

# Primjena SCARA robota s četiri stupnja slobode gibanja za potrebe raspoređivanja

---

**Puljić, Tajana**

**Master's thesis / Diplomski rad**

**2020**

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

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:108729>

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

*Download date / Datum preuzimanja:* **2024-12-19**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Tajana Puljić**

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentori:

Prof. dr. sc. Danijel Pavković, dipl. ing.

Student:

Tajana Puljić

Zagreb, 2020.

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

Zahvaljujem se svom mentoru, prof. dr. sc. Danijelu Pavkoviću, na podršci i savjetima pruženima tijekom pisanja diplomskog rada. Iznimno sam zahvalna što ste vjerovali u moje ideje, sposobnosti i moj rad te mi dali dovoljno prostora da izrazim vlastite želje u okviru ovog rada.

Nadalje, želim se zahvaliti mojim roditeljima, Leonardnu i Biljani, i mojoj sestri Tei. Vi ste me podupirali ne samo tijekom mog studiranja nego i tijekom cijelog mog života. Razgovori s vama i vaši savjeti uvijek su mi pomagali i usmjeravali me.

Veliko hvala i mojim prijateljima. Josipa, za razumijevanje svake sitnice i prije nego bih ju stigla objasniti i za svaku poteškoću koju si na taj način podijelila na pola. Studiranje na FSB-u za mene bi, bez tebe, bilo nezamislivo. Te Ivoni, Kristini i Jeleni za sve sate slušanja o temama koje nije uvijek bilo lako razumjeti i za svu pruženu potporu.

And last but not least, dank je wel Dalí. For believing in me and learning alongside me. For all the support and all the patience.

Tajana Puljić



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:  
procesno-energetski, konstrukcijski, brodstrojarski i inženjersko modeliranje i računalne simulacije

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602 - 04 / 20 - 6 / 3	
Ur. broj: 15 - 1703 - 20 -	

## DIPLOMSKI ZADATAK

Student: **Tajana Puljić** Mat. br.: 0035200101

Naslov rada na hrvatskom jeziku: **Primjena SCARA robota s četiri stupnja slobode gibanja za potrebe raspoređivanja**

Naslov rada na engleskom jeziku: **Application of a four degrees of freedom SCARA robot for arranging purposes**

Opis zadatka:

SCARA roboti zbog svoje relativne jednostavnosti konstrukcije i mogućnosti brzog pozicioniranja nalaze široku primjenu u industriji za probleme automatizirane manipulacije i premještanja artikala, kao i probleme sortiranja velikog broja elemenata. Shodno tome, u radu je potrebno osmisliti sustav sortiranja elemenata iz više neovisnih lokacija na novu precizno uređenu lokaciju primjenom SCARA robota kao izvršnog elementa.

U radu je potrebno napraviti sljedeće:

1. Ukratko opisati SCARA robotske strukture s više stupnjeva slobode, uz poseban osvrt i detaljniji opis ciljne SCARA robotske strukture sa četiri stupnja slobode gibanja.
2. Opisati simulacijsko okruženje za ispitivanje funkcionalnosti SCARA robota i kreirati odgovarajući simulacijski model SCARA robota sa četiri stupnja slobode gibanja.
3. Definirati problem sortiranja elemenata iz više neovisnih lokacija na novu precizno uređenu lokaciju.
4. Izraditi programski kod koji će sortirati elemente prema unaprijed definiranim pravilima za sortiranje i ispitati njegovu funkcionalnost u predloženom simulacijskom okruženju.
5. Izraditi 3D modele dijelova potrebnih za obavljanje prethodno navedene funkcije.

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

Zadatak zadan:  
30. travnja 2020.

Datum predaje rada:  
2. srpnja 2020.

Predviđeni datum obrane:  
6. – 10.7.2020.

Zadatak zadao:

Predsjednica Povjerenstva:

Izv. prof. dr. sc. Danijel Pavković

Prof. dr. sc. Tanja Jurčević Lulić

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS TABLICA.....	IV
POPIS ISJEČAKA KODA.....	V
POPIS TEHNIČKE DOKUMENTACIJE .....	VI
SAŽETAK.....	VII
SUMMARY .....	VIII
1. UVOD.....	1
2. SCARA.....	3
2.1. SCARA manipulator sa četiri stupnja slobode gibanja .....	5
2.2. Odabir SCARA manipulatora - IRB 910SC.....	6
2.3. Dijelovi sustava industrijskog manipulatora .....	8
3. SIMULACIJSKO OKRUŽENJE .....	10
3.1. RobotStudio .....	10
3.2. Direktna i inverzna kinematika.....	11
3.2.1. Direktna kinematika .....	11
3.2.2. Inverzna kinematika .....	14
4. RADNO OKRUŽENJE I ALATI.....	17
4.1. Radno okruženje .....	17
4.2. Alat za razmještanje.....	20
4.3. Robotska ćelija.....	24
5. UPRAVLJANJE SCARA ROBOTSKOM RUKOM.....	27
5.1. Radne pozicije manipulatora .....	28
5.1.1. Određivanje konfiguracija radnih pozicija.....	29
5.2. Premještanje komponente.....	31
5.3. Ispis premještanjem niza komponenata .....	33
6. INTERAKTIVNOST KROZ IGRU VJEŠALA.....	35
6.1. RADNO OKRUŽENJE .....	36
6.2. SUČELJE.....	39
6.3. UPRAVLJANJE.....	42
7. OPTIMIZACIJA POLOŽAJA KOMPONENATA.....	52

7.1. Frekvencija slova u riječima hrvatskog jezika.....	52
7.2. Standardno konfiguracijsko rješenje.....	53
7.3. Optimizacija položaja komponenata.....	55
8. ZAKLJUČAK.....	58
9. LITERATURA .....	59
PRILOZI.....	60

**POPIS SLIKA**

Slika 1	Princip razmještaja komponenata sa pozicije A na poziciju B .....	1
Slika 2	Rotacijski i translacijski zglobovi kinematičkog lanca .....	3
Slika 3	Minimalna struktura SCARA manipulatora .....	3
Slika 4	Struktura sfernog manipulatora .....	4
Slika 5	Radni prostor SCARA manipulatora .....	4
Slika 6	SCARA manipulatori tvrtke Staubli (stropno montirani) i tvrtke Mitsubishi .....	5
Slika 7	Prikaz strukture SCARA manipulatora sa 4SSG u nereferentnoj konfiguraciji .....	5
Slika 8	IRB 910SC-3/0.45 .....	7
Slika 9	Osnovne komponente sustava industrijskog manipulatora .....	8
Slika 10	Odnos virtualnih i stvarnih komponenata sustava (online i offline programiranje) ....	9
Slika 11	SCARA manipulator i virtualni upravljački uređaj unutar simulacijskog okruženja	10
Slika 12	Prikaz stupnjeva slobode gibanja IRB 910SC SCARA manipulatora .....	12
Slika 13	Koordinatni sustavi SCARA manipulatora sa 4SSG prema DH notaciji .....	12
Slika 14	Osam konfiguracijskih rješenja robota sa 6SSG za isti položaj end-effector-a .....	15
Slika 15	Problem višestrukosti rješenja SCARA manipulatora .....	15
Slika 16	Osnovni oblik i primjeri premještanja fizičkih komponenata .....	17
Slika 17	Radni prostor IRB 910SC-3/0.45 SCARA manipulatora .....	18
Slika 18	Raspored osnovnih komponenata radnog okoliša .....	18
Slika 19	Raspored komponenata na lijevoj i desnoj ploči .....	20
Slika 20	Primjer ispisa premještanjem komponenata na prednju ploču .....	20
Slika 21	Mjesto prihvata alata na IRB 910SC-3/0.45 SCARA manipulatoru .....	21
Slika 22	Exploded view prihvata hvataljke .....	22
Slika 23	Alat SCARA manipulatora .....	23
Slika 24	Alat pri pozicioniranju (lijevo) i prijemu (desno) .....	24
Slika 25	Robotska ćelija robota za razmještanje .....	25
Slika 26	Radna ploča na donjem okviru robotske ćelije .....	26
Slika 27	Gornji okvir robotske ćelije .....	26
Slika 28	Uređenje radnog okruženja diplomskog rada .....	27
Slika 29	Koordinatni sustavi R_R3_P8 radne pozicije .....	28
Slika 30	Dva konfiguracijska rješenja L_R1_P8 radne točke .....	29
Slika 31	Konfiguracijska rješenja: lakat lijevo i lakat desno .....	30
Slika 32	Put pomaka vršnog koordinatnog sustava u odnosu na zadanu točku .....	32
Slika 33	Položaj komponenata radnog okoliša kao rezultat ispisa ("SCARA") .....	34
Slika 34	Primjer igre "Vješala" .....	35
Slika 35	Sustav komponenata za brojanje neispravnih pokušaja pogađanja .....	37
Slika 36	Raspored komponenata radnog okoliša za igru "Vješala" .....	37
Slika 37	Prikaz aplikacije za igru vješala u glavnom izborniku .....	39
Slika 38	Osnovni ekran aplikacije za igru "Vješala" .....	40
Slika 39	Ekran za odabir slova aplikacije za igru "Vješala" .....	41
Slika 40	Definiranje duljine riječi pozicioniranjem komponenata bez oznaka .....	44
Slika 41	Primjer izgubljene igre "Vješala" .....	47
Slika 42	Primjer dobivene igre "Vješala" .....	50
Slika 43	Radne pozicije lijeve ploče za konfiguracijsko rješenje lakat desno .....	53
Slika 44	Usporedba putanja nakon promjene konfiguracijskog rješenja .....	54
Slika 45	Optimizirani položaj komponenata .....	55
Slika 46	Usporedba udaljenosti iste komponente na dvije različite pozicije .....	56



**POPIS TABLICA**

Tablica 1	Parametri čvorova SCARA manipulatora sa 4SSG .....	13
Tablica 2	Radni raspon zglobova IRB 910SC-3/0.45 SCARA manipulatora.....	16
Tablica 3	Dimenzije radnog prostora IRB 910SC-3/0.45 SCARA manipulatora .....	18
Tablica 4	Komponente uključene u radni okoliš.....	19
Tablica 5	Vrijednosti konfiguracijskih rješenja L_R1_P8 radne točke .....	30
Tablica 6	Komponente lijeve i desne ploče namijenjene za igru "Vješala" .....	38
Tablica 7	Učestalost slova u riječima hrvatskog jezika .....	52

**POPIS ISJEČAKA KODA**

Isječak koda 1	Prijevod položaja koordinate sustava u odnosu na manipulator .....	29
Isječak koda 2	Naredbe za prihvatanje komponente (na radnoj poziciji L_R5_P1).....	31
Isječak koda 3	Naredbe za premještanje komponente (na radnu poziciju F_R2_P7).....	32
Isječak koda 4	Proces ispisa riječi SCARA niza procesa premještanja komponenta....	33
Isječak koda 5	Boolean vrijednosti prema odabiru slova igrača .....	42
Isječak koda 6	"Vješala" – SCARA – dio I: prijenos praznih komponenta na prednju ploču .....	43
Isječak koda 7	"Vješala" – SCARA – dio II: zamjena komponenta prema odabiru igrača...	45
Isječak koda 8	"Vješala" – SCARA – potproces Vjesalo.....	46
Isječak koda 9	"Vješala" – SCARA – dio III: Uspostavljanje početnih uvjeta (izgubljena igra) .....	48
Isječak koda 10	"Vješala" – SCARA – potproces Vjesalo_Return .....	49
Isječak koda 11	"Vješala" – SCARA – dio IV: Uspostavljanje početnih uvjeta (dobivena igra) .....	51

## **POPIS TEHNIČKE DOKUMENTACIJE**

RC-00-00-00	Robotska stanica za razmještaj komponenata
RC-02-01-00	Alat manipulatora
RC-02-00-10	Osnovni oblik komponente

## SAŽETAK

Industrijske robotske ruke sve više čine sastavni dio automatizacije te se primjenjuju u sve širem spektru industrija. Sam proces automatizacije, u okviru primjene navedenih manipulatora, uključuje niz različitih disciplina među kojima se jednako ističu razvoj software-a i hardware-a. U okviru ovog diplomskog rada stoga je obuhvaćen razvojni proces koji uključuje konstrukciju potrebnih komponenata te kreiranje upravljačkog koda koji omogućuje upravljanje manipulatorom i alatima u svrhu obavljanja željenog zadatka. Odabrani se manipulator, SCARA robotska ruka sa četiri stupnja slobode gibanja, pri tome koristiti za raspoređivanje većeg broja komponenata sa početnih nedefiniranih pozicija na točno određene finalne pozicije. Za opisanu je pick&place primjenu SCARA manipulator, zbog jedinstvenog radnog prostora i specifičnih karakteristika, odabran kao optimalno rješenje. Premještane će komponente uglavnom predstavljati slova abecede čijim će se nizanjem na finalnoj lokaciji omogućavati interakcija s korisnikom. Kako bi omogućilo premještanje opisanih komponenata, osim samog alata za prihvat komponenata, razvijeni su dodatni dijelovi nužni za obavljanje definiranog zadatka, pozicionirani u radnom okolišu manipulatora. Kao glavni fokus diplomskog rada, stvaranje interaktivnog okoliša je, osim ispisa nizanjem određenog broja komponenata, ostvareno i u obliku interaktivne igre "Vješala" gdje odabrani SCARA manipulator zamjenjuje jednog od igrača nužnih za igranje odabrane igre. Manipulator pri tome, uporabom upravljačkog koda, obavlja zadatke odabira riječi za pogađanje, premještanja određene komponente ovisno o odabiru slova igrača te definiranje pobjede ili gubitka igre, nakon čega slijedi povrat komponenata na njihovu originalnu poziciju. U završnom su dijelu rada spomenuta potencijalna proširenja, nadogradnje i različite moguće primjene kreiranog sustava. Samo je rješenje predstavljeno i kreireno u okviru virtualnog okruženja odabranog ABB softwarea, RobotStudio, koji je odabran u kombinaciji sa IRB 910SC-3/0.45 SCARA manipulatorom. Ostatak je komponenata kreiran pomoću CAD softwarea, Solidworks.

Ključne riječi: industrijska robotska ruka, SCARA, manipulator, igra "Vješala", premještanje komponenata, interaktivni okoliš, RobotStudio,

## SUMMARY

Industrial robot arms are increasingly becoming an essential part of automation and are being implemented in a growing number of industries. The process of automation itself, including the application of aforementioned manipulators, consists of several different disciplines among which the development of the software and the hardware are equally important. This Master's thesis will therefore deal with the development processes, including both the design of necessary components and the control code which enables the control of the manipulator and adjacent tools, with the purpose of performing a desired task. The chosen manipulator, SCARA robot arm with four degrees of freedom is used within the scope of this work for the arranging of a relatively large number of components from undefined start positions to exactly defined final position. SCARA manipulator was chosen as an optimal solution for the described pick&place application due to its unique work envelope and specific attributes. The components, which are mostly representing alphabet letters, will be transferred to their final location subsequently, where they are used to facilitate straightforward interaction with the user. To enable the transfer of described components, beside the tool needed to grip the components, additional parts (required to fulfil the required task) were designed and then positioned inside of the manipulator work envelope. As the main focus of the thesis was to create an interactive environment, beside write-outs which are created by subsequently positioning a number of components, an interactive game of "Hangman" was created where the chosen SCARA manipulator acts as one of the players necessary for the chosen game. The manipulator, through the use of the control code, performs the following tasks: choosing a word which is to be guessed, repositioning of the components depending on the choice of the player and defining a victory or a loss of a game, after which the components are repositioned back to their original positions. In the final part of this thesis possible expansions, build-ups and different applications of the created system are mentioned. The solution itself is presented and created in a virtual environment of a chosen ABB software, Robotstudio, which was chosen alongside an IRB 910SC-3/0.45 SCARA manipulator. The rest of the components were created using a CAD software, Solidworks.

.Key words: industrial robot arm, SCARA, manipulator, "Hangman" game, component repositioning, interactive environment, Robotstudio

## 1. UVOD

Termin „robot“ prvi je puta uveden u naš vokabular u okviru Češke drame Karela Čapeka iz 1920. godine pod imenom „*Rossumovi univerzální roboti*“, gdje je riječ *robotnik* značila rob ili radnik, a bio je zamišljen kao čovjekoliki stroj sa mogućnošću rasuđivanja. Službena definicija robota u današnje vrijeme od strane Američkog Instituta za Robotiku (RIA) glasi: Robot je reprogramabilni multifunkcionalni manipulator konstruiran u svrhu premještanja materijala, dijelova, alata ili specijaliziranih uređaja kroz različite programirane kretnje u svrhu obavljanja raspona zadataka. [1]

Danas robotske ruke čine sastavni dio industrije automatizacije te se kao takve sve više nalaze u primjeni. Među najčešćim se primjenama pri tome ističu manipulacija i premještanje artikala, koji će se kao glavna tema istraživati i u okviru ovog diplomskog rada. Preciznije, istraživati će se mogućnosti primjene industrijskog manipulatora u svrhu raspoređivanja velikog broja neovisno pozicioniranih komponenata na novu uređenu lokaciju. Radom će se pri tome obuhvatiti centralni dio razvojnog procesa, odnosno obuhvatiti će se definiranje pojedinosti danog problema premještanja komponenata, razvoj komponentata i alata koji su potrebni za rješavanje danog problema te rješenje upravljanja odabranim manipulatorom u obliku upravljačkog koda.

<b>A</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	<b>P6</b>	<b>P7</b>	<b>P8</b>	<b>P9</b>	<b>P10</b>
<b>R1</b>	A	A	E	E	I	I	O	O	U	U
<b>R2</b>	B	C	D	F	G	H	J	K	L	
<b>R3</b>	M	N	P	R	S	T	V	Z		

<b>Pozicija A</b>	<b>R3P5</b>	<b>R2P2</b>	<b>R1P1</b>	<b>R3P3</b>	<b>R1P2</b>
<b>Pozicija B</b>	<b>R1P3</b>	<b>R1P4</b>	<b>R1P5</b>	<b>R1P6</b>	<b>R1P7</b>

<b>B</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	<b>P6</b>	<b>P7</b>	<b>P8</b>	<b>P9</b>	<b>P10</b>
<b>R2</b>			S	C	A	R	A			

Slika 1 Princip razmjestaja komponenata sa pozicije A na poziciju B

Sam zadatak manipulatora, koji prema osnovnim karakteristikama spada u pick&place kategoriju, očituje se u raspoređivanju većeg broja komponenata sa njihovih početnih pozicija na upravljačkim kodom određene finalne lokacije. Premještanje komponente pri tome čine sastavni dio zadatka, a u okviru rada biti će realizirane u obliku valjkastih komponenata na čijem će se vidljivim plohama nalaziti oznake, većinskim dijelom u obliku slova abecede. Razmještanjem će se navedenih komponenata, kao glavni cilj zadatka, omogućavati interakcija sa korisnikom.

Osnovni je princip zadatka prikazan na slici 1 gdje se komponente tablice A koriste kao baza dostupnih slova za ispis. Premještanjem niza komponenata, točno određenim redoslijedom, na nove uređene lokacije definirane redom i pozicijom tablice B omogućuje se ostvarivanje interakcije sa korisnikom, pri čemu se kao konačan rezultat dobije ispis kratice SCARA, kako jedne od okosnica ovoga diplomskog rada.

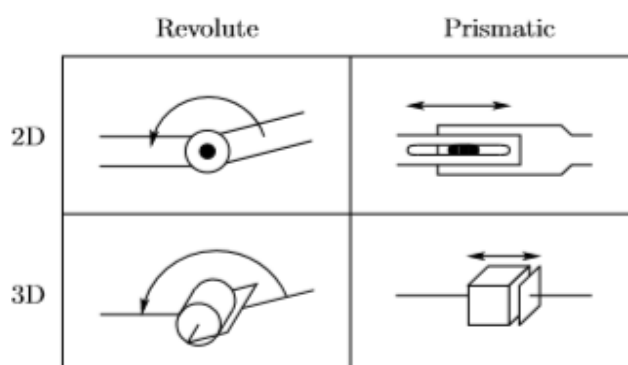
Stvaranje interaktivnog okoliša, osim samim ispisom pomoću dostupnih komponenata, realizirati će se i u obliku interaktivne igre "Vješala". U okvirima igre minimalan broj sudionika uključivati će jednog igrača te sam manipulator, koji će zamjenjivati drugog sudionika u igri. Uporabom različitih vrsta komponenata omogućiti će se uspostavljanje svih potrebnih uvjeta za definiranje pogađane riječi, pogađanje slova koji se nalaze u odabranoj riječi te ostatak koraka potrebnih za igranje, prema pravilima, klasične igre "Vješala".

Za obavljanje opisanih je zadataka, zbog jedinstvenog radnog prostora i specifičnih prednosti, odabran SCARA manipulator sa četiri stupnja slobode gibanja. Da bi se odabrana robotska ruka mogla koristiti za premještanje komponenata sa nedefiniranih početnih pozicija na točno uređene krajnje pozicije bilo je potrebno razviti alate, komponente i set naredbi koji će zajedničkim djelovanjem ostvarivati željenu funkciju. Oni će biti detaljno prikazani u okviru poglavlja koja slijede.

## 2. SCARA

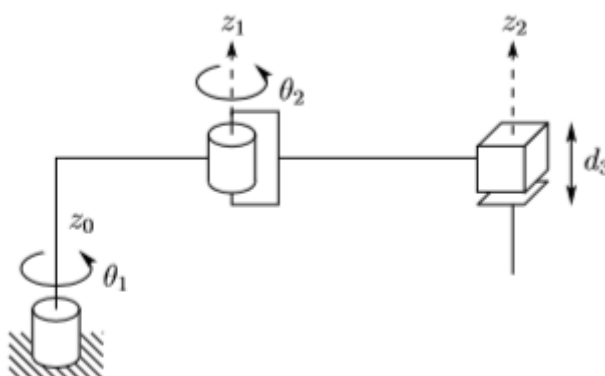
SCARA (Selective Compliant Articulated Robot for Assembly) robotska ruka popularni je manipulator kojega je 1980. godine razvio Hiroshi Makino. U trenutku razvoja, SCARA robotska ruka predstavlja je izniman napredak u području automatizacije što je utjecalo i na sam naziv robotske ruke.

Kao i svi robotski manipulatori, SCARA je sastavljena od segmenata (links), koji su međusobno povezani zglobovima (joints). Zajedno oni čine kinematički lanac. Rotacijski zglob (R) pri tome omogućava rotaciju između dva susjedna članka, a translacijski (T) omogućava relativni linearni pomak.



**Slika 2 Rotacijski i translacijski zglob kinematičkog lanca [1]**

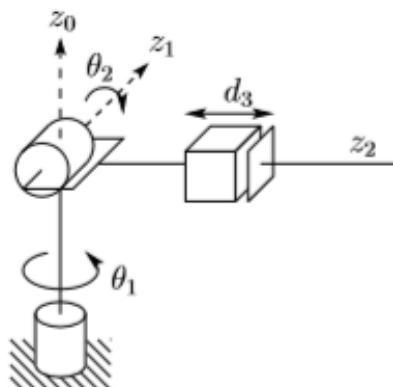
Minimalna konfiguracija SCARA robota, koja omogućuje pozicioniranje u prostoru, ima RRT strukturu, tj. sastoji se od dva rotacijska i jednog translacijskog zgloba. Varijablu rotacijskog zgloba označavamo sa  $\theta$ , a varijablu translacijskog zgloba sa  $d$ , predstavljajući pri tome relativni pomak između dva susjedna članka.



**Slika 3 Minimalna struktura SCARA manipulatora [1]**

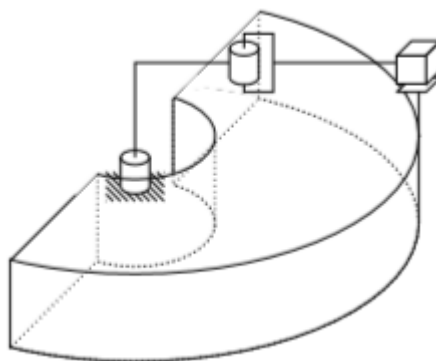


Unatoč tome što SCARA dijeli RRT strukturu sa sfernim manipulatorom, koji je nastao nešto ranije, inovativnost SCARA robotske ruke nalazi se u orijentaciji njenih zglobova.



**Slika 4** Struktura sfernog manipulatora [1]

Za razliku od sfernog manipulatora, SCARA ima osi  $z_0$ ,  $z_1$  i  $z_2$  paralelne što je uvelike utjecalo na mogućnosti primjene same robotske ruke, kao i na izgled radnog prostora. Ovako orijentirana RRT struktura omogućila je stvaranje velikog radnog prostora u x-y ravnini uz odgovarajući pomak u smjeru z-osi.



**Slika 5** Radni prostor SCARA manipulatora [1]

Među dodatnim prednostima SCARA manipulatora ističu se brzina, niska cijena, preciznost i kompaktnost, što ih čini jednim od najpopularnijih izbora u području montaže manjih dijelova (umetanje vijaka, montaža i demontaža), upravljanja dijelovima (pick&place, transport i upravljanje materijalom, pakiranje i sortiranje) te inspekcije (kontrola kvalitete, inspekcija dijelova i testiranje). [2]

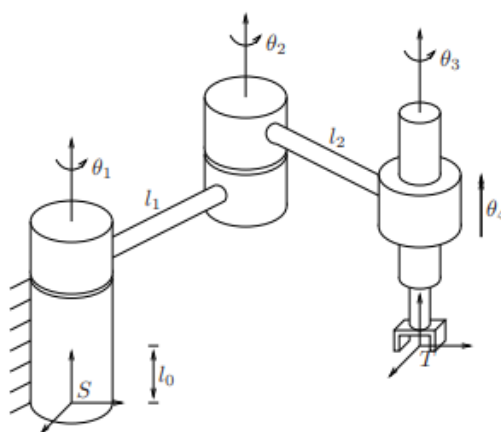
## 2.1. SCARA manipulator sa četiri stupnja slobode gibanja

Zbog njihove popularnosti SCARA robote nalazimo kao sastavni dio proizvodnog asortimana raznih proizvođača manipulatora, uz varijabilne nosivosti, raspon dohvata (ovisan o duljini članaka) te uz dodatnu mogućnost montiranja same robotske ruke na strop. Takva nam montaža omogućuje povećanje efikasnosti i fleksibilnost u slučajevima obavljanja zadataka u ograničenom prostoru.



**Slika 6 SCARA manipulatori tvrtke Staübli (stropno montirani) i tvrtke Mitsubishi**

Neovisno o odabranim svojstvima SCARA manipulatora, sam pojam SCARA (u današnje vrijeme) podrazumijeva manipulator sa četiri stupnja slobode gibanja. U odnosu na minimalnu strukturu SCARA manipulatora (slika 3) u kinematički je lanac dodan dodatni rotacijski stupanj slobode gibanja koji omogućuje rotaciju i bolje rukovanje predmetom kojim upravljamo.



**Slika 7 Prikaz strukture SCARA manipulatora sa 4SSG u nereferentnoj konfiguraciji [3]**

Zbog navedene strukture SCARA roboti imaju najbolje rezultate ponovljivosti u usporedbi sa svim drugim industrijskim robotima. Greške pozicije koje se javljaju na x-y

ravnini rezultat su samo dva motora (J1 i J2), a ne tri ili više kao što je obično slučaj kod drugih manipulatora. Prema tome su i greške pozicioniranja koje se javljaju manje.

Manji broj stupnjeva slobode gibanja uz sebe nosi i veću brzinu, u čemu se SCARA roboti ponovno izdvajaju kao najbolji izbor u odnosu na druge manipulatore. Pri uporabi inverzne kinematike, koja čini sastavni dio upravljanja industrijskim robotima, konfiguracija samog manipulatora u x-y osi određena je motorima J1 i J2, a položaj alata u smjeru osi z i njegova orijentacija određeni su motorima J3 i J4. Ovakva raspodjela pojednostavljuje proračune uključene u proces inverzne kinematike, te pozitivno utječe na smanjenje vremena potrebnog za njihovo izračunavanje, a samim time i na brzinu SCARA manipulatora.

S druge strane jednostavnost SCARA robotske ruke i uključeni broj stupnjeva slobode gibanja ograničavaju manipulator na uporabe koje ne zahtijevaju rotaciju oko x-osi i y-osi, koje za SCARA robotsku ruke nisu omogućene. Osim toga, u odnosu na druge manipulatore, SCARA ima relativno malu nosivost. Nakon oduzimanja mase koja odlazi na alat koji je nužan za obavljanje određenog zadatka (koji može biti jednostavna hvataljka ili kompliciraniji alat sa nekoliko namjena) ograničeni smo na rukovanje dijelovima manjih masa gdje SCARA roboti i nalaze najefikasniju primjenu. [4]

Zadatak obuhvaćem ovim diplomskim radom spada u "pick&plac" kategoriju, tj. pomicanje velikog broja dijelova male mase sa točke A na točku B. Većinskim je dijelom u to uključen pomak alata u smjerovima x,y i z uz moguću rotaciju oko z-osi, pri čemu SCARA omogućuje veliku brzinu temeljenu na jednostavnosti konstrukcije. Zbog prethodno navedenih razloga SCARA manipulator odabran je kao najefikasnije rješenje za obuhvaćeni zadatak. Pri tome su odabirom zadovoljeni svi nužni zahtjevi, uz zadržavanje maksimalnog stupnja jednostavnosti cijelog sustava.

## **2.2. Odabir SCARA manipulatora - IRB 910SC**

Među glavnim kriterijima za odabir odgovarajućeg SCARA manipulatora, uz nosivost, veličinu radnog prostora i odabir samog proizvođača, bilo je nužno odabrati i software u kojemu će SCARA manipulator biti programiran.

Software-i za offline (ili off-line) programiranje robota sve više čine sastavni dio automatizacije proizvodnje jer omogućavaju programiranje daleko od samih manipulatora.

Takvi se sustavi temelje na digitalnom blizancu robotske ruke kojime upravljamo na isti način kao i stvarnim robotom.

Programiranje se tada može izvesti na računalu, te se naknadno, uz minimalne prilagodbe primjenjuje na stvarni sustav. Među glavnim prednostima offline programiranja ističe se gotovo potpuno uklanjanje vremena potrebnog za programiranje na mjestu instalacije industrijskog robota (smanjujući pri tome troškove zaustavljanja proizvodnje). Unaprijed pripremljeni kod se, uz minimalne prilagodbe, primjenjuje na stvarnu robotsku ruku. [5]

Izbor proizvođača manipulatora za sobom često povlači i odabir samog software-a za offline programiranje, i obrnuto. Tako je npr. Roboguide razvijen od strane Fanuc-a za offline programiranje Fanuc robota, KUKA.Sim za Kuka robote, a MotoSim za Yaskawa Motoman.

U okviru ovog rada za rješavanje zadatka odabrani su ABB proizvodi, tj. RobotStudio (koji se ističe kao jedan od najčešće korištenih alata za simuliranje i offline programiranje) i IRB 910SC SCARA robotska ruka. Detaljniji opis i primjena RobotStudio-a detaljnije će biti objašnjeni u narednim poglavljima.

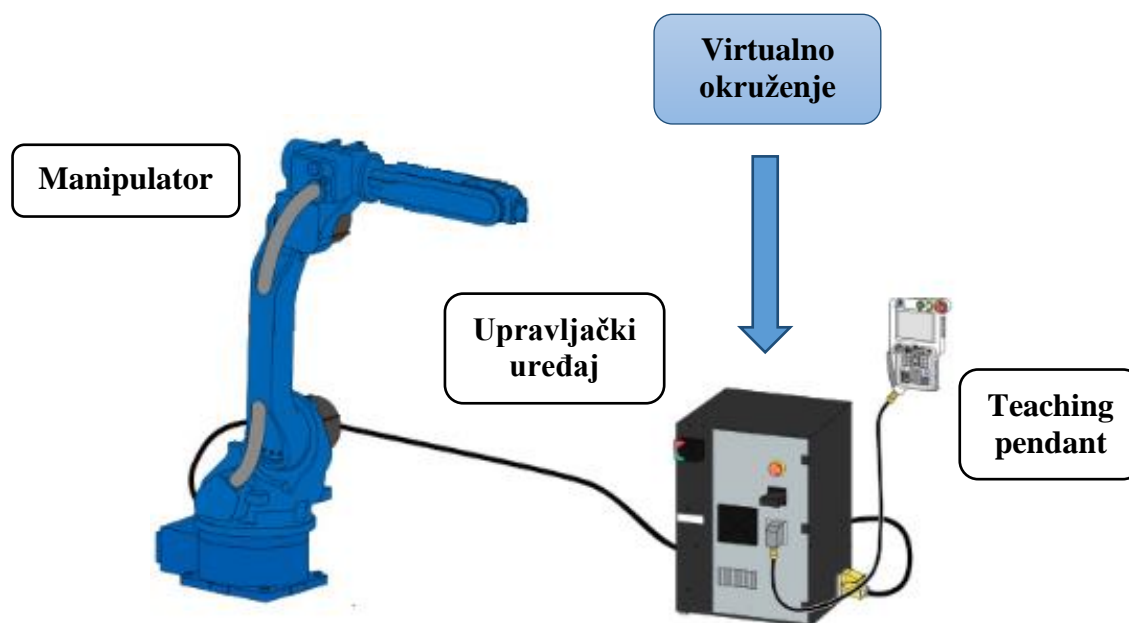


**Slika 8 IRB 910SC-3/0.45 [2]**

Nazivna nosivost IRB 910SC SCARA manipulatora iznosi 3 kg (maksimalno doseže do 6 kg), a među modelima dostupni su dohvat od 450, 550 i 650 mm. [2] Među ponuđenim je opcijama odabran dohvat od 450 mm koji je zadovoljio minimalne potrebne zahtjeve i pokazao se kao najefikasnije rješenje. Na temelju prethodno navedenih razloga manipulator IRB 910SC-3/0.45 odabran je za obavljanje zadatka u okviru ovog diplomskog rada.

### 2.3. Dijelovi sustava industrijskog manipulatora

Kako bi se određeni proces automatizirao korištenjem industrijske robotske ruke, osim samog manipulatora nužno su nam potrebne dodatne komponente koje omogućavaju upravljanje istim. Stoga se na mjestu instalacije, osim samog manipulatora, uvijek minimalno nalaze upravljački uređaj i programator (engl. Teaching peandant).



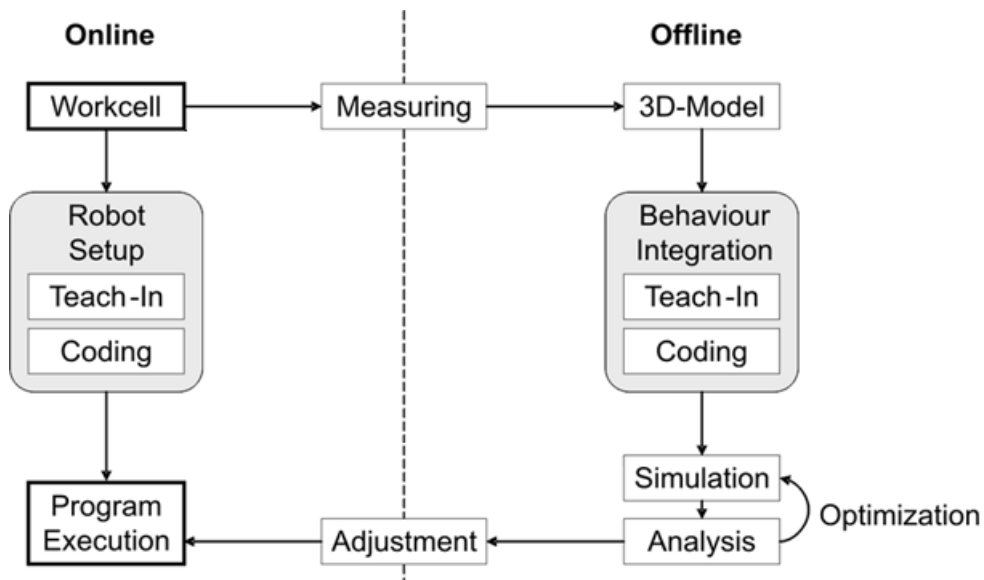
**Slika 9 Osnovne komponente sustava industrijskog manipulatora**

Teaching pendant pri tome služi kao sučelje koje omogućava kreiranje naredbi za upravljanje kretanjama manipulatora. Osim samog upravljanja u odnosu na lokalne i globalne koordinatne sustave te upravljanje pojedinačnim zglobovima, omogućava stvaranje upravljačkih programa koji osiguravaju obavljanje željenog zadatka na mjestu instalacije.

Upravljački uređaj naredbe kretanja stvorene na Teaching Pendant-u prevodi u kretnje mehaničkih sustava manipulatora te na taj način njime direktno upravlja. U tom je smislu upravljački uređaj mozak industrijskog manipulatora. On omogućava pohranjivanje informacija o samom manipulatoru, njegovom radnom prostoru i skupovima naredbi koji zatim omogućavaju upravljanje odabranom robotskom rukom.

U okviru odabranog virtualnog okoliša se stoga, kao sastavne komponente sustava, nalaze i upravljački uređaj, ali i virtualni Teaching penadant. Oba će se u okviru ovog diplomskog rada koristiti za obavljanje različitih koraka zadanog zadatka. Njihova će primjena biti vidljiva u poglavljima koja slijede.

Odnos virtualnog i stvarnog sustava, tj. online i offline programiranja prikazan je na slici 10. Prema slici je vidljivo da raspoložive komponente omogućavaju programiranje na mjestu instalacije, no korištenjem istih u virtualnom sustavu u proces se dodaju prednosti.



**Slika 10 Odnos virtualnih i stvarnih komponenata sustava (online i offline programiranje)**

Za odabrani se IRB 910SC SCARA manipulator koristi IRC5 Compact upravljački uređaj i ABB Teaching Pendant. Oni će u okviru obabranog virtualnog software-a biti prikazani u ograničenim kapacitetima koji su dovoljni za obavljanje željenih zadataka.

U okviru ovog će se diplomskog zadatka, zbog određenih ograničenja, koristiti isključivo virtualne verzije navedenih uređaja u oblicima u kojima se one nalaze unutar virtualnog okruženja.

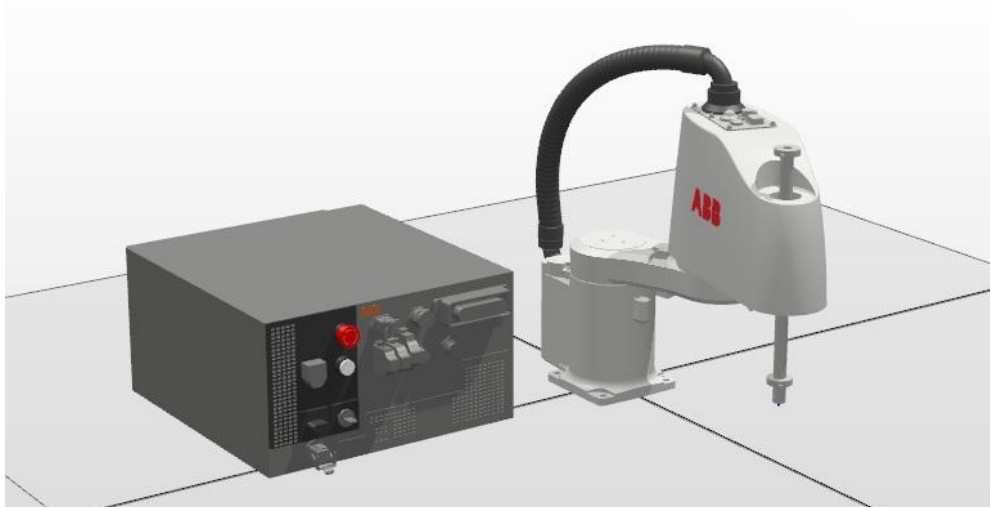
### 3. SIMULACIJSKO OKRUŽENJE

Prema prethodnom poglavlju, u okviru ovog diplomskog rada za upravljanje IRB 910SC-3/0.45 SCARA manipulatorom koristit će se software za offline programiranje, RobotStudio.

#### 3.1. RobotStudio

Kao što je već spomenuto, alati za offline programiranje čine sastavni dio automatizacije proizvodnje jer omogućavaju provedbu programiranja industrijskih robota unutar virtualnog okruženja na računalu. RobotStudio se temelji na ABB virtualnom upravljačkom uređaju (koji sadrži egzaktnu kopija izvršnog koda koji inače upravlja manipulatorom u proizvodnji) i na digitalnom blizancu same robotske ruke.

Mogućnost implementacije konstrukcije identične radnom okolišu u kojemu će se robot nalaziti omogućuje nam stvaranje vrlo realističnih programa. [6] Program upravljanja se zatim iz virtualnog okruženje prenosi na upravljački uređaj i upravlja stvarnom robotskom rukom, u ovom slučaju IRB 910SC-3/0.45 SCARA manipulatorom.



**Slika 11 SCARA manipulator i virtualni upravljački uređaj unutar simulacijskog okruženja**

RobotStudio, kao i ostali alati za offline programiranje, sa sobom nosi brojne prednosti koje utječu na smanjenje troškova te na sam proces razvoja proizvoda. Osim toga što je njihovom uporabom gotovo potpuno izbjegnuto vrijeme koje je potrebno za programiranje robota na mjestu instalacije (smanjujući na taj način troškove zaustavljanja proizvodnje) omogućeno je i optimiziranje samog procesa programiranja.

S obzirom da simulaciju temeljenu na stvarnom kodu možemo testirati u trenutku kada fizikalni dijelovi još ne postoje, tj. testiramo program paralelno sa razvojem konstrukcije, unaprijed možemo predvidjeti greške koje bi se javile programiranjem isključivo na mjestu instalacije. U istom simulacijskom okruženju možemo ispitati i veliki broj različitih scenarija te na taj način optimizirati putanje korištenog manipulatora. [5]

Programski jezik koji se koristi za upravljanje ABB industrijskim robotima naziva se RAPID. Razvijen je od strane ABB Grupe 1994. godine kada je naslijedio programski jezik ARLA, te spada u high-level programske jezike.

U okviru ovog diplomskog rada, programski se kod neće implementirati na stvarnu IRB 910SC-3/0.45 robotsku ruku (zbog različitih ograničenja), tj. kod razvijen u virtualnom okruženju RobotStudia predstavljati će finalnu točku u primjeni manipulatora za obavljanje zadanog zadatka. Daljnja implementacija na stvarni manipulator prema tome ostaje kao dodatna nadogradnja na ovaj diplomski rad.

### **3.2. Direktna i inverzna kinematika**

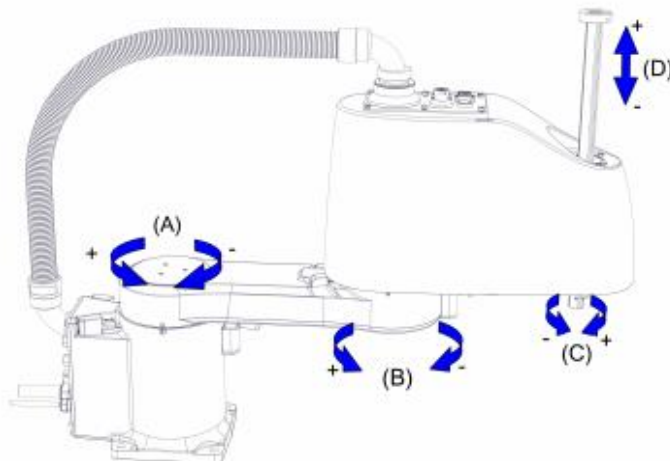
Sustav upravljanja industrijskim manipulatorima temelji se na direktnoj i inverznoj kinematici. One nam u suštini omogućavaju zadavanje pokreta za objekte sačinjene u obliku kinematičkog lanca, što se odnosi i na odabrani ABB SCARA manipulator. Kinematski je opis prema tome geometrijske prirode. Osnove ovih principa nužne su za razumijevanje rada u virtualnom okruženju (koje će biti predstavljeno u poglavljima koja slijede) zbog čega će u okviru ovog poglavlja glavna problematika biti objašnjena na korištenoj SCARA robotskoj ruci sa četiri stupnja slobode gibanja.

#### **3.2.1. Direktna kinematika**

Direktna ili unaprijedna kinematika temelji se na određivanju pozicije i orijentacije end-effector-a (vrha manipulatora, tj. vršnog koordinatnog sustava) uz poznate vrijednosti varijabli kinematičkog lanca. Korišteni SCARA manipulator ima četiri stupnja slobode gibanja u obliku RRTR strukture vidljive na slici 13. Poznajemo li relativne kuteve zakreta sva tri rotacijska zgloba i relativni pomak translacijskog, pozicija i orijentacija end-effectora je uvijek jednoznačno određena.

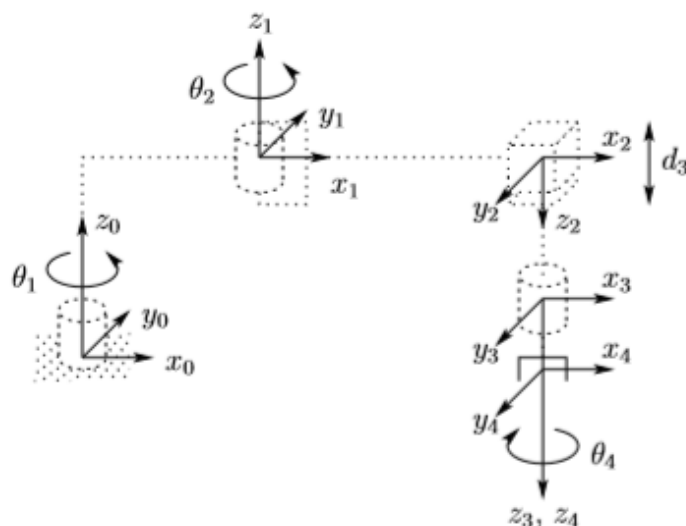


Cilj unaprijedne kinematike je prema tome analiziranje kumulativnog utjecaja dobivenog kao rezultat niza djelovanja pojedinih varijabli, u svrhu određivanja krajnjih vrijednosti. [1]



**Slika 12 Prikaz stupnjeva slobode gibanja IRB 910SC SCARA manipulatora**

U svrhu provođenja kinematske analize, za svaki od segmenata čvrsto povezujemo koordinatni sustav. Manipulator sa  $n$  brojem zglobova (stupnjeva slobode) imati će  $n+1$  koordinatnih sustava, s tim da prvi koordinatni sustav uvijek postavljamo u bazu manipulatora. Taj koordinatni sustav uvijek označavamo sa  $0$ .



**Slika 13 Koordinatni sustavi SCARA manipulatora sa 4SSG prema DH notaciji**

Korištena IRB 910SC-3/0.45 robotska ruka sastoji se prema tome od pet međusobno ovisnih koordinatnih sustava gdje je pozicija i orijentacija end-effector-a predstavljena koordinatnim sustavom sa sufiksom "4". Vrijednosti pozicije i orijentacije end-effector-a

određene su vrijednostima pozicija i orijentacija svih koordinatnih sustava koji prethodne vršnom u nizu kinematičkog lanca. Koordinatni sustav  $o_0x_0y_0z_0$ , koji je pozicioniran u bazi manipulatora, nepomičan je i smatra se referentnim sustavom u odnosu na koji su definiraju svi drugi koordinatni sustavi.

U slučaju korištenja jednostavnih manipulatora, sa npr. dva stupnja slobode gibanja, problematika direktne kinematike rješava se relativno lako. Uporabom jednostavnih trigonometrijskih funkcija kroz jednadžbe unaprijedne kinematike, uz zadane je varijable, relativno lako doći do pozicije i orijentacije vršnog koordinatnog sustava. No svakim se dodatnim stupnjem slobode gibanja povećava kompliciranost cijelog sustava i samih jednadžbi. Zbog toga za manipulatore sa više stupnjeva slobode gibanja, uključujući SCARA robotsku ruku sa četiri stupnja slobode gibanja, koristimo Denavit-Hartenberg notaciju koja nam omogućava sistematsku transformaciju među koordinatnim sustavima kinematskog lanca. Denavit-Hartenberg notacija optimizira pozicioniranje koordinatnih sustava u svrhu smanjena vremena potrebnog za računanje potrebnih funkcija te se samim time omogućava i brže upravljanje samim manipulatorom.

Denavit-Hartenberg notacija smatra se standardom za određivanje položaja i orijentacije koordinatnih sustava u području robotike. Prema ovoj notaciji transformacijska matrica svaka dva susjedna čvora predstavljena je kao produkt četiri osnovne transformacije, gdje su četiri parametra  $\theta_i$ ,  $a_i$ ,  $d_i$  i  $\alpha_i$  vezana uz zglobove dvaju susjednih segmenata. Parametri  $a_i$ ,  $\alpha_i$ ,  $d_i$  i  $\theta_i$  redom se nazivaju: duljina segmenta, zakret segmenta, pomak segmenta i zakret zgloba.

**Tablica 1 Parametri čvorova SCARA manipulatora sa 4SSG**

<i>Segment</i>	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	$a_1$	0	0	$\theta_1^*$
2	$a_2$	180°	0	$\theta_2^*$
3	0	0	$d_3^*$	0
4	0	0	$d_4$	$\theta_4^*$

Uporabom Denavit-Hartenberg notacije pozicionirani su koordinatni sustavi kinematičkog lanca SCARA manipulatora, vidljivi na slici 13, prema čemu su zatim određeni parametri svih čvorova vidljivi u tablici 1. Parametri  $\theta^*$  i  $d^*$ , vidljivi u tablici, predstavljaju

varijable zglobova koje ovise o poziciji robotske ruke, čijim je poznavanjem određen položaj i orijentacija end-effector-a.

Uporabom parametara iz tablice 1 razvijamo matrice transformacije između svih parova susjednih segmenata. Množenjem četiri uzastopne matrice transformacije kinematičkog lanca SCARA manipulatora dobivamo:

$$T_4^0 = \begin{bmatrix} c_{12}c_4 + s_{12}s_4 & -c_{12}s_4 + s_{12}c_4 & 0 & a_1c_1 + a_2c_{12} \\ s_{12}c_4 - c_{12}s_4 & -s_{12}s_4 - c_{12}c_4 & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & -1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Ukupna homogena matrica transformacije  $T_4^0$  sadrži jednadžbe direktne kinematike koje su potrebne za jednoznačno određivanje položaja vršnog koordinatnog sustava u odnosu na bazu manipulatora, uz poznavanje varijabli svih čvorova kinematičkog lanca. Navedene jednadžbe osnova su na kojoj leži upravljanje IRB 910SC-3/0.45 SCARA robotskom rukom.

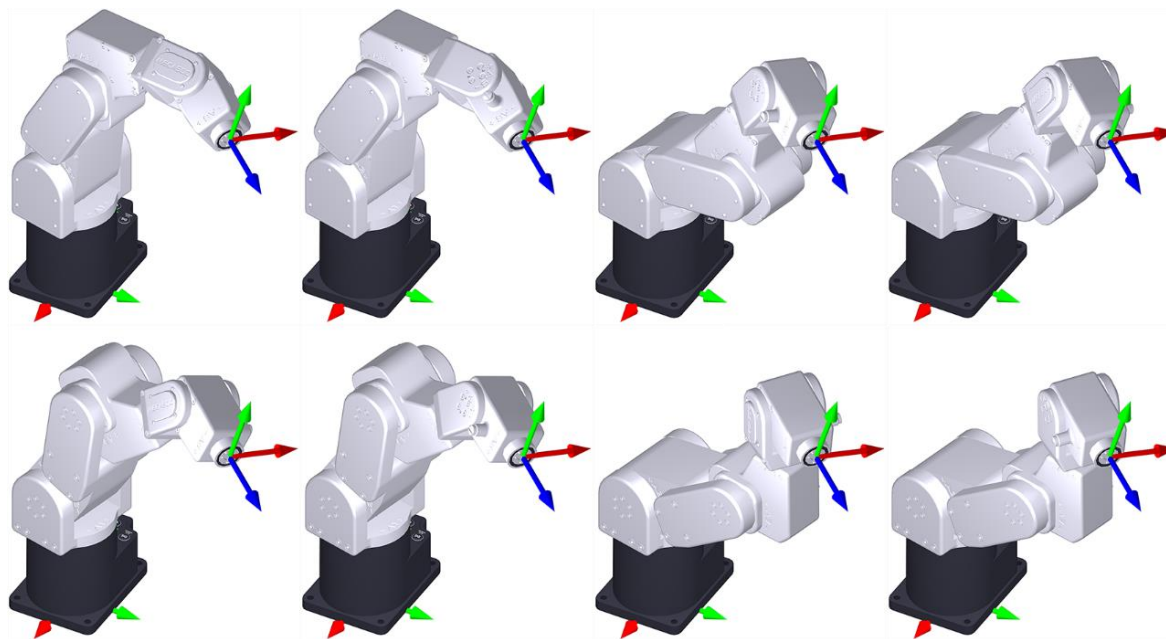
### 3.2.2. Inverzna kinematika

U prethodnom je poglavlju objašnjen proces direktne kinematike, tj. određivanje položaja i orijentacije end-effectora prema poznatim varijablama zglobova. Ovo će se poglavlje baviti suprotnim procesom, tj. određivanjem varijabli zglobova prema željenoj poziciji i orijentaciji end-effector-a. Sa ovim se slijedom operacija, u okviru upravljanja manipulatorima, susrećemo puno češće jer je određivanje pozicije i orijentacije vršnog koordinatnog sustava najčešće polazna točka kod upravljanja dijelovima ili alatima.

Kako bi zatim ostvarili pomak industrijske robotske ruke sa početne točke A na krajnju točku B potrebne su nam vrijednosti varijabli svih zglobova kinematskog lanca, u okviru kojih su definirane točke A i B. Predstavljeni proces, suprotan procesu direktne ili unaprijedne kinematike, opisuje problematiku inverzne ili unazadne kinematike koja je u suštini puno kompliciranija od problema direktne kinematike.

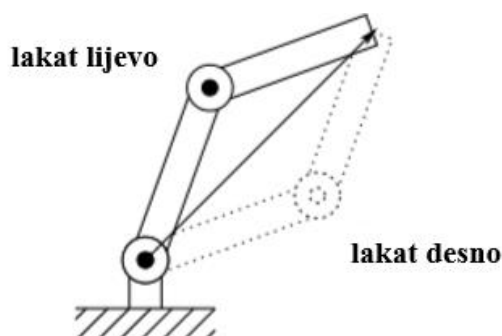
Kao što je prethodno spomenuto, direktna kinematika uvijek daje jednoznačno rješenje do kojega se jednostavno može doći rješavajući jednadžbe direktne kinematike specifičnog manipulatora. U slučaju inverzne kinematike situacija nije jednaka. Za postavljene jednadžbe može, a i ne mora postojati rješenje. Čak i kada rješenje postoji, ono ne mora biti jednoznačno. Nadalje, uzmemo li u obzir da su jednadžbe direktne kinematike nelinearne funkcije varijabli zglobova, do rješenja se ponekad teško dolazi čak i kada ona postoje.

Na slici ispod prikazana je problematika inverzne kinematike na primjeru PUMA manipulatora sa šest stupnjeva slobode gibanja gdje za istu orijentaciju i položaj vršnog koordinatnog sustava postoji osim različitih konfiguracijska rješenja. Na istom je primjeru vidljivo da se položajem end-effector-a varijable zglobova kinematskog lanca manipulatora ne određuju jednoznačno.



**Slika 14** Osam konfiguracijskih rješenja robota sa 6SSG za isti položaj end-effector-a [7]

Za korišteni se SCARA manipulator sa četiri stupnja slobode gibanja problematika inverzne kinematike, u okviru mnogobrojnih rješenja, očituje u dvije varijante konfiguracijskih rješenja. Promatramo li SCARA manipulator okomito na x-y ravninu, tj. gledamo li ga od gore, pozicija end-effector-a na ravnini x-y gotovo će uvijek imati dva rješenja.



**Slika 15** Problem višestrukosti rješenja SCARA manipulatora [1]

Prvo je rješenje lakat lijevo, a drugo lakat desno (u odnosu na ravninu simetrije) pri čemu su oba određena kutevima rotacije zglobova 1 i 2, tj.  $\theta_1^*$  i  $\theta_2^*$ .

U okviru matematički pogodnih rješenja nužno je uzeti u obzir sva mehanička ograničenja manipulatora. Rotacijski zglobovi kinematičkog lanca u praksi gotovo nikada nemaju radni raspon od  $360^\circ$  zbog mehaničkih ograničenja. To znači da se određeni broj matematički ispravnih rješenja ne mogu primijeniti u praksi.

Radni prostor IRB 910SC-3/0.45 SCARA robotske ruke određen je ograničenjima tri rotacijska i jednog translacijskog zgloba, koji su navedeni u tablici ispod.

**Tablica 2 Radni raspon zglobova IRB 910SC-3/0.45 SCARA manipulatora**

Zglob	Tip kretanja	Radni raspon
1	rotacijski	$-140^\circ \dots 140^\circ$
2	rotacijski	$-150^\circ \dots 150^\circ$
3	linearni	$-180 \text{ mm} \dots 0 \text{ mm}$
4	rotacijski	$-400^\circ \dots 400^\circ$

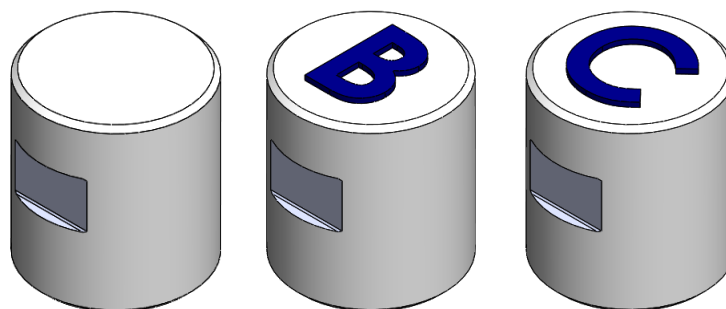
Inverzna nam kinematika, kao prethodno opisani proces, omogućuje upravljanje industrijskim manipulatorima, počevši pri tome od postavljanja potrebnih radnih pozicija. Za navedene radne pozicije, varijable zglobova kinematičkog lanca dobivamo kao rješenja matematičkih jednadžbi. Dobivene vrijednosti zatim dodatno provjeravamo u svrhu utvrđivanja rješenja koja su fizikalno ispravna, uzimajući pri tome fizička ograničenja manipulatora. Kao posljednji korak, u slučaju višestrukih matematički i fizikalno ispravnih rješenja, odabiremo ono koje nam najbolje odgovara za obavljanje potrebne funkcije.

U poglavlju 5 (Upravljanje SCARA robotskom rukom) opisani će koraci biti vidljivi u zadatku diplomskog zadatka.

## 4. RADNO OKRUŽENJE I ALATI

Kao što je prethodno spomenuto u uvodu, SCARA manipulator sa četiri stupnja slobode gibanja koristiti će se za razmještanje velikog broja komponenata sa nedefinirane pozicije A na prethodno definiranu poziciju B prema principu prikazanom na slici 1. Fizičke će komponente pri tome predstavljati slova abecede čijim će se poretkom i organiziranim položajem omogućavati interakcija i komunikacija sa korisnikom.

Na slici ispod prikazan je osnovni oblik fizičke komponente koja se raspoređuje (lijevo), odnosno valjka sa bočnim utorima za lakše hvatanje. Prema njemu su zatim oblikovane sve ostale komponente, tj. sva slova abecede, čiji je primjer vidljiv na prikazanom slovu "B" (sredina) i slovu "C" (desno). Upravljanje svim komponentama obavljat će se odabranom IRB 910SC-3/0.45 SCARA robotskom rukom.

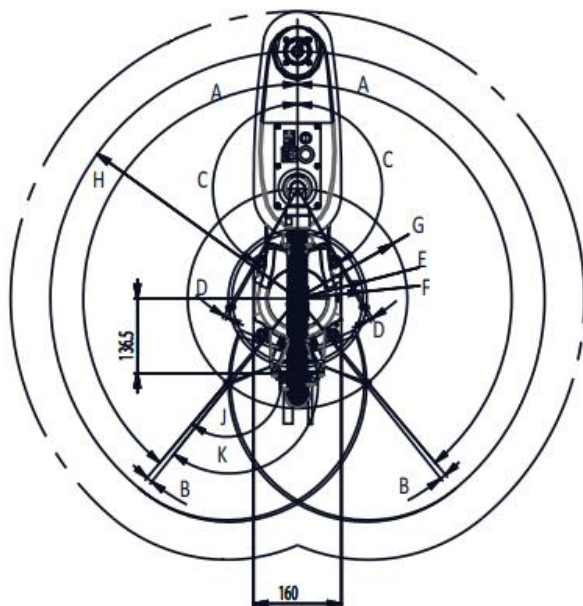


**Slika 16 Osnovni oblik i primjeri premještanih fizičkih komponenata**

Osim odabira samog manipulatora bilo je nužno stvoriti uvjete u kojima bi se omogućilo zadovoljavajuće rukovanje komponentama. Preciznije, bilo je potrebno definirati uvjete radnog okoliša i kreirati optimalno rješenje alata koji bi, zajedno sa robotskom rukom, omogućavao premještanje komponenata na željenu poziciju. Međusobnom interakcijom robota, alata i komponenata sa oznakama slova u okviru radnog okruženja omogućit će se obavljanje zadatka ovog diplomskog rada.

### 4.1. Radno okruženje

Radni prostor IRB 910SC-3/0.45 SCARA robotske ruke, prikazan na slici 17, velikim je dijelom utjecao na oblikovanje elemenata nužnih za pozicioniranje svih komponenata slova. Bilo je nužno osigurati da se sve početne i krajnje točke, pri pozicioniranju, nalaze u doseg SCARA manipulatora, tj. da se nalaze između linija minimalnog i maksimalnog dohvata uzevši u obzir duljine svih segmenata manipulatora te ograničenja svih zglobova.

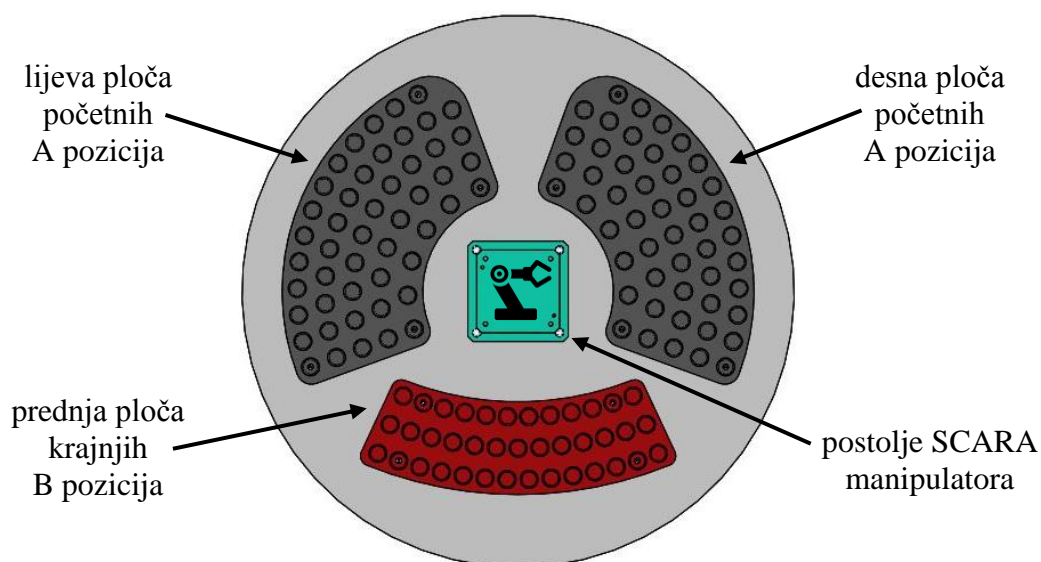


**Tablica 3 Dimenzije radnog prostora IRB 910SC-3/0.45 SCARA manipulatora**

A	140°	F	119 mm
B	1.5°	G	200 mm
C	150°	H	450 mm
D	1.2°	J	150°
E	126 mm	K	151.2°

**Slika 17 Radni prostor IRB 910SC-3/0.45 SCARA manipulatora**

Prema navedenim je ograničenjima definirano osnovno radno okruženje prikazano na slici ispod. Na baznu se ploču vijcima učvršćuju sve potrebne komponente uključujući: postolje SCARA manipulatora kojim se postiže željena visina, desnu i lijevu ploču na kojima su originalno pozicionirane sve komponente, odnosno sva slova koja se zatim programom određenim redoslijedom razmještaju na krajnje pozicije, uključene u okviru prednje ploče.



**Slika 18 Raspored osnovnih komponenata radnog okoliša**

Radi jednostavnosti, tijekom prikazivanja sustava koji je potreban za rješavanje zadatka, uglavnom će se prikazivati samo komponente koje direktno okružuju SCARA robotsku ruku, kao centralni dio ovog diplomskog rada.

Prikazani radni okoliš uključuje ukupno 137 radnih pozicija. Od ukupnog broja, na njih 98 (podijeljeno na desnu i lijevu ploču početnih pozicija) pozicionirane su komponente sa slovima koje služe kao baza za buduće razmještanje. To znači da se u sustavu u svakom trenutku nalazi ukupno 98 raspoloživih komponenata, tj. 98 slova čijim se premještanjem sa pozicije A na točno određenu poziciju B omogućava ispisivanje. Nizanjem određenog broja komponenata točno određenim redoslijedom i rasporedom na poziciju B omogućuje se ispis u tri retka.

Kako bi se osigurao povoljan raspored broja komponenata u okviru raspoloživog prostora i broja pozicija zadržan je minimalan broj različitih slova koji bi još uvijek osiguralo zadovoljenje potreba zadatka. U tablici 4 su prema tome kriteriju prikazana sva slova koja će unutar radnog okoliša SCARA manipulatora biti reprezentirana fizičkim komponentama u obliku prikazanom na slici 16.

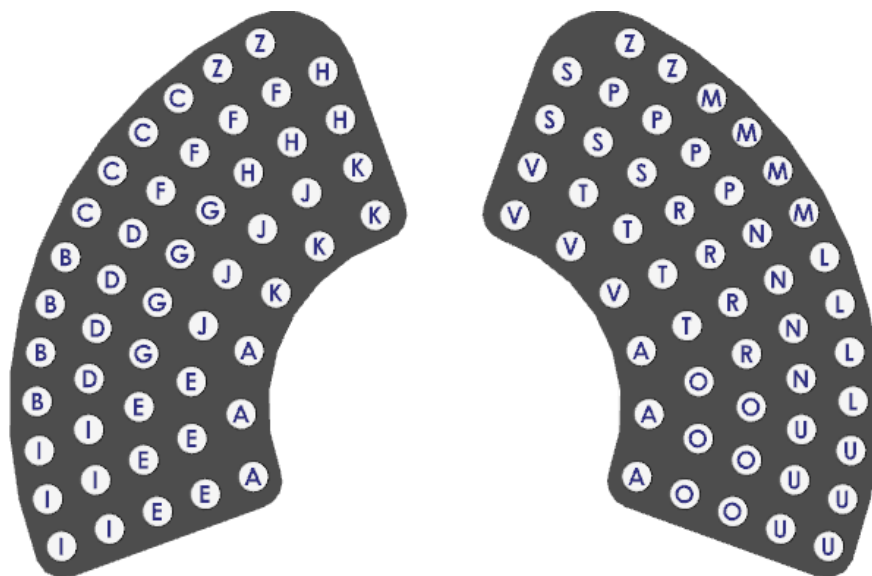
**Tablica 4** Komponente uključene u radni okoliš

<i>Samoglasnici</i>	<i>Suglasnici</i>			
A	B	H	N	V
E	C	J	P	Z
I	D	K	R	
O	F	L	S	
U	G	M	T	
<i>30 komponenata</i>	<i>68 komponenata</i>			

Pri tome je učestalost samoglasnika veća od učestalosti suglasnika. Svaki je samoglasnik predodređen sa šest jednakih komponenata (čineći pri tome ukupno 30 komponenata samoglasnika), a svaki suglasnik sa četiri (čineći pri tome ukupno 68 komponenata suglasnika). Ukupan broj komponenata u sustavu je prema tome, već spomenutih, 98 slova.

Navedene su komponente smještene na lijevu i desnu ploču, grupirajući pri tome komponente istog tipa. Pri završetku rada sa određenom komponentom, ista se uvijek vraća na njenu originalnu poziciju čime osiguravamo da je lokacija određenog slova uvijek poznata u svrhu potrebe ponovnog razmještaja SCARA manipulatorom.





**Slika 19 Raspored komponentata na lijevoj i desnoj ploči**

Uporabom IRB 910SC-3/0.45 SCARA robotske ruke, sa lijeve ili desne ploče točno određene komponente premještamo na programom definiranu poziciju prednje ploče. Svrha procesa je nizanje komponentata, tj. ispisivanje kojim se ostvaruje komunikacija sa korisnikom. Primjer takve primjene prikazan je na slici 20.

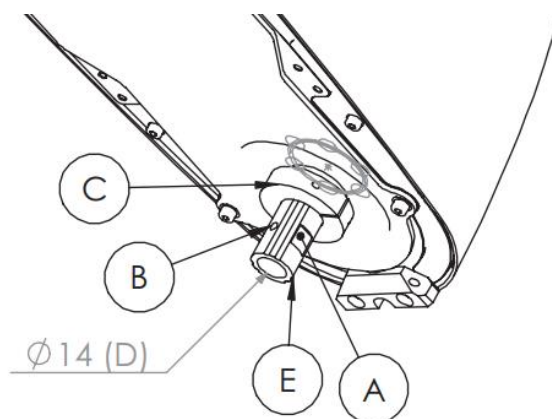


**Slika 20 Primjer ispisa premještanjem komponentata na prednju ploču**

#### 4.2. Alat za razmještanje

Uporaba same robotske ruke bilo kojeg tipa, uključujući SCARA manipulatore, nije dovoljna da bi se omogućilo rukovanje komponentama. Za svaki je proces automatizacije (u obliku različitih primjena) potrebno razviti specijalizirani alat koji se zatim dodaje na vršni segment kinematskog lanca. U kombinaciji sa robotskom rukom on omogućava obavljanje potrebne funkcije. Mjesto prihвата alata za odabrani ABB manipulator prikazan je na slici 21.

U ovom bi slučaju alat omogućio prihvat i premještanje komponentata sa njihovih početnih pozicija na programom određene krajnje pozicije. U tu su svrhu u okviru ovog zadatka razmotrene dvije glavne opcije: uporaba vakuumskih usisnih čašica i uporaba hvataljki.



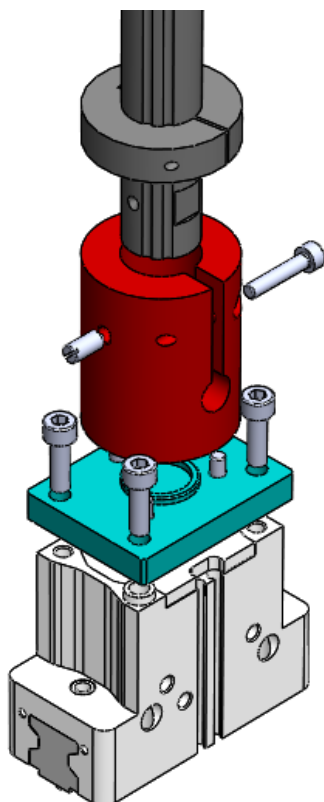
**Slika 21 Mjesto prihвата alata na IRB 910SC-3/0.45 SCARA manipulatoru**

Između prethodno spomenute dvije se opcije vakuumska usisna čašica ističe kao konstrukcijski jednostavnije rješenje. Pneumatski bi se prihvat čašicom osiguravao sa gornje strane valjkaste komponente. Ona bi se zatim pomakom manipulatora premjestila na krajnju poziciju, nakon čega bi uslijedilo jednostavno odvajanje od alata uslijed djelovanja gravitacijske sile (kao rezultat prestanka djelovanja vakuuma). Navedeni proces za sobom povlači jednostavniju izvedbu sa manjim brojem komponenata te manje potrebnog radnog volumena u odnosu na izvedbu alata pomoću hvataljki. To bi omogućilo gušće postavljanje valjkastih komponenata, tj. omogućilo bi uključivanje većeg broja raspoloživih slova.

Funkcionalnost vakuumnog alata gotovo bi potpuno ovisila o tlaku koji se ostvaruje vakuumskom usisnom čašicom. Prema tome bi se moralo osigurati da je minimalno gornja površina valjkastih komponenata povoljno oblikovana i da je izrađena od materijala na koji možemo djelovati podtlakom. No i u slučaju kombinacije idealno oblikovanih komponenata i povoljno izabranih vakumskih čašica glavni nedostatak vakuumnog alata ostaje neriješen. Trenutni nestanak vakuuma u sustavu uzrokovao bi slobodan pad valjkastih komponenata, tj. uzrokovao bi praktički trenutno odvajanje komponente od alata. U takvoj bi se situaciji ugrozio rad cijelog sustava zbog sila koje bi se javile tijekom tranzijenta uspostave atmosferskog tlaka na mjestu djelovanja vakuuma.

Stoga se razmatra alternativno rješenje sa pneumatski aktuiranom hvataljkom.. Kako bi se pozicija komponenata u odnosu na alat osigurala i u situaciji nestanka tlaka u sustavu, za glavnu komponentu alata odabire se NC (normally closed) hvataljka. Uporabom ovog tipa pneumatske hvataljke, koji ima obrnutu logiku rada, sustav se osigurava od odvajanja komponente od alata i uklanja se nedostatak koji se javlja kod vakuumske hvataljke. To je osigurano konstrukcijom NC hvataljke koja u uključenom (1) stanju ima otvorene čeljusti, a u

isključenom (0) stanju zatvorene čeljusti. Prema tome će tijekom premještaja komponente hvataljka uvijek biti isključena i potencijalni nestanak napajanja aktuatora neće moći uzrokovati odvajanje komponente od alata, te će ista uvijek biti osigurana od ispadanja. Tek dolaskom na završnu poziciju hvataljka se uključuje, čeljusti hvataljke se otvaraju i komponenta se odvaja. Isti je proces odvija i kod prihvata komponente gdje se hvataljka uključuje i čeljusti otvaraju prije pozicioniranja oko odabrane komponente, a zatim isključuje i čeljusti se zatvaraju kako bi se ostvario prihvata željenog slova. Za obavljanje navedenog zadatka odabrana je GPP5006NC-00-A Zimmer hvataljka čije čeljusti pri otvaranju ostvaruju maksimalni pomak od 6 mm.

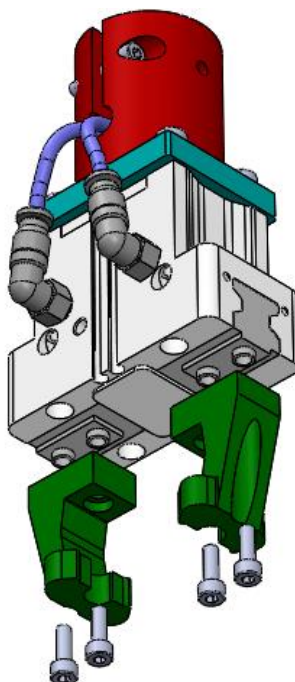


**Slika 22 Exploded view prihvata hvataljke**

Kako bi se omogućila uporaba odabrane Zimmer hvataljke, mora se osigurati njen prihvata na vršni segment SCARA manipulatora, prikazan na slici 21. U tu je svrhu kreirana prirubnica prikazana na slici iznad. Osim prihvata ona osigurava ispravno pozicioniranje i orijentaciju alata u odnosu na vršni segment robotske ruke, što je nužno za pravilno rukovanje komponentama. Ispravna orijentacija alata ostvaruje se korištenjem uvrtnog vijka čiji se konusni završetak naslanja na stranice konusne rupe vršnog segmenta manipulatora označene slovom 'B' na slici 21.

Na pomične su čeljusti hvataljke zatim dodani prsti koji zajedno sa čeljustima ostvaraju linearni pomak od 6 mm u suprotnim smjerovima. Oni su u direktnom kontaktu sa komponentama koje premještamo i njihovim se pomakom ostvaruje prihvat i otpuštanje komponenata, prema principu opisanom ranije.

Stlačeni zrak potreban za pomak prstiju hvataljke dostavlja se kroz pneumatske cijevi te se pomoću pneumatskog priključka dovodi direktno do komora hvataljke. Zbog stalnog je pomaka robotske ruke nužno osigurati da tijekom rada ne dođe prekida dostave zraka, tj. da ne dođe do oštećenja ili zapetljanja pneumatskih cijevi. Iste zbog toga do alata dovodimo kroz središte vršnog segmenta kinematičkog lanca, koji ostvaruje aksijalni pomak. Sa pneumatskim priključcima cijevi povezujemo pomoću izlaza koji je ostvaren u prirubnici alata.

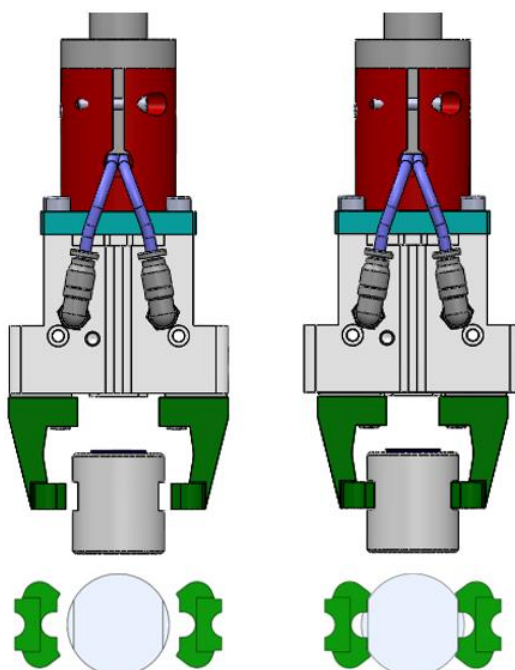


**Slika 23** Alat SCARA manipulatora

Prsti hvataljke i oblik osnovne komponente prikazane na slici 16 Slika 16 oblikovani su ovisno jedan o drugome kako bi se omogućio prijenos odabrane komponente sa pozicije A na poziciju B. Za omogućavanje prijenosa bilo je nužno osigurati da se u isključenom stanju hvataljke ostvaruje prihvat komponente, a da se u uključenom stanju omoguće otpuštanje komponente ili pozicioniranje u odnosu na komponentu čije će se premještane vršiti. Opisana su stanja alata vidljiva na slici 24.

U svrhu osiguranja orijentacije komponenata (na početnim i krajnjim pozicijama) na prstima hvataljke te na osnovnom obliku komponente slova dodajemo ravne površine koje

naliježu jedna na drugu. Na taj se način neutraliziraju minimalne pogreške u orijentaciji koje nastaju kao rezultat greške robotske ruke. U suprotnome, one bi mogle uzrokovati veća odstupanja od željene orijentacije kao rezultat zbroja velikog broja minimalnih pogrešaka ponavljanjem procesa premještanja.



**Slika 24** Alat pri pozicioniranju (lijevo) i prijvatu (desno)

Na alatu se prihvata prema tome ostvaruje i oblikom i silom. A premještanje komponente ostvaruje se zajedničkim djelovanjem alata i pomakom robotske ruke.

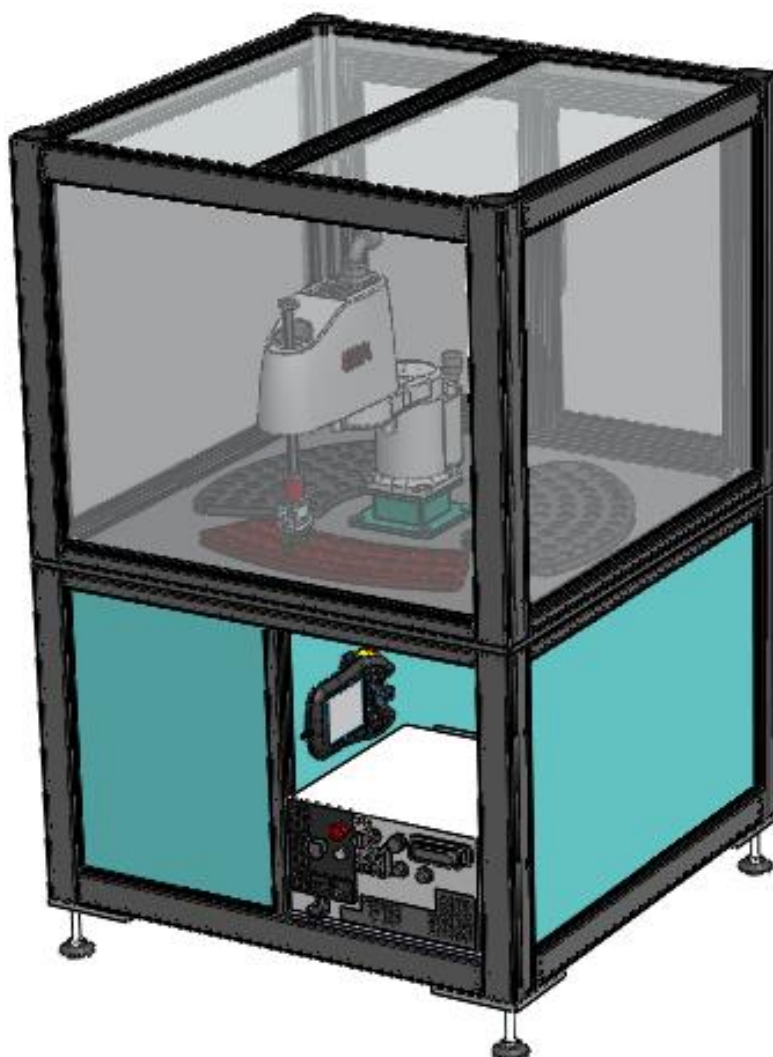
#### 4.3. Robotska stanica

Prethodno navedene komponente će većinskim dijelom, u okviru diplomskog rada, biti prikazane kao zasebna i samostalna jedinica. Prikaz gotovo isključivo tih komponenata odabran je kao optimalan zbog jednostavnosti i lakšeg pregleda pri obavljanju zadanog zadatka. Navedene su komponente (uključujući ABB SCARA robotsku ruku, alat manipulatora, ploče u radnom prostoru i komponente koje se premještaju) pri tome pozicionirane na prethodno spomenutu baznu ploču. Ona je zbog jednostavnosti prikaza na većini priloženih slika vidljiva kao kružna ploča na koju su vijcima učvršćene dodatne komponente.

Spomenuta bazna ploča, u cijelosti, sa svim navedenim dodatnim komponentama čini sastavni dio robotske stanice, kao cjelovitog rješenja priloženog zadatka. Robotska je stanica u cijelosti prikazana na slici 25.

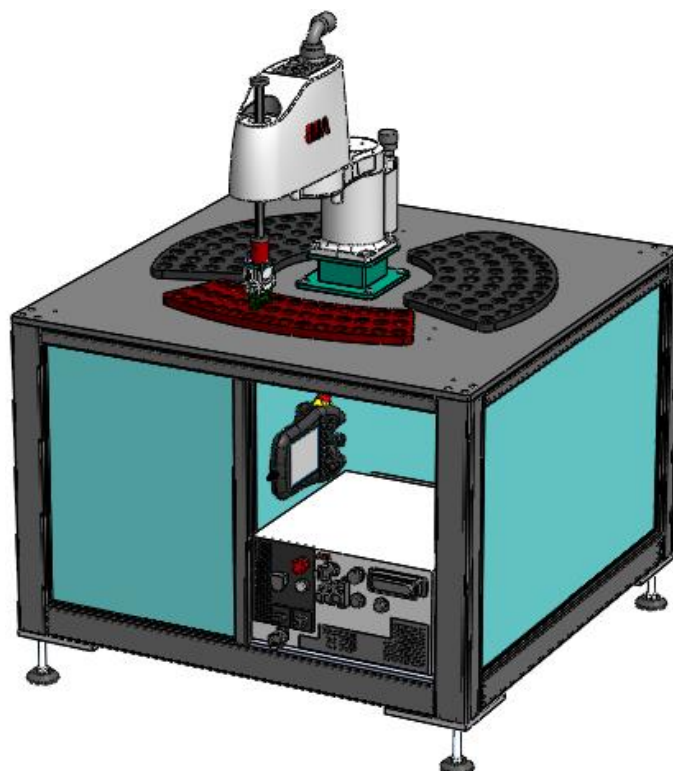
Uz baznu ploču, sastavne dijelove robotske stanice čine donji i gornji okvir, prikazani na slikama 26 i 27. Konstrukcija obuhvaćena gornjim i donjim okvirom radne ćelije osigurava uvjete za ispravan rad manipulatora i obavljanje željenog zadatka.

Donji okvir obuhvaća sve sustave potrebne za rad manipulatora, uključujući vidljivi upravljački uređaj. Osim toga on služi kao konstrukcija koja određuje visinu radne ploče te joj daje stabilnost. Okvir je izrađen korištenjem standardnih profila i komponenata, te polimernih ploča kako bi se, uz vizualno privlačniju strukturu, zaštitili sustavi potrebni za rad odabranog IRB 910SC-3/0.45 SCARA manipulatora.



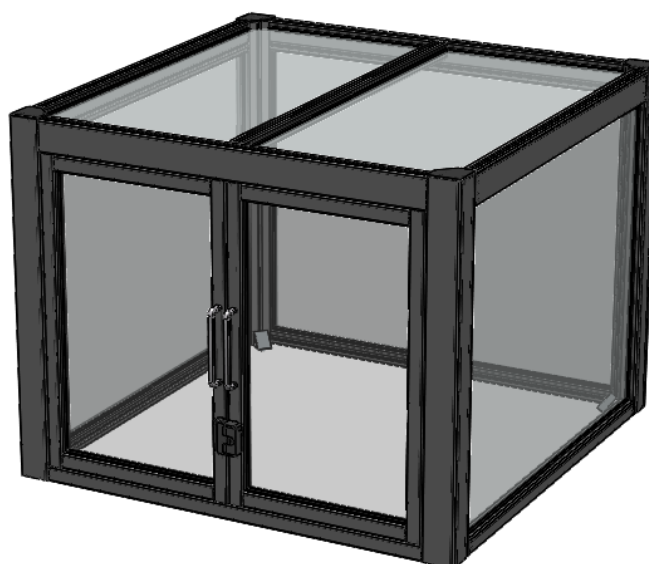
**Slika 25 Robotska stanica za razmještanje**

Gornji je okvir, kao odvojiva komponenta, dodan u sustav zbog povećanja mjera sigurnosti. Jednako kao i donji okvir sastavljen je od standardnih profila i komponenata te prozirnih polimernih ploča kako bi se uz željenu sigurnost osigurala vidljivost radnog procesa.



**Slika 26 Radna ploča na donjem okviru robotske stanice**

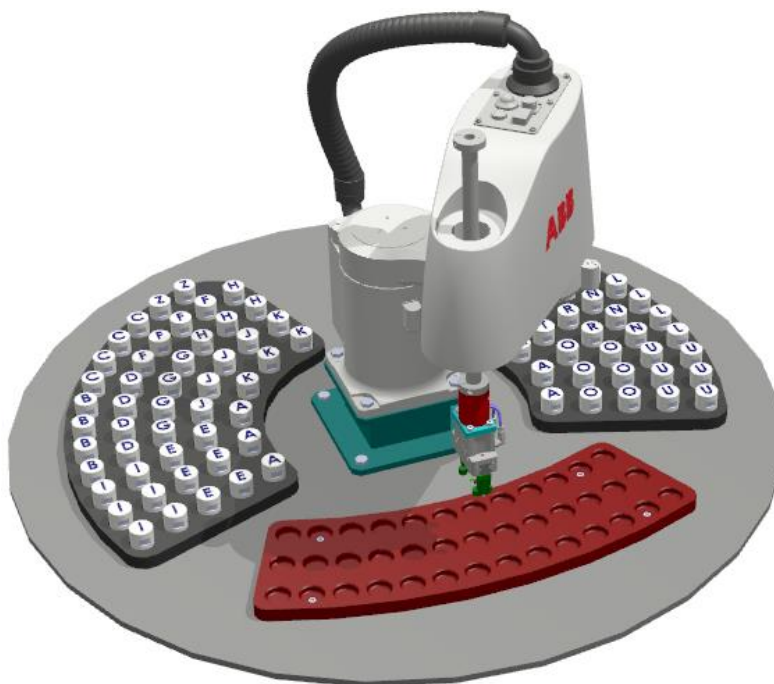
Na stražnjoj se strani gornjeg okvira nalaze vrata koja osiguravaju pristup manipulatoru i radnim komponentama u slučaju potrebe. Vijci kojima se gornji okvir pričvršćuje na radnu ploču nalaze se s unutrašnje strane kako bi odvajanje gornjeg okvira omogućilo samo uz dopušteni pristup manipulatoru.



**Slika 27 Gornji okvir robotske stanice**

## 5. UPRAVLJANJE SCARA ROBOTSKOM RUKOM

Zadavanje uputa za IRB 910SC-3/0.45 SCARA robotsku ruku u okviru virtualnog je okruženja ostvareno korištenjem RAPID programskog jezika. Nizom različitih uputa omogućava se upravljanje manipulatorom u okviru osnovnog radnog okruženja koje je detaljno objašnjeno u prethodnom poglavlju, a u cijelosti je vidljivo na slici ispod.



**Slika 28** Uređenje radnog okruženja diplomskog rada

Navedene nam naredbe omogućavaju premještanje određene komponente sa njene početne A pozicije na krajnju B poziciju gdje ona u ovisnosti o položaju susjednih komponenata čini sastavni dio programom određene cjeline.

Proces premještanja komponente započinje definiranjem radnih pozicija manipulatora u globalnom koordinatnom sustavu. Između određenih se točaka stvaraju procesi gibanja vršnog koordinatnog sustava robotske ruke, tj. procesi gibanja alata sa potencijalno prihvaćenom komponentom. Definiranjem uputa do svih 137 pozicija uključenih u osnovni okoliš SCARA manipulatora omogućuju se sve kombinacije premještanja dostupnih komponenata slova sa lijeve i desne ploče na prednju ploču gdje se ostvaruje interakcija sa korisnikom.

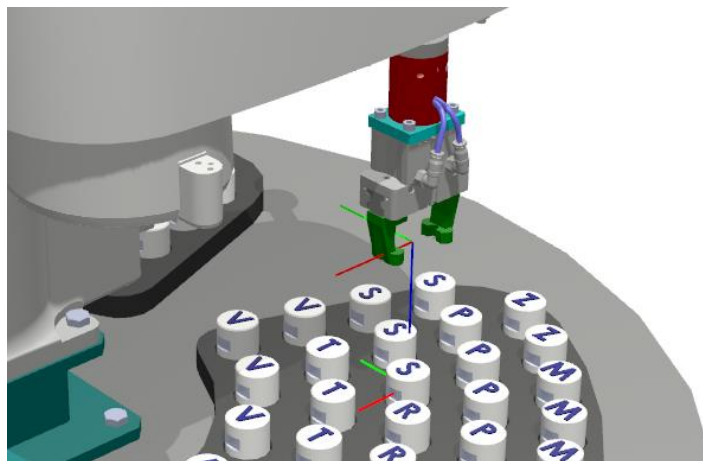
U okviru ovog će poglavlja principi upravljanja biti prikazani na primjeru ispisa riječi "SCARA". Prema istom je procesu omogućeno stvaranje svih kombinacija ispisa. Pri tome je nužno uzeti u obzir broj dostupnih pozicija i dostupnih komponenata koje definiraju slova.



### 5.1. Radne pozicije manipulatora

Određivanje radnih pozicija manipulatora početna je točka pri omogućavanju procesa upravljanja, tj. procesa premještanja komponenata. Osnovna je pozicija pri tome određena položajem alata koji nužno omogućava prihvata željene komponente (u prikazanom slučaju komponente koja predstavlja slovo "S"). Kako bi svaka od pozicija bila jednoznačno određena osim položaja nužno je odrediti i orijentaciju koordinatnog sustava u toj točki kako bi se prihvata komponente ostvario iz željenog smjera.

Uzmemo li u obzir da je pri korištenju SCARA manipulatora onemogućena rotacija oko x-osi i y-osi znamo da će os z na svim radnim pozicijama biti orijentirana u istom smjeru, tj. prema dolje. Kako bi se proces upravljanja dodatno pojednostavio koordinatni su sustavi, tj. sve su komponente potpuno identično orijentirane. Os x je prema tome na svim pozicijama orijentirana prema naprijed (prema prednjoj ploči)



Slika 29 Koordinatni sustavi R\_R3\_P8 radne pozicije

Osim osnovnog je koordinatnog sustava (koji se u globalnom koordinatnom sustavu nalazi na visini od 30 mm) za svaku radnu poziciju dodan dodatni koordinatni sustav (na visini od 150 mm globalnog koordinatnog sustava). Pri pomicanju manipulatora prolazak kroz dodatni koordinatni sustav prethodi prolasku kroz osnovni koordinatni sustav kako bi se pri pomaku osigurali od zapinjanja alatom za ostale komponente.

Unutar virtualnog se okruženja pozicije lako definiraju u odnosu na globalni koordinatni sustav kako bi se omogućio prihvata potrebnih komponenata neovisno o položaju robotske ruke u odnosu na isti. Pri sinkronizaciji koordinatnih sustava na virtualni upravljački uređaj položaj svih lokalnih koordinatnih sustava jednoznačno je određen u odnosu na ishodišni koordinatni sustav manipulatora, tj. u odnosu na  $o_0x_0y_0z_0$  koordinatni sustav. Na taj se način omogućuje

jednostavan prijenos koda sa virtualnog na stvarni upravljački uređaj koji zatim definira radne točne stvarnog manipulatora u odnosu na pripadajući  $o_0x_0y_0z_0$  koordinatni sustav. Na isječku koda ispod definirani su koordinatni sustavi gornjeg i donjeg položaja radne pozicije R\_R3\_P8 u odnosu na bazni koordinatni sustav IRB 910SC-3/0.45 SCARA manipulatora.

#### Isječak koda 1 Prijevod položaja koorfinatnih sustava u odnosu na manipulator

```

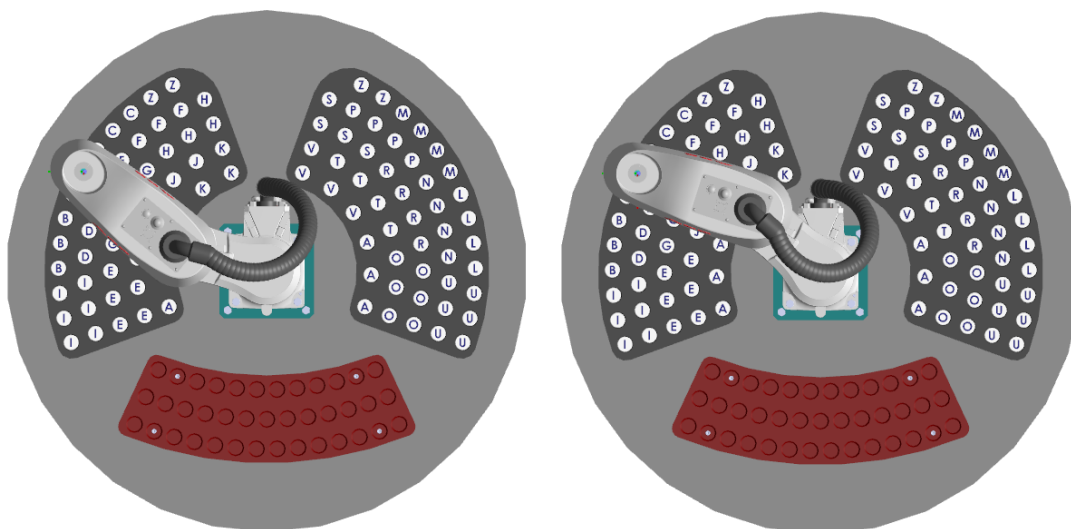
1  CONST robtarget R_R3_P8_high:=[[-252.795,212.12,70],
    [0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
2  CONST robtarget R_R3_P8_low:=[[-252.795,212.12,-50],
    [0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

```

Na prikazani se način određuju položaji svih koordinatnih sustava potrebnih za proces premještanja komponente. Za jednoznačno je određivanje odabrane radne pozicije osim položaja vršnog koordinatnog sustava potrebno odrediti i konfiguraciju robota. Prema prethodno opisanom procesu u poglavlju o inverznoj kinematici znamo da položaj robotske ruke nije jednoznačno određen poznavanjem koordinata krajnje točke ruke robota (alata, odnosno end-effector-a) već se za određene pozicije javlja nekoliko mogućih konfiguracijskih rješenja među kojima je potrebno odabrati najpovoljnije.

#### 5.1.1. Određivanje konfiguracija radnih pozicija

Za SCARA se robotsku ruku, u slučaju nekoliko potencijalnih konfiguracijskih rješenja, uvijek javljaju dva koja su fizikalno zadovoljavajuća. Na slici ispod ova su rješenja vidljiva na primjeru koordinatnih sustava L\_R1\_P8 radne točke, tj. pri prihvatu komponente slova "C".



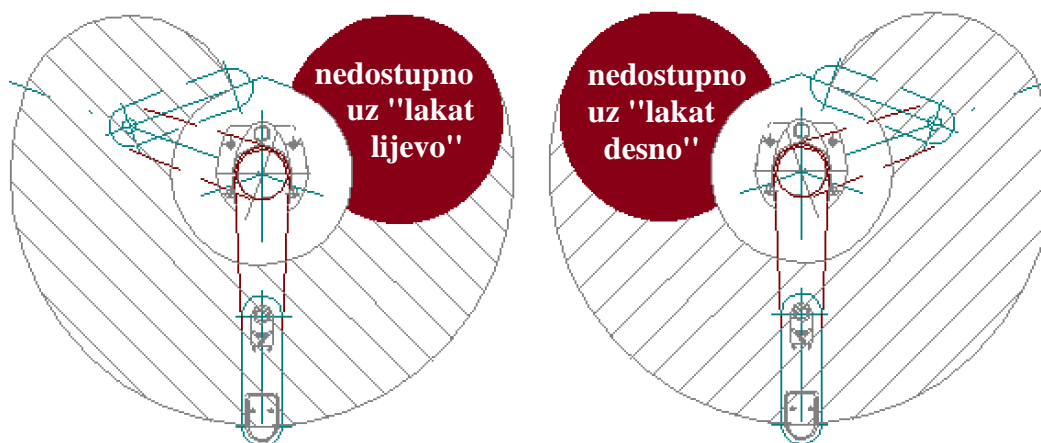
Slika 30 Dva konfiguracijska rješenja L\_R1\_P8 radne točke

Dvije se prikazane konfiguracije za točku L\_R1\_P8\_low, tj. za točku prihvata slova "C" očituju u prikazanim vrijednostima (ispod) zglobova kinematičkog lanca.

**Tablica 5** Vrijednosti konfiguracijskih rješenja L\_R1\_P8 radne točke

	$(-2,-2,0,1)$	$(-2,-2,0,0)$
J1	-104,91	-132,01
J2	-24,36	24,36
J3	-30,70	-30,70
J4	-129,27	-107,66

Za sve se ostale radne pozicije kao rješenje javlja isključivo jedno fizikalno zadovoljavajuće rješenje. Ovisno o lokaciji radne pozicije ono se postiže konfiguracijom "lakat lijevo" za većinu radnih pozicija koje se nalaze na lijevoj ploči, a konfiguracijom "lakat desno" za većinu radnih pozicija na desnoj ploči. Položaj lakta je pri tome određen u odnosu na simetralu koja povezuje bazni i vršni koordinatni sustav. Zbog mehaničkih je ograničenja manipulatora (spomenutih u tablici 2) veliki broj radnih pozicija nedostupan u jednoj od spomenutih konfiguracija. Prikaz opisanih rješenja vidljiv je na slici 31.



**Slika 31** Konfiguracijska rješenja: lakat lijevo i lakat desno

Prema opisanim je zahtjevima jasno da ne postoji jedinstveno konfiguracijsko rješenje zadatka, tj. ni konfiguracijom lakat lijevo ni lakat desno nije moguće dohvatiti sve radne pozicije. Preporuku da se točku singularnosti pri upravljanju manipulatorom izbjegava je prema tome u okviru ovog diplomskog zadatka nemoguće slijediti. Točke singularnosti se za SCARA manipulator nalaze na rubnoj liniji radnog prostora, tj. kada se segmenti koji tvore lakat manipulatora nalaze na istom pravcu.

Prolazak kroz jednu od točaka singularnosti će se prema tome morati ostvariti pri svakom prelasku iz konfiguracijskog rješenja "lakat lijevo" u "lakat desno" i obrnuto. Liniju promjene konfiguracijskog rješenja stoga povlačimo između prednje i lijeve ploče, tj. sve će se radne pozicije na desnoj i prednjoj ploči ostvarivati uz konfiguracijsko rješenje "lakat desno", a sve radne pozicije na lijevoj ploči uz "lakat lijevo".

## 5.2. Premještanje komponente

Kako bi se omogućilo premještanje komponente sa početne točke A na krajnju točku B potrebno je kreirati točne upute kretanja manipulatora između definiranih koordinatnih sustava. Grupiranjem željenog niza uputa omogućuje se stvaranje procesa koji ostvaruju određenu funkciju, kao što je npr. prihvat komponente koja se nalazi na poziciji L\_R5\_P1, tj. prihvat komponente slova "A".

### Isječak koda 2 Naredbe za prihvat komponente (na radnoj poziciji L\_R5\_P1)

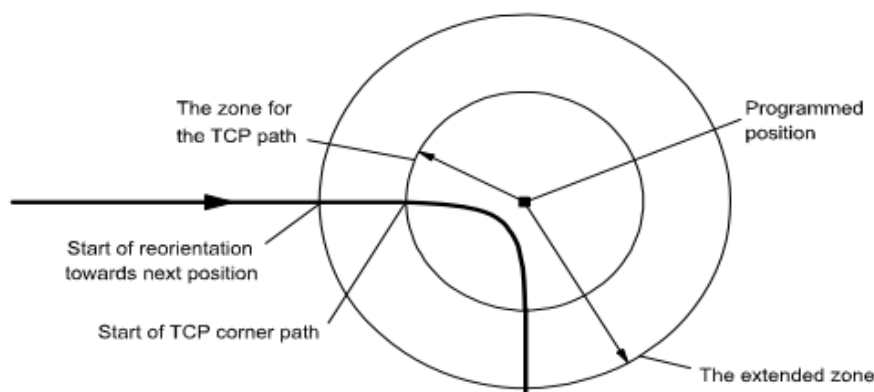
```
1  PROC F_L_R5_P1_Path()
2      MoveJ L_R5_P1_high,v1000,z100,My_Mechanism_1\WObj:=wobj0;
3      Set NCgripper;
4      MoveL L_R5_P1_low,v1000,z100,My_Mechanism_1\WObj:=wobj0;
5      WaitTime 1;
6      Reset NCgripper;
7      MoveL L_R5_P1_high,v1000,z100,My_Mechanism_1\WObj:=wobj0;
8  ENDPROC
```

U isječku koda iznad vidljivi su koraci koje je potrebno ostvariti u svrhu obavljanja te funkcije. Alat manipulatora prvenstveno se pomiče u poziciju direktno iznad točke prihvata. U toj se poziciji naredbom "Set NCgripper" hvataljka uključuje, tj. prsti hvataljke se razmiču kako bi se omogućio prihvat komponente. Pomakom translacijskog zgloba alat se zatim spušta u donji koordinatni sustav, tj u točku L\_R5\_P1\_low. Naredbom "Reset NCgripper" isključuje se hvataljka i prsti se zatvaraju, tj. ostvaruje se prihvat komponente. Ponovnim pomakom alata u točku L\_R5\_P1\_high prihvaćena se komponenta vertikalno podiže. Tim se pomakom završava proces prihvata komponente na poziciji L\_R5\_P1.

Prilikom definiranja načina pomaka manipulatora u okviru priloženog koda razlikujemo dva tipa naredbi: "MoveJ" i "MoveL". "MoveL" koristimo kako bi end-effector, tj. alat manipulatora prema željenoj lokaciji pomaknuli linearno. [8] Ovaj tip naredbe prema tome koristimo kada određenu komponentu spuštamo ili podižemo (osiguranje od zapinjanja).

"MoveJ" s druge strane koristimo kada manipulator želimo pomaknuti s jedne točke na drugu, a da pri tome pomak ne mora biti u ravnoj liniji. Robotska se ruka i njene osi prema destinaciji tada pomiču nelinearnim putem, a sve osi na destinaciju dolaze u istom trenutku. [8] Ovaj je tip kretanja stoga odabran kao povoljniji od linearnog za pomak s jedne radne pozicije na drugu. Osim toga, pri prijenosu određene komponente sa radnih pozicija bočnih ploča na prednju i obrnuto ovaj se tip kretanja očituje u lučnoj putanji što je pogodnije za oblik radnog prostora SCARA manipulatora.

Osim tipa putanje pri zadavanju pomaka manipulatora nužno je definirati brzinu i zonu odstupanja (zonedata) end-effector-a. Na prikazanim je isječcima za "MoveJ" i "MoveL" zadana brzina od 1000 mm/s za vršni koordinatni sustav, koja se smatra brzinom srednjeg iznosa. Definiranjem zone dodatno utječemo na putanju gibanja vršnog koordinatnog sustava određivanjem točne vrijednosti dopuštenih odstupanja od zadanih koordinatnih sustava. Uporabom takvih "fly-by" točaka na kojima se ne obavlja specifična radnja, ali kroz koje se mora osigurati prolaz da bi se ostvarila određena funkcija, omogućuje se postizanje glađe putanje gibanja. Takve su sve gornje ("\_high") točke radnih pozicija.



**Slika 32 Put pomaka vršnog koordinatnog sustava u odnosu na zadanu točku [8]**

Istim se principom nizanja uputa koristimo i pri pozicioniranju komponente na odabroj krajnjoj poziciji. Za prihvaćenu je komponentu "A" to odabrana radna pozicija F\_R2\_P7.

### Isječak koda 3 Naredbe za premještanje komponente (na radnu poziciju F\_R2\_P7)

```

1  PROC F_F_R2_P7_Path()
2      MoveJ F_R2_P7_high,v1000,z100,My_Mechanism_1\WObj:=wobj0;
3      MoveL F_R2_P7_low,v1000,z100,My_Mechanism_1\WObj:=wobj0;
4      WaitTime 1;
5      Set NCgripper;
6      MoveL F_R2_P7_high,v1000,z100,My_Mechanism_1\WObj:=wobj0;
7  ENDPROC

```

Naredbom "MoveJ" ostvaruje se pomak alata sa komponentom iznad finalne pozicije. Linearnim se pomakom komponenta spušta na točku F\_R2\_P7\_low gdje se razmicanjem prstiju hvatalje komponenta "A" otpušta. Nakon pozicioniranja komponente na njenu finalnu poziciju, linearnim se pomakom segmenta robotske ruke, alat ponovno vraća u poziciju iznad.

Završetkom dva prikazana procesa obavljeno je premještanje odabrane komponente na željenu finalnu poziciju. Prijenos sljedećih komponenata u nizu obavlja se na isti način.

### 5.3. Ispis premještanjem niza komponenata

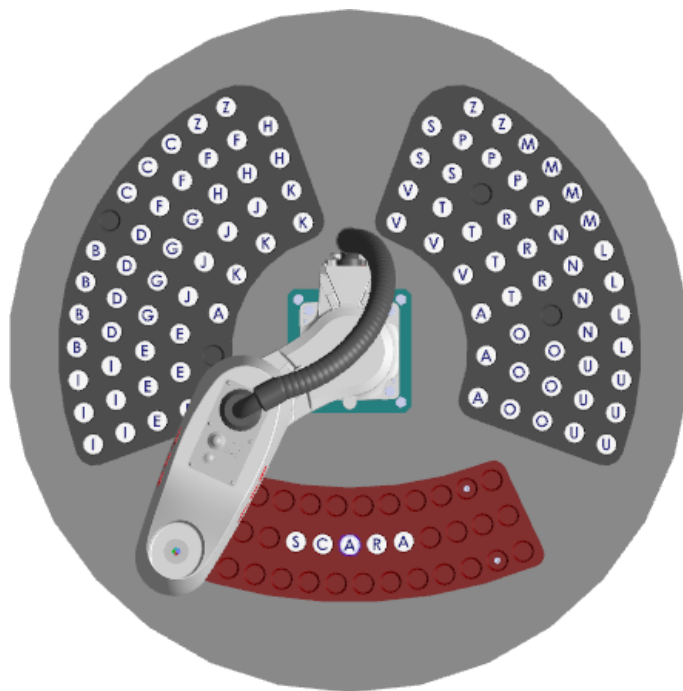
Nizanjem željenih procesa premještanja komponenata omogućuje se ispisivanje. Za odabrani je primjer ispisa riječi "SCARA" bilo potrebno ostvariti premještanje pet različitih komponenata, tj. četiri različita slova. Proces ispisivanja osvaruje se kreiranjem procesa (isječak koda 4 - linija 5) unutar kojega se pozivamo na prethodno opisane potprocese koji omogućavaju prihvrat komponente na njenoj početnoj poziciji i premještanje komponente na njenu finalnu poziciju (isječak koda 4 – linija 6 do 15).

Premještanje slova "S" prema tome ostvarujemo prihvatom komponente na poziciji R\_R3\_P8 i njenim premještanjem na odabranu finalnu poziciju F\_R2\_P5, a slova "C" premještanjem komponente sa L\_R1\_P8 na sljedeću u nizu finalnu poziciju F\_R2\_P6. Detaljni koraci premještanja sljedeće komponente u nizu (slovo "A") sa početne pozicije na odabranu finalnu poziciju prikazani su u okviru isječka koda 1 i isječka koda 2.

#### Isječak koda 4 Proces ispisa riječi SCARA nizanjem procesa premještanja komponenata

```
1   PROC main()
2       Ispis_SCARA;
3   ENDPROC
4
5   PROC Ispis_SCARA()
6       F_R_R3_P8_Path;
7       F_F_R2_P5_Path;
8       F_L_R1_P8_Path;
9       F_F_R2_P6_Path;
10      F_L_R5_P1_Path;
11      F_F_R2_P7_Path;
12      F_R_R3_P4_Path;
13      F_F_R2_P8_Path;
14      F_L_R5_P2_Path;
15      F_F_R2_P9_Path;
16  ENDPROC
```

Kako bi se zatim među brojem postojećih procesa omogućio odabir onih koje želimo iskoristiti u trenutnoj primjeni manipulatora (u okviru simulacije ili na proizvodnoj liniji) isti moraju biti definirani unutar glavnog (main) procesa. Na liniji 2 isječka koda 4 se stoga pozivamo na "Ispis\_SCARA". Rezultat koda prikazanog u isječku vidljiv je na slici 33.



**Slika 33 Položaj komponenata radnog okoliša kao rezultat ispisa ("SCARA")**

Kako bi se omogućio novi ispis nužno je ukloniti prethodno korištene komponente sa prednje ploče. Kao što je prethodno spomenuto, korištene se komponente uvijek vraćaju na njihovu originalnu početnu poziciju. Na taj je način lokacija svake komponente uvijek jednoznačno određena i omogućeno je njeno ponovno premještanje sa iste lokacije.

## 6. INTERAKTIVNOST KROZ IGRU VJEŠALA

Kako bi se unutar postojećeg okruženja SCARA manipulatora, uz određene nadogradnje, omogućila komunikacija između korisnika i robotske ruke kreirana je interaktivna igra unutar koje je sudjelovanje korisnika, tj. njegov unos, nužan za definiranje rada manipulatora. Za interaktivnu je igru odabrana društvena igra "Vješala". Ona obično uključuje olovku i papir, te minimalno dva igrača od kojih će jedan, u okviru ovog diplomskog rada, biti zamijenjen IRB 910SC-3/0.45 SCARA manipulatorom.



Slika 34 Primjer igre "Vješala"

Zadatak prvog igrača je pri tome da osmisli riječ ili rečenicu koju na praznom papiru predstavlja određenim brojem crta, jednak broju znakova u odabranoj riječi ili rečenici. Zadatak drugog igrača je zatim da u određenom broju pokušaja pogodi koja je to riječ, na način da u svakoj rundi ponudi jedno slovo koje se potencijalno nalazi u riječi na papiru. U slučaju da je igrač 2 ispravno pogodio i da se slovo nalazi u riječi, to se slovo ispisuje na pripadajuću crtu. U suprotnome se, na prethodno nacrtano vješalo dodaju dijelovi obješenjaka (engl. Stickman) koji označavaju broj pogrešnih pokušaja igrača. Igra je završena ako igrač 2 dosegne dopušteni broj pogrešaka (ako je Stickman na vješalu dovršen i samim time obješen) ili ako igrač 2 uspije pogoditi riječ osmišljenu od strane igrača 1 prije nego dosegne dopušteni broj pogrešaka. U prvom je slučaju pobijedio igrač 1, a u drugom igrač 2. Primjer igre prikazan je na slici 34.

Igrač 1 i njegovi zadaci u igri "Vješala" biti će zamijenjeni sustavom odabranog manipulatora. Na taj će se način omogućiti stvaranje interaktivnog okoliša u kojemu će minimalan broj sudionika uključivati robotsku ruku unutar čijeg upravljačkog sustava se "osmišljava" riječ koju zatim igrač 2 u određenom broju pokušaja pogađa.

Kako bi se prethodno opisani sustav realizirao bilo je potrebno osigurati nekoliko osnovnih stavki:

- komponente u radnom prostoru manipulatora koje bi predstavljale slova, tj. crte u klasičnoj verziji igre prije nego igrač počne sa pogađanjem,



- sustav komponenata koji bi omogućavao brojanje netočnih pokušaja pogađanja igrača,
- sustav koji bi omogućavao komunikaciju između manipulatora i korisnika, tj. koji bi korisniku omogućavao odabir određenog slova,
- kod koji bi upravljao robotskom rukom i na taj način odrađivao zadatke koji inače pripadaju igraču 1.

U narednim će potpoglavljima detaljno biti obrađene ove stavke.

## 6.1. RADNO OKRUŽENJE

Igra "Vješala" koja je odabrana za stvaranje interaktivnog okoliša u okviru ovog diplomskog rada samo je jedan od primjera mogućih proširenja, tj. nadogradnji na osnovni sustav koji je objašnjen u poglavlju 4. Zadržavanjem manipulatora u centru sustava te dodavanjem dodatnih ploča sa komponentama ili zamjenom postojećih komponenata slova drugim komponentama moguće je proširiti i varirati osnovni koncept premještanja komponenata u svrhu nekog tipa komunikacije sa korisnikom, tj. igračem.

Kao što je prethodno spomenuto, proširenja koja su potrebna kako bi se omogućilo stvaranje interaktivnog okoliša u obliku igre "Vješala" su komponente koje će predstavljati slova za tvorbu riječi prije pokušaja pogađanja igrača, te sustav komponenata koji će voditi računa o broju neispravnih pokušaja. Kako bi sustav u procesu prilagodbe ostao što slučajniji originalnom sustavu, te kako bi se za prihvrat mogao koristiti isti alat, nove će uvedene komponente imati isti osnovni oblik kao i prethodno korištene komponente slova, prikazane na slici 16.

Umjesto crta klasične igre koje predstavljaju slova odabrane riječi (definirajući pri tome njenu duljinu) koristiti će se komponente bez slova, tj. sam osnovni oblik komponente bez ikakve oznake na gornjoj plohi. On je na slici 16 prikazan potpuno lijevo i imati će istu funkciju kao i prazne crte u klasičnoj igri "Vješala". Pomakom određenog broja komponenata bez oznaka na točno definirane pozicije prednje ploče definirat će se duljina odabrane riječi za pogađanje.

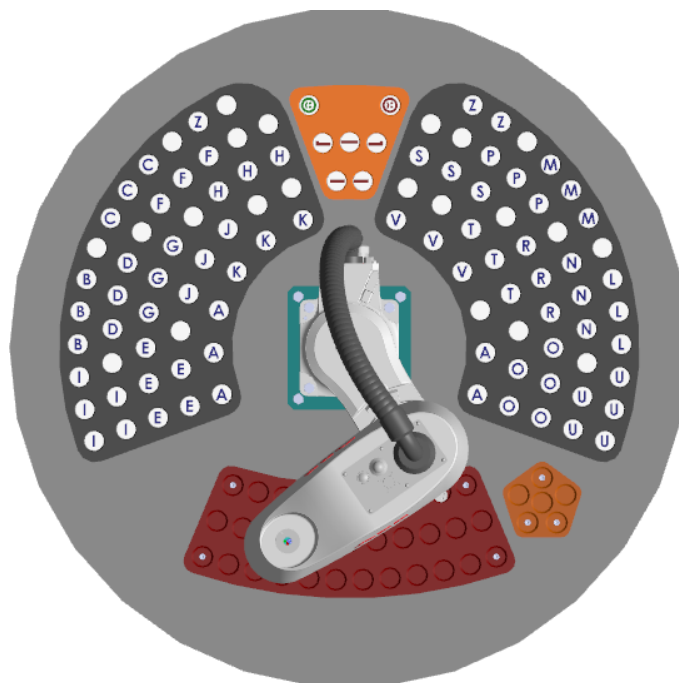
Isti princip oblikovanja koristit će se i za sustav komponenata koje omogućavaju brojanje neispravnih pokušaja pogađanja igrača, tj. osnovni će oblik komponente biti isti. Kako bi se u okviru automatizirane igre "Vješala" zadržali što bliže glavnim obilježjima klasične igre, svaki će neispravan pokušaj pogađanja biti praćen dodavanjem jednog od dijelova tijela Stickman-a na vješalo, tj. na definiranu poziciju. Tijelo Stickman-a kreirano je od šest komponenata dodavanjem obilježja na čelnu plohu valjkastih oblika. Njihovim se zavisnim

pozicioniranjem na dodatnu ploču u obliku pravilnog peterokuta tvori ukupna slika Stickman-a (vidljiva na slici 35). Dovořavanje Stickman-a na prikazanoj ploči označava kraj igre i pobjedu za IRB 910SC-3/0.45 SCARA manipulator. Dopušteni broj pogreřaka u pogađanju je (prema broju komponenata koje čine tijelo Stickman-a) ukupno šest.



**Slika 35 Sustav komponenata za brojanje neispravnih pokušaja pogađanja**

Dodatne komponente igre "Vješala" uvodimo u postojeći radni okoliř SCARA manipulatora predstavljen u prethodnom poglavlju. Pored prednje ploče dodajemo ploču finalnih Stickman pozicija gdje ostvarujemo prijenos jedne od šest komponenata prikazanih na slici iznad. Kako bi se dodana ploča pozicionirala na navedeno mjesto bilo je potrebni skratiti postojeću prednju ploču kojoj su u svakom redu oduzete dvije radne pozicije.



**Slika 36 Raspored komponenata radnog okoliřa za igru "Vješala"**

Osim ploče finalnih pozicija u sustav je dodana i ploča početnih Stickman pozicija. Na njoj su dijelovi tijela neovisno pozicionirani, a smještena je između lijeve i desne ploče. Kao i

navedene ploče, ploča početnih Stickman pozicija služi kao bazna ploča. Prijenos sa stražnje na jednu od prednjih ploča ostvaruje se djelovanjem igrača, tj. odabirom slova koje se ne nalazi u odabranoj riječi. Na ploču početnih Stickman pozicija je osim dijelova tijela dodana dodatna komponenta koja označava pobjedu za igrača u slučaju pogođene riječi. Ukupni broj dodanih radnih pozicija je prema tome 13, od čega 6 pripada prednjoj Stickman ploči, a 7 stražnjoj.

Dodane se ploče pozicioniraju na isti način kao i osnovne ploče radnog okoliša. Na određenim se radnim pozicijama nalaze prolazne rupe koje omogućavaju učvršćivanje za baznu ploču. Za skraćenu se prednju ploču koriste iste prihvatne točke kao i za originalnu.

Komponente bez oznaka koje dodajemo u radni okoliš SCARA manipulatora (kako bi definirali duljinu riječi koju igrač pogađa) postavljamo na postojeće radne pozicije lijeve i desne ploče. Pri tome se uklanja po jedna komponenta svakog raspoloživog slova u sustavu kako bi se osigurao prostor za komponente bez oznaka. Nova raspodjela komponenata, uključujući one bez oznaka, vidljiva je na slici 36. Komponente bez oznaka se u početku nalaze na početnim pozicijama lijeve i desne ploče, tj. nalaze se u bazi dostupnih komponenata, a zatim se po potrebi i prema upravljačkom kodu premještaju na prednju ploču. Pritom se prazne komponente pozicioniraju tako da se njih 5 nalazi među samoglasnicima, a 17 među suglasnicima, kako je prikazano u tablici ispod.

**Tablica 6 Komponente lijeve i desne ploče namijenjene za igru "Vješala"**

<i>Samoglasnici</i>	<i>Suglasnici</i>				<i>Komponente bez oznaka</i>
A	B	H	N	V	–
E	C	J	P	Z	
I	D	K	R		
O	F	L	S		
U	G	M	T		
$30 - 5 =$ <i>25 komponenata</i>	$68 - 17 =$ <i>51 komponenta</i>				<i>22 komponente</i>

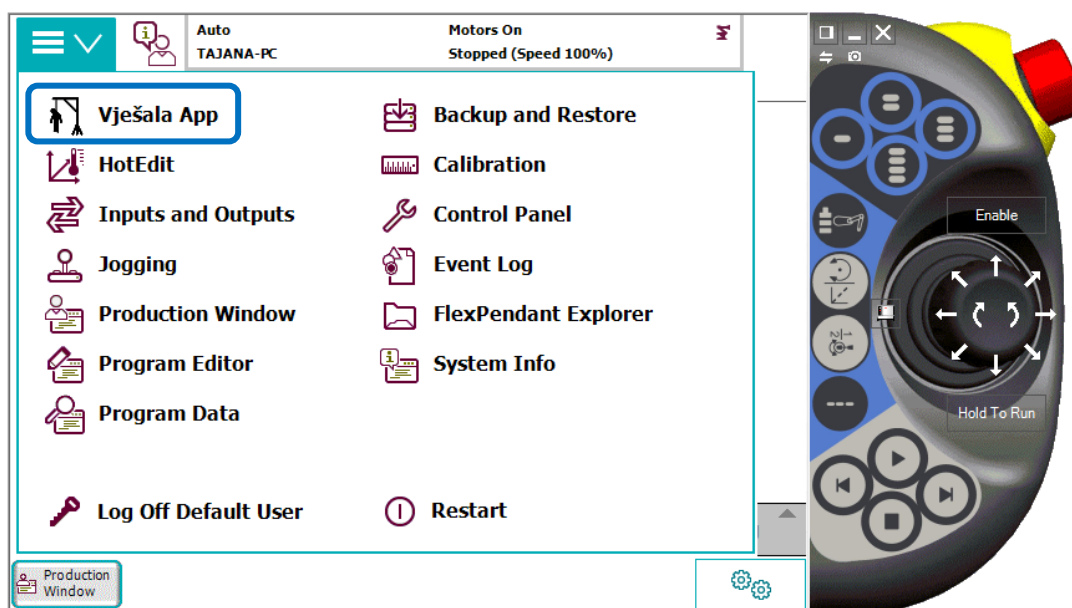
Uklanjanjem jedne komponente svakog slova smanjuje se broj ostvarivih mogućnosti ispisa riječi, no proces je bio nužan kako bi se osnovni okoliš manipulatora minimalno mijenjao te kako bi se sve komponente nalazile u doseg SCARA robotske ruke. Ukupni je broj komponenata na lijevoj i desnoj ploči na taj način ostao isti, zadržavajući pri tome položaj zadržanih komponenata sa slovima. Sve su komponente bez oznaka pri tome pozicionirane na

četvrtu u redu poziciju komponente koju zamjenjuju. Osiguravanjem ukupno 22 radne pozicije za komponente bez oznaka određen je maksimalan broj znakova riječi koja se pogađa.

## 6.2. SUČELJE

U svrhu stvaranja interaktivnog je okoliša bilo nužno osigurati da oba sudionika, tj. i manipulator i igrač mogu obavljati svoju funkciju. Zadatak igrača 2 u igri "Vješala" je pogađanje riječi osmišljene od strane drugog sudionika igre, tj. odabir slova koje se potencijalno nalazi u riječi definiranoj određenim brojem praznih crta ili komponenata bez oznaka.

Osim same potrebe da se stvori način na koji bi igrač odabrao određeno slovo, bilo je potrebno osigurati vezu između odabira igrača i sustava manipulatora kako bi se omogućila međusobna interaktivnost. Među različitim je opcijama kao najjednostavnije rješenje odabrano korištenje ručnog programatora (engl. Teaching Pendant), tj. modula za "učenje" robota.



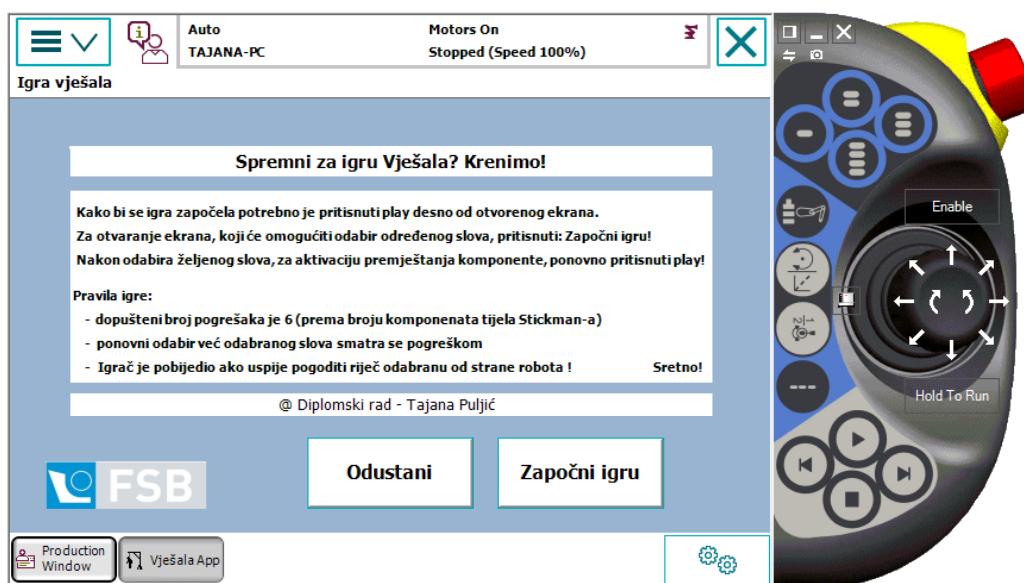
Slika 37 Prikaz aplikacije za igru vješala u glavnom izborniku

Kao uređaj koji se uvijek nalazi na mjestu instalacije manipulatora (u svrhu upravljanja manipulatorom i kreiranja programa) omogućio je stvaranje interaktivnog sučelja za igru "Vješala" izbjegavajući pri tome potrebu da se u radni okoliš manipulatora dodaju dodatne komponente potrebne za obavljanje zadatka. Nadalje, korištenjem uređaja koji već čini sastavni dio sustava manipulatora i povezan je sa upravljačkim uređajem omogućilo se korištenje postojeće veze između upravljačkog uređaja i Teaching Pendant-a, odnosno odabira određenog slova igrača.

Sučelje za igru "Vješala" kreirano je kao aplikacija vidljiva u glavnom izborniku virtualnog Teaching Pendant-a (slika 37). Prethodno opisani princip prijenosa upravljačkog koda sa virtualnog upravljačkog uređaja na stvarni primjenjuje se i za prijenos stvorenog sučelja na Teaching Pendant na mjestu instalacije manipulatora. Na taj se način ponovno pospješuje brza i jednostavna instalacija manipulatora na ciljanoj lokaciji. Osim toga, korištenjem virtualnog Teaching Pendant-a omogućuje se testiranje i prilagodba rada cijelog sustava prije instalacije.

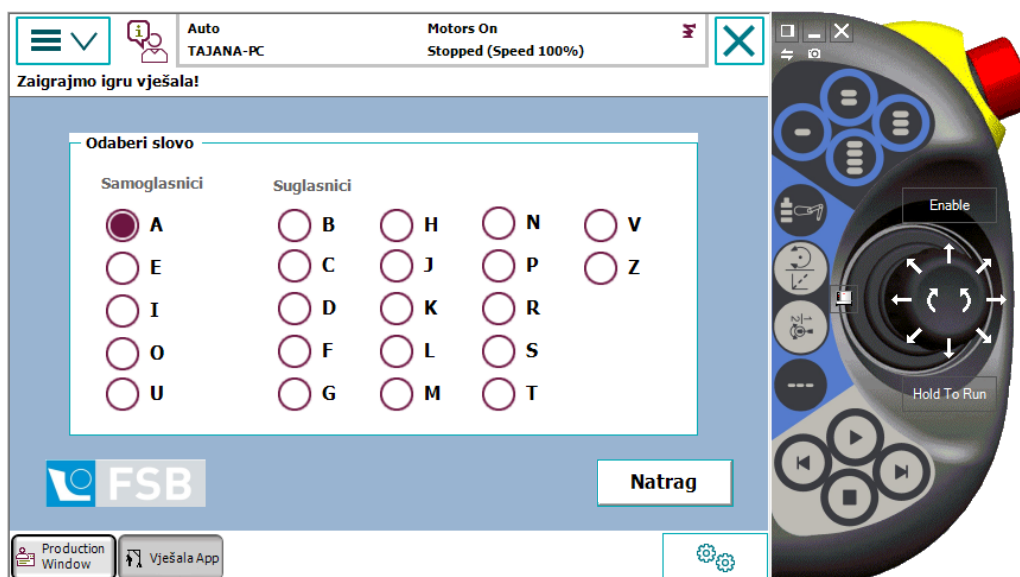
Prilagođeni alati, tj. aplikacije se kreiraju pomoću ScreenMaker-a, alata RobotStudio-a. Njegovim se korištenjem omogućuje pojednostavljenje robotskih sustava te izrada specifičnih ekrana za željene primjene (u ovom slučaju za igru "Vješala"). Takvi ekrani uključuju određeni broj međusobno raspoređenih komponenti kao što su tipke, izbornici, slike i polja sa tekstom koje omogućuju interakciju sa korisnikom. Izvođenjem određene akcije, kao što je npr. pristiskanje gumba potencijalno se izvodi određeni dio koda. Različite komponente pri tome aktiviraju različite akcije. Pri razvoju takve je aplikacije stoga bitno obratiti pažnju na najmanje dvije strane razvoja: vidljivo sučelje (raspored i konfiguracija komponenata) te procesni dio (dijelovi koda koji su posljedica aktivacije određene komponente). [9]

Kreiranjem projekta pomoću ScreenMaker-a najprije je stvoren osnovni ekran aplikacije za igru "Vješala" prikazan na slici 38. Ekran se otvara pritiskom na "Vješala App" u glavnom izborniku.



Slika 38 Osnovni ekran aplikacije za igru "Vješala"

Na osnovnom je ekranu ponuđen kratak opis igre te pravila koja se primjenjuju u okviru automatizirane igre "Vješala" obuhvaćene ovim diplomskim radom. Kako bi igrač odustao od započinjanja igre potrebno je pritisnuti tipku "Odustani" koja će zatvoriti prikazani osnovni ekran. U suprotnome, za započinjanje igre, potrebno je kliknuti tipku "Započni igru" čime se otvara novi ekran koji omogućuje odabir željenog slova. Ona su vizualno podijeljena na samoglasnike (koji su u radnom prostoru manipulatora prema slici 36 predstavljeni sa 5 komponentata i u riječima se obično javljaju češće od suglasnika) i na suglasnike (koji su u radnom prostoru manipulatora predstavljeni sa 3 komponente). Jednako kao i na osnovnom ekranu omogućeno je zatvaranje trenutno otvorenog ekrana. Pritiskom tipke "Natrag" zatvara se ekran sa izbornikom za odabir željenog slova i vraćamo se na osnovni ekran.



**Slika 39** Ekran za odabir slova aplikacije za igru "Vješala"

Ekran za odabir slova (prikazan na slici 39) kao sastavni dio igre "Vješala" načinjen je primarno od komponente pod imenom `RadioButton`. Među definiranim setom opcija `RadioButton` dopušta odabir maksimalno jedne. Odabir određenog slova pri tome se vrši klikom na kružnicu ispred odabranog slova nakon čega će se odabrana kružnica ispuniti. Odabirom narednog slova njegova se kružnica ispunjava, te se uklanja ispunjenje prethodno odabrane kružnice. [9] Kako igrač u svakom krugu igre odabire isključivo jedno slovo koje se potencijalno nalazi u riječi predstavljenom komponentama bez oznaka, među ponuđenim je opcijama korištenje `RadioButton`-a odabrano kao optimalno rješenje za obavljanje zadatka.

Nakon definiranja vidljivog sučelja bilo je nužno povezati odabir određenog `RadioButton`-a sa nizom naredbi upravljačkog koda. Odabir svakog slova je prema tome,

definiran kao boolean, tj. kao tip podataka koji isključivo može imati dvije vrijednosti: TRUE ili FALSE (isječak koda 5). Pri tome je vrijednost TRUE vezana uz odabrano slovo, tj. slovo sa ispunjenom kružnicom.

Svaki vidljivi RadioButton, u alatu ScreenMaker, funkcijom "Bind checked to a Controller Object" vezan je uz pripadajući boolean. Na taj je način uspostavljena direktna veza između odabranog boolean-a, tj. upravljačkog koda i aplikacije Teaching Pendant-a.

Ovisno o slovu odabranom od strane igrača i prema prethodno opisanim pravilima korištenja RadioButton-a u svakom trenutku od 22 dostupna boolean-a isključivo jedan uz sebe ima vrijednost TRUE. Na slici 39, na ekranu za odabir slova odabrano je slovo "A" prema kojemu su na isječku koda 5 vidljive priložene vrijednosti boolean-a.

#### Isječak koda 5 Boolean vrijednosti prema odabiru slova igrača

```
1    PERS bool slovoA := TRUE;
2    PERS bool slovoB := FALSE;
3    PERS bool slovoC := FALSE;
4    PERS bool slovoD := FALSE;
    ⋮
22   PERS bool slovoZ := FALSE;
```

Igra "Vješala" započinje pritiskom tipke "Play" koja se nalazi desno od otvorenog ekrana, tj. u donjem desnom kutu Teaching Pendant-a. Prvi je pritisak na tipku nužan kako bi se igra započela, tj. kako bi se ostvario prijenos komponenata bez oznaka i time definirala dužina riječi ili rečenice koja se pogađa. Prema pravilima igre igrač tada odabire prvo slovo za koje smatra da se potencijalno nalazi u odabranoj riječi ili rečenici, a odabir vrši klikom na kružnicu ispred tog slova. Nakon odabira igrač pritišće tipku "Play" kako bi aktivirao dio koda vezan uz odabir te određene komponente. Taj se proces ponavlja do trenutka završetka igre, tj. do trenutka kada su sve prazne komponente zamijenjene slovima ili do vješanja Stickman-a.

### 6.3. UPRAVLJANJE

Kao posljedni korak u procesu automatizacije igre "Vješala" bilo je potrebno osigurati da manipulator može obavljati prethodno navedene zadatke igrača 1, u svrhu stvaranja interaktivnog okoliša. U ovom će potpoglavlju biti prikazan proces upravljanja koji će omogućiti definiranje duljine riječi odabrane za pogađanje koristeći komponente bez oznaka te određivanje reakcije manipulatora u obliku premještaja određene komponente ovisno o odabiru

određenog slova od strane igrača 2. Glavni principi upravljanja objašnjeni u poglavlju 5 i radne pozicije stvorene u okviru osnovnog radnog okoliša primjenjuju se kao osnova za razvoj igre "Vješala".

U okviru ovog poglavlja biti će objašnjen princip upravljanja za igru "Vješala" na primjeru riječi "SCARA". Za druge je riječi, kao i za prikazani primjer, bilo potrebno razviti poseban proces, tj. poseban skup naredbi koji bi omogućio pogađanje te odabrane riječi. Dodatni se upravljački kodovi za igru "Vješala" sa različitim riječima nalaze u prilogu diplomskog rada. Način na koji se sama riječ za igru odabire također je prikazan u prilogu.

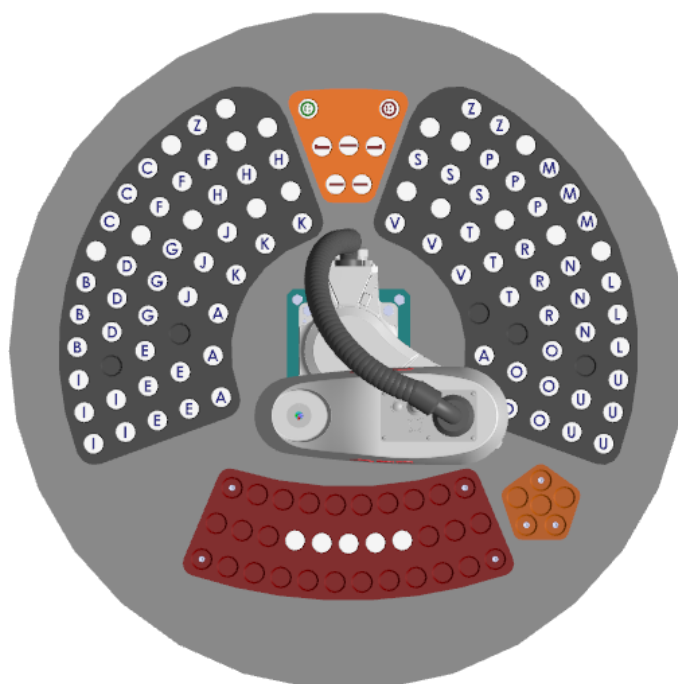
Kao što je prethodno spomenuto da bi se započela igra "Vješala" na Teaching Pendant-u potrebno je pritisnuti "Play". Time aktiviramo niz naredbi definiranih unutar glavnog (main) procesa. U ovom se slučaju pozivamo na "Vjesala\_igra1", proces definiran na liniji 49 isječka koda 6 koji obuhvaća naredbe potrebne za obavljanje zadataka igrača 1 za igru "Vješala", pri čemu je odabrana riječ koja se pogađa: "SCARA".

#### Isječak koda 6 "Vješala" – SCARA – dio I: prijenos praznih komponenta na prednju ploču

```
1   PERS bool slovoA := TRUE;
2   PERS bool pokusajA := FALSE;
3   PERS bool slovoB := FALSE;
4   PERS bool pokusajB := FALSE;
   ⋮
43  PERS bool slovoZ := FALSE;
44  PERS bool pokusajZ := FALSE;
45
46  PERS bool ispis := FALSE;
47  PERS num greska := 0;
48
49  PROC Vjesala_igra1()
50    IF ispis = FALSE THEN
51      F_L_R2_P3_Path;
52      F_F_R2_P5_Path;
53      F_L_R4_P3_Path;
54      F_F_R2_P6_Path;
55      F_R_R2_P3_Path;
56      F_F_R2_P7_Path;
57      F_R_R4_P3_Path;
58      F_F_R2_P8_Path;
59      F_R_R5_P3_Path;
60      F_F_R2_P9_Path;
61
62    ispis := TRUE
```



Aktivacijom procesa prvenstveno se vrši premještanje komponenata bez oznaka sa njihovih originalnih pozicija na finalne pozicije na prednjoj ploči. Komponente bez oznaka koje premještamo sa lijeve i desne ploče (prema slici 36) najbliže su prednjoj ploči te se na taj način nastoji skratiti vrijeme koje je potrebno za premještaj istih. Finalne pozicije na prednjoj ploči odabrane su tako da riječ "SCARA" na prednjoj ploči bude centrirana. Zbog toga se premještaj komponenata vrši na pozicije F\_R2\_P5 do F\_R2\_P9. Premještanjem komponenata bez oznaka postavlja se igra "Vješala". Rezultat priloženog koda vidljiv je na slici 40 gdje je duljina riječi "SCARA" definirana odgovarajućim brojem komponenata bez oznaka na prednjoj ploči.



**Slika 40 Definicija duljine riječi pozicioniranjem komponenata bez oznaka**

Prema pravilima igre naredni je korak odabir slova koje se potencijalno nalazi u riječi "ispisanoj" komponentama bez oznaka. Prema prethodnom poglavlju slovo odabire igrač 2 pomoću ekrana na Teaching Pendant-u. Odabir igrača se u sklopu upravljačkog koda očituje kao boolean vrijednost, tj. uz isključivo jedno odabrano slovo vezana je vrijednost TRUE. Prema toj se vrijednosti, ovisno o odabiru igrača, aktivira jedan od skupova naredbi vidljivih u isječku koda 7. Pri tome se proces razlikuje ovisno o tome nalazi li se odabrano slovo u riječi (u ovom slučaju "SCARA") koja se pogađa ili ne.

Uzmemo li za primjer da je igrač odabrao slovo "S" i kliknuo "Play" aktivirat će se dio upute od linije 67 do 71 isječka koda 7. Priloženi skup naredbi osigurati će premještanje komponente bez oznake, koja u prvotnom ispisu riječi "SCARA" zamjenjuje slovo "S", sa

pozicije F\_R2\_P5 (na prednjoj ploči) na originalnu poziciju na lijevoj ploči. Komponenta bez oznake je time uklonjena i može se zamijeniti odabranim slovom "S" koje se sa desne ploče tada prenosi na istu radnu poziciju F\_R2\_P5 na prednjoj ploči.

Na isti se način, u slučaju odabira slova "C", "A" i "R", tj. slova koja se nalaze u riječi koja se pogađa, ostvaruje premještanje prazne komponente sa odgovarajuće pozicije na prednjoj ploči na originalnu poziciju (na lijevoj ili desnoj ploči) te zamjena iste sa odgovarajućim odabranim slovom. Kao dodatni je uvjet za premještanje komponente za svako slovo dodan boolean (za npr. "slovoS" dodan je "pokusajS") kako se pri ponovnom odabiru istog slova zadani skup naredbi ne bi ponavljao. Navedeni su boolean-i vidljivi u isječku koda 6.

#### Isječak koda 7 "Vješala" – SCARA – dio II: zamjena komponenata prema odabiru igrača

```
64     ELSEIF ispis = TRUE THEN
65
66     IF slovoS = TRUE and pokusajS = FALSE THEN
67         B_F_R2_P5_Path;
68         B_L_R2_P3_Path;
69         F_R_R3_P8_Path;
70         F_F_R2_P5_Path;
71         pokusajS := TRUE;
72     ELSEIF slovoC = TRUE and pokusajC = FALSE THEN
73         :
74         :
75     ELSEIF slovoR = TRUE and pokusajR = FALSE THEN
76         B_F_R2_P8_Path;
77         B_R_R4_P3_Path;
78         F_R_R3_P4_Path;
79         F_F_R2_P8_Path;
80         pokusajR := TRUE;
81
82     ELSE
83         Vjesalo;
84         :
85         :
```

Pri odabiru bilo kojeg osim četiri navedena slova, pozivamo se na podproces "Vjesalo" (isječak koda 7 - linija 95). Navedeni se potproces prema tome aktivira za svako slovo odabrano od strane igrača 2 koje se ne nalazi u pogađanoj riječi. Za svaku se grešku igrača 2, prema pravilima igre, dodaje jedan dio tijela Stickman-a čijim se vješanjem igra završava. Kao što je prethodno spomenuto, to ostvarujemo uzastopnim pomakom šest komponenata koje zajedno tvore jednu sliku. Njihov je premještaj ostvaren nizom naredbi obuhvaćenih podprocesom "Vjesalo" prikazanih u okviru isječka koda 8.

Prva greška igrača 2 izazvati će aktivaciju premještanja tijela Stickman-a koje se nužno postavlja kao prva komponenta. U suprotnome bi, zbog lokacije ostalih komponenata, bilo nemoguće pristupiti odgovarajućoj radnoj poziciji.

Za svaku se sljedeću grešku igrača 2 numerička vrijednost uz "greška" povećava za jedan. Pri tome za različite numeričke vrijednosti (u rasponu od 1 do 6) dolazi do aktivacije različitih procesa premještanja, tj. svako povećanje vrijednosti praćeno je dodavanjem dodatnog dijela tijela na mjesto vješanja Stickman-a. Potencijalnim dostizanjem vrijednosti od šest grešaka ostvaruje se premještanje zadnjeg od ukupno 6 dijelova Stickman-a.

#### Isječak koda 8 "Vješala" – SCARA – potproces Vjesalo

```
1  PROC Vjesalo()
2      greska := greska + 1;
3      IF greska = 1 THEN
4          F_Stickman_body;
5      ELSEIF greska = 2 THEN
6          F_Stickman_armR;
7      ELSEIF greska = 3 THEN
8          F_Stickman_armL;
9          ⋮
13     ELSEIF greska = 6 THEN
14         F_Stickman_head;
15     ENDIF
16 ENDPROC
```

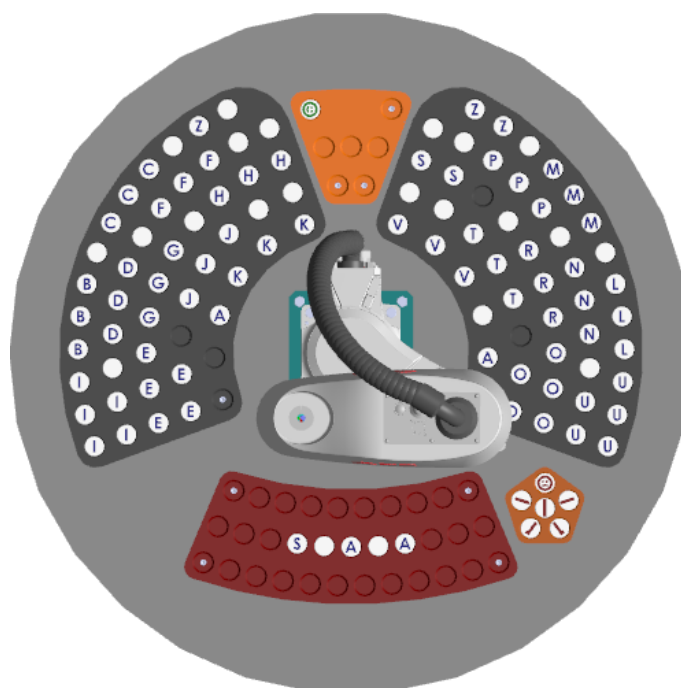
Nakon svakog se novog odabira slova prema tome provjerava dosegli broj ostvarenih grešaka, te se ovisno o njemu definira aktivacija određenog skupa naredbi. Za slučaj kada odabrana riječ nije pogođena i kada je greška manja od nula ostvariti će se: premještanje odabranog slova (za ispravan odabir na Teaching Pendant-u) ili premještanje jednog od dijelova tijela Stickman-a (za neispravan odabir na Teaching Pendant-u).

U suprotnome dolazi do završetka igre na jedan od dva načina: igrač 2 dostigao je maksimalan dopušteni broj pogrešaka (slovo koje se ne nalazi u pogađanoj riječi odabrano je šest puta i igrač 2 je time izgubio igru) prije nego je uspio pogoditi sva slova u pogađanoj riječi ili je uspio pogoditi sva slova u odabranoj riječi uz maksimalno 5 pogrešaka. U tome je slučaju igrač 2 pobjednik igre za odabranu riječ.

Neovisno o načinu završetka igre, korištene je komponente potrebno ukloniti sa prednje ploče kako bi se omogućilo ponovno igranje igre "Vješala" sa sljedećom odabranom riječi. Kao što je prethodno spomenuto, korištene se komponente vraćaju na svoje originalne pozicije, tj.

komponente sa slovima i komponente bez oznaka vraćamo na početne pozicije na lijevoj i desnoj ploči, a komponente Stickman tijela vraćamo na ploču dodanu između lijeve i desne.

Primjer izgubljene igre u kojoj je dopušteni broj grešaka dostignut prije nego je riječ pogodena vidljiv je na slici 41. Pri tome su za pogađanu riječ "SCARA" pogodena slova "S" i "A", no dopušteni je broj grešaka postignut prije nego su pogodena ostala slova riječi. Da je igra izgubljena vidljivo je na prednjoj Stickman ploči na kojoj je pozicionirano svih šest komponenata koje zajedno čine ukupnu sliku "obješenog" Stickman-a.



**Slika 41 Primjer izgubljene igre "Vješala"**

Odabrani manipulator nakon završetka izgubljene igre prvo premješta komponente sa prednje ploče, a zatim komponente sa dijelovima Stickman tijela. Prema slici je vidljivo da za komponente na prednjoj ploči razlikujemo radne pozicije sa praznim komponentama i one sa slovima. Prema isječku koda 9 za se svako slovo, ovisno o tome je li ono pogodeno ili ne, aktivira niz naredbi koji ostvaruje premještanje komponente slova ili Stickman komponente.

**Isječak koda 9 "Vješala" – SCARA – dio III: Uspostavljanje početnih uvjeta (izgubljena igra)**

```

94     ELSE
95         Vjesalo;
96
97         IF greska = 6 THEN
98             IF pokusajS = TRUE THEN
99                 B_F_R2_P5_Path;
```

```

100     B_R_R3_P8_Path;
101     ELSE
102     B_F_R2_P5_Path;
103     B_F_R2_P3_Path;
104     ENDIF
      ⋮
126     IF pokusajR = TRUE THEN
127     B_F_R2_P8_Path;
128     B_R_R3_P4_Path;
129     ELSE
130     B_F_R2_P8_Path;
131     B_R_R4_P3_Path;
132     ENDIF
134
135     Vjesalo_return;
136     Reset_vrijednosti;
137     ENDIF
138     ENDIF

```

Po završetku premještanja svih komponenta sa prednje ploče, na liniji 135 isječka koda 9 pozivamo se na potproces "Vjesalo\_return". On obuhvaća skup naredbi za povratak komponenta Stickman tijela na njihovu originalnu poziciju kako bi se omogućilo pokretanje nove igre. Za izgublenu se igru, zbog maksimalnog broja pogrešaka, uvijek aktivira upravljački kod između linija 23 i 28 isječka koda 10 koji osigurava da se uklone sve komponente sa prednje Stickman ploče. Time su sve korištene komponente vraćene na originalnu poziciju.

#### Isječak koda 10 "Vješala" – SCARA – potproces Vjesalo\_Return

```

1     PROC Vjesalo_return()
2     IF greska = 1 THEN
3     B_Stickman_body;
4     ELSEIF greska = 2 THEN
      ⋮
22    ELSEIF greska = 6 THEN
23    B_Stickman_head;
24    B_Stickman_legL;
25    B_Stickman_legR;
26    B_Stickman_armL;
27    B_Stickman_armR;
28    B_Stickman_body;
29    ENDIF
30    ENDPROC

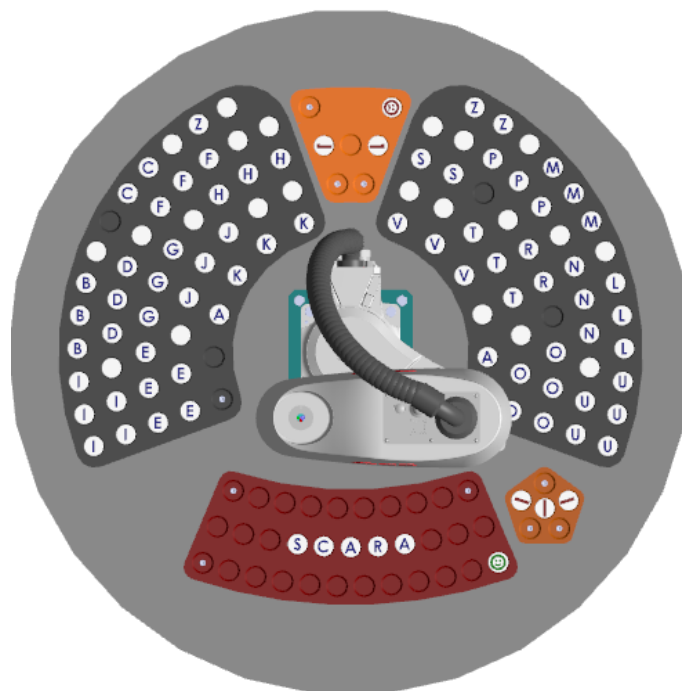
```

Za početak nove igre, osim povratka komponenta na originalne pozicije, potrebno je vratiti sve promijenjene podatke na početne vrijednosti. Boolean-e pri tome vraćamo na

FALSE, a podatak "greška" vraćamo na nulu. To ostvarujemo potprocesom na liniji 136 isječka koda 9. Time su u okolišu manipulatora ostvaruju uvjeti jednaki onima na početku igre.

Za igru koju je je igrač 2 dobio, tj. u kojoj su pogođena sva slova prije nego je dostignut dopušteni broj grešaka, proces povratka komponenata na originalnu poziciju nešto je drugačiji. Pri tome je glavni princip igre isti, tj. uklanjanjem komponenata sa prednjih ploča i vraćanjem promijenjenih podataka na originalne vrijednosti omogućava se započinjanje nove igre.

Primjer takve dobivene igre "Vješala" prikazan je na slici 42 gdje je vidljivo da su pogođena sva slova riječi "SCARA" pri čemu su u procesu ostvarene tri greške. Prema pravilima igre to je manje od ukupnog dopuštenog broja grešaka te je igrač 2 odnio pobjedu. Za dodatnu potvrdu pobjede igrača 2 na prednjoj je ploči u donjem desnom kutu ostvareno premještanje dodatne komponente koja potvrđuje da je pogođena odabrana riječ.



**Slika 42** Primjer dobivene igre "Vješala"

Za ovaj se tip završetka igre, kada su pogođena sva slova riječi, povratak komponenata na početne pozicije i uspostavljanje originalnih uvjeta s početka igre, ostvaruje prema naredbama isječka koda 11. Prema njemu se prvenstveno aktivira spomenuti potproces (isječak koda 10) koji omogućuje premještanje komponenata Stickman tijela na originalne pozicije. Broj komponenata na prednjoj Stickman ploči se pri tome za dobivenu igru razlikuje, ovisno o broju grešaka koje je igrač tijekom igre napravio. Prema tom je broju određeno u kojem se broju i koji se točno procesi premještanja komponenata Stickman tijela odvijaju.

S obzirom da su sva slova pogođena, tj. da na prednjoj ploči nema praznih komponenata, proces uklanjanja komponenata sa prednje ploče je nešto jednostavniji nego u prethodno objašnjenom primjeru gdje se na prednjoj ploči nalaze dva tipa komponenata. Prema tome se za ovaj primjer sa prednje ploče uklanjaju isključivo komponente slova.

Kao zadnji korak u procesu ostvaruje se premještanje dodane komponente (na originalnu poziciju). Osim toga, jednako prvoj opciji završetka igre, svi se podaci promijenjeni u toku igre vraćaju na originalne vrijednosti. Time je završen proces upravljanja igrom "Vješala" za odabranu riječ "SCARA" te se omogućuje započinjanje igre sa novom riječi.

#### Isječak koda 11 "Vješala" – SCARA – dio IV: Uspostavljanje početnih uvjeta (dobivena igra)

```
140      IF greska < 6 AND pokusajS = TRUE AND pokusajC = TRUE AND
141          pokusajA = TRUE AND pokusajR = TRUE THEN
142          F_Stickman_head_happy;
143          Vjesalo_return;
144          B_F_R2_P5_Path;
145          B_R_R3_P8_Path;
146          B_F_R2_P6_Path;
147          B_L_R1_P8_Path;
148          B_F_R2_P7_Path;
149          B_L_R5_P1_Path;
150          B_F_R2_P8_Path;
151          B_R_R3_P4_Path;
152          B_F_R2_P9_Path;
153          B_L_R5_P2_Path;
154          B_Stickman_head_happy;
155          Reset_vrijednosti;
156      ENDIF
157  ENDIF
158  ENDPROC
```

Prema prikazanim je isječcima kodova (te prema sličnima koji su za druge riječi priloženi u dodatku diplomskog rada) vidljivo da će se odabir slova koje je u toku igre već jednom prethodno odabrano brojati kao greška. Tako će se npr. i ponovnim odabirom slova "A" koje se nalazi u pogađanoj riječi i ponovnim odabirom slova "E" koje se ne nalazi u pogađanoj riječi aktivirati premještanje sljedećeg dijela tijela Stickman-a. Kao takvo, ovo je pravilo igre uključeno u opis koji se nalazi na početnom ekranu Teaching Pendant-a.

## 7. OPTIMIZACIJA POLOŽAJA KOMPONENATA

Raspored komponenata slova na radnim pozicijama lijeve i desne ploče, prikazan na slikama 19 i 36, temeljen je na kronološkom redosljedju pojavljivanja određenog slova u abecedi. Iznimka su pri tome komponente sa samoglasnicima koje se, uz povećani broj komponenata, nalaze na pozicijama najbližima prednjoj ploči. Kao takav, prikazani je raspored minimalno optimiziran u svrhu smanjenja vremena trajanja prenošenja određene komponente, a samim time i vremena trajanja ukupnog radnog ciklusa, načinjenog od niza individualnih prijenosa željenih komponenata. Stoga se u svrhu smanjenja vremena trajanja ukupnih radnih procesa razmatra optimizacija položaja komponenata u odnosu na postojeće radne pozicije.

Preraspodjela komponenata se pri tome razmatra isključivo za sustav u kojemu se nalaze samo komponente sa slovima (prikazan na slici 19), tj. kada se premještanjem komponenata vrši ispis. U navedenom je sustavu utjecaj optimizacije položaja komponenata najveći, tj. veći je od utjecaja optimizacije komponenata za igru "Vješala" gdje većinski postotak ukupnog radnog ciklusa manipulator miruje, a igrač vrši odabir željene komponente. Pri ispisu, s druge strane, radni ciklus uključuje niz uzastopnih kretnji manipulatora. Optimizacijom položaja komponenata i smanjenjem trajanja svake od kretnji u nizu direktno se utječe na vrijeme trajanja ukupnog radnog ciklusa željenog ispisa.

### 7.1. Frekvencija slova u riječima hrvatskog jezika

Optimizaciju položaja komponenata, tj. slova abecede, pri tome temeljimo na frekvenciji određenog slova u riječima hrvatskog jezika. Učestalost pojavljivanja svakog od slova abecede je drugačija, a prosječne su vrijednosti prikazane u tablici 7. Pri tome se slova "lj" i "nj" rasčlanjuju i pribrojavaju njihovim sastavnim slovima, a izvedena slova "č", "ć", "đ", "š" i "ž" pribrajaju se pripadajućim slovima bez dijakritika (c, s, z), dok se u slučaj slova đ koristi sličan princip kao kod slova "lj" i "nj", odnosno koristi se supstitucija "đ" = "dj".

**Tablica 7** Učestalost slova u riječima hrvatskog jezika

A	I	O	E	N	S	R	J	T	U	D
11.5%	9.8%	9%	8.4%	6.6%	5.6%	5.4%	5.1%	4.8%	4.3%	3.7%
K	V	L	M	P	C	Z	G	B	H	F
3.6%	3.5%	3.3%	3.1%	2.9%	2.8%	2.3%	1.6%	1.5%	0.8%	0.3%



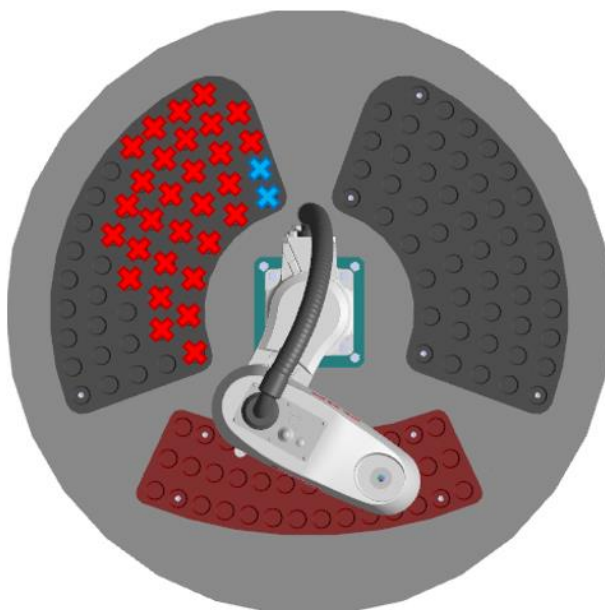
Prema navedenoj raspodjeli priložena tablica odgovara komponentama slova u radnom okolišu odabranog ABB manipulatora.

Uz optimizaciju položaja postojećih komponenta javlja se i mogućnost optimizacije samog broja svakog od korištenih slova. Primjenom postotaka učestalosti pojavljivnja određenog slova na raspoloživi broj (98) komponentata u sustavu, ona bi se u međusobno drugačije rasporedila. Direktnom bi primjenom postotaka to značilo da bi se u sustavu nalazilo 11 komponentata slova "A", 10 komponentata slova "I" i tek jedna komponenta slova "H". Slovo "F" pri tome ne bi bilo predstavljeno niti jednom komponentom u sustavu.

Uzme li se pri tome u obzir broj dostupnih radnih pozicija na prednjoj ploči, tj. činjenica da smo pri ispisu ograničeni na definiranu duljinu riječi i rečenica, predložena promjena broja komponentata u velikom broju slučajeva neće zadovoljavati potrebe odabranog ispisa. Razlog tome su manji uzorci ispisa, tj. jedna riječ ili kraća rečenica, u kojoj će učestalost određenog slova u velikoj mjeri odstupati od postotaka prikazanih u tablici 7. Stoga će se za optimizaciju koristiti već postojeća raspodjela broja slova prikazana u tablici 4, no njihov će se položaj optimizirati ovisno o vjerojatnosti pojavljivanja.

## 7.2. Standardno konfiguracijsko rješenje

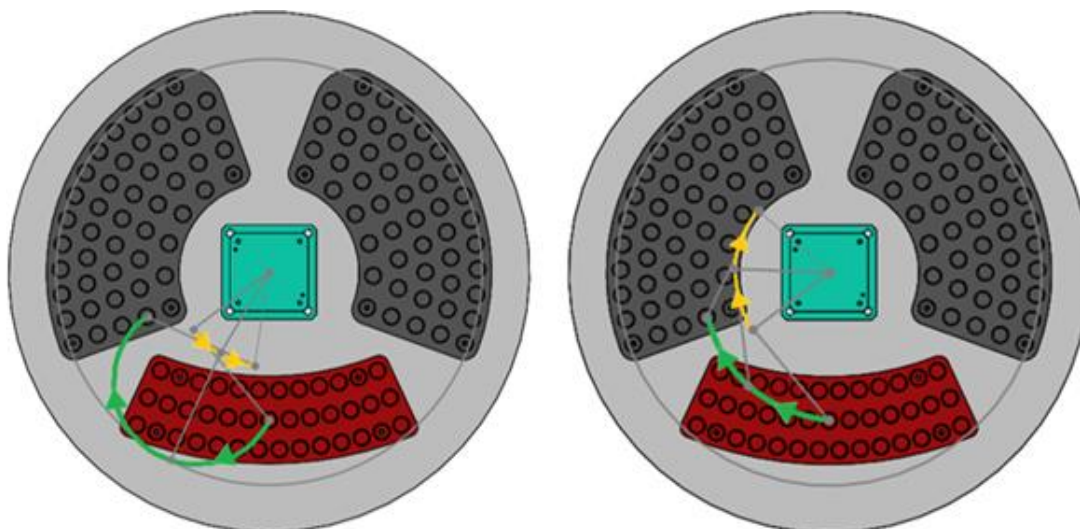
Kao prvi korak u optimizaciji maksimizira se broj radnih pozicija koje se mogu doseći istim konfiguracijskim rješenjem.



Slika 43 Radne pozicije lijeve ploče za konfiguracijsko rješenje lakat desno

Prethodno je zbog jednostavnosti odabrano da se konfiguracijsko rješenje lakat desno koristi za radne pozicije na desnoj i prednjoj ploči, dok se za radne pozicije na lijevoj ploči koristilo rješenje lakat lijevo. Sada se kao standard, za sve pozicije (na kojima je ono ostvarivo) uzima konfiguracijsko rješenje lakat desno.

Na lijevoj ploči, određeni broj radnih pozicija (označen crvenom bojom na slici 43) pri tome nije ostvariv odabranim konfiguracijskim rješenjem te se za njih pri pomaku manipulatora vrši promjena u konfiguracijsko rješenje lakat lijevo. Plavom su bojom pri tome označene pozicije koje se mogu doseći konfiguracijom lakat desno. Za pozicije na kojima je ono ostvarivo, promjena konfiguracijskog rješenja iz lakat lijevo u lakat desno direktno utječe na smanjenje duljine puta kojime se ostvaruje pomak manipulatora. Smanjenje putanje vršnog koordinatnog sustava promjenom korištenog konfiguracijskog rješenja vidljivo je na slici 44.



**Slika 44 Usporedba putanja nakon promjene konfiguracijskog rješenja**

Na priloženoj je slici konfiguracijsko rješenje na radnoj poziciji L\_R4\_P1 promijenjeno u lakat desno čime se putanja vršnog koordinatnog sustava smanjila sa približno 625 mm (slika 44, lijevo) na 362 mm (Slika 44 44, desno).

Za prikazani je pomak kao početna točka odabrana radna pozicija F\_R2\_P7 sa koje se zatim vršio pomak na finalnu radnu poziciju L\_R4\_P1. Na lijevoj je slici pri tome vidljiv prolaz vršnog koordinatnog sustava kroz točku singularnosti. Preciznije, vidljiva je tangentnost putanje sa kružnicom koja označava doseg manipulatora, tj. njegov radni prostor. U točki dodira ostvaruje se promjena konfiguracijskog rješenja.

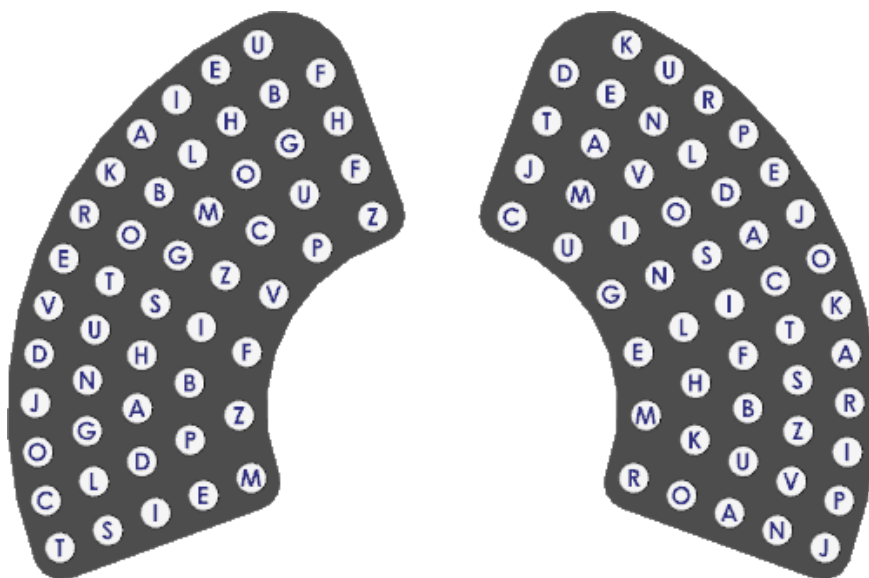
Skraćenje putanje gibanja vršnog koordinatnog sustava ima direktan utjecaj na skraćenje vremena radnog ciklusa ispisa. Korištenjem naredbe MoveJ ili MoveL nužno se

definira brzina gibanja. Time bi se, za konstantnu brzinu gibanja vršnog koordinatnog sustava v1000 mm/s, na prethodnom primjeru vrijeme pomaka sa početne na finalnu poziciju skratilo sa 0,62 s na 0,36 s. U okviru procesa ispisa, gdje se ostvaruje uzastopan niz pomaka alata manipulatora, smanjenjem trajanja svakog od pojedinačnih procesa moguće je u velikoj mjeri utjecati na ukupno trajanje radnog procesa te na taj način optimizirati radni proces.

### 7.3. Optimizacija položaja komponenata

Isti se princip (maksimalno smanjenje putanje vršnog koordinatnog sustava) primjenjuje i pri optimizaciji položaja komponenata sa slovima. Pri tome je cilj minimizirati duljinu putanja do radnih pozicija na kojima se nalaze slova sa najvećom frekvencijom, tj. pozicionirati komponente sa najvećom učestalosti što bliže prednjoj ploči. Na taj se način direktno utječe na smanjenje ukupnog radnog ciklusa procesa ispisa. Komponente sa slovima se stoga preraspoređuju prema frekvencijama pojedinog slova iz tablice 7.

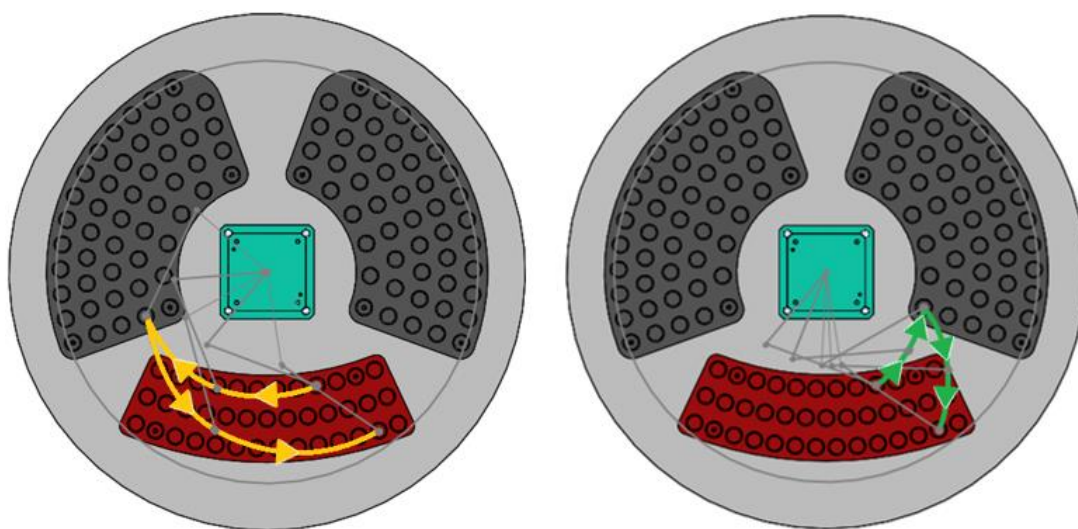
Kako se na radnim pozicija lijeve i desne ploče svaka komponenta nalazi više od jednog puta u obzir su uzete i vjerojatnosti ponovnog pojavljivanja slova u određenoj riječi što je u velikoj mjeri utjecalo na generalni raspored komponenata. Predložena optimizacija položaja komponenata slova nalazi se na slici ispod. Frekvencija pojavljivanja određenog slova odredila je pri tome radnu poziciju pripadajuće komponente postavljajući pri tome komponente čija je vjerojatnost pojavljivanja veća niže na prikazanoj slici.



Slika 45 Optimizirani položaj komponenata

No za prednju ploču sa nekoliko radnih pozicija putanja do određene komponente nije definirana samo njenom udaljenosti od prednje ploče već i točnom pozicijom vršnog koordinatnog sustava manipulatora. Za sustav u kojemu se svako od prisutnih slova pojavljuje više od jednog puta, trenutna pozicija manipulatora određuje koja je od komponenata najpogodnija za prenošenje na prednju ploču. Komponenta odabranog slova koja se nalazi najbliže prednjoj ploči, u većini će slučajeva imati najkraću putanju u odnosu na vršni koordinatni sustav manipulatora, no u nekolicini će se slučajeva za premještaj odabrati komponenta koja je dalje od prednje ploče no bliže alatu manipulatora. Stoga se usporedba putanja u svrhu pronalaska najbliže javlja kao dodatna mogućnost pri optimizaciji procesa ispisa te nudi mogućnost za dodatnim smanjenjem trajanja ukupnog procesa.

Pri usporedbi procesa premještaja, uz udaljenost vršnog koordinatnog sustava od odabrane komponente (tj. putanje prihvata komponente) razmatramo i duljinu putanje od odabrane komponente do njene finalne pozicije (tj. putanju premještaja odabrane komponente). Na taj se način zaokružuje proces premještaja te se omogućuje kvalitetnija usporedba procesa.



**Slika 46 Usporedba udaljenosti iste komponente na dvije različite pozicije**

Primjer situacije gdje se među više od dvije mogućnosti odabire optimalnija, prikazan je na slici 46. Zamislimo pri tome da se identične komponente odabranog slova nalaze na radnoj poziciji L\_R4\_P1 i R\_R5\_P1. Na oba su primjera vidljive dvije putanje. Prva od njih prikazuje pomak manipulatora sa početkom u radnoj poziciji F\_R1\_P9, a druga pomak manipulatora prema odabranoj finalnoj poziciji F\_R3\_P13. Na lijevoj se slici pri tome željena komponenta nalazi na lijevoj ploči, tj. na radnoj poziciji L\_R4\_P1, a na desnoj slici komponenta se nalazi na radnoj poziciji R\_5\_P1. Duljine prikazanih pitanja pri tome se vidno razlikuju.

Na prikazanom je primjeru vidljivo da trenutna pozicija vršnog koordinatnog sustava manipulatora i odabrana radna pozicija na koju se premješta komponenta u velikoj mjeri utječu na duljinu putanja vršnog koordinatnog sustava manipulatora.

Isti je princip moguće primijeniti i u procesu ispisa gdje bi se za npr. četiri komponente sa slovom "B" uspoređivale udaljenosti do svih pripadajućih radnih pozicija. Na taj bi se način dodatno optimizirao proces premještanja komponenata jer bi za premještanje uvijek odabirala ona komponenta čija je putanja pomaka najkraća, tj. za čije je premještanje potrebno najmanje vremena. U odnosu na općenitu optimizaciju položaja komponenata na slici 45, odabir komponenata ovisan o duljini pripadajućih putanja (kao dodatak na već optimiziran položaj komponenata) imao bi manji utjecaj na smanjenje trajanja ukupnog radnog ciklusa, no omogućavao bi, u slučaju potrebe, dodatnu optimizaciju istog.

## 8. ZAKLJUČAK

Industrijski robotski manipulatori u današnje se vrijeme nalaze se u širokom spektru primjena. Osim automatizacije proizvodnih procesa, što se ističe kao najčešća primjena robotskih ruku, često se primjenjuju za edukacijske svrhe, za repetitivne i zamorne zadatke te za primjene u okviru različitih industrija kao što je npr. medicina. No u nekoliko se zadnjih godina posebna pažnja posvećuje interaktivnosti između čovjeka i manipulatora. Osim kolaborativnih robota, koji se obično primjenjuju kao direktna pomoć pri obavljanju određenog zadatka, interaktivnost nalazimo i u sustavima gdje je za rad sustava jednako bitan i čovjek i manipulator. Preciznije, svrha sustava je njihova interakcija.

Proces ispisa nizanjem određenog broja komponenata i igra "Vješala", obuhvaćeni ovim diplomskim radom, primjer su takve primjene. To je posebno točno za igru "Vješala" gdje je jedan, od nužno dva potrebna igrača za igru, upravo u tu svrhu odabrani ABB manipulator. Input drugog igrača, tj. korisnika pri tome je definiran kao odabir slova u pogađanoj riječi, a nužan je za definiranje putanje manipulatora, a samim time i za rad sustava. Obuhvaćenim se diplomskim radom ponovno prikazuje širina spektra primjene industrijskih robotskih ruku, a u okviru ovog se primjera ona velikim dijelom odnosi na automatizaciju društvene igre kao odabranog oblika interakcije.

U okviru prikazanog je rada, osim navedenih primjena, cilj bio stvoriti sustav koji bi omogućio laku prenamjenu, prilagodbu i nadogradnje. Kao glavni medij interakcije korištene su valjkaste komponente sa različitim oznakama (najčešće sa oznakom određenog slova) čijim se nizanjem, tj. međusobnim položajem na finalnoj lokaciji omogućavalo kreiranje riječi ili rečenica. Dodavanjem dodatnih komponenata u sustav te zamjenom nekih od postojećih oznaka u istom je radnom okolišu (uz promjenu upravljačkog koda) kreirana igra "Vješala".

Na sličan je način, uz minimalne promjene, prikazani sustav moguće primijeniti za obavljanje nove interaktivne funkcije. Proces prenamjene pri tome ponovno uključuje konstrukciju i razvoj potrebnih dijelova te razvoj upravljačkog koda kao sastavne dijelove razvojnog procesa automatizacije.

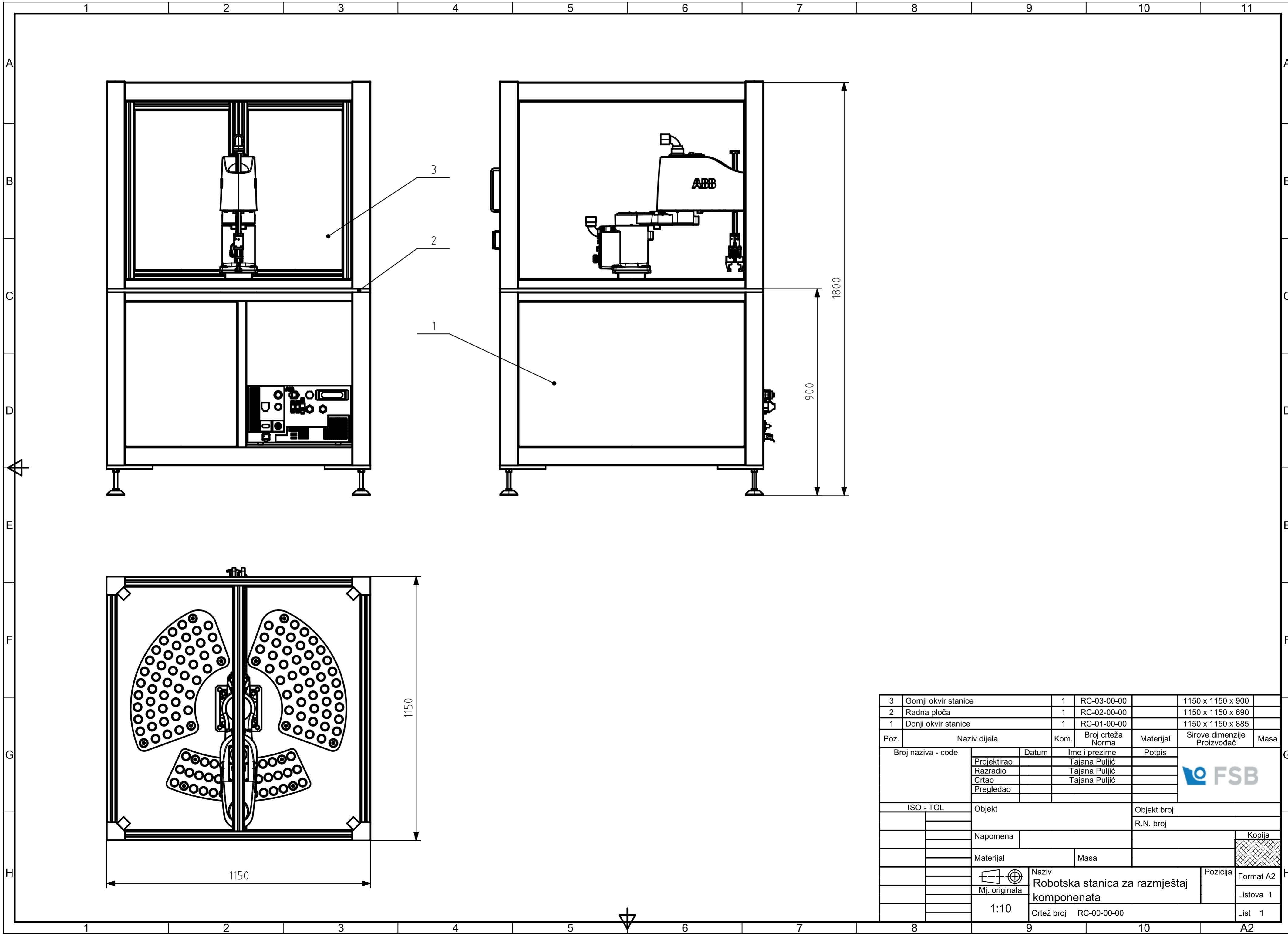
## 9. LITERATURA

- [1] S. H. M. V. Mark W. Spong, Robot Modeling and Control, John Wiley and sons, inc. , 2006.
- [2] ABB, »ABB,« ABB, [Mrežno]. Available: <https://new.abb.com/products/robotics/industrial-robots/irb-910sc>. [Pokušaj pristupa 11. svibanj 2020.].
- [3] Z. L. S. S. S. Richard M. Murray, A Mathematical Introduction to, CRC Press, 1994.
- [4] Fanuc, »How to Know When A SCARA Robot is the Right Choice for Your Application,« 07 Studeni 2018. [Mrežno]. Available: <https://www.fanucamerica.com/news-resources/articles/how-to-know-when-a-scara-robot-is-the-right-choice-for-your-application>.
- [5] RoboDK, »RoboDK,« [Mrežno]. Available: <https://robodk.com/offline-programming>.
- [6] ABB, »ABB,« 14. 05. 2020.. [Mrežno]. Available: <https://new.abb.com/products/robotics/robotstudio>.
- [7] Mecademic, »What are singularities in a six-axis robot arm?,« 18. 05. 2020.. [Mrežno]. Available: <https://www.mecademic.com/resources/Singularities/Robot-singularities>.
- [8] ABB Robotics, Technical reference manual - RAPID Instructions, Functions and Data types, Västerås : ABB, 2010.
- [9] ABB Robotics, Operating manual - RobotStudio, Västerås: ABB, 2010.

## **PRILOZI**

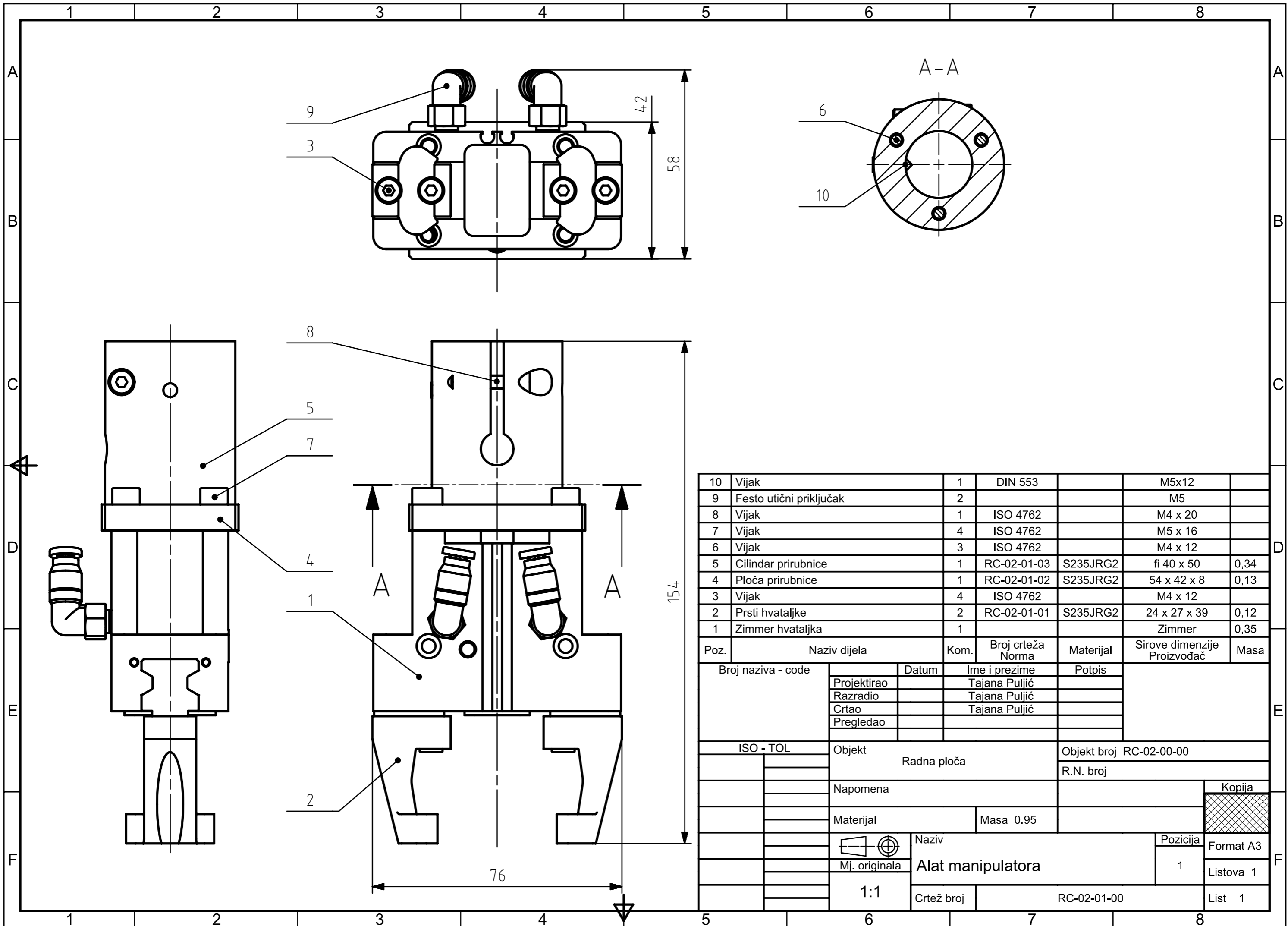
- I. CD-R disc
- II. Tehnička dokumentacija
- III. Položaj komponenata radnog prostora u odnosu na radne pozicije
- IV. RAPID programski kodovi za igre "Vjesala"



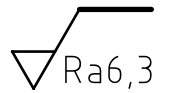


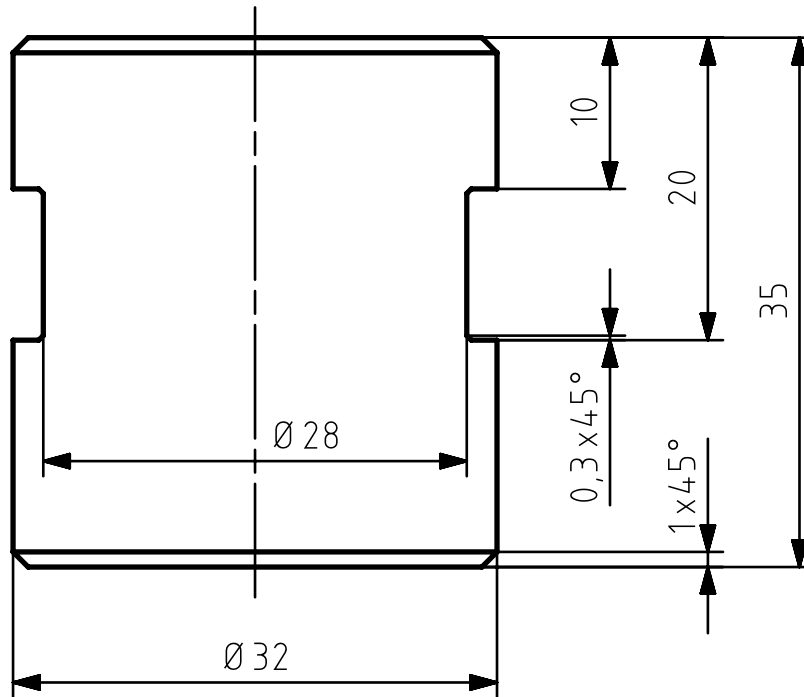
3	Gornji okvir stanice	1	RC-03-00-00		1150 x 1150 x 900	
2	Radna ploča	1	RC-02-00-00		1150 x 1150 x 690	
1	Donji okvir stanice	1	RC-01-00-00		1150 x 1150 x 885	
Poz.	Naziv dijela	Kom.	Broj crteža Norma	Materijal	Sirove dimenzije Proizvođač	Masa
Broj naziva - code		Datum	Ime i prezime		Potpis	
Projektirao			Tajana Puljić			
Razradio			Tajana Puljić			
Crtao			Tajana Puljić			
Pregledao						
ISO - TOL		Objekt			Objekt broj	
					R.N. broj	
		Napomena			Kopija	
		Materijal			Masa	
					Naziv	
		Mj. originala			<b>Robotska stanica za razmještaj komponentata</b>	
		1:10			Pozicija Format A2	
		Crtež broj RC-00-00-00			Listova 1 List 1	

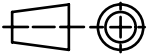




10	Vijak	1	DIN 553		M5x12	
9	Festo utični priključak	2			M5	
8	Vijak	1	ISO 4762		M4 x 20	
7	Vijak	4	ISO 4762		M5 x 16	
6	Vijak	3	ISO 4762		M4 x 12	
5	Cilindar prirubnice	1	RC-02-01-03	S235JRG2	fi 40 x 50	0,34
4	Ploča prirubnice	1	RC-02-01-02	S235JRG2	54 x 42 x 8	0,13
3	Vijak	4	ISO 4762		M4 x 12	
2	Prsti hvataljke	2	RC-02-01-01	S235JRG2	24 x 27 x 39	0,12
1	Zimmer hvataljka	1			Zimmer	0,35
Poz.	Naziv dijela	Kom.	Broj crteža Norma	Materijal	Sirove dimenzije Proizvođač	Masa
Broj naziva - code		Datum	Ime i prezime		Potpis	
Projektirao			Tajana Puljić			
Razradio			Tajana Puljić			
Crtao			Tajana Puljić			
Pregledao						
ISO - TOL		Objekt	Radna ploča		Objekt broj RC-02-00-00	
					R.N. broj	
Napomena					Kopija	
Materijal		Masa 0.95				
Mj. originala		Naziv			Pozicija	Format A3
1:1		Alat manipulatora			1	Listova 1
		Crtež broj	RC-02-01-00			List 1

 Ra6,3



Broj naziva - code	Datum	Ime i prezime	Potpis	
	Projektirao	Tajana Puljić		
	Razradio	Tajana Puljić		
	Crtao	Tajana Puljić		
	Pregledao			
ISO - TOL	Objekt	Radna ploča		Objekt broj RC-02-00-00
				R.N. broj
	Napomena			Kopija
	Materijal	ABS	Masa 0.03	
		Naziv		Pozicija
	Mj. originala	Osnovni oblik komponente		10
	2:1	Crtež broj RC-02-00-10		List 1
				List 1

A1	F_L_R5_P1_Path
A2	F_L_R5_P2_Path
A3	F_L_R5_P3_Path
A4	F_R_R5_P1_Path
A5	F_R_R5_P2_Path
A6	F_R_R5_P3_Path
B1	F_L_R1_P4_Path
B2	F_L_R1_P5_Path
B3	F_L_R1_P6_Path
B4	F_L_R1_P7_Path
C1	F_L_R1_P8_Path
C2	F_L_R1_P9_Path
C3	F_L_R1_P10_Path
C4	F_L_R1_P11_Path
D1	F_L_R2_P4_Path
D2	F_L_R2_P5_Path
D3	F_L_R2_P6_Path
D4	F_L_R2_P7_Path
E1	F_L_R3_P1_Path
E2	F_L_R3_P2_Path
E3	F_L_R3_P3_Path
E4	F_L_R4_P1_Path
E5	F_L_R4_P2_Path
E6	F_L_R4_P3_Path
F1	F_L_R2_P8_Path
F2	F_L_R2_P9_Path
F3	F_L_R2_P10_Path
F4	F_L_R2_P11_Path
G1	F_L_R3_P4_Path
G2	F_L_R3_P5_Path
G3	F_L_R3_P6_Path
G4	F_L_R3_P7_Path
H1	F_L_R3_P8_Path
H2	F_L_R3_P9_Path
H3	F_L_R3_P10_Path
H4	F_L_R2_P12_Path
I1	F_L_R1_P1_Path
I2	F_L_R1_P2_Path
I3	F_L_R1_P3_Path
I4	F_L_R2_P1_Path
I5	F_L_R2_P2_Path
I6	F_L_R2_P3_Path
J1	F_L_R4_P4_Path
J2	F_L_R4_P5_Path
J3	F_L_R4_P6_Path
J4	F_L_R4_P7_Path
K1	F_L_R5_P4_Path
K2	F_L_R5_P5_Path
K3	F_L_R5_P6_Path
K4	F_L_R4_P8_Path

L1	F_R_R1_P4_Path
L2	F_R_R1_P5_Path
L3	F_R_R1_P6_Path
L4	F_R_R1_P7_Path
M1	F_R_R1_P8_Path
M2	F_R_R1_P9_Path
M3	F_R_R1_P10_Path
M4	F_R_R1_P11_Path
N1	F_R_R2_P4_Path
N2	F_R_R2_P5_Path
N3	F_R_R2_P6_Path
N4	F_R_R2_P7_Path
O1	F_R_R3_P1_Path
O2	F_R_R3_P2_Path
O3	F_R_R3_P3_Path
O4	F_R_R4_P1_Path
O5	F_R_R4_P2_Path
O6	F_R_R4_P3_Path
P1	F_R_R2_P8_Path
P2	F_R_R2_P9_Path
P3	F_R_R2_P10_Path
P4	F_R_R2_P11_Path
R1	F_R_R3_P4_Path
R2	F_R_R3_P5_Path
R3	F_R_R3_P6_Path
R4	F_R_R3_P7_Path
S1	F_R_R3_P8_Path
S2	F_R_P3_P9_Path
S3	F_R_R3_P10_Path
S4	F_R_R2_P12_Path
T1	F_R_R4_P4_Path
T2	F_R_R4_P5_Path
T3	F_R_R4_P6_Path
T4	F_R_R4_P7_Path
U1	F_R_R1_P1_Path
U2	F_R_R1_P2_Path
U3	F_R_R1_P3_Path
U4	F_R_R2_P1_Path
U5	F_R_R2_P2_Path
U6	F_R_R2_P3_Path
V1	F_R_R5_P4_Path
V2	F_R_R5_P5_Path
V3	F_R_R5_P6_Path
V4	F_R_R4_P8_Path
Z1	F_R_R1_P12_Path
Z2	F_R_R1_P13_Path
Z3	F_L_R1_P12_Path
Z4	F_L_R1_P13_Path

1	F_L_R2_P3_Path
2	F_L_R4_P3_Path
3	F_R_R2_P3_Path
4	F_R_R4_P3_Path
5	F_R_R4_P5_Path
6	F_L_R1_P7_Path
7	F_L_R2_P7_Path
8	F_L_R3_P7_Path
9	F_L_R4_P7_Path
10	F_L_R4_P8_Path
11	F_R_R1_P7_Path
12	F_R_R2_P7_Path
13	F_R_R3_P7_Path
14	F_R_R4_P7_Path
15	F_R_R4_P8_Path
16	F_L_R1_P11_Path
17	F_L_R2_P11_Path
18	F_L_R2_P12_Path
19	F_R_R1_P11_Path
20	F_R_R2_P8_Path
21	F_R_R2_P12_Path
22	F_L_R1_P13_Path

## **RAPID PROGRAMSKI KODOVI ZA IGRE "VJESALA"**

```
PROC main()
  IF igra1 = TRUE THEN
    Vjesala_igra1;
  ELSEIF igra2= TRUE THEN
    Vjesala_igra2;
  ELSEIF igra3= TRUE THEN
    Vjesala_igra3;
  ELSE
    Vjesala_igra4;
  ENDIF
ENDPROC
```

**! IGRE VJESALA****PROC** Vjesala\_igral()**!SCARA****IF** ispis = **FALSE THEN**

F\_L\_R2\_P3\_Path;  
F\_F\_R2\_P5\_Path;  
F\_L\_R4\_P3\_Path;  
F\_F\_R2\_P6\_Path;  
F\_R\_R2\_P3\_Path;  
F\_F\_R2\_P7\_Path;  
F\_R\_R4\_P3\_Path;  
F\_F\_R2\_P8\_Path;  
F\_R\_R5\_P3\_Path;  
F\_F\_R2\_P9\_Path;

ispis := **TRUE**;**ELSEIF** ispis = **TRUE THEN****IF** slovoS = **TRUE and** pokusajS = **FALSE THEN**

B\_F\_R2\_P5\_Path;  
B\_L\_R2\_P3\_Path;  
F\_R\_R3\_P8\_Path;  
F\_F\_R2\_P5\_Path;  
pokusajS := **TRUE**;

**ELSEIF** slovoC = **TRUE and** pokusajC = **FALSE THEN**

B\_F\_R2\_P6\_Path;  
B\_L\_R4\_P3\_Path;  
F\_L\_R1\_P8\_Path;  
F\_F\_R2\_P6\_Path;  
pokusajC := **TRUE**;

**ELSEIF** slovoA = **TRUE and** pokusajA = **FALSE THEN**

B\_F\_R2\_P7\_Path;  
B\_R\_R2\_P3\_Path;  
F\_L\_R5\_P1\_Path;  
F\_F\_R2\_P7\_Path;  
B\_F\_R2\_P9\_Path;  
B\_R\_R5\_P3\_Path;  
F\_L\_R5\_P2\_Path;  
F\_F\_R2\_P9\_Path;  
pokusajA := **TRUE**;

**ELSEIF** slovoR = **TRUE and** pokusajR = **FALSE THEN**

B\_F\_R2\_P8\_Path;  
B\_R\_R4\_P3\_Path;  
F\_R\_R3\_P4\_Path;  
F\_F\_R2\_P8\_Path;  
pokusajR := **TRUE**;

```
ELSE
  Vjesala;

  IF greska = 6 THEN
    IF pokusajS = TRUE THEN
      B_F_R2_P5_Path;
      B_R_R3_P8_Path;
    ELSE
      B_F_R2_P5_Path;
      B_L_R2_P3_Path;
    ENDIF
    IF pokusajC = TRUE THEN
      B_F_R2_P6_Path;
      B_L_R1_P8_Path;
    ELSE
      B_F_R2_P6_Path;
      B_L_R4_P3_Path;
    ENDIF
    IF pokusajA = TRUE THEN
      B_F_R2_P7_Path;
      B_L_R5_P1_Path;
      B_F_R2_P9_Path;
      B_L_R5_P2_Path;
    ELSE
      B_F_R2_P7_Path;
      B_R_R2_P3_Path;
      B_F_R2_P9_Path;
      B_R_R5_P3_Path;
    ENDIF
    IF pokusajR = TRUE THEN
      B_F_R2_P8_Path;
      B_R_R3_P4_Path;
    ELSE
      B_F_R2_P8_Path;
      B_R_R4_P3_Path;
    ENDIF

    Vjesala_return;
    Reset_vrijednosti;
    igra1 := FALSE;
    igra2 := TRUE;

  ENDIF
ENDIF
```

```
IF greska < 6 AND pokusajS = TRUE AND pokusajC = TRUE AND pokusajA =  
TRUE AND pokusajR = TRUE THEN  
    F_Stickman_head_happy;  
    Vjesala_return;  
    B_F_R2_P5_Path;  
    B_R_R3_P8_Path;  
    B_F_R2_P6_Path;  
    B_L_R1_P8_Path;  
    B_F_R2_P7_Path;  
    B_L_R5_P1_Path;  
    B_F_R2_P8_Path;  
    B_R_R3_P4_Path;  
    B_F_R2_P9_Path;  
    B_L_R5_P2_Path;  
  
    B_Stickman_head_happy;  
    Reset_vrijednosti;  
    igra1 := FALSE;  
    igra2 := TRUE;  
ENDIF  
ENDIF  
ENDPROC
```



```
PROC Vjesala_igra2()
  !MANIPULATOR
  IF ispis = FALSE THEN
    F_L_R2_P3_Path;
    F_F_R2_P2_Path;
    F_L_R4_P3_Path;
    F_F_R2_P3_Path;
    F_R_R2_P3_Path;
    F_F_R2_P4_Path;
    F_R_R4_P3_Path;
    F_F_R2_P5_Path;
    F_R_R5_P3_Path;
    F_F_R2_P6_Path;
    F_L_R1_P7_Path;
    F_F_R2_P7_Path;
    F_L_R2_P7_Path;
    F_F_R2_P8_Path;
    F_L_R3_P7_Path;
    F_F_R2_P9_Path;
    F_L_R4_P7_Path;
    F_F_R2_P10_Path;
    F_L_R4_P8_Path;
    F_F_R2_P11_Path;
    F_R_R1_P7_Path;
    F_F_R2_P12_Path;

    ispis := TRUE;

  ELSEIF ispis = TRUE THEN

    IF slovoM = TRUE and pokusajM = FALSE THEN
      B_F_R2_P2_Path;
      B_L_R2_P3_Path;
      F_R_R1_P8_Path;
      F_F_R2_P2_Path;
      pokusajM := TRUE;
    ELSEIF slovoA = TRUE and pokusajA = FALSE THEN
      B_F_R2_P3_Path;
      B_L_R4_P3_Path;
      F_L_R5_P1_Path;
      F_F_R2_P3_Path;
      B_F_R2_P9_Path;
      B_L_R3_P7_Path;
      F_L_R5_P2_Path;
      F_F_R2_P9_Path;
      pokusajA := TRUE;
    ELSEIF slovoN = TRUE and pokusajN = FALSE THEN
      B_F_R2_P4_Path;
```

```
B_R_R2_P3_Path;  
F_R_R2_P4_Path;  
F_F_R2_P4_Path;  
pokusajN := TRUE;  
ELSEIF slovoI = TRUE and pokusajI = FALSE THEN  
B_F_R2_P5_Path;  
B_R_R4_P3_Path;  
F_L_R1_P1_Path;  
F_F_R2_P5_Path;  
pokusajI := TRUE;  
ELSEIF slovoP = TRUE and pokusajP = FALSE THEN  
B_F_R2_P6_Path;  
B_R_R5_P3_Path;  
F_R_R2_P8_Path;  
F_F_R2_P6_Path;  
pokusajP := TRUE;  
ELSEIF slovoU = TRUE and pokusajU = FALSE THEN  
B_F_R2_P7_Path;  
B_L_R1_P7_Path;  
F_R_R1_P1_Path;  
F_F_R2_P7_Path;  
pokusajU := TRUE;  
ELSEIF slovoL = TRUE and pokusajL = FALSE THEN  
B_F_R2_P8_Path;  
B_L_R2_P7_Path;  
F_R_R1_P4_Path;  
F_F_R2_P8_Path;  
pokusajL := TRUE;  
ELSEIF slovoT = TRUE and pokusajT = FALSE THEN  
B_F_R2_P10_Path;  
B_L_R4_P7_Path;  
F_R_R4_P4_Path;  
F_F_R2_P10_Path;  
pokusajT := TRUE;  
ELSEIF slovoO = TRUE and pokusajO = FALSE THEN  
B_F_R2_P11_Path;  
B_L_R4_P8_Path;  
F_R_R3_P1_Path;  
F_F_R2_P11_Path;  
pokusajO := TRUE;  
ELSEIF slovoR = TRUE and pokusajR = FALSE THEN  
B_F_R2_P12_Path;  
B_R_R1_P7_Path;  
F_R_R3_P4_Path;  
F_F_R2_P12_Path;  
pokusajR := TRUE;  
  
ELSE
```

Vjesala;

```
IF greska = 6 THEN

    IF pokusajM = TRUE THEN
        B_F_R2_P2_Path;
        B_R_R1_P8_Path;
    ELSE
        B_F_R2_P2_Path;
        B_L_R2_P3_Path;
    ENDIF
    IF pokusajA = TRUE THEN
        B_F_R2_P3_Path;
        B_L_R5_P1_Path;
        B_F_R2_P9_Path;
        B_L_R5_P2_Path;
    ELSE
        B_F_R2_P3_Path;
        B_L_R4_P3_Path;
        B_F_R2_P9_Path;
        B_L_R3_P7_Path;
    ENDIF
    IF pokusajN = TRUE THEN
        B_F_R2_P4_Path;
        B_R_R2_P4_Path;
    ELSE
        B_F_R2_P4_Path;
        B_R_R2_P3_Path;
    ENDIF
    IF pokusajI = TRUE THEN
        B_F_R2_P5_Path;
        B_L_R1_P1_Path;
    ELSE
        B_F_R2_P5_Path;
        B_R_R4_P3_Path;
    ENDIF
    IF pokusajP = TRUE THEN
        B_F_R2_P6_Path;
        B_R_R2_P8_Path;
    ELSE
        B_F_R2_P6_Path;
        B_R_R5_P3_Path;
    ENDIF
    IF pokusajU = TRUE THEN
        B_F_R2_P7_Path;
        B_R_R1_P1_Path;
    ELSE
        B_F_R2_P7_Path;
```

```
        B_L_R1_P7_Path;
    ENDIF
    IF pokusajL = TRUE THEN
        B_F_R2_P8_Path;
        B_R_R1_P4_Path;
    ELSE
        B_F_R2_P8_Path;
        B_L_R2_P7_Path;
    ENDIF
    IF pokusajT = TRUE THEN
        B_F_R2_P10_Path;
        B_R_R4_P4_Path;
    ELSE
        B_F_R2_P10_Path;
        B_L_R4_P7_Path;
    ENDIF
    IF pokusajO = TRUE THEN
        B_F_R2_P11_Path;
        B_R_R3_P1_Path;
    ELSE
        B_F_R2_P11_Path;
        B_L_R4_P8_Path;
    ENDIF
    IF pokusajR = TRUE THEN
        B_F_R2_P12_Path;
        B_R_R3_P4_Path;
    ELSE
        B_F_R2_P12_Path;
        B_R_R1_P7_Path;
    ENDIF

    Vjesala_return;
    Reset_vrijednosti;
    igra2 := FALSE;
    igra3 := TRUE;

    ENDIF
ENDIF

    IF greska < 6 AND pokusajM = TRUE AND pokusajA = TRUE AND pokusajN =
    TRUE AND pokusajI AND pokusajP = TRUE AND pokusajU = TRUE AND pokusajL =
    TRUE AND pokusajT AND pokusajO = TRUE AND pokusajR = TRUE THEN
        F_Stickman_head_happy;
        Vjesala_return;

        B_F_R2_P2_Path;
        B_R_R1_P8_Path;
        B_F_R2_P3_Path;
```

```
B_L_R5_P1_Path;  
B_F_R2_P4_Path;  
B_R_R2_P4_Path;  
B_F_R2_P5_Path;  
B_L_R1_P1_Path;  
B_F_R2_P6_Path;  
B_R_R2_P8_Path;  
B_F_R2_P7_Path;  
B_R_R1_P1_Path;  
B_F_R2_P8_Path;  
B_R_R1_P4_Path;  
B_F_R2_P9_Path;  
B_L_R5_P2_Path;  
B_F_R2_P10_Path;  
B_R_R4_P4_Path;  
B_F_R2_P11_Path;  
B_R_R3_P1_Path;  
B_F_R2_P12_Path;  
B_R_R3_P4_Path;  
  
B_Stickman_head_happy;  
Reset_vrijednosti;  
igra2 := FALSE;  
igra3 := TRUE;
```

```
ENDIF  
ENDIF  
ENDPROC
```

```
PROC Vjesala_igra3()
  !MEHANIZMI_I_ROBOTI
  IF ispis = FALSE THEN

    F_L_R2_P3_Path;
    F_F_R1_P3_Path;
    F_L_R4_P3_Path;
    F_F_R1_P4_Path;
    F_R_R2_P3_Path;
    F_F_R1_P5_Path;
    F_R_R4_P3_Path;
    F_F_R1_P6_Path;
    F_R_R5_P3_Path;
    F_F_R1_P7_Path;
    F_L_R1_P7_Path;
    F_F_R1_P8_Path;
    F_L_R2_P7_Path;
    F_F_R1_P9_Path;
    F_L_R3_P7_Path;
    F_F_R1_P10_Path;
    F_L_R4_P7_Path;
    F_F_R1_P11_Path;

    F_L_R4_P8_Path;
    F_F_R2_P7_Path;

    F_R_R1_P7_Path;
    F_F_R3_P5_Path;
    F_R_R2_P7_Path;
    F_F_R3_P6_Path;
    F_R_R3_P7_Path;
    F_F_R3_P7_Path;
    F_R_R4_P7_Path;
    F_F_R3_P8_Path;
    F_R_R4_P8_Path;
    F_F_R3_P9_Path;
    F_L_R1_P11_Path;
    F_F_R3_P10_Path;

    ispis := TRUE;

  ELSEIF ispis = TRUE THEN

    IF slovoM = TRUE and pokusajM = FALSE THEN
      B_F_R1_P3_Path;
      B_L_R2_P3_Path;
      F_R_R1_P8_Path;
      F_F_R1_P3_Path;
```

```
B_F_R1_P10_Path;  
B_L_R3_P7_Path;  
F_R_R1_P9_Path;  
F_F_R1_P10_Path;  
pokusajM := TRUE;  
ELSEIF slovoE = TRUE and pokusajE = FALSE THEN  
B_F_R1_P4_Path;  
B_L_R4_P3_Path;  
F_L_R3_P1_Path;  
F_F_R1_P4_Path;  
pokusajE := TRUE;  
ELSEIF slovoH = TRUE and pokusajH = FALSE THEN  
B_F_R1_P5_Path;  
B_R_R2_P3_Path;  
F_L_R3_P8_Path;  
F_F_R1_P5_Path;  
pokusajH := TRUE;  
ELSEIF slovoA = TRUE and pokusajA = FALSE THEN  
B_F_R1_P6_Path;  
B_R_R4_P3_Path;  
F_L_R5_P1_Path;  
F_F_R1_P6_Path;  
pokusajA := TRUE;  
ELSEIF slovoN = TRUE and pokusajN = FALSE THEN  
B_F_R1_P7_Path;  
B_R_R5_P3_Path;  
F_R_R2_P4_Path;  
F_F_R1_P7_Path;  
pokusajN := TRUE;  
ELSEIF slovoI = TRUE and pokusajI = FALSE THEN  
B_F_R1_P8_Path;  
B_L_R1_P7_Path;  
F_L_R1_P1_Path;  
F_F_R1_P8_Path;  
B_F_R1_P11_Path;  
B_L_R4_P7_Path;  
F_L_R1_P2_Path;  
F_F_R1_P11_Path;  
B_F_R2_P7_Path;  
B_L_R4_P8_Path;  
F_L_R1_P3_Path;  
F_F_R2_P7_Path;  
B_F_R3_P10_Path;  
B_L_R1_P11_Path;  
F_L_R2_P1_Path;  
F_F_R3_P10_Path;  
pokusajI := TRUE;  
ELSEIF slovoZ = TRUE and pokusajZ = FALSE THEN
```

```
B_F_R1_P9_Path;
B_L_R2_P7_Path;
F_R_R1_P12_Path;
F_F_R1_P9_Path;
pokusajZ := TRUE;
ELSEIF slovoR = TRUE and pokusajR = FALSE THEN
  B_F_R3_P5_Path;
  B_R_R1_P7_Path;
  F_R_R3_P4_Path;
  F_F_R3_P5_Path;
  pokusajR := TRUE;
ELSEIF slovoO = TRUE and pokusajO = FALSE THEN
  B_F_R3_P6_Path;
  B_R_R2_P7_Path;
  F_R_R3_P1_Path;
  F_F_R3_P6_Path;
  B_F_R3_P8_Path;
  B_R_R4_P7_Path;
  F_R_R3_P2_Path;
  F_F_R3_P8_Path;
  pokusajO := TRUE;
ELSEIF slovoB = TRUE and pokusajB = FALSE THEN
  B_F_R3_P7_Path;
  B_R_R3_P7_Path;
  F_L_R1_P4_Path;
  F_F_R3_P7_Path;
  pokusajB := TRUE;
ELSEIF slovoT = TRUE and pokusajT = FALSE THEN
  B_F_R3_P9_Path;
  B_R_R4_P8_Path;
  F_R_R4_P4_Path;
  F_F_R3_P9_Path;
  pokusajT := TRUE;

ELSE
  Vjesala;
  IF greska = 6 THEN
    IF pokusajM = TRUE THEN
      B_F_R1_P3_Path;
      B_R_R1_P8_Path;
      B_F_R1_P10_Path;
      B_R_R1_P9_Path;
    ELSE
      B_F_R1_P3_Path;
      B_L_R2_P3_Path;
      B_F_R1_P10_Path;
      B_L_R3_P7_Path;
    ENDIF
  ENDIF
```



```
IF pokusajE = TRUE THEN
  B_F_R1_P4_Path;
  B_L_R3_P1_Path;
ELSE
  B_F_R1_P4_Path;
  B_L_R4_P3_Path;
ENDIF
IF pokusajH = TRUE THEN
  B_F_R1_P5_Path;
  B_L_R3_P8_Path;
ELSE
  B_F_R1_P5_Path;
  B_R_R2_P3_Path;
ENDIF
IF pokusajA = TRUE THEN
  B_F_R1_P6_Path;
  B_L_R5_P1_Path;
ELSE
  B_F_R1_P6_Path;
  B_R_R4_P3_Path;
ENDIF
IF pokusajN = TRUE THEN
  B_F_R1_P7_Path;
  B_R_R2_P4_Path;
ELSE
  B_F_R1_P7_Path;
  B_R_R5_P3_Path;
ENDIF
IF pokusajI = TRUE THEN
  B_F_R1_P8_Path;
  B_L_R1_P1_Path;
  B_F_R1_P11_Path;
  B_L_R1_P2_Path;
  B_F_R2_P7_Path;
  B_L_R1_P3_Path;
  B_F_R3_P10_Path;
  B_L_R2_P1_Path;
ELSE
  B_F_R1_P8_Path;
  B_L_R1_P7_Path;
  B_F_R1_P11_Path;
  B_L_R4_P7_Path;
  B_F_R2_P7_Path;
  B_L_R4_P8_Path;
  B_F_R3_P10_Path;
  B_L_R1_P11_Path;
ENDIF
IF pokusajZ = TRUE THEN
```

```
        B_F_R1_P9_Path;
        B_R_R1_P12_Path;
    ELSE
        B_F_R1_P9_Path;
        B_L_R2_P7_Path;
    ENDIF
    IF pokusajR = TRUE THEN
        B_F_R3_P5_Path;
        B_R_R3_P4_Path;
    ELSE
        B_F_R3_P5_Path;
        B_R_R1_P7_Path;
    ENDIF
    IF pokusajO = TRUE THEN
        B_F_R3_P6_Path;
        B_R_R3_P1_Path;
        B_F_R3_P8_Path;
        B_R_R3_P2_Path;
    ELSE
        B_F_R3_P6_Path;
        B_R_R2_P7_Path;
        B_F_R3_P8_Path;
        B_R_R4_P7_Path;
    ENDIF
    IF pokusajB = TRUE THEN
        B_F_R3_P7_Path;
        B_L_R1_P4_Path;
    ELSE
        B_F_R3_P7_Path;
        B_R_R3_P7_Path;
    ENDIF
    IF pokusajT = TRUE THEN
        B_F_R3_P9_Path;
        B_R_R4_P4_Path;
    ELSE
        B_F_R3_P9_Path;
        B_R_R4_P8_Path;
    ENDIF

    Vjesala_return;
    Reset_vrijednosti;
    igra3 := FALSE;
    igra4 := TRUE;

    ENDIF
ENDIF
```

```
IF greska < 6 AND pokusajM = TRUE AND pokusajE = TRUE AND pokusajH =  
TRUE AND pokusajA AND pokusajN = TRUE AND pokusajI = TRUE AND pokusajZ =  
TRUE AND pokusajR AND pokusajO = TRUE AND pokusajB = TRUE AND pokusajT =  
TRUE THEN
```

```
    F_Stickman_head_happy;
```

```
    Vjesala_return;
```

```
    B_F_R1_P3_Path;
```

```
    B_R_R1_P8_Path;
```

```
    B_F_R1_P4_Path;
```

```
    B_L_R3_P1_Path;
```

```
    B_F_R1_P5_Path;
```

```
    B_L_R3_P8_Path;
```

```
    B_F_R1_P6_Path;
```

```
    B_L_R5_P1_Path;
```

```
    B_F_R1_P7_Path;
```

```
    B_R_R2_P4_Path;
```

```
    B_F_R1_P8_Path;
```

```
    B_L_R1_P1_Path;
```

```
    B_F_R1_P9_Path;
```

```
    B_R_R1_P12_Path;
```

```
    B_F_R1_P10_Path;
```

```
    B_R_R1_P9_Path;
```

```
    B_F_R1_P11_Path;
```

```
    B_L_R1_P2_Path;
```

```
    B_F_R2_P7_Path;
```

```
    B_L_R1_P3_Path;
```

```
    B_F_R3_P5_Path;
```

```
    B_R_R3_P4_Path;
```

```
    B_F_R3_P6_Path;
```

```
    B_R_R3_P1_Path;
```

```
    B_F_R3_P7_Path;
```

```
    B_L_R1_P4_Path;
```

```
    B_F_R3_P8_Path;
```

```
    B_R_R3_P2_Path;
```

```
    B_F_R3_P9_Path;
```

```
    B_R_R4_P4_Path;
```

```
    B_F_R3_P10_Path;
```

```
    B_L_R2_P1_Path;
```

```
    B_Stickman_head_happy;
```

```
    Reset_vrijednosti;
```

```
    igra3 := FALSE;
```

```
    igra4 := TRUE;
```

```
ENDIF
```

```
ENDIF
```

```
ENDPROC
```

```
PROC Vjesala_igra4()
!STROJARSTVO
IF ispis = FALSE THEN
  F_L_R2_P3_Path;
  F_F_R2_P2_Path;
  F_L_R4_P3_Path;
  F_F_R2_P3_Path;
  F_R_R2_P3_Path;
  F_F_R2_P4_Path;
  F_R_R4_P3_Path;
  F_F_R2_P5_Path;
  F_R_R5_P3_Path;
  F_F_R2_P6_Path;
  F_L_R1_P7_Path;
  F_F_R2_P7_Path;
  F_L_R2_P7_Path;
  F_F_R2_P8_Path;
  F_L_R3_P7_Path;
  F_F_R2_P9_Path;
  F_L_R4_P7_Path;
  F_F_R2_P10_Path;
  F_L_R4_P8_Path;
  F_F_R2_P11_Path;
  F_R_R1_P7_Path;
  F_F_R2_P12_Path;

  ispis := TRUE;

ELSEIF ispis = TRUE THEN

  IF slovoS = TRUE and pokusajS = FALSE THEN
    B_F_R2_P2_Path;
    B_L_R2_P3_Path;
    F_R_R3_P8_Path;
    F_F_R2_P2_Path;
    B_F_R2_P9_Path;
    B_L_R3_P7_Path;
    F_R_R3_P9_Path;
    F_F_R2_P9_Path;
    pokusajS := TRUE;
  ELSEIF slovoT = TRUE and pokusajT = FALSE THEN
    B_F_R2_P3_Path;
    B_L_R4_P3_Path;
    F_R_R4_P4_Path;
    F_F_R2_P3_Path;
    B_F_R2_P10_Path;
    B_L_R4_P7_Path;
    F_R_R4_P5_Path;
```

```
F_F_R2_P10_Path;
pokusajT := TRUE;
ELSEIF slovoR = TRUE and pokusajR = FALSE THEN
  B_F_R2_P4_Path;
  B_R_R2_P3_Path;
  F_R_R3_P4_Path;
  F_F_R2_P4_Path;
  B_F_R2_P8_Path;
  B_L_R2_P7_Path;
  F_R_R3_P5_Path;
  F_F_R2_P8_Path;
  pokusajR := TRUE;
ELSEIF slovoO = TRUE and pokusajO = FALSE THEN
  B_F_R2_P5_Path;
  B_R_R4_P3_Path;
  F_R_R3_P1_Path;
  F_F_R2_P5_Path;
  B_F_R2_P12_Path;
  B_R_R1_P7_Path;
  F_R_R3_P2_Path;
  F_F_R2_P12_Path;
  pokusajO := TRUE;
ELSEIF slovoJ = TRUE and pokusajJ = FALSE THEN
  B_F_R2_P6_Path;
  B_R_R5_P3_Path;
  F_L_R4_P4_Path;
  F_F_R2_P6_Path;
  pokusajJ := TRUE;
ELSEIF slovoA = TRUE and pokusajA = FALSE THEN
  B_F_R2_P7_Path;
  B_L_R1_P7_Path;
  F_L_R5_P1_Path;
  F_F_R2_P7_Path;
  pokusajA := TRUE;
ELSEIF slovoV = TRUE and pokusajV = FALSE THEN
  B_F_R2_P11_Path;
  B_L_R4_P8_Path;
  F_R_R5_P4_Path;
  F_F_R2_P11_Path;
  pokusajV := TRUE;

ELSE
  Vjesala;

IF greska = 6 THEN
  IF pokusajS = TRUE THEN
    B_F_R2_P2_Path;
    B_R_R3_P8_Path;
```

```
B_F_R2_P9_Path;  
B_R_R3_P9_Path;  
ELSE  
B_F_R2_P2_Path;  
B_L_R2_P3_Path;  
B_F_R2_P9_Path;  
B_L_R3_P7_Path;  
ENDIF  
IF pokusajT = TRUE THEN  
B_F_R2_P3_Path;  
B_R_R4_P4_Path;  
B_F_R2_P10_Path;  
B_R_R4_P5_Path;  
ELSE  
B_F_R2_P3_Path;  
B_L_R4_P3_Path;  
B_F_R2_P10_Path;  
B_L_R4_P7_Path;  
ENDIF  
IF pokusajR = TRUE THEN  
B_F_R2_P4_Path;  
B_R_R3_P4_Path;  
B_F_R2_P8_Path;  
B_R_R3_P5_Path;  
ELSE  
B_F_R2_P4_Path;  
B_R_R2_P3_Path;  
B_F_R2_P8_Path;  
B_L_R2_P7_Path;  
ENDIF  
IF pokusajO = TRUE THEN  
B_F_R2_P5_Path;  
B_R_R3_P1_Path;  
B_F_R2_P12_Path;  
B_R_R3_P2_Path;  
ELSE  
B_F_R2_P5_Path;  
B_R_R4_P3_Path;  
B_F_R2_P12_Path;  
B_R_R1_P7_Path;  
ENDIF  
IF pokusajJ = TRUE THEN  
B_F_R2_P6_Path;  
B_L_R4_P4_Path;  
ELSE  
B_F_R2_P6_Path;  
B_R_R5_P3_Path;  
ENDIF
```

```
IF pokusajA = TRUE THEN
  B_F_R2_P7_Path;
  B_L_R5_P1_Path;
ELSE
  B_F_R2_P7_Path;
  B_L_R1_P7_Path;
ENDIF
IF pokusajV = TRUE THEN
  B_F_R2_P11_Path;
  B_R_R5_P4_Path;
ELSE
  B_F_R2_P11_Path;
  B_L_R4_P8_Path;
ENDIF

Vjesala_return;
Reset_vrijednosti;
igra4 := FALSE;
igra1 := TRUE;
```

```
ENDIF
ENDIF
```

```
IF greska < 6 AND pokusajS = TRUE AND pokusajT = TRUE AND pokusajR =
TRUE AND pokusajO AND pokusajJ = TRUE AND pokusajA = TRUE AND pokusajV =
TRUE THEN
```

```
F_Stickman_head_happy;
Vjesala_return;
```

```
B_F_R2_P2_Path;
B_R_R3_P8_Path;
B_F_R2_P3_Path;
B_R_R4_P4_Path;
B_F_R2_P4_Path;
B_R_R3_P4_Path;
B_F_R2_P5_Path;
B_R_R3_P1_Path;
B_F_R2_P6_Path;
B_L_R4_P4_Path;
B_F_R2_P7_Path;
B_L_R5_P1_Path;
B_F_R2_P8_Path;
B_R_R3_P5_Path;
B_F_R2_P9_Path;
B_R_R3_P9_Path;
B_F_R2_P10_Path;
B_R_R4_P5_Path;
B_F_R2_P11_Path;
```

```
B_R_R5_P4_Path;  
B_F_R2_P12_Path;  
B_R_R3_P2_Path;  
  
B_Stickman_head_happy;  
Reset_vrijednosti;  
igra4 := FALSE;  
igra1:= TRUE;  
ENDIF  
ENDIF  
ENDPROC
```