

Razvoj robotske ćelije za lakiranje u programskome alatu V-REP

Tkalčec, Danijel

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:209837>

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

Download date / Datum preuzimanja: **2024-07-15**



Repository / Repozitorij:

[University North Digital Repository](#)





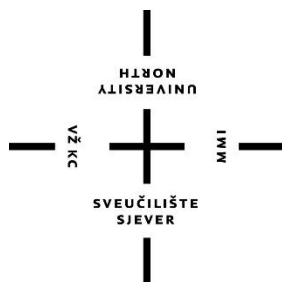
**Sveučilište
Sjever**

Završni rad br. 297/PS/2019

**Razvoj robotske ćelije za lakiranje u programskome alatu
V-REP**

Danijel Tkalčec, 4319/601

Varaždin, rujan 2019. godine



Sveučilište Sjever

Odjel za strojarstvo

Završni rad br. 297/PS/2019

Razvoj robotske ćelije za lakiranje u programskome alatu V-REP

Student

Danijel Tkalčec, 4319/601

Mentor

Tomislav Pavlic, mag.ing.mech.

Varaždin, rujan 2019. godine

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za strojarstvo		
STUDIJ	preddiplomski stručni studij Proizvodno strojarstvo		
PRISTUPNIK	Danijel Tkalčec	MATIČNI BROJ	4319/601
DATUM	03.04.2019.	KOLEGIJ	CNC obradni sustavi
NASLOV RADA	Razvoj robotske ćelije za lakiranje u programskome alatu V-REP		

NASLOV RADA NA ENGL. JEZIKU The development of robotic cell for painting in the programming tool V-REP

MENTOR	Tomislav Pavlic, mag.ing.mech.	ZVANJE	viši predavač
ČLANOVI POVJERENSTVA	1. dr. sc. Zlatko Botak, viši predavač		
	2. Tomislav Pavlic, mag.ing.mech., viši predavač		
	3. Marko Horvat, dipl.ing., predavač		
	4. prof.dr.sc. Antun Stoić		
	5. _____		

Zadatak završnog rada

BROJ	297/PS/2019
OPIS	

U završnome je radu potrebno:

- opisati virtualnu stvarnost na primjeru primjene industrijskih robota u strojarstvu
- opisati programski alat V-REP
- opisati procese i sustave koji se koriste kod površinske zaštite (bojanje, plastifikacija..)
- navesti značajke odabranog industrijskog robota
- objasniti izradu programa za bojanje robotom u programskome alatu V-REP
- izraditi i simulirati robotsku ćeliju za lakiranje sa 7 stupnjeva slobode gibanja (robot i pozicioner)

ZADATAK URUČEN 17.07.2019.



T. Pavlic

Predgovor

Zahvaljujem se mentoru Tomislavu Pavlicu, mag. ing. mech. na pruženoj mogućnosti izrade završnog rada na vrlo zanimljivoj tematici. Zahvaljujem na pomoći, savjetima, utrošenome vremenu i trudu, te pruženom znanju koje sam dobio prilikom izrade rada.

Zahvaljujem prijateljima i kolegama na zajedničkom svladavanju studijskih obaveza, te lijepom i nezaboravnom druženju. Hvala profesorima i asistentima na pruženom znanju i poticaju za daljnjim učenjem i usavršavanjem.

Također zahvaljujem se tvrtki MIV d.d. na povjerenju i ustupljenim materijalima, te svim susretljivim djelatnicima na korisnim i praktičnim savjetima.

Najveća zahvala roditeljima koji su mi svojim odricanjem omogućili studiranje, te velikom strpljenju, potpori i pomoći tokom dosadašnjeg studiranja.

Posebnu zahvalu iskazujem Lidiji na nevjerojatnoj motivaciji, potpori, nesebičnom razumijevanju i strpljenju.

Sažetak

Ovaj završni rad opisati će uvođenje robotizacije u industrijski proces za potrebe tehnološke operacije površinske zaštite.

U prvom dijelu će se objasniti procesi i sustavi površinske zaštite (bojanje, plastifikacija i sve rijeđe bitumenizacija), te priprema površina (sačmarenje) prije samog postupka zaštite. Primjeri postrojenja u realnom sektoru proizvodnje kao što su linija za plastifikaciju u fluidiziranom sloju i linija za bitumeniziranje vodenim postupkom. Približno sličnu liniju za lakiranje prikazati će se pomoću programskog alata V- REP. U programu će biti odabrani robot za bojanje KUKA koji raspršivanjem nanosi boju na radni komad kojeg će rotirati pozicioner. Stalak „Viktorija zdenac“ poslužiti će kao model radnog komada koji će se okretati dok robot na njega nanosi slojeve boje. Budući da je V- REP najnapredniji i vrlo prilagodljiv simulator, moguće ga je koristiti i u području virtualne stvarnosti (VR) za potrebe programiranja pokreta preko VR naočala i kontrolera.

Drugi dio i glavna tema ovog rada je izrađivanje i simulacija ćelije za bojanje robotom u programskome alatu V- REP. Opisati će se postavljanje kinematike robota KUKA i senzora za prihvat raspršivača te postupak umetanja modela stalka iz Solid Edge-a u V- REP i njegove putanje koja će služiti za upravljanje robotom iz skripti preko funkcija programskog jezika Lua. Pozicioner koji će okretati radni komad dok robot preko putanje vrši bojanje upravljan je također iz skripte sa svojim funkcijama i parametrima. Na kraju će se prikazati dijelovi ostalih proizvodnih procesa prije i poslije samog postupka bojanja, te zaštitni uvjeti kako za ljude tako i za okoliš.

Nakon što se izradi robotska ćelija za bojanje, simulacijom se pregledava sami postupak nanosa boje na radni komad.

Ključne riječi: bojanje, plastifikacija, robot, ćelija za bojanje, stalak, V-REP, VR, Lua

Abstract

This final paper will describe the introduction of robotization into the industrial process for the purposes of a technological surface protection operation.

The first part will explain the processes and systems of surface protection (painting, plastification and less frequently bitumenisation), and the preparation of surfaces (shot blasting) before the protection process itself. Examples of installations in the real manufacturing sector include fluidized bed plastification line and water bitumenisation line. A roughly similar painting line will be displayed using the V-REP software tool. The program will select the KUKA painting robot that will spray paint onto a workpiece that will be rotated by the positioner. The stand „Viktorija zdenac“ will serve as a workpiece model that will rotate as the robot applies layers of paint on it. Because V-REP is the most advanced and highly customizable simulator, it can also be used in virtual reality (VR) for motion programming via VR glasses and controllers.

The second part and mainly the main theme of this final paper is to design and simulate a robot painting cell in the V-REP software tool. It will describe the setting of the kinematics of the KUKA robot and the spray sensor. The process of inserting a stand model from Solid Edge into V-REP and its path will be used to control the robot from scripts via the functions of the Lua programming language. The positioner who will rotate the workpiece while the robot is painting the path is also controlled from the script with its functions and parameters. Finally, parts of other manufacturing processes will be presented before and after the painting process itself, as well as the protective conditions for both humans and the environment.

After the robotic painting cell is created, the simulation examines the process of applying the paint to the workpiece itself.

Keywords: painting, plastification, robot, painting cell, stand, V-REP, VR, Lua

Popis korištenih kratica

V- REP	Virtual robot experimentation platform Virtualna platforma za eksperimentiranje robota
VR	Virtual Reality Virtualna stvarnost
API	Application programming interface Sučelje za programiranje aplikacija
CAD	Computer Aided Design Oblikovanje s pomoću računala
CIM	Computer Integrated Manufacturing Proizvodnja integrirana računalom
AKZ	Antikorozivna zaštita
PLC	Programmable Logic Controller Programblini logički kontroler
CNC	Computer Numerical Control Računalno numeričko upravljanje
KRL	Kuka Robot Language Programski jezik Kuka robota
3D	3 Dimensional 3 Dimenzijski prikaz

Sadržaj

1.	Uvod.....	1
2.	Procesi i sustavi površinske zaštite	2
2.1.	Bojanje	5
2.2.	Plastifikacija	9
2.2.1	Primjer postrojenja za plastifikaciju u fluidiziranom sloju	12
2.3.	Bitumenizacija.....	17
2.3.1	Linija za bitumeniziranje vodenim postupkom (uranjanjem)	17
3.	Programski alat V-REP	19
3.1.	Robot KUKA	21
3.2.	Model stalka	24
3.3.	Virtualna stvarnost	25
4.	Izrada programa za bojanje robotom u V-REPu.....	30
4.1.	Pokretanje programskog alata V-REP.....	31
4.2.	Dodavanje robota Kuka KR-16.....	32
4.2.1	Postavljanje inverzne kinematike robota.....	33
4.2.2	Postavljanje senzora i pištolja za boju.....	36
4.3.	Ubacivanje modela stalka iz SolidEdge-a u V-REP	38
4.4.	Izrada putanje za model stalka	40
4.5.	Upravljanje skriptom pomoću Lua funkcija.....	42
4.6.	Postavljanje rotacijskog pozicionera	46
4.7.	Prikazi dijelova postrojenja za bojanje.....	51
4.7.1	Vješalica radnih komada	51
4.7.2	Sačmara za čišćenje površine radnog komada	53
4.7.3	Sušara obojenih komada.....	54
4.7.4	Prikaz filtera	56
4.7.5	Sigurnosna ograda linije za bojanje.....	58
4.8.	Simulacija ćelije za bojanje robotom	59
5.	Zaključak.....	62
6.	Literatura.....	64
7.	Popis slika	66
8.	Prilozi.....	69

1. Uvod

Razvojem tehnologije u novije vrijeme računala upravljaju i povezuju sve dijelove proizvodnog procesa. Pored kompjuterizacije vrlo je važna i robotizacija koja predstavlja spoj strojeva i računalne tehnike. Nagli tehnološki napredak doveo je do sve šire primjene robotike u gotovo sva područja ljudskog djelovanja kao što su informatika, elektronika, mehanika i automatika, čime su postavljeni temelji za razvoj suvremenih robota. Kompjuterizirane tehnologije obuhvaćaju računalno poduprtu konstrukciju (CAD), numeričke upravljane strojeve, fleksibilne proizvodne sustave, industrijske robote, računalom integrirana proizvodnja (CIM) te primjenu umjetne inteligencije.

Industrijski roboti pojavili su se onog trenutka kada je tehničko- tehnološki razvoj dostigao potrebnu razinu radi želje za oslobađanjem čovjeka od nehumanog, monotonog i opasnog fizičkog rada i nastojanja za povećanjem profita. Upotrebom robota u industriji, smanjuje se ljudski faktor greške i vrijeme operacija se svodi na minimum, te su pouzdaniji i puno brži od čovjeka. Vrlo brzo se pokazalo da se klasični oblici proizvodnje nisu mogli „natjecati“ sa modernim kompjutorskim i robotiziranim sustavima proizvodnje [1].

Danas u industriji postoji veliki broj primjena gdje se roboti mogu efikasno upotrijebiti, od transporta i manipulacije, pa sve do procesnih operacija poput brušenja, glodanja, bušenja, zavarivanja, bojanja itd.

Uvođenjem robota za bojanje smanjuje se izlaganje ljudi otrovnim supstancama. Zbog posebnog dizajna, roboti koji se koriste za bojanje otporni su na teške uvjete i ne predstavljaju opasnost u blizini zapaljivih tvari ili u eksplozivnoj atmosferi. Robotski sustavi za bojenje su pogodno rješenje čak i za situacije u kojima je potrebno oponašati ljudsku ruku, jer ih je moguće vrlo precizno programirati za takve zadatke. Korištenjem robota dolazi do smanjenja potrebe ljudskog fizičkog rada, ali naravno tu dolazi do potrebe usavršavanja i edukacija ljudi za programiranje [2].

Svrha ovog rada jest da se поближе objasne procesi i sustavi površinske zaštite bojanja, plastifikacije i bitumenizacije u realnom proizvodnom sektoru. Pored linije za bitumenizaciju i linije za plastifikaciju robotom potreba za uvođenjem linije za lakiranje robotom u budućnosti je neizbježna. Razvoj i funkcioniranje ćelije za lakiranje robotom prikazati će se u programskome alatu V-REP. Simulirati će se robotska ćelija za lakiranje sa 7 stupnjeva slobode gibanja koje izvode robot i pozicioner.

2. Procesi i sustavi površinske zaštite

Štete koje nam nanosi korozija vrlo su velike pa se neprekidno traže načini kako da se ona suzbije. Budući da su uzroci korozije različiti, primjenjuju se i raznovrsni načini obrane. Razvijanje korozije može se spriječiti na način da se metalne površine izoliraju od tvari koje izazivaju koroziju. U tu svrhu primjenjuju se prevlake koje onemogućavaju zraku, vodi, morskoj vodi, plinovima, kiselinama i drugim kemikalijama da dođu u dodir s metalnim površinama.

Zaštitnim prevlakama poklanja se osobita važnost jer se pomoću njih povećava trajnost i proširuje mogućnost primjene metala neotpornih prema koroziji. Pored toga, zaštitnim se prevlakama uljepšava izgled metalnih dijelova.

Koji će se materijal odabrati za prevlake, ovisi o:

- materijalu koji treba zaštititi od korozije;
- raspoloživim tehničkim sredstvima;
- vremenskoj trajnosti koja se želi postići zaštitom;
- ekonomičnosti zaštite od korozije [3].

Metal ili legura	Maksimalna temperatura upotrebe u vrućem zraku, °C	Brzina korozije, mm/god.					Specijalni oblici razaranja
		u gradskoj atmosferi	u morskoj vodi	u H ₂ SO ₄ (5%)	u HNO ₃ (5%)	u NaOH (5%)	
Ugljični čelik	570	0,03...0,2	0,003...0,3	0,4...10	vrlo velika	do 0,005	kaustična krtost
Sivi lijev	570	0,01...0,1	0,2...2	1,3	3	0,05...0,5	grafitizacija, »rast«
Ferosilicij sa 14% Si	650	do 0,005	do 0,08	do 0,13	do 0,5	do 0,3	
Kromni čelik sa 13% Cr	795	do 0,01	do 0,1	vrlo velika	0,08...0,9	do 0,1	točkasta, interkristalna i napetostna korozija
Krom-nikleni čelik sa 19% Cr i 10% Ni	865	do 0,005	do 0,05	0,1...3	do 0,05	do 0,05	
Krom	1000	do 0,01	do 0,1	0,6...3	oko 0,3	do 0,1	
Nikal	800	do 0,05	0,05...0,5	0,05...1	1...10	do 0,005	
Bakar	700	do 0,05	0,05...0,5	0,08...1	vrlo velika	0,05...0,5	
Mjed (mesing) sa 30...42% Zn	750	do 0,03	0,05...0,5	0,05...0,5	vrlo velika	0,05...0,5	decinkacija, sezonski lom
Cink	400	do 0,01	0,01...0,3	vrlo velika	vrlo velika	0,4...5	
Kadmij	280	do 0,01	do 0,01	vrlo velika	vrlo velika	do 0,1	
Olovo	300	do 0,005	0,005...0,4	do 0,05	2,5...15	0,15...15	
Kositar	200	do 0,005	do 0,003	0,05...15	2,5...10	0,13...1	»kuga«, amalgamiranje
Aluminij	600	do 0,013	0,03...1,3	0,2...2,5	0,4...2	vrlo velika	
Magnezijeve legure	200	0,02...0,05	vrlo velika	vrlo velika	vrlo velika	0,03...0,3	
Titan	750	0	0	0,25...2,5	0,003...0,03	do 0,005	
Srebro	960	0	0	do 0,03	vrlo velika	0	amalgamiranje
Zlato	1063	0	0	0	0	0	amalgamiranje
Platina	1769	0	0	0	0	0	krtost
Rodij	1960	0	0	0	0	0	

Slika 2.1 Korozijsko ponašanje nekih metala [4]

Postoji mnogo materijala koji se upotrebljavaju za zaštitne prevlake, ali se gotovo sve prevlake mogu razvrstati u metalne, oksidne i nemetalne prevlake.

Zaštita od korozije organskim nemetalnim prevlakama najzastupljenija je i s njom se ljudi najviše susreću u životu. Konkretno misli se na tekuće premaze koji se svojom jednostavnošću već desetljećima nameću kao najčešće odabrano zaštitno sredstvo. Cijena i lakoća upotrebe samo su neke od njihovih prednosti.

Prevlake se dobiju nanošenjem organskih premaznih sredstava (bojenjem i lakiranjem), bitumenizacijom i plastifikacijom. Pritom, učinkovitost zaštite ne ovisi isključivo samo o svojstvima prevlake već i o karakteru same metalne osnove, pripremi površine i tehnologiji nanošenja prevlake. U praksi je stoga neophodno uzeti u obzir cijeli sustav i tehnologiju izvođenja zaštite uključujući i najčešće kompliciran sastav same prevlake.

Premazi su tekući ili praškasti proizvodi u kojima su sadržani pigmenti, a nanese u vrlo tankim slojevima na podlogu prelaze u tvrd, dobro prijanjajući film, koji služi u zaštitne i/ili dekorativne svrhe [4].

Osnovni cilj nanošenja premaza je razdvajanje metalne podloge i okoliša pa stoga oni moraju biti dovoljno postojani i trajni u uvjetima eksploatacije. Tako je i najvažnije tehničko svojstvo premaza njihova trajnost.

Sustav zaštite premazima sastoji se obično od temeljnog sloja, jednog ili više međuslojnih slojeva te završnog sloja, od kojih svaki ima svoju ulogu. Općenito se može reći da je premaz materijal koji je nakon nanošenja na podlogu stvorio čvrsti film.

Materijali za prevlačenje u kontaktu s pitkom vodom ne smiju otpuštati nikakve tvari koje bi utjecale na okus, miris ili boju vode za piće ili na rezultat rasta mikroorganizama.

Površine koje se prevlače organskim prevlakama moraju biti očišćene od nečistoća, ulja, masti, vlage, korozije ili drugih onečišćujućih tvari sačmarenjem.

Priprema pozicija za postupak sačmarenja izvodi se odgovarajućom zaštitom toleriranih dosjeda i navojnih provrta. Za zaštitu površina upotrijebljava se guma ili silikonski zaštitni čepovi, samoljepljive trake za sačmarenje ili čelične zaštitne naprave i limove.

Odljevci koji su sačmareni smiju se premještati jedino uz upotrebu rukavica koje ne otpuštaju vlakna, te ih treba što prije prenjeti u peć. Ako je period čekanja dulji, osaćmarene odljevke treba zaštititi prekrivanjem i skladištiti na suhom [5].



Slika 2.2 Jedinica za sačmarenje odljevaka [5]

Zahtjevani stupanj očišćenosti površine je Sa 2½ ili veći prema ISO 8501-1 čeličnom lomljenom sačmom, uz hrapavost sačmarene površine 40-75 µm za površine koje se prevlače bojom, odnosno minimalno 60 µm za površine koje se prevlače prahom [5].

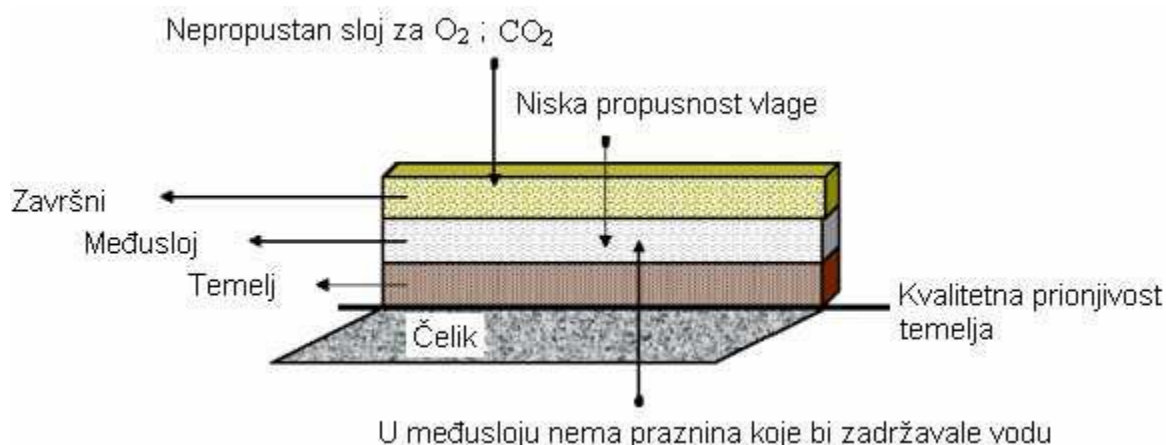


Slika 2.3 Izgled površine čistoće Sa 2½ [5]

2.1. Bojanje

Bojanje je vrlo poznat način za zaštitu metalnih površina od korozije. Boja se nanosi na čistu osušenu površinu bezračnim raspršivačem (airless), zračnom špricom, valjkom, kistom, prelijevanjem boje po površini ili uranjanjem pozicije u smjesu boje. Na željezni komad kad je pripremljen nanose se boje u više slojeva. Osnovni temeljni sloj, koji se nanosi prvi, treba dobro prijanjati za površinu i sam ne smije uzrokovati koroziju, a antikorozijska svojstva premaza sadržana su u inhibirajućim pigmentima. Međusloj u sistemu pojačava zaštitna svojstva i prijonjivost između dva sloja, te barijerni efekt (pigmentacija laminarne strukture). Gornji završni sloj, određuje konačnu boju, nijansu, sjaj, površinsku otpornost i elastičnost na djelovanje atmosfere i tekućine, te ostala posebna svojstva [3, 4].

Ovisno o temperaturi okoline (zimsko/ljetno razdoblje), prije nanošenja boje pozicije treba zagrijati unutar komore za bojenje. Nepovoljna temperatura površine u odnosu na uvjete u komori uzrokuje rošenje površine (kondenzacija vlage iz zraka) i narušava kvalitetu zaštitnog premaza [6].



Slika 2.4 Prikaz pojedinih slojeva premaza [7]

Svi premazi imaju i svoje osnovne komponente:

- **Vezivo** je smjesa prirodnih ili sintetičkih smola i drugih kemijskih komponenata. Neisparljivi je dio medija boje, povezuje sve komponente premaznog sredstva i stvaraju tvrdi zaštitni sloj. (Sušiva ulja, smole, derivati celuloze...).
- **Otapalo ili razrjeđivač** su obično hlapljivi organski spojevi u kojima se otapa vezivo i regulira viskoznost premaza, te omogućavaju postizavanje točne debljine suhog filma. Ne dolazi do kemijskih promjena. (Ugljikovodici, alkoholi, esteri, ketoni...).

- **Pigment** je sastojak premaza koji daje pokrivnu moć, pojačava film premaza, daje mu boju, utječe na postojanost filma i štiti od UV zračenja.
- **Punilo** su bijele ili slabo obojene, netopljive anorganske tvari. Poboljšavaju mehanička svojstva, mazivost, svojstva tečenja. Reguliranje sjaja i poboljšanje barijernih svojstava omogućavajući nanošenje debljih premaza.
- **Aditivi** imaju funkciju kontroliranja sušenja premaza. Mali udio u premazu, značajan utjecaj na svojstva. Koriste se za spriječavanje nedostataka u premazima (pjena, sedimentacija...) ili za davanje specifičnih svojstava (klizavost, svjetlostabilnost...) [4].

Uvjeti prilikom bojanja ovise o postupku planiranja svih aktivnosti, odgovarajućim radnim uvjetima, ispravnom odabiru metode, povoljnim mikroklimatskim uvjetima te vještini radnika. Izbor materijala za AKZ (Antikorozivnu zaštitu) je definiran radnim nalogom s točno specificiranim tipovima premaza, katkad su definirani različiti sustavi bojenja za vanjske i unutarnje površine proizvoda. Procesu su razrađeni od pripreme i zaštite površine, pri čemu treba uzeti vrijeme potrebno za montažu i demontažu skele, sušenje, otvrdnjavanje i za međupremazne intervale, te mikroklimatske uvjete (temperaturu podloge, rosište, okolnu temperaturu, relativnu vlažnost i temperaturu boje).

U zatvorenom ili ograničenom prostoru potrebno je instalirati ventilaciju zbog opće sigurnosti, a i isto tako i zbog sušenja premaza. Nadalje, premaz se nanosi kod relativne vlažnosti manje od 85%, jer previsoka relativna vlažnost može uzrokovati rošenje površina. Temperatura boje je također važna, jer kod niskih temperatura neke se dvokomponentne boje neće stvrdnuti, zato što kemijska reakcija između veziva i otvrdnjivača gotovo prestaje. Također, kod opadanja temperature, viskoznost boje raste, što rezultira slabom raspršljivošću boje tj. boja se teško nanosi [6, 8].

Prije nanošenja boje površina mora biti čista i suha! Bojiti je potrebno za suha vremena, kada su temperature iznad 5°C, izbjegavajući direktan utjecaj sunčevih zraka. Kako bi se izbjegla kondenzacija, temperatura površine mora biti iznad točke rošenja. To se može provjeriti na vrlo jednostavan način: krpom navlažite površinu, pa ako se ona osuši nakon 10 do 15 minuta, možete započeti s bojenjem. Kod dvokomponentnih je premaza poželjno da mješavina odstoji desetak minuta prije upotrebe. Jednokomponentni se premazi isporučuju već spremni za nanošenje kistom, valjkom ili raspršivačem. Ako se lakirana ili obojena površina ne osuši prije nego se poveća stupanj vlage u zraku, ili prije nego padne rosa primjerice, ona može poprimiti mat izgled, što znači da je pažljivo planiranje nanošenja pojedinih premaza od velikog značaja za postizanje dobrog rezultata.

Nakon što se pažljivo pripremi površina slijedi nanošenje premaza. Boja se ne nanosi pri visokoj vlazi niti na temperaturama ispod 10°C kod dvokomponentnih premaza ili 5°C kod jednokomponentnih boja. Najbolji je način nanošenja temeljnih boja, a da se pritom izbjegne nastajanje rupica, pomoću kista ili raspršivača. Međupremazi i završni premazi mogu se nanositi kistom, valjkom i, dakako, raspršivačem [9].



Slika 2.5 Postupak bojenja ručnim raspršivačem [6]

Debljina sloja boje (mokri postupak) mora biti izmjerena na svim bojanim pozicijama koje su obojane završnom bojom. Mjerenje se vrši neposredno prije premještanja pozicije izvan prostora za lakiranje. Ukoliko izmjerena debljina površinske zaštite ne odgovara specifikaciji, površinska zaštita se dorađuje.

Kod svakodnevnog rada sa premazima ponekad dolazi do grešaka koje nisu posljedica njihove loše kvalitete, već često nastaju zbog nedovoljnog poznavanja i nepravilne upotrebe materijala, loših uvjeta aplikacije i nepravilne tehnike nanošenja.

Vrlo je važno znati analizirati takve greške, otkriti uzrok i što je najvažnije otkloniti uzrok i popraviti greške. Pokazati će se neke najveće greške pri lakiranju, koji se susreću u praksi, definirati uzorke i dati uputstva za njihovo spriječavanje [10].

Stvaranje mjehurića



Uzroci:

- prekratko vrijeme otparavanja među slojevima
- predebeli slojevi nanesenog predlaka
- ostaci vode od brušenja u kutovima, na rubovima, pregibima i pod ukrasnim letvicama
- previsoka tvrdoća vode za brušenje
- onečišćen komprimirani zrak
- previsoka relativna vlažnost zraka (nad 80% kod 2K materijala)
- nastajanje kondenzacije zbog temperaturnih promjena

Sprječavanje:

- minimalno otparavanje među slojevima 10 min./20 °C
- kod jednokratnog nanošenja max. 80 µm za HS predlak i max. 40 µm za MS predlak
- po mogućnosti uvijek demontirati sastavne dijelove
- temeljito ispuhati
- ne dopustiti da se ostaci vode od brušenja osuše, obrisati ih dok su mokri
- komprimirani zrak mora biti suh i čist
- izbjegavati nastajanje kondenzacije, osigurati stalnu temperaturu i vlagu

Popravlak nastalog defekta:

- izbrusiti do lima i lakiranje predlaka ponoviti po uputama

Slabo prijanjanje na podlogu



Uzroci:

- slabo pripremljena podloga, tragovi masnoće, otisaka prstiju, prašine
- uporaba materijala koji nisu primjereni za određene vrste limova
- razrjeđivanje materijala neodgovarajućim (neoriginalnim) razrjeđivačem

Sprječavanje:

- površinu prije lakiranja dobro očistiti
- uvijek provjeriti koji se materijal može nanositi na određenu vrstu lima
- uporaba propisanih razrjeđivača

Popravlak nastalog defekta:

- izbrusiti do lima i lakiranje predlaka ponoviti po uputama
- paziti na pravilan izbor 2K predlaka u odnosu na podlogu

Krateri



Uzroci:

- ostaci voska, masnoće, silikona
- komprimirani zrak zagađen ostacima kondenzacije ili ulja
- upotreba silikonskih polimnih sredsteva ili sprejeva
- nedovoljno očišćena površina prije lakiranja

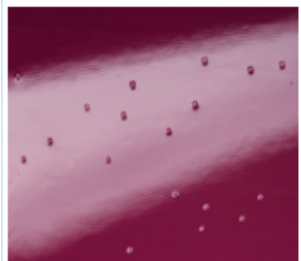
Sprječavanje:

- prije početka lakiranja temeljito očistiti MOBIHEL Antisilikonskim čistilom low VOC
- redovno održavanje filtera za izlučivanje ulja i vode

Popravlak nastalog defekta:

- izbrusiti do predlaka i lakiranje pokravnog premaza ponoviti po uputama

Zakuhavanje



Uzroci:

- nedovoljno osušen predlak
- zatvaranje otapala i zraka predebelim slojevima premaza
- kod nanošenja pojedinih slojeva nije bilo otparavanja
- upotreba neodgovarajućeg utvrđivača ili razrjeđivača
- prekinuto sušenje (nepridržavanje vremena sušenja)

Sprječavanje:

- nanijeti propisanu debljinu premaza
- međusoljni interval treba biti 10–15 min./20°C
- redovito provjeravati rad i temperaturu sušila
- pridržavati se propisanih vremena otparavanja između nanošenja pojedinih slojeva i konačnog otparavanja
- upotreba propisanog utvrđivača i razrjeđivača

Popravlak nastalog defekta:

- izbrusiti do lima i ponoviti lakiranje predlaka i pokravnog premaza po uputama

Slika 2.6 Uzroci i sprječavanje nekih grešaka pri lakiranju [10]

2.2. Plastifikacija

Zaštita okoliša značajno je utjecala i na povećanu upotrebu polimernih prevlaka koje se nanose u čvrstom stanju za zaštitu različitih konstrukcijskih materijala od korozije, obzirom da kod postupaka plastifikacije ne dolazi do isparavanja hlapljivog otapala koje značajno opterećuje okoliš.

Plastifikacija je metoda površinske zaštite metala od korozije nanošenjem praha, te polimerizaciji praha na određenoj temperaturi. Plastificiranje metala je postupak kojim se po naelektriziranom komadu metala, posipa obrnutog polariteta nabijeni prah u pištolju za elektrostatsko prašenje, te se na taj način prah lijepi i po dijelovima koji nisu na direktnom udaru praha. Elektricitet privlači prah gdje ga nema, a odbija gdje je nanesen. U daljnjem posipanju, prah se ponaša kao izolator, jer je poliuretanskog sastava, pa novi sloj praha otpada sa pokrivenog dijela, te imamo ravnomjerno raspoređen sloj koji se tijekom polimerizacije na cca 180°C pretvara u ravnomjeran sloj zaštitne plastike [11].

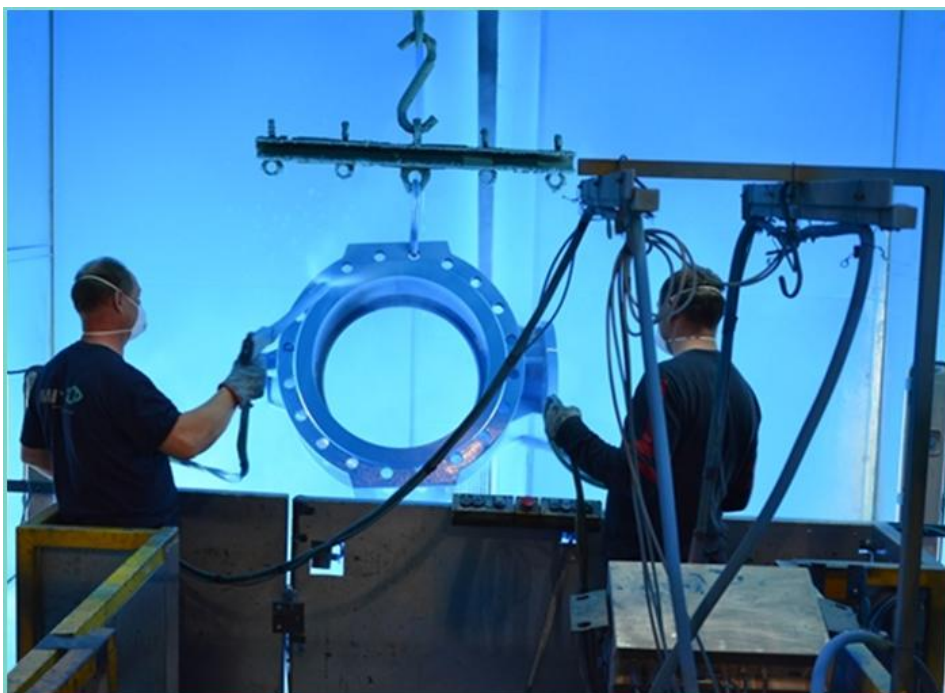


Slika 2.7 Prah za plastifikaciju [12]

Plastificirani sloj, osim šta je odlična zaštita od korozije, otporan je na toplinu i abraziju. Plastificiranje zadovoljava suvremene ekološke standarde, te je za okolinu i zdravlje najprihvatljivija metoda oplemenjivanja metala. Osim navedenih svojstava, plastifikacija omogućava i neograničen izbor boja, različite površinske strukture.

Postupcima plastifikacije se na različite metalne i nemetalne proizvode nanose prevlake od plastičnih masa, najčešće većih debljina (od 0,2 do 0,5 mm). Iako su po svom sastavu ove prevlake vrlo slične premazima s poliplastičnim vezivima, uvelike se razlikuju od njih po tehnologiji nanošenja.

Za nanošenje polimernih prevlaka razvijen je veliki broj postupaka, od kojih su najvažniji: lijepljenje i oblaganje, vruće prskanje, nataljivanje, fluidizacija i elektrostatičko naprašivanje. Od nabrojanih postupaka nanošenja polimernih prevlaka najveći tehnički značaj imaju postupci elektrostatskog naprašivanja i fluidizacije [8, 13].



Slika 2.8 Ručna plastifikacija epoxy prahom [13]

Elektrostatičko naprašivanje vrlo često naziva se i suhim ličenjem, stoga što se kod ovog postupka plastifikacije metalnih podloga ne primjenjuje razrjeđivač niti otapalo, već se nanosi suhi polimerni prašak [8].





Slika 2.9 Predgrijavanje komada i komora za fluidizaciju [13]

Fluidizacija je postupak nanošenja polimerne prevlake zagrijani predmet „uranja“ se u uskovitlani sloj poliplasta koji nastaje propuhivanjem zraka ili dušika kroz porozno dno posebnog spremnika. U pravilu, predmeti se zagrijavaju na 100 do 200 °C iznad tališta poliplasta. Ovisno o debljini podloge, kao i vrsti praha koji se nanosi, temperature predgrijavanja kreću se od 180 do 400 °C, te su znatno više nego kod elektrostatskog naprašivanja. Ovisno o toplinskom kapacitetu obrađivanog predmeta, tj. podloge, kao i o temperaturi predgrijavanja i vremenu uranjanja u sloj uskovitlanog polimernog praha stvara se prevlaka taljenjem sinteriranog polimera izvan samog spremnika. Uobičajene debljine prevlaka koje se mogu postići jednim uranjanjem kreću se od 200 do 500 μm , a u posebnim slučajevima moguće je postići prevlake debljine i preko 1 mm [8].

Na pozicije predviđene za dvoslojno oblaganje prahom, nanosi se završni sloj poliesterskog praha (50 μm). Postupak se obavlja u istom ciklusu, odmah po završetku nanošenja prvoga sloja obloge.

Prevlaka od epoksi praha na gotovom proizvodu mora ispuniti sljedeće zahtjeve:

- biti jednolike boje
- biti ujednačenog izgleda i glatkoće, izuzev dopuštenih područja na kojima su obavljani popravci
- ne smije na njoj biti vidljivih grešaka (rupice, mjehuri, nabori, pukotine ili šupljine ispunjene zrakom)

Manje promjene boje na površini nastale kao posljedica popravka ili izlaganja sunčevoj svjetlosti tijekom duljih perioda (izbjeljivanje) ili razlike u razinama sjaja radi prejakog prskanja su dopuštene. (Izbjeljivanje je reakcija koja se pojavi na površinama prevučenima epoksidom izloženima ultraljubičastom zračenju. Proces je uravnotežen na dubini 2-5 µm proizvodima reakcije i vezan je uz gubitak sjaja. Nema utjecaja na zaštitne karakteristike prevlake).

Proizvodi osigurane kvalitete procesa su označeni oznakom kvalitete - Udruženja za osiguranje kvalitete (Gütegemeinschaft) antikorozijske zaštite armatura i fazona u teškim uvjetima korozije prevlačenjem epoxy prahom, uz informacije koje se odnose na kvalitetu u skladu sa sljedećim simbolom oznake kvalitete RAL – GSK [13].



Slika 2.10 Oznaka kvalitete antikorozijske zaštite [13]

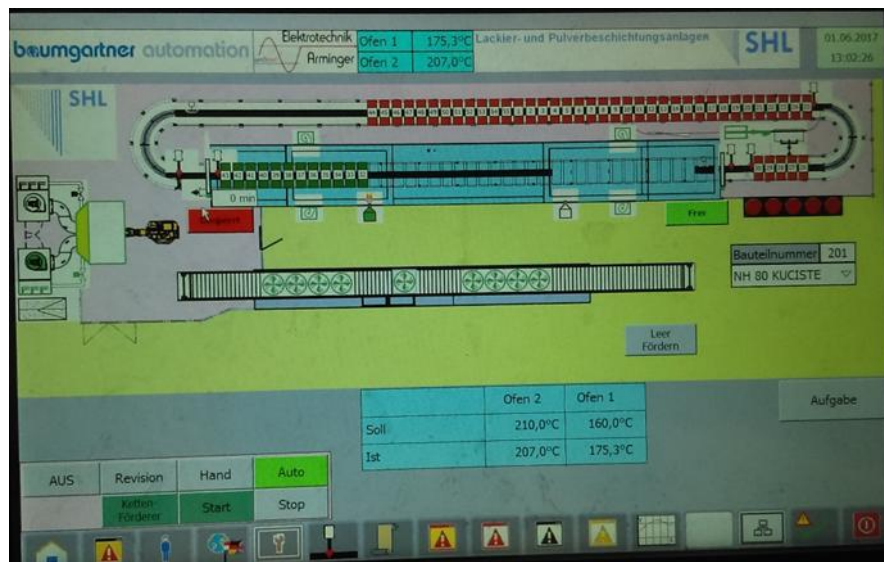
2.2.1 Primjer postrojenja za plastifikaciju u fluidiziranom sloju

Postrojenje za potpuno automatiziranu plastifikaciju odljevaka u fluidiziranom sloju pomoću robota KUKA KR 240. Prije plastifikacije, potrebno je pripremiti površinu odljevaka u komori za sačmarenje. Odljevci se postavljaju na pokretnu traku ručno ili uz pomoć dizalice ovisno o težini. Pomoću sučelja se unosi kod, primjerice 6- znamenkasta brojevana oznaka, koja točno označava pojedinu vrstu odljevka. Preko te oznake proizvoda, robot dohvaća sve informacije važne za proces plastifikacije:

- Vrsta steznih naprava za prihvatanje odljevka
- Pokrete robota tijekom postupka prihvaćanja, plastifikacije i odlaganja odljevka na traku
- Vrijeme stanke nakon plastifikacije pojedinog proizvoda

Istovremeno je na zaslonu prikazana točna pozicija odljevka smještenog na pokretnoj traci.

Upravljanje se vrši pomoću Siemens Simatic S7-1513F SPS (njem. Speicherprogrammierbare Steuerung - programibilni logički kontroler). Operacija i ulaz se odvija preko Siemens TP1500 dodirnog zaslona instaliranog na upravljačku konzolu s postoljem [14].



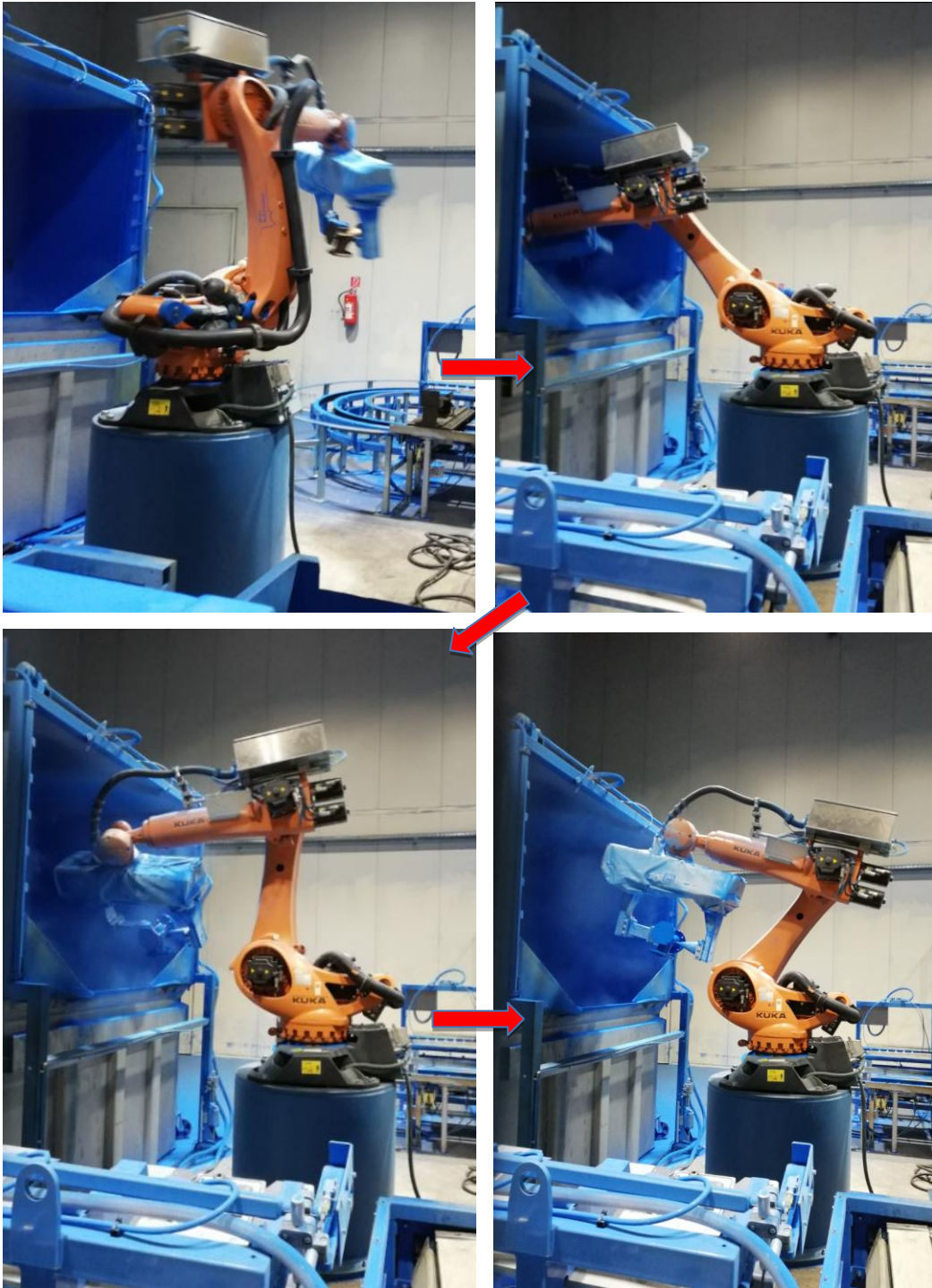
Slika 2.11 Zaslona upravljačke jedinice peći [13]

Nakon postavljanja na pokretnu traku, odljevci prolaze kroz peć za predgrijavanje. Peć je podijeljena u dva dijela s jednakim uvjetima zagrijavanja i dostupnim mogućnostima regulacije. U slučaju zagrijavanja težih odljevaka, zadana temperatura u prvoj polovici peći može biti primjerice 250 °C, dok u drugoj polovici može biti niža (220 °C). Na taj se način prema potrebi može ubrzati proces predgrijavanja [14].



Slika 2.12 Peć za predgrijavanje odljevaka [13]

Nakon grijanja, robot opremljen prikladnim steznim napravama automatski izvodi postupak plastifikacije. Ovisno o geometriji odljevka, robot sa stanice za zamjenu steznih naprava dohvaća određenu steznu napravu kojom će prihvatiti odljevak. Slijed pokreta robota tijekom samog postupka plastifikacije se može mijenjati po potrebi, tako da se dobije ravnomjerno oblaganje s mogućnošću ponavljanja istih radnji. Po potrebi, ili za plastifikaciju nove vrste odljevka, slijed radnji se može promijeniti.



Slika 2.13 Plastifikacija robotom KUKA KR 240 [13]

Fluidizacija praha u bazenu se postiže pomoću bočnih kanala za upuhivanje komprimiranog zraka. Usisani zrak iz prostorije prolazi kroz filtar i upuhuje se u bazen kroz otvore u stijenkama bazena. Kroz sinter ploču zrak ustrojava u prahom ispunjen gornji dio bazena. Prah se fluidizira (suspendira). Iznad bazena je postavljen usisni uređaj povezan s filtrom za regeneraciju. Sustav usisava prah koji se uskovitla tijekom oblaganja, pročišćava ga i kroz injektor s crijevom vraća nazad u bazen. Razina praha u bazenu se regulira uz pomoć detektora. U slučaju pada razine, spremnik se dopunjava preko stanice za doziranje svježeg praha do postizanja željene razine.

Postrojenje je opremljeno s dva usisna filtra. Za prah standardne boje (plava RAL 5015) je osiguran odvojeni filtar s automatskim povratom praha. Drugi filtar se koristi za prah drugih boja. Zbog neupotrebljivosti mješavine praha različitih boja, isti nije povezan s regeneracijskim sustavom [14].



Slika 2.14 Bazen za fluidizaciju praha [13]

Nakon plastifikacije, proizvodi se hlade prolaskom kroz rashladni tunel. Odljevci se smještaju na pokretnu traku. Zona hlađenja je opremljena s 8 aksijalnih ventilatora i ispušnim ventilatorom. Kroz aksijalne ventilatore se upuhuje zrak koji hladi odljevke. Ispušni ventilator, pričvršćen na stropu rashladnog tunela, usisava zrak iz tunela i upuhuje ga u otvoreni prostor, ili ga upuhuje u susjedni odjel proizvodnog prostora. Usisani zrak struji iz prostorije kroz dva kraja rashladnoga tunela.



Slika 2.15 Rashladni tunel [13]

Nakon što su proizvodi prošli kroz zonu hlađenja, proces plastifikacije je završen. Oštećenja prevlake na mjestima prihvata steznih naprava trebaju biti popravljena.



Slika 2.16 Oštećenje od prihvata [13]

2.3. Bitumenizacija

Pojavom plastifikacije u industrijama sve manje se koristi zaštita željeza bitumenom. Željezne cijevi za kanalizaciju, vodovod ili plinovod, koje se upotrebljavaju za izgradnju podzemnih cjevovoda i spremnika, zaštićuju se od korozije prevlačenjem bitumenom ili katranom. Cijevi se uranjaju u rastaljeni bitumen i zagrijavaju na temperaturu oko 300°C.

Materijal za premaze mora se sastojati od bitumenskog materijala ili od sintetičke smole. Odgovarajući dodaci (otapala, anorganske ispune itd.) su dozvoljeni da bi se postiglo lako nanošenje i sušenje. Prije nanošenja površina mora biti suha, bez hrđe ili slobodnih dijelića, te bez ulja ili masti. Premaz mora cijelu površinu jednoliko prikriti, mora biti gladak i homogen. Mora se dobro osušiti da se komadi ne bi međusobno slijepili. Premaz se nanaša uranjanjem, špricanjem ili kistom [3].



Slika 2.17 Bitumenizirani podzemni hidrant [15]

2.3.1 Linija za bitumeniziranje vodenim postupkom (uranjanjem)

Nakon sačmarenja do zadovoljavajuće čistoće površine, radni komadi se stavljaju na transporter – transportni lanac i ulaze u tunel na predgrijavanje. Tunel za predgrijavanje je istovremeno i tunel za sušenje, ima ugrađenu opremu za odsis i dobavu zraka. Zagrijavaju se na potrebnu temperaturu i dovode se do mjesta za uranjanje. Za nadoknadu odsisanog zraka te zagrijavanje koristi se plinski zagrijač zraka koji omogućuje potrebnu temperaturu predgrijavanja, odnosno sušenja. Za regulaciju temperature u tunelu koristi se termometar. Upravljanje procesom rada u tunelu (nadtlak/potlak, kontrola ventilacije, kontrola temperature, kontrola zapunjenosti filtra, i sl.) je preko elektrokomandnog ormara upravljanog PLC-om [15].

Predgrijani radni komadi uranjaju se u spremnik s bojom u vremenu koje je potrebno da boja prione na samu površinu. Isto se vrši na način da kada sa bojom putem hidrauličkog stola slijedi takt lanca i diže se gore dolje, dok komadi miruju na lancu. Na spremniku za uranjanje montirana je membranska pumpa i sistem cirkuliranja same boje. Isto je potrebno da se onemogući taloženje boje koja je na vodenoj osnovi i slabo je dispergirana.

Završetkom procesa uranjanja radni komad se pomiče zajedno sa lancem do kade za cijeđenje. Na komadu je boje u nanosu cca 40 – 60 microns. Nakon cijeđenja radnog komada isti se pomiče do radnog mjesta popravka nedostataka. Završetkom popravka radni komad se pomiče u tunel za odzračivanje. Tunel za odzračivanje ima ugrađenu zasebnu opremu za odsis zraka, dok se za nadoknadu zraka koristi termogen koji se ujedno koristi i za tunel za sušenje/predgrijavanje. Radni komad se ponovo predgrijava i ponovo potapa u boju. Sada na sebi ima boje u debljini 80 – 120 microns koliki je i zahtijev standarda. Ponovo slijedi tunel za odzračivanje u kojem se naglo odsisava zrak kako bi većina vode isparila prije tunela za sušenje.

Komadi se skidaju s transportne trake – lanca i odlažu u spremište (postojeća građevina) na min 24 do 78 sati kako bi boja do kraja polimerizirala i kako se ne bi skidala (vrijeme minimalnog stajanja ovisi o vanjskoj temperaturi i vlazi) [15].



Slika 2.18 Linija za bitumenizaciju uranjanjem [15]

3. Programski alat V-REP

V-REP (eng. Virtual Robot Experimentation Platform) od švicarske tvrtke Coppelia Robotics je virtualna platforma za razvoj, simulaciju i testiranje robota i rješenja vezanih uz robotiku i automatizaciju. S bogatom funkcionalnosti, dobrim grafičkim sučeljem i širokim API- jem (Sučelje za programiranje aplikacija), V-REP je možda i najnapredniji simulator takve vrste.

Alat V-REP ima otvoreni kod pa stoga postoje dvije vrste licenci:

- Komercijalna licenca
- Besplatna edukacijska licenca – namijenjena je za hobiste, studente, profesore, fakultete, učenike i škole [16, 18].



Slika 2.19 Edukacijska verzija V-REP [16]

Neke od značajke V-REP-a su:

- Podrška više operacijskih sustava (Windows, Linux)
- Više programskih pristupa (pluginovi, udaljeni API klijenti, embedded skripte, dodaci, ROS čvorovi)
- Vrlo bogat API (obični i udaljeni) s bibliotekama za više jezika (C/C++, Lua, Python, Java, Matlab...).
- Simulacija senzora blizine i vida.
- Vizualizacija i snimanje podataka. Moguća vizualizacija više simulacijskih izvedbi istovremeno za vizualnu usporedbu.
- Detekcija kolizija, računanje minimalne udaljenosti, sustav za prikaz dinamičkih čestica.
- Modularnost emulatora fizike (eng. physics engines) uključujući Bullet, ODE, Vortex.
- Planiranje puta i pokreta [16].

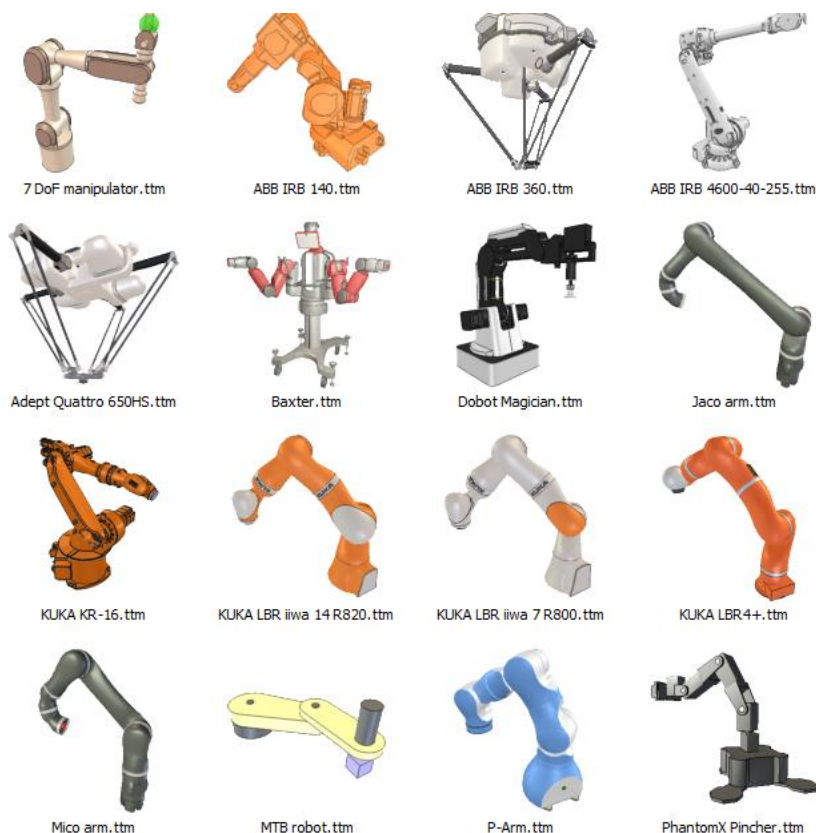
Unutar V-REP-a koristi se Lua programski jezik, ali omogućeno je povezivanje i s vanjskim aplikacijama koje koriste druge programske jezike, kao što su C++, Java, Python, Matlab, Octave i Urbi, a također je omogućeno i sučelje prema ROS-u (Robot Operating System). Sve navedeno čini V-REP vrlo moćnim alatom za brzi razvoj i testiranje rješenja vezanih uz robotiku i automatizaciju [17].

V-REP je vrlo prilagodljiv simulator: gotovo svaki korak simulacije definira korisnik. Ova fleksibilnost dopuštena je putem integriranog tumača skripti. V-REP proširuje naredbe Lua i dodaje V-REPU specifične naredbe koje se mogu prepoznati po njihovim sim-prefiksima (npr. Sim.handleCollision) [16].

Postoji 6 različitih pristupa za programiranje/ kontrolu modela:

1. Ugrađena skripta (eng. Embedded script) – koristi se programski jezik Lua
2. Dodatak (eng. Add-on)
3. Priključak (eng. Plugin)
4. Udaljeni API (eng. Remote API)
5. ROS (eng. Robot Operating System)
6. Prilagođeni klijent/poslužitelj (eng. Custom Client/Server)

U alatu su odmah dostupni već razvijeni modeli nekih poznatih robota (Kuka YouBot) i robotskih ruka (pojedini Jaco, ABB, KUKA...) koji su zamišljeni ili stvarni. Na scenu je moguće dodavati modele robota, strojeva, ljudi te puno druge opreme koja je razvrstana po direktorijima preglednika modela [18].

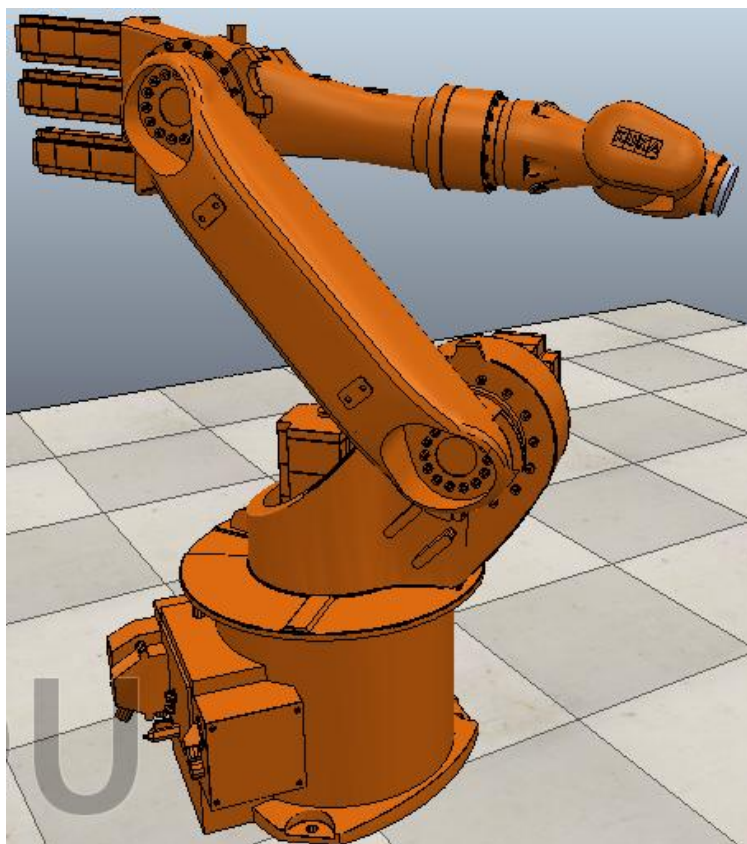


Slika 2.20 Modeli robotskih ruka u V-REP [16]

3.1. Robot KUKA

KUKA Roboter je njemačka tvrtka poznata u svijetu kao jedan od najvećih proizvođača industrijskih robota, koji se koriste u proizvodnji, paletiranju, rukovanju materijalom, automatizaciji utovara, točkastom i elektrolučnom zavarivanju i mnogim drugim područjima.

Automobilska industrija je njihov primarni klijent, ali oni automatizacijskim proizvodima opskrbljuju i svemirsku industriju. KUKA roboti koriste se u brojnim velikim kompanijama, najviše u industriji automobila, ali i u drugim industrijama, kao što su svemirska industrija, prijevoz, proizvodnja hrane i pića, ljevaonice i kovnice, proizvodnja stakla, obrada drva, obrada metala i industrija obrade kamena [19].

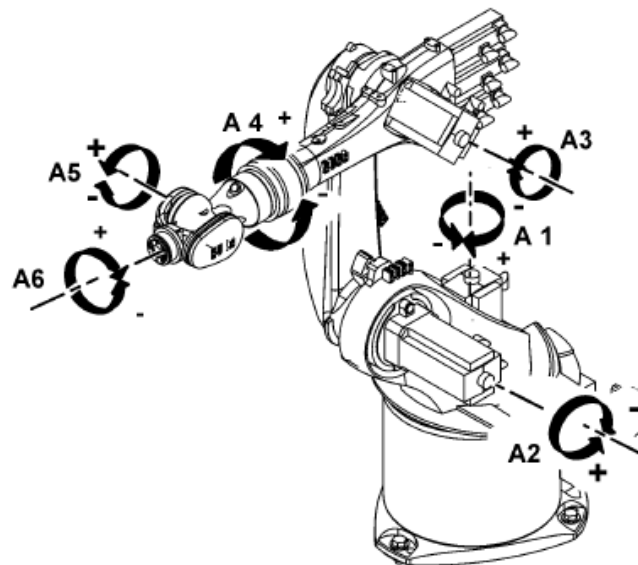


Slika 2.21 Model robotske ruke KUKA KR16 u V-REPu [16]

Odabrani model robota je Kuka- KR16 koji je prethodno uvezen u programski alat V-REP. Robot srednjeg kapaciteta s CNC KUKA upravljanjem prikladan je za radove bojanja, mjerenja ili montiranja te može raditi s teretom mase do najviše 35 kg [19].

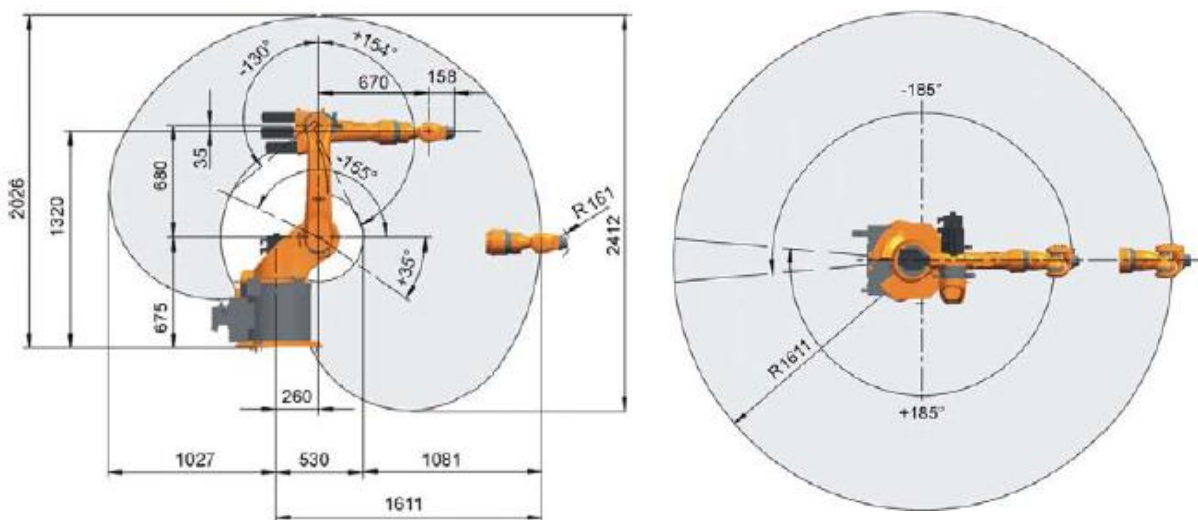
Precizni, brzi pokretač KR 16 ima nosivost od 16 kg i maksimalni domet od 1611 mm. Odlično rješenje za raznoliku primjenu, KR 16 lako se prilagođava. Izrađen od lagane legure, dizajn KR 16 pruža snagu i fleksibilnost.

Njegova kompaktna struktura omogućuje ugradnju na pod, strop i zid. Zaštićeni zglobovi daju KR 16 čistu i modernu izvedbu. Jednostavan za instalaciju i pokretanje, KR 16 nudi vrhunske performanse u bilo kojem procesu - od rukovanja materijalom do zavarivanja, rezanja ili lakiranja. KR 16 može koristiti bilo KR C2 ili KR C4 regulator [20].



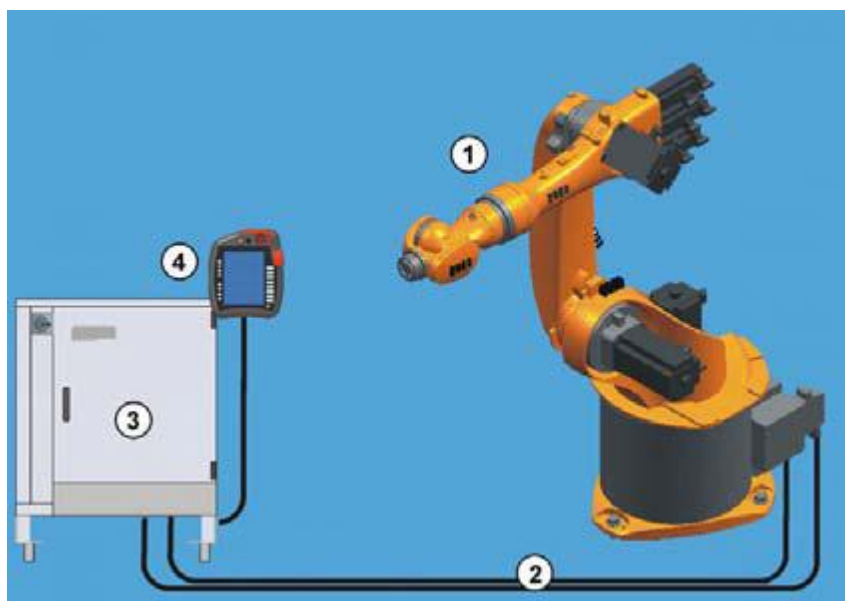
Os	Brzina kretanja	Granica kretanja
A1	156 °/s (2.72 rad/s)	±185°
A2	156 °/s (2.72 rad/s)	+35° - 155°
A3	156 °/s (2.72 rad/s)	+154° - 130°
A4	330 °/s (5.76 rad/s)	±350°
A5	330 °/s (5.76 rad/s)	±130°
A6	615 °/s (10.73 rad/s)	±350°

Slika 2.22 Smjer vrtnje osi robota [21]



Slika 2.23 Radni prostor robota KUKA KR-16 [21]

Svi KUKA roboti rade uz CNC upravljački sustav kojeg je moguće slobodno programirati. KR C4 compact najnovija je i najnaprednija upravljačka jedinica iz Kukiinog asortimana. Omogućava visoke performanse i veliku pouzdanost u kompaktnome dizajnu. Fleksibilna konfiguracija i mogućnost dodatnog proširivanja (programskog ili implementacijom dodatnih osi i pomoćnih elemenata) koju ova upravljačka jedinica nudi, daju joj velike mogućnosti različitih primjena. Broj hardvera, električnih kablova i konektora značajno je smanjen i zamijenjen programskim rješenjima. Robusne i kvalitetne izrade, ova upravljačka jedinica ne zahtjeva veliko održavanje, a rashladni sustav se uključuje samo po potrebi i gotovo je nečujan. Zahvaljujući otvorenoj arhitekturi, KR C4 compact osim robota može kontrolirati i upravljati dodatnim vanjskim osima (linearna traka, rotacijski stol i sl.). Osim standardnog KUKA programskog jezika (KRL), KR C4 može čitati i izvršavati naredbe pisane u standardnom CNC programskom jeziku (G kod), te u jezicima za programiranje PLC-a [22, 23].



- | | |
|-------------------|--------------------------------------|
| 1 Manipulator | 3 Upravljačka jedinica KR C4 compact |
| 2 Spojeni kablovi | 4 Upravljačko sučelje SmartPAD |

Slika 2.24 Primjer robotskog sustava [21]

Sa svojim velikim ekranom osjetljivim na dodir, upravljačko sučelje KUKA SmartPAD omogućava upravljanje i kontrolu nad robotom i nad cjelokupnim sustavom, koji može biti prikazan na ekranu. Sučelje se samostalno prilagođava na način da korisniku prikazuje samo one informacije, i omogućava samo one naredbe koje su u danome trenutku potrebne za potpunu kontrolu nad sustavom. Naglasak je uvijek na onome što je bitno, tako da je rad puno intuitivniji, brži, lakši i učinkovitiji [22, 23].

3.2. Model stalka

Model koji će poslužiti za prikaz bojanja u V-REPU je stalak „Viktorija zdenca“ koji služi za uzimanje pitke vode iz vodovodne mreže. Ugrađuje se na ulicama, kolodvorima, sportskim terenima, tvornicama, školama, naseljima itd. Osiguran je protiv smrzavanja i zagađivanja, jednostavan za rukovanje. Priključak na vodovodnu mrežu izveden je sa G 1/2.



Slika 2.25 Viktorija zdenac

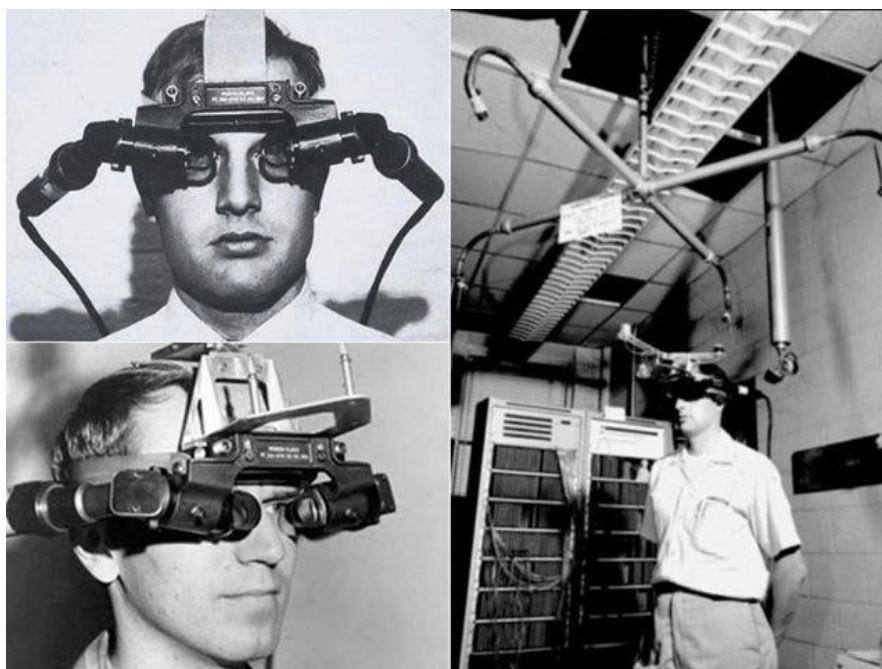
Iz SolidEdge-a pretvara se u datoteku 3D objekta (.stl) zatim se objekt ubacuje u V-REP.



Slika 2.26 Stalak u SolidEdge-u i V-REP-u

3.3. Virtualna stvarnost

Virtualna stvarnost (engleski Virtual Reality - VR) je računalno stvoreno sučelje koje imitira stvarnost, odnosno zbilju i omogućuje interakciju čovjeka u tom imitiranom svijetu. Na taj način simulira se virtualno okruženje u kojem djeluje korisnik. Ivan Sutherland, jedan od najznačajnijih istraživača u području virtualne stvarnosti, 1965. godine iznosi zamisao kako „taj (virtualni) svijet treba izgledati stvarno, zvučati stvarno, imati osjećaj stvarnosti i realistično odgovarati na djelovanje korisnika“ [24].



Slika 2.27 Virtualni koncept iz 1965. godine [25]

Važan koncept koji predstavlja VR je sustav predočavanja, koji se odnosi na osjećaj potpune uronjenosti u trodimenzijskom računalno generiranom okruženju, pomoću koje se iz korisničke perspektive uvelike olakšava. Glavna prednost korištenja virtualne stvarnosti očituje se kao mogućnost stvaranja i simuliranja novih tehnoloških jedinica prije njihove realizacije. Ovo može doprinijeti povećanju sigurnosti, ergonomije i smanjenju troškova. Virtualni model radnog mjesta mogao bi otkriti skrivene pogreške koje se eliminiraju, a koje bi u kasnijoj fazi na tom radnom mjestu mogle uzrokovati velike poteškoće [26].

Virtualna stvarnost uvodi se zbog smanjenja troškova razvoja kod pojedinih procesa, povećanja produktivnosti, lakšeg rukovanja dijelovima ali i prvenstveno zbog različitih mogućnosti koje pruža, a koje su sve donedavno bile nezamislive. Tipični strojarski primjeri mogu se naći u području robotike, razvoja proizvoda i virtualnog sklapanja.

Primjena virtualne stvarnosti uranja korisnika u računalno generiranu okolinu koja simulira stvarnost kroz upotrebu interaktivnih uređaja koji šalju i primaju informacije i nose se kao naočale, slušalice, rukavice ili odijela.



Slika 2.28 HTC Vive VR set [25]

Danas razvijenost tehnologije omogućuje da virtualna stvarnost ima sve širu primjenu, pa se ona susreće u poljima poput medicine, strojarstva, robotike, raznim granama industrije, ali i u području zabave, gdje omogućuje upoznavanje i sudjelovanje vrlo široke populacije.

Ona se primjerice koristi vrlo često u obuci za novi posao ili u prezentiranju nekog koncepta ili iskustva publici. Brojna su područja koja koriste virtualnu realnost i načina na koji je uklapaju u svakodnevni život, a to su zabava, automobilska industrija, marketing, obrazovanje, svemirska istraživanja te sve više u obučavanju vojske i policije. Privlačnost simulacija je što one mogu pružiti obuku jednaku ili gotovo jednaku praksama sa stvarnim sustavima, ali o manjem financijskom trošku i s većom razinom sigurnosti [27].



Slika 2.29 Primjer primjene u industriji [28]

Virtualna stvarnost u strojarstvu uključuje upotrebu 3D alata za modeliranje i vizualizacijske tehnike kao dio dizajnerskog procesa. Tehnologija omogućava inženjerima da pregledaju svoj projekt u 3D, te da steknu bolje razumijevanje kako taj objekt radi. Još jedna prednost je da se svaka mana ili potencijalni rizik može uočiti prije same proizvodnje i implementacije (Virtual Reality in Engineering, url). Samim time i dizajnerski tim može promatrati svoj projekt u sigurnoj okolini i raditi promjene tamo gdje su potrebne.

Proizvođači automobila ovu tehnologiju koriste tijekom crtanja i kreiranja prototipova. To im omogućava da proizvedu nekoliko verzija prototipova te da ih mijenjaju sukladno dobivenim rezultatima iz testova. Još jedna prednost je da se ne mora raditi fizički prototip te se upravo time štedi na vremenu i novcu (Virtual Reality in Engineering, url). Važnost virtualne stvarnosti je njezina sposobnost prikazivanja najmanjih detalja u projektu kako bi se održala iluzija stvarnosti. Za to su potrebna vrlo snažna računala, video sa visokom brzinom osvježavanja slike i realistični zvuk i fizika tijela [24].



Slika 2.30 3D modeliranje pomoću virtualne stvarnosti [29]

Neke od glavnih prednosti implementacije VR-a na 3D projektiranja:

1. Pregled, analiza i validacija sklopa ili cijelog proizvoda

VR omogućava projektantima, dizajnerima, prodajnim i marketing osobama, kupcima, da pomoću virtualnog modela proizvoda pregledaju i analiziraju proizvod u kontekstu validacije konstrukcije i dizajna, montaže i demontaže, provjere ergonomije, provjere vizualnih karakteristika, ponašanja proizvoda itd.

2. Unaprijeđenje metode edukacije

Tvrtke mogu unaprijeđivati metodologiju edukacije zaposlenika prije finaliziranja proizvodnje i izrade prototipa, kao i nakon izrade proizvoda i njegove eksploatacije. Zaposleni mogu steći realan osjećaj kako će proizvod izgledati i mogu efikasno pripremiti komunikaciju s potencijalnim kupcima i educirati tržište.

3. Rano otkrivanje nedostataka

Kako bi se izbjegli eventualni nedostaci u procesu projektiranja i omogućio redizajn prije finalizacije, VR pomaže inženjerima identifikaciju potencijalnih problema i uklanjanje nepravilnosti. Radi se o izrazito korisnoj funkcionalnosti jer se prevencijom grešaka maksimalno smanjuju nepotrebni troškovi.

4. Simulacija stvarnih situacija

Inženjeri mogu gotovo fizički zakoračiti u svoj projektirani prostor i realnije iskusiti dijelove ili cijeli proizvod te uočiti nepravilnosti na vrijeme.

5. Poboljšana komunikacija i odsluškivanje tržišta

Odobrenja u procesu projektiranja, na koja se često dugo čeka i zbog kojih proizvodni ciklus može kasniti, se znatno brže provode. Projektanti i kupci, odnosno naručitelji posla, mogu s različitih lokacija pristupati istom virtualnom svijetu. To znatno olakšava komunikaciju, razumijevanje i instrukcije vezano za naknadne izmjene, a sve to u realnom vremenu i prije završne narudžbe. [29]

Automobilska industrija

Kompanija Ford već koristi virtualnu realnost u svom laboratoriju za uživanje kako bi pružila osjećaj koji ima kupac u svom novom automobilu. Pritom koristi uređaj Oculus Rift kako bi osobi omogućila pogled na interijer i eksterijer automobila. Također je razvila alate koji će biti korišteni u VR iskustvu razgledavanja automobila u mraku. To je vrlo korisno jer Ford može brže raditi na razvoju proizvoda bez čekanja na fizički prototip novog modela.

I Audi će koristiti virtualnu realnost kako bi potencijalnim kupcima automobila pružio dubinski pogled na automobile.

Toyota je koristila virtualnu realnost u sklopu svoje kampanje TeenDrive365 kako bi educirala tinejdžere i roditelje o dekoncentraciji u vožnji. Simulator demokracije u vožnji uključuje senzore koji prevode ono što korisnik čini s pedalama ili volanom u simulaciju, te uključuje odvlačenje pažnje poput zvonjave mobitela [30].



Slika 2.31 Primjena VR u automobilskoj industriji [29]



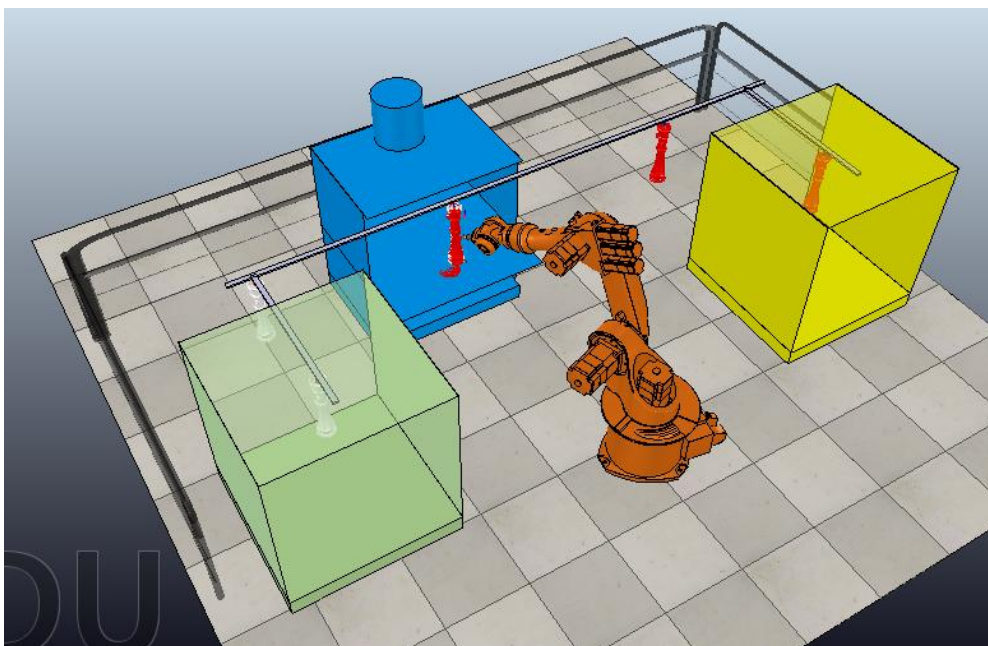
Slika 2.32 Upravljanje robotskom rukom pomoću VR u V-REPU [29]

4. Izrada programa za bojanje robotom u alatu V-REP

Zbog što manjeg izlaganja ljudi na teške uvjete rada, opasnosti u blizini zapaljivih tvari i otrovnim supstancama iz boja i lakova dolazi do potrebe automatizirane proizvodnje. Klasično bojanje raspršivačem zamjenjuje se robotom koji se precizno programira tako da oponaša ljudsku ruku. Preciznim programiranjem pokreta robotske ruke može se povećati kvaliteta premaza, potrebna debljina premaza i brzina lakiranja radnih komada. Količina i narudžbe od strane kupaca su naravno za veliko serijsku proizvodnju, jer za male pojedinačne količine ne bi bilo isplativo automatizirati proizvodnju.

U projektu će se prikazati automatizirano postrojenje za lakiranje radnih komada pomoću robotske ruke. Radni komadi vješaju se na transporter, te ulaze u sačmaru gdje se čiste do zadovoljavajuće čistoće površine. Preko vodilica radni komad dolazi do pozicije gdje pozicioner okreće radni komad dok istovremeno robotska ruka nanosi slojeve premaza ovisno o debljini sloja koji je potrebni. Kretanje robota i linije su tehnološki razrađene prije nego se pristupi samoj realizaciji posla. Kod pozicije zaustavljanja radnog komada istovremeno i radni prostor robota, nalazi se filter za usisivanje štetnih supstanci iz boja i prikupljanje čestica viška boje koje se prilikom raspršivanja prenesu izvan radnoga komada. Nakon što se program bojanja završi obojani komadi se transportiraju po vodilici do zagrijane sušare koja je opremljena sa ventilatorima koji upuhuju topli zrak za brzo sušenje komada. Izlaskom iz sušare radni komadi se premještaju u spremište u kojem moraju komadi stajati ovisno o temperaturi i vlazi minimalno 24 sata kako bi se boja što bolje polimerizirala.

Postrojenje za lakiranje robotom prikazati će se u programskome alatu V-REP.



Slika 4.1 Robotska ćelija u V-REPu

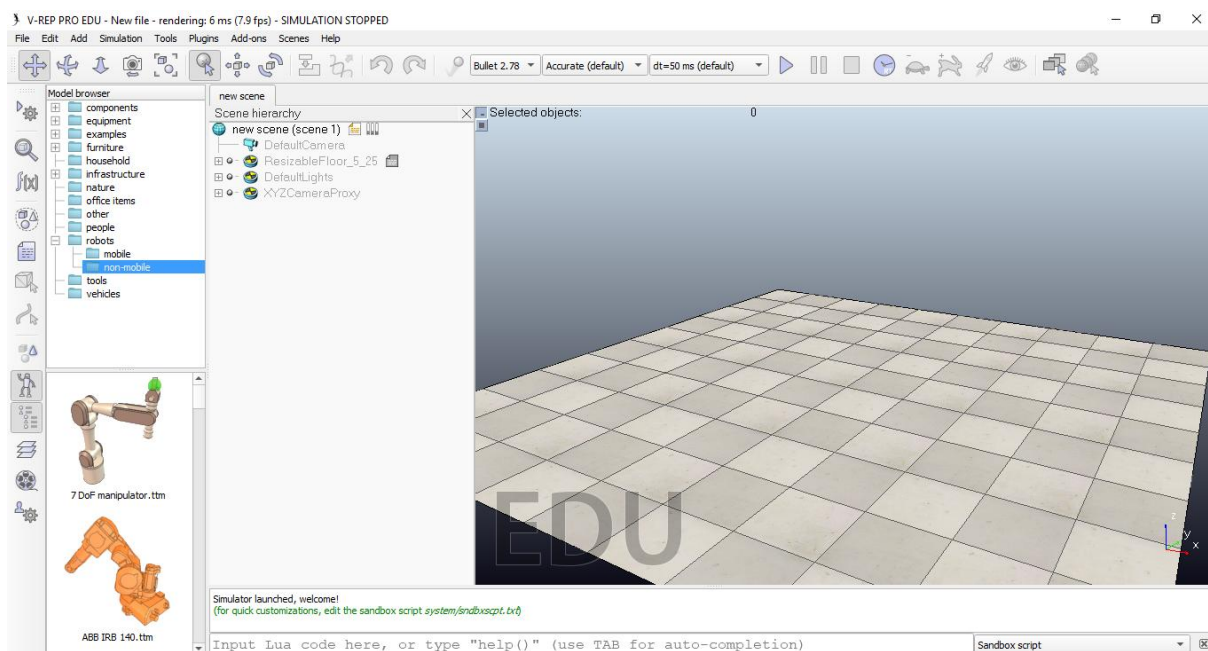
4.1 Pokretanje programskog alata V- REP

Programski alat V- REP PRO EDU je besplatni edukativni alat za sve studente, profesore i hobiste.



Slika 4.2 Ikona za pokretanje programskog alata V-REP

Dvoklikom na ikonu V-REP PRO EDU otvara se zaslon programa ujedno i nova scena. U zaslonu se nalazi pod na koji se ubacuju modeli, stablo koje prikazuje povezanost i pretraživač već gotovih modela. Ostale funkcije promjene položaja modela, prikaz simulacije, izrada putanje itd. nalaze se također unutar zaslona programa.

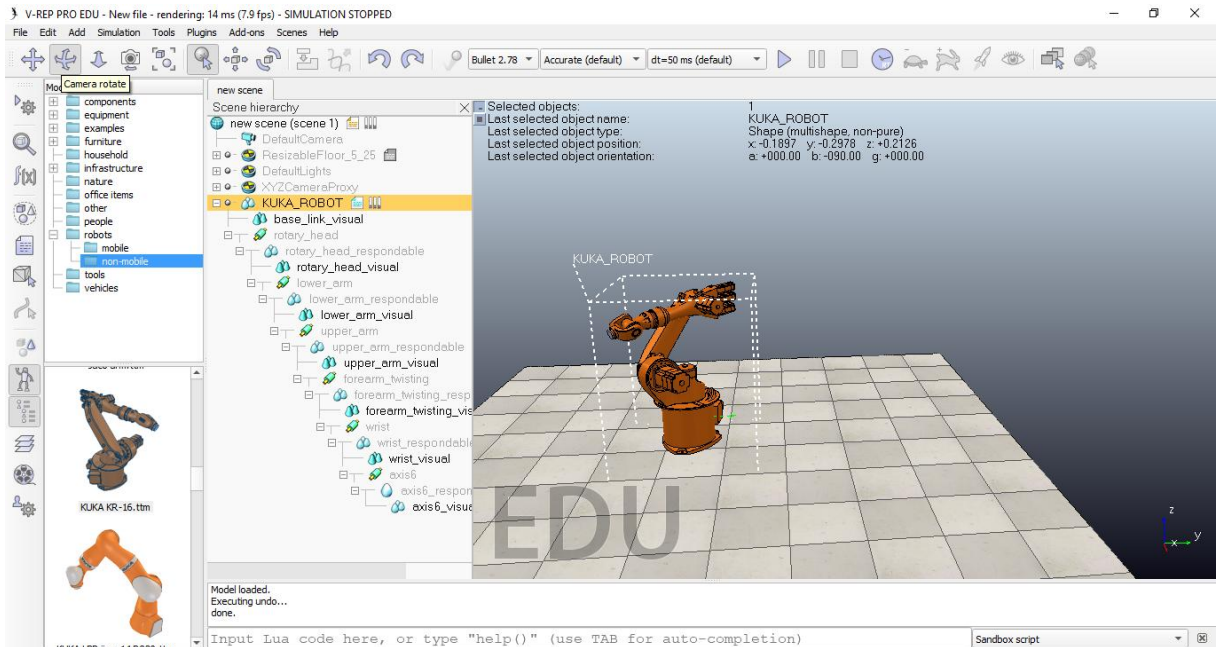


Slika 4.3 Zaslona i nova scena programskog alata V-REP

4.2 Dodavanje robota KUKA KR-16

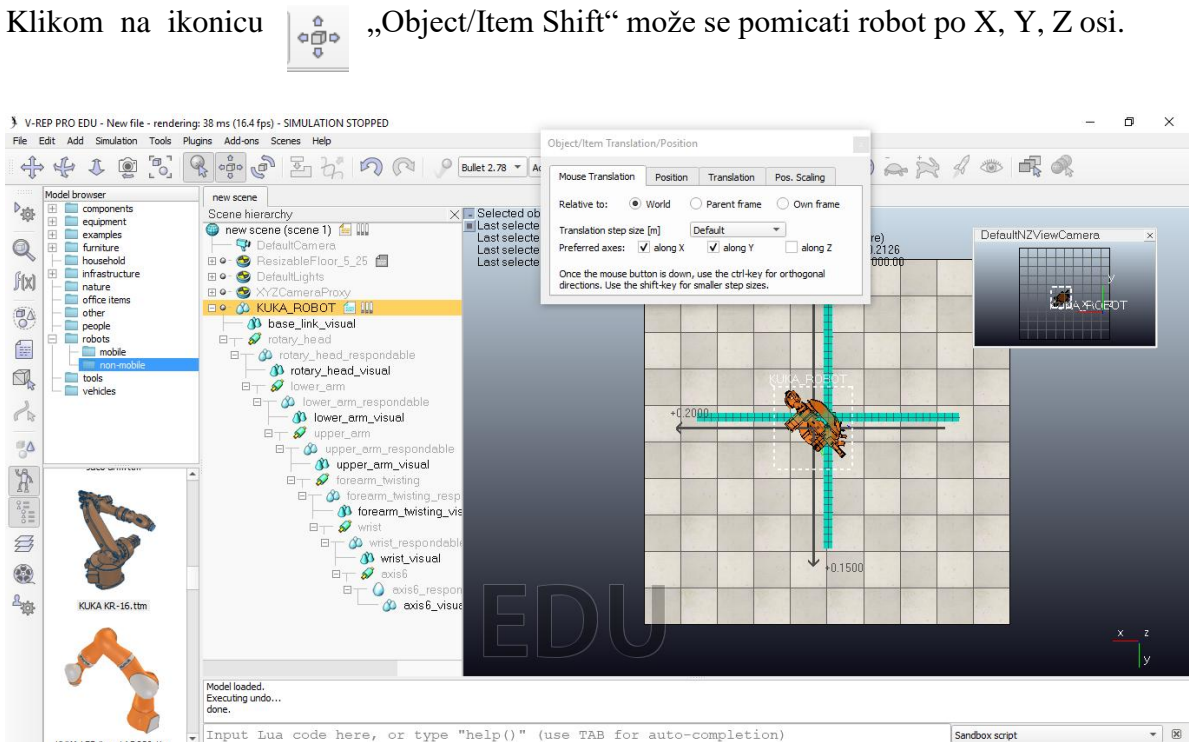
U pretraživaču modela postoje dvije vrste modela robota mobilni ili pokretni roboti i nepokretni roboti tj. oni koji se ne kreću cijelim tijelom, odnosno ne mijenjaju položaj u nekom prostoru, većinom industrijske ruke.

Odabrani je model robota KUKA KR-16, te se lijevim klikom miša prenese u prozor scene.



Slika 4.4 Odabir robota i stavljanje u scenu

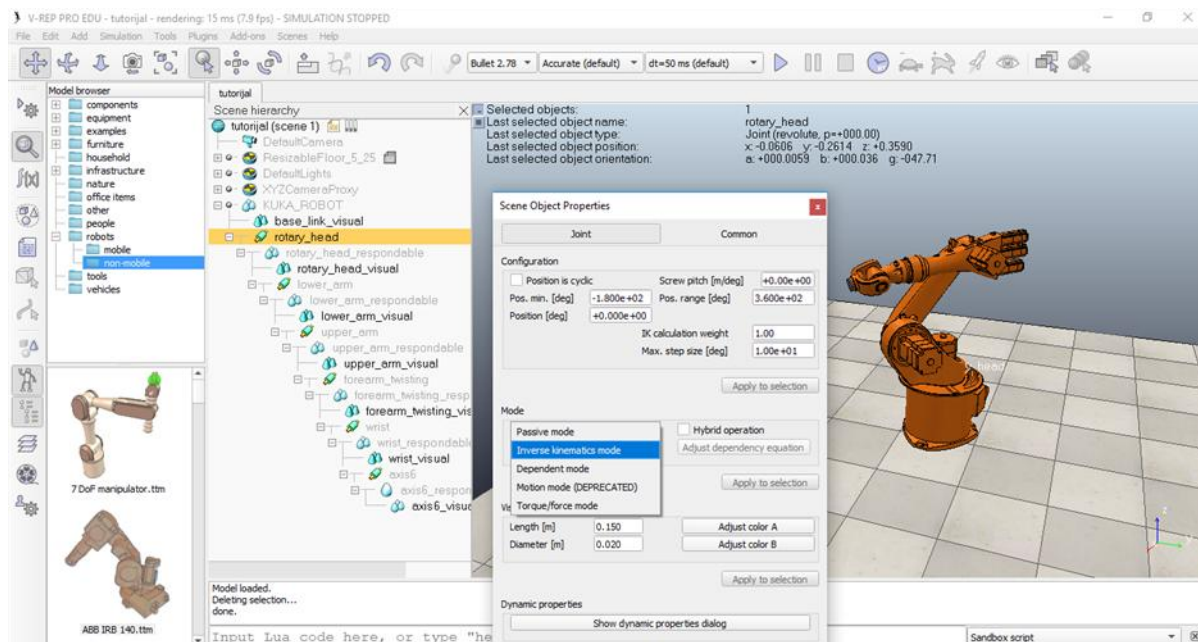
Klikom na ikonicu „Object/Item Shift“ može se pomicati robot po X, Y, Z osi.



Slika 4.5 Pomicanje i promjena položaja robota po koordinatnim osima

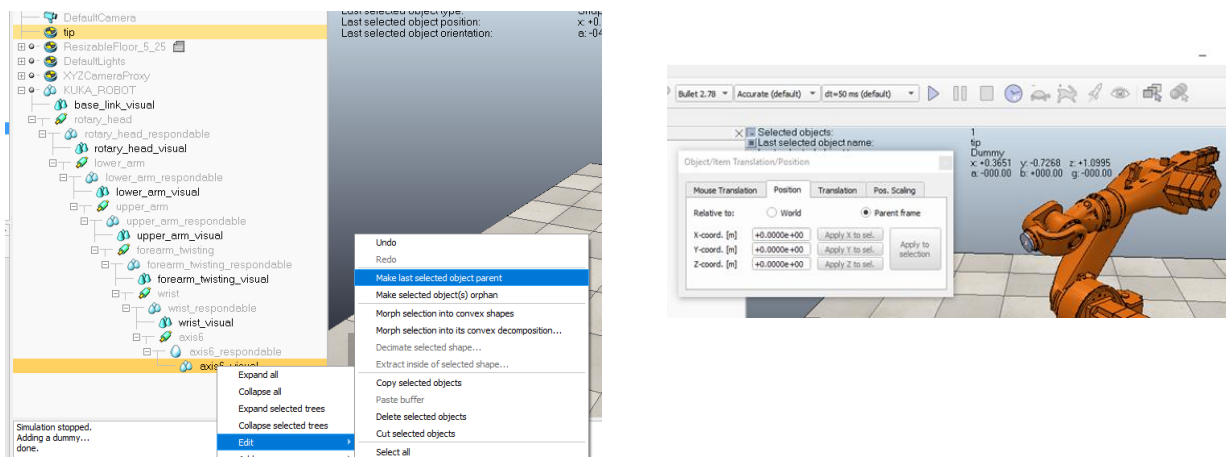
4.2.1 Postavljanje inverzne kinematike robota

Na svaki udruženi spoj modela od kojih se sastoji robot potrebno je postaviti način u inverznu kinematiku, koja pronalazi položaje i kuteve svih segmenata i zglobova, tako da manipulator bude u željenoj krajnjoj točki. Lijevim dvoklikom na „Joint“ otvara se prozor u kojem se promijeni način (Mode → Inverse kinematics mode).



Slika 4.6 Postavljanje u način inverzne kinematike

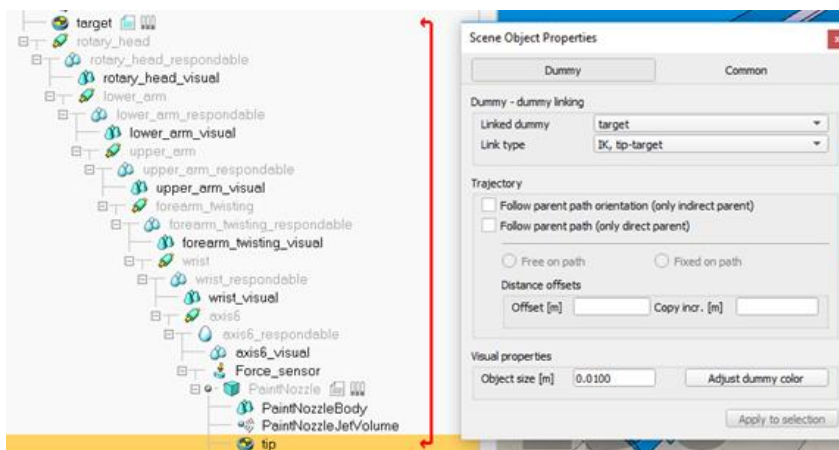
Ubacivanje objekta ili točke s orijentacijom „Dummy“ (Menu bar → Add → Dummy) koji su višenamjenski ili pomoćni predmeti, te se koriste u kombinaciji s drugim objektom ili modulima za proračun minimalne udaljenosti od drugog objekta da ne dođe do sudara. Preimenuje se u „tip“, označi se tip i axis6_visual, zatim se klikne (Menu bar → Edit → Make last selected object parent), te se spaja sa nadređenim objektom na zadnjoj osi robota



Slika 4.7 Spajanje i pozicioniranje Dummy „tip“ sa zadnjom osi robota

Zatim se ponovo ubacuje točka s orijentacijom „Dummy“ (Menu bar → Add→ Dummy) ili se kopira „tip“, te zaljepi i preimenuje u „target“. „Target“ je točka na putanji koju će slijediti u položaju i orijentaciji točka „tip“ koja je povezana sa robotom, dakle točka „tip“ dovodi robot do neke putanje na kojoj se nalazi točka „target“ koja putuje po njoj.

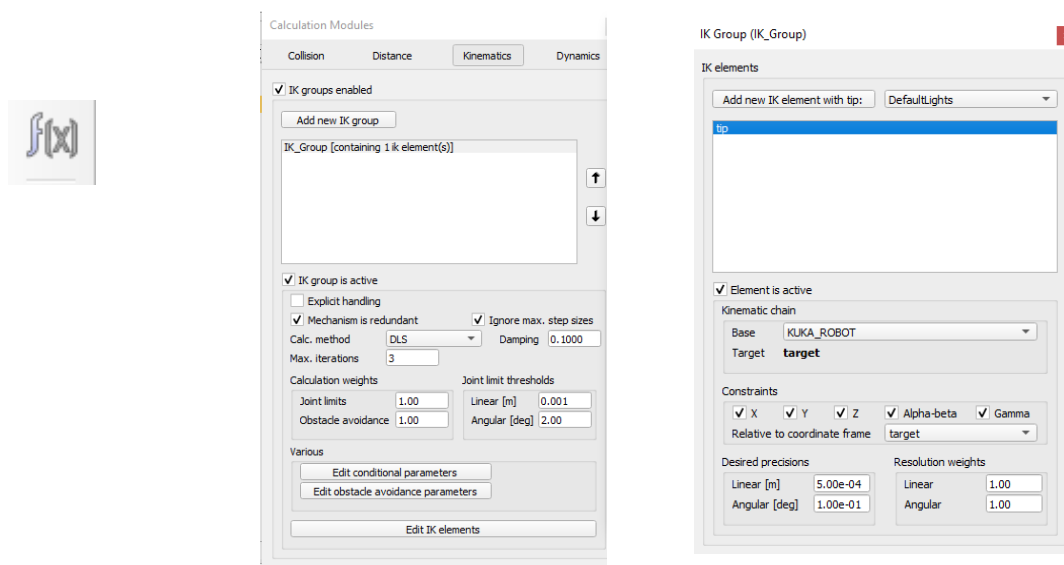
Nadalje mora se obavijestiti V-REP da su „tip“ i „target“ par za inverznu rezoluciju kinematike. Dvostrukim lijevim klikom na „tip“ otvara se prozor u kojem se povežu „tip“ i „target“ (Linked dummy → target), te se prikažu crvenom strelicom u stablu scene.



Slika 4.8 Spajanje Dummy „tip“ i „target“

Otvaranje računskih modula za proračun koji omogućavaju rješavanje na vrlo učinkovit način bilo koje vrste inverznog kinematičkog problema

(Calculation modules → Kinematics → Add new IK group → Edit IK elements → odabere se „tip“ i Add new IK elements with tip → izabire se da će baza biti KUKA_ROBOT → koordinate prema „target“).



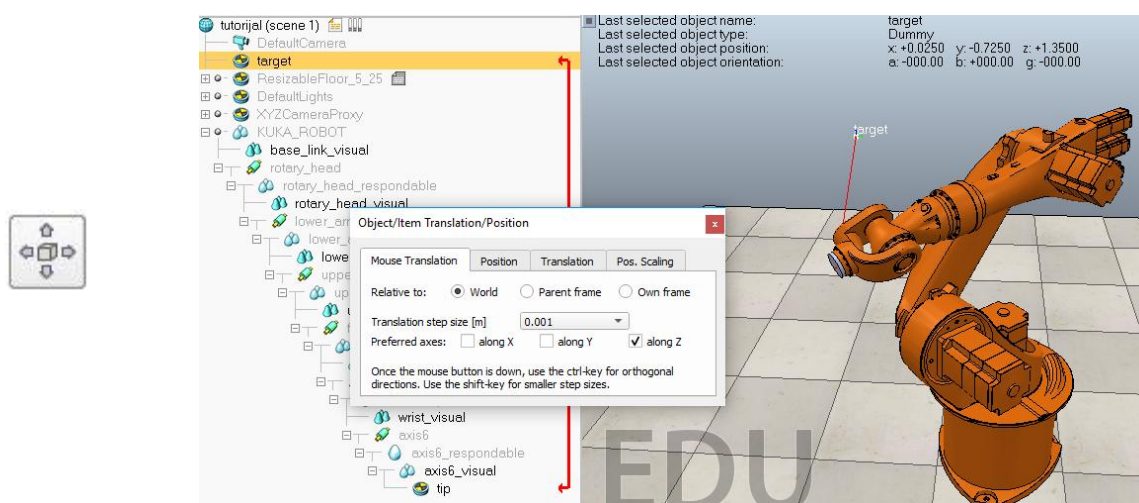
Slika 4.9 Postavljanje kinematičkih računskih modula

Povratkom u prozor Calculation Modules uključi se kvačica **Mechanism is redundant**, te odabire da se primjene zajedničke korekcije krajnjih granica tijekom IK rezolucije, a u suprotnom ako se provode nakon može dovesti do nestabilnosti.

Calc. method- metoda izračuna koja se koristi za specificiranu razlučivost IK grupe odabiremo DLS jer je sporiji, ali stabilniji kada meta i vrh leže predaleko, kad je kinematički lanac preopterećen ili kad je mehanizam u blizini pojedinačne konfiguracije.

Nakon što se postave računski moduli, može se pozicionirati „target“ da bude vidljiv i udaljeniji od „tip“, te da se vidi linija kretanja od početnog položaja do ciljnog.

(Object/item shift → i pomiče se po Z- osi)



Slika 4.10 Pomicanje ciljne točke „target“

Na kraju „target“ označi se klikom i prenese u podnivo stabla sklopa robota KUKA_ROBOT.

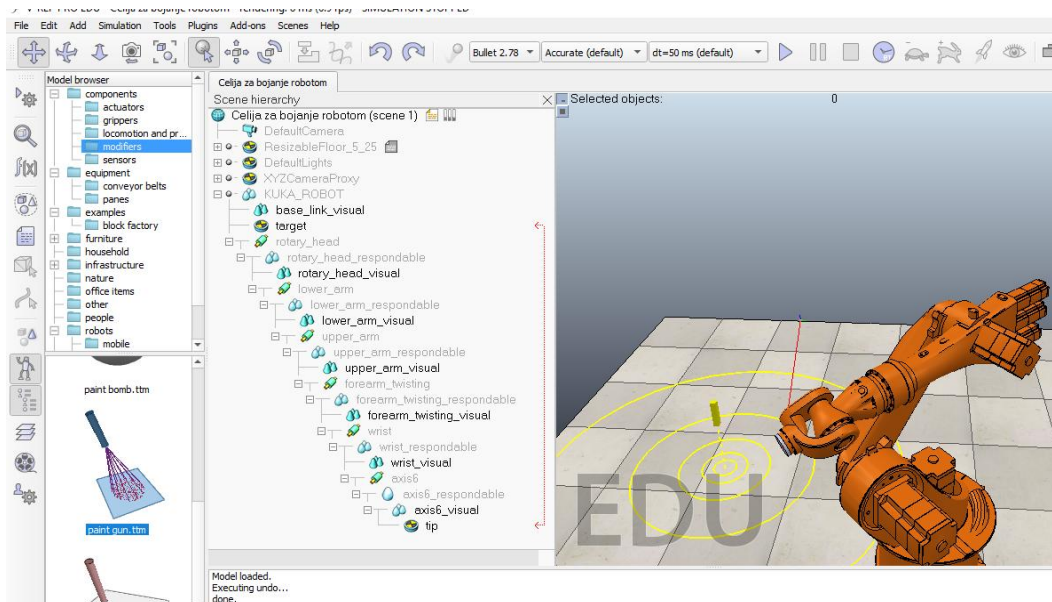
Time je uspješno postavljena inverzna kinematika robota, tako da je sad robot pod kontrolom kretnje prilikom mijenjanja pozicije „target-a“.

Također prije sljedećih radnji i koraka mogu se isprobati prikazi kretnje i vidjeti da li je sve dobro postavljeno u inverznoj kinematici robota, preko simulacije koja se otvara pritiskom na tipku Start/Resume simulation.



4.2.2 Postavljanje senzora i pištolja za boju

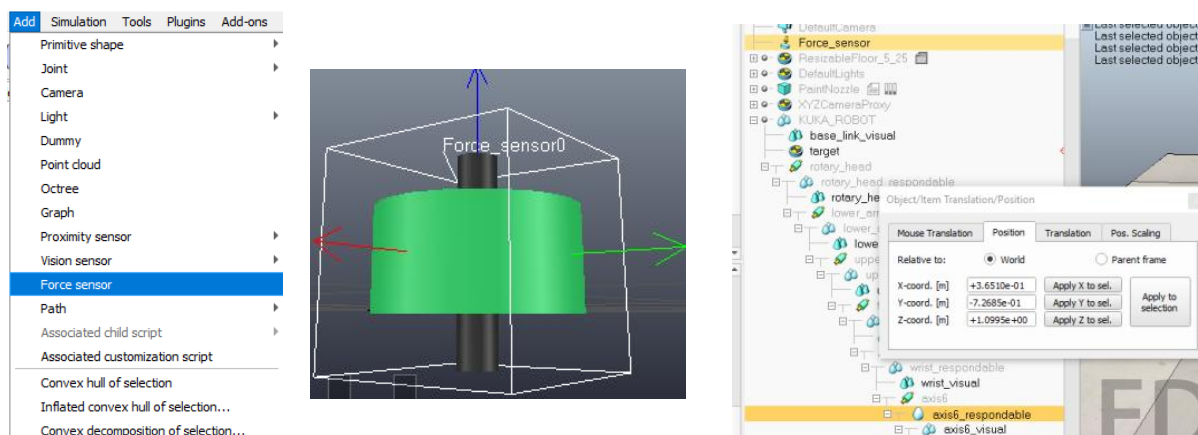
Prvo će se pronaći model pištolja za boju (Model browser→ components→ modifiers→ paint gun i prenese se u prozor sa scenom).



Slika 4.11 Prijenos pištolja za boju

Da bi se mogao postaviti pištolj za boju potrebno je postaviti senzor sile koji mjeri prenesene sile i zakretne momente. (Add→ Force sensor)

Potom označi se Force sensor i axis6_respondable, te otvori Object position→ i klik na Apply to selection. Tada se prenese senzor na istu poziciju na kojoj je vrh zadnjeg modela robota

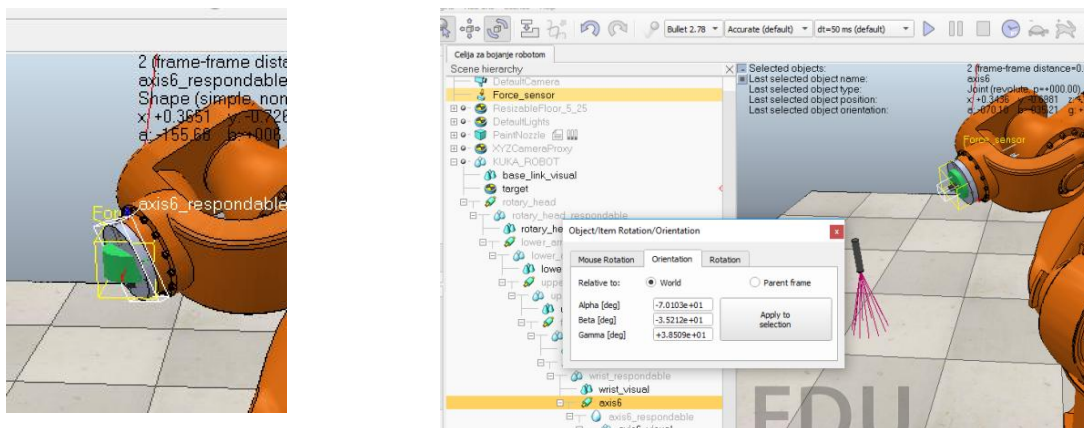


Slika 4.12 Dodavanje, izgled i pozicija senzora

Senzor je potrebno okrenuti, te će poslužiti alat **Object rotation**.



(Označi se Force sensor i axis6 → Object rotation → Apply to selection).

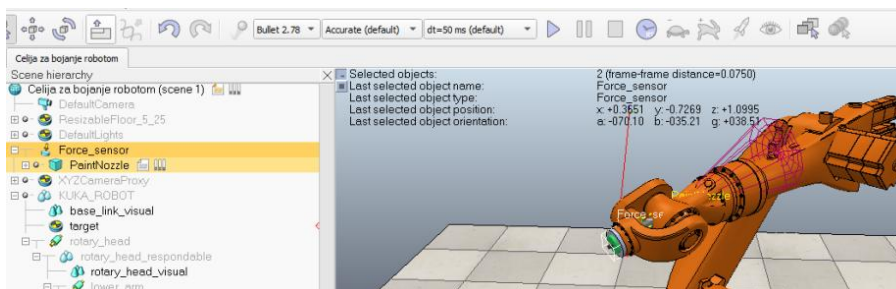


Slika 4.13 Okretanje senzora u položaj osi

Slijedi spajanje senzora i pištolja za bojanje, kod kojeg će pomoći alat **Assemble**.

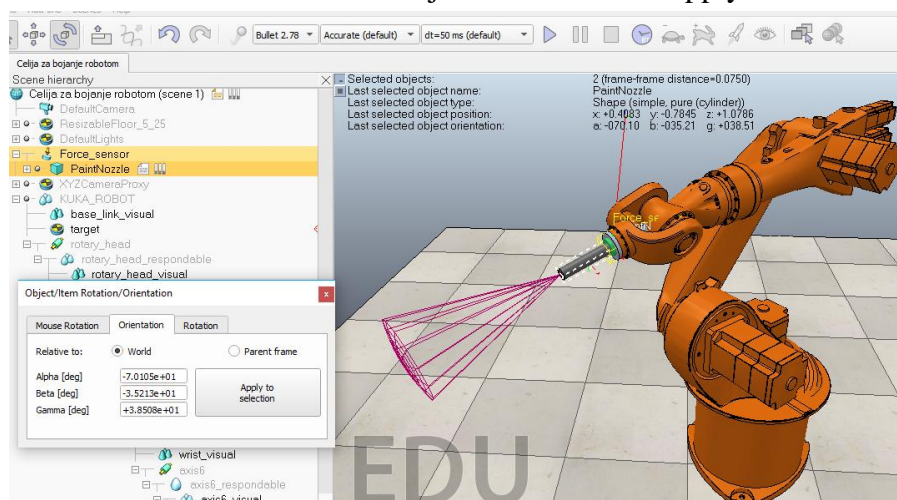


(Označi se PaintNozzle i Force sensor → Assemble).



Slika 4.14 Sastavljanje senzora i pištolja za boju

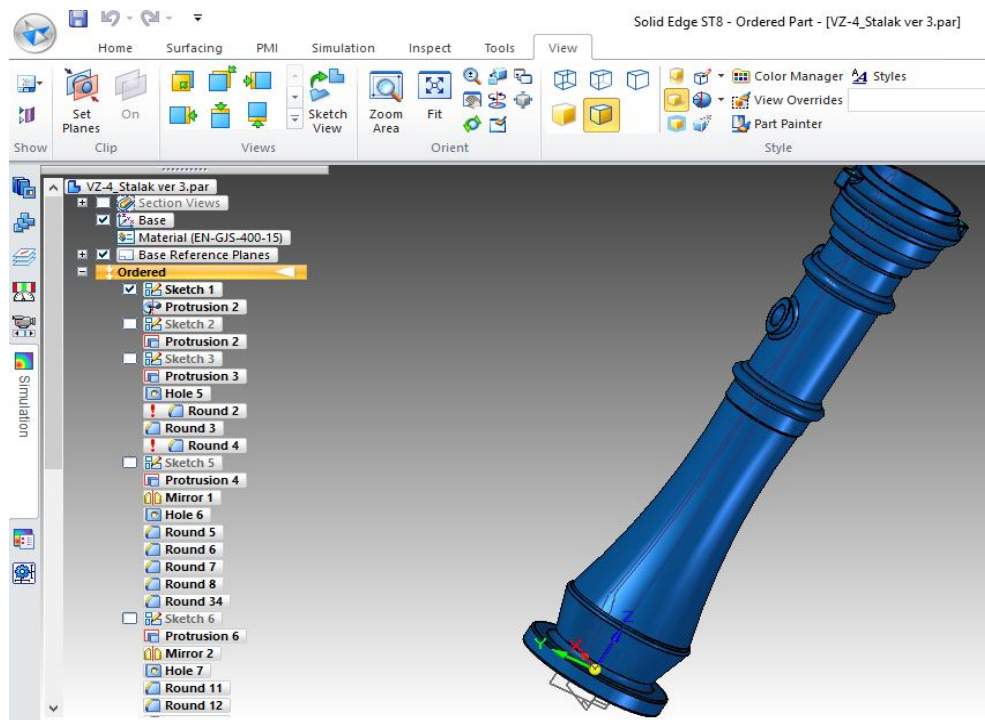
Došlo je do problema da se pištolj za boju spojio sa senzorom u krivu stranu, rješenje je da se (označi ponovo Force sensor i PaintNozzle → Object orientation → Apply to selection).



Slika 4.15 Okretanje pištolja za boju u ispravnu stranu

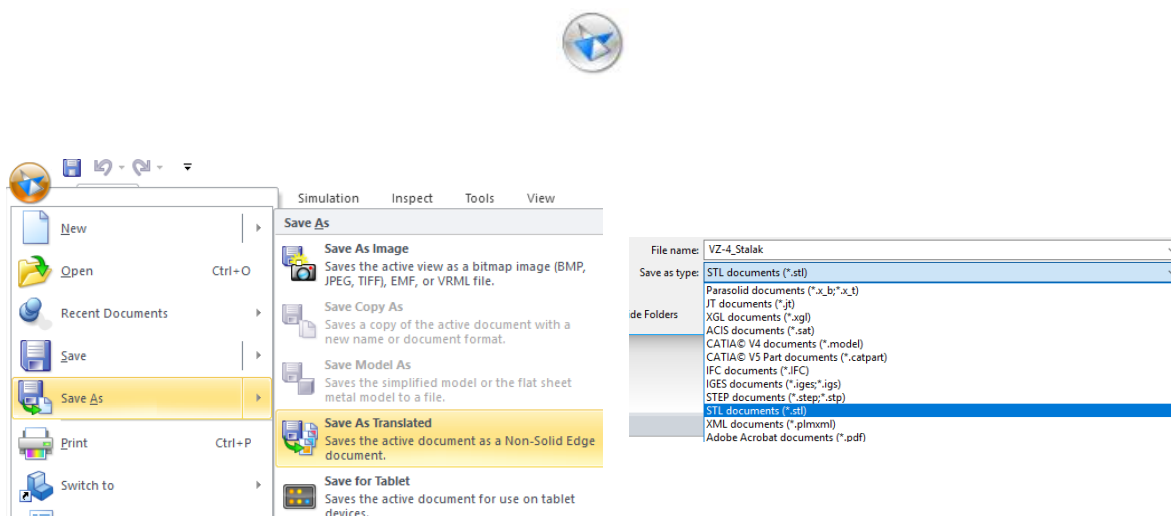
4.3 Ubacivanje modela iz Solid Edge-a u V-REP

Model stalka koji je izmodeliran u programu Solid Edge, potrebno je prvo pretvoriti u 3D model u STL dokument kojeg se može uvesti u V-REP.



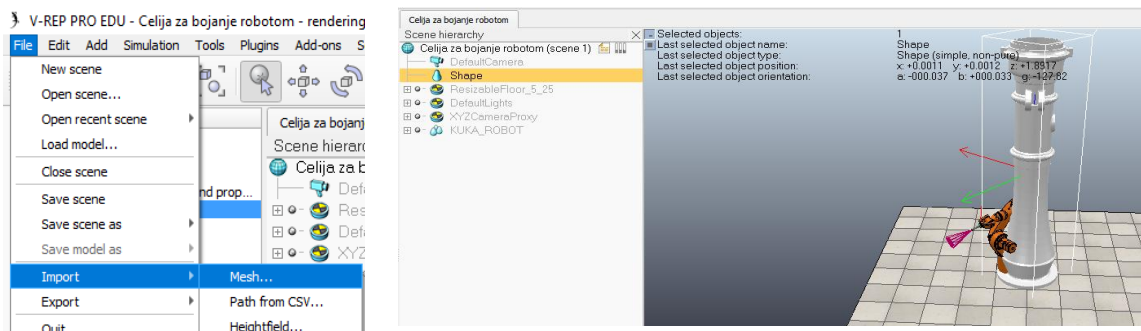
Slika 4.16 Model stalka u Solid Edge- u

U Solid Edge- u klikom na (Application button → Save as → Save as Translated → Save as type → odabire se STL documents (*.stl) i potvrdi sa Save).



Slika 4.17 Pretvaranje modela u STL dokument

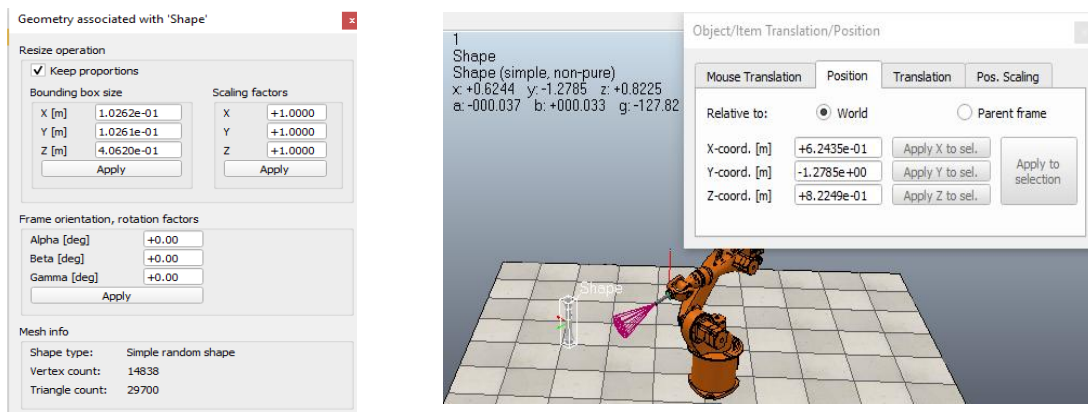
U V-REP se uveze model stalka preko (File→ Import→ Mesh... → odabere se pretvoreni STL dokument→ Open).



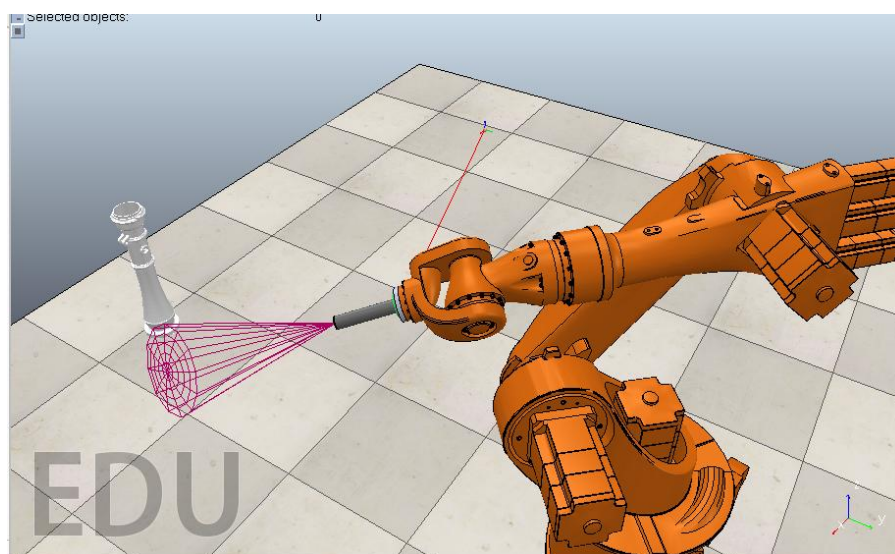
Slika 4.18 Uvoz 3D modela u V-REP

Ako je model nešto veći moguće ga je smanjiti, (dvostukim klikom na model→ view/modify geometry→ Bounding box size→ nakon unosa podataka potvrdi →Apply).

Sa Object position se dovede model na željenu poziciju.



Slika 4.19 Smanjivanje modela stalka i njegovo pozicioniranje



Slika 4.20 Prikaz položaja modela stalka

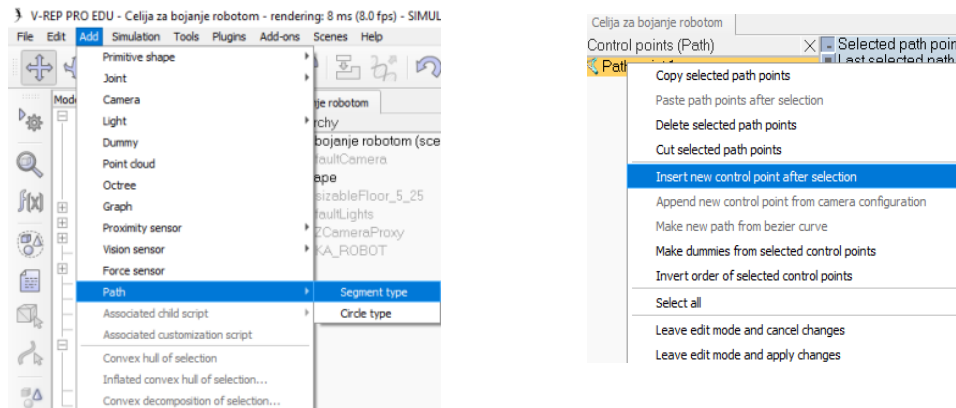
4.4 Izrada putanje za model stalka

Putanja je objekt koji definira stazu ili putanju u prostoru, koristi se zajedno sa objektima „Dummy“ koji putuju po putanji koja se spaja sa kontrolnim točkama.

(Add → Path → Segment type → označi se path i klikne na alat path edit mode

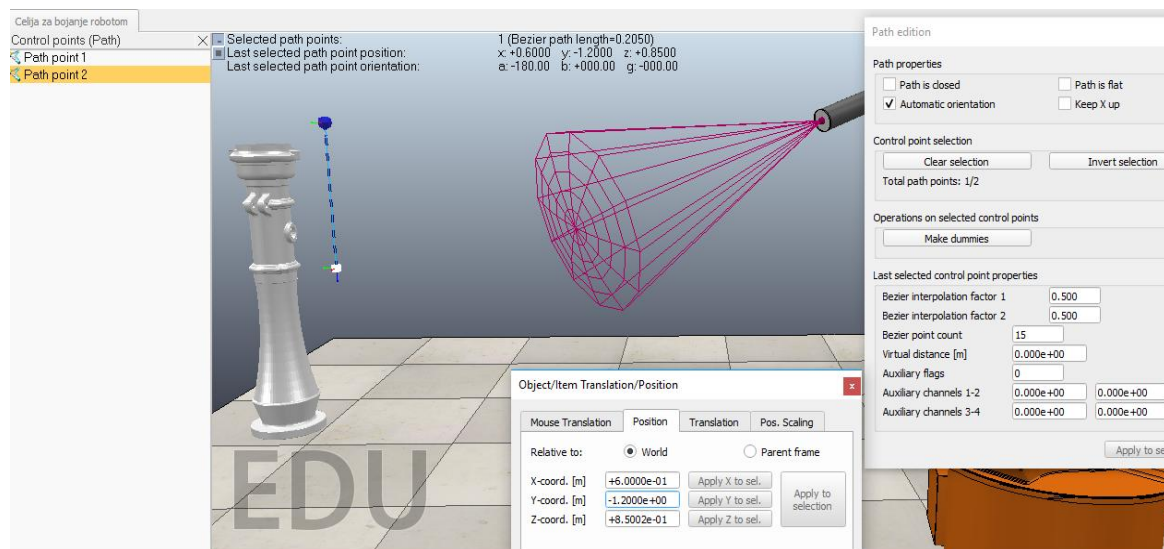


→ Otvara se prozor u kojem se radi putanja, te se dodaje nova točka → Desnim klikom miša i „Insert new control point after selection“).



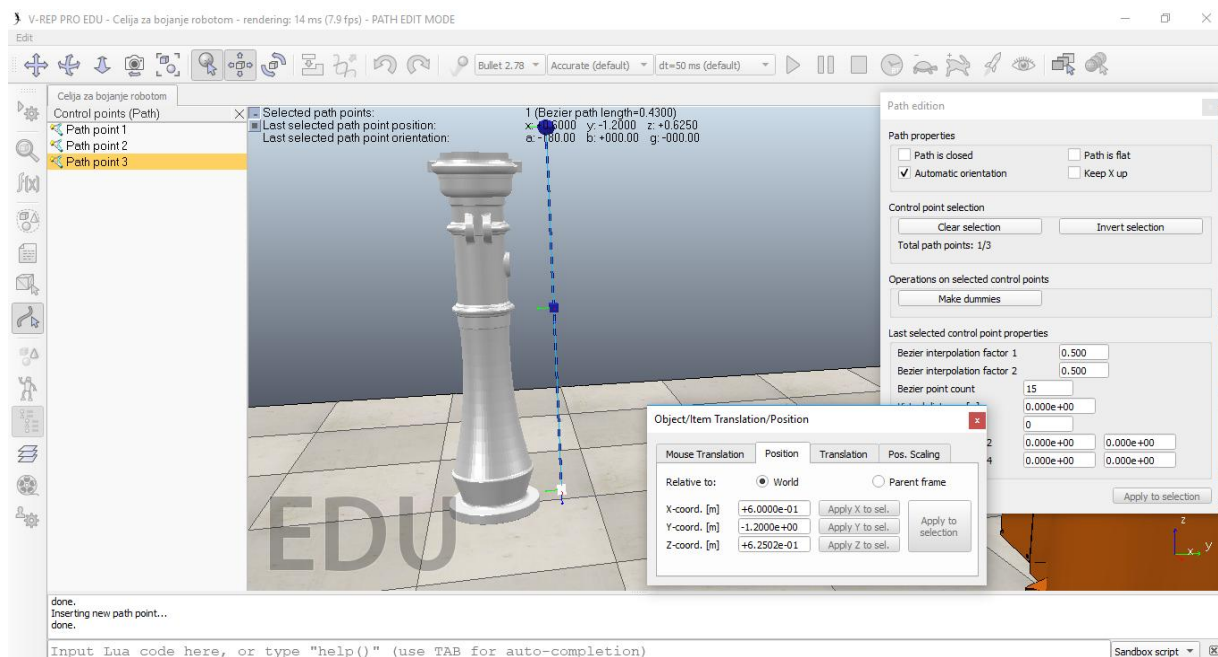
Slika 4.21 Dodavanje točki putanje

Nakon što se doda druga točka potrebno ju je pozicionirati sa Object position.



Slika 4.22 Dodavanje druge točke putanje i pozicioniranje

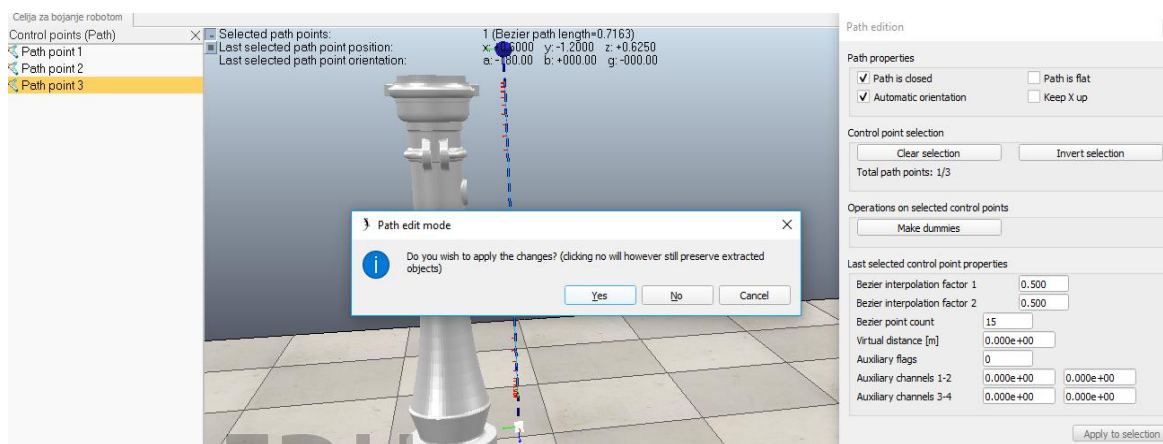
Dodavanje treće točke putanje dodaje se na isti način (Desnim klikom miša i „Insert new control point after selection“ → pozicionira se sa Object position).



Slika 4.23 Dodavanje treće točke putanje i pozicioniranje

Prije nego se zatvori prozor za uređivanje putanje, potrebno je rad učiniti cikličnim. Povezivanjem posljednje kontrolne točke putanje s njegovom prvom kontrolnom točkom kako bi se zatvorio put, potrebne su najmanje 3 kontrolne točke.

Odabrati kvačicom na **Path is closed** → pritiskom na X u Path edition izlazi se iz prozora za uređivanje putanje, te nas pita da li želimo primjeniti promjene potvrdimo → **Yes**

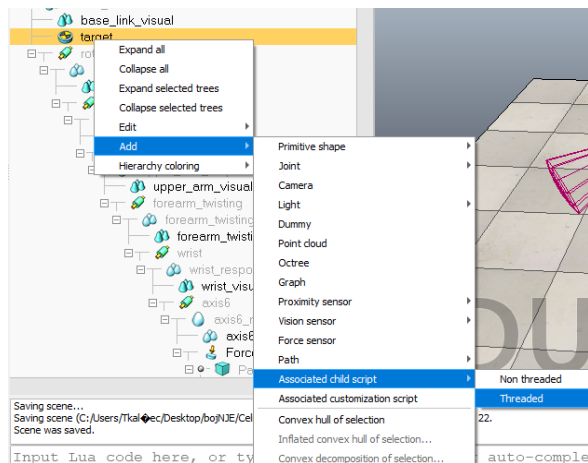


Slika 4.24 Povezivanje kontrolnih točki i izlaz iz prozora za uređivanje putanje

4.5 Upravljanje skriptom pomoću Lua funkcija

Roboti i modeli upravljaju se pomoću podređenih simulacijskih skripti i koriste se za prilagodbu simulacije ili simulacijskog modela. Skripte su povezane sa scenskim objektima i omogućuju rukovanje određenom funkcijom u simulaciji.

(Target desni klik → Add → Associated child script → Threaded).



Slika 4.25 Dodavanje skripte „target-u“

Nakon što se doda skripta „target- u“ potrebno je upisati naredbe pomoću Lua funkcija koje se nalaze u korisničkom priručniku V- REPa.

(U tabu Help → Help topics → otvara se V-REP User Manual).

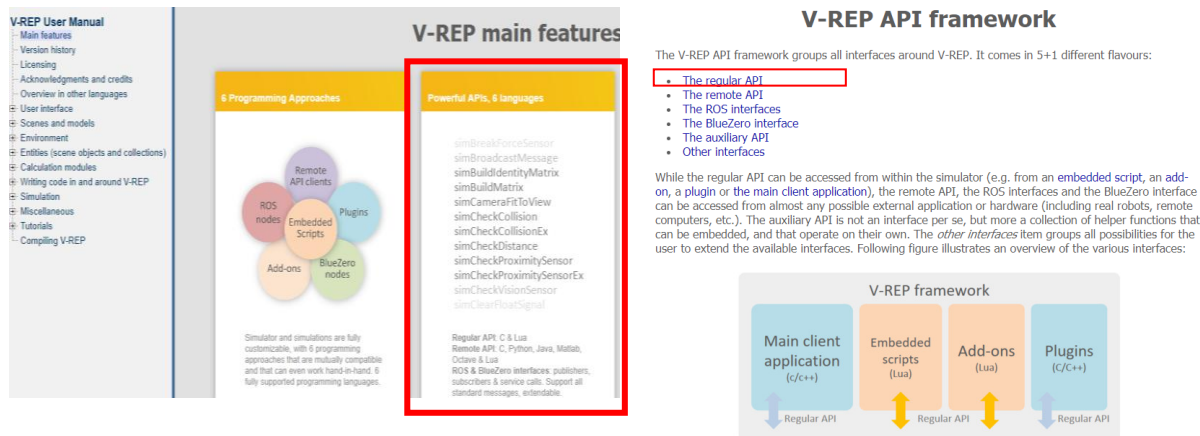
```
celija za bojanje robotom - rendering: 15 ms (7.9 fps) - SIMULATION STOPPED
Simulation Tools Plugins Add-ons Scenes Help
Help topics
About V-REP PRO EDU...
Credits...
Debug

Celija za bojanje robotom
Threaded child script (target)

1  -- if you wish to execute code contained in an external file instead,
2  -- use the require-directive, e.g.:
3  --
4  -- require 'myExternalFile'
5  --
6  -- Above will look for <V-REP executable path>/myExternalFile.lua or
7  -- <V-REP executable path>/lua/myExternalFile.lua
8  -- (the file can be opened in this editor with the popup menu over
9  -- the file name)
10
11 function sysCall_threadmain()
12     -- Put some initialization code here
13
14
15     -- Put your main loop here, e.g.:
16     --
17     -- while sim.getSimulationState() ~= sim.simulation_advancing_abouttostop do
18     --     local p=sim.getObjectPosition(objHandle,-1)
19     --     p[1]=p[1]+0.001
20     --     sim.setObjectPosition(objHandle,-1,p)
21     --     sim.switchThread() -- resume in next simulation step
22     -- end
23 end
24
25 function sysCall_cleanup()
26     -- Put some clean-up code here
27 end
28
```

Slika 4.26 Izgled skripte u koju je potrebno upisati naredbe

U izborniku se odabere (Main features→ Powerful APIs, 6 languages→ V-REP API framework→ The regular API→ functions).



Slika 4.27 V- REP korisnički priručnik sa uputstvima

Tada se otvara prozor sa svim funkcijama i parametrima koji se koriste kod Lua programiranja u programskome alatu V- REP.

Regular API function list (alphabetical order)

The list of API functions below allows you to access many V-REP parameters. There are however too many parameters in V-REP to have a specific API function for each one of them. *Auxiliary* parameters can be accessed via a set of given functions that use *object* parameter IDs. Refer also to the global parameter IDs.

```

sim.addBanner (Lua)
sim.addDrawingObject (Lua)
sim.addDrawingObjectItem (Lua)
sim.addForce (Lua)
sim.addForceAndTorque (Lua)
sim.addGhost (Lua)

sim.addObjectToCollection (Lua)
sim.addObjectToSelection (Lua)
sim.addParticleObject (Lua)
sim.addParticleObjectItem (Lua)
sim.addScript (Lua)
sim.addStatusBarMessage (Lua)

sim.adjustView (Lua)

sim.announceSceneContentChange (Lua)

sim.addBanner (C/C++)
sim.addDrawingObject (C/C++)
sim.addDrawingObjectItem (C/C++)
sim.addForce (C/C++)
sim.addForceAndTorque (C/C++)
sim.addGhost (C/C++)
sim.addModuleMenuItem (C/C++)
sim.addObjectToCollection (C/C++)
sim.addObjectToSelection (C/C++)
sim.addParticleObject (C/C++)
sim.addParticleObjectItem (C/C++)
sim.addScript (C/C++)
sim.addStatusBarMessage (C/C++)
sim.adjustRealTimeTimer (C/C++)
sim.adjustView (C/C++)
sim.advanceSimulationByOneStep (C/C++)
sim.announceSceneContentChange (C/C++)
sim.applyTexture (C/C++)

sim.getJointPosition (Lua)
sim.getJointTargetPosition (Lua)
sim.getJointTargetVelocity (Lua)
sim.getJointType (Lua)
sim.getLastError (Lua)
sim.getLightParameters (Lua)
sim.getLinkDummy (Lua)

sim.getMatchingPersistentDataTags (Lua)
sim.getModelProperty (Lua)
sim.getModuleInfo (Lua)
sim.getModuleName (Lua)
sim.getNameSuffix (Lua)
sim.getNavigationMode (Lua)
sim.getObjectAssociatedWithScript (Lua)
sim.getObjectConfiguration (Lua)
sim.getObjectChild (Lua)
sim.getObjectFloatParameter (Lua)
sim.getObjectHandle (Lua)
sim.getObjectInt32Parameter (Lua)
sim.getObjectMatrix (Lua)
sim.getObjectName (Lua)
sim.getObjectOrientation (Lua)
sim.getObjectParent (Lua)
sim.getObjectPosition (Lua)
sim.getObjectProperty (Lua)
sim.getObjectQuaternion (Lua)
sim.getObjects (Lua)
sim.getObjectSelection (Lua)

sim.getObjectsInTree (Lua)
sim.getObjectSizeFactor (Lua)
sim.getObjectSizeValues (Lua)
sim.getObjectSpecialProperty (Lua)

```

Slika 4.28 V- REP funkcije za skriptu

Obrišu se zapisi u skripti (označi se sve i delete), sad imamo novu praznu skriptu. Prvo je potrebno unijeti u skriptu targeta, da za putanju dohvati objekt izrađenog puta kojeg smo izradili kroz 3 kontrolne točke po imenu „Path“.

U V- REP funkcijama (potraži se **sim.getObjectHandle** sa Lua programskim jezikom→ otvara se prozor funkcije→ iskopira Lua synopsis → zaljepi u skriptu „target“ i dohvaćeni objekt nazove se prema objektu u kojem je prethodno izrađen put s 3 točke „Path“).

simGetObjectHandle / sim.GetObjectHandle	
Description	Retrieves an object handle based on its name. The operation of this function depends on the current name suffix settings (see sim.getNameSuffix , sim.setNameSuffix , and the section on accessing general-type objects). See also sim.isHandleValid and sim.getObjectUniqueIdentifier .
C synopsis	simInt sim.GetObjectHandle(const simChar* objectName)
C parameters	objectName : name of object. If the name is appended by a "@alt" suffix, then the object handle based on the object's alternative name will be retrieved. If the name is appended by a "@silentError" suffix, then no error will be output if the object does not exist.
C return value	handle of object or -1 if operation was not successful
Lua synopsis	number objectHandle=sim.GetObjectHandle(string objectName)
Lua parameters	Same as C-function
Lua return values	Same as C-function
Remote API equiv.	B0-based remote API: sim.GetObjectHandle Legacy remote API: sim.GetObjectHandle

Slika 4.29 V- REP funkcija za dohvaćanje objekta

Zatim je potrebno udružiti objekt „target“ sa skriptom, (preko V-REP funkcije **sim.GetObjectAssociatedWithScript** → kopira se Lua synopsis i zaljepi u drugi blok u skripti → u zagradi (number scriptHandle) upiše se „sim.handle_self“ za rukovanje trenutne skripte).

sim.GetObjectAssociatedWithScript / sim.GetObjectAssociatedWithScript	
Description	Retrieves the handle of the object the script is attached to. See also sim.getScriptAssociatedWithObject , sim.getCustomizationScriptAssociatedWithObject and sim.associateScriptWithObject .
C synopsis	simInt sim.GetObjectAssociatedWithScript(simInt scriptHandle)
C parameters	scriptHandle : handle of the script
C return value	Handle of the object that is associated with the script, or -1 if no object is associated with the script, or in case of an error.
Lua synopsis	number objectHandle=sim.GetObjectAssociatedWithScript (number scriptHandle)
Lua parameters	scriptHandle : handle of the script, or sim.handle_self for the handle of the current script
Lua return values	objectHandle : handle of the object that the script is associated with, or -1 if the script is not associated (e.g. main scripts or add-ons don't have associated objects) or in case of an error.

Slika 4.30 Funkcija za udruživanje objekta sa skriptom

Na kraju je potrebno da se objekt „target“ pomiče duž putanje „path“ za to će se upotrijebiti funkcija **sim.followPath** iz liste funkcija.

(Iskopira se Lua synopsis → zaljepi u skriptu u zadnji blok operacije).

sim.FollowPath / sim.followPath	
Description	Moves an object along a path object. This function can only be called from child scripts running in a thread (since this is a blocking operation) and is not available from the C-API. See also sim.rmlPos , sim.rmlVel and sim.moveToObject .
C synopsis	-
C parameters	-
C return value	-
Lua synopsis	number deltaTimeLeft=sim.followPath(number objectHandle,number pathHandle,number positionAndOrientation,number relativeDistanceOnPath,number velocity,number accel)
Lua parameters	objectHandle : handle of the object to be moved pathHandle : handle of the path object positionAndOrientation : a value between 1 and 3 (1: only position is modified, 2: only orientation is modified, 3: position and orientation is modified). Can be nil in which case 3 is applied. relativeDistanceOnPath : a value between 0 and 1, where 0 is the beginning of the path, and 1 the end of the path. Make sure you selected the appropriate path length calculation method (refer to the path position calculation method section). velocity : movement nominal velocity. accel : the acceleration/deceleration.
Lua return values	deltaTimeLeft : if the time needed to follow the path is not a multiple of the simulation time step, deltaTimeLeft is the execution time left at current simulation time. deltaTimeLeft is also memorized internally on a thread-basis and used as compensation or correction factor in subsequent blocking commands. deltaTimeLeft is nil in case of an error.

Slika 4.31 Funkcija za pomicanje duž putanje

U `sim.followPath` funkciji potrebno je uvesti parametre, pa tako umjesto:

„**number objectHandle**“ – objekt koji se premješta upisuje se „**Target**“

„**number pathHandle**“ - rukovanje objekta putanje upisuje se „**path**“

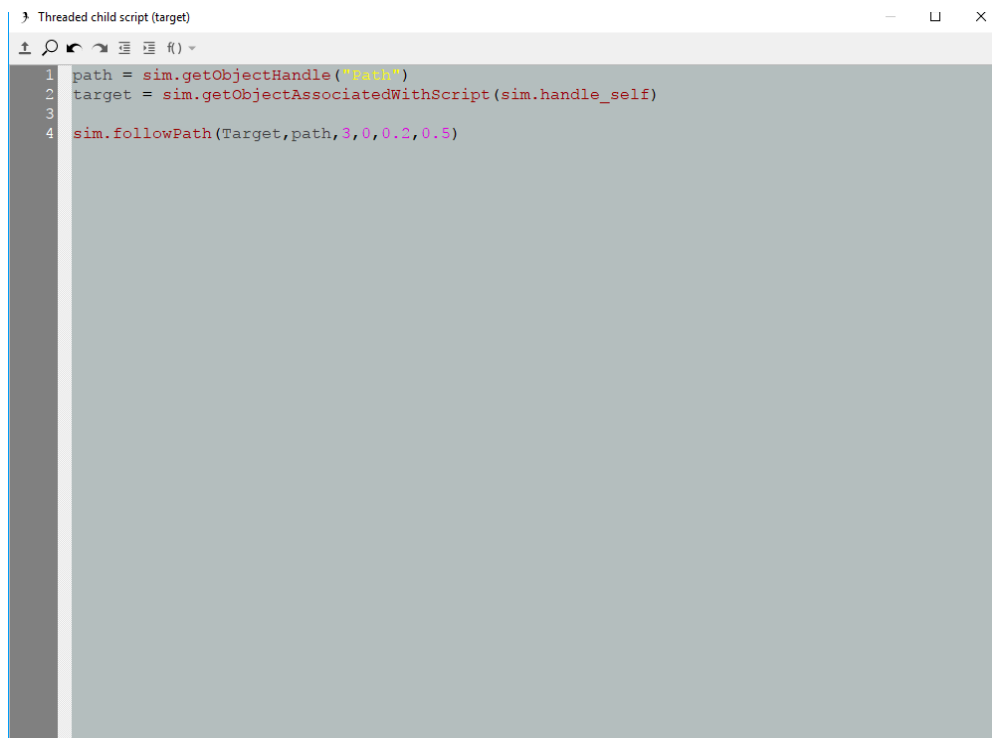
„**number positionAndOrientation**“ - ako su položaj i orijentacija promijenjeni upisuje se „**3**“

„**number relativeDistanceOnPath**“ - relativna udaljenost na putanju, pošto je početak puta upisuje se „**0**“

„**number velocity**“ - nominalna brzina kretanja upisuje se „**0.2**“ m/s

„**number accel**“ - ubrzanje upisuje se „**0.5**“ m/s²

Kad se završi sa upisom parametara i naredbi dobije se gotova skripta „targeta“ .



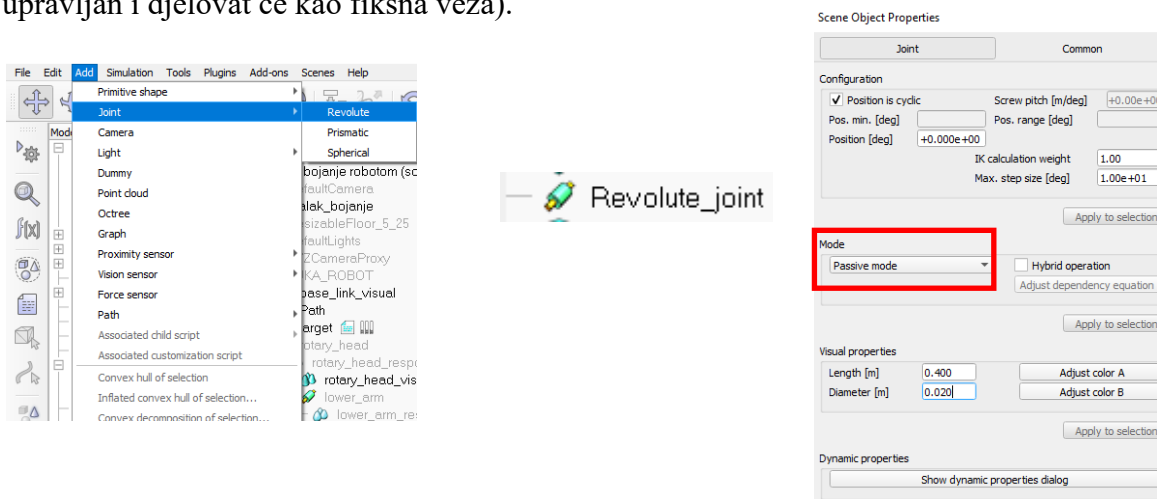
```
1 path = sim.getObjectHandle("Path")
2 target = sim.getObjectAssociatedWithScript(sim.handle_self)
3
4 sim.followPath(Target,path,3,0,0.2,0.5)
```

Slika 4.32 Skripta „target-a“ sa funkcijama i parametrima

4.6 Postavljanje rotacijskog pozicionera

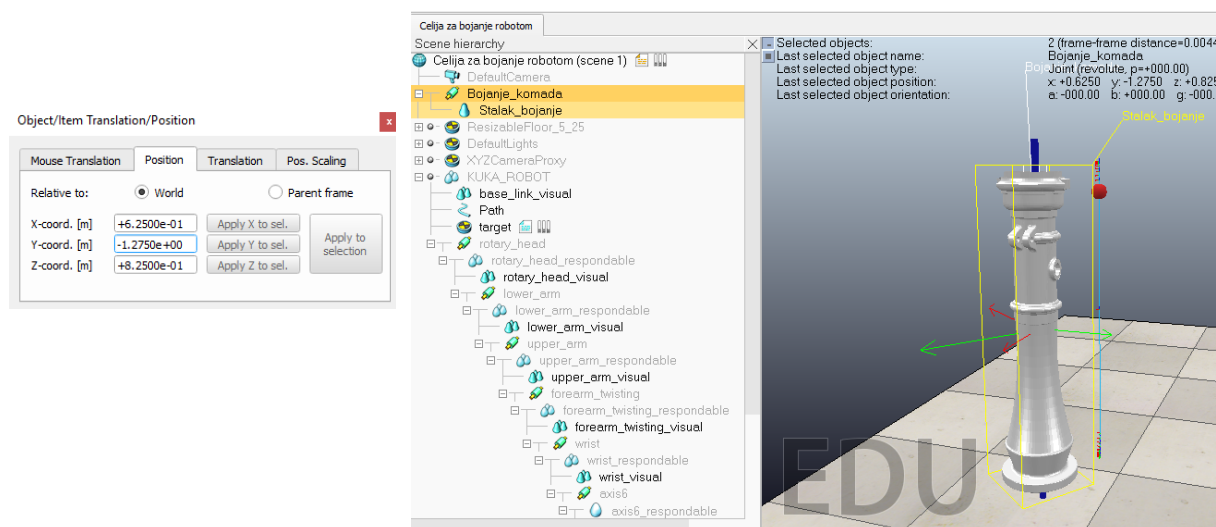
Uz robot potreban je i pozicioner sa steznim napravama za pričvršćivanje radnog komada, te će oni zajedno rotirati oko z- osi.

Odabire se rotacijsko gibanje pozicionera (Add → Joint → Revolute → Dvoklikom na „Revolute_joint“ otvaraju se postavke → odabere se „Passive mode“, jer spoj nije izravno upravljani i djelovat će kao fiksna veza).



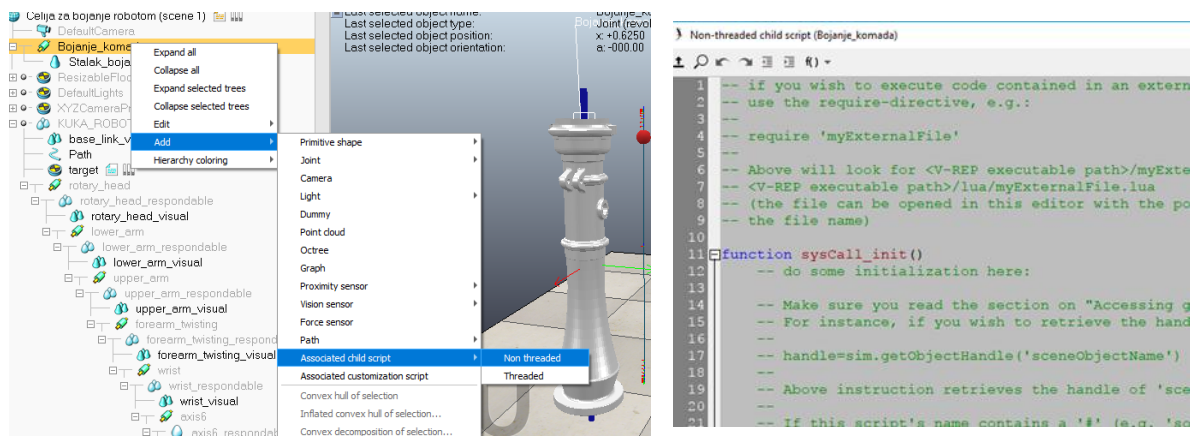
Slika 4.33 Dodavanje pozicionera

Sa operacijom „Object position“ dovede se pozicioner u poziciju radnog komada stalka. Radni komad stalka prenese se u stablu scene pozicioneru, te time radni komad postaje ovisan o vrtnji pozicionera koji mu je nadređeni.



Slika 4.34 Veza radnog komada i pozicionara

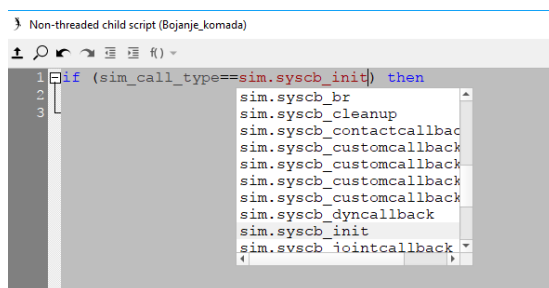
Da bi postavili zajedničku vrtnju radnog komada i pozicionera oko z- osi, potrebno je postaviti skriptu pozicioneru (Add→ Associated child script→ Non threaded).



Slika 4.35 Dodavanje skripte pozicioneru

Početni korak je da se provjeri da li je prvo inicijalno pokretanje. Da bi se omogućilo pokretanje i upravljanje simulacijom potreban je osnovni kod koji Lua poziva iz glavne skripte. Funkcija **sysCall_init** je kod koji je zadužen za pripremu simulacije i izvodi se točno jednom, prvi put kad se izvršava ova skripta.

(V-REP User Manual→ 6 Programming Approaches→ main script→ upiše se vrsta poziva **init** koja se potraži u padajućem izborniku).



Slika 4.36 Provjera inicijalnog pokretanja

Kad se provjeri da li je prvo inicijalno pokretanje, potrebno je dohvatiti objekt koji je povezan sa skriptom.

(V- REP User Manual→ Main features→ Powerful APIs, 6 languages→ V-REP API framework→ The regular API→ functions→ **sim.getObjectAssociatedWithScript** (Lua)).

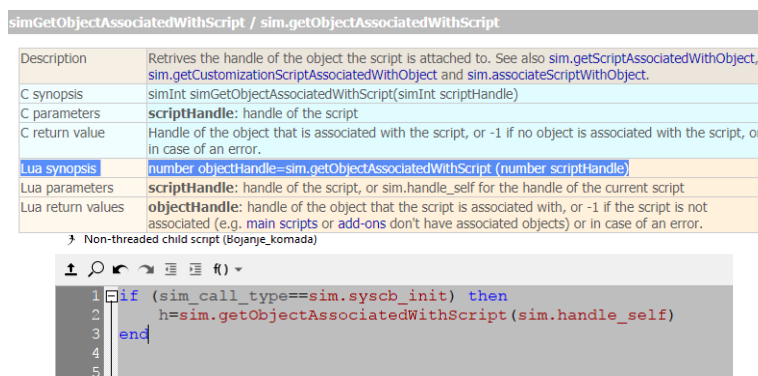
```

sim.getMatchingPersistentDataTags (Lua)
sim.getModelProperty (Lua)
sim.getModuleInfo (Lua)
sim.getModuleName (Lua)
sim.getNameSuffix (Lua)
sim.getNavigationMode (Lua)
sim.getObjectAssociatedWithScript (Lua)
sim.getObjectConfiguration (Lua)
sim.getObjectChild (Lua)

```

Slika 4.37 Funkcija za dohvaćanje objekta povezanog sa skriptom

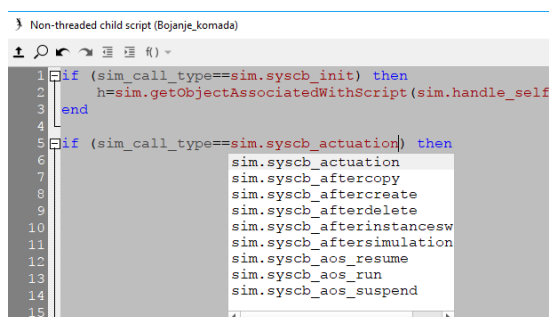
Iskopira se Lua synopsis u drugi blok skripte, upiše se ime parametra za rukovanje trenutne skripte „sim.handle_self“. I tu je kraj prvog djela naredbi.



Slika 4.38 Upis parametara za dohvaćanje objekta sa skriptom

U drugom djelu naredbi potreban je kod **sysCall_actuation** koji je općenito odgovoran za rukovanje svim funkcijama pokretanja simulatora (inverzna kinematika, dinamika itd.). Ovaj dio se također izvodi u svakom simulacijskom prolazu. Prvo se provjeri da li je inicijalizacija izvršena odnosno aktivirana, ako je izvršena nastavlja se dalje.

(V- REP User Manual → 6 Programming Approaches → main script → upiše se vrsta poziva **actuation** koji se potraži u padajućem izborniku)



Slika 4.39 Provjera aktivacije simulacije

Definiraju se lokalne varijable ključnom riječi **local**, te je potrebno dohvatiti vremenski korak simulacije (virtualno vrijeme simulacije koje prolazi pri svakom prolasku simulacije glavne skripte).

(V- REP User Manual → Main features → Powerful APIs, 6 languages → V-REP API framework → The regular API → functions → `sim.getSimulationTimeStep` (Lua)). Sa tom funkcijom postavi se vremenski korak simulacije.

simGetSimulationTimeStep / sim.getSimulationTimeStep	
Description	Retrieves the simulation time step (the simulation time (i.e. not real-time) that passes at each main script simulation pass). This value might not be constant for a given simulation.
C synopsis	simFloat simGetSimulationTimeStep()
C parameters	None
C return value	negative value (-1.0) if operation not successful, otherwise the simulation time step
Lua synopsis	number timeStep=getSimulationTimeStep()
Lua parameters	Same as C-function
Lua return values	Same as C-function
Remote API equiv.	BO-based remote API: simxGetSimulationTimeStep Legacy remote API: -

Non-threaded child script (Bojanje_komada)

```

1 if (sim_call_type==sim.syscb_init) then
2   h=sim.getObjectAssociatedWithScript(sim.handle_self)
3 end
4
5 if (sim_call_type==sim.syscb_actuation) then
6   local dt=sim.getSimulationTimeStep()
7

```

Slika 4.40 Funkcija za vremenski korak simulacije

Druga lokalna varijabla biti će da se dohvati unutarnja vrijednost kuta zakretanja zgloba. Manipulacija objekta će biti jednaka manipulaciji objekta zgloba koji je prethodno udružen sa skriptom („h“).

(V- REP User Manual → Main features → Powerful APIs, 6 languages → V-REP API framework → The regular API → functions → sim.getJointPosition (Lua)).

simGetJointPosition / sim.getJointPosition	
Description	Retrieves the intrinsic position of a joint. This function cannot be used with spherical joints (use sim.getJointMatrix instead). See also sim.setJointPosition .
C synopsis	simInt simGetJointPosition(simInt objectHandle, simFloat* position)
C parameters	objectHandle : handle of the joint position : intrinsic position of the joint. This is a one-dimensional value: if the joint is revolute, the rotation angle is returned, if the joint is prismatic, the translation amount is returned, etc.
C return value	-1 if operation was not successful. In a future release, a more differentiated return value might be available
Lua synopsis	number position= sim.getJointPosition (number objectHandle)
Lua parameters	objectHandle : handle of the joint
Lua return values	position : intrinsic position of the joint or nil in case of an error
Remote API equiv.	BO-based remote API: simxGetJointPosition Legacy remote API: simxGetJointPosition

Non-threaded child script (Bojanje_komada)

```

1 if (sim_call_type==sim.syscb_init) then
2   h=sim.getObjectAssociatedWithScript(sim.handle_self)
3 end
4
5 if (sim_call_type==sim.syscb_actuation) then
6   local dt=sim.getSimulationTimeStep()
7   local p=sim.getJointPosition(h)
8

```

Slika 4.41 Funkcija za zakretanje zgloba

Upotrebom te dvije funkcije potrebna je formula koja izračunava novu poziciju na način da se povećava vrijednost kuta zakretanja zgloba u nekom vremenu pri brzini od 0.07m/s.

Non-threaded child script (Bojanje_komada)

```

1 if (sim_call_type==sim.syscb_init) then
2   h=sim.getObjectAssociatedWithScript(sim.handle_self)
3 end
4
5 if (sim_call_type==sim.syscb_actuation) then
6   local dt=sim.getSimulationTimeStep()
7   local p=sim.getJointPosition(h)
8   p=p+dt*0.07
9

```

Slika 4.42 Formula promjene pozicije

Također potrebno je postaviti unutarnji položaj zakretanja zgloba.

(V-REP User Manual → Main features → Powerful APIs, 6 languages → V-REP API framework → The regular API → functions → `sim.setJointPosition` (Lua)).

Kopira se Lua synopsis u skriptu, upiše se objekt koji je povezan sa skriptom (number objectHandle – „h“). Položaj zgloba (kutna ili linearna vrijednost ovisno o vrsti spoja) (number position- „p“).

simSetJointPosition / sim.setJointPosition	
Description	Sets the intrinsic position of a joint. May have no effect depending on the joint mode. This function cannot be used with spherical joints (use <code>sim.setSphericalJointMatrix</code> instead). See also <code>sim.getJointPosition</code> and <code>sim.setJointTargetPosition</code> .
C synopsis	<code>simInt simSetJointPosition(simInt objectHandle, simFloat position)</code>
C parameters	objectHandle : handle of the joint object position : position of the joint (angular or linear value depending on the joint type)
C return value	-1 if operation was not successful. In a future release, a more differentiated return value might be available
Lua synopsis	<code>number result = sim.setJointPosition(number objectHandle, number position)</code>
Lua parameters	Same as C-function
Lua return values	Same as C-function
Remote API equiv.	B0-based remote API: <code>simxSetJointPosition</code> Legacy remote API: <code>simxSetJointPosition</code>

Slika 4.43 Funkcija postavljanja unutarnjeg položaja

Skripta se završava naredbom „end“, te se zatvori prozor. (Prilog)

Završetkom programiranja skripte kod pozicionera, potrebno je isprobati i vidjeti u simulaciji da li sve radi.

Ukoliko je potrebno, povećati ili smanjiti brzinu vrtnje radnog komada, da nebi neki dijelovi ostali neobjanjeni zbog prebrze rotacije.

```
Non-threaded child script (Bojanje_komada)
1 if (sim_call_type==sim.syscb_init) then
2   h=sim.getObjectAssociatedWithScript(sim.handle_self)
3   end
4
5 if (sim_call_type==sim.syscb_actuation) then
6   local dt=sim.getSimulationTimeStep()
7   local p=sim.getJointPosition(h)
8   p=p+dt*0.07
9   sim.setJointPosition(h,p)
10 end
```

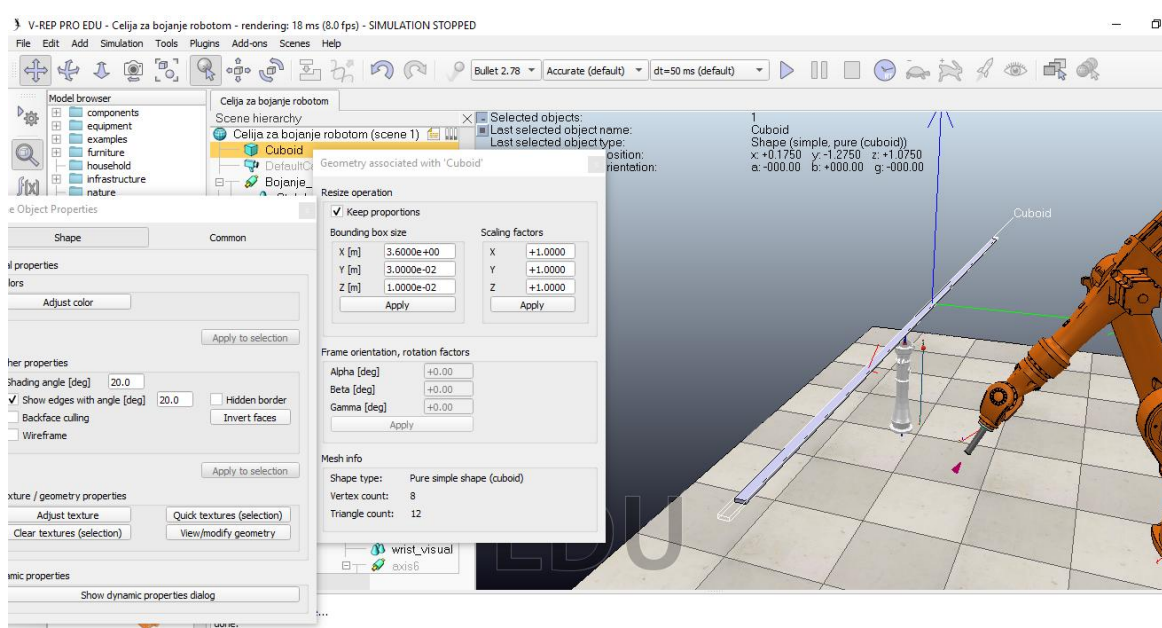
Slika 4.44 Završni izgled skripte pozicionera

4.7 Prikazi dijelova postrojenja za bojanje

4.7.1 Vješalica radnih komada stalka

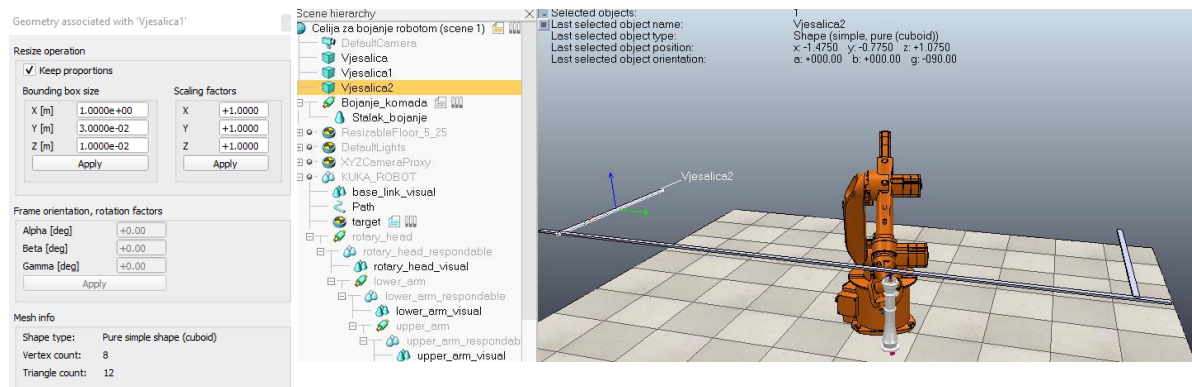
Ideja je da se radni komadi transportiraju po vodilici od sačmare do sušare. Tom prilikom komadi se vjese na vješalicu prije ulaska u sačmaru, nakon sačmarenja komad izlazi preko vodilica koje ga vode do robota.

Izradi se prvo duža vješalica (Add → Primitive shape → Cuboid → Odaberu se dimenzije → dvostruki klik na isto otvara se Scene Object Properties → Show dynamic properties dialog → i odznači se Body is dynamic, da tijelo nema masu i da ostane u zadanoj poziciji.)



Slika 4.45 Izrada veće vješalice

Istim principom izrade se ostale dvije manje vješalice s manjim dimenzijama, koje će služiti za izlaz radnih komada iz sačmare i ulaz u sušaru.

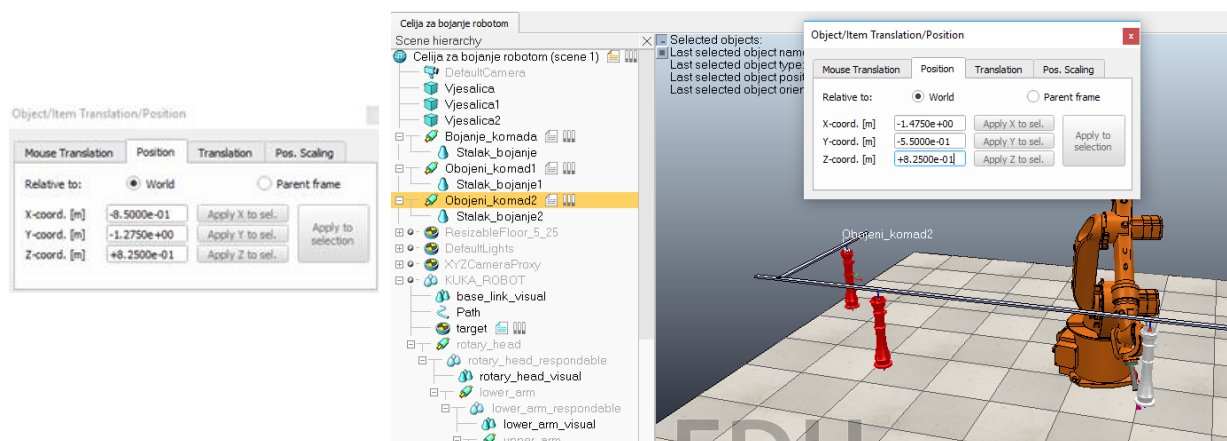


Slika 4.46 Izrada dvije manje vješalice

Nakon završetka vješalice potrebno je prikazati transport radnih komada koje su prethodno obojani robotom do sušare.

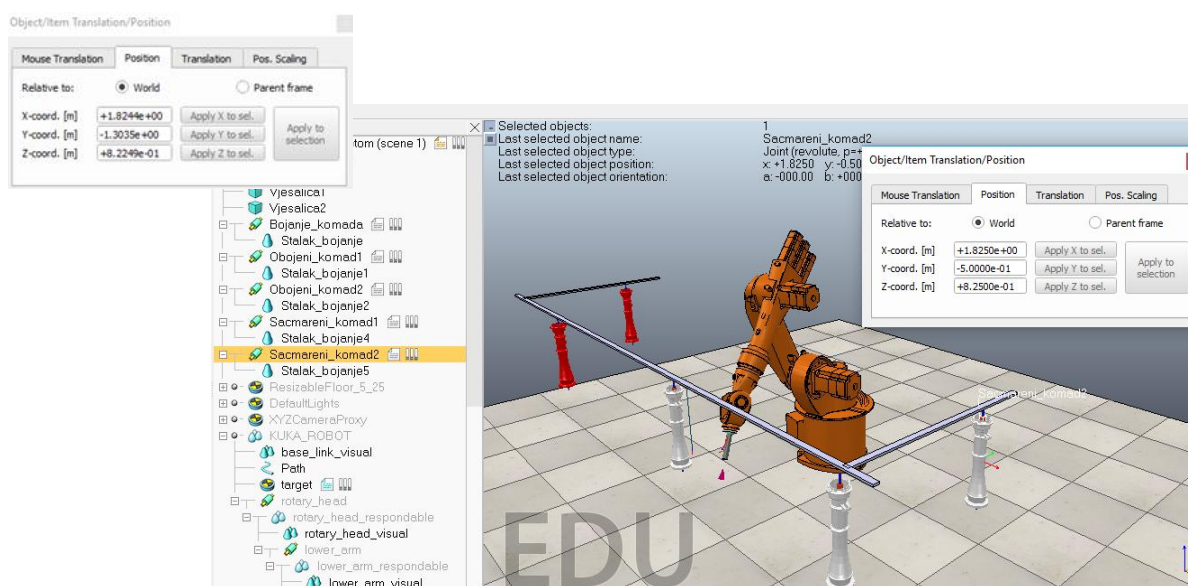
Iskopira se radni komad sa pozicionerom i promjeni mu se položaj, jedan za prikaz nakon završetka bojanja prilikom ulaska u sušaru i drugi komad u zamišljenoj sušari.

(Označi se u stablu scene „Bojanje_komada“ → drži se *ctrl* → „Stalak_bojanje“ → desni klik na označeno → Edit → Copy selected objects → desni klik u stablu Add → Paste buffer → promjeni se željeni položaj za prvi i drugi obojeni komad operacijom **Object Position**).



Slika 4.47 Položaji prvog i drugog obojenog komada

Istim principom će se napraviti i sa sačmarenim komadima (kopirati i zaljepiti), te im izmjeniti položaje. Prvi komad je za izlaz iz sačmare, a drugi je u procesu sačmarenja.

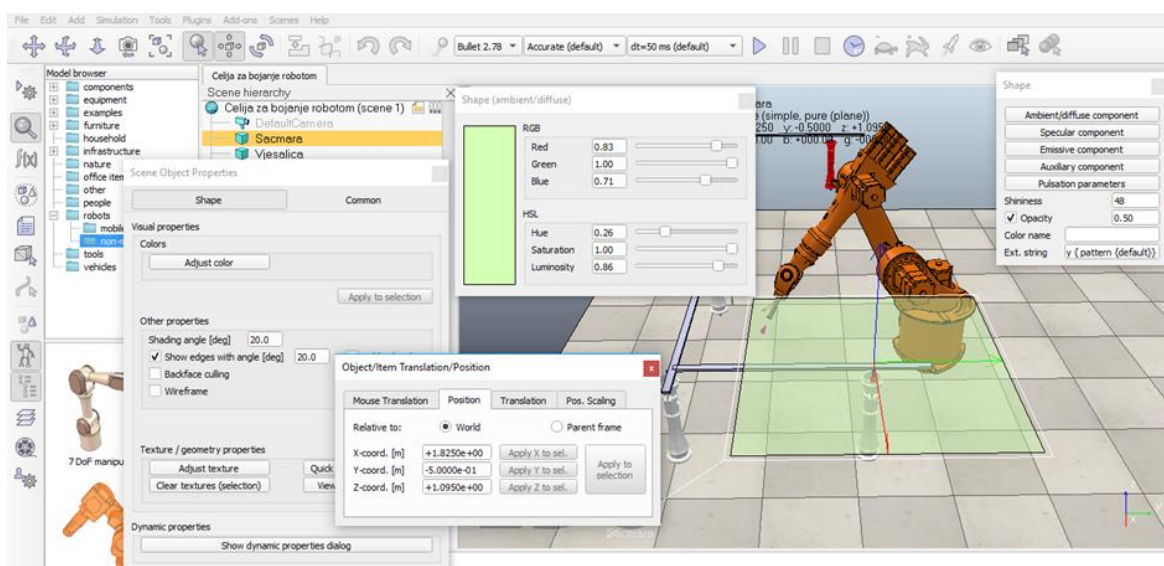


Slika 4.48 Položaji prvog i drugog sačmarenog komada

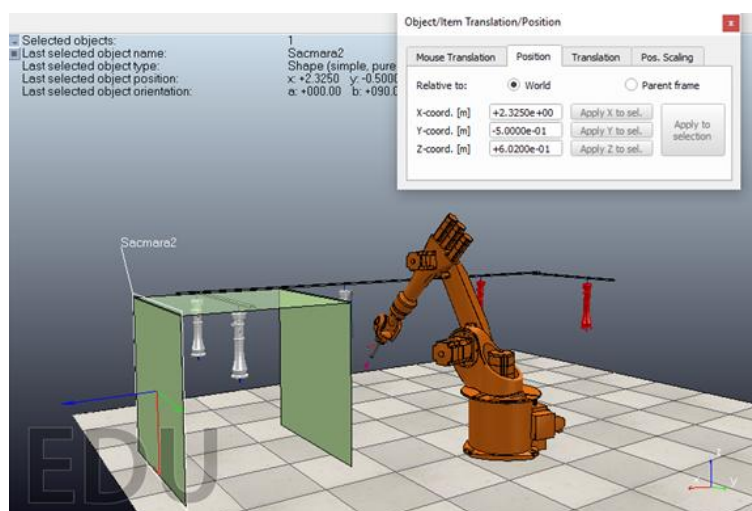
4.7.2 Sačmara za čišćenje površine radnog komada

Dodavanjem planova prikazat će se sačmara u kojoj se nalazi komad koji se prije samog postupka bojanja dobro počisti. Pomoću kuglica ili sačme čisti se vanjska površina radnog komada od nečistoća, masnoća, grubih površina...itd.

(Add→ Primitive shape→ Plane→ Upišu se dimenzije i promjeni pozicija→ Dvostrukim klikom na objekt otvaraju se postavke objekta u kojem se promjeni boja→ Adjust color→ Ambient/diffuse component i promjeni se boja→ ukoliko želimo da je objekt proziran odabere se „Opacity“).

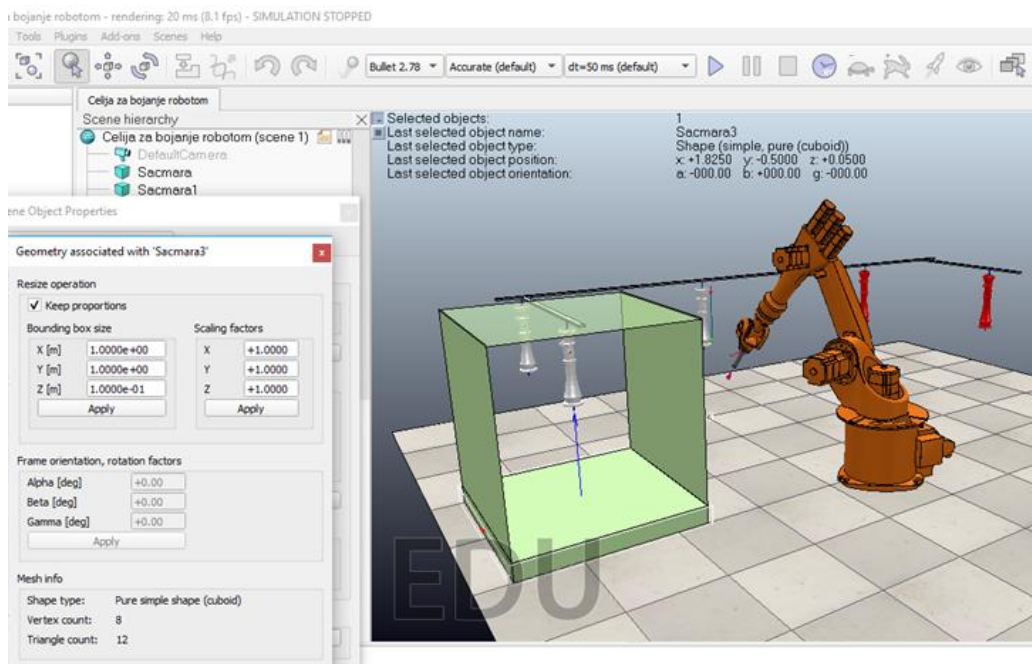


Slika 4.49 Izrada krova sačmare



Slika 4.50 Bočne stranice sačmare

Izradom poda sačmare dobiva se konačni prikaz da je radni komad u sačmari koji je prethodno obješen na vješalicu. Nakon postupka sačmarenja radni komad preko vodilica izlazi iz sačmare prema procesu bojanja.



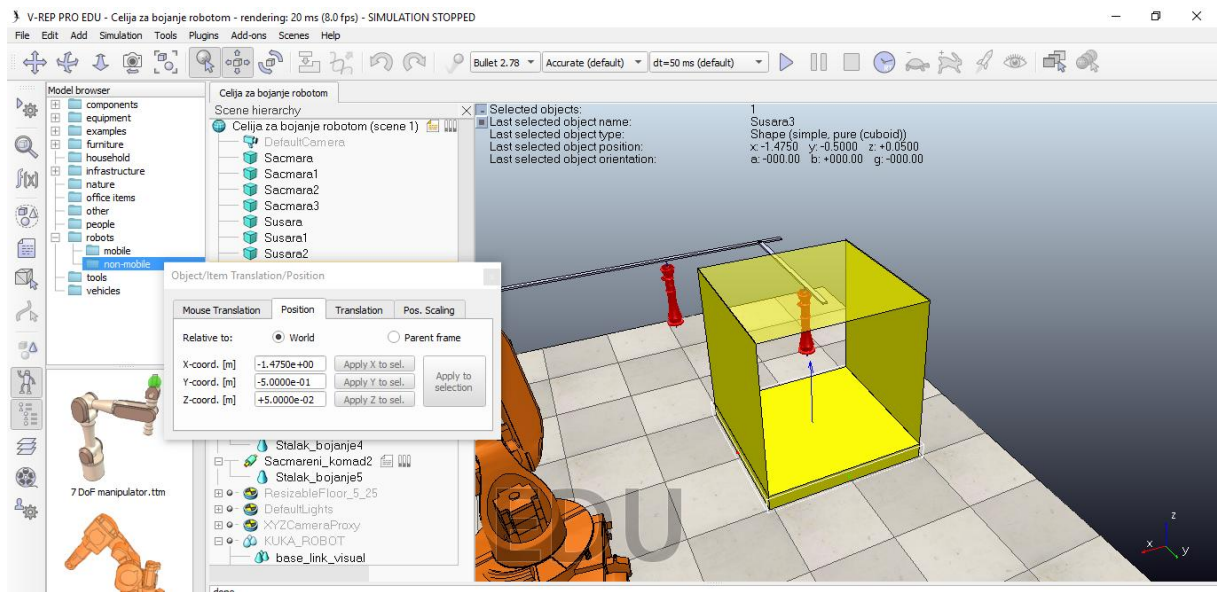
Slika 4.51 Prikaz sačmare

4.7.3 Sušara obojenih komada

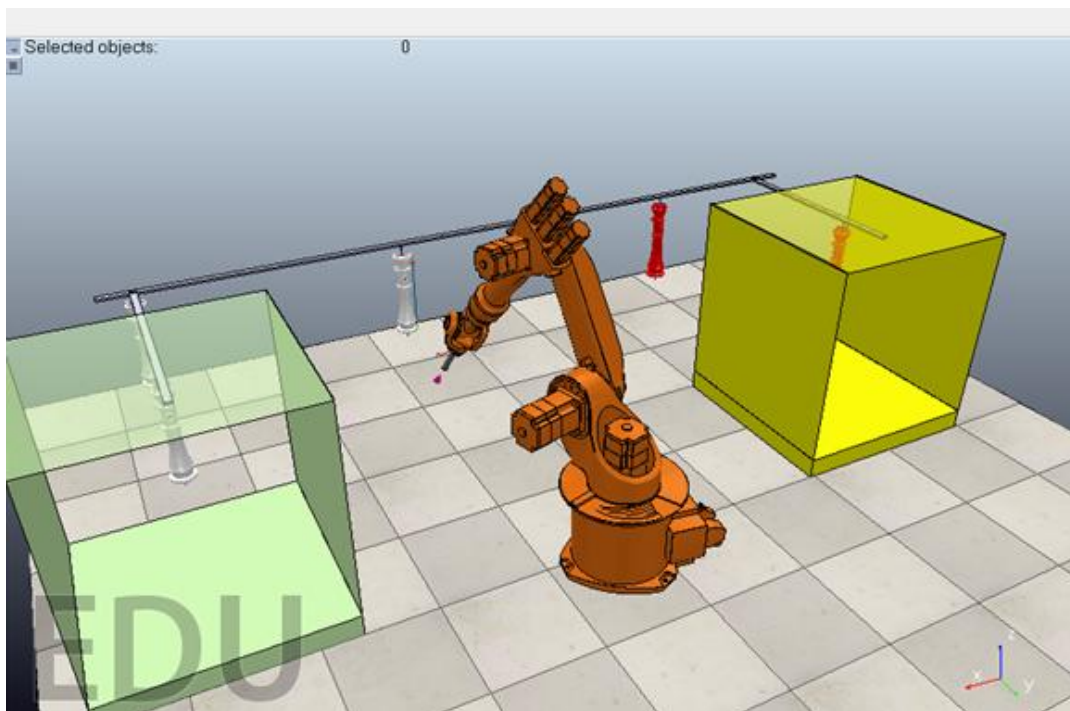
Da bi se prethodno obojani radni komad čim brže posušio, potrebna je sušara sa ventilatorom koji upuhuje topli zrak. Radni komad do sušare dolazi preko vodilica na vješalici, nakon sušenja komad se skida i odlaže na siguran i pažljiv način.

Istim principom izrađuju se planovi krova, bočnih stranica i poda kao i kod sačmare.

(Add→ Primitive shape→ Plane→ Upišu se dimenzije i promjeni pozicija→ Dvostrukim klikom na objekt otvaraju se postavke objekta u kojem se promjeni boja→ Adjust color→ Ambient/diffuse component i promjeni se boja→ ukoliko želimo da je objekt proziran odabere se „Opacity“.)



Slika 4.52 Prikaz sušare

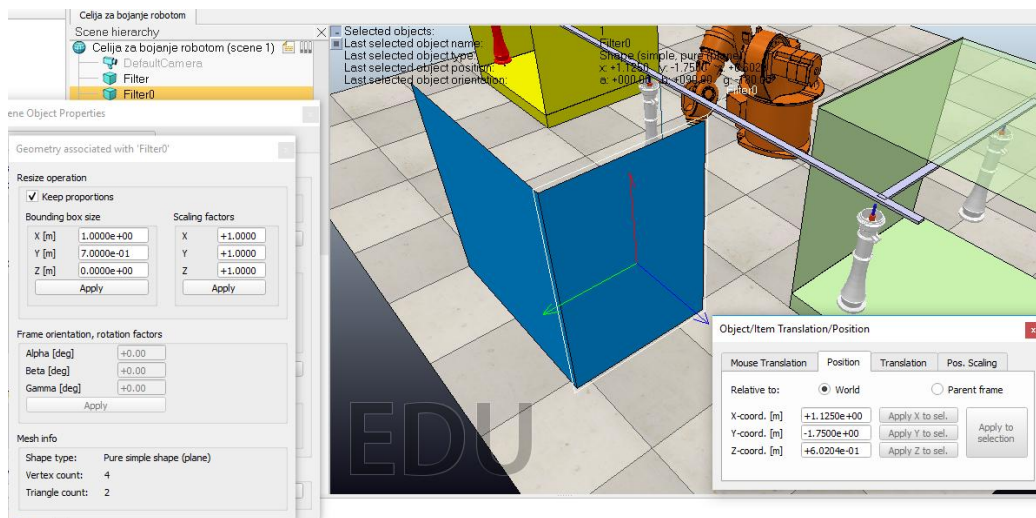


Slika 4.53 Položaji sačmare i sušare

4.7.4 Prikaz filtera

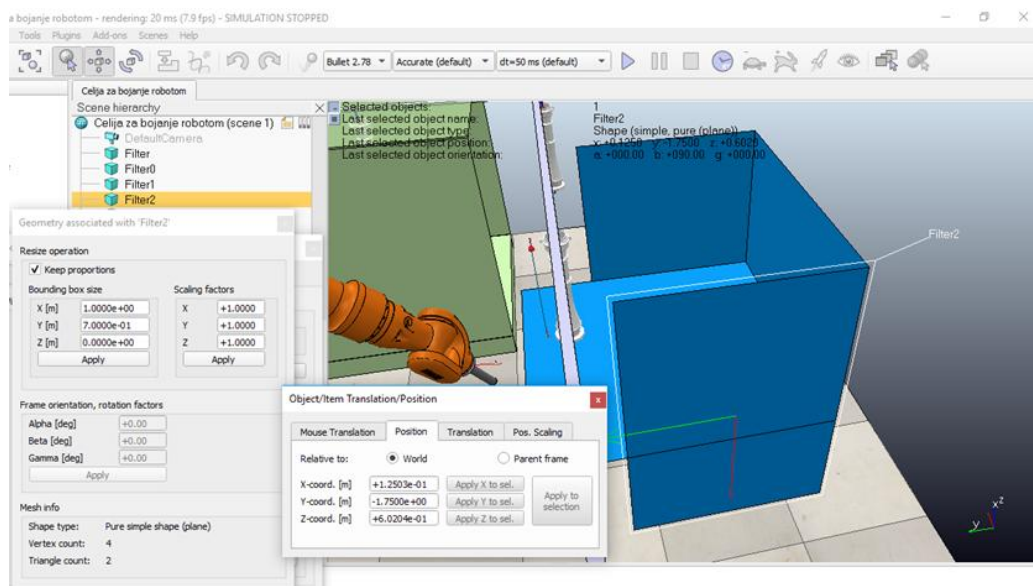
U današnje vrijeme izrazito se vodi briga o okolišu i njegovom zagađenju opasnim plinovima. Tako je i u bojanju potreban filter koji prilikom postupka usisava i zadržava štetne plinove i nedozvoljava im da izlaze u prirodu.

Prvo se izrade bočne stranice (Add→ Primitive shape→ Plane→ Upišu dimenzije i promjeni pozicija→ Zatim se zarotira stranica→ Dvostrukim klikom na objekt otvaraju se postavke objekta u kojem se promjeni boja→ Adjust color→ Ambient/diffuse component i promjeni se boja→ ukoliko želimo da je objekt proziran odabere se „Opacity“.)



Slika 4.54 Bočne stranice filtera

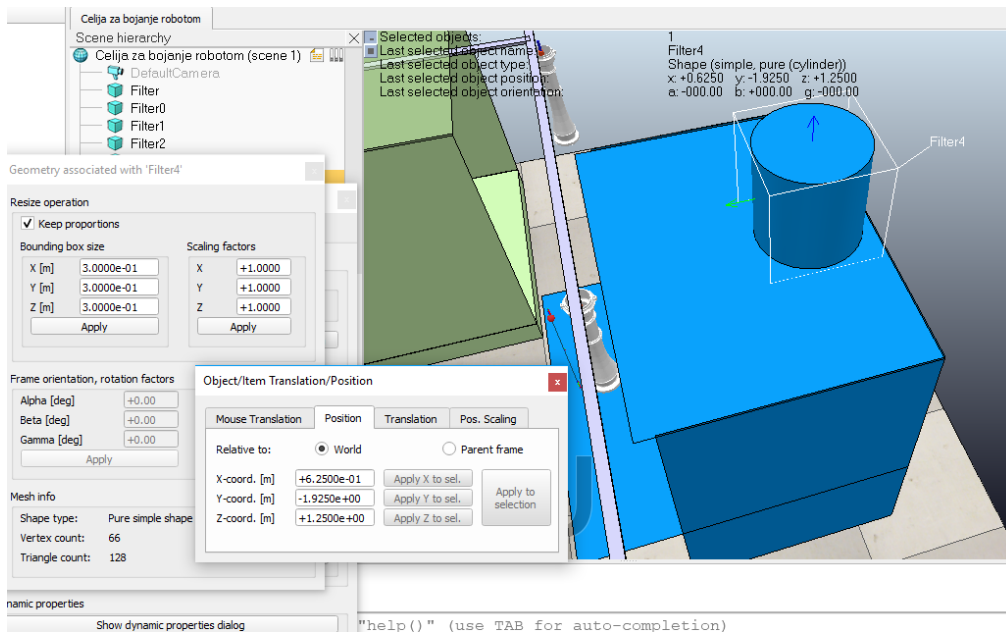
Nakon bočnih stranica izradit će se pladanj koji će biti malo veći od ostalih stranica filtera. Tako da sad sav višak boje prilikom bojanja radnog komada pada na njega, a ne oko ili po podu.



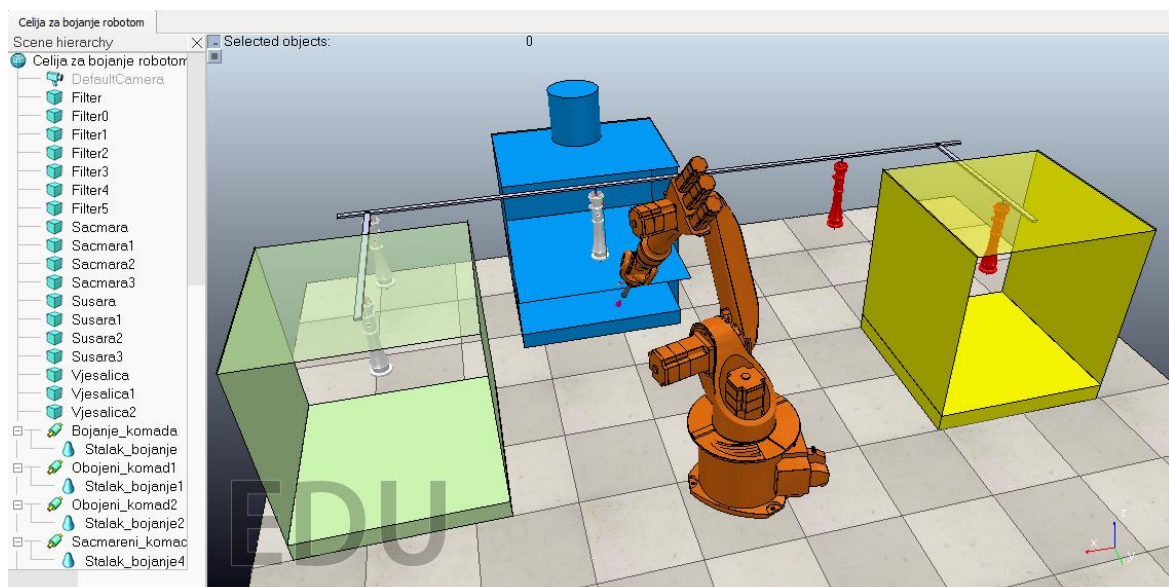
Slika 4.55 Pladanj filtera

Nakon što se postavi krov filtera, potrebno je dodati prikaz usisa filtera.

(Add → Primitive shape → Cylinder → postavke se dimenzije i pozicionira na vrh krova filtera).



Slika 4.56 Usis filtera



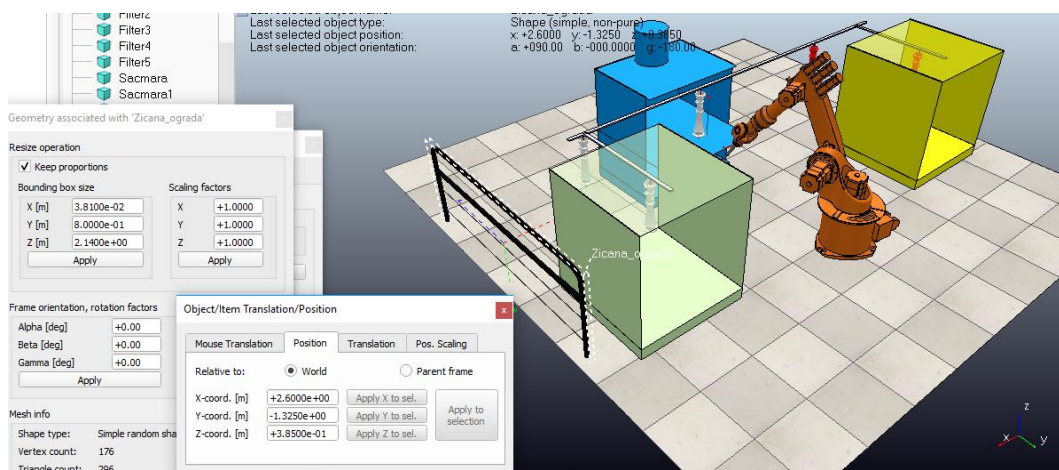
Slika 4.57 Prikaz položaja filtera

4.7.5 Sigurnosna ograda linije za bojanje

Žićana ograda kod linije za bojanje je također važna stavka kod sigurnosti. Onemogućuje pristup ljudima kad je robot u pokretu, tek nakon što robot stane može se ući u ćeliju.

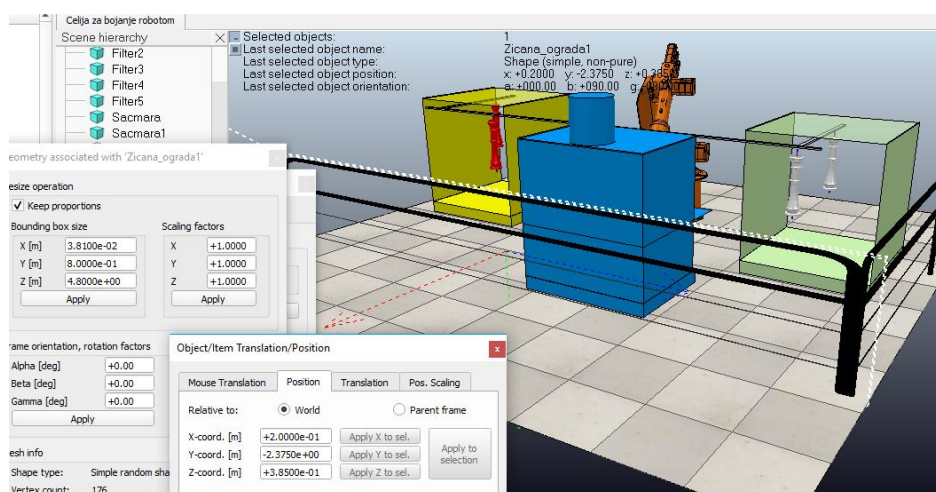
(Model browser → infrastructure → other → handrail se odabere, poslužiti će za prikaz ćelije → dvoklik na objekt ograde → Adjust color → Ambient diffuse component → promjeni se boja).

Prilagode se dimenzije, zarotira i prenese na željeno mjesto. Također iskopira se za drugu stranu, samo se promjeni pozicija.



Slika 4.58 Izrada manje ograde

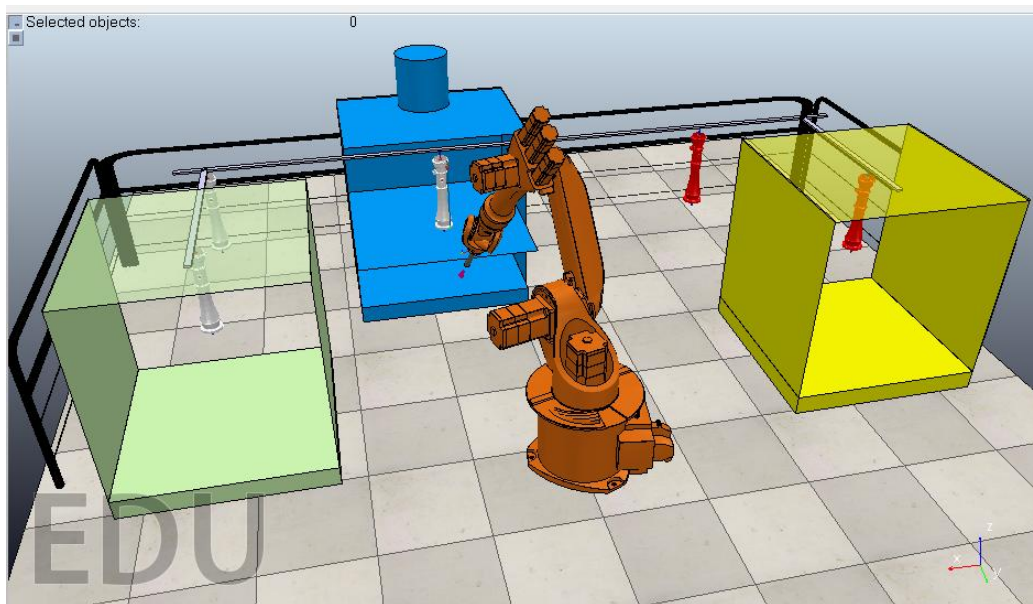
Za dužu ogradu povećavaju se samo dimenzije prethodne ograde iskopira se i zarotira.



Slika 4.59 Izrada duže ograde

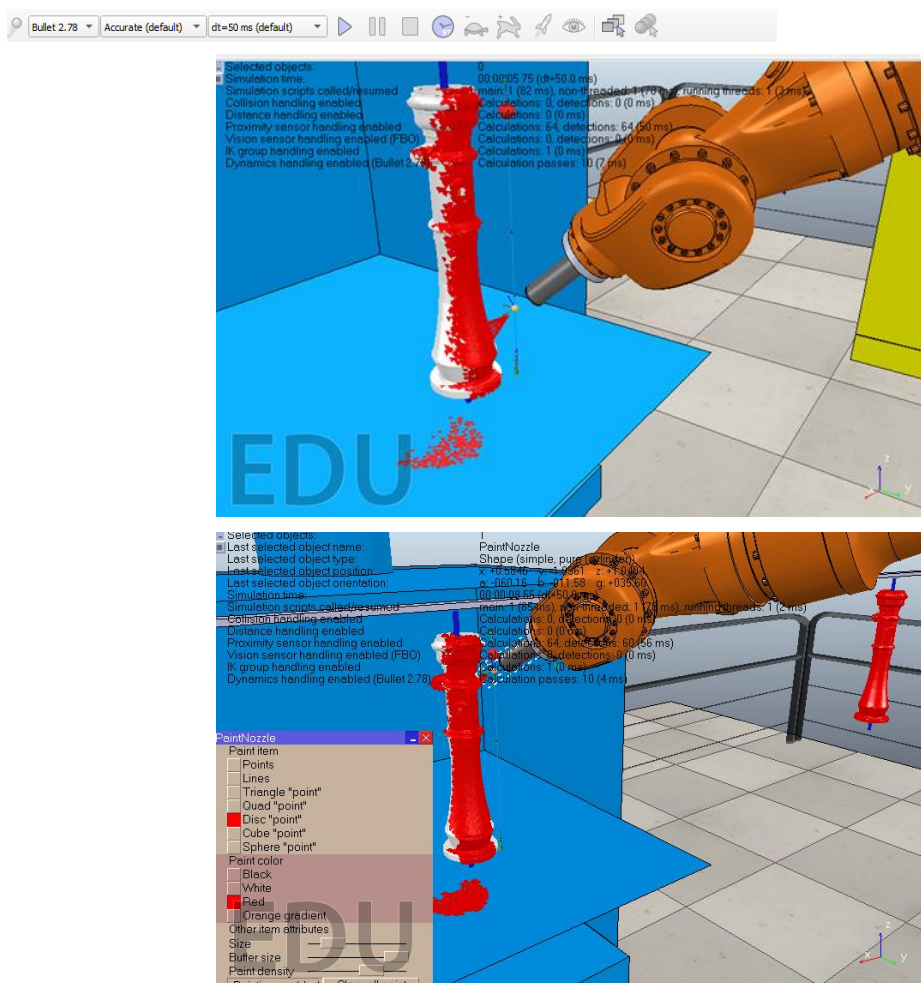
4.8 Simulacija ćelije za bojanje robotom

I nakon izrade ograde, dobiva se konačni prikaz ćelije za bojanje.



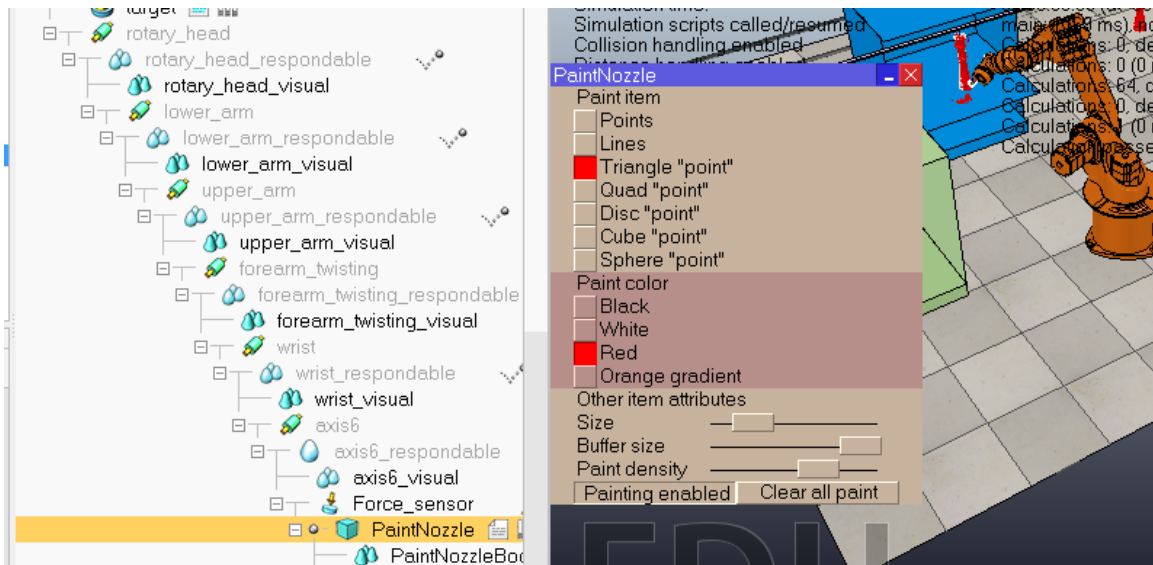
Slika 4.60 Konačni prikaz ćelije za bojanje

Sljedeće što preostaje je da se isproba simulacija „Start/resume simulation“.



Slika 4.61 Bojanje radnog komada

Nakon što se uključi simulacija klikne se na postavljeni pištolj za boju „PaintNozzle“ i otvara se izbornik sa podvrstama nanosa boje. Tijekom bojanja radnog komada moguće je mijenjati oblike raspršivača kao što su trokutasti, kockasti, točkasti, okrugli itd. Također i boju koja se želi nanjeti na radni komad i debljinu ili količinu nanosa boje.



Slika 4.62 Podešavanje vrste i količine boje

Podešavanja boje se može odvijati i kad nije pokrenuta simulacija, jer pištolj za boju dolazi sa svojom pripadajućom skriptom u kojoj se može isto tako podesiti željena boja brojčano. Postavke se odmah primjene prije simulacije i također moguće ih je mijenjati u prozoru „PaintNozzle“.

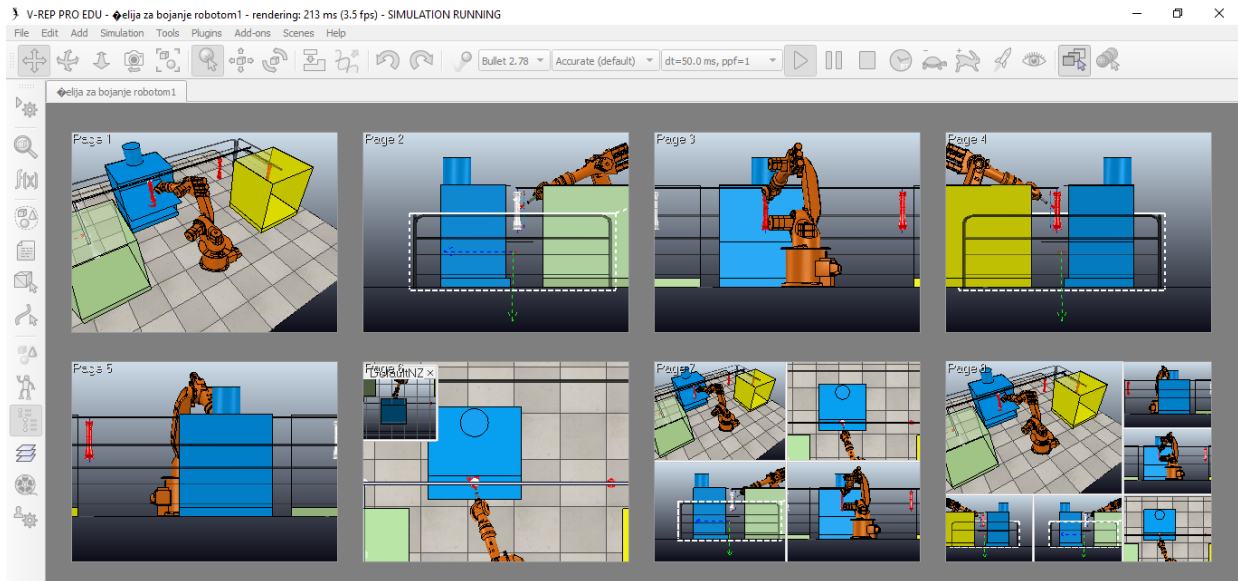
```

Non-threaded child script (PaintNozzle)
function sysCall_init()
    jetHandle=sim.getObjectHandle("PaintNozzleJet")
    joint0Handle=sim.getObjectHandle("PaintNozzleJoint0")
    joint1Handle=sim.getObjectHandle("PaintNozzleJoint1")
    dlg=simGetUIHandle("PaintNozzleDialog")
    simSetUIButtonLabel(dlg,100,sim.getScriptName(sim.handle_self))
    jetAngle=30*math.pi/180
    density=64
    objList={}
    allDrawingObjects={}
    mode=sim.drawing_trianglepoints
    colorIndex=2
    itemSize=5
    bufferSize=90000
    paintingEnabled=true
    refreshDlg=true
end
getDrawingObject=function(objectHandle)
    for i=0,(#objList-1)/2,1 do
        if (objList[2*i+1]==objectHandle) then
            return(objList[2*i+2])
        end
    end
end
table.insert(objList,objectHandle)

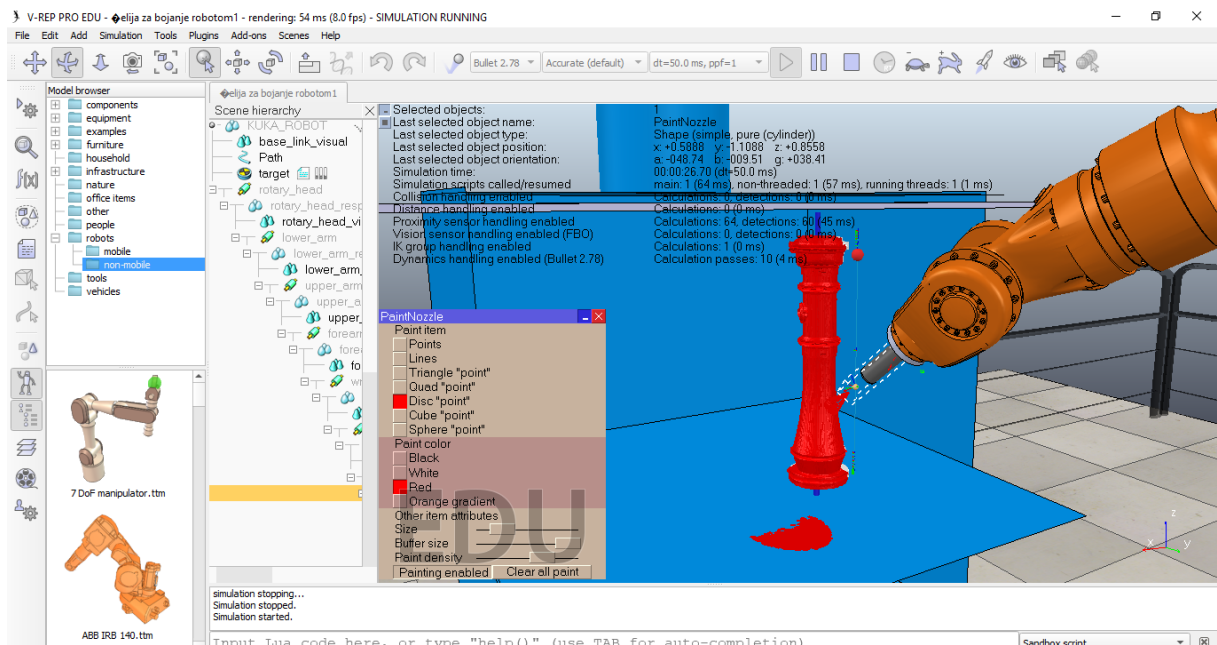
```

Slika 4.63 Postavke u skripti pištolja za boju

Klikom na tipku „Page selector“ otvaraju se više manjih prozora scene iz različitih perspektiva.



Slika 4.64 Bojanje iz različitih perspektiva



Slika 4.65 Obojani komad

5. Zaključak

Zaštita od korozije i estetski izgled proizvoda danas na tržištu igraju veliku ulogu uz funkcionalnost. Kupac ima na biranje kakvu vrstu površinske zaštite želi, većinom se odlučuju za plastificirane i bojane proizvode. Osim toga može birati da vanjska i unutarnja površina bude različite boje, te debljinu nanosa koja je minimalno 250 μ m prema propisu kvalitete. Zbog sve težeg pronalaska ljudske radne snage za izloženosti teškim uvjetima rada dolazi do sve veće potrebe za uvođenje robotizacije i automatizacije u proizvodnju. U počecima napredne tehnologije postojala je bojaznost da će uzrokovati rast nezaposlenosti, ali napredak industrijskih robota dovodi do otvaranja novih još stručnijih i kvalitetnijih radnih mjesta koja zamjenjuju ona manja kvalificirana radna mjesta.

Primjer automatizirane ćelije za bojanje robotom prikazana je simulacijom u programskome alatu V-REP koji ima široku paletu funkcija. U korisničkom priručniku svaka funkcija je objašnjena kao i svi njezini parametri. Neki od modela imaju već ugrađene funkcionalnosti, ali moguće je dodati i vlastita kontroliranja objekta preko nekoliko načina programiranja. Jedan od načina je i primjena virtualne stvarnosti koja pruža značajne prednosti i koja će u narednom periodu doseći novu razinu i sve širu primjenu na svim razinama, pa tako i u industriji. U području strojarstva se već koristi za modeliranje u novoj vizualnoj 3D percepciji u kojoj se mogu postići značajne cjenovne uštede prije izrada prototipa, povećanje sigurnosti i uštede materijala ukoliko se otkrije greška sklopa u ranoj fazi.

Dokazano je da roboti poslodavcu predstavljaju manji trošak od klasičnog radnika, kako su u obavljanju rutinskih radnji najčešće točniji od čovjeka, da poboljšavaju kvalitetu proizvodnje te minimiziraju potrošnju različitih resursa i ne stvaraju otpad. Primjer linije za bojanje može poslužiti dakako u velikoserijskoj proizvodnji jer za pojedinačnu nije isplativo. Danas je robotika ključni pokretač konkurentnosti u velikim proizvodnim industrijama.

Budućnost proizvodnje i daljnjim razvojem robotizacije sve manje će ovisiti isključivo o radnoj snazi čovjeka, te će s vremenom doći da takvi sustavi potpuno zamjene radnika u proizvodnim procesima. Strojevi će proizvoditi, ali njima će i dalje upravljati ljudi umnim radom programiranjem, nadgledanjem izvođenja programa i istraživanjem.



IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, DANIJELO TRALČEĆ (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/iza završnog/diplomskog (obrisati nepotrebno) rada pod naslovom RAZVOJ ROBOTIČKE ČELIJE ZA LAKIRANJE U PROGRAMSKOME ALATU V-REP (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

Danijele Tralčec
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, DANIJELO TRALČEĆ (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom RAZVOJ ROBOTIČKE ČELIJE ZA LAKIRANJE U PROGRAMSKOME ALATU V-REP (upisati naslov) čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)

Danijele Tralčec
(vlastoručni potpis)

6. Literatura

- [1] M. Gojić: Tehnike spajanja i razdvajanja materijala, Sveučilište u Zagrebu Metalurški fakultet, Sisak, 2003., str. 244.
- [2] http://www.hunor.hr/index.php?option=com_virtuemart&page=shop.browse&category_id=163&Itemid=176&TreeId=63, dostupno 17.08.2019.
- [3] A.Radović, Tehnologija materijala, Školska knjiga, Zagreb, 1971., str. 101.
- [4] V. Alar, V. Šimunović, I. Juraga: Teorijske osnove korozijskih procesa i metode zaštite: Autorizirana predavanja, Fakultet strojarstva i brodogradnje, Zagreb, 2011.
- [5] Postupak sačmarenja, MIV d.d, 2019.
- [6] Površinska zaštita postupkom bojenja, MIV d.d, 2019.
- [7] Katedra za zaštitu materijala: Korozija i metode zaštite od korozije, Fakultet strojarstva i brodogradnje, 37, 53. dostupno 10.08.2019.
- [8] I.Juraga, V.Šimunović, I.Stojanović, V.Alar: Mehanizmi zaštite od korozije: Autorizirana predavanja, Fakultet strojarstva i brodogradnje, Zagreb, 2012.
- [9] Priručnik za Hempelove boje, 2010.
- [10] <http://mobihel.helios.si/cro/greske-pri-lakiranju>, dostupno 11.08.2019.
- [11] <http://viking.hr/plastifikacija/>, dostupno 11.08.2019.
- [12] <https://ogradni-sistemi.com/plastificiranje/>, dostupno 11.08.2019.
- [13] Postupak površinske zaštite plastifikacijom, MIV d.d, 2019.
- [14] Tehnički opis postrojenja za plastifikaciju u fluidiziranom sloju, MIV d.d, 2017.
- [15] Hala za bojanje vodenim postupkom uranjanjem u bitumen, MIV d.d, 2013.
- [16] <http://www.coppeliarobotics.com/helpFiles/index.html>, dostupno 12.08.2019.
- [17] V. Prkačin, Upravljanje modularnim robotima, Sveučilište u Dubrovniku, Dubrovnik 2017.
- [18] N.Varšić, Agentna simulacija bespilotne letjelice u 3D okolini, FOI Varaždin, 2016.
- [19] <https://www.surplex.com/hr/proizvodi/kuka.html>, dostupno 14.08.2019.
- [20] <https://www.robots.com/robots/kuka-kr-16>, dostupno 14.08.2019.
- [21] Robots KR 16-3 Operating Instructions, Kuka roboter GmbH, 2012.
- [22] L. Sumić, Primjena robota u neurokirurgiji, Sveučilište u Zagrebu FSB, Zagreb 2012.
- [23] <https://www.kuka.com/en-us/products/robotics-systems>, dostupno 14.08.2019.
- [24] Katarina Haler, Poslovna primjena virtualne i proširene stvarnosti, Požega,2018.
- [25] <https://desordemnosmiolos.blogspot.com/2013/07/realidade-virtual-parte-1.html>, dostupno 15.08.2019.

- [26] J. Novak-Marcincin, P. Brazda, M. Janak, M. Kocisko: Application of virtual reality technology in simulation of automated, Prešov, 2011.
- [27] <https://geek.hr/pojmovnik/sto-virtualna-stvarnost/>, dostupno 15.08.2019.
- [28] <https://virtualreality.esi-group.com/head-mounted-displays-virtual-engineering/work-meets-play-virtual-reality-engineering>, dostupno 15.08.2019.
- [29].<http://test.cadcam-group.eu/hr/blog/virtualna-stvarnost-na-cad-programskim-rjesenjima>, dostupno 15.08.2019.
- [30].http://novilist.hr/novilist_public/Znanost-i-tehnologija/Tehnologija/Virtualna-stvarnost-u-zdravstvu-turizmu-industriji-i-vojsci, dostupno 15.08.2019.

7. Popis slika

Slika 2.1 Korozijsko ponašanje nekih metala [4]	2
Slika 2.2 Jedinica za sačmarenje odljevaka [5]	4
Slika 2.3 Izgled površine čistoće Sa 2½ [5]	4
Slika 2.4 Prikaz pojedinih slojeva premaza [7]	5
Slika 2.5 Postupak bojenja ručnim raspršivačem [6]	7
Slika 2.6 Uzroci i sprječavanje nekih grešaka pri lakiranju [10]	8
Slika 2.7 Prah za plastifikaciju [12]	9
Slika 2.8 Ručna plastifikacija epoxy prahom [13]	10
Slika 2.9 Predgrijavanje komada i komora za fluidizaciju [13]	11
Slika 2.10 Oznaka kvalitete antikorozijske zaštite [13]	12
Slika 2.11 Zaslon upravljačke peći [13]	13
Slika 2.12 Peć za predgrijavanje odljevaka [13]	13
Slika 2.13 Plastifikacija robotom KUKA KR-240 [13]	14
Slika 2.14 Bazen za fluidizaciju praha [13]	15
Slika 2.15 Rashladni tunel [13]	16
Slika 2.16 Oštećenje od prihvata [13]	16
Slika 2.17 Bitumenizirani podzemni hidrant [15]	17
Slika 2.18 Linija za bitumenizaciju uranjanjem [15]	18
Slika 2.19 Edukacijska verzija V-REP [16]	19
Slika 2.20 Modeli robotskih ruka u V-REP [16]	20
Slika 2.21 Model robotske ruke KUKA KR16 u V-REPU [16]	21
Slika 2.22 Smjer vrtnje osi robota [21]	22
Slika 2.23 Radni prostor robota KUKA KR16 [21]	22
Slika 2.24 Primjer robotskog sustava [21]	23
Slika 2.25 Viktorija zdenac	24
Slika 2.26 Stalak u SolidEdge-u i V-REP-u	24
Slika 2.27 Virtualni koncept iz 1965. godine [25]	25
Slika 2.28 HTC Vive VR set [25]	26
Slika 2.29 Primjer primjene u industriji [28]	26
Slika 2.30 3D modeliranje pomoću virtualne stvarnosti [29]	27
Slika 2.31 Primjena VR u automobilskoj industriji [29]	29
Slika 2.32 Upravljanje robotskom rukom pomoću VR u V-REPU [29]	29

Slika 4.1 Robotska ćelija u V-REPu	30
Slika 4.2 Ikona za pokretanje programskoga alata V-REP	31
Slika 4.3 Zaslona i nova scena programskoga alata V-REP	31
Slika 4.4 Odabir robota i stavljanje u scenu	32
Slika 4.5 Pomicanje i promjena položaja robota po koordinatnim osima	32
Slika 4.6 Postavljanje u način inverzne kinematike	33
Slika 4.7 Spajanje i pozicioniranje Dummy "tip" sa zadnjom osi robota	33
Slika 4.8 Spajanje Dummy „tip“ i „target“	34
Slika 4.9 Postavljanje kinematičkih računskih modula	34
Slika 4.10 Pomicanje ciljne točke „target“	35
Slika 4.11 Prijenos pištolja za boju	36
Slika 4.12 Dodavanje, igled i pozicija senzora	36
Slika 4.13 Okretanje senzora u položaj osi	37
Slika 4.14 Sastavljanje senzora i pištolja za boju	37
Slika 4.15 Okretanje pištolja za boju u ispravnu stranu	37
Slika 4.16 Model stalka u SolidEdge- u	38
Slika 4.17 Pretvaranje modela u STL dokument	38
Slika 4.18 Uvoz 3D modela u V-REP	39
Slika 4.19 Smanjivanje modela stalka i njegovo pozicioniranje	39
Slika 4.20 Prikaz položaja modela stalka	39
Slika 4.21 Dodavanje točki putanje	40
Slika 4.22 Dodavanje druge točke putanje i pozicioniranje	40
Slika 4.23 Dodavanje treće točke putanje i pozicioniranje	41
Slika 4.24 Povezivanje kontrolnih točki i izlaz iz prozora za uređivanje putanje	41
Slika 4.25 Dodavanje skripte „target-u“	42
Slika 4.26 Izgled skripte u koju je potrebno upisati naredbe	42
Slika 4.27 V- REP korisnički priručnik sa uputstvima	43
Slika 4.28 V- REP funkcije za skriptu	43
Slika 4.29 V- REP funkcija za dohvatanje objekta	44
Slika 4.30 Funkcija za udruživanje objekta sa skriptom	44
Slika 4.31 Funkcija za pomicanje duž putanje	44
Slika 4.32 Skripta „target-a“ sa funkcijama i parametrima	45
Slika 4.33 Dodavanje pozicionera	46
Slika 4.34 Veza radnog komada i pozicionara	46
Slika 4.35 Dodavanje skripte pozicioneru	47

Slika 4.36 Provjera inicijalnog pokretanja	47
Slika 4.37 Funkcija za dohvaćanje objekta povezanog sa skriptom	47
Slika 4.38 Upis parametara za dohvaćanje objekta sa skriptom	48
Slika 4.39 Provjera aktivacije simulacije	48
Slika 4.40 Funkcija za vremenski korak simulacije	49
Slika 4.41 Funkcija za zakretanje zgloba	49
Slika 4.42 Formula promjene pozicije	49
Slika 4.43 Funkcija postavljanja unutarnjeg položaja	50
Slika 4.44 Završni izgled skripte pozicionera	50
Slika 4.45 Izrada veće vješalice	51
Slika 4.46 Izrada dvije manje vješalice	51
Slika 4.47 Položaji prvog i drugog obojenog komada	52
Slika 4.48 Položaji prvog i drugog sačmarenog komada	52
Slika 4.49 Izrada krova sačmare	53
Slika 4.50 Bočne stranice sačmare	53
Slika 4.51 Prikaz sačmare	54
Slika 4.52 Prikaz sušare	55
Slika 4.53 Položaji sačmare i sušare	55
Slika 4.54 Bočne stranice filtera	56
Slika 4.55 Pladanj filtera	56
Slika 4.56 Usis filtera	57
Slika 4.57 Prikaz položaja filtera	57
Slika 4.58 Izrada manje ograde	58
Slika 4.59 Izrada duže ograde	58
Slika 4.60 Konačni prikaz ćelije za bojanje.....	59
Slika 4.61 Bojanje radnog komada	59
Slika 4.62 Podešavanje vrste i količine boje	60
Slika 4.63 Postavke u skripti pištolja za boju	60
Slika 4.64 Bojanje iz različitih perspektiva	61
Slika 4.65 Obojani komad	61

8. Prilozi

Prilog 1.

V-REP kod u skripti za rotaciju pozicionera

```
if(sim_call_type==sim.syscb_init)then    --provjeri   dali   je   prvo
inicijalno pokretanje

    h=sim.getObjectAssociatedWithScript(sim.handle_self)    -- dohvati
    objekt koji je povezan sa skriptom u zagradi parameter "ime skripte"
end

if(sim_call_type==sim.syscb_actuation)then    -- provjeri   dali   je
inicijalizacija izvršena. Ako je izvršena nastavlja dalje
local dt=sim.getSimulationTimeStep()    -- Dohvaća vremenski korak
simulacije (virtualno vrijeme simulacije koje prolazi pri svakom
prolasku simulacije glavne skripte)
local p=sim.getJointPosition(h)    --Dohvaća unutarnju vrijednost kuta
zakretanja zgloba
p=p+dt*0.07    -- formula koja izračunava novu poziciju na način da se
povećava vrijednost kuta zakretanja zgloba
sim.setJointPosition(h,p)    -- Postavlja unutarnji položaj zakretanja
zgloba
end
```

Prilog 2.

CD-R disk