tj. da brza Fourierova transformacija nije neka vrsta aproksimacije koja bi potencijalno uvela greške tokom obrade podataka. Zato je primjenjiva za sve postupke obrade signala i svaki programski jezik ima unaprijed isprogramiran algoritam brze Fourierove i inverzne brze Fourierove transformacije.

3.4. Gaussov filter

U kontekstu Fouirerove transformacije potrebno je naglasiti da svaki UPR predstavlja jedan harmonik u frekvencijskoj domeni. Pošto se kružnost promatra do UPR-a 50, potrebno je ukloniti sve više harmonike. Međutim, nije dovoljno samo ih ukloniti, pošto bi onda rezultat bio neprirodan; već se harmonici visokog reda postepeno sve jače prigušuju dok konačno ne postanu jednaki nuli. Način prema kojem se harmonici prigušuju definiran je u normi ISO 12181-2 [13]. Amplituda harmonika množi se sa koeficijentom između nula i jedan; što je odabrani harmonik višeg reda to se koeficijent više približava nuli, dok konačno ne postane nula, [Slika 30].



Slika 30. „Long wave pass filter“ [13]

Iznos koeficijenta ovisno o redu harmonika definiran je izrazom (18):

$$\frac{a_0}{a_1} = e^{-\pi\left(\frac{ak\cdot f}{f_c}\right)^2}, ak = \sqrt{\frac{\ln 2}{\pi}} = 0{,}4697 \qquad \textbf{(18)}$$

Gdje je:

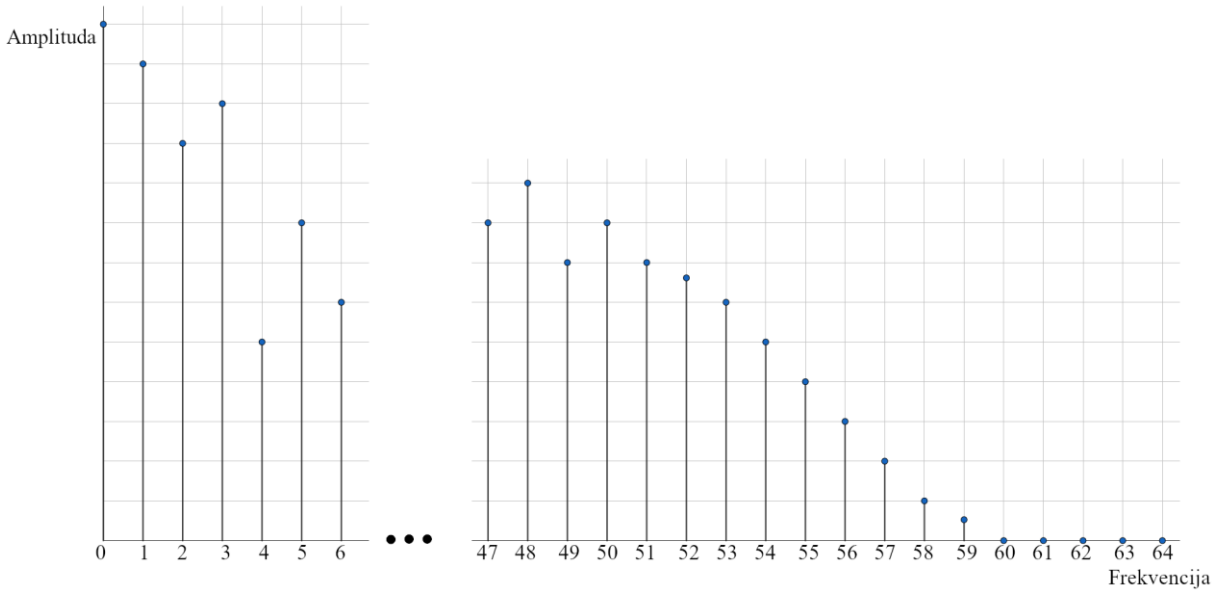
a_0 — amplituda sinusoide prije filtriranja

a_1 — amplituda sinusoide poslije filtriranja

f_c Hz frekvencija od koje se počinje filtrirati

f Hz frekvencija sinusoide

Ovakvo prigušivanje daje efekt normalne razdiobe, te se pomoću nje imitira prirodno smanjenje utjecajnosti, te se izbjegava naglo „rezanje“ informacija tj. podataka. [Slika 31] daje vizualan prikaz utjecaja filtera na amplitude ovisno o frekvenciji sinusoide.



Slika 31. Prigušivanje harmonika

4. PROGRAMIRANJE BIBLIOTEKE ZA ANALIZU KRUŽNOSTI

U okviru diplomskog rada napravljena su dva programa koji zajedno rade u svrhu određivanja odstupanja od kružnosti. Prvi program učitava podatke, te provodi analizu odstupanja od kružnosti. Drugi program simulira mjerenje na uređaju, te generira podatke na temelju odabranih parametara. Programi su napravljeni u programskom jeziku Python, koristeći integrirano okruženje za pisanje programa „Spyder“. U ovom poglavlju biti će objašnjen rad oba programa i zašto se koriste. Programi predstavljaju biblioteke funkcija te se ne mogu koristiti kao gotova aplikacija. Rad programa biti će opisan tako da je paralelno prikazan kod programa i dano objašnjenje što radi svaka linija. Prije objašnjenja koda potrebno je objasniti osnove sintakse pythona:

Stvaranje funkcije u pythonu radi se u formatu „def ImeFunkcije (var1,var2)“ gdje je „ImeFunkcije“ proizvoljno odabrano ime funkcije, a var1,var2 predstavljaju varijable unutar funkcije u koje se pohranjuje vrijednost, te varijable ne mogu se koristiti izvan funkcije (ako su zagrade prazne znači da funkcija ne učitava vrijednosti iz ostatka programa). Naredbe unutar funkcije zahtijevaju uvlaku kako bi se znalo koja naredba spada u funkciju, a koja naredba se nalazi izvan funkcije, [Slika 32].

```
def ImeFukncije():
    Funkcija_naredba1
    Funkcija_naredba2
Naredba3
```

Slika 32. Funkcija u pythonu

Radi jednostavnosti, ako se pojedina funkcija koristi samo jednom, naziv varijable s kojima funkcija računa je isti nazivu varijabli koje se daju funkciji, ali je potrebno naglasiti da unatoč tome što se varijable jednako zovu, one u memoriji računala nisu jednake.

4.1. Program za analizu odstupanja od kružnosti

Program za analizu odstupanja od kružnosti sačinjen je od niza funkcija koje omogućavaju analizu kružnosti. Svaka funkcija biti će objašnjena u zasebnom potpoglavlju radi preglednosti. Sveukupno, program pomoću zasebnog prozora omogućava odabir datoteke koja sadrži rezultate mjerenja, te unosi podatke iz odabrane datoteke u program kako bi se mogla provoditi potrebna obrada. Potom, program provodi aproksimaciju limakona na temelju poglavlja 3.2.1., te daje prikaz originalnih i obrađenih podataka u polarnim koordinatama. Sa varijablama izračunatim tokom aproksimacije limakona, program računa i ispisuje parametre vezane uz odstupanje od kružnosti (*RONv, RONp, RONt* i *RONq*). Potom program provodi Fourierovu transformaciju podataka iz vremenskog u frekvencijsko područje, uklanja harmonik nultog i prvog reda, te obrađeni signal inverznom Fourierovom transformacijom vraća u vremensko područje. Opet se računaju i ispisuju parametri *RONv, RONp, RONt* i *RONq,* te se ponovno daje prikaz podataka u polarnim koordinatama. Uklanjanjem nultog i prvog harmonika pomoću Fourierove transformacije drukčijim pristupom se provodi aproksimacija limakona; *RONv, RONp, RONt* i *RONq* trebaju biti jednaki neovisno o pristupu. Konačno, program u frekvencijskom području prigušuje sve harmonike veće od 50 (moguće je odabrati i drugu vrijednost), ponovno računa i ispisuje *RONv, RONp, RONt* i *RONq*, daje prikaz zaglađenog profila i prikaz usporedbe neobrađenih i filtriranih podataka.

Radi preglednosti, slika koda iz programa je prikazana na lijevoj strani stranice, te objašnjenje koda je prikazano na desnoj strani stranice. Radi jednostavnijeg paralelnog čitanja koda i njegovog objašnjenja, stavljeno je više inačica istog dijela koda, kako bi objašnjenje i relevantan kod uvijek bili u istoj razini. Pošto slike služe samo kao uredniji prikaz koda, nisu numerirane, već je prikaz sveukupnog koda dan u prilogu.

4.1.1. Importovi glavnog programa

Na početku svakog programa prvo je potrebno uvesti unaprijed napravljene biblioteke funkcija i koeficijenta koji će se koristiti unutar programa, pomoću naredbe „import“.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import tkinter as tk
4 from tkinter import filedialog
```

1- uvoz biblioteke „NumPy“, te se u kodu poziva kao „np“. Omogućava korištenje nizova, matrica, koeficijenta π itd.

2- uvoz biblioteke „Pyplot“, te se u kodu poziva kao „plt“. Omogućava crtanje grafova i općenito vizualizacije podataka.

3- uvoz biblioteke „Tkinter“, te se u kodu poziva kao „tk“. Omogućava jednostavno učitavanje podataka u program bez da se moraju ručno unositi.

4- uvoz modula „filedialog“ iz biblioteke „tkinter“. Ovaj modul nije dio glavnih modula koji se učitavaju sa prijašnjom naredbom, stoga se naknadno uvozi.

4.1.2. Odabir datoteke

Sa uvezenim osnovnim funkcijama koje omogućavaju rad programa bez kodiranja već postojećih standardnih funkcija, moguće je efikasno programirati. Potrebna je funkcija za učitavanje podataka koja će iz datoteke sve podatke pohraniti u varijablu:

```
10 def read_data_file():
11
12     root = tk.Tk()
13     root.withdraw()
14
15     root.attributes('-topmost', True)
16
17     file_path = filedialog.askopenfilename(
18         title="Select a file",
19         filetypes=(("Data files", "*.*"), ("All files", "*.*"))
20     )
21
22     root.destroy()
23
```

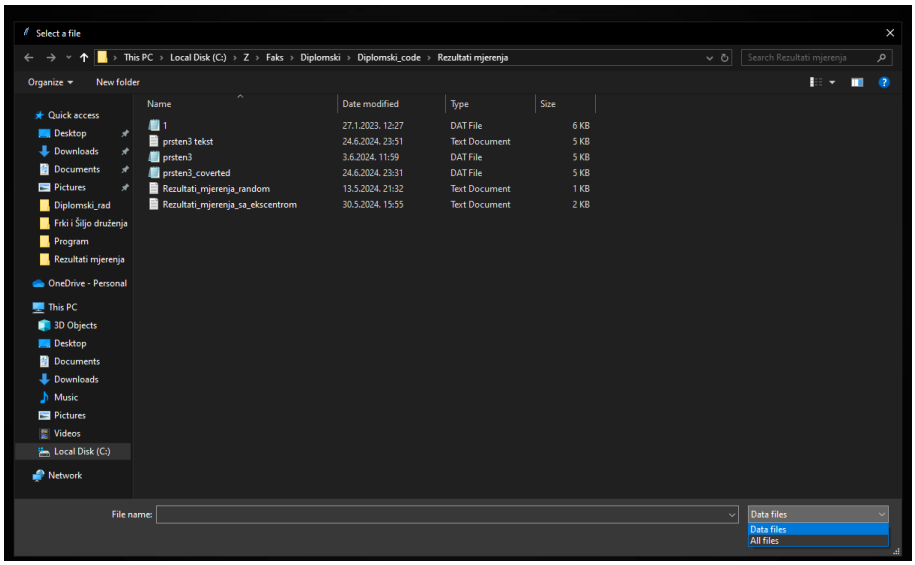
10- stvaranje funkcije koja ne učitava varijable iz ostatka programa, naziva „read_data_file“.

12- poziva funkciju „Tk()“ iz biblioteke „Tkinter“, koja stvara prozor (podatak „Tk“ klase). Takav podatak pohranjuje u varijablu „root“.

```
10 def read_data_file():
11
12     root = tk.Tk()
13     root.withdraw()
14
15     root.attributes('-topmost', True)
16
17     file_path = filedialog.askopenfilename(
18         title="Select a file",
19         filetypes=(("Data files", "*.*"), ("All files", "*.*"))
20     )
21
22     root.destroy()
23
```

```
10 def read_data_file():
11
12     root = tk.Tk()
13     root.withdraw()
14
15     root.attributes('-topmost', True)
16
17     file_path = filedialog.askopenfilename(
18         title="Select a file",
19         filetypes=(("Data files", "*.*"), ("All files", "*.*"))
20     )
21
22     root.destroy()
23
```

13- skriva napravljeni prozor pošto nije nužan za odabir datoteke, ali je nužan kako bi osigurao da prozor u kojem se odabire datoteka bude ispred svih drugih prozora.
15- stavlja prozor za odabir datoteke ispred svih drugih prozora.
17- poziva funkciju „askopenfilename“ iz „filedialog“ modula koja otvara prozor za odabir datoteke (linije 18 i 19 se nalaze unutar zagrada, mogle bi biti u liniji 17, ali radi preglednosti stavljene su u zasebne linije), te stvara varijablu „file_path“ u koju se pohranjuju podatci o lokaciji datoteke na računalu nakon što je datoteka odabrana.
18- definira naslov otvorenog prozora.
19- definira koji tip datoteka će se moći odabrati, prvo datoteke tipa „Data“, te se na padajućem izborniku može promijeniti da se mogu odabrati sve vrste datoteka, [Slika 33].
22- zatvara prozor za odabir datoteke.



Slika 33. Izgled prozora za odabir datoteke

```
24      if file_path:
25          try:
26              data = np.loadtxt(file_path)
27              return data
28          except OSError as e:
29              print(f"Error loading file: {e}")
30              return np.array([])
31      else:
32          return np.array([])
33
34  data = read_data_file()
35
```

24- postavlja uvjet da varijabla „file_path“ ima vrijednost

25- početak komandi ako ima „file_path“ ima vrijednost.

26- u novu varijablu „data“ pohranjuje podatke iz odabrane datoteke koristeći funkciju „loadtxt“ iz „Numpy“ biblioteke koja koristi varijablu „file_path“ za odabir datoteke.

27- kao izlaznu varijablu funkcije koristi varijablu „data“.

```
24      if file_path:
25          try:
26              data = np.loadtxt(file_path)
27              return data
28          except OSError as e:
29              print(f"Error loading file: {e}")
30              return np.array([])
31      else:
32          return np.array([])
33
34  data = read_data_file()
35
```

28- početak komandi u slučaju da ako „file_path“ ima vrijednost, ali je došlo do greške (npr. korisnik nema pristup datoteci).

29- ispisuje poruku „Error loading file: „detalji o grešci“ “.

30- vraća prazan niz kao izlaznu varijablu funkcije.

31- postavlja uvjet da varijabla „file_path“ nema vrijednost.

```
24      if file_path:
25          try:
26              data = np.loadtxt(file_path)
27              return data
28          except OSError as e:
29              print(f"Error loading file: {e}")
30              return np.array([])
31      else:
32          return np.array([])
33
34  data = read_data_file()
35
```

32- vraća prazan niz kao izlaznu varijablu funkcije.

34- poziva se funkcija „read_data_file“, te se njena izlazna vrijednost pohranjuje u varijablu „data“ („data“ linije 26 i linije 34 nisu ista varijabla).

4.1.3. Uklanjanje ekscentra aproksimacijom limakona

Ovaj dio programa provodi matematičku aproksimaciju limakona koja je objašnjena u poglavlju 3.2.1, te koristi prilagođene jednadžbe koje su opisane u tom poglavlju.

```
42 n = len(data)
43 def lslimacon(data):
44
45     theta = np.arange(0, n) * 2 * np.pi / n
46     r = np.mean(data)
47     a = 2 * np.sum(data * np.cos(theta)) / n
48     b = 2 * np.sum(data * np.sin(theta)) / n
49     R_ls = r + a * np.cos(theta) + b * np.sin(theta)
50     Deviation = data - R_ls
51     OOR = np.max(Deviation) - np.min(Deviation)
52     print ("OOR",OOR)
53     return Deviation, r, a, b, OOR, R_ls, theta
54
55 if data.size > 0:
56     Deviation, r, a, b, OOR, R_ls, theta = lslimacon(data)
57     print("r:", r)
58     print("a:", a)
59     print("b:", b)
60     print("OOR:", OOR)
61     print("R_ls:", R_ls)
62 else:
63     print("No data to process.")
64
```

42- stvara se varijabla „n“ te se u nju stavlja broj podataka u datoteci.

43- definira se funkcija imena „lslimacon“ kojoj se pripisuje varijabla „data“.

45- stvara se varijabla „theta“ koja predstavlja kutove pomaka ticala, tj. kut između svake točke na predmetu na kojoj su zabilježeni podatci; funkcija „np.arrange“ stvara niz jednako razmaknutih brojeva na definiranom rasponu, taj niz množi sa 2π („np.pi“ iz biblioteke „NumPy“ uzima vrijednost π) i svaki broj u nizu dijeli sa varijablom „n“, izraženo u radijanima.

46- stvara se varijabla „r“ te se u nju pohranjuje prosječna vrijednost svih podataka iz datoteke pomoću funkcije „np.mean“.

47- stvara se varijabla „a“ koja predstavlja udaljenost središta aproksimirane kružnice od ekscentra po x osi, ekvivalentno jednadžbi (7). „data“ i „theta“ su nizovi brojeva, te ova linija računa ukupnu sumu umnožaka svakog člana niza „data“ sa kosinusom odgovarajućeg člana niza „theta“, pomnoženu sa 2/n. „np.cos“ i „np.sum“ funkcije su kosinusa i sume pozvane iz biblioteke „NumPy“.

```
42 n = len(data)
43 def lslimacon(data):
44
45     theta = np.arange(0, n) * 2 * np.pi / n
46     r = np.mean(data)
47     a = 2 * np.sum(data * np.cos(theta)) / n
48     b = 2 * np.sum(data * np.sin(theta)) / n
49     R_ls = r + a * np.cos(theta) + b * np.sin(theta)
50     Deviation = data - R_ls
51     OOR = np.max(Deviation) - np.min(Deviation)
52     print ("OOR",OOR)
53     return Deviation, r, a, b, OOR, R_ls, theta
54
55 if data.size > 0:
56     Deviation, r, a, b, OOR, R_ls, theta = lslimacon(data)
57     print("r:", r)
58     print("a:", a)
59     print("b:", b)
60     print("OOR:", OOR)
61     print("R_ls:", R_ls)
62 else:
63     print("No data to process.")
64
```

48- stvara se varijabla „b“ koja predstavlja udaljenost središta aproksimirane kružnice od ekscentra po y osi, ekvivalentno jednadžbi (8). Način računanja analogan je računanju varijable „a“, samo se koristi sinus od „theta“.

49- stvara se varijabla „R_ls“ koja je niz brojeva koji predstavljaju polumjer kružnice oblika ispitnog uzorka, ekvivalentno jednadžbi (9).

50- stvara se varijabla „Deviation“ koja predstavlja niz brojava; svaki broj u tom nizu iznosi razliku između člana „data“ i „R_ls“, ekvivalentno jednadžbi (10).

51- stvara se nova varijabla „OOR“ koja računa razliku između najvećeg i najmanjeg člana niza „Deviation“ pomoću „NumPy“ funkcije „max“ i „min“.

52- ispisuje „OOR“ i njezinu vrijednost

53- funkcija daje izlazne varijable „Deviation“, „r“, „a“, „b“, „OOR“, „R_ls“ i „theta“.

55- ako je broj podataka u varijabli „data“ veći od nule provedu se naredbe 56-61

56- stvaraju se varijable „Deviation“, „r“, „a“, „b“, „OOR“, „R_ls“ i „theta“ te se poziva funkcija „lslimacon“ kojoj se daje varijabla „data“.

57-61- ispisuju se vrijednosti varijabli „r“, „a“, „b“, „OOR“, „R_ls“.

62-63- ako „data“ ne sadrži podatke, ispisuje se „No data to process“.

4.1.4. Crtanje profila

Ovaj dio programa definira funkciju za crtanje profila sličnom kružnici.

```
70
71 def plot_circle(R, theta, title):
72     plt.figure()
73     ax = plt.subplot(111, projection='polar')
74     ax.scatter(theta, R, s=0)
75     ax.plot(theta, R, color='blue', alpha=0.5)
76     ax.set_rmin(-8)
77     ax.set_title(title)
78     plt.show()
79
```

71- definira se funkcija imena „plot_circle“ kojoj se pripisuju varijable „R“, „theta“, „title“.

72- stvara pozadinu na kojoj se crta graf pomoću funkcije „figure“ iz biblioteke „matplotlib“ sa modulom „pyplot“.

73- crta graf u polarnim koordinatama, koristeći jednu sliku, jedan stupac i jedan redak, te stvara varijablu „ax“ u koju se graf pohranjuje.

74- vrši „scatter“ funkciju na varijablu „ax“; stavlja točke na graf ovisno o radijusu „R“ i kutu „theta“, te su točke sakrivene radi preglednosti.

75- povezuje sve točke sa plavom linijom, transparentnosti 0,5.

76- definira da je raspon vrijednosti grafa minimalno do minus osam.

77- stavlja da je naslov grafa jednak varijabli „title“.

78- prikazuje graf koristeći funkciju „show“.

Sa napravljenom funkcijom za crtanje kružnice, crtaju se dvije kružnice, prva koristi neobrađene podatke dobivene sa stroja, dok druga koristi varijablu „Deviation“.

```
85
86 plot_circle(data, theta, 'Sirovi podaci')
87 plot_circle(Deviation, theta,'Korigiran limakon')
88
```

86- poziva funkciju „plot_circle“ te joj se daju varijable „data“, „theta“ i naslov „Sirovi podatci“.

87- anaolgno liniji 86, samo sa „Deviation“, i naslovom „Korigirani limakon“.

4.1.5. Računanje RONp, RONv, RONq i RONt

Ovaj dio programa računa i ispisuje parametre *RONp, RONv, RONq* i *RONt*.

```
95 def find_RONp_and_RONv(data):
96     RONp = np.max(data)
97     RONv = np.min(data)
98     RONq= np.sqrt((np.sum(np.square(data)) / len(data)))
99     return RONv, RONp, RONq
100
101 def calculate_difference(RONp, RONv):
102     return RONp - RONv
103
104 if data.size == 0:
105     print("Data is empty or file could not be loaded. Exiting.")
106 else:
107     RONv, RONp, RONq = find_RONp_and_RONv(Deviation)
108     RONt = calculate_difference(RONp, RONv)
109     print(f'RONv= {RONv}')
110     print(f'RONp= {RONp}')
111     print(f'RONt= {RONt}')
112     print(f'RONq= {RONq}')
```

95- definira se funkcija imena „find_RONp_and_RONv“ kojoj se pripisuje varijabla „data“.

96- stvara se varijabla „RONp“ i u nju se stavlja najveći član iz varijable „data“.

97- stvara se varijabla „RONv“ i u nju se stavlja najmanji član iz varijable „data“.

98- stvara se varijabla „RONq“ i u nju se stavlja korijen (pomoću funkcije „sqrt“ iz biblioteke „NumPy“) sume članova niza; svaki član niza jednak je kvadratu člana varijable „data“ (pomoću funkcije „square“ iz biblioteke „NumPy“) podijeljen sa brojem članova varijable „data“, analogno jednadžbi (4).


```
95 def find_RONp_and_RONv(data):
96     RONp = np.max(data)
97     RONv = np.min(data)
98     RONq= np.sqrt((np.sum(np.square(data)) / len(data)))
99     return RONv, RONp, RONq
100
101 def calculate_difference(RONp, RONv):
102     return RONp - RONv
103
104 if data.size == 0:
105     print("Data is empty or file could not be loaded. Exiting.")
106 else:
107     RONv, RONp, RONq = find_RONp_and_RONv(Deviation)
108     RONt = calculate_difference(RONp, RONv)
109     print(f'RONv= {RONv}')
110     print(f'RONp= {RONp}')
111     print(f'RONt= {RONt}')
112     print(f'RONq= {RONq}')
```


99- funkcija vraća izlazne varijable „RONv“, „RONp“ i „RONq“.

101-definira se funkcija imena „calculate_difference“ kojoj se pripisuju varijable „RONp“ i „RONv“.

102- funkcija daje izlaznu varijablu jednaku razlici varijabli „RONp“ i „RONv“.


```
95 def find_RONp_and_RONv(data):
96     RONp = np.max(data)
97     RONv = np.min(data)
98     RONq= np.sqrt((np.sum(np.square(data)) / len(data)))
99     return RONv, RONp, RONq
100
101 def calculate_difference(RONp, RONv):
102     return RONp - RONv
103
104 if data.size == 0:
105     print("Data is empty or file could not be loaded. Exiting.")
106 else:
107     RONv, RONp, RONq = find_RONp_and_RONv(Deviation)
108     RONt = calculate_difference(RONp, RONv)
109     print(f'RONv= {RONv}')
110     print(f'RONp= {RONp}')
111     print(f'RONt= {RONt}')
112     print(f'RONq= {RONq}')
```


104- provjerava jeli broj članova u varijabli „data“ jednak nuli.

105- ako je, ispisuje „Data is empty or file could not be loaded. Exiting.“

106- ako broj članova u varijabli „data“ nije jednak nuli;

```
95 def find_RONp_and_RONv(data):
96     RONp = np.max(data)
97     RONv = np.min(data)
98     RONq= np.sqrt((np.sum(np.square(data)) / len(data)))
99     return RONv, RONp, RONq
100
101 def calculate_difference(RONp, RONv):
102     return RONp - RONv
103
104 if data.size == 0:
105     print("Data is empty or file could not be loaded. Exiting.")
106 else:
107     RONv, RONp, RONq = find_RONp_and_RONv(Deviation)
108     RONt = calculate_difference(RONp, RONv)
109     print(f'RONv= {RONv}')
110     print(f'RONp= {RONp}')
111     print(f'RONt= {RONt}')
112     print(f'RONq= {RONq}')
```

107- stvaraju se varijable „RONv“, „RONp“ i „RONq“, te se poziva funkcija „find_RONp_and_RONv“ kojoj joj se daje varijabla „Deviation“.
108- stvara se varijabla „RONt“, te se poziva funkcija „calculate_difference“ kojoj se daju varijable „RONp“ i „RONv“.
109-112- ispisuju se parametri „RONv“, „RONp“, „RONt“ i „RONq“ i njihove pripadajuće vrijednosti.

4.1.6. Fourierova transformacija

Ovaj dio programa provodi Fourierovu transformaciju, uklanja nulti i prvi harmonik, računa *RONp*, *RONv*, *RONq* i *RONt* i crta kružnicu bez nultog i prvog harmonika. Drukčijim pristupom provodi uklanjanje limakona, i svi izračunati parametri i nacrtana kružnica trebaju biti jednaki onima dobivenim u poglavlju 4.1.5.

```
120 har_orig = np.fft.rfft(data)
121 har = np.fft.rfft(Deviation)
122 har_alter = har_orig.copy()
123 har_alter[0] = 0
124 har_alter[1] = 0
125 Deviation_1 = np.fft.irfft(har_alter)
126
127 RONv, RONp, RONq = find_RONp_and_RONv(Deviation_1)
128 RONt = calculate_difference(RONp, RONv)
129
130 print(f'RONv_FFT= {RONv}')
131 print(f'RONp_FFT= {RONp}')
132 print(f'RONt_FFT= {RONt}')
133 print(f'RONq_FFT= {RONq}')
134 plot_circle (Deviation_1, theta,'Bez 0 i 1 harmonika')
```

```
120 har_orig = np.fft.rfft(data)
121 har = np.fft.rfft(Deviation)
122 har_alter = har_orig.copy()
123 har_alter[0] = 0
124 har_alter[1] = 0
125 Deviation_1 = np.fft.irfft(har_alter)
126
127 RONv, RONp, RONq = find_RONp_and_RONv(Deviation_1)
128 RONt = calculate_difference(RONp, RONv)
129
130 print(f'RONv_FFT= {RONv}')
131 print(f'RONp_FFT= {RONp}')
132 print(f'RONt_FFT= {RONt}')
133 print(f'RONq_FFT= {RONq}')
134 plot_circle (Deviation_1, theta,'Bez 0 i 1 harmonika')
```

120- provodi se Fourierova transformacija pomoću funkcije „fft.rfft“, („rfft“ se koristi zato što je skup podataka koji se obrađuje realan) sa varijablom „data“, te se podatci dobiveni Fourierovom transformacijom pohranjuju u novu varijablu „har_orig“.
121- analogno liniji 121, samo se koristi varijabla „Deviation“ i stvara se varijabla „har“
122- stvara se kopija varijable „har_orig“.
123, 124- nulti i prvi član varijable „har_alter“ stavljaju se da su jednaki nuli, objašnjeno u poglavlju 3.2.2.

```
120 har_orig = np.fft.rfft(data)
121 har = np.fft.rfft(Deviation)
122 har_alter = har_orig.copy()
123 har_alter[0] = 0
124 har_alter[1] = 0
125 Deviation_1 = np.fft.irfft(har_alter)
126
127 RONv, RONp, RONq = find_RONp_and_RONv(Deviation_1)
128 RONt = calculate_difference(RONp, RONv)
129
130 print(f'RONv_FFT= {RONv}')
131 print(f'RONp_FFT= {RONp}')
132 print(f'RONt_FFT= {RONt}')
133 print(f'RONq_FFT= {RONq}')
134 plot_circle (Deviation_1, theta,'Bez 0 i 1 harmonika')
```

125- provodi se inverzna realna Fourierova transformacija na varijabli „har_alter“ te se pohranjuje u novu varijablu „Deviation_1“.

127- poziva se funkcija „find_RONp_and RONv“ te joj se daje varijabla „Deviation_1“, te se izlazne varijable funkcije pohranjuju u „RONv“, „RONp“ i „RONq“ (ove varijable su već bile ispisane te se mogu usporediti sa novim vrijednostima koje su poprimile nakon Fourierove transformacije).

128- poziva se funkcija „calculate_difference“ te joj se daju varijable „RONp“ i „RONv“, te se izlaz funkcije pohranjuje u varijablu „RONt“.

130-133- ispisuje „RONv_FFT“, „RONp_FFT“, „RONt_FFT“ i „RONq_FFT“ i pripadajuće vrijednosti novo izračunatih parametara. (varijable su izračunate drukčijim pristupom, ali nove vrijednosti trebaju biti jednake prijašnje izračunatima).

134- poziva se funkcija „plot_circle“ i pripisuju joj se varijable „Deviation_1“, „theta“ i „Bez 0 i 1 harmonika“.

4.1.7. Gaussova filtracija

Ovaj dio programa filtrira UPR-ove veće od 50, crta profil nakon filtriranja i prikazuje usporedbu podataka prije i poslije filtriranja.

```
142 fc=50
143
144 alpha=np.sqrt(np.log(2)/np.pi)
145
146 mag=np.abs(har_alter)
147
148 faza=np.angle((har_alter))
149
150 a2 = [0] * len(mag)
151
152 for i in range(len(mag)):
153     a2[i] = mag[i] * (np.exp(-np.pi * (alpha * i / fc)**2))
154
155 a2[0] = 0
156 a2[1] = 0
157
158 complex_numbers = a2 * np.exp(1j * faza)
159
160 smoothed_arr = np.fft.irfft(complex_numbers )
```

142- stvara varijablu „fc“ koja predstavlja od kojeg UPR-a počinje prigušivanje.

144- stvara varijablu „alpha“ koja iznosi korijen vrijednosti dijeljenja prirodnog logaritma od 2 (pomoću funkcije „log“ iz biblioteke „NumPy“) sa π, približno 0,4697.

146- stvara varijablu „mag“ u koju se pohranjuje niz brojeva koji predstavljaju apsolutnu vrijednost članova varijable „har_alter“ (pomoću funkcije „abs“ iz biblioteke „NumPy“). „har_alter“ je niz kompleksnih brojeva.

148- stvara varijablu „faza“ u koju se pohranjuje faza (pomak) svakog člana varijable „har_alter“ (pomoću funkcije „angle“ iz biblioteke „NumPy“).

150- stvara varijablu „a2“ koja je napravljena kao niz nula dugačak kao broj članova varijable „mag“.

152- početak „for“ petlje sa promjenjivom varijablom „i“. Svakim prolaskom petlje „i“ se povećava za jedan. Petlja kreće sa „i“ jednako jedan, te završava kada „i“ bude jednak broju članova varijable „mag“.

```
142 fc=50
143
144 alpha=np.sqrt(np.log(2)/np.pi)
145
146 mag=np.abs(har_alter)
147
148 faza=np.angle((har_alter))
149
150 a2 = [0] * len(mag)
151
152 for i in range(len(mag)):
153     a2[i] = mag[i] * (np.exp(-np.pi * (alpha * i / fc)**2))
154
155 a2[0] = 0
156 a2[1] = 0
157
158 complex_numbers = a2 * np.exp(1j * faza)
159
160 smoothed_arr = np.fft.irfft(complex_numbers )
```

153- množi vrijednost pojedinačnog člana varijable „mag“ sa koeficijentom e (uvezen iz biblioteke „NumPy“) koji ima eksponent jednak umnošku -π i kvadratu vrijednosti umnoška varijable „alpha“ i „i“, podijeljeno sa varijablom „fc“, ekvivalentno jednadžbi (18) kada bi se cijela jednadžba pomnožila sa a_0.

155- nulti član varijable „a2“ stavlja se da je jednak nuli.


```
142 fc=50
143
144 alpha=np.sqrt(np.log(2)/np.pi)
145
146 mag=np.abs(har_alter)
147
148 faza=np.angle((har_alter))
149
150 a2 = [0] * len(mag)
151
152 for i in range(len(mag)):
153     a2[i] = mag[i] * (np.exp(-np.pi * (alpha * i / fc)**2))
154
155 a2[0] = 0
156 a2[1] = 0
157
158 complex_numbers = a2 * np.exp(1j * faza)
159
160 smoothed_arr = np.fft.irfft(complex_numbers )
```


155- prvi član varijable „a2“ stavlja se da je jednak nuli (uklanjanje nultog i prvog harmonika).

158- stvara se varijabla „complex_numbers“ koja je jednaka umnošku varijable „a2“ i koeficijenta e sa potencijom koja iznosi umnožak varijable „faza“ i imaginarnog broja j (kombinira amplitudu i fazu u kompleksan broj).


```
142 fc=50
143
144 alpha=np.sqrt(np.log(2)/np.pi)
145
146 mag=np.abs(har_alter)
147
148 faza=np.angle((har_alter))
149
150 a2 = [0] * len(mag)
151
152 for i in range(len(mag)):
153     a2[i] = mag[i] * (np.exp(-np.pi * (alpha * i / fc)**2))
154
155 a2[0] = 0
156 a2[1] = 0
157
158 complex_numbers = a2 * np.exp(1j * faza)
159
160 smoothed_arr = np.fft.irfft(complex_numbers )
```


160- provodi inverznu realnu Fourierovu transformaciju i niz podataka stavlja u varijablu „smoothed_arr“.

Sa dobivenim nizom koji sadrži filtrirane podatke crta se zaglađeni profil i graf koji uspoređuje podatke prije i poslije zaglađivanja.

```
162 plot_circle (smoothed_arr, theta,'filtrirano')
163
164 def plot_graph(Deviation_1, smoothed_arr):
165
166   plt.figure()
167   plt.plot(Deviation_1, label='Deviations_1 prije zaglađenja', color='tab:blue')
168   plt.plot(smoothed_arr, label='Nakon zaglađenja', color='orange')
169   plt.legend()
170   plt.title("Zaglađenje Gaussovim filtrom")
171   plt.xlabel("Stupnjevi")
172   plt.ylabel("R")
173   plt.show()
174
175 plot_graph(Deviation_1, smoothed_arr)
176
177 RONv, RONp, RONq = find_RONp_and_RONv(smoothed_arr)
178 RONt = calculate_difference(RONp, RONv)
179
180 print(f'RONv_Filtrirano= {RONv}')
181 print(f'RONp_Filtrirano= {RONp}')
182 print(f'RONt_Filtrirano= {RONt}')
183 print(f'RONq_Filtrirano= {RONq}')
```

162- poziva funkciju „plot_circle“ i daje joj varijable „smoothed_arr“, „theta“ i naslov „filtrirano“.

164- definira funkciju imena „plot_graph“ i pripisuje joj varijable „Deviation_1“ i „smoothed_arr“.

166- stvara pozadinu na kojoj se crta graf

167- stvara krivulju za varijablu „Deviation_1“ , plave boje, i stvara naziv krivulje „Deviations_1 prije zaglađenja“.

168- stvara krivulju za varijablu „smoothed_arr“, narančaste boje, i stvara naziv krivulje „Nakon zaglađenja“.

169- stavlja legendu na graf sa nazivima krivulja pomoću funkcije „legend“ iz biblioteke „matplotlib“ sa modulom „pyplot“.

170- stavlja naslov na graf „Zaglađivanje Gaussovim filtrom“

171- označava x os grafa pomoću funkcije „xlabel“ iz biblioteke „matplotlib“ sa modulom „pyplot“ sa nazivom „Stupnjevi“.

172- označava y os grafa pomoću funkcije „ylabel“ iz biblioteke „matplotlib“ sa modulom „pyplot“ sa nazivom „R“.

173- prikazuje graf.

175- poziva funkciju „plot_graph“ i daje joj varijable „Deviation_1“ i „smoothed_arr“.

```
plot_circle (smoothed_arr, theta,'filtrirano')

def plot_graph(Deviation_1, smoothed_arr):

  plt.figure()
  plt.plot(Deviation_1, label='Deviations_1 prije zaglađenja', color='tab:blue')
  plt.plot(smoothed_arr, label='Nakon zaglađenja', color='orange')
  plt.legend()
  plt.title("Zaglađenje Gaussovim filtrom")
  plt.xlabel("Stupnjevi")
  plt.ylabel("R")
  plt.show()

plot_graph(Deviation_1, smoothed_arr)

RONv, RONp, RONq = find_RONp_and_RONv(smoothed_arr)
RONt = calculate_difference(RONp, RONv)

print(f'RONv_Filtrirano= {RONv}')
print(f'RONp_Filtrirano= {RONp}')
print(f'RONt_Filtrirano= {RONt}')
print(f'RONq_Filtrirano= {RONq}')
```


```
plot_circle (smoothed_arr, theta,'filtrirano')

def plot_graph(Deviation_1, smoothed_arr):

  plt.figure()
  plt.plot(Deviation_1, label='Deviations_1 prije zaglađenja', color='tab:blue')
  plt.plot(smoothed_arr, label='Nakon zaglađenja', color='orange')
  plt.legend()
  plt.title("Zaglađenje Gaussovim filtrom")
  plt.xlabel("Stupnjevi")
  plt.ylabel("R")
  plt.show()

plot_graph(Deviation_1, smoothed_arr)

RONv, RONp, RONq = find_RONp_and_RONv(smoothed_arr)
RONt = calculate_difference(RONp, RONv)

print(f'RONv_Filtrirano= {RONv}')
print(f'RONp_Filtrirano= {RONp}')
print(f'RONt_Filtrirano= {RONt}')
print(f'RONq_Filtrirano= {RONq}')
```


177- poziva se funkcija „find_RONp_and RONv“ te joj se daje varijabla „smoothed_arr“, te se izlazne varijable funkcije pohranjuju u „RONv“, „RONp“ i „RONq“

178- poziva se funkcija „calculate_difference“ te joj se daju varijable „RONp“ i „RONv“, te se izlaz funkcije pohranjuje u varijablu „RONt“.

180-183- ispisuje „RONv_Filtrirano“, „RONp_ Filtrirano“, „RONt_ Filtrirano“ i „RONq_ Filtrirano“ i pripadajuće vrijednosti novo izračunatih parametara.

4.2. Generiranje signala

Uz program koji obrađuje dobivene podatke, napravljen je i popratni program, koji generira podatke na temelju zadanih parametara. Popratni program služi kako bi se provjerila točnost rada glavnog programa prije nego što se analizira odstupanje od kružnosti ispitnog uzorka. Program stvara proizvoljan broj podataka, te generira tri harmonika proizvoljno odabranog reda i amplitude, te se može odabrati amplituda koja se dodaje ukupnom signalu i jačina šuma koji će uvesti nasumičnost u podatke. Potom se generiraju tri odvojena signala, te se generira utjecaj šuma i zbrajaju se svi signali, šum i početna amplituda. Generirani signal se prikazuje u polarnim koordinatama i u grafu sa osima vrijeme, amplituda, te se za sve generirane signale i šum generira datoteka koja se može koristiti kao sirovi podatci za glavni program. Radi preglednosti, opis programa za generiranje podataka podijeljen je u potpoglavlja.

4.2.1. Importovi programa za generiranje signala

Na početku svakog programa prvo je potrebno uvesti unaprijed napravljene biblioteke funkcija i koeficijenta koji će se koristiti unutar programa, pomoću naredbe „import“.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from tkinter import Tk, filedialog
4 import os
```

1- uvoz biblioteke „NumPy“, te se u kodu poziva kao „np“. Omogućava korištenje nizova, matrica, koeficijenta π itd.

2- uvoz biblioteke „Pyplot“, te se u kodu poziva kao „plt“.

3- iz biblioteke „tkinter“ uvozi se moduli „Tk“ i „filedialog“.

4- uvozi „os“ modul koji omogućava interakciju između programa i operativnog sustava (spremanje datoteka).

4.2.2. Generiranje signala

Potom se stvaraju harmonici, razina šuma i početna amplituda, te se generiraju signali, utjecaj šuma i početne amplitude, koji se potom zbrajaju u sveukupan signal.

```
6  n = 1500
7
8  theta = np.arange(0, n) * 2 * np.pi / n
9
10 harmonic_1 = 1
11 amplitude_1 = 2
12
13 harmonic_2 = 5
14 amplitude_2 = 2
15
16 harmonic_3 = 80
17 amplitude_3 = 1
18
19 amplitude_noise = 0.8
20
21 bias = 1
22
23
24 signal_1 = amplitude_1 * np.sin(np.linspace(0, 360 * harmonic_1, n) * np.pi / 180)
25 signal_2 = amplitude_2 * np.sin(np.linspace(0, 360 * harmonic_2, n) * np.pi / 180)
26 signal_3 = amplitude_3 * np.sin(np.linspace(0, 360 * harmonic_3, n) * np.pi / 180)
27 noise = np.random.uniform(-amplitude_noise, amplitude_noise, n)
28
29 signal_t = signal_1 + signal_2 + signal_3 + noise + bias
```

6- odabir koliko se podataka generira

8- računanje pripadajućeg kuta za podatke.

10- stvaranje varijable „harmonic_1“ kojoj se proizvoljno odabire vrijednost.

11- stvaranje varijable „amplitude_1“ kojoj se proizvoljno odabire vrijednost.

13,14 i 16,17- ekvivalentno naredbama 10,11; više harmonika omogućava bolje upravljanje oblika podataka.

19- stvara se varijabla „amplitude_noise“ koja određuje jačinu šuma.

```
n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 1
amplitude_1 = 2

harmonic_2 = 5
amplitude_2 = 2

harmonic_3 = 80
amplitude_3 = 1

amplitude_noise = 0.8

bias = 1


signal_1 = amplitude_1 * np.sin(np.linspace(0, 360 * harmonic_1, n) * np.pi / 180)
signal_2 = amplitude_2 * np.sin(np.linspace(0, 360 * harmonic_2, n) * np.pi / 180)
signal_3 = amplitude_3 * np.sin(np.linspace(0, 360 * harmonic_3, n) * np.pi / 180)
noise = np.random.uniform(-amplitude_noise, amplitude_noise, n)

signal_t = signal_1 + signal_2 + signal_3 + noise + bias
```

21- stvara se varijabla „bias“ koja određuje jačinu početne amplitude.

24- generira se varijabla „signal_1“ koja predstavlja komponentu signala koje ticalo generira tokom mjerenja ispitnog uzorka. U slučaju da je vrijednost svih ostalih varijabla (osim „n“) jednaka nuli, onda bi varijabla „signal_1“ predstavljala ukupni signal koje ticalo generira.

Sama varijabla „signal_1“ je niz brojeva; računa se kao umnožak varijable „amplitude_1“ i sinusne funkcije. Sinusna funkcija računa vrijednost umnoška π/180 sa nizom brojeva, veličine „n“ od 0 do 360 · „harmonic_1“, sa jednakim razmakom između svakog broja (pomoću funkcije „linspace“ iz biblioteke „NumPy“).


```
n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 1
amplitude_1 = 2

harmonic_2 = 5
amplitude_2 = 2

harmonic_3 = 80
amplitude_3 = 1

amplitude_noise = 0.8

bias = 1


signal_1 = amplitude_1 * np.sin(np.linspace(0, 360 * harmonic_1, n) * np.pi / 180)
signal_2 = amplitude_2 * np.sin(np.linspace(0, 360 * harmonic_2, n) * np.pi / 180)
signal_3 = amplitude_3 * np.sin(np.linspace(0, 360 * harmonic_3, n) * np.pi / 180)
noise = np.random.uniform(-amplitude_noise, amplitude_noise, n)

signal_t = signal_1 + signal_2 + signal_3 + noise + bias
```


25,26- analogno liniji 24.

27- stvara se varijabla „noise“ koja predstavlja šum; ona je oblika niza brojeva generiranog pomoću funkcije „uniform“ koja koristi generator nasumičnih brojeva iz „Numpy“ biblioteke. Funkcije je definirana donjom i gornjom granicom, koje su jednake pozitivnoj i negativnoj vrijednosti varijable „amplitude_noise“, te se stvara „n“ podataka.


```
n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 1
amplitude_1 = 2

harmonic_2 = 5
amplitude_2 = 2

harmonic_3 = 80
amplitude_3 = 1

amplitude_noise = 0.8

bias = 1


signal_1 = amplitude_1 * np.sin(np.linspace(0, 360 * harmonic_1, n) * np.pi / 180)
signal_2 = amplitude_2 * np.sin(np.linspace(0, 360 * harmonic_2, n) * np.pi / 180)
signal_3 = amplitude_3 * np.sin(np.linspace(0, 360 * harmonic_3, n) * np.pi / 180)
noise = np.random.uniform(-amplitude_noise, amplitude_noise, n)

signal_t = signal_1 + signal_2 + signal_3 + noise + bias
```


29- zbroja varijable „signal_1“, „signal_2“, „signal_3“, „noise“ i „bias“.

4.2.3. Prikaz signala i profila podataka

Generirani podatci se prikazuju u polarnim koordinatama i na grafu sa osima broj podatka, amplituda.

```
31 def plot_graph(signal_t):
32
33   plt.figure()
34   plt.plot(signal_t)
35   plt.title('Combined Signal')
36   plt.xlabel('Sample')
37   plt.ylabel('Amplitude')
38   plt.show()
39
40 def plot_circle(R, theta):
41     plt.figure()
42     ax = plt.subplot(111, projection='polar')
43     ax.scatter(theta, R, s=0)
44     ax.plot(theta, R, color='blue', alpha=0.5)
45     ax.set_rmin(-20)
46     ax.set_title('Polar Coordinates Plot')
47     plt.show()
48
49 plot_circle(signal_t, theta)
50 plot_graph(signal_t)
```

31- analogno funkciji „plot_graph“ iz poglavlja 4.1.7, sa razlikom što se crta samo jedna krivulja i nema legende.

40- analogno funkciji „plot_circle“ iz poglavlja 4.1.4, sa razlikom što se ne stavlja naslov.

49- poziva funkciju „plot_circle“, te joj se daju varijable „signal_t“ i „theta“.

50- poziva funkciju „plot_graph“ te joj daje varijablu „signal_t“.

4.2.4. Generiranje datoteka

Sa svim potrebnim podatcima koji predstavljaju signal mjerenja kakav bi se dobio da se provelo ispitivanje kružnosti na realnom ispitnom uzorku, potrebno je napraviti datoteke koje se mogu učitati u glavni program.

```
52 def select_save_directory():
53     root = Tk()
54     root.withdraw()
55     root.attributes('-topmost', True)
56     save_dir = filedialog.askdirectory(title="Select Save Directory")
57     root.destroy()
58     return save_dir
59
60 save_dir = select_save_directory()
61 if not save_dir:
62     print("Save directory selection cancelled. Exiting.")
63 else:
64     np.savetxt(os.path.join(save_dir, "signal_t.txt"), signal_t, delimiter=',')
65     np.savetxt(os.path.join(save_dir, "signal_1.txt"), signal_1, delimiter=',')
66     np.savetxt(os.path.join(save_dir, "signal_2.txt"), signal_2, delimiter=',')
67     np.savetxt(os.path.join(save_dir, "signal_3.txt"), signal_3, delimiter=',')
68     np.savetxt(os.path.join(save_dir, "noise.txt"), noise, delimiter=',')
69     print(f"Files saved to {save_dir}")
70
```


```
52 def select_save_directory():
53     root = Tk()
54     root.withdraw()
55     root.attributes('-topmost', True)
56     save_dir = filedialog.askdirectory(title="Select Save Directory")
57     root.destroy()
58     return save_dir
59
60 save_dir = select_save_directory()
61 if not save_dir:
62     print("Save directory selection cancelled. Exiting.")
63 else:
64     np.savetxt(os.path.join(save_dir, "signal_t.txt"), signal_t, delimiter=',')
65     np.savetxt(os.path.join(save_dir, "signal_1.txt"), signal_1, delimiter=',')
66     np.savetxt(os.path.join(save_dir, "signal_2.txt"), signal_2, delimiter=',')
67     np.savetxt(os.path.join(save_dir, "signal_3.txt"), signal_3, delimiter=',')
68     np.savetxt(os.path.join(save_dir, "noise.txt"), noise, delimiter=',')
69     print(f"Files saved to {save_dir}")
70
```


52-58- definira se funkcija „select_save_directory“ kojoj se ne pripisuje varijabla, ona je analogna funkciji iz poglavlja 4.1.2, sa razlikom što se koristi funkcija „askdirectory“ umjesto funkcije „askopenfilename“ koja upisuje lokaciju mjesta gdje se datoteka želi spremiti u varijablu „save_dir“ koja se daje kao izlazna varijabla funkcije.

60- stvara se varijabla „save_dir“ i poziva se funkcija „select_save_directory“.

```
52 def select_save_directory():
53     root = Tk()
54     root.withdraw()
55     root.attributes('-topmost', True)
56     save_dir = filedialog.askdirectory(title="Select Save Directory")
57     root.destroy()
58     return save_dir
59
60 save_dir = select_save_directory()
61 if not save_dir:
62     print("Save directory selection cancelled. Exiting.")
63 else:
64     np.savetxt(os.path.join(save_dir, "signal_t.txt"), signal_t, delimiter=',')
65     np.savetxt(os.path.join(save_dir, "signal_1.txt"), signal_1, delimiter=',')
66     np.savetxt(os.path.join(save_dir, "signal_2.txt"), signal_2, delimiter=',')
67     np.savetxt(os.path.join(save_dir, "signal_3.txt"), signal_3, delimiter=',')
68     np.savetxt(os.path.join(save_dir, "noise.txt"), noise, delimiter=',')
69     print(f"Files saved to {save_dir}")
70
```

61,62- provjerava jeli varijabla „save_dir“ prazna, ako je, ispisuje poruku „Save directory selection cancelled. Exiting.“

63- ako varijabla „save_dir“ nije prazna, provodi se blok linija.

64- poziva se funkcija „savetxt“ iz biblioteke „Numpy“ koja koristi tri parametra pomoću kojih sprema datoteku. Prvi parametar je definiran funkcijom „join“ iz submodula „path“ od modula „os“; funkciji se daju dva argumenta, varijabla „save_dir“, koja sadrži lokaciju gdje se želi spremiti datoteka i ime datoteke, „signal_t.txt“.Drugi parametar definira koji podatci se upisuju u datoteku, niz brojeva varijable „signal_t“, dok treći parametar definira da se koristi zarez kao separator kod decimalnog zapisa.

65-68- analogno liniji 64

69- ispisuje se poruka „Files saved to „odabrana lokacija“ “.

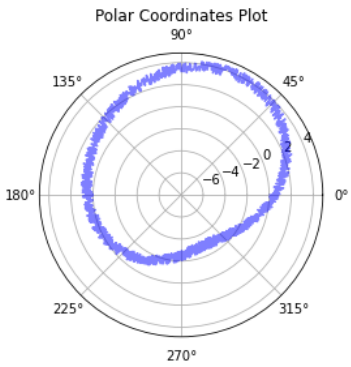
5. ANALIZA REZULTATA MJERENJA

U ovom poglavlju, program za obradu kružnosti biti će korišten za analizu i filtraciju podataka, te za računanje parametara odstupanja od kružnosti. Prije obrade realnih podataka, najprije je potrebno sa umjetno generiranim podatcima provjeriti točnost rada programa.

5.1. Provjera učitavanja podataka

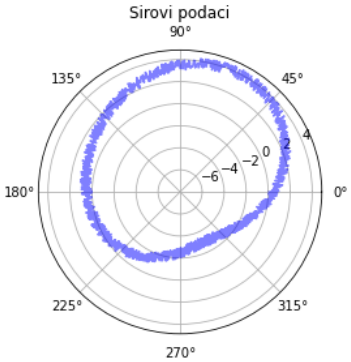
Kako bi se provjerilo učitava li program točno podatke, zadani su nasumični parametri, generirana je datoteka, te je datoteka učitana u program za obradu kružnosti. [Slika 34] prikazuje zadane početne parametre i prikaz koji stvara program za generiranje signala, dok [Slika 35] prikazuje učitane podatke u programu za obradu podataka, te se može zaključiti da su podatci točno učitani..

```
n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 1
amplitude_1 = 2

harmonic_2 = 2
amplitude_2 = 1

harmonic_3 = 1
amplitude_3 = 1

amplitude_noise = 0.5

bias = 0.5
```



a) **b)**

Slika 34. Generirani podatci za provjeru učitavanja podataka, a) početni parametri, b) prikaz podataka

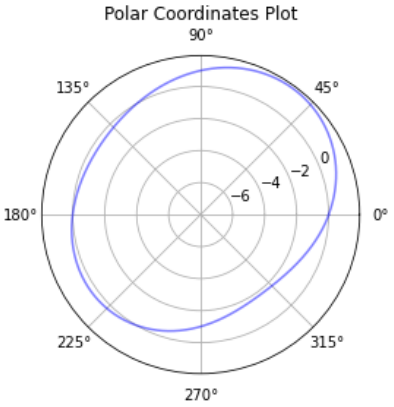


Slika 35. Prikaz učitanih podataka

5.2. Provjera uklanjanja ekscentra limakon aproksimacijom i Fourierovom transformacijom

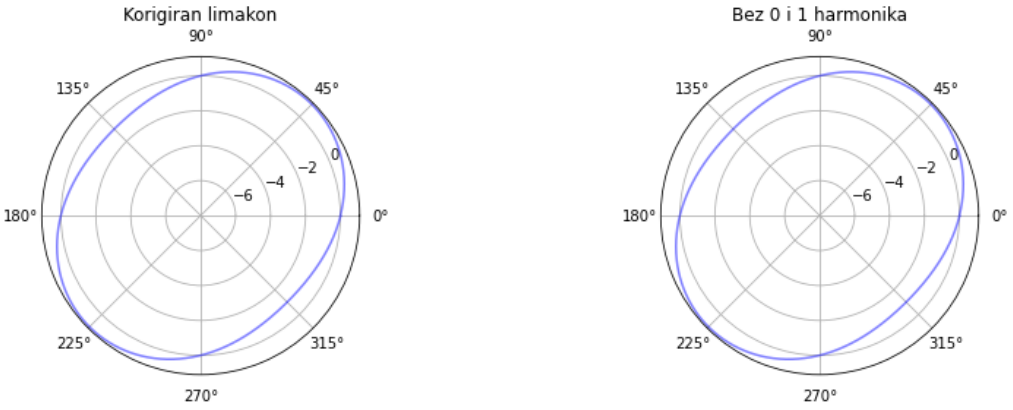
Slijedi provjera točnosti matematičke aproksimacije limakona, te uklanjanja limakona pomoću Fourierove transformacije. [Slika 36] prikazuje početne parametre i prikaz koji stvara program za generiranje signala, dok [Slika 37] daje prikaz podataka sa uklonjenim ekscentrom, tj. limakonom. Parametri su zadani tako da se jasno vidi korekcija limakona, te početni profil treba poprimiti oblik sličniji kružnici.

```
n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 1
amplitude_1 = 1

harmonic_2 = 2
amplitude_2 = 1

harmonic_3 = 0
amplitude_3 = 0

amplitude_noise = 0

bias = 0
```



a) **b)**

Slika 36. Generirani podatci za provjeru uklanjanja ekscentra, a) početni parametri, b) prikaz podataka



a) **b)**

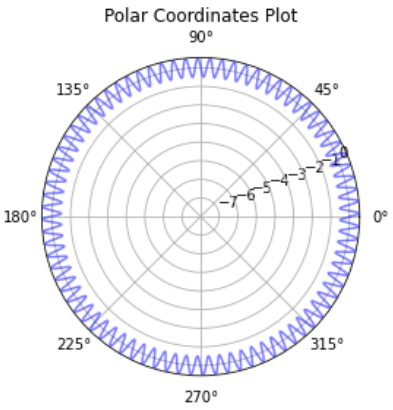
Slika 37. Prikaz uklanjanja ekscentra, a) limakon aproksimacijom, b) Fourierovom transformacijom

Donosi se zaključak da je program dobro uklonio utjecaj ekscentra iz podataka na oba načina; očekivano je da [Slika 37] daje dva ista prikaza, što se uistinu i dogodilo.

5.3. Provjera Gaussove filtracije podataka

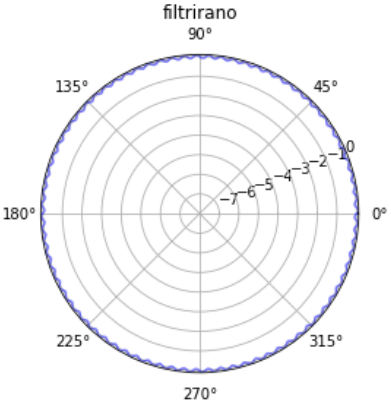
Slijedi provjera rada funkcije prigušivanja signala koja uklanja sve harmonike reda većeg od 50. Parametri su zadani tako da nakon filtracije profil poprima oblik gotovo savršene kružnice; zadani harmonik je reda 80, dok je svi harmonici redova većeg od 50 prigušuju. [Slika 38] prikazuje početne parametre i prikaz koji stvara program za generiranje signala, dok [Slika 39] daje prikaz podataka nakon provedenog filtera za prigušivanje.

```
n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 80
amplitude_1 = 0.5

harmonic_2 = 0
amplitude_2 = 0

harmonic_3 = 0
amplitude_3 = 0

amplitude_noise = 0

bias = 0
```



a) **b)**

Slika 38. Generirani podatci za provjeru prigušivanja podataka, a) početni parametri, b) prikaz podataka



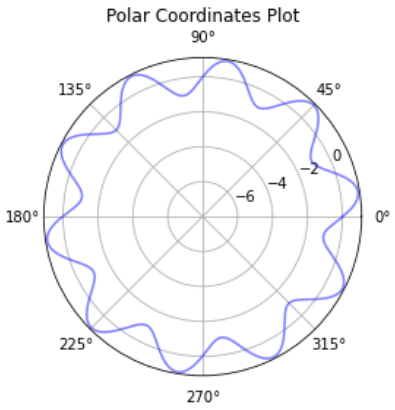
Slika 39. Prikaz prigušivanja podataka Gaussovim filterom

Donosi se zaključak da je program dobro prigušio harmonike reda većeg od 50.

5.4. Provjera proračuna parametara kružnosti

Konačno, potrebno je ustvrditi računa li program dobro parametre vezane uz kružnost. Parametri su zadani tako da se jednostavno može provjeriti jeli izračun dobar; postavljanjem da varijabla „harmonic_1“ iznosi 10, i da varijabla „amplitude_1“ iznosi 1, dok su svi ostali parametri jednaki nuli, [Slika 40], jednostavno se može odrediti da su RON_v i RON_p jednaki jedan, dok će RON_t iznositi dva, [Tablica 4].

```
n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 10
amplitude_1 = 1

harmonic_2 = 0
amplitude_2 = 0

harmonic_3 = 0
amplitude_3 = 0

amplitude_noise = 0

bias = 0
```



a) **b)**

Slika 40. Generirani podatci za provjeru računanja parametara kružnosti, a) početni parametri, b) prikaz podataka

Tablica 4. Izračunati parametri kružnosti za postavljene parametre

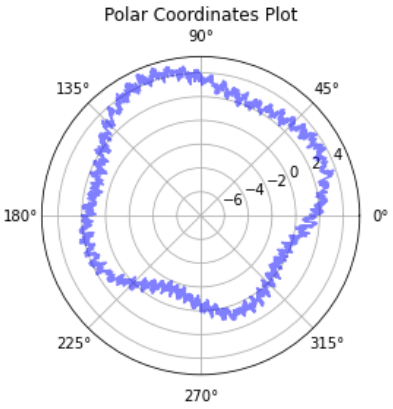
	RONv, µm	*RONp*, µm	*RONt*, µm	*RONq*, µm
a)	-1,000	1,000	2,000	0,707
b)	-1,000	1,000	2,000	0,707
c)	-0,973	0,973	1,945	0,688

Mala odstupanja od očekivanih vrijednosti posljedica su primjenjivana Gaussove filtracije na profil koji ne sadrži harmonike višeg reda, međutim takav profil u realnosti ne postoji, te je greška mala i stoga se može ignorirati.

5.5. Provjera ukupnog rada programa sa nasumično zadanim parametrima

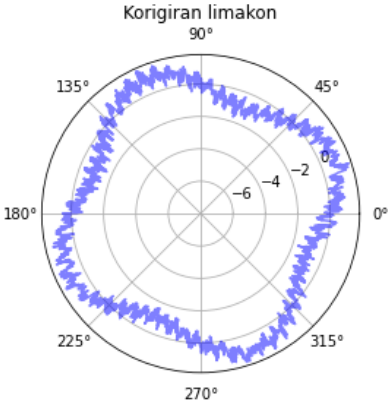
Ustvrđeno je da svaki aspekt programa za obradu podataka točno radi. Sada će biti zadani početni parametri tako da otprilike predstavljaju realne rezultate mjerenja, te će uz grafički prikaz obrađenih podataka također biti napisani i parametri vezani uz kružnost, nakon svake iteracije njihovog računanja, te će naposlijetku biti prikazana usporedba sirovih podataka koji nisu zaglađeni Gaussovim filtrom sa onim podatcima na kojima je provedena filtracija. [Slika 41] prikazuje postavljene početne parametre i prikaz generiranih podataka. [Slika 42] prikazuje rezultate uklanjanja ekscentra limakon aproksimacijom, dok [Slika 43] prikazuje rezultate uklanjanja ekscentra Fourierovom transformacijom. [Slika 44] prikazuje rezultate ukupne obrade podatka, te [Slika 45] uspoređuje podatke prije i poslije primjene Gaussovog filtera.

```
6  n = 1500
7
8  theta = np.arange(0, n) * 2 * np.pi / n
9
10 harmonic_1 = 1
11 amplitude_1 = 2
12
13 harmonic_2 = 4
14 amplitude_2 = 1
15
16 harmonic_3 = 64
17 amplitude_3 = 0.25
18
19 amplitude_noise = 0.5
20
21 bias = 1.5
```



a) **b)**

Slika 41. Generirani podatci za sveukupnu provjeru programa, a) početni parametri, b) prikaz podataka



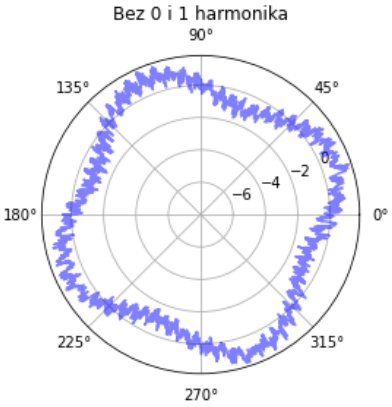
„RONv“= -1,700

„RONp“= 1,658

„RONt“= 3,358

„RONq“= 0,781

Slika 42. Uklanjanje ekscentra limakon aproksimacijom



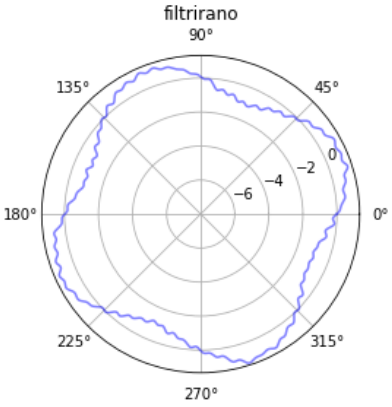
„RONv_FFT“= -1,700

„RONp_FFT“= 1,658

„RONt_FFT“= 3,357

„RONq_FFT“= 0,781

Slika 43. Uklanjanje ekscentra Fourierovom transformacijom



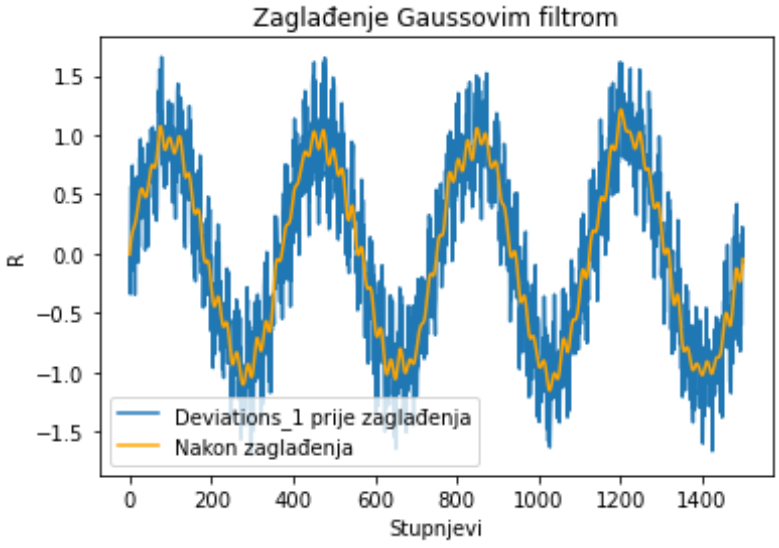
„RONv_Filtrirano“= -1,154

„RONp_Filtrirano“= 1,215

„RONt_Filtrirano“= 2,369

„RONq_Filtrirano“= 0,707

Slika 44. Prigušivanje podataka Gaussovim filterom



Slika 45. Usporedba filtriranih i nefiltriranih generiranih podataka

Pošto je utvrđeno da program dobro radi, moguće je analizirati rezultate obrade podataka. [Tablica 5] prikazuje parametre kružnosti i njihovu promjenu nakon primjene Gaussove filtracije.

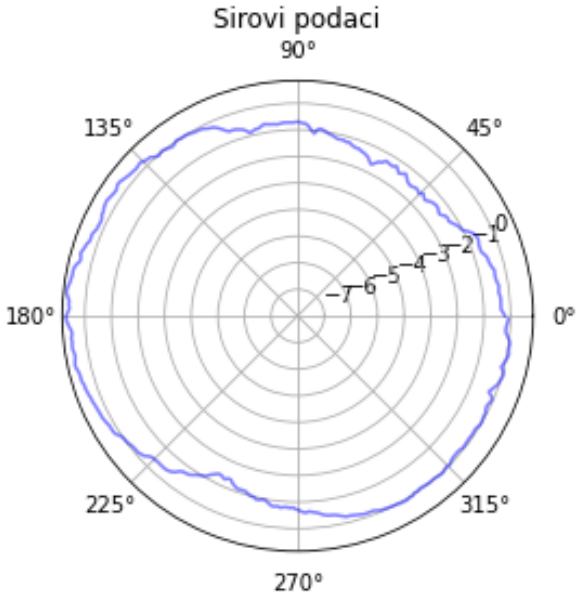
Tablica 5. Parametri kružnosti tokom rada programa pri obradi generiranih podataka, a) nakon uklanjanja ekscentra limakon aproksimacijom, b) nakon uklanjanja ekscentra Fourierovom transformacijom, c) nakon prigušivanja Gaussovim filterom, d) promjena parametara, e) promjena parametara, %

	RONv, µm	*RONp*, µm	*RONt*, µm	*RONq*, µm
a)	-1,700	1,658	3,358	0,781
b)	-1,700	1,658	3,358	0,781
c)	-1,154	1,215	2,369	0,707
d)	0,545	0,443	0,989	0,074
e)	32,084%	26,731%	29,440%	9,433%

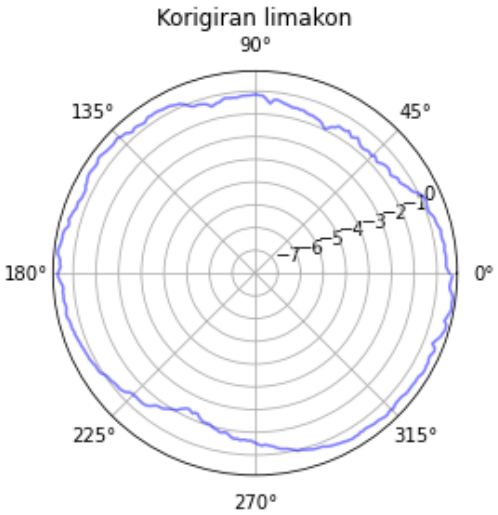
Prisutna je iznimno velika promjena parametara, što je očekivano pošto je generirani profil napravljen sa nasumičnim parametrima, koji samo vizualno stvaraju sliku profila koja bi sličila nekom rezultatu mjerenja; program uklanja greške pri mjerenju (ekscentar) i prigušuje harmonike visokih redova (šum i UPR>50), dok zadani početni parametri stvaraju vrlo jak utjecaj ekscentra i buke; upravo stoga obrada podataka daje toliko različite rezultate. Kod obrade realnih podataka, promjena parametara biti će znatno manja, pošto se za ispitni uzorak koristi etalonski prsten koji je iznimno napravljen sukladno normi, te njegove dimenzije su definirane vrlo uskim tolerancijama, promjena u parametrima bit će značajno manja.

5.6. Obrada realnih mjernih rezultata

Za podatke koriste se rezultati mjerenja etalonskog prstena, mjerenje provedeno u Laboratoriju za precizna mjerenja dužina Fakulteta strojarstva i brodogradnje, na uređaju Mahr MMQ3 (rezultati mjerenja nalaze se u prilogu). [Slika 46] prikazuje potpuno neobrađene podatke, dok [Slika 47] prikazuje rezultate uklanjanja ekscentra limakon aproksimacijom, dok [Slika 48] prikazuje rezultate uklanjanja ekscentra Fourierovom transformacijom.



Slika 46. Neobrađeni podatci referentnog prstena



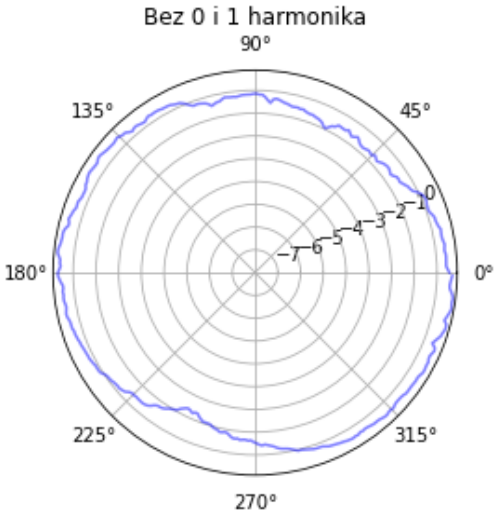
„RONv“= -1,352

„RONp“= 0,783

„RONt“= 2,135

„RONq“= 0,534

Slika 47. Podatci referentnog prstena nakon uklanjanja ekscentra limakon aproksimacijom



„RONv_FFT“= -1,352

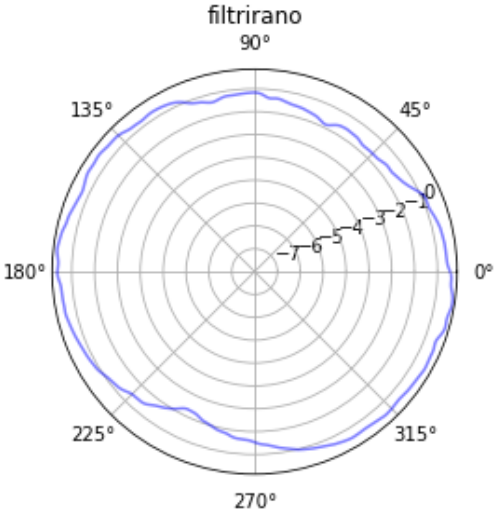
„RONp_FFT“= 0,783

„RONt_FFT“= 2,135

„RONq_FFT“= 0,534

Slika 48. Podatci referentnog prstena nakon uklanjanja ekscentra Fourierovom transformacijom

[Slika 49] prikazuje rezultate ukupne obrade podataka, te [Slika 50] prikazuje usporedbu podataka prije i poslije primjene gaussovog filtera. [Tablica 6] sadrži vrijednosti parametra vezanih uz kružnost, te njihovu promjenu tokom rada rada programa (obrade podataka).



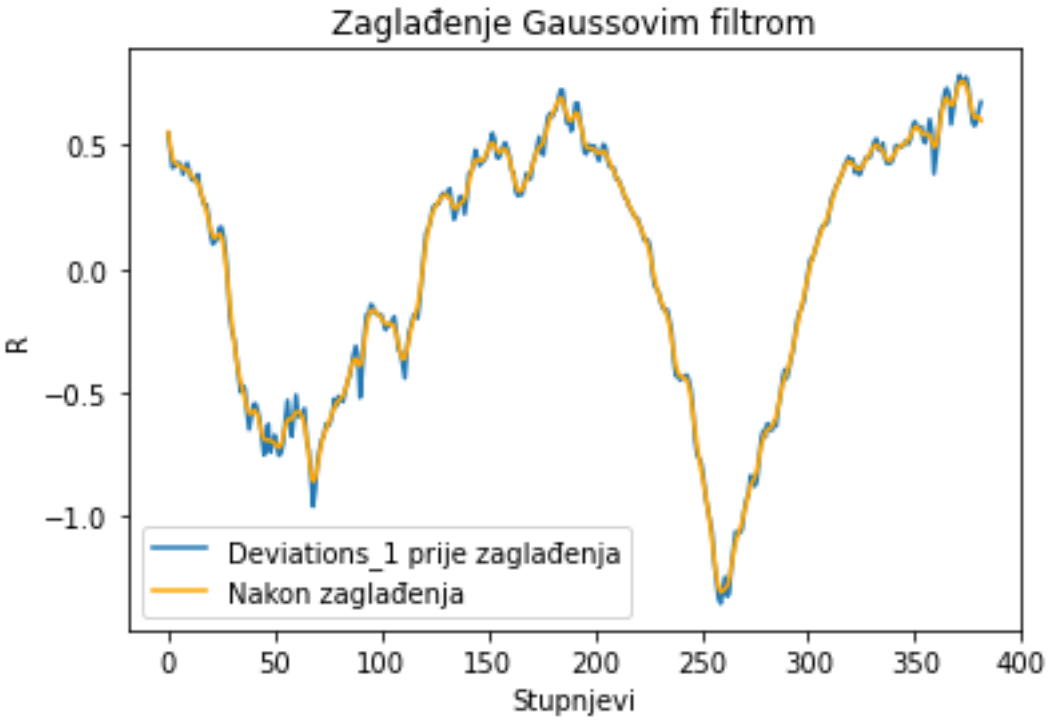
„RONv_Filtrirano“= -1,306

„RONp_Filtrirano“= 0,758

„RONt_Filtrirano“= 2,065

„RONq_Filtrirano“= 0,531

Slika 49. U potpunosti obrađeni podatci referentnog prstena



Slika 50. Usporedba filtriranih i nefiltriranih podataka referentnog prstena

Tablica 6. Parametri kružnosti tokom rada programa pri obradi izmjerenih podataka, a) nakon uklanjanja ekscentra limakon aproksimacijom, b) nakon uklanjanja ekscentra Fourierovom transformacijom, c) nakon prigušivanja Gaussovim filterom, d) promjena parametara, e) promjena parametara, %

	RONv, µm	*RONp*, µm	*RONt*, µm	*RONq*, µm
a)	-1,352	0,783	2,135	0,534
b)	-1,352	0,783	2,135	0,534
c)	-1,306	0,758	2,065	0,531
d)	-0,045	0,024	0,070	0,003
e)	3,363%	3,167%	3,291%	0,489%

Kao što je i očekivano, utjecaj programa na promjenu parametara je znatno manji nego kod generiranih podataka, međutim, 3% (stotine mikrometara) u strojarskom kontekstu je i dalje značajna razlika; može utjecati na životni vijek proizvoda, čvrstoću dosjeda, montažu komponenata itd., dok na uzorcima koji nisu etaloni bi postojala još veća razlika. Dokazano je da je obrada podataka potrebna.

6. ZAKLJUČAK

Predmeti kružnog presjeka omogućavaju niz strojarski primjena; prenošenje sile, smanjivanje trenja, promjena smjera gibanja, ravnomjernu raspodjelu sile, stvaranje dosjeda i sl. Kako bi se optimirala efikasnost strojeva, raste potreba za sve užim tolerancijama kružnosti. U teoretskom dijelu diplomskog rada objašnjeno je što je kružnost, kojim je parametrima opisana, kojim se metodama mjeri, koje su njihove prednosti i mane, koja je metoda odabrana za potrebe ovog diplomskog rada. Također su opisane greške koje se mogu pojaviti tokom mjerenja, te načini na koje se uklanjaju; uređaj za mjerenje kružnosti prikuplja mjerne podatke koji su definirani svojim iznosom i redoslijedom uzimanja, te predstavljaju mjerni signal. Takav signal se Fourierovom transformacijom preslikava u frekvencijsku domenu, te se provodi obrada podataka; uklanjanje ekscentra i filtriranje harmonika viših redova. U svrhu obrade podataka napravljena su dva programa; glavni program koji učitava podatke te ih obrađuje u svrhu određivanja odstupanja od kružnosti i popratni program koji generira signal na temelju zadanih parametara. Način rada oba programa opisan je u eksperimentalnom dijelu diplomskog rada; program za generiranje signala koristi se kako bi se ispitala točnost programa za obradu podataka. Generiranjem signala za koje se unaprijed zna kakvi će biti rezultati nakon obrade podataka potvrđena je točnost glavnog programa, te je potom provedeno ispitivanje na kontrolnom prstenu koji se koristi kao etalon za kružnost. Uočena je značajna razlika u stupnju promjene parametara kružnosti ovisno o podatcima koji se obrađuju; kod predmeta koji značajno odstupaju od kružnosti digitalna filtracija može imati značajan utjecaj na krajnje parametre, dok kod predmeta kojima je kružnost propisana uskim tolerancijama, filtracija će imati značajno manji (ali nipošto zanemariv) utjecaj. Dokazano je da je korištenje programa za obradu podataka potrebno, neovisno o tome koliko predmet odstupa od kružnosti. Za potrebe diplomskog rada, program za obradu podataka je napravljen u obliku biblioteke funkcija, te nije primjenjiv kao gotova aplikacija koja se može koristiti u praksi. Za daljnji razvoj programa, potrebno je napraviti korisničko sučelje i integrirati program da radi u skladu sa strojem za mjerenje kružnosti, bez posrednog učitavanja podataka.

LITERATURA

[1] Geometrical product specifications (GPS)-Geometrical tolerancing-Tolerances of form, orientation, location and run-out (ISO 1101:2017) BSI Standards Publication, 2017.
[2] Runje, B. Predavanja iz kolegija Teorija i tehnika mjerenja. Zagreb: Fakultet strojarstva i brodogradnje; 2014.
[3] Sui W, Zhang D. Four Methods for Roundness Evaluation. Phys Procedia 2012;24:2159–64. https://doi.org/10.1016/j.phpro.2012.02.317.
[4] Exploring Roundness A fundamental guide to the measurement of cylindrical form 3rd Edition,. Leicester: Taylor Hobson Limited; 2011.
[5] BSI Standards Publication Geometrical product specifications (GPS)-Roundness. 2011.
[6] Prikaz izbočenja na predmetu: https://www.mmsonline.com/articles/determining-out-of-roundness-at-the-point-of-manufacture, pristupljeno: 09.07. 2024.
[7] Runje, B., Šimunović, V., Baršič, G. Vježbe iz kolegija teorija i tehnika mjerenja; 2024.
[8] Farago FT, Curtis MA. Handbook of Dimensional Measurement Fifth Edition. 2014.
[9] Muralikrishnan B, Raja J. Computational surface and roundness metrology. Springer London; 2009. https://doi.org/10.1007/978-1-84800-297-5.
[10] Prikaz Fourierove transformacije: https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Signal_Processing_with_NumPy_Fourier_Transform_FFT_DFT.php, pristupljeno: 09.07. 2024.
[11] Prikaz mogućih rezultata Fourierove transformacija: https://community.sw.siemens.com/s/article/what-is-the-fourier-transform, pristupljeno: 09.07. 2024.
[12] Jednadžbe vezane uz Fourierovu transformaciju: https://www.robots.ox.ac.uk/~sjrob/Teaching/SP/l7.pdf, pristupljeno: 09.07. 2024.
[13] BSI Standards Publication Geometrical product specifications (GPS)-Roundness. 2011.

PRILOZI

I. Program za obradu signala

```python
import numpy as np
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter import filedialog




def read_data_file():

    root = tk.Tk()
    root.withdraw()

    root.attributes('-topmost', True)

    file_path = filedialog.askopenfilename(
        title="Select a file",
        filetypes=(("Data files", "*.*"), ("All files", "*.*"))
    )

    root.destroy()

    if file_path:
        try:
            data = np.loadtxt(file_path)
            return data
        except OSError as e:
            print(f"Error loading file: {e}")
            return np.array([])
    else:
        return np.array([])

data = read_data_file()







n = len(data)
def lslimacon(data):

    theta = np.arange(0, n) * 2 * np.pi / n
    r = np.mean(data)
    a = 2 * np.sum(data * np.cos(theta)) / n
    b = 2 * np.sum(data * np.sin(theta)) / n
    R_ls = r + a * np.cos(theta) + b * np.sin(theta)
    Deviation = data - R_ls
    OOR = np.max(Deviation) - np.min(Deviation)
    print ("OOR",OOR)
    return Deviation, r, a, b, OOR, R_ls, theta

if data.size > 0:
    Deviation, r, a, b, OOR, R_ls, theta = lslimacon(data)
    print("r:", r)
    print("a:", a)
    print("b:", b)
    print("OOR:", OOR)
    print("R_ls:", R_ls)
else:
    print("No data to process.")
```

```
64
65
66
67
68
69
70
71  def plot_circle(R, theta, title):
72      plt.figure()
73      ax = plt.subplot(111, projection='polar')
74      ax.scatter(theta, R, s=0)
75      ax.plot(theta, R, color='blue', alpha=0.5)
76      ax.set_rmin(-8)
77      ax.set_title(title)
78      plt.show()
79
80
81
82
83
84
85
86  plot_circle(data, theta, 'Sirovi podaci')
87  plot_circle(Deviation, theta,'Korigiran limakon')
88
89
90
91
92
93
94
95  def find_RONp_and_RONv(data):
96      RONp = np.max(data)
97      RONv = np.min(data)
98      RONq= np.sqrt((np.sum(np.square(data)) / len(data)))
99      return RONv, RONp, RONq
100
101 def calculate_difference(RONp, RONv):
102     return RONp - RONv
103
104 if data.size == 0:
105     print("Data is empty or file could not be loaded. Exiting.")
106 else:
107     RONv, RONp, RONq = find_RONp_and_RONv(Deviation)
108     RONt = calculate_difference(RONp, RONv)
109     print(f'RONv= {RONv}')
110     print(f'RONp= {RONp}')
111     print(f'RONt= {RONt}')
112     print(f'RONq= {RONq}')
113
114
115
116
117
118
119
120 har_orig = np.fft.rfft(data)
121 har = np.fft.rfft(Deviation)
122 har_alter = har_orig.copy()
123 har_alter[0] = 0
124 har_alter[1] = 0
125 Deviation_1 = np.fft.irfft(har_alter)
126
```

```
RONv, RONp, RONq = find_RONp_and_RONv(Deviation_1)
RONt = calculate_difference(RONp, RONv)

print(f'RONv_FFT= {RONv}')
print(f'RONp_FFT= {RONp}')
print(f'RONt_FFT= {RONt}')
print(f'RONq_FFT= {RONq}')
plot_circle (Deviation_1, theta,'Bez 0 i 1 harmonika')









fc=50

alpha=np.sqrt(np.log(2)/np.pi)

mag=np.abs(har_alter)

faza=np.angle((har_alter))

a2 = [0] * len(mag)

for i in range(len(mag)):
    a2[i] = mag[i] * (np.exp(-np.pi * (alpha * i / fc)**2))

a2[0] = 0
a2[1] = 0

complex_numbers = a2 * np.exp(1j * faza)

smoothed_arr = np.fft.irfft(complex_numbers )

plot_circle (smoothed_arr, theta,'filtrirano')

def plot_graph(Deviation_1, smoothed_arr):

  plt.figure()
  plt.plot(Deviation_1, label='Deviations_1 prije zaglađenja', color='tab:blue')
  plt.plot(smoothed_arr, label='Nakon zaglađenja', color='orange')
  plt.legend()
  plt.title("Zaglađenje Gaussovim filtrom")
  plt.xlabel("Stupnjevi")
  plt.ylabel("R")
  plt.show()

plot_graph(Deviation_1, smoothed_arr)

RONv, RONp, RONq = find_RONp_and_RONv(smoothed_arr)
RONt = calculate_difference(RONp, RONv)

print(f'RONv_Filtrirano= {RONv}')
print(f'RONp_Filtrirano= {RONp}')
print(f'RONt_Filtrirano= {RONt}')
print(f'RONq_Filtrirano= {RONq}')

```

II. Generator signala

```python
import numpy as np
import matplotlib.pyplot as plt
from tkinter import Tk, filedialog
import os

n = 1500

theta = np.arange(0, n) * 2 * np.pi / n

harmonic_1 = 1
amplitude_1 = 2

harmonic_2 = 4
amplitude_2 = 1

harmonic_3 = 64
amplitude_3 = 0.25

amplitude_noise = 0.5

bias = 1.5


signal_1 = amplitude_1 * np.sin(np.linspace(0, 360 * harmonic_1, n) * np.pi / 180)
signal_2 = amplitude_2 * np.sin(np.linspace(0, 360 * harmonic_2, n) * np.pi / 180)
signal_3 = amplitude_3 * np.sin(np.linspace(0, 360 * harmonic_3, n) * np.pi / 180)
noise = np.random.uniform(-amplitude_noise, amplitude_noise, n)

signal_t = signal_1 + signal_2 + signal_3 + noise + bias

def plot_graph(signal_t):

  plt.figure()
  plt.plot(signal_t)
  plt.title('Combined Signal')
  plt.xlabel('Sample')
  plt.ylabel('Amplitude')
  plt.show()

def plot_circle(R, theta):
    plt.figure()
    ax = plt.subplot(111, projection='polar')
    ax.scatter(theta, R, s=0)
    ax.plot(theta, R, color='blue', alpha=0.5)
    ax.set_rmin(-8)
    ax.set_title('Polar Coordinates Plot')
    plt.show()

plot_circle(signal_t, theta)
plot_graph(signal_t)

def select_save_directory():
    root = Tk()
    root.withdraw()
    root.attributes('-topmost', True)
    save_dir = filedialog.askdirectory(title="Select Save Directory")
    root.destroy()
    return save_dir

save_dir = select_save_directory()
if not save_dir:
    print("Save directory selection cancelled. Exiting.")
else:
    np.savetxt(os.path.join(save_dir, "signal_t.txt"), signal_t, delimiter=',')
    np.savetxt(os.path.join(save_dir, "signal_1.txt"), signal_1, delimiter=',')
    np.savetxt(os.path.join(save_dir, "signal_2.txt"), signal_2, delimiter=',')
    np.savetxt(os.path.join(save_dir, "signal_3.txt"), signal_3, delimiter=',')
    np.savetxt(os.path.join(save_dir, "noise.txt"), noise, delimiter=',')
    print(f"Files saved to {save_dir}")


```

III. Podaci dobiveni mjerenjem ispitnog uzorka

-0.20736580	-0.63152320	-1.44503500	-1.31259200	-0.71369610
-0.28107980	-0.66825930	-1.48974700	-1.29132400	-0.73593120
-0.35382700	-0.73762300	-1.52164900	-1.25096200	-0.76034130
-0.35261860	-0.84976480	-1.50835700	-1.18570700	-0.73254760
-0.34125940	-0.99525930	-1.46364500	-1.20915100	-0.72771390
-0.34246780	-1.05640600	-1.36600400	-1.16564700	-0.70209530
-0.35116850	-1.08419900	-1.28673100	-1.17555600	-0.67913510
-0.39418840	-1.15960500	-1.36987100	-1.17604000	-0.73399770
-0.37461190	-1.22945200	-1.42545900	-1.12915300	-0.80142780
-0.35286030	-1.29688200	-1.33531000	-1.08564900	-0.80142780
-0.38621280	-1.27102200	-1.25144600	-1.05882300	-0.84855640
-0.42391570	-1.28214000	-1.33531000	-1.00637700	-0.89326820
-0.42971610	-1.37639700	-1.32854300	-0.95441450	-0.76831700
-0.42367400	-1.44406900	-1.30993300	-0.91598660	-0.69170280
-0.40820620	-1.39186500	-1.29059900	-0.94547220	-0.65665850
-0.47732810	-1.34086900	-1.42135000	-1.11223500	-0.61267170
-0.50488020	-1.33651900	-1.44866100	-0.99695110	-0.59647890
-0.53799100	-1.36117100	-1.56225300	-0.85242340	-0.60832140
-0.53509090	-1.40080700	-1.66907700	-0.75768280	-0.52614850
-0.58681140	-1.46968700	-1.59608900	-0.75333250	-0.41859860
-0.66487570	-1.53639200	-1.48201300	-0.70378700	-0.31902440
-0.69895330	-1.47669600	-1.42594200	-0.70402870	-0.23298450
-0.68783580	-1.40781600	-1.38147200	-0.72843890	-0.21099110
-0.68082700	-1.51850800	-1.36092900	-0.72022160	-0.17183810
-0.63442340	-1.47379600	-1.30824100	-0.72215510	-0.11383370

-0.09280708	0.38524610	0.61412190	0.50778040	-0.10585810
-0.08386473	0.36615290	0.60300430	0.48047000	-0.10126610
-0.07226385	0.30041460	0.62765620	0.47901990	-0.14791130
-0.03407760	0.30283140	0.64481590	0.47829480	-0.18972280
-0.01595122	0.32700000	0.69194450	0.43261640	-0.30863190
-0.00990909	0.36156090	0.72964730	0.43575830	-0.37751220
-0.01740133	0.38355430	0.73158090	0.41569840	-0.37364520
0.02368514	0.36760310	0.66197550	0.39467180	-0.40023060
-0.02706873	0.34222610	0.60517950	0.36252770	-0.39104650
-0.08676495	0.28808860	0.61726380	0.35793560	-0.39201330
-0.05534589	0.28688020	0.58125270	0.32941680	-0.38548780
-0.00990909	0.21582480	0.63611520	0.31636580	-0.40699770
0.02924390	0.20422390	0.70112850	0.29823940	-0.44808420
0.02223503	0.21896670	0.70378700	0.28398000	-0.53074050
-0.02972727	0.22114190	0.64457420	0.28277160	-0.65859190
0.03915299	0.26730370	0.59696220	0.25546120	-0.72843890
0.14791130	0.31781590	0.51962300	0.23201770	-0.74269840
0.15588690	0.30549000	0.50971390	0.19310640	-0.76807530
0.19794010	0.30573170	0.55031700	0.19334810	-0.82825490
0.26343680	0.37702880	0.52880700	0.18996450	-0.91115290
0.24192680	0.40627270	0.55128370	0.15806210	-0.96384020
0.21437470	0.45340130	0.55080040	0.06235476	-1.01193600
0.23322610	0.49593790	0.51406430	0.00435033	-1.06293100
0.24700220	0.43503320	0.49666290	-0.01063414	-1.18788200
0.26682040	0.43237470	0.53436580	-0.02658536	-1.26884700
0.30524830	0.53049880	0.56675160	-0.06912194	-1.34401100
0.33860080	0.59357860	0.55901770	-0.09329046	-1.36769600

-1.31549200	-0.68251880	-0.01522616	-0.05172062	-0.06090465
-1.27029700	-0.62209750	0.00217517	-0.05558757	-0.01619290
-1.35319500	-0.59164520	0.00894235	-0.06960531	0.03770288
-1.34642800	-0.63466510	0.03673614	-0.07468070	0.06791352
-1.24371200	-0.60493790	0.04398669	-0.06936362	0.03891130
-1.18087400	-0.56578490	0.05727937	-0.06162971	0.02513525
-1.11102600	-0.53992460	0.02296009	-0.08289799	0.04737028
-1.12504400	-0.47491120	0.03238580	-0.06815520	0.00773392
-1.12625300	-0.42874940	-0.02924390	-0.02175166	-0.06477161
-1.11755200	-0.40506430	-0.01039246	-0.00942572	-0.14621950
-1.04939700	-0.39588020	-0.05220398	-0.03431929	-0.16507090
-1.00782700	-0.37485360	-0.03963636	-0.05413746	-0.14380260
-0.99525930	-0.32845010	-0.01933481	-0.04785365	-0.10803320
-0.92178700	-0.27310420	-0.00362528	-0.10972500	-0.07878935
-0.93242110	-0.22766740	-0.00797561	-0.12567630	
-0.97495780	-0.23564300	-0.01715964	-0.08724833	
-0.95876490	-0.21268290	0.00918404	-0.04108647	
-0.87852540	-0.19407310	0.03383591	-0.09474056	
-0.79828590	-0.17062970	0.03770288	-0.27213740	
-0.77411740	-0.14138580	-0.01015077	-0.20204870	
-0.80650320	-0.13437690	-0.02586031	-0.13751880	
-0.75840790	-0.13703550	-0.00072506	-0.04253658	
-0.78861850	-0.14017740	-0.06598003	-0.01764301	
-0.79876930	-0.09498225	-0.09739910	0.02827716	
-0.76831700	-0.05268735	-0.10223280	0.04253658	
-0.79441900	-0.04108647	-0.10609980	0.01474279	
-0.72626380	-0.01933481	-0.09353214	-0.11262530	