

# PROGRAMIRANJE INDUSTRIJSKIH ROBOTA ZA POTREBE ROBOTIZIRANOG ZAVARIVANJA

Danijel Radočaj, Maja Jurica, Sebastijan Kos, Tomislav Pavlic

**Ključne riječi:** programiranje, industrijski robot, on-line programiranje, off-line programiranje, zavarivanje, RobotStudio, ABB

## Sažetak:

U ovom radu opisano je programiranje industrijskih robota. Opisan je pojam, povijest i struktura industrijskih robota. Nabrojane su i opisane metode programiranja industrijskih robota te njihove karakteristike. Detaljno su opisani postupci on-line i off-line programiranja na realnim primjerima. On-line programiranje izvršeno je u Laboratoriju za zavarivanje Fakulteta strojarstva i brodogradnje, Sveučilišta u Zagrebu. Primjer on-line programiranja je robotsko zavarivanje testnog CAD modela radnog komada. Off-line programiranje izvršeno je u programskom alatu RobotStudio, tvrtke ABB, na istom modelu radnog komada. Navedene su i smjernice prema kojima će se dalje razvijati ovaj rad.

## PROGRAMMING OF INDUSTRIAL ROBOTS FOR NEEDS OF ROBOTIC WELDING

**Key words:** programming, industrial robot, on-line programming, off-line programming, welding, RobotStudio, ABB

## Abstract:

This thesis describes programming of industrial robots. What robots are, history and structure of robotic systems are mentioned. Methods of programming are listed and explained, with their specifications. In detail description are ways of programming industrial robots with realistic examples. On-line programming is completed in Laboratory of welding, on Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb. Robotic welding is example of CAD test workpiece. Off-line programming is completed in software package RobotStudio made by company ABB, on same workpiece. There are mentioned future goals for this thesis.

## 1. UVODNA RAZMATRANJA

Primjena robota u industriji u stalnom je porastu. Industrijski procesi koji koriste robote znatno su ubrzani, poboljšane kvalitete te na velikim serijama i znatno ekonomski isplativiji. Povećanje broja robota u industriji također povećava i potrebu za operaterima na robotskim stanicama. U radu je detaljnije opisan postupak off-line programiranja industrijskih robota. Glavno težište je stavljeno na programiranje, te metode programiranja industrijskih robota, za potrebe zavarivanja.

Metode su izvršene na realnim primjerima, uspostavljene su neke osnovne teze o razlikama između metoda programiranja, prednosti i mane između metoda. Osnove rada u programskom alatu RobotStudio i osnove rada sa robotskom stanicom VRC-1G MIG + 1G TIG/1dm su navedene. Obavljeno je on-line programiranje industrijskog robota. Navedene su kompleksnije inačice off-line programiranja i pripadne simulacije.

### 1.1 Općenito o industrijskim robotima

Pojam industrijski robot definiran je standardom ISO 8373, kao uređaj s automatskom kontrolom, reprograbilan, višenamjenski manipulator s tri ili više osi tj. stupnjeva slobode gibanja.

Područje robotike u praktičnom smislu može se sagledati kao projektiranje i upotreba robota za proizvodne procese (najšire područje uporabe industrijskih robota).

Uobičajene uloge robota su zavarivanje, bojanje, sastavljanje, pakiranje, paletiziranje, ispitivanje proizvoda i drugo. Sve uloge roboti obavljaju sa visokom izdržljivošću, brzinom i preciznošću (slika 1).



*Slika 1. Zavarivanje pomoći industrijskog robota*

Najzastupljenije područje upotrebe industrijskih robota je u procesu zavarivanja. Automobilska industrija je vodeća u korištenju industrijskih robota u procesima elektrootpornog zavarivanja. Procesi zavarivanja te manipulacije materijalom i pozicijama, iznosili su približno 80% sveukupnog korištenja industrijskih robota u 2003. [1]

## 1.2 Općenite značajke robota

• Broj osi: Tri osi potrebne su kako bi se dosegnula bilo koja točka u prostoru. Za potpunu kontrolu orijentacije vrha manipulatora (npr. pištolj za zavarivanje), potrebne su još tri osi. Neke robotske konfiguracije kompromisno mijenjaju ograničenja kod kretanja u zamjenu za cijenu, brzinu i preciznost same konfiguracije.

- Stupnjevi slobode: obično ih označava broj i vrsta osi.
- Radni prostor: skup svih točaka u prostoru koje robot može dosegnuti.
- Struktura robota: opisuje vrste članova koji određuju moguća gibanja robota. Vrste struktura uključuju pravokutnu, cilindričnu, sfernu i rotacijsku.
- Nosivost robota: nazivna masa s kojom robot može manipulirati.
- Brzina: opisuje vrijeme potrebno da robot pozicionira izvršni (alatni) koordinatni sustav u željenu poziciju.
- Ubrzanje: opisuje koliko brzo robot može ubrzati promjenu pozicije.
- Preciznost: opisuje s kojim odstupanjem se robot pozicionira u zadanu poziciju.
- Ponovljivost: opisuje s kojom preciznošću se robot vraća u isprogramiranu poziciju.
- Kontrola kretnji: u ovisno o zadatku koji robot izvršava, kretnje moraju biti isključivo kontinuirane (npr. zavarivanje) ili nekog drugog oblika.
- Vrsta pokretne sile: vrste pogonskih motora i aktuatora u zglobovima robota.
- Prijenos: postoje direktni spojevi na zglove i preko mjenjačkih mehanizama.
- Popustljivost: značajka koja opisuje manje promjene položaja kod velikog opterećenja robota.

## 2. METODE PROGRAMIRANJA

U industrijskoj primjeni robota postoje dvije glavne kategorije programiranja industrijskih robota. To su on-line programiranje i off-line programiranje. On-line programiranje zahtijeva manju razinu obuke osoblja te je više oblikovana opcija kao prijateljska korisniku. *Off - line* programiranje zahtijeva obuku rukovanja programskim alatom za izradu off-line simulacija procesa koji se zahtijeva od robota.

### 2.1 Metoda *on – line* programiranja

On-line programiranje (*eng. Teach-in*) je proces „učenja“ robota da odradi željeni zadatak. Ovu vrstu programiranja izvršava iskusan operater programer, na način da provodi robota kroz željene putanje pomoću privjeska za učenje (*eng. Teaching pendant*). Ova metoda obično sadrži prolazak robota kroz željene pozicije, uz snimanje željenih točaka između putanja u memoriju kontrolera te definiranje značajki snimljenih točaka kako bi se opisale željene putanje. Iako je ova metoda jednostavna i vrlo raširena, ona ima nekoliko nedostataka. Upravljanje kretnjama robota nije tako jednostavno i iziskuje iznimnu pažnju.

Potrebno je pratiti u kojem se koordinatnom sustavu robot nalazi, paziti na pozitivnu i negativnu orientaciju koordinatnog sustava kako se ne bi umjesto odmaka dogodila kolizija sa radnim komadom. Potrebno je biti u mogućnosti vizualizirati koordinatni sustav s obzirom na njegovu trenutnu zakrenutost, što je posebno izraženo kod postavljanja koordinatnog sustava alata (*eng. Tool center point TCP*). Također, vrlo bitno je paziti na putanje između točaka, uskladiti ih da se ne dogodi kolizija sa radnim komadom tijekom prijelaznih putanja kada robot ne radi proces za koji je zadužen (npr. zavarivanje).

Robot ima i svoja fizička ograničenja koja često dolaze do izražaja kod nemogućnosti dohvaćanja želenog položaja robota (*eng. Dead Zone*). S obzirom na gore navedeno, postupak programiranja traje određeno vrijeme u kojem se operater mora intenzivno uključiti u svijet robota i koordinatnih sustava. Postupak je iscrpljujući za operatera. Nakon izvršenog procesa programiranja točaka i putanja, cijeli tijek programa je potrebno višestruko provjeriti. Potrebno je obratiti pozornost na brzine kretanja robota, naredbe kontroliranja vanjskih uređaja (npr. aparat za *MIG/MAG* zavarivanje) te njihove postavke, položaje članaka robota i njihov ulazak u eventualnu mrtvu zonu kretanja te moguće nepredviđene kolizije koje nastaju tijekom prijelaznih putanja.

Još jedna od loših strana on-line programiranja je da za svaku i najmanju promjenu na radnom komadu ili procesu kojeg robot obavlja, potrebno je ponoviti cijeli proces kontrole isprogramiranih pozicija. Potrebno je utvrditi ispravnost tijeka programa i spremnost za automatsku izvedbu. Bez obzira na nedostatke, on-line programiranje robota je još uvijek najraširenija vrsta programiranja robota za male do srednje velike sustave.

### 2.2 Metoda *off – line* programiranja

Off-line metoda programiranja industrijskih robota je metoda koja se izvršava na računalu. Ne zahtijeva isključivanje robota iz proizvodnog procesa, što je prednost u usporedbi sa on-line programiranjem. Ova metoda programiranja se zasniva na bazama podataka koje sadrže robote, pozicionere i drugu opremu robotskih stanica. Roboti sadrže precizna kinematska gibanja istovjetna onima u stvarnoj okolini. Postupak *off-line* programiranja izvršava se u programskim alatima, kao što su od općenitih: *Delmia*, a većinom se izvršava u programskim alatima proizvođača robotskih stanica, kao što je *ABB RobotStudio*.

Glavne prednosti ove metode su brzina i točnost programiranja putanja robotskog manipulatora, naročito kod kompleksnijih radnih komada. Također, jedna od velikih prednosti je mogućnost programiranja simultanog gibanja više robotskih mehanizama unutar jednog robotskog sustava, što je prilično vremenski zahtjevan posao ukoliko se izvršava putem *on - line* programske

postupka. Uvozom vlastitih CAD modela u programske alate dobiva se neograničenost programiranja s velikom vjernosti stvarnoj konfiguraciji robotske stanice.

Postupak se sastoji od prvotne analize materijala u svrhu odabira vrste postupka zavarivanja te po kojim putanjama na radnom komadu je potrebno izvršiti zavarivanje. Nakon definiranja putanja zavarivanja i tehničkih karakteristika, slijedi programiranje robotske linije ili stanice. Simulacijom se provjerava tijek programa, analiziraju se eventualne pogreške i izvode se poboljšanja. Nakon odabira konačne ispravne verzije programa, on se šalje u memoriju kontrolera i time je postupak *off-line* programiranja završen.

### 3. EKSPERIMENTALNI DIO: *ON-LINE* PROGRAMIRANJE

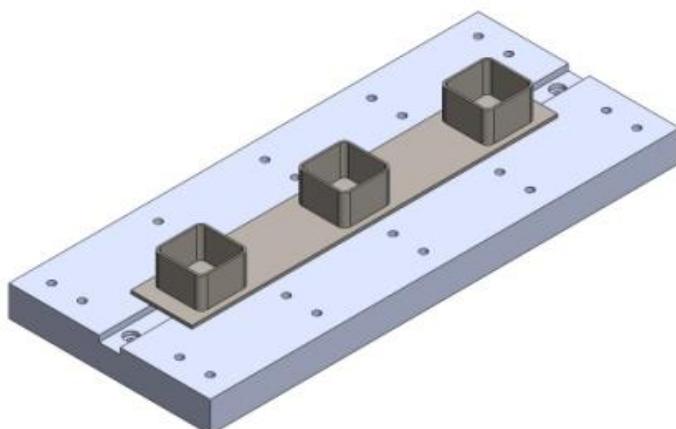
U eksperimentalnom dijelu rada korišten je sustav prikazan u [2]. Glavna komponenta sustava je robotski manipulator, OTC Daihen Almega AX-V6, prikazan na slici 2.



*Slika 2. Robotski sustav u Laboratoriju za zavarivanje, FSB Zagreb*

#### 3.1 Projektiranje, izrada i priprema radnog komada za zavarivanje

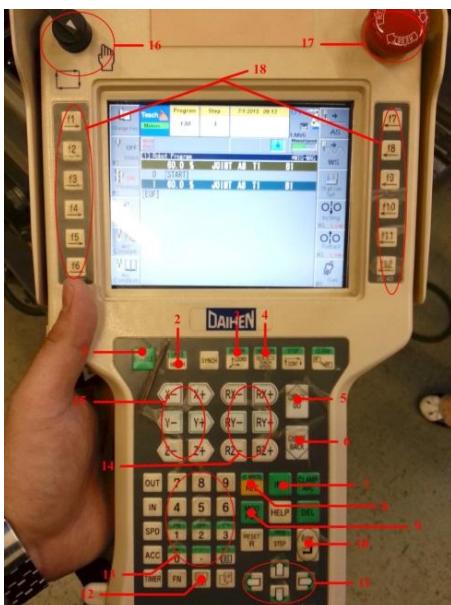
Prema vrsti robotskog sustava i dostupnosti metaličnih profila, projektiran je radni komad na kojem je obavljen eksperimentalni dio *on-line* programiranja. Projektiranje je izrađeno u *CAD* programskom alatu *SolidWorks*. Za model uzeti su metalični profili pravokutnog oblika 80x80x50 mm, zavareni na željezni lim debljine 3 mm. Nakon narezivanja profila, potrebno je označiti pozicije profila na limu (slika 3).



*Slika 3. Izrađeni CAD model radnog komada pričvršćen na radnu ploču*

### 3.2 On - line programiranje procesa zavarivanja

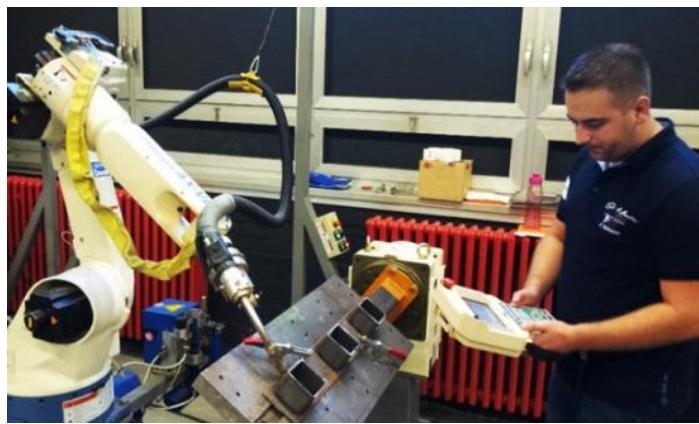
*On - line* proces programiranja izvršava se izravno na robotskoj stanici, vodeći pištolj za zavarivanje kroz točke na radnom komadu, u kojima se događaju promjene u postupku zavarivanja. Tim postupkom se definiraju putanje i ono što robot radi prilikom prolaska kroz te putanje. Definiranje točaka i putanja vrši se uz pomoć privjeska za učenje. Privjesak za učenje robotskog sustava prikazan je na slici 4.



*Slika 4. Privjesak za učenje*

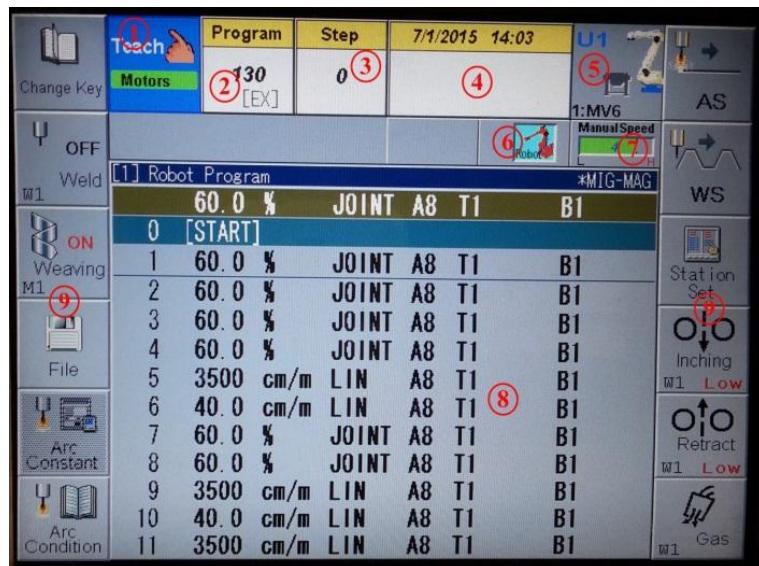
Operater koristi privjesak za učenje u svrhu kontrole i memoriranja željenih putanja. Privjesak ima duži kabel kako bi operater bio neograničen u kretanju tijekom programiranja. Operater s obje ruke prihvati za dršku privjesak te prstima pritisne tipke na poleđini koje moraju biti aktivne kako bi robot imao energije u sebi potrebne za pokretanje motora. Čim operater pritisne prejako ili preslabo tipke na poleđini, robot se zaustavlja u trenutnom položaju te motori nisu više pod napajanjem. Ta funkcija je uvedena iz sigurnosnih razloga i nepotrebnog rada motora tijekom definiranja parametara procesa koji nisu vezani izravno za kretnju robota.

Programiranje je predviđeno obavljati u stojećem položaju s dobrim pogledom na radni komad kako bi se optimalno postavio kut pištolja za zavarivanje i odmak sapnice pištolja od radnog komada.



*Slika 5. On - line programiranje industrijskog robota Almega AX-V6*

Svaku radnju koja se izvršava na robotu moguće je pratiti na LCD zaslonu privjeska za učenje. LCD zaslon privjeska za učenje prikazan je na slici 6.



*Slika 6. Zaslon privjeska za učenje*

Korake *on - line* programiranja industrijskog robota moguće je opisati kronološki prema tijeku izvođenja. Prvi korak je uključivanje robota uz uvjet da ne postoje nikakve prethodno detektirane pogreške. Drugi korak je odabir režima rada, tj. potrebno je uskladiti odabire režima rada na svim kontrolnim sklopkama robotskog sustava.

Treći korak se sastoji od uključivanja motora, koje se vrši pomoću odabira režima rada *Teach* te pritiska na tipku na poleđini privjeska za učenje. Četvrti korak je učenje programa. Učenje se sastoji od: dolaska robota u točke željenih putanja, memoriranje točaka, putanja i parametara u memoriju, definiranja brzina kretanja pojedinih putanja, definiranja vrsta interpolacija putanja i tehničkih parametara. Slijedi peti korak, a to je provjera naučenog programa. Naredbama *Check GO/BACK* postupno se provjerava ispravnost programa.

Zadnji korak je puštanje u rad, odabirom automatskog režima rada. U slučaju hitnog zaustavljanja radi greške u programu ili neke nepredviđene situacije, koristi se tipka za hitno zaustavljanje, tzv. gljiva.

## 4. EKSPERIMENTALNI DIO: OFF - LINE PROGRAMIRANJE

*Off - line* programiranje se izvršava u programskim alatima, na osobnim računalima. Uz pomoć biblioteka koje sadrže *CAD* modele s mehanizmima robota, pozicionera i opreme za robotske stanice, moguće je izraditi stanicu koja je potpuno istovjetna realnoj koju je potrebno isprogramirati putem programskog alata. Uz dodatak opcije uvoza *CAD* modela radnih komada, moguće je vjerno prikazati rad robota na istom radnom komadu kao i u stvarnosti. Cilj je pokazati na istom primjeru on-line i off-line metodu programiranja.

Zbog dostupnosti isključivo *ABB RobotStudio* programskog alata, potrebno je pronaći robota i pozicioner što sličnijih karakteristika realnom primjeru koji se nalazi u Laboratoriju za zavarivanje, na Fakultetu strojarstva i brodogradnje u Zagrebu.

### 4.1 Osnove novog projekta (*RobotStudio*)

Potrebno je dodati elemente robotskog sustava, poput robota, pozicionera i drugih elemenata u radni prostor robotske stanice unutar programskog alata *RobotStudio*. Dodavanje elemenata se vrši pomoću opcije *ABB Library*. Tijekom dodavanja mehanizama, oni će imati razne orijentacije i položaje koji su neispravni ili neodgovarajući, ali još nije bitno postavljanje istih u ispravni položaj jer za to je potrebno prvo postaviti virtualni kontroler. Ukoliko bi se postavljale pozicije bez mijenjanja položaja baznih koordinatnih sustava mehanizama, nastao bi u dalnjem tijeku programiranja problem nesinkroniziranih koordinatnih sustava alata, apsolutnog i robotske koordinatnog sustava. To bi vodilo u pogreške tijekom programiranja koje bi pokazivale na neispravne konfiguracije robotskih zglobova.

Potrebno je izraditi virtualni kontroler koji će imati ulogu upravljanja robotom i ostalim osima u robotskom sustavu. Virtualni kontroler se ponaša istovjetno kontroleru u stvarnosti. To je bitno zbog opcija memoriskog prostora kontrolera u kojega će biti poslan program koji se napravi off-line metodom. Isključivo je moguće poslati program u kontroler istog proizvođača i modela. U virtualni kontroler se spremaju svi parametri vezani za robotski sustav tj. stanicu, kao što su orijentacije i položaji baznih koordinatnih sustava mehanizama, podaci o mehanizmima te podaci o konfiguraciji stanice. Izrada kontrolera se vrši pomoću opcije *Robot system, From layout*.

Postupak izrade virtualnog kontrolera se sastoji od četiri koraka. U prvom koraku odabire se ime sustava i verzija *RobotWare* programskog modula, uz lokaciju gdje će sustav biti smješten na tvrdom disku računala. U drugom koraku odabiru se mehanizmi koji bi trebali biti konfiguirani u virtualnom kontroleru. Treći korak se sastoji od postavljanja zasebnih mehanizama, tj. sustava mehanizama. Bitno je postaviti robot i pozicioner svaki u zaseban mehanizam (*T\_ROB* i *T\_POS*). To je bitno iz razloga korištenja opcije *MultiMove* koja omoguće sinkronizirane kretnje pozicionera i robota istovremeno, uz potrebu razdvajanja točaka i putanja svakog mehanizma jer one nisu iste. Posljednji korak je provjera postavljenog sustava te poravnjanje glavnog koordinatnog sustava robotske stanice, u ovom slučaju to je robot *IRB 1600ID*. Postupak izrade virtualnog kontrolera završava se pritiskom na opciju *Finish*.

Nakon izrade virtualnog kontrolera, potrebno je postaviti ispravne orijentacije i položaje mehanizama u radnom prostoru robotske stanice. Pomoću izbornika *Controller*, opcije *Edit system*, ulazi se u postavke robotskog sustava. Nakon otvaranja izbornika za promjenu postavki sustava, odabiru se položaji baznih koordinatnih sustava pozicionera i robota. Tako se pozicioniraju mehanizmi s obzirom na apsolutni koordinatni sustav.

Slijedi odabir alata za zavarivanje. Odabran je pištolj za zavarivanje *Binzel Abirob A500*. Pištolj pripada *MIG/MAG* tehnologiji zavarivanja. Izvedba pištolja je isključivo za robotsko zavarivanje. Pištolj postoji i u biblioteci programskog alata *RobotStudio* te se dodaje preko izbornika, u radni prostor robotske stanice. Nakon ubacivanja alata u robotski sustav, potrebno je priključiti isti na prihvatinicu za alat robota. Priključivanje se obavlja putem opcije *Attach* te izbora robota *IRB 1600ID*.

Programski alat *RobotStudio* ima mogućnost uvoza *CAD* modela u radni prostor. Da bi bilo moguće pričvrstiti radni komad na pozicioner, potreban je nosač i na njemu radna ploča. Model

nosača i radne ploče izmodelirani su u programskom alatu *SolidWorks*, koji je namijenjen računalnom dizajnu modela (*CAD*). Radni komad je također izmodeliran u istom programskom alatu. Korištene su opcije *Sketch*, *Extrude*, *Extrude cut*, *Mate*, *Weldment module (Structural member)* i druge. Izmodelirani nosač na sebi ima pričvršćenu radnu ploču, istovjetnu onoj kod eksperimentalnog dijela on-line programiranja. Na radnu ploču pričvršćen je radni komad, izmodeliran prema nacrtu i mogućnostima. Model je potrebno je spremiti u *ACIS (\*.sat)* formatu, koji je zahtijevan od strane programskog alata *RobotStudio*, za uvoz modela u radni prostor.

Programski alat *RobotStudio* ima mogućnost uvoza vlastitih *CAD* modela koji nisu podržani u biblioteci programa. Moguće je uvesti neograničen broj modela, s uvjetom da su u *ACIS (\*.sat)* formatu. Svi *CAD* alati imaju mogućnost spremanja modela u *ACIS* format. Ovom opcijom je ujedno i otvorena mogućnost izrade identične off-line simulacije sustava kao što je u stvarnosti.

Uvezeni *CAD* model je potrebno pozicionirati u hvatišta pozicionera. Najtočniji način za to je izrada dva pomoćna koordinatna sustava; jedan u centru modela i drugi u centru hvatišta pozicionera.

Nakon uvoza radnog modela (nosač, radna ploča i radni komad), potrebno je odrediti novi radni objekt u prostoru. Potrebno je postaviti da se radnim modelom upravlja mehanizmom *STN1* (pozicioner).

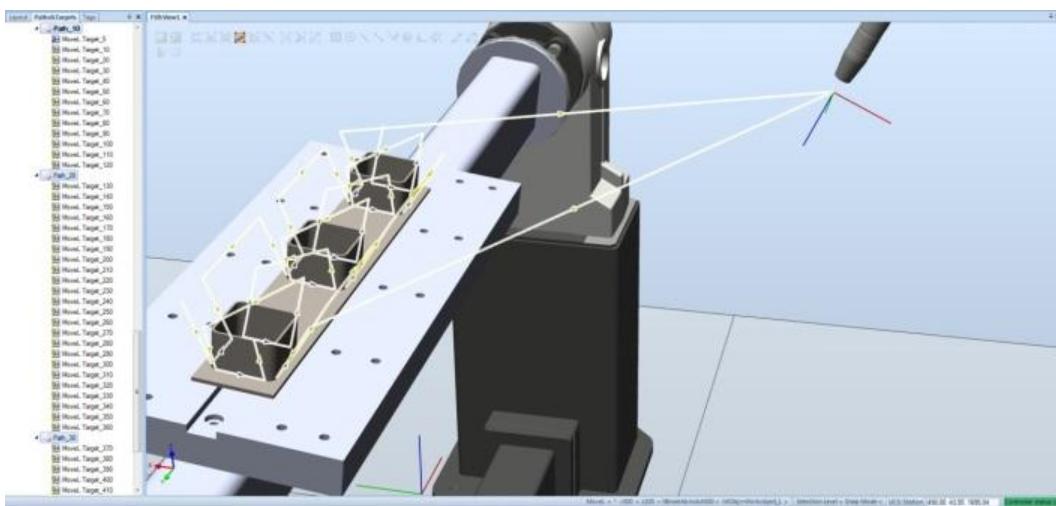
Nakon što je određen radni objekt, potrebno je izraditi točke koje će biti granice putanja robota. Točke (*eng. Targets*) su mete kroz koje će ishodište koordinatnog sustava alata proći tijekom zadanoг procesa zavarivanja. Izrađuju se pomoću opcije *Target*, *Create target*. Potrebno je odabrati korisnički koordinatni sustav (*UCS*), koji je baziran na radnom komadu i odabrati opciju *Align to the closest part*.

Nakon odabranog koordinatnog sustava, potrebno je odabrati po potrebi, alat za prihvaćanje točaka (*eng. Snap*) te označiti točke. Kod izrade točaka, potrebno je paziti na ulazne i izlazne točke. Točke ulaza i izlaza bi trebale biti odmaknute od radnog komada kako pištolj nebi zapeo za radni komad. Potrebno je ulazne i izlazne točke podići iznad radnog komada kako bi pištolj mogao nesmetano prijeći u drugu ulaznu točku.

Nakon izrade ulaznih, izlaznih i radnih točaka, potrebno je odrediti početnu točku robota. Robota se prvo postavi u željenu poziciju alatima pozicioniranja. Kada se robot dovede u željenu poziciju koja se smatra početnom, naredbom *Teach target*, izrađuje se nova točka u trenutnoj poziciji robota. Nakon što se izrade sve točke prolaska robota, potrebno je odrediti orientaciju pištolja za zavarivanje. Orientacija se određuje na pojedinačnoj točci. Označivši željenu točku, aktivira se padajući izbornik desnim klikom miša i odabire se opcija *Modify target, Rotate*. U novo otvorenom prozoru, postavi se željeni koordinatni sustav i os, oko koje je potrebno rotirati pištolj.

Nakon unosa numeričke vrijednosti, moguće je više puta primjeniti postavljeni zakret na istu točku. Nakon postavljanja orientacije, moguće je kopirati orientaciju, te je primjeniti na pojedinu ili skup označenih točaka.

Nakon izrade svih potrebnih točaka koje definiranju pozicije gdje koordinatni sustav alata mora proći, potrebno je izraditi putanje od skupova točaka. Putanje su potrebne za daljnju izradu sinkroniziranih putanja robota i pozicionera. Postupak izrade putanja sastoji se od označavanja željenih točaka u putanji te odabirom opcije *Add to new path*. Nakon izrade svih putanja potrebnih za zavarivanje i upravljanje robotom, moguće je aktivirati grafički pregled sa smjerom kretanja svih putanja. Grafički pregled moguće je vidjeti na slici 7.



*Slika 7. Izrada putanje prolaska robota*

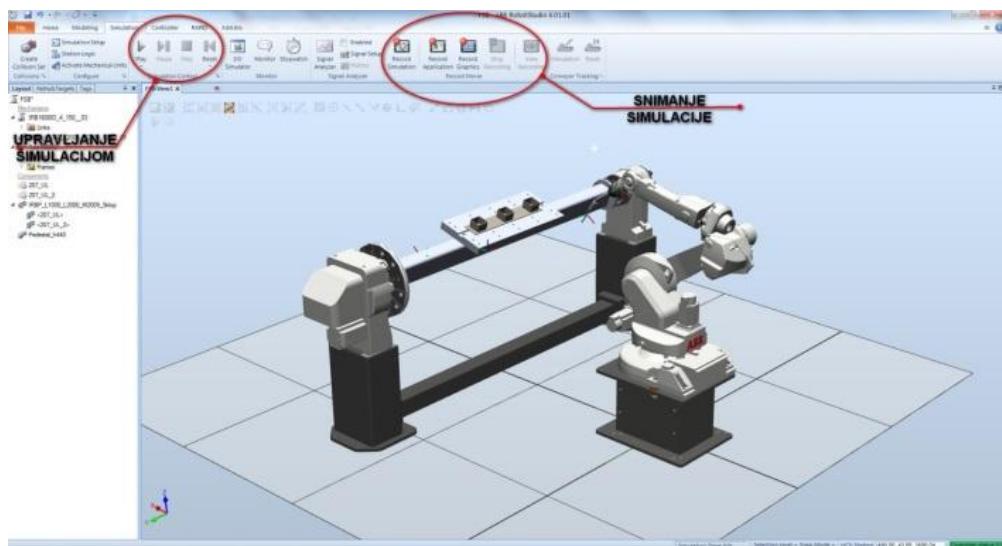
Putanje ne sadrže točke nego instrukcije. Instrukcije sadrže informacije poput vrste interpolacije (krivuljna ili linear), brzine kretanja i drugih bitnih informacija za kretanje robota. Na radnom komadu u ovom poglavlju, brzine su postavljene različito za ulaznu, izlaznu i radnu putanju. Zavarivanje se obavlja najmanjom brzinom, iz tehnoloških podataka ta brzina je oko 30 cm/min. U sklopu simulacije postavljene su pokazne brzine, koje je prije slanja programa u kontroler, potrebno uskladiti sa tehnologom za odabrani tehnološki postupak.

Kako bi se kretanje zavarivanja izvršavale simultano s kretanjima pozicionera, potrebno je pretvoriti putanje u *MultiMove* putanje. *MultiMove* je funkcija programskog alata *RobotStudio* koja omogućuje sinkronizirane kretanje više mehanizama unutar jednog robotskog sustava. Pretvaranje klasičnih putanja u *MultiMove* putanje zahtijeva određene radnje. Prvo je potrebno pomoći konfiguratoru. U prvom koraku konfiguriranja *MultiMove* sustava, potrebno je odabrati robota koji nosi radni komad, u drugom koraku potrebno je odabrati robota koji sadrži alat za zavarivanje. Potrebno je odabrati radni objekt u trećem koraku te u posljednjem koraku se nalazi pregled postavljenje konfiguracije sustava. Nakon konfiguriranja sustava, potrebno je pretvoriti klasične putanje u *MultiMove* putanje.

Prvi korak je izbor robotskog sustava i osvježivanje popisa putanja. Potom je potrebno izabrati sve putanja te napraviti testnu simulaciju *MultiMove* putanja. Nakon što simulacija izvrši proračune *MultiMove* putanja, opcijom Create Paths, izrađuju se sinkronizirane *MultiMove* putanje. Nakon izrade putanja, potrebno je napraviti rekalkulaciju novo izrađenih putanja s opcijom Recalculate ID's. Isto je potrebno napraviti za *MultiMove* putanje u robotskom i pozicionerskom sustavu, kako bi one bile sinkronizirane.

Nakon izrade *MultiMove* putanja, postavljanja parametara brzine, interpolacije i dr., potrebno je sinkronizirati sustav s virtualnim kontrolerom.

Prije pokretanja simulacije, potrebno je postaviti glavnu sekvencu koja se izvodi prilikom pokretanja simulacije. U ovom slučaju, postoji samo jedna *MultiMove* putanja pa ona je ujedno i glavna sekvencia simulacije. Sekvencu je potrebno postaviti kao glavnu u oba mehanizma sustava, *T\_POS1* i *T\_ROB1*. Nakon postavljanja glavne sekvence, preostalo je pokrenuti simulaciju te tijekom simulacije zapaziti eventualne logičke pogreške u programiranju. Simulaciju je moguće i snimiti. Simulacijski prozor vidljiv je na slici 8. Simulacija je dostupna u cijelosti na [3].



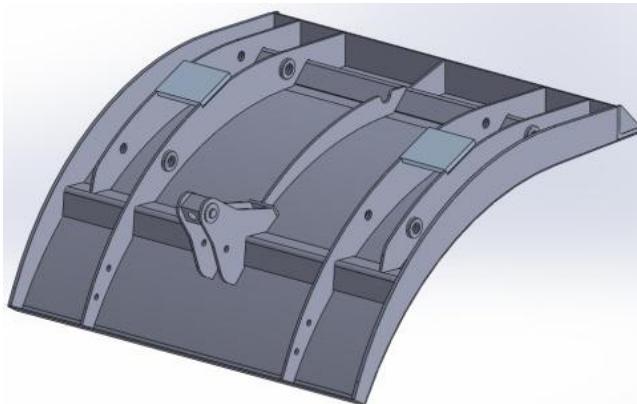
Slika 8. Simulacijski prozor progamskog alata RobotStudio

## 5. PLANIRANJI EKSPERIMENTALNI DIO

Tvrtka *Rasco* koristi robotski sustav s robotom Motoman EA1900N i pozicionerom, prikazanim na slici 9., za zavarivanje srednjeg krila snježnog pluga 900S Kalnik XL, vidljivog na slici 10.

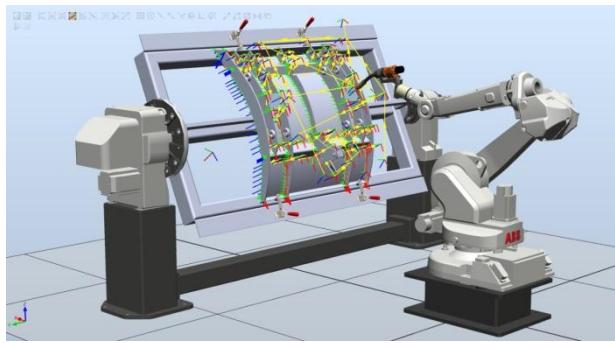


Slika 9. Motoman EA1900N robot s pozicionerom, tvrtka *Rasco*



*Slika 10. CAD model srednjeg krila snježnog pluga Kalnik – 900S*

Napravljena je i off-line simulacija programiranja zavarivanja pluga 900S Kalnik XL, po svim konturama koje su zavarljive prema pretpostavljenom izgledu. Putanje i točke zavarivanja pluga prikazane su na slici 11.



*Slika 11. Putanje i točke zavarivanja pluga*

Korišten je robot ABB IRB 1600ID te pozicioner IRBP L (nosivost 1000 kg, raspon 2000 mm). Nakon izrade putanja, prikazanih na slici 5.3., klasične putanje pretvorene su u *MultiMove* putanje te su im rekalkulirane *ID* oznake. Nakon sinkronizacije s virtualnim kontrolerom, izrađena je glavna sekvenca simulacije u *RAPID* editoru, prikazana na slici 12.

*RAPID* je programski jezik koji se koristi u programskom alatu *RobotStudio* kod kompleksnijih simulacija s većim brojem mehanizama u robotskom sustavu te kod većeg broja radnji koje se događaju istovremeno na različitim mjestima u robotskom sustavu. Nakon izrade glavne sekvene pokrenuta je simulacija, koju je moguće vidjeti u cijelosti na [4].

```

1961   PROC Main()
1962     mmPath_22;      1980   mmPath_11;
1963     mmPath_1;       1981   mmPath_29;
1964     mmPath_2;       1982   mmPath_12;
1965     mmPath_3;       1983   mmPath_30;
1966     mmPath_4;       1984   mmPath_13;
1967     mmPath_23;      1985   mmPath_31;
1968     mmPath_5;       1986   mmPath_14;
1969     mmPath_24;      1987   mmPath_31;
1970     mmPath_7;       1988   mmPath_15;
1971     mmPath_25;      1989   mmPath_32;
1972     mmPath_6;       1990   mmPath_16;
1973     mmPath_21;      1991   mmPath_32;
1974     mmPath_26;      1992   mmPath_17;
1975     mmPath_8;       1993   mmPath_33;
1976     mmPath_27;      1994   mmPath_18;
1977     mmPath_9;       1995   mmPath_34;
1978     mmPath_10;      1996   mmPath_19;
1979     mmPath_28;      1997   mmPath_35;
1980     mmPath_11;      1998   ENDPROC

```

*Slika 12. Izvod RAPID programske koda glavne sekvence*

## 6. ZAKLJUČAK

Programiranje industrijskih robota kompleksan je posao. Proučene su glavne metode programiranja. *On - line* metoda programiranja je jednostavna, no zahtijeva puno vremena da bi se izvodila često i na kompleksnijim radnim komadima. Operater za *on - line* metodu programiranja mora imati obuku i kvalifikaciju za takav način programiranja robota, te mora biti u dogovoru sa tehnologom. *On - line* metoda programiranja je metoda koja se koristi kod jednostavnijih radnih komada i kod većih serijskih proizvodnji pozicija. Iako ima svoje nedostatke zbog češće potrebe za programiranjem i dalje se koristi kao glavna metoda programiranja industrijskih robota.

*Off - line* metoda programiranja je zamjetno kompleksnija za izvesti, ali uz dobro obučenog operatera na osobnom računalu, moguće su velike uštede kako u vremenu tako i ekonomski. Ovom metodom moguće je uz manje truda isprogramirati kompleksne radne komade, te jednostavnim postupkom poslati program u kontroler robotske stanice. Programski alati za *off - line* programiranje su komercijalni i njihove cijene rangiraju visoko.

Metoda *off-line* programiranja omogućava detaljniji razvoj proizvodnog postupka koji se obavlja robotskom stanicom, a samim time i kvalitetnije izведен postupak. Kod obje metode se pojavljuju isti problemi tijekom postupka programiranja kao što su interpolacija gibanja, brzina kretanja pištolja za zavarivanje, orientacija istog, kolizije sa radnim komadom i drugo. Moguće je zaključiti da su metode potpuno istog karaktera i postavki, samo je drugačiji način izvedbe. Ukoliko je moguće, uvijek je preporučljiva *off - line* metoda, naravno ako se uzme u obzir da postoji obučeni operater i odgovarajući modul u konfiguraciji kontrolera.

Dalnjim istraživanjem tehnoloških podataka oko zavarivanja modela pluga Kalnik 900S, planirana je usporedba i eventualna poboljšanja u smislu brzine zavarivanja, uz usporedbu sa ostalim konkurenčnim proizvođačima robotskih sustava. Video pregled cijelog dostupan je pregledati na [5].

## 4. LITERATURA

- [1] [http://www.ijetae.com/files/Volume3Issue5/IJETAE\\_0513\\_126.pdf](http://www.ijetae.com/files/Volume3Issue5/IJETAE_0513_126.pdf) (dostupno 4.7.2015.)
- [2] [http://repozitorij.fsb.hr/928/1/18\\_03\\_2010\\_Vedran\\_Petrovic\\_Diplomski\\_rad.pdf](http://repozitorij.fsb.hr/928/1/18_03_2010_Vedran_Petrovic_Diplomski_rad.pdf) (dostupno 4.7.2015.)
- [3] <https://www.youtube.com/watch?v=KqEN78o0s3Y> (dostupno 4.7.2015.)
- [4] <https://www.youtube.com/watch?v=j882C7-wdEE> (dostupno 4.7.2015.)
- [5] <https://www.youtube.com/watch?v=bE0qSclsdk> (dostupno 4.7.2015.)