



УНИВЕРЗИТЕТ
У НОВОМ САДУ



ФАКУЛТЕТ
ТЕХНИЧКИХ НАУКА

Трг Доситеја Обрадовића 6, 21000 Нови Сад, Република Србија
Деканат: 021 6350-413; 021 450-810; Централа: 021 485 2000
Рачуноводство: 021 458-220; Студентска служба: 021 6350-763
Телефакс: 021 458-133; e-mail: ftndean@uns.ac.rs

ИНТЕГРИСАНИ
СИСТЕМ
МЕНАџМЕНТА
СЕРТИФИКОВАН ОД:



PROJEKAT IZ PREDMETA MEHATRONIKA

NAZIV PROJEKTA:

Cohesion Robotics – Eurobot 2017

TEKST ZADATKA:

Realizacija autonomnog robota

MENTOR PROJEKTA:

Borovac Branislav

Mirko Raković

Jovica Tasevski

PROJEKAT IZRADILI:

Predrag Bogdanić MH15/2013

Đorđe Jović MH26/2013

Marko Marčeta MH1/2013

Aleksandar Hašović MH44/2011

Boris Stanković MH48/2013

Branislav Vajagić MH5/2013

DATUM ODBRANE PROJEKTA:

Sadržaj

PROJEKAT IZ PREDMETA MEHATRONIKA.....	1
Predgovor.....	5
Opis zadatka Eurobot 2017.....	6
Lunarni Moduli.....	6
Rude.....	7
Ostala pravila.....	7
Uvod u problem.....	9
Mašinska konstrukcija.....	10
Noseća ploča i odometrija.....	10
Pogonski mehanizam.....	11
Robotski manipulator i skladište.....	13
Kretanje.....	17
Koordinatni sistemi.....	17
Kretanje robota sa velikim ograničenjima.....	17
Računanje trenutnog položaja i orijentacije robota.....	19
Čista translacija.....	19
Čista rotacija.....	20
Položaj i orijentacija.....	20
Problem sa Omni-direkcionim točkom.....	22
Ideja za kretanje robota sa preprekama.....	22
Abstrakt.....	23
Sinteza tangente na dve kružnice.....	23
Formiranje domena oko prepreke.....	30
Realizacija.....	32
Robotski manipulator.....	33
Osnovna konfiguracija.....	33
Prva faza.....	34
Druga faza.....	36
Treća faza.....	36
Ostavljanje lunarnog modula u malu bazu igrališta.....	36
Ostavljanje lunarnog modula u veliku bazu igrališta.....	37
Senzori.....	38
Infracrveni senzor SHARP GP2Y0A60SZLF.....	38
Infracrveni senzor SHARP 2D120X.....	38

PING))) ultrazvučni senzor	41
Princip rada i komunikacioni protokol:	41
Praktična ograničenja u primeni senzora:	42
TCS34725 senzor za prepoznavanje boja	43
ISC3806 kvadraturni enkoder za odometriju	43
SICK WTB8-P1131 industrijski infracrveni senzor prisustva objekata	44
RPLIDAR A2 laserski skener	45
Određivanje daljine i ugla uzorka	45
Izlazne informacije LIDAR-a:	46
LS7366R kvadraturni dekoder	48
Princip rada 32-bit kvadraturnog brojača.....	49
Aktuatori	50
FR5311M servo motor	50
9g servo motori	50
Koračni motor 28BYJ-48.....	50
Maxon DC motor	51
Upravljačka kola	52
EasyDrive modul za upravljanje koračnim motorom	52
PCA9685 PWM modul	52
VNH2SP40 H-most modul	53
Izvori napajanja	54
Mikrokontroleri i procesori.....	56
Arduino Due	56
Raspberry Pi 3 model B	57
PCB pločice	58
Falcon	58
Eagle	60
Upravljanje	62
Osnovni zadaci.....	62
Komunikacija.....	62
Softver za Raspberry Pi	63
Grafički deo	64
Komunikacija na Raspberry Pi računaru	64
Menadžer zadataka	64
Menadžer logike	65

Mašinska vizija	65
Krajnja realuizacija	69
Zaključak	70

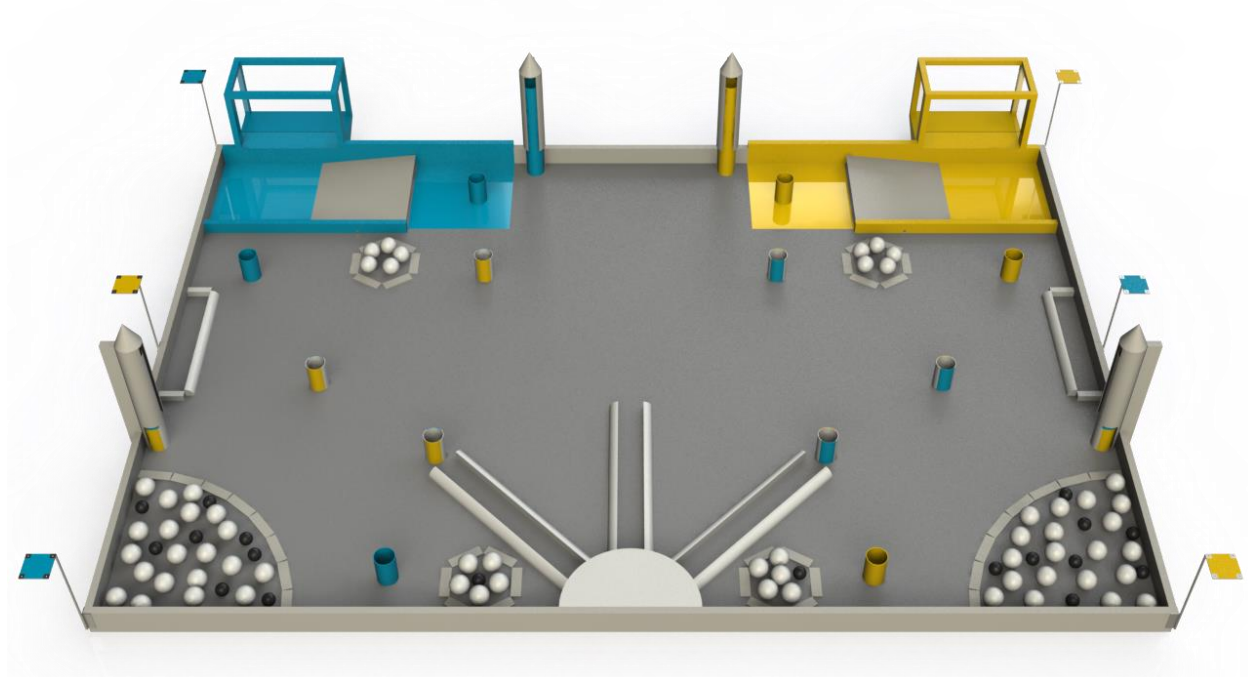
Predgovor

Čitav projekat se svodi na istraživanje novih tehnologija i praktičnu primenu stečenog znanja u ovim oblastima. Ideja je da se napravi robot koji može da ispuni više različitih zadataka, bude relativno svestan svog okruženja, samostalno donosi odluke i ima primenu u industriji za rešavanje svakodnevnih problema. Kako je ovakav pristup problemu vrlo ambiciozan i rešavanje svakog od segmenata nije nimalo naivno, čitav projekat je nailazio na veliki broj prepreka, što je zbog nedostatka vremena, resursa i znanja dovelo do toga da projekat nije u potpunosti realizovan. U ovom papiru se nalazi kompletan opis robota, faze kroz koje je projekat prolazio, na koje je probleme nailazio i kako su problemi rešavani ili ne.

Projekat se u trenutnoj fazi vodi kao neuspešan jer veliki broj gore navedenih zadataka nije završen za takmičenje Eurobot 2017, koje je bilo rok da se zadaci izvrše i eksperimentalno potvrde. Bez obzira na ovakav rezultat, ovim projektom je rešen veliki broj problema i stečeno je znanje koje je neophodno da bi ovaj ili neki naredni projekat bio uspešnije realizovan.

Opis zadatka Eurobot 2017

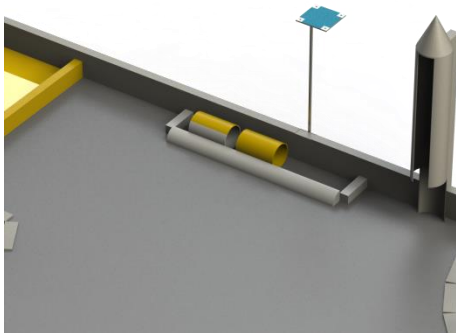
Zadatak za takmičenje Eurobot 2017 sastojao se od igrališta koje je predstavljalo površinu Meseca. Timovi su trebali da sastave lunarnu bazu uz pomoć lunarnih modula (plastičnih šupljih valjaka), skupe rude i ostave ih na adekvatno mesto.



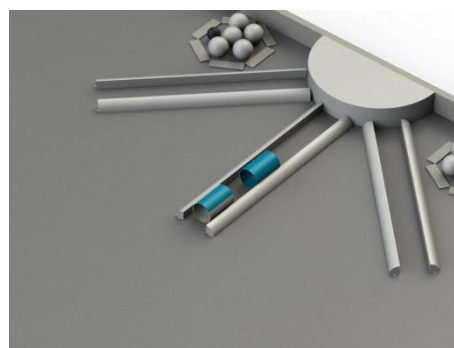
Slika 1 – Igralište za takmičenje Eurobot 2017

Lunarni Moduli

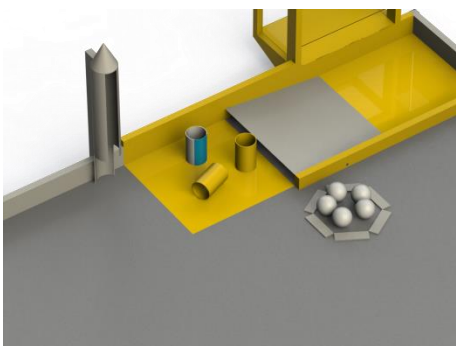
Na igralištu postoje dva tipa lunarnih modula, jednobojni i šareni. Da bi tim osvojio poen uz pomoć lunarnog modula, bilo je potrebno da se on položi na adekvatan način u neku od baza na igralištu (bele ograđene oblasti, gde tri baze nalaze u sredini igrališta i dve se nalaze na stranama) što donosi 10 bodova (slika 2 i slika3); ili da se lunarni modul bilo kako ostavi u početnu oblast, gde je maksimalan broj modula u početnoj oblasti 10 i svaki modul donosi 2 boda (slika 4 i slika 5).



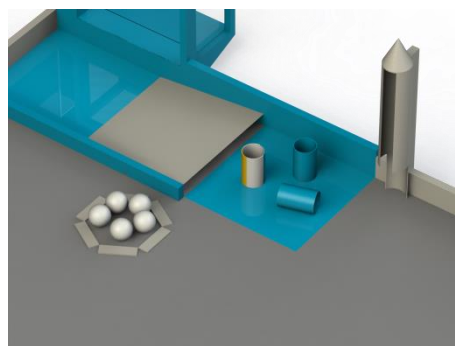
Slika 2 – Adekvatno ostavljeni šareni i jednobojni lunarni modul za žuti tim u malu lunarnu bazu.



Slika 3 – Adekvatno ostavljeni šareni i plavi lunarni moduli za plavi tim u veliku lunarnu bazu.



Slika 4 – Adekvatno ostavljena tri lunarna modula u početnu oblast za žuti tim.

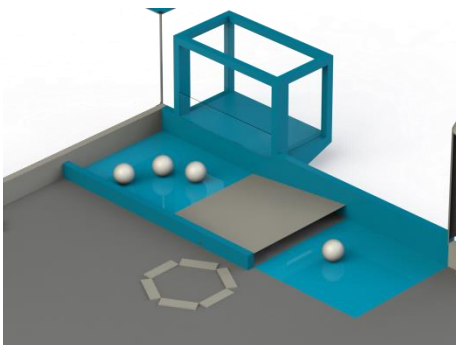


Slika 5 – Adekvatno ostavljena tri lunarna modula u početnu oblast za plavi tim.

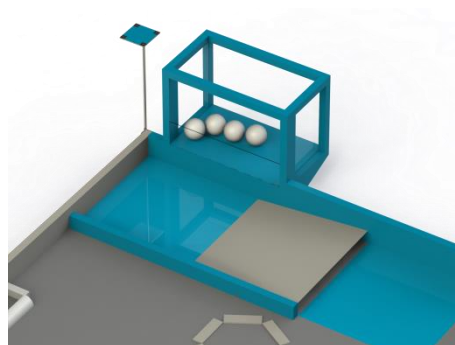
Ovde se primećuje da šareni lunarni moduli koji se ostavljaju u neku od baza na igralištu moraju biti tako ostavljeni da je modul orijentisan sa bojom tima na gore. Bitna stvar je naglasiti da jednobojni lunarni moduli protivničkog tima ne smeju da se diraju, odnesu u bilo koju od baza ili početnu oblast.

Rude

Na igralištu postoje dva tipa rude, Moon Rock i Titanium Ore (koje će se u nastavku teksta zvati kamen i titanium) koje se nalaze unutar kratera. Sa kamenom, šta god da se uradi nikad se ne dobijaju bodovi, dok je bodove sa titanium moguće ostvariti na dva načina. Ostavljanjem titaniuma u početnu oblast dobija se 1 bod (slika 6), gde u početnoj oblasti ne može biti više od 10 ruda zajedno sa lunarnim modulima ako ih tamo ima (zbir lunarnih modula i titaniuma u početnoj oblasti treba da bude manji ili jednak 10). Ostavljanjem titaniuma u korpu početne oblasti dobija se 3 boda po rudi (slika 7).



Slika 6 – Titanium ostavljen u početnu oblast plavog tima.



Slika 7 – Titanium ostavljen u korpu plavog tima.

Ostala pravila

Pored načina na koji se osvajaju poeni, postoje i pravila koja roboti moraju da ispoštuju.

1. Roboti moraju biti autonomni.
2. Ne sme doći do kontakta između dva robota.
3. Robot ne sme da ošteti igralište.
4. Svaki tim može da ima maksimalno dva robota, velikog i malog, gde je maksimalan obim velikog 1200mm, a malog 900mm.
5. Roboti ne smeju biti viši od 350mm računajući od površine igrališta.
6. Robot posle 90 sekundi mora potpuno softverski da se ugasi.

Pored ovih pravila postoje i mnoga druga koja nisu bitna za realizaciju ovog projekta i o kojima neće biti reči. Bitno je naglasiti da se ovaj projekat malo bavi rešavanjem zadatka za takmičenje i usmeren je, kao što je rečeno u predgovoru, ka istraživanju novih tehnologija i pravljanju robota koji u budućnosti može da ima primenu u industriji.

Uvod u problem

Posmatrajući zadatak takmičenja, može se primetiti da strategije i načini realizacija samog robota mogu biti mnogobrojni. Kako jedan dobro ostavljen lunarni modul u neku od baza na igralištu donosi više poena nego tri rude ostavljene u korpu, kompletan fokus i strategija ovog projekta usmerava se na rad sa lunarnim modulima. Ovako je odlučeno iz razloga što su rude velikog prečnika (60mm) zbog čega nije moguće staviti veliki broj ruda u zapreminu koja je na raspolaganju za realizaciju robota. Pored toga rude je potrebno podići na visinu, kako bi se ubacile u korpu i prilikom dolaska do iste potrebno je preći preko klackalice koja se nalazi u početnoj oblasti. Ovo su prilično kompleksni zadaci, a nagrada za rešavanje istih nije dovoljno velika (broj bodova koje je moguće ostvariti).

Sa druge strane rad sa lunarnim modulima je takođe vrlo kompleksan zadatak. Svi moduli su na stolu postavljeni uspravno i da bi se ostavili u neku od baza, potrebno ih je oboriti tj. promeniti im orijentaciju. Pored toga baze su ograđene i šareni moduli se moraju ostaviti na adekvatan način kako bi doneli bodove. Za manipulaciju modulima odlučeno je da se koriste dva identična robotska manipulatora sa 4 stepena slobode, da se moduli skladište unutar robota sa kapacitetom od 6 modula, da se u skladištu orijentišu kako treba i čuvaju. O ovome će više reči biti u poglavlju Robotski manipulator.

Sledeći veliki problem predstavlja kretanje robota. Kako je želja da robot može da vrši složeno kretanje, bilo je potrebno prevazići konvencionalan način za vraćanje pozicije robota i smisliti metode koje ovakvo kretanje može da zadovolji. Pored toga, da bi složeno kretanje moglo da se izvede i kako bi se robot optimalno kretao, potrebno je razviti matematički model koji dovoljno dobro opisuje robota u opštem slučaju i nad takvim modelom izvršiti različite metode optimizacije kretanja. Više reči će biti u poglavlju Kretanje.

Veliki izazovi su postojali i u mašinskoj konstrukciji robota zbog ograničenog prostora i mogućnosti da se model skalira. Jedan od zadataka bio je da robot bude relativno svestan svog okruženja kako bi mogao da planira svoje kretanje i donese adekvatne odluke. Rešavanje ovog problema je pokušano uz pomoć lasera sa dva stepena slobode i taktički postavljenih senzora na robotu. Kako dobijanje informacije o trenutnoj poziciji robota uz pomoć odometrijskih točkova nije najbolje rešenje (jer preziznost merenja zavisi od trenja), pokušano je dobijanje ovih informacija uz pomoć lasera.

Mašinska konstrukcija

Kako je jedan od glavnih zadataka ovog projekta bio da se omogući složeno kretanje robota, bilo je neophodno osmisлити takvu mašinsku konstrukciju koja ovakvo kretanje može da obezbedi. Pored toga bilo je potrebno napraviti konstrukciju koja može da se skalira, na taj način omogućavajući da se iskoristi za različite aplikacije.

Prvo se razmatra oblik baze robota kako bi se pronašla figura koja će sa maksimalnom visinom robota dati najveću zapreminu. Figura koja pri maksimalnim dozvoljenim dimenzijama ima najveću zapreminu jeste cilindar, ali se ubrzo odustalo od izrade robota cilindričnog oblika. Iako cilindar ima najveću zapreminu i omogućava najveće međuosovinsko rastojanje pogonskih točkova, postoje mnogobrojni problemi sa kružnim figurama, zbog kojih je odlučeno da robot bude u obliku kvadra.

Noseća ploča i odometrija

Da bi se postiglo maksimalno međuosovinsko rastojanje, pogonski točkovi su smešteni na krajevima dve naspramne stranice kvadrata, kao što je prikazano na slici 11. Sledeći korak je bio da se odabere mesto za odometrijske točkove. Odometrijski točkovi su u većini slučajeva postavljeni tako da su međusobno paralelni i leže na istoj horizontalnoj osi kao i pogonski točkovi. Ovo nije moglo biti iskorišćeno iz razloga što se na taj način ne mogu dobiti precizni podaci o složenom kretanju, o čemu će biti više reči u poglavlju Kretanje. Odlučeno je da se koriste dva odometrijska točka, jedan paralelan sa pogonskim, a drugi normalan na njega sa mestom na sredini jedne od normalnih stranica kvadrata (normalno u odnosu na stranice na kojima se nalaze pogonski točkovi), što je prikazano na slici 8.



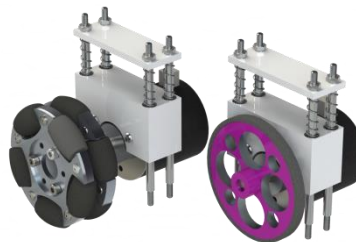
Slika 8 – Noseća ploča robota: levo i desno su prorezi za pogonske točkove, u sredini ploče je prvi pasivni odometrijski točak, na zadnjoj strani prorez za drugi pasivni odometrijski točak, u kružnim rupama se nalaze točkovi za stabilizaciju robota.

Drugi pasivni točak je normalan u odnosu na pravac kretanja robota zbog čega on ne može biti standardni točak već će se koristiti Omni-direkcioni točak kao što je prikazano na slici 9. Ovakvim rešavanjem odometrije postižu se dve stvari, prvi pasivni točak daje informaciju o pređenom putu robota, dok drugi omni-direkcioni pasivni točak daje informaciju o promeni orijentacije robota. Na ovaj način, uz par pretpostavki se može vrlo precizno odrediti složeno kretanje robota, ali više reči o tome će biti u poglavlju Kretanje.

Mora se obezbediti da pasivni točkovi u svakom trenutku dodiruju podlogu po kojoj se robot kreće i moraju biti dovoljno jako pritisnuti uz nju kako bi se obezbedilo dovoljno veliko trenje ali u isto vreme ne smeju nositi veliki deo težine samog robota. Ovo je obezbeđeno plivajućim mehanizmom kao što je prikazano na slici 10.



Slika 9 – Omni-direkcionni točak upotrebljen na robotu.

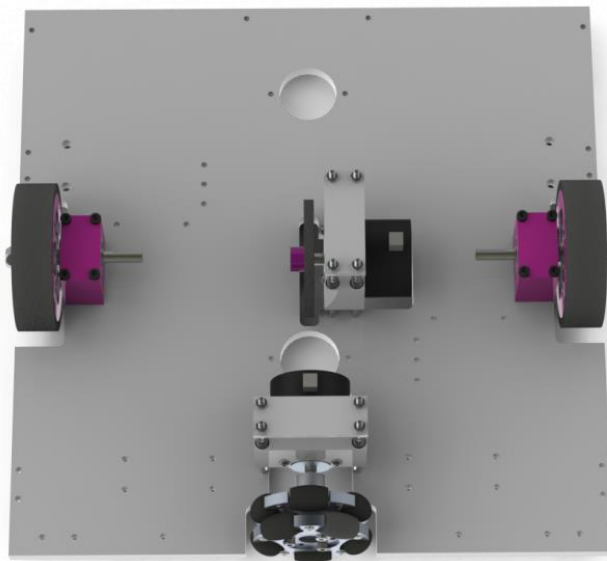


Slika 10 – Plivajući mehanizam pasivnih točkova.

Na ovaj način obezbeđuje se konstantno prijanjanje točka uz podlogu gde se sa brojem opruga, njihovom krutošću i jačinom naprežanja mogu obezbediti različiti pritisci na podlogu po kojoj se robot kreće. Sam Omni-direkcionni točak nije modelovan od strane ovog projekta već je iskorišćen točak dostupan na tržištu, što je ujedno i najveća greška ovog projekta. Zbog lošeg kvaliteta točka kasnije u projektu će se pojaviti veliki problemi.

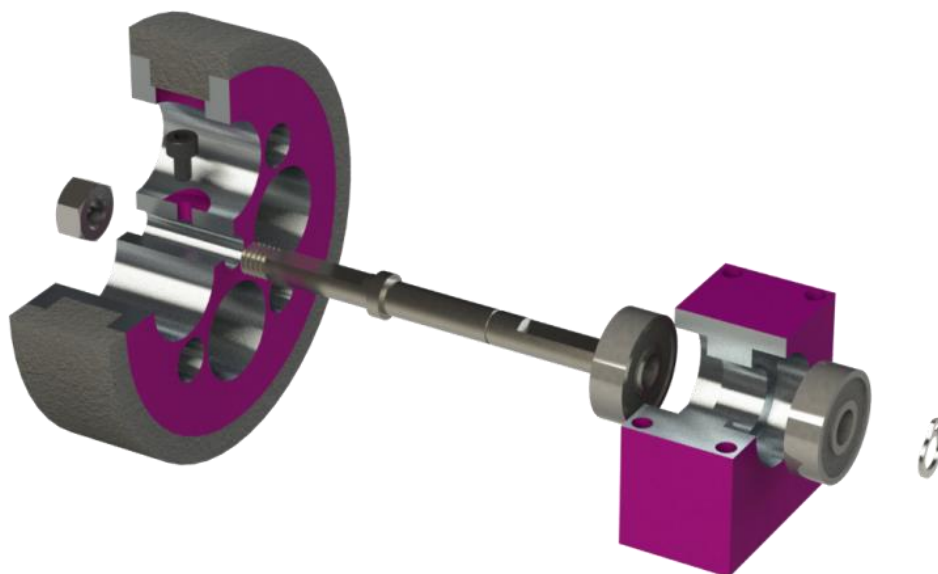
Pogonski mehanizam

Kako svaki pogonski točak treba da ima svoj aktuator moraju se obezbediti nezavisne osovine za oba točka. Ovaj problem je rešen postavljanjem držača sa dva ležaja za svaki točak kao što je prikazano na slici 11.



Slika 11 – Noseća ploča robota sa pogonskim i pasivnim točkovima.

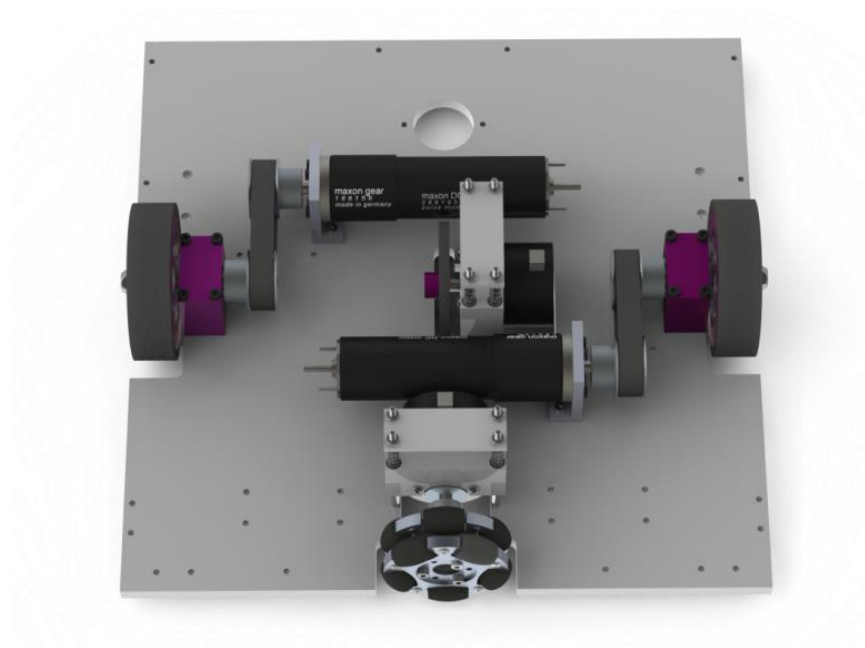
Točak se postavlja što bliže glavčini kako bi se smanjio normalni moment koji se javlja u osovine i glavčini. Pošto se točak vezuje za ploču u samo “jednoj” tački potrebno je koristiti dva ležaja u glavčini i omogućiti što veće rastojanje između njih kako bi se sprečilo savijanje osovine i moguće kidanje vijaka koji glavčinu drže vezanu za ploču. Na način prikazan na slici 12, zadovoljeni su svi potrebni zahtevi.



Slika 12 – Mehanizam držanja pogonskog točka robota.

Za svaki pogonski točak mora se obezbediti prenos obrtnog momenta motora na osovinu točka. Dva razmatrana načina su prenos zupčastim parom i zupčastim kaišem. Iako zupčasti prenos ima svoje velike prednosti, zbog skupe izrade zupčastih parova sa kosim zubcima i malim zazorom bira se druga opcija. Velika mana prenosa momenta zupčastim kaišem je u tome što vrlo lako može doći do pucanja kaiša, kaiš se konstantno isteže i izaziva veliku normalnu silu na osovinama između kojih se moment prenosi. Do poslednje stvari dolazi usled zatezanja kaiša koje je neophodno da bi se obezbedio što manji prazan hod točka. Ovo je posebno opasno ukoliko se jedna remenica nalazi na osovini samog motora jer ona nije predvižena da trpi velika normalna opterećenja. U ovom slučaju je morao da se napravi kompromis zbog nedostatka prostora i kompaktnosti, zbog čega je jedna remenica vezana za osovinu reduktora, a druga na osovinu pogonskog točka.

Motori su na ploči postavljeni tako da osovina motora bude paralelna sa osovinoom točka sa odgovarajućim međuosnim rastojanjem, takvim da se obezbedi lako stavljanje remenice i kaiša, sa uslovom da kaiš može dovoljno dobro da se zategne. Rešenje je prikazano na slici 13. Odabir remenica i njihovog odnosa svodi se na stvar kompromisa između maksimalne brzine kretanja robota i preciznosti pozicije. Tako se prvo razmatrao prenosni odnos od 1:2 pogonske i gonjene remenice respektivno, kako bi se obezbedio veći moment na pogonskim točkovima i preciznija pozicija robota. Obrtni moment od $\sim 30\text{Nm}$ po točku i maksimalna brzina od 0.68m/s nisu bili dovoljno dobri; moment je bio nepotrebno velik za veličinu i težinu robota, dok je maksimalna brzina bila neprihvatljivo mala. Kako postavljanjem prenosnog odnosa od 1:1 pogonske i gonjene remenice respektivno, nije značajno opao kvalitet pozicioniranja robota, a maksimalna brzina je duplo porasla, ovakav prenosni odnos je bio usvojen.



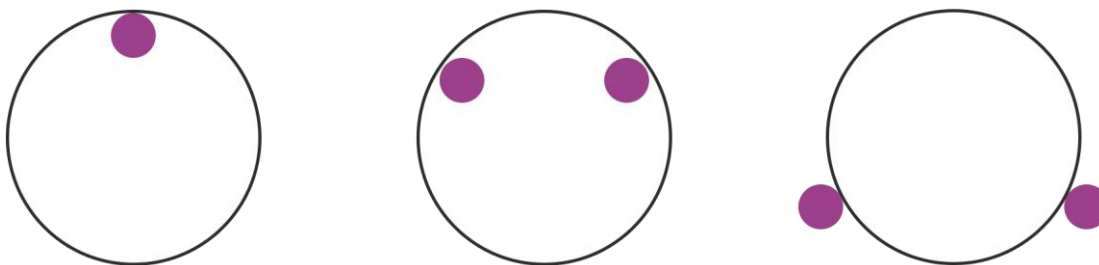
Slika 13 – Noseća ploča robota sa pogonskim mehanizmom i odometrijom.

Na ovaj način obezbeđuje se relativno dobra simetričnost leve i desne strane robota što je bitno kako bi se centar mase čitavog robota kasnije nalazio na pravcu njegovog kretanja. Omni-direkcionni točak kvari simetričnost prednje i zadnje strane ali je neophodno napraviti neke kompromise. Kako bi robot uvek bio paralelan sa podlogom po kojoj se kreće, u kružne rupe stavljaju se direkcionni točkovi tako da zadnji direkcionni točak bude niži u odnosu na prednji. Ovo se radi da bi se izbegao kontakt u četiri tačke prilikom kretanja robota jer bi se u tom slučaju robot zaglavio ili bi došlo do smanjenja trenja između pogonskih točkova i podloge. Na ovaj način robot u svakom trenutku vremena ima kontakt oslanjanja samo u tri tačke.

Robotski manipulator i skladište

Za manipulaciju čunjevima odlučeno je da se koristi robotski manipulator. Iako je vrlo kompleksan za projektovanje i rad on nudi veliki broj prednosti kao što su: mogućnost skupljanja različitih oblika, laku promenu orijentacije predmeta koji se nalazi u hvataljci, može da kompenzuje grešku u poziciji i mnoge druge stvari. Zbog ograničenog prostora neophodno je videti koje sve položaje manipulator mora da postigne kako bi se odredio broj stepeni slobode, i na taj način obezbediti da manipulator ne bude redundantan (više stepeni slobode nego što je potrebno, značilo bi da prostor i resursi nisu iskorišćeni na adekvatan način). Stepni slobode će se odrediti nakon što se reši način skladištenja lunarnih modula.

Moduli se u robota mogu sladištiti na dva načina tako da moduli budu uspravni ili položeni. Ovde je važno da se vodi računa i o tome da šareni lunarni moduli moraju da se orijentišu na odgovarajući način, tako da kada se izvade iz robota i ostave, odgovarajuća boja gleda na gore. U koliko bi se moduli ostavljali uspravno, potrebno je da se slažu jedan na drugi ili ostavljaju uspravni na neku ravnu površinu unutar robota. Ovako bi se unutar robota morao rezervisati veliki prostor samo za module, kompleksan sistem za kretanje modula u robotu i sistem za njihovu orijentaciju. Zbog toga se odlučuje da se moduli postavljaju oboreni na šipke za vođenje modula. Kako su moduli šuplji, horizontalno se mogu ostaviti na šipke ili da se ostavljaju tako da šipke prolaze kroz njih kao što je prikazano na slici 14.



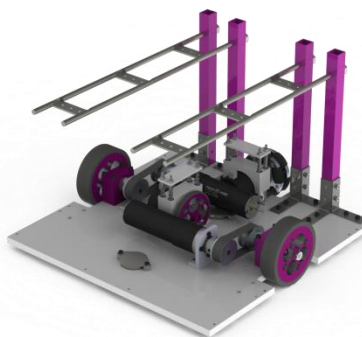
Slika 14 – Ilustracija ostavljanja lunarnih modula u skladište robota. Sa leve strane je prikazano ostavljanje na jednu šipku, zatim na dve sa kontaktom na unutrašnjoj strani modula i sa kontaktom na spoljnoj strani modula.

U prvom slučaju sa leve strane na slici 14, modul je u vrlo nestabilnom položaju, može da osciluje levo, desno (u odnosu na horizontalu) i veliku silu je potrebno upotrebiti kako se modul ne bi kretao po osi šipke. U slučaju na desnoj strani teško je obezbediti da šipke ostanu na konstantnom međurastojanju sa malim oscilacijama, postoji mogućnost da modul ispadne sa šipki i u koliko bi se modul hvatao sa spoljne strane, šipke bi pravile veliki problem prilikom ostavljanja. Zbog ovih razloga odlučuje se na realizaciju u sredini. Modul ne može da ispadne sa šipki, lako se obezbeđuje treća tačka kontakta koja sprečava oscilovanje čunja i njegovo kretanje po osi šipki, relativno se lako implementuje mehanizam za rotaciju čunja i sila kojom je potrebno pritisnuti čunj uz šipke ne mora da bude velika.

Da bi se obezbedio kapacitet od 6 modula unutar robota neophodna su 2 identična skladišta, gde svako skladište može da sadrži 3 modula. Jedno skladište se realizuje tako da pravac kretanja čunjeva u skladištu bude u pravcu kretanja robota.



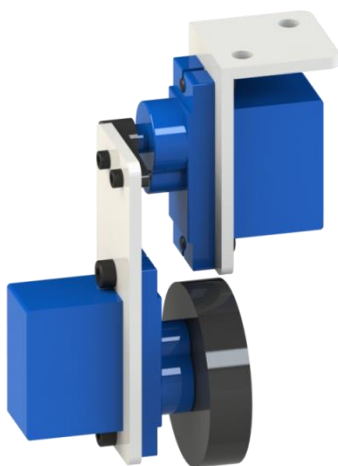
Slika 15 – Skladište za lunarne module



Slika 16 – Povezivanje skladišta na noseću ploču robota

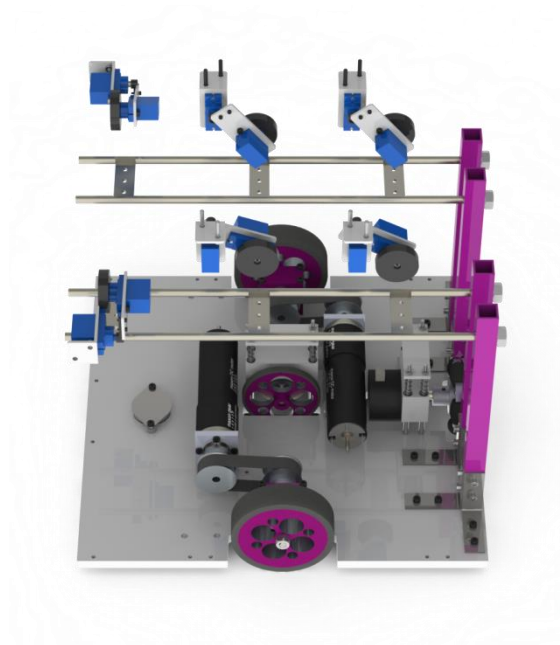
Kao što je prikazano na slici 16, skladište je izdignuto u odnosu na ploču robota. Stubovima i mostovima između šipki obezbeđeno je konstantno rastojanje i vrlo male, zanemarljive oscilacije na krajevima šipki. Kako skladište radi po principu FILO (*First In Last Out*) potrebno je obezbediti mehanizam koji će da omogući kontrolisano kretanje modula po skladištu i mehanizam koji će omogućiti orijentisanje modula.

Ovaj problem je rešen tako što se modul koji je stavljen na početak skladišta, pritisne uz šipke i zatim se postavi u odgovarajuću orijentaciju. Kada je sledeći modul spreman da bude smešten u skladište, uklanja se sila koja je modul pritiskala uz šipke, novi modul odgurne stari na sledeće mesto u skladištu, obezbeđuje se sila koja oba modula pritiska uz šipke i obezbeđuje se odgovarajuća orijentacija modula na početku skladišta. Ceo proces se ponavlja prilikom smeštanja novog modula. Kada u skladištu postoje moduli tako da je prvo mesto skladišta slobodno, preostali moduli se pomeraju za jedno mesto. Ovakav mehanizam se realizuje sa 3 grupe od po dva servo motora postavljenih u odgovarajuću konfiguraciju (slika 17) od kojih je jedan servo kontinualne rotacije, a drugi servo koji može da postigne odgovarajuće uglove.

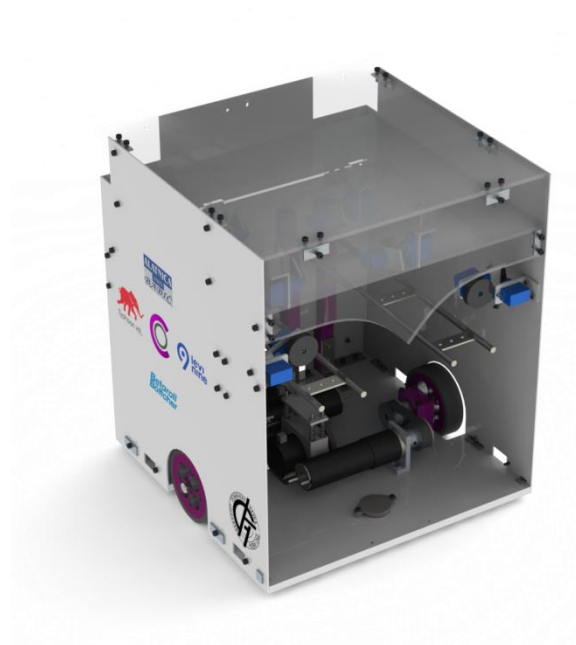


Slika 17 – Mehanizam za okretanje i vađenje lunarnih modula.

Prva grupa se postavlja tako da je točak normalan u odnosu na smer kretanja modula u skladištu i to na prvo mesto skladišta. Druge dve grupe se postavljaju na drugo i treće mesto u skladištu tako da je točak u smeru kretanja modula u skladištu. Ove mehanizme je moguće postaviti nakon pravljenja oklopa robota kao što je prikazano na slici 19.

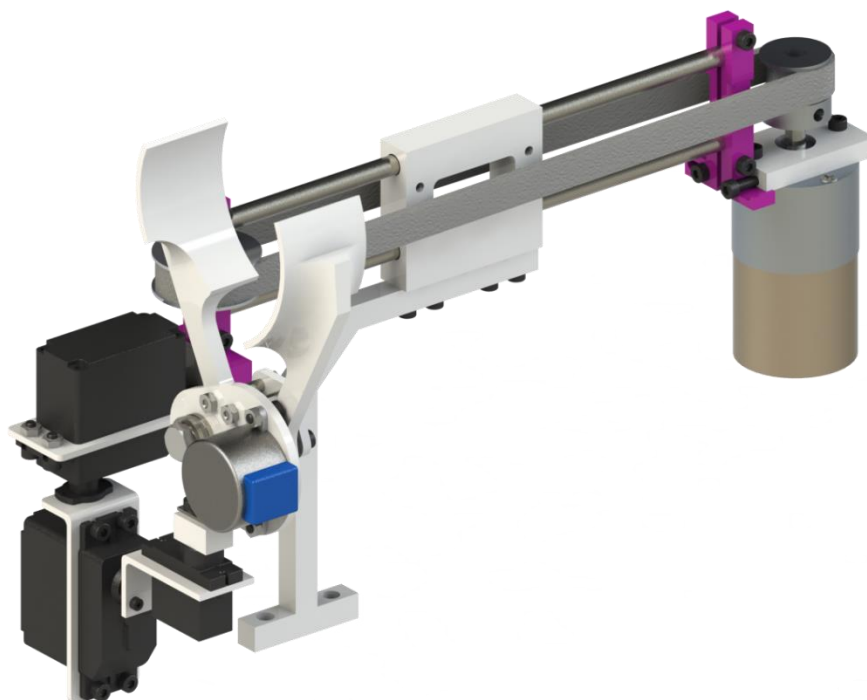


Slika 18 – Skladište robota

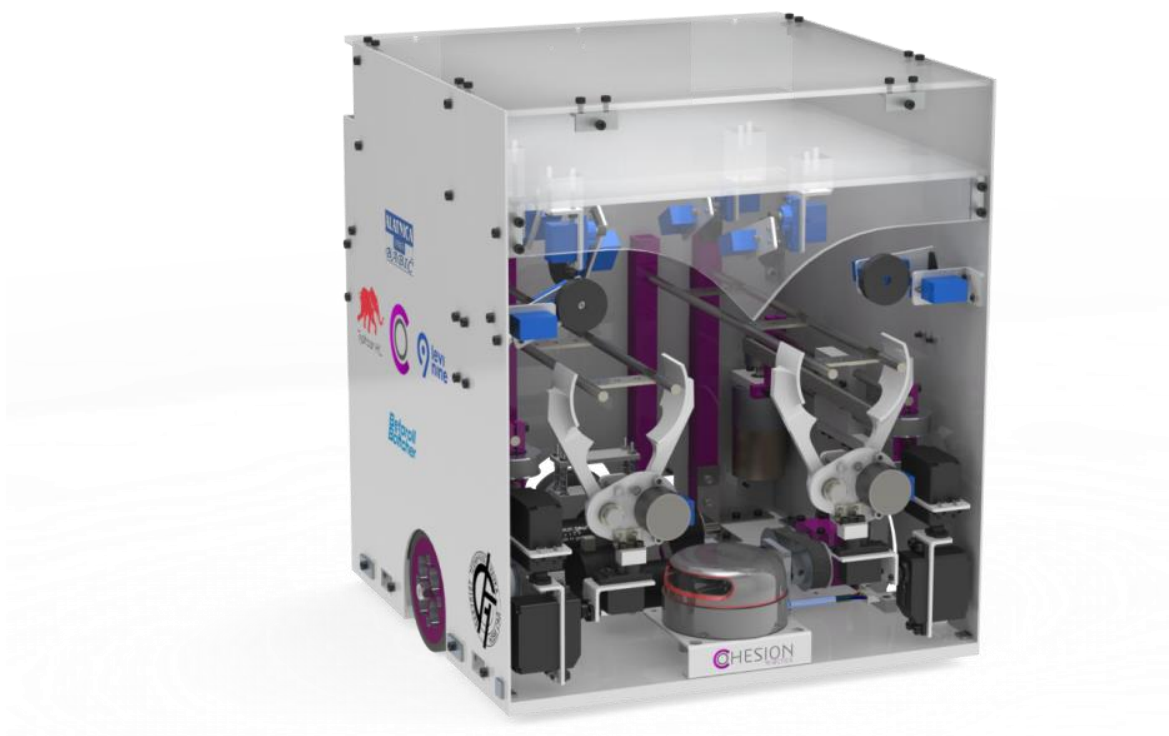


Slika 19 – Robot sa mehanizmima

Sada kad postoji mehanizam za skladište moguće je odrediti potreban broj stepeni slobode za robotski manipulator i konfiguraciju samih zglobova. Da bi se modul oborio i postavio na prvo mesto skladišta, potreban je jedan stepen slobode; da bi se hvataljka manipulatora poravnala sa osom modula potreban je još jedan stepen slobode; da bi se modul postavio oboren na površinu kretanja robota potreban je još jedan stepen slobode. Ova tri uslova ispunjavaju se sa 3 rotaciona zgloba na odgovarajući način postavljena u prostoru. Pored ova 3 stepena slobode potreban je još jedan linearni koji olakšava vađenje modula iz raketa na igralištu.



Slika 20 – Robotski manipulator sa 4 stepena slobode



Slika 21

Kako manipulator automatski cilja čunjeve i time kompenzuje grešku u poziciji robota, biće objašnjeno u poglavlju Robotski manipulator. Na kraju se u robota postavljaju dva manipulatora i na njegovo čelo se postavlja RPLIDAR. Između dva Plexy Glass-a nalazi se prostor za upravljačka kola robota.

Kretanje

Dosadašnja rešenja za kretanje robota su podrazumevala da robot u svakom trenutku vremena može da se translira po pravoj ili da se rotira oko svoje ose. Prvo će se postaviti koordinatni sistemi, obraditi model ovako jednostavnog kretanja, nakon čega će se model dopuniti i dati predlog za rešavanje složenog kretanja.

Koordinatni sistemi

Kako se robot kreće u ravni igrališta, koristiće se dvodimenzionalni dekartov koordinatni sistem, gde se prvo postavlja inercijalni. Postavlja se tako da bude vezan za igralište i da se sva kretanja robota nalaze unutar prvog kvadranta. x-osa se postavlja uz dužu stranicu igrališta, a y-osa uz kraću. Ovaj koordinatni sistem se naziva $\Sigma O(X_O, Y_O)$ i svi ostali koordinatni sistemi predstavljaju se u ovom. ΣO je opisan jediničnim vektorima $\mathbf{i}, \mathbf{j}$ respektivno.

Za robota se vezuje koordinatni sistem $\Sigma R(X_R, Y_R, \varphi_R)$ tako da je x_R -osa u pravcu kretanja robota i sa y_R -osom gradi desni dekartov koordinatni sistem. x_R -osa je u svakom trenutku vremena tangenta na putanju robota, dok je ΣR opisan jediničnim vektorima $\mathbf{x}_R, \mathbf{y}_R$, iz čega sledi:

$$\mathbf{x}_R = \cos(\varphi_R) \mathbf{i} + \sin(\varphi_R) \mathbf{j}, \quad (1)$$

$$\mathbf{y}_R = -\sin(\varphi_R) \mathbf{i} + \cos(\varphi_R) \mathbf{j}, \quad (2)$$

gde je, φ_R ugao orijentacije robota i predstavlja ugao između osa x_O i x_R . Za početni položaj robota se zna da je $\Sigma R(X_R(0), Y_R(0), \varphi_R(0))$ neka poznata vrednost. Kasnije će se videti kako se dobijaju vrednosti za $t > 0$.

Mesto u koje robot treba da stigne opisano je koordinatnim sistemom $\Sigma R_1(X_{R1}, Y_{R1}, \varphi_{R1})$.

Kako orijentacija lunarnih modula na stolu nije od važnosti, svi moduli se opisuju vektorom položaja $\mathbf{r}_{chl}$, $l=1,2,3,\dots, u$; gde je u broj lunarnih modula na igralištu, a vektor položaja je predstavljen kao:

$$\mathbf{r}_{chl} = X_{chl} \mathbf{i} + Y_{chl} \mathbf{j}. \quad (3)$$

Kretanje robota sa velikim ograničenjima

Ovakvo kretanje robota se već uveliko koristi zbog svoje jednostavnost, lake implementacije u realan sistem i da bi robot ovakvo kretanje vršio nije potreban matematički model sistema već se kretanje može rešiti jednostavnom regulacijom. Ovo je ujeno i problem ovakvog rešenja jer kretanje u velikom broju slučajeva nije optimalno i nije moguće zadovoljiti kriterijum minimalnog vremena. Takođe optimizaciju trajektorije i izbegavanje sudara nije moguće uraditi na adekvatan način. Zbog toga će se u sledećem poglavlju predložiti model koji dopunjuje ovaj, nakon čega će se predložiti model potpuno neholonomnog kretanja.

Ograničenja ovakvog kretanja za bilo koji trenutak vremena su:

1. translacija duž x_R -ose,
2. rotacija oko z_R -ose.

U svakom trenutku može biti zadovoljen samo jedan od dva uslova. U opštem slučaju ovo znači da robot prilikom kretanja između dva položaja mora da prođe kroz tri faze:

1. robot se rotira oko z_R -ose kako bi se x_R -osa poravnala sa duži RR_1 ,
2. robot se translira duž RR_1 ,
3. robot se rotira oko z_R -ose kako bi se x_R -osa poravnala sa x_{R1} -osom.

Ovde su poznati parametri $\Sigma R(X_R(t_1), Y_R(t_1), \varphi_R(t_1))$, gde je t_1 trenutak vremena neposredno pre početka kretanja i $\Sigma R_1(X_{R1}(t_2), Y_{R1}(t_2), \varphi_{R1}(t_2))$ gde je t_2 nepoznat trenutak vremena ali se zna da su u njemu završava kretanje i znaju se položaj i orijentacija robota u tom trenutku. Ovaj trenutak je moguće odrediti ali se ne može izvršiti optimizacija vremena jer je sistem potpuno definisan i ograničen.

Potrebno je pronaći dva relativna ugla, $\varphi_{RR1}^{(I)}$ – ugao prve rotacije i $\varphi_{RR1}^{(II)}$ – ugao druge rotacije kao i rastojanje RR1 za koje se treba translirati. Ovo će se uraditi na sledeći način:

$$\mathbf{r}_{RR1} = \mathbf{r}_{R1} - \mathbf{r}_R, \quad (4)$$

$$X_{RR1} = X_{R1} - X_R, \quad (5)$$

$$Y_{RR1} = Y_{R1} - Y_R, \quad (6)$$

$$RR1 = \sqrt{X_{RR1}^2 + Y_{RR1}^2} \quad (7)$$

Ugao $\varphi_{RR1}^{(I)}$ se računa na sledeći način:

$$1. \quad X_{R1} > X_R \wedge Y_{R1} > Y_R$$

$$\varphi_{RR1}^{(I)} = \text{atan}\left(\frac{Y_{RR1}}{X_{RR1}}\right), \quad (8)$$

$$2. \quad X_{R1} < X_R \wedge Y_{R1} > Y_R \vee X_{R1} < X_R \wedge Y_{R1} < Y_R$$

$$\varphi_{RR1}^{(I)} = \text{atan}\left(\frac{Y_{RR1}}{X_{RR1}}\right) + \pi, \quad (9)$$

$$3. \quad X_{R1} > X_R \wedge Y_{R1} < Y_R$$

$$\varphi_{RR1}^{(I)} = \text{atan}\left(\frac{Y_{RR1}}{X_{RR1}}\right) + 2\pi, \quad (10)$$

$$4. \quad X_{R1} = X_R \wedge Y_{R1} = Y_R$$

$$\begin{aligned} \varphi_{RR1}^{(I)} &= \varphi_{R1} - \varphi_R, \\ \forall \varphi_{RR1}^{(I)} > \pi &\rightarrow \varphi_{RR1}^{(I)} = \varphi_{RR1}^{(I)} - 2\pi, \\ \varphi_{RR1} &= \varphi_R + \varphi_{RR1}^{(I)}, \end{aligned} \quad (11)$$

gde je φ_{RR1} ugao koji vektor $\mathbf{r}_{RR1}$ zaklapa sa x_O -osom.

5. U svim ostalim slučajevima:

$$\varphi_{RR1}^{(I)} = 2 \text{atan}\left(\frac{\sqrt{1 - \left(\frac{Y_{RR1}}{X_{RR1}}\right)^2}}{1 + \frac{Y_{RR1}}{X_{RR1}}}\right). \quad (12)$$

Nakon što se odrede i izračunaju prethodni uslovi, postoji još jedan dodatni kako bi relativan ugao $\varphi_{RR1}^{(I)}$ bio za svaki slučaj sa adekvatnim predznakom:

$$X_{R1} \neq X_R \wedge Y_{R1} \neq Y_R$$

$$\varphi_{RR1} = \varphi_{RR1}^{(I)}, \quad (13)$$

$$\varphi_{RR1}^{(I)} = \varphi_{RR1}^{(I)} - \varphi_R, \quad (14)$$

$$\forall(\varphi_{RR1}^{(I)} < -\pi) \rightarrow \varphi_{RR1}^{(I)} = 2\pi - \varphi_{RR1}^{(I)}, \quad (15)$$

$$\forall(\varphi_{RR1}^{(I)} > \pi) \rightarrow \varphi_{RR1}^{(I)} = \varphi_{RR1}^{(I)} - 2\pi. \quad (16)$$

Sada je potrebno da se odredi relativan ugao $\varphi_{RR1}^{(II)}$ kako bi se kompletirali svi potrebni parametri za tri faze, a on se određuje na sledeći način:

$$\varphi_{RR1}^{(II)} = \varphi_{R1} - \varphi_{RR1}, \quad (17)$$

$$\forall(\varphi_{RR1}^{(II)} < -\pi) \rightarrow \varphi_{RR1}^{(II)} = 2\pi - \varphi_{RR1}^{(II)}, \quad (18)$$

$$\forall(\varphi_{RR1}^{(II)} > \pi) \rightarrow \varphi_{RR1}^{(II)} = \varphi_{RR1}^{(II)} - 2\pi. \quad (19)$$

Kada su izračunati parametri za sve tri faze kroz njih se prolazi tako što se na početni ugao robota doda relativan ugao $\varphi_{RR1}^{(I)}$, zatim se robot translira za dužinu RR1 nakon čega se na trenutni ugao robota doda ugao $\varphi_{RR1}^{(II)}$.

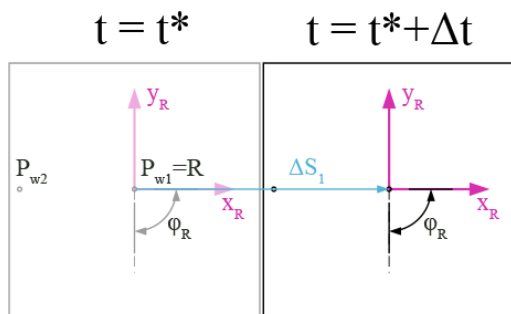
Računanje trenutnog položaja i orijentacije robota

Glavni način kako se određuje trenutna pozicija robota jeste putem informacije enkodera sa odometrijskih točkova. U nastavku teksta, prvi pasivni točak odnosiće se na odometrijski točak koji je paralelan sa pogonskim i daje informaciju o pređenom putu, dok se drugi pasivni točak odnosi na Omni-direkcionu odometrijski točak i daje informaciju o promeni orijentacije robota.

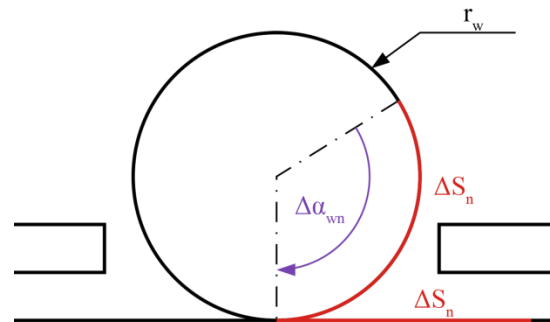
Položaj i orijentacija robota za trenutak $t = 0$ su poznati. tj. $X_R(t_0) = const.$, $Y_R(t_0) = const.$, $\varphi_R(t_0) = const.$, $\mathbf{v}_R(t_0) = 0$ i $\boldsymbol{\omega}_R(t_0) = 0$, gde je $\mathbf{v}_R$ brzina geometrijskog centra robota, a $\boldsymbol{\omega}_R$ ugaona brzina robota. Kako se radi o diskretnom sistemu $t = t^* + \Delta t$, gde je t^* poslednji trenutak u kom su poznati svi parametri sistema, Δt predstavlja vreme koje protekne između dva diskretna stanja. Naredno rešenje može da funkcioniše samo za dovoljno malo Δt što podrazumeva da je $\Delta t < 16.7ms$, što važi za sistem predstavljen u ovom projektu. Bitno je da za vreme Δt ne dođe do značajnog pomeranja sistema kako bi se greška proračuna svela na minimum.

Čista translacija

Uzima se za primer da je robot u trenutku t^* bio u položaju kao što je prikazano na slici 22. Za vreme Δt sistem će se pomeriti duž x_R -ose za rastojanje ΔS_1 (slika 23), pod pretpostavkom da nije došlo do promene orijentacije. ΔS_1 rastojanje je potrebno predstaviti preko ugaonog zakretanja točka, tj. preko broja koraka enkodera, koraka koji su se desili za vreme Δt . Odometrijski točak će se predstaviti kao što je prikazano slikom 23, gde je $n=1,2$; r_w – poluprečnik točka, $\Delta\alpha_{wn}$ – ugao zakretanja točka i ΔS_n – pređeni put počka.



Slika 22 – Ilustracija čiste translacije robota.



Slika 23 – Ilustracija pasivnog točka robota.

Pređeni put ΔS_1 se dobija na sledeći način:

$$\Delta\alpha_{wn} = \frac{2\pi}{ENC_n} \Delta enc_n, \quad (20)$$

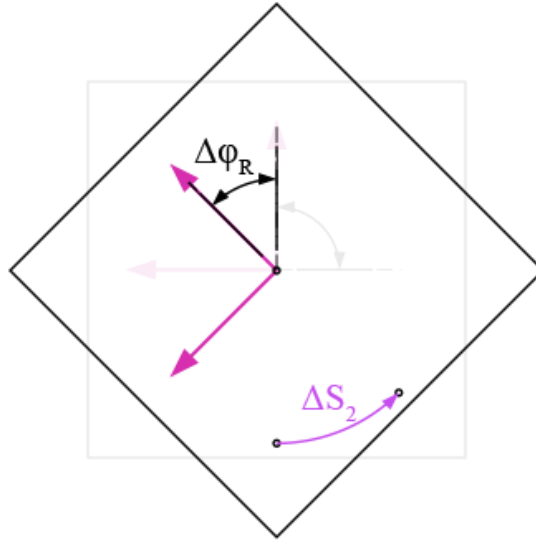
gde je ENC_n broj zareza enkodera dok je Δenc_n broj koraka koji se smenio za vreme Δt ;

$$\Delta S_1 = r_{w1} \Delta\alpha_{w1}, \quad (21)$$

$$\Delta S_1 = \frac{2\pi r_{w1}}{ENC_1} \Delta enc_1. \quad (22)$$

Važno je primetiti da jednačina (22) ne govori o položaju sistema već samo o pređenom putu.

Čista rotacija



Slika 24 – Ilustracija čiste rotacije robota.

$$\Delta S_2 = r_{w2} \Delta\alpha_{w2}, \quad (23)$$

$$\Delta S_2 = DPW \Delta\varphi_R, \quad (24)$$

gde je DPW rastojanje između tačaka P_{w1} i P_{w2} , prikazanih na slici 22. Ubacivanjem jednačine (23) u (24), dobija se:

$$\Delta\varphi_R = \frac{2\pi r_{w2}}{DPW ENC_2} enc_2, \quad (25)$$

Položaj i orijentacija

ΔS_1 govori o tome koliki je put robot prešao za vreme Δt ali ne daje informaciju o tome kako je taj put pređen. $\Delta\varphi_R$ daje informaciju o priraštaju rotacije robota gde je $\varphi_R(t) = \varphi_R(t^*) + \Delta\varphi_R$. Na osnovu ovog se zna da na kraju puta ΔS_1 robot ima orijentaciju $\varphi_R(t^*) + \Delta\varphi_R$. Primećuje se da u ovako postavljenom sistemu nije moguće odrediti položaj robota u trenutku t , zbog čega se uvodi sledeća pretpostavka;

Kako je vreme Δt dovoljno malo, ne dolazi do značajne promene položaja sistema. Kako je sistem dovoljno trom, za vreme Δt neće doći do značajne promene ubrzanja.

1. Ako je $\Delta enc_1 \neq 0$ i $\Delta enc_2 = 0$, robot se za vreme Δt kretao po pravoj koja leži na x_R -osi i prešao je put ΔS_1 :

$$X_R(t) = X_R(t^*) + \Delta S_1 \cos(\varphi_R(t^*)), \quad (26)$$

$$Y_R(t) = Y_R(t^*) + \Delta S_1 \sin(\varphi_R(t^*)), \quad (27)$$

$$\varphi_R(t) = \varphi_R(t^*). \quad (28)$$

2. Ako je $\Delta \text{enc}_1 = 0$ i $\Delta \text{enc}_2 \neq 0$, robot se za vreme Δt zarotirao oko z_R -ose za ugao $\Delta \varphi_R$:

$$X_R(t) = X_R(t^*), \quad (29)$$

$$Y_R(t) = Y_R(t^*), \quad (30)$$

$$\varphi_R(t) = \varphi_R(t^*) + \Delta \varphi_R \quad (31)$$

3. Ako je $\Delta \text{enc}_1 \neq 0$ i $\Delta \text{enc}_2 \neq 0$, robot se kretao po zamišljenoj kružnici C.

$$\begin{aligned} \forall(\Delta \varphi_R < 0) &\rightarrow \alpha_{norm} = -\frac{\pi}{2}, \\ \forall(\Delta \varphi_R \geq 0) &\rightarrow \alpha_{norm} = \frac{\pi}{2}, \end{aligned} \quad (32)$$

$$R_C = \frac{\Delta S_1}{|\Delta \varphi_R|}, \quad (33)$$

$$X_C = X_R(t^*) + R_C \cos(\varphi_R(t^*) + \alpha_{norm}), \quad (34)$$

$$Y_C = Y_R(t^*) + R_C \sin(\varphi_R(t^*) + \alpha_{norm}), \quad (35)$$

$$r_C = \sqrt{X_C^2 + Y_C^2}, \quad (36)$$

$$\varphi_C = \text{atan}\left(\frac{Y_C}{X_C}\right), \quad (37)$$

$$\forall(Y_C > 0 \wedge \varphi_C < 0 \vee Y_C < 0 \wedge \varphi_C > 0) \rightarrow \pi + \varphi_C, \quad (38)$$

$$X_R(t) = r_C \cos(\varphi_C) + R_C \cos(\varphi_R(t^*) - \alpha_{norm} + \Delta \varphi_R), \quad (39)$$

$$Y_R(t) = r_C \sin(\varphi_C) + R_C \sin(\varphi_R(t^*) - \alpha_{norm} + \Delta \varphi_R), \quad (40)$$

$$\varphi_R(t) = \varphi_R(t^*) + \Delta \varphi_R, \quad (41)$$

gde je:

α_{norm} – pomoćni ugao,

R_C – poluprečnik zamišljene kružnice po kojoj se robot kreće,

X_C, Y_C – položaj centra kružnice,

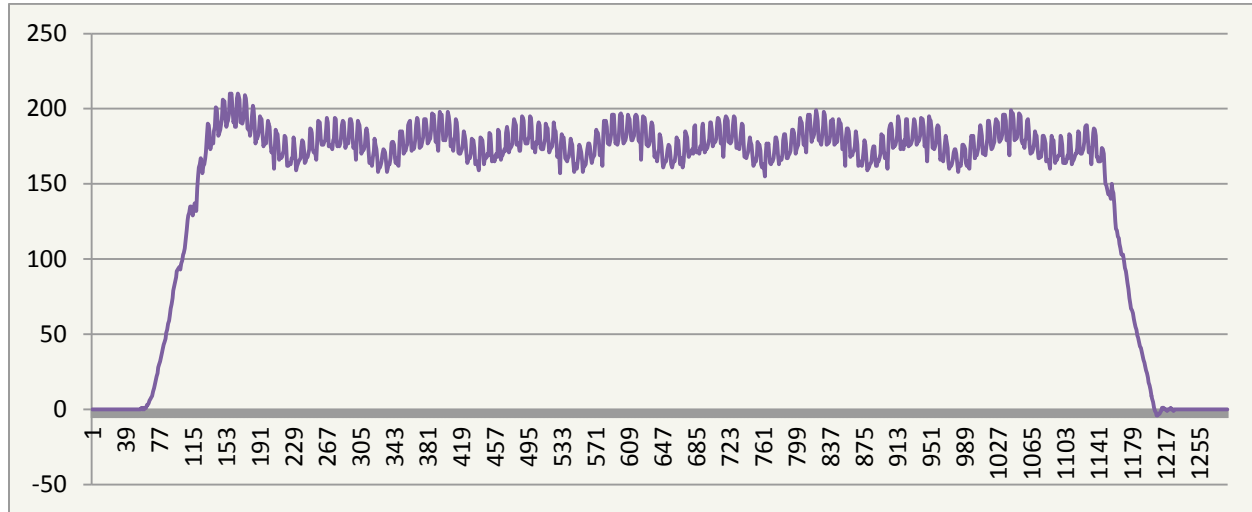
r_C – intenzitet vektora $\mathbf{r}_C$,

φ_C – ugao koji vektor $\mathbf{r}_C$ zaklapa sa x_R -osom.

Na ovaj način pokrivena su sva moguća kretanja robota. Greška ovog proračuna je kumulativna ali za frekvenciju smplovanja od 100Hz i maksimalnom brzinom robota, najgora greška je u najgorem slučaju reda veličina 10^{-10} m zbog čega se smatra zanemarljivom.

Problem sa Omni-direkcionim točkom

Kao što je rečeno u poglavlju Noseća ploča i odometrija, omni-direkcionni točak korišćen na robotu nije bio visokog kvaliteta što na samom početku nije razmatrano kao potencijalni problem jer se nije očekivalo da će rezultati biti toliko loši. Pre svega radius točka nije bio konstantan i postojali su veliki prelazi sa jednog malog točka na drugi. Zbog ovog je dolazilo do nepredvidivog proklizavanja točka što je rezultovalo lošom ponovljivosti robota. Na slici 25 je prikazan dijagram snimljenje brzine okretanja točka posle pet punih rotacija robota.



Slika 25 – Dijagram očitanih vrednosti sa enkodera za pet punih rotacija robota: x-osa – broj sempla (100Hz); y-osa – broj koraka za odgovarajući sempl.

Može se primetiti da je karakteristika vrlo loša i da nešto nije u redu. Razlog zašto dijagram ovako izgleda ne mora biti samo zbog omni-direkcionog točka. Ove je moguće da se nalaze i uticaji loše regulacije kao i nedeovoljno tačne matematičke pretpostavke iz prethodnog segmenta. Kako se očima moglo videti da omni-direkcionni točak ne dolazi u isti položaj nakon više puštanja robota, druge dve opcije nisu razmatrane kao glavni problem. Nakon velikog broja puštanja primećeno je da je ponovljivost u opsegu oko $\pm 3^\circ$.

Glavnim razlogom zašto do ovog dolazi smatran je nedovoljno dobar prelaz sa točkića na točkić, ove prelaze je moguće uočiti i na samom dijagramu. Zbog toga je pokušano ovaj problem rešiti korišćenjem niskopropusnog filtera. Filter je primenjen na ubrzanja. Nakon filtracije sam dijagram izgleda daleko bolje ali rezultat nije mnogo poboljšan. Sledeća stvar koja je probana jeste da se obezbedi veći pritisak točka na podlogu čime dolazi do povećanja trenja ali je ovo negativno uticalo na rezultat.

Zbog nedostatka vremena ovaj problem nije rešen do kraja. Pretpostavka je da bi kvalitetniji omni-direkcionni točak pozitivno uticao na rezultat kao i kvalitetnija regulacija. Ovde se može primetiti da ovakav sistem nije robusan jer u trenutku kada se izgubi trenje bilo kog odometriskog točka dolazi do netačnog proračuna pozicije i orijentacije sistema što je nedopustivo.

Ideja za kretanje robota sa preprekama

Model opisan u poglavlju Kretanje robota sa velikim ograničenjima, nailazi na velike probleme ako ne postoji mogućnost da se translacija duž prave RR1 izvrši, tačnije ako se prilikom kretanja robota zaključi da na tom putu postoji prepreka. Ovo bi značilo da je potrebno promeniti trajektoriju. Za ovaj model bi se mogao definisati set pravila koja bi robot poštovao u ovakvim trenucima ali su sva moguća pravila daleko od nečega što bi bilo optimalno. Jedno od pravila bi moglo da bude da se robot zaustavi i sačeka da se prepreka pomeri, ali ovde se dolazi do problema ako je prepreka fiksna. Još jedno od pravila bi moglo da bude da se doda nova tačka koja je van prepreke, u koju bi robot prvo otišao, pa zatim u željenu tačku. Jedno od glavnih pitanja ovde jeste gde postaviti novu tačku.

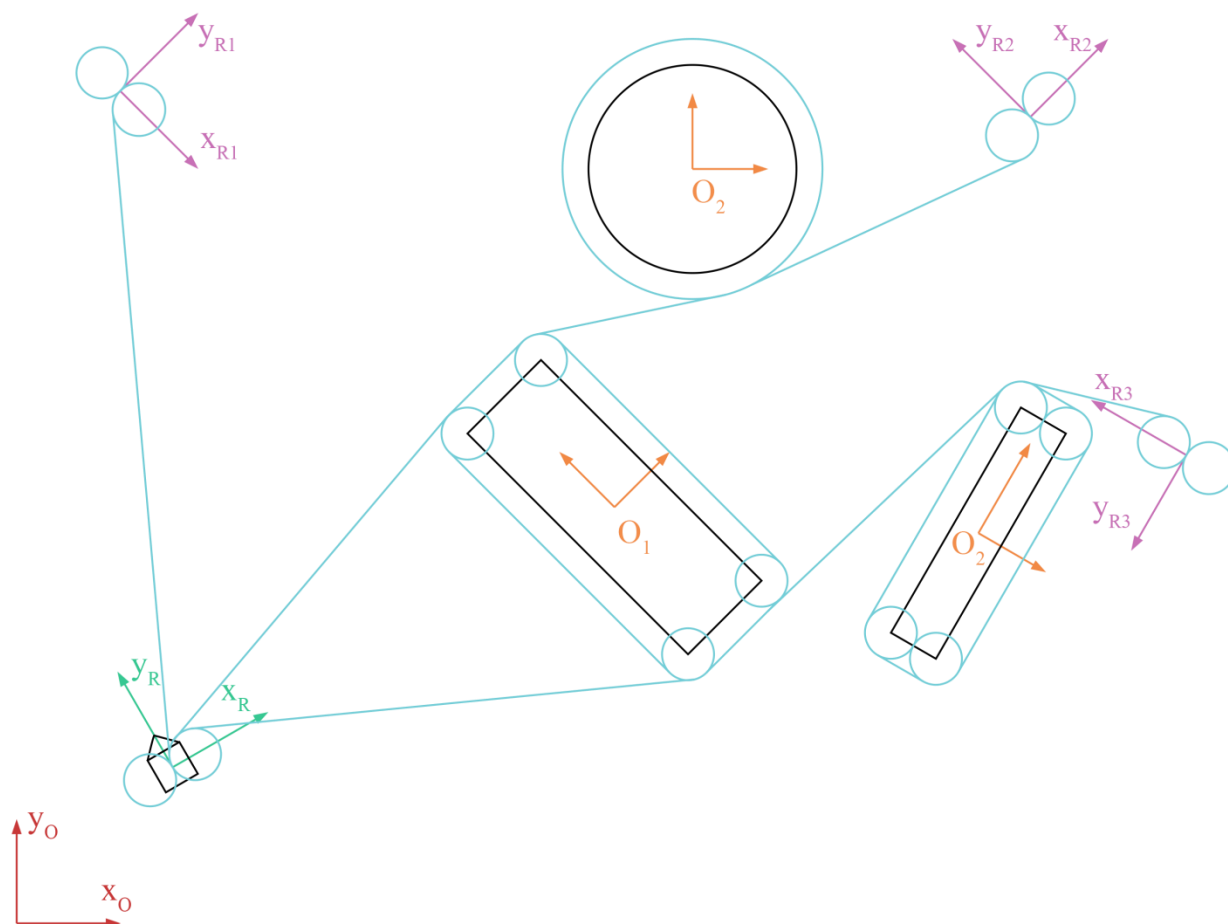
Lako se primećuje da ovaj model ima velike nedostatke ako u bilo kom trenutku vremena treba da dođe do dinamičke promene trajektorije. Zbog toga će se u ovom poglavlju predložiti jednostavan model koji rešava neke od ovih problema. Model koji se predlaže za vreme pisanja ovog projekta nije eksperimentalno testiran i nije u potpunosti dovršen zbog čega su obrađene samo prepreke koje se mogu predstaviti kao kružnica ili pravougaonik i nije dat algoritam za određivanje najkraće putanje i kriterijum minimalnog vremena.

Abstrakt

Robot se kreće po ravni sa preprekama. Prepreke mogu biti fiksne (ne menjaju svoj položaj i orijentaciju) ili dinamične. Potrebne informacije o prepreci su:

1. Položaj i orijentacija prepreke u odnosu na inercijalni koordinatni sistem,
2. Veličina prepreke.

Potrebno je odrediti najkraći put do željenog položaja tako da trajektorija bude neprekidna funkcija. Na ovaj način obezbeđuje se to da robot ni u jednom trenutku vremena ne stoji u mestu već uvek smanjuje relativno rastojanje od željenog položaja. Tri moguća slučaja ilustrovana su na slici 26.



Slika 26 – Ilustracija predloženog modela za kretanje robota u ravni sa preprekama do jedne od tri moguća položaja

U pitanju su tri moguće putanje kojima robot može iz trenutnog položaja da stigne u željeni. Sada je potrebno razviti model koji ove trajektorije može da pronađe.

Sinteza tangente na dve kružnice

U ovom poglavlju biće predstavljen matematički model koji će se kasnije koristiti za određivanje putanja prikazanih na slici 26.

Posmatra se inercijalni koordinatni sistem ΣO unutar čijeg se domena (X_O, Y_O) nalaze dve kružnice sa centima $C_{ST1}(X_{ST1}, Y_{ST1})$ i $C_{ST2}(X_{ST2}, Y_{ST2})$. Poluprečnici ovih kružnica su R_{ST1} i R_{ST2} respektivno. Uslov je da kružnice ne mogu biti koncentrične i da se manja kružnica (ako postoji) ne može nalaziti unutar veće tako da se ne seku. Ako se kružnice dodiruju u jednoj tački, taj dodir mora biti spoljašnji, kružnice se mogu seći, a ako se kružnice ni ne dodiruju ni ne seku, onda se moraju nalaziti jedna van druge. Iz tog razloga mora biti ispunjen makar jedan od uslova:

1. Da bi se kružnice spolja dodirivale u jednoj tački potreban i dovoljan uslov je da zbir poluprečnika kružnica bude jednak rastojanju između njihovih centara:

$$\sqrt{(X_{ST2} - X_{ST1})^2 + (Y_{ST2} - Y_{ST1})^2} = R_{ST1} + R_{ST2}, \quad (42)$$

2. Ako se kružnice seku potrebno je i dovoljno da zbir rastojanja između centara i poluprečnika manje kružnice bude veći od poluprečnika veće kružnice. Ako su poluprečnici iste veličine nije bitno koji će se u jednačinu (43) ubaciti kao manji:

$$\sqrt{(X_{ST2} - X_{ST1})^2 + (Y_{ST2} - Y_{ST1})^2} + R_{STmanje} > R_{STveće}, \quad (43)$$

3. Ako se kružnice ne dodiruju i ne seku potrebno je i dovoljno da rastojanje između centara bude veće od zbira poluprečnika kružnica:

$$\sqrt{(X_{ST2} - X_{ST1})^2 + (Y_{ST2} - Y_{ST1})^2} > R_{ST1} + R_{ST2}. \quad (44)$$

U nastavku razviće se samo matematički model za 3. slučaj jer on funkcioniše i u druga dva slučaja. U ovom slučaju postoje 4 tangente dok su jednačine ove dve kružnice:

$$(x - X_{ST1})^2 + (y - Y_{ST1})^2 - R_{ST1}^2 = 0, \quad (45)$$

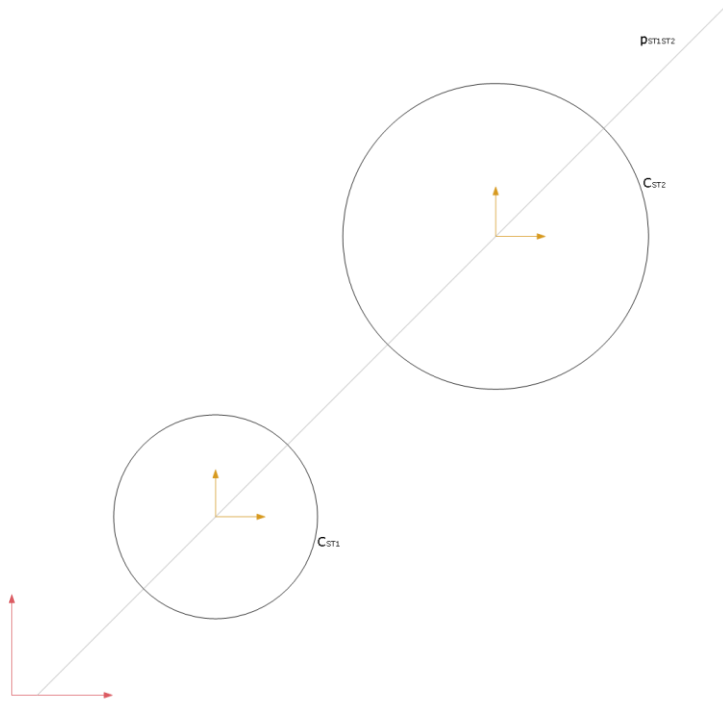
$$(x - X_{ST2})^2 + (y - Y_{ST2})^2 - R_{ST2}^2 = 0. \quad (46)$$

Uvodi se prva pomoćna prava p_{ST1ST2} (slika 27) koja prolazi kroz centre kružnica:

$$p_{ST1ST2} \rightarrow y_{ST1ST2} = k_{ST1ST2}x_{ST1ST2} + n_{ST1ST2}, \quad (47)$$

$$k_{ST1ST2} = \frac{Y_{ST2} - Y_{ST1}}{X_{ST2} - X_{ST1}}, \quad (48)$$

$$n_{ST1ST2} = Y_{ST1} - k_{ST1ST2}X_{ST1}. \quad (49)$$



Slika 27 – Ilustracija sinteze tangente na dve kružnice, prava koja prolazi kroz centre.

Sada se nalaze još dve pomoćne prave (slika 28) koje su normalne na p_{ST1ST2} i prolaze kroz centre kružnica:

$$p_{ST1norm} \rightarrow y_{ST1norm} = k_{ST1norm}x_{ST1norm} + n_{ST1norm}, \quad (50)$$

$$p_{ST2norm} \rightarrow y_{ST2norm} = k_{ST2norm}x_{ST2norm} + n_{ST2norm}, \quad (51)$$

$$\varphi_{p_{ST1ST2}} = \text{atan}\left(\frac{Y_{ST2} - Y_{ST1}}{X_{ST2} - X_{ST1}}\right), \quad (52)$$

$$X_{ST1norm} = X_{ST1} + \cos\left(\frac{3\pi}{2} + \varphi_{p_{ST1ST2}}\right), \quad (53)$$

$$Y_{ST1norm} = Y_{ST1} + \sin\left(\frac{3\pi}{2} + \varphi_{p_{ST1ST2}}\right), \quad (54)$$

$$X_{ST2norm} = X_{ST2} + \cos\left(\frac{3\pi}{2} + \varphi_{p_{ST1ST2}}\right), \quad (55)$$

$$Y_{ST2norm} = Y_{ST2} + \sin\left(\frac{3\pi}{2} + \varphi_{p_{ST1ST2}}\right), \quad (56)$$

$$k_{ST1norm} = \frac{Y_{ST1norm} - Y_{ST1}}{X_{ST1norm} - X_{ST1}}, \quad (57)$$

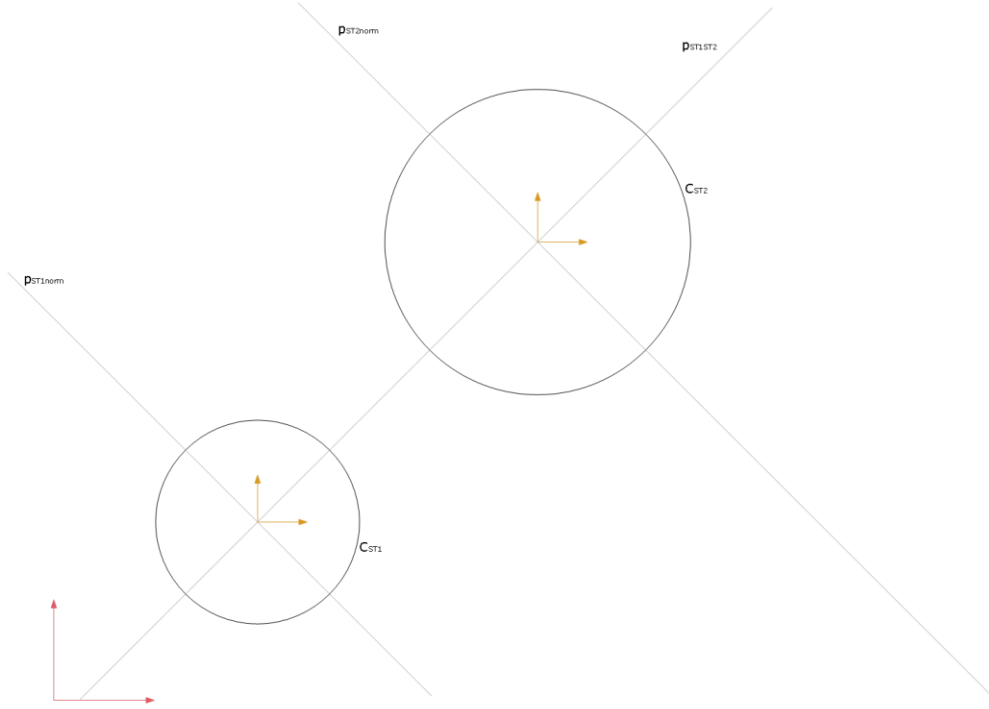
$$n_{ST1norm} = Y_{ST1} - k_{ST1norm}X_{ST1}, \quad (58)$$

$$k_{ST2norm} = \frac{Y_{ST2norm} - Y_{ST2}}{X_{ST2norm} - X_{ST2}}, \quad (59)$$

$$n_{ST2norm} = Y_{ST2} - k_{ST2norm}X_{ST2}, \quad (60)$$

gde je:

$\varphi_{p_{ST1ST2}}$ – ugao koji prava p_{ST1ST2} zaklapa sa x_0 -osom,



Slika 28 – Ilustracija sinteze tangente na dve kružnice, prave koje prolaze kroz centre kružnica i normalne su na p_{ST1ST2} .

Prva stvar koja će se sada uraditi jeste pronalaženje unutrašnjih tangenti gde je prvi korak pronalaženje tačke u kojima preve seku kružnicu, uvrštavanjem jednačine pravih u jednačine kružnica. Ovde će se ponaci tačke samo za jednu kružnicu jer za drugu važe iste jednačine samo sa odgovarajućim parametrima:

$$p_{ST1norm} \rightarrow y_{ST1norm} = k_{ST1norm}x_{ST1norm} + n_{ST1norm}, \quad (61)$$

$$(x - X_{ST1})^2 + (y - Y_{ST1})^2 - R_{ST1}^2 = 0. \quad (62)$$

Uvrštanjem (61) u (62) dobija se:

$$x^2(1 + k_{ST1norm}^2) + x(2k_{ST1norm}n_{ST1norm} - 2X_{ST1} - 2k_{ST1norm}Y_{ST1}) + (X_{ST1}^2 + Y_{ST1}^2 + n_{ST1norm}^2 - 2n_{ST1norm}Y_{ST1} - R_{ST1}^2) = 0, \quad (63)$$

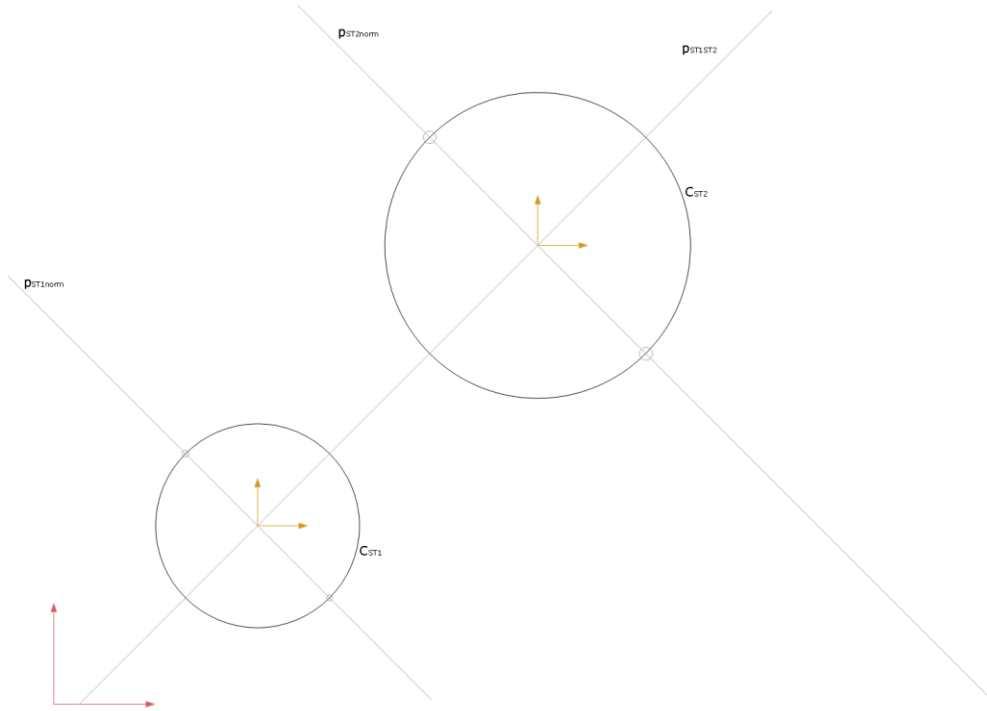
$$a_{ST1norm} = 1 + k_{ST1norm}^2, \quad (64)$$

$$b_{ST1norm} = 2k_{ST1norm}n_{ST1norm} - 2X_{ST1} - 2k_{ST1norm}Y_{ST1}, \quad (65)$$

$$c_{ST1norm} = X_{ST1}^2 + Y_{ST1}^2 + n_{ST1norm}^2 - 2n_{ST1norm}Y_{ST1} - R_{ST1}^2, \quad (66)$$

$$X_{ST1norm_{1/2}} = \frac{-b_{ST1norm} \pm \sqrt{b_{ST1norm}^2 - 4a_{ST1norm}c_{ST1norm}}}{2a_{ST1norm}}, \quad (67)$$

$$Y_{ST1norm_{1/2}} = k_{ST1norm}X_{ST1norm_{1/2}} + n_{ST1norm}. \quad (68)$$



Slika 29 – Tače preseka normala i kružnica.

Naći će se još jedna pomoćna prava pomoću koje će se pronaći žiža što su svi potrebni parametri da bi se odredile unutrašnje tangente. Uvrštavanjem tačaka $(X_{ST1norm_1}, Y_{ST1norm_1})$ i $(X_{ST2norm_2}, Y_{ST2norm_2})$ ili tačaka $(X_{ST1norm_2}, Y_{ST1norm_2})$ i $(X_{ST2norm_1}, Y_{ST2norm_1})$ u jednačinu prave, dobija se prava p_{z1} koja sa p_{ST1ST2} daje žižnu tačku kroz koju prolaze unutrašnje tangente.

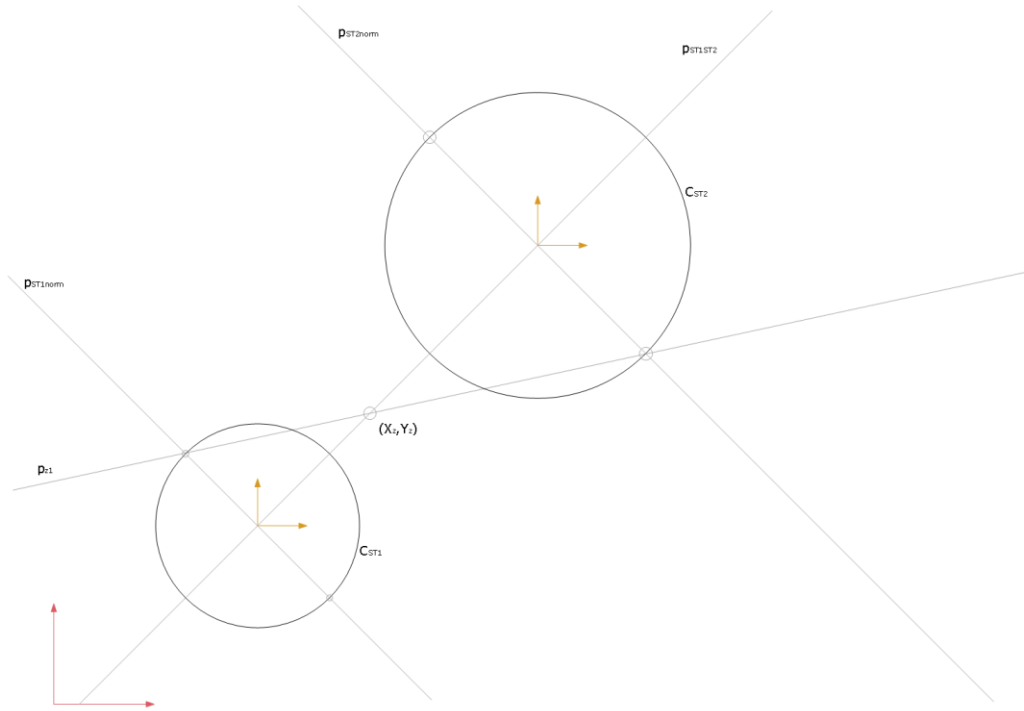
$$p_{z1} \rightarrow y_{z1} = k_{z1}x_{z1} + n_{z1}, \quad (69)$$

$$k_{z1} = \frac{Y_{ST2norm_2} - Y_{ST1norm_1}}{X_{ST2norm_2} - X_{ST1norm_1}}, \quad (70)$$

$$n_{z1} = Y_{ST1norm_1} - k_{z1}X_{ST1norm_1}, \quad (71)$$

$$X_z = \frac{n_{z1} - n_{ST1ST2}}{k_{ST1ST2} - k_{z1}}, \quad (72)$$

$$Y_z = k_{z1}X_z + n_{z1}. \quad (73)$$



Slika 30 – Određivanje žirže

Unutrašnje tangente će prolaziti kroz žirnu tačku i dodirivati obe kružnice. Ovo znači da je potrebno pronaći prave čije jednačine zadovoljavaju ove uslove. U pitanju je vrlo složen analitički problem za čije rešavanje će se koristiti program pod nazivom *Mupad*. U uslov tangencnosti prave na kružnicu (74) uvrštava se prava koja prolazi kroz žiru (75):

$$r^2(1 + k^2) - (kp - q + n)^2 = 0, \quad (74)$$

$$Y_z = k_{tin_{1,2}}X_z + n_{tin_{1,2}}, \quad (75)$$

$$R_{ST1}^2(1 + k_{tin_{1,2}}^2) - (k_{tin_{1,2}}X_{ST1} - Y_{ST1} + Y_z - k_{tin_{1,2}}X_z)^2 = 0, \quad (76)$$

$$\forall (\sigma_3 \neq \sigma_2),$$

$$k_{tin_1} = \frac{X_{ST1}Y_z - X_{ST1}Y_{ST1} + X_zY_{ST1} - X_zY_z + \sigma_1}{\sigma_4}, \quad (77)$$

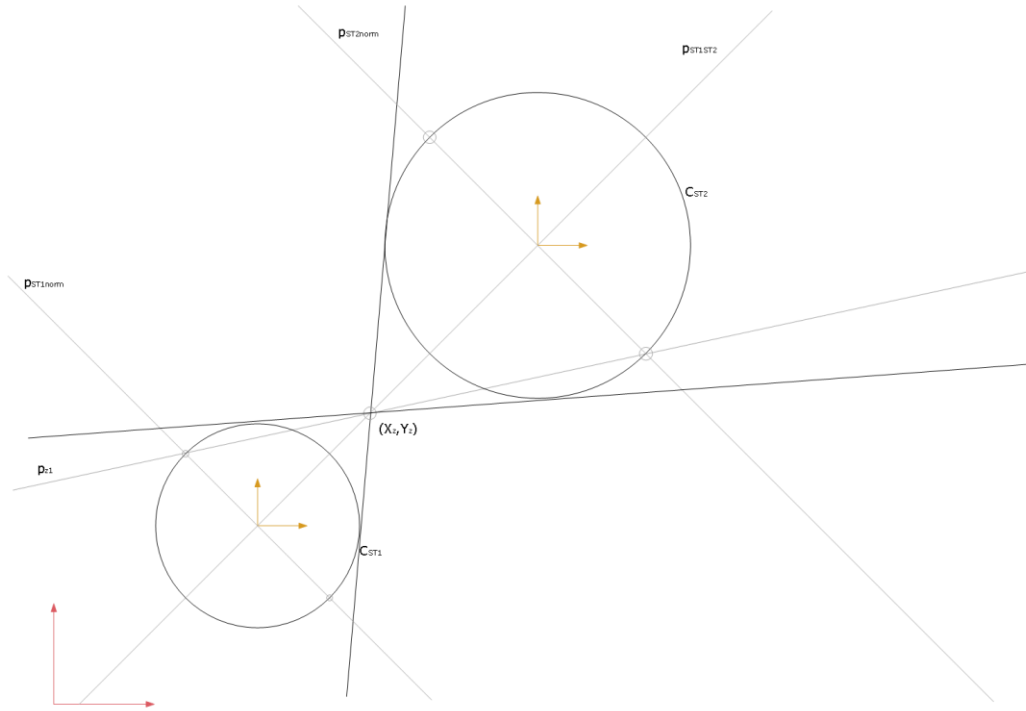
$$k_{tin_2} = -\frac{X_{ST1}Y_{ST1} - X_{ST1}Y_z - X_zY_{ST1} + X_zY_z + \sigma_1}{\sigma_4},$$

$$\sigma_1 = R_{ST1} \sqrt{X_{ST1}^2 - 2X_{ST1}X_z + X_z^2 + Y_{ST1}^2 - 2Y_{ST1}Y_z + Y_z^2 - R_{ST1}^2}, \quad (78)$$

$$\sigma_4 = -X_{ST1}^2 + 2X_{ST1}X_z - X_z^2 + R_{ST1}^2,$$

$$n_{tin_{1,2}} = Y_z - k_{tin_{1,2}}X_z, \quad (79)$$

$$p_{tin_{1,2}} \rightarrow y = k_{tin_{1,2}}x + n_{tin_{1,2}}. \quad (80)$$



Slika 31 – Unutrašnje tangente.

Ostalo je još da se odrede tačke u kojima ove prave dodiruju kružnice za čiji će se analitički proračun ponovo iskoristiti *Mupad*:

$$X_{STmp_{tin_l}} = \frac{X_{STm} + k_{tin_l} Y_{STm} + \sigma_1 - k_{tin_l} n_{tin_l}}{k_{tin_l}^2 + 1}, \quad (81)$$

$$l = 1, 2; \quad m = 1, 2;$$

$$\sigma_1 = \sqrt{k_{tin_l}^2 R_{STm}^2 - k_{tin_l}^2 X_{STm}^2 - 2k_{tin_l} n_{tin_l} X_{STm} + 2k_{tin_l} X_{STm} Y_{STm} - n_{tin_l}^2 + 2n_{tin_l} Y_{STm} + R_{STm}^2 - Y_{STm}^2}, \quad (82)$$

$$Y_{STmp_{tin_l}} = k_{tin_l} X_{STmp_{tin_l}} + n_{tin_l}. \quad (83)$$

Ovim su određeni svi potrebni parametri za unutrašnje tangente i može se preći na spoljašnje. Važno je napomenuti da za slučajeve 1. i 2. sa početka poglavlja neće postojati unutrašnje tangente, tačnije za slučaj 2. žiža će se nalaziti u kontaktnoj tački kružnica i obe tangente će prolaziti kroz tu tačku i dodirivati kružnice u toj tački (što je trivijalno) dok u 1. slučaju unutrašnjih tangenti neće biti. U ovom projektu simulacijom je dokazan samo slučaj 3.

Kod spoljašnjih tangenti žiža će se nalaziti na pravoj p_{ST1ST2} negde van prostora između kružnica, ukoliko su poluprečnici različitih veličina, dok žiža neće postojati ukoliko su poluprečnici isti jer će tangente biti paralelne. Obradiće se samo slučaj kada su poluprečnici različitih veličina jer kada su istih, tangente će biti paralelne sa p_{ST1ST2} izmeštene za veličinu poluprečnika.

Prvo se unutar veće kružnice dodaje pomoćna koncentrična kružnica čiji je poluprečnik jednak razlici poluprečnika veće i manje kružnice i njena jednačina će biti:

$$c_{h1} \rightarrow (x - X_{STveće})^2 + (y - Y_{STveće})^2 - (R_{STveće} - R_{STmanje})^2 = 0. \quad (84)$$

Sada je potrebno pronaći prave koje prolaze kroz centar manje kružnice i tangentne su na c_{h1} . Za ovo se koriste jednačine (77) i (79) gde je potrebno zameniti promenljive odgovarajućim; $Y_z \rightarrow Y_{STmanje}$, $X_z \rightarrow X_{STmanje}$, $R_{ST1} \rightarrow (R_{STveće} - R_{STmanje})$, $X_{ST1} \rightarrow X_{STveće}$, $Y_{ST1} \rightarrow Y_{STveće}$, čime se dobijaju prave:

$$p_{tc_{h1,2}} \rightarrow y = k_{c_{h1,2}} x + n_{c_{h1,2}}. \quad (85)$$

Zatim je potrebno pronaći tačke u kojima se prave (85) dodiruju sa kružnicom (84), a to se dobija uz pomoć jednačina (81) i (83). Zamenom odgovarajućih promenljivih dobijaju se tačke $(X_{tc_{h1,2}}, Y_{tc_{h1,2}})$. Potom se traže nove pomoćne prave koje prolaze kroz novodobijene tačke i centar veće kružnice.

$$k_{p_{tcnorm_{1,2}}} = \frac{Y_{tc_{h1,2}} - Y_{STveće}}{X_{tc_{h1,2}} - X_{STveće}}, \quad (86)$$

$$n_{p_{tcnorm_{1,2}}} = Y_{STveće} - k_{p_{tcnorm_{1,2}}} X_{STveće}. \quad (87)$$

Presek svake od ovih pravi sa većom kružnicom kao rezultat daje dve tačke, a koristi se onaj set tačaka koji je bliži centru manje kružnice:

$$X_{STvećeptout_{1,2}} = \frac{X_{STveće} + k_{p_{tcnorm_{1,2}}} Y_{STveće} + \sigma_1 - k_{p_{tcnorm_{1,2}}} n_{p_{tcnorm_{1,2}}}}{k_{p_{tcnorm_{1,2}}}^2 + 1}, \quad (88)$$

$$X_{STvećeptout_{3,4}} = \frac{X_{STveće} + k_{p_{tcnorm_{1,2}}} Y_{STveće} - \sigma_1 - k_{p_{tcnorm_{1,2}}} n_{p_{tcnorm_{1,2}}}}{k_{p_{tcnorm_{1,2}}}^2 + 1},$$

$$\sigma_1 = \sqrt{\frac{k_{p_{tcnorm_{1,2}}}^2 R_{STveće}^2 - k_{p_{tcnorm_{1,2}}}^2 X_{STveće}^2 - 2k_{p_{tcnorm_{1,2}}} n_{p_{tcnorm_{1,2}}} X_{STveće} +}{2k_{p_{tcnorm_{1,2}}} X_{STveće} Y_{STveće} - n_{p_{tcnorm_{1,2}}}^2 + 2n_{p_{tcnorm_{1,2}}} Y_{STveće} + R_{STveće}^2 - Y_{STveće}^2}}, \quad (89)$$

$$Y_{STvećeptout_{1,2,3,4}} = k_{p_{tcnorm_{1,2}}} X_{STvećeptout_{1,2,3,4}} + n_{p_{tcnorm_{1,2}}}. \quad (90)$$

U ovim tačkama spoljašnje tangente dodiruju veću kružnicu (treba voditi računa o tome koji se set tačaka koristi pošto je jedno rešenje trivijalno). Uvrstanjem ovih tačaka u jednačinu prave i jednačinu za uslov tangentnosti, dobijaju se spoljašnje tangente. Ove prave mogu se dobiti uvrstanjem odgovarajućih parametara u jednačine (74) - (80) s' time da će jedno od rešenja biti trivijalno. Preva koja se seče sa pravom p_{ST1ST2} u prostoru između kružnica, je prava koja se odbacuje što će od ukupno četiri dobijene prave dati dve prave koje su ujedno i tangente na kružnice.

$$p_{tout_{1,2}} \rightarrow y = k_{tout_{1,2}} x + n_{tout_{1,2}}. \quad (91)$$

Kako tačka dodira ovih pravih i veće kružnice već postoji, potrebno je samo uz pomoć jednačina (81) – (83) odrediti tačke dodira pravih sa manjom kružnicom.

Formiranje domena oko prepreke

Kao što je rečeno na početku, sve realne prepreke se aproksimiraju u obliku kružnice ili pravougaonika. Ova dva tipa figure predstavljaju površine unutar kojih se nijedna tačka robota ne sme naći ni u jednom trenutku vremena. Kako ove figure predstavljaju realna geometrijska tela, ako prethodni uslov nije zadovoljen, to znači da je došlo do kontakta između robota i tela posmatranog kao prepreka. Zbog toga prepreka ima svoje atribute:

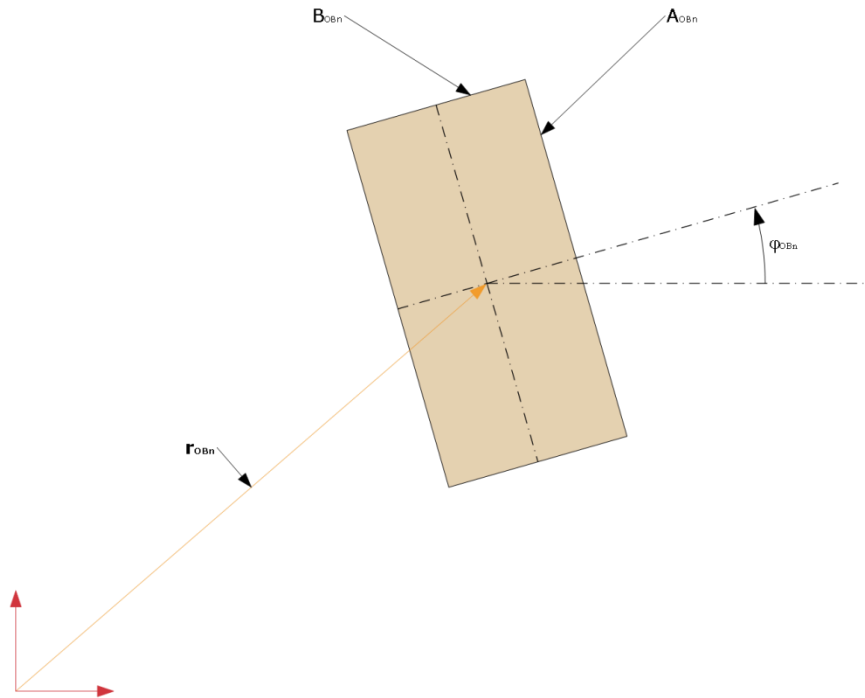
1. Prepreka ima svoj položaj i orijentaciju u inercijalnom koordinatnom sistemu (kod kružne prepreke orijentacija nije od značaja),
2. Prepreka ne mora imati fiksni položaj već može imati brzinu i ubrzanje,
3. Prepreka ne mora imati fiksnu orijentaciju već može imati ugaonu brzinu i ubrzanje,

4. Prepreka ima dimenzije koje opisuju površinu unutar koje se ni u jednom trenutku ne sme naći ni jedna tačka robota.

Zbog ovih uslova formira se domen oko prepreke koji opisuje kriva linija. Ova linija predstavlja najbliže rastojanje od centra prepreke na kojoj sme da se nađe geometrijski centar robota.

Formiranje domena oko kružne prepreke se neće razmatrati. Za kružnu prepreku potrebno je da se poznaje njen poluprečnik i položaj, nakon čega se formira zamišljena kružnica čiji je poluprečnik unutar domena kretanja robota.

Sa druge strane pravougaonu prepreku je potrebno opisati. Ona ima četiri stranice koje predstavljaju spojene duži. Postoje dva para paralelnih duži istih dužina. Ako su duži u oba para duži istih dužina, onda se takva figura naziva kvadrat. Da bi se domen formirao oko pravougaone prepreke, za prepreku je potrebno znati parametre koji su prikazani na slici 32,



Slika 32 – Parametri koje je potrebno poznavati za prepreku.

gde je:

$\mathbf{r}_{OBn}$ – vektor položaja centra n-te prepreke,

φ_{OBn} – ugao koji x_{OBn} -osa zaklapa sa x_O -osom (orijentacija prepreke),

A_{OBn} – Dužina duže stranice n-te pravougaone prepreke,

B_{OBn} – Dužina kraće stranice n-te pravougaone prepreke,

n – dodeljeni broj prepreke.

Uz ove parametre moguće je odrediti položaje ćoškova prepreke. Ovde je bitno primetiti da je moguće poznavati i položaje tri od četiri ćoška prepreke, gde je inverznim putem moguće doći do centra prepreke. Takođe ove vrednosti je moguće dobiti preko transformacije koordinatnih sistema ako su poznati samo relativni parametri prepreke.

Sa X_{OBnCm} će se označiti x koordinata m-tog ćoška na n-toj prepreci, gde je $m=1,2,3,4$, a sa Y_{OBnCm} će se označiti y koordinata m-tog ćoška n-te prepreke. Radi opštosti uzeće se da je $m=1$ za ćošak u prvom kvadrantu koordinatnog sistema prepreke, $m=2$ za ćošak u drugom kvadrantu itd.

$$X_{OBn_{Cm}} = X_{OBn} + \frac{B_{OBn}}{2} \cos(\varphi_{OBn}) - \frac{A_{OBn}}{2} \sin(\varphi_{OBn}), \quad (92)$$

$$Y_{OBn_{Cm}} = Y_{OBn} + \frac{B_{OBn}}{2} \sin(\varphi_{OBn}) + \frac{A_{OBn}}{2} \cos(\varphi_{OBn}). \quad (93)$$

U ćoškovce se postavljaju zamišljene kružnice željenog poluprečnika sa ograničenjem da poluprečnik ne sme biti veći od kraće stranice prepreke, jednak nuli i sve četiri kružnice moraju imati isti poluprečnik.

$$(x - X_{OBn_{Cm}})^2 + (y - Y_{OBn_{Cm}})^2 = R_{OBn_{Cm}}^2. \quad (94)$$

Realizacija

Kao što je rečeno na početku ovaj model je u fazi razvoja i ne postoje simulaciona rešenja već samo ideja realizacije koju je neophodno razviti i testirati. Kada postoje domeni kretanja oko svih tela moguće je formirati putanju kretanja robota tako da put između te dve tačke bude minimalan. Kako su svi parametri putanje dinamični, putanju je moguće modifikovati prilikom kretanja i na taj način, umesto minimalnog puta, može se iskoristiti kriterijum minimalnog vremena, gde u slučaju kretanja minimalan put najčešće podrazumeva i minimalno vreme (jer je podloga ravna i od istog materijala na svakom delu puta).

Ako se obrati pažnja na sliku 26, prilikom kretanja robota iz koordinatnog sistema $\sum R$ u $\sum R_2$, ne mora se pre početka kretanja imati informacija o prepreci O_2 . Ovo je zbog toga što sistem u trenutku t^* može da izračuna sve potrebne parametre o tome kako upravljanje treba da se izvrši da bi se u trenutku $t^* + \Delta t$ našao u željenoj poziciji, sa željenom orijentacijom, čime se proces ponavlja sve dok se ne završi kretanje. Na ovaj način primećuje se da razvijanjem ovog jednostavnog modela može izvršiti optimizacija na višem nivou kao i izbegavanje sudara.

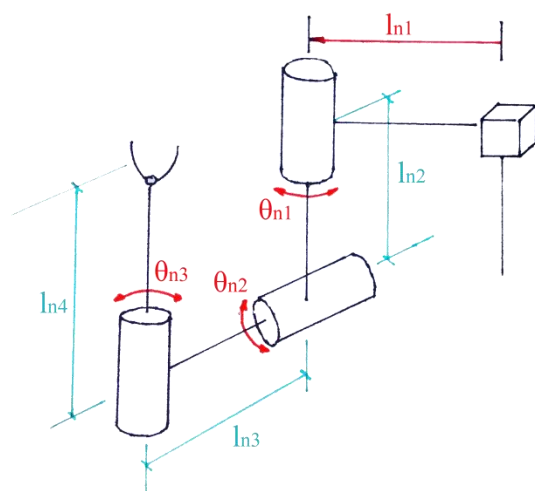
Izazovi koji nastaju jesu razvijanje algoritma za traženje mogućih puteva i formiranje matematičkog modela robota.

Robotski manipulator

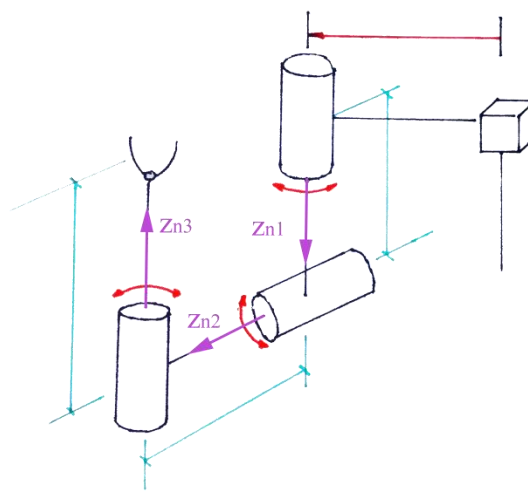
Manipulator je prikazan na slici 20. Njegov zadatak je, kao što je rečeno, da skuplja lunarne module, ostavlja ih u skladište i iz skladišta ih adekvatno ostavlja na igralište. Sve funkcije manipulatora sem jedne, rešene su vremensim intervalima, tačnije predefinisane su u kodu. Jedna specijalna funkcija manipulatora je da može automatski da cilja lunarni modul o čemu će biti reči u ovom poglavlju. Sve funkcije manipulatora podeliće se u faze.

Osnovna konfiguracija

Osnovna konfiguracija manipulatora prikazana je na slici 33 i 34 gde se smatra da su svi uglovi u zglobovima, u ovom položaju jednaki nuli. Dakle svi zglobovi će se okretati za neki pozitivan ili negativan ugao u odnosu na nulti ugao.

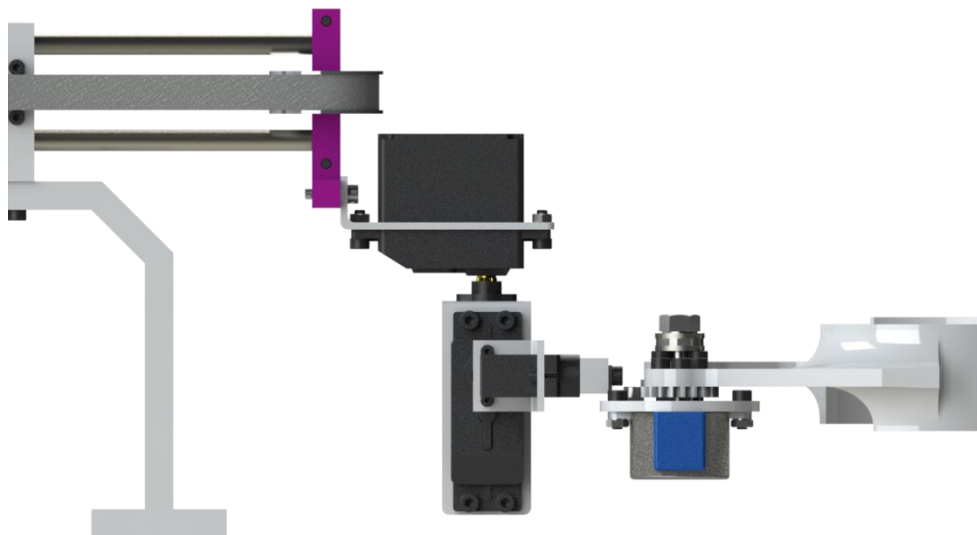


Slika 33 – Ilustracija osnovne konfiguracije robotskog manipulatora, gde je $n=1, 2$.



Slika 34 – Ose koje prolaze kroz rotacione zglobove i njihova orijentacija.

Pošto u sistemu postoje dva robotska manipulatora uvešće se set pravila kako bi upravljanje bilo jednostavnije. Nulti ugao za svaki zglob u manipulatoru postavlja se tako da manipulatori budu u konfiguraciji kao što je na slici 33 (odavde se primećuje da je ilustrovana osnovna konfiguracija desne ruke). Levi manipulator uzima $n=1$. Kroz svaki rotacioni zglob prolazi osa z_{qm} , $m=1, 2, 3$, orijentisana u pravcu osovine motora (kao što je prikazano na slici 34). Za rotacione zglobove levog manipulatora važi: ako osa z_{q11} izlazi iz papira, svi uglovi veći od nule su u pravcu negativnog matematičkog smera dok za negativne uglove važi obrnuto; ako osa z_{q12} izlazi iz papira, svi uglovi veći od nule su u pravcu negativnog matematičkog smera dok za negativne uglove važi obrnuto; ako osa z_{q13} izlazi iz papira, svi uglovi veći od nule su u pravcu negativnog matematičkog smera dok za negativne uglove važi obrnuto. Na primer, ako bi se uglovi postavili u konfiguraciju $0^\circ-90^\circ-0^\circ$ počevši od prvog ka poslednjem, dobila bi se konfiguracija prikazana na slici 35.

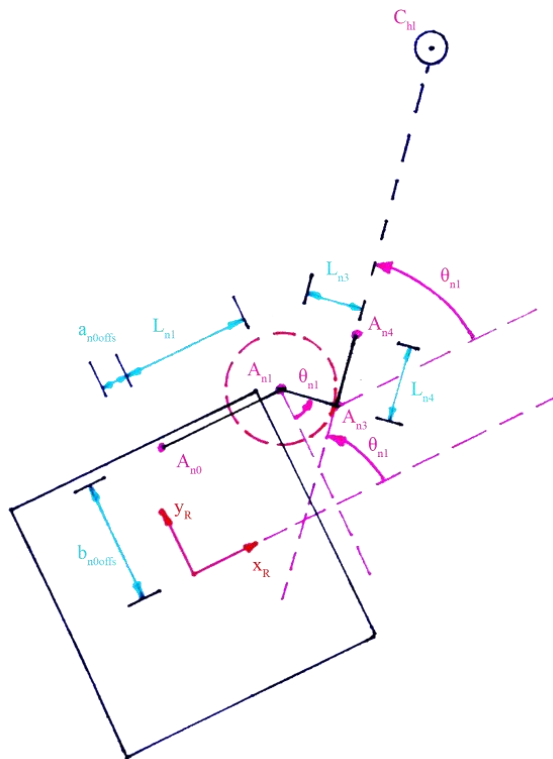


Slika 35 – Levi robotski manipulator u konfiguraciji 0° - 90° - 0° .

Za desni manipulator važe ista pravila samo što će svi pozitivni uglovi biti suprotni u odnosu na levi.

Prva faza

Podrazumeva ciljanje lunarnog modula i njegovo hvatanje. Ciljanje podrazumeva da bez obzira na orijentaciju robota i njegovo rastojanje od modula, sama hvataljka manipulatora uvek gleda prema modulu. Fizički manipulator ima svoja ograničenja kao što su minimalni i maksimalni ugao koji motor može da postigne, kao i nedozvoljene konfiguracije. Dakle potrebno je razviti matematički model koji rešava inverzni kinematski problem i upravlja motorima u skladu sa ograničenjima i mogućnostima motora.



Slika 36 – Postavljanje sistema za automatsko ciljanje lunarnog modula.

Na slici 36 je:

a_{n0offs} – rastojanje tačke A_{n0} po x_R -osi,

b_{n0offs} – rastojanje tačke A_{n0} po y_R -osi,

C_{hl} – položaj l-tog lunarnog modula na stolu u odnosu na inercijalni koordinatni sistem,

n – uzima vrednosti 1, 2.

Kako su manipulatori simetrični, u nastavku će se napraviti matematički model za levi manipulator i prokomentarisati kako će se model prilagoditi za desni. Zadatak je da se pronađe ugao θ_{11} za trenutni položaj robota i odabrani lunarni modul. Ulazni parametri u model su ΣR i C_{hl} , dok je izlazni θ_{11} .

Zamišljena kružnica sa centrom u tački A_{11} , po kojoj se kreće tačka A_{12} , opisana je jednačinom:

$$(x - X_{A_{11}})^2 + (y - Y_{A_{11}})^2 - L_{13}^2 = 0. \quad (95)$$

$$\mathbf{r}_{A_{11}} = X_{A_{11}} \mathbf{x}_R + Y_{A_{11}} \mathbf{y}_R, \quad (96)$$

$$\mathbf{r}_{A_{11}} = \mathbf{r}_R + \mathbf{r}_{RA_{10}} + \mathbf{r}_{A_{10}A_{11}}, \quad (97)$$

$$X_{A_{11}} = (X_R + a_{A_{11}} \cos(\varphi_R) - b_{A_{11}} \sin(\varphi_R) + L_{11} \cos(\varphi_R)), \quad (98)$$

$$Y_{A_{11}} = (Y_R + a_{A_{11}} \sin(\varphi_R) + b_{A_{11}} \cos(\varphi_R) + L_{11} \sin(\varphi_R)). \quad (99)$$

Sada je potrebno pronaći pravu $p_{A_{12}C_{hl}}$ koja prolazi kroz tačku C_{hl} i tangenta je na zamišljenu kružnicu uz pomoć jednačina (74) – (80), uz zamenu odgovarajućih promenljivih, čime se dobijaju parametri prave:

$$p_{A_{12}C_{hl}} \rightarrow y = k_{A_{12}C_{hl}} x + n_{A_{12}C_{hl}}, \quad (100)$$

sa obzirom da se za računanje k parametra koristi jednačina sa predznakom minus. Na mestu gde se ova prava dodiruje sa kružnicom nalazi se tačka A_{13} . Ugao θ_{11} se može dobiti i na ovaj način ali će se u nastavku ovaj ugao dobiti kao ugao između prave $p_{A_{12}C_{hl}}$ i prave opisane jediničnim vektorom $\mathbf{x}_R$:

$$p_{x_R} \rightarrow y = k_{x_R} x + n_{x_R}, \quad (101)$$

$$\theta_{11} = \text{atan}\left(\frac{k_{A_{12}C_{hl}} - k_{x_R}}{1 + k_{A_{12}C_{hl}} k_{x_R}}\right), \theta_{11} \in \left(\frac{5\pi}{6}, -\frac{\pi}{6}\right). \quad (102)$$

Ako se robot kreće, dovoljno čestim prolaskom kroz ove jednačine, hvataljka će uvek gledati prema lunarnom modulu koji se cilja. Ostalo je još da se odredi relativno rastojanje hvataljke od lunarnog modula. Za ovo je potrebo da se odredi položaj vrha hvataljke u odnosu na inercijalni koordinatni sistem.

$$\mathbf{r}_{A_{13}} = \mathbf{r}_{A_{11}} + \mathbf{r}_{A_{11}A_{13}}, \quad (103)$$

$$\mathbf{r}_{A_{11}A_{13}} = \mathbf{r}_{A_{11}} + L_{13} \cos(\theta_{11}) (-\mathbf{y}_R) + L_{13} \sin(\theta_{11}) \mathbf{x}_R,$$

$$\mathbf{r}_{A_{14}} = \mathbf{r}_{A_{13}} + \mathbf{r}_{A_{13}A_{14}}, \quad (104)$$

$$\mathbf{r}_{A_{13}A_{14}} = \mathbf{r}_{A_{13}} + L_{14} \cos(\theta_{11}) \mathbf{x}_R + L_{14} \sin(\theta_{11}) \mathbf{y}_R,$$

$$\overline{A_{14}C_{hl}} = \sqrt{(X_{C_{hl}} - X_R)^2 + (Y_{C_{hl}} - Y_R)^2},$$

gde je:

$\overline{A_{14}C_{hl}}$ – rastojanje od vrha hvataljke do centra lunarnog modula.

Kada je $\overline{A_{14}C_{hl}} \sim 0$ hvataljka se nalazi iznad lunarnog modula. Ukoliko je $\overline{A_{14}C_{hl}}(t-1) > \overline{A_{14}C_{hl}}(t)$ to znači da se hvataljka udaljava od modula, što može biti znak da je robot promašio modul.

Na osnovu ovog modela, kada robot želi da pokupi neki lunarni modul dovoljno je da zna koji je njegov položaj na igralištu, gde se nalazi lunarni modul u odnosu na inercijalni koordinatni sistem i da prođe dovoljno blizu modula kako bi ga pokupio. Ovo znači da nema potrebe za preciznim pozicioniranjem ispred lunarnog modula i robot u prolazu može da pokupi isti, bez potrebe da zaustavi kretanje. Za desnu ruku važe potpuno isti model uz to da se moraju iskoristiti odgovarajući vektori položaja i konstante.

Druga faza

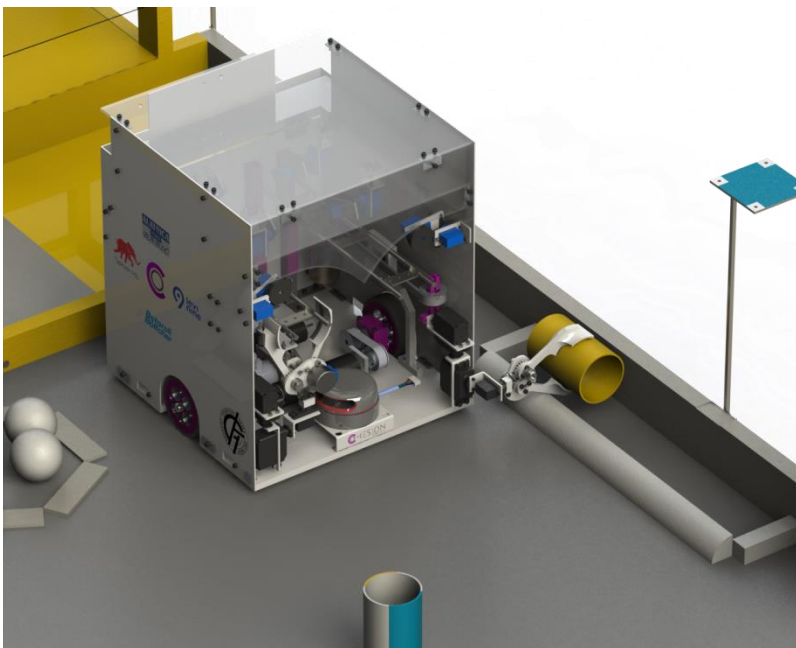
Ova faza podrazumeva ostavljanje prethodno uhvaćenog lunarnog modula u skladište. Kako na samoj hvataljci ne postoji nikakav senzor koji detektuje prisustvo lunarnog modula, neophodno je na neki način saznati da li je modul prisutan ili ne, pre nego što se manipulator vrati u robota. Ovo se radi uz pomoć senzora koji je postavljen u ravni modula kada se ruka nalazi u konfiguraciji $0^\circ-90^\circ-0$. Manipulator se dakle nakon uspešno obavljene prve faze postavlja u ovu konfiguraciju i proverava prisustvo modula. Ukoliko je modul prisutan, manipulator se postavlja u konfiguraciju $0^\circ-0^\circ-0^\circ$, nakon čega se otvaraju prsti hvataljke. U ovom trenutku se ponovo proverava prisustvo modula ali ovog puta u skladištu. Ukoliko je modul prisutan, osigurava se, čime robot zna da sigurno u tom skladištu ima jedan lunarni modul više. Ako senzor za prisustvo lunarnog modula na prvom mestu skladišta ne detektuje modul to znači da je on ispaao iz hvataljke prilikom stavljanja.

Treća faza

U ovoj fazi se lunarni modul iz skladišta ostavlja na igralište. Šta će se odvijati u ovoj fazi zavisi od toga u koju bazu je potrebno ostaviti lunarni modul.

Ostavljanje lunarnog modula u malu bazu igrališta

Potrebno je da robot bude pozicioniran paralelno sa bazom kao što je prikazano na slici 37.



Slika 37 – Prikaz ostavljanja lunarnog modula u malu bazu igrališta.

Kada je manipulator u osnovnoj konfiguraciji, prvo se otpušta mehanizam za zadržku i okretanje čunja nakon čega se zatvara hvataljka. Zatim se manipulator postavlja u konfiguraciju 90° - 90° - 90° nakon čega se hvataljka otvara i manipulator se vraća u osnovnu konfiguraciju. U međuvremenu dok je manipulator ostavljao lunarni modul, preostali lunarni moduli u skladištu se pomeraju za jedno mesto ka izlazu.

Ostavljanje lunarnog modula u veliku bazu igrališta

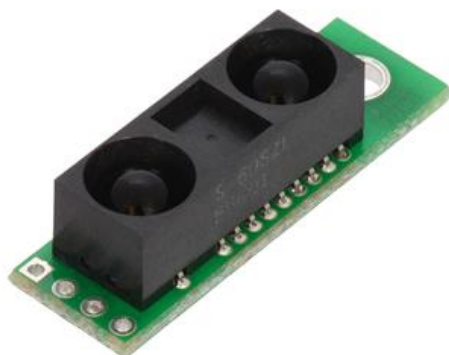
Kao i u prethodnom slučaju robot mora da bude pozicioniran paralelno sa bazom u koju se modul ostavlja. Manipulator iz osnovne konfiguracija hvata lunarni modul nakon čega se postavlja u konfiguraciju 90° - 90° - 90° . U ovom trenutku se ruka linearno pomera za 80mm i paralelno sa time se i robot translatorno pomera za 40mm. Kada su ova pomeranja izvršena, hvataljka se otvara, manipulator se vraća u osnovnu konfiguraciju i robot se translira nazad u početni položaj.

Senzori

Infracrveni senzor SHARP GP2Y0A60SZLF

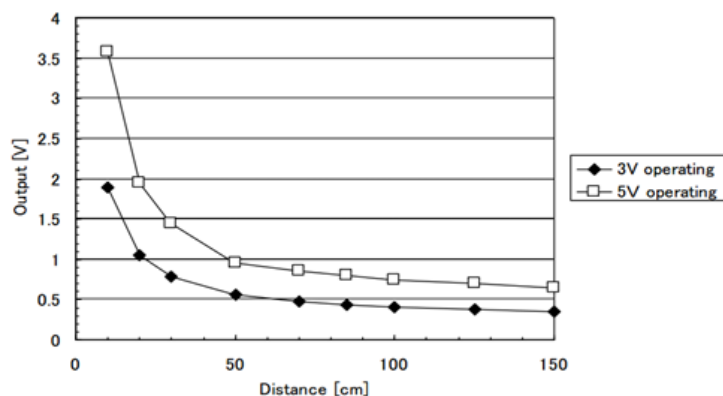
Osnovna uloga ovog senzora je detektovanje objekata i njihove udaljenosti sa bočne i zadnje strane kako bi se stekle informacije o perifernoj okolini samog robota. Ovom informacijom utiče se na dalji pravac i kretanje robota u skladu sa okolinom. Osnovne karakteristike senzora (slika 38):

- Dimenzije: 22.0 x 8.0 x 7.2 (mm)
- Analogni izlaz
- Domet senzora: od 100 do 1500 (mm)

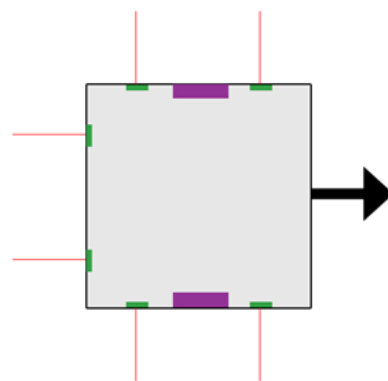


Slika 38 - SHARP GP2Y0A60SZLF

Iako izvanrednih karakteristika i dimenzija, upotreba ovog senzora se pokazala nepraktičnom. Vreme odziva ovog senzora od 20ms - 30ms, prisustvo šuma analognog signala pri većim dužinama komunikacionog kabla kao i relativna nepreciznost koja se menja u zavisnosti od daljine detektovanog objekta, učinili su upotrebu ovog senzora dosta nepouzdanom i ograničenom. Zavisnost analogne vrednosti signala na izlazu u odnosu na udaljenost objekta prikazana je na slici 39. U ovu svrhu korišćeno je šest infracrvenih senzora poredjenih kao na slici 40.



Slika 39 – Karakteristika senzora SHARP GP2Y0A60SZLF.



Slika 40 – Raspored infracrvenih senzora na robotu.

Infracrveni seznor SHARP 2D120X

Osnovna uloga ovog senzora je određivanje položaja translatorskog dela robotskog manipulatora. Uz pomoć ove konfiguracije dobija se tačna udaljenosti ruke od njenog početnog položaja. Upotrebom infracrvenog senzora za merenje

položaja ruke, umesto enkodera, izabrani su iz konstrukcionih razloga. Time je stečena veća fleksibilnost pri određivanju mesta samog senzora, smanjena je kompleksnost mehaničke konstrukcije i broj pokretnih delova.

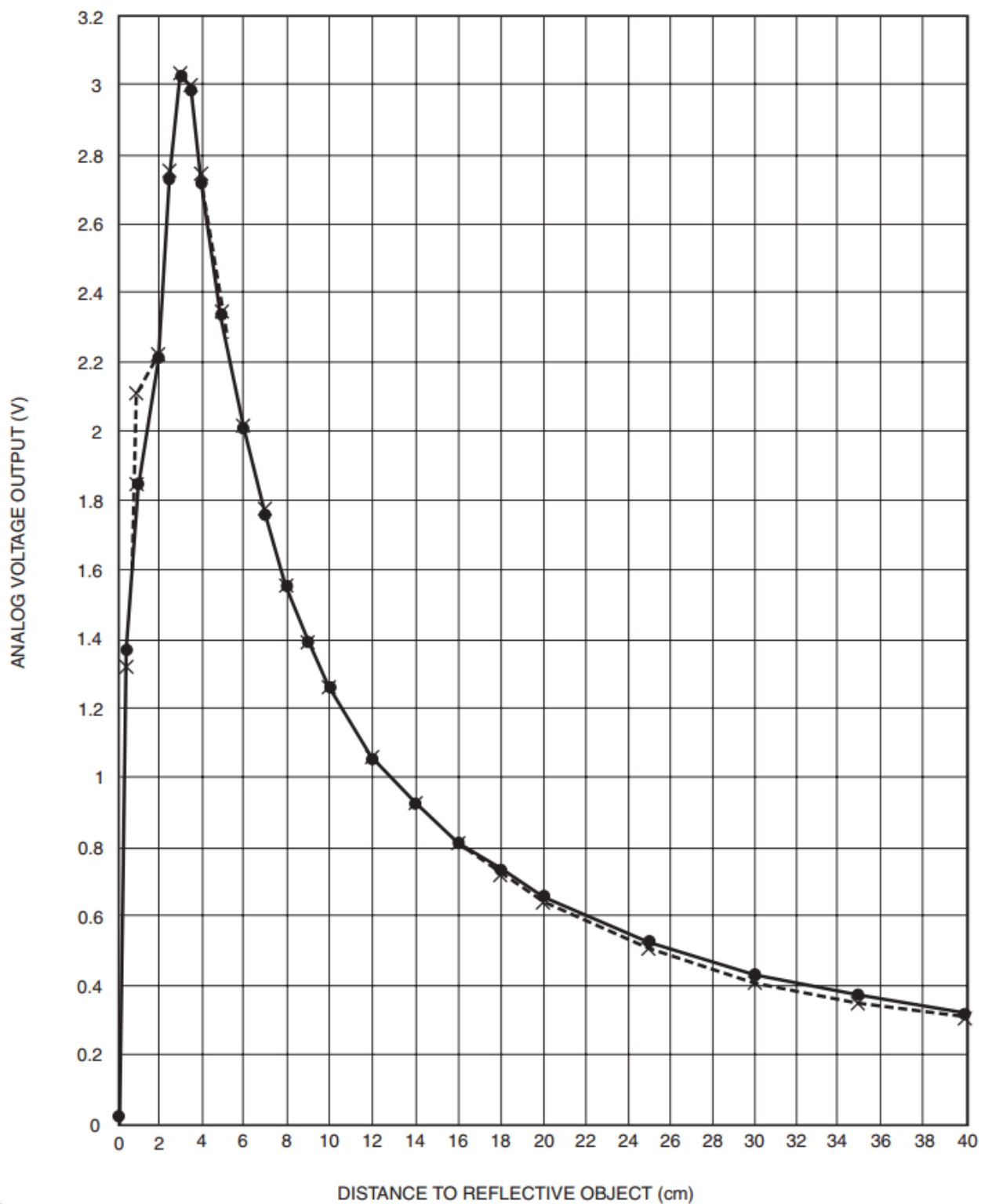


Slika 41 – SHARP 2D120X

Osnovne karakteristike senzora sa slike 41:

- Vreme odziva 40ms
- Analogni izlaz
- Domet senzora: od 40 do 300 (mm)

Pored dužeg vremena odziva, primena ovog senzora se pokazala tolerantnom zbog samog mesta primene jer je brzina kretanja translatorsnog zgloba daleko ispod brzine odziva senzora. Preciznost seznora je takođe ispunjavalo opšte uslove i nije pretsavljalo problem prilikom upravljanja translatorsnim zglobom. Zavisnost analogne vrednosti signala na izlazu u odnosu na udaljenost objekta prikazana je na slici 42.



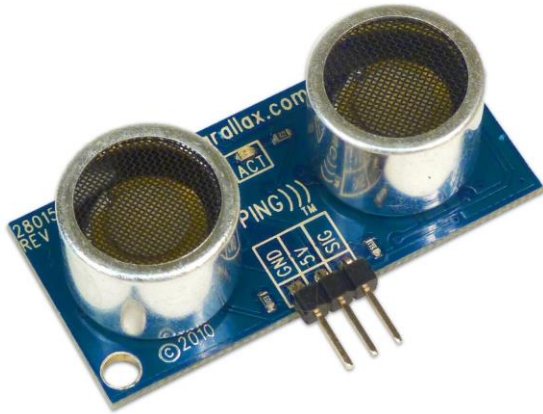
NOTES:

- White paper (90% Reflectance)
- - x - - Gray paper (18% Reflectance)

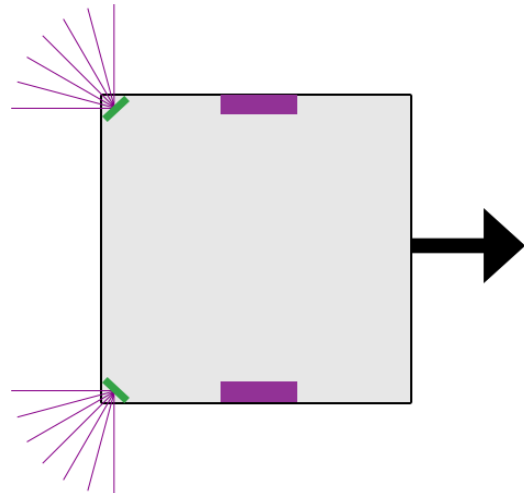
Slika 42 – Zavisnost izlaznog napona senzora SHARP 2D120X u zavisnosti od udaljenosti prepreke.

PING))) ultrazvučni senzor

Primarna uloga ovog senzora je detekcija prisustva protivničkih robota i određivanje njihove udaljenosti. Senzori su postavljeni kao što je prikazano na slici 44 sa razlogom da pokriju jedinu slepu tačku robota. Njihov primarni zadatak je da detektuju protivničkog robota i njegovu udaljenost kada se robot okreće ili kreće u nazad.



Slika 43 – Ultrazvučni senzor PING))).



Slika 44 – Položaj ultrazvučnih senzora na robotu.

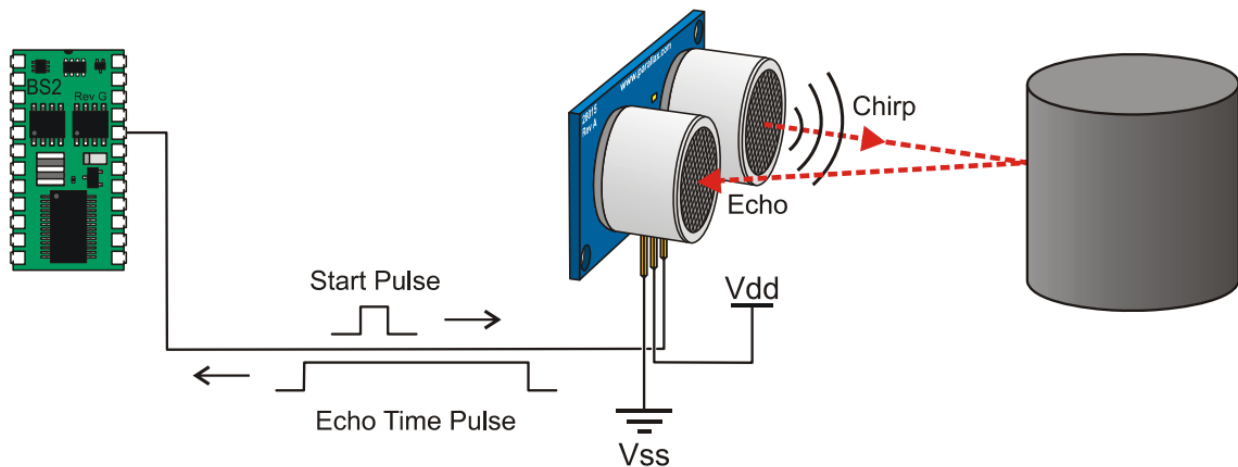
Osnovne karakteristike senzora (slika 43):

- Vreme odziva 40ms
- Digitalni izlaz
- Domet senzora: od 2 do 300cm

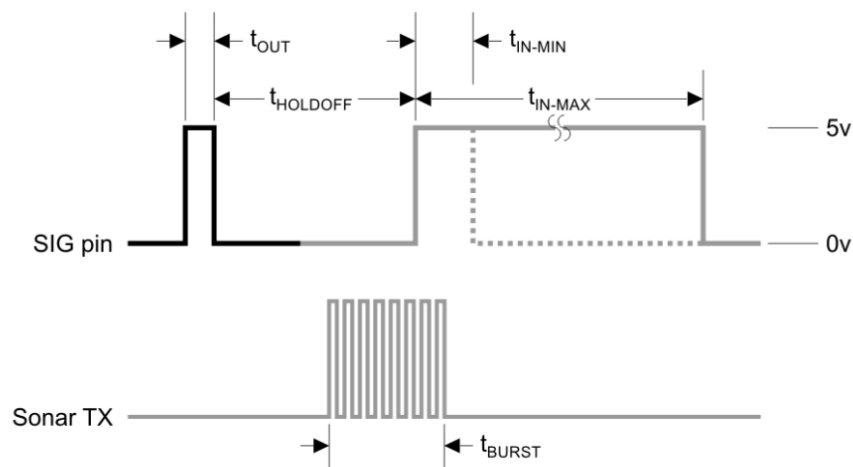
Ovi senzori su se pokazali kao pouzdani i tačni sa mogućnošću detektovanja predmeta i do 0,5cm veličine sa veoma malo šuma.

Princip rada i komunikacioni protokol:

PING))) senzor radi na principu emitovanja ultrazvučnog talasa frekvencije 40kHz u obliku niza impulsa. Nakon emitovanja talasa senzor čeka odziv. Vreme koje je potrebno da se talas emituje, odbije od prepreke i vrati nazad do senzora, direktno je proporcionalan udaljenosti prepreke od izvora talasa (senzora). Ovaj proces ilustrovan je na slici 45.



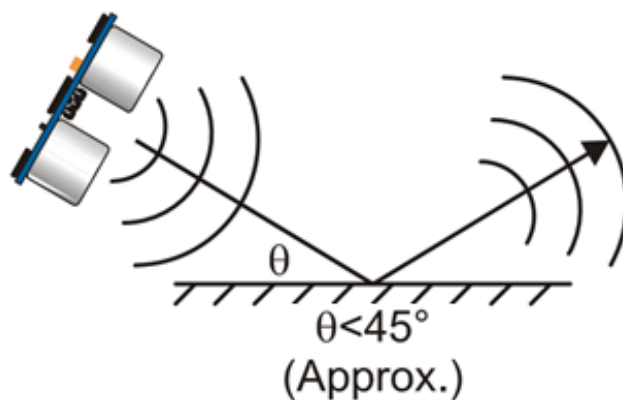
Slika 45 – Način detekcije udaljenosti prepreke pomoću ultrazvučnog senzora.



—	Host Device	Input Trigger Pulse	t_{OUT}	2 μ s (min), 5 μ s typical
—	PING))) Sensor	Echo Holdoff	$t_{HOLDOFF}$	750 μ s
—		Burst Frequency	t_{BURST}	200 μ s @ 40 kHz
—		Echo Return Pulse Minimum	t_{IN-MIN}	115 μ s
—		Echo Return Pulse Maximum	t_{IN-MAX}	18.5 ms
—		Delay before next measurement		200 μ s

Slika 46 – Vremena prilikom detekcije ultrazvučnog talasa.

Praktična ograničenja u primeni senzora:



Slika 47 – Slučaj odbijanja ultrazvučnog talasa pri uglu odbijanja manjim od 45.

Ultrazvučni senzor ne može precizno da meri rastojanje prepreka koje se nalaze pod uglom manjim od 45° u odnosu na pravac emitovanja ultrazvučnog talasa jer se tada signal ne reflektuje nazad do senzora.

Temperatura vazduha ima veliki uticaj na rezultat meranja. U rasponu od 0° do 70° greška pri merenju može da iznosi i do 12%. Pa je za preciznija meranja potrebna i upotreba temperaturnih senzora pomoću kojih bi se greška kompenzovala uz pomoć jednačine (105).

$$C_{air} = 331.5 + 0.6 \cdot T_c \left(\frac{m}{s} \right) \quad (105)$$

TCS34725 senzor za prepoznavanje boja

Kao senzor za adekvatno prepoznavanje boja lunarnih modula korišćen je TCS34725 senzor (slika 48) koji se nalazi unutar robota - iznad prostora rezervisanog za prvi lunarni modul.



Slika 48 – Senzor boje TCS34725.

Njegove karakteristike, kao što su lako upravljanje preko I2C komunikacije, brz odziv i prisustvo infracrvenog filtera je razlog zašto je odlučeno da se baš ovaj senzor koristi. TCS34725 senzor se sastoji od niza 3 x 4 fotodioda, četiri analogno digitalna konvertera (ADC) koji pretvaraju promenljivu struju fotodioda u digitalni signal koji prolaskom kroz određenu logiku, popunjava registre senzora sa informacijama o crvenoj, zelenoj i plavoj komponenti boje. Ovi podaci se iz lokalnih registara čitaju preko I2C komunikacije (do maksimalne brzine od 400kHz). Mana ovog senzora je u tome što kalibracija jako utiče na tačnost prepoznavanja boje.

ISC3806 kvadraturni enkoder za odometriju

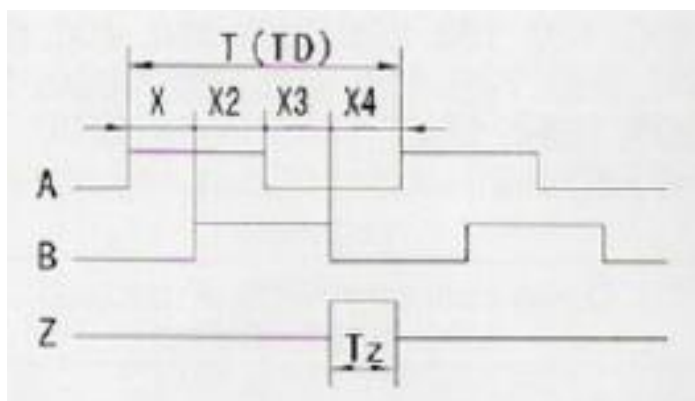
Karakteristike koje su odredile kompetentnost ovog enkodera u odnosu na druge na tržištu jesu:

- Napon napajanja od 5V do 12V
- Maksimalna struje manja od 120mA
- Broj zareza: 2048

Izlaz A i B se koriste za brojanje koraka dok se izlaz Z aktivira jednom nakon pune rotacije enkodera. Kako se može videti sa slike 50, A i B se aktiviraju u različitim trenutcima zavisno od smeru rotacije enkodera. Na taj način dobija se, ne samo informacija o broju koraka prilikom rotacije već i o smeru rotacije enkodera. Ubacivanjem ovih vrednosti u jednačine (22) i (25) dobija se promena orijentacije i pozicije robota.



Slika 49 - ISC3806.



Slika 50 – Povorka impulsa pri kontinualnoj rotaciji.

SICK WTB8-P1131 industrijski infracrveni senzor prisustva objekata

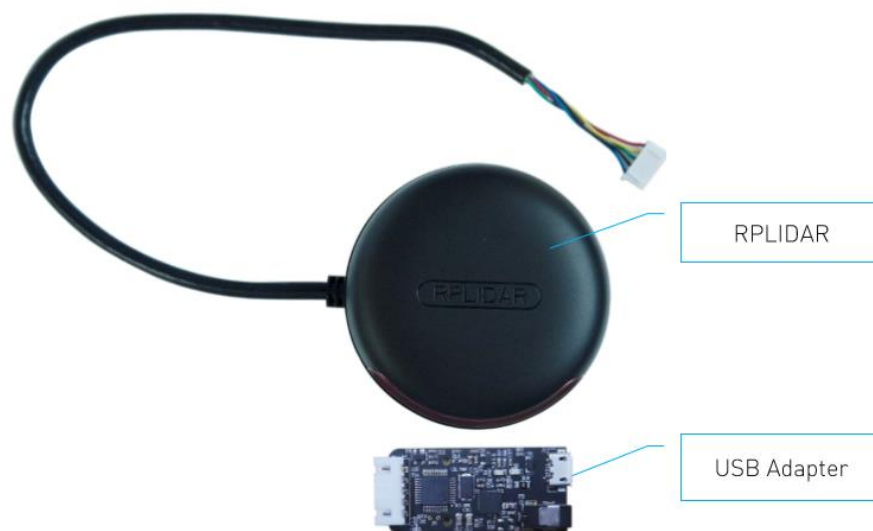


Slika 51 - SICK WTB8-P1131

Ovaj industrijski infracrveni senzor je korišćen za dobijanje informacije o prisustvu lunarnog modula u prstima robotskog manipulatora pre skladištenja. Razlog njegove upotrebe je velika pouzdanost informacije koju daje. Ovaj senzor nema mogućnost davanja informacije i udaljenosti objekta već samo da li se objekat nalazi na manjoj razdaljini od prethodno zadate.

Zbog snage motora korišćenih u zglobovima manipulatora, postavljanje ruke u zabranjeni položaj može izazvati lomljenje segmenata, prstiju i oštećenje samih motora. Kako je jedan od zabranjenih položaja uvlačenje manipulatora sa zatvorenim prstima, bez prisustva lunarnog modula, kvalitet i pouzdanost senzora za merenje ovog stanja su od ključne važnosti. To je razlog zašto je korišćen ovaj senzor.

RPLIDAR A2 laserski skener



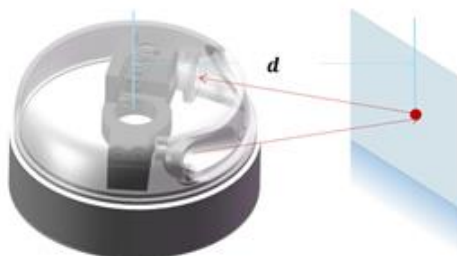
Slika 52 – RPLIDAR A2 i adapter.

Ideja korišćenja ovog uređaja u robotu je prepoznavanje lunarnih modula, protivničkih robota i uspostavljanje SLAM algoritma za snalaženje u prostoru. Prvi problem u realizaciji ovih ideja bilo je oskudno znanje SLAM algoritma, problemi prilikom prepoznavanja lunarnih modula i ivica stola. Najznačajniji problem vezan za LIDAR je manjak vremena za rešavanje zadatih ciljeva.

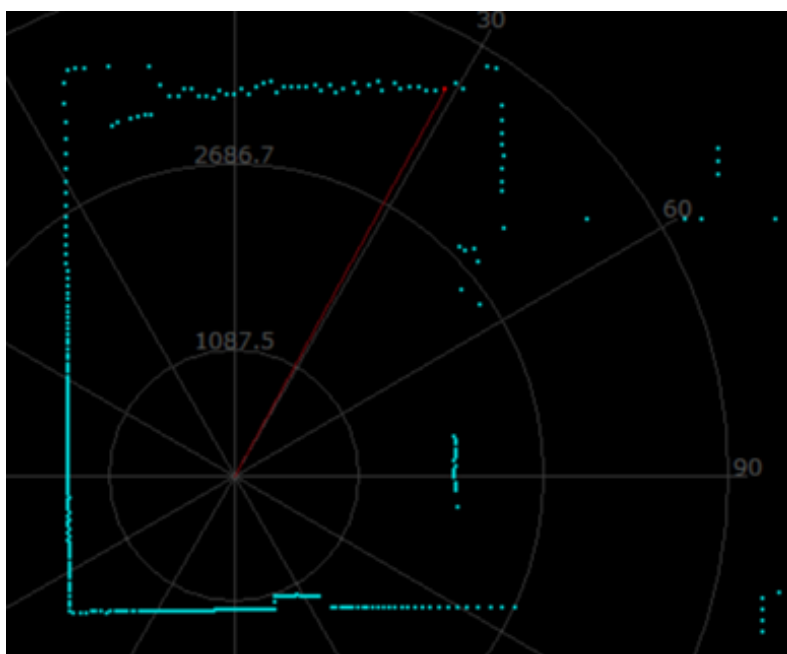
LIDAR RP A2 je 2D laserski skener koji skenira krug od 360° sa maksimalnom daljinom od 6m. Tipična frekvencija skeniranja je 10Hz (600 rpm) dok je maksimalna brzina 15Hz. Pri frekvenciji od 10Hz rezolucija skeniranja je 0.9°. Ovaj uređaj koristi laser klase 1, što znači da je bezbedan za korišćenje u prisustvu ljudi i životinja.

Određivanje daljine i ugla uzorka

LIDAR koristi lasersku triangulaciju za određivanje daljine koju je razvio proizvođač SLAMTEC. Sadrži DC motor i enkoder koji rotira pritom mereći trenutnu rotaciju lasera. Kako je za proračun triangulacije sempla potrebno odlično poznavati ugaonu brzinu rotacije, precizna kontrola ove veličine je prisutna. Na osnovu ovih parametara dobija se daljina uzorka i položaj u odnosu na koordinatni sistem lasera (slika 56). Tokom svakog procesa određivanja daljine, LIDAR emituje infracrveni laserski zrak koji se odbija od prepreku kada na nju naiđe. Vraćeni signal prolazi kroz DSP koji ga obrađuje i na svom izlazu daje vrednosti daljine i ugla između objekta i LIDAR-a preko komunikacionog interfejsa.



Slika 53 - Ilustracija procesa triangulacije laserskog zraka

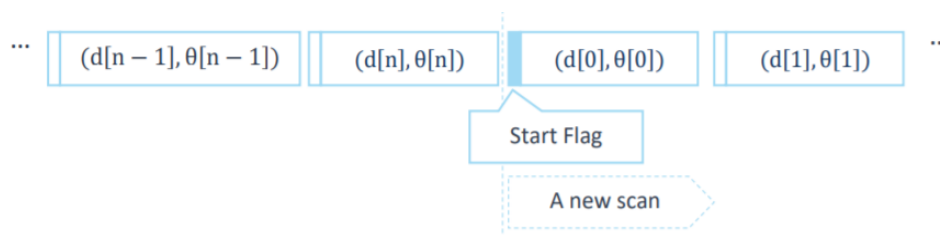


Slika 54 – Primer jednog snimka okoline. Uzorci su prikazani u x-y ravni.

Izlazne informacije LIDAR-a:

Vrednosti daljine i ugla se preko komunikacionog porta mogu preuzeti sa LIDAR-a. Format jednog uzorka je:

- Daljina (mm)
- Ugao ($^{\circ}$)
- Start flag (bool - true ili false u zavisnosti da li počinje novi uzorak)
- Checksum

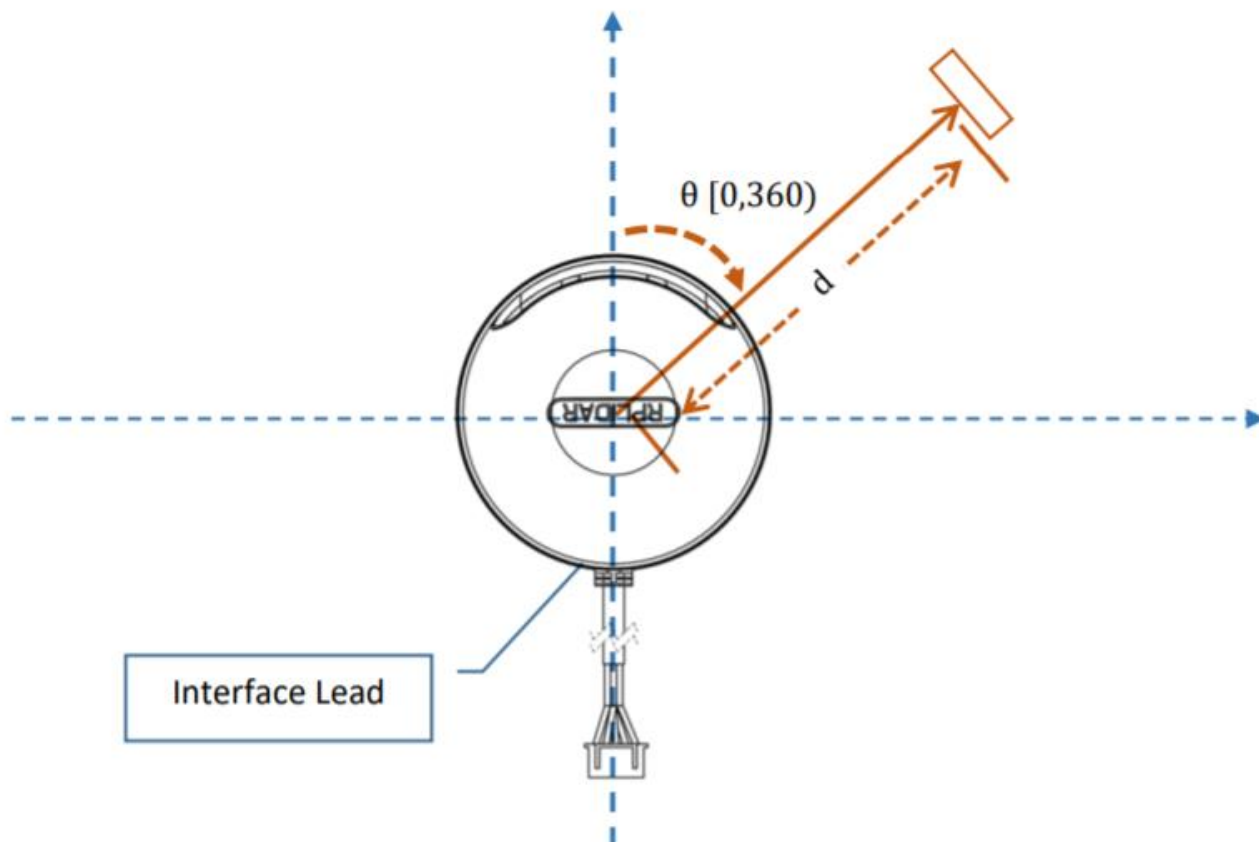


Slika 55 – Ilustracija komunikacione poruke.

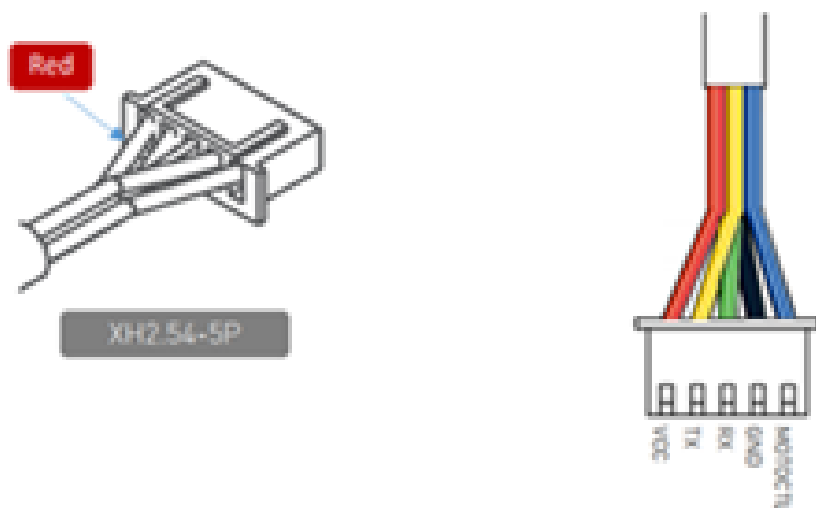
Karakteristike:

- Daljina:
 - Minimalna: 150mm
 - Maksimalna 6000mm
 - Rezolucija daljine
 - $< 0.5\%$ za uzorke do 1500mm
 - $< 1\%$ za sve uzorke
- Ugao:
 - 0° - 360°
- Vreme jednog uzorka:
 - 0.25 ms
- Frekvencija smplovanja:
 - Minimalna: 2000Hz
 - Maksimalna 4100Hz

- Frekvencija skeniranja:
 - Minimalna: 5Hz
 - Maksimalna: 15Hz
 - Tipična frekvencija skeniranja je 10Hz i pri toj frekvenciji, LIDAR uzima 400 uzoraka prilikom svakog kruga



Slika 56 – Koordinatni sistem LIDAR-a.



Slika 57 - RPLIDAR konektor za napajanje i komunikaciju.

Boja	Naziv	Tip	Opis	Minimalno	Tipično	Maksimalno
Crvena	VCC	Napon	Pozitivan napon	4.9V	5V	5.5V
Žuta	TX	Izlaz	Izlaz TTL komunikacije	0V	3.3V	3.5V
Zelena	RX	Ulaz	Ulaz TTL komunikacije	0V	3.3V	3.5V
Crna	GND	Napon	Nulti potencijal	0V	0V	0V
Plava	MOTOCTL	Ulaz	PWM signal za kontrolu brzine rotacije motora	0V	3.3V	5V

Tabela 1 – Specifikacija konektora za LIDAR.

Naziv	Veličina	Minimalno	Tipično	Maksimalno
Baud	bps	-	115200	-
Radni mod	-	-	8N1	-
Izlazni visok nivo	V	2.9	-	3.5
Izlazni nizak nivo	V	-	-	0.4
Ulazni visok nivo	V	1.6	-	3.5
Ulazni nizak nivo	V	-0.3	-	0.4

Tabela 2 – Specifikacija komunikacije za LIDAR.

Više informacija i specifikaciji ovog senzora moguće je pronaći na sajtu proizvođača.

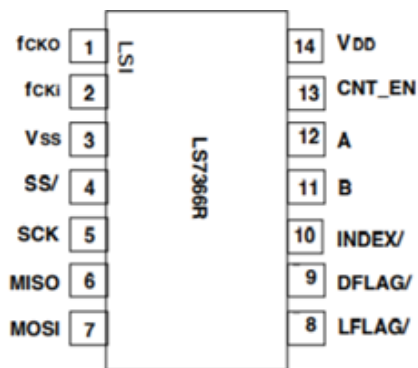
LS7366R kvadraturni dekodler



Slika 58 – LS7366R.

Kako kvadraturni enkoder na svojim izlaznim pinovima šalje signale frekvencije zavisne od brzine rotacije, a čija faza zavisi od smera rotacije, za tačnu interpretaciju te informacije potrebno je dovoljno brzo odreagovati kako ne bi došlo do preskakanja koraka. LS7366R je 32-bit kvadraturni brojač brzine do 40MHz koji se upravlja korišćenjem ugrađene SPI komunikacije, što daje mogućnost upravljanja preko velikog broja uređaja. Prilagođenost hardverskog dekodera optičkim enkoderima i enkoderima koji rade na principu Holovog efekta takođe ide u prilog ideji o modularnom pristupu projektovanja.

Princip rada 32-bit kvadraturnog brojača



Slika 59 – Raspored pinova.

Zavisno od napona napajanja, maksimalna frekvencija brojanja je 40MHz za napon napajanja od 5V ili 20MHz za napon napajanja od 3.3V. Korišćenje ovog integrisanog kola se svodi na dovođenje signala od 40MHz između pinova **fcko** i **fcki**, zatim dovođenje signala sa A i B izlaza kvadraturnog enkodera na **A** i **B** ulaze ovog integrisanog kola. Dovođenjem napona od 5V između **Vdd** i **Vss** pinova, informacija o stanju enkodera se čita preko **SPI** komunikacije (**SS SCK MISO MOSI** pinovi).

Aktuatori

FR5311M servo motor



Slika 60 – Reduktor servo motora.



Slika 61 – FR5311M.

Ovaj servo motor korišćen je kao aktuator za ostvarivanje dva rotaciona kretanja na robotskom manipulatoru. Aktuatori su se pokazali kao odličan izbor shodno ceni i kvalitetu, sa odličnim momentom i preciznošću. Osnovna karakteristika ovih aktuatora je da su u pitanju kontinualni digitalni servo motori(360°) sa mogućnošću kontrole pozicije, upravljani PWM signalom. Njegova brzina i snaga su direktno proporcionalni naponu koji može da se kreće u rasponu od 4,8V do 8,4V.

9g servo motori



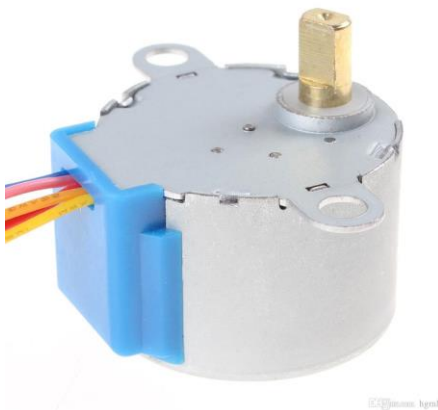
Slika 62

Ovi servo motori su praktično najjednostavniji i najrasprostranjeniji tip standardnih motora upravljani PWM signalom sa mogućnošću rotacije od oko 180°. Primenjivali su se za sve ostale segmente u kojima se javila potreba za aktuatorom male snage i dimenzija. Njihova osnovna karakteristika je što su mali, jeftini, lako se mogu implementovati i upravljati sa relativno dobrom preciznošću.

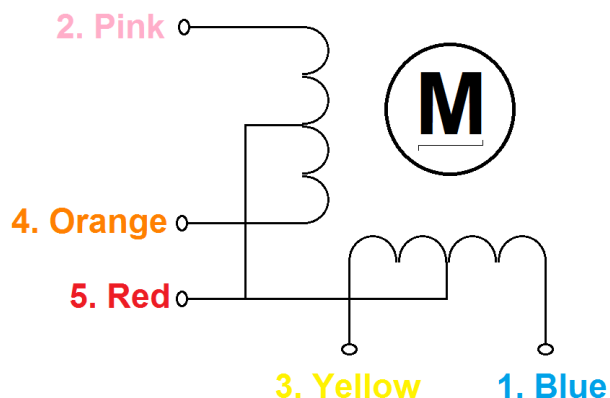
Koračni motor 28BYJ-48

28BYJ-48 je mali jeftini unipolarni koračni motor sa naponom napajanja u rasponu od 5V do 12V, sa ugrađenim reduktorom prenosnog odnosa 64:1 i brzinom od 15 do 25 obr./min. Zbog prenosnog odnosa ovaj motor može da ostvari precizno pozicioniranje(5.625°/koraku) sa zadovoljavajućim momentom shodno njegovom gabaritu. Ograničenja se javljaju u pogledu kvaliteta izrade gde se često dešavalo da dođe do proklizavanja samih zubaca zubčanika zbog njihove izrade od

meke plastike. Motor se sastoji od četiri namotaja čija šema može da se vidi na slici. Koračni motor je upravlján uz pomoć *EasyDrive* upravljačkog kola o kome će u daljem tekstu biti reči.



Slika 63 – Koračni motor 28BYJ-48.



Slika 64 – Raspored napotaja koračnog motora.

Maxon DC motor



Slika 65 – Maxon DC motor sa četkicama.

RC30 Maxon DC motori sa četkicama su korišćeni kao pogonski motori na robotu. Pokazali su se kao nepraktični i komplikovani za upravljanje, što je mana većine DC motora sa četkicama. Karakteristike između dva motora, koja su korišćena na robotu, bile su znatno drugačije jedna od druge pa je njihovo usklađivanje bilo komplikovano. Ovaj problem je jednim delom prevaziđen korišćenjem različitih programskih regulatora za svaki motor što je ograničilo maksimalno ubrzanje robota. Kako su korišćeni motori bili polovni, problem nejednake regulacije bi mogao da se reši korišćenjem novije generacije DC motora sa četkicama koji imaju bliske karakteristike.

Za upravljanje motorima korišćena su dva VNH2SP40 H-mosta (slika 68), dva enkodera na osocini motora i Arduino Due kao mikrokontroler za regulaciju upravljačkog PWM signala. Enkoderi povezani na osovinu motora služili su kao povratna sprega u regulator. Na ovaj način relativno uspešno su usklađene brzine i ubrzanja motora sa velikim ograničenjima na maksimalno ubrzanje.

Upravljačka kola

EasyDrive modul za upravljanje koračnim motorom



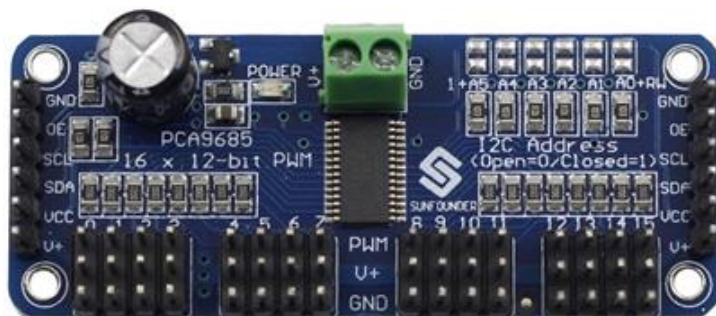
Slika 66

EasyDrive je mali i jednostavan modul za upravljanje koračnim motorima. Kompatibilan je sa svim upravljačkim uređajima koji na svom izlazu mogu da daju puls od 0V do 5V (ili od 0V do 3.3V ako se prespoji džemper na pcb pločici).

EasyDrive može da upravlja koračnim motorima sa naponom napajanja u rasponu od 6V do 30V. Posедуje sopstveni regulator napona za digitalni interfejs koji može biti podešen na 5V ili 3.3V. Povezivanje koračnog motora se vrši povezivanjem 4 žice koračnog motora na izlazne terminale modula. EasyDrive može da upravlja samo sa bipolarnim koračnim motorima ili bilo kojim motorima koji su bipolarne konstrukcije (motori sa 2, 4, 6 ili 8 žica). Osnovna komponenta ovog modula je čip A3967 koji vrši kontrolu koračnog motora. Treba napomenuti da ne treba prikačiti ili otkaćiti motor pri prisutnom naponu na modulu jer će to dovesti do trajnog uništenja samog čipa A3967. Ovaj modul poseduje mogućnost podešavanja koraka koračnog motora, gde se podešavanjem pozicije dva džampera na modulu (MS1 i MS2) može izabrati jedan od 4 režima rada: korak, pola koraka, četvrtina koraka i osmina koraka. (podrazumevana pozicija je osmina koraka).

Na modulu postoji i potencijometar sa kojim se može podešavati količina struje kroz namotaje motora i to u rasponu do 150mA/fazi do 700mA/fazi. Upotreba ovog modula za upravljanje koračnim motorima se pokazala veoma praktična, pouzdana, jednostavna i dobra zbog velikog izbora u pogledu načina upravljanja samim motorom.

PCA9685 PWM modul

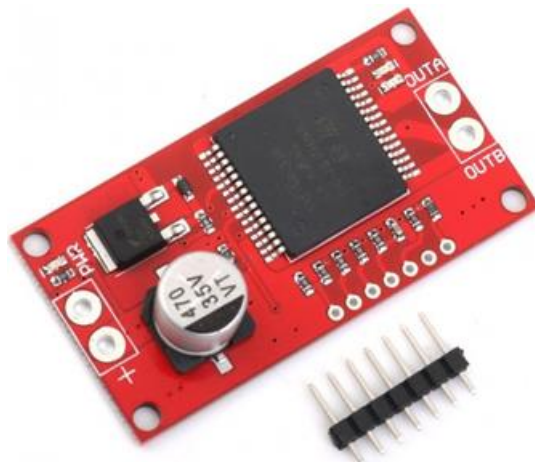


Slika 67

PCA9685 predstavlja veoma koristan čip pomoću kojeg se može proširi broj PWM izlaza sa kojima mikrokontroler može da upravlja. Upravljanje ovim modulom se vrši upotrebom samo dva I/O pina na mikrokontroleru čime se broj PWM izlaza proširuje na još 16 individualnih PWM signala. Pored ovog proširenja, postoji mogućnost da se na ovaj modul nadoveže još 62 modula što bi praktično značilo da se broj PWM izlaza može proširiti na čak 992 PWM signala. Komunikacija sa ovim modulom se vrši putem I2C. Na samoj pločici postoji 6 džampera pomoću kojih se može podesiti željena adresa modula. Kompatibilan je sa 5V i 3.3V upravljačkim signalom i poseduje mogućnost priključivanja posebnog izvora napajanja čiji se

napon može sprovesti do uređaja koji se upravlja PWM signalom. Posедуje mogućnost promene frekvencije PWM signala sve do 1.6KHz sa rezolucija od 12-bit. OutputEnable(OE) pin omogućava brzo isključivanje svih PWM izlaza modula u koliko se na njegov ulaz dovede logička jedinica. Ovaj modul je našao idealnu primenu za upravljanje velikog broja servo motora za manipulaciju lunarnim modulima u skladištu robota (slika 18).

VNH2SP40 H-most modul



Slika 68

Primenom ovog modula, upravljačko kolo se izoluje od samog visokonaponskog voda što sprečava pojavu mogućnosti uništenja čitavog kontrolera, na kojem je čip montiran, u koliko dođe do havarije. Smanjuje se mogućnost pogrešnog povezivanja i mogućnost dovođenja visokog napona na neke od ne predviđenih ulaza samog kontrolera.

Izdvajanjem VNH2SP40 modula se postiže smanjenje dužine visokonaponskih kablova jer se modul može montirati blizu samog uređaja. Smanjenjem dužine visokonaponskih vodova se postiže i manja interferencija signala i pojava smetnji koje mogu da ometaju rad upravljačkih kola i senzora.

Upravljanje ovim modulom se vrši uz pomoć PWM signala pomoću kojeg se određuje brzina rada motora, Enable(EN) pina za aktivaciju ili deaktivaciju rada modula, INA i INB ulaza sa kojima se definiše režim rada samog H-mosta(promena smera obrtanja motora, prespajanje namotaja na motoru, itd).

Izvori napajanja

Kako se sistem sastojao od niza elektronskih uređaja, senzora i mikrokontrolera čiji su naponi napajanja 5V, 3.3V ili 10V, pri čemu je jedini izvor napajanja u robotu jedna Li-pol (slika 69) baterija ukupnog kapaciteta 5200mAh, nastala je potreba da se napon sa Li-pol baterije koji varira u opsegu od 14.4V do 16.8V, spusti i stabilizuje na željene napone. U tu svrhu izabrani su takozvani prekidački naponski regulator koji sa velikom efikasnošću spuštaju i održavaju napon konstantnim na željenoj vrednosti pri čemu se gubici energije svode na minimum.



Slika 69 – Li-pol baterija

Za napajanje uređaja sa stabilnih 5V izabrano je 4 bak konvertora koji su podeljeni tako da ravnomerno snabdevaju delove robota sa naponom. Razlog izbora više bak konvertora umesto jednog je prvenstveno cena, dimenzije i modularnost. Kako se većina uređaja napaja sa 5V, bilo je potrebno pronaći bak konvertor koji može da pruži zadovoljavajuće preformanse, reda i do 10 ampera na svom izlazu, a da pri tome napon ostaje stabilan. Takav bak konvertor je bilo teško naći jer su njihove dimenzije bile neprihvatljive i cene često previsoke za naša raspoloživa sredstva. Zbog toga je odlučeno da se iskoristi više bak konverota, malih dimenzija (slika 70), skromne cene i podeli njihova oblast snabdevanja u sektore čime se ujedno obezbeđuje modularnost i veća pouzdanost sistema.

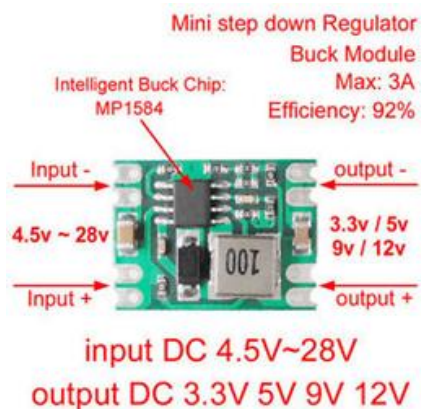
Naime, izvori napajanja sa 4 bak konvertora su bili podeljeni na sledeći način:

- Napajanje levog robotskog manipulatora (3 Servo motora, IR senzor...)
- Napajanje desnog robotskog manipulatora (3 Servo motora, IR senzor...)
- Napajanje upravljačkog kola (2 mikrokontrolera Arduino Due, Ultrazvučni senzori, IR senzori...)
- Napajanje skladišta (10 malih servo motora, senzori boja...)

Ovakvom raspodelom omogućeno je stabilnije i sigurnije napajanje komponenti. Zbog malih dimenzija konverota pojavila se mogućnost za znatno fleksibilniji izbor prostora pri njihovom postavljanju. U slučaju dolaska do nekog kvara, izbegnuta je mogućnost apsolutne havarije sistema. U slučaju korišćenja samo jednog bak konverota za snabdevanje komponenti konstantnim naponom i strujom, u slučaju kvara nastaje mnogo veća šteta i nemogućnost popravke bez nove komponente.



Slika 70 – Bak konverter za napon od 5VDC.



Slika 71 – Bak konverter za napajanje od 3.3VDC.

Osnovna komponenta ovog uređaja je integrisano kolo TPS40057 kompanije Texas Instruments. Opseg ulaznog napona napajanja je od 8V do 40V a izlazni napon se određuje u zavisnosti od vrednosti napona na set pinu koji je u ovom slučaju predefinisani i postavljen pomoću razdelnika napona tako da na izlazu održava napon od 5V. Izlazna struja ovog modula je maksimalno 5A za kratke impulse i 3A za kontinualno napajanje.



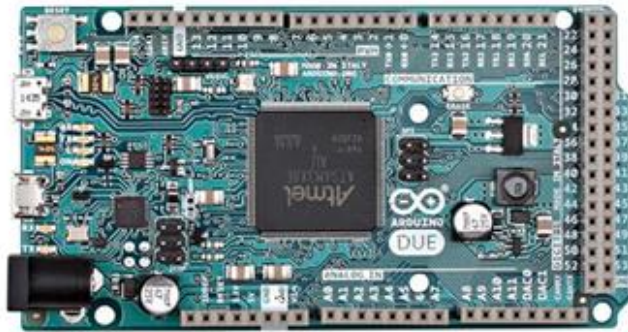
Slika 72

Kako je broj komponenti koje zahtevaju napon napajanja od 3.3V mali, i uglavnom njihova potrošnja takođe mala. U tu svrhu je izabran bak konvertor veoma malih dimezija (slika 71) sa zadovoljavajućim performansama.

Treća vrsta bak konvertora je korišćena za snabdevanje koračnih motora sa naponom od 10V (slika N). Osnovno integrisano kolo ovog bak konvertora je LM2596 kompanije Texas Instruments. Ovakva konfiguracija modula omogućava podešavanje željenog napona na izlazu uz pomoć potencijometra.

Mikrokontroleri i procesori

Arduino Due



Slika 73

Arduino **Due** je Arduino razvojna ploča koja u svom središtu sadrži Atmel ARM Cortex-M3 32-bitni CPU koji radi na frekvenciji od 84MHz, poseduje internu memoriju od 512KB i radi na naponu od 3.3V. Sadrži 54 digitalna ulazno/izlazna porta od kojih se 12 može koristiti kao PWM izlaz, 12 analognih ulaza, 4 hardverska serijska porta(UART), dva I2C-a, CAN i SPI. Kako je Arduino Due “Open source” postoji mnogobrojna literatura, velika količina foruma, integrisanih biblioteka i program posebno prilagođen za jednostavno pisanje koda. Sve ovo je navelo da implementujemo dva ovakva mikrokontrolera što je znatno olakšalo i ubrzalo njihovu primenu.

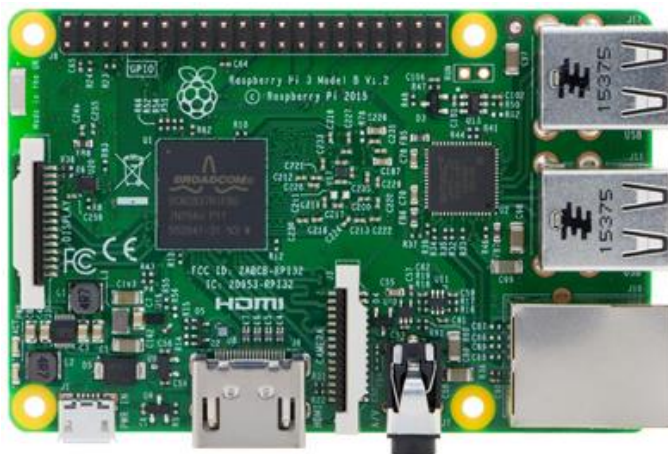
Mikrokontroler	Atmel SAM3X8E ARM Cortex-M3
Radni napon	3.3V
Ulazni napon u integrisani regulator	7V-12V
Limiti ulaznog napona	6V-20V
Maksimalna izlazna struja na svim ulazno/izlaznim pinovima	130mA
Maksimalna struja na 3.3V pin-u	800mA
Maksimalna struja na 5V pin-u	800mA
Flash memorija	512KB
SRAM	96KB (dva čipa: 64KB i 32KB)
Brzina radnog takta	84MHz
Dimenzije	53.3x101.52mm

Tabela 3 – Specifikacija Arduino Due mikrokontrolera.

Razlog korišćenja dva mikrokontrolera umesto jednog je u obimu procesa koje je potrebno obraditi u kratkom vremenskom intervalu. Stime su procesi podeljeni na sledeći način; Jedan mikrokontroler (slika 75) je zadužen za upravljenje motorima, prikupljanje vrednosti sa odometrijskih enkodera, enkodera na motorima, vrednosti sa infracrvenih i ultrazvučnih senzora; Drugi mikrokontroler (slika 76) je zadužen za upravljanje robotskim manipulatorima, motorima za skladištenje lunarnih modula i prikupljanje vrednosti sa senzora tog dela sistema. Kako bi mikrokontroleri bili sinhronizovani i mogli da

razmenjuju prikupljene podatke, korišćena je UART hardverska komunikacija na brzini od 115200bps. Ovakvo formiranje sistema omogućilo je brzu obradu velikog broja signala gde su se upravljačke odluke mogu donositi brzinom od 100Hz.

Raspberry Pi 3 model B



Slika 74

Raspberry Pi model 3 je treća generacija Raspberry Pi platforme sa Broadcom 64-bit procesorom, 1GB RAM memorije i taktom od 1,2GHz. Potreba za mikroprocesorom je nastala zbog želje da se postigne visok nivo autonomnosti robota i snalaženja u prostoru. Ovo je značilo da se velika količina podataka mora obraditi za jako kratko vreme, gde se na osnovu podataka donosi veliki broj upravljačkih odluka. Iako se u sistemu nalaze da brza mikrokontrolera, oni nisu predviđeni za rešavanje kompleksnih matematičkih problema i rad sa velikim bazama podataka. Zbog toga je u sistem uvedena Raspberry Pi platforma. Ona je korišćena umesto platforma kao što su Intel NUC, UDO86x itd. zbog odnosa cena/kvalitet i dostupne dokumentacije.

Raspberry Pi je u konfiguraciji sistema izdvojen od PCB pločica i povezan je sa mikrokontrolerima preko UART komunikacije. Na njemu se nalazio API za donošenje upravljačkih odluka i njihovo prosleđivanje na mikrokontrolere za dalje izvršavanje.

Uvođenjem Raspberry-a u sistem omogućeno je veoma lako otklanjanje grešaka, podešavanje parametara regulatora, podešavanje putanja kretanja robota, praćenje rada robota i to sve uz pomoć mobilnog telefona putem WiFi ili Bluetooth komunikacije. Time je vreme treniranja robota znatno skraćeno a radni vek mikrokontrolera znatno produžen jer nema potrebe za čestim otpremanje novog koda u same mikrokontrolere.

PCB pločice

Vođeni modularnim dizajnom, upravljačka kola sa svojim pratećim elementima podeljeni su na dve PCB pločice:

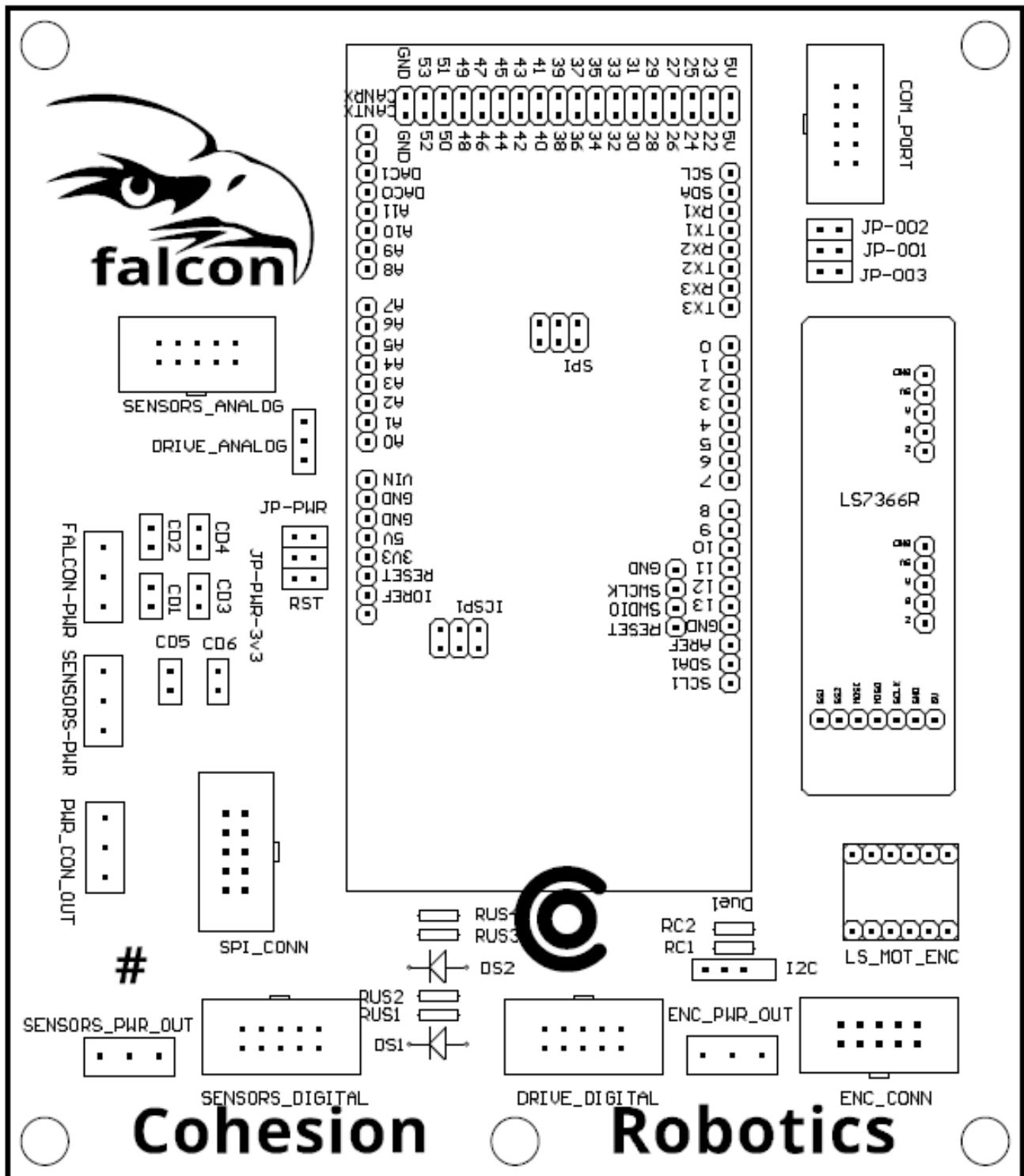
1. Falcon (slika 75) - sve komponente vezane za kretanje i upravljanje robotom
2. Eagle (slika 76) - sve komponente vezane za manipulaciju lunarnih modula i njihovo skladištenje

Falcon

Ovaj deo elektronike robota sadrži sve što je potrebno da se robot snalazi u prostoru, upravlja aktuatorima za kretanje, prikuplja podatke sa senzora, enkodera i komunicira sa *Raspberry Pi*-om i *Arduino Due*-om koji se nalazi na drugoj PCB pločici.

Falcon PCB pločica sadrži:

1. *Arduino Due*
2. Konektore za očitavanje enkodera koji se nalaze na aktuatorima.
3. Kvadraturni dekodera za očitavanje enkodera odometrije.
4. Konektore za digitalne senzore (Ultrazvučne)
5. Konektor za analogne senzore (IR)
6. Konektor za VNH2SP40 H-mostove koji direktno upravljaju aktuatorima za kretanje
7. Konektor za SPI komunikaciju
8. "Logic level shifter" za spuštanje enkoderovih 5V logičkog nivoa na 3.3V
9. Konektor za UART komunikaciju sa drugim *Arduino*-m i *Raspberry Pi*-om
10. Konektore za napajanje pločice sa potrebnim naponima



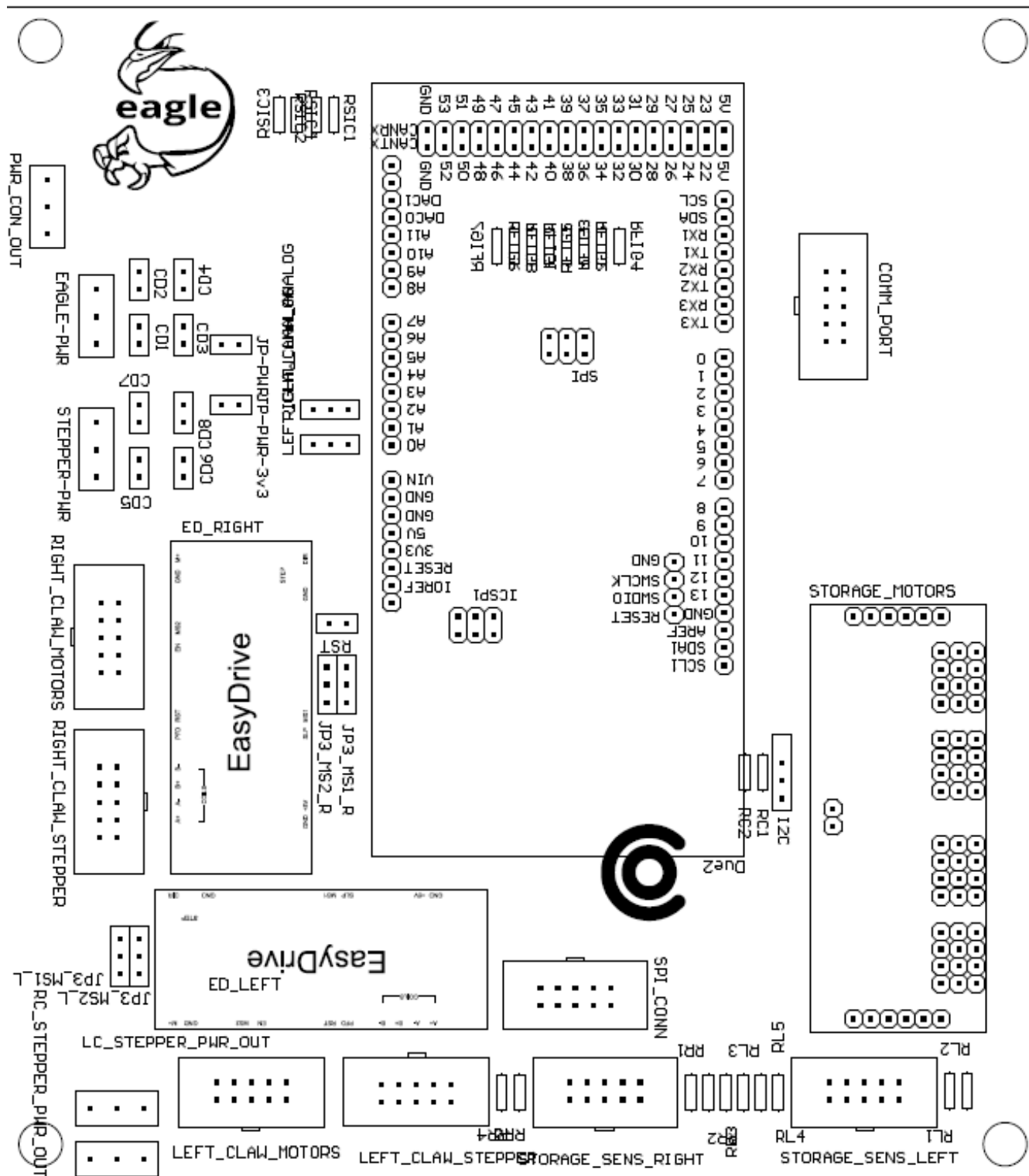
Slika 75 – Falcon PCB pločica

Eagle

Ovaj deo elektronike robota sadrži sve što je potrebno za osvajanje poena. Dakle, upravljanje skladištem i robotskim manipulatorima koji sadrže servo, koračne, DC motore, senzore boje i IR senzore.

Eagle PCB pločica sadrži:

1. Arduino Due
2. *PCA9685 PWM* modul za proširenje PWM konektora preko kojeg se upravlja servo motorima
3. Dva *EasyDrive* modula za upravljanje koračnim motorima
4. Konektor za *VNH2SP40* H-mostove koji direktno upravljaju sa DC motorima za translatorno kretanje robotske ruke
5. Konektor za UART komunikaciju sa Arduino-m na drugoj pločici
6. Konektore za senzore prisustva lunarnih modula u skladištu
7. Konektor za SPI komunikaciju
8. Konektor za analogne senzore (IR senzori i SICK senzori)
9. Konektore za napajanje pločice sa potrebnim naponima



Slika 76 – Eagle PCB pločica.

Upravljanje

Osnovni zadaci

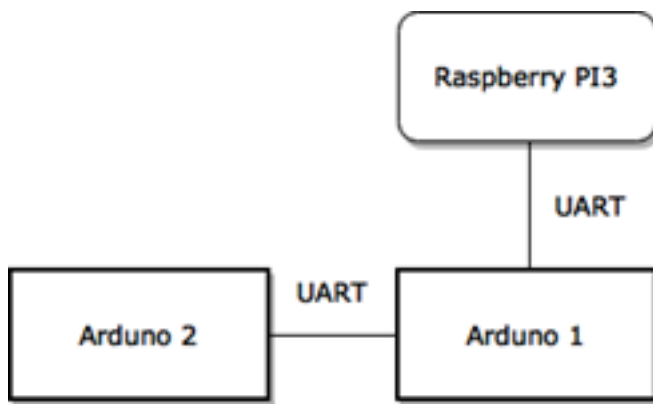
Osnovni zadatak je upravljanje i navođenje robota u cilju izvršavanja zadataka opisanog u poglavlju Opis zadatka Eurobot 2017. Ovo predstavlja razne izazove koje je potrebno rešiti. Neki od njih koje je robot morao da obavlja su:

- uspešno kretanje po igralištu,
- praćenje pozicije,
- skupljanje lunarnih modula i njihovo ostavljanje na adekvana mesta igrališa (detaljnije u poglavlju Uvod u problem),
- izbegavanje sudara,
- i drugi.

Kao što je rečeno, osnovni koncept upravljanja robota je baziran na korišćenju dva Arduino Due mikrokontrolera i Raspberry Pi 3 računara. Do sada nevedeni zadaci su se mogli rešiti korišćenjem i samo jednog mikrokontrolera, međutim ovakvo rešenje bi podrazumevalo pisanje softvera sa samo jednom svrhom koje je potpuno ne fleksibilno. Zbog potrebe za flaksibilnosti softvera i procesa koji se moraju paralelno izvršavati, odlučeno je da se koristi Raspberry Pi 3 računar.

Važno je ponoviti, razlog korišćenja Arduino Due mikrokontrolera i Raspberry Pi 3 računara jeste široka primena ovih platformi što podrazumeva veliki broj korisnika, već rešenih problema, biblioteka za opravljanje aktuatorima, senzorima, mogućnost pisanja koda u programskom jeziku C++ za koji takođe postoji veliki broj primera. Za pistanje softvera porišćena su programska okruženja Atom, Platfomio i Qt Creator.

Zbog korišćenja većeg broja upravljačkih komponenti, moduli su povezani korišćenjem UART komunikacije. Kako na Raspberry Pi računaru postoji samo jedan UART modul za komunikaciju, nije bilo moguće komunicirati sa oba mikrokontrolera u isto vreme, zbog toga su kontroleri i računar povezani u Daisy-chain konfiguraciju (slika 77) gde Raspberry Pi komunicira sa Falcon pločicom (slika 75), a Falcon sa Eagle pločicom (slike 76).



Slika 77 – Daisy-chain konfiguracija.

Na osnovu liste prioriteta i trenutnih vrednosti sa senzora, Raspberry Pi donosi zaključak koja akcija treba da se preduzme i šalje ih mikrokontrolerima.

Komunikacija

UART komunikacija sa brzinom od 115200bps je korišćena. Za ovaj projekat je razvijem standard za razmenjivanje poruka, međutim kako su poruke mogle biti različitih dužina, morala su biti ispoštovana određena pravila prilikom krazmene poruka kako bi se proverila validnost. Format jedne poruke prikazan je u tabeli 4.

Upravljačka poruka: okreni se za ugao N radijana		
Redni broj bajta	vrednost	Opis
1	0x5E	Početak poruke
2	0x09	Dužina poruke
3	0x03	Naziv funkcije
4	Promenljiv	Ugao (LSB), podatak tipa FLOAT
5	Promenljiv	Ugao
6	Promenljiv	Ugao
7	Promenljiv	Ugao (MSB)
8	Zavisi od prethodnih bajta	Suma bajtova [2...7] * (-1)
9	0x0D	Kraj poruke

Tabela 4 – Primer jedne upravljačke poruke za promenu orijentacije robota.

- Bajt 1 predstavlja početak svake poruke,
- Bajt 2 predstavlja dužinu poruke, tačije ukupan broj bajtova u poruci,
- Bajt 3 predstavlja jednoznačnu identifikaciju funkcije.

Nakon prva tri bajta sledi različiti broj bajtova u zavisnosti od identifikacijone funkcije, dok ti bajtovi predstavljaju vrednosti promenljive koja se šalje. Prvo se šalje bajt najmanje težine (LSB) dok je poslednji u nizu najveće težine (MSB). Pretposlednji bajt u poruci sadrži vrednost sume kako bi se proverilo da li je poruka stigla onakva kakva je poslata. Poslednji bajt u nizu predstavlja kraj poruke.

Pri slanju poruka sa kontrolera ka računaru nije postojao red čekanja jer su se poruke sa svim vrednostima senzora slale na svakih 10ms i ukoliko bi se i desilo da neka poruka nije dobro poslata već u narednih 10ms bi se ta greška ispravila.

Softver za Raspberry Pi

Prilikom razvoja softvera, težilo se kao razvijanju više modula, koji u celini, međusobnom kolaboracijom, upravljaju čitavim sistemom robota, a razlozi su lako održavanje, čitanje, prapравljanje i dodavanja novog koda. Moduli su:

- grafički deo,
- komunikacija,
- menadžer zadataka,
- menadžer logike.

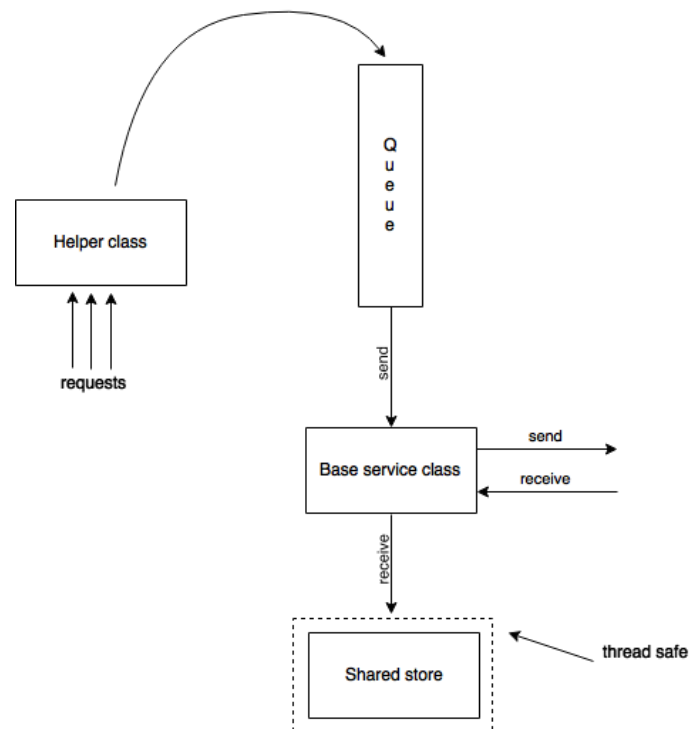
Grafički deo

Kako se Raspberry Pi nalazio zatvoren unutar sistema robota nije bilo praktično povezati ekran i ostalu periferiju direktno na njega kako bi se pristupilo operativnom sistemu i aplikacijama, već je za to korišćen VNC Viewer [x]. Za ovu namenu je razvijen program sa grafičkim okruženjem kao alat za brži razvoj softvera i rešavanje mogućih problema u sistemu. Ovaj program je takođe sadržao i konzolu u kojoj su se ispisivali svi parametri sistema.

Komunikacija na Raspberry Pi računaru

Modul za komunikaciju je radio u dve niti, koje su nezavisne od ostatka programa i to tako da je jedna nit zadužena za slanje podataka kontroleru, a druga za primanje podataka od kontrolera. Razlog koncepta pravljenja dve niti samo za komunikaciju je da ne bi došlo do preopterećivanja komunikacione linije. Uz to, pri svakom slanju poruke, čeka se odgovor kontrolera da je poruka prisitlgla i da se može slati sledeća. Time je rešen jedan moguć problem, ali je postojao još jedan, a to je “šta raditi ako poruka nikad ne stigne”. Logično objašnjenje je ništa, jer onda nešto ide po zlu, ali smo ipak opredelili da ograničimo vreme čekanja na odgovor da je poruka uspešno poslata. Koraci cljanja poruke su se sastojali od:

- zahteva za slanje poruke,
- kreiranje poruke,
- stavljanja u red za čekanje,
- preuzimanje od strane niti za slanje,
- slanje,
- čekanja druge niti za primenje poruke o uspešnosti slanja,
- čekanje pojave novog sahteava.



Slika 78 – Blok diagram komunikacionog protokola.

Menadžer zadatka

Početni koncept ovog modula je bio da inicijalno napravi sve moguće zadatke koji se mogu desiti na stolu, a da modul za logiku određuje njihov prioritet u listi na osnovu relativne pozicije do njih, stanja na stolu... Međutim kako su rokovi bili

kratki, kao i nebrojani problemi nastali tokom razvoja robota, konačna odluka je bila da se zavisno od inicijalne strane na stolu (ljubičasta ili žuta), naprave unapred zadaci u redosledu kojim smo hteli da se izvršavaju.

Menadžer logike

Modul koji je samo “srce” bilo koje odluke donešene tamo na stolu je modul logike, njegov zadatak je inicijalno bio, kako smo gore naveli da sam određuje prioritet zadataka, međutim to se nije desilo. Krajnji zadatak ovog modula je da izvrši početna podešavanja u zavisnosti od konfiguracije na terenu i da očekuje start programa koji se zadavao povlačenjem pina na robotu. Tok daljeg programa se bazirao na dobijanju zadataka od menadžera zadataka, njihovog raspakivanja i zadavanje zadataka za slanje podataka modulu za slanje. Koncept zadataka je podeljen na zadatke i manje zadatke, gde jedan zadatak sadrži više manjih. Primer jednog zadatka, skupljanje jednog lunarnog modula:

- odlazak iz tačke A u tačku B,
- odlazak iz tačke B u tačku C,
- otvaranje leve ruke,
- odlazak iz tačke C u tačku D,
- zatvaranje leve ruke,
- odlazak iz tačke D u tačku G,
- odlazak iz tačke G u tačku D,
- ostavljanje lunarnog modula.

Postoje manji zadaci koje ne traže da se prethodni mini zadatak završi, ali postoje i one koje traže, tako da nit koja je zadužena za slanje poruka, čeka odgovor da je robot uradio prethodan mini zadatak. U devedesetoj sekundi meča se ispaljivao funny action i robot je isključivao svoje aktuatorne.

Mašinska vizija

Pošto smo zadali sebi zadatak da napravimo “pametnog” robota, koji neće imati prazan hod, tj. neće pokušavati da izvrši nemoguće zadatke, znali smo da moramo povećati broj i vrstu senzora koji će se nalaziti na robotu. Time bi robot postao svesniji svoje okoline i mogao bi bolje da rešava zadatke i donosi poene. Između ostalog odlučili smo se i za primenu mašinske vizije koja je trebalo da omogući robotu da “vidi”, tj. kada bi robot došao do kratera pomoću kamere bi snimio krater i prepoznao da li u njemu ima kuglica i da li uopšte treba da se zadržava tu i kupi kuglice ili da nastavi dalje, ukoliko je prepoznato da je krater prazan.

Isti slučaj je trebao da se desi i sa lunarnim modulima. Robot bi došao do skladišta gde se nalaze lunarni moduli i ukoliko kamera očitava da u skladištu postoje lunarni moduli onda bi ih robot skupljao, u suprotnom bi samo nastavio sa nekim sledećim zadatkom.

Takodje zamišljeno je da po potrebi robot može da proveriti da li je i mesto za ostavljanje lunarnog modula slobodno i da spreči akciju ostavljanja modula u već popunjeno skladište.

Posle istraživanja na internetu odlučili smo se za primenu biblioteke OpenCV (open computer vision) koja se veoma mnogo primenjuje i koja je kako joj samo ime kaže otvorenog tipa i stalno se dopunjuje.

Pošto je korišćena samo obična kamera kao ulaz za dalju obradu je uvek dobijana neka 2D slika 3D prostora i najveći problem je što nismo imali “dubinu” tj normalnu udaljenost tačaka prostora u odnosu na kameru.

Pošto su poene donosili lunarni moduli u obliku valjka i mesečevo kamenje u obliku kuglice, pri razvijanju mašinske vizije prvo se krenulo od prepoznavanja kuglica koje su kao geometrijsko telo mnogo pogodnije za detekciju zbog toga što posmatrano iz bilo koje projekcije, na 2D projekciji kamere dobijamo krug.

Na kraju se ispostavilo da ni prepoznavanje okruglih tela nije toliko lako koliko bi se očekivalo. Osnovni problem je bio kako dobro prepoznati kuglicu bele boje, pošto su samo bele kuglice donosile poene, i izolovati je u odnosu na pozadi, tj. ne prepoznati “lažne” kuglice na pozadini.

Medju prvim pokušajima bilo je prepoznavanje na osnovu boje. Pošto su kuglice bele trebalo je samo izolovati belu boju zar ne? Naravno to deluje jednostavno ali stvar se komplikuje kada se snimi par slika kuglice i pri drugacijim osvetljenjima se vidi da ta bela boja ustvari varira u velikom spektru RGB vrednosti. Pravljena je binarna slika koja je imala samo nule i jedinice, nule su označavale one pixele na slici cije RGB vrednosti ne zadovoljavaju skup RGB vrednosti bele boje, a jedinice su označavale pixele koji odgovaraju beloj boji. Taj skup RGB vrednosti koje odgovaraju beloj boji je ručno određivan , ali pri drugačijem osvetljenju taj skup se menjao tako da na ovaj način nije mogla pouzdano da se detektuje kuglica.

Druga grupa pokušaja zasnivala se na prepoznavanja oblika, kruga u slučaju kuglice. Algoritam se zasnivao na tome da se iz osnovne slike dobijene sa kamere za određeni pixel analiziraju svi susedni pixeli, i tamo gde je velika razlika izmedju RGB vrednosti pixela to je značilo da je to ivica nekog predmeta. Na ovaj način bi se dobio velik broj haotično raspoređenih linija. Podešavanjem osetljivosti tj. razlike vrednosti pixela (granice) koja će značiti da je to granica predmeta mogao se smanjiti broj linija i dobiti bolje rešenje. Ali te granice su opet određivane ručno i takodje su se menjale u zavisnosti od osvetljenja. Naravno od svih tih linija trebalo je izolovati one koje pripadaju krugu tj. Loptici. Tako da ni ovaj pristup nije bio dobar.

Kombinovanje prethodna dva metoda dalo je dosta bolje rezultate ali opet su bili problem granični parametri koji su se menjali u zavisnosti od vrste i jačine osvetljenja.

Naravno mi sigurno nismo prvi koji su se susreli sa ovim problemima, tako da trebalo je samo potražiti drugačiji pristup ljudi koji su doprineli razvoju OpenCV biblioteke. Taj drugačiji pristup ustvari predstavlja primena mašinskog učenja i neuronskih mreža pri detekciji. To u stvari i jeste logično, jer ako uzmemo u obzir da se neuronske mreže baš najviše i koriste gde rešenja zavise od dosta dinamičnih međusobno zavisnih parametara onda se dolazi do toga da je ovo idealan pristup problemu.

Rešenje je bilo korišćenje algoritama, u okviru OpenCV biblioteke, koji su od velikog broj slika predmeta koji se želi detektovati i još većeg broja slika same pozadine bez tog predmeta izvlačili pravilnosti i omogućavali dobro prepoznavanje kuglice.

Korišćeni su takozvani klasifikatori, HAAR classifier i Cascade classifier. Za prepoznavanje kuglica napravljeno je oko 700 slika kuglice pri različitom osvetljenjima i oko 4000 slika pozadine koje su bile najrazličitije moguće, bitno je bilo samo da nemaju belu kuglicu na sebi. Na osnovu toga je “istreniran” klasifikator koji je dosta brzo mogao da kaze da li na slici postoji kuglica ili ne i gde se ona nalazi.

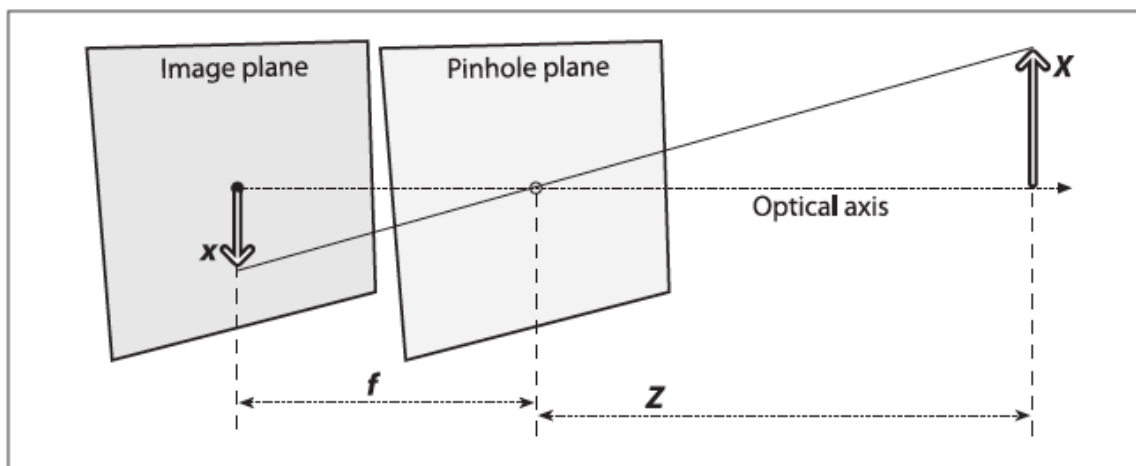
Posle ovog uspeha došlo se do ideje da se proba odrediti udaljenost tih kuglica u odnosu na kameru (slika 79), tj. u odnosu na robota. To se moglo postići jer je bila poznata veličina kuglica, žižna daljina za kameru, i broj veličina detektovane kuglice na slici gde se iz jednakosti trouglova dobijala normalna udaljenost u odnosu na kuglicu kao:

$$x = f \frac{X}{Z} , \quad (106)$$

gde je:

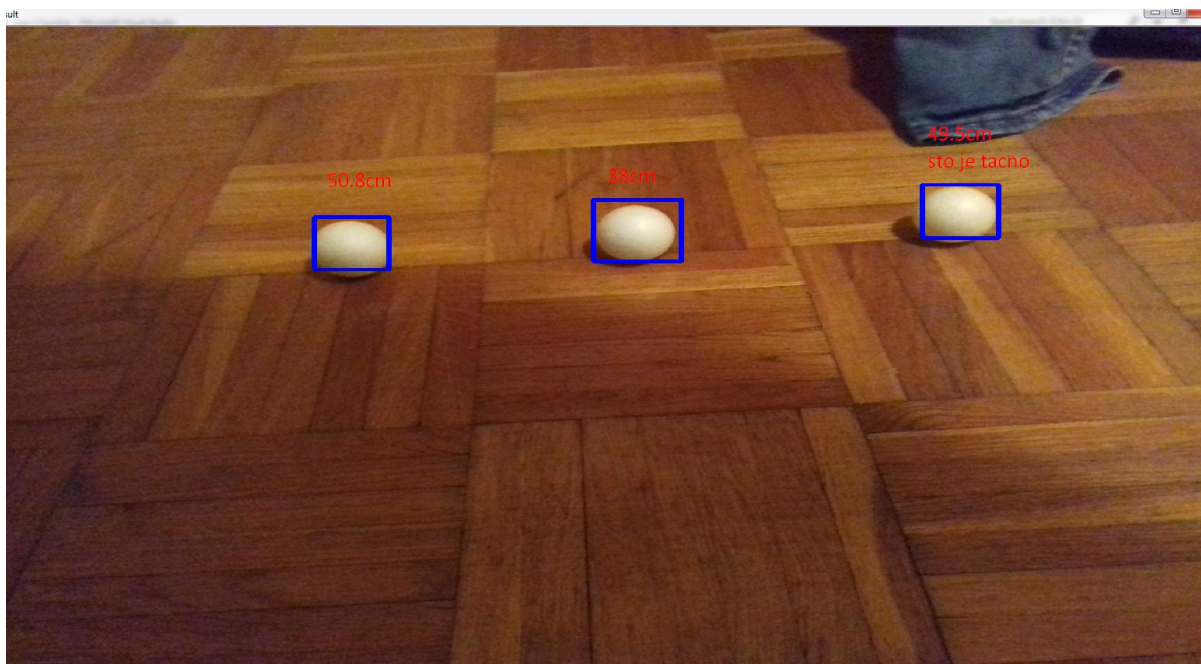
- x – veličina predmeta na slici (2D projekcija),
- X – veličina predmeta u realnom (3D) svetu,
- Z – udaljenost od kamere od predmeta u realnom (3D) svetu,
- f – žižna daljina kamere.

Naravno pošto u realnosti nije moguće napraviti savršeno sočivo moraju da se uvedu dodatni paramteri vezani za samu kameru. Tu spadaju paramteri koji govore koliko je ustvari je ustvari sočivo pomereno iz centra, takodje uvode se parametri koji govore o veličini distorzije.



Slika 79 – Način dobijanja udaljenosti objekta na osnovu slike.

Svi ovi parametri dobijaju se primenom odgovarajućeg OpenCV algoritma za kalibraciju gde se u odnosu na više slika šahovske table odgovarajuće veličine, iz drugačijih perspektiva dobijaju vrednosti parametara.



Slika 80 – Eksperiment detekcije udaljenosti kulice.

Na slici 80 je prikazano detektovanje normalne udaljenosti jednake za sve 3 kuglice (49.5cm). Sa slike se vidi da vraćanje udaljenosti do kuglice nije toliko precizno, ali može da se vidi otprilike kolika je udaljenost. Kako je vreme odmicalo odlučili smo se da robot neće skupljati kuglice tako da odradjena detekcija kuglica nije imala mnogo koristi. Pri detekciji lunarnih modula (valjaka) velik problem se javio u tome što valjak može da bude različito orijentisan u prostoru u odnosu na robota i tako ne dobijamo jednostavno rešenje kao što je to krug za loptice. Takodje OpenCV klasifikatori nisu nezavisni od ugla pod kojim je slika, tj ako se istrenira klasifikator koji će da prepozna valjak koji se postavljen vertikalno, taj isti klasifikator neće moći da prepozna valjak koji je postavljen pod uglom od 45 stepeni u odnosu na podlogu.

Jedno od rešenja je bilo da se slika rotira za po 10-15 stepeni i da se trazi vertikalni valjak na svakoj od njih, ali pošto je sve to trebalo da se odvija u realnom vremenu to bi značilo minimum 30 slika u sekundi i pomnoženo to sa brojem različitih rotacija, dovelo je do toga da hardver ne može da podrži to (u pitanju je ipak raspberry pi). Pošto neko drugo rešenje nije

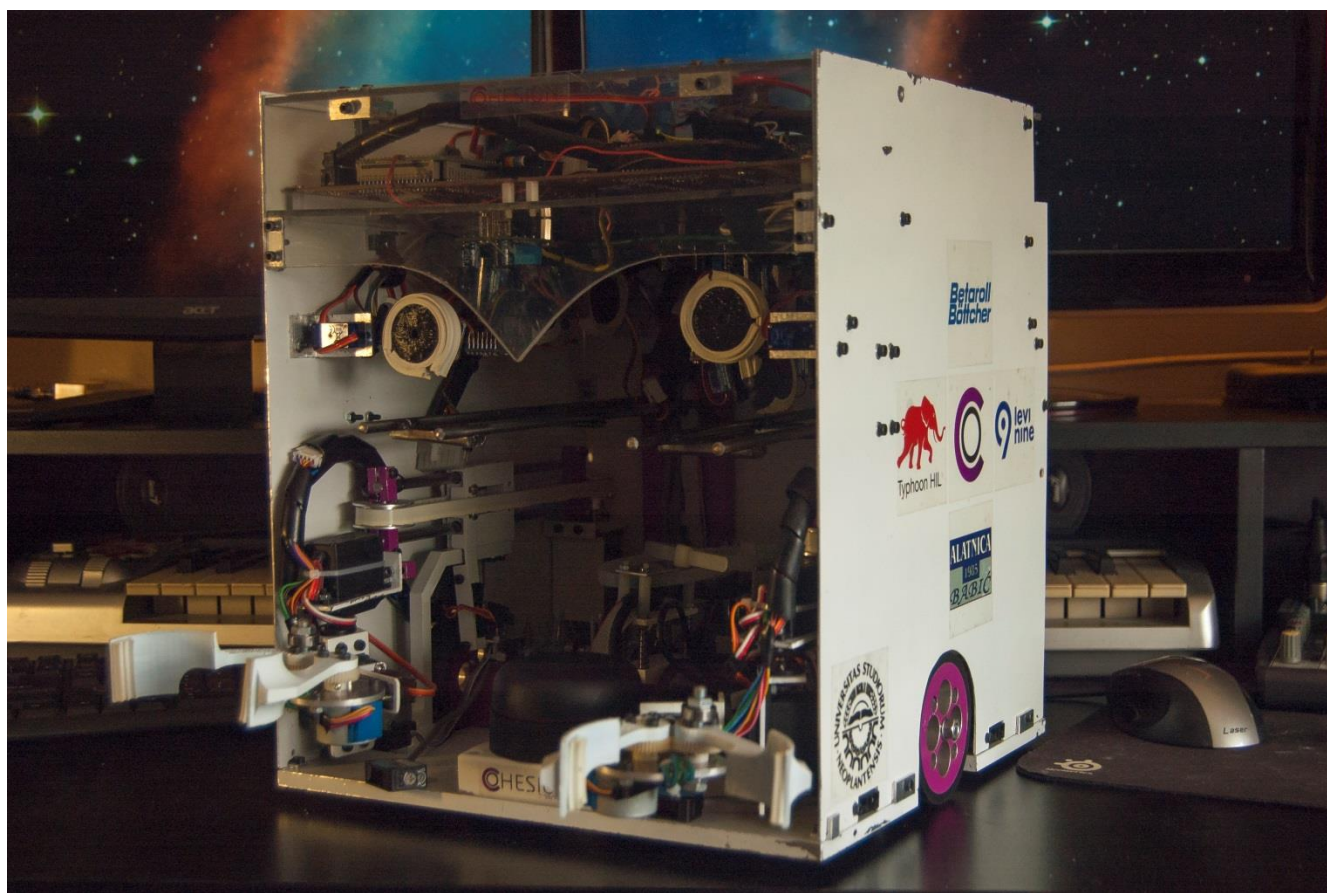
pronadjeno, a vremena do takmičenja je ostalo sve manje, na kraju je rešeno da se u potpunosti odustane od mašinske vizije i da se pažnja posveti nekim osnovnim stvarima, vezanim za kretanje robota.

Krajnja realuizacija

Kako matematički model robota nije u potpunosti razvijem do takmičenja, za kretanje robota se koristio model kretanja sa velikim ograničenjima kao što je opisano u poglavlju Kretanje – Kretanje robota sa velikim ograničenjima. Iako je kretanje robota bilo predefinisano, zadavanje novih putanja nije moralo da se radi promenom koda na mikrokontrolerima već jednostavnim dodavanjem novih tačaka u niz putanja. Na ovaj način robot je u krakom vremenskom preiodu mogao da se reprogramira.

Zbog nedostatka vremena, algoritam za snalaženje robora u prostoru uz pomoć IR senzora i LIDAR-a nije uspešno završen iako su eksperimenti pokazivali dobre rezultate pri jednostavnijim problemima. Na ovaj način nastala je ideja kako da se zaobiđe odometrije, jer kao što je rečeno, poseduje veliki broj mana na koje se ne može uticati.

Realizacija robotskog manipulatora za skupljanje lunarnih modula se pokazala kao dobro rešenje jer su skupljanje i skladištenje bili velike puzdanosti i ponovljivosti. Kako su manipulatori mogli da menjaju svoju orijentaciju i poziciju nezavisno od orijentacije i pozicije robota, postojala je velika fleksibilnost prilikom formiranja strategije za prikupljanje pojena.



Slika 81- Slika robota u kreamjem izdanju.

Zaključak

Jedan od razlog koji je učinio da se ovaj projekat vodi kao neuspešan jeste timska odluka da se reši veliki broj potpuno novih zadataka. Ovo je podrazumevalo potpuni redizajn robota u odnosu na prethodne generacije sa fokusom na odometriju. Iako se smatra da je ideja predstavljena u ovom projektu daleko superiornija u odnosu na do sadašnja rešenja, za njenu realizaciju je potrebno daleko više vremena i resursa nego što je bilo na raspolaganju, zbog istraživanja koje je potrebo uraditi i eksperimenata, kako bi se teorija praktično dokazala.

Kako je rešavanje odometrije robota uzelo najveći deo raspoloživog vremena i na kreju nije uspešno realizovana iz rezloga opisanih u projektu, za istraživanje i realizaciju ostalih ideja nije ostalo dovoljno vremena.

Pored svega, najveći uticaj na neuspešnost je imalo kašnjenje u dostavi elektronskih komponenti korišćenih na robotu. Robot je u obliku prikazanom na slici 81 u rukama tima trebao da bude krajem decembra 2016. godine ali se to dogodilo tek krajem februara 2017. godine čime je izgubljeno dragoceno vreme. Kako je ovo problem na koji ne može previše da se utiče, zaključuje se da je bilo bolje da se usvoje neka od dosadašnjih rešenja i da se na njih nadgrade nova. Smatra se da bi projekat pokazao daleko bolje rezultate da je usvojen način rešavanja odometrije kao što je to rađeno prethodnih godina i da su se na to nadogradile ostale ideje projekta.