Edukacijska robotska ruka

Keretić, Damian

Undergraduate thesis / Završni rad

2020

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:461175

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Damian Keretić

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentori:

Prof. dr. sc. Mladen Crneković, dipl. ing.

Student:

Damian Keretić

Zagreb, 2020.

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

Zahvaljujem se mentoru prof.dr.sc. Mladenu Crnekoviću na pomoći i sugestijama pruženim tijekom izrade ovog rada.

Zahvaljujem se cijeloj svojoj obitelji, posebno roditeljima koji su mi uvijek bili velika podrška.

Svojoj djevojci Eleni koja je uvijek bila tu kad je zatrebalo.

Svojim prijateljima i kolegama s faksa s kojima sam sve ovo zajedno prolazio.

I na kraju se posebno zahvaljujem svom djedu Franji na pomoći i savjetima.

Damian Keretić

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

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **DAMIAN KERETIĆ** Mat. br.: 0035210157

Naslov rada na hrvatskom jeziku: **EDUKACIJSKA ROBOTSKA RUKA**

Naslov rada na engleskom jeziku: **EDUCATIONAL ROBOT ARM**

Opis zadatka:

Za mnoge zaljubljenike u robotiku edukacija počinje željom za izradom vlastite robotske ruke. Zato je najbolje uzeti gotov projekt s uputama, a tijekom izrade i montaže naučiti način rada pojedinih elemenata. U vrijeme 3D printera, jeftine elektronike i ostalih elemenata od kojih je sastavljena robotska ruka, moguće je za prihvatljiv iznos izraditi vlastitog robota.

Potrebno je istražiti tržište gotovih edukacijskih robota i napraviti njihovu usporedbu, a zatim predložiti svoje rješenje uz pretpostavku da korisnik ima 3D printer i može nabaviti hobističke dijelove koje sam ne izrađuje (motori, senzori, kontroler itd.).

U radu je potrebno:

- definirati mehaničku konstrukciju robot s 4 stupnjeva slobode gibanja i mogućnošću hvatanja predmeta
- odabrati motore s pripadnim kontrolerima
- odabrati potrebne senzore i upravljački mikrokontroler
- definirati sučelje robota prema korisniku
- procijeniti ukupnu cijenu izrade robota

Potrebno je navesti korištenu literaturu i ostale izvore informacija te eventualno dobivenu pomoć.

Zadatak zadan:
28. studenog 2019.

Zadatak zadao:
Prof.dr.sc. Mladen Crneković

Datum predaje rada:
1. rok: 21. veljače 2020.
2. rok (izvanredni): 1. srpnja 2020.
3. rok: 17. rujna 2020.

Predviđeni datumi obrane:
1. rok: 24.2. – 28.2.2020.
2. rok (izvanredni): 3.7.2020.
3. rok: 21.9. - 25.9.2020.

Predsjednik Povjerenstva:
Prof. dr. sc. Branko Bauer

SADRŽAJ

POPIS SLIKA II
POPIS TABLICA III
POPIS OZNAKA V
SAŽETAK VI
SUMMARY VII
1. UVOD 1
1.1. Povijesni pregled 1
1.2. Edukacijske robotske ruke 5
2. KINEMATIKA ROBOTA 8
2.1. Osnove kinematike 8
2.2. Kinematski model 10
2.2.1. Direktna kinematika 11
2.2.2. Inverzna kinematika 13
3. CAD MODEL 20
3.1. Sklop 20
3.2. Usporedba CAD-a i stvarnog modela 22
4. PCB UPRAVLJAČKA PLOČICA 27
4.1. Izbor komponenata 27
4.1.1. Servo motori 27
4.1.2. Mikroprocesor 31
4.1.3. Baterije 33
4.1.4. USB komunikacija 33
4.1.5. Modul za SD karticu 34
4.1.6. Ostale komponente 34
5. PROGRAMIRANJE 36
5.1. Opis programa 36
6. KUČIŠTE 39
7. PROCJENA TROŠKOVA 41
8. ZAVRŠNI SKLOP 42
9. ZAKLJUČAK 43
LITERATURA 44
PRILOZI 45

POPIS SLIKA

Slika 1. Da Vincijev model pokretnog lava ... 1
Slika 2. Robot B-9 iz starih filmova ... 2
Slika 3. Terminator iz kultnog filma ... 2
Slika 4. Unimate prvi industrijski robot ... 3
Slika 5. Robotske ruke u autoindustriji ... 4
Slika 6. Edukacijska robotska ruka 1 ... 5
Slika 7. Edukacijska robotska ruka 2 ... 6
Slika 8. Edukacijska robotska ruka 3 ... 6
Slika 9. Stupnjevi slobode tijela u kartezijevu koordinatnom sustavu ... 8
Slika 10. Dva osnovna zgloba kod robotskih ruka: rotacijski i translacijski ... 9
Slika 11. Kinematska struktura robotske ruke za početni položaj q_i=0 ... 11
Slika 12. Kinematska struktura u x-z ravnini ... 17
Slika 13. Položaj lakta ... 18
Slika 14. Trokut W-T-R ... 18
Slika 15. Sklop robotske ruke ... 20
Slika 16. Servo JX PDI-6221mg i Tower pro MG995 dimenzije ... 21
Slika 17. Servo SG90 dimenzije ... 21
Slika 18. Usporedba za početni položaj ... 22
Slika 19. Usporedba 2 ... 23
Slika 20. Podaci o mehaničim karakteristikama iz Catie ... 23
Slika 21. Izvagana masa robota ... 24
Slika 22. Usporedba u najnepovoljnijem položaju ... 25
Slika 23. Regulacijski krug servo motora ... 27
Slika 24. Presjek hobističkog servo motora ... 28
Slika 25. PMW signal kod servo motora ... 29
Slika 26. JX PDI-6221mg servo motor ... 29
Slika 27. Karakteristike JX PDI-6221mg servo motora ... 30
Slika 28. Tower pro MG995 servo motor ... 30
Slika 29. Karakteristike Tower pro MG995 servo motora ... 30
Slika 30. Tower pro SG90 servo motor ... 31
Slika 31. Karakteristike Tower pro SG90 servo motora ... 31
Slika 32. ATMEGA 328p-pu pinovi ... 32
Slika 33. ATMEGA 328p-pu karakteristike ... 32
Slika 34. Li-ion 18650 baterije karakteristike ... 33
Slika 35. FT232-rl karakteristike ... 33
Slika 36. Modul za SD karticu karakteristike ... 34
Slika 37. 3D model PCB pločice ... 35
Slika 38. Izrađena PCB pločica ... 35
Slika 39. Inicijaliziranje varijabli ... 36
Slika 40. Definiranje pinova motora ... 37
Slika 41. Definiranje modova robota ... 37
Slika 42. Snimanje i spremanje podataka na SD karticu ... 38
Slika 43. Učitavanje podataka sa SD kartice ... 38
Slika 44. 3D model kučišta ... 39
Slika 45. Zglobovi robota po imenima ... 40
Slika 46. Završni sklop ... 42

POPIS TABLICA

Tablica 1. Procjena troškova 41

POPIS OZNAKA

A	- matrica homogenih transformacija
p	- vektor položaja
i , **j** , **k**	- jedinični vektori u pravcima karterijevih koordinatnih osi
T	- matrica transformacija
q	- vektor unutarnjih koordinata
φ	- Euleorov kut, označava nagib ili posrtanje
ϑ	- Euleorov kut, označava skretanje
α ,β ,γ	- pomoćni kutevi
l	- pomoćna dužina
L	- udaljenost između zglobova robota
l	- krak na kojem djeluje maximalni moment
M_m	- moment koji motor mora savladati bez tereta
M_t	- korisni moment koji preostaje

SAŽETAK

Ideja ovog rada je da se uz izradu hobističke robotske ruke nauče osnovni koncepti funkcioniranja pravih industrijskih robota. Rad se sastoji od više područja: konstruiranja, izrade PCB upravljačke pločice i programiranja. Time je objedinjeno više područja s kojima smo se do sad susreli na fakultetu: strojarstvo, elektronika i programiranje.

U programu „Catia“ modeliran je robot po dijelovima koji su izrađeni 3D printanjem i sklopljeni u cijelinu pomoću vijaka i motora.

U programu „Altium Designer“ izrađena je PCB pločica preko koje se robot upravlja. Odabrani su servo motori, senzori položaja, mikrokontroler, konektori i ostale potrebne komponente.

U programu „Solid Works“ izrađeno je kučište za 3D printanje u koje je postavljena upravljačka pločica preko koje se robotom upravlja i ona je korisničko sučelje između robota i korisnika.

U programskom sučelju „Arduino IDE“ robot je programiran tako da se može upravljati ručno, ali i da može pamtiti i oponašati kretnje koje su mu prethodno zadane.

SUMMARY

SUMMARY

The idea of this project is to learn basic principles of how industrial robots operate and function throughout making hobbystic robotic arm. This final task consists of areas such as constructing, making PCB control panel and programming. Within there have been combined several areas learned throughout the past 3 years of studying: mechanical engineering, electronics, and programming.

In programme package ″Catia″ , the robotic arm has been made part by part and then 3D printed, parts are connected with screws and motors.With help of ″Altium Designer″ PCB control panel has been made. Containing servo motors, sensors, microcontrollers, connectors, and other components.Casing has been made using the ″ Solid Works″ program and 3D printer. With all the components and PCB inside, the casing is interference between robot and the user.

Robotic arm is programmed in ″Arduino IDE″ . It can be controled manually but it is also capable to remember and repeat previously assigned movements.

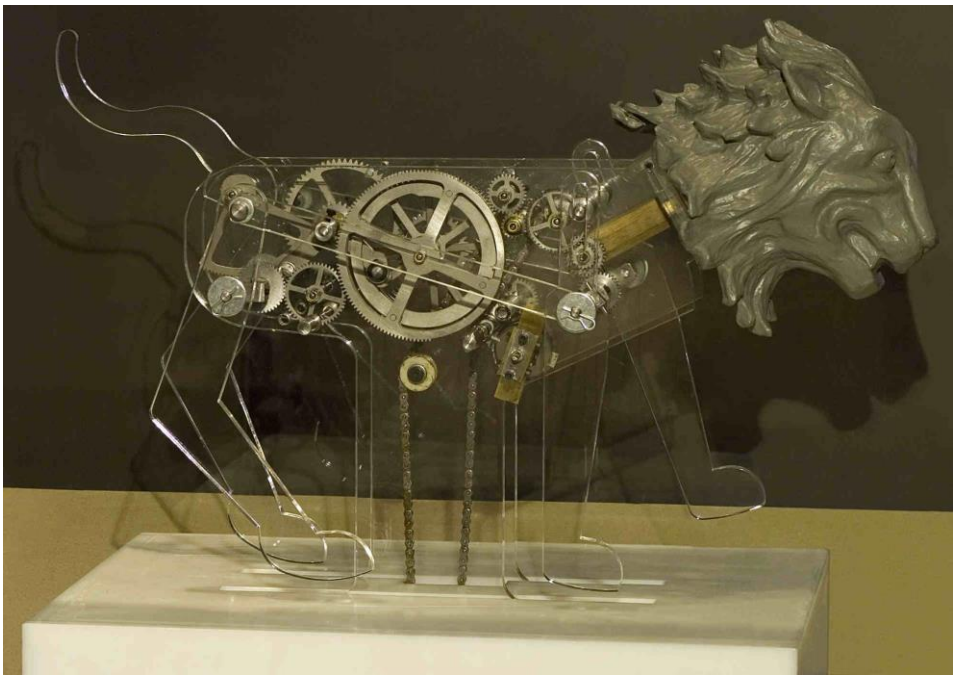
1. UVOD

Robotske ruke temelj su današnje moderne industrije. Sve češće zamjenjuju ljude na teškim, zamornim i dosadnim fizičkim poslovima. Mogu biti hidrauličke, električne ili kombinacija između te dvije. Spadaju pod statičke robote, a mogu biti manipulatori(za posluživanje strojeva i opreme), operacijski roboti, roboti za montažu, za mjerenje i kontrolu itd.

Glavne prednosti u industriji su velika preciznost, brzina, točnost, nosivost i ponovljivost. Osobito do izražaja dolaze u današnje doba, ključan su dio Industrije 4.0. gdje se teži automatizaciji i povezivanju svih dijelova industrije na globalnoj razini.

1.1. Povijesni pregled

Ideja robota postoji još od davnih vremena. Jedna od prvih ideja koja se može povezati s današnjim robotima potječe još od Leonarda Da Vincija i njegovog modela pokretnog stroja u obliku lava.

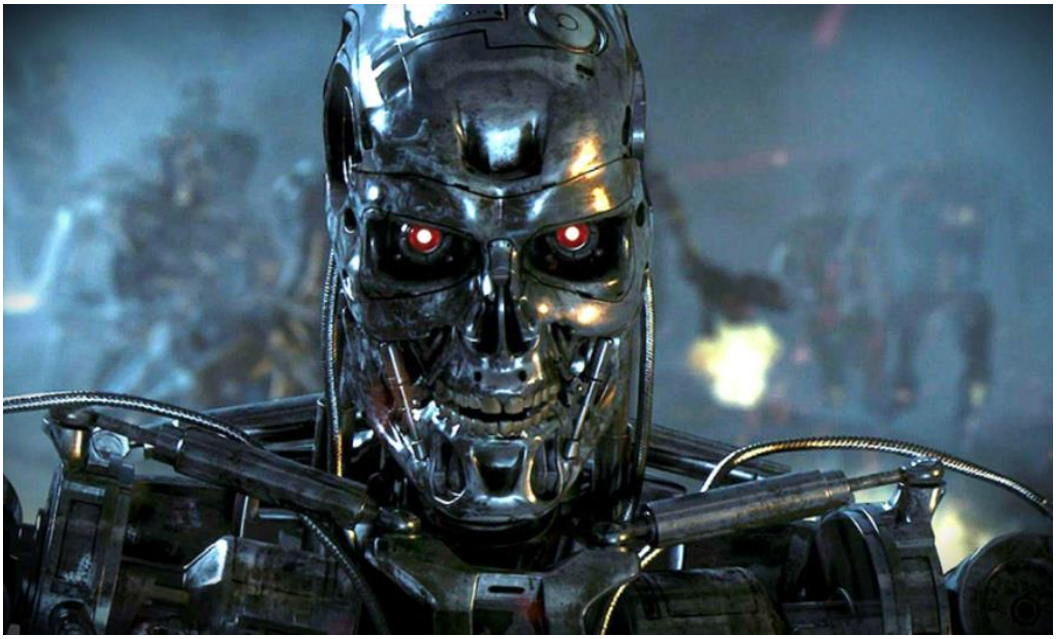


Slika 1. Da Vincijev model pokretnog lava

Često su se pojavljivali i u raznim filmovima. Ideja robota kao kućnog pomoćnika, prijatelja, pa sve do modernijih verzija kao što su transformeri ili roboti kao strojevi za ubijanje kao npr. u kultnom filmu Terminator.

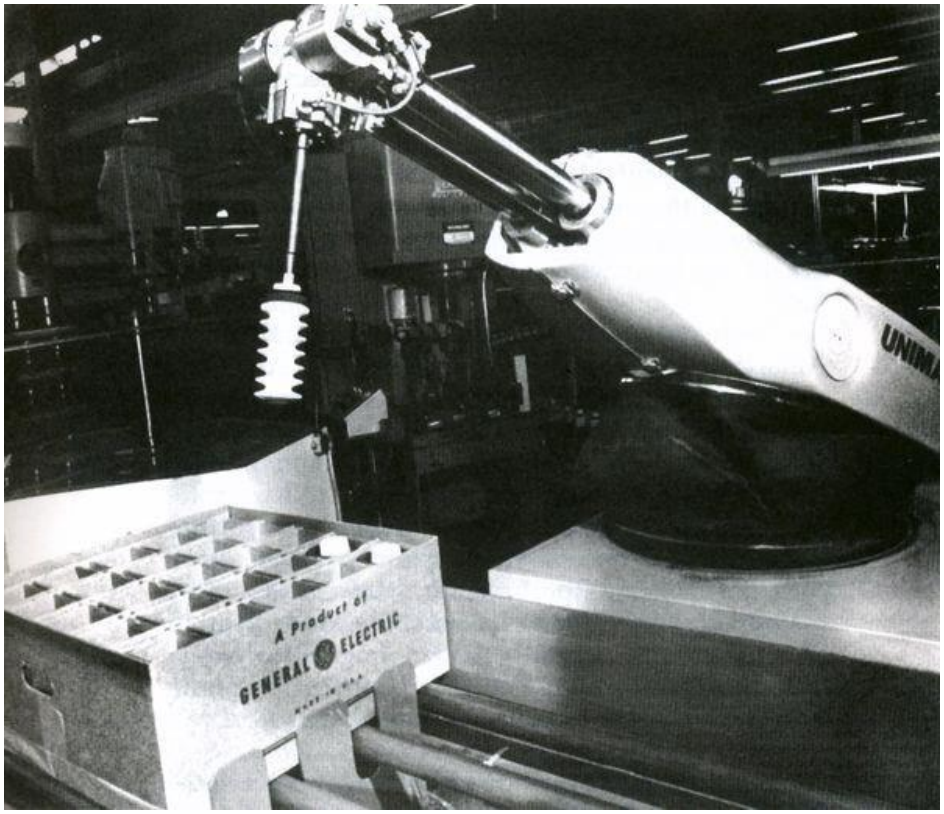


Slika 2. Robot B-9 iz starih filmova



Slika 3. Terminator iz kultnog filma

Kao što se i vidi iz prethodnih primjera ljudi su uvijek zamišljali robote kao nešto nalik sebi sa sličnim funkcijama, izgledom i kretnjama. Što se tiče stvarne industrije tu su stvari ipak bile malo drugačije, takvi humanoidi roboti kakve smo zamišljali su ipak i dan danas veliki izazov za inženjere. Ipak izumitelj Georga C. Devol je 1961.godine proizveo prvi industrijski robot i to manipulator imena Unimate koji je ušao u industrijsku proizvodnju kao prava revolucija. Instaliran je u tvornici General Motorsa, a služio je za jednostavnije zadatke kao što su programirano prenošenje stvari, neke osnovne mehaničke operacije poput zavarivanja i slično.



Slika 4. Unimate prvi industrijski robot

On je bio snažna okosnica i ohrabrenje za inženjere da roboti zapravo mogu donijeti puno prednosti u industrijsku proizvodnju i od tada do danas počeo je snažan razvoj takve vrste robota.

Kad pogledamo njegov izgled zapravo bi mogli reći da je on prva robotska ruka ikad napravljena. Od tada do danas robotske ruke su najrazvijenija grana robotike, ne postoji niti jedna ozbiljnija tvornica u kojoj robotske ruke nisu zadužene barem za neke rutinske, dosadne i ljudima prenaporne poslove.

Joseph Engelberg koji je prvi započeo s prodajom robota smatra se ocem robotike. Iako njegov proizvod nije naišao na preveliki interes u njegovoj domovini jest u Japanu. Gdje je 1970. u funkciju ušao prvi pravi industrijski robot koji je radio u Nissanovoj tvornici kao zavarivač. Od tada pa sve do danas Japan je bio i ostao vodeća sila u području robotike te danas „zapošljavaju“ više od polovice svjetskih robota.

Snažan razvoj robotike usko je povezan s autoindustrijom gdje danas imamo tvornice u kojima 90% poslova rade samostalno roboti i računala. S velikim napretkom elektronike, računala i programiranja i roboti su doživjeli snažni uzlet i ušli u svakodnevnu primjeru u globalnoj svjetskoj industriji.



Slika 5. Robotske ruke u autoindustriji

Povijesni razvoj robota možemo podijeliti u nekoliko etapa:

a) Prva generacija – mogućnost automatskog ponavljanja zadanog pokreta, upravljački sustav lako se prilagođava ručnim operacijama, primjenjivi u tvornicama za jednostavne operacije kao što su zavarivanje, prešanje itd.

b) Druga generacija – uvođenje senzora, roboti dobivaju „osjećaj“ prostora u kojem djeluju, njihovo snalaženje i postupci u radnom prostoru su programirani, koristili su se na pokretnim trakama, montažnim vrpcama, bojanju itd.

c) Treća generacija – uvođenje umjetne inteligencije, opremanje računalima, usavršeni programi, sposobnost raspoznavanja okoline, analiza svojih postupaka i učenje iz pogrešaka, prilagođavanje uvjetima rada da bi se povećala njihova efikasnost

1.2. Edukacijske robotske ruke

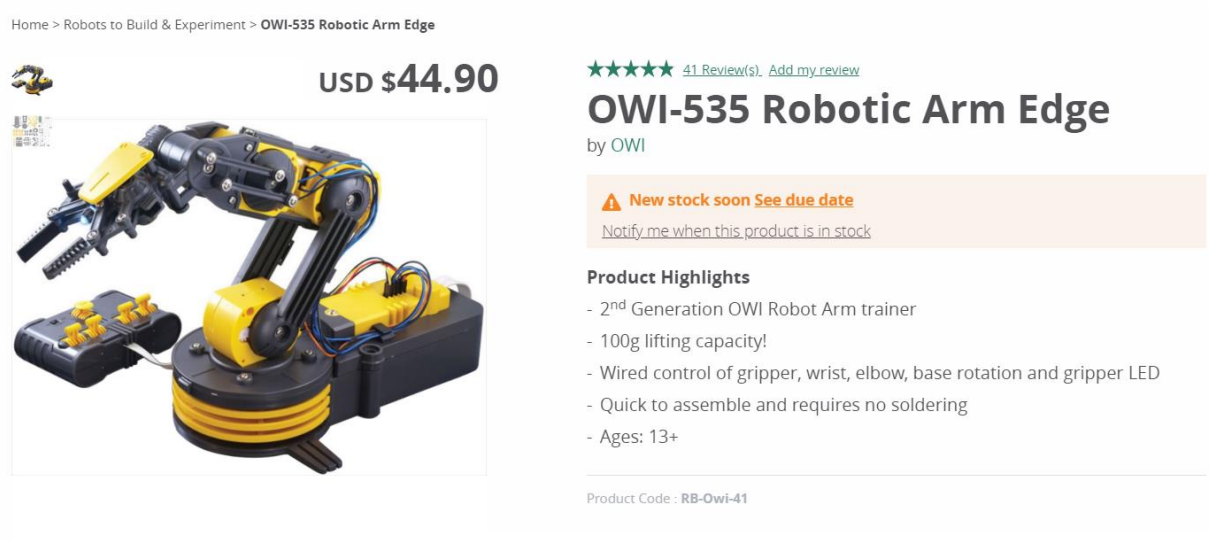
Kao što je već ranije napomenuto od svih vrsta robota najveću i najuspješniju primjeru u industriji ali i širi imale su i još uvijek imaju robotske ruke i razni manipulatori.

Današnja robotika je vrlo složeni mehatronički sustav koji zahtjeva širok spektar vještina od elektronike, mehanike, pneumatike, programiranja itd. Pa je u današnje vrijeme s obzirom da živimo u doba interneta, globalne povezanosti i umreženosti, jeftine i lako dostupne elektronike i raznih komponenata razvijena i vrlo široka ponuda lako dostupnih edukacijskih, hobističkih robota i manipulatora koji služe da bi se na njima učilo i upoznalo s osnovama funkcioniranja takvih složenih sustava.

Prije samog opisa izrade robotske ruke predstavljeno je istraživanje tržišta edukacijskih robotskih ruku da se vidi koliko okvirno takva tehnologija za učenje košta i kolike su njezine mogućnosti.

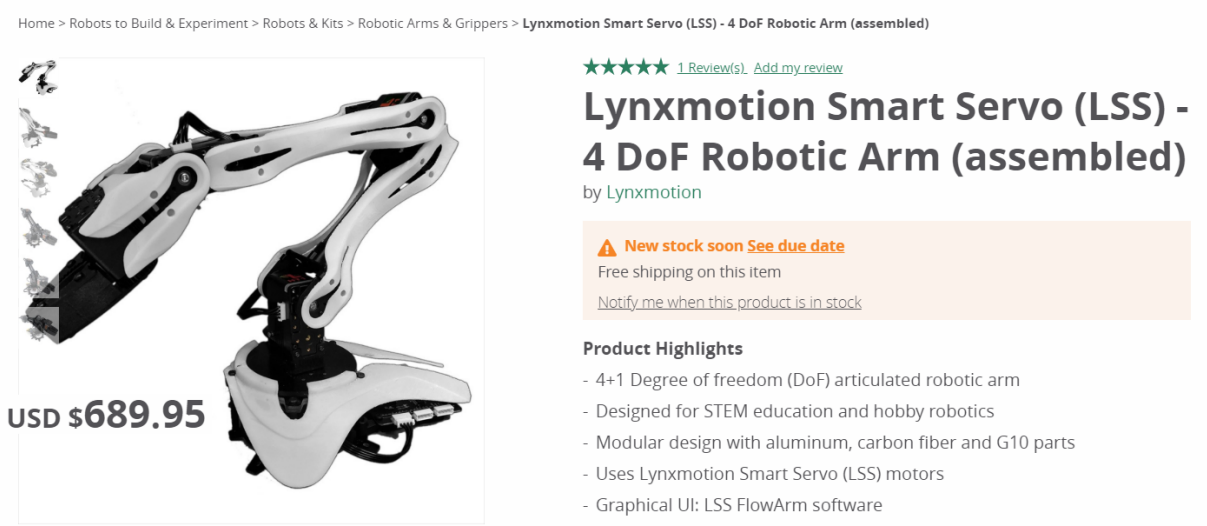
Kao izvor korištena je poznata stranica za hobističku robotiku imena RobotShop.

Prvi primjer je jednostavna robotska ruka s 4 stupnja slobode gibanja i mogučnošću hvatanja predmeta. Nosivosti 100g. Dolazi u paketu sa svim dijelovima i potrebno ju je samo sklopiti. Cijena joj je oko 300 kuna.



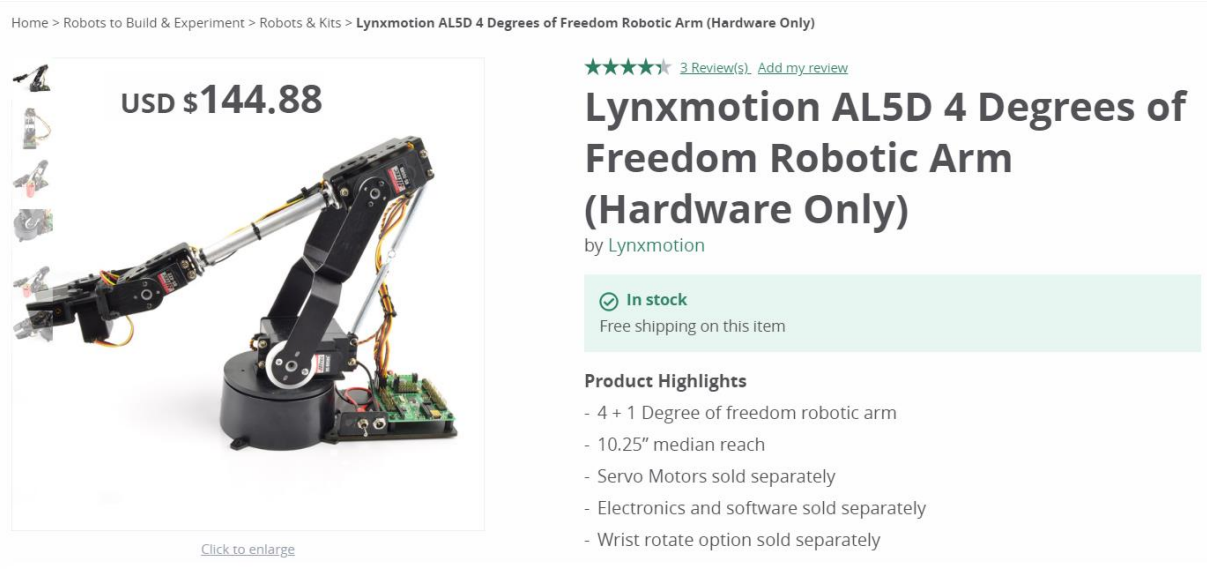
Slika 6. Edukacijska robotska ruka 1 [1]

Kao drugi primjer uzeta je najskuplja robotska ruka koja je pronađena na toj stranici. Cijena joj je oko 4600 kuna. Dolazi sklopljena, a ima 4 stupnja slobode gibanja plus gripper. Izrađena je iz aluminija i karbonskih vlakana, a za nju postoji i grafički software preko kojeg se može upravljati.



Slika 7. Edukacijska robotska ruka 2 [1]

I kao posljednji primjer izabran je sklop koji je najsličniji sklopu koji je izrađen u sklopu ovog završnog ispita. Također ima mogućnost hvatanja predmeta i 4 stupnja slobode gibanja. A cijena bez motora i elektronike joj je nešto manje od 1000 kn.



Slika 8. Edukacijska robotska ruka 3 [1]

Kao što se iz priloženog vidi danas se za relativno malu cijenu može dobiti gotov proizvod na kojem se može vježbati, učiti i educirati. Pomoću interneta tehnologija je danas dostupna gotovo svima koji za nju iskažu interes. Najjeftiniji robot dođe samo 300 kuna što stvarno nije velika cijena za nešto što je prije 70,80 godina koštalo milijune dolara i bilo relativno nedostupno i neshvatljivo i to ne samo studentima nego i većini inženjera tog doba.

S obzirom da tehnologija užurbano napreduje iz dana u dan inženjeri u ovom području su sve potrebniji i zato su ovakve jeftine edukacijske naprave odlična stvar za uvođenje u složen svijet moderne robotike.

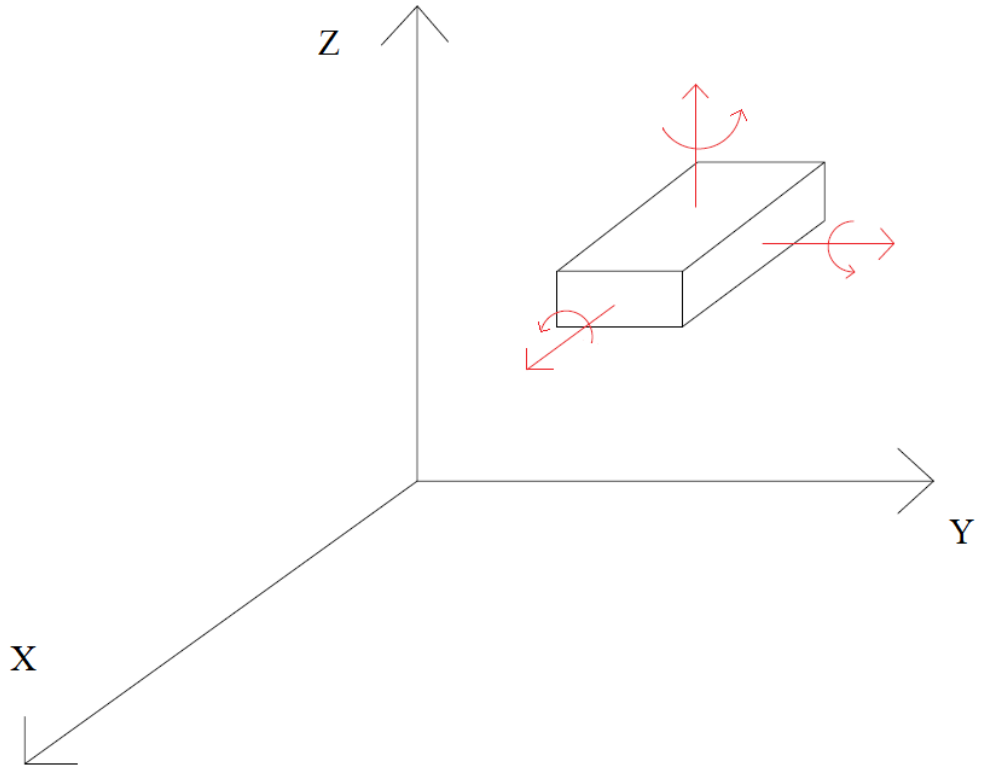
2. KINEMATIKA ROBOTA

2.1. Osnove kinematike

U kartezijevom koordinatnom sustavu tijelo ima 6 stupnjeva slobode gibanja:

- 3 translacije, svaka po jednoj od koordinatnih osi x,y,z
- 3 rotacije, svaka rotacije oko jedne od osi x,y,z

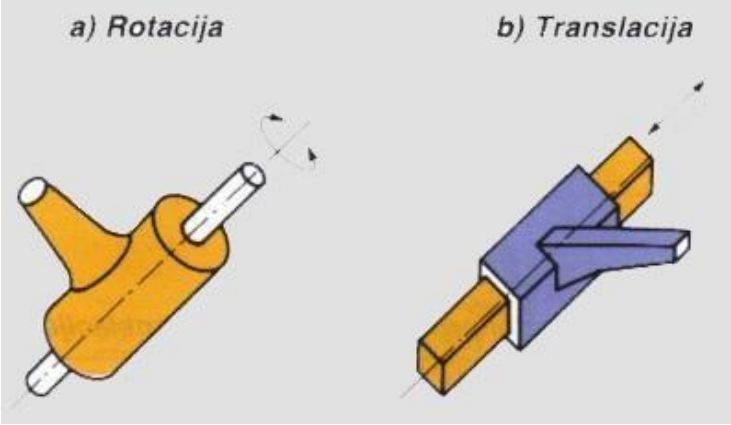
Kad to primjenimo na robota zaključak je da se s 3 stupnjeva slobode robot može pozicionirati u bilo kojoj točki u svom radnom prostoru, ali samo ako ima i dodatna 3 stupnja slobode može u te točke doći iz bilo koje orijentacije. Pa da bi robot u svom radnom prostoru mogao doći u bilo koju točku mora imati minimalno 3 stupnja slobode gibanja, a da bi se mogao i proizvoljno orijentirati treba nam 6 stupnjeva slobode gibanja koliko industrijski roboti i najčešće imaju.



Slika 9. Stupnjevi slobode tijela u kartezijevom koordinatnom sustavu

Vezanjem jednog pokretnog tijela na drugo nastaje zglob. Zglobovi nam ograničavaju gibanje to jest oduzimaju stupnjeve slobode. Neki od osnovnih zglobova su[2]:

- Translacijski zglob – pomak po jednoj od osi
- Rotacijski zglob – zakret oko jedne osi
- Vijčani – zakretanje duž navoja i pomak po osi
- Valjkasti – zakretanje i pomak po osi (valjak unutar valjka)
- Sferni – 3 neovisna zakreta



Slika 10. Dva osnovna zgloba kod robotskih ruka, rotacijski i translacijski[2]

U ovom robotu korišteni su rotacijski zglobovi, točnije 4 rotacijska zgloba koja čine robotsku ruku s 4 stupnja slobode i mogućnošću hvatanja predmeta.

Da bi izveli kinematski model potrebna nam je matrica homogenih transformacija **A** koja nam opisuje položaj i orijentaciju koordinatnog sustava n prema koordinatnom sustavu m.

Izraz za matricu **A** prema [3]:

$$ {}^{m}\mathbf{A}_{n} = \begin{bmatrix} i_x & j_x & k_x & p_x \\ i_y & j_y & k_y & p_y \\ i_z & j_z & k_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.1} $$

Gdje su **i**, **j**, **k** jedinični vektori u pravcima osi x ,y ,z a **p** je vektor položaja.

Općeniti slučajevi matrice **A** koje ćemo kasnije i koristiti za rješavanje kinematskog modela prema izrazu (1.1) i [3]:

- Rotacije oko osi z za kut α

$$ \mathbf{A} = \mathrm{Rot}(\mathrm{z}, \alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2} $$

- Translacija po osi x,y i z

$$\mathbf{A} = \mathrm{Tran(a,b,c)} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.3}$$

- Rotacija oko osi x za kut β

$$\mathbf{A} = \mathrm{Rot(x,\beta)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.4}$$

Primjer za prijelaz između koordinatnih sustava prema [3]:

$$\mathbf{p}_0 = {}^0\mathbf{A}_1\mathbf{p}_1 \tag{1.5}$$

$$\mathbf{p}_1 = \left({}^0\mathbf{A}_1\right)^{-1}\mathbf{p}_0 \tag{1.6}$$

Gdje je ${}^0\mathbf{A}_1$ matrica homogenih transformacija za prijelaz iz prvog u nulti koordinatni sustav, a $\mathbf{p}_0$ i $\mathbf{p}_1$ su vektori položaja neke točke u tim koordinatnim sustavima.

Matrica transformacije **T** nam je umnožak matrica homogenih transformacija ovisno o tome iz kojeg koordinatnog sustava se želimo prebaciti, primjer prema [3]:

$$\mathbf{p}_0 = {}^0\mathbf{A}_1\ {}^1\mathbf{A}_2\,\mathbf{p}_2 \tag{1.7}$$

$${}^0\mathbf{T}_2 = {}^0\mathbf{A}_1\ {}^1\mathbf{A}_2 \tag{1.8}$$

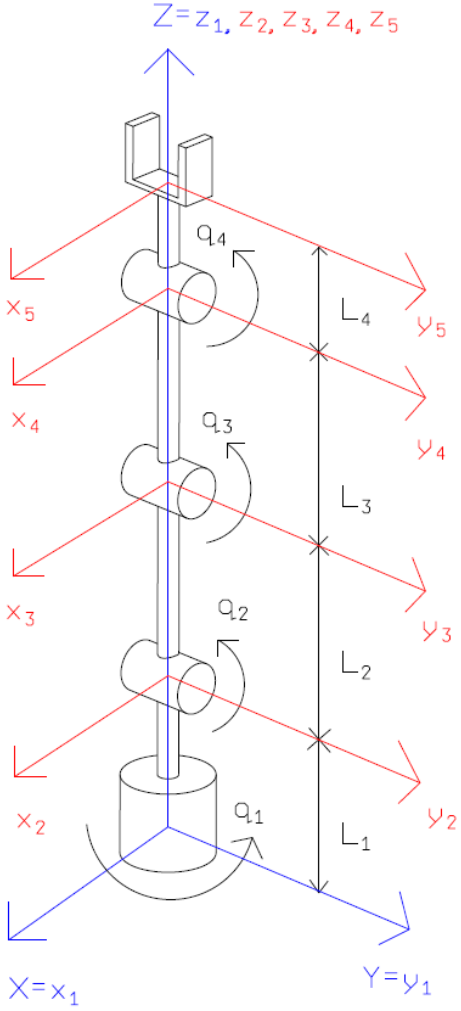
Pa onda za matricu **T** za prijelaz iz n-tog u m-ti koordinatni sustav vrijedi[3]:

$${}^m\mathbf{T}_n = \prod_{i=m+1}^{n} {}^{i-1}\mathbf{A}_i \tag{1.9}$$

2.2. Kinematski model

Edukacijska robotska ruka rađena za ovaj završni rad sastoji se od 5 servo motora ali s obzirom da jedan od njih upravlja hvataljkom to jest gripperom on se ne računa u stupnjeve slobode.

Tako da robot ima 4 stupnjeva slobode gibanja i svi su izvedeni s rotacijskim zglobom.



Slika 11. Kinematska struktura robotske ruke za početni položaj $q_i = 0$

Iz slike je vidljivo da se radi o RRRR strukturi robota. Svi stupnjevi slobode gibanja q_i na slici se nalaze u početnom položaju koji iznosi 0. S obzirom da servo motori koji su korišteni imaju opseg gibanja 180 stupnjeva to znači da svaki stupanj slobode može poprimati vrijednosti od -90 do 90.

2.2.1 Direktni kinematički problem

U direktnom kinematičkom problemu polazimo od vektora unutarnjih koordinata koji je za slučaj ove robotske ruke:

$$\mathbf{q}^T = [q_1 \;\; q_2 \;\; q_3 \;\; q_4] \tag{1.10}$$

A cilj nam je dobiti vektor položaja **p** :

$$\mathbf{p}^T = [p_x \;\; p_y \;\; p_z \;\; \varphi] \tag{1.11}$$

Gdje nam je φ jedan od Euleorovih kuteva koji nam predstavlja posrtanje ili nagib i u slučaju našeg robota $\varphi = q_2+q_3+q_4$ ..

Od ostalih Eulerovih kutova imamo još i skretanje ϑ a ono nam je jednako $\vartheta = q_1$.

Prema slici 11 i izrazima (1.2), (1.3) i (1.4) izvodimo matrice homogenih transformacija za prijelaze između lokalnih koordinatnih sustava.

$$\mathbf{A}_1 = \mathrm{Rot}(\mathrm{z}, q_1) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.12}$$

$$\mathbf{A}_2 = \mathrm{Tran}(0,0,\mathrm{L}_1)\ \mathrm{Rot}(\mathrm{y}, q_2) = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.13}$$

$$\mathbf{A}_3 = \mathrm{Tran}(0,0,L_2)\ \mathrm{Rot}(\mathrm{y}, q_3) = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.14}$$

$$\mathbf{A}_4 = \mathrm{Tran}(0,0,\mathrm{L}_3)\ \mathrm{Rot}(\mathrm{y}, q_4) = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ 0 & 1 & 0 & 0 \\ -s_4 & 0 & c_4 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.15}$$

$$\mathbf{A}_5 = \mathrm{Tran}(0,0,\mathrm{L}_4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.16}$$

Gdje je uvedena supstitucija radi lakšeg zapisivanja:

$$\begin{aligned} s_i &= \sin q_i \\ c_i &= \cos q_i \end{aligned} \tag{1.17}$$

Matrica transformacije iz petog u nulti koordinatni sustav jednaka je:

$$ {}^{0}\mathbf{T}_5 = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5 \tag{1.18} $$

Dobivamo izraz za matricu **T** :

$$ {}^{0}\mathbf{T}_5 = \begin{bmatrix} c_{234}c_1 & -s_1 & s_{234}c_1 & c_1(L_3s_{23} + L_2s_2 + L_4s_{234}) \\ c_{234}s_1 & c_1 & s_{234}s_1 & s_1(L_3s_{23} + L_2s_2 + L_4s_{234}) \\ -s_{234} & 0 & c_{234} & L_1 + L_3c_{23} + L_2c_2 + L_4c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.19} $$

Gdje je:

$$ c_{234} = \cos(q_2 + q_3 + q_4)\ ,\ s_{234} = \sin(q_2 + q_3 + q_4) $$
$$ c_{23} = \cos(q_2 + q_3)\ ,\ s_{23} = \sin(q_2 + q_3) \tag{1.20} $$

Iz izraza (1.20) kao rješenje direktnog kinematičkog problema možemo dobiti položaj točke hvatišta robota u baznom koordinatnom sustavu ovisno o zakretu pojedinog motora to jest zgloba.

Kao provjeru u izraz (1.20) unosimo početne vrijednosti stupnjeva slobode q_1 ,q_2, q_3 ,q_4 = 0 te s obzirom da je cos(0) = 1 ,a sin(0) = 0 dobivamo matricu **T** koja odgovara početnom položaju robota.

$$ {}^{0}\mathbf{T}_5(q_i = 0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 + L_2 + L_3 + L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.21} $$

2.2.2 Inverzni kinematički problem

Kod inverznog kinematičkog problema radimo suprotan postupak u odnosu na direktni kinematički problem. Iz poznatog vektora položaja **p** cilj nam je dobiti vektor unutarnjih koordinata **q** . Što zapravo znači da tražimo koliko se koji motor treba zakrenuti da bi se naš robot našao u određenoj zadanoj poziciji.

Korišteni su izrazi iz [2].

$$ ({}^{o}\mathbf{A}_1)^{-1} \cdot {}^{0}\mathbf{T}_5 = {}^{1}\mathbf{T}_5 \tag{1.22} $$

$$\begin{bmatrix} f_{11}(n) & f_{11}(o) & f_{11}(n) & f_{11}(p) \\ f_{12}(n) & f_{12}(o) & f_{12}(n) & f_{12}(p) \\ f_{13}(n) & f_{13}(o) & f_{13}(n) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{A_2 A_3 A_4 A_5} \tag{1.23}$$

Gdje je prema[2] :

$$f_{11} = c_1 x + s_1 y$$

$$f_{12} = -s_1 x + c_1 y$$

$$f_{13} = z \tag{1.24}$$

Izjednačavanjem izraza s lijeve i desne strane na mjestu 2.redka i 4. stupca dobivamo:

$$-s_1 p_x + c_1 p_y = 0 \tag{1.25}$$

Uz pomoć izraza prema[2]:

$$p_x = r \cdot \cos\eta \ , \ p_y = r \cdot \sin\eta$$

$$r = \sqrt{p_x^2 + p_y^2} \ , \eta = arc\tan\frac{p_y}{p_x} \tag{1.26}$$

Dobivamo izraz za koordinatu q_1 :

$$q_1 = arc\tan\frac{p_y}{p_x} \tag{1.27}$$

Izjednačavanjem izraza s lijeve i desne strane na mjestu 1.redka i 4. stupca dobivamo:

$$s_2 = \frac{c_1 p_x + s_1 p_y - L_3 s_{23} - L_4 s_{234}}{L_2} \tag{1.28}$$

Izjednačavanjem 3.redka i 4.stupca s lijeve i desne strane dobivamo jednadžbu:

$$p_z = L_1 + L_3 c_{23} + L_2 c_2 + L_4 c_{234} \tag{1.29}$$

Iz izraza (1.29) izlučujemo c_2 :

$$c_2 = \frac{p_z - L_1 - L_3 c_{23} - L_4 c_{234}}{L_2} \tag{1.30}$$

Potom raspišemo izraz za tan(q_2) koji je jednak $\frac{s_2}{c_2}$ i potom pomoću arc tan funkcije dobivamo rješenje za drugu unutarnju koordinatu q_2.

$$q_2 = arc\tan\left(\frac{c_1 p_x + s_1 p_y - L_3 s_{23} - L_4 s_{234}}{p_z - L_1 - L_3 c_{23} - L_4 c_{234}}\right)$$

(1.31)

Daljnjim raspisivanjem dobivamo dodatne jednadžbe iz kojih tražimo preostale unutarnje koordinate:

$$({}^{o}\mathbf{A}_2)^{-1} \cdot ({}^{o}\mathbf{A}_1)^{-1} \cdot {}^{0}\mathbf{T}_5 = {}^{2}\mathbf{T}_5 \tag{1.32}$$

$$\begin{bmatrix} f_{21}(n) & f_{21}(o) & f_{21}(n) & f_{21}(p) \\ f_{22}(n) & f_{22}(o) & f_{22}(n) & f_{22}(p) \\ f_{23}(n) & f_{23}(o) & f_{23}(n) & f_{23}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{A}_3\mathbf{A}_4\mathbf{A}_5 \tag{1.33}$$

Gdje su opet prema [2]:

$$f_{21} = c_2(c_1 p_x + s_1 p_y) - s_2 p_z + s_2 L_1$$

$$f_{22} = -s_1 p_x + c_1 p_y$$

$$f_{23} = s_2(c_1 p_x + s_1 p_y) - c_2 p_z - c_2 L_1 \tag{1.34}$$

Izjednačavanjem izraza u 1.redku i 4.stupcu, te u 3.redku i 4 stupcu s lijeve i desne strane dobivamo:

$$c_2(c_1 p_x + s_1 p_y) - s_2 p_z + s_2 L_1 = L_4 s_{34} + L_3 s_3$$

$$s_2(c_1 p_x + s_1 p_y) + c_2 p_z - c_2 L_1 = L_2 + L_4 c_{34} + L_3 c_3 \tag{1.35}$$

Ponovo ponavljamo prijašnji postupak te iz izraza (1.36) izlučujemo c_3 i s_3.

$$s_3 = \frac{c_2(c_1 p_x + s_1 p_y) - s_2 p_z + s_2 L_1 - L_4 s_{34}}{L_3}$$

$$c_3 = \frac{s_2(c_1 p_x + s_1 p_y) + c_2 p_z - c_2 L_1 - L_2 - L_4 c_{34}}{L_3}$$

(1.36)

Raspisujemo izraz za tan(q_3) i pomoću arc tan funkcije dobivamo rješenje za treću unutarnju koordinatu:

$$q_3 = arc\tan\left(\frac{c_2\left(c_1p_x + s_1p_y\right) - s_2p_z + s_2L_1 - L_4s_{34}}{s_2\left(c_1p_x + s_1p_y\right) + c_2p_z - c_2L_1 - L_2 - L_4c_{34}}\right) \tag{1.37}$$

Posljednju unutarnju koordinatu q_4 dobivamo iz izraza za nagib φ.

$$\varphi = q_{234} = q_2 + q_3 + q_4 \tag{1.38}$$

Pa je onda:

$$q_4 = \varphi - q_2 - q_3 \tag{1.39}$$

Izrazi za unutarnje koordinate koje nam pokazuju koliko za neki zadani vektor položaja **p** moramo zakrenuti svaki od naših motora.

$$q_1 = \eta$$

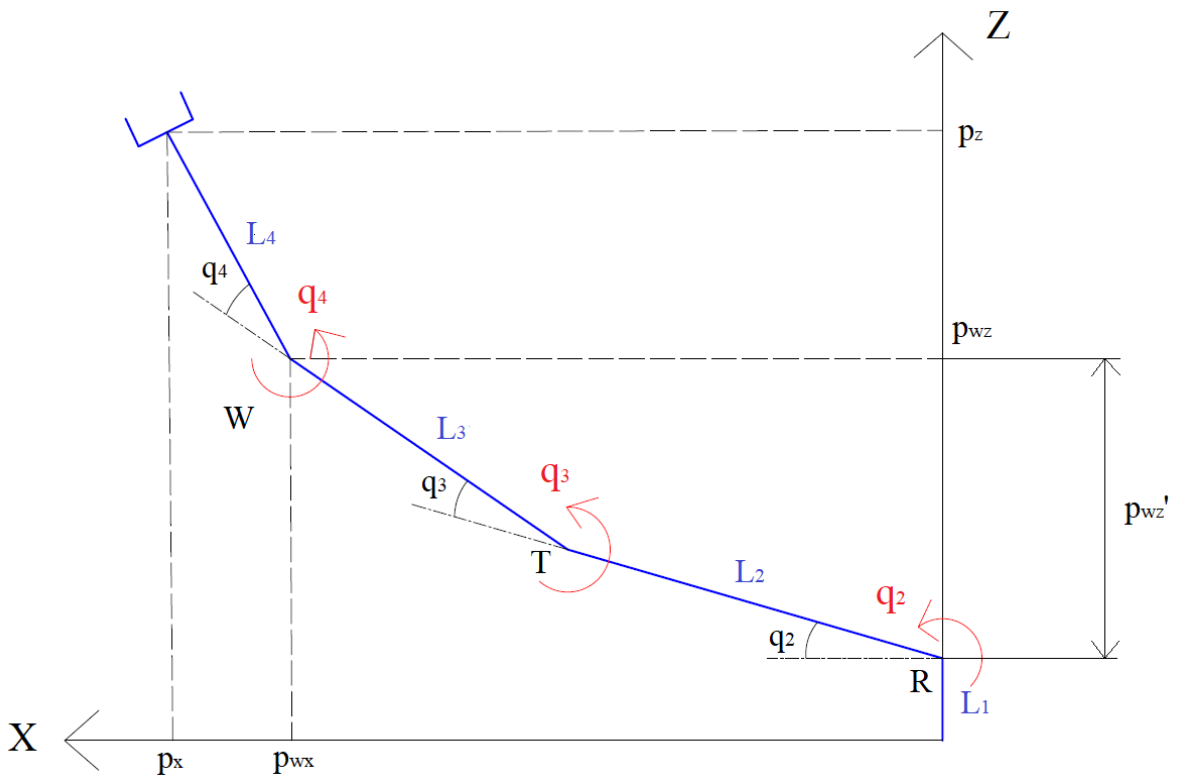
$$q_2 = arc\tan\left(\frac{c_1p_x + s_1p_y - L_3s_{23} - L_4s_{234}}{p_z - L_1 - L_3c_{23} - L_4c_{234}}\right)$$

$$q_3 = arc\tan\left(\frac{c_2\left(c_1p_x + s_1p_y\right) - s_2p_z + s_2L_1 - L_4s_{34}}{s_2\left(c_1p_x + s_1p_y\right) + c_2p_z - c_2L_1 - L_2 - L_4c_{34}}\right)$$

$$q_4 = \varphi - q_2 - q_3 \tag{1.40}$$

Iz izraza (1.40) je vidljivo da smo dobili nelinearne jednadžbe koje se analitički ne mogu rješiti jer u izrazu za q_2 imamo članove s_{23} i c_{23} koji ovise o koordinatama q_2 i q_3 , a u izrazu za koordinatu q_3 imamo članove s_{34} i c_{34} koji ovise o koordinatama q_3 i q_4. S obzirom da imamo 3 paralelna zgloba q_2 , q_3 i q_4 koji svi rotiraju oko osi y, navedeni postupak nam ne donosi zadovoljavajuća analitička rješenja.

Stoga rješenja za te koordinate nalazimo iz same geometrije robota u x-z ravnini.



Slika 12. Kinematska struktura u x-z ravnini

Prema slici 12. najprije izračunavamo koordinatu q_3 iz geometrije vrijede sljedeći izrazi :

$$p_{wx} = p_x - L_4 c_{234} = L_2 c_2 + L_3 c_{23}$$
$$p'_{wz} = p_z - L_4 s_{234} - L_1 = L_2 s_2 + L_3 s_{23} \tag{1.41}$$

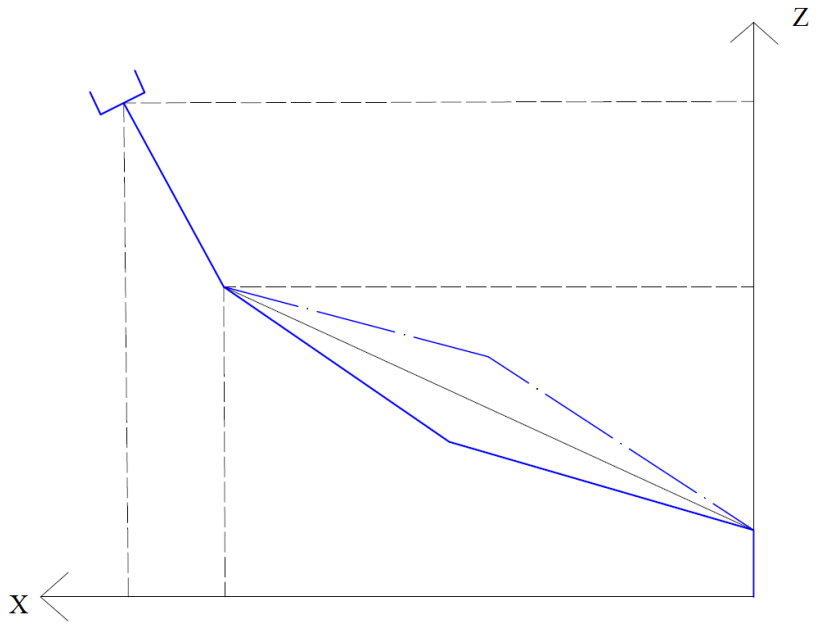
Izraze (1.41) prvo kvadriramo te potom sumiramo i dobivamo izraz iz kojeg izlučujemo c_3 :

$$c_3 = \frac{{p_{wx}}^2 + {p_{wz}}'^2 - L_2^2 - L_3^2}{2L_2L_3} \tag{1.42}$$

Potom iz odnosa sinusa i kosinusa određujemo s_3 :

$$s_3 = \pm\sqrt{1 - c_3^2} \tag{1.43}$$

Gdje nam pozitivan predznak označava gornji položaj lakta robota, a negativan donji položaj prema slici 13.

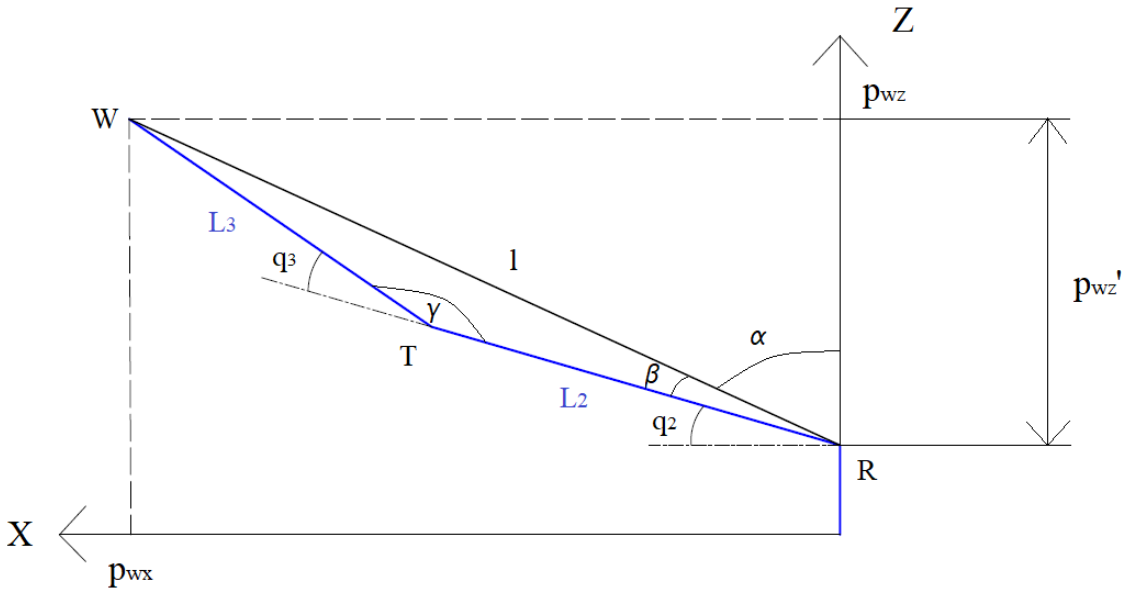


Slika 13. Položaj lakta

Prema tome nam je koordinata q_3 :

$$q_3 = \arctan\left(\frac{s_3}{c_3}\right) \tag{1.44}$$

Koordinatu q_2 izračunavamo iz trokuta W-T-R za konfiguraciju u kojoj je lakat robota u donjem položaju primjenom kosinusovog i sinusovog poučka, te pitagore uz dodavanje pomoćnih kuteva α ,β ,γ .



Slika 14. Trokut W-T-R

Primjenom kosinusovog poučka prema slici 13. dobivamo kut γ :

$$p_{wx}{}^2 + p_{wz}{}'^2 = L_2^2 + L_3^2 - 2L_2L_3\cos\gamma \tag{1.45}$$

Iz čega slijedi da je:

$$\gamma = \arccos\left(\frac{L_2^2 + L_3^2 - l^2}{2L_2L_3}\right) \tag{1.46}$$

Pri čemu je:

$$l = p_{wx}{}^2 + p_{wz}{}'^2 \tag{1.47}$$

Nadalje iz sinusovog poučka slijedi da je:

$$\frac{l}{\sin\gamma} = \frac{L_3}{\sin\beta} \tag{1.48}$$

Te slijedi:

$$\beta = \arcsin\left(\frac{L_3}{l}\sin\gamma\right) \tag{1.49}$$

Kut α određen je prema slici 13. sa izrazom:

$$\alpha = \arctan\left(\frac{p_{wx}}{p_{wz}{}'}\right) \tag{1.50}$$

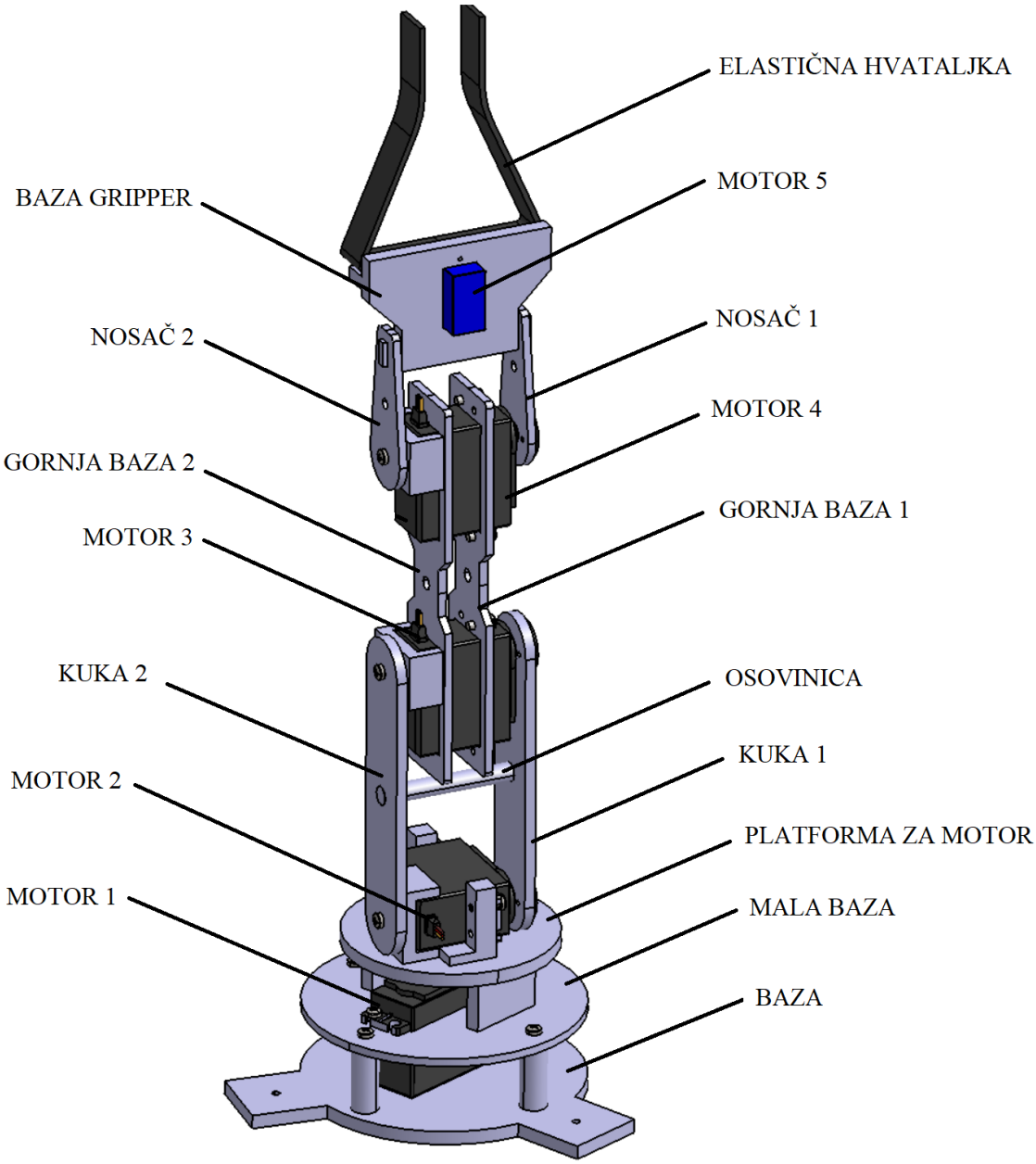
Nakon što smo odredili pomoćne kuteve određujemo koordinatu q_2 :

$$q_2 = 90° - \alpha - \beta \tag{1.51}$$

3. CAD MODEL

3.1. Sklop

CAD model rađen je u programskom paketu Catia. A svi dijelovi izrađeni su za 3D printanje te povezani u sklop pomoću vijaka i motora. Modelirano je na način da su prvo odabrani motori te su prema njihovim modelima i dimnezijama izrađeni dijelovi koji ih pridržavaju a potom su izrađeni i ostali spojni dijelovi.



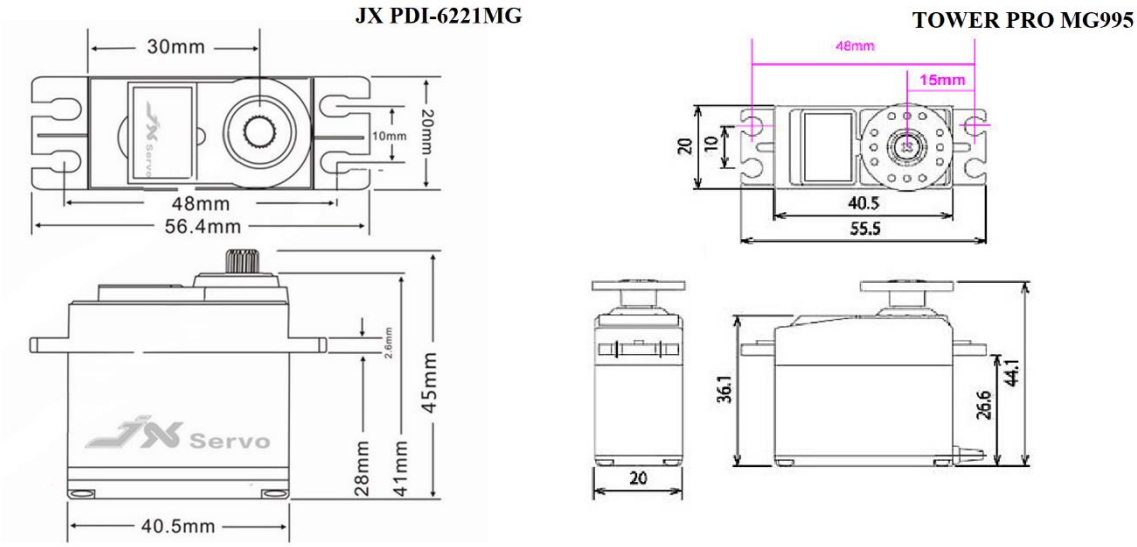
Slika 13. Sklop robotske ruke

Ako ne računamo vijke i spojne diskove koji dolaze s motorima, sklop se sastoji od ukupno 17 dijelova, od kojih su 5 servo motori. Dijelovi su međusobno spojeni pomoću 4 tower pro spojnih diskova i vijaka ISO 7045 od kojih je 8 vijaka M3x10 , 7 vijaka M3x8 i 4 vijka M1.6x10.

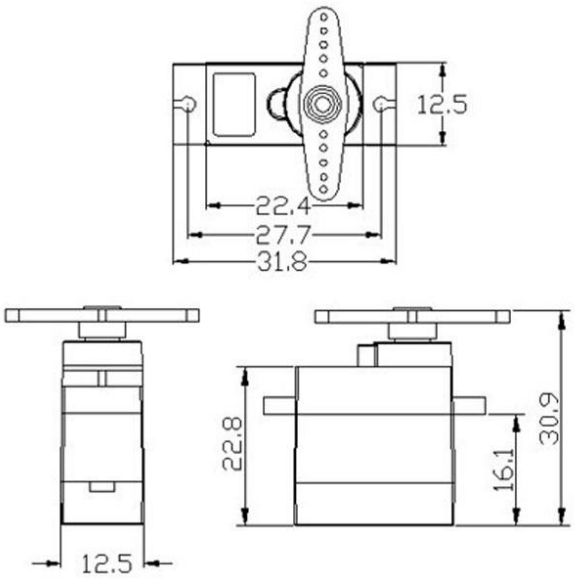
Motori 1 i 4 su Tower pro MG995 servo motori, motori 2 i 3 su JX PDI-6221MG servo motori i motor 5 je Tower pro SG90. O njihovim karakteristikama i specifikacijama će više biti rečeno kasnije u poglavlju 4.

Elastična hvataljka izrađena je ručno, a materijal je akrilna ploča (pleksiglas).

Svi ostali dijelovi izrađeni su 3D printanjem, a kao materijal je korištena PLA plastika.



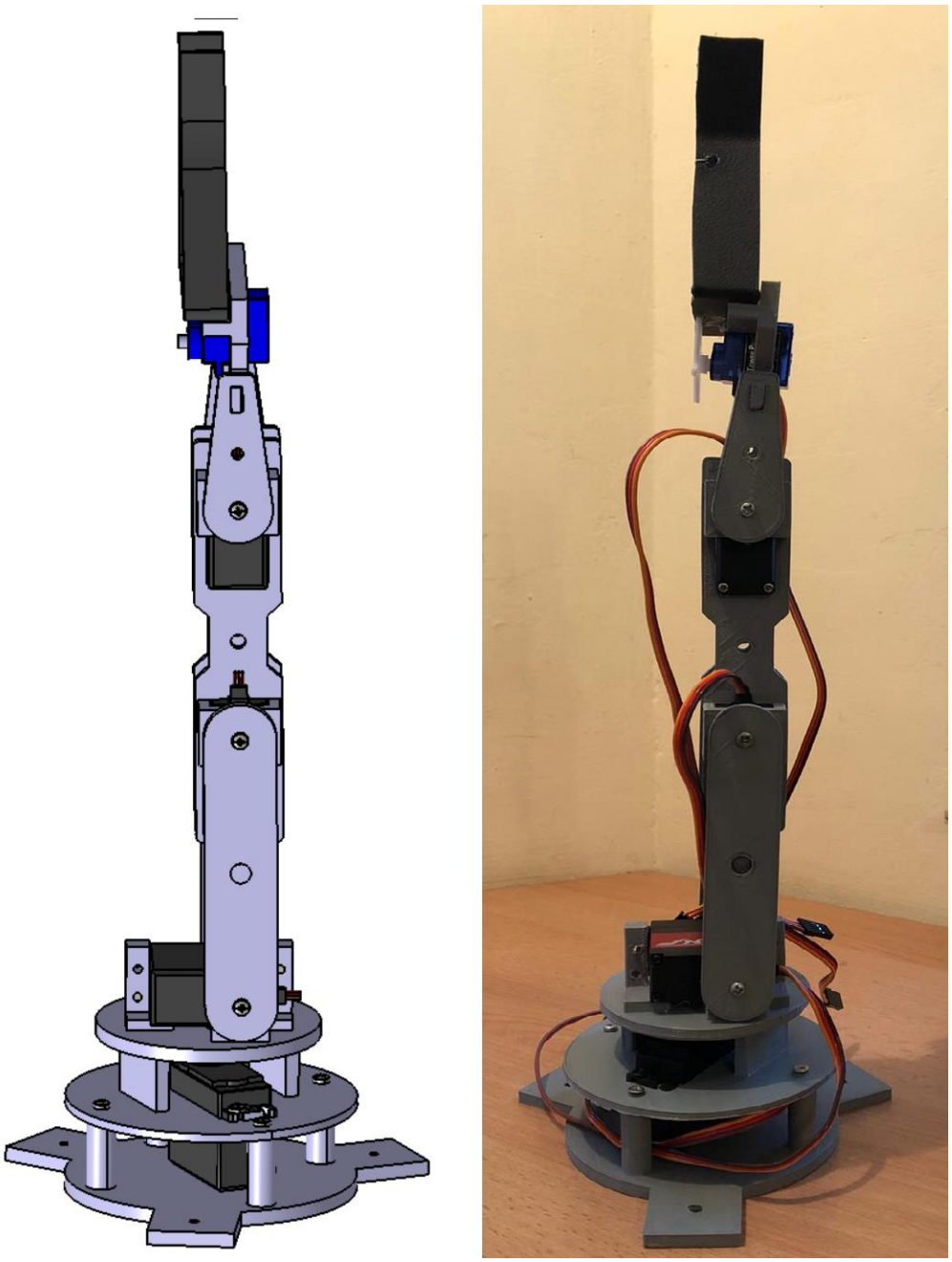
Slika 14. Servo JX PDI-6221MG i Tower PRO MG995 dimenzije[4][5]



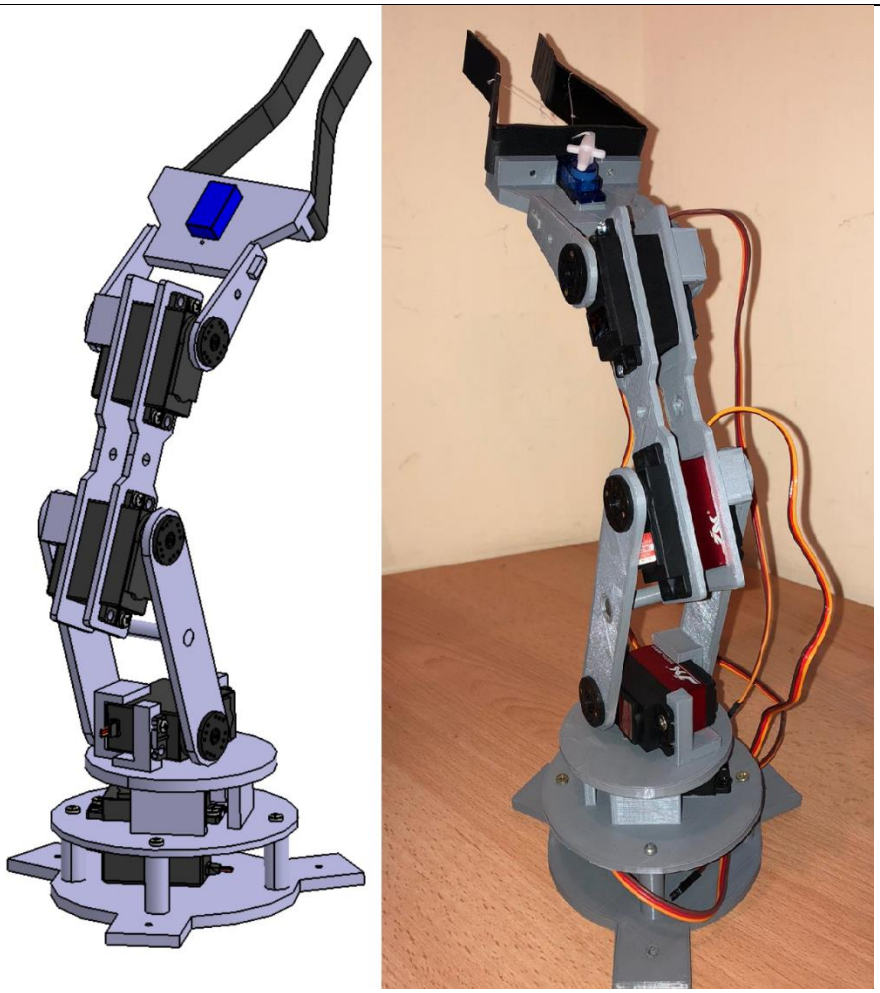
Slika 15. Servo SG90 dimenzije[6]

3.2. Usporedba CAD-a i stvarnog modela

U ovom potpoglavlju uspoređen je CAD-model sa stvarnim izrađenim i sklopljenim modelom. 3D printanje iako za neke složenije sklopove vjerojatno ne bi bilo dobro rješenje jer ipak još uvijek ne može dati tako precizne mjere i tolerancije kao npr. CNC strojevi ali za potrebe ovog završnog rada rezultati su bili i više nego zadovoljavajući.

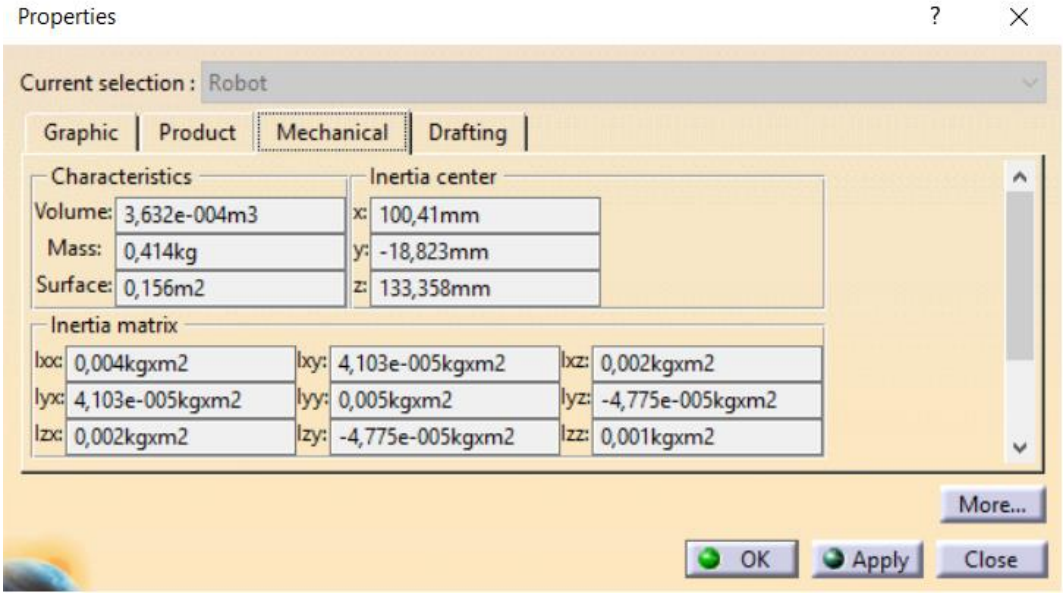


Slika 16. Usporedba za početni položaj



Slika 17. Usporedba 2

Također uspoređene su ukupne mase robota prema izračunu programskog paketa Catia i izvaganog stvarnog modela.



Slika 18. Podaci o mehaničkim karakteristikama iz Catie

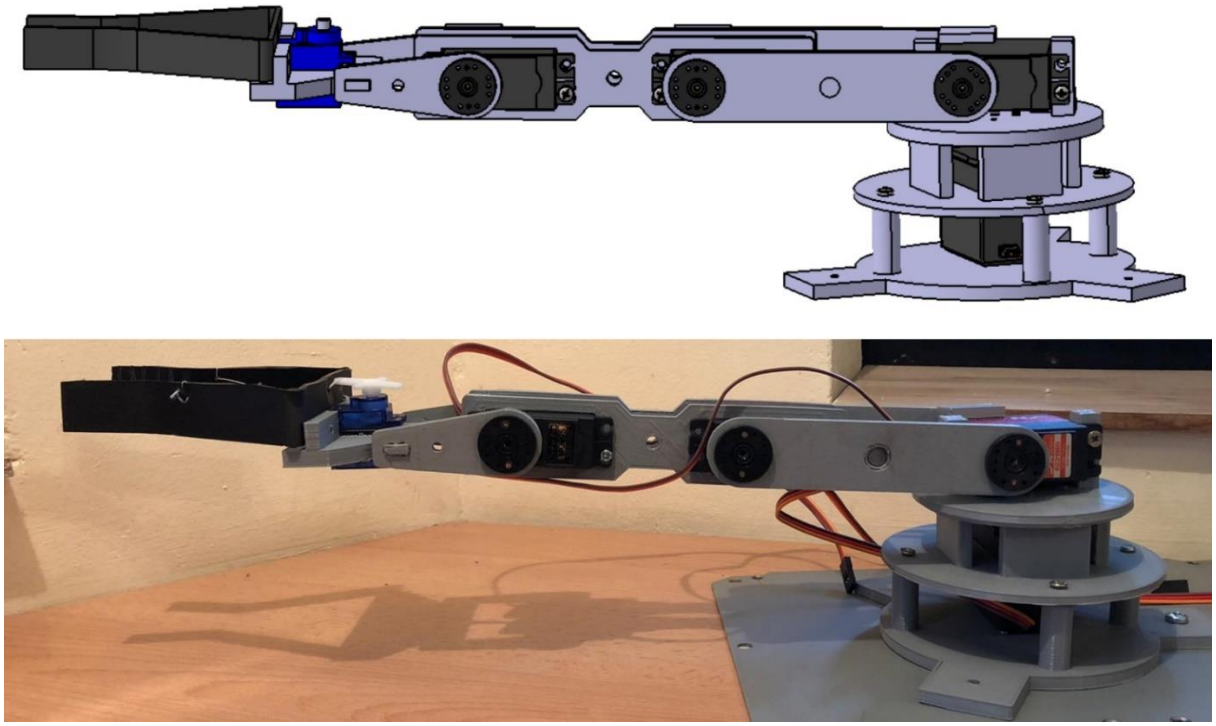


Slika 19. Izvagana masa robota

Iz slika 17. i 18. vidimo da je Catia na temelju zadanih karakteristika materijala i zadanim težinama motora dosta dobro procijenila masu robota i da je razlika između procijenjene i stvarne mase robota samo 4 grama.

Za krajnju usporedbu uzet je najnepovoljniji položaj robota, a to je položaj kada je motor 2 na jednoj od svojih maximalnih vrijednosti a to su kutevi od 90° ili -90°.

U tim položajima u točki hvatanja djeluje najveći moment koji je uzrokovan samom težinom robota ali i težinom predmeta koji se hvata.



Slika 20. Usporedba u najnepovoljnijem položaju

Udaljenost od osovine motora 2 do točke hvatanja predstavlja krak pomoću kojeg možemo odprilike izračunati da li motor 2 po karakteristikama zadovoljava traženi moment.

Taj krak određujemo prema udaljenostima sa slike 11. i on iznosi $L_2 + L_3 + L_4$.

Prema mjerama iz Catie podaci sa slike 11. iznose:

$$L_1 = 20 \text{ mm}$$

$$L_2 = 95 \text{ mm}$$

$$L_3 = 82 \text{ mm}$$

$$L_4 = 148 \text{ mm} \quad (2.1)$$

Pa onda krak kojeg ćemo označiti sa l iznosi:

$$l = L_2 + L_3 + L_4 = 325 \text{ mm} = 32{,}5 \text{ cm} \quad (2.2)$$

Da bi izračunali zadovoljava li moment motora težinu robota, te dobili korisni moment u najopterećenijem zglobu koji nam preostaje za podizanje predmeta uzet ćemo da dvije trećine mase robota što iznosi približno 280 grama prema podacima sa slike 18. djeluje na najudaljenijem kraku l.

Motor 2 je JX PDI-6221MG i njegov maximalni stabilni moment iznosi 20 kg/cm kao što je vidljivo iz njegovih karakteristika iz poglavlja 4.

Stoga grame pretvaramo u kg i dobivamo da je 280g = 0,28 kg.

Moment koji motor mora svladati bez tereta u tom slučaju iznosi:

$$M_m = 0{,}280 \cdot l \ = 0{,}280 \cdot 32{,}5 = \ 9{,}1 \ \text{kg/cm} \tag{2.3}$$

Pa s obzirom da je M_m manje od momenta motora zaključujemo da motor može savladati osnovne otpore uslijed vlastite težine robota.

S obzirom da je maximalni stabilni moment motora 20kg/cm kad se svladaju otpori zbog vlastite težine robota moment koji nam ostaje je za podizanje predmeta u najopterećenijem motoru iznosi:

$$M_t = 20 - \ M_m = 20 - 9{,}1 = 10{,}9 \ \text{kg/cm} \tag{2.4}$$

4. PCB upravljačka pločica

Robotska struktura je zamišljena tako da se svakim od 5 motora upravlja sa 5 potenciometara čijom vrtnjom se mijenja njihov otpor a time i upravljački napon prema motorima koji je proporcionalan njihovom kutu zakreta.

Napajanje prema motorima iznosi 6v i da nebi dolazilo do smetnja odvojeno je od napajanja procesora i ostalih komponenata čiji napon iznosi 5v.

Da bi se dobili ti naponi korišteni su ispravljači koji spuštaju ulazni napon na 5v i 6v.

Napajanje se vrši preko baterija, ali postoji i mogućnost ako se pojavi potreba za većim strujama zbog ograničenog kapaciteta baterije da se motori mogu napajati i iz vanjskog izvora.

Podaci o pozicijama motora koji se prvo upravljaju ručno putem potenciometara sprema se na SD karticu. Pritiskom gumba „*play*“ robot prelazi u automatski način rada, očitava podatke spremljene na SD karticu i ponavlja prethodno zadane pokrete sve dok ga se ne zaustavi s gumbom „*reset*“ kada robot ponovno prelazi u manualni način rada.

Na pločici se nalazi i USB komunikacije preko koje se program na procesoru može mijenjati ili nadopunjavati.

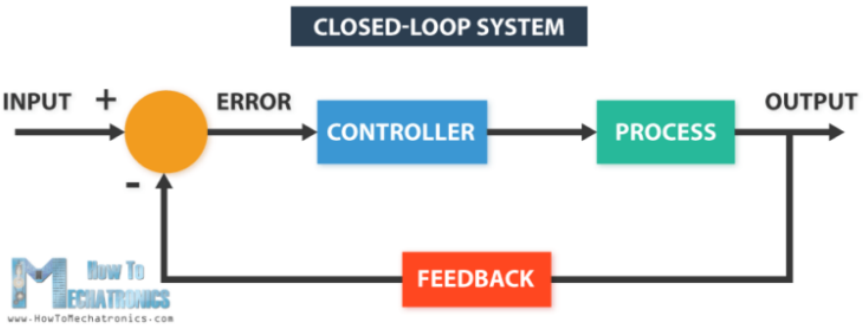
4.1. Izbor komponenata

Prvi i osnovni zadatak prije početka dizajniranja same pločice jest odabir komponenata koje će omogućiti da robot izvršava planirane funkcije.

4.1.1 Servo motori

Kao aktuatori odabrani su hobistički servo motori. Servo motori se općenito koriste kada nam je potrebno precizno kontroliranje pozicije osovine motora, a samim time i zgloba.

Funkcioniraju po načelu povratne veze po poziciji kao što je prikazano na slici 20. prema [3]:



Slika 21. Regulacijski krug servo motora [6]

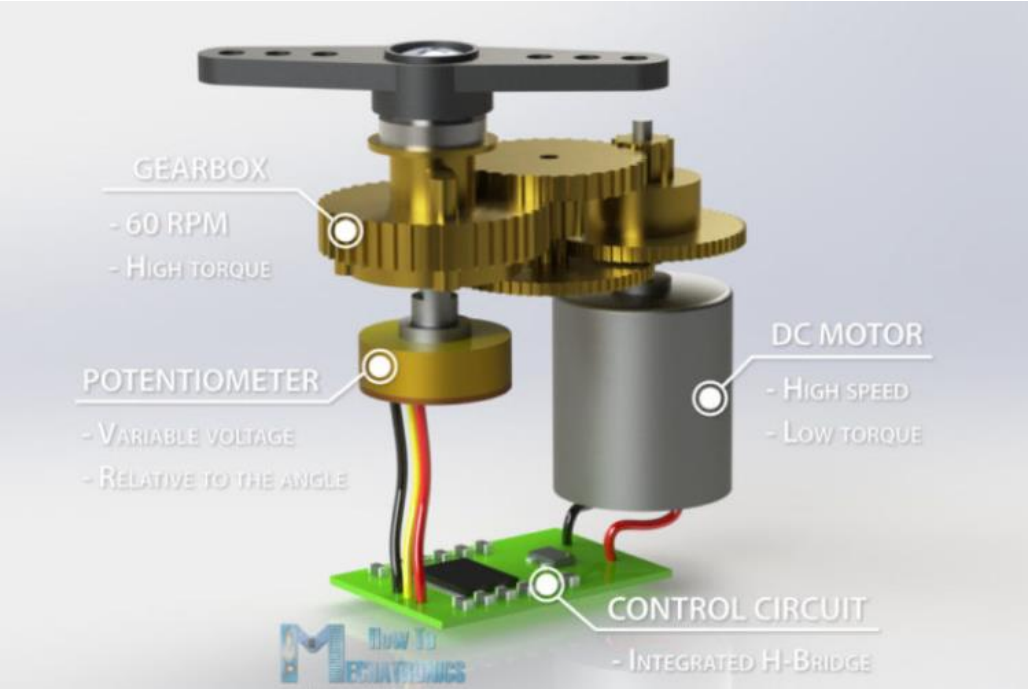
Kao ulaz dajemo neku određenu poziciju, senzor pozicije nam vraća informaciju o trenutnoj poziciji motora, nastaje greška na koju reagira kontroler te pokreće osovinu motora sve dok greška ne nestane ili bude unutar neke određene zadane vrijednosti.

Kod industrijskih servo motora kao senzori pozicije obično se koriste enkoderi, dok se kod hobističkih umjesto enkodera koristi potenciometar. Što je slučaj i kod motora odabranih u sklopu ovog završnog zadatka.

Također kao i većina hobističkih servo motora tako i ovi odabrani u sklopu ovog zadatka imaju 4 glavne komponente: DC motor, reduktor, potenciometar i kontrolni krug.

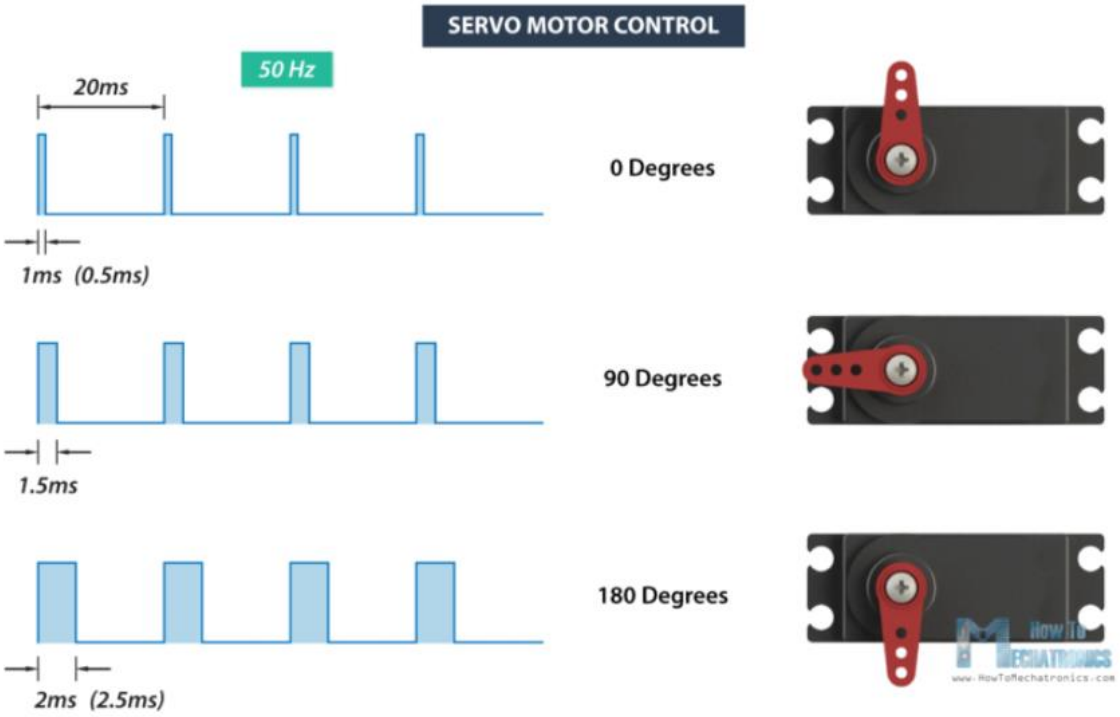
S obzirom da je DC motor visoke brzine vrnje, koristi se reduktor koji tu brzinu reducira da bi se povećao izlazni moment. Potenciometar je vezan na posljednji, izlazni zupčanik reduktora pa se rotira zajedno s osovinom motora te nam na temelju napona koji je proporcionalan kutu zakreta osovine daje informaciju o poziciji.

U kontrolnom krugu napon na potenciometru se uspoređuje sa naponom koji dolazi sa upravljačke linije te se na taj način regulira pozicija.



Slika 22. Presjek hobističkog servo motora [6]

Upravljanje se vrši slanjem serije pulseva kroz signalnu liniju. Frekvencija kontrolnog signala je 50 Hz ili se puls generira svakih 20 ms. Širina pulsa određuje kut zakreta osovine motora.



Slika 23. PMW signal kod servo motora [6]

Kao što se vidi iz slike 22. uobičajeno je da širina pulsa od 1 ms odgovara poziciji osovine od 0° . Širina pulsa od 1,5 ms odgovara 90° , a širina pulsa od 2 ms odgovara poziciji od 180° .

Ovisno o modelu širina pulsa može varirati od 0,5 do 2,5 ms. Takav tip signala zovemo PMW (Pulse-width modulation) signal.

Kao što je već ranije spomenuto odabrana su 3 tipa servo motora:

- JX PDI-6221MG



Slika 24. JX PDI-6221MG servo motor [4]

Brand:JX
Item:PDI-6221MG 20KG Large Torque Digital Standard Servo
Degree: 120°
Dead band: 2μs
Working frequence: 1520μs / 330hz
Operating Speed (4.8V): 0.18 sec/60°
Operating Speed (6V): 0.16 sec/60°
Stall Torque (4.8V): 17.25 kg.cm (239.55oz/in)
Stall Torque (6V): 20.32 kg-cm (281.89 oz/in)
Dimensions: 40.5X20.2X38mm /1.59 X0.80X49 in
Weight: 62 g (2.18oz)
Connector Wire Length: JR 265 mm (10.43in)
Bearing: 2BB
Usage:For RC Models

Slika 25. Karakteristike JX PDI-6221MG servo motora [4]

- Tower Pro MG995



Slika 26. Tower Pro MG995 servo motor [5]

MG995

Specification:

Weight: 55g
Dimension: 40.7×19.7×42.9mm
Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6v)
Operating speed: 0.20sec/60degree (4.8v); 0.16sec/60degree (6.0v)
Operating voltage: 4.8~ 6.6v
Gear Type: Metal gear
Temperature range: 0- 55deg
Dead band width: 1us
servo wire length: 32cm
Current draw at idle 10MA
No load operating current draw 170MA
Stall current draw 1200MA

Slika 27. Karakteristike Tower Pro MG995 servo motora [5]

- Tower Pro SG90



Slika 28. Tower Pro SG90 servo motor [6]

SG90 **Digital**

Specification

- Operating Voltage : 4.8~6.0V
- Operating Speed : 0.12sec/60 degree(4.8V)~0.1sec/60 degree(6.0V)
- Torque : 1.6kg/cm(4.8V)
- Dead Band Width : 5usec
- Temperature Range : -30~+60°C
- Cable Length : 25cm
- Servo Type : Analog Servo
- Brand Model : Tower Pro SG90

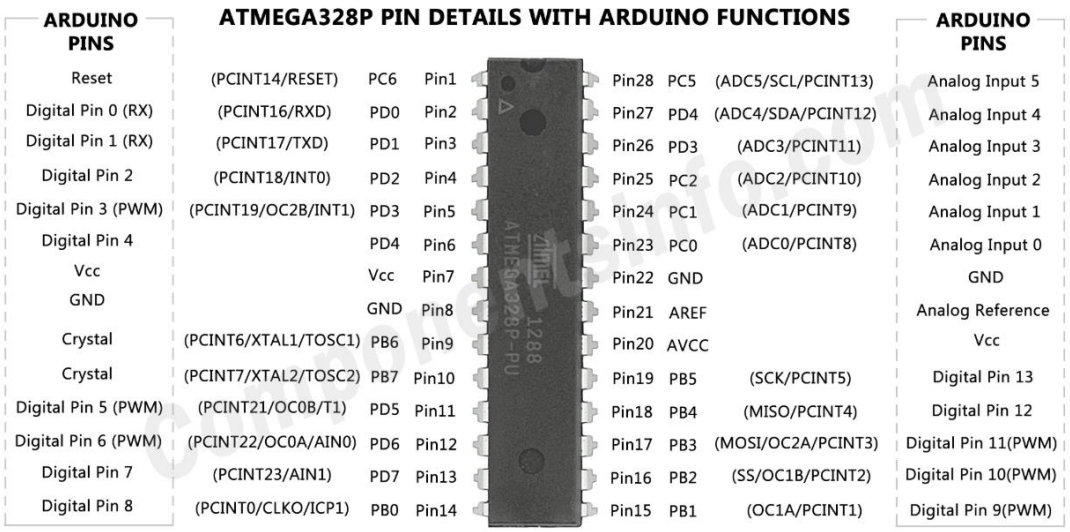
Slika 29. Karakteristike Tower Pro SG90 servo motora [6]

Svi motori na pločici napajani su sa naponom od 6V zato jer na tom operativnom naponu prema dokumentaciju produciraju najveći moment.

4.1.2 Mikroprocesor

Za mikroprocesor je odabran ATMEGA 328p-pu čije su karakteristike dane u nastavku.

Odabran je iz razloga što je lako kompatibilan s Arduinom a s obzirom da je program pisan u Arduino IDE sučelju zbog laganog prebacivanja programa najlakše je uzeti procesor koji se i nalazi na jednoj od Arduino pločica u ovom slučaju Arduino Uno.



Slika 30. Atmega 328p-pu pinovi [8]

ATMEGA328P-PU - ATMega328 8-bit AVR Microcontroller with 32kBytes Memory



Features

- 28-pin AVR Microcontroller
- Flash Program Memory: 32 kbytes
- EEPROM Data Memory: 1 kbytes
- SRAM Data Memory: 2 kbytes
- I/O Pins: 23
- Timers: Two 8-bit / One 16-bit
- A/D Converter: 10-bit Six Channel
- PWM: Six Channels
- RTC: Yes with Separate Oscillator
- MSSP: SPI and I²C Master and Slave Support
- USART: Yes
- External Oscillator: up to 20MHz

Part Summary	
Manufacturer	Atmel / Microchip
Manufacturer's Part Number	ATMEGA328P-PU
Manufacturer's Web Site	Microchip.com
Futurlec Part Number	ATMEGA328P-PU
Department	Microcontrollers
Category	Atmel
RoHS Compliant	Yes
Package Type	28 Pin Narrow DIP
Technical Data	ATMEGA328P-PU Datasheet

Slika 31. Atmega 328p-pu karakteristike [8]

Od ukupno 28 pinova 6 je analognih, a 13 je digitalnih. Od 13 digitalnih 6 ih je s mogućnošću upravljanja PMW signala i njih 5 je iskorišteno za upravljanje motorima.

5 analognih je iskorišteno za upravljanje potenciometrima i još 6 digitalnih za spajanje SD kartice i USB komunikacije.

4.1.3 Baterije

Korištene su dvije Li-ion 18650 punjive baterije napona 3.6v i kapaciteta 2600mAh, te su spojene u seriju što znači da nam je ukupni napon 7.2v koji onda pomoću ispravljača spuštamo na 6v za napajanje motora i 5v za napajanje procesora i ostalih komponenata. Dok nam ukupni kapacitet iznosi 2600mAh.



Brand:	LG
Model:	LGDB118650 (ICR18650B1)
Capacity:	2600mAh Rated
Voltage:	3.75V Nominal
Charging:	4.35V Maximum 1275mA Standard 2550mA Maximum
Discharging:	3.00V Cutoff 510mA Standard 5100mA Maximum
Description:	Teal Cell Wrapper White Insulator Ring 18650 Form Factor

Slika 32. Li-ion 18650 baterija karakteristike [9]

4.1.4 USB komunikacija

USB komunikacija izvedena je preko čipa FT232-rl koji pretvara USB u RS232 komunikaciju i obratno i preko njega se izvršava programiranje procesora.

Categories	Integrated Circuits (ICs) Interface - Controllers
Manufacturer	FTDI, Future Technology Devices International Ltd
Series	USBmadeEZ-UART
Packaging	Cut Tape (CT)
Part Status	Active
Protocol	USB
Function	Bridge, USB to UART

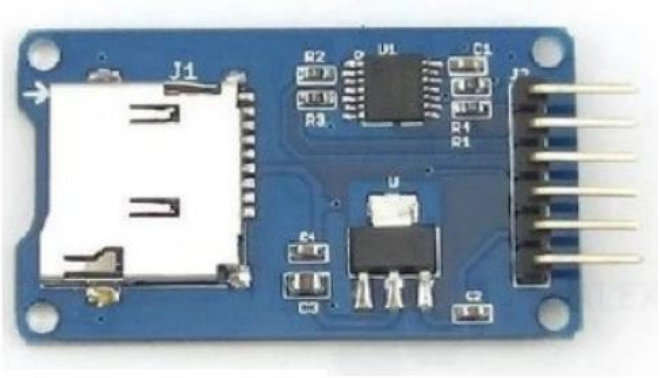
Interface	UART
Standards	USB 2.0
Voltage - Supply	3.3V ~ 5.25V
Current - Supply	15mA
Operating Temperature	-40°C ~ 85°C
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP

Slika 33. FT232-rl karakteristike [10]

4.1.5 Modul za SD karticu

Za ubacivanje SD kartice korišten je SD module koji između ostalog omogućava da i SD kartica bude spojena na napon 5v, umjesto na uobičajen napon za SD kartice koji iznosi 3,3v.

Na taj način je izbjegnuto ubacivanje posebnog ispravljača koji bi spuštao napon za napajanje SD kartice.



Specifications

- Working Voltage: 5V/3.3V
- Size:20x28mm
- Interface: SPI
- Compatible: MicroSD

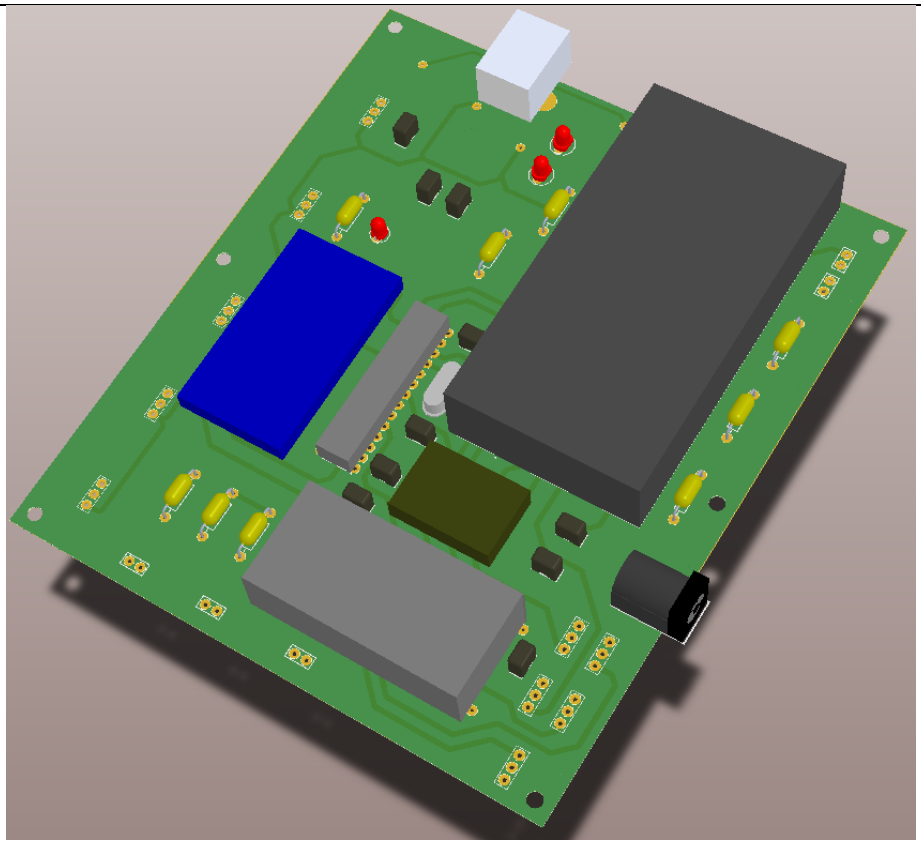
Slika 34. Module za SD karticu karakteristike [11]

4.1.6 Ostale komponente

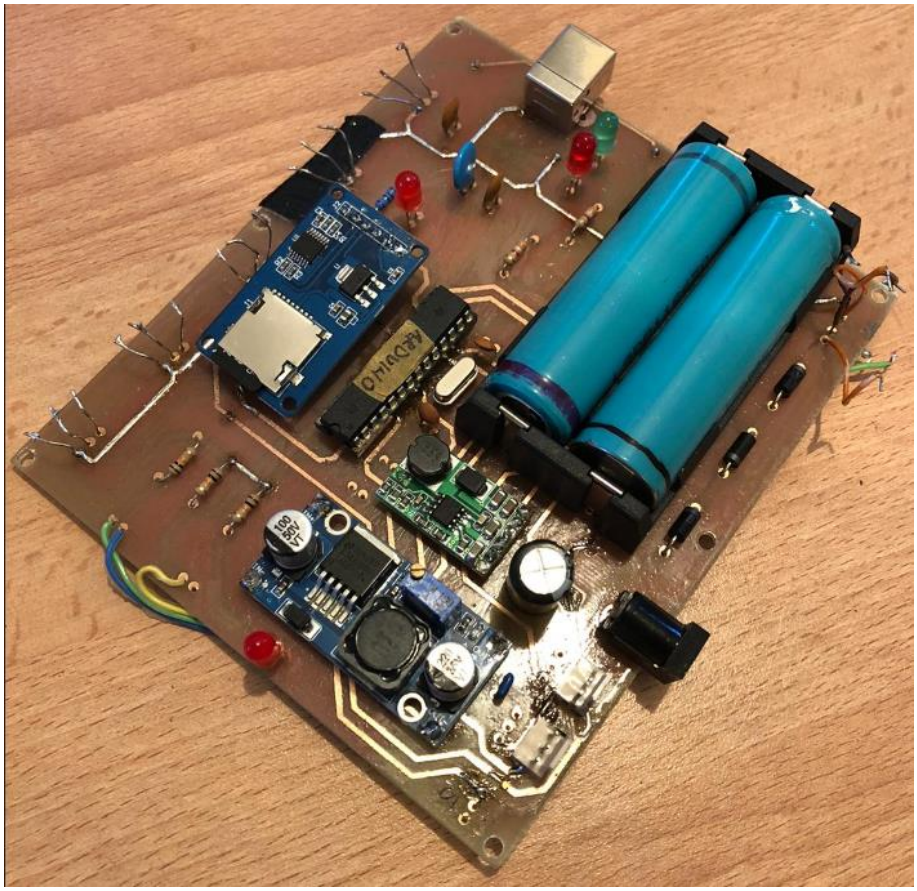
Od ostalih komponenata koje se nalaze ili su povezane s pločicom korišteno je:

- 5 potenciometara 50k
- Ispravljači na 5v i 6v
- Kristal 16Mhz
- Diode
- LED diode
- Otpornici
- Kondenzatori
- Gumbi
- Prekidač

Nakon odabira komponenata izrađena je dvoslojna pločica ručnim postupkom prema schematicu i slikama layera u prilogu.



Slika 35. 3D model PCB pločice



Slika 36. Izrađena PCB pločica

5. PROGRAMIRANJE

Programiranje je izvršeno u sučelju Arduino IDE. Napisan je jednostavan program pomoću kojeg se robot može upravljati ručno tako da svaki potenciometar kontrolira pojedini zglob to jest motor. Pozicije se snimaju i spremaju na SD karticu i pritiskom gumba robot očitava spremljene podatke i ponavlja zadane radnje.

5.1. Opis programa

Program započinjemo učitavanjem potrebnih libraryja. U ovom slučaju učitani su sljedeći:

- SPI – koji nam omogućuju komunikaciju s perifernim uređajima uz mikroprocesor kao „master“
- SD – koji nam omogućuju upravljanje SD karticom
- VarSpeedServo – poseban library koji nam omogućava upravljanje servo motorima ali i podešavanje brzine kojima će servo prelaziti između zadanih pozicija

Nakon toga učitavamo potrebne analogne i digitalne pinove, definiramo servo motore i inicijaliziramo gumb i njegovo početno stanje.

```
#include <SPI.h>
#include <SD.h>
#include <VarSpeedServo.h>

// CS pin za SD Card Module
const int chipSelect = 4;

// Analogni pinovi za potenciometar
int analogPin = A0;
int analogPin2 = A1;
int analogPin3 = A2;
int analogPin4 = A3;
int analogPin5 = A4;

// Integer za vrijednosti potenciometra
int val1;
int val2;
int val3;
int val4;
int val5;

// buffer za spremanje podataka
String buffer;

// definiranje servo motora
VarSpeedServo myservo;
VarSpeedServo myservo2;
VarSpeedServo myservo3;
VarSpeedServo myservo4;
VarSpeedServo myservo5;

// definiranje gumba i njegovo stanja
int pin_Button = 8;
int pin_Button_State = 1;
int pin_Button_State_Last = 1;
int recording = 0;
```

Slika 37. Inicijaliziranje varijabla

Nakon toga pridružujemo pojedini motor određenom digitalnom pinu koji nam može dati PMW signal. Ubacujemo naredbu kojom brišemo fail u koji ćemo spremati pozicije motora tako da svaki put kada resetiramo robota da nam se upisuju nove pozicije.

```
  Serial.println("card initialized.");
  SD.remove("servopos.txt");

  // Attach servo on pin 9 to the servo object
  myservo.attach(3);
  myservo2.attach(5);
  myservo3.attach(6);
  myservo4.attach(9);
  myservo5.attach(10);

   myservo.write(0);
  myservo2.write(70);
  myservo3.write(140);
  myservo4.write(50);
  myservo5.write(140);
}
```

Slika 38. Definiranje pinova motora

Potom definiramo prekidačku logiku kod gumba na način da kad ga pritisnemo prvi put gumb nam mijenja stanje i stavlja robot u mod manualnog upravljanja i snimanja.

A kad ga pritisnemo drugi put prelazi u mod automatskog rada i oponašanja snimljenih kretanja.

```
void loop()
{
  pin_Button_State = digitalRead(pin_Button);
  if (pin_Button_State != pin_Button_State_Last) {
    if (pin_Button_State == LOW) {
      recording++;
      Serial.println();
      Serial.print("   Buttonpress: ");
      Serial.println(recording);
    }
    else {
    }
    delay(50);
  }

       if ( recording == 1) {
          Record();
       }
      if (recording > 1){
        Play();
       }
}
```

Slika 39. Definiranje modova robota

Zatim definiramo upravljanje robota preko potenciometara i inicijaliziramo string varijablu u koju ćemo spremati podatke očitane s potenciometara te ih kasnije prebaciti na SD karticu, pokazan je taj postupak na prvom motoru. Kod prebacivanja na SD karticu za svaki motor dodan je i njegov inicijalizator, npr. za motor jedan dodan je inicijalizator „S01:“ tako da kod čitanja podataka s SD kartice možemo lagano pripojiti snimljene podatke i korišteni motor.

```
102 void Record()
103 {
104    String dataString = "";
110 
111          val1 = map(analogRead(analogPin), 0, 1023, 0, 180);
112          dataString += String(val1);
113          myservo.write(val1,20,true);
114 
137          File dataFile = SD.open("servopos.txt", FILE_WRITE);
138 
139 
140       if (dataFile) {
141 
142                dataFile.print("S01:");
143                dataFile.println(dataString);
```

Slika 40. Snimanje i spremanje podataka na SD karticu

Potom prelazimo u automatski mod rada gdje učitavamo podatke sa kartice i pridružujemo ih zadanom motoru. Prvo pretražujemo stringove koji su spremljeni na SD karticu redak po redak, konkretno za motor 1 tražimo da li vrijednost ima predmetak „S01“ i u slučaju da ima tu vrijednost pridružujemo motoru 1, te postavljamo brzinu tako da nam polagano prelazi iz pozicije u poziciju da što je više moguće izbjegnemo trzaje i vibracije.

```
166 void Play()
167 {
168    File dataFile = SD.open("servopos.txt");
169 
170   // ako je file dostupan čitamo ga
171   if (dataFile) {
172     while (dataFile.available()) {
173     // pišemo liniju po liniju u varijablu buffer
174     buffer = dataFile.readStringUntil('\n');
175 
176         if(buffer.indexOf("S01:") >= 0){
177               Serial.print("S01:");
178               Serial.println(buffer.substring(4));
179               myservo.write(buffer.substring(4).toInt(),20,true);
180 
181         }
```

Slika 41. Učitavanje podataka sa SD kartice

6. KUČIŠTE

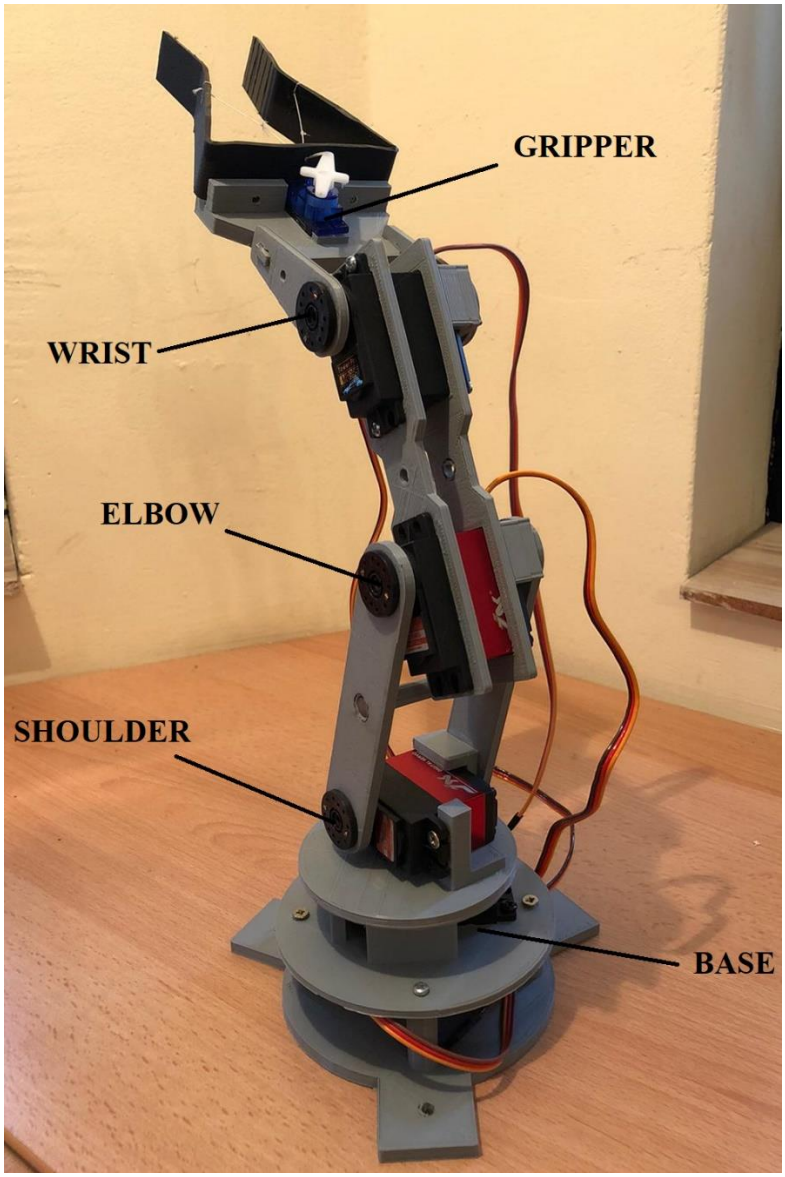
Kučište je izrađeno također 3D printanjem iz PLA plastike. Modelirano je u Solidworksu i u njega se stavlja pločica i sve ostale komponente preko kojih se robot upravlja. Također ima prekidač, utore u koje se ukapčaju motori, USB utor za programiranje i utor za napajanje motora u slučaju da se baterije potroše ili su potrebne veće struje.



Slika 42. 3D model kučišta

Svaki potenciometar upravlja jednim od zglobova čija su imena redom „base“ , „shoulder“ , „elbow“ , „wrist“ i „gripper“ .

Na slici 41. prikazani su ti zglobovi plus hvataljka na robotu.



Slika 43. Zglobovi robota po imenima

7. PROCJENA TROŠKOVA

U tablici 1. prikazani su procijenjeni troškovi izrade robota po komponentama i potrebnom materijalu. Većina komponenata kupljena je sa Ali expressa ili eBaya.

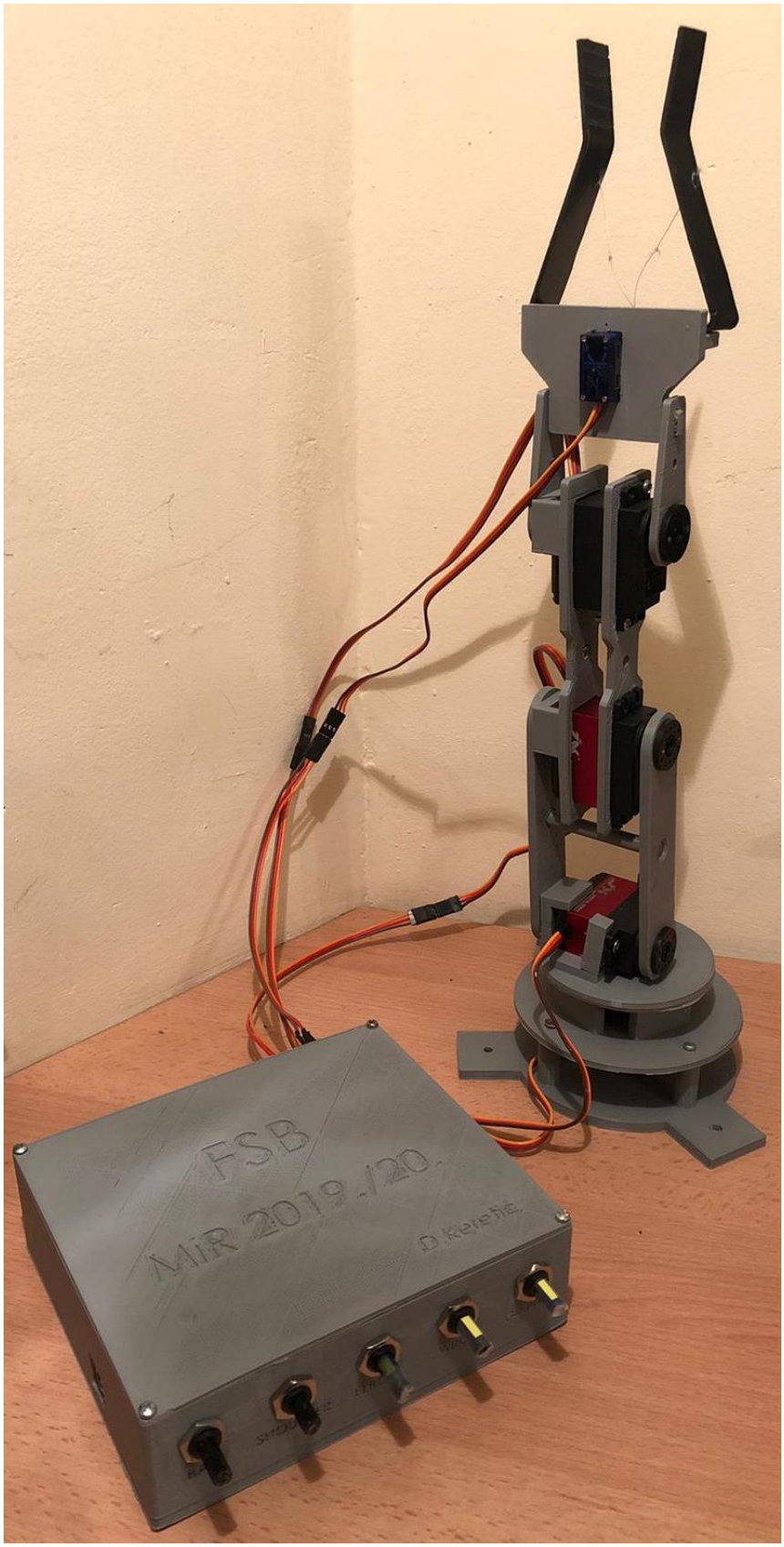
Troškovi 3D printanja nisu uračunati s obzirom da je sve printano kod prijatelja koji posjeduje 3D printer.

Tablica 1. Procjena troškova

Dio	Komada	Prosječna cijena
PLA PLASTIKA	1	150 kn
Tower Pro servo motori	5	50 kn
JX-PDI servo motori	2	200 kn
Ispravljači 5v i 6v	2	20 kn
Micro SD module	1	5 kn
Potenciometri 50k	5	10 kn
Baterije i nosač za baterije	4	140 kn
ATMEGA 328p-pu procesor	1	10 kn
USB connector	1	10 kn
Power Jack	1	10 kn
LED diode, diode, otpornici, kondenzatori, prekidač		20 kn
Ukupno:		625 kn

Ukupni zbroj troškova iznosi otprilike 600 kn za sve dijelove ne uključujući troškove 3D printanja i izrade pločice jer za to nisu potrošena nikakva sredstva.

8. ZAVRŠNI SKLOP



Slika 44. Završni sklop

9. ZAKLJUČAK

U ovom radu uspješno je izrađena robotska ruka sa 4 stupnjeva slobode i mogućnošću hvatanja predmeta, njezina upravljačka pločica i kučište, te je napisan jednostavan program da se prikažu neke od mogućnosti takvog sklopa.

Naučeni su osnovni proračuni iz kinematike robota, neki glavni problemi prilikom stvarne izrade nečega što na CAD modelu izgleda vrlo jednostavno i skladno. Način funkcioniranja servo motora, SD kartice, ispravljača, ručni način izrade i lemljenja komponenata na PCB pločicu itd.

Ipak ovo je tak mali dio onoga što je danas potrebno u modernoj robotici i predstavlja tek početnu razinu znanja koju je potrebno imati da bi se moglo projektirati i raditi s modernim robotskim rukama i robotima općenito.

Područja poboljšanja su stvarno velika, najprije od same konstrukcije robota gdje se mogu koristiti bolji i skuplji materijali da bi se povećala krutost robota, samim time i bolji i precizniji motori čime bi se smanjilo trzanje i dobili bi se puno finiji prijelazi između pozicija. Ubacivanje raznih senzora koji bi robotu dobavljali informacije o okolini npr. senzori pozicije, sile, kamere itd. Ubacivanje umjetne inteligencije, koja bi dala određenu autonomnost i sposobnost odlučivanja. Regulacija npr. po poziciji i brzini čime bi se dobivala veća točnost, ponovljivost i finije kretanje robota. Kvalitetniji ispravljači koji bi stvarali manje smetnja pri upravljanju motorima, kvalitetnije baterije, naprednije upravljanje npr. joystickom umjesto potenciometrima itd. Ali ipak sva ta poboljšanja donose i puno veću cijenu samog robota i njegovih komponenata što ipak treba uzeti u obzir kada se plaća iz vlastitog studentskog džepa.

Kad se usporedi cijena izrade vlastite robotske ruke sa cijenama hobističkih kakve se danas mogu naći na tržištu dolazimo do zaključka da je vjerojatno puno jednostavnije kupiti gotovu robotsku ruku ako se želi raditi samo na algoritmima njezinog upravljanja, ali ako se želi malo više naučiti o njezinim komponentama, pozadinskoj elektronici i osnovnom načinu izrade, te svim problemima koji sa time dolaze izrađivanje svojeg vlastitog sklopa sigurno ima puno prednosti u odnosu na kupovanje gotovog proizvoda.

LITERATURA

[1] https://www.robotshop.com/

[2] Predavanja iz kolegija Robotika, portal e-učenje

[3] T.Šurina, M.Crneković, Industrijski roboti, Školska knjiga, Zagreb 1990.

[4] https://www.banggood.com/JX-PDI-6221MG-20KG-Large-Torque-Digital-Standard-Servo-120-Degree-For-RC-Model-p-973947.html

[5] https://www.towerpro.com.tw/product/mg995/

[6] https://servodatabase.com/servo/towerpro/sg90

[7] https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/

[8] https://www.futurlec.com/Atmel/ATMEGA328P-PUpr.shtml

[9] https://secondlifestorage.com/showthread.php?tid=1751

[10] https://www.digikey.com/product-detail/en/ftdi-future-technology-devices-international-ltd/FT232RL-REEL/768-1007-1-ND/1836402

[11] https://store.fut-electronics.com/products/micro-sd-card-module

PRILOZI

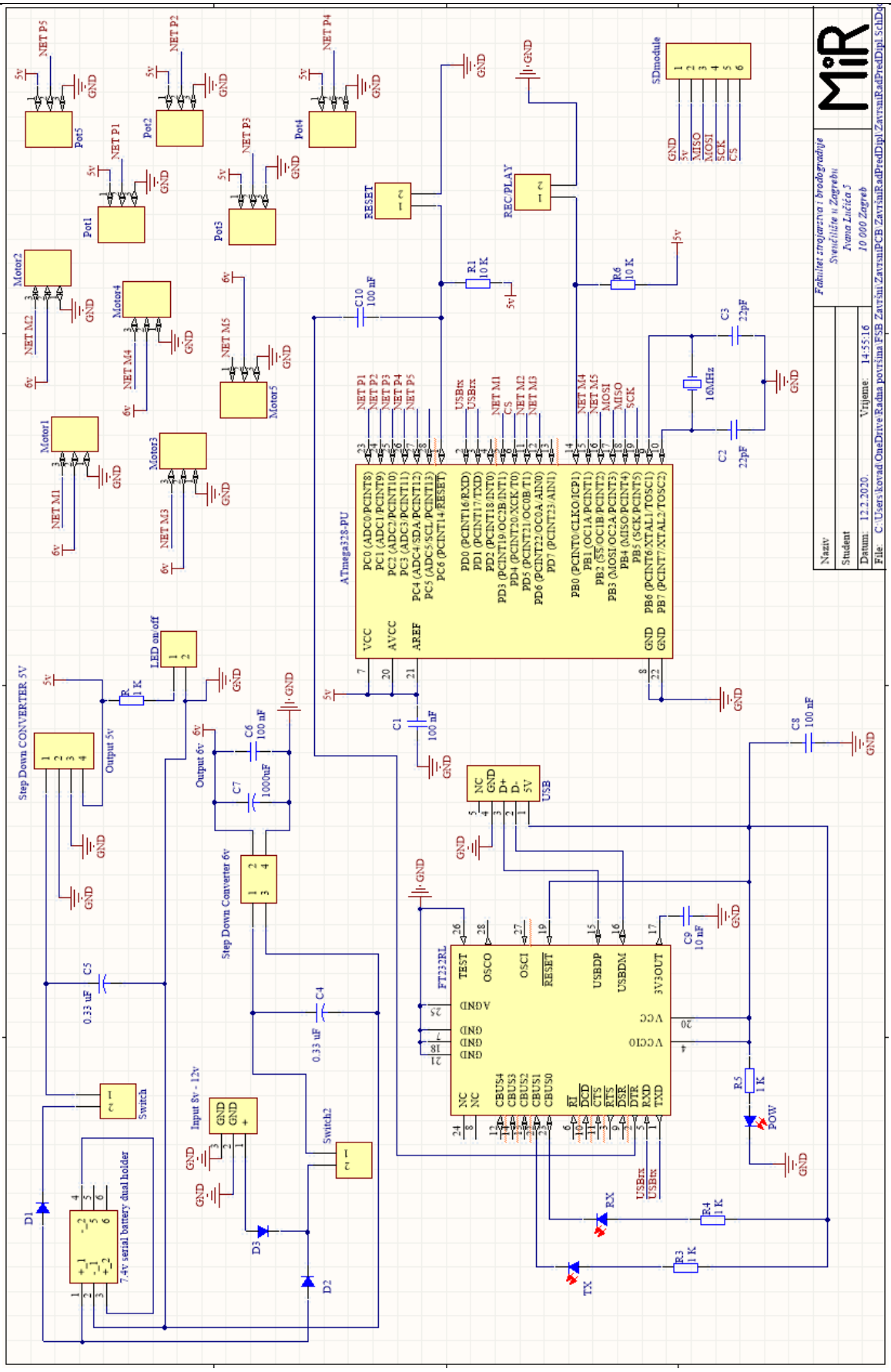
I. CD-R disc

II. Schematic

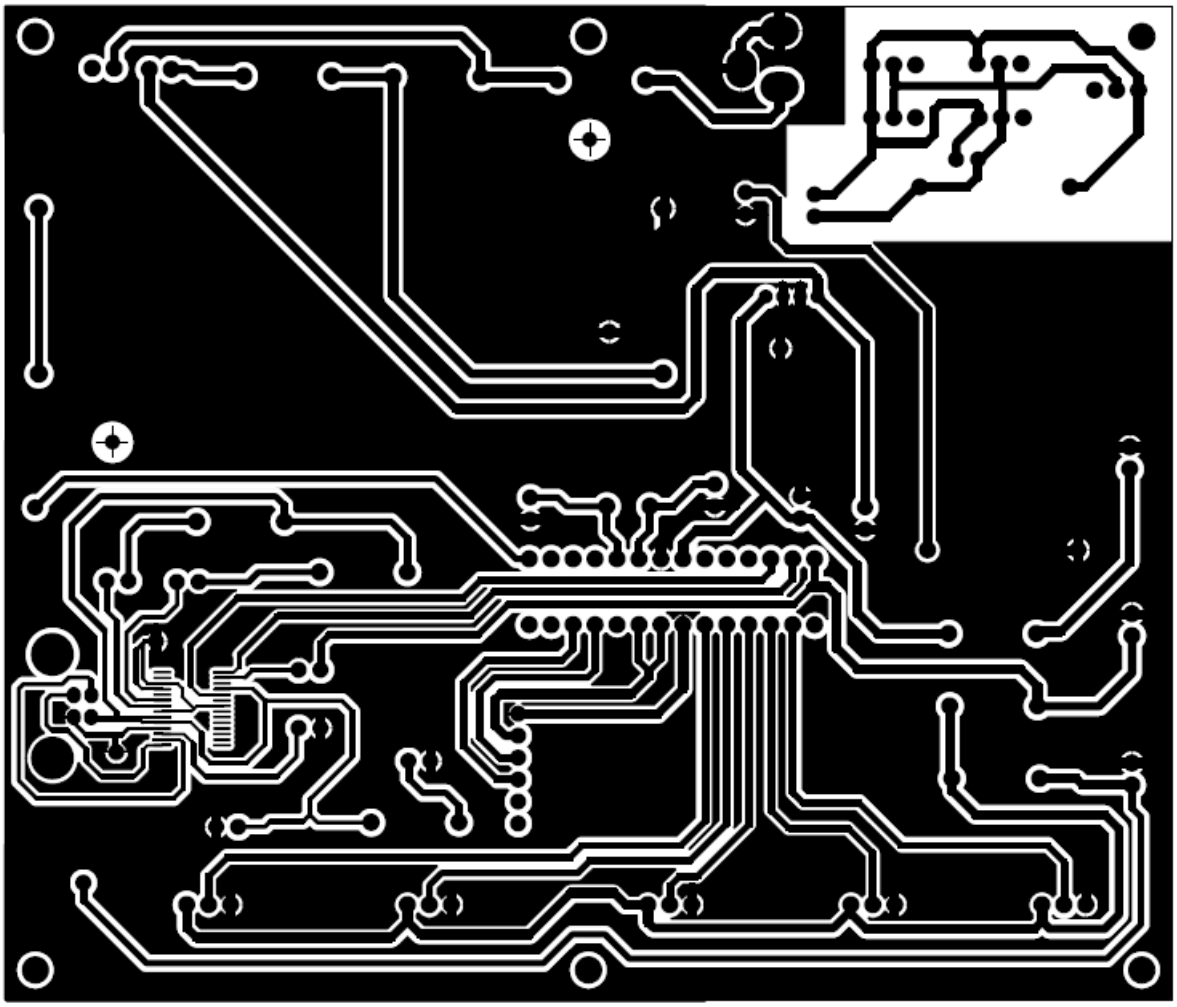
III. Bottom layer

IV. Top layer

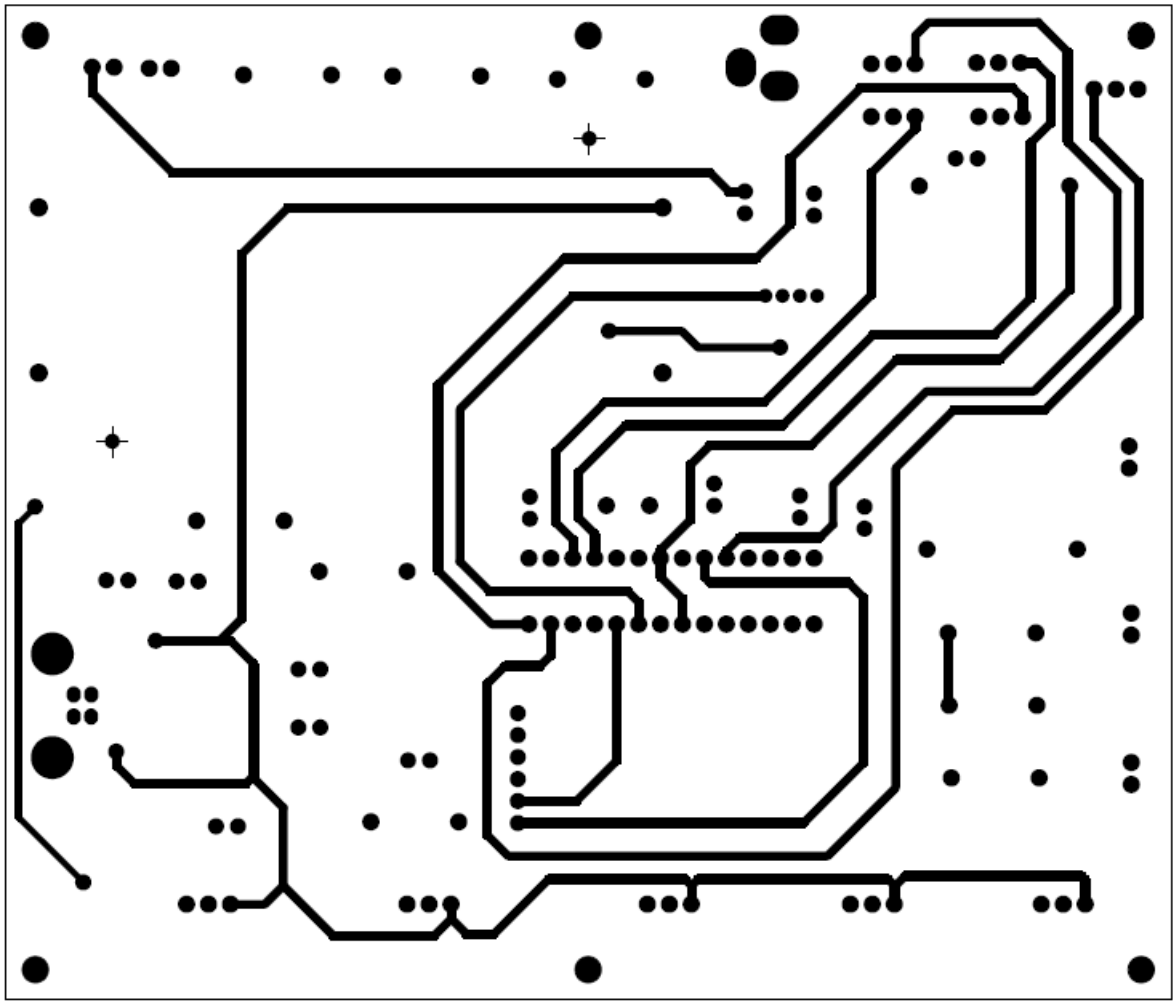
V. Programski kod



Prilog 1. Schematic



Prilog 2. Bottom layer



Prilog 3. Top layer

```
#include <SPI.h>
#include <SD.h>
#include <VarSpeedServo.h>

// CS pin
const int chipSelect = 4;

// Analogni pinovi za potenciometre
int analogPin = A0;
int analogPin2 = A1;
int analogPin3 = A2;
int analogPin4 = A3;
int analogPin5 = A4;


// Integer za vrijednosti potenciometara
int val1;
int val2;
int val3;
int val4;
int val5;


// buffer za spremanje podataka
String buffer;


// definiranje servo motora
VarSpeedServo myservo;
VarSpeedServo myservo2;
VarSpeedServo myservo3;
VarSpeedServo myservo4;
VarSpeedServo myservo5;
```

```
// definiranje gumba i njegovog stanja
int pin_Button = 8;
int pin_Button_State = 1;
int pin_Button_State_Last = 1;
int recording = 0;



void setup()
{

Serial.begin(9600);
while (!Serial) {
    ; // cekanje da se serijski port poveže
  }

  Serial.print("Initializing SD card...");


  // provjera da li je kartica dostupna
       if (!SD.begin(chipSelect)) {
              Serial.println("Card failed, or not present");

              while (1);
                                    }
  Serial.println("card initialized.");
  SD.remove("servopos.txt");

  // pridavanje servo motora digitalnim PMW pinovima
  myservo.attach(3);
  myservo2.attach(5);
  myservo3.attach(6);
  myservo4.attach(9);
  myservo5.attach(10);
```

```
   myservo.write(0);
  myservo2.write(70);
  myservo3.write(140);
  myservo4.write(50);
  myservo5.write(140);
}



void loop()
{


  pin_Button_State = digitalRead(pin_Button);
  if (pin_Button_State != pin_Button_State_Last) {
    if (pin_Button_State == LOW) {
      recording++;
      Serial.println();
      Serial.print("   Buttonpress: ");
      Serial.println(recording);
    }
    else {
    }
    delay(50);
  }


       if ( recording == 1) {
          Record();
       }

      if (recording > 1){
        Play();
        }
}
```

```
void Record()
{
   String dataString = "";
   String dataString2 = "";
   String dataString3 = "";
   String dataString4 = "";
   String dataString5 = "";



        val1 = map(analogRead(analogPin), 0, 1023, 0, 180);
        dataString += String(val1);
        myservo.write(val1,20,true);


        val2 = map(analogRead(analogPin2), 0, 1023, 0, 180);
        dataString2 += String(val2);
        myservo2.write(val2,20,true);


        val3 = map(analogRead(analogPin3), 0, 1023, 0, 180);
        dataString3 += String(val3);
        myservo3.write(val3,20,true);


        val4 = map(analogRead(analogPin4), 0, 1023, 0, 180);
        dataString4 += String(val4);
        myservo4.write(val4,20,true);


        val5 = map(analogRead(analogPin5), 0, 1023, 50, 180);
        dataString5 += String(val5);
        myservo5.write(val5,20,true);
```

```
        File dataFile = SD.open("servopos.txt", FILE_WRITE);


    if (dataFile) {

            dataFile.print("S01:");
            dataFile.println(dataString);

            dataFile.print("S02:");
            dataFile.println(dataString2);

            dataFile.print("S03:");
            dataFile.println(dataString3);

            dataFile.print("S04:");
            dataFile.println(dataString4);

            dataFile.print("S05:");
            dataFile.println(dataString5);

            dataFile.close();
            }
// ako ne može otvoriti file - error
else {
  Serial.println("error opening servopos.txt");
}
```

```
 File dataFile = SD.open("servopos.txt");


// If the file is available, read it
if (dataFile) {
  while (dataFile.available()) {
  // Write one line to buffer
  buffer = dataFile.readStringUntil('\n');


      if(buffer.indexOf("S01:") >= 0){
            Serial.print("S01:");
            Serial.println(buffer.substring(4));
            myservo.write(buffer.substring(4).toInt(),20,true);

      }


      if(buffer.indexOf("S02:") >= 0){
            Serial.print("S02:");
            Serial.println(buffer.substring(4));
            myservo2.write(buffer.substring(4).toInt(),20,true);

      }


      if(buffer.indexOf("S03:") >= 0){
            Serial.print("S03:");
            Serial.println(buffer.substring(4));
            myservo3.write(buffer.substring(4).toInt(),20,true);
```

```
      if(buffer.indexOf("S04:") >= 0){
            Serial.print("S04:");
            Serial.println(buffer.substring(4));
            myservo4.write(buffer.substring(4).toInt(),20,true);

      }


      if(buffer.indexOf("S05:") >= 0){
            Serial.print("S05:");
            Serial.println(buffer.substring(4));
            myservo5.write(buffer.substring(4).toInt(),20,true);

      }
    }

    dataFile.close();
  }


  else {
    Serial.println("error opening servopos.txt");
  }
}
```

Prilog 4. Programski kod