

# Dizajn, proračun i izrada paralelnog robotskog manipulatora korištenjem aditivnih proizvodnih tehnologija

---

**Miškulin, Petar**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:333809>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-11-01**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**DIZAJN, PRORAČUN I IZRADA PARALELNOG  
ROBOTSKEG MANIPULATORA KORIŠTENJEM  
ADITIVNIH PROIZVODNIH TEHNOLOGIJA**

Rijeka, rujna 2024.

Petar Miškulin,

0069081203

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**DIZAJN, PRORAČUN I IZRADA PARALELNOG  
ROBOTSKOG MANIPULATORA KORIŠTENJEM  
ADITIVNIH PROIZVODNIH TEHNOLOGIJA**

Mentor: prof. dr. sc. Zlatan Car

Komentor: Sandi Baressi Šegota mag. ing. comp.

Rijeka, rujna 2024.

Petar Miškulin,

0069081203

Rijeka, 14.03.2024.

Zavod: Zavod za automatiku i elektroniku  
Predmet: Osnove robotike

## ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Petar Miškulin (0069081203)**  
Studij: Sveučilišni diplomski studij elektrotehnike (1300)  
Modul: Automatika (1331)

Zadatak: **Dizajn, proračun i izrada paralelnog robotskog manipulatora korištenjem aditivnih proizvodnih tehnologija / Design, calculation and creation parallel robotic manipulators using additive production technologies**

### Opis zadatka:

Izvršiti pregled literature kontrolu i dizajn paralelnih robotskih manipulatora. Kreirati CAD model industrijskog robotskog manipulatora, te 3D printanjem kreirati dijelove istog. Sastaviti manipulator, te izvršiti proračun direktne i inverzne kinematike. Izvršiti osnovno programiranje za pozicioniranje vrha manipulatora u prostoru. Diplomski rad mora biti napisan prema uputama za pisanje diplomskih radova koje su objavljene na mrežnim stranicama studija.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:  
prof. dr. sc. Zlatan Car

Komentor:  
Sandi Baressi Šegota

Predsjednik povjerenstva za  
diplomski ispit:  
prof. dr. sc. Dubravko Franković

## IZJAVA O SAMOSTALNOJ IZRADI DIPLOMSKOG RADA

Izjavljujem da sam ja, Petar Miškulin, sukladno pravilniku Tehničkog fakulteta Sveučilišta u Rijeci samostalno napisao i izradio ovaj diplomski rad *Dizajn, proračun i izrada paralelnog robotskog manipulatora korištenjem aditivnih proizvodnih tehnologija* uz pomoć niže navedene literature.

Rijeka, rujna 2024.

Petar Miškulin,

0069081203



## **ZAHVALA**

Zahvaljujem obitelji i prijateljima koji su mi pružali izuzetnu podršku kroz sve godine studija.

Također želim zahvaliti svome mentoru prof. dr. sc. Zlatanu Caru i komentoru Sandiu Baressiu Šegoti mag. ing. comp., za stručno vođenje pri izradi ovog rada.

## Sadržaj

<b>1. UVOD</b> .....	2
<b>2. PODJELA I RAZVOJ ROBOTSKIH MANIPULATORA</b> .....	4
<b>2.1. Paralelni manipulatori</b> .....	5
<b>2.2. Usporedba paralelnih i serijskih manipulatora</b> .....	6
<b>3. PRINCIP RADA DELTA ROBOTA</b> .....	8
<b>3.1. Zašto koristiti Delta robote?</b> .....	9
<b>4. IZRADA TIJELA DELTA ROBOTA</b> .....	10
<b>4.1. Servo motori</b> .....	10
4.1.1. Princip rada Servo motora .....	10
4.1.2. Teorija PWM upravljanja .....	11
4.1.3. Upravljanje Servo motorima .....	13
4.1.4. Podjela i primjena Servo motora.....	14
4.1.5. Odabir Servo motora za pokretanje Delta robota.....	15
<b>4.2. Arduino mikrokontroler</b> .....	16
4.2.1. Tehničke specifikacije Arduino Uno mikrokontrolera .....	17
<b>4.3. 3D printanje i fizičke dimenzije robota</b> .....	20
4.3.1. Izračun fizičke dimenzije dijelova robota.....	21
4.3.2. Ispis i nacrti 3D dijelova Delta robota .....	22
<b>4.4. Električna shema spajanja Servo motora i Arduino razvojne pločice</b> .....	28
<b>4.5. Sastavljanje i spajanje robota</b> .....	29
<b>5. PROGRAMIRANJE I TESTIRANJE DELTA ROBOTA</b> .....	34
<b>5.1. Funkcija izračuna inverzne kinematike</b> .....	35
<b>5.2. Opis programskog koda Delta robota</b> .....	43
<b>5.3. Kalibracija Servo motora na apsolutnu nulu</b> .....	51
<b>6. ZAKLJUČAK</b> .....	53
<b>7. LITERATURA</b> .....	54
<b>8. SAŽETAK I KLJUČNE RIJEČI NA HRVATSKOM I ENGLESKOM JEZIKU</b> .....	56
<b>9. PRILOG A – CIJELI PROGRAMSKI KOD ARDUINO MIKROKONTROLERA</b> .....	57
<b>10. PRILOG B – KOTIRANI NACRTI BAZE</b> .....	63
<b>11. PRILOG C – KOTIRANI NACRTI END EFFECTORA</b> .....	64
<b>12. PRILOG D – KOTIRANI NACRTI NOSAČA BAZE</b> .....	65
<b>12. PRILOG E – KOTIRANI NACRTI KRAĆEG KRAKA ROBOTA</b> .....	66

**13. PRILOG F – KOTIRANI NACRTI DULJEG KRAKA ROBOTA ..... 67**

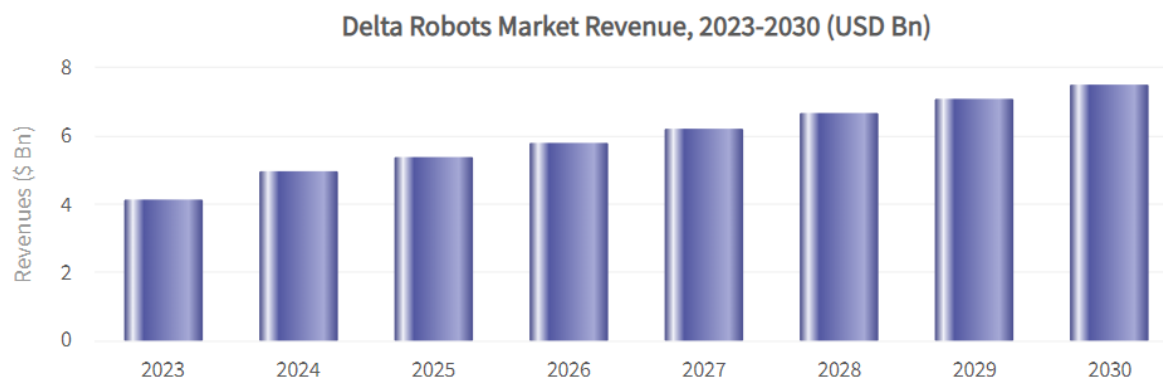


## 1. UVOD

Kao društvo sve više težimo automatizaciji različitih procesa, što radi brzine i učinkovitosti, ili radi cijene. U takvom okruženju došlo je do razvoja takozvanih Delta robota koji postaju sve značajniji u industrijskim primjenama zbog svoje brzine, preciznosti i fleksibilnosti u obavljanju različitih zadataka. Njihov jedinstveni dizajn omogućava brzo i efikasno obavljanje složenih operacija, što smanjuje vrijeme ciklusa i povećava produktivnost. Delta roboti su posebno korisni za zadatke kao što su pakiranje, sortiranje, montaža i pick-and-place operacije, jer se odlikuju velikom brzinom i točnošću.

Ovaj diplomski rad opisat će princip rada Delta robota te izraditi stolnu verziju robota kako bi se fizički prikazao princip rada. U radu će se opisati teorijska osnova rada Delta robota, uključujući proračune kinematike i dinamike, kao i prednosti koje njihova specifična struktura donosi u odnosu na druge tipove robota. Nakon toga bit će prikazan postupak izrade, programiranja i puštanja u rad jednog takvog stolnog robota te svi njegovi dijelovi od mikrokontrolera koji upravlja njime do end effectora. Vizualizacija i interakcija s robotom realizirat će se putem intuitivnog korisničkog sučelja koje operaterima omogućuje jednostavno upravljanje i praćenje statusa robota u stvarnom vremenu.

Globalno tržište Delta robota (Slika 1.1.) procijenjeno je na 4,14 milijardi USD u 2022. godini, a očekuje se da će dosegnuti 5,51 milijardu USD do 2030.



*Slika 1.1. Globalno tržište Delta robota, 2023-2030 (milijarde USD) [8]*

Sve veće prihvaćanje automatizacije unutar prehrambene i farmaceutske industrije daje direktan doprinos povećanju tržišta Delta robota. Ove industrije svjedoče sve većem pomaku prema automatizaciji da bi mogle zadovoljiti rastuće zahtjeve potrošača, učinkovitosti i strogo održavanje kvalitete. Najveća kočnica u rastu tog tržišta je visoka cijena samih robota. Iako su poznati po svojoj preciznosti i učinkovitosti, iste karakteristike često dolaze s velikom početnom cijenom ulaganja u takav sustav, posebno u malim i srednjim poduzećima. To izravno utječe na period povrata uloženih sredstava, što znači da si samo veće korporacije to mogu priuštiti i imati koristi od ove napredne tehnologije.

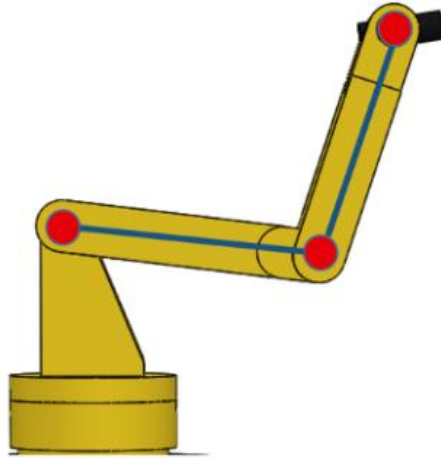
## 2. PODJELA I RAZVOJ ROBOTSKIH MANIPULATORA

Jedna definicija robota je da isti predstavlja integrirani mehatronički sustav koji se sastoji od tri podsustava: upravljačkog, senzorskog i aktuatorskog. Upravljački sustav predstavlja sam „mozak“ robota koji kontrolira tijekom radnji na temelju upravljačkog programa i povratnih informacija senzora iz senzorskog podsustava koji prikuplja parametre iz okoline. Konačno, aktuatorski podsustav koristi se za utjecaj na okolinu robota preko manipulatora.

Robotski manipulator je dio robota koji nalikuje ljudskoj ruci i koja je pričvršćena za tijelo robota te služi za izvršavanje zadataka. Bez nje, robot koji izvršava neku radnju bi bio usporediv s čovjekom kojemu su ruke vezane iza leđa. Robotski manipulatori imaju mnoge primjene u industriji, kao što su:

- „Pick and Place“ – često se koriste za izvršavanje unaprijed programiranih istih niza pick and place radnji kao što su pakiranje artikala;
- robotska kirurgija – posljednjih se godina roboti koriste u malim medicinskim operacijama, a u budućnosti, napretkom robotske tehnologije, će se koristiti i u složenijim zahvatima koji zahtijevaju veću preciznost;
- žetva i branje – robote možemo koristiti za poljoprivredne primjene poput žetve i branja. Ovo je područje pogodno za automatizaciju jer su roboti učinkoviti, fleksibilni, točni i brzi te mogu poboljšati ukupnu proizvodnju;
- zavarivanje – u procesu zavarivanja potrebna je visoka razina obuke i znanja radnika što roboti mogu savladati i točnije ponavljati istu radnju. Također, povećavaju produktivnost i učinkovitost, smanjuju rizik od ozljeda i smanjuju troškove proizvodnje.

Postoje dvije glavne vrste robotskih manipulatora: serijski i paralelni [1]. Glavna razlika između serijskih i paralelnih manipulatora je ta što serijski manipulator ima svoje zglobove povezane jedne na drugu (kao što možemo vidjeti na slici 2.1.), dok je kod paralelnog manipulatora svaki zglob spojen na bazu robota i na krajnji aktuator (eng. End effector). Najpoznatiji paralelni robot je Delta robot.



*Slika 2.1. Primjer robotskog manipulatora [9]*

## **2.1. Paralelni manipulatori**

Paralelni manipulator dizajniran je tako da ima jednostavne, kratke zglobove što mu daje krutost protiv neželjenih pokreta koje može izazvati okolina, u usporedbi sa serijskim manipulatorom. Pogreške u pozicioniranju jednog zgloba u prosijeku se izračunavaju s ostalima, umjesto da budu kumulativne. Svaki se zglob i dalje mora kretati unutar vlastitog stupnja slobode, kao kod serijskog manipulatora, no mogućnost kretanja zglobova izvan svoje osi ograničena je drugim zglobovima. Upravo ta ovisnost zglobova jedno o drugom čini paralelni manipulator krutim, za razliku od serijskog koji postaje sve manje krut s više zglobova.

Velika prednost paralelnog manipulatora [2] je da može biti centralno postavljen na jednu temeljnu platformu, čime smanjujemo masu zglobova i aktuatora za istu snagu manipulatora. Ova centralizacija mase aktuatora također smanjuje ukupni moment inercije robota, što može biti prednost za pokretne ili hodajuće robote. Budući da je njihova brzina djelovanja često ograničena samom krutošću manipulatora, a ne čistom snagom, mogu djelovati brzo u usporedbi sa serijskim manipulatorima.

Slika 2.1.1. prikazuje primjer Stewartove platforme. Ona je također poznata vrsta paralelnog manipulatora koja ima šest prizmatičnih hidrauličkih ili električnih aktuatora, pričvršćenih u parovima na 3 mjesta baze. Uređaji postavljeni na gornjoj ploči mogu se pomicati sa šest stupnjeva slobode: tri linearna gibanja  $x$ ,  $y$ ,  $z$  (bočno, uzdužno i okomito) i tri rotacije (nagib, kotrljanje i skretanje). Stewartove platforme pronalaze najveću primjenu u simulatorima

letenja, alatnim strojevima, simulaciji potresa, pozicioniranju satelitskih i drugih antena koje zahtijevaju veliku preciznost, Hexapod-teleskopu i robotici.



*Slika 2.1.1. Primjer Stewartove platforme [10]*

## **2.2. Usporedba paralelnih i serijskih manipulatora**

Većinska primjena robota iziskuje čvrstoću i stabilnost njihovih dijelova. Serijski manipulatori to mogu postići korištenjem visokokvalitetnih rotacijskih zglobova koji dopuštaju kretanje samo u jednoj osi. Kretanje koje zahtijeva nekoliko osi za postizanje željene radnje treba imati više takvih zglobova. Neželjena fleksibilnost ili velika tolerancija u jednom zglobu uzrokuje sličan pomak krajnjeg end effectora. Taj se pomak povećava u odnosu s brojem zglobova i dodatno raste s udaljenošću između njih. Precizni serijski manipulator je kompromis između preciznosti, složenosti, mase (manipulatora i objekta kojim se rukuje) i cijene. S druge strane, kod paralelnih manipulatora, ista ili veća krutost u odnosu na serijski se može postići manjom masom manipulatora. To omogućuje visoku preciznosti veliku brzinu kretanja što je ključno kod simulatora letenja (velike brzine s velikim masama).

Najveći nedostatak paralelnih manipulatora u usporedbi sa serijskim su ograničen radni prostor i osi djelovanja. Kod serijskih, radni prostor je ograničen geometrijskim i mehaničkim granicama dizajna robota. Drugi nedostatak paralelnih manipulatora je njihovo nelinearno

ponašanje, tj. naredba koja je potrebna za dobivanje linearnog ili kružnog kretanja krajnjeg effectora dramatično ovisi o lokaciji u radnom prostoru i ne varira linearno tijekom kretanja.

Donja tablica uspoređuje osnovne karakteristike serijskih i paralelnih manipulatora:

*Tablica 2.2.1. Osnovne karakteristike serijskih i paralelnih manipulatora*

PARAMETAR	SERIJSKI MANIPULATOR	PARALELNI MANIPULATOR
Radni prostor	Velik	Složen i kratak
Pogreška položaja	Akumulira na krajnjem end effectoru	Mala
Krutost	Mala	Velika
Rješavanje dinamike i modeliranja	Lako	Vrlo teško
Inercija	Visoka	Mala
End effectori	Hvatanje	Platforma
Veličina robota za isti radni prostor	Velika	Mala

Iz priložene tablice vidimo da za istu masu robota, paralelni manipulator ima veću stabilnost i nosivost krajnjeg end effectora. Ukoliko nam nije potrebna velika nosiva masa već veličina radnog prostora (npr. pakiranje, zavarivanje) utoliko je bolje koristiti serijski manipulator s kojim, osim osi djelovanja, imamo i smanjen trošak zbog jednostavnijeg upravljanja i izračuna pokreta samih zglobova.

### 3. PRINCIP RADA DELTA ROBOTA

Većina delta robota sastoji se od 3 servo motora montiranih na krutu bazu (njihovo upravljanje će biti opisano u sljedećem poglavlju) koji imaju veliki okretni moment. Na svaku osovinu motora je montirana ruka ili „krak“ koji moraju biti okomiti na rotacijsku os osovine. Sve 3 ruke su spojene na središnju, krajnju platformu. Svaki paralelni krak ima kuglaste zglobove na oba kraja koji omogućuju neograničeno kretanje. Na krajnju/središnju platformu može se dodati end effector i razni drugi aktuatori ovisno o željenoj primjeni robota. Budući da su servo motori montirani na bazu i da su ruke robota duže nego kod serijskog manipulatora, motori moraju imati veliki okretni moment kako bi se suprotstavili velikim silama koje stvara čak i mali teret na središnjoj platformi.

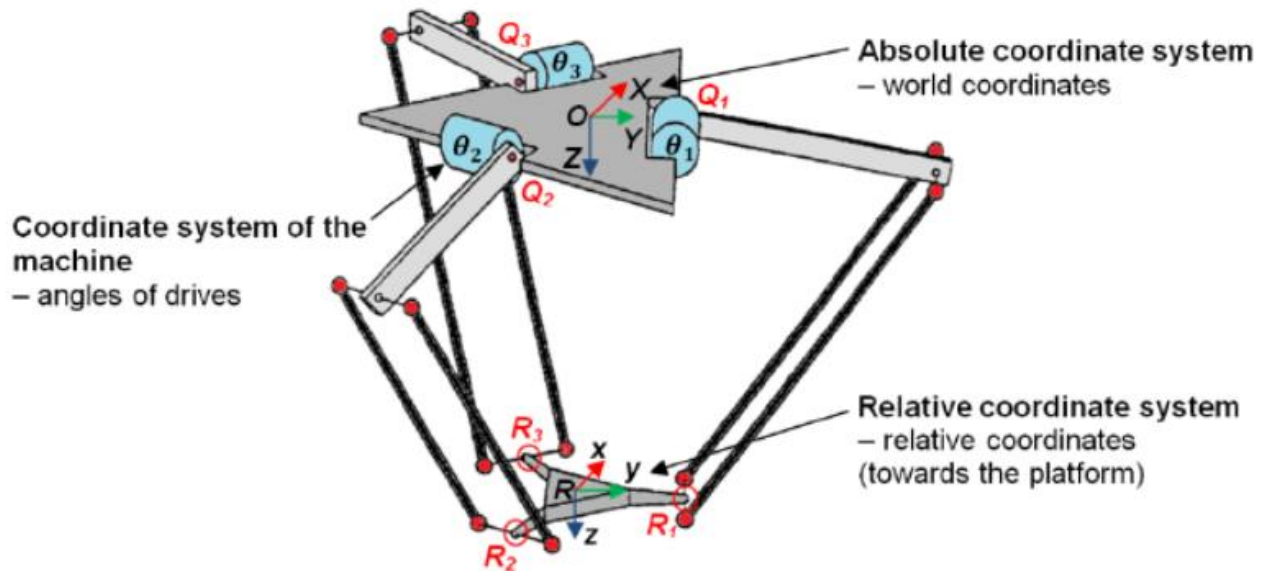
Kinematika je temelj za proučavanje veze između koordinata zglobova i njihove pozicije u prostoru. Kinematika Delta robota sastoji se od inverzne i direktne kinematike. Inverzna kinematika sastoji se od pronalaženja kutova ( $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ) za svaki od 3 kraka (X, Y, Z). Direktna kinematika uključuje određivanje položaja krajnjeg end effectora (X, Y, Z) ako su poznati kutovi ( $\theta_1$ ,  $\theta_2$  i  $\theta_3$ ). Samom kinematikom našeg Delta robota baviti ćemo se u daljnjim poglavljima gdje ćemo napraviti proračune za inverznu i direktnu kinematiku.

U industriji se koriste različite varijacije delta robota u ovisnosti o željenom stupnju slobode:

- 3-DOF – delta roboti kod kojih se end effector može slobodno kretati oko X i Y osi, no ima ograničenu slobodu u Z osi;
- 4-DOF – imaju 4 stupnja slobode. Nadogradnja na 3-DOF robote gdje se Z osi dodaje dodatni stupanj slobode;
- 6-DOF – osim 3 stupnja slobode koji imaju 3-DOF roboti imaju još dodatna 3 stupnja slobode zahvaljujući 3 dodatna rotacijska stupnja na end effectoru.

Delta roboti izvrsni su u radnjama poput brzih „pick-and-place“ operacija, ali su nažalost još uvijek podložni zakonima fizike, posebice Newtonovim zakonom. Neki delta roboti mogu postići ubrzanje do 12-15g. Zbog brzog pokretanja i zaustavljanja, prilikom usporavanja ako end effector nema dovoljnu snagu držanja proizvoda, on može odletjeti. Dodatno, neki proizvodi se ne mogu rukovati i micati usisnom snagom ili zbog svoje težine ili oblika pa se u tim slučajevima koriste hvataljke koje daju dodatnu težinu end effectoru, čime se smanjuje kapacitet samog robota. Na primjer, objektima s hrapavom ili nejednakom površinom obično

se rukuje hvataljkama. Međutim, mehaničke hvataljke povećavaju vrijeme trajanja ciklusa jer se prsti moraju otvarati i zatvarati kako bi podigli i spustili svaki predmet.



Slika 3.1.3-DOF Delta robot [11]

### 3.1. Zašto koristiti Delta robote?

Delta roboti postaju glavna alternativa SCARA tehnologiji za brzo rukovanje proizvodima u postrojenjima. Njima možemo podići teške predmete i značajno povećati brzinu samog procesa. Svjetsko tržište se sve više okreće proizvodnim linijama koje postaju sve brže. Zbog svoje paralelne kinematičke arhitekture, Delta robot ima vrlo veliku mehaničku prednost u performansama i fleksibilnosti u usporedbi s tradicionalnijim robotima. U većini slučajeva koristimo ga ako imamo stalnu brzinu pokretne trake po kojoj se miču stotine dijelova u minuti. Postoje i drugi roboti koji mogu raditi na takvim velikim brzinama, ali ne mogu pratiti brzinu Delta robota bez pregrijavanja ili prekoračenja svojih mehaničkih mogućnosti.



## 4. IZRADA TIJELA DELTA ROBOTA

Delta roboti se većinom sastoje od 3 servo motora montiranih na okruglu bazu. Dužina krakova svakog pojedinog servo motora ovisna je o radijusu baze te o željenom doseg samog robota. Što je baza veća to i krakovi mogu biti duži, što znači da će robot imati veći doseg, no to povlači potrebu za većim i snažnijim motorima koji mogu izdržati takva opterećenja i naprezanja. Kako je cilj ovog diplomskog rada samo prikazati princip izrade, kalkulacije pozicija i rada Delta robota, minimizirat ćemo njegove dimenzije što će i smanjiti sveukupan trošak izrade i nabavke svih komponenti. U sljedećim poglavljima ćemo opisati sve komponente Delta robota te koju ulogu imaju u njegovom radu.

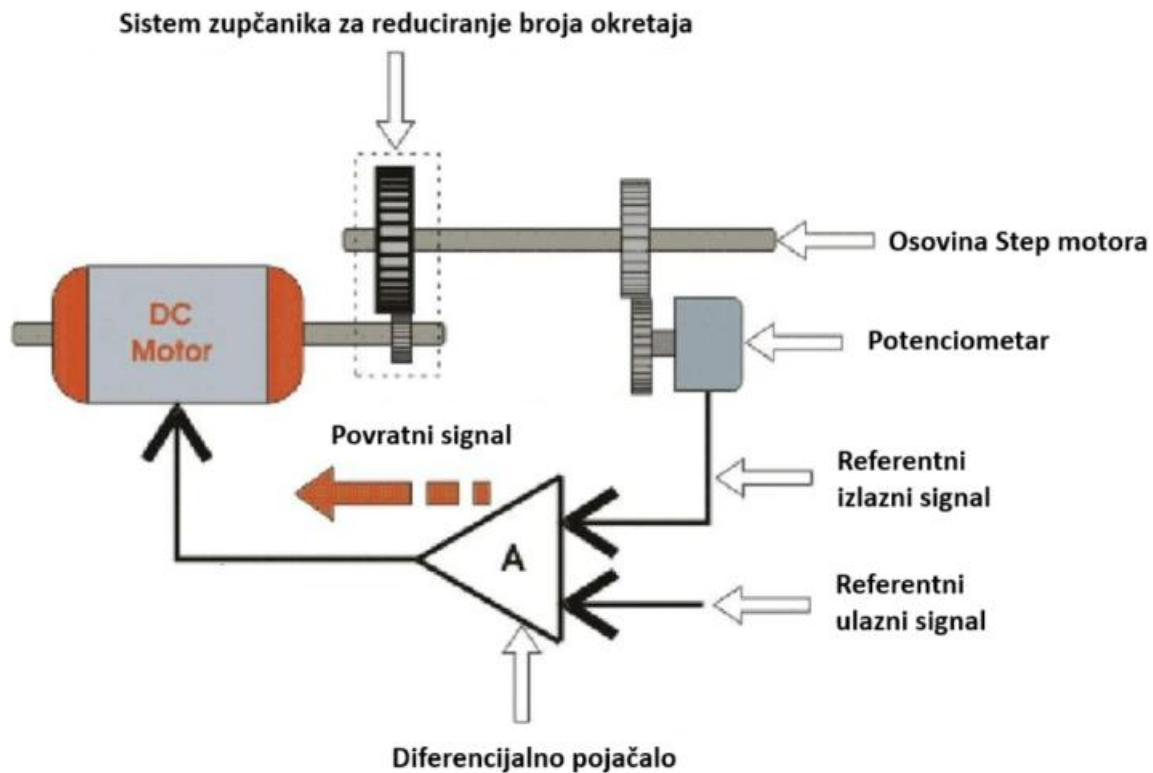
### 4.1. Servo motori

Servo motori su među najučinkovitijim i najkorištenijim vrstama motora u industriji. Oni u osnovi omogućuju preciznu kontrolu linearnog ili kutnog položaja objekta, brzine i ubrzanja, što ih čini idealnim za mnoge poslove koje drugi motori ne mogu obavljati (fina automatizacija) ili nisu pogodni u takvim zadacima zbog složenijeg sustava upravljanja. No, koliko god izgledali jednostavno, servo motori se sastoje od nekoliko ključnih komponenti koje iziskuju sinkroniziran rad da bi postigli preciznost zbog koje su poznati.

#### 4.1.1. Princip rada Servo motora

Servo motor je zatvoren sustav koji se sastoji od motora (AC ili DC), sustava zupčanika, senzora položaja (najčešće potenciometra) i upravljačkog djela. Osovina motora je povezana na sustav zupčanika koji smanjuje brzinu motora, povećava moment i daje povratnu informaciju potenciometru. Potenciometar pretvara fizičku rotaciju u električni signal (razliku otpora) i šalje tu informaciju upravljačkom krugu. On mjeri tu razliku otpora i uspoređuje ju sa željenim položajem te zatim finom kontrolom samog motora postiže željeni položaj. Na slici 4.1.1.1. [3] vidimo blokovski prikaz Servo motora. Potenciometar je preko prijenosnog omjera zupčanika spojen na glavnu osovinu Step motora. Kako se on okreće tako se mijenja otpor potenciometra te diferencijalno pojačalo mjeri tu razliku. Na temelju dobivene pozicije od potenciometra željene/referentne pozicije koju dobijemo od vanjskog upravljačkog uređaja,

diferencijalno pojačalo generira signal razlike koji šalje na DC motor koji se pomiče na željeni položaj. Diferencijalno pojačalo se u slučaju Servo motora ne sastoji samo od jednog integriranog kruga kako je prikazano na donjoj blok shemi, već od cijelog upravljačkog sklopa koji dekodira ulazni signal željene pozicije te na temelju toga djeluje na DC motor. U poglavlju 4.1.2. i 4.1.3. objasnit ćemo metodu upravljanja te vrstu signala koji je potreban za pozicioniranje Servo motora.

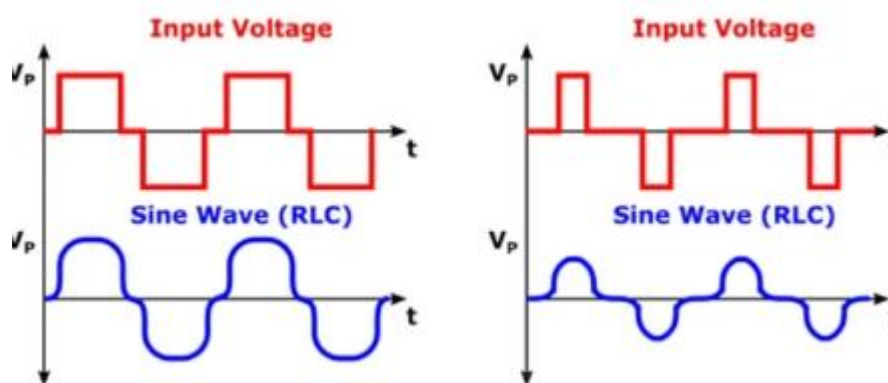


Slika 4.1.1.1-Blok shema Servo motora [12]

#### 4.1.2. Teorija PWM upravljanja

Pulsno-širinska modulacija ili PWM (*eng. Pulse width modulation*) je metoda gdje se manipulira signalom tako da su mu amplituda i frekvencija konstante, ali se mijenja radni ciklus signala čime signal poprima svoj karakterističan pravokutni oblik. Time možemo kontrolirati prosječnu vrijednost napona i struje koja se dovodi do trošila. PWM radi tako da se napajanje prebacuje između 0 i 100% brzinom većom od promijene opterećenja trošila. U suštini, što je sklopka duže uključena, veća je ukupna snaga dovedena do opterećenja. Cilj PWM-a je kontrolirati trošilo, međutim, frekvencija mora biti pažljivo odabrana da bi se kontrola vršila

glatko. Frekvencija PWM upravljanja može uvelike varirati ovisno o trošilo kojega želimo upravljati. Na primjer, za upravljanje jačine električnog štednjaka potreban je PWM signal od 100-120Hz dok za upravljanje motora frekvencija PWM modulacije se povećava na nekoliko desetaka kiloherca (kHz). Audio pojačala koriste PWM signal i do nekoliko megahertza (MHz). Glavna prednost PWM je da je gubitak snage u sklopnim uređajima vrlo nizak. Kada je sklopka isključena, praktički nema struje, a kada je uključena i snaga se prenosi na opterećenje, gotovo da nema pada napona na sklopki. Gubitak snage, kao proizvod napona i struje, je stoga u oba slučaja blizu nule. PWM također dobro radi s digitalnim kontrolama, koje, zbog svoje prirode uključivanja/isključivanja, mogu lako postaviti potrebni radni ciklus. PWM se također koristi u određenim komunikacijskim sustavima gdje se njegov radni ciklus koristi za prijenos informacija preko komunikacijskog kanala. U našem slučaju, koristiti ćemo PWM signal za upravljanje Servo motora Delta robota, ali ne kao metodu za upravljanje snagom samog Servo motora već za prijenos informacija o željenom položaju motora.

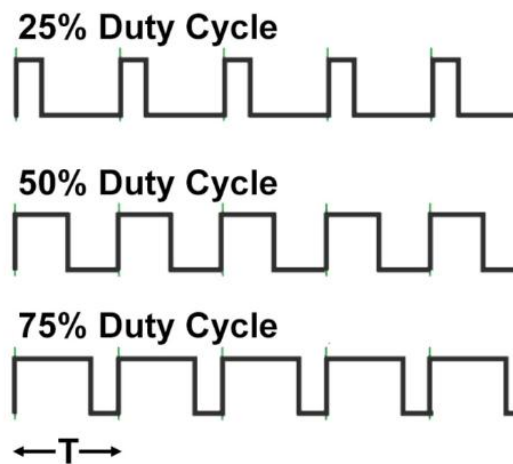


*Slika 4.1.2.1-Prikaz ovisnosti prosječne snage predane trošilu o frekvenciji PWM signala [13]*

Na slici 4.1.2.1. vidimo ovisnost prosječne snage predane trošilu o frekvenciji (radnom ciklusu) PWM signala. Što je impuls PWM signala duži, tj. ima veću periodu, to je sinusoida pravilnijeg oblika i time je i ukupna snaga na trošilu veća.

Radni ciklus PWM signala opisuje vrijeme uključenosti signala u svakom intervalu. Mali radni ciklus odgovara maloj prenesenoj snazi prema trošilu, jer je napajanje isključeno većinu vremena. Kada je digitalni signal pola vremena uključen, a drugu polovicu isključen, digitalni signal ima radni ciklus od 50% i nalikuje "kvadratnom" valu. Kada digitalni signal provede više vremena u uključenom nego u isključenom stanju, ima radni ciklus od  $>50\%$ . Kada

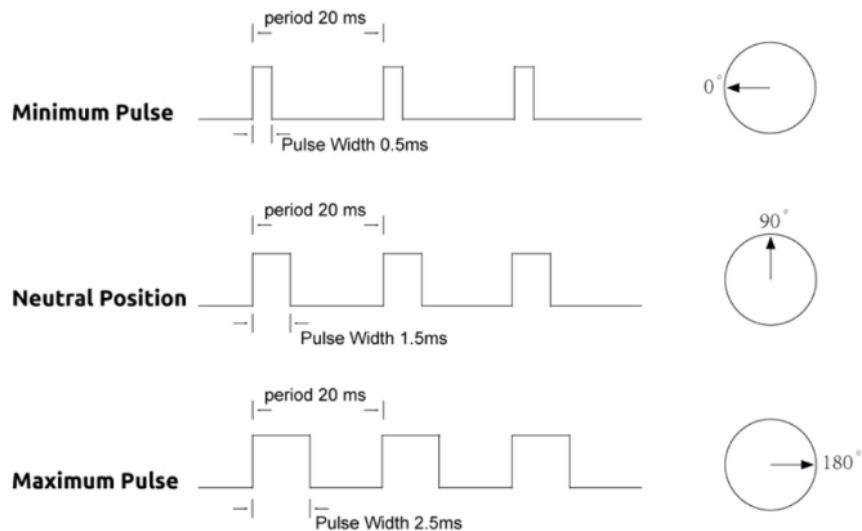
digitalni signal provede više vremena u stanju isključenosti nego u stanju uključenosti, njegov radni ciklus je <50%.



Slika 4.1.2.2-Različiti radni ciklusi PWM signala [14]

#### 4.1.3. Upravljanje Servo motorima

Servo motori imaju 3 vodiča: GND (minus pol napajanja), VCC (plus pol napajanja) i kontrolna žica (koristi se za zadavanje željenog položaja). Napon napajanja Servo motora je direktno vezan s veličinom i snagom samog motora. Veći motori će zahtijevati veći napon napajanja no u našem slučaju ćemo koristiti motore s +5V DC naponom napajanja. Servo motorima se upravlja preko PWM impulsa na kontrolnoj žici. Radni ciklus tog impulsa određuje poziciju motora. To je čisto digitalni signal koji služi za upravljanje, samo napajanje motora se vrši preko VCC vodiča. Postoji minimalni puls, maksimalni puls i period ponavljanja. Većina servo motora se može okrenuti od 0° do maksimalno 180° (90° u svakom smjeru). Neutralni položaj motora definiran je kao položaj u kojem motor ima istu količinu potencijalne rotacije u smjeru kazaljke na satu ili u suprotnom smjeru. PWM koji se šalje motoru određuje položaj osovine, a na temelju trajanja impulsa, rotor će se okrenuti u željeni položaj. Servo motor očekuje puls svakih 20 milisekundi (ms) [4], a duljina impulsa određuje koliko se motor okreće. Primjer sa slike 4.1.3.1: puls od 1,5 ms govori motoru da se okrene u položaj od 90°. Puls kraći od 1,5 ms pomiče ga u smjeru suprotnom od kazaljke na satu prema položaju 0°, a dulji od 1,5 ms okretat će motor u smjeru kazaljke na satu prema položaju 180°.



*Slika 4.1.3.1-Primjer rotacije Servo motora u ovisnosti o duljini PWM impulsa [15]*

Kada se Servo motoru izdaje naredba da se pomakne u određeni položaj, on će se tamo pomaknuti i zadržati isti položaj. Ako neka vanjska sila djeluje na osovinu motora, on će se opirati toj promjeni. Maksimalna jačina sile koju Servo motor može izdržati naziva se okretni moment Servo motora. On će zadržati tu poziciju jedino ako se impuls ponavlja. Ključno je razumjeti tehnički način rada Servo motora kako bi se mogao odabrati pravi tip motora za određenu radnju.

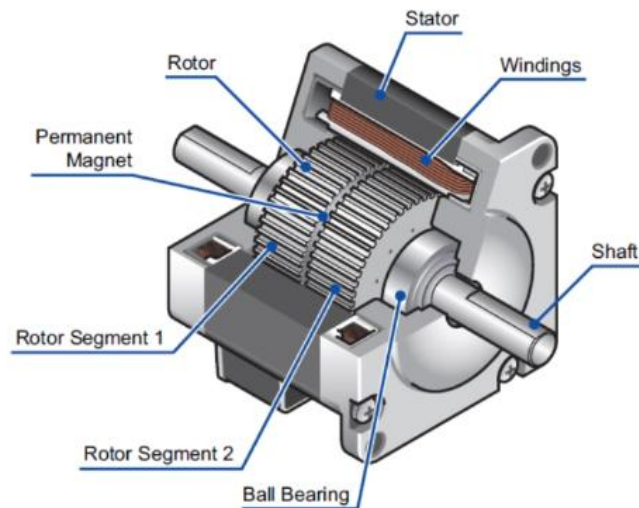
#### 4.1.4. Podjela i primjena Servo motora

Glavna podjela Servo motora vrši se na temelju napona napajanja, tj. postoje DC i AC Servo motori. AC motori bolje podnose struje udare pa se češće koriste u industrijskim strojevima i automatizaciji. DC motori ne podnose velike struje udare i općenito su bolji za manje, finije primjene. Uglavnom, istosmjerni su motori jeftiniji od izmjeničnih. Neki od njih napravljeni su za kontinuiranu rotaciju (puni krug, 360°), što olakšava pokretanje. Imaju dva kuglična ležaja na izlaznoj osovinu za smanjenje trenja i lakši dolazak do potencijometra koji mijenja točku mirovanja.

Servo motori se koriste u radio-upravljanim zrakoplovima za postavljanje kontrolnih površina poput dizala, kormila, hodajućih robota ili hvataljki. Manji su, imaju ugrađeni upravljački krug i imaju dobru snagu za svoju veličinu. Također, koriste se u in-line proizvodnji, gdje je

potrebno puno ponavljanja, ali i precizan rad. Još jedna popularna primjena je u automatskim vratima gdje kontroliraju otvaranje i zatvaranje samih vrata koja se najčešće koriste u supermarketima i bolnicama.

Još jedna popularna vrsta motora koja se koristi u industrijskoj automatizaciji je koračni motor. Glavna razlika je ta da Servo motori rade sa zatvorenom petljom, što znači da imaju povratni podatak o svojem položaju, dok koračni motori rade s otvorenom petljom. U otvorenoj petlji motor nema povratnu informaciju o svom položaju bez nekog vanjskog senzora. Koračni motori obično rade s velikim okretnim momentom i malim brzinama i prikladni su za kretanje na kratkim udaljenostima. Nasuprot tome, Servo motori obično rade pri velikim brzinama i općenito proizvode veći moment pri određenoj brzini od koračnih motora.



*Slika 4.1.4.1-Presijek koračnog motora [16]*

#### 4.1.5. Odabir Servo motora za pokretanje Delta robota

Kako veličina i snaga Delta robota direktno utječe na veličinu i snagu potrebnih Servo motora za pomicanje krakova trebamo odabrati kakve motore želimo koristiti. U praksi, radnja koju bismo željeli obavljati Delta robotom prvo određuje njegovu veličinu, a veličina samog robota utječe na odabir motora. U našem slučaju, kako će finalni Delta robot biti samo za prikazivanje mogućnosti takvog robota, nisu potrebni snažni Servo motori. Kontrola Delta robota vršiti će se preko Arduino mikro kontrolera (detaljnije u poglavlju 4.2.) koji radi na naponu napajanja od 5V, što znači da bi zbog jednostavnosti, poželjno bilo odabrati Servo motore s istim naponom napajanja. Osim tog zahtjeva, motori trebaju imati preciznost i finu kontrolu položaja kako bi se tolerancija end effectora svela na minimum.

Uzimajući u obzir sve zahtjeve, odabrali smo TowerPro MG92B Servo motor [5]. Na naponu napajanja od 5V ima brzinu rada 0.13sec/60° i zakretni moment od 3.1kg/cm. Ostale kompletne karakteristike MG92B Servo motora nalaze se u donjoj tablici:

*Tablica 4.1.5.1. Karakteristike TowerPro MG92B Servo motora*

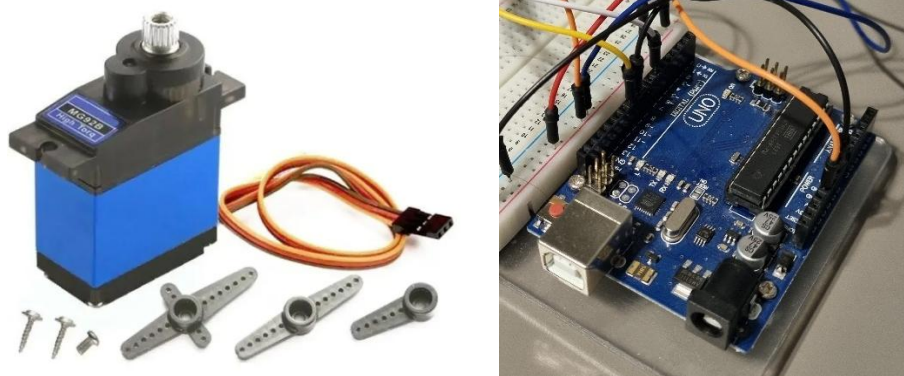
PARAMETAR	VRIJEDNOST
Težina	13,8 g
Dimenzije	22,8x12x31 mm
Zakretni moment	3,1 kg/cm (5,0 v); 3,5 kg/cm (6,0 v)
Radna brzina	0,13 s/60 stupnjeva (5,0 v); 0,08 s/60 stupnjeva (6,0 v)
Radni napon	5,0 ~ 6,6 v
Tip zupčanika	Metalni
Ležaj	Dvostruki kuglični ležaj
Kućište	Legura
Radna temperatura	0 - 55 °C
Utikač	JR (Odgovara JR i Futaba)
Širina mrtvog pojasa	1us

## 4.2. Arduino mikrokontroler

Pri svakom pomicanju end effectora Delta robota iz jedne točke koordinatnog sustava u drugu potrebno je napraviti proračun inverzne kinematike kako bi se odredili točni kutovi pomaka Servo motora. Matematički proračun nije procesno intenzivan, no to trebati uzeti u obzir prilikom odabira mikrokontrolera ili mikroracunala koji će vršiti te proračune i na temelju njih kontrolirati motore. Jedan od najpoznatijih mikroracunala za takve projekte je Raspberry Pi, no on je snažniji nego što bi trebao biti za naše potrebe te bi ostao neiskorišten. Druga opcija su mikrokontroleri. Oni su mali integrirani krugovi koji imaju sve što je potrebno da budu računala: programsku memoriju, procesor i ulazno/izlazne jedinice. U našem slučaju koristiti ćemo Arduino Uno Rev3 mikrokontrolersku ploču koja ima Atmega328p integriani krug koji postiže frekvenciju rada do 16 MHz. Ta brzina izvođenja radnji je dovoljno brza za računanje

inverzne kinematike našeg Delta robota, a istovremeno je do 10 puta jeftinija od prije spomenutog Raspberry Pi mikroračunala.

Arduino Uno [6] opremljen je skupovima digitalnih i analognih ulaza/ izlazna (I/O) pinova koji se mogu povezati s raznim pločama za proširenje funkcionalnosti i ostalim sklopovima. Mikrokontroler ima 14 digitalnih I/O pinova (od kojih 6 ima mogućnost PWM izlaza), 6 analognih I/O pinova te podržava napon napajanja od 7-20V. Programiranje se vrši preko Arduino IDE programskog paketa na podržanom računalu.



*Slika 4.2.1-Prikaz korištenog Arduino Uno mikrokontrolera (desno)  
i TowerPro MG92B Step motora (lijevo)*

#### 4.2.1. Tehničke specifikacije Arduino Uno mikrokontrolera

Sve varijacije Arduino Uno razvojne pločice od revizije 1 do revizije 3 koriste neku verziju Atmega328p mikročipa (novije verzije imaju manje fizičke dimenzije istog čipa) te isti „popis“ komponenti. Jedina razlika je fizička veličina pojedinih komponenti te njihov poredak na samoj pločici, ali raspodjela ulazno/izlaznih pinova je identična. Detaljne specifikacije ATmega mikrokontrolera:

- IC: Microchip ATmega328P (8-bitna AVR jezgra),
- radni takt od 16 MHz, iako je IC sposoban za maksimalno 20 MHz na 5 volti,
- flash memorija: 32 KB, od čega 0,5 KB koristi bootloader,
- SRAM: 2KB,
- EEPROM: 1KB,



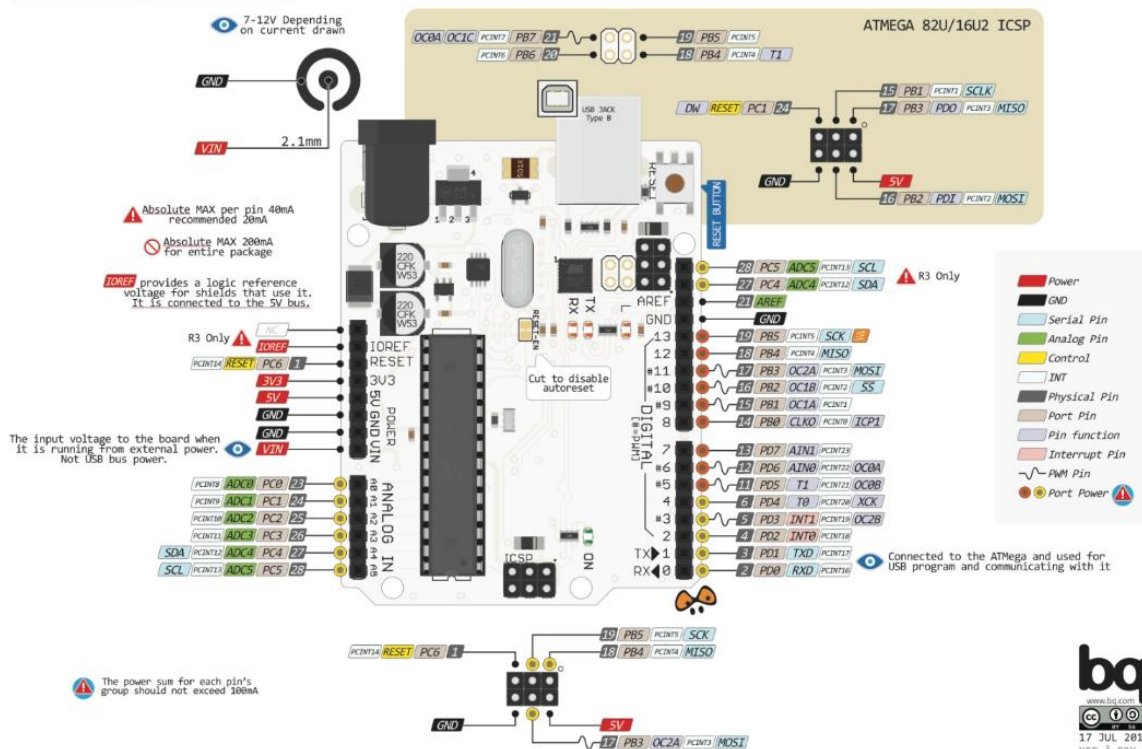
- SPI komunikacijski čip: 1,
- I<sup>2</sup>C komunikacijski čip: 1,
- napon napajanja: 5V.

Tablica 4.2.1.1. Specifikacije Arduino Uno Rev3 razvojne pločice

PARAMETAR	VRIJEDNOST
Digitalni I/O pinovi	14
PWM pinovi	6 (pinovi 3, 5, 6, 9, 10 i 11)
Analogni pinovi	6
Istosmjerna struja po I/O pinu	20 mA
Istosmjerna struja 3,3 V pina	50 mA
Veličina	68,6 mm x 53,4 mm
Težina	25 g
Kućište	Legura
Radna temperatura	0 - 55 °C
Utikač	JR (Odgovara JR i Futaba)
Širina mrtvog pojasa	1us

Pločica se može napajati direktno preko USB kabela s računala. No ta opcija, pri standardnom naponu napajanja USB-a od 5V može dati maksimalnu struju od 500 mA što bi u pojedinim slučajevima gdje nam treba dodatna snaga za vanjske periferije (Servo motore) bilo malo. Zbog toga, pločica ima dvije dodatne mogućnosti za izbor napajanja: 5,5 mm DC konektor ili VIN pin. U oba slučaja, podržani opseg ulaznog napona je od 6 – 20V. Struja potrebna za pokretanje Servo motora ovisi o njegovom opterećenju, tj. što je veća sila koju motor treba prevladati na svojoj osovini, to će mu trebati veća struja. U našem Delta robotu, najveće opterećenje koje će Servo motori trebati prevladati je sila trenja svih zglobova pri pomicanju end effectora. Dakle, ne znamo točno kolika je potrebna snaga svih triju Servo motora jer je jako ovisna o fizičkoj izvedbi robota pa će tu vrijednost trebati naknadno izmjeriti.

# UNO PINOUT



Slika 4.2.1.1-Specifikacije pinova korištene Arduino Uno razvojne pločice [17]

Neke od bitnih dodatnih funkcija razvojne pločice:

- **LED:** postoji ugrađena LED dioda koju pokreće digitalni pin 13. Kada je pin visoke vrijednosti, LED je uključen, kada je pin niska, isključen je;
- **5V:** ovaj pin dopušta korištenje napon napajanja 5V iz regulatora na ploči. Sam regulator na pločici se napaja iz DC utičnice (7 - 20V), USB konektora (5V) ili VIN pina (7-20V). Spajanje vanjskog napajanja izravno preko pinova od 5V ili 3,3V u cilju napajanja Arduina zaobilazi regulator i može oštetiti ploču;
- **3.3V:** napajanje od 3,3V koje stvara ugrađeni regulator, maksimalna potrošnja struje je 50 mA;
- **IOREF:** ovaj pin na Arduino ploči daje referentni napon s kojim radi mikrokontroler. Ispravno konfigurirana vanjska, ekspanzijska pločica može očitati napon IOREF pina i odabrati odgovarajući izvor napajanja ili omogućiti regulatorima napona na izlazima da rade s 5V ili 3,3V;
- **RESET:** obično se koristi za dodavanje gumba za resetiranje mikrokontrolera.

### 4.3. 3D printanje i fizičke dimenzije robota

Za pravila rad Delta robota, osim krakova i end effectora, potrebni su zglobovi koji će povezati sve dijelove robota i osigurati mu nesmetano gibanje kroz cijelo svoje područje rada. Takvi zglobovi se najčešće moraju moći okretati oko svih svojih osi kako bi postigli ravnomjerno pomicanje end effectora. Kako Delta roboti nisu dosegli popularnost u kućanstvima kao što su to uspjeli 3D printeri u zadnjih 10 godina, cijena pojedinih elemenata je još uvijek visoka. Velika adaptacija 3D printera pomogla im je da dosegnu rekordno niske cijene sastavljenog printera, ali time i rezervnih dijelova. Zglobovi i krakovi kakvi su potrebni za izradu Delta robota se još uvijek primarno koriste u industrijskim okruženjima pa je i njihova cijena time veća. Zbog navedenih razloga i zbog jednostavnosti samog postupka, za izradu baze, end effectora i krakova robota koristit ćemo 3D printer. Kako je glavna tema ovog diplomskog rada „Dizajn, proračun i izrada paralelnog robotskog manipulatora“ nećemo se previše usredotočiti na rad 3D printer te njegove mane i prednosti. No zbog odabira vrste materijala koje će utjecati na rad samog robota, opisat ćemo ukratko specifikacije korištenog 3D printera, korištenu vrstu plastike za print te postupak dobivanja pojedinih dijelova.

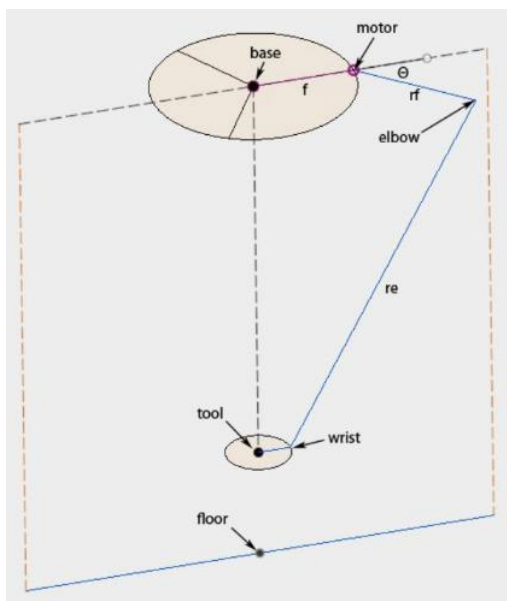
Za naše potrebe koristit ćemo Creality Ender 3 3D printer. To je popularan model poznat po pouzdanosti, pristupačnoj cijeni i jednostavnosti korištenja. U sljedećoj tablici imamo sažetak najbitnijih specifikacija tog printera:

*Tablica 4.3.1. Specifikacije korištenog Creality Ender 3 3D printera*

PARAMETAR	VRIJEDNOST
Volumen printanja	220x220x250 mm
Tolerancije printanja	±0.1 mm
Brzina printanja (Maksimalna)	60 mm/s (200 mm/s)
Vrsta filamenta	PLA, AABS, PETG, TPU 1.75 mm
Maksimalna temperatura ispisa	255 °C
Maksimalna temperatura grijane podloge	110 °C
Snaga napajanja	270W

#### 4.3.1. Izračun fizičke dimenzije dijelova robota

Delta robot ima 4 glavna dijela: bazu, end effector, kraći krak i dulji krak. Duljine i promjeri tih dijelova te njih međusobni odnos određuju karakteristike koje će imati naš robot. Ako imamo neke posebne zahtjeve što se tiče veličine radnog prostora, težine ili vrstu end effectora to trebamo sada uzeti u obzir. Kako mi nemamo neke posebne zahtjeve jer želimo samo prikazati primjer rada takvog robota, duljine ćemo birati proizvoljno. Postoje nekoliko online alata za dobivanje radnog prostora Delta robota koji preko matematičkih formula [7], inverzne i direktne kinematike određuju rezultatni radni prostor Delta robota s unesenim dimenzijama. To nam može pomoći dobiti ideju što očekivati od gotovog robota i podesiti duljine krakova i baze ako želimo veći radni prostor. Moramo uzeti i u obzir volumen printanja našeg 3D printera i da niti jedan dio Delta robota ne bude veći od 220x220x250 mm. Na slikama 4.3.1.1 i 4.3.1.2 možemo vidjeti online program koji ćemo koristiti za proračune radnog prostora Delta robota:



Slika 4.3.1.1-Skicirani Delta robot online programa za izračun radnog prostora [18]

<b>Polumjer baze (f)</b>	<input type="text" value="31"/>	mm	Udaljenost od središta baze stroja do središta svake osovine motora.
<b>Duljina bicepsa (rf)</b>	<input type="text" value="50"/>	mm	Udaljenost od osovine motora do koljena
<b>Duljina podlaktice (re)</b>	<input type="text" value="90"/>	mm	Razmak od lakta do zgloba
<b>Polumjer krajnjeg efektor (e)</b>	<input type="text" value="15"/>	mm	Udaljenost od zapešća do alata
<b>Udaljenost od baze do poda (b)</b>	<input type="text" value="130"/>	mm	Udaljenost od poda do baze
<b>Koraci po okretaju</b>	<input type="text" value="3200"/>		Preciznost motora. Steperi od 1,8 stupnjeva imaju 200 koraka po krugu. Na 1/16 mikrokoraka to je 3200 koraka po okretaju.
<b>Pravokutna kuboidna ovojnica</b>	X=-37,553 do 37,553 mm Y=-37,553 do 37,553 mm Z=-122,419 do -47,314 mm		Kolikju kutiju može doseći krajnji efektor? (krajnji efektor se zapravo može pomaknuti više od ovoga)
<b>ograničenja kuta motora</b>	theta 1=-43,49 do 95,22 theta 2=-53,21 do 95,25 theta 3=-53,21 do 95,25		Kako se motori moraju okretati da bi se kretali kroz pravokutni kvadar?
<b>Centar</b>	(0,0,-84.866)		Gdje je sredina ovojnice u odnosu na bazu (0,0,0)?
<b>Dom</b>	(0,0,-71.532)		Gdje je alat kada su ruke paralelne s podom?
<b>Rezolucija</b>	+/-0,086 mm		Koliko pokreti mogu biti precizni?

*Slika 4.3.1.2-Označene kote online programa za izračun radnog prostora Delta robota [19]*

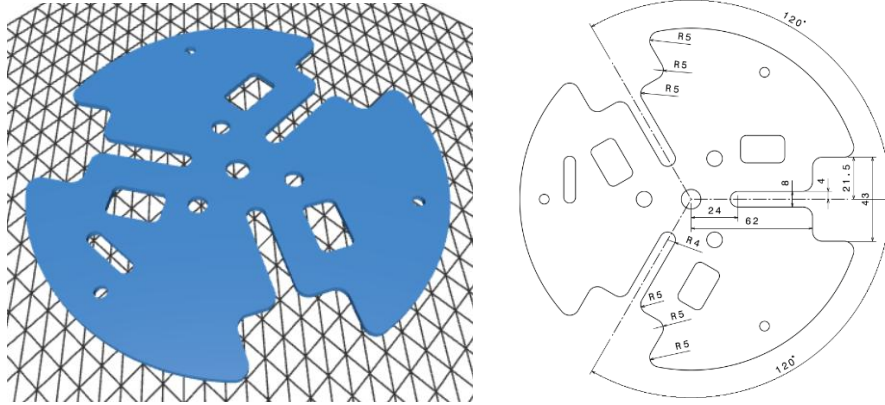
Na slici 4.3.1.2. vidimo naše konačne odabrane vrijednosti veličine Delta robota. Duljine krakova i dimenzije baze su tako odabrane da se mogu printati na 3D printeru, a da dobijemo što veći radni prostor end effectora. Vrijednost „Koraci po okretaju“ se odnosi na korak koračnog motora jer je ovaj kalkulator napravljen s idejom da se Delta robot realizira preko koračnih motora pa ju u ovom slučaju možemo ignorirati. Ne utječe na efektivni radni prostor robota i koristi se primarno za izračun zadnje vrijednosti programa „Rezolucija“ koju također možemo ignorirati jer rezolucija u našem slučaju ovisi o rezoluciji korištenih Servo motora i izmjerit ćemo je naknadno. Izračunane koordinate, označene kao „Pravokutni kuboid“ su radni prostor Delta robota. Te vrijednosti ćemo koristiti u našem upravljačkom programu za određivanje kuteva Step motora preko inverzne kinematike.

#### 4.3.2. Ispis i nacrti 3D dijelova Delta robota

Naš konačni Delta robot će se sastojati od nekoliko dijelova. Kako 3D printer može isprintati samo statičke dijelove, svaki krak i svaki pokretni dio se mora razbiti na svoje osnovne dijelove da bi se oni nakon 3D ispisa mogli sastaviti u gotov zglob. Osim osnovnih dijelova robota (zglobovi, end effector) trebamo još ispisati postolje na kojemu će stajati robot, samu bazu robota koja će se sastojati od nekoliko slojeva, mjesto za Arduino razvojnu pločicu, itd... Svaki pojedini dio trebamo ponajprije izmodelirati u 3D programu te poslati taj model na 3D printer. Mi ćemo koristiti Autodesk Fusion 360 programski paket jer ima besplatnu

studentsku/edukacijsku verziju. U naknadnom tekstu ćemo prikazati i opisati svaki pojedini 3D dio Delta robota i njegovu funkciju u konačnom sklopu:

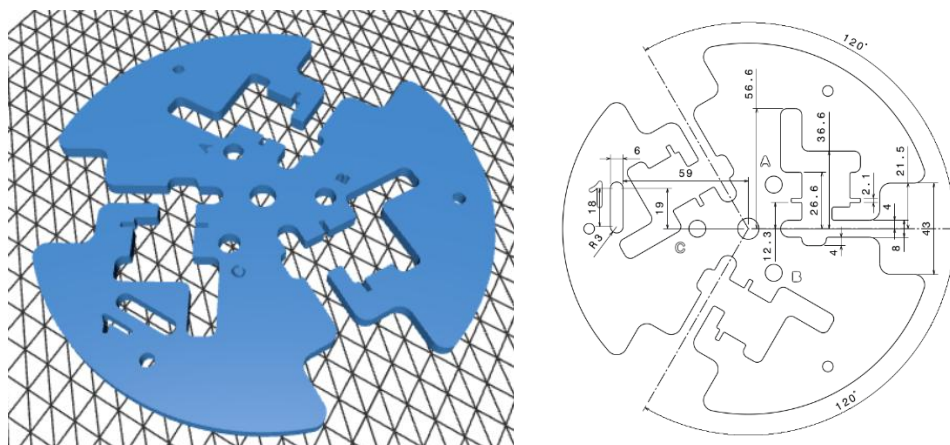
- 1. sloj baze robota



*Slika 4.3.2.1-1.sloj baze robota, 3D model (lijevo) i nacrt (desno)*

Ovaj 1. sloj baze robota nam služi kako bi držao na mjestu Servo motore te osigurano dobar oslonac za ostale slojeve baze koji se pričvršćuju na njega. Vidimo kako je polumjer baze veći od izračunatog u prijašnjem poglavlju gdje je iznosio 31 mm. To je zbog toga što se polumjer računa od sredine baze do glavne osovine Servo motora, a ona će se u ovo slučaju nalaziti na 31 mm od sredine, točno kao po izračunu.

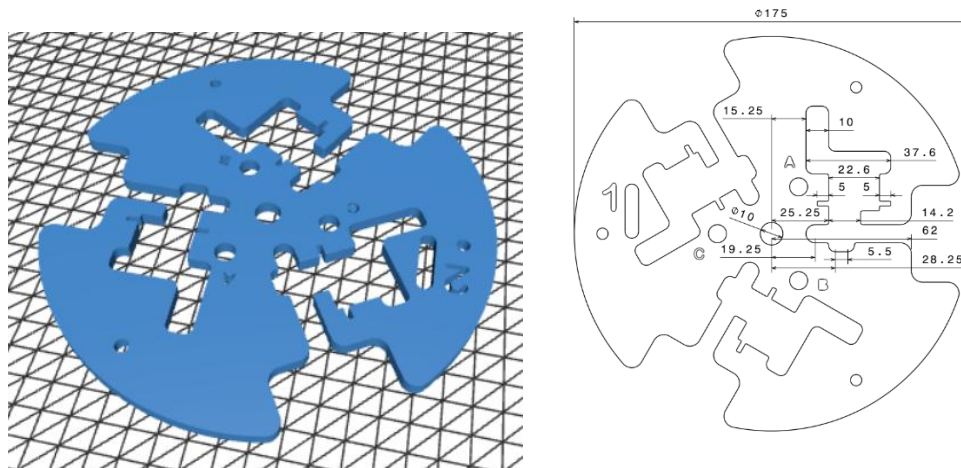
- 2. sloj baze robota



*Slika 4.3.2.2-2.sloj baze robota, 3D model (lijevo) i nacrt (desno)*

Drugi sloj baze robota ima rupe koje su u obliku korištenih TowerPro Servo motora da se mogu montirati bez vijaka. Ovaj sloj se postavlja iznad 1.sloja te ima oznake „A“, „B“ i „C“ za svaki pojedini Servo motor. To će nam kasnije služiti u programskom djelu da znamo koji je motor spojen na koji digitalni izlaz Arduino razvojne pločice.

- 3. sloj baze robota

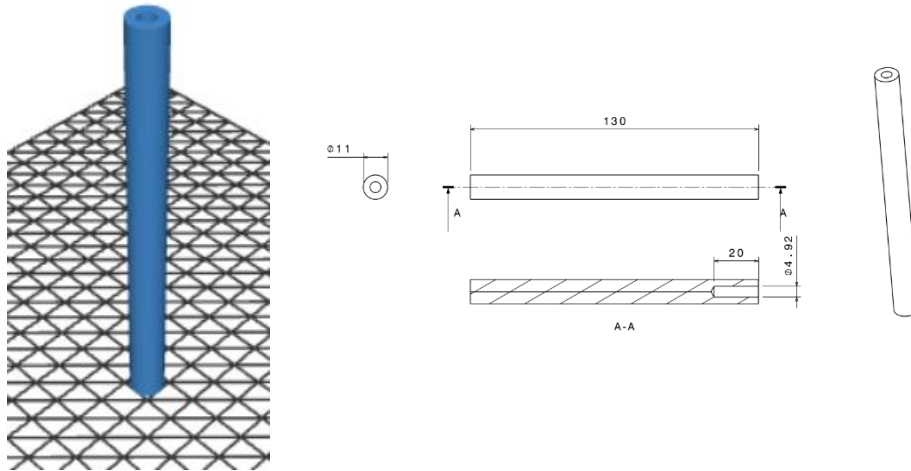


*Slika 4.3.2.3-3.sloj baze robota, 3D model (lijevo) i nacrt (desno)*

Treći je sloj baze robota identičan drugome te služi istoj svrsi. Skupna visina drugog i trećeg sloja baze odgovara debljini korištenih Servo motora. To nam daje stabilnost samog Robota i osigurava da se motori neće pomaknuti pri pomicanju zglobova samog robota.



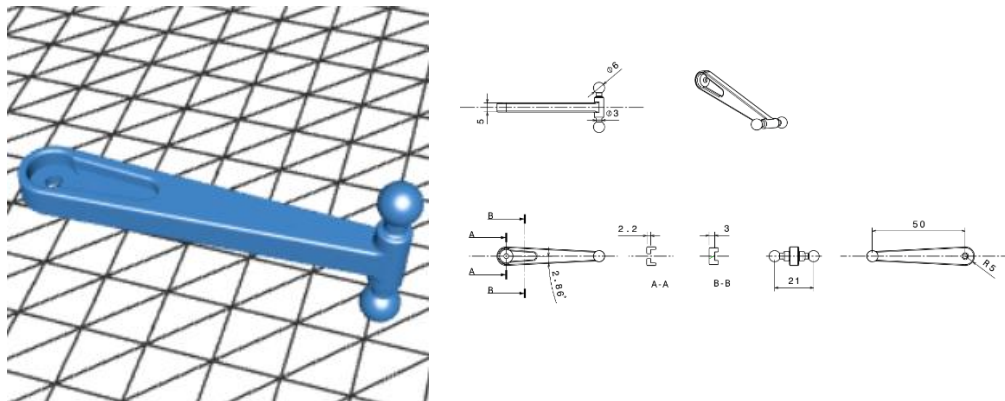
- Postolje baze



*Slika 4.3.2.4-postolje baze robota, 3D model (lijevo) i nacrt (desno)*

Postolje ili nožice robota se pričvršćuju na bazu s vijkom te spajaju sva 3 sloja baze robota. Njihova visina određuje vertikalnu os radnog prostora robota. Dimenzionirani su tako da, s obzirom na duljine krakova robota, dobijemo što veći radni prostor. Potrebna su nam 3 komada.

- Spoj servo motora s krakovima robota

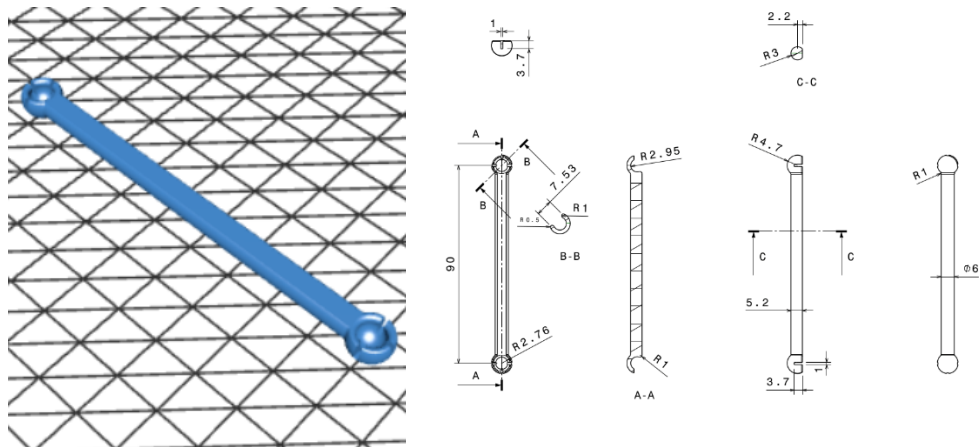


*Slika 4.3.2.5-spoj servo motora sa krakovima Delta robota, 3D model (lijevo) i nacrt (desno)*



Ovo je krucijalan dio robota jer predstavlja spoj između Servo motora i prvog zgloba dužeg dijela kraka robota. Dimenzioniran je prema proračunima u poglavlju 4.3.1. Ovaj se dio pričvršćuje na Servo motore preko vijaka koji se dobiju sa svakim motorom. Ovaj smo dio printali sa 100% ispunom plastike da bismo imali čvrstoću svakog kuglastog zgloba. 3D printer ima veću toleranciju pa će postojati dosta veliko trenje pri pomicanju zglobova 3D printanog Delta robota što trebamo uzeti u obzir. Trebaju nam 3 takva komada koji predstavljaju sva 3 kraka robota.

- Duži krak Delta robota



*Slika 4.3.2.6-duži krak Delta robota,  
3D model (lijevo) i nacrt (desno)*

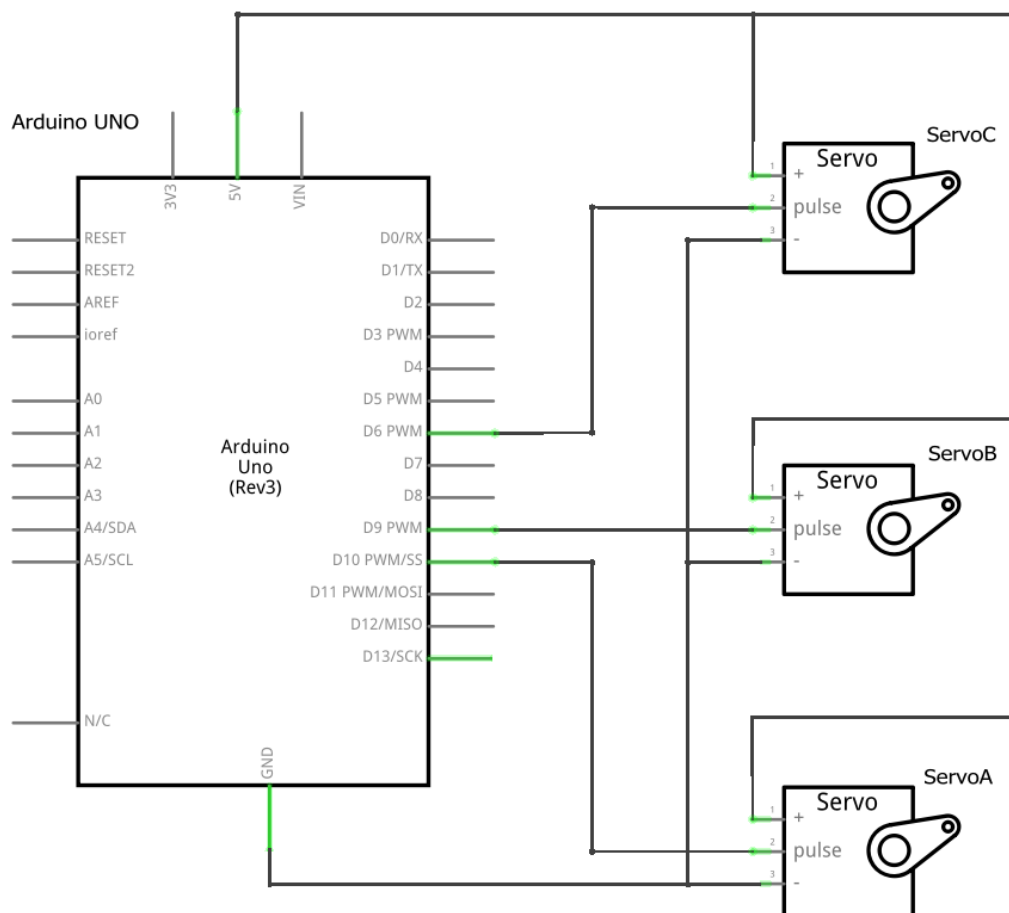
Potrebno nam je šest (dužih krakova) ovakvih komada. Oni spajaju kraći krak koji pomiče Servo motor i end effector robota. Na svakom vrhu ima kuglasti zglob koji omogućuje nesmetano pomicanje end effectora i samog kraka robota na svaku točku radnog prostora.



Baza nam služi za pričvrščivanje mozga, tj. Arduino Uno razvojne pločice koja će upravljati Servo motorima. Baza treba biti dosta povišena da ne bi smetala zglobovima robota kada se pomiču vertikalno.

#### 4.4. Električna shema spajanja Servo motora i Arduino razvojne pločice

Servo motorima se upravlja pomoći PWM signala kako smo i opisali u prijašnjim poglavljima, što znači da nemamo nikakvih dodatnih pasivnih/aktivnih komponenti. Odabrani TowerPro motori koriste napajanje od 5V te zbog svoje dimenzije trebaju i malu struju za rad. To znači da ih možemo napajati izravno iz 5V regulatora Arduina. U nastavku je slika sheme spajanja motora:



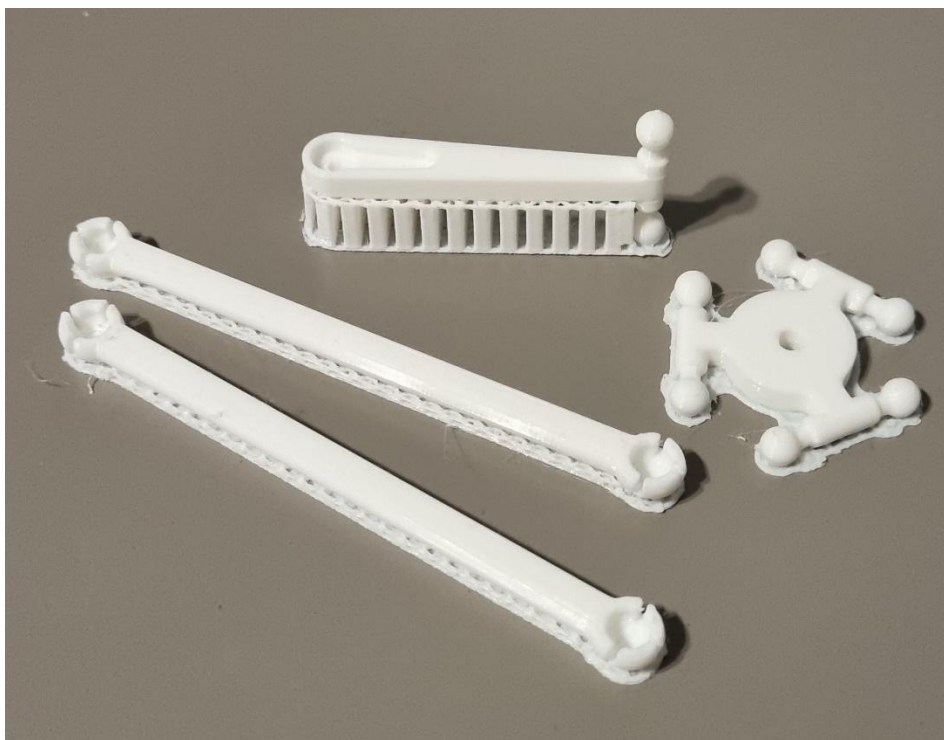
Slika 4.4.1-shema spajanja Arduino Uno razvojne pločice i Servo motora

Konektor svakog Servo motora se sastoji od 3 žice/kontakta koji idu sljedećim redom:

- **SIGNAL (žuta)** – signalna žica preko koje se upravlja Servo motor. Nju spajamo na bilo koji digitalni izlaz Arduina koji ima mogućnost generiranja PWM signala,
- **VCC (crvena)** – pozitivni pol napajanja, u ovom slučaju je 5V koji spajamo direktno na 5V regulator Arduino mikrokontrolera,
- **GND (crna)** – negativni pol napajanja, spajamo ga direktno na GND Arduina.

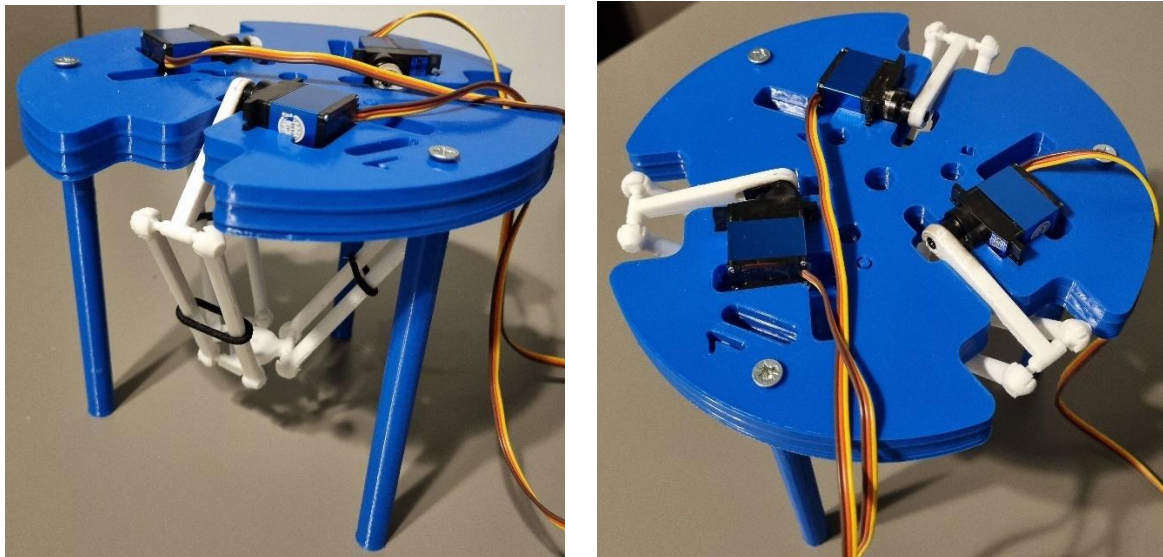
#### 4.5. Sastavljanje i spajanje robota

Nakon printanja svih dijelova možemo krenuti sa sastavljanjem robota. Na slici 4.5.1. vidimo jedan set isprintanih 3D dijelova potrebnih za sastavljanje jednog kraka robota. Nama su potrebno 3 takva seta.



*Slika 4.5.1-svi isprintani dijelovi jednog kraka Delta robota*

Najprije smo sastavili bazu sa sva 3 sloja te pričvrstili nožice pomoću vijaka. Zatim smo umetnuli Servo motore te na njih pričvrstili krakove. Sastavili smo sve kuglaste zglobove, pričvrstili end effector i pokušali ručno pomaknuti end effector. No, naišli smo na problem. Naime, najduži krakovi koji spajaju kraće krakove i end effector preko kuglastih zglobova su imali prevelike utore za kuglaste zglobove. To nije greška u dizajnu, već su ležajevi kuglastih zglobova bili namjerno predimenzionirani kako bi se stvaralo što manje trenja između plastičnih dijelova. To je rezultiralo ispadanjem zgloba iz ležaja u svakih nekoliko pokreta koje bi robot napravio. Rješenje tog problema bilo je pričvršćivanje dugih krakova elastičnim materijalom.



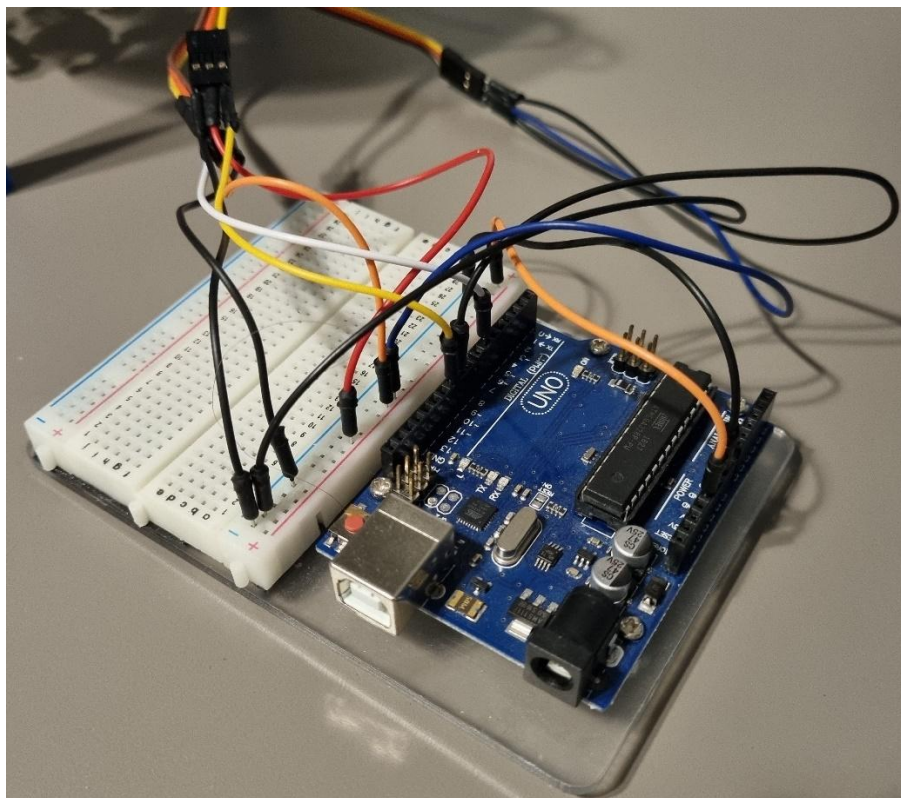
*Slika 4.5.2-sastavljen Delta robot bez Arduino Uno razvijene pločice i njene baze*

Slika 4.5.3. (lijevo) prikazuje bočni prikaz jednog sastavljenog kraka Delta robota na kojemu se mogu primijetiti sva 3 sloja baze, dok desna slika prikazuje maksimalno „ispruženu“ ruku Delta robota (najniža točka radnog prostora).



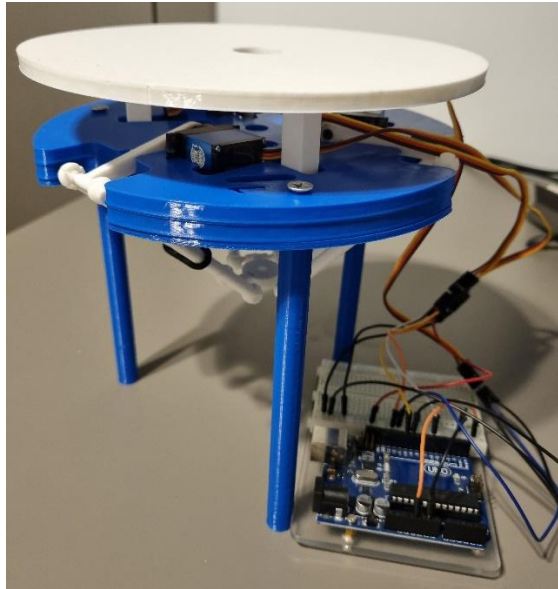


*Slika 4.5.3-bočni prikaz sastavljenog Delta robota (lijevo) i maksimalno „ispružena“ ruka (desno)*



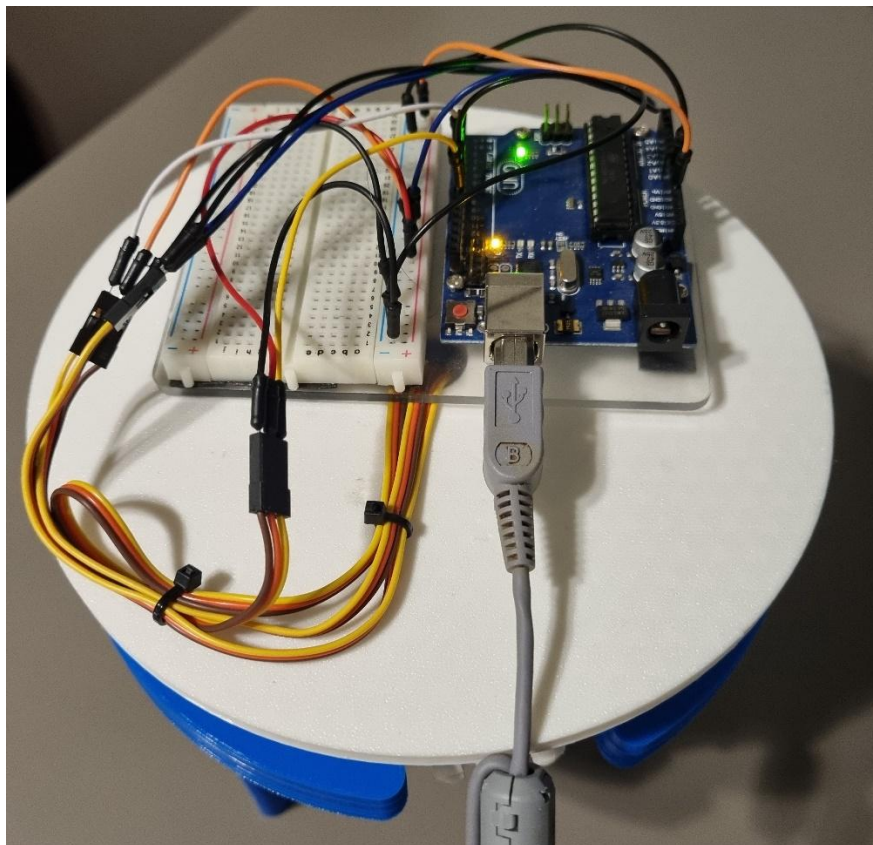
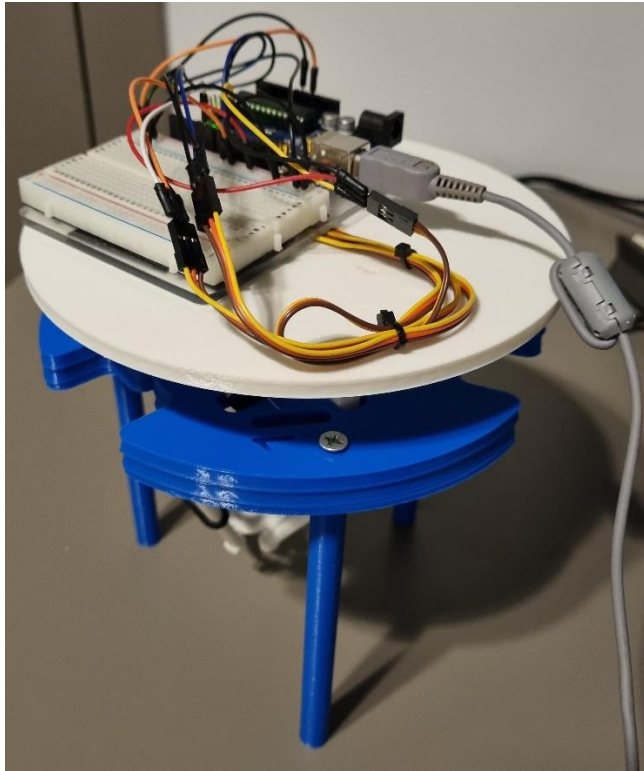
*Slika 4.5.4-korištena Arduino Uno razvojna pločica*

Nedostaje nam još printanje i postavljanje baze koja će držati Arduino razvojnu pločicu i dati jedan estetski dio našem radu kako se žice Servo motora ne bi zaplele oko zglobova robota.



*Slika 4.5.5-instalirana baza za Arduino mikrokontroler*

Nakon spajanja motora i pričvršćenja žica dobivamo konačni, 3D printani paralelni robotski manipulator prikazan na slikama 4.5.6. i 4.5.7.:

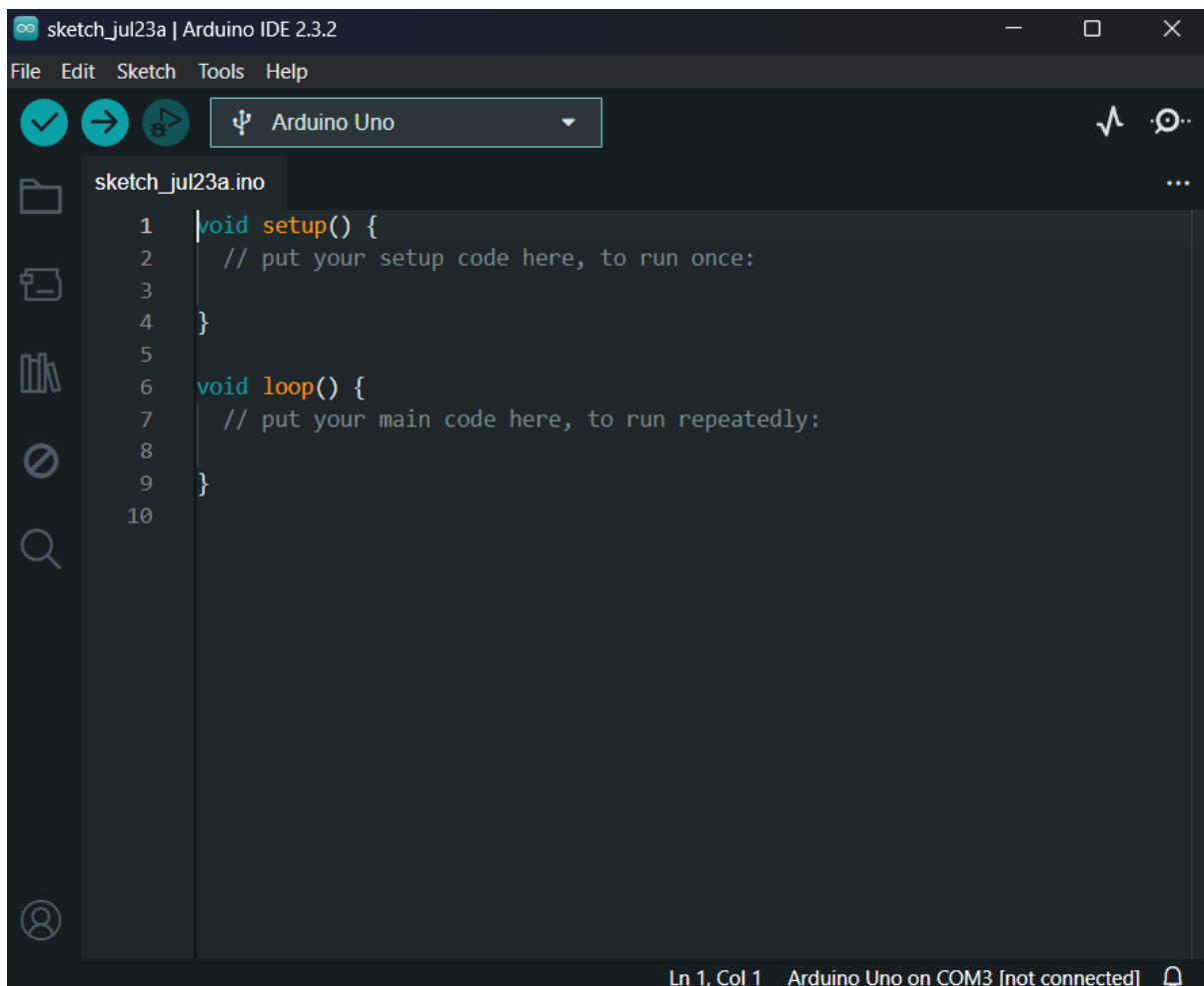


*Slika 4.5.6 i 4.5.7-sastavljen 3D printani Delta robot*



## 5. PROGRAMIRANJE I TESTIRANJE DELTA ROBOTA

Za ovaj zadatak potrebno je razviti program koji omogućuje komunikaciju između mikrokontrolera i Servo motora. Program ćemo napisati u C++ jeziku, koristeći dodatne funkcije za lakšu implementaciju i kontrolu motora. Pretvorba ovog, C++ koda u jezik razumljiv mikrokontroleru obavlja se putem Arduino IDE kompajlera. Instalacija Arduino IDE-a je besplatna i dostupna na službenoj web stranici.



```
sketch_jul23a.ino
1  void setup() {
2      // put your setup code here, to run once:
3
4  }
5
6  void loop() {
7      // put your main code here, to run repeatedly:
8
9  }
10
```

Slika 5.1-Arduino IDE programski paket

Na slici 5.1. možemo vidjeti da pri pokretanju programa postoje dvije prazne funkcije: „void setup()“ i „void loop()“. Ove funkcije su esencijalni dijelovi svakog Arduino programa. Funkcija „void setup()“ služi za inicijalizaciju programa i mikrokontrolera te se izvršava samo jednom, prilikom prvog pokretanja mikrokontrolera. Nakon nje, „void loop()“ funkcija se

neprestano vrti u krug, omogućujući kontinuirano izvršavanje koda sve dok je mikrokontroler uključen.

C++ programski jezik omogućuje dodavanje vanjskih biblioteka koje sadrže funkcije i druge vrste koda napisane u C++. Te biblioteke pojednostavljaju korištenje određenih funkcija tako da se one definiraju jednom u biblioteci te se mogu pozivati u kodu po potrebi. U našem slučaju koristit ćemo dvije biblioteke:

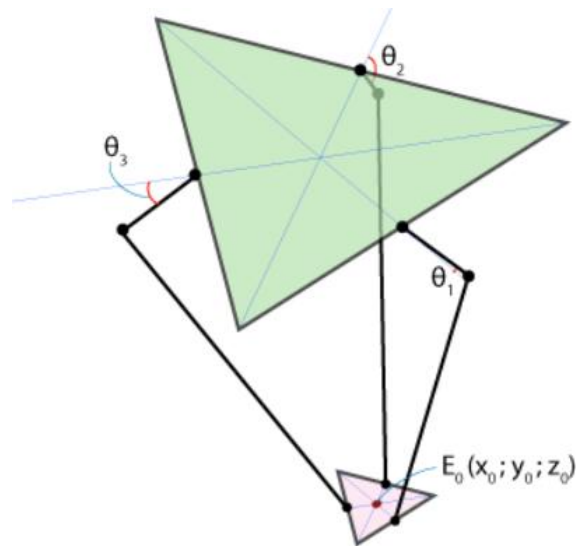
- „Servo.h“ – ova biblioteka nam služi za upravljanje samim Servo motorima i pojednostavljuje kod tako da sama generira potreban PWM signal koji će postaviti motor u željeni kut. Pozivamo je preko „servo.write(KUT)“ gdje KUT predstavlja broj između 0 i 180. Biblioteka je univerzalna, tj. može upravljati skoro svim malim Servo motorima te jedino trebamo pripaziti na maksimalni kut koji naš motor podržava, npr. naš TowerPro motor ima područje rada od 0-180°.
- „DeltaKinematics.h“ – biblioteka koja računa inverznu kinematiku potrebnu za pomicanje krakova našeg Delta robota. Kako je izračun kinematike sadržan u biblioteci, smanjuje obim potrebnog gotovog koda jer biblioteku možemo pozvati u kodu po potrebi

## 5.1. Funkcija izračuna inverzne kinematike

Naš robot radi na principu trodimenzionalnog koordinatnog sustava u kojem imamo definirane određene točke (x,y,z) na koje želimo da se end effector pomakne. U programskom kodu koji će biti detaljno opisan u poglavlju 5.2. imamo definirane x, y, z koordinate točaka koje robot treba slijediti od prve do zadnje kako bi izvršio određenu radnju. To znači da za svaku točku trebamo odrediti odgovarajuće kutove osovina svaka od 3 kraka robota te pomaknuti Servo motore u taj položaj. Proces takvog određivanja poznat je kao inverzna kinematika.

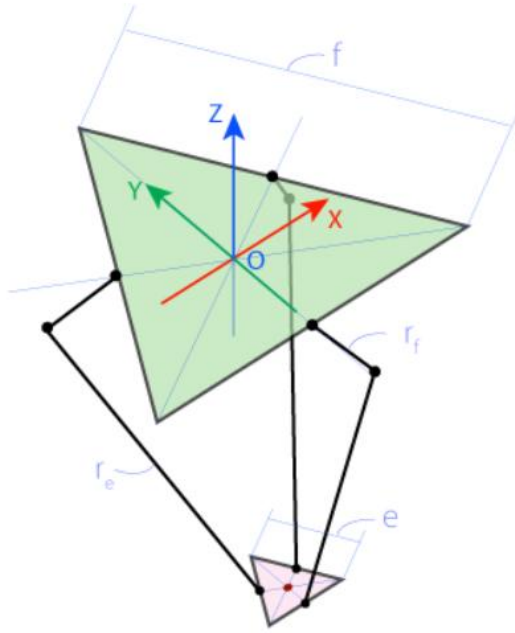
U ovom slučaju imamo definirane samo 4 točke u koordinatnom sustavu te se može reći da je jednostavnije ručno definirati kutove za svaki položaj i ne računati inverznu kinematiku u programskom kodu. Ovaj rad pokazuje primjer rada Delta robota i u stvarnom svijetu robot može imati stotine točaka pa je dobra praksa napisati funkciju za inverznu kinematiku. Funkcija

će kao ulazne varijable koristiti  $x$ ,  $y$ ,  $z$  koordinate željene točke te će izračunati kutove svakog Servo motora za tu točku. Za dobivanje formule inverzne kinematike trebamo skicirati Delta robot i označiti sve kutove. Kako vidimo na slici 5.1.1., baze robota predstavljaju 2 trokuta: fiksna baza sa Servo motorima je zelene boje, a pokretna baza s end effectorom je roza. Kutovi zglobova su  $\theta_1$ ,  $\theta_2$  i  $\theta_3$ , a točka  $E_0$  je krajnji, željeni položaj s koordinatama  $(x_0, y_0, z_0)$ .



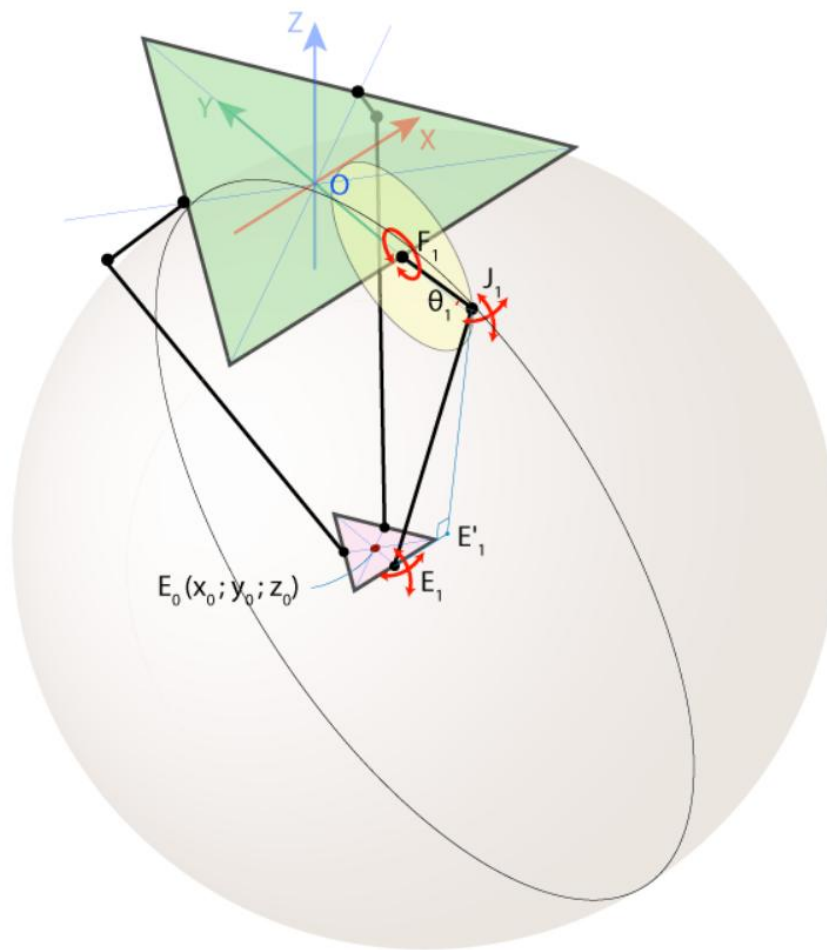
*Slika 5.1.1-Kinematska shema Delta robota*

Prvi korak određivanja formula inverzne kinematike je označavanje fiksnih veličina našeg robota:



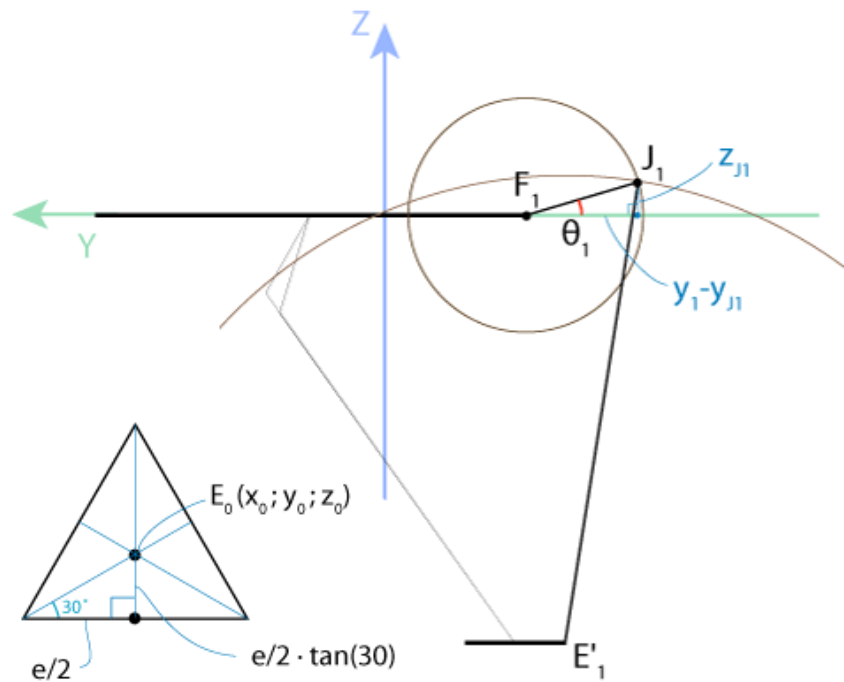
Slika 5.1.2-Označavanje svih fiksnih veličina našeg robota

Duljinu jedne strane trokuta baze označit ćemo sa  $f$ , duljinu jedne strane baze sa  $e$ , duljinu kraćeg kraka kao  $rf$  i duljinu dužeg kraka kao  $re$ . Referentni koordinatni sustav će se pozicionirati u središte trokuta baze, kako je prikazano na slici 5.1.3., tako da je z-koordinata end effectora uvijek negativna. Zbog oblika, fizičke građe našeg Delta robota te odabrane pozicije koordinatnog sustava, zglob  $F_1J_1$  se može rotirati samo u smjeru Y i Z plohe, formirajući tako krug s centrom u točki  $F_1$  i radijusom  $rf$ . Za razliku od  $F_1, J_1$  i  $E_1$  su „slobodni“ zglobovi, što znači da se vektor  $E_1J_1$  može slobodno rotirati relativno na  $E_1$ , formirajući tako 2D YZ krug s ishodištem u  $E_1$  i radijusom od  $re$ .



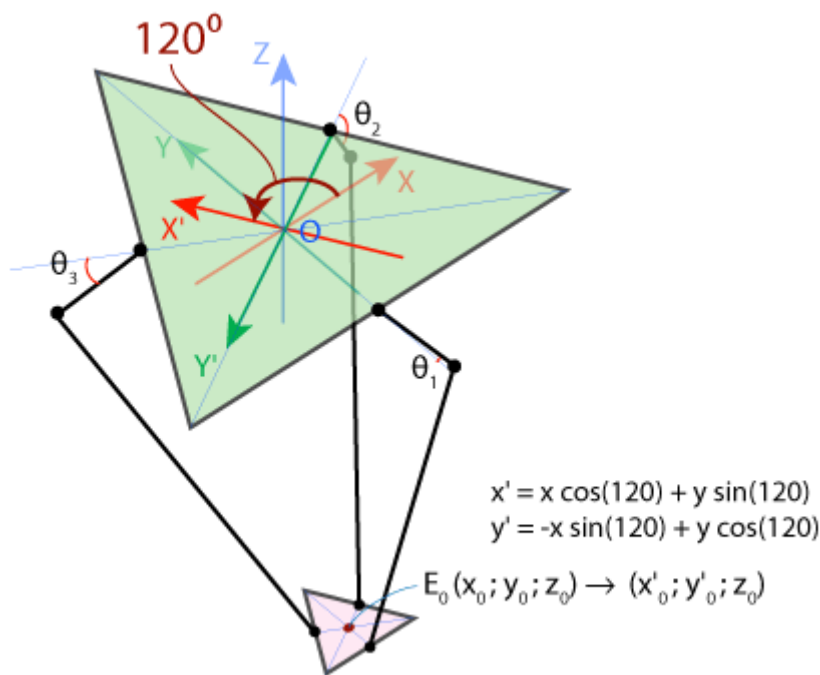
Slika 5.1.3-Formiranje 2D kruga s ishodištem u točki  $E_1$

Sjecište ove novonastale sfere i YZ plohe je krug u točki  $E'_1$  i radijusom od  $E'_1J_1$ , gdje je  $E'_1$  projekcija točke  $E_1$  na YZ plohu. Točka  $J_1$  u ovom slučaju predstavlja sjecište krugova sa poznatim radijusima i centralnim točkama u  $E'_1$  i  $F_1$ . To znači da ako znamo  $J_1$  možemo izračunati i kut  $\theta_1$ . Na slici 5.1.4. možemo vidjeti 2D projekciju navedenih sfera na vektorski prikaz Delta robota:



Slika 5.1.4-Projeksija kutova Delta robota na koordinatni sustav

Kako naš Delta robot ima 3 kraka, tj. Servo motora kojima je potrebno izračunati kut pomaka, osim **theta1** (izvod 5.1) trebamo izračunati kutove **theta2** i **theta3**. Zglob  $F_1J_1$  se može rotirati samo u smjeru Y i Z plohe, čime smo u potpunosti eliminirali X plohu. Ovu prednost ćemo iskoristiti za izračun preostalih 2 kutova robota tako da ćemo koordinatni sustav rotirati oko Z-osi za 120 stupnjeva (3 kraka, svaki 120°) suprotno od kazaljke na satu te na isti način izračunati preostale kutove. Rotiranjem koordinatnog sustava dobivamo novi X'Y'Z' sustav pomoću kojeg računamo kut **theta2**. Za određivanje kuta **theta3**, rotiramo koordinatni sustav 120 stupnjeva u smjeru kazaljke na satu. Taj princip ćemo iskoristiti u kodu u nastavku gdje koristimo funkciju za izračun kuta u YZ koordinatnom sustavu te ju pozivamo 3 puta, jednom za svaki referentni sustav.



Slika 5.1.5-Rotirani koordinatni sustav za izračun preostalih kutova robota

Da bismo postavili vrh robota u određenu točku, koristimo sljedeći postupak. Pretpostavimo da je ciljna točka E (prema slici 5.1.5.),

Točka E ima koordinate  $(x_0, y_0, z_0)$ ,

$$E(x_0, y_0, z_0) \quad (5.1)$$

Moramo izračunati kutove za postavljanje zglobova. Prvo, koristimo sljedeću jednakost:

$$EE_1 = \frac{e}{2} \tan 30^\circ = \frac{e}{2\sqrt{3}} \quad (5.2)$$

Iz ove jednadžbe zaključujemo da je prijenos proizvoljne pozicije E1 u E1' jednak:

$$E_1 \left( x_0, y_0 - \frac{e}{2\sqrt{3}}, z_0 \right) \rightarrow E'_1 \left( 0, y_0 - \frac{e}{2\sqrt{3}}, z_0 \right) \quad (5.3)$$

Ako pretpostavimo da vrijedi sljedeće:

$$E_1 E'_1 = x_0 \rightarrow E'_1 J_1 = \sqrt{E_1 J_1^2 - E_1 E'_1{}^2} = \sqrt{r_e^2 - x_0^2} \quad (5.4)$$

Transformacija u poziciju F definirana je kao:

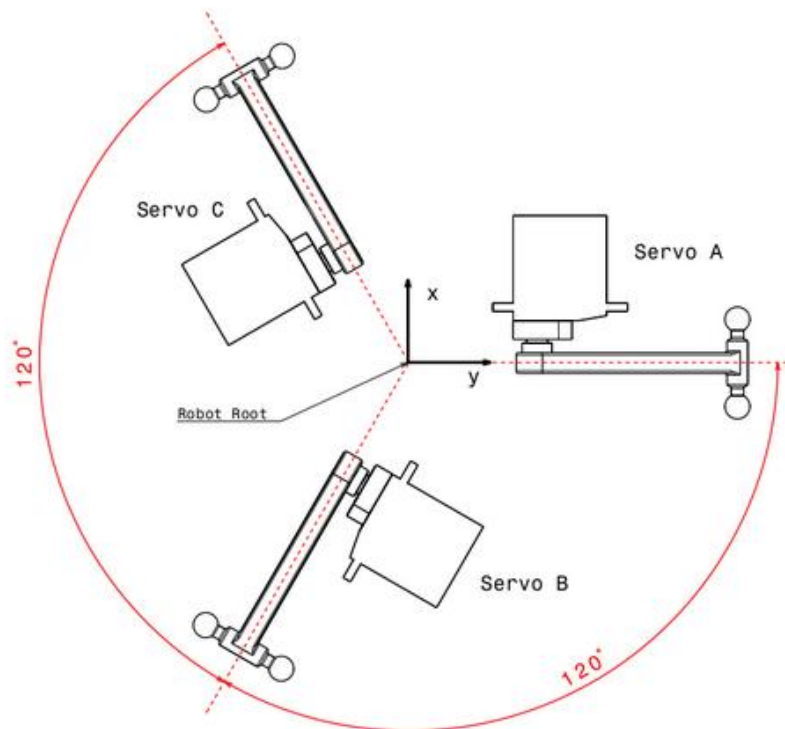
$$F_1 \left( 0, -\frac{f}{2\sqrt{3}}, 0 \right) \quad (5.5)$$

Te iz sustava jednažbi možemo dobiti: (5.6)

$$\begin{cases} (y_{J1} - y_{F1})^2 + (z_{J1} - z_{F1})^2 = r_f^2 \\ (y_{J1} - y_{E'1})^2 + (z_{J1} - z_{E'1})^2 = r_e^2 - x_0^2 \end{cases} \rightarrow \begin{cases} (y_{J1} + \frac{f}{2\sqrt{3}})^2 + z_{J1}^2 = r_f^2 \\ (y_{J1} - y_0 + \frac{e}{2\sqrt{3}})^2 + (z_{J1} - z_0)^2 = r_e^2 - x_0^2 \end{cases} \rightarrow J_1(0, y_{J1}, z_{J1})$$

Rješenjem ovog sustava jednažbi dobivamo konačnu jednažbu:

$$\theta_1 = \arctan\left(\frac{z_{J1}}{y_{F1} - y_{J1}}\right) \quad (5.7)$$



Slika 5.1.6-Prikaz kutne razlike između krakova koju koristimo

U izračunu inverzne kinematike



U nastavku slijedi kod funkcije koju koristimo za izračun kuta *theta* preko formule inverzne kinematike dobivene u izvodu 5.1.:

```
int DeltaKinematics::delta_calcAngleYZ(double *Angle, double x0, double y0,
double z0)
{
    double y1 = -0.5 * tan30 * BassTri; // f/2 * tan(30 deg)
        y0 -= 0.5 * tan30 * PlatformTri; // pomakni centar kružnice

    // z = a + b*y
    double aV = (x0*x0 + y0*y0 + z0*z0 + ArmLength*ArmLength -
RodLength*RodLength - y1*y1)/(2.0*z0);
    double bV = (y1-y0)/z0;

    double dV = -(aV+bV*y1)*(aV+bV*y1)+ArmLength*(bV*bV*ArmLength+ArmLength);

    double yj = (y1 - aV*bV - sqrt(dV))/(bV*bV + 1);
    double zj = aV + bV*yj;
    *Angle = atan2(-zj,(y1 - yj)) * 180.0/pi;

    return no_error; // return error, theta
}
```

Za izračun kutova *theta1*, *theta2* i *theta3* koristimo prije opisani postupak, pri čemu najprije izračunamo prvi kut te nakon toga pomaknemo koordinatni sustav za 120 stupnjeva:

```
a = 0; // kut theta1
b = 0; // kut theta2
c = 0; // kut theta3
int error = delta_calcAngleYZ(&a, x0, y0, z0);
if(error != no_error)
    return error;
error = delta_calcAngleYZ(&b, x0*cos120 + y0*sin120, y0*cos120-x0*sin120,
z0);
if(error != no_error)
    return error;
error = delta_calcAngleYZ(&c, x0*cos120 - y0*sin120, y0*cos120+x0*sin120,
z0);
```

U slučaju da izračun pojedinog kuta rezultira greškom, kod će se zaustaviti na toj grešci te će je vratiti u glavni kod, odakle se je ova funkcija pozvala.

## 5.2. Opis programskog koda Delta robota

Kako smo i prije opisali, naš kod počinje s definicijom svih vanjskih biblioteka koje ćemo koristiti. U ovom slučaju su to „Servo.h“ i „DeltaKinematics“.

```
#include <DeltaKinematics.h>
#include <Servo.h>
```

Zatim ćemo definirati našu konstrukcijsku funkciju. To u suštini služi kao pomoćna varijabla u koju možemo zapisati 3 float varijable (svaka predstavlja jednu os x, y, z koordinatnog sustava). Tako smo za svaku točku u našem radnom prostoru robota umjesto s 3 varijable (x1,y1,z1) definirali preko jedne konstrukcijske funkcije. One su u našem kodu definirane kao POS1, POS2, POS3 i POS4. Svaka ta definicija predstavlja jednu točku „Pick and place“ operacije koju će robot obavljati.

```
// Struktura za spremanje koordinata
struct Pos {
    float x;
    float y;
    float z;

    // Konstruktor za inicijalizaciju koordinata
    Pos(float xCoord, float yCoord, float zCoord) : x(xCoord), y(yCoord),
z(zCoord) {}
};

// Fiksne pozicije za operaciju pick and place
Pos POS1(15, 15, -100); // Pozicija za uzimanje objekta
Pos POS2(15, 15, -85); // Pozicija za podizanje objekta
Pos POS3(-25, -25, -75); // Pozicija za premještanje objekta
Pos POS4(-25, -25, -125); // Pozicija za spuštanje objekta
```

Dalje definiramo broj Servo motora koji ćemo koristiti u kodu te broj digitalnog pina na Arduino razvojnoj pločici koja pripada pojedinom Servo motoru. Ti pinovi trebaju podržavati generiranje PWM signala. Definiramo ih kao „const int“ što znači da je to varijabla tipa „int“ i da je konstanta, tj. ne može se mijenjati dalje u kodu jer su pinovi motora fiksno spojeni na Arduino.

```
Servo servo1;
Servo servo2;
Servo servo3;

const int pinA = 10; // Definiranje pina za servo A
const int pinB = 9;  // Definiranje pina za servo B
const int pinC = 6;  // Definiranje pina za servo C
```

Ovdje definiramo fizičke dimenzije robota koje koristimo u „DeltaKinematics.h“ biblioteci za izračun inverzne kinematike. Ovi brojevi su duljine pojedinih elemenata Delta robota u mm te predstavljaju redom: duljina manjeg kraka, duljina dužeg kraka, radius end effectora, radius baze.

```
DeltaKinematics DK(50, 90, 15, 31); // Fizičke dimenzije Delta robota
```

Ostalo nam je samo definiranje operativnih granica robota. U poglavlju 4.3.1. izračunali smo fizičke dimenzije robota te dobili i predviđeni radni prostor. Pomicanjem end effectora našeg fizičkog Delta robota dobili smo nešto manji radni prostor zbog tolerancija zglobova i duljina krakova. Taj radni prostor je efektivno područje u kojem je robot najprecizniji te ćemo u ovom slučaju pomicanje end effectora limitirati na spomenute koordinate. Osim radnog prostora, trebamo kalibrirati i Servo motore robota na apsolutnu nulu. To je opisano u poglavlju 5.3. te smo iz toga dobili operativne granice našeg Delta robota:

```
// Kalibracija servo motora na 0 stupnjeva
int zeroA = 178;
int zeroB = 170;
int zeroC = 175;

// Definiranje operativnih granica za svaku os (u mm)
const float Xmin = -25.0;
```

```
const float Xmax = 25.0;
const float Ymin = -25.0;
const float Ymax = 25.0;
const float Zmin = -135.0;
const float Zmax = -75.0;
```

U funkciji „void setup()“ koja se poziva samo jednom prilikom uključanja Arduino mikrokontrolera, definiramo početne vrijednosti varijabli te ulazno/izlazne pinove Arduina. U našem slučaju određujemo da je „servo1“ priključen na pin „pinA“ koji smo prije definirali da se odnosi na pin 10 te tako radimo i za servo2 i servo3 koji su spojeni na pinove 9 i 6. Osim toga, pokrećemo i serijsku komunikaciju s računalom kako bi mogli kontrolirati i vidjeti informacije o robotu te postavljamo end effector na svoj početni položaj. Taj položaj je 20° manji od apsolutne nule Servo motora.

```
void setup() {
  Serial.begin(115200);

  servo1.attach(pinA);
  servo2.attach(pinB);
  servo3.attach(pinC);

  // Pomicanje servo motora na početni položaj
  servo1.write(zeroA - 20);
  servo2.write(zeroB - 20);
  servo3.write(zeroC - 20);

  Serial.println("Delta Robot Početak Rada");
  Serial.println("Petar Miškulin - RiTeh 2024.");
  Serial.println("Unesi 'm' za manualni način rada, 'a' za automatski način
rada");
  Serial.println("U manualnom načinu, unesi koordinate kao 'x,y,z'");
}
```

Funkcija „servoControl()“, koja je prikazana u nastavku, je najbitnija jer ona naređuje poziciju na koju će se servo motori pomaknuti i time kontrolira sve pomake našeg end effectora. U početnom dijelu, preko funkcije „Serial.println“, na računalo ispisujemo koordinate koje je

funkcija dobila kako bi pomakla robot. Nakon toga ispisujemo dobivene kutove preko izračuna inverzne kinematike.

```
void servoControl() {
  Serial.print("Pomaknut na poziciju: ");
  Serial.print(String(DK.x));
  Serial.print(", ");
  Serial.print(String(DK.y));
  Serial.print(", ");
  Serial.println(String(DK.z));
  Serial.print("Inverzna kinematika: ");
  Serial.println(String(DK.a) + "," + String(DK.b) + "," + String(DK.c));
  Serial.println();

  thetaA = zeroA - DK.a;
  thetaB = zeroB - DK.b;
  thetaC = zeroC - DK.c;
  servo1.write(thetaA);
  servo2.write(thetaB);
  servo3.write(thetaC);
}
```

Kutevi  $\theta_A$ ,  $\theta_B$  i  $\theta_C$  predstavljaju željene kutove na koje se servo motori trebaju pomaknuti. Preko funkcije inverzne kinematike dobili smo  $\theta_1$ ,  $\theta_2$  i  $\theta_3$  kuteve (u kodu  $DK.a$ ,  $DK.b$  i  $DK.c$ ) i oni se kreću od 0-180°. Servo motori su fizički zakrenuti za 180°, što znači da, ako preko izračuna inverzne kinematike dobijemo da za željenu poziciju robota treba pomaknuti servo motora na poziciju od 30°, stvarna željena pozicija bit će 180°-30°, tj. „ $\theta_A = zeroA - DK.a$ “. U tom slučaju  $zeroA$  je apsolutna nula koja u ovom slučaju ima pomak od nekoliko stupnjeva od idealnih 180°. Funkcija „`servo1.write(thetaA)`“ naređuje servo1 motoru da se pomakne za kut  $\theta_A$ , što vrijedi i za ostale motore.

Funkcija „`isWithinLimits()`“ provjerava je li željena točka, na koju pomičemo end effector, u operativnim granicama motora koje smo definirali na početku koda.

```
bool isWithinLimits(float x, float y, float z) {
  return (x >= Xmin && x <= Xmax &&
         y >= Ymin && y <= Ymax &&
```

```
z >= Zmin && z <= Zmax);  
}
```

Funkcija „pickAndPlaceOperation()“ poziva „servoControl()“ funkciju kojoj daje željene koordinate na koje da pomakne end effector. Ona preko „switch“ naredbe pomiče end effector između 4 predefiniране pozicije i tako simulira „Pick and place“ operaciju robota. Za svaku točku izračunamo inverznu kinematiku preko „DK.inverse()“ funkcije. Između svakog pomaka imamo pauzu od 2s.

```
void pickAndPlaceOperation() {  
    static int step = 0; // Korak za praćenje pozicijske sekvence  
  
    switch (step) {  
        case 0:  
            DK.x = POS1.x;  
            DK.y = POS1.y;  
            DK.z = POS1.z;  
            break;  
        case 1:  
            DK.x = POS2.x;  
            DK.y = POS2.y;  
            DK.z = POS2.z;  
            break;  
        case 2:  
            DK.x = POS3.x;  
            DK.y = POS3.y;  
            DK.z = POS3.z;  
            break;  
        case 3:  
            DK.x = POS4.x;  
            DK.y = POS4.y;  
            DK.z = POS4.z;  
            break;  
        case 4:  
            DK.x = POS3.x;  
            DK.y = POS3.y;  
            DK.z = POS3.z;
```

```

        break;
    case 5:
        DK.x = POS2.x;
        DK.y = POS2.y;
        DK.z = POS2.z;
        break;
    }

    DK.inverse();
    servoControl();
    delay(2000);

    step = (step + 1) % 6; // Povećaj indeks koraka i vrati na 0 nakon 5
}

```

Funkcija „void loop()“ spaja sve prijašnje funkcije u gotov programski kod koji se izvršava kontinuirano u krug, sve dok Arduino mikrokontroler ima napajanje. Najprije provjeravamo jesmo li dobili naredbu od računala na kojem je pokrenut Arduino IDE programski paket. Naredba dolazi preko serijske veze te u ovom slučaju može biti „m“ ili „a“. Dobivenu naredbu spremamo u String varijablu „inputString“ i u slučaju da je „m“ pokrećemo manualni način rada robota, a u slučaju da je „a“ pokrećemo automatski način rada. Početno stanje rada robota je u automatskom stanju.

```

// Provjeri je li dostupno nešto na serijskoj vezi
if (Serial.available() > 0) {
    char incomingByte = Serial.read();
    // Provjeri je li pritisnuta tipka za novi redak ili povratak
    if (incomingByte == '\n' || incomingByte == '\r') {
        if (inputString.length() > 0) {
            if (inputString == "m") {
                manualMode = true;
                Serial.println("Manualni način rada aktiviran. Unesi koordinate kao 'x,y,z'");
            } else if (inputString == "a") {
                manualMode = false;
                Serial.println("Automatski način rada aktiviran");
            }
        }
    }
}

```

- **Automatski način rada** – robot simulira unaprijed programiranu „Pick and place“ operaciju u 4 predefinirane točke.
- **Manualni način rada** – robot čeka naredbu od korisnika preko serijske komunikacije s računalom koje mu naređuje u koju točku će se pomaknuti.

Ukoliko odaberemo automatski način rada ili ne pošaljemo nikakvu naredbu, robot pokreće „pickAndPlaceOperation()“ funkciju:

```
// Ako nije manualni način rada, izvrši pick and place operaciju
if (!manualMode) {
    pickAndPlaceOperation();
}
```

U slučaju da smo odabrali manualni način rada, robot čeka koordinate koje mu zadaje korisnik. Dobivena naredba treba sadržavati X, Y i Z koordinate odvojene zarezom, npr. 50, 40, 55. U manualnom načinu najprije odvajamo sva 3 dobivena broja nakon zareza u svoje varijable (x\_manual, y\_manual i z\_manual). Dobivene koordinate ispišemo korisniku preko „Serial.println“ radi potvrde dobivenih podataka.

```
else if (manualMode) {
    inputString.trim();

    int commaIndex1 = inputString.indexOf(',');
    int commaIndex2 = inputString.lastIndexOf(',');

    // Provjeri je li ulaz valjan (ima dva zareza)
    if (commaIndex1 != -1 && commaIndex2 != -1 && commaIndex1 !=
commaIndex2) {
        float x_manual = inputString.substring(0, commaIndex1).toFloat();
        float y_manual = inputString.substring(commaIndex1 + 1,
commaIndex2).toFloat();
        float z_manual = inputString.substring(commaIndex2 + 1).toFloat();

        Serial.print("Primljene koordinate: ");
        Serial.print(x_manual); Serial.print(", ");
        Serial.print(y_manual); Serial.print(", ");
        Serial.println(z_manual);
    }
}
```



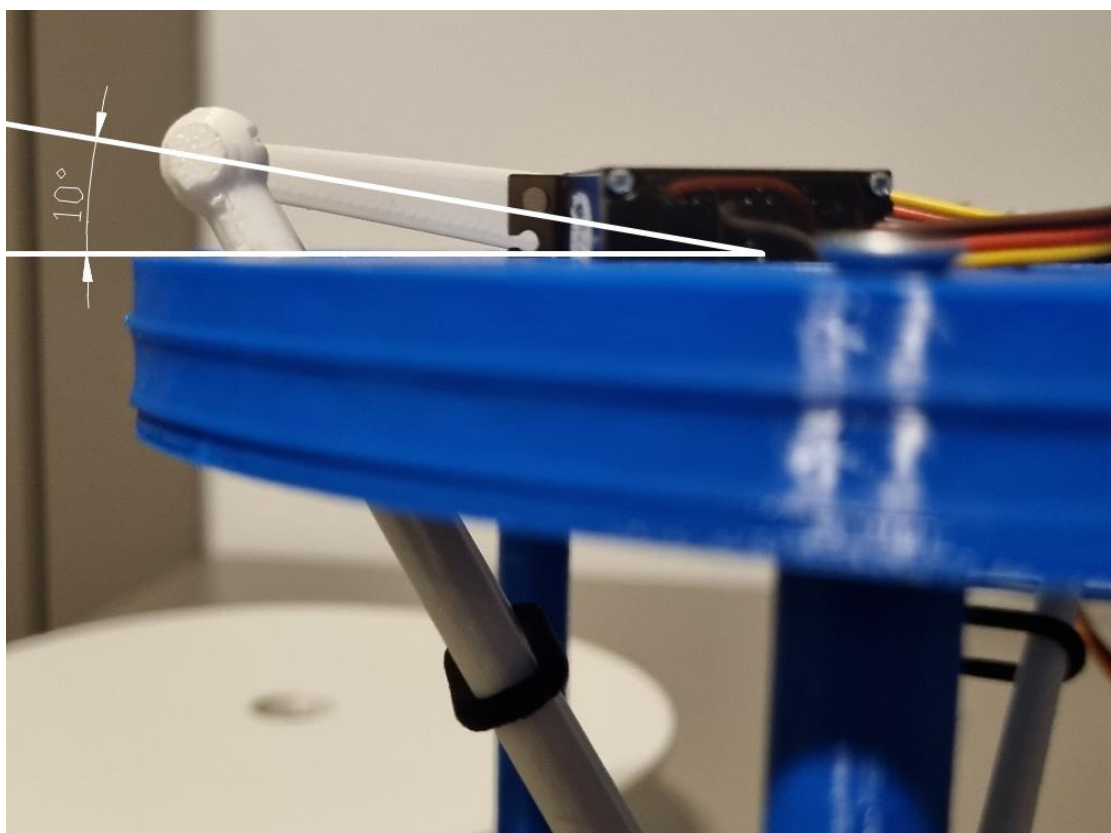
Ukoliko su dobivene koordinate točne, provjeravamo jesu li unutar predefiniranih operativnih granica robota sa „isWithinLimits()“ funkcijom. Ako jesu, izračunavamo inverznu kinematiku svake točke te pozivamo „servoControl()“ funkciju koja pomiče end effector na željenu poziciju.

```
// Provjeri je li unutar operativnih granica
if (isWithinLimits(x_manual, y_manual, z_manual)) {
    DK.x = x_manual;
    DK.y = y_manual;
    DK.z = z_manual;
    DK.inverse();
    servoControl();
    delay(1000);
} else {
    Serial.println("Koordinate izvan radne plohe!");
}
} else {
    Serial.println("Neispravan format unosa. Unesi koordinate kao
'x,y,z'");
}
}
inputString = ""; // Očisti ulazni string
}
} else {
    inputString += incomingByte; // Dodaj znak ulaznom stringu
}
}
```

### 5.3. Kalibracija Servo motora na apsolutnu nulu

U prijašnjim poglavljima izračunali smo dimenzije te izradili bazu Delta robota na kojoj su smješteni Servo motori. Oni su, zbog orijentacije, zakrenuti na bazi za  $180^\circ$ , što znači da umjesto od  $0^\circ$  do  $180^\circ$ , oni se okreću od  $180^\circ$  do  $0^\circ$  za pomicanje end effectora od svoje najviše pozicije Z-osi do najniže.

Prilikom pričvršćivanja 3D printanih krakova na osovine Servo motora dolazi do razlike u visini zbog tolerancije samih printanih dijelova i tolerancije pozicije osovine od motora do motora. To znači da za postavljenih  $180^\circ$ , umjesto vodoravno, krakovi će imati pomak od nekoliko stupnjeva što će utjecati na poziciju end effectora i ona neće biti precizna. Rješenje je dodavanje pomaka u kodu za svaki Servo motor tako da se on nalazi u apsolutnoj nuli, tzv. kalibracija osovine. Nju ćemo napraviti tako da postavimo sve motore na poziciju od  $180^\circ$ , smanjivati ili povećavati poziciju s korakom od  $1^\circ$  sve dok krak robota ne stoji vodoravno. Kada dobijemo željeni kut upisat ćemo ga u djelu koda koji se odnosi na kalibraciju. Na sljedećoj slici vidimo inicijalno odstupanje B Servo motora od apsolutne nule:



*Slika 5.3.1-Odstupanje B Servo motora od apsolutne nule prije kalibracije*

Dobivena odstupanja su prikazana u početnom dijelu koda (u ovom slučaju vidimo da je apsolutna nula B motora postavljena na 170° jer prema slici 5.3.1 vidimo da ima odstupanje od 10°):

```
// Kalibracija servo motora na 0 stupnjeva  
int zeroA = 178;  
int zeroB = 170;  
int zeroC = 175;
```

## 6. ZAKLJUČAK

U ovom smo radu opisali postupak izrade Delta robota s 3 kraka. Izračunali smo radni prostor i dimenzije te prema njima izradili Delta robot kako bismo prikazali prednosti i mane paralelnih manipulatora. Naš robot se ne može koristiti za bilo koji ozbiljan rad jer su 3D printani dijelovi krhki i zato smo Servo motore, odabrali tako da mogu podnijeti samo težinu end effectora, bez dodatnih opterećenja. Također, zbog veće tolerancije 3D printanih zglobova, one se prenose na end effector te dolaze do izražaja zbog duljine samih krakova robota. Delta roboti se sve više koriste u industrijskim postrojenjima diljem svijeta zbog njihove brzine, preciznosti i velike nosivosti te daljnjim razvojem te grane robotike možemo postići sve veću i precizniju automatizaciju procesa, što je krajnjem potrošaču dostupno kao jeftiniji i standardiziraniji proizvod.

## 7. LITERATURA

[1] Rey L, Clavel R. The delta parallel robot. In *Parallel Kinematic Machines: Theoretical Aspects and Industrial Requirements* 1999 Sep 22 (pp. 401-417). London: Springer London.

[2] Pierrot F, Reynaud C, Fournier A. DELTA: a simple and efficient parallel robot. *Robotica*. 1990 Apr;8(2):105-9.

[3] „Servo motor working principle“, s Interneta (pregledano 8.7.2024):

<https://www.globalspec.com/pfdetail/motors/servo-motor-working-principle#:~:text=A%20servo%20motor%20is%20a,provides%20feedback%20to%20the%20potentiometer>

[4] „How servo motors work“, s Interneta (pregledano 11.7.2024):

<https://www.jameco.com/Jameco/workshop/Howitworks/how-servo-motors-work.html>

[5] „TowerPro MG92B“, s Interneta (pregledano 2.6.2024):

<https://www.towerpro.com.tw/product/mg92b/>

[6] Arduino, "Arduino Uno R3 Product Reference Manual", 2024.

[7] „Kinematics calculator“, s Interneta (pregledano 14.7.2024):

<https://www.marginallyclever.com/other/samples/fk-ik-test.html>

[8] „Delta robots market“, s Interneta (pregledano 9.7.2024):

<https://www.nextmsc.com/report/delta-robots-market>

[9] „What are delta robots used for“, s Interneta (pregledano 9.7.2024):

<https://acrome.net/post/what-are-delta-robots-used-for>

[10] „Gough-Stewart platform and a cable robot“, s Interneta (pregledano 10.7.2024):

[https://www.researchgate.net/figure/Comparative-display-of-a-general-Gough-Stewart-platform-a-and-a-cable-robot-b\\_fig1\\_317721191](https://www.researchgate.net/figure/Comparative-display-of-a-general-Gough-Stewart-platform-a-and-a-cable-robot-b_fig1_317721191)

[11] Bouri M, Clavel R. The linear delta: Developments and applications. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)* 2010 Jun 7 (pp. 1-8). VDE.

[12] „Servo Motor Working Principle“, s Interneta (pregledano 11.7.2024):

<https://www.globalspec.com/pfdetail/motors/servo-motor-working-principle#:~:text=A%20servo%20motor%20is%20a,provides%20feedback%20to%20the%20potentiometer>

[13] „Pulse-width modulation“, s Interneta (pregledano 20.7.2024):

<https://www.shutterstock.com/search/pulse-width-modulation>

[14] „PWM led dimming“, s Interneta (pregledano 21.7.2024):

<https://nightsea.com/articles/pwm-led-dimming>

[15] „How servo motors work“, s Interneta (pregledano 11.7.2024):

<https://www.jameco.com/Jameco/workshop/Howitworks/how-servo-motors-work.html>

[16] „The choice between servo motors and stepper motors“, s Interneta (pregledano 28.7.2024):

<https://blog.orientalmotor.com/the-choice-between-servo-motors-and-stepper-motors>

[17] Arduino Docs, "Arduino Uno R3 Full Pinout", 2024.

[18] „Inverse Kinematics Calculations“, s Interneta (pregledano 2.8.2024):

<https://www.marginallyclever.com/other/samples/fk-ik-test.html>

[19] Hadfield H, Wei L, Lasenby J. The forward and inverse kinematics of a delta robot. In Advances in Computer Graphics: 37th Computer Graphics International Conference, CGI 2020, Geneva, Switzerland, October 20–23, 2020, Proceedings 37 2020 (pp. 447-458). Springer International Publishing.

## 8. SAŽETAK I KLJUČNE RIJEČI NA HRVATSKOM I ENGLLESKOM JEZIKU

U ovom je radu opisan postupak izračuna, izrade i rada Delta robota. Izračunali smo radni prostor i dimenzije te prema njima izradili Delta robot kako bismo prikazali prednosti i mane paralelnih manipulatora. Govorili smo o postupku 3D printanja te kako nam isto može pomoći smanjiti troškove samih materijala za izradu robota. Također, prikazali smo kako male tolerancije zglobova mogu utjecati na preciznost samog end effectora te da se one povećavaju s duljim krakovima Delta robota. I za kraj, dobili smo formulu za izračun inverzne kinematike te ju primijenili u kodu Arduino razvojne pločice koja upravlja robotom.

**Ključne riječi:** robot, Delta robot, paralelni manipulator, Arduino, mikrokontroler, kod, kinematika, inverzna kinematika, motori, Servo

This paper describes the calculation, manufacturing and operation of the Delta robot. We calculated the working space and dimensions and built the Delta robot according to them in order to show the advantages and disadvantages of parallel manipulators. We talked about the 3D printing process and how it can help us reduce the cost of the materials used to make robots. We also showed how small joint tolerances can affect the precision of the end effector itself and that they increase with longer arms of the Delta robot. And finally, we got the formula for calculating the inverse kinematics and applied it in the code of the Arduino development board that controls the robot.

**Key words:** robot, Delta robot, parallel manipulator, Arduino, microcontroller, code, kinematics, inverse kinematics, motors, Servo

## 9. PRILOG A – CIJELI PROGRAMSKI KOD ARDUINO MIKROKONTROLERA

```
#include <DeltaKinematics.h>
#include <Servo.h>

// Struktura za spremanje koordinata
struct Pos {
    float x;
    float y;
    float z;

    // Konstruktor za inicijalizaciju koordinata
    Pos(float xCoord, float yCoord, float zCoord) : x(xCoord), y(yCoord),
z(zCoord) {}
};

// Fiksne pozicije za operaciju pick and place
Pos POS1(15, 15, -100); // Pozicija za uzimanje objekta
Pos POS2(15, 15, -85); // Pozicija za podizanje objekta
Pos POS3(-25, -25, -75); // Pozicija za premještanje objekta
Pos POS4(-25, -25, -125); // Pozicija za spuštanje objekta

Servo servo1;
Servo servo2;
Servo servo3;

const int pinA = 10; // Definiranje pina za servo A
const int pinB = 9; // Definiranje pina za servo B
const int pinC = 6; // Definiranje pina za servo C

DeltaKinematics DK(50, 90, 15, 31); // Fizičke dimenzije Delta robota

// Kalibracija servo motora na 0 stupnjeva
int zeroA = 178;
int zeroB = 170;
int zeroC = 175;
```



```

// Deklaracija ostalih varijabli
int thetaA, thetaB, thetaC;
bool manualMode = false;

// Definiranje operativnih granica za svaku os (u mm)
const float Xmin = -25.0;
const float Xmax = 25.0;
const float Ymin = -25.0;
const float Ymax = 25.0;
const float Zmin = -135.0;
const float Zmax = -75.0;

void setup() {
  Serial.begin(115200);

  servo1.attach(pinA);
  servo2.attach(pinB);
  servo3.attach(pinC);

  // Pomicanje servo motora na početni položaj
  servo1.write(zeroA - 20);
  servo2.write(zeroB - 20);
  servo3.write(zeroC - 20);

  Serial.println("Delta Robot Početak Rada");
  Serial.println("Petar Miškulin - RiTeh 2024.");
  Serial.println("Unesi 'm' za manualni način rada, 'a' za automatski način
rada");
  Serial.println("U manualnom načinu, unesi koordinate kao 'x,y,z'");
}

void servoControl() {
  Serial.print("Pomaknut na poziciju: ");
  Serial.print(String(DK.x));
  Serial.print(", ");
  Serial.print(String(DK.y));
  Serial.print(", ");
  Serial.println(String(DK.z));
}

```

```

Serial.print("Inverzna kinematika: ");
Serial.println(String(DK.a) + "," + String(DK.b) + "," + String(DK.c));
Serial.println();

thetaA = zeroA - DK.a;
thetaB = zeroB - DK.b;
thetaC = zeroC - DK.c;
servo1.write(thetaA);
servo2.write(thetaB);
servo3.write(thetaC);
}

bool isWithinLimits(float x, float y, float z) {
    return (x >= Xmin && x <= Xmax &&
            y >= Ymin && y <= Ymax &&
            z >= Zmin && z <= Zmax);
}

void pickAndPlaceOperation() {
    static int step = 0; // Korak za praćenje pozicijske sekvence

    switch (step) {
        case 0:
            DK.x = POS1.x;
            DK.y = POS1.y;
            DK.z = POS1.z;
            break;
        case 1:
            DK.x = POS2.x;
            DK.y = POS2.y;
            DK.z = POS2.z;
            break;
        case 2:
            DK.x = POS3.x;
            DK.y = POS3.y;
            DK.z = POS3.z;
            break;
        case 3:

```

```

    DK.x = POS4.x;
    DK.y = POS4.y;
    DK.z = POS4.z;
    break;
case 4:
    DK.x = POS3.x;
    DK.y = POS3.y;
    DK.z = POS3.z;
    break;
case 5:
    DK.x = POS2.x;
    DK.y = POS2.y;
    DK.z = POS2.z;
    break;
}

DK.inverse();
servoControl();
delay(2000);

step = (step + 1) % 6; // Povećaj indeks koraka i vrati na 0 nakon 5
}

void loop() {
    static String inputString = ""; // String za spremanje ulaznih podataka

    // Provjeri je li dostupno nešto na serijskoj vezi
    if (Serial.available() > 0) {
        char incomingByte = Serial.read();
        // Provjeri je li pritisnuta tipka za novi redak ili povratak
        if (incomingByte == '\n' || incomingByte == '\r') {
            if (inputString.length() > 0) {
                if (inputString == "m") {
                    manualMode = true;
                    Serial.println("Manualni način rada aktiviran. Unesi koordinate kao
'x,y,z'");
                } else if (inputString == "a") {
                    manualMode = false;

```

```

    Serial.println("Automatski način rada aktiviran");
} else if (manualMode) {
    inputString.trim();

    int commaIndex1 = inputString.indexOf(',');
    int commaIndex2 = inputString.lastIndexOf(',');

    // Provjeri je li ulaz valjan (ima dva zareza)
    if (commaIndex1 != -1 && commaIndex2 != -1 && commaIndex1 !=
commaIndex2) {
        float x_manual = inputString.substring(0, commaIndex1).toFloat();
        float y_manual = inputString.substring(commaIndex1 + 1,
commaIndex2).toFloat();
        float z_manual = inputString.substring(commaIndex2 + 1).toFloat();

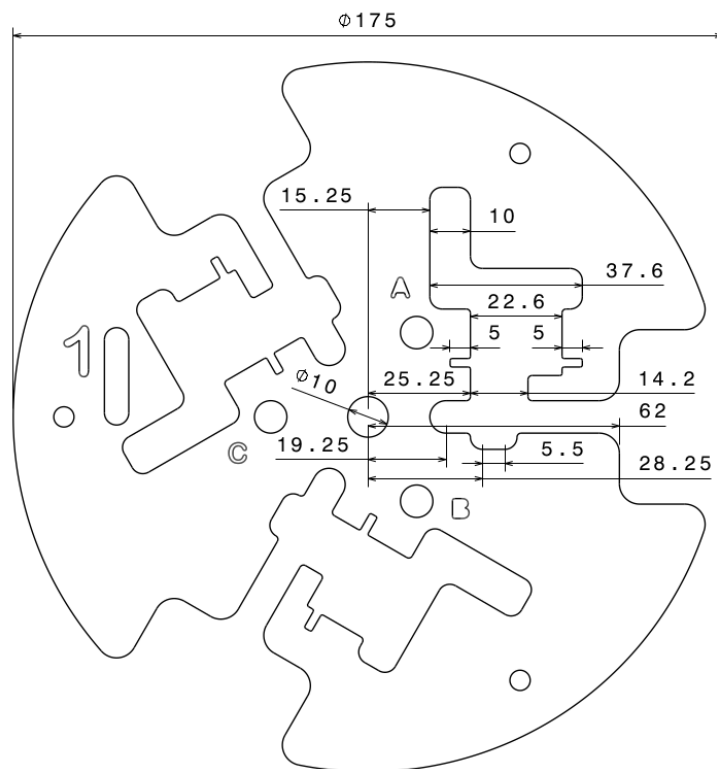
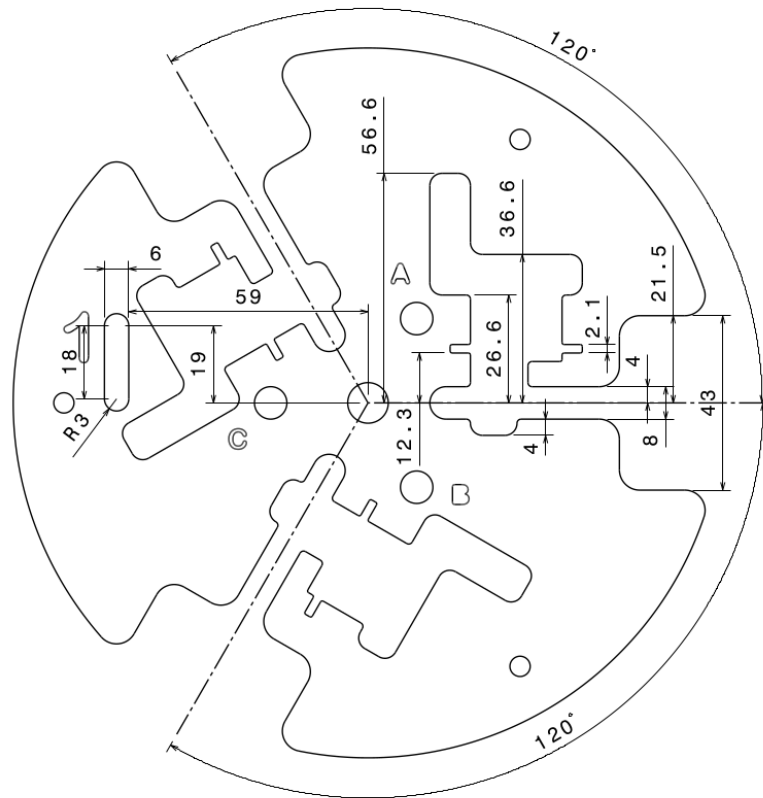
        Serial.print("Primljene koordinate: ");
        Serial.print(x_manual); Serial.print(", ");
        Serial.print(y_manual); Serial.print(", ");
        Serial.println(z_manual);

        // Provjeri je li unutar operativnih granica
        if (isWithinLimits(x_manual, y_manual, z_manual)) {
            DK.x = x_manual;
            DK.y = y_manual;
            DK.z = z_manual;
            DK.inverse();
            servoControl();
            delay(1000);
        } else {
            Serial.println("Koordinate izvan radne plohe!");
        }
    } else {
        Serial.println("Neispravan format unosa. Unesi koordinate kao
'x,y,z'");
    }
    inputString = ""; // Očisti ulazni string
}

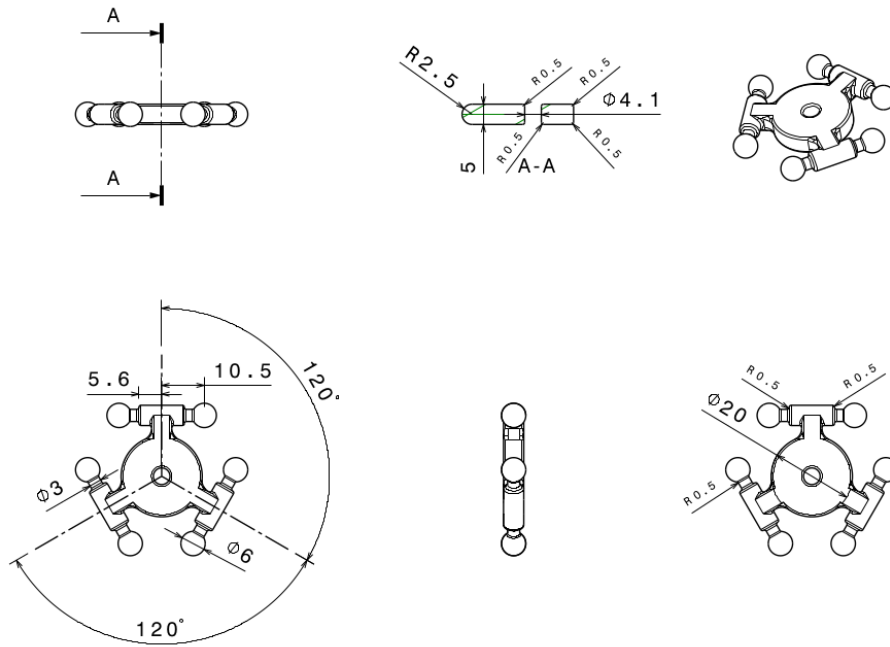
```

```
} else {  
    inputString += incomingByte; // Dodaj znak ulaznom stringu  
}  
  
}  
  
// Ako nije manualni način rada, izvrši pick and place operaciju  
if (!manualMode) {  
    pickAndPlaceOperation();  
}  
}
```

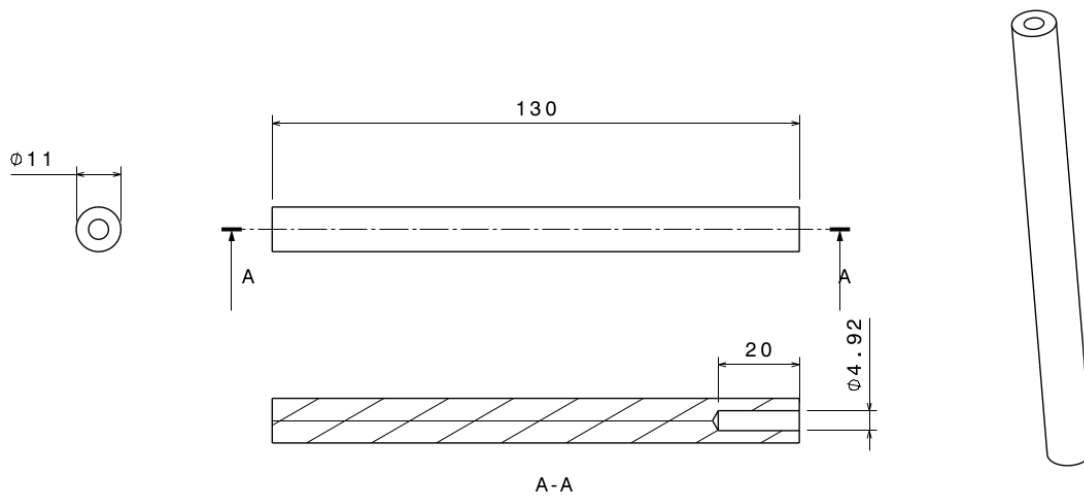
## 10. PRILOG B – KOTIRANI NACRTI BAZE



# 11. PRILOG C – KOTIRANI NACRTI END EFFECTORA

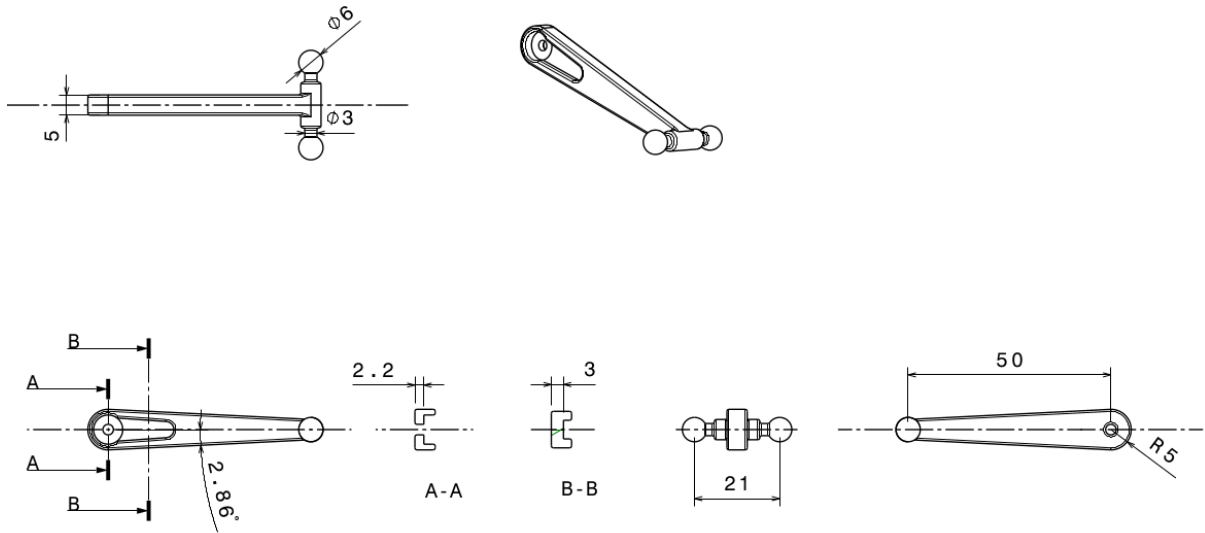


## 12. PRILOG D – KOTIRANI NACRTI NOSAČA BAZE





## 12. PRILOG E – KOTIRANI NACRTI KRAČEG KRAKA ROBOTA



### 13. PRILOG F – KOTIRANI NACRTI DULJEG KRAKA ROBOTA

