Konstrukcija i vođenje rotacijskog crtala

Buzjak, Dražen

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:104442

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-23**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Dražen Buzjak

Zagreb, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Mladen Crneković

Student:

Držen Buzjak

Zagreb, 2017.

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

Zahvaljujem se prof. dr. sc. Mladenu Crnekoviću na korisnim savjetima i primjedbama te na izradi terminala za komunikaciju između računala i mikrokontrolera.

Dražen Buzjak

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur. broj:	

DIPLOMSKI ZADATAK

Student: **Dražen Buzjak** Mat. br.:0035194222

Naslov rada na hrvatskom jeziku: **Konstrukcija i vođenje rotacijskog crtala**

Naslov rada na engleskom jeziku: **Design and Control of Rotational Plotter**

Opis zadatka:

Brz i točan prijenos crteža na papir za inženjere i danas predstavlja izazov. Pored klasičnih translacijskih konstrukcija crtala postoje i drugi koncepti koji nisu dovoljno istraženi, a mogli bi imati prednosti. Jedan takav koncept je kombinacija translacijsko-rotacijskog crtala. Papir je pričvršćen za rotacijski bubanj, a na uzdužnom translacijskom stupnju gibanja nalazi se sredstvo za pisanje. Potrebno je napraviti konstrukciju mehaničkog, energetskog i upravljačkog sustava za A4 veličinu papira.
U radu je potrebno:

- Konstruirati crtalo sa svim funkcionalnim dijelovima.
- Odabrati potrebne motore, energetsku elektroniku, mjerne uređaje i elemente upravljačkog sustava.
- Napraviti programsku podršku upravljačkog sustava sposobnu za crtanje elementarnih geometrijskih likova (najmanje za pravac i kružnicu).
- Predložiti s kojim komercijalnim programskim alatima za crtanje će se uređaj spajati i objasniti način spajanja (tip komunikacije i protokol).
- Procijeniti materijalnu vrijednost uređaja.

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

Zadatak zadan: 28. rujna 2017.

Datum predaje rada: 30. studenog 2017.

Predviđeni datum obrane: 6., 7. i 8. prosinca 2017.

Zadatak zadao:

Prof. dr. sc. Mladen Crneković

Predsjednica Povjerenstva:

Prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ I
POPIS SLIKA II
POPIS TABLICA IV
POPIS OZNAKA V
SAŽETAK VII
SUMMARY VIII
1. UVOD 1
1.1. Općenito o crtalima (plotterima) 1
1.2. Cilj 4
2. KONSTRUKCIJA CRTALA 5
2.1. Koncept 5
2.2. CAD model 6
2.3. Motori 7
2.4. Bubanj i pomični klizač 9
2.5. Elementi za prijenos snage i gibanja 10
2.6. Vodilice i ležajevi 12
3. ELEKTRONIKA 13
3.1. Senzori 13
3.2. Upravljanje motorom 18
3.3. Mikrokontroler 20
4. UPRAVLJANJE CRTALOM 23
4.1. Struktura CNC sustava 23
4.2. G kôd 24
4.3. Struktura upravljanja rotacijskim crtalom 26
4.4. Implementacija interpretera 27
4.5. Očitavanje pozicije i brzine 30
4.6. Linearni interpolator 32
4.7. Regulacija pozicije i brzine 35
4.8. Kružna interpolacija 45
5. PRIJENOS CRTEŽA NA PAPIR 52
6. PROCJENA VRIJEDNOSTI UREĐAJA 57
7. ZAKLJUČAK 61
LITERATURA 63
PRILOZI 64

POPIS SLIKA

Slika 1. Primjer crtaće ploče ... 2
Slika 2. Primjer rasterske (lijevo) i vektorske (desno) grafike [2] ... 2
Slika 3. Translacijski plotter ... 3
Slika 4. Pojednostavljeni koncept upravljanja ... 4
Slika 5. Koncept rotacijskog crtala ... 5
Slika 6. CAD model 1 ... 6
Slika 7. CAD model 2 ... 7
Slika 8. IG-32GM servomotor ... 8
Slika 9. Pomični klizač ... 9
Slika 10. Presjek bubnja ... 10
Slika 11. Koncepti mjerenja pozicije: direktno (gore) i indirektno (dolje) ... 13
Slika 12. Dvokanalni inkrementalni magnetski enkoder ... 14
Slika 13. LS7084 raspored pinova i multiplikacija impulsa [7] ... 15
Slika 14. Način spajanja LS7084 i određivanje širine impulsa [7] ... 16
Slika 15. Prikaz širine multipliciranih impulsa oscilokopom ... 16
Slika 16. Smještaj optičkih vrata ... 17
Slika 17. Načelo PWM modulacije [8] ... 18
Slika 18. Tranzistorski H-most ... 20
Slika 19. LMD18200T shema (lijevo), izvedba (desno) ... 20
Slika 20. Atmega 2560, Arduino MEGA razvojna pločica ... 22
Slika 21. UART komunikacija ... 22
Slika 22. Struktura CNC sustava [6] ... 23
Slika 23. Struktura upravljanja rotacijskim crtalom ... 26
Slika 24. Blok dijagram interpretera ... 27
Slika 25. Dijagram toka za traženje brojeva iza „X“ adrese ... 29
Slika 26. Linearna interpolacija [13] ... 32
Slika 27. Dijagram toka linearnog interpolatora ... 34
Slika 28. Struktura regulacije pozicije i brzine ... 35
Slika 29. Odziv Y osi za vrijednosti: K_{PY}=200, K_{RY}=0.01, K_{IY}=0.04, F=2000 ... 37
Slika 30. Odziv Y osi za vrijednosti: K_{PY}=300,K_{RY}=0.01, K_{IY}=0.04,F=2000 ... 38
Slika 31. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}=0.01, K_{IY}=0.04, F=2000 ... 39
Slika 32. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}=0.001, K_{IY}=0.04, F=2000 ... 40
Slika 33. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}0.001, K_{IY}=0.04, F=1000 ... 41
Slika 34. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}=0.001, K_{IY}=0.04, F=5000 ... 41
Slika 35. Odziv X osi za vrijednosti: K_{PX}=400, K_{RX}=0.005, K_{IX}=0.04, F=2000 ... 42
Slika 36. Odziv X osi za vrijednosti: K_{PX}=400, K_{RX}=0.005, K_{IX}=0.04, F=1000 ... 43
Slika 37. Odziv X osi za vrijednosti: K_{PX}=400, K_{RX}=0.005, K_{IX}=0.04, F=4000 ... 43
Slika 38. Rezultat odziva X i Y osi ... 44
Slika 39. Greške kružne interpolacije [6] ... 45
Slika 40. Usporedba odziva različitih metoda kružne interpolacije ... 47
Slika 41. Prikaz općeg slučaja kružne interpolacije ... 48
Slika 42. Dijagram toka za računanje referentnih točaka kod kružne interpolacije ... 50
Slika 43. Odziv na kružnu interpolaciju ... 51
Slika 44. Pozicioniranje slike ... 52
Slika 45. Izdvajanje rubova slike ... 53

Slika 46. Trajektorije slike 53
Slika 47. Ishodište slike i parametri za generiranje G kôda 54
Slika 48. Putanja alata 55
Slika 49. Terminal za prijenos G kôda na mikrokontroler 55
Slika 50. Ispis slike 56
Slika 51. Prikaz rotacijskog crtala s prednje strane 57
Slika 52. Prikaz rotacijskog crtala s bočne strane 58

POPIS TABLICA

Tablica 1. Karakteristike motora IG-32GM 8
Tablica 2. S2M remenice 11
Tablica 3. Korištene riječi G kôda 25
Tablica 4. Značajke metoda kružne interpolacije [15] 47
Tablica 5. Popis elektroničkih i mehaničkih komponentni 58

POPIS OZNAKA

Oznaka	Jedinica	Opis
U_N	V	Nazivni napon armature DC motora
I_N	A	Nazivna struja DC motora
P_N	W	Nazivna snaga DC motora
n_N	min^{-1}	Nazivna brzina vrtnje DC motora
n_R	min^{-1}	Nazivna brzina vrtnje na izlazu iz reduktora DC motora
i	-	Prijenosni omjer planetarnoga reduktora
T_R	Nm	Nazivni okretni moment na iznaznom vratilu reduktora
D_B	mm	Promjer bubnja
$\Delta smax$	mm	Točnost pozicioniranja
N_m	Imp/okr	Rezolucija enkodera
D_1	mm	Diobeni promjer remenice 1 (br. 6 slika 6)
D_2	mm	Diobeni promjer remenice 2 (br. 7 slika 6)
D_3	mm	Diobeni promjer remenice 3
β	∘	Inkrement obuhvatnog kuta na remenici
L_W	mm	Duljina zupčastog remena
U_{SR}	V	Srednja vrijednost napona
t_c	s	Vrijeme ciklusa
U_b	V	Napona napajanja(baterije)
f_{1s}	Hz	Broj impulsa u jednoj sekundi
n_0	min^{-1}	Brzina u praznom hodu DC motora
T_{INT}	s	Vrijeme izvođenja prekidne rutine
Δx	mm	Jedinični pomak za X os
Δy	mm	Jedinični pomak za Y os
v_{xmax}	mm/min	Najveća ostvariva linearna brzina za X os
v_{ymax}	mm/min	Najveća ostvariva linearna brzina za Y os
L	mm	Duljina ukupne putanje(trajektorije)
X_{REF}	mm	Udaljenost konačne točke od početne (X_0,Y_0), po X osi
Y_{REF}	mm	Udaljenost konačne točke od početne (X_0,Y_0), po Y osi
ΔL	mm	Duljina linijskog segmenta
ΔX	mm	Duljina linijskog segmenta projicirana na X os

ΔY	mm	Duljina linijskog segmenta projicirana na Y os
incX	mm	Varijabla za akumuliranje reference po X osi
incY	mm	Varijabla za akumuliranje reference po Y osi
KR	-	Proporcionalno pojačanje PI regulatora
KI	-	Integralno pojačanje PI regulatora
Δv_Y	mm/min	Jedinična brzina za Y os
Δv_x	mm/min	Jedinična brzina za X os
$\Theta(i)$	rad	Kut u *i*-tom koraku kod kružne interpolacije
α	rad	Inkrement kuta kod kružne interpolacije
S	mm	Duljina tetive kod kružne interpolacije
δ	rad	Ukupni kut kod kružne interpolacije
N	-	Broj linijskih segmenata kod kružne interpolacije
A	-	Koeficijent kod kružen interpolacije
B	-	Koeficijent kod kružne interpolacije
DX	mm	Inkrement pomaka za X os kod grube interpolacije kružnice
DY	mm	Inkrement pomaka za Y os kod grube interpolacije kružnice

SAŽETAK

Kroz ovaj rad izrađeno je funkcionalno rotacijsko crtalo prema uzoru na CNC sustave. Rad se ugrubo može podijeliti u tri dijela. Prvi dio je sama konstrukcija rotacijskog crtala, u kojem su dimenzionirani i odabrani elementi za prijenos snage i gibanja te su odabrani ležajevi i vodilice. Uz sve to prikazan je i CAD model cijeloga crtala. U drugom su dijelu predstavljeni elektronički elementi pomoću kojih se upravlja rotacijskim crtalom. Dok su u trećem, objašnjeni algoritmi upravljanja te je implementirana kaskadna regulacija pozicije i brzine za pojedine osi crtala. Također, implementirani su interpreter te linearni i kružni interpolator u mikrokontroleru prema uzoru na CNC sustave. Na kraju su dani odzivi i rezultati ispisa probne slike.

Ključne riječi: mikrokontroler, regulacija pozicije i brzine, CNC, DC servomotor

SUMMARY

Through this paper functional rotary plotter was made, according to the CNC systems. Paper can be divided into three sections. The first section is the construction itself, in which elements for power and motion transfer are dimensioned and selected, as well as bearings and guides. Also the CAD model of entire rotary plotter is shown. In the second section electronic components are presented, by which rotary plotter is controlled. While in the third section control algorithms are explained, and cascade structure of position and velocity control loops are implemented for each axis separately. Also, interpreter, linear and circular interpolator are implemented in the microcontroller, according to the CNC systems. At the end of the paper, results of trial plot, and responses of axis are given.

Key words: microcontroller, positon anda velocity closed loop controll, CNC, DC servomotor

1. UVOD

Gledajući unazad kroz prošlost, čovjeku je u genima zapisivanje svojih ideja. Počevši od kamenog doba, gdje je čovjek na zidovima stijena i špilja ostavljao svoje tragove pa do kasnijih razdoblja, gdje se pojavljuje papir. Pojavom idustrijske revolucije te masovne proizvodnje, određena skupina ljudi trebala je davati naputke što ostali trebaju raditi. Tu se pojavljuju tehnički nacrti i slična dokumentacija. Tehnički crtež kao sredstvo za sporazumijevanje, odnosno komunikaciju između projektanta/konstruktora i radnika, npr. na alatnom stroju, jednoznačno definira informacije za izradu nekog proizvoda.

U drugoj polovici 20. stoljeća, razvijanjem računala te računalne grafike, nacrti se više ne crtaju ručno, već se crtanje izvodi pomoću računala. Problem koji se tu nameće je kako prenijeti taj crtež na papir ili neko drugo sredstvo, budući da je sam crtež na računalu beskoristan. Upravo se zbog toga počinju razvijati prva crtala (plotteri), koja omogućavaju prijenos nacrta ili neke slike na papir.

Važno je napomenuti da, kako se sve više razvijaju CAD (*eng.* Computer Aided Design) tehnologije, koncept proizvodnje pomoću nacrta polako izumire. Naime, danas je CAD model sredstvo sporazumijevanja između proizvodnje i konstrukcije.

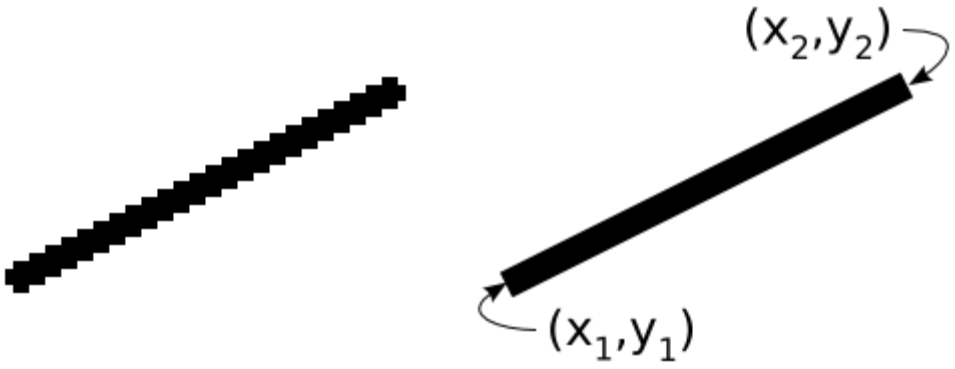
1.1. Općenito o crtalima (plotterima)

Kako je spomenuto već u uvodnim riječima, u drugoj polovici 20. stoljeća razvojem računala, počinje se razvijati i računalna grafika. Projektiranje i crtanje počinje se odvijati na računalima, te se javlja potreba za prijenos crteža s računala na papir. Uređaji koji to omogućuju nazivaju se plotteri. Oni su zamijenili cijelu „vojsku “ crtača koji su ručno crtali tehničke nacrte na crtaćim pločama. Jedna crtaća ploča prikazana je na slici 1.

Plotter je računalni pisač za ispis vektorske grafike [1]. U samom početku bili su poznatiiji kao analogni XY-pisači, nakon čega se javljaju digitalno upravljani plotteri. Postoje dvije vrste grafike, a to su: rasterska ili bitmap grafika i vektorska grafika [2]. Kod rasterske

Slika 1. Primjer crtaće ploče

grafike, slika je definirana redovima i stupcima piksela, gdje svaki piksel ima svoju boju, dok je kod vektorske grafike slika definirana ravnim i zakrivljenim linijama. Dakle, ravna linija će biti definirana dvjema točkama kao kod vektora. Razlika između rasterske i vektorske grafike uočljiva je ukoliko se prikaz slike poveća, kao što je prikazano na slici 2.



Slika 2. Primjer rasterske (lijevo) i vektorske (desno) grafike [2]

Razlika između plotera i konvencionalnog printera je u tome što plotter pomiče alat u ravnini, dok obični printer (npr. InkJet) nema alata. Taj alat može biti laserska zraka, nož ili kao što je slučaj u ovom radu, može biti olovka. Prednost plottera nasprem običnih pisača bila je ta što su mogli ispisivati velike formate nacrta. Danas se plotteri koriste za rezanje papira i slično.

Tipična kofiguracija translacijkog plottera prikazana je na slici 3. Tu se može primijetiti da nema razlike između plottera i modernog CNC upravljanog alatnog stroja. Upravo je ta činjenica iskorištena u ovom radu, kod izrade upravljanja rotacijskog crtala.

Slika 3. Translacijski plotter

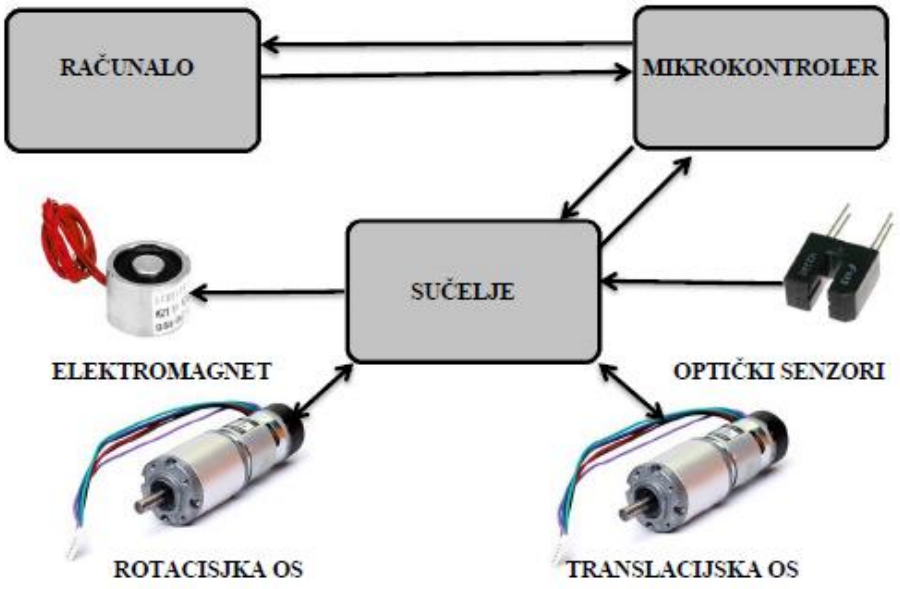
Translacijski plotter, prikazan na slici 3, ima dva i pol stupnja slobode gibanja. Dva stupnja su translacijska, kao kod CNC alatnog stroja, dok ovih pola stupnja slobode gibanja stoji iza toga da alat može postići samo dvije krajnje pozicije. Kako se radi o uređaju koji treba prenijeti crtež u dvije dimenzije, onda je nepotrebno implementirati i treću os.

Uz translacijke plottere, postoje i rotacijski koji imaju jednu translacijsku os, a drugu rotacijsku. Takvi plotteri nisu baš komercijalno zastupljeni, stoga su zanimljiv predmet za istraživanje njihovih mogućnosti.

1.2. Cilj

Cilj ovoga rada je napraviti maketu rotacijskog crtala s dva stupnja slobode gibanja koja može proizvoljno pozicionirati alat koji ostavlja trag na papiru. Prvi korak koji je potrebno učiniti je izraditi CAD model makete kako bi se mogli proizvesti potrebni dijelovi. Konstrukcijski dio je opisan u drugom poglavlju ovoga rada. U njemu su proračunati i izabrani prijenosnici snage i gibanja, ležajevi i motori te je dan funkcionalni prikaz mehanizma kao i CAD model.

Drugi korak je prijenos linija sa slike u naredbe razumljive upravljačkom dijelu. Problem koji se nameće je kako postići da se motori gibaju po određenoj putanji, a da pritom opisuju željenu trajektoriju. Ideja je da se pomoću besplatnog software-a vektorska slika pretvori u G-kod koji će se kasnije interpretirati u računalu razumljiv jezik. Ideja je preuzeta iz CNC sustava. Pomoću mikrokontrolera zamišljeno je implementirati jednostavni CNC sustav sastavljen od interpretera, koji podatke koji pristižu sa serijske veze od računala interpretira u referentne signale, a oni služe za zadavanje referenci regulacije pozicije. Mikrokontroler ujedno ima i ulogu servo-regulatora, budući da se brine o tome da svaki motor prati određenu referencu uz određenu brzinu. Na slici 4. je prikazan koncept upravljanja maketom rotacijskog crtala.



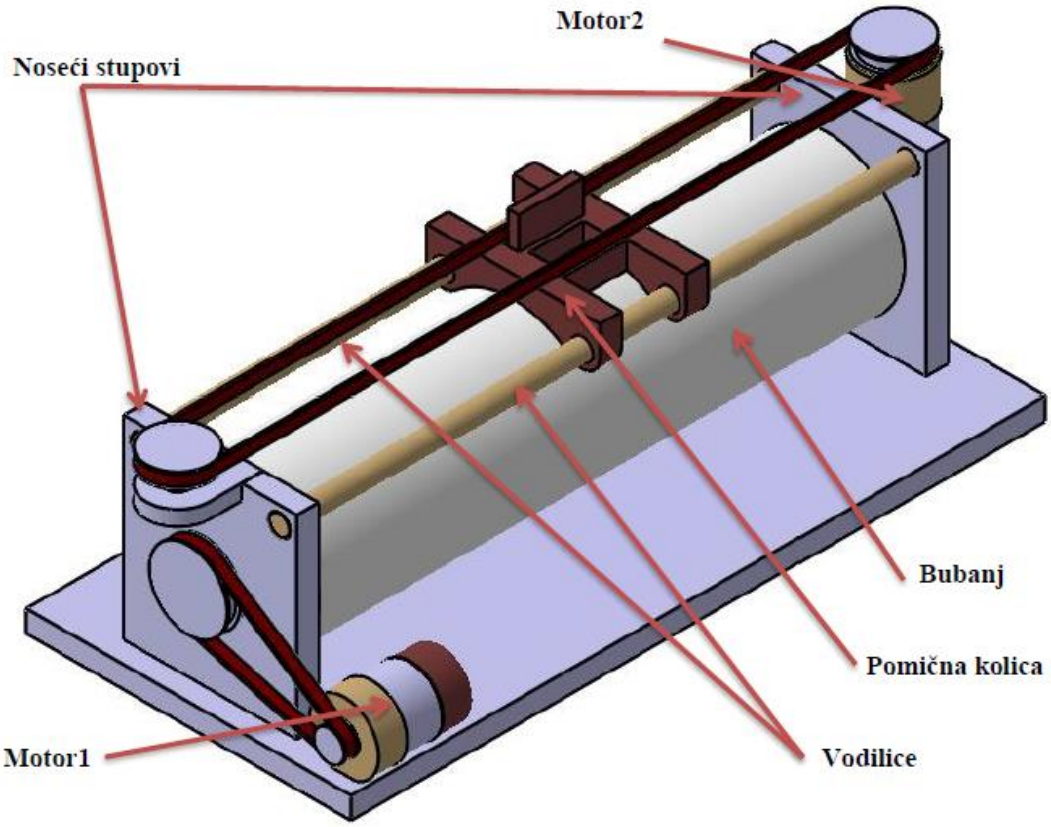
Slika 4. Pojednostavljeni koncept upravljanja

2. KONSTRUKCIJA CRTALA

U ovom poglavlju opisan je postupak konstrukcije makete rotacijskog crtala te su odabrani svi potrebni elementi. U uvodnom dijelu je predstavljena ideja, odnosno koncept rotacijskog crtala. Za potrebe konstrukcije CAD modela rotacijskog crtala korišten je programski paket Catia V5R19.

2.1. Koncept

Koncept rotacijskog crtala prikazan je na slici 5. Sa slike 5 mogu se vidjeti osnovni dijelovi makete. Zamišljeno je da se na bubanj pričvrsti papir A4 formata, a olovka na pomični klizač. Olovka ovdje ima ulogu alata, koji ostavlja trag na papiru. Sa svake strane nalaze se noseći stupovi načinjeni od aluminija, tako da doprinose krutosti cijelog crtala. Pomični klizač klizi po čeličnim šipkama koje imaju ulogu vodilica.



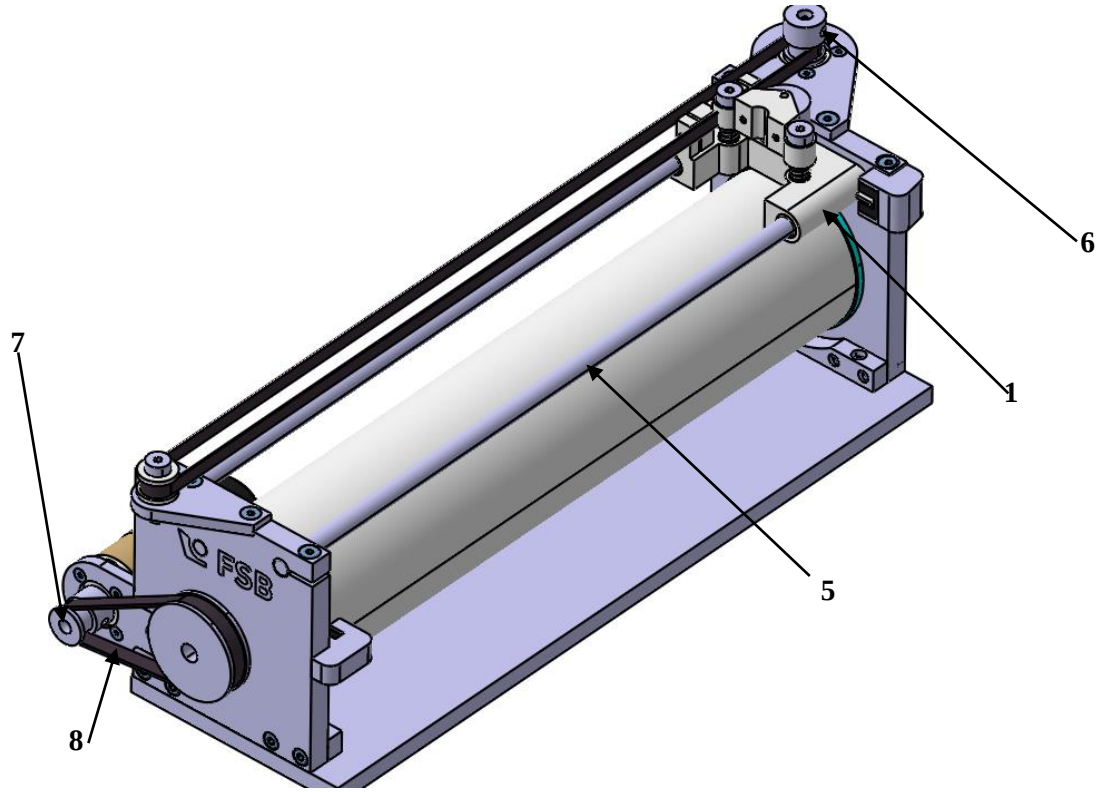
Slika 5. Koncept rotacijskog crtala

Rotacijsku os pogoni motor 1 sa slike 5., preko remenskog prijenosa. Obodna brzina bubnja odgovara brzini gibanja rotacijske osi. Linarnu os pogoni motor 2 sa slike 5. Zupčasti remen je čvrsto spojen s pomičnim klizačem, te je samim time obodna brzina remenice na motoru 2 jednaka brzini gibanja linearne osi. Kasnije, kod upravljanja, rotacijska os će se zamjeniti linearnom, kako bi se moglo lakše upravljati samim crtalom.

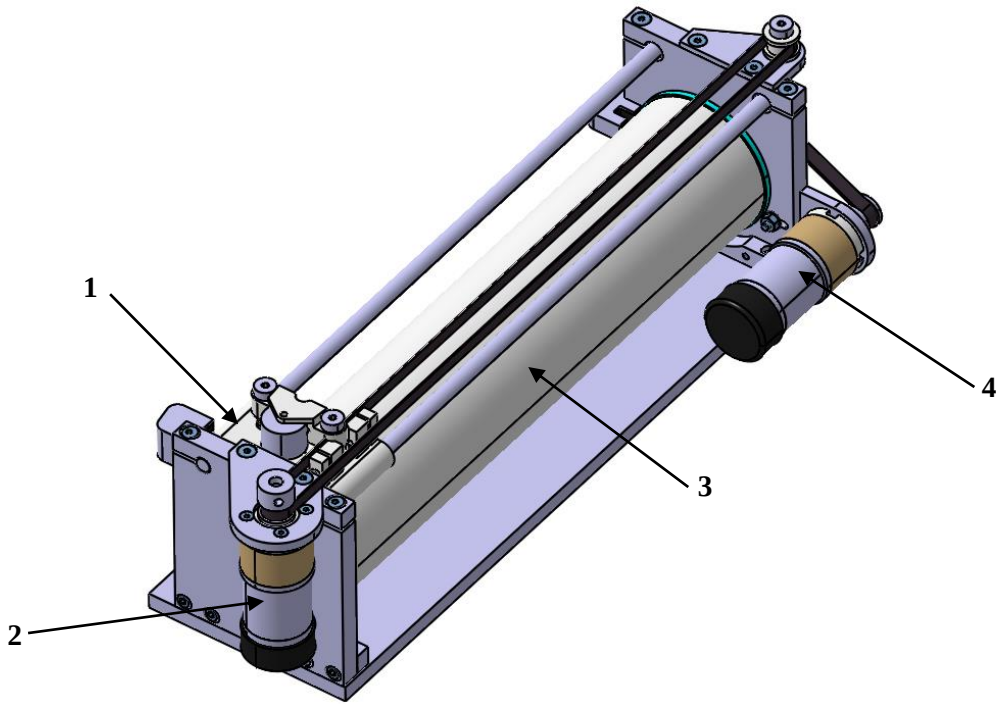
U sljedećim potpoglavljima opisani su osnovni elementi te je dan prikaz CAD modela crtala.

2.2. CAD model

Već je u početku spomenuto da je korišten programski paket Catia V5R19 koji ima ulogu CAD/CAM software-a. Dijelovi su konstruirani tako da se kasnije mogu izraditi obradom odvajanja te aditivnom tehnologijom. Na slikama 6 i 7 prikazan je CAD model iz dvije različite perspektive.



Slika 6. CAD model 1



Slika 7. CAD model 2

U daljnjim potpoglavljima opisani su pojedini dijelovi sa slika 6. i 7. Prema CAD modelu prikazanom na slikama 6. i 7., izrađeni su dotični dijelovi.

2.3. Motori

Inženjerksa praksa je da se motori odabiru na temelju zahtjeva i karakteristika sustava kojega oni pogone. Naime, u ovom radu takav pristup nije ostvaren, budući da su motori već unaprijed bili poznati, stoga se na temelju njihovih karakteristika određuju karakteristike sustava kojega će pogoniti.

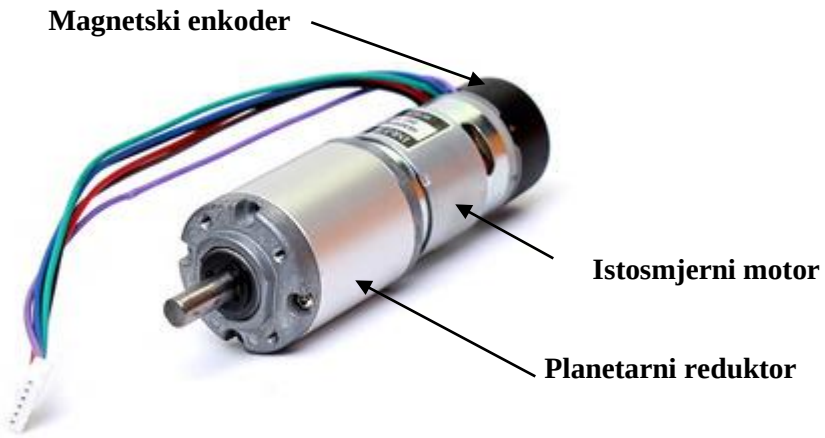
Motori korišteni u ovom radu su IG-32GM. Radi se o istosmjernom servomotoru koji na svojemu izlaznom vratilu motora ima reduktor koji povećava okretni moment te smanjuje izlaznu brzinu vrtnje. Na stražnjoj strani motora nalazi se dvokanalni magnetski inkrementalni enkoder, rezolucije 14 impulsa po okretaju. Ako su nam poznati svi prijenosni omjeri mehanizma, pomoću inkrementalnog enkodera, brojanjem impulsa, prati se zakret vratila elektromotora te se na taj način može izračunati točan položaj objekta upravljanja. Objekti

upravljanja u ovome radu su pomični klizač označen brojem 1 na slici 6., kojega pogoni motor označen brojem 2 na slici 7. te bubanj označen brojem 3 na slici 7., kojega pogoni motor označen brojem 4 na slici 6. Karakteristike servomotora dane su u tablici 1. Pomoću tih karakteristika će se dalje vršiti proračun elemenata za prijenos snage i gibanja.

Tablica 1. Karakteristike motora IG-32GM

Nazivni napon U_N	24V
Nazivna struja I_N	530 mA
Nazivni moment M_N	$0.765 \cdot 10^{-2}$ Nm
Nazivna snaga motora P_N	4.22 W
Nazivna brzina vrtnje n_N	5290 min^{-1}
Prijenosni omjer reduktora i	27
Nazivna brzina na izlazu iz reduktora n_R	195 min^{-1}
Nazivni moment na izlazu iz reduktora T_R	0.207 Nm

Na slici 8. prikazan je motor IG-32GM sa sastavnim dijelovima.



Slika 8. IG-32GM servomotor

2.4. Bubanj i pomični klizač

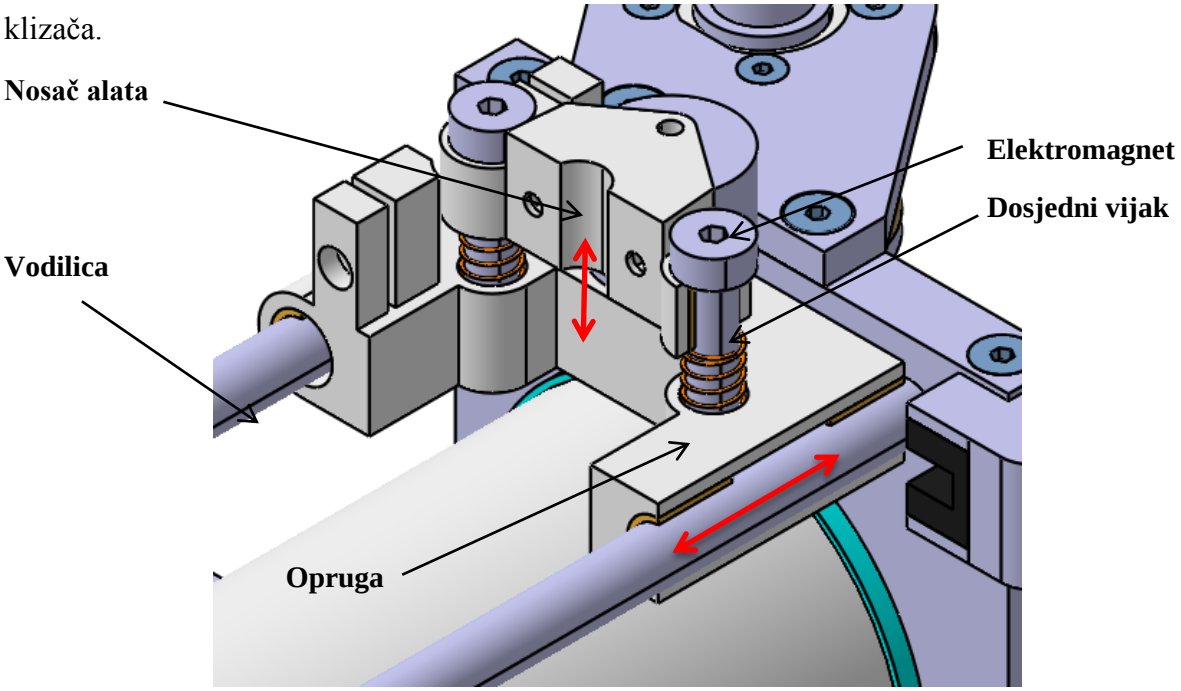
U ovom potpoglavlju opisani su bubanj na kojega se pričvršćuje papir te pomični klizač na kojega se pričvršćuje alat, odnosno olovka.

Kako je predviđeno da se na bubanj pričvrsti papir A4 formata, onda je potrebno odrediti promjer bubnja na kojega će isti i stati. Ako znamo dimenzije A4 formata (210x297mm) te ako kao kraću stranicu uzmemo opseg bubnja, može se dobiti promjer bubnja.

$$D_B \geq \frac{210}{\pi} = 66{,}88mm, \quad D_B = 68\ mm \tag{1}$$

Promjer bubnja odabran je da bude 68 mm. Kod dimenzioniranja duljine bubnja, uzeta je u obzir duljina duže stranice A4 papira te duljina pomičnog klizača. Naime, kako se pisaći element nalazi u sredini pomičnog klizača, potrebno je kompenzirati duljinu pomičnog klizača, kako bi se pisaći element mogao pozicionirati u bilo kojem dijelu papira.

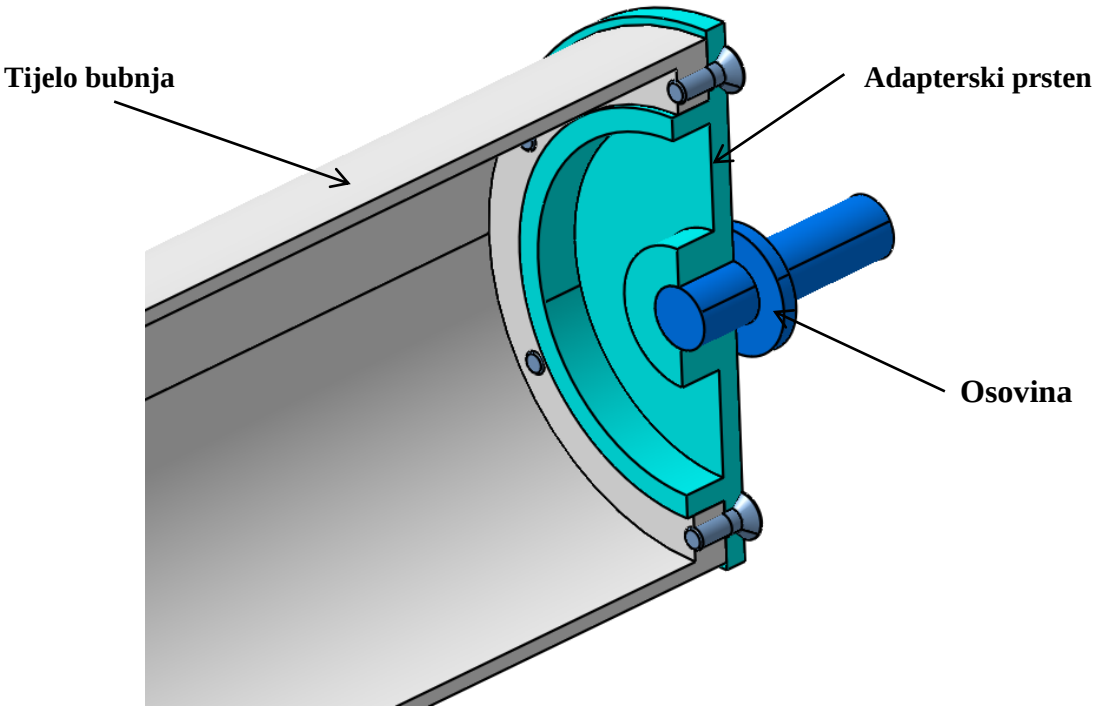
Već je ranije u ovom radu spomenuto da se radi o sustavu s dva i pol stupnja slobode. Ovih pola stupnja slobode koji se ovdje spominju, realizirano je pomoću elektromagneta, koji pozicionira alat, odnosno olovku u dva položaja. Na slici 9. je detaljan presjek pomičnog klizača.



Slika 9. Pomični klizač

Gibanje nosača alata u krajnji položaj ostvaruje se pomoću elektromagneta koji se može vidjeti sa slike 9. Elektromagnet je čvrsto vezan s nosačem alata, a u pomični klizač je smješten trajni magnet kojega privlači elektromagnet kada je aktiviran. Samim time ostvaruje se gibanje nosača alata. Dosjedni vijci (DIN 7379) služe kao vodilice za nosač alata. Opruge imaju ulogu da vrate nosač alata u početni položaj, nakon što se elektromagnet isključi.

Bubanj se sastoji od tri dijela, koji se mogu vidjeti na slici 10. Naime, bubanj je nemoguće izraditi iz jednog komada, a da mu unutrašnjost bude šuplja. Stoga je konstruiran adapterski prsten koji dolazi na čelo bubnja te se čvrsto spaja s osovinom.



Slika 10. Presjek bubnja

Bubanj je izrađen od ABS-a (Akronitril-Butadien-Stirel), dok su osovina i adapterski prsten izrađeni iz aluminija.

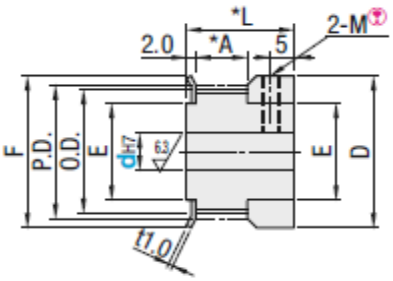
2.5. Elementi za prijenos snage i gibanja

Kod elemenata za prijenos snage i gibanja, prvenstveno se misli na zupčaste remene i remenice. Naime, kako se radi o sustavu za pozicioniranje, mora se osigurati da kod prijenosa gibanja ne dođe do proklizavanja remena po remenici, stoga se koriste zupčasti remeni i

remenice. Kako je promjer vrha olovke približno 0.5mm, onda se nameće zahtjev za točnošću pozicioniranja od Δs_{max}=0.5mm.

Remenica na motoru 1, označena brojem 6 na slici 6., odabrana je prema [3] tako da je diobeni promjer remenice čim manji, kako bi se smanjila radijalna sila na izlaznom vratilu na reduktoru. Preporuka je da se radijalna sila kod remenskog prijenosa sa zupčastim remenom uzima kao 150% obodne sile koja se prenosi, prema [4]. Podaci za remenice dani su u tablici 2.

Tablica 2. S2M remenice



mm	Number of Teeth																				
	14	15	16	18	20	22	23	24	25	26	28	30	32	34	36	40	44	48	50	60	72
P.D.	8.91	9.55	10.19	11.46	12.73	14.01	14.64	15.28	15.92	16.55	17.83	19.10	20.37	21.65	22.92	25.46	28.01	30.56	31.83	38.20	45.84
O.D.	8.40	9.04	9.68	10.95	12.22	13.50	14.13	14.77	15.41	16.04	17.32	18.59	19.86	21.14	22.41	24.96	27.50	30.05	31.32	37.69	45.33
D	12	12	14	14	16	18	18	20	20	22	22	22	12	14	14	18	20	20	20	30	34
F	12	12	14	14	16	18	18	20	20	22	22	22	25	25	28	30	32	35	35	44	50
E	6	6	8	8	10	11	11	13	13	14	14	14	16	16	18	20	23	25	25	32	38

Na zadnjem dijelu motora nalazi se inkrementalni enkoder s rezolucijom N_m= 14 impulsa po okretaju motora, prijenosni omjer reduktora nam je poznat iz tablice 1., i=27 te nam je poznata točnost pozicioniranja Δs_{max}=0.5 mm. Uzimjući sve te parametre u obzir, može se izračunati najveći mogući promjer remenice koji će zadovoljiti točnost pozicioniranja.

$$D_1 \leq \frac{\Delta s_{max} N_m i}{\pi} = \frac{0.5 \cdot 14 \cdot 27}{\pi} = 60.19\ mm \qquad (2)$$

Iz tablice 2. odabire se remenica s 20 zubi koja ima diobeni promjer D_1=12.73 mm. Za motor 2, označen brojem 7 na slici 6., odabire se ista remenica, dok je veću potrebno izračunati. Najmanji promjer veće remenice može se dobiti prema jednadžbi (3). Promjer bubnja D_B izračunat je pomoću jednadžbe (1).

$$D_3 \geq \frac{\pi D_B D_2}{N_m i \Delta s_{max}} = \frac{\pi \cdot 68 \cdot 12.73}{14 \cdot 27 \cdot 0.5} = 14.38\, mm \tag{3}$$

Iz tablice 2. odabire se remenica sa 60 zubi, diobenog promjera D_3=38.20 mm. Remenica sa 60 zubi odabrana je zato da bi se ostvario veći prijenosni omjer između pogonske i gonjene remenice, a time se i povećala točnost pozicioniranja. Sada kada su odabrane remenice, potrebno je odabrati adekvatni zupčasti remen. Konkretno treba izračunati duljinu remena označenog brojem 8 na slici 6. Duljina remena računa se kao:

$$\beta = \arcsin\left(\frac{d_3 - d_2}{2L}\right) = 11.13° \tag{4}$$

gdje je β inkrement obodnog kuta na remenici, koji govori koliko remen po obodu zahvaća remenicu. Nakon što je β izračunat, može se izračunati duljina remena L_W.

$$L_W = 2cos\beta L + \frac{d_3}{2}\left(\pi + 2\frac{\pi}{180}\beta\right) + \frac{d_2}{2}\left(\pi - 2\frac{\pi}{180}\beta\right) = 214.4mm \tag{5}$$

Sada je potrebno odabrati standardnu duljinu remena prema [3]. Kako nosač motora 2 sa slike 6., ima mogućnost napinjanja remena, koji dopušta promjenu međuosnog razmaka L=60±5 mm, onda se odabire tip remena S2M-220-6. Brojevi iz oznake imaju sljedeće značenje: 220 je duljina remena, dok je 6 širina remena.

2.6. Vodilice i ležajevi

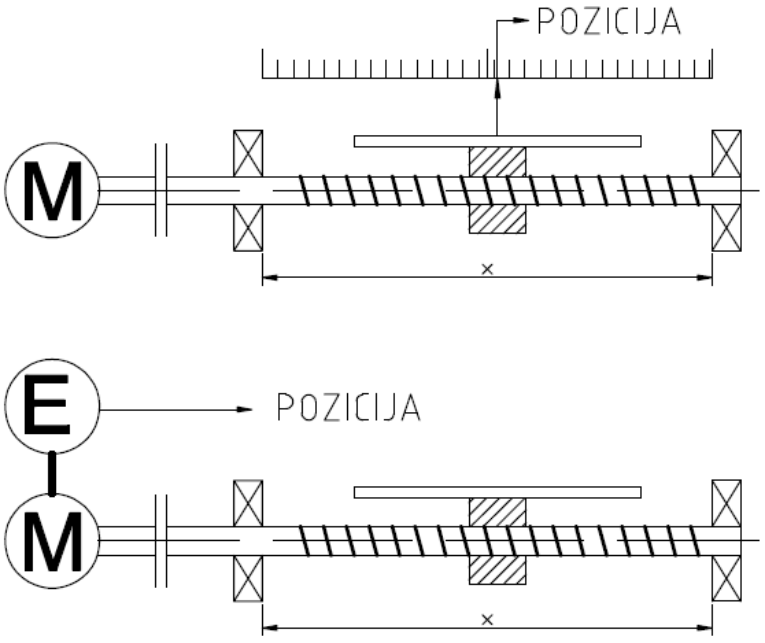
Vodilice pomičnog klizača realizirane su kao šipke od nehrđajućeg čelika, strojno obrađene na određeni promjer. Sa slike se može vidjeti izvedba vođenja pomičnih kolica. Između pomičnih kolica i vodilica nalaze se klizni ležajevi načinjeni od poliamida. Samim time isključena je potreba za podmazivanje vodilica. Ležajevi koji nose valjak također su izrađeni od poliamida, pomoću aditivne tehnologije. Ležajevi su konstruirani prema uzoru na DryLin firme IGUS[5].

3. ELEKTRONIKA

Rotacijsko crtalo tipičan je primjer mehatroničkog sustava, koji se ugrubo sastoji od mehaničkog i elektroničkog dijela. U prošlom poglavlju opisan je mehanički odnosno konstrukcijski dio, dok je u ovom poglavlju razmatran elektronički dio. Elektronički dio se sastoji od mikrokontrolera, senzora i periferne elektronike za obradu signala s istih te sklopova za upravljanje istomjernim motorom.

3.1. Senzori

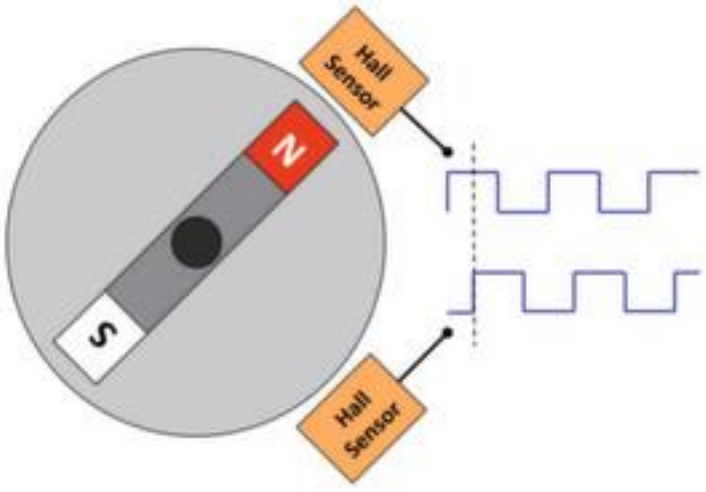
Rotacijsko crtalo kao sustav koji pozicionira alat (pisaći element) u proizvoljnu poziciju, ili vodi alat po određenoj trajektoriji, pogonjen je dvama istosmjernim motorima, kako je već ranije rečeno. Da bi se moglo izvršiti pozicioniranje alata ili vođenje po željenoj trajektoriji, potrebno je u svakom trenutku znati položaj alata. Tu se, u suštini, mogu primijeniti dva koncepta. Na slici 11. dani su shematski prikazi mjerenja pozicije stola jedne osi alatnog stroja, s direktnim pogonom preko navojnog vretena.



Slika 11. Koncepti mjerenja pozicije: direktno (gore) i indirektno (dolje)

Kod direktnog mjerenja pozicije, prikazanog na slici 11 (gore), mjeri se stvarni pomak objekta upravljanja preko mjerne letve, dok se kod indirektnog mjerenja pozicije mjeri pozicija motora te se preko prijenosnog omjera sustava može dobiti pozicija objekta upravljanja. Nedostatak indikretnog mjerenja je osjetljivost na zračnost sustava, dok je kod direktnog nedostatak manja rezolucija, ukoliko se radi o inkrementalnim enkoderima. U sklopu ovog rada realizirano je indirektno mjerenje pozicije, budući da korišteni motori već imaju na sebi inkrementalne enkodre pomoću kojih se dobiva pozicija motora, odnosno pozicija objekta upravljanja. Valja napomenuti da postoji i kombinacija navedenih koncepta mjerenja, no oni nisu razmatrani u ovome radu.

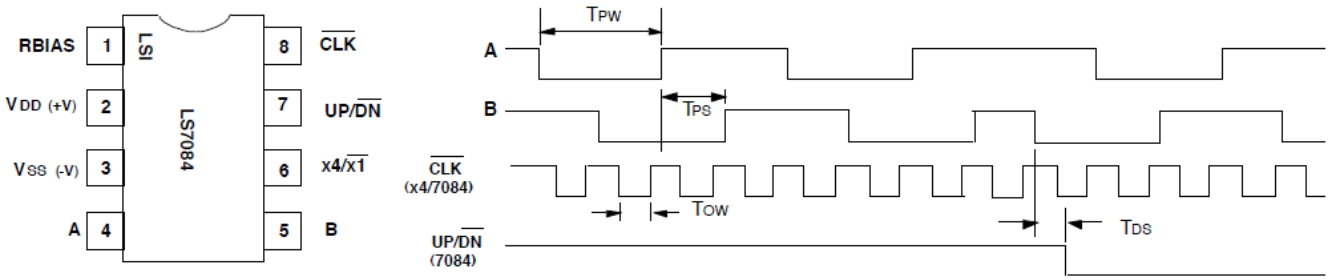
Inkrementalni enkoderi, ovisno o zakretu motora, daju na svojem izlazu niz impulsa. Broj impulsa koje enkoder generira, za jedan puni zakret vratila motora, zove se rezolucija enkodera. Inkrementalni enkoderi u suštini mogu biti izvedeni kao optički ili kao magnetski. Optički generiraju impulse na temelju presjecanja svjetlosnog snopa između izvora svjetlosti i fotoprijemnika, dok se kod magnetskih impulsi generiraju promjenom magnetskog polja, koje uzrokuje okidanje Hall-ovog senzora. Optički enkoderi u pravilu imaju mnogo veću rezoluciju od magnetskih. Kod optičkih enkodera je problem implementacija u mala kućišta, stoga se obično kod malih motora implementiraju magnetski inkrementalni enkoderi. Shematski prikaz magnetskog enkodera dan je na slici 12. Kako bi se mogao dokučiti smjer vrtnje motora, inkrementalni enkoderi obično imaju dva kanala koji su međusobno razmaknuti za određeni kut. Smjer vrtnje određuje se na temelju faznog pomaka između dvaju



Slika 12. Dvokanalni inkrementalni magnetski enkoder

signala. Ukoliko jedan prethodi drugoma, znači da se motor vrti u jednom smjeru, a u slučaju da drugi prethodi prvome, onda se motor vrti u drugom smjeru.

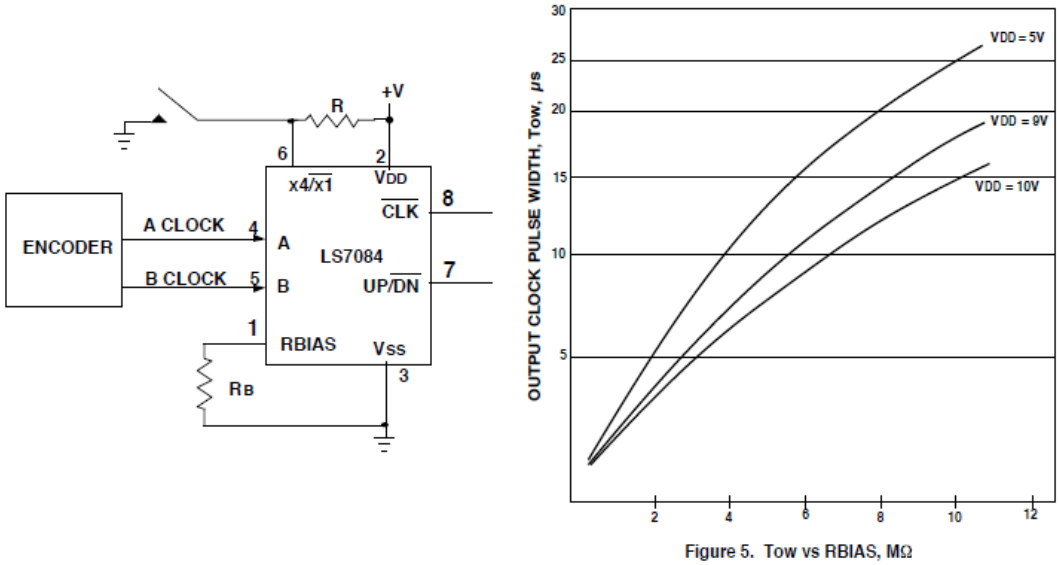
Inkrementalni enkoderi korišteni u ovom radu imaju rezoluciju od 14 impulsa po okretaju, dakle ima dva kanala sa po sedam impulsa po okretaju motora. Kako je to vrlo niska rezolucija, postoje elektronički sklopovi koji multipliciraju broj impulsa pristiglih s enkodera. Ideja koja stoji iza toga je da se broje impulsi na padajući i rastući brid jednog impulsa, čime se udvostručuje broj impulsa po kanalu, odnosno rezolucija po kanalu. Korišteni sklop u ovom konkretnom slučaju je LS7084 koji ima funkciju multiplikacije impulsa te davanja informacije o smjeru vrtnje motora na temelju prethođenja impulsa A i B kanala.



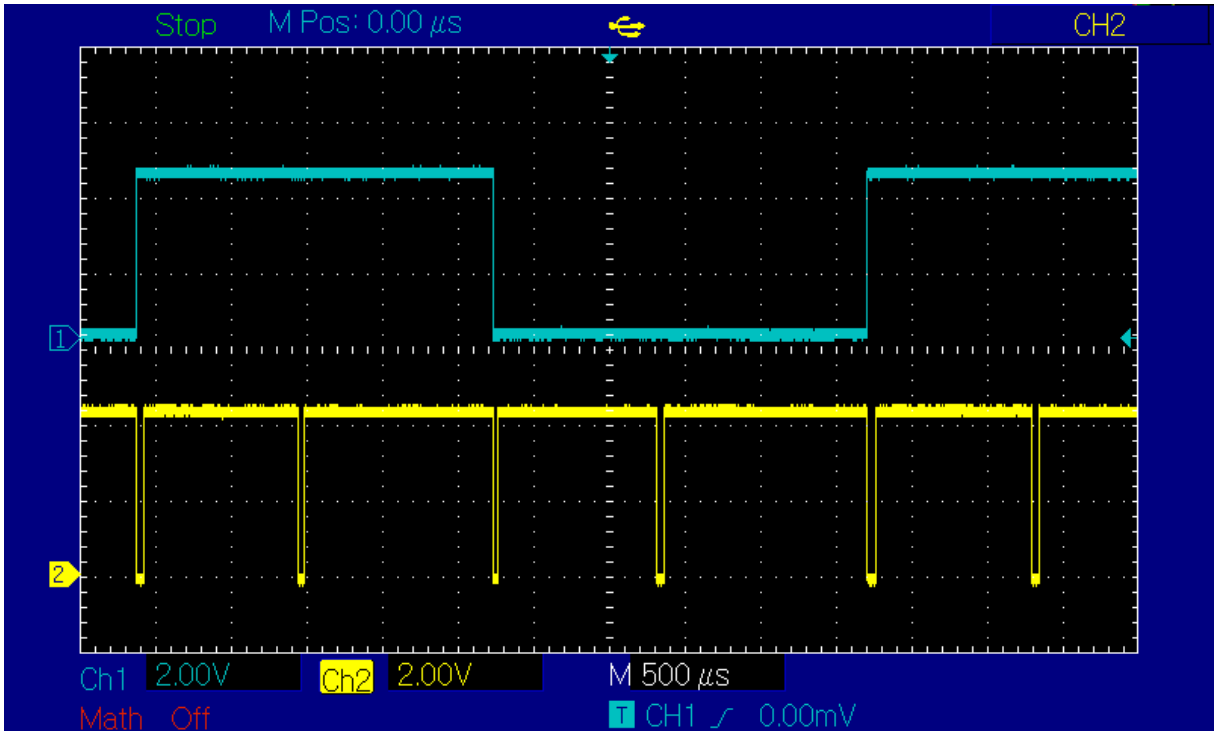
Slika 13. LS7084 raspored pinova i multiplikacija impulsa [7]

Pin označen s CLK daje na izlazu multiplicirani broj impulsa, dok pin UP/DN daje informaciju o smjeru vrtnje motora, kao što se može vidjeti sa slike 13. Kod LS7084 potrebno je odrediti širinu multipliciranih impulsa T_{OW}, koja je kotirana na slici 13. Naime, ukoliko širina impulsa bude premalena (ispod μs) može se dogoditi da mikrokontroler ne registira impuls. Širina impulsa se odabire izborom prave vrijednosti otpora R_B sa slike 14. Kako korišteni nekoder ima vrlo malu rezoluciju, onda će i potrebna širina impulsa biti veća, stoga se odabire veći otpor prema dijagramu na slici 14. Odabrani otpor je R_B= 56 MΩ(za VDD=5V).

Na slici 15. dani je prikaz snimljen osciloskopom, gdje su na kanalu jedan (plavo) impulsi s A kanala enkodera, a na kanalu dva (žuto) multiplicirani niz impulsa sa pina CLK. Sa slike 15. se može vidjeti i širina impulsa T_{OW}.



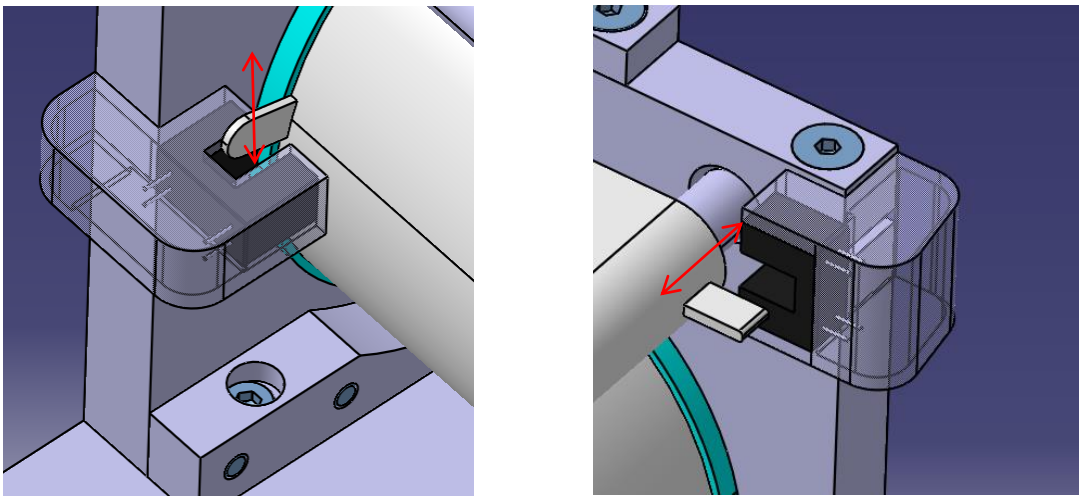
Slika 14. Način spajanja LS7084 i određivanje širine impulsa [7]



Slika 15. Prikaz širine multipliciranih impulsa oscilokopom

Jedan od velikih nedostataka inkrementalnih enkodera je problem inicijalizacije početne pozicije. Naime, kako oni funkcioniraju na temelju generiranja impulsa, informacija o poziciji sadržana je u broju impulsa. Problem koji se javlja ovdje je nepoznata referentna točka, odnosno ne zna se od kuda se broje impulsi. Taj problem se javlja samo kod pokretanja sustava, odnosno kod uključivanja, stoga se može riješiti početnom inicijalizacijom pozicija.

Inicijalizacija početne pozicije izvodi se tako da se objekt upravljanja dovede u točno određeni, poznati položaj, te se od toga položaja počinju brojati impulsi. Nakon što objekt upravljanja prekine snop svjetlosti između optičkih vrata, on je u početnom položaju. Optička vrata načinjena su od izvora svjetlosti, realiziran kao infracrvena svjetleća dioda LED (eng. **L**ight **E**mitting **D**iode) te fotodiode, koja u ovisnosti je li snop između prekinut ili nije, vodi ili ne vodi struju. Samim time se optička vrata ponašaju kao prekidač. Smještaj optičkih vrata prikazan je na slici 16.



Slika 16. Smještaj optičkih vrata

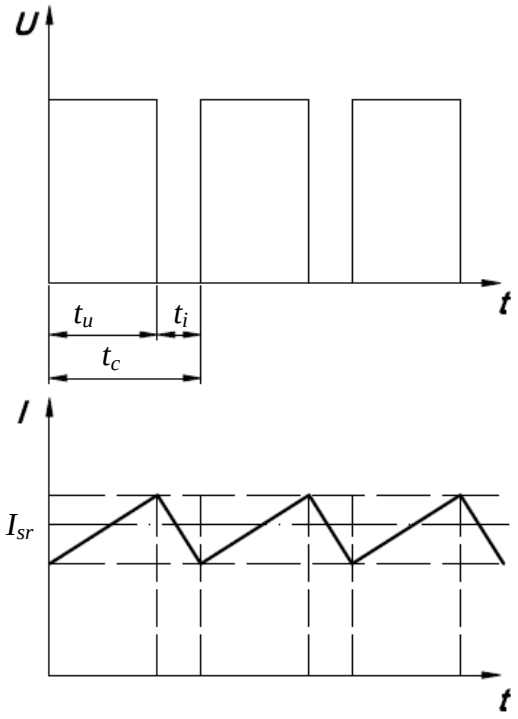
Prikaz inicijalizacije bubnja dan je na slici 16. lijevo, dok je inicijalizacija pomičnog klizača prikazana na slici 16. desno. Jedan od nedostataka optičkih vrata je njihova osjetljivost na okolinu, konkretno na čestice prašine. Naime, može se dogoditi da se s vremenom čestice prašine natalože između izvora svjetlosti i prijemnika, te samim time blokiraju prolaz snopu svjetlosti, čime se dobiva pogrešna informacija. Kako se ovdje radi o maketi rotacijskog crtala, koje neće biti izloženo prašini, onda su optička vrata opravdani izbor. Kod primjene u ozbiljnijim uvjetima, najčešće se koriste mehanički prekidači ili Hallovi senzori.

3.2. Upravljanje motorom

Korišteni motori opisani su u konstrukcijskom dijelu te se u tablici 1. nalaze podaci dotičnog motora. U ovom poglavlju bit će predstavljen način upravljanja motorima s energetskog aspekta. Kod istosmjernog motora s nezavisnom uzbudom brzina vrtnje proporcionalno ovisi o naponu na armaturi motora, samim time se brzinom vrtnje može upravljati naponom armature. Upravljanje brzinom vrtnje promjenom napona armature u suštini se može izvesti pomoću [8]:

1.) istosmjernog generatora
2.) elektroničkog sklopa napajanog iz istosmjerne mreže (čoper)
3.) elektroničkog sklopa napajanog iz izmjenične mreže

Najčešće je korišten drugi način upravljanja, gdje se brzinom upravlja brzim uključivanjem i isključivanjem armature motora na konstantan izvor istosmjernog napajanja. Ovisno o tome koliko je armatura motora dugo uključena, odnosno isključena, mijenjat će se napon na njoj, odnosno brzine vrtnje samoga motora. Takvo upravljanje naponom, u konačnici brzinom vrtnje elektromotora, naziva se pulsno širinska modulacija (eng. **P**ulse **W**idth **M**odulation), gdje se širinom impulsa mijenja srednja vrijednost napona. Kako kod motora postoji omski otpor i induktivitet, struja će kasniti za naponom, te će samim time oscilirati, kao što je i prikazano na slici 17.



Slika 17. Načelo PWM modulacije [8]

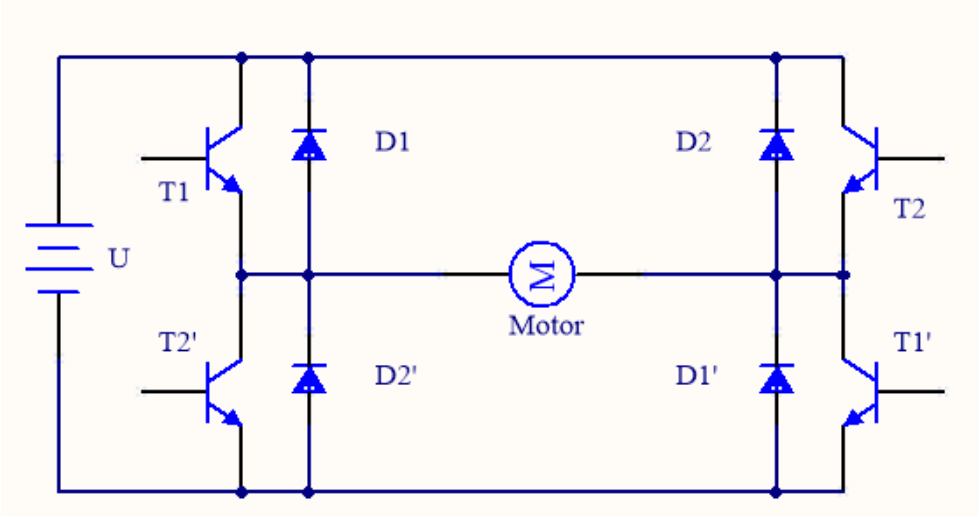
Oscilacije u struji kod elektromotora imaju za posljedicu oscilaciju i momenta motora što dovodi kod neravnomjerne vrtnje motora. No, kako je obično frekvencija isključivanja i uključivanja vrlo velika (reda veličine nekoliko kHz), onda su te pojave neprimjetne, pogotovo ako se radi o sustavu s velikom inercijom. Srednja vrijednost napona može se odrediti prema izrazu (6):

$$U_{sr} = \frac{1}{t_c}\int_0^{t_c} u(t)dt = \frac{U_b t_u}{t_c} = dU_b \tag{6}$$

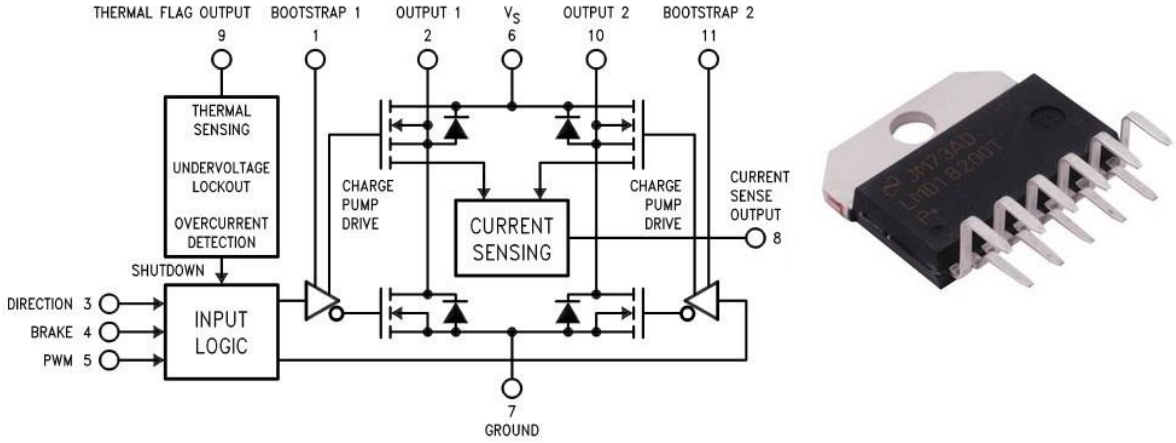
gdje je t_c vrijeme ciklusa sa slike 17, t_u vrijeme uključenja sa slike 17, U_b napon baterije koja napaja strujni krug. Omjer vremena uključenja t_u i vremena ciklusa t_c označava se s d, a naziva se faktor popunjenosti (eng. duty cycle) [9], a izražava se u postocima. Tako vrijednost d=50% daje pola napona napajanja strujnog kruga U_b. Važno je napomenuti da postoje dvije definicije PWM-a. Kod jedne varijante se koriste i pozitivni i negativni pravokutni impulsi te 50% faktora popunjenosti odgovara vrijednosti srednjeg napona od 0V, vrijednost faktora popunjenosti manje od 50% rezultirat će vrtnjom motora u jednu stranu, dok će za vrijednosti veće od 50% rezultirati vrtnjom motora u drugu stranu. Samim time izbjegnuta je potreba za dodatnim signalom koji nosi informaciju za potrebni smjer vrtnje motora. Druga varijanta, koja je ujedno i korištena u ovom radu, gdje je korišten samo jedan polaritet impulsa kako je prikazano na slici 17. Kod te varijante potreban je još jedan signal koji kazuje željeni smjer vrtnje.

Kako se PWM signal sa mikrokontrolera ne može direktno spojiti na motor,već ga treba pojačati, potreban je sklop koji će pojačati taj signal na razinu koja je potrebna motoru. Takav sklop zove se tranzistorski H-most, a njegova princepijalna shema dana je na slici 18. Na baze tranzistora dowodi se PWM signal s mikrokontrolera. Da bi potekla struja kroz motor jedan par tranzistora T_1 i T_1' ili T_2 i T_2' mora biti propusno polariziran. Ovisno o tome koji je par tranzistora propusno polariziran, na armaturu motora dovodi se određeni polaritet napona, a time se mijenja i smjer vrtnje motora.

Postoje izvedbe integriranih elektroničkih sklopova, koji u sebi imaju H-most. Jedna takva izvedba, korištena u ovome radu, je LMD18200T koji na svojem ulazu prima unipolarnu PWM modulaciju i smjer vrtnje.



Slika 18. Tranzistorski H-most



Slika 19. LMD18200T shema (lijevo), izvedba (desno)

Sa slike 19. se može vidjeti izvedba kućišta LMD18200T te principijalna shema istoga.

3.3. Mikrokontroler

Mikrokontroler je digitalni automat spospoban za izvršavanje računskih operacija na osnovu programa pohranjenog u memoriji [10]. Mikrokontroler sekvencijalno izvršava instrukcije koje u konačnici rezultiraju izvršavanjem nekog zadatka. Sastoji se od mikroprocesora koji je vođen vremenskim taktom i izvršava aritmetičke i logičke operacije, RAM (eng. **R**andom **A**ccess **M**emory) memorije koja služi za pohranjivanje rezlutata operacija koje izvodi mikroprocesor, ROM (eng. **R**ead **O**nly **M**emory) memorije u kojoj su pohranjene instrukcije

za mikroprocesor, ulazno-izlaznih registara koji služe za komunikaciju s vanjskim uređajima te podatkovne sabirnice putem koje pojedini moduli komuniciraju i razmjenjuju podatke.

U današnje vrijeme postoji mnogo raznih mikrokontrolera, stoga je kod projektiranja upravljačkog sustava potrebno odabrati adekvatni mikrokontroler. Za potrebe upravljanja rotacijskim crtalom odabran je Atmelov Atmega2560 mirkokontroler, dobro poznate RISC (eng. **R**educed **I**nstrucion **S**et **C**omputer) strukture. Odabran je zbog resursa kojima raspolaže, a to su[11] :

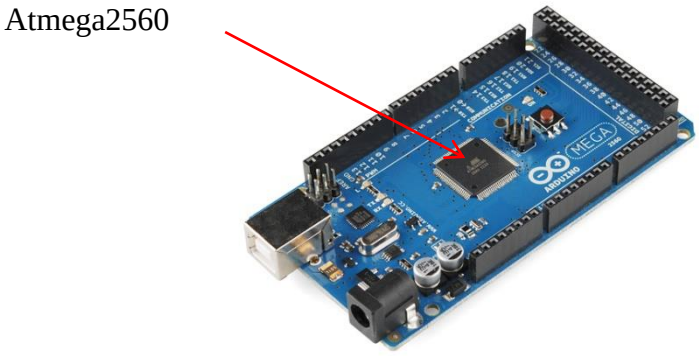
1.) FLASH memorija od 64 Kbyte-a
2.) dva 8-bitna brojača
3.) dva 16-bitna brojača
4.) šest PWM izlaznih kanala
5.) dovoljan broj ulazno/izlaznih pinova
6.) USART komunikacija

Uz dovoljne resurse, potrebna je i tehnička dokumentacija mikrokontrolera koja je lako dostupna i pregledna. Velika FLASH memorija potrebna je kako bi se spremio program koji se treba izvoditi te rezultati matematičkih operacija u aritmetici pomičnog zareza (eng. floating point). Dva 8-bitna brojača potrebna su kako bi brojili impulse pristigle s enkodera dvaju motora. Stanje dvaju 8-bitnih brojača očitava se svakih nekoliko ms, te se postavlja na 0. To je nužno raditi jer 8-bita nisu dovoljna da u njih stane broj impulsa koji bi se generirali kada bi npr. pomični klizač prevalio cijelu dužinu ili bubanj napravio puni krug. Jedan 16-bitni brojač ima ulogu generiranja PWM signala, dok drugi ima ulogu ostvarivanja prekidne rutine svakih nekoliko ms, kako bi se očitala pozicija s enkodera, izvršila interpolacija i regulacija, što je zapravo slično izvršavanju u realnom vremenu. Potrebna su dva digitalna ulaza s prekidnom rutinom, koji će se koristiti kod incijalizacije početnog položaja, dva obična digitalna ulaza koje će davati informaciju o smjeru vrtnje motora te jedan digitalni izlaz, koji će uključivati ili isključivati elektromagnet.

Na slici 20. prikazan je Atmega2560 izveden na Arduino platformi. Razlozi korištenja su mogućnost nabave po vrlo niskoj cijeni, dok istovremeno nudi mogućnost brzog spajanja pinova, na sebi sadrži komponente koje omogućavaju mikrokontroleru samostalni rad te komunikaciju preko serijske veze.

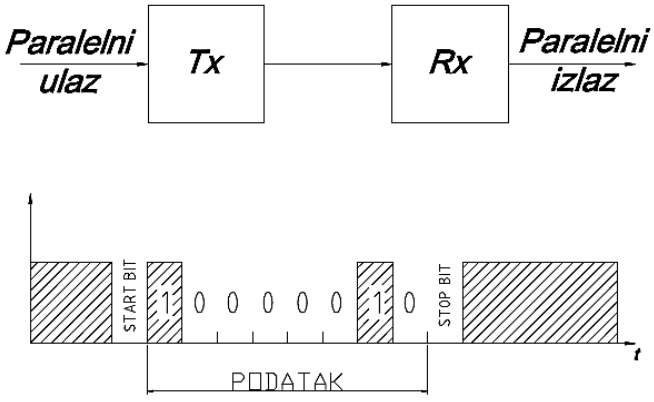
Mikrokontroler s računalom komunicira preko UART (eng. **U**niversial **A**synchronous **R**eceiver-**T**ransmitter) veze. Kod UART komunikacije željeni podatak koji se želi poslati

dijeli se na pakete podataka veličine 8 bita odnosno 1 Byte, što odgovara jednom znaku prema ASCII tablici. Predajnik šalje niz impulsa koji predstavljaju 8-bitni zapis znaka, s pomoćnim bitovima kako je prikazano na slici 21., dok se na strani prijemnika nalazi SHIFT registar u koji se upisuju bitovi iz podatka. Tu je važno da oba uređaja koji sudjeluju u komunikaciji šalju i primaju podatke na istoj brzini te isti format podataka, kako nebi došlo do krive pretvorbe podataka.



Slika 20. Atmega 2560, Arduino MEGA razvojna pločica

Sa slike 21. može se vidjeti shematski prikaz UART komunikacije, te primjer signala koji se prenosi. Konkretno je dan primjer za prenošenje decimalnog broja „65“ koji prema ASCII tablici odgovara slovu „A“. Može se vidjeti da prijenos svakog podatka počinje sa START bitom te završava sa STOP bitom. Redoslijed bitova podataka ovisi o UART standardu, kod primjera sa slike 21. podatak počinje sa njamanje značajnim bitom, jer je $(65)_{10}=(01000001)_2$



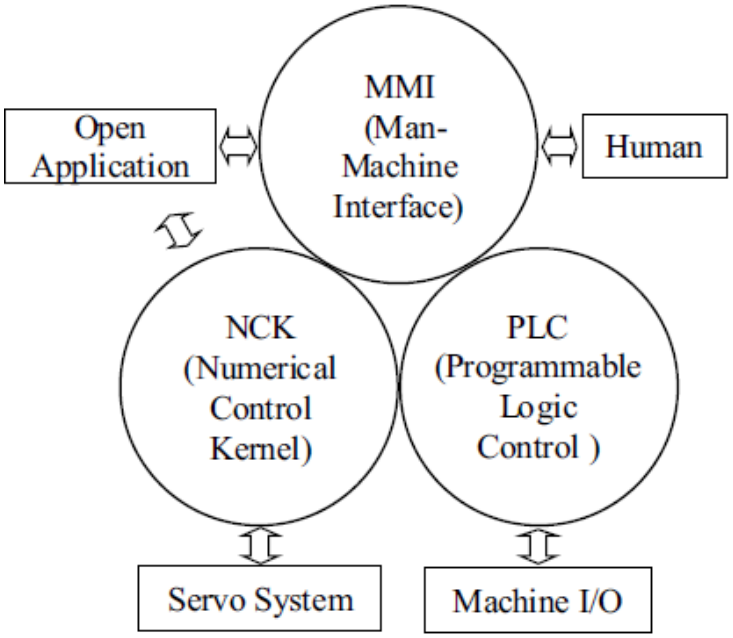
Slika 21. UART komunikacija

4. UPRAVLJANJE CRTALOM

Već je ranije spomenuto da rotacijsko crtalo, u suštini, nije ništa drugo nego pojednostavljeni CNC stroj, koji vodi alat (pisaći element) po određenoj trajektoriji. Da bi se ostvarilo pozicioniranje pisaćeg elementa u određeni položaj ili da bi se pisaći element vodio po određenoj putanji, potrebno je davati motorima određenu referencu koja će dovesti do željenog gibanja. U ovom poglavlju opisan je algoritam koji izvršava mikrokontroler, koji ima ulogu numeričke jezgre (eng. Numerical Control Kernel.).

4.1. Struktura CNC sustava

Moderni CNC sustavi sastoje se od MMI (eng. **M**an **M**achine **I**nterface) koje ima ulogu, kao što i samo ime govori, sučelja između čovjeka i stroja. MMI izvšava naredbe koje unosi čovjek ručno, prikazuje trenutni status stroja te omogućuje promjene part programa. U samome središtu CNC sustava nalazi se numerička jezgra ili NCK (eng. **N**umerical **C**ontrol **K**ernel), koja interpretira part program (G kôd), izvodi planiranje trajektorije i interpolaciju trajektorije te izvodi regulaciju pozicije pojedinih osi. Posljednji dio CNC sustava je PLC (eng. **P**rogrammable **L**ogic **C**ontrol) ili programibilno logičko upravljanje, koje sekvencijalno upravlja izmjenom alata, brzinom motorvretena i slično.

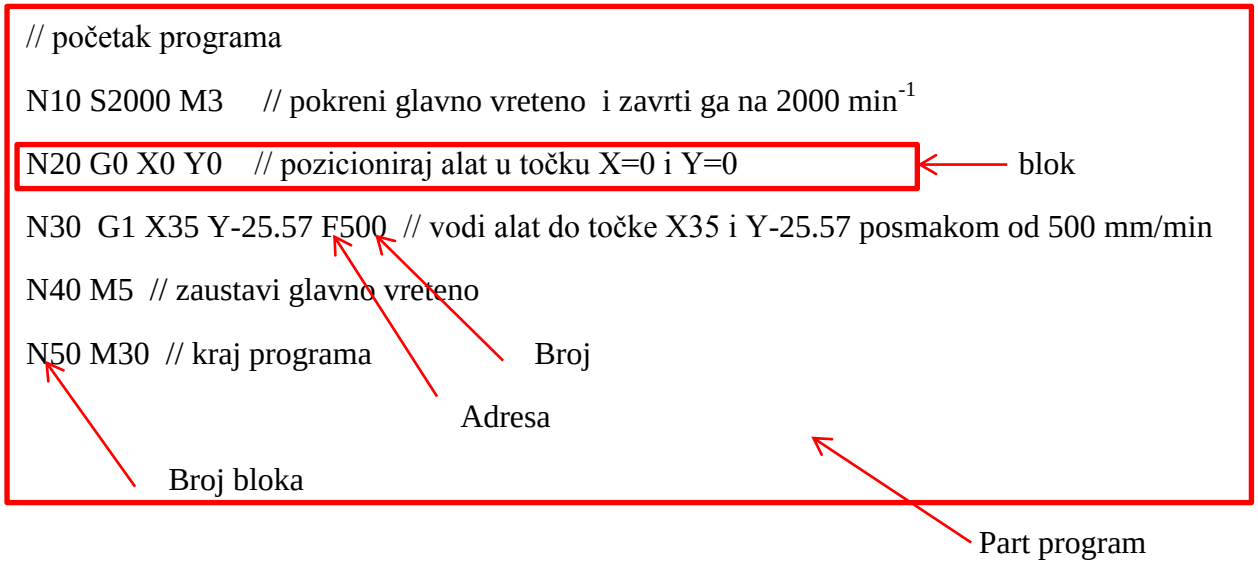


Slika 22. Struktura CNC sustava[6]

Uglavnom, može se reći da PLC upravlja perifernim funkcijama stroja, dok se za glavne funkcije stroja kao što je regulacija pozicije pojedinih osi, brine NCK. Na slici 22. prikazan je blokovski prikaz strukture modernog CNC sustava. Rotacijsko crtalo se također sastoji od segmenata koji su prethodno navedeni, uz razliku da su neki segmenti pojednostavljeni, kao primjerice MMI. MMI u suštini nije ništa drugo već terminal na računalu koji šalje putem UART serijske veze G kôd. PLC funkcija kod rotacijskog crtala izvodi uključivanje i isključivanje elektromagneta te provjeru stanja krajnjih senzora položaja. Naredba za uključivanje ili isključivanje dolazi iz NCK, odnosno iz interpretera koji interpretira G kôd. U nastavku će biti više riječi o implementaciji CNC sustava na rotacijsko crtalo.

4.2. G kôd

G kôd je vrsta jezika kojim se daju naredbe alatnom stroju. Naredbe mogu biti raznovrsne, kao npr. gibanje od točke do točke određenom brzinom, vrtnja glanog vretena određenom brzinom, uključivanje ili isključivanje podmazivanja i sl. Format programa pisanog G kôdom dan je ispod:



Part program se sastoji od niza blokova, gdje se svaki blok sastoji od nekoliko riječi ili jedne riječi. Jedna riječ sastoji se od adrese i od broja. Adresa može biti jedno od slova abdecede (A-Z) ili kombinacija slova. Pripadajući broj poslije adrese daje informaciju koja je potrebna da se izvrši željena radnja s adresom [6]. U tablici 3. dane su riječi odnosno adrese G kôda koje su korištene u ovome radu, koje implementirani interpreter može razumijeti.

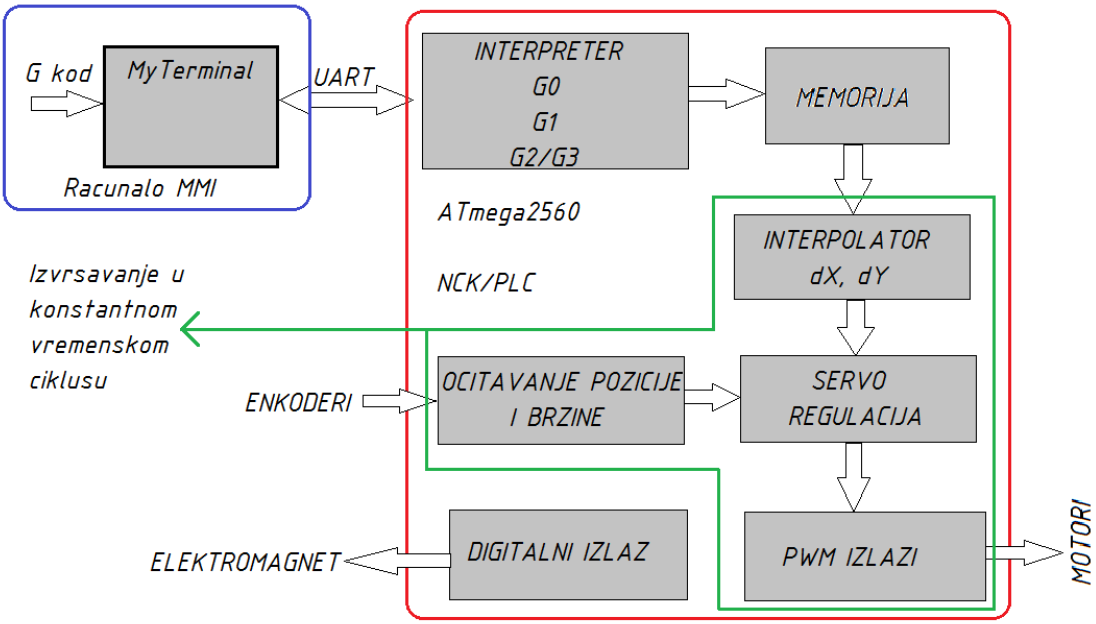
Tablica 3. Korištene riječi G kôda

Riječ (eng. Word)	Opis naredbe
G0	Brzo pozicioniranje alata, uz maksimalnu brzinu
G1	Linearna interpolacija uz određenu brzinu gibanja alata
G2	Kružna interpolacija u smjeru kazaljke na satu
G3	Kružna interpolacija u smjeru suprotnom od kazaljke na satu
Fn	Zadavanje posmaka kod linearne i kružne interpolacije
Xn	Koordinata krajnje točke po X osi (n je broj)
Ym	Koordinata krajnje točke po Y osi (m je broj)
M3	Uključivanje elektromagneta
M5	Isključivanje elektromagneta
M30	Kraj programa
In	Udaljenost centra kružnice od početne točke po X osi
Jm	Udaljenost centra kružnice od početne točke po Y osi

Valja napomenuti da je ova varijanta G kôda prilagođena izvedbi upravljanja rotacijskog crtala, kako bi se interpreter mogao lakše realizirati. Kod naredbe „G0“, motori će se pozicionirati u točki koja je dana naredbama Xn i Ym, gdje će se motori gibati brzinom koja je ograničena softwerski u mikrokontroleru, izračunatom u nastavku rada. Kod linearne interpolacije, odnosno naredbe „G1“, motor će se od početne do krajnje točke gibati brzinom koja je definirana u istom bloku ili u bloku prije, pomoću naredbe Fn. Broj *n* odgovara brzini pojedine osi u mm/min. Naredbe M3 i M5, koje se koriste kod konvencionalnih CNC sustava, uza sebe moraju još imati i informaciju o brzini vrtnje glavnog vretena. No, kako na rotacijskom crtalu nema glavnoga vretena, onda nema potrebe za zadavanjem reference brzine vrtnje, već su dovoljne naredbe M3/M5. Kod kružne interpolacije motori će se pozicionirati u točkama koje su zadane adresama X i Y, gdje će putanja do tih počaka biti kružnica čiji je centar udaljen od početne točke brojevima uz adrese I i J. Smjer putanje bit će određen ovisno o broju uz adresu G, kako je prikazno u tablici 3.

4.3. Struktura upravljanja rotacijskim crtalom

Na slici 23. dani je prikaz strukture upravljanja rotacijskim crtalom. Može se vidjeti da računalo koje komunicira s mikrokontrolerom preko UART veze ima ulogu sučelja između stroja i čovjeka, odnosno MMI. Mikrokontroler ima ulogu numeričke jezgre (NCK) te rukovanja ulazno izlaznim signalima, odnosno PLC funkciju. Moduli koji se nalaze unutar zelene konture na slici 23 izvršavaju se svaki put u jednakim vremenskim razmacima, odnosno u konstantnom vremenskom ciklusu. Da to nije slučaj, ne bi se mogla rekonstruirati brzina iz broja impulsa, odnosno pozicije.

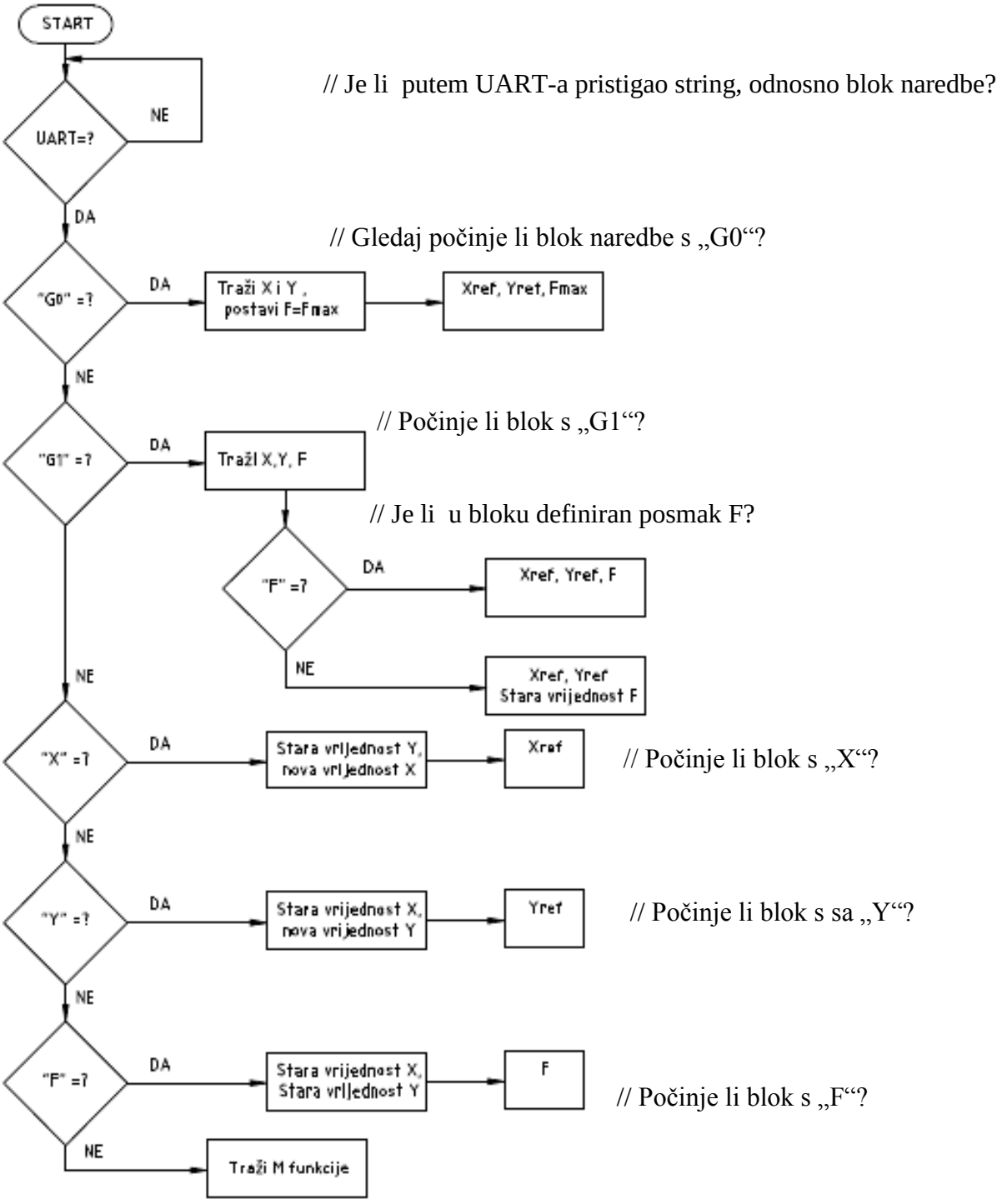


Slika 23. Struktura upravljanja rotacijskim crtalom

Komunikacija između računala i mikrokontrolera obavlja se kada se izvrše sve operacije koje se izvršavaju u konstantnom vremenskom ciklusu. Nakon toga slijedi interpreter koji rezultate sprema u međuspremnik, odnosno memoriju. Rezultati interpretacije su reference pozicije i brzine te način gibanja, odnosno interpolacije. U nastavku će biti opisan svaki modul za sebe te način na koji je implementiran u C jeziku za mikrokontrolere uz dijagrame toka.

4.4. Implementacija interpretera

Već je ranije površno spomenuta uloga interpretera u CNC sustavu, a u ovom potpoglavlju je interpreter obrađen detaljnije te je predstavljen algoritam implementiran u mikrokontroleru. Algoritam interpretacije priakzan je na slici 24.



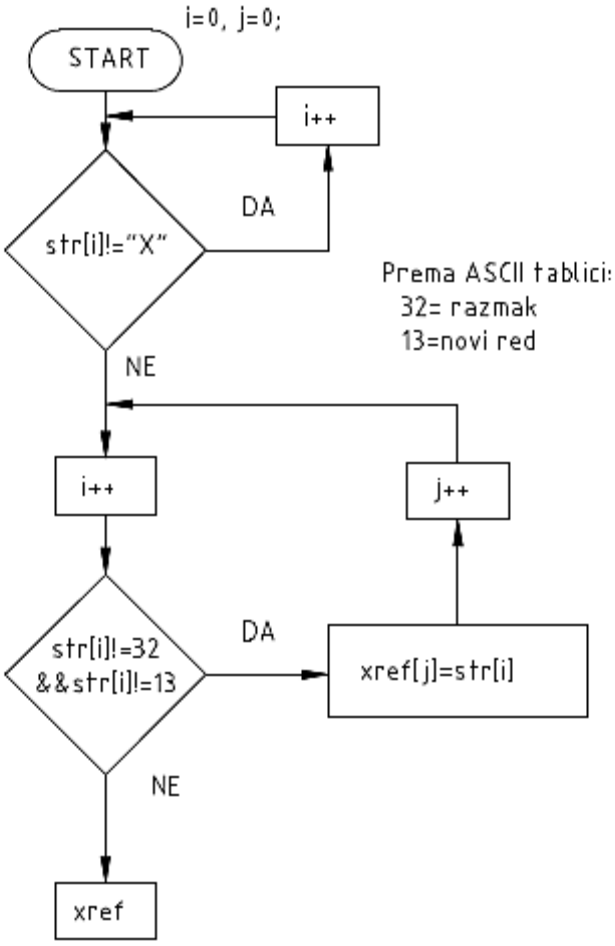
Slika 24. Blok dijagram interpretera

Ukoliko je putem serijske veze stigao niz znakova te je spremljen u memoriju, taj niz znakova postaje rečenica, odnosno string. Algoritam gleda odgovara li početni znak rečenice ili stringa određenom slovu, te na temelju toga dalje izvršava zadatak. Najprije se provjerava počinje li string s „G0“. Ukoliko počinje, algoritam dalje traži brojeve koji stoje uz adrese „X“ i „Y“, te pretvara te brojeve iz niza znakova u decimalni broj, pomoću funkcije atof(char *str), koja je sastavni dio biblioteke „Standard Library Functions“(stdlib.h) unutar CodeVisionAVR IDE. Pomoću CodeVisionAVR-a se piše program za mikrokontroler u C jeziku, stvoren za Atmelove mikrokontrolere. Nakon što se znakovi pretvore u decimalni broj, mikroprocesor s njima može raditi aritmetičko-logičke operacije.

Ukoliko pristigli blok naredbi počinje sa „G1“, onda se kao i u prethodnom primjeru traže brojevi uz adrese „X“ i „Y“ te se pretvaraju u decimalni broj. Tu se sad može dogoditi da se unutar bloka naredbe ne definira vrijednost posmaka zbog toga jer je već prethodno definiran. Stoga je potrebno provjeriti da li u bloku postoji adresa „F“. Ukoliko ne postoji, referentna vrijednost posmaka ostaje nepromijenjana, odnosno ostaje ona vrijednost koja je definirana u jednom od prošlih blokova. Ukoliko ne počinje sa G1, onda se gleda počinje li blok sa adresom „X“. Ukoliko počinje, znak iza adrese se pretvara u decimalni broj i dobiva se referentna vrijednost za X, dok Y i posmak F ostaju nepromijenjeni. Ukoliko ne počinje s adresom „X“, onda se gleda počinje li sa „Y“, te se provodi ista procedura kao i u prethodnom primjeru. Zadnji slučaj koji se može dogoditi je da se u bloku definira samo posmak. Ukoliko se to dogodi, procedura je ista kao i za prethodna dva primjera.

Algoritam traženja brojeva iza adrese temelji se na ispitivanju pojedninih znakova. Na slici 25. dan je prikaz dijagrama toka, za traženje znakova iza adrese „X“. Algoritam je isti za sve ostale adrese. U početku se inicijaliziraju početne varijable *i* i *j* koje služe za brojanje pozicije znaka unutar niza znakova (stringa str[]). Prvo se u nultom koraku provjerava je li za vrijednost i=0, odnosno početni znak odgovara adresi „X“.Ukoliko ne odgovara, znači da je str[0] različit od „X“, stoga se i povećava za 1. Nakon što se naiđe na znak koji je jednak „X“, *i* se poveća za jedan te se gleda je li znak u stringu na poziciji *i* različit od razmaka ili novog reda (*carriage return*). Ukoliko je, znači da se radi o broju, stoga se taj znak sprema u string xref[j], dakle na prvo mjesto, budući da je u prvom koraku *j*=0. Algoritam zapisuje brojeve u

xref[j], sve dok se jedan od dva uvjeta ne promijeni, odnosno ukoliko ne dođe do razmaka ili novoga reda.



Slika 25. Dijagram toka za traženje brojeva iza „X“ adrese

Ukoliko algoritam dođe do znaka razmaka ili novog reda, vraća string xref[] koji se kasnije pretvara u decimalni broj pomoću funkcije atof(*xstr), već od prije spomenute. Algoritam za traženje M funkcija, također, radi na istom principu, uz razliku da ne vraća decimalni broj, već cijeli broj koji dalje sugerira što se treba poduzeti. Primjerice, da se jedan od izlaza postavi u visoko ili nisko stanje.

Za kružnu interpolaciju, također , vrijede isti algoritmi izdvajanja referenci. Ovisno o kojoj interpolaciji se radi, reference se šalju ili linearnom interpolatoru ili kružnom interpolatoru.

4.5. Očitavanje pozicije i brzine

Broj impulsa pristiglih s enkodera servomotora predstavljaja pomak istoga, koji se u ovisnosti od prijenosnog omjera i referentne točke može prikazati kao pozicija objekta upravljanja. Jedan impuls predstavlja inkrement pomaka, odnosno najmanji pomak koji se zadanim servomotorem može ostvariti. Često se naziva jedinični pomak ili BLU (eng. **B**asic **L**ength **U**nit). Broj pristiglih impulsa u vremenu predstavlja brzinu vrtnje servomotora.

Impulse koji pristižu s inkrementalnog enkodera obrađuje elektronički sklop LS7084 koji multiplicira broj impulsa i daje informaciju o smjeru vrtnje servomotora. Impulsi koji dolaze s LS7084 zbrajaju se u brojaču mikrokontrolera koji može brojiti najviše 255 impulsa pristiglih s LS7084. Tu se javlja problem, budući da 255 impulsa otprilike predstavlja pomak klizača od cca 15 mm, što ni u kojem slučaju nije dovoljno. Iz tog razloga se broj impulsa iz brojača mikrokontrolora periodički uzima, te se prilikom očitavanja brojač postavlja na 0. Samim time potrebno je odrediti frekvenciju, odnosno vrijeme uzorkovanja za koje brojač neće premašiti 255. Vrijeme uzorkovanja će se odrediti pomoću podataka iz tablice 1, poznate rezolucije enkodera uvećane za 2 zbog elektroničkog sklopa LS7084 te brzine vrtnje servomotora u praznom hodu koja iznosi n_0=6500 min^{-1}. Najveći broj impulsa u jednoj sekundi iznosi:

$$f_{1s} = \frac{n_0}{60} 2N = \frac{6500}{60} 2 \cdot 14 = 3033.3\ Hz \quad (7)$$

Kako je vrijeme između dva impulsa jedanko recipročnoj vrijednosti frekvencije, odnosno broju impulsa u određenom vremenu, onda se može pisati da je vrijme koje je potrebno da brojač izbroji 255 impulsa jednako:

$$T_{INT} = \frac{255}{f_{1s}} = 0.084\ s \quad (8)$$

Radi sigurnosti odabire se T_{INT}=80ms. T_{INT} je vremenski ciklus u kojem će se izvršavati prekidna rutina za očitavanje impulsa, računanje brzine, interpolacije te regulacije dviju osi. Ukoliko bi se vrijeme uzorkovanja T_{INT} još više smanjilo, došlo bi do problema kod određivanja brzine motora, stoga je odabrano da je T_{INT}=80ms. Povećavanjem vremena

uzorkovanja, regulacija pozicije i brzine bila bi neizvediva, budući da se ne bi moglo kvalitetno mjeriti poziciju i brzinu.

U drugom poglavlju zadan je uvjet da jedinični pomak, odnosno rezolucija sustava ne smije biti manji od 0.5 mm. Ta vrijednost okvirno je odabrana kako bi se dimenzionirali elementi za prijenos snage i gibanja. Kako bi se odredila točna pozicija objekta upravljanja, potrebno je izračunati jedinični pomak za svaku os posebno, uzevši u obzir odabrane elemente za prijenos snage i gibanja. Tako je za x os jedinični pomak:

$$\Delta x = \frac{\pi D_1}{2Ni} = \frac{\pi \cdot 12.73}{2 \cdot 14 \cdot 27} = 0.053 \ \frac{mm}{imp} \tag{9}$$

Dok je za bubanj, odnosno Y os:

$$\Delta y = \frac{\pi D_B D_2}{2D_3 Ni} = \frac{\pi \cdot 68 \cdot 12.73}{2 \cdot 38.2 \cdot 14 \cdot 27} = 0.094 \ \frac{mm}{imp} \tag{10}$$

Maksimalne brzine koje osi mogu ostvariti mogu se izračunati prema:

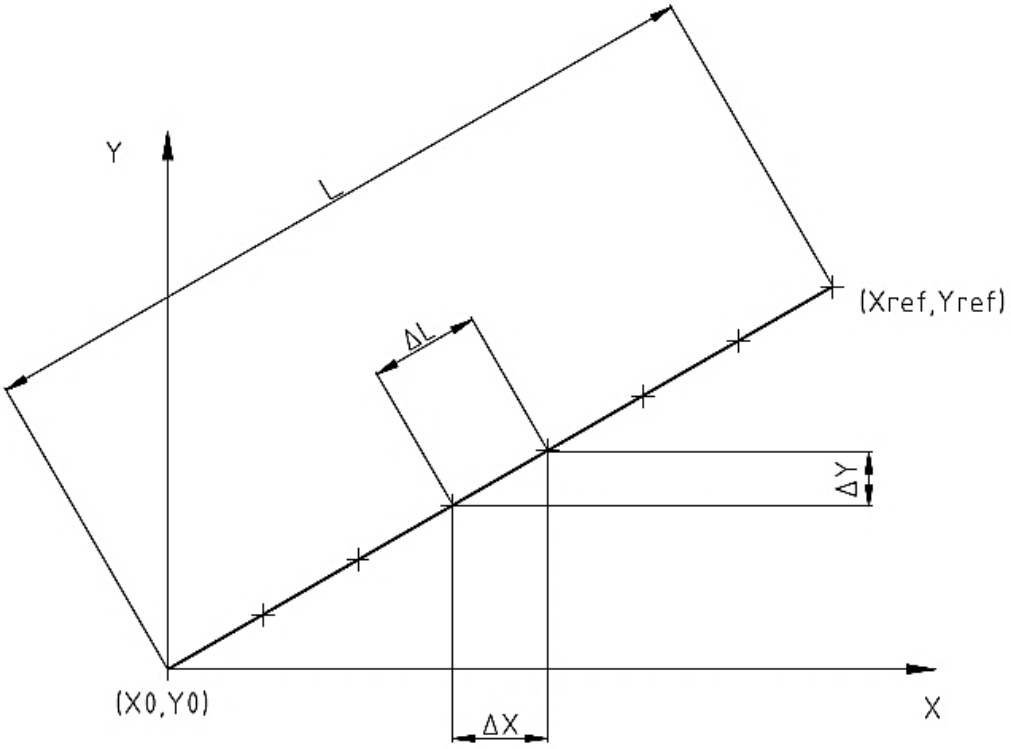
$$v_{xmax} = D_1 \pi n_R = 12.73 \cdot \pi \cdot 195 \cong 7795 \ \text{mm/min} \tag{11}$$

$$v_{ymax} = \frac{D_B D_2}{D_3} \pi n_R = \frac{68 \cdot 12.73}{38.20} \pi \cdot 195 \cong 13875 \ \text{mm/min} \tag{12}$$

Kako je za Y os brzina približno dva puta veća od najveće moguće brzine X osi, potrebno je ograničiti maksimalni mogući posmak na najmanju ostvarivu brzinu. U ovom slučaju, na najveću moguću brzinu gibanja X osi, prema izrazu (11). Ukoliko se brzina ne bi ograničila, moglo bi se dogoditi da se prilikom brzog pozicioniranja („G0“) Y os brže pozicionira od X osi, što je nedopustivo jer se osi trebaju gibati istovremeno, u jednakim vremenskim razmacima. Drugi slučaj bi mogao biti da se u programu traži da je referentna vrijednost posmaka $F_{ref} > v_{xmax}$, što se nebi moglo ostvariti. Stoga se brzina obiju osi ograničava na F_{max}=4500 mm/min.

4.6. Linearni interpolator

U suštini se gibanjem alata (olovke) može upravljati na dva načina. Prvi način je točka-točka (eng. Point- to- Point), gdje se alat pozicionira u određenoj točki, dok kod drugog načina alat prati proizvoljnu krivulju. Važno je napomenuti da se alat treba gibati određenom brzinom duž trajektorije ili od točke do točke. Obično se kod upravljanja točka-točka alat giba najvećom mogućom dopuštenom brzinom te se u G kôdu poziva naredbom „G0“. Samim time se upravljanje točka-točka može izvršiti i bez interpolatora. No, kod praćenja konture određenom brzinom, interpolator je neizbježan. Referencu interpolatoru zadaje interpreter koji tu istu referencu „izvlači“ iz part programa, odnosno G kôda.



Slika 26. Linearna interpolacija [13]

Na slici 26. prikazan je primjer linearne interpolacije. Dakle, alat se treba gibati od početne točke (X_0,Y_0) do krajnje točke koja dolazi od interpretera (X_{ref},Y_{ref}), zadanom brzinom *F*ref. Duljina puta kojeg alat treba prijeći iznosi:

$$L = \sqrt{(X_{REF} - X_0)^2 + (Y_{REF} - Y_0)^2} \tag{13}$$

Sama ideja interpolatora je razdijeliti ukupnu duljinu puta na segmente, u ovisnosti od vremena izvršavanja prekidne rutine T_{INT} iz izraza (8) te referentne brzine, odnosno referentnog posmaka F_{REF}. Iz čega slijedi duljina segmenta puta ΔL:

$$\Delta L = F_{REF} T_{INT} \tag{14}$$

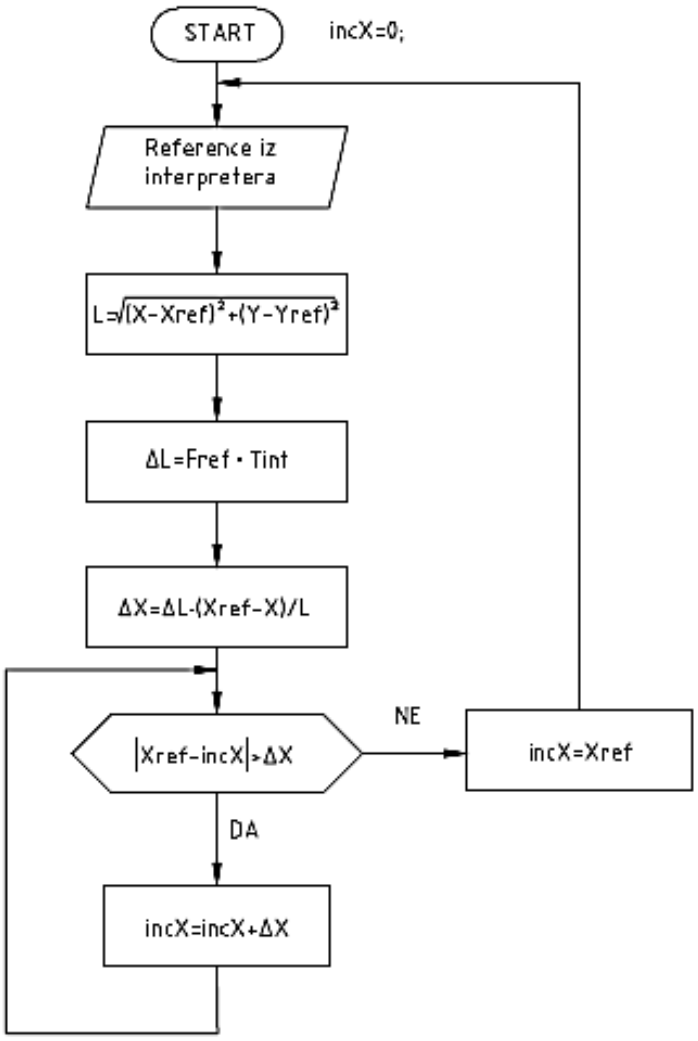
Iz duljine segmenta dalje se računaju duljine segmenata za pojedine osi ΔX te ΔY, kako slijedi:

$$\Delta X = F_{REF} T_{INT} \frac{X_{REF} - X}{L} = (X_{REF} - X) \frac{\Delta L}{L} \tag{15}$$

$$\Delta Y = F_{REF} T_{INT} \frac{Y_{REF} - Y}{L} = (Y_{REF} - Y) \frac{\Delta L}{L} \tag{16}$$

Duljine segmenata za pojedine osi su reference za regulator pozicije, tako da se duljine segmenata u svakom trenutku interpolacije zbrajaju s prošlim vrijednostima, sve dok njihov zbroj ne bude jednak vrijednosti X_{REF} i Y_{REF}. Problem koji se ovdje javlja je taj što kvocijent duljine puta L i duljine segmenta ΔL mora biti cijeli broj. Ukoliko nije, javlja se greška reference, o kojoj treba voditi računa.

Rješenje je dano na slici 27. na dijagramu toka lineranog interpolatora. Dijagram obuhvaća samo X os, dok za Y os vrijedi isto. Nakon što se izračuna segment duljine za X os ΔX , provjerava se je li varijabla *incX*, u koju se zbrajaju segmenti duljine, dostigla vrijednost reference. Ukoliko je razlika između reference X_{REF} i *incX* veća od duljine segmenta ΔX, onda se vrijednost ΔX pribraja varijabli *incX*. Ukoliko je razlika manja, onda se varijabla *incX* postavlja u vrijdnost X_{REF}, čime se rješava gore navedeni problem kvocijenta duljine puta i segmenta jer je konačna vrijednost X_{REF}.

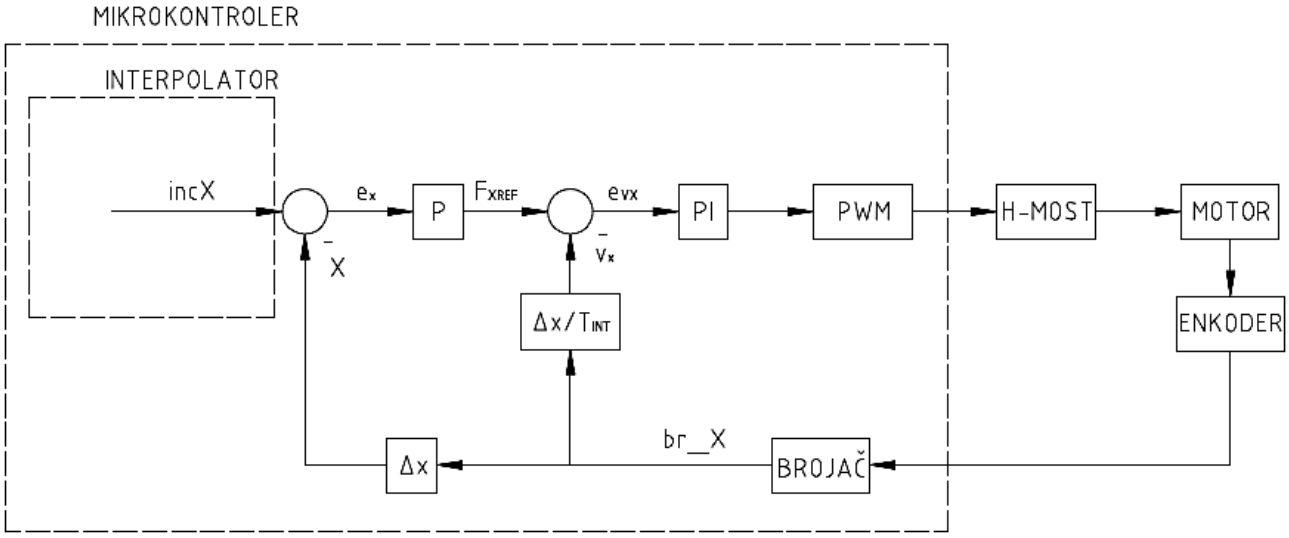


Slika 27. Dijagram toka linearnog interpolatora

Uvođenjem linearnog interpolatora omogućeno je praćenje linearne trajektorije. To konkretno znači da se kod istovremenog gibanja dviju osi, obje osi gibaju isti vremenski period, odnosno u isto vrijeme stižu na željenu poziciju. Kada ne bi bilo interpolatora, praćenje linearne trajektorije ne bi bilo moguće, zato jer bi se brža os uvijek brže pozicionirala od sporije. Samim time to više nije željena trajektorija.

4.7. Regulacija pozicije i brzine

Da bi se alat gibao željenom brzinom i po određenoj trajektoriji, brinu se regulacijski krug pozicije i brzine. Na slici 28. prikazana je struktura regulacije za os X. Kao što je već ranije rečeno, reference pozicije *incX* ili *incY* dobivaju se iz interpolatora, pomoću referentnog posmaka F_{REF} i referentne pozicije X_{REF} ili Y_{REF}. Pomoću reference *incX* te stvarnog položaja *X*, koji se dobije množenjem broja impulsa iz brojača te jediničnim pomacima iz izraza (9) i (10), ovisno o kojoj osi se radi, dobiva se greška pozicije. Greška pozicije se pojačava u P regulatoru pozicije, te se dobiva referenca brzine F_{XREF} ili F_{YREF}, opet ovisno o kojoj osi se radi. Stvarna brzina dobije se ako se broj pristiglih impulsa u vremenu T_{INT}, pomnoži sa jediničnim pomakom, kao što se može vidjeti sa slike 28.



Slika 28. Struktura regulacije pozicije i brzine

Nakon oduzimanja reference brzine i stvarne brzine, dobiva se greška regulacije brzine e_{VX}, s kojom se ulazi u PI regulator brzine. Razlog implementiranja PI regulatora brzine leži u osiguravanju stacionarne točnosti. Stacionarana točnost postiže se zahvaljujući I djelovanju u PI regulatoru. Naime, u slučaju da nema I dijelovanja u regulatoru brzine, onda bi prilikom pojave poremećaja kao, npr. kod uključivanja olovke, došlo do propada brzine zbog povećanja tereta na klizaču i na bubnju, budući da je olovka u kontaktu s bubnjem.

Kako je upravljanje rotacijskim crtalom izvedeno pomoću mikrokontrolera, radi se o diskretnom sustavu, budući da se algoritam regulacije izvršava u konstantnom vremenskom ciklusu, a ne kontinuirano. Stoga forma PI regulatora poprima sljedeći oblik[14]:

$$uPI(k) = K_R \left[e(k) + \frac{T_{INT}}{T_I} \sum_{i=0}^{k-1} e(i) \right] \tag{17}$$

Dakle, vrijednost u *k*-tom trenutku koja izlazi iz PI regulatora jednaka je umošku pojačanja i trenutne greške te umnošku pojačanja s omjerom vremenskih konstanti i sumi svih prošlih vrijednosti. Kako u ovom radu nije provedena simulacija dinamike sustava pomoću čega bi se mogla odrediti potrebna pojačanja, onda se pojačanja traže metodom pokušaja i promašaja, te samim time umnožak pojačanja K_R i kvocijenta dviju vremenskih konstanti možemo zapisati kao jedno pojačanje K_I, kako bi se izraz pojednostavio.

$$uPI(k) = K_R e(k) + K_I \sum_{i=0}^{k-1} e(i) \tag{18}$$

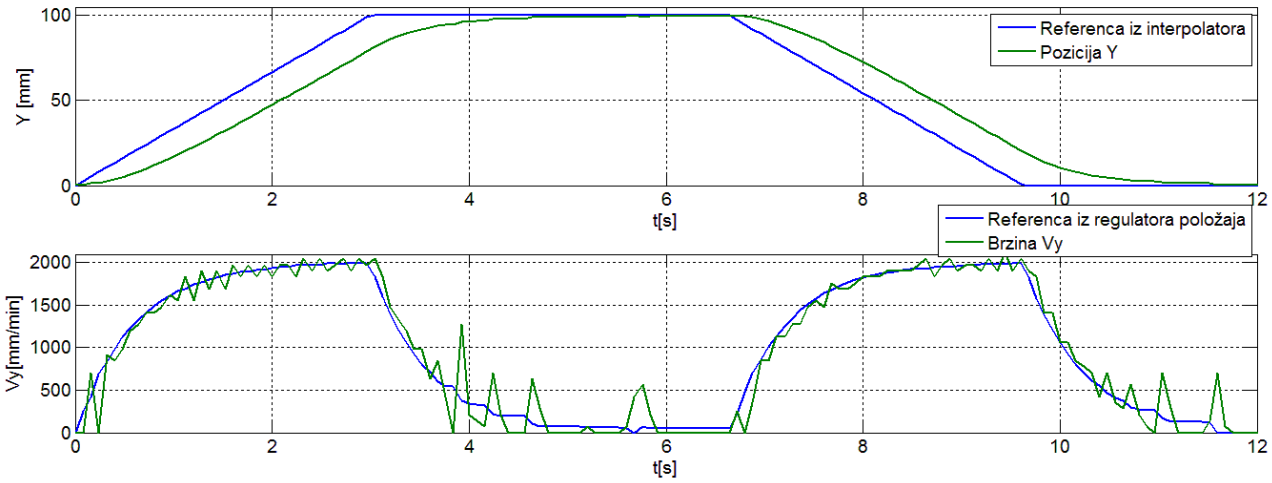
Problem koji se može javiti kod PI regulatora je „prenabijanje“ integratora na visoke vrijednosti. Odnosno, kod režima velikih signala, gdje aktuator ulazi u zasićenje, vrijednost integratora se u svakom sljedećem koraku sumira za iznos umnoška greške i integralnog pojačanja. Zbog čega dolazi do prebačaja i velikih oscilacija jer integratoru opet treba određeno vrijeme da spusti „nabijenu“ vrijednost kad dođe u okolinu ciljane točke. Da bi se to spriječilo, implementira se „anti-windup“ algoritam koji limitira iznos integratora. Izraz za „anti-windup“ se može izvesti kao:

$$U_I = \begin{cases} U_I, & |U_I| < |U_{MAX}| \\ U_{MAX} - U_P, & U_I > U_{MAX} \\ -U_{MAX} - U_P, & U_I < -U_{MAX} \end{cases} \tag{19}$$

Gdje je U_P vrijednost proporcionalnog dijela PI regulaotra, U_I vrijednost integralnog dijelovanja PI regulaotra te U_{MAX}, najveća moguća vrijednost koja se može postići zadnim aktuatorom. U ovom slučaju je U_{MAX}= 255, odnosno ograničenje 8-bitnog PWM-a.

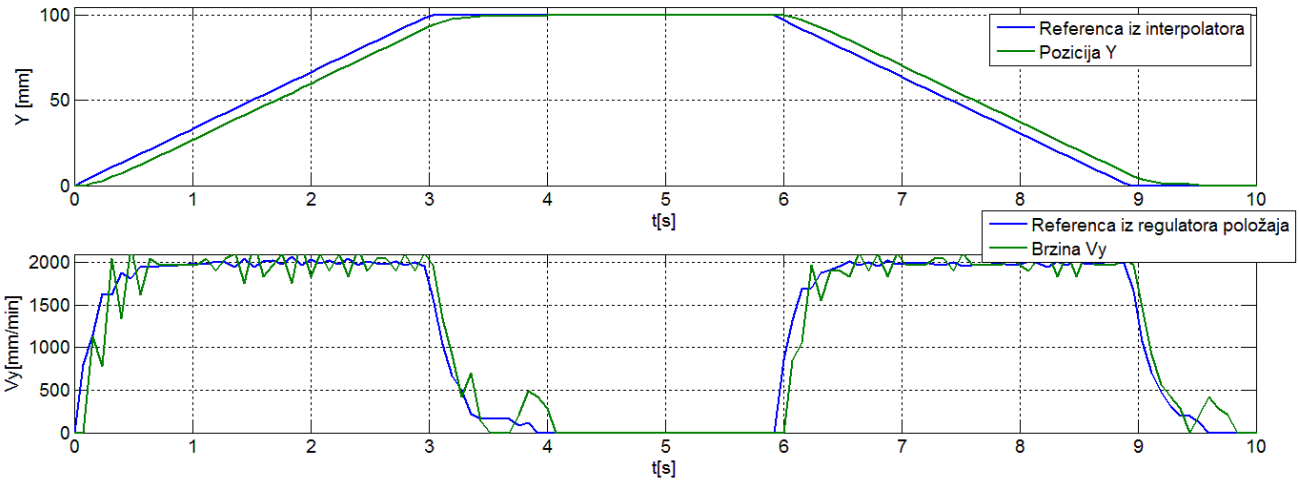
Dakle, postoje tri parametra regulatora za svaku os pojedinačno koje je potrebno podesiti. Podešavanje parametara regulatora pozicije i brzine izvršavaju se nakon montiranja motora i elemenata za prijenos snage na samo rotacijsko crtalo. Kada bi se parametri podešavali na motoru koji nije spojen na radni sustav, onda se s velikom vjerojatnošću može očekivati neželjeno ponašanje motora kod priključivanja u sustav, budući da u obzir nisu uzeti otpori i tereti koje dotični sustav ima.

U nastavku su prikazani snimljeni odzivi pozicije i brzine za os Y i X s enkodera motora rotacijskog crtala. Tako je na slici 29. prikazan odziv Y osi, odnosno bubnja za pomak od 100 mm posmakom od 2000 mm/min. Pojačanja su vidljiva u opisu slike, gdje je K_{PY} – pojačanje P regulatora pozicije za Y os, K_{RY} – proporcionalno pojačanje PI regulatora brzine za Y os, K_{IY} – integralno pojačanje PI regulatora brzine za Y os. Sa slike 29. vidljivo je da odziv pozicije kasni za referencom iz interpolatora, što je normalno za P regulator položaja. Međutim, nedopustivo je da se greška slijeđenja s vremenom akumulira. Greška slijeđenja bi tijekom vremena trebala biti konstantna, što u ovom slučaju nije tako, već se povećava kako se Y os približava vrijednosti od 100 mm. Razlog tome je premalo pojačanje P regulatora pozicije K_{PY}. Kako je izlaz iz P regulatora pozicije referenca brzine PI regulatoru brzine, kao što se može vidjeti sa slike 28., onda će male vrijednosti proporcionalnog pojačanja kod regulatora pozicije iziskivati od brzinske petlje spori odziv brzine, kao što se može vidjeti sa slike 29., gdje je prikazana referenca iz regulatora položaja te stvarni odziv brzine. Kašnjenje pozicijske petlje za referencom iz interpolatora se ne može izmjeriti jer se akumulira tijekom vremena, odnosno raste.



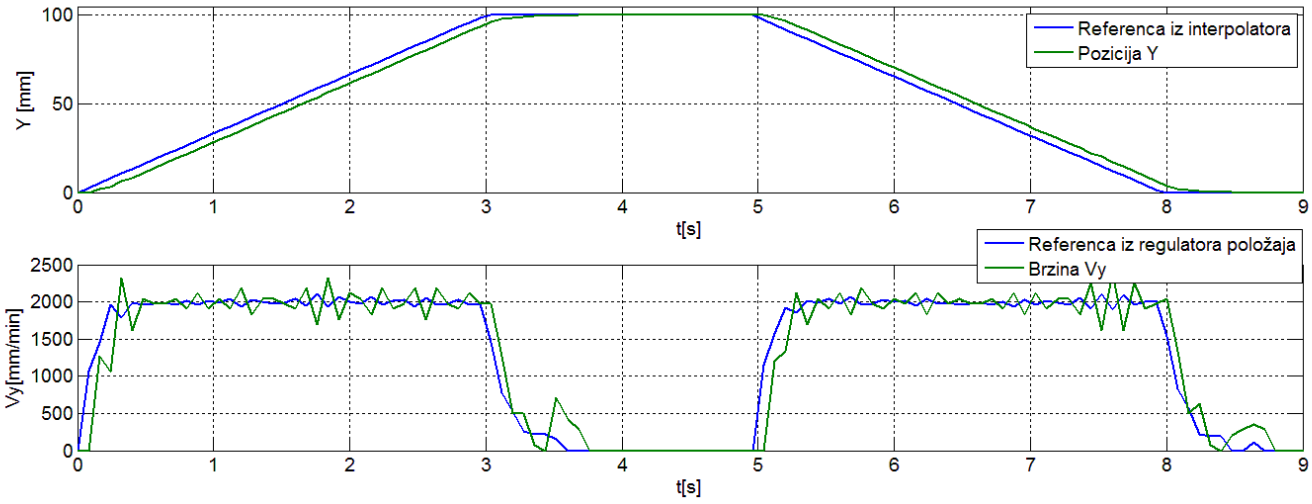
Slika 29. Odziv Y osi za vrijednosti: K_{PY}=200, K_{RY}=0.01, K_{IY}=0.04, F=2000

Samim time potrebno je povećati proporcionalno pojačanje regulacijske petlje pozicije, kako bi se od regulacijskog kruga brzine zahtijevao brži odziv, a samim time i konstantna greška slijeđenja. Na slici 30. prikazan je odziv pozicije i brzine Y osi gdje je povećano proporcionalno pojačanje pozicijske petlje za 50%, odnosno K_{PY} =300. Može se vidjeti da je odziv pozicijske petlje, kao i brzinske, puno bolji nego sa slike 29. Brzina puno brže dostiže referentnu vrijednost od 2000 mm/min, te je samim time greška slijeđenja pozicije konstantna. Kašnjenje pozicijske petlje za referencom iz interpolatora iznosi otprilike 200 ms. Sa slike 30. kod odziva brzina, može se vidjeti da ima još prostora za povećanje proporcionalnog djelovanja regulacijske petlje pozicije, zbog brže dinamike motora od same petlje.



Slika 30. Odziv Y osi za vrijednosti: K_{PY}=300,K_{RY}=0.01, K_{IY}=0.04,F=2000

Na slici 31. prikazan je odziv pozicije i brzine Y osi gdje proporcionalno pojačanje pozicijske petlje iznosi K_{PY} =400. Može se vidjeti da je traženi odziv brzine brži od onoga sa slike 30. Kašnjenje pozicijske petlje za interpolatorom iznosi otprilike 150 ms. Također, sa svih prethodnih odziva, može se vidjeti dominantan utjecaj trenja te greška određivanja brzine zbog male rezolucije enkodera.



Slika 31. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}=0.01, K_{IY}=0.04, F=2000

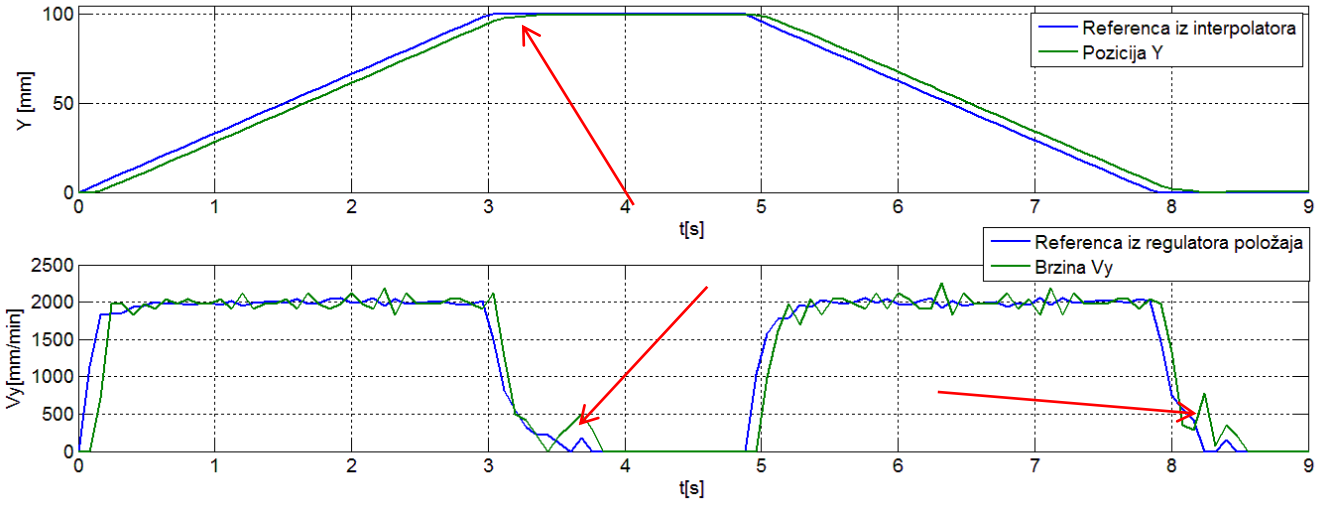
Greška određivanja pozicije zbog premale rezolucije enkodera može se predočiti prema izrazu (20). Dakle, jedan impuls enkodera za Y os u vremenu T_{INT} iznosi:

$$\Delta v_y = \frac{\Delta y}{T_{INT}} 60 = \frac{0.094}{0.08} 60 = 70.5\ mm/\min \tag{20}$$

dok za X os iznosi:

$$\Delta v_x = \frac{\Delta x}{T_{INT}} 60 = \frac{0.054}{0.08} 60 = 40.05\ mm/\min \tag{21}$$

Stoga se mogu vidjeti oscilacije brzine za Y os za ±140 mm/min, što odgovara razlici od 2 impulsa. Ta greška regulacije je neizbježna zbog postojećeg enkodera, no može se ublažiti na dva načina. Prvi način je da se vrijeme interpolacije T_{INT} poveća, a drugi način je da se smanji proporcionalno djelovanje kod brzinske petlje. Prvi način ne bi dao dobre rezultate, jer bi zasigurno došlo do prelijeva brojača koji mjeri poziciju te bi se javila greška pozicije. Drugi problem koji bi se javio je da bi vrijeme uzorkovanja bilo preveliko, te bi sama interpolacija bila upitna. Stoga se primjenjuje drugi način te se smanjuje proporcionalno pojačanje brzinske petlje na K_{RY}=0.001. Sa slike 32. može se vidjeti smanjenje oscilacija u brzini. Utjecaj trenja se naročito vidi kod zaustavljanja motora, kao što se može vidjeti sa slike 32., označeno s crvenom strelicom.

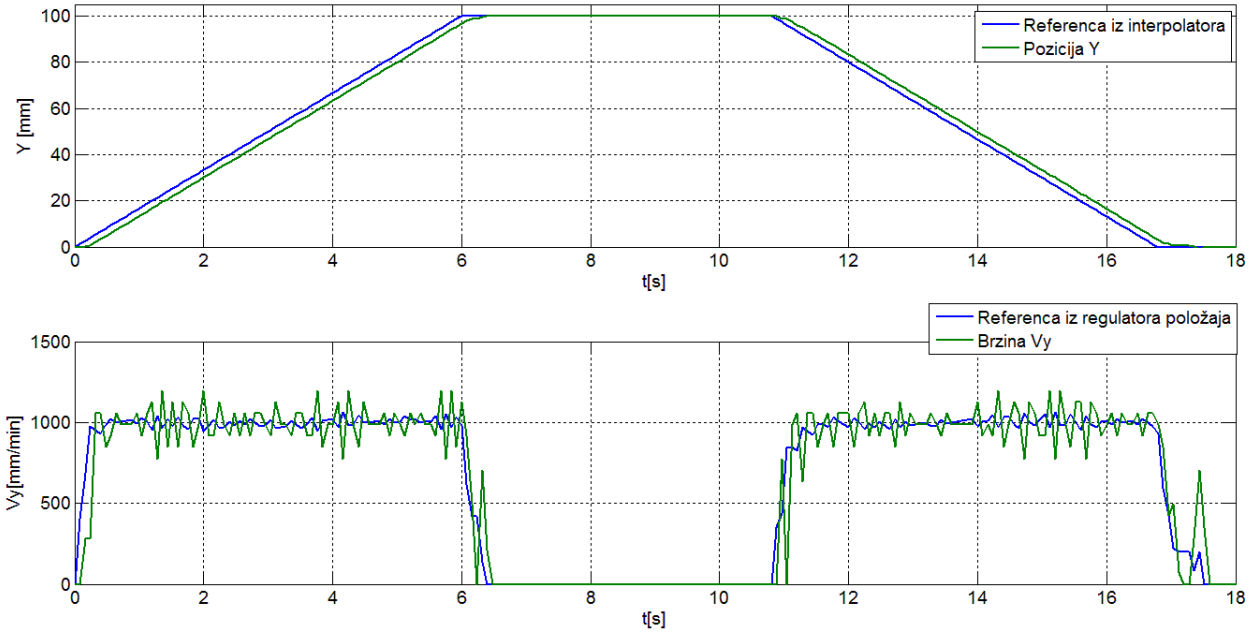


Slika 32. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}=0.001, K_{IY}=0.04, F=2000

Naime, zbog same karakteristike istosmjernog motora, kod niskog napona armature koji je proporcionalan brzini vrtnje, niska je i struja armature koja je proporcionalna okretnom momentu motora. Tako kod usporavanja, dolazi do apsolutne dominacije trenja, zbog trenja u ležajevima i samome planetarnom reduktoru. Sa slike 32. može se vidjeti da kod usporavanja motora, dakle kada Y dostigne otprilike 98 mm, motor stane jer okrenti moment motora nije dovoljan da savlada otpore trenja. Samim time integralno pojačanje unutar brzinske petlje počinje podizati napon armature, što dovodi do pomicanja motora do konačne pozicije Y=100 mm. No, problem koji se tu javlja je smanjenje otpora trenja kod početka gibanja, što uzrokuje trzaj kod zaustavljanja koji je označen s crvenom strelicom na slici 32. Trzaj se javlja zato jer integrator ima određenu akumuliranu vrijednost koja ne može odjednom doći na 0. Stoga nije dobro pretjerivati s integralnim pojačanjem, jer se prevelikim integralnim pojačanjem povećava taj trzaj te se javljaju niskofrekventne oscilacije u brzini. Utjecaj trenja isto bi se mogao ublažiti povećanjem proporcionalnog djelovanja pozicijske petlje, no onda se može dogoditi da dođe do prebačaja, što je nedozvoljivo kod regulacije pozicije. Stoga je potrebno naći kompromis između pojačanja.

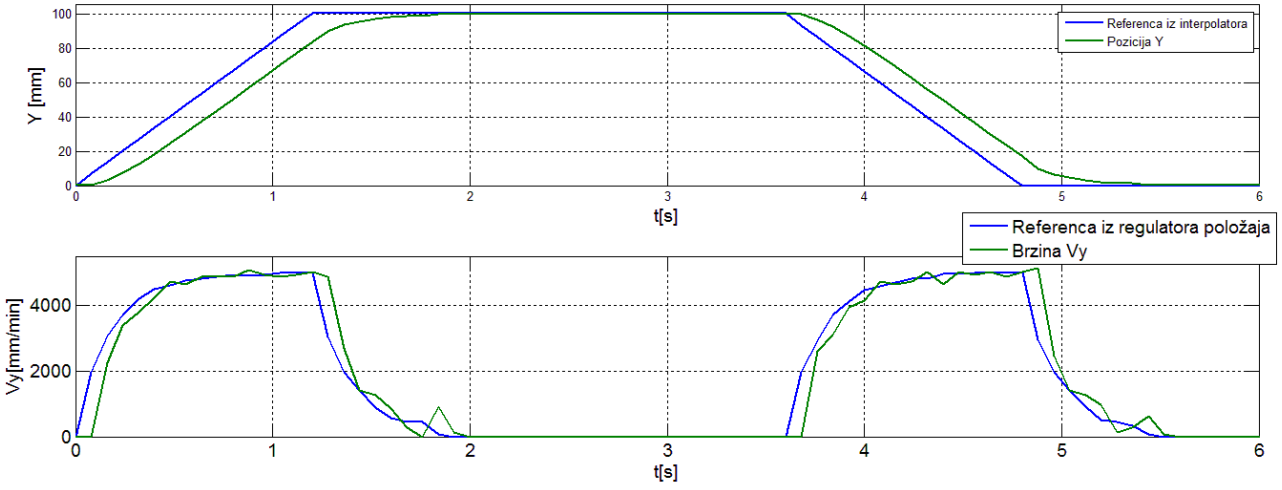
Sa slike 33. može se vidjeti koliko je izražen utjecaj trenja ukoliko se smanji referentna brzina, odnosno posmak. Referentna brzina na slici 33. iznosi 1000 mm/min, te je samim time

napon armature smanjen za pola, čime trenje postaje dominantno u sustavu. Zbog toga se javljaju oscilacije brzine za ±210 mm/min, oko referentne vrijednosti brzine.



Slika 33. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}0.001, K_{IY}=0.04, F=1000

Na slici 34. prikazan je odziv za posmak od *F*=5000 mm/min, što je i maksimalna dopuštena brzina, odnosno brzina kojom se alat giba kod brzog pozicioniranja „G0“. Odziv na slici 33. odgovara narebi „G0 Y100 „ u G kôdu.

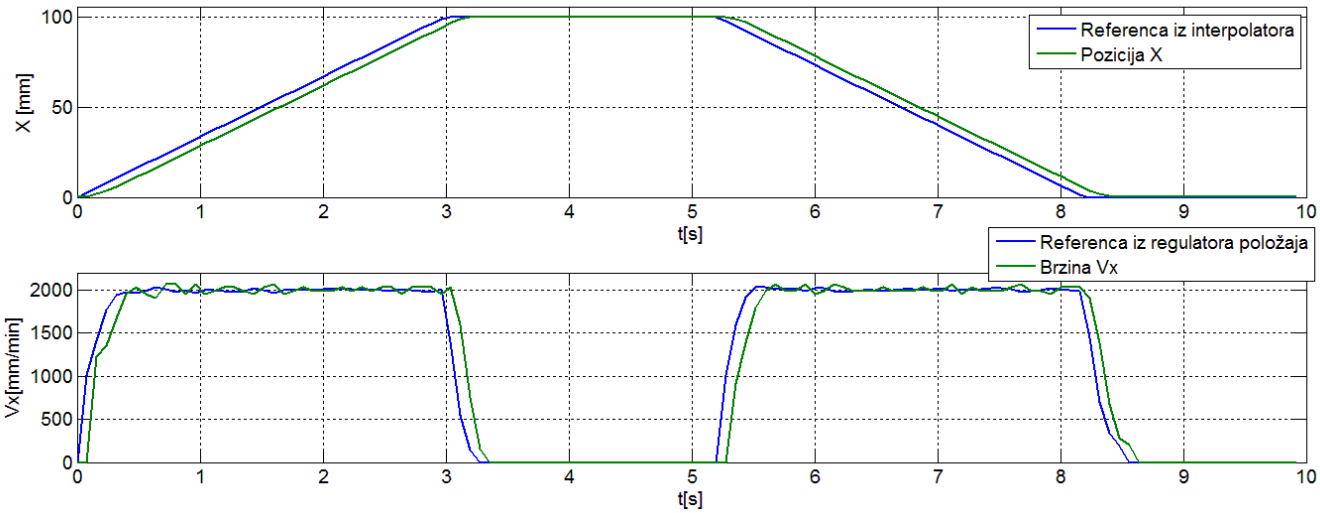


Slika 34. Odziv Y osi za vrijednosti: K_{PY}=400, K_{RY}=0.001, K_{IY}=0.04, F=5000

Može se vidjeti da se opet greška slijeđenja akumullira s vremenom, budući da kod brzog pozicioniranja alat nije u kontaktu s obratkom, odnosno s papirom, onda je takav odziv

dopustiv. Odabrana pojačanja regulatora za Y-os su kako slijedi: K_{PY}=400, K_{RY}=0.001, K_{IY}=0.04.

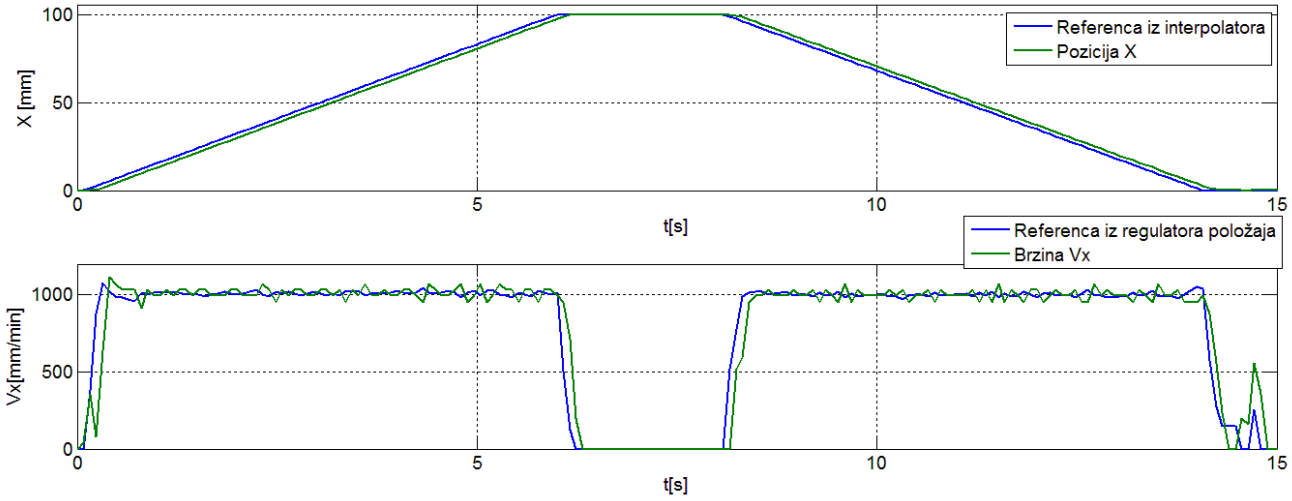
Kako u početku gibanja, X-os i Y-os ne prate rampu, već se zaletavaju, tu se javlja problem kada dvije osi moraju raditi istovremeno. Naime, ukoliko bi X-os bila mnogo sporija od Y-osi, onda bi se iscrtavao pravac na papiru, tek kada bi obje osi dostigle referentnu vrijednost posmaka. To se ne može dopustiti, stoga se početna zaletavanja moraju kompenzirati. Ideja kako kompenzirati zalet motora leži u tome da se odzivi brzina jedne i druge osi približno izjednače. Odnosno, kašnjenje jedne i druge osi za referencom iz interpolatora mora biti jednako. Samim time ublažava se greška kod pokretanja i zaustavljanja. Na slici 35. prikazan je odziv X-osi za vrijednosti parametara dane u opisu slike.



Slika 35. Odziv X osi za vrijednosti: K_{PX}=400, K_{RX}=0.005, K_{IX}=0.04, F=2000

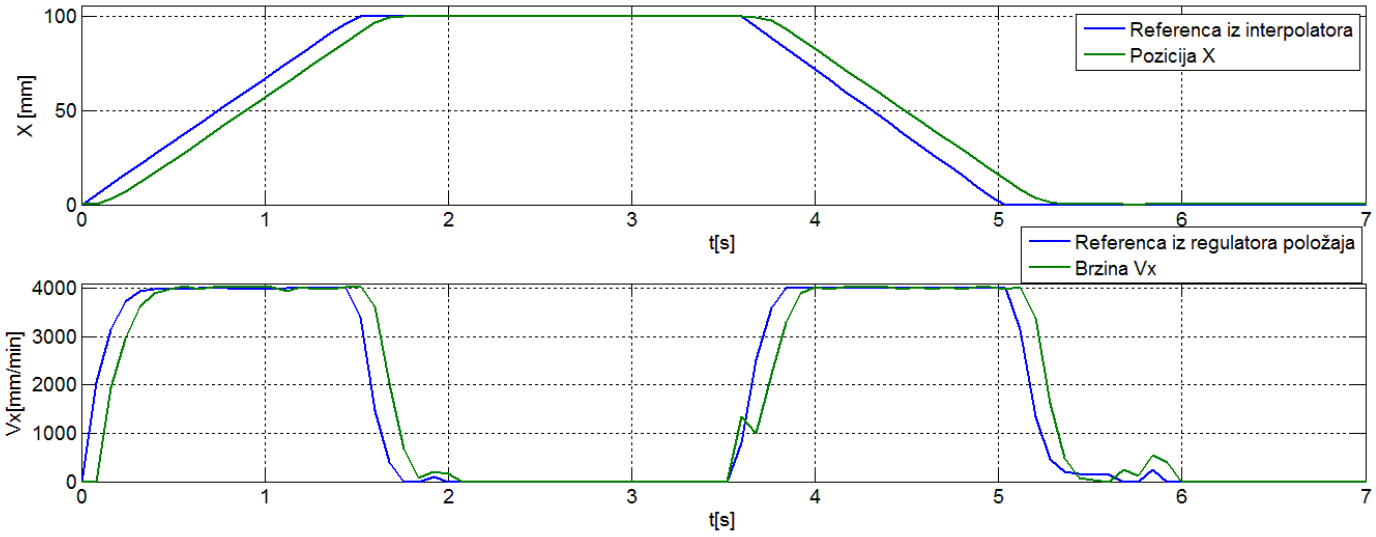
Pojačanja P regulatora pozicije i PI regulatora brzine podešena su pomoću metode pokušaja i pogrešaka. Kašnjenje odziva za referencom iz interpolatora iznosi otprilike 150 ms, kao i kod Y osi, čime se ublažava greška kod zalijetanja motora. Može se vidjeti da su oscilacije brzine kod X osi puno manje nego kod Y osi. Razlog tome leži u izrazu (21), gdje se može vijdeti da je rezolucija brzine za X os skoro dvostruko veća nego rezolucija brzine za Y os, izraz (20). Samim time je omogućena kvalitetnija regulacija brzine, koja dovodi do smanjenja oscilacija.

Slično kao i kod Y osi, za X os prikazani su odzivi za smanjeni posmak koji iznosi F_{REF}=1000 mm/m, te za povećani posmak, odnosno odziv za brzo pozicioniranje „G0“. Tako je odziv za smanjeni posmak prikazan na slici 36., dok je za brzo pozicioniranje prikazan na slici 37.



Slika 36. Odziv X osi za vrijednosti: K_{PX}=400, K_{RX}=0.005, K_{IX}=0.04, F=1000

Sa slike 36. može se vidjeti da se javljaju veće oscilacije brzine kada se smanji referentni posmak, što je i bilo za očekivati zbog dominacije trenja u sustavu i niske rezolucije mjerene brzine. Sa slike 37. može se vidjeti da je kod većih brzina utjecaj trenja skoro i zanemariv, budući da su oscilacije brzine puno manje.

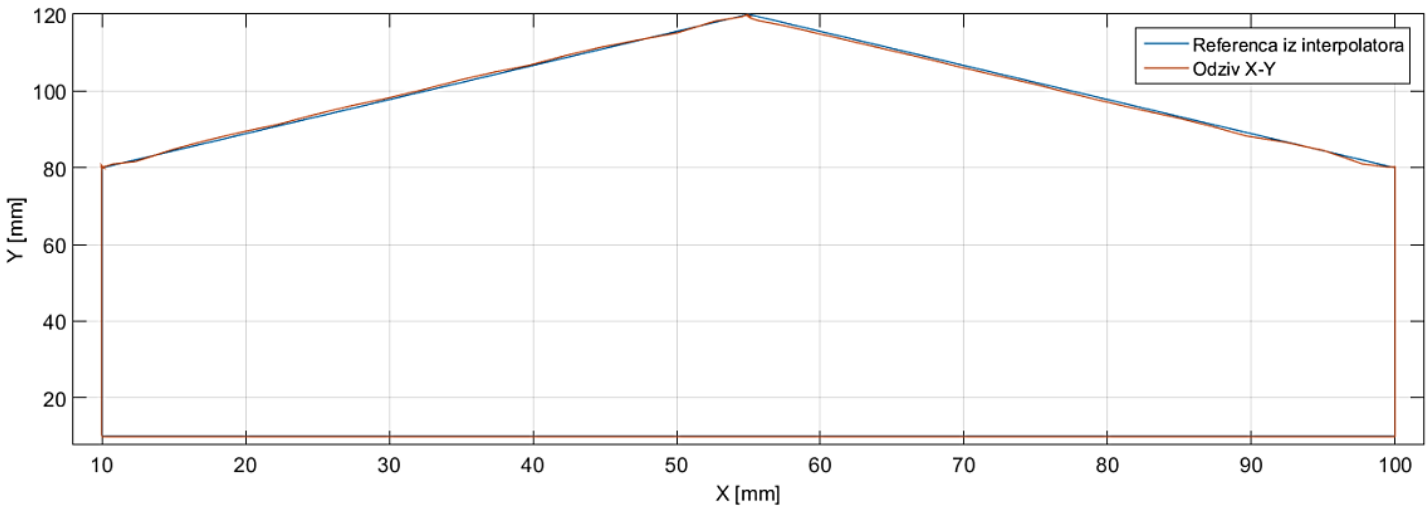


Slika 37. Odziv X osi za vrijednosti: K_{PX}=400, K_{RX}=0.005, K_{IX}=0.04, F=4000

Kako je maksimalna brzina sporije osi, odnosno X osi, 4000 mm/min, onda se i globalno ograničava maksimalna brzina za Y os na 4000 mm/min. Dakle F_{MAX}=4000 mm/min. Ta referenca posmaka će se koristiti isključivo kada je potrebno brzo pozicioniranje, kada olovka nije u dodiru s bubnjem.

Na slici 38. prikazan je odziv dviju osi te referenca iz interpolatora. Odziv odgovara G kôdu:

```
N10 G0 X10 Y10
N20 M3  (Uključi olovku )
N30 G1 X100 F2500
N40 Y80
N50 X55 Y120
N60 X10 Y80
N70 Y10
N80 M5 (Islkjuči olovku)
```

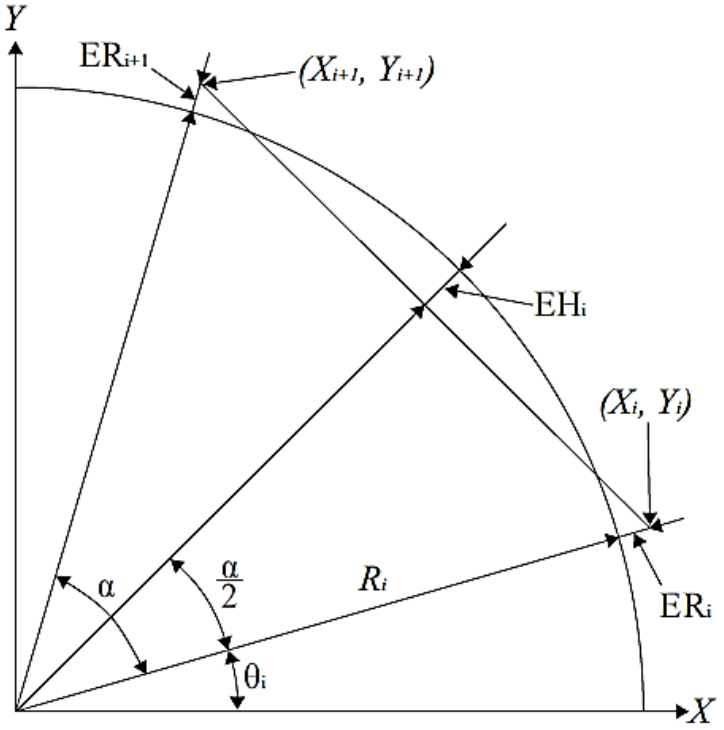


Slika 38. Rezultat odziva X i Y osi

Sa slike se može vidjeti da se odziv sustava odlično poklapa kada ne rade obje osi istovremeno. To se može objasniti pomoći izraza (15) i (16). Dakle, iz G kôda se vidi da je referentni posmak F_{REF}=2500 mm/min. Kada radi samo jedna os, brzina te iste osi odgovara referentnom posmaku, te je samim time ista os manje osjetljiva na trenje. Što se može vidjeti sa slika 34. i 37. gdje je dan odziv s povećanim posmakom. Kada rade dvije osi istovremeno, referentni posmak se dijeli po svakoj osi, ovisno o tome koliki put pojedina os treba prijeći. Ukoliko os treba prijeći veći put, onda se mora gibati većom brzinom nego druga os koja treba prijeći manju udaljenost za isto vrijeme. Smanjenjem brzine osi, dolazi do dominacije trenja kod iste osi te brzina osi varira tijekom vremena, kao što je prikazano na slikama 36. i 33. Upravo zbog toga dolazi do odstupanja odziva sustava od reference iz interpolatora.

4.8. Kružna interpolacija

Kada bi rotacijsko crtalo bilo u mogućnosti izvađati samo linearna gibanja, odnosno iscrtavati samo pravce, njegova primjena bila bi dosta ograničena. Stoga je potrebno realizirati algoritam koji će omogućiti rotacijskom crtalu da može ispisivati kružnice i kružne segmente. Kod CNC sustava kružna interpolacija poziva se naredbom „G2“ ili „G3“, ovisno o željenom smjeru, kako je opisano u tablici 3. Sama ideja kružne interpolacije leži u podjeli kružnice u linijske segmente. Podjelom kružnice u linijske segmente dolazi do dvije vrste grešaka [15]. Prva greška je radijalna greška (eng. Radial error), dok je druga, greška visine tetive (eng. Chord height error). Obje greške prikazane su na slici 39.



Slika 39. Greške kružne interpolacije[6]

Radijalna greška se može izračunati iz poznatih koordinata, odnosno udaljenosti od centra *R(i)* te referentnog radijusa *R*. Izraz (22) prikazuje radijalnu grešku u *i*-tom koraku iteracije, odnosno kod *i*-tog linijskog segmenta [15].

$$E_R(i) = R(i) - R = \sqrt{X^2(i) + Y^2(i)} - R \tag{22}$$

Gdje su *X(i)* i *Y(i)* referentne koordinate u *i*-tom trenutku, kao što je prikazano na slici 39. Greška visine tetive može se izračunati prema izrazu (23).

$$E_H = R - R(i)\cos\left(\frac{\alpha}{2}\right) \tag{23}$$

Gdje je α inkrement kuta, prikazan na slici 39. Iz izraza (22) i (23) određuje se potrebni broj linjskih segmenata koji će aproksimirati kružnicu ili kružni segment. Uvjet koji se postavlja je da obje greške budu manje ili jednake jediničnom pomaku, odnosno izrazima (9) i (10). Kako prema izrazima (9) i (10) Y os ima manju rezoluciju, onda se za referentnu grešku odabire jedinični pomak iz izraza (10). U protivnom, kada bi se za referetnu grešku uzela vrijednost iz izraza (9), odnosno jedinični pomak za X os, onda Y os ne bi mogla zadovoljiti uvjet koji se nameće na greške aproksimacije.

Ako se definira kut u *i*-tom trenutku $\theta(i)$, onda će kut u sljedećem trenutku biti povećan za inkrement kuta α, odnosno $\theta(i+1) = \theta(i) + \alpha$. Iz slike 39. može se izvesti jednadžba diferencija [15]:

$$cos\theta(i+1) = Acos\theta(i) - Bsin\theta(i) \tag{24}$$

te:

$$sin\theta(i+1) = Asin\theta(i) + Bcos\theta(i) \tag{25}$$

Gdje su koeficijenti A=cosα te B=sinα. Koordinate X i Y za kut $\theta(i+1)$ iznosit će:

$$X(i+1) = R(i)\cos\theta(i+1), \qquad Y(i+1) = R(i)\sin\theta(i+1) \tag{26}$$

Nakon uvrštavanja izraza (24) i (25) u izraz (26), dobiva se:

$$X(i+1) = AX(i) - BY(i), \qquad Y(i+1) = AY(i) + BX(i) \tag{27}$$

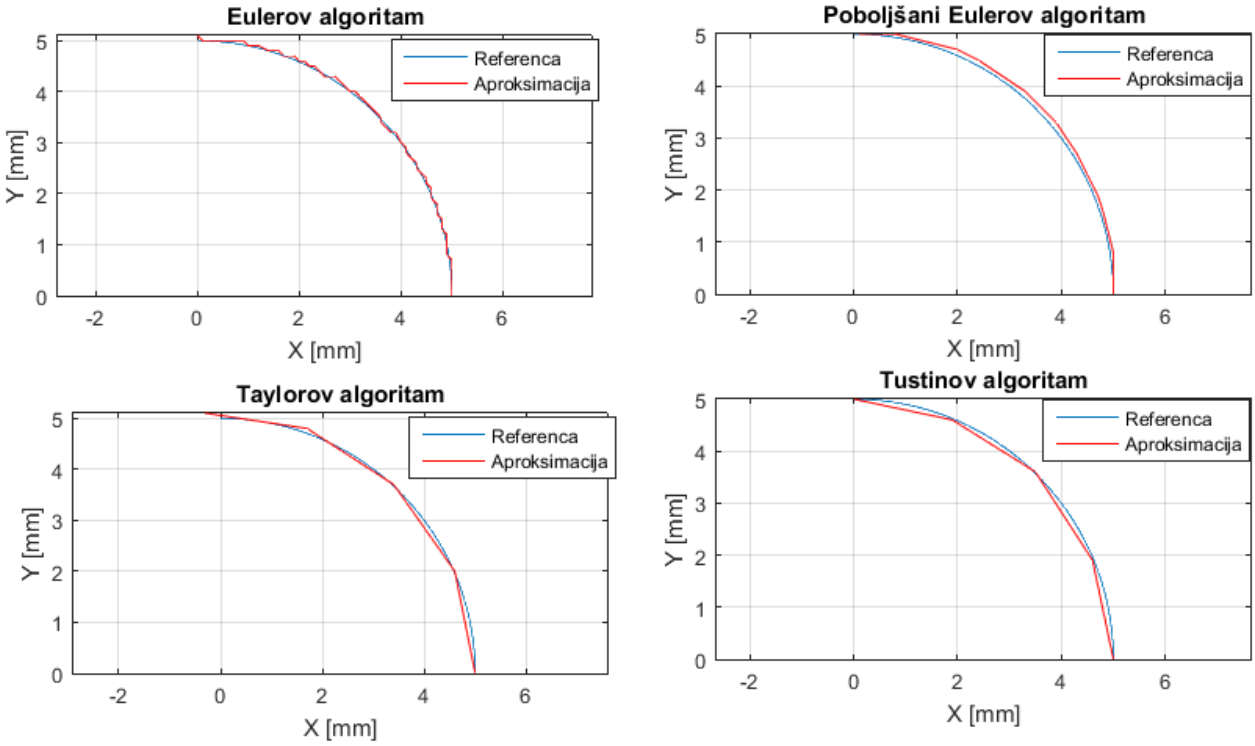
Čime su dobiveni izrazi za računanje koordinata u sljedećem koraku. Važno je napomenuti da izraz (27) vrijedi samo za smjer suprotan kazaljci na satu, odnosno „G3“ u G kôdu, dok je za suprotan smjer potrebno promijeniti predznake.

Postoji više metoda za kružnu interpolaciju, no sve metode koriste prethodno navedene izraze te sa razlikuju samo u određivanju koeficijenata *A* i *B*, odnosno kuta α. Kako je numerički vrlo zahtjevno računati trigonometrijske funkcije, onda se te trigononometrijske funkcije aproskimiraju jednostavnijim izrazima. U tablici 4. dani su gotovi izvedeni izrazi za računanje inkrementa kuta α te broj potrebnih iteracija, odnosno segmenata za opisivanje četvrtine kružnice [15].

Tablica 4. Značajke metoda kružne interpolacije [15]

Metoda	Inkrement kuta α	Broj segmenata N
Euler	$4/\pi R$	$\pi^2 R/8$
Poboljšana Eulerova.metoda	$4/R$	$\pi R/8$
Taylor	$\sqrt{8/R}$	$\pi/4\sqrt{R/2}$
Tustin	$\sqrt{8/R}$	$\pi/4\sqrt{R/2}$
Poboljašana Tustinova metoda	$4/\sqrt{R}$	$\pi/8\sqrt{R}$

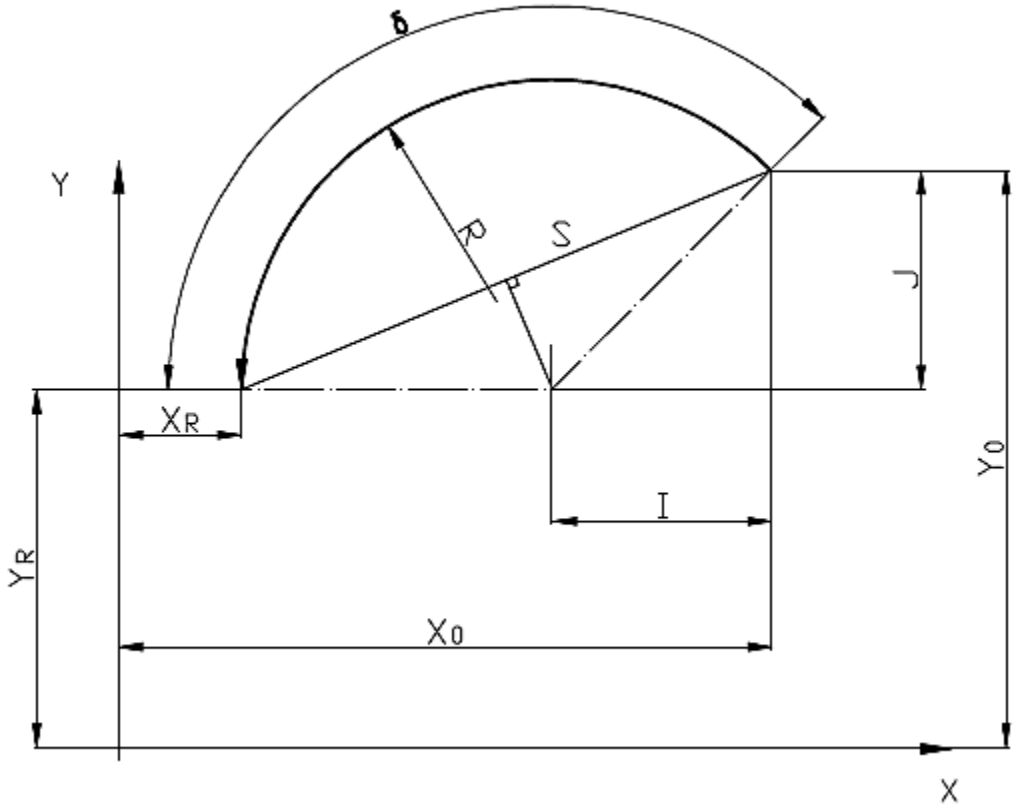
Na slici 40. prikazani su odzivi kod različitih metoda za četvrtinu kruga i za radijus od 5 mm.



Slika 40. Usporedba odziva različitih metoda kružne interpolacije

Sa slike 40. može se vidjeti da je kod Eulerove metode odziv najbliže referenci, no to ima za posljedicu veliki broj segmenata (čak 61.), što kod postojećeg vremena uzorkovanjaT_{INT} ne bi davalo dobre rezultate. Manji broj segmenata postiže se kod poboljšane Eulerove metode, gdje je broj segmenata N=19. Tustinova i Taylorova metoda imaju jednak broj segmenata, N=4, no kod Tustinove metode ne postoji radijalna greška kao kod Taylorove, kao što se može vidjeti iz simulacije na slici 40. U ovom radu dalje je korišten Tustinov algoritam.

Na slici 41. prikazan je opći slučaj kružne interpolacije, gdje je ukupni kut δ veći od $\pi/2$, te se centar kružnice ne nalazi u ishodištu. Također, radi se o smjeru suprotnom od kazaljke na satu. Naredba koja odgovara prikazu sa slike 41. u G kôdu bi glasila : G3 X_R Y_R –I –J. Koordinate X_0 i Y_0 predstavljaju točku iz koje interpolacija počinje. I i J predstavljaju relativnu udaljenost početne točke interpolacije od centra kružnice.



Slika 41. Prikaz općeg slučaja kružne interpolacije

Nakon što interpreter naiđe na oznaku u kodu „G2“ ili „G3“, izvršavaju se sljedeće operacije:

1.) Računanje radijusa kružnice pomoću adresa I i J. Radijus *R* se može lako izračunati pomoću Pitagorina poučka:

$$R = \sqrt{I^2 + J^2} \tag{28}$$

2.) Računanje duljine tetive, označene sa S na slici 41.

$$S = \sqrt{(X_0 - X_R)^2 + (Y - Y_R)^2} \tag{29}$$

3.) Računanje ukupnog kuta δ sa slike 41, pomoću tetive. Na slici 41. može se vidjeti da, ukoliko se spoje početna i krajnja točka, dobiva se tetiva na kružnici. Ukoliko se na tu tetivu povuće iz centra kružnice pravac koji tetivu raspolavlja na dva jednaka dijela, dobivaju se dva pravokutna trokuta iz kojih se jednostavno odredi kut δ.

$$\delta = 2 \cdot \arcsin\left(\frac{S}{2R}\right) \tag{30}$$

4.) Nakon što je izračunati ukupni kut δ, računa se inkrement kuta α prema Tustinovoj metodi iz tablice 4. Važno je napomenuti da su u tablici 4. izrazi prikazani s radijusom u rezoluciji pomaka, odnosno vrijednost radijusa u mm je podijeljena s jediničnim pomakom (BLU) iz izraza (9) i (10). Već je ranije napomenuto da je za referentni jedinični pomak odabrana vrijednost iz izraza (10), stoga se inkrement kuta određuje kako slijedi:

$$\alpha = \sqrt{\frac{8\Delta y}{R}} \tag{31}$$

5.) Nakon izračunatih dvaju kutova, može se izračunati potrebni broj segmenata *N*. Kako inkrement kuta α najčešće nije cjelobrojni dijelitelj ukupnog kuta δ, a broj segmenata mora biti cijeli broj, onda se kvocijent dvaju kutova zaokružuje na cijeli broj. Problem koji se tu može javiti je da se kvocijent zaokruži na manji broj, te je samim time broj segmenata premalen da se zadovolje, već ranije spomenute, dvije greške. Stoga se za svaki slučaj kvocijentu dodaje 1, kako bi se osigurala točnost metode.

$$N = int\left(\frac{\delta}{\alpha}\right) + 1 \tag{32}$$

6.) Budući da se broj segmenata promijenio, odnosno više ne odgovara izvornom kvocijentu ukupnog kuta i inkrementa kuta, onda je potrebno ponovno izračunati inkrement kuta α. Kada se ne bi ponovno izračunao inkrement kuta α, ukoliko

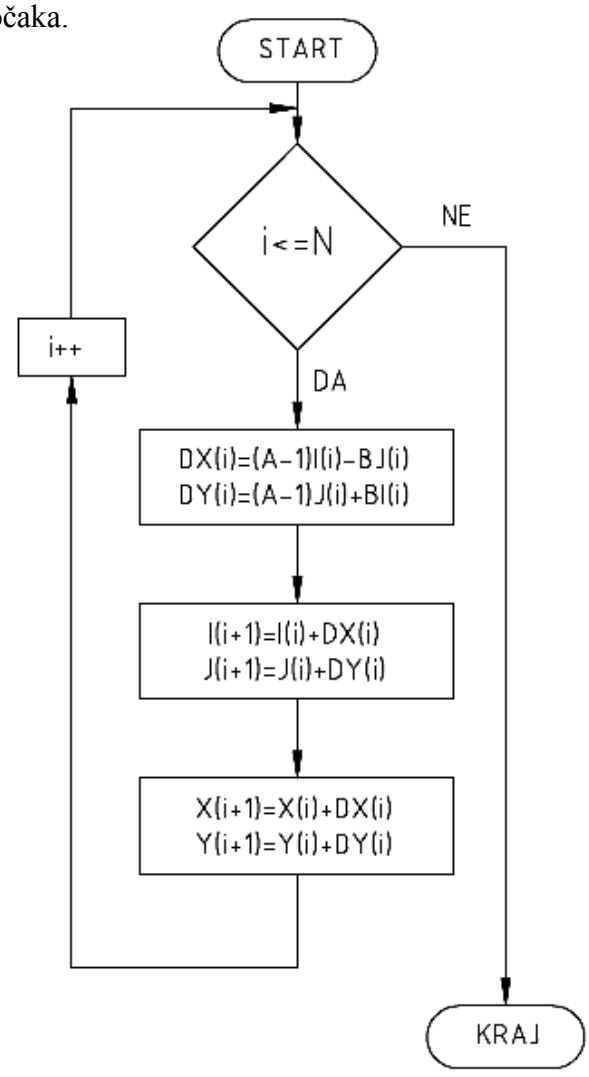
kvocijent dvaju kutova ne bi bio cijeli broj, onda se ne bi postigla krajnja željena točka.

$$\alpha = \frac{\delta}{N} \tag{33}$$

7.) Sada kada su sve veličine poznate, određuju se koeficijenti A i B za Tustinovu metodu[15].

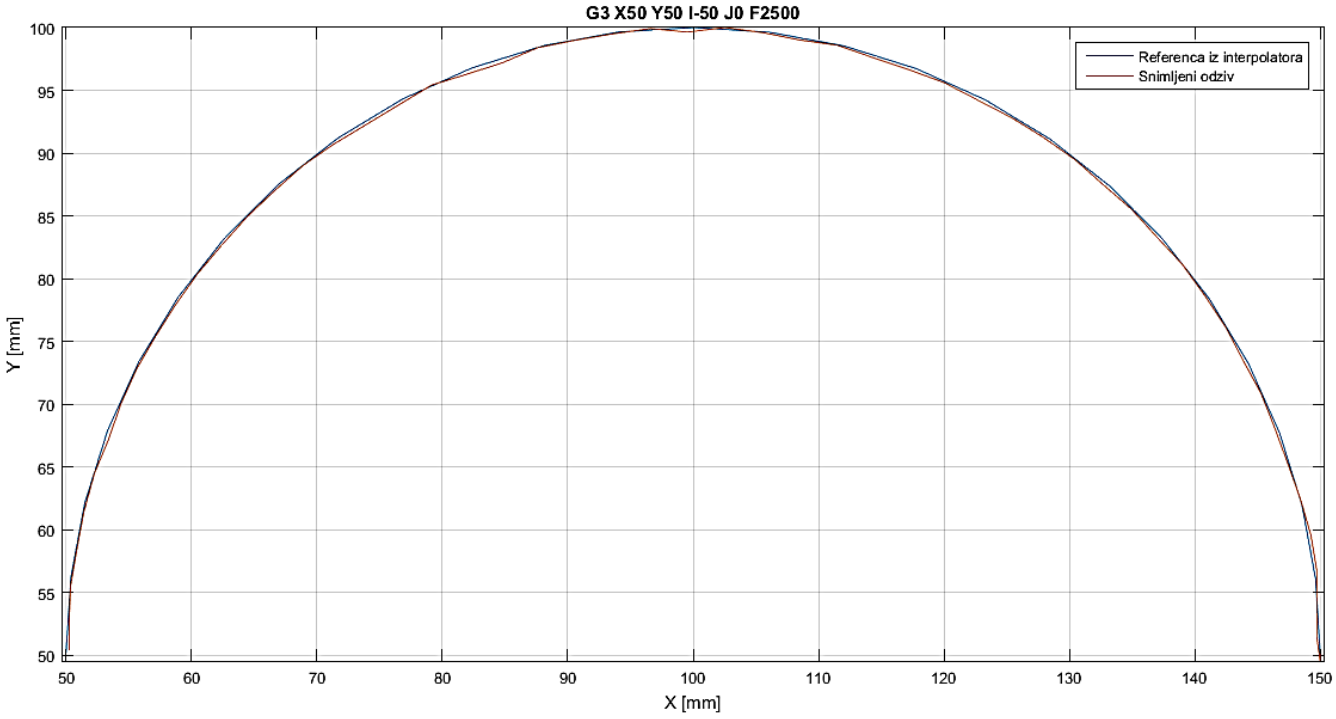
$$A = \frac{1-(\alpha/2)^2}{1+(\alpha/2)^2}, \qquad B = \frac{1}{1+(\alpha/2)^2} \tag{34}$$

Koraci od jedan do sedam se izvršavaju samo jednom, nakon što interpreter naiđe u kodu na oznake „G2“ ili „G3“. Nakon što se izračunaju sve točke od jedan do sedam, pomoću izraza (27) računaju se referentne točke za svaki novi segment, koje se pohranjuju u privremenom međuspremniku u mikrokontroleru. Pomoću blok dijagrama na slici 42. prikazan je algoritam računanja referentnih točaka.



Slika 42. Dijagram toka za računanje referentnih točaka kod kružne interpolacije

S dijagrama toka sa slike 42. može se vidjeti da se kod računanja pomaka *DX* i *DY* upotrebljavaju adrese I i J. Razlog tome je što je jednadžba (27) izvedena za kružnicu koja ima centar u ishodištu koordinatnog sustava. Kako centar kružnice gotovo nikada nije u ishodištu, onda se pomoću adresa I i J računaju pomaci *DX* i *DY* , kao da je centar u ishodištu, te se ti pomaci zbrajaju i oduzimaju od početne točke kružnice. Krajnje točke linijskih segmenata *X(i+1)* te *Y(i+1)* pohranjuju se u međuspremnik. Dobivanjem krajnih točaka linijskih segmenata izvršena je gruba interpolacija. Kako bi se postigla željena obodna brzina gibanja alata, potrebno je podatke koji su dobiveni grubom interpolacijom provesti kroz linearnu interpolaciju, opisanu u potpoglavlju 4.6. Na slici 43. prikazan je odziv rotacijskog crtala na „G3 X50 Y50 I-50 J0 F2500 “.



Slika 43. Odziv na kružnu interpolaciju

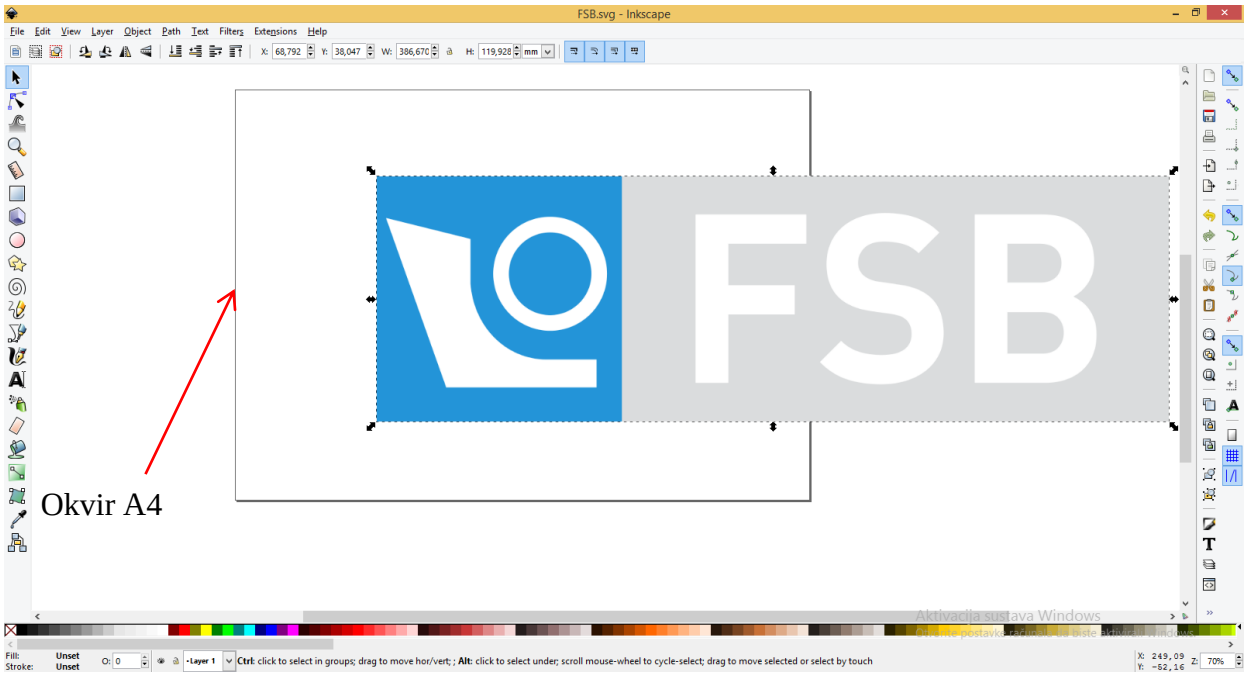
Sa slike 43. se može vidjeti da referenca iz interpolatora dobro opisuje kružnicu, prema Tustinovom algoritmu. No, problem se javlja kod odziva rotacijskog crtala. Kada je brzina Y osi mala (okolica točke X=100, Y=50), utjecaj trenja postaje dominantan, te nastaju greške slijeđenja reference, kao što je opisano u prethodnom potpoglavlju. Unatoč trenju, koje dominira kod malih brzina, greška slijeđenja reference iz interpolatora ne prelazi zahtijevanu točnost od 0.5 mm, stoga su tako male perturbacije u odzivu dopuštene.

5. PRIJENOS CRTEŽA NA PAPIR

Već je ranije spomenuto da je rotacijsko crtalo izvedeno kao CNC stroj. Kako je jedan od zadataka ovog rada prenijeti crtež s računala na papir, onda je potrebno željeni crtež pretvoriti u G kôd, razumljiv mikrokontroleru. Za pretvorbu slike u G kôd korišten je besplatni program Inkscape. Inkscape je program za vektorsko crtanje, baziran na SVG (eng. **S**calable **V**ector **G**raphics) koji se najčešće koristi kod grafike za mobitele, grafike kod Web stranica i slično [2]. Jedna od mogućnosti koje Inkscape pruža je izdvajanje rubova na slikama te generiranje G kôda.

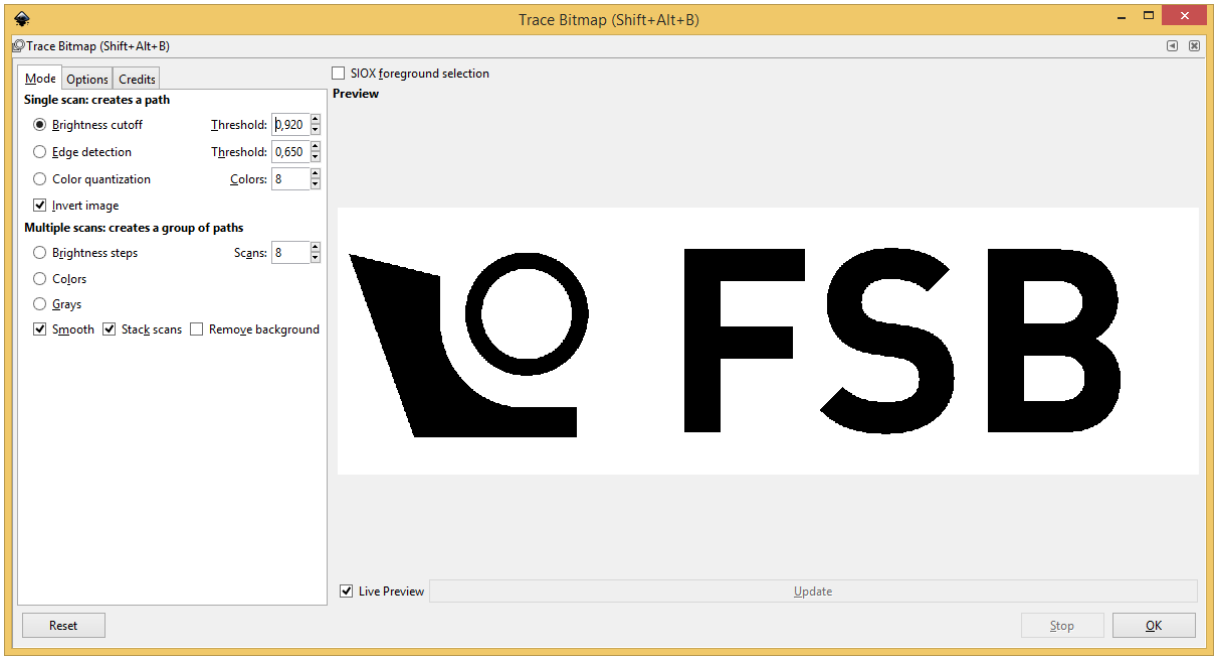
Koraci kod prijenosa željene slike na papir su sljedeći:

1.) Odabrati željenu sliku. Poželjno je da slika bude vektroskog formata, zbog lakšeg izdvajanja rubova. Ukoliko to nije moguće, poželjno je da bude *.png formata, tako da ne gubi detalje prilikom kompresije. Na slici 44. prikazana je odabrana slika, koja je proizvoljno orijentirana i pozicionirana unutar A4 okvira.



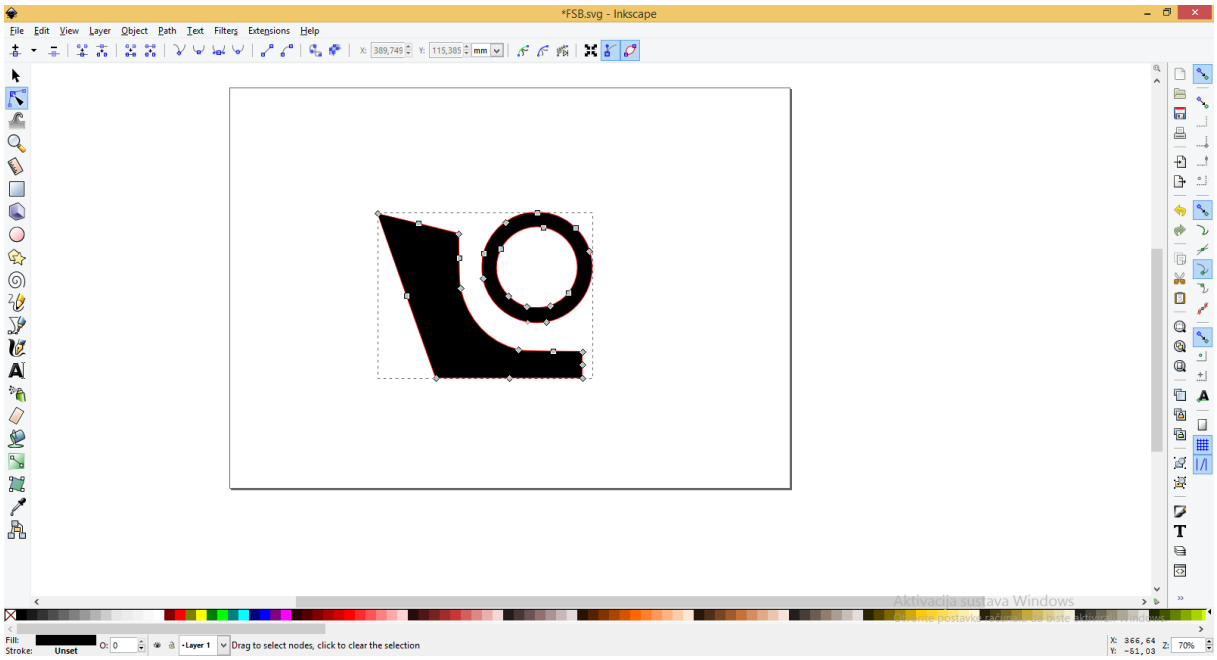
Slika 44. Pozicioniranje slike

2.) Nakon proizvoljnog pozicioniranja i orijentiranja slike slijedi izdvajanje rubova i pomoću narebe „Trace bitmap“, kako je prikazano na slici 45.



Slika 45. Izdvajanje rubova slike

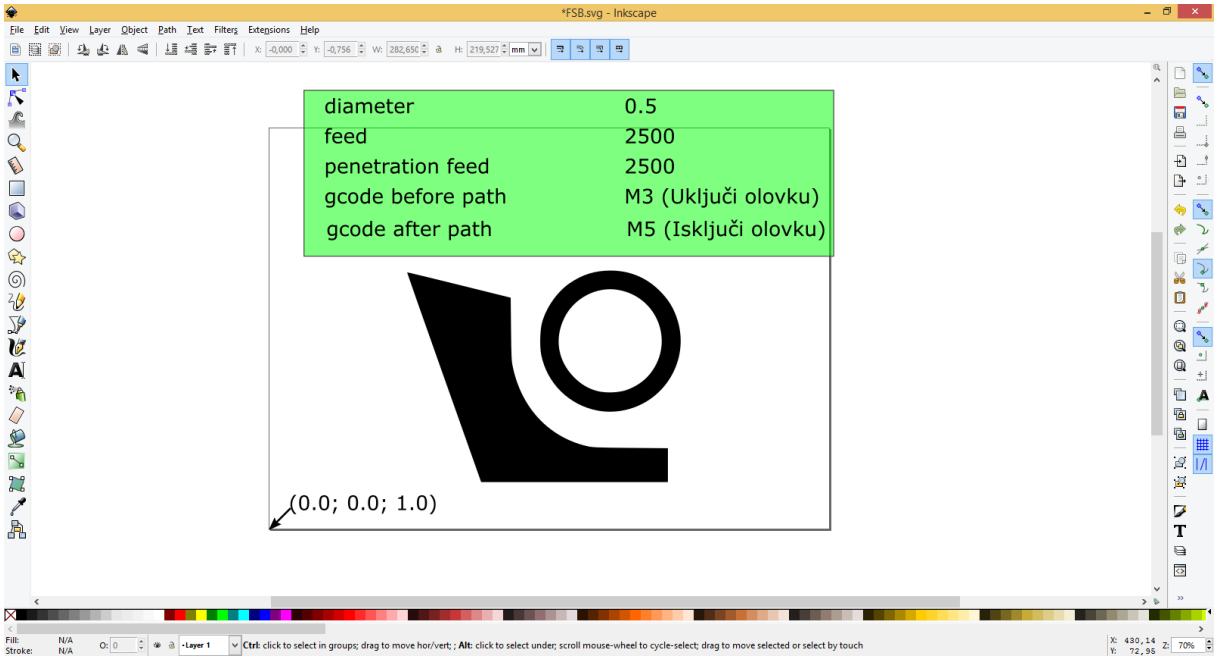
Rubovi se izdvajaju pomoću praga svjetline. Ukoliko svjetlina pojedinog piksela na slici ne prelazi određeni prag (eng. treshold), onda se taj piksel stavlja u crno, a u slučaju da prijeđe, postavlja se u bijelo.



Slika 46. Trajektorije slike

3.) Nakon izdvojenih rubova, na slici se na objektima sa slike 45. pojavljuju se tzv. trajektorije (eng. path), koje će kasnije služiti kao reference za generiranje G kôda. One trajektorije koje ne želimo da se pojave na slici mogu se lako obrisati. Kao što je, radi primjera, učinjeno na slici 46.

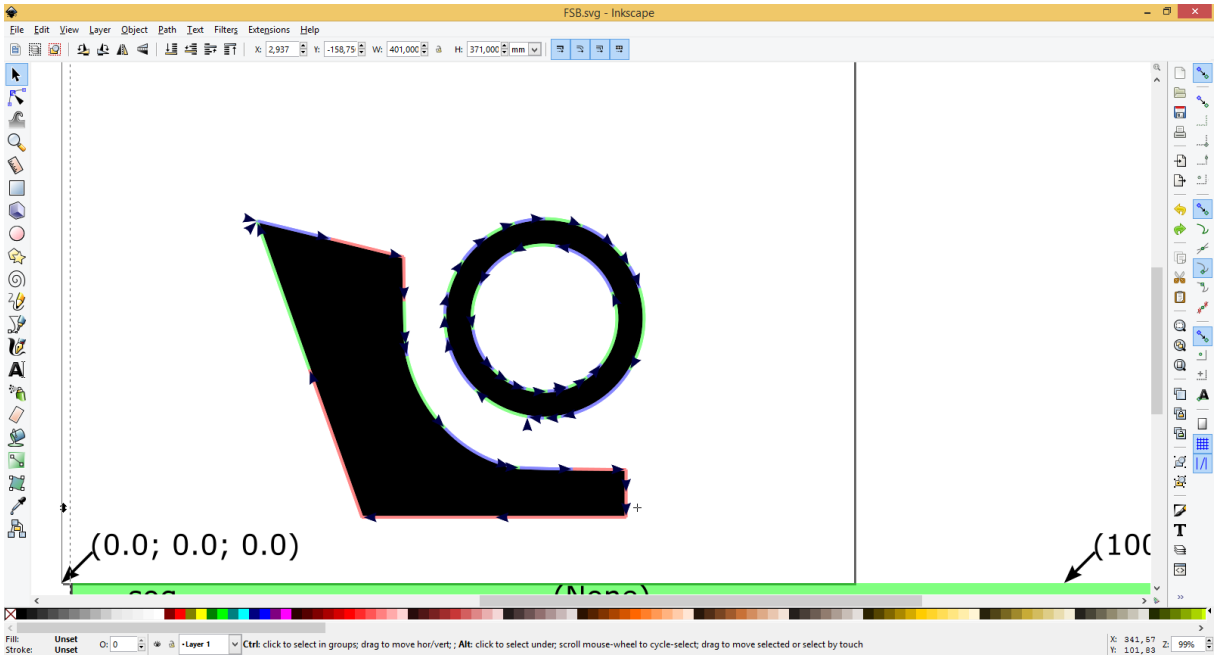
4.) Sljedeće je potrebno odrediti ishodište sustava te odrediti parametre za generiranje G kôda.



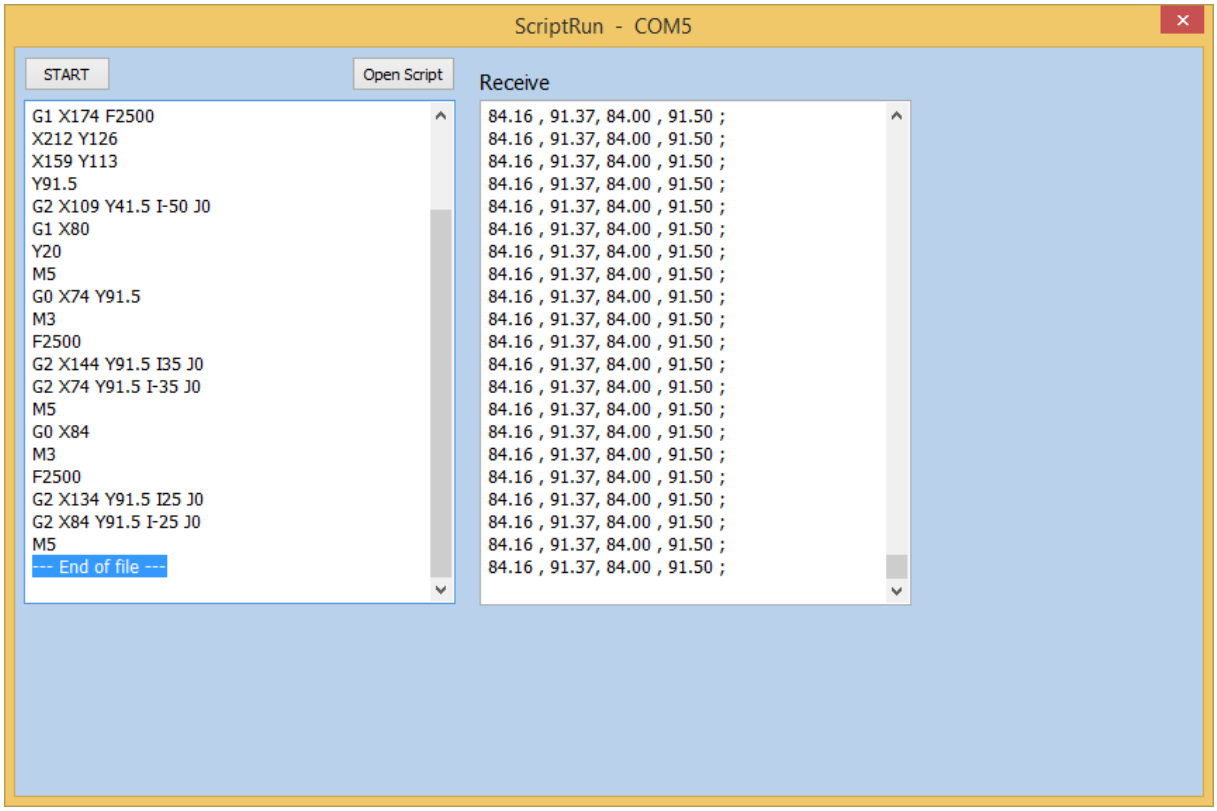
Slika 47. Ishodište slike i parametri za generiranje G kôda

5.) Zadnji korak je samo generiranje G kôda pomoću naredbe „Path to Gcode“. Na slici 48. prikazana je putanja alata koja prikazuje kuda i kako će se alat gibati. Nakon generiranja G kôda, potrebno je urediti sam kôd. Odnosno, pobrisati sve linije gdje se pojavljuje Z os, budući da rotacijsko crtalo nema Z osi pa bi moglo doći do problem prilikom interpretacije u mikrokontroleru.

Nakon generiranog kôda, potrebno je isti poslati na mikrokontroler. To se odvija pomoću terminala koji putem UART-a, koji je opisan u potpoglavlju 3.3, šalje blok po blok kôda, ovisno o tome je li naredba iz prethodnog bloka izvršena. Prikaz sučelja terminala dan je na slici 49. Na lijevoj strani se prikazuju blokovi koji se izvršavaju, dok na desnoj strani mikrokontroler vraća referencu iz interpolatora te poziciju X i Y osi s enkodera svakih T_{INT}. Samim time terminal predstavlja neku pojednostavljenu vrstu MMI-a, koji je spomenut u prethodnom poglavlju.

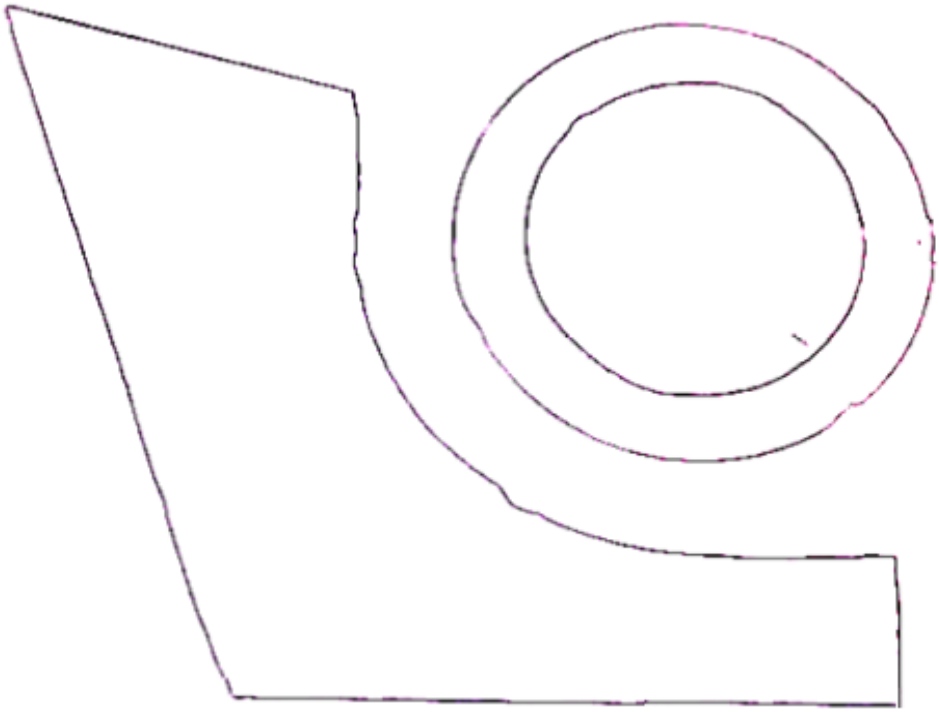


Slika 48. Putanja alata



Slika 49. Terminal za prijenos G kôda na mikrokontroler

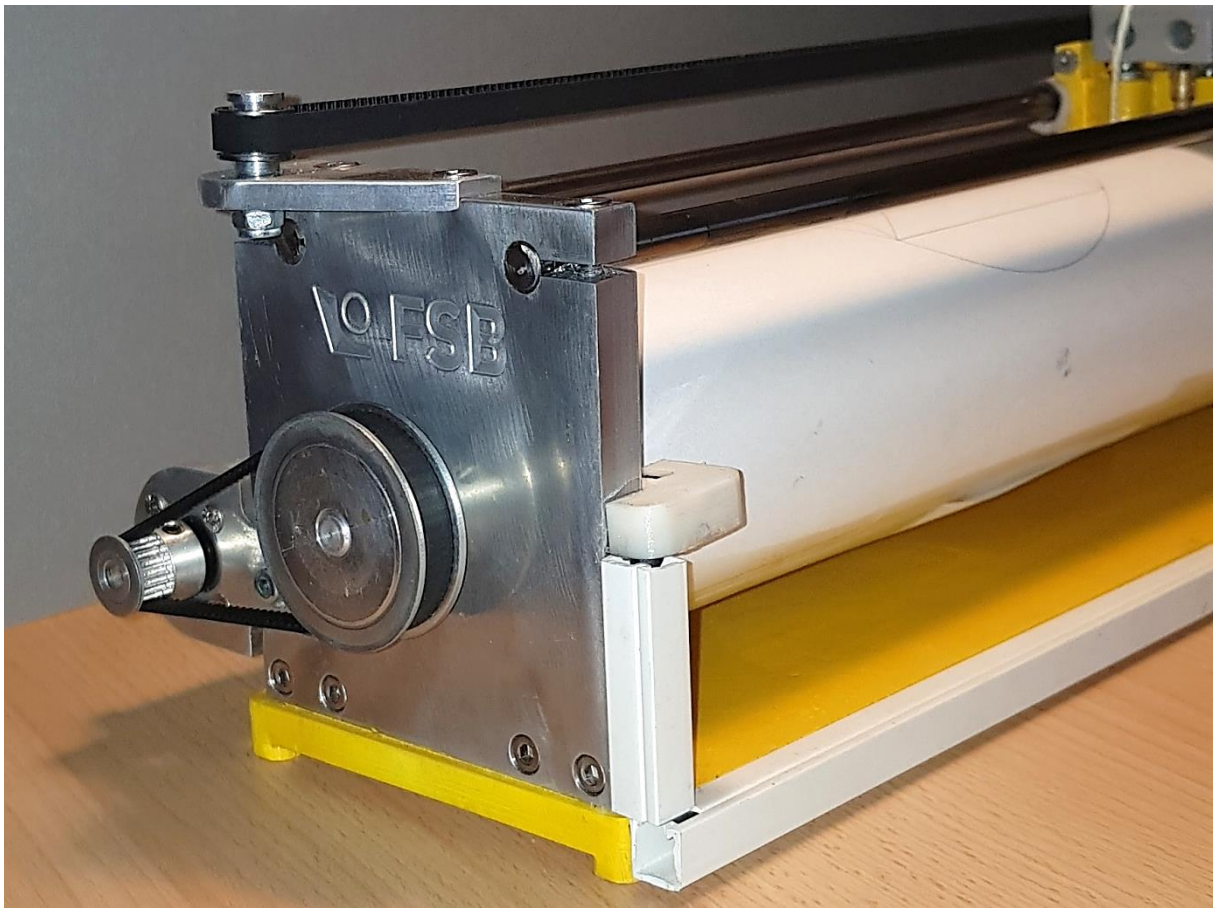
Na slici 50. nalazi se skenirani prikaz stvarnog ispisa na rotacijskom crtalu. Može se vidjeti problem „iskrivljenja slike“ zbog premalene rezolucije brzine i utjecaja trenja kod kružne interpolacije. Dodatni utjecaj ima i zračnost u planetarnom reduktoru te zračnost kod vodilica, koje vode klizač i olovku.



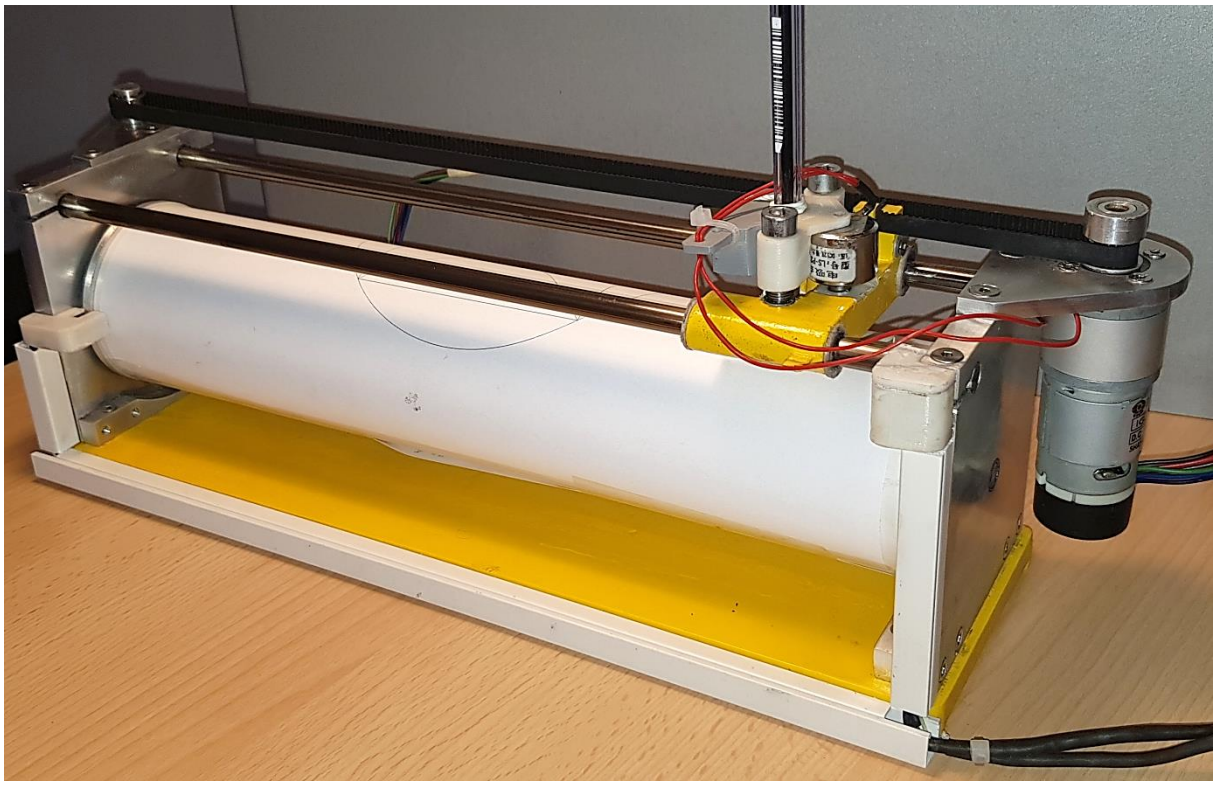
Slika 50. Ispis slike

6. PROCJENA VRIJEDNOSTI UREĐAJA

Kako je dotično rotacijsko crtalo u stvarnosti izrađeno prema CAD modelu, onda je u ovom poglavlju dana procjena vrijednosti svakog dijela crtala. Važno je napomenuti da su neki dijelovi samostalno izrađeni na CNC glodalici te pomoću 3D printera. Samim time je teško procijeniti cijenu dijelova, stoga je u tablici 5. dana vrijednost materijala za svaki dio ili grupu dijelova posebno te ukupna procjena usluge izrade dijelova od aluminija te polimernih dijelova. Procjena vrijednosti usluge izrade dijelova na CNC glodalici temeljena je na potrebnom vremenu za izradu te broju operacija, dok se procjena usluge za 3D printanje svodi samo na vrijeme izrade. Ostali dijelovi poput vijaka, remenica i sl. su standardni te za njih postoji konkretna cijena u maloprodaji, koja je i navedena. Na slikama 51. i 52. dani su prikazi izvedbe rotacijskog crtala.



Slika 51. Prikaz rotacijskog crtala s prednje strane



Slika 52. Prikaz rotacijskog crtala s bočne strane

U tablici 5. navedene su mehaničke i elektroničke komponente, krucijalne za sastavljanje i funkcioniranje rotacijskog crtala.

Tablica 5. Popis elektroničkih i mehaničkih komponentni

Rbr.	Naziv	Količina	Cijena [HRK]
VIJCI			
1	DIN 912 M4x14	8	6,00
2	DIN 912 M3x12	4	2,80
3	DIN 7991 M4x16	9	7,20
4	DIN 7991 M3x10	10	7,50
5	DIN 7991 M3x8	12	9,00
6	DIN 7379 6x20	2	8,70
7	DIN 7379 6x16	1	4,20
8	DIN 985 M5	3	2,76

9	DIN 985 M4	1	0,86
10	DIN 985 M3	2	1,60
11	DIN 125 M4	1	0,20
12	DIN 125 M5	1	0,20
13	ŠIPKA W 08 H6	1000 mm	109,55
REMENICE I REMENI			
1	Remenica GT2 6mm, 20 zubi	2	21,20
2	Remenica GT2 6mm, 60 zubi	1	35,40
3	Remen 240-2GT-6	1	18,35
4	Remen 2GT-6	1x1m	19,70
DIJELOVI IZRAĐENI IZ ALUMINIJA			
1	Noseći stupovi	2	10,00
2	Osovina/vratilo	2	6,00
3	Adapterski prsteni	2	8,00
4	Razni dijelovi	-	10,00
5	Usluge CNC glodanja	-	200,00
DIJELOVI IZRAĐENI 3D PRINTANJEM			
1	Bubanj	1	20,00
2	Klizač+nosač olovke	1	10,00
3	Razni dijelovi(Ležajevi)	-	15,00
4	Usluga 3D printanja	-	300,00
ELEKTRONIČKE KOMPONENTE			
1	Arduino MEGA R3 razvojna pločica	1	130,55
2	LS7084	2	30,60
3	Otpornik 1KΩ	2	0,20
4	Otpornik 10KΩ	2	0,20
5	Otpornik 56MΩ	2	6,40
6	LMD 18200T	2	73,50

7	Motor IG-32GM-01	2	456,00
8	Senzor KTIR0611S	2	19,00
9	Elektromagnet 5V LMR2-5D	1	19,00
10	Dioda 1N4007	2	1,00
		Ukupno:	1570,67

7. ZAKLJUČAK

U sklopu ovog rada konstruirano je rotacijsko crtalo koje ima mogućnost iscrtavanja slika na papir A4 formata. U konstrukcijskom dijelu razrađen je koncept rotacijskog crtala, kao stroja s jednom linearnom osi te jednom rotacijskom osi. Na temelju koncepta, izrađen je CAD model crtala, prema kojem su, pomoću obrade odvajanjem na CNC glodalici te aditivne tehnologije izrađeni svi potrebni dijelovi. Na izrađenoj maketi ispitani su algoritmi upravljanja koji su preuzeti s klasičnih CNC sustava.

Pomoću mikrokontrolera implementirano je pojednostavljeno CNC upravljanje u C jeziku, koje ima ulogu interpretacije koda pristiglog putem UART veze, kružne ili linearne interpolacije te regulacije pozicije i brzine u kaskadnom obliku. Algoritam interpretacije, opisan u potpoglavlju 4.4, radi kako je i zamišljeno, tako da izdvaja reference iz bloka naredbe pristigloga putem serijske veze. Nadalje, proveden je algoritam linearne interpolacije, koji uz pomoć referenci iz interpretera dijeli putanju od početne do referentne točke u male linijske segmente. Problem koji se tu javlja je zaokruživanje zadnjeg linijskog segmenta na način da bude zadnja točke jednaka referentoj. Naime, zbog relativno velikog vremena uzorkovanja T_{INT}, duljina segmenta puta iz izraza (14) je relativno velika, što dovodi i do primjetljive greške kod zaokruživanja zadnjeg segmenta. Na realnim CNC sustavima vrijeme izvađanja interpolacije je oko 1 ms pa je time i ta greška zaokruživanja mala. Valja napomenuti da je, unatoč toj grešci zaokruživanja kod rotacijskog crtala, trajektorija unutar granica tolerancije.

Uz linearnu interpolaciju, također je realizirana i kružna interpolacija. Algoritam kružne interpolacije gleda u kojem smjeru se treba iscrtavati kružnica preko broja uz adresu „G“, kako je prikazanao u tablici 3. Prikazan je algoritam izračunavnja interpoliranih točaka na kružnici te je odabrana Tustinova metoda za izračunavanje koeficijenata A i B, na temelju malog broja linijskih segmenata. Najveći kut koji se kružnom interpolacijom može opisati iznosi 180 °.

Da bi se motori gibali po željenoj trajektoriji (pravcu ili kružnici), implementirana je kaskadna struktura regulacije pozicije i brzine za svaku os zasebno. Kod regulacije položaja korišten je P regulator, dok je kod brzinskog regulatora korišten PI regulator. Na odzivima snimljenim s enkodera motora u potpoglavlju 4.7. vidljivo je da trenje ima dominantnu ulogu

u sustavu kod niskih brzina vrtnje motora, odnosno malih posmaka, što je pogotovo izraženo kod kružne interpolacije, gdje se prilikom opisivanja kružnice brzina mijenja od nula do maksimalne vrijednosti. Uz dominaciju trenja, problem koji se javlja kod regulacije je vrlo niska razlučivost enkodera, koja stvara problem kod mjerenja brzine. Naime, kako se brzina mjeri kao broj pristiglih impulsa u vremenu T_{INT}, onda će niska rezolucija enkodera davati mali broj impulsa, iz čega će biti teško rekonstruirati točnu brzinu. Stoga se kod odziva na slikama u potpoglavlju 4.7. javljaju oscilacije brzine oko reference. Odzivi bi se mogli poboljšati implementacijom estimatora brzine te implementacijom predkomepenzatora trenja na postojeći sustav.

U konačnici, kada se sve zbroji, implementiran je jednostavni i funkcionalni CNC sustav koji može interpretirati i odrađivati naredbe koje mu pristižu putem serijske veze. Kada bi motori imali enkodere veće razlučivosti, tada bi i odzivi bili puno bolji od postignutih.

LITERATURA

[1] https://en.wikipedia.org/wiki/Plotter

[2] http://tavmjong.free.fr/INKSCAPE/MANUAL/html/Introduction-VectorGraphics.html

[3] http://www.misumi-europe.com/en/catalog/vona2/detail/110300405850/?CategorySpec=00000235694%3A%3Aa

[4] Vučković K.: Elementi konstrukcija 2, Remenski prijenos (podloge uz predavaja), FSB, 2014.

[5] http://www.igus.com/contentData/Product_Files/Download/pdf/2016%20drylin%20section.pdf

[6] S. Suh, S. Kang, D. Chung, I. Stroud; Theory and Design of CNC Systems, Springer, London, 2008

[7] https://www.boondog.com/tutorials/mouse/ls708384.pdf

[8] Skalicki B. , Grilec J. : Električni strojevi i pogoni, FSB, 2011.

[9] Pavković D.: Elektromotorni servopogoni (auditorne vježbe), FSB, 2015

[10] Zorc D.: Mikroprocesorko upravljanje (podloga uz predavanja), FSB, 2015

[11] http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

[12] https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter#UART_models

[13] Koren Y. : Interpolator for a Computer Numerical Control System, IEEE Transactions on Computers, Veljača 1976

[14] R. Iserman : Digital Control Systems, Springer, Njemačka, 1989

[15] Y. Koren, O. Masory: Reference- Word Circular Interpolators for CNC Systems, IEEE Transactions on Computers, Studeni 1982

PRILOZI

I. CodeVisionAVR kôd

CodeVision kôd

```
/
Project : Diplomski rad
Version :  1.0
Date    : 21.10.2017.
Chip type               : ATmega2560
Program type            : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 2048
*****************************************************/
#include <mega2560.h>
#include <delay.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <postavke.c>
#define Fmax 4000
#define poz_tol 0.25
#define brz_tol 2
#define Kpx  400
#define KRx   0.005
#define KIx 0.04
#define Kpy  400  //350
#define KRy   0.001
#define KIy 0.04
#define Umax 255
#define T 0.08
#define  dx 0.053
#define  dy 0.094

   char podatak[26];
   char *data;
   float xref=0;
   float yref=0;
   float Fref=0;
   float incX=0;
   float incY=0;
   float dL,L,x,y,deltaX,deltaY, vx,vy, S, R, DX, DY, delta, alfa,A,B;
   int imp_x, imp_y,br_x, br_y,k, N, i,j;
   float buffer[4][40];  // međuspremnik
//******************FUNKCIJA PI REGULATOR**********************
int PIx_reg(float vr, float vm)
```

```
{
  float e,greska;
  static float uI,uPI,uP;

  e=vr-vm;
  greska =fabs(e) ;

   if(greska>brz_tol)
   {
   uP=KRx*e;
   uI+=KIx*e;
   uPI= uP+uI;

   };

   if(uPI>Umax){
   uPI=Umax;
   uI=Umax-uP;

   };

    if(uPI<-Umax){
   uPI=-Umax;
   uI=-Umax-uP;

   };

   return (int)fabs(uPI);

}
 //-------------------------------------------------------------------------------



 //******************FUNKCIJA PI REGULATOR*********************
int PIy_reg(float vr, float vm)
{
  float e,greska;
  static float uI,uPI,uP;
  e=vr-vm;
  greska =fabs(e) ;

   if(greska>brz_tol)
   {
```

```
   uP=KRy*e;
   uI+=KIy*e;
   uPI= uP+uI;

   };


   if(uPI>Umax){
   uPI=Umax;
   uI=Umax-uP;

   };

    if(uPI<-Umax){
   uPI=-Umax;
   uI=-Umax-uP;

   };
      return (int)fabs(uPI);

}
   //---------------------------------------------------------------------------

//*******************FUNKCIJA P REGULATOR**********************
float Px_reg(float ref)
{    float e,A,Up;

     e=ref-x;
     A=fabs(e);

     if(A>poz_tol)
      {
      Up=Kpx*A;
      }else Up=0;

     if(e<0) PORTF.0=0;
     if(e>0) PORTF.0=1;
     if(Up>Fmax) Up=Fmax;

     return Up;
}
// ---------------------------------------------------------------------------
float Py_reg(float ref)
{    float e,A,Up;
```

```
    e=ref-y;
    A=fabs(e);

    if(A>poz_tol)
     {
     Up=Kpy*A;
     }else Up=0;

    if(e<0) PORTF.1=1;
    if(e>0) PORTF.1=0;
    if(Up>Fmax) Up=Fmax;


     return Up;
}
// -----------------------------------------------------------------------------

// Prekidna rutina
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    imp_x=TCNT0;   // Ocitavanje pozicije X os
    imp_y=TCNT5L;  // Ocitavanje pozicije Y os

    TCNT0=0;
    TCNT5L=0;

    // Odredivanje smjera vrtnje
    if (PINF.7==0) br_x+=imp_x;
    if (PINF.7==1) br_x-=imp_x;
    if (PINF.6==0) br_y+=imp_y;
    if (PINF.6==1) br_y-=imp_y;


    // Pretvorba informacije o pozicije iz impulsa u mm
    x=br_x*dx;
    vx=imp_x*dx/T*60;

    y=br_y*dy;
    vy=imp_y*dy/T*60;

    // INTERPOLACIJA
```

```
    if(data[0]=='G'&&(data[1]=='3'||data[1]=='2'))
    { // KRUŽNA INTERPOLACIJA

      if (buffer[2][i]==incX&&buffer[3][i]==incY&&i<N+1)i++;
       L=sqrt((incX-buffer[2][i])*(incX-buffer[2][i])+(incY-buffer[3][i])*(incY-
buffer[3][i]));
       dL=Fref*T/60;
       if(L!=0)
       {
        deltaX=dL*(buffer[2][i]-incX)/L;
        deltaY=dL*(buffer[3][i]-incY)/L;
       };

       if(fabs(buffer[2][i]-incX)>fabs(deltaX))incX+=deltaX;
       else {
       incX=buffer[2][i];
       deltaX=0;
            };

       if(fabs(buffer[3][i]-incY)>fabs(deltaY))incY+=deltaY;
       else {
       incY=buffer[3][i];
       deltaY=0;
           };


     }else
     { // LINEARNA INTERPOLACIJA

  if(fabs(xref-incX)>fabs(deltaX))incX+=deltaX;
  else {
  incX=xref;
   };

     if(fabs(yref-incY)>fabs(deltaY))incY+=deltaY;
  else {
  incY=yref;
   };
   };




  OCR2A=PIx_reg(Px_reg(incX), vx)  ;
```

```
    OCR2B=PIy_reg(Py_reg(incY), vy)  ;
    printf("#%.2f , %.2f, %.2f , %.2f ;/",x,y, incX, incY);

}


// Declare your global variables here
//*******************FUNKCIJA INTERPRETERA G**********************
  float referenca(int ASCII, char *string)
 {
      float ref;
     char refstr[8];
     unsigned char i;
     int j=0;
     int a;
     i=strpos(string, ASCII);

      if(i!=255)       // Ako je kraj bloka?
      {
      while(string[i+1]!=32&&data[i+1]!=13)       // space  I KRAJ
      {
      refstr[j]=string[i+1];
      j++;
      i++;
      };
      ref=atof(refstr);
      for (a=0;a<8;a++)refstr[a]='0';
      return ref;
      }else if(ASCII==88) return xref ;   // stari xref
      else if(ASCII=='Y') return yref ;   // stari yref
      else if(ASCII=='F') return Fref ;   // stari Fref

 }
//-------------------------------------------------------------------------
//*******************FUNKCIJA INTERPRETERA M**********************
 void M_kod( char *string)
 {

     if (string[0]==77&&string[1]==51) // M3
     { PORTF.2=1;
       };
      if (string[0]==77&&string[1]==53) // M5
     { PORTF.2=0;
       };
```

```
  }
void main(void)
{
postavke();

   printf("#INICIJALIZACIJA X OSI /");
   while(PINF.4==0)
   {
    OCR2A=160;
    };
OCR2A=0;
TCNT0=0;


   printf("#INICIJALIZACIJA Y OSI /") ;
   while(PINF.5==0)
   {
   OCR2B=90;
   };
OCR2B=0;
TCNT5L=0;

   printf("#INICIJALIZACIJA USPJEŠNO PROVEDENA /");

while (1)
      {
      if(fabs(y-yref)<poz_tol&&fabs(x-xref)<poz_tol)
    {
       printf("#e/");    // Mikrokontroler je spreman primiti sljedecu naredbu
       data=gets(podatak,26);   // Uzimanje stringa putem UART-a
       M_kod(data);


    xref=referenca('X', data); // Uzmi X referencu iz stringa
    printf("#Xref= %.2f/", xref);
    yref=referenca('Y', data); // Uzmi X referencu iz stringa
    printf("#Yref= %.2f/", yref);
    if(data[0]=='G'&&data[1]=='0') Fref=Fmax;  // Ako je G0
    else {
    Fref=referenca('F', data); // Uzmi F referencu iz stringa
    printf("#Fref=%.2f/", Fref);
    };
     if(data[0]=='G'&&(data[1]=='3'||data[1]=='2'))
```

```
{  //******* KRUŽNA INTERPOLACIJA**********
 buffer[2][0]=incX;
 buffer[3][0]=incY;
i=0;
buffer[0][0]=-referenca('I', data); // Uzmi X referencu iz stringa
printf("#Iref= %.2f/",- buffer[0][0]);
buffer[1][0]=-referenca('J', data); // Uzmi X referencu iz stringa
printf("#Jref= %.2f/",- buffer[1][0]);

R=sqrt(buffer[0][0]*buffer[0][0]+buffer[1][0]*buffer[1][0]);   //Radijus kružnice
S=sqrt((incX-xref)*(incX-xref)+(incY-yref)*(incY-yref)); //    Tangenta
delta=2*asin(S/(2*R));   // ukupni kut delta

if (delta>3.14)delta=3.141;
alfa=sqrt(8*dy/R);
N=1+delta/alfa;
alfa=delta/N;

A=(1-(alfa/2)*(alfa/2))/(1+(alfa/2)*(alfa/2));
B=alfa/(1+(alfa/2)*(alfa/2));
if (data[1]=='2')B=-B;  // G2 ili G3?
// Spremanje tocaka u medjuspremnik
for(j=0;j<N;j++)
{ DX=(A-1)*buffer[0][j]-B*buffer[1][j];
  DY=(A-1)*buffer[1][j]+B*buffer[0][j];
  buffer[0][j+1]=buffer[0][j]+DX;
  buffer[1][j+1]=buffer[1][j]+DY;
  buffer[2][j+1]=buffer[2][j]+DX;
  buffer[3][j+1]=buffer[3][j]+DY;

};
 buffer[2][N+1]=xref;
 buffer[3][N+1]=yref;

j=0;

}
else
{
 //******** LINEARNA INTERPOLACIJA  *********

L=sqrt((incX-xref)*(incX-xref)+(incY-yref)*(incY-yref));
dL=Fref*T/60;
```

```
    if(L!=0)
    {
      deltaX=dL*(xref-incX)/L;
      deltaY=dL*(yref-incY)/L;
    };

   };
   };

    while(k==0)
    {
    // Global enable interrupts
    #asm("sei")
    k++;
    };

     }
}
```