# Razvoj robotske kolaborativne hvataljke

**Selak, Hrvoje**

**Master's thesis / Diplomski rad**

**2019**

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

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:235:054838

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-05-04**

*Repository / Repozitorij:*

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

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

**Hrvoje Selak**

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor: Student:

Prof. dr. sc. Bojan Jerbić, dipl. ing. Hrvoje Selak

Zagreb, 2019.

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

Zahvaljujem se mentoru prof. dr. sc. Bojanu Jerbiću na prihvaćanju mentorstva i korisnim savjetima pri izradi ovog rada i dr. sc. Marku Švaci na pomoći, vođenju kroz ovaj diplomski rad, čestim savjetima, konzultacijama i osiguravanju potrebnih dijelova za izradu ovog rada.

Zahvaljujem se kolegi Mariu Klasiću na pruženoj pomoći i mogućnosti printanja prototipova na 3D tiskaču kojeg je radio za svoj diplomski rad, kao i ostalim kolegama iz Laboratorija za projektiranje izradbenih i montažnih sustava.

Najviše se zahvaljujem svojim roditeljima koji su mi omogućili sve ovo, te uz njih, cijeloj obitelji, prijateljima i djevojci na neizmjernoj podršci koju su mi pružili tijekom studija i izrade ovog rada.

Hrvoje Selak

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

| Sveučilište u Zagrebu<br>Fakultet strojarstva i brodogradnje | |
|---|---|
| Datum: | Prilog: |
| Klasa: | |
| Ur. broj: | |

# DIPLOMSKI ZADATAK

Student: **HRVOJE SELAK** Mat. br.: 0035195513

Naslov rada na hrvatskom jeziku: **Razvoj robotske kolaborativne hvataljke**

Naslov rada na engleskom jeziku: **Development of robotic collaborative griper**

Opis zadatka:

Primjena kolaborativnih robota u industrijskim postrojenjima sve je veća zahvaljujući napretku tehnologije koji je omogućio siguran rad robota i ljudi. Kroz fizičku interakciju s robotom, kombinacijom ljudskih kognitivnih sposobnosti te robotske ponovljivosti i točnosti moguće je rješavati raznovrsne zadatke u domeni kolaborativnog sklapanja ili učenja robotskih zadataka. Jedna od ključnih komponenata današnjih kolaborativnih robota su inteligentne kolaborativne hvataljke koje omogućavaju jednostavnu i intuitivnu interakciju korištenjem senzora sile i tipkala integriranih u hvataljci.

U okviru diplomskog rada potrebno je napraviti pregled postojećih robotskih kolaborativnih hvataljki te izraditi vlastitu robotsku kolaborativnu hvataljku. Hvataljku je potrebno integrirati na postojeći šest-osni industrijski robot u Laboratoriju za projektiranje izradbenih i montažnih sustava. Hvataljka treba imati sljedeće mogućnosti:

1. Kompenzacija mase alata i tereta u šest osi koristeći integrirani senzor sile i momenata na prirubnici robota.
2. Upravljanje robotom u popustljivom načinu rada (kompenzacija gravitacije).
3. Učenje robotskog programa koji se sastoji od niza točaka i stanja hvataljke s mogućnošću poziva naučenog programa.
4. Upravljanje robotom kroz detekciju kolizija.
5. Kolaborativna hvataljka treba biti intuitivna za korištenje i upravljanje robotom, treba imati integrirane statusne LED diode, tipkala, zaobljeno kućište koje sakriva sve pomične elemente osim prstiju hvataljke.

Kolaborativnu hvataljku je potrebno primijeniti i verificirati u scenariju montaže jednostavnog sklopa koji se sastoji od najmanje dva ugradbena elementa.

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

Zadatak zadan:
15. studenog 2018.

Rok predaje rada:
17. siječnja 2019.

Predviđeni datum obrane:
23. siječnja 2019.
24. siječnja 2019.
25. siječnja 2019.

Zadatak zadao:

prof. dr. sc. Bojan Jerbić

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

# SADRŽAJ

SADRŽAJ ..... I
POPIS SLIKA ..... III
POPIS TABLICA ..... V
POPIS OZNAKA ..... VI
SAŽETAK ..... VII
SUMMARY ..... VIII
1. UVOD ..... 1
2. KOLABORATIVNA HVATALJKA ..... 2
2.1. *Fanuc* - kolaborativni roboti i hvataljke ..... 3
2.2. *Kuka* - kolaborativni roboti i hvataljke ..... 5
2.3. *Schunk* - kolaborativna hvataljka ..... 7
3. CILJ RADA ..... 9
4. OPREMA I ALATI ..... 11
4.1. Robot *Fanuc LR Mate 200iC - 5L* ..... 11
4.1.1. Upravljačka jedinica R-30iA Mate ..... 13
4.1.2. Upravljačka konzola ..... 15
4.2. Senzor sile *FS - 10iA* ..... 16
4.3. Računalni alati ..... 16
4.3.1. RoboGuide ..... 17
4.3.2. KAREL ..... 17
4.3.2.1. Struktura programa ..... 18
4.3.2.2. Tipovi varijabli ..... 19
4.3.2.3. Uvjetne naredbe [8] ..... 20
4.3.2.4. Petlje [8] ..... 21
5. OBLIKOVANJE KOLABORATIVNE HVATALJKE ..... 22
5.1. Početne verzije kolaborativne hvataljke ..... 22
5.2. Prirubnica ..... 24
5.3. Ovir ..... 26
5.4. Nosač i pločica za kameru ..... 28
5.5. Prsti hvataljke ..... 29
6. UGRADNJA TIPKALA ..... 31
7. RAZVOJ PROGRAMSKE PODRŠKE ZA KOLABORATIVNU HVATALJKU ..... 34
7.1. Prva verzija ..... 35
7.2. Komunikacija Karel - Matlab ..... 36
7.3. Homogene transformacije ..... 37
7.4. Druga verzija ..... 39
7.5. Treća verzija ..... 41
7.6. Završna (četvrta) verzija ..... 41

8. PRIMJENA KOLABORATIVNE HVATALJKE ..... 45
9. ZAKLJUČAK ..... 49
LITERATURA ..... 51
PRILOZI ..... 53

# POPIS SLIKA

Slika 1. Kolaborativna hvataljka [1] ..... 1
Slika 2. *Fanuc*-ov kolaborativni robot [3] ..... 4
Slika 3. Skica učenja robota [3] ..... 4
Slika 4. *KUKA LBR iisy* [4] ..... 5
Slika 5. Kolaborativna hvataljka za *LBR* robote [5] ..... 6
Slika 6. Suradnja robota i čovjeka [5] ..... 6
Slika 7. *Schunk*-ova kolaborativna hvataljka *Co-act JL1* [6] ..... 7
Slika 8. Detekcija kolizije kapacitivnim senzorom ..... 8
Slika 9. Različiti načini hvatanja ..... 8
Slika 10. *Fanuc LR Mate 200iC - 5L* ..... 11
Slika 11. Shematski prikaz radnog prostora robota *LR Mate 200iC - 5L* [8] ..... 12
Slika 12. Tehničke karakteristike robota *LR Mate 200iC - 5L* [8] ..... 13
Slika 13. Upravljačka jedinica *R-30iA Mate* ..... 14
Slika 14. Upravljačka konzola (eng. *Teach Pendant*) ..... 15
Slika 15. Senzor sile *FS - 10iA* ..... 16
Slika 16. Primjer *Karel* programa ..... 17
Slika 17. Model prve verzije kućišta kolaborativne hvataljke ..... 22
Slika 18. Model druge verzije ..... 23
Slika 19. Treća verzija modela ..... 23
Slika 20. Završni model kućišta kolaborativne hvataljke ..... 24
Slika 21. Završni model kolaborativne hvataljke - pogled od ispod ..... 24
Slika 22. Prirubnica - jedna komponenta ..... 25
Slika 23. Prirubnica ..... 25
Slika 24. Prirubnica pričvršćena na adaptersku ploču robota ..... 25
Slika 25. Jedan dio okvira ..... 26
Slika 26. Unutarnja strana okvira ..... 26
Slika 27. Pogled na dodatne značajke za unutarnji spoj vijcima i protiv savijanja prilikom držanja rukom ..... 27
Slika 28. Pogled na rebra radi povećanja krutosti ..... 27
Slika 29. Spoj okvira i prirubnice ..... 27

Slika 30. Pogled od ispod na kolaborativnu hvataljku ..... 28
Slika 31. Okvir kolaborativne hvataljke ..... 28
Slika 32. Model nosača ..... 29
Slika 33. Model pločice za kameru ..... 29
Slika 34. Model prstiju hvataljke ..... 30
Slika 35. Model prstiju hvataljke - drugi pogled ..... 30
Slika 36. Strujni krug ..... 31
Slika 37. Detaljna shema spajanja ..... 32
Slika 38. Tipkala spojena na kabel ..... 33
Slika 39. Provjera signala i LED dioda ..... 33
Slika 40. Početna pozicija robota ..... 34
Slika 41. Druga pozicija robota ..... 35
Slika 42. Struktura TCP/IP protokola [13] ..... 36
Slika 43. Primjer homogenih transformacija [14] ..... 37
Slika 44. Prva orijentacija - usporedba ..... 39
Slika 45. Druga orijentacija - usporedba ..... 39
Slika 46. Treća orijentacija - usporedba ..... 40
Slika 47. Prikaz stanja primljene poruke ..... 42
Slika 48. Model kutije ..... 45
Slika 49. Model poklopca ..... 45
Slika 50. Sklopni model kutije i poklopca ..... 45
Slika 51. Isprintani 3D modeli kutije i poklopca ..... 46
Slika 52. Kolaborativna hvataljka s ugrađenim prstima ..... 46
Slika 53. Prikaz učenja robota ..... 47
Slika 54. Prelazak u izvršavanje naučene operacije ..... 47
Slika 55. Izvršavanje naučene operacije ..... 48

# POPIS TABLICA

Tablica 1. Usporedba kolaborativnih i industrijskih robota [2] ..... 3
Tablica 2. Tehničke karakteristike upravljačke jedinice [10] ..... 14

# POPIS OZNAKA

| Oznaka | Jedinica | Opis |
|---|---|---|
| $\mathbf{A}$ | | Matrica homogenih transformacija |
| ${}^{m}\mathbf{A}_{n}$ | | Matrica homogenih transformacija i orijentacija sustava *n* prema sustavu *m* |
| $F_{\mathrm{g}}$ | $\mathrm{N/mm^2}$ | Težina dijelova ispod senzora sile |
| $F_{\mathrm{x0}}$ | $\mathrm{N/mm^2}$ | Početna sila u *x* smjeru koordinatnog sustava senzora |
| $F_{\mathrm{y0}}$ | $\mathrm{N/mm^2}$ | Početna sila u *y* smjeru koordinatnog sustava senzora |
| $F_{\mathrm{z0}}$ | $\mathrm{N/mm^2}$ | Početna sila u *z* smjeru koordinatnog sustava senzora |
| $F_{\mathrm{zd}}$ | $\mathrm{N/mm^2}$ | Sila u početnoj poziciji robota u z smjeru koordinatnog sustava senzora |
| $F_{\mathrm{zg}}$ | $\mathrm{N/mm^2}$ | Sila u drugoj poziciji robota u z smjeru koordinatnog sustava senzora |
| $I_{LED}$ | A | Struja potrebna za LED diodu |
| $I_R$ | A | Struja koju daje robot |
| $M_{\mathrm{x}}$ | Nmm | Moment oko osi *x* |
| $M_{\mathrm{y}}$ | Nmm | Moment oko osi *y* |
| $M_{\mathrm{z}}$ | Nmm | Moment oko osi *z* |
| ${}^{m}\mathbf{p}_{n}$ | | Matrica translacija sustava *n* prema sustavu *m* |
| $p$ | ° | Rotacija oko *y* osi |
| ${}^{m}\mathbf{R}_{n}$ | | Matrica rotacija sustava *n* prema sustavu *m* |
| $r$ | ° | Rotacija oko *z* osi |
| $R$ | Ω | Otpornik |
| $R_{LED}$ | Ω | Otpor LED diode |
| $\mathbf{T}_{x}$ | | Matrica transformacija oko x-osi |
| $\mathbf{T}_{y}$ | | Matrica transformacija oko y-osi |
| $\mathbf{T}_{z}$ | | Matrica transformacija oko z-osi |
| $U_{LED}$ | V | Napon na LED diodi |
| $U_R$ | V | Napon koji daje robot |
| $w$ | ° | Rotacija oko *x* osi |
| $x$ | mm | Pomak u *x* smjeru koordinatnog sustava |
| $y$ | mm | Pomak u *y* smjeru koordinatnog sustava |
| $z$ | mm | Pomak u *z* smjeru koordinatnog sustava |

# SAŽETAK

Kolaborativni roboti se sve više primjenjuju u industrijskim postrojenjima zbog njihove sigurnosti u radu s čovjekom. Robotska ponovljivost i točnost, u fizičkoj interakciji s čovjekom, rezultira u rješavanju raznovrsnih zadataka u domeni kolaborativnog sklapanja ili učenja robotskih operacija. Prednost takvih robota je što rade „rame uz rame“ s čovjekom jer sigurnosne ograde nisu potrebne što rezultira uštedom prostora. Ključna komponenta kolaborativnih robota su inteligentne kolaborativne hvataljke. Funkcija takvih hvataljki je mogućnost ručnog vođenja robota prostorom i učenja određenih operacija preko tipkala. U ovom će radu biti opisan razvoj kolaborativne hvataljke od njezinog oblikovanja, odabira tipkala, ugradnje tipkala pa sve do razvoja programske podrške za nju. Kolaborativna hvataljka mora zadovoljiti određene funkcije, kao što su pamćenje točke, otvaranje - zatvaranje hvataljke, te između ostalog, mogućnost reprogramiranja novih operacija.

Ključne riječi: kolaborativni robot, industrijski robot, kolaborativna hvataljka, *Fanuc*, senzor sile, konstrukcija kućišta kolaborativne hvataljke

# SUMMARY

Collaborative robots are increasingly being used in industrial plants due to their safety in working with man. Robotic repeatability and accuracy, in accordance to physical interaction with a human, results in solving various tasks in the domain of collaborative assembly or robotic learning. The advantage of such robots is that they work "side by side" with a man because security fences are not needed which results in space savings. The key component of collaborative robots are intelligent collaborative grippers. Their function is the ability to manually control robot and teach it specific operations. This thesis will describe the development of a collaborative gripper from its design, push-buttons installation and its software development. Collaborative gripper must meet certain functions, such as saving position in positional register, opening and closing gripper and that can be reprogrammed.

Key words: collaborative robot, industrial robot, collaborative gripper, *Fanuc*, force sensor, housing design of collaborative gripper

# 1. UVOD

Primjena kolaborativnih robota u industrijskim postrojenjima sve je veća zahvaljujući napretku tehnologije koji je omogućio siguran rad robota i ljudi. Kolaborativni roboti ili skraćeno „koboti“ dolaze od engleske riječi *cobots* (*collaborative robots*). Kroz fizičku interakciju s robotom, kombinacijom ljudskih kognitivnih sposobnosti te robotske ponovljivosti i točnosti, moguće je rješavati raznovrsne zadatke u domeni kolaborativnog sklapanja ili učenja robotskih zadataka. Opremljen je integriranim senzorima koji pri nepredviđenom kontaktu detektiraju promjenu sile i usporavaju robota na male brzine u svrhu smanjenja kinetičke energije koja nije opasna po zdravlje čovjeka ili robota u potpunosti zaustavljaju. Nakon što je opasnost prošla, kolaborativnog robota se pokreće pritiskom na tipkalo nastavljajući radnju od točke na kojoj se zaustavio. Na taj način kolaborativni roboti pomažu čovjeku pri izvršavanju napornih i teških poslova, primjerice ako je potrebno prenijeti teški teret s jednog na drugo mjesto, čovjek upravlja robotom dok robot drži teret. Rezultat toga je efikasnije odrađen posao, ušteda na vremenu i najvažnije, sigurnost čovjeka.

Jedna od ključnih komponenti današnjih kolaborativnih robota su inteligentne kolaborativne hvataljke koje omogućavaju jednostavnu i intuitivnu interakciju korištenjem senzora sile i tipkala integriranih u hvataljkama. Na slici 1. prikazana je kolaborativna hvataljka s tipkalima proizvođača *Fanuc.*

**Slika 1. Kolaborativna hvataljka [1]**

# 2. KOLABORATIVNA HVATALJKA

Poveznica između kolaborativnog robota i čovjeka je upravo kolaborativna hvataljka. Služi da bi čovjek na jednostavan i lagani način upravljao robotom, te pomicao hvataljku po prostoru. Sastoji se od kućišta, tipkala, senzora sile i najčešće ima na sebi ručke koje služe za vođenje robota. U kućištu je smještena elektronika (različiti robotski signali na koje se spajaju tipkala), tipkala služe za pamćenje točke u prostoru, otvaranje ili zatvaranje hvataljke, može imati i *STOP* tipkalo za trenutno zaustavljanje i prekid svih operacija robota. Senzor sile očitava promjene iznosa sile na njemu i ako je robot u stanju učenja, robota će se moći voditi po prostoru zbog promjene sile, no ako je robot u stanju izvođenja naučenog programa, robot će se zaustaviti zbog nepredviđene promjene sile.

Dvije najvažnije karakteristike kolaborativnih hvataljki su sigurnost u interakciji s čovjekom i smanjenje čovjekovog fizičkog napora. Neke od primjena kolaborativnih hvataljki su *pick and place* operacije, odnosno operacije u kojima robot uzima predmet i prenosi ga na drugo mjesto. Također se mogu primjenjivati kad treba neki teret podići radi lakše ugradnje. Teret od 10 kilograma se ne čini težak, ali kad takvu operaciju treba ponoviti više stotina puta u danu, ona postaje naporna i iscrpljujuća za čovjeka i u takvim situacijama kolaborativne hvataljke dolaze do izražaja.

Neka od važnijih karakteristika kolaborativne hvataljke je njezina nosivost koja je ograničena na 10 kg zbog sigurnosnih razloga. Stoga je potrebno izabrati hvataljku s boljim zahvatom, a da bude što lakša. Druga važna karakteristika je ciklus izvođenja operacije. Jako brze hvataljke mogu napraviti veliku štetu ako kolaborativni robot i čovjek dijele radni prostor pa treba računati da ciklus izvođenja operacije bude umjeren. Ponovljivost bi trebala biti što veća da bi se radni zadaci što efikasnije i točnije izvodili. Izdržljivost kolaborativne hvataljke za kolaborativne robote ne mora biti velika kao za industrijske robote. Kod industrijskih robota, hvataljka mora izdržati i terete do 100 kg, a poželjno je da ima i detektor sudara s okolinom. No kod kolaborativnih robota, zbog male nosivosti hvataljke i ograničenih brzina i sila, rizik od oštećenja kolaborativne hvataljke je jako mali, ipak, sigurnije je izabrati pouzdaniju hvataljku da bi što manje zastoja bilo radi njezinog održavanja. Operacije koje kolaborativni

roboti izvode su najčešće *pick and place* operacije. Zbog toga je najbolje izabrati hvataljku koja može manipulirati sa širokim rasponom dijelova, po dimenzijama, obliku i silama hvatanja. S obzirom na to da se kolaborativni roboti koriste u okruženju s ljudima, okolina je puno čišća, bez prašine i vlage u zraku, a to pomaže robotu prvi efikasnijem obavljanju operacija jer zahtjeva manje preventivnog održavanja. U tablici 1. navedenoj ispod, nalazi se kratka usporedba karakteristika kolaborativnih i industrijskih robota [2].

**Tablica 1. Usporedba kolaborativnih i industrijskih robota [2]**

| | **Kolaborativni roboti** | **Industrijski roboti** |
|---|---|---|
| **Doseg** | Kao i čovjekov doseg | Do 6 m |
| **Nosivost** | Otprilike 10 kg | Do 900 kg |
| **Brzina** | Ograničena | Velika |
| **Ponovljivost** | Niska do srednja | Visoka |
| **Programiranje** | Ručno pomicanje i učenje robota | Privjesak za učenje, *offline* programiranje |
| **Sigurnost** | Zajednički prostor s čovjekom | Zaštitne ograde |
| **Operacije** | *Pick and place*, paketiranje | Roboti za rukovanje, zavarivanje, doziranje, sastavljanje… |
| **Integracija** | Cjelina za sebe | Dio veće cjeline |

Nakon svega navedenog, zaključak je da se kolaborativna hvataljka susreće s izazovima koje treba konstrukcijski i programski riješiti, a njezina izvedba mora biti fleksibilna, jednostavna za rukovanje i mora biti sigurna u radu s čovjekom [2].

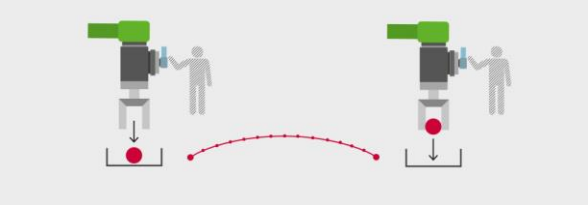
## 2.1. *Fanuc* - kolaborativni roboti i hvataljke

*Fanuc*-ovog kolaborativnog robota, CR serije, prikazanog na slici 2, može se prepoznati po njegovoj zelenoj boji. Vrlo su sigurni u radu s čovjekom, te im zaštitne ograde nisu potrebne. Koriste različite senzore, od detekcije sudara do automatskog pomicanja robota unazad, a da bi se spriječilo priklještenje, veći roboti su omotani u mekanu navlaku. Mogu biti opremljeni,

3D prostornim senzorom s kojim prepoznaju i izuzimaju različite predmete iz kutija, senzorom sile koji je koristan pri montaži sklopova ili vizijskim sustavom koji služi za lociranje različitih predmeta, čitanje bar kodova i sortiranje prema boji. Kolaborativni roboti se koriste u automobilskoj industriji pomažući radnicima pri podizanju teških tereta, a samim time se smanjuje rizik od ozljeda. Koriste se i za kontrolu kvalitete, paketiranje, paletizaciju, monotone i opasne radne zadatke. Za lakše rukovanje robotom koristi se kolaborativna hvataljka koja je prethodno prikazana u uvodnom poglavlju na slici 1 [3].

**Slika 2. *Fanuc*-ov kolaborativni robot [3]**

Kolaborativna hvataljka sastoji se od dvije ručke za lakše vođenje robota po prostoru. Na sebi ima tri tipkala za pamćenje točke i stanja hvataljke, ima tipkalo STOP koje zaustavlja robota u slučaju opasnosti i tipkalo koje kad je pritisnuto, pokreće se način rada u kojem se robota uči tražene operacije. Princip učenja robota, prikazan na slici 3., je takav da se robota dovede do zadanog predmeta, hvataljkom se primi predmete, robot se vodi do mjesta gdje treba prebaciti predmet pritom se pamte točke trajektorije robota i kad smo doveli robota do pozicije gdje treba prebaciti predmet, pokreće se način rada u kojem robot obavlja naučenu operaciju [3].



**Slika 3. Skica učenja robota [3]**

## 2.2. *Kuka* - kolaborativni roboti i hvataljke

*Kuka* je razvila prve robote koji su odobreni za ljudsko - robotsku suradnju (eng. *Human - robot collaboration*, skraćeno *HRC*): *KUKA LBR iisy*, prikazan ispod na slici 4. i *KUKA LBR iiwa* (LBR je skraćenica od njemačke riječi *Leichtbauroboter*, što u prijevodu znači lagani robot). Koriste inteligentnu tehnologiju upravljanja, senzore visokih performansi i jako razvijenu softversku podršku čime omogućavaju potpuno nova kolaborativna rješenja u proizvodnoj tehnologiji. *KUKA LBR* roboti su mobilni i fleksibilni u izvršavanju različitih zadataka, stoga ih se može postaviti na bilo koju lokaciju i u bilo koju svrhu ovisno što proizvodnja zahtjeva [4].

**Slika 4. *KUKA LBR iisy* [4]**

Neke prednosti korištenja *KUKA*-inih *LBR* robota su:

- maksimizirana fleksibilnost u proizvodnji,
- oslobađanje radnika od ergonomski nepovoljnih poslova koji prije nisu mogli biti automatizirani,
- smanjeni rizik od ozljeda i infekcija korištenjem specijalnih *HRC* hvataljki,
- visoka kvaliteta izvođenih operacija, bez potrebe za novim investicijama,
- povećana produktivnost i poboljšana složenost sustava zahvaljujući integriranim senzorima [4].

*KUKA LBR iisy* robot je vrlo efikasan i u serijskoj proizvodnji u industriji i u dinamičkom, nestrukturiranom okruženju koje je bilo nezamislivo automatizirati. Sadrži snažnu programsku podršku koja osigurava stručnjacima da implementiraju najsloženije aplikacije na njega. Novo razvijeno korisničko grafičko sučelje osigurava brzu i intuitivnu parametrizaciju radnih sekvenci bez potrebe za određenim programskim znanjima. A program se može učiti i preko jednostavne kolaborativne hvataljke, prikazane na slici 5 [5].

**Slika 5. Kolaborativna hvataljka za *LBR* robote [5]**

*LBR iisy* se može programirati da se automatski adaptira na novu radnu okolinu. U *search* (traženje) načinu rada, pronalazi bitne vijčane pozicije, izvodi fino namještanje vijka i autonomno ih priteže. Čak i u prisustvu ostalih strojeva, kolaborativni robot radi neovisno i pouzdano. Radna mjesta se ne moraju mijenjati jer se LBR fleksibilno prilagođava okolnim uvjetima. Može razmjenjivati znanje s ostalim kolaborativnim robotima u blizini, reagira sigurno i osjetljivo s ljudima, prikazano na slici 6, a kad radi samostalno lako postiže potrebne performanse i visoke brzine ovisno o primjeni [5].

**Slika 6. Suradnja robota i čovjeka [5]**

Sadrži visokokvalitetne senzore okretnog momenta u svakom zglobu u svih šest osi. Reagiraju na najmanju promjenu vanjske sile i time sprječavaju sudar. Značajno pomažu u poboljšanju radnog okruženja. Pametna podjela zadataka između kolaborativnih robota i ljudi dovodi do povećane preciznosti i brzine što rezultira višom kvalitetom, fleksibilnijim radnim mjestima i poboljšanom kvalitetom proizvoda.

## 2.3. *Schunk* - kolaborativna hvataljka

*Schunk* je jedan od najpoznatijih proizvođača robotskih hvataljki. Pod imenom *Co-act JL1* (skraćeno od *collaborative actuator*) krije se pametna kolaborativna hvataljka koja ima izravnu interakciju s ljudima, prikazana na slici 7 [6].

**Slika 7. *Schunk*-ova kolaborativna hvataljka *Co-act JL1* [6]**

Prilagodbe u procesu hvatanja predmeta se mogu napraviti u realnom vremenu koristeći različite sustave senzora. Koristi i senzore za komunikaciju, snimanje i procjenu okolnih i operativnih uvjeta. U budućnosti će *Co-act* hvataljke slati sve bitne podatke o procesima i okolini na kontrolne i procesne upravljačke sustave. Naglasak će biti na inteligentnom toku materijala, optimizaciji procesa i kontinuiranoj dokumentaciji [6].

Ima ugrađen kapacitivni senzor koji uočava prepreke i sprječava koliziju s njima, prikazano na slici 8. Zaslon osjetljiv na dodir omogućava komunikaciju s hvataljkom, učenje robota i odabir različitih načina hvatanja predmeta, a može se birati između paralelnog hvatanja predmeta, hvatanja predmeta pod kutem (za predmete kružnog presjeka) ili kombinacija oba način, prikazani na slici 9. Kamera, koja je smještena između prstiju hvataljke, služi za detekciju okoline, razlikovanje i traženje objekata. Sigurno ograničenje sile osigurava točnu silu hvatanja pri svakom rukovanju, čak i pri rukovanju teških predmeta. Statusne LED diode govore operateru status hvataljke i identifikaciju predmeta.

**Slika 8. Detekcija kolizije kapacitivnim senzorom**

**Slika 9. Različiti načini hvatanja**

# 3. CILJ RADA

Kao što je već navedeno u uvodnom poglavlju, sve više se uvode kolaborativni roboti u proizvodnju. Njihova je primjena jako raznolika, od jednostavnih *pick and place* operacija do podizanja teških predmeta radi lakše ugradnje. S obzirom na to da su kolaborativni roboti skupi, a kompletna pretvorba industrijskog robota u kolaborativnog zahtjeva mnogo opreme, rada i vremena, temelj ovog diplomskog rada je da se napravi kolaborativna hvataljka koja će se ugraditi na industrijskog robota i time olakšati njegovo rukovođenje.

Kolaborativna hvataljka se razvija za industrijskog *Fanuc*-ovog robota *LR Mate 200iC- 5L*, koji se nalazi u Laboratoriju za projektiranje izradbenih i montažnih sustava, a detaljnije je opisan u sljedećem poglavlju. Funkcije koje kolaborativna hvataljka mora imati su mogućnost vođenja robota po prostoru pritiskom na tipkalo, pamćenje točke u prostoru, otvaranje - zatvaranje hvataljke te izvršavanje naučenog programa. Mora imati mogućnost vođenja robota različitim brzinama, a sve na temelju primijenjene vanjske sile. Tipkala na sebi imaju LED diode radi lakšeg registriranja stanja tipkala. Također, robot mora zaustaviti svoju radnju, ako dođe do kolizije kolaborativne hvataljke s okolnim predmetima. Da bi sve to bilo izvedivo potrebno je napraviti program za robota koji sadrži sve nabrojane funkcije i izraditi kućište kolaborativne hvataljke koje će na sebi imati tipkala za upravljanje robotom.

Cilj i motivacija ovog rada leže u tome što su svi ti sustavi zatvoreni. Poznati su načini rada kolaborativnog robota i hvataljke no ne i sama pozadina, odnosno svi procesi i programi koji se vrte da bi kolaborativni robot i hvataljka radili na najbolji mogući način. Svi proizvođači kolaborativnih robota i hvataljki imaju posebne odjele i timove s velikim brojem inženjera koji usavršavaju takve stvari, dok će se u Laboratoriju za projektiranje izradbenih i montažnih sustava to ostvariti na industrijskom robotu korištenjem jednog senzora sile. Ne postoji određen smjer za rješavanje ovog problema. Potrebno je krenuti od jednostavniji radnji, kao na primjer vođenje robota u jednom smjeru, sve do onih zahtjevnijih, primjerice rotiranje i translatiranje robota istovremeno. S obzirom na to da ne postoji jednoznačno rješenje, javljaju se dodatni izazovi i potrebno je sagledati problem iz šire perspektive, isprobati različite načine

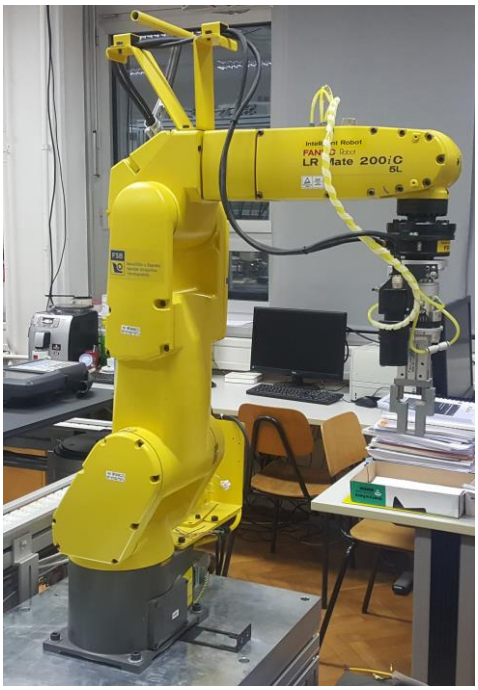
programiranja robota i odabrati onaj najpovoljniji, kontinuirano poboljšavati program i nadograđivati ga.

Dodatni razlog za odabir ove teme je u tome što će poslužiti u edukacijske svrhe. U Laboratoriju za projektiranje izradbenih i montažnih sustava može se raditi na industrijskim *Fanuc*-ovim i kolaborativnim *Universal*-ovim robotima, no ne postoji ni jedna kolaborativna hvataljka s tipkalima. Pick and place operacije koje se inače rade preko upravljačke konzole, moći će se raditi korištenjem samo kolaborativne hvataljke.

U sljedećim poglavljima opisan je tijek razvoja kolaborativne hvataljke te sva oprema i alati koja se koristila pri njenoj izradi.

# 4. OPREMA I ALATI

Za izradu ovog diplomskog zadatka korišten je *Fanuc*-ov industrijski robot *LR Mate 200 iC - 5L* koji se nalazi u Laboratoriju za projektiranje izradbenih i montažnih sustava, prikazan na slici 10. Robot ima instaliran senzor sile *FS - 10iA* koji se koristio pri programiranju robota. Programiranje robota napravljeno je u računalnom alatu *Karel*, koji se nalazi u *RoboGuide* paketu. *Karel* ne podržava rad s matricama, te je zbog toga za obradu podataka sa senzora sile korišten računalni alat *Matlab*. Modeli kolaborativne hvataljke su napravljeni u softveru *Catia*. Ostala oprema koja se koristila ili je ugrađena su tipkala, vanjski konektor za upravljanje s robotskim signalima, 3D tiskač za izradu prototipova, vijci, matice…

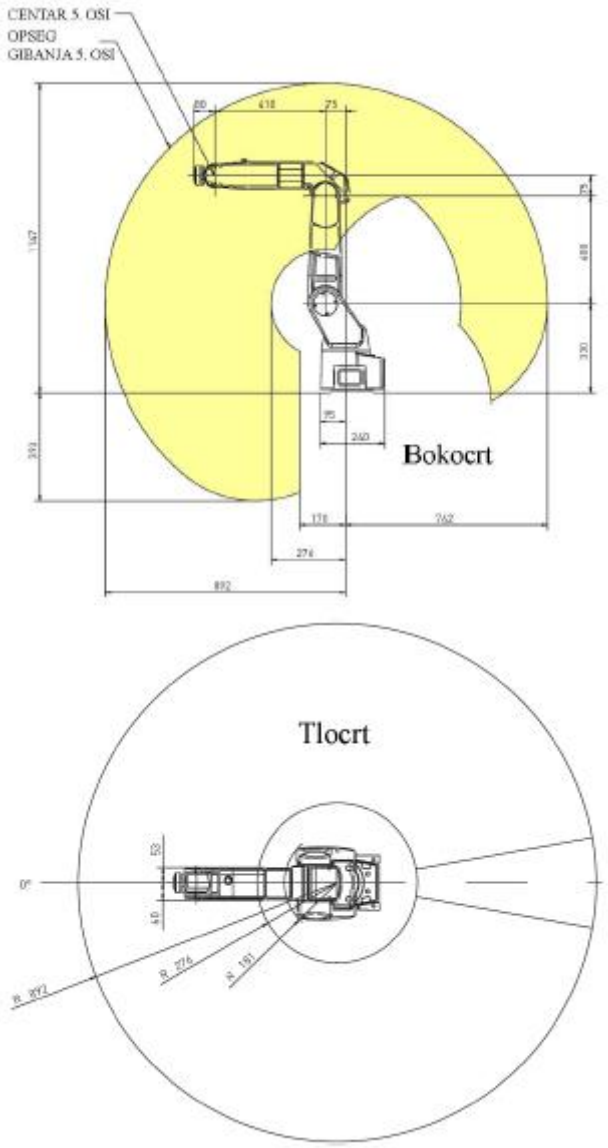


**Slika 10.** ***Fanuc LR Mate 200iC - 5L***

## 4.1. Robot *Fanuc LR Mate 200iC - 5L*

Robot *LR Mate 200iC - 5L* je šest-osni što znači da ima šest stupnjeva slobode gibanja plus otvaranje/zatvaranje hvataljke. Svi stupnjevi slobode gibanja su rotacijski. Jedan je od manjih *Fanuc*-ovih robota, ali bez obzira na to ima široku primjenu u industriji. Od posluživanja strojeva, rukovanja materijalom, montaže pa sve do doziranja i uklanjanja materijala. Uska ruka i malo postolje omogućuju mu rad u skučenim prostorima. Doseg mu je 892 mm, što je najveći doseg iz serije 200iC. Ima nosivost od 5 kg, a ponovljivost od 0,03 mm. Veće brzine i

mogućnost izokretanja ruke pospješuju njegovu primjenu i funkciju. Može se postaviti na različite pozicije (pod, strop, pod kutem ili unutar drugih strojeva) ovisno o primjeni. Ima veću krutost i naprednu servo tehnologiju koja omogućuje nesmetano gibanje bez vibracija pri velikim brzinama [7]. Slika 11. prikazuje shematski prikaz radnog prostora robota *LR Mate 200iC - 5L*.




**Slika 11. Shematski prikaz radnog prostora robota *LR Mate 200iC - 5L* [8]**

Na slici 12. prikazana je tablica s osnovnim tehničkim karakteristikama robota. Oznake J1 - J6 označavaju zglobove robota. Maksimalna brzina se odnosi kad je robot neopterećen, a u

nosivost robota nije uključen senzor sile, kamera i hvataljka tako da je stvarna nosivost nešto manja [8].

| | | |
|---|---|---|
| Broj osi | | 6 |
| Težina | | 29 kg |
| Doseg | | 892 mm |
| Točnost ponavljanja | | ±0.03 mm |
| Maksimalna nosivost na zglobu | | 5 kg |
| Opseg gibanja | J1 | 340° (5,93 rad) |
| | J2 | 230° (4,01 rad) |
| | J3 | 373° (6,51 rad) |
| | J4 | 380° (6,63 rad) |
| | J5 | 240° (4,19 rad) |
| | J6 | 720° (12,57 rad) |
| Maksimalna brzina | J1 | 270°/s (4,71 rad/s) |
| | J2 | 270°/s (4,71 rad/s) |
| | J3 | 270°/s (4,71 rad/s) |
| | J4 | 450°/s (7,85 rad/s) |
| | J5 | 450°/s (7,85 rad/s) |
| | J6 | 720°/s (12,57 rad/s) |

**Slika 12. Tehničke karakteristike robota *LR Mate 200iC - 5L* [8]**

### *4.1.1. Upravljačka jedinica R-30iA Mate*

Upravljačka jedinica (eng. *Controller*), prikazana na slici 13., koristi se za upravljanje robotom *LR Mate 200iC - 5L*. Ima kompaktnu konstrukciju pa je lako prenosiva. Neke od prednosti su vrijeme paljenja koje je manje od minute i brzo ponovno pokretanje, što uvelike povećava vrijeme rada robota. Postoji zaštita u slučaju gubitka električne energije tako da je robot zapamtio sve što je napravio i gdje je stao, pa pri ponovnom paljenju nastavlja od točke na kojoj je stao. Nakon ponovnog pokretanja, program provjerava je li se robot pomaknuo iz tolerancija da bi spriječio neočekivana gibanja ili kolizije. Zahtjeva minimalno održavanje zbog smanjenog broja komponenti i jednostavnim pristupom do njih. Upravljačka jedinica osigurava jednostavno skidanje ili učitavanje robotskih programa i korištenje električnih ulaznih i izlaznih signala u svrhu upravljanja stanjima hvataljke i ostalih vanjskih alata [9].

**Slika 13. Upravljačka jedinica *R-30iA Mate***

Na upravljačkoj jedinici se može birati između tri različita načina izvođenja programa, a to su: T1, T2 i AUTO.

T1 način rada je testni način rada u kojem je brzina ograničena na 250 mm/s. Koristi se prilikom učenja robota i za ispitivanje programa. Programi se izvršavaju uz pomoć upravljačke konzole. U T2 načinu rada nema ograničenja brzine, ali se programi također moraju izvoditi uz pomoć upravljačke konzole. Koristi se kad treba provjeriti vrijeme ciklusa izvođenja programa. Način rada AUTO je automatski način rada. Nema ograničenja brzine, a nije potrebna ni upravljačka konzola za izvođenje programa. Koristi se u industrijskim postrojenjima gdje robot iznova izvršava naučenu operaciju [8]. U tablici 2. prikazane se tehničke karakteristike upravljačke jedinice R-30iA Mate.

**Tablica 2. Tehničke karakteristike upravljačke jedinice [10]**

| | |
|---|---|
| **Težina** | **55 kg** |
| **Dužina** | 588 mm |
| **Širina** | 320 mm |
| **Visina** | 400 mm |

### *4.1.2. Upravljačka konzola*

Upravljačka konzola ili privjesak za učenje (eng. *Teach Pendant*), prikazana na slici 14., prvenstveno služi za programiranje robota, ručno pomicanje robota u tri translacije (*x*, *y*, *z*) i tri rotacije (*w*, *p*, *r*), a preko nje se dobivaju različite informacije ili upozorenja.

**Slika 14. Upravljačka konzola (eng. *Teach Pendant*)**

Može se koristiti za:

- pregled trenutne pozicije robota (u korisničkom koordinatnom sustavu, koordinatnom sustavu svijeta robota…),
- trenutne rotacije zglobova robota,
- pregled stanja ulaznih i izlaznih signala (digitalnih, analognih, robotskih),
- pisanje programa [8]…

## 4.2. Senzor sile *FS - 10iA*

Na robot je postavljen senzor sile *FS - 10iA*, prikazan na slici 15. Očitava tri komponente sile: $F_x$, $F_y$ i $F_z$ i momente oko svake osi: $M_x$, $M_y$, $M_z$. Sa senzorom sile robot ostvaruje dodatne mogućnosti. Primjerice umetanje radnog dijela u držač alata s malim tolerancijama bez zaglavljivanja. Može polirati metalne dijelove tako što mjeri primijenjenu silu i kontrolira putanju robota u realnom vremenu i na taj način drži alat pritisnut konstantnom silom na obrađivani dio. Omogućuje lakše sastavljanje velikih i teških sklopova. Senzor sile maksimizira efektivno vrijeme rada u oblikovanju proizvoda. A može i provoditi inspekciju provrta ili osovina pomoću graničnog mjerača pričvršćenog ispod senzora sile [11].

**Slika 15. Senzor sile *FS - 10iA***

## 4.3. Računalni alati

Za rješavanje zadataka potreban je *Fanuc*-ov programski alat *Karel* za programiranje njihovih robota. *Karel* je dio softverskog paketa *RoboGuide* koji služi za *offline* programiranje *Fanuc* robota. U rješavanju zadatka je korišten i *Matlab* i *Catia*. *Matlab* je korišten za obradu

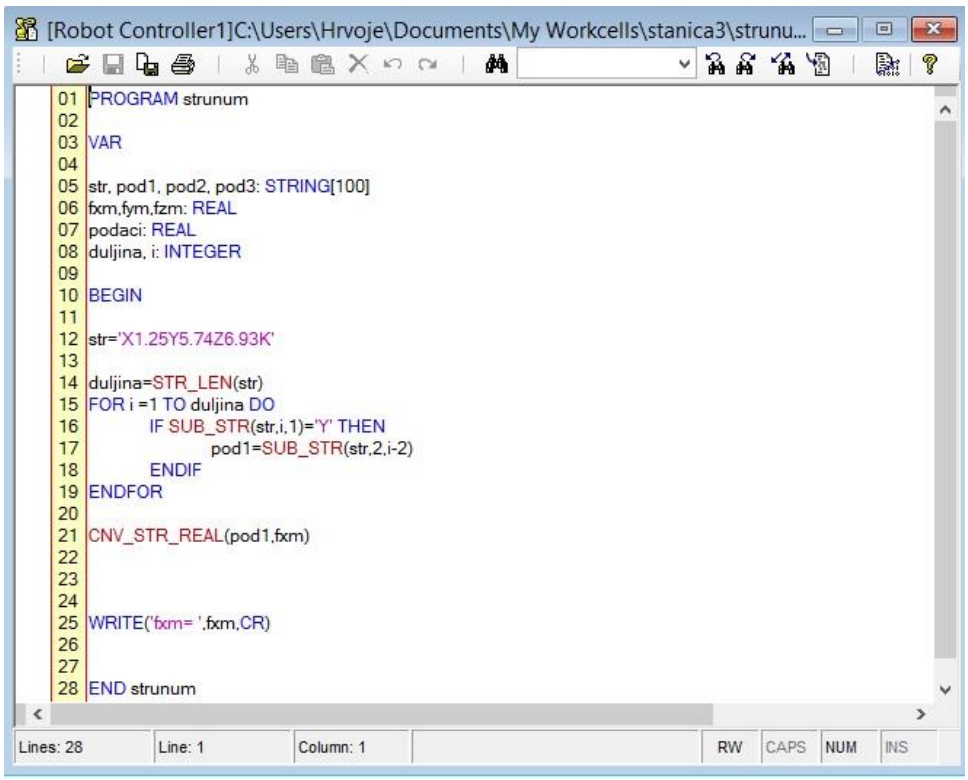
podataka koji su poslani iz *Karel*-a. Između ostalog napravljena je pretvorba *string* podataka u numeričke vrijednosti, računanje homogenih transformacija i slanje podataka natrag u *Karel* korištenjem TCP/IP protokola. *Catia* je računalni softver za 3D modeliranje i u ovom radu je korištena za modeliranje dijelova kolaborativne hvataljke.

### *4.3.1. RoboGuide*

Osim za *offline* programiranje robota, *RoboGuide* je simulacijski softver u kojem se može napraviti, programirati i simulirati cijela robotska stanica u 3D okruženju, bez potrebe fizičkih prototipova i troškova njihova postavljanja. *RoboGuide* omogućuje vizualizaciju jedne ili više robotskih stanica čime je rizik od krivog postavljanja robota u stvarnom svijetu manji [12].

### *4.3.2. KAREL*

*Karel* je programski jezik, jako sličan *Pascalu*. Napisani program treba prevesti u strojni jezik i poslati u upravljačku jedinicu robota. Program se može simulirati i na virtualnom robotu u *RoboGuide*-u. Na slici 16. prikazan je primjer pisanja *Karel* programa.




**Slika 16. Primjer *Karel* programa**

#### *4.3.2.1. Struktura programa*

Novi program u *Karel*-u započinje ključnom riječi **PROGRAM** i iza nje je potrebno napisati ime programa, koje je kraće od osam znakova. Dalje je potrebno definirati varijable (ključna riječ **VAR**) i konstante (ključna riječ **CONST**) koje će se koristiti u programu. Konstante mogu biti samo cjelobrojni brojevi. Nakon definiranja varijabli i konstanti, početak glavnog programa započinje naredbom **BEGIN** i ispod nje kreće glavni dio programa u kojem se izvršavaju zadane naredbe, pozivaju potprogrami, upravlja se robotom, pozivaju rutine, izvode petlje… Ključna riječ **END** i ime programa iza nje označavaju kraj programa. Komentari se pišu tako da se ispred komentara stave dvije crtice: -- i označeni su zelenom bojom.

Sintaksa programa:

```
PROGRAM ime_programa          --početak programa
VAR                           --definiranje varijabli
      cijeli_broj: INTEGER    --varijabla cijeli_broj je INTEGER tip
      decimalni_broj: REAL    --varijabla decimalni_broj je REAL tip
      rijec: STRING
      uvjet: BOOLEAN
      tocka: XYZWPR
CONST                         --definiranje konstanti
      konstanta1: 10          --konstanta1 poprima vrijednost 10
BEGIN                         --početak glavnog programa
      cijeli_broj=10
      FOR …
      …
END ime_programa              --kraj programa
```

#### *4.3.2.2. Tipovi varijabli*

Kao i ostali računalni programski jezici tako i *Karel* ima više tipova varijabli. Mnogi tipovi varijabli su slični ostalim programskim jezicima (INTEGER, REAL, BOOLEAN), no u *Karel*-u postoje varijable specifične za rad s robotima (XYZWPR, CONFIG). Najčešći tipovi varijabli su [8]:

- INTEGER - može poprimiti vrijednost cjelobrojnih brojeva (od -2147483648 do 2147483646)
- REAL - poprima vrijednost decimalnih brojeva od $-3{,}4028236 \cdot 10^{38}$ do $3{,}4028236 \cdot 10^{38}$
- STRING - može biti kombinacija slova, brojeva, i posebnih znakova
- BOOLEAN - može imati vrijednost TRUE (istina, 0) ili FALSE (laž, sve osim 0)
- XYZWPR - pozicijska varijabla, ima šest decimalnih vrijednosti, tri translacije i tri rotacije
- CONFIG - varijabla vezana uz konfiguraciju robota

Postoje još i sistemske varijable (eng. *system variable*) koje su preddefinirane u upravljačkoj jedinici robota, a u *Karel*-u je omogućen rad s većinom sistemskih varijabli. Sistemske varijable započinju znakom $ te su napisane velikim slovima bez razmaka. Neke od sistemskih varijabli [8]:

- $MOTYPE - određuje način gibanja od početne do krajnje točke, a može poprimiti vrijednosti LINEAR (linearno gibanje po pravcu), CIRCULAR (kružno gibanje) i JOINT (gibanje interpolacijom zglobova)
- $TERMTYPE - određuje način izvršavanja gibanja, odnosno vremena i brzinu ubrzanja i usporenja (FINE, VARDECEL, NODECEL…)
- $SPEED - određuje brzinu izvršavanja pojedinog gibanja (mm/s)
- $UFRAME - koristi se za definiranje pozicija i konfiguracija korisničkog koordinatnog sustava

- $UTOOL - koristi se za definiranje pozicija i konfiguracije koordinatnog sustava alata

#### *4.3.2.3. Uvjetne naredbe [8]*

Najčešća uvjetna naredba je IF…THEN…ELSE koja izvršava blok naredbi ako je uvjet zadovoljen. Ima dva oblika, a sintaksa prvog oblika je:

```
IF uvjet THEN
    Blok naredbi
ENDIF
```

Sintaksa drugog oblika je:

```
IF uvjet THEN
    Prvi blok naredbi
ELSE
    Drugi blok naredbi
ENDIF
```

Druga uvjetna naredba je SELECT naredba koja ima sljedeću sintaksu:

```
SELECT ulazna_var OF
    CASE (1):
        --blok naredbi koje se izvršavaju ako je ulazna_var = 1
    CASE (2):
        --blok naredbi koje se izvršavaju ako je ulazna_var = 2
    CASE (3):
        --blok naredbi koje se izvršavaju ako je ulazna_var = 3
```

#### *4.3.2.4. Petlje [8]*

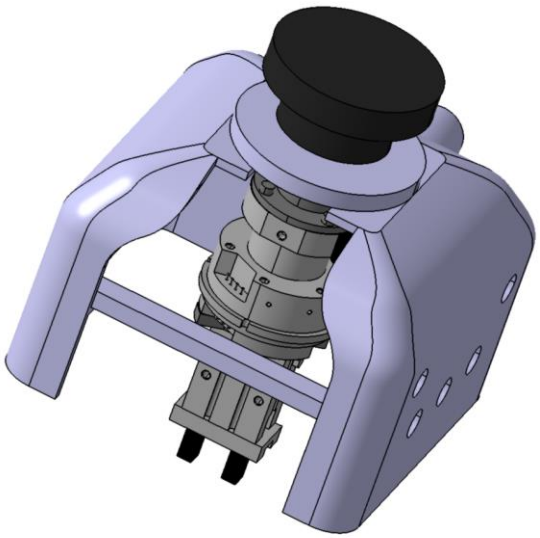
- FOR petlja - izvršava blok naredbi zadani broj puta
- WHILE petlja - izvršava blok naredbi sve dok je uvjet zadovoljen, a provjera uvjeta nalazi se na početku petlje
- REPEAT petlja - izvršava blok naredbi sve dok je uvjet zadovoljen, provjera uvjeta nalazi se na kraju petlje

# 5. OBLIKOVANJE KOLABORATIVNE HVATALJKE

Model kolaborativne hvataljke je napravljen u softveru Catia. Napravljeno je nekoliko verzija dok se nije došlo do najbolje i završne verzije. Prototip modela je napravljen 3D printanjem. Jedan dio je isprintan u Laboratoriju za projektiranje izradbenih i montažnih sustava na 3D printeru kojeg je kolega Mario Klasić napravio za svoj diplomski rad, a drugi dio u SLS tehnologiji.

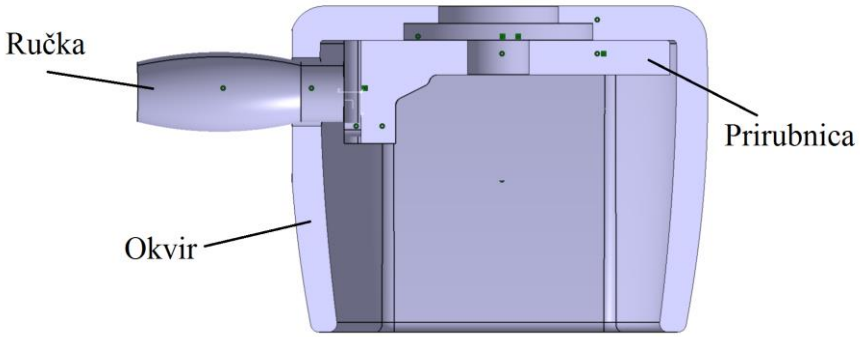
## 5.1. Početne verzije kolaborativne hvataljke

U prvoj verziji, kućište kolaborativne hvataljke je zamišljeno da bude iz jednog komada, koji na sebi ima ručku i otvor na suprotnoj stranici za lakšu ugradnju. Model prve verzije kućišta kolaborativne hvataljke je prikazan na slici 17.



**Slika 17. Model prve verzije kućišta kolaborativne hvataljke**

Nedostaci ove verzije su konstrukcija iz jednog komada, preveliki neiskorišteni prostor, prerobusna konstrukcija, te otvorena jedna stranica. U drugoj verziji išlo se na podjelu kućišta na više dijelova, te njihovog međusobnog sklapanja. na slici 18. prikazana je druga verzija koja se sastoji se od prirubnice, ručke i okvira. Nedostaci tog modela su ručka koja viri i znatno smanjuje radni prostor robota zbog mogućih kolizija, predimenzionirani dijelovi i neiskorišteni prostor.



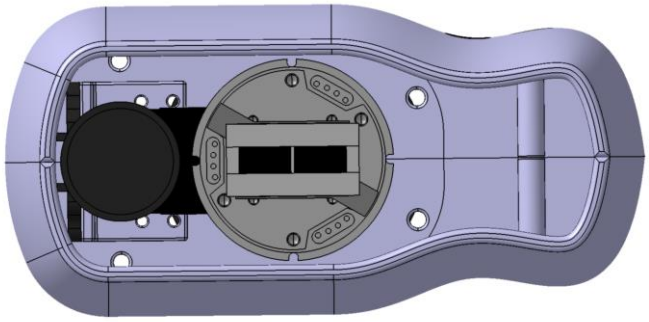

**Slika 18. Model druge verzije**

U trećoj verziji su smanjene dimenzije dijelova kućišta kolaborativne hvataljke, te je ručka napravljena zakrivljena kako bi se povećao radni prostor robota i smanjile mogućnosti kolizije s okolinom. Također su predviđena mjesta za tipkala, spajanje prirubnice i okvira vijcima i mjesto za nosače koji drže kameru. Nedostaci su i dalje mala predimenzioniranost i neiskorišteni prostor. Slika 19. prikazuje treću verziju modela kućišta kolaborativne hvataljke.

**Slika 19. Treća verzija modela**

Da bi se neiskorišteni prostor smanjio, adapterska ploča na robotu je zarotirana za 90°. Napravljen je i potpuno novi model kućišta kolaborativne hvataljke bez ručke, a okvir je modeliran tako da služi za vođenje robota po prostoru. Na slikama 20. i 21. prikazan je novi model kućišta kolaborativne hvataljke, koji je naknadno dorađivan, a u potpoglavljima ispod su malo detaljnije opisani njegovi dijelovi.

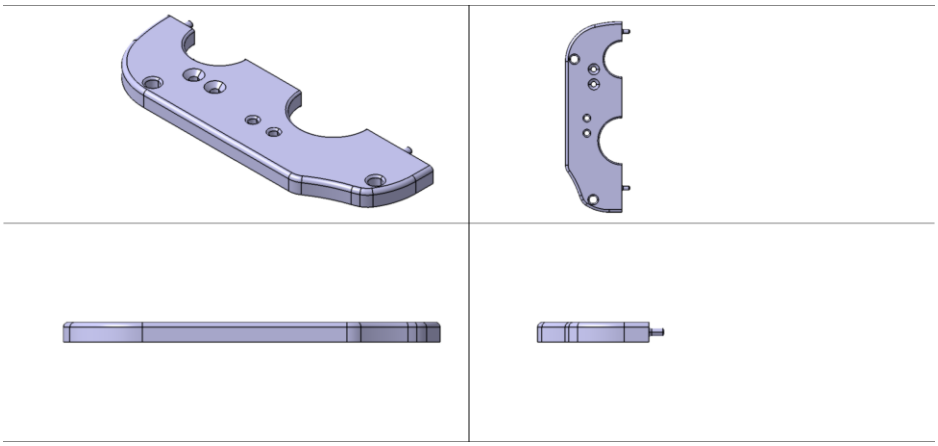
**Slika 20. Završni model kućišta kolaborativne hvataljke**



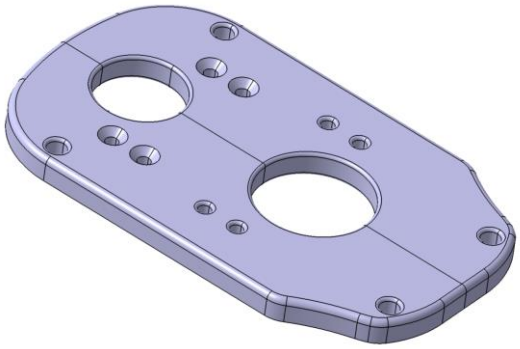
**Slika 21. Završni model kolaborativne hvataljke - pogled od ispod**

## 5.2. Prirubnica

U odnosu na početne prirubnice, u završnoj verziji nema mjesto predviđeno za ručku, smanjena je dužina, širina i debljina. Napravljena je iz dva dijela koja se međusobno spajaju značajkom oblika, odnosno napravljeni su „pinovi“ i uvrti. Jedan dio prirubnice prikazan je na slici 22., a sklopni model na slici 23. Prirubnica se vijcima pričvršćuje na adaptersku ploču robota, odmah ispod senzora sile. Na sebi ima provrte za vijke za spoj nosača koji drže pločicu na kojoj se nalazi kamera, ima provrte za vijke za učvršćenje okvira na prirubnicu, te ima veći provrt za prolaz kablova kamere i tipkala. Gabaritne mjere su: 152x90 x8 mm. Slika 24. prikazuje prirubnicu pričvršćenu na adaptersku ploču zajedno s nosačima i pločicom za kameru.



**Slika 22. Prirubnica - jedna komponenta**

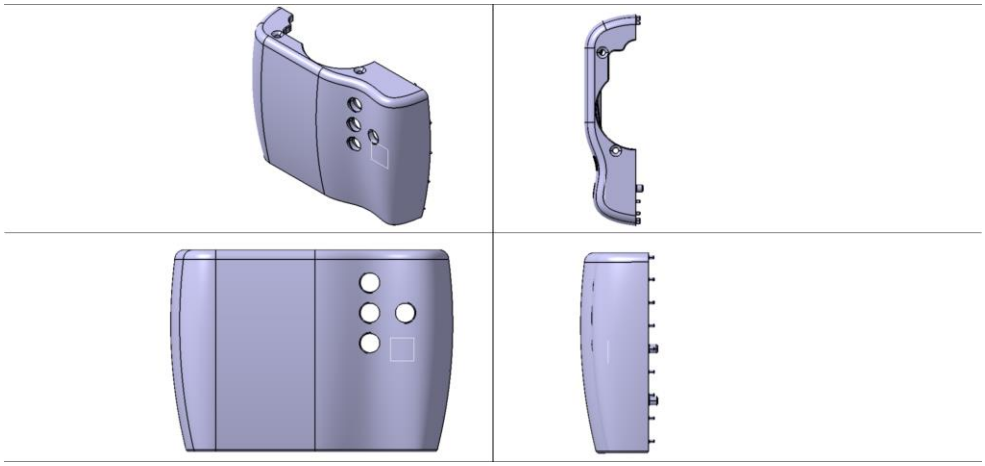


**Slika 23. Prirubnica**

**Slika 24. Prirubnica pričvršćena na adaptersku ploču robota**

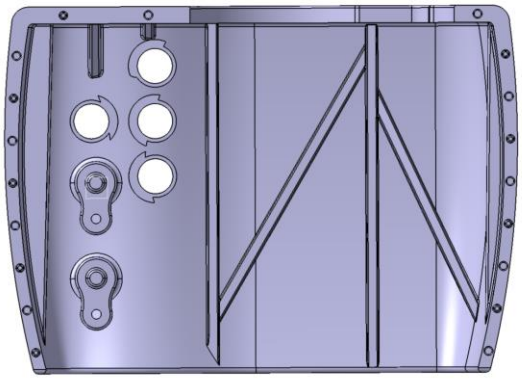
## 5.3. Ovir

Radi lakše montaže, okvir je modeliran iz dva simetrična dijela, osim što se na jednom dijelu nalaze provrti za ugradnju tipkala. Jedan dio okvira prikazan je na slici 25.

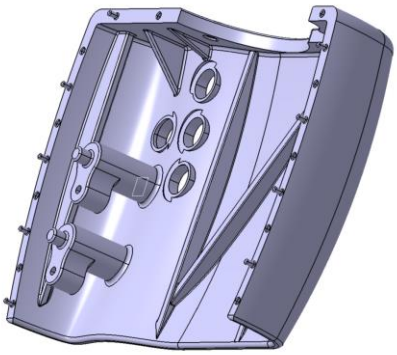


**Slika 25. Jedan dio okvira**

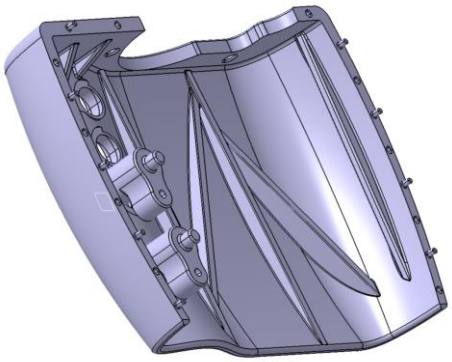
Okvir se spaja na prirubnicu s četiri vijka, a spoj dva dijela okvira je osiguran oblikom („pinovi“ i slijepe rupe duž spojnih stranica) i dodatnom značajkom s prolaznom rupom tako da se s unutarnje strane mogu postaviti metrički vijci i pričvrstiti jedan dio okvira za drugi. S unutarnje strane, na mjestu gdje se hvata okvir rukom, su napravljene dvije značajke koje osiguravaju da se okvir na savija prilikom pritiska. Također su napravljena rebra radi povećanja krutosti same konstrukcije. Slike 26., 27., i 28. prikazuju unutarnju stranu okvira.



**Slika 26. Unutarnja strana okvira**



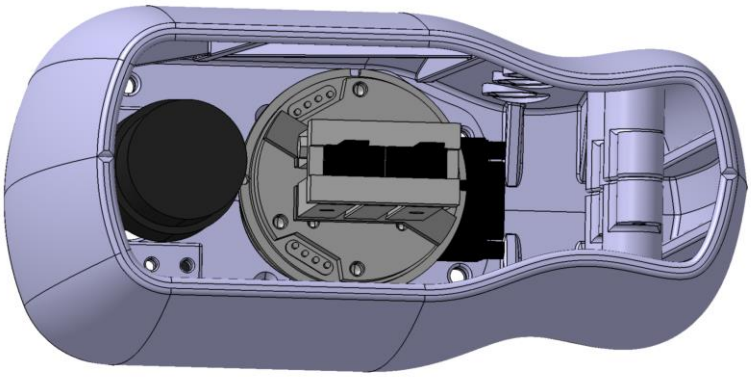
**Slika 27. Pogled na dodatne značajke za unutarnji spoj vijcima i protiv savijanja prilikom držanja rukom**



**Slika 28. Pogled na rebra radi povećanja krutosti**

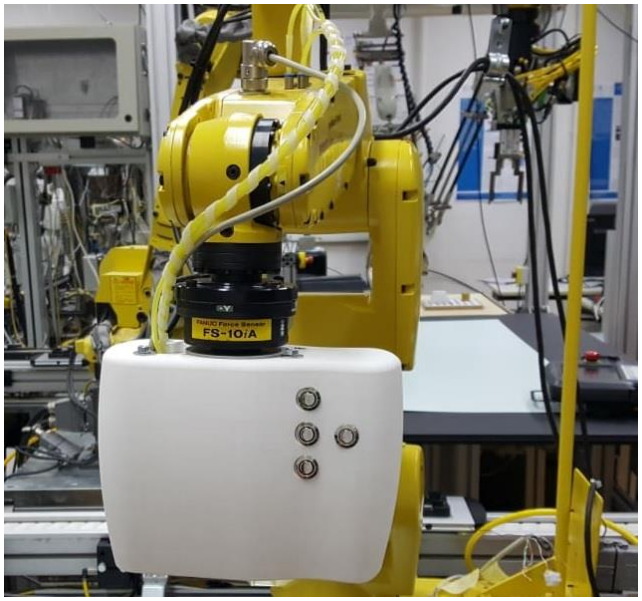
Slika 29. prikazuje spoj okvira i prirubnice, a slika 30. pogled od ispod kako bi kolaborativna hvataljka trebala izgledati kad se pričvrsti na robot.

**Slika 29. Spoj okvira i prirubnice**



**Slika 30. Pogled od ispod na kolaborativnu hvataljku**

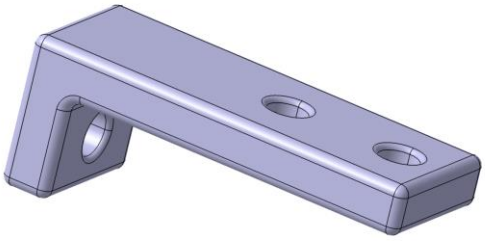
Na slici 31. prikazan je okvir kolaborativne hvataljke postavljen na robota. Na desnoj strani se nalaze tipkala, a na toj strani je i hvatište okvira za vođenje robota po prostoru.




**Slika 31. Okvir kolaborativne hvataljke**

## 5.4. Nosač i pločica za kameru

Za potrebe kamere modelirana je pločica na koju se kamera pričvršćuje i nosači koji drže tu pločicu i koji se pričvršćuju na prirubnicu. Spojevi nosača i prirubnice, te pločice za kameru i nosača ostvareni su vijcima. Gabaritne mjere nosača su: 40x15x15 mm, a model je prikazan na slici 32.



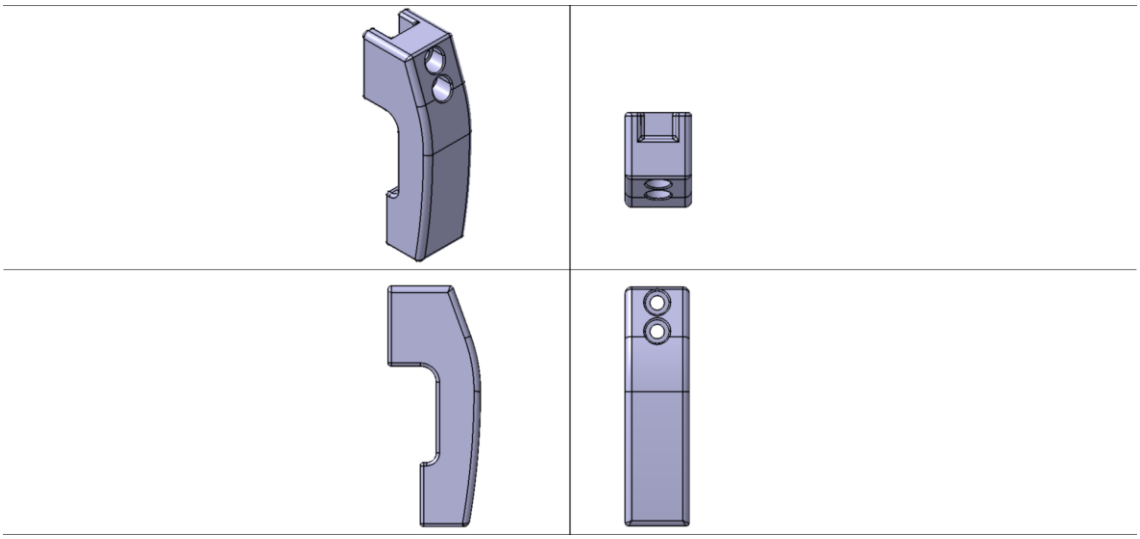
**Slika 32. Model nosača**

Model pločice koja drži kameru prikazan je na slici 33. Napravljena su rebra na jednoj stranici radi povećanja krutosti i debljina je povećana za 2 mm u odnosu na prvu verziju (prva verzija nije imala rebra). Na slici 24. se može vidjeti pločica pričvršćena na prirubnicu preko nosača.

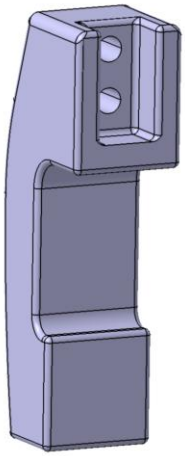
**Slika 33. Model pločice za kameru**

## 5.5. Prsti hvataljke

Za hvatanje predmeta modelirani su prsti hvataljke. Prsti se ugrađuju na model hvataljke *MHZ2-20DN-A3*, proizvođača *SMC*. Gabaritne mjere prstiju su: 75x28x20 mm. U zatvorenom položaju, udaljenost između prstiju je 27,2 mm. Na sebi imaju provrte za vijčani spoj s hvataljkom. Slike 34. i 35. prikazuju modelirane prste.



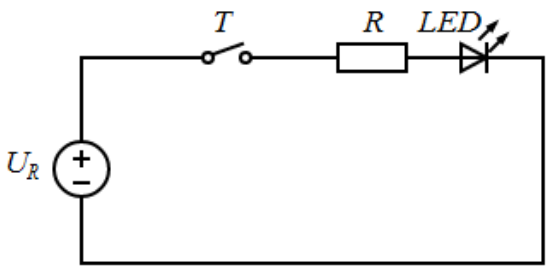
**Slika 34. Model prstiju hvataljke**



**Slika 35. Model prstiju hvataljke - drugi pogled**

# 6. UGRADNJA TIPKALA

Na kolaborativnu hvataljku su ugrađena tri radna tipkala i jedan prekidač. Razlika između radnih tipkala i prekidača je u načinu njihova aktiviranja. Radna tipkala su aktivna samo dok su pritisnuta, a nakon otpuštanja kontakt se prekida, za razliku od prekidača koji ostaju aktivni nakon otpuštanja, te se trebaju ponovno pritisnuti kako bi se kontakt prekinuo. Tipkala na sebi imaju statusne LED diode i svijetle kad su aktivirana. Eksperimentalnom provjerom je utvrđeno da je za napajanje LED dioda potreban napon $U_{LED}$ = 12 V i struja $I_{LED}$ = 0,012 A. S obzirom na to da robot daje napon $U_R$ = 24 V i struju $I_R$ = 0,5 A, potrebno je u strujni krug s tipkalima dodati otpornik $R$ koji će smanjiti napon i struju na LED diodama. Shema strujnog kruga prikazana je na slici 36.,




**Slika 36. Strujni krug**

gdje je: $U_R$ - napon koji daje robot,

$R$ - otpornik za smanjenje napona i struje na LED diodi

$LED$ - LED dioda na tipkalu

$T$ - tipkalo.

Otpor LED diode se može dobiti preko formule (1):

$$R_{LED} = \frac{U_{LED}}{I_{LED}}$$
$$R_{LED} = \frac{12}{0,012} \tag{1}$$
$$R_{LED} = 1000\ \Omega$$

Iz formule (2) za *Ohm*-ov zakon, može se doći do formule za potrebni otpornik *R* (3) i uvrštavanjem dobivenih vrijednosti izračunati vrijednost potrebnog otpornika:

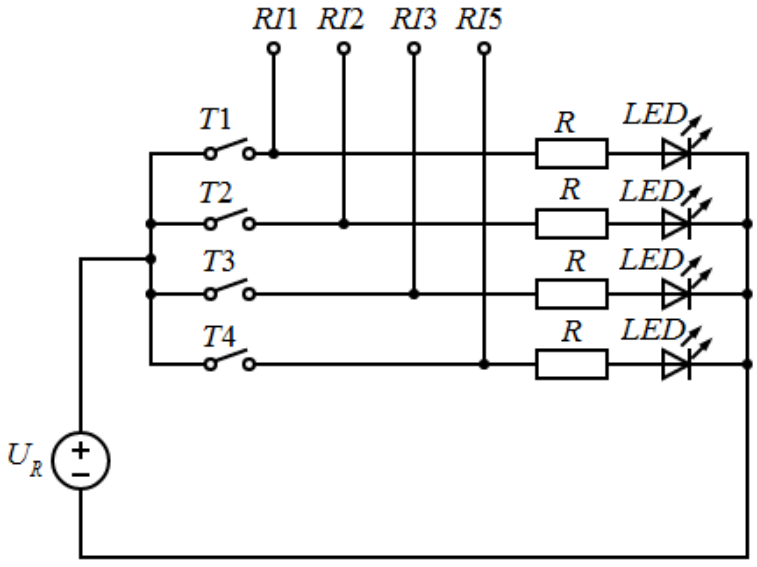
$$I_{LED} = \frac{U_R}{R + R_{LED}} \tag{2}$$

$$R = \frac{U_R}{I_{LED}} - R_{LED}$$

$$R = \frac{24}{0,012} - 1000 \tag{3}$$

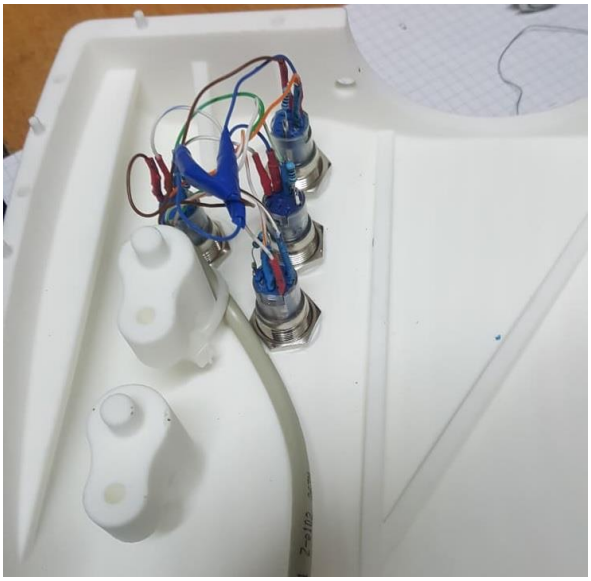
$$R = 1000\ \Omega$$

Nakon što je izračunato da je potreban otpornik od 1000 Ω, jedno tipkalo je spojeno u strujni krug s otpornikom od 1000 Ω radi provjere. LED dioda je svijetlila, no eksperimentalnim putem se došlo do zaključka da je bolje ugraditi otpornik od 510 Ω, radi boljeg osvjetljenja, a pritom se LED dioda ne grije. Nakon što je određen otpornik koji će biti ugrađen, napravljen je kabel koji spaja periferni konektor i tipkala. Kroz kabel prolazi šest žica, od toga su po jedna žica spojena na 24 V i uzemljenje, a ostale četiri na robotske digitalne ulazne signale (*RI*1, *RI*2, *RI*3, *RI*5). Detaljna shema spajanja prikazana je na slici 37.




**Slika 37. Detaljna shema spajanja**

Slika 38. prikazuje kabel zalemljen na tipkala.



**Slika 38. Tipkala spojena na kabel**

Sva četiri tipkala u početnom stanju su neaktivirana i strujni krug je otvoren. Tek pritiskom na jedno od tipkala strujni krug se zatvara, LED dioda zasvijetli i aktivira se jedan od digitalnih ulaznih signala robota. Slika 39. prikazuje provjeru spojenih tipkala s digitalnim ulaznim signalima, te provjeru statusnih LED dioda.

**Slika 39. Provjera signala i LED dioda**

# 7. RAZVOJ PROGRAMSKE PODRŠKE ZA KOLABORATIVNU HVATALJKU

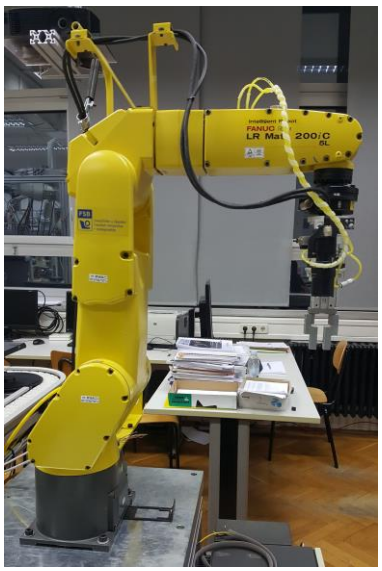
Kao što je već prije navedeno, za rješavanje ovoga zadatka koristi se *Karel* za programiranje robota *LR Mate 200iC - 5L*. Programiranje je postupno teklo, stoga postoji par verzija programa, koje će biti ukratko opisane, dok se nije došlo do najbolje verzije. Važno je još jednom napomenuti da *Karel* ne podržava rad s matricama, što ovaj zadatak čini zahtjevnijim.

Prije glavnog programiranja, potrebno je kalibrirati senzor sile jer postoje početne vrijednosti na njemu, pa da ne bi smetale u njegovom radu. Također je potrebno napraviti kompenzaciju gravitacije, odnosno odrediti točan iznos mase svih dijelova ispod senzora sile, a to se napravi tako da se robot dovede u početnu poziciju, prikazanu na slici 40., u toj se poziciji očita vrijednost sile na senzoru u z smjeru koordinatnog sustava senzora (**napomena**: z-os koordinatnog sustava senzora sile je usmjerena u smjeru gravitacije) i pohrani se u varijablu $F_{zd}$. Iznos sile $F_{zd}$ prikazan je jednadžbom (4),

$$F_{zd} = F_{z0} + F_g \tag{4}$$

gdje je: $F_{z0}$ - početna vrijednost sile na senzoru u smjeru z-osi,

$F_g$ - težina dijelova ispod senzora sile.



**Slika 40. Početna pozicija robota**

Nakon toga robot ide u drugu poziciju, prikazanu na slici 41., i očita se nova vrijednost sile u z smjeru koordinatnog sustava senzora i pohrani se u varijablu $F_{zg}$. Iznos sile $F_{zg}$ određuje jednadžba (5):

$$F_{zg} = F_{z0} - F_g \tag{5}$$

**Slika 41. Druga pozicija robota**

Rješavanjem jednostavnog sustava jednadžbi (4) i (5) dobije se jednadžba (6) kojom se odredi sila $F_g$:

$$F_g = \frac{1}{2}(F_{zd} - F_{zg}) \tag{6}$$

Kalibracija je ista za sve verzije programa.

### 7.1. Prva verzija

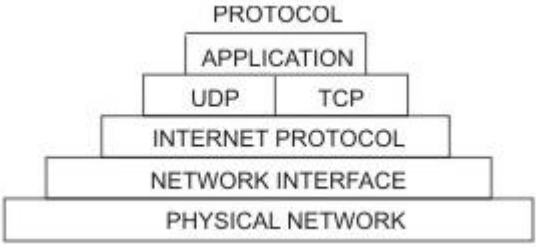
U prvoj verziji programa, nakon kalibracije, napravljene su samo translacije robota u prostoru. Robota se moglo pomicati vanjskim silama kada bi vrijednosti na senzoru sile premašili određeni iznos (eng. *treshold*). Program je dobro radio za određenu orijentaciju hvataljke, no problemi su se javili kad su dodane rotacije. Kad se robotsku hvataljku zarotira

u neku orijentaciju, različitu od početne pozicije robota, vrijednosti sile na senzoru se promijene zbog sile $F_g$, koja je i dalje u smjeru gravitacije, ali drugačije djeluje na senzor. Time se automatski premašio *treshold* i robot bi se gibao bez utjecaja vanjskih sila. To je otklonjeno množenjem kuteva orijentacije sa sinus i kosinus funkcijama, ali je radilo samo za male vrijednosti rotacija. Pri većim rotacijama, funkcije sinus i kosinus nisu mogle pratiti promjenu početnih sila na senzoru i robot bi se opet gibao bez utjecaja vanjskih sila.

Rješenje navedenog problema je u računanju homogenih transformacija u *Matlab*-u, a da bi to bilo ostvarivo potrebno je napraviti dvostranu TCP/IP komunikaciju između *Karel*-a i *Matlab*-a.

## 7.2. Komunikacija Karel - Matlab

Za slanje podataka iz *Karel*-a u *Matlab* i obratno koristi se TCP/IP komunikacija, koja je prisutna na svim računalima. Na slici 42. je prikazana struktura TCP/IP protokola. Informacija koja se šalje na udaljeno računalo prolazi kroz servis (eng. *Application*) koji podatke od te informacije prebacuje na TCP sloj (eng. *Transmission Control Protocol*). TCP sloj te podatke podijeli na manje pakete i svaki taj paket koristi najbrži put na mreži do svog odredišta. Da bi se osiguralo sastavljanje tih paketa pravilnim redoslijedom, TCP svakom paketu dodijeli opis kojim redoslijedom su se podaci podijelili. Nakon što su se podaci podijelili na pakete, paketi idu na IP sloj (eng. *Internet Protocol*), koji svakom paketu dodjeljuje podrijetlo i IP adresu na koju mora biti poslan. Nakon toga se paketi šalju preko mreže (eng. *Network Interface*) na zadano odredište (eng. *Physical Network*) gdje se ponovno spajaju u podatke istim redoslijedom. S obzirom na da se svi paketi na mreži šalju tako da stigne u što kraćem vremenu, TCP/IP komunikacija je jako brza, odnosno vrijeme između slanja i primanja podataka je jako kratko [13].




**Slika 42. Struktura TCP/IP protokola [13]**

## 7.3. Homogene transformacije

Transformacija koordinatnih sustava je osnovni koncept pri modeliranju i programiranju robota. Način na koji se to radi su matrice homogenih transformacija. Jednadžbom (7) prikazana je matrica ${}^{m}\mathbf{A}_{n}$ homogenih transformacija koja opisuje položaj i orijentaciju sustava *n* prema sustavu *m*:

$$ {}^{m}\mathbf{A}_{n} = \begin{bmatrix} i_x & j_x & k_x & p_x \\ i_y & j_y & k_y & p_y \\ i_z & j_z & k_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^{m}\mathbf{R}_{n} & {}^{m}\mathbf{p}_{n} \\ 0 & 1 \end{bmatrix} \tag{7} $$

gdje su: $i_{\dots}$ - orijentacija jediničnog vektora *x*-osi *n* sustava prikazan u *m* sustavu,

$j_{\dots}$ - orijentacija jediničnog vektora *y*-osi *n* sustava prikazan u *m* sustavu,

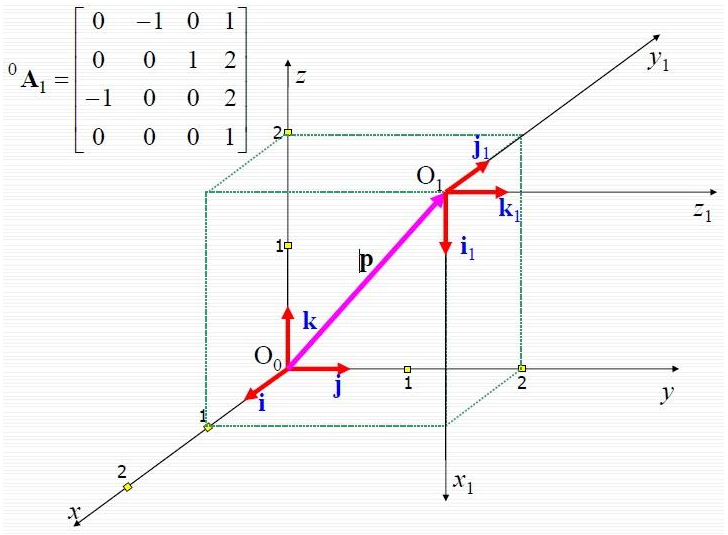
$k_{\dots}$ - orijentacija jediničnog vektora *z*-osi *n* sustava prikazan u *m* sustavu,

$p_{\dots}$ - translacija koordinate … *n* sustava u odnosu na *m* sustav,

${}^{m}\mathbf{R}_{n}$ - matrica rotacije sustava *n* prema sustavu *m* i

${}^{m}\mathbf{p}_{n}$ - matrica translacije sustava *n* prema sustavu *m* [14].

Na slici 43. prikazan je jednostavan primjer homogenih transformacija sustava 1 prema sustavu 0:




**Slika 43. Primjer homogenih transformacija [14]**

Iz primjera se vidi da je $x_1$ os usmjerena u suprotnom smjeru osi *z*, stoga u matrici homogenih transformacija na mjestu $i_z$ stoji vrijednost -1, isto tako $y_1$ os je usmjerena u suprotnom smjeru *x* osi, pa je na mjestu $j_x$ vrijednost -1, a $z_1$ os gleda u pozitivnom smjeru osi *y*, pa je na mjestu $k_y$ vrijednost 1. Sustav 1 je i translatiran u odnosu na sustav 0, te na mjestu $p_x$ stoji 1 jer je translatiran po osi *x* za 1, na mjestu $p_y$ stoji 2 i na mjestu $p_z$ stoji vrijednost 2.

U ovom radu korištene su samo rotacije, i to oko sve tri osi. Jednadžbe (8), (9) i (10) prikazuju oblik tih matrica. Matrica $\mathbf{T}_x$ označava rotacija oko *x* osi, matrica $\mathbf{T}_y$ oko *y* osi te matrica $\mathbf{T}_z$ oko *z* osi.

$$\mathbf{T}_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(w) & -\sin(w) & 0 \\ 0 & \sin(w) & \cos(w) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$\mathbf{T}_y = \begin{bmatrix} \cos(p) & 0 & \sin(p) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(p) & 0 & \cos(p) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

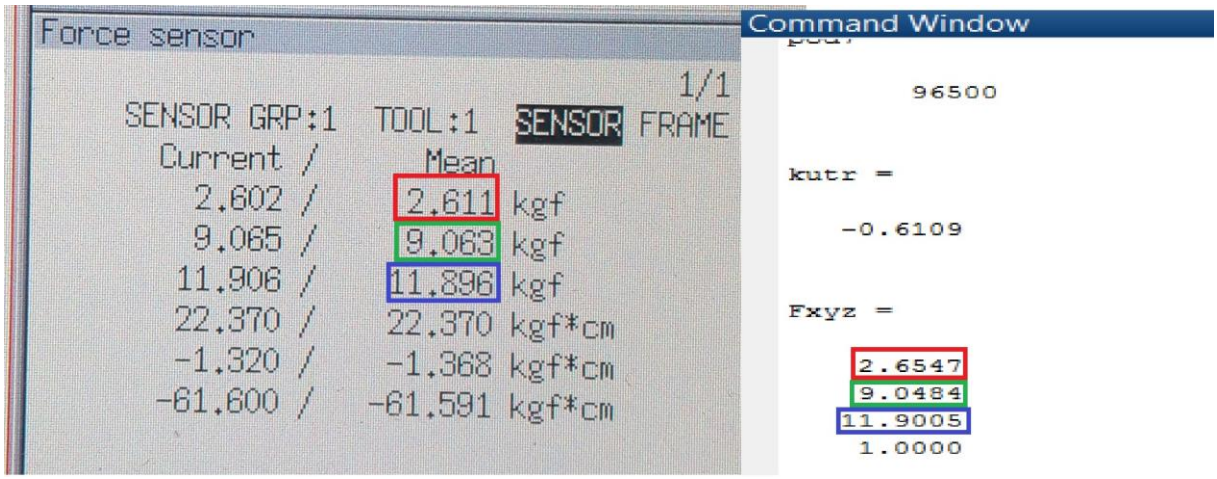
$$\mathbf{T}_z = \begin{bmatrix} \cos(r) & -\sin(r) & 0 & 0 \\ \sin(r) & \cos(r) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Množenjem jednadžbi (8), (9) i (10) dobije se ukupna matrica transformacija **A** korištena pri transformiranju vrijednosti sila iz koordinatnog sustava senzora sile u bazni koordinatni sustav robota, prikazana u jednadžbi (11):

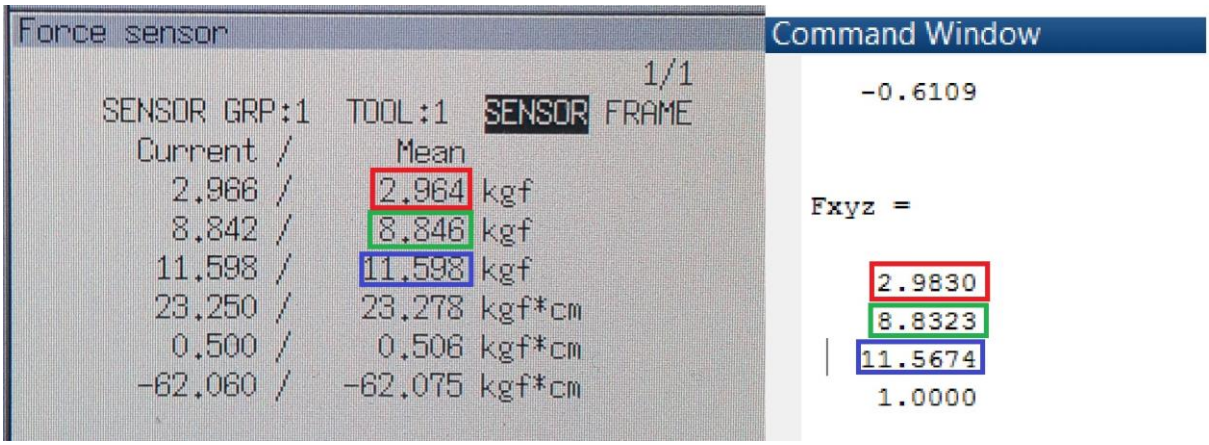
$$\mathbf{A} = \mathbf{T}_x \cdot \mathbf{T}_y \cdot \mathbf{T}_z \tag{11}$$

## 7.4. Druga verzija

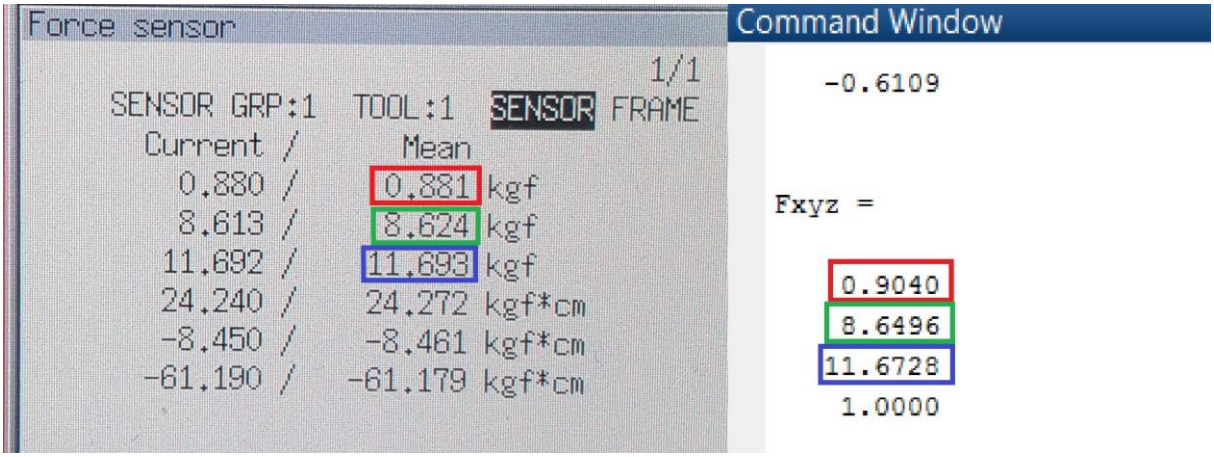
U drugoj verziji je uspostavljena TCP/IP komunikacija između *Karel*-a i *Matlab*-a u svrhu poboljšanja programa iz prve verzije. Prije samog programiranja napravljena je provjera homogenih transformacija. Vrijednosti sila poslane su u *Matlab* kad je robot bio u početnoj poziciji, zatim se robota dovelo u određenu orijentaciju i usporedile su se vrijednosti koje je davao senzor sile na upravljačkoj konzoli i vrijednosti koje su dobivene u *Matlab*-u. Na slikama 44., 45. i 46. prikazana je njihova usporedba u različitim položajima robota. Na slikama su vrijednosti koje se uspoređuju označene istim bojama.




**Slika 44. Prva orijentacija - usporedba**




**Slika 45. Druga orijentacija - usporedba**




**Slika 46. Treća orijentacija - usporedba**

Iz slika se vidi da je točnost računanja na drugoj decimali, što je za početak dobro, a odstupanje se događa jer vrijednosti na senzoru sile variraju i time smanjuju točnost.

Nakon toga kreće programiranje. *Karel* očitava sile na senzoru i orijentaciju prirubnice (kuteve *w*, *p* i *r*), izvršava se pretvorba tih varijabli u string preko ugrađene funkcije u *Karel*-u CNV_REAL_STR (konverzija REAL varijable u STRING varijablu), te se stringovi spajaju u jedan veći string, odvojeni znakom '@' radi lakšeg parsiranja, i šalju u *Matlab*.

*Matlab*-ov program se mora pokrenuti prije pokretanja *Karel* programa. Spajanje s *Karel*-om se vrši preko već napravljenih funkcija u *Matlab*-u, koje su uzete iz repozitorija gotovih programa iz Laboratorija za projektiranje izradbenih i montažnih sustava. Nakon uspješnog spajanja, u glavnom programu se izvodi parsiranje pristigloga stringa (sastavljen od sila i kuteva), te pretvorba stringa u decimalne brojeve funkcijom *str2num*. Preko matrice homogenih transformacija, kontinuirano se računaju sile za svaku orijentaciju robotske hvataljke, te se izračunate sile vraćaju iz *Matlab*-a natrag u *Karel* kao string varijabla. Zatim se u *Karel*-u vrši parsiranje pristigloga stringa, te pretvorba u REAL varijablu funkcijom CNV_STR_REAL. Za pomicanje robota, uvjet je da sile poslane iz *Matlab*-a moraju biti veće ili manje od početnih sila na senzoru

Nedostatak ove verzije je u vremenu potrebnom za izvršavanje programa. Vrijeme je jako dugo i zbog toga vođenje robota po prostoru nije kontinuirano, već je isprekidano.

Rješenje ovog problema je da se *Karel* program podijeli na dva manja, jedan za gibanje robota, a drugi za komunikaciju s *Matlab*-om.

### 7.5. Treća verzija

U trećoj verziji *Karel* program je podijeljen na dva manja, koji se zovu *matlab_giba* i *matlab_kom*. U *matlab_kom* se odvija komunikacija s *Matlab*-om, a u *matlab_giba* vođenje robota po prostoru. Oba programa komuniciraju međusobno tako da šalju vrijednosti varijabli jedan drugome. Da bi oba programa bila pokrenuta u isto vrijeme, potrebno je na upravljačkoj konzoli (eng. *Teach Pendant*) napraviti novi program i u taj program dodati programe *matlab_giba* i *matlab_kom* preko naredbe RUN. Naredba RUN osigurava pokretanje oba programa neovisno jedan o drugome, odnosno, izvršavanje nije sekvencijalno nego oba programa se izvršavaju istodobno.

Vrijeme izvršavanja programa se skratilo, vođenje robota je bolje, ali još uvijek nije zadovoljavajuće, a pojavili su se i problemi kod vraćanja velikog broja podataka iz *Matlab*-a u *Karel* (šest podataka) i njihove obrade u *Karel*-u.

Rješenje je da se iz *Matlaba* vraća manji broj podataka, uredi program radi još bržeg izvršavanja i skrati vrijeme čekanja.

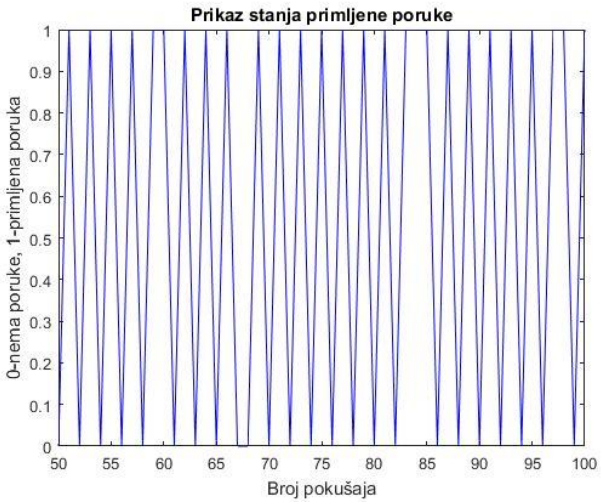
### 7.6. Završna (četvrta) verzija

Četvrta verzija ujedno je završna verzija i bit će detaljno opisana. Potrebno je prvo pokrenuti *Matlab*-ov program, koji vrti funkcije za povezivanje s *Karel*-om sve dok se ne ostvari TCP/IP komunikacija. Nakon što se pokrenuo *Matlab*-ov program, potrebno je pokrenuti *Karel* program na upravljačkoj konzoli pod nazivom MATLAB_RUN koji pokreće dva programa *matlab_giba* i *matlab_kom* preko naredbe RUN.

Oba se programa krenu izvršavati istodobno, no *matlab_kom* čeka na početku da se ostvari uvjet_prvi iz programa *matlab_giba*. Za to vrijeme *matlab_giba* šalje robota u početnu poziciju, sprema početne vrijednosti sila sa senzora u varijable, te izvršava kalibraciju na način koji je opisan na početku ovog poglavlja. Nakon napravljene kalibracije, uvjet_prvi se ostvario, program *matlab_giba* se zaustavlja i čeka uvjet_drugi iz programa *matlab_kom* da se ostvari. Nakon što se uvjet_prvi ostvario, program *matlab_kom* je nastavio svoje izvršavanje. Dok *matlab_giba* stoji, *matlab_kom* iz njega uzima vrijednosti početnih sila sa senzora i sprema ih u svoje varijable, te se povezuje s *Matlab*-om preko TCP/IP komunikacije. Nakon uspješnog povezivanja ostvaruje se uvjet_drugi, te *matlab_giba* nastavlja svoje izvršavanje.

U tom trenutku oba su programa u WHILE petljama i one se vrte sve dok se robota uči, odnosno dok mu se ne zada da izvrši naučenu operaciju. Program *matlab_kom*, dok je u WHILE petlji, sprema u varijable vrijednosti trenutnih sila na senzoru, prebacuje vrijednosti početnih sila, trenutnih sila i kuteva *w*, *p* i *r* u string varijable i sve te stringove povezuje u jednu string varijablu te ih šalje u *Matlab* na obradu.

U *Matlab*-u se vrši čitanje i spremanje dolazne poruke u string varijablu, radi se parsiranje jednog stringa na više njih i njihova konverzija u decimalne brojeve.




**Slika 47. Prikaz stanja primljene poruke**

Na slici 47. nalazi se graf koji pokazuje je li poruka došla u *Matlab* iz *Karel*-a ili nije. *X*-os predstavlja broj pokušaja čitanja poruke, a *y*-os stanje. Ako je stanje jednako 0 znači da

poruka još nije došla, a ako je stanje 1, poruka je stigla. Prikazano je 50 pokušaja koji se dogode u jednoj sekundi. Prikazani su pokušaji između pedesetog i stotog jer se prije toga mrežna veza ne uspostavi do kraja pa su svi pokušaji u stanju 0.

**Napomena:** početne sile uvijek su istog iznosa jer su očitane prije WHILE petlji i njihove vrijednosti su u koordinatnom sustavu senzora. Trenutne sile su promjenjivog iznosa, ovisno je li na senzor primijenjena vanjska sila (ručna) i očitane su za trenutnu orijentaciju hvataljke robota.

Stoga, nakon konverzije stringova u decimalne brojeve, početne sile idu kroz matricu homogenih transformacija da bi njihove vrijednosti bile prikazane u baznom koordinatnom sustavu robota. Trenutne sile idu kroz kompleksniju matricu homogenih transformacija, jer se njihove vrijednosti prvo prebacuju u početni koordinatni sustav senzora, a nakon toga u bazni koordinatni sustav robota. Nakon što su i početne i trenutne sile prikazane u baznom koordinatnom sustavu robota, one se oduzimaju, vrijednosti njihovih razlika se konvertiraju u string varijable i spajaju u jedan string te se vraćaju natrag preko TCP/IP komunikacije u *Karel* program *matlab_kom*.

U programu *matlab_kom* se dolazni string parsira na manje i prebacuje stringove u REAL varijable (decimalne brojeve). Program *matlab_giba* dohvaća te varijable iz programa *matlab_kom* te na temelju te razlike se vrši vođenje robota po prostoru. Ako nema primijenjene vanjske sile na senzor, razlika početnih i trenutnih sila će biti jednaka nuli te se robot neće pomicati. U suprotnom, ako je primijenjena vanjska sila, robot izvodi gibanje u smjeru vanjske sile. Ovisno o jačini vanjske sile, odnosno ako je razlika početnih i vanjskih sila velika, robot će se brže gibati i imat će veći korak gibanja i obratno. Pri jako velikoj primijenjenoj vanjskoj sili omogućavaju se rotacije hvataljke.

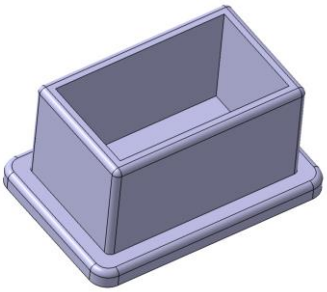
Prilikom razvijanja programske podrške robota se učilo preko tipki na upravljačkoj konzoli. Pritisnuta tipka 7 omogućuje vođenje robota po prostoru, a kad je otpuštena robot je u zakočenom stanju. Tipka 8 pamti točke i sprema ih u pozicijske registre, a tipke 4 i 5 služe za otvaranje i zatvaranje hvataljke. Nakon što se robota naučilo zadanu operaciju, za njezino izvršavanje potrebno je stisnuti tipku 9. Nakon izvršene operacije program se zaustavlja.

Nakon što je razvijena programska podrška, *Karel* program *matlab_giba* je dorađen tako da se umjesto tipki na upravljačkoj konzoli robota uči tipkalima koja su ugrađena na kolaborativnu hvataljku. Pritiskom na prekidačko tipkalo, bijela LED dioda zasvijetli te je robot u popustljivom stanju, odnosno robota se može voditi po prostoru. Prilikom ponovnog pritiska na njega robot ostaje zakočen u trenutnom položaju i onemogućeno je bilo kakvo gibanje. Da bi se mogle pamtiti točke, otvarati i zatvarati hvataljka potrebno je da prekidač bude aktiviran. Pritiskom na radno tipkalo sa zelenom LED diodom, sprema se pozicija robota u pozicijski registar, radno tipkalo s plavom LED diodom služi za otvaranje hvataljke, a s crvenom LED diodom za zatvaranje hvataljke. Za izvršavanje naučene operacije, potrebno je deaktivirati prekidač (bijela LED dioda ne smije svijetliti), te istovremeno pritisnuti radno tipkalo sa zelenom i s plavom LED diodom. Nakon izvršavanja operacije program ulazi u novu petlju koja se vrti sve dok se ne zada nova naredba. Postoje tri mogućnosti za daljnji rad. Prva je da robot ponovno izvrši već naučenu operaciju i to se ostvaruje istovremenim pritiskom na tipkala sa zelenom i plavom LED diodom. Druga mogućnost je da se robota uči nova operacija, a to se postiže istovremenim pritiskom na tipkala s plavom i crvenom LED diodom. Treća mogućnost je da se prekine izvođenje programa, a to se postiže kad je prekidačko tipkalo aktivirano (bijela LED dioda svijetli) i pritiskom tipkala s crvenom LED diodom.

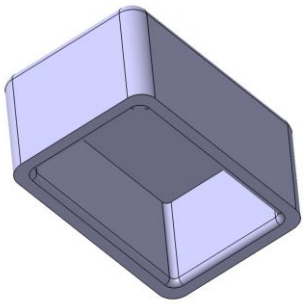
Prilikom izvođenja naučene operacije, robot prolazi kroz zapamćene točke. U točkama gdje se mijenja stanje hvataljke (otvaranje ili zatvaranje) postoji čekanje od 250 ms da se osigura sigurno hvatanje predmeta. Ako u procesu izvođenja operacije, robot naiđe na prepreku, vrijednosti sila na senzoru se povećavaju te se robot zaustavlja u sljedećoj točki. Nakon zaustavljanja, robot se više ne može pokrenuti, pa je potrebno program ponovno pokrenuti.

# 8. PRIMJENA KOLABORATIVNE HVATALJKE

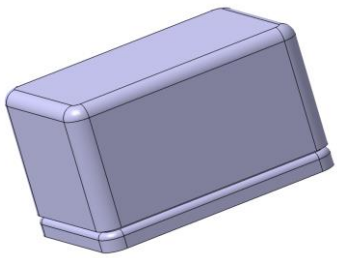
Nakon što je napravljena programska podrška za kolaborativnu hvataljku, potrebno ju je testirati i primijeniti u realnom okruženju. Za primjenu kolaborativne hvataljke napravljen je jednostavan model kutije i poklopca, te je potrebno izvršiti njihovo sklapanje/rastavljanje. Takav model je izabran radi lakšeg uočavanja mogućih nedostataka u programskoj podršci za kolaborativnu hvataljku, te u cilju što bržeg otklanjanja nedostataka programske podrške. Slike 48., 49. i 50. prikazuju redom kutiju, poklopac i njihov sklop.



**Slika 48. Model kutije**



**Slika 49. Model poklopca**



**Slika 50. Sklopni model kutije i poklopca**

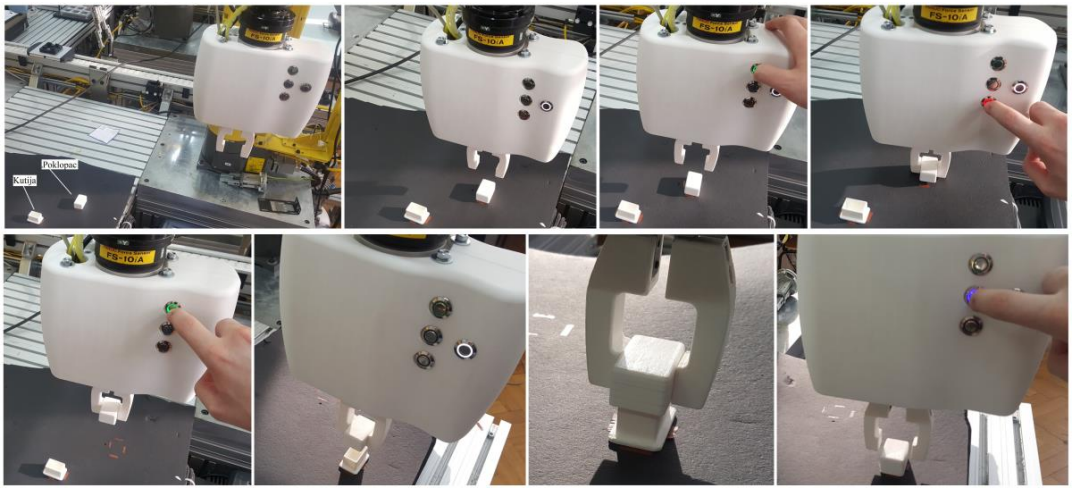
Slika 51. prikazuje isprintane 3D modele koji su prethodno modelirani. Učenje robota se odvija tako da se hvataljku, koja je prikazana na slici 52., vodi do poklopca, prsti hvataljke se zatvore i primaju poklopac, zatim se vodi do kutije te se na njoj prsti otvaraju i poklopac se stavlja na kutiju. Prilikom učenja je potrebno kontinuirano pamtiti točke putanje robota.

**Slika 51. Isprintani 3D modeli kutije i poklopca**

**Slika 52. Kolaborativna hvataljka s ugrađenim prstima**

Nakon što su modeli isprintani može se provjeriti funkcionalnost kolaborativne hvataljke. Na slici 53. je prikazan proces učenja robota. Kutija i poklopac su rastavljeni, te dok je tipkalo s bijelom LED diodom uključeno, kolaborativnu hvataljku se vodi u prostoru, pritiskom na tipkalo sa zelenom LED diodom se pamti točka, a pritiskom na crveno se prsti zatvaraju i

primaju poklopac. Zatim se kolaborativna hvataljka vodi do kutije, pamteći točke u prostoru. Iznad kutije se pritisne tipkalo s plavom LED diodom i poklopac se ispusti na kutiju.

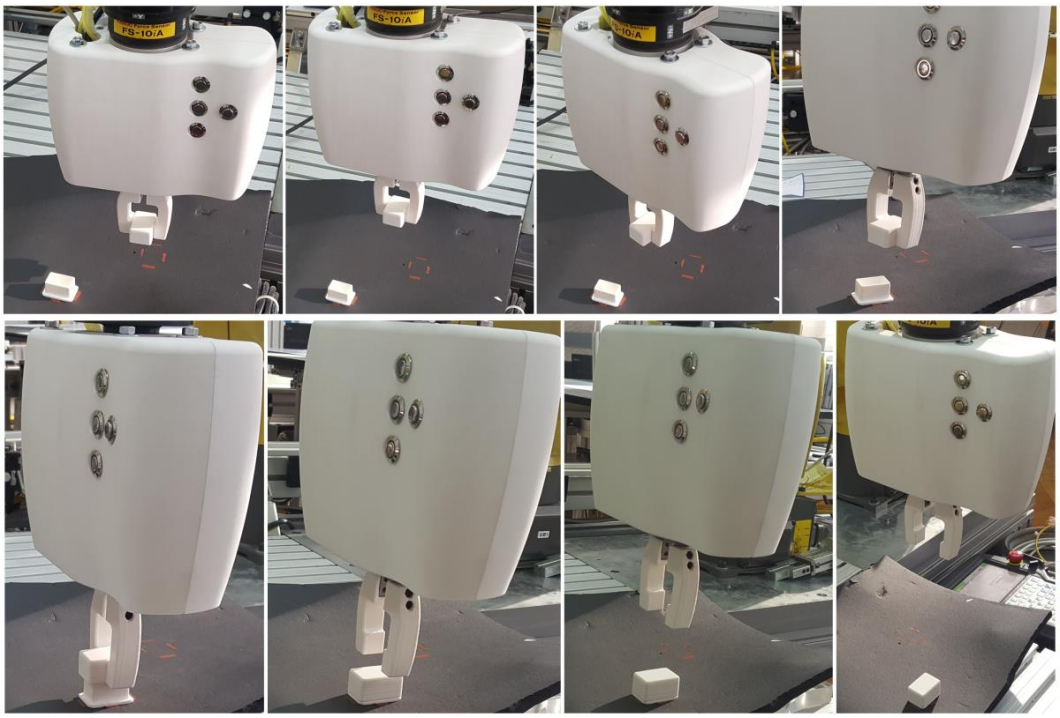



**Slika 53. Prikaz učenja robota**

Slika 54. prikazuje prelazak iz stanja učenja robota u izvršavanje naučene operacije. Da bi se to ostvarilo potrebno je tipkalo s bijelom LED diodom isključiti, te istovremeno pritisnuti zelena i plava tipkala. Zatim robot čeka dvije sekunde iz sigurnosnih razloga i kreće s ponavljanjem naučene operacije.

**Slika 54. Prelazak u izvršavanje naučene operacije**

Slika 55. prikazuje nastavak izvršavanja naučene operacije. Nakon uspješnog ponavljanja robot čeka novi zadatak. Ponovnim pritiskom na zeleno i plavo tipkalo, robot će ponoviti naučenu operaciju. No, ako je potrebno naučiti robota novu operaciju, to se može ostvariti istovremenim pritiskom na plavo i crveno tipkalo. Za potpuno prekidanje programa potrebno je upaliti tipkalo s bijelom LED diodom i pritisnuti crveno tipkalo.



**Slika 55. Izvršavanje naučene operacije**

# 9. ZAKLJUČAK

Tema ovog diplomskog rada bila je izrada kolaborativne hvataljke. Njihova upotreba u industriji je sve veća, pogotovo jer je s njima lako i sigurno rukovoditi. U početnim poglavljima ovog rada opisane su različite kolaborativne hvataljke i njihova primjena, pokazane su njihove funkcije i izvedbe, neke su toliko interaktivne da se upravljaju preko zaslona osjetljivog na dodir. Opisane kolaborativne hvataljke su bile inspiracija u razvijanju vlastitog modela koji će se ugraditi na robota *LR Mate 200iC - 5L* u Laboratoriju za projektiranje izradbenih i montažnih sustava na Fakultetu strojarstva i brodogradnje. Uz konzultacije s mentorima, došlo se do modela koji će biti izrađen i ugrađen na robota, a taj model je opisan u petom poglavlju. Dijelovi modela su izrađeni tehnologijom 3D printanja. Za model je bilo potrebno razviti programsku podršku, odnosno osigurati upravljanje robotom korištenjem kolaborativne hvataljke i tipkala na njoj. U razvijanje programske podrške uloženo je puno vremena i truda i vremenski je trajalo najduže zbog pojedinih problema koji su se javljali tijekom programiranja.

Najčešći problem javljao se prilikom mrežnog povezivanja *Karel* softvera i robota. Trebalo je puno vremena i pokušaja povezivanja kako bi se ostvarila komunikacija. Problemi su se javljali i prilikom TCP/IP protokola. Dolazilo je do pucanja veze i prekidanja prijenosa, no to se uspjelo softverski riješiti.

No, unatoč svim problemima, rezultati koji se postižu s napravljenom kolaborativnom hvataljkom su zadovoljavajući. Prostor unutar kućišta je maksimalno ispunjen, a korištenjem četiri tipkala koja se nalaze na kućištu i njihovih kombinacija napravljeno je sedam različitih naredbi koje pomažu pri upravljanju robotom. Izvršavanje naučene operacije funkcionira u potpunosti, kao i prekid gibanja ako robot naiđe na prepreku.

Postoje i vidljivi nedostaci u softverskoj podršci za kolaborativnu hvataljku. U svrhu njihovog uklanjanja bilo bi dobro napraviti bolju i točniju kalibraciju, jer vrijednosti na senzoru variraju čak i kad se ništa ne radi s njim. Te vrijednosti se šalju u Matlab i prolaze kroz dvije matrice homogenih transformacija čime se greška još više povećava. Pri vođenju robota na većim

brzinama dolazi do malih oscilacija u kontinuiranosti vođenja, a to bi se moglo riješiti nekim novim metodama za povećanje brzine odziva jer su čekanja u razvijenoj programskoj podršci minimalna. Zadnji nedostatak je uočen prilikom hvatanja predmeta jer se unosi dodatna sila koja opterećuje senzor sile. Kod predmeta malih masa to nije bitno, no kod predmeta većih masa može stvarati probleme. Rješenje bi bilo da robot u svakom trenutku zna nalazi li se predmet u hvataljki ili ne.

Prilikom izrade ovog diplomskog rada i kolaborativne hvataljke naučio sam mnogo toga, posebice što se tiče programiranja robota u *Karel* softveru. To mi je bio prvi susret s tim softverom, te uz priručnik, metodom pokušaja i promašaja i uz savjete mentora uspio sam savladati njegov programski jezik. Također sam naučio razliku između teorije i prakse. U teoriji uvijek sve dobro funkcionira, no praksa je nešto sasvim drugo. Ni jedan dio prakse ne smije se uzeti „zdravo za gotovo“, nego je potrebno sve dobro ispitati. Tako je bilo i s izradom ovog rada. Važno je bilo predvidjeti kako će se modelirani dijelovi izraditi, kako će se međusobno spojiti, hoće li biti mjesta za popratnu opremu, te hoće li odgovarati namjeni za koju su predviđeni. U životu se ništa ne postiže preko noći, a tako je bilo i prilikom izrade ovog rada. Potrebno je konstantno raditi i učiti da bi rezultati uloženog truda bili vidljivi.

# LITERATURA

[1] *Fanuc - Collaborative robot hand guidance,* https://www.fanuc.eu/~/media/corporate/products/robots/accessories/hand-guidance/fea-pr-cr-hand-guidance_1.jpg?w=400, 09.12.2018.

[2] *Grippers or Collaborative Robots VS Grippers for Industrial Robots,* Bouchard, Samuel, https://blog.robotiq.com/bid/72440/Grippers-for-Collaborative-Robots-VS-Grippers-for-Industrial-Robots, 09.12.2018.

[3] *Fanuc - Collaborative robots*, https://www.fanuc.eu/bg/en/robots/robot-filter-page/collaborative-robots, 11.12.2018.

[4] *Human - robot collaboration (HRC),* https://www.kuka.com/en-de/technologies/human-robot-collaboration, 12.12.2018.

[5] *KUKA - LBR iisy*, https://www.kuka.com/en-de/technologies/human-robot-collaboration/lbr-iisy, 12.12.2018.

[6] *SCHUNK - HRC gripper Co-act meets cobots*, https://schunk.com/de_en/co-act/, 12.12.2018

[7] *FANUC - LR Mate 200Ic/5L*, https://www.robots.com/robots/fanuc-lr-mate-200ic-5l, 14.12.2018.

[8] Jerbić, B., i ostali: *FANUC - Specijalistički_1*, predavanja za specijalistički studij, 14.12.2018.

[9] *FANUC R-30iA Mate Controller*, https://www.robots.com/controllers/fanuc-r-30ia-mate-controller, 14.12.2018.

[10] *R - 30iA Mate Controller,* brošura, http://pdf.directindustry.com/pdf/fanuc-europe-corporation/r-30ia-mate/32007-381993.html, 14.12.2018.

[11] *FANUC - Froce Sensor*, brošura, https://www.fanuc.eu/dk/en/robots/accessories/robot-vision/force-sensor, 15.12.2018.

[12] *FANUC - RoboGuide Simulation Software,* https://www.fanucamerica.com/products/robots/robot-simulation-software-FANUC-ROBOGUIDE, 15.12.2018.

[13] *TCP/IP Protocols,* https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.networkcomm/tcpip_protocols.htm, 17.12.2018.

[14] Jokić, A.,: *ROBOTIKA - Kinematika robota*, predavanja iz kolegija Industrijski i mobilni roboti, Fakultet strojarstva i brodogradnje, 2016.

# PRILOZI

I. *Karel* program *matlab_giba*

II. *Karel* program *matlab_kom*

III. *Matlab* program *poruka_proba*

IV. CD-R disk

**PRILOG I** - *Karel* program *matlab_giba*

```
PROGRAM MATLAB_GIBA
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE

VAR
  conf: CONFIG
  p_user,p_tool,p1,p2,p3,p4,p5,p6,p10: XYZWPR
  str: STRING[20]
  P, R, V_X, V_Y, V_Z, V_W, V_R, T_MAX, T_MIN, TR_MIN, TR_MAX, ROT, TM_MAX, TM_MIN: REAL
  MOMX, MOMY, MOMZ, VW, VP, VR: REAL
  FX_0,FY_0, FZ_0, FZ_0d, FZ_0g, MX_0, MY_0, MZ_0, FX, FY, FZ, FZd, MX, MY, MZ, real_value,citaj: REAL
  FG, gravitacija,senzor: REAL
  kutw,kutp,kutr:INTEGER
  silax,silay,silaz, silatx, silaty, silatz: REAL
  STATUS, i, j, k, int_value, a, b, entry: INTEGER
  real_flag, UVIJET_1, UVIJET_2, UVIJET: BOOLEAN
  indx IN DRAM: INTEGER
  nb_bytes: INTEGER
  TCP,iznosi:FILE
  sx,sy,sz,sg,kut,tx,ty,tz:STRING[10]  --tx - trenutna sila u x smjeru...
  STRI, STRI_d, pod1, pod2, pod3, pod4, pod5, pod6:STRING[128]
  deltax, deltay, deltaz, fxt, fyt, fzt, deltaxkp, deltaxkm, deltaykp, deltaykm, deltazkp, deltazkm: REAL
  duljina, i2, poz, i3, i_polja, i4, j1: INTEGER
  polje: ARRAY[100] OF INTEGER
  tipka1, tipka2,tipka3: BOOLEAN
  hvat_ceka: INTEGER


BEGIN
        hvat_ceka=250

        tipka1=FALSE
        tipka2=FALSE
        tipka3=FALSE

        CNV_STR_CONF('nut', conf, STATUS)
    p_tool=POS(0,0,0,180,0,135,conf)
    $GROUP[1].$UTOOL = p_tool  --isti kao i utool_num=0
    --
    p_user=POS(0,0,0,0,0,0,conf)
    $GROUP[1].$UFRAME = p_user   --isti kao u world 0
    --
    $GROUP[1].$MOTYPE = JOINT
    $GROUP[1].$TERMTYPE = FINE
    $GROUP[1].$SPEED = 400
    $GROUP[1].$TERMTYPE = NODECEL
    --
   p1=POS(0,500,400,0,0,0,conf)
   MOVE TO p1
        DELAY 1000


        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[1]',FX_0,STATUS)
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[2]',FY_0,STATUS)
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[3]',FZ_0d,STATUS)
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[4]',MX_0,STATUS)
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[5]',MY_0,STATUS)
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[6]',MZ_0,STATUS)

        p2=POS(80,500,480,55,-90,-56,conf)
        MOVE TO p2      NOWAIT
```

```
                p3=POS(50.127,499.201,542.297,178.646,-38.803,179.639,conf)
                MOVE TO p3 NOWAIT

                        p4=POS(0,500,560,179.9,-0.001,180,conf)
                        MOVE TO p4

        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[3]',FZ_0g,STATUS)
        DELAY 100

        FG=0.5*(FZ_0d-FZ_0g)
        FZ_0=FZ_0d-FG

        WRITE('FG=',FG,CR)
        WRITE('FZ_0=',FZ_0,CR)


        DELAY 1000

        MOVE TO p3 NOWAIT
        MOVE TO p2 NOWAIT
                MOVE TO p1 NOWAIT

                        DELAY 1000
                        MOVE TO p1

                                --tu stisnem tipku 1
        WRITE('pritisni tipku 1',CR)
        tipka1=TRUE


                p5=CURPOS(0,0)
                WRITE (p5,CR)

DELAY 200

--wait for TPIN[50]=TRUE
tipka2_pon::
GET_VAR(entry,'matlab_kom','tipka2',tipka2,STATUS)

IF tipka2=FALSE THEN
        GOTO tipka2_pon
ENDIF

DELAY 4

FOR i_polja=1 TO 90 DO
                polje[i_polja]=1
        ENDFOR
i3=1

----------GLAVNA PETLJA-------------------------
glavnapetlja::
WHILE 1=1 DO


GET_VAR(entry,'matlab_kom','deltax',deltax,STATUS)
GET_VAR(entry,'matlab_kom','deltay',deltay,STATUS)
GET_VAR(entry,'matlab_kom','deltaz',deltaz,STATUS)
--
--DELAY 2 -- mozda ne treba

WHILE RDI[1]=TRUE DO  --tipka 7  prije bilo tpin[55]=true
        --DELAY 2 -- mozda ne treba

GET_VAR(entry,'matlab_kom','deltax',deltax,STATUS)
GET_VAR(entry,'matlab_kom','deltay',deltay,STATUS)
GET_VAR(entry,'matlab_kom','deltaz',deltaz,STATUS)
```

```
--------TRANSLACIJA POMICANJE-------------------
IF ((deltax>0.25) OR (deltax<-0.25) OR (deltay>0.25) OR (deltay<-0.25) OR (deltaz>0.25) OR (deltaz<-0.25))
THEN

        IF ((deltax>2.5) OR (deltax<-2.5) OR (deltay>2.5) OR (deltay<-2.5) OR (deltaz>2.5) OR (deltaz<-2.5))
THEN

                p5.X= p5.X+5*(deltax) --pomicanje robota u baznom k.sustavu u pozitivnom smjeru osi x
                MOVE TO p5 NOWAIT
                p5.Y= p5.Y+5*(deltay)
                MOVE TO p5 NOWAIT
                p5.Z= p5.Z+5*(deltaz)

                MOVE TO p5 NOWAIT

                GET_VAR(entry,'matlab_kom','deltax',deltax,STATUS)
                GET_VAR(entry,'matlab_kom','deltay',deltay,STATUS)
                GET_VAR(entry,'matlab_kom','deltaz',deltaz,STATUS)

        ENDIF

        p5.X= p5.X+1*(deltax) --pomicanje robota u baznom k.sustavu u pozitivnom smjeru osi x
        MOVE TO p5 NOWAIT
        p5.Y= p5.Y+1*(deltay)
        MOVE TO p5 NOWAIT
        p5.Z= p5.Z+1*(deltaz)

        MOVE TO p5 NOWAIT

        GET_VAR(entry,'matlab_kom','deltax',deltax,STATUS)
        GET_VAR(entry,'matlab_kom','deltay',deltay,STATUS)
        GET_VAR(entry,'matlab_kom','deltaz',deltaz,STATUS)

ENDIF

--DELAY 2 -- mozda ne treba

--MOMENTIII
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[4]',MX,STATUS)
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[5]',MY,STATUS)

MOMX=ABS(MX-MX_0)
MOMY=ABS(MY-MY_0)
WHILE ((MOMX>30) OR (MOMY>30)) DO --ako su momenti oko x i y veliki radi samo rotacije oko x i y
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[4]',MX,STATUS)
        GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[5]',MY,STATUS)

    VW=ABS(MX-MX_0)/10
    VP= ABS(MY-MY_0)/10

    IF VW > 10 THEN VW=10
    ENDIF
    IF VP > 10 THEN VP=10
    ENDIF

    IF MX-MX_0>30 THEN p5.w= p5.W+0.05*VW
    ENDIF
    IF MX-MX_0<30 THEN p5.w= p5.W-0.05*VW
    ENDIF
    IF MY-MY_0>30 THEN p5.p= p5.P-0.05*VP
    ENDIF
    IF MY-MY_0<30 THEN p5.p= p5.P+0.05*VP
    ENDIF

    MOMX=ABS(MX-MX_0)
    MOMY=ABS(MY-MY_0)
```

```
      MOVE TO p5 NOWAIT

          endwhile
          MOVE TO p5 NOWAIT


GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[6]',MZ,STATUS)
MOMZ=ABS(MZ-MZ_0)
WHILE (MOMZ>15) DO  --kad je moment oko z veci od TR_MAX radi samo rotaciju oko z
           GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[6]',MZ,STATUS)

           VR= ABS(MZ-MZ_0)/10

      IF VR > 10 THEN VR=10
      ENDIF

      IF MZ-MZ_0>0 THEN p5.r= p5.R-0.05*VR
      ENDIF
      IF MZ-MZ_0<0 THEN p5.r= p5.R+0.05*VR
      ENDIF

      MOMZ=ABS(MZ-MZ_0)

      MOVE TO p5 NOWAIT

          ENDWHILE


          IF RDI[2] THEN  --tipka 8 na TPu  prije bilo tpin[56]
                          SET_POS_REG(i3,p5,STATUS)
                           WRITE('pozicija',GET_POS_REG(i3,STATUS),CR)
                           WRITE('p5',p5,CR)

                           WRITE('i3',i3,CR)

                           wait for RDI[2] = FALSE
                           i3=i3+1
          ENDIF

          IF RDI[3] THEN --tipka 4 na TPu. zrak pusten  tpin[52]
                          polje[i3]=10
                          SET_POS_REG(i3,p5,STATUS)
                          WRITE('polje',polje[i3],CR)
                          WRITE('i3',i3,CR)
                          CLOSE HAND 1
                          wait for RDI[3] = FALSE
                          i3=i3+1
          ENDIF
          IF RDI[5] THEN  --tipka 5 na TPu, zrak zatvoren  tpin[53]
                          polje[i3]=20
                          SET_POS_REG(i3,p5,STATUS)
                          WRITE('polje',polje[i3],CR)
                          WRITE('i3',i3,CR)
                          OPEN HAND 1
                          wait for RDI[5] = FALSE
                          i3=i3+1
          ENDIF

          endwhile

          IF ((RDI[2]) AND (RDI[3])) THEN
                    GOTO izvrsavanje
          ENDIF
DELAY 1
ENDWHILE
```

```
izvrsavanje:: --izvrsavanje naucenog programa

DELAY 2000

i4=i3-1   --bilo i3-1
FOR j1=1 TO i4 DO --(i-1) staviti u neku varijablu

                GET_VAR(entry,'matlab_kom','deltax',deltax,STATUS)
                GET_VAR(entry,'matlab_kom','deltay',deltay,STATUS)
                GET_VAR(entry,'matlab_kom','deltaz',deltaz,STATUS)

                WRITE('deltax= ',deltax,CR)
                WRITE('deltay= ',deltay,CR)
                WRITE('deltaz= ',deltaz,CR)

        IF ((deltax>0.5) OR (deltax<-0.5) OR (deltay>0.5) OR (deltay<-0.5) OR (deltaz>0.5) OR (deltaz<-0.5))
THEN

                ABORT_TASK('*ALL*',TRUE,TRUE,STATUS)

        ENDIF
                WRITE('registar; ',j1,CR)
                WRITE(GET_POS_REG(j1,STATUS),CR)
                p10=GET_POS_REG(j1,STATUS)

                MOVE TO p10
                IF polje[j1]=10 THEN
                        DELAY hvat_ceka
                        CLOSE HAND 1  --zrak zatvoren
                        DELAY hvat_ceka
                ENDIF
                IF polje[j1]=20 THEN
                        DELAY hvat_ceka
                        OPEN HAND 1  --zrak pusten
                        DELAY hvat_ceka
                ENDIF

                WRITE('polje: ',polje[j1],CR)


ENDFOR

        WHILE 1=1 DO
        --za izvođenje iste operacije ponovno prtisnuti zelenu i plavu
        IF ((RDI[2]) AND (RDI[3])) THEN
                GOTO izvrsavanje
        ENDIF

        --za učenje nove operacije pritsnuti plavu i crvenu
        IF ((RDI[3]) AND (RDI[5])) THEN
                i3=1
                GOTO glavnapetlja
        ENDIF
        --za kraj programa
        IF ((RDI[1]) AND (RDI[5])) THEN
                GOTO krajprograma
        ENDIF
        ENDWHILE

        krajprograma::
        tipka3=TRUE
        WRITE('Matlab_giba je zavrsio',CR)

END MATLAB_GIBA
```

## **PRILOG II** - *Karel* program *matlab_kom*

```
PROGRAM MATLAB_KOM
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE

VAR
  conf: CONFIG
  p_user,p_tool,p1,p2,p3,p4,p5,p6,p10: XYZWPR
  str: STRING[20]
  P, R, V_X, V_Y, V_Z, V_W, V_R, T_MAX, T_MIN, TR_MIN, TR_MAX, ROT, TM_MAX, TM_MIN: REAL
  MOMX, MOMY, MOMZ, VW, VP, VR: REAL
  FX_0,FY_0, FZ_0, FZ_0d, FZ_0g, MX_0, MY_0, MZ_0, FX, FY, FZ, FZd, MX, MY, MZ, real_value,citaj: REAL
  FG, gravitacija,senzor: REAL
  kutw,kutp,kutr:INTEGER
  silax,silay,silaz, silatx, silaty, silatz: REAL
  STATUS, i, j, k, int_value, a, b, entry,cekanje: INTEGER
  real_flag, UVIJET_1, UVIJET_2, UVIJET: BOOLEAN
  indx IN DRAM: INTEGER
  nb_bytes: INTEGER
  TCP,iznosi:FILE
  sx,sy,sz,sg,kut,tx,ty,tz:STRING[10]  --tx - trenutna sila u x smjeru...
  STRI, STRI_d, pod1, pod2, pod3, pod4, pod5, pod6:STRING[128]
  deltax, deltay, deltaz, fxt, fyt, fzt,rv: REAL
  duljina, i2, poz, i3, i_polja, i4, j1: INTEGER
  polje: ARRAY[100] OF INTEGER
  tipka1,tipka2,tipka3,rf: BOOLEAN

  ----------VANJSKE RUTINE----------
  ROUTINE OPEN_FILE_ (FILE_ : FILE; TAG_ : STRING) FROM LIB_FIL3
  ROUTINE CLOSE_FILE_(FILE_ : FILE; TAG_ : STRING) FROM LIB_FIL3

BEGIN
        tipka1=FALSE
        tipka2=FALSE
        tipka3=FALSE

        --wait for TPIN[49]=TRUE --tipka 1
        tipka1_pon::
        GET_VAR(entry,'matlab_giba','tipka1',tipka1,STATUS)

        IF tipka1=FALSE THEN
                GOTO tipka1_pon
        ENDIF

        GET_VAR(entry,'matlab_giba','FX_0',FX_0,STATUS)
        GET_VAR(entry,'matlab_giba','FY_0',FY_0,STATUS)
        GET_VAR(entry,'matlab_giba','FZ_0',FZ_0,STATUS)
        GET_VAR(entry,'matlab_giba','FG',FG,STATUS)
        --GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[5]',MY_0,STATUS)
        --GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[6]',MZ_0,STATUS)

    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$OPER',0,STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$STATE',0,STATUS) ; DELAY 20
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$PROTOCOL','SM',STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$REPERRS','FALSE',STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$TIMEOUT',9999,STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$PWRD_TIMEOUT',0,STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$SERVER_PORT',5555,STATUS) ;   --port
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$STRT_PATH','192.168.123.25',STATUS) ; ---IP OD M3
.192.168.123.27
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$STRT_REMOTE','192.168.123.25',STATUS) ;  --IP OD M3
.192.168.123.27
```

```
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$PATH','192.168.123.25',STATUS) ;  --IP OD M3
.192.168.123.27
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$remote','192.168.123.25',STATUS) ; --IP OD M3
.192.168.123.27 prije bilo .124



    DELAY 10 ; SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$OPER',3,STATUS);
      DELAY 10 ;
        SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[8].$STATE',3,STATUS) ;
        DELAY 10 ;
        --reconn_::    --
    DELAY 100;  CLOSE_FILE_(TCP,'S8:') ; DELAY 100;
    OPEN_FILE_(TCP,'S8:');


WRITE('pritisni tipku2',CR)
tipka2=TRUE
DELAY 4

------GLAVNA PETLJA----------------
WHILE 1=1 DO


  GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[1]',FX,STATUS)
  GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[2]',FY,STATUS)
  GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[3]',FZd,STATUS)
  GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[4]',MX,STATUS)
  GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[5]',MY,STATUS)
  GET_VAR(entry,'*SYSTEM*','$CCC_GRP[1].$FS_FORCE[6]',MZ,STATUS)
  GET_VAR(entry,'matlab_giba','p5',p5,STATUS)


        kutw=ROUND(p5.w*100)+100000
        kutp=ROUND(p5.p*100)+100000
        kutr=ROUND(p5.r*100)+100000

   --DELAY 2 -- mozda ne treba
   SET_REAL_REG(1,FX_0,STATUS)
   SET_REAL_REG(2,FY_0,STATUS)
   SET_REAL_REG(3,FZ_0,STATUS)
   SET_REAL_REG(4,FG,STATUS)
   SET_REAL_REG(5,FX,STATUS)
   SET_REAL_REG(6,FY,STATUS)
   SET_REAL_REG(7,FZd-FG,STATUS)
   silax=FX_0;
   silay=FY_0;
   silaz=FZ_0;
   silatx=FX;
   silaty=FY;
   silatz=FZd-FG;
   gravitacija=FG;
   CNV_REAL_STR(silax,5,2,sx)
   CNV_REAL_STR(silay,5,2,sy)
   CNV_REAL_STR(silaz,5,2,sz)
   CNV_REAL_STR(gravitacija,5,2,sg)
   CNV_REAL_STR(silatx,5,2,tx)
   CNV_REAL_STR(silaty,5,2,ty)
   CNV_REAL_STR(silatz,5,2,tz)

  --WRITE TCP ($FAST_CLOCK,CR)
   WRITE
TCP(silax,'@',silay,'@',silaz,'@',gravitacija,'@',kutw,'@',kutp,'@',kutr,'@',silatx,'@',silaty,'@',silatz,'~',CR);

   ponovo::

   BYTES_AHEAD(TCP,nb_bytes,STATUS)
```

```
    IF nb_bytes=0 THEN;
          GET_REG(200,rf,cekanje,rv,STATUS)
          DELAY cekanje; GOTO ponovo; ENDIF
    --WRITE('nb_bytes',nb_bytes,CR)

    deltax=0
    deltay=0
    deltaz=0
IF nb_bytes<>0 THEN
          WRITE('nb_bytes; ',nb_bytes,CR)

READ TCP(STRI::nb_bytes)
WRITE(STRI,CR)

STRI_d=STRI
duljina=STR_LEN(STRI_d)

FOR i2=1 TO duljina DO
          IF SUB_STR(STRI_d,i2,1)='Y' THEN
                pod1=SUB_STR(STRI_d,2,i2-2)
          ENDIF
ENDFOR

CNV_STR_REAL(pod1,deltax)

FOR i2=1 TO duljina DO
          IF SUB_STR(STRI_d,i2,1)='Y' THEN
                poz=i2+1
          ENDIF
          IF SUB_STR(STRI_d,i2,1)='Z' THEN
                pod2=SUB_STR(STRI_d,poz,i2-poz)
          ENDIF
ENDFOR

CNV_STR_REAL(pod2,deltay)

FOR i2=1 TO duljina DO
          IF SUB_STR(STRI_d,i2,1)='Z' THEN
                poz=i2+1
          ENDIF
          IF SUB_STR(STRI_d,i2,1)='K' THEN
                pod3=SUB_STR(STRI_d,poz,i2-poz)
          ENDIF
ENDFOR

CNV_STR_REAL(pod3,deltaz)

    nb_bytes=0

    GOTO van
ENDIF
van::

GET_VAR(entry,'matlab_giba','tipka3',tipka3,STATUS)
IF tipka3=TRUE THEN
          GOTO krajprograma
ENDIF

DELAY 1
ENDWHILE

  DELAY 100;  CLOSE_FILE_(TCP,'S8:') ; DELAY 100;

krajprograma::
WRITE('Matlab_kom je zavrsio',CR)

END MATLAB_KOM
```

**PRILOG III** - *Matlab* program *poruka_proba*

```
clear all
clc
k1=0;k2=0;k3=0;k4=0;k5=0;k6=0;k7=0;k8=0;k9=0;k10=0;
pod1=0;pod2=0;pod3=0;pod4=0;pod5=0;pod6=0;pod7=0;pod8=0;pod9=0;pod10=0;
crtanje=0;
global message_c
while true
    ip='192.168.123.25';port=5555;
    X_client_open_rw(ip,port,100000);
    fprintf('Veza uspješno uspostavljena\n')

    %%
    fprintf('ULAZIM U PETLJU\n');

    x(1:50)=0;
    y(1:50)=0;
    u(1:50)=0;
    v(1:50)=0;

    while true
        status=0;
        while status==0
           pause(0.01);
           status=poruka();
           status

           crtanje(end+1)=status;
           plot(crtanje,'-b')
           xlim([50 100])
           title('Prikaz stanja primljene poruke')
           xlabel('Broj pokušaja')
           ylabel('0-nema poruke, 1-primljena poruka')
           hold on

        end

        splitan=strsplit(message_c,'~');
        novi=splitan(1,1);
        trazeni=char(novi)
        a=size(trazeni);
        b=a(2);


        if isempty(trazeni)
            continue
        end


        for i=1:b
            if trazeni(i)=='@'
                k1=i+1;
            break;
            end
```

```
    pod1(i)=trazeni(i);
end
pod1=char(pod1);
FX_0=str2num(pod1);

for j=k1:b
    if trazeni(j)=='@'
        k2=j+1;
    break;
    end
    pod2(j-i)=trazeni(j);
end
pod2=char(pod2);
FY_0=str2num(pod2);

for z=k2:b
    if trazeni(z)=='@'
        k3=z+1;
    break;
    end
    pod3(z-j)=trazeni(z);
end
pod3=char(pod3);
FZ_0=str2num(pod3);

for w=k3:b
    if trazeni(w)=='@'
        k4=w+1;
    break;
    end
    pod4(w-z)=trazeni(w);
end
pod4=char(pod4);
FG_0=str2num(pod4);

for q=k4:b
    if trazeni(q)=='@'
        k5=q+1;
    break;
    end
    pod5(q-w)=trazeni(q);
end
pod5=char(pod5);
pod5=str2num(pod5);
kutw=(180+((pod5-100000)/100))*pi/180 %prije bio minus ispred
kutwtren=-((pod5-100000)/100)*pi/180;

for e=k5:b
    if trazeni(e)=='@'
        k6=e+1;
    break;
    end
    pod6(e-q)=trazeni(e);
end
pod6=char(pod6);
pod6=str2num(pod6);
kutp=-((pod6-100000)/100)*pi/180 %prije nije bilo minusa
kutptren=((pod6-100000)/100)*pi/180;
```

```
        for o=k6:b
            if trazeni(o)=='@'
                k7=o+1;
            break;
            end
            pod7(o-e)=trazeni(o);
        end
        pod7=char(pod7);
        pod7=str2num(pod7);
        kutr=-((pod7-100000)/100)*pi/180 %prije nije bilo minusa
        kutrtren=((pod7-100000)/100)*pi/180;

        for aaa=k7:b
            if trazeni(aaa)=='@'
                k8=aaa+1;
                break;
            end
            pod8(aaa-o)=trazeni(aaa);
        end
        pod8=char(pod8);
        FX=str2num(pod8);

        for o2=k8:b
            if trazeni(o2)=='@'
                k9=o2+1;
                break;
            end
            pod9(o2-aaa)=trazeni(o2);
        end
        pod9=char(pod9);
        FY=str2num(pod9);

        for o3=k9:b
            if trazeni(o3)=='@'
                k10=o3+1;
                break;
            end
            pod10(o3-o2)=trazeni(o3);
        end
        pod10=char(pod10);
        FZ=str2num(pod10)


        F0=[FX_0;FY_0;FZ_0;1];
        FG=[0;0;FG_0;0];
        Ftrenutne=[FX;FY;FZ;1]; %sa senzora

        A=[1 0 0 0;0 cos(kutw) -sin(kutw) 0;0 sin(kutw) cos(kutw) 0;0 0 0
1];
        B=[cos(kutp) 0 sin(kutp) 0;0 1 0 0;-sin(kutp) 0 cos(kutp) 0;0 0 0
1];
        C=[cos(kutr) -sin(kutr) 0 0;sin(kutr) cos(kutr) 0 0;0 0 1 0;0 0 0
1];

        Atren=[1 0 0 0;0 cos(kutwtren) -sin(kutwtren) 0;0 sin(kutwtren)
cos(kutwtren) 0;0 0 0 1];
        Btren=[cos(kutptren) 0 sin(kutptren) 0;0 1 0 0;-sin(kutptren) 0
cos(kutptren) 0;0 0 0 1];
```

```
        Ctren=[cos(kutrtren) -sin(kutrtren) 0 0;sin(kutrtren) cos(kutrtren)
0 0;0 0 1 0;0 0 0 1];

        T=A*B*C;
        Ttren=Atren*Btren*Ctren;

        Fbxyz=T*F0-FG; %dobro jee

        Fsenzor=F0+Ttren*FG;

        Fsenpoc=Ftrenutne-Ttren*FG+FG;

        Ftren=T*Fsenpoc-FG;

        Fbxyz1=Fbxyz(1);
        Fbxyz2=Fbxyz(2);
        Fbxyz3=Fbxyz(3);

        Ftren1=Ftren(1);
        Ftren2=Ftren(2);
        Ftren3=Ftren(3);

        deltax=Ftren1-Fbxyz1;
        deltay=Ftren2-Fbxyz2;
        deltaz=Ftren3-Fbxyz3;

        deltaxs=num2str(deltax);
        deltays=num2str(deltay);
        deltazs=num2str(deltaz);


        poruka_za_poslat=strcat('X',deltaxs,'Y',deltays,'Z',deltazs,'K')
        poruka_za_poslat;

        X_client_send_rw(poruka_za_poslat);


    end
end
```