

Razvoj prigona za orijentaciju modela u zračnom tunelu

Zelić, Filip

Master's thesis / Diplomski rad

2021

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

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:115582>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-08-11**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Filip Zelić

Zagreb, 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

doc. dr. sc. Tomislav Staroveški, dipl. ing.

Student:

Filip Zelić

Zagreb, 2021.

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

Najveću zahvalu za uspješnu realizaciju rada dugujem mentoru doc.dr.sc. Tomislavu Staroveškom. Svojim komentarima, detaljnim objašnjenima te uložnim vremenom ne samo da je podigao kvalitetu konačnog rada već je izradu istog učinio sjajnim iskustvom.

Zahvaljujem se dr.sc. Mihi Klaiću na pomoći kod izrade konstrukcijskog dijela uređaja.

Posebno bih se zahvalio asistentima s Katedre za alatne strojeve Dori Bagarić i Jošku Ištuku na opuštenoj radnoj atmosferi i pomoći prilikom sastavljanja elektro-ormara.

Na kraju bih se zahvalio majci Jadranki, ocu Ivici i bratu Juri na kontinuiranoj podršci tijekom cijelog trajanja studija.

Filip Zelić



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: 602-04/21-6/1	
Ur. broj: 15-1703-21	

DIPLOMSKI ZADATAK

Student: **FILIP ZELIĆ**

Mat. br.: 0035209981

Naslov rada na hrvatskom jeziku: **Razvoj prigona za orijentaciju modela u zračnom tunelu**

Naslov rada na engleskom jeziku: **Development of wind tunnel model positioning system**

Opis zadatka:

Na Zavodu za aeronautiku Fakulteta prometnih znanosti Sveučilišta u Zagrebu u tijeku je projekt revitalizacije zračnog tunela. Jedan od ključnih elemenata predmetnog ispitnog postava je dvo-osni rotacijski prigona koji omogućuje preciznu i dinamičku orijentaciju ispitnog modela unutar mjerne sekcije tunela. U sklopu dosadašnjih aktivnosti izrađeno je idejno konstrukcijsko rješenje prigona i definirani su odgovarajući pogonski i mjerni elementi. Također je za realizaciju upravljačkog sustava odabran javno dostupan softver otvorene arhitekture LinuxCNC.

Stoga je u radu potrebno:

1. Odabrati sve potrebne elemente upravljačkog sustava prigona prema karakteristikama pogonskih motora i senzora sile. Pri tome je posebnu pažnju potrebno je posvetiti kompatibilnosti odabranih elemenata s aktualnom verzijom LinuxCNC softvera.
2. Izraditi odgovarajuće električne spojne sheme upravljačkog sustava.
3. Parametrirati LinuxCNC softver i sve elemente upravljačkog sustava prema karakteristikama prigona.
4. Testirati gibanje prigona.
5. Dati zaključke rada.

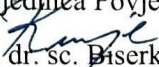
U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
30. rujna 2021.

Rok predaje rada:
2. prosinca 2021.

Predvideni datum obrane:
13. prosinca do 17. prosinca 2021.

Zadatak zadao: 
doc. dr. sc. Tomislav Staroveški

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	IV
POPIS TABLICA.....	VII
POPIS KRATICA	VIII
POPIS OZNAKA	X
SAŽETAK.....	XI
SUMMARY	XII
1. UVOD.....	1
1.1. Računalno numeričko upravljanje	2
1.2. Upravljački sustavi otvorene arhitekture	3
2. UPRAVLJAČKI SUSTAV LINUXCNC.....	6
2.1. Povijesni razvoj LinuxCNC-a.....	6
2.2. Arhitektura sustava LinuxCNC.....	7
2.2.1. EMCMOT	9
2.2.2. EMCTASK	10
2.2.3. EMCIO.....	10
2.2.4. GUI	11
2.3. Hardverski apstrakcijski sloj.....	13
3. PARAMETRIRANJE REGULATORA POSMIČNIH OSI U UPRAVLJAČKOM SUSTAVU LINUXCNC	17
3.1. Pregled teorije automatske regulacije	17
3.1.1. Kaskadna regulacija i PID regulator	17
3.1.2. Analiza sustava u vremenskoj domeni.....	19
3.1.3. Analiza sustava u frekvencijskoj domeni	20
3.2. Ispitivanje latencije	21
3.3. Parametriranje regulatora koračnih motora.....	21
3.3.1. Stepper Configuration Wizard	22
3.4. Parametriranje regulatora servomotora	24

3.4.1.	Regulacijska petlja momenta (struje).....	24
3.4.2.	Regulacijska petlja brzine	25
3.4.3.	Regulacijska petlja pozicije	26
4.	ETHERCAT SABIRNICA.....	27
4.1.	OSI model	27
4.2.	EtherCAT	29
4.2.1.	Fizički sloj.....	32
4.2.2.	Podatkovni sloj.....	33
4.2.3.	Aplikacijski sloj	39
4.2.4.	Distribuirani satovi.....	43
4.2.5.	Adresiranje	44
4.2.6.	Automat stanja podređenog uređaja.....	46
5.	KONSTRUKCIJSKO RJEŠENJE, POGONSKI I MJERNI ELEMENTI PRIGONA.....	47
5.1.	Konstruktivsko rješenje uređaja	47
5.2.	Pogonski elementi prigona.....	50
5.3.	Mjerni element prigona.....	53
6.	ODABIR ELEMENATA UPRAVLJAČKOG SUSTAVA I IZRADA ELEKTRO- ORMARA.....	54
6.1.	Odabir elemenata upravljačkog sustava.....	54
6.1.1.	Napajanje sustava.....	54
6.1.2.	Grebenasta sklopka	57
6.1.3.	Automatski osigurači	58
6.1.4.	Motorna zaštitna sklopka	59
6.1.5.	Sigurnosni logički krug.....	60
6.1.6.	Servo regulatori.....	64
6.1.7.	EtherCAT I/O modul	69
6.1.8.	Upravljačko računalo	70
6.1.9.	Mrežna topologija	70
6.2.	Izrada elektro-ormara	72
6.2.1.	Određivanje prostornog rasporeda elemenata.....	72
6.2.2.	Priprema konektora za digitalne I/O servo regulatora	73
6.2.3.	Ožičenje elemenata sustava	75

7. PARAMETRIRANJE I TESTIRANJE SUSTAVA.....	77
7.1. Početno konfiguriranje sustava	77
7.1.1. INI datoteka	77
7.1.2. Konfiguracijska datoteka za EtherCAT	78
7.1.3. HAL datoteka.....	80
7.2. Podešavanje regulacijskih petlji i testiranje sustava	81
8. ZAKLJUČAK.....	86

POPIS SLIKA

Slika 1-1. Sastavni dijelovi CNC sustava [1].....	2
Slika 2-1. Ciklusi izvršavanja zadataka različitih prioriteta	7
Slika 2-2. Arhitektura LinuxCNC-a [4]	8
Slika 2-3. Gmoccapy grafičko korisničko sučelje	11
Slika 2-4. Virtualni kontrolni paneli	12
Slika 2-5. XML opisna datoteka	12
Slika 2-6. Primjer konfiguracije hardverskog apstrakcijskog sloja [4].....	14
Slika 2-7. Povezivanje komponenata unutar HAL prostora	16
Slika 3-1. Kaskadna regulacijska struktura.....	18
Slika 3-2. Karakteristične veličine za opis odziva sustava na jediničnu odskočnu funkciju [7]	19
Slika 3-3. Analiza sustava u frekvencijskom području.....	20
Slika 3-4. Histogram latencije – lijevo petlja 25 μ s, desno petlja 1000 μ s [3].....	21
Slika 3-5. Shema upravljanja koračnim motorima	21
Slika 3-6. Koraci za konfiguriranje stroja s koračnim motorima.....	23
Slika 3-7. Shema podešavanja regulacijske petlje po brzini.....	25
Slika 3-8. Zatvaranje regulacijske petlje po poziciji u HAL prostoru	26
Slika 4-1. OSI model.....	28
Slika 4-2. Tržišni udjeli pojedinih industrijskih mreža u 2021. godini [11].....	31
Slika 4-3. Mrežna kartica (lijevo), Cat5 kabel s RJ45 konektorom (sredina) i ESC čip (desno).....	32
Slika 4-4. Princip rada EtherCAT sabirnice, prilagođeno iz [12].....	33
Slika 4-5. Struktura EtherCAT telegrama.....	34
Slika 4-6. Adresna mjesta EEPROM (SII) memorije, prilagođeno iz [14]	37
Slika 4-7. Sinkronizirani prijenos podataka preko DPRAM memorije [15]	38
Slika 4-8. Arhitektura podređenog uređaja [10]	38
Slika 4-9. Aplikacijski protokoli EtherCAT sabirnice [10]	39
Slika 4-10. Primjer PDO mapiranja [19]	42
Slika 4-11. Princip rada distribuiranih satova, prilagođeno iz [15]	43
Slika 4-12. Logičko adresiranje [15]	45
Slika 5-1. Konstrukcija uređaja.....	47
Slika 5-2. Prigon, model letjelice i zračni tunel.....	48

Slika 5-3. Nosiva konstrukcija prigona.....	49
Slika 5-4. Servomotor prve osi s pripadajućom karakteristikom [22].....	50
Slika 5-5. Servomotor druge osi i pripadajuća karakteristika [22]	51
Slika 5-6. Reduktor GSN200-10K-SV [23].....	52
Slika 5-7. Reduktor GSN130-10K-SV [23].....	52
Slika 5-8. Mjerna ćelija ATI Mini27 Titanium [24]	53
Slika 5-9. Način spajanja mjerne ćelije, prilagođeno iz [24].....	53
Slika 6-1. Ispravljači MEAN WELL EDR-120-24 i NDR-240-48 [25].....	55
Slika 6-2. Električna shema izmjeničnog napajanja	56
Slika 6-3. Grebenasta sklopka (Q1) [26]	57
Slika 6-4. Automatski prekidač iC60N 2P-4A-C [26].....	58
Slika 6-5. Motorna zaštitna sklopka TeSys GV2 [26]	59
Slika 6-6. Sigurnosni relej (K1) [27] i kontaktor (K2 i K3) [26].....	61
Slika 6-7. Električna shema spajanja sigurnosnog releja.....	62
Slika 6-8. Tipkalo za pokretanje (S4) i tipkalo za zaustavljanje u nuždi (S1 i S2) i njihovi kontakti [26]	63
Slika 6-9. Servo regulator ProNet-04AEG-EC (lijevo) i ProNet-08AEG-EC (desno) [22]..	64
Slika 6-10. Digitalni ulazi i izlazi servo regulatora	67
Slika 6-11. Relej za paljenje kočnice s prihvatom za DIN šinu (K5 i K8) [28]	67
Slika 6-12. Princip upravljanja kočnicom motora iz servo regulatora.....	68
Slika 6-13. EtherCAT modul s I/O proširenjima [29]	69
Slika 6-14. Industrijsko računalo XCY X30 J1900 [30].....	70
Slika 6-15. Mrežna topologija.....	71
Slika 6-16. Razmještaj elemenata na temeljnoj ploči elektro-ormara	72
Slika 6-17. SCSI 20-pinski muški konektor i oklopljeni kabel s vodičima	73
Slika 6-18. Raspored pinova na konektoru	74
Slika 6-19. Provođenje napojnih vodova	75
Slika 6-20. Ožičenje logičkih dijelova sustava i spajanje priključaka servo regulatora	76
Slika 7-1. Testiranje posmičnih prigona	81
Slika 7-2. Prva os - odziv na referencu brzine prije podešavanja	82
Slika 7-3. Prva os – odziv na referencu brzine nakon podešavanja.....	83
Slika 7-4. Prva os – odziv na referencu pozicije.....	83
Slika 7-5. Druga os – odziv na referencu brzine prije podešavanja.....	84
Slika 7-6. Druga os – odziv na referencu brzine nakon podešavanja	84

Slika 7-7. Druga os – odziv na referencu pozicije prije podešavanja	85
Slika 7-8. Druga os – odziv na referencu pozicije nakon podešavanja	85

POPIS TABLICA

Tablica 1-1. Prednosti otvorenih CNC sustava	4
Tablica 1-2. Pregled CNC sustava otvorene arhitekture	5
Tablica 2-1. HAL komponente.....	16
Tablica 3-1. Utjecaj osnovnih dinamičkih članova na karakteristike odziva sustava.....	18
Tablica 4-1. Značenje oznaka sa slike 4-5.	35
Tablica 4-2. Osnovna stanja podređenih uređaja	46
Tablica 5-1. Tehničke specifikacije odabranih elektromotora	51
Tablica 6-1. Dijelovi sustava koji se napajaju izmjeničnom strujom	54
Tablica 6-2. Izvori istosmjernog napajanja	56
Tablica 6-3. Opis terminala sigurnosnog releja	61
Tablica 6-4. Tehničke specifikacije odabranih servo regulatora.....	64
Tablica 6-5. Priključci odabranih ESTUN ProNet servo regulatora.....	65
Tablica 6-6. Digitalni ulazi i izlazi odabranih servo regulatora.....	66
Tablica 6-7. Uparivanje vodiča i 20-pinskog konektora SCSI.....	73

POPIS KRATICA

AC	Alternating Current
ADS	Automation Device Specification
AoE	ADS over EtherCAT
API	Application Programming Interface
AT	Acknowledge Telegram
CAD	Computer-aided Design
CNC	Computer Numerical Control
CoE	CANOpen over EtherCAT
CPU	Central Processing Unit
DC	Direct Current
DPRAM	Dual-Port Random Access Memory
EAP	EtherCAT Automation Protocol
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMC	Enhanced Machine Controller
EoE	Ethernet over EtherCAT
EPU	EtherCAT Processing Unit
ESC	EtherCAT Slave Controller
EtherCAT	Ethernet for Control Automation Technology
FIFO	First In First Out
FMMU	Fieldbus Memory Management Unit
FoE	File Access over EtherCAT
FSoE	Failsafe over EtherCAT
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
IDN	Identification Number
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
IP	Internet Protocol
LED	Light Emitting Diode
MAC	Media Access Control
MDT	Master Data Telegram

MMI	Man-Machine Interface
NCK	Numerical Control Kernel
NML	Neutral Messaginig Language
OAC	Open Architecture Controller
OSI	Open Systems Interconnection
PDI	Process Data Interface
PDO	Process Data Object
PDU	Protocol Data Unit
PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller
PMAC	Programmable Multi-Axis Controller
PoE	Power over Ethernet
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse Width Modulation
RT	Real-time
SDO	Service Data Object
SERCOS	Serial Real-time Communication System
SII	Slave Information Interface
SoE	SERCOS over EtherCAT
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
VCP	Virtual Control Panel

POPIS OZNAKA

Oznaka	Jedinica	Opis
e_0	%	trajno regulacijsko odstupanje
i_a	A	jakost struje armature
K_m	N·m/A	koeficijent momenta
M_p	%	postotni iznos najvećeg prebačaja
T	N·m	moment motora
t_p	s	vrijeme najvećeg prebačaja
t_r	s	vrijeme porasta
t_s	s	vrijeme smirivanja
X_i	-	amplituda izlaznog signala
X_u	-	amplituda ulaznog signala
φ	rad	fazni pomak
ω	rad/s	kružna frekvencija

SAŽETAK

U sklopu diplomskog rada, proveden je projekt revitalizacije postojećeg zračnog tunela. Konkretnije, naglasak je stavljen na razvoj upravljačkog dijela sustava dvo-osnog rotacijskog posmičnog prigona koji će se koristiti za precizno pozicioniranje modela letjelice u mjernoj sekciji zračnog tunela. Središnji dio upravljačkog sustava je softver otvorene arhitekture LinuxCNC. Komunikacija između NC računala i ostalih dijelova sustava ostvarena je korištenjem EtherCAT sabirnice. EtherCAT tehnologija je, zajedno s LinuxCNC-om, detaljno opisana u teorijskom dijelu rada. U nastavku je proveden proces odabira elemenata upravljačkog elektro-ormara, te su nacrtane pripadajuće električne sheme. Nakon uspješne integracije svih elemenata upravljačkog sustava, obavljeno je početno parametriranje LinuxCNC softvera i servo regulatora. Konačno, posmični prigoni su testirani te je provedeno podešavanje parametara regulacijskih petlji.

Ključne riječi: rotacijski posmični prigon, LinuxCNC, EtherCAT, upravljački elektro-ormar, servo motor, upravljanje gibanjem, CNC

SUMMARY

In this thesis, a project of revitalization of existing wind tunnel was undertaken. More specifically, the focus was put on the development of the control system for two-axis rotary feed drive which will be used for precise positioning of the aircraft model inside the test section of the wind tunnel. The central part of the designed control system is open architecture machine controller, LinuxCNC. Communication between NC computer and different parts of the system was realized using EtherCAT fieldbus system. EtherCAT technology is, along with LinuxCNC, described in detail in the theoretical part of the thesis. Furthermore, a process of selecting control cabinet elements was conducted and corresponding electrical schemes were drawn. After successful integration of all the control system elements, initial commissioning of LinuxCNC software and servo drives was carried out. Finally, feed drives were tested and tuning of control loops was performed.

Key words: rotary feed drive, LinuxCNC, EtherCAT, control cabinet, servo motor, motion control, CNC

1. UVOD

Iako su prve asocijacije pri spomenu računalnog numeričkog upravljanja (u daljnjem tekstu CNC upravljanja) obradni procesi kao što su glodanje ili tokarenje, ova tehnologija ima puno šire područje primjene. Od ostalih primjena u proizvodnom poduzeću, mogu se spomenuti još i 3D printeri, laserske i plazma rezačice, strojevi za bušenje i brušenje te industrijski roboti. No, uporaba ove tehnologije nije ograničena samo na proizvodna postrojenja već se može primijeniti kad god je potrebna precizna kontrola gibanja prigona nekog stroja ili uređaja.

Gledano iz te perspektive, CNC upravljanje prepoznato je kao idealan kandidat za realizaciju uređaja koji je glavni predmet ovog diplomskog rada. Uređaj se razvija u sklopu revitalizacije postojećeg zračnog tunela, a glavni zadatak mu je omogućiti precizno pozicioniranje modela letjelice unutar mjerne sekcije tunela. Predviđeno je podešavanje dvaju kutova rotacije (skretanja i propinjanja) modela letjelice, čijom će se pozicijom upravljati pomoću dva rotacijska prigona sa servomotorima. Kad se u obzir uzme i postojeći ventilator s varijabilnim frekvencijskim pretvaračem, uređaj se može zamisliti kao CNC stroj s dvije posmične rotacijske osi i s ventilatorom kao glavnim prigonom. Sekundarni zadatak uređaja je prihvati samog modela letjelice i mjerne ćelije koja će se koristiti za mjerenje aerodinamičkih sila i momenata koji djeluju na model.

Prvi dio rada teorijske je prirode, a daje uvod u problematiku upravljanja gibanjem i opisuje strukturu programa korištenih kroz praktični dio rada. U njemu je detaljno predstavljen upravljački sustav otvorene arhitekture LinuxCNC, koji je središnji dio cijelog sustava. Također je opisana i komunikacijska sabirnica EtherCAT koja je sučelje između LinuxCNC-a i elemenata sustava s kojima upravljački sustav izmjenjuje informacije.

U sklopu dosadašnjeg razvoja izrađeno je idejno konstrukcijsko rješenje prigona te su odabrani odgovarajući servomotori s pripadajućim reduktorima i regulatorima. U ovom radu nastavljen je proces izrade cjelovitog rješenja, a sljedeći koraci koje ono podrazumijeva jesu odabir svih elemenata upravljačkog elektro-ormara i izrada električnih shema prema kojima će se elementi međusobno povezati.

U završnom dijelu rada bit će provedeno konfiguriranje upravljačkog sustava LinuxCNC i parametriranje upravljačkih elemenata prigona. Također će biti provedeno testiranje gibanja rotacijskih osi prigona kako bi se utvrdilo jesu li karakteristike odziva na zadanu referencu brzine i pozicije zadovoljavajuće.

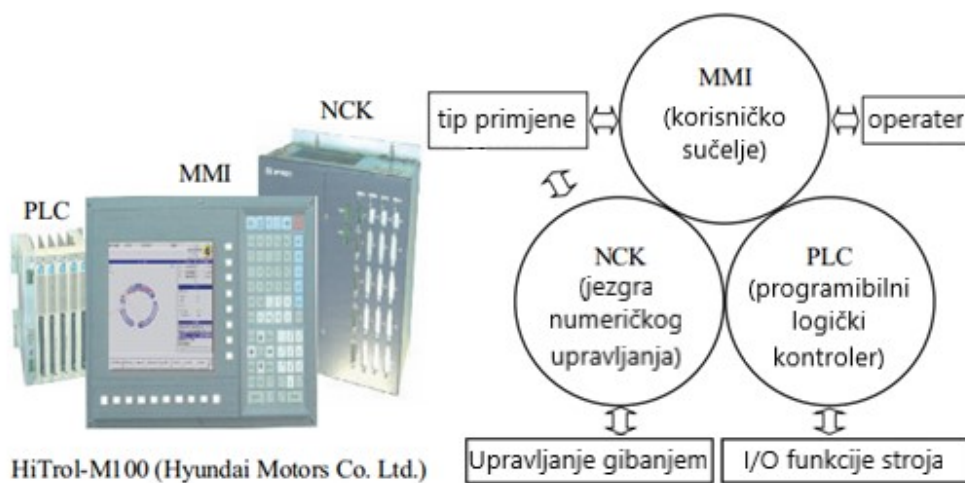
1.1. Računalno numeričko upravljanje

Računalno numeričko upravljanje (CNC) vrsta je upravljačkog sustava kod kojeg se za čitanje i izvršavanje programskog koda (uglavnom G-koda (RS-274)) koristi računalo. Sastoji se od nekoliko odvojenih hardverskih i softverskih modula (slika 1-1.).

Jezgra numeričkog upravljanja (NCK) središte je upravljačkog sustava, a neke od glavnih uloga su joj interpretacija programskog koda, interpolacija putanje gibanja posmičnih osi, upravljanje akceleracijskim/deceleracijskim karakteristikama pojedine osi, te regulacija pozicije na temelju razlike (greške) između željenog položaja (reference pozicije) osi i položaja osi izmjerene mjernom opremom (npr. inkrementalnim enkoderom) [1]. Za kontrolu položaja osi najčešće se koristi PID regulator koji će detaljnije biti objašnjen su poglavljima koja slijede.

Operatorski panel odnosno sučelje između čovjeka i stroja (MMI) predstavlja drugi važan modul u sklopu CNC upravljačkog sustava, a operateru omogućava uređivanje, pisanje i učitavanje programa, praćenje statusnih informacija stroja i stanja obradnog procesa, te početno referenciranje stroja i pokretanje stroja u različitim načinima rada.

Svim perifernim funkcijama stroja – onima koje se ne odnose na upravljanje gibanjem osi (kao što su provjera sigurnosnih krugova, signala s graničnih prekidača ili kontrola sustavom izmjene alata), upravlja logička jedinica upravljačkog sustava, to jest programibilni logički kontroler (PLC). Program PLC modula izvršava se ciklički i to na način da PLC čita ulazne varijable, izvršava logičke operacije nad njima i ispisuje rezultate tih operacija na izlazu. Glavni dijelovi PLC-a su ulazno i izlazno sučelje, procesorska jedinica (CPU) za izvršavanje programa, memorijska jedinica za učitavanje programa i operacijskog sustava, sustav napajanja i komunikacijsko sučelje za uspostavljanje veze s NCK i/ili drugim PLC-ovima.



Slika 1-1. Sastavni dijelovi CNC sustava [1]

1.2. Upravljački sustavi otvorene arhitekture

Upravljački sustavi otvorene arhitekture (OAC) omogućavaju prilagođavanje i nadograđivanje funkcionalnosti stroja i procesa, a razvijeni su kako bi se prevladali nedostaci klasičnih (zatvorenih) CNC sustava. Neki od glavnih nedostataka zatvorenih upravljačkih sustava su:

- Softver se razvija za specifičan hardver što smanjuje fleksibilnost i agilnost proizvodnog sustava
- Nije moguće implementirati dodatne varijable u upravljačku jedinicu stroja (npr. snimljene signale obradnog procesa)
- Integracija korisničkih hardverskih modula nije moguća (npr. senzora vibracija)
- Upravljačke varijable nisu dostupne (npr. signal reference jakosti struje iz upravljačkih sklopova motora posmičnih osi)
- Korisnička sučelja razlikuju se od proizvođača do proizvođača

Prema IEEE 1003.0 standardu otvoreni upravljački sustav definiran je kao sustav koji omogućava pravilno implementiranim aplikacijama pokretanje na različitim platformama različitih proizvođača, interoperabilnost s ostalim aplikacijama sustava i kao onaj koji predstavlja konzistentan način interakcije sa korisnikom [2].

Otvorenost upravljačkog sustava može se procijeniti korištenjem sljedećih kriterija [2]:

- **Prenosivost** → aplikacijski moduli mogu se koristiti na različitim platformama uz zadržavanje potpune funkcionalnosti
- **Modularnost** → različiti broj aplikacijskih modula se može izvršavati na platformi bez međusobne interferencije
- **Interoperabilnost** → aplikacijski moduli mogu raditi zajedno i izmjenjivati podatke na unaprijed određen način
- **Mogućnost proširenja** → moguće je prilagoditi funkcionalnost aplikacijskih modula, te performanse i veličinu hardvera u ovisnosti o potrebama korisnika

Prema prethodno spomenutom IEEE standardu i navedenim kriterijima otvorenosti, otvoreni upravljački sustav morao bi biti neovisan o proizvođaču softvera (kako bi se izbjegli partikularni interesi), vođen dogovorom između proizvođača i korisnika, temeljen na standardima i slobodno dostupan (besplatan).

U tablici 1-1. (prilagođeno iz [2]) navedene su prednosti primjene OAC i to za sve sudionike koji sudjeluju u izradi i primjeni numerički upravljanih strojeva. Zajednička prednost im je smanjenje troškova upravljačkog sustava iz razloga što je OAC moguće koristiti na različitim konfiguracijama hardvera, te ga je moguće nadograđivati s vremenom i prilagoditi specifičnim zahtjevima krajnjeg korisnika stroja što dovodi do veće agilnosti i fleksibilnosti proizvodnog procesa. Također, odvojenim razvojem softverskih i hardverskih komponenti, strojevima (računalnim jedinicama stroja) je omogućeno kontinuirano praćenje razvoja poluvodičke industrije i informacijskih tehnologija na način da se jednostavnom zamjenom elektroničkog hardvera postignu bolje performanse (zastarjelog) stroja.

Tablica 1-1. Prednosti otvorenih CNC sustava

Proizvođači softvera, akademska zajednica	Proizvođači (alatnih) strojeva	Korisnik
Višekratna upotreba softvera	Integracija korisnički specifičnog softvera neovisno o proizvođaču softvera	
	Standardizirana korisnička sučelja	
Jednostavna implementacija funkcija prilagođenih korisniku		
Hardver se može nadograđivati		
Nezavisan razvoj upravljačkog sustava i alatnog stroja		
Smanjeni troškovi upravljačkog sustava		

Tijekom godina nekoliko je istraživačkih grupa nastojalo razviti upravljački sustav otvorene arhitekture koji bi sadržavao sve prethodno navedene karakteristike. Neke od najznačajnijih inicijativa na ovom području bile su sljedeće [2]: OSACA (Europa, 1992.), OSEC (Japan, 1994.), OMAC (USA, 1994.) i JOP (Japan, 1996.). U istom radu spomenuta su i određena rješenja proistekla iz akademske zajednice, te još nekoliko upravljačkih sustava koji su komercijalizirani (Delta Tau PMAC-NC, IBH PA 8000, Galil DMC 1000).

Najveći izazovi za širu primjenu otvorenih upravljačkih sustava su nedostatak standardizacije, podrške i međusobna nekompatibilnost trenutno razvijenih OAC-ova. To može dovesti do paradoksalne situacije da se iz početnog nastojanja da se prevladaju razlike između upravljačkih sustava različitih proizvođača dođe do situacije da na tržištu postoji velik broj međusobno nekompatibilnih OAC-ova. Iz tog razloga su otvoreni CNC upravljački sustavi još uvijek primarno korišteni za edukativne, istraživačke i hobističke svrhe. U tablici 1-2. naveden je kratki pregled nekoliko dostupnih OAC-ova koji se prema prethodno navedenim kriterijima razlikuju u stupnju otvorenosti, a često se koriste u prethodno navedene svrhe. U daljnjem radu će detaljno biti opisan sustav LinuxCNC.

Tablica 1-2. Pregled CNC sustava otvorene arhitekture

	LinuxCNC	Mach3	Mach4*	PlanetCNC**
OS	Linux	Windows	Windows	Linux Windows MacOS RaspberryPi
Cijena	besplatno	175 USD	200 USD	besplatno
Glavne značajke	- softver otvorenog koda, zrelost koda	- pretvara standardno PC računalo u 6-osni CNC kontroler	- opcija upravljanja na daljinu	- automatizirano mjerenje ispona alata i referenciranje
	- mogućnost izvršavanja u stvarnom vremenu	- prilagodba M-kodova i makronaredbi preko VBScripta	- modularna arhitektura, velika mogućnost kastomizacije	- potpuno podesiva procedura izmjene alata
	- podrška za puno različitih konfiguracija CNC strojeva	- izravno uvažanje DXF, BMP, JPG i HPGL datoteka kroz LazyCam	- proračun kontrole gibanja izvršava se u NC jezgri	- podržava različite načine rada (simulacija, mjerenje)
	- mogućnost izrade vlastitog grafičkog sučelja	- vizualni prikaz naredbi G-koda	- brza i jednostavna prilagodba GUI-a	- USB/Ethernet veza s računalom
	- aktivna zajednica korisnika, jednostavna instalacija	- prilagodljivo grafičko korisničko sučelje (GUI)	- jednostavno povećavanje i smanjivanje funkcionalnosti	- konverzija DXF, PLT/HPGL, NC Drill i Gerber datoteka te slika u G-kod

OS – operacijski sustav

*Postoji hobistička i industrijska verzija Mach4, cijena potonje je 1400 USD

**Besplatno uz upravljačke sklopove s valjanom licencom

2. UPRAVLJAČKI SUSTAV LINUXCNC

LinuxCNC fleksibilan je i modularan upravljački sustav otvorene arhitekture, a sposoban je koordinirano upravljati gibanjem osi CNC strojeva kao što su 3D printeri, laserski rezači, tokarilice, glodalice i roboti. Izvršava se u operacijskom sustavu s Linux jezgrom i ima mogućnost rada u stvarnom vremenu (uz dodavanje odgovarajućih ekstenzija). Zbog modularne gradnje, vrlo jednostavno ga je prilagoditi različitim konfiguracijama opreme što ga čini idealnim za istraživačke i edukacijske svrhe. U nastavku poglavlja naveden je kratki povijesni razvoj, opisana je njegova arhitektura te je objašnjen koncept hardverskog apstrakcijskog sloja.

2.1. Povijesni razvoj LinuxCNC-a

LinuxCNC započeo je svoj razvoj pod imenom *Enhanced Machine Controller* (EMC) kao projekt američkog Instituta za standarde i tehnologiju (NIST). Početna ideja bila je razviti programski paket za upravljanje gibanjem kao platformu za ispitivanje koncepata i standarda. Uz podršku General Motorsa razvijena je verzija tog sustava koja se koristila za upravljanje velikim tokarskim strojem, a izvršavala se na operacijskom sustavu Windows NT [3].

Softverski kod je nakon završetka projekta, zajedno s pripadajućim izvještajem, bio javno objavljen na mrežnim stranicama gdje je privukao veliko zanimanje određenog dijela javnosti. Nastavak razvoja EMC softvera bio je viđen u njegovoj prilagodbi za upravljanje jeftinijim hardverom koji bi se koristio kao nadogradnja ili zamjena zastarjelih ili pokvarenih CNC sustava. Dogovoreno je kako će novorazvijeni softver ostati u javnoj domeni te kako će se komercijalni sustav Windows NT, zamijeniti Linux sustavom s ekstenzijom za rad u stvarnom vremenu. Postojećem korisničkom sučelju i RS274 interpreteru naknadno su dodani još i softver za planiranje trajektorije gibanja te PID regulacijska petlja [3].

Zajednica korisnika ovog sustava sve je brže počela rasti nakon objedinjavanja svih potrebnih instalacijskih datoteka (Linux-a, programskih zakrpa i EMC-a) na jednom instalacijskom CD-u. Godine 2003. programski kod je doživio značajne promjene – neki dijelovi su ponovno napisani, drugi reorganizirani i pojednostavljeni te je preimenovan u EMC2. Najvažnija promjena u novoj verziji bila je dodavanje hardverskog apstrakcijskog sloja (engl. *Hardware Abstraction Layer*). Konačno ime LinuxCNC dobiva 2011. godine i to zato što je iz pravnih razloga zabranjeno korištenje naziva EMC i EMC2 [3].

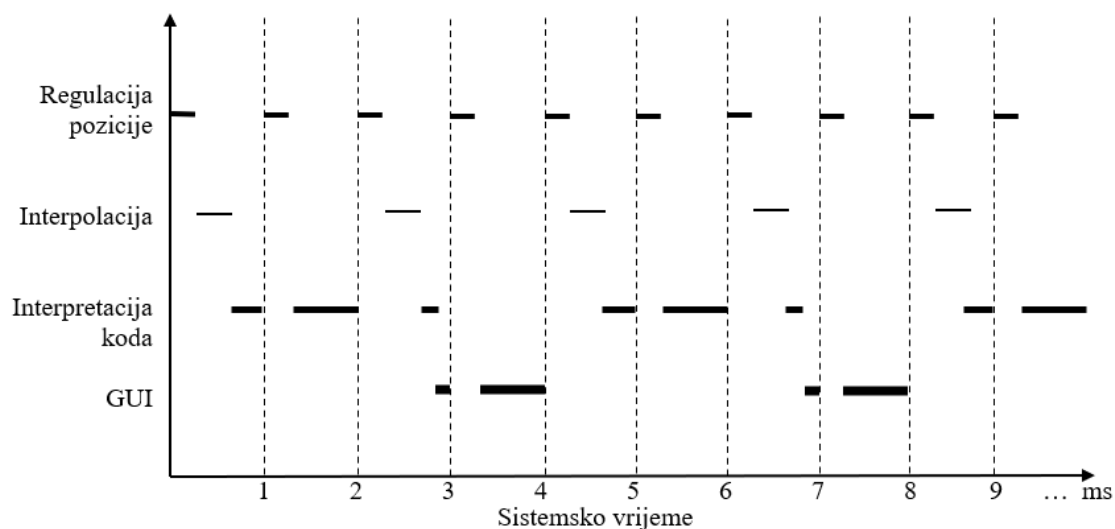
2.2. Arhitektura sustava LinuxCNC

Struktura LinuxCNC sustava hijerarhijskog je tipa, a podijeljena je na četiri osnovna modula:

- GUI – grafička i tekstualna korisnička sučelja
- EMCTASK – koordinacijski modul
- EMCMOT – modul za upravljanje gibanjem
- EMCIO – diskretni logični kontroler

Od navedenih modula jedino se modul za upravljanje gibanjem (EMCMOT) izvršava u stvarnom vremenu, a komunikacija između njega i EMCTASK modula odvija se preko spremnika dijeljene memorije ili RT-Linux FIFO (*First In First Out*) memorijskih međuspremnik. Komunikacija između modula koji se ne izvršavaju u stvarnom vremenu (između modula GUI i EMCTASK te EMCTASK i EMCIO) odvija se preko *Neutral Messaginig Language* (NML) protokola [4].

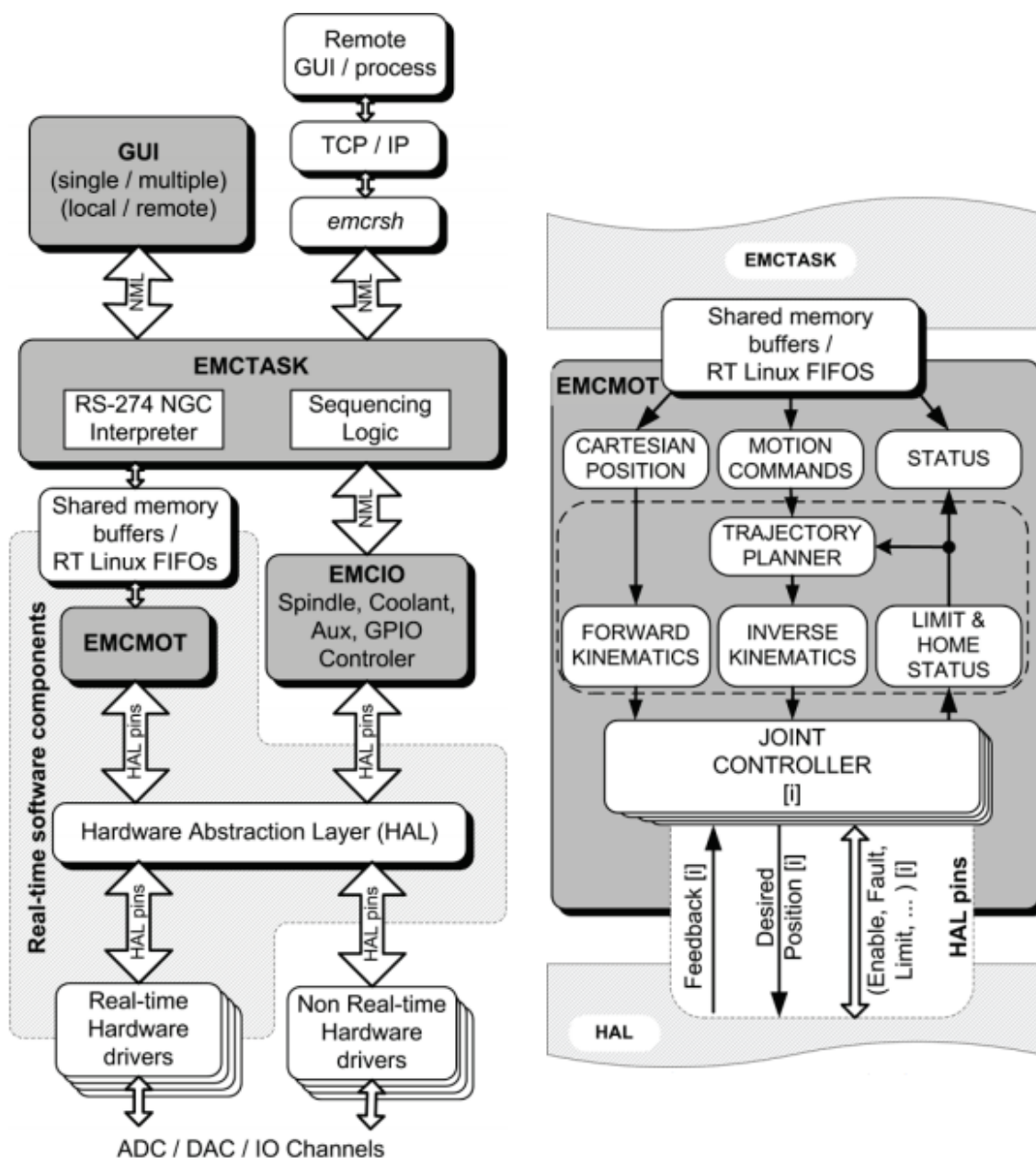
Pod stvarnim vremenom podrazumijeva se cikličko izvršavanje pojedinih funkcija unutar upravljačkog sustava (petlji u HAL prostoru) u diskretnim vremenskim intervalima (tipično jedna milisekunda). Unutar CNC sustava postoje zadaci koji imaju viši prioritet (oni koji su povezani s upravljanjem gibanjem) i oni s nižim prioritetom (npr. GUI). Slika 2-1. (prilagođeno iz [1]) prikazuje način raspodjele procesorskih resursa računala za izvršavanje različitih zadataka sustava u odnosu na njihov prioritet. Regulacija pozicije predstavlja najviši prioritet, a funkcije nižeg prioriteta (one koje ne zahtijevaju stvarno vrijeme) izvršavaju se prema važnosti u preostalom vremenu (slika 2-1.).



Slika 2-1. Ciklusi izvršavanja zadataka različitih prioriteta

Izvršavanje LinuxCNC sustava u stvarnom vremenu moguće je osigurati na dva načina. U prvom bi se, za komponente sustava koje zahtijevaju stvarno vrijeme, izdvojile zasebne procesorske jezgre. Alternativno se može uz brzu petlju dodati sporija petlja (s većim periodom) u kojoj bi se izvršavalo računanje izlaza komponenata koje su vremenski manje zahtjevne.

Slika 2-2. prikazuje hijerarhijsku strukturu LinuxCNC upravljačkog sustava na kojoj se mogu vidjeti moduli od kojih se sustav sastoji i način na koji su međusobno povezani. Pojedinačni moduli (uključujući i EMC MOT modul prikazan na desnoj strani slike) zajedno s hardverskim apstrakcijskim slojem (HAL) bit će detaljno opisani u nastavku.



Slika 2-2. Arhitektura LinuxCNC-a [4]

2.2.1. EMCMOT

Modul za upravljanje gibanjem jedini je koji se izvršava u stvarnom vremenu, a napisan je u programskom jeziku C. Glavne uloge su mu planiranje trajektorije, proračun direktne i inverzne kinematike te računanje i slanje izlaza (reference pozicije) prema regulatorima pozicija posmičnih osi. Ovaj postupak sastoji se od uzorkovanja položaja upravljanih osi koji dolaze s enkodera, izračuna sljedeće točke trajektorije gibanja, interpolacije između točaka trajektorije i slanja rezultata interpolacije (reference pozicije) do regulatora pozicije [5]. Unutarnja struktura EMCMOT modula prikazana je na desnoj strani slike 2-2.

Konfiguracijska datoteka (INI) koristi se za početno postavljanje parametara, a sastoji se od više različitih sekcija kao što su broj i tip (linearne ili rotacijske) posmičnih osi, pojačanja pojedinih regulacijskih članova, odabir mjernih jedinica, postavljanje odnosa između rotacijskog gibanja zgloba (motora) i linearnog pomaka osi (npr. za koliko se linearno pomakne os za jednu punu rotaciju motora), podešavanje trajanja ciklusa kod planiranja trajektorije i ostalih parametara sustava.

Servomotorima se upravlja u zatvorenoj petlji uz upotrebu PID regulatora s unaprijednim petljama s pojačanjima nultog, prvog i drugog reda. PID kompenzacijski algoritam izveden je kao zaseban modul u HAL prostoru, a uobičajeno je da njihov broj (PID regulatora) odgovara broju upravljanih osi. Koračnim motorima upravlja se u otvorenoj petlji signalima koraka i smjera. Signali se prema regulatorima koračnih motora šalju iz HAL komponente *stepgen* odgovorne za generiranje impulsa koraka [5].

Aplikacijsko programsko sučelje (engl. *Application Programming Interface*, API) je softver napisan u C programskom jeziku koji osigurava jednostavnu integraciju pogonskih programa (engl. *device driver*) hardverskih komponenti u sustav LinuxCNC u obliku HAL modula i njihovo povezivanje HAL signalima s EMCMOT modulom [4].

Komunikacija između EMCMOT modula i njemu podređenih modula, kao što su prethodno spomenuti PID regulator ili pogonski programi hardverskih komponenata, odvija se korištenjem HAL signala. Statusne, operativne i informacije o greškama šalju se preko spremnika dijeljene memorije ili FIFO spremnika, a naredbe se na isti način primaju iz EMCTASK-a. Kontroler za upravljanje gibanjem za komunikaciju ne koristi NML poruke jer se one koriste samo za komunikaciju između softverskih modula napisanih u C++ programskom jeziku (EMCMOT je napisan u C jeziku) [4].

2.2.2. EMCTASK

EMCTASK je koordinirajući modul i interpreter programskog koda. U hijerarhijskoj strukturi LinuxCNC-a (slika 2-2.) nalazi se ispod GUI, a iznad EMCOT i EMCTASK modula. Naredbe u obliku RS-274 NGC (G-koda) ulaze u EMCTASK, a šalje ih operater kroz GUI ili dolaze od nekog drugog procesa. EMCTASK zatim analizira pristigle naredbe uzimajući u obzir trenutni status EMCOT i EMCIO modula, prevodi ih u NML poruke i šalje ih dalje do podređenih modula (EMCOT i EMCIO) ili samome sebi u odgovarajuće vrijeme (poštujući slijednu logiku) [4].

2.2.3. EMCIO

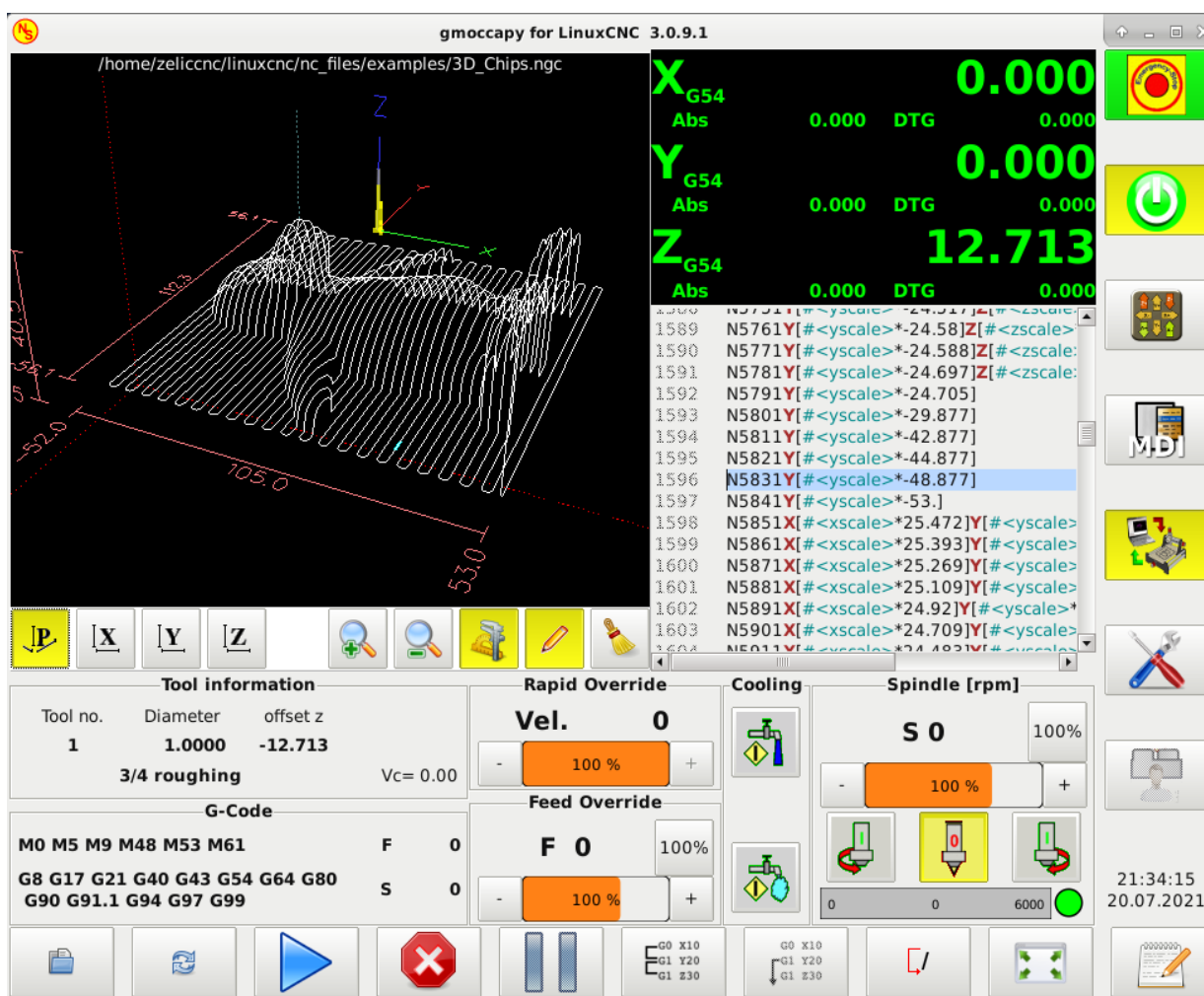
Diskretni logički kontroler upravlja svim logičkim funkcijama (kodirane kao M funkcije u G kodu) koje ne sudjeluju direktno u kontroli gibanja posmičnih osi (primjerice upravljanje magazinom alata, upravljanje brzinom vrtnje glavnog vretena, uključivanje/isključivanje sredstva za hlađenje i podmazivanje, itd.) i ostalim pomoćnim funkcijama (npr. tipkalo za zaustavljanje u nuždi).

EMCIO se konfigurira tako da se za pojedine funkcije zasebno napišu vlastite konfiguracijske (HAL) datoteke u kojima je opisano ulazno-izlazno sučelje za pojedinu primjenu ili određeni podsustav. Konfiguracijske datoteke sastoje se od različitih HAL modula i signala koji ih povezuju (više o HAL-u u poglavlju 2.3.). Zbog preglednosti se može više HAL datoteka spojiti u jednu, glavnu konfiguracijsku datoteku koja se učitava prilikom pokretanja sustava [4].

U LinuxCNC sustavu već postoji veliki broj unaprijed konfiguriranih podsustava, a po potrebi mogu se modificirati ili u potpunosti izostaviti ovisno o razmatranoj primjeni. Također je moguće napisati HAL datoteke za vlastite korisničke podsustave i to ili preko programabilnog logičkog kontrolera (PLC je jedan od modula u HAL-u) ili korištenjem posebnog alata za pisanje HAL modula (*Comp*). PLC se konfigurira u početnoj fazi podešavanja sustava korištenjem ljestvičastih dijagrama unutar zasebnog grafičkog sučelja (*ClassicLadder*), dok se *Comp* upotrebljava za izradu, kompajliranje i instalaciju HAL modula iz izvornog koda [4].

2.2.4. GUI

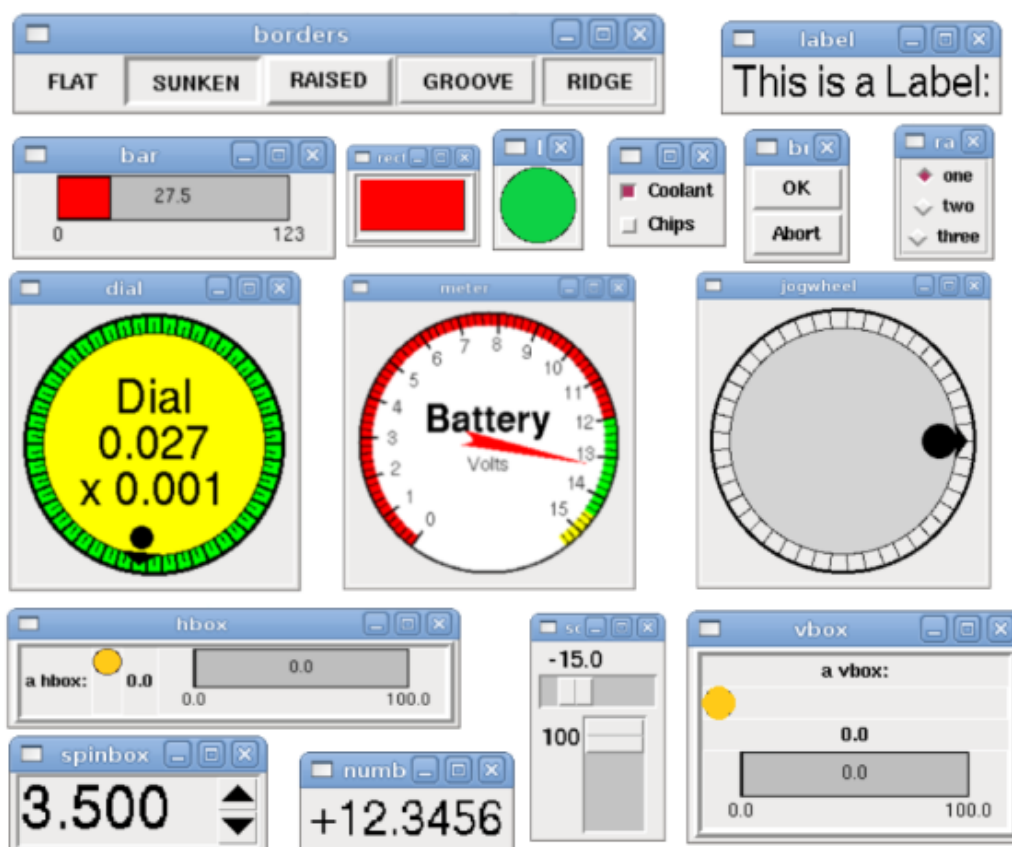
U sklopu LinuxCNC-a postoji nekoliko različitih grafičkih korisničkih sučelja, a neki od njih su *Axis*, *Gmoccapy*, *Touchy* i *Gscreen*. *Axis* sučelje je najnaprednije i predviđeno je za korištenje na standardnom računalu s monitorom. *Touchy* sučelje prilagođeno je za rad na zaslonima na dodir, a *Gmoccapy* se može koristiti na oba načina. Na slici 2-3. može se vidjeti primjer GUI-a koji sadrži sve funkcije potrebne za referenciranje, upravljanje i nadzor troosnog numerički upravljanog stroja (najviše se može upravljati do 9 različitih osi).



Slika 2-3. Gmoccapy grafičko korisničko sučelje

LinuxCNC sadrži alate za konfiguriranje i podešavanje sustava u stvarnom vremenu i oni imaju vlastita grafička sučelja. *ClassicLadder* pruža mogućnost izrade i uređivanja ljestvičastih logičkih dijagrama PLC-ova, softverski osciloskop pomoću kojeg je moguće vizualizirati HAL signale zove se *HALScope*, a *HALConfig* služi za parametriziranje i testiranje konfiguracije stroja.

Standardna korisnička sučelja mogu se prilagoditi željama korisnika i potrebama razvijanog sustava dodavanjem raznih virtualnih kontrolnih panela kao što su klizači, prekidači, LED indikatori, gumbi i slično (slika 2-4.), a njihovom kombinacijom moguće je napraviti i vlastiti GUI. Na ovaj način jednostavno je proširiti funkcionalnost korisničkog sučelja i povećati njegovu fleksibilnost.



Slika 2-4. Virtualni kontrolni paneli

Virtualni kontrolni paneli (VCP) napisani su u XML jeziku, a u HAL prostoru učitavaju se kao komponente s vlastitim ulazima i izlazima i pripadaju komponentama koje se ne izvršavaju u stvarnom vremenu. Ispod je naveden izgled XML datoteke koja opisuje *bar* VCP (slika 2-5.), a koji se može vidjeti u gornjem lijevom kutu slike 2-4.

```
<pyvcp>
  <bar>
    <halpin>"my-bar"</halpin>
    <min_>0</min_>
    <max_>123</max_>
    <bgcolor>"grey"</bgcolor>
    <fillcolor>"red"</fillcolor>
  </bar>
</pyvcp>
```

Slika 2-5. XML opisna datoteka

2.3. Hardverski apstrakcijski sloj

Hardverski apstrakcijski sloj (HAL) dio je LinuxCNC sustava koji omogućuje jednostavno učitavanje i funkcionalno povezivanje većeg broja pogonskih programa hardverskih komponenata (engl. *device drivers*) i raznih softverskih modula (kao što su *motmod*, *iocontrol* ili *halscope*). Oni predstavljaju osnovne gradivne blokove u HAL prostoru [3].

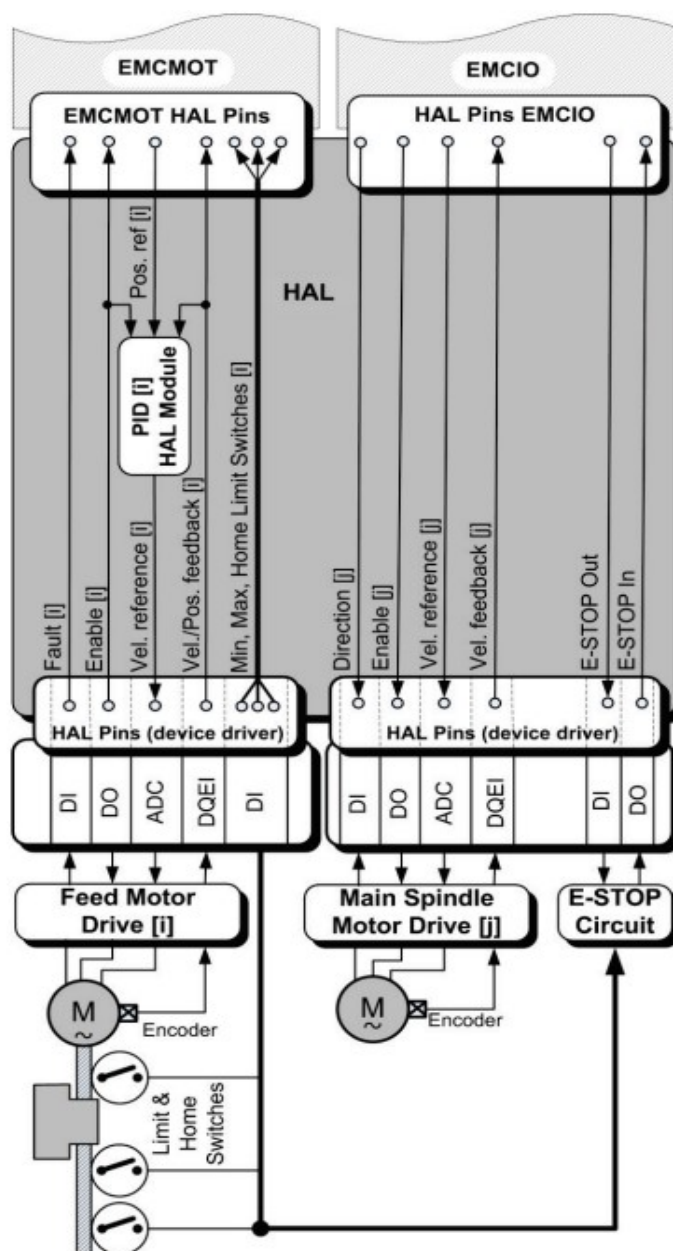
HAL omogućava izmjenu podataka u stvarnom vremenu između pojedinih modula LinuxCNC sustava, te između LinuxCNC-a i upravljanog hardvera ili softverskih komponenata nižeg reda. Također, implementacijom apstrakcijskog sloja postignuta je visoka razina fleksibilnosti upravljačkog sustava jer ga se može jednostavno prilagoditi za različite konfiguracije. Učitavanje, isključivanje i uređivanje HAL modula moguće je i za vrijeme rada LinuxCNC sustava, što je velika prednost pogotovo u fazi testiranja rada sustava [4].

Principi na kojima se temelji HAL jednaki su onima koji vrijede kod spajanja fizičkih komponenti prilikom gradnje CNC stroja. CNC strojevi sastoje se od komponenata kao što su varijabilni frekvencijski pretvarači za upravljanje frekvencijom glavnog vretena, PLC-a za kontrolu sustava izmjene alata kao i niza drugih perifernih funkcija stroja, regulatora motora posmičnih osi, graničnih prekidača, enkodera i ostalih. Jednom kad se odaberu komponente od kojih će se stroj izgraditi, potrebno ih je integrirati u jednu cjelinu i međusobno povezati upotrebom elektro spojnih shema kojima se prikazuje način povezivanja pojedinih komponenata.

Analogno tome, moguće je izraditi složene upravljačke strukture unutar HAL prostora. Prvo se odaberu HAL moduli (komponente) od kojih će se sustav sastojati (kao što su generator signala koraka, *driver* za paralelni port ili PID regulator), a zatim se pojedine komponente povežu HAL signalima u funkcionalnu cjelinu. Svaka komponenta u HAL prostoru može se zamisliti kao crna kutija s pinovima (izvodnicama) koji predstavljaju ulaze i/ili izlaze.

HAL se u strukturi LinuxCNC-a nalazi između modula EMCIO/EMCMOT i drivera komunikacijskih i upravljačkih kartica (slika 2-5.) koji omogućavaju komunikaciju između modula u HAL prostoru i podređenog hardvera. Nedostatak odgovarajućih pogonskih programa (drivera) i dokumentacije za Linux sustave, za većinu trenutno dostupnih komunikacijskih i upravljačkih kartica na tržištu, predstavlja jedno od najvećih ograničenja prilikom njihove implementacije u LinuxCNC sustav [4].

Slika 2-6. prikazuje primjer konfiguracije hardverskog apstrakcijskog sloja za slučaj upravljanja trima posmičnim osima i prigonom glavnog vretena. HAL signalima omogućena je komunikacija između logičkog i kontrolera gibanja s jedne strane i pogonskih programa upravljačkih i komunikacijskih kartica s druge. Posmičnim prigonima se upravlja po poziciji (PID modul zatvara pozicijsku petlju i šalje referencu brzine prema regulatorima motora), a prigonom glavnog vretena se upravlja po brzini. Povratno se u upravljački sustav šalju signali s graničnih i referentnih prekidača, sigurnosnog kruga i enkodera.



Slika 2-6. Primjer konfiguracije hardverskog apstrakcijskog sloja [4]

Osnovna struktura HAL prostora može se opisati pomoću sljedećih pojmova [3]:

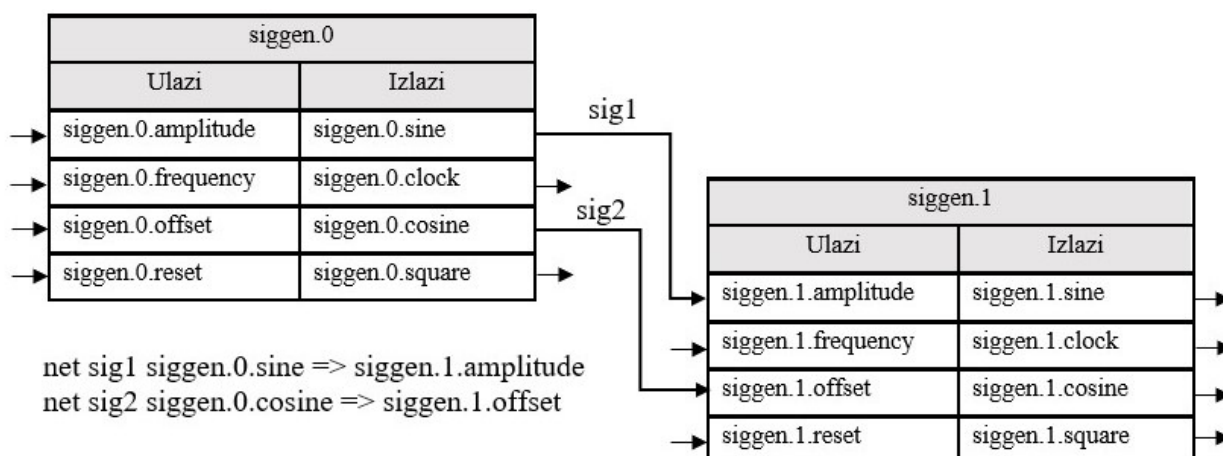
- **komponente** – softverski kod s jasno definiranim ulazima, izlazima i ponašanjem. Povezivanjem više komponenata gradi se HAL prostor.
- **parametri** – vrijednosti koje opisuju stanje pinova neke komponente. Ulazne parametre podešava korisnik, a izlazni parametri su fiksni i mogu se samo mjeriti (promatrati).
- **pinovi** – predstavljaju softversku reprezentaciju fizičkih izvodnica hardverskih komponenti. Svaki pin ima posebno ime što je bitno kod njihovog kasnijeg povezivanja.
- **signali** – povezuju HAL pinove. Predstavljaju zamjenu za žice u fizičkom sustavu.

Petlja (engl. *thread*) je komponenta u HAL prostoru koja se ciklički izvršava u diskretnim vremenskim intervalima. Uobičajeno je da se kod izgradnje HAL prostora koriste najmanje dvije petlje i to jedna brža (npr. s periodom od 1 ms) u kojoj će se izvršavati funkcije komponenata s najvišim prioritetom (one povezane s upravljanjem gibanjem posmičnih osi), a u sporijoj petlji će se izvršavati funkcije komponenata koje upravljaju perifernim procesima (izmjenom alata i slično). Funkcija je programski kod pridružen komponenti čija je uloga čitanje ulaza, te izračun i pisanje izlaza, a period izvršavanja joj ovisi o periodu petlje kojoj je pridružena.

Postoje različiti tipovi podataka HAL pinova i signala, stoga je prilikom njihovog povezivanja potrebno pripaziti da su i pinovi koji se povezuju i signali koji ih povezuju istog tipa. Četiri osnovna tipa podataka jesu:

- bit – binarna vrijednost, 1 ili 0
- float – 64-bitni decimalni broj
- u32 – 32-bitni cijeli broj bez predznaka (samo pozitivne vrijednosti)
- s32 – 32-bitni cijeli broj s predznakom (može poprimiti i negativne vrijednosti)

Na slici 2-7. grafički je prikazan način povezivanja dvije komponente u HAL prostoru. Sinusni i kosinusni izlazi nultog signal generatora povezani su preko signala sig1 i sig2 s ulazima prvog signal generatora. Također je navedena i naredba koja bi se koristila za spajanje te dvije komponente u HAL prostoru.



Slika 2-7. Povezivanje komponenta unutar HAL prostora

Općenito se komponente u HAL prostoru mogu podijeliti na one koje se izvršavaju u stvarnom vremenu (engl. *realtime*) i one koje se ne izvršavaju u stvarnom vremenu (u LinuxCNC-u nazvane *userspace* komponentama). Prethodno spomenuti se pogonski programi hardverskog sučelja (raznih upravljačkih i komunikacijskih kartica) također u HAL prostor učitavaju kao HAL komponente. Većina takvih drivera predviđena je za rad u stvarnom vremenu. U tablici 2-1. navedeno je nekoliko primjera oba tipa komponenta uz kratki opis. Detaljni ispis svih komponenta naveden je u [3].

Tablica 2-1. HAL komponente

<i>Userspace</i> komponente		<i>Realtime</i> komponente	
Naziv	Opis	Naziv	Opis
<i>halmeter</i>	-softverski voltmetar, služi za mjerenje stanja pinova, signala i parametara	<i>siggen</i>	-generator signala sinusnog, kvadratnog, trokutastog i pilastog valnog oblika
<i>halscope</i>	-softverski osciloskop, služi za vizualizaciju signala	<i>stepgen</i>	-generator impulsa koraka, koristi se za upravljanje koračnim motorima
<i>iocontrol</i>	-prima I/O NML poruke i komunicira s HAL-om	<i>pmwgen</i>	-komponenta za generiranje pulsno-širinski moduliranih signala
<i>pyvcp</i>	-virtualni kontrolni paneli iz pyVCP biblioteke	<i>pid</i>	-PID regulator za regulaciju pozicije ili brzine u zatvorenoj petlji
<i>gladevcp</i>	-virtualni kontrolni paneli iz GTK biblioteke	<i>encoder</i>	-mjeri poziciju brojanjem kvadraturnih signala enkodera

3. PARAMETRIRANJE REGULATORA POSMIČNIH OSI U UPRAVLJAČKOM SUSTAVU LINUXCNC

U ovom poglavlju opisan je način parametriranja regulatora posmičnih osi koračnih i servomotora u LinuxCNC sustavu. Također je, kao uvod u podešavanje zatvorene regulacijske petlje regulatora servomotora (poglavlje 7.), dan kratki pregled osnovnih pojmova iz teorije automatske regulacije.

3.1. Pregled teorije automatske regulacije

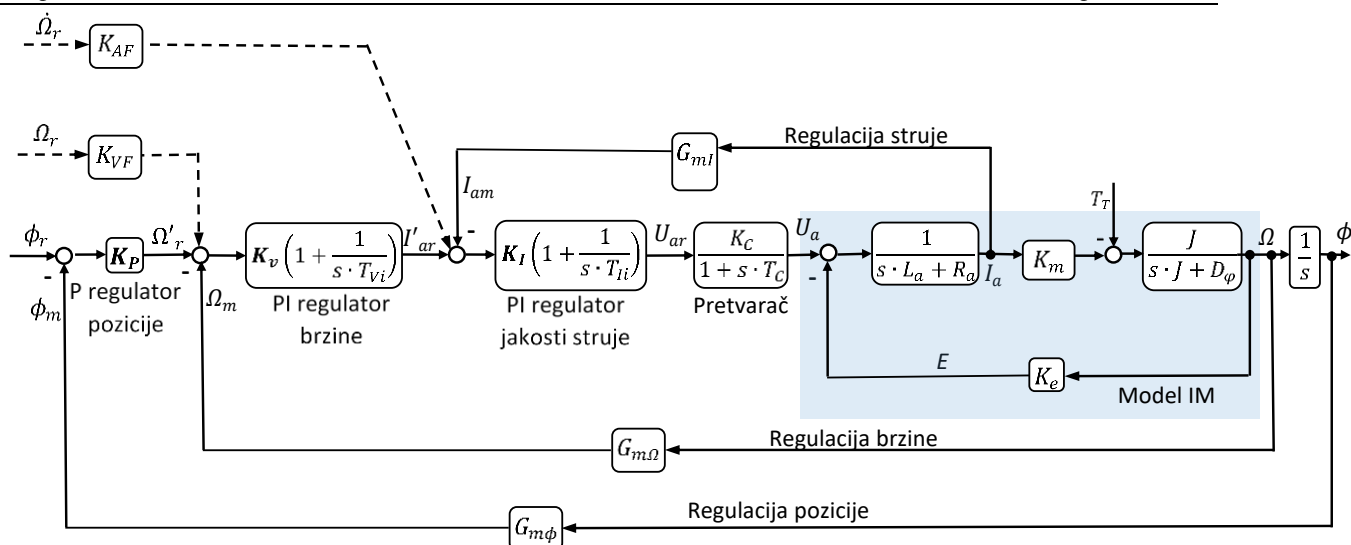
3.1.1. Kaskadna regulacija i PID regulator

Kaskadna regulacijska struktura posmičnih prigona sastavljena je od tri nezavisne regulacijske petlje i to redom (od najbrže do najsporije) po jakosti struje (odnosno momentu), brzini i poziciji (slika 3-1, prilagođeno iz [6]). Prednosti hijerarhijske strukture su jednostavno prebacivanje između različitih načina rada, lokalno otklanjanje poremećaja u podređenim kaskadama, mogućnost limitiranja varijabli stanja, te jednostavna sinteza regulatora.

Regulator momenta podređen je regulatoru brzine, a on je podređen regulatoru pozicije. Nadređeni regulator daje referencu podređenom regulatoru. Ulaz u svaki regulator predstavlja regulacijsko odstupanje (grešku) koje se dobije oduzimanjem izmjerene vrijednosti od referentne vrijednosti određene varijable stanja (momenta, brzine ili pozicije). Senzori koji se koriste za njihovo mjerenje (enkoder, rezolver) modelirani su proporcionalnim članovima s kašnjenjem prvog reda. Izlaz iz regulatora struje je referenca napona koju elektronički pretvarač pretvara u napon motora (slika 3-1.).

Referenca pozicije u regulator pozicije dolazi iz NC jezgre. Petlja pozicije može biti zatvorena ili u NC jezgri ili u servo regulatoru, dok se regulacijske petlje po brzini i momentu gotovo uvijek realiziraju u servo regulatoru. Najčešći je slučaj da se pozicijska petlja smjesti u NC jezgru, a onda se iz nje kroz hardversko sučelje prigona šalje referenca brzine prema podređenim regulatorima motora.

Jedan nedostatak strukture pojedinog regulatora unutar kaskade, je taj da ne uklanja trajno regulacijsko odstupanje. Iz tog razloga potrebno je dodati unaprijednu regulaciju (prikazana isprekidanim crtama na slici 3-1.) koja za cilj ima smanjenje (uklanjanje) tog odstupanja. Prvo se dodaje unaprijedno pojačanje po brzini koje povećava iznos reference brzine (referenca brzine iz regulatora pozicije i referenca brzine pomnožena s unaprijednim pojačanjem po brzini se zbroje) i tako se ubrzava odziv sustava. Kako bi se uklonili prebačaji uzrokovani unaprijednom regulacijom po brzini, uvodi se unaprijedna petlja po akceleraciji koja dodatno ubrzava odziv regulacijskog kruga [6].



Legenda:

- | | | |
|---|--------------------------------|---|
| $\dot{\Omega}_r$ – referenca akceleracije | ϕ – pozicija rotora | R_a – električni otpor armature |
| Ω_r, Ω'_r – referenca brzine | Ω – brzina rotora | K_{VF} -unaprijedno pojačanje po brzini |
| ϕ_r – referenca pozicije | I_a – jakost struje armature | K_{AF} -unaprijedno pojačanje po akceleraciji |
| I'_{ar} – referenca jakosti struje | U_a – napon armature | K_P – pojačanje proporcionalnog djelovanja |
| U_{ar} – referenca napona | E – inducirani napon | K_V – pojačanje derivacijskog djelovanja |
| ϕ_m – izmjerena pozicija | T_T – moment tereta | K_I – pojačanje integralnog djelovanja |
| Ω_m – izmjerena brzina | J – moment inercije | T_{Vi}, T_{Ii}, T_C – vremenske konstante |
| I_{am} – izmjerena jakost struje | L_a – induktivitet armature | K_C – pojačanje pretvarača |

Slika 3-1. Kaskadna regulacijska struktura

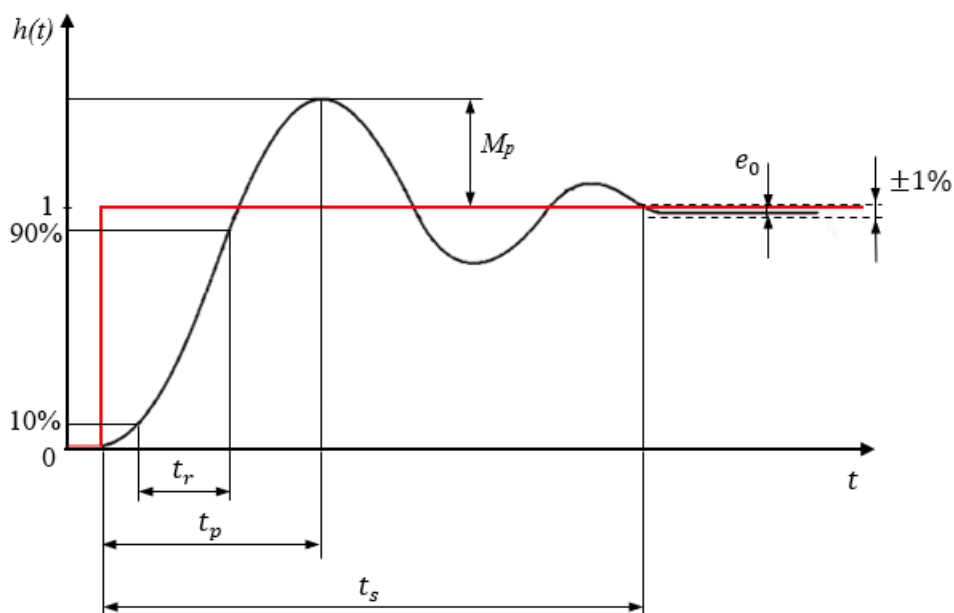
Osnovni dinamički članovi regulatora su proporcionalni, integralni i derivacijski. Proporcionalni član množi trenutnu regulacijsku pogrešku s odgovarajućim pojačanjem, integralni član integrira prošle vrijednosti greške u vremenu i omogućuje otklanjanje trajnog regulacijskog odstupanja, a derivacijski prati brzinu promjene greške, te pojačava upravljačku varijablu u slučaju njezinog porasta. Njihovom kombinacijom dobivaju se različite konfiguracije regulatora kao što su PI, PD i PID regulator. Strujna petlja gotovo uvijek je izvedena s PI regulatorom [6], dok brzinska i pozicijska petlja poprimaju različite konfiguracije (u ovom slučaju P-PI). U tablici 3-1. (preuzeto iz [7]) naveden je utjecaj pojedinog člana na karakteristike odziva sustava koje su prikazane na slici 3-2. Parametre regulatora moguće je odrediti eksperimentalno (više riječi o ovom načinu u nastavku) i analitički.

Tablica 3-1. Utjecaj osnovnih dinamičkih članova na karakteristike odziva sustava

Parametar	Vrijeme porasta	Prebačaj	Vrijeme smirivanja	Trajno regulacijsko odstupanje
P	smanjuje	povećava	mali utjecaj	smanjuje
I	smanjuje	povećava	povećava	uklanja
D	mali utjecaj	smanjuje	smanjuje	mali utjecaj

3.1.2. Analiza sustava u vremenskoj domeni

Prilikom podešavanja željenih svojstava odziva dinamičkog sustava, potrebno je definirati veličine preko kojih se odziv sustava može kvantificirati. Slika 3-2. prikazuje karakteristične veličine koje se koriste za opis odziva sustava na jediničnu odskočnu funkciju (označena crvenom bojom na slici) u vremenskom području. Prikazani izgled krivulje tipičan je za prigušeni oscilatorni odziv, a odgovara odzivu proporcionalnog člana drugog reda (P_2) na jediničnu pobudu. U najvećem broju slučajeva cilj podešavanja sustava (parametara regulatora) je postizanje najbržeg mogućeg odziva sustava pri kojem još ne dolazi do prebačaja (granični aperiodski odziv). Nužan preduvjet za postizanje točnosti je stabilnost sustava koja je funkcija samog sustava.



Slika 3-2. Karakteristične veličine za opis odziva sustava na jediničnu odskočnu funkciju [7]

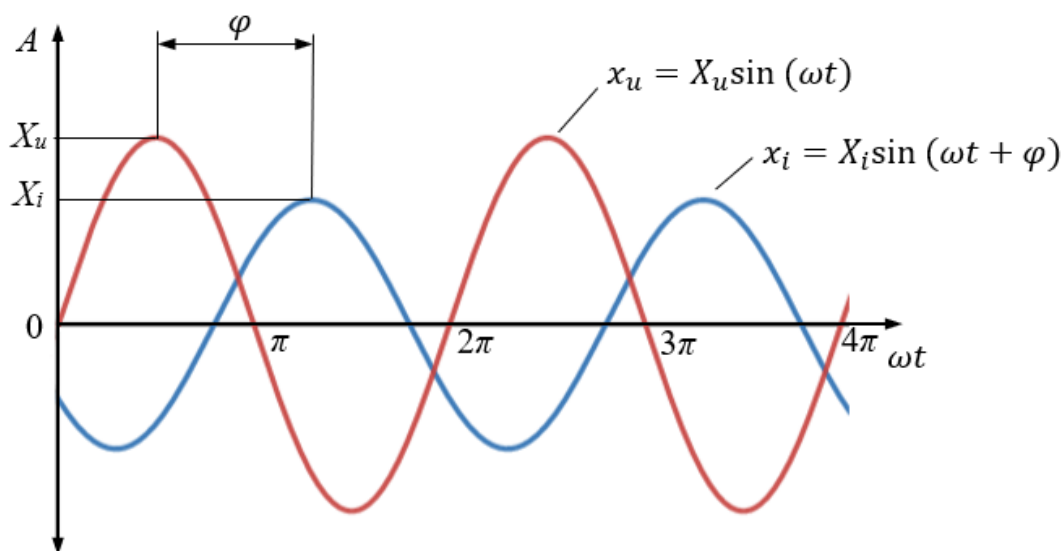
Gdje su:

- t_s - vrijeme smirivanja \rightarrow vrijeme potrebno da odziv postigne stacionarnu vrijednost odnosno da njegova vrijednost oscilira najviše $\pm 1\%$ (može biti i 2 ili 5% ovisno o izvoru) od željene vrijednosti
- t_r - vrijeme porasta \rightarrow vrijeme u kojem odziv sustava poraste od 10% na 90% svoje konačne vrijednosti
- t_p – vrijeme najvećeg prebačaja (vrijeme prvog prebačaja)
- M_p - postotni iznos najvećeg prebačaja
- e_0 - trajno regulacijsko odstupanje \rightarrow regulacijska pogreška u stacionarnom stanju

3.1.3. Analiza sustava u frekvencijskoj domeni

Analizi i sintezi sustava može se pristupiti i u frekvencijskom području. Na ulaz linearnog vremenski invarijantnog sustava dovodi se sinusna funkcija. Na izlazu sustava dobije se sinusna funkcija iste frekvencije, ali različite amplitude i različitog faznog pomaka (označena plavom bojom na slici 3-3.). Promjenom frekvencije ulaznog signala moguće je analizirati promjenu amplitude i faznog pomaka odziva sustava. Ovo je moguće grafički predočiti u obliku dvaju Bodeovih dijagrama kojima se prikazuje promjena omjera amplitude ulaznog i izlaznog signala (logaritamski omjer iskazan u decibelima), te faznog pomaka u ovisnosti o promjeni frekvencije. Osim Bodeovog, koriste se i Nyquistov, te Nicholsov dijagram [7].

Lomna frekvencija u Bodeovom dijagramu predstavlja onu frekvenciju iznad koje dolazi do prigušenja odziva (smanjenja amplitude izlaznog u odnosu na ulazni signal). U kontekstu podešavanja regulacijske petlje brzine, dobro je da je ta frekvencija što veća jer se na taj način i dinamika sustava povećava pa je regulacijski krug sposoban djelovati na pojavu šumova i poremećaja koji se često pojavljuju upravo pri višim frekvencijama. Ispravnim odabirom parametara regulatora moguće je povećati pojasnu širinu sustava (pomaknuti lomnu frekvenciju u desno) [6]. Pojasna širina je mjera za brzinu odziva sustava u frekvencijskom području. Rezonantno izdizanje je pojava do koje dolazi kod slabo prigušenih sustava na rezonantnoj frekvenciji, a koristi se kao mjera za relativnu stabilnost zatvorenog kruga [7].



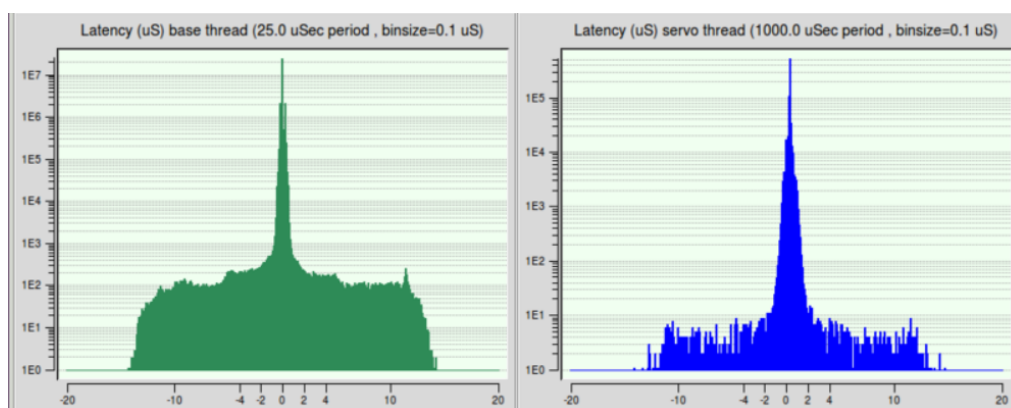
Slika 3-3. Analiza sustava u frekvencijskom području

Gdje su:

- X_u – amplituda ulaznog signala
- ω – kružna frekvencija sustava
- X_i – amplituda izlaznog signala
- φ – fazni pomak između ulaznog i izlaznog signala

3.2. Ispitivanje latencije

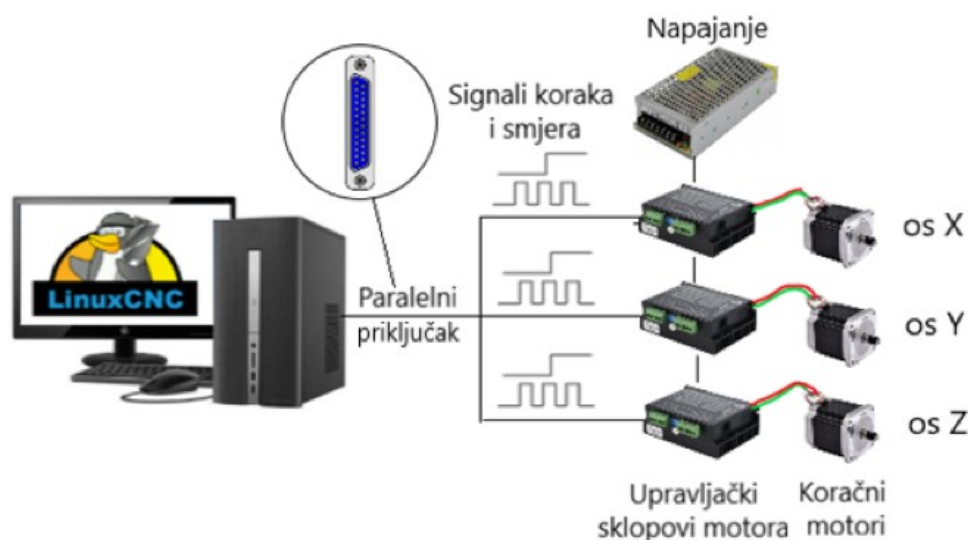
Prije softverskog generiranja upravljačkih signala, potrebno je provjeriti latenciju, odnosno rasipanje u vremenskom intervalu u kojem je ciklički potrebno izvršavati pojedine operacije. Slika 3-4. prikazuje histograme rasipanja za dvije petlje različitih perioda. Vidljivo je da period izvršavanja petlji nije konstantan i da brža petlja (očekivano) ima veće rasipanje.



Slika 3-4. Histogram latencije – lijevo petlja 25 μ s, desno petlja 1000 μ s [3]

3.3. Parametriranje regulatora koračnih motora

Koračnim motorima upravlja se signalima koraka i smjera u otvorenoj upravljačkoj petlji. Često se koriste za upravljanje posmičnim gibanjima 3D printera, plazma rezača i sličnih strojeva gdje ne postoje značajniji poremećaji u obliku obradnih sila ili slično. Slika 3-5. shematski prikazuje način upravljanja koračnim motorima korištenjem LinuxCNC-a. Upravljački signali generiraju se softverski ili specijaliziranim upravljačkim karticama, od kuda se šalju na izvodnice paralelnog (LPT) priključka ili druge upravljačke kartice dalje do regulatora prigona osi.

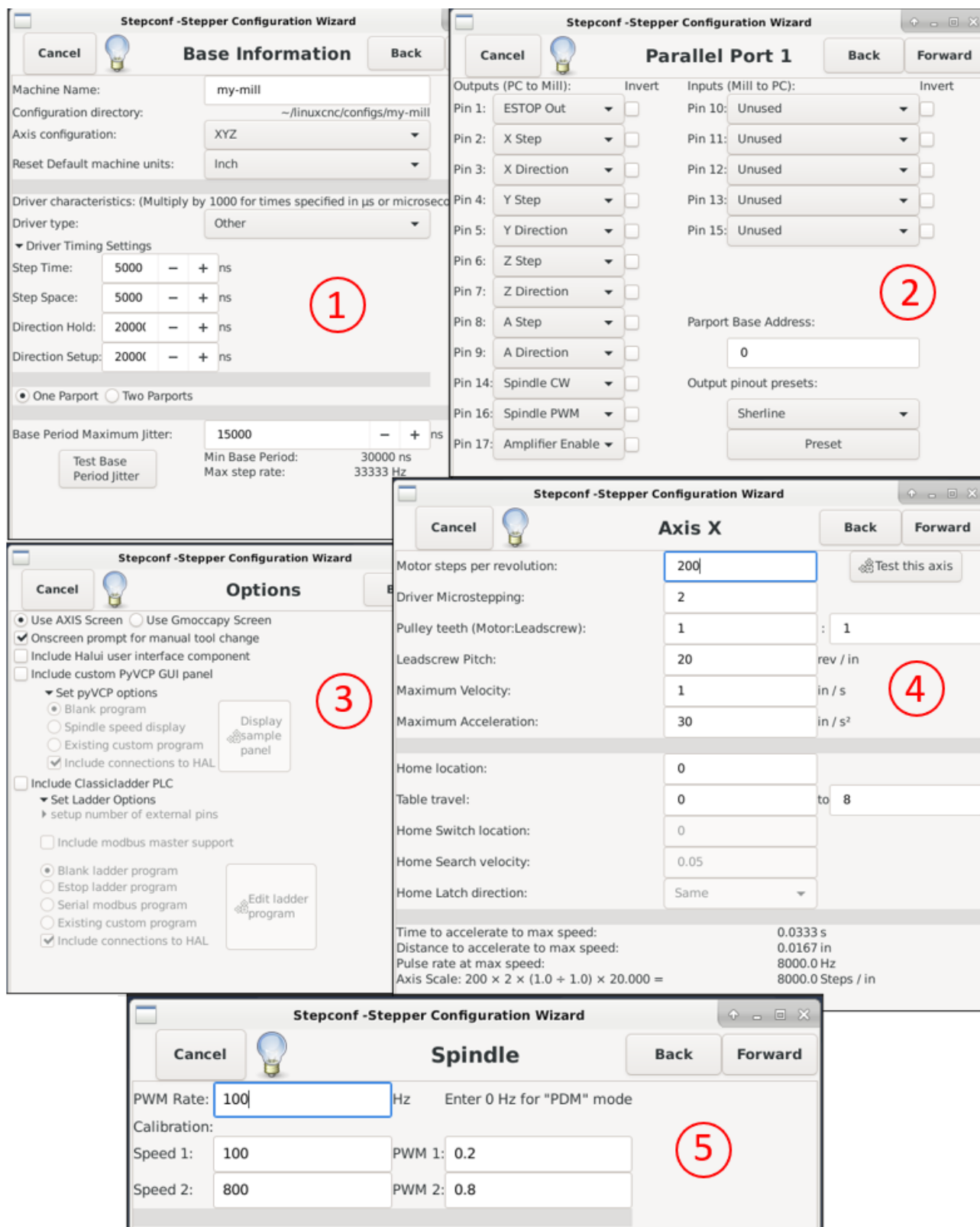


Slika 3-5. Shema upravljanja koračnim motorima

3.3.1. Stepper Configuration Wizard

Konfiguriranje stroja s koračnim motorima se u LinuxCNC-u provodi u posebnom grafičkom sučelju *Stepper Configuration Wizard*. Uvjet za korištenje ovakvog načina konfiguriranja je taj da se prijenos signala vrši preko paralelnog priključka. Slika 3-6. prikazuje osnovne korake ovog postupka. Rezultat konfiguriranja odgovarajuće su HAL i INI konfiguracijske datoteke, a po potrebi ih je moguće i uređivati. Koraci kod ovog postupka su sljedeći [3]:

- 1) U prvom programskom prozoru koji se pojavi potrebno je unijeti osnovne informacije kao što su naziv stroja, konfiguracija osi stroja (3-osna ili 4-osna glodalica, ili tokarilica) i zadane mjerne jedinice (milimetri ili inči). Zatim se odabire tip regulatora motora (*Gecko*, *Sherline*, *Xylotex*), nakon čega sustav automatski postavi vremenske parametre regulatora. U slučaju da korišteni tip regulatora nije ponuđen, ove parametre je potrebno unijeti ručno. Završno se unosi maksimalno kašnjenje prethodno izmjereno testiranjem latencije.
- 2) U sljedećem koraku se pinovima paralelnog priključka pridružuju odgovarajući signali. Signali koji će se koristiti određeni su konfiguracijom stroja, a u ovom slučaju prikazan je primjer za 4-osnu glodalicu. Izlazni signali su signali koraka i smjera posmičnih osi, signal dozvole, te signali za određivanje smjera i brzine vrtnje glavnog vretena. Na ulazne pinove paralelnog porta, moguće je dovesti referentne i granične signale pojedinih osi, te signal s tipkala za zaustavljanje u nuždi.
- 3) U trećem koraku odabire se GUI koji će se koristiti (*Axis* ili *Gmoccapy*). Moguće je dodati virtualne upravljačke panele (poglavlje 2.2.4.), PLC funkcionalnost i uključiti obavijesti za izmjenu alata (ako se koristi naredba M6 u G kodu).
- 4) Sada je potrebno unijeti parametre kojima su opisane karakteristike posmičnih osi (slika 3-6. prikazuje samo X os, ali je postupak jednak za sve osi). Parametri koje je potrebno navesti su sljedeći: broj koraka potrebnih za jedan okret motora (rezolucija motora), prijenosni omjer prigona, uspon navojnog vretena, maksimalna brzina i akceleracija osi, početna pozicija, hod osi, parametri regulatora za generiranje mikro koraka i ostali.
- 5) U zadnjem koraku potrebno je unijeti frekvenciju PWM signala. Također je potrebno unijeti dvije vrijednosti frekvencije vrtnje glavnog vretena i za tu svrhu pridruženih vrijednosti PWM signala, kako bi se uspostavila linearna veza između njih. Ove vrijednosti se podešavaju u procesu kalibracije glavnog vretena.



Slika 3-6. Koraci za konfiguriranje stroja s koračnim motorima

3.4. Parametriranje regulatora servomotora

Upravljanje u zatvorenoj petlji ima brojne prednosti u odnosu na otvorenu petlju, ali je zato potrebno provesti postupak podešavanja parametara regulatora kako bi se dobile željene karakteristike odziva. Neke od prednosti primjene servomotora su brži odziv, velika točnost pozicioniranja, podnošenje većih opterećenja na višim frekvencijama vrtnje i općenito bolje dinamičke karakteristike u odnosu na koračne motore.

Kaskadna regulacijska struktura dozvoljava (u odnosu na kompaktnu strukturu regulatora) postepeno podešavanje parametara pojedinih regulatora i to tako da se prvo podešavaju parametri strujne petlje, a nakon toga petlje po brzini. Parametri pozicijske petlje podešavaju se tek nakon što su prethodne dvije petlje ispravno podešene.

Određivanje parametara regulatora može se izvršiti analitički ili eksperimentalno. Kod analitičkog načina potrebno je poznavati matematički model sustava. Neke od analitičkih metoda su metoda poništavanja polova, metoda podešavanja polova i metoda frekvencijskog odziva. Eksperimentalni način podrazumijeva prilagodbe parametara na temelju iskustva odnosno ručnu promjenu parametara i praćenje njihovog utjecaja na karakteristike odziva (brzinu porasta, vrijeme smirivanja, postotni prebačaj, trajno regulacijsko odstupanje i drugih). Utjecaj promjene pojedinih parametara na karakteristike odziva je prethodno naveden u tablici 3-1. U velikoj većini slučajeva cilj je postići granični aperiodski odziv. On predstavlja najbrži mogući odziv kod kojeg još ne dolazi do prebačaja (sustav je granično prigušen). U LinuxCNC sustavu se za vizualizaciju referentnih signala i odziva sustava koristi HAL osciloskop.

3.4.1. Regulacijska petlja momenta (struje)

Podešavanje strujne petlje uglavnom nije potrebno provoditi ukoliko se motor i upravljački sklop motora kupuju od istog proizvođača. U tom slučaju su parametri unaprijed tvornički podešeni. Regulator struje gotovo je uvijek izveden kao PI regulator. U slučaju da je ipak potrebno podesiti parametre regulatora struje, oni se podešavaju slično kao kod brzinske petlje pa se ovdje neće detaljnije objašnjavati. Važno je napomenuti kako je to prva petlja koju je potrebno podesiti i ona izravno upravlja momentom motora (što se vidi iz osnovne jednadžbe istosmjernog motora ($T(t) = K_m \cdot i_a(t)$). Moment motora određuje dinamičke karakteristike motora odnosno ubrzanje motora, te je zadužen za otklanjanje vanjskih poremećaja (npr. sila rezanja) s ciljem održavanja konstantne brzine vrtnje. Strujna petlja najbrže se izvršava i zbog toga je izvedena hardverski (nalazi se u upravljačkom sklopu motora).

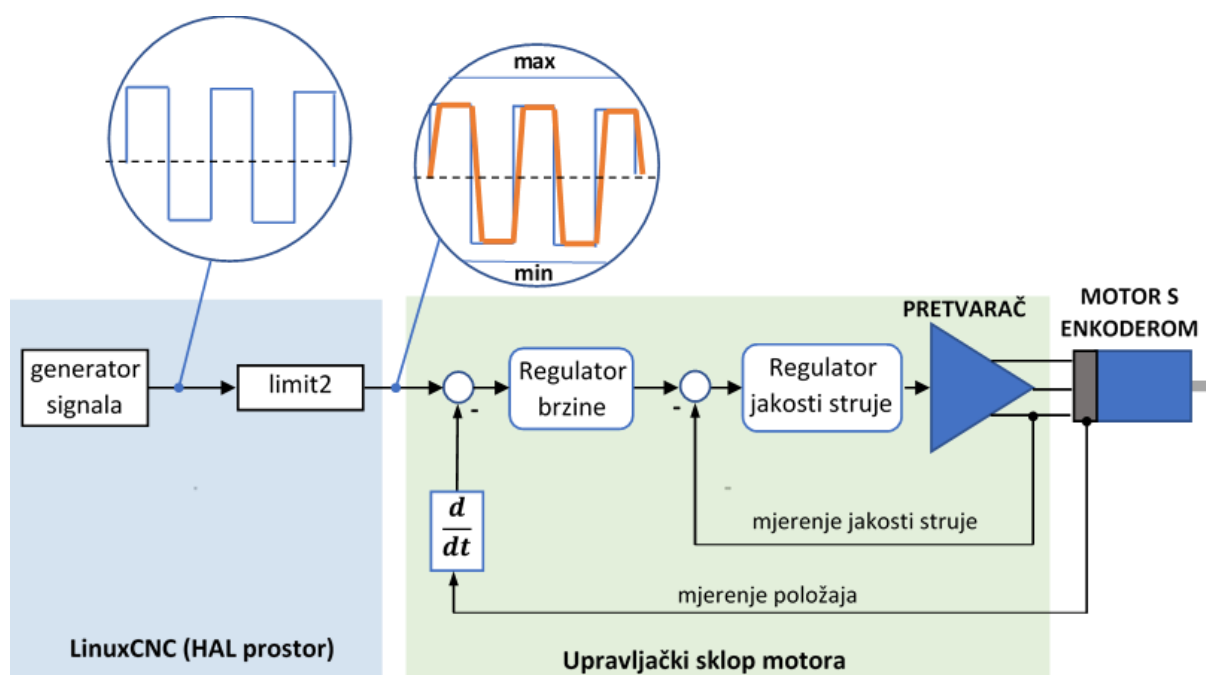
3.4.2. Regulacijska petlja brzine

Sljedeća petlja koju je potrebno podesiti je regulacijska petlja brzine (frekvencije vrtnje motora). Jednom kad se u potpunosti namjeste sve regulacijske petlje, referenca brzine u regulator brzine dolazi iz regulatora pozicije. No, u postupku namještanja parametara regulatora brzine, referenca brzine šalje se iz generatora signala (*siggen* komponenta u HAL prostoru).

Signal kojim se šalje referenca brzine može biti različitih valnih oblika. Neki od primjera signala koji se koriste su trokutasti, sinusni i trapezoidni. Kvadratne signale nije preporučljivo koristiti jer su prenaglog karaktera i doveli bi motor u stanje velikih mehaničkih udarnih opterećenja koja bi mogla oštetiti motor i ostale strojne elemente.

Kako bi se zaštitio motor, potrebno je ograničiti vrijednost i dinamiku referentnog signala brzine pomoću limitatora. On je inače izveden hardverski u upravljačkom sklopu motora, ali je na slici 3-7. prikazan kao komponenta u HAL prostoru, kako bi se zornije objasnila njegova uloga. *Limit2* je HAL komponenta koja ograničava i limitira dinamiku ulaznog signala. Vrijednost je limitirana parametrima *min* i *max*, a dinamika parametrom *maxv* kojim je propisana maksimalna brzina do koje motor može ubrzati u jednoj sekundi.

Podešavanjem parametara regulatora brzine (prvo P pojačanja, a zatim I), nastoji se minimizirati regulacijska pogreška između referentnog signala (u ovom slučaju trapezoidnog) i izmjenjenog signala brzine s enkodera koji se dobije deriviranjem signala pozicije (slika 3-7.).



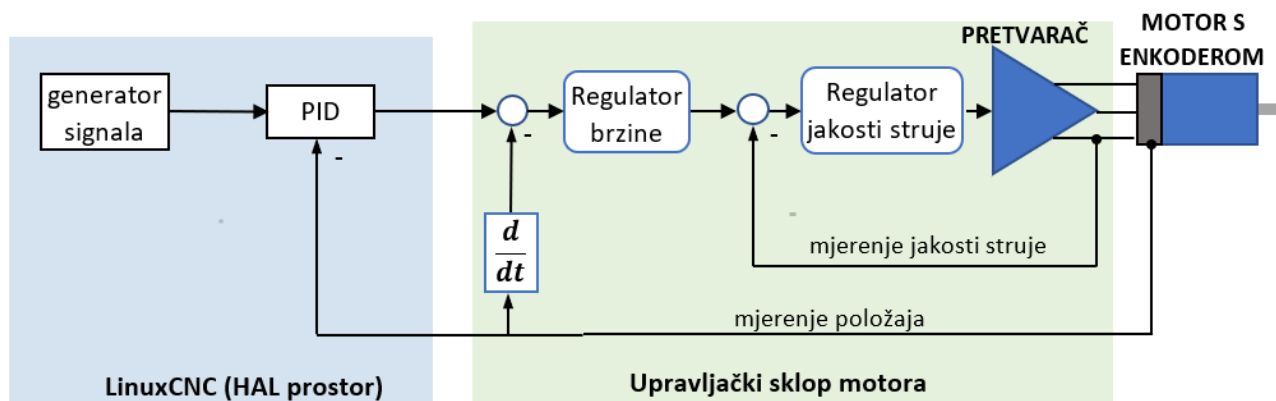
Slika 3-7. Shema podešavanja regulacijske petlje po brzini

3.4.3. Regulacijska petlja pozicije

U normalnom režimu rada, referenca pozicije se u regulator pozicije šalje iz jezgre numeričkog upravljanja (interpolatora). Tijekom podešavanja parametara regulatora pozicije, referentni signal pozicije nadomjesno se šalje iz generatora signala. Tipovi signala jednaki su onima prethodno korištenima za referencu brzine.

Regulator pozicije učitava se kao PID komponenta u HAL prostoru. U njega se dovode referentni signal pozicije iz generatora signala i signal stvarne pozicije motora s enkodera (slika 3-8.). Glavni zadatak regulatora pozicije je ukloniti regulacijsku grešku koja se dobije oduzimanjem ta dva signala. Regulacijska pogreška pozicije množi se s proporcionalnim pojačanjem (regulator je P tipa pa ima samo proporcionalni član) i šalje na izlaz kao referenca brzine podređenom regulatoru brzine.

Iterativnim postupkom potrebno je pronaći optimalno proporcionalno pojačanje regulatora koje će osigurati najbrži odziv sustava kod kojeg još nema prebačaja. Pozicijska petlja, podešena na ovaj način, nije sposobna s visokom točnošću reagirati na brze promjene reference pozicije. Iz tog razloga joj je potrebno povećati dinamiku uvođenjem unaprijednih pojačanja prvog i drugog reda koji se također podešavaju iterativnim postupkom.



Slika 3-8. Zatvaranje regulacijske petlje po poziciji u HAL prostoru

4. ETHERCAT SABIRNICA

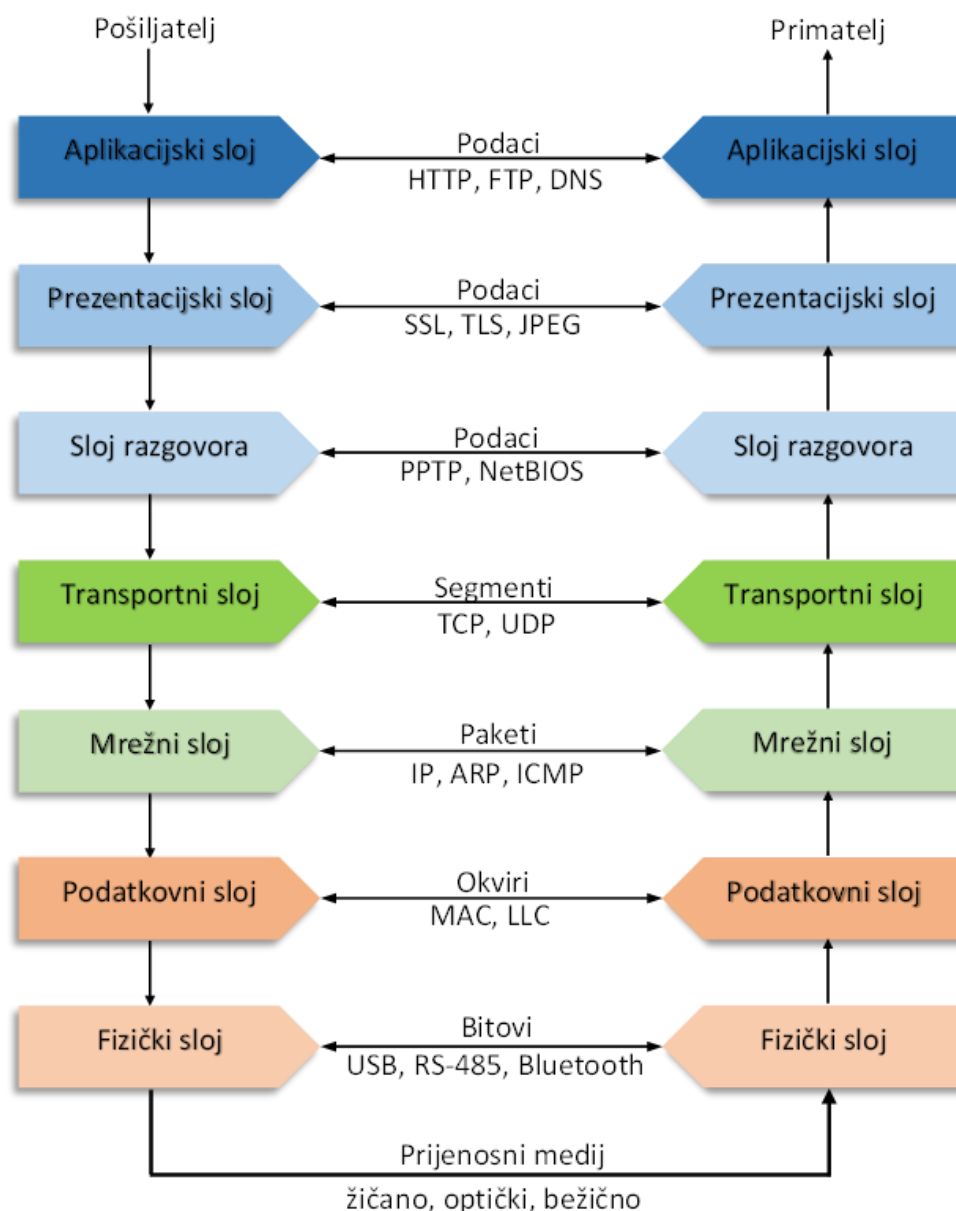
Digitalizacija i automatizacija zahvatile su sva područja ljudske djelatnosti, pa tako i sektor industrijske proizvodnje koji upravo u njima vidi način povećanja produktivnosti i kvalitete uz smanjenje troškova proizvodnje. Procesorska snaga računala eksponencijalno je porasla kroz zadnjih nekoliko desetljeća pa su i brzine računanja postale iznimno velike. Stoga je, osiguravanje cikličkog prijenosa razmjerno manje količine podataka uz minimalne latencije, postalo od iznimne važnosti. Ovdje do izražaja dolaze industrijske komunikacijske mreže čiji je zadatak osigurati determinističnu izmjenu informacija.

Prethodno razmatrani upravljački sustav LinuxCNC potrebno je na neki način povezati s hardverskim komponentama kako bi se osigurao prijenos procesnih podataka te način za parametrizaciju i konfiguriranje objekata kojima se upravlja. Ovo poglavlje za cilj ima opisati industrijsku sabirnicu EtherCAT koja će biti korištena upravo za tu svrhu u drugom dijelu rada.

4.1. OSI model

OSI (*Open Systems Interconnection*) je referentni model kojeg je razvila organizacija ISO, a na konceptualnoj razini daje razvojni okvir i skup preporuka koje bi se trebale uzeti u obzir prilikom razvoja protokola za komunikacijske sustave. Glavna motivacija prilikom razvoja ovog standarda bila je omogućiti međusobno povezivanje računalnih sustava i opreme različitih proizvođača, a s namjerom stvaranja heterogenih informatičkih mreža [8].

Referentni model ima slojevitiju hijerarhijsku strukturu, a svaki sloj sastoji se od više protokola koji obavljaju slične funkcije. Slojevi su redom (od više razine prema nižoj): aplikacijski, prezentacijski, sloj razgovora, transportni, mrežni, podatkovni i fizički. Svaki sloj u istom mrežnom čvoru prima i šalje informacije samo prvom sloju iznad i ispod sebe. Slojevi iste razine između dva mrežna čvora izmjenjuju informacije na unaprijed propisan način koji je određen protokolom. Da bi informacija došla od jednog čvora u mreži (pošiljatelja) do drugog (primatelja), treba proći svih sedam slojeva na obje strane. Prvo se odvija proces enkapsulacije - proces u kojem podatak putuje od višeg do nižeg sloja, a pritom mu svaki sloj dodaje svoje informacije u obliku zaglavlja i završetaka. U čvoru koji prima informaciju odvija se suprotan proces, odnosno deenkapsulacija. Informacija putuje od fizičkog do aplikacijskog sloja, a putem svaki sloj primatelja preuzima informacije sloja iste razine pošiljatelja. Slojevi kroz koje informacija prolazi imaju različiti način pakiranja podataka (engl. *Protocol Data Unit*, PDU), a to mogu biti podatak, segment, paket, okvir i bit (slika 4-1).



Slika 4-1. OSI model

Fizički sloj zadužen je za fizički dio prijenosa informacije. Propisuje specifikacije kao što su tip signala, naponske razine signala, električke i mehaničke karakteristike sučelja i ostale parametre koji su potrebni za uspostavljanje veze. Također je zadužen za zadatke kao što su uspostavljanje, održavanje i prekid fizičke povezanosti.

Podatkovni sloj odgovoran je za povezivanje dva čvora na istom mrežnom segmentu i ispravljanje pogrešaka koje nastaju prilikom prijenosa informacije u fizičkom sloju. Dodaje adresu izvorišta i odredišta podacima koji se šalju.

Mrežni sloj određuje putanju kretanja paketa kroz mrežu od izvora do odredišta. Često se paketi ne prenose direktno do primatelja već putuju kroz različite dijelove mreže prije nego što stignu na konačno odredište. Ovo se postiže usmjeravanjem paketa korištenjem posebnih algoritama. Ovaj sloj još ima ulogu uspostavljanja i prekida mrežne povezanosti.

Transportni sloj nadzire protok podataka od jednog korisnika do drugog i pridružuje logičke adrese fizičkim adresama koje koristi mrežni sloj. Koristi sposobnost mrežnog sloja da uspostavi vezu i na taj način osigurava da paketi sigurno dođu do krajnjih odredišta, što uključuje i ponovno slanje izgubljenih paketa.

Sloj razgovora prije uspostave razgovora (sesije) između dva korisnika provodi usuglašavanje protokola kojim će se on odvijati i koji će se koristiti kroz period trajanja sesije. Ovaj sloj određuje način pokretanja i prekidanja razgovora, način na koji se uspostavlja razmjena podataka i odgovoran je za identifikaciju krajnjeg korisnika (primjerice lozinkom).

Prezentacijski sloj određuje način na koji će se informacija oblikovati kako bi bila razumljiva krajnjem korisniku. Primjerice ovaj bi sloj vršio pretvorbu iz binarnog zapisa broja u neki cjelobrojni broj koji bi bio prikazan u aplikacijskom sloju.

Aplikacijski sloj predstavlja sučelje prema korisničkim aplikacijama. Njegov izgled uvelike ovisi o tome što aplikacije ili operativni sustav iznad njega trebaju, pa se oblikuje na način da najbolje odgovara njihovim zahtjevima.

Opsežniji opis OSI modela i uloge svakog sloja mogu se pronaći u [8]. Ipak, model koji se koristi pri opisu industrijskih komunikacijskih mreža uvelike je pojednostavljen, te se njegova struktura opisuje sa samo tri sloja – aplikacijskim, podatkovnim fizičkim.

4.2. EtherCAT

EtherCAT je komunikacijska sabirnica koja se temelji na industrijskoj Ethernet mreži. Razvila ga je tvrtka Beckhoff Automation 2003. godine, a nekoliko godina kasnije definiran je IEC 61158 standardom. Za promicanje i razvijanje ove tehnologije zadužena je zajednica proizvođača i korisnika okupljenih u *EtherCAT Technology Group*. Tehnologija je otvorenog tipa što znači da ju svi zainteresirani mogu implementirati i koristiti u svojim uređajima, uz uvjet da se zadrži njihova interoperabilnost s ostalim EtherCAT uređajima. Ova sabirnica preuzela je fizički sloj i izgled podatkovnih okvira od starijeg Ethernet protokola, a određene prilagodbe omogućile su joj primjenu u zahtjevnom industrijskom okruženju.

Glavne prednosti koje se mogu ostvariti primjenom EtherCAT tehnologije jesu [10]:

- **Izvanredne radne karakteristike**

- Trenutačno najbrža tehnologija bazirana na industrijskom Ethernetu.
- Omogućena je sinkronizacija do točnosti u nanosekundama.
- U odnosu na druge sabirnice opterećenje procesora je do 25-30% manje.

- **Fleksibilna topologija**

- Topologija mreže u potpunosti se prilagođava konfiguraciji određenog stroja ili procesa (dozvoljene su linijska te topologija u obliku stabla i zvijezde i njihove međusobne kombinacije s velikim brojem čvorova).
- Pojedine dijelove mreže i čvorove moguće je isključivati i prespajati čak i za vrijeme rada sustava.
- Omogućena je redundantna kabelska povezanost.

- **Jednostavnost i robusnost**

- EtherCAT ima opciju automatskog adresiranja spojenih uređaja.
- Malo opterećenje sabirnice i međusobna povezanost čvorova povećavaju otpornost na elektromagnetski šum.
- Sposobnost lokaliziranja poremećaja u mreži smanjuje vrijeme pronalaženja i uklanjanja grešaka.
- Mreža tijekom pokretanja uspoređuje planiranu i stvarnu topologiju mreže kako bi se otkrila eventualna odstupanja.

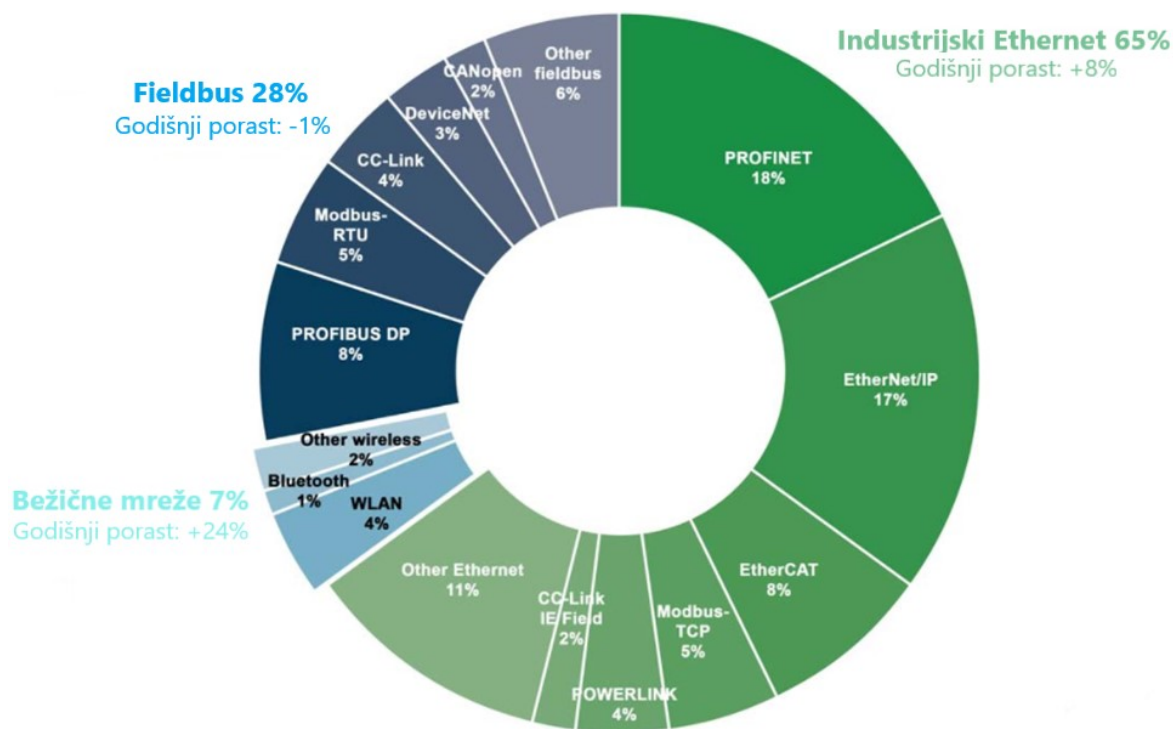
- **Integrirana sigurnost**

- Podaci vezani uz sigurnost i kontrolu procesa prenose se kroz isti prijenosni medij (bakrenih kablova, optičkih kablova ili bežično).
- Ispravnost FSoE (*Failsafe over EtherCAT*) protokola ispitana je na TÜV certificiranim uređajima.
- FSoE protokol je pogodan za upotrebu na centraliziranim i distribuiranim sustavima, a može se koristiti i na drugim komunikacijskim sustavima (npr. Ethernetu).

- **Dostupnost**

- EtherCAT pruža značajke industrijskog Etherneta po cijenama sličnim ili nižim od cijene tradicionalnih protokola zasnovanih na RS-485 sabirnici.
- Za povezivanje s računalom (nadređenim uređajem) nisu potrebne posebne komunikacijske kartice već se može koristiti većina do mrežnih kartica koje podržavaju 100BASE-TX standard.
- Upravljačke sklopove za podređene uređaje u EtherCAT mreži proizvodi velik broj proizvođača što im snižava cijenu.
- EtherCAT tehnologija ne zahtijeva upotrebu preklopnika i ostalih mrežnih elemenata, a indirektno se još ušteda ostvaruje jer nije potrebna njihova instalacija, konfiguriranje i održavanje.

Slika 4-2. prikazuje procjenu tržišne zastupljenosti pojedine industrijske mreže u 2021. godini. Procjena se temelji na broju strojeva i uređaja instaliranih i povezanih na industrijsku mrežu u 2020. godini. Udio industrijskog Etherneta bilježi kontinuirani porast, dok tradicionalni protokoli iz godine u godinu gube tržišni udio. Značajan porast zabilježile su bežične komunikacijske mreže, a očekuje se da će se njihov udio sve više povećavati s daljnjim razvojem mobilnih 5G mreža i kako se bude širio koncept industrijskog interneta stvari.



Slika 4-2. Tržišni udjeli pojedinih industrijskih mreža u 2021. godini [11]

Ethernet je prvotno bio zamišljen kao protokol za razmjenu informacija između uređaja koji ne zahtijevaju determinističku komunikaciju u stvarnom vremenu (printer, skener i ostala uredska oprema). Iz tog razloga bilo je potrebno uvesti neke promjene kako bi se Ethernet prilagodio zahtjevima koje na njega postavlja industrija, te je tako nastao industrijski Ethernet. Glavni zahtjevi bili su kratki diskretni vremenski ciklusi, čvorovi (uređaji) u mreži kojih ima puno, ali svaki izmjenjuje malu količinu informacija po ciklusu, te osiguravanje cjenovne prihvatljivosti podređenih uređaja u mreži.

4.2.1. Fizički sloj

EtherCAT sabirnica oslanja se na standardni Ethernet protokol za fizički prijenos informacija (slika 4-3.). Osnovna struktura EtherCAT mreže sastoji se od podređenih uređaja i jednog nadređenog uređaja koji njima upravlja (slika 4-4.). Za implementaciju EtherCAT protokola nisu potrebni posebni mrežni uređaji, kao kod većine ostalih protokola temeljenih na industrijskom Ethernetu. Na slici 4-3. prikazane su fizičke komponente potrebne za uspostavu komunikacije preko EtherCAT-a.

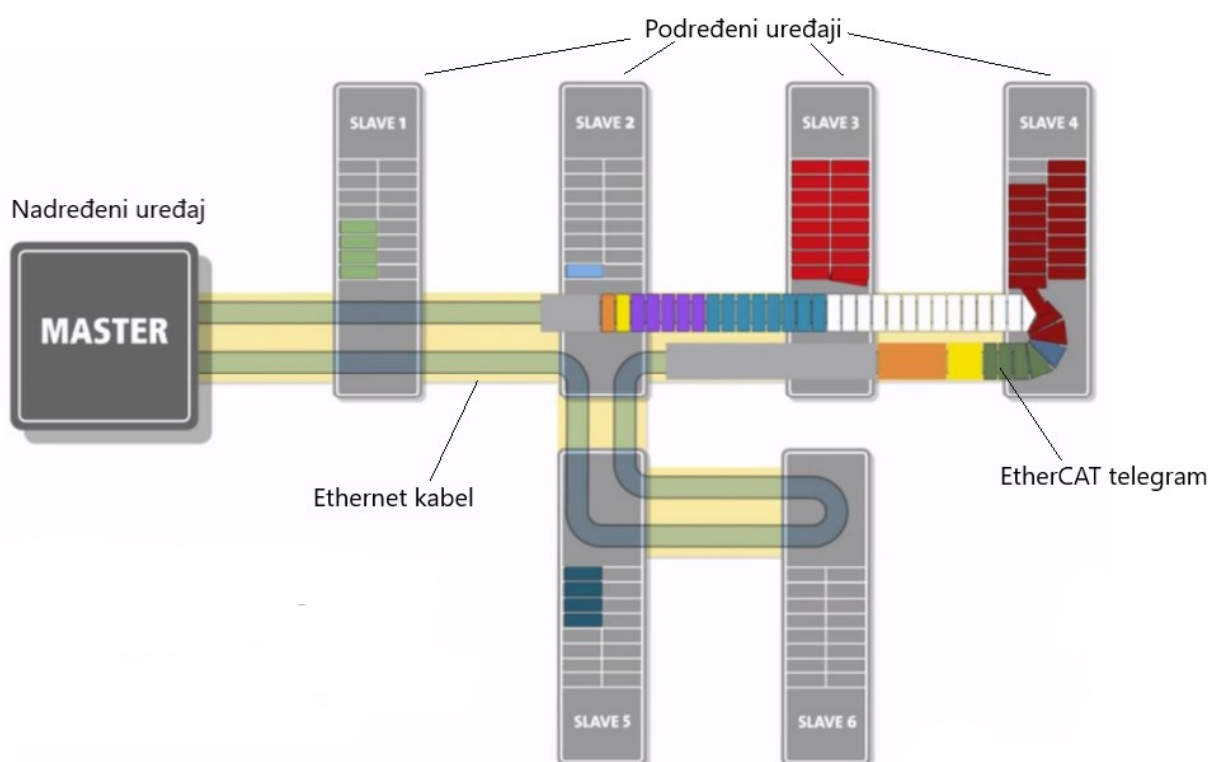


Slika 4-3. Mrežna kartica (lijevo), Cat5 kabel s RJ45 konektorom (sredina) i ESC čip (desno)

Nadređenom uređaju jedinom je dozvoljeno slati naredbe kroz mrežu. Naredba (EtherCAT telegram) prosljeđuje se kroz mrežu od čvora do čvora, te pritom sa svakim izmjenjuje informacije (slika 4-4.). Kad dođe do zadnjeg čvora u mreži, vrati se istim putem prema nadređenom uređaju, čime je završen jedan komunikacijski ciklus. Svaki podređeni uređaj u mreži posjeduje minimalno dva mrežna priključka (najviše četiri) kako bi se mogao povezati sa susjednim čvorovima, a tako povezani uređaji čine lančanu konfiguraciju (otvoreni krug). Ovo je moguće ostvariti jer se podaci prenose istovremeno u oba smjera (engl. *full-duplex*).

Povezivanjem podređenih uređaja (čvorova) mogu se postići različiti oblici topologije mreže kao što su stablo, zvijezda ili linijska topologija i njihove kombinacije. Najveći broj čvorova koji se mogu povezati iznosi 65 535, a najveća dopuštena udaljenost između dva čvora iznosi 100 metara (može se povećati primjenom optičkih kablova) [10].

Prekid kabela, elektromagnetski šum ili kvar na konektoru mogu uzrokovati obustavu komunikacije na mreži. Kako bi se to izbjeglo, EtherCAT pruža mogućnost redundantnog povezivanja. Kod ovakvog povezivanja se preko drugog Ethernet utora na nadređenom uređaju može zatvoriti mrežni krug (spoji se izlaz iz zadnjeg čvora s nadređenim uređajem). U normalnom režimu rada, podređenim uređajima se upravlja preko primarnog utora. U slučaju prekida kabela jednom granom mreže upravlja se preko primarnog, a drugom preko sekundarnog utora. Topologija prstena najpogodnija je za implementaciju ove opcije [8].



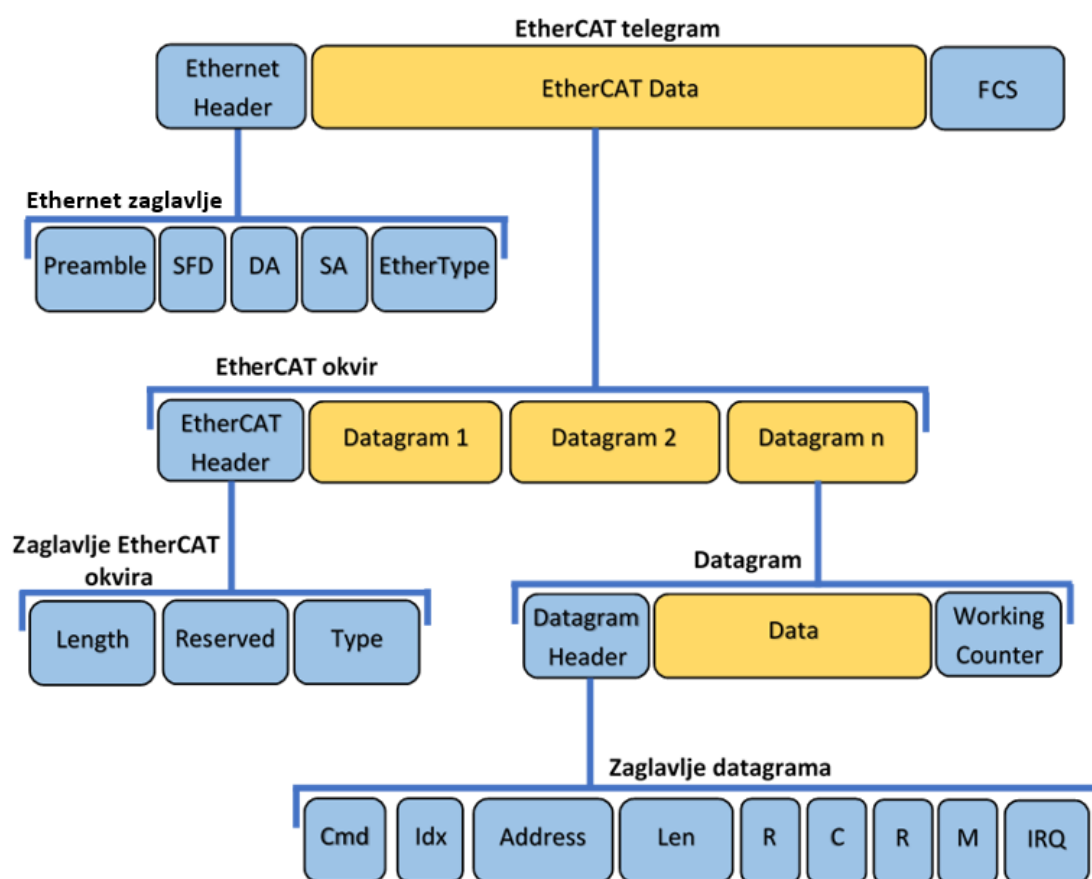
Slika 4-4. Princip rada EtherCAT sabirnice, prilagođeno iz [12]

4.2.2. Podatkovni sloj

EtherCAT tehnologija je uz fizički sloj, preuzela i izgled podatkovnog okvira od Ethernet protokola. EtherCAT okvir ugrađen je u standardni Ethernet podatkovni okvir sa Ethernet zaglavljem i segmentom FCS koji služi za provjeru integriteta podataka koji se prenose. EtherCAT telegram čine EtherCAT podaci enkapsulirani u Ethernet podatkovni okvir, a telegram se sastoji od zaglavlja i od jednog do petnaest datagrama skupa podataka (slika 4-5.). U zaglavlju datagrama zapisana je vrsta pristupa kojeg nadređeni uređaj želi ostvariti, te adresa podređenog uređaja (kod fiksnog adresiranja), odnosno podređenih uređaja (kod logičkog adresiranja) [10].

Datagram je osnovna jedinica za prijenos podataka do i od podređenih uređaja, a sadrži točno jednu instrukciju za pisanje ili čitanje podataka. Primjerice, datagram veličine dva bajta mogao bi se koristiti za čitanje analognog ulaza terminala, ali bi isto tako mogao biti i naredba za pisanje 400 bajta podataka na 10 servo regulatora [8].

Svaki datagram završava s brojačem (*Working Counter*). Brojač prati broj interakcija (pisanja i/ili čitanja) koje je određeni datagram imao s podređenim uređajima. Na temelju tog broja i broja interakcija koje su se trebale izvršiti, nadređeni uređaj može provjeriti odvija li se komunikacija ispravno [8]. Pojedine komponente podatkovnih okvira prikazane su na slici 4-5., a značenje oznaka sa slike navedeno je u tablici 4-1.



Slika 4-5. Struktura EtherCAT telegrama

Kod Ethernet tehnologije, nadređeni uređaj šalje zaseban podatkovni okvir prema svakom čvoru u mreži. To dovodi do male učinkovitosti prijenosa podataka i do smanjenja pojase propusnosti mreže. S druge strane, EtherCAT tehnologija se u velikom broju slučajeva koristi samo jednim podatkovnim okvirom (telegramom) kojim podatke dostavlja do svih čvorova u mreži. Svaki čvor obrađuje samo podatke koji su namijenjeni njemu i odmah prosljeđuje telegram do sljedećeg čvora.

Tablica 4-1. Značenje oznaka sa slike 4-5.

Ethernet podatkovni okvir			
<i>Preamble</i>	Koristi se za sinkronizaciju na razini bita između primatelja i pošiljatelja.	<i>SA</i>	MAC adresa izvorišta (pošiljatelja).
<i>SFD</i>	Označava kraj preambule i početak Ethernet okvira.	<i>EtherType</i>	Oznaka protokola (0x08A4 za EtherCAT).
<i>DA</i>	MAC adresa odredišta (primatelja).	<i>FCS</i>	Koristi se za provjeru integriteta Ethernet podatkovnog okvira.
Zaglavlje EtherCAT podatkovnog okvira			
<i>Length</i>	Duljina EtherCAT okvira.	<i>Type</i>	Tip EtherCAT protokola (npr. 0x01 za <i>CAN over EtherCAT</i>).
<i>Reserved</i>	Rezervirano polje.		
Dijelovi datagrama			
<i>Cmd</i>	Tip EtherCAT naredbe.	<i>C</i>	Bit koji indicira cirkulira li okvir kroz mrežu ili ne.
<i>Idx</i>	Oznaka kojom se nadređeni uređaj služi za identifikaciju dupliranih ili izgubljenih datagrama.	<i>M</i>	Određuje poziciju datagrama u okviru (zadnji ima vrijednost 0).
<i>Address</i>	Adresa: automatski inkrementalna, fiksna ili logička.	<i>IRQ</i>	Polje koje koristi nadređeni uređaj za prikupljanje informacija o podređenim uređajima.
<i>Len</i>	Označava duljinu podataka u datagramu.	<i>Working Counter</i>	Brojač koji se koristi za provjeru uspješnosti prijenosa podataka.
<i>R</i>	Rezervirano polje.	<i>Data</i>	Podaci za pisanje i čitanje (od 46 do 1496 bajtova).

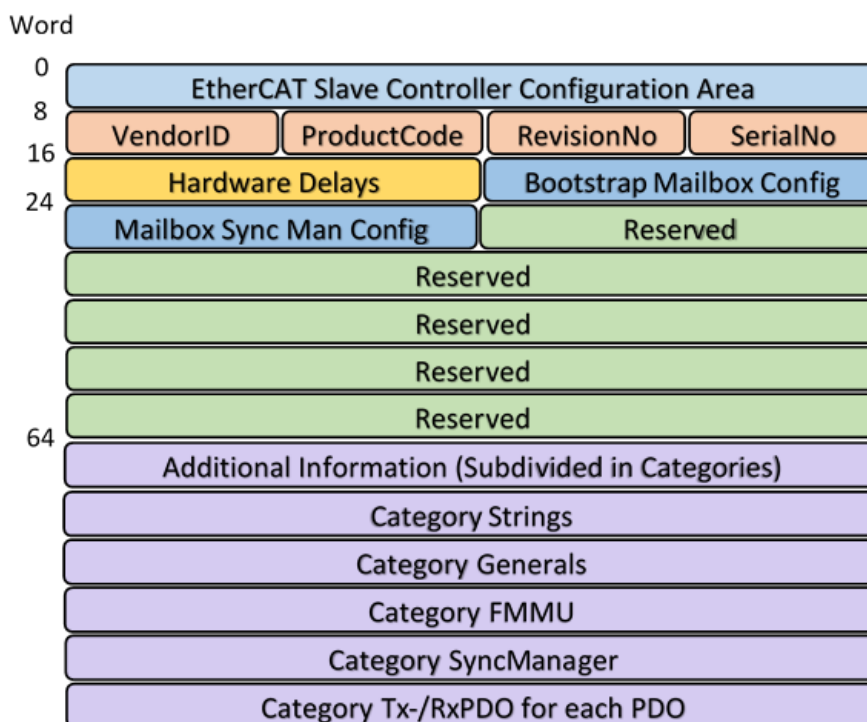
Također se, za razliku od Etherneta, podatkovni okvir ne procesuiru (deenkapsulira) softverski u nadređenom uređaju kako bi se iz njega izdvojile relevantne informacije, već se to odvija hardverski u logičkom sklopovlju (ESC čipu) podređenog uređaja. Ovom izmjenom je smanjeno opterećenje nadređenog uređaja, čija je zadaća postala samo dovoljno brzo pisanje podataka u EtherCAT okvire. Nadređeni uređaj kod EtherCAT-a moguće je implementirati softverski u računalu, a prijenos podataka se vrši preko standardne mrežne kartice. Na ovaj način postignute su velike brzine prijenosa podataka koje dosežu i do 100 Mbit/s.

Svaki EtherCAT podređeni uređaj sadrži jedan ESC čip (slika 4-3. desno). ESC (*EtherCAT Slave Controller*) je odgovoran za izvršavanje EtherCAT protokola u stvarnom vremenu i ima ulogu posrednika prilikom razmjene podataka između nadređenog uređaja i mikrokontrolera podređenog uređaja. EPU (*EtherCAT Processing Unit*) prihvaća i procesira EtherCAT okvire, a zatim podatke iz njih prosljeđuje do kontrolera ili direktno do digitalnih ulaza i izlaza podređenog uređaja. Ta komunikacija se odvija preko DPRAM (*Dual-Port RAM*) memorije kroz PDI (*Process Data Interface*) sučelje. Brzina obrade okvira neovisna je o radnim karakteristikama mikrokontrolera podređenog uređaja (ovisi jedino o ESC-u), a to znači da je brzina procesuiranja jednaka za sve čvorove u mreži [13].

PDI sučelje povezuje procesorsku jedinicu (mikrokontroler) podređenog uređaja s ESC kontrolerom, odnosno podatkovni i aplikacijski sloj EtherCAT sabirnice. Jednostavni uređaji u mreži, kao što su senzori i aktuatori, nemaju mikrokontroler pa PDI sučelje direktno povezuje ESC s digitalnim ulazima i izlazima. ESC s uređajima koji imaju procesorsku jedinicu može komunicirati serijski ili paralelno. Serijska komunikacija se vrši preko SPI (*Serial Peripheral Interface*) sučelja. Paralelna komunikacija odvija se preko paralelnog porta 8 ili 16-bitnim asinkronim ili sinkronim prijenosom podataka kroz sučelje mikrokontrolera [13].

EEPROM je električki izbrisiva programibilna memorija predviđena samo za čitanje. Naziva se još i SII (*Slave Information Interface*) sučeljem. Veličina EEPROM memorije varira u ovisnosti o tipu ESC-a, a podržane su veličine u rasponu od 1 kbit do 4 Mbit [14]. Nalazi se izvan ESC-a (slika 4-8.), a s njim je povezana I²C serijskom komunikacijskom sabirnicom. Sadrži informacije potrebne za konfiguraciju hardvera koje se učitavaju u ESC registre prilikom pokretanja podređenih uređaja. ESC registri su memorijska mjesta u koje se pohranjuju konfiguracijske postavke podređenog uređaja, a veličina im iznosi 4 kilobajta [13].

Adresiranje dijelova EEPROM memorije izvedeno je upotrebom riječi (slika 4-6.). U prva 64 adresna mjesta navedene su osnovne informacije kao što su hardverska kašnjenja, postavke distribuiranih satova (koriste se za sinkronizaciju) i konfiguracija za PDI. U ovisnosti o složenosti konfiguracije podređenog uređaja moguće je dodati opcionalne kategorije (npr. za *SyncManager*). Početni dio memorije zauzimaju podaci vezani za konfiguraciju ESC-a, a ESC ih automatski čita kod pokretanja. Nakon toga slijede podaci vezani uz ID proizvođača, šifru proizvoda, te serijski broj i broj revizije [14].



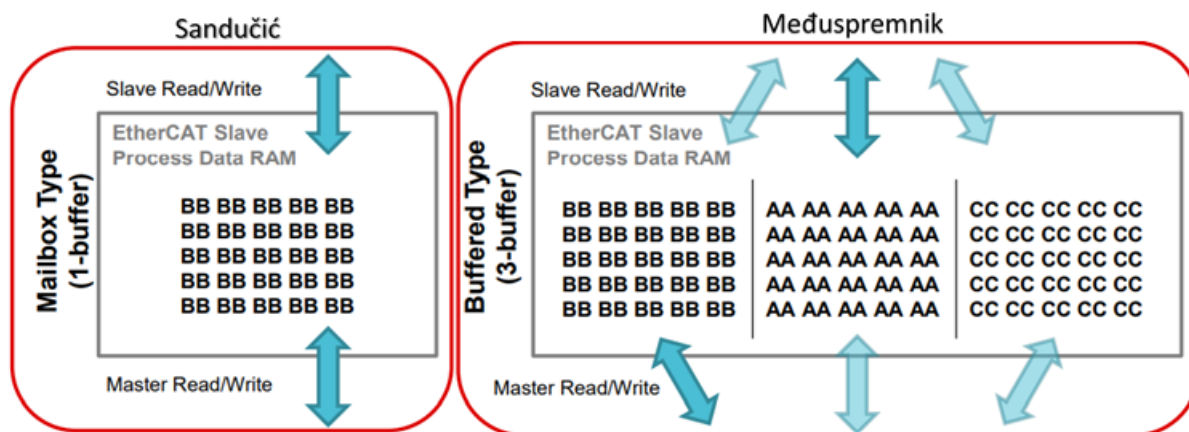
Slika 4-6. Adresna mjesta EEPROM (SII) memorije, prilagođeno iz [14]

Upravitelj sinkronizacije (*SyncManager*) ima ulogu omogućiti siguran i konzistentan pristup podacima, iz memorije ESC čipa (DPRAM-a), nadređenom EtherCAT uređaju i mikrokontroleru podređenog uređaja. Njima nije dozvoljen istovremeni pristup memoriji (prilikom izmjene podataka preko sandučića), pa je dodatni zadatak upravitelja sinkronizacije koordinacija pristupa pojedine strane memoriji. Smjer izmjene podataka i način komunikacije *SyncManager*-a konfiguriraju se pomoću nadređenog uređaja. Dva osnovna načina rada su komunikacija preko sandučića i preko međuspremnika. Najčešća konfiguracija je ona u kojoj se koriste četiri kanala *SyncManager*-a – po jedan za ulaz i izlaz iz sandučića (aciklička razmjena podataka), te za ulaz i izlaz iz međuspremnika (ciklička razmjena podataka).

U načinu rada izmjene podataka preko sandučića (*Mailbox Mode*), pristup sandučiću nadređenom uređaju i mikrokontroleru dopušten je jedino kad je druga strana završila pristup. Ovakav tip komunikacije može se opisati u nekoliko koraka. Prvo pošiljatelj piše podatke u spremnik dijeljene memorije. Zatim je pisanje u spremnik zabranjeno sve dok primatelj nije pročitao te podatke. Nakon toga je dozvola za pisanje opet omogućena pošiljatelju, a pristup spremniku je onemogućen primatelju. U ovom načinu rada nema gubitaka podataka, a koriste ga protokoli aplikacijskog sloja [14].

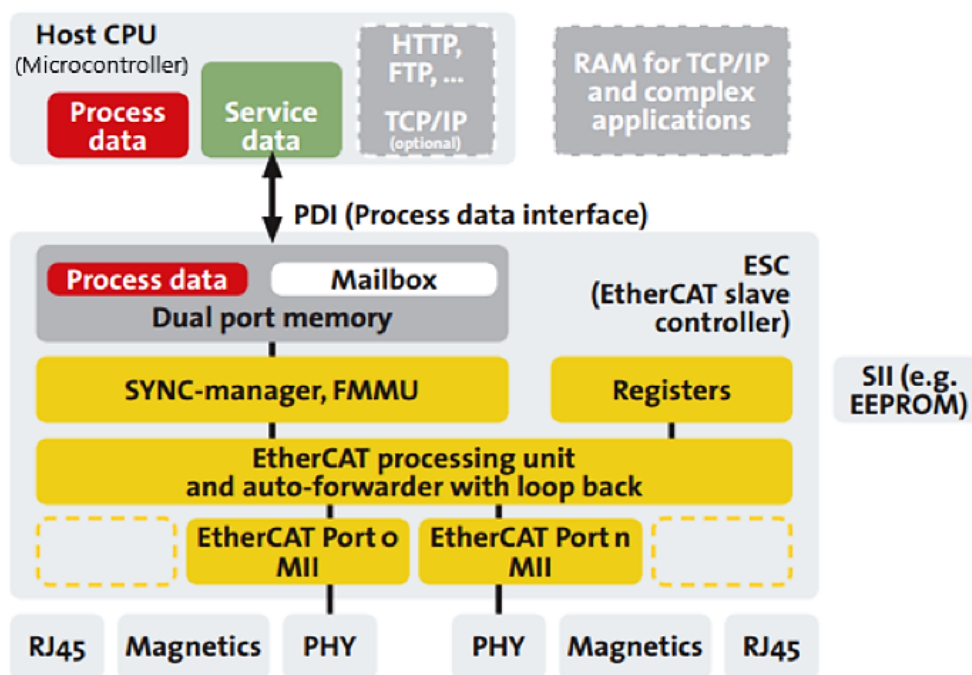
U načinu rada izmjene podataka preko međuspremnika (*Buffered Mode*), nadređenom uređaju i mikrokontroleru podređenog uređaja je u svakom trenutku dozvoljen istovremeni pristup spremniku dijeljene memorije koji je podijeljen na tri jednaka dijela (slika 4-7. desno).

Primatelj kod ovog tipa komunikacije uvijek dobiva najnovije podatke koje pošiljalatelj zapisuje u spremnik. U slučaju da se podaci pišu većom brzinom od one kojom se čitaju, stariji podaci se odbacuju i primatelj čita samo najnovije podatke. Ovaj način komunikacije koristi se za prenošenje cikličkih procesnih podataka [14].



Slika 4-7. Sinkronizirani prijenos podataka preko DPRAM memorije [15]

Slika 4-8. prikazuje arhitekturu podređenog uređaja. Prema prethodno opisanom OSI modelu, može se podijeliti na tri sloja – fizički, podatkovni i aplikacijski. Fizički sloj identičan je kao kod Ethernet protokola opisanim u IEEE 802.3 standardu, a povezan je preko MII (*Media-Independent Interface*) sučelja s podatkovnim slojem u kojem se nalaze ESC i EEPROM. Aplikacijski sloj je s podatkovnim slojem povezan preko PDI sučelja i u njemu se nalazi mikrokontroler podređenog uređaja.



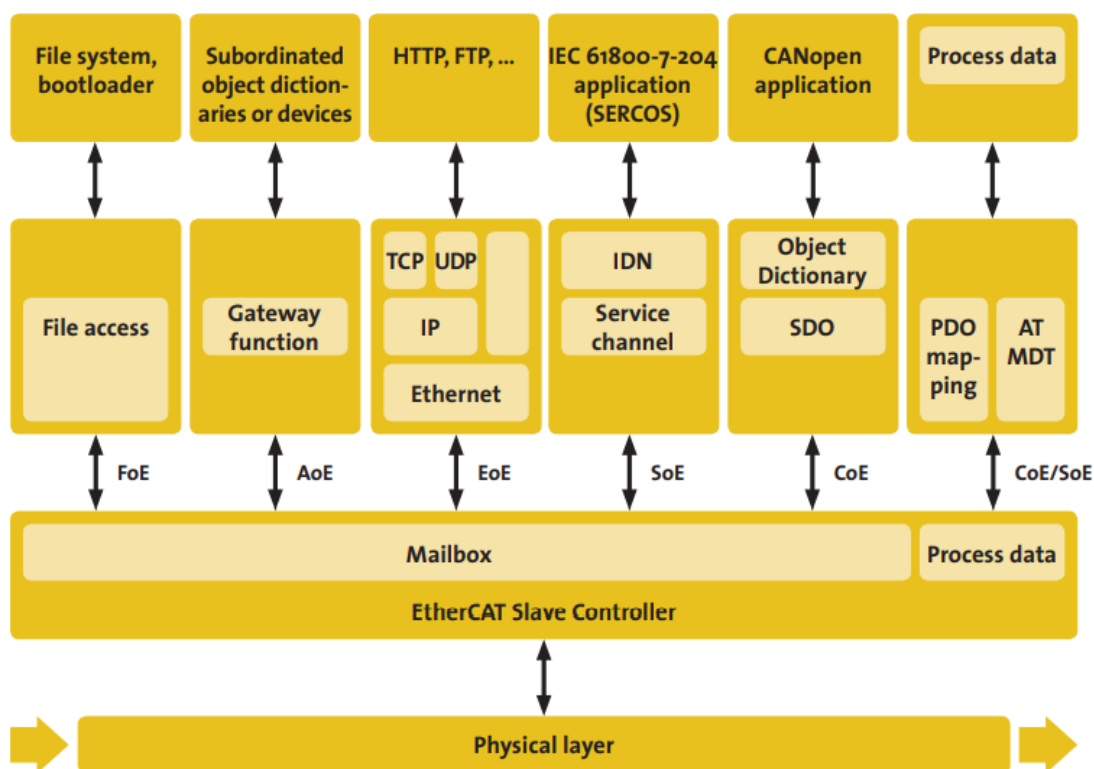
Slika 4-8. Arhitektura podređenog uređaja [10]

4.2.3. Aplikacijski sloj

Mikrokontroler (mikroprocesor) podređenog uređaja zadužen je obavljanje funkcija propisanih u protokolima aplikacijskog sloja. EtherCAT sabirnica se za komunikaciju u aplikacijskom sloju služi sljedećim protokolima:

- SERCOS over EtherCAT (SoE)
- File access over EtherCAT (FoE)
- ADS over EtherCAT (AoE)
- Ethernet over EtherCAT (EoE)
- CANOpen over EtherCAT (CoE)

Podređeni uređaj ne mora podržavati sve navedene protokole. Može koristiti samo one koji su najprikladniji za određenu primjenu, a nadređeni uređaj čitanjem opisne datoteke podređenog uređaja saznaje koji su protokoli dostupni za komunikaciju s uređajima u mreži. Većina protokola u aplikacijskom sloju temelji se na acikličkoj izmjeni podataka preko sandučića, a koriste se u različite svrhe koje su pobliže opisane u nastavku. Za razmjenu vremenski osjetljivih procesnih podataka koriste se CoE i SoE protokoli (slika 4-9).



Slika 4-9. Aplikacijski protokoli EtherCAT sabirnice [10]

SoE protokol omogućava implementaciju uređaja zasnovanih na SERCOS tehnologiji u EtherCAT mrežu. Ovako je stariji SERCOS protokol iskoristio prednosti novijeg EtherCAT protokola kao što su fleksibilnija topologija, brži prijenos podataka i manja kašnjenja u mreži. SERCOS osigurava prijenos dvaju tipova podataka – cikličkih i acikličkih (slika 4-9.). Ciklički podaci izmjenjuju se sinkronizirano u preciznim vremenskim intervalima između upravljačke jedinice i podređenih uređaja (regulatora, I/O modula). Naredbe nadređeni uređaj šalje preko MDT telegrama, a procesne podatke iz podređenih uređaja prima preko AT telegrama (oba telegrama šalju se iz nadređenog uređaja). Za mapiranje procesnih podataka, koriste se identifikacijski brojevi (IDN) koji su pridruženi svakom parametru uređaja. Aciklički prijenos informacija vrši se asinkrono. Unutar telegrama postoji rezervirano mjesto (*Service Channel*) za acikličke podatke. Kroz njega nadređeni uređaj može slati samo dva bajta acikličkih (servisnih) podataka prema podređenim uređajima po jednom ciklusu, što znači da se moraju slati segmentirano [16].

FoE protokol se u EtherCAT mreži upotrebljava za jednostavan pristup datotekama. Baziran je na TFTP protokolu, a koristi se za nadogradnju ugrađenih programa (engl. *firmware*) mikrokontrolera podređeni uređaja. Tijekom nadogradnje programa uređaja jedino je ovaj vid komunikacije dozvoljen. FoE protokolom omogućeno je standardno preuzimanje softvera i brz prijenos datoteka, čak i bez podrške TCP/IP protokola [13].

AoE protokol omogućava komunikaciju između klijenta i poslužitelja preko sandučića. Ne pruža detaljni semantički opis primjene kao CoE ili SoE protokoli, već se koristi kao njihova nadopuna u situacijama koje zahtijevaju usluge usmjeravanja i paralelnog prijenosa podataka. Ima ulogu mrežnog pristupnika (engl. *Gateway function*) pomoću kojeg je moguće povezati dvije različite komunikacijske mreže. Koristi se još i za acikličku komunikaciju putem EAP protokola, te za prikupljanje dijagnostičkih mrežnih informacija iz udaljenih dijagnostičkih alata [10].

EoE protokol koristi se za prijenos Ethernet podatkovnog prometa preko EtherCAT mreže. Tehnika slanja podataka iz jedne mreže preko druge naziva se tuneliranje. Ethernet uređaji se na EtherCAT mrežu povezuju preko mrežnih preklopnika, a Ethernet okviri se tuneliraju kroz EtherCAT protokol slično kao kod TCP/IP protokola što čini Ethernet uređaje u potpunosti kompatibilnima s EtherCAT mrežom. Mrežni preklopnik šalje okvire korištenjem EoE protokola prema odgovarajućim čvorovima u mreži na temelju njihove MAC adrese, te vrši umetanje TCP/IP segmenata u EtherCAT promet kako se ne bi narušio tijek prijenosa podataka u stvarnom vremenu [10].

CoE protokolom je u EtherCAT mrežu dodana mogućnost implementacije uređaja koji u višim slojevima koriste CANOpen protokol. Ovim protokolom predviđena je razmjena cikličkih i acikličkih podataka, te je najčešće korišten protokol za te svrhe kod EtherCAT sabirnice. Sastoji se od nekoliko osnovnih elemenata kao što su rječnik objekata, te PDO i SDO poruka. U nastavku su ukratko objašnjeni zajedno s postupkom mapiranja procesnih podataka. Detaljan opis CANOpen protokola može se pronaći u standardu EN 50325-4.

Unutarnja stanja regulatora motora (npr. najveća dopuštena jakost struje) opisana su na jednostavan i standardiziran način pomoću parametara. Svi parametri navedeni su u rječniku objekata, a svaki uređaj u CAN mreži posjeduje vlastiti rječnik objekata. Kako bi se moglo pristupiti pojedinom parametru iz nekog rječnika objekata, svakom parametru dodijeljen je jedinstveni identifikacijski broj (indeks i pod-indeks, slika 4-10.). Pristup parametrima u rječnicima objekata omogućen je na dva načina. Prvi način je slanjem SDO poruka gdje regulator motora nadređenom uređaju mora potvrditi pristup pojedinom parametru. Drugi način je preko PDO poruka gdje odobrenje nije potrebno, već se promjena procesnih veličina u rječniku objekata unosi automatski. Kod CoE protokola, prijenosom PDO i SDO poruka upravlja prethodno spomenuti *SyncManager* (pomoću svojih kanala) preko sandučića ili preko međusprenika [17].

SDO (*Service Data Object*) poruke predstavljaju jednostavniji vid komunikacije od dva prethodno spomenuta. Poruke se izmjenjuju preko sandučića, a koriste se za početno postavljanje parametara uređaja i za uspostavu složenije komunikacije preko PDO poruka. Smjer slanja naredbi je uvijek iz više upravljačke jedinice (klijenta) prema nižoj razini (poslužiteljima). Ta naredba može biti ili promjena vrijednosti nekog parametra u rječniku objekata (pisanje) ili očitavanje vrijednosti parametra. Za svaku poslanu naredbu upravljačka jedinica zaprima odgovor koji sadrži ili očitane vrijednosti parametra ili potvrdu promjene parametra. U slučaju da dođe do neispravnog čitanja ili pisanja parametara, podređeni uređaj umjesto potvrde šalje obavijest o pogrešci [17].

PDO (*Process Data Object*) poruke omogućavaju brz prijenos procesnih podataka. Podaci se ili šalju ciklički ili je njihovo slanje uvjetovano nekim vanjskim događajem (npr. očitanjem s graničnih prekidača). Komunikacija se odvija po principu proizvođača i potrošača. Proizvođač odašilje podatke kroz mrežu, a ostali čvorovi u mreži (potrošači) koriste identifikatore kako bi prepoznali jesu li podaci relevantni za njih. PDO poruke sastoje se od segmenata maksimalne veličine 8 bajta. Gledano iz perspektive mrežnog uređaja, dijele se na PDO poruke za primanje (RxPDO) i odašiljanje (TxPDO) podataka [18].

PDO mapiranje postupak je kod kojeg se pojedini objekti iz rječnika objekata zajedno slažu u PDO poruke kako bi se ubrzalo njihovo slanje kroz mrežu. Rječnik objekata sadrži procesne parametre (označeni indeksom 6xxx na slici 4-10.) i parametre mapiranja (indeks 1xxx). Većina mjesta u rječniku objekata je predodređena CANOpen standardom, ali postoje i ona koja korisnik (proizvođač) može konfigurirati prema svojim potrebama. Parametri mapiranja određuju koji procesni podaci će se mapirati u PDO poruke. Na slici 4-10. prikazan je primjer postupka mapiranja. Vrijednosti dvaju analognih i digitalnih ulaza ukupne veličine 5 bajta mapiraju se u jednu PDO poruku (TPDO1) kojom se šalju dalje kroz mrežu. Maksimalna veličina PDO poruke je 8 bajta što znači da su 3 bajta ostala neiskorištena.

Index	SubIdx	Type	Description
1A00h			1st Transmit PDO - Mapping
	0	UNSIGNED8	= 4 (Number of supported map entries)
	1	UNSIGNED32	= 6000 01 08h (Idx 6000, SubIdx 1, 8 bit)
	2	UNSIGNED32	= 6000 02 08h (Idx 6000, SubIdx 2, 8 bit)
	3	UNSIGNED32	= 6401 01 0Ch (Idx 6401, SubIdx 1, 12 bit)
6000h			Process data, digital inputs
	0	UNSIGNED8	= 2 (Number of sub-index entries)
	1	UNSIGNED8	8-bit digital input
6401h			Process data, analog inputs
	0	UNSIGNED8	= 2 (Number of sub-index entries)
	1	UNSIGNED16	12-bit analog input
	2	UNSIGNED16	12-bit analog input

TPDO1	D In 1	D In 2	A In 1	A In 2	Unused		
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Slika 4-10. Primjer PDO mapiranja [19]

CANOpen protokol predviđa nekoliko načina rada regulatora motora. Promjenom predodređenog parametra u rječniku objekata može ga se postaviti u sljedeće načine rada [17]:

- početno referenciranje (*Homing Mode*)
- upravljanje po poziciji (*Profile Position Mode*)
- upravljanje po brzini (*Profile Velocity Mode*)
- upravljanje po momentu (*Profile Torque Mode*)
- interpolirano upravljanje pozicijom (*Interpolated Position Mode*)

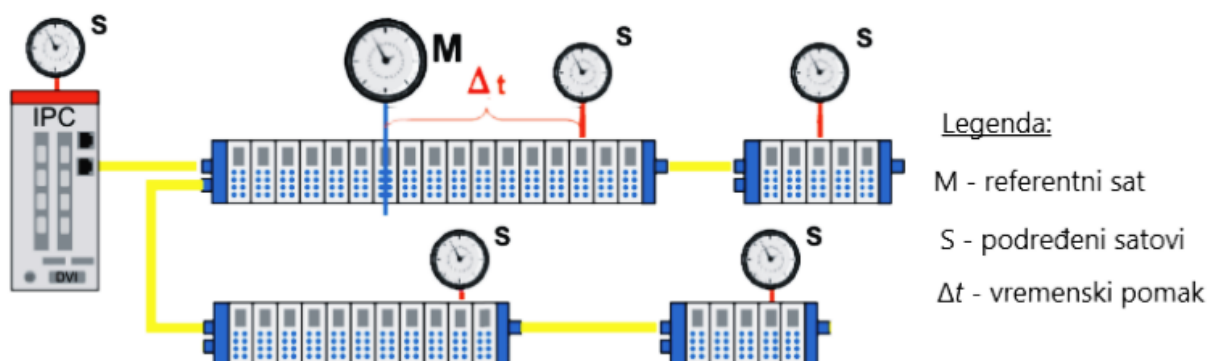
4.2.4. Distribuirani satovi

Komunikacija preko EtherCAT sabirnice odvija se deterministički u stvarnom vremenu. Ipak, EtherCAT okviru je potrebno određeno vrijeme da dođe do pojedinih čvorova, a ono varira u ovisnosti o udaljenosti i poziciji čvora u mreži. Kako bi se osiguralo isto sistemsko vrijeme između svih uređaja u mreži, uvedena je metoda distribuiranih satova koja je implementirana u ESC kontrolere podređenih uređaja. Nužnost vremenske sinkronizacije može se pokazati na primjeru 6-osnog CNC stroja gdje svaki regulator posmične osi predstavlja jedan podređeni uređaj u mreži. Kako bi se osigurala optimalne radne karakteristike stroja i ostvarila željena kvaliteta obrade, potrebno je da sve osi istovremeno izvršavaju pristigle naredbe.

Svaki podređeni uređaj (koji podržava primjenu distribuiranih satova) posjeduje vlastiti sat koji, nakon pokretanja uređaja, mjeri njegovo lokalno vrijeme. Zatim se u EtherCAT mreži odabere jedan čvor čije se vrijeme uzme kao referentno, a prema njemu se podešavaju vremena ostalih uređaja u mreži. Obično se sat prvog čvora u mreži uzima kao referentni, te tada njegovo vrijeme predstavlja vrijeme sustava. Podešavanje satova preostalih podređenih uređaja obavlja se automatski. Nadređeni uređaj šalje datagrame u koje podređeni uređaji po primitku unose trenutno vrijeme sa svojih satova (slika 4-11.). Na temelju ovih informacija nadređeni uređaj zasebno računa potrebni vremenski pomak za svaki podređeni sat u mreži [20].

Metoda distribuiranih satova osigurava sinkronizaciju uređaja u EtherCAT mreži do točnosti ispod jedne mikrosekunde. Dodavanjem ove funkcionalnosti moguće je ostvariti [14]:

- Sinkronizaciju satova podređenih i nadređenog uređaja
- Generiranje sinkronih izlaznih signala (*SyncSignals*)
- Precizno vremensko datiranje ulaznih događaja (*LatchSignals*)
- Sinkrono ažuriranje digitalnih izlaza
- Sinkrono uzorkovanje digitalnih ulaza



Slika 4-11. Princip rada distribuiranih satova, prilagođeno iz [15]

4.2.5. Adresiranje

Svaki podređeni uređaj može se adresirati zasebno tako da mu se dodjeli vlastita adresa (*Device Addressing*) ili se više uređaja može skupno adresirati na temelju slike procesa korištenjem logičkog adresiranja (*Logical Addressing*). Slika procesa je kopija stanja procesa u nekom vremenskom trenutku, a spremljena je u nadređenom uređaju. Postoje tri načina provedbe adresiranja kod prvog tipa adresiranja, a to su automatski-inkrementalno (pozicijsko), fiksno (čvorišno) i *broadcast* adresiranje.

Kod automatski-inkrementalnog adresiranja nadređeni uređaj pristupa podređenom na temelju njegove pozicije u mreži. Datagram, koji se šalje prema podređenom uređaju, u adresnom polju ima negativnu vrijednost koja se inkrementalno povećava od jednog do drugog čvora u mreži (na putu prema adresiranom čvoru). Čvor koji po primitku datagrama očitava vrijednost adrese nula izvršava naredbu poslanu tim datagramom. Ovaj način adresiranja koristi se samo prilikom pokretanja EtherCAT sustava kako bi se odredila konfiguracija mreže i kako bi se otkrili naknadno dodani čvorovi u mrežu [14].

Kod fiksnog adresiranja svakom čvoru u mreži dodijeljena je zasebna adresa preko koje nadređeni uređaj može slati poruke pojedinačnim podređenim uređajima. Ovo je moguće postići na dva načina. U prvom slučaju, nadređeni uređaj prilikom pokretanja EtherCAT sustava dodjeljuje adrese (*Station Address*) podređenim uređajima koje podređeni uređaj ne može promijeniti. Alternativno se može koristiti adresa (*Configured Station Alias*) koju svaki podređeni uređaj ima spremljenu u EEPROM memoriji, a pri pokretanju sustava šalje se prema nadređenom uređaju. Ova adresa se po potrebi može mijenjati. Ukoliko se adresa poslanog datagrama i jedne od dvije prethodno navedene adrese poklapaju, podređeni uređaj izvršava naredbu zapisanu u datagramu [14].

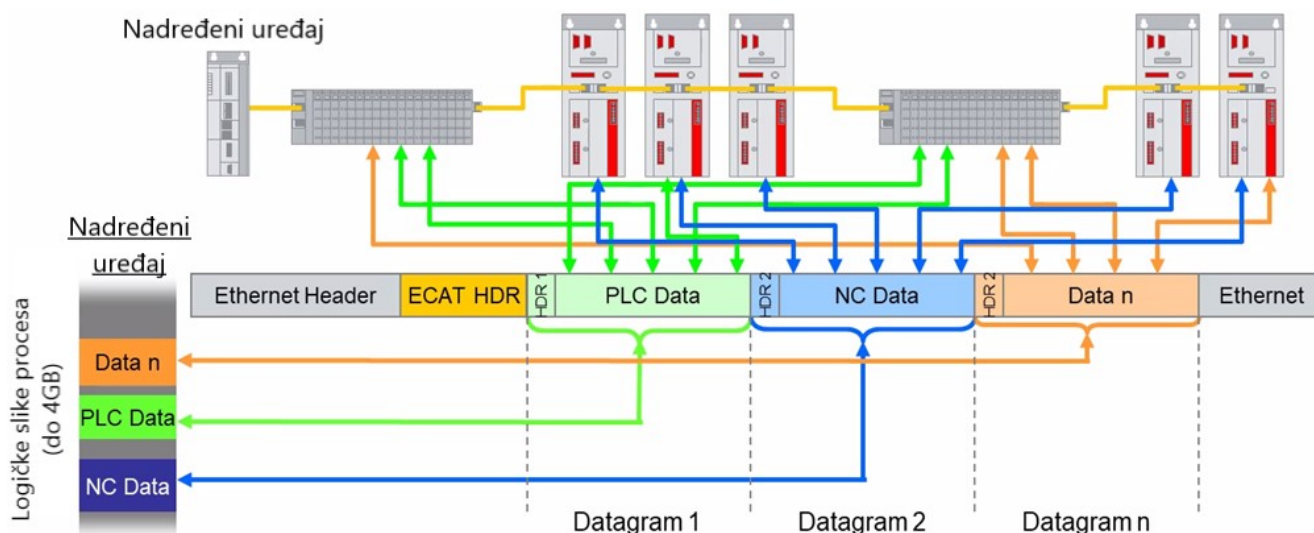
Broadcast adresiranje koristi se kod inicijalizacije podređenih uređaja i za provjeru statusa svih uređaja u mreži kada se očekuje da je on identičan. Također se koristi za adresiranje svih podređenih uređaja u mreži [14].

Logičko adresiranje koristi se kod cikličke razmjene procesnih podataka. Svaki datagram adresira određeni dio slike procesa u EtherCAT segmentu, za koju je dostupno 4 GB adresnog prostora. Prilikom inicijalizacije sustava svakom se uređaju dodijeli jedna ili više adresa. Uređajima, kojima su dodijeljene adrese u istom području, može se pristupiti s istim datagramom. Datagram sadrži sve informacije vezane za pristup podacima iz memorije podređenih uređaja, pa nadređeni uređaj može odlučiti kada i kojim podacima pristupiti [10].

Primjerice, nadređeni uređaj može koristiti kratke cikluse za ažuriranje podataka koje šalje prema regulatorima motora, a dulje vremenske cikluse za uzorkovanje ulazno-izlaznih terminala. Također je dozvoljeno i acikličko slanje podataka. Ovo dovodi do rasterećenja upravljačke jedinice u odnosu na konvencionalne sabirnice gdje se podaci sa svakog čvora čitaju, procesiraju i kopiraju u memoriju zasebno [10].

FMMU (*Fieldbus Memory Management Unit*) upravljačka jedinica sastavni je dio ESC čipa, a uloga joj je povezivanje logičke adrese s fizičkim adresama EtherCAT uređaja. Ovaj postupak naziva se još i mapiranje. Nadređeni uređaj koristi se za početno konfiguriranje FMMU-a, a informacije koje je potrebno postaviti su početna logička adresa, početna adresa fizičke memorije, duljina podataka i tip kojim se određuje smjer mapiranja (pisanje, čitanje ili oboje). Također, nadređeni uređaj na temelju slike procesa podešavanjem FMMU-a određuje koji podređeni uređaji mogu mapirati podatke u iste datagrame. U memoriji svakog podređenog uređaja nalaze se procesni podaci. FMMU dovodi u odnos logičku memoriju s fizičkom memorijom korištenjem alokacijske tablice. Nadređeni uređaj koristi konfiguracijske informacije FMMU-a kako bi prepoznao koje dijelove logičke slike procesa treba mapirati s kojim lokalnim adresama uređaja.

Slika 4-12. prikazuje način prijenosa podataka kod logičkog adresiranja. Podaci koji se primaju i šalju prema podređenim uređajima logički su grupirani i čine dijelove slike procesa u nadređenom uređaju. Tako se primjerice PLC ili NC podaci mogu primiti i slati prema više podređenih uređaja pomoću jednog datagrama. Pojedini podređeni uređaji prepoznaju koje podatke trebaju razmijeniti s kojim datagramom korištenjem FMMU-a koji su unaprijed konfigurirani od strane nadređenog uređaja.



Slika 4-12. Logičko adresiranje [15]

4.2.6. Automat stanja podređenog uređaja

Podređeni uređaj u EtherCAT mreži može biti u nekoliko unaprijed određenih stanja kojima upravlja nadređeni uređaj. U ovisnosti o položaju u automatu stanja, variraju i funkcije koje podređeni uređaj može izvršavati. Postoje specifične naredbe (kontrolne riječi) koje nadređeni uređaj šalje podređenima u pojedinom stanju, a od podređenog uređaja se očekuje povratna informacija o potvrdi statusa nakon prelaska iz jednog u drugo stanje [9].

Stanja kroz koja prolazi podređeni uređaj tijekom početnog postavljanja navedena su u tablici 4-2. Dodatno stanje, u kojem se uređaj nalazi prilikom nadogradnje softvera, je stanje pokretanja (*Boot*). Njemu se može pristupiti jedino iz *Init* stanja, a komunikacija je u njemu dozvoljena jedino preko sandučića korištenjem FoE protokola [9].

Tablica 4-2. Osnovna stanja podređenih uređaja

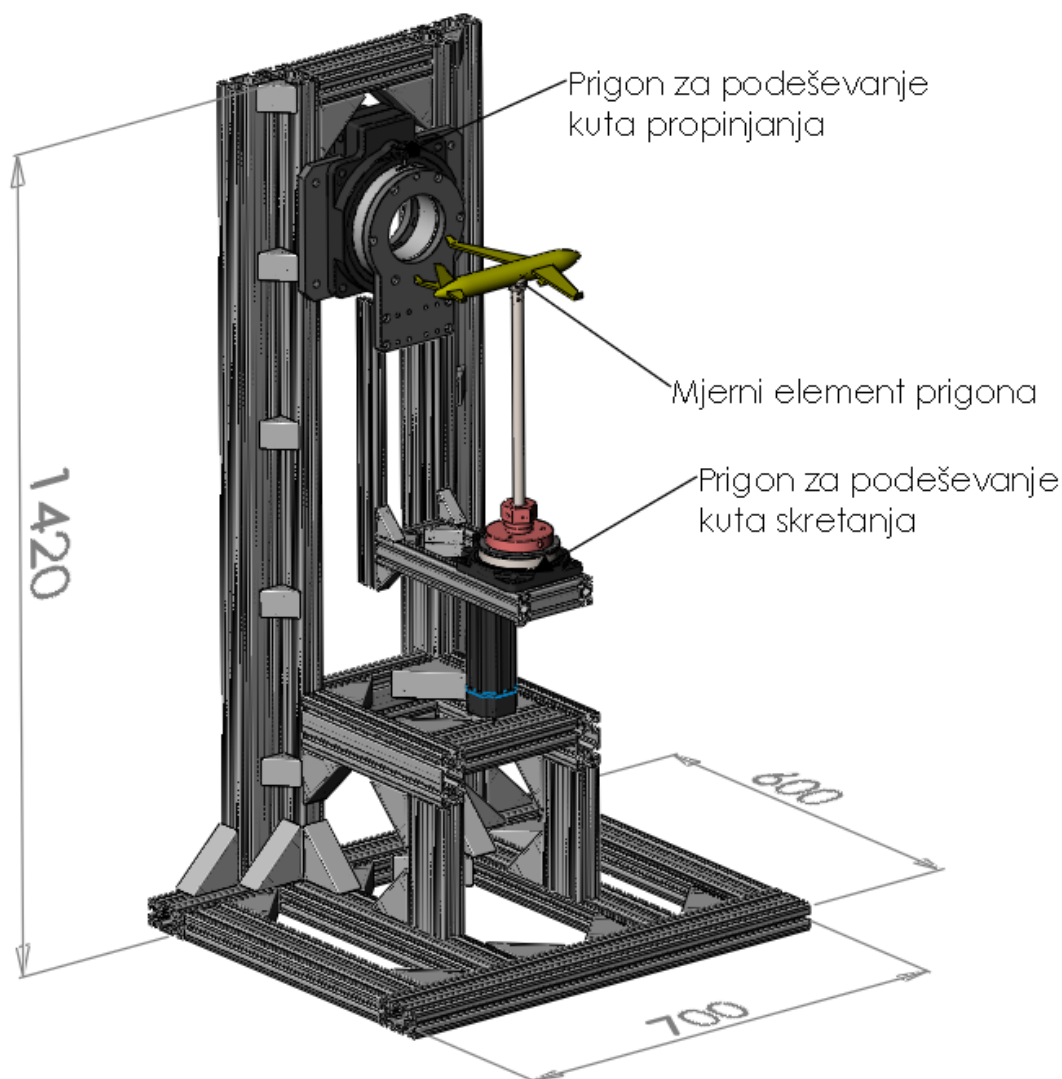
Stanje	SDO komunikacija	PDO prijenos	PDO prijem	Napomena
Inicijalizacija (<i>Init</i>)	✗	✗	✗	Stanje u kojem se podređeni uređaj nalazi nakon uključivanja. Komunikacija nije moguća.
Pred-operativno (<i>Pre-operational</i>)	✓	✗	✗	Komunikacija preko sandučića (SDO) je omogućena. Koristi se za početno postavljanje mrežnih postavki i PDO mapiranje.
Sigurno-operativno (<i>Safe-operational</i>)	✓	✓	✗	Omogućeno je zaprimanje PDO poruka i komunikacija preko sandučića. Statusni i ostali podaci mogu se ciklički slati sa podređenog uređaja.
Operativno (<i>Operational</i>)	✓	✓	✓	Komunikacija se odvija normalno. Omogućeno je upravljanje podređenim uređajima cikličkim slanjem poruka.

5. KONSTRUKCIJSKO RJEŠENJE, POGONSKI I MJERNI ELEMENTI PRIGONA

U ovom poglavlju je predstavljeno i realizirano idejno konstrukcijsko rješenje prigona uređaja za pozicioniranje modela letjelice u zračnom tunelu. U nastavku su navedeni i opisani pogonski elementi i mjerni element prigona.

5.1. Konstrukcijsko rješenje uređaja

Predloženo rješenje (slika 5-1) predstavlja poboljšanje u odnosu na prethodno rješenje predloženo u [21]. Poboljšanje se očituje u povećanoj krutosti konstrukcije i u tome što je izrađena od standardnih aluminijskih profila čime je smanjena potreba za strojnom obradom. U novom prijedlogu strojna obada se koristi samo za izradu jedne prihvatne ploče i dviju prirubnica.

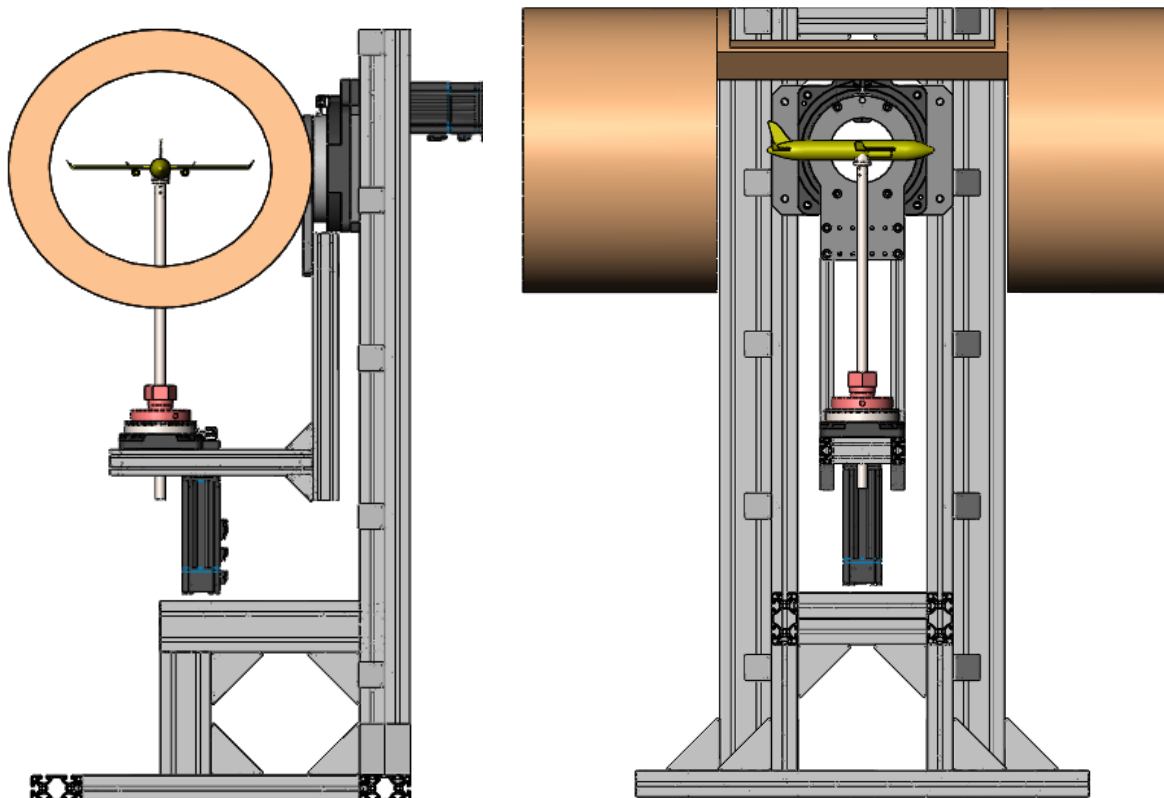


Slika 5-1. Konstrukcija uređaja

Slika 5-1. prikazuje dimenzije uređaja. Uređaj se sastoji od dva posmična rotacijska prigona. Prvi prigon koristi se za podešavanje kuta propinjanja, dok se drugi prigon koristi za podešavanje kuta skretanja. Model letjelice spojen je preko potpornja i stezne čahure na priрубnicu reduktora manjeg prigona. Na vrhu potpornja (u unutrašnjosti modela letjelice) nalazi se 6-osna mjerna ćelija za mjerenje aerodinamičkih sila i momenata, a detaljno je opisana u [21].

Uređaj će se montirati u sklopu postojećeg zračnog tunela. Prilikom montaže potrebno je osigurati precizno pozicioniranje modela letjelice u središte mjerne sekcije tunela. Iz tog razloga predviđena je mogućnost pomicanja prigona (reduktora) kuta skretanja po horizontalnim aluminijskim profilima za koje je pričvršćen. Visina modela letjelice u mjestu mjernoj sekciji jednostavno se prilagođava uvlačenjem i izvlačenjem potpornja iz stezne čahure.

Slika 5-2. prikazuje nacrt i bokocrt uređaja i zračnog tunela. Osim letjelice, bit će moguće ispitivati i aerodinamične karakteristike pojedinačnih modela krila (aeroprofila) tako da se direktno pričvrste na priрубnicu reduktora prigona za podešavanje kuta propinjanja. Ovaj zahvat nadilazi opseg ovog rada i njegov potencijalni razvoj bit će predmet budućeg rada.



Slika 5-2. Prigon, model letjelice i zračni tunel

Slika 5-3. prikazuje nosivu konstrukciju prigona koja se sastoji od aluminijskih profila spojenih kutnicima. Iako je napravljena priprema (CAM program) za izradu prirubnica potrebnih za pričvršćivanje pogonskih elemenata na konstrukciju, iste se nisu stigle izraditi do završetka pisanja ovoga rada. To ne predstavlja prepreku jer se pogonski elementi prigona uvijek u početnoj fazi testiraju samostalno, odvojeno od ostatka konstrukcije.

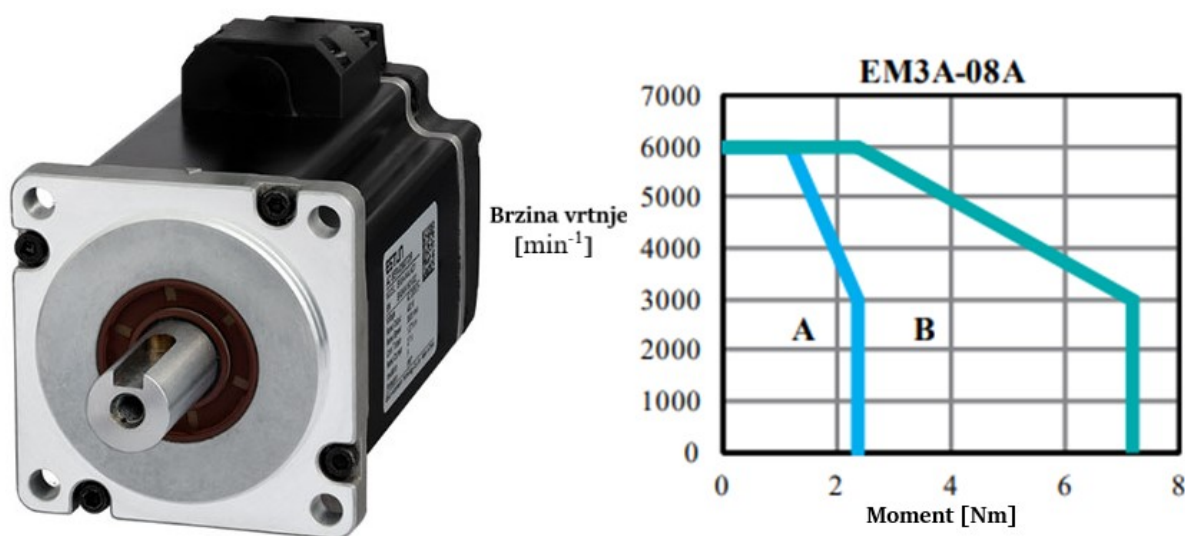


Slika 5-3. Nosiva konstrukcija prigona

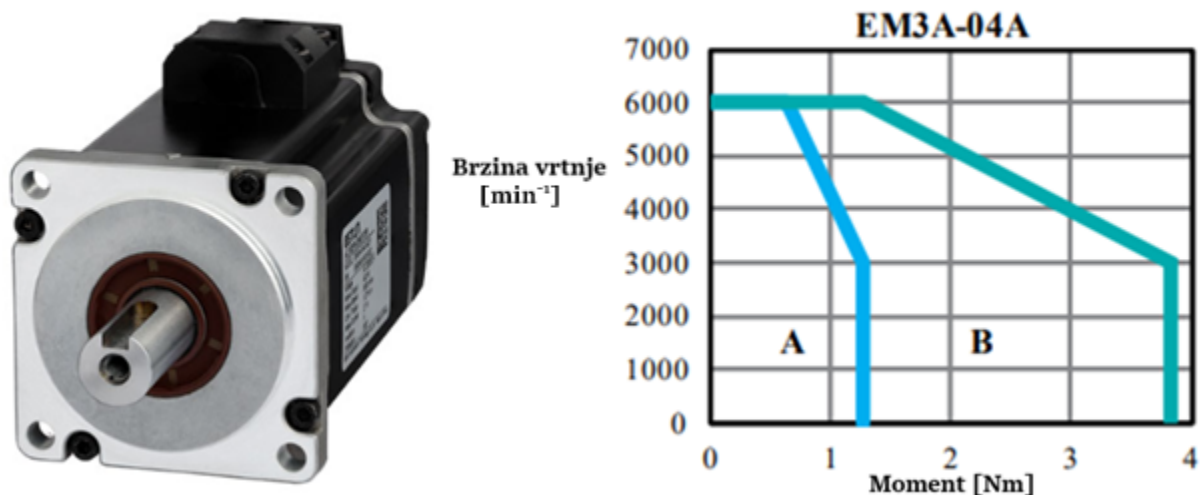
5.2. Pogonski elementi prigona

Pogonski elementi prigona odabrani su prema najvećim očekivanim radnim opterećenjima i dinamičkim svojstvima koja se od uređaja zahtijevaju. Rotacijske osi pogonjene su sinkronim motorima s permanentnim magnetima (SMPM) proizvođača ESTUN. Korištenje servomotora također predstavlja nadogradnju u odnosu na koračne motore korištene u prethodnom idejnom rješenju (prezentiranom u [21]). Oba motora (slike 5-4. i 5-5.) opremljena su apsolutnim enkoderom, mehaničkom kočnicom, a na izlaznom vratilu imaju utor za pero. Tehnički podaci su im navedeni u tablici 5-1.

Rotacijski prigon koji regulira kut propinjanja (prva os rotacije) ima veću izlaznu snagu i nazivni moment jer uz model letjelice zakreće i cijeli prigon kuta skretanja (druga os), što je vidljivo i na slici 5-1. Karakteristike označene slovom A predstavljaju nazivno opterećenje u normalnom režimu rada, a one označene slovom B predstavljaju maksimalno dozvoljeno vršno opterećenje. Kako je već ranije spomenuto, jedna od prednosti servomotora je ta da može držati konstantnu frekvenciju vrtnje kod različitog opterećenja na vratilu što se jasno vidi na priloženim karakteristikama.



Slika 5-4. Servomotor prve osi s pripadajućom karakteristikom [22]



Slika 5-5. Servomotor druge osi i pripadajuća karakteristika [22]

Tablica 5-1. Tehničke specifikacije odabranih elektromotora

		Oznaka motora	
	Jedinica	EM3A-04ALA241	EM3A-08ALA241
Nazivni napon napajanja	V AC	200	200
Nazivna izlazna snaga	kW	0,4	0,75
Nazivni moment	N·m	1,27	2,39
Maksimalni moment	N·m	3,82	7,16
Nazivna struja	A _{RMS}	2,9	5,1
Maksimalna struja	A _{RMS}	9,2	16,1
Nazivna brzina	min ⁻¹	3000	
Maksimalna brzina	min ⁻¹	6000	
Moment inercije rotora	kg·m ² ×10 ⁻⁴	0,276	1,07
Masa	kg	1,7	3,2
Nazivni napon kočnice	V DC	24	
Nazivna snaga kočnice	W	7,4	9,6
Nazivni moment kočnice	N·m	1,5	3,2
Tip enkodera		23-bitni apsolutni	

Dodavanjem reduktora prethodno odabranim motorima postignuti su redukcija izlazne brzine vrtnje i povećanje zakretnog momenta, te je također osiguran jednostavan način za montiranje ostalih dijelova konstrukcije. S pogonske strane po obodu nalaze se provrti za direktno pričvršćivanje objekta (npr. stola ili robotske ruke) kojem se želi kontrolirati kut rotacije. Izlazno vratilo elektromotora spojeno je preko pera na gonjenu stranu reduktora. Reduktori imaju izgled šupljeg rotacijskog stola, a kroz otvor je moguće provući cijevi i/ili ožičenje. Proizvodi ih kineska kompanija GIGA Precision specijalizirana za proizvodnju prijenosnika.



Prijenosni omjer	1:10
Maksimalni zakretni moment	65 N·m
Maksimalna brzina	250 min ⁻¹
Maksimalno aksijalno opterećenje	4000 N
Točnost pozicioniranja	≅ 1 min
Ponovljivost	±5 s (0,001°)
Masa	8,2 kg

Slika 5-6. Reduktor GSN200-10K-SV [23]

Odabrani reduktori imaju jednaki prijenosni omjer, maksimalnu brzinu, te ponovljivost i točnost pozicioniranja. Razlikuju se u maksimalnom zakretnom momentu i aksijalnom opterećenju koje mogu podnijeti, a reduktor prve osi (slika 5-6.) ima više od tri puta veću masu u odnosu manji reduktor (slika 5-7.) za koji će biti pričvršćena stezna čahura s potpornjem.



Prijenosni omjer	1:10
Maksimalni zakretni moment	32 N·m
Maksimalna brzina	250 min ⁻¹
Maksimalno aksijalno opterećenje	2000 N
Točnost pozicioniranja	≅ 1 min
Ponovljivost	±5 s (0,001°)
Masa	2,5 kg

Slika 5-7. Reduktor GSN130-10K-SV [23]

5.3. Mjerni element prigona

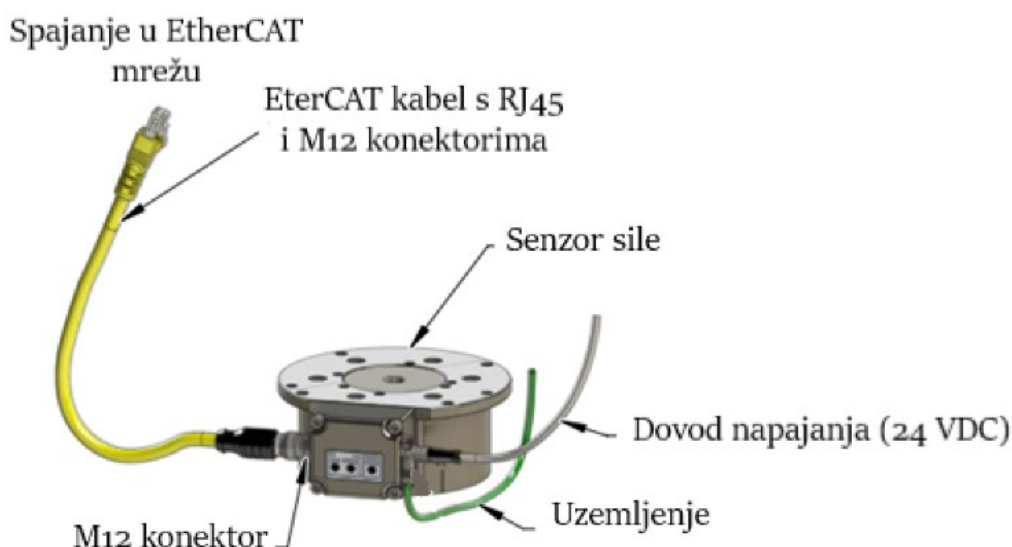
Za mjerenje aerodinamičkih sila i momenata, odabrana je piezoelektrična mjerna ćelija prikazana na slici 5-8. Sposobna je mjeriti svih šest komponenti opterećenja. Glavne prednosti su joj iznimno kompaktne dimenzije, odlična zaštita od preopterećenja, velika otpornost na šumove i visoka osjetljivost. Bit će smještena u težište modela i pričvršćena na vrh potpornja. Ovdje su navedene samo osnovne specifikacije, dok se detaljnije mogu pronaći u [21] ili na mrežnim stranicama proizvođača [24].



Nazivno opterećenje	
F_x, F_y	80 N
F_z	160 N
T_x, T_y	4 Nm
T_z	2 Nm
Fizička svojstva	
Masa	0,0334 kg
Visina	18,2 mm
Promjer	27 mm

Slika 5-8. Mjerna ćelija ATI Mini27 Titanium [24]

Senzor će biti jedan od podređenih uređaja u EtherCAT mreži. Napajanje senzora moguće je ili preko istog EtherCAT kabela koji se koristi za komunikaciju (*Power over EtherCAT*) ili preko vanjskog izvora napajanja. Ovdje je korištena druga opcija gdje se senzor napaja iz istosmjernog izvora napona 24 V kako je prikazano na slici 5-9.



Slika 5-9. Način spajanja mjerne ćelije, prilagođeno iz [24]

6. ODABIR ELEMENATA UPRAVLJAČKOG SUSTAVA I IZRADA ELEKTRO-ORMARA

U prvom dijelu poglavlja će biti odabrani elementi upravljačkog sustava prigona. Kao glavni kriterij prilikom odabira elemenata, uzete su radne karakteristike prethodno odabranih pogonskih motora i senzora sile. Upravljački sustav sastoji se od napajanja, sigurnosnih funkcija, automatskih osigurača i sklopki, logičkih krugova, mrežnih elemenata, servo regulatora te upravljačkog računala. Drugi dio poglavlja opisuje korake poduzete u smjeru izrade elektro-ormara. Nakon odabira elemenata pristupilo se njihovom pričvršćivanju na temeljnu ploču na temelju prethodno izrađenog rasporeda u CAD programu. Na kraju su elementi ožičeni prema električnim shemama nacrtanim u programskom alatu EPLAN.

6.1. Odabir elemenata upravljačkog sustava

6.1.1. Napajanje sustava

Kao glavni izvor napajanja upravljačkog sustava koristi se jednofazna struja napona 230 V i frekvencije 50 Hz. Dovodi se izravno iz električne mreže do rednih stezaljki (X1) smještenih na ulazu u elektro-ormar te se sa njih distribuira dalje do ostalih dijelova sustava (slika 6-2.).

Dijelovi sustava koji su spojeni na izmjeničnu struju (230V/50Hz) su, zajedno s pripadajućom oznakom korištenom u električnim shemama u prilogu, prikazane u tablici 6-1.

Tablica 6-1. Dijelovi sustava koji se napajaju izmjeničnom strujom

Element sustava	Oznaka
Energetski i logički dio servo regulatora ProNet-08AEG-EC	U1
Energetski i logički dio servo regulatora ProNet-08AEG-EC	U2
Utičnica 1	X1
Utičnica 2	X2
Ispravljač EDR-120-12	V1
Ispravljač EDR-120-24	V2
Ispravljač EDR-120-24	V3
Ispravljač NDR-240-480	V4

Servo regulator većeg motora se, prema specifikacijama, napaja trofaznom strujom izmjeničnog napona 200 V. Kako bi se izbjegla ugradnja transformatora, u prvoj fazi razvoja prigona će se pokušati i njega napajati jednofazno te provjeriti je li takvo rješenje prihvatljivo. Cijeli sustav će biti projektiran na način da se omogući jednostavno retroaktivno uvođenje trofazne struje, ukoliko se pokaže da je to ipak potrebno.

Određene komponente sustava, kao što su upravljačko računalo, senzor sile i EtherCAT modul, zahtijevaju istosmjernu struju zbog čega je bilo potrebno odabrati odgovarajući broj ispravljača. U tu svrhu korištena su četiri ispravljača proizvođača MEAN WELL. Svi odabrani ispravljači su konstrukcijski svojim kompaktnim dimenzijama prilagođeni za montažu na standardne din šine unutar elektro-ormara (slika 6-1.). Ispravljači iz EDR serije (prva tri) imaju jednake fizičke dimenzije i nazivnu radnu snagu, a razlikuju se po izlaznom naponu. Iz četvrtog ispravljača, NDR-240-480, će se napajati Ethernet preklopnik čime će mu se osigurati PoE (*Power over Ethernet*) funkcionalnost. PoE (802.3af/at) je standard kojim je definiran način istovremenog prijenosa električne energije i podataka unutar jednog (standardnog) Ethernet kabela.

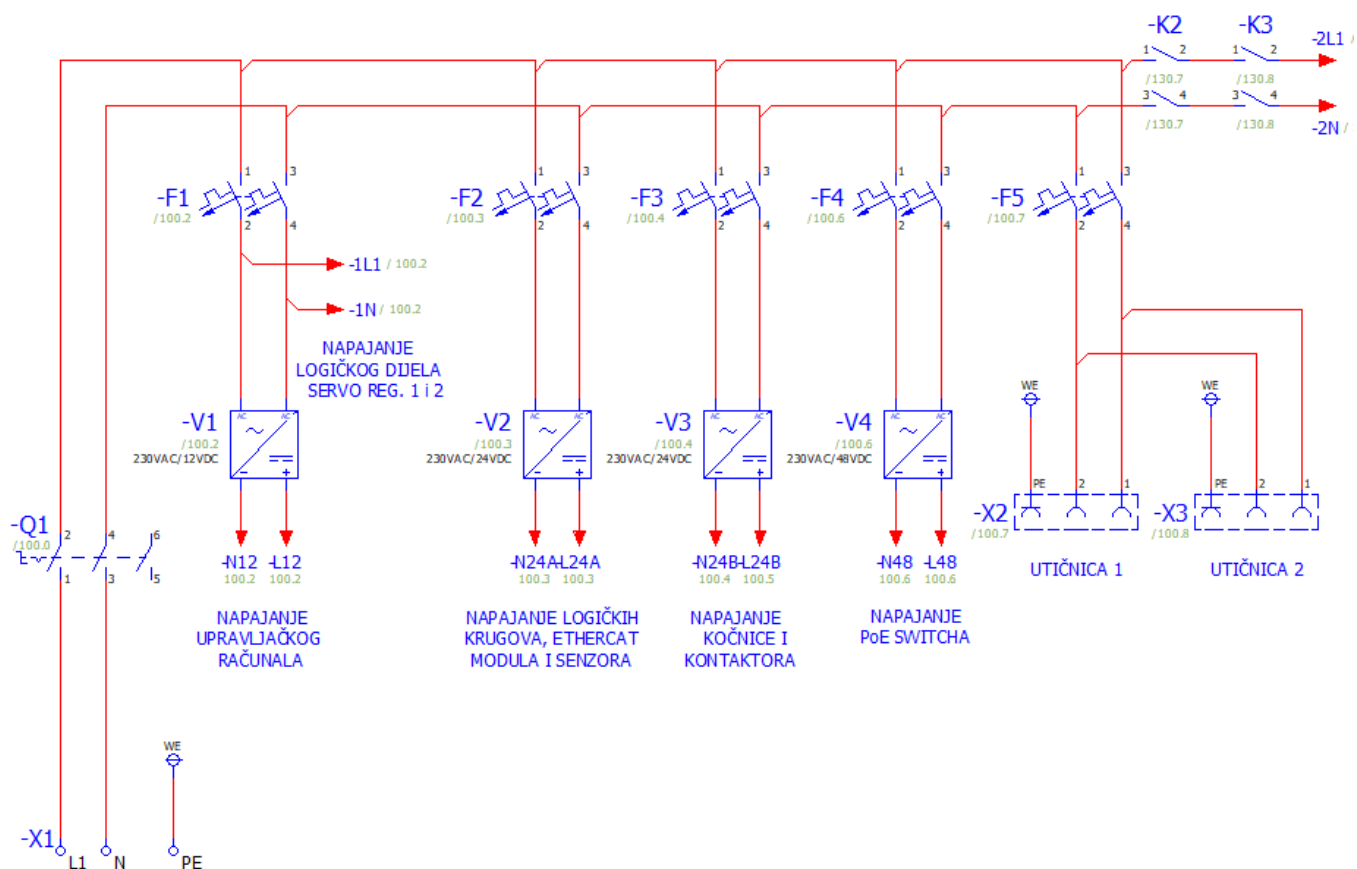


Slika 6-1. Ispravljači MEAN WELL EDR-120-24 i NDR-240-48 [25]

Razlog korištenja dva ista ispravljača (EDR-120-24) je iz nastojanja da se odvoji napajanje logičkih funkcija sustava od induktivnih tereta kao što su kočnice i kontaktori. Problem nastaje jer zavojnice kočnica i kontaktora mogu pohraniti energiju koja bi kratkotrajno još bila prisutna u sustavu nakon prekida napajanja. Na ovaj način su se nastojale spriječiti moguće smetnje između ta dva dijela upravljačkog sustava. Također je dobro razdvojiti jer će na ispravljač V2 biti spojen i senzor sile pa se time smanjila opasnost od pojave šuma. U tablici 6-2. navedene su namjene za koje se koristio pojedini ispravljač.

Tablica 6-2. Izvori istosmjernog napajanja

Izvor	Naziv	Ulazni napon	Izlazni napon	Nazivna radna snaga	Namjena napajanja
Ispravljač V1	MEAN WELL EDR-120-12	230 VAC	12 VDC	120 W	▪ upravljačko računalo
Ispravljač V2	MEAN WELL EDR-120-24		24 VDC		▪ logički krugovi ▪ EtherCAT modul (EK1100) ▪ senzor sile
Ispravljač V3	MEAN WELL EDR-120-24		24 VDC		▪ kontaktori ▪ kočnice servomotora
Ispravljač V4	MEAN WELL NDR-240-48		48 VDC	240 W	▪ PoE preklopnik (Ethernet switch)



Slika 6-2. Električna shema izmjeničnog napajanja

6.1.2. Grebenasta sklopka

Grebenasta sklopka prva je sklopka koja se ugrađuje u sustav, a funkcija joj je omogućavanje fizičkog prekida dovoda napajanja koji se grana dalje po elektro-ormaru. Njezinim zakretanjem osigurano je da niti jedan element elektro-ormara nije pod naponom prilikom provođenja popravka, održavanja ili ožičenja. Zamišljeno je da bude pričvršćena na vrata elektro ormara.

Kad sustav kao glavni izvor energije koristi trofaznu struju, grebenasta sklopka u isključenom stanju prekida sve tri faze. Kako će se pokušati koristiti isključivo jednofazno napajanje, zakretanjem sklopke će se u tom slučaju prekidati fazni i neutralni vodič. To je iz razloga što se kabel od napajanja može u električnu utičnicu ukopčati obostrano, pa se zato ne može sa sigurnošću znati koja žica je faza, a koja je neutralna.

Dimenzioniranje sklopke obavljeno je preko očekivane snage potrebne za rad cijelog sustava. Glavni energetske potrošači su prethodno odabrani motori – veći snage 750 W i manji snage 400 W. Upravljačko računalo, EtherCAT modul i ostale periferne komponente imaju znatno manju nazivnu snagu pa se može pretpostaviti najveća snaga cijelog sustava na oko 1,5 kW.

Odabrana je grebenasta sklopka proizvođača Schneider Electric, model TeSys Vario (slika 6-3.). Nazivna radna snaga joj iznosi 4 kW na naponu od 230 V AC što zadovoljava prethodno navedene pretpostavke. Na električnoj shemi AC napajanja u prilogu ima oznaku Q1.



Slika 6-3. Grebenasta sklopka (Q1) [26]

6.1.3. Automatski osigurači

Automatski osigurači (prekidači) štite elemente spojene u električni krug nakon njih od strujnog preopterećenja (termalna zaštita) i kratkog spoja. Strujni krug se prekida ukoliko struja naraste iznad nazivne vrijednosti osigurača. Prednost u odnosu na klasične osigurače je to što se nakon prekida kruga (nakon što prođe preopterećenje) mogu opet uključiti. Na električnoj shemi napajanja u prilogu označeni su slovom F. Koristit će se pet automatskih osigurača – po jedan za svaki ispravljač te jedan na kojeg će biti spojene električne utičnice. Utičnice će se nalaziti na vanjskoj strani elektro-ormara, a predviđene su za napajanje monitora i sličnih perifernih komponenata.

Odabrani su minijaturni prekidači proizvođača Schneider Electric, nazivne struje 4 A (slika 6-4). Imaju dva pola na koje se spajaju fazni i neutralni vodič. Dvopolni osigurač je skuplji od jednopolnog, a odabran je jer (kao što je već prethodno objašnjeno) se ne može sa sigurnošću znati koji će vodič biti faza, a koji nula. Na taj način se prekidaju oba vodiča istovremeno ukoliko dođe do preopterećenja u jednom od njih.

Iza prvog osigurača spojeno je i napajanje logičkog dijela servo regulatora (vidljivo na prvoj električnoj shemi u prilogu) kako bi se uštedilo na broju potrebnih osigurača, a posljedično i na prostoru u elektro-ormaru. Naime, ovo ima smisla napraviti jer u slučaju da upravljačko računalo ne radi (a napaja se iz prvog ispravljača), nema smisla ni da servo regulatori rade tako da je ova ušteda opravdana.



Slika 6-4. Automatski prekidač iC60N 2P-4A-C [26]

6.1.4. Motorna zaštitna sklopka

Motorna zaštitna sklopka pruža zaštitu elektromotoru na dvije razine. Štiti motor od dugotrajnog strujnog preopterećenja koje bi uzrokovalo prekomjerno zagrijavanje te posljedično dovelo do kvara tj. smanjenja životnog vijeka motora. Ovaj tip zaštite naziva se termalna zaštita, a temelji se na bimetalu koji se zagrijavanjem savija i tako prekida strujni krug. Bitno je napomenuti da zaštitna sklopka dozvoljava kratkotrajni porast struje u periodu paljenja motora i u tom slučaju ne prekida strujni krug. Sklopka štiti motor i od kratkog spoja na način da se uslijed velike struje kratkog spoja elektromagnet (koji je sastavni dio sklopke) dovoljno magnetizira da privuče armaturu te se strujni krug prekida.

Odabir zaštitne sklopke vrši se prema radnim karakteristikama prethodno izabranih motora (tablica 5-1.). Sklopka je dimenzionirana prema nazivnim strujama motora koje iznose 2,9 i 5,1 A. Odabrane su motorne zaštitne sklopke proizvođača Schneider Electric, model TeSys Deca. Sklopke imaju podesivi raspon termalne zaštite od 2,5 do 4 A, a zaštita od vršnih strujnih opterećenja aktivira se na 51 A.

Na slici 6-5. prikazana je odabrana motorna zaštitna sklopka. Zakretanjem vijka na prednjoj strani potrebno je postaviti željenu razinu termalne zaštite. Sklopka također sadrži mehanički prekidač preko kojeg je moguće ručno isključiti i uključiti dovod napajanja do motora u slučaju njegove zamjene i slično. Na električnim shemama u prilogu su označene sa Q2 i Q3.



Slika 6-5. Motorna zaštitna sklopka TeSys GV2 [26]

6.1.5. Sigurnosni logički krug

Jedan od najbitnijih dijelova sustava je sigurnosni logički krug koji omogućava sigurno pokretanje kritičnih dijelova sustava (npr. motora) i njihovo zaustavljanje u nuždi. Upravljački elementi koji se spajaju u ovaj krug su sigurnosni relej, kontaktori, tipkala za pokretanje i za zaustavljanje u nuždi.

Glavni element sigurnosnog kruga je sigurnosni relej koji radi na principu klasičnog releja, ali za razliku od njega ima veći stupanj pouzdanosti (veći broj radnih ciklusa) što znači da je to komponenta od koje se očekuje da će zadnja zakazati. Veći broj radnih ciklusa kod sigurnosnog releja osiguran je izradom njegovih kontakata od posebnih legura (AgSn0) što ga čini znatno skupljim od klasičnog releja.

Odabran je SRB-E-301MC sigurnosni relej proizvođača Schmersal (slika 6-6.). Ima tri sigurnosna kontakta i jedan signalni koji se zatvaraju u trenutku kad je unutarnja logika releja zadovoljena. Zavojnica releja napaja se naponom od 24 V DC iz ispravljača V2 (slika 6-7. A). U tablici 6-3. opisani su svi priključci korištenog sigurnosnog releja.

Zavojnica sigurnosnog releja i zavojnice kontaktora napajaju se istosmjernom strujom iz dva odvojena ispravljača zbog sprječavanja smetnji u logičkom strujnom krugu. Sigurnosni relej u ovom slučaju kontrolira protok struje iz ispravljača V3 do zavojnica kontaktora K2 i K3 (slika 6-6.).

Preduvjeti koji moraju biti ispunjeni da bi sigurnosni relej (pritiskom tipkala za pokretanje) zatvorio svoje kontakte i propustio napajanje do kontaktora jesu:

- Tipkala za zaustavljanje u nuždi S1 i S2 su u otpuštenom stanju
- Radni kontakt releja K4 je zatvoren
- Mirni kontakti kontaktora (K2 i K3) su zatvoreni

U slučaju da su svi preduvjeti ispunjeni, zavojnice kontaktora zatvaraju radne kontakte i propuštaju jednofazno napajanje (230 V/50 Hz) prema motorima i ostalim elementima koji se spajaju iza njih. Preklapanjem mehaničkih kontakata unutar kontaktora dolazi do njihovog trošenja i uslijed iskrenja može se dogoditi da se zavare u zatvorenoj poziciji (ostanu u tom položaju nakon pritiska tipkala za zaustavljanje). Iz sigurnosnih razloga koriste se dva kontaktora jer u slučaju da dođe do prethodno spomenute situacije (a koristi se samo jedan kontaktor) ne bi bilo moguće prekinuti dovod energije prema motorima što bi moglo dovesti do ozbiljnih posljedica na radnike i na sami stroj.



Slika 6-6. Sigurnosni relej (K1) [27] i kontaktor (K2 i K3) [26]

Tablica 6-3. Opis terminala sigurnosnog releja

Oznaka	Funkcija	Oznaka	Funkcija
A1	Radni napon zavojnice, 24 V DC	S21	Izlazni kanal 2
A2	Neutralni kraj zavojnice, 0 V	S12	Ulazni kanal 1
X1	Izlaz kruga za pokretanje	S22	Ulazni kanal 2
X2	Ulaz kruga za pokretanje	41/42	Signalni mirni kontakt
S11	Izlazni kanal 1	13/14 23/24 33/34	Sigurnosni radni kontakti

Na slici 6-7. B prikazan je krug u koji se spajaju tipkala za zaustavljanje. Uz njih je također serijski spojen radni kontakt releja K4. Kao što je vidljivo sa slike koristit će se dva tipkala S1 i S2. Jedno će se nalaziti na vanjskoj strani elektro-ormara, dok će se drugo smjestiti negdje na zračni tunel kako bi se osigurala pristupačnost u slučaju nužde. Imaju potpuno jednaku funkcionalnost – pritiskom na njih prekida se napajanje prema motorima. Logički dio sustava (svi elementi spojeni prije kontaktora kao što su logički dio servo regulatora ili upravljačko računalo) ostaju pod napajanjem i nakon pritiska tipkala.

Tipkalo S4 ima dva radna kontakta, od kojih se jedan koristi za pokretanje sustava (slika 6-7 C). Drugi je spojen kako je to prikazano na slici 6-7. pod D, a koristi se za poništavanje greške alarma servo regulatora prilikom ponovnog pokretanja sustava (digitalni ulazi i izlazi regulatora objašnjeni su u sljedećem potpoglavlju).

Između priključaka 23 i 24 sigurnosnog releja, spojeno je napajanje iz ispravljača V3 prema kontaktorima K2 i K3 kao što je prikazano na slici 6-7. E. Također je vidljivo da su po jedan radni i mirni kontakt sigurnosnog releja ostali neiskorišteni. Dodatna razina sigurnosti kod sigurnosnog releja postignuta je upotrebom dvostrukih radnih kontakata u unutrašnjosti releja. Na slici 6-8. prikazana su korištena tipkala zajedno s pripadajućim radnim (engl. *normally open*, NO) i mirnim (engl. *normally closed*, NC) kontaktima koji se pričvršćuju sa stražnje strane tipkala.



Slika 6-8. Tipkalo za pokretanje (S4) i tipkalo za zaustavljanje u nuždi (S1 i S2) i njihovi kontakti [26]

6.1.6. Servo regulatori

U ovom potpoglavlju predstavljeni su odabrani servo regulatori ProNet serije, proizvođača ESTUN (slika 6-9.). Dolaze s podrškom za EtherCAT komunikacijsku sabirnicu. Servo regulatori su upareni s pripadajućim motorima EM3A-08A i EM3A-04A na temelju radnih karakteristika motora i prema preporukama proizvođača [22]. Tehničke specifikacije odabranih servo regulatora navedene su u tablici 6-4.



Slika 6-9. Servo regulator ProNet-04AEG-EC (lijevo) i ProNet-08AEG-EC (desno) [22]

Tablica 6-4. Tehničke specifikacije odabranih servo regulatora

	Jedinica	ProNet-04AEG-EC	ProNet-08AEG-EC
Nazivna snaga	kW	0,4	0,75
Napajanje glavnog kruga	V AC	1-fazno 200 ~ 230	3-fazno 200 ~ 230
Napajanje regulacijskog kruga	V AC	1-fazno 200 ~ 230	
Kontinuirana izlazna struja	A _{RMS}	2,8	4
Maksimalna izlazna struja	A _{RMS}	8,4	12

Oba odabrana servo regulatora imaju iste oznake priključaka, a funkcije su im navedene u tablici 6-5. Jedina razlika je što se servo regulator manjeg motor (ProNet-04AEG) napaja jednofazno pa ima samo L1 i L2 priključke na koje se spajaju fazni i neutralni vod. Priključci L1, L2 i L3 servo regulatora većeg motora (ProNet-08AEG) predviđeni su za tri faze, ali kako će se i on pokušati napajati jednofazno koristit će se samo priključci L1 i L2.

Motor koči mehaničkom kočnicom jedino u slučaju iznenadnog nestanka napajanja, pritiska tipke za zaustavljanje u nuždi ili greške u servo regulatoru. U normalnom radu motor koči protustrujno pa, kako bi se ograničio nagli porast struje (koja u tom slučaju teče kroz armaturu) potrebno je dodati otpornik odgovarajućeg iznosa otpora.

Servo regulatori u sebi imaju ugrađen unutranji otpornik koji je dovoljan u većini slučajeva, a kod odabranih regulatora spojen je tvornički između priključaka B2 i B3. U slučaju naglog kočenja motora koji pokreće teret velikog momenta inercije, potrebno je spojiti vanjski otpornik s većim iznosom otpora između priključaka B1 i B2 kako bi se mogla disipirati velika količina energije.

Tablica 6-5. Priključci odabranih ESTUN ProNet servo regulatora

Oznaka	Funkcija	Komentar
L1, L2, L3	napajanje energetskog dijela regulatora	▪ kod servo regulatora napajanog jednofaznim naponom samo L1 i L2
+1, +2, -	istosmjerni međukrug	▪ moguća regeneracija energije spajanjem u seriju s drugim servo regulatorima
L1C, L2C	napajanje logičkog dijela regulatora	
B1, B2, B3	spajanje kočionih otpornika	▪ između B2-B3 spaja se unutarnji otpornik, a između B1-B2 vanjski otpornik
U, V, W	priključci za napajanje motora	▪ pulsno-širinski modulirani naponski valni oblici, kabel je potrebno oklopiti i uzemljiti
CN1	priključak za digitalne ulaze i izlaze	▪ smjer signala je moguće invertirati, nije potrebno koristiti sve I/O signale
CN2	priključak za kabel od enkodera	▪ kabel je potrebno oklopiti i uzemljiti samo s jedne strane
CN3	ulazni RJ45 konektor	▪ EtherCAT komunikacija
CN4	izlazni RJ45 konektor	▪ EtherCAT komunikacija
CN5	konektor za univerzalnu serijsku sabirnicu (USB)	▪ uspostava komunikacije između PC-a i regulatora preko USB sabirnice

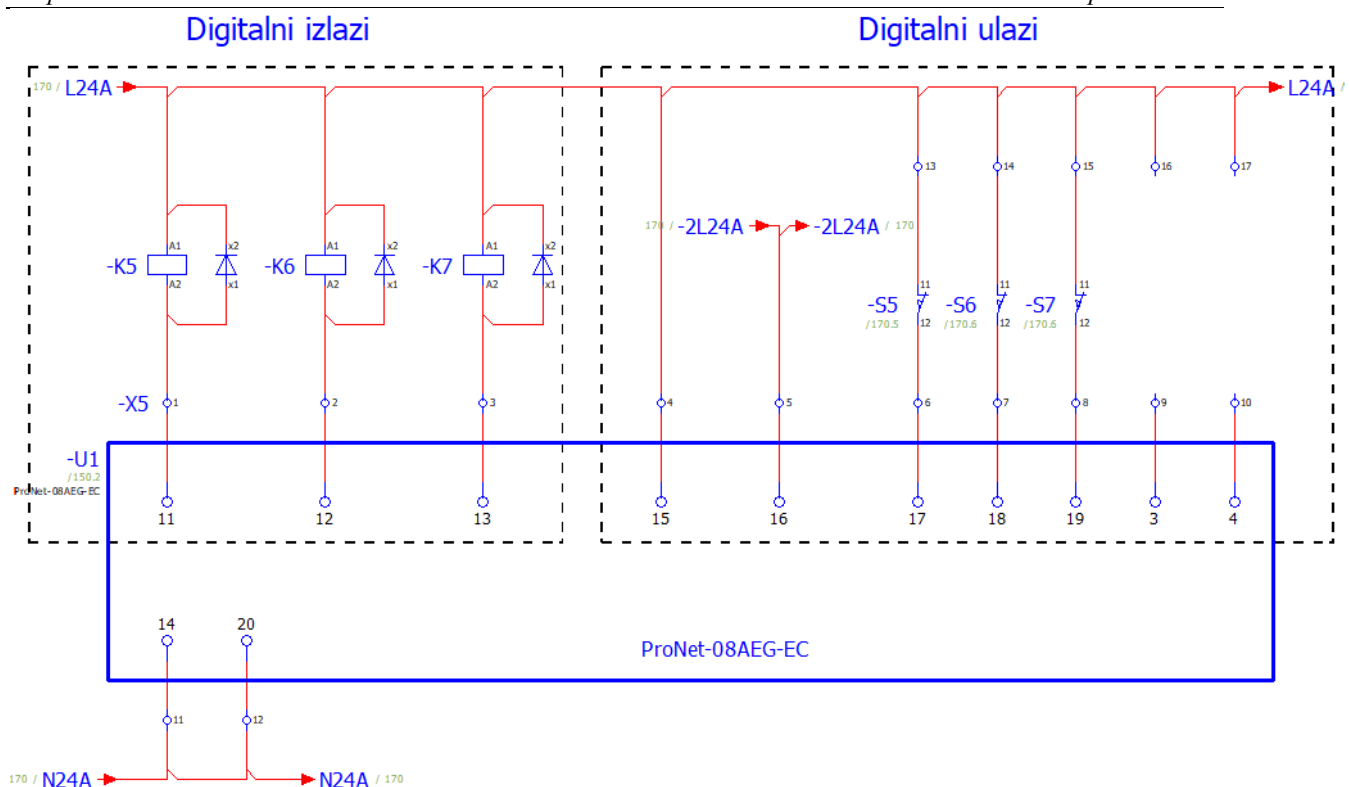
ESTUN ProNet regulatori s podrškom za EtherCAT imaju tri izlazna signala od kojih je jedan uvijek dojava greške u regulatoru, dok se preostala dva mogu dodijeliti nekom od izlaza navedenih u tablici 6-6. promjenom parametra Pn511. Imaju i sedam ulaznih signala od kojih je pet dozvoljeno proizvoljno alocirati promjenom parametra Pn509.

Za digitalne ulaze i izlaze koriste se naponski signali od 24 V. Osim ulaza EXT1 i EXT2, sve ostale signale moguće je invertirati promjenom parametara regulatora. Za invertiranje smjera ulaznih signala potrebno je promijeniti stanja parametara Pn516 i Pn517, a izlaznih signala stanja parametar Pn528.

Tablica 6-6. Digitalni ulazi i izlazi odabranih servo regulatora

	Oznaka	Opis
Ulazi	S-ON	▪ dozvola gibanja
	P-CON	▪ uključivanje upravljanja po poziciji
	P-OT	▪ zabrana gibanja u pozitivnom smjeru
	N-OT	▪ zabrana gibanja u negativnom smjeru
	ALM-RST	▪ poništavanje alarma
	EXT1	▪ referentni signal 1
	EXT2	▪ referentni signal 2
Izlazi	COIN	▪ zadana pozicija je postignuta
	TGON	▪ motor je postigao brzinu veću od postavljene vrijednosti
	S-RDY	▪ servo regulator je spreman za rad
	CLT	▪ iznos zakretnog momenta došao na postavljeni limit
	BK	▪ signal za uključivanje kočnice
	PGC	▪ indeksni signal enkodera
	OT	▪ prigon se nalazi na graničnom prekidaču
	RD	▪ uključeno napajanje motora
	HOME	▪ početno referenciranje je izvršeno
	ALM	▪ dojava greške u regulatoru

Ulazni signali koji će biti korišteni su signal dozvole gibanja, signali s graničnih i referentnog prekidača te signal za poništavanje greške alarma. Od izlaznih signala koristit će se signali za dojavu greške i paljenje kočnice te signal za potvrdu stanja servo regulatora (slika 6-10). Oba servo regulatora imaju iste digitalne ulaze i izlaze na priključku CN1.



Slika 6-10. Digitalni ulazi i izlazi servo regulatora

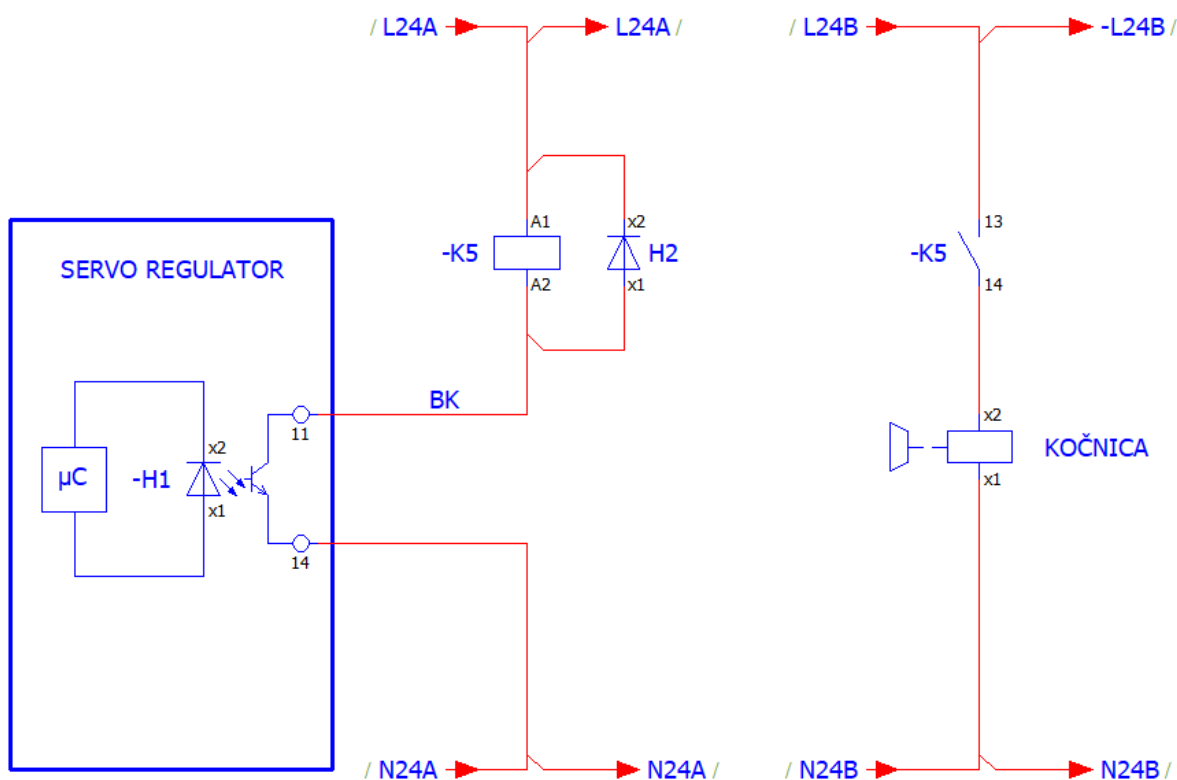
Dva granična prekidača (po jedan za svaki smjer rotacije prigona) označena su kao S5 i S6 (kod drugog regulatora su to S8 i S9). Imaju mirne kontakte jer će u slučaju prekida kabla kojim su spojeni sa servo regulatorom signalizirati da se nalaze na graničnom prekidaču i gibanje će se zaustaviti. Referentni prekidač također ima mirni kontakt, a na slici ima oznaku S7. Dozvola gibanja (S-ON) dovodi se na pin 15, a poništavanje greške alarma prilikom ponovnog pokretanja sustava na pin 16. Po dva izlaza (12 i 13) i ulaza (3 i 4) su ostala neiskorištena u trenutnoj konfiguraciji sustava.



Slika 6-11. Releji za paljenje kočnice s prihvatom za DIN šinu (K5 i K8) [28]

Elemente upravljačkog sustava kao što su servo regulatori ili I/O modul potrebno je galvanski odvojiti od električkih krugova kojima upravljaju kako bi ih se zaštitilo od prevelike jakosti struje. Naime, struje koje teku kroz logičke krugove unutar servo regulatora iznose nekoliko miliampera, dok su struje čijim protokom upravljaju značajno veće. Galvanska izolacija osigurana je pomoću elektroničke komponente koja se zove opto-izolator, a sastoji se od LED diode i fototranzistora. Svjetlost s LED diode upada na bazu od fototranzistora koji posljedično zatvara strujni krug propuštanjem struje između emitera i kolektora.

Na slici 6-12. prikazan je način uključivanja (odnosno isključivanja) kočnice upravljačkim signalom iz servo regulatora (koji predstavlja digitalni izlaz). Mehanička kočnica nalazi se na kraju vratila servo motora odmah uz enkoder. U početnom stanju (dok nije pod napajanjem) rotor motora se ne može okretati jer je kočnica zatvorena. Kočnici je potrebno dovesti napajanje od 24 V DC kako bi se otvorila i motor mogao početi okretati. Servo regulator upravlja protokom struje kroz zavojnicu releja K5 (slika 6-11.) čiji se radni kontakti zatvaraju i otvaraju u ovisnosti o kontrolnoj struji kroz relej. Koristi se radni kontakt jer, u slučaju da cijeli sustav ostane bez napajanja, kočnica će i dalje raditi (u stanju bez napajanja je zatvorena). Povratna dioda H2 pruža zaštitu elektroničkim komponentama od kratkotrajnog vršnog napona koji se javlja nakon isključivanja releja K5 zbog induktiviteta zavojnice.

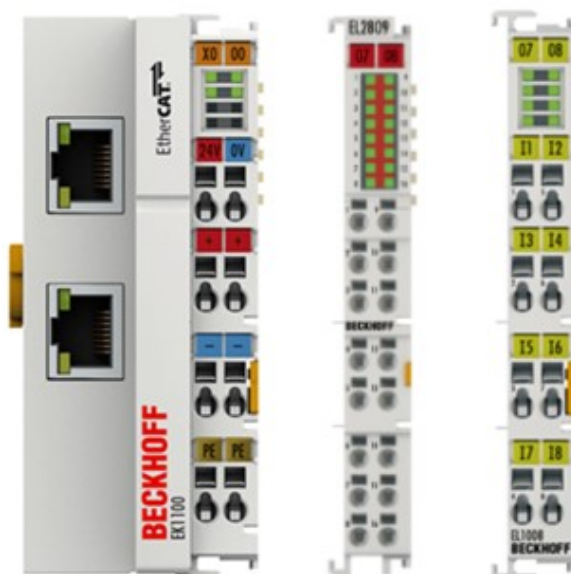


Slika 6-12. Princip upravljanja kočnicom motora iz servo regulatora

6.1.7. EtherCAT I/O modul

Beckhoff EK1100 EtherCAT modul (slika 6-13.) koristit će se kako bi se omogućilo upravljanje dijelovima sustava izravno iz LinuxCNC-a i kako bi se moglo nadzirati trenutačno stanje pojedinih elemenata upravljačkog sustava. Modul EK1100 napaja se istosmjernim naponom od 24 V, a sa strane mu se mogu prikvačiti proširenja u vidu digitalnih izlaza i ulaza. Proširenje EL2809 osigurava četiri digitalna izlaza, a proširenje EL1008 ima osam digitalnih ulaza (slika 6-13.). Oba koriste digitalne naponske signale od 24 V DC.

Primjerice preko digitalnog ulaza moglo bi se saznati je li tipkalo za zaustavljanje pritisnuto ili ne, te nalazi li se prigon na graničnom prekidaču i slično. Također se mogu dovoditi signali s pomoćnih kontakata releja i kontaktora i na taj način izravno iz grafičkog sučelja nadzirati stanje određenih dijelova sustava. S druge strane, digitalni izlazi pružaju mogućnost upravljanja dijelovima sustava direktno iz upravljačkog sustava (u ovom slučaju LinuxCNC-a). Na električnoj shemi spajanja sigurnosnog releja prikazanoj na slici 6-7., može se uočiti da je u seriju s dva tipkala za zaustavljanje (S1 i S2) spojen i preklopni kontakt releja K4. Kako bi se propustila električna energija do pogonskog dijela sustava (zatvorio logički krug između izlaza S11 i S21, te ulaza S12 i S22 sigurnosnog releja), potrebno je zatvoriti taj kontakt. Kontaktima releja K4 moguće je upravljati na dva načina. U prvoj fazi testiranja to se vrši ručno preko sklopke na prednjoj strani releja. U normalnom režimu rada će se njegovim stanjem upravljati iz LinuxCNC-a preko digitalnog izlaza. Ovime je omogućeno zaustavljanje prigona iz LinuxCNC-a ukoliko dođe do greške u radu sustava.



Slika 6-13. EtherCAT modul s I/O proširenjima [29]

6.1.8. Upravljačko računalo

Računalo se nalazi na najvišoj hijerarhijskoj razini unutar CNC sustava, a u EtherCAT mreži predstavlja nadređeni uređaj iz kojeg se naredbe šalju prema podređenim uređajima. Na njemu su instalirani Linux operativni sustav, LinuxCNC s odgovarajućim ekstenzijama za rad u stvarnom vremenu, te driveri za EtherCAT sabirnicu. Korišteno je industrijsko računalo kompaktnih dimenzija prikazano na slici 6-14. Napaja se s prvog ispravljača naponom od 12 V. Sastoji se od Intel Celeron J1900 procesora s četiri jezgre, DDR3 radne memorije od 4 GB, te tvrdog diska memorije 120 GB. Ima četiri USB ulaza, po dva RS232 i LAN ulaza, te priključke za VGA i HDMI.



Slika 6-14. Industrijsko računalo XCY X30 J1900 [30]

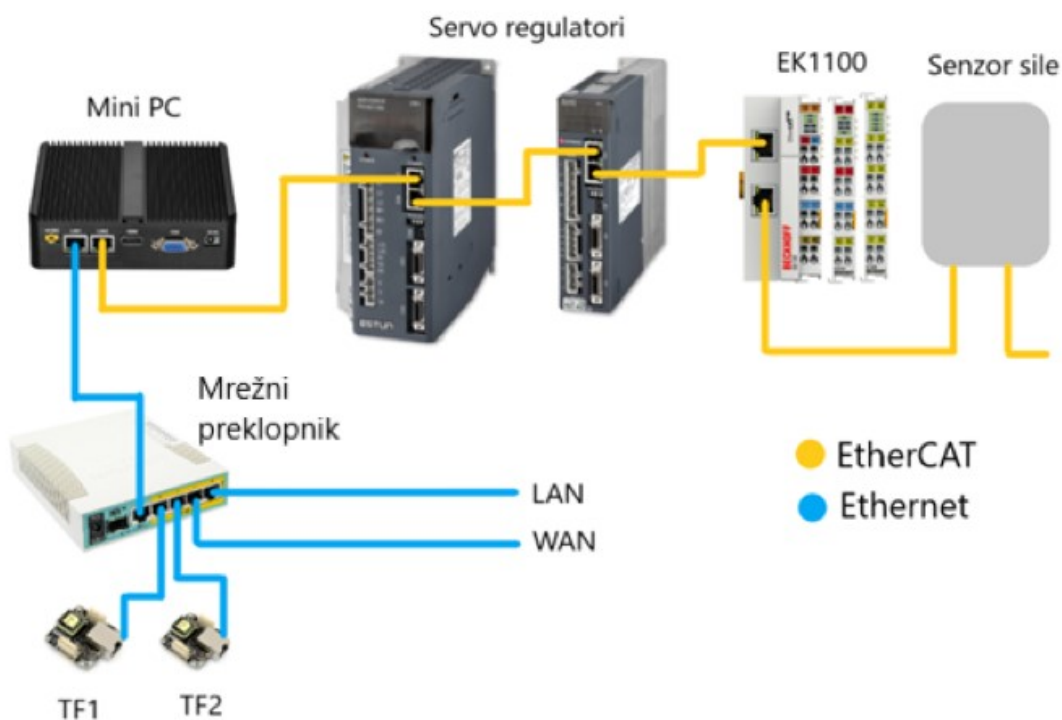
6.1.9. Mrežna topologija

Na slici 6-15. prikazana je mrežna topologija odnosno način na koji je uspostavljena komunikacija između različitih dijelova sustava. Preko prvog mrežnog priključka na upravljačkom računalu odvija se Ethernet komunikacija korištenjem standardnog TCP/IP protokola, dok su na drugom LAN priključku lančano spojeni uređaji u EtherCAT mrežu.

Ethernet komunikacija nije deterministička i koristi se jer neki elementi sustava ne podržavaju EtherCAT komunikaciju. To se odnosi na TinkerForge module spojene na drugi i treći LAN priključak mrežnog preklopnika. Mrežni preklopnik napaja se iz četvrtog ispravljača naponom od 48 V čime mu je osigurana PoE (*Power over Ethernet*) funkcionalnost.

TinkerForge je hardverska modularna platforma otvorenog tipa, a osnovni gradivni elementi su joj *brick*, *bricklet* i *master extension* moduli (više u [31]). Prvi TinkerForge modul će se koristiti za uspostavu komunikacije s postojećim varijabilnim frekvencijskim pretvaračem čime će biti omogućeno upravljanje brzinom strujanja zraka kroz tunel iz LinuxCNC-a. Drugi modul će se koristiti za akviziciju podataka o temperaturi i vlažnosti zraka u zračnom tunelu.

U EtherCAT mrežu spajaju se servo regulatori, EtherCAT EK1100 modul i senzor sile. Preko nje se PDO porukama prenose vremenski osjetljive procesne informacije do servo regulatora (npr. referenca brzine), a u suprotnom smjeru trenutni položaj prigona s enkodera i izmjerena opterećenja s mjerne ćelije do upravljačkog računala. EtherCAT sabirnica se također koristi za početno konfiguriranje sustava preko SDO poruka kako je to već objašnjeno u 4. poglavlju.



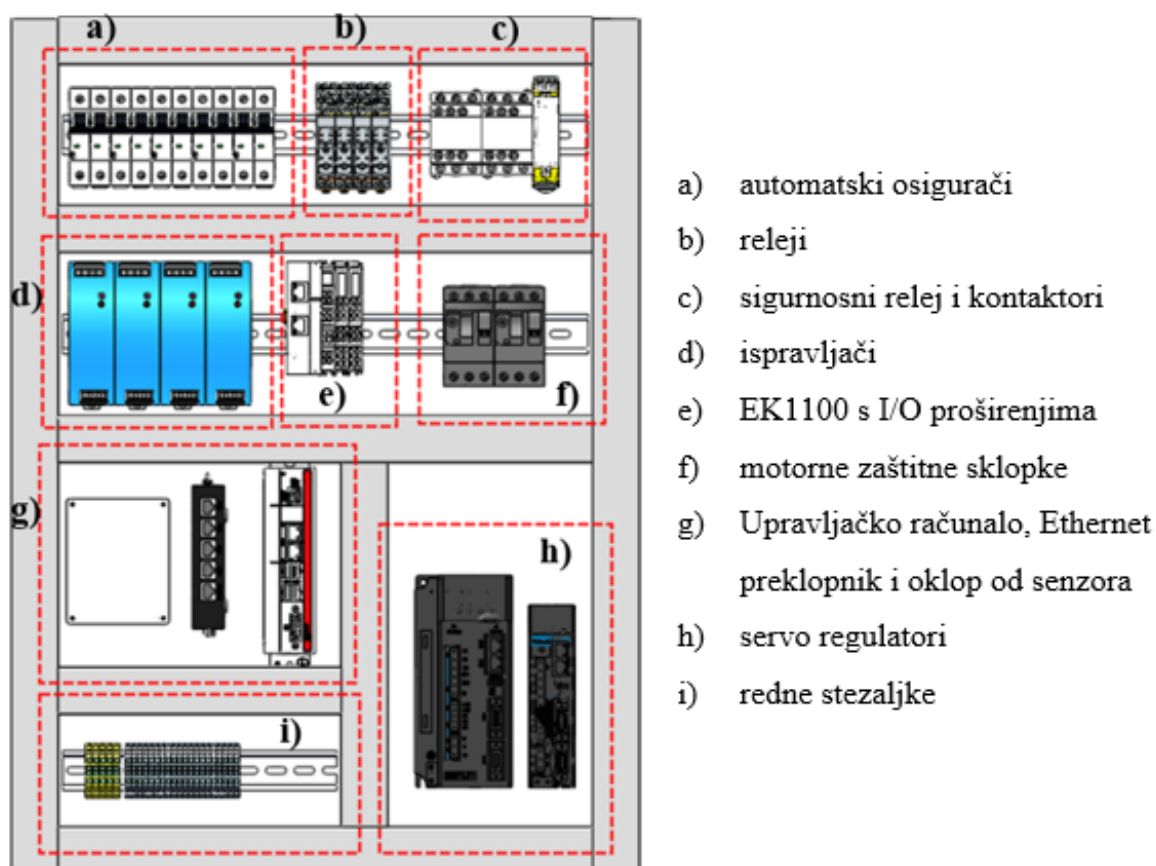
Slika 6-15. Mrežna topologija

6.2. Izrada elektro-ormara

Nakon odabira svih elemenata upravljačkog sustava, bilo je potrebno odrediti njihov prostorni raspored unutar elektro-ormara. Raspored elemenata inicijalno je izrađen u CAD programu SolidWorks 2020. Zatim se pristupilo pričvršćivanju kanala, din šina i upravljačkih elemenata na temeljnu ploču elektro-ormara i njihovom međusobnom ožičenju na temelju prethodno izrađenih električkih shema u programskom alatu EPLAN.

6.2.1. Određivanje prostornog rasporeda elemenata

U početnoj fazi odlučeno je da će svi energetske i signalni kablovi u elektro-ormar ulaziti s donje strane. Jedno od ograničenja kod planiranja razmještaja elemenata bile su ograničene duljine kablova kojima su enkoder i motor spojeni sa servo regulatorima. Servo regulatori se nalaze unutar elektro-ormara, dok su motori pričvršćeni na sami uređaj. Iz tog razloga se nastojalo spustiti regulatore na dno temeljne ploče (slika 6-16). Dodatno ograničenje je bilo nastojanje da se odvoje signalni od energetskih vodiča kako bi se spriječila njihova međusobna interferencija.

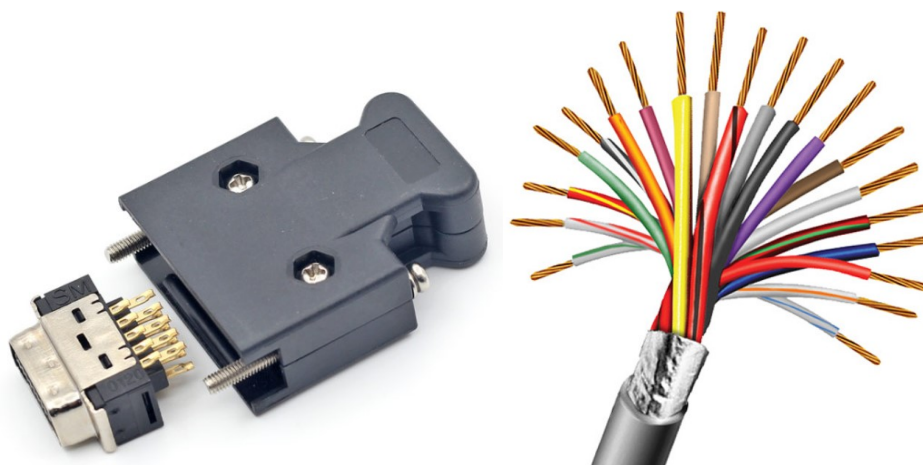


Slika 6-16. Razmještaj elemenata na temeljnoj ploči elektro-ormara

Kako bi se oslobodio prostor za potencijalna buduća proširenja sustava u vidu dodatnih releja ili digitalnih ulaza i izlaza na EtherCAT 1100 modul, odlučeno je da će se upravljačko računalo, mrežni preklopnik i sučelje senzora sile s odgovarajućim kućištem izmjestiti izvan elektro-ormara. Oni će biti pričvršćeni na bočnu stranu elektro-ormara, a napajanje (i mrežni kabel) će se prema tim komponentama provesti kroz za to predviđen otvor na ormaru.

6.2.2. Priprema konektora za digitalne I/O servo regulatora

Na priključak CN1 servo regulatora spajaju se digitalni ulazi i izlazi kao što su signal za paljenje kočnice, signal za dozvolu gibanja ili signali s graničnih prekidača (prethodno navedeni u tablici 6-6.). U ovom koraku bilo je potrebno proizvoljno dodijeliti pojedinim signalima boje žica vodova kojima će se povezati (tablica 6-7.). Jedini uvjet je bio da se poštuju jednom izabrane boje i na drugom konektoru (od drugog servo regulatora) radi lakšeg snalaženja prilikom njihovog spajanja.

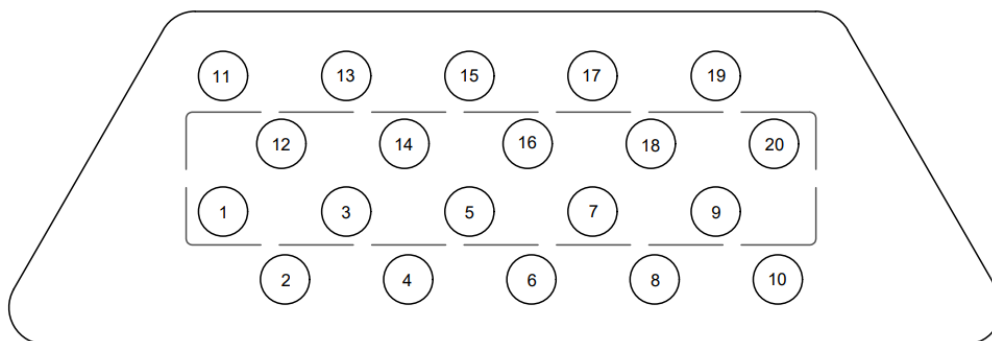


Slika 6-17. SCSi 20-pinski muški konektor i oklopljeni kabel s vodičima

Tablica 6-7. Uparivanje vodiča i 20-pinskog konektora SCSi

Pin	Oznaka	Boja vodiča	Pin	Oznaka	Boja vodiča
3	EXT1	crveno-plava	15	S-ON	smeđa
4	EXT2	žuto-smeđa	16	P-CON	roza
11	TGON+	ljubičasta	17	P-OT	siva
12	ALM+	crna	18	N-OT	zelena
13	S-RDY+	bijela	19	ALM-RST	žuta
14	COM2	crvena	20	DICOM	plava

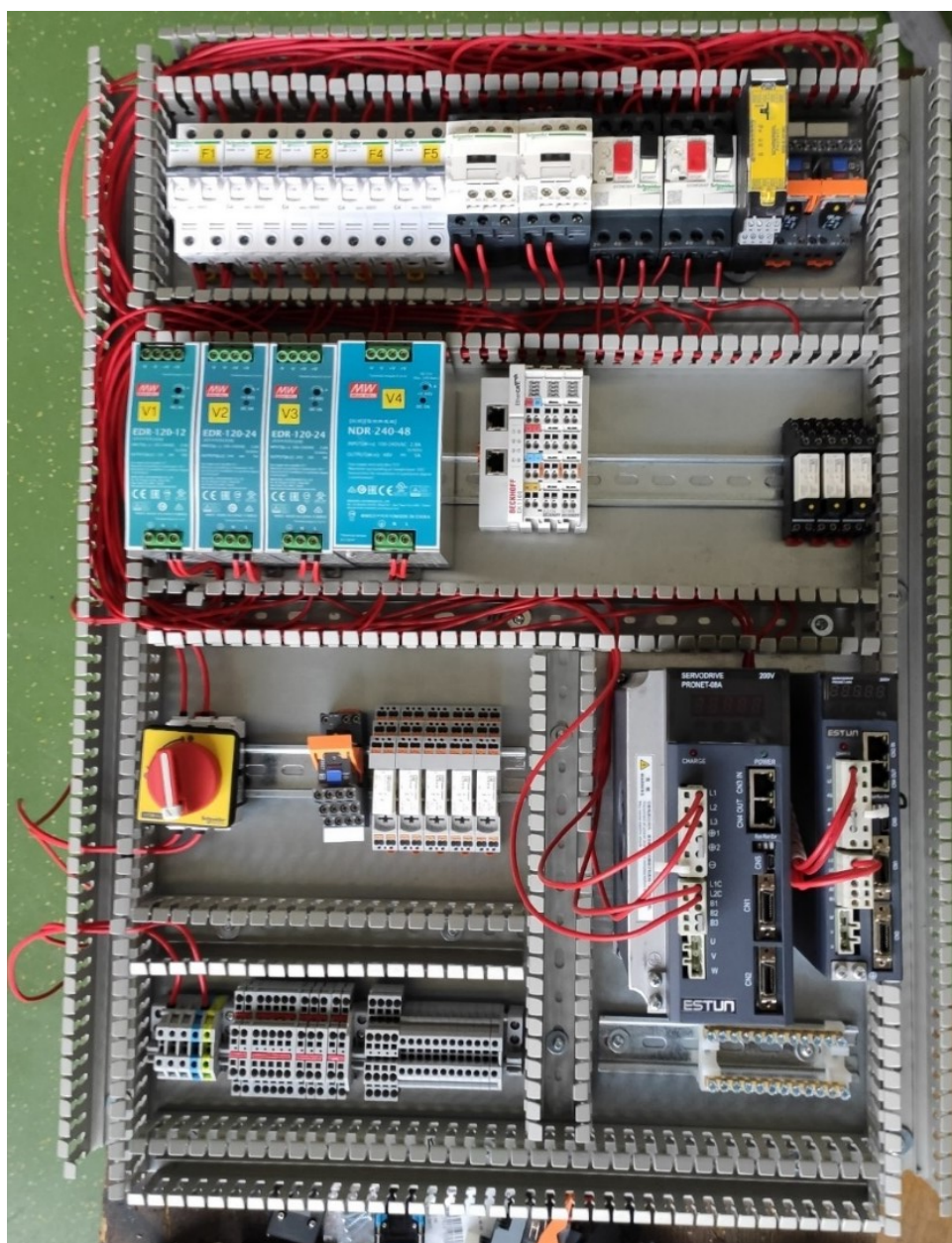
Broj pina pojedinog signala naveden je u EtherCAT priručniku proizvođača servo regulatora [32], dok je broj pojedinog pina na konektoru prikazan na slici 6-18. Od navedenih signala u tablici 6-6., na pinove konektora SCSI (slika 6-17.) spojeni su svi osim EXT1 i EXT2, no oni se mogu naknadno spojiti ukoliko se za to pokaže potreba.



Slika 6-18. Raspored pinova na konektoru

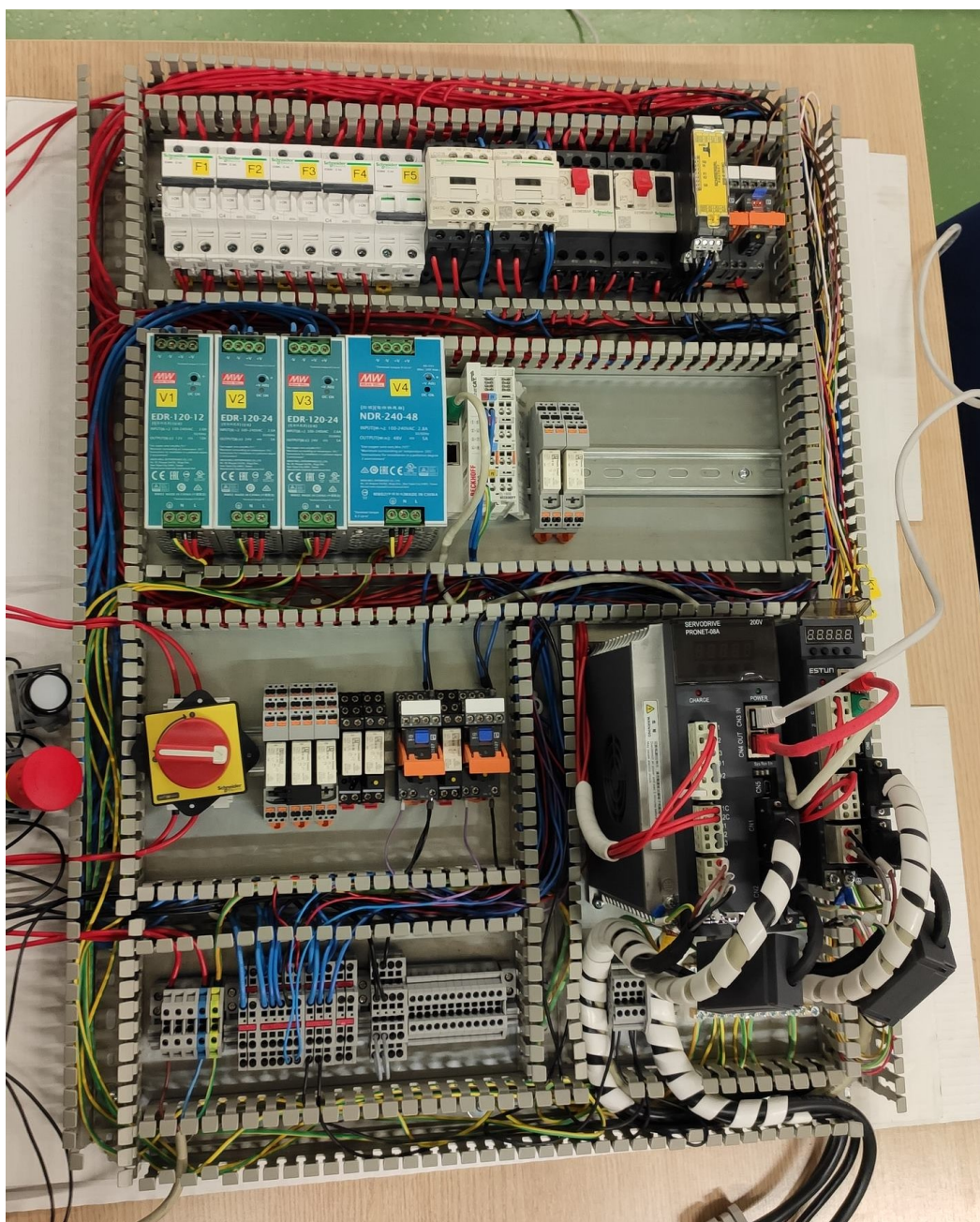
6.2.3. Ožičenje elemenata sustava

Slika 6-19. prikazuje provođenje napojnih vodova do elemenata sustava, a koje je izvedeno prema prvoj električnoj shemi u prilogu. Struja se iz električne mreže dovodi do rednih stezaljki u donjem lijevom kutu upravljačkog ormara, od kud se potom distribuira do ostalih dijelova sustava. Fazni i neutralni vodič su u ovome slučaju iste boje (crvene) iz razloga što im se mogu zamijeniti uloge u ovisnosti o tome na koju stranu je kabel od napajanja uključen u utičnicu. Iz istoga razloga će grebenasta sklopka svojim zakretanjem prekidati oba vodiča. Energetski (L1 i L2) i logički dijelovi (L1C i L2C) oba servo regulatora se napajaju jednofaznim napajanjem 230 V AC.



Slika 6-19. Provođenje napojnih vodova

U zadnjem koraku prije uključivanja sustava provedeno je spajanje sigurnosnog logičkog kruga. Ispravljači, EK1100 modul i servo regulatori uzemljeni su preko sabirnice uzemljenja (donji desni kut slike 6-20.). Također su spojeni priključci CN1 i CN2 servo regulatora, te napajanja do motora (oznake U, V, W) i kočnice. Kočnice su zbog svog induktivnog karaktera spojene na treći ispravljač (24 V DC) odvojeno od napajanja logičkih funkcija (npr. releja) koje su na drugom ispravljaču. Nakon provedenog ožičenja, multimetrom je provedeno ispitivanje svih spojeva u elektro-ormaru kako bi se utvrdilo jesu li ispravno spojeni.



Slika 6-20. Ožičenje logičkih dijelova sustava i spajanje priključaka servo regulatora

7. PARAMETRIRANJE I TESTIRANJE SUSTAVA

U ovom poglavlju opisani su koraci kod konfiguriranja sustava LinuxCNC koje je potrebno provesti prije testiranja gibanja rotacijskih prigona. Dvije osnovne datoteke u LinuxCNC-u su INI i HAL datoteka. Prva se koristi za opis arhitekture stroja (npr. broj i tip posmičnih osi) i sadrži sve parametre sustava (npr. tip mjernih jedinica, pojačanja PID regulatora i dr.). HAL datoteka opisuje izgled apstrakcijskog sloja, odnosno u njoj se učitavaju komponente i povezuju signalima kako je to već opisano u poglavlju 2.3. Za uspostavu komunikacije između LinuxCNC-a (NC računala) i servo regulatora motora, potrebno je učitati konfiguracijsku datoteku za EtherCAT. Nakon početnog postavljanja upravljačkog sustava, pristupilo se eksperimentalnom podešavanju pojačanja regulacijskih petlji prigona slanjem kontinuiranih signala sa softverskog generatora signala i promatranju odziva u HAL osciloskopu.

7.1. Početno konfiguriranje sustava

7.1.1. INI datoteka

INI datoteka je konfiguracijska datoteka LinuxCNC softvera. Svaka konfiguracija stroja zahtijeva vlastitu INI datoteku. Njome se opisuje kinematika stroja, zadaju ograničenja brzine i ubrzanja pojedinih osi, postavljaju softverska ograničenja hoda osi, navode korištene HAL datoteke, unose postavke glavnih modula LinuxCNC-a, te podešavaju svi ostali parametri. Neke od sekcija u ovoj datoteci su sljedeće:

- [EMC] – navođenje verzije konfiguracije i naziva stroja
- [DISPLAY] – postavke grafičkog sučelja
- [TASK] – postavke EMCTASK modula
- [HAL] – navođenje HAL datoteka koje se učitavaju pri pokretanju sustava
- [KINS] – definiranje broja zglobova stroja (broj motora)
- [EMCIO] – postavke EMCIO modula
- [TRAJ] – parametri za planer trajektorije (naziv osi, mjerne jedinice, limit brzine)
- [AXIS_B] – zadavanje hoda, te maksimalnog ubrzanja i brzine za prvu os
- [AXIS_C] – zadavanje hoda, te maksimalnog ubrzanja i brzine za drugu os
- [JOINT_0] – postavljanje limita pozicije i pojačanja PID regulatora prve osi
- [JOINT_1] – postavljanje limita pozicije i pojačanja PID regulatora druge osi
- [ECAT] – unošenje indeksa (adresa) uređaja u EtherCAT mreži

7.1.2. Konfiguracijska datoteka za EtherCAT

Konfiguracijska datoteka za EtherCAT napisana je u XML formatu. U njoj je za svaki podređeni uređaj u EtherCAT mreži definiran skup poruka koje se izmjenjuju između njega i nadređenog uređaja (NC računala). Svaki uređaj u mreži posjeduje vlastiti rječnik objekata u kojem su navedeni svi parametri koji mu se mogu mijenjati i procesni podaci koje može izmjenjivati s ostalim uređajima u mreži. Komunikacija se (prema CANOpen protokolu) odvija izmjenom SDO i PDO poruka.

SDO poruke se koriste za početno postavljanje parametara regulatora, šalju se uvijek u smjeru od nadređenog prema podređenim uređajima, te je potrebna potvrda s podređenog uređaja o zaprimanju poruke. Servo regulator pri svakom pokretanju sustava zaprima SDO poruke u kojima se nalaze vrijednosti unaprijed određenih parametara koje je potrebno podesiti. Parametri regulatora koji se podešavaju na ovaj način su:

- Parametri I/O za CN1 konektor
- Parametri referenciranja
- Parametri skaliranja

Ulazni signali su pridruženi spojnim točkama (pinovima) na konektoru CN1 preko parametara Pn509 i Pn510. Parametrom Pn511 je to isto postignuto za izlazne signale. Na primjer, signal za dozvolu gibanja (S-ON) je preko parametra Pn509 pridružen pinu 15, a signal za uključivanje kočnice (BK) je preko parametra Pn511 pridružen pinu 11 konektora CN1. Moguće je još i invertirati smjer signala promjenom parametara Pn516 i Pn517 za ulazne signale i Pn528 za izlazne.

Parametrima referenciranja (Pn6098, Pn6099 i Pn609A) određuje se način referenciranja stroja, te se definiraju brzine i ubrzanja kod traženja graničnih prekidača i indeksnog signala enkodera. U trenutnoj konfiguraciji stroja početno referenciranje nije potrebno provoditi jer se koriste apsolutni enkoderi.

Parametrima skaliranja se interne mjerne jedinice servo regulatora pretvaraju u željene mjerne jedinice koje određuje korisnik. Skaliranje se provodi za poziciju, brzinu i ubrzanje, a to se vrši promjenom parametara Pn6093, Pn6094 i Pn6097.

PDO poruke predstavljaju brži vid komunikacije i pomoću njih se izmjenjuju vremenski osjetljivi podaci kao što su zadavanje naredbi pozicije i brzine ili slanje izmjerenih opterećenja sa senzora sile. Jednom uspostavljena PDO komunikacija se odvija automatski, a prije toga je potrebno provesti postupak PDO mapiranja.

PDO mapiranje podrazumijeva slaganje objekata iz rječnika objekata zajedno u PDO registre kako bi se zajedno mogli slati kroz mrežu. PDO registri dijele se na one u kojima se podaci šalju prema podređenim uređajima (u rječniku objekata označeni indeksima 1600, 1601, 1602 i 1603) i na one iz kojih se podaci primaju (objekti 1A00, 1A01, 1A02 i 1A03). Primjerice, naredbe za zadavanje pozicije prvom regulatoru (objekt 607A) i za zadavanje brzine drugom regulatoru (objekt 60FF) mapiraju se u isti PDO registar (npr. 1601).

U poruke koje se šalju iz NC računala prema servo regulatorima mapiraju se:

- kontrolna riječ
- naredba za postavljanje načina rada regulatora
- naredba za zadavanje momenta
- naredba za zadavanje brzine
- naredba za zadavanje pozicije

U poruke koje se šalju od servo regulatora do NC računala mapiraju se:

- statusna riječ
- povratni podaci o momentu (struji)
- povratni podaci o brzini
- povratni podaci o poziciji

Svaka od prethodno navedenih naredbi ima vlastito mjesto u rječniku objekata, a koji se može pronaći u EtherCAT priručniku proizvođača servo regulatora [32].

Kontrolna riječ je dugačka 16 bitova, a koristi ju nadređeni uređaj kako bi upravljao stanjima u kojima se nalazi servo regulator. Servo regulator općenito može biti u tri stanja – omogućenog i prekinutog dovoda energije, te stanju greške. Interna stanja servo regulatora opisana su protokolom CiA-402. Nadređeni uređaj preko statusne riječi može saznati trenutno stanje servo regulatora.

Naredbom za postavljanje načina rada moguće je prebacivati servo regulator iz upravljanja po brzini u upravljanje po poziciji ili momentu. Korišteni regulatori nalaze se u načinu upravljanja po brzini jer je izlaz iz PID regulatora petlje po poziciji (zatvorene u LinuxCNC-u) referenca brzine koja se šalje do servo regulatora.

7.1.3. HAL datoteka

HAL datotekom opisuje se struktura HAL prostora. U početnoj fazi testiranja sustava neće biti učitani EMCOT modul LinuxCNC-a što znači da se referenca pozicije neće zadavati preko naredbi G koda, već će se zadavati ručnim postavljanjem vrijednosti signala (odskočna funkcija) ili slati sa signal generatora. Minimalna konfiguracija potrebna za ispitivanje odziva sustava zahtijeva učitavanje sljedećih HAL komponenata:

- generatora signala
- PID regulatora
- EtherCAT drivera za LinuxCNC

EtherCAT driver se nakon učitavanja ponaša kao komponenta u HAL prostoru koja ima onoliko izvodnica koliko ima definiranih SDO i PDO poruka u XML datotekama podređenih uređaja. Primjerice za tro-osni stroj se može pretpostaviti da će sigurno tri signala biti referenca pozicije koja će se slati do regulatora osi i tri signala dozvole gibanja (u ovom primjeru to predstavlja šest izvodnica na EtherCAT driveru). Drugim riječima, driver predstavlja sučelje preko kojeg se podaci izmjenjuju između LinuxCNC-a kao nadređenog uređaja i servo regulatora (tj. modula EK1100 i senzora sile) kao podređenih uređaja.

PID regulator je komponenta kojom je u HAL prostoru zatvorena regulacijska petlja po poziciji. U trenutku kad je sustav u potpunosti funkcionalan, ulaz u PID regulator je interpolirana naredba pozicije koja dolazi iz EMCOT modula. Za vrijeme testiranja, naredba pozicije šalje se iz softverskog generatora signala (*siggen* HAL komponenta). U PID regulator se još dovodi i stvarna vrijednost pozicije s enkodera. Oduzimanjem zadane i ostvarene pozicije dobiva se greška slijeđenja, koju regulator nastoji ukloniti izdavanjem referentne vrijednosti brzine sebi podređenom regulatoru po brzini (koji je izveden hardverski u servo regulatoru). Kod podešavanja brzinske petlje se na izvodnicu EtherCAT drivera, izravno iz generatora signala, šalje naredba brzine.

Sustav pri pokretanju učitava jednu INI datoteku i jednu glavnu HAL datoteku. Glavna HAL datoteka sastoji se od pojedinačnih HAL datoteka u kojima se odvojeno učitavaju različiti dijelovi HAL prostora. Tako se u jednoj HAL datoteci učitavaju PID regulatori (njihov broj odgovara broju osi) i bazni period izvršavanja (HAL komponenta *threads*), u drugoj EtherCAT driver, a u trećoj i četvrtoj su dodijeljeni HAL signali, koji se koriste za upravljanje prvom i drugom osi, izvodnicama EtherCAT drivera.

7.2. Podešavanje regulacijskih petlji i testiranje sustava

U ovom poglavlju opisan je postupak ispitivanja gibanja prigona rotacijskih osi. Slika 7-1. prikazuje servomotore spojene na pripadajuće reduktore. Enkoderski kablovi od motora spojeni su na CN2 priključak servo regulatora, napajanja motora spojena su na priključke U, V i W, a kočnice su spojene na napajanje od 24 V DC. Cilj je bio ispitati karakteristike odziva motora na zadane reference brzine i pozicije, te podesiti regulacijska pojačanja pojedinih osi kako bi se postigao brz i točan odziv sa što manjom regulacijskom pogreškom.

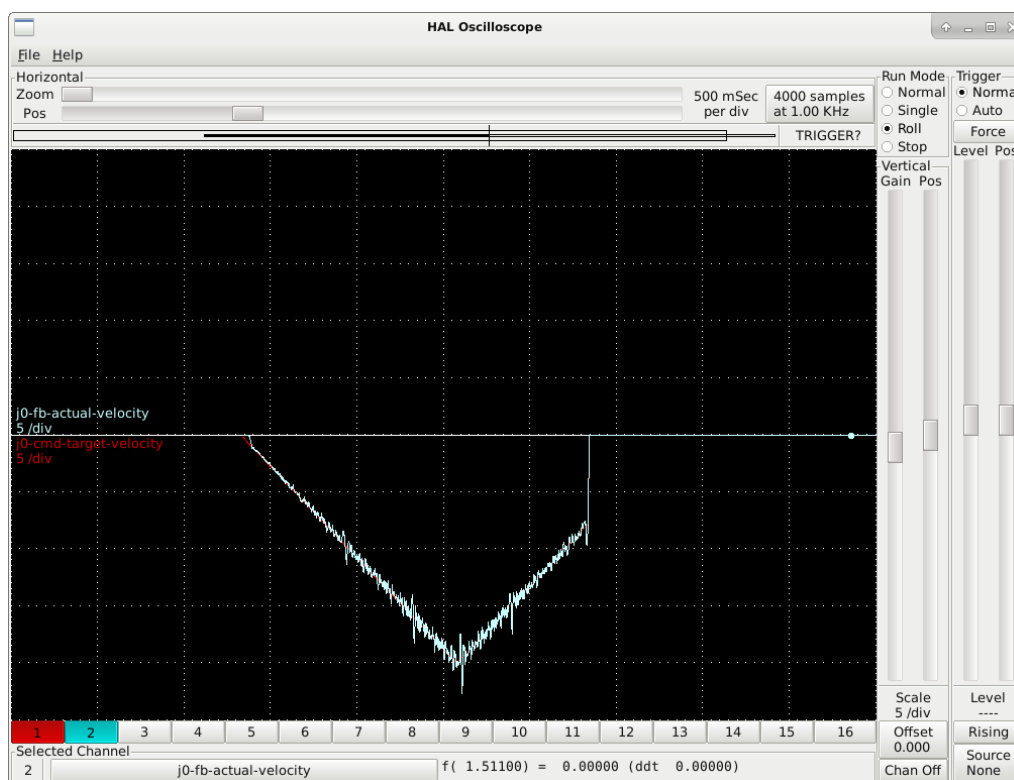


Slika 7-1. Testiranje posmičnih prigona

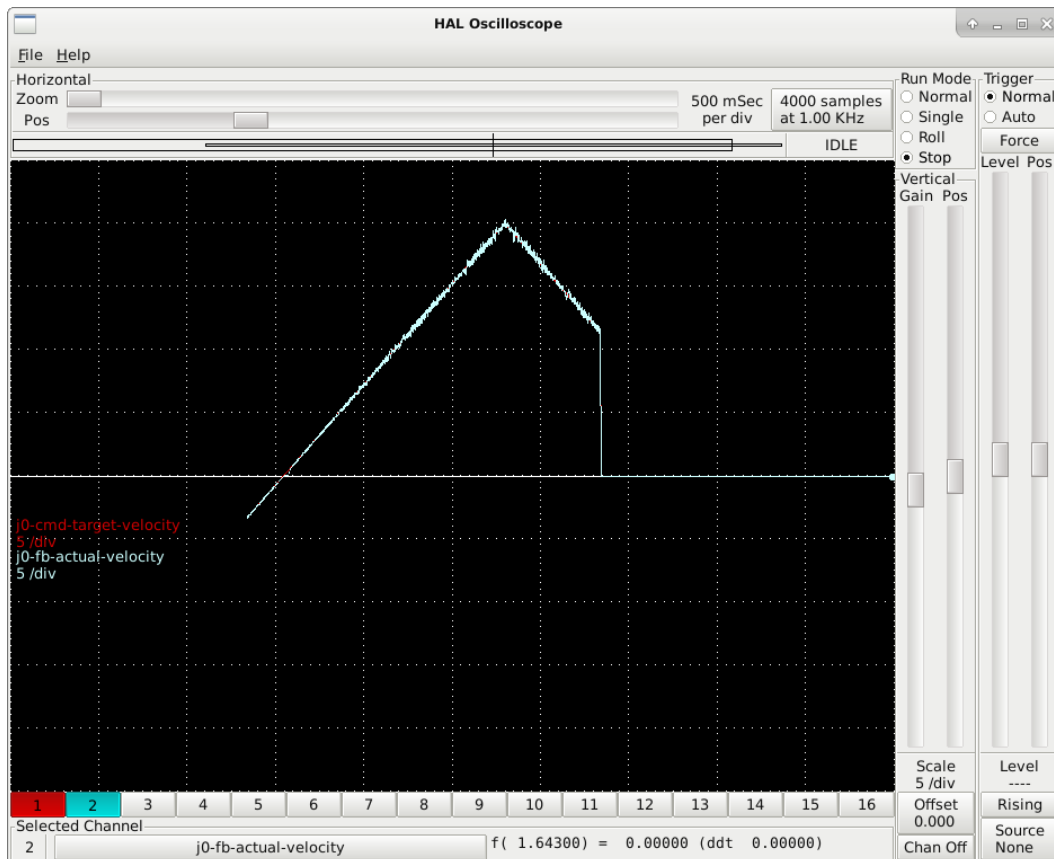
Ispitivanje je provedeno pomoću softverskog generatora signala i HAL osciloskopa u kojemu su promatrani referentni signali, izmjerene vrijednosti s enkodera i greška između njih. Za referencu brzine korišten je trokutasti signal, a za referencu pozicije sinusni signal. Općenito je bolje koristiti signale sa što blažom promjenom smjera kako bi se izbjegla udarna opterećenja na motor. Promjenom parametara generatora signala se signalima zadaju željena frekvencija i amplituda. Korištena su dva generatora signala – po jedan za svaku os. Smjer pojedinih signala prethodno je definiran u HAL datoteci.

Prvo je bilo potrebno podesiti parametre pojačanja regulacijskih petlji po brzini. Ona je zatvorena unutar servo regulatora pa se promjena parametara regulatora vrši slanjem SDO poruka. Alternativna opcija, koja je korištena ovdje, je ta da se regulator postavi u automatski način podešavanja parametara. U tom načinu rada servo regulator sam mijenja iznose proporcionalnog pojačanja i integracijske konstante na temelju vrijednosti parametra Pn100 i Pn101. Parametrom Pn100.0 definiraju se karakteristike inercije opterećenja, a parametrom Pn100.1 postavke za automatsko podešavanje. Krutost stroja postavlja se parametrom Pn101. Iterativnim mijenjanjem navedenih parametara postignute su zadovoljavajuće karakteristike odziva za obje osi (slika 7-3. i slika 7-6.) na zadane reference brzine. Na slikama 7-2. i 7-5. primjetna su značajna odstupanja signala stvarne i referentne brzine zbog neodgovarajućih parametara regulatora.

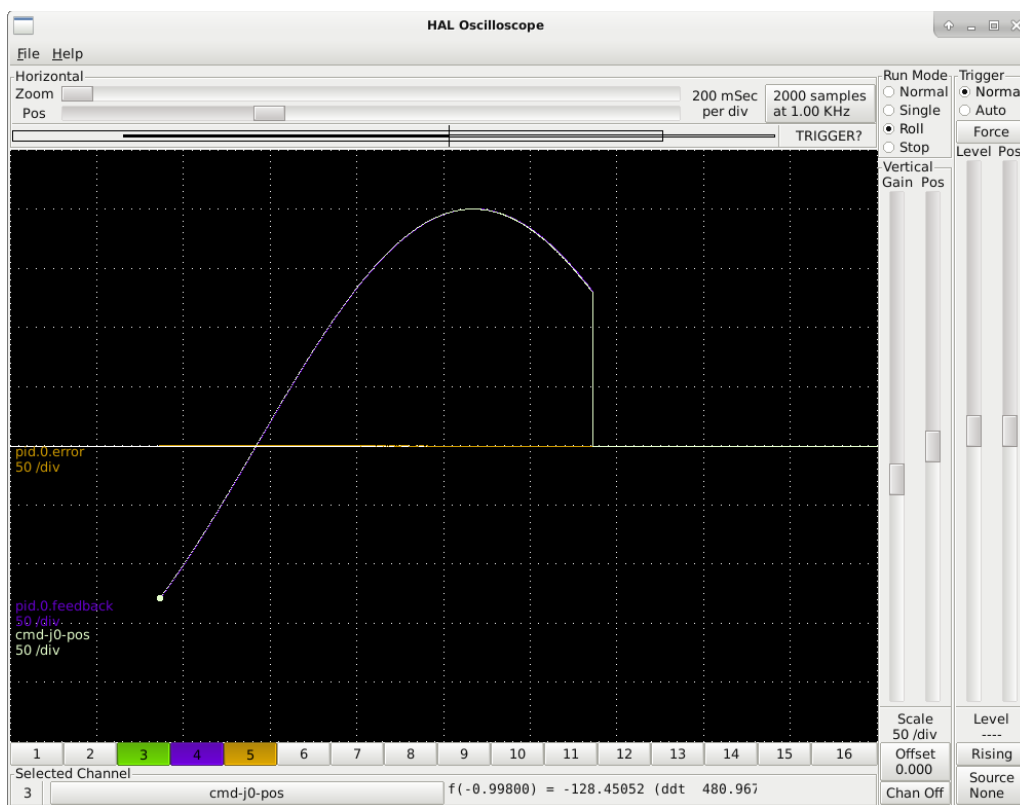
Zatim se pristupilo određivanju pojačanja regulacijskih petlji po poziciji. One su zatvorene u HAL prostoru LinuxCNC-a preko PID komponente. U slučaju da se servo regulator prebaci u način rada po poziciji (tad mu se ne bi iz LinuxCNC-a slala referenca brzine, već pozicije), parametre bi mu bilo moguće namjestiti također automatski. Ovako se to podesilo ručno iterativnim postavljanjem proporcionalnog i unaprijednog pojačanja dok se nisu postigle željene karakteristike odziva (slika 7-4. i 7-8.). Slika 7-7. prikazuje regulacijsko odstupanje po poziciji druge osi prije uvođenja unaprijednog pojačanja koje je grešku uklonilo (slika 7-8.)



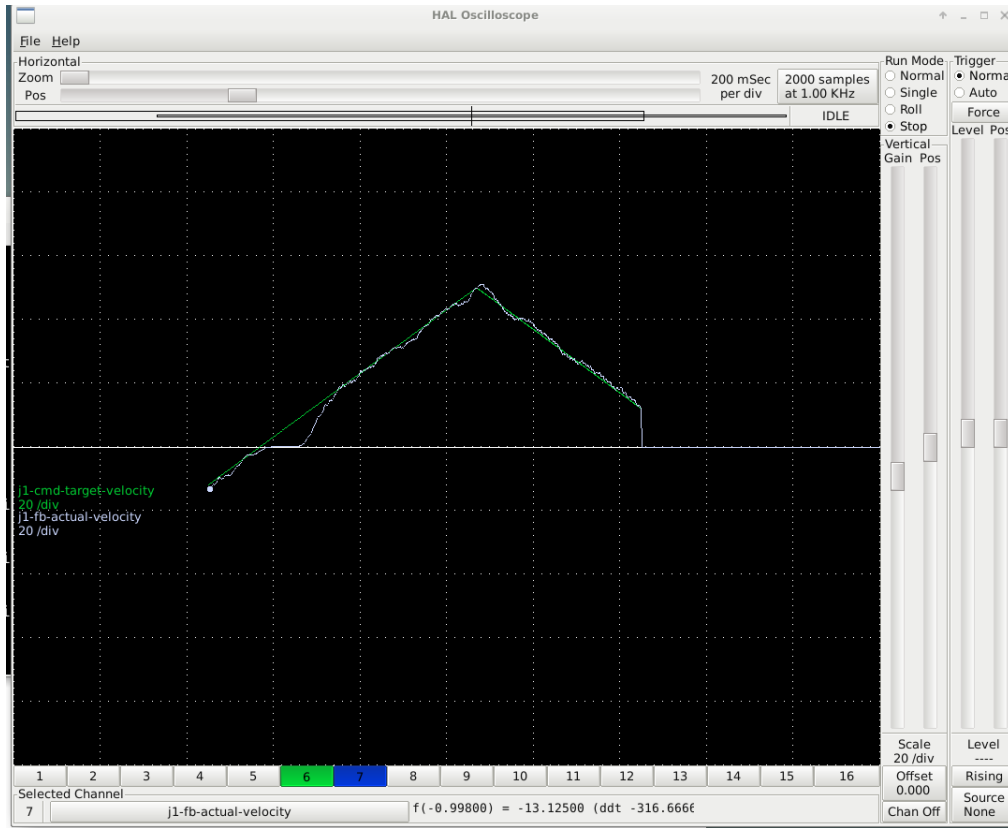
Slika 7-2. Prva os - odziv na referencu brzine prije podešavanja



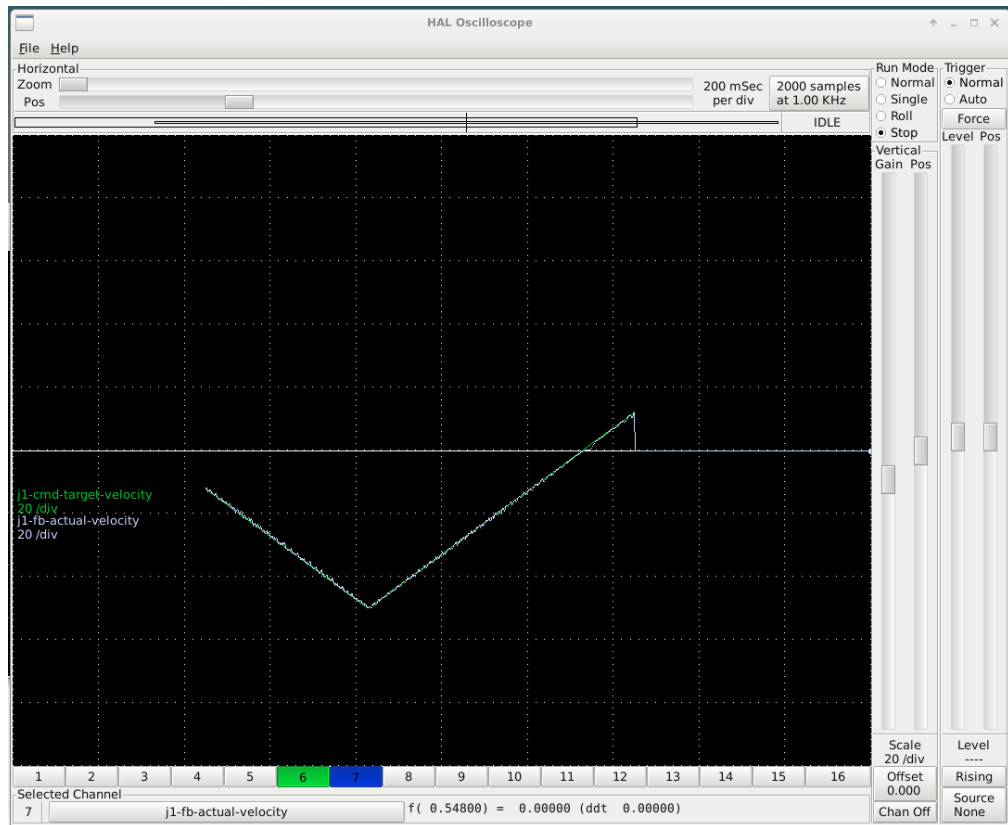
Slika 7-3. Prva os – odziv na referencu brzine nakon podešavanja



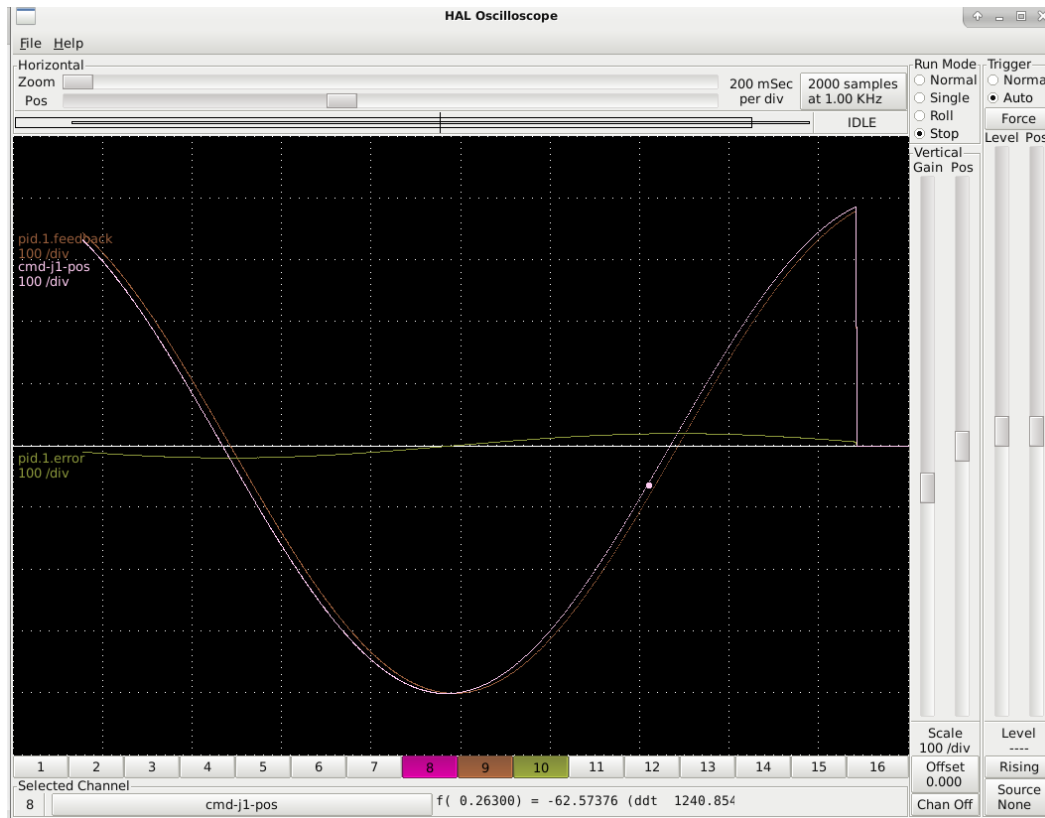
Slika 7-4. Prva os – odziv na referencu pozicije



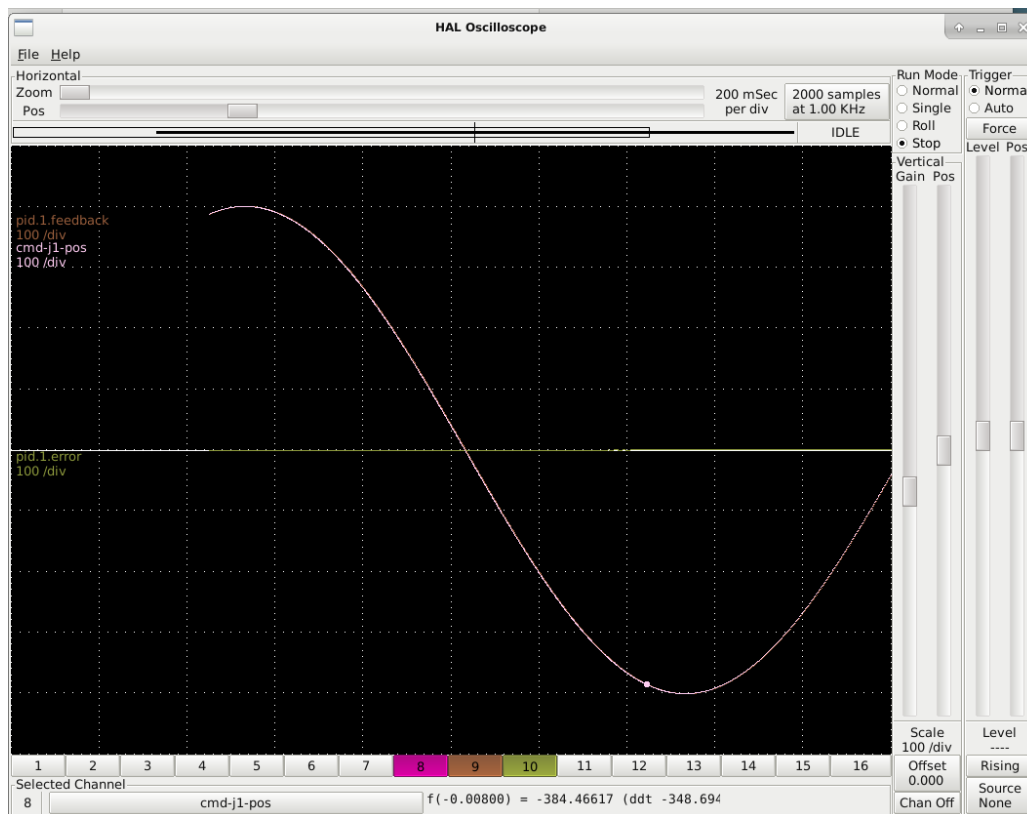
Slika 7-5. Druga os – odziv na referencu brzine prije podešavanja



Slika 7-6. Druga os – odziv na referencu brzine nakon podešavanja



Slika 7-7. Druga os – odziv na referencu pozicije prije podešavanja



Slika 7-8. Druga os – odziv na referencu pozicije nakon podešavanja

8. ZAKLJUČAK

Glavni zadatak rada bio je razvoj prigona uređaja za precizno pozicioniranje modela letjelica u zračnom tunelu. Uređaj se sastoji od dva posmična rotacijska prigona sa servomotorima i 6-osne piezoelektrične mjerne ćelije koja se koristi za akviziciju mjernih signala opterećenja koja djeluju na model uslijed strujanja zraka kroz tunel. U cilju realizacije upravljačkog dijela prigona odabrani su svi elementi upravljačkog elektro-ormara prema karakteristikama pogonskih elemenata sustava i senzora sile. U programskom alatu EPLAN izrađene su električne spojne sheme distribucije izmjeničnog i istosmjernog napajanja, spajanja logičkih funkcija sustava, te je prikazana mrežna topologija kojom je ostvarena komunikacija između upravljačkog računala i njemu podređenih dijelova sustava. Nakon što je utvrđeno da su ispunjeni svi preduvjeti, pristupilo se izradi elektro-ormara. Ona je uključivala izradu prostornog rasporeda elemenata, pričvršćivanje din šina i kablskih kanala vijcima kroz prethodno izbušene provrte, montiranje elemenata na šine, te ožičenje elemenata.

U završnom dijelu rada opisana je konfiguracija sustava LinuxCNC korištena za testiranje prigona. Ona se sastoji od INI i HAL datoteka, te opisne XML datoteke potrebne za uspostavu EtherCAT komunikacije između upravljačkog računala i servo regulatora. Testiranje sustava provedeno je slanjem referentnih signala (brzine i pozicije) iz softverskog generatora signala, a odziv sustava u pozicijskoj i brzinskoj petlji promatran je u HAL osciloskopu. U osciloskopu su se također prikazale vrijednosti brzine i pozicije s enkodera. Prvo su za obje osi podešene regulacijske petlje brzine, koje su zatvorene u servo regulatorima. Njihovo podešavanje obavljeno je u automatskom načinu preko podešavanja parametra krutosti, čime su postignute zadovoljavajuće karakteristike odziva. Nakon brzinske petlje, pristupilo se namještanju parametara regulacijske petlje po brzini. Kod nje su se parametri unosili ručno jer je ista realizirana u HAL prostoru LinuxCNC-a. Za nju je također uspješno proveden izbor optimalnih iznosa pojačanja.

Daljnji razvoj uređaja bit će usmjeren na integraciju mjernog elementa prigona s ostatkom upravljačkog sustava, te na razvoj grafičkog korisničkog sučelja kako bi se osigurao intuitivan način upravljanja i nadzora različitih dijelova upravljačkog sustava izravno iz LinuxCNC-a.

LITERATURA

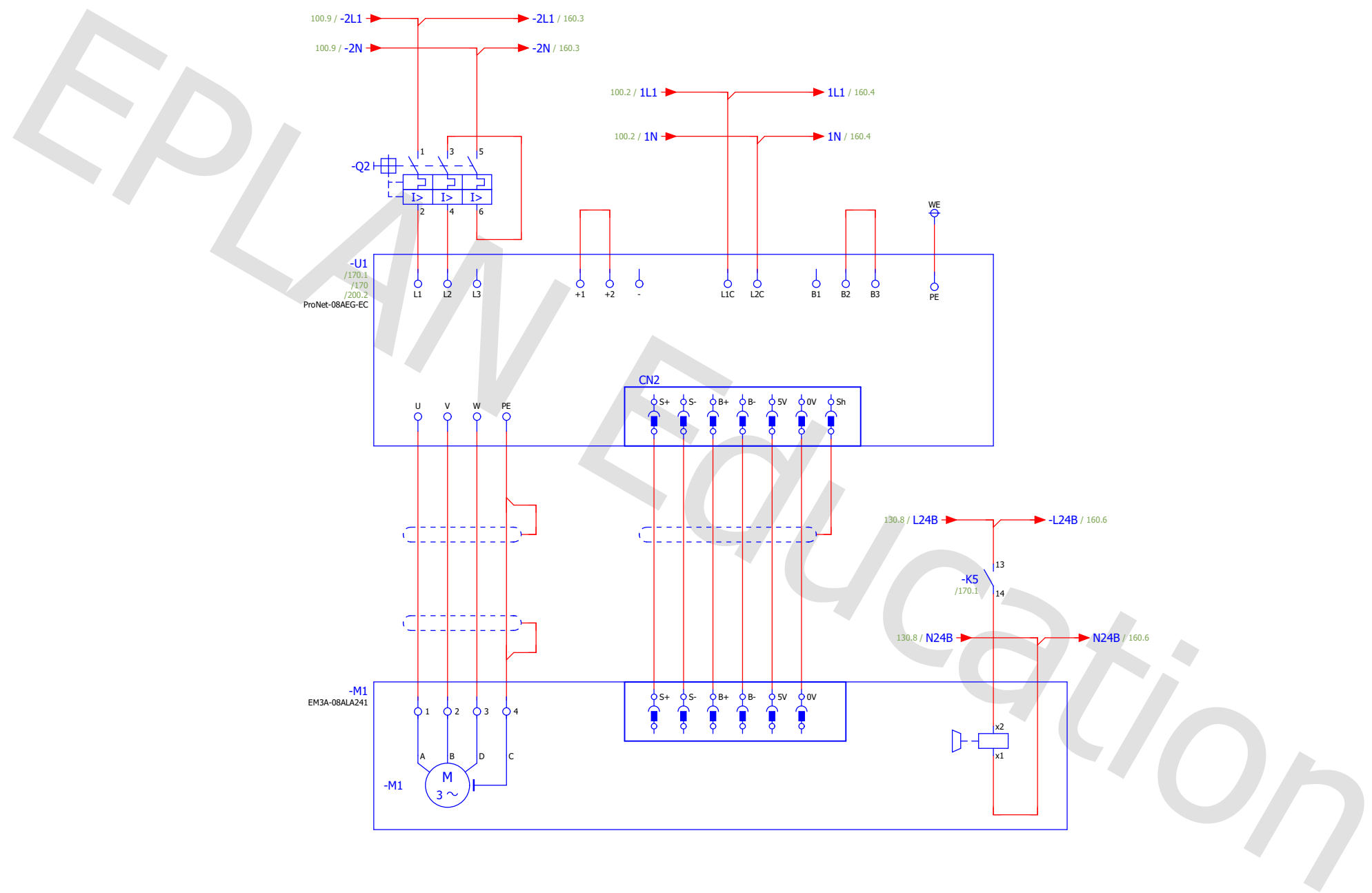
- [1] Suh S-H, Kang S-K, Chung D-H, Stroud I, Theory and Design of CNC Systems, Springer, London, 2008.
- [2] Pritschow G et al., Open Controller Architecture - Past, Present and Future, CIRP Annals - Manufacturing Technology, 50(2), 2001., p. 463-470
- [3] LinuxCNC.org, LinuxCNC Documentation v2.8., <https://www.linuxcnc.org/documents/>
- [4] Staroveški T, Brezak D i Udiljak T, LinuxCNC - The enhanced machine controller: application and an overview, Tehnički vjesnik, svez. 20, br. 6, 2013., p. 1103-1110
- [5] LinuxCNC Knowledge base, <http://wiki.linuxcnc.org/cgi-bin/wiki.pl>, pristupio 30.9. 2021.
- [6] Brezak D, Staroveški T, Regulacija obradnih strojeva, Fakultet strojarstva i brodogradnje, Zagreb, 2021.
- [7] Petrić J, Automatska regulacija: uvod u analizu i sintezu, Fakultet strojarstva i brodogradnje, Zagreb, 2012.
- [8] Zucker G, Dietrich D, Industrial Communication Systems: ISO/OSI Model, CRC Press, Boca Raton, 2011., p. 37-46
- [9] Beckhoff Automation, EtherCAT System Documentation, 2020., https://download.beckhoff.com/download/document/io/ethercat-terminals/ethercatsystem_en.pdf
- [10] Beckhoff Automation, EtherCAT – The Ethernet Fieldbus, 2020., https://www.beckhoff.com/media/downloads/informationsmedien/etg_brochure_en.pdf
- [11] HMS Networks, Industrial network market shares 2021 according to HMS Networks, <https://www.hms-networks.com/news-and-insights/news-from-hms/2021/03/31/continued-growth-for-industrial-networks-despite-pandemic>, pristupio 5.10.2021.
- [12] EtherCAT Technology Group, Layering with TSN and EtherCAT, 2017., https://www.ethercat.org/download/documents/Layering_with_TSN_and_EtherCAT.pdf
- [13] EtherCAT Technology Group, Slave Implementation Guide, 2018., https://www.ethercat.org/download/documents/ETG2200_V3i1i0_G_R_SlaveImplementationGuide.pdf

- [14] Beckhoff Automation, Hardware Data Sheet Section I, 2014., https://download.beckhoff.com/download/document/io/ethercat-development-products/ethercat_esc_datasheet_sec1_technology_2i2.pdf
- [15] Texas Instruments, EtherCAT protocol: C2000 real-time controller, <https://training.ti.com/ethercat-protocol-c2000-real-time-controller?context=1137766-1147233>, pristupio 10.10.2021.
- [16] Festo, SERCOS for Motor Controller, 2008., https://www.festo.com/net/SupportPortal/Files/324326/CMMP-AS-SC_2007-08_557362g1.pdf
- [17] Festo, CiA 402 for motor controller, https://www.festo.com/net/SupportPortal/Files/326671/CMMP-AS-M3_M0-C-CO_2013-04a_8022083g1.pdf
- [18] Beckhoff Automation, Beckhoff Information System: CANopen Introduction, https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_io_intro/1446518411.html&id=, pristupio 10.10.2021.
- [19] Pfeiffer O, Embedded Networking With CANopen, Circuit Cellar, <https://www.esacadeemy.com/en/library/technical-articles-and-documents/can-and-canopen/embedded-networking-with-canopen.html>, pristupio 10.10.2021.
- [20] Beckhoff Automation, Beckhoff Information System: EtherCAT Distributed Clocks, <https://infosys.beckhoff.com/english.php?content=../content/1033/ethercatsystem/2469118347.html&id=>, pristupio 15.10.2021.
- [21] Zelić F, Idejno konstrukcijsko rješenje uređaja za mjerenje aerodinamičnih opterećenja, Fakultet strojarstva i brodogradnje, Zagreb, 2020.
- [22] ESTUN, <https://www.estuneurope.eu/>, pristupio 1.11.2021.
- [23] Giga Precision, <https://www.gigager.net/>, pristupio 1.11.2021.
- [24] ATI Industrial Automation, F/T Sensor: Mini27 Titanium, https://www.ati-ia.com/products/ft/ft_models.aspx?id=Mini27+Titanium, pristupio 1.11.2021.
- [25] MEAN WELL, DIN Rail Power Supply, <https://www.meanwell.com/productPdf.aspx?i=136>, pristupio 5.11.2021.
- [26] Schneider Electric, <https://www.se.com/ww/en/>, pristupio 5.11.2021.
- [27] Schmersal, SRB-E-301MC, https://products.schmersal.com/en_CA/srb-e-301mc-4858.html, pristupio 8.11.2021.

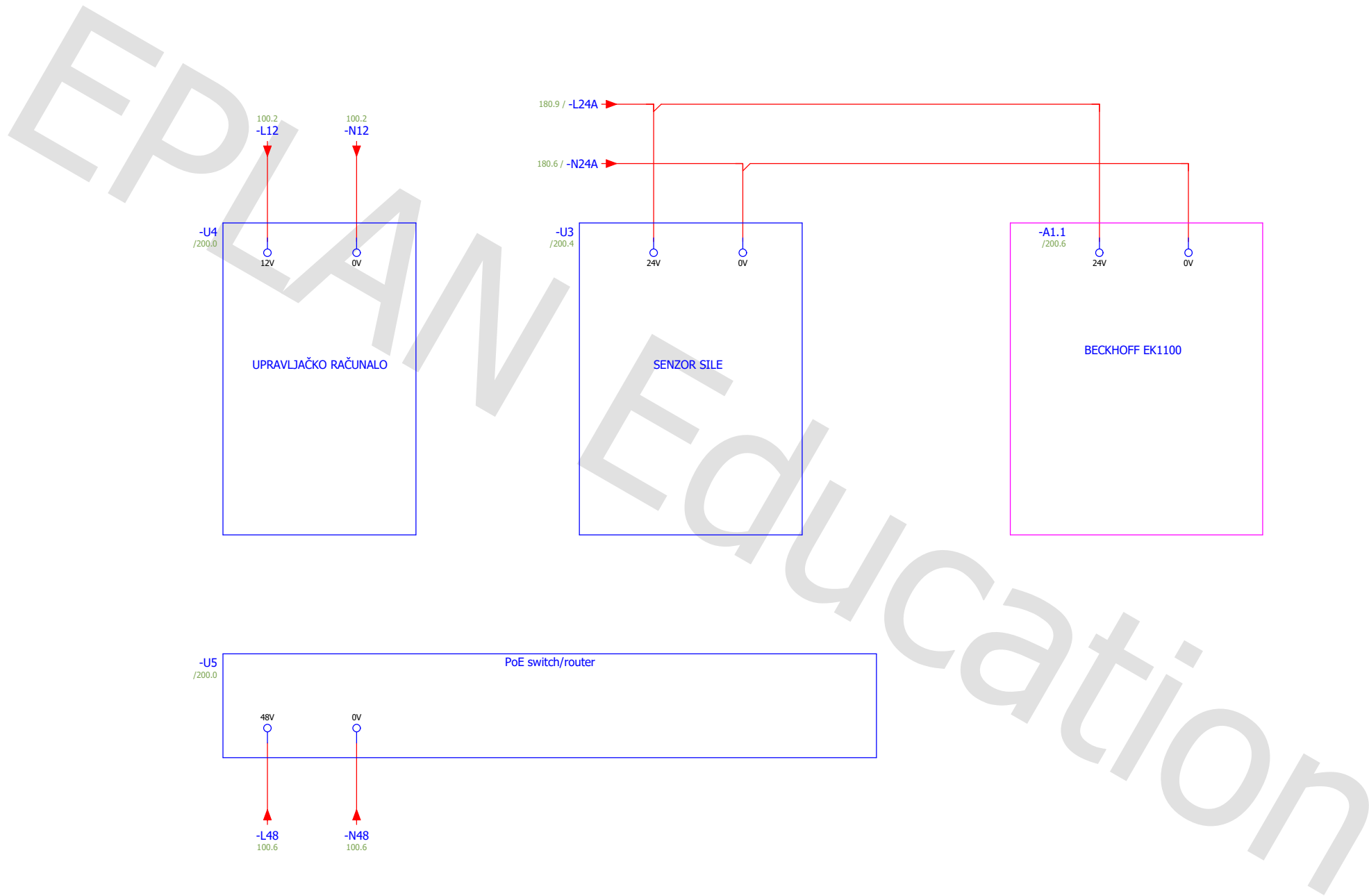
-
- [28] Phoenix Contact, <https://www.phoenixcontact.com/>, pristupio 8.11.2021.
- [29] Beckhoff Automation, EtherCAT terminals, <https://www.beckhoff.com/en-en/products/i-o/ethercat-terminals/>, pristupio 10.11.2021.
- [30] XCY Mini PC X30 J1900, http://www.cnxcy.com.cn/product/60667901618-822400949/XCY_fanless_mini_pc_computer_desktop_J4125_J1900_N3530_i3_i5_NUC_mini_pc_with_Dual_RS232_Dual_Lan_industrial_computer.html?spm=a2700.shop_index.82.5.66012568OrGr1d, pristupio 15.11.2021.
- [31] Tinkerforge, Building Blocks for professional Makers, <https://www.tinkerforge.com/en/home/how-it-works/>, pristupio 16.11.2021.
- [32] ESTUN Automation Technology, EtherCAT User's Manual V1.08, 2018.

PRILOZI

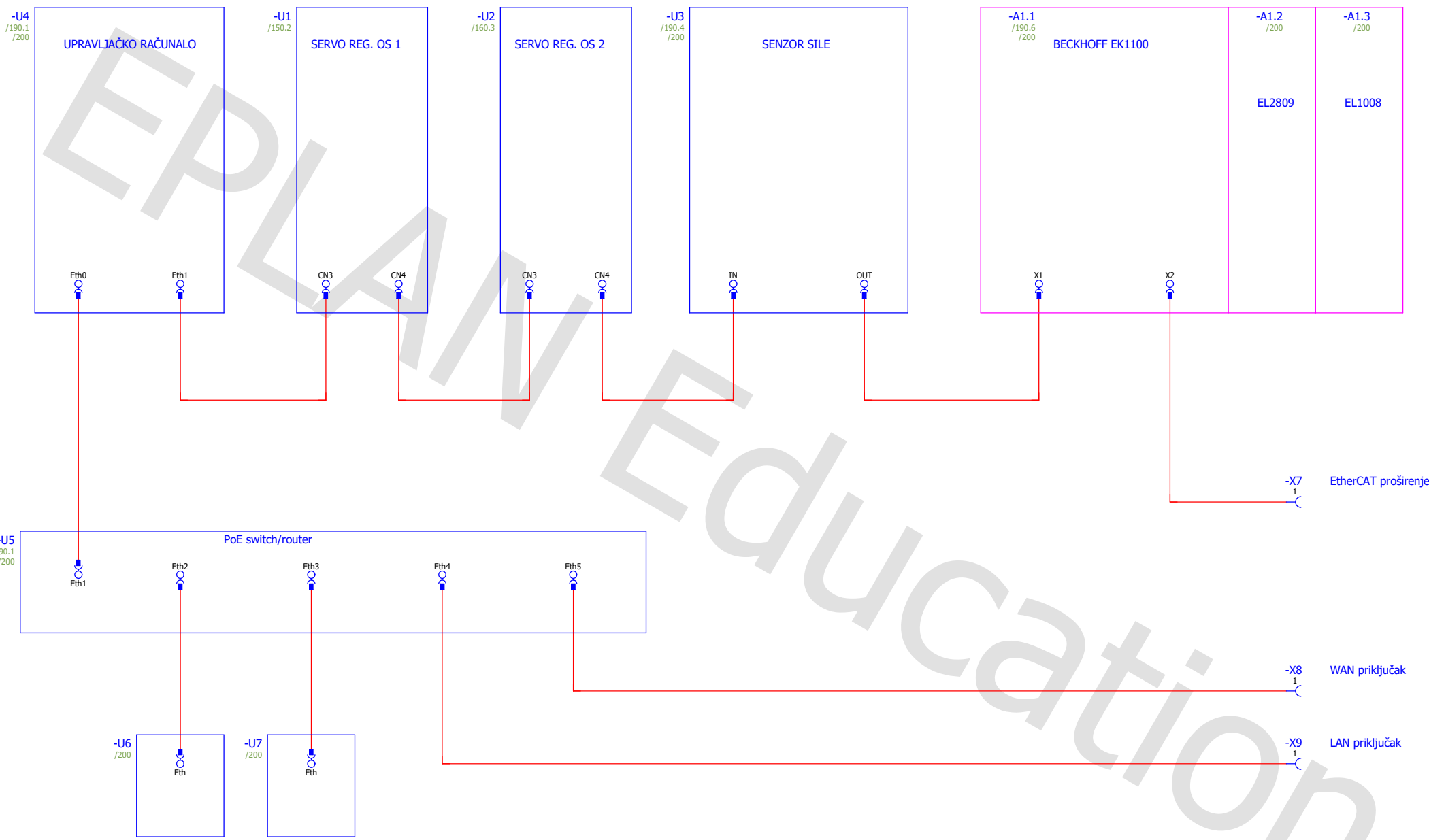
I. Električne spojne sheme



		Date	13/11/2021	EPLAN	EPLAN Software & Service	Servo regulator OS 1-Napajanje	= ESS	
		Ed.	Filip Zelić	Project template with identification structure in accordance with IEC standard: Page structure	GmbH & Co. KG	with higher-level function and mounting location	+ U01	
Modification	Date	Name	Original	Replacement of	Replaced by		IEC_tp001	
							Page	150
							Page	3 / 8



		Date	01/11/2021	EPLAN	EPLAN Software & Service GmbH & Co. KG	Napajanje CPU i IO modula	= ESS	
		Ed.	Filip Zelić	Project template with identification structure in accordance with IEC standard: Page structure	with higher-level function and mounting location		+ U01	
Modification	Date	Name	Original	Replacement of	Replaced by		IEC_tpl001	
							Page	190
							Page	7 / 8



		Date	01/12/2021	EPLAN	EPLAN Software & Service GmbH & Co. KG	Mrežna topologija	= ESS	
		Ed.	Filip Zelić	Project template with identification structure in accordance with IEC standard: Page structure with higher-level function and mounting location			+ U01	
Modification	Date	Name	Original	Replacement of	Replaced by		IEC_tp1001	Page 200
								Page 8 / 8