

Osjetljivo robotsko bušenje

Rorbach, Tomislav

Master's thesis / Diplomski rad

2018

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://um.nsk.hr/um:nbn:hr:235:676709>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-24**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Tomislav Rorbach

Zagreb, 2018.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentori:

Prof. dr. sc. Bojan Jerbić

Student:

Tomislav Rorbach

Zagreb, 2018.

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

Zahvaljujem se mentoru prof. dr. sc. Bojanu Jerbiću i na pomoći, sugestijama i strpljenju prilikom izrade diplomskog rada.

Tomislav Rorbach



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 Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	
Ur. broj:	

DIPLOMSKI ZADATAK

Student: **TOMISLAV RORBACH**

Mat. br.: 0035179542

Naslov rada na hrvatskom jeziku: **Osjetljivo robotsko bušenje**

Naslov rada na engleskom jeziku: **Force sensitive robot drilling**

Opis zadatka:

Proces bušenja nehomogenih materijala kao što su ljudske i životinjske kosti pokazao se kao izazovan upravljački problem s izrazito naglašenom mogućnošću primjene u medicini. U svrhu kontrole procesa bušenja nehomogenih materijala robotskom rukom, potrebno je razviti upravljački model koji će omogućiti kvalitetnu izvedbu zadatka bušenja prema unaprijed utvrđenim kriterijima bušenja primjenjivim u medicini. Upravljački model bušenja treba uzeti u obzir silu bušenja za podešavanje ostalih parametara modela.

U sklopu rada potrebno je razvijeni simulacijski model testirati te odrediti inicijalne parametre modela koji daju stabilno ponašanje modela uz brzinu odziva koji minimira hod kod proboja bušenog materijala. Model verificirati na realnom laboratorijskom postavu koji uključuje 6-osnu robotsku ruku sa senzorom sile i momenata i medicinskom bušilicom u scenariju bušenja životinjske kosti ili drugih prikladnih nehomogenih materijala.

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

Zadatak zadan:
08. ožujka 2018.

Rok predaje rada:
10. svibnja 2018.

Predviđeni datum obrane:
16. svibnja 2018.
17. svibnja 2018.
18. svibnja 2018.

Zadatak zadao:


prof. dr. sc. Bojan Jerbić

Predsjednica Povjerenstva:


prof. dr. sc. Biserka Runje

SADRŽAJ

POPIS SLIKA	2
POPIS TABLICA.....	4
SAŽETAK.....	5
SUMMARY	6
1 UVOD.....	7
2 NEHOMOGENI MATERIJALI.....	10
2.1 Ljudske kosti.....	10
3 BUŠENJE	12
3.1 Svrkla u medicini	15
4 MATEMATIČKI MODEL BUŠENJA	16
5 LABORATORIJSKI POSTAV	18
5.1 Komunikacija	19
5.1.1 OSI referentni model.....	19
5.1.2 Topologija računalnih mreža	21
5.1.3 Ethernet	23
5.1.4 DoD mrežni model – TCP/IP porodica protokola	25
5.1.5 Internet Protocol - IP.....	27
5.1.6 User Datagram Protocol - UDP	28
5.1.7 Transmission Control Protocol - TCP.....	29
5.2 Net F/T	32
5.2.1 Net F/T Gamma pretvornik.....	32
5.2.2 NETB sučelje	34
5.2.3 Komunikacija Net F/T – računalo.....	35
5.3 Robotski sustav Kuka Lightweight Robot LWR4+	38
5.3.1 Komunikacija Kuka LWR4+ - vanjsko računalo.....	39
5.4 Sustav za bušenje Medtronic.....	42
5.5 Računalo.....	44
5.6 Upravljački sustav.....	45
6 KALIBRACIJA ALATA.....	48
6.1 Matrica rotacije sile.....	51
7 EKSPERIMENTALNI REZULTATI	54
8 ZAKLJUČAK.....	62
LITERATURA.....	64
PRILOZI.....	65

POPIS SLIKA

Slika 1.	Industrijske revolucije - ilustracija [1]	7
Slika 2.	Razvoj kompleksnosti kirurških zahvata [1]	8
Slika 3.	Ljudski skelet	10
Slika 4.	Duga kost.....	11
Slika 5.	Pločasta kost.....	11
Slika 6.	Vijčano svrdlo [12].....	12
Slika 7.	Svrdlo oslobođeno veza[4].....	12
Slika 8.	Shematski prikaz bušenja vijčanim svrdlom [4]	14
Slika 9.	Medicinska svrdla.....	15
Slika 10.	Očitane sile pritiska svrdla prilikom bušenja kosti [12].....	17
Slika 11.	Jedinična barijera kosti.....	17
Slika 12.	Laboratorijski postav za bušenje	18
Slika 13.	OSI referentni model	19
Slika 14.	Topologija računalnih mreža [6]	21
Slika 15.	Provodnici definirani Ethernet protokolom [6]	24
Slika 16.	Usporedba OSI, DoD i TCP/IP modela.....	25
Slika 17.	Enkapsulacija / Dekapsulacija [5]	26
Slika 18.	Protokoli TCP/IP modela	27
Slika 19.	Zaglavlje IPv4 protokola	27
Slika 20.	Zaglavlje UDP protokola.....	28
Slika 21.	<i>Three-way handshake</i> – uspostava veze korištenjem TCP protokola [7]	30
Slika 22.	Zaglavlje TCP protokola	30
Slika 23.	Net F/T sustav [4].....	32
Slika 24.	Gamma pretvornik (SI-65-5).....	32
Slika 25.	Mjerni opsezi Gamma pretvornika (SI-65-5) [8]	33
Slika 26.	NETB sučelje – fizičko spajanje UTP kabla [8]	34
Slika 27.	NETB sučelje – fizičko definiranje IP adrese	35
Slika 28.	Komunikacije postavke Net F/T sustava.....	36
Slika 29.	Vrijednosti mjerenih komponenti sile i momenta	37
Slika 30.	Odziv senzora sile	37

Slika 31.	Robotski sustav Kuka LWR 4+[9]	38
Slika 32.	Kuka LWR 4+ – elementi i upravljačke rotacijske osi [9].....	39
Slika 33.	Dijagram toka FRI-a [10]	40
Slika 34.	Kvaliteta pijenosa podataka [10]	41
Slika 35.	IPC Model EC300 [11].....	42
Slika 36.	Midas Rex® Legend EHS® Motor [11]	43
Slika 37.	Jacobs® reduktor sa steznom glavom AD02 (5/32") [11]	43
Slika 38.	Višenamjenska nožna kontrolna jedinica [11]	44
Slika 39.	Shematski prikaz komunikacije unutar laboratorijskog postava za bušenje	45
Slika 40.	Nehomogeni materijal – prikaz parametara algoritma[12]	46
Slika 41.	Algoritam za detekciju proboja	47
Slika 42.	Koordinatni sustavi robota [9].....	48
Slika 43.	Kalibracija TCP-a svrdla i TCP-a senzora sile.....	50
Slika 44.	Kalibracija orijentacije koordirdinatnih sustava svrdla i senzora sile.....	50
Slika 45.	Koordinatni sustavi prirubnice, senzora i svrdla[9]	51
Slika 46.	Komponente sile na svrdlu	53
Slika 47.	Sila pritiska i posmična brzina bušenja;drvo ; $K_p = 10, K_i = 0$	54
Slika 48.	Sila pritiska i posmična brzina bušenja; Al lim ; $K_p = 40, K_i = 20$	55
Slika 49.	Nehomogena struktura (lim-eps-lim)	55
Slika 50.	Sila pritiska i posmična brzina bušenja; lim – eps - lim; $K_p = 40, K_i = 20$	56
Slika 51.	Sila pritiska i posmična brzina bušenja; lim – eps - lim; $K_p = 40, K_i = 20$; detekcija proboja	57
Slika 52.	Teleča lopatična kost	57
Slika 53.	Sila pritiska i posmična brzina bušenja; kompaktna kost; $K_p = 40, K_i = 20$	58
Slika 54.	Sila pritiska i posmična brzina bušenja; kost ; $K_p = 40, K_i = 20$	58
Slika 55.	Sila pritiska i posmična brzina bušenja; kost;	59
Slika 56.	Kvantifikacija duljine hoda svrdla prilikom proboja	60
Slika 57.	Udaljenost očitana sa laserskoga mjerača udaljenosti.....	61

POPIS TABLICA

Tablica 1. Osnovne karakteristike Gamma pretvornika (SI-65-5)	31
Tablica 2. Mjerni opsezi i rezolucija Gamma pretvornika (SI-65-5).....	31
Tablica 3. Kuka LWR 4+ – osnovni tehnički podatci [9].....	36

SAŽETAK

U sklopu ovoga rada postavljen je i testiran u labaratorijskim uvjetima postav za osjetljivo robotsko bušenje. Postav se sastoji od nekoliko sustava: robotskog sustav Kuka Lightweight Robot LWR4+, sustava za očitavanje sile i momenta ATI Net F/T sa mjernim osjetnikom „Gamma“, sustava namjenjenog za bušenje u medicini proizvođača Medtronic, te osobnog računala (PC-a). Na osobnome računalu programski jezik C++ (cpp) u sklopu programskoga paketa Microsoft Visual Studio 2010, simultano izvršavajući više programskih kodova (eng. *multithreading*) održava komunikacijske veze unutar postava, te PI regulator upravljačkom posmičnom brzinom svrdla održava silu pritiska svrdla tijekom bušenja konstantnom. Detekcija proboja bušenoga materijala odvija se izvršavanjem algoritma osjetljivoga na pad sile pritiska svrdla prilikom bušenja, na temelju postavljenih vrijednosnih pragova (eng. *threshold*). Postav je testiran bušeci homogeni materijal (Al lim), umjetnu nehomogenu strukturu (Al lim – eps – Al lim), te teleću lopatičnu kost.

Ključne riječi: nehomogeni materijali, bušenje, regulacija, senzor sile i momenta

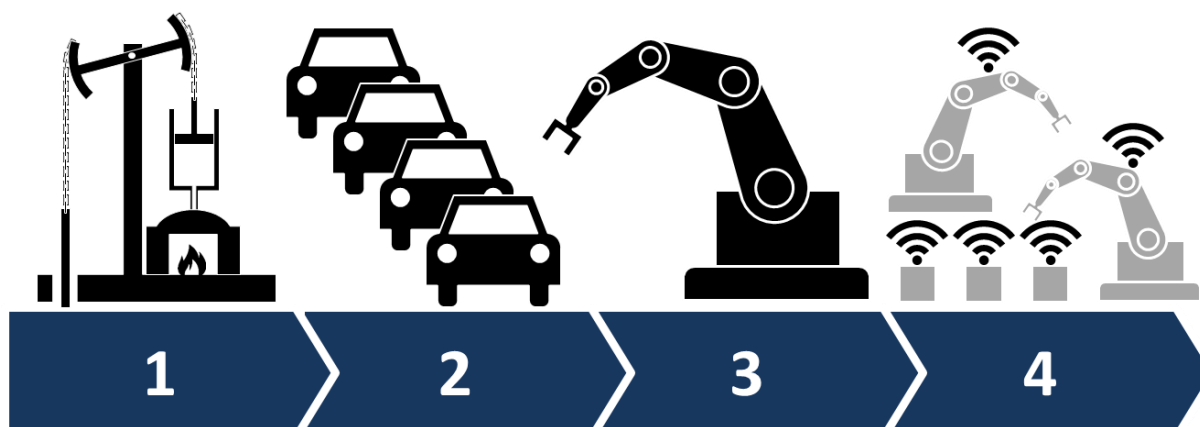
SUMMARY

Within this work, a predetermined robotic drilling pattern was set up and tested in laboratory conditions. The set consists of several systems: Lightweight Robot LWR4 + Robotic System, Network Force/Torque Sensor System with Gamma sensor, Medtronic Drill System, and personal computer (PC). On a personal computer, C ++ (cpp) programming language within the Microsoft Visual Studio 2010 program package, simultaneously executing multiple program codes (multithreading), maintains communication within the setup, and the PI controller with feed rate maintains the force of the drill bit during drilling constant. Detection of breakthrough is performed by performing a drilling bit-sensitive thrust force algorithm based on the set threshold. The installation was tested by drilling a homogeneous material (Al sheet), an artificial inhomogeneous structure (Al sheet - eps - Al sheet), and a calf bone.

Key words: inhomogeneous materials, drilling, control, force and torque sensor

1 UVOD

Trenutno stanje globalne ekonomije, predviđeni pravci daljnjeg ravoja industrije i sve veće korištenje visokih tehnologija, govore nam kako smo zakoračili u novo industrijsko doba, Industriju 4.0. Ovaj termin prezentiran je od strane Njemačke vlade i podrazumijeva niz promjena kojima svjedočimo i koje nas još u budućnosti očekuju uslijed sve većeg korištenja novih tehnologija, ne samo u industriji, već i u ostalim područjima svakodnevnoga života. [1]



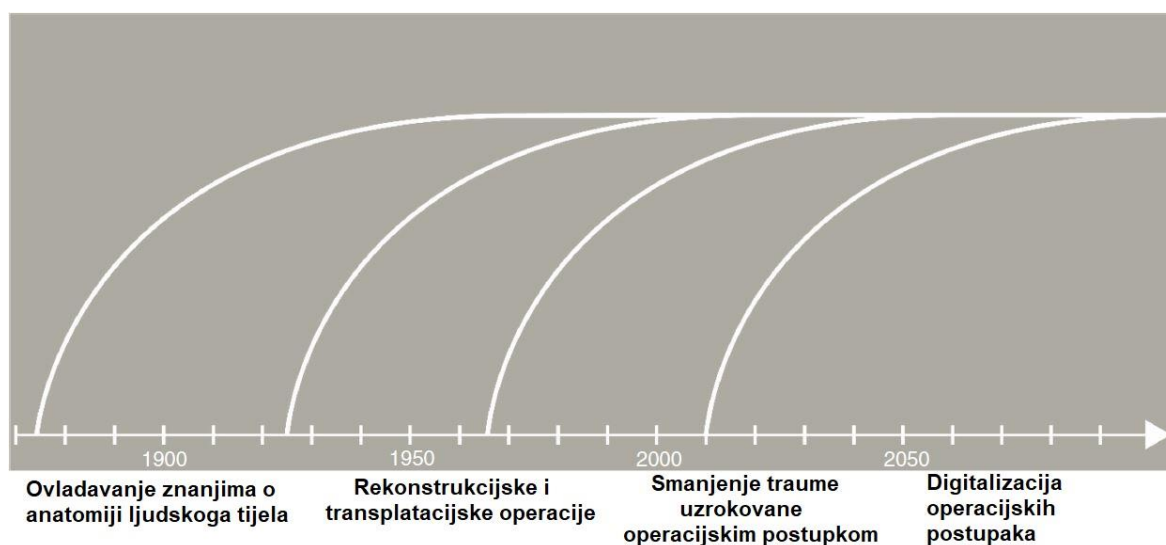
Slika 1. Industrijske revolucije - ilustracija [1]

Na slici 1. simbolično su prikazane industrijske revolucije :

1. Industrijska revolucija - početak korištenja prvih strojeva u industriji čije je korištenje ovisilo o dostupnosti vode i vodene pare
2. Industrijska revolucija - početak masovne proizvodnje omogućeno korištenjem električnih strojeva
3. Industrijska revolucija - početak korištenja informacijskih tehnologija (IT) i robotskih sustava za automatiziranu proizvodnju
4. Industrijska revolucija (4.0) - podrazumijeva korištenje visoko automatiziranih tehnologija sposobnih koristiti metode poput samo-podešavanja, samo-optimiranja, samo-dijagnostike te u konačnici i samostalnog odlučivanja, a sve skupa omogućeno korištenjem umjetne inteligencije, velikih baza podataka i umreženosti opreme.

Razvoj medicine kroz povijest bio je usko vezan uz tehnološki napredak. Analogno industrijskim revolucijama i medicinu također možemo podijeliti u četiri stupnja razvoja, sa karakterističnim pojavama i trendovima za pojedina razdoblja:[2]

1. „Medicina 1.0“ - stoljećima se medicina oslanjala na znanje visokoobrazovanih liječnika, čije se liječenje pacijenata zasnivalo na korištenju skromnog broja lijekova uglavno baziranih na prirodnim tvarima.
2. „Medicina 2.0“ - razdoblje otkrića i primjene antibiotika, te dijagnostičke primjene rendgenskih zraka.
3. „Medicina 3.0“ - razdoblje primjene elektroničke opreme, korištenje novih tehnologija i elektronskih mikrosustava. Korištenje navigacije u kirurgiji, prepoznavanje slika, izrada implantanata i početak korištenja robota u medicini.
4. „Medicina 4.0“ – razdoblje u koje polako zakoračujemo, početak korištenja sustava temeljenih na umjetnoj inteligenciji sa pristupom velikim bazama podataka za primjenu u dijagnostici, prevenciji ali i samome liječenju pacijenata. Sve veća autonomija i sloboda samoodlučivanja robotskih sustava korištenih u operacijskim postupcima.



Slika 2. Razvoj kompleksnosti kiruških zahvata [1]

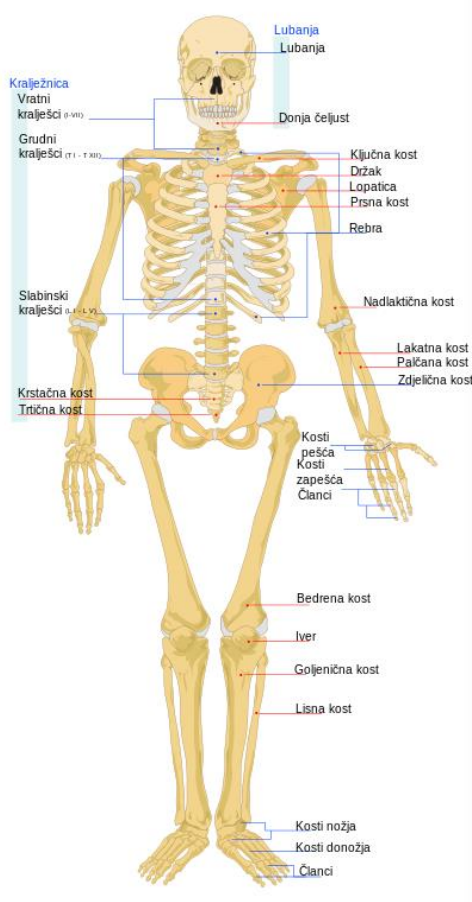
Na slici 2. prikazana je vremenska linija na kojoj su navedene najznačajnije karakteristike pojedinih vremenskih razdoblja na polju kirurgije. Trenutno se jako velika važnost pridaje redukciji oštećenja tkiva prilikom invanzivnih operativnih postupaka, sa ciljem što manjeg broja negativnih nuspojava i što bržeg oporavka pacijenta. Postupci bušenja prilikom operativnih kirurških zahvata pripadaju skupini invanzivnih postupaka. Sami postupak bušenja kosti oštećuje kost, mehanički – odvajanjem čestica, ali isto tako i termički - zagrijavanjem kosti zbog samoga procesa bušenja. Povišena temperatura, koja se oslobađa tijekom procesa bušenja, ima dvije negativne posljedice: Termičku nekrozu kosti i deformaciju samoga provrta. Termička nekroza kosti je proces odumiranja stanica u kostima uslijed povišene temperature. Prema nekim istraživanjima[4] već prilikom temperature kosti od 47°C kroz 1 minutu dolazi do ireverzibilnih promjena građe i funkcije kosti. Također prilikom bušenja kosti postoji velika mogućnost oštećenja tkiva prilikom proboja. Sa ciljem smanjenja negativnih nuspojava proces je potrebno optimizirati i dovesti do stanja što manjeg utjecaja na organizam pacijenta.

2 NEHOMOGENI MATERIJALI

Homogenost označava jednodolnost nekoga svojstva unutar promatranoga objekta, sustava, skupa, područja ili cjeline. Homogeni materijali su oni materijali kod kojih je gustoća jednoliko raspodijeljena u svakoj točki volumena tijela. Analogno definiciji homogenosti materijala, nehomogeni materijali su stoga oni materijali koji nemaju jednoliko raspoređenu gustoću, a samim time niti mehaničko-fizikalna svojstva po cijelome volumenu promatranoga tijela. Primjeri nehomogenih materijala svakako su ljudske i životinjske kosti.

2.1 Ljudske kosti

Kosti su mezenhimno potporno tkivo, koje se sastoji od stanica i međustanične supstancije koja je u slučaju kosti čvrsta, jer se u nju uložene vapnene soli. Zadaća je koštanog sustava da daje potporu tijelu, da zaštićuje pojedine tjelesne šupljine i organe u njima i da služi kao insercija mišićima. Broj kostiju skeleta kod odrasle osobe iznosi 206. [3]

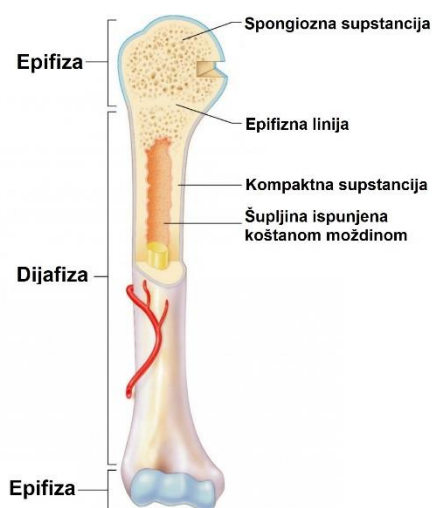


Slika 3. Ljudski skelet

Prema formi se kosti mogu podijeliti na duge, kratke i pločaste. Duga kost je npr. nadlaktična i bedrena kost, kratke kosti su kralješci i kosti šake, dok su pločaste kosti lopatica, kosti kalvarije (lubanjski svod) i zdjelične kosti. [3]

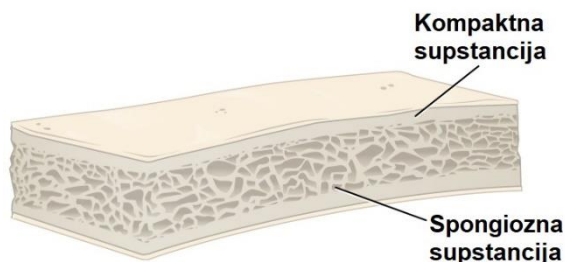
Kosti se sastoje od organske i anorganske supstancije. Glavni anorganski dio čine kalcijev i magnezijev fosfat, kalcijev karbonat i kalcijev fluorid, a organski dio čine fibrile i stanice. Dovođenjem topline na kost, odstranjuje se organski dio, te kost postaje krhka, dok otapanjem u 5% solnoj kiselini iz kosti se odstranjuje anorganski dio, te kost postaje elastična, uz smanjenje čvrstoće. Samo kombinacija organskog i anorganskog dijela kosti daju potrebnu čvrstoću i elastičnost. [3]

Koštana supstancija dolazi u dva oblika, i to kao kompaktna i kao spongiozna. Kod dugih kosti epifize se uglavnom sastoje od spongiozne supstancije, koja je presvučena tankim slojem kompakte, dok se dijafize sastoje uglavnom od kompaktne supstancije koja okružuje šupljinu ispunjenu koštanom moždinom. [3]



Slika 4. Duga kost

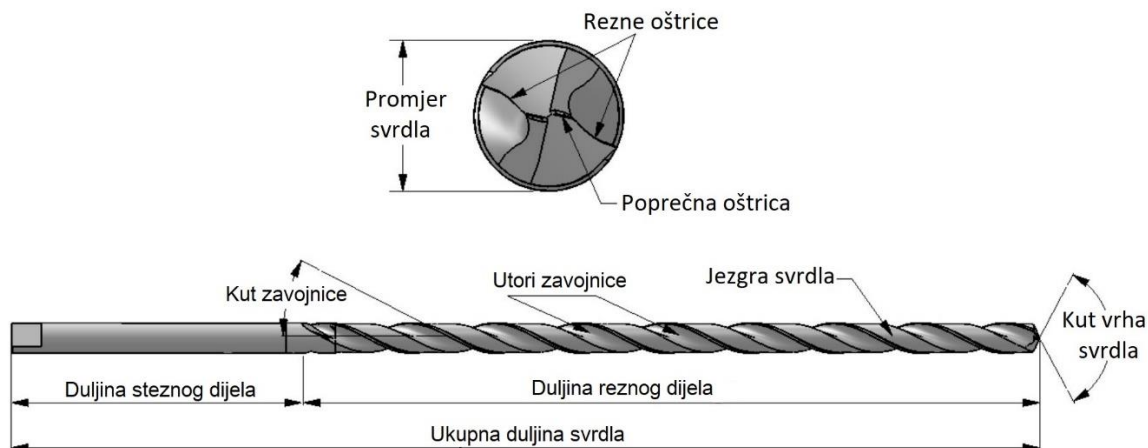
Pločaste kosti građene su tako da imaju s obje strane tanki sloj kompakte, a između njih se nalazi spongioza. Na lubanjskome svodu unutrašnja ploča kompaktne supstancije, zato jer je lomna, naziva se staklasta ploča, a spongiozna supstancija zove se diploë. [3]



Slika 5. Pločasta kost

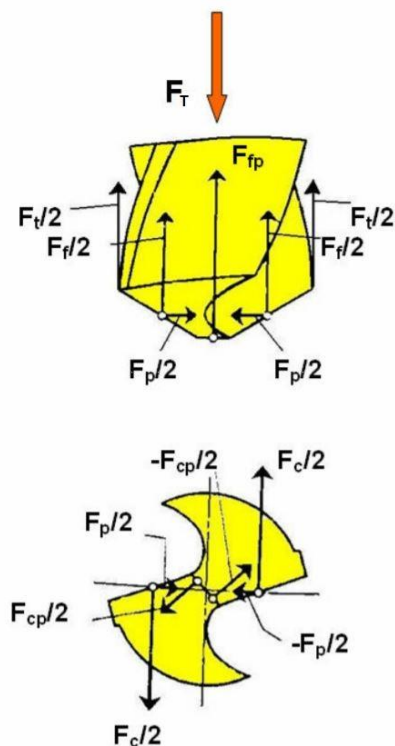
3 BUŠENJE

Bušenje je postupak obrade odvajanjem čestica kod koje alat vrši glavno gibanje, rotaciju, i posmično gibanje, translaciju, a služi za izradu rupa i provrta. Alat kojim se vrši bušenje je svrdlo, definirane geometrije reznog dijela, [4] slika 6.



Slika 6. Vijčano svrdlo [12]

Svrdlo materijal odvaja dvijema reznim oštricama i poprečnom oštricom, no udio odvojenog materijala poprečnom oštricom zanemariv je u usporedbi sa odvojenim materijalom reznim oštricama. Štoviše, poprečna oštrica otežava sami proces bušenja.



Slika 7. Svrdlo oslobođeno veza[4]

Na slici 7. prikazano je svrdlo oslobođeno veza sa naznačenim silama koje se javljaju prilikom procesa bušenja. Na dvije glavne oštrice djeluju sljedeće sile:

- Dvije jednake glavne sile rezanja $F_c/2$, ali suprotno usmjerne
- Dvije jednake i isto usmjerene posmične sile $F_f/2$
- Dvije jednake pasivne sile $F_p/2$, ali suprotno usmjerene

Dok na poprečnu oštricu djeluju:

- Dvije jednake, suprotno usmjerene glavne sile rezanja $F_{cp}/2$
- Posmična sila F_{fp}

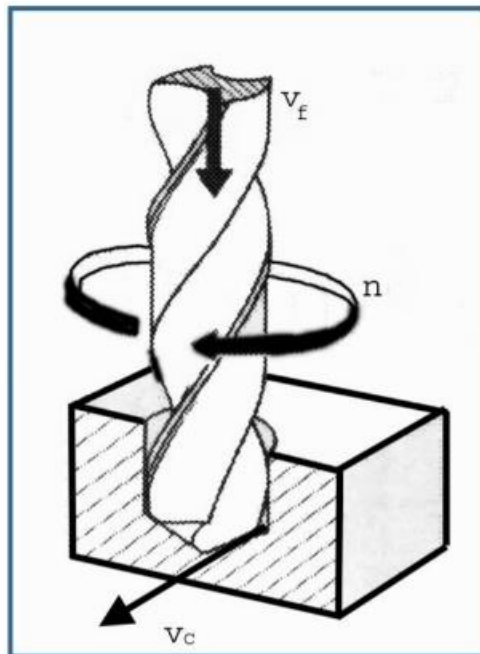
Uz sve navedene sile prilikom procesa bušenja na svrdlo djeluje i sila trenja F_t , koja se javlja na mjestu dodira svrdla, obradka i odvojene čestice. Spomenute sile koje djeluju u aksijalnome smjeru svrdla savladava aksijalna sila bušenja, tj. sila pritiska:

$$F_T = F_f + F_{fp} + F_t \quad (1)$$

Istraživanja su pokazala da sila pritiska F_T sa svojih 50% kapaciteta savladava posmičnu silu poprečne oštrice F_{fp} , 45% sile F_T savladava posmičnu silu rezne oštrice F_f , dok svega 5% otpada na sile trenja F_t . Uslijed djelovanja tangencijalnih sila na reznim oštricama i na poprečnoj oštrici javlja se moment T kojim je opterećeno svrdlo. Glavna sila rezanja reznih oštrica F_c , glavna sila rezanja poprečne oštrice F_{cp} i sile trenja F_t sudjeluju u generiranju toga okretnoga momenta. No istraživanja su pokazala da 80% okretnoga momenta svrdlo troši na savladavanje momenta uzrokovanog glavnom silom rezanja reznih oštrica, dok po 10% otpada na momente uzrokovane glavnom silom rezanja poprečne oštrice i silama trenja. Stoga se posljednja dva okretna momenta u većini proračuna zanemaruju, te izraz za okretni moment svrdla glasi:

$$T = F_c \frac{D}{2} \quad (2)$$

Na slici 8. nalazi se shematski prikaz bušenja gdje je v_f posmična brzina, v_c brzina rezanja, dok je n broj okretaja, odnosno rotacijska brzina svrdla.



Slika 8. Shematski prikaz bušenja vijčanim svrdlom [4]

Posmična brzina ili posmak u jedinici vremena računa se iz poznatog posmaka f i broja okretaja svrdla n :

$$v_f = 2f * n \quad (3)$$

Posmak f (mm/okr) definira se kao aksijalni pomak koji svrdlo napravi tijekom jednog punog okretaja.

Brzina rezanja v_c brzina je rezne oštrice svrdla. No sve točke na reznoj oštrici ne nalaze se na istoj udaljenosti od centra svrdla, te samim time i nemaju istu brzinu. U proračunima se pod brzinom rezanja uzima brzina rezne oštrice na obodu svrdla:

$$v_c = D * \pi * n \quad (4)$$

3.1 Svrkla u medicini

Važan uvijet za korištenje materijala u medicini je biokompatibilnost. U medicini se najčešće koriste nehrđajući čelici oznaka DIN X46Cr13 i AISI 420, sa 13% Cr, do 1% Si, do 1% Mn, te 0,42 do 0,5 C. [4]



Slika 9. Medicinska svrdla

Na slici 9. prikazana su svrdla koja se koriste prilikom operativnih zahvata. Za svrdla korištena u medicinske svrhe definirani su standardi DIN 1.4112 i AISI 440B. Za svrdla nije toliko neophodno da zadovoljavaju rigorozne standarde otpornosti na koroziju iz razloga što provode relativno kratko vrijeme u doticaju sa ljudskim organizmom. Puno veća važnost od otpornosti na koroziju daje se očuvanju reznih oštrica svrdla. Bilo da se radi o istrošenju oštrica prilikom velikoga broja bušenja ili o oštećenju oštrica uslijed sterilizacijskih postupaka opreme.[4]

4 MATEMATIČKI MODEL BUŠENJA

Matematički model je opis realnoga sustava matematičkim jednadžbama s ciljem razvoja i uporabe matematičkog modela za kasnije analize, projektiranja i optimiranja sustava za koji je model izrađen. Matematički model opisuje sustav pomoću skupova varijabli i jednadžbi, koje opisuju odnose među varijablama. Što je više veza među varijablama poznato, to će model biti točniji. Nekada je korisno u model ugraditi selektivne informacije na temelju intuicije, iskustva, stručnog mišljenja ili uvjerljivosti matematičkoga izraza. Većim brojem varijabli i većim brojem veza među njima model postaje složeniji i precizniji, ali isto tako i kompliciraniji za korištenje. Stoga je potrebno prilikom izrade matematičkoga modela sustava znati što nam je potrebnije, jednostavnos ili preciznost.

Pojednostavljeni matematički model procesa bušenja opisan je izrazom (5):

$$m\ddot{x} = F_k(x, \omega) + F_p \quad (5)$$

Prevedeno u prostor stanja, izraz (5) se može zapisati u sljedećem obliku:

$$\dot{x} = v \quad (6)$$

$$\dot{v} = \frac{1}{m}F_k(x, \omega) + \frac{1}{m}F_p \quad (7)$$

Gdje su :

- x – translacijski pomak svrdla [m]
- v – translacijska brzina svrdla – posmična brzina [m/s]
- ω – rotacijska brzina svrdla [rad/s]
- F_p – translacijska/aksijalna sila pritiska svrdla [N]
- F_k – sila otpora bušenog materijala [N]
- m – masa [kg]

Pretpostavlja se da je sila otpora F_k , barijera širine jednake debljini bušenog materijala i visine koja je obrnuto proporcionalna rotacijskoj brzini svrdla. Obrnuta proporcionalnost modelirana je opadajućom eksponencijalnom funkcijom:

$$F_k(x, \omega) = F_0 e^{-k\omega} f(x) \quad (8)$$

Gdje su:

- F_0 – sila barijere koja je potrebna da se probije materijal, bez rotacijske brzine svrdla.
- k – koeficijent brzine opadanja sile otpora
- $f(x)$ – jedinična barijera

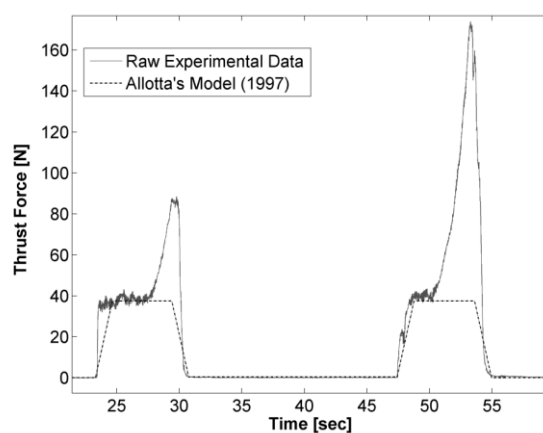
Sila F_0 puno je veća od sile F_p , no porastom rotacijske brzine svrdla ta razlika se smanjuje. Prilikom određene kritične rotacijske brzine svrdla ω_c njihovi iznosi postaju jednaki:

$$F_0 e^{-k\omega_c} = F_p ; \omega = \omega_c \quad (9)$$

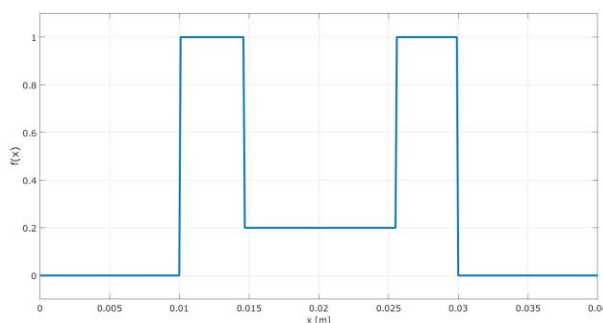
Daljnim porastom rotacijske brzine svrdla, sila pritiska postaje veća od sile otpora:

$$F_0 e^{-k\omega} < F_p ; \omega > \omega_c \quad (10)$$

Funkcija $f(x)$ je jedinična barijera, slika 11., širine jednake debljini bušenog materijala. Za slučaj bušenja kosti, jedinična barijera je oblikovana prema stvarnim očitanim silama pritiska prilikom bušenja kosti, slika 10.



Slika 10. Očitane sile pritiska svrdla prilikom bušenja kosti [12]



Slika 11. Jedinična barijera kosti

5 LABORATORIJSKI POSTAV

Laboratorijski postav za bušenje sastoji se od nekoliko sustava:

- Sustava za očitavanje sile i momenta ATI Net F/T sa mjernim osjetnikom „Gamma“
- Robotski sustav Kuka Lightweight Robot LWR4+
- Sustav za medicinsko bušenje proizvođača Medtronic
- Računalo (PC)



Slika 12. Laboratorijski postav za bušenje

Na slici 12. prikazan je laboratorijski postav za bušenje. Unutar ovoga poglavlja opisane su osnove računalne komunikacije, uz detaljniji opis onih segmenata koji su korišteni u ovome radu. Također u nastavku su opisani gore navedeni sustavi i prezentirane su njihove tehničke specifikacije.

5.1 Komunikacija

"Računalne mreže su skupovi računala povezanih komunikacijskim sklopovljem i odgovarajućom programskom podrškom." [1]

U definiciji koja opisuje računalne mreže spomenuti pojam komunikacijsko sklopovlje uključuje mrežna sučelja na računalima, usmjerivače i provodnike. Dok pojam programska podrška podrazumjeva operacijske sustave i mrežne uslužne programe koji omogućavaju prijenos i interpretaciju podataka putem komunikacijskog sklopovlja. Računalne mreže sastoje se od čvorova, koji mogu biti računala ili druge računalne mreže. S obzirom da čvorovi općenito mogu poticati od različitih proizvođača, da mogu koristiti različite procesore, operacijske sustave i uslužne programe, potrebno je definirati sustave zajedničkih pravila - protokole. Komunikacijski protokol je skup jednoznačno određenih pravila za razmjenu informacija između dva čvora na mreži koji uključuje sintaksu informacije, semantiku informacije te pravila za razmjenu informacije. [5]

5.1.1 OSI referentni model

Od strane Međunarodne organizacije za standarde (eng. *ISO - International Organization of Standards*) usvojen je OSI (eng. *Open System Interconnection - otvoreni sustav povezivanja*) referentni model za razvoj standardnih mrežnih protokola. OSI referentni model je podjeljen u sedam razina, gdje svaka razina opisuje skup povezanih funkcija koje omogućuju jedan dio računalne komunikacije.

7	Application – Razina primjene
6	Presentation – Razina prezentacije
5	Session – Razina sastanka\sustreta
4	Transport – Prijenosna razina
3	Network – Mrežna razina
2	Data Link – Podatkovna razina
1	Physical – Fizička razina

Slika 13. OSI referentni model

OSI referentni model predstavlja samo model procesa komuniciranja, te nudi predložak za formiranje konkretnih mrežnih protokola. OSI referentni model ne definira konkretne komunikacijske protokole već definira razine i funkcije na pojedinim razinama koje bi protokoli trebali zadovoljavati. Razine unutar OSI referentnog modela komuniciraju samo

sa prvim razinama poviše i prvim razinama ispod sebe. Svaka razina ovisi o funkcionalnosti koju pruža razina ispod nje. [5]

Fizička razina (eng. *Physical layer*) definira mehanička, električka, funkcionalna i proceduralna svojstva medija za prijenos. Odnosno definira dimenzije priključaka i raspored pinova (mehanička svojstva); dozvoljene napone (električka svojstva); značenje pojedinih signala (funkcionalna svojstva); dozvoljeni redoslijed signala (proceduralna svojstva). Osnovna funkcija fizičke razine je prijenos niza bitova. [5]

Podatkovna razina (eng. *Data Link layer*) mora osigurati konzistenciju prijenosa podataka. Identifikacijom okvira, čija osnovna jedinica može biti oktet ili bit, provjerava se stanje kontrolnih polja kod prijema. Okviri se dijele na adresno polje, kontrolno polje, podatkovno polje i kontrolno polje grešaka. Adresno polje definira primaoca paketa. Kontrolno polje sadrži podatak o vrsti paketa. Dok se u podatkovnom polju nalaze podaci koji su predmet komunikacije. Na temelju podataka iz kontrolnog polja grešaka na podatkovnoj razini se ispituje valjanost paketa i ako je došlo do greške u prijenosu može se zatražiti ponovni prijenos paketa. [5]

Mrežna razina (eng. *Network layer*) osigurava usmjeravanje (eng. *routing*) između udaljenih računala kroz mrežu. Ako dva računala pripadaju različitim lokalnim mrežama između njih moraju postojati mrežni uređaji koji ih povezuju. Isti uređaji mogu povezivati i mnoštvo drugih mreža, usmjeravajući pakete na odredišta. S obzirom da paketi mogu stizati iz mreža različitih topologija (npr. Token Ring i Ethernet), koji imaju različite oblike adresiranja, na mrežnoj se razini mora osigurati jedinstveni mehanizam adresiranja. [5]

Prijenosna razina (eng. *Transport layer*) mora osigurati prijenos podataka između komunikacijskih programa na udaljenim računalima, za razliku od prethodnih razina koje definiraju protokole za uspostavljanje komunikacije između računala, mrežnih sučelja i/ili računalnih mreža. Budući da više programa ili komunikacijskih procesa može ostvarivati vezu putem iste mrežne adrese, na ovoj se razini definira prijenosna adresa koja nadopunjuje mrežnu adresu s jedinstvenim identifikacijskim brojem programa (*socket* ili *port number*). [5]

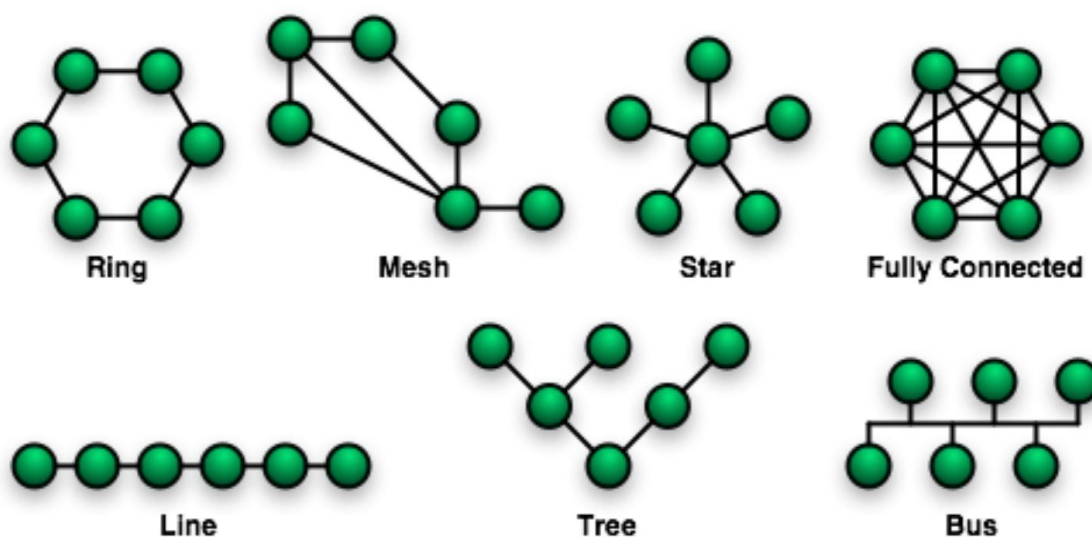
Razina sastanka\susreta (eng. *Session layer*) proširuje funkcije prijenosne razine, upravljajući procesom razmjene podataka između udaljenih procesa. Komunikacija (sastanak) između dva udaljena procesa se može odvijati kao dvosmjerni ili jednosmjerni dijalog, a komuniciranje može biti istovremeno ili naizmjenično. [5]

Razina prezentacije (eng. *Presentation layer*) upravlja prikazom podataka. Brine se o poslovima kao što su sažimanje i raspakiravanje podataka, pretvorbe grafičkih prikaza i općenito pretvorbe podataka različitih oblika koji se prenose. Ova razina mora osigurati ispravni prijem podataka neovisno o međusobnim razlikama udaljenih računala. [5]

Razina primjene (eng. *Application layer*) je najviša razina u OSI referentnom modelu i predstavlja mrežno sučelje prema korisniku. Prethodne razine tek su u funkciji primjene. Korisnik ne mora biti svjestan procesa koji se odvijaju na tim prethodnim razinama. Na razini primjene definiraju se usluge i protokoli po kojima komuniciraju mrežni programi. [5]

5.1.2 Topologija računalnih mreža

Topologija mreža je geometrijsko uređenje čvorova i veza. Ovisno o veličini, namjeni i tehnologiji mreže, kao i udaljenostima povezivanja. Tehnologija mreže podrazumijeva komunikacijsku tehnologiju i protokol. Mrežna topologija može biti fizička ili logička (prva ili druga razina OSI). Fizička topologija predstavlja nacrt fizičkog rasporeda čvorova u mreži, dok logička topologija predstavlja logički tok podataka između čvorova te mreže. Fizička topologija ne mora ujedno biti i logička topologija.



Slika 14. Topologija računalnih mreža [6]

Prema fizičkoj mrežnoj topologiji mreže možemo podijeliti na: [6]

- Prstenastu (eng. *Ring*) - svaki član je povezan sa točno 2 druga člana i zajedno formiraju kružni tok signala
- Mrežnu (eng. *Mesh*) - može biti sa djelomično ili potpuno povezanim članovima. Kod potpune povezanosti svaki član unutar mreže je povezan sa svakim drugim
- Zvezdastu (eng. *Star*) – svi članovi mreže su spojeni različitim vodovima na središnji čvor koji djeluje kao poveznica ostalim članovima.
- Hijerarhijsku (eng. *Tree*) - mreža u kojoj postoji centralni čvor koji je povezan sa niže pozicioniranim čvorovima drugog nivoa, a oni dalje mogu biti povezani sa još niže pozicioniranim čvorovima trećeg nivoa, itd...
- Sabirničku (eng. *Bus*) – svi članovi mreže su spojeni na jedinstveni, zajednički vod - sabirnicu

Prema funkcionalnim povezanostima između pojedinih elemenata, računalne mreže možemo podijeliti na: [6]

- Aktivne mreže (eng. *Active networking*) – predstavlja komunikacijski model koji omogućuje paketima koji prolaze kroz telekomunikacijsku mrežu dinamičku promjenu rada te mreže.
- Poslužitelj – korisnik (eng. *Server - Client*) - računalna arhitektura u kojoj su razdvojene uloge korisnika i poslužitelja. Uloga korisnika je da uputi zahtjev za određenom uslugom, a uloga poslužitelja je da tražene podatke dostavi. Svaki korisnik i svaki poslužitelj u ovoj arhitekturi predstavljaju jednog člana ili jedan čvor mreže.
- *Peer-to-peer* - računalna arhitektura u kojoj su svi članovi mreže ravnopravni. Ne postoji podjela na korisnike i poslužitelje. Svi članovi su istovremeno korisnici i poslužitelji.

Prema veličini računalne mreže se dijele na: [6]

- *Personal Area Network* – PAN – mreža za povezivanje uređaja (telefon, dlanovnik,...) na računalo koji obično služi jednome korisniku. Prostire se najviše unutar nekoliko metara.
- *Local Area Network* – LAN – računalna mreža u kojoj su računala smještena na manjim udaljenostima (unutar doma, ureda, ili blisko smještenih zgrada). Značajka lokalnih mreža je da su one najčešće u cijelosti u vlasništvu i pod upravljanjem onih koji ih koriste (osobno, vlasništvo tvrtke ili institucije), tako da je prijenos podataka putem njih za korisnike besplatan. Značajno je i da su moguće jako velike brzine prijenosa podataka (Gbps).
- *Metropolitan Area Network* – MAN – mreža u kojoj su računala smještena na nešto većim udaljenostima od onih u lokalnim mrežama. Najčešće pokriva područje jednog dijela ili cijelog grada. Mogu biti u vlasništvu neke organizacije ili više njih. Brzine prijenosa su obično manje nego u lokalnim mrežama.
- *Wide Area Network* – WAN – mreža koja se proteže preko granica grada, regije ili države. Za povezivanje se koriste usmjerivači i javne komunikacijske veze. Značajka WAN mreža je da nisu u vlasništvu osoba ili organizacija koje ih koriste i prijenos podataka preko njih je ograničen prema brzini, količini i cijeni. Potrebno je platiti za korištenje komunikacijskih veza. U odnosu na lokalne mreže brzine su dosta ograničene.

5.1.3 Ethernet

Ethernet predstavlja najrašireniji skup mrežnih računalnih tehnologija primjenjenih unutar lokalnih mreža – LAN. Na fizičkoj razini OSI referentnog modela Ethernet definira raspored ožičenja, te vrste i razine signala za prijenos podataka. Na prijenosnoj razini Ethernet definira način pristupa mediju za prijenos podataka (MAC – *Media Access Control*) i definira zajednički adresni format.

Ethernet je razvijen sredinom 70-tih unutar tvrtke Xerox. U početku je imao brzinu prijenosa od 3Mbps i koristio je 8-bitno adresiranje. Današnji standardi propisuju brzine od 1Gbps i 48 bitno adresiranje (MAC adresa). U početku je kao standardni medij za prijenos podataka korišten koaksijalni kabel, dok se danas standardno koristi neki od oblika UTP (eng. *Unshielded Twisted Pair*) kabela. Pored navedenih, kao mediji se još koriste optička vlakna.[6]



Slika 15. Provodnici definirani Ethernet protokolom [6]

Ethernet je temeljen na sabirnici, dakle, sva računala na lokalnoj razini dijele isti provodnik. S obzirom da se samo jedan paket - okvir podataka može emitirati istovremeno, Ethernet koristi poseban mehanizam CSMA/CD koji sprečava računala da međusobno interferiraju. Naziv CSMA/CD dolazi od *Carrier Sense Multiple Access with Collision Detection*. Što u prijevodu znači "slušaj prije nego govoriš". *Carrier Sensing* označava mehanizam slušanja. Kada računalo želi emitirati pakete kroz Ethernet mrežu, prvo mora ispitati da li je mreža zauzeta slanjem podataka od nekog drugog računala. Ako jest, Ethernet uređaj računala će nakratko pričekati te ponovno ispitati stanje zauzetosti mreže. Ako nije registriran nikakav promet na mreži, započinje emitiranje podataka.

Multiple Access znači da više umreženih računala može koristiti isti prijenosni medij. *Collision Detection* definira ponašanje računala ako se dogodi da dva ili više računala započnu emitiranje podataka u isto vrijeme. Tijekom emitiranja podataka Ethernet uređaj nastavlja osluškivati mrežu. Ako se u međuvremenu pojavi paket nekog drugog računala nastaje kolizija i računalo prekida emitiranje. Slijedi odašiljanje signala o zakrčenosti, koji upozorava ostala računala u mreži da podaci koji nailaze kolidiraju. Potom sva računala u lokalnoj mreži zaustavljaju emitiranje za slučajni iznos vremena čekanja. Na taj način vjerojatnost da će opet dva računala započeti istovremeno emitiranje značajno se smanjuje. [5]

S obzirom da je Ethernet zasnovan na slučajnom pristupu, kapacitet prijenosa podataka nije u pravilu kontinuiran. Kada se radi o prijenosu vremenski nepovezanih podataka onda ta značajka ne dolazi do izražaja. Ali, ako se radi npr. o video ili audio informacijama, svako kašnjenje, nastalo zbog zauzetosti mreže ili kolizija, odrazit će se izravno na kvaliteti reprodukcije istih. [5]

5.1.4 DoD mrežni model – TCP/IP porodica protokola

DoD je također višerazinski hijerarhijski model umrežavanja računala, te prethodnik OSI referentnog modela. Za razliku od OSI referentnog modela koji ima 7 razina, DoD ima 4 razine. Samo ime mrežnog modela DoD izvedeno je iz naziva *Department of Defense* (Ministarstvo obrane SAD-a), ustanove koja je vodila projekt *DARPA*, iz kojeg je ovaj model proizašao.

Porodica protokola TCP/IP (eng. *Transmission Control Protocol / Internet Protocol*) dio je DoD mrežnog modela, te podrazumijeva porodicu komunikacijskih protokola. U TCP/IP porodicu protokola ubrajaju se IP protokol, TCP protokol, UDP protokol, ICMP protokol kao i mnogi drugi protokoli. Pod tim imenom obuhvaćene su i mrežne usluge ali i uslužni programi koji ostvaruju mrežne usluge.

Uspoređujući OSI referentni model i TCP/IP važno je naglasiti da OSI predstavlja teoretski model dok TCP/IP predstavlja skup konkretnih komunikacijskih protokola koji čine komunikacijsku osnovu za mnoge računalne mreže. [5]

OSI model	DoD model	TCP/IP model
Application – Razina primjene	Application – Razina primjene	Telenet , FTP, SMTP, DNS, RIP, SNMP
Presentation – Razina prezentacije		
Session – Razina sastanka\ sustreta		
Transport – Prijenosna razina	Transport – Prijenosna razina	TCP, UDP
Network – Mrežna razina	Internet – Internet razina	IP, ARP, IGMP, ICMP
Data Link – Podatkovna razina	Network interface – Razina mrežnog sučelja	Ethernet, Token Ring, Frame Relay, ATM
Physical – Fizička razina		

Slika 16. Usporedba OSI, DoD i TCP/IP modela

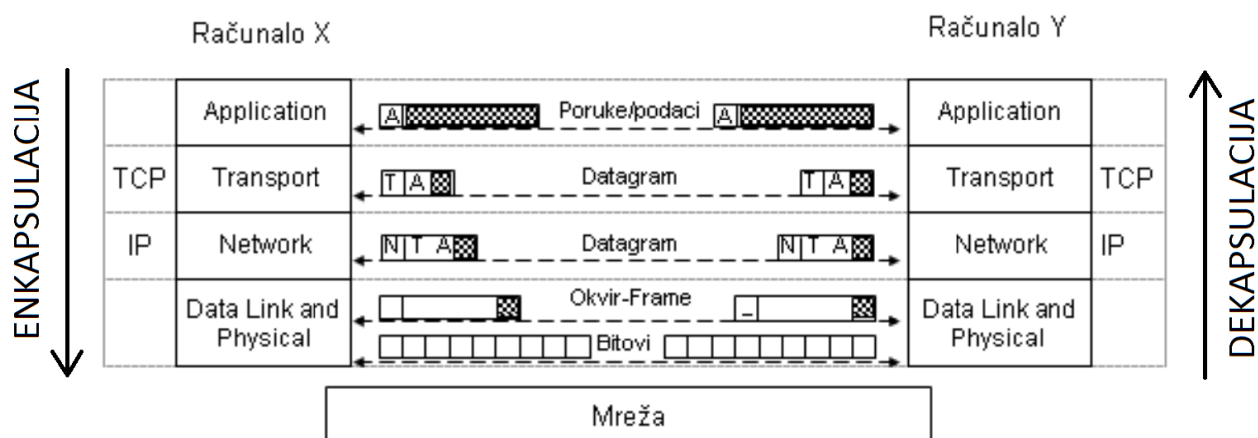
Razina mrežnog sučelja (eng. *Network interface*) rješava probleme pristupanja fizičkom mediju koji se koristi za prijenos. Najčešće je to programski veznik koji komunicira s Ethernet karticom ili mrežnim poslužiocem. Odgovara prvoj i drugoj razini OSI referentnog modela (*Physical* i *Data Link*).

Internet razina (eng. *Internet layer*) je odgovorna za povezivanje logičkih adresa s fizičkim mrežnim sučeljem, odnosno fizičkim adresama. Implementirana je u obliku internet protokola (IP) koji radi s IP paketima (datagramima) i brine za njihovo usmjeravanje. Osnovna zadaća ove razine je ostvarivanje komunikacije između dva udaljena računala.

Prijenosna razina (eng. *Transport layer*) odgovara četvrtoj OSI razini. Ostvaruje komunikaciju između mrežnih programa udaljenih računala. Programi na udaljenim računalima se adresiraju kao *port number*. Podaci se prenose na principu paketne komunikacije, odnosno dijeleći se na datagrame koji se na odredištu ponovno sastavljaju. Važniji protokoli prijenosne razine su TCP (Transmission Control Protocol) i UDP (User Datagram Protocol)

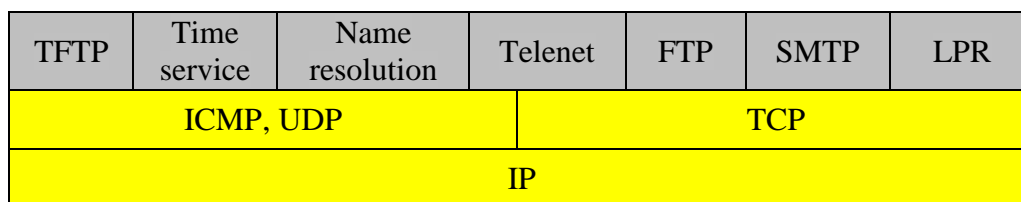
Razina primjene (eng. *Application layer*) obuhvaća razinu sastanka, prezentacijsku i aplikacijsku razinu OSI referentnog modela. Aplikacija je bilo koji proces koji se događa iznad prijenosne razine TCP/IP komunikacijske arhitekture. Neki od aplikacijskih protokola su: SMTP, HTTP, FTP, DNS itd.

TCP/IP komunikacija između spojenih računala odvija se postupkom enkapsulacije. Enkapsulacija podrazumijeva postupak koji podatkovne pakete više razine umeće u pakete niže razine. Svaka razina dodaje svoje zaglavlje na podatke koji se prenose. Podaci najniže razine putuju mrežom na drugo računalo gdje se zatim odvija obrnuti proces koji se naziva dekapsulacija.



Slika 17. Enkapsulacija / Dekapsulacija [5]

Porodica TCP/IP protokola može se podijeliti u dvije skupine. Nižu skupinu čine protokoli razine interneta i transportne razine DoD modela, kao npr. IP, TCP i UDP protokoli. Dok se u višu skupinu protokola ubrajaju protokoli sa razine primjene, kao npr. FTP, Telenet itd.

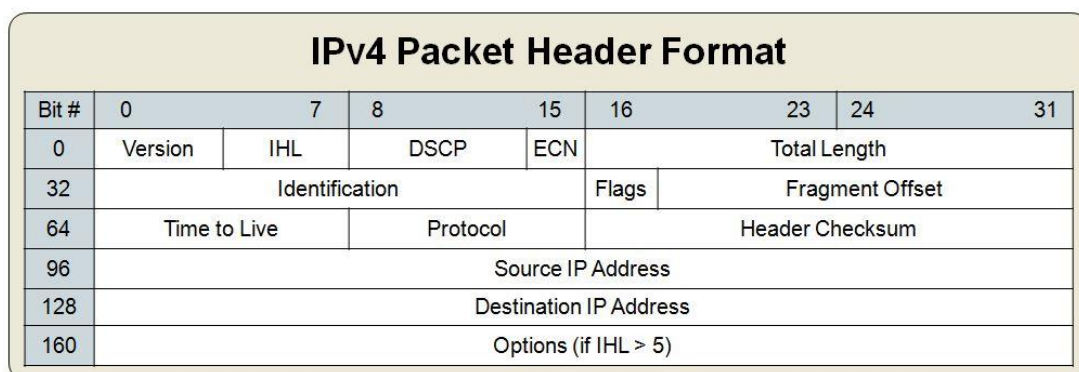


Slika 18. Protokoli TCP/IP modela

5.1.5 Internet Protocol - IP

IP je temeljni protokol internet razine TCP/IP arhitekture, a koriste ga protokoli svih viših razina. To je bespojni protokol, što znači da se predajnik i prijemnik ne dogovaraju o početku ili završetku prijenosa podataka, nego predajnik pošalje paket i nema potvrde primitka paketa. Osnovne funkcije IP protokola su: [3]

- Definiranje sheme adresiranja na internetu
- Definiranje IP paketa
- Prosljeđivanje podataka između razine pristupa mreži i prijenosne razine
- Fragmentacija i sastavljanje paketa



Slika 19. Zaglavlje IPv4 protokola

Na slici 19. je prikazano zaglavlje Ipv4 protokola, a pojednina polja imaju sljedeća značenja:

- **Version** - verzija IP protokola, određuje format zaglavlja. [7]
- **Internet Header Length (IHL)** - duljina IP zaglavlja u 32-bitnim riječima, omogućava određivanje početka podataka; minimalna duljina ispravnog zaglavlja je 5.
- **Type of Service** - tip usluge, omogućava usmjernicima različit tretman pojedinih paketa u cilju postizanja zadovoljavajuće kvalitete usluge (QoS), a s obzirom na dopušteno kašnjenje, količinu prometa i zahtijevanu pouzdanost.
- **Total Length** - ukupna duljina IP paketa u oktetima, uključujući IP zaglavlje i podatke; najveća duljina paketa je 65 535 okteta

- **Identification** - identifikator paketa, važan je pri povezivanju svih fragmenata u paket.
- **Flags** - kontrolne zastavice, definiraju je li fragmentacija dopuštena i ako jest, ima li još fragmenata istog paketa.
- **Fragment Offset** - definira mjesto fragmenta u originalnom paketu, mjereno u jedinicama od 8 okteta (64 bita); odstupanje prvog fragmenta je nula.
- **Time to Live (TTL)** - maksimalno vrijeme života paketa u mreži, nakon čega se neisporučeni paket odbacuje; mjeri se u sekundama, čvor koji obrađuje paket umanjuje vrijednost za najmanje 1, a ako je vrijednost nula paket se odbacuje.
- **Protocol** - označava protokol više razine kojem se podaci prosljeđuju.
- **Header Checksum** - kontrolni zbroj zaglavlja; ponovno se obračunava i provjerava pri svakoj promjeni podataka u zaglavlju.
- **Source IP Address** - IP adresa predajnika paketa.
- **Destination Address** - IP adresa prijemnika paketa.
- **Options** - varijabilne duljine, mogućnost; sadrži kontrolne informacije o usmjeravanju

5.1.6 User Datagram Protocol - UDP

UDP pripada prijenosnoj razini TCP/IP modela. Predstavlja proširenje IP protokola, a osnovna zadaća mu je postavljanje neslijedne (eng. *connectionless*) komunikacije između udaljenih računala. S obzirom da prijenos nije slijedan u nekim slučajevima može uzrokovati duplicirajuću komunikaciju. Najčešće se upotrebljava za slanje manjih poruka minimalnim mehanizmom. Iako se iz ovakvog opisa koji se najčešće nalazi u literaturi može zaključiti da je to neki loš i nepouzdan protokol u praksi se na ovom protokolu baziraju mnogi važni viši protokoli. [7]

UDP Datagram Header Format								
Bit #	0	7	8	15	16	23	24	31
	Source Port			Destination Port				
	Length			Header and Data Checksum				

Slika 20. Zaglavlje UDP protokola

Značenja polja sa slike 20.: [7]

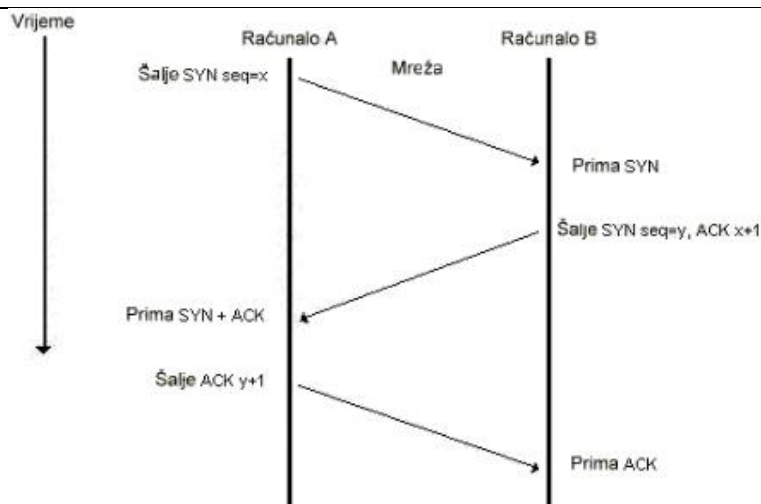
- **Source port** - izvorišna priključna točka usluge je polje koje se koristi po potrebi. Kada se koristi, označava priključnu točku procesa koji šalje podatke. Ako se polje ne koristi popuni se nulama.

- **Destination port** - odredišna priključna točka usluge.
- **Length** - duljina UDP datagrama u oktetima uključujući zaglavlje i podatke. Minimalna duljina UDP datagrama je 8 okteta.
- **Header and Data Checksum** - kontrolni zbroj zaglavlja, računa se na osnovu pseudo zaglavlja iz IP i UDP zaglavlja i podataka. Ako je polje ispunjeno nulama znači da predajnik nije računao zbroj; ako je zbroj jednak nuli, prenosi se kao niz jedinica.

5.1.7 Transmission Control Protocol - TCP

TCP pripada prijenosnoj razini TCP/IP modela. Za razliku od UDP protokola, omogućuje slijednu komunikaciju. To je "end-to-end" protokol koji povezuje dva procesa na udaljenim računalima, a osigurava sigurnu, neduplicirajuću isporuku okteta udaljenom korisniku. TCP predstavlja bazu za mnoge druge više protokole (Telnet, FTP) koji zahtijevaju pouzdan prijenos većeg broja okteta. Prilikom korištenja TCP usluga entiteti prolaze kroz tri faze: 1. Uspostava veze, 2. Razmjena podataka, 3. Prekid veze. [7]

Uspostava veze je proces koji se izvodi na jednom računalu želi uspostaviti vezu s procesom na nekom drugom računalu. Računalo koje traži uspostavu veze naziva se klijent, a drugo računalo se naziva poslužitelj. Klijentski proces informira klijentski TCP da želi uspostaviti vezu s poslužiteljem. Klijentsko računalo tada šalje poslužitelju prvi specijalni segment. Poslužitelj odgovara drugim specijalnim TCP segmentom i konačno klijent odgovara trećim specijalnim segmentom. Ova procedura se naziva *three-way handshake*, te je shematski prikazana na slici 21.: [7]



Slika 21. *Three-way handshake* – uspostava veze korištenjem TCP protokola [7]

Razmjena podataka - TCP protokoli razmjenjuju podatke u obliku segmenata. Segment se sastoji od zaglavlja koje ima 20 okteta za kojim slijedi nula ili više okteta podataka, a nastaje skupljanjem podataka od nekoliko upisivanja ili razbijanjem podataka od jednog upisivanja. Veličina segmenta je varijabilna.

TCP Segment Header Format						
Bit #	0	7	8	15	16	23 24 31
0	Source Port			Destination Port		
32	Sequence Number					
64	Acknowledgment Number					
96	Data Offset	Res	Flags		Window Size	
128	Header and Data Checksum			Urgent Pointer		
160...	Options					

Slika 22. *Zaglavlje TCP protokola*

Na slici 22. prikazan je format zaglavlja TCP protokola, a pojedina polja imaju sljedeća značenja: [7]

- **Source Port** - broj priključne točke usluge izvorišta.
- **Destination Port** - broj priključne točke usluge odredišta.
- **Sequence Number** - redni broj prvog okteta podataka u tom segmentu; ako je postavljena zastavica S (SYN), onda je to početni redni broj (ISN - Initial Sequence Number), a prvi oktet podataka ima broj ISN+1.
- **Acknowledgment Number** - broj potvrde; ako je postavljen A (ACK) bit, polje sadrži redni broj sljedećeg okteta kojeg primatelj očekuje.

- **Data Offset** - pomak podataka, pokazuje na početak podataka u TCP segmentu, izraženo u 32-bitnim riječima (TCP zaglavlje je uvijek višekratnik 32-bitne riječi).
- **Reserved** – polje je rezervirano za buduće potrebe; popunjeno je nulama.
- **Flags**
 - URG - indikator hitnih podataka
 - ACK - indikator paketa potvrde
 - PSH - inicira prosljeđivanje svih do tada neproslijeđenih podataka korisniku
 - RST - ponovna inicijalizacija veze
 - SYN - sinkronizacija rednih brojeva
 - FIN - izvorište više nema podataka za slanje
- **Window size** – prozor, označava koliko je okteta prijemnik spreman primiti
- **Header and Data Checksum**- kontrolni zbroj; računa se kao 16-bitni komplement jedinice komplementa zbroja svih 16-bitnih riječi u zaglavlju i podacima; pokriva i 96 bitova pseudozaglavlja koje sadrži izvorišnu i odredišnu adresu, protokol i duljinu TCP zaglavlja i podataka.
- **Urgent Pointer** - Pokazivač na redni broj okteta gdje se nalaze hitni podaci; polje se gleda jedino ako je postavljena zastavica URG.
- **Options** – može, a i ne mora biti uključeno; ako je uključeno, veličine je $x \times 8$ bita,

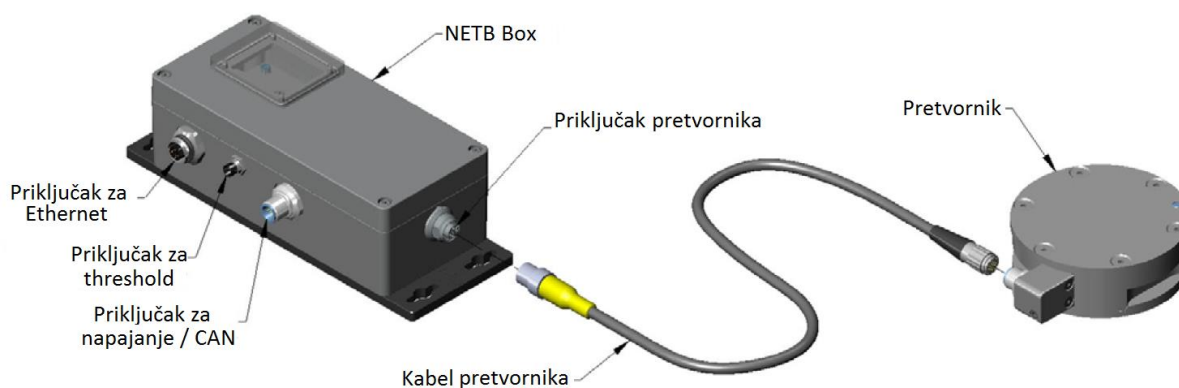
TCP i UDP protokoli pripadaju istoj razini komunikacije ali definiraju različite mogućnosti. Osnovne razlike između TCP protokola i UDP protokola mogu se sažeto prikazati u nekoliko točaka: [5]

- TCP dijeli (i ponovno sakuplja) veće poruke na IP datagrame.
- UDP radi s porukama koje stanu u jedan IP datagram
- TCP vodi računa o poslanim datagramima i može napraviti retransmisiju.
- UDP ne vodi računa o poslanim datagramima i ne može napraviti retransmisiju
- UDP je jednostavniji, UDP zaglavlje zauzima manje mjesta.

5.2 Net F/T

U ovome radu korišten je sustav Net F/T (eng. *Network Force/Torque Sensor System*) proizvođača ATI (*Industrial Automotion*). Sustav je namjenjen za simultano mjerenje sve tri komponente sile (F_x , F_y , F_z), te sve tri komponente momenta (T_x , T_y , T_z). Glavne komponente sustava su:

- Net F/T pretvornik (eng. *transducer*)
- NETB sučelje
- Net kabel



Slika 23. Net F/T sustav [4]

5.2.1 Net F/T Gamma pretvornik

Net F/T pretvornik je elektroničko-mehanički sklop osjetnika i samoga pretvornika. Osjetnik osjeća/mjeri stvarno fizičko opterećenje u obliku sile i momenta, dok pretvornik pretvara navedena opterećenja u električne signale. Osjetnik i pretvornik se nalaze u kućištu, čija robusna i kompaktna struktura štiti osjetljive elektroničke komponente.



Slika 24. Gamma pretvornik (SI-65-5)

U ovome radu korišten je Gamma pretvornik (SI-65-5). U tablici 1. su navedene osnovne karakteristike pretvornika:

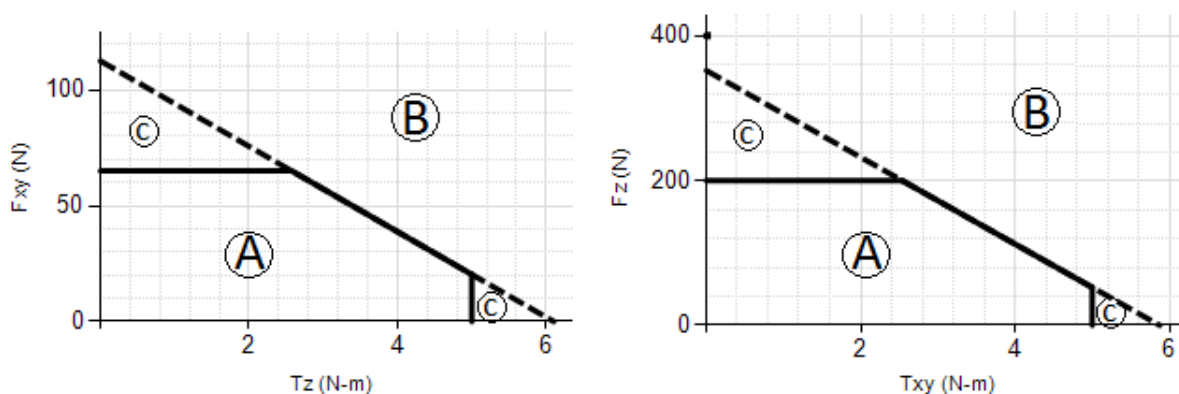
Tablica 1. Osnovne karakteristike Gamma pretvornika (SI-65-5)

Jednoosno preopterećenje	
F_x, F_y	± 1200 N
F_z	± 4100 N
T_x, T_y	± 79 Nm
T_z	± 82 Nm
Fizičke karakteristike	
Masa	0.255 kg
Promjer kućišta	75.4 mm
Visina kućišta	33.3 mm

Zbog mogućnosti mjerenja sve tri komponente sile i sve tri komponente momenta, ova vrsta pretvornika još je poznata i pod nazivima „više-osni senzor (pretvornik) sile i momenta“ i „šesto-osni senzor sile“. U tablici 2. su navedeni i na slici 25. prikazani mjerni opsezi i rezolucija pojedinih komponenata sile i momenta.

Tablica 2. Mjerni opsezi i rezolucija Gamma pretvornika (SI-65-5)

Mjerni opseg			Rezolucija		
F_x, F_y	F_z	T_x, T_y, T_z	F_x, F_y	F_z	T_x, T_y, T_z
65 N	200 N	5 Nm	0.0125 N	0.025 N	0.00075 Nm



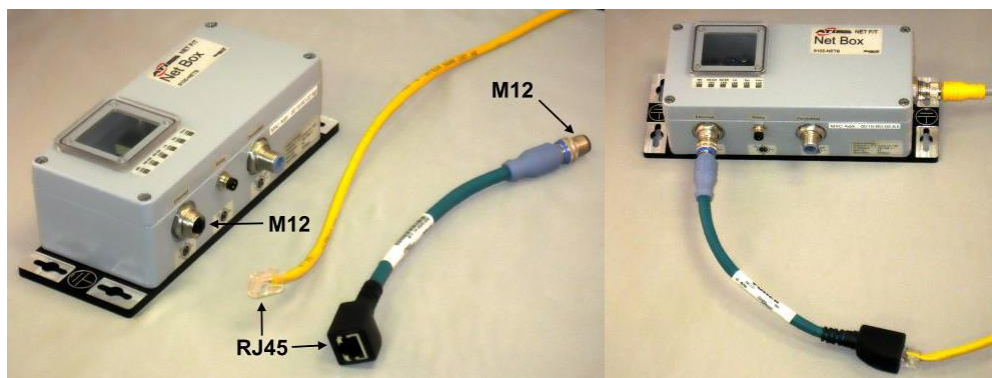
Slika 25. Mjerni opsezi Gamma pretvornika (SI-65-5) [8]

Na grafovima, slika 25., slovima su označena područja rada pretvornika: [8]

- A. Područje normalnog rada pretvornika. U ovome području očekuju se najtočniji rezultati mjerenja.
- B. Područje zasićenja. U ovome području bez obzira na vanjska opterećenja dobiveni rezultati mjerenja biti će jednaki vrijednostima zasićenja.
- C. Produljeno područje normalnoga rada. U ovome području možemo očekivati točne rezultate mjerenja, ali proizvođač ne daje garanciju na točnost dobivenih rezultata mjerenja za svih 6 komponenti sile i momenta.

5.2.2 NETB sučelje

NETB sučelje je elektronički sklop mrežnog sučelja i sustava napajanja, zaštićen robusnim kućištem. Glavna zadaća mrežnog sučelja je prikupljenje podataka sa Net F/T pretvornika, te slanje prikupljenih podataka drugoj opremi (npr. robot, PC, PLC). Ostvarivanje komunikacije sa ostalom opremom izvedivo je koristeći Ethernet, EtherNet/IP, CAN bus ili DeviceNet protokol.



Slika 26. NETB sučelje – fizičko spajanje UTP kabla [8]

Moguće su dvije izvedbe napajanja NETB sučelja. Prva izvedba je korištenje samostalnog izvora napajanja, dok je druga izvedba korištenje PoE (eng. *Power over Ethernet*). U ovome radu korišten je samostalni izvor napajanja (11V DC, 6W).

Fizičko spajanje računala (mrežna kartica) i NETB sučelja izvodi se UTP kablom sa RJ45 priključcima. Priključak RJ45 nije moguće izravno priključiti na NETB sučelje, te je stoga potrebno koristiti adapter koji omogućuje spajanje spomenutoga priključka i priključka M12, kako je prikazano na slici 26.

5.2.3 Komunikacija Net F/T – računalu

Postavkama Net F/T sustava pristupa se preko web preglednika na računalu. U tu svrhu potrebno je prvo fizički spojiti sustava sa računalom, kako je opisano u poglavlju 5.2.2 *NETB sučelje*, te zatim podesiti parametre LAN-a (eng. *Local Area Connection*) na računalu. Prilikom prvoga povezivanja, ukoliko nam nije poznata IP adresa Net F/T sustava, potrebno je fizički otvoriti NETB sučelje, te privremeno podesiti DIP prekidače (eng. *dual in-line package*). Određenim položajem (ON/OFF) DIP prekidača koji se nalazi unutar NETB sučelja možemo definirati IP adresu sustava.



Slika 27. NETB sučelje – fizičko definiranje IP adrese

Kako je prikazan na slici 27., DIP prekidač br. 9 potrebno je postaviti u položaj „ON“, kako bih IP adresa Net F/T sustava bila postavljena na vrijednost „192.168.1.1“. Zatim se pristupi postavkama Net F/T sustava preko navedene IP adrese, te se postavi nova željena statična IP adresa. Vraćanjem DIP prekidača br. 9 u „OFF“ položaj, aktivirana je statična IP adresa.

U postavkama Net F/T sustava moguće je podešavati niz parametara sustava. Na slici 28. su prikazani konkretni parametri i podešenja koja su korištena u sklopu ovoga rada za ostvarivanje komunikacije između Net F/T sustava i PC-a.

System Status: Healthy

Communications

These settings control how the Net F/T communicates with external equipment. Most settings require the Net F/T to be powered off and then back on before they take effect.

Values are not stored unless the *Apply* button is clicked.

Ethernet Network Settings

DIP switch 9 must be off to enable IP Address Mode. If DIP switch 9 is on then the IP address is set to 192.168.1.1 regardless of the IP Address Mode settings below. A LAN connection must be present at power up for DHCP to function. If DHCP is enabled and no DHCP server is found then the static IP address will be used.

IP Address Mode: DHCP Static IP

see above note regarding DIP switch 9

Static IP Address:

Static IP Subnet Mask:

Static IP Default Gateway:

EtherNet/IP Protocol: Enabled Disabled

EtherNet/IP O2T Data: Enabled Disabled

Ethernet MAC Address: 00:16:BD:00:13:07

Fieldbus Module Settings

The fieldbus module is not supported in this product.

Fieldbus Module Firmware:

Fieldbus Module Enabled: Enabled Disabled

CAN Network Settings

If power is not provided to the Pwr/CAN connector, then CAN Bus Base Address, DeviceNet MAC ID, and Baud Rate are not correctly reported and communications over the Pwr/CAN connector are not available.

Protocol: CAN Bus DeviceNet with Quick Connect Enabled

CAN Bus Base Address: 432
set by DIP switches 1 to 6

DeviceNet MAC ID: 54
set by DIP switches 1 to 6 (inaccurate without DeviceNet connection)

Baud Rate: 500 kHz
set by DIP switches 7 and 8

Raw Data Transfer (RDT) Settings

RDT data is routed through the local network and does not get routed through the default gateway.

RDT Interface: Enabled Disabled
demo application requires RDT to be enabled

RDT Output Rate (1 to 7000): Hz
value may be rounded up; see manual for details

RDT Buffer Size (1 to 40):

Multi-Unit Synchronization: Enabled Disabled

Multi-Unit Id (1 to 9):

Slika 28. Komunikacije postavke Net F/T sustava

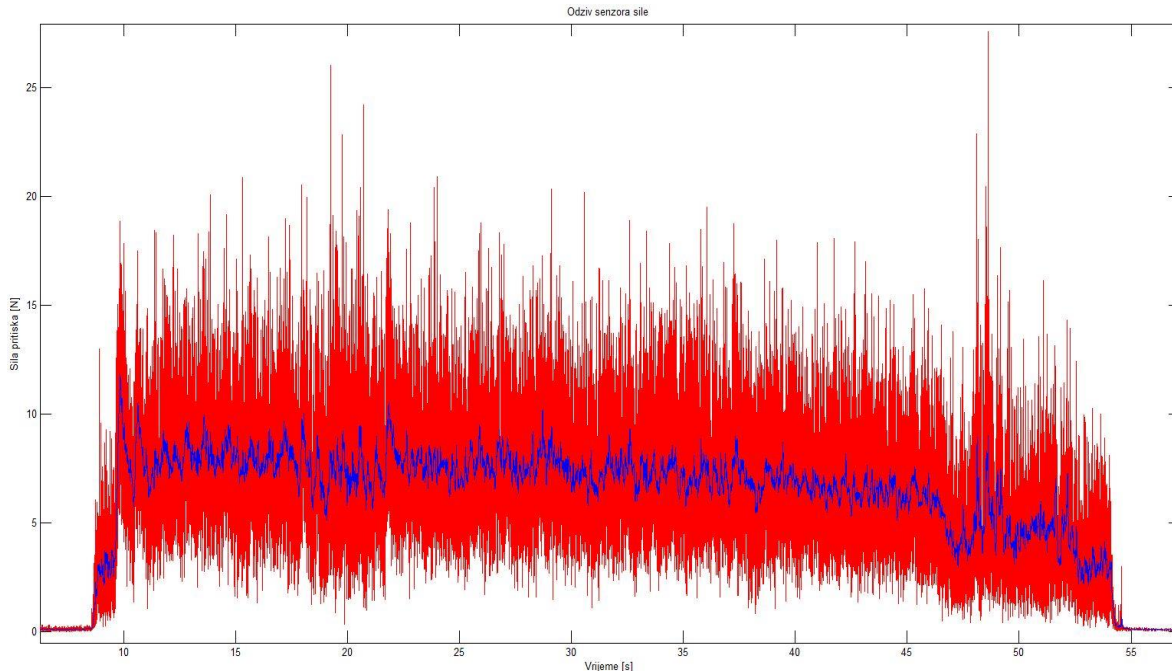
Za potrebe ovoga rada komunikacija PC-a i Net F/T-a ostvarena je korištenjem tri protokola. Prema TCP/IP modelu, na razini mrežnoga sučelja korišten je Ethernet protokol. Na internet razini IP protokol, dok je na prijenosnoj razini korišten UDP protokol, tj. njegova inačica RDT (eng. *Raw Data Transfer*). RDT je metoda brzoga prikupljanja podataka temeljena na UDP protokolu. Ovom metodom je moguće slati podatke velikom frekvencijom, do 7000Hz.

Komunikacija je jednosmjerna, podaci se šalju sa Net F/T-a na računalo. Cpp programski kod očitava vrijednosti komponenata sile i momenta iz UDP zaglavlja, te popunjavaju varijable pripadnih komponenti. Na slici 29. prikazani su zaprimljeni podaci.

```
Fx: 0.009992
Fy: -0.028408
Fz: -0.228507
Tx: 0.002256
Ty: -0.002415
Tz: -0.004805
-----
Fx: 0.001302
Fy: -0.025759
Fz: -0.228560
Tx: 0.001718
Ty: -0.002517
Tz: -0.004872
-----
Fx: 0.001448
Fy: -0.033561
Fz: -0.208351
Tx: 0.001503
Ty: -0.002843
Tz: -0.004859
```

Slika 29. Vrijednosti mjerenih komponenti sile i momenta

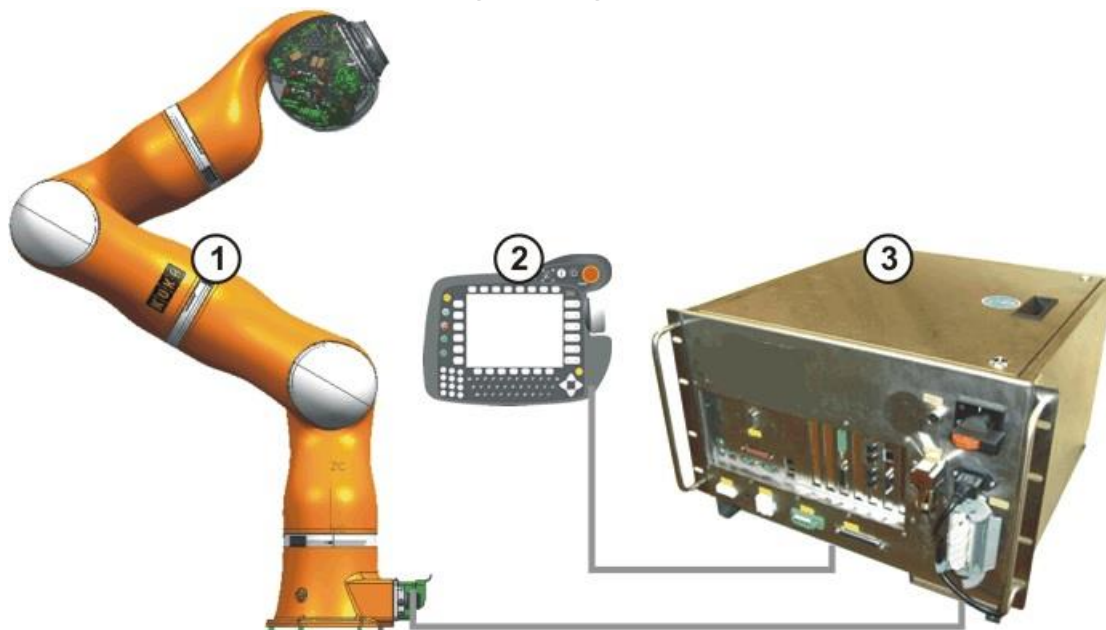
Kako bih mogli očitane podatke na računalo koristiti za daljnje procesuiranje prvo ih je potrebno filtrirati. Na slici 30. Prikazane su očitane vrijednosti sile na senzoru, gdje je vidljiv šum koji otežava i onemogućuje daljnju obradu podataka.



Slika 30. Odziv senzora sile

Proizvođač u sklopu sustava nudi niskopropusni filter, uz odabir granične frekvencije propuštanja. U sklopu ovoga rada, odabrana je frekvencija od 500Hz za prikupljanje podataka uz niskopropusni filter sa graničnom frekvencijom od 152Hz.

5.3 Robotski sustav Kuka Lightweight Robot LWR4+



Slika 31. Robotski sustav Kuka LWR 4+[9]

Na slici 31. je prikazan Kuka LWR 4+ robotski sustav, gdje su brojevima od 1 do 3 označeni glavni dijelovi sustava:

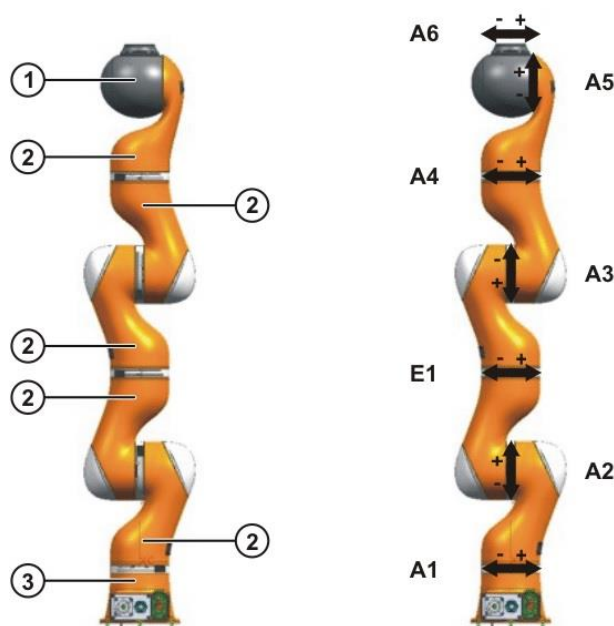
1. Robotska ruka - LWR 4+
2. KCP privjesak za učenje
3. Upravljačka jedinica - KR C2 lr

Robotska ruka LWR 4+ ima serijsku kinematsku strukturu kojom ostvaruje 6 stupnjeva slobode gibanja. Osnovne karakteristike navedene su u tablici 3. :

Tablica 3. Kuka LWR 4+ – osnovni tehnički podatci [9]

Nosivost	max. 7 kg
Masa	cca. 16 kg
Broj osi / broj upravljačkih osi / broj SSG	7 / 7 / 6
Napajanje	200 - 230 V AC - externo 27.1 ±0.1 V DC - interno
Ponovljivost	±0.05 mm
Volumen radnoga prostora	1.84 m ³

Na slici 32. prikazana je robotska ruka sa označenim elementima, te robotska ruka sa označenim upravljačkim osima rotacija .



Slika 32. Kuka LWR 4+ – elementi i upravljačke rotacijske osi [9]

Kuka LWR 4+ je 7-osna robotska ruka. Zglobovi su pokretani servo motorima koji su zaštićeni unutar konstrukcije robota. Svaki zglob opremljen je sensorima pozicije i sensorima momenta, što omogućuje upravljanje ovom robotskom rukom po poziciji, brzini ili momentu. Prema slici 32. Kuka LWR 4+ sastoji se od dvoosnog rotacijskog zgloba (1) kojeg u paru čine upravljačke osi A5 i A6, od preostalih 5 elemenata (2) povezanih rotacijskim zglobovima, te od baze (3) kojom je robotska ruka pričvršćena na čvrstu podlogu.

5.3.1 Komunikacija Kuka LWR4+ - vanjsko računalo

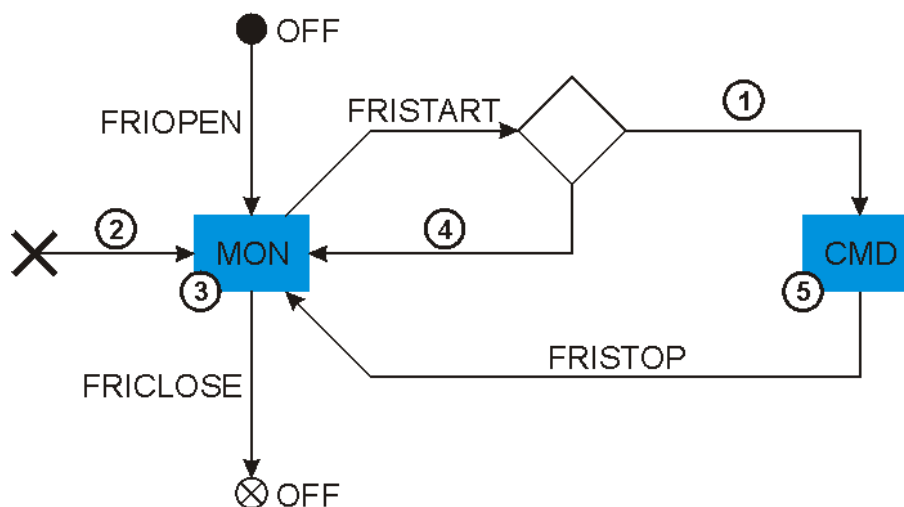
Fast Research Interface (FRI) programski je dodatak Kuka LWR4+ robotskog sustava namjenjen za komunikaciju Kuka Robot Controller-a (KRC-a) sa vanjskim računalom. U nastavku su navedene mogućnosti FRI-a: [10]

1. Određivanje vremena ciklusa za prijenos podataka (1-100 ms)
2. Slanje upravljačkih signala u stvarnom vremenu
3. Nadzor kretanja robotske ruke sa vanjskog računala
4. Pristup podacima robota sa vanjskog računala
5. Prijenos poruka sa KRC-a na vanjskog računalo
6. Primanje i slanje vrijednosti sistemskih i programskih varijabli robota
7. Izvršavanje programa sa vanjskog računala

Komunikacija između KRC-a i vanjskog računala preko FRI-a može biti ostvarena na dva načina:

1. Nadgledajući način rada (eng. *Monitor mode*) – dvosmjerna komunikacija FRI-a i vanjskog računala. Upravljanje gibanjem robotskom rukom obavlja KRC preko Kuka Robot Language-a (KRL).
2. Upravljački način rada (eng. *Command mode*) – dvosmjerna komunikacija FRI-a i vanjskog računala. Upravljanje gibanjem robotskom rukom preuzima vanjsko računalo.

FRI-a za komunikaciju KRC-a i vanjskoga računala koristi Ethernet, IP i UDP protokole. Vanjsko računalo mora osigurati serverski priključak za UDP komunikaciju. U ovome radu serverski priključak i sve ostale potrebne funkcije na strani vanjskoga računala ostvareni su pomoću cpp skripte. Cpp skripta napisana je od strane proizvođača, sa ciljem da korisnik prema svojim potrebama prilagodi željene parametre komunikacije. Na slici 32. nalazi se dijagram toka kojim je prikazan način funkcioniranja FRI-a. Pojedininim naredbama Brojevima su označena pojedina stanja ili događaji u vremenskom ciklusu komunikacije. [10]



Slika 33. Dijagram toka FRI-a [10]

1. Podatkovni paket postavljen u točno određenom vremenskom okviru
2. Greška
3. Nadgledajući način rada (MON)
4. Kvaliteta prijenosa podataka nije zadovoljavajuća
5. Upravljački način rada (CMD)

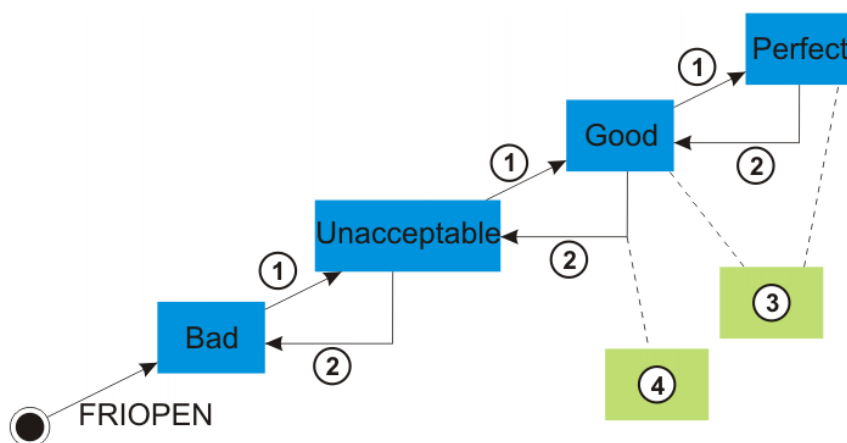
Na strani KRC-a FRI komunikacijom se upravlja sljedećim naredbama:

FRIOPEN - otvara FRI komunikacijski kanal između KRC-a i vanjskog računala. Nadgledajući način rada omogućuje prijenos podataka na relaciji KRL – KRC – vanjsko računalo.

FRISTART - postavlja FRI-a u upravljački način rada uz uvijete da se FRI nalazi u nadgledajućem načinu rada i da je kvaliteta prijenosa podataka klasificirana kao „Dobra“ ili „Odlična“. Ukoliko kvaliteta prijenosa podataka u određenom trenutku nije zadovoljavajuća ili se pojavi određena greška u komunikaciji, FRI se bez vremenske odgode vraća u nadgledajući način rada.

FRISTOP – zaustavlja upravljački način rada FRI-a i postavlja ga u nadgledajući način rada.

FRICLOSE – zatvara FRI komunikacijski kanal.



Slika 34. Kvaliteta pijenosa podataka [10]

Na slici 34. prikazan je dijagram toka određivanja kvalitete prijenosa podataka. Brojevima su označeni sljedeći koraci: [10]

1. „Rukovanje“(eng. *Handshaking*) je provedeno
2. UDP paket podataka je izgubljen
3. Moguće korištenje upravljačkoga načina rada
4. Upravljački način rada je ugašen; FRI se nalazi u nadgledajućem načinu rada

Prilikom uspostave komunikacije preko FRI-a, kvaliteta prijenosa podataka uvijek je klasificirana kao „Loša“ (eng. *Bad*). Uspješnim „rukovanjem“ klasificira se kao „Neprihvatljiva“ (eng. *Unacceptable*), ta zatim kao „Dobra“ (eng. *Bad*) i „Odlična“ (eng. *Perfect*).

5.4 Sustav za bušenje Medtronic

Sustav za bušenje, proizvođača Medtronic, sastoji se od upravljačke jedinice – Integrated Power Console (IPC™), višenamjenske nožne upravljačke jedinice, kablova za napajanje i upravljanje, te raznih alata za rezanje, brušenje i glodanje.

IPC je upravljačka jedinica sustava za bušenje, rezanje i otklanjanje mekanoga i tvrdoga tkiva, kosti i ostalog biomaterijala tijekom kirurških operacija.



Slika 35. IPC Model EC300 [11]

Na slici 35. nalazi se IPC Model EC300 na kojem su slovima naznačeni glavni dijelovi:

- A. Zaslona osjetljiv na dodir – korisničko sučelje
- B. Pumpa 1 - namjenjena za rashladna sredstva i navodnavanje
- C. Pumpa 2 - namjenjena za rashladna sredstva i navodnavanje
- D. Niz priključaka za razne alate i periferne sklopove
- E. Sklopka za paljenje i gašenje uređaja

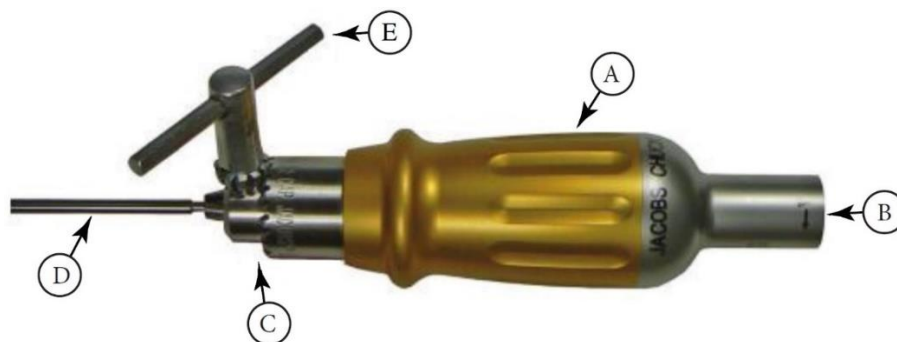
Midas Rex® Legend EHS® Motor je električni motor koji je sposoban razviti velike brzine uz velike okretno momente. Mogućnost mjenjanja alata koju ovaj motor pogoni, pruža mu široku primjenu u medicini. Koristi se za rezanje i bušenje kosti, ali isto tako i za odvajanje biomaterijala prilikom kirurških zahvata. Iako malih dimenzija, duljine 9.02 cm i promjera kućišta 2.03 cm, te težine od svega 183 g, Legend EHS® Motor razvija brzine od 200 do 75000 rpm. [11]



Slika 36. Midas Rex® Legend EHS® Motor [11]

Na slici 36. prikazan je Midas Rex® Legend EHS® Motor sa označenim dijelovima:

- A. Midas Rex® Legend EHS® Motor
- B. 4-pinski priključak za kabel
- C. Rotirajući prsten za zaključavanje nastavka
- D. Stacionarni prsten za zaključavanje nastavka



Slika 37. Jacobs® reduktor sa steznom glavom AD02 (5/32'') [11]

Na slici 37. je prikazan reduktor sa steznom glavom za svrdlo proizvođača Jacobs. Reduktor je posebno konstruiran za potrebe korištenja u medicini kao pomoćni nastavak pogonskih elemenata proizvođača Medtronic. Prijenosni omjer reduktora iznosi 30.

- A. Kućište reduktora
- B. Utor za spajanje pogonskog uređaja
- C. Stezna glava za svrdlo
- D. Svrdlo
- E. Ključ za stezanje stezne glave



Slika 38. Višenamjenska nožna kontrolna jedinica [11]

Na slici 38. prikazana je višenamjenska nožna kontrolna jedinica koja služi za upravljanje IPC sustavom. U nastavku su opisane funkcije pojedinih sklopki :

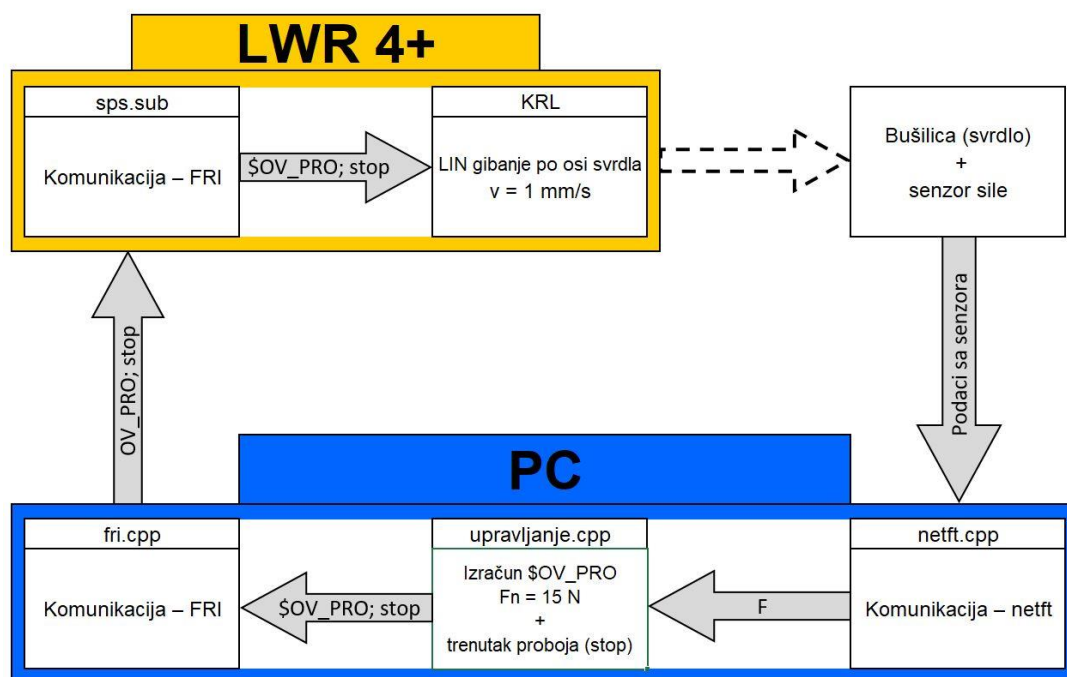
- Aa. Protuklizna površina
- A. Nožna pedala – Start/Stop ili regulacija brzine
- B. Funkcijska sklopka za odabir načina rada nožne pedale (Start/Stop ili reg. brzine)
- C. Funkcijska sklopka za odabir alata (mogućnost spajanja 4 alata istovremeno na IPC)
- D. Funkcijska sklopka za odabir smjera vrtnje svrdla

5.5 Računalo

Na strani računala korišten je programski paket Microsoft Visual Studio 2010, programski jezik cpp (C++). Korištene su tri simultano izvršavane cpp skripte. Takav način izvršavanja programskoga koda naziva se *multithreading*. Izvršavanjem jedne skripte odvija se komunikacija sa Net F/T komunikacijskim sučeljem senzora sile, programski kod druge cpp skripte zadužen je za FRI komunikaciju računala sa KRC-om LWR4+ robotske ruke, dok treća cpp skripta izvršava kod zadužen za upravljanje procesom bušenja. Cpp skripte se nalaze u prilogu.

5.6 Upravljački sustav

Prilikom procesa bušenje potrebno je održavati silu pritiska svrdla konstantnom. U ovome radu ulazni parametri su iznos željene konstantne sile pritiska svrdla prilikom bušenja i iznos maksimalne posmične brzine svrdla. Svrldo se kreće zadanom maksimalnom posmičnom brzinom ukoliko je sila pritiska svrdla manja od željene sile pritiska. Posmična brzina svrdla se smanjuje ovisno o iznosu sile pritiska na svrdlu. Maksimalna posmična brzina svrdla postavlja se u KRL programskom kodu kao jedan od parametara linearnoga gibanja svrdla. Kako parametar brzine linearnoga gibanja nije moguće unutar KRL-a mjenjati prilikom izvršavanja samoga gibanja, iskorištena je mogućnost mjenjanja sistemske varijable \$OV_PRO unutar samoga KRC-a. \$OV_PRO je sistemska varijabla koja određuje brzinu gibanja robota prilikom izvršavanja KRL programskoga koda, opisana je kao postotak (0-100%) brzine koja je podešena u parametrima KRL gibanja.



Slika 39. Shematski prikaz komunikacije unutar laboratorijskog postava za bušenje

Na slici 39. shematski je prikazana smjer komunikacije unutar laboratorijskog postava za bušenje, gdje je prikazano koji podaci se razmijenjuju između kojih segmenata sustava. Isprekidana strelica koja pokazuje smjer između „LWR4+“ i „Bušilica (svrdlo) + senzor sile“ ne ukazuje na komunikaciju između ova dva segmenta, već na fizički utjecaj, pošto su bušilica i senzor sile pričvršćeni na pribornicu robota, te samim time pod izravnim su utjecajem djelovanja robotske ruke.

Za regulaciju sile pritiska svrdla prilikom bušenja u ovome radu korišten je PI regulator implementiran unutar cpp skripte *upravljanje*.

$$u = K_p e + K_i e_i \quad (11)$$

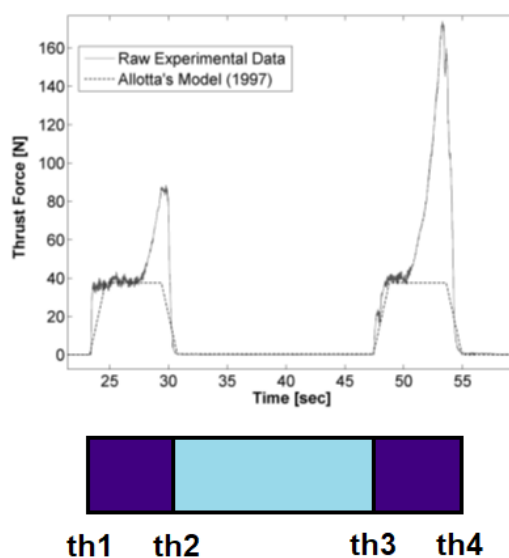
Gdje su K_p i K_i pojačanja proporcionalnog i integralnoga člana regulatora, dok su e i e_i greške regulirane veličine:

$$e = F_n - F_{x_svrdlo} \quad (12)$$

$$e_i = e_i + e dt \quad (13)$$

Pravovremeno zaustavljanje translacijskog gibanja prilikom proboja svrdla jako je bitan segment unutar teme osjetljivoga robotskoga bušenja. Kod homogenih materijala jasni pokazatelj proboja je pad sile pritiska svrdla, a povezano sa time i povećanje posmične brzine svrdla. Kod nehomogenih materijala pad sile pritiska svrdla nije nužno povezan sa probojem, već može označavati i promjenu strukture materijala. Ukoliko bušimo nehomogeni materijal kod kojeg niti približno neznamo raspodjelu svojstava unutar materijala, tada je vrlo teško razaznati dali pad sile označava proboj ili promjenu strukture unutar nehomogenog materijala.

Polazeći od pretpostavke da okvirno poznajemo strukturu bušenoga materijala, znamo da se mekši sloj nalazi unutar dva tvrđa sloja, dizajniran je algoritam koji detektira promjene strukture unutar materijala i proboj. Algoritam detektira granice na temelju pada ili prekoračenja sile pritiska svrdla prilikom bušenja ispod ili iznad određenog praga sile pritiska (eng. *threshold*).

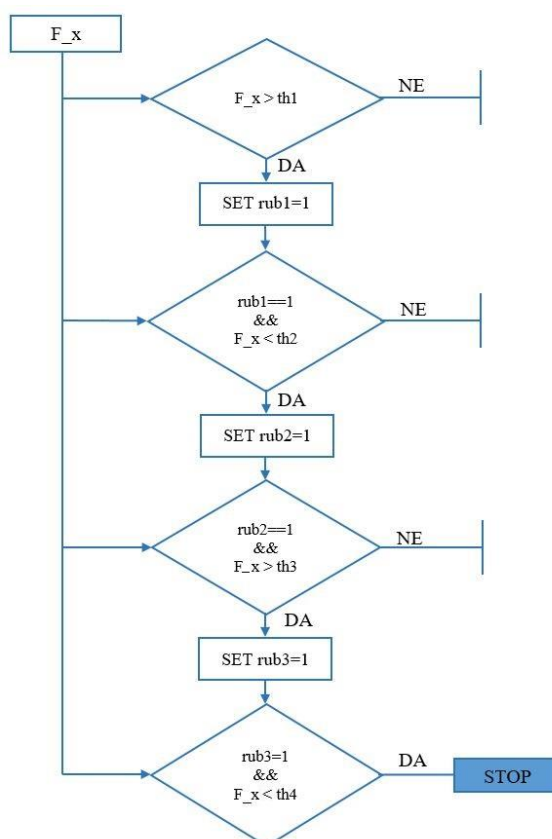


Slika 40. Nehomogeni materijal – prikaz parametara algoritma[12]

Na slici 40. prikazane su vrijednosti sile pritiska prilikom bušenja nehomogenog materijala, konkretno ljudske kosti. Na grafu su prikazane vrijednosti pretpostavljene sile pritiska svrdla prilikom bušenja, te stvarni rezultati dobiveni bušenjem duge ljudske kosti. [12] Ispod grafa ilustrirana su tri sloja, dva vanjska kompaktna (tvrda), te jedan unutarnji spongiozni(mekani) sloj. Gdje su nazačene vrijednosti praga (eng. *threshold*) sile pritiska na pojedinim područjima promjene strukture unutar kosti:

- th1 – vrijednost praga sile pritiska kod koje je započelo bušenje prvoga sloja
- th2 - vrijednost praga sile pritiska prilikom pada sile zbog promjene strukture materijala – drugi sloj
- th3 - vrijednost praga sile pritiska prilikom povećanja sile zbog promjene strukture materijala –treći sloj
- th4 – vrijednost praga sile pritiska prilikom proboja

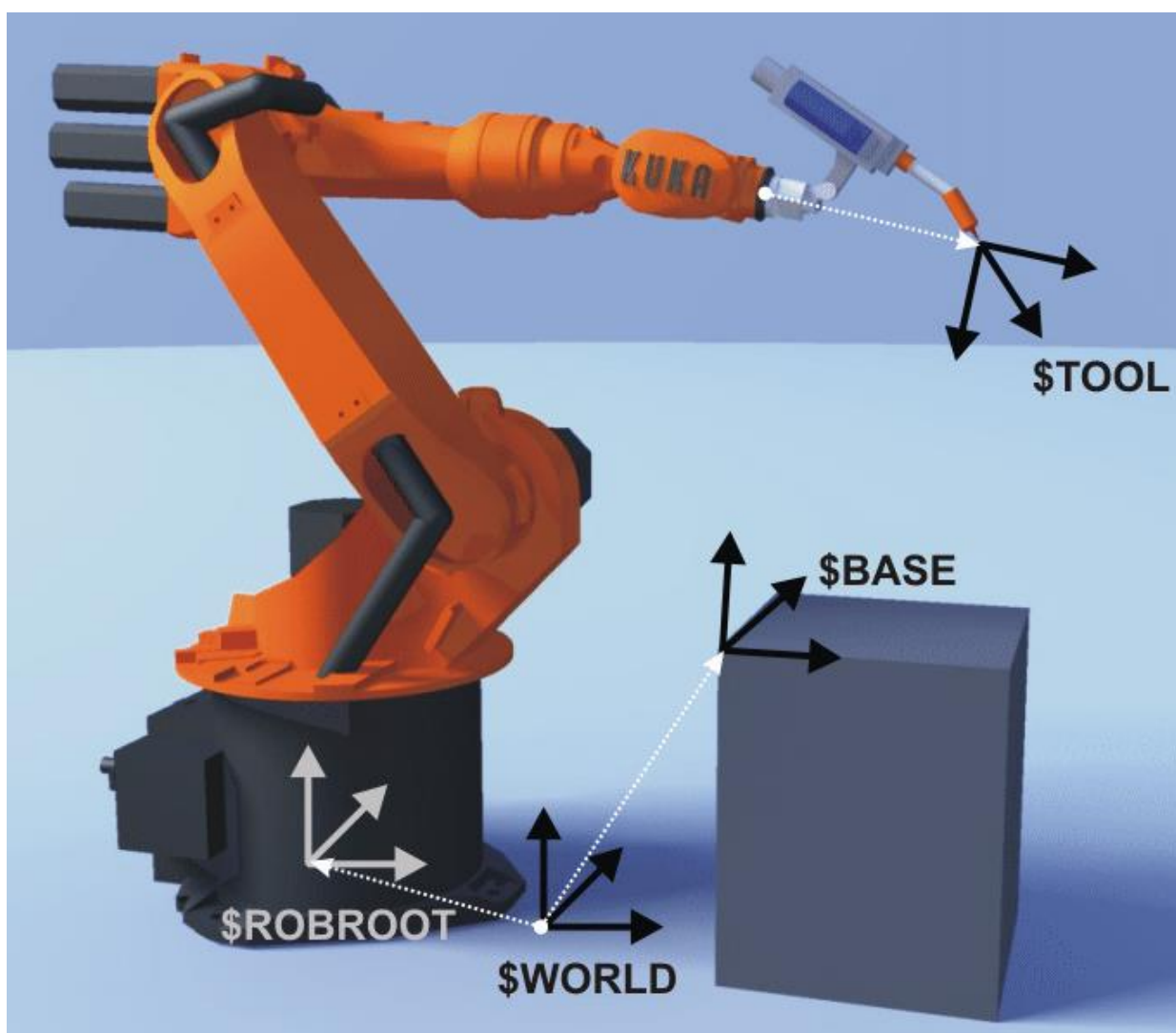
Na slici 41. prikazan je dijagram toka algoritma za detekciju proboja svrdla. Prije početka korištenja algoritma potrebno je podesiti vrijednosti praga sile pritiska svrdla. Sve četiri vrijednosti praga sile pritiska mogu biti podešene na jednake vrijednosti, ovisno o materijalu bušenja i dopuštenome hodu svrdla nakon proboja.



Slika 41. Algoritam za detekciju proboja

6 KALIBRACIJA ALATA

Robotska ruka u užem smislu riječi, dakle sama konstrukcija i pogonski elementi, je fizička interpretacija računala, tj. upravljačke jedinice robota. Svojim djelovanjem, odnosno gibanjem unutar radnoga prostora izvršava naredbe producirane djelovanjem upravljačke jedinice ili računala. Kako bih upravljačka jedinica mogla upravljati gibanjem robotske ruke, potrebno je poznavanje matematičkog opisa prostora u kojem se nalazi robotska ruka, te poznavanje kinematičkoga modela robota. U robotici je najzastupljeniji opis prostora desnokretnim kartezijskim koordinatnim sustavom.



Slika 42. Koordinatni sustavi robota [9]

Na slici 42. nalazi se robotska ruka sa prikazanim koordinatnim sustavima. Navedeni koordinatni sustavi koriste se prilikom upravljanja gibanjem robotske ruke: [9]

- WORLD – Globalni koordinatni sustav je trajno opisan kartezijski koordinatni sustav. U odnosu na globalni koordinatni sustav opisani su koordinatni sustavi robota i postolja .
- ROBROOT – Koordinatni sustav robota je kartezijski koordinatni sustav koji je smješten unutar postolja samoga robota. Opisan je relativnim odmakom u odnosu na globalni koordinatni sustav.
- BASE – Koordinatni sustav postolja je kartezijski koordinatni sustav koji opisuje relativnu poziciju i orijentaciju postolja sa radnim izratkom u odnosu na globalni koordinatni sustav.
- TOOL – Koordinatni sustav alata je kartezijski koordinatni sustav čije se iskodište nalazi u vrhu alata (eng. *Tool Centar Point*, TCP). Prema tvorničkim postavkama iskodište koordinatnog sustava alata nalazi se u centru prirubnice robota. Koordinatni sustav alata opisan je relativnim odmakom u odnosu na koordinatni sustav prirubnice robota.

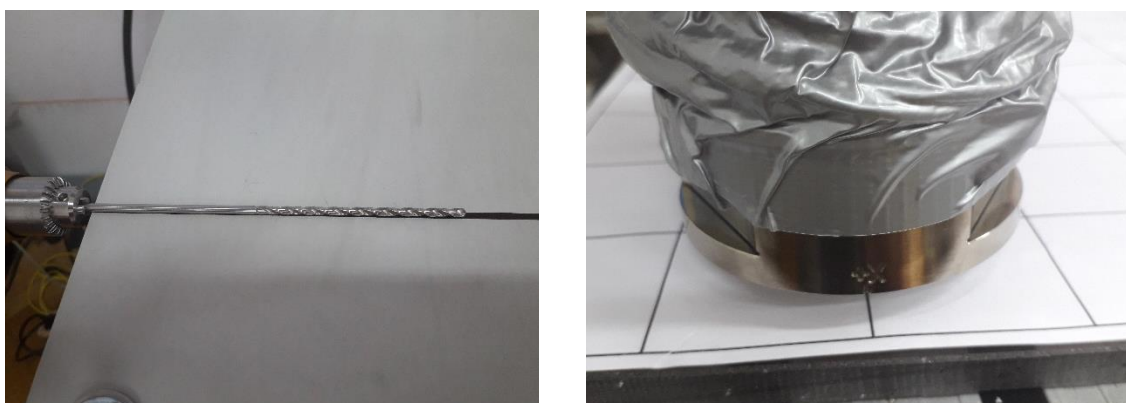
Za osnovne operacije gibanja robota, pogotovo ako se željene točke gibanja zadaju dovođenjem robota u točno željene pozicije, nije najnužnije poznavati koordinatni sustav alata koji koristimo. No ukoliko želimo izvršavati napredna gibanja, kao naprimjer u našem slučaju gibanje po točno određenoj osi alata, tada je potrebno odrediti poziciju i orijentaciju koordinatnog sustava alata. TCP (eng. *Tool Centar Point*) skraćunica je koja označava središnju točku alata, tj. iskodišnu točku koordinatnog sustava alata. Kalibracija alata sastoji se od dvije faze. U prvoj fazi određujemo poziciju TCP-a, dok u drugoj fazi određujemo orijentaciju koordinatnog sustava kome je TCP iskodišna točka. U ovome radu bilo je potrebno izvršiti kalibraciju glavnoga alata, tj. svrdla i kalibraciju senzora sile.

Za određivanje TCP-a, odnosno iskodišne točke koordinatnih sustava korištena je metoda „XYZ 4-point“. Nulti korak ove metode je odabir statične referentne točke u radnome prostoru robota. Zatim se vrh alata dovodi u referentnu točku i pohranjuje se trenutna pozicija i orijentacija. Ovaj korak se ponavlja četiri puta iz što različitijih orijentacija vrha alata. Na slici 44. prikazana je kalibracija TCP-a svrdla i TCP-a senzora sile.



Slika 43. Kalibracija TCP-a svrdla i TCP-a senzora sile

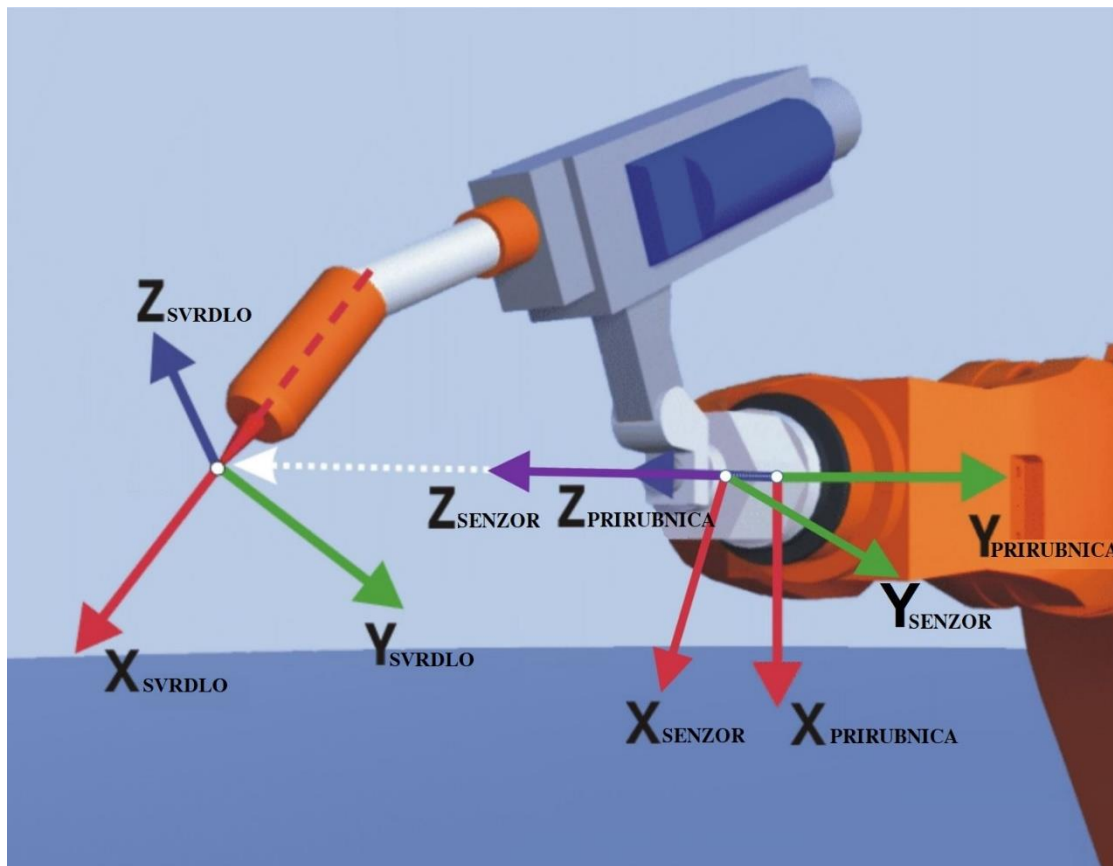
Orijentacija koordinatnih susava svrdla i senzora sile odrađena je metodom „ABC 2-point“. Kalibracija se provodi na način da se alat prvo dovede u referentnu poziciju i orijentaciju, te se navedena točka pohrani. Zatim se alat pomakne u negativnom smjeru osi X alata, te se pohrani navedena točka. U zadnjem koraku kalibracije, alat se pomakne u XY ravnini u negativnom smjeru osi Y alata, te se pohranjuje točka.



Slika 44. Kalibracija orijentacije koordirdinatnih sustava svrdla i senzora sile

Završetkom kalibracija dobivamo izlazne podatke o pozicijama (X, Y, Z) i orijentacijama (A, B, C) koordinatnih sustava svrdla i senzora sile u odnosu na koordinatni sustav prirubnice robota.

6.1 Matrica rotacije sile



Slika 45. Koordinatni sustavi prirubnice, senzora i svrdla[9]

Na slici 45. prikazani su koordinatni sustavi prirubnice robota, senzora sile i svrdla. Prilikom procesa bušenja potrebno je očitanoj silom u smjeru osi gibanja svrdla regulirati posmičnu brzinu bušenja. Zbog konstrukcijskih ograničenja silu koja se javlja na svrdlu u smjeru njegovog gibanja prilikom bušenja, nije moguće izjednačiti samo sa jednom od tri komponente sile očitane na senzoru. Stoga je potrebno matricom rotacija očitane komponente sile sa senzora rotirati u koordinatni sustav svrdla, što je opisano matričnom jednadžbom:

$$\mathbf{F}_{svrdlo} = \mathbf{F}_{senzor} ({}^{senzor}R_{svrdlo}) \quad (14)$$

Gdje su \mathbf{F}_{svrdlo} i \mathbf{F}_{senzor} vektori čiji su elementi komponente transformirane sile na svrdlu odnosno očitane komponente sile na senzoru:

$$\mathbf{F}_{svrdlo} = [F_{x_svrdlo} \quad F_{y_svrdlo} \quad F_{z_svrdlo}] \quad (15)$$

$$\mathbf{F}_{senzor} = [F_{x_senzor} \quad F_{y_senzor} \quad F_{z_senzor}] \quad (16)$$

Matrica ${}^{senzor}R_{svrdlo}$ je matrica dimenzija 3x3 koja sadrži komponente rotacija osi koordinatnog sustava svrdla u odnosu na koordinatni sustav senzora. Izraz (17) opisuje matricu rotacija iz koordinatnog sustava pribubnice u koordinatni sustav svrdla:

$${}^pR_{svrdlo} = ({}^pR_{senzor})({}^{senzor}R_{svrdlo}) \quad (17)$$

Sređivanjem izraza dobivamo matični izraz za izračun matrice ${}^{senzor}R_{svrdlo}$ koji glasi:

$${}^{senzor}R_{svrdlo} = ({}^pR_{senzor})^{-1}({}^pR_{svrdlo}) \quad (18)$$

Matrice ${}^pR_{svrdlo}$ i ${}^pR_{senzor}$ definirane su izrazima (19) i (20) :

$${}^pR_{senzor} = Rot(z_p, A_{senzor}) \times Rot(y_p, B_{senzor}) \times Rot(x_p, C_{senzor}) \quad (19)$$

$${}^pR_{svrdlo} = Rot(z_p, A_{svrdlo}) \times Rot(y_p, B_{svrdlo}) \times Rot(x_p, C_{svrdlo}) \quad (20)$$

Gdje su :

$$Rot(z_p, A_{senzor}) = \begin{bmatrix} \cos(A_{senzor}) & -\sin(A_{senzor}) & 0 \\ \sin(A_{senzor}) & \cos(A_{senzor}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

$$Rot(y_p, B_{senzor}) = \begin{bmatrix} \cos(B_{senzor}) & 0 & \sin(B_{senzor}) \\ 0 & 1 & 0 \\ -\sin(B_{senzor}) & 0 & \cos(B_{senzor}) \end{bmatrix} \quad (22)$$

$$Rot(x_p, C_{senzor}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(C_{senzor}) & -\sin(C_{senzor}) \\ 0 & \sin(C_{senzor}) & \cos(C_{senzor}) \end{bmatrix} \quad (23)$$

$$Rot(z_p, A_{svrdlo}) = \begin{bmatrix} \cos(A_{svrdlo}) & -\sin(A_{svrdlo}) & 0 \\ \sin(A_{svrdlo}) & \cos(A_{svrdlo}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

$$Rot(y_p, B_{svrdlo}) = \begin{bmatrix} \cos(B_{svrdlo}) & 0 & \sin(B_{svrdlo}) \\ 0 & 1 & 0 \\ -\sin(B_{svrdlo}) & 0 & \cos(B_{svrdlo}) \end{bmatrix} \quad (25)$$

$$Rot(x_p, C_{svrdlo}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(C_{svrdlo}) & -\sin(C_{svrdlo}) \\ 0 & \sin(C_{svrdlo}) & \cos(C_{svrdlo}) \end{bmatrix} \quad (26)$$

Gdje su:

x_p, y_p, z_p – osi koordinatnoga sustava prirubnice

A_{senzor} – kut zakreta koordinatnog sustava senzora oko z_p [rad]

B_{senzor} – kut zakreta koordinatnog sustava senzora oko y_p [rad]

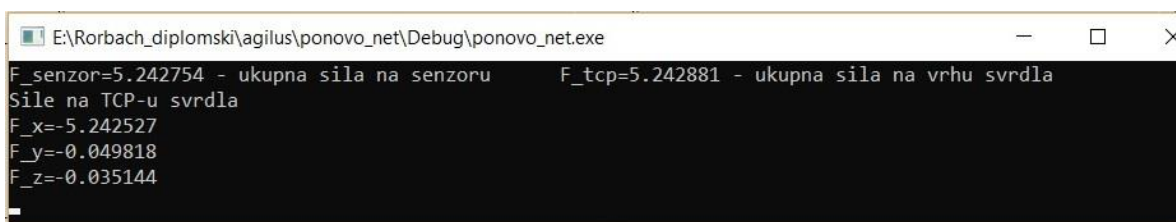
C_{senzor} – kut zakreta koordinatnog sustava senzora oko x_p [rad]

A_{svrdlo} – kut zakreta koordinatnog sustava svrdla oko z_p [rad]

B_{svrdlo} – kut zakreta koordinatnog sustava svrdla oko y_p [rad]

C_{svrdlo} – kut zakreta koordinatnog sustava svrdla oko x_p [rad]

Navedeni kutevi poznati su nam iz poglavlja 6. *Kalibracija alata*, gdje smo ih odredili koristeći kalibracijsku metodu „ABC 2-point“. Uvrštavanjem kuteva u matrice (21) do (26) i rješavanjem matričnih jednadžbi (18), (19) i (20) dobivamo matricu ${}^{senzor}R_{svrdlo}$.



```

E:\Rorbach_diplomski\agilus\ponovo_net\Debug\ponovo_net.exe
F_senzor=5.242754 - ukupna sila na senzoru      F_tcp=5.242881 - ukupna sila na vrhu svrdla
Sila na TCP-u svrdla
F_x=-5.242527
F_y=-0.049818
F_z=-0.035144
  
```

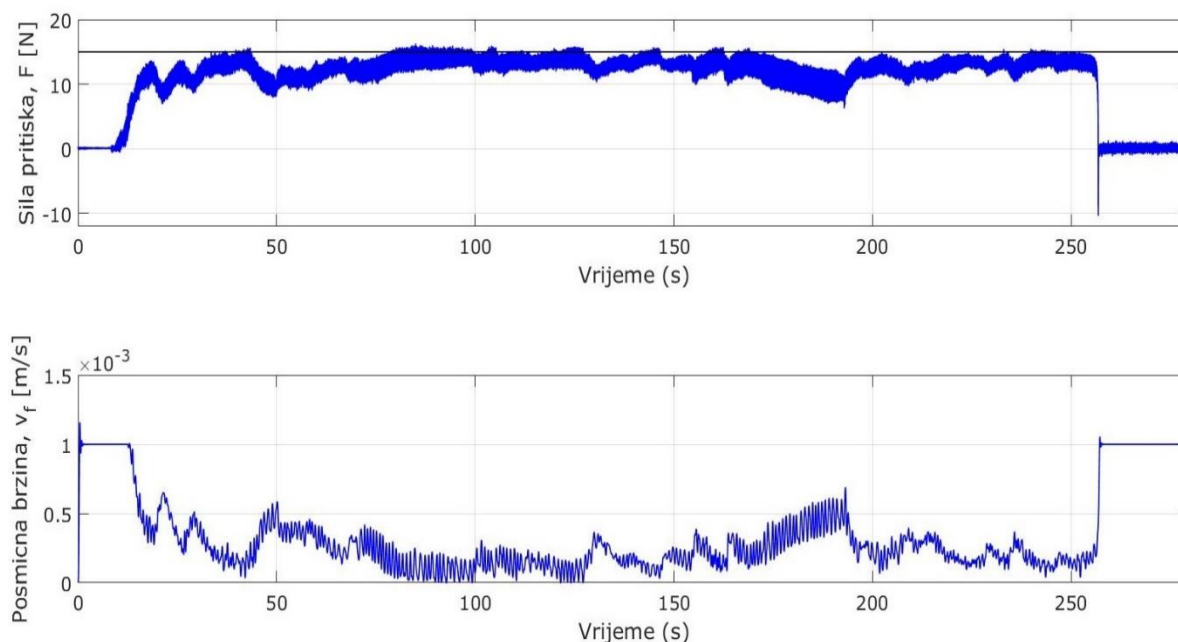
Slika 46. Komponente sile na svrdlu

Na slici 47. prikazani su iznosi komponenata sile u koordinatnom sustavu svrdla, ukupna sila u koordinatnome sustavu svrdla, te ukupna sila na senzoru sile. Na svrdlo je djelovala samo sila u aksijalnome smjeru. Vidljivo je da radialne sile (F_y i F_z) imaju određene male iznose. Tijekom rada je ustanovljeno da se javlja greška do 10% prilikom rotacije komponenata sile. Mogućih razloga je više, od kojih dva imaju zasigurno najveći utjecaj. Nedovoljno precizna kalibracija svrdla i senzora sile, te pojava grešaka uslijed zaokruživanja tijekom računanja matrice rotacija ${}^{senzor}R_{svrdlo}$.

7 EKSPERIMENTALNI REZULTATI

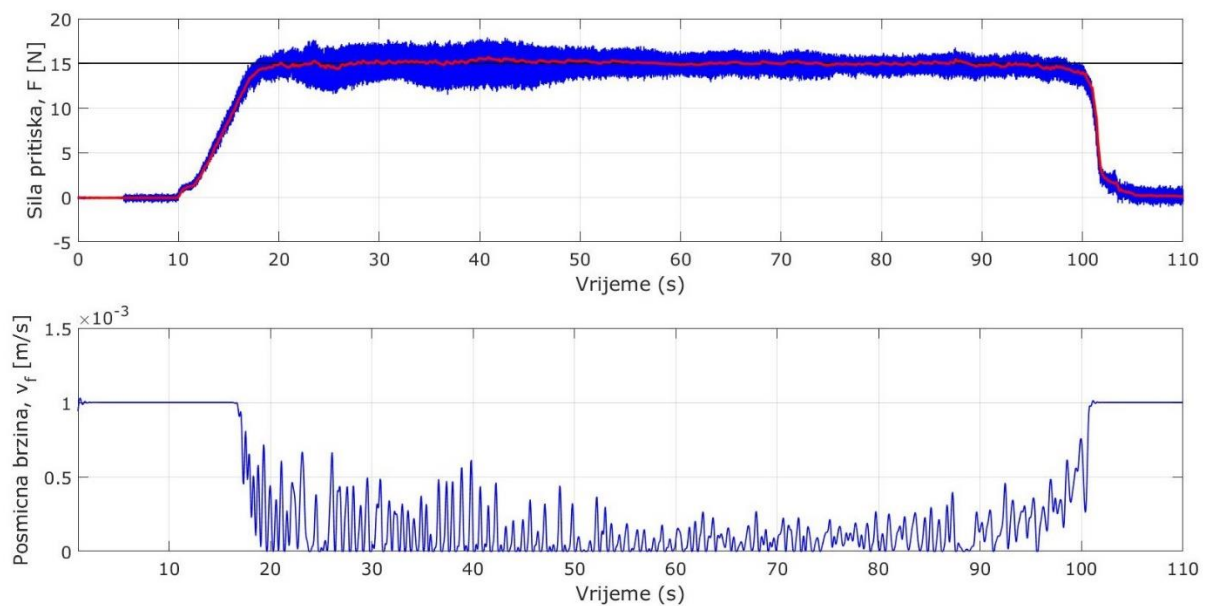
Kako je već ranije navedeno u tekstu, ulazni parametri u proces bušenja su željena sila pritiska svrdla i maksimalna posmična brzina svrdla. Željena sila pritiska svrdla podešena je na 15 N, dok je maksimalna posmična brzina svrdla podešena na 1 mm/s. Korišteno je svrdlo $\phi 3$ mm, uz rotacijsku brzinu od 450 o/min.

Podešavanje PI regulatora moguće je izvesti empirijski, npr. Ziegler–Nicholsovom, Tyreus Luybenovom, Cohen–Coonovom metodom ili analitički, npr. optimum dvostrukoga odnosa. U ovome radu pojačanja PI regulatora podešena su ručno eksperimentalnom metodom bušeci više različitih homogenih i nehomogenih materijala. Kao npr. Al lim, drvo i polimetil-metakrilat (pleksiglas). Polimetil-metakrilat nije se pokazao dobrim materijalom za eksperiment, uslijed zagrijavanja svrdla i materijala bušenja, polimetil-metakrilat bih omekšao što bi dovelo do pada sile pritiska.



Slika 47. Sila pritiska i posmična brzina bušenja; drvo ; $K_p = 10$, $K_i = 0$

Na slici 47. prikazani su sila pritiska svrdla i posmična brzina svrdla prilikom bušenja drvenog materijala. Pojačanja PI regulatora su podešena na $K_p = 10$, $K_i = 0$. Vidljivo je kako samo proporcionalni član uz ovako malo pojačanje ne može održati silu pritiska konstantnom.



Slika 48. Sila pritiska i posmična brzina bušenja; Al lim ; $K_p = 40$, $K_i = 20$

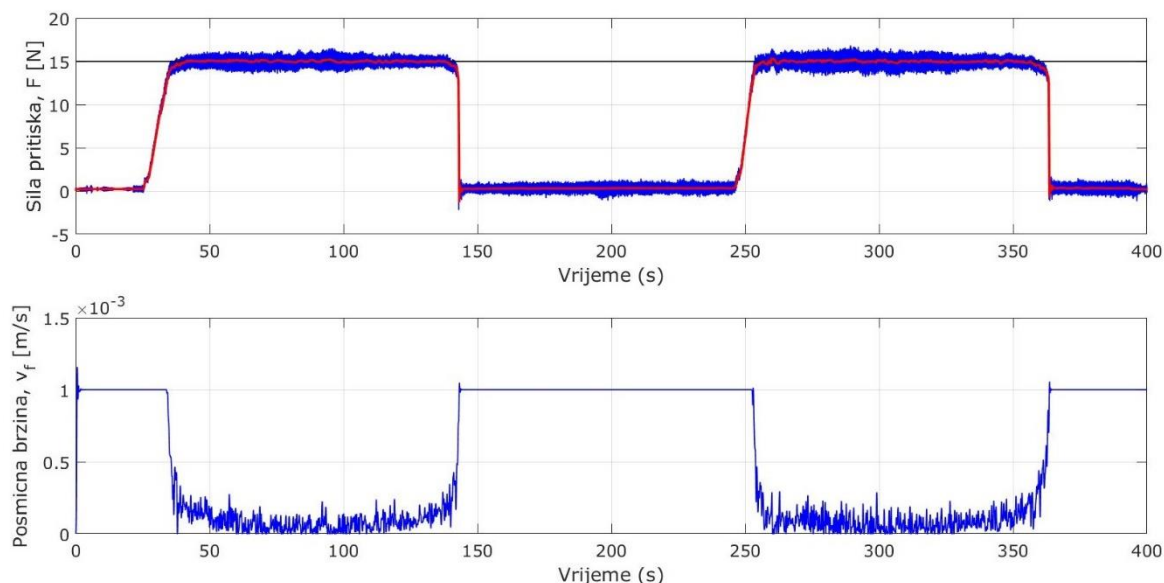
Na slici 48. prikazani su rezultati bušenja aluminijskoga lima debljine 1 mm, sa podešenim pojačanjima regulatora $K_p = 40$ i $K_i = 20$. Crvena linija predstavlja filtrirani signal iznosa sile pritiska svrdla. Na slici je vidljivo kako je PI regulator s ovako postavljenim pojačanjima jako dobro održava silu pritiska konstantnom tijekom cijelog ciklusa bušenja.

Nehomogena struktura kojom je testiran sustav prikazana je na slici 49. Sastavljena je od tri sloja, dva vanjska sloja aluminijska lima, te unutanji sloj ekspaniranoga polistirena – eps (stiropor).



Slika 49. Nehomogena struktura (lim-eps-lim)

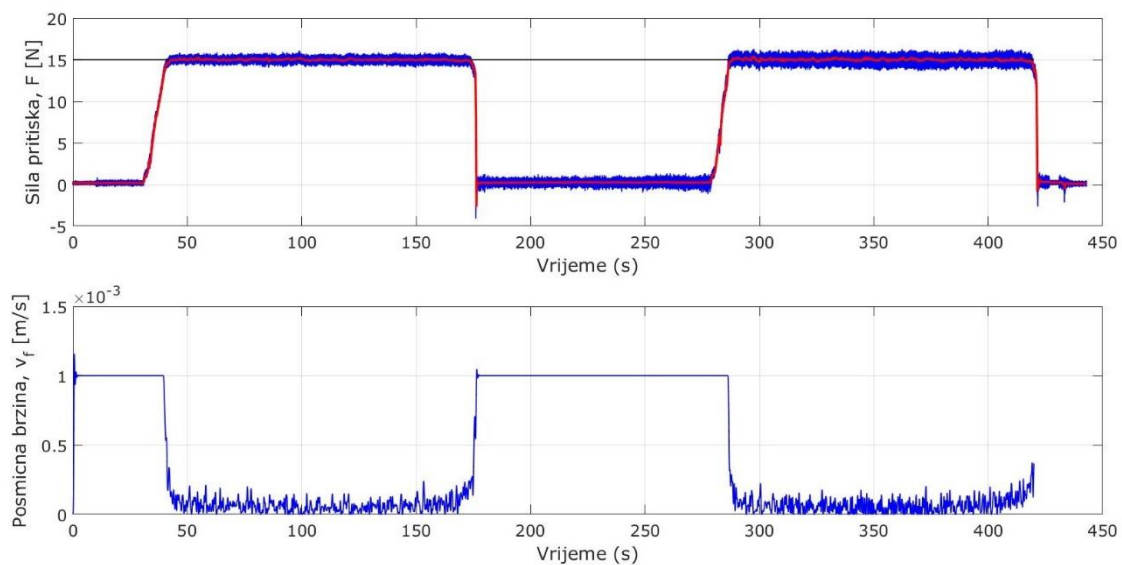
Koristeći iste postavke PI regulatora kao i kod bušenja aluminijskoga lima, provedeno je nekoliko bušenja ove nehomogene strukture.



Slika 50. Sila pritiska i posmična brzina bušenja; lim – eps - lim; $K_p = 40$, $K_i = 20$

Na slici 50. prema iznosu sile pritiska i posmične brzine svrdla jasno se razaznaju područja unutar strukture sa tvrdim vanjskim slojem materijala i unutarnjim mekanijim slojem materijala. PI regulator očekivano održava silu pritiska konstantnom u dijelu nehomogene strukture gdje se nalazi aluminijski lim, dok u području expandiranoga polistirena nije moguće postići nominalnu silu bušenja zbog ograničenja posmične brzine svrdla. Upravo ove pojave promjene sile pritiska unutar nehomogene strukture omogućuju nam pronalaženje granica između materijala različitih svojstava, te u konačnici detekcije proboja svrdla.

Prema rezultatima bušenja prikazanima na slici 50. postavljeni su parametri (th1,th2,th3,th4) algoritma za detekciju proboja svrdla. Na slici 51. nalaze se rezultati ponovljenog bušenja nehomogene strukture lim-eps-lim sa pravovremenim zaustavljanjem svrdla nakon proboja.



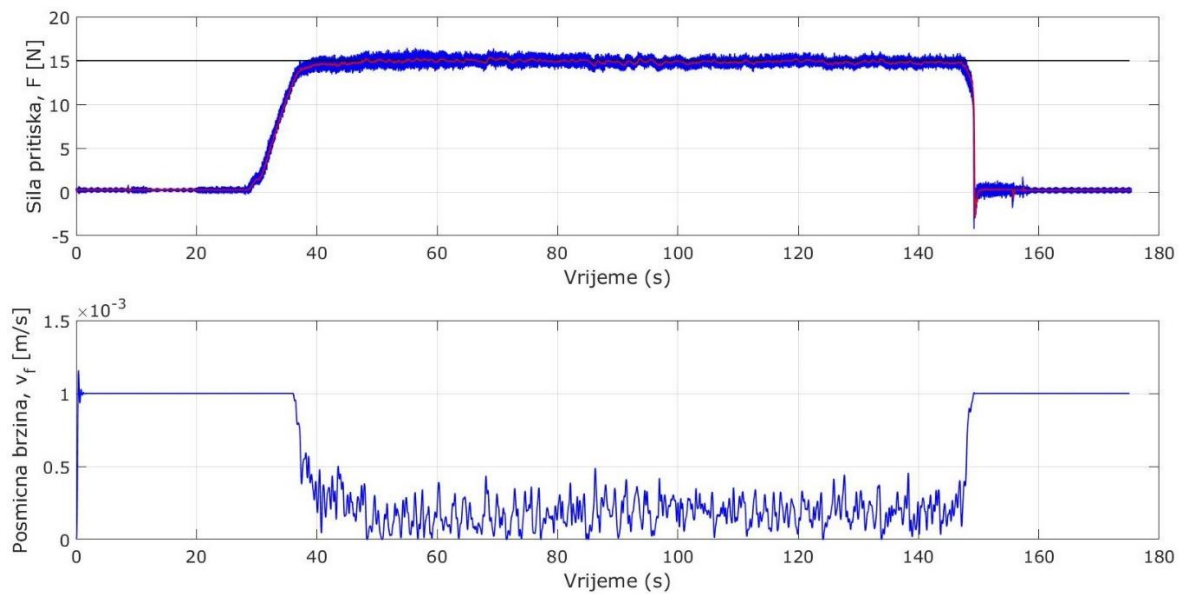
**Slika 51. Sila pritiska i posmična brzina bušenja; lim – eps - lim;
 $K_p = 40$, $K_i = 20$; detekcija proboja**

Teleća lopatična kost svojom gradivnom strukturom i svojim mehaničkim svojstvima približno odgovara ljudskim pločastim kostima. Na slici 52. nalazi se teleća lopatična kost koja je korištena u sklopu ovoga rada za testiranje upravljačkoga sustava.



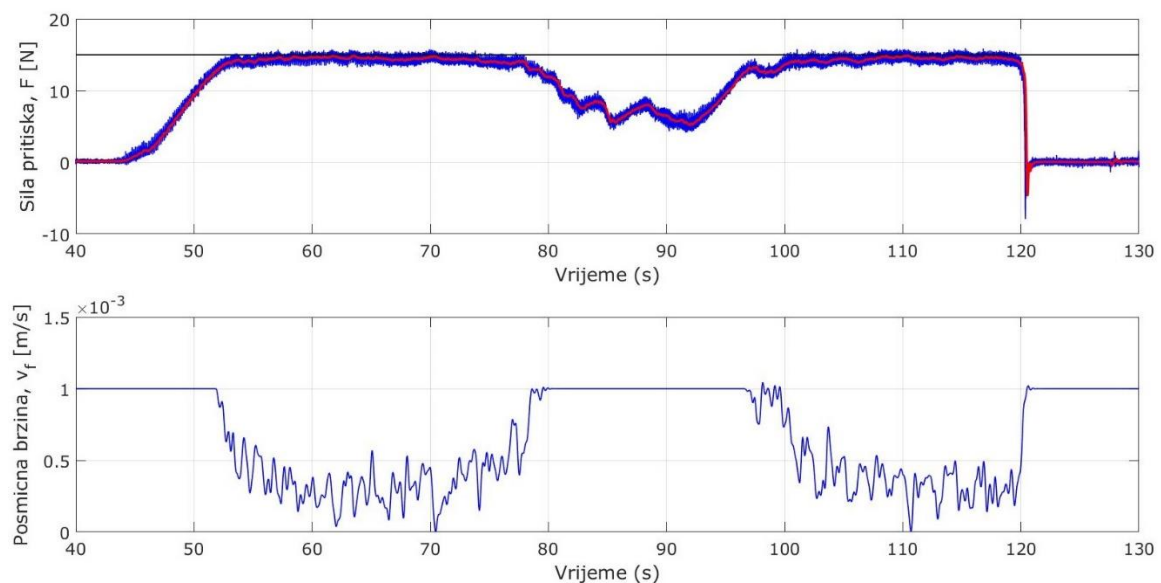
Slika 52. Teleća lopatična kost

Sama kost nema jednaku strukturu rasprostranjenu po cijelome svome volumenu. Širi i plosnati dio lopatične kosti ima jasno izražena tri strukturalna sloja, no kako se kost sužuje prema lopatičnome zglobu, srednji sloj postaje sve tanji i neizraženiji, dok u konačnici uz sami zglob kost ne postane u potpunosti kompaktna bez srednjeg mekanoga sloja.



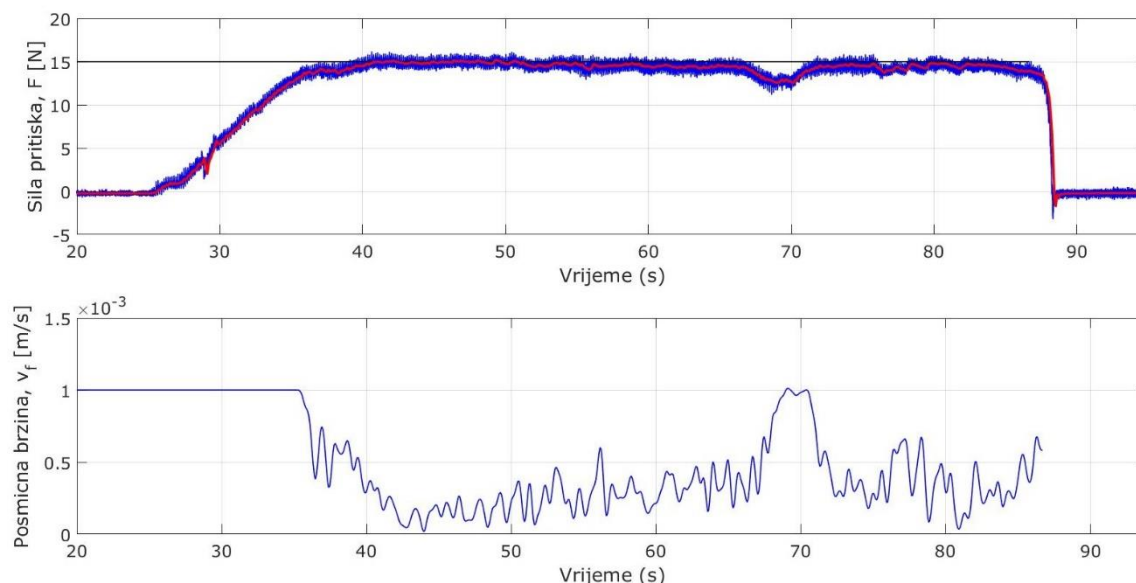
Slika 53. Sila pritiska i posmična brzina bušenja; kompaktna kost; $K_p = 40$, $K_i = 20$

Na slici 53. prikazani su rezultati bušenja teleće lopatične kosti, njenoga kompaktnoga dijela. Svrđlo se nije zaustavilo nakon proboja kosti, iako je algoritam za detekciju proboja bio uključen. Razlog? Algoritam je detektirao povećanje sile pritiska, dakle prvi tvrdi sloj. No kako je mekaniji srednji sloj izostao, pad sile prilikom proboja svrdla algoritam je protumačio kao granicu između prvoga i drugoga sloja i nastavio proces bušenja. U nastavku je obavljen veći broj bušenja u predjelu kosti sa nehomogenom strukturom, dakle na širem i plosnatijem dijelu kosti.



Slika 54. Sila pritiska i posmična brzina bušenja; kost ; $K_p = 40$, $K_i = 20$

Na slici 54. prikazani su rezultati bušenja teleće lopatične kosti, gdje je uočljiv pad sile pritiska u predjelu srednjeg mekanijeg dijela kosti. No algoritam za detekciju proboja niti ovoga puta nije zaustavio svrdlo nakon proboj, iako ovoga puta postoji izražen pad sile pritiska. Razlog je u pregrubo podešenim parametrima algoritma, tj. granične vrijednosti sile ($th1, th2, th3$ i $th4$) podešeni su prema skoro idealnome primjeru pada sile uslijed nehomogene strukture, a to je nehomogena struktura lim-eps-lim.



Slika 55. Sila pritiska i posmična brzina bušenja; kost;

$K_p = 40, K_i = 20$; detekcija proboja

Na slici 55. prikazani su rezultati bušenja lopatične kosti sa pravovremenim zaustavljanjem. Iako je pad sile uzrokovan srednjim mekanijim slojem relativno mali u odnosu na ostale primjere, sa dobro podešenim graničnim vrijednostima sile algoritam je detektirao sva tri sloja kosti, te proboj svrdla.

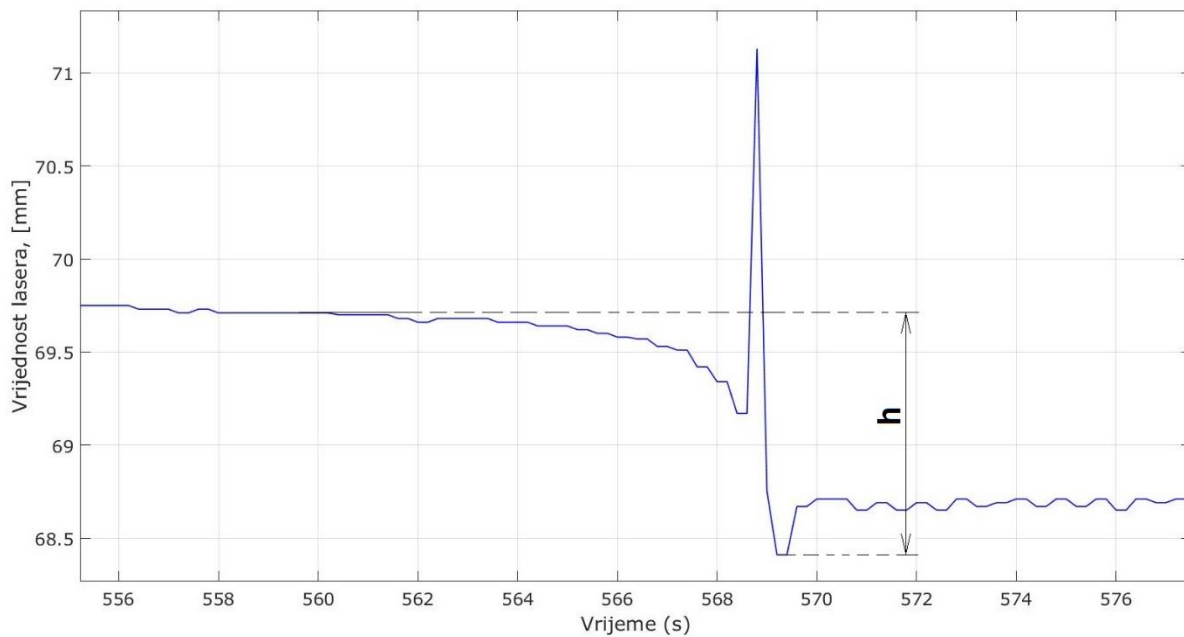
Poželjno je da hod svrdla nakon proboja kosti bude što manji. Kod korištenja ovakve vrste algoritma za detekciju proboja, duljinu hoda svrdla nakon proboja određuje parametar algoritma $th4$. Odnosno granična sila pritiska prilikom koje algoritam zaustavlja translacijsko gibanje svrdla i vraća svrdlo u početnu poziciju. Ukoliko je vrijednost parametra preniska, dolazi do velikoga hoda nakon proboja, a isto tako ukoliko je parametar podešen na preveliku vrijednost postoji mogućnost preranoga zaustavljanja svrdla, prije nego li je uopće probilo kost. Prilikom uzastopnih bušenja teleće lopatične kosti ustanovljeno je da algoritam vrši sigurnu detekciju proboja prilikom sljedećih iznosa parametara: $th1=Fn$; $th2=0.85*Fn$; $th3=Fn$; $th4=0.2*Fn$, gdje je Fn željena sila pritiska svrdla prilikom bušenja.

U svrhu kvantifikacije hoda svrdla nakon proboja korišten je laserski mjerač udaljenosti AccuRange AR700. AR700 je triangulacijski senzor za mjerenje udaljenosti proiciranjem laserske zrake na željeni objekt. Na slici 56. prikazan je postupak bušenja teleće lopatične kosti, te laserski mjerač udaljenosti koji projicira lasersku zraku na površinu kosti na točno mjesto potencijalnoga proboja.



Slika 56. Kvantifikacija duljine hoda svrdla prilikom proboja

Prije postavljanja kosti u stezni element, laserski mjerač je pričvršćen na nosač ispod svrdla, te su orijentacija laserske zrake i orijentacija svrdla podešeni na taj način da je laserska zraka proicirana na vrh svrdla bez obzira na gibanje svrdla u aksijalnome smjeru, odnosno u smjeru bušenja. Zatim je postavljena kost u stezni element, nakon čega laser proicira zraku na kost umjesto na svrdlo, odnosno na potencijalno mjesto proboja. Laserski mjerač udaljenosti pohranjuje vrijednosti udaljenosti kosti u tekstualnu datoteku na računalu. Ukoliko dođe do proboja, laserska zraka je opet proicirana na vrh svrdla, te dolazi do promjene vrijednosti očitane udaljenosti na laserskom mjeraču. Razlika između vrijednosti udaljenosti prije samoga proboja proicirane laserske zrake na kost i proicirane laserske zrake na svrdlo nakon proboja daje nam podatak o hodu (h) svrdla nakon proboja, slika 57..



Slika 57. Udaljenost očitana sa laserskoga mjerača udaljenosti

Ponavljajući bušenja sa istim parametrima PI regulatora i sa istim parametrima algoritma, vrijednosti hoda nakon proboja svrdla kreću se od 0,9 do 1,5 mm. Duljina hoda ovisi o kompaktnosti kosti na dijelu na kojem je obavljeno bušenje. Povećanjem parametra th_4 unutar algoritma za detekciju proboja hod svrdla nakon proboja bih se smanjio, ali isto tako postojala bih mogućnost da niti ne dođe do proboja.

8 ZAKLJUČAK

U ovome radu opisan je laboratorijski postav za osjetljivo robotsko bušenje. Navedeni su dijelovi i definirane su komunikacijske veze između pojedinih sustava. Postav je testiran bušeći umjetnu nehomogenu strukturu (lim - eps - lim), te teleću lopatičnu kost.

U poglavlju 2. *Nehomogeni materijali* prezentirana je definicija nehomogenih materijala, te su kao primjer nehomogenih materijala navedene ljudske i životinjske kosti. Ukratko je opisana struktura kosti i njihova podijela.

U poglavlju 3. *Bušenje* opisan je postupak obrade odvajanjem čestica – bušenje. Navedena su i prikazana opterećenja na svrdlo prilikom procesa bušenja, te su navedeni osnovni parametri samoga procesa. Navedeni su standardi koje moraju zadovoljiti svrdla korištena u medicinske svrhe i zahtjevi koji se pred njih stavljaju. U poglavlju 4. *Matematički kodel bušenja* – opisan je pojednostavljeni matematički model bušenja.

U poglavlju 5. *Laboratorijski postav za bušenje* opisani su dijelovi postava, komunikacijske veze i upravljački sustav. U poglavlju 5.1 *Komunikacija* dana je definicija računalnih mreža, te su definirani osnovni pojmovi vezani uz računalne mreže. Opisan je OSI referentni model i njegovih sedam razina koje u teoriji definiraju komunikaciju unutar računalnih mreža. Navedene su i opisane podjele računalnih mreža prema fizičkoj topologiji, prema veličini te prema funkcionalnosti. Detaljnije je opisan Ethernet protokol kao najrašireniji protokol unutar lokalnih mreža – LAN-a (eng. *Local Area Network*). Opisan je DoD mrežni model. Uspoređene su četiri razine komunikacije Dod modela sa sedam razina OSI referentnog modela. Opisana je porodica TCP/IP protokola, te je detaljnije objašnjen princip rada IP, UDP i TCP protokola. U poglavlju 5.2. *Net F/T* opisan je sustav za mjerenje sile i momenta. Navedene su glavne komponente sustava, te su prikazane osnovne tehničke karakteristike. Detaljno je opisan postupak ostvarivanja komunikacije sa računalom. U poglavlju 5.3 *Robotski sustav Kuka Lightweight robot LWR4+* navedene su osnovne tehničke karakteristike robotskog sustava, te je detaljnije opisana robotska ruka LWR4+. Opisan je postupak ostvarivanja komunikacije sa računalom preko FRI sučelja.

U poglavlju 6. *Kalibracija alata* opisani su postupci kalibracije TCP-a senzora sile i TCP-a svrdla, te isto tako i kalibracije orijentacije senzora sile i svrdla. prezentiran je postupak rotacije komponenti sile iz koordinatnog sustava senzora u koordinatni sustav svrdla.

U poglavlju 7. *Eksperimentalni rezultati* – prikazani su rezultati bušenja različitih homogenih i nehomogenih materijala, između ostaloga i teleće lopatične kosti.

Korišteni PI regulator sile pritiska svrdla sa dobro podešenim parametrima iznenađujuće uspješno održava silu pritiska svrdla konstantnom. Mana ovakvoga načina regulacije sile pritiska svrdla leži u tome što je unaprijed potrebno definirati osim željene sile pritiska svrdla i maksimalnu posmičnu brzinu svrdla. Postoji mogućnost da željena sila pritiska svrdla prilikom bušenja nebude ostvarena ukoliko je zadana premala maksimalna posmična brzina svrdla.

Algoritam za detekciju proboja pokazao se prihvatljiv za nehomogene materijale kod kojih postoji jasno izražena razlika u svojstvima pojedinih slojeva. Ukoliko je ta razlika slabo izražena ili u krajnjem slučaju na području bušenja imamo kompaktan materijal tada ovakva vrsta algoritma za detekciju proboja neće zaustaviti svrdlo prilikom proboja, već će nastaviti proces bušenja.

Područje osjetljivog robotskog bušenja nehomogenih materijala u medicinske svrhe nudi velike prilike za daljnja istraživanja. Kako u smjeru pronalaženja robusnijih i stabilnijih načina održavanja konstantne sile pritiska prilikom bušenja, tako i u osmišljavanju kvalitetnijih i pouzdanijih metodama detekcije proboja, te minimizacije hoda prilikom proboja. Smjer u kojem se kreću trenutna istraživanja zasniva se na korištenju nekih od grana umjetne inteligencije poput umjetnih neuronskih mreža ili upravljanja zasnovanog na fuzzy logici.

LITERATURA

- [1] Feußner ,H.;Park, A.:”Surgery 4.0: the natural culmination of the industrial revolution?”, <https://doi.org/10.1515/iss-2017-0036>
- [2] Wolf,B.;Scholze,C.:“Medicine 4.0“, <https://doi.org/10.1515/cdbme-2017-0038>
- [3] Medicinska enciklopedija, Leksikografski zavod FNRJ, Zagreb, 1962.g
- [4] Augustin,G.:Termičko oštećenj kosti tijekombušenja dvostupnjevanim svrdlom i svrdlom s protočnim hlađenjem, Disertacija, Sveučilište u Zagrebu, Medicinski fakultet, 2011.g
- [5] Jerbić, B.: Predavanja iz predmeta Računalne mreže, e-učenje, 2018.
- [6] <https://sysportal.carnet.hr/node/343>, 10.2.2018.
- [7] Mujarić,E.: <http://mreze.layer-x.com> , 12.2.2018.
- [8] ATI,: Network Force/Torque Sensor System, Compilation of Manuals
- [9] KUKA,: System Software 5.6, 23.07.2010.
- [10] KUKA,:FastResearchInterface 1.0, 16.09.2011.
- [11] Medtronic: Integrated Power Console (IPC™), Service manual console and attachments
- [12] MacAvelia.T.:Force and torque modeling for the drilling of bone for use in orthopaedic haptic simulation systems, 1.1.2012.

PRILOZI

Glavna C++ skripta

```
#include <stdio.h>
#include <windows.h>
#include <process.h>           // needed for _beginthread()
#include <iostream>

#include <cstdlib>
#include <math.h>
#include <limits.h>

using namespace std;
void pozivfri( void * ); // function prototype
void fri(void *argc, void *argv[]);
void upravljanje(void *P);
void netF(void *P);
float d;

int main()
{
    cout<<"Debljina bušene strukture?:"<<endl;
    cin>>d;

    HANDLE hThreads[3];
    hThreads[0] = (HANDLE)_beginthread( upravljanje, 0, (void*)0 );
    hThreads[1] = (HANDLE)_beginthread( netF, 0, (void*)0 );
    hThreads[2] = (HANDLE)_beginthread( pozivfri, 0, (void*)12 );

    WaitForMultipleObjects(3, hThreads, FALSE, INFINITE);
    Sleep( 100 );

    cout<<"Gotov proces";
    Sleep( 3000 );
}

void pozivfri( void *arg )
{
    printf( "Pozivanje FRI", (INT_PTR)arg );
    fri(NULL, NULL);
}
```

C++ Komunikacija FRI

```

#ifdef __WIN32
#include "stdafx.h"
#endif

#ifndef PI
#define PI 3.1415926535897932384626433832795
#endif

#include <iostream>
#include <fstream>
#include <cstdlib>
#include <math.h>
#include <limits.h>
#include <conio.h>
#include <string>
#include <stdio.h>
#include <time.h>
#include "friudp.h"
#include "friremote.h"

using namespace std;

extern friRemote friInst;
extern int ov_pro;
extern float jntSpeed[LBR_MNJ];
extern float mojjoint[LBR_MNJ];
extern float mojjointcart[FRI_CART_FRM_DIM];
extern bool rub4,k,nastavak,z;
bool gotovo;
float poz[6];
const std::string currentDateTime() {
    time_t now = time(0);
    struct tm tstruct;
    char buf[80];
    tstruct = *localtime(&now);
    strftime(buf, sizeof(buf), "%Y-%m-%d.%X", &tstruct);

    return buf;
}

void fri(void *argc, void *argv[])
{
    cout << "Opening FRI Version " << FRI_MAJOR_VERSION << "." << FRI_SUB_VERSION <<
    "." << FRI_DATAGRAM_ID_CMD << "." << FRI_DATAGRAM_ID_MSR << " Interface for
    First Sample" << endl;

    // do checks, whether the interface - and the host meets the requirements
    // Note:: This Check remains in friRempte.cpp -- should go to your code
    ...
    FRI_PREPARE_CHECK_BYTE_ORDER;
    if (!FRI_CHECK_BYTE_ORDER_OK)
    {
        cerr << "Byte order on your system is not appropriate - expect
    deep trouble" <<endl;
    }
    if (!FRI_CHECK_SIZES_OK)

```

```

        cout << "Sizes of datastructures not appropriate - expect even
deeper trouble" << endl;

    //friRemote friInst;
    FRI_CTRL lastCtrlScheme = FRI_CTRL_CART_IMP;
    FRI_QUALITY lastQuality = FRI_QUALITY_BAD;
    double timeCounter=0;
    ofstream myfile;

    /* enter main loop - wait until we enter stable command mode */
    do{

        mode already

        /// perform some arbitrary handshake to KRL -- possible in monitor

        // send to krl int a value

        if ( friInst.getQuality() >= FRI_QUALITY_OK)
        {
            // send a second marker
            friInst.setToKRLInt(0,10);
        }

        if (friInst.doReceiveData())>=0)
            {
                friInst.setToKRLInt(4,ov_pro);
                friInst.setToKRLBool(1,rub4);
                friInst.setToKRLBool(2,nastavak);
                friInst.doSendData();

            }
            else
            {
                cout << "greska!";
            }
            gotovo=friInst.getFrmKRLBool(0);
            poz[0]=friInst.getFrmKRLReal(0);
            poz[1]=friInst.getFrmKRLReal(1);
            poz[2]=friInst.getFrmKRLReal(2);
            poz[3]=friInst.getFrmKRLReal(3);
            poz[4]=friInst.getFrmKRLReal(4);
            poz[5]=friInst.getFrmKRLReal(5);
        // Call to data exchange - and the like
        //friInst.doDataExchange();
        int divider = (int)( 1./friInst.getSampleTime()) *20.0);
        if ( friInst.getSequenceCount() % divider == 0)
        {
            //cout << "Quality:"<<friInst.getQuality() << ", " ;
            //cout << "State:"<<friInst.getState()<< "\n" ;
            //cout << "Motor:"<<friInst.isPowerOn()<< "\n" ;
        }
        // Stop request is issued from the other side
        if ( friInst.getFrmKRLInt(0) == -1)
        {
            cout << "leaving \n";
            break;
        }

    }while (k==0);
    cout << "Izlazimo iz podfunkcije \n";
    return;
}

```


C++ Komunikacija Net F/T

```

#include <winsock2.h>
#include <ws2tcpip.h>
#pragma comment(lib, "Ws2_32.lib")
#include <sys/types.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <iostream>
#include <cstdio>
#include <ctime>
#define PORT 49152 /* Port the Net F/T always uses */
#define COMMAND 2 /* Command code 2 starts streaming */
#define NUM_SAMPLES 1 /* Will send 1 sample before stopping */

/* Typedefs used so integer sizes are more explicit */
typedef unsigned int uint32;
typedef int int32;
typedef unsigned short uint16;
typedef short int16;
typedef unsigned char byte;
typedef struct response_struct {
    uint32 rdt_sequence;
    uint32 ft_sequence;
    uint32 status;
    int32 FTData[6];
} RESPONSE;

float ftt[3];
float ft[6];
extern bool k;
void netF(void *)
{
    double t=0;
    std::clock_t start;
    FILE *f;
    f= fopen("F_svrkla.txt","w");
    fclose(f);
    f= fopen("F_svrkla.txt","a");

    float ftt_old=0,dFs;

    SOCKET socketHandle; /* Handle to UDP socket used to communicate with Net
F/T. */
    WSADATA wsaData;
    WORD wVersionRequested;

    struct sockaddr_in addr; /* Address of Net F/T. */
    struct hostent *he; /* Host entry for Net F/T. */
    byte request[8]; /* The request data sent to the Net F/T. */
    RESPONSE resp; /* The structured response received
from the Net F/T. */
    byte response[36]; /* The raw response data received from the Net
F/T. */
    int i; /* Generic loop/array index. */
    int err; /* Error status of operations. */

    char * AXES[] = { "Fx", "Fy", "Fz", "Tx", "Ty", "Tz" }; /* The names of the
force and torque axes. */

```

```

    /*if ( 2 > argc )
    {
        fprintf( stderr, "Usage: %s IPADDRESS\n", argv[0] );
        return -1;
    }
    */

    wVersionRequested = MAKEWORD(2, 2);
    WSStartup(wVersionRequested, &wsaData);

    start=std::clock();
    do{

        /* Calculate number of samples, command code, and open socket here. */
        label:
        socketHandle = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
        if (socketHandle == -1) {

            goto label;
        }
        *(uint16*)&request[0] = htons(0x1234); /* standard header. */
        *(uint16*)&request[2] = htons(COMMAND); /* per table 9.1 in Net F/T user manual.
    */
        *(uint32*)&request[4] = htonl(NUM_SAMPLES); /* see section 9.1 in Net F/T user
    manual. */
        /* Sending the request. */
        he = gethostbyname("192.168.1.10");
        memcpy(&addr.sin_addr, he->h_addr_list[0], he->h_length);
        addr.sin_family = AF_INET;
        addr.sin_port = htons(PORT);
        err= connect( socketHandle, (struct sockaddr *)&addr, sizeof(addr) );
        if (err == -1) {
            closesocket(socketHandle);

            goto label;
        }
        send( socketHandle, (const char *)request, 8, 0 );
        /* Receiving the response. */
        recv( socketHandle, (char *)response, 36, 0 );
        resp.rdt_sequence = ntohl(*(uint32*)&response[0]);
        resp.ft_sequence = ntohl(*(uint32*)&response[4]);
        resp.status = ntohl(*(uint32*)&response[8]);
        for( i = 0; i < 6; i++ ) {
            resp.FTData[i] = ntohl(*(int32*)&response[12 + i * 4]);
            ft[i]=float(resp.FTData[i])/1000000;
        }
        closesocket(socketHandle);
        ftt[0]=-0.2237*ft[0]-0.9303*ft[1]+0.2908*ft[2]; // F_x svrdla
        ftt[1]=-0.7203*ft[0]+0.3588*ft[1]+0.5936*ft[2]; // F_y svrdla
        ftt[2]=-0.6566*ft[0]-0.0767*ft[1]-0.7504*ft[2]; // F_z svrdla

        double dt = ( std::clock() - start ) / (double) CLOCKS_PER_SEC;
        start=std::clock();
        dFs=(-ftt[0])-ftt_old)/(dt);
        ftt_old=-ftt[0];
        t=t+dt;
        fprintf(f, "%f %f %f %lf %lf \n", ftt[0], ftt[1], ftt[2], dt, t);
        //system("CLS");
    }while(k==0);
    fclose (f);
    return;}

```

C++ Upravljanje

```

#include <winsock2.h>
#include <ws2tcpip.h>
#pragma comment(lib, "Ws2_32.lib")
#include <Windows.h>
#include <sys/types.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include <iostream>
#include <cstdio>
#include <ctime>
using namespace std;
extern float d;
int stanje;
int ov_pro;
extern float ftt[3];
extern float ft[6];
bool rub4=false,k=0,nastavak=1;
extern bool gotovo;
extern float poz[6];

void upravljanje (void *)
{
    int dtt,Kp=45,Kd=0,Ki=20,p1=0,Fn=15;
    float x1,x2,y1,y2,z1,z2,d_rob,e,de,ei=0,e_old=0,F_s_old=0,dF,F_s,kva,er=0;

    double ov,v,t=0;
    double dt=0,FS,FSV;
    bool rub1=false,rub2=false,rub3=false,sloj1=false,sloj2=false,sloj3=false;
    double v_ref=0.003;
    std::clock_t start;
    FILE *f;
    f= fopen("Data.txt", "w");
    fclose(f);
    f= fopen("Data.txt", "a");

    start=std::clock();
do{
    F_s=-ftt[0];
    e=Fn-F_s;
    ei=ei+e*dt;
    if(ei>1)
        ei=0;
    if (ei<-1)
        ei=0;
    de=(e-e_old)*dt;

    ov=e*Kp+ei*Ki;
    if (ov>100)
        ov_pro=100;
    else if (ov<0)
        ov_pro=0;
    else
        ov_pro=(int)ov;

    FS=sqrt((ftt[0]*ftt[0])+(ftt[1]*ftt[1])+(ftt[2]*ftt[2]));
    FSV=sqrt((ft[0]*ft[0])+(ft[1]*ft[1])+(ft[2]*ft[2]));

```

```

cout<<"SVRDLO="<<FS<<"\n"<<endl;
cout<<"Fx="<<ftt[0]<<"\n"<<endl;
cout<<"Fy="<<ftt[1]<<"\n"<<endl;
cout<<"Fz="<<ftt[2]<<"\n"<<endl;
/*cout<<"SENZOR="<<FSV<<"\n"<<endl;
cout<<"Fx="<<ft[0]<<"\n"<<endl;
cout<<"Fy="<<ft[1]<<"\n"<<endl;
cout<<"Fz="<<ft[2]<<"\n"<<endl;*/

cout<<"ov="<<ov<<"\n"<<endl;
cout<<"$OV_PRO="<<ov_pro<<"\n"<<endl;
cout<<"\nX="<<poz[0]<<endl;
cout<<"\nY="<<poz[1]<<endl;
cout<<"\nZ="<<poz[2]<<endl;
cout<<"\nA="<<poz[3]<<endl;
cout<<"\nB="<<poz[4]<<endl;
cout<<"\nC="<<poz[5]<<endl;
if (F_s>0.8)
    {
        rub1=true;
        p1++;
    }
if(p1==1)
    {
        x1=poz[0];
        y1=poz[1];
        z1=poz[2];
    }
if (F_s>Fn)
    {
        sloj1=true;
    }
if ((sloj1==true) && (F_s<Fn*0.85))
    {
        rub2=true;
    }
if ((rub2==true) && (F_s>Fn))
    {
        rub3=true;
    }

if ((rub3==true) && (F_s<(Fn*0.2)))
    {
        rub4=true;
        ov_pro=0;
        x2=poz[0];
        y2=poz[1];
        z2=poz[2];
    }
kva=(x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)+(z2-z1)*(z2-z1);
d_rob=sqrt(kva);
er=d_rob-d;
cout<<"\nGreška proboja ="<<er<<endl;
cout<<"\n Za povratak u pocetnu poziciju pritisnite NULU [0]:";
cin>>nastavak;

do{
    ov_pro=50;
}while(gotovo==1);
k=1;
}

dt = ( std::clock() - start ) / (double) CLOCKS_PER_SEC;
t=t+dt;

```

```

    dF=(F_s-F_s_old)/dt;
    start=std::clock();
    fprintf(f,"%f %d %1f %1f %f %f %f %f \n",er,ov_pro, dt,t,dF,F_s,ftt[1],ftt[2]);
    cout<<"rub1="<<rub1<<endl;
    cout<<"sloj1="<<sloj1<<endl;
    cout<<"rub2="<<rub2<<endl;
    cout<<"rub3="<<rub3<<endl;
    cout<<"rub4="<<rub4<<endl;
    system("CLS");
    e_old=e;
    F_s_old=F_s;
}while(k==0);
    fclose (f);

    return;
}

```

Matlab - matrica rotacije

```

A=165.495;
B=-16.058;
C=141.136;
alfa =degtorad(A);
betha =degtorad(B);
gamma = degtorad(C);
x=-181.112;
y=37.819;
z=236.017;
tran=[1 0 0 x ;0 1 0 y ;0 0 1 z;0 0 0 1];
rotz=[cos(alfa) -sin(alfa) 0 ;sin(alfa) cos(alfa) 0 ;0 0 1];
roty=[cos(betha) 0 sin(betha);0 1 0;-sin(betha) 0 cos(betha)];
rotx=[1 0 0 ; 0 cos(gamma) -sin(gamma); 0 sin(gamma) cos(gamma)];

TR=rotz*roty*rotx;
As=-91.130;
Bs=0.284;
Cs=-0.940;
alfaS =degtorad(As) ;
bethaS = degtorad(Bs);
gammaS = degtorad(Cs);
xs=0.310;
ys=0.034;;
zs=49.937;

rotzS=[cos(alfaS) -sin(alfaS) 0;sin(alfaS) cos(alfaS) 0;0 0 1 ];
rotyS=[cos(bethaS) 0 sin(bethaS) ;0 1 0;-sin(bethaS) 0 cos(bethaS)];
rotxS=[1 0 0 ; 0 cos(gammaS) -sin(gammaS); 0 sin(gammaS) cos(gammaS)];

SR=inv (rotzS*rotyS*rotxS);

R=SR*TR;

```

KRL – linearno gibanje po osi svrdla

```
DEF rorbach (S:out,ST:out)
```

```
DECL E6POS ST,S
```

```
INI
```

```
PTP ST CONT Vel=50% PDAT1 Tool[13]:rorbach_bušilica Base[1]:agi-lwr
```

```
LIN A CONT Vel=1 m/s CPDAT9 Tool[13]:rorbach_bušilica Base[1]:agi-lwr
```

```
INTERRUPT DECL 1 WHEN $fri_frm_bool[2]==true do kraj()
```

```
gibanje(s)
```

```
wait for $fri_frm_bool[3]==false
```

```
LIN ST CONT Vel=1 m/s CPDAT9 Tool[13]:rorbach_bušilica Base[1]:agi-lwr
```

```
$fri_to_bool[1]=true
```

```
END
```

```
-----  
DEF gibanje (s:out)
```

```
DECL E6POS S
```

```
INTERRUPT on 1
```

```
LIN S CONT Vel=0.001 m/s CPDAT10 Tool[13]:rorbach_bušilica Base[1]:agi-lwr
```

```
INTERRUPT off 1
```

```
RESUME
```

```
END
```

```
-----  
DEF KRAJ()
```

```
BRAKE
```

```
RESUME
```

```
END
```