

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA STROJNIŠTVO

Gregor Germadnik

**Načrtovanje poprocesorja za pretvorbo NC/ISO G-
kode v translacije za 3/5 osni robot ACMA ABB**

XR701

Zaključno delo

Maribor, Avgust 2019

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA STROJNIŠTVO

Gregor Germadnik

**Načrtovanje poprocesorja za pretvorbo NC/ISO G-
kode v translacije za 3/5 osni robot ACMA ABB**

XR701

Zaključno delo

Maribor, Avgust 2019

**Načrtovanje poprocesorja za pretvorbo NC/ISO G-kode v
translacije za 3/5 osni robot ACMA ABB XR701
Zaključno delo**

Študent: Gregor Germadnik
Študijski program: Visokošolski študijski program
Mehatronika
Mentor FERI: izr. prof. dr. ALEŠ HACE
Mentor FS: izr. prof. dr. KARL GOTLIH
Somentor: Timi Karner
Lektorica: Manca Potušek, prof. angleščine in slovenščine

Zahvala

Zahvaljujem se mentorjema za podporo, pomoč in možnost pisanja zaključnega dela, somentorju za intenzivno pomoč pri razvoju in diskusiji praktičnega dela.

Načrtovanje poprocesorja za pretvorbo NC/ISO G-kode v translacije za 3/5 osni robot ACMA ABB XR701

Ključne besede: stroj CNC, programski jezik C++, ISO G-koda, linearna algebra, program NC, poprocesor, robot

UDK: 004.4'42:007.52(043.2)

Povzetek

V zaključnem delu prikazujemo, kako z implementacijo poprocesorja za pretvorbe NC/ISO G-kode v translacije 3/5 osnega robota ACMA-proizvajalca ABB-model XR701 zmanjšamo strošek nakupa robota in pripadajoče programske opreme. Poprocesor z linearno algebro in razčlenjevanjem pretvori program za rezkanje kompleksnejših oblik, večjih dimenzij in pohitri nastavitve ter pretvorbo kode z uporabo uporabnikom prijaznega vmesnika izdelanega v programski opremi Visual Studio 2017 CLI (angl. Combined Language Infrastructure) in programskim jezikom C++. Na ta način privarčujemo pri nakupu licence programske opreme za pretvarjanje ISO G-kode v robotsko kodo.

Planning a postprocessor to convert NC/ISO G-code to translations for a 3/5 axis robot ACMA ABB XR701

Key words: machine CNC, programming language C++, ISO G-code, linear algebra, program NC, postprocessor, robot

UDK: 004.4'42:007.52(043.2)

Abstract

In the final work, it is presented how with the implementation of a postprocessor for converting NC/ISO G-code into translations for the robot ACMA manufacturer ABB model XR701, the costs of purchasing the robot and the associated software are reduced. The postprocessor with linear algebra and parsing converts the program for cutting complex shapes and bigger dimensions, speeds up the settings and conversion using a user-friendly interface created in the software Visual Studio 2017 CLI (Combined Language Infrastructure) and C++ programming language. This saves us the purchase of licensed software programs for converting ISO G-code into a robotic code.

KAZALO

1	UVOD.....	1
1.1	Opis splošnega področja dela	1
1.2	Cilj in namen zaključne naloge	1
1.3	Struktura diplomskega dela	2
2	PREGLED STANJA SISTEMA IN ZAHTEV OBRAVNAVANE PROBLEMATIKE	3
3	OPREDELITEV SISTEMA	5
3.1	Industrijski roboti in robot ACMA ABB XR701	5
3.1.1	Opis robota ACMA ABB XR701	6
3.1.2	Vrste zapestja robotov in robota ACMA ABB XR701	8
3.1.3	Glavne značilnosti robota ACMA ABB XR701	9
3.2	Obdelovalni stroji CNC	11
3.2.1	Programiranje NC- in CNC-strojev	12
3.2.2	NC-program ali G-Koda	13
3.3	Primerjava CNC-stroja z robotom	16
3.4	Poprocesor za pretvorbo G-kode	17
4	RAZVOJ POPROCESORJA	19
4.1	Uporabniški vmesnik	19
4.1.1	Nastavitev robota	21
4.1.2	Konfiguracija robota	21
4.1.3	Vhodni podatki	22
4.1.4	Izhodni podatki	23
4.2	Pretvorba G-kode v program robota ACMA	23
4.2.1	Zajem podatkov in struktura numeričnega programa za učenje točk robota ACMA XR701	26
4.2.2	Razčlenjevanje podatkov in preverjanje zapolnjenost vrstice	29
4.2.3	Pretvorba krožnih izsekov G2 in G3	32
4.3	Evaluacija	39
5	SKLEP	41

KAZALO SLIK

SLIKA 1 PRIKAZ ROBOTA ACMA ABB XR701	6
SLIKA 2 PRIKAZ KINEMATIČNE VERIGE ROBOTA ACMA ABB XR701.....	6
SLIKA 3 ZAPESTJE RBR OSNA IZVEDBA.....	8
SLIKA 4 RBR – KOTNA IZVEDBA	8
SLIKA 5 RRR – DVOJNA KOTNA IZVEDBA	9
SLIKA 6 SHEMATSKI PRIKAZ KOMPONENT ROBOTA ACMA ABB XR701	10
SLIKA 7 DELOVNI PROSTOR ROBOTA ACMA ABB XR701.....	10
SLIKA 8 PRIKAZ 3-OSNEGA CNC-STROJA.....	11
SLIKA 9 PRIKAZ 5-OSNEGA CNC-STROJA.....	11
SLIKA 10 ZGRADBA CNC-PROGRAMA	14
SLIKA 11 PRIKAZ PRETOKA PODATKOV.....	18
SLIKA 12 OKNO ZA NASTAVITEV ROBOTA	21
SLIKA 13 OKNO ZA KONFIGURACIJO ROBOTA	22
SLIKA 14 OKNO ZA NALAGANJE VHODNE G-KODE	22
SLIKA 15 OKNO ZA UPRAVLJANJE Z IZHODNIMI PODATKI.....	23
SLIKA 16 DIAGRAM POTEKA GLAVNEGA API ZA PRETVORBO G-KODE.....	25
SLIKA 17 SEKVENCA ZA NASTAVITEV NIČELNE TOČKE	27
SLIKA 18 SEKVENCA ZA NASTAVITEV NAČINA DODAJANJA KOORDINAT	27
SLIKA 19 KOORDINATNI SISTEMI ROBOTA.....	28
SLIKA 20 SEKVENCA ZA DOLOČANJE TOČK, HITROSTI IN POSPEŠKA.....	28
SLIKA 21 SEKVENCA NOGE PROGRAMA ACMA ZA DODAJANJE TOČK.....	28
SLIKA 22 DIAGRAM POTEKA ZA RAZČLENJEVANJE G-KODE.....	31
SLIKA 23 ZASEBNA METODA »G2_ALGEBRA« ZA PRETVORBO UKAZA G2 KROŽNEGA GIBA	33
SLIKA 24 DIAGRAM POTEKA ZA IZPIS PROGRAMA ACMA	34
SLIKA 25 PRIMER NC-PROGRAMA.....	35
SLIKA 26 PRIMER G-KODE OZIROMA NC-PROGRAMA KROŽNEGA IZSEKA G2 IN G3	35
SLIKA 27 PRIKAZ IZPELJAVE INVERZNEGA TANGENSA 2.....	37

KAZALO TABEL

TABELA 1 PRIKAZ KARAKTERISTIKE ROBOTA.....	9
TABELA 2 VREDNOSTI TIPK KONZOLE ZA PROGRAMIRANJE ROBOTA ACMA.....	26

KAZALO PSEVDOKOD

PSEVDOKOD 1 GLAVNI APLIKACIJSKI VMESNIK ZA PRETVORBO G-KODE	24
PSEVDOKOD 2 PSEVDOKODA ZA DOLOČANJE SEKVENCE ZA NASTAVITEV KONFIGURACIJE ROBOTA.....	27
PSEVDOKOD 3 RAZČLENJEVANJE VHODNE G-KODE	29
PSEVDOKOD 4 PRIDOBIVANJE VREDNOSTI PARAMETROV	30
PSEVDOKOD 5 PREVERJANJE ZAPOLNJENOST VRSTICE	30
PSEVDOKOD 6 DOLOČANJE DELOVNE POVRŠINE	38

Kratice

- API – Application Programming Interface
- CAD – Computer Aided Design
- CAM – Computer Aided Manufacturing
- CL – Cutter Location
- CNC – Computer Numerical Control
- GUI – Graphical User Interface
- ISO – International Standards Organization
- NC – Numerical Control
- PID – Proportional Integral Derivative

1 UVOD

1.1 Opis splošnega področja dela

Industrijski roboti so bili v preteklosti redko uporabljeni, saj so ročno krmiljeni stroji zagotavljali zadovoljivo natančnost in dolgoročno ponovljivost. Z razvojem regulatorja PID (angl. Proportional Integral Derivative, slo. proporcionalno integralno diferencialni) je postala aktualna uporaba računalniško numerično krmiljenih strojev, saj je regulator omogočal izjemno natančno vodenje motorjev stroja. Robotizacija se je uveljavila v proizvodnjo z razvojem gonil, in sicer cikloidnih, planetarnih in harmoničnih gonil s poudarkom na harmoničnih gonilih, ki zagotavljajo natančnejše določanje položajev osi robota. S tem se roboti približajo natančnosti enostavnejšim strojem CNC (angl. Computer Numerical Control, slo. računalniško numerično krmiljenje), vendar jih z vodenjem več prostorskih stopenj težko presegajo. Namenjeni so za enostavne, monotone in nevarne operacije, ki povzročajo poklicne bolezni pri človeku. Z eksponentnim razvojem se zaradi Moorovega zakona vedno hitreje izdelujejo novejši roboti in posledično njihova programska oprema, kar povzroča zahtevno uporabo starejših modelov različnih proizvajalcev. Mehanske lastnosti se ne ujemajo več z novo programsko opremo in omejujejo nakup novejšega robota istega proizvajalca.

1.2 Cilj in namen zaključne naloge

Namen diplomske naloge je opisati možnost uporabe starejšega robota za izvajanje zahtevnejše operacije. Cilj diplomske naloge je izdelati poprocesor za enostavno pretvorbo NC (angl. Numerical Control, slo. numerično krmiljenje) oz. ISO (angl. International Standards Organization, slo. Internacionalna organizacija standardov) G-Kodo v 6-osni program oziroma robotu razumljiv numerični program.

Cilji zaključne naloge so:

- preučiti NC/ISO G-Kodo;
- preučiti program za dodajanje točk robota ACMA ABB XR701;
- pripraviti enačbe za izračun absolutnih položajev robota glede na NC/ISO G-Kodo;
- izdelava poprocesorja za razčlenjevanje vhodne NC/G-Kode in strukturiranje izhodnega programa za dodajanje točk;
- izdelava grafičnega uporabniškega vmesnika za enostavnejšo nastavitvev robota in pretvorbo NC/G-Kode.

1.3 Struktura diplomskega dela

Struktura zaključnega dela je v nadaljevanju naslednja: v poglavju 2 predstavimo stanje robota in navedemo zahteve zaključnega dela. V poglavju 3 sta opisana robot ACMA ABB XR701 in obdelovalni stroj CNC, njuna primerjava ter poprocesor za pretvorbo. V četrtem poglavju opisujem razvoj poprocesorja, in sicer zgradbo uporabniškega vmesnika in kako ga uporabljamo. V drugem delu opisujemo, kako zajamemo podatke, jih razčlenimo, pretvorimo in strukturiramo v program, ki ga razume program za dodajanje točk ACMA ABB XR701. V tretjem delu del predstavimo evaluacijo razvoja, opisujemo težave, ki smo imeli, in ugotovitve pri poskusih na realnem robotu in simulaciji. V zadnjem, petem poglavju povzamemo opravljeno delo in napovemo možnosti za nadaljnje delo.

2 PREGLED STANJA SISTEMA IN ZAHTEV OBRAVNAVANE PROBLEMATIKE

Leta 2005 je bil Fakulteti za strojništvo v Mariboru podarjen robot ACMA XR701, ki je namenjen izobraževanju študentov. Robotova glavna naloga prejšnjega okolja v podjetju Revoz je bilo točkovno varjenje kosovne proizvodnje. Robot je bil uporabljen za opravljanje projektnih, diplomskih in magistrskih nalog, na njem je bila opravljena tudi doktorska disertacija. Z nalogami in sodelovanjem med študenti ter profesorji so omogočili izvajanje različnih operacij. Implementacija varnosti robota ACMA s programabilnim logičnim krmilnikom APRIL 3000 in komunikacijo JNET med njima je bil najpomembnejša implementacija, saj zagotavlja varno upravljanje robota in izvajanje programov v avtomatskem načinu.

V tem zaključnem delu smo podrobno opisali uporabo robota za izvajanje nalog CNC-stroja glede na že obstoječ članek Načrtovanje poprocesorja za pretvorbo NC/ISO G-kode v translacije za 3/5-osnega robota ACMA ABB XR701.

Glavna tematika zaključnega dela je uporaba programa NC, ki je pridobljen s poprocesiranjem CL (angl. Cutter location) oziroma položajev orodja za določanje trajektorije robotov oziroma strojev CNC. Poprocesor je vmesno orodje, ki pripravi unikaten program za proizvodne stroje s specifično določenimi mehanskimi lastnostmi z obdelavo podatkov programske opreme CAD (angl. Computer Aided Design, slo. računalniško podprto načrtovanje) v programsko opremo CAM (angl. Computer Aided Manufacturing slo. računalniško podprta proizvodnja).

Na to tematiko se navezuje sorodna diplomska naloga Sekirnik [12], kjer opisuje poprocesiranje CAD- in CAM-podatkov s programsko opremo Siemens NX 8.5 v NC-program. NC-program nato s pomočjo programa izdelanega v programski opremi Microsoft Office Excel generira numerični program za dodajanje točk robotu ACMA ABB XR701. Format numeričnega krmilnega programa je kompleksen. Zanj je treba imeti predznanje samega programa za dodajanje točk in hkrati tudi robota za njegovo nastavitvev. Omenjena metoda programiranja robota je neposredna. V magistrskem

delu Filipič [2] opisuje posredno programiranje robota z uporabo programske opreme RobotStudio. Oprema omogoča učenje robota proizvajalca ABB brez poseganja v industrijske procese z izdelavo virtualnega okolja, ki vsebuje ustvarjen model robota ACMA ABB XR701. S programsko kodo ABB vodi robota z ukazi MoveJ in MoveL. Z dodatnim programom nato ustvari tekstovno datoteko za dodajanje točk v realnem robotu. Magistrsko delo Nilssona [9] in članek [11] še dodatno uporabita pretvorbo ukazov G2 in G3 v ukaz MoveC za krožni gib robota ABB. [3]

Zaključno delo opisuje razvoj dodatnega poprocesorja, ki iz pridobljene NC- oziroma G-Kode ponovno pretvori program v numerični jezik, razumljiv programu za dodajanje točk. Vhodni podatek poprocesorja je 3- ali večosna NC-koda oziroma G-Koda za CNC-stroj, ki vsebuje CL-pozicije s podatki obdelave. Kot opisujejo sorodna dela, smo tudi pri tem poprocesorju z metodo razčlenitve izluščili s kode CL-pozicije, hitrosti in pospeške orodja. Vendar smo pri tem zaključnem delu ukaze G2 in G3 pretvorili s pomočjo linearne algebre, in sicer so bili krožni gibi pretvorjeni v interpolirane daljice, katerih gostota se je ročno nastavila. Obdelava kode z ukazi G0 in G1 pa se obdeluje na podoben način kot v sorodnih delih. Izdelali smo grafični uporabniški vmesnik GUI, ki je prikazoval in izvedel pretvorbo. Prav tako je omogočal dodatne nastavitve robota kot na primer nastavitve položaja polizdelka in konfiguracije robota. Poprocesor po pretvorbi NC- oziroma G-Kode strukturira numerični program za učenje točk robota in ga prikaže v GUI. Po poprocesiranju je treba ponovno obdelati podatke s programom, ki iz programa za učenje točk pretvori točke v numerični robotski jezik.

3 OPREDELITEV SISTEMA

3.1 Industrijski roboti in robot ACMA ABB XR701

Industrijski robot je opisan s standardom ISO 8373, in sicer: »Industrijski robot je manipulator, ki je povratno zračno voden, reprogramabilen in večnamenski sistem. Lahko je fiksni ali mobilni. Programabilen je v treh ali več prostostnih stopnjah.

Uporabljamo ga v procesih industrijske avtomatizacije.

- Robotske mehanizme poganjajo električni ali hidravlični motorji, ki niso odprtozračno krmiljeni. Sestavni del robotskega manipulatorja so vselej tudi senzorji. Tu gre predvsem za notranje senzorje, ki so nameščeni v sklepih robota. To so pretvorniki kota ali razdalje in hitrosti. Pomembni pa so tudi zunanji senzorji, med njimi predvsem senzor sile dotika. Glede na želeno gibanje vrha robota, ki ga določi uporabnik, ter glede na informacijo senzorjev robotski regulacijski sistem vodi manipulator bodisi po položaju ali po sili.
- Sodobni industrijski proizvodni procesi se odvijajo brez velikih zalog materiala. Kot posledica se na isti liniji v istem dnevu pojavijo različni tipi posameznega izdelka. Lastnost reprogramabilnosti nam omogoča, da zgolj s pritiskom na gumb na zelo enostaven način preidemo s proizvodnje enega tipa izdelka na drugega.
- Robot poskuša biti nekakšen posnetek človekove roke. Enako kot uporabljamo roko za natančna in groba opravila, skušamo isti robotski manipulator uporabiti za različne naloge. To je še posebej pomembno, ker je ekonomska življenjska doba robota razmeroma dolga (12 do 16 let). Tako se zgodi, da smo nek industrijski robot kupili za varjenje, kasneje pa nam bo služil za prenašanje in urejanje izdelkov v palete. [1]

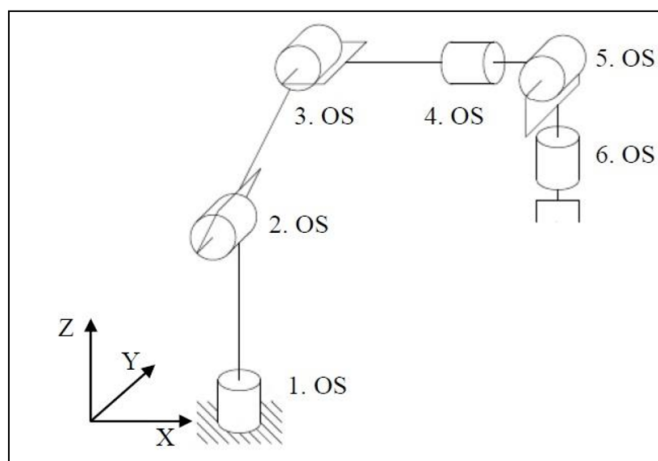
3.1.1 Opis robota ACMA ABB XR701

Renault Automatization ACMA je leta 1994 leta izdelal robota ACMA XR701 (Slika 1), sedaj spada pod podjetje ABB. Prejšnji lastnik ga je uporabljal v avtomatizirani kosovni proizvodnji za točkovno varjenje zaradi njegove zanesljivosti, hitrosti in ponovljivosti.



Slika 1 Prikaz robota ACMA ABB XR701

Antropomorfni industrijski robot ima šest prostostnih stopenj (Slika 2) in velik delovni prostor (Slika 7), zato je namenjen za manipulacijo, strego in sestavo. Slika 2 prikazuje odprto kinematično verigo, ki zagotavlja večjo gibljivost in ima za posledico manjšo togost.



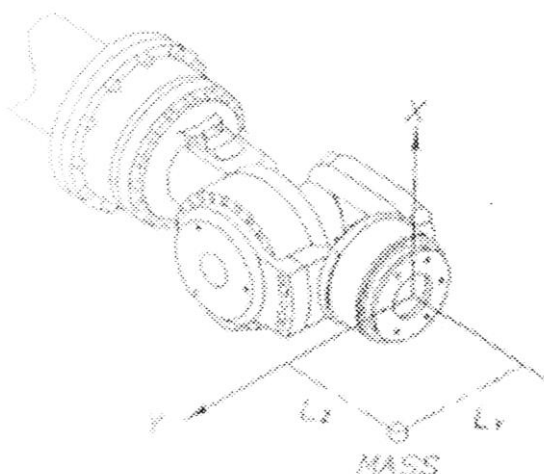
Slika 2 Prikaz kinematične verige robota ACMA ABB XR701

Robot je zgrajen iz šest rotacijskih osi, in sicer:

- 1. os – rotacijsko gibanje za usmeritev podnožja. Vodi ga orientacijski kolut z zunanjimi zobniki, ki so gnani z zobatim kolesom preko reduktorja.
- 2. os – zagotavlja rotacijo vertikalnega člena okoli horizontalne osi, kar orodju omogoča gibe v obliki loka. Gibi so vodeni posredno preko zobniškega reduktorja.
- 3. os – zagotavlja rotacijo horizontalnega člena okrog horizontalne osi. Je na skrajnem koncu drugega člena in omogoča orodju skoraj vertikalne premike. Gibanje reduktorja se prenaša na zgornjo roko indirektno preko prekucnika in gonilnega droga. Uporaba zobatih reduktorjev na prvih treh oseh, ki so najmočnejše, zagotavlja trdnost in izjemno dolgo življenjsko dobo robotov.
- 4. os – zagotavlja rotacijo horizontalne roke okrog njene podolžne osi s pomočjo zelo preproste kinematike, ki sestoji iz cikloidnega reduktorja.
- 5. os – motor se nahaja v roki robota, preko zelo enostavne kinematične verige, sestavljene iz jermena in cikloidnega reduktorja.
- 6. os – transmisija je zelo podobna 5. osi. Poudarili bi, da so kinematične verige zapetja, ki sestoji iz osi 5 in 6, izredno preproste, kar še povečuje visoko stopnjo zanesljivosti in operativne razpoložljivosti. [4]

3.1.2 Vrste zapestja robotov in robota ACMA ABB XR701

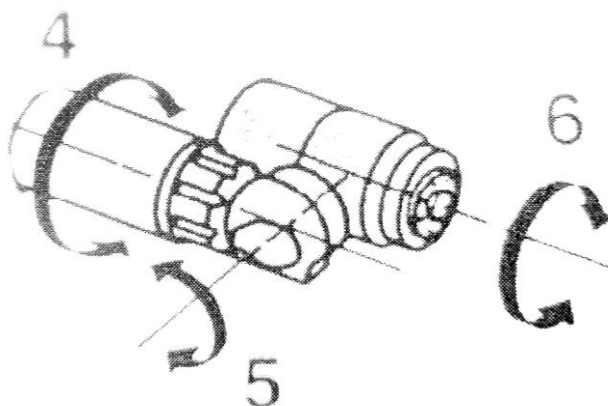
Zapestje RBR (Roll-Bend-Roll (Rotacija-Upogib-Rotacija) osna izvedba Slika 3) robota ACMA ABB XR701 je cenovno ugodno, ki omogoča dobro gibljivost in enostavno napeljavo kablov ter transformacijo koordinat. Edina slabost je, da ima veliko sestavnih delov.



Slika 3 Zapestje RBR osna izvedba

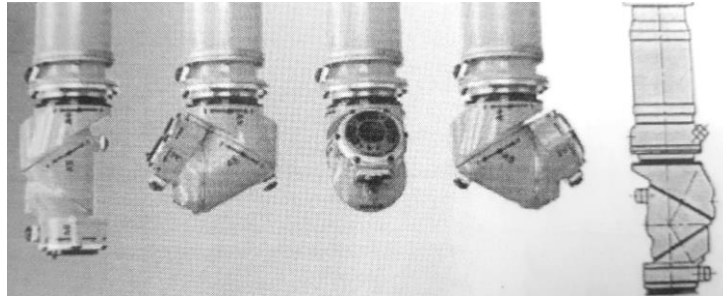
Obstajajo še tudi druge konfiguracije mehanizmov zapestja robota, in sicer:

- zapestje RBR (Roll-Bend-Roll (Rotacija-Upogib-Rotacija) kotna izvedba Slika 4), ki ima tudi dobro gibljivost, malo sestavnih delov in je cenovno zelo ugodno. Zapletena je napeljava kablov in transformacija koordinat.



Slika 4 RBR – kotna izvedba

- Zapestje RRR (Roll-Roll-Roll (Rotacija-Rotacija-Rotacija) dvojna kotna izvedba Slika 5), ki omogoča zelo dobro gibljivost in enostavno napeljavo kablov ter transformacijo koordinat. Vendar je sestavljeno iz zelo veliko sestavnih delov in je cenovno neugodno. [8]



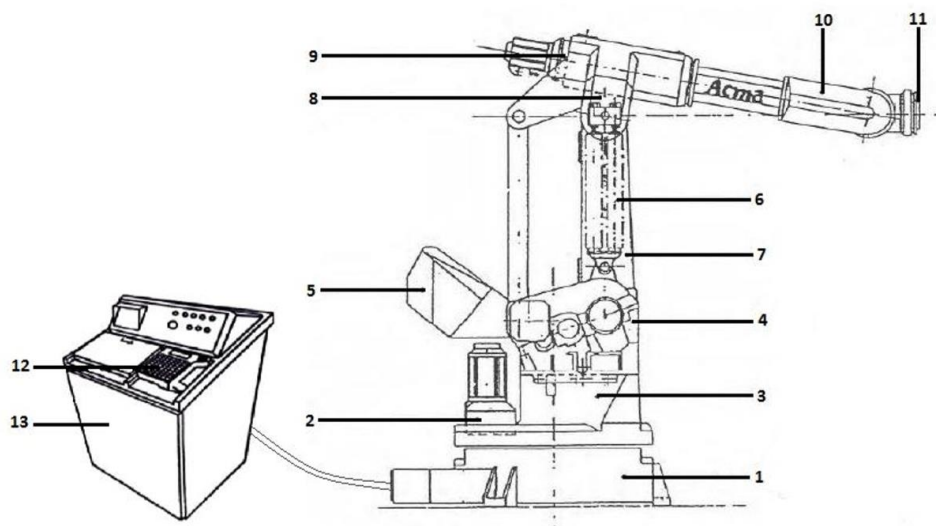
Slika 5 RRR – dvojna kotna izvedba

3.1.3 Glavne značilnosti robota ACMA ABB XR701

Robot je oblike paralelograma (Slika 6) in omogoča večjo nosilnost zaradi večje vztrajnosti in nižjega težišča [1, 3, 8]. Nosilnost robota na 6. osi je od 125 kg (standard) do 150 kg. Zgornja roka je lahko dodatno obremenjena z maso do 160 kg. Osi robota so opremljene s trifaznimi sinhronimi električnimi brezkrtačnimi motorji »Brushless« in omogočajo hitrosti do 3 m/s. Kot omenjeno, je robot zaprtozančno voden in s položajno regulacijo vsak trenutek preverja položaj osi z resolverjem, ki je montiran na motor. V motor je vgrajena elektromagnetna zavora, na izstopni gredi je reduktor, ki zagotavlja večjo ponovljivost oziroma točnost. Zaradi starosti robota in obrabe mehanskih elementov je natančnost robota 0,3 mm [3, 4]. Na Slika 6 je prikazan robot brez omare s krmilnikom APRIL 5000. Glavne karakteristike robota so prikazane v Tabela 1.

Tabela 1 Prikaz karakteristike robota

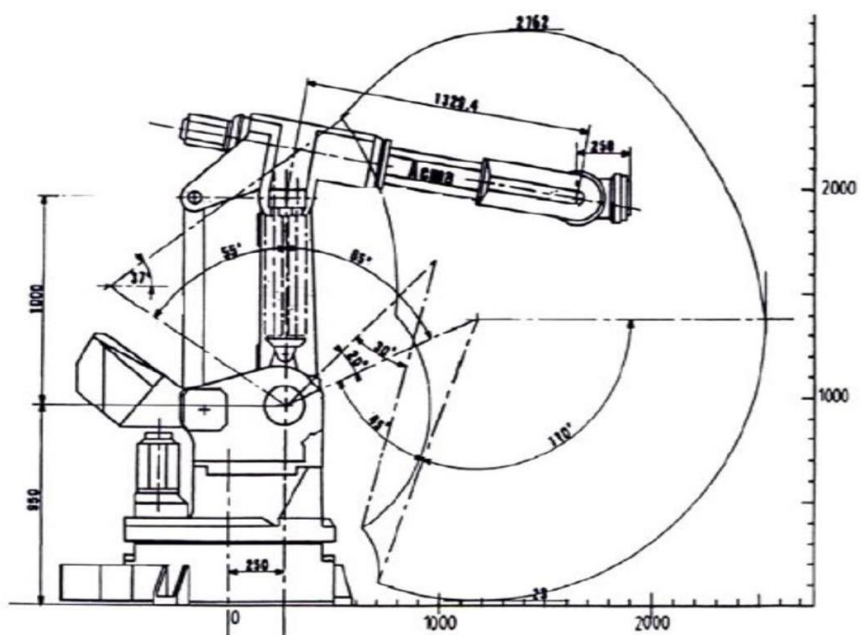
Število osi	Kot gibanja osi	Maksimalna hitrost	Čas dosega maksimalne hitrosti
1	360°	1,9 rd/s	690 ms
2	+65° / -55°	2,2 rd/s	650 ms
3	+110° / -27°	2 rd/s	730 ms
4	400°	3 rd/s	435 ms
5	240°	3 rd/s	435 ms
6	600°	4,2 rd/s	585 ms



Slika 6 Shematski prikaz komponent robota ACMA ABB XR701

Legenda:

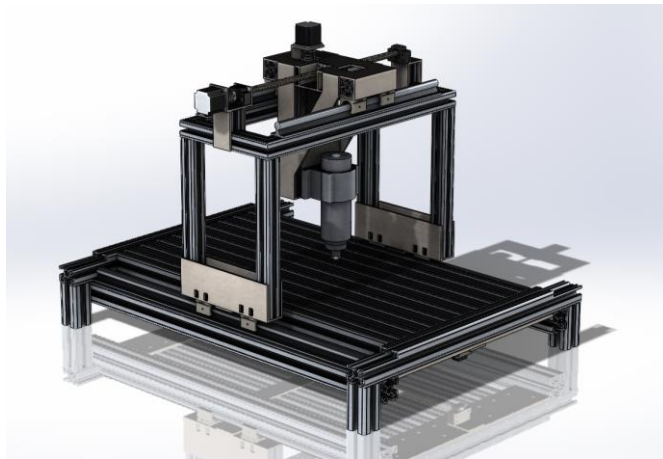
- | | |
|-----------------------------------|---|
| 1. Podstavek robotskega mehanizma | 8. Tretji člen robota (3. os) |
| 2. Motor z gonilom (1. os) | 9. Motor z gonilom (4. os) |
| 3. Vrtljivo podnožje | 10. Sklep drugega in tretjega člena (5. os) |
| 4. Motor z gonilom (2. os) | 11. Pritrdilna prirobnica (6. os) |
| 5. Protiutež | 12. Prenosna konzola |
| 6. Ravnotežna vzmet | 13. Krmilna omara |
| 7. Drugi člen robota | |



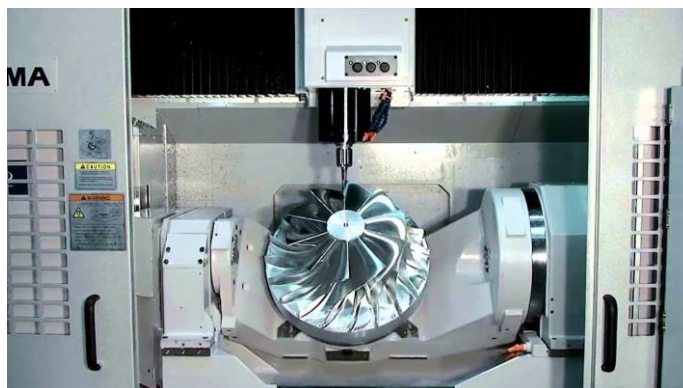
Slika 7 Delovni prostor robota ACMA ABB XR701

3.2 Obdelovalni stroji CNC

CNC-stroj je sestavljen iz dveh delov: mehanizma, s katerim obdelujemo polizdelek, in krmilnika, ki vodi mehanizem. Stroji CNC so vodeni z NC- in G-Kodo, ki opisujeta trajektorijo in način obdelave. Delovni prostor je kartezičen in omogoča montažo obdelovalnega dela nad glavo, kar posledično otežuje vzdrževanje. Zaradi kartezične oblike ima veliko nosilnost in enostavno konstrukcijo. Najbolj razširjen CNC-stroj je 3-osni, vendar se v specializirani maloserijski proizvodnji velikokrat uporabljajo 5-osni (Slika 9) obdelovalni stroji, ki omogočajo obdelavo kompleksnejših izdelkov z večjo natančnostjo.



Slika 8 Prikaz 3-osnega CNC-stroja



Slika 9 Prikaz 5-osnega CNC-stroja

Glavna značilnost stroja CNC je njegova fleksibilnost, saj omogoča hitro zamenjavo programa in polizdelka z manjšimi spremembami nastavitve in preureditve stroja. Posebej primeren je za avtomatizacijo maloserijske in srednjoserijske proizvodnje.

Skozi leta uporabe CNC-strojov se je sistem izboljšal z naslednjo mehansko in programsko opremo:

- brezstopenjsko krmiljenje glavnega vretena;
- programsko vodena avtomatska menjava orodja (revolverska glava s shrambo);
- uporaba krogličnega vretena, ki je drastično povečal natančnost orodja in delovne mize;
- nov optični, magnetni in induktivni merilniki položaja (direktno ali indirektno);
- konstrukcije so bolj toge, zmanjšajo mikrovibracije in povečajo natančnost;
- zaznava kolizije za zmanjšanje nevarnosti poškodb;
- zunanji senzorji za zaznavanje polizdelka (taktilni, optični).

3.2.1 Programiranje NC- in CNC-strojov

Uvajanje NC-strojov v proizvodnjo zahteva nov pristop priprave delovnega procesa za njihovo uporabo. Obseg tehnološkega postopka obdelave se močno poveča pri modernejših NC-strojih, ki morajo izpolnjevati tudi specifične zahteve uporabnika. Zaradi sodobnih NC-strojov je izdelava ekonomična in ponovljiva. Zahtevnost trga zahteva vedno kompleksnejše oblike polizdelkov ali končnih izdelkov, kar drastično podaljša načrtovanje obdelave (programiranje) in že danes obsegajo več kot 1/3 stroškov proizvodnje. V ta namen se razvijajo racionalnejše oblike izdelave programa za vodenje strojev, kot so na primer vmesniki oziroma namenske programske opreme za enostavnejšo in hitrejšo uporabo. Z razvojem CAD/CAM-sistema se je programiranje poenostavilo, saj omogoča hitro in učinkovito programiranje izdelkov ter dokumentacijo izdelanih programov in izdelkov. Temu primerno se je povečala tudi cena uporabe, kar v mnogih primerih pomeni, da je za enostavnejše kosovne proizvodnje bolj ekonomično ročno NC-programiranje. [5]

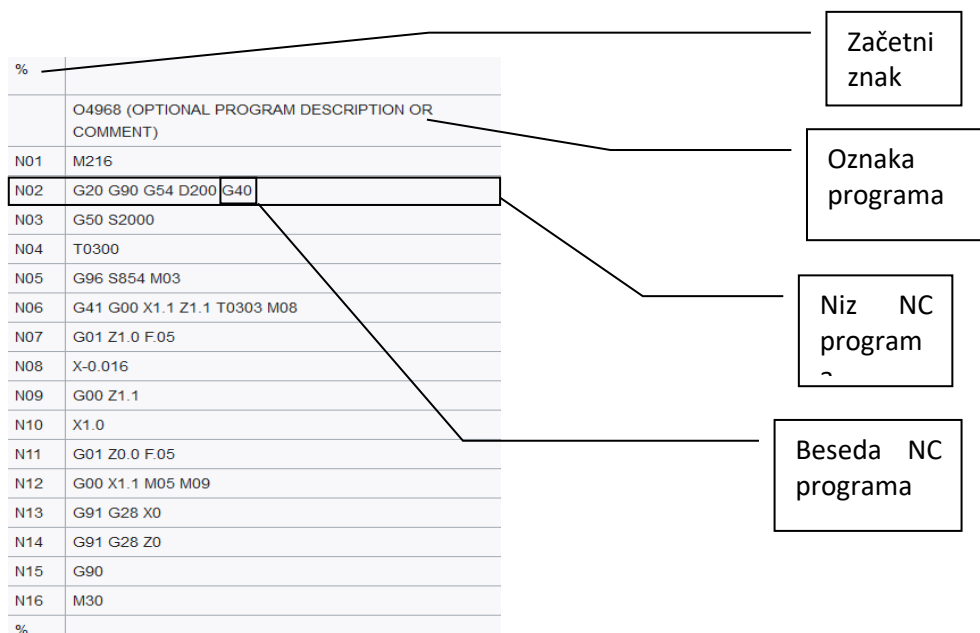
Za pretvorbo CL-podatkov v NC- ali G-Kodo se najpogosteje uporabljajo naslednje programske opreme:

- CATIA,
- Creo,
- Edgecam,
- Mastercam,
- NX (Unigraphics).

3.2.2 NC-program ali G-Koda

G-Koda (NC-program) ali RS-274 (Slika 10) je najbolj razširjen in uporabljen programski jezik za numerično krmiljenje. Uporablja se predvsem za programiranje strojev NC in CNC s CAM-programsko opremo ali z ročnim programiranjem.

Sestavljena je iz nizov ukazov, ki jih krmilnik izvaja zaporedno. Za vsak ukaz je opisana trajektorija do naslednje točke. Krmilnik izračuna trajektorijo z linearnim ali krožnim interpolatorjem, odvisno od vrste ukaza. Pomik končnih elementov se preverja z zaprtozančno regulacijo položaja do željene pozicije po trajektoriji. Programska oprema CAM izdelava G-Kodo z manj zahtevnejšimi ukazi, ki posledično zavzema več prostora. Z ročnimi ukazi se lahko napiše bolj kompleksen program, ki razume tudi funkcije, kot so na primer zanka, ponavljanje, pogojni stavki in drugo.



Slika 10 Zgradba CNC-programa

G-Kodo imenujemo tudi pripravljalna koda in se navezuje na besedo v programu CNC, ki se začne s črko G. Na splošno je to koda, ki strojnemu orodju pove, katero vrsto dejanja naj izvede, na primer:

- hiter premik (orodje giblje v najkrajšem času do obdelovalnega mesta);
- krmiljeno krmilo se premika po ravni črti ali loku;
- serija nadzorovanih premikov podajanja, ki bi povzročila vrtanje luknje, usmerjen rez obdelovanca na določeno dimenzijo ali izdelava profila (konture) na rob obdelovanca;
- nastavitev informacije o orodju, na primer odmik;
- preklop koordinatnega sistema.

G-Koda se je z leti izpopolnila in omogoča uporabo ukazov za opravljanje raznovrstnih funkcij in operacij. Za nas uporabne so bile funkcije, ki so prikazane odebeljeno:

- **G00 hitri gib;**
- **G01 linearna interpolacija;**
- **G02 krožna interpolacija v urini smeri;**
- **G03 krožna interpolacija v proti-urini smeri;**
- G04 prekinitev priprave naslednjih NC-stavkov;
- G16 izbira in opredelitev delovne ravnine;

- G17 delovna ravnina x-y;
- G18 delovna ravnina z-x;
- G19 delovna ravnina y-z;
- G21 programiranje v milimetrih;
- G33 rezanje navojev;
- G40 korekcija radija orodja – začetno stanje / preklic korekcije;
- G41 korekcija radija orodja – levo;
- G42 korekcija radija orodja – desno;
- G43 kompenzacija dolžine orodja;
- G44 preklic kompenzacije dolžine orodja;
- G51 branje kode naprej;
- G53 programiranje koordinat z ozirom na strojne ničelne točke;
- G54 premik ničelne točke 1;
- G55 premik ničelne točke 2;
- G56 premik ničelne točke 3;
- G57 premik ničelne točke 4;
- G58 dodatni premik ničelne točke 1;
- G59 dodatni premik ničelne točke 2;
- G74 iskanje referenčne točke;
- G75 iskanje referenčne točke;
- G76 iskanje referenčne točke;
- G80 preklic strojnega cikla;
- G81 vrtalni cikel;
- G82 vrtalni cikel z zastojem;
- G83 enostaven cikel globokega vrtanja;
- G84 cikel rezanja navoja;
- G85 cikel povrtavanja;
- G86 cikel za izvrtavanje s hitrim umikom;
- G89 cikel za izvrtavanje z delovnim umikom;
- G90 absolutni način programiranja;
- G91 inkrementalni način programiranja;
- G92 premik koordinatnega sistema / nastavljanje meje vrtilne hitrosti;
- G94 podajanje v milimetrih na minuto;
- G95 podajanje v milimetrih na vrtljaj;
- G96 konstantna rezalna hitrost;
- G97 konstantna hitrost središča orodja;
- G98 umik do začetne ravnine;
- G99 umik do referenčne ravnine.

3.3 Primerjava CNC-stroja z robotom

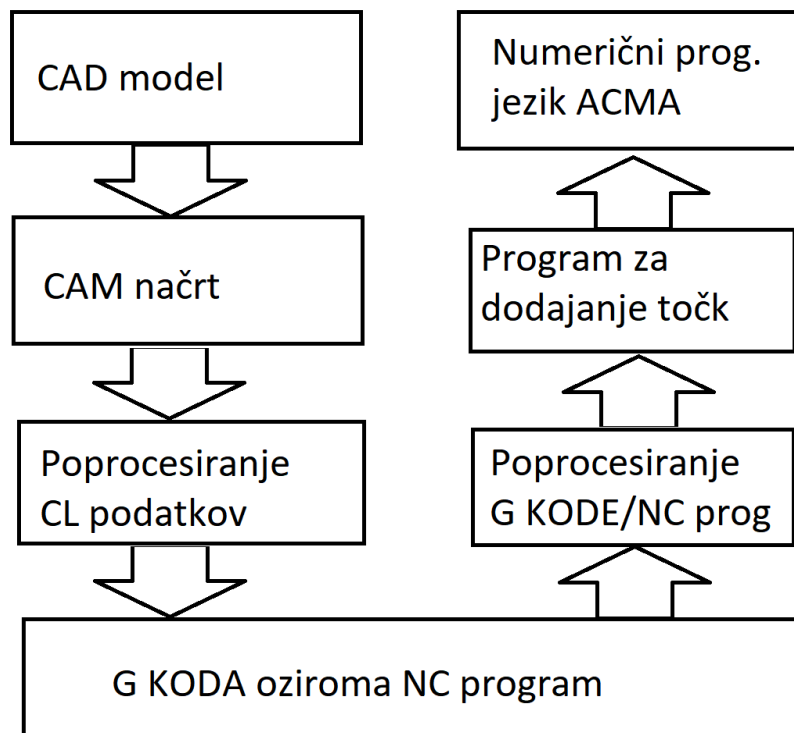
Poglavje 3.1 in 3.2 opisuje industrijski robot ACMA ABB XR701 in obdelovalne stroje CNC. »Z vidika avtomatizacije je obdelovalni stroj CNC avtomatiziran sistem, ki samostojno obdeluje polizdelek po zapisani ISO G-kodi ali programu NC.« [3] Preprosti obdelovalni stroji (3-osni kartezični CNC stroj) imajo tri translacijske člene, ki omogočajo translacije v smeri x, y in z v ortonormiranem koordinatnem sistemu. Z njimi težko obdelujejo kompleksnejše oblike, razen v primeru uporabe kroglastega rezkarja. Za obdelavo kompleksnejših oblik se je razvoj osredotočil na dodajanje dodatnih rotacijskih osi CNC-stroju. Kartezična oblika stroja CNC omogoča togo konstrukcijo, ki zmanjša mikrovibracije. Elementi prvih treh prostostnih stopenj stroja CNC z največjo maso zaradi enostavnih pretvornih členov iz rotacijskega gibanja motorja v translacijsko gibanje preko kroglastega vretena zagotavljajo dolgotrajno ponovljivost in natančnost ter manjšo obrabo. V primerjavi s CNC-strojem ima robot zaradi antropomorfne oblike (odprta kinematična veriga) in mehanskih lastnosti večjo vztrajnost. S tem povzroča, da se pri veliki obremenitvi in hitrim pomikom pojavijo mikrovibracije in deformacije konstrukcije, kar vpliva na natančnost giba. Robotske osi so po navadi rotacijske, odvisno od konfiguracije robota. Manipulirajo z večjimi masami in imajo zaradi reduktorjev večjo obrabo kot CNC-stroji [6]. »Njihova slabost je majhen delovni prostor. Za večji delovni prostor je treba povečati konstrukcijo stroja in izbrati material in obliko elementov konstrukcije, ki so odporni na upogib za doseganje dobre natančnosti. Zaradi pravokotnega delovnega prostora in konstrukcije prav tako težko opravlja zahtevnejše obdelovalne položaje, s čimer pa robot nima težav, saj je delovni prostor, po katerem ga lahko poljubno vodimo, sferičen in dostopen s poljubno postavitvijo robota.« [3]

3.4 Poprocesor za pretvorbo G-kode

V industriji se vedno pogosteje pojavlja obdelava površine poljubne oblike, kar zahteva uporabo specialnih obdelovalnih strojev, omenjenih v poglavju 3.2, in tehnoloških postopkov, omenjenih v poglavju 3.2.1. Za takšne obdelave je ročno programiranje nesmiselno. V ta namen se uporablja programska oprema za pretvorbo modela CAD v ISO G-kodo ali program NC 3/5-osnega obdelovalnega stroja. Za pretvorbo uporabimo namenski program CAM za poprocesiranje neposredno pridobljenih CL-lokacij iz modela CAD v tehnološki postopek obdelave. [3]

Pri pretvorbi CAD in CAM v G-Kodo večkrat pride do nekompatibilnosti tehnološkega postopka s strojem NC, CNC, robota ali drugih strojev, zato je treba ustvariti unikaten poprocesor oziroma vmesnik za pretvorbo podatkov CL glede na uporabljen stroj, orodje in tehnološki proces v ustrezen numerični strojni jezik. [6]

V zaključnem delu smo to izvedli z implementacijo poprocesorja, ki pretvori G-Kodo v točke za učenje robota ACMA. Zaradi obsežnosti pretvorbe uporabimo programsko opremo CAM, ki iz modela CAD pripravi G-Kodo glede na uporabljeno orodje in konfiguracijo stroja. Če se osredotočimo na poprocesiranje G-Kode v jezik za učenje robota ACMA, je prvi korak, da vhodno kodo razčlenimo in po potrebi s pomočjo našega krožnega interpolatorja izračunamo dodatne točke gibov. Izhodno kodo nato primerno strukturiramo in ta predstavlja program za učenje točk. Po obdelavi program za učenje točk ponovno pretvori kodo v numerično robotsko kodo ACMA. Končna pretvorba nato te točke s specializiranim programom za učenje točk pretvori v numerični strojni jezik ACMA. Proces pretvorbe prikazuje Slika 11.



Slika 11 Prikaz pretoka podatkov

4 RAZVOJ POPROCESORJA

Razvoj poprocesorja za pretvorbo 3- ali 5-osnega NC-programa ali G-kode v robotski jezik ACMA je sestavljen iz treh faz:

- kreiranja uporabniškega vmesnika;
- pretvorbe G-kode v program robota ACMA;
- evaluacije.

4.1 Uporabniški vmesnik

Moderni svet se vedno bolj izpopolnjuje, kar je vidno pri kompleksnosti stroja in programov, ki operaterjem povzročajo vedno večje težave. Po navadi jih uporabljajo manj izobraženi kadri ali starejši uporabniki, zato se za pohitritev in zmanjševanje napak pri krmiljenju strojev CNC in robotov uporablja namenski uporabniški vmesnik. V našem primeru smo izdelali dodatni uporabniški vmesnik, predstavljen kot grafični uporabniški vmesnik GUI (angl. Graphical User Interface, slo. grafični uporabniški vmesnik), ki natančno opredeli nastavljanje parametrov stroja in pretvarjanje NC-programa ali G-Kode ter s tem olajša uporabo robota. V našem primeru smo razvili GUI, preko katerega lahko nastavimo položaj polizdelka, konfiguracijo robota in druge manj specifične nastavitve., na primer natančnost robota in nastavitve natančnosti števil s plavajočo vejico, kjer nastavimo število decimalnih mest koordinat. API (angl. Application Programming Interface, slo. aplikacijski programski vmesnik) je poskrbel za pridobivanje vhodne G-Kode, nastavljanje in izvedbo pretvorbe, strukturiranje izhodnega numeričnega programa ACMA in shranjevanje pridobljenega programa.

Vmesnik smo razvili v programskem okolju Visual studio 2017 s pomočjo programskega paketa Windows Forms C++/CLI in izdelali GUI. C++/CLI je jezikovna specifikacija. Razvil jo je Microsoft in nadomešča upravljalne (angl. managed) razširitve za C++, ki se uporablja za poenostavitev starejše upravljalne sintakse C++, ki je zdaj

opuščena. C++/CLI je standardizirana v standard ECMA-372. Trenutno je na voljo kot programski paket za Visual Studio 2005–2017, vključno z izdajami Express. [3, 10, 14]

Za izdelavo GUI smo uporabili orodja:

- za izdelavo zavihkov in njihovo izbiro (angl. TabControl, slo. nadzor zavihkov);
- za vpisovanje podatkov (angl. TextBox, slo. vnosno polje);
- za opisovanje vnosnih polj in oken (angl. Label, slo. oznaka);
- za vnos in izpis besedila oziroma G-Kode ali ACMA-programa (angl. RichTextBox, slo. bogato vnosno polje);
- za nalaganje G-Kode v GUI (angl. LoadFileDialog, slo. okno za nalaganje iz datoteke);
- za shranjevanje programa ACMA iz GUI (angl. SaveFileDialog, slo. okno za shranjevanje v datoteko);
- za izbiro vrednosti (angl. RadioButton, slo. izbirni gumb);
- za zagon operacije (angl. Button, slo. gumb).

Uporabniški vmesnik je zgrajen iz več zavihkov, ki omogočajo:

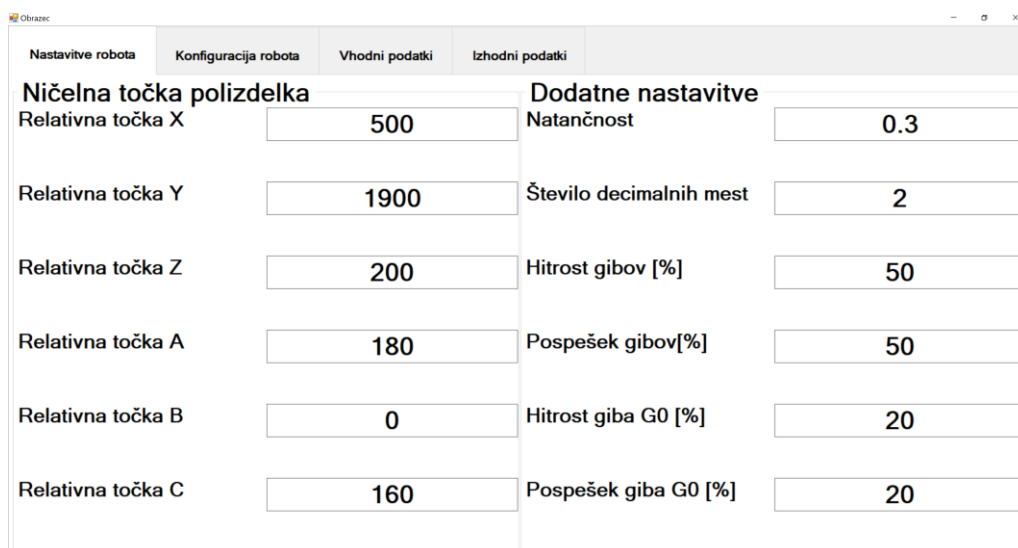
- nastavljanje robota;
- konfiguracijo robota;
- nalaganje in prikaz vhodne G-Kode;
- pretvorbo, prikaz in shranjevanje izhodnega programa ACMA za dodajanje točk.

4.1.1 Nastavitev robota

Ta zavihek (Slika 12) smo načrtovali za nastavitev ničelne točke polizdelka, katerega bomo obdelovali, in za določitev dodatnih nastavitev.

Dodatne nastavitve so:

- natančnost robota, ki določa gostoto oziroma število interpolacij točk krožnega giba;
- število decimalnih mest, izračunanih oziroma pretvorjenih koordinat;
- privzeta hitrost in pospešek gibov G0 hitrega giba in G1 delovnega giba.



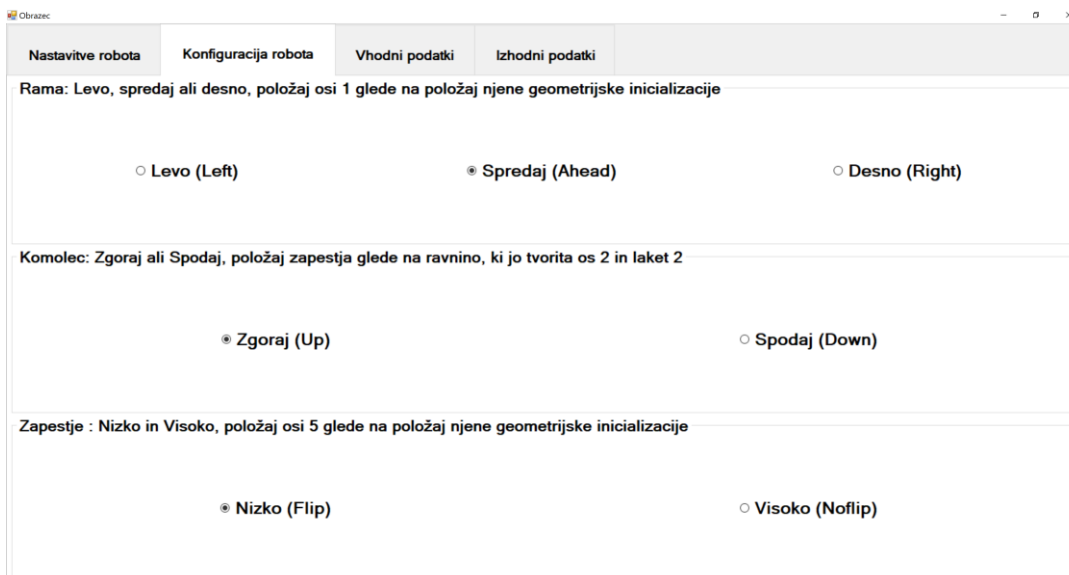
Ničelna točka polizdelka		Dodatne nastavitve	
Relativna točka X	500	Natančnost	0.3
Relativna točka Y	1900	Število decimalnih mest	2
Relativna točka Z	200	Hitrost gibov [%]	50
Relativna točka A	180	Pospešek gibov[%]	50
Relativna točka B	0	Hitrost giba G0 [%]	20
Relativna točka C	160	Pospešek giba G0 [%]	20

Slika 12 Okno za nastavitev robota

4.1.2 Konfiguracija robota

V tem oknu nastavimo konfiguracijo robota s pomočjo izbirnih gumbov (Slika 13), in sicer:

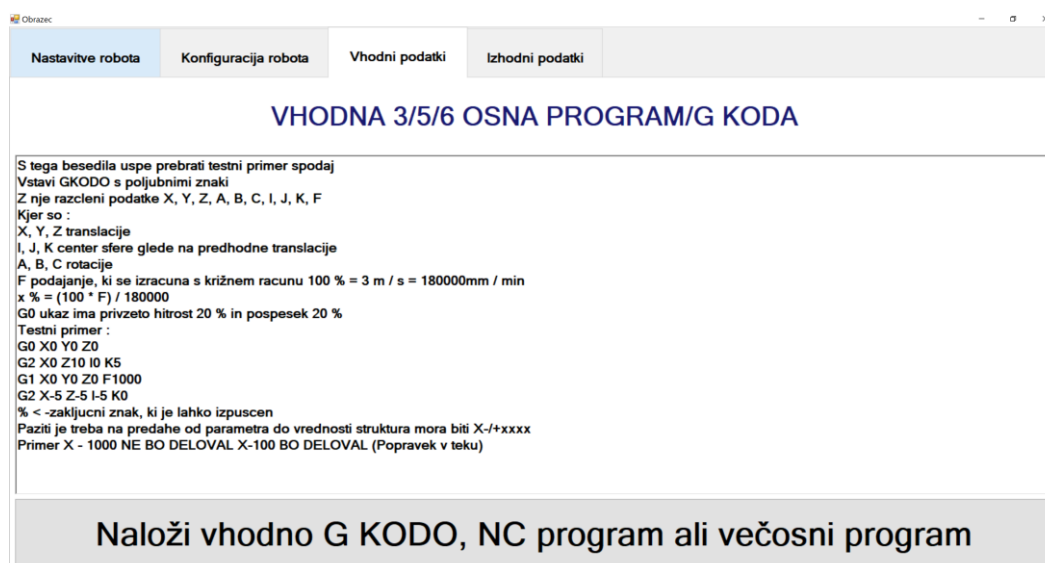
- rame prvega člena (podnožja), če se bo robot nahajal v levem, sprednjem ali desnem položaju;
- komolca horizontalne roke, če bo v zgornjem ali spodnjem položaju;
- zapestja osi 5, če bo obrnjena navzgor ali navzdol;



Slika 13 Okno za konfiguracijo robota

4.1.3 Vhodni podatki

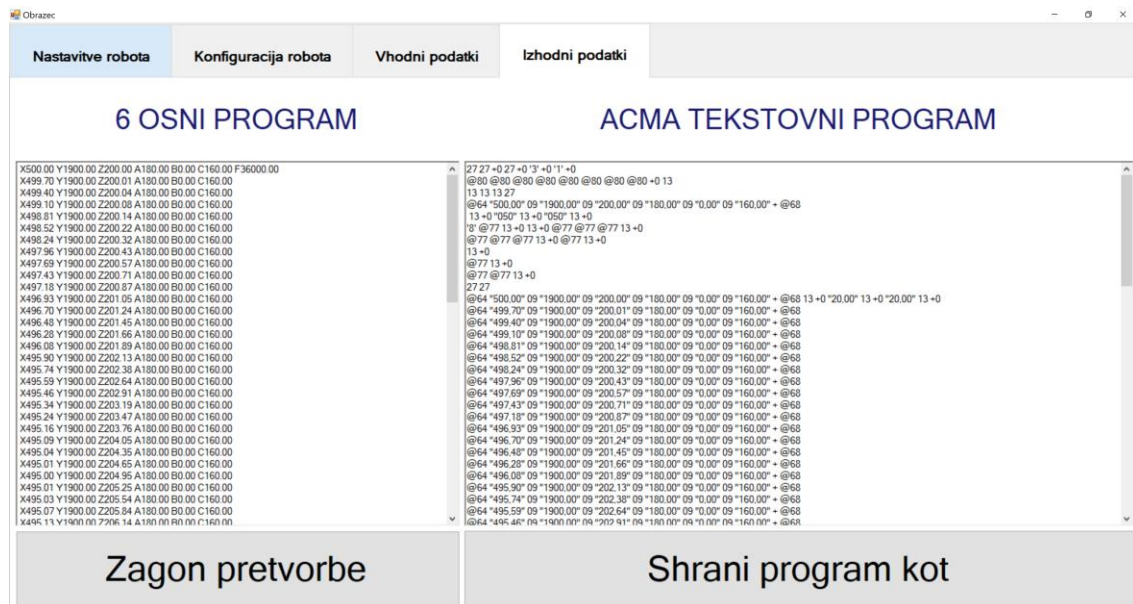
Za zagon pretvorbe je treba iz datoteke naložiti G-Kodo, kar izvedemo s klikom na gumb. Iz okna za nalaganje nato izberemo ustrezen format datoteke, kjer se nahaja tekstovna koda z G-ukazi. Po izbiri se v bogato tekstovno polje izpiše izbrana datoteka, kjer je bilo predhodno privzeto besedilo s kratkim opisom delovanja (Slika 14).



Slika 14 Okno za nalaganje vhodne G-Kode

4.1.4 Izhodni podatki

Za preglednost smo v tem oknu prikazali v bogatem tekstovnem polju razčlenjen in izračunan 6-osni program in strukturiran ACMA-tekstovni program. Za prikaz je treba zagnati pretvorbo s klikom na gumb »Zagon pretvorbe«. ACMA-tekstovni program lahko z gumbom »Shrani program kot« shranimo z oknom za shranjevanje v datoteko.



Slika 15 Okno za upravljanje z izhodnimi podatki

4.2 Pretvorba G-kode v program robota ACMA

Program temelji na uporabi standardnih knjižnic, ki olajšajo delo s pisanjem funkcij. Knjižnica ima zbirko razredov in funkcij, ki so napisane v baznem jeziku C++. Za pretvorbo G-Kode smo napisali tudi lastne razrede z javnimi in zasebnimi metodami. Zasebne metode so uporabljene znotraj razreda, medtem ko lahko javne metode uporabljamo v celotnem programu, kjer imamo vključen razred. Program ima ustvarjen razred »Pridobitev_Koordinat«, kjer se izvaja celotna pretvorba.

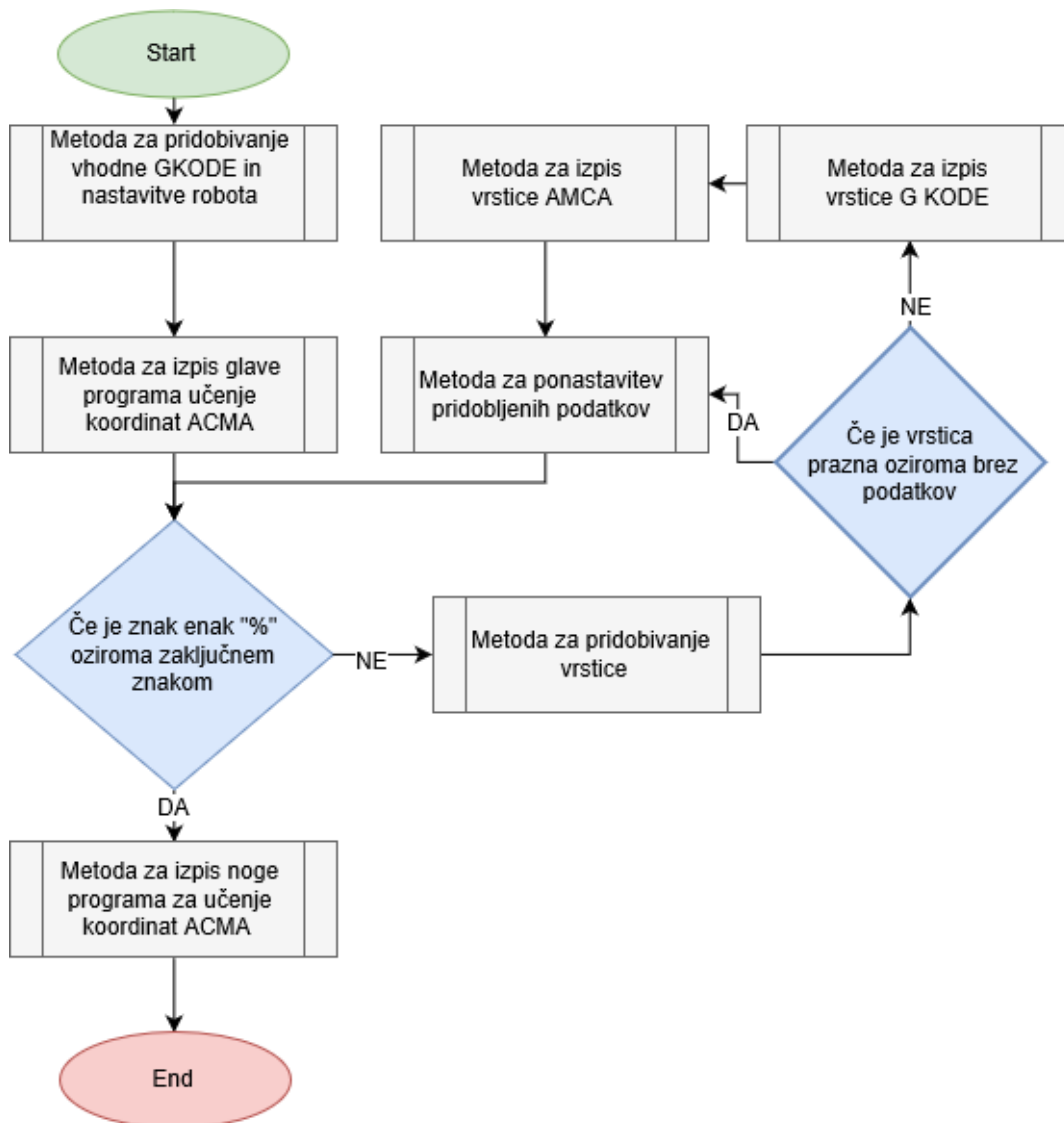
Razred vsebuje javne metode za arhiviranje in upravljanje s spremenljivkami, pridobljenih iz vhodne G-kode in nastavitvenih podatkov iz GUI. V razredu ima tudi

zasebne metode, ki se uporabljajo za razčlenjevanje vhodne G-Kode, transformacijo koordinatnega sistema vhodne G-Kode in preračunavanje interpolacij krožnih gibov ukaza G2 in G3.

Aplikacijski vmesnik "Pretvori vhodno G-kodo" je predstavljen s Pseudokod 1 oziroma z diagramom poteka na Slika 16. Glavni aplikacijski vmesnik deklarira razred in vse potrebne spremenljivke, ki so uporabljene za pretvorbo. Spremenljivke, ki so zapisane z velikimi tiskanimi črkami, predstavljajo sistemski niz, ki se uporabi za izpis v vmesnik GUI. API vsebuje tudi klice javnih metod, katerih ime je ločeno s podčrtaji.

Pseudokod 1 Glavni aplikacijski vmesnik za pretvorbo G-kode

```
1: String^ Acmalzpis, Gkodalzpis;
2: Pridobitev_Koordinat Koord;
3: Koord.Vhodni_Podatki(VHGKODA, NATANC, RELX,
  RELY, RELZ, RELA, RELB, RELC, STDEC, V, A,
  RAMA, KOMOLEC, ZAPESTJE);
4: OUTACMA->Clear();
5: OUTGKODA->Clear();
6: Acmalzpis = Koord.Izpis_Glave_Prog_ACMA();
7: OUTAACMA->AppendText(Acmalzpis);
8: Koord.Clear_Data();
9: while Koord.Zakljucni_Znak() is not true do
10:   Koord.Pridobitev_Vrstice();
11:   if Koord.Prazna_Vrstica() is not true then
12:     Acmalzpis = Koord.Izpis_Koord_Prog_ACMA();
13:     Gkodalzpis = Koord.Izpis_Vrstice_Gkode();
14:     Koord.Prepis_Podatkov();
15:     OUTGKODA->AppendText(Gkodalzpis);
16:     OUTAMCA->AppendText(Acmalzpis);
17:     Koord.Clear_Data();
18:   end if
19: end while
20: Acmalzpis = Koord.Izpis_Noge_Prog_ACMA();
21: OUTACMA->AppendText(Acmalzpis);
22: Koord.Clear_Data();
```



Slika 16 Diagram poteka glavnega API za pretvorbo G-Kode

Razred z zasebnimi metodami po klicu javne metode pridobivanje vrstice samodejno izvede celoten proces, ki je uporabljen v naslednjih delih. Omenjeni GUI sestoji iz treh glavnih delov:

- zajema podatkov in strukturiranje numeričnega programa ACMA za učenje točk;
- razčlenjevanja podatkov in preverjanje zapognjenosti vrstice;
- pretvorbe krožnih izsekov v translacije, kjer je to potrebno.

4.2.1 Zajem podatkov in struktura numeričnega programa za učenje točk robota ACMA XR701

Za programiranje robota ACMA je treba v niz zapisati določeno zaporedje znakov, ki upravljajo program za dodajanje točk linearnih gibov oziroma translacij, predstavljenih v Tabela 2. Za program moramo strukturirati glavo za določanje načina vodenja robota in osnovnih nastavitvev, vsebino točk in nogo za zaključek oziroma ustavitvev programa. Ustvarjeno tekstovno datoteko s sintakso nato zapišemo na medij (magnetno zapisana zgoščanka), ki je namenjen upravljanju robota.

Tabela 2 Vrednosti tipk konzole za programiranje robota ACMA

Funkcija	Tekstovni znak
Izbira menija	'x'
Vrednost	"x"
Tab	09
Enter	13
Esc	27
F1	@59
F6	@64
F9	@67
F10	@68
Tipka desno	@77

Vhodne podatke, ničelno točko polizdelka, postavitev robota, točke gibov oziroma translacij in druge manj pomembne podatke z GUI arhiviramo v razred »Pridobitev_Koordinat« z javno metodo »Vhodni_Podatki«. Za izpis uporabimo javne metode:

- Izpis_Glave_Programa_ACMA (prikazano v prilogi);
- Izpis_Koordinat_Programa_ACMA (prikazano v prilogi);
- Izpis_Vrstice_Gkode (prikazano v prilogi);
- Izpis_Noge_Programa_ACMA (prikazano v prilogi).

Preden izpisujemo podatke, jih je treba z zasebnimi metodami razčleniti in pretvoriti, če je to potrebno, kar opisujeta poglavji 4.2.2 in 4.2.3. Niz znakov polnimo v niz glede na strukturo opredeljeno in po zaporedju, prikazanemu na slikah Slika 17, Slika 18, Slika 20 in Slika 21 ter Pseudokod 2., ki ga nato izpiše v bogato tekstovno polje. Za ponovno pretvorbo in izpis smo z javno metodo »Clear_Data« počistili spremenljivki za izpisovanje izhodne G-kode in programa ACMA za učenje točk ter priredili spremenljivkam znotraj razreda vrednost 0 in vrednosti, ki določajo prazno vrstico, prikazano v diagramu poteka. [3]

V glavo strukture vpišemo ničelno točko, nastavimo način vodenja robota in konfiguracijo robota, kar prikazujejo Slika 17 in Slika 18 ter Pseudokod 2.

```
27 27 +0 27 +0 '3' +0 '1' +0
@80 @80 @80 @80 @80 @80 @80 @80 +0 13
13 13 13 27
@64 "x" "09" "x" "09" "x" "09" "x" "09" "x" "09" "x" +@68
```

Slika 17 Sekvenca za nastavitve ničelne točke

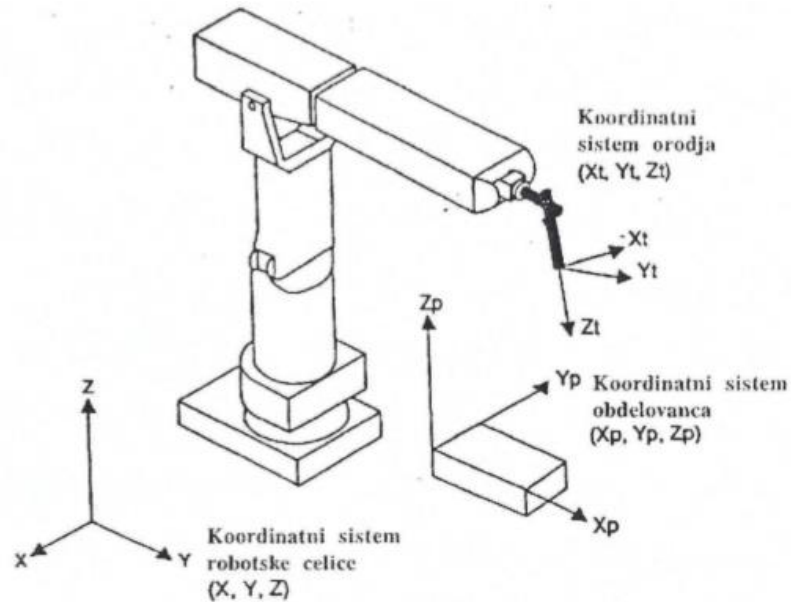
```
'8' @77 13 +0 13 +0 @77 @77 @77 13 +0
```

Slika 18 Sekvenca za nastavitve načina dodajanja koordinat

Pseudokod 2 Pseudokoda za določanje sekvence za nastavitve konfiguracije robota

```
1: if Rama is Levo then
2: Acmalzhod.append("13 +0 13 +0nn");
3: else if Rama is Spredaj then
4: Acmalzhod.append("13 +0nn");
5: end if
6: if Komolec is Zgoraj then
7: Acmalzhod.append("@77 13 +0nn");
8: end if
9: if Zapestje is Nizko then
10: Acmalzhod.append("@77 @77 13 +0nn");
11: end if
```

Za določitev konfiguracije je treba transformirati tudi koordinatni sistem. G-Koda ima koordinatni sistem CNC-stroja. Pri normalni konfiguraciji (Rama->levo, Komolec->zgoraj in Zapestje->nizko) koordinatni sistem robotske celice transformira os x za 180° in os z za 180° v koordinatni sistem orodja, kar je prikazano na Slika 19.



Slika 19 Koordinatni sistemi robota

Vsebina programa ACMA je sestavljena z vpisovanjem točk in poteka na enak način kot dodajanje ničelne točke. Pri dodajanju točk dodamo še hitrost in pospešek giba, kar je prikazano na Slika 20.

```
@64 "X"09"Y"09"Z"09"A"09"B"09"C" +@68
13 +0 "v" 13 +0 "a" 13 +0
```

Slika 20 Sekvenca za določanje točk, hitrosti in pospeška

Noga zaključi program s sekvenco, prikazano na Slika 21.

```
27 27 27 27 27 +0 @59
```

Slika 21 Sekvenca noge programa ACMA za dodajanje točk

4.2.2 Razčlenjevanje podatkov in preverjanje zapolnjenost vrstice

»Zamenjava obdelave polizdelka v industriji zahteva predznanje, saj operater težko razloči in uporablja G-kodo različnih struktur. Za enostavnejšo uporabo smo strukturirali metodo za razčlenitev ključnih podatkov iz poljubnih G-kod. Za razčlenjevanje podatkov niza smo napisali javno metodo »Pridobitev_Vrstice« (Slika 22 Diagram poteka za razčlenjevanje G-Kode), ki preverja znake vrstice do ukaza G, nato se preverjanje nadaljuje do parametrov X, Y, Z, A, B, C, I, J in K, kar je prikazano v Pseudokod 3 Razčlenjevanje vhodne G-kode. Za vsak parameter se z zasebno metodo »Pridobi_Vrednost« (Slika 22 Diagram poteka za razčlenjevanje G-Kode) zapisujejo znaki v medpomnilnik, dokler so enaki številom, piki ali minusom. Metoda vrne medpomnilnik, ki se pretvori v tip (long double) s pomočjo funkcije »stof« (angl. string to float, slo. niz v 16-bitno realno število), prikazano v Pseudokod 4.« [3]

Pseudokod 3 Razčlenjevanje vhodne G-kode

```
1: while TrenutniZnak is not \n do
2:   if TrenutniZnak is G and (NaslednjiZnak is Stevilo 0 - 3) then
3:     Nastavi na naslednji znak;
4:     Vrstica[0] = Trenutna vrednost;
5:     UkazInicializiran = true;
6:     UkazVpisan[0] = true;
7:     Zmanjšaj StPraznihPolj;
8:   end if
9:   if TrenutniZnak is X and Naslednjiznak is Stevilo
   and UkazVpisan[0] is true and UkazInicializiran is true then
10:     Nastavi na naslednji znak;
11:     Vrstica[1] = Pridobi_Vrednost();
12:     UkazVpisan[1] = true;
13:     Zmanjšaj StPraznihPolj;
14:   end if
15:   //Ponovitev za parametre Y, Z, A, B, C, I, J, K
16:   Nastavi na naslednji znak;
17: end while
```

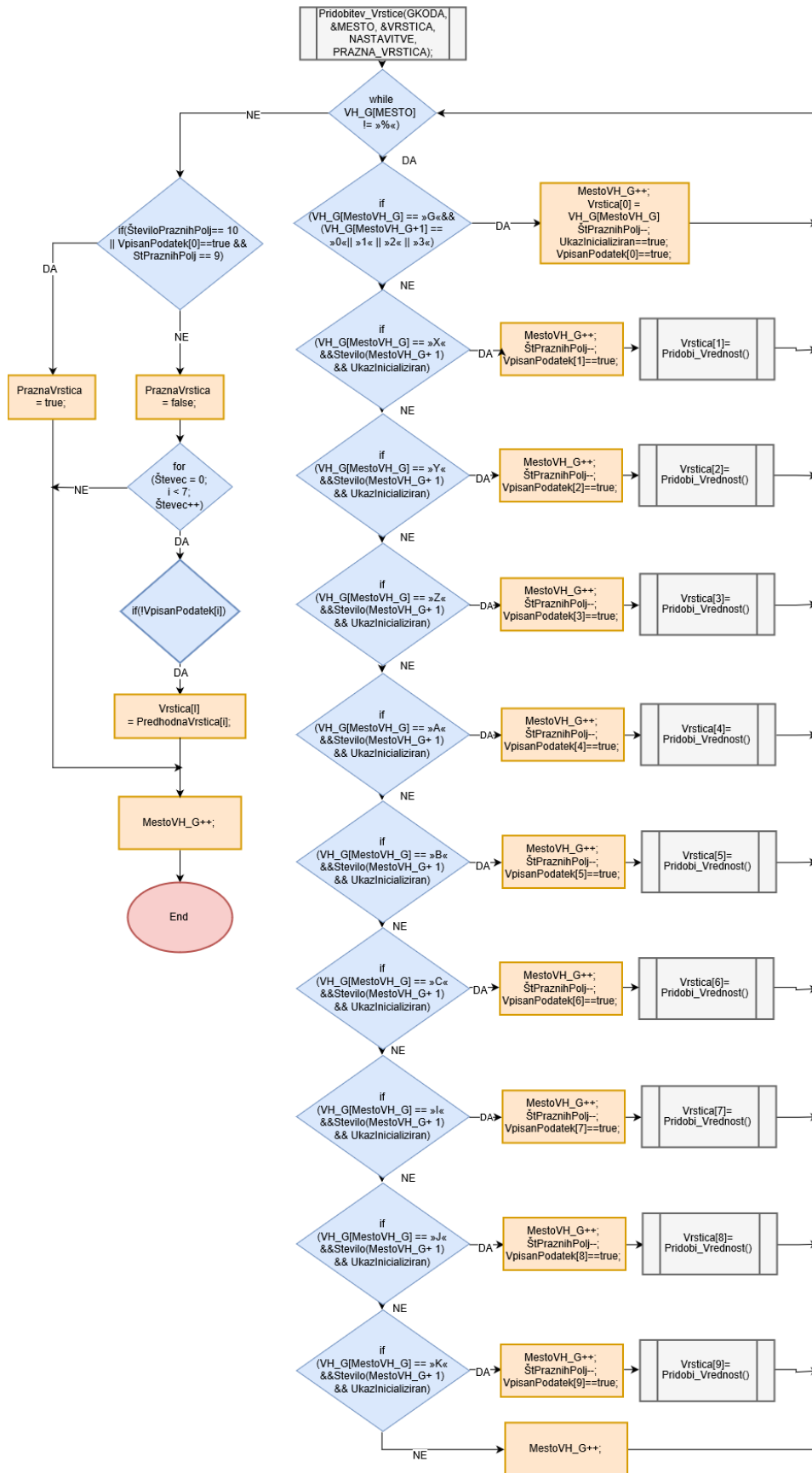

Pseudokod 4 Pridobivanje vrednosti parametrov

```
1: string Medpomnilnik;  
2: while TrenutniZnak is Stevilo do  
3:     Medpomnilnik += TrenutniZnak;  
4:     Nastavi na naslednji znak;  
5: end while  
6: Nastavi na prejšnji znak;  
7: return stof(Medpomnilnik);
```

»Za določanje prazne vrstice preverimo polje binarnih vrednosti »UkazVpisan« in spremenljivko »StPraznihPolj« pridobljenih (Pseudokod 3). Če pogoji, ki so predstavljeni v Pseudokod 5, veljajo, vrstico preskočimo oziroma je ne izpišemo. To je prikazano v Pseudokod 1. Z javno metodo »Prazna_Vrstica« dostopamo do te vrednosti, ki vrne dvojiško vrednost.« [3]

Pseudokod 5 Preverjanje zapolnjenost vrstice

```
1: if StPraznihPolj is 10 or (VpisanPodatek[0] is true and StPraznihPolj is 9) then  
2:     PraznaVrstica = true;  
3: else  
4:     PraznaVrstica = false;  
5:     for  $i \in 0, \dots, 6$  do  
6:         if VpisanPodatek[ $i$ ] is false then  
7:             Vrstica[ $i$ ] = PredhodnaVrstica[ $i$ ];  
8:         end if  
9:     end for  
10: end if
```

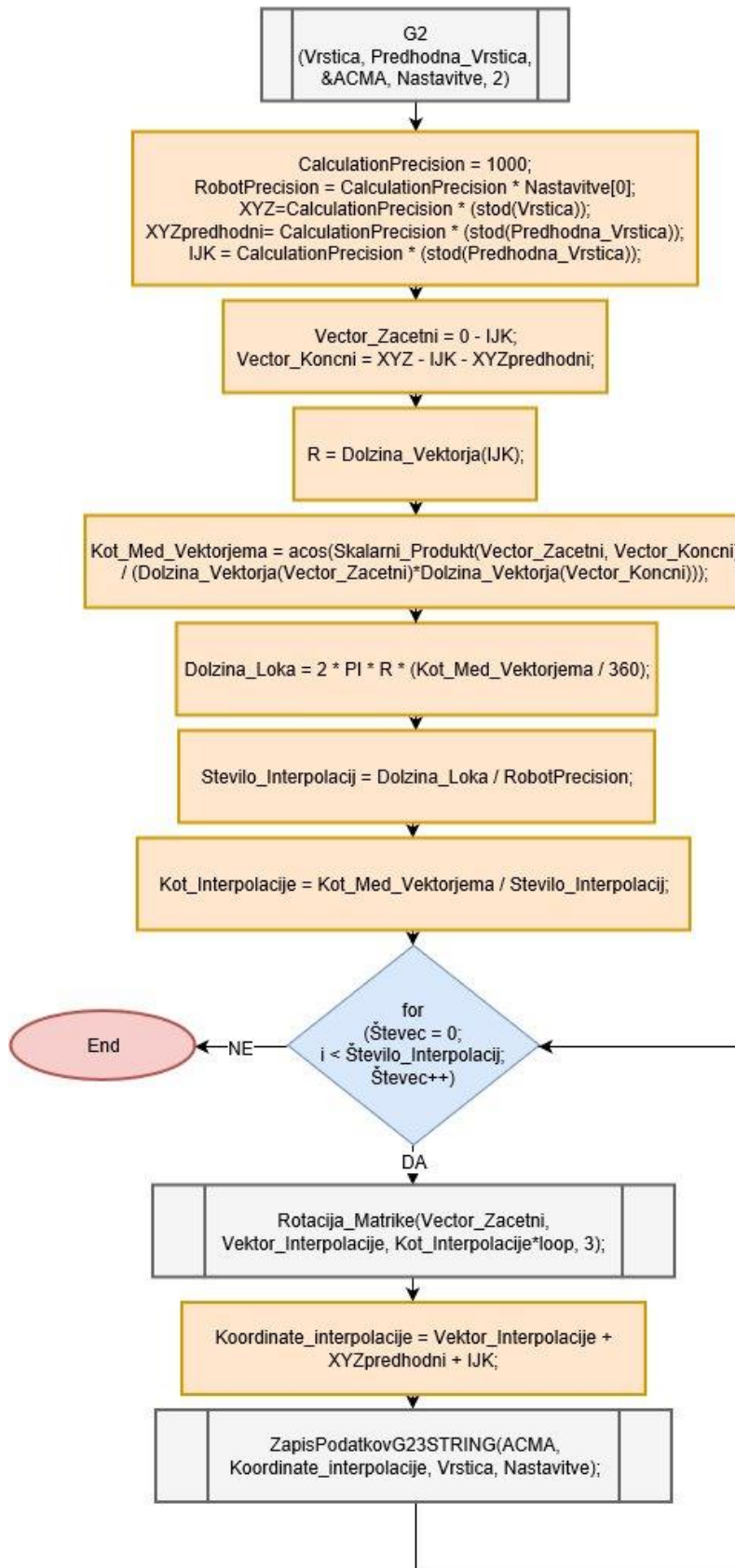


Slika 22 Diagram poteka za razčlenjevanje G-Kode

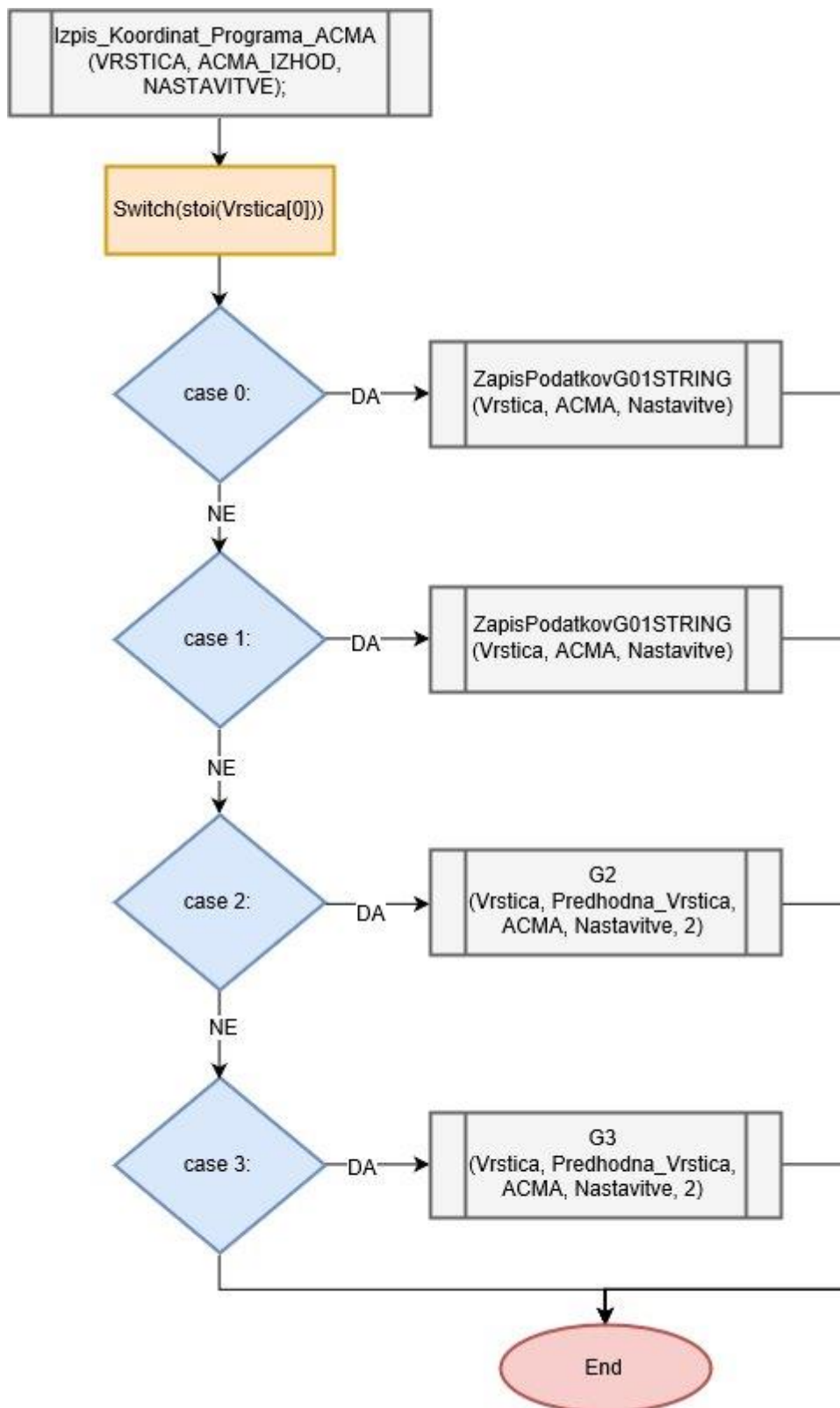
4.2.3 Pretvorba krožnih izsekov G2 in G3

V poglavju 4.1.3 je bilo omenjeno, da v GUI naložimo program NC oziroma G-Kodo za robot ACMA ABB XR701, opremljenega z rezkalnikom, zato je uporabljenih samo pet prostostnih stopenj. Pred pretvorbo smo prišteli nastavitve ničelne točke polizdelka razčlenjeni G-Kodi, kar se izvede vzporedno z razčlenjevanjem. Pri 5- oziroma 6-osni G-Kodi se vrednosti vsakega parametra preprosto pripiše ničelna točka in strukturira izhodni program ACMA za dodajanje točk oziroma izvedba inverzne kinematične transformacije. Pri 3-osni G-Kodi se pojavijo težave, saj je treba določiti trajektorije krožnega izseka ukaza G2 in G3. Trajektorijo določimo z izračunom interpolacij oziroma odsekov, ki se definirajo glede na natančnost robotovega giba. V ta namen so se zapisali zasebni metodi »G2_Algebra in »G3_Algebra« za izračun giba G2 ter G3. Diagram poteka za zasebno metodo »G2_Algebra«, ki je prikazana na Slika 23 Zasebna metoda »G2_Algebra« za pretvorbo ukaza G2 krožnega giba.

Pri javni metodi za izpis »Izpis_Koordinat_Programa_ACMA« (Slika 24 Diagram poteka za izpis programa ACMA) smo uporabili izjavo (switch case), ki glede na vhodno celo število G-ukaza (0, 1, 2, 3) izvede določen primer. G0 in G1 izpišeta vhodne podatke v izhoden niz z ustrežno strukturo in ukaza G2 in G3 se pred izpisom pretvorita v ustrezne translacijske gibe. [3]



Slika 23 Zasebna metoda »G2_Algebra« za pretvorbo ukaza G2 krožnega giba

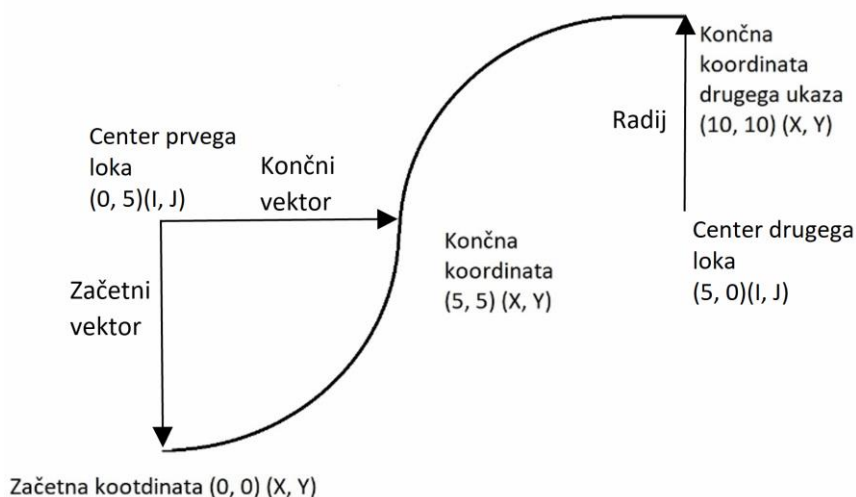


Slika 24 Diagram poteka za izpis programa ACMA

Da bi dosegli lažjo predstavo, smo pretvorbo ukaza G2 in G3 krožnega giba prikazali z enostavnim realnim primerom, prikazanim na Slika 25 in Slika 26 ter v prilogi.

```
N0 G1 X0 Y0
N1 G3 X5 Y5 I0 J5
N2 G2 X10 Y10 I5 J0
```

Slika 25 Primer NC-programa



Slika 26 Primer G-Kode oziroma NC-programa krožnega izseka G2 in G3

Izračun interpolacij oziroma odsekov smo izvedli z linearno algebro in trigonometričnimi funkcijami iz študijske literature [13]. Iz podatkov G-Kode, pridobljenih iz razčlenitve, določimo vektorja, ki izhajata iz središča krožnega izseka do predhodnih koordinat ZV (začetni vektor) ter do trenutnih koordinat KV (končni vektor), kar je prikazano na Slika 26.

Začetni vektor je definiran od centra krožnega giba do začetne koordinate; enačba

$$ZV = \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} = - \begin{bmatrix} i \\ j \\ k \end{bmatrix} \quad (1).$$

$$ZV = \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} = - \begin{bmatrix} i \\ j \\ k \end{bmatrix} \quad (1)$$

Končni vektor je definiran kot razlika predhodne (začetne) koordinate in vektorja središča krožnega izseka od začetne (končne) koordinate,

$$KV = \begin{bmatrix} x_{kv} \\ y_{kv} \\ z_{kv} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_{pred} \\ y_{pred} \\ z_{pred} \end{bmatrix} - \begin{bmatrix} i \\ j \\ k \end{bmatrix} \quad (2).$$

$$KV = \begin{bmatrix} x_{kv} \\ y_{kv} \\ z_{kv} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_{pred} \\ y_{pred} \\ z_{pred} \end{bmatrix} - \begin{bmatrix} i \\ j \\ k \end{bmatrix} \quad (2)$$

Radij krožnega izseka se izračuna s pomočjo skalarnega produkta

$$r = |\vec{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2} = \sqrt{i^2 + j^2 + k^2} \quad (3), \text{ in sicer dolžina vektorja } \vec{a} \text{ z vrednostmi}$$

i, j in k , kar razvidno s Slika 26.

$$r = |\vec{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2} = \sqrt{i^2 + j^2 + k^2} \quad (3)$$

Kot med vektorjema θ je razlika od inverznega tangensa začetnega vektorja in

$$\text{končnega vektorja. Kot vektorja je izražen v radianih, med } -\pi \text{ in } \pi, \theta = \arctan \frac{y}{x} \quad (4)$$

$$\theta = \arctan \frac{y}{x} \quad (4)$$

Če je bila vrednost imenovalca x enaka 0, so se pojavile težave, saj takrat ulomek ni definiran. Za rešitev težave s Slika 27 predpostavimo, da velja $y = \sin \theta$ in $x = r + \cos \theta$,

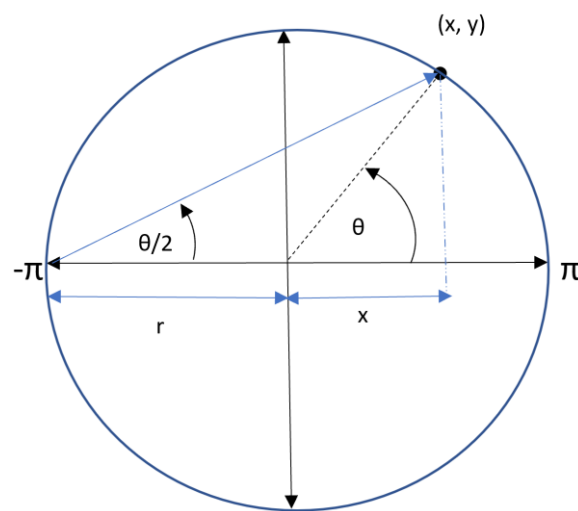
$$\text{določimo novo } \frac{\theta}{2} = \arctan \frac{y}{r+x} \quad (5) \text{ ter jo preuredimo v } \arctan(x, y) = \theta = 2 \frac{\theta}{2} \quad (6). \text{ S}$$

pomočjo skalarnega produkta in Pitagorovega izreka izračunamo polmer krožnega izseka in s tem dobimo končno enačbo $\theta = 2 \arctan \frac{y}{\sqrt{x^2 + y^2} + x}$ (7).

$$\frac{\theta}{2} = \arctan \frac{y}{r+x} \quad (5)$$

$$\arctan(x, y) = \theta = 2 \frac{\theta}{2} \quad (6)$$

$$\theta = 2 \arctan \frac{y}{\sqrt{x^2 + y^2} + x} \quad (7)$$



Slika 27 Prikaz izpeljave inverznega tangensa 2

Z $l = 2 \cdot \pi \cdot \frac{\theta}{360}$ (8) lahko iz obsega kroga določimo lok l oziroma krožni izsek kota θ .

$$l = 2 \cdot \pi \cdot \frac{\theta}{360} \quad (8)$$

Iz dolžine loka izračunamo število interpolacij v odvisnosti od natančnosti robota in nato določimo kot interpolacije. Kot med vektorjema določa število interpolacij. Z rotacijsko matriko na koncu še izračunamo koordinate posamezne interpolacije. G-Koda lahko izvajajo krožne gibe na različnih delovnih površinah, zato moramo preveriti

parametre vektorja središča krožnega odseka. Delovno površino določimo s Pseudokod
6 Določanje delovne površine.

Pseudokod 6 Določanje delovne površine

```

1: if I is NotDefined then
2:     Delovna površina YZ
3: else
4:     if J is NotDefined then
5:         Delovna površina XZ
6:     else
7:         Delovna površina XY
8:     end if
9: end if

```

V našem primeru je pri G2 krožni gib v smeri urnega kazalca, podana sta parametra I in

J, za katera sledi enačba
$$\begin{bmatrix} x_{inter} \\ y_{inter} \\ z_{inter} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} \quad (9).$$

$$\begin{bmatrix} x_{inter} \\ y_{inter} \\ z_{inter} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} \quad (9)$$

Za gib G3 proti smeri urnega kazalca preoblikujemo rotacijsko matriko v

enačbo
$$\begin{bmatrix} x_{inter} \\ y_{inter} \\ z_{inter} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} \quad (10).$$

$$\begin{bmatrix} x_{inter} \\ y_{inter} \\ z_{inter} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} \quad (10)$$

Za ravnine XZ in YZ uporabljamo ukaza G2 in G3 za gibanje v obliki vijačnice (Angl. Helix). Pri tem programu lahko v teh ravninah uporabimo ukaza G2 in G3 v obliki krožnega giba, ki ni standardiziran, giba pa potujeta v smeri osi z. Za izvedbo je treba prilagoditi rotacije koordinatnega sistema robota. Krožni gib v ravnini XZ prilagaja rotacijo okoli osi x (rotacija R ψ), gib v ravnini YZ prilagaja rotacijo okoli osi y (rotacija

R θ). To izvedemo z izračunom kota interpolacije po enačbi

$$\begin{bmatrix} x_{inter} \\ y_{inter} \\ z_{inter} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} \quad (9) \text{ in } \begin{bmatrix} x_{inter} \\ y_{inter} \\ z_{inter} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{zv} \\ y_{zv} \\ z_{zv} \end{bmatrix} \quad (10)$$

in tega prištejemo oziroma odštejemo od ustrezne rotacije robota. »S tem dosežemo, da je rezkar vedno pravokoten na obdelovalno površino in da se enakomerno obrablja orodje ter odvzema material, kar posledično vpliva na kakovost obdelave.« [3]

4.3 Evaluacija

Pri razvoju poprocesorja za pretvorbo G-Kode smo opazili kompleksnost aplikacije. Njegova izvedba je trajala dlje časa in zahteva popolno razumevanje robota za katerega se vmesnik razvija. Izvedli smo simulacije poprocesorja, vendar smo na realnem robotu opazili, da teorija in praksa ne gresta vedno z roko v roki. Na realnem primeru običajno hitro naletimo na nepredvidene težave. Iz rezultatov poskusov lahko izluščimo naslednje ugotovitve:

1. Za izdelavo poprocesorja in vmesnika GUI je potrebno obsežno znanje programiranja v jeziku C++/CLI ter dobro poznavanje programskega okolja Visual Studio 2017.
2. Prvotno je bil poprocesor zapisan tako, da je celotno vhodno in izhodno G-kodo skupaj z generiranim programom ACMA shranjeval v pomnilnik. To je zahtevalo 3-krat več pomnilnika od izboljšane verzije, ki obdeluje vrstico po vrstico in jo sproti izpisuje. To se pravi, da pri vsaki vrstici prepíše predhodne podatke in jih med delovanjem izpisuje na izhodno mesto (bogato tekstovno polje), kar privarčuje pomnilnik.
3. S programiranjem poprocesorja smo imeli težave s pridobivanjem ključnih podatkov z vhodne G-kode, saj je po navadi G-koda bogata z dodatnimi nastavitvami, ki pa v našem primeru niso nujne. Da bi se izognili shranjevanju neuporabnih podatkov, smo dopisali »if stavkom« pogoje za zagotavljanje iskanja ključnih podatkov. Za našo pridobljeno G-kodo iz CAM-programске opreme Siemens NX 8.5 je program deloval brezhibno, G-kode drugih programskih oprem bi lahko povzročale težave.

4. Težave smo imeli tudi z doseganjem natančnih gibov, saj robotu ni uspelo izvesti majhnih oziroma kratkih translacij. Ta problem smo odpravili z zmanjšanjem hitrosti in pospeška gibov, saj zaradi svoje vztrajnosti ni bil sposoben izvesti giba. Problematika bi se lahko rešila z izdelavo interpolatorja, ki bi glede na velikost samodejno prilagajal hitrost in pospešek ter izračunal novo trajektorijo.
5. Težave pri računanju interpolacij smo imeli, ker določenih kotov nismo mogli izračunati zaradi nedefiniranega ulomka. Odpravili smo jo z implementacijo inverznega tangensa 2.
6. Uporaba programa na starejših sistemih lahko predstavlja težave, saj je program preveden za 64-bitno arhitekturo procesorjev. Program smo prevedli tudi za 32-bitno arhitekturo procesorjev, kjer bi morali preveriti združljivost z operacijskim sistemom Windows 98. V nasprotnem primeru je treba program prevesti z uporabo programa Microsoft Visual C++ 2005, saj je zapis podatkov v našem primeru v Unicode. Druga možnost je uporaba MSLU (angl. Microsoft Layer for Unicode) za operacijske sisteme Windows 95, 98 in Me, ki bi omogočala delovanje zgrajenega programa.

5 SKLEP

Zaključno delo opisuje razvoj poprocesorja za pretvorbo G-kode, ki omogoča 3- oziroma 5-osne translacije za robota ACMA. Omenjeni poprocesor izpolnjuje vse cilje, ki smo si jih zadali na začetku zaključnega dela.

Poprocesor bi lahko izboljšali, da bi vključili v GUI tudi pretvorbo v numerični programski jezik ACMA. S tem bi lahko implementirali neposredno povezavo preko serijske komunikacije, če robot to omogoča. Na ta način bi ga poljubno programirali oziroma vodili in dosegli hitrejše prilagajanje programa in zmanjšali čas priprave, ki najbolj vpliva na ekonomičnost, saj med tem časom robot stoji.

Za krožni gib bi lahko izračunali vmesno točko krožnega izseka G-Kode z enačbami poglavja 4.2.3 in preko serijske komunikacije neposredno programirali z ABB-programskim jezikom uporabili ukaz »MoveC« oziroma bolj podrobno preučili program za dodajanje točk ACMA, če omogoča učenje krožnih gibov.

Zaključna naloga je bila uspešna in zajema vse aspekte študija Mehatronike. Lahko služi kot priročnik ostalim študentom, ki lahko izboljšajo vmesnik oziroma izdelajo boljše možnosti za uporabo robota ACMA ABB XR701 za obdelavo polizdelkov.

Viri

- [1] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik, and M. Munih. Robotika. Fakulteta za elektrotehniko, 2008.
- [2] M. Filipič. Posredno programiranje robota ACMA XR701: Magistrsko delo. Fakulteta za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, 2014.
- [3] Germadnik, G., Karner T., Berkovič, K., Fister, I. Načrtovanje poprocesorja za pretvorbo NC/ISO G-kode v translacije za 3/5 osnega robota ACMA ABB XR701. V: Fister, I. jr., Brodnik, A. (ur.). *5th Student Computer Science Research Conference*, Ljubljana, 09.10.2018. Koper: University of Primorska Press, 2018, str. 27–33. Dostopno na: <http://www.hippocampus.si/> [5. 8. 2019].
- [4] F. Klobučar and J. Zgonc. Robotika: Vodenje in programiranje robota ABB XR701.
- [5] D. Kurečič. Tehnološka in ekonomska analiza CNC proizvodnega sistema: Diplomsko delo. Fakulteta za strojništvo Univerze v Mariboru, 2014.
- [6] R. S. Lee and C. H. She. Developing a postprocessor for three types of five-axis machine tools. *The International Journal of Advanced Manufacturing Technology*, 13, (1997), 9, str. 658–665. Dostopno na: <https://link.springer.com/> [2. 8. 2018].
- [7] W. Lei and Y. Hsu. Accuracy test of five-axis cnc machine tool with 3d probe-ball Part I: Design and modeling. *International Journal of Machine Tools and Manufacture*, 42, (2002), 10, str. 1163–1170. Dostopno na: <https://www.sciencedirect.com/> [4. 8. 2018].
- [8] A. Lesnika. Robotski sistemi: interno gradivo za program mehatronika. Višja strokovna šola, 2012.
- [9] D. Nilsson. G-Code to RAPID translator for Robot-Studio. Degree Master of science, Trollhättan, SWEDEN: Department of Engineering Science University West, 2016. Dostopno na: <http://www.diva-portal.org/> [8. 9. 2018].
- [10] V. Rangunathan. C++/CLI Primer. Springer, 2016. Dostopno na: <https://www.apress.com/> [6. 9. 2018].
- [11] F. Ribeiro. 3d printing with metals. *Computing & Control Engineering Journal*, 1998. Dostopno na: <https://ieeexplore.ieee.org/> [10. 9. 2018].

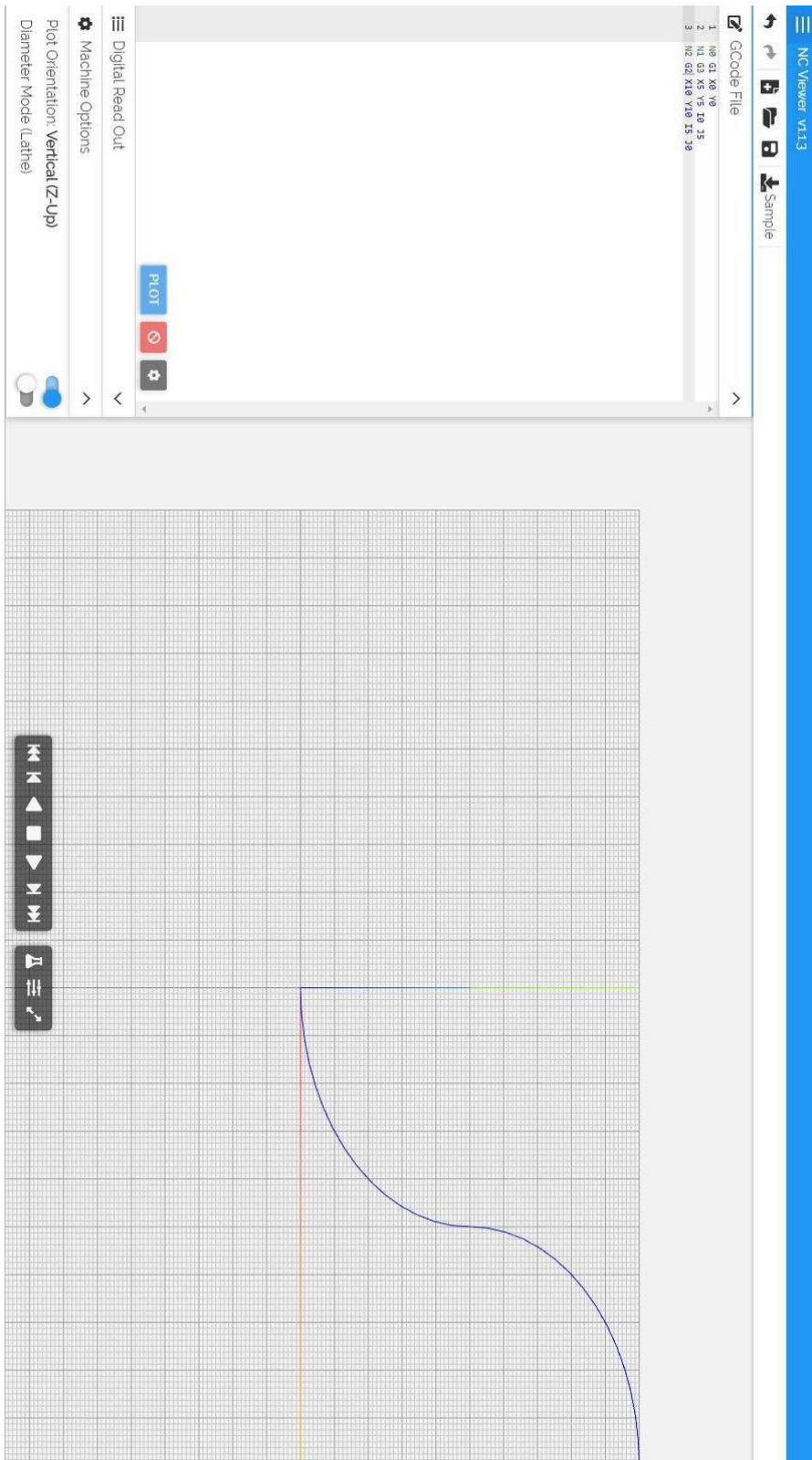
[12]M. Sekirnik. Priprava robotiziranega sistema ACMA XR701 in obračalne mize za postopke 3D frezanja: Diplomsko delo, Maribor: Fakulteta za strojništvo Univerze v Mariboru, 2015.

[13]B. Z. Sovič Tina, Simon Špacapan and R. Erveš. Matematika 1 : skripta. Technical report, Univerzitetna založba, 2018. Dostopno na: <http://www.dlib.si/> [6. 7. 2018].

[14]M. D. Network. .NET Programming with C++/CLI (Visual C++). Dostopno na: <https://docs.microsoft.com/> [6. 9. 2018].

Priloge

- Prikaz simulacije uporabljenega primera



- Prikaz pretvorjen G-Kode

X500.00 Y1900.00 Z200.00 A180.00 B0.00 C160.00 F50.00

X500.30 Y1900.01 Z200.00 A180.00 B0.00 C160.00

X500.60 Y1900.04 Z200.00 A180.00 B0.00 C160.00

X500.90 Y1900.08 Z200.00 A180.00 B0.00 C160.00

X501.19 Y1900.14 Z200.00 A180.00 B0.00 C160.00

X501.48 Y1900.22 Z200.00 A180.00 B0.00 C160.00

X501.76 Y1900.32 Z200.00 A180.00 B0.00 C160.00

X502.04 Y1900.43 Z200.00 A180.00 B0.00 C160.00

X502.31 Y1900.57 Z200.00 A180.00 B0.00 C160.00

X502.57 Y1900.71 Z200.00 A180.00 B0.00 C160.00

X502.82 Y1900.87 Z200.00 A180.00 B0.00 C160.00

X503.07 Y1901.05 Z200.00 A180.00 B0.00 C160.00

X503.30 Y1901.24 Z200.00 A180.00 B0.00 C160.00

X503.52 Y1901.45 Z200.00 A180.00 B0.00 C160.00

X503.72 Y1901.66 Z200.00 A180.00 B0.00 C160.00

X503.92 Y1901.89 Z200.00 A180.00 B0.00 C160.00

X504.10 Y1902.13 Z200.00 A180.00 B0.00 C160.00

X504.26 Y1902.38 Z200.00 A180.00 B0.00 C160.00

X504.41 Y1902.64 Z200.00 A180.00 B0.00 C160.00

X504.54 Y1902.91 Z200.00 A180.00 B0.00 C160.00

X504.66 Y1903.19 Z200.00 A180.00 B0.00 C160.00

X504.76 Y1903.47 Z200.00 A180.00 B0.00 C160.00

X504.84 Y1903.76 Z200.00 A180.00 B0.00 C160.00

X504.91 Y1904.05 Z200.00 A180.00 B0.00 C160.00

X504.96 Y1904.35 Z200.00 A180.00 B0.00 C160.00

X504.99 Y1904.65 Z200.00 A180.00 B0.00 C160.00

X505.00 Y1904.95 Z200.00 A180.00 B0.00 C160.00

X505.01 Y1905.30 Z205.00 A180.00 B0.00 C160.00

X505.04 Y1905.60 Z205.00 A180.00 B0.00 C160.00

X505.08 Y1905.90 Z205.00 A180.00 B0.00 C160.00
X505.14 Y1906.19 Z205.00 A180.00 B0.00 C160.00
X505.22 Y1906.48 Z205.00 A180.00 B0.00 C160.00
X505.32 Y1906.76 Z205.00 A180.00 B0.00 C160.00
X505.43 Y1907.04 Z205.00 A180.00 B0.00 C160.00
X505.57 Y1907.31 Z205.00 A180.00 B0.00 C160.00
X505.71 Y1907.57 Z205.00 A180.00 B0.00 C160.00
X505.87 Y1907.82 Z205.00 A180.00 B0.00 C160.00
X506.05 Y1908.07 Z205.00 A180.00 B0.00 C160.00
X506.24 Y1908.30 Z205.00 A180.00 B0.00 C160.00
X506.45 Y1908.52 Z205.00 A180.00 B0.00 C160.00
X506.66 Y1908.72 Z205.00 A180.00 B0.00 C160.00
X506.89 Y1908.92 Z205.00 A180.00 B0.00 C160.00
X507.13 Y1909.10 Z205.00 A180.00 B0.00 C160.00
X507.38 Y1909.26 Z205.00 A180.00 B0.00 C160.00
X507.64 Y1909.41 Z205.00 A180.00 B0.00 C160.00
X507.91 Y1909.54 Z205.00 A180.00 B0.00 C160.00
X508.19 Y1909.66 Z205.00 A180.00 B0.00 C160.00
X508.47 Y1909.76 Z205.00 A180.00 B0.00 C160.00
X508.76 Y1909.84 Z205.00 A180.00 B0.00 C160.00
X509.05 Y1909.91 Z205.00 A180.00 B0.00 C160.00
X509.35 Y1909.96 Z205.00 A180.00 B0.00 C160.00
X509.65 Y1909.99 Z205.00 A180.00 B0.00 C160.00
X509.95 Y1910.00 Z205.00 A180.00 B0.00 C160.00

- Prikaz strukturiranega programa za dodajanje točk ACMA

27 27 +0 27 +0 '3' +0 '1' +0

@80 @80 @80 @80 @80 @80 @80 @80 +0 13

13 13 13 27

@64 "500,00" 09 "1900,00" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
13 +0 "050" 13 +0 "050" 13 +0

'8' @77 13 +0 13 +0 @77 @77 @77 13 +0

@77 @77 @77 13 +0 @77 13 +0

13 +0

@77 13 +0

@77 @77 13 +0

27 27

@64 "500,00" 09 "1900,00" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68 13
+0 "50,00" 13 +0 "50,00" 13 +0

@64 "500,30" 09 "1900,01" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "500,60" 09 "1900,04" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "500,90" 09 "1900,08" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "501,19" 09 "1900,14" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "501,48" 09 "1900,22" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "501,76" 09 "1900,32" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "502,04" 09 "1900,43" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "502,31" 09 "1900,57" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "502,57" 09 "1900,71" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "502,82" 09 "1900,87" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "503,07" 09 "1901,05" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "503,30" 09 "1901,24" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "503,52" 09 "1901,45" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "503,72" 09 "1901,66" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "503,92" 09 "1901,89" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,10" 09 "1902,13" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,26" 09 "1902,38" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,41" 09 "1902,64" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,54" 09 "1902,91" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,66" 09 "1903,19" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,76" 09 "1903,47" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,84" 09 "1903,76" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,91" 09 "1904,05" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,96" 09 "1904,35" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "504,99" 09 "1904,65" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68

@64 "505,00" 09 "1904,95" 09 "200,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,01" 09 "1905,30" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,04" 09 "1905,60" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,08" 09 "1905,90" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,14" 09 "1906,19" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,22" 09 "1906,48" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,32" 09 "1906,76" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,43" 09 "1907,04" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,57" 09 "1907,31" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,71" 09 "1907,57" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "505,87" 09 "1907,82" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "506,05" 09 "1908,07" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "506,24" 09 "1908,30" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "506,45" 09 "1908,52" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "506,66" 09 "1908,72" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "506,89" 09 "1908,92" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "507,13" 09 "1909,10" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "507,38" 09 "1909,26" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "507,64" 09 "1909,41" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "507,91" 09 "1909,54" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "508,19" 09 "1909,66" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "508,47" 09 "1909,76" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "508,76" 09 "1909,84" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "509,05" 09 "1909,91" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "509,35" 09 "1909,96" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "509,65" 09 "1909,99" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
@64 "509,95" 09 "1910,00" 09 "205,00" 09 "180,00" 09 "0,00" 09 "160,00" + @68
27 27 27 27 27 +0 @59