Parametriranje posmičnih prigona CNC stroja

Bagarić, Dora

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Dora Bagarić

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Dr. sc. Tomislav Staroveški, dipl. ing.

Student:

Dora Bagarić

Zagreb, 2020.

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

Zahvaljujem mentoru doc. dr. sc. Tomislavu Staroveškom i komentoru dr. sc. Mihi Klaiću na pomoći, strpljenju i uloženom vremenu.

Dora Bagarić

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 Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602 - 04 / 20 - 6 / 3	
Ur. broj: 15 - 1703 - 20 -	

DIPLOMSKI ZADATAK

Student: **DORA BAGARIĆ** Mat. br.: 0035202821

Naslov rada na hrvatskom jeziku: **Parametriranje posmičnih prigona CNC stroja**

Naslov rada na engleskom jeziku: **Commissioning of CNC machine feed drives**

Opis zadatka:

U Laboratoriju za alatne strojeve trenutno se razvija ispitni CNC postav prikladan za nadzor obradnih procesa u stvarnom vremenu. Sustav je temeljen na otvorenom upravljačkom sustavu LinuxCNC, koji se izvršava u okruženju operacijskog sustava u stvarnom vremenu s Linux jezgrom. Upravljačko računalo povezano je s regulatorima glavnog i posmičnih prigona, te ostalim ulazno izlaznim modulima putem EtherCAT sabirnice. Idući korak u cilju cjelovite realizacije predmetnog postava je parametriranje sustava. Stoga je u radu potrebno:
1. Opisati strukturu LinuxCNC sustava.
2. Opisati princip rada EtherCAT sabirnice.
3. Parametrirati sustav u dijelu koji se odnosi na regulaciju posmičnih prigona.
4. Testirati regulacijski sustav posmičnih prigona.
5. Dati zaključke rada.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:	Rok predaje rada:	Predviđeni datum obrane:
30. travnja 2020.	2. srpnja 2020.	6. srpnja do 10. srpnja 2020.

Zadatak zadao:	Komentor:	Predsjednica Povjerenstva:
doc. dr. sc. Tomislav Staroveški	dr. sc. Miho Klaić	prof. dr. sc. Biserka Runje

SADRŽAJ

POPIS SLIKA

POPIS TABLICA

POPIS KRATICA

Kratica	**Izvorni naziv**
NC	numeric control
CNC	computer numeric control
MIT	Massachusetts Institute of Technology
OAC	open architecture controller
NIST	National Institute for Standards and Technology
EMC	Enhanced Machine Controller
OSACA	Open System Architecture for Controls within Automation System
OMAC	Open Modular Architecture Controllers
OSEC	Open System Environment for Controller
JOP	Japanese Open Promotion Group
HAL	Hardware Abstraction Layer
3D	trodimenzionalan
EMCMOT	Motion controller
EMCIO	Discrete I/O controller
EMCTASK	Task coordinating module
NML	Neutral Messaging Language
FIFO	First In First Out
I/O	input/output
SHIP	sredstvo za hlađenje ispiranje i podmazivanje
PLC	Programmable Logic Controller

GUI	Graphical user interface
VCP	virtual control panel
pyVCP	Python Virtual Control Panel
EtherCAT	Ethernet for Control Automation Technology
ESC	EtherCAT Slave Controller
PDI	Process Data Interface
DPRAM	Dual-Port-Random-Access Memory
SM	SyncManager
EEPROM	Electrically Erasable Programmable Read-Only Memory
MAC	media access control
FCS	frame check sequence
IFG	interframe gap
DLPDU	Data Link Protocol Data Units
WKC	Working Counter
RD	read
WR	write
RW	read/write
RMW	read/multiple write
ADP	polje adrese za uređaj koji se adresira
ADO	polje adrese za fizičku lokaciju memorije ili registra podređenog uređaja
ADR	adresa veličine svih 32 bita adresnog polja zaglavlja datagrama
FMMU	Fieldbus Memory Management Unit
CoE	CANopen over EtherCAT

EoE	Ethernet over EtherCAT
FoE	File access over EtherCAT
SoE	Servo drive profile over EtherCAT
OD	object dictionary
SDO	Service Data Objects
PDO	Process Data Objects
TPDO	transfer Process Data Object
RPDO	receive Process Data Object
CiA	CANopen in Automation
ESI	EtherCAT Slave Information
ENI	EtherCAT Network Information

POPIS OZNAKA

Oznaka	Jedinica	Opis

SAŽETAK

U diplomskom radu se opisao proces parametriranja posmičnih prigona CNC stroja. Rad je podijeljen na pet dijelova. U uvodu se govori o upravljačkim sustavima otvorene arhitekture te o opravdanosti njihove primjene. Zatim je detaljnije opisan upravljački sustav otvorene arhitekture korišten tijekom izrade eksperimentalnog dijela diplomskog rada, odnosno LinuxCNC. Tema trećeg dijela je opis principa rada EtherCAT sabirnice kojom se ostvarila komunikacija, tj. razmjena podataka između upravljačkog sustava i servo regulatora pogonskih motora stroja. Četvrto poglavlje se odnosi na eksperimentalni dio rada. Predstavljena je konfiguracija servo regulatora i upravljačkog sustava nužna za realizaciju gibanja posmičnih prigona CNC stroja. U petom i posljednjem dijelu se dao zaključak rada.

Ključne riječi: CNC stroj, otvoreni upravljački sustav i LinuxCNC, EtherCAT, servo regulator

SUMMARY

The thesis describes the process of commissioning of CNC machine tool feed drives. The thesis is divided into five parts. The introduction deals with the open architecture controllers and the motivation for their application. Furthermore, LinuxCNC, an open architecture controller used in the experimental part of the thesis is described in greater detail. The topic of the third part is the principle of operation of EtherCAT used to establish the communication between the controller and the servomotor drives. The fourth part is dedicated to the experiment. The configurations of the feed drives and the LinuxCNC necessary for the realization of the CNC machine feed motion are presented. Concluding remarks are presented in the final chapter.

Key words: CNC machine, open architecture controller, LinuxCNC, EtherCAT, servodrive

1. UVOD

Numerički upravljani (eng. *numeric control, NC)* sustavi razvijeni su od strane Instituta za tehnologiju iz Massachusettsa (eng. *Massachusetts Institute of Technology, MIT)* još ranih 1950-ih godina. Svoj napredak su doživjeli s pojavom i razvojem mikroprocesora. Njihovom primjenom nastali su računalno numerički upravljani (eng. *Computer Numeric Control, CNC)* sustavi. S obzirom na kompleksnost CNC sustava, njihovim je tržištem dominiralo samo nekoliko tvrtki iz Japana i Njemačke. Proizvođači su svoje CNC sustave radili kao zatvorene kako bi zaštitili svoju tehnologiju i zadržali mjesto na tržištu. No 1980-ih godina promjenu donosi razvoj računalne tehnologije i računalnih mreža te potreba za naprednim upravljačkim sustavima koje zahtijevaju visokobrzinski i visokoprecizni strojevi. Troškovi daljnjeg razvoja zatvorenih sustava bili su vrlo visoki te se krenulo u razvoj otvorenih upravljačkih sustava [1].

Značajni napori su uloženi u razvoj takvih sustava za alatne strojeve kako bi se zadovoljila konstantna potražnja za višom funkcionalnošću i fleksibilnošću alatnih strojeva, kvalitetnijim proizvodima i smanjenim troškovima. Da bi se odgovorilo na takve zahtjeve upravljački sustavi otvorene arhitekture (eng. *open architecture controller*, OAC) mora biti hardverski i softverski fleksibilan i standardiziran [2]. To bi značilo da se mora omogućiti korištenje različitih programskih jezika, operacijskih sustava [3], senzora i softverskih te hardverskih komponenti različitih proizvođača. Uspješna implementacija OAC je posebice važna za razvoj rekonfigurabilnih proizvodnih sustava i za razvoj naprednih proizvodnih sustava s visokom razinom autonomnog rada, koja se postiže nadzorom procesa i kontrolnim modulima [2].

Američki Nacionalni institut za standardizaciju i tehnologiju (eng. *National Institute for Standards and Technology, NIST*) je predložio prvo rješenje upravljačkog sustava otvorene arhitekture imena *Enhanced Machine Controller* (EMC). Ispunjavao je sve zahtjeve za ostvarenje sustava napredne numeričke kontrole. Sustav se mogao instalirati na obično osobno računalo u okruženju operacijskog sustava s Linux ili FreeBSD jezgrom. Nakon toga su se pokrenuli još neki projekti u Europi, Sjedinjenim Američkim Državama i Japanu. Najbitniji bi bili [2]:

- OSACA (Open System Architecture for Controls within Automation System)
- OMAC (Open Modular Architecture Controllers)

- OSEC (Open System Environment for Controller)
- JOP (Japanese Open Promotion Group)

Projekti su pokretani i podržavani od strane raznih proizvođača alatnih strojeva, proizvođača upravljačkih sustava za CNC strojeve, korisnika i sveučilišnih ustanova. Također su razvijeni i komercijalni sustavi otvorene arhitekture: *Delta Tau PMAC-NC, IBH PA 8000, Galil DMC 1000, Creonics MCC VME, Adept Series A, Aerotech Unidex 31, CIMplus i Typ3 osa*. Takve vrste upravljačkih sustava su odlični za istraživanje i razvoj, ali problem i dalje predstavljaju standardizacija i podrška što sprječava primjenu ovih sustava i u području industrije [3].

2. LINUXCNC

U ovom radu se koristio upravljački sustav koji je novija verzija u odnosu na EMC, a naziv „LinuxCNC“ je dobio kada su se, iz pravnih razloga, imena „EMC“ popraćena bilo kojom brojkom morala izmijeniti. LinuxCNC je predviđen za izvršavanje u okruženju operacijskog sustava s Linux jezgrom u stvarnom vremenu. Odabran je zbog njegovog otvorenog koda, njegove modularne strukture, zrelosti koda i visoke stabilnosti. Najveći iskorak LinuxCNC sustava je postignut uvođenjem hardverskog apstrakcijskog sloja (eng. *Hardware Abstraction Layer*, HAL) kojim je omogućeno jednostavno povezivanje modula sustava. Kako bi LinuxCNC funkcionirao, Linux jezgri je potrebno nadodati ekstenzije za izvršavanje u stvarnom vremenu. U okruženju bez stvarnog vremena, rad sustava je moguć, ali samo za simulacije pri kojima se ne koristi stvarni hardver [2].

LinuxCNC se kao softverski sustav za računalno upravljanje može koristiti za alatne strojeve, ali i robote, 3D printere, laserske i plazma rezače ili neke druge numerički upravljane strojeve. Može kontrolirati servomotore, koračne motore, releje i ostale uređaje vezane za kontrolu gibanja što je moguće u zavisnosti o odgovarajućem hardveru. Ovim sustavom može se upravljati s do 9 simultanih osi koje se mogu sinkronizirano gibati i kojima se upravlja G-kodom (RS-274NGC) [4]. Potencijal komercijalne primjene LinuxCNC sustava nazire se u mnogim uspješnim primjenama na revitaliziranim strojevima i razvijenim prototipovima strojeva koji mogu biti i složene kinematike [2].

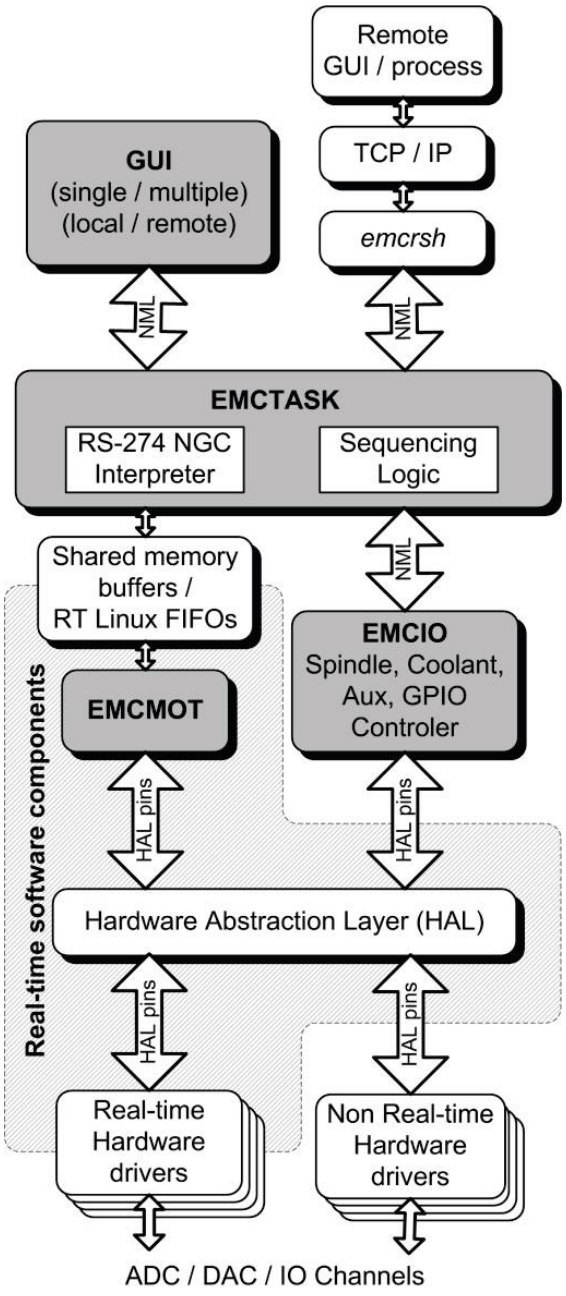
2.1. Struktura LinuxCNC-a

LinuxCNC softver se sastoji od četiri glavne komponente [2]:

1. Kontroler gibanja (eng. *Motion controller,* EMCMOT)
2. Diskretni I/O kontroler (eng. *Discrete I/O controller,* EMCIO)
3. Koordinirajući modul (eng. *Task coordinating module,* EMCTASK)
4. Skupine različitih modula za prikaz korisničkog sučelja

Od navedena četiri modula, samo EMCMOT je modul koji se izvršava u stvarnom vremenu. Kao što se može vidjeti na sljedećoj slici (slika 1), komunikacija između korisničkog sučelja i EMCTASK-a te između EMCTASK-a i modula EMCIO ostvaruje se pomoću kanala *Neutral*

Messaging jezika (eng. *Neutral Messaging Language, NML*) te se ne izvršava u stvarnom vremenu. Komunikacija između modula EMCMOT koji se izvršava u stvarnom vremenu i EMCTASK modula izvedena je pomoću dijeljene memorije ili RT-Linux FIFO (eng. *First In First Out*) mehanizma. Moduli koji se izvršavaju u stvarnom vremenu, npr. driveri za hardver ili PID regulatori, komuniciraju putem HAL signala, ali se u okviru HAL-a također mogu povezati i moduli koji se ne izvršavaju u stvarnom vremenu [2].



Slika 1. Struktura LinuxCNC sustava [2]

2.1.1. Hardverski apstrakcijski sloj

Hardverski apstrakcijski sloj (eng. *Hardware Abstraction Layer*, HAL) se može opisati kao način na koji je omogućeno da se pomoću više različitih blokova (HAL modula) strukturira cjeloviti složeni sustav. Riječ „hardverski“ se nalazi u imenu jer je namjena za koju je originalno zamišljen bila ta da se olakša konfiguracija LinuxCNC-a za široki spektar hardverskih uređaja. Veliki broj tih blokova čine driveri za hardver [4]. HAL je izrazito modularne strukture te ga korisnik može rekonfigurirati prema svojim potrebama. Omogućava prijenos podataka u stvarnom vremenu između pojedinih modula LinuxCNC-a i prijenos podataka prema hardveru. Također pruža i mogućnost razvoja vlastitih drivera za hardver i softverskih modula koji se onda mogu implementirati u sustav te izvršavati u stvarnom vremenu. Na prikazu strukture LinuxCNC sustava (slika 1) može se vidjeti kako se hijerarhijski nalazi između modula EMCMOT i EMCIO i drivera za hardver [2].

Kako se npr. CNC stroj sastoji od brojnih fizičkih dijelova – motori, enkoderi, granični prekidači itd., koji se prvo moraju odabrati, zatim ugraditi i međusobno povezati ili ožičiti da bi činili funkcionalnu cjelinu, tako se mora učiniti i na softverskoj razini. U HAL prostoru se ekvivalenti tim fizičkim dijelovima nazivaju HAL moduli. Dakle HAL modul je program koji ima točno definirane ulaze, izlaze i ponašanje te se može učitati i povezati s ostalim modulima u HAL prostoru. Kako su fizički dijelovi spojeni pomoću žica, tako se u HAL modulu komponente povezuju „signalima“ (logičkim vezama formiranim mehanizmima dijeljene memorije). Da bi se moduli uopće mogli spajati, stvaraju im se softverske značajke, tzv. HAL izvodnice (eng. *HAL pin*). Izvodnice su terminali kojima se dodjeljuje ime i koriste se kao ulazi i/ili izlazi za povezivanje HAL modula. Prilikom spajanja HAL modula se mora uzeti u obzir to da postoje različiti tipovi izvodnica i signala, a da bi se oni mogli spojiti moraju biti istoga tipa. Trenutno postoji 4 tipa signala [4]:

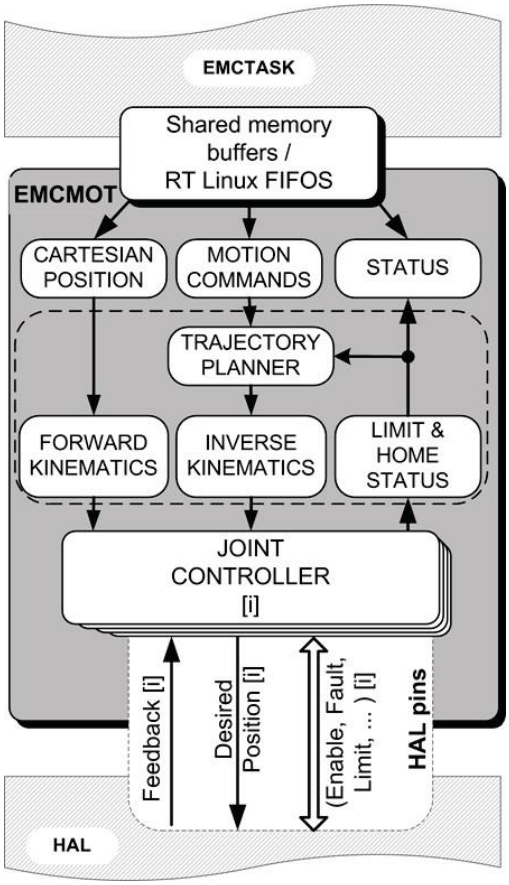
1. bit – vrijednosti mogu biti 1 ili 0
2. float – vrijednosti u kojima se decimalni zarez može pomicati po potrebi, ima max. 64 bitni iznos
3. u32 (eng. *unsigned integer*) – cijeli broj koji može biti samo pozitivan (od 0 do 4294967295)
4. s32 (eng. *signed integer*) – cijeli broj koji može imati negativnu ili pozitivnu vrijednost (od -2147483648 do 2147483647)

2.1.2. Kontroler gibanja

Modul EMCMOT (eng. *Motion controller*) (slika 2) se izvršava ciklički u stvarnom vremenu. Njegova uloga je planiranje trajektorije, izračunavanje direktne i inverzne kinematike i proračunavanje reference pozicije prema servo regulatorima posmičnih prigona. Modul obuhvaća i nadziranje pozicija upravljanih osi, proračun sljedeće točke trajektorije i interpolaciju između tih točaka trajektorije. EMCMOT podržava logičke ulaze za povezivanje signala koji dopiru iz graničnih i referentnih prekidača za svaku numerički upravljanu os [2].

Parametri koje je za ovaj modul potrebno postaviti tijekom učitavanja sustava su: broj i vrsta osi (linearna ili rotacijska), najveća brzina i akceleracija, mjerne jedinice i vrijeme jednog ciklusa proračunavanja trajektorije. Ti parametri se u sustav postavljaju u konfiguracijskoj datoteci u fazi inicijalizacije sustava [2].

EMCMOT na podređene module koji se izvršavaju u stvarnom vremenu, uzajamno djeluje pomoću HAL signala. Primjer podređenog HAL modula je PID kompenzacijski algoritam, koji prima referencu pozicije iz EMCMOT modula, a izlaz (odn. referencu brzine) šalje podređenom servo regulatoru posmičnog prigona [2].



Slika 2. Struktura EMCMOT modula LinuxCNC-a [2]

2.1.3. Diskretni I/O kontroler

EMCIO modul (eng. *Discrete I/O controller*) LinuxCNC sustava upravlja svim ulazno-izlaznim (eng. *input/output*, I/O) funkcijama koje nemaju direktnu vezu s gibanjem posmičnih osi stroja, tj. sve funkcije koje bi se u kontekstu G-koda mogle definirati kao pomoćne ili M funkcije. Izveden je iz većeg broja podređenih kontrolera za upravljanje prigonima glavnog vretena, automatskom izmjenom alata, sredstvom za hlađenje, ispiranje i podmazivanje (akronim: SHIP) i ostalim ulogama koje su najčešće potrebne za cjelovito upravljanje CNC strojem. Kako se radi o funkcijama koje su vrlo specifične, tj. ovise o karakteristikama konkretnog stoja na kojem se želi primijeniti LinuxCNC sustav, za svaku funkciju se može izraditi posebna HAL datoteka. Kada se zna koje će funkcije biti potrebne, odgovarajuća HAL datoteka se navodi u glavnoj konfiguracijskoj datoteci i time funkcija postaje dio sustava. Međutim, u jednostavnijim konfiguracijama, cjelovita konfiguracija koja opisuje logičke veze među pojedinim modulima može biti zapisana u jednoj HAL datoteci [2].

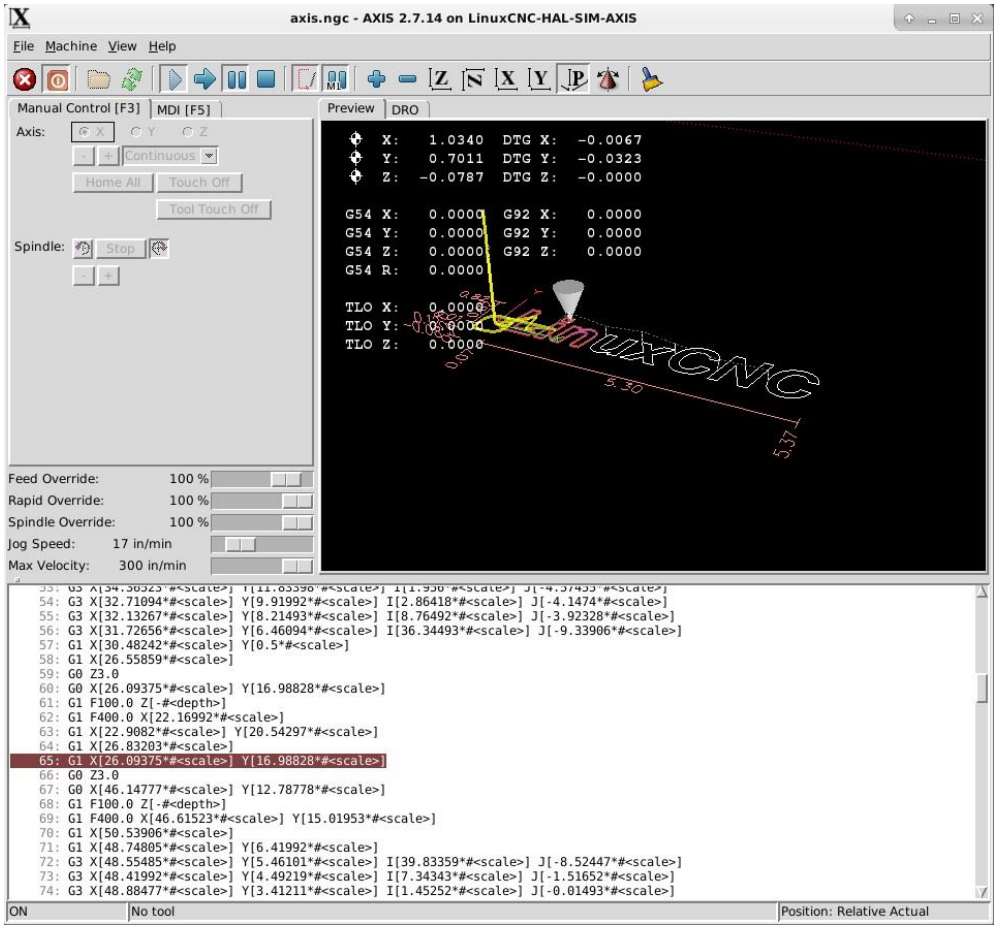
Navedeni podsustavi, tj. podređeni kontroleri su već kodirani i postoje unutar LinuxCNC-a, ali se mogu lako prilagoditi, proširiti ili izostaviti. Prilagodbe sustava moguće su primjenom programabilnog logičkog kontrolera (eng. *Programmable Logic Controller*, PLC) ili specijaliziranog alata za pisanje HAL modula (*„Comp"*). PLC se konfigurira prilikom postavljanja sustava i to pomoću ljestvičastih logičkih dijagrama unutar posebnog grafičkog sučelja pod nazivom „*ClassicLadder*" [2].

2.1.4. Koordinirajući modul

Modul EMCTASK (eng. *Task coordinating module*) nadzire status podređenih modula (EMCMOT i EMCIO) i koordinira ih. Ujedno prima i analizira naredbe koje dolaze ili od operatora preko grafičkog korisničkog sučelja (eng. *Graphical user interface*, GUI) ili od nekog drugog procesa. Te naredbe zatim interpretira i pretvara u NML (eng. *Neutral Messaging Language*) poruke i šalje ih u EMCMOT, EMCIO ili samom sebi (EMCTASK). Naredbe su pisane u G-kodu pomoću G i M funkcija [2].

2.1.5. Skupine različitih modula za prikaz korisničkog sučelja

Postojeća grafička sučelja za LinuxCNC su: *keystick, xemc, tkemc, mini, AXIS, GMOCCAPY* te još nekolicina. *AXIS* je naprednije grafičko sučelje koje ima i interaktivni prikaz G-koda koji se izvršava (slika 3).



Slika 3. Prikaz *AXIS* grafičkog sučelja

Većina GUI-ja se može proširivati i prilagođavati potrebama korisnika, a moguće je napisati i vlastiti GUI. To se provodi pomoću tzv. virtualnih kontrolnih panela (eng. *virtual control panels*, VCP), a biblioteka kojom se to omogućava naziva se pyVCP ili *Python Virtual Control Panel* [2].

Slika 4 prikazuje datoteku kojom se opisuje izgled i HAL izvodnice (eng. *HAL pin*) grafičkog sučelja koje se izrađuje.

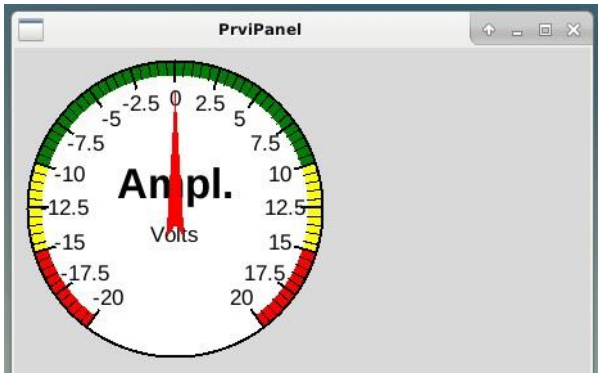
Open vcp1p.xml ~/

```
<pyvcp>
    <hbox>
    <meter>
        <halpin>"sinus1"</halpin>
        <text>"Ampl."</text>
        <subtext>"Volts"</subtext>
        <size>250</size>
        <min_>-20</min_>
        <max_>20</max_>
        <majorscale>2.5</majorscale>
        <minorscale>0.5</minorscale>
        <region1>(-20,20,"red")</region1>
        <region2>(-15,15,"yellow")</region2>
        <region3>(-10,10,"green")</region3>
    </meter>
    <meter>
        <halpin>"sinus2"</halpin>
        <text>"Ampl."</text>
        <subtext>"Volts"</subtext>
        <size>250</size>
        <min_>-20</min_>
        <max_>20</max_>
        <majorscale>2.5</majorscale>
        <minorscale>0.5</minorscale>
        <region1>(-20,20,"red")</region1>
        <region2>(-15,15,"yellow")</region2>
        <region3>(-10,10,"green")</region3>
    </meter>
    </hbox>

        <spinbox>
            <halpin>"sg1ampl"</halpin>
            <min_>-20</min_>
            <max_>20</max_>
            <initval>0</initval>
            <resolution>0.1</resolution>
            <format>"2.3f"</format>
        </spinbox>
</pyvcp>
```

Slika 4. Opisna datoteka grafičkog sučelja

Slika ispod (slika 5) prikazuje kako izgleda pokrenuto grafičko sučelje koje je opisano gore navedenom datotekom.



Slika 5. Prikaz grafičkog sučelja

3. ETHERCAT

EtherCAT (eng. *Ethernet for Control Automation Technology*) je industrijski Ethernet i koristi se standardnim okvirima i fizičkim slojem kako je opisano u Ethernet standardu IEEE 802.3. EtherCAT mreža odgovara na posebne zahtjeve koje postavlja industrija [5]:

1. komunikacija u stvarnom vremenu (eng. *real time*)
2. mogućnost ugradnje velikog broja čvorova u sustavu, od kojih svaki ciklički šalje ili prima određenu, točno definiranu količinu procesnih podataka
3. niski troškovi ugradnje

Međunarodno je standardizirana i otvorena tehnologija koju je osmislila tvrtka Beckhoff. Otvorena bi značilo da je svima dostupna i slobodna za korištenje. Na primjeru LinuxCNC upravljačkog sustava je pojašnjeno da je program otvorenog koda slobodan za korištenje, ali i za nadograđivanje i preinake. To kod EtherCAT-a nije slučaj. Prilagođavanje EtherCAT-a pojedinom korisniku, označilo bi kraj interoperabilnosti. Za EtherCAT-ov daljnji razvoj zadužena je organizacija *EtherCAT Technology Group*. Organizacija okuplja proizvođače i korisnike u svrhu unaprjeđenja EtherCAT tehnologije [5].

Gore navedeni zahtjevi koje industrija postavlja pred Ethernet mrežu, čine ju gotovo neiskoristivom. Ako se u podatkovnom okviru koji se šalje u mrežu, nalazi samo jedan datagram (naredba) i koristi se za samo jedan čvor u mreži, iskoristivost izmjene podataka drastično pada, a takav je slučaj upravo kod Ethernet mreže. Najkraći Ethernet datagram ima 84 bajta i ako se, npr. od strane servo regulatora šalje 4 bajta cikličkih procesnih podataka koji daju stvarnu poziciju i statusne informacije, a prima 4 bajta cikličkih procesnih podataka koji daju željenu poziciju i kontrolne informacije, iskoristivost pada na 4/84 = 4,8% [5]. Taj problem se EtherCAT-om riješio na način da se u mreži nalazi jedan nadređeni uređaj (eng. *EtherCAT master device*) i više podređenih uređaja (eng. *EtherCAT slave device*). Nadređeni uređaj je jedini koji može inicirati izmjenu podataka, a podatkovni okvir koji šalje dolazi do svakog pojedinog čvora u mreži. Okvir se obrađuje i šalje dalje do drugog čvora u mrežu [6].

3.1. Fizički sloj

EtherCAT se oslanja na standardni Ethernet fizički sloj. Ova tehnologija omogućava brzine do 100 Mb/s i istovremenu dvosmjernu komunikaciju (eng. *full-duplex)*. Vrijeme ciklusa je

0,1 ms, ali po potrebi može biti i duže. Sustav se sastoji od jednog nadređenog uređaja i više podređenih uređaja. Kao prijenosni medij se koriste standardizirani mrežni kablovi kategorije 5 (Cat 5). Konektori koji se mogu koristiti su RJ45 i M12 [6].

3.1.1. Topologija mreže

S obzirom na nužnost adresiranja, broj uređaja koje EtherCAT mreža podržava ovisi o veličini adrese uređaja. Veličina adrese iznosi 16 bita stoga je moguć broj uređaja od 2^{16} što je jednako 65 535. Oni mogu biti spojeni u: liniju, zvijezdu, drvo ili bilo koju kombinaciju navedenih topologija [5]. Uobičajeno su podređeni uređaji spojeni u liniju te se takva shema ožičenja naziva serija ili serijski spoj, što ukazuje na lančanu povezanost elemenata. Svaki podređeni uređaj ima najmanje dva Ethernet priključka kako bi se povezali na uređaj prije i uređaj iza u mreži. Posljednji podređeni uređaj ima povratnu (eng. *loopback*) funkciju i u suprotnom smjeru vraća okvir nadređenom uređaju koji ima samo jedan Ethernet priključak predviđen za tu svrhu. Ethernet omogućuje istovremenu dvosmjernu komunikaciju za koju se koriste dva para žica unutar istog kabla i zbog toga topologija nalikuje na liniju, kao kod mnogih industrijskih sabirnica polja. Najveća udaljenost na koju se mogu postaviti dva uređaja spojena Ethernet kablom je čak 100 m [6].

3.1.2. Struktura uređaja

Mnoge komunikacijske tehnologije koje su osmišljene specifično za industrijsko okruženje imaju drugačiju strukturu nadređenog i podređenog uređaja. Isto vrijedi i za EtherCAT. EtherCAT nadređeni uređaji se oslanjaju na standardni komunikacijski hardver (full-duplex Ethernet kontrolere mrežnog sučelja – eng. *network interface controller*) i odgovarajući softver, a mogu se koristiti i opcije otvorenoga koda kao što je operativni sustav s Linux jezgrom. S druge strane, podređeni uređaji imaju posebno konstruirane hardverske komponente – EtherCAT kontrolere podređenog uređaja (eng. *EtherCAT Slave Controller*, ESC). Obrada okvira odvija se samo u jednom smjeru, dok se u drugom okvir samo prosljeđuje za što je zadužen zadnji čvor u mreži. Podređeni uređaji se u oba smjera ponašaju kao repetitori, tj. mogu ponovno generirati električni signal koji su primili tako da dodatne komponente nisu potrebne [6].

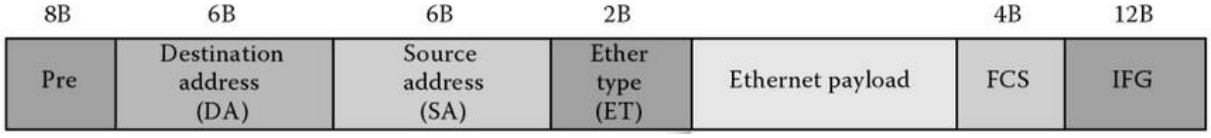
Kao što je spomenuto gore, podređeni uređaji koriste ESC integrirani krug. Taj krug ima ulogu pružanja sučelja za izmjenu podataka između nadređenog i podređenog uređaja.

Također provodi i EtherCAT protokol u stvarnom vremenu. To se ostvaruje tako što se EtherCAT okvire obrađuje „*on the fly*“, tj. s obradom datagrama se započinje prije nego li se primio cijeli okvir. Zatim se mikrokontroler preko PDI (eng. *Process Data Interface*) sučelja opskrbljuje podacima dobivenim iz memorije uređaja. Podaci o procesu i parametri se izmjenjuju pomoću DPRAM (eng. *Dual-Port-Random-Access Memory)* memorije koja zajedno s registrima čini lokalnu memoriju ESC-a. Kako bi se osigurala konzistentnost podataka, uvedeni su odgovarajući mehanizmi, tzv. *SyncManager*, SM. U EEPROM (eng. *Electrically Erasable Programmable Read-only Memory)* memoriji podređenog uređaja se pohranjuju informacije o njegovom identitetu i funkcijama. Sadržaj te memorije treba biti definiran od strane proizvođača tijekom razvoja uređaja [7].

3.2. Sloj veza

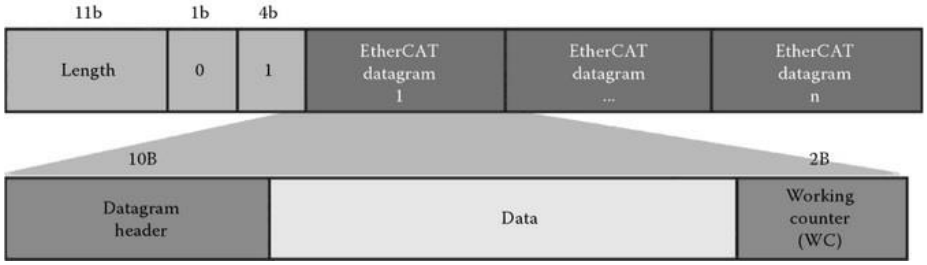
Spomenuto je kako je način na koji se pomoću EtherCAT-a riješio problem slabe iskoristivosti izmjene podataka, taj da se uveo samo jedan okvir koji je obično dovoljan za slanje podataka prema svim čvorovima i primanje podataka od svih čvorova. Nadređeni uređaj ili čvor šalje datagram koji prolazi kroz svaki čvor i njime nadređeni uređaj obavlja operacije upisivanja i/ili čitanja podataka iz specifične lokacije u memoriji podređenog čvora. Svaki čvor, tj. podređeni uređaj momentalno čita i/ili upisuje podatke u okvir, odnosno tijekom primanja i slanja ovira idućem elementu. Pri tome je bitno da uređaj obrađuje samo onaj datagram koji je adresiran na njega i na neku specifičnu lokaciju u njegovoj memoriji. EtherCAT datagramima se iskoristivost komunikacije povećava na 90%, a time što samo nadređeni uređaj može slati naredbe, sprječavaju se nepredviđena kašnjenja i osigurava se izvršavanje u stvarnom vremenu [5].

Naredbe se šalju u obliku standardnih Ethernet okvira te se sastoje od (slika 6): preambule (8 bajtova), MAC (eng. *media access control*) adrese izvora i odredišta (svaka 6 bajtova), vrste Ethernet paketa (2 bajta, namješteno na 0x88A4 kako bi se EtherCAT okvir razlikovao od onih koji to nisu), FCS (eng. *frame check sequence;* 4 bajta) i IFG (eng. *interframe gap*) [6].



Slika 6. Struktura Ethernet okvira [6]

EtherCAT podaci, tj. naredbe transportiraju se u podatkovnom dijelu Ethernet okvira u obliku EtherCAT datagrama (slika 7). Prema standardnim specifikacijama se EtherCAT datagrami također nazivaju i „*Type 12 Data Link Protocol Data Units"* (DLPDU). Kao što se može vidjeti na slici ispod (slika 7), svaki datagram se sastoji od zaglavlja, polja s podacima i brojača (eng. *Working Counter*, WKC). U zaglavlju se upisuju informacije kao što su adresa ili vrsta naredbe (*read* – RD, *write* - WR, *read*/*write* - RW, *read/multiple write* - RMW). Brojaču se inkrementalno povećava vrijednost svaki puta kada je EtherCAT uređaj na kojeg je bila adresirana neka naredba, čitao/pisao podatke u/iz datagrama. Kako svaki datagram ima očekivanu vrijednost brojača, tako se pri povratku okvira u nadređeni uređaj može provjeriti točnost obrade datagrama na način da se vrijednost brojača dobivenog datagrama usporedi s očekivanim brojem [6].

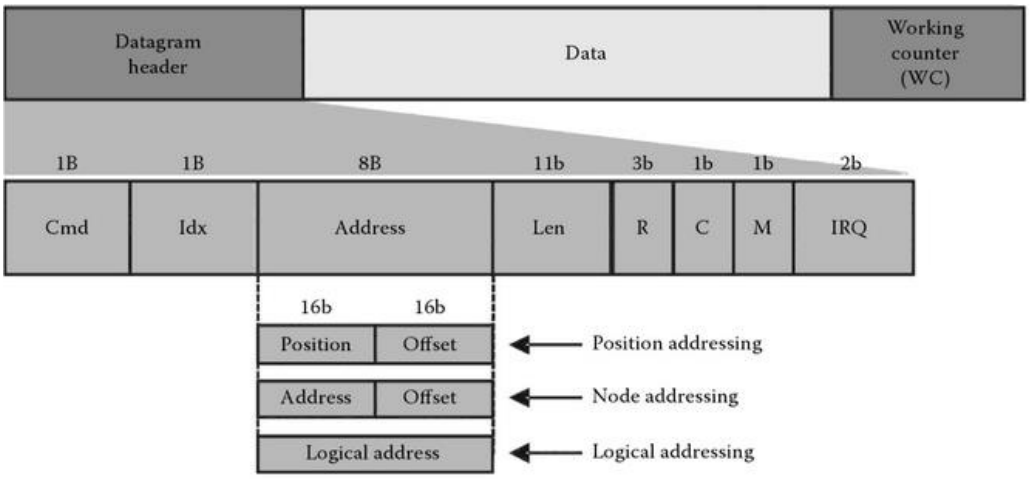


Slika 7. Sadržaj EtherCAT datagrama [6]

3.2.1. Adresiranje

Podržana su dva načina adresiranja podređenih uređaja: fizičko i logičko adresiranje. Za oba načina se koriste zaglavlja DLPDU-a (EtherCAT datagrama) koji je veličine 32 bita [6].

Kod fizičkog adresiranja, polje adrese se dijeli na dva dijela, tzv. „riječi" od 16-bita. Prvi se odnosi na uređaj koji se adresira (ADP), dok drugi na fizičku lokaciju memorije ili registra u podređenom uređaju (ADO).



Slika 8. Načini adresiranja EtherCAT datagrama [6]

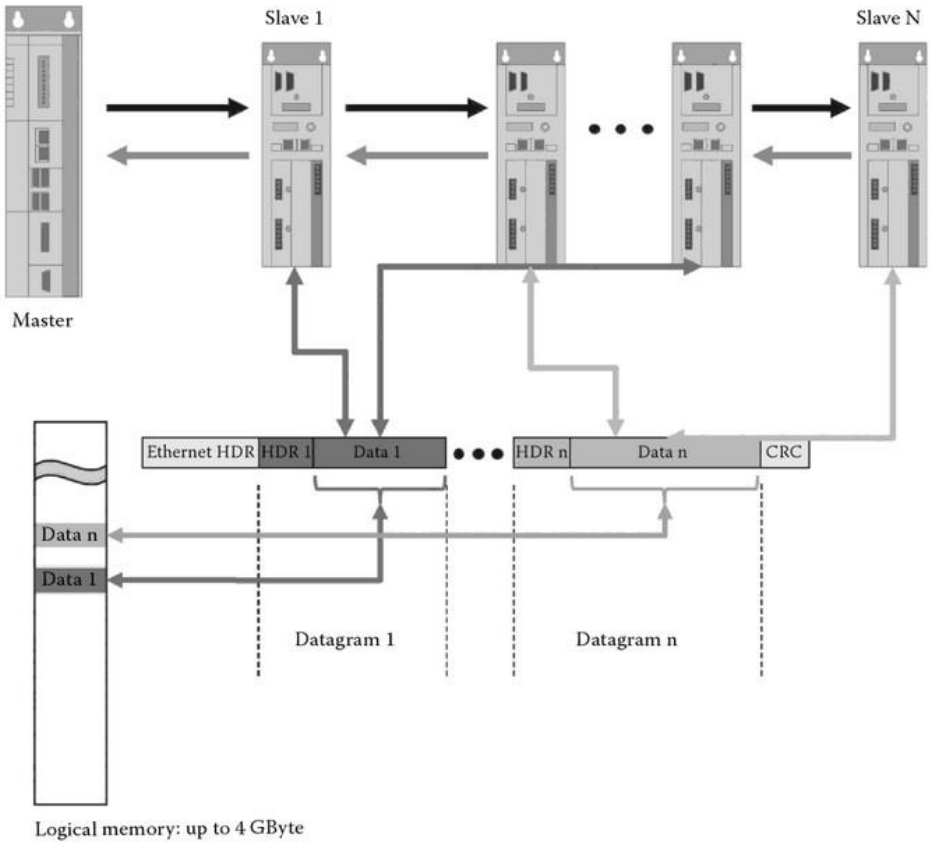
Ovakvo rješenje omogućava do 2^{16} uređaja u mreži. Dakle svaki datagram koji se šalje unutar okvira, referira se na točno određeni dio memorije ili registra podređenog uređaja. Zbog toga je prikladan za prijenos informacija o parametrizaciji uređaja jer svaki parametar ima svoje mjesto u memoriji [6].

Unutar samog fizičkog načina adresiranja postoje još dva načina [6]:

- automatski inkrementalno (pozicijsko) adresiranje – kao primjer se može navesti sljedeći slučaj: nadređeni uređaj vrijednost ADP-a postavlja na -2 što bi značilo da se datagram odnosi na treći podređeni uređaj u mreži (prvi uređaj ima adresu 0). Prilikom prolaska okvira kroz uređaje, oni tu vrijednost ADP-a inkrementalno povećaju za 1. Kada uređaj primi okvir s adresom 0, pretpostavlja se da je okvir na odgovarajućem odredištu. Problem se može javiti ukoliko negdje u mreži dođe do kvara i kao posljedica se javlja krivo brojanje.
- konfiguracijsko adresiranje – podređeni uređaji se identificiraju prema adresama čvorova, tj. prema adresama konfiguriranim prilikom pokretanja uređaja. Te adrese se uspoređuju s adresom u ADP polju datagrama. Ovime se eliminira moguća pogreška ako negdje u mreži dođe do kvara jer adrese ostaju fiksne.

Kod logičkog adresiranja je podatak koji se izmjenjuje, identificiran adresom koja ima veličinu od svih 32-bita adresnog polja zaglavlja datagrama (ADR). Zahtjeva tzv. *„Fieldbus Memory Management Unit“* ili FMMU koji je implementiran u hardver ESC-a. Zadužen je za prevođenje logičke adrese u fizičku adresu lokalne memorije uređaja. FMMU omogućuje

mapiranje individualnih adresa za svaki uređaj. Npr. 2 bita podataka se mogu mapirati individualno i praktički bilo gdje u prostoru logičke adrese. Ako je EtherCAT naredba poslana da se čita/piše u područje logičke memorije koje je dodijeljeno tim dvama bitovima, FMMU će u uređaju biti u mogućnosti prebaciti 2 bita podataka iz memorije i u točnu poziciju u podatkovnom okviru EtherCAT datagrama ili obrnuto ako se radi o upisivanju podatka u podređeni uređaj. Isto se događa i drugim podređenim uređajima kada koriste logičku adresu unutar istog datagrama. Drugim riječima, ako se više ne adresiraju cijeli datagrami za jedan uređaj nego specifična mjesta unutar datagrama, onda se omogućuje da više uređaja može koristiti logičke adrese unutar istog datagrama. Ako se jednim datagramom može obuhvatiti više podređenih uređaja tada dolazi do povećanja učinkovitosti komunikacije (slika 9) [6].



Slika 9. Izmjena podataka (logičko adresiranje uređaja) [6]

3.2.2. SyncManager

DPRAM memoriji ESC-a pristupa nadređeni uređaj, ali i mikrokontroler podređenog uređaja. S obzirom na takvu situaciju, mogli bi se javiti problemi ukoliko ne bi postojalo nikakvih

restrikcija. Kao što je spomenuto ranije, konzistentnost podataka bi se dovela u pitanje ako se ne bi uveo neki mehanizam, tj. „semafor“ koji će koordinirati izmjenu podataka [6].

EtherCAT nudi kontrolu pristupa memoriji podređenog uređaja uz pomoć *SyncManagera*. Oni su implementirani u hardveru ESC-a, a konfiguriraju se od strane nadređenog uređaja. Može se odabrati smjer komunikacije kao i način komunikacije. Svaki SyncManager koristi međumemoriju (eng. *buffer*) u lokalnoj memoriji (DPRAM-u) za izmjenu podataka i transparentno kontrolira pristup međumemoriji. Dva načina komunikacije pomoću SyncManagera su [6]:

1. Memorijskim međuspremnikom (eng. *Buffer*) – interakcija između elementa koji zapisuje (generator podataka) i elementa koji čita podatke (korisnik podataka) nije ograničena, dakle mogu pristupiti međumemoriji u bilo koje vrijeme. Onaj koji čita, uvijek dobiva najnovije podatke jer u slučaju da se u međumemoriju podaci brže upisuju nego što se čitaju, stari podaci se jednostavno odbace. Ovaj način se koristi za cikličku izmjenu podataka.
2. Sandučić (eng. *mailbox*) način – za izmjenu podataka se implementira mehanizam „rukovanja“ kojim se sprječava ranije spomenuto prepisivanje podataka i osigurava se da niti jedan podatak neće biti izgubljen. Za svaki sandučić postoji samo jedan memorijski međuspremnik, a pisanje i čitanje nije moguće istovremeno.

   Način na koji funkcionira je slijedeći: kada se želi upisati neki podatak, generator upisuje podatke u memorijski međuspremnik sandučića (eng. *mailbox buffer*). Kada se upisivanje završi, SyncManager „zaključava“ sandučić, tj. onemogućava pisanje i omogućava čitanje podataka od strane korisnika. Tek kada korisnik pročita sve podatke, generatoru se ponovno omogućava pisanje, a korisniku onemogućava čitanje. Ovaj način se uobičajeno koristi za acikličke podatke kod kojih nije bitno vrijeme izmjene podataka – npr. postavljanje parametara.

3.3. Distribuirani satovi

Metodom distribuiranih satova (eng. *Distributed Clocks*) se ostvaruje visokoprecizna vremenska sinkronizacija između podređenih uređaja u EtherCAT mreži. Kako u procesu sinkronizacije sudjeluju interni satovi (brojači) kontrolera podređenog uređaja (ESC), sinkronizacijsko vrijeme može biti zagarantirano u intervalu koji je manji od 1 μs. Potreba za

ovom metodom ovisi o tome koliko je važna sinkronost uređaja. Primjerice, kod strojeva kod kojih su podređeni uređaji servo regulatori koji upravljaju gibanjem osi, a osi se moraju gibati tako da im je rezultantno gibanje po nekoj točno određenoj krivulji, sinkronost je prijeko potrebna [7].

3.4. Aplikacijski sloj

Aplikacijski sloj implementira, tzv. automat stanja EtherCAT uređaja (eng. *State Machine*) te ostvaruje komunikaciju korištenjem jednog od standardiziranih komunikacijskih protokola.

Važna karakteristika EtherCAT-a je ta da pomoću standardiziranih sandučića podržava nekoliko protokola za komunikaciju na višoj razini. Zastupljeniji protokoli koje podržava su [6]:

1. *CANopen over EtherCAT* (CoE)
2. *Ethernet over EtherCAT* (EoE)
3. *File access over EtherCAT* (FoE)
4. *Servo drive profile over EtherCAT* (SoE)

Činjenica da EtherCAT podržava popularne komunikacijske protokole, pomaže poboljšanju kompatibilnosti i iskoristivosti izmjene podataka između novih i starih komponenti u nekom automatiziranom sustavu [6].

Servo regulatori koji su dio ispitnog CNC postava, najčešće koriste CoE komunikacijski protokol koji omogućuje iste mehanizme komunikacije kao i *CANopen* standard.

Rječnik objekata

Središnji pojam CANopen-a je rječnik objekata (eng. *object dictionary,* OD). On se može shvatiti kao tablica u kojoj se pohranjuju konfiguracijski i procesni podaci. Standard definira mogućih 2^{16} ili 65536 adresa i 2^{8} ili 256 podadresa. Neki su indeksi u rječniku obavezni, kao npr. tip uređaja (indeks 1000). Pomoću rječnika objekata se ustvari može komunicirati s nekim podređenim uređajem odnosno postoji mogućnost pristupanja parametrima nekog uređaja. Također je moguće da nadređeni uređaj želi pročitati podatak iz rječnika objekta ili saznati kako je uređaj trenutno konfiguriran. Za pristup rječniku se koriste dva mehanizma: SDO (eng. *Service Data Objects*) i PDO (eng. *Process Data Objects*) [8].

SDO

SDO je mehanizam za direktan pristup (čitanje/pisanje) rječniku objekata. Podređeni uređaj čijem rječniku se pristupa, naziva se SDO poslužitelj (eng. *SDO server*), a nadređeni uređaj koji želi pristupiti je SDO klijent (eng. *SDO client*). Kada klijent želi pristupiti rječniku, serveru šalje naredbu sa SDO zahtjevom. U podatkovnom polju datagrama će biti definiran indeks i podindeks objekta kojem nadređeni želi pristupiti. Ciljani podređeni uređaj odgovara i šalje tražene podatke [8].

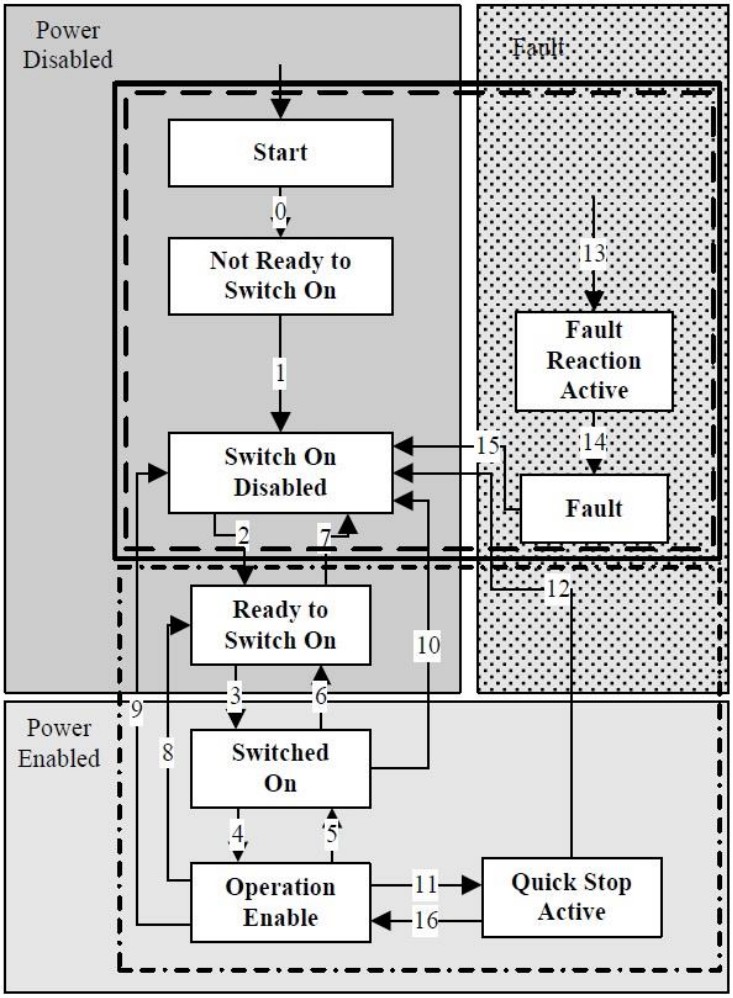
PDO

Procesni podaci su podaci koji se mijenjaju u vremenu. Primjeri takvih podataka mogu biti podaci iz senzora koji dolaze u nadređeni uređaj ili izlazni podaci nadređenog uređaja koji odlaze u regulator elektromotora itd. Ti podaci se isto tako pohranjuju u rječniku objekta. Ovaj mehanizam pristupa rječniku postoji jer SDO mehanizam dozvoljava pristup samo jednom indeksu u jednom ciklusu. To je vremenski nepovoljno za podatke koji se kontinuirano ciklički mijenjaju. Uz to, podređeni uređaj mora moći slati podatke bez da to prvo od njega mora zahtjevati nadređeni uređaj [8].

PDO podaci se dijele na: prijenosne PDO (eng. *transfer PDOs*, TPDO) i prijemne (eng. *receive PDOs*, RPDO). Kao što im imena sugeriraju, prijenosni podaci su izlazni podaci podređenog uređaja, a prijemni su ulazni podaci podređenog uređaja.

CANopen in Automation (CiA) 402 protokol

CiA 402 je protokol kojim se standardiziraju ponašanja servo regulatora, kao i frekventnih pretvornika i koračnih motora. Uvodi nekoliko stanja rada koji se mijenjaju na zahtjev nadređenog uređaja. Ta stanja je moguće kontrolirati upisivanjem u kontrolni registar (objekt pod nazivom *control-word*). Uz zahtjev, stanja se kontrolnog registra mogu još mijenjati i na temelju internih događaja. Kontrolni registar se najčešće zajedno s ostalim registrima važnim za upravljanje pogonskim motorom mapiraju u RPDO poruke. S druge strane, postoji i statusni registar (objekt pod nazivom *status-word*) kojim se opisuje trenutno stanje regulatora. Statusni registar se također zajedno s ostalim procesnim podacima najčešće mapira u TPDO poruke [9].



Slika 10. Stanja opisana CiA 402 protokolom [9]

Automat stanja definiran CiA 402 protokolom prikazan je na slici 10 [9], [10].

1. Onemogućeno energetsko stanje (eng. *Power Disabled*)
   1.1. Start
   1.2. Nije spreman za uključivanje (eng. *Not Ready To Switch On*)
   - servo regulator se inicijalizira
   1.3. Onemogućeno uključivanje (eng. *Switch On Disabled*)
   - inicijalizacija završena
   1.4. Spreman za uključivanje (eng. *Ready To Switch On*)
   - servo regulator ulazi u stanje *Switched On*, ali upravljanje motorom nije moguće

2. Omogućeno energetsko stanje (eng. *Power Enabled*)
   2.1. Uključen (eng. *Switched On*)
      - servo regulator je spreman i dovedena je energija motoru
   2.2. Omogućen rad (eng. *Operation Enable*)
      - servo regulator upravlja radom motora prema određenom načinu kontrole
   2.3. Brzo zaustavljanje (eng. *Quick Stop Active*)
      - servo motor se zaustavlja na unaprijed određen način, može se ponovno prijeći u podstanje omogućenog rada i nastaviti gibanje
3. Greška (eng. *Fault*)
   3.1. Aktivna greška (eng. *Fault Reaction Active*)
      - ukoliko bi se javila greška, servo pogon automatski odlazi u stanje onemogućenog rada i da bi se došlo u stanje u kojem se servo motor može normalno raditi potrebno je ponoviti postupak pokretanja motora
   3.2. Greška

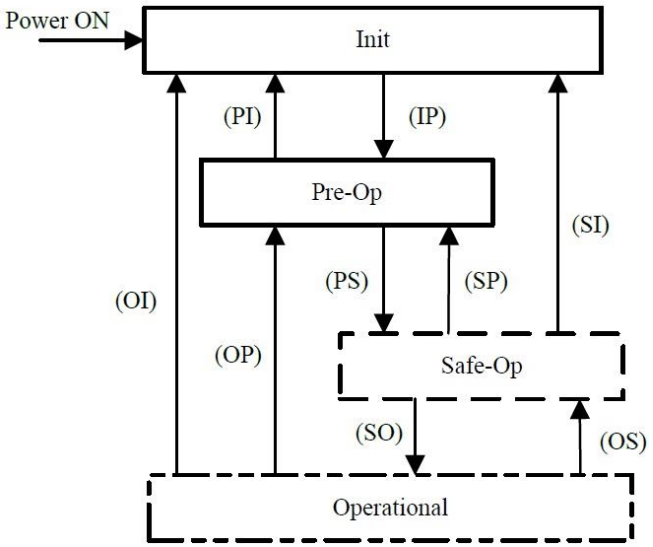
Automat stanja EtherCAT uređaja (eng. *State Machine*)

Automat stanja se koristi za opis stanja u kojem se podređeni uređaj nalazi i opis promjene stanja. Promjena stanja je zahtjevana od strane nadređenog uređaja, a na taj zahtjev odgovara podređeni uređaj [11].

Postoje 4 osnovna stanja EtherCAT podređenog uređaja [11]:

1. *Init* – nije moguća niti komunikacija putem sandučića (SDO), niti razmjena procesnih podataka (PDO)
2. *Preoperational* – komunikacija SDO porukama je moguća, ali PDO poruka još nije
3. *Safe operational* – PDO komunikacija je omogućena od podređenog prema nadređenom uređaju
4. *Operational* – PDO komunikacija moguća u oba smjera

Redoslijed stanja može se vidjeti na slici ispod (slika 11).



Slika 11. Automat stanja EtherCAT uređaja [11]

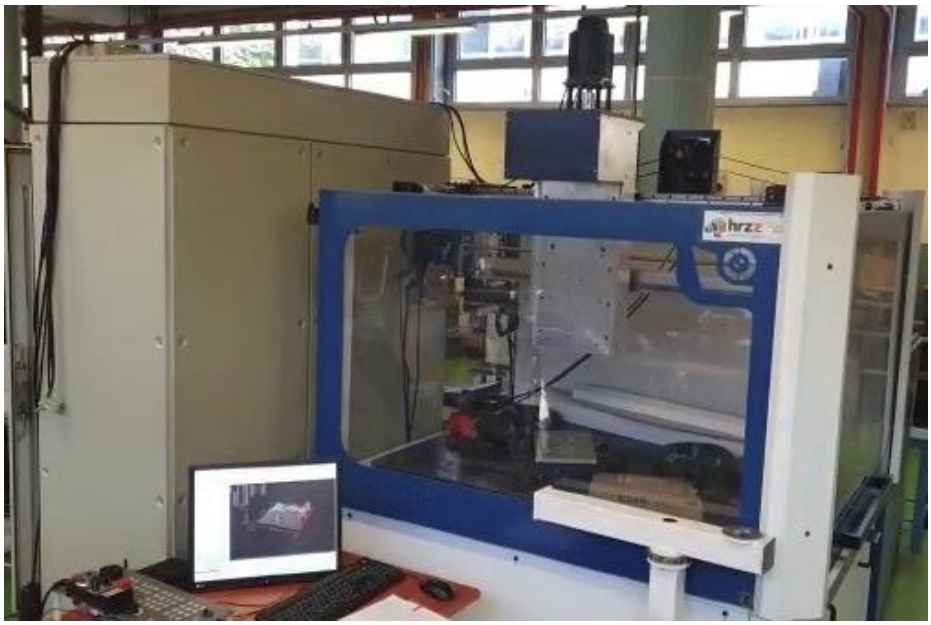
U predoperacijskom stanju uređaja, SDO podaci su mogući jer se koriste za inicijaliziranje parametara u, tzv. rječnik objekata i taj proces nije vremenski osjetljiv, dok se PDO podaci koriste za prijenos podataka o procesu koji se kontinuirano osvježavaju u stvarnom vremenu i prilikom paljenja uređaja nisu toliko važni [8].

3.5. Konfiguracija

Svaki EtherCAT podređeni uređaj mora imati svoju informacijsku datoteku (eng. *EtherCAT Slave Information,* ESI). Datoteka je zapisana u XML formatu i opisnog je karaktera. Sadrži informacije o funkcijama i postavkama uređaja te rječnik objekata. Koristi se pri opisu mreže, tj. pri stvaranju njezine informacijske datoteke (eng. *EtherCAT Network Information*, ENI) u *offline* načinu rada. ENI datoteka sadrži opis topologije mreže, inicijalizacijske naredbe, naredbe koje se moraju slati ciklički i sl. [7].

4. EKSPERIMENTALNI DIO

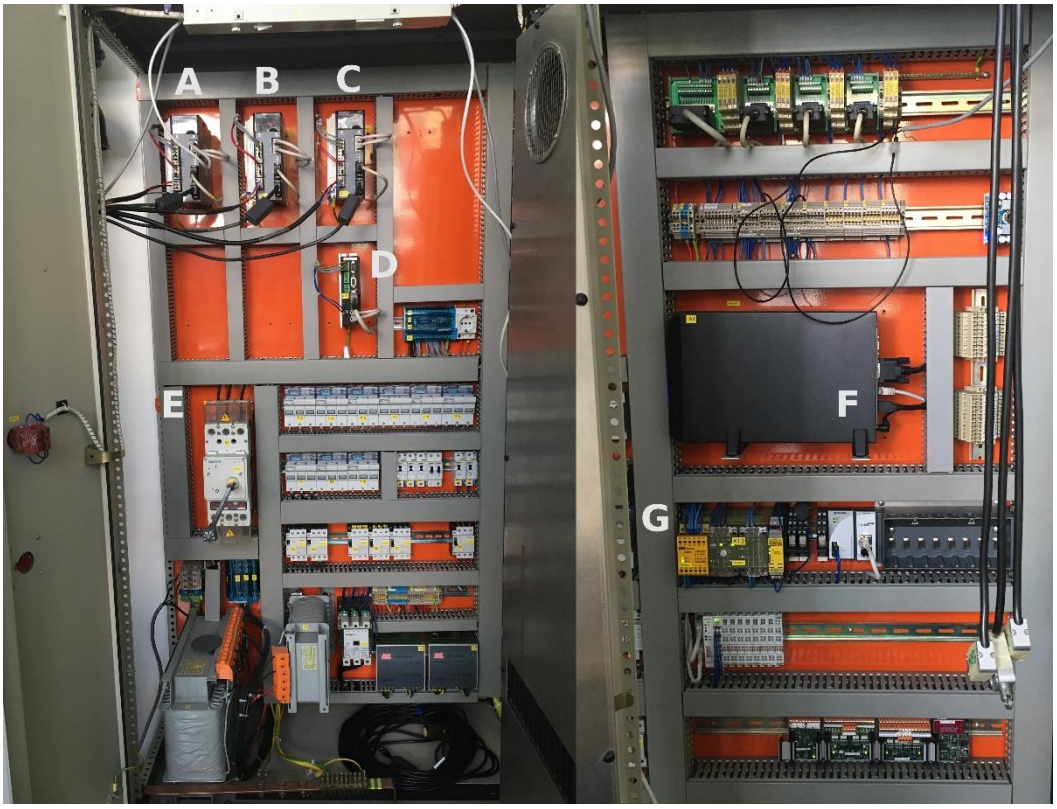
CNC stroj korišten u eksperimentalnom dijelu rada prikazan je slikama ispod (slika 12 i slika 13). Upravljačko računalo je bilo povezano sa servo regulatorima posmičnih prigona putem EtherCAT sabirnice. U cilju cjelovite realizacije upravljačkog sustava bilo je potrebno izraditi odgovarajuće konfiguracijske datoteke. U njima su inicijalizirani odgovarajući HAL moduli s odgovarajućim logičkim vezama (HAL signalima) koji opisuju strukturu upravljačkog sustava. Konfiguracijske datoteke također sadrže i parametre nužne za ispravno funkcioniranje ispitnog CNC uređaja.



Slika 12. Ispitni CNC postav

4.1. Oprema za rad

Upravljački ormar



Slika 13. Dijelovi upravljačkog ormara ispitnog CNC postava

Na slici iznad mogu se vidjeti oznake dijelova ormara:

A = regulator X osi

B = regulator Y osi

C = regulator Z osi

D = regulator glavnog prigona

E = izvori napajanja i osigurači

F = upravljačko računalo

G = logički krug za zaustavljanje u nuždi

Nadređeni uređaj (upravljačko računalo)

Na slici iznad se, označeno slovom F, može vidjeti upravljačko računalo. Na računalu su instalirani: operativni sustav s Linux jezgrom [12], LinuxCNC [13], IgH EtherCAT Master za konfiguraciju nadređenog uređaja [14], LinuxCNC-ethercat [15] i EtherCAT HAL driver [15]. Upravljačko računalo ima ulogu nadređenog uređaja u mreži.

Podređeni uređaji

Podređeni uređaji su servo regulatori ESTUN ProNet – 10 A E A – EC (slika 14). Iz naziva uređaja se mogu dobiti sljedeće informacije [16]:

10 – izlazna snaga elektromotora je 1 kW

A – uređaj se napaja izmjeničnom strujom od 200 V

E – podržava upravljanje pozicijom, brzinom i momentom

A – koristi 17-bitni apsolutni enkoder

EC – koristi EtherCAT komunikacijski modul

Slika 14. Servo regulator

Značenje oznaka priključaka na servo regulatoru su sljedeće:

Tablica 1. Opis priključaka servo regulatora [11]

Oznaka priključka	Opis
L1, L2, L3	napajanje glavnog strujnog kruga (200 ~ 230VAC; 50/60 Hz)
L1C, L2C	napajanje regulacijskog kruga (200 ~ 230VAC; 50/60 Hz)
B2 - B3	regenerativni otpornik
U, V, W	napajanje servomotora (tri faze; 200V; snaga 1 kW)
CN3, CN4	komunikacijski priključci
CN1	priključak za I/O signale
CN2	priključak za enkoder

EtherCAT komunikacijski modul

EtherCAT komunikacijski modul služi za povezivanje upravljačkog računala i servo regulatora posmičnih prigona. Specifikacije se mogu vidjeti u tablici (Tablica 2) koja slijedi.

Tablica 2. Specifikacije EtherCAT modula [16]

Komunikacijski standard	IEC 61158 Type12, IEC 61800-7 CiA402 Drive Profile
Fizički sloj	100BASE-TX (IEEE802.3)
Sučelje sabirnice	CN4 (RJ45): EtherCAT ulaz signala CN5 (RJ45): EtherCAT izlaz signala
Kabel	Cat 5 kabel s upletenom paricom
SyncManager	SM0: izlazni sandučić, SM1: ulazni sandučić SM2: izlazni procesni podaci, SM3: ulazni procesni podaci
FMMU	FMMU0: mapiranje u memoriju za primanje procesnih podataka (RxPDO) FMMU1: mapiranje u memoriju za slanje procesnih podataka (TxPDO) FMMU2: mapiranje statusa u sandučić
PDO podaci	dinamično PDO mapiranje
Sandučić (CoE)	poruke upozorenja, SDO zahtjev, odgovor, SDO informacija
CiA402 profil prigona	*homing* profil, pozicijski, brzinski, ciklički sinkroni pozicijski profil

Elektromotori

Elektromotori koji su se koristili su asinkroni servomotori proizvođača ESTUN. Oznake su:

1. Z-os: EMG – 10ASA24

Slika 15. Natpisna pločica elektromotora Z-osi

Značenje oznaka:

10 – nazivna snaga je 1,0 kW

A – uređaj se napaja izmjeničnom strujom od 200V

S – koristi 17-bitni apsolutni enkoder

A – sekvenca konstruiranja

2 – izvedba kraja osovine

4 – dodatni dijelovi: s uljnom brtvom, s kočnicom (DC24V)

Karakteristike modula su sljedeće: nazivni broj okretaja je 2000 okr/min, najveći broj okretaja je 3000 okr/min, nazivni moment je 4,78 Nm, nazivna jačina struje je 6,0 A, motor je trofazni te je namijenjen za trajni pogon (oznaka S1) i moment kočnice je 12 Nm [16].

2. X i Y – os: EMG – 10ASA22

Slika 16. Natpisna pločica elektromotora X i Y osi

Karakteristike su jednake kao i za prethodni elektromotor izuzev toga što nemaju kočnicu te je zato u oznaci zadnja znamenka 2.

4.2. Tijek rada

4.2.1. Izrada konfiguracijske datoteke za EtherCAT

U ovom koraku su definirane SDO poruke kojima će se postavljati parametri unutar određenih indeksa rječnika objekata servo regulatora. Također je određeno kako će se mapirati PDO registri. Konfiguracija je za sve servo regulatore jednaka osim oznake indeksa podređenog uređaja. Stoga će se ukratko opisati konfiguracija za servo regulator osi X. Kao konačni rezultat dobiva se datoteka u XML formatu.

- ESTUN ProNet regulator (indeks 0)

Parametriranje konektora CN1

Prvo su određeni parametri kojim su se kodirale funkcije ulaza na konektoru CN1 (priključak za I/O signale).

Zatim je određen parametar kojim se kodira koji je od ulaznih signala na konektoru CN1 potrebno invertirati.

Parametri referenciranja

Ovdje se odredila metoda kojom se odrađuje referenciranje. Zatim su se odredile brzine kojom se traži granični prekidač te brzina kojom se traži referentni prekidač. Iako su posmični prigoni u konačnici parametrirani kao prigoni s apsolutnom mjernim sustavom gdje referenciranje nije potrebno, korišteni regulatori podržavaju mogućnost parametriranja prigona s emulacijom inkrementalnog mjernog sustava. Stoga su ove operacije uvedene kako bi se prigoni mogli testirati i s inkrementalnim mjernim sustavom.

Parametri za faktore skaliranja

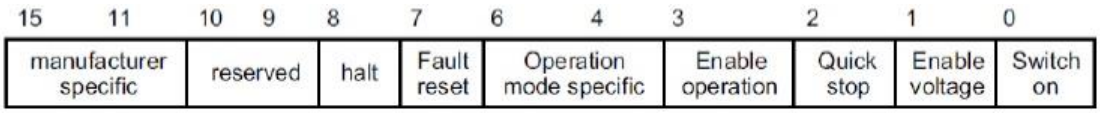
Skaliranje procesnih veličina općenito je potrebno kako bi se upravljačke veličine, kao i ostale relevantne informacije koje se izmjenjuju između CNC upravljačkog računala i regulatora prigona svele na veličine SI sustava. Proces je proveden za jakost struje motora, izmjerenu i zadanu brzinu gibanja prigona, te za izmjerenu poziciju prigona.

Konstrukcija ispitnog CNC postava realizirana je na način da su enkoderi motora korišteni kao indirektni mjerni sustav (na CNC stroju nisu ugrađeni direktni mjerni sustavi). Stoga je vrijednost pozicije enkodera, čija je zadana vrijednost izvorno iskazana brojem impulsa, bilo potrebno iskazati u milimetrima, dok je brzinu gibanja motora bilo potrebno skalirati iz frekvencije vrtnje u milimetre po sekundi. Isto tako, referentnu vrijednost gibanja prigona trebalo je svesti na milimetre u sekundi.

Mapiranje PDO registara

Prvi i drugi Sync Manager određeni su za sandučić, tj. razmjenu SDO podataka koji se ne izmjenjuju u stvarnom vremenu. Prvi je zadužen za primanje podataka, a drugi za slanje podataka prema nadređenom uređaju.

Treći Sync Manager je zadužen za dolazne PDO podatke i njihovo mapiranje u memoriju uređaja, tj. u rječnik objekata. U rječnik objekata se mapira i kontrolni registar kojim se upravlja radnim stanjem podređenog uređaja. Od čega se sve sastoji kontrolna riječ može se vidjeti na slici ispod (slika 17).



Slika 17. Struktura control word naredbe [11]

Bitovi koji utječu na promjenu stanja stroja su bitovi 0, 1, 2, 3 i 7. Različite kombinacije jedinica i nula predstavljaju drugačija stanja uređaja. Značenje bitova od 4 do 6 ovisi o načinu rada elektromotora (pozicijski, brzinski, referenciranje). 8. bit zaustavlja rotaciju elektromotora, 9. i 10. su rezervirani, a značenje bitova od 11 do 15 ovisi o proizvođaču [11].

Također je mapiran i parametar čijim se namještanjem pomoću PDO poruke određuje način rada (eng. *modes of operation*). Upisivanjem odgovarajuće vrijednosti u taj registar moguće je tijekom rada parametrirati regulator na način da isti radi u zatvorenoj regulacijskoj petlji po momentu, frekvenciji vrtnje ili po poziciji.

U rječnik objekata se mapiraju i referentne vrijednosti (eng. *target*) pozicije, brzine i momenta. U zavisnosti o vrijednosti registra „modes of operation“, regulator će slijediti jednu od spomenutih referentnih vrijednosti.

Četvrti Sync Manager je zadužen za lociranje odgovarajućih PDO podataka unutar rječnika objekata i njihovo slanje prema nadređenom uređaju. U rječniku objekata nalazi se i statusni registar koji nadređenom uređaju daje povratnu informaciju o radnom stanju podređenog uređaja. Slično kao i kod kontrolne riječi se određenim bitovima, te njihovim različitim kombinacijama jedinica i nula predstavljaju drugačija stanja uređaja.

Mapira se i povratna informacija o tome u kojem načinu rada radi elektromotor (eng. *modes of operation display*).

Mapiraju se povratni podaci o stvarnom (eng. *actual*) iznosu pozicije, brzine i momenta koji se također skaliraju, kako bi se dobile veličine u SI mjernim jedinicama.

4.2.2. Konfiguracijska datoteka za LinuxCNC

Prvo se izradila konfiguracijska datoteka koja se navodi pri pokretanju LinuxCNC-a. Datoteka je podijeljena na sekcije od kojih svaka sadrži informacije koje se koriste za inicijalizaciju upravljačkog sustava. Sekcije koje su se koristile su [4]:

[EMC]

Sadrži informaciju o verziji konfiguracijske datoteke, imenu stroja i pruža mogućnost odabira da li da se u terminalu ispisuju poruke kada se LinuxCNC pokrene u svrhu otklanjanja eventualnih grešaka.

[DISPLAY]

Sekcija opisuje grafičko sučelje koje se koristi. Može se odrediti vrsta sučelja (*Axis, Gmocappy, Touchy, Gscreen* i dr.), koja je pozicija GUI-ja, primjerice na monitoru računala, koje informacije će se prikazivati itd.

[TASK]

U ovoj sekciji se definiraju parametri koji uvjetuju ponašanje modula EMCTASK: komunikacija s korisničkim sučeljem pomoću NML-a, komunikacija s planerom gibanja koji se izvršava u stvarnom vremenu (EMCMOT modul) i interpretacija G-koda.

[RS274NGC]

U sekciji se definira datoteka koja sadrži varijable korištene od strane interpretera. U istoj sekciji je moguće postaviti i NC kod koji se koristi pri inicijalizaciji interpretera, a sastoji se od niza G funkcija.

[EMCMOT]

U ovoj sekciji se definiraju parametri EMCMOT modula (*motmod*). Među najvažnijim parametrima je ciklus izvršavanja regulacijskih petlji u stvarnom vremenu, bazni period čiji ciklus mora biti jednak ili brži od ciklusa izvršavanja regulacijskih petlji te vrijeme koje se treba čekati da motmod (dio EMCMOT modula koji se izvršava u stvarnom vremenu) primi poruke od EMCTASK modula.

[HAL]

Sekcija u kojoj se navodi jedna ili više glavnih HAL datoteka u kojima može biti jedna ili više HAL poddatoteka, a kojima se učitavaju HAL moduli i signali koji povezuju izvodnice HAL modula. Također se uvodi modul *HALUI* koji u HAL prostor dodaje dodatne izvodnice kojima je moguće upravljati strojem putem tipkala i ostalih fizičkih elemenata.

[KINS]

Pomoću ove sekcije se definira kinematika CNC stroja. U ovom slučaju korištena je trivijalna kinematika (stupnjevi slobode gibanja stroja se poklapaju s osima pravokutnog Kartezijevog koordinatnog sustava). Također je definiran i broj stupnjeva slobode gibanja (može se reći i broj motora) u sustavu.

[EMCIO]

U ovoj sekciji se navodi ime programa koji upravlja ulaznim i izlaznim signalima, vrijeme ciklusa rada te datoteka koja sadrži informacije o alatima.

[TRAJ]

Sekcija u kojoj se imenuju osi i zglobovi, odnosno stupnjevi slobode gibanja stroja kojima se upravlja. Također se definira koju mjernu jedinicu za linearne osi koristi stroj, a koju za rotacijske osi. Navodi se zadani iznos brzine gibanja i akceleracije linearnih osi pri njihovom ručnom pokretanju (eng. *jog*) te najveći iznosi brzine gibanja i akceleracije u slučaju kada više osi radi istovremeno.

[AXIS_X]

U ovoj sekciji se određuju donja i gornja granica položaja osi stroja od referentne točke te maksimalna brzina gibanja i akceleracije osi.

[JOINT_0]

Potreba za stvaranjem [JOINT_n] sekcije uz prethodnu [AXIS_n] sekciju proizlazi iz toga što postoje strojevi u kojima se primjerice, gibanje po jednoj osi ostvaruje pomoću više elektromotora (robotske ruke i sl.) ili iz slučaja kada je stroj izveden netrivijalnom kinematikom. U slučaju da je broj osi, npr. tri, a jedna os je izvedena s dva elektromotora, broj zglobova (eng. *joints*) ili elektromotora bi u tom slučaju bio četiri.

U ovom slučaju svaku os pogoni jedan elektromotor te se u sekciji određuje o kojoj se vrsti osi radi – linearnoj ili rotacijskoj, najveća brzina i akceleracija prigona, iznos parametra možebitne zračnosti (eng. *backlash*) (zazor između boka navoja matice i boka navojnog vretena po kojem se matica giba pri posmičnom gibanju koji s vremenom postaje sve veći zbog djelovanja trenja), najveći dopušteni iznos greške praćenja, dopušteni iznos greške praćenja pri jako niskim brzinama, faktori skaliranja podataka koji se primaju/šalju (u ovom slučaju su jednaki jedan zbog toga što su prethodno u konfiguracijskoj datoteci podređenih uređaja ti faktori već uvedeni), gornja i donja granica položaja osi od referentne točke (moraju odgovarati onima u [AXIS_X]. Također se određuju i iznosi pojačanja PID regulatora (proporcionalno, integralno i derivacijsko pojačanje, unaprijedno pojačanje pozicije (FF0), brzine (FF1) i akceleracije (FF2), itd.).

[AXIS_Y], [JOINT_1], [AXIS_Z], [JOINT_2]

U ovim sekcijama se za Y i Z os određuju vrijednosti istih parametara kao i za X os.

[ECAT]

Sekcija [ECAT] je naknadno uvedena radi lakšeg adresiranja podređenih EtherCAT uređaja u konfiguracijskoj datoteci.

4.2.3. Oblikovanje HAL prostora

U konfiguracijskoj datoteci LinuxCNC upravljačkog sustava se u sekciji [HAL] navodi minimalno jedna glavna HAL datoteka. U njoj se navode sve potrebne HAL poddatoteke kako bi se stvorio funkcionalni HAL prostor. Takvom raspodjelom koda u različite HAL datoteke se postiže olakšana rekonfigurabilnost sustava.

Osnovni dio konfiguracije mora uključivati linije kojima se učitavaju sljedeći moduli koji se izvršavaju u stvarnom vremenu:

1. motmod ili kontrolere gibanja (eng. *joint controller*) koji su dio EMCMOT modula zaduženog za upravljanje gibanjem svih osi (slika 2 u 2. poglavlju prikazuje strukturu EMCMOT modula), a broj kontrolera ovisi o broju elektromotora s upravljanjem u zatvorenom krugu (regulacijom)
2. PID regulatore čiji broj mora odgovarati broju elektromotora s upravljanjem u zatvorenom krugu (regulacijom)
3. driver za EtherCAT
   - za svaki ciklički procesni podatak koji se mapira u memoriju podređenog uređaja se također stvara odgovarajuća izvodnica u EtherCAT driver HAL modulu. Broj izvodnica koje će nastati u sklopu modula ovisi o broju konfiguriranih EtherCAT uređaja, tj. o broju TPDO i RPDO poruka konfiguriranih za svaki EtherCAT uređaj u konfiguracijskoj datoteci.

Na primjeru osi X će se dati objašnjenje povezivanja modula koji sačinjavaju HAL prostor.

Pri paljenju stroja modul EMCTASK koji je zadužen za koordinaciju rada podređenih modula (EMCMOT i EMCIO) šalje naredbu EMCMOT modulu za paljenje servo regulatora. Kontroler gibanja osi X sa svoje izvodnice šalje signal za dozvolu gibanja servo regulatora osi X na izvodnicu drivera za EtherCAT (cword-switch-on) te vrijednost izvodnice postavlja na

iznos jedan. Servo upravljanje će biti moguće ako je uz izvodnice cword-quick-stop, cword-enable-voltage i cword-enable-operation postavljene u stanje jedan, stanje izvodnice cword-switch-on također postavljeno u stanje jedan, a rad se onemogućuje postavljanjem u stanje 0. Kontrolna riječ se upisuje u EtherCAT datagram te šalje u mrežu, zatim dolazi do servo regulatora osi X i kada se pročita, stanje regulatora x-osi promijeni se u *„switched on“* što se može vidjeti iz statusnog registra koji se šalje prema nadređenom uređaju, odnosno čitanjem stanja u koje se, na temelju statusnog registra, postavlja izvodnica sword-switched-on. Kontroler gibanja osi X, koji je dio EMCMOT modula, isti signal za dozvolu gibanja servo regulatora šalje i u modul – PID regulator koji ostvaruje regulacijsku petlju X osi po poziciji. Signal dolazi na izvodnicu, stanje izvodnice se mijenja u jedan i regulator se pokreće.

```
net j0-cw-switch-on <= joint.0.amp-enable-out
net j0-cw-switch-on => pid.0.enable
```

Slika 18. Primjer koda za pokretanje PID regulatora

Regulator pozicije osi na izvodnicu na koju se dovodi povratna informacija dobiva informaciju o stvarnoj poziciji osi. Ta informacija se na izvodnicu dovodi putem signala koji kao ulaz ima izvodnicu EtherCAT driver modula. Izvodnica EtherCAT driver modula informaciju o stvarnoj poziciji osi X dobiva od servo regulatora, a servo regulator preko enkodera koji je spojen na elektromotor. Informacija o stvarnoj poziciji X-osi kao i ostale informacije koje su mapirane u rječniku objekata u memoriji podređenog uređaja se ciklički izmjenjuju između podređenog i nadređenog uređaja.

Osim u regulator, povratna informacija o stvarnoj poziciji osi se iz EtherCAT driver modula istim signalom šalje prema kontroleru gibanja X-osi zbog toga što proces računanja sljedeće referentne vrijednosti pozicije uključuje i uzorkovanje pozicije osi X. Dobivena referentna pozicija se s izvodnice kontrolera gibanja X-osi šalje signalom prema odgovarajućoj izvodnici za postavljanje referentne vrijednosti pozicije regulatora.

Oduzimanjem povratne informacije od referentne vrijednosti pozicije dobiva se greška slijeđenja te se primjenom pojačanja proračunava naredba brzine kojom se nastoji doći iz trenutne u željenu poziciju. Kod regulatora se kao parametar postavlja i ograničenje na najveću vrijednost izlazne vrijednosti kako bi se spriječilo naglo mijenjanje brzine uslijed čega bi moglo doći do oštećenja mehaničkih komponenti posmičnog prigona.

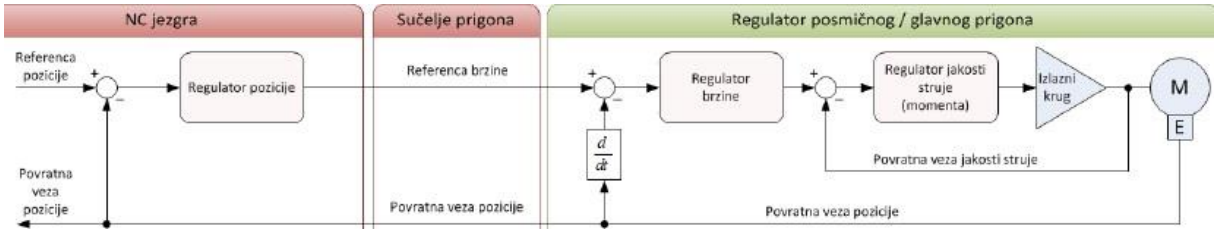
Dobivena vrijednost brzine se signalom od regulatora odvodi u EtherCAT driver modul na izvodnicu za referentnu vrijednost brzine. Referentna vrijednost brzina se u sljedećem ciklusu, s ostalim podacima šalje EtherCAT sabirnicom u obliku TPDO poruke u servo regulator X-osi.

Način povezivanja modula u HAL prostoru za Y i Z-os jednak je onom za X-os.

4.2.4. Određivanje parametara pojačanja regulacijskih petlji

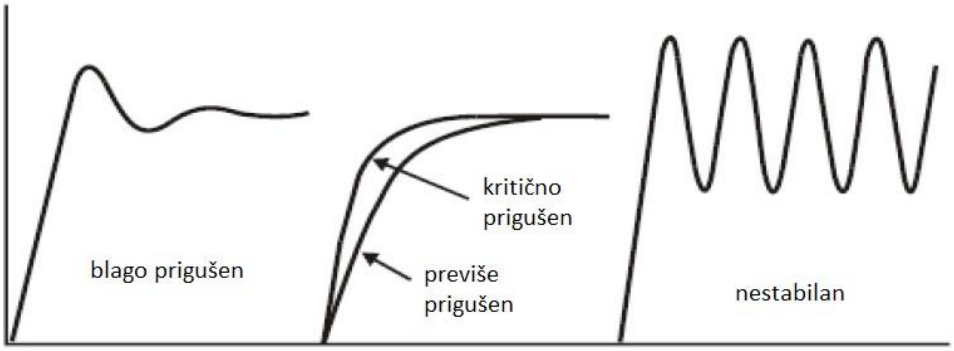
Poziciju i brzinu osi stroja regulira regulacijska petlja koja osigurava točnost i robusnost u pozicioniranju kod širokog raspona brzina. U većini slučajeva se svaka os stroja regulira zasebno i to na temelju reference zadane od odgovarajućeg kontrolera gibanja osi koji je dio EMCMOT modula. Općenito vrijedi da se upravljanje može odvijati u otvorenom i zatvorenom krugu.

Prednosti upravljanja u zatvorenom krugu su te da daje brži odziv, veću točnost u pozicioniranju jer omogućuje otklanjanje poremećaja što je vrlo važna činjenica zato što postoji niz unutarnjih i vanjskih uzroka poremećaja koji se moraju uzeti u obzir. Primjeri uzroka su: utjecaj sila rezanja, zračnosti, trenje, deformacije i naprezanja radnog stola, karakteristike servosustava koje posebice dolaze do izražaja kod visokobrzinskih obrada (u operacijama ubrzavanja/usporavanja, kod promjene smjera) itd. Posljedica tih pojava je greška slijeđenja. Ona se ne može u potpunosti eliminirati, a da bi se dovela na najmanju moguću vrijednost upravljanje se gotovo uvijek realizira u zatvorenom krugu i najčešće primjenom PID regulatora, tj. njegovih varijanti (P-PI, PI-P). Kako bi se postigla željena pozicija i brzina, upravljanje servomotorom je ostvareno u kaskadnoj formi s 3 nezavisne petlje: pozicijska, brzinska i strujna petlja (petlja po momentu). Najčešća izvedba je takva da se pozicijska petlja smješta u NC modul, a petlja po brzini i momentu u servo regulator [17]. Prikaz takve izvedbe može se vidjeti na slici ispod (slika 19).



Slika 19. Izvedba upravljanja servomotorom [18]

Kako bi se postigle zahtijevane karakteristike servo sustava, stabilnost regulacijskog kruga predstavlja nužni uvjet [17]. Sustav se smatra nestabilnim u slučaju kada naredba pozicije rezultira eksponencijalnim porastom greške pozicije, tj. sustav je nestabilan ako pokušaji da se postigne zadana pozicija rezultiraju oscilacijama koje se nikada ne priguše (slika 20; desni dijagram) [19]. Da bi se postigao odziv koji je stabilan, blago prigušen ili kritično prigušen (slika 20; lijevi i srednji dijagram) potrebno je odrediti pojačanja regulacijskog kruga [17].

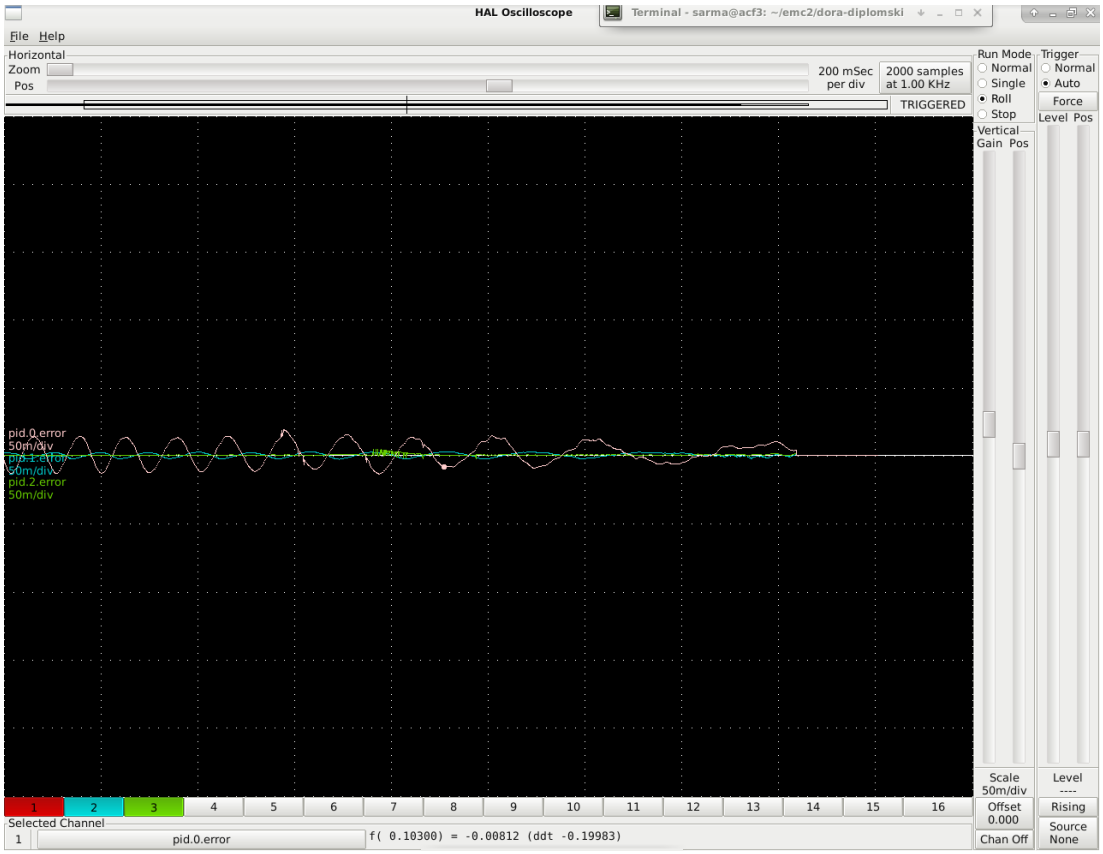


Slika 20. Vrste odziva [17]

Kao opcija postoji automatsko parametriranje regulatora (eng. *auto-tune*), ali kako bi se postigao optimalno parametriran, stabilan sustav najčešće je potrebno provesti ručno parametriranje [19].

Parametriranje pojačanja regulacijske petlje po poziciji

HAL prostor se, za ovu primjenu, konfigurira bez motmod modula koji se nadomještava signal generatorom (HAL modul koji u stvarnom vremenu može generirati više različitih valnih oblika referentnih signala). Signal generator dovodi referentnu poziciju na izvodnicu PID regulatora. Odziv razmatranog regulatora (odn. prigona) promatra se također uz pomoć HAL modula, a to je osciloskop (eng. *halscope*). Na slici ispod (slika 21) se može vidjeti prikaz osciloskopa, a na lijevoj strani je naveden izvor signala. Bijela krivulja prikazuje grešku slijeđenja za X os, plava za Y os, a zelena za Z os. Greška ne prelazi vrijednost od 0,02 mm pri posmičnim brzinama od 10 m/min.



Slika 21. Osciloskop – greška slijeđenja: za X os (bijelo), Y os (plavo) i Z os (zeleno)

Parametri pojačanja se određuju ručno, iterativnom metodom postavljanjem vrijednosti ulaznih izvodnica HAL modula PID regulatora (izvodnica za proporcionalno, integralno i derivacijsko pojačanje te unaprijedna pojačanja) u različita stanja i proučavanjem odziva sustava. Pojačanja regulatora podešavanju se na način da se regulacijsko odstupanje pokuša svesti na minimalnu moguću mjeru.

Također je potrebno pod komentar staviti i liniju koda kojom se signal za stvarnu poziciju dovodi na izvodnicu kontrolera gibanja kako se prilikom učitavanja ne bi javila greška jer modul u ovoj konfiguraciji ne postoji. Potrebno je i odspojiti signal prema izvodnici kojom se pokreće regulator te ručno promijeniti stanje izvodnice u jedan kako bi se modul ipak pokrenuo.

Parametriranje pojačanja regulacijske petlje po brzini

Parametriranje pojačanja brzinske petlje se provelo slično načinu za pozicijsku petlju. Regulator je realiziran hardverski, odnosno unutar servo regulatora, a ne unutar NC jezgre. Da bi se na ulaz u regulacijsku petlju brzine, moglo dovesti nadomjesnu referentnu vrijednost brzine, u HAL prostoru je na odgovarajući signal potrebno umjesto izlaza PID regulatora ponovno dovesti signal iz signal generatora. U osciloskop se kao prvi vod dovodi referentni signal iz generatora signala, a u drugi vod se pomoću signala dovodi povratna informacija servo regulatora o stvarnoj brzini.

Kako brzinska petlja nije smještena u NC modulu već u servo regulatoru, promjena parametara se ne vrši promjenom stanja izvodnica, već postavljanjem različitih vrijednosti određenih parametara servo regulatora. Parametri servo regulatora postavljaju se slanjem SDO poruka. U korištenom regulatoru moguće je podesiti: krutost stroja, pojačanje brzinske petlje te integralnu vremensku konstantu brzinske petlje [11]. Pojačanja se nastoji podesiti tako da odziv regulatora bude što sličniji referentnoj vrijednosti brzine.

Parametriranje pojačanja strujne petlje (petlje po momentu) nije se provelo zbog toga što je regulator jakosti struje tvornički podešen prema odabranom motoru.

4.2.5. Testiranje posmičnih prigona

CNC sustav je nakon parametriranja svih posmičnih osi testiran s nekoliko ispitnih programa. U trenutku realizacije svih posmičnih prigona glavni prigon još uvijek nije bio u cijelosti realiziran, stoga se stroj nije mogao u cjelosti testirati.

5. ZAKLJUČAK

Cilj ovog rada bio je parametrirati regulatore posmičnih prigona ispitnog CNC postava zasnovanog na otvorenom upravljačkom sustavu *LinuxCNC*. U prvom dijelu rada opisana je struktura tog sustava te su detaljnije objašnjeni najvažniji moduli istog. Također je objašnjen i princip rada EtherCAT mreže koja je korištena za povezivanje upravljačkog računala s regulatorima posmičnih prigona.

Drugi dio rada detaljnije opisuje način na koji je ispitni CNC postav realiziran te korake provedene u cilju realizacije tog sustava, s detaljnijim osvrtom na postupak parametriranja posmičnih prigona postava. U sklopu postupka parametriranja podešeni su parametri pojačanja regulacijskih petlji po brzini i poziciji. Parametriranje je provedeno na način da su se nadomjesni referentni signali brzine i pozicije (koje inače generira NC jezgra) generirali odgovarajućim nadomjesnim elementima odnosno signal generatorima, dok se odziv sustava istovremeno nadzirao pomoću osciloskopa. Pri tome su kao signal generator i osciloskop korišteni odgovarajući moduli dostupni u sklopu LinuxCNC sustava. Naposlijetku je ispitni postav testiran sa nekoliko NC programa.

Dobiveni rezultati ukazuju da je nakon provedenog postupka parametriranja moguće ostvariti zatvorenu regulacijsku petlju posmičnih prigona po poziciji, pri čemu greška slijeđenja svake razmatrane osi ne prelazi iznos od 0,02 mm pri posmičnim brzinama od 10 m/min. Time se može zaključiti da je dinamika realiziranih prigona zadovoljavajuća za laboratorijsku i industrijsku primjenu.

Za cjelovitu realizaciju predmetnog ispitnog CNC postava potrebno je još parametrirati glavni prigon te ostale periferne funkcije stroja (sustav za izmjenu alata i sustav za podmazivanje). Stoga će buduće aktivnosti biti usmjerene u gradnji glavnog prigona i sustava za izmjenu alata.

LITERATURA

[1] S.-H. Suh, S.-K. Kang, D.-H. Chung i I. Stroud, Theory and Design of CNC Systems, London: Springer, 2008.

[2] T. Staroveški, D. Brezak i T. Udiljak, »LinuxCNC - Napredni sustav CNC upravljanja: primjena i kritički osvrt,« *Tehnički vijesnik,* svez. 20, br. 6, pp. 1103-1110, 2013.

[3] S. Y. Liang, R. L. Hecker i R. G. Landers, »Machining Process Monitoring and Control: The State-of-the-Art,« *Journal of Manufacturing Science and Engineering,* svez. 126, pp. 297-310, 2004.

[4] »LinuxCNC,« 10 2019. [Mrežno]. Available: http://linuxcnc.org/documents/.

[5] »EtherCAT Technology Group,« 3 2020. [Mrežno]. Available: https://www.ethercat.org/download/documents/ETG_Brochure_EN.pdf.

[6] »Ethernet for Control Automation Technology,« u *Industrial Communication Technology Handbook,* Boca Raton, CRC Press, 2015, pp. 18-1 - 18-26.

[7] »EtherCAT - Slave implementation Guide,« EtherCAT Technology Group, 2018.

[8] »The Basics of CANopen,« NI, ožujak 2019. [Mrežno]. Available: https://www.ni.com/en-rs/innovations/white-papers/13/the-basics-of-canopen.html.

[9] »CiA 402 series: CANopen device profile for drives and motion control,« CiA, [Mrežno]. Available: https://www.can-cia.org/can-knowledge/canopen/cia402/.

[10] M. Mandir, »Primjena EtherCAT sabirnice u upravljačkim sustavima obradnih strojeva,« Zagreb, 2015.

[11] ESTUN, »EtherCAT User's Manual (V1.08)«.

[12] »The Linux Kernel Archives,« [Mrežno]. Available: https://www.kernel.org/.

[13] »LinuxCNC,« [Mrežno]. Available: https://www.linuxcnc.org/.

[14] »EtherLab,« IgH, [Mrežno]. Available: http://www.etherlab.org/en/ethercat/.

[15] »GitHub,« [Mrežno]. Available: https://github.com/sittner/linuxcnc-ethercat.

[16] ESTUN, »ProNet - All Digital AC Servo Systems katalog«.

[17] D. Brezak i T. Staroveški, »Regulacija obradnih strojeva,« Fakultet strojarstva i brodogradnje, Zagreb, 2019.

[18] T. Staroveški, »predavanje iz kolegija Proizvodnja podržana računalom (CAM),« Fakultet strojarstva i brodogradnje, Zagreb, 2018.

[19] »Understanding Servo Tune,« NI, veljača 2020. [Mrežno]. Available: https://www.ni.com/en-rs/support/documentation/supplemental/12/understanding-servo-tune.html.

[20] D. Androić, »Podjela računalnih mreža,« [Mrežno]. Available: http://www.phy.pmf.unizg.hr/~dandroic/nastava/ramr/poglavlje_1_4.html.

[21] »IOC Robotic's Lab,« Universitat politecnica de Catalunya, [Mrežno]. Available: https://sir.upc.edu/wikis/roblab/index.php/Development/Ethercat.

PRILOZI

I. CD-R disc