3D snimač prostora mobilnog robota

Pfeiffer, Domagoj

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Domagoj Pfeiffer

Zagreb, godina. 2024

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

3D snimač prostora mobilnog robota

Mentori:

Prof. dr. sc. Mladen Crneković

Student:

Domagoj Pfeiffer

Zagreb, godina. 2024

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

Zahvaljujem se obitelji na podršci, prijateljima na dobrim vremenima i profesoru na strpljenju.

Domagoj Pfeiffer.

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

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: **602 – 04 / 24 – 06 / 01**	
Ur.broj: **15 – 24 –**	

ZAVRŠNI ZADATAK

Student: **Domagoj Pfeiffer** JMBAG: **0035213276**

Naslov rada na hrvatskom jeziku: **3D snimač prostora mobilnog robota**

Naslov rada na engleskom jeziku: **3D space scaner of a mobile robot**

Opis zadatka:

Za pretragu okolnog prostora najveći broj mobilnih robota danas koristi LIDAR. Time se dobiva samo jedan presjek prostora, pa odluka o gibanju i mogućoj koliziji robota s nekom od prepreka, nije pouzdana. Dodatnim stupnjem slobode gibanja robota moguće je 2D snimač pretvoriti u 3D snimač.

Za potrebe vođenja mobilnih robota potrebno je projektirati i testirati 3D snimač.

U radu je potrebno:

- predložiti konstrukciju dodatnog stupnja slobode gibanja LIDAR-a
- organizirati prikupljanje podataka u oblake točaka
- u snimljenom oblaku točaka i u realnom vremenu locirati smjer i opasnost potencijalne kolizije robota

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

Zadatak zadan:
30. 11. 2023.

Datum predaje rada:
1. rok: 22. i 23. 2. 2024.
2. rok (izvanredni): 11. 7. 2024.
3. rok: 19. i 20. 9. 2024.

Predviđeni datumi obrane:
1. rok: 26. 2. – 1. 3. 2024.
2. rok (izvanredni): 15. 7. 2024.
3. rok: 23. 9. – 27. 9. 2024.

Zadatak zadao:
Prof. dr. sc. Mladen Crneković

Predsjednik Povjerenstva:
Prof. dr. sc. Damir Godec

SADRŽAJ

POPIS SLIKA

POPIS OZNAKA

Oznaka	Jedinica	Opis
$\Delta\beta$	'	Greška koračnog motora
p	/	Broj impulsa
C	Ah	Kapacitet baterije ili baterijskog paketa
t	h	Vrijeme izraženo u satima
Imax	A	Maksimalna struja
K	/	Faktor sigurnosti

SAŽETAK

U radu je opisan postupak pretvorbe 2D LIDAR senzora u 3D senzor dodavanjem jedne osi rotacije namijenjenog za mobilnog robota. U prvom dijelu se opisuju komponente, njihova uloga u ovom senzoru i način rada. Nakon odabira komponenti bilo je potrebno osmisliti konstrukciju koja omogućava dodatnu os rotacije. Završetkom konstrukcijskog dijela rada bilo je potrebno napisati kod za zapisivanje i slanje podatka u oblaku točaka sa senzora na računalo i u realnom vremenu locirati smjer i opasnost moguće kolizije. Kroz čitav rad obraća se pozornost na točnost, brzinu i cijenu uređaja. Dodatno kod osmišljavanja svakog uređaja ili proizvoda, svaka odluka kod konstruiranja ili odabira komponenti donosi svoje prednosti i mane. Finalno u zaključku rada predlažu se dodatna rješenja koja bi bila odabrana kada bi se rad počeo raditi ispočetka sa stečenim znanjima.

Ključne riječi: LIDAR, senzor, mobilni robot, oblak točaka,

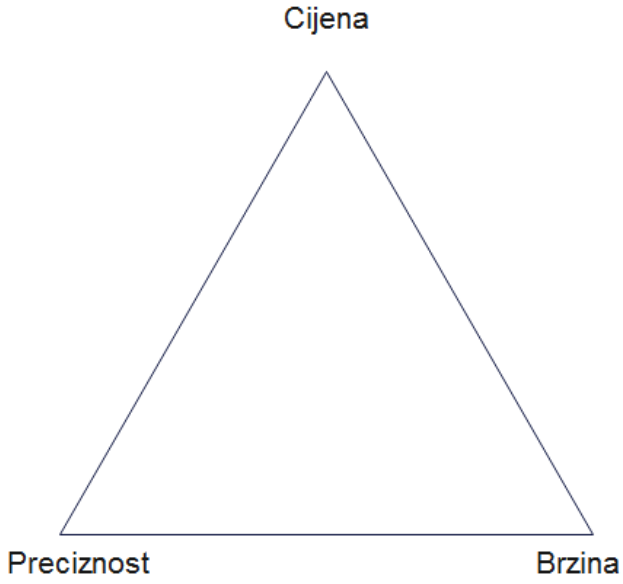
SUMMARY

The peper describes the proces of converting a 2D LIDAR senzor into a 3D sensor by adding one axis of rotation inteded for a mobile robot. The first part describes the components, their role in this sensor and how they work. After the componets were chosen, it was necessary to design a construction that enables an additional axis of rotation. After the complition of hardwere part, it was necessary to write a code that records and sends data in point cloud format from sensor to computer and locates the direction of possible collision in real time. Throught the paper, attention is pait to accuracy, speed and price of device. In addtion to desing of each device or product, each decision in the construction or selction of componets brings advantages and disadvatages that have been obtained and should be emphasized. Finally in the conclusion of the paper, addtional solutions are proposed that would be chosen if the sensor would made from scratch with the acquired knowledge.

Key words:LIDAR, sensor, mobile robot, point cloud.

1. UVOD

Napretkom robotike dolazi do potrebe sa sve većim brojem informacija koje omogućuju točnije i brže određivanje pozicije, orijentacije i okoline u svome radnom prostoru. To je pogotovo bitno za autonomne mobilne robote. Iako u današnje vrijeme postoje razni senzori i sustavi koji omogućuju određivanje pozicije i orijentacije u prostoru poput GPS-a(Global Position System), INS-a(Inertial Navigation System), akcelerometra i žiroskopa, takvi senzori i sustavi omogućuju poznavanje pozicije i orijentacije u nekom proizvoljnom koordinatnom sustavu ali ne i okolini mobilnog robota. Detekciju objekata moguće je obaviti pomoću više vrste senzora poput ultrazvučnih, taktilnih, induktivnih, infracrvenih pa čak i preko vizualnih sustava poput kamera. Kod svakog senzorskog sustava uvijek je bitno naglasiti omjer cijene, brzine i točnosti. Drugim riječima, ako senzor zahtjeva veliku brzinu i visoku točnost najčešće je takav sustav potrebno kompenzirat većom cijenom.

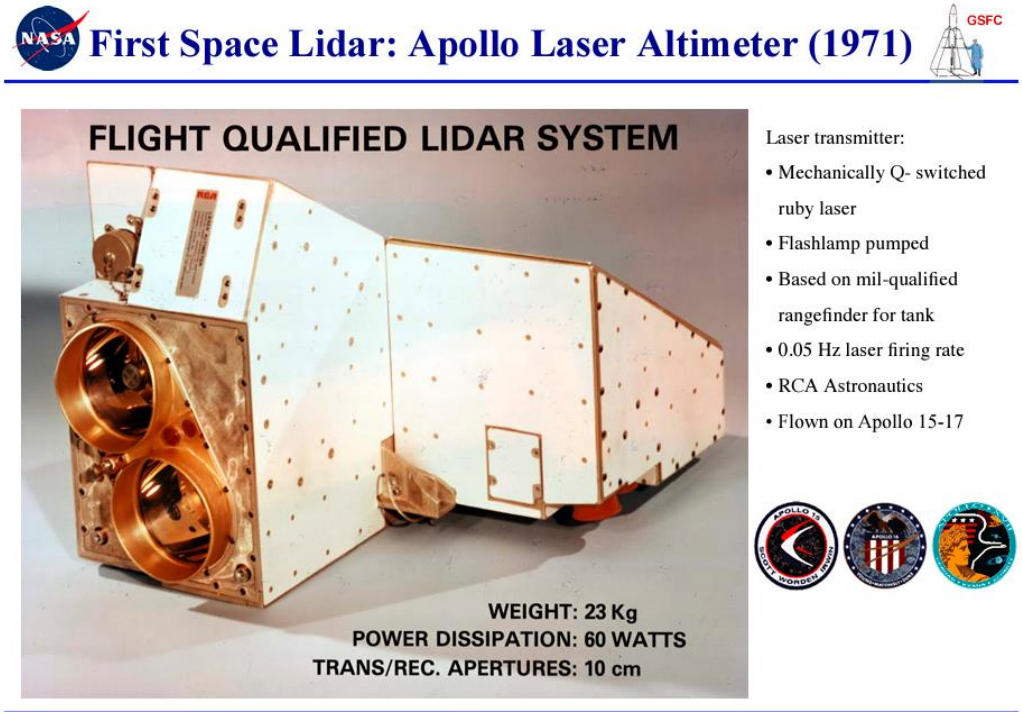


Slika 1. Omjer brzine, točnosti i cijene

Dodatno treba uzeti u obzir u koju vrstu mobilnog robota se ugrađuje senzor i njegovoj okolini. Ako se ugrađuje senzor za autonomnu letjelicu neće biti isti zahtjevi udaljenosti, točnosti i brzine dobivanja informacija kao kod mobilnog robota na zemlji, na vodi ili pod vodom. Kod svakog mobilnih robota je potrebno odrediti u kojoj okolini senzor radi. Kamere u mobilnim robotima ispod vode neće moći vidjeti ništa poslije 2 do 3 metra udaljenosti dok, ultrazvučni senzor ne bi mogao odrediti visinu letjelice ako bi letjelica letjela brzinom većom od brzine zvuka.

1.1. Povijest LIDAR-a(Light Detection and Ranging)

Jedan od prvih funkcionalnih laserskih mjerača udaljenosti je „Colidar Mark II“ osmišljen od strane Rod Smitha 1963. godine bio je omogućen prijašnjim izumom samog lasera Theodora Maimana godine 1960. Jedan od najpoznatijih primjera uporabe laserskog senzora je na Apollu 15 za mjerenje visine letjelice, gdje je NASA prenamijenila laserski mjerač udaljenosti koja je Američka vojska koristila na tenkovima.



Slika 2. Apollo 15. Laserski visinomjer

1.2. Uporaba LIDAR-a u vojnoj tehnologiji

Jedna od zanimljivijih uporaba mjerenje udaljenosti pomoću lasera je u helikopteru AH-64 Apache. Kućiste s laserom i kamerom sa dvije osi rotacije kada izmjeri udaljenost od helikoptera do označene točke, neovisno o pozi i trajektoriji helikoptera, motori kompenziraju dodatnom rotacijom kako bi kućište ostao gledati u istu točku u prostoru neovisno o inputima kopilota ako je on tako odabrao.

Slika 3. AH-64/ASQ 170

1.3. Uporaba LIDAR-a u civilnoj tehnologiji

Najčešća uporaba LIDAR sustava je za mapiranje prostora u autonomnim robotima i automobilima.

Slika 4. Autonomni automobil sa 3D LIDAR sustavom

Glavna prednost LIDAR sustava u odnosu na današnje 3D kamera njihova cijena. Dok se najjeftiniji 2D LIDAR može nabaviti danas po cijeni od 150 Eura, početne cijene 3D kamera su oko 500 Eura ovisno o proizvođaču.

Dodatno potrebno je spomenuti kako kamere imaju puno veću količinu podataka i smetnji koje je potrebno obraditi ili ukloniti kako bi se došlo do korisne informacije što dodatno povećava cijenu krajnjeg uređaja zbog potrebe za snažnijim procesorom.

Slika 5. Autonomni robot za čišćenje sa 2D LIDAR sustavom

2. Cilj završnog rada

Cilj ovog završnog rada je predložiti konstrukciju dodatnog stupnja slobode gibanja pretvorbe 2D LIDAR senzora u 3D, prikazati organizaciju podataka u oblake točaka i prikazanom oblaku točaka i u realnom vremenu locirati smjer i opasnost potencijalne kolizije robota. Iako se 3D sustav može konstruirati pomoću 2 motora i jednodimenzionalnog LIDAR-a(mjerenje udaljenosti u smjeru pravca), u ovom završnom radu je odabran 2D LIDAR sustav(mjerenje udaljenosti u jednoj ravnini) te je dodana jedna os rotacije pomoću dodatnog step motora kojim se omogućuje mjerenje udaljenosti u prostoru(sferni koordinatni sustav opisan s dva kuta i jednom udaljenosti). Dodatno pokazati stečena znanja tijekom studija u osmišljanju, konstruiranju i izradi jednog senzora, odabiru komponenti te spajanju fizičkog i programskog dijela za izrađeni senzor. Finalno obratiti pozornost na problematiku točnosti kod izrađenog senzora, dati okvirnu usporedbu cijene komponenti i sličnih gotovih proizvoda na tržištu i predložiti neka rješenja kada bi se ovaj završni rad počeo raditi ispočetka.

3. Komponente

Za bolje razumijevanje rada konstrukcije i cijelog sustava potrebno je razmotriti elektroničke i električne komponente u ovom završnom radu. Kao i kod svake konstrukcije, robota ili proizvoda, potrebno je naglasiti i obrazložiti prednosti i mane koje su dobivene odabirom. Komponente koje se koriste u ovom završnom radu sa kratkim opisom su sljedeće:

1. Raspberry Pi 3B je mikroračunalo s modificiranim operativnim sustavom Linux(Rasbian) koje omogućuje upravljanje step motora, prikupljanje i obradu podataka sa senzora te komunikaciju s drugim računalima.
2. Koračni motor NEMA17 s redukcijom 100:1 omogućuje rotaciju oko dodatne osi. Razlog odabira reduktora je povećana preciznost između svakog koraka ali zbog redukcije brzine potrebno je dulje vrijeme skeniranja prostora.
3. Kontroler za step motor DM332T. Koračni motori zahtijevaju poseban način upravljanja za razliku od klasičnih DC motora. Dodatno kako koračni motori koriste velike struje za stvaranje momenta dok mikrokontroleri koriste male struje za rad i komunikaciju, sekundarna uloga ovog uređaja je odvajanje visoke i niske struje pomoću svjetlosnih signala, odnosno optokaplera.
4. Baterijski paket koji osigurava izvor električne energije za napajanje elektroničkog sklopovlja i elektromotora. Konfiguracija ovog baterijskog paketa je 4s1p sa Liti-Ionskim ćelijama LG18850 koje u ovoj konfiguraciji osiguravaju maksimalni napon od 15.2 V i maksimalni kapacitet od 3.2 Ah.
5. DC/DC konverter čija je uloga pretvorba napona od 15.2 V na napon od 5 V na kojem radi Raspberry Pi 3B i ostalo sklopovlje malih snaga.
6. Slamtech RPLIDAR A1 se sastoji od DC motora, infracrvenog lasera i svjetlosnog senzora sa lećom. DC motor je na statoru pokreće rotor pomoću remenice dok glava rotacije šalje informacije o udaljenostima statoru pomoću infracrvene diode u središtu rotacije.

3.1. Raspberry Pi 3B

Kao centrali sustav za prikupljanje, kontrolu i slanje podataka odabran je Raspberry pi 3B. Ovaj uređaj može se koristi kao računalo i u isto vrijeme kao mikrokontroler, koristi operativni sustav Rasbian ali pomoću svojih 40 GPIO(General Purpose Input-Output) pinova može upravljati raznom elektronikom ovisno o potrebama korisnika. Bitno je napomenuti da kako se koristi operativni sustav, to može dovesti do greške za vrijeme izvođenja programa jer se u pozadini vrti niz drugih procesa i programa za koje korisnik ne mora biti ni svjestan da se izvode.

Slika 6. Raspberry Pi 3B

Specifikacije uređaja:

- Procesor: Četvero jezgreni, 64-bitni procesor BCM2837
- 1GB Radne memorije (RAM)
- 4 USB-A port-a
- HDMI port
- Ethernet port
- BCM43438 čip omogućuje bežičnu komunikaciju s ostalim uređajima
- SD kartica od 64 GB koja se koristi kao trajna memorija za spremanje podataka i pokretanje operativnog sustava i svih programa na njemu

Raspberry Pi 3B odabire se u ovom završnom radu iz sljedećih razloga:

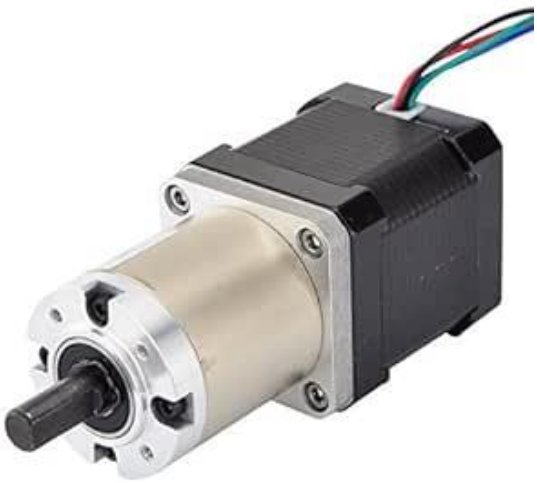
1. Procesorska moć samog uređaja koja omogućuje prikupljanje podataka i kontrolu step motora
2. Jednostavno spajanje svih komponenti na isto sučelje. SLAMTEC RPLIDAR A1 koristi USB-A konektor, iako je moguće uspostaviti komunikaciju preko GPIO sučelja, ovakvo pojednostavljenje omogućuje spajanje pokretanja LIDAR uređaja u jako kratkom roku. Pomoću GPIO pinova omogućeno je upravljane step motora NEMA 17, dodatno preko istih pinova Raspberry Pi može se napajati ili napajati druge uređaje niskog napona.

3. Jednostavna komunikacija s drugim uređajima i računalima. Operativni sustav Rasbian dolazi s instaliranim programom VNC server pomoću kojeg je moguće spajanje i komuniciranje s drugog računala. Komunikacija se može izvesti žičano (ethernet kabel) ili bežično(bežična lokalna mreža).
4. Odabire se zbog već stečenog iskustva na prijašnjim projektima, programski jezik za upravljanje i prikupljanje podataka je Python u kojem je već stečena određena razina znanja na kolegiju „Objektno programiranje“ za vrijeme studija

3.2. NEMA 17 koračni motor sa planetarnim reduktorom 100:1

3.2.1. Kratki opis koračnih motora

Koračni motori najčešće se koriste u robotima, manipulatorima i CNC strojevima kod kojih je potreban veliki moment i točnost pozicioniranja. Veliki nedostatak ovog motora je njegova masa(kako bi se razvio veliki moment potreban je veliki broj namotaja što deblje bakrene žice). Planetarnim reduktorom ostvaruje se mogućnost točnijeg pozicioniranja jer svaki puni okret vratila na kraju reduktora zahtjeva 100 punih okretaja odabranog motora prije reduktora. Kao nedostatak reduktora, reducira se broj okretaja u minuti, kako je primarni kriteriji ovog rada bila odabrana točnost pozicioniranja a kao sekundarna brzina odabran je reduktor velikog prijenosnog omjera. Treba naglasiti da bi se redukcija mijenjala ovisno o potrebnim brzina i potrebama točnosti mobilnog robota kao kompromis točnosti i brzine mjerenja.



Slika 7. NEMA 17 sa reduktrom 100:1

Neke od specifikacija odabranog NEMA 17 motora s reduktorom 100:1 su:

- Vrsta motora: Bipolarni
- Napon: 2.8 V
- Struja po fazi: 1.68 A
- Minimalni korak: 0.018° ± 5%
- Moment držanja: 44 Nm
- Maksimalno radijalno opterećenje: 100N
- Maksimalno aksijalno opterećenje: 50N
- Planetarni reduktor
- Utor za pero na izlaznom vratilu (D-tip)

3.3. Kontroler za koračni motor DM332T

Prvobitna funkcija kontrolera za koračne motore je pretvaranje PWM impulsa sa mikrokontrolera u niz otvaranja i zatvaranja sklopki, odnosno tranzistora koje propuštaju struju kroz pojedine faze motora kako bi nastalo rotacijsko magnetno polje na statoru. Druga funkcija kontrolera za koračne motora je odvajanje struje velikih iznosa koje služe za pokretanje motora i struja malih iznosa koje dolaze od strane mikrokontrolera. Način na koji je to postignuto je pomoću optokaplera. Na samom uređaju postoje dvije pločice koje su strujno i naponski nezavisne. Jedini način na koji komuniciraju je preko svjetlosnih signala, gdje pločica koja je spojena s mikrokontrolerom ima infracrvene diode koje šalju svjetlosne signale fotoosjetljivim tranzistorima na drugoj pločici uređaja koje ovisno o diodi propuštaju ili zaustavljaju protok struje određene faze motora čime se rotira magnetno polje statora.

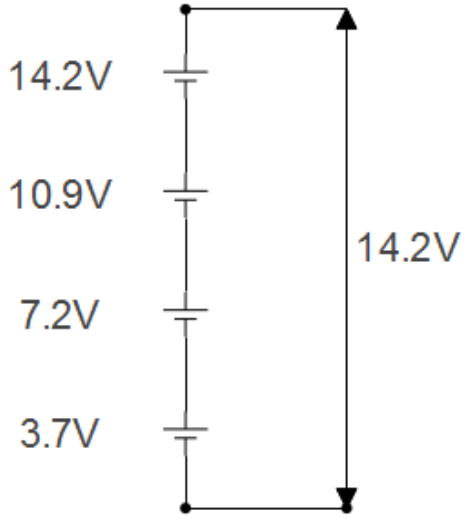
Slika 8. Kontroler za koračni motor DM332T

Tablica koja prikazana na slici 9. za kontroler koračnog motora određuje ovisno o uključenim ili isključenim prekidačima(SW1 do SW6) maksimalnu struju i broj impulsa od strane mikrokontrolera koji su potrebni za puni krug rotacije, odnosno omogućuje mikrokoraka.

Slika 9. Tablica kontrolera step motora

3.4. Baterijski paket

Kao osnovni izvor napajanja koristi se baterijski paket s 4 Liti-Ion baterije LG18850. Svaka pojedina baterija ima nominalni napon od 3.7 V i maksimalni kapacitet od 3.2 Ah. Kada se baterije spajaju u seriju, ukupni napon baterijskog paketa se povećava ali kapacitet ostaje isti, dok kod spajanja baterija u paralelu, napon baterijskog paketa ostaje isti ali kapacitet baterijskog paketa se povećava.



Slika 10. Serijsko spajanje baterija

Razlog odabira baterijskog paketa je mogućnost montiranja svih električnih komponenti na glavu rotacije 3D snimača prostora kako bi se senzor mogao rotirati beskonačni broj puta oko svoje osi. Naprotiv bi broj okretaja koračnog motora ovisio o duljini žice izvora napajanja.

Kako bi se odredilo okvirno vrijeme rada čitavog uređaja određen je maksimalni potrošač, u ovom završnom radu to je bio motor NEMA17, njegova maksimalna potrošnja iznosi 1.68 A te ukupni kapacitet baterijskog paketa iznosi 3.2 Ah te je uzet faktor sigurnosti od 1.5 kako bi se uračunale i ostale komponente niske potrošnje energije. Iz sljedeće formule dobiveno je vrijeme rada.

$$t[h] = \frac{C}{Imax \cdot K} \quad \textbf{(2)}$$

Za pretpostavljenim i zadanim vrijednostima:

$C = 3.2\ [Ah]$
$Imax = 1.68\ [A]$
$K = 1.5$

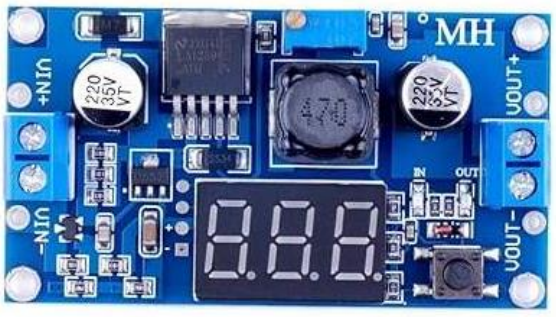
Izračunato je vrijeme rada:

$t = 1.27[h] = 76.2[min]$

Što zadovoljava zahtjeve ovog završnog rada kako bi se pokazao koncept rada 3D snimača prostora.

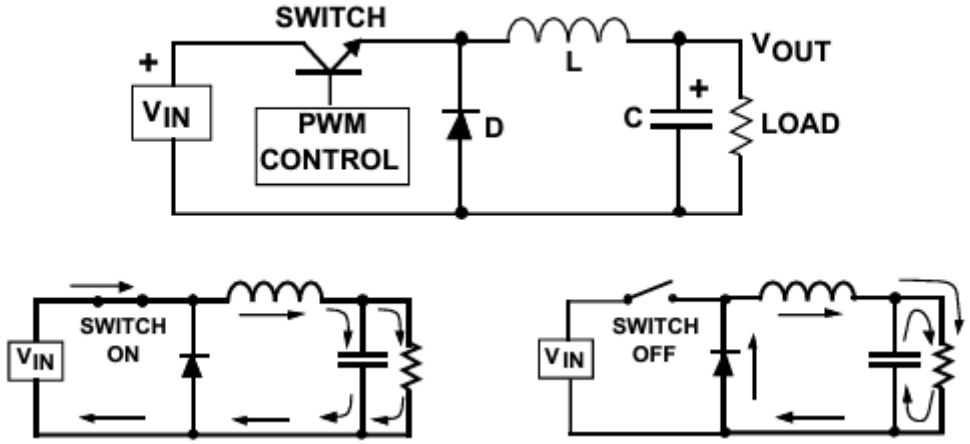
3.5. DC-DC konverter LM2596S

Kako baterijski paket ima maksimalni napon od 16.8 V s napunjenim baterijama, a Raspberry Pi 3B i Slamtech RPLIDAR A1 rade na naponima od 3.3 V i 5 V. Potrebno je smanjiti napom kako navedene elektroničke komponente ne bi pregorjele. Tiskana pločica koja smanjuje napom(Step down module) nije razmotrena u potpunosti zbog manjka tehničke dokumentacije ali provjerena njena funkcija pomoću voltmetra te su dane osnovne upute za rad s njom.



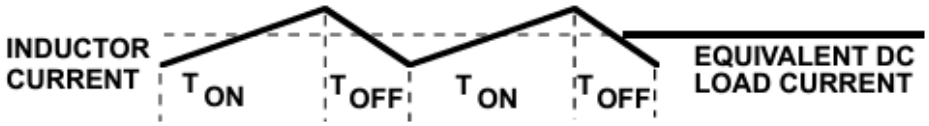
Slika 11. DC-DC konverter

Način rada glavnog čipa LM2596 od firme Texas Instruments na tiskanoj pločici je detaljno opisan u njihovoj dokumentaciji. Način konfiguracije vrijednost izlaznog napona je pomoću potenciometra koji određuje frekvenciju PWM signala paljenja i gašenja izlaznog tranzistora.



Slika 12. Opis rada LM2596 čipa

Zavojnica usporuje nagli porast struje za vrijeme dok je tranzistor(sklopka) uključen. Dok kondenzator služi kao izvor napona i struje trošilu za vrijeme dok je tranzistor isključen. Kao izlazna vrijednost dobije se srednja vrijednost napona i struje ovisno o frekvenciji paljenja i gašenja izlaznog tranzistora i ulaznog napona.



Slika 13. Srednja vrijednost struje trošila

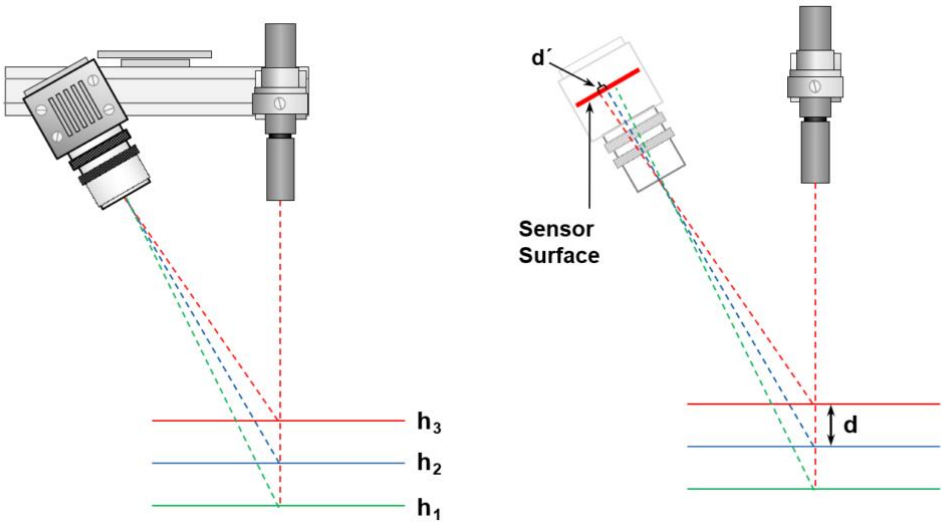
Glavni problem kod korištenja ovakvih DC/DC konvertera su njihove izlazne smetnje. Smetnje mogu izazvati nepravilan rad komponenti koje su spojne na kraju konvertera. Smetnje mogu biti izazvane zbog nedovoljno velike frekvencije prekidanja, te kako bi bio izlazni napon bio dovoljno nizak, konverter produljuje vrijeme isključene sklopke koje kondenzator ne može popratit.

3.6. Slamtec RPLidar A1

Glavni senzor u ovom završnom radu je Slamtec RPLIDAR A1. Njegova uloga je mjerenje točaka udaljenosti jednog presjeka površine u prostoru.

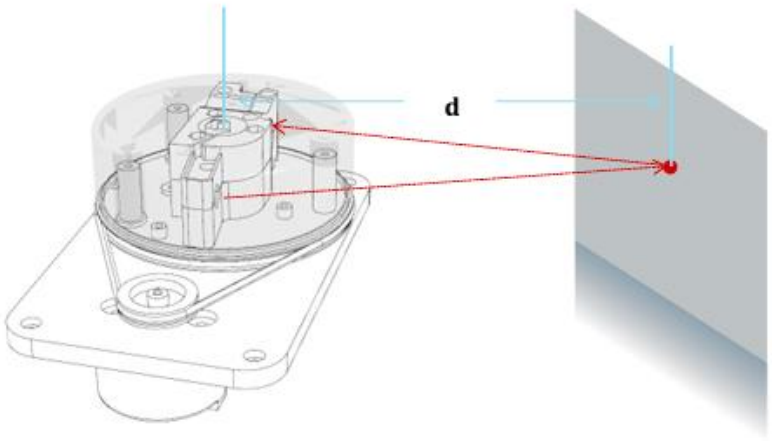
3.6.1. Opis rada RPLIDAR-a A1

RPLIDAR A1 koristi laser valne duljine od 785 nanometara, snage do 5mW i duljine trajanja impulsa od 110 µs. Glavni princip rada ovog senzora je laserska triangulacija. Naime ako su poznate vrijednosti udaljenosti i kutovi gledišta lasera i leće senzora moguće je preko trigonometrije odrediti udaljenost.



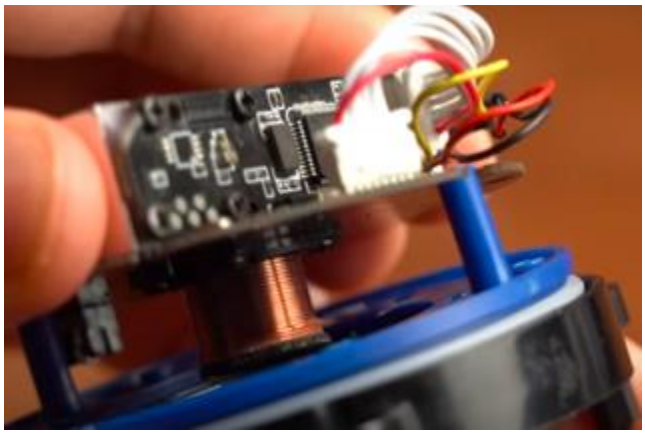
Slika 14. Laserska triangulacija

Ovisno o dijelu površine CMOS senzora kamere koji je osvijetljen refleksijom lasera određuje se udaljenost od senzora do objekta preko sličnosti trokuta. Prednost ovakvog mjerenja udaljenosti je nekoliko. Laserski signal moguće je digitalno kodirati kako bi se se digitalnim procesuiranjem signala moglo ukloniti bilo kakvi šumovi iz okoline na infracrvenom spektru. Cijena samog senzora kamere je puno manja jer nisu potrebni RGB(Red, Green, Blue) filteri boja kao kod klasičnih kamera. Naravno točnost samog mjerenja kao i kod svakog senzora ovisi o nizu faktora pri izradi poput točnosti pozicioniranja udaljenosti i kuta između kamere i lasera, rezoluciji kamere koje se koristi. Boljom rezolucijom kamere, finija je mreža piksela kojom se izračunava udaljenost od predmeta do senzora.



Slika 15. RPLIDAR A1

Sama rotacija ostvaruje se pomoću elektromotora u postolju RP Lidara A1. Preko gumene remenice moment se prenosi s elektromotora na rotacijsku glavu s laserskim senzorom. Način napajanja sklopovlja na rotacijskom dijelu senzora izveden je pomoću zavojnice na rotacijskoj glavi, koja kada rotira oko permanetnog magneta na statoru generira struju za elektroničko sklopovlje na rotacijskoj glavi.



Slika 16. Zavojnica na rotaciskoj glavi Lidara

Jedan od problema koji može nastat ako nije dan dovoljan napon na motoru, glava senzora ne okreće se dovoljnim brojem okretaja u minuti i nije generiran dovoljan napon i struja na glavi senzora. Prednost ovog sustava je što nema nikakvog mehaničkog trošenja između statora i glave rotacije. Komunikacija između statora i rotora ostvarena je pomoću dvije diode i dva fotoosjetljiva tranzistora. Jedan par tranzistora i diode su postavljeni na kućište i drugi par je postavljen na rotacijsku glavu. Dioda služi za slanje podataka a fototranzistor služi za primanje podataka.

Slika 17. Fotodioda i fotoosjetljiv tranzisotor RPLidara

3.6.2. Vrste komunikacije sa RPLidar A1

Prije samog korištenja senzora potrebno je razmotriti načine spajanja senzora na Raspberry Pi 3B. Prvi način je direktnim spajanjem žica prema danoj tablici iz priloga[2] na slici 18. direktno na GPIO pinove Raspberry-a.

Interface	Signal Name	Type	Description	Min	Typical	Max
Motor Interface	5V_MOTO	Power	Power for RPLIDAR A1 Motor	-	5V	9V
	CTRL_MOTO	Input	Enable signal for RPLIDAR A1 Motor/PWM Control Signal	0V	-	5V_MOTO
	GND_MOTO	Power	GND for RPLIDAR A1 Motor	-	0V	-
Core Interface	VCC_5	Power	Power for RPLIDAR A1 Range Scanner Core	4.9V	5V	5.5V
	TX	Output	Serial output for Range Scanner Core	0V	-	5V
	RX	Input	Serial input for Range Scanner Core	0V	-	5V
	GND	Power	GND for RPLIDAR A1 Range Scanner Core	-	0V	V5.0

Slika 18. Tablica pinova RPLidar A1

Drugi način spajanja je pomoću USB adaptera koji pretvara UART[Universal Asynchronous Receiver / Transmiter] komunikaciju u USB[Universal Serial Bus] komunikaciju. U ovom završnom radu koristio se USB adapter.

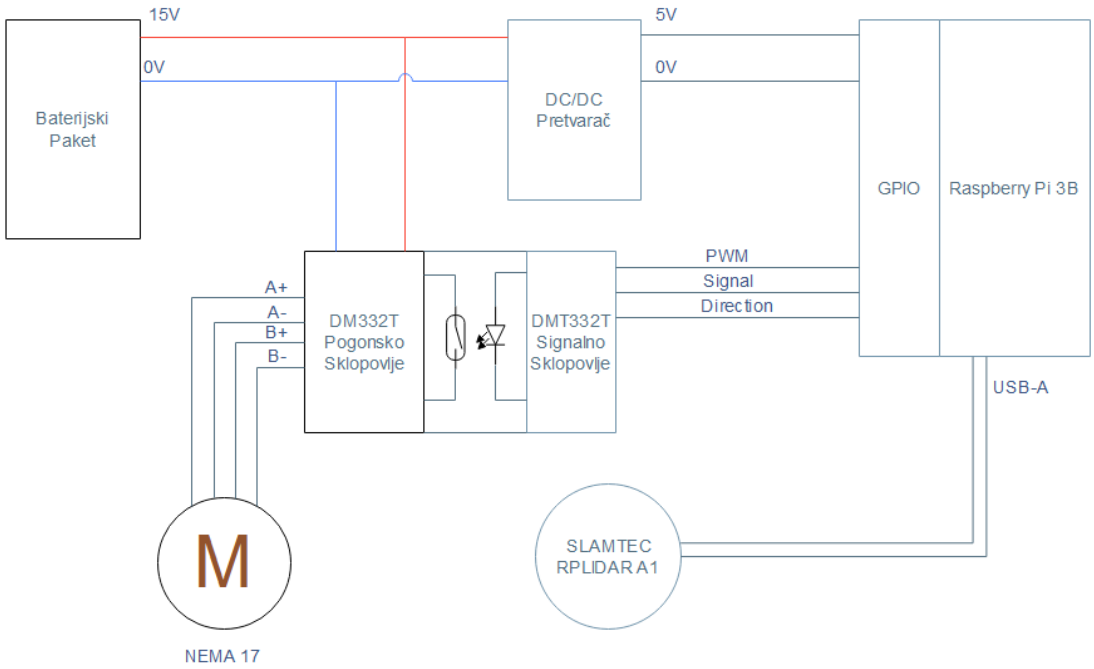
Slika 19. UART na USB adapter

Iako je prednost ove vrste komunikacije lako spajanje na druge uređaje bez GPIO pinova, nedostatak je komunikacija u paketima koje usporavaju komunikaciju.

Dodatno, nije moguća direktna kontrola brzine vrtnje Lidar senzora. Jedan od razloga može biti kako USB komunikacija radi na 5 V dok maksimalna brzina motora je na 9 V, nije ni moguće upravljati brzinom motora ako je tipična brzina vrtnje na 5 V.

3.7. Blok schema spajanja koponenti

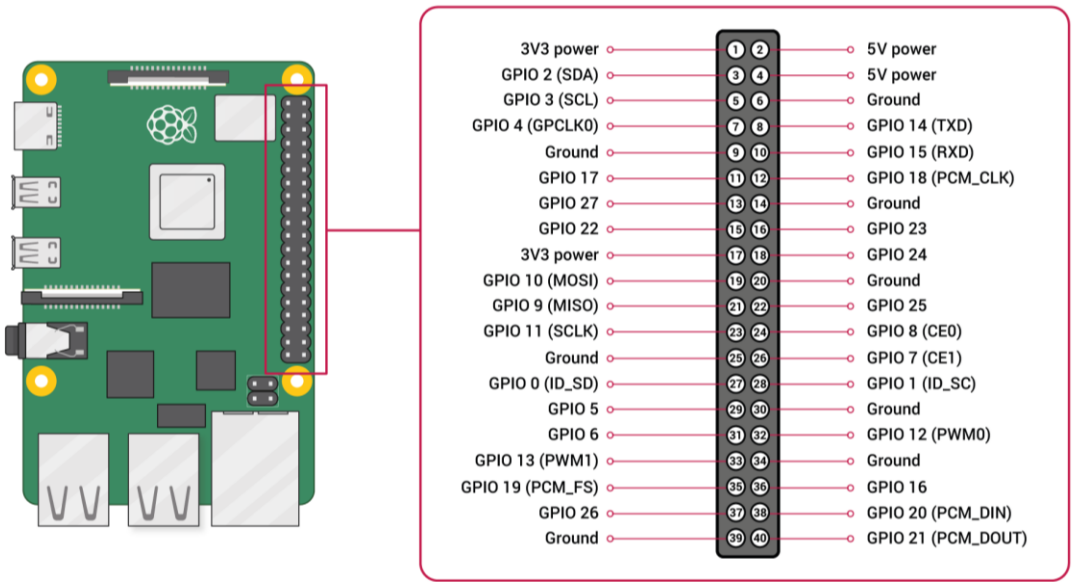
Za lakše razumijevanje cijelog sustava dana je schema spajana elektroničkih komponenti na slici 20. Na baterijski paket spojeni su paralelno DC/DC pretvarač i DM332T Pogonsko sklopovlje. DC/DC pretvarač konvertira napon od 15 V na 5 V s kojim se napaja Raspberry pi, signalno sklopovlje za DMT332T i RPLidar A1 pomoću USB konektora koji je spojen na Raspberry Pi. Treba napomenuti kako raspberry pi nema naponsku zaštitu na GPIO pinovima ali ima na primarnom utoru za napajanje s micro-USB adapterom.



Slika 20. Blok schema spajanja elektroničkih komponenti

Za napajanje raspberry-a koristi se pin 2 kao VCC i pin 6 kao GND. Za upravljanje signalnog sklopovlja koriste se pinovi 38(GPIO-20) i 12(GPIO18) a za napajanje signalnog sklopovlja step kontrolera koristi se pin 4 je direktno spojen sa pinom 2. Zanimljivo je napomenuti kako signalno sklopovlje step kontrolera nema GND žicu.

Na slici 21 moguće je vidjeti GPIO pinove za Raspberry pi.



Slika 21. Raspberry Pi 3B Pinovi

Iako je bilo moguće spojiti signalni pin step kontrole na GPIO 12[Pin 32] i slati PWM(Pulse width modulation) signale, u ovom završnom radu to nije odabrano. Razlog tome je teža izvedba programa za kontroliranje koraka step motora za točno pozicioniranje. Naime, PWM signali na Raspberry Pi-u imaju točno određene frekvencije koje se mogu koristit.

Ako je odabrana neka vrijednost između dvije zadane frekvencije, na prije 6000 Hz, bit će odabrana prva viša frekvencija, 8000 Hz. Popis frekvencija vidljive su na slici 22. Prije pokretanja programa koji koristi step moto potrebno je omogućiti GPIO deamon i specificirati „Sample rate“, u suprotnom će biti odabran „Sample rate 5“

Sample Rate	Hertz								
1:	40000	20000	10000	8000	5000	4000	2500	2000	1600
	1250	1000	800	500	400	250	200	100	50
2:	20000	10000	5000	4000	2500	2000	1250	1000	800
	625	500	400	250	200	125	100	50	25
4:	10000	5000	2500	2000	1250	1000	625	500	400
	313	250	200	125	100	63	50	25	13
5:	8000	4000	2000	1600	1000	800	500	400	320
	250	200	160	100	80	50	40	20	10
8:	5000	2500	1250	1000	625	500	313	250	200
	156	125	100	63	50	31	25	13	6
10:	4000	2000	1000	800	500	400	250	200	160
	125	100	80	50	40	25	20	10	5

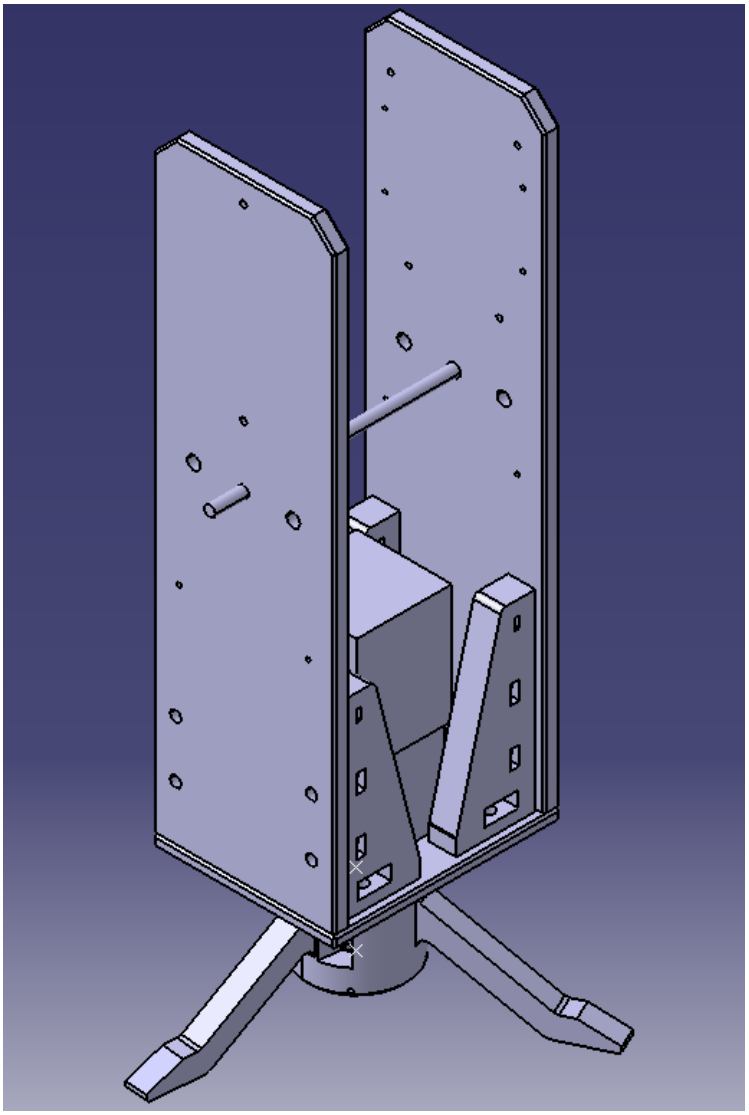
Default sample rate of 5 is set when PiGPIO daemon is started. -s to specify a different rate

Slika 22. Raspberry Pi GPIO deamon.

Tijekom pisanja programa upravljanja step motora u slučaju korištenja PWM pin-a 32 bi bilo potrebno mjeriti vrijeme trajanja PWM signala neke određene frekvencije za točno pozicioniranje izlaznog vratila step motora. Rješenje koje odabrano je prikazano u kasnijem poglavlju koje koristi jednu **for** petlju i za jedan korak step motora pošalje jedan impuls jednim prolaskom kroz petlju, a broj prolazaka ovisi o parametrima step motor kontrolera i željenom pomaku kuta motora vratila.

4. Konstukcija

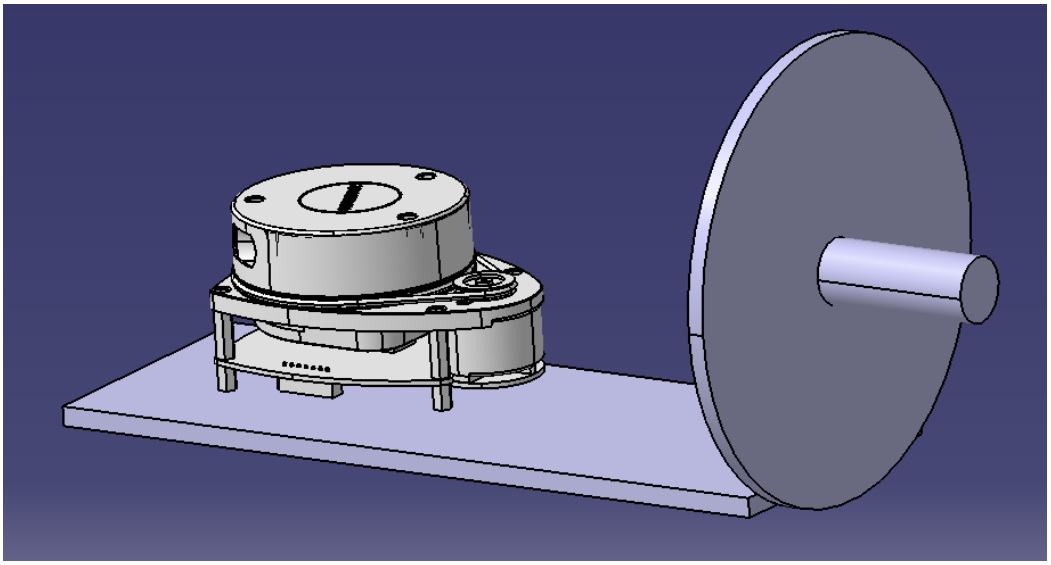
U ovom djelu rada opisana je odabrana dodatna os rotacije, problemi točnosti pozicioniranja senzora te generalno konstrukciji zahtjevi točnosti, manjka vibracija i čvrstoće. Dio elementa konstrukcija izrađena je pomoću PLA(Polylactic Acid) aditivnog printera a za povezivanje elemenata konstrukcije koriste se vijci i matice od Inoxa.



Slika 23. Konstrukcija 3D snimača prostora

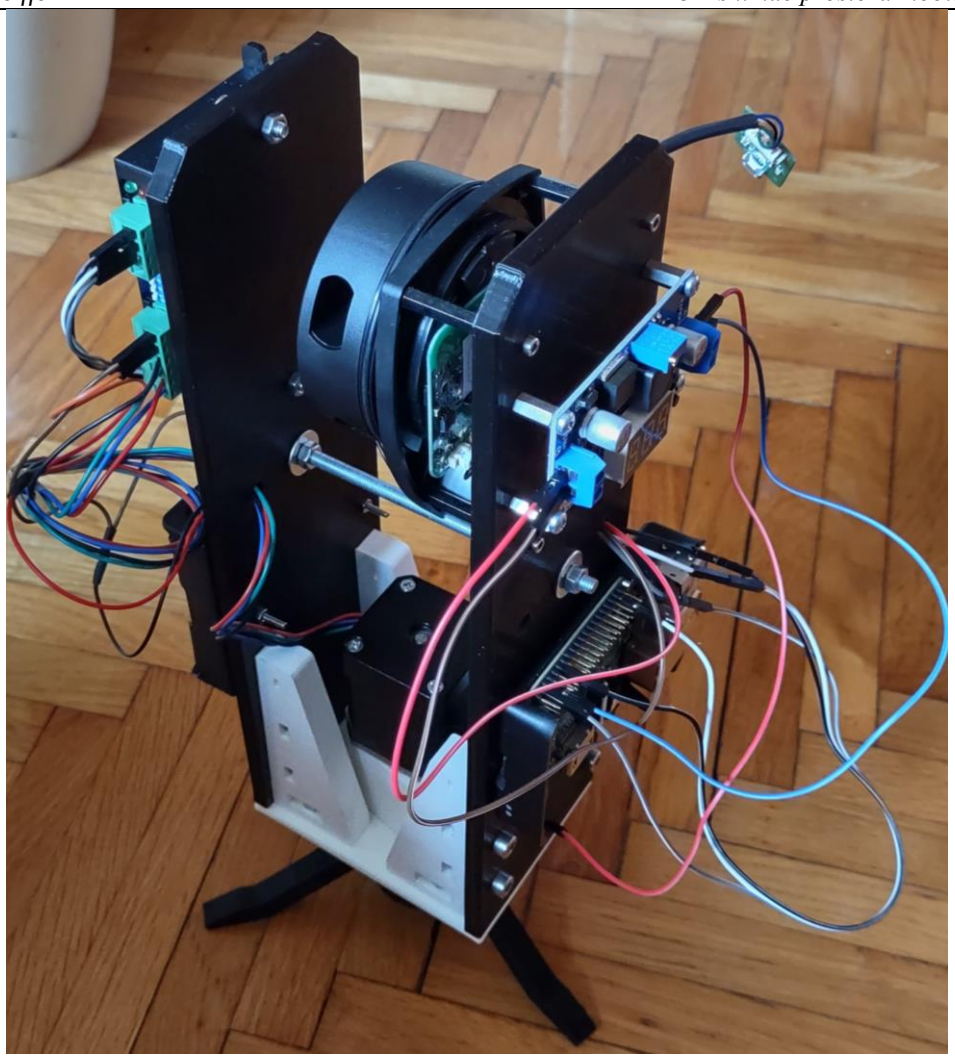
4.1. Odabir dodatne osi

Prvi problem ovog završnog zadatka bilo je određivanje dodatne osi rotacije. Cilj je osmisliti konstrukciju koja je što jednostavnija za izvest i zaklanja minimalni dio senzora za vrijeme skeniranja. Jedna od ideja bila je postavljanje senzora u horizontalnoj (X-Y) ravnini te omogućiti okretanje oko Y osi kao na sljedećoj slici. Dodatno potrebno je naglasiti kako je jedan od odabranih zahtjeva bio ostvariti poklapanje odabrane osi rotacije s 2D ravninom skeniranja RPLIDARA A1. Razlog odabira tog zahtjeva je bila točnost mjerenja te kasnije jednostavnije pretvorbe sfernog u kartezijev koordinatni sustav za vrijeme programiranja. Iako je moguće ostvariti skeniranje prostora s odmaknutim senzorom od dodatne osi rotacije ta ideja bila je odbačena jer se nije mogla iskoristiti visoka točnost pozicioniranja koračnog motora.



Slika 24. Rotacija oko Y-osi konstrukcije

Odabirom zahtjeva o poklapanju osi rotacije s površinom skeniranja, sljedeći zahtjev bio je odabir osi rotacije. Glavni problem je uočljiv na slici 24. je zaklanjanje velikog dijela prostora desno od senzora. Kako bi se riješio taj problem bilo je potrebno sagledati koji dio prostora je najpoznatiji odnosno koji dio prostora je najmanje važan za skeniranje za vrijeme rada mobilnog robota. Glavna ideja senzora bila je mogućnost montiranja na vrh mobilnog robota. Omogućavanjem rotacije oko Z-osi te postavljanjem LIDAR-a u što višu točku mobilnog robota s površinom skeniranja X-Z ili Y-Z ovisno o kutu rotacije Z-osi, zaklanja se samo dio prostora ispod senzora, odnosno dio mobilnog robota.

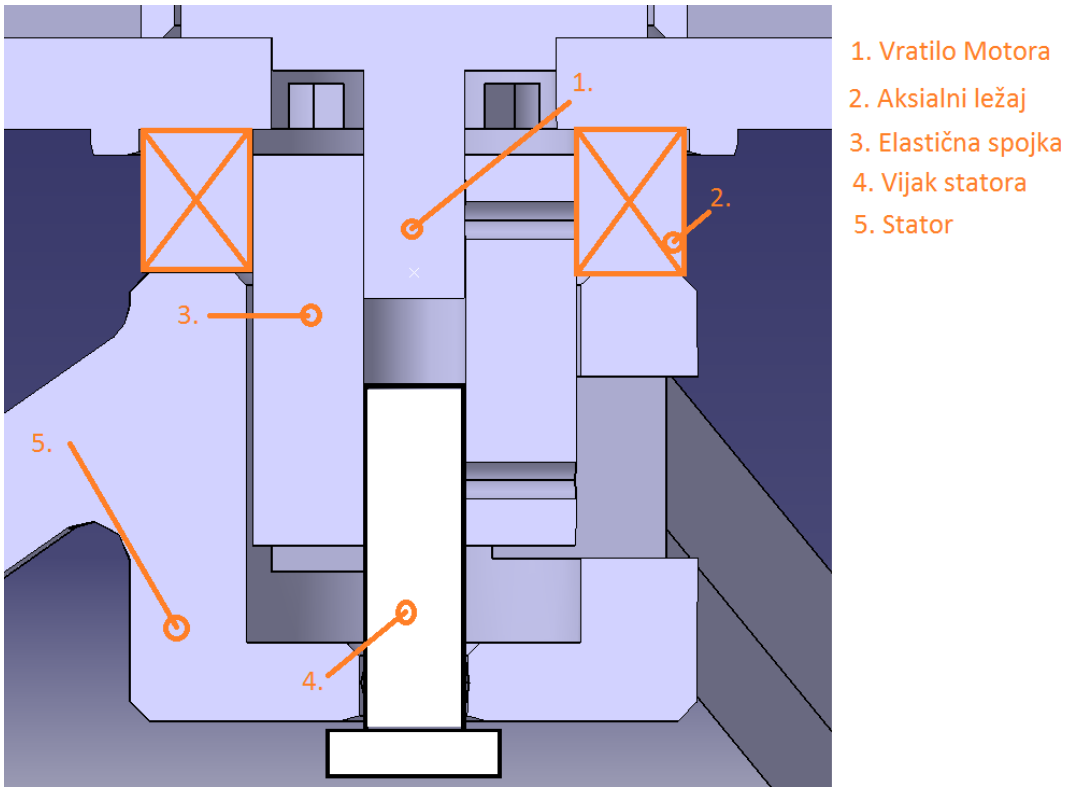


Slika 25. Finalna konstrkcija 3D snimača prostora

4.2. Obrazloženja nekih konstrukciskih odluka

Prilikom osmišljavanja konstrukcije odlučeno je će se motor nalaziti na glavi rotacije. Razlog tome je bila mogućnost beskonačnog broja okretaja oko svoje osi bez smetnje od zapetljavanja kablova napajanja između statora i rotora. Dodatno je zamišljeno da sve komponente kako se nalaze na glavi rotacije mogu biti skinute s mobilnog robota i stavljene stalak veće visine ako je potrebno.

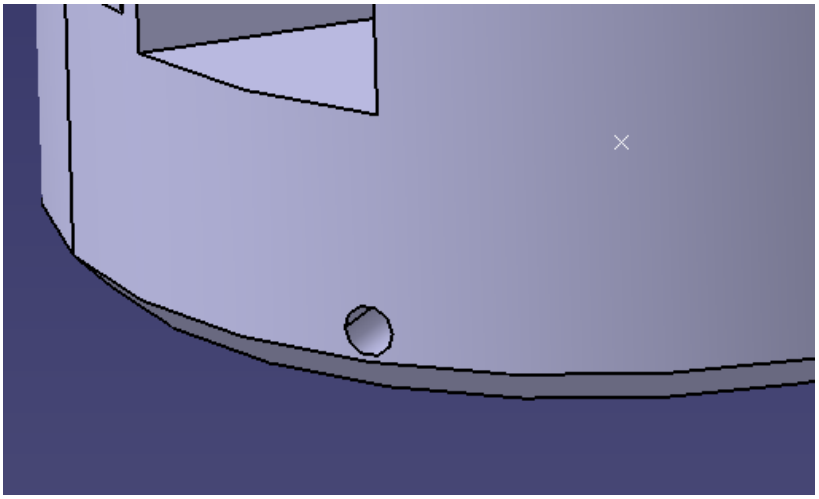
4.3. Postolje konstrukcije



Slika 26. Postolje konstrukcije

Kako bi se osigurao pravilan rad senzora i točno mjerenje, jedan od bitnijih dijelova konstrukcije je bio stator, odnosno postolje. Na slici 26. označeni su neki od najbitnijih dijelova. Aksijalni ležaj omogućuje prijenos sile izazvan težinom glave rotacije da se prenosi na postolje konstrukcije, u protivnom cijela težina bi pada na vratilo koračnog motora. Elastična spojka omogućuje prijenos momenta s vratila rotora na vijak statora i dodatno sprječava bilo kakve aksijalne udarce u smjeru Z-osi. Elastična spojka koristi dva pužna vijka za pritezanje vijka na statoru i vratila na rotoru.

Posljednje, vijak statora omogućuje prijenos momenta s elastične spojke na postolje. Prijenos momenta se ostvaruje trenjem, stezanjem vijka(4) i matice(koja nije ucrtana na slici 26.) u statoru, a kao dodatno osiguranje koristi se vijak koji onemogućuje okretanje vijka(4) oblikom. Dio navoja vijka(4) uklonjen kako bi se dobila slična površina kao kod vratila sa utorom za pero. Rupa za vijak koji onemogućuje okretanje oblikom prikazana je na sljedećoj slici.



Slika 27. Rupa osiguranje oblikom

5. Programski dio 3D snimača prostora

U ovom dijelu završnog rada opisan je rad softwareskog dijela završnog rada, kratki osvrt na programe koji se koriste, kratki opis instalacije operativnog sustava na Raspberry Pi, biblioteke koje se koriste i najbitniji dijelovi koda koji se koriste. Detaljni opis rada koda sa komentarima nalazi se u dodacima završnog rada.

Za pokretanje Raspberry Pi 3B prvo je potrebno instalirati operativni sustav Rasbian. Instalacija se izvodi pomoću programa Raspberry Pi Imager. Kako Raspberry Pi koristi SD karticu za primarnu boot particiju i spremnik podataka, preporuka je koristi SD kartice veće od 16Gb jer sam Raspbian operativni sustav koristi 4Gb prostora kada je raspakiran. Nakon „Flashanja" SD kartice i prvog pokretanja(koja traje nešto dulje) potrebno je ažurirati operativni sustav sa sljedeće navedenim naredbama u terminalu.

sudo apt update

sudo apt upgrade

Za vrijeme izrade završnog rada koristio se VNC server na Raspberry-u i VNC viewer na osobnom računalu. Prije početka programiranja preporučeno je napraviti virtualnu okolinu za programiranje kako ne bi došlo do kolizije različitih programskih biblioteka ili različitih verzija istih biblioteka ako se koriste. Virtualna okolina sprema programske biblioteka u zasebne cjeline, stvaranje i pokretanje virtualne okoline moguće je u Thonny editoru ili u terminalu. Nakon što je stvorena virtualna okolina potrebno je dodati programske biblioteke pomoću sljedećih naredbi u terminalu. Programski jezik koji se koristi u ovom završnom radu je Python.

pip3 install RPi.GPIO

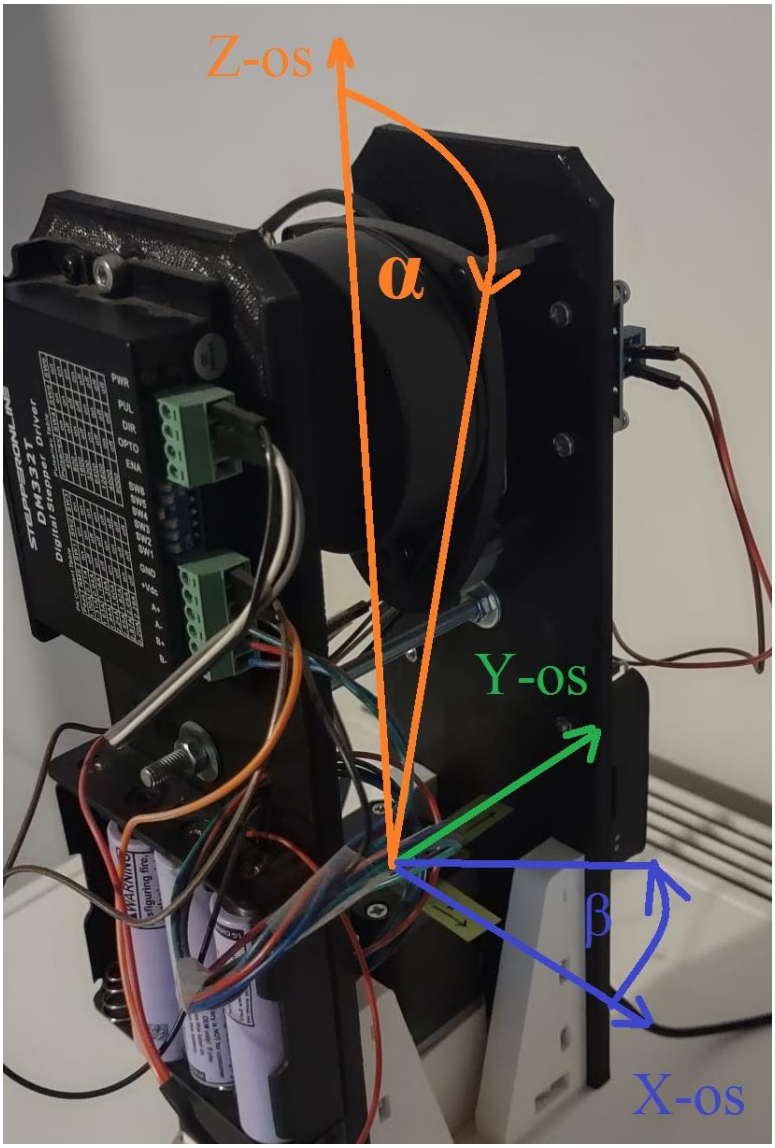
pip3 install adafruit-circuitpython-rplidar

pip3 install numpy

RPi.GPIO programska biblioteka omogućuje korištenje GPIO pinova na raspberry-u kako to nije standardno dostupno, *adafruit-circuitpython-rplidar* koristi se za uspostavljanje USB komunikacije i pretvorbu podataka u korisne informacije. Numpy programska biblioteka koristi za lakšu manipulaciju matrica(python array-a). „*Pip 3*" naredba označuje instalaciju biblioteke za programski jezik python verzije 3 nadalje.

5.1. Programiranje Python koda

U ovom dijelu rada objašnjavaju se najbitnije funkcije python koda ovog završnog rada. Napisan program služi za prikupljanje podataka s lidar senzora na jednoj ravnini, kada je zadovoljen broj točaka na jednoj ravnini, 3D skener se rotira oko Z osi te skenira novu ravninu. Kada je zadovoljen broj točaka na svim ravninama, zaustavlja se petlja skeniranja i rotacije motora. Lista svih validnih udaljenosti, alfa kutova i beta kutova zapisuju se CSV datoteku. Kako osmišljeni 3D skener koristi Sferni koordinatni sustav(2 kuta i jednu udaljenost) potrebno je odrediti koordinatni sustav za lakšu vizualizaciju prostora tijekom programiranja.



Slika 28. Zamišljeni kordinatni sustav programa

Bitno je naglasiti da moguće i odabrati drugačiji koordinatni sustav ali u tom slučaju potrebno je voditi računa o desnokretnom koordinatnom sustavu. Kako Lidar senzor mjeri Alfa kut od vrha Z osi u kasnijem dijelu programa će se koristi kosinus kuta kako bi se odredila visina Z osi a sinus kuta za izračun dijagonalne na x i y ravnini.

5.1.1. Programiranje Lidar senzora

Za dodavanje programske biblioteke za komunikaciju s lidarom koristi se sljedeća naredba.

```
from adafruit_rplidar import RPLidar
    from adafruit_rplidar import RPLidarException
```

Zbog čestih komunikacijskih problema s Lidar senzorom dodaje se iznimka (Exception). Uloga ove iznimke u kodu je ponovno pokretanje komunikacije sa senzorom u slučaju komunikacijske greške, u suprotnom izvođenje programa se prekida i potrebno je ponovno „ručno pokretanje koda“.

Dodatno potrebno je kreirati 3 array-a u koje će se spremati podaci udaljenosti i dva kuta. Prednost neograničenih spremnika je što se može upisati „beskonačan broj“ podataka. Nedostatak je to što memorija nije unaprijed alocirana za spremanje podataka što može dovesti duljeg izvođenja programa pri ispisu podataka.

```
send_data_rng=[]
send_data_alf=[]
send_data_bet=[]
```

Klasa Rplidar() započinje komunikaciju sa senzorom.

```
PORT_NAME = '/dev/ttyUSB0'
Lidar = RPLidar(none, Port_name, timeout=7)
```

Za prikupljanje podataka koriste se dvije for petlje. Podaci sa senzora dolaze u paketima, svaki paket zapisan je u obliku array-a unutar kojeg se nalaze u prosjeku od 140 do 200 array-a, a unutar svakog tog array-a nalaze se tri podatka, jačina vraćenog signala, kut alfa i udaljenost.

[....[signal1, kut1, udaljenost 1][signal2, kut 2, udaljenost 2]....]

U ovoj verziji program zapisuju se točke koje se nalaze „ispred“ senzora, odnosno u pozitivnom smjeru X-osi pomoću if funkcije koja zapisuje točke s kutom većim od 15° i manjim od 165°. Glavni dio python programa za prikupljanje podataka zapisan je u sljedećim linijama koda.

```
for scan in lidar.iter_scans():
    for(_, angle, distance) in scan:
        if distance != 0 and angle >= 15 and angle <= 165:
            data_pk=data_pk+1
            count=count+1
            send_data_rng.append(distance)
            send_data_alf.append(round(angle*2/2))
            send_data_bet.append(beta)
```

Dodana su dva brojila, brojilo data_pk kada je popunjeno pokreće podfunkciju za upravljanje koračnog motora. Brojilo count zaustavlja prikupljanje podataka i pokreće podfunkciju za spremanje podataka u CSV datoteku i kasnije bash program za slanje podataka natrag na osobno računalo.

5.1.2. Programiranje koračnog motora

U ovom dijelu program koriste se dvije programske biblioteke. RPi.GPIO omogućuje manipulaciju input/output pinovima dok sleep funkcija iz time biblioteke se koristi za simulaciju PWM signala.

```
import RPi.GPIO as gpio
from time import sleep
```

Za odabir pinova koji će se koristit za pokretanje step motora zapisuju se dvije variable, DIR(direction) i STEP(signalna linija). Dodatno upisuju se dvije variable CCW i CW koje određuju smjer okretanja 3D skenera.

```
DIR = 20
STEP = 18
CCW=0
CW=1
```

Ako se Raspberry GPIO pinovi koriste s drugog računala potrebno je omogućit „Remote GPIO“ u postavkama raspberry-a. Nadalje, potrebno je odrediti koji će se pinovi koristi u ovom programu i hoće li se koristi za input ili output.

gpio.setmode(gpio.BCM)

gpio.setup(DIR, gpio.OUT)

gpio.setup(STEP, gpio.OUT)

gpio.output(DIR, CCW)

Linija koda gpio.setmode() koristi se određivanje dogovora zapisa pinova, BCM mod referira se na GPIO pinove kada je upisan broj pin-a u gpio.setup(). Drugim riječima, BCM pin 20 isti je kao i BOARD pin 38(referirati se na sliku 21.). Jednom kada se odredi mod zapisivanja pinova u programu mogu se zadati pinovi koji se koriste pomoću naredbe gpio.setup().

Glavna uloga ovog dijela programa je pokretanje step motora za određeni broj stupnjeva, odnosno koraka. Potrebno je odrediti koliko je impulsa potrebno za željeni pomak u stupnjevima. Za ovaj program odabran je odmak između svake skenirane ravnine od 5°. Dodatno na tablici kontroler step motora odabrano je 800 impulsa za jednu punu revoluciju, kako je dodan reduktor s redukcijom 1:100, potrebno je 80000 impulsa za punu revoluciju.

$$p = \frac{80000 \cdot 5°}{360°} = 1111.11\dot{1} \tag{3}$$

Uzimajući u obzir grešku koja nastaje zbog decimalnog zareza, for petlja će se izvodit 1111 puta. Greška koja nastaje jer se impuls ne može poslat u decimalnom obliku može se izračunati na sljedeći način.

$$\Delta\beta = \frac{0.11\dot{1} \cdot 360°}{80000} = 4.955° \cdot 10^{-3} = 17.9982' \tag{4}$$

Potrebno je naglasiti kako ova greška se akumulira od 18 sekundi svakom promjenom ravnine skeniranja. Već nakon 201 promjena ravnine dolazi do greške od jednog stupnja. Ovakva greška može se ukloniti na nekoliko načina. Odabirom odgovarajućeg pomaka β kuta između svake ravnine, odabirom odgovarajućih maksimalnih β kutova. Kada motor dođe do maksimalne vrijednosti β kuta, smjer vrtnje se mijenja. Dodatno, odabirom drugačijih vrijednosti

mikrokoraka za punu rotaciju step kontrola. Finalno, greška se može smanjiti programski. Potrebno je uvesti variable koje će se koristi za izračun impulsa u programu i zadati kašnjenje za promjenu STEP pin-a iz visokog stanja u nisko i obratno koje iznosi 30 milisekundi.

SPR = 80000

RIS = 3

delay = 3/(100000)

Variable SPR(Steps per rotation) označava signals per revolution i RIS(razmak između skena) označava razmak između skenova. Za stvaranje impulsa na GPIO pinovima za step kontroler koristi se funkcija sleep(). Dodatno koriste se dvije if funkcije za provjeru maksimalnog kuta. Ako je if funkcija zadovoljena, dolazi do mijenja predznaka za kasnije računanje beta kuta i promjena smjera vrtnje motora.

```
if beta == -30:
        dodatak = RIS
        smjer = CCW
if beta ==30:
        dodatak =-RIS
        smjer = CW
beta = beta + dodatak
gpio.output(DIR,smjer)
```

Iz ovog dijela programa vidljivo je da greška nastaje u liniji koda beta=beta + dodatak. Varijabla beta upisuje se u array koji se kasnije šalje za prikaz oblaka točaka varijabla dodatak uvijek je ista i neovisna o broju impulsa koji su poslani na step motor kontroler.

```
for pulse in range(int(SPR*RIS/360)):
        gpio.output(STEP, gpio.HIGH)
        sleep(delay)
        gpio.output(STEP.gpio.LOW)
        sleep(delay)
```

Ovom for petljom ostvaruju se impulsi potrebni za pokretanje step motora, za vrijeme izvođenja ove petlje podaci s Lidar senzora se ne upisuju. Funkcija gpio.output() na prvom mjestu variable određuje GPIO pin na koji se referira (gpio pin 18) a druga varijabla određuje željeno stanje (LOW ili HIGH). Jednim prolaskom kroz ovu petlju ostvaruje se jedan impuls.

5.1.3. Programiranje CSV datoteke

Kako bi se podaci mogli slati s jednog računala na drugo, u slučaju ovog završnog rada s raspberry-a na osobno računalo koristi se CSV(Comma-separated values) zapis podataka. Kada se prikupi zadovoljavajući broj podataka s 3D skenerom, podaci s spremaju i ovisno o vrsti programa, skeniranje se nastavlja ili prekida. U slučaju mobilnog robota bi se podaci spremili i poslali, zatim bi se dio programa za prikupljanje podataka nastavio. Za pozivanje programske biblioteke koristi se sljedeća naredba.

import csv

Kada je zadovoljen broj točaka u programu koristi se **if** funkcija za pokretanje potprograma za spremanje podataka.

filename = „real_time_scan.csv“

with open(filename, 'w') as csvfile:

Potrebno je imenovati datoteku koja će spremati, ako već postoji datoteka pod tim imenom, bit će zamijenjena, ako ne postoji, bit će kreirana. Druga varijabla *'w'* označava *write* da će u datoteku biti upisane vrijednosti. Dodatno, moguće je kreirati svaki put novu CSV datoteku manipulacijom strig variable *filename.* U slučaju kada bi htjeli češće slanje podataka ili automatizirat spremanje svake skenirane ravnine u posebnu datoteku. Za zapis podataka koristi se sljedeća klasa.

csvwriter = csv.writer(csvfile)

csvwreiter.writerows(zip(send_data_rng, send_data_alf, send_data_bet))

Naredba *writerows* upisuje podatke u redove. Kako su podaci zapisani u liste, kada bi se zapisivali u CSV datoteke bili bi zapisani u jednom redu i više stupaca, problem nastaje kako CSV datoteke podržavaju oko 16 tisuća stupaca i oko 1 milijun redova. Zato se koristi funkcija *zip* koja spaja sve tri liste podataka ovisno o poziciji u jednu listu. Za lakši opis dan je primjer.

Prije zip funkcije:

rng =[a, b, c...]

alf=[x, y, z...]

bet=[1, 2, 3...]

Nakon zip funkcije:

Podaci=[[a, x, 1] [b, y, 2] [c, z, 3]]

Stvorena je nova lista podataka i unutar te liste za svaki stupac spajaju se vrijednosti istog rednog broja u podlistu koje dijeli zarez. Naknadno se svaka ta podlista upisuje u novi red pomoću funkcije *writerows()*.

Na sljedećoj slici prikazan je zapis podataka u Microsoft Excel-u.

	A	B
1	1074.25,15.5,0	
2	1081.5,17.0,0	
3	1091.25,18.5,0	
4	1099.75,20.0,0	
5	1107.75,21.0,0	
6	1119.0,22.5,0	
7	1130.25,24.0,0	
8	1141.75,25.5,0	
9	1155.25,27.0,0	
10	1171.75,28.0,0	

Slika 29. Prikaz podataka CSV datoteke

Svaki red predstavlja jednu točku point cloud-a. Prva vrijednost je udaljenost, druga vrijednost α kuta i treća vrijednost β kuta, a svaku vrijednost dijeli zarez(Comma).

5.2. Programiranje Matlab koda

Za prikaz dobivenih podata koristi se računalni program Matlab. Iako je moguće prikazati u drugim programskim jezicima i programima, Matlab već ima ugrađene funkcije za kvalitetan prikaz i manipulaciju oblaka točaka. Nadalje Matlab može koristi i SLAM(Simultaneous Location and mapping) biblioteke ali u ovom završnom radu nisu korištene jer su te biblioteke uvedene u verziji 2020 godine, dok u ovom završnom radu koristi verzija 2018 godine.

5.2.1. Učitavanje CSV datoteke

Prije samog prikaza točaka potrebno je učitati podatke u program. Ako se datoteka s podacima ne nalazi u istom folderu, potrebno je navesti path uz ime datoteke.

```
data = csvread(,,real_time_scan.csv")
range = data(:, 1)
alfa = data(:, 2)
beta=data(:, 3)
```

Nakon što su učitani podaci pod varijablom data koja je zapisana u obliku matrice (i,j=3), u variable range, alfa, beta spremaju se podaci u obliku stupaca. Dodatno potrebno je imenovati 3 variable x,y,z u koje će spremati novo transformirane vrijednosti iz sfernog u kartezijev koordinatnog sustava X,Y,Z.

```
x=[]
y=[]
z=[]
[rowb,colb]=size(beta);
```

Ovo su ne definirane veličine matrice kako bi program funkcionirao neovisno o veličini CSV datoteke. Funkcija size(beta) vraća vrijednosti veličine matrice beta koja se kasnije za for petlju transformacije.

```
For j=1:rowb
        z(end+1)=range(j)*cos(alfa(j)*pi/180);
        diag=range(j)*sin(alfa(j)*pi/180);
        y(end+1)=diag*sin(beta(j)*pi/180);
        x(end+1)=diag*cos(beta(j)*pi/180);
end
```

Broj prolazaka kroz for petlju ovisi o broju redova matrice CSV datoteke. Svakim prolaskom kroz petlju transformira se jedna točka. Prilikom programiranja prve verzije ovog programa x i y vrijednosti bile su zrcaljenje. Kasnijom promjenom sinusa i kosinusa za x i y vrijednosti, greška je bila uklonjena.

Zapis točaka u obliku oblaku točaka izvodi se pomoću funkcija point cloud. Prije toga potrebno je zapisati koordinate x,y,z u jednu matricu.

```
xyz=[x(:) ,y(:) ,z(:)]
ptCloud=pointcloud(xyz)
```

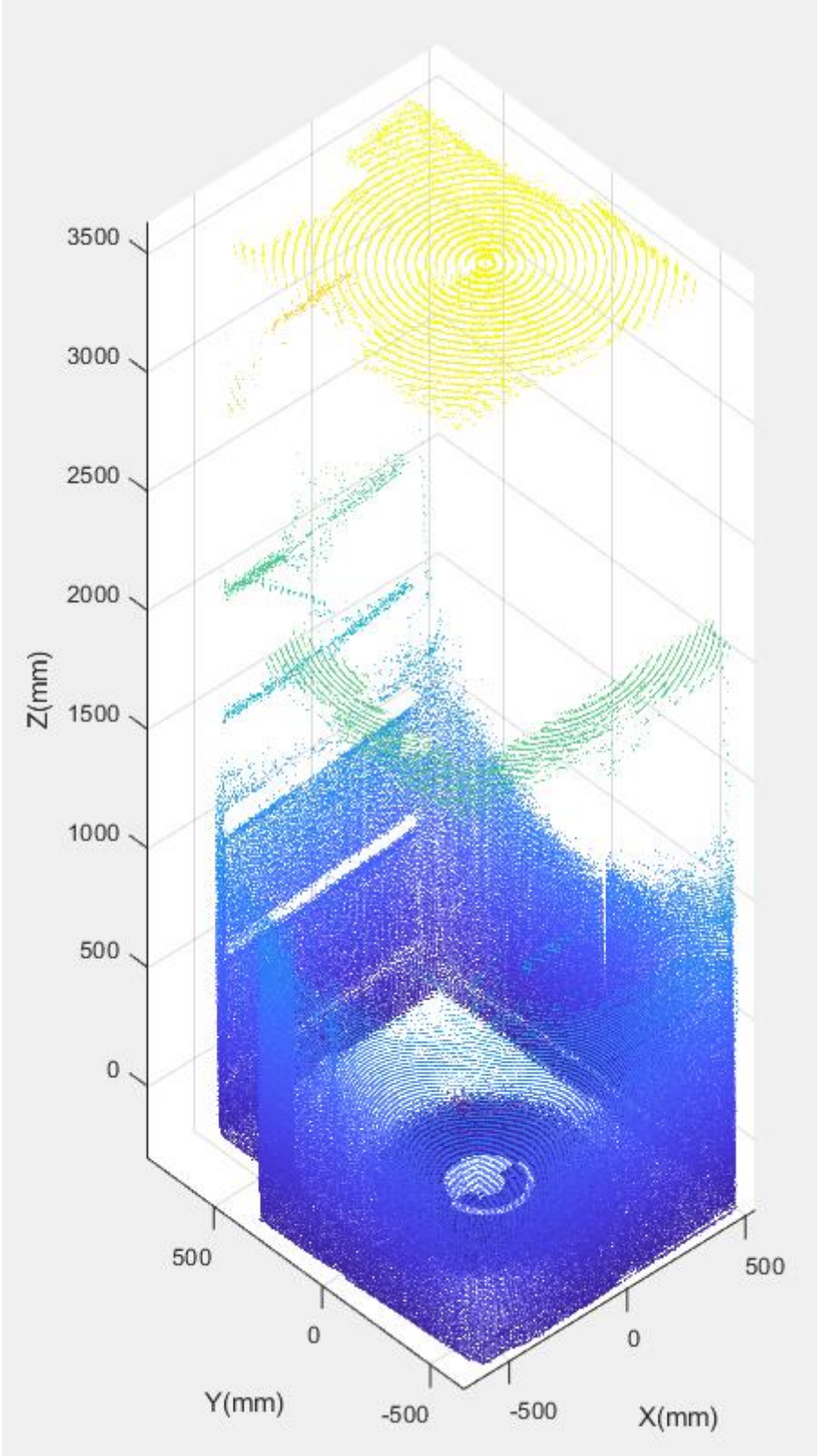
Na kraju za prikaz oblaka točaka koristi se sljedeća naredba.

```
pcshow(ptCloud)
```

Prije pokazivanja jednog rezultata skeniranja u oblaku točaka dana je slika skeniranog prostora na slici 30. Prostor se nalazi u hodniku zgrade, sastoji se od 3 vrata koja su bila zatvorena za vrijeme skeniranja i gabaritnih izmjera [1160x1380x3550]. Ovakvo detaljno skeniranje nije namijenjeno za mobilne robote već s pokazivanje mogućnosti ovakvih senzora. Trajanjem oko 6 minuta, razmakom od četvrtine stupnja između ravnina skeniranja i ukupnim brojem točaka od 670 tisuća.

Slika 30. Prostor skeniranja iz tri perspektive

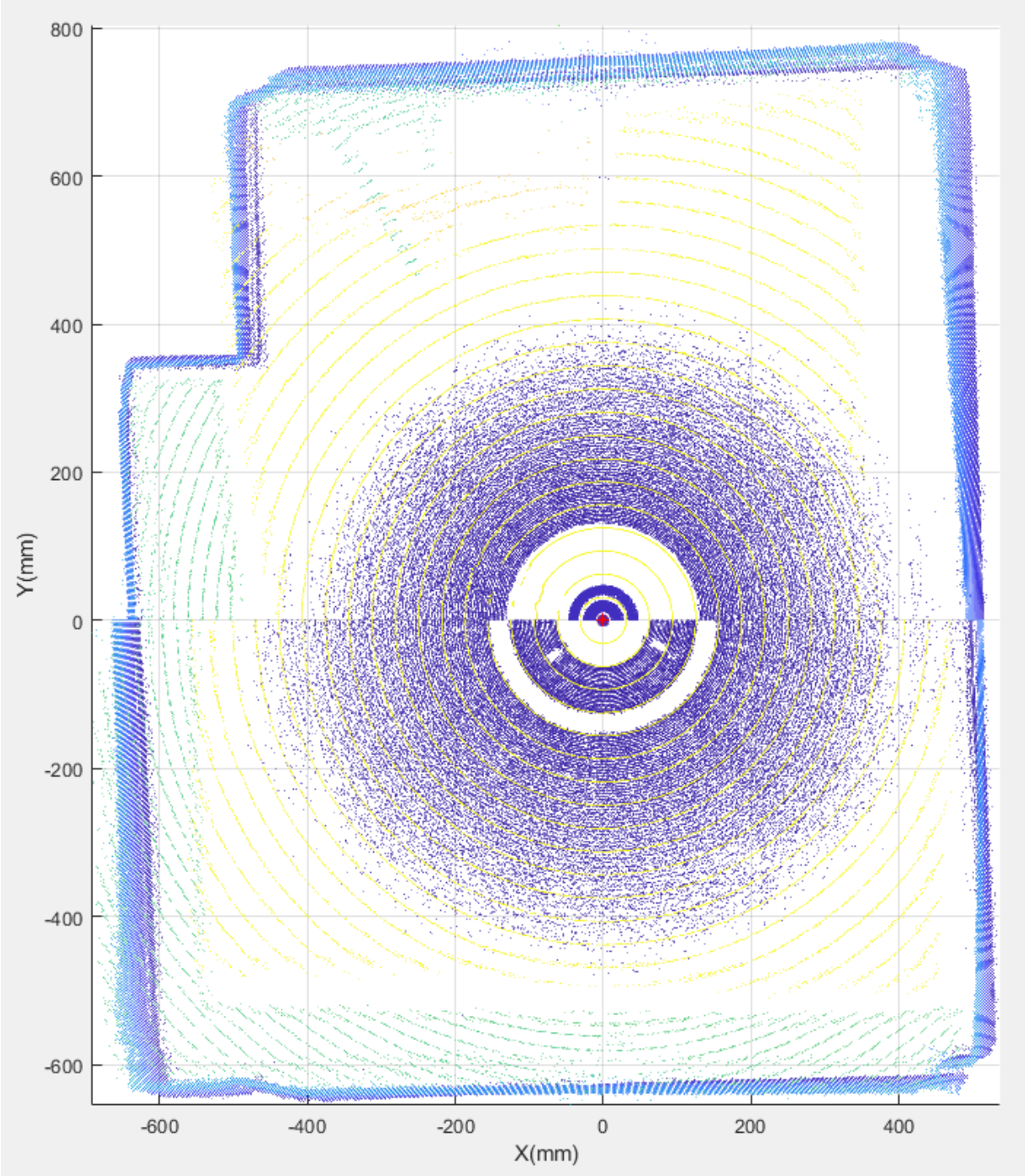
U Matlabu programu moguće je rotirati i manipulirati oblakom točaka pomoću miša, te se lakše orijentirati u danom prostoru. Sljedeće tri slike su dane iz različitih perspektiva oblaka točaka.



Slika 31. Oblak točaka skeniranog prostora

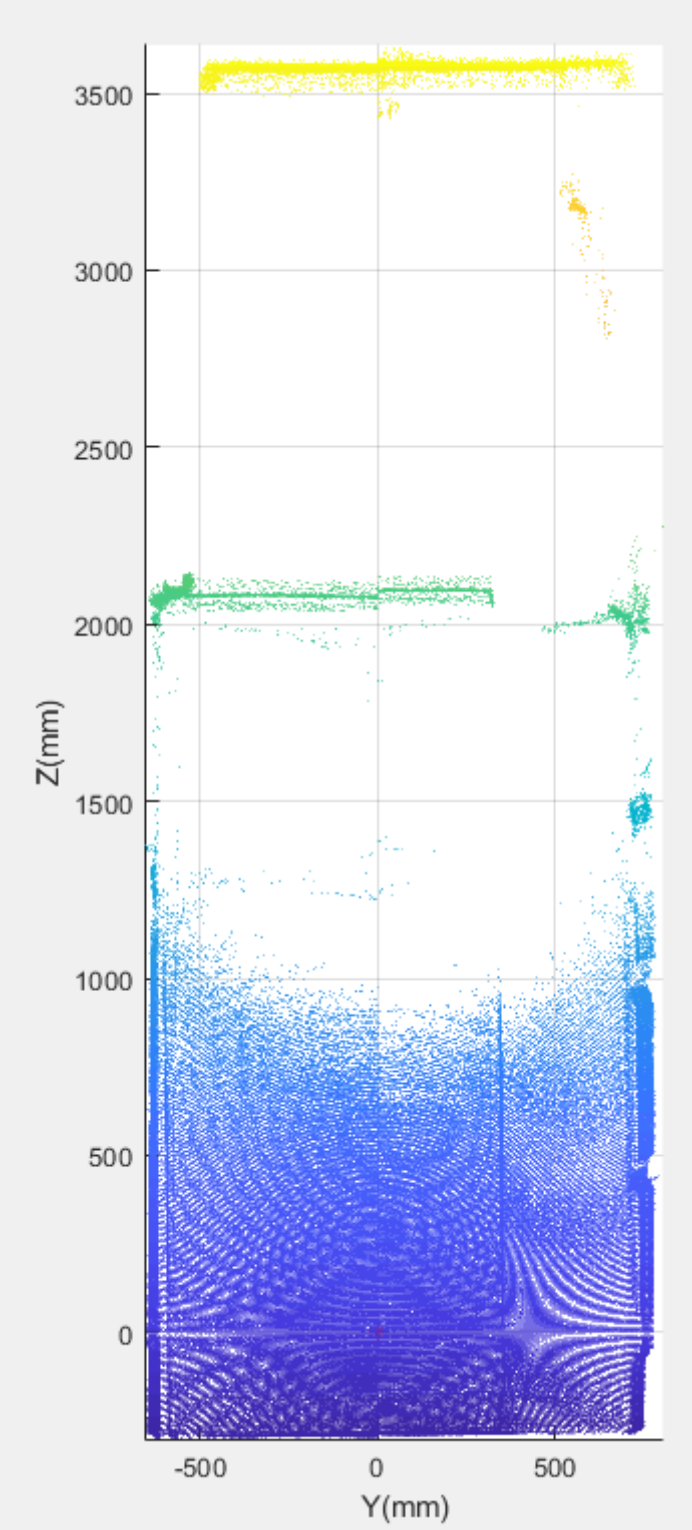
Na slici 30. nije prikazan u potpunosti prostor, poput stropa, okvira vrata i prozora koji se mogu uočiti na slici 31. Zelenom bojom, na visini oko 2000 mm vidljivi su uvučeni okviri vrata,

narančastom bojom na visini 3000 mm vidljiv je obris otvorenog prozora i na visini 3500 mm žutom bojom označen je strop prostorije.



Slika 32. Oblak točaka X-Y ravnivne

Crveni križić označava ishodište koordinatnog sustava senzora. Na slici 32. vidljiv je uzorak snimanja. Koncentrični krugovi imaju puno veći broj točaka po kružnici nego između kružnica. Razlog je minimalni mogući kut pomaka Alfa i Beta kuta senzora. Minimalni alfa kut između lidara i Z-osi je 0.5° a minimalni kut Beta, pomak step motora je 0.225°.



Slika 33. Oblak točaka Y-Z ravnine

Na slici 33. može se uočiti nedostatak točaka na visini iznad 1000 mm. Pretpostavlja se gubitak točaka zbog prevelikog upadnog kuta laserske zrake, ne reflektira se dovoljno laserske zrake s površine zida natrag u Lidar senzor. Kada bi se senzor stavio na visinu od 1000 mm, oblak točaka bio bi popunjeni.

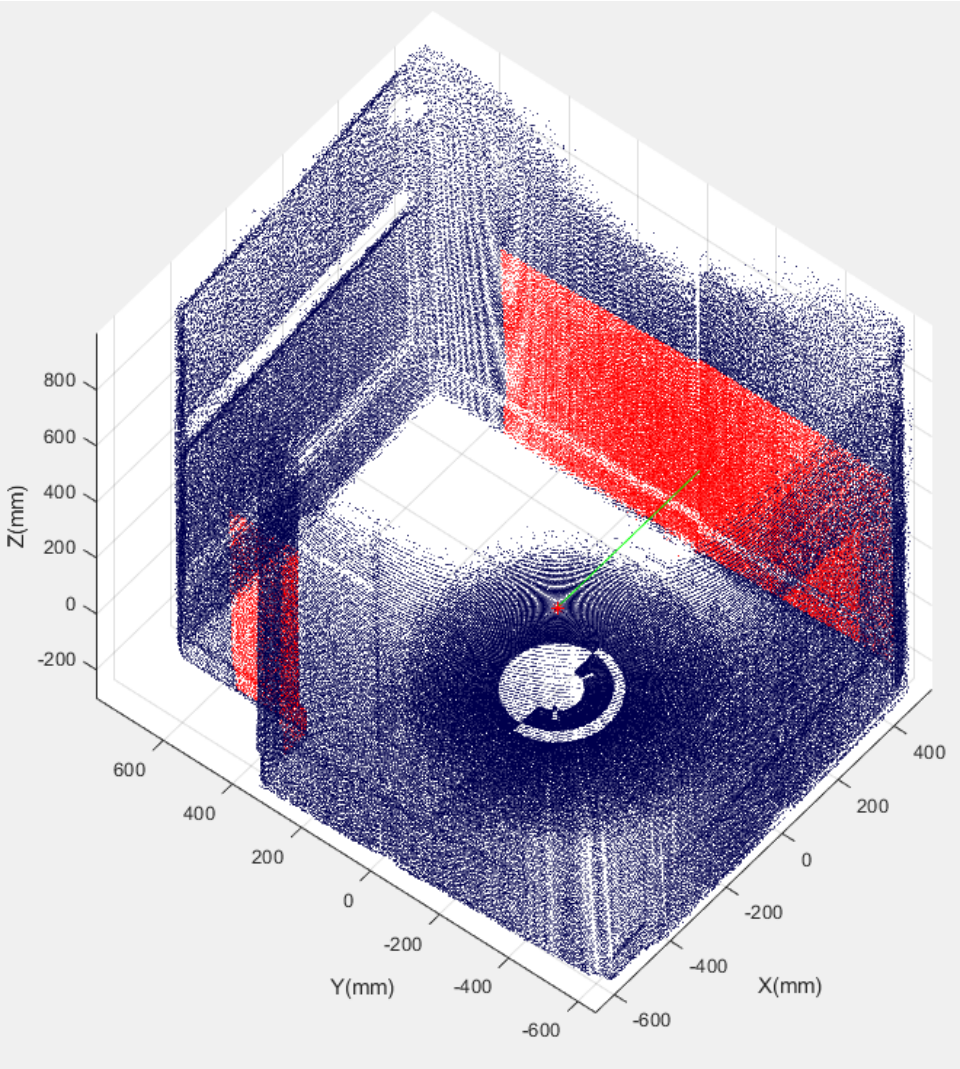
5.2.2. Lociranje smjera kolizije pomoću regije interesa

U snimljenom oblaku točaka potrebno je odrediti moguće točke kolizije. Za takav proces koristi se matlab funkcija findPoIntinROI(). Pomoću ove funkcije moguće je prikazati određeni dio oblaka točaka ili ispisati koordinate za kasnije korištenja algoritme izbjegavanja prepreka. Prvo je potrebno odrediti vrijednosti minimalnih i maksimalnih vrijednosti x,y,z koordinata unutar regije interesa.

roi3=[-550 550 -550 550 -270 400];

indices3=findPointsInROI(ptcloud, roi3);

Varijabla indices3 zapisuje redne brojeve točaka unutar regije interesa oblaka točaka. U ovom završnom radu te točke se ponovno prikazuju u oblaku točaka crvenom bojom.



Slika 34. Prikaz mogućih točaka kolizije

Dodatno, funkcija za pronalazak regije interesa koristi se za uklanjanje postolja u ovom završnom radu odnosno može se koristi za uklanjanje gabaritnih mjera mobilnog robota.

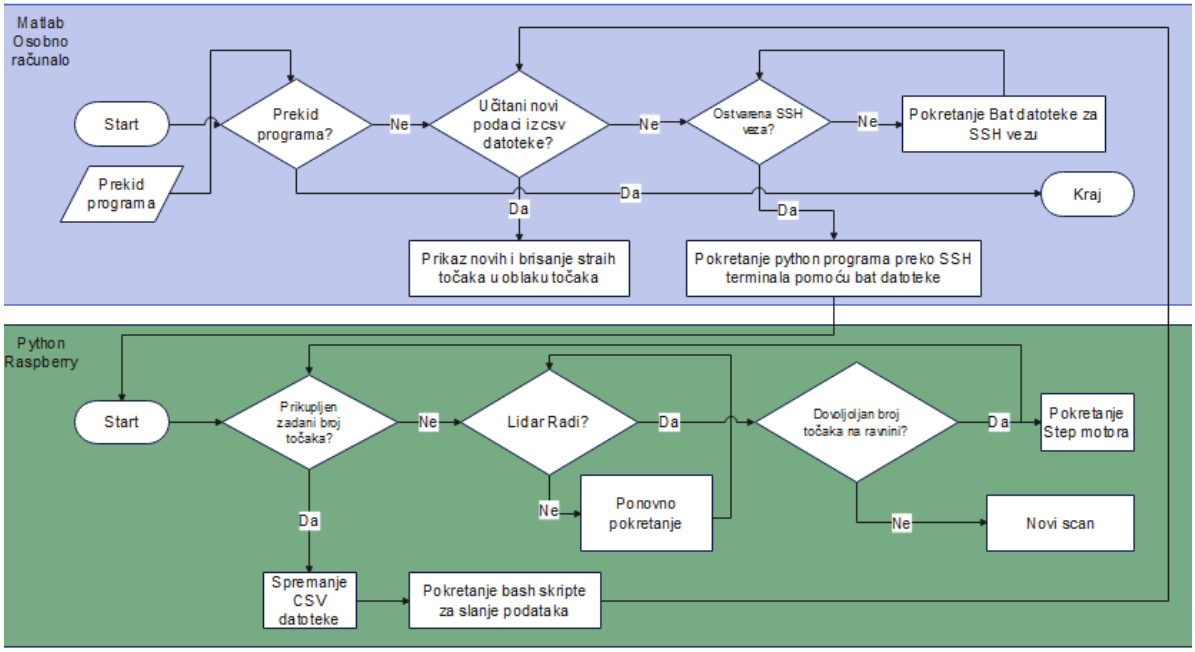
Finalno, zelenom linijom označena je minimalna udaljenost od senzora do točke u regiji interesa kolizije. Ta linija se mijenja ovisno o poziciji robota u prostoru. Iznos kuteva i udaljenosti ispisuju se u terminalu Matlab-a.

```
Kut Alfa(deg):
         89.50

Kut Beta(deg):
         90.23

Udaljenost(mm):
        507.75
```

Slika 35. Ispis najbliže točke kolizije u Matlab-u

5.3. Pojednostavljen prikaz toka programa mobilnog robota

Za povezivanje programa Matlab na osobnom računalu i Python programa na Raspberry Pi-u potrebno je ostvariti komunikaciju. Dodatno potrebno je automatizirat proces prikupljanja podataka i izvođenja svih programa s različitih računala. Za ostvarivanje veze koristi SSH(Secure Shell) komunikacija koja omogućuje slanje i primanje podataka, te pokretanje programa uz pomoć terminala. Jednom kada se pokrene program iz Matlaba na osobnom računalu, proces je automatiziran i automatski obnovljeni oblak točaka prikazuje u određenom vremenu. Za uspostavljanje SSH veze i slanje naredbi za pokretanje python programa na raspberry pi koriste se .bat programi. Pokretanje tih programa izvodi Matlab. Dodatno kada je prikupljen dovoljan broj točaka na raspberry-u pokreće se linux verzija .bat programa pod ekstenzijom bash.



Slika 36. Tok izvođenja programa mobilnog robota.

6. Usporedba sa profesionalnim uređajima

Kako bi se dobio dojam točnosti i cijene profesionalnih 3D laserskih skenera uzet je za primjer Leica BLK360 3D Laser Scanner. Ostali dostupni uređaji od firme *Datum Tech Solution* nemaju ni navedene cijene bez direktnog kontakta s proizvođačem.

Slika 37. Laica BLK360 3D Laser Scanner

Cijena iznajmljivanja ovog uređaja počinje od 2250$ za mjesec dana. Vrijeme skeniranja cijele sfere iznosi malo manje od 3 minute te navode kako se skenira 360000 točaka po sekundi. Dodatno koristi se kamera kako bi se stvorila sferna slika od 150MP te se integrira u oblak točaka. Težina uređaja iznosi 1 kg, visina iznosi 165 mm a promjer 100 mm.

7. Zaključak

Iako je ovaj završni rad mogao biti napisan samo teoretski, tek kada se proizvodi izrade i stave u primjenu dolazi do izražaja koliko stvari nije bilo uzeto u obzir, do tada su sve to samo idealne brojke na papiru. Iako nije prikazana prva verzija konstrukcije u ovome završnom radu, pri rotaciji Z-osi i rotaciji LIDARA, cijela glava rotacije je vidljivo vibrirala, te se moglo zaključit da mjerenja nisu od velike točnosti. Nadalje, kada se gleda sustav za bilo koje mjerenje, točnost sustava ovisi o točnosti njegovog najslabijeg člana. Najslabiji član može biti konstrukcijska točnost senzora, način rada senzora ili mjerno područje senzora. Kada se birao koračni motor, jedan od kriterija je bio što manji kut rotacije između koraka, iako prema specifikacija motora s reduktorom to je 0.018°, najveći problem točnosti dolazi kod LIDAR-a zbog specifikacije mjerenja presjeka površine pod nazivom „Scan field flatness“ koje iznosi od -1.5° do 1.5°, te iz tog razloga mogu se odabrati koračni motor bez reduktora jer bi njegova točnost bila 1.8°. Točnost izrade same konstrukcije nije ni bila uzeta u obzir u potpunosti, iako se pazilo na poklapanje osi rotacija LIDARA i koračnog motora, nije se mogla garantirati. Broj okretaja koračnog motora bitno se smanjuje zbog reduktora te bi se moralo pripazit na maksimalnu brzinu mobilnog robota, ovisno o prostoru i vremenu potrebnom za jednu punu rotaciju.

7.1. Prijedlozi unapređenja 3D laserskog senzora

Kada bi se ovaj završni rad počeo radit iznova, predložene su sljedeće smjernice s obzirom na stečena iskustva. Prilikom odabira laserskog senzora bio bi odabran 1D Lidar kako bi se mogla lakše odredit mjerna nesigurnost senzora i osigurati točan centar rotacije prilikom konstruiranja cijelog senzora. Dodatno, odabirom 1D lidara smanjuje se težina glave rotacije i potrebnog sklopovlja za pokretanje. Prilikom izbora 2 motora, bili bi izabrani manji koračni motori veće brzine i manjeg redukcijskog omjera, u rasponu od 10:1 do 30:1. Dodatna preciznost bi bila dobivena mikrokoracima koračnih motora. Izbor manjih motora bio bi popraćen uklanjanjem svih komponenti osim senzora s rotacijskih dijelova konstrukcije i ne bi bio prioritet ostvariti beskonačni broj okretaja, već ostaviti dovoljno dugačke žice za ostvarivanje rotacije od 360 do 540 stupnjeva rotacije u smjeru Z-osi. Motor rotacije X ili Y-osi bi ostvario rotaciju od 180 do 270 stupnjeva.

LITERATURA

[1] https://ssed.gsfc.nasa.gov/co2sounder/pdf/IeeeSpaceLidAbshireFinal4-5-11.pdf 27.8.2024

[2] https://www.slamtec.ai/wp-content/uploads/2023/11/LM108_SLAMTEC_rplidarkit_usermaunal_A1M8_v2.2_en.pdf , 27.8.2024

[3] https://www.slamtec.ai/wp-content/uploads/2023/11/LD108_SLAMTEC_rplidar_datasheet_A1M8_v3.0_en.pdf , 27.8.2024

[4] https://www.ti.com/lit/an/snva559c/snva559c.pdf?ts=1708434269667 , 27.8.2024

[5] https://www.ti.com/lit/ds/symlink/lm2596.pdf , 27.8.2024

[6] https://www.ti.com/video/series/common-mistakes-in-dc-dc-converters-and-how-to-fix-them.html , 27.8.2024

[7] https://www.raspberrypi.com/documentation/computers/raspberry-pi.html, , 27.8.2024

[8] https://www.raspberrypi.com/documentation/computers/os.html#measure_temp 27.8.2024

[9] https://www.youtube.com/watch?v=LUbhPKBL_IU&t=77s ,27.8.2024

[10] https://www.youtube.com/watch?v=AERfie9wlWM&pp=ygUTZ3JlYXQgc2NvdHQgcnBsaWRhcg%3D%3D , 27.8.2024

[11] https://www.youtube.com/watch?v=_--8K1BW_O4&t=223s , 27.8.2024

[12] https://docs.python.org/3/library/csv.html , 27.8.2024

[13] https://pypi.org/project/RPi.GPIO/ , 27.8.2024

[14] https://docs.circuitpython.org/projects/rplidar/en/latest/ , 27.8.2024

[15] https://www.mathworks.com/help/vision/ref/pcshow.html , 27.8.2024

[16] https://www.mathworks.com/help/vision/ref/pointcloud.findpointsinroi.html , 27.8.2024

[17] https://www.mathworks.com/help/matlab/ref/csvread.html , 27.8.2024

[18] https://www.youtube.com/watch?v=MfVOUSGbe60 , 27.8.2024

[19] https://www.youtube.com/watch?v=SmKw-n-48j8 , 27.8.2024

[20] https://github.com/PowerShell/Win32-OpenSSH/releases , 27.8.2024

[21] https://datumtechsolutions.com/products/leica-rtc360-3d-laser-scanner , 27.8.2024

[22] https://datumtechsolutions.com/products/leica-blk360-3d-laser-scanner , 27.8.2024

PRILOZI

I. Detail_scan.py

II. Real_time_scan.py

III. Pointcloud.m

IV. Point_data.csv

V. Point_data_stol.csv

VI. Point_data_box.csv