# Rukovanje predmetima rada u pokretu pomoću vizijskih sustava

**Širinić, Davor**

**Master's thesis / Diplomski rad**

**2019**

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

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:235:163767

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-04-19**

*Repository / Repozitorij:*

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

**Davor Širinić**

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentori:

Prof. dr. sc. Bojan Jerbić, dipl. ing.
Dr. sc. Bojan Šekoranja, mag. ing.

Student:

Davor Širinić

Zagreb, 2019.

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

Zahvaljujem se obitelji na podršci tokom studija, svim profesorima od kojih sam imao zadovoljstvo učiti, posebno svojim mentorima Bojanu Jerbiću i Bojanu Šekoranji, te zaposlenicima laboratorija za izradu izradbenih i montažnih sustava.

Davor Širinić

SVEUČILIŠTE U ZAGREBU
**FAKULTET STROJARSTVA I BRODOGRADNJE**
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala te mehatronika i robotika

| Sveučilište u Zagrebu<br>Fakultet strojarstva i brodogradnje | |
|---|---|
| Datum: | Prilog: |
| Klasa: | |
| Ur. broj: | |

# DIPLOMSKI ZADATAK

Student: **DAVOR ŠIRINIĆ** Mat. br.: 0035199031

Naslov rada na hrvatskom jeziku: **Rukovanje predmetima rada u pokretu u pomoću vizijskih sustava**

Naslov rada na engleskom jeziku: **Manipulating of work parts in motion using vision systems**

Opis zadatka:

Suvremeni fleksibilni montažni sustavi zahtijevaju minimalni utrošak vremena za pripremu dijelova. Stoga se nastoje uvoditi različita rješenja za rukovanje predmetima rada bez njihovog zaustavljanja i pozicioniranja, odnosno izvođenje operacija na predmetima u pokretu.

U okviru zadatka potrebno je implementirati sustave vizualnog praćenja i robotskog rukovanja predmeta u pokretu na transportnim trakama. U sustavu jedna traka služi za dobavu dijelova, dok se na drugoj traci nalaze nosači predmeta rada. Potrebno je izuzeti dijelove s prve trake i odložiti ih u nosače. Za traku s dijelovima potrebno je oblikovati vlastiti vizijski sustav i korištenjem komunikacijskih protokola povezati ga s robotom. Na drugoj traci potrebno je primijeniti vizijski sustav robota. Obje trake povezane su robotom pomoću davača impulsa ("inkrementalni enkoder"), a pogonjene elektromotorom s mogućnosti podešavanja brzine trake. Sustav je potrebno izvesti na postojećoj robotskoj stanici te transportnom sustavu u Laboratoriju za projektiranje izradbenih i montažnih sustava.

U radu je potrebno navesti literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
02. svibnja 2019.

Rok predaje rada:
04. srpnja 2019.

Predviđeni datum obrane:
10. srpnja 2019.
11. srpnja 2019.
12. srpnja 2019.

Zadatak zadao:

prof. dr. sc. Bojan Jerbić

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

# SADRŽAJ

1. UVOD ..... 1
2. PREGLED KORIŠTENIH KOMPONENTI ..... 4
3. PROJEKTIRANJE I REALIZACIJA STROJNOG VIDA ZA IZUZIMANJE PREDMETA RADA SA TRANSPORTNE TRAKE ..... 9
4. RAZVOJ PROGRAMA ROBOTSKOG IZUZIMANJA I ODLAGANJA PREDMETA RADA U POKRETU POMOĆU STROJNOG VIDA ..... 21
5. PREGLED RADA RAZVIJENOG SUSTAVA ..... 30
6. MJERENJE I ANALIZA TOČNOSTI IZUZIMANJA I ODLAGANJA PREDMETA RADA, TE KOMPENZACIJA DOBIVENE GREŠKE ..... 33
7. ZAKLJUČAK ..... 58

# POPIS SLIKA


Slika 1. Ford montažna linija, 1913. [1] ..... 1
Slika 2. Potpuno automatizirana montažna linija [2]. ..... 1
Slika 3. Shema robotskog vizijskog sustava. ..... 2
Slika 4. Pokretne trake. ..... 4
Slika 5. Bonfiglioli elektromotor (lijevo) sa nazivnom pločicom (desno). ..... 5
Slika 6. Allen-Bradley PowerFlex 4 upravljačka jedinica. ..... 5
Slika 7. Inkrementalni enkoderi ..... 6
Slika 8. Sony industrijska kamera (lijevo), Basler industrijska kamera (desno). ..... 6
Slika 9. a. FANUC M-3iA 6A industrijski robot, b. FANUC SYSTEM R-30iA Mate kontroler, c. FANUC multiplekser sa napajanjem. ..... 8
Slika 10. Vakuum sisaljka na FESTO vakuum generatoru. ..... 8
Slika 11. TCP/IP informacije na privjesku za učenje. ..... 10
Slika 12. KAREL kod za uspostavljanje socket messaging komunikacije. ..... 10
Slika 13. TCP Open Connection blok sa ulazima (lijevo) i izlazom (desno). ..... 11
Slika 14. Flat Sqeuence Structure okvir (lijevo) i Case Structure okvir (desno). ..... 11
Slika 15. Čitanje okidača i pokretanje niza funkcija za akviziciju i obradu slike. ..... 12
Slika 16. Izbornik NI MAX programa, na dnu korištena *Basler* kamera naziva „cam1“. ..... 12
Slika 17. Niz funkcijskih blokova za akviziciju slike. ..... 13
Slika 18. Učitavanje i učenje uzorka (lijevo) predmeta rada (desno). ..... 13
Slika 19. „IMAQ Match Pattern 4 VI“ funkcijski blok. ..... 14
Slika 20. Parametri traženja predmeta. ..... 14
Slika 21. Funkcijski blok „Index Array“ za izuzimanje elemenata iz polja. ..... 14
Slika 22. Mreža točaka za kalibraciju kamere. ..... 15
Slika 23. Filteri za pripremu slike kalibracije kamere. ..... 15
Slika 24. Funkcijski blokovi za kalibraciju kamere. ..... 16
Slika 25. Definiranje koordinatnog sustava pomoću traženja učitanog uzorka. ..... 16
Slika 26. Transformacija koordinata točke u ravnini [8]. ..... 17
Slika 27. Transformacija koordinata nađenog predmeta. ..... 17
Slika 28. Tok signala od trenutka prekida optičke barijere do dobivanja povratne informacije o predmetu rada. ..... 18
Slika 29. Shema transportnog sustava: a. smjer gibanja trake, b. linija optičke barijere, c. vidno polje kamere ..... 19
Slika 30. Nađeni predmeti rada sa linijom optičke barijere, ujedno i y os koordinatnih sustava (crveno) i x osi koordinatnih sustava, tj. smjerom kretanja (žuto). ..... 19
Slika 31. Parsiranje i slanje koordinata nađenih predmeta. ..... 20
Slika 32. Glavni program „PR_TRACKING“. ..... 21
Slika 33. *KAREL* program za komunikaciju i parsiranje primljenih informacija. ..... 22
Slika 34. Kontrolni program. ..... 23
Slika 35. Program za provjeru rastućeg brida signala (lijevo) i sigurnosni program za zaustavljanje transportne trake (desno). ..... 23
Slika 36. Program za upisivanje primljenih koordinata u registre. ..... 24
Slika 37. Program za izuzimanje predmeta rada. ..... 24
Slika 38. iRVision sučelje. ..... 25
Slika 39. Konfiguracija kamere za *iRVision* strojni vid. ..... 25

Slika 40. Konfiguracija radnog prostora. 26
Slika 41. Kalibracija kamere. 26
Slika 42. Predmet rada sa naučenim značajkama za prepoznavanje (zeleno) i maskom koja naglašava najbitnije značajke (plavo). 27
Slika 43. Sučelje za konfiguraciju vizijskog procesa. 27
Slika 44. Program za odlaganje predmeta rada. 28
Slika 45. Inicijalizacija radnog prostora i strojnog vida. 28
Slika 46. Program za akviziciju slike. 29
Slika 47. Blok dijagram rada sustava. 30
Slika 48. Sučelje *LabView* programa za strojni vid sa nađenim predmetima rada. 31
Slika 49. Izuzimanje predmeta rada sa transportne trake. 31
Slika 50. Sučelje *iRVision* strojnog vida sa nađenim podlogama za odlaganje predmeta rada. 32
Slika 51. Odlaganje predmeta rada na transportnu traku. 32
Slika 52. Predmeti rada. 33
Slika 53. Pomicanje predmeta rada po transportnoj traci uz zadržavanje orijentacije. 34
Slika 54. Predmet rada sa kartezijskim koordinatnim sustavom (lijevo) i polarnim (desno). 34
Slika 55. Grafički prikaz greške izuzimanja predmeta prilikom Y pomaka. 35
Slika 56. Usporedba aproksimacija greške y pomaka: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno). 36
Slika 57. Kompenzacija greške pozicioniranja nastale uslijed y pomaka predmeta rada. 38
Slika 58. Grafički prikaz kontrole kompenzacije greške uslijed y pomaka predmeta rada. 38
Slika 59. Orijentiranje predmeta rada na transportnoj traci uz zadržavanje referentne pozicije. 39
Slika 60. Grafički prikaz greške prilikom orijentiranja predmeta. 40
Slika 61. Usporedba aproksimacija greške orijentacije: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno). 40
Slika 62. Grafički prikaz kontrole kompenzacije greške pozicioniranja uslijed orijentacije predmeta rada. 42
Slika 63. Grafički prikaz kontrolnog mjerenja ponovljene kompenzacije greške pozicioniranja uslijed orijentacije predmeta rada. 43
Slika 64. Kompenzacija greške pozicioniranja uslijed orijentacije predmeta rada. 43
Slika 65. Program za očitavanje okretaja u minuti elektromotora transportne trake. 44
Slika 66. Grafički prikaz greške pozicioniranja uslijed brzine transportne trake. 45
Slika 67. Aproksimacija y komponente greške jednadžbom pravca (crveno) i x komponente greške polinomom drugog reda (plavo). 45
Slika 68. Grafički prikaz kontrole kompenzacije greške pozicioniranja uslijed brzine transportne trake. 46
Slika 69. Kompenzacija greške pozicioniranja uslijed brzine transportne trake. 47
Slika 70. Podloga za odlaganje predmeta rada sa ucrtanim kartezijevim koordinatnim sustavom. 47
Slika 71. Osiguravanje orijentacije prilikom mjerenja y pomaka podloge za odlaganje predmeta rada. 48
Slika 72. Grafički prikaz greške odlaganja predmeta uslijed y pomaka na transportnoj traci. 49

Slika 73. Usporedba aproksimacija greške y pomaka: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno). ..... 49
Slika 74. Grafički prikaz kontrole kompenzacije greške odlaganja predmeta rada uslijed y pomaka na transportnoj traci. ..... 50
Slika 75. Osiguravanje pozicije podloge za odlaganje predmeta rada. ..... 51
Slika 76. Grafički prikaz greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada. ..... 52
Slika 77. Usporedba aproksimacija greške orijentacije: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno). ..... 52
Slika 78. Grafički prikaz kontrole kompenzacije greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada. ..... 53
Slika 79. Program za kompenzaciju greške pri odlaganju predmeta rada na transportnu traku. ..... 54

# POPIS TABLICA

Tablica 1. Mjerenje greške izuzimanja predmeta prilikom Y pomaka. ..... 35
Tablica 2. Kontrolno mjerenje kompenzacije greške uslijed y pomaka predmeta rada. ..... 37
Tablica 3. Mjerenje greške izuzimanja predmeta uslijed orijentacije predmeta. ..... 39
Tablica 4. Kontrolno mjerenje kompenzacije greške uslijed orijentacije predmeta rada. ..... 41
Tablica 5. Kontrolno mjerenje nakon ponovljene kompenzacije greške pozicioniranja uslijed orijentacije predmeta rada. ..... 42
Tablica 6. Mjerenje greške izuzimanja uslijed brzine transportne trake. ..... 44
Tablica 7. Kontrola kompenzacije greške pozicioniranja uslijed brzine transportne trake. ..... 46
Tablica 8. Mjerenje greške odlaganja predmeta uslijed y pomaka predmeta na transportnoj traci. ..... 48
Tablica 9. Kontrola kompenzacije greške odlaganja predmeta rada uslijed y pomaka na transportnoj traci. ..... 50
Tablica 10. Mjerenje greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada. ..... 51
Tablica 11. Kontrola kompenzacije greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada. ..... 53
Tablica 12. Mjerenje greške odlaganja predmeta uslijed brzine transportne trake. ..... 54
Tablica 13. Prvi set mjerenja greške izuzimanja i odlaganja predmeta rada bez uključene kompenzacije. ..... 56
Tablica 14. Drugi set mjerenja greške izuzimanja i odlaganja predmeta rada sa uključenom kompenzacijom. ..... 57

# POPIS OZNAKA

| Oznaka | Jedinica | Opis |
|---|---|---|
| $\alpha$ | ° | Kut zakreta predmeta rada |
| $\varphi$ | ° | Kut zakreta koordinatnog sustava |
| A | | Matrica transformacije koordinatnog sustava u ravnini |
| cnt | | Trenutačna vrijednost enkodera |
| cnt2 | | Vrijednost enkodera u trenutku izračuna brzine transportne trake |
| $ey_x$ | mm | X komponenta greške uslijed y pomaka predmeta rada |
| $ey_y$ | mm | Y komponenta greške uslijed y pomaka predmeta rada |
| $e\alpha_x$ | mm | X komponenta greške uslijed orijentacije predmeta rada |
| $e\alpha_y$ | mm | Y komponenta greške uslijed orijentacije predmeta rada |
| $en_x$ | mm | X komponenta greške uslijed brzine transportne trake |
| $en_y$ | mm | Y komponenta greške uslijed brzine transportne trake |
| n | $min^{-1}$ | Brzina transportne trake izražena u okretajima u minuti |
| r | mm | Polumjer pogonskog bubnja transportne trake |
| scl | $mm^{-1}$ | Skala enkodera |
| v | m/s | Obodna brzina transportne trake |
| x | mm | Pomak predmeta rada u smjeru osi x |
| y | mm | Pomak predmeta rada u smjeru osi y |

# SAŽETAK

Tema ovog rada je projektiranje i sinkrono upravljanje robotskim sustavom s dvije transportne trake. Cilj je pokazati eksperimentom kako različiti kontrolni parametri poput brzine transportnih traka, orijentacije i pozicije predmeta rada utječu na točnost i preciznost izuzimanja, odnosno odlaganja predmeta rada, te sukladno sa dobivenim rezultatima, argumentirati moguće područje primjene takvog transportnog sustava u industriji.

U sljedećem tekstu detaljno će se opisati komponente transportnog sustava, obrazložiti izbor istih, razraditi projektiranje vlastitog vizijskog sustava, programiranje i kalibriranje robota, te na kraju opisati postupak provođenja eksperimenta, analiza rezultata i pokušaj kompenzacije dobivene greške pozicioniranja robota.

Ključne riječi: transportni sustav, vizijski sustav, heksapodni industrijski robot

## SUMMARY

Topic of this project is designing and constructing a transport system made of two belt conveyors, two vision systems and a hexapod industrial robot. The goal of the project is to experimentally elaborate how different control parameters, such as belt conveyor speed, orientation and position of the work piece, affect the accuracy and precision of picking and placing the work piece on the moving conveyors, and with the results of that experiment, argument possible applications of such transporation systems in the industry.

The following text will describe used transport system components in detail, as well as designing the vision system, programming and calibrating of the robot and lastly give the detailed analysis of the experiment results in attempt to compensate the possible positioning error of the robot.

Key words: transport system, vision system, hexapod industrial robot

# 1. UVOD

Transportni sustavi su ključni element u skoro svim granama industrije. Neovisno o postrojenju uvijek postoji potreba da se materijal i komponente transportiraju između radnih mjesta, uz što veću uštedu vremena i energije. Prvu montažnu liniju Henry Ford-a su karakterizirale upravo transportne trake, najčešće korišteni uređaji za transport djelova i materijala [1] [Slika 1]. Uzduž linije su ručna radna mjesta gdje je svaki radnik obavljao specifični zadatak sklapanja djelova. Danas u automobilskoj industriji, uslijed visokog stupnja automatizacije potreba za ljudskom radnom snagom svedena je na minimum [Slika 2]. Automatizirani su svi postupci, od sklapanja, obrade, inspekcije dijelova do lakiranja, testiranja i transportiranja konačnog proizvoda. Unatoč visokom stupnju automatizacije postoji prostor za napredak i istraživanje, s obzirom da se većina automatiziranih postupaka događa u sređenoj, ili bar polu-sređenoj okolini.

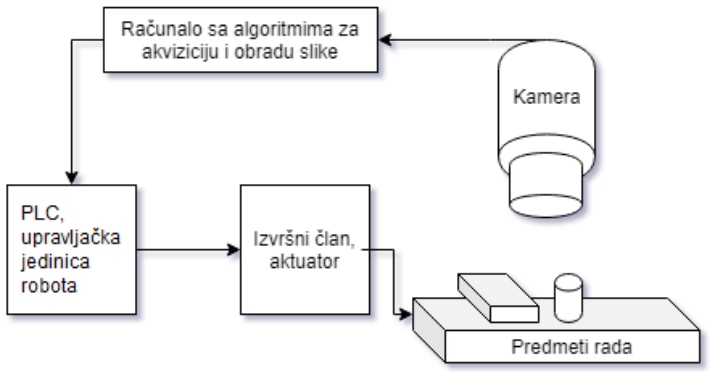
**Slika 1. Ford montažna linija, 1913. [1]**

**Slika 2. Potpuno automatizirana montažna linija [2].**

Idući korak u novom dobu industrije, tkz. Četvrtoj industrijskoj revoluciji, je doseći dovoljan stupanj umjetne inteligencije tehničkih sustava kako bi funkcionirali u determinističkom kaosu, baš poput ljudi. Prepoznavanje  boja i oblika u nesređenoj okolini, stavljanje istih u kontekst, izuzimanje dijelova iz nesređenog skupa, ljudima su jednostavni zadaci koje obavljaju refleksivno, skoro bez razmišljanja. No, za računalo, to su izuzetno složene zadaće koje zahtijevaju uzimanje u obzir mnogo faktora prije realizacije. Na transportnoj traci montažne linije, radnik s lakoćom prepoznaje razbacane dijelove različitih orijentacija, te zahvaljujući svome živčanom sustavu skoro trenutačno može prepoznati te sukladiti svoju motoriku kako bi uhvatio predmet rada. Robotski sustav koji bi obavljao istu radnju sastoji se od nekoliko vrlo složenih podsustava. Ako se premoste ključni problemi koji se javljaju kod usklađivanja takvih podsustava, rezultat je inteligentni uređaj koji zamjenjuje monotoni i ponavljajući ljudski rad sa mnogo većom brzinom, preciznošću, ponovljivosti i kapacitetom rada.

Osnovne komponente sustava za izuzimanje djelova sa pokretne trake su [Slika 3]:

- vizijski sustav,
- računalo za obradu vizualnih informacija,
- robot kao izvršni član,
- upravljačka jedinica robota.




**Slika 3. Shema robotskog vizijskog sustava.**

Kako bi u laboratorijskim uvjetima simulirali opisane uvjete korištene su dvije pokretne trake, dva vizijska sustava i industrijski robot paralelne kinematike. Osnovni zadatak robota je izuzimanje predmeta sa jedne transportne trake i odlaganje istih na drugu uz određeni stupanj točnosti i ponovljivosti.

Prvi problem je moguća greška pozicioniranja robota pri različitim brzinama transportnih traka kod izuzimanja i odlaganja predmeta.

Drugi problem javlja se pri projektiranju vizijskog sustava - obavljanje izuzimanja pokretnog predmeta u stvarnom vremenu. Kašnjenje o informaciji pozicije predmeta duž pokretne trake prisutno je zbog vremena potrebnog da se obavi akvizicija slike, obrada slike, te informacije o poziciji i orijentaciji pošalju nazad na upravljačku jedinicu robota.

Treći problem javlja se uslijed velike greške pozicioniranja robota, koji ima grešku vlastite kinematike zbog pretrpljenog oštećenja. Takav robot bi bio neupotrebljiv u industrijskoj primjeni, no u ovom radu njegov defekt je iskorišten za ispitivanje mogućnosti kompenzacije greške pozicioniranja pomoću mjerenih kontrolnih varijabli.

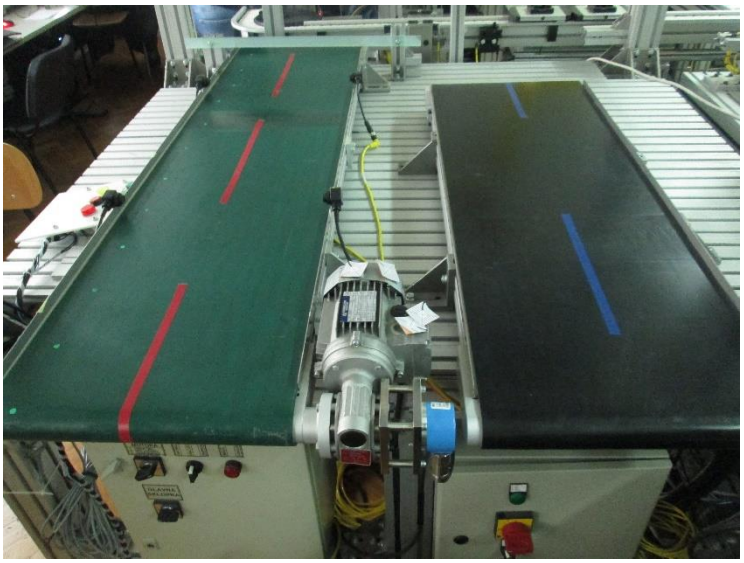
# 2. PREGLED KORIŠTENIH KOMPONENTI

## 2.1. Transportni sustav

Transportni sustavi su osnova za transport strojnih djelova i materijala s jednog mjesta na drugo, prisutni u svim granama industrije. Najčešće korišteni transportni sustavi su transportne trake koje se sastoje od beskonačne gumene ili plastične trake, te dva bubnja od kojih je jedan pogonski dok drugi služi za zatezanje trake. Za pogonjenje trake najčešće se koristi elektromotor na koji može biti montiran i enkoder za povratnu informaciju o kretanju trake.

### *2.1.1. Transportne trake*

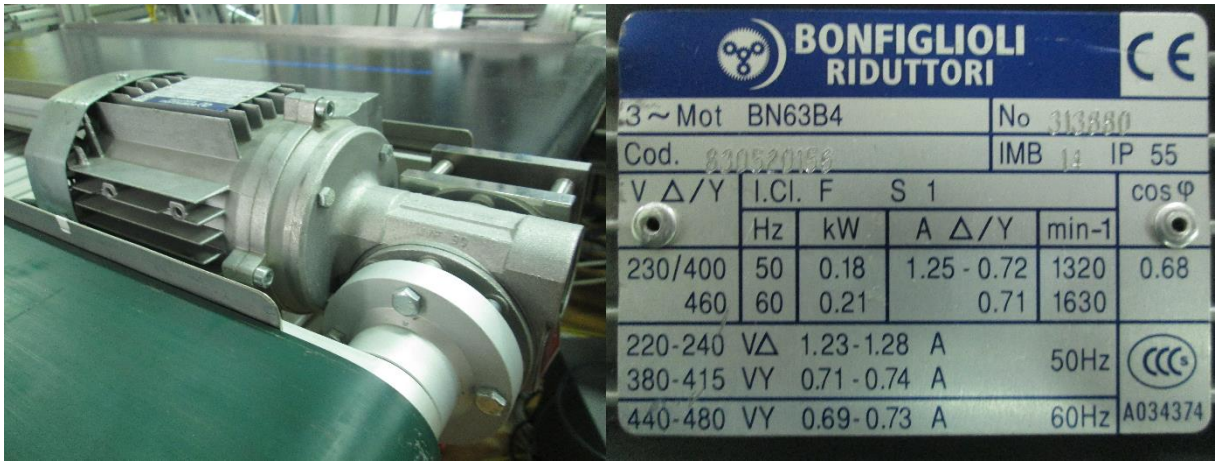
U sklopu transportnog sustava korištene su dvije transportne trake slovenskog proizvođača *Lipro.d.o.o*. Na nazivnim pločicama nisu otisnuti parametri poput nosivosti, maksimalne brzine okretanja, te se može zaključit samo da su trake izrađene od gume, duljina 160 cm i 120 cm. Za potrebe ovog rada to su dovoljne informacije, jer će se raditi sa relativno malim brzinama i laganim predmetima za testiranje [Slika 4].



**Slika 4. Pokretne trake.**

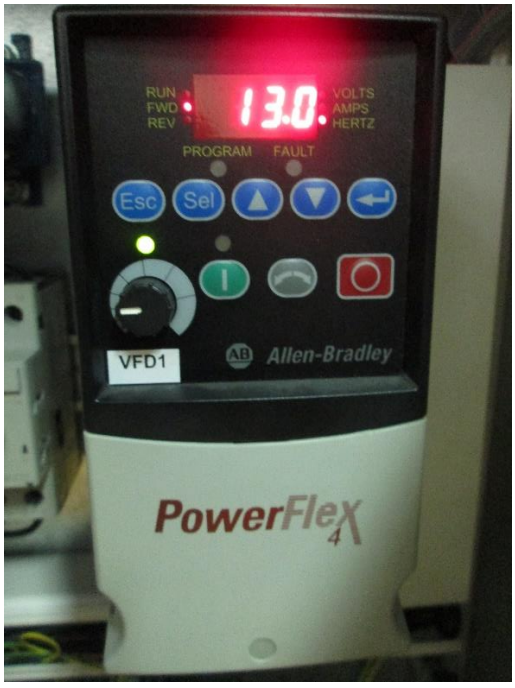
### *2.1.2. Elektromotori i upravljačke jedinice elektromotora*

Za pogon pokretnih traka korišteni su elektromotori sa ugrađenim reduktorima talijanskog proizvođača *Bonfiglioli*. Parametri elektromotora vidljivi su na nazivnoj pločici [Slika 5].




**Slika 5. Bonfiglioli elektromotor (lijevo) sa nazivnom pločicom (desno).**

Upravljanje elektromotorima ostvareno je putem dvije *Allen-Bradley PowerFlex 4* upravljačke jedinice. U ovom slučaju, motorima se upravlja putem frekvencije izmjenične struje [Slika 6].




**Slika 6. Allen-Bradley PowerFlex 4 upravljačka jedinica.**

### *2.1.3. Enkoderi*

U slučaju rotacijskih strojnih elemenata koriste se kutni digitalni pretvornici pomaka, ili jednostavnije rotacijski enkoderi. Osnovna podjela je na apsolutne i inkrementalne enkodere. Razlika je u tome što apsolutni enkoderi imaju jedinstven kod za svaku poziciju, dok inkrementalni enkoderi daju informacije o pomaku u odnosu na referentnu točku, te se koriste

u slučajevima kada treba precizno odrediti brzinu, udaljenost, poziciju ili smjer [3]. Za potrebe ovog rada korištena su dva inkrementalna enkodera, *SICK inc. encoder* i *FANUC pulse code unit* [Slika 7].

**Slika 7. Inkrementalni enkoderi.**

## 2.2. Vizijski sustavi

Kada se govori o vizijskim sustavima razlikuju se dva osnovna pojma: računalni i strojni vid. Računalni vid podrazumijeva akviziciju slike ili videa putem senzora (kamere), te obradu i interpretaciju dobivenih informacija računalnim programima i algoritmima. S druge strane, strojni vid ima svrhu da osim interpretacije slike šalje povratne informacije na neki izvršni član (upravljačku jedinicu robota, PLC) kako bi se dobivene informacije iskoristile za rukovanje predmetima rada u realnom vremenu i okolini, što je slučaj i u ovom radu [4] [5].

### *2.2.1. Kamere*

Iznad transportnih traka montirane su *Basler ace 1300 – 30gm* i *Sony XC – 56* industrijske kamere. Kako bi fokus slike bio u ravnini traka, na svaku od kamera montiran je i objektiv za izoštravanje, zumiranje i namještanje ekspozicije slike [Slika 8].

**Slika 8. Sony industrijska kamera (lijevo), Basler industrijska kamera (desno).**

### *2.2.2. Komunikacija računalo – kamera – računalo*

Za komunikaciju kamera sa računalom, odnosno upravljačkom jedinicom robota, korišteni su mrežni „ethernet“ kabeli. „Ethernet“ standard je danas najčešće korištena tehnologija za lokalne mreže (LAN). Radi na principu komunicirajućih sustava, koji putem „ethernet-a“ podijele niz podataka u manje djelove zvane okviri, gdje svaki okvir sadržava adrese izvora i cilja, te u isto vrijeme provjerava podatke na greške kako bi se oštećeni podaci uklonili.

### *2.2.3. Računalni programi za obradu i interpretaciju slike*

U slučaju transportne trake za odlaganje predmeta rada, korišten je *FANUC*-ov integrirani vizijski sustav, *iR-Vision*, dok je na pokretnoj traci za izuzimanje predmeta rada projektiran vlastiti vizijski sustav pomoću *National Instruments* računalnog programa *LabView 2013*. Programiranje u *LabView 2013.* programu izvodi se putem povezivanja blokova različitih funkcija. Premda relativno sporo procesuira velike pakete podataka kao što su slike, izabran je zbog svoje jednostavnosti i praktičnog načina na koji se mogu regulirati parametri interperatacije slike, odnosno nalaženja predmeta.

## 2.3. Industrijski robot

Korišteni industrijski robot je robot paralelne kinematike (heksapod, delta robot), *FANUC M-3iA 6A* [Slika 9]. Osnovna konstrukcija paralelne kinematike sa tri stupnja slobode gibanja daje ovakvim robotima veliku radnu brzinu i preciznost. U ovom slučaju robot ima dodatna tri stupnja slobode za potpunu orijentaciju alata u prostoru, što mu malo smanjuje preciznost. No, u ovom slučaju preciznost robota je iznimno narušena zbog oštećenja pri radu, gdje je došlo do razlike u stvarnom položaju robota i onom u računalnom modelu. Greška je smanjena servisom i postupkom kalibracije, ali i dalje iznosi 20 mm u prosjeku. Posebno je izražena prilikom orijentiranja alata u prostoru oko z osi [6].

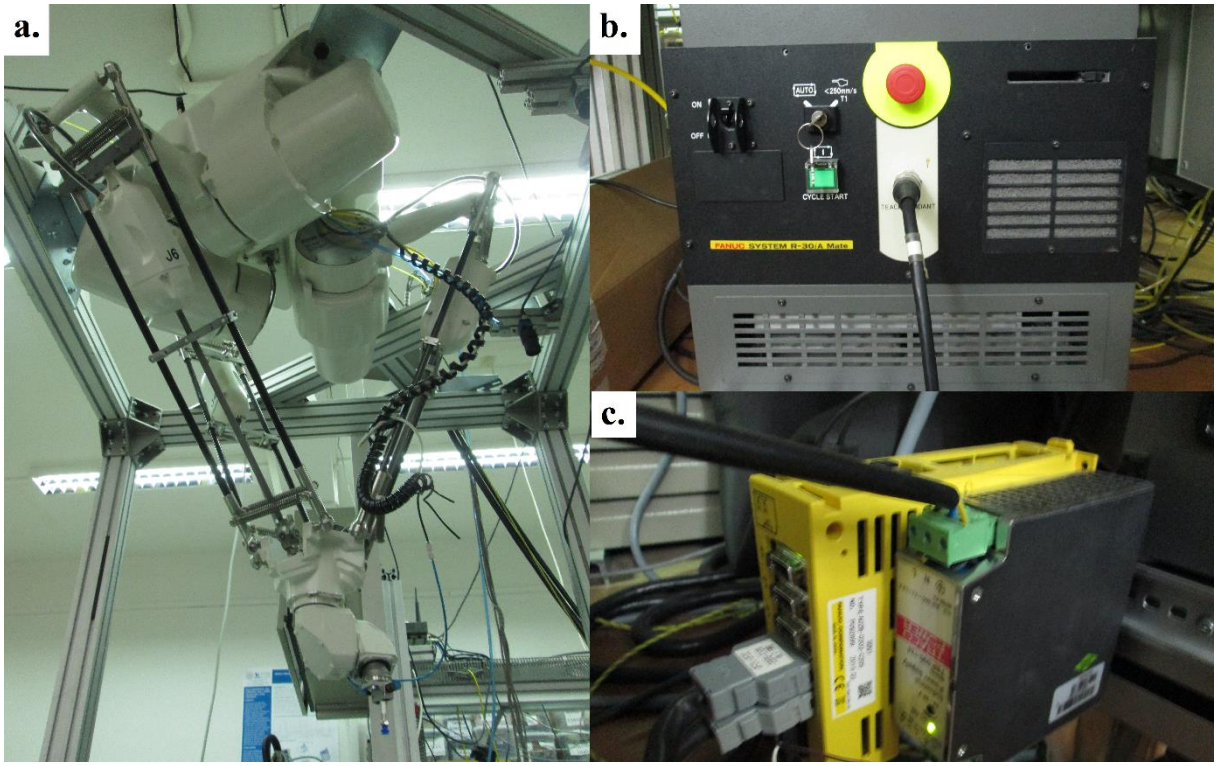
### *2.3.1. Upravljačka jedinica robota*

Upravljačka jedinica robota je *FANUC R-30iA Mate* uređaj, nadograđen sa multiplekserom i napajenjem za isti kako bi se na upravljačku jedinicu mogla spojiti oba enkodera pokretnih traka [Slika 9].

### *2.3.2. TCP komunikacija sa računalom*

Kako bi se povezali upravljačka jedinica robota i računalo u svrhu izmjene informacija za vlastiti vizijski sustav (slanje okidača akvizicije slike i povratnih informacija o položaju, orijentaciji predmeta), korišten je TCP (eng. Transmission Control Protocol) protokol putem „ethernet“ kabela. Korištenjem TCP protokola aplikacija na nekom poslužitelju umreženom u

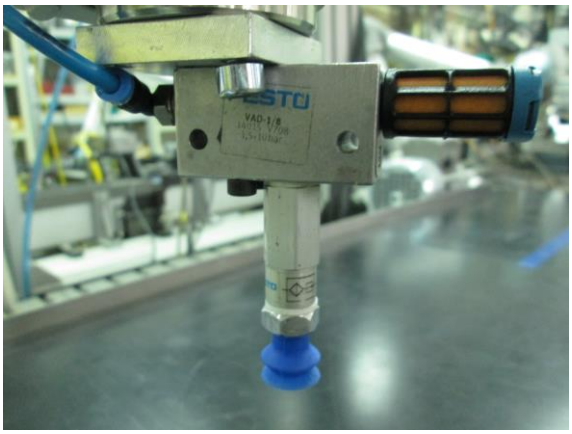
računalnu mrežu kreira virtualnu vezu prema drugom poslužitelju, te putem ostvarene veze prenosi podatke [7].




**Slika 9. a. FANUC M-3iA 6A industrijski robot, b. FANUC SYSTEM R-30iA Mate kontroler, c. FANUC multiplekser sa napajanjem.**

### *2.3.3. Predmet rada i alat – vakuum sisaljka*

S obzirom na istraživačku prirodu ovog rada i potrebu za mjerenjem, korišteni su jednostavni predmeti rada sa isprintanim teksturama preko kojih se lako može očitati točnost izuzimanja, odnosno odlaganja predmeta [Slika 10]. Za alat je upotrebljena vakuum sisaljka vanjskog promjera 14 mm, montirana na *FESTO* vakuum generator [8].




**Slika 10. Vakuum sisaljka na FESTO vakuum generatoru.**

# 3. PROJEKTIRANJE I REALIZACIJA STROJNOG VIDA ZA IZUZIMANJE PREDMETA RADA SA TRANSPORTNE TRAKE

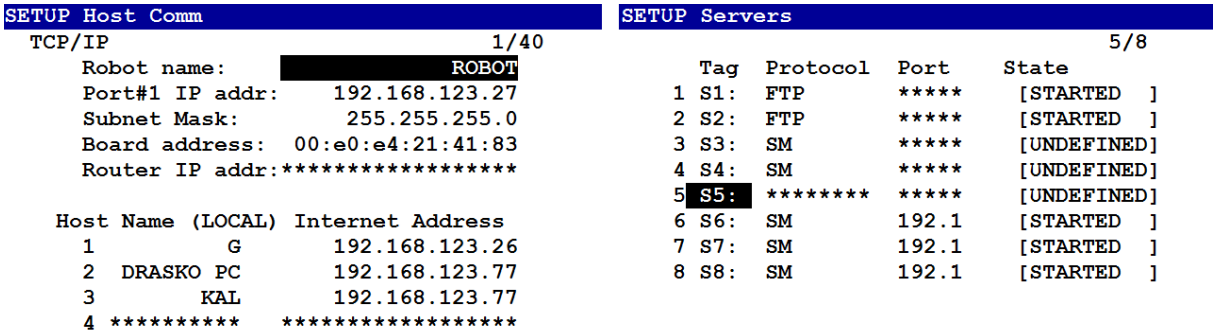
Za računalni program strojnog vida izuzimanja predmeta rada sa pokretne trake postoji nekoliko izbora koji su u širokoj uporabi. *Open CV* program sa knjižnicom za programiranje vizije, koji pruža veliki spektar mogućnosti prilagođavanja potrebama korisnika, *National Instruments LabView* paket koji i dalje pruža fleksibilnost, ali uz jednostavnije grafičko sučelje i programiranje pomoću blok funkcija, te *Nation Instruments Vision Assistant,* pojednostavljeni program za potrebe vizijskih sustava sa ograničenim mogućnostima. Za projektiranje strojnog vida u ovom radu izabran je *LabView 2013* program, upravo zbog intuitivnog načina programiranja, te dovoljno funkcija da se ostvari komunikacija sa upravljačkom jedinicom robota, fino podese parametri prepoznavanja predmeta rada i matematički obrade potrebne informacije o poziciji i orijentaciji istog. Negativna strana je kašnjenje koje se javlja zbog vremena potrebnog da se obavi akvizicija i obrada slike i informacije pošalju nazad na upravljačku jedinicu. Jedan od glavnih ciljeva ovog rada je naći jednostavno i robusno rješenje za taj problem.

## 3.1. Uspostavljanje komunikacije sa upravljačkom jedinicom robota

Komunikacija između *FANUC* upravljačke jedinice i *LabView* programa fizički je ostvarena povezivanjem „ethernet“ kabelom, a prijenos podataka vrši se klijent-server modelom „socket messaging-a“. „Socket messaging“ podrazumijeva otvaranje kanala za prijenos podataka između dva uređaja, gdje je u ovom slučaju upravljačka jedinica robota server, a *LabView* program klijent [9]. Pri pokretanju programa, server je u stanju slušanja, čekajući na inicijativu za spajanjem od strane klijenta. Ostvarivanje takve veze počinje sa strane upravljačke jedinice, pisanjem koda za inicijalizaciju komunikacije u programskom jeziku *KAREL,* namijenjenom za programiranje *FANUC* industrijskih robota.

Pomoću privjeska za učenje pristupa se TCP/IP postavkama robota, odakle se može vidjeti zadana IP adresa, te definirirati do osam server i klijent kanala [Slika 11]. Prvi dio programa služi upravo za resetiranje varijabli u izabranom server kanalu pomoću SET_VAR funkcije, gdje gasimo taj kanal u slučaju da je bio upaljen, te osiguravamo željene parametre servera kao što su ispravan protokol za povezivanje, „port“, „timeout“ vrijeme itd. Idući korak je postaviti željeni server u operativno stanje, također pomoću SET_VAR funkcije, te definiriati atribute FILE datoteke na koju će se pisati, odnosno izmjenjivati informacije između dva

uređaja. Definiranje atributa FILE varijable, u ovom slučaju nazvane file_var, izvodi se pomoću SET_FILE_ATR funkcije, gdje definiramo atribut interaktivnog pisanja na datoteteku pomoću ATR_IA značajke. Na taj način informacije se šalju kad god upravljačka jedinica izvrši funkciju pisanja na datoteku [10]. „Socket messaging“ postupak započinje naredbom MSG_CONNECT gdje se definira kanal preko kojeg će se odvijati komunikacija. WHILE petljom osigurava se stanje slušanja, kako bi server neprekidno čekao da se klijent spoji. Zadnji bitni korak spajanja je otvoriti datoteku file_var naredbom OPEN FILE, gdje se „string“ parametrom 'rw' definira „read-write“ svojstvo te datoteke, kako bi se informacije mogle slati i primati u isto vrijeme [Slika 12].

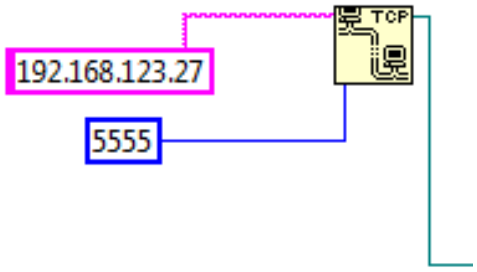



**Slika 11. TCP/IP informacije na privjesku za učenje.**

```
020   --RESETIRANJE TAGOVA
021   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$OPER',0,STATUS);
022   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STATE',0,STATUS); DELAY 20
023   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$COMMENT','SOUND',STATUS);
024   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$PROTOCOL','SM',STATUS);
025   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$REPERRS','FALSE',STATUS);
026   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$TIMEOUT',9999,STATUS);
027   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$PWRD_TIMEOUT',0,STATUS);
028   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$SERVER_PORT',5555,STATUS);
029   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STRT_PATH','192.168.123.27',STATUS);
030   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STRT_REMOTE','192.168.123.27',STATUS);
031   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$PATH','192.168.123.27',STATUS);
032   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$remote','192.168.123.27',STATUS);
033   ----------------------------------------
034   --POKRETANJE TAG 5 NAKON RESETIRANJA I KONEKCIJA
035   SET_FILE_ATR(file_var, ATR_IA)
036   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$OPER',3,STATUS);
037   SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STATE',3,STATUS);
038   WRITE('Connecting..',CR)
039   MSG_DISCO('S5:',STATUS)
040   IF STATUS=0 THEN
041       WRITE('odspojen',CR)
042   ENDIF
043   MSG_CONNECT('S5:',STATUS)
044   WHILE STATUS<>0 DO
045       MSG_CONNECT('S5:',STATUS)
046   ENDWHILE
047   WRITE (STATUS,CR)
048   CLR_IO_STAT(file_var)
049   OPEN FILE file_var('rw','S5:')
```

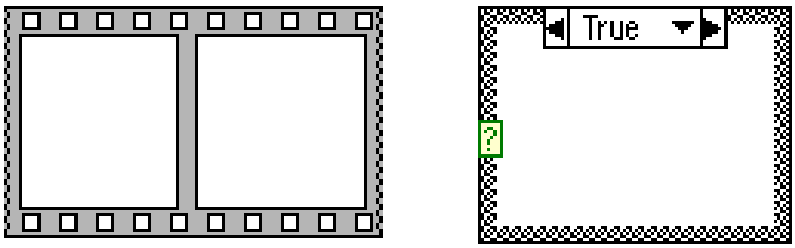
**Slika 12. KAREL kod za uspostavljanje socket messaging komunikacije.**

Nakon što je osigurano stanje slušanja na strani servera, *LabView* se mora spojiti na zadanu IP adresu (192.168.123.27) i „port“ (5555) koji su zadani pri inicijalizaciji u *KAREL* kodu. *LabView* koristi grafički pristup programiranju pomoću blok dijagrama gdje svaki blok predstavlja jednu od naredbi, odnosno funkcija, sa naznačenim ulazima, izlazima ili parametrima koje treba spojiti. Iz izbornika „Data Communication – Protocols – TCP“ bira se blok „TCP Open Connection Function“. Kao ulaze obavezno je unijeti IP adresu i „port“ na koji se želi spojiti. Na izlazu daje identitet (ID) ostvarene veze koji se kasnije koristi u ostalim TCP funkcijama kao referenca [Slika 13].




**Slika 13. TCP Open Connection blok sa ulazima (lijevo) i izlazom (desno).**

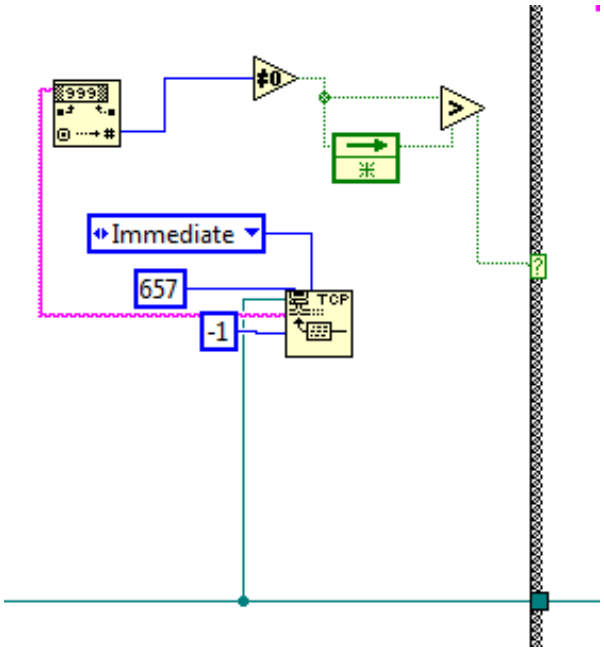
Osnovna struktura ovog programa napravljena je unutar okvira strukture nazavne „Flat Sequence Structure“ [Slika 14], gdje se unutar svakog okvira izvrši niz funkcija, a prelazak u idući okvir ostvaruje se kada iz svih izlaza prethodnog okvira prođu informacije. U prvom okviru uspostavlja se konekcija (izlaz je ID veze), te učitavanje referente („template“) slike za pronalaženje predmeta (izlaz je informacija o slici). U drugom okviru čeka se na okidač sa upravljačke jedinice za akviziciju i obradu slike. Funkcije akvizicije i obrade informacija o predmetima nalaze se unutar uvjetnog („Case Structure“ [Slika 14]) okvira, tako da se niz za akviziciju i obradu odradi samo kada *LabView* dobije informaciju od upravljačke jedinice.




**Slika 14. Flat Sqeuence Structure okvir (lijevo) i Case Structure okvir (desno).**

Okidač koji kontroler šalje je jednostavna „string“ datoteka sadržaja „1“. Pomoću „TCP Read“ bloka pročita se dobivena informacija, te se „Decimal String To Number“ funkcijom

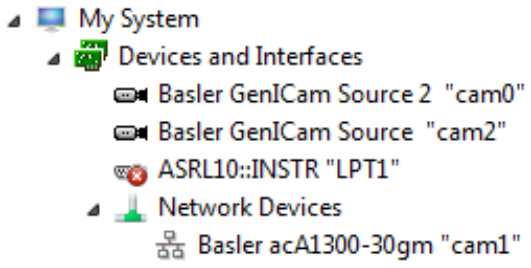
konvertira iz „string“ varijable u „number“ varijablu. Kako bi okidač pokrenuo „Case Structure“ okvir samo jednom, „Feedback Node“ blokom se osigurava pokretanje samo na podižući brid signala. „Feedback Node“ pamti zadnju informaciju na izlazu iz nekog bloka, te je pomoću logičke funkcije „veće od“, uspoređuje sa novom. Na taj način ako program očita okidač više puta u nekom kratkom periodu vremena, „Case Structure“ će odraditi niz samo prvi puta kada okidač prelazi iz vrijednosti 0 u 1 [Slika 15].



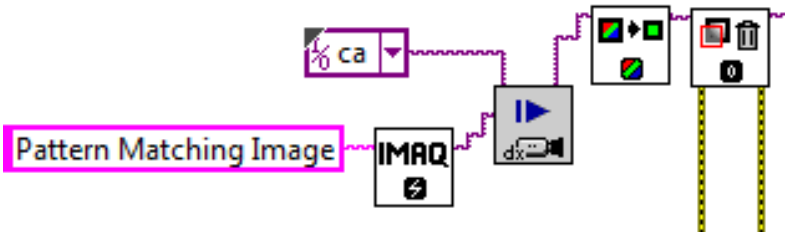

**Slika 15. Čitanje okidača i pokretanje niza funkcija za akviziciju i obradu slike.**

## 3.2. Akvizicija slike i prepoznavanje predmeta rada

Prvi korak je odrediti mjesto u privremenoj memoriji za spremanje slike prilikom akvizicije, funkcijom „IMAQ Create VI“, te u zasebnom programu *NI MAX* , registrirati korištenu kameru i dati joj naziv [Slika 16]. Prazna slika iz „IMAQ Create VI“ bloka i naziv korištene kamere su ulazi u blok za akviziciju slike, „IMAQdx Snap VI“. Dobivena slika prolazi filter za izdvajanje određenih kanala boja zbog prilagodbe formata slike koji je potreban za daljnje čitanje iste [Slika 17].

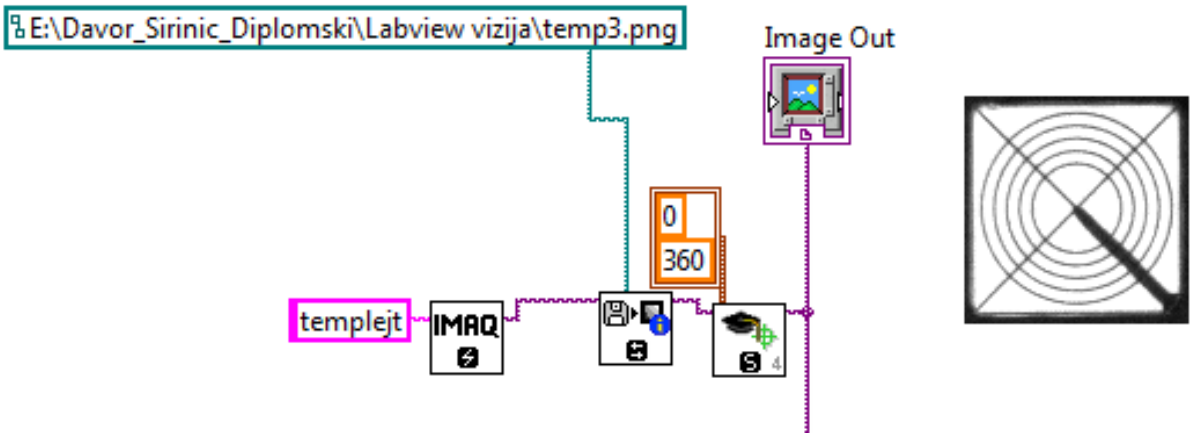



**Slika 16. Izbornik NI MAX programa, na dnu korištena *Basler* kamera naziva „cam1“.**



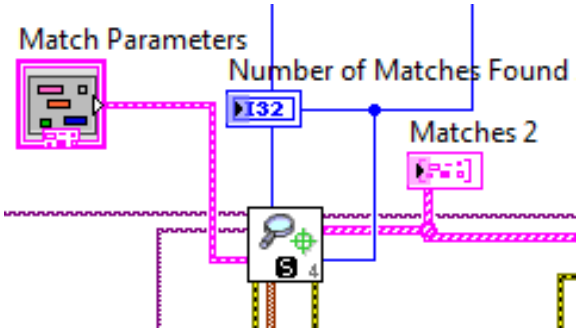

**Slika 17. Niz funkcijskih blokova za akviziciju slike.**

Kako bi se predmet rada pronašao na slici, mora se učitati uzorak željenog predmeta rada. U prvom okviru na početku samog programa se pomoću „IMAQ Create VI“ funkcije stvara mjesto u privremenoj memoriji za sliku uzorka, te se isti učitava sa lokacije na računalu. „IMAQ Learn Pattern 4 VI“ blok omogućava stvaranje opisa uzorka pomoću kojeg će se tražiti predmet na slici. Moguće je podesiti parametre kao što je orijentacija predmeta, algoritmi kojima će se učiti određene značajke uzorka, te maska kojom je moguće označiti značajke koje se ignoriraju pri prepoznavanju [Slika 18].

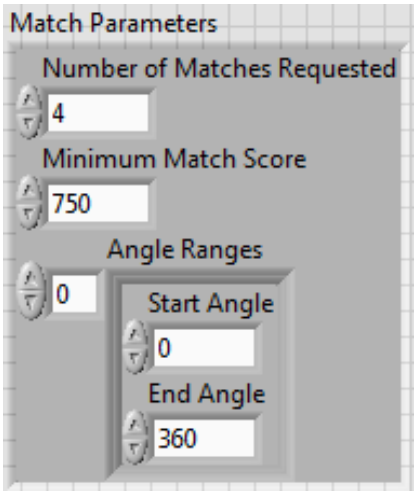



**Slika 18. Učitavanje i učenje uzorka (lijevo) predmeta rada (desno).**

Blok za prepoznavanje predmeta rada na slici „IMAQ Match Pattern 4 VI“ za ulaze ima sliku uzorka sa opisom značajki za prepoznavanje, te sliku na kojoj se traži predmet. Otvaranjem bloka, dobije se pristup parametrima traženja kao što su količina preklapanja, raspon orijentacije predmeta, te maksimalni broj nađenih predmeta. U svrhu korištenog eksperimentalnog postava taj broj je ograničen na 4 [Slika 20]. Bitni izlazi su polje koje sadrži nađene predmete i informacije o njihovom položaju i orijentaciji, te broj nađenih uzoraka. Broj nađenih uzoraka koristi se kao uvjet za odrađivanje niza funkcija u „Case Structure“ okviru [Slika 19]. U slučaju da nije nađen ni jedan predmet, informacije o predmetu se neće obrađivati ni slati nazad na upravljačku jedinicu robota.

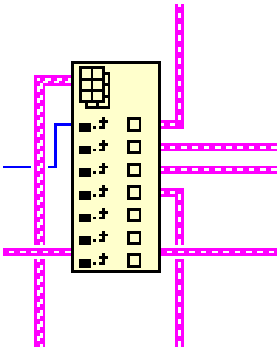



**Slika 19. „IMAQ Match Pattern 4 VI“ funkcijski blok.**




**Slika 20. Parametri traženja predmeta.**

Izlaz sa nađenim predmetima rada i njihovim informacijama šalje se u „Index Array“ blok, gdje se izdvajaju u četri zasebna izlaza polja koja sadrže informacije o poziciji i orijentaciji za svaki predmet rada [Slika 21].

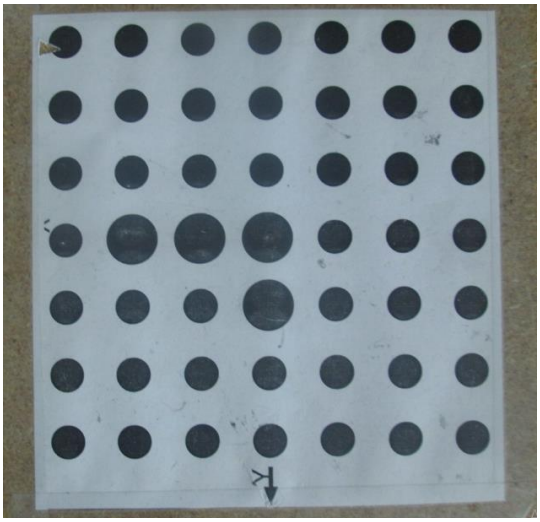


**Slika 21. Funkcijski blok „Index Array“ za izuzimanje elemenata iz polja.**

Drugi korak prije obrade informacija o nađenim predmetima je kalibrirati kameru i definirati referentni koordinati sustav u vidnom polju kamere. Za potrebe kalibracije projektiran je zasebni *LabView* program.

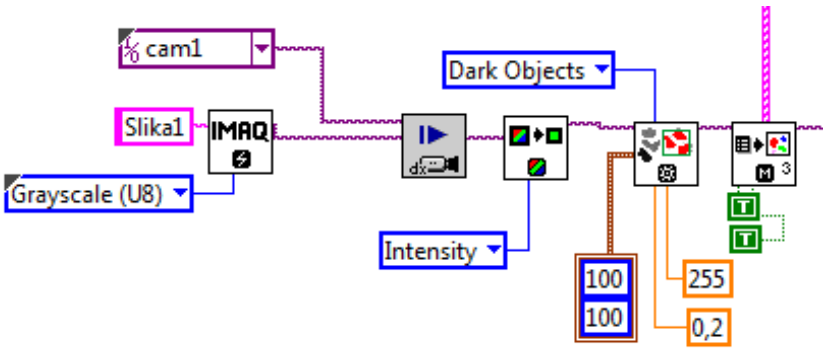
## 3.3. Kalibracija kamere i zadavanje referentnog koordinatnog sustava

Za kalibraciju kamere odabran je postupak pomoću mreže točaka za kalibraciju [Slika 22].



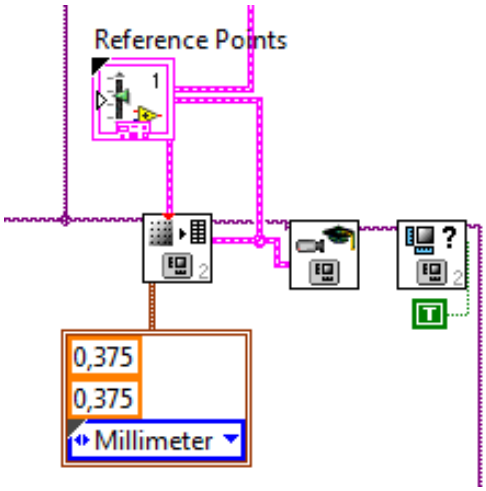
**Slika 22. Mreža točaka za kalibraciju kamere.**

Za konačnu točnost pronalaženja predmeta u referentnom koordinatnom sustavu, bitno je što točnije napraviti kalibraciju kamere i odrediti koordinatni sustav. Prije učenja karakteristika kamere, prikupljena slika kalibracijske mreže prolazi kroz nekoliko filtera kako bi se što bolje izdvojile točke na mreži [Slika 23]. S obzirom da funkcijski blok za detekciju točaka kalibracijske mreže zahtjeva binarnu sliku, slika se nakon filtriranja konvertira iz 8 bitne slike sivih tonova u binarnu.



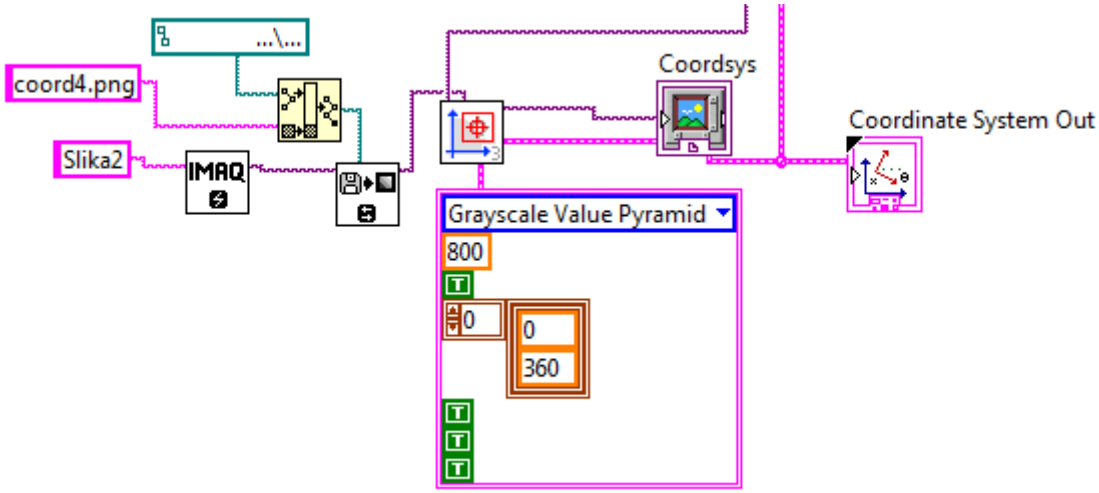

**Slika 23. Filteri za pripremu slike kalibracije kamere.**

Binarna slika kalibracijske mreže prolazi kroz „IMAQ Calibration Target to Points – Circular Dots VI“ blok u kojem se definira odnos između piksela na slici i stvarnih udaljenosti u milimetrima (30 mm udaljenosti između točaka mreže), te kao izlaz daje referentne točke za kalibraciju. Kalibracija se odvija unutar „IMAQ Learn Camera Model VI“ bloka, gdje program uči žarišnu daljinu, optički centar i distorziju kamere [Slika 24].




**Slika 24. Funkcijski blokovi za kalibraciju kamere.**

Informacije o modelu kamere prosljeđuju se u „IMAQ Find CoordSys (Pattern) 3 VI“ blok gdje se pomoću traženja uzorka na slici definira pozicija i orijentacija referentnog koordinatnog sustava. Odabrani uzorak je centar mreže za kalibraciju kako bi se kalibracija i definiranje koordinatnog sustava obavili u istom postupku. Konačno, kao izlaz cijelog programa dobiva se globalna varijabla koordinatnog sustava u vidnom polju kamere [Slika 25].

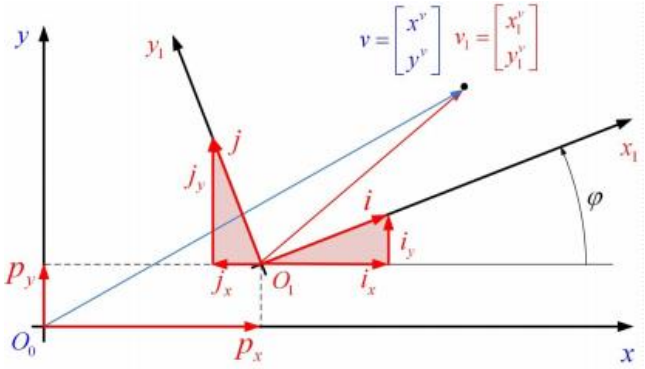



**Slika 25. Definiranje koordinatnog sustava pomoću traženja učitanog uzorka.**

## 3.4. Transformiranje koordinata nađenih predmeta rada u referentni koordinatni sustav

Nakon što je jednom obavljena kalibracija kamere i definiran referentni koordinatni sustav, program može obrađivati informacije o poziciji i orijentaciji nađenih predmeta rada, te ih slati na upravljačku jedinicu robota. Transformacija koordinata u ravnini sastoji se od translacije i rotacije. Za transformaciju potrebno je znati x i y koordinate točke u početnom koordinatnom

sustavu, x i y koordinate ishodišta novog koordinatnog sustava unutar početnog, te kut zakreta između istih. Transformacija se matematički izvodi putem matrice transformacija A (2), a u *LabView* programu nizom blokova matematičkih funkcija.

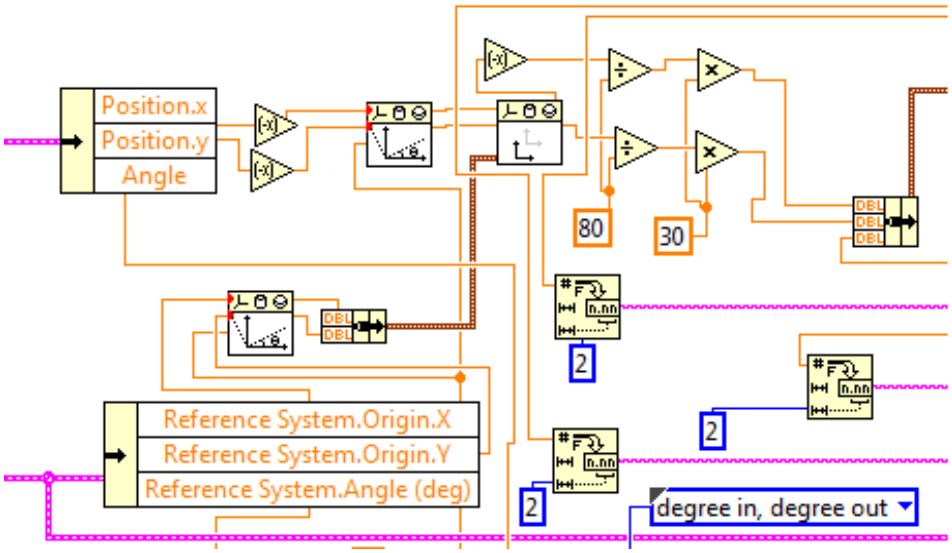



**Slika 26. Transformacija koordinata točke u ravnini [11].**

$$\begin{bmatrix} x^v \\ y^v \\ 1 \end{bmatrix} = \begin{bmatrix} cos\varphi & -sin\varphi & p_x \\ sin\varphi & cos\varphi & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1^v \\ y_1^v \\ 1 \end{bmatrix} \quad \textbf{(1)}$$

$$A = \begin{bmatrix} cos\varphi & -sin\varphi & p_x \\ sin\varphi & cos\varphi & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad \textbf{(2)}$$

Funkcijskim blokovima za rotaciju i translaciju transformira se centar nađenog predmeta rada iz koordinatnog sustava vidnog polja kamere (ishodište u donjem lijevom kutu slike) u koordinatni sustav mreže za kalibraciju, čije su se $p_x$ i $p_y$ koordinate spremljene u globalnoj varijabli učitane iz programa za kalibraciju.

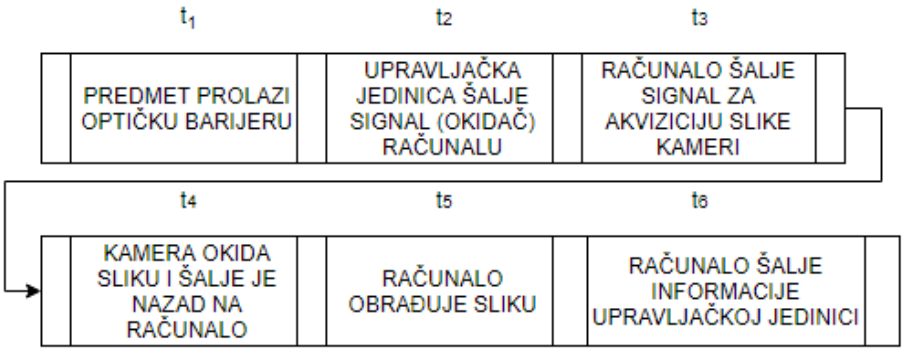



**Slika 27. Transformacija koordinata nađenog predmeta.**

Transformacija koordinata se izvršava u pikselima, tako da se preko omjera 80/30 dobiju vrijednosti u milimetrima. Omjer je dobiven mjerenjem piksela na slici između dvije točke na mreži za kalibraciju (80 piksela), za koje znamo da su udaljene 30mm u stvarnosti [Slika 27] [12] [13].

## 3.5. Rješenje problema kašnjenja informacija o poziciji predmeta rada

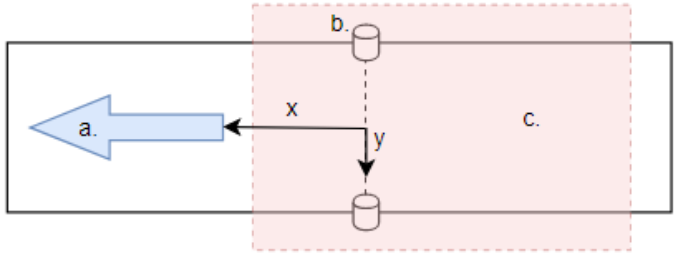
U slučaju kamere fiksirane van robota i nepomičnog radnog prostora, robot se uči referentnoj poziciji i orijentaciji za hvatanje predmeta rada, te informacije o pomaku predmeta dobiva preko vizijskog sustava. U slučaju transportne trake sustav postaje dinamičan, unosi se vrijeme kao nova varijabla, odnosno pomak predmeta rada duž transportne trake u vremenu. Problem pomaka predmeta rada u vremenu rješen je i bez vizijskog sustava integriranom *„FANUC Line tracking“* opcijom. Inkrementalnim enkoderom na motoru transportne trake i optičkom barijerom na samoj traci, dobiva se povratna informacija o brzini gibanja predmeta i trenutku kada predmet prođe točku definiranu optičkom barijerom. No, to je rješenje jednodimenzijskog problema, gdje se predmet giba po jednom pravcu. U slučaju kada su potrebne dodatne informacije o orijentaciji i poziciji predmeta duž pravca okomitog na smjer gibanja trake koristi se vizijski sustav. Novi problem javlja se zbog kašnjenja informacija o predmetu rada, jer je potrebno određeno vrijeme dok vizijski sustav primi okidač sa upravljačke jedinice robota, obavi akviziciju i obradu slike, te pošalje nazad informacije [Slika 28].




**Slika 28. Tok signala od trenutka prekida optičke barijere do dobivanja povratne informacije o predmetu rada.**

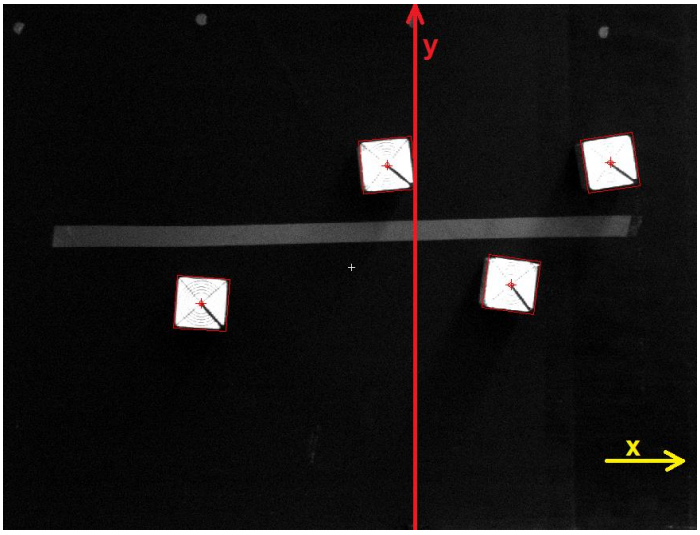
Kao što se vidi na dijagramu toka signala, ukupno vrijeme kašnjenja *t*, jednako je zbroju vremena svakog procesa, od kojih je najduže vrijeme obrade slike i pronalaženja predmeta rada, $t_5$. Ostala vremena su reda par desetaka milisekundi do par stotina milisekundi, tako da ukupno vrijeme kašnjenja može iznositi i više od jedne sekunde. Prema tome, prvi većim

brzinama transportne trake predmet rada se u trenutku izuzimanja može nalaziti i do par stotina milimetara dalje od naučene točke izuzimanja. Drugi problem javlja se kod zbroja prva tri vremena ($t_1$, $t_2$, $t_3$) , jer ovisno o brzini trake predmet ima različitu x koordinatu prilikom akvizicije slike, uslijed kašnjenja od trenutka prekidanja barijere do okidanja slike. Y koordinata predmeta, te orijentacija ne predstavljaju problem jer su konstantne u vremenu.




**Slika 29. Shema transportnog sustava: a. smjer gibanja trake, b. linija optičke barijere, c. vidno polje kamere**

Prvi problem zbog ukupnog vremena kašnjenja rješava se poklapanjem ishodišta koordinatnih sustava robota i kamere na transportnoj traci i to tako da su im ishodišta u točci gdje predmet prekida optičku barijeru. Na taj način pomoću *„FANUC Line tracking“* procesa, predmet će se uvijek pratiti od točke prekidanja barijere, jer robot tad zapamti referentnu vrijednost enkodera prema kojoj u trenutku izuzimanja putem naučene skale enkodera zna gdje se predmet nalazi duž trake [14]. No, zbog kašnjenja do akvizicije slike računalo će slati x koordinatu prvog predmeta različitu od nule, dok robot računa x pomak od trenutka prekidanja optičke barijere (ishodište koordinatnog sustava) [Slika 30].

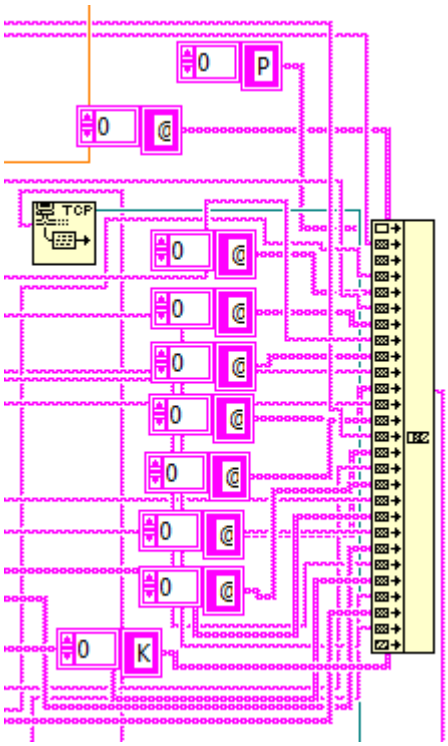



**Slika 30. Nađeni predmeti rada sa linijom optičke barijere, ujedno i y os koordinatnih sustava (crveno) i x osi koordinatnih sustava, tj. smjerom kretanja (žuto).**

Problem je rješen u *LabView* programu sortiranjem nađenih predmeta na jednoj slici prema x koordinatama, od najmanjeg prema najvećem. Zbog praktičnih razloga (laboratorijskih uvjeta, ograničenog radnog prostora) maksimalni broj nađenih predmeta na jednoj slici ograničen je na četri. Kada su predmeti sortirani po x koordinati, od svakog se oduzima x koordinata prvog predmeta, kako bi se kompenzirao x pomak zbog kašnjenja. Prvi predmet ima x koordinatu nula, dok je ostalim predmetima x koordinata jednaka udaljenosti od prvog predmeta, odnosno ishodišta koordinatnog sustava. Na taj način robot neovisno o brzini transportne trake promatra pomak predmeta rada od trenutka prekidanja optičke barijere.

### 3.6. Slanje koordinata predmeta rada na upravljačku jedinicu robota

Slanje koordinata nađenih predmeta rada na upravljačku jedinicu robota izvršava se u *LabView* programu blokom „TCP Write Function“. Provodi se parsiranje, odnosno koordinate se slažu u „string“ varijablu koja započinje sa znakom „P“ koji označava početak koda, razdvajaju se znakom „@“, te poruka završava znakom „K“. Prve četri koordinate su x koordinate, druge četri y koordinate i zadnje četri orijentacije predmeta rada. Trinaesta informacija je broj nađenih predmeta rada na slici.




**Slika 31. Parsiranje i slanje koordinata nađenih predmeta.**

# 4. RAZVOJ PROGRAMA ROBOTSKOG IZUZIMANJA I ODLAGANJA PREDMETA RADA U POKRETU POMOĆU STROJNOG VIDA

## 4.1. Razvoj programa za izuzimanje predmeta rada u pokretu

Glavni program za inicijalizaciju registara, enkodera i pozadinskih programa nazvan je „PR_TRACKING“ [Slika 32].

```
PR_TRACKING
                                                    1/30
   1:  DO[109]=ON
   2:L @P[1] 350mm/sec CNT100
   3:  R[36]=0
   4:  R[21]=1
   5:  R[16]=0
   6:  R[14]=1
   7:  R[17]=0
   8:  R[27]=0
   9:  WAIT   1.50(sec)
  10:  CALL ENKODER
  11:  RUN SIRINIC_KOM
  12:  RUN PR_TEST5
  13:  CALL VSTKCLRQ('RADNA_PLOHA')
  14:  CALL VSTKSTRT
  15:  RUN PR_ACQ
  16:  LINE[1] ON
  17:  WAIT DI[102]=ON
  18:  DO[112]=ON
  19:  LBL[100]
  20:
  21:  CALL PROBA
  22:  IF R[17]=1,JMP LBL[900]
  23:  CALL VSTKSTRT
  24:L @P[1] 650mm/sec CNT100
  25:  JMP LBL[100]
  26:
  27:  LBL[900]
  28:  CALL VSTKSTOP
  29:  DO[109]=OFF
[End]
```

**Slika 32. Glavni program „PR_TRACKING“.**

U razvoju su korištene dvije vrste *FANUC* programa, programi pisani u *KAREL* jeziku i *TP* (eng. Teach Pendant) programi, pisani na privjesku za učenje. *KAREL* programi su pretežito pisani u svrhu pozadinske obrade informacija i za uspostavljanje komunikacije, dok su preko *TP* programa učene točke u prostoru i izvršavanje radnji za izuzimanje, odnosno odlaganje predmeta. Prvi program koji se poziva je *KAREL* program za komunikaciju u kojem upravljačka jedinica kao server čeka na spajanje sa klijentom, računalom [Slika 12]. U nastavku programa nalazi se „WHILE“ petlja gdje se konstantno čeka na signal rastućeg brida okidača (prekid optičke barijere), te se u tom slučaju šalje signal („string“ varijabla koja sadrži znak „1“) *LabView* programu za akviziciju i obradu slike. Unutar glavne „WHILE“

petlje koristi se druga „WHILE“ petlja koja se vrti u beskonačnost dok upravljačka jedinica na otvorenom kanalu za komunikaciju ne primi podatke od računala. Ako primi obavlja se parsiranje pomoću zadanih znakova („P“, „@“ i „K“), te se prvi set koordinata sprema u matricu „KOORD[]“ dimenzija 14x14, čiji su elementi varijable „string“, koje sadrže koordinate nađenih predmeta rada. Na taj način stvara se spremnik setova koordinata nađenih predmeta. Kako nebi došlo do preklapanja nađenih predmeta, odnosno da se za isti predmet više puta pospreme koordinate uslijed višestrukog prekidanja svjetlosne barijere, pomoću razlike u vrijednosti enkodera dobivena je duljina vidnog polja kamere. Na taj način akvizicija slike neće se obavljati po prekidu optičke barijere ako svi predmeti rada uhvaćeni pri prvoj akviziciji nisu izašli iz vidnog polja kamere [Slika 33].

```
060     WHILE loop1=TRUE DO
061       WAIT FOR DIN[102]=OFF
062         WAIT FOR DIN[102]=ON
063             SET_INT_REG(34,0,STATUS);
064             DELAY 100
065             GET_REG(22,real_flag,int_value,real_value,STATUS)
066             GET_REG(25,real_flag,int_value2,real_value,STATUS)
067             GET_REG(19,real_flag,n,real_value,STATUS);
068             SET_INT_REG(30,(int_value-int_value2),STATUS);
069           IF((int_value-int_value2)<1984) OR (( int_value-int_value2)>33480)  THEN
070             SET_INT_REG(19,n+1,STATUS);
071             CNV_INT_STR(int_value,1,0,trig);
072             WRITE file_var('1',CR)
073
074             ----------READ PETLJA---------
075            SET_INT_REG(20,1,STATUS);
076            GET_REG(19,real_flag,n,real_value,STATUS);
077
078                 NIZ='';   S='';     K=0 ;
079             BYTES_AHEAD(file_var,nb,STATUS)
080             WHILE nb=0 DO
081                 BYTES_AHEAD(file_var,nb,STATUS)
082             ENDWHILE
083                 WRITE('NB ',nb::3::0,' ', CR)
084                 WHILE nb>0 DO
085                     READ file_var(S::1); IF S=',' THEN;S=' ';ENDIF
086                      SET_INT_REG(20,K,STATUS);
087                     IF (S='@') OR (S='P') THEN; K=K+1; KOORD[n,K]='';
088                             -- IF K=10 THEN;K=9;ENDIF   --dodatan uvijet da K ne pređe izvan raspona od 70
089                          ENDIF
090                         WRITE(S);     -- ISPIS CIJELE PORUKE NA EKRAN
091                          IF (S<>'@') AND (S<>'P') AND (S<>'K') THEN ;KOORD[n,K]=KOORD[n,K]+S;ENDIF
092                          NIZ=NIZ+S
093                         nb=nb-1
094                             BYTES_AHEAD(file_var,nb,STATUS)
095                     IF nb=0 THEN; WRITE (CR);ENDIF
096                 ENDWHILE
097               KOORD[n,14]=trig
098                 SET_INT_REG(34,1,STATUS);
099           ENDIF
100     ENDWHILE
101 END SIRINIC_KOM
```

**Slika 33. *KAREL* program za komunikaciju i parsiranje primljenih informacija.**

Osim programa za komunikaciju inicijalizira se i pozadinski program za kontrolu predmeta rada na traci, kako nebi došlo do višestrukog registriranja predmeta uslijed niza prekidanja optičke barijere. Dok se obavlja izuzimanje predmeta rada sa prve akvizicije slike, neće se zapisivati nova referentna vrijednost enkodera pri prekidanju optičke barijere [Slika 34].

```
PR_TEST5                                    18/18
     1:  LBL[102]
     2:  IF R[16]=0,JMP LBL[102]
     3:  IF R[16]=1,JMP LBL[101]
     4:  LBL[101]
     5:  IF R[27]=1,JMP LBL[103]
     6:  IF R[27]=0,JMP LBL[104]
     7:  LBL[103]
     8:  LINECOUNT[1] R[22]
     9:  R[16]=0
    10:  JMP LBL[102]
    11:  LBL[104]
    12:  LINECOUNT[1] R[22]
    13:  //SETTRIG LNSCH[2] R[22]
    14:  SELBOUND LNSCH[2] BOUND[1]
    15:  R[27]=1
    16:  R[16]=0
    17:  JMP LBL[102]
[End]
```

**Slika 34. Kontrolni program.**

U sklopu kontrole predmeta rada i akvizicije istih, u pozadini se konstantno vrte dva jednostavna programa pozadinske logike, jedan za osiguravanje signala samo na rastućem bridu prekidanja optičke barijere, te drugi za zaustavljanje transportne trake u slučaju da predmet rada prođe sigurnosnu optičku barijeru na granici radnog prostora robota [Slika 35].

```
REDGE                                        5/5
     1:  F[1]=(OFF)
     2:  IF (DI[102] AND !F[2]),F[1]=(ON)
     3:  F[2]=(DI[102])
     4:  IF (F[1]),R[16]=(1)
[End]
```

```
PR_TEST2                                     2/2
     1:  IF (DI[101]),DO[109]=(OFF)
[End]
```

**Slika 35. Program za provjeru rastućeg brida signala (lijevo) i sigurnosni program za zaustavljanje transportne trake (desno).**

Kada se ispune svi kontrolni uvjeti, poziva se program „PROBA_TRACK“ za izuzimanje predmeta rada. Na početku programa također postoji sigurnosni uvjet da su sve informacije o predmetima rada primljene, te se potom poziva program za razvrstavanje primljenih koordinata iz „KOORD []“ matrice u registre [Slika 36].

```
12 BEGIN
13     GET_REG(24,real_flag,n,real_value,STATUS)
14         FOR i = 1 TO 12 DO
15                 CNV_STR_REAL(KOORD[n,i] ,X);
16                 SET_REAL_REG(i,X,STATUS);
17         ENDFOR
```

**Slika 36. Program za upisivanje primljenih koordinata u registre.**

Program za izuzimanje je struktuiran pomoću dvije „FOR“ petlje, gdje se glavna „FOR“ petlja odnosi na setove koordinata, tako da se unutar svakog seta izvrši druga „FOR“ petlja koja redom popuni pozicijski registar PR[3] sa koordinatama pomaka za svaki pronađeni predmet rada [Slika 37].

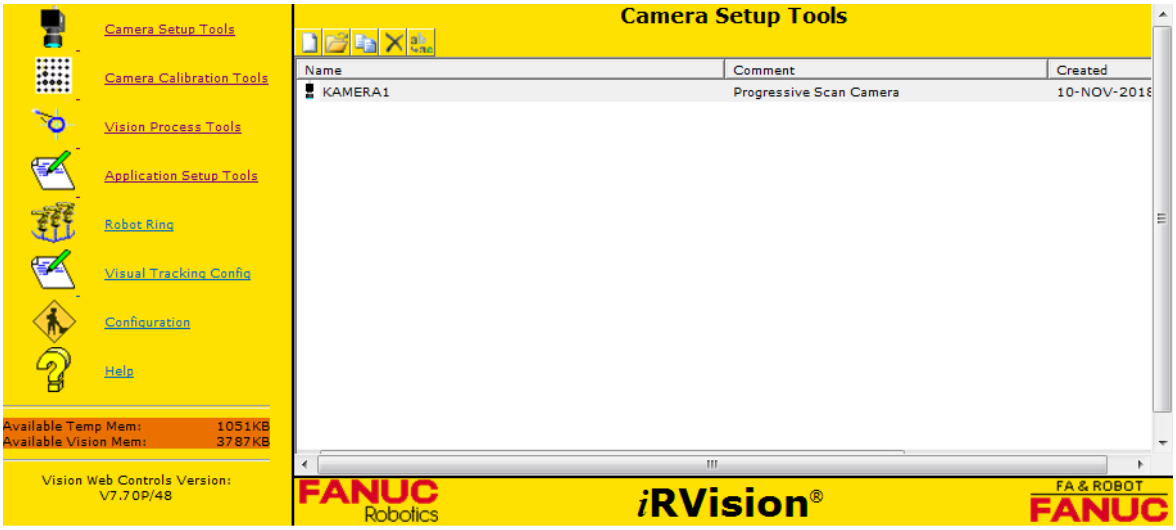
```
PROBA_TRACK
                                                    1/36
      1:  WAIT R[19]>0
      2:  FOR R[24]=1 TO R[19]
      3:  WAIT R[34]=1
      4:  CALL KOORD_SET
      5:  SETTRIG LNSCH[2] R[14]
      6:  PR[3,2]=R[1]
      7:  WAIT R[35]=1
      8:  FOR R[21]=1 TO R[13]
      9:  PR[3]=PR[6]
     10:  R[45]=R[21]+40
     11:  PR[3,1]=R[R[21]]-R[1]-R[R[45]]
       :
     12:  R[21]=R[21]+4
     13:  PR[3,2]=R[R[21]]
     14:  R[21]=R[21]+4
     15:  PR[3,6]=R[R[21]]
     16:  R[21]=R[21]-8
     17:L  PR[7] 650mm/sec FINE
       :  Tool_Offset,PR[2] Offset,PR[3]
       :
     18:L  PR[7] 650mm/sec FINE
     19:  DO[114]=ON
     20:  DO[113]=OFF
     21:  WAIT    .20(sec)
     22:L  PR[7] 650mm/sec FINE
       :  Tool_Offset,PR[2] Offset,PR[3]
       :
     23:  STOP_TRACKING
     24:  DO[109]=ON
     25:  CALL PR_PICK
     26:  ENDFOR
     27:  R[27]=0
     28:  R[35]=0
     29:  R[21]=0
     30:  R[45]=0
     31:  IF R[24]=R[19],JMP LBL[666]
     32:  ENDFOR
     33:  LBL[666]
     34:  R[24]=1
     35:  R[19]=0
[End]
```

**Slika 37. Program za izuzimanje predmeta rada.**

Nakon što se pozicijski registar PR[3] popuni, izvršava se izuzimanje predmeta na naučenoj poziciji spremljenoj u pozicijski registar PR[7]. Nakon učenja pozicije iz registra PR[7] izbrisana je y komponenta, jer se y pomak dobiva preko vizijskog sustava.

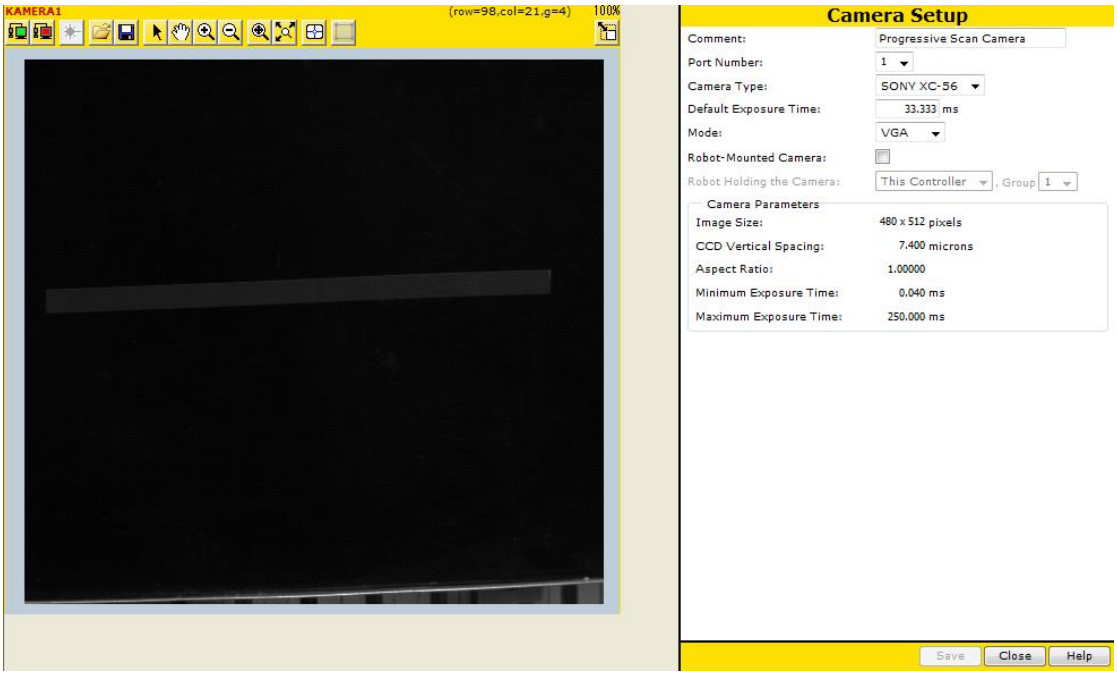
## 4.2. Konfiguracija strojnog vida za odlaganje predmeta rada na transportnu traku u pokretu

Strojni vid za odlaganje predmeta na transportnu traku u pokretu realiziran je putem integriranog *FANUC* vizijskog sučelja *iRVision* i *SONY* industrijske kamere [15] [Slika 38].



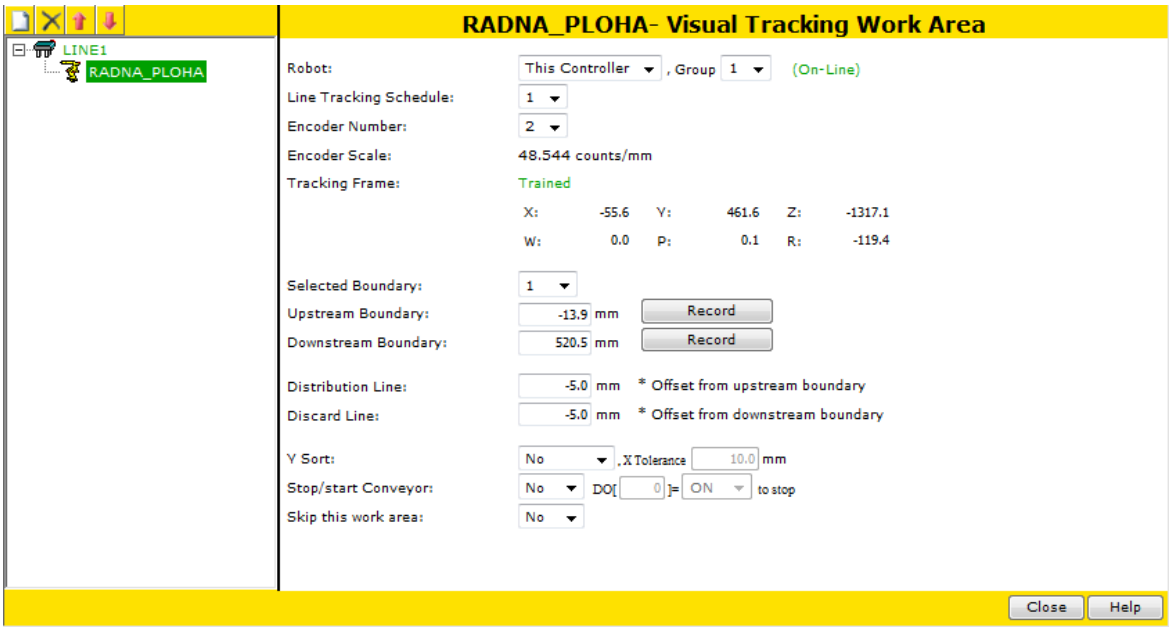

**Slika 38. iRVision sučelje.**

Prvi korak je odabir i konfiguracija kamere korištene za *iRVision* strojni vid [Slika 39].



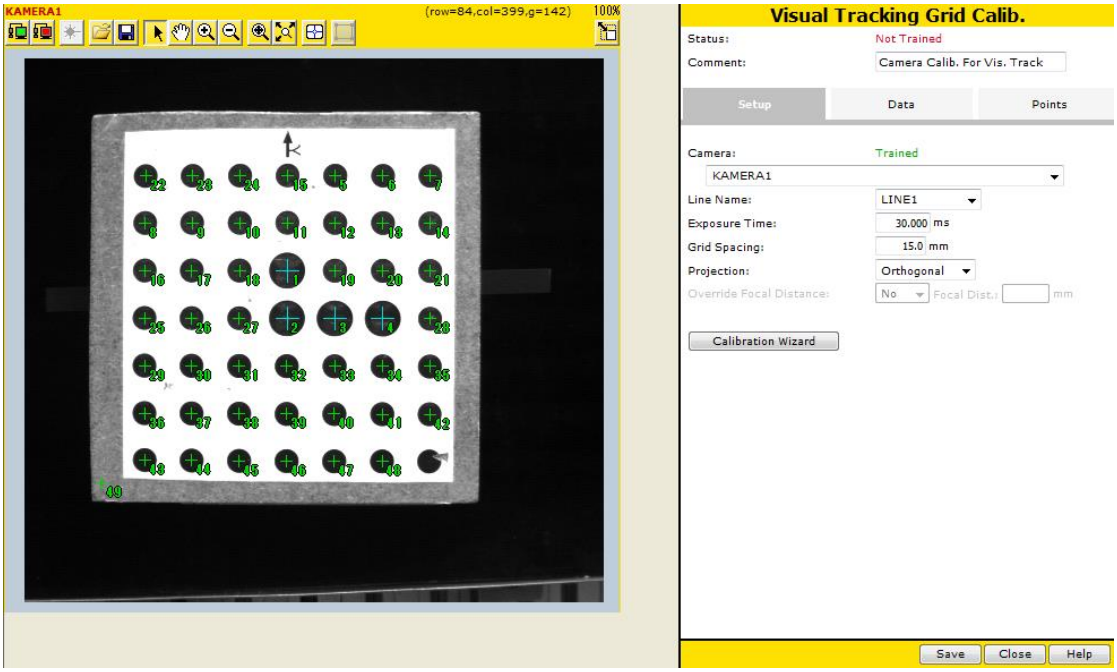

**Slika 39. Konfiguracija kamere za *iRVision* strojni vid.**

Kako se radi o praćenju predmeta na transportnoj traci, mora se definirati radni prostor, odnosno parametri transportne trake kao što su skala enkodera, granice radnog prostora robota i koordinatni sustav za praćenje predmeta [Slika 40].



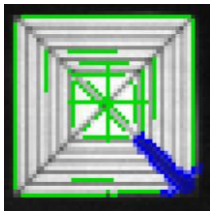

**Slika 40. Konfiguracija radnog prostora.**

Idući korak je kalibracija kamere kako bi se napravila poveznica između poznate značajke u stvarnom svijetu (udaljenost između točaka kalibracijske mreže) i koordinatnog sustava kamere [Slika 41].

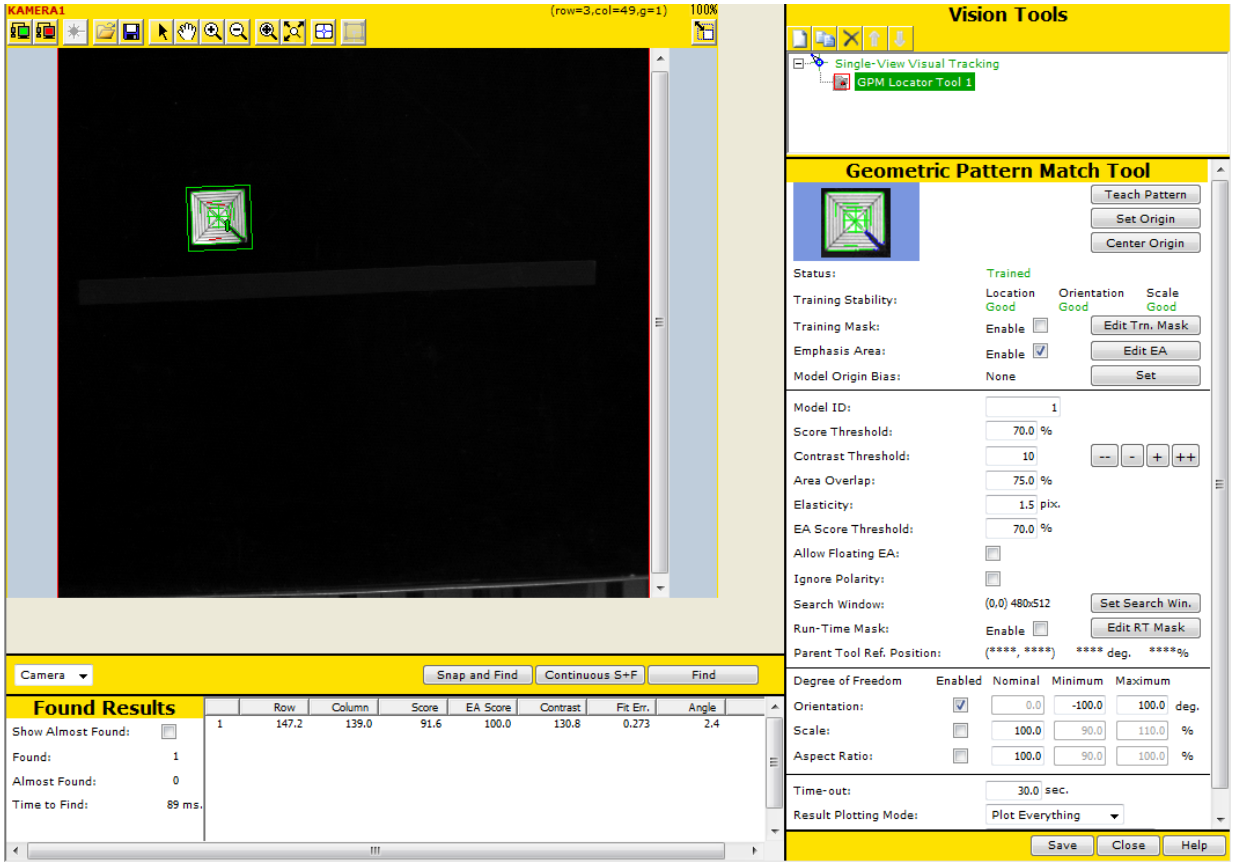



**Slika 41. Kalibracija kamere.**

Naposljetku konfigurira se vizijski proces, u ovom slučaju vizijski proces za praćenje predmeta na transportnoj traci, „Single-View Visual Tracking“. U procesu se definira alat za prepoznavanje predmeta rada. Zbog jasnih geometrijskih značajki predmeta rada (podloge za odlaganje [Slika 42]), izabran je alat za traženje istih, „Geometric Pattern Match Tool“. Predmet rada se stavlja u vidno polje kamere, okida se slika, te se definira što je točno predmet rada. Pomoću maske moguće je definirati koje su značajke više, a koje manje bitne za prepoznavanje [Slika 43]. Na predmetu je označena crnom bojom linija koja ga čini asimetričnim, te tako jasno definira orijentaciju predmeta. Definiraju se i parametri poput raspona orijentacije predmeta, stupnja podudarnosti, ekspozicije slike itd.



**Slika 42. Predmet rada sa naučenim značajkama za prepoznavanje (zeleno) i maskom koja naglašava najbitnije značajke (plavo).**




**Slika 43. Sučelje za konfiguraciju vizijskog procesa.**

Nakon što je konfiguriran alat za traženje predmeta rada, isti se pronalazi u vidnom polju kamere opcijom „Snap and Find“, kako bi se definirala referentna pozicija za rukovanje predmetom rada. Robot se dovodi u željenu poziciju koja se sprema u pozicijski registar. S time završava konfiguracija vizijskog procesa i može se pisati program u kojem će se pozivati vizijski proces i radni prostor na transportnoj traci.

### 4.3. Razvoj programa za odlaganje predmeta na transportnu traku u pokretu

Unutar druge „FOR“ petlje programa za izuzimanje predmeta rada, nakon same operacije izuzimanja, poziva se program „PR_PICK“ [Slika 44]. Inicijalizacija vizijskog procesa i radnog prostora na transportnoj traci obavlja se u sklopu glavnog programa „PR_TRACKING“ pri inicijalizaciji ostalih parametara sustava [Slika 45]. Naredbe „CALL VSTKCLRQ ('RADNA_PLOHA')“ i „CALL VSTKSTRT“ služe definiranju radnog prostora i inicijalizaciji strojnog vida. Konfigurirani vizijski proces poziva se unutar potprograma „PR_ACQ“ [Slika 46].

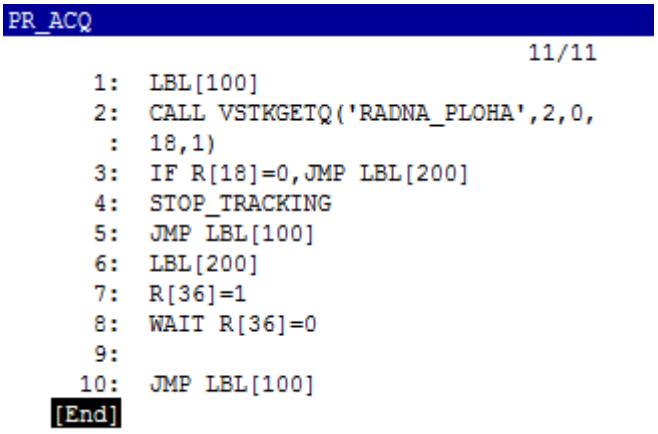
```
PR_PICK                                          1/17
   1:  UTOOL_NUM=6
   2:  UFRAME_NUM=0
   3:  WAIT R[36]=1
   4:  CALL KOMPENZACIJA
   5:  LBL[200]
   6:  LINECOUNT[2] R[23]
   7:L  PR[1] 666mm/sec CNT80
    :  VOFFSET,VR[2] Tool_Offset,PR[2]
    :  Offset,PR[8]
   8:L  PR[1] 666mm/sec CNT80
    :  VOFFSET,VR[2] Offset,PR[8]
   9:  WAIT    .50(sec)
  10:  DO[114]=OFF
  11:  DO[113]=ON
  12:L  PR[1] 666mm/sec CNT80
    :  VOFFSET,VR[2] Tool_Offset,PR[2]
    :  Offset,PR[8]
  13:  DO[112]=ON
  14:  CALL VSTKACKQ('RADNA_PLOHA',2,1)
  15:  R[36]=0
  16:  LBL[900]
```

**Slika 44. Program za odlaganje predmeta rada.**

```
  13:  CALL VSTKCLRQ('RADNA_PLOHA')
  14:  CALL VSTKSTRT
  15:  RUN PR_ACQ
```

**Slika 45. Inicijalizacija radnog prostora i strojnog vida.**

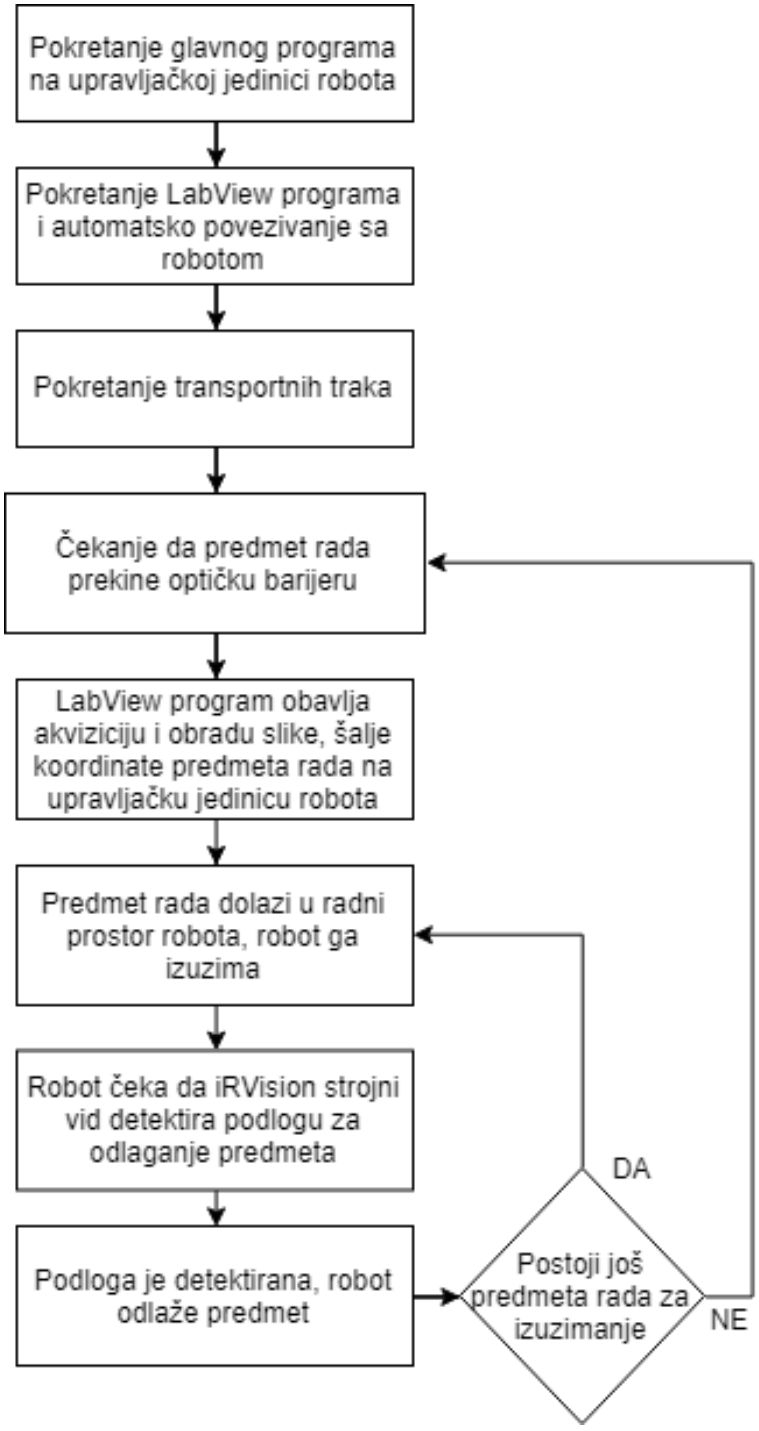
Naredbom „VSTKETGQ“, uzimaju se vizijski parametri s definiranog radnog prostora i spremaju u određeni vizijski registar, u ovom slučaju VR[2]. Naučenoj točci odlaganja predmeta pridodaje se pomak iz vizijskog registra „VOFFSET“ naredbom, te po završetku obavljene operacije program se završava i nastavlja se vrtiti „FOR“ petlja programa za izuzimanje predmeta. Po završetku obje „FOR“ petlje, svi detektirani predmeti rada su obrađeni, izvršava se ponovna inicijalizacija potrebnih registara, te se petlja vrti u beskonačnost čekajući na novu detekciju predmeta rada.



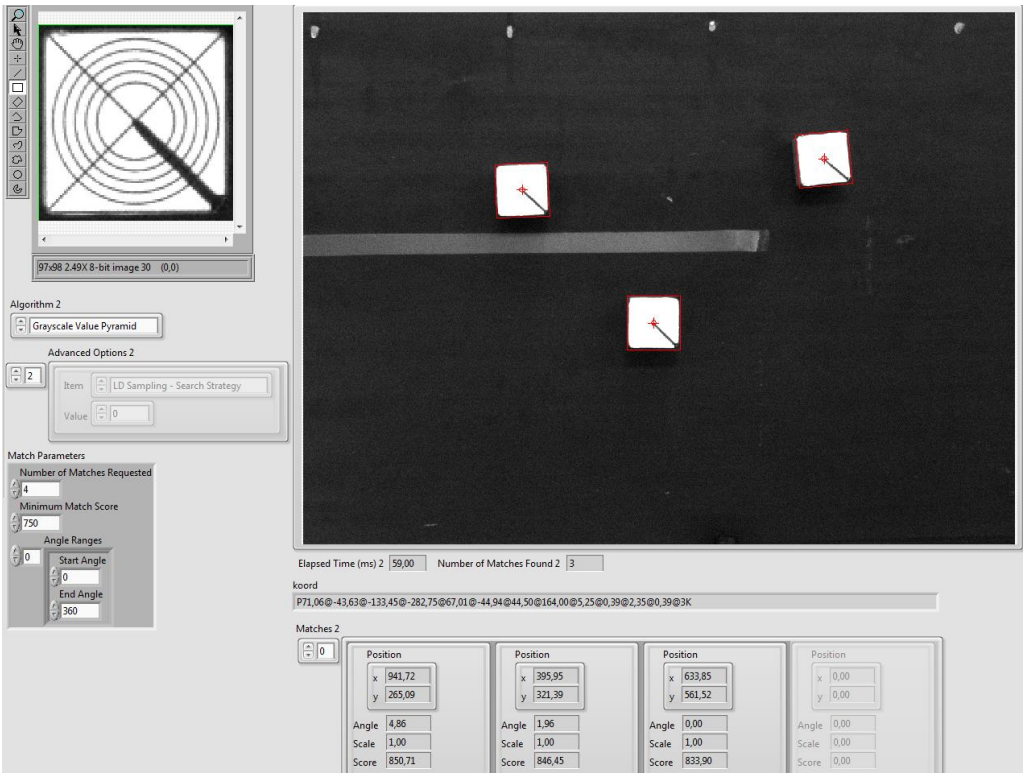

**Slika 46. Program za akviziciju slike.**

# 5. PREGLED RADA RAZVIJENOG SUSTAVA

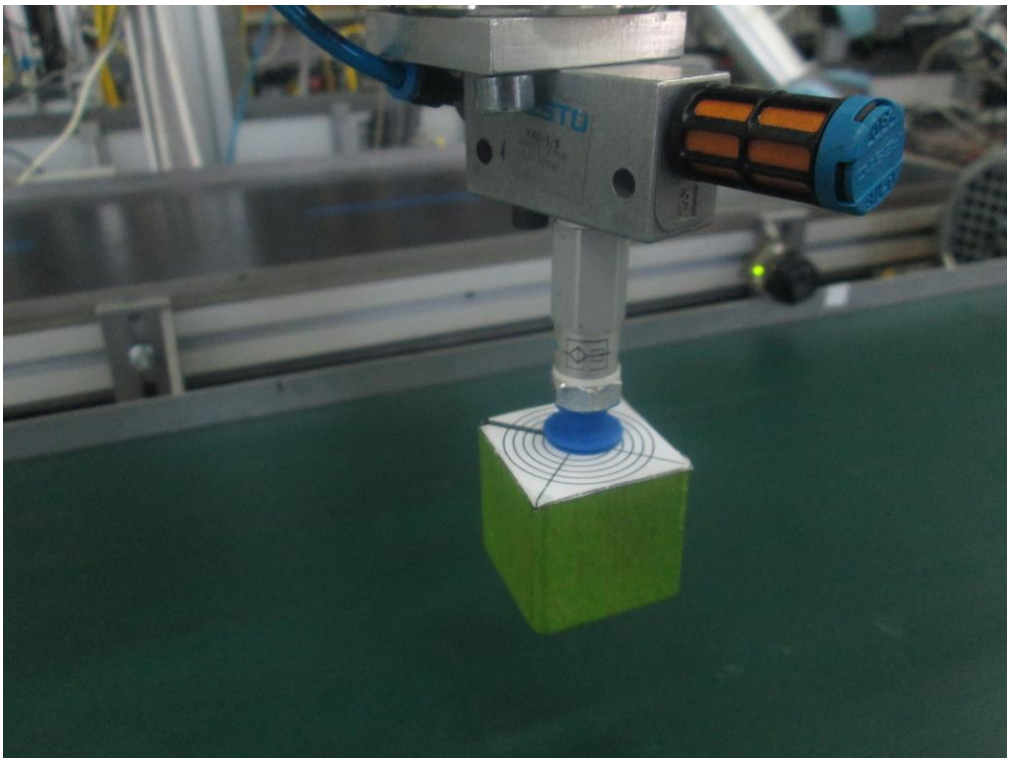
Tok informacija i operacija koje razvijeni sustav robotskog rukovanja predmetom rada u pokretu obavlja, dan je sljedećim dijagramom [Slika 47].



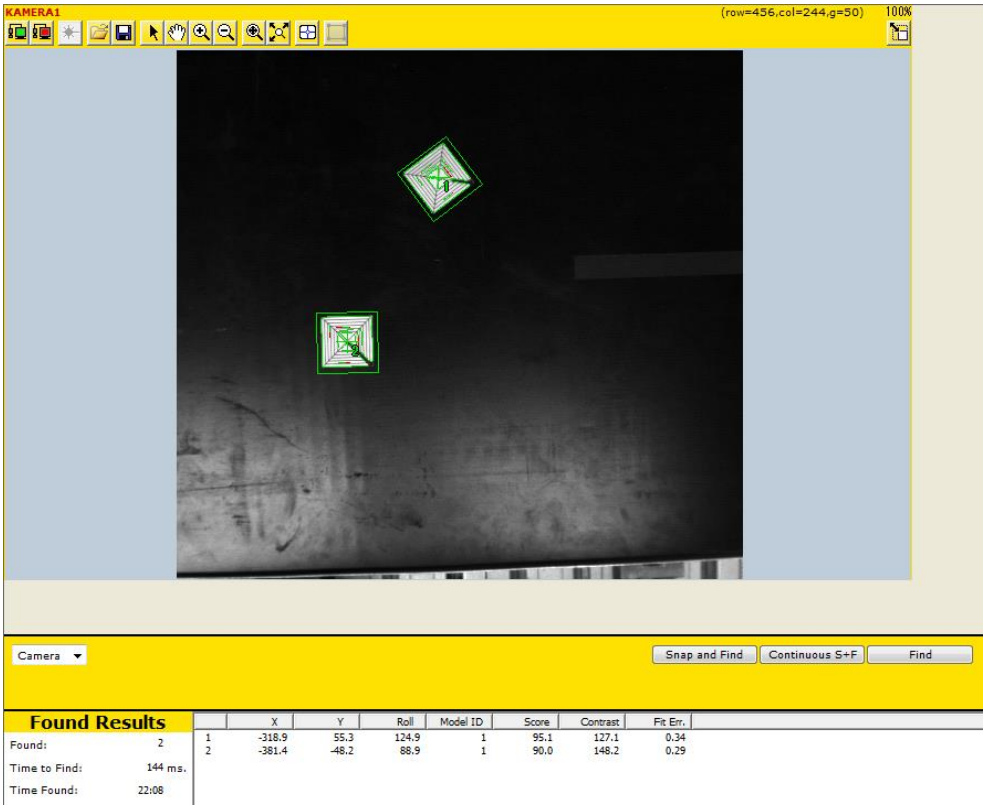

**Slika 47. Blok dijagram rada sustava.**



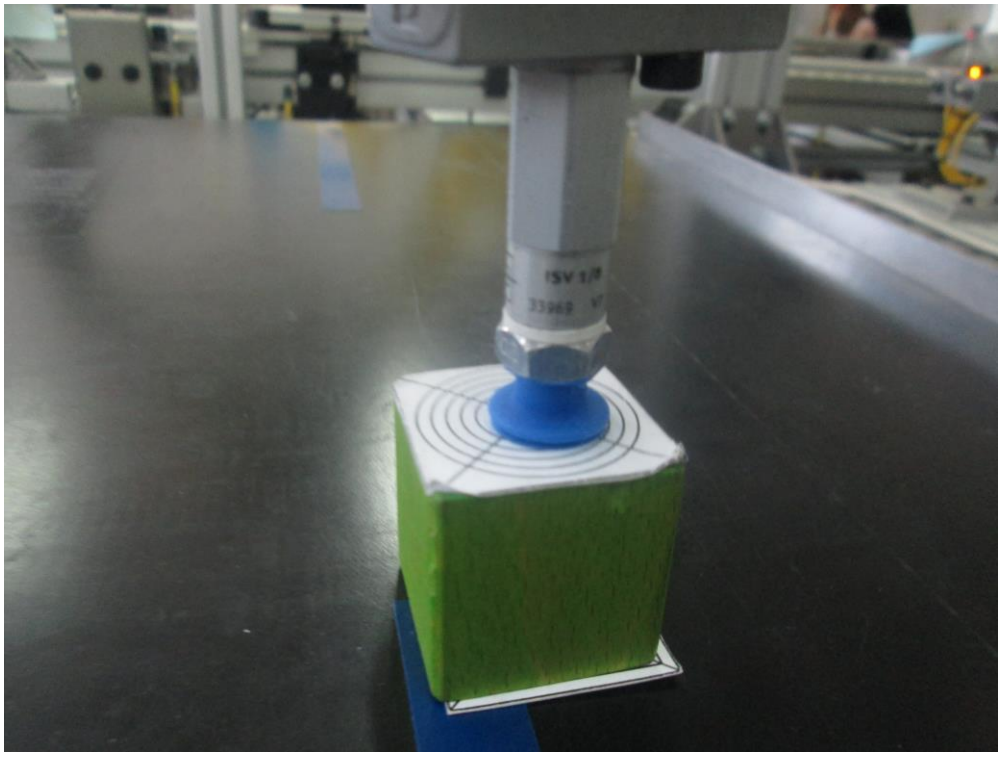

**Slika 48. Sučelje *LabView* programa za strojni vid sa nađenim predmetima rada.**



**Slika 49. Izuzimanje predmeta rada sa transportne trake.**




**Slika 50. Sučelje *iRVision* strojnog vida sa nađenim podlogama za odlaganje predmeta rada.**



**Slika 51. Odlaganje predmeta rada na transportnu traku.**

# 6. MJERENJE I ANALIZA TOČNOSTI IZUZIMANJA I ODLAGANJA PREDMETA RADA, TE KOMPENZACIJA DOBIVENE GREŠKE

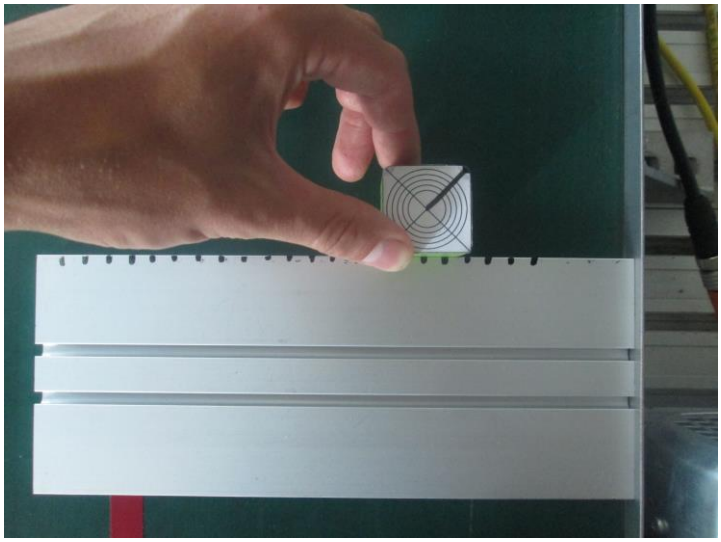
Pri puštanju sustava u rad i bez mjerenja se primijeti velika greška točnosti pri izuzimanju i odlaganju predmeta rada. Kako bi se dokazalo moguće ponavljanje greške, a samim time pokušala kompenzacija iste, potrebno je izvršiti niz mjerenja kontrolnih varijabli. Iako se u nesređenom sustavu (nasumičan položaj i orijentacija predmeta rada u ravnini) ne može jednostavno vizualno zaključiti ponovljivost greške, može se primijetiti nekoliko parametara koji znatnije utječu na istu od drugih, kao što su položaj i orijentacija predmeta na transportnoj traci prilikom akvizicije slike, te brzina transportne trake prilikom izuzimanja predmeta. Odvojeno će se promatrati proces izuzimanja i odlaganja predmeta, te će se u kontroliranim uvjetima pokušati kaskadno otkloniti komponente ukupne greške. Za kontrolne varijable izabrani su y pomak predmeta i orijentacija pri akviziciji slike, te brzina transportne trake. Očitavanje pomaka i orijentacije vrši se preko računala, odnosno privjeska za učenje prilikom akvizicije slike, dok se greška pozicioniranja robota vrši vizualno putem isprintanih tekstura na predmetima rada, razlučivosti 1 mm [Slika 52]. Takva razlučivost izabrana je zbog realnog postava sustava gdje vizualno nije moguće preciznije odrediti grešku. Preciznija mjerenja dobila bi se uvođenjem novog vizijskog sustava za mjerenje, no takvo što nije potrebno zbog relativno velikog reda greške (maksimalna greška do 20 mm) i ograničenosti radnog prostora.

**Slika 52. Predmeti rada.**

## 6.1. Mjerenje točnosti pozicioniranja robota pri izuzimanju predmeta rada sa transportne trake i kompenzacija greške

Prva kontrolna varijabla na koju se testiralo izuzimanje predmeta rada sa transportne trake je y pomak predmeta. X pomak se ne promatra, jer je jedank nuli uslijed obavljene

kompenzacije pri razvoju vizijskog sustava. Na kontroleru elektromotora transportne trake namještena je najniža moguća brzina, dok se konzistentnost orijentacije predmeta osigurala pomoću aluminijskog profila sa označenim razmacima od 10 mm, kako bi se predmet rada preciznije i lakše pozicionirao prilikom mjerenja [Slika 53].

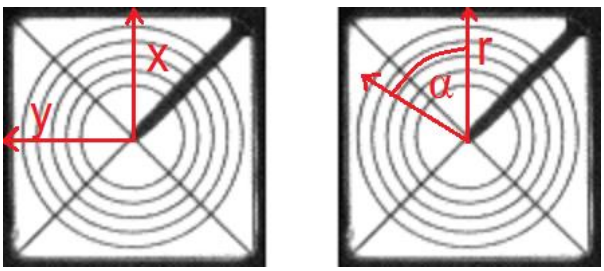


**Slika 53. Pomicanje predmeta rada po transportnoj traci uz zadržavanje orijentacije.**

Pri izuzimanju predmeta greška se očitava približno pomoću koncetričnih kružnica udaljenih po 2 mm, dok najmanja odgovara radijusu vakuum sisalje (14 mm). Očitava se radijalna greška odstupanja centra vakuum sisaljke od centra kružnica, te približan kut referentnog polarnog koordinatnog sustava na predmetu [Slika 54]. Nakon obavljenog mjerenja polarne koordinate prebačene su u kartezijske kako bi se x i y komponente greške kasnije iskoristile za kompenzaciju iste (3), (4).

$$x = r \cdot \cos(\alpha) \quad \textbf{(3)}$$

$$y = r \cdot \sin(\alpha) \quad \textbf{(4)}$$

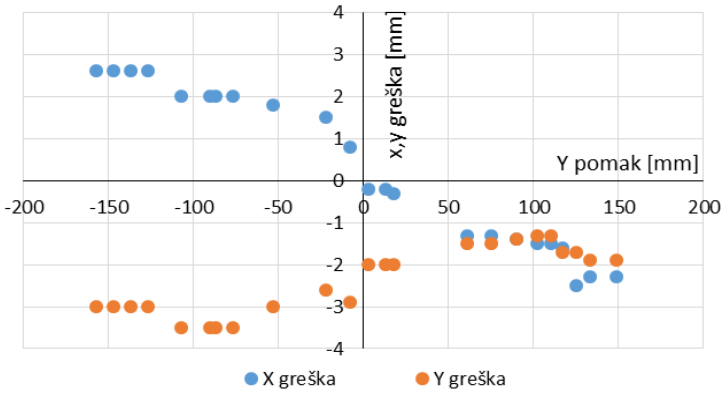



**Slika 54. Predmet rada sa kartezijskim koordinatnim sustavom (lijevo) i polarnim (desno).**

U tablici [Tablica 1] su zapisani rezultati prvog seta mjerenja kontrolne varijable y pomaka predmeta. Ponašanje greške je lakše protumačiti vizualno [Slika 55].

| | Y pomak [mm] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -157 | 2.6 | -3 |
| 2 | -147 | 2.6 | -3 |
| 3 | -137 | 2.6 | -3 |
| 4 | -127 | 2.6 | -3 |
| 5 | -107 | 2 | -3.5 |
| 6 | -90 | 2 | -3.5 |
| 7 | -87 | 2 | -3.5 |
| 8 | -77 | 2 | -3.5 |
| 9 | -53 | 1.8 | -3 |
| 10 | -22 | 1.5 | -2.6 |
| 11 | -8 | 0.8 | -2.9 |
| 12 | 3 | -0.2 | -2 |
| 13 | 13 | -0.2 | -2 |
| 14 | 18 | -0.3 | -2 |
| 15 | 61 | -1.3 | -1.5 |
| 16 | 75 | -1.3 | -1.5 |
| 17 | 90 | -1.4 | -1.4 |
| 18 | 102 | -1.5 | -1.3 |
| 19 | 110 | -1.5 | -1.3 |
| 20 | 117 | -1.6 | -1.7 |
| 21 | 125 | -2.5 | -1.7 |
| 22 | 133 | -2.3 | -1.9 |
| 23 | 149 | -2.3 | -1.9 |

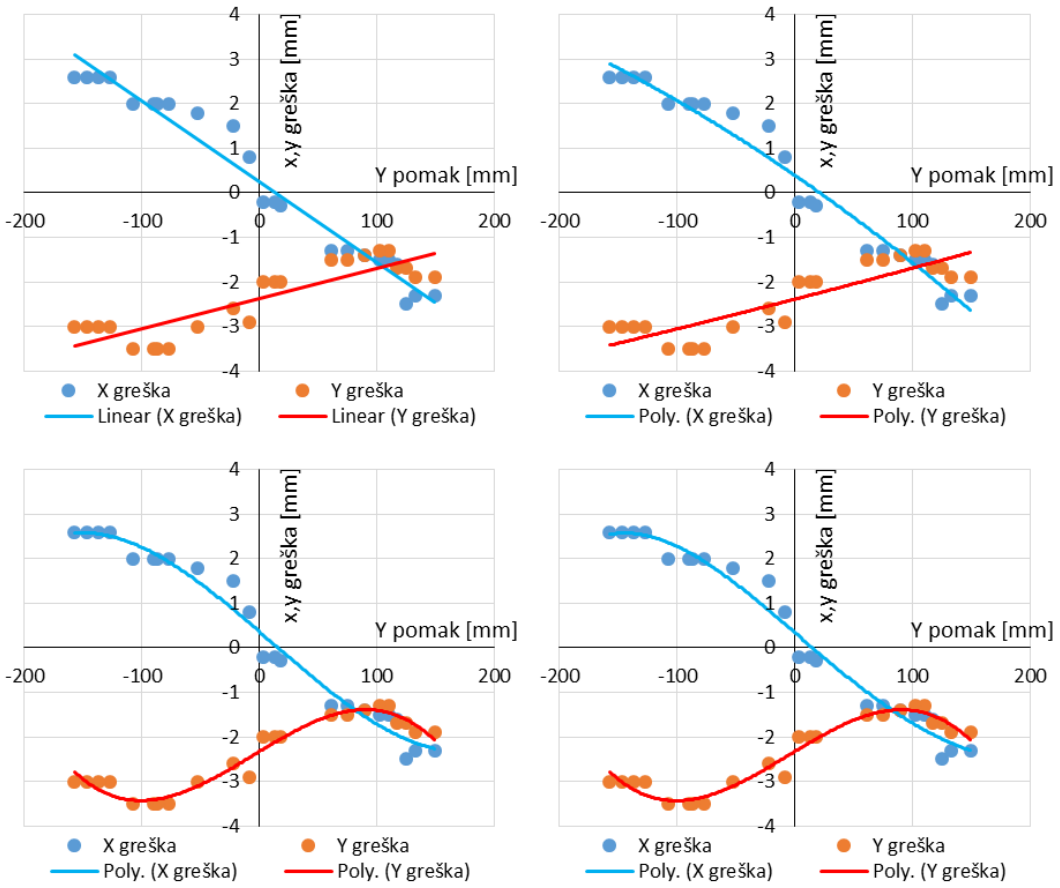
**Tablica 1. Mjerenje greške izuzimanja predmeta prilikom Y pomaka.**




**Slika 55. Grafički prikaz greške izuzimanja predmeta prilikom Y pomaka.**

Koristeći *Microsoft Office* numeričke alate za obradu seta podataka isprobavaju se različite aproksimacije dobivenog uzorka ponašanja greške. Grafičkom usporedbom aproksimacija

ponašanja x i y komponenti greške jednadžbama pravca, te polinomima drugog, trećeg i četvrtog reda dolazi se do zaključka da bi najbolja aproksimacija mogla biti ona polinomom trećeg reda, jer povišenjem stupnja polinoma ne dobiva se bolji rezultat, samo se komplicira jednadžba, samim time i računanje kompenzacije.




**Slika 56. Usporedba aproksimacija greške y pomaka: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno).**

Dobivene su jednadžbe aproksimacije x komponente greške (5) i y komponente greške (6), gdje je $ey_x$ x komponenta greške, $ey_y$ y komponenta, a $y$ mjerena kontrolna varijabla.

$$ey_x = 3 \cdot 10^{-7} \cdot y^3 - 9 \cdot 10^{-6} \cdot y^2 - 0.0228 \cdot y + 0.3636 \quad \textbf{(5)}$$

$$ey_y = -6 \cdot 10^{-7} \cdot y^3 - 9 \cdot 10^{-6} \cdot y^2 - 0.016 \cdot y + 2.3293 \quad \textbf{(6)}$$

Greška će se pokušati kompenzirati oduzimanjem vrijednosti $ey_x$ i $ey_y$ od x i y koordinata u *KAREL* programu za popunjavanje registara sa koordinatama nađenih predmeta rada [Slika 57]. Komponente greške se očitavaju u koordinatnom sustavu predmeta rada, stoga je potrebno obaviti transformaciju u koordinatni sustav robota, odnosno transportne trake. S

obzirom da se greška oduzima od neke vrijednosti u matrici transformacija nije bitna komponenta translacije, tako da se transformacija vrši preko matrice rotacije u ravnini (7), (8).

$$R = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \quad \textbf{(7)}$$

$$\begin{bmatrix} x^v \\ y^v \end{bmatrix} = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \begin{bmatrix} x_1^v \\ y_1^v \end{bmatrix} \quad \textbf{(8)}$$

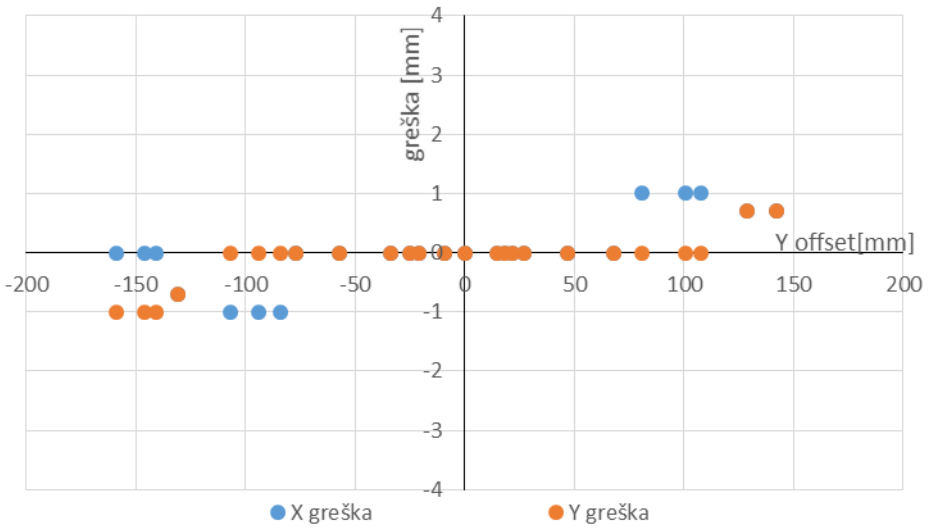
Uzevši u obzir funkciju aproksimacije koja jako dobro opisuje ponašanje greške, te provjerom iznosa kompenzacije na ekranu privjeska za učenje, dalo bi se zaključiti da će greška biti potpuno anulirana, što se u ovom slučaju skoro i postiglo. Rezultati kontrolnog mjerenja vide se u tablici [Tablica 2] i na slici [Slika 58].

| | Y pomak [mm] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -159 | 0 | -1 |
| 2 | -146 | 0 | -1 |
| 3 | -141 | 0 | -1 |
| 4 | -131 | -0.7 | -0.7 |
| 5 | -107 | -1 | 0 |
| 6 | -94 | -1 | 0 |
| 7 | -84 | -1 | 0 |
| 8 | -77 | 0 | 0 |
| 9 | -57 | 0 | 0 |
| 10 | -34 | 0 | 0 |
| 11 | -25 | 0 | 0 |
| 12 | -21 | 0 | 0 |
| 13 | -9 | 0 | 0 |
| 14 | 0 | 0 | 0 |
| 15 | 15 | 0 | 0 |
| 16 | 18 | 0 | 0 |
| 17 | 22 | 0 | 0 |
| 18 | 27 | 0 | 0 |
| 19 | 47 | 0 | 0 |
| 20 | 68 | 0 | 0 |
| 21 | 81 | 1 | 0 |
| 22 | 101 | 1 | 0 |
| 23 | 108 | 1 | 0 |
| 24 | 129 | 0.7 | 0.7 |
| 25 | 142 | 0.7 | 0.7 |

**Tablica 2. Kontrolno mjerenje kompenzacije greške uslijed y pomaka predmeta rada.**

```
23          --KOMPENZACIJA GRESKE Y OFFSETA PREDMETA
24          ----------------------------------------------------------------
25      FOR i=5 TO 8 DO
26        GET_REG(i,real_flag,int_value,yy,STATUS);
27        dxy1=3E-7*yy*yy*yy-9E-6*yy*yy-0.0228*yy+0.3636;
28        dyy1=-6E-7*yy*yy*yy-9E-6*yy*yy+0.016*yy-2.3293;
29        GET_REG((i+4),real_flag,int_value,a,STATUS);
30        dxy=COS(a)*dxy1-SIN(a)*dyy1-1;
31        dyy=SIN(a)*dxy1+COS(a)*dyy1;
32        GET_REG((i-4),real_flag,int_value,xx,STATUS);
33        --IF i=5 THEN
34            --WRITE(dxy,CR);
35            --WRITE(dyy,CR);
36        --ENDIF
37        SET_REAL_REG((i+36),dxy,STATUS);
38        SET_REAL_REG(i,yy-dyy,STATUS);
39      ENDFOR
```

**Slika 57. Kompenzacija greške pozicioniranja nastale uslijed y pomaka predmeta rada.**

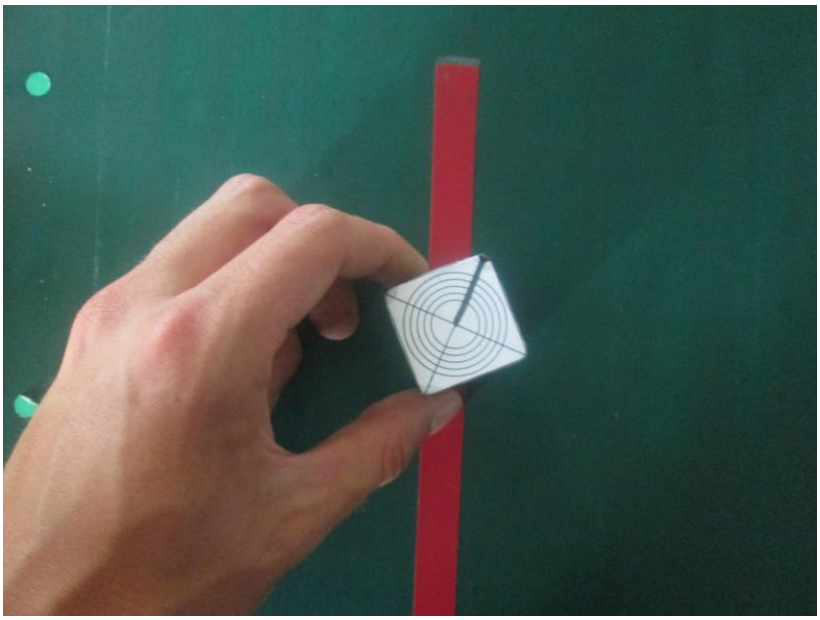



**Slika 58. Grafički prikaz kontrole kompenzacije greške uslijed y pomaka predmeta rada.**

Maksimalna radijalna greška od 4 mm smanjena je na 1.7 mm, dok je u rasponu y koordinate predmeta na pokretnoj traci od -77 mm do 68 mm greška poništena.

Sljedeći korak je nadograditi dobivenu kompenzaciju greške sa kompenzacijom greške ovisnoj o orijentaciji predmeta rada. S obzirom da je greška anulirana na sredini pokretne trake, na proizvoljnoj udaljenosti lijepi se traka u boji kako bi postupak orijentiranja predmeta bio što točniji, a mjerenje lakše izvodilo [Slika 59].

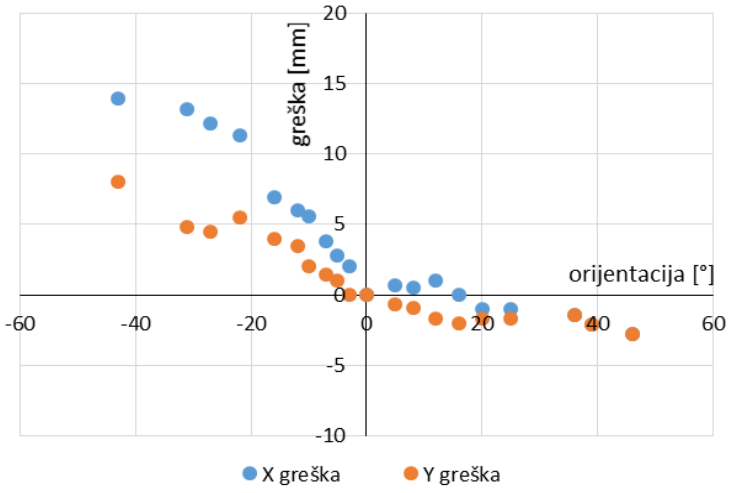
Zbog velike greške koja se javlja pri zakretanju robota oko svoje z osi, iz praktičnih razloga, orijentiranje predmeta ograničeno je na raspon od -45 do 45 stupnjeva. Prilikom postupka mjerenja te granice variraju +/-5 stupnjeva. Rezultati mjerenja greške pozicioniranja robota uslijed orijentacije predmeta prikazani su na sljedećoj tablici [Tablica 3] i slici [Slika 60].



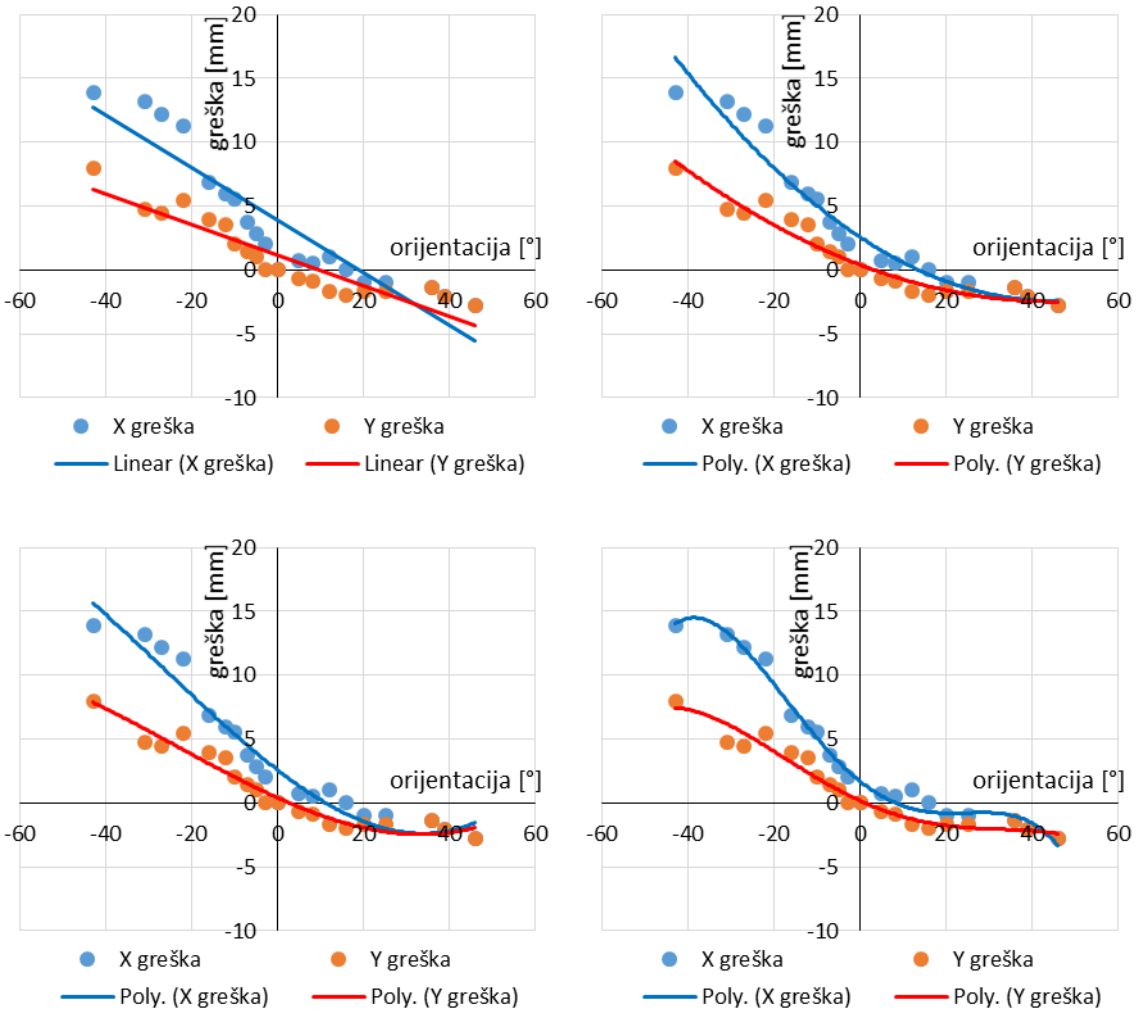
**Slika 59. Orijentiranje predmeta rada na transportnoj traci uz zadržavanje referentne pozicije.**

| | α orijentacija [°] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -43 | 13.9 | 8 |
| 2 | -31 | 13.2 | 4.8 |
| 3 | -27 | 12.2 | 4.5 |
| 4 | -22 | 11.3 | 5.5 |
| 5 | -16 | 6.9 | 4 |
| 6 | -12 | 6 | 3.5 |
| 7 | -10 | 5.6 | 2 |
| 8 | -7 | 3.8 | 1.4 |
| 9 | -5 | 2.8 | 1 |
| 10 | -3 | 2 | 0 |
| 11 | 0 | 0 | 0 |
| 12 | 5 | 0.7 | -0.7 |
| 13 | 8 | 0.5 | -0.9 |
| 14 | 12 | 1 | -1.7 |
| 15 | 16 | 0 | -2 |
| 16 | 20 | -1 | -1.7 |
| 17 | 25 | -1 | -1.7 |
| 18 | 36 | -1.4 | -1.4 |
| 19 | 39 | -2.1 | -2.1 |
| 20 | 46 | -2.8 | -2.8 |

**Tablica 3. Mjerenje greške izuzimanja predmeta uslijed orijentacije predmeta.**




**Slika 60. Grafički prikaz greške prilikom orijentiranja predmeta.**




**Slika 61. Usporedba aproksimacija greške orijentacije: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno).**

Iz dobivenih rezultata očito je da se greška prilikom orijentiranja predmeta eksponencijalno povećava kada se robot zakreće u negativnom smjeru. Maksimalna greška je mnogo veća nego u prethodnom slučaju, te se opet razmatra nekoliko mogućnosti kompenzacije [Slika 61]. Sudeći prema grafičkom prikazu izabiru se polinomi četvrtog reda koji daju jednadžbe (9) i (10).

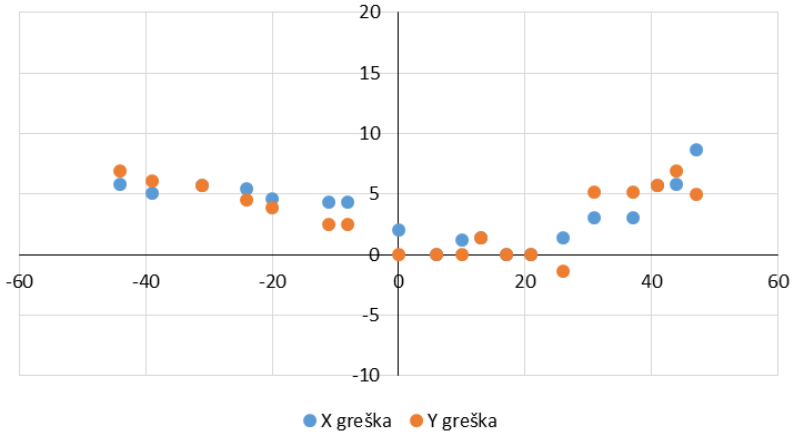
$$e\alpha_x = -3 \cdot 10^{-6} \cdot \alpha^4 + 4 \cdot 10^{-5} \cdot \alpha^3 + 0.0075 \cdot \alpha^2 - 0.2694 \cdot \alpha + 1.6928 \quad \textbf{(9)}$$

$$e\alpha_y = -7 \cdot 10^{-7} \cdot \alpha^4 + 2 \cdot 10^{-5} \cdot \alpha^3 + 0.0027 \cdot \alpha^2 - 0.1537 \cdot \alpha + 0.1756 \quad \textbf{(10)}$$

| | α orijentacija [°] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -44 | 5.8 | 6.9 |
| 2 | -39 | 5.1 | 6.1 |
| 3 | -31 | 5.7 | 5.7 |
| 4 | -24 | 5.4 | 4.5 |
| 5 | .-20 | 4.6 | 3.9 |
| 6 | -11 | 4.3 | 2.5 |
| 7 | -8 | 4.3 | 2.5 |
| 8 | 0 | 2 | 0 |
| 9 | 6 | 0 | 0 |
| 10 | 10 | 1.2 | 0 |
| 11 | 13 | 1.4 | 1.4 |
| 12 | 17 | 0 | 0 |
| 13 | 21 | 0 | 0 |
| 14 | 26 | 1.4 | -1.4 |
| 15 | 31 | 3 | 5.2 |
| 16 | 37 | 3 | 5.2 |
| 17 | 41 | 5.7 | 5.7 |
| 18 | 44 | 5.8 | 6.9 |
| 19 | 47 | 8.7 | 5 |

**Tablica 4. Kontrolno mjerenje kompenzacije greške uslijed orijentacije predmeta rada.**

Dobiveni rezultati nisu prema očekivanjima, maksimalna radijalna greška smanjila se sa 16 mm na 10 mm, ali se povećala na strani pozitivne orijentacije predmeta, gdje je prvenstveno bila niža, te je dobiven relativno mali raspon orijentacije predmeta gdje je radijalna greška manja od 2 mm. Uzrok neočekivanim rezultatima može biti akumulirana greška prilikom očitavanja, ali to je faktor koji ne smije imati znatan utjecaj na konačni rezultat. Biraju se nove jednadžbe, polinom drugog reda za aproksimaciju x komponente greške i polinom trećeg reda za aproksimaciju y komponente, tako da se dobivaju jednadžbe (11) i (12).



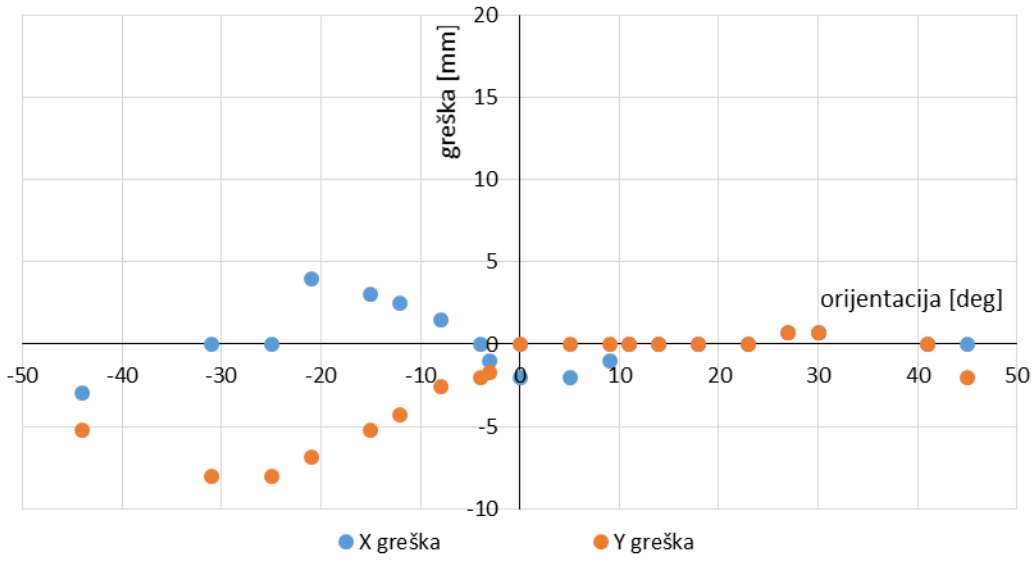

**Slika 62. Grafički prikaz kontrole kompenzacije greške pozicioniranja uslijed orijentacije predmeta rada.**

$$e\alpha_x = 0.0025 \cdot \alpha^2 - 0.2216 \cdot \alpha + 2.5613 \quad \textbf{(11)}$$

$$e\alpha_y = 2 \cdot 10^{-5} \cdot \alpha^3 + 0.0013 \cdot \alpha^2 - 0.1512 \cdot \alpha + 0.4056 \quad \textbf{(12)}$$

| | α orijentacija [°] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -44 | -3 | -5.2 |
| 2 | -31 | 0 | -8 |
| 3 | -25 | 0 | -8 |
| 4 | -21 | 4 | -6.9 |
| 5 | -15 | 3 | -5.2 |
| 6 | -12 | 2.5 | -4.3 |
| 7 | -8 | 1.5 | -2.6 |
| 8 | -4 | 0 | -2 |
| 9 | -3 | -1 | -1.7 |
| 10 | 0 | -2 | 0 |
| 11 | 5 | -2 | 0 |
| 12 | 9 | -1 | 0 |
| 13 | 11 | 0 | 0 |
| 14 | 14 | 0 | 0 |
| 15 | 18 | 0 | 0 |
| 16 | 23 | 0 | 0 |
| 17 | 27 | 0.7 | 0.7 |
| 18 | 30 | 0.7 | 0.7 |
| 19 | 41 | 0 | 0 |
| 20 | 45 | 0 | -2 |

**Tablica 5. Kontrolno mjerenje nakon ponovljene kompenzacije greške pozicioniranja uslijed orijentacije predmeta rada.**




**Slika 63. Grafički prikaz kontrolnog mjerenja ponovljene kompenzacije greške pozicioniranja uslijed orijentacije predmeta rada.**

Nakon snižavanja reda polinoma aproksimacije greške, dobivaju se bolji rezultati. Maksimalna radijalna greška sa 16 mm snižena je na 6 mm, te je povećan raspon orijentacije predmeta sa minimalnom greškom (0 do 45 stupnjeva).

```
40          ----KOMPENZACIJA GRESKE ORIJENTACIJE PREDMETA
41          ------------------------------------------------------------
42    FOR i=9 TO 12 DO
43      GET_REG(i,real_flag,int_value,a,STATUS);
44      dxa1=0.0025*a*a-0.2216*a+2.5613;
45      dya1=2E-5*a*a*a+0.0013*a*a-0.1512*a+0.4056;
46      dxa=(COS(a)*dxa1-SIN(a)*dya1)*0.85;
47      dya=SIN(a)*dxa1+COS(a)*dya1;
48      GET_REG((i+32),real_flag,int_value,kx,STATUS);
49      GET_REG((i-4),real_flag,int_value,yy,STATUS);
50      --IF i=9 THEN
51          --WRITE(dxa,CR);
52          --WRITE(dya,CR);
53      --ENDIF
54      SET_REAL_REG((i+32),kx+dxa,STATUS);
55      SET_REAL_REG((i-4),yy-dya,STATUS);
56    ENDFOR
```

**Slika 64. Kompenzacija greške pozicioniranja uslijed orijentacije predmeta rada.**

Naposlijetku, mjeri se greška izuzimanja predmeta pri različitim brzinama transportne trake. Brzina transportne trake podešava se potenciometrom na upravljačkoj jedinici elektromotora, frekvencijskim upravljanjem. Jedinica u kojoj se promatra brzina u ovom slučaju je broj okretaja u minuti koji se dobiva iz izraza za obodnu brzinu (13), gdje je *v* obodna brzina, a *r* polumjer bubnja pogonskog bubnja transportne trake. Pomoću „GET_VAR“ *KAREL* funkcije

dobiva se vrijednost skale zadanog enkodera i trenutačna vrijednost istog. Vrijednost se uzima u razmaku od jedne sekunde kako bi se iz izraza (14) dobila obodna brzina, koja se zatim uvrštava u izraz (13). Pri tome su *cnt2* i *cnt* trenutačne vrijednosti enkodera, a *scl* skala istog.

$$n = \frac{v \cdot 30}{r \cdot \pi} \ [min^{-1}] \quad \textbf{(13)}$$

$$v = \frac{cnt2 - cnt}{scl \cdot 1s} \left[\frac{mm}{s}\right] \quad \textbf{(14)}$$

Pri inicijalizaciji glavnog programa, nakon pokretanja transportne trake poziva se potprogram „ENKODER“ koji očitava broj okretaja u minuti elektromotora [Slika 65]. Prije pokretanja potprograma, glavni program čeka 1.5 sekunde prije očitavanja kako bi transportna traka pri većim brzinama imala dovoljno vremena da ubrza do željene brzine.

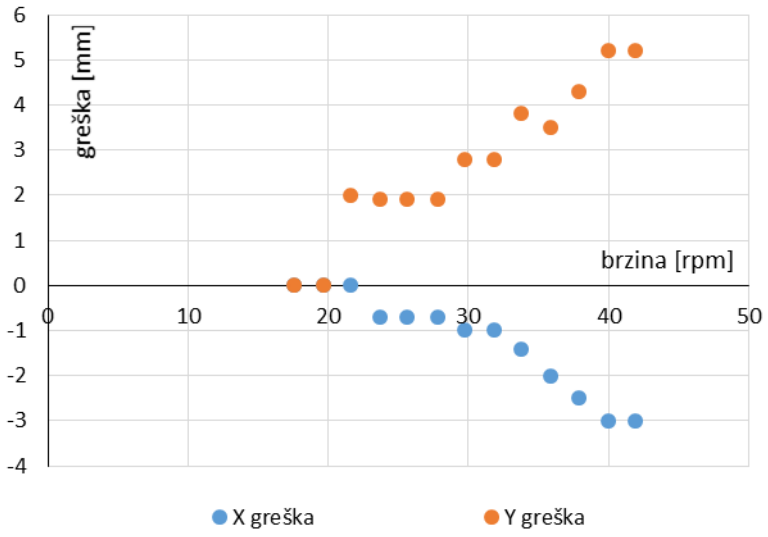
```
11 BEGIN
12     GET_VAR(entry,'*SYSTEM*','$LNSCH[2].$scale',scl,STATUS);
13     GET_VAR(entry,'*SYSTEM*','$ENC_STAT[1].$enc_count',cnt,STATUS);
14     DELAY 1000;
15     GET_VAR(entry,'*SYSTEM*','$ENC_STAT[1].$enc_count',cnt2,STATUS);
16     SET_REAL_REG(40,(cnt2-cnt)/(scl*25*0.10472),STATUS);
17 END ENKODER
```

**Slika 65. Program za očitavanje okretaja u minuti elektromotora transportne trake.**

Predmet rada pozicioniran je i orijentiran tako da ima najmanju moguću grešku izuzimanja sa do sad implementiranim kompenzacijama.

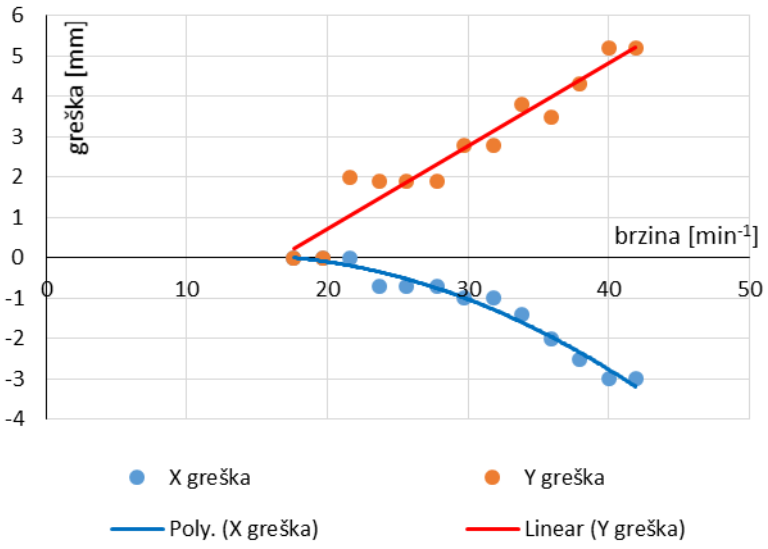
| | okretaji u minuti [$min^{-1}$] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | 17.6 | 0 | 0 |
| 2 | 19.7 | 0 | 0 |
| 3 | 21.6 | 0 | 2 |
| 4 | 23.7 | -0.7 | 1.9 |
| 5 | 25.6 | -0.7 | 1.9 |
| 6 | 27.8 | -0.7 | 1.9 |
| 7 | 29.7 | -1 | 2.8 |
| 8 | 31.8 | -1 | 2.8 |
| 9 | 33.8 | -1.4 | 3.8 |
| 10 | 35.9 | -2 | 3.5 |
| 11 | 37.9 | -2.5 | 4.3 |
| 12 | 40 | -3 | 5.2 |
| 13 | 41.9 | -3 | 5.2 |

**Tablica 6. Mjerenje greške izuzimanja uslijed brzine transportne trake.**




**Slika 66. Grafički prikaz greške pozicioniranja uslijed brzine transportne trake.**

Rezultati mjerenja pokazuju da bi se greška uslijed brzine transportne trake mogla aproksimirati jednadžbom pravca za y komponentu i polinomom drugog reda za x komponentu, što se vidi i na grafu [Slika 67].




**Slika 67. Aproksimacija y komponente greške jednadžbom pravca (crveno) i x komponente greške polinomom drugog reda (plavo).**

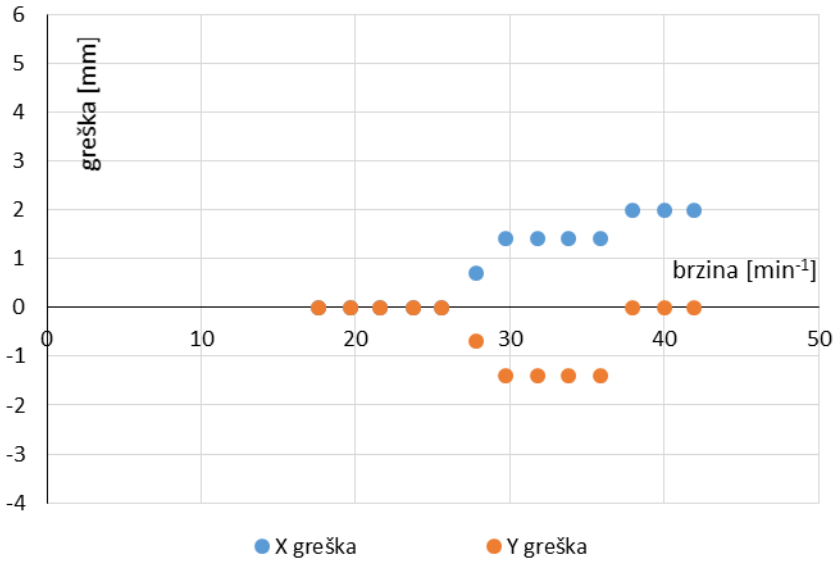
Jednadžbe korištene za kompenzaciju greške uslijed brzine transportne trake (15) i (16).

$$en_x = -0.0041 \cdot n^2 + 0.11 \cdot n - 0.6766 \quad \textbf{(15)}$$

$$en_y = 0.2052 \cdot n - 3.3946 \quad \textbf{(16)}$$

| | okretaji u minuti [$min^{-1}$] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | 17.6 | 0 | 0 |
| 2 | 19.7 | 0 | 0 |
| 3 | 21.6 | 0 | 0 |
| 4 | 23.7 | 0 | 0 |
| 5 | 25.6 | 0 | 0 |
| 6 | 27.8 | 0.7 | -0.7 |
| 7 | 29.7 | 1.4 | -1.4 |
| 8 | 31.8 | 1.4 | -1.4 |
| 9 | 33.8 | 1.4 | -1.4 |
| 10 | 35.9 | 1.4 | -1.4 |
| 11 | 37.9 | 2 | 0 |
| 12 | 40 | 2 | 0 |
| 13 | 41.9 | 2 | 0 |

**Tablica 7. Kontrola kompenzacije greške pozicioniranja uslijed brzine transportne trake.**




**Slika 68. Grafički prikaz kontrole kompenzacije greške pozicioniranja uslijed brzine transportne trake.**

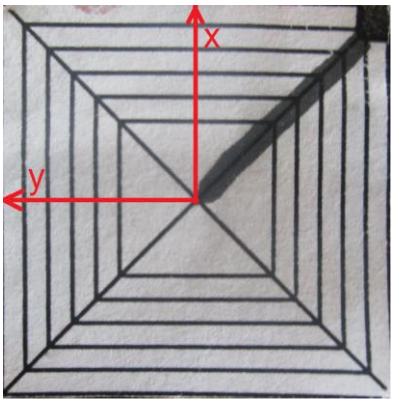
Sa tablice [Tablica 7] i grafa [Slika 68] jasno se vidi kako je kompenzacija smanjila maksimalnu radijalnu grešku sa 6 mm na 2 mm, te dok se prije kompenzacije greška javlja pri brzini od 21.6 $min^{-1}$, nakon primjene kompenzacije javlja se pri brzini od 27.8 $min^{-1}$.

```
57  ----KOMPENZACIJA BRZINE KONVEJERA
58          ------------------------------------------------------------------
59  FOR i=9 TO 12 DO
60    GET_REG(i,real_flag,int_value,a,STATUS);
61    GET_REG(40,real_flag,int_value,v,STATUS);
62    dxv1=-0.0041*v*v+0.11*v-0.6766;
63    dyv1=0.2052*v-3.3946;
64    dxv=COS(a)*dxv1-SIN(a)*dyv1;
65    dyv=SIN(a)*dxv1+COS(a)*dyv1;
66    GET_REG((i+32),real_flag,int_value,kx,STATUS);
67    GET_REG((i-4),real_flag,int_value,yy,STATUS);
68    --IF i=9 THEN
69          --WRITE(dxv,CR);
70          --WRITE(xx,CR);
71    --ENDIF
72    --IF i=9 THEN
73       --GET_REG(1,real_flag,int_value,xx,STATUS);
74       --WRITE(xx,CR);
75    --ENDIF
76    SET_REAL_REG((i+32),kx+dxv,STATUS);
77    SET_REAL_REG((i-4),yy-dyv,STATUS);
78  ENDFOR
```

**Slika 69. Kompenzacija greške pozicioniranja uslijed brzine transportne trake.**

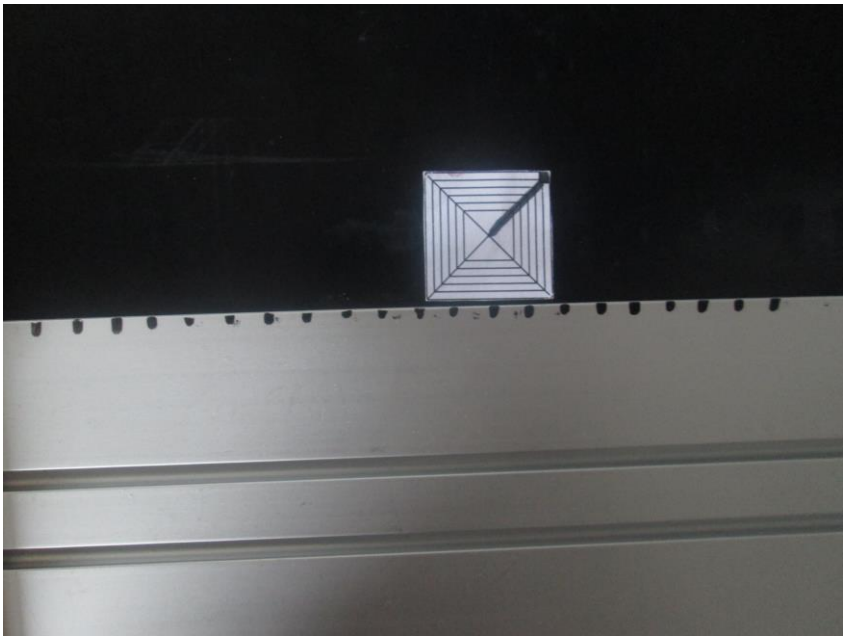
## 6.2. Mjerenje točnosti pozicioniranja robota pri odlaganju predmeta rada na transportnu traku i kompenzacija greške

S obzirom da je jedna od komponenti greške odlaganja ukupna greška izuzimanja predmeta, kako bi rezultati mjerenja greške odlaganja predmeta bili što vjerodostojniji, prvi korak je obaviti izuzimanje predmeta sa greškom približnoj nuli. Nakon toga mjerenja se obavljaju po kontrolnim varijablama na isti način kao kod izuzimanja predmeta. Razlika je jedino u tome što se pri odlaganju predmeta rada greška očitava u kartezijevom koordinatnom sustavu zbog kvadratnog profila predmeta [Slika 70].




**Slika 70. Podloga za odlaganje predmeta rada sa ucrtanim kartezijevim koordinatnim sustavom.**

Prva kontrolna varijabla je y pomak u odnosu na transportnu traku prilikom akvizicije slike. Konzistentnost orijentacije podloge odlaganja predmeta rada ostvarena je na isti način kao u prethodnom slučaju [Slika 71]. Primjećeno je da je greška odlaganja mnogo manja u odnosu na onu izuzimanja predmeta rada, pa je samim time potreban i manji broj mjerenja.

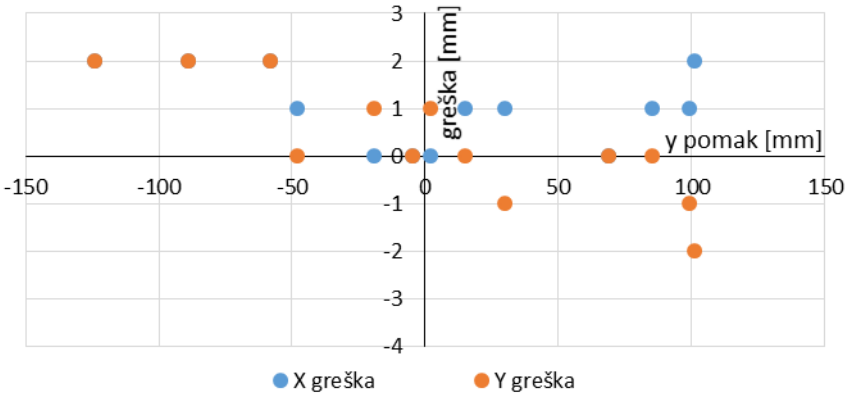


**Slika 71. Osiguravanje orijentacije prilikom mjerenja y pomaka podloge za odlaganje predmeta rada.**

Rezultati prvog mjerenja su vidljivi u tablici [Tablica 8] i na grafu [Slika 72].

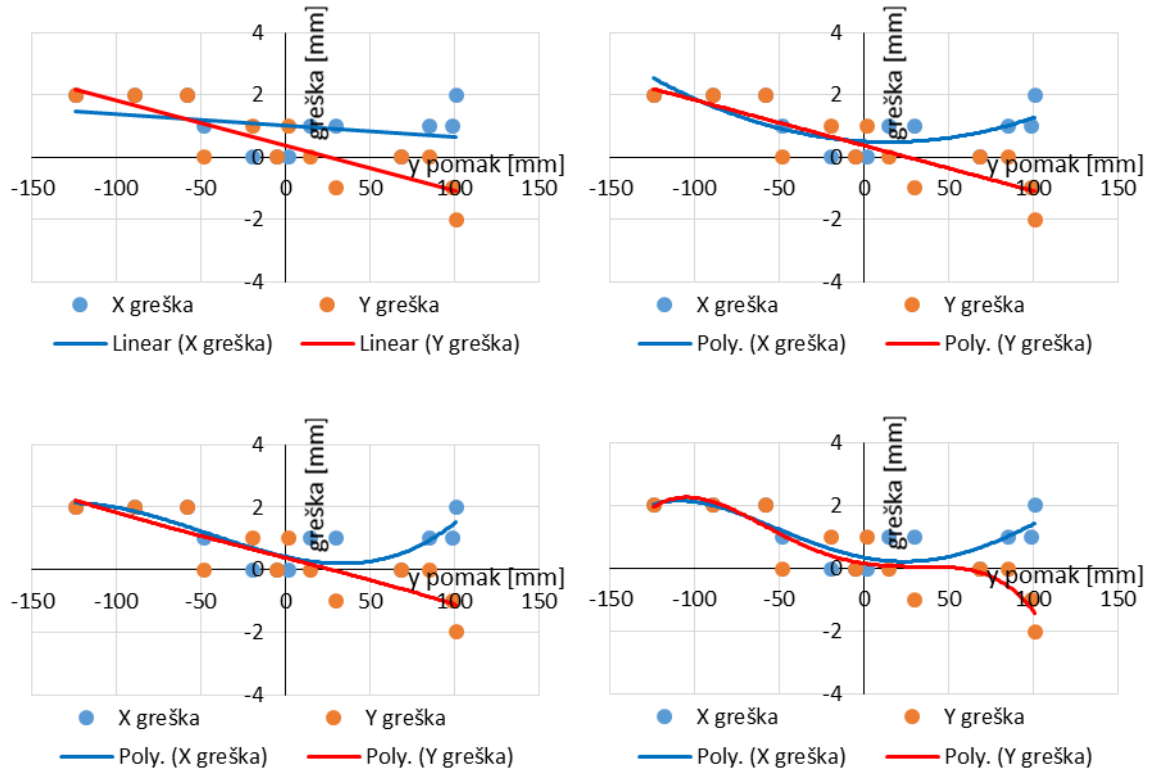
| | Y pomak [mm] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -124 | 2 | 2 |
| 2 | -89 | 2 | 2 |
| 3 | -58 | 2 | 2 |
| 4 | -48 | 1 | 0 |
| 5 | -19 | 0 | 1 |
| 6 | -5 | 0 | 0 |
| 7 | 2 | 0 | 1 |
| 8 | 15 | 1 | 0 |
| 9 | 30 | 1 | -1 |
| 10 | 69 | 0 | 0 |
| 11 | 85 | 1 | 0 |
| 12 | 99 | 1 | -1 |
| 13 | 101 | 2 | -2 |

**Tablica 8. Mjerenje greške odlaganja predmeta uslijed y pomaka predmeta na transportnoj traci.**




**Slika 72. Grafički prikaz greške odlaganja predmeta uslijed y pomaka na transportnoj traci.**

Iz rezultata je vidljivo da je maksimalna radijalna greška 2 mm, što je mnogo manje i od prosječne radijalne greške izuzimanja predmeta. No, u ovom malom rasponu ne pojavljuje se očito pravilo ponašanja greške. Da bi se isto našlo u malom rasponu amplitude greške, potrebno je povećati razlučivost očitavanja greške, što nije moguće u ovom slučaju zbog vizualnog načina očitavanja. Svejedno, probat će se primijeniti kompenzacija nekom od jednadžbi [Slika 73].




**Slika 73. Usporedba aproksimacija greške y pomaka: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno).**

S obzirom na malu ukupnu grešku, za aproksimaciju x i y komponenti greške odlaganje predmeta biraju se jednadžbe pravca (17) i (18).

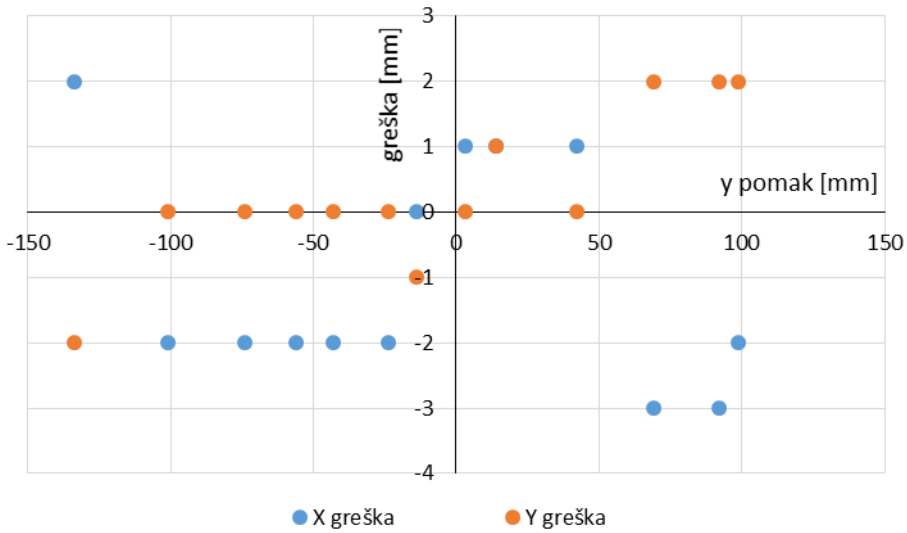
$$ey_x = -0.0035 \cdot y + 1.0156 \quad \textbf{(17)}$$

$$ey_y = -0.0146 \cdot y + 0.3728 \quad \textbf{(18)}$$

Nakon oduzimanja dobivenih kompenzacija od greške ponovo se provodi mjerenje, vidljivo u tablici [Tablica 9] i grafu [Slika 74].

| | Y pomak [mm] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -134 | 2 | -2 |
| 2 | -101 | -2 | 0 |
| 3 | -74 | -2 | 0 |
| 4 | -56 | -2 | 0 |
| 5 | -43 | -2 | 0 |
| 6 | -24 | -2 | 0 |
| 7 | -14 | 0 | -1 |
| 8 | 3 | 1 | 0 |
| 9 | 14 | 1 | 1 |
| 10 | 42 | 1 | 0 |
| 11 | 69 | -3 | 2 |
| 12 | 92 | -3 | 2 |
| 13 | 99 | -2 | 2 |

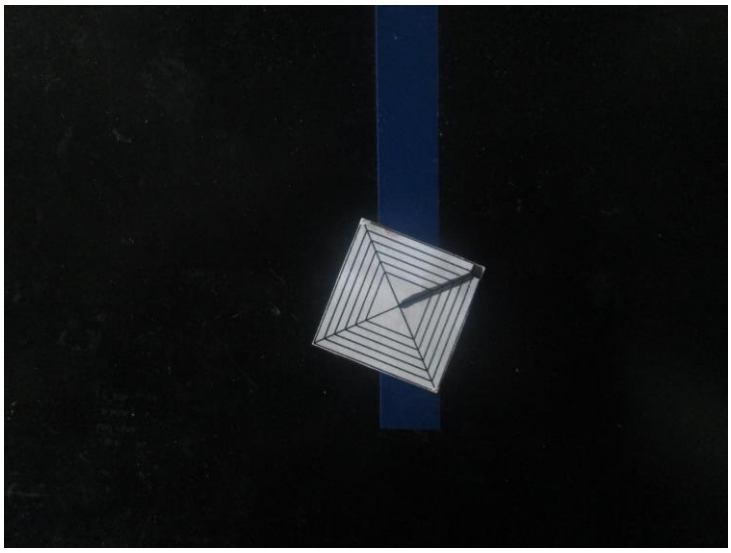
**Tablica 9. Kontrola kompenzacije greške odlaganja predmeta rada uslijed y pomaka na transportnoj traci.**




**Slika 74. Grafički prikaz kontrole kompenzacije greške odlaganja predmeta rada uslijed y pomaka na transportnoj traci.**

Dobiveni rezultati nisu zadovoljavajući, maksimalna radijalna greška povećala se sa 2 mm na 3.6 mmm, te se smanjio raspon y pomaka podloge odlaganja predmeta rada gdje je prosječna radijalna greška manja od 2 mm. Kompenzacija greške uslijed y pomaka predmeta na transportnoj traci se stoga neće koristiti.

Sljedeća kontrolna varijabla je orijentacija podloge odlaganja predmeta rada. Minimalni utjecaj greške y pomaka osigurava se referentnom linijom na transportnoj traci, isto kao u prethodnom slučaju [Slika 75].

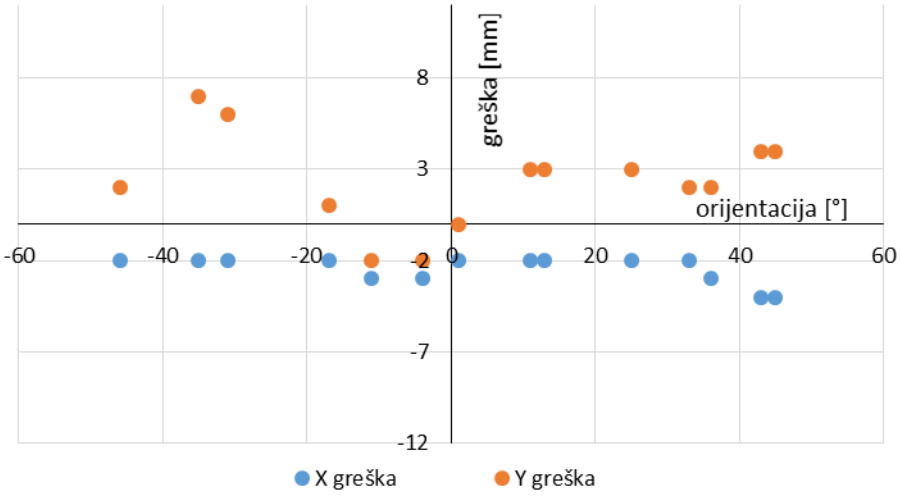


**Slika 75. Osiguravanje pozicije podloge za odlaganje predmeta rada.**

Rezultati mjerenja su prikazani u tablici [Tablica 10] i na grafu [Slika 76].

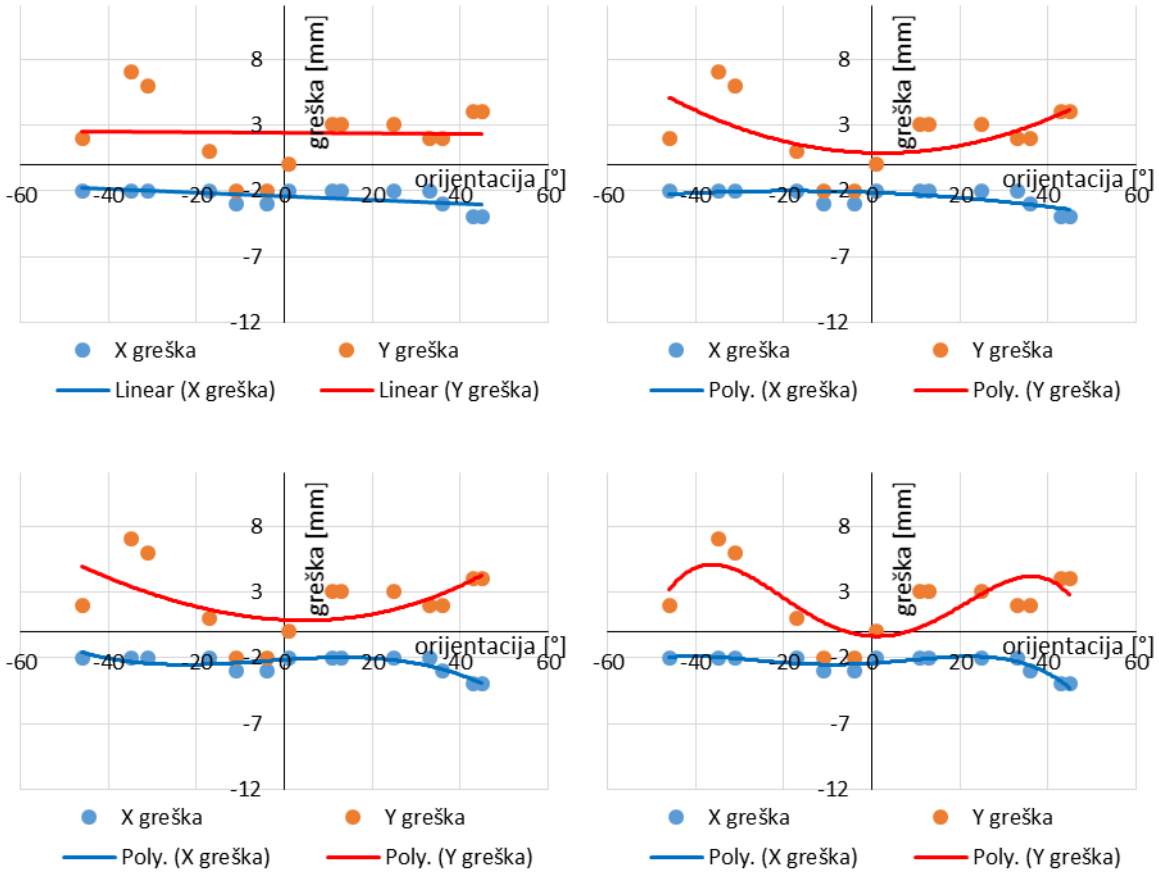
| | orijentacija [°] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -46 | -2 | 2 |
| 2 | -35 | -2 | 7 |
| 3 | -31 | -2 | 6 |
| 4 | -17 | -2 | 1 |
| 5 | -11 | -3 | -2 |
| 6 | -4 | -3 | -2 |
| 7 | 1 | -2 | 0 |
| 8 | 11 | -2 | 3 |
| 9 | 13 | -2 | 3 |
| 10 | 25 | -2 | 3 |
| 11 | 33 | -2 | 2 |
| 12 | 36 | -3 | 2 |
| 13 | 43 | -4 | 4 |
| 14 | 45 | -4 | 4 |

**Tablica 10. Mjerenje greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada.**




**Slika 76. Grafički prikaz greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada.**

Greška je nešto veća nego u prethodnom slučaju, te se kao i do sad razmatra nekoliko opcija kompenzacije iste [Slika 77].




**Slika 77. Usporedba aproksimacija greške orijentacije: jednadžbom pravca (gore lijevo), polinomom drugog reda (gore desno), polinomom trećeg reda (dolje lijevo), polinomom četvrtog reda (dolje desno).**

Za kompenzaciju greške orijentacije podloge odlaganja predmeta rada biraju se polinomi drugog reda (19) i (20).

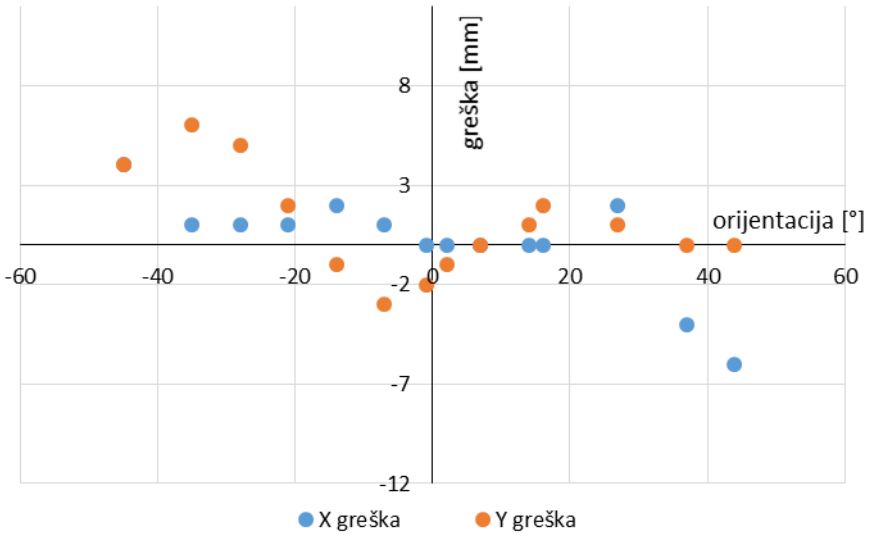
$$e\alpha_x = -0.0003 \cdot \alpha^2 - 0.013 \cdot \alpha - 2.1459 \quad \textbf{(19)}$$

$$e\alpha_y = 0.0018 \cdot \alpha^2 - 0.0085 \cdot \alpha + 0.86 \quad \textbf{(20)}$$

Nakon primjene kompenzacije, vrši se kontrolno mjerenje, vidljivo na tablici [Tablica 11] i grafu [Slika 78].

| | orijentacija [°] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | -45 | 4 | 4 |
| 2 | -35 | 1 | 6 |
| 3 | -28 | 1 | 5 |
| 4 | -21 | 1 | 2 |
| 5 | -14 | 2 | -1 |
| 6 | -7 | 1 | -3 |
| 7 | -1 | 0 | -2 |
| 8 | 2 | 0 | -1 |
| 9 | 7 | 0 | 0 |
| 10 | 14 | 0 | 1 |
| 11 | 16 | 0 | 2 |
| 12 | 27 | 2 | 1 |
| 13 | 37 | -4 | 0 |
| 14 | 44 | -6 | 0 |

**Tablica 11. Kontrola kompenzacije greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada.**




**Slika 78. Grafički prikaz kontrole kompenzacije greške odlaganja uslijed orijentacije podloge za odlaganje predmeta rada.**

Maksimalna radijalna greška smanjena je sa 7.3 mm na 6.1 mm, te je u rasponu od skoro 30° maksimalna greška smanjena na 2 mm. Ovakva kompenzacija je zadovoljavajuća, te je idući korak provjera utjecaja brzine transportne trake na točnost odlaganja predmeta rada. Podloga za odlaganje predmeta rada se pozicionira i orijentira kako bi greška bila minimalna, te se pomoću potenciometra na upravljačkoj jedinici elektromotora transportne trake postepeno povećava brzina. Rezultati mjerenja su ispisani u tablici [Tablica 12].

| | okretaji u minuti [$min^{-1}$] | X greška [mm] | Y greška [mm] |
|---|---|---|---|
| 1 | 17.6 | 0 | 0 |
| 2 | 19 | 0 | 0 |
| 3 | 21.6 | 0 | 0 |
| 4 | 23.7 | 0 | 0 |
| 5 | 25.6 | 0 | 0 |
| 6 | 27.8 | 0 | 0 |
| 7 | 29.7 | 0 | 0 |
| 8 | 31.8 | 0 | 0 |
| 9 | 33.8 | 0 | 0 |
| 10 | 35.9 | 0 | 0 |
| 11 | 37.9 | 0 | 0 |
| 12 | 40 | 0 | 0 |
| 13 | 41.9 | 0 | 0 |

**Tablica 12. Mjerenje greške odlaganja predmeta uslijed brzine transportne trake.**

Iz priložene tablice se vidi da brzina transportne trake nema vidljivog utjecaja na točnost odlaganja predmeta rada na transportnu traku.

S obzirom na mjerenja, u program za kompenzaciju unosi se samo kompenzacija greške uslijed orijentacije podloge za odlaganje predmeta [Slika 79].

```
13 BEGIN
14     GET_VAR (entry, '*SYSTEM*', '$VR[2].$OFFSET.$Y',yy, STATUS);
15     GET_VAR (entry, '*SYSTEM*', '$VR[2].$OFFSET.$R',aa, STATUS)
16     dx2=-0.0003*aa*aa-0.013*aa-2.1459;
17     dy2=0.0018*aa*aa-0.0085*aa+0.86;
18     WRITE(dx2,CR);
19     WRITE(dy2,CR);
20     visoffset = CURPOS(0,0);
21     konf = visoffset.config_data;
22     visoffset = POS(-dx2,-dy2,0,0,0,0,konf);
23     SET_POS_REG(8,visoffset, STATUS);
24 END KOMPENZACIJA
```

**Slika 79. Program za kompenzaciju greške pri odlaganju predmeta rada na transportnu traku.**

## 6.3. Validacija primjenjene kompenzacije greške

Na kraju, za validaciju razvijenog sustava, potrebno je izračunati prosječnu vrijednost greške prije i poslije primjenjene kompenzacije iste. Prije početka mjerenja bitno je imati na umu kako konačna greška odlaganja u sebi sadrži i grešku izuzimanja predmeta, te da će komponente kompenzacije greške koje su se do sad promatrale odvojeno, u kombinaciji drugačije ponašati. Neke komponente će pridonijeti boljoj ukupnoj kompenzaciji, dok je moguće da će neke kombinacije kontrolnih parametara donijeti do veće greške od očekivane. Jedini način da se donese pravilna validacija cijeloukupnog rada sustava je da se napravi niz mjerenja u što raznovrsnijoj kombinaciji kontrolnih parametara. Rasponi kontrolnih parametara biraju se iz do sad obavljenih mjerenja, na način da se uzimaju rasponi gdje se očekuje najmanja greška izuzimanja, odnosno odlaganja predmeta.

Rasponi kontrolnih parametara za izuzimanje predmeta rada:

- y pomak od -60 do 60 mm u referentnom koordinatnom sustavu transportne trake,
- orijentacija predmeta od 0 do 45°,
- brzina transportne trake od 17.6 do 40 $min^{-1}$.

Rasponi kontrolnih parametara za odlaganje predmeta rada:

- y pomak od -100 do 100 mm u referentnom koordinatnom sustavu transportne trake,
- orijentacija predmeta od -20 do 30°,
- brzina transportne trake od 17.6 do 40 $min^{-1}$.

Prvi set od 64 mjerenja (32 mjerenja izuzimanja i 32 mjerenja odlaganja predmeta rada), vrši se sa isključenim mehanizmima kompenzacije greške. Kako prilikom mjerenja nije primjećena greška orijentacije alata robota u odnosu na predmet rada, promatra se samo radijalna greška pozicioniranja, odnosno koliko je centar alata robota udaljen od centra predmeta rada.

U tablici [Tablica 13] parametri koji se odnose na transportnu traku za izuzimanje predmeta indeksirani su brojem 1, dok oni koji se odnose na transportnu traku za odlaganje predmeta su indeksirani brojem 2.

Na isti način proveden je drugi set od 64 mjerenja, ali sa uključenim mehanizmom kompenzacije greške. Rezultati su vidljivi u tablici [Tablica 14].

| $n_1$[min$^{-1}$] | $n_2$[min$^{-1}$] | $y_1$[mm] | $\alpha_1$[°] | $e_1$[mm] | $y_2$[mm] | $\alpha_2$[°] | $e_2$[mm] |
|---|---|---|---|---|---|---|---|
| 17.6 | 17.6 | -53 | 2 | 5 | -8 | 7 | 5.1 |
| | | -38 | 31 | 6 | -35 | -12 | 2.8 |
| | | 26 | 13 | 5 | -58 | 18 | 7.6 |
| | | 48 | 26 | 5 | 59 | -19 | 6.4 |
| | | 79 | 28 | 7 | -8 | 22 | 6.4 |
| | | -55 | -6 | 7 | 58 | -19 | 4.5 |
| | | 28 | 30 | 5 | 38 | 32 | 2.2 |
| | | 15 | 40 | 5 | -5 | 12 | 6.7 |
| 29.7 | 29.7 | 48 | 17 | 2 | 0 | -10 | 6.4 |
| | | 68 | 31 | 3 | -37 | 15 | 3 |
| | | -80 | 15 | 3 | -77 | -13 | 0 |
| | | 14 | 1 | 2 | -34 | -9 | 4.5 |
| | | 76 | 2 | 3 | 37 | 7 | 2 |
| | | 2 | 24 | 2 | -5 | 5 | 2.8 |
| | | -35 | -2 | 3 | -99 | -15 | 6.1 |
| | | 46 | -3 | 4 | -6 | 2 | 2 |
| 17.6 | 40 | -65 | 3 | 4 | 28 | -8 | 0 |
| | | -20 | 30 | 5 | 91 | -13 | 2.8 |
| | | 65 | 8 | 2 | 81 | 22 | 0 |
| | | 13 | 32 | 5 | -52 | 1 | 3 |
| | | 54 | 0 | 3 | -52 | 1 | 3 |
| | | -8 | 22 | 5 | -8 | 28 | 2.8 |
| | | 76 | 2 | 3 | -93 | -3 | 2 |
| | | 16 | 44 | 4 | 15 | -15 | 7.3 |
| 40 | 17.6 | -65 | 1 | 4 | -16 | -11 | 6.3 |
| | | -4 | 16 | 2 | -61 | 0 | 7.3 |
| | | -100 | 32 | 0 | 45 | 15 | 3 |
| | | 69 | -4 | 4 | 53 | -15 | 2.8 |
| | | 10 | 35 | 4 | -50 | 29 | 2.2 |
| | | -79 | 0 | 2 | -8 | -10 | 4.5 |
| | | 41 | 22 | 2 | 0 | 26 | 2 |
| | | -14 | -1 | 4 | 66 | 17 | 2.8 |

**Tablica 13. Prvi set mjerenja greške izuzimanja i odlaganja predmeta rada bez uključene kompenzacije.**

Iz dobivenih rezultata mjerenja računa se prosječna greška izuzimanja predmeta rada od 3.75 mm, te odlaganja predmeta rada od 3.76 mm. Pri tom maksimalna greška izuzimanja predmeta rada iznosi 7 mm, dok je maksimalna greška odlaganja istog 7.6 mm.

| $n_1$[$min^{-1}$] | $n_2$[$min^{-1}$] | $y_1$[mm] | $\alpha_1$[°] | $e_1$[mm] | $y_2$[mm] | $\alpha_2$[°] | $e_2$[mm] |
|---|---|---|---|---|---|---|---|
| 17.6 | 17.6 | -63 | 0 | 2 | -56 | 1 | 2 |
| | | -42 | 10 | 0 | 14 | -11 | 2.8 |
| | | -51 | 23 | 0 | 9 | 3 | 0 |
| | | 43 | 11 | 1 | 58 | 8 | 2 |
| | | 37 | 24 | 2 | -10 | 1 | 2 |
| | | -21 | -4 | 2 | 49 | -7 | 5.1 |
| | | -2 | 32 | 1 | 9 | 9 | 0 |
| | | -27 | 24 | 0 | -44 | 11 | 3 |
| 29.7 | 29.7 | -42 | 2 | 4 | 0 | 3 | 4.5 |
| | | 1 | 11 | 1 | 31 | -10 | 2.8 |
| | | 64 | 32 | 2 | 13 | 14 | 2 |
| | | 6 | 40 | 3 | -4.5 | 24 | 3 |
| | | -55 | 10 | 2 | -60 | 0 | 4.5 |
| | | -29 | 37 | 2 | -39 | -11 | 4.5 |
| | | 12 | 33 | 2 | 96 | 0 | 4.5 |
| | | 35 | 17 | 0 | -7 | -4 | 2.8 |
| 17.6 | 40 | -50 | 1 | 3 | -7 | 0 | 4.5 |
| | | -14 | 26 | 1 | -45 | -4 | 3 |
| | | 17 | 31 | 2 | -8 | 12 | 0 |
| | | 27 | 8 | 0 | -58 | 23 | 4.5 |
| | | 45 | 40 | 3 | 20 | -7 | 5 |
| | | 13 | 18 | 0 | -5 | -3 | 2.8 |
| | | -43 | 3 | 2 | 45 | -14 | 5.7 |
| | | 48 | 2 | 2 | -6 | -3 | 2 |
| 40 | 17.6 | -19 | 12 | 0 | -14 | 0 | 0 |
| | | 50 | 2 | 4 | 41 | 5 | 5 |
| | | -49 | 13 | 0 | 4 | -7 | 4.5 |
| | | 64 | 27 | 2 | -59 | 18 | 2 |
| | | -45 | 28 | 2 | 16 | 4 | 5.7 |
| | | 45 | 2 | 4 | -3 | -9 | 2.2 |
| | | -52 | 33 | 2 | -47 | 14 | 2.8 |
| | | -22 | 1 | 4 | -44 | 8 | 5.7 |

**Tablica 14. Drugi set mjerenja greške izuzimanja i odlaganja predmeta rada sa uključenom kompenzacijom.**

Nakon primjene mehanizma kompenzacije prosječna vrijednost greške izuzimanja predmeta rada je 1.72 mm, a odlaganja predmeta rada 3.15 mm. Maksimalna greška izuzimanja predmeta rada je 4 mm, te 5.7 mm za odlaganje predmeta rada.

Konačno, dobije se da je maksimalna greška izuzimanja predmeta rada smanjena sa 7 na 4 mm, prosječna sa 3.75 mm na 1.72 mm, te maksimalna greška odlaganja predmeta rada sa 7.6 na 5.7 mm i prosječna sa 3.76 mm na 3.15 mm.

# 7. ZAKLJUČAK

U razvoju sustava za robotsko rukovanje predmetima rada u pokretu korištena su dva vizijska sustava, dvije transportne trake i industrijski robot. Za odlaganje predmeta rada implementiran je *FANUC* strojni vid, dok je za izuzimanje predmeta rada sa pokretne trake projektiran vlastiti strojni vid u *LabView* računalnom programu.

Pri projektiranju vlastitog strojnog vida rješavao se problem kašnjenja informacija o predmetu rada uslijed realnog vremena potrebnog za komuniciranje između kamere, računala i upravljačke jedinice robota. Problem je uspješno rješen poklapanjem referentnih koordinatnih sustava robota i vizijskog sustava, te matematičkom obradom prikupljenih podataka predmeta rada.

Realizirani sustav strojnog vida omogućava kontinuirano rukovanje predmetima rada pomoću spremnika setova koordinata nađenih predmeta i senzorom koji pauziranjem trake osigurava da predmeti rada ne izađu iz radnog prostora robota. Robustan, autonoman sustav koji bez prekidanja proizvodnog procesa i potrebe za ljudskim djelovanjem može rukovati kontinuiranim tokom predmeta rada bitan je za industrijske primjene.

Drugi problem razrađen u ovom radu je velika greška pozicioniranja robota. Oštećeni robot je nakon servisa i kalibracije pokazivao grešku pozicioniranja od 20 mm, što ga za bilo kakvu industrijsku primjenu čini neupotrebljivim. Problemu se pristupilo mjerenjem greške u odnosu na kontrolne parametre pozicije, orijentacije i brzine predmeta rada, te analizom i aproksimacijom ponašanja greške, ista se pokušala kompenzirati.

Rezultati su ukazali na to da se prosječna greška izuzimanja uspjela smanjiti za 55%, dok se prosječna greška odlaganja predmeta smanjila za 26%. Unatoč tome, maksimalne vrijednosti greški su i dalje prevelike za većinu industrijskih primjena, te bi detaljni postupak kalibracije robota bio potreban za spuštanje greške u prihvatljivi raspon.

No, ovakav pristup kompenzaciji greške primjenjiv je u slučajevima „crne kutije“, kada se ne zna koji su unutarnji uzroci greške pozicioniranja robota, ali su vanjske posljedice greške vidljive i mjerljive.

# LITERATURA

[1] https://en.wikipedia.org/wiki/Assembly_line#Industrial_revolution

[2] https://robohub.org/wp-content/uploads/2014/08/Industrial-Robots-Hyundai-Heavy.jpg

[3] https://hr.wikipedia.org/wiki/Kutni_digitalni_pretvornik_pomaka

[4] Stipančić, T.: Vizijski sustavi, Fakultet strojarstva i brodogradnje, Zagreb, 2019.

[5] Ljeljak, Z.: Kalibracija vizijskog sustava korištenjem virtualne projekcije, Sveučilište u Zagrebu, 2016.

[6] Jakšić, T.: Dinamičko rukovanje predmetima rada na pokretnoj traci, Sveučilište u Zagrebu, 2014.

[7] https://hr.wikipedia.org/wiki/TCP

[8] FESTO Handling technology and positioning, System Guide Volume 1, 2002.

[9] https://en.wikipedia.org/wiki/Network_socket#Socket_states_in_the_client-server_model

[10] FANUC Robot series R-30iA CONTROLLER Karel Reference Manual, 2009.

[11] Crneković, M.: ROBOTIKA - uvod, Fakultet strojarstva i brodogradnje, Zagreb, 2017.

[12] NI Vision Concepts Manual, Natonial instruments 2008.

[13] NI Vision Builder for Automated Inspection Tutorial, National instruments 2008.

[14] FANUC Robot series R-30iA CONTROLLER Line Tracking OPERATOR'S MANUAL, 2009.

[15] FANUC Robot series R-30iA CONTROLLER iRVision Visual Tracking START-UP GUIDANCE, 2009.

# PRILOZI

I. CD-R disc

II. Korišteni *FANUC* i *KAREL* programski kodovi

# PROGRAMSKI KOD „ENKODER“

```
PROGRAM ENKODER
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE
%COMMENT= 'ENKODER'
---------------------------------------------
VAR
STATUS,cnt2,cnt, entry: INTEGER
scl: REAL
---------------------------------------------
BEGIN
        GET_VAR(entry,'*SYSTEM*','$LNSCH[2].$scale',scl,STATUS);
        GET_VAR(entry,'*SYSTEM*','$ENC_STAT[1].$enc_count',cnt,STATUS);
        DELAY 1000;
        GET_VAR(entry,'*SYSTEM*','$ENC_STAT[1].$enc_count',cnt2,STATUS);
        SET_REAL_REG(40,(cnt2-cnt)/(scl*25*0.10472),STATUS);
END ENKODER
```

# PROGRAMSKI KOD „KOMPENZACIJA“

```
PROGRAM KOMPENZACIJA
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE
------------------------------------------------------
VAR
aa,yy, dx, dy,x,y, dx2, dy2: REAL
STATUS, entry: INTEGER
visoffset: XYZWPR
konf: CONFIG
---------------------
BEGIN
        GET_VAR (entry, '*SYSTEM*', '$VR[2].$OFFSET.$Y',yy, STATUS);
        GET_VAR (entry, '*SYSTEM*', '$VR[2].$OFFSET.$R',aa, STATUS)
        --dx=-0.0035*yy+1.0156;
        --dy=-0.0146*yy+0.3728;
        dx2=-0.0003*aa*aa-0.013*aa-2.1459;
        dy2=0.0018*aa*aa-0.0085*aa+0.86;
        --WRITE(dx,CR);
        --WRITE(dy,CR);
        WRITE(dx2,CR);
        WRITE(dy2,CR);
        visoffset = CURPOS(0,0);
        konf = visoffset.config_data;
        visoffset = POS(-dx2,-dy2,0,0,0,0,konf);
        SET_POS_REG(8,visoffset, STATUS);
END KOMPENZACIJA
```

# PROGRAMSKI KOD „KOORD_SET“

```
PROGRAM KOORD_SET
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE
%COMMENT= 'PROBATRIGER'
VAR
KOORD FROM SIRINIC_KOM:  ARRAY[14,14] OF STRING[100]
X, real_value,a,dxa1,dya1,dxa,dya,xx,yy,dxy1,dyy1,dxy,dyy,v,dxv,dxv1,dyv,dyv1,kx: REAL
Z,Y, n, m, i, STATUS,entry,int_value: INTEGER
real_flag: BOOLEAN
--------------------
BEGIN
        GET_REG(24,real_flag,n,real_value,STATUS)
        FOR i = 1 TO 12 DO
                        CNV_STR_REAL(KOORD[n,i] ,X);
        --              IF UNINIT(X) THEN
                --              WRITE('dalje ne idje',CR)
                --      ELSE
                --              WRITE('ovaj brajo je dobar',CR)
                --      ENDIF
                                SET_REAL_REG(i,X,STATUS);
        ENDFOR
                --KOMPENZACIJA GRESKE Y OFFSETA PREDMETA
        ------------------------------------------------------------------------
    FOR i=5 TO 8 DO
        GET_REG(i,real_flag,int_value,yy,STATUS);
        dxy1=3E-7*yy*yy*yy-9E-6*yy*yy-0.0228*yy+0.3636;
```

```
        dyy1=-6E-7*yy*yy*yy-9E-6*yy*yy+0.016*yy-2.3293;
        GET_REG((i+4),real_flag,int_value,a,STATUS);
        dxy=COS(a)*dxy1-SIN(a)*dyy1-1;
        dyy=SIN(a)*dxy1+COS(a)*dyy1;
        GET_REG((i-4),real_flag,int_value,xx,STATUS);
        SET_REAL_REG((i+36),dxy,STATUS);
        SET_REAL_REG(i,yy-dyy,STATUS);
    ENDFOR
        ----KOMPENZACIJA GRESKE ORIJENTACIJE PREDMETA
        ------------------------------------------------------------------------
    FOR i=9 TO 12 DO
        GET_REG(i,real_flag,int_value,a,STATUS);
        dxa1=0.0025*a*a-0.2216*a+2.5613;
        dya1=2E-5*a*a*a+0.0013*a*a-0.1512*a+0.4056;
        dxa=(COS(a)*dxa1-SIN(a)*dya1)*0.85;
        dya=SIN(a)*dxa1+COS(a)*dya1;
        GET_REG((i+32),real_flag,int_value,kx,STATUS);
        GET_REG((i-4),real_flag,int_value,yy,STATUS);
      SET_REAL_REG((i+32),kx+dxa,STATUS);
        SET_REAL_REG((i-4),yy-dya,STATUS);
    ENDFOR
    ----KOMPENZACIJA BRZINE KONVEJERA
        ------------------------------------------------------------------------
    FOR i=9 TO 12 DO
        GET_REG(i,real_flag,int_value,a,STATUS);
        GET_REG(40,real_flag,int_value,v,STATUS);
        dxv1=-0.0041*v*v+0.11*v-0.6766;
        dyv1=0.2052*v-3.3946;
        dxv=COS(a)*dxv1-SIN(a)*dyv1;
```

```
        dyv=SIN(a)*dxv1+COS(a)*dyv1;
        GET_REG((i+32),real_flag,int_value,kx,STATUS);
        GET_REG((i-4),real_flag,int_value,yy,STATUS);
        --IF i=9 THEN
         --GET_REG(1,real_flag,int_value,xx,STATUS);
         --WRITE(xx,CR);
        --ENDIF
        SET_REAL_REG((i+32),kx+dxv,STATUS);
        SET_REAL_REG((i-4),yy-dyv,STATUS);
   ENDFOR
    GET_REG(41,real_flag,int_value,kx,STATUS);
    WRITE(kx,CR);
    WRITE(dxv,CR);
   CNV_STR_INT(KOORD[n,13],Y);
   CNV_STR_INT(KOORD[n,14],Z);
  SET_INT_REG(13,Y,STATUS);
  SET_INT_REG(14,Z,STATUS);
     SET_INT_REG(35,1,STATUS);
END KOORD_SET
```

## PROGRAMSKI KOD „PR_ACQ“

```
/PROG  PR_ACQ
/ATTR
OWNER                 = MNEDITOR;
COMMENT               = "";
PROG_SIZE   = 337;
CREATE                = DATE 19-03-26  TIME 16:55:00;
MODIFIED    = DATE 19-03-26  TIME 17:23:30;
FILE_NAME = ;
VERSION               = 0;
LINE_COUNT          = 10;
MEMORY_SIZE       = 809;
PROTECT               = READ_WRITE;
TCD:  STACK_SIZE = 0,
    TASK_PRIORITY          = 50,
    TIME_SLICE      = 0,
    BUSY_LAMP_OFF        = 0,
    ABORT_REQUEST        = 0,
    PAUSE_REQUEST        = 0;
DEFAULT_GROUP  = *,*,*,*,*;
CONTROL_CODE    = 00000000 00000000;
/APPL
/APPL
LINE_TRACK;
 LINE_TRACK_SCHEDULE_NUMBER      : 1;
 LINE_TRACK_BOUNDARY_NUMBER      : 0;
 CONTINUE_TRACK_AT_PROG_END      : TRUE;
```

```
/MN
   1:  LBL[100] ;
   2:  CALL VSTKGETQ('RADNA_PLOHA',2,0,18,1) ;
   3:  IF R[18]=0,JMP LBL[200] ;
   4:  STOP_TRACKING ;
   5:  JMP LBL[100] ;
   6:  LBL[200] ;
   7:  R[36]=1    ;
   8:  WAIT R[36]=0    ;
   9:   ;
  10:  JMP LBL[100] ;
/POS
/END
```

# PROGRAMSKI KOD „PR_PICK“

```
/PROG  PR_PICK
/ATTR
OWNER                = MNEDITOR;
COMMENT              = "";
PROG_SIZE   = 469;
CREATE               = DATE 18-11-14  TIME 02:03:12;
MODIFIED    = DATE 19-05-06  TIME 16:47:04;
FILE_NAME = ;
VERSION              = 0;
LINE_COUNT           = 16;
MEMORY_SIZE          = 925;
PROTECT              = READ_WRITE;
TCD:  STACK_SIZE = 0,
    TASK_PRIORITY          = 50,
    TIME_SLICE      = 0,
    BUSY_LAMP_OFF          = 0,
    ABORT_REQUEST          = 0,
    PAUSE_REQUEST          = 0;
DEFAULT_GROUP  = 1,*,*,*,*;
CONTROL_CODE    = 00000000 00000000;
/APPL
/APPL

LINE_TRACK;
 LINE_TRACK_SCHEDULE_NUMBER      : 1;
 LINE_TRACK_BOUNDARY_NUMBER      : 0;
```

```
  CONTINUE_TRACK_AT_PROG_END      : TRUE;


/MN
   1:  UTOOL_NUM=6 ;
   2:  UFRAME_NUM=0 ;
   3:  WAIT R[36]=1    ;
   4:  CALL KOMPENZACIJA    ;
   5:  LBL[200] ;
   6:  LINECOUNT[2] R[23] ;
   7:L PR[1] 666mm/sec CNT80 VOFFSET,VR[2] Tool_Offset,PR[2] Offset,PR[8]    ;
   8:L PR[1] 666mm/sec CNT80 VOFFSET,VR[2] Offset,PR[8]    ;
   9:  WAIT    .50(sec) ;
  10:  DO[114]=OFF ;
  11:  DO[113]=ON ;
  12:L PR[1] 666mm/sec CNT80 VOFFSET,VR[2] Tool_Offset,PR[2] Offset,PR[8]    ;
  13:  DO[112]=ON ;
  14:  CALL VSTKACKQ('RADNA_PLOHA',2,1) ;
  15:  R[36]=0    ;
  16:  LBL[900] ;
/POS
/END
```

## PROGRAMSKI KOD „PR_TEST5“

```
/PROG  PR_TEST5
/ATTR
OWNER                = MNEDITOR;
COMMENT              = "";
PROG_SIZE   = 447;
CREATE               = DATE 19-03-22  TIME 22:28:32;
MODIFIED    = DATE 19-04-01  TIME 22:15:22;
FILE_NAME = ;
VERSION              = 0;
LINE_COUNT           = 17;
MEMORY_SIZE          = 883;
PROTECT              = READ_WRITE;
TCD:  STACK_SIZE = 0,
    TASK_PRIORITY          = 50,
    TIME_SLICE      = 0,
    BUSY_LAMP_OFF          = 0,
    ABORT_REQUEST          = 0,
    PAUSE_REQUEST          = 0;
DEFAULT_GROUP  = *,*,*,*,*;
CONTROL_CODE    = 00000000 00000000;
/APPL
/APPL

LINE_TRACK;
 LINE_TRACK_SCHEDULE_NUMBER      : 0;
 LINE_TRACK_BOUNDARY_NUMBER      : 0;
```

```
  CONTINUE_TRACK_AT_PROG_END      : TRUE;


/MN
   1:  LBL[102] ;
   2:  IF R[16]=0,JMP LBL[102] ;
   3:  IF R[16]=1,JMP LBL[101] ;
   4:  LBL[101] ;
   5:  IF R[27]=1,JMP LBL[103] ;
   6:  IF R[27]=0,JMP LBL[104] ;
   7:  LBL[103] ;
   8:  LINECOUNT[1] R[22] ;
   9:  R[16]=0    ;
  10:  JMP LBL[102] ;
  11:  LBL[104] ;
  12:  LINECOUNT[1] R[22] ;
  13:  //SETTRIG LNSCH[2] R[22] ;
  14:  SELBOUND LNSCH[2] BOUND[1] ;
  15:  R[27]=1    ;
  16:  R[16]=0    ;
  17:  JMP LBL[102] ;
/POS
/END
```

# PROGRAMSKI KOD „PR_TRACKING“

```
/PROG  PR_TRACKING
/ATTR
OWNER                 = MNEDITOR;
COMMENT               = "";
PROG_SIZE   = 952;
CREATE                = DATE 18-11-13  TIME 20:36:20;
MODIFIED    = DATE 19-05-14  TIME 22:52:58;
FILE_NAME = ;
VERSION               = 0;
LINE_COUNT            = 29;
MEMORY_SIZE           = 1404;
PROTECT               = READ_WRITE;
LINE_TRACK;
 LINE_TRACK_SCHEDULE_NUMBER      : 0;
 LINE_TRACK_BOUNDARY_NUMBER      : 0;
 CONTINUE_TRACK_AT_PROG_END      : TRUE;

/MN
   1:  DO[109]=ON ;
   2:L P[1] 350mm/sec CNT100    ;
   3:  R[36]=0    ;
   4:  R[21]=1    ;
   5:  R[16]=0    ;
   6:  R[14]=1    ;
   7:  R[17]=0    ;
   8:  R[27]=0    ;
```

```
  9:  WAIT   1.50(sec) ;
 10:  CALL ENKODER    ;
 11:  RUN SIRINIC_KOM ;
 12:  RUN PR_TEST5 ;
 13:  CALL VSTKCLRQ('RADNA_PLOHA') ;
 14:  CALL VSTKSTRT    ;
 15:  RUN PR_ACQ ;
 16:  LINE[1] ON ;
 17:  WAIT DI[102]=ON    ;
 18:  DO[112]=ON ;
 19:  LBL[100] ;
 20:   ;
 21:  CALL PROBA    ;
 22:  IF R[17]=1,JMP LBL[900] ;
 23:  CALL VSTKSTRT    ;
 24:L P[1] 650mm/sec CNT100    ;
 25:  JMP LBL[100] ;
 26:   ;
 27:  LBL[900] ;
 28:  CALL VSTKSTOP    ;
 29:  DO[109]=OFF ;
/POS
P[1]{
  GP1:
        UF : 0, UT : 6,                  CONFIG : 'N, , 0, -1',
        X =    78.615  mm,    Y =   102.056  mm,    Z = -1231.571  mm,
        W =  -179.658 deg,    P =    -3.290 deg,       R =   163.444 deg
};
/END
```

# PROGRAMSKI KOD „PROBA“

```
/PROG  PROBA
/ATTR
OWNER                = MNEDITOR;
COMMENT              = "";
PROG_SIZE   = 289;
CREATE               = DATE 18-11-09  TIME 19:32:52;
MODIFIED    = DATE 19-04-01  TIME 21:14:36;
FILE_NAME = ;
VERSION              = 0;
LINE_COUNT           = 7;
MEMORY_SIZE          = 773;
LINE_TRACK;
 LINE_TRACK_SCHEDULE_NUMBER      : 0;
 LINE_TRACK_BOUNDARY_NUMBER      : 0;
 CONTINUE_TRACK_AT_PROG_END      : FALSE;

/MN
   1:  LBL[102] ;
   2:  IF R[27]=0,JMP LBL[102] ;
   3:  IF R[27]=1,JMP LBL[101] ;
   4:  LBL[101] ;
   5:  R[25]=R[22]    ;
   6:  CALL PROBA_TRACK    ;
   7:  JMP LBL[102] ;
/POS
/END
```

# PROGRAMSKI KOD „PROBA_TRACK“

```
/PROG  PROBA_TRACK
/ATTR
OWNER                 = MNEDITOR;
COMMENT               = "";
PROG_SIZE   = 805;
CREATE                = DATE 18-11-09  TIME 19:37:08;
MODIFIED    = DATE 19-05-17  TIME 18:35:16;
FILE_NAME = ;
VERSION               = 0;
LINE_COUNT            = 35;
MEMORY_SIZE           = 1313;
PROTECT               = READ_WRITE;
LINE_TRACK;
 LINE_TRACK_SCHEDULE_NUMBER      : 2;
 LINE_TRACK_BOUNDARY_NUMBER      : 0;
 CONTINUE_TRACK_AT_PROG_END      : FALSE;

/MN
   1:  WAIT R[19]>0    ;
   2:  FOR R[24]=1 TO R[19] ;
   3:  WAIT R[34]=1    ;
   4:  CALL KOORD_SET    ;
   5:  SETTRIG LNSCH[2] R[14] ;
   6:  PR[3,2]=R[1]    ;
   7:  WAIT R[35]=1    ;
   8:  FOR R[21]=1 TO R[13] ;
```

```
 9:  PR[3]=PR[6]    ;
 10:  R[45]=R[21]+40    ;
 11:  PR[3,1]=R[R[21]]-R[1]-R[R[45]]    ;
 12:  R[21]=R[21]+4    ;
 13:  PR[3,2]=R[R[21]]    ;
 14:  R[21]=R[21]+4    ;
 15:  PR[3,6]=R[R[21]]    ;
 16:  R[21]=R[21]-8    ;
 17:L PR[7] 650mm/sec FINE Tool_Offset,PR[2] Offset,PR[3]    ;
 18:L PR[7] 650mm/sec FINE Offset,PR[3]    ;
 19:  DO[114]=ON ;
 20:  DO[113]=OFF ;
 21:  WAIT    .20(sec) ;
 22:L PR[7] 650mm/sec FINE Tool_Offset,PR[2] Offset,PR[3]    ;
 23:  STOP_TRACKING ;
 24:  DO[109]=ON ;
 25:  CALL PR_PICK    ;
 26:  ENDFOR ;
 27:  R[27]=0    ;
 28:  R[35]=0    ;
 29:  R[21]=0    ;
 30:  R[45]=0    ;
 31:  IF R[24]=R[19],JMP LBL[666] ;
 32:  ENDFOR ;
 33:  LBL[666] ;
 34:  R[24]=1    ;
 35:  R[19]=0    ;
/POS
/END
```

# PROGRAMSKI KOD „REDGE“

```
/PROG  REDGE
/ATTR
OWNER                = MNEDITOR;
COMMENT              = "";
PROG_SIZE   = 251;
CREATE               = DATE 19-03-06  TIME 21:34:26;
MODIFIED    = DATE 19-03-25  TIME 16:37:52;
FILE_NAME = ;
VERSION              = 0;
LINE_COUNT           = 4;
MEMORY_SIZE          = 607;
PROTECT              = READ_WRITE;
LINE_TRACK;
 LINE_TRACK_SCHEDULE_NUMBER      : 0;
 LINE_TRACK_BOUNDARY_NUMBER      : 0;
 CONTINUE_TRACK_AT_PROG_END      : TRUE;
/MN
   1:  F[1]=(OFF) ;
   2:  IF (DI[102] AND !F[2]),F[1]=(ON) ;
   3:  F[2]=(DI[102]) ;
   4:  IF (F[1]),R[16]=(1) ;
/POS
/END
```

# PROGRAMSKI KOD „SIRINIC_KOM“

```
PROGRAM SIRINIC_KOM
%NOLOCKGROUP
%NOPAUSE = ERROR + COMMAND + TPENABLE
%COMMENT= 'KOMUNIKACIJA'
------------------------------------------------------
VAR
x,z, real_value,broj : REAL
prog_index, int_value, int_value2, tmp_int, ENTRY, STATUS, K, nb, i,j, ack, Y,n,m : INTEGER
S,NIZ,trig : STRING[100]
tmp_str : STRING[128]
loop1,real_flag : BOOLEAN
KOORD : ARRAY[14,14] OF STRING[100]
file_var : FILE
jpos: JOINTPOS
------------------------------------------------------------
BEGIN
        WRITE('SIRINIC_KOM',CR);

        ----------------------------------------
        --RESETIRANJE TAGOVA
        SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$OPER',0,STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STATE',0,STATUS) ; DELAY 20
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$COMMENT','SOUND',STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$PROTOCOL','SM',STATUS) ;
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$REPERRS','FALSE',STATUS) ;
```

```
SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$TIMEOUT',9999,STATUS) ;
SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$PWRD_TIMEOUT',0,STATUS) ;
SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$SERVER_PORT',5555,STATUS) ;

SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STRT_PATH','192.168.123.27',STATUS)
;

SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STRT_REMOTE','192.168.123.27',STAT
US) ;
SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$PATH','192.168.123.27',STATUS) ;
SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$remote','192.168.123.27',STATUS) ;
-------------------------------------------
--POKRETANJE TAG 5 NAKON RESETIRANJA I KONEKCIJA
 SET_FILE_ATR(file_var, ATR_IA)
 SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$OPER',3,STATUS);
    SET_VAR(entry,'*SYSTEM*','$HOSTS_CFG[5].$STATE',3,STATUS);
    WRITE('Connecting..',CR)
    MSG_DISCO('S5:',STATUS)
    IF STATUS=0 THEN
        WRITE('odspojen',CR)
    ENDIF
    MSG_CONNECT('S5:',STATUS)
    WHILE STATUS<>0 DO
        MSG_CONNECT('S5:',STATUS)
    ENDWHILE
    WRITE (STATUS,CR)
    CLR_IO_STAT(file_var)
    OPEN FILE file_var('rw','S5:')
    loop1=TRUE;
```

```
        SET_INT_REG(35,0,STATUS);
        SET_INT_REG(19,0,STATUS);
        SET_INT_REG(15,0,STATUS);
        SET_INT_REG(34,0,STATUS);
        FOR i = 1 TO 12 DO
                        SET_REAL_REG(i,0,STATUS);
        ENDFOR
      SET_INT_REG(13,0,STATUS);
        GET_REG(19,real_flag,n,real_value,STATUS);
        WHILE loop1=TRUE DO
          WAIT FOR DIN[102]=OFF
              WAIT FOR DIN[102]=ON
                  SET_INT_REG(34,0,STATUS);
                  DELAY 100
                GET_REG(22,real_flag,int_value,real_value,STATUS)
                GET_REG(25,real_flag,int_value2,real_value,STATUS)
                GET_REG(19,real_flag,n,real_value,STATUS);
                SET_INT_REG(30,(int_value-int_value2),STATUS);
             IF((int_value-int_value2)<1984)  OR  ((  int_value-int_value2)>33480)
THEN
                SET_INT_REG(19,n+1,STATUS);
                CNV_INT_STR(int_value,1,0,trig);
                  WRITE file_var('1',CR)

                  ----------READ PETLJA---------
               SET_INT_REG(20,1,STATUS);
          --  CNV_STR_REAL(KOORD[1],X);
              GET_REG(19,real_flag,n,real_value,STATUS);
```

```
                    NIZ=''; S='';      K=0 ;
                BYTES_AHEAD(file_var,nb,STATUS)
            WHILE nb=0 DO
                    BYTES_AHEAD(file_var,nb,STATUS)
            ENDWHILE
              -- IF STATUS<>0 THEN; GOTO reconn_;ENDIF
             --   IF nb>0
                    WRITE('NB ',nb::3::0,' ', CR)
                    WHILE nb>0 DO
                        READ file_var(S::1); IF S=',' THEN;S='.';ENDIF
                        SET_INT_REG(20,K,STATUS);
                        IF (S='@') OR (S='P') THEN; K=K+1; KOORD[n,K]='';
                            --  IF K=10 THEN;K=9;ENDIF      --dodatan uvijet da K ne pređe izvan raspona od 70
                            ENDIF
                        WRITE(S);      -- MMM ISPIS CIJELE PORUKE NA EKRAN
                        IF (S<>'@') AND (S<>'P') AND (S<>'K') THEN ;KOORD[n,K]=KOORD[n,K]+S;ENDIF
                        NIZ=NIZ+S
                        nb=nb-1
                            BYTES_AHEAD(file_var,nb,STATUS)
                        IF nb=0 THEN; WRITE (CR);ENDIF
                    ENDWHILE
             KOORD[n,14]=trig
                    SET_INT_REG(34,1,STATUS);
            ENDIF
        ENDWHILE
END SIRINIC_KOM
```