



UNIVERZITET U BANJOJ LUCI
ELEKTROTEHNIČKI FAKULTET



Aleksandar Ritan

**AUTOMATSKO PISANJE TEKSTA
POMOĆU MITSUBISHI ROBOTA**

Diplomski rad

Banja Luka, oktobar 2021. godine

Tema: Automatsko pisanje teksta pomoću Mitsubishi robota

Komisija: prof. dr Igor Krčmar, predsjednik
prof. dr Petar Marić, mentor
ma Velibor Đalić, član

Napomena: Uz štampanu verziju rada predaje se i kompakt disk (CD)
sa tekstrom rada i prilozima.

Kandidat:
Aleksandar Ritan

UNIVERZITET U BANJOJ LUCI
ELEKTROTEHNIČKI FAKULTET
KATEDRA ZA AUTOMATIKU

Predmet: Robotika

Tema: Automatsko pisanje teksta pomoću Mitsubishi robota

Zadatak: Realizovati pisanje teksta pomoću robota Mitsubishi. Izvršiti kalibraciju robota, opisati okruženje za programiranje, programirati upravljanje i komunikaciju sa drugim elementima sistema. Realizovati automatsko pisanje teksta za zadate: veličinu površine za pisanje, brzinu pisanja, veličinu slova, stil pisanja (obični ili kurziv) i dodatne efekte kao što su podvučena, precrtana i dvostruko precrtana slova.

Mentor: prof. dr Petar Marić

Kandidat: Aleksandar Ritan [1150/14]

Banja Luka, septembar 2021. godine

Zahvaljujem mentoru prof. dr Petru Mariću na pomoći pri izboru teme za diplomski rad, kao i na korisnim sugestijama i strpljenju tokom izrade rada.

Takođe, zahvaljujem ma Veliboru Đaliću, asistentu, na izdvojenom vremenu i na savjesnoj stručnoj pomoći prilikom izrade praktičnog dijela diplomskog rada.

Želim da zahvalim i svim profesorima i asistentima Elektrotehničkog fakulteta u Banjoj Luci, čije je zalaganje i trud da nam prenesu znanje rezultiralo mojim uspješnim završetkom prvog ciklusa studija.

Naposljetu zahvaljujem svojoj porodici na pruženoj podršci, na pomoći i razumijevanju u toku studija.

Sadržaj

1. Uvod.....	1
2. Osnovni pojmovi iz robotike i tehnike	2
2.1. Osnovni stavovi iz teorije mehanizama	2
2.2. Homogene transformacije	5
2.3. Geometrija manipulacionih robota.....	9
2.3.1. Geometrija segmenata.....	9
2.3.2. Položaj završnog uređaja.....	12
2.4. Kinematika manipulacionih robota.....	18
2.4.1. Direktni problem kinematike	18
2.4.2. Inverzni problem kinematike	21
2.5. Tehničko pismo	22
3. Realizacija sistema za automatsko pisanje teksta pomoću Mitsubishi RV-2SDB robo ta	24
3.1. Opis glavnih komponenata sistema	24
3.1.1. Mitsubishi RV-2SDB industrijski robot	24
3.1.2. Mitsubishi CR1DA-700 robotski kontroler	27
3.2. Kalibracija sistema	28
3.2.1. Kalibracija položaja vrha alata.....	28
3.2.2. Kalibracija položaja površine za pisanje	35
3.3. Opis programskog okruženja i struktura programa.....	37
3.3.1. Glavna aplikacija.....	37
3.3.2. Aplikacija za podešavanje parametara robota.....	42
3.3.3. Aplikacija za podešavanje parametara radnog prostora.....	44
3.3.4. Aplikacija za podešavanje parametara upravljanja robotom.....	46
3.3.5. Aplikacija za prenos programa na robota	47
3.4. Rezultati	50
4. Zaključak	53
5. Literatura.....	54

1. Uvod

U današnje vrijeme svijet se suočava sa ubrzanim razvojem nauke i tehnike, koji je podstaknut željom za unapređenjem postojećih tehnologija i za poboljšanjem efikasnosti izvršavanja različitih poslova. Industrija je jedan od glavnih pokretača tih promjena. Visokim stepenom automatizacije u industriji teži se ka smanjenju ljudskog faktora u proizvodnim procesima, čime se postiže veća produktivnost, ekonomičnost i bolji kvalitet finalnih proizvoda. U postojećim industrijskim pogonima većinom je zastupljena fiksna automatizacija, koja podrazumijeva visokoautomatizovane proizvodne linije namijenjene serijskoj proizvodnji određenog proizvoda. Međutim, zbog veoma oštре konkurencije i ubrzanog razvoja često se javlja potreba za poboljšanjima i izmjenama proizvoda radi produženja njegove aktuelnosti. Zbog toga fiksne proizvodne linije nisu isplative i sve se više teži ka uvođenju fleksibilne automatizacije u proizvodne procese. Ona omogućava jednostavne izmjene proizvodnog programa i proizvodnju različitih proizvoda, bez promjene opreme koja učestvuje u proizvodnji [1],[2]. Neizostavan dio fleksibilne automatizacije predstavljaju robotski sistemi. Tu su u velikoj mjeri prisutni i računarski sistemi koji se koriste za upravljanje proizvodnim procesom (engl. *Computer-aided manufacturing – CAM*) ili se koriste u fazi planiranja i projektovanja (engl. *Computer-aided design – CAD*). Jedno takvo CAD programsko rješenje korišteno je i u sklopu diplomskog rada sa svrhom vizuelizacije robota i radnog okruženja.

Robotika predstavlja jednu interdisciplinarnu primijenjenu nauku, koja obuhvata područja mehanike, elektronike, računarske tehnike i automatike. Za uspješnu realizaciju ovog rada bilo je potrebno objediniti znanja iz svih ovih oblasti i pretočiti ih u jednu funkcionalnu cjelinu.

U sklopu rada realizovan je sistem za automatsko pisanje teksta pomoću robota Mitsubishi RV-2SDB. Radom je demonstrirana fleksibilnost primjene jednog takvog industrijskog robota, koji u sklopu industrijskih pogona najčešće obavlja potpuno različite zadatke kao što su: premještanje objekata, skladištenje, zavarivanje, farbanje, određene precizne mašinske radnje i slično. Realizovani sistem može se koristiti u edukativne svrhe, ali se uz manje modifikacije može prilagoditi i konkretnom zadatku u sklopu nekog realnog procesa. Na primjer, sistem se, upotrebom posebnog alata, može koristiti i za graviranje teksta na određenim površinama ili mašinskim pozicijama. Sistem je praktično realizovan u laboratoriji na Elektrotehničkom fakultetu, koja posjeduje svu neophodnu opremu korištenu u ovu svrhu.

Rad se sastoji iz pet poglavlja, koja pokrivaju sve, od potrebne teorijske osnove, pa do demonstracije rješenja i rezultata koji su postignuti u laboratoriji, što uključuje i programsko rješenje koje je realizovano korištenjem MATLAB programskog okruženja. Opisani su i svi korišteni elementi sistema, kao i način uspostavljanja njihove međusobne komunikacije. U posljednjem poglavlju dat je spisak literature koja je korištena u toku izrade rada, a poredana je u redoslijedu referenciranja u tekstu.

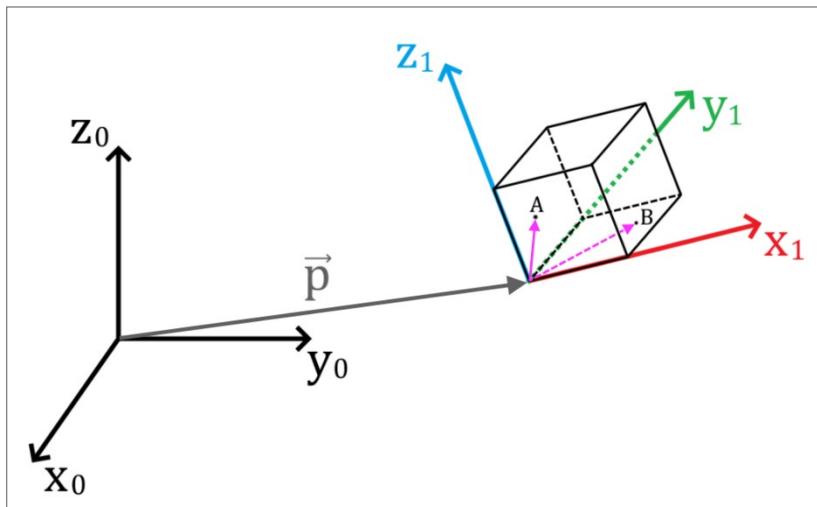
2. Osnovni pojmovi iz robotike i tehnike

Glavna komponenta robotskog manipulatora jeste njegova mehanička struktura. Da bi se pomoću robota izvršio bilo kakav manipulacioni zadatak, potrebno je da se takva struktura opiše sa stanovišta geometrije i kinematike. Kako je mehanička struktura robota, u stvari, jedan zglobni mehanizam, najprije će biti definisani neophodni pojmovi iz teorije mehanizama, a zatim i način određivanja geometrije segmenata robotskog mehanizma. Takođe, s obzirom na to da je jedan od suštinskih zadataka robotskog manipulatora kretanje i dostizanje određenog položaja u prostoru, u ovom su poglavlju izloženi i osnovni stavovi iz kinematike, na osnovu kojih se može jednoznačno opisati takvo kretanje.

2.1. Osnovni stavovi iz teorije mehanizama

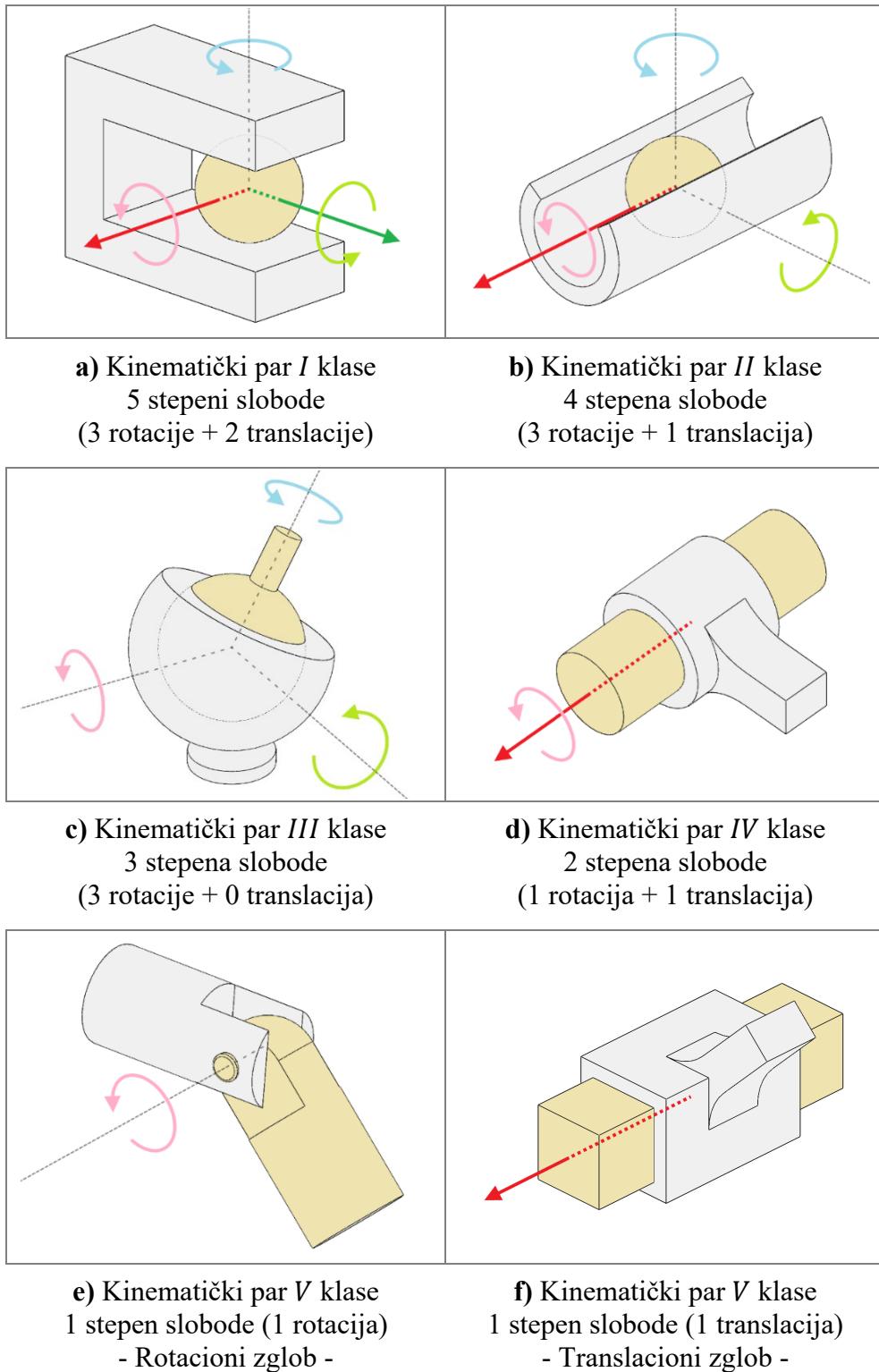
Teorija mehanizama jeste oblast primijenjene nauke i tehnike, koja se bavi proučavanjem sistema međusobno povezanih tijela različitim tipovima veza [1],[2]. Iz ove su oblasti izvučeni oni pojmovi koji su bitni za realizaciju zadatka u sklopu diplomskog rada, a tiču se tipova veza tijela koji sačinjavaju mehanizam robota, kao i strukture jednog takvog mehanizma.

Za svaku tačku u trodimenzionalnom euklidskom prostoru mogu se definisati koordinate x , y i z , koje jednoznačno određuju poziciju tačke u odnosu na neki usvojeni referentni koordinatni sistem. Da bi se u potpunosti odredio položaj tijela u prostoru, nije dovoljno znati samo poziciju određene tačke tijela, jer se tijelo sastoji od beskonačnog broja tačaka. Međutim, ukoliko se definiše koordinatni sistem vezan za tijelo, tada se svaka tačka tijela može opisati konstantnim radijus-vektorom u odnosu na ishodište tog koordinatnog sistema, kao na slici 2.1. Dakle, dovoljno je odrediti položaj vezanog koordinatnog sistema u odnosu na usvojeni referentni koordinatni sistem da bi se odredio i položaj tijela. Za razliku od tačke koja je u potpunosti definisana svojom pozicijom, tijelo, odnosno koordinatni sistem vezan za tijelo može i da rotira. Dakle, potrebno je šest parametara da bi se jednoznačno odredio položaj tijela. Svako moguće nezavisno kretanje tijela nazivamo stepenom slobode tijela. Slobodno tijelo, dakle, ima šest stepeni slobode. Ono što je značajnije u robotici jesu veze više tijela koja čine mehanizam robota.



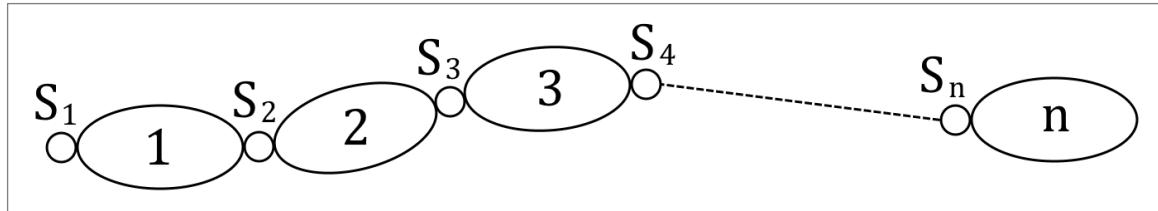
Slika 2.1 – Tijelo i vezani koordinatni sistem

Kinematički par podrazumijeva dva međusobno povezana tijela tako da jedno drugom, u određenoj mjeri, ograničava kretanje. U zavisnosti od načina vezivanja tijela gube određeni broj stepeni slobode, pa se kinematički parovi dijele na različite klase [1],[2]. Primjeri kinematičkih parova prikazani su na slici 2.2. Za primjenu u robotici najvažniji su kinematički parovi pete klase, koji dozvoljavaju samo jednu vrstu kretanja. Takvi su: rotacioni par ili zglob, prikazan na slici 2.2 e) i translacioni par ili zglob, prikazan na slici 2.2 f).



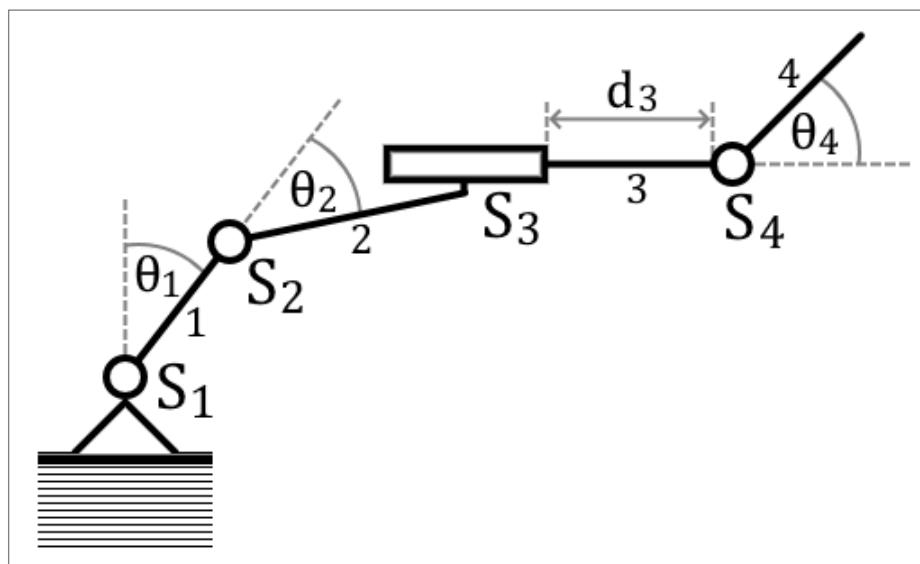
Slika 2.2 – Primjeri kinematičkih parova

Kinematicki lanac predstavlja niz tijela međusobno povezanih kinematickim parovima. U robotici se svako tijelo zove segment lanca, dok su kinematicki parovi najčešće rotacioni i translacioni zglobovi [1],[2]. Na slici 2.3 šematski je prikazan jedan kinematicki lanac, gdje su segmenti predstavljeni elipsama i numerisani sa brojevima 1, 2, 3,..., n , dok su zglobovi predstavljeni kružićima i označeni sa S_1, S_2, \dots, S_n .



Slika 2.3 – Šematski prikaz kinematickog lanca

Broj stepeni slobode kinematickog lanca jednak je broju nezavisnih mogućih kretanja lanca, odnosno broju nezavisnih parametara potrebnih da bi se jednoznačno odredio položaj cijelog lanca [1],[2]. Kinematicki lanci mogu se podijeliti na različite načine, ali je za izradu zadatka u sklopu diplomskog rada bilo dovoljno razmotriti prost otvoren kinematicki lanac sa zglobovima V klase. Prost lanac podrazumijeva samo jedan niz segmenata, pri čemu svaki segment u nizu ima dva zgloba, prethodni i naredni, osim posljednjeg, koji ima jedan zglob. Otvoren lanac jeste onaj kod kojeg segmenti ne formiraju zatvoren niz, to jest, iz jedne tačke lanca u drugu može se doći samo jednom putanjom. Veliki dio industrijskih manipulacionih robova spada u grupu prostih otvorenih lanaca, pa tako i robot Mitsubishi RV-2SDB. S obzirom na to da zglobovi V klase imaju jedan stepen slobode, može se zaključiti da je broj stepeni slobode takvih kinematickih lanaca jednak broju zglobova, te je za određivanje položaja lanca dovoljno znati pomjeraje u svim zglobovima lanca. Primjer prostog otvorenog lanca prikazan je na slici 2.4 Zglobovi S_1, S_2 i S_4 su rotacioni, dok je zglob S_3 translacioni.



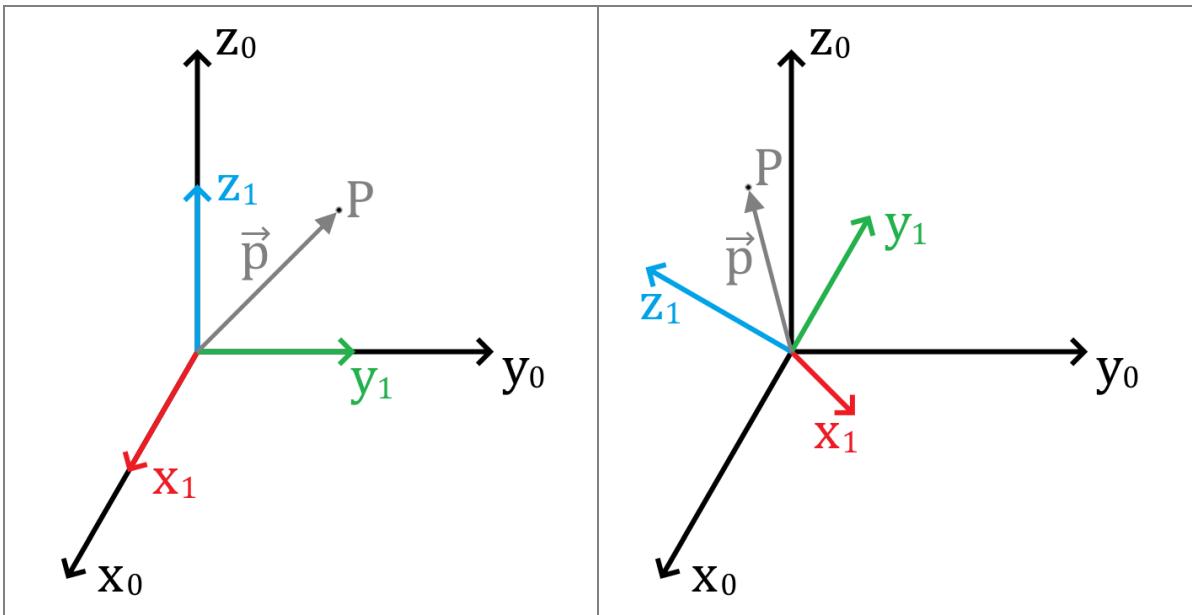
Slika 2.4 – Primjer prostog otvorenog kinematickog lanca

Da bi se opisao kinematicki lanac robova, neophodno je da se definije način vezivanja koordinatnih sistema za segmente, da se odrede relacije između tih koordinatnih sistema, kao i njihove veze sa referentnim (nepokretnim) koordinatnim sistemom [2].

2.2. Homogene transformacije

Prilikom obavljanja različitih manipulacionih zadataka, struktura kinematičkog lanca robota mijenja se uslijed kretanja njegovih segmenata. Geometrijske transformacije koordinatnih sistema vezanih za segmente koje se tada dešavaju, poput translacije i rotacije, zgodno je predstaviti tako da se lako kombinuju pomoću osnovnih matematičkih operacija.

Neka su u trodimenzionalnom euklidskom prostoru dati referentni nepokretni koordinatni sistem $Ox_0y_0z_0$ i sistem $Ox_1y_1z_1$, koji može da vrši rotaciju. Neka su $(\vec{i}_0, \vec{j}_0, \vec{k}_0)$ i $(\vec{i}_1, \vec{j}_1, \vec{k}_1)$ bazni jedinični vektori duž koordinatnih osa odgovarajućih sistema. Početni slučaj, kada se sistemi poklapaju, prikazan je na slici 2.5 a).



a) Početni slučaj – sistemi se poklapaju

b) Nakon rotacije sistema $Ox_1y_1z_1$

Slika 2.5 – Neprekidni i pokretni koordinatni sistemi

Vektor položaja \vec{p} , koji odgovara tački P , može se predstaviti u odnosu na oba koordinatna sistema na sljedeći način [2]:

$$\begin{aligned}\vec{p}_0 &= (\vec{i}_0 \cdot \vec{p}, \vec{j}_0 \cdot \vec{p}, \vec{k}_0 \cdot \vec{p}) = (p_{x_0}, p_{y_0}, p_{z_0}) \\ \vec{p}_1 &= (\vec{i}_1 \cdot \vec{p}, \vec{j}_1 \cdot \vec{p}, \vec{k}_1 \cdot \vec{p}) = (p_{x_1}, p_{y_1}, p_{z_1})\end{aligned}\quad (2.1)$$

Uzima se da je \vec{p} fiksni u odnosu na sistem $Ox_1y_1z_1$, tako da i nakon proizvoljne rotacije sistema $Ox_1y_1z_1$, kao na slici 2.5 b), važi $\vec{p}_1 = \text{const.}$

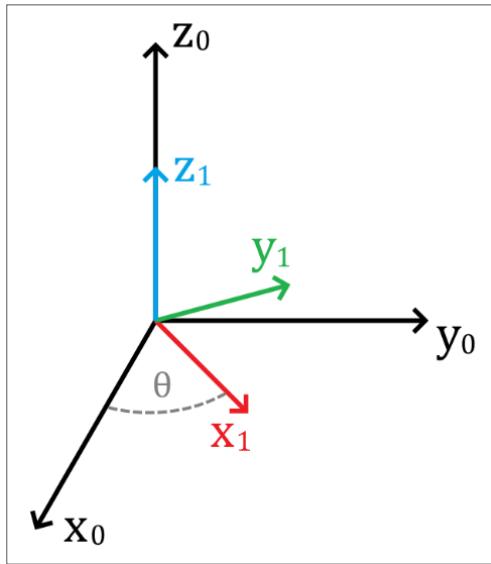
Rotaciona matica R , dimenzija 3×3 , predstavlja matricu koja transformiše koordinate vektora \vec{p} , izražene u odnosu na pokretni sistem $Ox_1y_1z_1$, u koordinate izražene u odnosu na referentni sistem $Ox_0y_0z_0$. Dakle, treba da vrijedi: $\vec{p}_0 = R \cdot \vec{p}_1$. S obzirom na to da su koordinate vektora \vec{p} , u stvari, njegova projekcija na ose odgovarajućeg koordinatnog sistema, prethodne jednačine mogu se ukombinovati na način prikazan u (2.2) [2].

$$\begin{aligned}
\vec{p}_0 &= \begin{bmatrix} p_{x_0} \\ p_{y_0} \\ p_{z_0} \end{bmatrix} = \begin{bmatrix} \vec{i}_0 \cdot \vec{p} \\ \vec{j}_0 \cdot \vec{p} \\ \vec{k}_0 \cdot \vec{p} \end{bmatrix} = \begin{bmatrix} \vec{i}_0 \cdot \vec{i}_1 \cdot p_{x_1} + \vec{i}_0 \cdot \vec{j}_1 \cdot p_{y_1} + \vec{i}_0 \cdot \vec{k}_1 \cdot p_{z_1} \\ \vec{j}_0 \cdot \vec{i}_1 \cdot p_{x_1} + \vec{j}_0 \cdot \vec{j}_1 \cdot p_{y_1} + \vec{j}_0 \cdot \vec{k}_1 \cdot p_{z_1} \\ \vec{k}_0 \cdot \vec{i}_1 \cdot p_{x_1} + \vec{k}_0 \cdot \vec{j}_1 \cdot p_{y_1} + \vec{k}_0 \cdot \vec{k}_1 \cdot p_{z_1} \end{bmatrix} = \\
&= \begin{bmatrix} \vec{i}_0 \cdot \vec{i}_1 & \vec{i}_0 \cdot \vec{j}_1 & \vec{i}_0 \cdot \vec{k}_1 \\ \vec{j}_0 \cdot \vec{i}_1 & \vec{j}_0 \cdot \vec{j}_1 & \vec{j}_0 \cdot \vec{k}_1 \\ \vec{k}_0 \cdot \vec{i}_1 & \vec{k}_0 \cdot \vec{j}_1 & \vec{k}_0 \cdot \vec{k}_1 \end{bmatrix} \cdot \begin{bmatrix} p_{x_1} \\ p_{y_1} \\ p_{z_1} \end{bmatrix} = R \cdot \begin{bmatrix} p_{x_1} \\ p_{y_1} \\ p_{z_1} \end{bmatrix} = R \cdot \vec{p}_1
\end{aligned} \tag{2.2}$$

Na sličan način moguće je transformisati i koordinate vektora izražene u odnosu na sistem $Ox_0y_0z_0$ u koordinate izražene u odnosu na sistem $Ox_1y_1z_1$. Uzimajući u obzir osobinu komutativnosti skalarnog proizvoda vektora, može se pokazati sljedeće [2]:

$$\vec{p}_1 = R^{-1} \cdot \vec{p}_0 = \begin{bmatrix} \vec{i}_1 \cdot \vec{i}_0 & \vec{i}_1 \cdot \vec{j}_0 & \vec{i}_1 \cdot \vec{k}_0 \\ \vec{j}_1 \cdot \vec{i}_0 & \vec{j}_1 \cdot \vec{j}_0 & \vec{j}_1 \cdot \vec{k}_0 \\ \vec{k}_1 \cdot \vec{i}_0 & \vec{k}_1 \cdot \vec{j}_0 & \vec{k}_1 \cdot \vec{k}_0 \end{bmatrix} \cdot \begin{bmatrix} p_{x_0} \\ p_{y_0} \\ p_{z_0} \end{bmatrix} = R^T \cdot \vec{p}_0 \tag{2.3}$$

Za primjenu u robotici zgodno je posmatrati elementarne rotacije, to jest, rotacije oko pojedinačnih osa referentnog koordinatnog sistema. Neka se, na primjer, vrši rotacija oko z_0 ose referentnog sistema, kao na slici 2.6. Poznato je da je skalarni proizvod dva jedinična vektora jednak kosinusu ugla izmedju njih. Može se uočiti da jedinični vektori \vec{k}_0 i \vec{k}_1 pri ovoj rotaciji uvijek ostaju kolinearni, te je njihov skalarni proizvod jednak 1. Takođe, ovi vektori uvijek su ortogonalni sa ostalim jediničnim vektorima, te su njihovi međusobni skalarni proizvodi jednakci 0.



Slika 2.6 – Elementarna rotacija oko z_0 ose

Slično se može uočiti i za elementarne rotacije oko x_0 i y_0 osa referentnog sistema. Elementarne rotacije rezultiraju osnovnim (elementarnim) matricama rotacije, prikazanim u (2.4). Sve se ostale rotacije mogu dobiti uzastopnim elementarnim rotacijama, odnosno množenjem osnovnih matrica rotacije. Pritom je potrebno voditi računa o redoslijedu množenja, odnosno redoslijedu izvršenih rotacija, jer množenje matrica nije komutativno.

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad R_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.4)$$

$$R_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Da bi se opisale i ostale geometrijske transformacije kao što su translacija, transformacija perspektive i skaliranje potrebno je proširiti pojам vektora i rotacione matrice. Pored tri prostorne koordinate koje određuju vektor položaja u trodimenzionalnom prostoru, uvodi se i četvrta komponenta – ω , koja predstavlja faktor skaliranja prostornih koordinata, tako da je: $\vec{\hat{p}} = (\omega p_x, \omega p_y, \omega p_z, \omega)$. Kaže se da je vektor $\vec{\hat{p}}$ izražen u homogenim koordinatama [2]. Za faktor skaliranja uglavnom se uzima vrijednost 1, tako da homogene koordinate vektora odgovaraju njegovim fizičkim koordinatama.

Homogena prelazna (transformaciona) matrica T , dimenzija 4×4 , predstavlja matricu koja transformiše homogene koordinate vektora $\vec{\hat{p}}$, izražene u odnosu na pokretni sistem $Ox_1y_1z_1$, u homogene koordinate izražene u odnosu na referentni sistem $Ox_0y_0z_0$, tako da vrijedi: $\vec{\hat{p}}_0 = T \cdot \vec{\hat{p}}_1$. Sastoji se iz četiri submatrice [2]:

$$T = \begin{bmatrix} R_{3x3} & p_{3x1} \\ f_{1x3} & \omega_{1x1} \end{bmatrix} = \begin{bmatrix} \text{matrica rotacije} & \text{vektor translacije} \\ \text{transformacija perspektive} & \text{faktor skaliranja} \end{bmatrix} \quad (2.5)$$

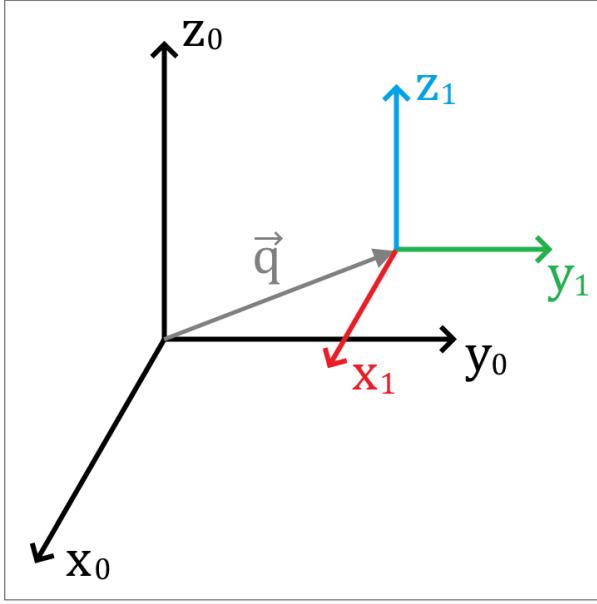
Gornja lijeva submatrica R_{3x3} homogene prelazne matrice ekvivalentna je prethodno definisanim rotacionim matricama, te predstavlja orijentaciju pokretnog sistema u odnosu na referentni sistem. Analogno osnovnim matricama rotacije definišu se i osnovne homogene prelazne matrice rotacije:

$$T_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$$T_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Gornja desna submatrica p_{3x1} homogene prelazne matrice odgovara translaciji pokretnog koordinatnog sistema paralelno osama referentnog sistema. Ukoliko je sistem $Ox_1y_1z_1$ isključivo transliran u odnosu na sistem $Ox_0y_0z_0$ za proizvoljan vektor $\vec{q} = (q_x, q_y, q_z)$, kao na slici 2.7, tada se homogena matrica koja opisuje takvu transformaciju naziva osnovna (elementarna) homogena matrica translacije:

$$T_{tran} = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$



Slika 2.7 – Translacija pokretnog sistema za proizvoljan vektor

Donja lijeva submatrica f_{1x3} homogene prelazne matrice odgovara transformaciji perspektive i pogodna je za vizuelne sisteme i kalibraciju modela kamere. Ukoliko se posmatra nevizueleni sistem kao što je i sistem automatskog pisanja pomoću robota u sklopu diplomskog rada vrijednosti ovog vektora jednaki su 0, tako da se uvijek radi o nultoj transformaciji perspektive.

Donja desna submatrica ω_{1x1} homogene prelazne matrice obezbjeduje globalno skaliranje fizičkih koordinata vektora. Na primjer, ukoliko je potrebno skalirati vektor \hat{p} za određeni faktor s , tada je [2]:

$$\hat{p}_s = T_s \cdot \hat{p} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ s \end{bmatrix} \quad (2.8)$$

Fizičke koordinate vektora \hat{p}_s tada su: $p_{s_x} = \frac{p_x}{s}$, $p_{s_y} = \frac{p_y}{s}$ i $p_{s_z} = \frac{p_z}{s}$. Vektor \hat{p}_s smanjen je u odnosu na vektor \hat{p} , ukoliko je $s > 1$, odnosno povećan je, ukoliko je $0 < s < 1$.

Osnovne homogene matrice rotacije i translacije mogu se međusobno množiti, tako da predstavljaju niz rotacija i translacija pokretnog sistema u odnosu na referentni koordinatni sistem. Tako dobijena matrica naziva se rezultantna (kompozitna) homogena prelazna matrica. Trebalo bi da se pridržava sljedećih pravila radi lakšeg nalaženja rezultantne homogene prelazne matrice [2]:

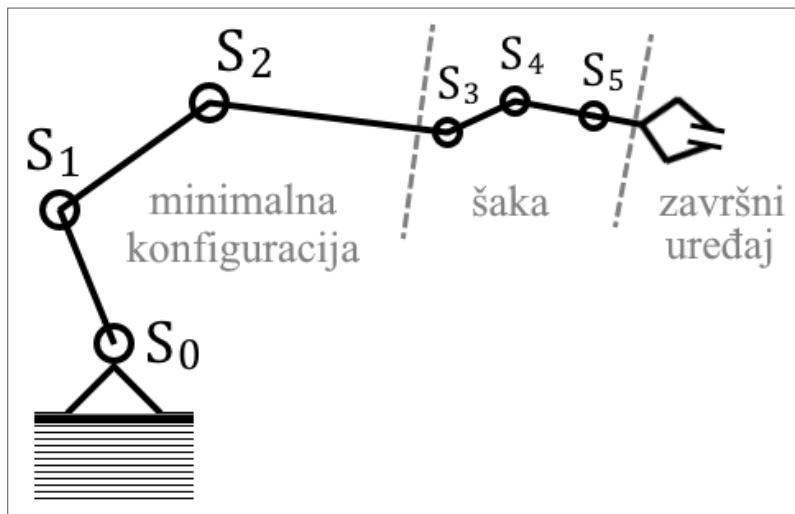
1. Početno stanje karakteriše poklapanje pokretnog i referentnog koordinatnog sistema, pa je homogena prelazna matrica, u stvari, jedinična matrica:

$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

2. Ako pokretni koordinatni sistem vrši rotaciju/translaciјu oko osa referentnog koordinatnog sistema, tada se prethodna rezultantna homogena prelazna matrica množi slijeva sa odgovarajućom osnovnom homogenom matricom rotacije/translaciјe.
3. Ako pokretni koordinatni sistem vrši rotaciju/translaciјu oko vlastitih osa, tada se prethodna rezultantna homogena prelazna matrica množi zdesna sa odgovarajućom osnovnom homogenom matricom rotacije/translaciјe.

2.3. Geometrija manipulacionih robota

Mehanizam industrijskog manipulacionog robota obično se sastoji iz dva međusobno povezana dijela: minimalne konfiguracije (ruke) i završnog mehanizma (šake) robota. Minimalna konfiguracija jeste mehanizam sa tri zglobo, od kojih svaki može biti rotacioni ili translacioni. Zadatak minimalne konfiguracije jeste da vrši pozicioniranje šake, odnosno da dovede šaku robota na željeno mjesto u radnom prostoru. Mehanizam šake ima jedan, dva ili tri zglobo, u zavisnosti od potrebnog broja stepeni slobode manipulatora. Zglobovi šake po pravilu su rotacioni. Na vrh mehanizma šake učvršćuje se završni uređaj (alat), koji neposredno izvršava zadatok koji je postavljen robotu. Mehanizam šake zajedno sa minimalnom konfiguracijom obezbeđuje da vrh alata dođe u željenu poziciju i da alat zauzme odgovarajuću orientaciju u prostoru [1],[2].

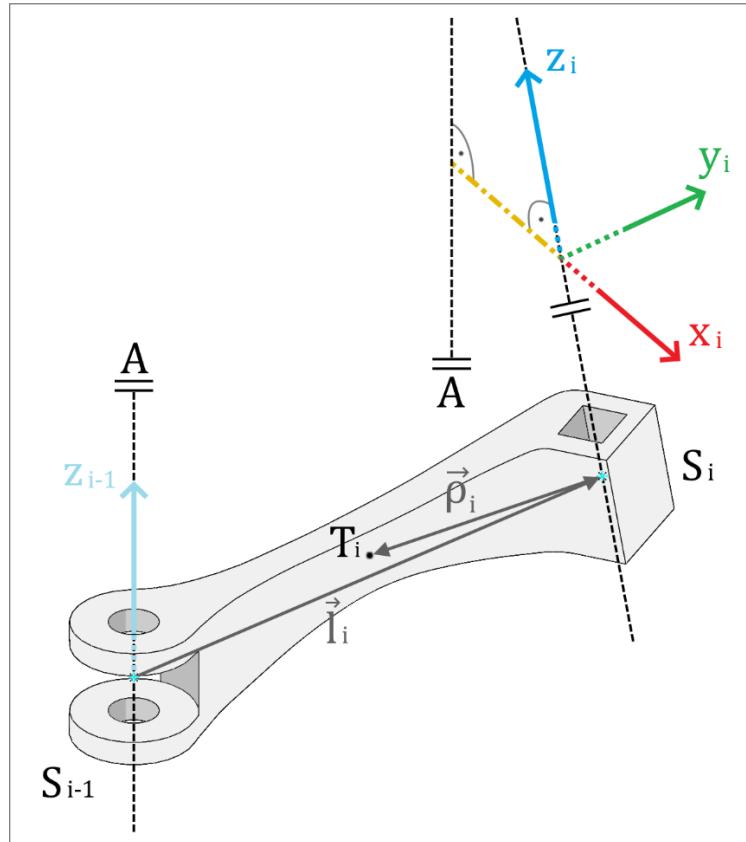


Slika 2.8 – Minimalna konfiguracija i šaka robota sa završnim uređajem

2.3.1. Geometrija segmenata

Da bi se mogao odrediti položaj kinematičkog lanca robota i njegovog završnog uređaja, potrebno je prvo opisati geometriju svih segmenata kinematičkog lanca. Postoje različiti pristupi za zadavanje geometrije segmenata, kao što su: pristup Jurija Stepanjenka, Denavit–Hartenbergov pristup i slično. Ovi se pristupi razlikuju prvenstveno po načinu postavljanja vezanih koordinatnih sistema i shodno tome načinu izražavanja vektora geometrije. Pristup Jurija Stepanjenka zasnovan je na proizvolnjom postavljanju osa vezanog koordinatnog sistema. Ovo je često značajno u dinamičkoj analizi robotskog manipulatora, gdje se ose vezanog sistema mogu postaviti u pravcima glavnih osa inercije segmenta, pa se njihove inercijalne osobine lakše izražavaju i lakše koriste u proračunima [1],[2]. Denavit–

Hartenbergov pristup manje je opšti od prethodnog. Postavljanje osa vezanog koordinatnog sistema nije proizvoljno, već je uslovljeno samom geometrijom segmenata ili, preciznije, položajem osa djelovanja zglobova. Međutim, taj pristup omogućava jednostavnije i brže izračunavanje veza jednog i drugog koordinatnog sistema, što je pogodno u kinematičkoj analizi robota [1],[2]. Stoga se u sklopu diplomskog rada koristi Denavit–Hartenbergov pristup, te je detaljnije objašnjen u nastavku.



Slika 2.9 – Postavljanje vezanog koordinatnog sistema prema DH pristupu

Denavit–Hartenbergov (DH) pristup formulisali su J. Denavit i R. S. Hartenberg uvodeći koncept zajedničke normale između dvije linije u cilju postizanja minimalne reprezentacije geometrije segmenta [3]. Zglobovi kinematičkog lanca obično se numerišu od nultog, pa je za i -ti segment prethodni zglob S_{i-1} , a naredni S_i [1],[2]. Koordinatni sistem $Ox_iy_iz_i$ (i -ti koordinatni sistem), vezan za i -ti segment, smješta se na osnovu sljedećih pravila [4]:

1. Osa z_i u smjeru je ose djelovanja zgloba S_i .
2. Osa x_i kolinearna je sa zajedničkom normalom osa z_i i z_{i-1} , koja je ujedno i najkraća linija između njih. Ishodište i -tog koordinatnog sistema leži na presjeku osa z_i i x_i .
3. Osa y_i normalna je na ose z_i i x_i , tako da se dobije desni koordinatni sistem.

Za nulti (referentni) koordinatni sistem važe 1. i 3. pravilo, a osa x_0 postavlja se proizvoljno. Geometrija segmenta još je određena vektorima $\vec{\rho}_i = \overline{S_i T_i}$ i $\vec{l}_i = \overline{S_{i-1} S_i}$. Vektor $\vec{\rho}_i$ definiše položaj težišta T_i u odnosu na zglob S_i , dok \vec{l}_i određuje položaj prethodnog zgloba

u odnosu na naredni zglob [1],[2]. Na slici 2.9 prikazan je i -ti segment i vezani koordinatni sistem $Ox_iy_iz_i$, postavljen prema opisanim pravilima. Može se primijetiti da je prethodni koordinatni sistem samo djelimično prikazan, jer je za postavljanje i -tog koordinatnog sistema bitna samo z_{i-1} osa prethodnog koordinatnog sistema, odnosno osa djelovanja prethodnog zgloba. Bitno je napomenuti da je u ovom primjeru ishodište koordinatnog sistema $i - 1$ proizvoljno smješteno u centar zgloba, dok u realnom slučaju, ukoliko se ne radi o referentnom sistemu, pozicija tog ishodišta zavisi od sistema $i - 2$. Takođe, zbog specifičnosti prikazanog segmenta i položaja zglobova i -ti koordinatni sistem leži mnogo iznad samog segmenta, što je demonstrirano presjekom osa zglobova.

Nakon što se koordinatni sistemi postave na opisani način, geometrija svakog segmenta prema DH pristupu može se odrediti sa četiri parametra: θ , d , a i α . Prelaz iz vezanog koordinatnog sistema $i - 1$ u vezani koordinatni sistem i može se provesti u četiri faze [2]:

1. Rotacija oko ose z_{i-1} za ugao θ_i dok osa x_{i-1} ne postane paralelna sa x_i ;
2. Translacija duž ose z_{i-1} za dužinu d_i dok se osa x_{i-1} ne poklopi sa x_i ;
3. Translacija duž ose x_i za dužinu a_i dok se O_{i-1} ne poklopi sa O_i ;
4. Rotacija oko ose x_i za ugao α_i dok se dva sistema sasvim ne poklope.

Za rotacioni zglob parametri d_i , a_i i α_i imaju konstantnu vrijednost, dok se θ_i mijenja kada se segment i pomjera u odnosu na segment $i - 1$. Za translacioni zglob θ_i , a_i i α_i jesu parametri koji ostaju konstantni, dok je d_i promjenljiva zgloba [2].

Svakoj fazi transformacije sistema $i-1$ u sistem i odgovara osnovna homogena matrica rotacije/translacije. Proizvod ove četiri osnovne prelazne matrice daje resultantnu homogenu prelaznu matricu $T_{i-1,i}$, poznatu kao DH prelazna matrica za susjedne koordinatne sisteme i i $i-1$, prikazanu u (2.10) [2]. Funkcije sin i cos napisane su u skraćenom obliku (kao S i C).

$$T_{i-1,i} = T_{z,\theta} \cdot T_{z,d} \cdot T_{x,a} \cdot T_{x,\alpha} = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Pored segmenata definišu se i zglobovi kinematičkog lanca. Uvodi se dodatni parametar s_i koji određuje tip i -tog zgloba i to [1],[2]:

$$s_i = \begin{cases} 0, & \text{ako je zglob rotacioni} \\ 1, & \text{ako je zglob translacioni} \end{cases} \quad (2.11)$$

Sada se promjenljivi parametri zglobova, odnosno pomjeraji u zglobovima mogu predstaviti pomoću generalisanih promjenljivih q_i , koje se još nazivaju i unutrašnjim koordinatama robota, na sljedeći način [4]:

$$q_i = \bar{s}_i \theta_i + s_i d_i \quad (2.12)$$

Položaj robota u prostoru zglobova određen je vektorom unutrašnjih koordinata [1],[2]:

$$q = [q_0, q_1, q_2, \dots, q_{n-1}]^T \quad (2.13)$$

Koordinate proizvoljnog vektora, na osnovu prethodnog, moguće je prebacivati iz jednog vezanog koordinatnog sistema u drugi, koristeći lančani proizvod DH prelaznih matrica za susjedne sisteme:

$$\vec{p}_{i-k} = T_{i-k,i} \cdot \vec{p}_i \quad k \in \mathbb{N} \quad (2.14)$$

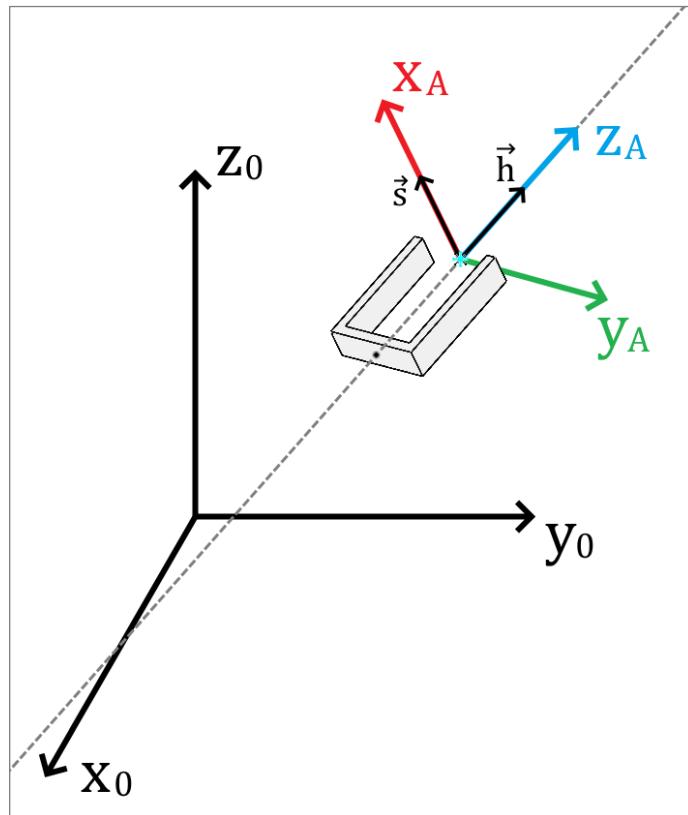
$$T_{i-k,i} = T_{i-k,i-k+1} \cdot T_{i-k+1,i-k+2} \cdots T_{i-2,i-1} \cdot T_{i-1,i} \quad k \in \mathbb{N}$$

Moguće je vršiti i obrnutu transformaciju koordinata:

$$\vec{p}_i = T_{i,i-k} \cdot \vec{p}_{i-k} = T_{i-k,i}^{-1} \cdot \vec{p}_{i-k} \quad k \in \mathbb{N} \quad (2.15)$$

2.3.2. Položaj završnog uređaja

Da bi se odredio položaj alata pričvršćenog za mehanizam robota, potrebno je definisati dodatni koordinatni sistem $Ox_Ay_Az_A$, koji se veže za vrh alata. Pod pojmom vrh alata smatra se ona tačka završnog uređaja čije je pozicioniranje bitno u konkretnom manipulacionom zadatku [1]. Kao i slobodno tijelo i alat može da ima najviše šest stepeni slobode: tri translacije i tri rotacije. Da bi se u potpunosti odredio položaj alata, potrebne su minimalno tri veličine za definisanje pozicije i tri veličine koje će odrediti orijentaciju alata. Položaj alata određuje se u odnosu na spoljašnji koordinatni sistem, koji se u većini slučajeva poistovjećuje sa nultim (referentnim) vezanim koordinatnim sistemom kinematičkog lanca robota.



Slika 2.10 – Koordinatni sistem alata

Pozicija alata određuje se na osnovu ishodišta vezanog koordinatnog sistema $Ox_Ay_Az_A$, te je jednoznačno definisana sa tri Dekartove koordinate: x_A , y_A i z_A [1],[2].

Za definisanje orijentacije alata pogodno je koristiti uzdužnu osu alata. Uvode se dva jedinična vektori: vektor \vec{h} , koji se postavlja u pravcu uzdužne ose alata, i vektor \vec{s} , koji je okomit na vektor \vec{h} i u pravcu karakterističnih geometrijskih obilježja alata [1],[2]. Ukoliko se kao alat uzme hvataljka, kao na slici 2.10, vektor \vec{s} se može postaviti tako da bude okomit na ravan u kojoj leže prsti hvataljke. Orijentacija alata u potpunosti je definisana s ova dva vektora.

Sistem $Ox_Ay_Az_A$ postavlja se tako da je osa z_A u smjeru vektora \vec{h} , osa x_A u smjeru vektora \vec{s} , a osa y_A , zajedno sa ostale dvije ose, gradi desni koordinatni sistem [1],[2].

Orijentacija alata sada se može odrediti na osnovu tri ugla: θ , ϕ i ψ . Postoje različiti načini za definisanje ovih uglova, a najčešće su u upotrebi Ojlerovi uglovi i aeronautički uglovi. Ovi uglovi predstavljaju elementarne rotacije pokretnog koordinatnog sistema u odnosu na sopstvene ose ili na ose referentnog sistema. U tabeli 1. prikazane su različite kombinacije elementarnih rotacija [5] koje odgovaraju uglovima θ , ϕ i ψ , grupisane na osnovu tipa i redoslijeda rotacija. Za pokretni koordinatni sistem uzima se sistem $Ox_Ay_Az_A$.

	Rotacije pokretnog sistema oko sopstvenih osa	Rotacije pokretnog sistema oko osa referentnog sistema
Ojlerovi uglovi	$z_A(\theta) \rightarrow y'_A(\phi) \rightarrow z''_A(\psi)$	\Leftrightarrow $z_0(\psi) \rightarrow y_0(\phi) \rightarrow z_0(\theta)$
	$z_A(\theta) \rightarrow x'_A(\phi) \rightarrow z''_A(\psi)$	\Leftrightarrow $z_0(\psi) \rightarrow x_0(\phi) \rightarrow z_0(\theta)$
	$y_A(\theta) \rightarrow z'_A(\phi) \rightarrow y''_A(\psi)$	\Leftrightarrow $y_0(\psi) \rightarrow z_0(\phi) \rightarrow y_0(\theta)$
	$y_A(\theta) \rightarrow x'_A(\phi) \rightarrow y''_A(\psi)$	\Leftrightarrow $y_0(\psi) \rightarrow x_0(\phi) \rightarrow y_0(\theta)$
	$x_A(\theta) \rightarrow z'_A(\phi) \rightarrow x''_A(\psi)$	\Leftrightarrow $x_0(\psi) \rightarrow z_0(\phi) \rightarrow x_0(\theta)$
	$x_A(\theta) \rightarrow y'_A(\phi) \rightarrow x''_A(\psi)$	\Leftrightarrow $x_0(\psi) \rightarrow y_0(\phi) \rightarrow x_0(\theta)$
Aeronautički uglovi	$z_A(\theta) \rightarrow y'_A(\phi) \rightarrow x''_A(\psi)$	\Leftrightarrow $x_0(\psi) \rightarrow y_0(\phi) \rightarrow z_0(\theta)$
	$z_A(\theta) \rightarrow x'_A(\phi) \rightarrow y''_A(\psi)$	\Leftrightarrow $y_0(\psi) \rightarrow x_0(\phi) \rightarrow z_0(\theta)$
	$y_A(\theta) \rightarrow z'_A(\phi) \rightarrow x''_A(\psi)$	\Leftrightarrow $x_0(\psi) \rightarrow z_0(\phi) \rightarrow y_0(\theta)$
	$y_A(\theta) \rightarrow x'_A(\phi) \rightarrow z''_A(\psi)$	\Leftrightarrow $z_0(\psi) \rightarrow x_0(\phi) \rightarrow y_0(\theta)$
	$x_A(\theta) \rightarrow z'_A(\phi) \rightarrow y''_A(\psi)$	\Leftrightarrow $y_0(\psi) \rightarrow z_0(\phi) \rightarrow x_0(\theta)$
	$x_A(\theta) \rightarrow y'_A(\phi) \rightarrow z''_A(\psi)$	\Leftrightarrow $z_0(\psi) \rightarrow y_0(\phi) \rightarrow x_0(\theta)$

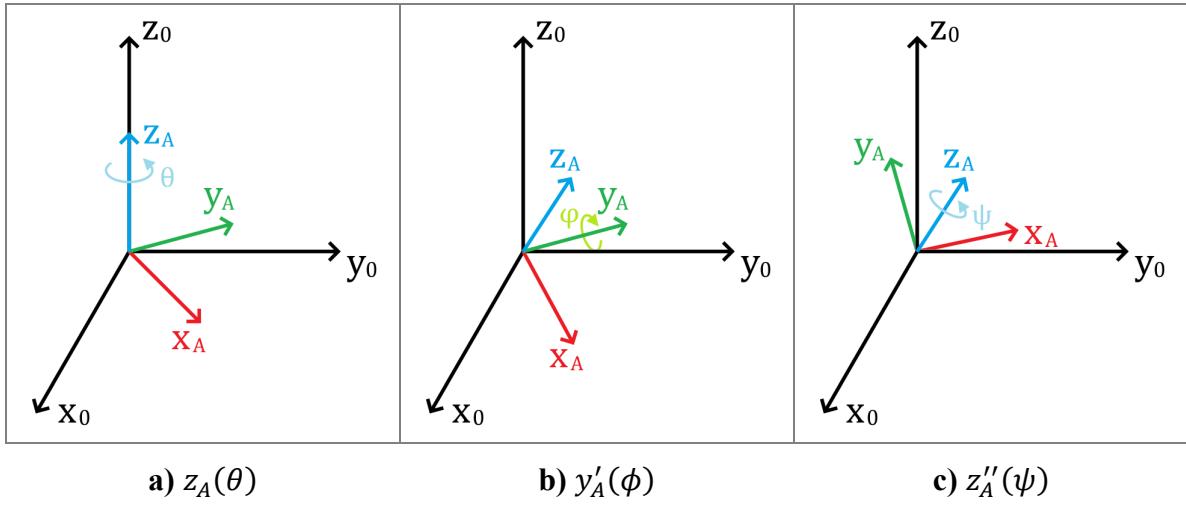
Tabela 1 – Ojlerovi i aeronautički uglovi

Na primjer, neka pokretni sistem rotira na osnovu prvog tipa Ojlerovih uglova:

$$z_A(\theta) \rightarrow y'_A(\phi) \rightarrow z''_A(\psi) \Leftrightarrow z_0(\psi) \rightarrow y_0(\phi) \rightarrow z_0(\theta) \quad (2.16)$$

Ova se varijanta skraćeno naziva i *zzyz* varijanta Ojlerovih uglova. Kao što je prikazano u (2.16), redoslijed rotacija pokretnog sistema $Ox_Ay_Az_A$ oko sopstvenih osa jeste:

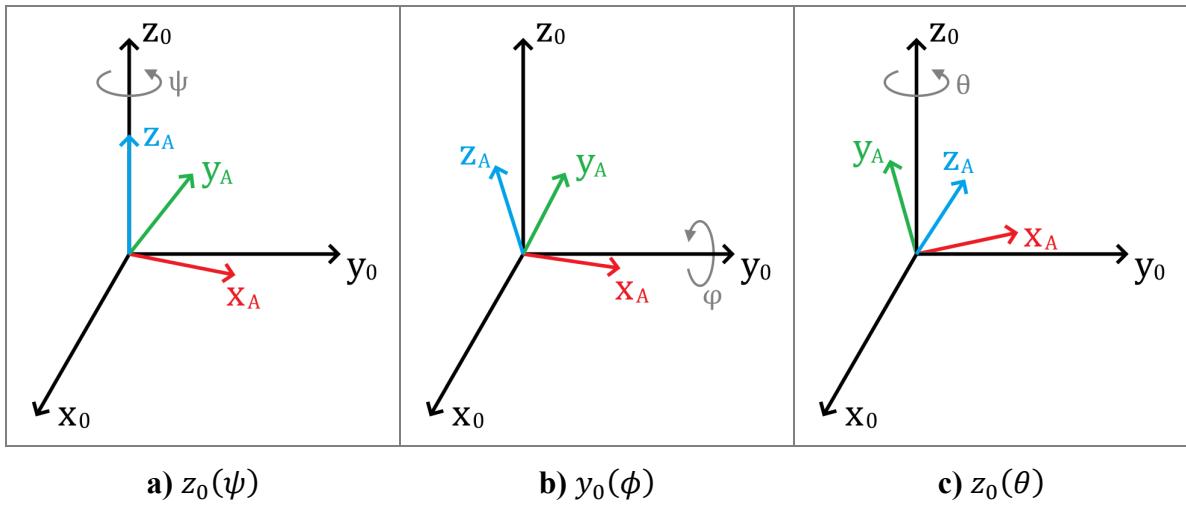
1. Rotacija oko z_A ose za ugao θ (slika 2.11 a));
2. Rotacija oko nove y_A ose za ugao ϕ (slika 2.11 b));
3. Rotacija oko nove z_A ose za ugao ψ (slika 2.11 c)).



Slika 2.11 – Ojlerova zyz rotacija pokretnog sistema oko sopstvenih osa

Identična orientacija sistema $Ox_Ay_Az_A$, kako je i naznačeno u (2.16), može se postići rotacijama oko osa referentnog sistema u obrnutom redoslijedu:

1. Rotacija oko z_0 ose za ugao ψ (slika 2.12 a));
2. Rotacija oko y_0 ose za ugao ϕ (slika 2.12 b));
3. Rotacija oko z_0 ose za ugao θ (slika 2.12 c)).



Slika 2.12 – Ojlerova zyz rotacija pokretnog sistema oko osa referentnog sistema

Rotaciona matrica koja odgovara z - y - z rotacijama lako se dobija uzastopnim množenjem osnovnih matrica rotacije [6]:

$$\begin{aligned}
 R &= R_{z_A}(\theta) \cdot R_{y'_A}(\phi) \cdot R_{z''_A}(\psi) = R_{z_0}(\theta) \cdot R_{y_0}(\phi) \cdot R_{z_0}(\psi) = \\
 &= \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{bmatrix} \cdot \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} C\phi C\psi C\theta - S\psi S\theta & -C\psi S\theta - C\phi S\psi C\theta & S\phi C\theta \\ C\phi C\psi S\theta + S\psi C\theta & C\psi C\theta - C\phi S\psi S\theta & S\phi S\theta \\ -C\psi S\phi & S\psi S\phi & C\phi \end{bmatrix}
 \end{aligned} \tag{2.17}$$

Rješavanje inverznog problema, to jest, određivanje Ojlerovih uglova koji odgovaraju dатој матрици ротације nije jednoznačно. Главни разлог за то јесте парност косинусне функције. С друге стране, кодомени инверзних тригонометријских функција не покривају цијелу тригонометријску кружницу. Да би се donekle заобиша та ограничења, згодно је увести функцију два аргумента, $\text{atan2}(y, x)$, која има опсег vrijednosti $(-\pi, \pi]$. Специфичност ове функције у односу на функцију $\text{atan}\left(\frac{y}{x}\right)$ јесте да што није дозволено директно скраћивање аргумента y и x , jer се предзнаки оба аргумента користе за одређивање квадранта којем угao припада. Функција $\text{atan2}(y, x)$, узимајући у обзир и граничне slučajeve, дефинише се на следећи начин [7]:

$$\text{atan2}(y, x) = \begin{cases} \text{atan}\left(\frac{y}{x}\right), & x > 0 \\ \text{atan}\left(\frac{y}{x}\right) + \pi, & y \geq 0, x < 0 \\ \text{atan}\left(\frac{y}{x}\right) - \pi, & y < 0, x < 0 \\ \frac{\pi}{2}, & y > 0, x = 0 \\ -\frac{\pi}{2}, & y < 0, x = 0 \\ \text{nedefinisano}, & y = 0, x = 0 \end{cases} \tag{2.18}$$

Vrijednosti z - y - z Ojlerovih uglova сада је могуће одредити на следећи начин [8]:

$$1. \text{ Ako je } |R_{33}| \neq 1 \rightarrow \cos \phi \neq \pm 1, \sin \phi \neq 0 \tag{2.19}$$

- постоје два решења: $(\theta_1, \phi_1, \psi_1)$ и $(\theta_2, \phi_2, \psi_2)$

$$\phi_{1,2} = \pm \text{acos}(R_{33})$$

$$\theta_{1,2} = \text{atan2}\left(\frac{R_{23}}{\sin \phi_{1,2}}, \frac{R_{13}}{\sin \phi_{1,2}}\right)$$

$$\psi_{1,2} = \text{atan2}\left(\frac{R_{32}}{\sin \phi_{1,2}}, -\frac{R_{31}}{\sin \phi_{1,2}}\right)$$

$$2. \text{ Ako je } |R_{33}| = 1$$

- nastaje привидни singularitet, углови θ и ψ нису јединствено одређени, постоји бескonačno mnogo решења.

a) Ako je $R_{33} = 1 \rightarrow \cos \phi = 1, \sin \phi = 0 \rightarrow \phi = 0$ (2.20)

$$R_{21} = \sin \psi \cos \theta + \cos \psi \sin \theta = \sin(\psi + \theta)$$

$$R_{22} = \cos \psi \cos \theta - \sin \psi \sin \theta = \cos(\psi + \theta)$$

Slijedi:

$$\psi + \theta = \text{atan2}(R_{21}, R_{22})$$

Proizvoljno se može uzeti $\theta = 0$, pa je:

$$\psi = \text{atan2}(R_{21}, R_{22})$$

b) Ako je $R_{33} = -1 \rightarrow \cos \phi = -1, \sin \phi = 0 \rightarrow \phi = \pi$ (2.21)

$$R_{21} = \sin \psi \cos \theta - \cos \psi \sin \theta = \sin(\psi - \theta)$$

$$R_{22} = \cos \psi \cos \theta + \sin \psi \sin \theta = \cos(\psi - \theta)$$

Slijedi:

$$\psi - \theta = \text{atan2}(R_{21}, R_{22})$$

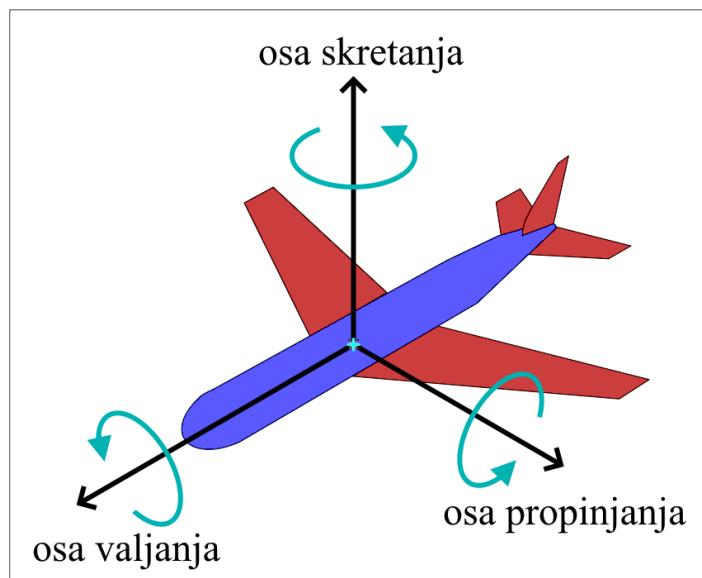
Proizvoljno se može uzeti $\theta = 0$, pa je:

$$\psi = \text{atan2}(R_{21}, R_{22})$$

Neka sada pokretni koordinatni sistem rotira na osnovu prvog tipa aeronautičkih uglova iz tabele 1:

$$z_A(\theta) \rightarrow y'_A(\phi) \rightarrow x''_A(\psi) \Leftrightarrow x_0(\psi) \rightarrow y_0(\phi) \rightarrow z_0(\theta) \quad (2.22)$$

Ova se varijanta skraćeno naziva i *xyz* varijanta aeronautičkih uglova. Zbog specifičnosti ovih rotacija, ugao θ obično se naziva ugao valjanja, ϕ ugao propinjanja, a ψ ugao skretanja [1],[2]. Ovi nazivi potiču iz aeronautike, te otuda i naziv aeronautički uglovi. Na slici 2.13 na primjeru aviona prikazane su ove vrste rotacija.



Slika 2.13 – Prikaz aeronautičkih rotacija

Rotaciona matrica koja odgovara ovoj vrsti rotacija jeste [6]:

$$\begin{aligned}
 R &= R_{z_A}(\theta) \cdot R_{y'_A}(\phi) \cdot R_{x''_A}(\psi) = R_{z_0}(\theta) \cdot R_{y_0}(\phi) \cdot R_{x_0}(\psi) = \\
 &\begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\psi & -S\psi \\ 0 & S\psi & C\psi \end{bmatrix} = \\
 &\begin{bmatrix} C\phi C\theta & S\phi S\psi C\theta - C\psi S\theta & S\phi C\psi C\theta + S\psi S\theta \\ C\phi S\theta & S\phi S\psi S\theta + C\psi C\theta & S\phi C\psi S\theta - S\psi C\theta \\ -S\phi & C\phi S\psi & C\phi C\psi \end{bmatrix} \tag{2.23}
 \end{aligned}$$

Slično kao i kod Ojlerovih uglova, nije moguće jednoznačno odrediti aeronautičke uglove koji odgovaraju datoj matrici rotacije. U ovom slučaju glavni razlog za to jeste neparnost sinusne funkcije, kao i ograničenost kodomena inverznih trigonometrijskih funkcija. Međutim, koristeći funkciju atan2 moguće je nejednoznačnost rješenja svesti na minimum.

Vrijednosti aeronautičkih uglova mogu se odrediti na sljedeći način [8]:

1. Ako je $|R_{31}| \neq 1 \rightarrow \sin \phi \neq \pm 1, \cos \phi \neq 0$ (2.24)
- postoje dva rješenja: $(\theta_1, \phi_1, \psi_1)$ i $(\theta_2, \phi_2, \psi_2)$

$$\begin{aligned}
 \phi_1 &= -\text{asin}(R_{31}), \quad \phi_2 = \pi + \text{asin}(R_{31}) \\
 \theta_{1,2} &= \text{atan2}\left(\frac{R_{21}}{\cos \phi_{1,2}}, \frac{R_{11}}{\cos \phi_{1,2}}\right) \\
 \psi_{1,2} &= \text{atan2}\left(\frac{R_{32}}{\cos \phi_{12}}, \frac{R_{33}}{\cos \phi_{12}}\right)
 \end{aligned}$$

2. Ako je $|R_{31}| = 1$
- nastaje prividni singularitet, uglovi θ i ψ nisu jedinstveno određeni, postoji beskonačno mnogo rješenja.

- a) Ako je $R_{31} = 1 \rightarrow \sin \phi = -1, \cos \phi = 0 \rightarrow \phi = -\frac{\pi}{2}$ (2.25)

$$R_{22} = \cos \psi \cos \theta + \sin \psi \sin \theta = \cos(\psi - \theta)$$

$$R_{23} = \cos \psi \sin \theta - \cos \theta \sin \psi = -\sin(\psi - \theta)$$

Slijedi:

$$\psi - \theta = \text{atan2}(-R_{23}, R_{22})$$

Proizvoljno se može uzeti $\theta = 0$, pa je:

$$\psi = \text{atan2}(-R_{23}, R_{22})$$

- b) Ako je $R_{31} = -1 \rightarrow \sin \phi = 1, \cos \phi = 0 \rightarrow \phi = \frac{\pi}{2}$ (2.26)

$$R_{22} = \cos \psi \cos \theta - \sin \psi \sin \theta = \cos(\psi + \theta)$$

$$R_{23} = -\cos \psi \sin \theta - \cos \theta \sin \psi = -\sin(\psi + \theta)$$

Slijedi:

$$\psi + \theta = \text{atan}2(-R_{23}, R_{22})$$

Proizvoljno se može uzeti $\theta = 0$, pa je:

$$\psi = \text{atan}2(-R_{23}, R_{22})$$

Dakle, položaj završnog uređaja sa šest stepeni slobode, u odnosu na spoljašnji referentni koordinatni sistem, određen je vektorom koordinata:

$$X = [x_A, y_A, z_A, \theta, \phi, \psi]^T \quad (2.27)$$

Analogno (2.13), koordinate vektora X nazivaju se vanjskim koordinatama robota.

2.4. Kinematika manipulacionih robota

Obavljanje određenog manipulacionog zadatka podrazumijeva kretanje alata i segmenata manipulatora u prostoru i vremenu. Ako se u obzir uzme samo geometrija manipulatora, tada je moguće izračunati samo položaj alata za svaku poznatu vrijednost unutrašnjih koordinata, ali ne i trenutke u kojima će ti položaji biti ostvareni. Takođe, brzinu kretanja ne bi bilo moguće zadavati. Kao posljedica svega navedenog bilo bi kretanje manipulatora sa trzajima ili s povećanom greškom pri praćenju zadatog niza tačaka. Zato je nužno da se mehanizam robota posmatra i sa stanovišta kinematike.

2.4.1. Direktni problem kinematike

Položaj manipulatora u stanju mirovanja određen je generalisanim koordinatama u svim zglobovima: $q_0, q_1, q_2, \dots, q_{n-1}$. Na osnovu (2.14) moguće je formirati prelaznu matricu $T_{0,6}$, koja daje poziciju i orientaciju alata robota u zavisnosti od vrijednosti unutrašnjih koordinata:

$$T_{0,6} = T_{0,6}(q) = \begin{bmatrix} R(q) & p(q) \\ 0^T & 1 \end{bmatrix} \quad (2.28)$$

Unutrašnje koordinate robota, prilikom kretanja manipulatora, mijenjaju se u vremenu: $q_0(t), q_1(t), q_2(t), \dots, q_{n-1}(t)$. Unutrašnje brzine predstavljaju prve izvode po vremenu unutrašnjih koordinata: $\dot{q}_0(t), \dot{q}_1(t), \dot{q}_2(t), \dots, \dot{q}_{n-1}(t)$. Na sličan način definišu se i unutrašnja ubrzanja kao prvi izvodi po vremenu unutrašnjih brzina, odnosno, drugi izvodi po vremenu unutrašnjih koordinata: $\ddot{q}_0(t), \ddot{q}_1(t), \ddot{q}_2(t), \dots, \ddot{q}_{n-1}(t)$. Analogno (2.13), uvode se i n -dimenzionalni vektori unutrašnjih brzina i ubrzanja [1],[2]:

$$\begin{aligned} \dot{q}(t) &= [\dot{q}_0(t), \dot{q}_1(t), \dot{q}_2(t), \dots, \dot{q}_{n-1}(t)]^T \\ \ddot{q}(t) &= [\ddot{q}_0(t), \ddot{q}_1(t), \ddot{q}_2(t), \dots, \ddot{q}_{n-1}(t)]^T \end{aligned} \quad (2.29)$$

Takođe, na osnovu (2.27), uvode se i vektori vanjskih brzina i ubrzanja:

$$\begin{aligned} \dot{X}(t) &= [\dot{x}_A(t), \dot{y}_A(t), \dot{z}_A(t), \dot{\theta}(t), \dot{\phi}(t), \dot{\psi}(t)]^T \\ \ddot{X}(t) &= [\ddot{x}_A(t), \ddot{y}_A(t), \ddot{z}_A(t), \ddot{\theta}(t), \ddot{\phi}(t), \ddot{\psi}(t)]^T \end{aligned} \quad (2.30)$$

Vrijednosti vanjskih koordinata robota kada su poznate unutrašnje koordinate mogu se lako izračunati pomoću izraza [1],[2]:

$$X = \eta(q) \quad (2.31)$$

Međutim, inverzni postupak $q = \eta^{-1}(X)$ nije lako odrediti. Zato se formira veza unutrašnjih i vanjskih brzina (\dot{q} i \dot{X}), kao i unutrašnjih i vanjskih ubrzanja (\ddot{q} i \ddot{X}). Krećući od izraza u (2.31) i računanjem prvog izvoda po vremenu dobija se [1],[2]:

$$\begin{aligned} \frac{dX}{dt} &= \frac{d\eta(q)}{dt} \\ \frac{dX}{dt} &= \frac{d\eta(q)}{dq} \cdot \frac{dq}{dt} \end{aligned} \quad (2.32)$$

Slijedi:

$$\dot{X} = J(q) \cdot \dot{q} \quad (2.33)$$

Daljim računanjem prvog izvoda po vremenu izraza u (2.33), odnosno drugog izvoda po vremenu izraza u (2.31), dobija se [1],[2]:

$$\frac{d^2X}{dt^2} = \frac{d\eta(q)}{dq} \cdot \frac{d^2q}{dt^2} + \frac{d^2\eta(q)}{dq^2} \cdot \left(\frac{dq}{dt} \right)^2 \quad (2.34)$$

Slijedi:

$$\ddot{X} = J(q) \cdot \ddot{q} + B(q, \dot{q}) \quad (2.35)$$

Matrica $J(q)$, dimenzija $n_a \times n$, naziva se Jakobijan, a matrica $B(q, \dot{q})$, dimenzija $n_a \times 1$, pridružena matrica [1],[2]. Veličina n_a predstavlja broj stepeni slobode alata, a n broj zglobova. Za robot Mitsubishi RV-2SDB važi: $n = 6$ i $n_a = 6$ (osim u slučajevima prividnog singulariteta). Izraz u (2.33) daje vezu između vanjskih i unutrašnjih brzina, a izraz u (2.35) vezu između vanjskih i unutrašnjih ubrzanja. Kao što je prikazano u (2.30), vanjske brzine i ubrzanja robota podrazumijevaju linearne i ugaone komponente. Komponente unutrašnjih brzina i ubrzanja mogu biti linearne ili ugaone, u zavisnosti od toga da li je odgovarajući zglob translacioni ili rotacioni. S obzirom na to izraz u (2.33) može se razložiti na dva dijela [2]:

$$\begin{aligned} v &= J_p(q) \cdot \dot{q} \\ \omega &= J_o(q) \cdot \dot{q} \end{aligned} \quad (2.36)$$

Gornja $3 \times n$ submatrica Jakobijana $-J_p(q)$, predstavlja doprinos unutrašnjih brzina linearnoj brzini vrha manipulatora, dok donja $3 \times n$ submatrica $-J_o(q)$, predstavlja doprinos unutrašnjih brzina ugaonoj brzini vrha manipulatora. Na osnovu toga, Jakobijan se može napisati i u razvijenom obliku pomoću kolona vektora J_{p_i} i J_{o_i} dimenzija 3×1 [2]:

$$J = \begin{bmatrix} J_{p_0} & \cdots & J_{p_{n-1}} \\ J_{o_0} & \cdots & J_{o_{n-1}} \end{bmatrix} \quad (2.37)$$

Elementi vektora J_{p_i} predstavljaju doprinos kretanja u i -tom zglobu linearnej brzini kretanja vrha manipulatora duž svake od osa referentnog sistema, dok elementi vektora J_{o_i} predstavljaju doprinos kretanja u istom zglobu ugaonoj brzini vrha manipulatora oko svake ose referentnog sistema. Elemente Jakobijana J_{p_i} i J_{o_i} moguće je izračunati na osnovu parametara geometrije i unutrašnjih koordinata, te su u zavisnosti od tipa zgloba jednaki [2]:

$$J_i = \begin{bmatrix} J_{p_i} \\ J_{o_i} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \times (p - p_{i-1}) \\ z_{i-1} \\ z_{i-1} \end{bmatrix}, & \text{za rotacioni zglob} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, & \text{za translacioni zglob} \end{cases} \quad (2.38)$$

Vektori z_{i-1} , p i p_{i-1} jesu funkcije unutrašnjih koordinata i mogu se izračunati na sljedeći način [2]:

- z_{i-1} dat je trećom kolonom matrice rotacije $R_{0,i-1}$:

$$z_{i-1} = R_{0,1} \cdot R_{1,2} \cdots R_{i-2,i-1} \cdot [0 \ 0 \ 1]^T \quad (2.39)$$

- p je dat s prva tri elementa četvrte kolone prelazne matrice $T_{0,n}$:

$$p = T_{0,1}(q_0) \cdot T_{1,2}(q_1) \cdots T_{n-1,n}(q_{n-1}) \cdot [0 \ 0 \ 1]^T \quad (1:3) \quad (2.40)$$

- p_{i-1} dat je s prva tri elementa četvrte kolone prelazne matrice $T_{0,i-1}$:

$$p_{i-1} = T_{0,1}(q_0) \cdot T_{1,2}(q_1) \cdots T_{i-2,i-1}(q_{i-2}) \cdot [0 \ 0 \ 1]^T \quad (1:3) \quad (2.41)$$

Za razliku od Jakobijana, pridružena matrica još zavisi i od brzina unutrašnjih koordinata, a njena i -ta vrsta može se odrediti na sljedeći način [2]:

$$B_i = \left[\sum_{j=1}^n e_{ij} \dot{q}_j \right] \quad (2.42)$$

gdje vektor e_{ij} predstavlja parcijalni izvod i -te kolone Jakobijana po j -toj unutrašnjoj promjenljivoj i dat je izrazima [2]:

$$\begin{aligned} e_{ij} &= z_{i-1} \cdot J_i, & \text{za } i \leq j \\ e_{ij} &= e_{ji}, & \text{za } i > j \end{aligned} \quad (2.43)$$

Kretanje robota sada je moguće direktno zadati pomoću vremenskih promjena unutrašnjih koordinata $q(t)$. Zakonima $q(t)$ određeni su i izvodi $\dot{q}(t)$ i $\ddot{q}(t)$. Ovakvo sračunavanje spoljašnjeg kretanja $X(t)$, kada je poznato unutrašnje kretanje $q(t)$, naziva se direktnim problemom kinematike. Takođe, aproksimacijom izraza u (2.33) moguće je izračunati i približnu vrijednost vanjskih koordinata za male promjene unutrašnjih koordinata:

$$\begin{aligned} \Delta X &\cong J(q) \cdot \Delta q \\ X - X_0 &\cong J(q) \cdot (q - q_0) \\ X &\cong J(q) \cdot (q - q_0) + X_0 \end{aligned} \quad (2.44)$$

2.4.2. Inverzni problem kinematike

Prethodno opisani način zadavanja kretanja pogodan je samo za jednostavne manipulacione zadatke, koje izvršavaju manje složeni roboti. U složenijim zadacima uvijek je neophodno ostvariti određeno kretanje alata za koje je nemoguće unaprijed predvidjeti pomjeranja u zglobovima manipulatora. Takođe, često je potrebno precizno održavati brzinu alata duž cijele putanje ili se brzina indirektno definiše vremenskim ograničavanjem kretanja alata. Nekad je potrebno voditi računa i o ubrzanjima pri kretanju da bi se smanjili trzaji ili nagli pokreti robota. Sve ovo dodatno komplikuje direktno zadavanje kretanja manipulatora preko unutrašnjih koordinata. Takav se tip zadatka zato određuje preko vektora spoljašnjeg položaja hvataljke, to jest, vektora vanjskih koordinata $X(t)$, prikazanog u (2.27), gdje $x_A(t)$, $y_A(t)$ i $z_A(t)$ predstavljaju položaj vrha hvataljke, a $\theta(t)$, $\phi(t)$ i $\psi(t)$ orijentaciju hvataljke.

Vrijednosti unutrašnjih koordinata robota, kada su poznate vanjske koordinate, moguće je izračunati inverznim postupkom u odnosu na (2.31) [1],[2]:

$$q = \eta^{-1}(X) \quad (2.45)$$

Slično kao i kod direktnog problema kinematike, računanjem prvog izvoda po vremenu izraza u (2.45) dobija se [1],[2]:

$$\begin{aligned} \frac{dq}{dt} &= \frac{d\eta^{-1}(X)}{dt} \\ \frac{dq}{dt} &= \frac{d\eta^{-1}(X)}{dX} \cdot \frac{dX}{dt} \end{aligned} \quad (2.46)$$

Izraz $\frac{d\eta^{-1}(X)}{dX}$ predstavlja inverznu Jakobijanovu matricu, pa slijedi:

$$\dot{q} = J^{-1}(q) \cdot \dot{X} \quad (2.47)$$

Postupak u kojem je potrebno odrediti unutrašnje kretanje $q(t)$ za poznato spoljašnje kretanje $X(t)$ naziva se inverzni problem kinematike. Inverzna kinematika nešto je složeniji problem od direktne kinematike. Izraz u (2.47) nije moguće odrediti u sljedećim situacijama:

- Ukoliko je broj vanjskih koordinata veći od broja unutrašnjih koordinata ($n_a > n$);
- Ukoliko je broj vanjskih koordinata jednak broju unutrašnjih koordinata ($n_a = n$), ali se manipulator nalazi u singularnom položaju ($\det J(q) \neq 0$).

S druge strane, izraz u (2.47) može imati više od jednog rješenja, i to:

- Ukoliko je broj vanjskih koordinata manji od broja unutrašnjih koordinata ($n_a < n$), to jest, ukoliko je manipulator redundantan;
- Ukoliko je broj vanjskih koordinata jednak broju unutrašnjih koordinata ($n_a = n$), ali je preslikavanje η^{-1} nejednoznačno.

Ipak, ukoliko postoji rješenje datog problema, odnosno ukoliko je broj vanjskih koordinata jednak broju unutrašnjih koordinata ($n_a = n$) i ukoliko se manipulator ne nalazi

u singularnom položaju ($\det J(q) = 0$), tada je unutrašnje kretanje moguće odrediti na jedan od sljedeća dva načina [2]:

1. Kretanjem od tačke do tačke;
2. Kontinualnim praćenjem trajektorije.

U sklopu diplomskog rada kretanje manipulatora realizovano je na prvi način, odnosno kao kretanje od tačke do tačke, pri čemu se inverzni problem kinematike rješava Njutnovim iterativnim metodom. Ukoliko se za proizvoljan položaj X_k na trajektoriji prepostavi inverzno rješenje q_k , tada je greška u vanjskim koordinatama između zadane vrijednosti X_k i vrijednosti, koja odgovara prepostavljenom rješenju, jednaka:

$$\Delta X_k = X_k - \eta(q_k) \quad (2.48)$$

Na osnovu ove greške potrebno je procijeniti novo rješenje q_{k+1} , koje bi bilo bolje od prethodnog. Aproximacijom izraza iz (2.47) slijedi:

$$\Delta q = J^{-1}(q) \cdot \Delta X \quad (2.49)$$

Dalje, za proizvoljan položaj X_k i prepostavljeno rješenje q_k vrijedi:

$$q_{k+1} \cong J^{-1}(q) \cdot \Delta X_k + q_k \quad (2.50)$$

Prethodni se koraci ponavljaju sve dok je $|\Delta X_k| > \epsilon$, gdje je ϵ konstanta koja definiše zahtijevanu tačnost vanjskih koordinata. Na ovaj se način definiju svi položaji X_k , koje manipulator treba da postigne za rješavanje traženog zadatka.

2.5. Tehničko pismo

Za pisanje teksta pomoću robota korišteno je tehničko pismo. Ono je definisano standardom ISO 3098/1 i koristi se za ispisivanje natpisa, oznaka i brojeva u tehničkim crtežima [9]. Standardom se propisuju tri parametra:

- nazivna visina pisma – H ;
- širina crte (u sklopu ovog zadatka definisana širinom markera za pisanje);
- širina znakova.

Nazivna visina pisma H jeste visina velikih slova i brojeva. Veličinom H određene su i sve druge dimenzije slova i brojeva. Nazivna visina pisma odabira se na osnovu raspoloživog prostora i značenja teksta. Razmaci između pojedinačnih znakova, riječi i linija nisu strogo definisani standardom, te su u sklopu diplomskega rada izabrani proizvoljno. Tehničko pismo može biti uspravno, pod uglom od 90° , ili koso, pod uglom od 75° u odnosu na horizontalu.

Postoje tri tipa tehničkog pisma:

- tip A – usko;
- tip B – normalno;
- tip C – široko.

U zavisnosti od tipa tehničkog pisma razlikuju se i prethodno navedeni parametri pisma. U tabeli 2. prikazani su različiti parametri tehničkog pisma, grupisani po tipu.

Parametri \ Tip	A	B	C
Visina velikih slova i brojeva	H	H	H
Širina velikih slova	$\frac{6}{14}H$	$\frac{7}{10}H$	$\frac{5}{7}H$
Visina malih slova	$\frac{10}{14}H$	$\frac{7}{10}H$	$\frac{5}{7}H$
Širina malih slova i brojeva	$\frac{6}{14}H$	$\frac{6}{10}H$	$\frac{4}{7}H$
Širina crte	$\frac{1}{14}H$	$\frac{1}{10}H$	$\frac{1}{7}H$
Rastojanje između slova i/ili brojeva	$\frac{2}{14}H$	$\frac{2}{10}H$	$\frac{1}{7}H$
Rastojanje između riječi	$\frac{6}{14}H$	$\frac{6}{10}H$	$\frac{4}{7}H$
Rastojanje između pojedinačnih linija	$\frac{22}{14}H$	$\frac{16}{10}H$	$\frac{11}{7}H$

Tabela 2 – Parametri tehničkog pisma

U sklopu diplomskog rada realizovano je pisanje ciriličnim tehničkim pismom. Uspravno i koso cirilično tehničko pismo tipa B prikazano je na slici 2.14.



a) Uspravno

b) Koso

Slika 2.14 – Cirilično tehničko pismo tipa B

3. Realizacija sistema za automatsko pisanje teksta pomoću Mitsubishi RV-2SDB robota

Sistem za automatsko pisanje teksta pomoću robota podrazumijeva sistem u kojem korisnik putem digitalnog računara upravlja parametrima radnog okruženja, unosi tekst i njegove osobine, uspostavlja komunikaciju sa robotskim kontrolerom i šalje komande za upravljanje robotom, na osnovu kojih se vrši ispisivanje zadatog teksta. Sistem se sastoji iz sljedećih komponenata:

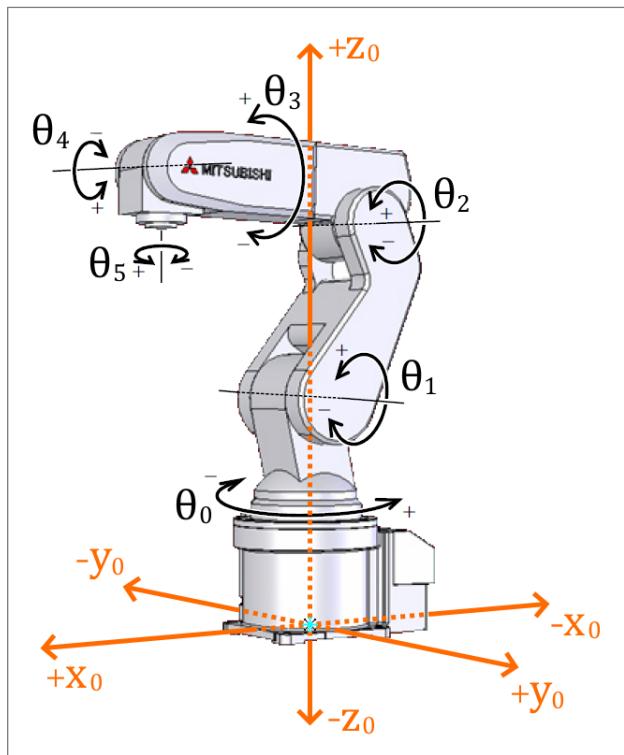
- Industrijski robot Mitsubishi RV-2SDB;
- Robotski kontroler Mitsubishi CR1DA-700 i ručni upravljački uređaj R32TB;
- Marker sa držačem kao završni uređaj;
- Površina za pisanje;
- Digitalni računar i programi napisani u MATLAB programskom okruženju;

Za normalno funkcioniranje sistema neophodno je da se ispoštuju određeni preduslovi. Baza robota treba da bude pričvršćena za čvrstu podlogu, da ne bi dolazilo do trzaja prilikom kretanja robota. Radni prostor robota treba da bude bez fizičkih prepreka, jer robot ne posjeduje nikakvu vrstu vanjskih senzora na osnovu kojih bi mogao primati informacije iz okruženja i orijentisati se u prostoru. Površinu za pisanje neophodno je fiksirati i postaviti unutar radnog prostora robota, tako da je ravan površine za pisanje paralelna sa x_0 - y_0 ravni referentnog koordinatnog sistema robota. Pri svakoj izmjeni markera, a pogotovo ako se dimenzije novog markera razlikuju od prethodnog, potrebno je izvršiti kalibraciju položaja vrha markera, da bi pisanje pomoću robota bilo što tačnije i preciznije. Računar na kojem se pokreću programi mora imati mogućnost povezivanja sa robotskim kontrolerom putem Ethernet konekcije. Što se tiče softverskih zahtjeva, ukoliko na računaru nije instalirana MATLAB R2020a verzija, potrebno je instalirati MATLAB Runtime R2020a, pomoću kojeg se mogu pokrenuti MATLAB izvršne datoteke.

3.1. Opis glavnih komponenata sistema

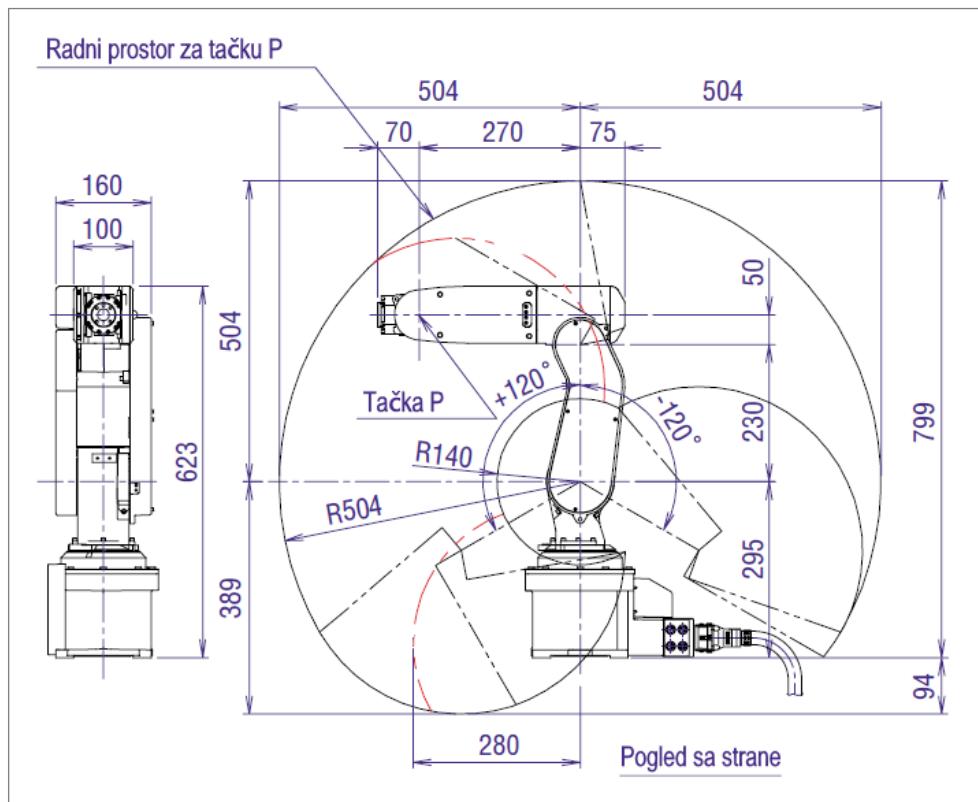
3.1.1. Mitsubishi RV-2SDB industrijski robot

Mitsubishi RV-2SDB [10] jeste industrijski robot vertikalnog tipa namijenjen različitim vrstama zadataka u cilju automatizacije i povećanja efikasnosti izvršavanja. Ovaj se tip robota često naziva i robotska ruka zbog svoje strukture i načina rada, koji je sličan ljudskoj ruci. Posjeduje šest rotacionih zglobova i šest stepeni slobode, odnosno, vrh manipulatora može da ostvari šest nezavisnih kretanja. U mogućnosti je da postigne isti položaj sa ponovljivošću od 0.02 mm . Pogonski sistem robota sastoji se od servo motora, te se za detekciju pozicije koriste apsolutni enkoderi. Maksimalna brzina vrha manipulatora iznosi 4400 mm/s . Maksimalna nosivost jeste 3 kg . Zbog široke primjene ovog tipa robota postoji veliki izbor završnih uređaja, koji se mogu montirati na robotski sistem. U cilju rješavanja zadatka u sklopu diplomskog rada kao završni uređaj koristi se marker pričvršćen za manipulator pomoću metalnog nosača. Robot, njegov referentni koordinatni sistem i ose svih zglobova prikazani su na slici 3.1.



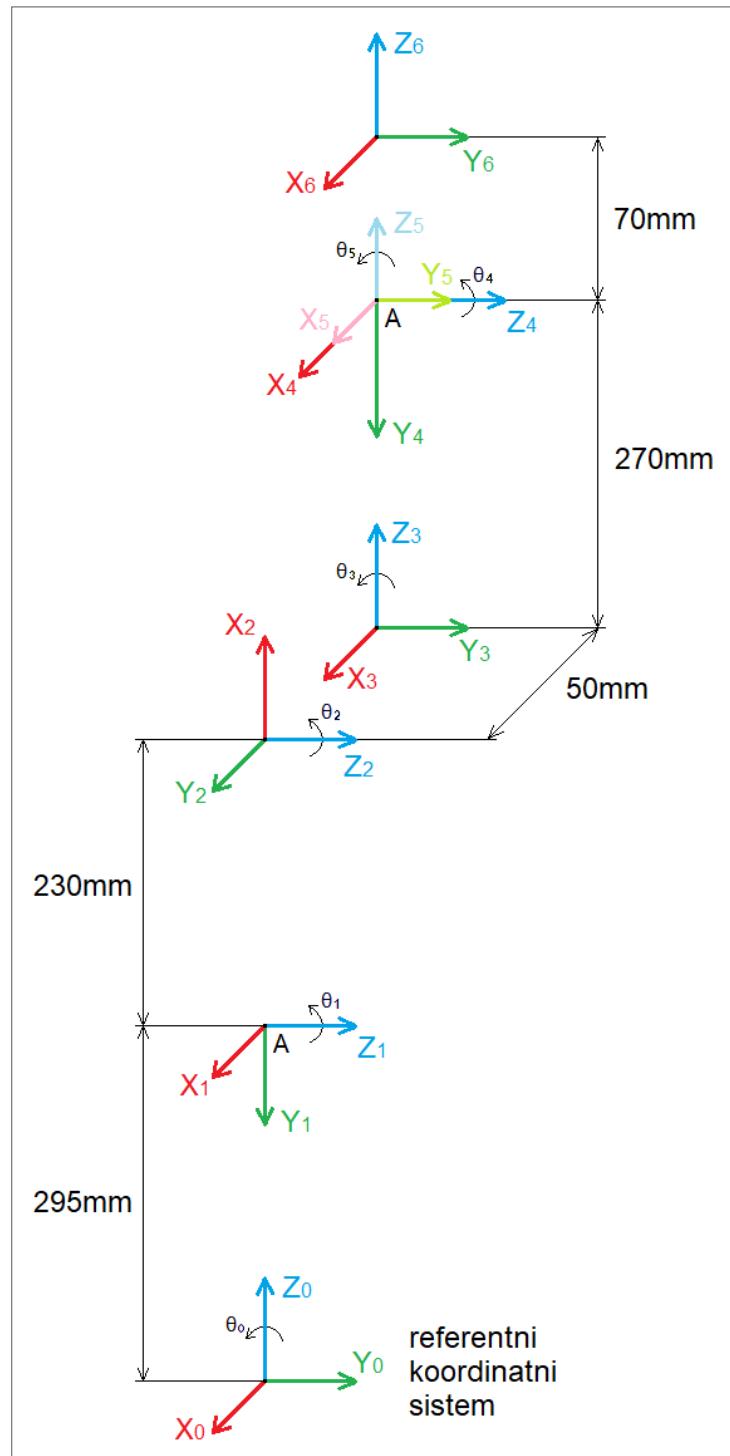
Slika 3.1 – Mitsubishi RV-2SDB – referentni sistem i ose svih zglobova

Da bi se formirala DH tabela kinematičkih parametara, neophodno je poznavati dimenzije svih segmenata robota. Na slici 3.2 prikazane su vanjske dimenzije robota iz zvanične specifikacije [10].



Slika 3.2 – Vanjske dimenzije robota Mitsubishi RV-2SDB

U početnom položaju, kada su sve unutrašnje koordinate jednake 0 , robot je u potpunosti ispravljen u pozitivnom smjeru z_0 ose referentnog koordinatnog sistema. Vezani koordinatni sistemi postavljaju se na osnovu Denavit–Hartenbergove konvencije, kao što je prikazano na slici 3.3.



Slika 3.3 – Vezani koordinatni sistemi za robot Mitsubishi RV-2SDB

Nakon što se postave vezani koordinatni sistemi, formira se DH tabela kinematickih parametara, koja predstavlja osnovu i za kasnije kreiranje modela robota unutar MATLAB okruženja.

	θ ↓	d ↓	a ↓	α ↓	s ↓
$0 - 1 \rightarrow$	θ_0	295	0	$-\pi/2$	0
$1 - 2 \rightarrow$	$\theta_1 - \pi/2$	0	230	0	0
$2 - 3 \rightarrow$	$\theta_2 + \pi/2$	0	-50	$\pi/2$	0
$3 - 4 \rightarrow$	θ_3	270	0	$-\pi/2$	0
$4 - 5 \rightarrow$	θ_4	0	0	$\pi/2$	0
$5 - 6 \rightarrow$	θ_5	70	0	0	0

Tabela 3 – DH tabela kinematickih parametara za robot Mitsubishi RV-2SDB

Svaki redak u DH tabeli predstavlja parametre i -tog segmenta robota (θ_i , d_i , a_i i α_i), na osnovu kojih se vrši prelaz iz vezanog koordinatnog sistema $i - 1$ u vezani koordinatni sistem i . Parametar s_i predstavlja tip zgloba S_{i-1} , kako je i opisano u (2.11).

3.1.2. Mitsubishi CR1DA-700 robotski kontroler

Centralni dio industrijskog robotskog sistema jeste robotski kontroler. Predstavlja mozak sistema i njegov je zadatak da upravlja robotom i ostalim dijelovima sistema. Kontroler je moguće povezati i sa drugim sistemima unutar određenog industrijskog procesa, tako da se omogući njihova međusobna komunikacija i interoperabilnost. Model kontrolera koji je korišten u sklopu praktičnog dijela rada jeste Mitsubishi CR1DA-700 [10]. Broj osa koji ovaj kontroler može istovremeno da kontroliše jeste šest. Posjeduje memoriju, u koju je moguće pohraniti i do 256 različitih programa. Programska jezik koji je podržan od strane kontrolera jeste MELFA-BASIC V. Komunikaciju sa kontrolerom moguće je ostvariti putem različitih komunikacionih protokola, od kojih su najviše u upotrebi: RS232, Ethernet i USB, a uz dodatne module moguće je koristiti i robusnije industrijske protokole, kao što su Profibus i DeviceNet. U sklopu praktičnog dijela rada korištena je Ethernet konekcija. Na slici 3.4 prikazan je kontroler CR1DA-700, sa prednje i sa zadnje strane.



a) Prednja strana

b) Zadnja strana

Slika 3.4 – Mitsubishi CR1DA-700 robotski kontroler

Na robotski kontroler moguće je priključiti i ručni upravljački uređaj (engl. *Teach Pendant*), koji omogućava upravljanje robotom bez potrebe za klasičnim programiranjem

putem računara. Takvo upravljanje zasnovano je na principu „obuci i ponavljam“ (engl. *teach and repeat*), što je veoma pogodno s obzirom na to da se industrijski roboti, u većini slučajeva, koriste za repetitivne zadatke. Pomoću ovog uređaja moguće je vršiti i kontrolu parametara robota kao što su: brzina kretanja, ubrzanje i usporenje pri kretanju, položaj koordinatnog sistema alata i slično. Takođe, u svakom je trenutku moguće očitati i položaj robota u vanjskim ili u unutrašnjim koordinatama. Uz Mitsubishi RV-2SDB robot dolazi R32TB ručni upravljač [10], koji je prikazan na slici 3.5. U sklopu praktičnog dijela diplomskog rada uglavnom je korišten za kalibraciju položaja vrha markera i površine za pisanje.



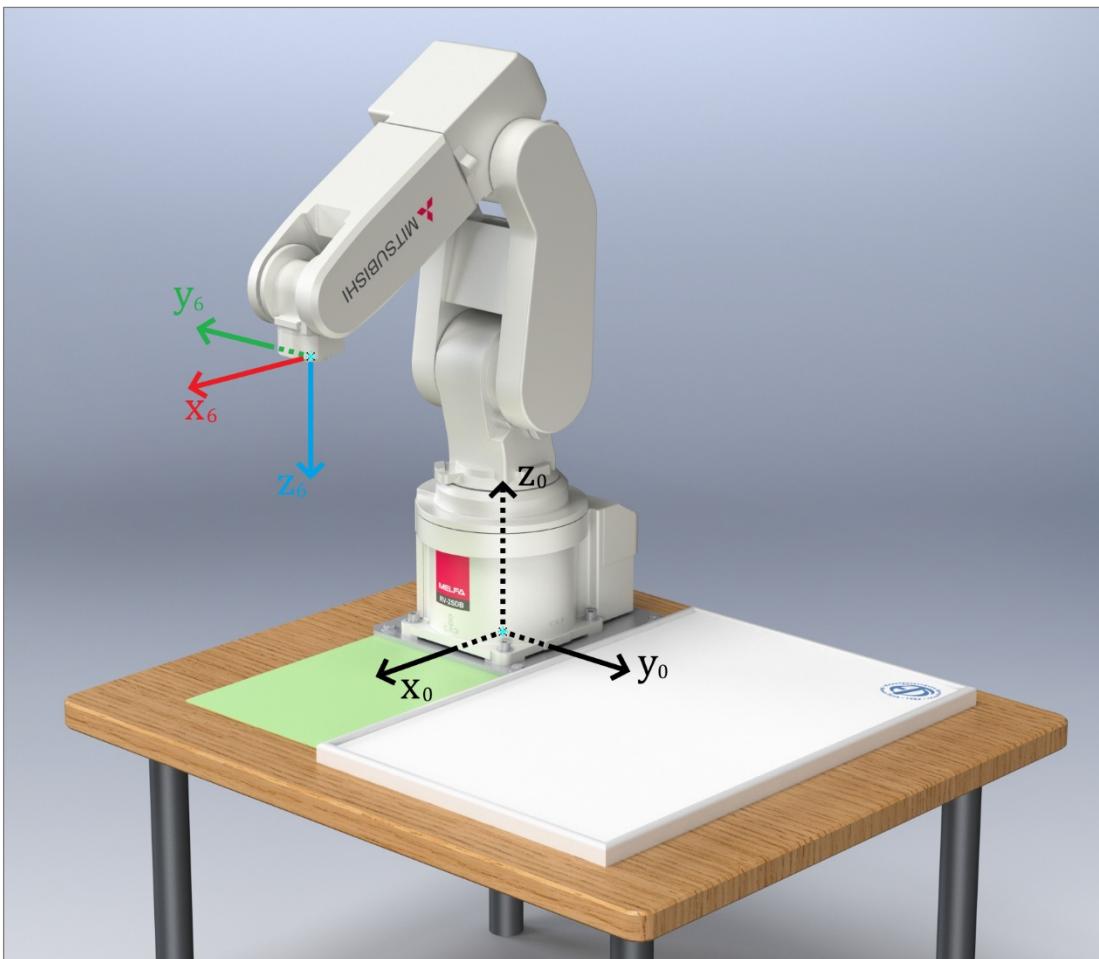
Slika 3.5 – Mitsubishi R32TB ručni upravljač

3.2. Kalibracija sistema

Prije nego što bude moguće ispisivati tekst pomoću robota, potrebno je izvršiti kalibraciju sistema, što podrazumijeva kalibraciju položaja vrha markera i površine za pisanje. Kalibracija se odrađuje pomoću ručnog upravljača – navođenjem robota na željeni položaj u prostoru, očitavanjem vanjskih koordinata robota, vršenjem odgovarajućih proračuna i modifikovanjem programskih varijabli u MATLAB-u. Radi što boljeg prikaza procesa kalibracije, radni prostor u toku kalibracije modelovan je i vizuelizovan korištenjem SOLIDWORKS programske rješenja. Vizuelizacija uključuje robotski manipulator, koji je postavljen na metalnu ploču debljine 8 mm, površinu za pisanje i površinu za kalibraciju, kako je prikazano na slici 3.6.

3.2.1. Kalibracija položaja vrha alata

Ovaj se tip kalibracije naziva još i kalibracija centralne tačke alata (engl. *tool center point calibration*), pri čemu se u ovom slučaju alatom smatra marker. U industriji se kalibracija centralne tačke alata, uz pomoć različitih uređaja i softverskih paketa, najčešće vrši automatski. U sklopu praktičnog dijela diplomskog rada ona je rađena ručno, i to u nekoliko koraka.



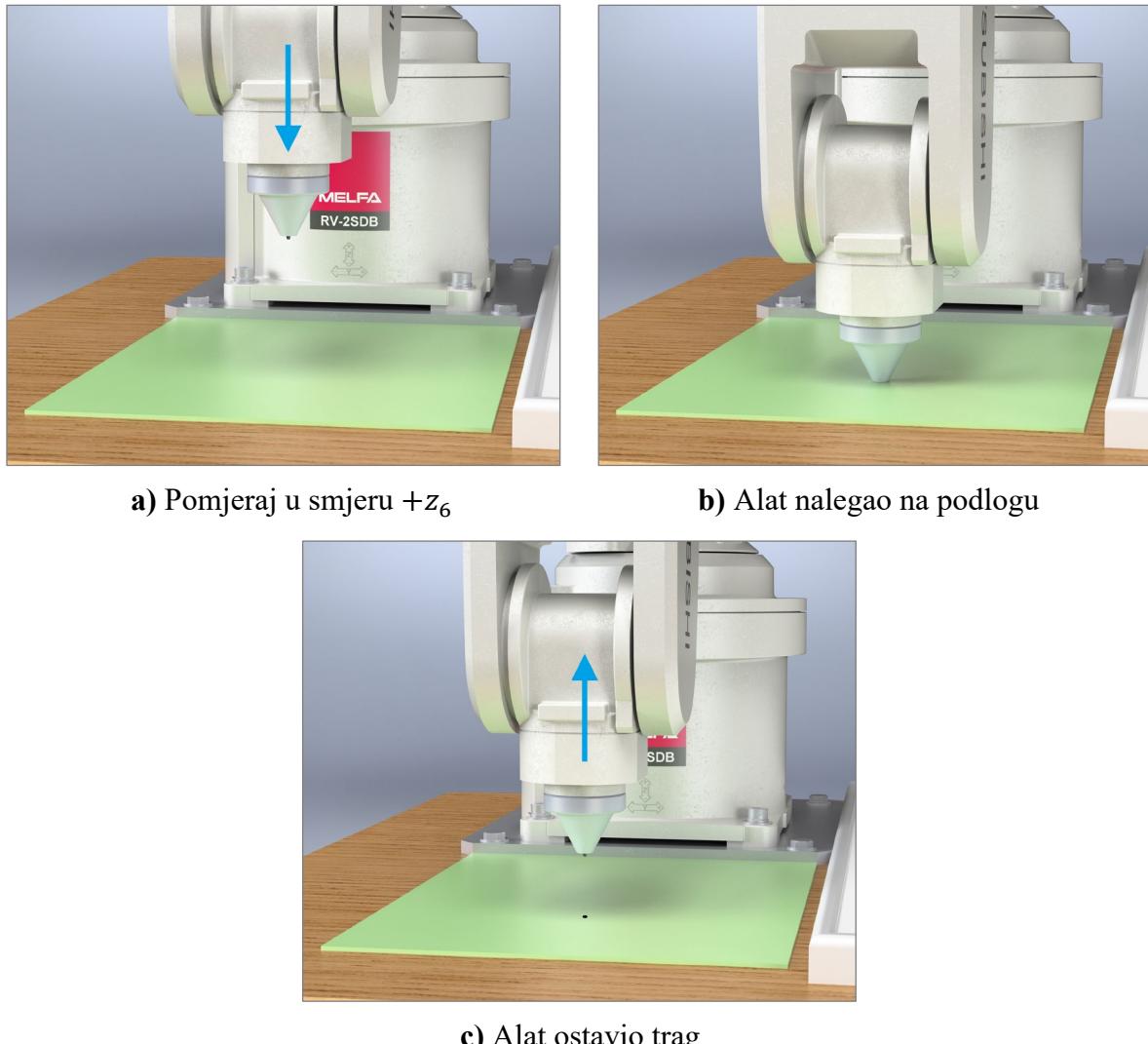
Slika 3.6 – Vizuelizacija radnog prostora u toku kalibracije

Prvi korak podrazumijeva montiranje posebnog kalibracionog alata, poznatih dimenzija, na vrh manipulatora. Alat je prikazan na slici 3.7. Njegova visina iznosi 30 mm , ne uključujući vrh koji se može uvlačiti. Bitno je istaći da vrh alata leži na osi simetrije alata, koja se poklapa sa z_6 osom posljednjeg vezanog koordinatnog sistema kada je alat montiran na vrh manipulatora. To ga čini idealnim za označavanje referentne (centralne) tačke na površini za kalibraciju, na osnovu koje se kasnije mjeri pomjeraj tačke koju označi marker. Površina za kalibraciju prikazana je zelenom bojom na slici 3.6, i njena debljina iznosi 2 mm .



Slika 3.7 – Kalibracioni alat

Nakon što se alat montira, manipulator se pozicionira tako da je z_6 osa posljednjeg vezanog koordinatnog sistema (vrh manipulatora) u suprotnom smjeru u odnosu na z_0 osu referentnog koordinatnog sistema, kako je i prikazano na slici 3.6. Zatim se vrh manipulatora pomoću ručnog upravljača pomjera u pozitivnom smjeru z_6 ose, sve dok kalibracioni alat ne nalegne na površinu za kalibraciju i ne ostavi trag, kako je prikazano na slici 3.8.



Slika 3.8 – Prvi korak kalibracije položaja vrha markera

Kada se kalibracioni alat nalazi u položaju kao na slici 3.8 b), na displeju ručnog upravljača prikazane su vanjske koordinate robota kao na slici 3.9.



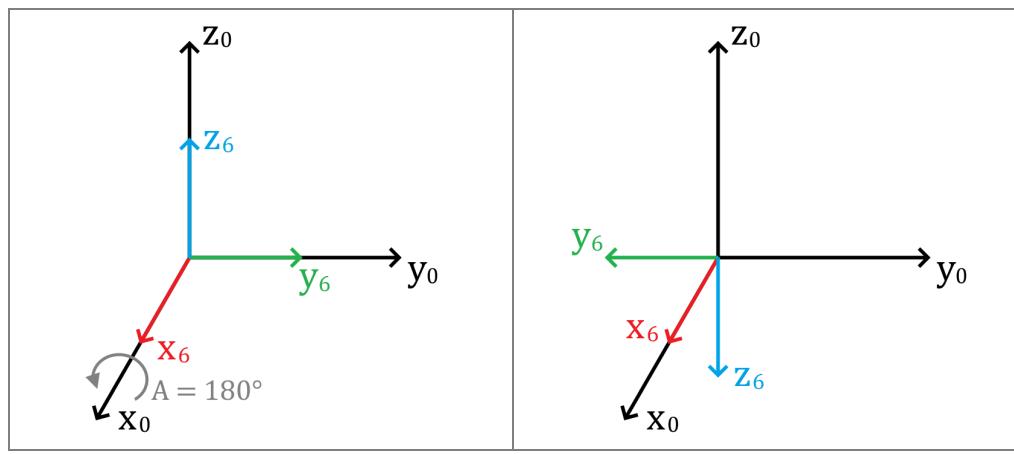
Slika 3.9 – Displej ručnog upravljača za položaj alata na slici 3.8 b)

Ono što je važno razmotriti jeste Z koordinata vrha manipulatora, te uglovi A , B i C , koji odgovaraju xyz varijanti aeronautičkih uglova. Na osnovu visine ploče na koju je postavljen robot, referentni koordinatni sistem leži 8 mm iznad podloge (stola). Stoga je površina za kalibraciju za 6 mm niža u odnosu na ploču. Uzimajući u obzir i dimenzije kalibracionog alata, jasno je da je Z koordinata vrha manipulatora tada:

$$Z = 30 - 6 = 24\text{ mm} \quad (3.1)$$

kao što je i prikazano na displeju, uz zanemarivu grešku, zbog toga što alat nije u potpunosti nalegao na površinu za kalibraciju.

Uglovi orientacije A , B i C izaženi su u stepenima i pokazuju rotacije koordinatnog sistema vezanog za vrh manipulatora u odnosu na referentni sistem. Na osnovu slike 3.6, da se zaključiti da se sistem $Ox_6y_6z_6$ dobija rotacijom oko x_0 ose referentnog sistema za 180° .



a) Početni položaj

b) Nakon rotacije

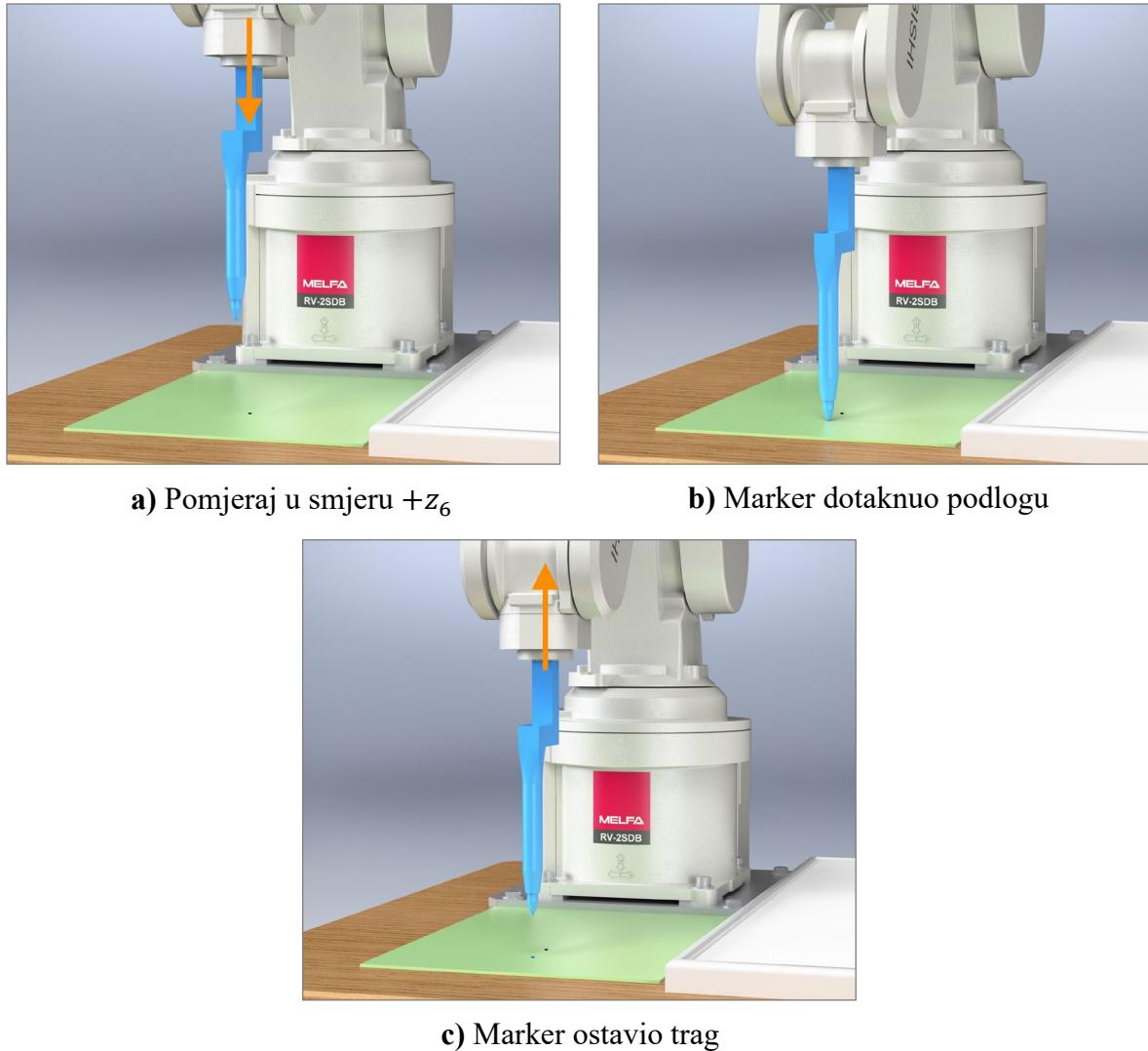
Slika 3.10 – Orientacija sistema $Ox_6y_6z_6$ u toku kalibracije položaja vrha markera

Drugi korak podrazumijeva montiranje markera sa nosačem na vrh manipulatora. Ukoliko se ne može obezbijediti idealno ravna površina za pisanje, potrebno je izabrati marker s vrhom koji se može uvlačiti, tako da eventualne neravnine na površini za pisanje ili mali nagib površine za pisanje ne utiču na mogućnost pisanja nakon izvršene kalibracije. Marker korišten u toku praktičnog dijela zadatka prikazan je na slici 3.11.



Slika 3.11 – Korišteni marker

Kao i u prvom koraku vrh manipulatora se pomjera u pozitivnom smjeru z_6 ose, sve dok marker ne dotakne podlogu i ne ostavi trag, kao što je prikazano na slici 3.12. Marker sa nosačem, radi vizuelizacije, modelovan je uprošteno, pri čemu položaj vrha modelovanog markera odgovara stvarnom položaju.



Slika 3.12 – Drugi korak kalibracije položaja vrha markera

Kada se marker nalazi u položaju kao na slici 3.12 b), na displeju ručnog upravljača prikazane su vanjske koordinate robota kao na slici 3.13.

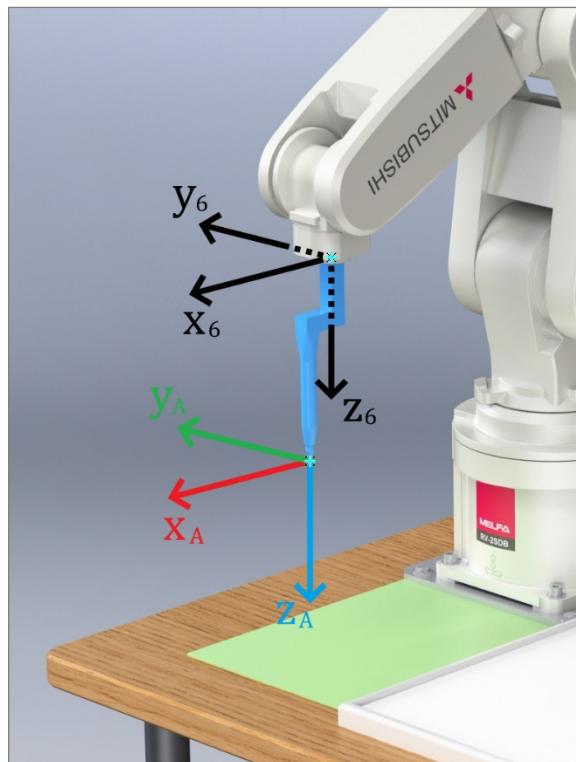


Slika 3.13 – Displej ručnog upravljača za položaj markera na slici 3.12 b)

Može se primijetiti da je Z koordinata vrha manipulatora, kada se zanemari decimalni dio, jednaka 170 mm . Uzimajući u obzir da je vrh markera za 6 mm niži od ploče na koju je postavljen robot, ukupna visina markera sa nosačem iznosi:

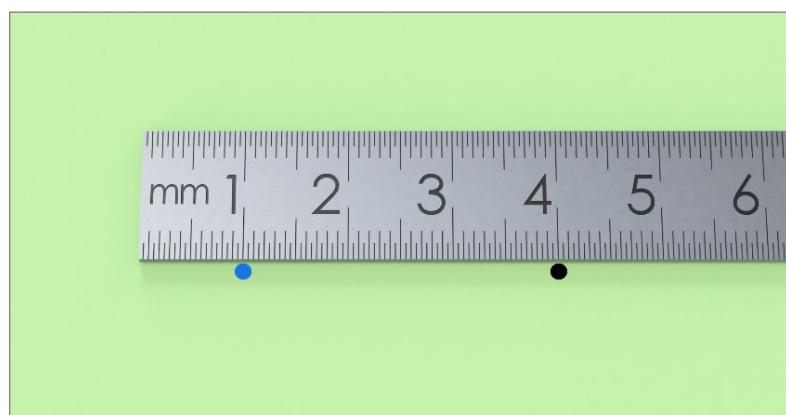
$$p_{A_z} = 170 + 6 = 176\text{ mm} \quad (3.2)$$

Još je potrebno da se izmjeri udaljenost označenih tačaka na površini za kalibraciju da bi se mogla formirati prelazna matrica koja opisuje transformaciju koordinatnog sistema $Ox_6y_6z_6$ u koordinatni sistem $Ox_Ay_Az_A$, vezan za vrh markera. Sistem $Ox_Ay_Az_A$ postavlja se tako da nema dodatnih rotacija u odnosu na sistem $Ox_6y_6z_6$ kao što je prikazano na slici 3.14.



Slika 3.14 – Koordinatni sistem $Ox_Ay_Az_A$

Udaljenost u pravcu y_6 ose bila je zanemariva, pa je mjerena samo udaljenost u pravcu x_6 ose. Za mjerjenje udaljenosti korišten je lenjir kao što je prikazano na slici 3.15.



Slika 3.15 – Mjerjenje udaljenosti označenih tačaka na površini za kalibraciju

Dakle, pomjeraj vrha markera u odnosu na referentnu (centralnu) tačku u pozitivnom smjeru x_6 ose iznosi:

$$p_{Ax} = 30 \text{ mm} \quad (3.3)$$

Homogena prelazna matrica markera sada je:

$$T_A = \begin{bmatrix} 1 & 0 & 0 & 30 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 176 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Ono što je još preostalo jeste da se stari položaj vrha alata (vrh manipulatora) ažurira na novi (vrh markera) promjenom MEXTL parametra robota, kako je prikazano na slici 3.16. Ovaj parametar podrazumijeva šest veličina, od kojih su tri za poziciju i tri za orijentaciju aeronautičkog tipa xyz . Homogenu prelaznu matricu markera u ovom obliku zapisujemo kao:

$$T_A = [30, 0, 176, 0, 0, 0] \quad (3.5)$$



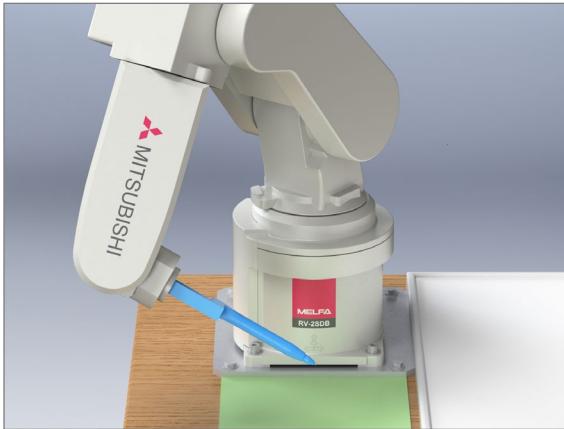
a) Prvobitna vrijednost parametra – vrh manipulatora



b) Nova vrijednost parametra – vrh markera

Slika 3.16 – Promjena MEXTL parametra pomoći ručnog upravljača

Ovim je završena kalibracija položaja vrha markera. Pored toga što je sada moguće tačno pozicionirati vrh markera u željenu tačku u prostoru, moguće je i mijenjati orijentaciju markera bez toga da se mijenja pozicija vrha markera, kako je prikazano na slici 3.17. Ovo je veoma pogodno iz više razloga. Ukoliko se, na primjer, vrh markera nalazi blizu podloge ili okolnih objekata, a prethodno nije odrđena kalibracija položaja vrha markera, promjenom orijentacije vrha manipulatora može doći do njihovog međusobnog sudara, jer robot nema informaciju o dimenzijama alata ili o tome gdje se nalazi vrh alata. Takođe, ukoliko se želi promijeniti orijentacija markera u toku pisanja, to je moguće uraditi bez remećenja procesa pisanja i bez mijenjanja trenutne pozicije vrha markera.



```

<CURRENT> XYZ      100% M1 T00 B*
X : 276.01   A : 69.03
Y : 0.00     B : 165.20
Z : 90.00    C : 0.01
FL1:00000006  FL2:00000000
JOINT TOOL JOG 3-XYZ CYLNDR>

```

a) Orijentacija 1



```

<CURRENT> XYZ      100% M1 T00 B*
X : 276.00   A : 15.01
Y : 0.00     B : 150.09
Z : 90.01    C : 15.32
FL1:00000006  FL2:00000000
JOINT TOOL JOG 3-XYZ CYLNDR>

```

b) Orijentacija 2



```

<CURRENT> XYZ      100% M1 T00 B*
X : 276.00   A : 135.61
Y : 0.00     B : -30.04
Z : 89.99    C : -15.17
FL1:00000006  FL2:00000000
JOINT TOOL JOG 3-XYZ CYLNDR>

```

c) Orijentacija 3



```

<CURRENT> XYZ      100% M1 T00 B*
X : 276.00   A : 149.87
Y : 0.00     B : 45.06
Z : 90.00    C : -0.03
FL1:00000006  FL2:00000000
JOINT TOOL JOG 3-XYZ CYLNDR>

```

d) Orijentacija 4

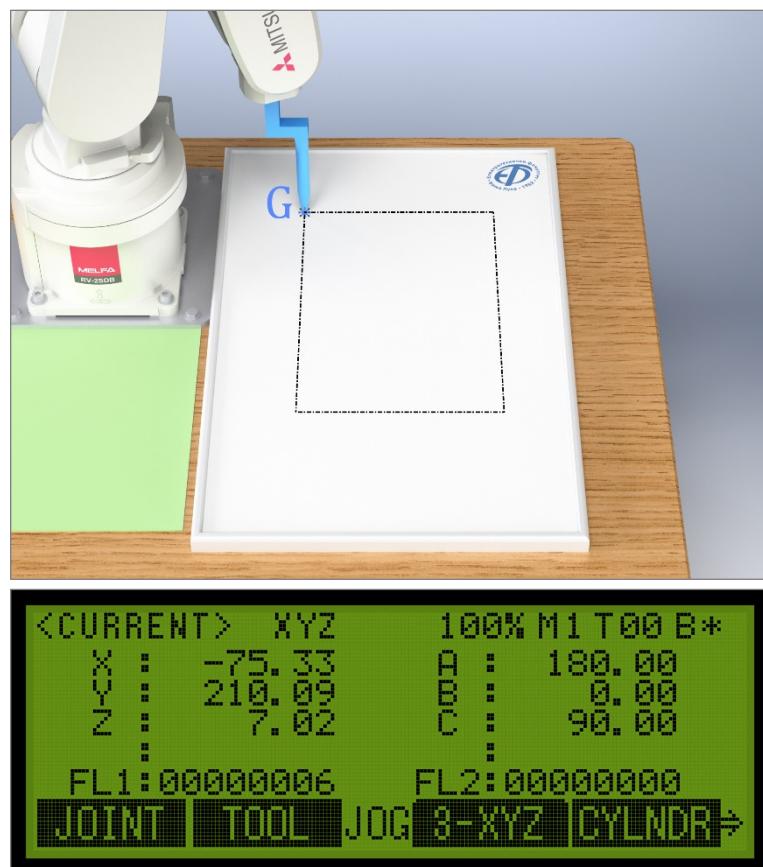
Slika 3.17 – Primjeri različitih orijentacija markera sa istom pozicijom

3.2.2. Kalibracija položaja površine za pisanje

Kalibracija površine za pisanje podrazumijeva pronađenje koordinata njene gornje lijeve granične tačke u referentnom koordinatnom sistemu robota, koja ima oznaku G na slici 3.18, a zatim definisanje njenih dimenzija, odnosno visine u pravcu x_0 ose i širine u pravcu y_0 ose referentnog sistema. S obzirom na to da je ravan površine za pisanje paralelna sa x_0 - y_0 ravni referentnog sistema, nije bilo potrebe za definisanjem njene orijentacije. Vrh se markera pomoću ručnog upravljačkog uređaja dovodi u tačku G , te se očitavaju koordinate sa displeja upravljača, kako je prikazano na slici 3.19.



Slika 3.18 – Površina za pisanje i njene odrednice



Slika 3.19 – Koordinate tačke G

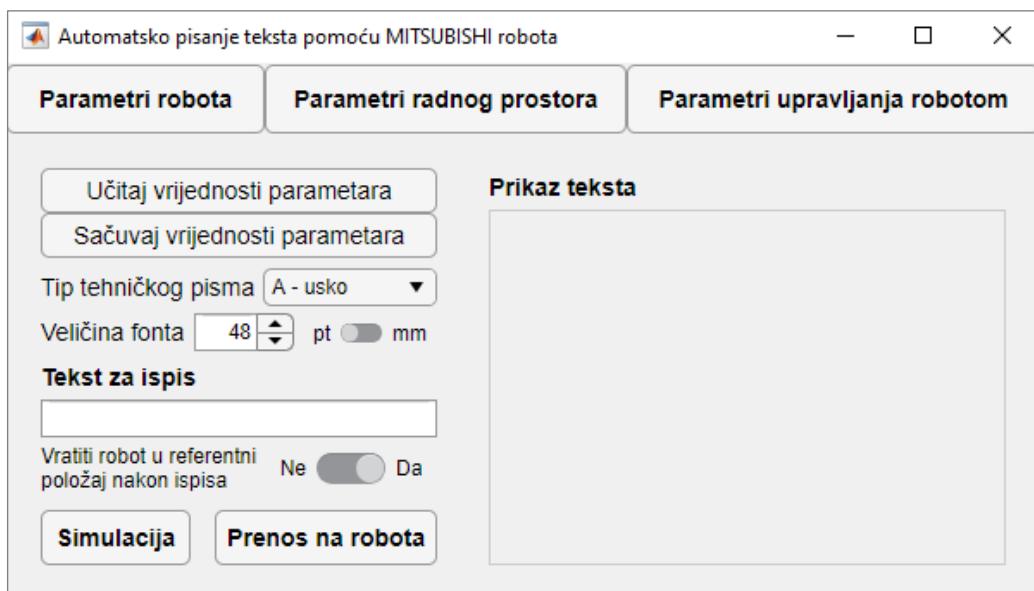
Bitno je napomenuti da vrh markera pri proizvoljno definisanoj orijentaciji mora biti u mogućnosti da dostigne sve četiri granične tačke površine za pisanje, da bi se svaki znak unutar granica površine za pisanje mogao uspješno ispisati. Ova se provjera lako može uraditi unutar programa, te je demonstrirana u odjeljku 3.3.3.

3.3. Opis programskog okruženja i struktura programa

Programsko rješenje za automatsko pisanje teksta pomoću Mitsubishi robota rađeno je unutar MATLAB programskog okruženja. MATLAB nudi širok spektar ugrađenih funkcija i dodataka koji olakšavaju računanje kompleksnih operacija u vezi sa: direktnom i inverznom kinematikom, generisanjem vektora pozicije i orientacije, slanjem komandi za upravljanje robotom i slično. Programska rješenja obuhvataju glavnu aplikaciju i četiri dodatne aplikacije, koje se pokreću iz glavne aplikacije. Svaka od aplikacija smisleno je vezana za određeni dio zadatka. Aplikacije su programirane i dizajnirane korištenjem MATLAB App Designer okruženja i posjeduju grafički korisnički interfejs. U nastavku su aplikacije opisane detaljnije, sa fokusom na njihovoj funkcionalnosti. Ključni dijelovi izvornog koda prikazani su u isjećcima, a kompletan kod nalazi se na priloženom CD-u.

3.3.1. Glavna aplikacija

Glavna aplikacija predstavlja centralni dio programa i njen grafički korisnički interfejs prikazan je na slici 3.20. Unutar glavne aplikacije nalaze se osnovni elementi potrebni za izvršavanje pisanja pomoću robota.



Slika 3.20 – Grafički korisnički interfejs glavne aplikacije

Prilikom pokretanja glavne aplikacije automatski se pokreću i određene funkcije za inicijalizaciju parametara programa. To, između ostalog, podrazumijeva i inicijalizaciju parametara robota, simulacije, teksta i radnog okruženja. Ovi se parametri mogu modifikovati unutar neke od četiri dodatne aplikacije, te će biti kasnije obrazloženi. Ukoliko se izvrši modifikacija parametara, novo stanje parametara moguće je sačuvati u *.mat formatu pritiskom na dugme „Sačuvaj vrijednosti parametara“, te ih učitati pri ponovnom pokretanju glavne aplikacije pritiskom na dugme „Učitaj vrijednosti parametara“.

Unutar glavne aplikacije bira se tip tehničkog pisma, veličina fonta, jedinica veličine fonta *pt* (engl. *point*) ili *mm*, unosi se tekst koji treba da ispiše robot, a ponuđena je i opcija vraćanja robota u referentni položaj nakon što završi ispisivanje. Tekst za ispis potrebno je unijeti u tačno definisanom formatu. S obzirom na to da je u sklopu rada realizovano pisanje

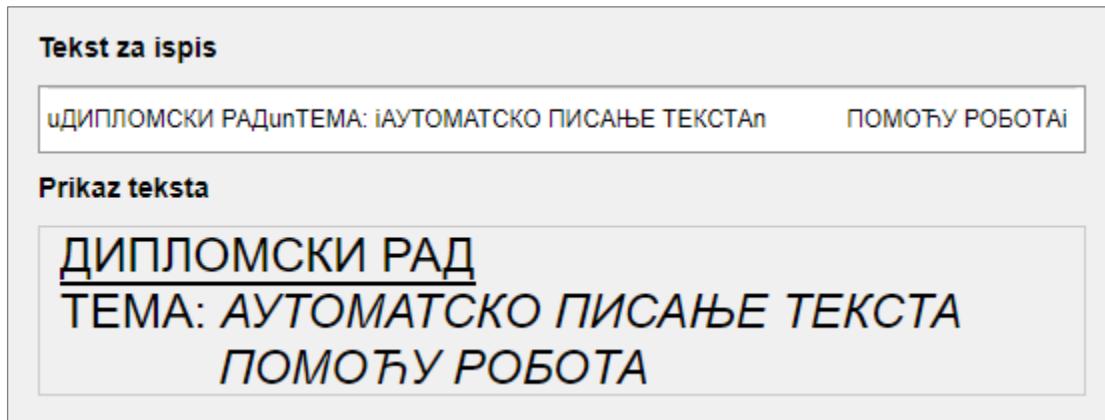
ćiriličnim tehničkim pismom, tekst se u predviđeno polje unosi u ćirilici, dok se određena latinična slova koriste kao specifikatori stila i formata, kako je prikazano u tabeli 4.

Specifikatori formata	Opis	Primjer unesenog teksta	Rezultat formatiranja
i	Specifikator stila za kurziv uvijek se stavlja u paru, tako da tekst koji treba da se ispiše u kurzivu bude između dva „i“ specifikatora. Odsustvo ovog specifikatora podrazumijeva obični (uspravni) stil pisanja.	iЕТФiБЛ	<i>ЕТФБЛ</i>
u	Specifikator formata za podvučene znakove uvijek se stavlja u paru, tako da tekst koji treba da je podvučen bude između dva „u“ specifikatora.	uЕТФuБЛ	<u>ЕТФБЛ</u>
s	Specifikator formata za precrte značajke uvijek se stavlja u paru, tako da tekst koji treba da je precrtan bude između dva „s“ specifikatora.	sЕТФsБЛ	ЕТФБЛ
d	Specifikator formata za dvostruko precrte značajke uvijek se stavlja u paru, tako da tekst koji treba da je dvostruko precrtan bude između dva „d“ specifikatora.	dЕТФdБЛ	ЕТФБЛ
n	Specifikator za prelazak u novi red. Stavlja se na kraj tekućeg reda.	ЕТФnБЛ	ЕТФ БЛ
Kombinovani specifikatori	Specifikatori stila i formata mogu se kombinovati, tako da se na određenom tekstu primjeni više efekata. Pri tome, potrebno je voditi računa da su specifikatori pravilno ugniježđeni kao u HTML-u.	uЕТФsБЛIsniУНИБЛiu	<u>ЕТФБЛ</u> <u>УНИБЛ</u>

Tabela 4 – Specifikatori stila i formata

Validnost unesenog teksta provjerava se unutar funkcije `checkTextValidity`. Ako je tekst validan, vrši se konverzija teksta u HTML format pomoću funkcije `text2html`, te se formatirani tekst prikazuje u HTML isječku na desnoj strani prozora, kako je prikazano na slici 3.21. Formatirani tekst služi kao provjera unesenog neformatiranog teksta, u smislu

postignutog stila (obični ili kurziv) i efekata (podvučeni, precrtni, ili dvostruko precrtni znakovi) u tekstu.



Slika 3.21 – Prikaz formatiranog teksta u HTML isječku

Unutar glavne aplikacije moguće je vršiti i simulaciju ispisa pomoću Mitsubishi RV-2SDB robota. Prilikom pokretanja glavne aplikacije učitavaju se i *.STL modeli segmenata robota, markera i površine za pisanje, koji se koriste u svrhu simulacije. Prije nego što se pokrene simulacija, prvo se izvršavaju određene funkcije za pripremu simulacije. Funkcija `generatePR`, uz pomoć funkcije `genchr`, generiše homogene prelazne matrice za svaku karakterističnu tačku pojedinog znaka unutar vektora `points`, dimenzija $4 \times 4 \times n$, kao i binarne maske `trails`, dimenzija $1 \times n$, koje definišu kada marker u toku simulacije ostavlja trag (vrijednost 1), a kada ne (vrijednost 0). Dio koda iz funkcije `genchr` prikazan je u isječku 1.

```
% Ulazni parametri:  
% app - Aplikacija koja poziva funkciju  
% chr - Znak koji treba biti generisan  
% it - Ako je 0 - obican tekst, ako je 1 - kurziv  
% ch - Nazivna visina znaka  
% cw - Sirina znaka  
% hf - Horizontalni pomjeraj u smjeru +y0  
% vf - Vertikalni pomjeraj u smjeru +x0  
% d - Sigurna udaljenost markera od podloge (kada ne crta liniju)  
% Izlazni parametri:  
% points - Generisane homogene prelazne matrice za položaj vrha manipulatora u  
% odnosu na referentni koordinatni sistem  
% trails - Binarna maska - kada će robot ostaviti trag na papiru (za simulator)  
function [points, trails] = genchr(app, chr, it, ch, cw, hf, vf, d)  
    rotper = [app.robot_params.tom; [0 0 0]]; % Lijevi 4x3 dio homogenih prelaznih  
    % matrica koje se generisu (orientacija alata i perspektiva) koji je isti za svaku  
    % matricu  
  
    switch chr % U zavisnosti od traženog znaka, generisu se razliciti vektori pozicije  
    % za svaku karakteristicnu tacku znaka (faktor skaliranja uvijek 1)  
    case 'A'  
        tana = ch/(3/6*cw);  
        transcl = [ 0 0 0 ;  
                    0 0 0 ;  
                    ch 3/6*cw 0 ] ;
```

```

          0           cw  0  1 ;
          0           cw  d  1 ;
5/18*ch   (5/18*ch)/tana d  1 ;
5/18*ch   (5/18*ch)/tana 0  1 ;
5/18*ch   cw-(5/18*ch)/tana 0  1 ;
5/18*ch   cw-(5/18*ch)/tana d  1 ];
pnum = 9;
trails = [0 0 1 1 0 0 0 1 0];

case 'B'
transcl = [
            ch  4/6*cw  d  1 ;
            ch  4/6*cw  0  1 ;
            ch  0  0  1 ;
            0   0  0  1 ;
            0   7/12*cw  0  1 ;
            3/18*ch 5/6*cw  0  1 ;
            7/18*ch 5/6*cw  0  1 ;
            10/18*ch 7/12*cw  0  1 ;
            10/18*ch 0  0  1 ;
            10/18*ch 0  d  1 ];
pnum = 10;
trails = [0 0 1 1 1 1 1 1 1 0];
% .
% .
% Na slican nacin su definisani i ostali znakovi
% .
% .
end

points = zeros(4, 4, pnum);
for i = 1 : pnum % Formiranje konacnog niza homogenih prelaznih matrica spajanjem
dijela za orijentaciju i dijela za translaciju
    points(:,:,i) = [rotper, transcl(i,:)'];
end
if it % Ukoliko je trazen kurziv, homogene prelazne matrice se modifikuju
    points(2,4,:) = points(2,4,:)+points(1,4,:)*tan(pi/12);
end
points(1,4,:) = -points(1,4,:) + vf; % Modifikacija x koordinata tacaka u skladu sa
trenutnim vertikalnim pomjerajem
points(2,4,:) = points(2,4,:) + hf; % Modifikacija y koordinata tacaka u skladu sa
trenutnim horizontalnim pomjerajem
end

```

Isječak 1 – Dio koda iz funkcije genchr

Nakon što se izvrši funkcija `generatePR`, pokreće se funkcija `simul` koja prvo bitno vrši povećanje prostorne rezolucije simulacije. To se postiže generisanjem novih tačaka, to jest, odgovarajućih homogenih prelaznih matrica između svake dvije karakteristične tačke znakova koji se ispisuju. Nakon toga se, pomoću inverzne kinematike za kretanje od tačke do tačke, izračunavaju unutrašnje koordinate robota za svaku prethodno generisalu homogenu prelaznu matricu. Dio koda iz funkcije `simul` u vezi sa inverznom kinematikom prikazan je u isječku 2.

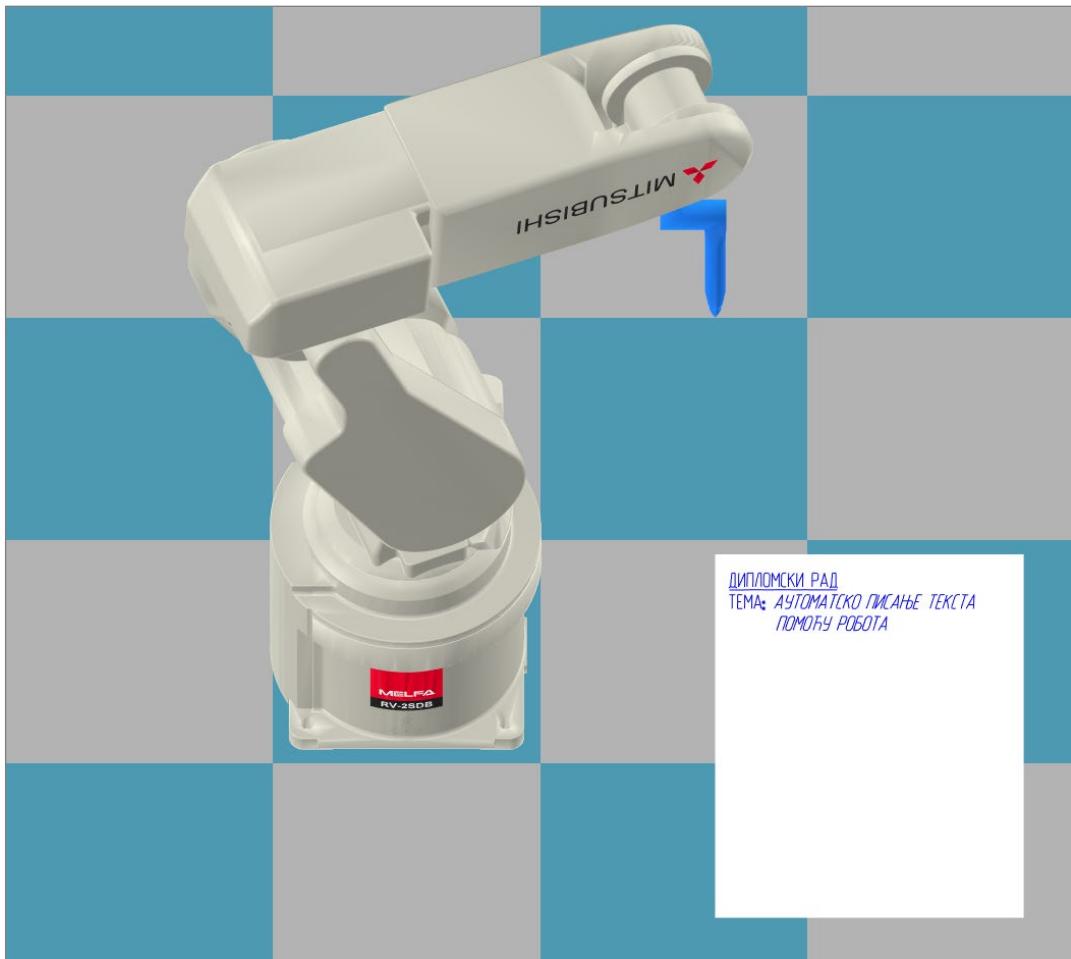
```

% Ulazni parametri:
% app - Aplikacija koja poziva funkciju
% Izlazni parametri:
% qs - Generisani niz unutrasnjih koordinata za svaku tacku na putanji
% ts - Binarna maska - kada ce robot ostaviti trag na papiru (za simulator)
function [qs,ts] = simul(app)
% ..... Dio za povecanje prostorne rezolucije .....
% Inverzna kinematika
q0 = app.robot_params.q; % Pocetna pozicija je trenutna pozicija robota
qfound = false; % Pokazatelj validnosti pronadjenog vektora unutrasnjih koordinata
qtemp = []; % Pomocni niz pronadjenih unutrasnjih koordinata
qs = [];
ts = [];
for i = 1 : size(p2p,2) % Za svaku rijec
    for j = 1 : size(p2p{1,i},2) % Za svaki znak unutar rijeci
        for k = 1 : size(p2p{1,i}{1,j},2) % Za svaku putanju izmedju dvije
            karakteristicne tacke znaka
                p = 1;
                qfailed = 0; % Broj promasenih vektora unutrasnjih koordinata
                while p <= size(p2p{1,i}{1,j}{1,k}{1,1},3) % Za svaku tacku na putanji
                    while ~qfound % Dok nije pronadjen odgovarajuci vektor unutrasnjih
                        koordinata
                            [qc, err] = app.robot_params.R.rikcon(p2p{1,i}{1,j}{1,k}{1,1}(:, :, p), q0);
                            % Inverzna kinematika za trenutni polozenje
                            if err < app.robot_params.rep % Ukoliko je pronadjen odgovarajuci vektor
                                unutrasnjih koordinata
                                    q0 = qc; % Pocetna pozicija se azurira
                                    qfound = true; % Pokazatelj validnosti se azurira
                                    qtemp = [qtemp; qc];
                                else % Ukoliko nije pronadjen odgovarajuci vektor unutrasnjih koordinata
                                    qfailed = qfailed + 1; % Broj promasaja se azurira
                                    if qfailed > 30 % Ukoliko je broj promasaja veci od 30, simulacija se
                                        prekida i izbacuje se greska
                                        error('Nije moguce izvrsiti simulaciju. Vrh alata ne moze da postigne
                                              odredjeni polozenje na putanji.');
                                    end
                                q0 = app.genric(app.robot_params.R.n, app.robot_params.R qlim); %
                                Generise se novi pocetni pogodak unutrasnjih koordinata
                                qtemp = [];
                                p = 0;
                                break;
                            end
                        end
                    end
                end
                p = p + 1;
                qfound = false;
            end
            % Azuriraju se niz unutrasnjih koordinata i binarna maska
            qs = [qs; qtemp];
            ts = [ts; zeros(size(p2p{1,i}{1,j}{1,k}{1,1},3),1)+p2p{1,i}{1,j}{1,k}{2,1}];
            qtemp = [];
        end
    end
end
% ..... Dio za uklanjanje uzastopnih duplih vektora unutrasnjih koordinata .....
end

```

Isječak 2 – Dio koda iz funkcije *simul* u vezi sa inverznom kinematikom

Nakon izvršenih funkcija `generatePR` i `simul`, pokreće se funkcija `initializeAxesFigure` koja otvara novi prozor i inicijalizuje parametre prikaza simulacije. Simulacija se potom pokreće pomoću funkcije `visualize3d` i prikazuje se u prethodno kreiranom prozoru. Neka je, na primjer, pokrenuta simulacija za ispis teksta sa slike 3.21 veličinom fonta *32 pt*. Rezultat simulacije prikazan je na slici 3.22.



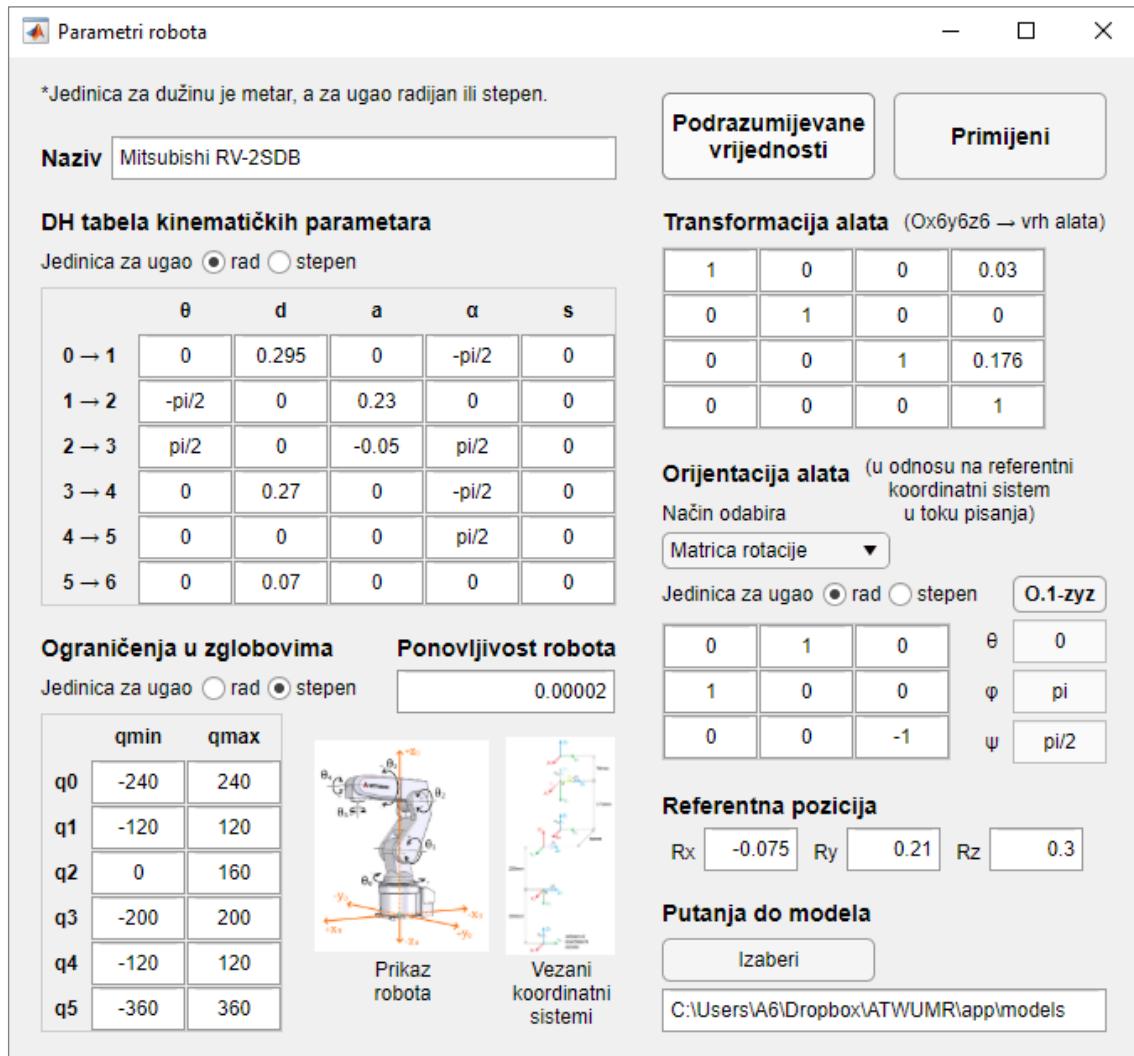
Slika 3.22 – Primjer rezultata simulacije

3.3.2. Aplikacija za podešavanje parametara robota

Parametri robota koji se mogu podešavati u sklopu ove aplikacije su sljedeći:

- Naziv robota;
- DH tabela kinematičkih parametara;
- Ograničenja u zglobovima;
- Ponovljivost robota;
- Homogena prelazna matrica za vrh alata (markera);
- Orientacija alata u toku pisanja;
- Referentna pozicija robota;
- Putanja do *.STL modela robota.

Grafički korisnički interfejs aplikacije za podešavanje parametara robota prikazan je na slici 3.23.



Slika 3.23 – Grafički korisnički interfejs aplikacije za podešavanje parametara robota

Parametri koji se odnose na dužinu unose se u metrima, dok je za ugao moguće birati radjan ili stepen. Podrazumijevane vrijednosti DH parametara, ograničenja u zglobovima i ponovljivosti robota u skladu su sa specifikacijom robota Mitsubishi RV-2SDB, te je njegov prikaz i vezane koordinate sisteme moguće vidjeti u sklopu aplikacije. Podrazumijevane vrijednosti ostalih parametara birane su u skladu sa korištenim vrijednostima u toku praktičnog dijela diplomskog rada. Sve parametre po potrebi je moguće modifikovati i prilagoditi konkretnom zadatku pisanja pomoću robota. Orijentaciju alata (markera) moguće je definisati na tri načina, i to pomoću rotacione matrice, Ojlerovih uglova ili aeronautičkih uglova. Ukoliko se izabere rotaciona matrica, automatski se, pomoću funkcije `rotmat2angles`, vrši i konverzija u određeni tip uglova koji su prikazani s desne strane matrice, koristeći formule opisane u odjeljku 2.3.2. S druge strane, ukoliko se izaberu Ojlerovi ili aeronautički uglovi, automatski se vrši konverzija u suprotnom smjeru, pomoću funkcije `angles2rotmat`, tako da je prikazana rotaciona matrica dosljedna izabranim uglovima. Na primjeru sa slike 3.24 demonstrirana je ova funkcionalnost.

<p>Orijentacija alata (u odnosu na referentni koordinatni sistem u toku pisanja)</p> <p>Način odabira</p> <p>Matrica rotacije</p> <p>Jedinica za ugao <input checked="" type="radio"/> rad <input type="radio"/> stepen</p> <table border="1" style="margin-left: 10px; margin-top: 10px;"> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>-1</td></tr> </table> <table border="1" style="margin-left: 10px; margin-top: 10px;"> <tr><td>θ</td><td>$\pi/2$</td></tr> <tr><td>φ</td><td>0</td></tr> <tr><td>ψ</td><td>π</td></tr> </table> <p>A.1-xyz</p>	0	1	0	1	0	0	0	0	-1	θ	$\pi/2$	φ	0	ψ	π	<p>Orijentacija alata (u odnosu na referentni koordinatni sistem u toku pisanja)</p> <p>Način odabira</p> <p>Ojlerovi uglovi</p> <p>Tip $z \rightarrow y' \rightarrow z'' / \dots$</p> <p>Jedinica za ugao <input checked="" type="radio"/> rad <input type="radio"/> stepen</p> <table border="1" style="margin-left: 10px; margin-top: 10px;"> <tr><td>0</td><td>-0.86602</td><td>0.5</td></tr> <tr><td>0</td><td>0.5</td><td>0.86602!</td></tr> <tr><td>-1</td><td>0</td><td>0</td></tr> </table> <table border="1" style="margin-left: 10px; margin-top: 10px;"> <tr><td>θ</td><td>$\pi/3$</td></tr> <tr><td>φ</td><td>$\pi/2$</td></tr> <tr><td>ψ</td><td>0</td></tr> </table> <p>O.1-zxy</p>	0	-0.86602	0.5	0	0.5	0.86602!	-1	0	0	θ	$\pi/3$	φ	$\pi/2$	ψ	0
0	1	0																													
1	0	0																													
0	0	-1																													
θ	$\pi/2$																														
φ	0																														
ψ	π																														
0	-0.86602	0.5																													
0	0.5	0.86602!																													
-1	0	0																													
θ	$\pi/3$																														
φ	$\pi/2$																														
ψ	0																														

a) Matrica rotacije

b) zyz tip Ojlerovih uglova

<p>Orijentacija alata (u odnosu na referentni koordinatni sistem u toku pisanja)</p> <p>Način odabira</p> <p>Aeronautički uglovi</p> <p>Tip $x \rightarrow y' \rightarrow z'' / \dots$</p> <p>Jedinica za ugao <input checked="" type="radio"/> rad <input type="radio"/> stepen</p> <table border="1" style="margin-left: 10px; margin-top: 10px;"> <tr><td>0.35355</td><td>-0.61237</td><td>0.70710</td></tr> <tr><td>0.86602!</td><td>0.5</td><td>0</td></tr> <tr><td>-0.35355</td><td>0.61237!</td><td>0.70710</td></tr> </table> <table border="1" style="margin-left: 10px; margin-top: 10px;"> <tr><td>θ</td><td>0</td></tr> <tr><td>φ</td><td>$\pi/4$</td></tr> <tr><td>ψ</td><td>$\pi/3$</td></tr> </table> <p>A.6-zyx</p>	0.35355	-0.61237	0.70710	0.86602!	0.5	0	-0.35355	0.61237!	0.70710	θ	0	φ	$\pi/4$	ψ	$\pi/3$	<p>$z \rightarrow y' \rightarrow x'' / z0 \leftarrow y0 \leftarrow x0 - \text{ROLL-PITCH-YAW}$</p> <p>$z \rightarrow x' \rightarrow y'' / z0 \leftarrow x0 \leftarrow y0$</p> <p>$y \rightarrow z' \rightarrow x'' / y0 \leftarrow z0 \leftarrow x0$</p> <p>$y \rightarrow x' \rightarrow z'' / y0 \leftarrow x0 \leftarrow z0$</p> <p>$x \rightarrow z' \rightarrow y'' / x0 \leftarrow z0 \leftarrow y0$</p> <p>$x \rightarrow y' \rightarrow z'' / x0 \leftarrow y0 \leftarrow z0 - \text{YAW-PITCH-ROLL}$</p>
0.35355	-0.61237	0.70710														
0.86602!	0.5	0														
-0.35355	0.61237!	0.70710														
θ	0															
φ	$\pi/4$															
ψ	$\pi/3$															

c) zyx tip aeronautičkih uglova

Slika 3.24 – Primjeri definisanja orijentacije alata

Za definisanje robota u programu korištena je *SerialLink* klasa iz *Robotics Toolbox* [11] dodatka za MATLAB, razvijenog od strane P. I. Corke. Ova klasa, između ostalog, omogućava i jednostavno zadavanje parametara robota, te izračunavanje direktnе i inverzne kinematike pomoću već implementiranih funkcija.

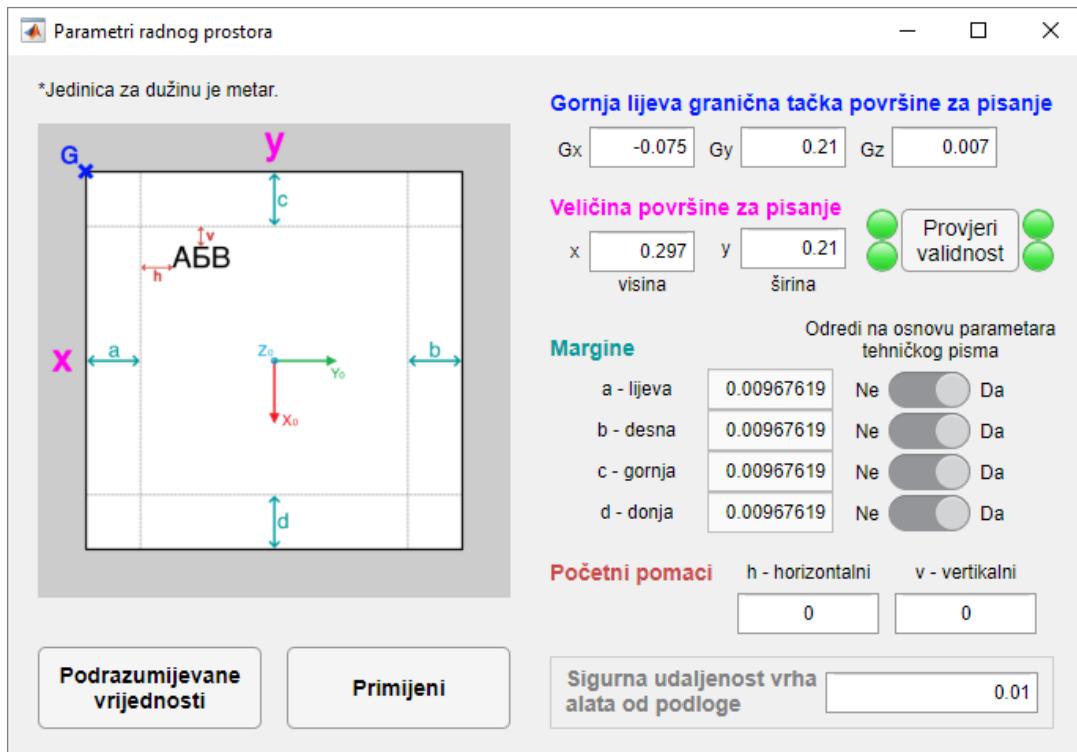
3.3.3. Aplikacija za podešavanje parametara radnog prostora

Parametri radnog prostora koji se mogu podešavati u sklopu ove aplikacije su sljedeći:

- Pozicija gornje lijeve granične tačke površine za pisanje;
- Veličina površine za pisanje;
- Margine površine za pisanje;
- Početni pomaci;
- Sigurna udaljenost vrha alata (markera) od podloge.

Grafički korisnički interfejs aplikacije za podešavanje parametara radnog prostora prikazan je na slici 3.25. Kao što je prethodno pomenuto površina za pisanje paralelna je sa x_0 - y_0 ravni referentnog koordinatnog sistema robota. Samim tim, njena visina definiše se u pravcu x_0 ose, a širina u pravcu y_0 ose referentnog sistema. Unutar grafičkog korisničkog interfejsa aplikacije nalazi se i ilustracija površine za pisanje sa svojim odrednicama i referentnim sistemom, što doprinosi jednostavnosti podešavanja parametara radnog prostora. Margine površine za pisanje moguće je birati proizvoljno ili automatski, na osnovu parametara tehničkog pisma. Početni pomaci mogu biti od koristi ukoliko je potrebno da ispis počne od tačno određenog mesta na površini za pisanje, a ukoliko su postavljeni na 0, ispis počinje od

gornjeg lijevog ugla definisanog marginama. Sigurna udaljenost vrha alata od podloge predstavlja udaljenost iznad površine za pisanje na koju se podiže vrh markera u toku pisanja, kada ne treba da ostavlja trag na površini.



Slika 3.25 – Grafički korisnički interfejs aplikacije za podešavanje parametara radnog prostora

Kao što je pomenuto u odjeljku 3.2.2. vrh markera mora biti u mogućnosti da dostigne sve četiri granične tačke površine za pisanje, da bi se svaki znak unutar granica površine za pisanje mogao uspješno ispisati. Ova provjera izvršava se pritiskom na dugme „Provjeri validnost“, oko kojeg se nalaze četiri lampice, koje daju vizuelnu informaciju o uspješnosti provjere za odgovarajuću graničnu tačku. Validnost se provjerava tako što se za svaku graničnu tačku, pomoću inverzne kinematike, pokušavaju izračunati unutrašnje koordinate robota na osnovu kojih bi marker dostigao položaj sa vrhom u posmatranoj tački. Odgovarajuća lampica za tačku koja se trenutno provjerava svijetli žutom bojom. Ukoliko je provjera uspješna, lampica zasvjetli zelenom bojom, a u suprotnom, crvenom bojom. Na slici 3.25 prikazan je slučaj pravilno definisane površine za pisanje, jer sve lampice nakon provjere svijetle zelenom bojom. Na slici 3.26 dat je primjer prikaza za slučaj nepravilno definisane površine za pisanje.



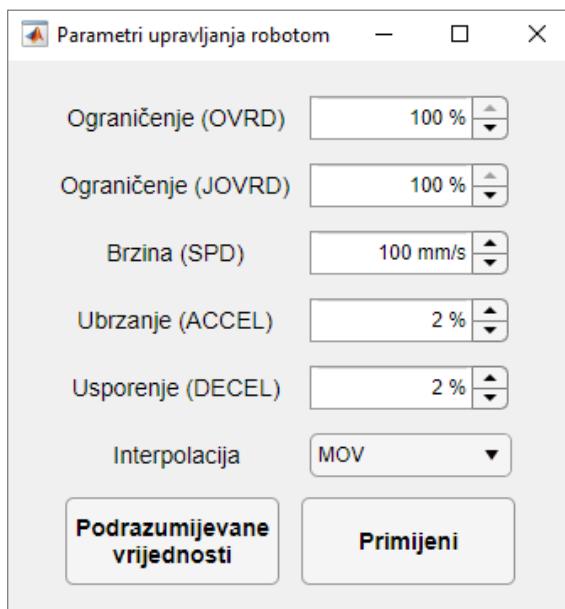
Slika 3.26 – Primjer prikaza za slučaj nepravilno definisane površine za pisanje

3.3.4. Aplikacija za podešavanje parametara upravljanja robotom

Parametri upravljanja robotom, u stvari, predstavljaju vrijednosti MELFA-BASIC V funkcija koje se prosljeđuju pri prenosu programa na robotski kontroler. U te parametre spadaju [10],[12]:

- Ograničenje brzine OV RD – Izražava se u procentima i predstavlja ograničenje brzine za sve metode interpolacije pri kretanju robota;
- Ograničenje brzine JOV RD – Izražava se u procentima i predstavlja ograničenje brzine za interpolaciju kretanja u unutrašnjim koordinatama;
- Brzina SPD – Izražava se u mm/s i predstavlja absolutnu brzinu vrha manipulatora za pravolinijsku i kružnu interpolaciju pri kretanju robota;
- Ubrzanje ACCEL – Prestavlja postotak maksimalnog ubrzanja pri kretanju robota;
- Usporenje DECEL – Predstavlja postotak maksimalnog usporenja pri kretanju robota;
- Vrsta interpolacije – MOV za interpolaciju u unutrašnjim koordinatama i MVS za pravolinijsku interpolaciju.

Grafički korisnički interfejs aplikacije za podešavanje parametara upravljanja robotom prikazan je na slici 3.27.



Slika 3.27 – Grafički korisnički interfejs aplikacije za podešavanje parametara upravljanja robotom

Kretanje robota može biti ostvareno kao kretanje od tačke do tačke ili kao kontinualno praćenje trajektorije. Korištenjem MOV komande ostvaruje se kretanje od tačke do tačke, pri čemu se vrši interpolacija u unutrašnjim koordinatama. Putanja između dvije tačke na ovaj se način ne može direktno kontrolisati. Ova vrsta interpolacije podrazumijeva određivanje vektora unutrašnjih koordinata za početnu i ciljnu tačku, a zatim računanje pomjeraja u zglobovima, tako da budu proporcionalni ukupnom potrebnom pomjeranju u njima, pri čemu

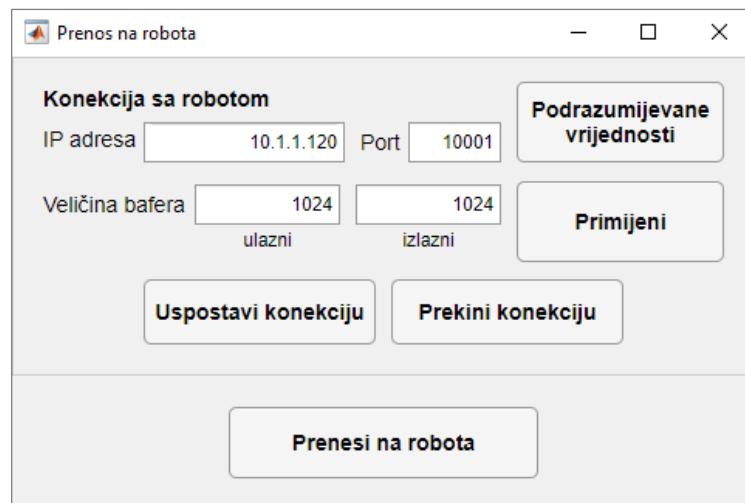
istovremeno treba da počnu i završe pomjeranje [13]. Ovaj način kretanja pogodno je koristiti, jer je računski dosta prostiji u odnosu na kontinualno praćenje trajektorije. Međutim, s obzirom na to da nije moguće kontrolisati putanju, ne preporučuje se zadavanje kretanja na ovakav način ukoliko se ispis vrši većom veličinom fonta. U tom slučaju preporučuje se korištenje MVS komande, koja podrazumijeva kontinualno praćenje trajektorije, tako da se vrh alata (markera) kreće pravolinijski između početne i ciljne tačke [13].

3.3.5. Aplikacija za prenos programa na robota

U sklopu ove aplikacije moguće je podešavati parametre TCP/IP konekcije između računara i robotskog kontrolera, i to:

- IP adresu;
- Port;
- Veličinu ulaznog bafera;
- Veličinu izlaznog bafera.

IP adresa i port podešavaju se u skladu sa vrijednostima parametara NETIP i NETPORT [12] robotskog kontrolera. Kao i MEXTL parametar, ove parametre je moguće modifikovati pomoću ručnog upravljača. Potrebno je takođe podesiti i TCP/IP parametre Ethernet konekcije na računaru na kojem se pokreću programi, tako da IP adresa računara pripada istoj lokalnoj mreži kao i IP adresa robotskog kontrolera, a podmrežna maska bude jednaka NETMSK [12] parametru robotskog kontrolera. Veličine ulaznog i izlaznog bafera ne treba mijenjati, osim u slučaju kada postoje posebni mrežni komunikacioni zahtjevi između računara i robotskog kontrolera. Na slici 3.28 prikazan je grafički korisnički interfejs aplikacije za prenos programa na robota.



Slika 3.28 – Grafički korisnički interfejs aplikacije za prenos programa na robota

Prilikom pokretanja ove aplikacije izvršava se i funkcija generatePR, koja pored homogenih prelaznih matrica, potrebnih za simulaciju, generiše i vektore položaja u obliku (2.27), pri čemu se orijentacija izražava u xyz varijanti aeronautičkih uglova. Pomoću ovih vektora i neke od komandi MOV ili MVS u sklopu programa zadaje se kretanje robota. Prije nego što se izvrši prenos programa, potrebno je uspostaviti konekciju sa robotskim

kontrolerom. Pritiskom na dugme „Uspostavi konekciju“ poziva se funkcija `openConnection`, koja inicijalizuje TCP/IP konekciju i vrši prenos osnovnih instrukcija na robotski kontroler, što između ostalog podrazumijeva paljenje servo motora robota i omogućavanje kontrole robota preko programa. Funkcija koja formira program i vrši prenos programa na robotski kontroler naziva se `robotcontrol` i prikazana je u isječku 3.

```
% Ulazni parametri:
% app - Aplikacija koja poziva funkciju
function robotcontrol(app)
    % Prvo se ucitava prethodno kreirani program sa robota
    fprintf(app.t,'1; 1;NEW');
    fprintf("%s\n", '1; 1;NEW');
    app.waitrobot();
    fprintf(app.t,'1; 1;LOAD=RITAN');
    fprintf("%s\n", '1; 1;LOAD=RITAN');
    app.waitrobot();
    fprintf(app.t,'1; 1;PRTVERLISTL');
    fprintf("%s\n", '1; 1;PRTVERLISTL');
    app.waitrobot();
    fprintf(app.t,'1; 1;PRTVEREMDAT');
    fprintf("%s\n", '1; 1;PRTVEREMDAT');
    app.waitrobot();

    % Zatim se brise sadržaj ucitanog programa
    fprintf(app.t,'1; 9;LISTL<');
    fprintf("%s\n", '1; 9;LISTL<');
    app.waitrobot();
    prog = '';
    lines = 0;
    len = app.t.BytesAvailable;
    if(len > 0)
        prog = [char(fread(app.t, len))'];
    end
    temp = extractAfter(prog, ';');
    lines = lines + str2num(extractBefore(temp, ';'));
    while(str2num(extractAfter(temp, ';')) == 1)
        fprintf(app.t, '1; 9;LISTL+1');
        fprintf("%s\n", '1; 9;LISTL+1');
        app.waitrobot();
        len = app.t.BytesAvailable;
        if(len > 0)
            prog = [char(fread(app.t, len))'];
        end
        temp = extractAfter(prog, ';');
        lines = lines + str2num(extractBefore(temp, ';'));
    end
    for i = 1 : lines-1
        prog = ['1; 9;EMDAT1' str2num('0xb')];
        fprintf(app.t, prog);
        app.waitrobot();
    end
    fprintf(app.t, '1; 9;EMDAT1');
    app.waitrobot();

    % Novi vektori položaja upisuju se u memoriju
    for i = 1 : size(app.caller_app.posrot,2) % Za svaku rijec
        for j = 1 : size(app.caller_app.posrot{1,i}, 2) % Za svaki znak u rijeci
            for k = 1 : size(app.caller_app.posrot{1,i}{1,j},2) % Za svaki vektor položaja u sklopu znaka

```

```

str = '';
for m = 1 : 6
    str = [str num2str(app.caller_app.posrot{1,i}{1,j}(m,k), '%.3f')];
    if m < 6
        str = [str ' ', ''];
    end
end
cm = ['1; 9;EMDATPR_W' num2str(i) 'C' num2str(j) 'N' num2str(k) ' = (' str
')(6, 0')];
fprintf(app.t, cm);
fprintf("%s\n", cm)
app.waitrobot();
end
end
end

% Formira se MELFA BASIC V program
prog = ['1; 9;EMDAT' ...
    '1 OVRD ' num2str(app.caller_app.robotcontrol_params.override, '%.1f') ...
    str2num('0xb') ...
    '2 JOVRD ' num2str(app.caller_app.robotcontrol_params.joverride, '%.1f') ...
    str2num('0xb') ...
    '3 SPD ' num2str(app.caller_app.robotcontrol_params.speed, '%.3f') ...
    str2num('0xb') ...
    '4 ACCEL ' num2str(app.caller_app.robotcontrol_params.accel, '%d') ',' ...
    num2str(app.caller_app.robotcontrol_params.decel, '%d') ...
    str2num('0xb')];
fprintf(t, prog);
fprintf("%s\n", prog)
waitrobot();
lnum = 5; % Tekuci broj linije u programu
for i = 1 : size(app.caller_app.posrot,2) % Za svaku rijec
    for j = 1 : size(app.caller_app.posrot{1,i}, 2) % Za svaki znak u rijeci
        for k = 1 : size(app.caller_app.posrot{1,i}{1,j},2) % Za svaki vektor poloza u
sklopu znaka
            prog = ['1; 9;EMDAT' num2str(lnum, '%d') ' '
app.caller_app.robotcontrol_params.interpolation_cmd ' PR_W' num2str(i) 'C' num2str(j)
'N' num2str(k) str2num('0xb')];
            fprintf(app.t, prog);
            fprintf("%s\n", prog);
            app.waitrobot();
            lnum = lnum + 1;
        end
    end
end
prog = ['1; 9;EMDAT' num2str(lnum, '%d') ' END'];
fprintf(app.t, prog);
fprintf("%s\n", prog);
app.waitrobot();

% Program se upisuje u memoriju i pokreće
fprintf(app.t,'1; 1;SAVE');
fprintf("%s\n", '1; 1;SAVE');
app.waitrobot();
fprintf(app.t,'1; 1;RUNRITAN;1');
fprintf("%s\n", '1; 1;RUNRITAN;1');
app.waitrobot();
end

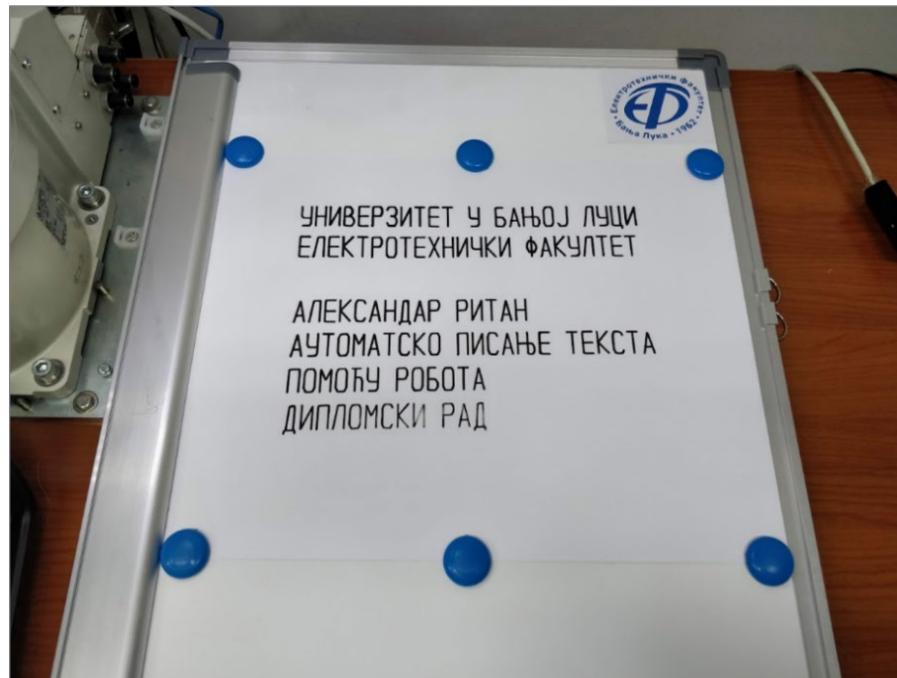
```

Isječak 3 – Funkcija robotcontrol

3.4. Rezultati

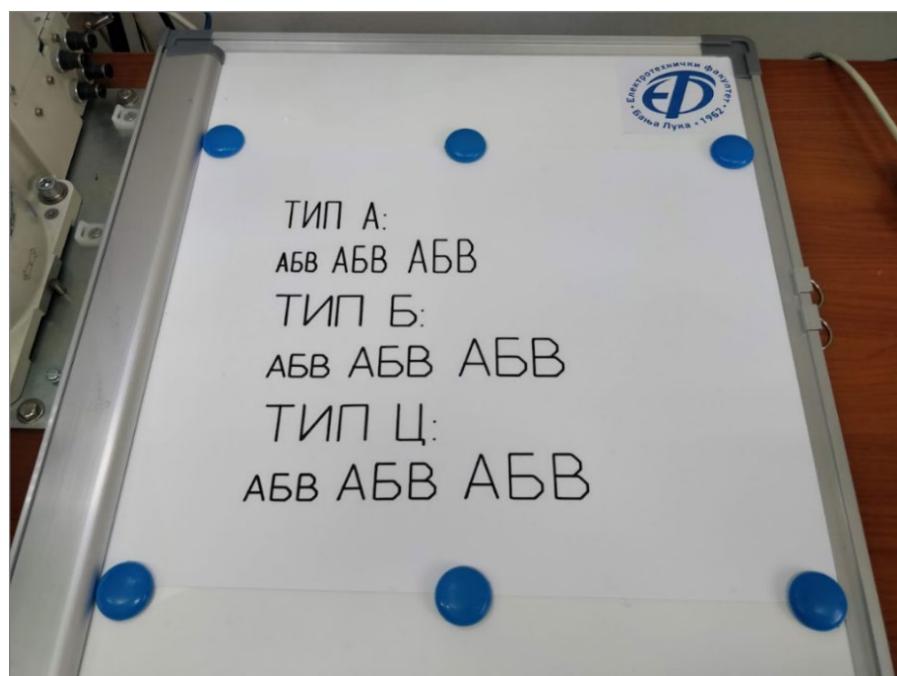
U ovom odjeljku su kroz nekoliko primjera prikazani rezultati ispisa pomoću robota koji su realizovani u laboratoriji. Primjerima su demonstrirane neke od funkcionalnosti programskog rješenja.

Na slici 3.29 prikazan je ispis teksta sa naslovne strane diplomskog rada.



Slika 3.29 – Ispis teksta sa naslovne strane diplomskog rada

Na slici 3.30 prikazani su ispisi pomoću različitih tipova tehničkog pisma.



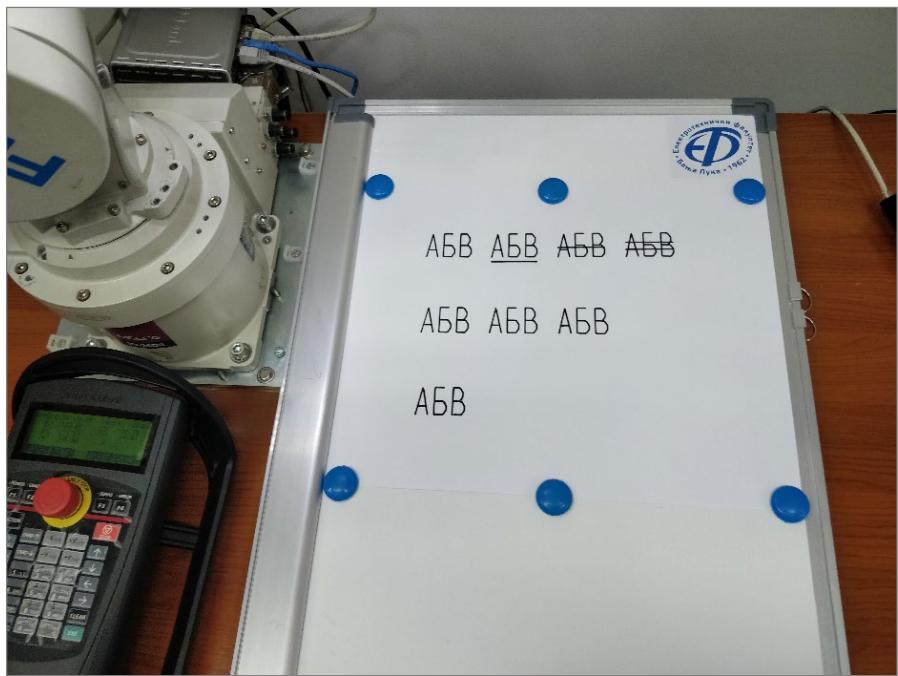
Slika 3.30 – Ispisi pomoću različitih tipova tehničkog pisma

Na slici 3.31 prikazan je ispis teksta „ОТОРИНОЛАРИНГОЛОГИЈА ПРОБА“. Ovim primjerom demonstrirano je automatsko prelamanje teksta na osnovu definisane veličine površine za pisanje. To podrazumijeva i prenos riječi u sljedeći red ukoliko ne može stati u tekući, pri čemu se na kraj tekućeg reda dodaje crtica.



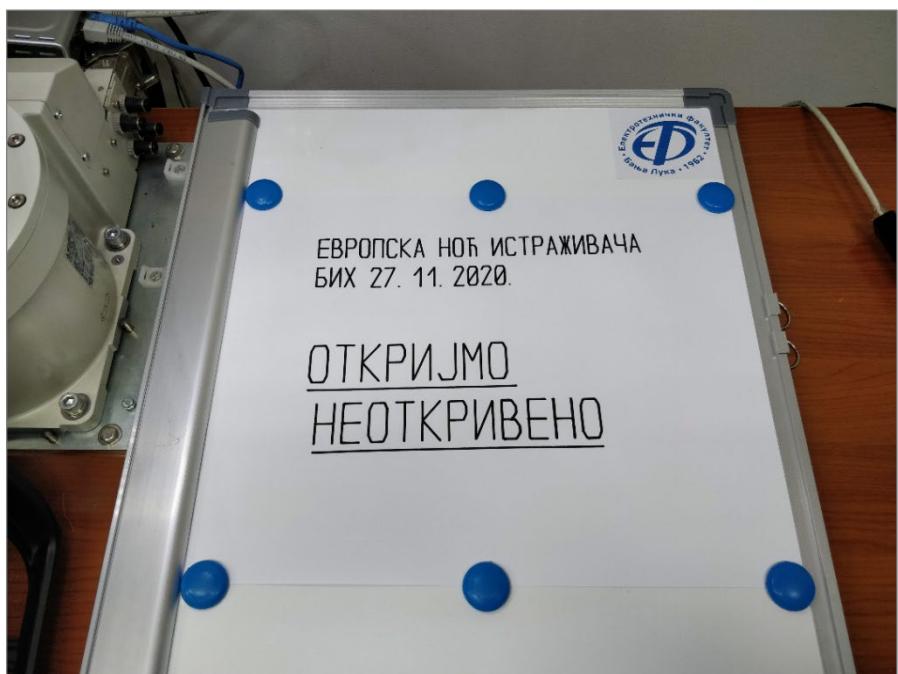
Slika 3.31 – Ispis koji demonstrira automatski prenos riječi u sljedeći red

Na slici 3.32 prikazani su ispisi koji demonstriraju različite funkcionalnosti u svakom redu. U prvom redu prikazani su različiti efekti teksta (podvučena, precrtačana i dvostruko precrtačana slova). U drugom redu, modifikacijom parametara OVRD i JOVRD uz interpolaciju u unutrašnjim koordinatama, ispis svake grupe slova „АБВ“ izvršen je različitom brzinom. U trećem redu demonstrirane su tačnost, preciznost i ponovljivost robota tako što je ispis grupe slova „АБВ“ izvršen nekoliko puta vraćajući se svaki put na početni horizontalni pomjeraj u redu. Uzimajući u obzir precizno definisana pravila pisanja tehničkog pisma, ovim je demonstrirana prednost korištenja robota u ovu svrhu, zbog odsustva tremora koji može da se manifestuje u slučaju pisanja ljudskom rukom.



Slika 3.32 – Ispisi koji demonstriraju različite funkcionalnosti u svakom redu

U saradnji sa prodekanom za naučnoistraživački rad prof. dr Vladimirom Risojevićem, kao i sa Veliborom Đalićem i sa Vedranom Jovanovićem, višim asistentima na Elektrotehničkom fakultetu u Banjoj Luci, rezultati diplomskog rada predstavljeni su i na Evropskoj noći istraživača 2020. godine [14]. U sklopu toga napravljen je video snimak [15] u kojem je opisan sistem za automatsko pisanje teksta pomoću Mitsubishi robota i u kojem su demonstrirane funkcionalnosti realizovanog programskog rješenja. Video snimak nalazi se i na priloženom CD-u. Na slici 3.33 prikazan je ispis uvodnog teksta za Evropsku noć istraživača BiH 2020. godine sa sloganom za tu godinu.



Slika 3.33 – Ispis uvodnog teksta za Evropsku noć istraživača BiH 2020. godine

4. Zaključak

U ovom radu opisan je sistem za automatsko pisanje teksta pomoću Mitsubishi RV-2SDB robota. U sklopu toga napravljen je teorijski uvod u kojem su objašnjeni neophodni pojmovi iz teorije mehanizama, homogene transformacije, način definisanja geometrije robotskog mehanizma, osnovni stavovi iz kinematike pomoću kojih se opisuje i zadaje kretanje robota, kao i parametri tehničkog pisma korištenog za pisanje pomoću robota. U vezi sa praktičnim dijelom zadatka opisane su osnovne komponente sistema i načini uspostavljanja komunikacije među njima, metode kalibracije sistema, te programsko rješenje realizovano korištenjem MATLAB programskog okruženja, koje ispunjava sve potrebne zahtjeve definisane tekstom zadatka diplomskog rada.

Programsko rješenje čini pet međusobno povezanih aplikacija sa grafičkim korisničkim interfejsom. Izložene su funkcionalnosti svake od aplikacija, a gdje je bilo potrebno prikazan je i isječak iz izvornog koda, koji demonstrira određenu funkcionalnost. Pomoću aplikacija moguće je podešavati parametre robota, radnog prostora, kontrole robota, konekcije sa robotom, teksta koji se ispisuje, a moguće je vršiti i simulaciju ispisa prije prenosa programa na robotski kontroler.

Ono što nije direktno prikazano u tekstu rada, a čemu je posvećena velika pažnja, jeste robušnost ovog programskog rješenja, koja se ogleda u njegovoј otpornosti na greške izazvane ljudskim faktorom. To, između ostalog, uključuje i vršenje provjera za sve unesene vrijednosti parametara od strane korisnika, a ukoliko se desi neispravan unos, korisniku se odgovarajućim bojama signalizira greška.

Glavna poteškoća prilikom realizacije sistema u sklopu diplomskog rada bio je manjak izvora u vezi sa programiranjem robota putem računara i podešavanjem različitih parametara, potrebnih za uspješnu komunikaciju sa robotskim kontrolerom. S ciljem rješavanja ovog problema korišten je Wireshark programski paket za analizu TCP/IP saobraćaja između računara sa originalnim softverom za programiranje i upravljanje robotom i robotskog kontrolera.

Ovim je zaokužena cjelina koja podrazumijeva realizaciju automatskog pisanja pomoću robota. Poboljšanje koje se može postići u odnosu na trenutnu realizaciju zadatka jeste uvođenje dodatnih MELFA-BASIC V funkcija, kao što su MVR, MVR2 i MVR3, kojima se kretanje robota može ostvariti kružnom interpolacijom. Na taj način moguće je realizovati pisanje koristeći proizvoljan font, čiji znakovi sadrže kružne isječke odgovarajućih radijusa ili druge vrste krivih linija. Korištenjem ovih funkcija, zajedno sa MOV i MVS, moguće je proširiti sistem i na pisanje različitih Unicode znakova, crtanje geometrijskih figura, tabela, grafikona, i slično.

5. Literatura

- [1] Veljko Potkonjak, *Robotika*, Naučna knjiga, Beograd, 1989.
- [2] Petar Marić, *Skripta iz robotike*, Elektrotehnički fakultet, Banja Luka
- [3] Jacques Denavit, Richard S. Hartenberg, *A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices* u: ASME Journal of Applied Mechanics, vol. 6, 1955, str. 215-221, dostupno na:
<https://home.konkuk.ac.kr/~cgkang/courses/robotics/courseMaterials/DHpaper.pdf>
- [4] Wisama Khalil, Etienne Dombre, *Modeling, Identification & Control of Robots*, Butterworth-Heinemann, 2004.
- [5] Internet izvor (Wikipedia), *Euler angles*, dostupno na:
https://en.wikipedia.org/wiki/Euler_angles
- [6] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, *Robotics: Modelling, Planning and Control*, Springer, 2009.
- [7] Internet izvor (Wikipedia), *atan2*, dostupno na: <https://en.wikipedia.org/wiki/Atan2>
- [8] Gregory G. Slabaugh, *Computing Euler angles from a rotation matrix*, dostupno na: <https://www.eecs.qmul.ac.uk/~gslabaugh/publications/euler.pdf>
- [9] Internet izvor, *Tehničko pismo*, dostupno na:
<https://tehnickocrtanje.blogspot.com/2011/01/tehnicko-pismo-iso-3098.html>
- [10] Mitsubishi Electric, *RV-2SD/2SDB Standard Specifications Manual (CRIDA-700 series Controller)*, dostupno na:
<https://eu3a.mitsubishielectric.com/fa/en/dl/7255/bfp-a8790.pdf>
- [11] Peter I. Corke, *Robotics, Vision & Control*, Springer, 2017.
- [12] Mitsubishi Electric, *CRnQ/CRnD Controller INSTRUCTION MANUAL (Detailed explanations of functions and operations)*, dostupno na:
<https://eu3a.mitsubishielectric.com/fa/en/dl/6350/bfp-a8661.pdf>
- [13] Chen Zhou, *ISYE 4256 SUPPLEMENTAL MATERIAL*, Georgia Institutue of Technology, 1999, dostupno na:
<https://www2.isye.gatech.edu/~czhou/MotionTypes.pdf>
- [14] Internet izvor, *Evropska noć istraživača BiH 2020. godine*, dostupno na:
<https://nocistrazivaca.ba/noc-istrazivaca-2020/>
- [15] Internet izvor, *Automatsko pisanje teksta pomoću MITSUBISHI robota*, video snimak na zvaničnom YouTube kanalu Evropske noći istraživača BiH, dostupno na:
<https://youtu.be/AgklqCYOPwQ>