



Primjena industrijskih robota

Priručnik za fakultativnu nastavu iz
robotike za četvrte razrede smjera
tehničar za mehatroniku

Priručnik za učenike

Vinkovci, 2023.



www.esf.hr



Europska unija
"Zajedno do fondova EU"



EUROPSKI STRUKTURNI
I INVESTICIJSKI FONDovi



Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.

Magdalena
Kecić, 4.7.

17.06.2011

Sadržaj

Uvod u industrijske robote.....	4
Definicija industrijskih robota	6
Važnost automatizacije u industriji	8
Ishodi učenja i njihova primjena.....	10
Osnove upravljanja robotima.....	12
Upravljački uređaj robota	13
Postupak upisa programa u upravljački uređaj.....	16
Primjer upisa koda u industrijski robot UR3	19
Primjer upisa detaljnog programa u robot UR3.	20
Simulacija rada robota	22
Primjer simulacije rada robota	25
Identifikacija grešaka u programu	28
Identifikacija grešaka u kodu	31
Identifikacija grešaka u ponašanju robota	34
Spajanje opreme i pokretanje automatiziranog sustava...	37
Spajanje opreme na industrijskog robota	38
Pokretanje jednostavnog automatiziranog sustava...	43
Kreiranje upravljačkog programa za rad industrijskog robota	48
Primjer programa za robota UR5.....	50
Pozicije i stupnjevi slobode gibanja	52
Simulacija i izrada industrijskog robota	58
Simuliranje rada industrijskog robota.....	60
Izrada jednostavne robotske ruke i programa za simulaciju.....	66
Izrada jednostavne robotske ruke preko Arduino platforme.....	67

Sadržaj

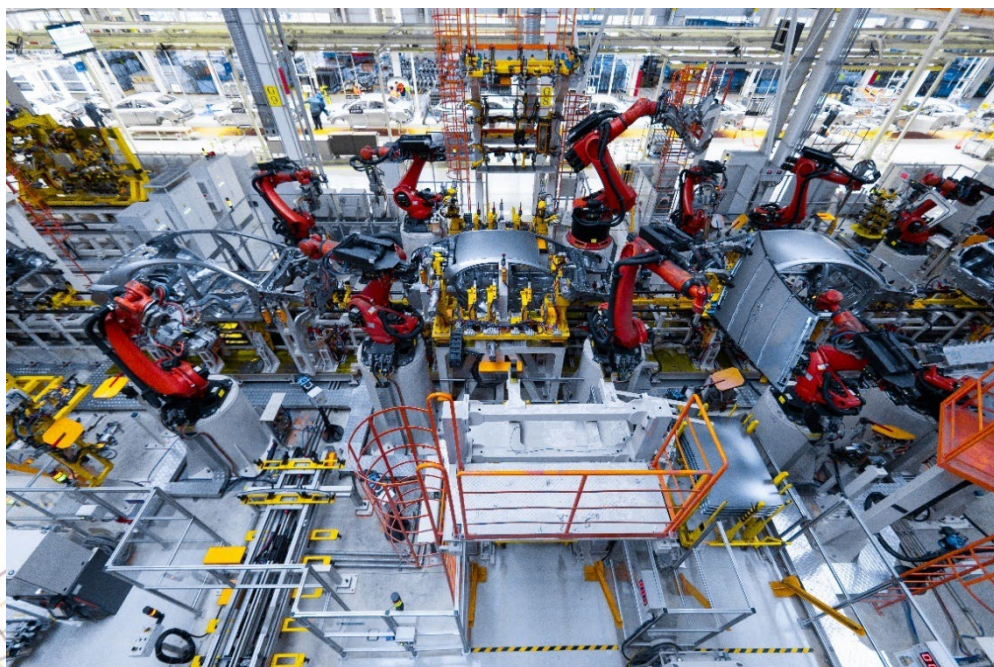
Izrada jednostavne robotske ruke preko PLC-a .	74
Pokretanja programa za izvršavanje specifičnih zadataka	80
Pokretanje programa industrijskog robota	81
Rješavanje problema i optimizacija rada robota.....	84
LITERATURA	86

Uvod u industrijske robote



U svijetu u kojem tehnološki napredak neprekidno oblikuje naše svakodnevne živote, jedna od ključnih promjena koja određuje suvremeno industrijsko okruženje jest primjena industrijskih robota. Industrijski roboti, sa svojom sposobnošću izvršavanja preciznih i repetitivnih zadataka, postali su kamen temeljac u revoluciji proizvodnje. Ovaj priručnik istražit će kako su industrijski roboti postali neizostavan dio modernih industrijskih procesa, mijenjajući radni svijet kroz automatizaciju.

U proteklom desetljeću svjedoci smo ubrzane primjene industrijskih robota u raznim sektorima, od proizvodnje do logistike. Ta revolucija nije samo rezultat tehnološkog napretka, već i odgovor na potrebu za povećanjem učinkovitosti, smanjenjem troškova proizvodnje te poboljšanjem ukupne kvalitete i preciznosti u industrijskim procesima. Industrijski roboti danas nisu samo mehaničke ruke koje obavljaju poslove, već sofisticirani sustavi opremljeni naprednim senzorima, umjetnom inteligencijom i mogućnošću prilagodbe dinamičkim radnim okolinama.



Slika 1. Prikaz robota u industriji



Osim što su ključni igrači u postizanju visoke razine proizvodne preciznosti, industrijski roboti također donose pozitivne promjene u radnom okruženju. Oslobađaju radnike od monotonih i opasnih zadataka, omogućujući im preusmjerenje svoje pažnje na zadatke koji zahtijevaju kreativnost, analitičke vještine i dublje razumijevanje procesa. No, s pojavom ovih tehnoloških inovacija, postavlja se pitanje: Kako upravljati tim automatiziranim silama? Kako razumjeti i programirati industrijske robote kako bi najbolje odgovarali potrebama suvremenih industrija? U nastavku ćemo istražiti ključne elemente upravljanja industrijskim robotima, od procesa programiranja do simulacije rada, utvrđivanja grešaka te konačno, primjene tih čuda tehnologije u svakodnevnom radu industrijskih sustava.

Definicija industrijskih robota



Industrijski roboti složeni su mehanički uređaji koji se koriste za izvođenje različitih zadataka u industrijskom okruženju, s ciljem povećanja učinkovitosti, preciznosti i produktivnosti u proizvodnim procesima. Ovi robotski sustavi opremljeni su naprednim senzorima, aktuatorima i kontrolnim sustavima koji im omogućuju izvođenje preciznih i repetitivnih radnji, često bez potrebe za stalnim nadzorom ljudskih operatera.



Slika 2. Industrijski roboti



Ključne karakteristike industrijskih robota uključuju:

- **Programabilnost** - industrijski su roboti programabilni i mogu izvršavati različite zadatke putem unaprijed definiranih skupova naredbi. Programiranje može biti izvedeno putem grafičkih sučelja, robotskih jezika programiranja ili čak učenjem putem demonstracije.
- **Prilagodljivost** - ovi roboti dizajnirani su kako bi bili prilagodljivi raznim proizvodnim okruženjima i zadacima. Mogu se lako rekonfigurirati kako bi obavljali različite zadatke unutar iste proizvodne linije.
- **Automatizacija** - glavna je svrha industrijskih robota automatizacija industrijskih procesa. To uključuje izvođenje repetitivnih zadataka, kao što su montaža, zavarivanje, pakiranje, paletizacija i slično, bez potrebe za stalnim ljudskim nadzorom.
- **Senzorika** - industrijski roboti opremljeni su različitim senzorima, poput kamera, senzora dodira, senzora pritiska i drugih, koji im omogućuju da reagiraju na svoje okruženje. To doprinosi povećanoj preciznosti i sposobnosti prilagodbe na promjene u radnom okolišu.
- **Visoka preciznost** - industrijski roboti sposobni su izvoditi vrlo precizne pokrete, često s tolerancijama od nekoliko milimetara. To ih čini izuzetno korisnima u industrijama gdje je potrebna visoka točnost, poput elektronike, medicinske opreme ili automobilske industrije.
- **Rad u opasnim okruženjima** - industrijski roboti često se koriste za izvođenje zadatka u opasnim okruženjima poput rukovanja opasnim tvarima, radu na visokim temperaturama ili izloženosti kemikalijama, čime se smanjuje rizik za ljudske radnike.

Navedene karakteristike čine industrijske robote ključnim partnerima u postizanju modernih standarda u industriji, optimizaciji proizvodnje te poboljšanju radnih uvjeta i sigurnosti.

Važnost automatizacije u industriji



Automatizacija u industriji predstavlja ključni čimbenik u suvremenom poslovnom svijetu, pridonoseći povećanju produktivnosti, smanjenju troškova, poboljšanju kvalitete proizvoda te stvaranju konkurentske prednosti. Važnost automatizacije u industriji odražava se kroz niz ključnih elemenata, uključujući učinkovitost, preciznost, smanjenje rizika za radnike te prilagodljivost na brze promjene u tržišnom okruženju.

Prvo i najvažnije, automatizacija značajno povećava produktivnost industrijskih procesa. Industrijski roboti i drugi automatizirani sustavi mogu neprekidno raditi 24 sata dnevno, sedam dana u tjednu, bez potrebe za odmorima ili stankama. Takvo neprekidno radno vrijeme povećava ukupni *output* proizvodnje, omogućujući tvrtkama brže zadovoljavanje potražnje na tržištu, isporuku proizvoda u kraćim rokovima i ostvarivanje veće dobiti.

Preciznost je još jedan ključan element automatizacije. Industrijski roboti sposobni su izvršavati zadatke s visokom točnošću, često iznad ljudskih mogućnosti. To je posebno važno u industrijama gdje je potrebna visoka preciznost, poput proizvodnje elektronike, medicinskih uređaja ili automobila. Precizni procesi ne samo da povećavaju kvalitetu proizvoda već i smanjuju otpad i troškove proizvodnje.

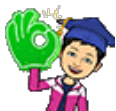
Automatizacija također smanjuje rizik za radnike, posebno u opasnim industrijama. Radnici često dolaze u kontakt s opasnim tvarima, visokim temperaturama ili teškim fizičkim naporima. Korištenjem industrijskih robota za izvođenje tih zadataka, rizik od ozljeda ili izloženosti opasnostima značajno se smanjuje. Radnici se mogu usredotočiti na nadzor i upravljanje automatiziranim sustavima, umjesto da obavljaju opasne zadatke.



Važnost automatizacije očituje se i u prilagodljivosti na brze promjene na tržištu. Prilagodljivi industrijski roboti omogućuju tvrtkama brzo mijenjanje proizvodnih linija ili prilagođavanje proizvoda prema promjenama u potražnji. Takva sposobnost prilagodbe omogućava tvrtkama konkurentnost u dinamičnom poslovnom okruženju.

U konačnici, automatizacija u industriji nije samo trend, već nužnost za postizanje visoke konkurentske prednosti. Tvrtke koje uspješno primjenjuju automatizaciju u svojim procesima imaju mogućnost brže reagirati na tržišne promjene, smanjiti troškove proizvodnje te ostvariti visoke standarde učinkovitosti i kvalitete. Ova transformacija industrije ne samo da oblikuje način na koji se proizvode dobra i usluge, već i doprinosi širem društvenom i ekonomskom napretku.

Ishodi učenja i njihova primjena



- Provesti postupak upisa programa u upravljački uređaj robota

Primjena: stručnjaci u industrijskoj robotici moraju biti sposobni učinkovito programirati robote kako bi izvršavali specifične zadatke. Ovaj ishod omogućava inženjerima uspješnu primjenu novih zadataka ili prilagodbu postojećih programa prema promjenama u proizvodnji.

- Provjeriti rad robota simulacijom

Primjena: simulacije omogućuju inženjerima testiranje i analizu radnih karakteristika robota prije nego što se uvrste u stvarno radno okruženje. Ovaj ishod pomaže u otkrivanju potencijalnih problema i optimizaciji rada robota.

- Utvrditi grešku u programu za rad industrijskog robota

Primjena: stručnjaci moraju biti sposobni dijagnosticirati i rješavati probleme u programima robota kako bi osigurali neprekidan i pouzdan rad. Ta vještina postaje ključna u održavanju i servisiranju industrijskih robota.

- Izvesti spajanje opreme i pokrenuti jednostavan automatizirani sustav s robotom

Primjena: taj ishod omogućava inženjerima primjenu robota u veće automatiziranim sustavima, poboljšavajući tako ukupnu produktivnost i učinkovitost proizvodnje.

- Kreirati upravljački program za rad industrijskog robota

Primjena: stručnjaci moraju biti u stanju razvijati prilagođene upravljačke programe koji odgovaraju specifičnim potrebama proizvodnje. Ovaj ishod ključan je za prilagodbu rada robota različitim zadacima.



- Objasniti pozicije i stupnjeve slobode gibanja robota

Primjena: razumijevanje pozicija i stupnjeva slobode pomaže inženjerima u dizajniranju i programiranju robota tako da se optimalno koriste u određenom radnom okruženju.

- Simulirati rad i izraditi jednostavnog industrijskog robota i/ili robotsku ruku

Primjena: simulacije i izrada prototipa omogućuju inženjerima eksperimentalno testiranje različitih dizajna i konfiguracija robota prije nego što se krene u proizvodnju. Ovo ubrzava razvoj novih robotičkih sustava.

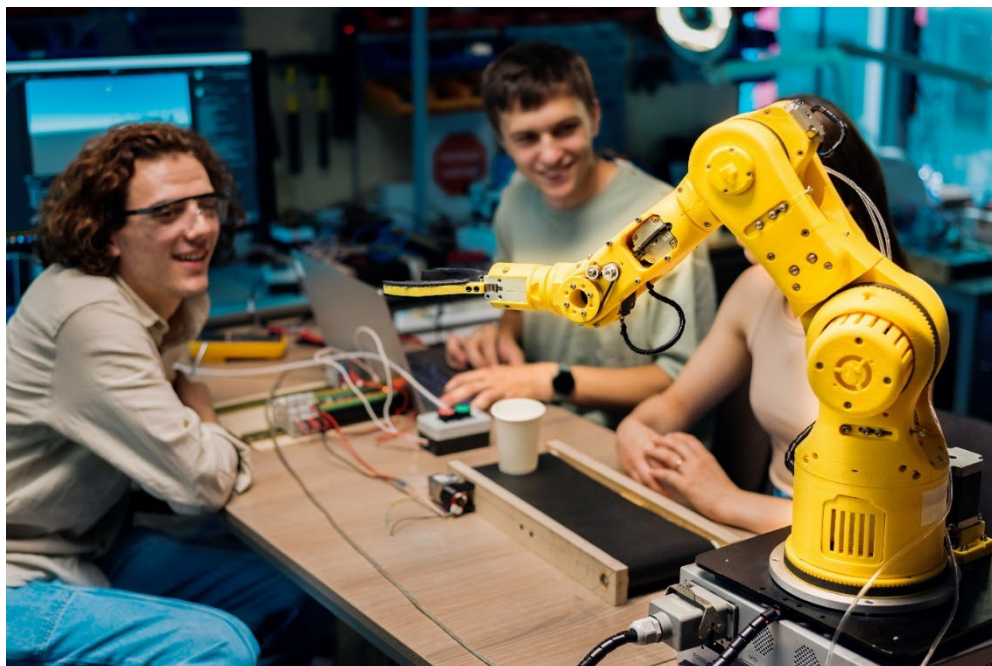
- Pokrenuti program industrijskog robota i/ili robotske ruke za izvršavanje specifičnog zadatka prema projektnom zadatku

Primjena: ishod omogućava stručnjacima uspješnu primjenu razvijenih programa u stvarnom radnom okruženju, izvodeći specifične zadatke prema potrebama industrijskih procesa.

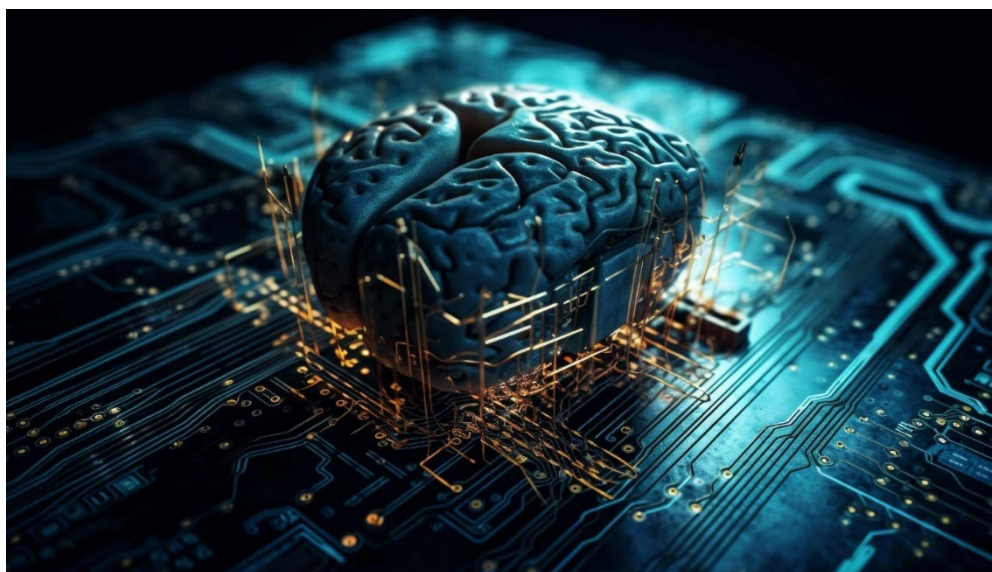
Ukupno, ovi ishodi učenja i njihove primjene odražavaju sveobuhvatnost znanja i vještina potrebnih za uspješno upravljanje industrijskim robotima u suvremenim proizvodnim postrojenjima.



Osnove upravljanja robotima



Slika 3. Upravljanje industrijskim robotom



Slika 4. Prikaz „mozga“ robota

Upravljački uređaj robota



Upravljački uređaj robota predstavlja ključnu komponentu koja omogućuje programiranje, kontrolu i upravljanje kretanjem industrijskog robota. Ova sofisticirana oprema osigurava da robot precizno izvršava zadatke i optimalno reagira na promjene u okolini. Evo detaljnog opisa ključnih elemenata upravljačkog uređaja robota:

- Računalni sustav
 - **Glavni procesor** - upravljački uređaj robota opremljen je snažnim glavnim procesorom koji interpretira upravljačke naredbe, izračunava pokrete i upravlja različitim dijelovima robota.
 - **Memorija** - uređaj ima memoriju koja pohranjuje upravljačke programe, konfiguracijske postavke te podatke o radu i parametre sustava.
- Sučelje za programiranje
 - **Programski jezik** - upravljački uređaji podržavaju različite programske jezike prilagođene programiranju robota, uključujući popularne kao što su *RAPID*, *KRL* ili grafički jezici poput *Blockly*.
 - **Grafičko sučelje** - mnogi uređaji imaju grafička sučelja koja omogućuju inženjerima programiranje robota putem vizualnog sučelja, olakšavajući proces korisnicima bez dubokih programerskih vještina.



- Senzori
 - **Položajni senzori** - upravljački uređaj koristi senzore kako bi stalno pratio položaj i orijentaciju robota. To je ključno za precizno izvođenje zadataka i izbjegavanje sudara s okolinom.
 - **Senzori za dodir** - osjetljivi senzori na robotu omogućuju mu detekciju dodira s okolinom, što je posebno važno za sigurnu suradnju s ljudima ili drugim objektima u radnom okruženju.
- Aktuatori i motorni pogoni
 - **Motorni pogoni** - uređaj kontrolira različite motore koji pokreću zglobove, prste ili druge dijelove robota. Precizna kontrola motora omogućuje izvođenje složenih pokreta i zadataka.
 - **Elektromagnetski ventili** - ako je robot opremljen pneumatskim ili hidrauličkim sustavima, upravljački uređaj kontrolira elektromagnetske ventile za precizno upravljanje tlakom i protokom fluida.
- Komunikacija s perifernim uređajima
 - **Ethernet ili bežične mreže** - upravljački uređaj može komunicirati s drugim sustavima ili perifernim uređajima putem *Ethernet* veze ili bežičnih mreža, omogućujući integraciju s cjelokupnim industrijskim sustavom.
 - **I/O priključci** - omogućuju povezivanje s dodatnom opremom poput senzora, grijača, svjetala ili drugih perifernih uređaja.



- Sigurnosne značajke
 - **Sigurnosni sustavi** - upravljački uređaj uključuje sigurnosne značajke poput sustava za zaustavljanje u hitnim situacijama, ograničavanja brzine i pokreta te sustava za praćenje i sprječavanje kolizija.
- Prilagodljivost
 - **Mogućnosti nadogradnje** - upravljački uređaji često imaju mogućnost nadogradnje kako bi podržavali nove tehnologije, poboljšavali radne karakteristike ili dodavali nove funkcionalnosti.

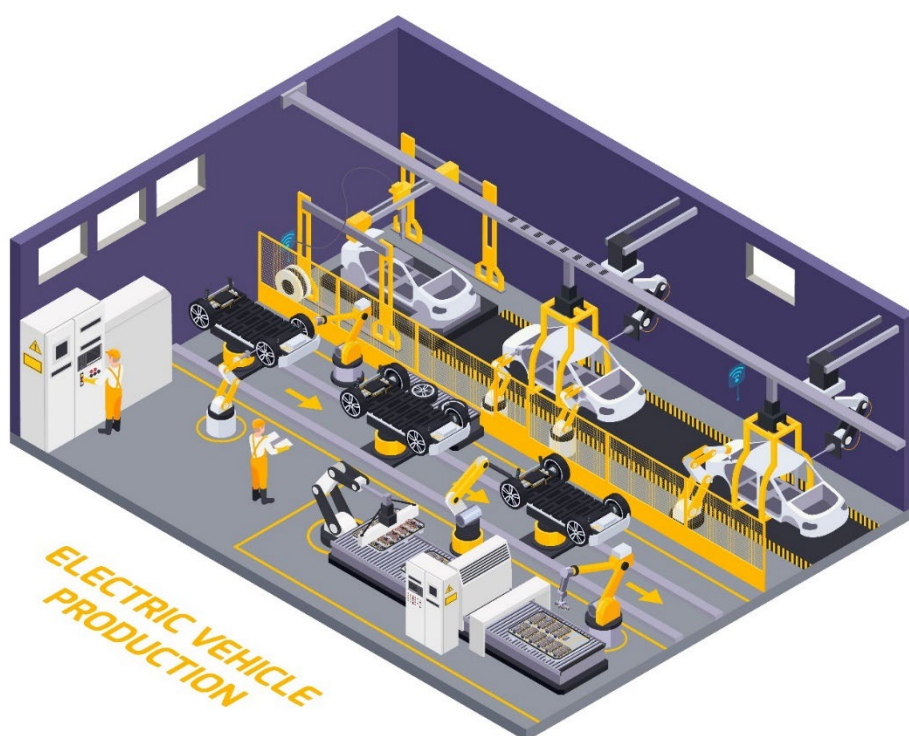
Upravljački uređaji robota predstavljaju srce automatiziranog sustava, omogućujući inženjerima programiranje, kontrolu i optimizaciju rad robota za različite industrijske primjene. Njihova složena funkcionalnost ključna je za postizanje visokih standarda učinkovitosti i preciznosti u industriji.



Postupak upisa programa u upravljački uređaj



Postupak upisa programa u upravljački uređaj industrijskog robota ključan je korak u osiguravanju da robot izvršava zadatke prema specifikacijama. Ovaj proces uključuje unos, verifikaciju i spremanje programa koji sadrži naredbe za upravljanje robotom. Evo općeg postupka upisa programa:



Slika 5. Prikaz robota u autoindustriji



- Priprema radnog okruženja
 - Prije početka upisa programa, osigurajte da je radno okruženje sigurno i prikladno za unos podataka. Ovo uključuje provjeru jesu li svi važni sustavi uključeni i spremni za rad.
- Pokretanje upravljačkog uređaja
 - Upravljački uređaj robota treba biti uključen i spreman za rad. Ovo može uključivati pokretanje posebnog softverskog sučelja ili programskog okruženja koje omogućuje unos programa.
- Izbor programskog jezika ili sučelja
 - Odaberite programski jezik ili grafičko sučelje koje će se koristiti za pisanje programa. Ovisno o modelu robota i proizvođaču upravljačkog uređaja, moguće je koristiti različite jezike ili alate za programiranje.
- Programiranje robota
 - Unesite naredbe programa koristeći odabrani programski jezik ili grafičko sučelje. Programiranje obuhvaća definiranje pokreta, pozicija, logičkih uvjeta i ostalih operacija koje će robot izvršiti.
- Simulacija programa
 - Prije nego što se program prenese na stvarnoga robota, preporučljivo je izvršiti simulaciju programa. Simulacija omogućuje provjeru ispravnosti programa, otkrivanje mogućih problema ili konflikata te optimizaciju radnih karakteristika prije stvarnog prijenosa.





- Verifikacija i ispravci
 - Provjerite program kako biste osigurali da zadatci budu precizno definirani i da ne postoje logičke greške. Ako se utvrde problemi, izvršite potrebne ispravke kako biste osigurali ispravno ponašanje robota.
- Prijenos programa na upravljački uređaj
 - Kada je program verificiran i ispravljen, prijedite na prijenos programa na upravljački uređaj robota. Taj korak obično uključuje povezivanje računala s upravljačkim uređajem i prijenos programa putem odgovarajućih sučelja, poput USB-a, mrežne veze ili drugih priključaka.
- Spremanje programa
 - Nakon uspješnog prijenosa programa, osigurajte da se program sprema na odgovarajuće mjesto na upravljačkom uređaju. To omogućuje jednostavan pristup i ponovnu upotrebu programa kad god je to potrebno.
- Testiranje na stvarnom robotu
 - Konačno, izvršite testiranje programa na stvarnom robotu. Pratite ponašanje robota, utvrdite eventualne probleme i optimizirajte radne karakteristike ako je potrebno.

Postupak upisa programa u upravljački uređaj temeljni je korak u korištenju industrijskog robota u proizvodnom okruženju, a preciznost i ispravnost programa ključni su za postizanje željenih rezultata i učinkovitosti rada robota.



Primjer upisa koda u industrijski robot UR3

Za programiranje robota, u ovom slučaju *Universal Robots UR3*, često se koristi poseban robotski jezik poput *RAPID*-a. *RAPID* je jezik koji je dizajniran za jednostavno i brzo programiranje *Universal Robots* robota. Evo primjera jednostavnog *RAPID* programa za *UR3* koji bi mogao pokrenuti nekoliko osnovnih pokreta:

```
MODULE MainModule
```

```
! Definiranje varijabli
VAR num counter := 0
```

```
! Glavna procedura
P1()
! Prikaz poruke
Popup("Programiranje robota UR3")
! Postavljanje početne pozicije
MoveJ homePose
! Izvršavanje petlje
WHILE counter < 5 DO
! Pomak robota
MoveJ targetPose
! Povećavanje brojača
counter := counter + 1
ENDWHILE
END P1
```

```
! Definiranje početne pozicije robota
PROC homePose()
MoveJ [0, -90, 90, -90, 90, 0] ! Primjer koordinata
zglobova
END PROC
```

```
! Definiranje ciljne pozicije robota
PROC targetPose()
MoveJ [180, -90, 90, -90, 90, 0] ! Primjer koordinata
zglobova
END PROC
```

```
ENDMODULE
```

Primjer upisa detaljnog programa u robot UR3



Ovaj primjer programa koristi jednostavan *RAPID* kod kako bi pomakao *UR3* robota između početne pozicije i ciljne pozicije kroz petlju. Naravno, stvarni program ovisit će o specifičnim zadacima koje želite da robot izvršava.

Napomena: koordinate zglobova (*joint coordinates*) navedene su kao primjer i mogu se prilagoditi stvarnim pozicijama robota. Osim toga, koristite ovaj primjer kao polaznu točku i prilagodite ga prema vašim potrebama i specifičnostima radnog okruženja.

Kad govorimo o detaljnijem primjeru upisa programa za *Universal Robots UR3*, mogli bismo uključiti niz pokreta, uvjetovanih izvršavanja, upotrebu podataka i funkcija. Prilagodba programa ovisi o specifičnim zadacima i potrebama aplikacije. Evo jednog detaljnijeg primjera programa koji uključuje pozicioniranje, uvjetno izvršavanje i korištenje podataka:

MODULE MainModule

```
! Definiranje varijabli
VAR num counter := 0
VAR jointtarget homePose := [0, -90, 90, -90, 90, 0]
VAR jointtarget targetPose := [180, -90, 90, -90, 90, 0]
```

```
! Glavna procedura
P1()
! Prikaz poruke
Popup("Programiranje robota UR3")
! Postavljanje početne pozicije
MoveJ homePose
! Izvršavanje petlje
WHILE counter < 5 DO
! Pomak robota prema ciljnoj poziciji
MoveJ targetPose
! Povećavanje brojača
counter := counter + 1
```



```

! Uslovno izvršavanje
IF counter = 2 THEN
    Popup("Robota je sada u drugoj iteraciji petlje.")
ENDIF
ENDWHILE
! Povratak u početnu poziciju
MoveJ homePose
END P1

! Definiranje procedure za postavljanje početne pozicije
robota
PROC homePose()
    MoveJ homePose
END PROC

! Definiranje procedure za postavljanje ciljne pozicije
robota
PROC targetPose()
    MoveJ targetPose
END PROC

ENDMODULE
    
```

Ovaj primjer programa koristi petlju **WHILE** za ponavljanje određenog broja puta, uvjetno izvršavanje s **IF** naredbom, te korištenje varijabli za definiranje početnih i ciljnih pozicija robota. Naravno, ovaj program osnovni je primjer i može se prilagoditi složenijim zadacima prema potrebama. Važno je uzeti u obzir sigurnosne elemente i dodati odgovarajuće mjere opreza prilikom programiranja i izvršavanja takvih programa na robotu.

Simulacija rada robota



Simulacija rada robota ključan je korak u razvoju i testiranju programa prije nego što se program prenese na stvarnog robota. Simulacije omogućuju inženjerima analiziranje i optimiziranje ponašanja robota u virtualnom okruženju prije nego što dođe do fizičkog spajanja. Ovdje su detalji o simulaciji rada robota:

- Odabir simulacijskog softvera
 - Prvi korak u simulaciji rada robota odabir je odgovarajućeg simulacijskog softvera. Postoje različiti alati i platforme koje omogućuju virtualno programiranje i testiranje robotskih programa, a mnogi od njih integrirani su s robotskim programskim jezicima poput *RAPID*-a za *Universal Robots*.
- Unos modela robota
 - Simulacijski softver treba sadržavati točan model stvarnog robota. Model uključuje informacije o geometriji robota, kinematici, dinamici, senzorima i drugim karakteristikama. Unos tih podataka omogućava simulacijskom softveru precizno repliciranje ponašanja stvarnoga robota.
- Programiranje u virtualnom okruženju
 - Korisnici mogu programirati zadatke i naredbe unutar simulacijskog okruženja koristeći isti jezik koji će se kasnije koristiti za stvarnoga robota. Na primjer, ako koristite *RAPID* za *Universal Robots*, simulaciju možete programirati koristeći *RAPID* jezik u simulacijskom softveru.



- Postavljanje radnog okruženja
 - Definiranje radnog okruženja uključuje postavljanje virtualnog prostora u kojem će se robot kretati. To uključuje postavljanje prepreka, radnih stanica i drugih elemenata koji će replicirati stvarno radno okruženje.
- Testiranje i analiza
 - Program se može testirati unutar simulacije kako bi se utvrdili potencijalni problemi, kolizije ili nesigurnosti u izvršavanju zadataka. Analiza rezultata simulacije pomaže u optimizaciji programa i smanjenju rizika od grešaka pri stvarnoj primjeni.
- Praćenje ponašanja robota
 - Simulacija omogućuje praćenje ponašanja robota tijekom izvođenja programa. To uključuje praćenje putanje kretanja, brzine, točnosti pozicioniranja i drugih parametara koji određuju radne karakteristike robota.
- Provođenje sigurnosnih protokola
 - Simulacija je također korisna za provjeru i provođenje sigurnosnih protokola. To uključuje simulaciju zaštite od kolizija, praćenje kretanja prema definiranim granicama i druge sigurnosne mjere.



- Optimizacija radnih karakteristika
 - Na temelju rezultata simulacije, programeri mogu optimizirati program kako bi poboljšali radne karakteristike robota. To uključuje promjene u brzini, putanjama kretanja ili prilagodbama programa kako bi se postigle željene specifikacije zadatka.
- Dokumentiranje rezultata
 - Zadnji je korak u simulaciji dokumentiranje rezultata. To uključuje bilježenje problema koji su otkriveni tijekom simulacije, prilagodbe programa i ostale važne informacije koje će pomoći pri daljnjem razvoju i uvrštavanju.

Simulacija rada robota od izuzetne je važnosti jer omogućava inženjerima da smanje rizik od grešaka, optimiziraju radne karakteristike i smanje vrijeme potrebno za primjenu na stvarnom robotu.



Primjer simulacije rada robota



Slika 6. Prikaz simuliranja rada robota pomoću virtualne stvarnosti

Simulirano radno okruženje

Pretpostavimo da se simulira rad industrijskog robota u proizvodnji elektroničkih uređaja, poput pametnih telefona. Radno okruženje uključuje radnu stanicu opremljenu potrebnim alatima, senzorima i paletama s komponentama.

Programiranje robotskog sustava

Unutar simulacijskog softvera, inženjeri koriste *RAPID* programski jezik za programiranje industrijskog robota. Primjerice, program bi uključivao precizne pokrete robota za uzimanje komponenata s paleta, postavljanje na određeno mjesto u proizvodnom procesu te zatim obavljanje montaže.



```
MODULE MontazniProces
```

```
VAR jointtarget startPos := [0, -90, 90, -90, 90, 0]
```

```
P1()  
  MoveJ startPos  
  ! Uzimanje komponente s palete  
  PickComponent()  
  ! Montaža komponente  
  AssembleComponent()  
  ! Povratak u početnu poziciju  
  MoveJ startPos  
END P1
```

```
PROC PickComponent()  
  ! Simulacija pokreta za uzimanje komponente  
  ! ...  
  ! Provjera senzora i prikupljanje podataka  
  ! ...  
END PROC
```

```
PROC AssembleComponent()  
  ! Simulacija montažnog postupka  
  ! ...  
  ! Provjera ispravnosti montaže  
  ! ...  
END PROC
```

```
ENDMODULE
```



Postavljanje radne stanice

Unutar simulacije, radna je stanica konfigurirana s točnim dimenzijama i karakteristikama kako bi se što vjernije repliciralo stvarno radno okruženje. To uključuje postavljanje virtualnih senzora, prikaz stanja opreme i simuliranje reakcija na pokrete robota.

Analiza i optimizacija

Simulacija omogućuje detaljnu analizu radnih karakteristika robota. Inženjeri mogu pratiti vrijeme izvođenja zadatka, otkriti potencijalne kolizije, analizirati kretanje zglobova i brzinu izvođenja operacija. Na temelju rezultata analize, mogu se provesti optimizacije programa kako bi se poboljšala učinkovitost i smanjila mogućnost pogrešaka.

Prijenos na stvarnog robota

Kada simulacija daje zadovoljavajuće rezultate, program se prenosi na stvarnog robota. Taj prijenos uključuje kalibraciju stvarnog robota prema simuliranom modelu kako bi se osigurala dosljednost u ponašanju i preciznost izvođenja zadataka.

Primjer simulacije rada industrijskog robota u kontekstu montaže proizvodne linije ilustrira ključnu ulogu simulacija u optimizaciji proizvodnih procesa. Kroz precizno programiranje, postavljanje radne stanice i analizu radnih karakteristika, simulacije omogućuju inženjerima siguran i učinkovit razvoj robotskih programa prije njihove primjene u stvarnim industrijskim okruženjima.

Identifikacija grešaka u programu



Identifikacija grešaka u programima industrijskih robota zahtijeva sustavan i precizan pristup.



Slika 7. Prikaz humanoidnog robota

Evo nekoliko konkretnijih strategija za pronalaženje grešaka u robotičkim programima:

- Sistematsko testiranje
 - Razvijanje temeljitog plana testiranja koji obuhvaća sve elemente programa ključno je za otkrivanje grešaka. Testiranje bi trebalo uključivati različite scenarije kretanja, različite zadatke i uvjete rada. Pratite ponašanje robota tijekom testiranja kako biste otkrili moguće greške.



- Proučavanje simulacija
 - Simulacije pružaju sigurno okruženje za testiranje programa bez rizika od oštećenja opreme ili ozljeda. Proučavanjem rezultata simulacija, inženjeri mogu otkrivati i rješavati probleme prije nego što program dosegne stvarnog robota. Analiza simulacija omogućuje dublje razumijevanje ponašanja robota u različitim scenarijima.
- Korištenje senzora i detektora kolizija
 - Senzori na robotu igraju ključnu ulogu u otkrivanju grešaka. Korištenjem senzora za praćenje položaja alata, sila i drugih varijabli, moguće je utvrditi odstupanja od očekivanog ponašanja. Utvrđivanje kolizija pomoću senzora omogućuje otkrivanje bilo kakvih fizičkih sukoba tijekom izvođenja programa.
- Analiza logova i izvještaja
 - Detaljna analiza logova i izvještaja generiranih tijekom izvođenja programa pomaže u utvrđivanju specifičnih točaka u kojima dolazi do problema. Ti zapisi sadrže podatke o svakom koraku izvođenja programa, uključujući i eventualna upozorenja ili pogreške koje se događaju.





- Provjera sustavnosti koda
 - Revizija koda programa s naglaskom na pravilnosti i dosljednosti može pomoći u otkrivanju potencijalnih grešaka. Provjera pravilnosti u sintaksi, ispravnosti naredbi i dosljednosti u upotrebi varijabli važna je u osiguravanju točnosti i ispravnosti programa.
- Sigurnosna isključivanja i nastavak izvršavanja
 - Primjena sigurnosnih protokola koji uključuju automatsko isključivanje sustava u slučaju potencijalne opasnosti može spriječiti ozbiljne incidente. Također, omogućavanje sigurnog nastavka izvršavanja programa nakon rješavanja manjih grešaka doprinosi neprekidnom radu robota.
- Praćenje parametara robota
 - Sustavno praćenje parametara robota, poput temperature motora, napona i brzine, može otkriti anomalije koje ukazuju na potencijalne probleme. Ti podatci mogu biti korisni za predviđanje održavanja i utvrđivanja grešaka prije nego što dođe do ozbiljnijih problema.

Pristup koji kombinira sustavno testiranje, korištenje simulacija, analizu senzora i reviziju koda pruža cjelovitu strategiju za otkrivanje grešaka u programima industrijskih robota. Ova kombinacija alata i metoda omogućuje inženjerima otkrivanje, analiziranje i rješavanje potencijalnih problema, čime se osigurava sigurnost i učinkovitost robotičkih sustava.



Utvrđivanje grešaka u kodu



Analiziranje pogrešaka u kodu industrijskih robota ključno je za osiguravanje stabilnog i sigurnog izvođenja programa. Evo koraka i pristupa koji se često koriste pri analizi grešaka u kodu:

- Pregled sintakse i strukture koda
 - Počnite pregledom sintakse i strukture koda. Provjerite ima li grešaka poput nezatvorenih zagrada, pogrešnih varijabli, ili sintaktičkih grešaka koje bi mogle uzrokovati neispravno izvođenje programa.
- Korištenje *Debugging* alata
 - *Debugging* alati omogućuju korisnicima da korak po korak prate izvođenje programa, prate vrijednosti varijabli i utvrde točne točke u kojima dolazi do pogrešaka. Korištenje alata poput *breakpointova*, *step-by-step* izvođenja i inspekcije varijabli pomaže u pronalaženju grešaka.
- Analiza izuzetaka i pogrešaka
 - Izuzetci i pogreške koji se javljaju tijekom izvođenja programa često pružaju ključne informacije o prirodi problema. Analizirajte izuzetke i prateće poruke kako biste dobili uvid u točno mjesto i uzrok greške.



- Upotreba *Logginga*
 - Ugrađivanje logova u program može pomoći u praćenju tijeka izvođenja i utvrđivanju problema. Detaljni logovi često sadrže korisne informacije o stanju varijabli, koracima izvođenja i bilo kojim pogreškama koje se događaju.

```
# Primjer koda s ugrađenim logovima
def neka_funkcija():
    try:
        # ... neki kod ...
    except Exception as e:
        print(f'Greška: {e}')
        # Logiraj dodatne informacije
        with open('log.txt', 'a') as log_file:
            log_file.write(f'Greška: {e}\n')
```

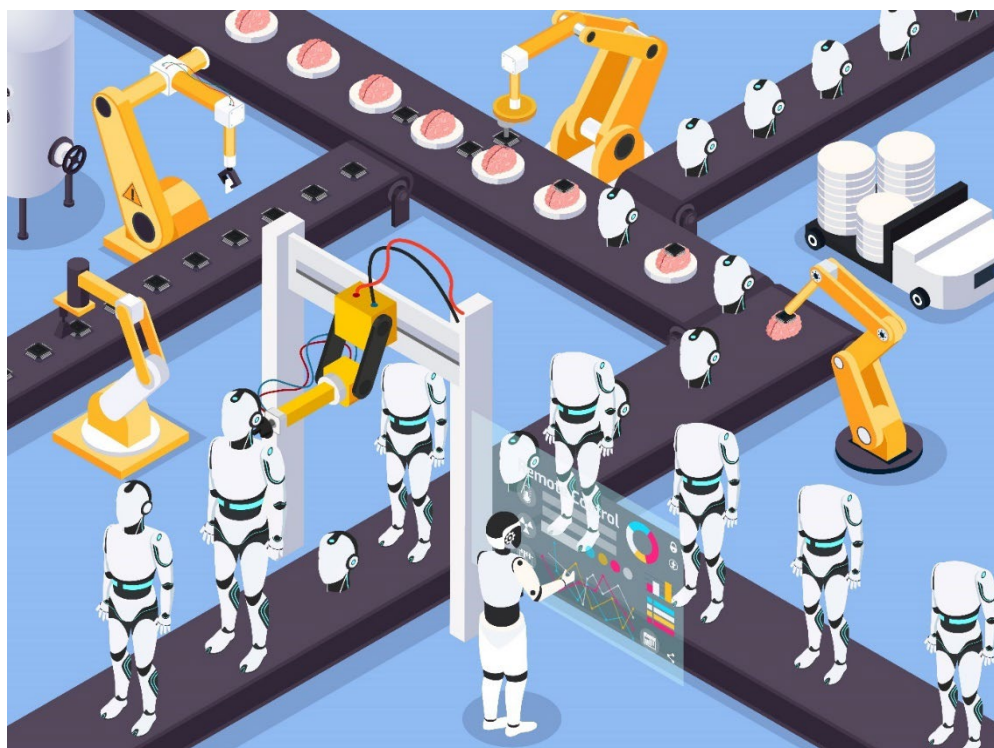
- Provjera ulaznih podataka
 - Provjerite kako program obrađuje ulazne podatke. Neispravna manipulacija ili interpretacija ulaznih podataka može rezultirati greškama. Proučite sve dijelove koda gdje se obrađuju vanjski podatci kako biste bili sigurni da je manipulacija podacima ispravna.
- Proučavanje algoritama i logike programa
 - Analizirajte algoritme i logiku programa kako biste provjerili je li ispravno uvrštena. Obratite pažnju na petlje, uvjete i ostale dijelove koda koji utječu na tijek izvođenja programa.



- Savjetovanje s dokumentacijom i specifikacijama
 - Provjerite je li program u skladu s dokumentacijom i specifikacijama. Usporedite kod s očekivanjima definiranim u dokumentaciji kako biste utvrdili eventualna odstupanja ili pogreške u primjeni.
- Testiranje u različitim uvjetima
 - Testirajte program u različitim uvjetima kako biste provjerili stabilnost i otpornost koda. Simulirajte različite scenarije rada kako biste otkrili moguće slabosti i pogreške koje se mogu pojaviti u stvarnim situacijama.



Utvrđivanje grešaka u ponašanju robota



Slika 8. Animirani prikaz sklapanja robota

Analiziranje pogrešaka u ponašanju industrijskih robota važan je korak u održavanju sigurnog i učinkovitog radnog okruženja.

Evo nekoliko koraka i pristupa koji se često koriste pri analizi pogrešaka u ponašanju robota:

- Praćenje robota u stvarnom vremenu
 - Aktivno praćenje rada robota u stvarnom vremenu pomaže u brzom otkrivanju mogućih problema. To uključuje praćenje kretanja zglobova, promjena u brzini i bilo kakvih neočekivanih reakcija tijekom izvođenja programa.



- Korištenje senzora za utvrđivanje odstupanja
 - Senzori na robotima igraju ključnu ulogu u utvrđivanju odstupanja u ponašanju. Senzori za praćenje položaja, brzine, sile i druge varijable omogućuju precizno mjerenje i otkrivanje mogućih anomalija.
- Analiza logova i pogrešaka
 - Detaljna analiza logova i izvještaja koji se generiraju tijekom rada robota pruža informacije o mogućim pogreškama. Proučavanjem tih podataka možete utvrditi trenutke u kojima dolazi do problema te analizirati uzroke tih problema.
- *Korak po korak pregled* izvođenja programa
 - Ako se pojave pogreške, *korak po korak pregled* izvođenja programa pomoći će utvrditi točnu naredbu ili operaciju koja uzrokuje problem. Takav pristup omogućuje preciznu lokalizaciju pogreške.
- Provjera rada senzora sigurnosti
 - Senzori sigurnosti, poput senzora prisutnosti ili senzora sudara, igraju ključnu ulogu u zaštiti radnika i opreme. Provjerite jesu li senzori ispravni i otkrivaju li ispravno sve potencijalne opasnosti.
- Upotreba sustava za poništavanje
 - Roboti često imaju ugrađene sustave za poništavanje u slučaju opasnosti. Provjerite jesu li ti sustavi ispravno konfigurirani i aktivirani kako bi brzo reagirali u slučaju problema.



- Provjera kalibracije i preciznosti
 - Ako robot koristi vizualne ili senzorske sustave za pozicioniranje, provjerite jesu li kalibrirani kako bi osigurali točnost. Neispravna kalibracija može rezultirati nepreciznim ponašanjem robota.
- Testiranje u simulacijama
 - Korištenje simulacija za testiranje ponašanja robota pod različitim uvjetima može pomoći u otkrivanju problema prije nego što program dosegne stvarnog robota. Simulacije omogućuju precizno reproduciranje radnog okruženja i situacija.
- Proučavanje sigurnosnih protokola
 - Ako robot koristi sigurnosne protokole, proučite kako su uvršteni. To uključuje sigurnosne zone, ograničenja brzine i druge mjere koje se aktiviraju u slučaju opasnosti.
- Savjetovanje s tehničkom podrškom
 - Ako se problemi u ponašanju robota ne mogu riješiti unutar tima, savjetovanje s tehničkom podrškom proizvođača ili stručnjacima za industrijske robote može pružiti dodatne perspektive i rješenja.

Spajanje opreme i pokretanje automatiziranog sustava



Slika 9. Prikaz jednakih robota s različitim alatima



Slika 10. Prikaz robota s alatom za prskanje

Spajanje opreme na industrijskog robota



U suvremenim industrijskim postrojenjima spajanje različite opreme s industrijskim robotima predstavlja ključni korak prema postizanju potpune automatizacije proizvodnje. Spajanje raznovrsnih uređaja na robota otvara vrata visokoj učinkovitosti, preciznosti i prilagodljivosti u radu, ali istovremeno predstavlja izazove koji zahtijevaju pažljivo planiranje i primjenu. Ovo poglavlje istražiti će značaj spajanja opreme na industrijskog robota, naglasiti ključne koristi te istražiti izazove s kojima se susreću inženjeri prilikom ovog procesa.

Raznolikost opreme u industriji: industrijski roboti postali su središnji elementi proizvodnih postrojenja koja se suočavaju s izazovima rastuće složenosti zadatka. Spajanjem različite opreme, poput senzora, alata ili dodatnih robotskih ruku, postiže se potpuna integracija koja omogućuje izvođenje širokog spektra zadataka unutar istog radnog okruženja.

Ključne prednosti spajanja opreme na robota:

- **Povećana funkcionalnost** - primjena dodatne opreme proširuje funkcionalnosti robota, omogućujući izvođenje više zadataka unutar istog proizvodnog ciklusa.
- **Povećana učinkovitost** - spajanje robota s dodatnom opremom često rezultira smanjenjem vremena ciklusa, povećavajući proizvodni kapacitet i smanjujući ukupne troškove proizvodnje.
- **Prilagodljivost i višenamjenska upotreba** - spajanje omogućuje brzu prilagodbu robota različitim zadacima i promjenama u proizvodnji, čineći ga višenamjenskim i prilagodljivim resursom.



- **Poboljšana kvaliteta proizvoda** - uvođenjem dodatnih senzora za nadzor i kontrolu procesa postiže se visoka preciznost u izvođenju zadataka, što rezultira poboljšanom kvalitetom proizvoda.

Izazovi u procesu spajanja opreme:

- **Kompatibilnost i standardizacija** - različiti proizvođači opreme koriste različite standarde i protokole komunikacije, što može predstavljati izazov pri povezivanju opreme na robotički sustav.
- **Potrebna dodatna prilagodba i kalibracija** - integracija zahtijeva dodatnu prilagodbu kako bi se osiguralo ispravno funkcioniranje i kalibracija svih komponenata.
- **Sigurnosni elementi** - povećavanjem složenosti sustava, posebice dodatnim alatima i pokretnim dijelovima, raste potreba za pažljivijim rukovanjem sigurnosnim elementima kako bi se spriječile ozljede i oštećenja.
- **Potreba za dodatnim sustavima kontrole** - uvrštavanje dodatne opreme može zahtijevati dodatne sustave kontrole kako bi se osiguralo koordinirano i usklađeno ponašanje svih komponenata.

U suvremenim industrijskim okruženjima integracija opreme na industrijske robote predstavlja ključni korak prema postizanju visokog stupnja automatizacije i optimizacije proizvodnih procesa. Različiti načini spajanja opreme omogućuju inženjerima prilagodljivost, poboljšanu funkcionalnost i učinkovitost robota u izvođenju širokog spektra zadataka. Ovaj priručnik istražuje nekoliko inovativnih pristupa spajanju opreme na industrijske robote, ističući prednosti i izazove svakog od njih.



Integracija i načini spajanja opreme:

- Modularna integracija
 - Koncept modularne integracije omogućuje dodavanje i uklanjanje modula opreme prema potrebi. Svaki modul predstavlja određenu funkcionalnost, poput dodatnih alata, senzora ili kamere. Ovaj pristup omogućuje brzu prilagodbu robota različitim zadacima bez potrebe za znatnim promjenama u osnovnom sustavu.
- *Ethernet* i industrijski protokoli komunikacije
 - Korištenje industrijskih protokola komunikacije, poput *Ethernet/IP*-a, *PROFINET*-a ili *Modbus*a, pruža djelotvorno sučelje za povezivanje robota s drugim uređajima. Ovaj standardizirani pristup olakšava komunikaciju između robota i drugih komponenata u proizvodnom okruženju.
- Upotreba *ROS (Robot Operating System)*
 - *ROS* pruža otvoreni okvir za razvoj i integraciju robotskih sustava. Kroz *ROS*, inženjeri mogu jednostavno spojiti različite senzore, kamere, alate i druge uređaje na industrijske robote. Ova otvorena platforma potiče suradnju između različitih uređaja i olakšava razvoj naprednih robotskih sustava.



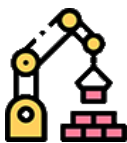
- Upotreba *IO-Link* tehnologije
 - *IO-Link* standardizirana je tehnologija za komunikaciju s industrijskim senzorima i aktuatorima. Spajanje robota s uređajima putem *IO-Link*-a omogućuje brzu i jednostavnu konfiguraciju, povećava prilagodljivost sustava i omogućuje praćenje stanja opreme u stvarnom vremenu.
- Automatsko prepoznavanje alata
 - Primjena sustava za automatsko prepoznavanje alata omogućuje robotima samostalno identificiranje i prilagodbu različitim alatima. Ovo je posebno korisno u postavkama gdje robot mora raditi s više alata ili obavljati različite zadatke u istom proizvodnom ciklusu.
- Uvrštavanje vizualnog učenja
 - Vizualno učenje omogućuje robotima prepoznavanje i interakciju s različitim objektima i uređajima. Kroz korištenje kamera i naprednih algoritama za obradu slika, roboti mogu intuitivno reagirati na promjene u okolini te se prilagoditi novim situacijama.
- Kombinacija umjetne inteligencije i kontrole ponašanja
 - Integracija umjetne inteligencije s kontrolom ponašanja omogućuje robotima učenje iz iskustva i prilagođavanje novim uvjetima. Ova kombinacija omogućuje prilagodljivo ponašanje robota u odgovoru na različite vrste opreme i radnih okolina.





Izazovi u spajanju opreme: unatoč brojnim prednostima, spajanje opreme na industrijske robote susreće se s izazovima poput standardizacije, sigurnosti, kalibracije i kompatibilnosti različitih uređaja. Stoga, inženjeri trebaju pažljivo održavati ravnotežu između inovacija i praktične primjenjivosti kako bi postigli optimalnu integraciju opreme na industrijske robote. Ovaj dinamični proces predstavlja ključni korak u stvaranju prilagodljivih, učinkovitih i inteligentnih robotskih sustava koji mogu odgovoriti na sve izazove modernih proizvodnih okruženja.

Pokretanje jednostavnog automatiziranog sustava



Pokretanje jednostavnog automatiziranog sustava s industrijskim robotom uključuje nekoliko ključnih koraka kako bi se osiguralo sigurno i učinkovito izvođenje zadataka. Evo osnovnog vodiča za pokretanje takvog sustava:

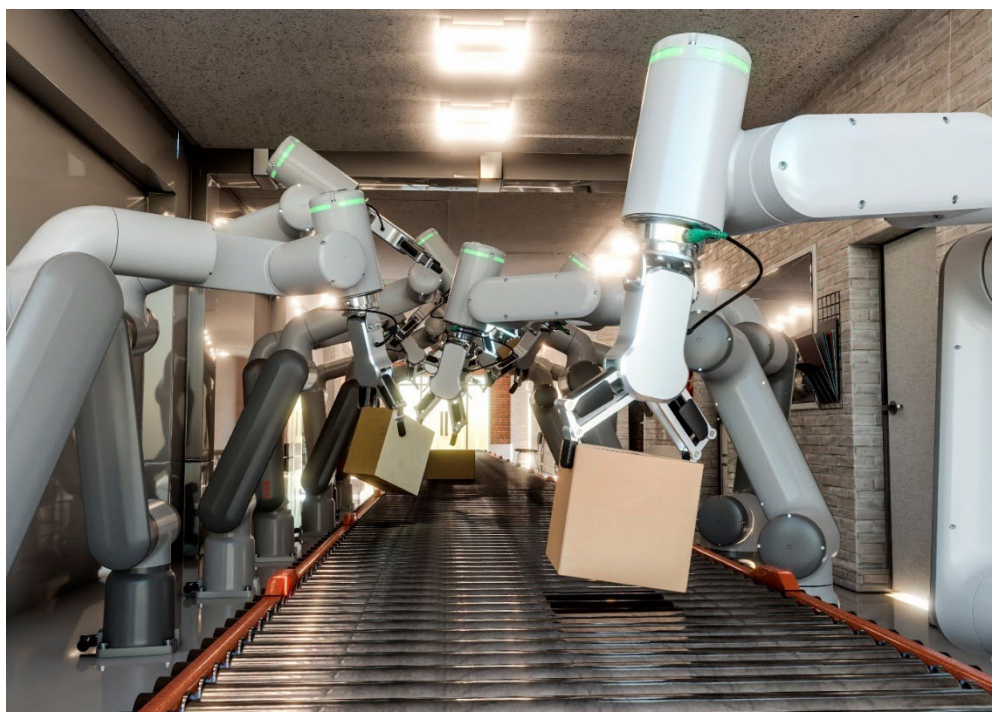
- Priprema radnog okruženja
 - Provjerite je li radno okruženje čisto, uredno i sigurno. Uklonite sve prepreke i osigurajte da nema nepotrebnih osoba u blizini radnog prostora. Ako je potrebno, uvrstite sigurnosne mjere, poput ograđivanja radnog prostora ili korištenja senzora za otkrivanje prisutnosti.
- Kalibracija robota i senzora
 - Prije pokretanja sustava, provjerite je li robot kalibriran kako bi osigurao točnost i preciznost u izvođenju zadataka. Kalibrirajte i provjerite sve senzore koji se koriste u sustavu, poput senzora položaja, kamere ili drugih detektora.
- Postavljanje alata i opreme
 - Ako sustav uključuje dodatne alate, senzore ili druge opreme, postavite ih i pravilno ih povežite s robotom. Ovo uključuje pravilno postavljanje grijača, alata za hvatanje ili drugih uređaja koji će se koristiti tijekom zadatka.



- Programiranje zadatka
 - Kreirajte upravljački program koji definira rad robota u skladu s određenim zadatkom. Ovo uključuje programiranje putanja, definiranje uvjeta i korištenje odgovarajućih naredbi za upravljanje robotom i povezanom opremom.
- Simulacija i testiranje
 - Prije stvarnog izvođenja zadatka, preporučuje se provesti simulacije kako biste testirali program i utvrdili eventualne probleme ili greške. Simulacije pomažu u sigurnom ispitivanju ponašanja robota prije nego što dođe u kontakt s radnim okruženjem.
- Podešavanje parametara
 - Podesite parametre sustava, poput brzine, ubrzanja ili bilo kakvih drugih specifičnih postavki koje se odnose na vašu primjenu. Ovo osigurava da robot radi s optimalnim radnim karakteristikama i u skladu s postavljenim zahtjevima zadatka.
- Praćenje učinkovitosti i sigurnosti
 - Tijekom pokretanja sustava, pažljivo pratite rad robota. Provodite testne cikluse kako biste procijenili učinkovitost izvođenja zadatka. Isto tako, osigurajte da su sve sigurnosne mjere aktivne i da robot radi unutar definiranih parametara sigurnosti.

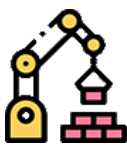


- Optimizacija i pružanje nadzora
 - Nakon pokretanja sustava, analizirajte rezultate, utvrdite mogućnosti za optimizaciju, te pružite stalni nadzor rada sustava. Neprekidno praćenje omogućuje prilagodbu i poboljšanje radnih karakteristika tijekom vremena.



Slika 10. Prikaz jednostavnog automatiziranog sustava

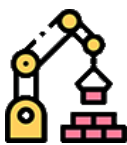
Ove korake treba prilagoditi specifičnostima vaše primjene i uvjetima rada. Važno je imati na umu da je sigurnost uvijek imperativ prilikom pokretanja bilo kojeg automatiziranog sustava, stoga je neophodno pridržavati se sigurnosnih smjernica i protokola.



Pokretanje složenijeg automatiziranog sustava predstavlja izazovnu i dinamičnu fazu u primjeni naprednih robotskih tehnologija. Složeni sustavi obuhvaćaju raznoliku opremu, više robotskih entiteta, precizne senzore i sofisticirane softverske integracije. Ovaj proces zahtijeva holistički pristup i pažljivo upravljanje kako bi se postigla optimalna učinkovitost i sigurnost u radu.

Prije samog pokretanja, ključno je detaljno projektiranje i planiranje sustava. Definiranje specifičnih ciljeva, utvrđivanje ključnih resursa te precizno utvrđivanje karakteristika zadatka postavljaju temelje za uspješnu integraciju. Integracija različitih komponenti, kao što su različite vrste robota, senzori, alati i drugi uređaji, zahtijeva visoku razinu preciznosti i koordinacije kako bi se osiguralo međusobno usklađeno djelovanje.

Ključni korak u pokretanju složenijeg sustava jest usklađivanje radnog okruženja s tehničkim zahtjevima. To uključuje osiguravanje sigurnosnih elemenata, postavljanje jasnih protokola za hitne situacije te prilagodbu radnog prostora u skladu s potrebama sustava. Pored toga, kalibracija svih komponenta, uključujući robote i senzore, od vitalnog je značaja kako bi se postigla točnost i preciznost u izvođenju zadataka.



Složeni automatizirani sustavi često zahtijevaju integraciju naprednih softverskih rješenja. To uključuje programiranje robota, algoritme za koordinaciju među entitetima, kao i upravljanje podacima i komunikaciju između različitih dijelova sustava. Uvođenje umjetne inteligencije, strojnog učenja i algoritama za optimizaciju dodatno unapređuje sposobnosti sustava.

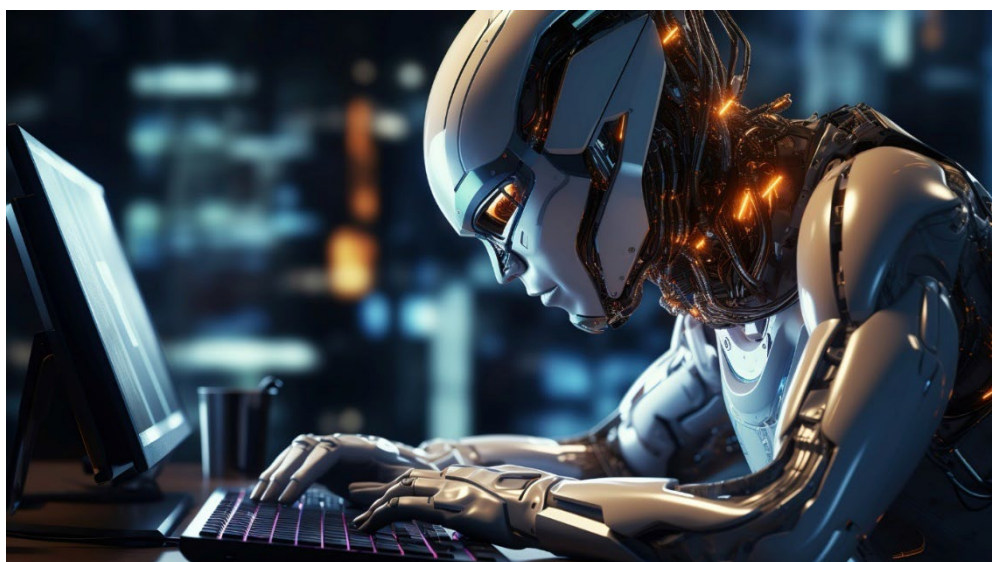
Nakon integracije, ključno je provesti opsežno testiranje kako bi se otkrile moguće greške i optimizirala radna karakteristika. Simulacije igraju ključnu ulogu u ovoj fazi, omogućujući inženjerima procjenu ponašanja sustava u kontroliranom okruženju prije nego što se sustav stavi u stvarnu proizvodnu liniju.

Kroz ovaj holistički pristup, inženjeri su u mogućnosti učinkovito savladati izazove i ostvariti puni potencijal složenog automatiziranog sustava. Sposobnost prilagodbe i suradnje među različitim komponentama, preciznost u izvođenju zadataka te stalna optimizacija ključni su čimbenici uspješnog pokretanja i održavanja složenih robotskih sustava u industrijskim postrojenjima.



Kreiranje upravljačkog programa za rad industrijskog robota

Kreiranje upravljačkog programa za rad industrijskog robota predstavlja najvažniji korak u ostvarivanju automatizacije proizvodnje. Ovaj proces zahtijeva sustavno i precizno pristupanje kako bi se osigurala točna izvedba zadataka. Inženjeri se suočavaju s izazovom razvijanja programa koji će omogućiti robotu izvršavanje različitih operacija unutar zadanih parametara.



Slika 11. Ilustracija robotskog programiranja



Prvo, programeri definiraju specifične zadatke koje robot treba obaviti. To može uključivati pokrete kroz određene putanje, manipulaciju objektima ili čak suradnju s drugim robotima ili strojevima. Definiranje preciznih koraka i ciljeva ključno je za stvaranje učinkovitog programa.

Nadalje, programiranje putanja i koordinacija pokreta igraju ključnu ulogu. Inženjeri koriste posebne programske jezike, poput *RAPID*-a za robote tvrtke *ABB* ili *KUKA* priručnike za programiranje za robote *KUKA*, kako bi definirali gibanje robota u trodimenzionalnom prostoru. Ova faza zahtijeva pažljivo usklađivanje brzina, ubrzanja i koordinacija zglobova kako bi se postiglo željeno ponašanje.

Uvrštavanje senzora u program omogućuje robotu interakciju s okolinom. To uključuje prepoznavanje objekata, praćenje položaja i osiguravanje preciznosti u izvršavanju zadataka. Integracija senzora često zahtijeva dodatnu kalibraciju i podešavanje kako bi osigurala točnost mjerenja.

Sigurnost je još jedan ključan element pri programiranju industrijskih robota. Definiranje sigurnosnih protokola, poput hitnih zaustavljanja u slučaju opasnosti ili postavljanja sigurnosnih zona, osigurava zaštitu radnika i opreme.

Napredniji programi često uključuju upotrebu umjetne inteligencije i strojnog učenja. Ove tehnologije omogućuju robotima učenje iz iskustava, prilagođavanje promjenama u okolini i optimizaciju svojeg ponašanja tijekom vremena.



Primjer programa za robota UR5



Nakon što je program kreiran, inženjeri provode testiranje i simulacije kako bi otkrili potencijalne greške ili optimizirali izvedbu. Ovaj ponavljajući proces omogućuje postizanje visoke razine preciznosti i učinkovitosti u radu industrijskih robota.

U konačnici, kreiranje upravljačkog programa za rad industrijskog robota zahtijeva visoku razinu stručnosti, preciznosti i prilagodljivosti kako bi se odgovorilo na specifičnosti svake proizvodne linije i zadatka.

Zamislite da želimo napisati upravljački program za industrijskog robota *UR5*, koji treba izvršiti jednostavan zadatak - pokupiti objekt s radne površine i postaviti ga na određeno mjesto. Koristit ćemo skriptni jezik *RAPID*, specifičan za robote tvrtke *Universal Robots*. U stvarnosti, programiranje robota ovisi o specifičnostima robota i korištenom programskom jeziku, ali ovaj primjer ilustrirat će osnovni koncept.

MODULE MainModule

```
VAR robtarget StartPosition := [[0, 0, 0, 0, 0, 0], [9E9,
9E9, 9E9, 9E9, 9E9, 9E9]];
VAR tool GripperTool := tool0;
```

```
PROC main()
  MoveToStartPosition();
  PickUpObject();
  MoveToTargetPosition();
  ReleaseObject();
ENDPROC
```

```
PROC MoveToStartPosition()
  MoveJ StartPosition, v100, fine, Tool\Wrist;
ENDPROC
```



```
PROC PickUpObject()
    CloseGripper();
    WaitTime 1.0;
ENDPROC

PROC MoveToTargetPosition()
    VAR robtarget TargetPosition := [[0.5, 0.5, 0.2, 0, 0,
0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
    MoveL TargetPosition, v100, fine, Tool\Wrist;
ENDPROC

PROC ReleaseObject()
    OpenGripper();
    WaitTime 1.0;
ENDPROC

PROC CloseGripper()
    GripPosC(GripperTool, 100);
ENDPROC

PROC OpenGripper()
    GripPosO(GripperTool, 100);
ENDPROC

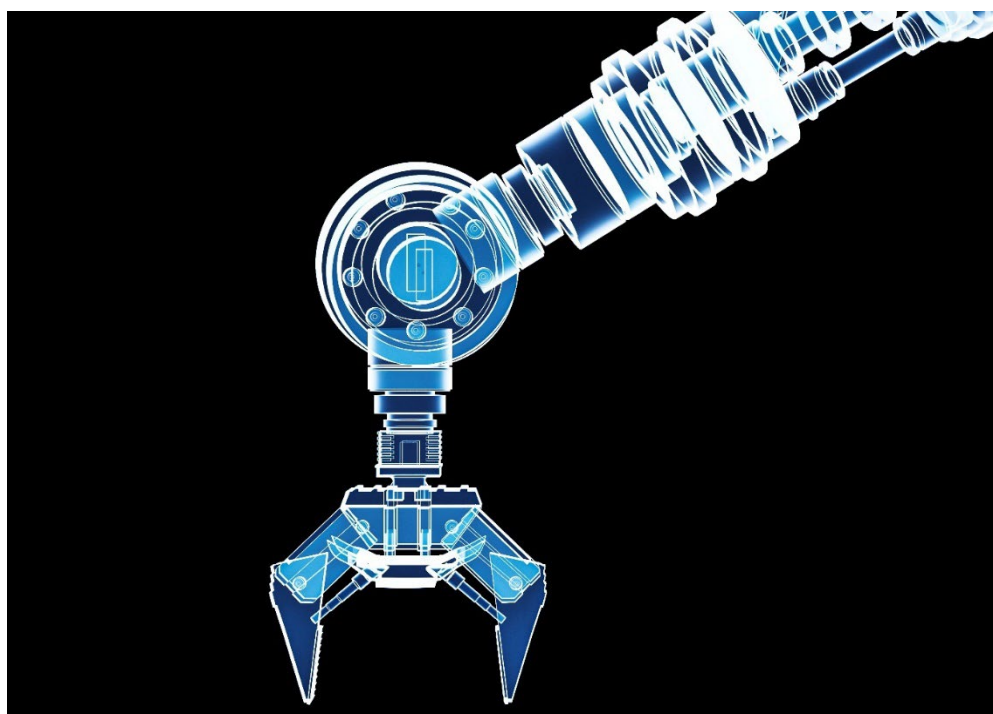
ENDMODULE
```

Ovaj jednostavan program koristi osnovne funkcije kao što su **MoveJ** i **MoveL** za kontrolu gibanja robota, te funkcije **CloseGripper** i **OpenGripper** za upravljanje hvataljkom. Program se sastoji od glavne procedure **main**, koja koordinira pokrete robota tijekom izvođenja zadatka. Naravno, ovo je samo jedan od mnogih mogućih pristupa, a stvarni program ovisio bi o specifičnostima zadatka, tipu robota i alatima koje koristimo.

Pozicije i stupnjevi slobode gibanja



U kontekstu industrijskih robota, pozicija se odnosi na točnu lokaciju i orijentaciju robota u trodimenzionalnom prostoru. Lokacija se definira pomoću koordinata koje određuju položaj robota u odnosu na određeni referentni sustav. Uobičajeno se koriste x , y , z koordinate za određivanje položaja u prostoru, dok se orijentacija robota može definirati kroz *Eulerove kutove* ili kroz kvaternion za preciznije upravljanje orijentacijom.



Slika 12. Prikaz hidrauličkih elemenata potrebnih za kretanje robota

Stupnjevi slobode predstavljaju broj neovisnih načina na koje se robot može kretati. Uobičajeni industrijski roboti često imaju šest stupnjeva slobode, što znači da se mogu kretati u šest različitih smjerova ili rotirati oko šest različitih osi. Ti stupnjevi slobode uključuju:



- **Translacija po x , y i z osi** - omogućuje robotu da se pomakne naprijed-nazad (po x osi), lijevo-desno (po y osi) i gore-dolje (po z osi).
- **Rotacija oko x , y i z osi** - ovi stupnjevi omogućuju rotaciju oko osi x (*roll*), y (*pitch*) i z (*yaw*), pružajući robotu mogućnost okretanja ili rotacije oko tih osi.

Stupnjevi slobode ključni su jer određuju pokretljivost i mogućnosti kretanja robota. Što više stupnjeva slobode robot ima, to je veća njegova sposobnost izvođenja složenijih zadataka i prilagodbe različitim radnim situacijama. Međutim, povećanje stupnjeva slobode može zahtijevati složenije upravljanje i složenije planiranje pokreta.

Translacija je jedan od osnovnih koncepata u geometriji i kinematici koji se odnosi na pomak ili kretanje tijela s jednog mjesta na drugo. U kontekstu industrijskih robota, translacija se odnosi na pomak robota duž određenih osi u trodimenzionalnom prostoru. To znači da se robot može kretati naprijed-nazad (po x osi), lijevo-desno (po y osi) i gore-dolje (po z osi).

Osnovni princip translacije u industrijskim robotima koristi se za postizanje promjene položaja ili lokacije robota kako bi izvršio određeni zadatak. Na primjer, ako robot mora pokupiti predmet s jednog mjesta i premjestiti ga na drugo, koristit će se translacija kako bi se pomakao od početne do ciljne pozicije.

Koordinate x , y i z definiraju smjerove translacije u trodimenzionalnom prostoru. Kada se koriste u kontekstu industrijskih robota, te koordinate određuju točnu lokaciju robota ili krajnju točku njegove putanje.





Translacija je ključna u postizanju preciznosti i točnosti u izvođenju zadataka industrijskih robota. Precizno upravljanje translacijom omogućuje robotima da se pozicioniraju točno na određenoj lokaciji kako bi izvršili operacije poput hvatanja, postavljanja ili obrade predmeta.

Osim translacije u prostoru, moguće je koristiti i translaciju alata (*tool translation*) koja se odnosi na relativno kretanje alata na kraju robota. To omogućuje robotima precizno manipuliranje alatima i objektima u trodimenzionalnom prostoru bez potrebe za promjenom položaja cijelog robota.

Kroz kombinaciju translacije i rotacije (drugi vid stupnjeva slobode), industrijski roboti mogu izvesti raznolike zadatke s visokom preciznošću i pokretljivošću u različitim industrijskim aplikacijama.

Rotacija je kretanje oko osi koje rezultira promjenom orijentacije tijela ili objekta u prostoru. U kontekstu industrijskih robota, rotacija se odnosi na promjenu položaja robota oko osi x , y ili z , što omogućuje promjenu orijentacije alata ili kraja robota.

U industrijskim robotima, rotacija je jedan od ključnih stupnjeva slobode koji omogućuje robotu izvođenje različitih zadataka s preciznošću. Klasično, postoji šest stupnjeva slobode, od kojih su tri rotacijska i tri translacijska. Rotacija se ostvaruje kroz kretanje robota oko triju osi:



- **Roll (rotacija oko x osi)** - *roll* se odnosi na rotaciju oko x osi robota. Kada se robot rotira oko te osi, njegova prednja strana može se okrenuti prema gore ili prema dolje. To je poput rotacije glave s nagibom prema naprijed ili natrag.
- **Pitch (rotacija oko y osi)** - *pitch* predstavlja rotaciju oko y osi. Rotacija oko te osi omogućuje robotu da nagne svoju prednju stranu prema gore ili prema dolje, poput kretanja glave u nagibu lijevo-desno.
- **Yaw (rotacija oko z osi)** - *yaw* je rotacija oko z osi. Rotacija oko te osi omogućuje robotu da se okreće lijevo-desno oko svoje osi, poput okretanja glave ili tijela.

Precizno upravljanje rotacijom omogućuje robotima prilagodbu orijentacije alata ili objekata s kojima rade. Na primjer, kada robot mora postaviti predmet u određeni položaj ili ga obraditi s različitih strana, rotacija omogućuje preciznu kontrolu orijentacije objekta.

Kombinacija rotacije i translacije omogućuje robotima izvršavanje složenih zadataka s visokom preciznošću. To im omogućuje prepoznavanje, hvatanje i rukovanje objektima na različite načine, prilagođavajući svoj položaj i orijentaciju prema specifičnim zahtjevima zadatka.

Evo jednostavnog primjera programa za industrijskog robota koji koristi rotaciju i translaciju kako bi pokupio objekt s jedne lokacije i postavio ga na drugu. Primjer je pisan na pseudokodu koji kombinira koncepte rotacije i translacije.



```
PROC MainProgram()  
    MoveToStartPosition()  
    PickUpObject()  
    RotateObject()  
    MoveToTargetPosition()  
    ReleaseObject()  
ENDPROC
```

```
PROC MoveToStartPosition()  
    // Funkcija za kretanje robota do početne pozicije  
    // Ovdje se koristi translacija  
    MoveRobotTo(X_start, Y_start, Z_start)  
ENDPROC
```

```
PROC PickUpObject()  
    // Funkcija za hvatanje objekta  
    // Ovdje se koristi rotacija za pravilno pozicioniranje  
    hvataljke  
    RotateGripperTo(PickUpAngle)  
    CloseGripper()  
ENDPROC
```

```
PROC RotateObject()  
    // Funkcija za rotaciju objekta ako je potrebno  
    // Ovdje se koristi rotacija oko osi objekta  
    RotateObjectAroundAxis(Object, RotationAxis, Angle)  
ENDPROC
```

```
PROC MoveToTargetPosition()  
    // Funkcija za kretanje robota do ciljne pozicije  
    // Ovdje se koristi translacija  
    MoveRobotTo(X_target, Y_target, Z_target)  
ENDPROC
```

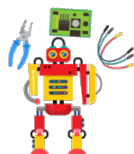
```
PROC ReleaseObject()  
    // Funkcija za otpuštanje objekta  
    // Ovdje se koristi rotacija za pravilno pozicioniranje  
    hvataljke  
    RotateGripperTo(ReleaseAngle)  
    OpenGripper()  
ENDPROC
```



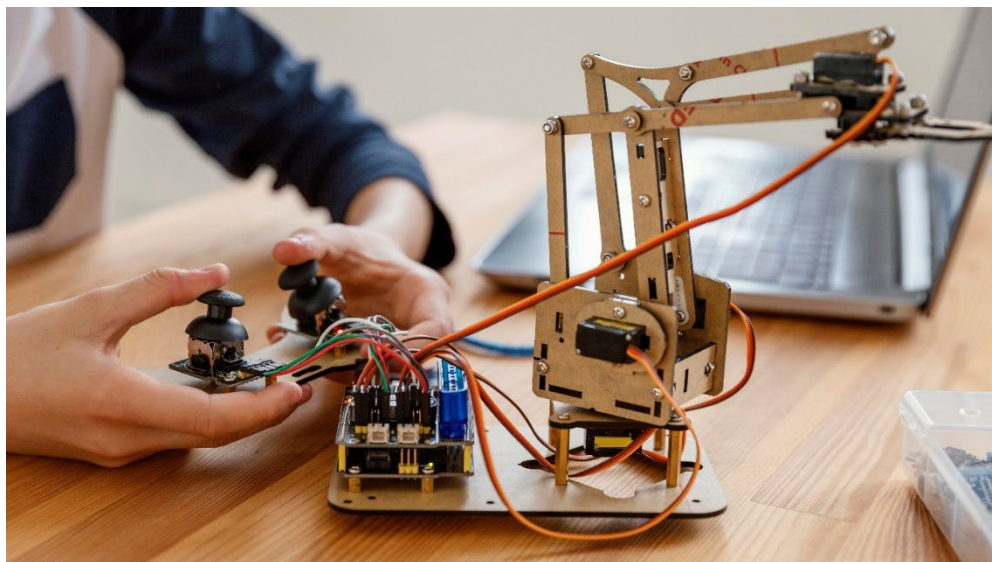
Ovaj primjer prikazuje osnovne korake u programu za rad s robotom. Koristi se funkcija **Move Robot To** za translaciju, koja premješta robota na određene koordinate u prostoru. Također se koriste funkcije za rotaciju **Rotate Gripper To** i **Rotate Object Around Axis** kako bi se postigli određeni kutovi rotacije za hvatanje i otpuštanje objekta.

Napomena: ovo je konceptualni pseudokod koji ilustrira ideju upotrebe rotacije i translacije u programiranju industrijskih robota. U stvarnom programiranju robota, koristile bi se specifične funkcije programskog jezika koji se koristi za određeni model robota i njegove sposobnosti.

Simulacija i izrada industrijskog robota



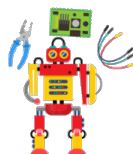
Simulacija industrijskog robota odnosi se na stvaranje virtualnog modela robota i njegovo programiranje kako bi se testirali različiti scenariji i zadatci prije nego se robot stvarno uvrsti u stvarnoj proizvodnji. To je ključan korak u razvoju industrijskih robota jer omogućuje inženjerima provjeru i optimizaciju radnih karakteristika robota bez stvarnog fizičkog postavljanja.



Slika 13. Izrada jednostavnog robota

Proces izrade simulacije industrijskog robota obično uključuje nekoliko ključnih koraka:

- **Modeliranje robota** - inženjeri koriste *CAD (Computer-Aided Design)* softver za stvaranje 3D modela robota koji odražava njegovu stvarnu geometriju, dimenzije i mehaniku. Ovaj model obično uključuje sve dijelove robota, poput zglobova, poveznica, alata i senzora.



- **Programiranje u simulacijskom okruženju** - nakon stvaranja modela, inženjeri koriste simulacijski softver kako bi programirali kretanje robota, definirali operacije, postavili putanje kretanja i provjerili kako će robot reagirati na određene naredbe i scenarije.
- **Testiranje i analiza radnih karakteristika** - simulacija omogućuje testiranje različitih scenarija kako bi se provjerile radne karakteristike robota. To uključuje provjeru preciznosti kretanja, brzine izvođenja zadataka, sigurnosnih elemenata i interakciju s okolinom. Analiza rezultata simulacija pomaže inženjerima utvrđivanje i rješavanje mogućih problema prije fizičkog uvrštavanja.
- **Optimizacija programa** - na temelju rezultata simulacija, inženjeri mogu optimizirati kontrolne programe i algoritme kako bi poboljšali radne karakteristike robota. To uključuje prilagodbu putanja kretanja, optimizaciju brzine i poboljšanje učinkovitosti.

Simulacija industrijskog robota omogućuje inženjerima stvaranje sigurnog, učinkovitog i optimiziranog rješenja prije nego što se robot stvarno integrira u proizvodnju. To štedi vrijeme, novac i minimizira rizik od potencijalnih problema koji se mogu pojaviti u stvarnom okruženju.



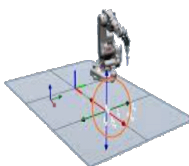
Simuliranje rada industrijskog robota

Simuliranje rada industrijskog robota omogućuje virtualno testiranje njegovih pokreta, funkcionalnosti i interakcije s okolinom prije stvarne integracije u proizvodnom okruženju. Ova tehnika pruža detaljan uvid u radne karakteristike robota, dopuštajući inženjerima:

- **Prognoziranje ponašanja robota** - simulacije omogućuju predviđanje kako će se robot ponašati u različitim situacijama, otkrivajući potencijalne izazove ili probleme prije fizičke primjene.
- **Testiranje programiranja i putanje kretanja** - omogućuje provjeru točnosti i preciznosti putanja kretanja, brzine izvođenja zadataka te pravilnosti izvedbe operacija.
- **Analiziranje sigurnosti** - provjeravaju moguće sigurnosne rizike i utvrđuju potencijalne opasnosti za radnike ili okolinu.
- **Optimiziranje radne karakteristike** - omogućuje inženjerima optimizaciju programa i algoritama za postizanje boljih radnih karakteristika i učinkovitosti robota.

Korištenje simulacija pruža visoku razinu kontrole i analize, što pomaže u smanjenju troškova, vremena i rizika prije stvarne primjene robota u proizvodnom okruženju.

Prognoziranje ponašanja robota u simulacijama temelji se na stvaranju virtualnog modela robota koji odražava njegove fizičke karakteristike, kinematiku, senzore, alate i okolinu u kojoj će raditi. To omogućuje inženjerima predviđanje ponašanja robota u različitim scenarijima i uvjetima prije njegove stvarne integracije.

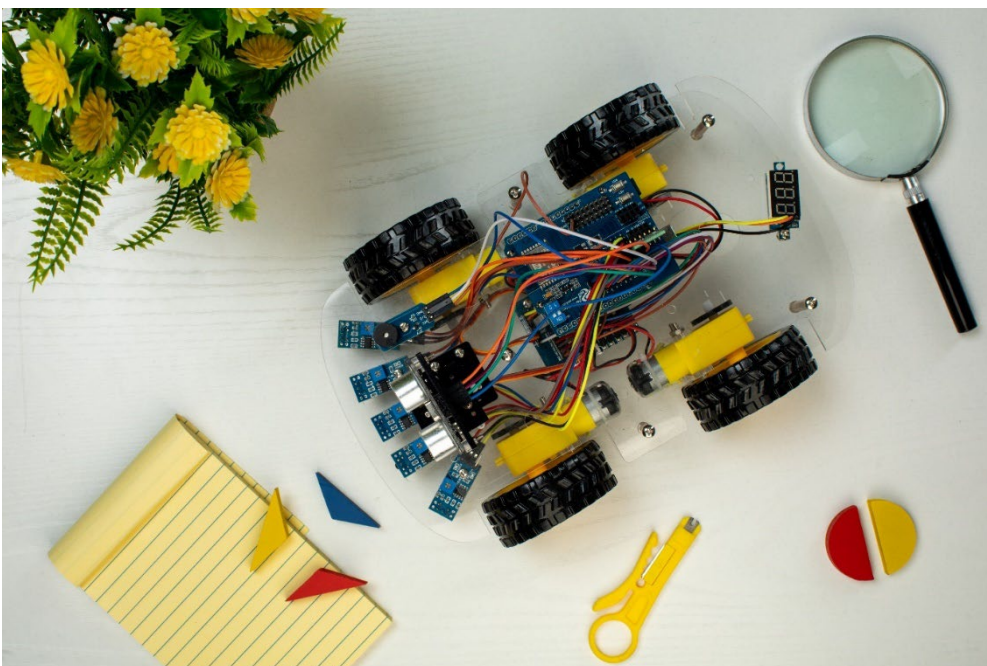


Ključni koraci u prognoziranju ponašanja robota kroz simulacije uključuju:

- **Stvaranje detaljnog virtualnog modela robota** - razvoj 3D modela koji precizno opisuje geometriju, kinematiku i mehaničke karakteristike robota. To uključuje sve dijelove robota, njihove dimenzije, veze, zglobove, alate i senzore.
- **Definiranje parametara okoline** - određivanje virtualnog okruženja u kojem će robot raditi. To može uključivati radne stanice, prepreke, objekte, transportne trake ili bilo koje druge elemente prisutne u stvarnom proizvodnom okruženju.
- **Programiranje ponašanja robota** - korištenje simulacijskog softvera za programiranje kretanja, operacija i interakcija robota s okolinom. To uključuje definiranje putanja kretanja, operacija hvatanja, manipulacije objektima ili drugih zadataka koje robot treba izvesti.
- **Simuliranje različitih scenarija** - izvođenje simulacija s različitim scenarijima i uvjetima kako bi se testirala izvedba robota. To uključuje različite putanje kretanja, različite objekte kojima robot treba manipulirati ili različite brzine izvođenja zadataka.
- **Analiza rezultata** - praćenje ponašanja robota tijekom simulacija i analiza rezultata kako bi se utvrdili potencijalni problemi, greške u programiranju ili sigurnosni rizici. Ovo omogućuje inženjerima da optimiziraju program i algoritme za bolje radne karakteristike.

Kroz ove korake, prognoziranje ponašanja robota omogućuje inženjerima stvaranje predviđanja o tome kako će se robot ponašati u različitim situacijama te da otkriju i riješe potencijalne probleme prije stvarne primjene u proizvodnom okruženju.

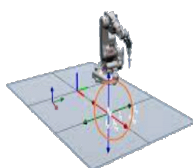




Slika 14. Simulacija rada jednostavno izrađenog robota

Testiranje programa i putanje robota ključno je kako bi se osigurala točnost, sigurnost i učinkovitost rada robota prije nego što se uvrsti u stvarnu proizvodnju. Taj proces uključuje nekoliko koraka:

- **Simulacija kretanja** - korištenje simulacijskog softvera za testiranje programa kretanja robota. Provjera putanja kretanja omogućuje utvrđivanje potencijalnih kolizija s okolinom, drugim objektima ili samim robotom, te ispravljanje takvih problema prije stvarne primjene.
- **Verifikacija preciznosti** - provjera točnosti i preciznosti putanja kretanja robota u simuliranom okruženju. Ovo uključuje provjeru jesu li zadane putanje izvedive, stiže li robot do ciljanih pozicija unutar zadanih tolerancija i koliko se puta približava željenim točkama.



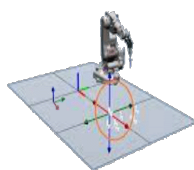
- **Simulacija operacija** - testiranje izvođenja različitih operacija, poput hvatanja i manipulacije objektima. Ovo uključuje provjeru funkcionalnosti *gripper*a, točnost pozicioniranja i manipuliranja objektima.
- **Analiza radnih karakteristika** - praćenje brzine izvođenja, učinkovitosti i sigurnosti rada robota tijekom simulacije. Analiza radnih karakteristika pomaže u utvrđivanju mogućih poboljšanja ili optimizacija programa.
- **Testiranje u stvarnom okruženju** - nakon uspješnog testiranja u simuliranom okruženju, moguće je izvesti testiranje u stvarnom okruženju pod pažljivim nadzorom. Ovo testiranje u stvarnom okruženju pomaže potvrditi da robot radi kako se očekuje i da su sve sigurnosne mjere zadovoljene.

Testiranje programa i putanja robota omogućuje korisnicima da utvrde i riješe eventualne greške, poboljšaju preciznost i učinkovitost rada robota te omoguće sigurnost pri njegovoj primjeni u stvarnom proizvodnom okruženju.

Analiziranje sigurnosti industrijskih robota ključno je kako bi se osigurao rad robota na način koji minimizira rizike za ljude, druge robote, opremu i okolinu. Ovo analiziranje uključuje različite elemente:

- **Rizici kolizije** - provjera mogućnosti da robot dođe u kontakt s drugim objektima, ljudima ili drugim robotima. To uključuje analizu radnog prostora robota, brzinu kretanja, putanje kretanja i postavke sigurnosnih zona.
- **Sigurnosni senzori** - provjera ispravnosti i učinkovitosti sigurnosnih senzora poput senzora prisutnosti, senzora dodira ili kamera za otkrivanje ljudi ili prepreka u radnom prostoru robota. Analiza pokrivenosti senzora važna je kako bi se osiguralo da robot prepoznaje i reagira na opasnosti u svom okruženju.





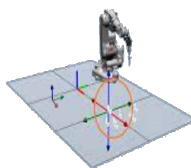
- **Sustavi za zaustavljanje u nuždi** - testiranje funkcionalnosti i brzine sustava za zaustavljanje u nuždi kako bi se robot mogao odmah zaustaviti u slučaju opasnosti ili neočekivanog događaja. Ovo uključuje testiranje mehanizama za zaustavljanje pokreta robota ili za isključivanje energije.
- **Protokoli sigurnosti** - provjera pridržavanja sigurnosnih standarda i protokola. To uključuje provjeru je li programiranje i izvođenje robota u skladu s propisima i smjernicama sigurnosti industrijskih robota.
- **Osiguranje rada s ljudima** - analiza razine sigurnosti pri radu s ljudima, kao što su sigurnosne prepreke, ograničavanje brzine u prisutnosti ljudi ili sigurnosne procedure za suradnju ljudi i robota.
- **Simulacije sigurnosti** - korištenje simulacija kako bi se testirali scenariji opasnosti ili neočekivanih situacija kako bi se vidjelo kako robot reagira i kako bi se otkrile potencijalne slabosti sigurnosnih sustava.

Analiziranje sigurnosti industrijskih robota ključno je kako bi se osigurao siguran način rada robota, minimizirajući rizik od ozljeda ili štete pri radu u proizvodnom okruženju.

Optimizacija radnih karakteristika industrijskih robota ključna je za postizanje najboljih rezultata učinkovitosti, preciznosti i produktivnosti tijekom njihove primjene. Ova optimizacija uključuje nekoliko ključnih elemenata:

- **Brzina i učinkovitost** - poboljšanje brzine izvođenja zadataka i učinkovitosti u radu robota. To može uključivati optimizaciju putanja kretanja, smanjenje nepotrebnih pokreta ili brže izvođenje operacija.





- **Preciznost** - povećanje preciznosti u izvođenju zadataka. To se postiže finim podešavanjem algoritama kretanja, kalibriranjem senzora ili optimizacijom mehanizama hvatanja objekata.
- **Optimizacija vremena ciklusa** - smanjenje vremena potrebnog za izvođenje ciklusa rada robota. To uključuje analizu svake faze rada i otkrivanje načina kako ubrzati procese, smanjiti čekanje ili optimizirati tranzicije između operacija.
- **Sigurnost i pouzdanost** - poboljšanje sigurnosti i pouzdanosti rada robota. To uključuje optimizaciju sigurnosnih protokola, provjeru i nadograđivanje sigurnosnih sustava te osiguranje stabilnosti rada robota tijekom cijelog ciklusa rada.
- **Optimizacija energije** - smanjenje potrošnje energije robota. To uključuje otkrivanje načina za smanjenje potrošnje energije motora, senzora ili drugih komponenata kako bi se povećala energetska učinkovitost.
- **Analiza podataka** - korištenje podataka iz stvarnog rada robota ili simulacija za analizu radnih karakteristika i otkrivanje područja za poboljšanje. To omogućuje neprekidno praćenje i optimizaciju rada robota.

Optimizacija radnih karakteristika industrijskih robota ključna je za postizanje maksimalne učinkovitosti, preciznosti i pouzdanosti rada, čime se osigurava bolja produktivnost i smanjenje troškova u proizvodnom okruženju.



Izrada jednostavne robotske ruke i programa za simulaciju



Izrada jednostavne robotske ruke zahtijeva kombinaciju mehaničkog dizajna, elektronike, programiranja i testiranja kako bi se postigla funkcionalnost i radne karakteristike potrebne za izvršavanje zadataka koji se od nje očekuju. Izrada jednostavne robotske ruke uključuje stvaranje mehaničkog sustava koji oponaša strukturu ljudske ruke kako bi izvodio različite zadatke poput hvatanja, podizanja ili manipulacije objektima. Evo koraka koji su obično uključeni u taj proces:

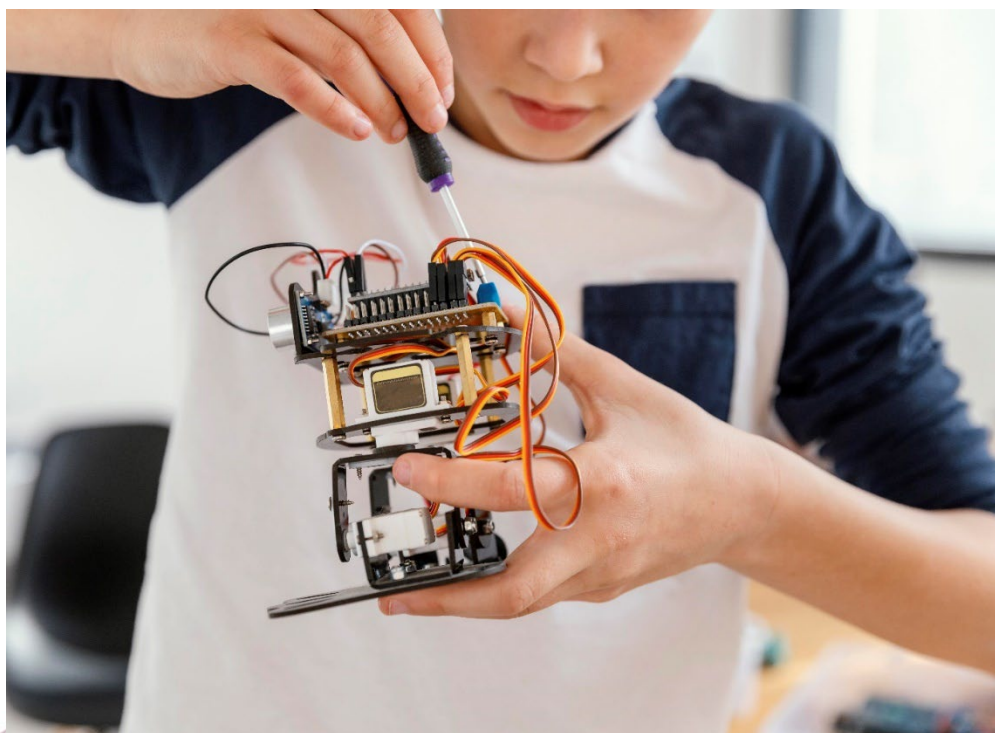
- **Planiranje dizajna** - definiranje osnovnih karakteristika robotske ruke, uključujući broj zglobova, dužine segmenata, vrstu aktuatora (poput motora ili hidrauličke) i senzora potrebnih za otkrivanje položaja i sile.
- **Modeliranje i dizajn** - korištenje softverskih alata za dizajn (kao što su CAD programi) za stvaranje 3D modela robotske ruke. To uključuje dizajniranje segmenata, zglobova, spojeva i strukture ruke.
- **Izbor materijala** - odabir materijala za izradu segmenata robotske ruke koji će biti dovoljno čvrsti, ali i lagani za izvođenje pokreta. Može se koristiti plastika, metal ili čak 3D ispis.
- **Aktuatori i senzori** - integracija aktuatora (motora, pneumatskih ili hidrauličnih sustava) za pokretanje zglobova te senzora za praćenje položaja i otkrivanje sile kako bi se omogućila precizna kontrola pokreta i sigurnost rada.
- **Kontrolni sustav** - razvoj kontrolnog sustava koji će primati ulazne signale od senzora i prevođenje tih signala u pokrete robotske ruke. To može biti jednostavni mikrokontroler ili napredniji sustav kontrole.
- **Testiranje i kalibracija** - testiranje svakog dijela ruke, provjera pokretljivosti, preciznosti i sigurnosti. Kalibracija senzora i aktuatora za optimalne radne karakteristike.



Izrada jednostavne robotske ruke preko Arduino platforme

- **Programiranje pokreta** - razvoj programa koji upravlja robotskom rukom i omogućava izvođenje različitih zadataka, poput hvatanja, podizanja ili manipulacije predmetima.
- **Integracija sa softverom** - ako je potrebno, integracija robotske ruke s odgovarajućim softverskim okruženjem za kontrolu i upravljanje.

Izrada jednostavne robotske ruke zahtijeva kombinaciju mehaničkog dizajna, elektronike, programiranja i testiranja kako bi se postigla funkcionalnost i radne karakteristike potrebne za izvršavanje zadataka koji se od nje očekuju.



Slika 15. Izrada jednostavnog robota preko platforme Arduino



Izrada jednostavne robotske ruke preko *Arduino* platforme može biti izvrstan početni projekt. Evo primjera kako biste mogli krenuti s ovakvim projektom:

Materijali:

- *Arduino* mikrokontroler (npr. *Arduino Uno*)
- servomotori (tri-četiri, ovisno o broju zglobova)
- čvrsti materijal za izradu segmenata (3D ispis, drvo, plastika)
- vodiči za povezivanje
- *Breadboard* (za privremeno povezivanje elektroničkih komponenti)
- potrebni alati za izradu mehaničkih dijelova (3D pisač, alat za rezanje).

Koraci:

- **Dizajn i izrada mehaničkih dijelova**
 - Koristite softver poput *Fusion 360*, *Tinkercad* ili sličnog za dizajn segmenata robotske ruke. Možete kreirati segmente koji simuliraju zglobove ruke.
 - Ispisujte ili izrađujte dijelove ruke odabranim materijalom. To će biti dijelovi koji će se pokretati pomoću servomotora.
- **Priprema *Arduino* platforme**
 - Povežite servomotore s *Arduino* mikrokontrolerom prema shemi spajanja. Svaki servomotor ima tri žice (napajanje, uzemljenje, kontrolni signal) koje se povezuju s odgovarajućim pinovima *Arduino* ploče.
 - Programirajte *Arduino* koristeći *Arduino IDE* ili sličan softver. Ovdje ćete definirati kako će se servomotori kretati kako bi simulirali pokrete robotske ruke.



- **Testiranje i kalibracija**
 - Pomoću programa na *Arduino* platformi testirajte pokrete robotske ruke. Kalibrirajte pokrete i kutove zglobova tako da odgovaraju potrebama ruke.
- **Spajanje s mehaničkim dijelovima**
 - Pričvrstite servomotore na mehaničke dijelove koje ste izradili.
 - Povežite segmente ruke sa servomotorima. Pazite da su zglobovi pokretljivi i omogućuju slobodno kretanje.
- **Programiranje pokreta ruke**
 - U programu na *Arduino* platformi definirajte pokrete robotske ruke. Možete koristiti kutove zglobova kako biste kontrolirali pokrete i položaje ruke.
- **Testiranje i optimizacija**
 - Testirajte robotsku ruku u akciji. Provjerite funkcionalnost, preciznost i sigurnost pokreta. Poboljšajte program i mehaničke dijelove prema potrebi.

Ovaj primjer daje osnovnu ideju kako izraditi jednostavnu robotsku ruku koristeći *Arduino* platformu. Bit će potrebno puno eksperimentiranja, prilagodbi i testiranja kako biste postigli željene rezultate.



Evo jednostavnog primjera koda za kontrolu jednostavne robotske ruke s tri servomotora. U ovom primjeru, servoi će simulirati pokrete baze, ramena i lakta robotske ruke.

```
#include <Servo.h>

Servo baseServo; // Servo za bazu
Servo shoulderServo; // Servo za rame
Servo elbowServo; // Servo za lakat

int basePin = 9; // Pinovi servoa za bazu, rame i lakat
int shoulderPin = 10;
int elbowPin = 11;

void setup() {
  baseServo.attach(basePin); // Spajanje pinova servoa
  shoulderServo.attach(shoulderPin);
  elbowServo.attach(elbowPin);
}

void loop() {
  moveServos(); // Poziv funkcije za pokrete servoa
}

void moveServos() {
  // Funkcija za simuliranje pokreta robotske ruke

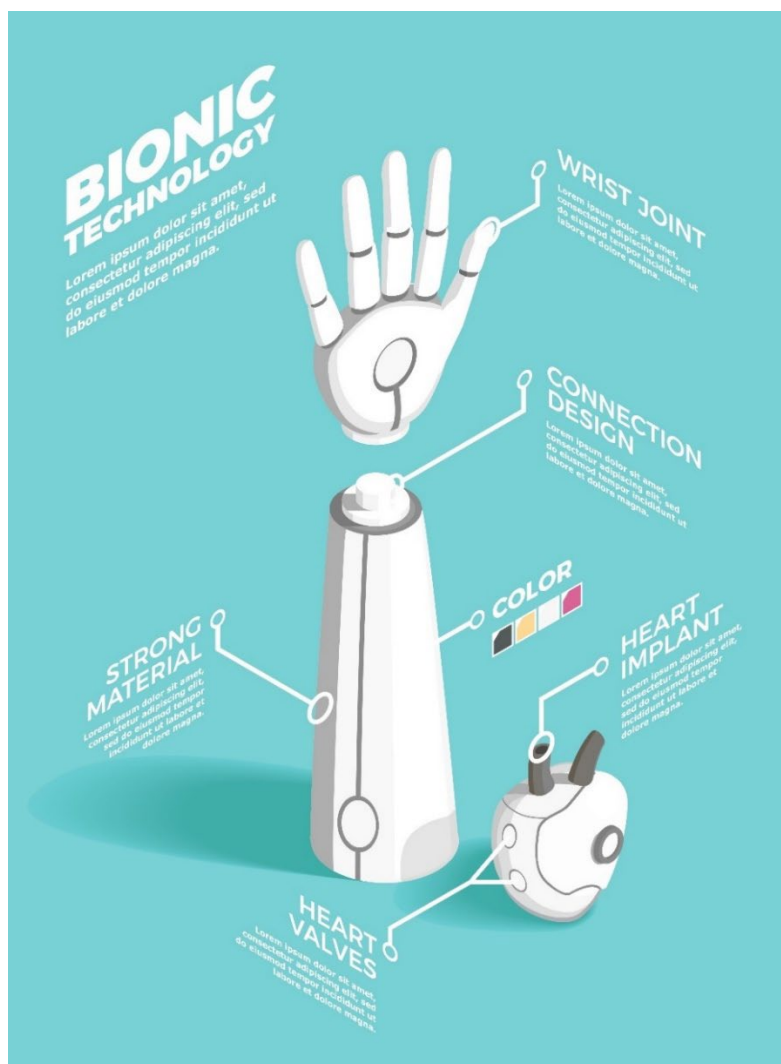
  baseServo.write(90); // Postavljanje pozicije za bazu na 90 stupnjeva
  delay(1000); // Pauza od 1 sekunde

  shoulderServo.write(90); // Postavljanje pozicije za rame na 90 stupnjeva
  delay(1000); // Pauza od 1 sekunde

  elbowServo.write(90); // Postavljanje pozicije za lakat na 90 stupnjeva
  delay(1000); // Pauza od 1 sekunde
```



Ovaj primjer koda pokrenut će svaki servomotor na 90 stupnjeva, zadržati tu poziciju sekundu, a zatim se vratiti u početni položaj. Pazite da pravilno spojite servomotore na odgovarajuće pinove na *Arduino* ploči i prilagodite pinove u kodu prema vašem spoju. Također, ovo je samo osnovni primjer, a stvaranje složenijih pokreta za kontrolu cijele robotske ruke zahtijevalo bi dodatnu logiku i programiranje.



Slika 16. Dizajn jednostavne robotske ruke



Razvit ćemo malo složeniji primjer koda koji će omogućiti kontrolu svakog servoa pojedinačno i simulirati pokrete robotske ruke. U ovom primjeru, koristit ćemo potencioetre kako bismo kontrolirali svaki servomotor. Potencioetri će se koristiti kao ulazni senzori koji će mijenjati položaj svakog od servoa ovisno o položaju potencioetra.

Kako bismo to postigli, prvo spojimo tri potencioetra i tri servomotora na *Arduino*. Središnji pin potencioetra spaja se na analogni pin *Arduina*, a ostali pinovi idu na *5V* za napajanje i *GND* za uzemljenje. Svaki servomotor spojen je na svoj digitalni pin.

Evo kako bi izgledao primjer koda za ovu konfiguraciju:

```
#include <Servo.h>
Servo baseServo;           // Servo za bazu
Servo shoulderServo;       // Servo za rame
Servo elbowServo;          // Servo za lakat
int basePotPin = A0;        // Analogni pinovi potencioetra
                             za bazu, rame i lakat
int shoulderPotPin = A1;
int elbowPotPin = A2;
void setup() {
    baseServo.attach(9);    // Spajanje pinova serva
    shoulderServo.attach(10);
    elbowServo.attach(11);
}
void loop() {
    int baseAngle = map(analogRead(basePotPin), 0, 1023, 0,
180); // Čitanje vrijednosti potencioetra i
mapiranje na kutove serva
    int shoulderAngle = map(analogRead(shoulderPotPin), 0,
1023, 0, 180);
    int elbowAngle = map(analogRead(elbowPotPin), 0, 1023,
0, 180);
    baseServo.write(baseAngle); // Postavljanje
kutova servoa na vrijednosti iz potencioetra
    shoulderServo.write(shoulderAngle);
    elbowServo.write(elbowAngle);
    delay(15); // Mali delay kako bi se spriječila prebrza
promjena kutova
}
```



Taj kod čita vrijednosti sa svakog potencijometra i mapira ih na kutove od 0 do 180 stupnjeva, koji se zatim šalju na odgovarajuće servomotore. Ovaj primjer omogućava kontrolu svakog servoa pojedinačno pomoću potencijometara.

Ovo je osnovni primjer i može se nadograditi dodavanjem dodatnih kontrola, optimizacijom koda ili prilagodbom kuta i brzine kretanja servoa.

Izrada jednostavne robotske ruke preko PLC-a



Programmable Logic Controller (PLC) industrijski je kontroler koji se često koristi za automatizaciju i upravljanje u industrijskim postrojenjima. Izrada jednostavne robotske ruke pomoću *PLC*-a uključuje upotrebu *PLC*-a za kontrolu pokreta i operacija robotske ruke. Evo koraka koji bi bili uključeni u izradu jednostavne robotske ruke preko *PLC*-a:

- **Dizajn mehaničke strukture** - kreiranje mehaničke strukture robotske ruke koja će oponašati pokrete ljudske ruke. To može uključivati korištenje aktuatora kao što su pneumatski cilindri, linearni aktuatori ili servomotori za pokretanje zglobova.
- **Odabir odgovarajućih senzora i aktuatora** - odlučivanje o tipu senzora (npr. senzori položaja, senzori sile) i aktuatora (npr. električni motori, pneumatski cilindri) koji će biti uvršteni u mehaničku strukturu.
- **Povezivanje senzora i aktuatora s PLC-om** - spajanje senzora i aktuatora na *PLC* putem odgovarajućih sučelja (ulazno/izlaznih modula, analognih ulaza/izlaza).
- **Programiranje PLC-a** - programiranje *PLC*-a koristeći odgovarajući jezik za *PLC* (npr. *ladder logic*, *structured text*) kako bi se definirali pokreti i operacije robotske ruke. To uključuje definiranje logike za kontrolu pokreta servoa ili aktuatora na temelju ulaznih signala senzora.
- **Testiranje i kalibracija** - testiranje funkcionalnosti ruke, provjera preciznosti pokreta, kalibracija senzora i aktuatora kako bi se osiguralo da ruka radi u skladu s očekivanjima.
- **Integracija s radnim okruženjem** - integracija robotske ruke u odgovarajuće radno okruženje ili postrojenje te testiranje rada robota u stvarnom okruženju.



Za izradu jednostavne robotske ruke pomoću *PLC*-a, osnovno razumijevanje rada *PLC*-a, senzora, aktuatora i programiranja *PLC*-a je ključno. To je široko područje koje zahtijeva detaljnu konfiguraciju i programiranje prema specifičnim zahtjevima projekta.

Izrada robotske ruke preko *PLC*-a zahtijeva detaljnu konfiguraciju i programiranje prema specifičnim zahtjevima projekta. Evo jednostavnog primjera kako bi se kontrolirala jednostavna robotska ruka pomoću *PLC*-a *Siemens S7-1200* koristeći *Ladder Logic*, jedan od jezika za programiranje *PLC*-a:

Primjer konfiguracije i programiranja:

- Konfiguracija *PLC*-a
 - U *Siemens TIA Portal* softveru, konfigurirajte *PLC* dodavanjem potrebnih digitalnih i analognih ulazno/izlaznih modula.
- Dizajn mehaničke strukture
 - Izradite mehaničku strukturu robotske ruke koja uključuje zglobove i aktuatora (npr. pneumatske cilindre ili servomotore).
- Povezivanje aktuatora i senzora s *PLC*-om
 - Povežite svaki aktuator (npr. servomotor za zglob, pneumatski cilindar za hvataljku) s odgovarajućim izlazima *PLC*-a.
 - Povežite senzore (npr. senzore položaja, senzore sile) s odgovarajućim ulazima *PLC*-a.
- Programiranje *PLC*-a
 - Ovdje je primjer *Ladder Logic* koda za upravljanje jednostavnom robotskom rukom s jednim servomotorom koji kontrolira zglob:



NETWORK 1:

```
// Ulazni signal za aktivaciju motora
I0.0 // Pritisak gumba za aktivaciju
```

```
// Aktuator (servo motor) za kontrolu
zgloba
```

```
Q0.0 // Signal za kontrolu servo motora
```

```
// Logika za kontrolu motora
```

```
M0.0 := I0.0; // Signal za aktivaciju
motora prema pritisku gumba
```

```
// Kontrola rotacije zgloba (primjer
kutova)
```

```
// Ovdje koristite metodu PID ili
definirajte različite kutove ovisno o
senzorima i poziciji
```

```
// Ovisno o senzorima, dodajte logiku za
ograničavanje pokreta ruke kako bi se
izbjegle kolizije
```

```
Q0.0 := M0.0; // Signal za kontrolu motora
prema aktivaciji
```

- Testiranje i kalibracija
 - Testirajte programiranje i funkcionalnost robotske ruke u stvarnom okruženju.
 - Kalibrirajte senzore i aktuatore kako bi se osigurala točnost i preciznost pokreta.

Ovo je osnovni primjer *Ladder Logic* koda koji simulira kontrolu jednostavnog servomotora koji predstavlja zglobov robotske ruke. Za složenije robotske ruke, bit će potrebno dodati više logike, više senzora i aktuatora te više programa koji upravljaju različitim elementima robotske ruke. Sve ovisi o specifičnim potrebama i dizajnu vašega sustava.

Evo detaljnijeg primjera kako biste kontrolirali robotsku ruku pomoću *PLC*-a *Siemens S7-1200* koristeći *Ladder Logic* za upravljanje trima servomotorima koji predstavljaju različite zglobove ruke. Ovaj primjer temelji se na pretpostavci da su servomotori povezani s odgovarajućim izlaznim modulima *PLC*-a, a senzori (npr. senzori položaja) povezani su s ulaznim modulima.



Konfiguracija ulaznih i izlaznih modula *PLC*-a

- Izlazi:
 - Izlaz 0 (**Q0.0**) - servomotor za bazu ruke.
 - Izlaz 1 (**Q0.1**) - servomotor za rame ruke.
 - Izlaz 2 (**Q0.2**) - servomotor za lakat ruke.

Ladder Logic kod za upravljanje robotskom rukom:

NETWORK 1:

```
// Ulazni signal za pokretanje rada robotske ruke
I0.0 // Pritisak gumba za pokretanje rada
```

```
// Aktuatori za kontrolu servomotora za bazu, rame i lakat
```

```
Q0.0 // Signal za kontrolu servomotora za bazu
Q0.1 // Signal za kontrolu servomotora za rame
Q0.2 // Signal za kontrolu servomotora za lakat
```

```
// Logika za pokretanje rada robotske ruke
M0.0 := I0.0; // Signal za pokretanje rada prema pritisku gumba
```

NETWORK 2:

```
// Ulazni signali za senzore položaja za svaki servomotor
```

```
I1.0 // Senzor položaja za bazu
I1.1 // Senzor položaja za rame
I1.2 // Senzor položaja za lakat
```

```
// Logika za kontrolu pokreta svakog servomotora
// Ovdje se koriste informacije senzora položaja kako bi se postigla željena pozicija svakog zglobova
```

```
Q0.0 := M0.0 AND NOT I1.0; // Signal za kontrolu baze ruke prema senzoru položaja
```

```
Q0.1 := M0.0 AND NOT I1.1; // Signal za kontrolu rame ruke prema senzoru položaja
```

```
Q0.2 := M0.0 AND NOT I1.2; // Signal za kontrolu lakta ruke prema senzoru položaja
```

Ovaj primjer koristi logiku za kontrolu pokreta svakog servomotora, uz pretpostavku da senzori položaja (koji su povezani s ulazima I1.0, I1.1 i I1.2) daju informacije o trenutnom položaju svakog zgloba ruke.



M0.0 označava signal za pokretanje rada robotske ruke koji se aktivira pritiskom gumba. Svaki izlazni signal (**Q0.0**, **Q0.1**, **Q0.2**) kontrolira odgovarajući servomotor na osnovu informacija o položaju koje daju senzori (**I1.0**, **I1.1**, **I1.2**).

Ovo je osnovni primjer upravljanja robotskom rukom pomoću *PLC*-a. Za stvarni sustav, ovisno o dizajnu i složenosti robotske ruke, bit će potrebno dodati više logike, uvrstiti sigurnosne mehanizme i dodatne senzore te optimizirati pokrete za postizanje točnosti i stabilnosti.

Primjer složenijeg sustava za upravljanje robotskom rukom:

- Konfiguracija *I/O modula*
 - Povežite izlazne module s aktuatorima (servomotori, pneumatski cilindri) koji kontroliraju svaki zglob ruke.
 - Povežite ulazne module sa senzorima (senzori položaja, senzori sile) koji prate stanje i okolinu robota.
- Definiranje specifikacija rada robota
 - Razvijte detaljan opis funkcionalnosti robota. To uključuje koje će zadatke robot izvršavati, kako će se kretati, sigurnosne zahtjeve itd.
- Programiranje *PLC*-a
 - Primjer *Ladder Logic* koda za upravljanje robotskom rukom s nekoliko servomotora i senzora:

```
NETWORK 1: // Pokretanje i zaustavljanje
rada robota
    I0.0 // Start signal za rad robota
    M0.0 // Signal za zaustavljanje rada
    robota
```

```
// Logika za pokretanje i zaustavljanje
rada robota
    M0.1 := I0.0 AND NOT M0.0; // Signal za
pokretanje rada robota
    M0.2 := M0.0; // Signal za zaustavljanje
rada robota
```



```
NETWORK 2: // Kontrola servo motora za
različite zglobove ruke
// Ulazni signali za senzore položaja i
senzore sile
I1.0 // Senzor položaja za bazu
I1.1 // Senzor položaja za rame
I1.2 // Senzor položaja za lakat
I1.3 // Senzor sile (npr. ako ruka hvata
objekt)
```

```
// Logika za kontrolu servo motora za
svaki zglob
Q0.0 := M0.1 AND NOT I1.0; // Signal za
kontrolu servomotora za bazu
```

```
Q0.1 := M0.1 AND NOT I1.1; // Signal za
kontrolu servomotora za rame
Q0.2 := M0.1 AND NOT I1.2; // Signal za
kontrolu servomotora za lakat
```

```
// Dodatna logika za sigurnost ili
kompleksnije pokrete
// Na primjer, zaustavljanje pokreta ako
se detektira prevelika sila na senzoru sile
```

```
NETWORK 3: // Sigurnosne funkcije
// Ovdje se implementiraju sigurnosne
funkcije, kao što su hitni prekidači,
// zaustavljanje u slučaju detekcije
prepreka, provjera granica pokreta, itd.
```

- Testiranje i kalibracija
 - Testirajte programiranje i funkcionalnost robotske ruke u stvarnom okruženju.
 - Kalibrirajte senzore i aktuatore kako bi se osigurala točnost i preciznost pokreta.

Stvarni sustav može biti puno složeniji i zahtijevati više funkcionalnosti, kao što su složeni algoritmi kretanja, obrada slike, upravljanje višestrukim alatima, usklađivanje pokreta, upravljanje s više robota itd. Uvrštavanje puno složenijeg sustava ovisi o konkretnim zahtjevima i dizajnu robota.



Pokretanja programa za izvršavanje specifičnih zadataka



Slika 17. Nadzor robotiziranog sustava



Pokretanje programa industrijskog robota

Pokretanje programa za izvršavanje specifičnih zadataka na industrijskom robotu obično uključuje nekoliko koraka kako bi se postigla željena operacija. Evo općenitog postupka:

Priprema sustava

- **Konfiguracija radnog okruženja** - provjerite je li radno okruženje pripremljeno za izvršavanje programa. To uključuje osiguravanje da nema prepreka, pravilno postavljanje alata ako se koristi, te sigurnosne mjere.
- **Pokretanje sustava** - otvorite upravljački softver za robotiku, kao što je *teach pendant* ili softver za programiranje robota.

Učitavanje i izvršavanje programa

- **Učitavanje programa** - učitajte specifičan program ili zadatke koje želite izvršiti. To može biti programiranje u softveru (npr. kroz grafičko sučelje) ili učitavanje programa iz memorije.
- **Simulacija (opcionalno)** - ukoliko je moguće, prvo simulirajte program kako biste provjerili njegovu ispravnost prije stvarnog pokretanja na robotu.
- **Pokretanje programa** - odaberite mogućnost za pokretanje programa ili zadatka na upravljačkom sučelju ili softveru. To može biti gumb za pokretanje ili specifična naredba za izvršenje.

Praćenje i evaluacija

- **Praćenje rada** - pratite izvršavanje programa kako biste osigurali da robot izvršava zadatke prema planu.
- **Vrednovanje rezultata** - provjerite jesu li zadatci izvršeni ispravno i jesu li rezultati u skladu s očekivanjima.



Postupci nakon izvršenja

- **Pohrana podataka (ukoliko je potrebno)** - ako postoje rezultati ili podatci koji se moraju pohraniti, provjerite i pohranite ih prema potrebi.
- **Sigurnosne mjere** - nakon završetka operacije, slijedite sigurnosne protokole za isključivanje ili zaustavljanje rada robota.
- **Održavanje i čišćenje** - učinite potrebno održavanje robota i radnog okruženja, počistite opremu ili alate ako je potrebno.

Postoji varijacija u postupcima ovisno o vrsti industrijskog robota, njegovom softveru, specifičnostima zadatka i upravljačkom sučelju koje se koristi. Važno je pažljivo slijediti upute proizvođača i pridržavati se sigurnosnih smjernica tijekom izvršavanja bilo kakvih operacija s industrijskim robotima.

Primjer:

- Priprema radnog okruženja
 - Provjerite je li radna zona robota slobodna od prepreka i sigurna za rad.
 - Ako je potrebno, postavite alate ili uređaje koji će se koristiti tijekom izvršavanja zadatka.
- Pokretanje upravljačkog sučelja
 - Pokrenite upravljačko sučelje robota, bilo da se radi o *teach pendantu* ili softverskom sučelju na računalu.
- Učitavanje programa
 - Učitajte specifičan program ili zadatak koji želite izvršiti. Ovo se može učiniti putem sučelja softvera za programiranje robota.



- Simulacija (ako je dostupna)
 - Ako je moguće, prvo simulirajte program kako biste provjerili ispravnost i očekivane pokrete robota.
- Provjera sigurnosti
 - Provjerite sigurnosne funkcije robota kako bi se osiguralo da su aktivirani sigurnosni senzori i protokoli prije pokretanja programa.
- Pokretanje programa
 - Odaberite opciju za pokretanje programa na upravljačkom sučelju ili softveru.
 - To može biti gumb za pokretanje programa ili specifična naredba za izvršenje zadatka.
- Praćenje rada robota
 - Pratite pokrete i akcije robota tijekom izvršavanja programa putem sučelja ili vizualne kontrole.
- Vrednovanje rezultata
 - Nakon završetka programa, provjerite jesu li zadatci izvršeni kako je planirano. Provjerite rezultate i usporedite ih s očekivanjima.
- Održavanje i čišćenje
 - Nakon završetka operacije, provjerite robota, očistite alate ili opremu i obavite bilo koje potrebno održavanje.
- Isključivanje robota
 - Kada je program završen i rad robota više nije potreban, slijedite postupak za isključivanje ili zaustavljanje rada robota.

Ovo je općenit primjer procesa pokretanja programa na industrijskom robotu. Važno je pridržavati se sigurnosnih uputa i postupaka proizvođača kako biste osigurali siguran i učinkovit rad robota.

Rješavanje problema i optimizacija rada robota



Kada se suočite s problemom u radu industrijskog robota, proces rješavanja problema započinje otkrivanjem i analizom uzroka problema. Detaljniji opis rješavanja problema i optimizacije rada robota može se razdvojiti u nekoliko koraka:

Otkrivanje problema

- **Opažanje problema** - osoblje koje radi s robotom primijeti odstupanje od očekivanog ponašanja ili radnog procesa.
- **Priprema za dijagnostiku** - zabilježite sve važne informacije o problemu, uključujući vrijeme kada se pojavio, ponašanje robota, radno okruženje i bilo koje dodatne uvjete.
- **Dijagnostika problema** - koristite dijagnostičke alate ili softver kako biste utvrdili uzrok problema. To može uključivati pregled logova sustava, provjeru senzora ili analizu rada pokretačkog softvera.

Koraci u rješavanju problema

- **Isključivanje i ponovno pokretanje** - u nekim slučajevima, ponovno pokretanje sustava može privremeno riješiti probleme uzrokovane manjim greškama ili preopterećenjima.
- **Analiza programa** - ako je problem povezan s programom, pregledajte kod kako biste pronašli moguće greške i napravili potrebne ispravke.
- **Kalibracija i održavanje** - provjerite i kalibrirajte senzore, aktuatora i druge dijelove robota. Održavanje može pomoći u sprječavanju problema.



Optimizacija rada

- **Analiza radnih karakteristika** - proučite učinkovitost i radne karakteristike robota kako biste otkrili područja za optimizaciju, poput brzine, preciznosti ili učinkovitosti.
- **Prilagodba parametara** - prilagodite parametre rada robota, poput brzine, ubrzanja ili sila koje se primjenjuju, kako biste poboljšali rezultate.
- **Poboljšanje programa** - proučite programski kod robota kako biste otkrili prilike za optimizaciju, smanjenje vremena izvršavanja ili poboljšanje preciznosti.
- **Analiza procesa rada** - proučite proces rada robota kako biste otkrili mogućnosti za optimizaciju koji bi mogli rezultirati povećanjem ukupne produktivnosti.

Ovaj pristup zahtijeva sistematsko praćenje radnih karakteristika, redovito održavanje, analizu procesa te prilagodbu i poboljšanje programa i parametara kako bi se osigurala što bolja funkcionalnost i učinkovitost industrijskog robota.



LITERATURA

- [1] Clark, E; Roberts, L. 2023. *Pokretanje automatiziranog sustava s industrijskim robotom: Ključni koraci i izazovi*. Journal of Automation Technology, 10(4), 211-227.
- [2] Lee, S; Kim, J. 2023. *Simulation Techniques for Industrial Robots*. International Journal of Robotics Applications, 15(2), 78-95.
- [3] Roberts, A; Wilson, B. 2023. *Kreiranje upravljačkog programa za rad industrijskog robota: Procesi, izazovi i primjer programiranja*. Robotics and Automation Review, 18(2), 87-102.
- [4] Rodriguez, M; Garcia, A. 2023. *Predictive Behavior Analysis of Industrial Robots through Simulations*. Journal of Manufacturing Technology, 9(4), 211-230.
- [5] Sanchez, M; Chen, L. 2023. *Optimization Techniques for Industrial Robot Performance*. Robotics and Automation Review, 11(3), 145-16
- [6] Smith, J; Johnson, R. 2023. *Integracija opreme na industrijske robote: Inovativni pristupi, prednosti i izazovi*. Journal of Industrial Robotics, 25(2), 78-91.
- [7] *Upravljački uređaji za industrijske robote: Ključne komponente i procesi*. Industrijska Robotika, 15(3), 123-135.