

Realizacija sustavlja upravljanja servomotorom pomoću Siemens PLC-a i HMI-a

Šokčić, Dario

Undergraduate thesis / Završni rad

2023

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:235:210742>

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

Download date / Datum preuzimanja: **2024-12-03**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Dario Šokčić

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Danijel Pavković, dipl.ing.

Student:

Dario Šokčić

Zagreb, 2023.

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

Zahvaljujem se mentoru prof. dr. sc. Danijelu Pavkoviću na prihaćanju ideje završnog rada te na pruženoj literaturi i pomoći.

Zahvaljujem se svojoj obitelji na podršci i strpljenju tijekom studiranja te prijateljima na pomoći u raznim teškim situacijama. Posebice se želim zahvaliti dugogodišnjem prijatelju i kolegi Mariju Jelašu na hvale vrijednim materijalima i zajedničkim učenjima kroz studij.

Posebno bih se za kraj zahvalio mentorima unutar tvrtke HUNOR d.o.o, Davidu Gyuriju i Denisu Petljaku na pruženoj šansi i povjerenju, odvojenom vremenu i strpljenju, zatim omogućenoj opremi te pruženim znanjima za izradu ovog završnog rada.

Dario Šokčić



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomске ispite
Povjerenstvo za završne i diplomске ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

| | |
|--|--------|
| Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje | |
| Datum | Prilog |
| Klasa: 602 - 04 / 23 - 6 / 1 | |
| Ur.broj: 15 - 1703 - 23 - | |

ZAVRŠNI ZADATAK

Student: **Dario Šokčić**

JMBAG: 0035222560

Naslov rada na hrvatskom jeziku: **Realizacija sustava upravljanja servomotorom pomoću Siemens PLC-a i HMI-a**

Naslov rada na engleskom jeziku: **Servomotor control system realization using Siemens PLC and HMI device**

Opis zadatka:

Elektromotorni servopogoni danas su dominantni u industriji zbog niza prednosti u odnosu na klasične elektromotorne pogone. Za implementaciju servopogona visokih performansi ključno je razraditi kvalitetnu programsku podršku, naročito na razini implementacije nadređenih sustava upravljanja (vođenja) pogona. Stoga je korisno razraditi i implementirati generički funkcijski blok u programabilnom logičkom kontroleru (engl. Programmable Logic Controller, PLC) opremljenom operatorskim panelom kao sučeljem prema korisniku (engl. Human-Machine Interface, HMI) za upravljanje osim servomotora koji će se moći koristiti univerzalno u raznim aplikacijama gdje je potrebno ostvariti upravljanje pozicijom, brzinom vrtnje i okretnim momentom (engl. motion control) radnog mehanizma.

U radu je potrebno napraviti sljedeće:

1. Ukratko opisati programabilni logički kontroler (PLC) i operatorski panel kao sučelje prema korisniku (HMI) proizvođača Siemens AG koji će se koristiti u realizaciji sustava vođenja servopogona, te također ukratko opisati programsko sučelje za razvoj korisničkog softvera.
2. Realizirati funkcijski blok korištenjem PLC Data Types u kojem će biti implementirano upravljanje servomotorom koristeći Motion Control funkcijske blokove prema PLCopen standardu.
3. Funkcijski blok treba implementirati sljedeće funkcionalnosti: omogućenje/onemogućenje rada osi, povratak na početnu poziciju uz limit momenta motora, modifikacija pozicije enkodera pomoću Motion Control blokova, parametriranje servo-osi (engl. axis tuning), podešavanje tehnoloških parametara objekta upravljanja (prijenosni omjeri, pozicijski limiti i brzinski limiti), parametriranje regulacijske petlje pozicije servomotora, ručni način rada preko HMI uređaja (engl. jog function), izrada recepata (upravljačkih sekvenci) i odgovarajućih funkcija za servo-pozicioniranje osi.
4. Ispitati funkcionalnost razvijene programske podrške na realnom industrijskom hardveru.

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

Zadatak zadan:

Datum predaje rada:

Predviđeni datumi obrane:

30. 11. 2022.

1. rok: 20. 2. 2023.
2. rok (izvanredni): 10. 7. 2023.
3. rok: 18. 9. 2023.

1. rok: 27. 2. - 3. 3. 2023.
2. rok (izvanredni): 14. 7. 2023.
3. rok: 25. 9. - 29. 9. 2023.

Zadatak zadao:

Predsjednik Povjerenstva:

Izv. prof. dr. sc. Danijel Pavković

Prof. dr. sc. Branke Bauer

Danijel Pavković

Branke Bauer

SADRŽAJ

| | |
|--|------|
| SADRŽAJ | I |
| POPIS SLIKA | III |
| POPIS TABLICA..... | VI |
| POPIS OZNAKA | VII |
| SAŽETAK..... | VIII |
| SUMMARY | IX |
| 1. UVOD..... | 1 |
| 2. KORIŠTENA OPREMA | 5 |
| 2.1. SIEMENS PLC S7-1513-1 PN | 5 |
| 2.2. SIEMENS HMI MTP1500 Unified Comfort..... | 7 |
| 2.3. SIMOTICS S-1FL6024-2AF21-1AA1 - servo motor..... | 9 |
| 2.4. Regulator – SINAMICS V90 | 13 |
| 2.4.1. Regulacija struje armature | 15 |
| 2.4.2. Regulacija brzine..... | 17 |
| 2.4.3. Regulacija pozicije..... | 18 |
| 3. FUNKCIJSKI BLOK ZA UPRAVLJANJE SERVOMOTOROM UNUTAR SIEMENS TIA PORTAL-A..... | 19 |
| 3.1. Konfiguracija uređaja u SIEMENS TIA Portal-u | 20 |
| 3.2. User Defined Types – UDT | 22 |
| 3.3. Tehnološki objekt upravljanja (eng. Technology objects)..... | 24 |
| 3.3.1. Motion Control funkcijski blokovi prema PLCopen standardu..... | 24 |
| 3.4. Funkcijski blok SERVOMOTOR FB | 25 |
| 3.4.1. Omogućenje/onemogućenje rada osi | 28 |
| 3.4.2. Podešavanje tehnoloških parametara objekta upravljanja - prijenosni omjeri, pozicijski i brzinski limiti | 30 |
| 3.4.3. Ručni način rada preko HMI uređaja (eng. jog function)..... | 32 |
| 3.4.4. Parametriranje servo-osi (eng. axis tuning) – strujna i brzinska petlja..... | 35 |
| 3.4.5. Parametriranje regulacijske petlje pozicije servo motora | 48 |
| 3.4.6. Izrada upravljačke sekvence za servo-pozicioniranje osi | 58 |
| 3.4.7. Povratak na početnu poziciju uz limit momenta servo motora..... | 65 |
| 4. ZAKLJUČAK..... | 69 |

LITERATURA..... 70

POPIS SLIKA

| | | |
|-----------|--|----|
| Slika 1. | Servo motor tvrtke SIEMENS..... | 3 |
| Slika 2. | SIEMENS PLC S7-1513-1 PN..... | 6 |
| Slika 3. | Osnovne karakteristike centralnih procesorskih jedinica, preuzeto sa [3] | 6 |
| Slika 4. | SIEMENS HMI MTP1500 Unified Comfort panel | 8 |
| Slika 5. | System Properties panela SIEMENS MTP1500 Unified Comfort | 8 |
| Slika 6. | Primjer pločice na servo motoru, preuzeto sa [7]..... | 9 |
| Slika 7. | SIMOTICS S-1FL6024-2AF21-1AA1 servo motor | 11 |
| Slika 8. | Mehanička karakteristika modela 1FL6024-2AF21-1AA1, preuzeto sa [6]..... | 12 |
| Slika 9. | Pojednostavljena opća blokovska shema regulacije IM, preuzeto sa [1] | 14 |
| Slika 10. | SINAMICS V90 | 15 |
| Slika 11. | Odziv strujne regulacijske petlje, preuzeto sa [1] | 17 |
| Slika 12. | Odziv motora na regulaciju brzine vrtnje, preuzeto sa [1]..... | 18 |
| Slika 13. | Uređaji umreženi Profinet protokolom..... | 20 |
| Slika 14. | Adrese ulaznog i izlaznog modula | 20 |
| Slika 15. | Prikaz u „Device view“ nakon dodavanja ulaznog i izlaznog modula..... | 21 |
| Slika 16. | Ladder logika za sigurnosnu gljivu i relej | 21 |
| Slika 17. | Sigurnosna gljiva (eng. Emergency Stop button)..... | 22 |
| Slika 18. | UDT_AppAxis_TO | 23 |
| Slika 19. | Podatkovni blok AppAxis_TO_DB za upravljanje radom servo motora..... | 23 |
| Slika 20. | Odabir standardnog telegrama 102 za razmjenu podataka s regulatorom..... | 24 |
| Slika 21. | Motion Control funkcijski blokovi prema PLCopen standardu | 25 |
| Slika 22. | Ulazi i izlazi u funkcijskom bloku „SERVOMOTOR FB“ | 26 |
| Slika 23. | Svi „Network“ unutar funkcijskog bloka „SERVOMOTOR FB“..... | 27 |
| Slika 24. | Instanca funkcijskog bloka „SERVOMOTOR FB“ u „Main“ | 28 |
| Slika 25. | Motion Control blok „MC_POWER“ | 29 |
| Slika 26. | Switch na HMI za omogućenje rada tehnološkog objekta upravljanja..... | 29 |
| Slika 27. | Prijenosni omjer | 30 |
| Slika 28. | Pozicijski limiti..... | 31 |
| Slika 29. | Brzinski limiti..... | 32 |
| Slika 30. | Motion Control blok „MC_MOVEJOG“ | 33 |
| Slika 31. | Odabir načina rada pomoću sklopke | 34 |
| Slika 32. | Sučelje HMI-a za ručno upravljanje servo motorom | 34 |

| | | |
|-----------|--|----|
| Slika 33. | „One button auto tuning“ unutar V-ASSISTANT | 36 |
| Slika 34. | Generator brzine | 37 |
| Slika 35. | Prikaz rada generatora brzine (1) | 38 |
| Slika 36. | Prikaz rada generatora brzine (2) | 39 |
| Slika 37. | Prikaz rada generatora brzine (3) | 40 |
| Slika 38. | Network 10: MOVE VELOCITY | 40 |
| Slika 39. | Parametri prilikom podešavanja regulacijske petlje brzine vrtnje | 41 |
| Slika 40. | STEP pobuda | 42 |
| Slika 41. | Pojačanja za slučaj odziva na [Slika 40] | 42 |
| Slika 42. | Proklizavanje servo motora | 43 |
| Slika 43. | Pojačanja za slučaj odziva na [Slika 42] | 43 |
| Slika 44. | Nestabilan odziv na pobudu | 44 |
| Slika 45. | Pojačanja za slučaj odziva na [Slika 44] | 44 |
| Slika 46. | Utjecaj povećanja integralnog pojačanja | 45 |
| Slika 47. | Pojačanja za slučaj odziva na [Slika 46] | 45 |
| Slika 48. | Utjecaj povećanja proporcionalnog djelovanja | 46 |
| Slika 49. | Pojačanja za slučaj odziva na [Slika 48] | 46 |
| Slika 50. | Zadovoljavajući odziv na postavljenu referencu brzine vrtnje | 47 |
| Slika 51. | Pojačanja za slučaj odziva na [Slika 50] | 47 |
| Slika 52. | Generator pozicije – ladder logika | 48 |
| Slika 53. | Prikaz rada generatora pozicije (1) | 49 |
| Slika 54. | Prikaz rada generatora pozicije (2) | 50 |
| Slika 55. | #MC_MOVERELATIVE potreban za generator pozicije | 51 |
| Slika 56. | Parametri prilikom podešavanja regulacijske petlje pozicije | 52 |
| Slika 57. | Sučelje za podešavanje regulacijske petlje pozicije | 53 |
| Slika 58. | Odziv servo motora na referencu pozicije (1) | 54 |
| Slika 59. | Odziv servo motora na referencu pozicije (2) | 54 |
| Slika 60. | Odziv servo motora na referencu pozicije (3) | 55 |
| Slika 61. | Odziv servo motora na referencu pozicije (4) – prihvatljiv odziv | 56 |
| Slika 62. | Odziv servo motora na referencu pozicije (5) | 56 |
| Slika 63. | Odziv servo motora na referencu pozicije (6) – prihvatljiv odziv | 57 |
| Slika 64. | SEQUENCER_FB: Network1 | 58 |
| Slika 65. | SEQUENCER_FB: Network2 | 59 |
| Slika 66. | SEQUENCER_FB: Network2 – bliži prikaz (1) | 59 |

| | | |
|-----------|---|----|
| Slika 67. | SEQUENCER_FB: Network2 – bliži prikaz (2)..... | 60 |
| Slika 68. | SEQUENCER_FB: Network3 | 61 |
| Slika 69. | SEQUENCER_FB: Network4 | 61 |
| Slika 70. | SEQUENCER_FB funkcijski blok pozvan u SERVOMOTOR FB | 62 |
| Slika 71. | UDT_Servo_Positioning | 63 |
| Slika 72. | UDT_STEP | 63 |
| Slika 73. | Servo_Positioning_DB podatkovni blok..... | 63 |
| Slika 74. | „Faceplate“ na HMI uređaju..... | 64 |
| Slika 75. | MC_TORQUELIMITING | 66 |
| Slika 76. | MC_HALT | 67 |
| Slika 77. | MC_HOME..... | 68 |

POPIS TABLICA

Tablica 1. Opis oznaka u oznaci servo motora S-1FL6024-2AF21-1AA1 10

POPIS OZNAKA

| Oznaka | Jedinica | Opis |
|------------------|-----------------|--------------------------|
| SH | mm | duljina vratila |
| M _N | Nm | nazivni moment |
| n _N | o/min | nazivna brzina vrtnje |
| n _{MAX} | o/min | maksimalna brzina vrtnje |
| t | s | vrijeme |

SAŽETAK

U ovom završnom radu projektiran je sustav upravljanja pozicijom, brzinom i okretnim momentom servomotora pomoću Siemens TIA Portal koristeći Siemens PLC (programabilni logički kontroler) opremljen HMI operatorskim panelom kao sučeljem prema korisniku. U prvom dijelu su detaljnije opisani programabilni logički kontroler, operatorski panel i servomotor, tvrtke Siemens AG i općenito funkcija regulatora u jednom takvom sustavu. U nastavku su dani uvidi u principe izrade funkcijskog bloka za upravljanje servomotorom koristeći Motion Control funkcijske blokove u kojem je implementirano nekoliko funkcionalnosti poput parametriranja servo-osi (eng. axis tuning) što uključuje razumijevanje reuglacijskih petlji struje, brzine i pozicije te je iz tog razloga ukratko opisan i taj dio.

Za kraj je funkcijski blok ispitan na realnoj industrijskoj opremi omogućenoj od tvrtke HUNOR d.o.o. kako bi se potvrdila funkcionalnost istog.

Ključne riječi: servomotor, funkcijski blok, sustav upravljanja, Siemens, TIA Portal, PLC, HMI, PLC Open

SUMMARY

In this undergraduate thesis a servo motor position, speed and torque control system was made using Siemens TIA Portal using a Siemens PLC (Programmable Logic Controller) equipped with an HMI operator panel as a user interface. In the first part, the programmable logic controller, operator panel and servo motor, from the Siemens AG company, and the general function of the regulator in such a system are described in more detail. After that, there are insights into the principles of creating a function block for controlling a servo motor using Motion Control function blocks, in which several functionalities are implemented, such as axis tuning, which includes understanding the regulation loops of current, speed and position regulation, and because of that, it is also briefly described.

Finally, the functional block was tested on real industrial equipment provided by HUNOR d.o.o. to confirm its functionality.

Key words: servo motor, function block, control system, Siemens, TIA Portal, PLC, HMI, PLC
open

1. UVOD

Svakodnevnim napretkom tehnologije povećava se i potražnja za kvalitetnim industrijskim rješenjima kako bi se povećala efikasnost u proizvodnji, a time i profit koji donosi mogućnost daljnjeg razvitka. Dok jeftiniji roboti koriste koračne ili istosmjerne motore, naprednija robotika zahtijeva upotrebu servo motora radi niza prednosti nad koračnim motorima.

Povijest izraza „servo motor“ seže u 19. stoljeće, a odnosio se na hidraulične i parne strojeve za kormilarenje brodovima. Prema [1] servo mehanizam je sustav za ostvarenje mehaničkog gibanja u regulacijskoj petlji s negativnom povratnom vezom. Navedena definicija obično vrijedi za sustave u kojima je regulirana veličina, odnosno izlaz, mehanička pozicija (kut zakreta) ili jedna od njezinih derivacija (brzina ili ubrzanje). Servo motor konačni je upravljački element servo mehanizma, ali pritom se ne može promatrati odvojeno, već uvijek u kontekstu nekog pogonskog sustava. [1] S povratnom informacijom o položaju ili brzini pomoću enkodera ili resolvera, služi za izrazito brzo i točno upravljanje pozicijom, brzinom i okretnim momentom (uz dodatno mjerenje struje armature), što ih čini idealnim za primjene gdje su točnost i ponovljivost ključni faktori za funkcioniranje samog procesa. Dizajnirani su za pružanje velikog okretnog momenta pri svim brzinama, što ih čini prikladnima za primjene koje zahtijevaju veliki moment pokretanja, a i za pomicanje tereta pri velikim brzinama. Također, imaju vrlo brzo vrijeme odziva, što ih čini idealnim za primjene koje zahtijevaju rapidno ubrzanje. Isto tako, oni mogu raditi u širokom području brzina, od vrlo niskih do vrlo visokih, bez gubitka točnosti ili preciznosti.

Servo pogoni zahtjevniji su dio širokog opsega pogona s promjenjivom brzinom, kod kojih u određenim aplikacijama postoji potreba za promjenom brzine, ali poznavanje njezine stvarne vrijednosti nije nužno, stoga povratna veza nije potrebna. Prema [1] opće su značajke pogona s promjenjivom brzinom, uključujući i servo pogone:

- energetska učinkovitost,
- meko pokretanje, manji mehanički stres (ovdje su iznimka visokodinamički servo pogoni),
- povećani stres za izolaciju, a zbog visokih i naglih porasta napona koji su posljedica napajanja preko frekvencijskog pretvarača,
- nužnost prisilnog hlađenja kod niskih brzina (hlađenje putem ventilatora na osovini motora nije dostatno),

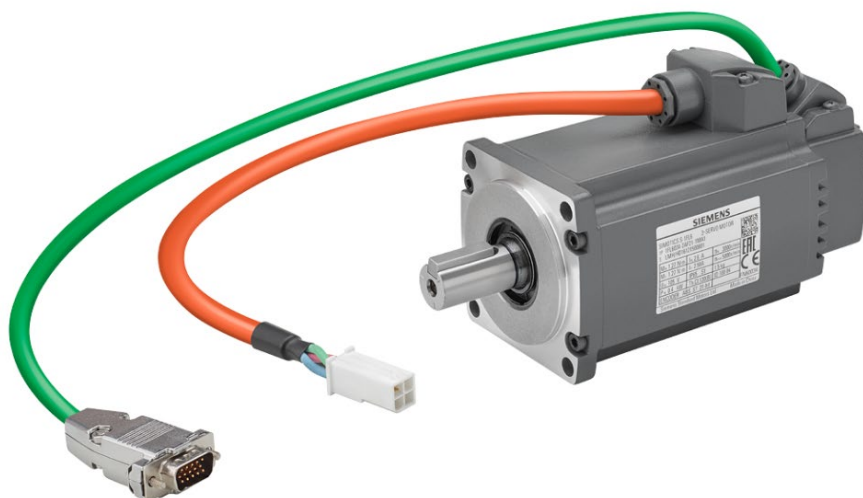
- nužnost posebnih konstrukcijskih izvedbi za smanjenje gubitaka koje uzrokuju viši harmonici struja i napona,
- problematika odabira regulacijske strukture i podešavanja regulacijskih parametara zbog promjena parametara stroja,
- vrlo često nužnost montiranja mehaničkih mjerača kuta zakreta.

Zahtjevi prema suvremenim servo motorima još su i stroži:

- preciznost pozicioniranja (kuta zakreta rotora),
- preciznost podešavanja brzine,
- široko područje regulacije brzine,
- stabilnost i niska valovitost momenta,
- mogućnost preopterećenja,
- visoka dinamika.

Gospodarsku važnost područja elektromotornih pogona istaknulo je njemačko udruženje elektrotehničke industrije s podacima iz 2017. godine gdje je iskazan promet u vrijednosti od 9,8 milijardi eura, a servo motor je u središtu takvog jednog pogona. Valja istaknuti kako opće uporabljiv motor, koji bi odgovarao svim primjenama i aplikacijama, ne postoji. Najrašireniji tip motora danas je asinkroni, a uobičajena mu je primjena u pokretanju ventilatora, crpki, industrijskih pogona itd. Kolektorski (komutatorski) strojevi (s četkicama) svoju primjenu imaju u aplikacijama manjih snaga, npr. kućanski aparati ili ručni alati, a prisutni su i dalje i u automobilskoj industriji (npr. brisači i motori za podizanje prozora). Istosmjerni strojevi bez četkica (komutatora) koriste se u manjim pogonima visoke učinkovitosti i/ili visoke dinamike te i u drugim aplikacijama, npr. za pokretanje malih ventilatora. Sinkroni strojevi s permanentnim magnetima (Brushless AC, poznatiji još i kao Permanent-Magnet Synchronous Machine ili PMSM), sa sinusnim induciranim naponom, isto tako se koriste u visokodinamičnim servo pogonima. I za kraj, koračni motori koji se koriste za jednostavnije primjene, recimo jednostavno pozicioniranje tereta malih snaga. [1]

Najčešće primjene servo motora su u robotici za preciznu kontrolu kretnji zglobova robota, u CNC (eng. Computer Numerical Control) strojevima za precizno pozicioniranje i kontrolu kretanja alata za rezanje, u medicinskoj opremi za preciznu kontrolu medicinskih uređaja, kao što su kirurški roboti te u industrijskim aplikacijama za npr. transportne sustave i montažne linije. Na [Slika 1] prikazan je jedan servo motor s ugrađenim enkoderom.



Slika 1. Servo motor tvrtke SIEMENS

Današnja razina industrijske automatizacije ne bi bila moguća bez ključnog izuma zvanog programabilni logički kontroler koji je omogućio revoluciju u automatizaciji industrijskih procesa. PLC je industrijsko računalo koje je ojačano i prilagođeno industrijskim uvjetima u kojima se provodi regulacija i upravljanje proizvodnih procesa, kao što su proizvodne linije, strojevi, robotski uređaji ili bilo koja aktivnost koja zahtijeva visoku pouzdanost, jednostavnost programiranja i dijagnostiku grešaka u procesu. PLC zaprima ulaze od ulaznih uređaja, a zatim procijenjuje te ulaze u odnosu na pohranjenu programsku logiku i generira izlaze za upravljanje perifernim izlaznim uređajima. Primjer je ožičenog sustava u stvarnom vremenu jer izlazi moraju proizlaziti kao odgovor na ulazne uvjete unutar ograničenog vremena, inače može doći do neželjene operacije. [2] Standardni programabilni logički kontroler se sastoji od centralne procesorske jedinice, izvora napajanja, memorije, komunikacijskih modula i odgovarajuće sklopove za pravilan rad s I/O podacima.

Kako bismo razumjeli zbog čega je doprinio takvoj revoluciji, kratko ćemo dati uvid u probleme na koje su nailazili inženjeri tadašnjeg doba bez programabilnog logičkog kontrolera, s obzirom da nisu imali izbora nego koristiti se s nefleksibilnim i skupim upravljačkim sustavom. Naime, jedini način na koji se moglo upravljati radom motora u nekom postrojenju bili su releji. Releji rade korištenjem zavojnice koja, kad je pod naponom, stvara magnetsku silu za promjenu stanja prekidača u stanje ON ili OFF, a kada ostane bez napona, vraća se u

početno stanje. Na primjer, ako je zadatak bio upravljati radom motora, mogao se staviti relej između izvora napajanja i motora te napajanjem istoga kontrolirati rad motora. S obzirom da je u tvornicama bilo puno više od samo jednog motora, trebalo je dakle i dosta releja što je dovelo do velikih električnih „kabineta“ punih releja. No, bili su potrebni i kontrolni releji koji su upravljali ranije navedenim relejima koji upravljaju prekidačem koji uključuje i isključuje motor. Sve u svemu, logistički je to bio veliki problem iz razloga što je svaki od tih releja trebao biti ožičen specifičnim poretkom kako bi radilo na željeni način, a u slučaju kvara jednog releja, cijeli sustav ne bi radio. Otkrivanje problema uzimalo bi sate i dane, a s obzirom da se radi o mehaničkom rješenju, kontakti u releju bili su podložni i trošenju. Nadalje, u slučaju kada su inženjeri htjeli napraviti neku nadogradnju ili promjenu, bilo je potrebno ispočetka raditi cijeli sustav, što je danas nezamisliva radnja.

Ideja za programabilnim logičkim kontrolerom razvila se u danas globalnoj korporaciji General Motors. Zbog navedenih problema s relejima, koncept računalnog upravljanja dobio je svoje mjesto u raspravama korporacija poput GM. Tadašnji inženjeri u GM-u istaknuli su zahtjeve koje bi trebao imati prvi PLC, osim zahtjeva za zamjenom releja, a to su:

- sustav koji je fleksibilan kao računalo, ali cjenovno kompetitivan, odnosno konkurentan sličnom relejnom logičkom sustavu,
- jednostavan za održavanje i programiranje u već prihvaćenom relejnom „ladder logic“ načinu određivanja upravljačkog zakona,
- mora pouzdano raditi u industrijskom okruženju, punom nečistoća, vlage, elektromagnetizma i vibracija,
- mora biti prilagodljiv, odnosno modularnog oblika, kako bi se omogućila laka izmjena komponenti i proširivost.

Većina elektromehaničkih komponenti potrebnih za ožičeni relejni upravljački sustav su potpuno eliminirane, rezultirajući velikom uštedom prostora, zatim potrošnje snage i zahtjeva na održavanje takvog sustava.

Kako se funkcionalnost programabilnog logičkog kontrolera razvijala, uređaji za programiranje i komunikaciju također su doživjeli brzi rast. Prvi uređaji za programiranje bili su namjenski, ali nažalost veličine kofera. Kasnije su se pojavili ručni uređaji za programiranje, ali ubrzo su zamijenjeni odgovarajućim softverom na osobnom računalu.

U idućem poglavlju bit će prikazan primjer PLC-a u realnom okruženju koji je bio korišten za upravljanje s osi servo motora.

2. KORIŠTENA OPREMA

U ovom poglavlju pružit će se uvid i dati opis korištene opreme u realizaciji upravljanja servo motorom.

2.1. SIEMENS PLC S7-1513-1 PN

SIEMENS PLC S7-1513-1 PN je Siemens-ov modularni programabilni logički kontroler vrhunskih performansi. Ovaj model dio je popularne serije S7-1500 i namijenjen je za primjenu u širokom rasponu industrijskih okruženja. Posjeduje brzi procesor koji omogućuje visoku brzinu rješavanja programa i obradu podataka, veliku količinu radne memorije (RAM) i podržava veliki broj kako digitalnih, tako i analognih ulaza i izlaza, što omogućuje povezivanje s različitim senzorima, aktuatorima i drugim uređajima. „PN“ u nazivu modela označava da PLC podržava Profinet (PN) komunikacijski protokol. Profinet je standard u industriji za komunikaciju između uređaja, a omogućuje pouzdano i brzo dijeljenje podataka. Uz to, podržava i druge komunikacijske protokole kao što su Ethernet, TCP/IP, UDP i MPI protokol. Programiranje se vrši pomoću softvera posebno razvijenog za Siemens PLC uređaje naziva Siemens TIA Portal (STEP7). U navedenom softveru je omogućeno jednostavno programiranje i konfiguracija PLC-a, a podržava ladder logiku (LAD), strukturirani tekstualni tekst (STL), funkcionalni blok dijagram (FBD) i slično. Ovaj model pruža visok stupanj pouzdanosti i sigurnosti na način da podržava napredne funkcije za dijagnostiku i nadzor sustava kako bi se brzo otkrile i efikasno otklonile pogreške u radu. Naravno, ima i mogućnost proširivanja pomoću različitih dodatnih modula i kartica. Može se povezati s drugim uređajima poput HMI (eng. Human Machine Interface) panela, SCADA (eng. Supervisory Control and Data Acquisition) sustava i drugih vanjskih uređaja. Razmatrajući način napajanja i montaže, koristi istosmjerno napajanje od 24 V te se može montirati na standardne DIN šine što olakšava integraciju u industrijske ormariće. Nadalje, Siemens kao tvrtka koja se između svega ostalog posvetila i sigurnosti, omogućava zaštitu od neovlaštenog pristupa i manipulacije podacima. Stoga, PLC podržava enkripciju i mogućnost definiranja prava pristupa kako bi spriječio pristup neautoriziranim korisnicima.

Za kraj opisa korištenog PLC-a, valja istaknuti i dugoročno podršku i održavanje od strane Siemens-a. Poznati su po svojoj dugoročnoj podršci kroz ažuriranja softvera, tehničku podršku te dostupnost rezervnih dijelova. Na [Slika 2] vidljiv je SIEMENS PLC S7-1513-1 PN u okruženju gdje se koristio kroz izradu ovog završnog rada, zajedno s ulaznim i izlaznim modulima, dok su na [Slika 3] prikazane osnovne karakteristike centralnih procesorskih jedinica, a koje se nalaze u [3].



Slika 2. SIEMENS PLC S7-1513-1 PN

| Order no.: | 6ES7511-1AK00-0AB0 | 6ES7513-1AL00-0AB0 | 6ES7516-3AN00-0AB0 |
|-----------------------------------|----------------------|----------------------|------------------------------|
| Short description | CPU 1511-1 PN | CPU 1513-1 PN | CPU 1516-3 PN/DP |
| Supply voltage, permissible range | 20.4 VDC to 28.8 VDC | 20.4 VDC to 28.8 VDC | 20.4 VDC to 28.8 VDC |
| Number of blocks | 2000 | 2000 | 6000 |
| Data work memory | 1 MB | 1.5 MB | 5 MB |
| Code work memory | 150 KB | 300 KB | 1 MB |
| Interfaces | 1 x PROFINET | 1 x PROFINET | 2 x PROFINET 1 x PROFIBUS |
| Number of PROFINET ports | 2 | 2 | 3 |
| Web server supported | Yes | Yes | Yes |
| Isochronous operation supported | Yes | Yes | Yes |

Slika 3. Osnovne karakteristike centralnih procesorskih jedinica, preuzeto sa [3]

2.2. SIEMENS HMI MTP1500 Unified Comfort

HMI MTP1500 Unified Comfort panel je napredan HMI (eng. Human-Machine Interface) uređaj namijenjen industrijskim postrojenjima. Ovaj panel pruža integriranu kontrolu i nadzor nad različitim automatizacijskim sustavima kako bi omogućio optimalno upravljanje procesima. Ističe se visokokvalitetnim LCD TFT zaslonom osjetljivim na dodir dijagonale od 15,6 " i rezolucije 1366 x 768 piksela, što omogućuje jasno prikazivanje informacija o sustavu kojeg se upravlja. Zbog velike dijagonale zaslona panela, masa iznosi 4,3 kg. Veliki zaslon i raspon boja do 16,7 milijuna uvelike korisnicima olakšava nadzor različitih parametara sustava te vizualizaciju u realnom vremenu. Kontrola svjetline pomoću panela ide od 10 do 100%, a putem programa od 0 do 100%, no sve vrijednosti ispod 10% se postavljaju se na vrijednost 10%. Broj radnih sati panela, odnosno životni vijek, na 50% svjetline iznosi 50000 sati. Navedeno se u korisničkom priručniku [4] može pronaći pod MTBF koji predstavlja radne sate nakon kojih se maksimalna svjetlina prepolovi u odnosu na zadanu vrijednost.

Korišteni moderni panel napaja se s istosmjernim naponom od 24 V (19,2 – 28,8 V), a potrošnja snage iznosi 16,8 W. Pokreće ga snažan procesor koji osigurava brzu obradu podataka i brzi odziv na promjene koje korisnik naređuje. Također, sadrži 4 GB RAM memorije i 32 GB ROM memorije koja je proširiva sa SD karticom s čak dva dodatna utora. [4]

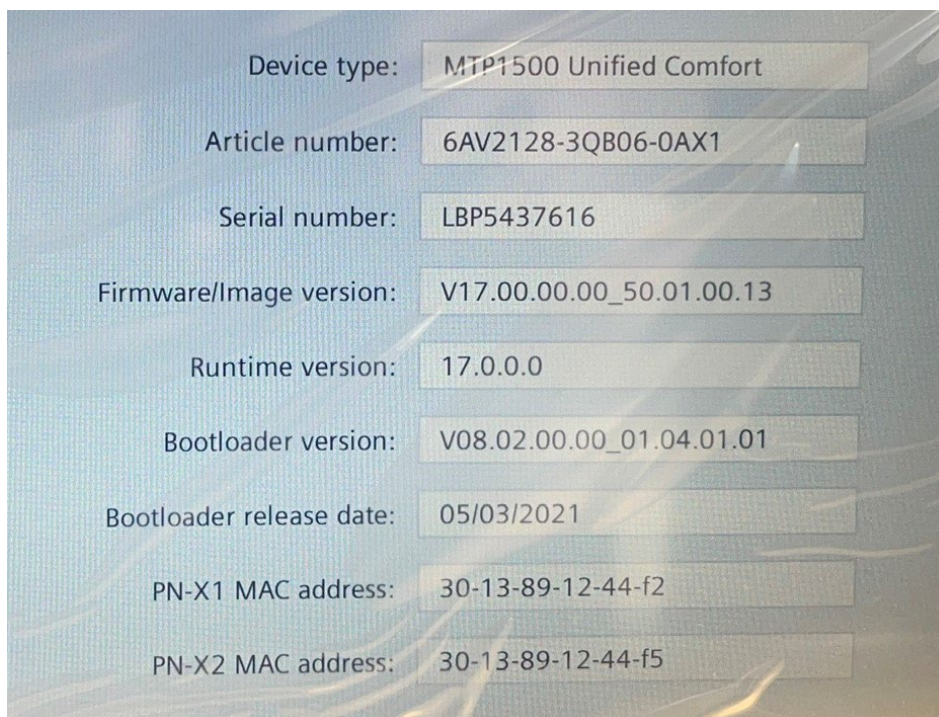
Nadalje, panel podržava različite komunikacijske protokole poput PROFINET (LAN) i Ethernet protokola, a sadrži dva RJ45 konektora s integriranim switch-om, jednim RJ45 konektorom koji daje brzinu prijenosa podataka do 1000 Mbit/s, zatim četiri USB 3.1 tip A konektora te za kraj jedan RS 422/485 konektor. [5]

Na [Slika 4] prikazan je početni zaslon opisanog HMI MTP1500 Unified Comfort panela u realnom okruženju pri izradi ovog završnog rada.



Slika 4. SIEMENS HMI MTP1500 Unified Comfort panel

Na [Slika 5] prikaze su bitne informacije HMI panela kao sustava do kojih se može doći unutar panela, poput broja artikla, serijskog broja i verzije softvera.



Slika 5. System Properties panela SIEMENS MTP1500 Unified Comfort

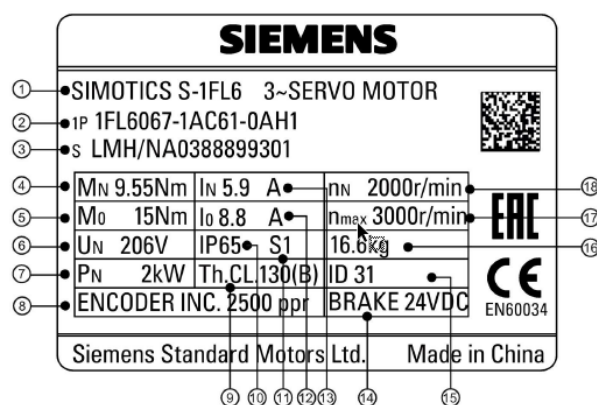
2.3. SIMOTICS S-1FL6024-2AF21-1AA1 - servo motor

Prema [6] SIMOTICS S-1FL6 servo motori su sinkroni motori s permanentim magnetima dizajnirani za operacije bez vanjskog hlađenja, a toplina se disipira putem vanjskog kućišta motora. Imaju mogućnost preopterećenja od 300 % te se mogu kombinirati sa SINAMICS V90 regulatorom za dobivanje moćnog servo sustava s visokim performansama. Benefiti ovakvih motora su sljedeći:

- materijal permanentnih magneta visokih performansi,
- robustan dizajn koji uključuje IP65 zaštitu za cijeli motor i konektore,
- mala valovitost momenta koja omogućuje rad motora bez mehaničkih vibracija,
- visoka nazivna brzina za određene motore,
- mogućnost značajnog ubrzanja zbog 300 % kapaciteta preopterećenja,
- visoka fleksibilnost odabira vrste motora s inkrementalnim/apsolutnim enkoderom i sa/bez kočnice.

Kako bi kranji korisnici mogli razumjeti oznaku servo motora, Siemens je u korisničkoj dokumentaciji dodao primjer jedne pločice koja se nalazi na svakom servo motoru. Na [Slika 6] se nalazi primjer pločice koja se nalazi na svakom servo motoru tvrtke Siemens, a koja objašnjava svaku veličinu i oznaku na pločici.

Motor rating plate (example)



| | | | | | |
|---|----------------|---|-----------------------------|---|---------------|
| ① | Motor type | ⑦ | Rated power | ⑬ | Rated current |
| ② | Article number | ⑧ | Encoder type and resolution | ⑭ | Holding brake |
| ③ | Serial number | ⑨ | Thermal class | ⑮ | Motor ID |
| ④ | Rated torque | ⑩ | Degree of protection | ⑯ | Weight |
| ⑤ | Stall torque | ⑪ | Motor operating mode | ⑰ | Maximum speed |
| ⑥ | Rated voltage | ⑫ | Stall current | ⑱ | Rated speed |

Slika 6. Primjer pločice na servo motoru, preuzeto sa [7]

S tim saznanjem, na [Slika 7] prikazan je model servo motora S-1FL6024-2AF21-1AA1 korištenog u ovom završnom radu. U korisničkoj dokumentaciji [7] može se pronaći i detaljan opis oznaka u broju artikla, a u nastavku će biti opisan za korišteni servomotor. U oznaci S-1FL6024-2AF21-1AA1, žuto podcrtane oznake objašnjene su u [Tablica 1]:

Tablica 1. Opis oznaka u oznaci servo motora S-1FL6024-2AF21-1AA1

| OZNAKA | ZNAČENJE |
|--------|--|
| 02 | oznaka za duljinu vratila (eng. Shaft Height, SH); u ovom konkretnom primjeru ta oznaka daje informaciju da je SH = 20 mm |
| 4 | oznaka za nazivni moment (eng. Rated Torque, M_N); u ovom konkretnom primjeru ta oznaka daje informaciju da je $M_N = 0,32$ Nm |
| 2 | informacija o vrsti inercije (eng. Inertia Type); u ovom konkretnom primjeru ta oznaka daje informaciju da se radi o motoru male inercije |
| F | informacija o nazivnoj brzini (eng. Rated Speed, n_N); u ovom konkretnom primjeru ta oznaka daje informaciju da je $n_N = 3000$ o/min |
| 2 | informacija o potrebnom napajanju (eng. Supply voltage); u ovom konkretnom primjeru ta oznaka daje informaciju da se motor napaja s 200 V |
| 1 | informacija o vrsti spajanja (eng. Connection type); u ovom konkretnom primjeru ta oznaka daje informaciju da motor ima izlaze za spajanje (eng. Cable Outlet) |
| A | informacija o tipu enkodera (eng. Encoder type); u ovom konkretnom primjeru ta oznaka daje informaciju da se radi o |

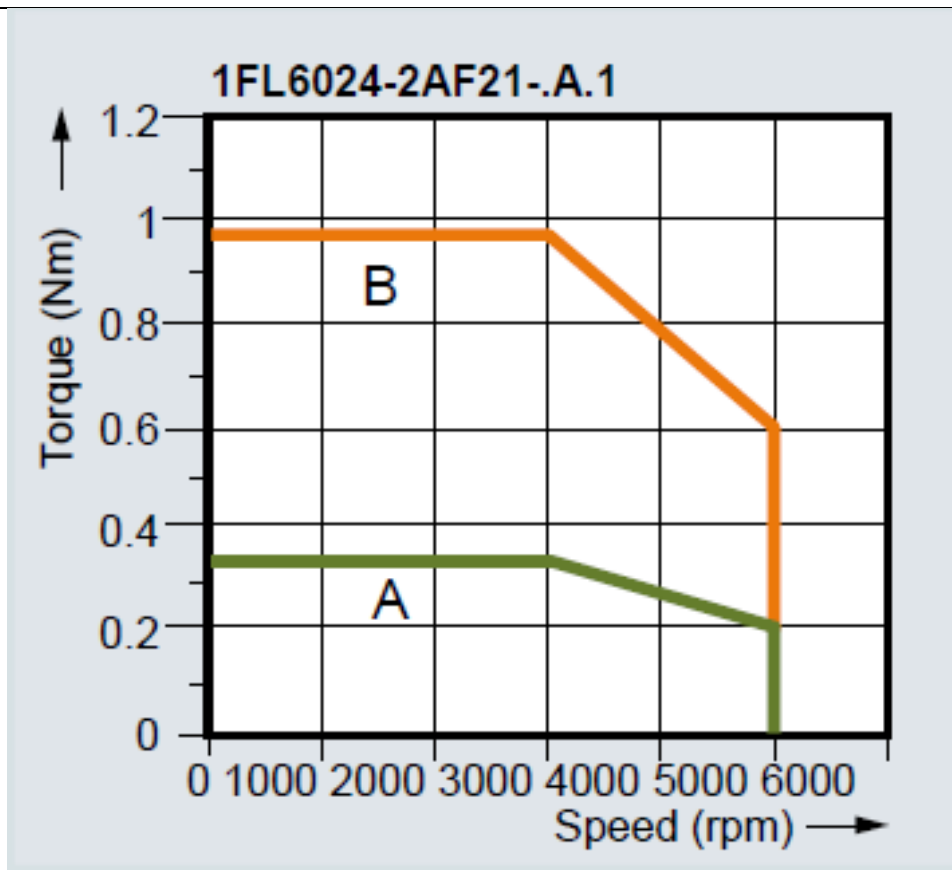
| | |
|---|---|
| | inkrementalnom TTL enkoderu s 2500 pulseva po okretaju |
| A | informacija o mehanici; u ovom konkretnom primjeru ta oznaka daje informaciju da servo motor koristi vratilo s perom, bez kočnice |



Slika 7. SIMOTICS S-1FL6024-2AF21-1AA1 servo motor

Vrijedi istaknuti još i kako je maksimalna brzina koju ovaj motor može dati $n_{MAX} = 5000$ o/min, no radi zaštite motora, preporuča se da po toj brzini ne radi dugotrajno, već je ta brzina za kratkotrajni rad.

U [6] se može pronaći i korisna mehanička karakteristika ovog modela servo motora, a vidljiva je na [Slika 8]. Zelena karakteristika pod slovom A je za dugotrajni rad iz koje je vidljivo kako servo motor daje nazivni moment sve do nazivne brzine za koju je predviđen dugotrajni rad, a nakon toga moment kojeg servo motor može dati počinje opadati. Narančasta karakteristika pod slovom B je za kratkotrajni rad iz koje je vidljivo kako servo motor u kratkotrajnom radu može dati i tri puta veći moment sve do nazivne brzine. Dakle, motor kratkotrajno odlazi u preopterećenje od 300 % te s obzirom da nije konstruiran za takav rad, preporuča se da što kraće bude u takvom maksimalnom preopterećenju. Pri većim brzinama zaliha momenta koju može osigurati brže opada u odnosu na zelenu karakteristiku.



Slika 8. Mehanička karakteristika modela 1FL6024-2AF21-1AA1, preuzeto sa [6]

2.4. Regulator – SINAMICS V90

Konačni cilj ovakvih pogona je regulacija zadanih referentnih veličina s namjerom postizanja određene dinamike. Obično se s tim misli na regulaciju brzine vrtnje ili zakretnog kuta, no pritom se dodatno moraju regulirati i neke druge veličine poput struje, a time i okretnog momenta na vratilu ukoliko se radi o konstantnoj uzbuđi. [1] Iako u svojoj strukturi korišteni regulator koristi napredne tehnike regulacije referentnih veličina, u ovom poglavlju će se objasniti kaskadna regulacijska struktura istosmjernog motora u kojoj se vanjske regulacijske petlje superponiraju na unutarnje, kako bi se dobio generalni uvid kako regulator zapravo regulira veličine. Regulacijski koncepti razvijeni za istosmjerni motor lako se mogu primjeniti i na sinkroni motor s permanentnim magnetima gdje je regulator po poziciji i brzini identičan, dok je regulacija struje nešto kompleksnija za izmjenične motore. Cilj regulacije kaskadnom strukturom je postizanje izvrsnih regulacijskih karakteristika kao što su stacionarna točnost, brzi, stabilni te dobro prigušeni odziv. Navedeno se postiže dinamičkim povratnim vezama pomoću senzora jer je mehaničke povratne veze, poput mehaničkih povratnih opruga, vrlo skupo za realizirati. Vrijedi napomenuti kako je u većini slučajeva brzina zadnja regulirana veličina, dok je u zahtjevnim sustavima za pozicioniranje (regulacija zakretnog kuta, npr. kod CNC strojeva ili robota) dodatno prisutna i nadređena petlja za regulaciju pozicije. [1]

S obzirom na svoju hijerarhijsku strukturu, svaka takva ima svoje prednosti, ali i određene nedostatke. Glavna prednost takve strukture je što je izrazito dobro i logično strukturirana, a može ju se pronaći i u nekim drugim sferama života. Jedna od ključnih vrijednosti kaskadne strukture je mogućnost limitiranja izlaznih veličina regulatora. Za primjer će se uzeti regulator brzine vrtnje čiji je izlaz referenca struje armature. Ukoliko je regulator struje dobro podešen, „ispoštovati“ će tu referencu struje, da bude što brži odziv s malim prebačajem. Dakle, ako regulator brzine vrtnje u međukoraku daje referencu struje, postoji mogućnost limitiranja te struje primjenom tzv. IF - THEN logike u digitalnim regulatorima. Na taj način se štiti motor i pretvarač od strujnog preopterećenja, odnosno limitira se struja motora na onu koju motor i pretvarač mogu podnijeti. Stoga, velika prednost kaskadne strukture je da se varijable stanja vide ne samo mjerenjem, nego su limitiranje i preko reference. Svaki nadređeni regulator zadaje referencu varijable stanja koju tada odrađuje podređeni regulator. Regulator pozicije zadaje referencu brzine, regulator brzine zadaje referencu struje, regulator struje zadaje referencu napona, a svaku od te četiri navedene moguće je i limitirati. Vrijedi istaknuti kako varijable stanja nisu linearno, već isključivo dinamički povezive, integriranjem ili deriviranjem.

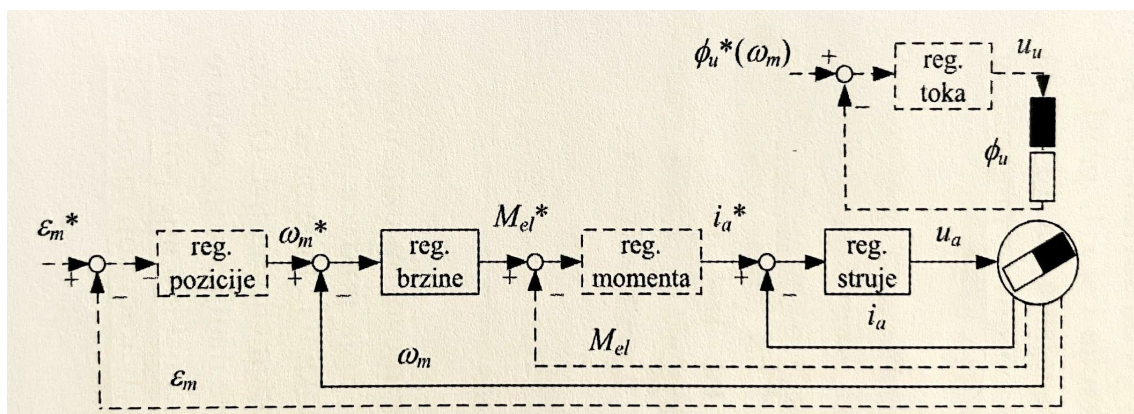
Druga prednost je da možemo podešavati regulatore „korak po korak“. Najprije podešavamo regulator struje (u praksi bez tereta) koji ima dva parametra jer, općenito, PI regulator ima dva parametra. Kad se to završi i kad se ispita taj regulacijski krug, onda je to apsolvirano. Zatim se ide ispitati nadređeni krug brzine vrtnje. Implementira se regulator brzine vrtnje (u praksi pod teretom), dajemo step i vidimo kako radi regulator brzine vrtnje. Podešavaju se njegova dva dodatna parametra, te onda idemo na regulator pozicije koji još ima jedan parametar. Stoga, umjesto da pet parametara odjednom namještamo što je poprilično zahtjevno, možemo podešavati korak po korak i u miru ispitivati pa ići dalje.

Nadalje, prednost kaskadne strukture je da je fleksibilna u prekapčanjima načina rada. Ne moraju se uvijek izvoditi sve petlje regulacije, već po potrebi prekapčamo ono što nam je potrebno. Primjer je automobil, gdje iz tempomata možemo preći u klasično upravljanje vozilom.

Također, s obzirom da je regulator struje iznimno brz, podešava se da dobro otklanja poremećaje, recimo induciranu elektromotornu silu. Jednom kada regulator dobro podesimo i ispitamo, regulator struje zbog svog brzog djelovanja otklanja te poremećaje na brz i efikasan način te su tada oni riješeni za nadređene krugove.

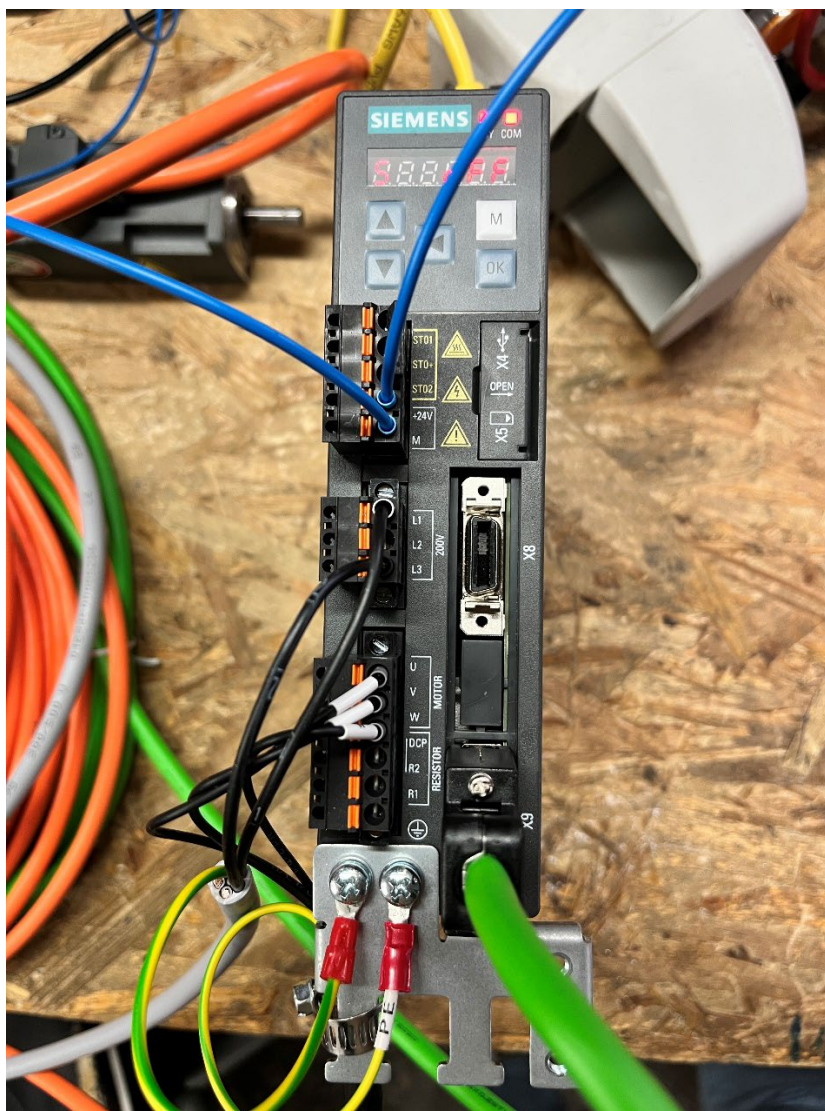
Međutim, svaka hijerarhijska struktura ima i nedostatak, a to je određena tromost. Naredbe idu korak po korak i u tom cijelom lancu može doći do određene tromosti koje recimo kompaktni regulator ne bi nužno imao. No, na sreću, u praksi je često taj nedostatak minoran i zanemariv.

Na [Slika 9] prikazana je pojednostavljena opća blokovska shema regulacije istosmjernog motora iz [1], a koja zorno prikazuje kaskadnu strukturu. Veličine na slici označene s „*“ označavaju referentne veličine, dok veličine bez takve oznake predstavljaju mjerene veličine koje se putem negativne povratne veze vraćaju u sumator.



Slika 9. Pojednostavljena opća blokovska shema regulacije IM, preuzeto sa [1]

Na [Slika 10] prikazan je korišteni SINAMIC V90 regulator u ovom završnom radu.



Slika 10. SINAMICS V90

2.4.1. Regulacija struje armature

Regulacijska petlja struje najpodređenija je regulacijska petlja iz razloga što struja armature predstavlja varijablu stanja s najbržom dinamikom. Općenito, kod kaskadne regulacijske strukture, uvijek se prvo rješava regulacija varijable koja je najbrža, a to je struja koja proporcionalno odgovara momentu. Dakle, prvo se mora regulirati ona veličina koja je preduvjet da bi nadređeni regulacijski krugovi mogli raditi, jer brzinu ne možemo podići bez momenta.

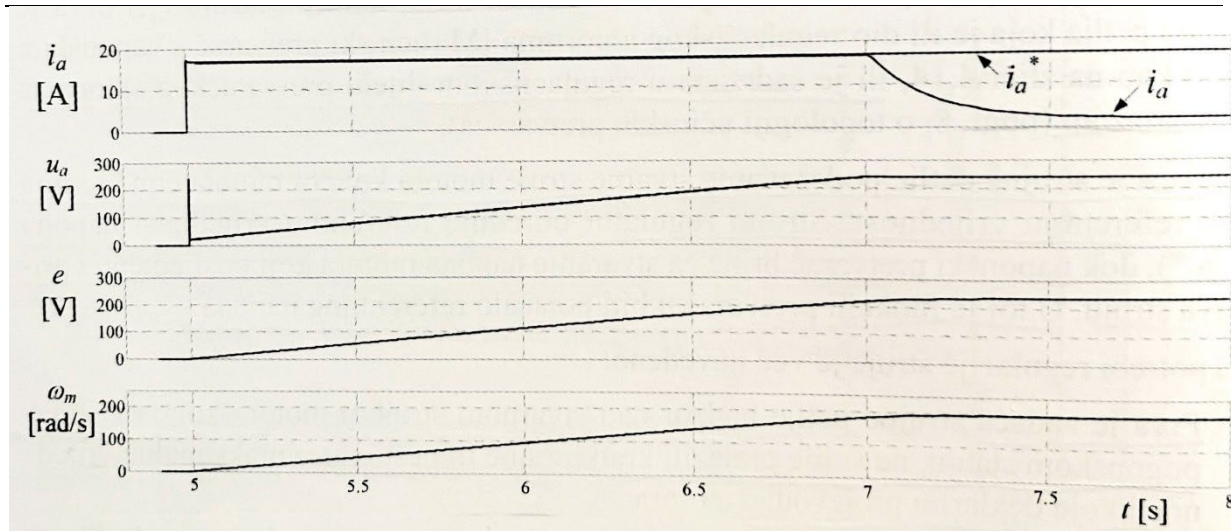
Zadaća strujne petlje je dakle podešavanje stvarne struje motora koja bi morala čim vjernije slijediti referentnu vrijednost. Regulator struje, koji je obično PI tipa, proračunava i određuje referentnu vrijednost napona armature, dok naponski pretvarač (poznatiji kao „chopper“) brine

o stvaranju referentnog iznosa napona i zadavanju istog motoru koji posljedično i prilagođava struju. Ovisno o pogonskom stanju, struja ne bi smjela prelaziti kratkotrajne ili nešto duže maksimalne vrijednosti koje deklarira proizvođač motora, a za istosmjerni motor je to tipično dva ili tri puta veće od nazivne struje. Druga zadaća bi bilo nadzirano stvaranje momenta kojeg diktira nadređena petlja brzine vrtnje. S obzirom da je moment posljedica magnetske uzbude i armaturne struje, a u većini radnog područja uzbuda konstantna, može se zaljučiti da je moment neposredno ovisan o armaturnoj struji. [1]

Kod ozbiljnijih pogona struja se mjeri najboljim senzorom za mjerenje struje, a to je Hallov senzor koji ima mogućnost mjerenja i AC i DC struja. Dobiveni signal obično treba dodatno filtrirati, a razlog tome su visokofrekvencijske smetnje. [1] Na manjim snagama se može koristiti i mjerni otpornik, tzv. shunt koji je manje točan, ali zato izrazito jeftin jer se radi o otporniku kroz kojeg teče struja i stvara pad napona koji se mjeri.

Kako je i ranije navedeno, budući da je kod konstantne uzbude regulacija struje armature u biti već i regulacija momenta, takva petlja bi bila dovoljna u sustavima koji zahtjevaju isključivo podešavanje momenta. Primjer je servo pogon u električnim vozilima (primjerice u električnom skuteru), gdje se preko ručice „gasa“ podešava moment. [1]

Na [Slika 11] prikazan je odziv samostalne strujne petlje, bez nadređenih regulacijskih petlji. Slika zorno prikazuje kako čak i uz pravilni izbor regulatora struje treba biti svjestan ograničena koje donosi takva samostalna petlja. Naime, na slici je vidljivo kako na početku struja u vrlo kratkom vremenu postigne svoju referentnu vrijednost zaslugom regulatora, koji na motor kratkotrajno narine maksimalni dozvoljeni napon pretvarača, koji je u ovom slučaju jednak nazivnom naponu. Rotor motora počinje ubrzavati konstantnim ubrzanjem koje je posljedica konstantnog momenta, a to se vidi i na slici gdje brzina raste „po rampi“. S povećanjem brzine raste i inducirani napon koji se suprotavlja narinutom naponu. Budući da struja teče zbog razlike između narinutog i inducirano napona, regulator mora povećavati narinuti napon, no ubrzanje ne može trajati beskonačno dugo. U određenoj točki (na slici u trenutku $t = 7$ s), napon izvora dosegne svoju maksimalnu vrijednost, a njoj se približi i inducirani napon te tada njihova razlika tada ide prema nuli što znači da daljnje održavanje referentne struje postaje nemoguće. Zbog toga stvarna struja, a time i moment pada skoro na nulu, motor prestane ubrzavati te se brzina ustali na brzini praznog hoda jer u idealnom praznom hodu moment teorijski nije niti potreban. Stoga, unatoč aktivnom, dobro optimiranom i uštimanom regulatoru, stvarna struja ne slijedi zadanu referencu. [1]



Slika 11. Odziv strujne regulacijske petlje, preuzeto sa [1]

2.4.2. Regulacija brzine

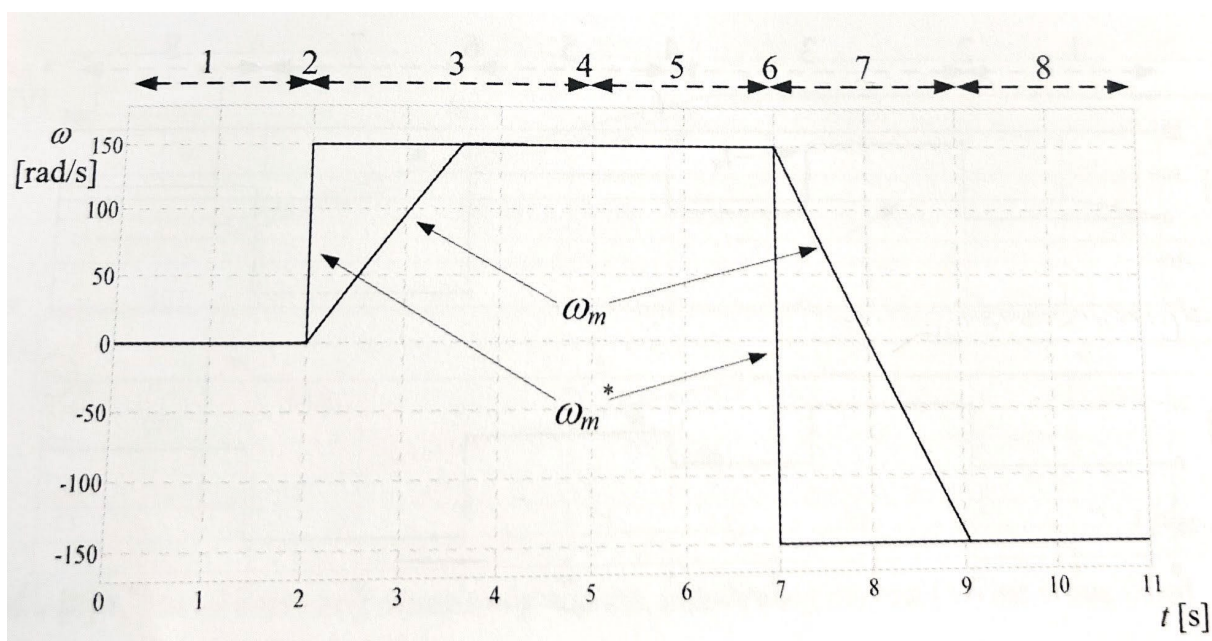
Korišteni motor u ovom završnom radu opremljen je mjeračem brzine vrtnje i pozicije, a radi se o inkrementalnom enkoderu. Radi se o mehaničkom senzoru iz kojeg se može izvući trenutna brzina vrtnje i pozicija, što je i cilj, da ne stavljamo više mehaničkih senzora jer je to povezano s troškovima i nezgrapnošću ugradnje.

Zadatak brzinske petlje je što vjernije osiguravanje stvarne brzine sukladno zadanoj referentnoj vrijednosti, uz najbolju moguću dinamiku i bez stacionarne pogreške koju otklanja integralno djelovanje. Brzina motora se mijenja podešavanjem momenta, što zapravo znači da je izlaz iz PI regulatora brzine referentni moment, a u ovom slučaju gdje je uzbuda konstantna, svodi se na referentnu armaturnu struju.

U primjeru iz [1] definirat će se određena područja, odnosno intervali i točke kako bi se lakše objasnila regulacija brzine prema [Slika 12]:

1. $0 \leq t \leq 2$ s: u ovom intervalu uzbudna struja dosegne stacionarnu vrijednost;
2. trenutak $t = 2$ s: u tom trenutku se referentna brzina skokovito promijeni uz neopterećeni rotor;
3. $2 \text{ s} < t < 3,5$ s: motor ubrzava uz konstantnu akceleraciju dok ne dosegne referentnu brzinu te se nakon toga vrti u praznom hodu;
4. trenutak $t = 5$ s: motor se optereti uz još uvijek konstantnu referentnu brzinu, a regulator brzine reagira povećanjem struje te se zato brzina nakon malog pada vrati na referentnu vrijednost, dok regulator struje novu struju održava povećanjem referentnog napona armature;

5. $5 \text{ s} < t < 7 \text{ s}$: nakon kratkotrajne prijelazne pojave stacionarno stanje ponovno se uspostavlja, ali sada uz opterećeni motor;
6. trenutak $t = 7 \text{ s}$: referentna vrijednost brzine u trenutku mijenja predznak
7. $7 \text{ s} < t < 9,1 \text{ s}$: brzina rotora smanjuje se do nule, a nakon toga se motor zavrti u suprotnom smjeru sve do referentne vrijednosti brzine – promjena smjera vrtnje naziva se reverziranje;
8. od $9,1 \text{ s}$ pa nadalje: motor se vrti u stacionarnom stanju uz nepromijenjeni moment tereta.



Slika 12. Odziv motora na regulaciju brzine vrtnje, preuzeto sa [1]

2.4.3. Regulacija pozicije

Regulacija pozicije (položaja, zakretnog kuta) potrebna je u zahtjevnijim aplikacijama za pozicioniranje. Većina pogona u praksi zahtijeva samo regulaciju brzine, no regulacija položaja potrebna je recimo kod CNC strojeva i robota. S obzirom da je regulacijski krug položaja nadređeni krugu regulacije brzine vrtnje, unutar regulacijskog kruga imamo integralno djelovanje. S obzirom da su poremećaji riješeni u podređenim regulacijskim petljama i kako ne bi unosili dodatno fazno kašnjenje te destabilizirali odziv, nije potrebno uvoditi dodatni integrator pa se regulacija u pravilu obavlja P regulatorom. Regulator pozicije daje referencu brzine, a u praksi se ne dopušta odziv s prebačajem kako bi se izbjeglo udaranje u odbojnice (eng. bumper). Stoga, cilj regulacije položaja je dobiti granični aperiodski odziv, odnosno najbrži mogući bez prebačaja kako bi se izbjegli neželjeni udari ili promjena smjera kretanja.

3. FUNKCIJSKI BLOK ZA UPRAVLJANJE SERVOMOTOROM UNUTAR SIEMENS TIA PORTAL-A

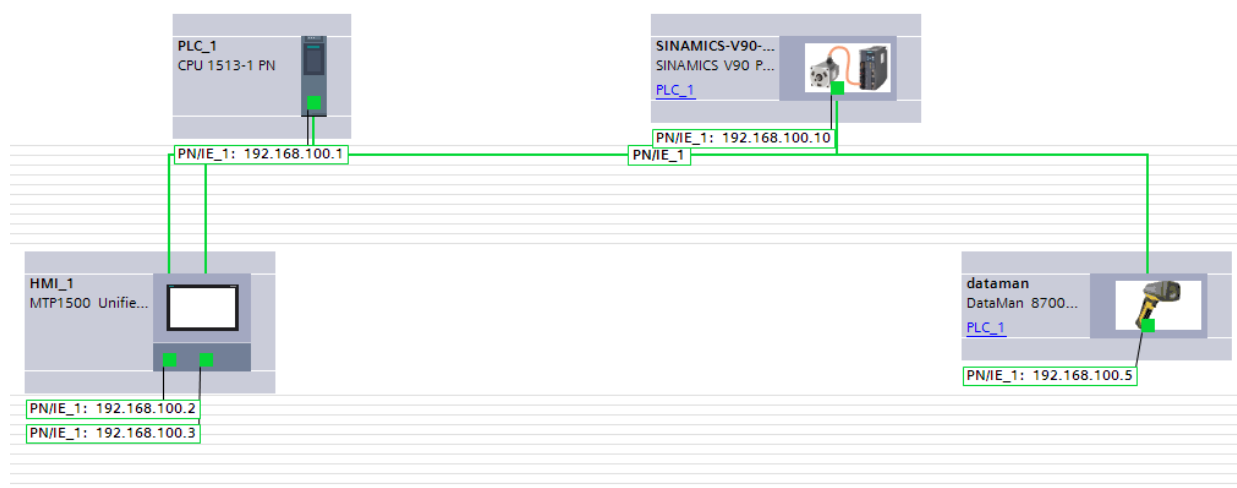
Općenito, funkcijski blokovi su programski blokovi koji trajno pohranjuju svoje ulazne, izlazne i ulazno-izlazne parametre u podatkovnim blokovima svake instance, omogućujući im pristup čak i nakon što je blok obrađen. Radi toga nazivaju se još i memorijski blokovi. Funkcijski blokovi sadrže potprograme koji se uvijek izvršavaju kada neki drugi blok u kodu pozove funkcijski blok. Također, funkcijski blok se može pozvati i nekoliko puta, na različitim mjestima u programu (i s različitim podatkovnim blokovima), što za rezultat daje jednostavnije programiranje funkcija koje se često ponavljaju. Svaki put kada se određeni funkcijski blok pozove, automatski se kreira instanca tog funkcijskog bloka, a da bi instanca ispravno radila, potreban je i podatkovni blok za tu instancu. Podatkovni blok sadrži vrijednosti specifične za tu instancu za formalne parametre deklarirane u funkcijskom bloku. Funkcijski blok može podatke specifične za instancu spremiti u svoj vlastiti podatkovni blok instance ili u podatkovni blok instance pozivajućeg bloka. [8]

S obzirom da se u ovom završnom radu upravlja servomotorom, korišteni su Motion Control funkcijski blokovi prema PLCopen standardu koji omogućava korištenje standardiziranih biblioteka (eng. library) za različite primjene. Navedeno štedi vrijeme pri izradi koncepta, umanjuje troškove održavanja i podrške uz eliminaciju bilo kakvog nerazumijevanja između programera. Standardizacija se provodi definiranjem biblioteka komponenti koje se mogu ponovno koristiti. Na taj način programiranje manje ovisi o hardveru, povećava se mogućnost ponovne upotrebe softvera za različite primjene i smanjuju se troškovi obuke. [9]

U nastavku će se opisati glavne korištene značajke u programskom sučelju Siemens TIA Portal, poput konfiguracije uređaja i tehnološkog objekta upravljanja, UDT (User Defined Types) koji se koristi u podatkovnim blokovima (eng. Data Block) te će se također opisati i dijelovi programskog koda.

3.1. Konfiguracija uređaja u SIEMENS TIA Portal-u

Za početak, potrebno je dodati i konfigurirati sve uređaje koji će se koristiti pod „Add new device“. Ranije opisani uređaji su konfigurirani i umreženi putem Profinet protokola, a dodatno je isproban i testiran i DataMan-ov skener za 2D kodove za odlazak u specifičnu poziciju. Prikaz na [Slika 13] može se vidjeti pod „Devices & networks“ te zatim „Network view“.



Slika 13. Uređaji umreženi Profinet protokolom

Otvaranjem PLC-a duplim klikom, potrebno je dodati i ulazni te izlazni modul te definirati opseg adresa koji će pokrivati. Korišten je digitalni ulazni modul s 32 ulaza (DI 32 x 24 VDC, 521-1BL10-0AA0) te digitalni izlazni modul sa 16 izlaza (DQ 16 x 24 VDC, 522-1BH10-0AA0). Ulazni modul koristi ulazne adrese od 0 do 3, s tim da svaka adresa ima osam ulaza, recimo %I0.1, %I0.2 i slično. Isto tako, izlazni modul koristi adrese od 0 do 1, s jednakom logikom s osam izlaza po adresi. Na [Slika 14] i [Slika 15] vidljiv je prikaz unutar „Device view“.

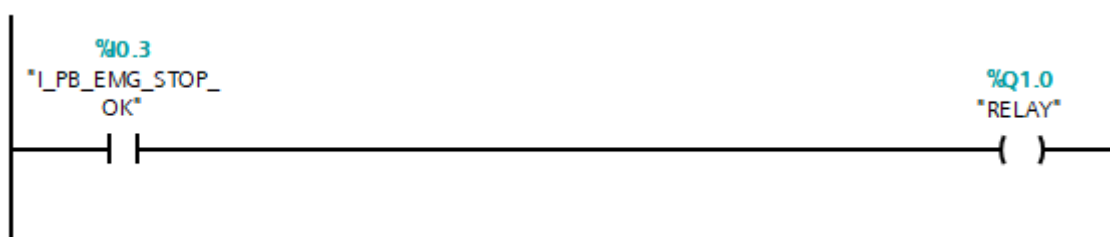
| | | | | | |
|-----------------------|---|------|-------|-------|--------------------|
| ▼ PLC_1 | 0 | 1 | | | CPU 1513-1 PN |
| ▶ PROFINETinterface_1 | 0 | 1 X1 | | | PROFINET interface |
| DI 32x24VDC BA_1 | 0 | 2 | 0...3 | | DI 32x24VDC BA |
| DQ 16x24VDC/0.5A BA_1 | 0 | 3 | | 0...1 | DQ 16x24VDC/0.5... |

Slika 14. Adrese ulaznog i izlaznog modula

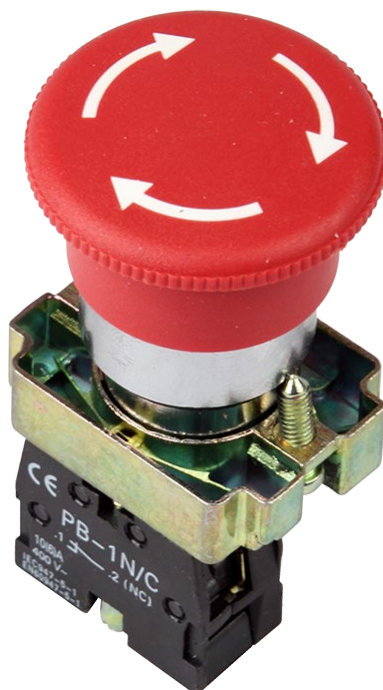


Slika 15. Prikaz u „Device view“ nakon dodavanja ulaznog i izlaznog modula

Nadalje, dodan je i relej između regulatora i napajanja, a rad releja koji propušta napon prema regulatoru može se zaustaviti sigurnosnom gljivom te je tada rad motora onemogućen. Inače, sigurnosna gljiva je sastavni dio svakog pogona radi sigurnosti u slučaju da se stvari krenu odvijati neželjenim tokom. Po svojoj strukturi, radi se o NC (eng. Normally Closed) kontaktu što znači da dok tipka nije pritisnuta, struja normalno protiče. Kada se tipka pritisne, strujni krug postaje otvoren te dotok struje do motora nije moguć. Na [Slika 16] vidljiva je Ladder logika za opisano u organizacijskom bloku „Main“, dok je na [Slika 17] vidljiva je sigurnosna sklopka (eng. Emergency Stop button).



Slika 16. Ladder logika za sigurnosnu sklopku i relej



Slika 17. Sigurnosna sklopka (eng. Emergency Stop button)

3.2. User Defined Types – UDT

UDT je vrsta podataka koju korisnik sam može kreirati te se uobičajeno koristi u programiranju programabilnog logičkog kontrolera kako bi se grupirali podaci i povećala mogućnost ponovne upotrebe. Ova struktura omogućuje korisniku kombiniranje bilo koje postojeće vrste podataka (Int, Bool, Real, itd.) u jedan paket koji se može iskoristiti na više podatkovnih blokova. Posljedično, promjenom podataka unutar UDT-a, promijenit će se svugdje unutar blokova gdje je ta vrsta podatka korištena. Stoga, navedeno daje veliku fleksibilnost kod velikog broja ponavljanja unutar programskog koda. Primjerice, ukoliko se u nekoj aplikaciji želi kontrolirati 10 servo osi, a znamo da ćemo za svaku servo os trebati iste komande, nema smisla za svaku ispočetka ih definirati, već na jednom mjestu napravimo generalno i koristimo ih u svakom servo motoru. U slučaju potrebe za nekakvom adaptacijom ili promjenom, ista će se primjeniti na svih 10 servo osi. Konvencija je da svaka takva vrsta podataka u nazivu kreće s „UDT“. Na [Slika 18] vidljiv je jedan UDT koji se koristio za upravljanje servo motorom iz zadatka. Unutar UDT-a su sve potrebne komande za kontroliranje jedne servo osi, a za potrebe završnog rada upravljalo se radom jedne servo osi što se može vidjeti na [Slika 19] gdje je kreiran podatkovni blok „AppAxis_TO_DB“ u kojem je pod nazivom „Axis_1“ ugniježden kreirani UDT. UDT se može iskoristiti dakle i za više servo osi, što znači da bi samo kreirali „Axis_2“ recimo i odabrali da se radi o „UDT_AppAxis_TO“.

| UDT_AppAxis_TO | | | | | | | |
|----------------|--------------------------------|-----------|---------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| | Name | Data type | Default value | Accessible f... | Writa... | Visible in ... | Setpoint |
| 1 | MC_POWER_Enable | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 2 | MC_RESET_Execute | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 3 | MC_RESET_Restart | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 4 | MC_HOME_Execute | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 5 | MC_MOVEJOG_JogForward | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 6 | MC_MOVEJOG_JogBackward | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 7 | MC_MOVEJOG_Velocity | Real | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 8 | MC_MOVEABSOLUTE_Execute | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 9 | MC_MOVEABSOLUTE_Position | Real | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 10 | MC_MOVERELATIVE_Execute | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 11 | MC_MOVERELATIVE_Distance | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 12 | MC_MOVEVELOCITY_Execute | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 13 | MC_MOVEVELOCITY_Velocity | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 14 | MC_HALT_Execute | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 15 | MC_STOP_Execute | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 16 | MC_TORQUELIMITING_Enable | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 17 | MC_TORQUELIMITING_Final_Torque | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 18 | Output_Actual_Position | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 19 | Output_Actual_Velocity | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 20 | Output_Actual_Speed | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 21 | Output_Actual_Acceleration | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 22 | Output_Actual_Torque | LReal | 0.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 23 | Output_MC_at_Standstill | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 24 | Output_MC_MoveAbs_Done | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 25 | Output_MC_MoveAbs_Busy | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 26 | Output_MC_Error | Bool | false | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Slika 18. UDT_AppAxis_TO

| AppAxis_TO_DB | | | | | | | | |
|---------------|---------------------|------------------|-------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | Name | Data type | Start value | Retain | Accessible f... | Writa... | Visible in ... | Setpoint |
| 1 | Static | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | Axis_1 | *UDT_AppAxis_... | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3 | MC_POWER_Enable | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 4 | MC_RESET_Execute | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 5 | MC_RESET_Restart | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 6 | MC_HOME_Execute | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 7 | MC_MOVEJOG_Jo... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 8 | MC_MOVEJOG_Jo... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 9 | MC_MOVEJOG_Ve... | Real | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 10 | MC_MOVEABSOLU... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 11 | MC_MOVEABSOLU... | Real | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 12 | MC_MOVERELATIV... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 13 | MC_MOVERELATIV... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 14 | MC_MOVEVELOCIT... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 15 | MC_MOVEVELOCIT... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 16 | MC_HALT_Execute | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 17 | MC_STOP_Execute | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 18 | MC_TORQUELIMITI... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 19 | MC_TORQUELIMITI... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 20 | Output_Actual_Po... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 21 | Output_Actual_Ve... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 22 | Output_Actual_Sp... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 23 | Output_Actual_Ac... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 24 | Output_Actual_To... | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 25 | Output_MC_at_Sta... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 26 | Output_MC_Move... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 27 | Output_MC_Move... | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 28 | Output_MC_Error | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Slika 19. Podatkovni blok AppAxis_TO_DB za upravljanje radom servo motora

3.3. Tehnološki objekt upravljanja (eng. Technology objects)

Unutar TIA Portal-a odabirom na „Technology objects“ te zatim „Add new object“ moguće je odabrati „TO_PositioningAxis“ pod „Motion Control“. Ovaj tehnološki objekt neophodan je za sve aplikacije u kojima je potrebno precizno i kontrolirano pozicioniranje osi servo motora iz korisničkog programa. Upravljanje pozicijom omogućuje brzo i točno definirano pozicioniranje osi te kompenzaciju mogućih pogrešaka pri pokretanju ili pri postizanju ciljane pozicije.

Prilikom konfiguriranja tehnološkog objekta upravljanja, konkretno za ovaj regulator SINAMICS V90 i enkoder na korištenom servo motoru, potrebno je odabrati podržani telegram preko kojeg je definiran način razmjene podataka i u kojem su sadržane sve potrebne informacije (recimo mjerena brzina i pozicija). Koristit će se „Standard telegram 102“, kao što je i vidljivo na [Slika 20]. Također, pod „Data exchange with encoder“ je potrebno odabrati identičan telegram.



Slika 20. Odabir standardnog telegrama 102 za razmjenu podataka s regulatorom

3.3.1. Motion Control funkcijski blokovi prema PLCopen standardu

Nešto ranije već opisani, standardizirani funkcijski blokovi prema PLCopen standardu za upravljanje s osi servo motora nalaze se na [Slika 21].

| | | | | | |
|---|----------------|-------------------------|-------------|--|---|
| ▼ | Folder | Motion Control | V6.0 | | ▲ |
| | Function Block | MC_Power | V6.0 | | |
| | Function Block | MC_Reset | V6.0 | | |
| | Function Block | MC_Home | V6.0 | | |
| | Function Block | MC_Halt | V6.0 | | |
| | Function Block | MC_MoveAbsolute | V6.0 | | |
| | Function Block | MC_MoveRelative | V6.0 | | |
| | Function Block | MC_MoveVelocity | V6.0 | | ☰ |
| | Function Block | MC_MoveJog | V6.0 | | |
| | Function Block | MC_MoveSuperimpo... | V6.0 | | |
| | Function Block | MC_SetSensor | V6.0 | | |
| | Function Block | MC_Stop | V6.0 | | |
| | Function Block | MC_SetAxisSTW | V6.0 | | |
| | Function Block | MC_WriteParameter | V6.0 | | |
| | Folder | Measuring input, out... | | | |
| | Folder | Synchronous motion | | | |
| | Folder | Cam | | | |
| | Folder | MotionIn | | | |
| | ▼ | Folder | Torque data | | |
| | Function Block | MC_TorqueAdditive | V6.0 | | |
| | Function Block | MC_TorqueRange | V6.0 | | |
| | Function Block | MC_TorqueLimiting | V6.0 | | |
| | Folder | Motion (kinematics) | | | |
| | Folder | Zones | | | |
| | Folder | Tools | | | |
| | Folder | Coordinate systems | | | ▼ |

Slika 21. Motion Control funkcijski blokovi prema PLCopen standardu

3.4. Funkcijski blok SERVOMOTOR FB

U nastavku će biti prikazane mogućnosti upravljanja servo motorom tvrtke Siemens AG koristeći kreirani funkcijski blok.

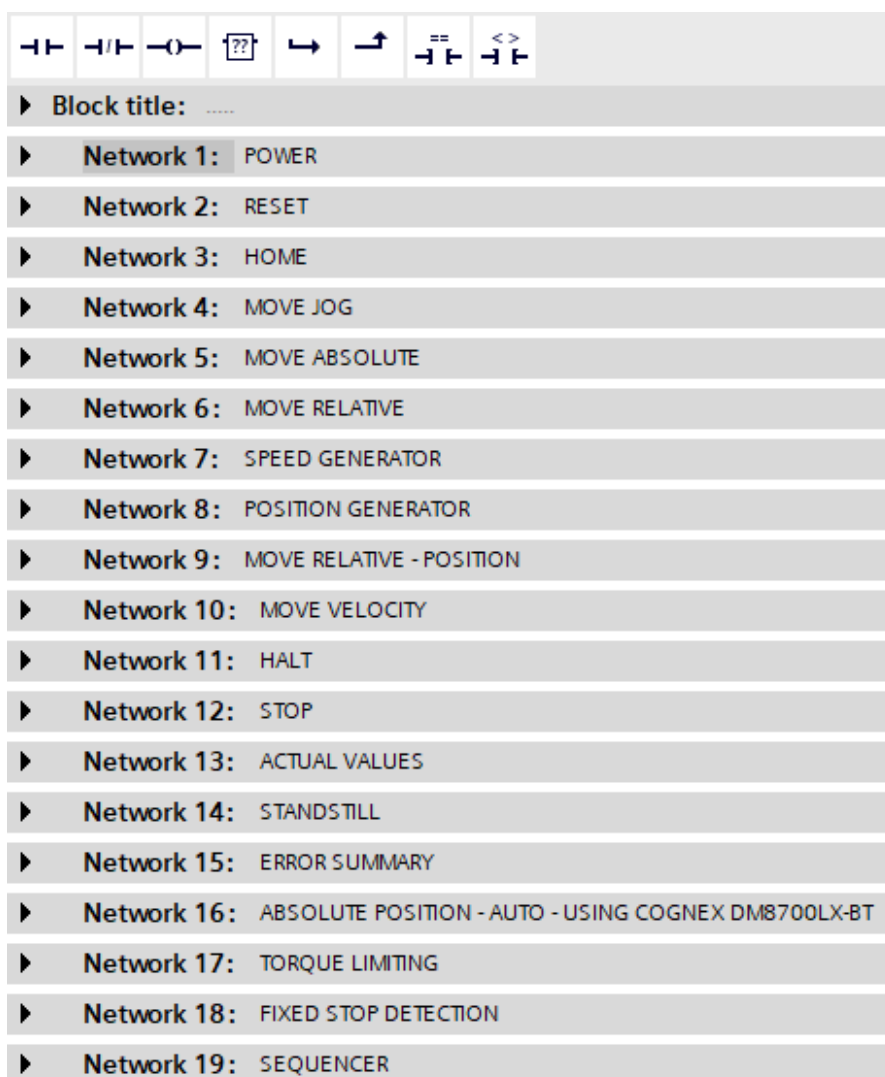
Za početak, na [Slika 22] prikazani su ulazi i izlazi funkcijskog bloka. Na ulaze iz podatkovnog bloka „AppAxis_TO_DB“ prikazanog na [Slika 19] možemo dovoditi podatke za „Axis_1“, a izlaze također spremati u isti podatkovni blok i tako u realnom vremenu pratiti parametre i upravljati servo motorom. U slučaju da je potrebno upravljati s više servo osi, potrebno je u organizacijskom bloku „Main“ pozvati željeni broj funkcijskih blokova „SERVOMOTOR FB“ i na njihove ulaze dovoditi željene parametre i iz izlaza također pratiti ponašanje servo motora. U tom slučaju bi u podatkovnom bloku „AppAxis_TO_DB“ kreirali „Axis_2“ i ponovno odabrali da se radi o „UDT_AppAxis_TO“, a također bi morali i kreirati novi tehnološki objekt upravljanja unutar programa gdje definiramo svojstva osi koja se upravlja, kako je ranije opisano. Također, na [Slika 22] vidljiv je i dio „InOut“ koji funkcijskom bloku služi da zna o kojoj servo osi se radi, odnosno od kojeg tehnološkog objekta upravljanja

će dobivati i davati van podatke. Dakle, unutar funkcijskog bloka je definirano da se radi o tipu podatka „TO_PositioningAxis“ koji interno sadrži sve potrebne podatke za upravljanje s osi te se iz tog ulaza/izlaza mogu pisati i čitati podaci.

| | Name | Data type | Default value | Retain | Accessible f... | Writa... | Visible in ... | Setpoint |
|----|------------------------|--------------------|---------------|------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| 1 | ▼ Input | | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | MC_POWER_Enable | Bool | false | Non-ret... | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 3 | MC_RESET_Execute | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 4 | MC_RESET_Restart | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 5 | MC_HOME_Execute | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 6 | MC_MOVEJOG_JogFor... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 7 | MC_MOVEJOG_JogBa... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 8 | MC_MOVEJOG_Velocity | Real | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 9 | MC_MOVEABSOLUTE_... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 10 | MC_MOVEABSOLUTE_... | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 11 | MC_MOVERELATIVE_E... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 12 | MC_MOVERELATIVE_D... | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 13 | MC_MOVEVELOCITY_... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 14 | MC_MOVEVELOCITY_... | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 15 | MC_HALT_Execute | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 16 | MC_STOP_Execute | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 17 | MC_TORQUELIMITING... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 18 | MC_TORQUELIMITING... | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 19 | ▼ Output | | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 20 | Output_MC_at_STAN... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 21 | Output_MC_ERROR | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 22 | Output_Actual_Position | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 23 | Output_Actual_Velocity | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 24 | Output_Actual_Accel... | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 25 | Output_Actual_Speed | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 26 | Output_Actual_Torque | LReal | 0.0 | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 27 | Output_MC_MoveAbs... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 28 | Output_MC_MoveAbs... | Bool | false | Non-retain | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 29 | ▼ InOut | | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 30 | ► TO_Axis | TO_PositioningAxis | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

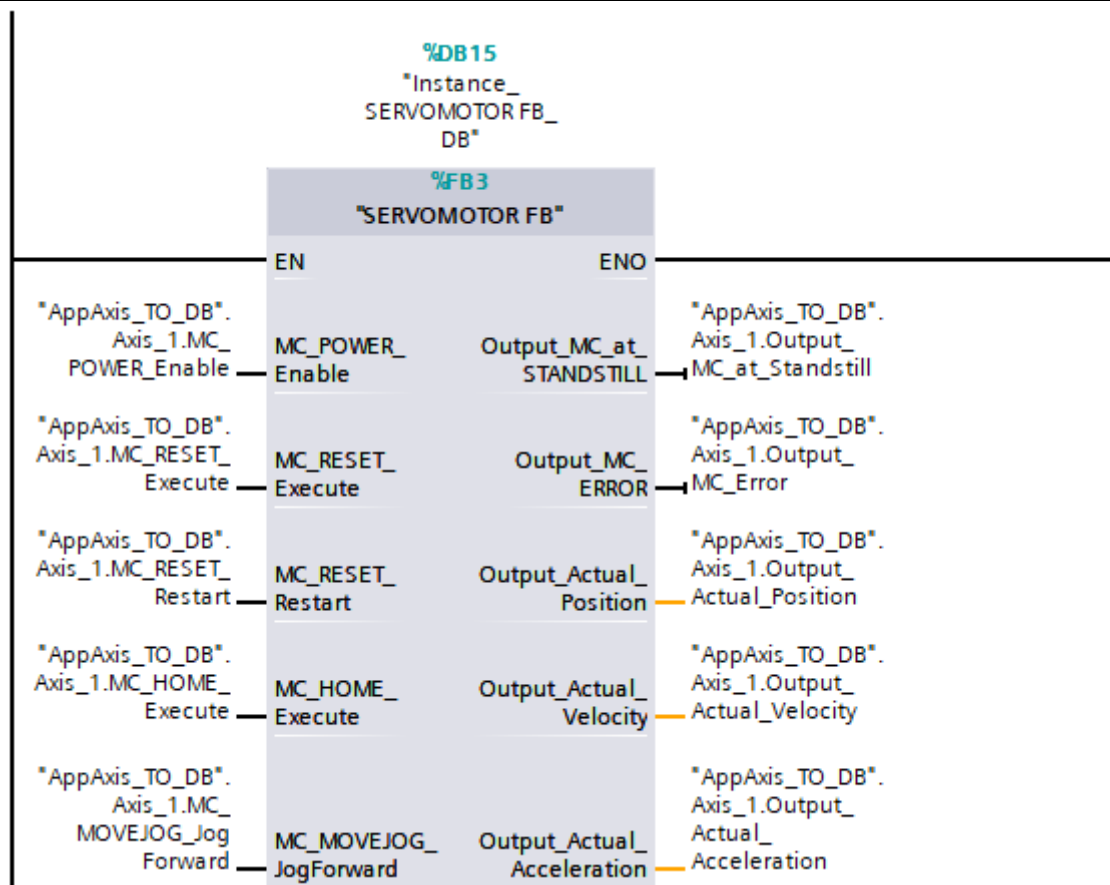
Slika 22. Ulazi i izlazi u funkcijskom bloku „SERVOMOTOR FB“

Na [Slika 23] vidljivi su svi „Network“, odnosno dijelovi funkcijskog bloka. Svaki „Network“ ima svoju zadaću te se svaki izvršava odozgo prema dolje te s lijeva prema desno. U nastavku završnog rada detaljnije će se objasniti glavni dijelovi funkcijskog bloka.



Slika 23. Svi „Network“ unutar funkcijskog bloka „SERVOMOTOR FB“

Na [Slika 24] vidljiv je dio instance nakon pozivanja funkcijskog bloka „SERVOMOTOR FB“ u organizacijskom bloku „Main“, a na slici se mogu vidjeti ranije opisani ulazi i izlazi funkcijskog bloka na koje su dodani podaci iz podatkovnog bloka „Axis_1“ iz „AppAxis_TO_DB“. Za nastalu instancu kreira se i podatkovni blok u kojem su također vidljivi svi podaci vezni uz upravljanje određene servo osi, no ne samo izlazi i ulazi, već i interni podaci unutar samog funkcijskog bloka, što omogućuje dodatnu fleksibilnost.

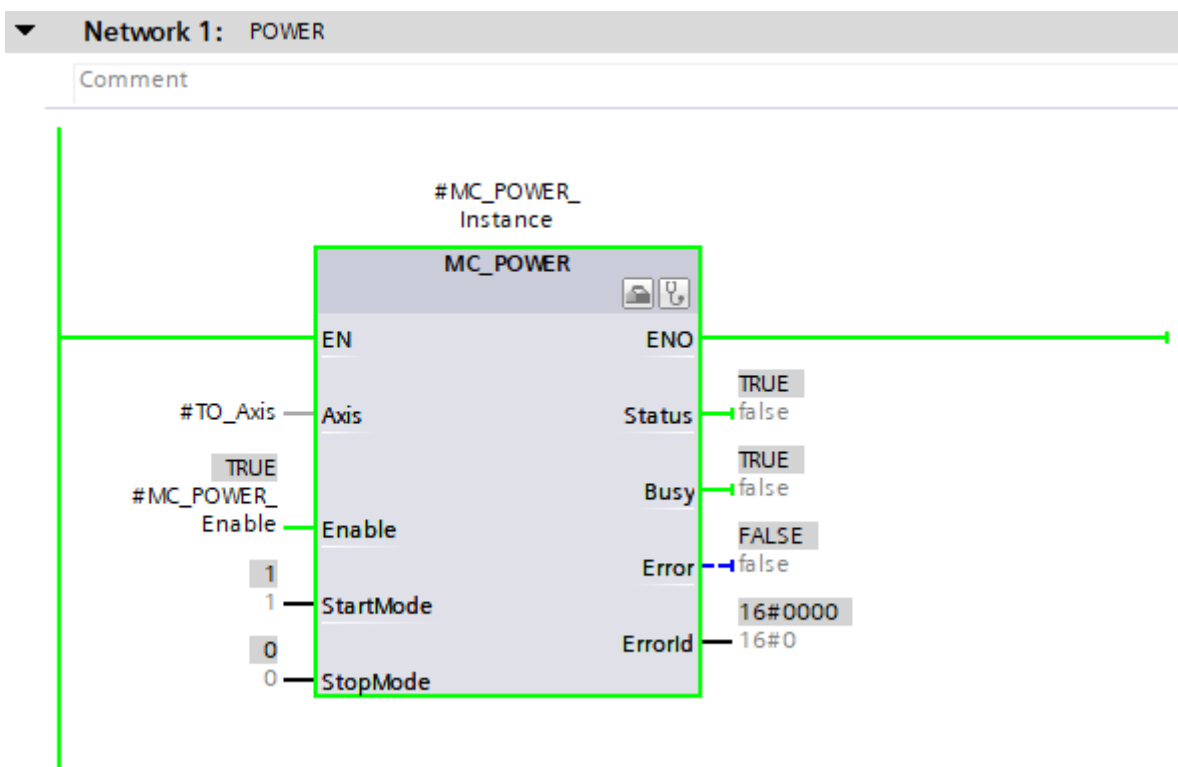


Slika 24. Instanca funkcijskog bloka „SERVOMOTOR FB“ u „Main“

3.4.1. Omogućenje/onemogućenje rada osi

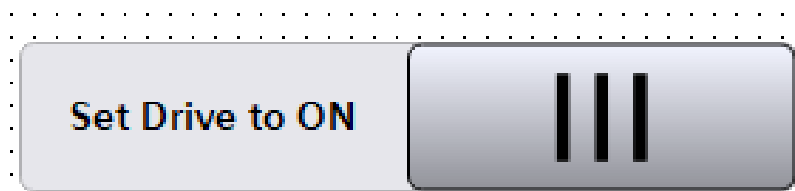
Prikaz Motion Control bloka „MC_POWER“ za omogućenje/onemogućenje rada osi, odnosno tehnološkog objekta upravljanja vidljiv je na [Slika 25], uz promatranje ponašanja bloka u realnom vremenu (zelena boja funkcijskog bloka). S ovim blokom se konfigurirani regulator može uključivati (status ON) ili isključivati (status OFF). Vrijedi napomenuti kako onemogućenje rada tehnološkog objekta (status OFF), prekida sve Motion Control zadatke na odgovarajućem tehnološkom objektu te navedeno ne može biti otkazano ili prilagođeno kako korisnik želi. U korisničkoj dokumentaciji, odnosno „Help“ unutar programa (klik F1 na blok za koji trebamo informacije), se mogu pronaći sve potrebne informacije poput detaljnog opisa svih parametara bloka, potrebnih početnih uvjeta za rad samog bloka i slično. Ondje se može i pronaći opis ulaza „StartMode“ i „StopMode“ koji definiraju na koji način funkcijski blok radi. U ovom slučaju, „1“ kod „StartMode“ označava kako se servo os upravlja po poziciji, a „0“ kod „StopMode“ označava kako se prilikom onemogućenja rada tehnološkog objekta, servo motor zaustavlja prema „Emergency stop“ načinu rada, odnosno u istom trenutku, s deceleracijom definiranom u samom tehnološkom objektu. Na [Slika 25] je vidljiv i status

greške koji prikazuje kako greške nema, no za slučaj da je „TRUE“, ispod pod „ErrorId“ se može vidjeti četveroznamenkasti kod greške, a opis greški za svaki pojedini kod je također vidljiv u korisničkoj dokumentaciji.



Slika 25. Motion Control blok „MC_POWER“

Na [Slika 26] vidljiv je „switch“ s kojim na HMI uređaju možemo omogućiti ili onemogućiti rad tehnološkog objekta upravljanja. U prikazanom položaju je u statusu OFF, a u slučaju pritiska na „switch“, rad regulatora i servo motora se omogućuje, što jasno označava i glasan zvuk. Switch, a i sve buduće naredbe, su putem „HMI Tags“ povezane sa podatkovnim blokom „Axis_1“ iz „AppAxis_TO_DB“ s kojim se upravlja radom tehnološkog objekta upravljanja.



Slika 26. Switch na HMI za omogućenje rada tehnološkog objekta upravljanja

3.4.2. Podešavanje tehnoloških parametara objekta upravljanja - prijenosni omjeri, pozicijski i brzinski limiti

Unutar programa TIA Portal V18 pod „Technology objects“ nalazi se opcija „Configuration“ u kojoj podešavamo sva svojstva potrebna za ispravan rad i upravljanje istim. Pod „Extended parameters“ nalazi se opcija „Mechanics“ u kojoj je moguće podesiti kakvu vrstu enkodera koristimo (na vratilu motora, na strani tereta ili vanjski mjerni uređaj), zatim želimo li omogućiti kompenzaciju zazora (eng. backlash compensation) te podešavanje prijenosnog omjera zupčanika ili remenica između vratila servo motora i recimo nekakvog vretena koje pomiče stol.

Na [Slika 27] je vidljivo kako se inkrementalni enkoder nalazi na vratilu servo motora, kako trenutno nije omogućena kompenzacija zazora te se može vidjeti kako je prijenosni omjer 1:1, a jedan okret motora odgovara pomaku od 10 mm na vretenu.

Encoder

Encoder mounting type: On motor shaft
 Invert encoder direction

Backlash compensation

Enable backlash compensation

Size of backlashes: 0.2 mm

Velocity of backlash compensation: 0.0 mm/min

Absolute homing direction: Positive

Drive mechanism

Invert drive direction

Load gear

Number of motor revolutions: 1

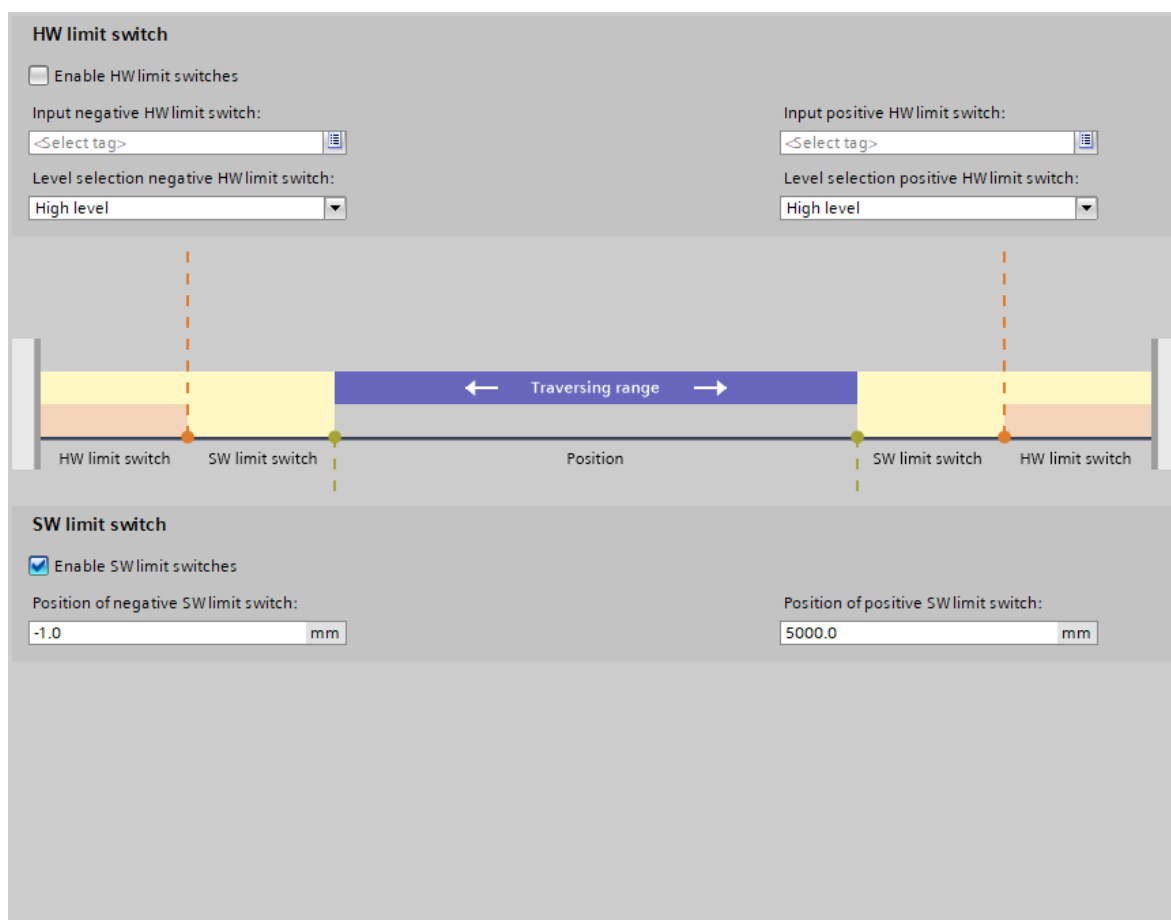
Number of load revolutions: 1

Position parameters

Leadscrew pitch: 10.0 mm/rot

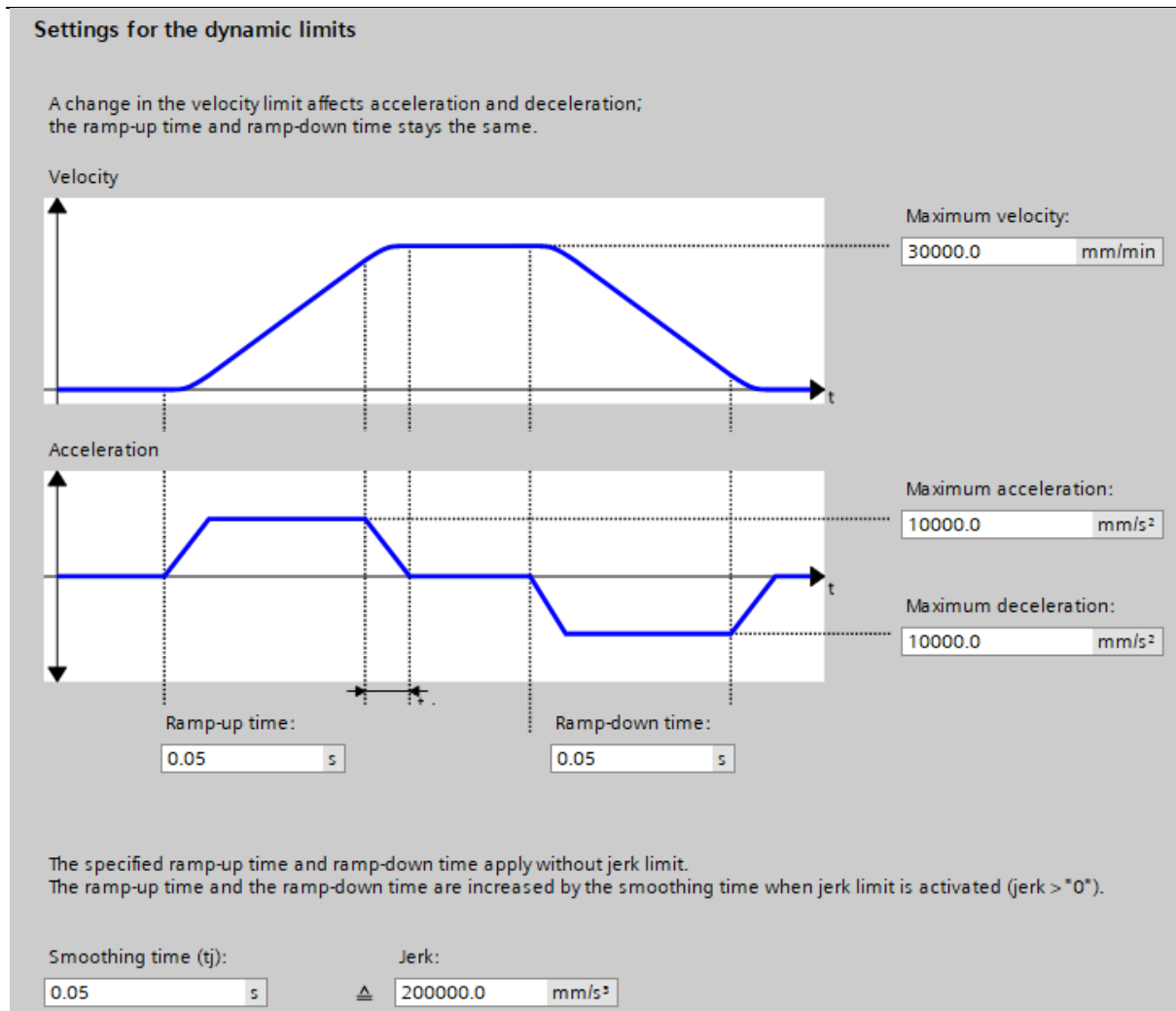
Slika 27. Prijenosni omjer

Nadalje, pod opcijom „Position limits“ moguće je podesiti hardverske i softverske pozicijske limite. Kako bi nakon podešavanja regulacijskih petlji i fino namještenog odlaska u „Home“ poziciju uz limitiranje momenta bili dodatno osigurani od neželjenih udara u odbojnice ili krajnje limite (nekakva tvrda podloga), možemo podesiti navedene limite. Hardverski limiti se koriste u slučaju da u projektu postoje prekidači koji se aktiviraju prije dolaska do tvrde podloge i tako zaustavljaju rad. Softverski limiti nastupaju prije hardverskih, kao što je vidljivo na [Slika 28]. U ovom slučaju se na pozicijama – 1 i 5000 mm servo os zaustavlja, neovisno o možda zadanom odlasku u neku poziciju van tog intervala.



Slika 28. Pozicijski limiti

Također, pod „Dynamic limits“ moguće je podesiti najveću dopuštenu brzinu kojom se može okretati vratilo upravljanog servo motora. Promjena parametra brzine automatski mijenja i ostale parametre vidljive na [Slika 29].



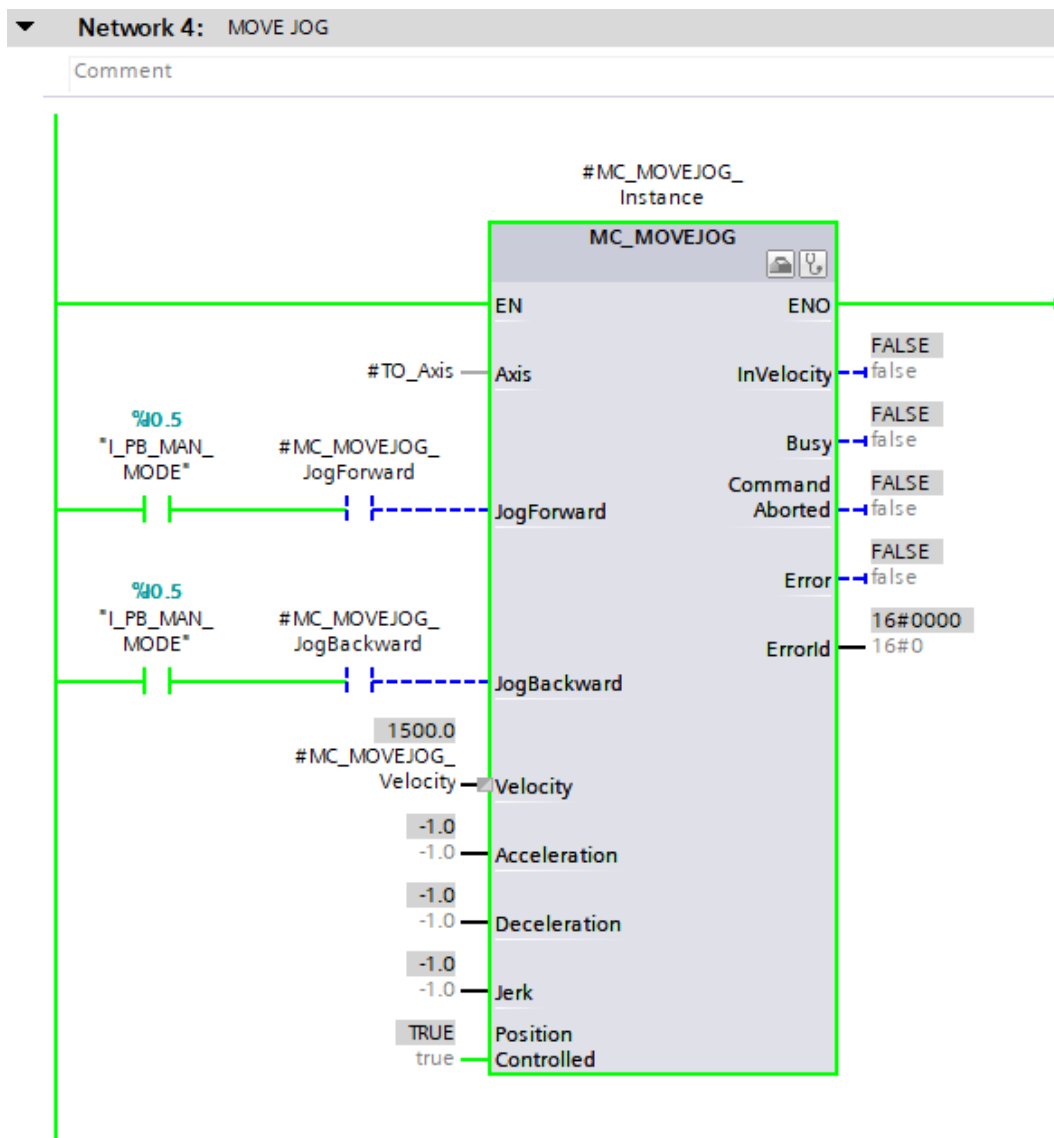
Slika 29. Brzinski limiti

3.4.3. Ručni način rada preko HMI uređaja (eng. jog function)

U ovom potpoglavlju će se opisati ručni način upravljanja servo motorom putem ranije opisanog HMI uređaja. Za ručno upravljanje korišten je Motion Control blok „MC_MOVEJOG“ koji je vidljiv na [Slika 30]. Ručno upravljanje servo motorom se recimo koristi prilikom puštanja pokretne trake (eng. conveyor belt) u pogon gdje je potrebno ispravno podesiti kretanja i rad trake za slučaj da je napravljena iz više dijelova.

U ovom završnom radu, a i u realnim industrijskim primjenama, postavlja se dodatni uvjet prije mogućnosti ručnog upravljanja na HMI uređaju, a to je uvjet u kojem je prekidač za ručni način uključen, a automatski u isto vrijeme isključen. Na [Slika 30] je vidljivo kako je radni kontakt pod adresom %I0.5 zelene boje, što označava da je prekidač upravo u toj poziciji koja omogućuje ručni rad. Naravno, vrijedi i dodati kako se u stvarnim industrijskim primjenama dodaju i ostali uvjeti prije mogućnosti pokretanja, a to su recimo da su vrata od ograde zatvorena, da senzori ne prepoznaju nikakav strani predmet unutar ćelije i slično. Nadalje,

vrijednost „-1“ pod „Acceleration“, „Deceleration“ i „Jerk“ označava kako se za te parametre uzimaju definirane vrijednosti prilikom definiranja brzinskih limita. S obzirom da sve vrijednosti manje od nule to predstavljaju, ovdje je postavljeno „-1“. Parametar „Velocity“ upisujemo ručno putem HMI uređaja u polje predviđeno za to te se zatim ta vrijednost upisuje u funkcijski blok. S tom brzinom se dakle vrši ručni način rada. Parametar „InVelocity“ može biti koristan jer nam daje informaciju kada je motor postigao traženu referencu brzine. To može biti okidač za neki drugi funkcijski blok.



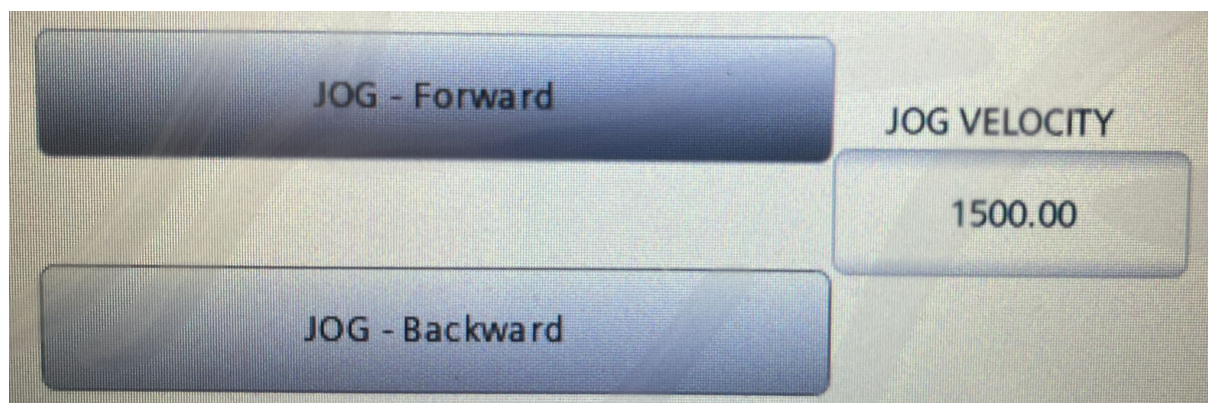
Slika 30. Motion Control blok „MC_MOVEJOG“

Sklopka s kojom je biran način rada između manualnog i automatskom u završnom radu vidljiva je na [Slika 31].



Slika 31. Odabir načina rada pomoću sklopke

Na samom HMI uređaju, točnije na zaslonu nazvanom „Servo_Motor“, dodani su i prekidači koje je potrebno držati kako bi se servo os okretala te polje gdje se upisuje brzina ručnog upravljanja. Navedeno je vidljivo na [Slika 32].



Slika 32. Sučelje HMI-a za ručno upravljanje servo motorom

3.4.4. Parametriranje servo-osi (eng. axis tuning) – strujna i brzinska petlja

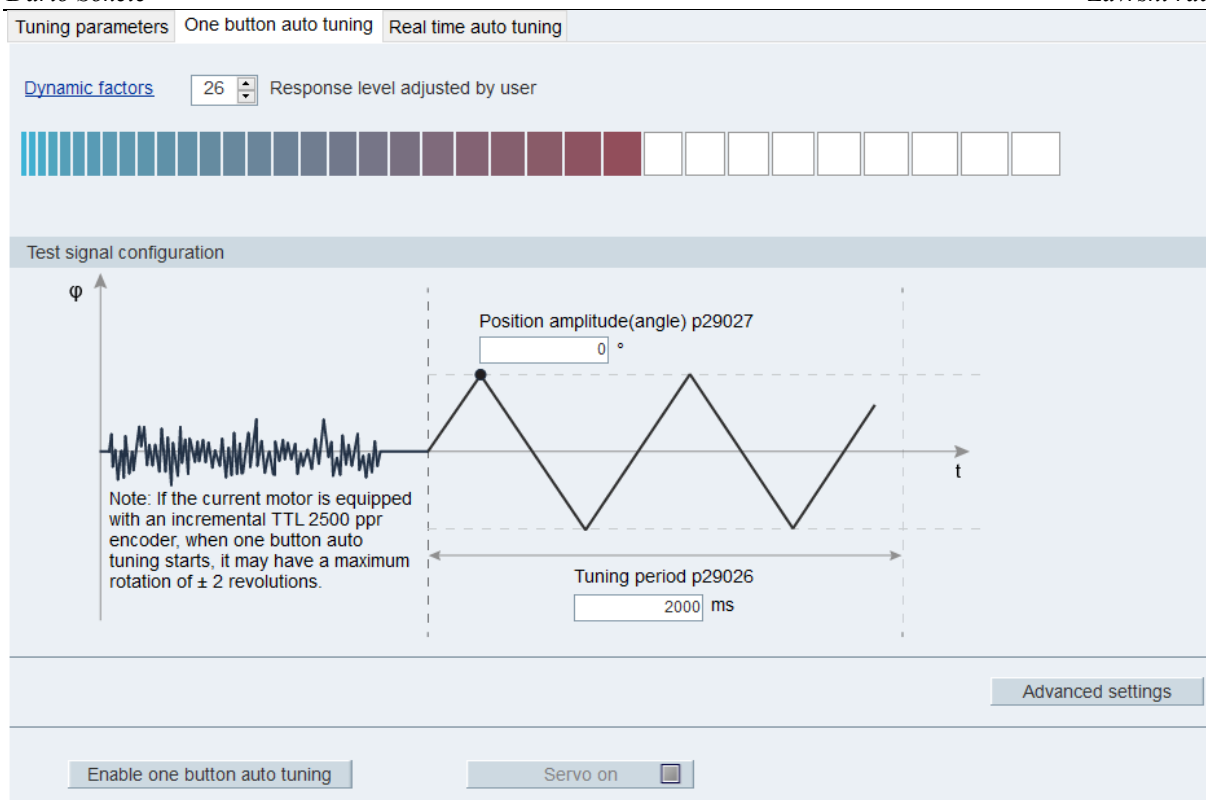
Slijedi opis dijela funkcijskog bloka i potrebnih opcija za ispravno podešavanje regulacijske petlje brzine vrtnje servo motora. Prije svega, vrijedi naglasiti kako se najpodređenija regulacijska petlja, strujna, podešava prema induktivitetu i otporu motora. S obzirom da se radi o SIEMENS-ovom servo motoru, navedene vrijednosti itekako su poznate unaprijed. Stoga, je strujnu regulacijsku petlju potrebno podešavati bez tereta na vratilu kako bi regulator ispravno podesio proporcionalno i integralno pojačanje upravo prema poznatom induktivitetu i otporu servo motora.

SINAMICS V90 PN omogućava dva načina automatskog parametriranja: „one-button“ i „real-time“. Funkcija automatskog parametriranja može optimizirati upravljačke parametre s omjerom momenta inercije opterećenja stroja i postaviti odgovarajuće parametre strujne petlje za automatsko suzbijanje rezonancije stroja. U ovom radu je korišteno „one-button“ automatsko parametriranje strujne petlje te će biti detaljnije opisano.

Navedena opcija procjenjuje moment inercije opterećenja stroja i mehaničke karakteristike s unutarnjim naredbama gibanja. Da bi se postigli željeni rezultati, proces se može izvršavati isponova prije nego što se s pogonom upravlja pomoću glavnog kontrolera, a maksimalna brzina je ograničena nazivnom brzinom. Prije korištenja ove opcije, u praksi je potrebno servo motor pomaknuti u srednji mehanički položaj kako bi se izbjeglo približavanje pozicijskom limitu. Preduvjeti za korištenje ove opcije su sljedeći:

- omjer momenta tromosti opterećenja stroja je nepoznat i treba ga procijeniti,
- servo motor se može okretati u smjeru i obrnuto od smjera kazaljke na satu,
- za servo motor s inkrementalnim enkoderom, motoru mora biti omogućena slobodna rotacija otprilike oko dva okreta kada započne automatsko parametriranje.

Na [Slika 33] može se vidjeti prikaz unutar programa SIEMENS SINAMICS V-ASSISTANT za opciju „One button auto tuning“. Parametar „Position amplitude (angle)“ je preporučeno staviti na vrijednost 360°.

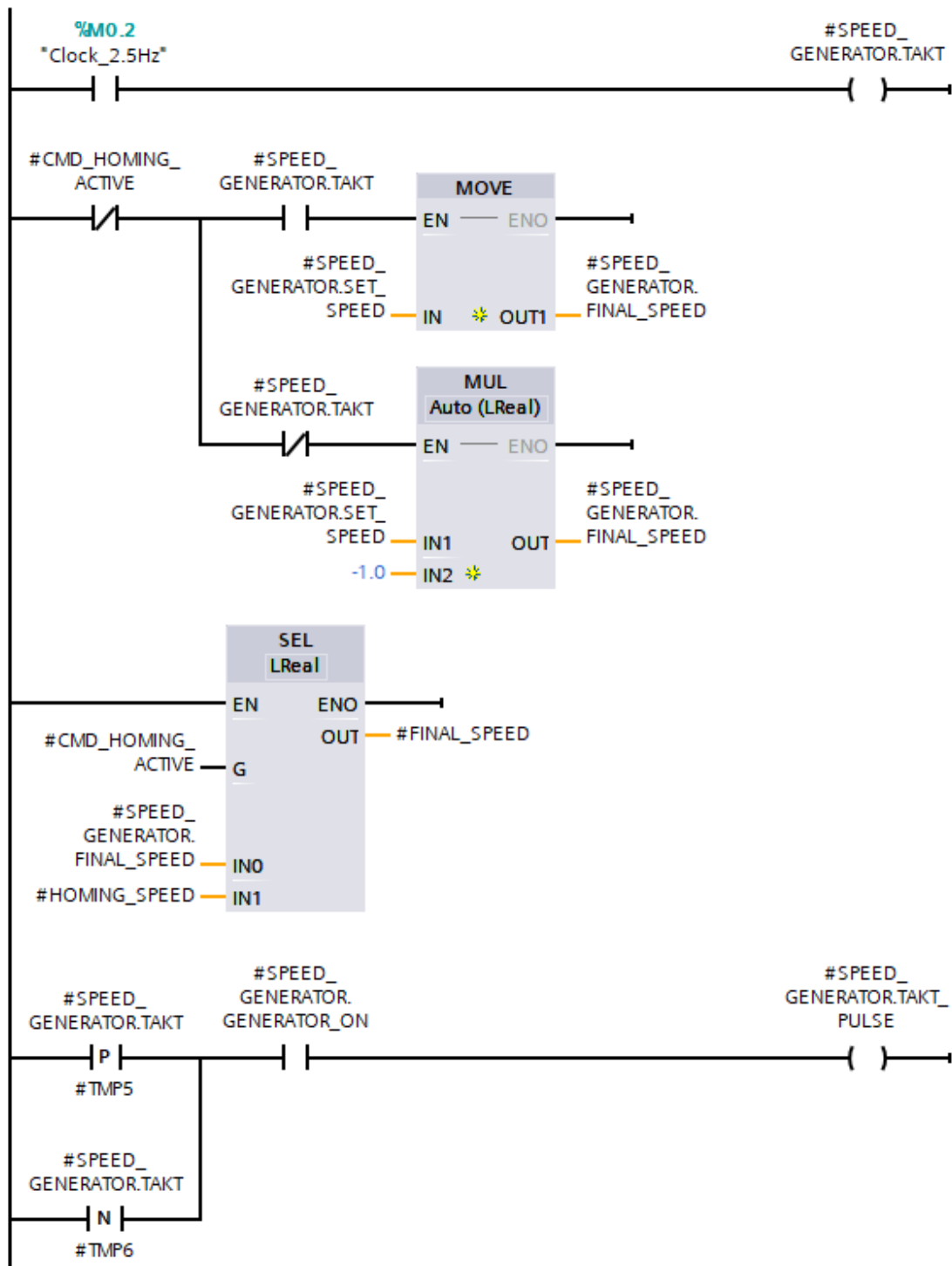


Slika 33. „One button auto tuning“ unutar V-ASSISTANT

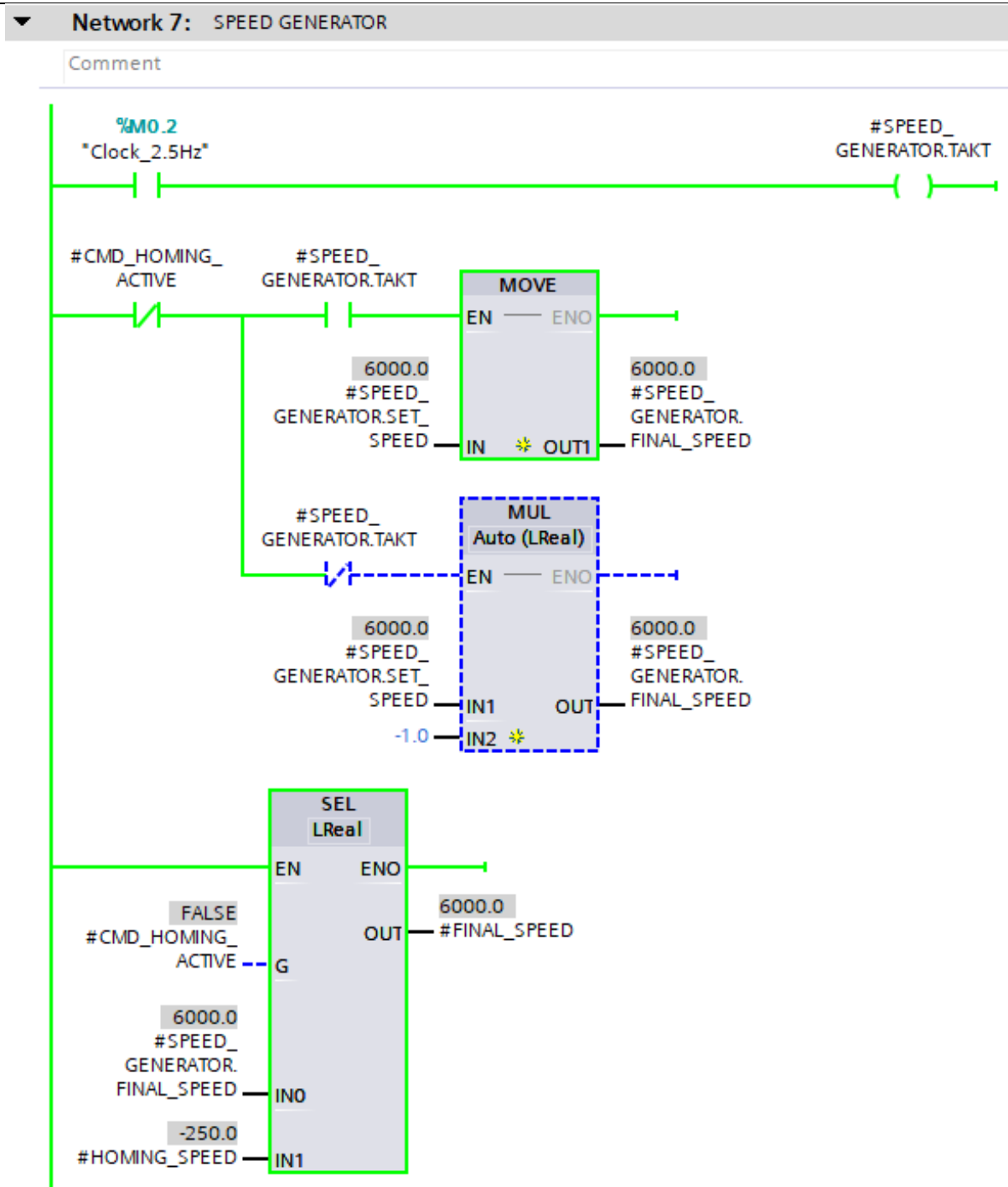
Nakon što je strujna petlja ispravno podešena, može se preći na podešavanje regulacijske petlje brzine vrtnje. U praksi, regulacijsku petlju brzine vrtnje potrebno je podešavati s teretom na vratilu kako bi se pravilno podesilo proporcionalno i integracijsko pojačanje baš za taj teret, no u ovom završnom radu tereta na vratilu nije bilo, stoga je podešeno bez. Također, u praksi je rijetko korištenje derivacijskog djelovanja prilikom parametriranja.

Kako bi se pravilno moglo izvršiti podešavanje regulacijske petlje brzine vrtnje, unutar funkcijskog bloka je kreiran generator brzine koji služi za generiranje STEP funkcije prema željenoj referenci brzine. Na [Slika 34] vidljiv je kompletan „Network 7: SPEED GENERATOR“ sa [Slika 23] u offline načinu prikaza, dok je na [Slika 35], [Slika 36] i [Slika 37] prikaz u realnom vremenu za vrijeme rada servo motora (eng. monitoring ON). Može se vidjeti kako za frekvenciju od 2,5 Hz i kada #CMD_HOMING_ACTIVE nije aktivan (koristi se u dijelu koji će biti kasnije opisan, a tiče se povratka na početnu poziciju uz limit momenta motora), kontakt #SPEED_GENERATOR.TAKT daje brzinu naizmjenično od 6000 mm/min i -6000 mm/min. Isto predstavlja ponašanje motora na način da se svakih 0,4 s mijenja smjer okretanja vratila. Blok SEL što predstavlja riječ „Select“ na engleskom jeziku, radi na način kada je #CMD_HOMING_ACTIVE u FALSE, uzima vrijednosti iz

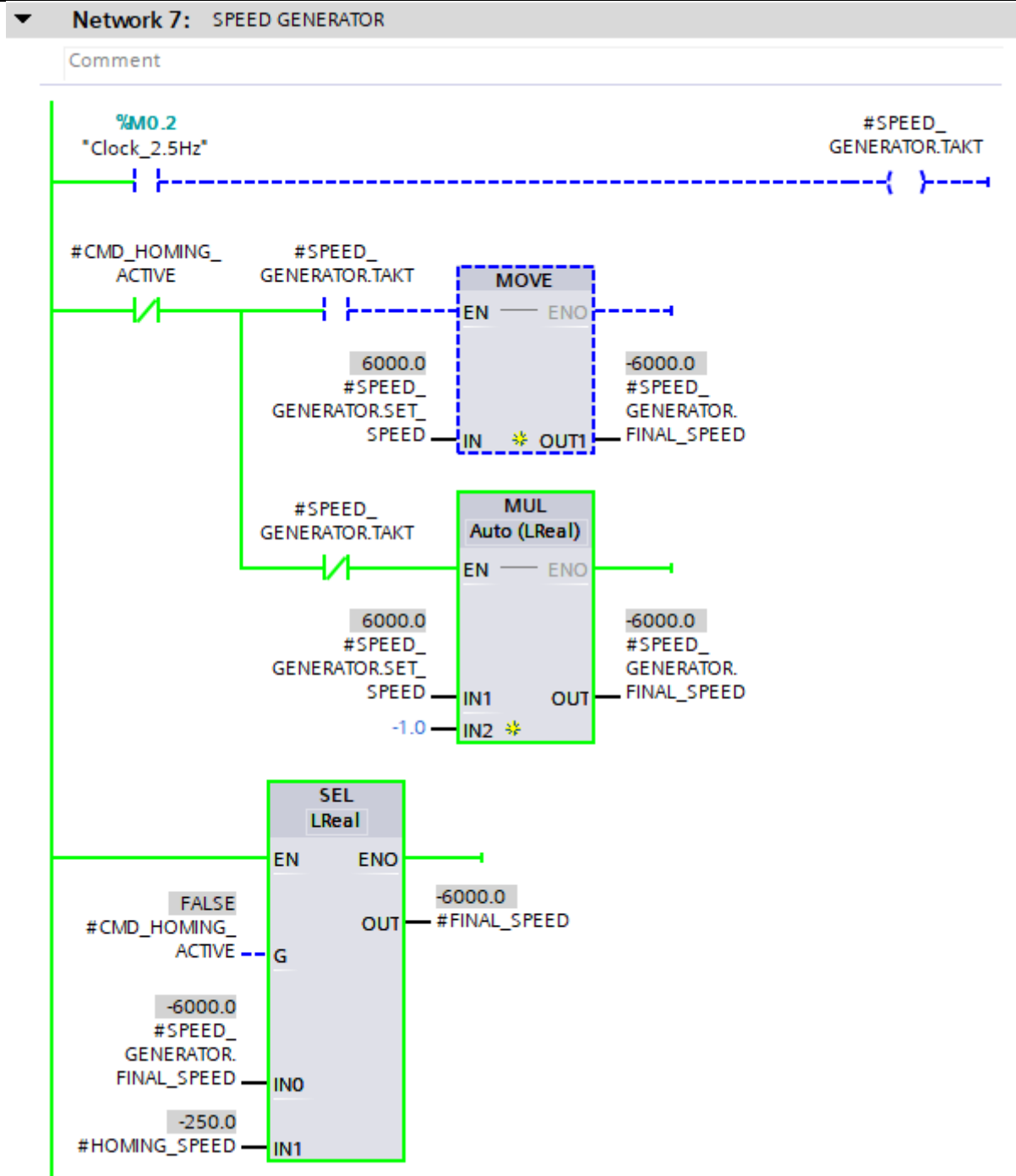
#SPEED_GENERATOR.FINAL_SPEED, a za vrijednost TRUE uzima vrijednost iz #HOMING_SPEED i šalje na #FINAL_SPEED.



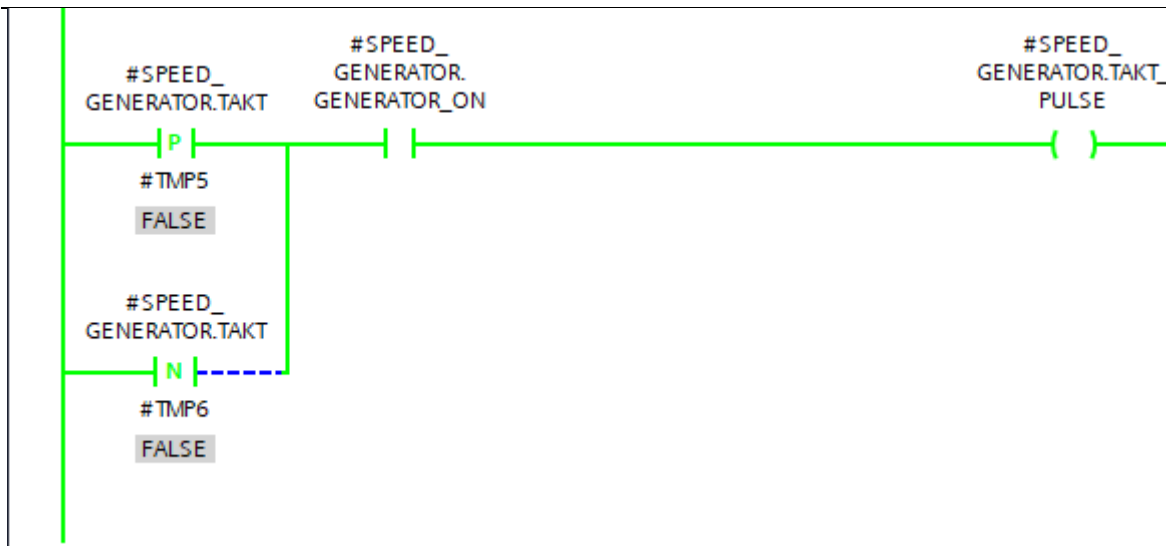
Slika 34. Generator brzine



Slika 35. Prikaz rada generatora brzine (1)

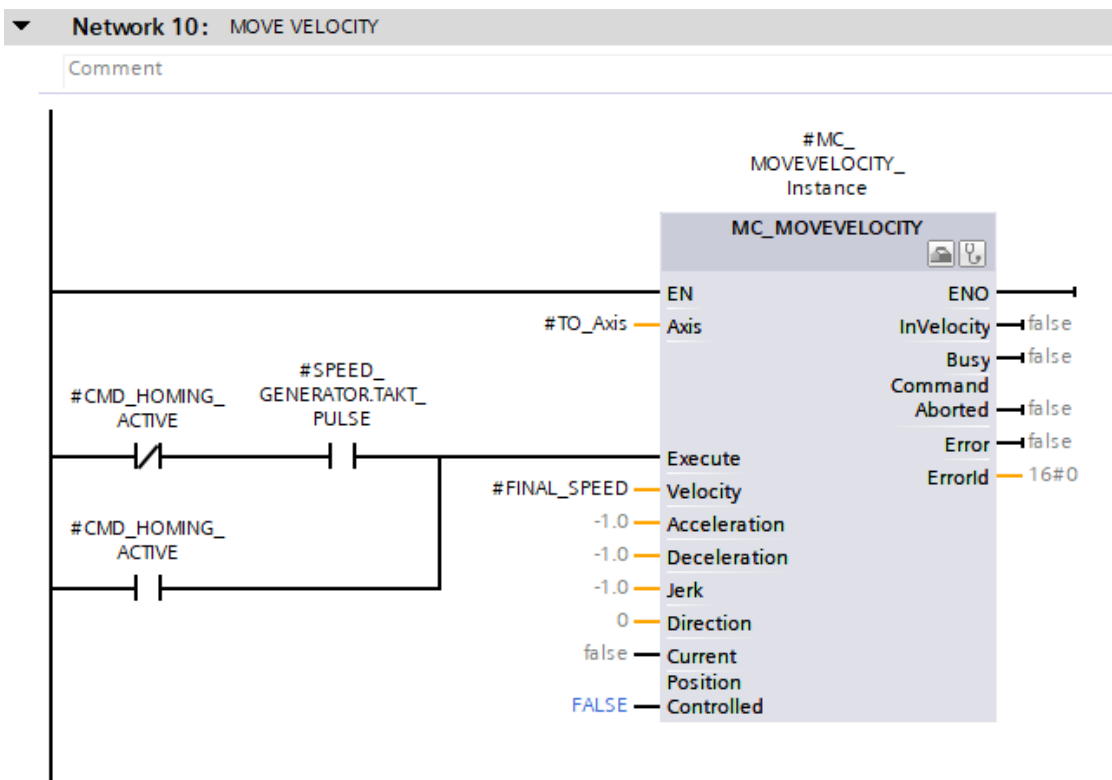


Slika 36. Prikaz rada generatora brzine (2)



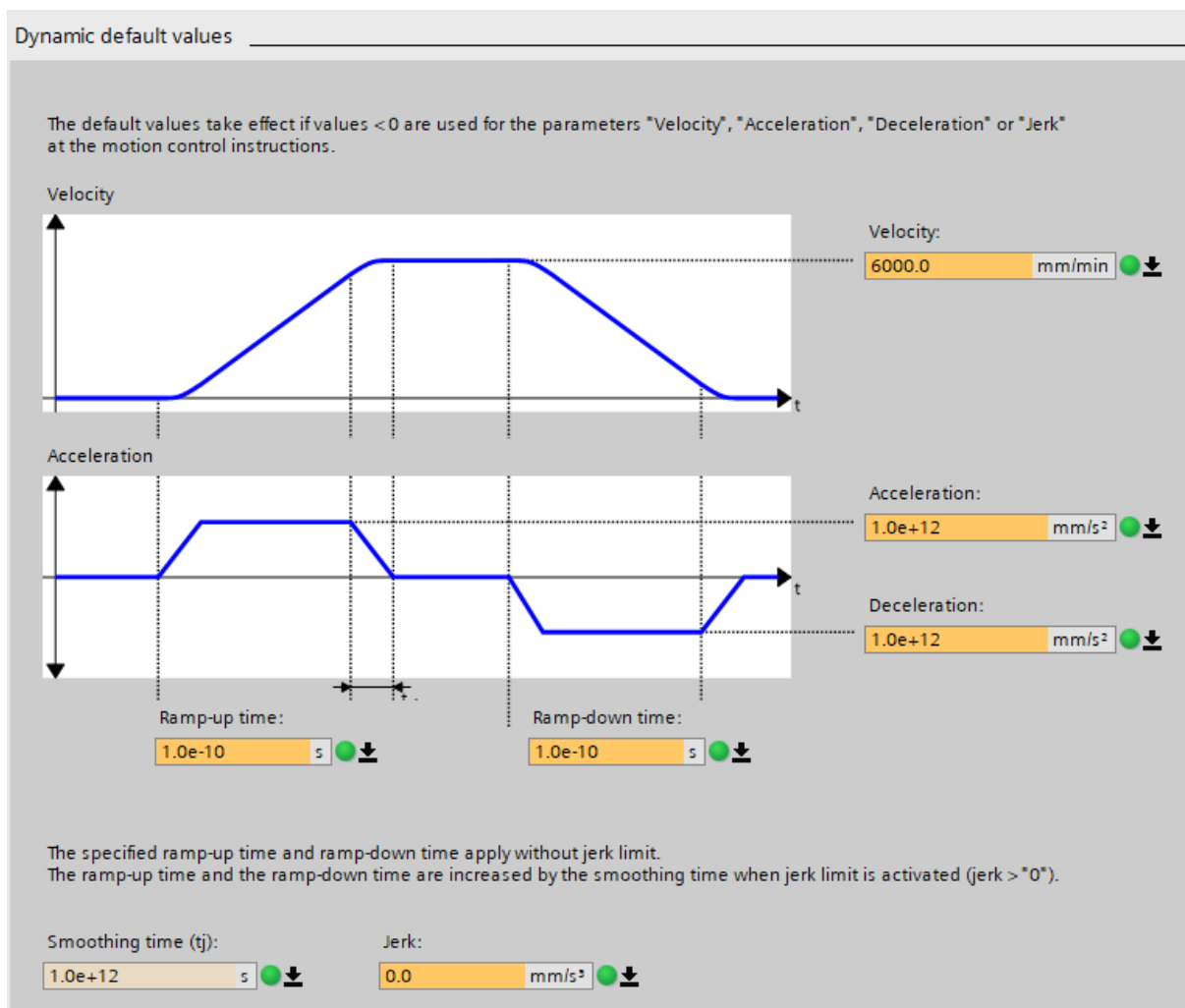
Slika 37. Prikaz rada generatora brzine (3)

Na [Slika 37] može se vidjeti da kada se omogući rad generatora brzine sa #SPEED_GENERATOR.GENERATOR_ON, da se za pozitivni i negativni brid od #SPEED_GENERATOR.TAKT aktivira izlaz #SPEED.GENERATOR.TAKT_PULSE koji kontrolira rad bloka #MC_MOVEVELOCITY, što je vidljivo i na [Slika 38]. Rad bloka ovisi i o #CMD_HOMING_ACTIVE na način da generator brzine može okidati blok #MC_MOVEVELOCITY samo u slučaju ako je #CMD_HOMING_ACTIVE u FALSE.



Slika 38. Network 10: MOVE VELOCITY

Nadalje, na [Slika 39] vidljivi su parametri pod „Dynamic default values“ u realnom vremenu koji se moraju postaviti točno na taj način prilikom pravilnog podešavanja regulacijskog kruga brzine vrtnje. Vidljivo je kako su akceleracija i deceleracija postavljene na maksimalnu vrijednost što posljedično smanjuje „ramp-up time“ i „ramp-down time“ na minimalnu vrijednost. Parametar „Jerk“ se postavlja na vrijednost nula kako bi smanjili zaobljenja kod odziva.

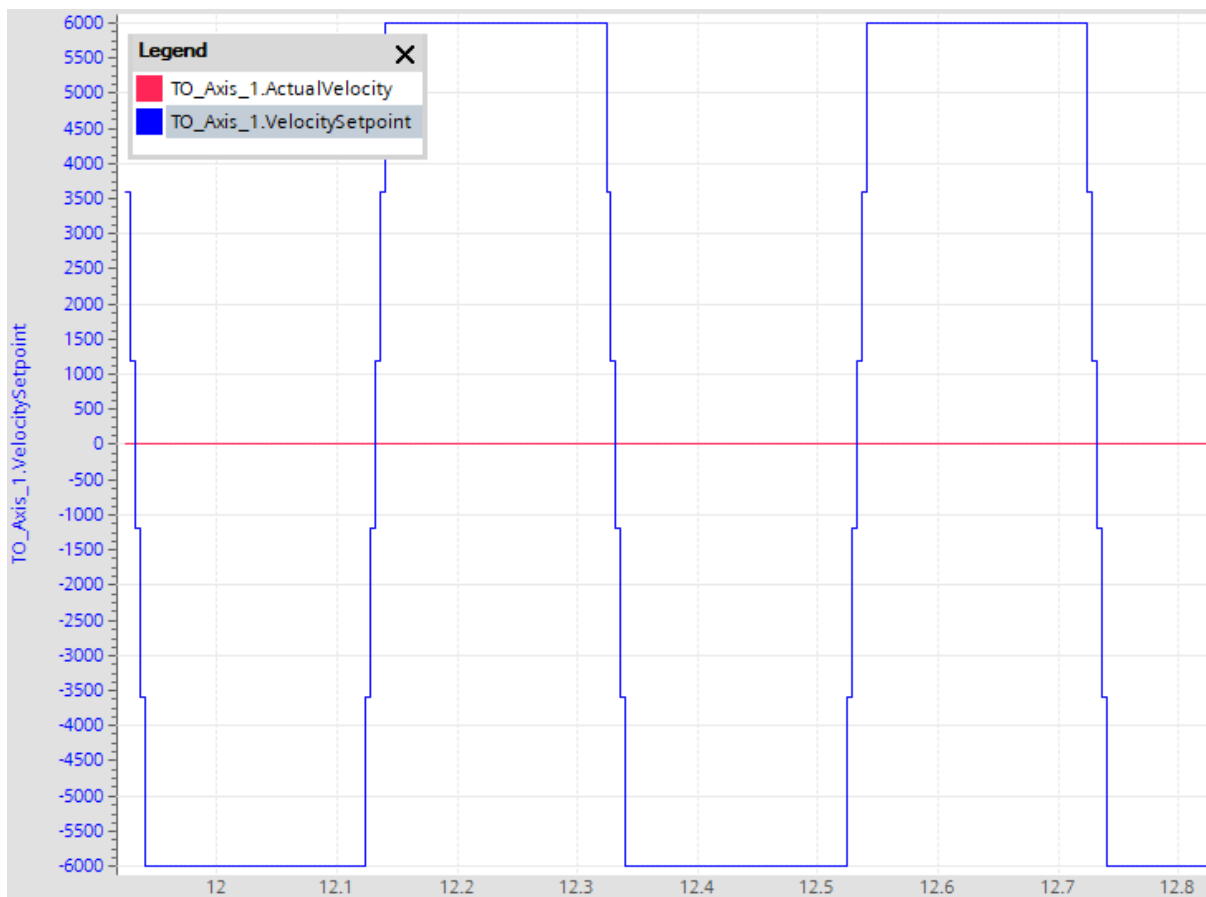


Slika 39. Parametri prilikom podešavanja regulacijske petlje brzine vrtnje

Unutar programa TIA Portal postoji opcija „Traces“ u kojoj se vrlo jednostavno mogu odabrati vrijednosti koje se žele pratiti na dijagramu, a navedeno smo iskoristili za usporedbu trenutne i referentne brzine vrtnje. Na [Slika 40] je vidljivo kako je prilikom postavljanja proporcionalnog pojačanja na vrijednost nula odziv također jednak nuli. Dakle, servo motor se u tom trenutku ne okreće. Stoga, može se jasno vidjeti dobivena STEP pobuda (plave boje) koja će biti referenca u daljnjim dijagramima. Prema legendi sa [Slika 40] vidljivo je kako odziv

crvene boje odgovara trenutnoj brzini servo motora, dok odziv plave boje odgovara postavljenoj referenci brzine.

Proporcionalnim i integralnim pojačanjem moguće je upravljati putem SIEMENS SINAMICS V-ASSISTANT odabirom na „Parameterize“ te zatim u padajućem izborniku odabirom na „View all parameters“. Parametri koje je potrebno fino ugađati su „Speed loop gain“ i „Speed loop integral time“. Na [Slika 41] vidljivi su iznosi proporcionalnog i integracijskog pojačanja za slučaj odziva sa [Slika 40].

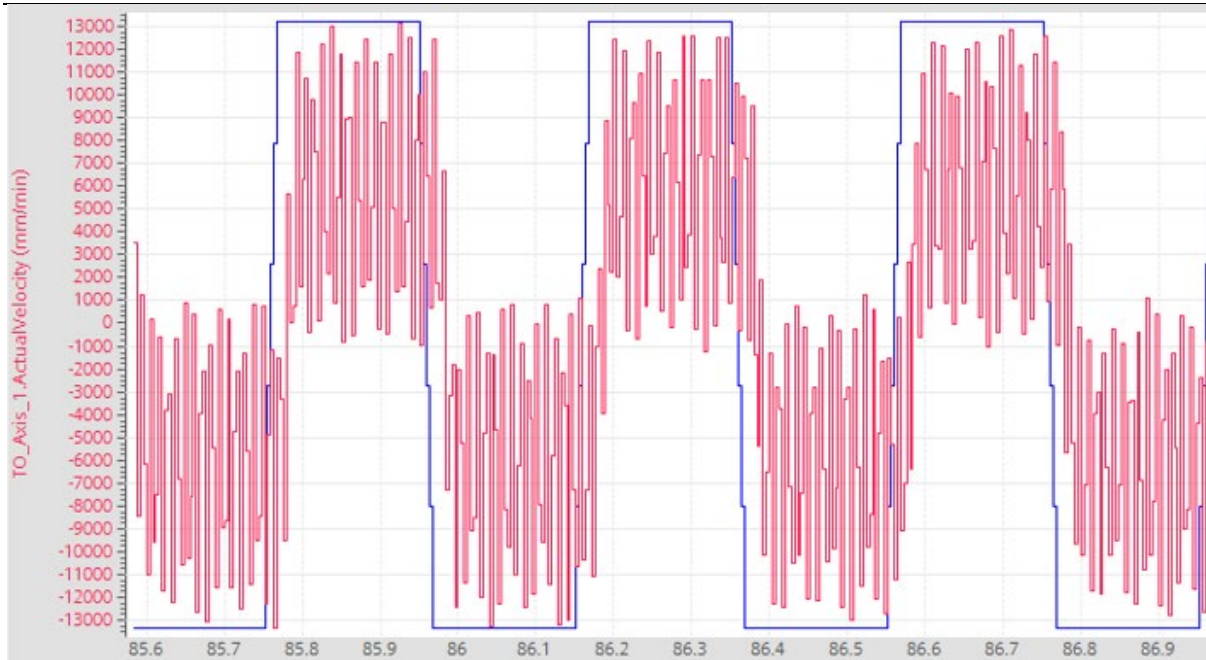


Slika 40. STEP pobuda

| | | |
|--------------------------|---------|---------|
| Speed loop gain | 0.0000 | Nms/rad |
| Speed loop integral time | 30.0000 | ms |

Slika 41. Pojačanja za slučaj odziva na [Slika 40]

U slučaju kao na [Slika 42] dolazi od tzv. proklizavanja servo motora koje je popraćeno glasnim zvukom i nestabilnim radom motora uz izrazite oscilacije, kao što je vidljivo prema odzivu crvene boje. Navedeno se događa zbog izrazito malog iznosa integracijskog djelovanja, zbog čega servo motor nema stacionarnu točnost. Na [Slika 43] vidljivi su iznosi proporcionalnog i integracijskog pojačanja za slučaj odziva sa [Slika 42].

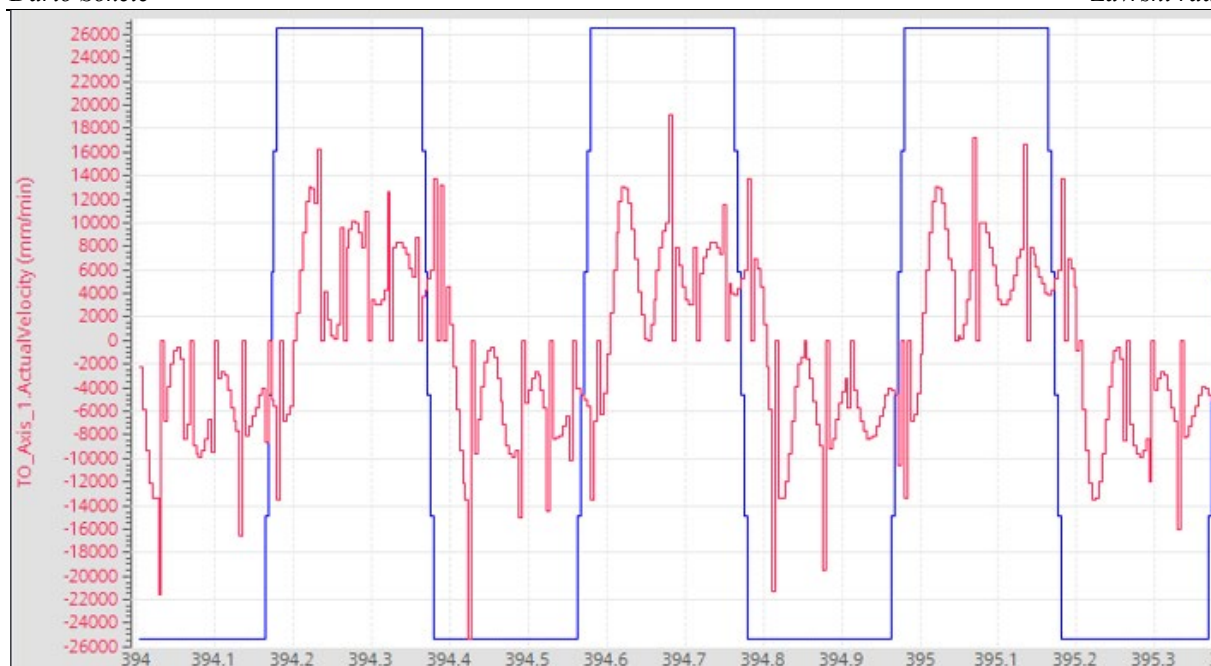


Slika 42. Proklizavanje servo motora

| | | |
|--------------------------|--------|---------|
| Speed loop gain | 0.0018 | Nms/rad |
| Speed loop integral time | 1.0000 | ms |

Slika 43. Pojačanja za slučaj odziva na [Slika 42]

Na [Slika 44] vidljiv je odziv servo motora za red manjeg proporcionalnog pojačanja u odnosu na slučaj sa [Slika 42]. Vidljivo je kako je odziv nestabilan i motor ne može pratiti željenu referencu brzine vrtnje. U praktičnom dijelu rada se prvo isprobao odziv s pojačanjima prema [Slika 44] te se zatim pojačalo proporcionalno djelovanje koje je dovelo do izrazito nestabilnog ponašanja i proklizavanja motora.



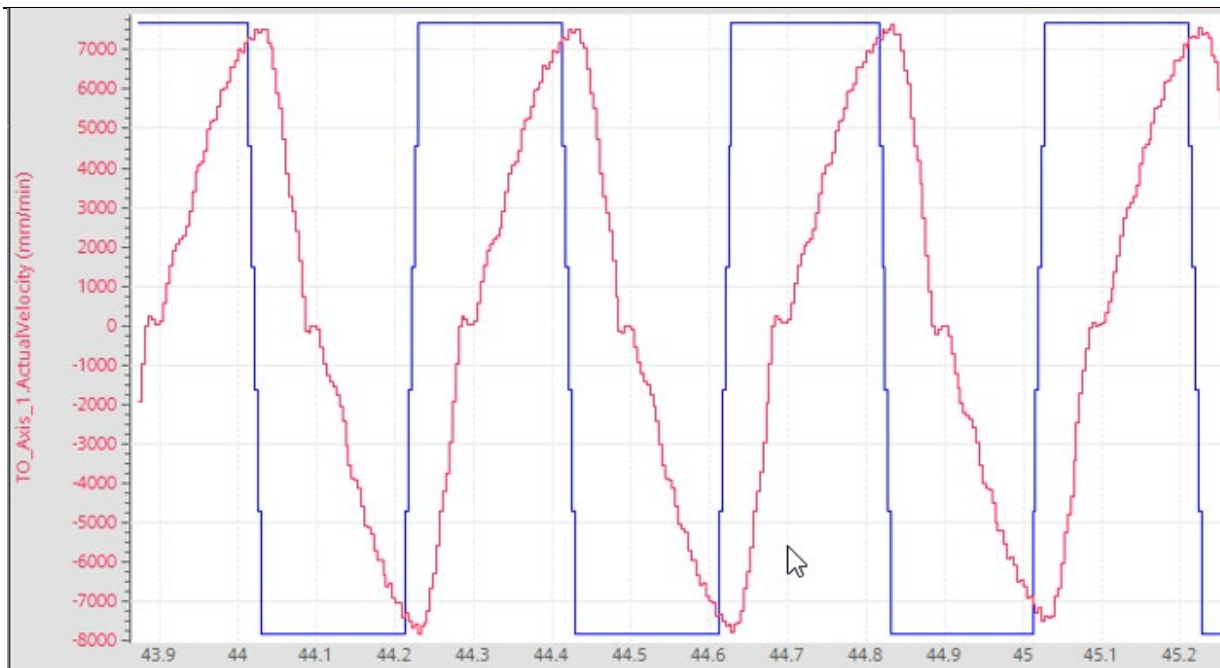
Slika 44. Nestabilan odziv na pobudu

| | | |
|--------------------------|--------|---------|
| Speed loop gain | 0.0001 | Nms/rad |
| Speed loop integral time | 1.0000 | ms |

Slika 45. Pojačanja za slučaj odziva na [Slika 44]

Pojava stacionarnog odstupanja je vrlo česta ukoliko je integracijska vremenska konstanta PI regulatora velikog iznosa (malo integracijsko pojačanje) zbog čega integralni član sporo integrira regulacijsku pogrešku pa za dinamički pobudni signal na slici ne stigne izregulirati pogrešku regulacije na nulu. Dodatno, ako je i proporcionalno pojačanje malog iznosa, moguća je pojava niskofrekvencijskih oscilacija. Naime, otvoreni regulacijski krug u sebi sadrži dva integracijska djelovanja: ono uslijed momenta inercije pogona (brzina je vremenski integral momenta ubrzanja podijeljenog sa inercijom) i samog regulatora. Ovo pak znači da ako parametri PI regulatora (proporcionalno pojačanje i integralna vremenska konstanta) nisu dobro podešeni, regulacijski sustav će biti sklon oscilacijama: niskofrekvencijskim pri malim iznosima proporcionalnog pojačanja, a visokofrekvencijskim pri velikim iznosima proporcionalnog pojačanja.

Na [Slika 46] vidljiv je odziv servo motora za slučaj u kojem se integralno djelovanje povećalo. Može se primjetiti kako je odziv stabilniji nego ranije, no i dalje to nije željeno ponašanje servo motora jer u ovom slučaju izostaje adekvatno proporcionalno djelovanje. Na [Slika 47] vidljivi su iznosi proporcionalnog i integracijskog pojačanja za slučaj odziva sa [Slika 46].

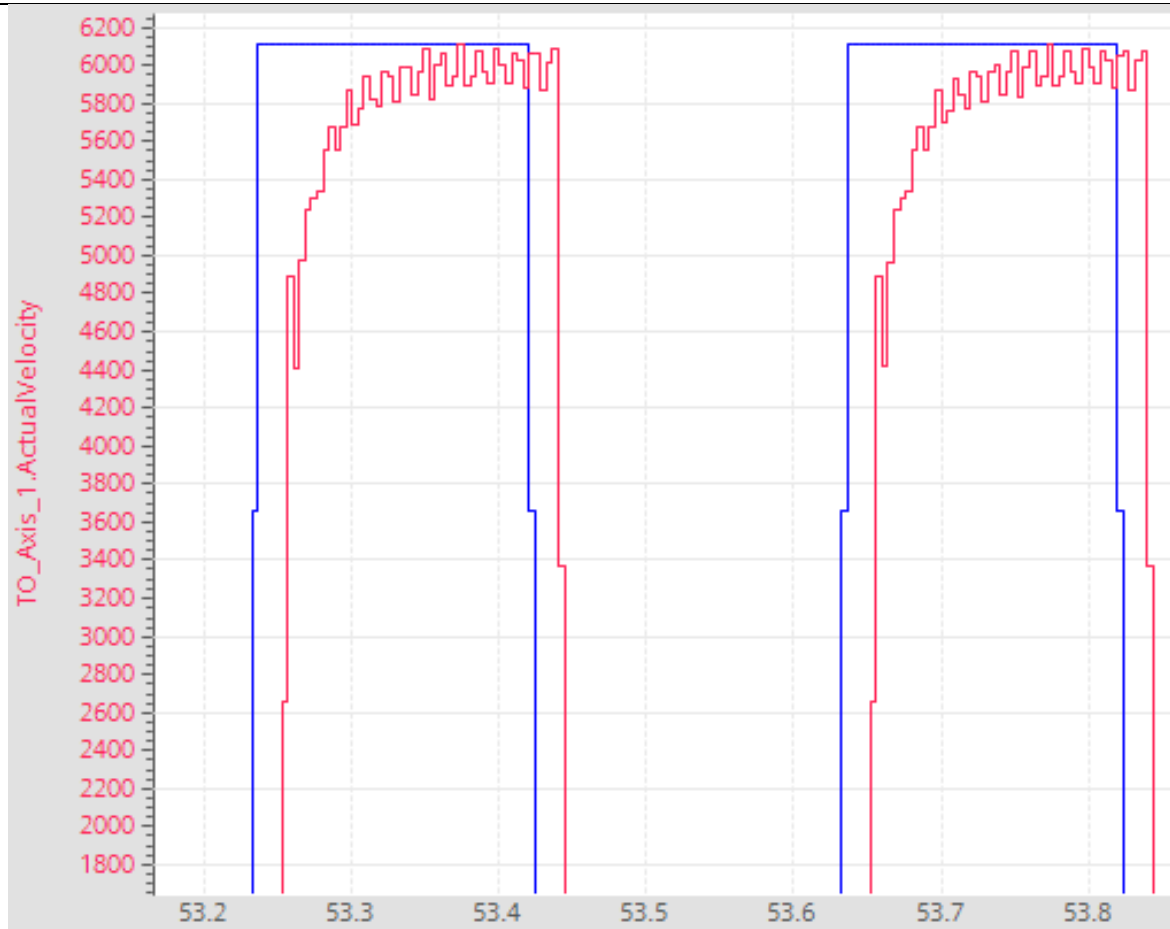


Slika 46. Utjecaj povećanja integralnog pojačanja

| | | |
|--------------------------|---------|---------|
| Speed loop gain | 0.0001 | Nms/rad |
| Speed loop integral time | 30.0000 | ms |

Slika 47. Pojačanja za slučaj odziva na [Slika 46]

Prema [Slika 48] može se primjetiti kako na ranije već povećano integracijsko djelovanje, dodavanjem i proporcionalnog djelovanja, dobiva se već približno točan odziv, no nedovoljno točan da bi bio zadovoljavajući. Dolazi do oscilacija odziva uz propadanje brzine, što nikako nije poželjno. Razlog tomu je preveliko narinuto proporcionalno pojačanje. Na [Slika 49] vidljivi su iznosi proporcionalnog i integracijskog pojačanja za slučaj odziva sa [Slika 48].

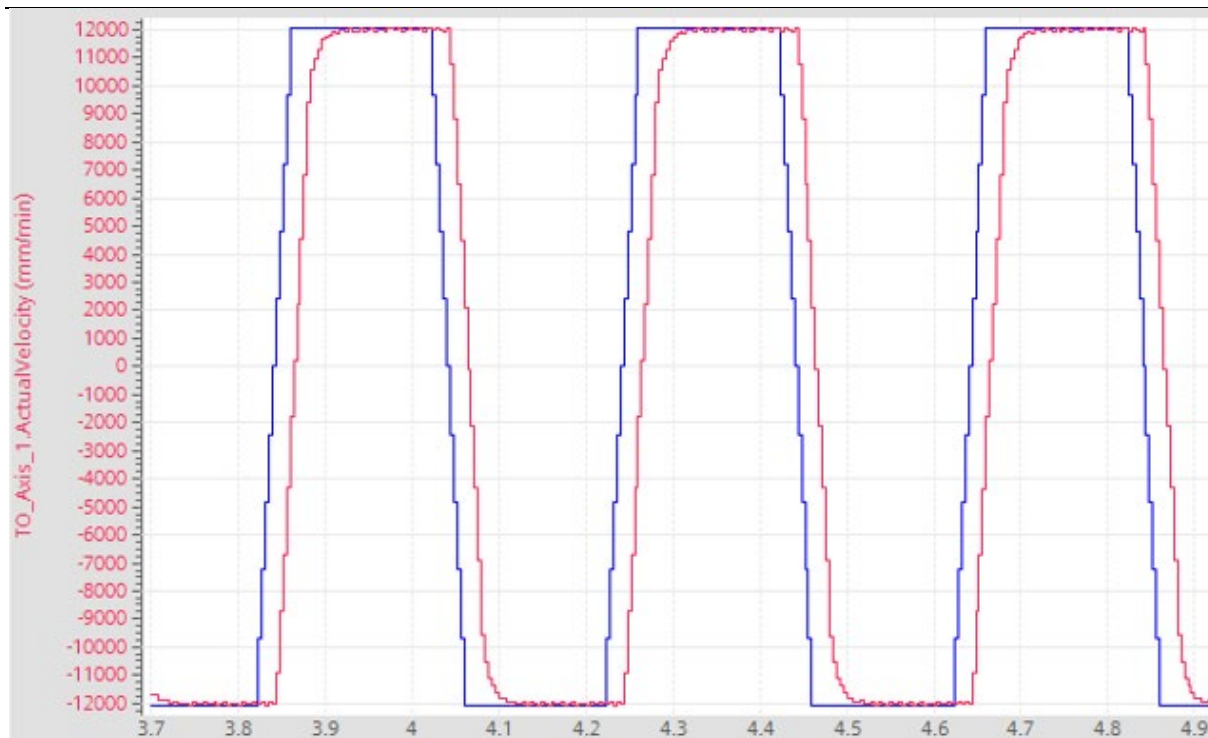


Slika 48. Utjecaj povećanja proporcionalnog djelovanja

| | | |
|--------------------------|---------|---------|
| Speed loop gain | 0.0050 | Nms/rad |
| Speed loop integral time | 30.0000 | ms |

Slika 49. Pojačanja za slučaj odziva na [Slika 48]

Konačno, na [Slika 50] može se vidjeti zadovoljavajući odziv brzine servo motora s obzirom na zadanu referencu brzine vrtnje. Smanjenjem proporcionalnog pojačanja smanjile su se oscilacije pri referentnoj vrijednosti brzine te uklonjena su neželjena propadanja brzine. Također, odziv je izrazito brz uz kašnjenje od svega 52 ms.



Slika 50. Zadovoljavajući odziv na postavljenu referencu brzine vrtnje

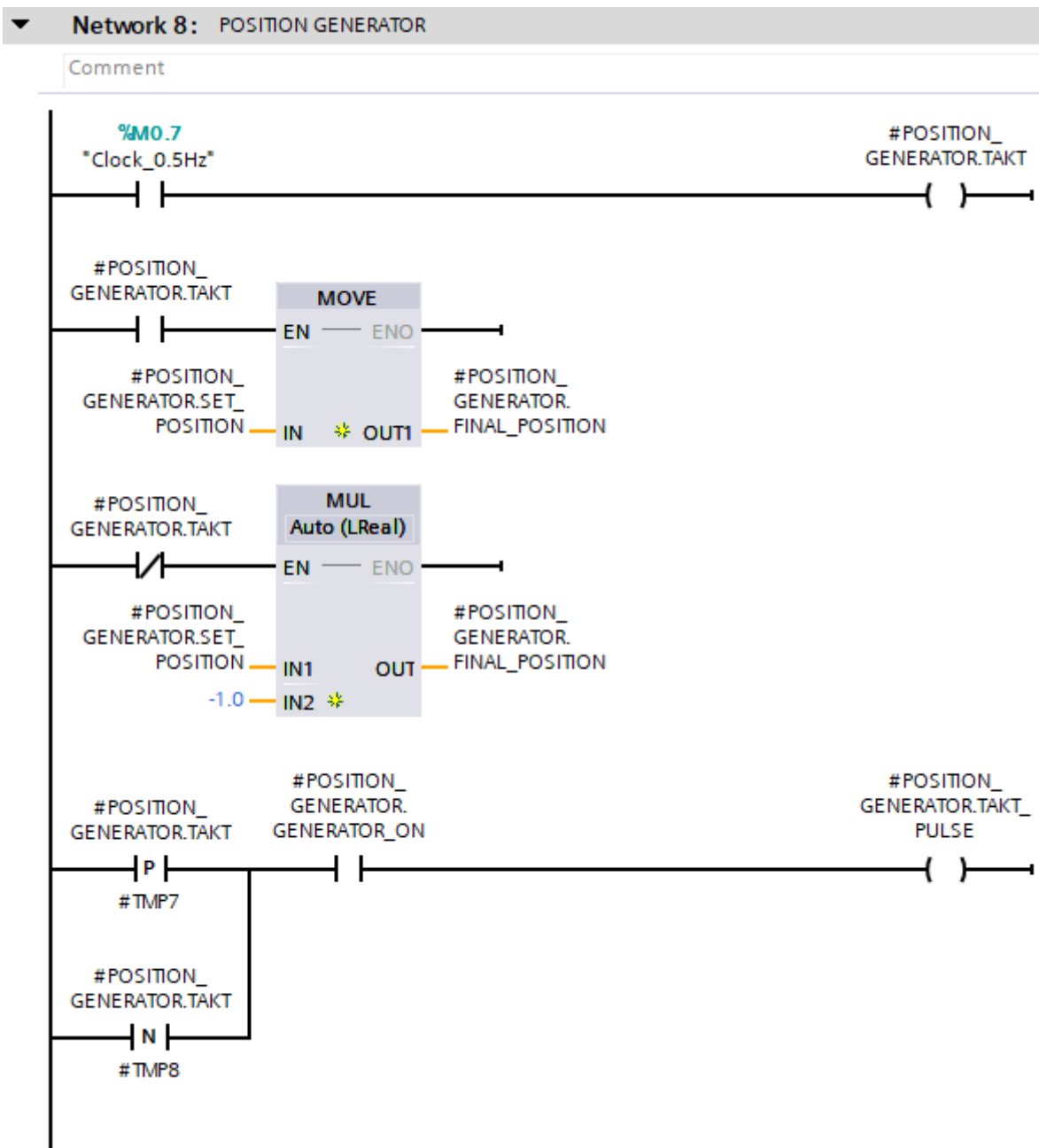
| | | |
|--------------------------|---------|---------|
| Speed loop gain | 0.0018 | Nms/rad |
| Speed loop integral time | 30.0000 | ms |

Slika 51. Pojačanja za slučaj odziva na [Slika 50]

Nakon postizanja ovakvog odziva, može se zaključiti kako je brzinska petlja sada ispravno podešena te se može krenuti na podešavanje regulacijske petlje pozicije.

3.4.5. Parametriranje regulacijske petlje pozicije servo motora

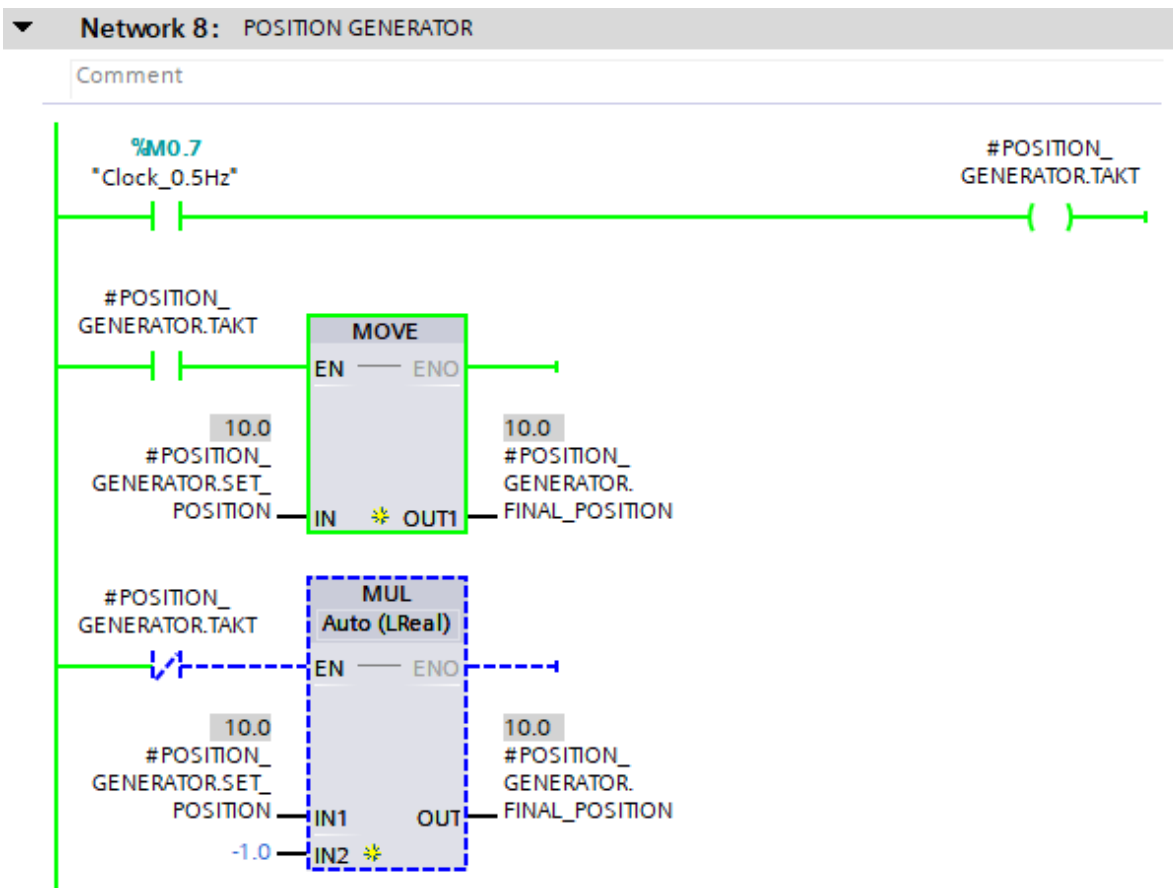
Slično kao i u [3.4.4], kako bi se parametriranje regulacijske petlje pozicije korištenog servo motora moglo točno provesti, kreiran je generator pozicije unutar programa TIA Portal. Na [Slika 52] prikazana je ladder logika za generator pozicije. Sama logika i rad generatora pozicije slijedi logiku i generatora brzine iz [3.4.4].



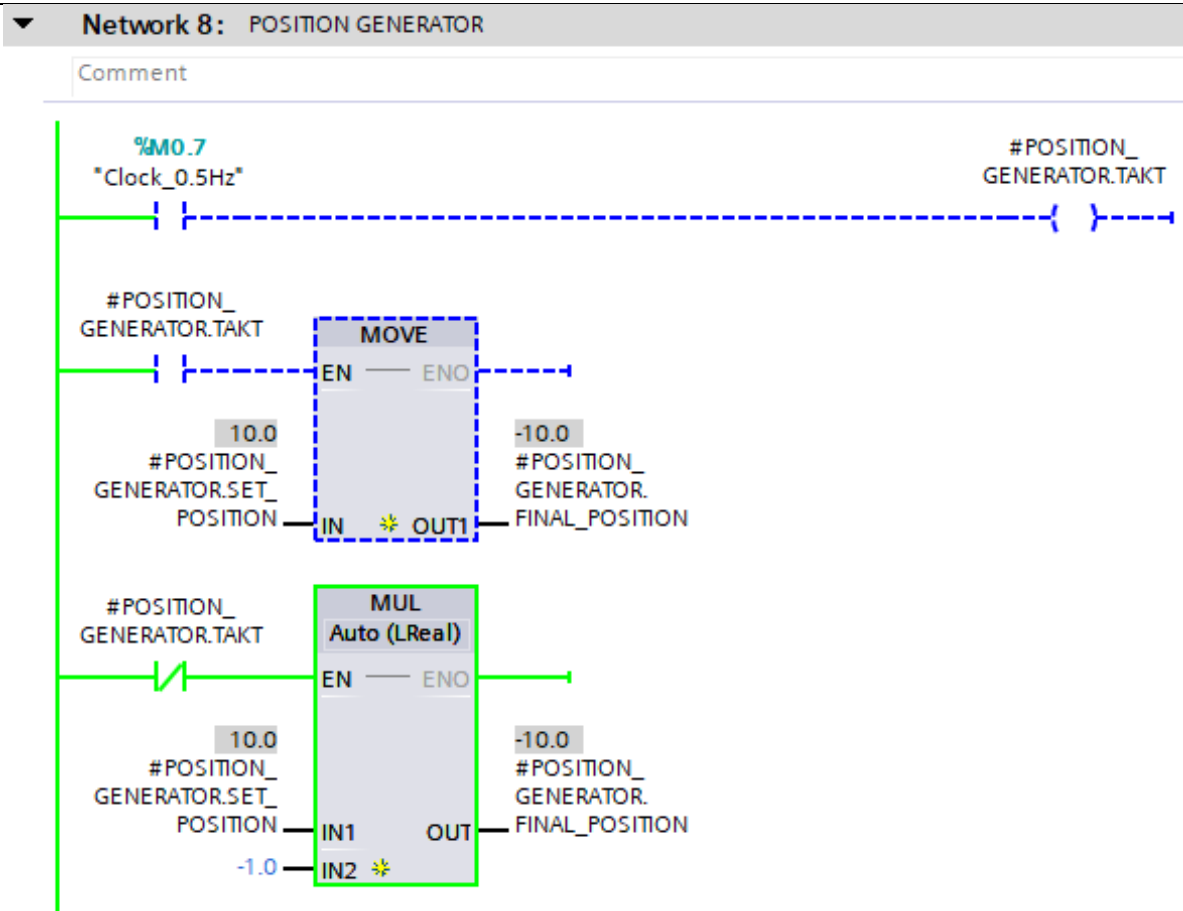
Slika 52. Generator pozicije – ladder logika

S varijablom `#POSITION_GENERATOR.SET_POSITION` unosi se željena vrijednost pomaka, dok se u varijablu `#POSITION_GENERATOR.FINAL_POSITION` upisuje pozitivna ili negativna vrijednost istog, ovisno o vrijednosti varijable

#POSITION_GENERATOR.TAKT. U slučaju kada pokrenemo rad generatora pozicije sa #POSITION_GENERATOR.GENERATOR_ON, za interval definiran frekvencijom s %M0.7 „Clock_0.5Hz“, dobiva se referenca pozicije kao što će to biti prikazano i na dijagramima. Na [Slika 53] i [Slika 54] vidljiv je „monitoring“ ladder logike.



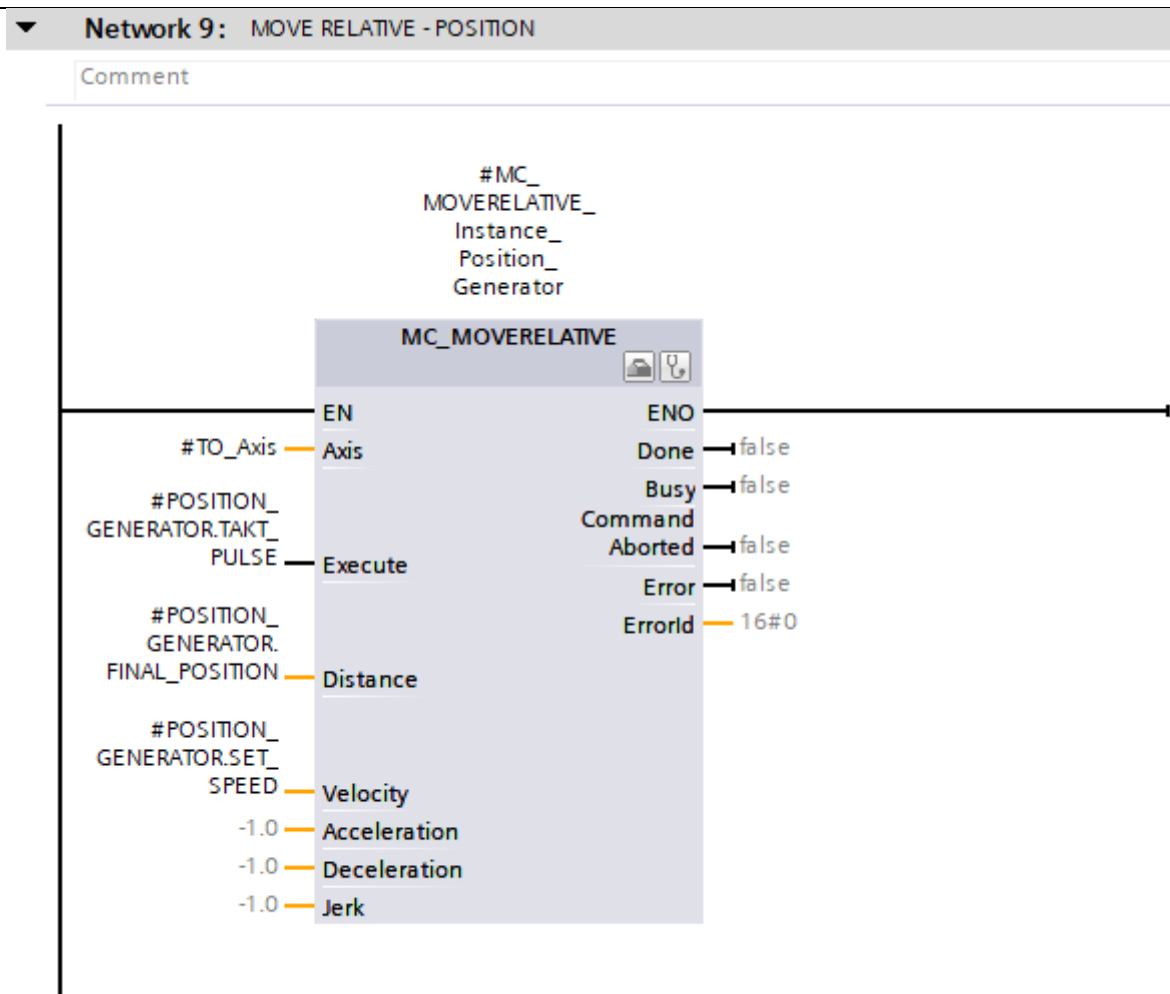
Slika 53. Prikaz rada generatora pozicije (1)



Slika 54. Prikaz rada generatora pozicije (2)

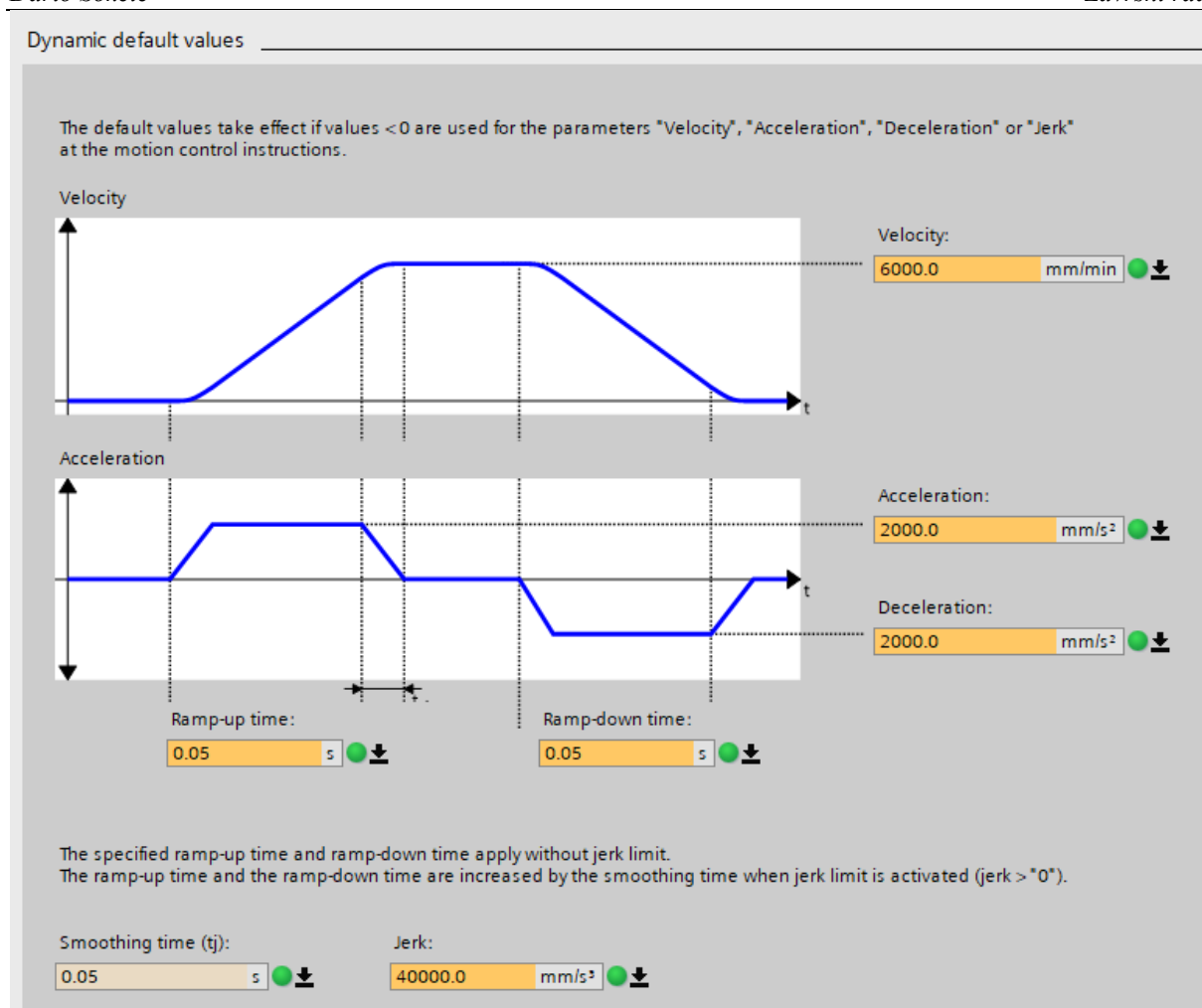
U slučaju kada je omogućen rad generatora pozicije, s varijablom `#POSITION_GENERATOR.TAKT_PULSE` vidljivom na [Slika 52], upravlja se okidanjem bloka `#MC_MOVERELATIVE_Instance_Position_generator` koji u ovom konkretnom primjeru daje naredbu servo motoru da se okrene za 10 mm te zatim u drugom smjeru za 10 mm. `#MC_MOVERELATIVE_Instance_Position_generator` vidljiv je na [Slika 55].

Također, s varijablom `#POSITION_GENERATOR.SET_SPEED` može se postaviti željena brzina za pomake koji će se izvršavati.



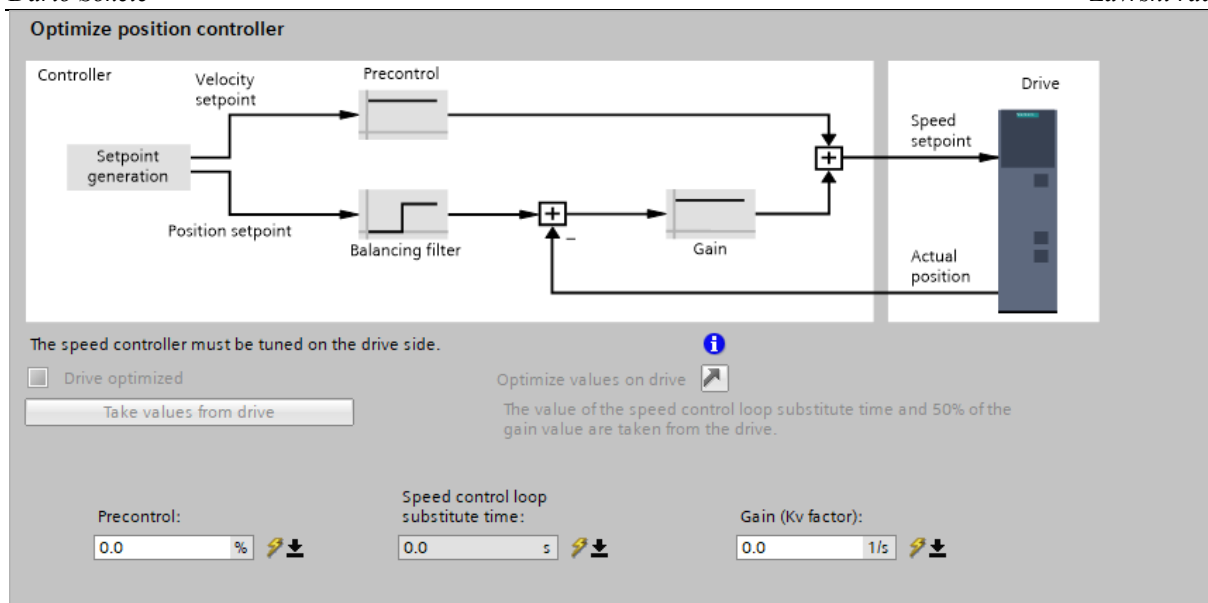
Slika 55. #MC_MOVERELATIVE potreban za generator pozicije

U praksi, podešavanje regulacijske petlje pozicije se kao i kod brzinske petlje odrađuje s teretom na vratilu servo motora kako bi pojačanja koja se podese odgovarala stvarnim uvjetima rada. U ovom slučaju će parametri tehnološkog objekta upravljanja pod „Dynamic default values“ biti podešeni kao što je vidljivo na [Slika 56]. Dakle, prilikom parametriranja brzinske petlje vrijednosti su bile postavljene u ekstreme kako bi se ispravno podesio regulacijski krug brzine vrtnje koji je neophodan za ispravan rad regulacijskog kruga pozicije. Sada prilikom podešavanja pozicijske petlje, parametri su postavljeni na realne vrijednosti kako bi pojačanja odgovarala realnoj primjeni.



Slika 56. Parametri prilikom podešavanja regulacijske petlje pozicije

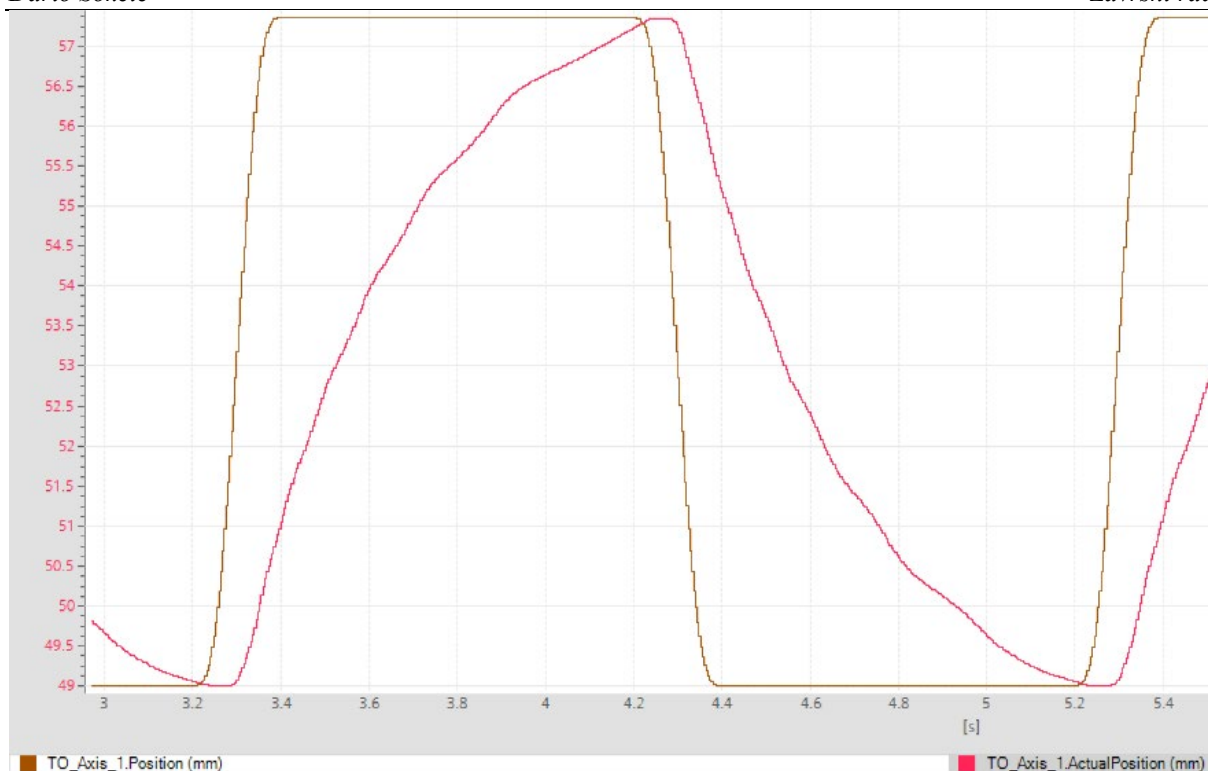
Nadalje, postupak parametriranja regulacijske petlje pozicije vrši se unutar TIA Portal-a odabirom na „Technology objects“, zatim odabirom željenog tehnološkog objekta te nakon toga odabirom „Commisioning“ i za kraj „Tuning“. S obzirom da je regulator pozicije P tipa, postoji jedan parametar s kojim se upravlja unutar programa, a to je „Gain (Kv factor)“ koji služi kako bi odziv imao stacionarnu točnost u odnosu na referencu, odnosno da vjerno prati referencu pozicije. Također, s obzirom da se u praksi zahtjevaju jako brzi odzivi na referencu pozicije i da pojačanje Kv unosi kašnjenje, koristi se i predupravljanje (eng. precontrol) koje utječe na brzinu odziva, odnosno ubrzava ga. Konačno rješenje je kompromis između ova dva parametra. Na [Slika 57] vidljiv je prikaz unutar Tia Portal-a.



Slika 57. Sučelje za podešavanje regulacijske petlje pozicije

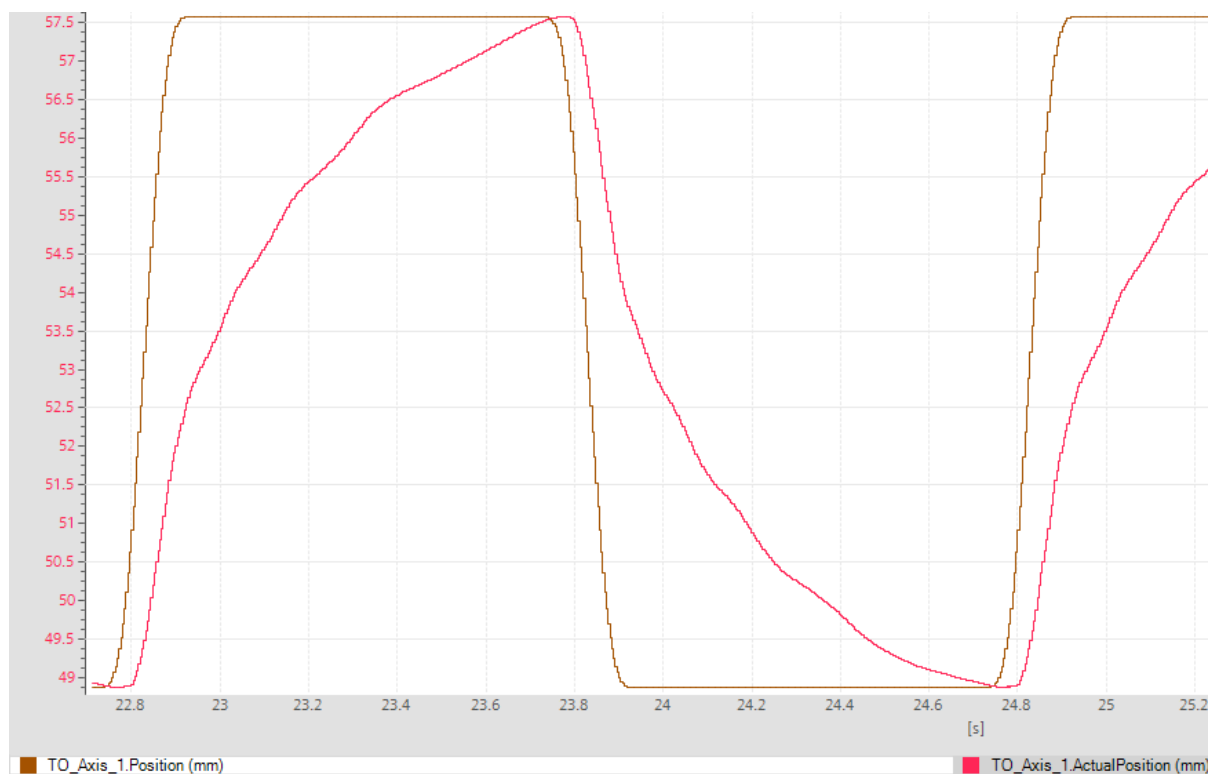
U nastavku će biti dani odzivi servo motora na referencu pozicije s različitim iznosima predupravljanja i proporcionalnog pojačanja iz kojih će moći biti vidljiv utjecaj vrijednosti s kojim upravljamo na konačni odziv. U legendi na svakoj slici je vidljivo kako je smeđom bojom prikazana referenca pozicije, a crvenom bojom stvarna pozicija osi servo motora.

Na [Slika 58] može se vidjeti kako s predupravljanjem od 10% i pojačanjem $K_v = 2,5 \text{ 1/s}$ servo motor ne može vjerno pratiti zadanu referencu pozicije te kako je kašnjenje u odzivu izrazito prisutno.



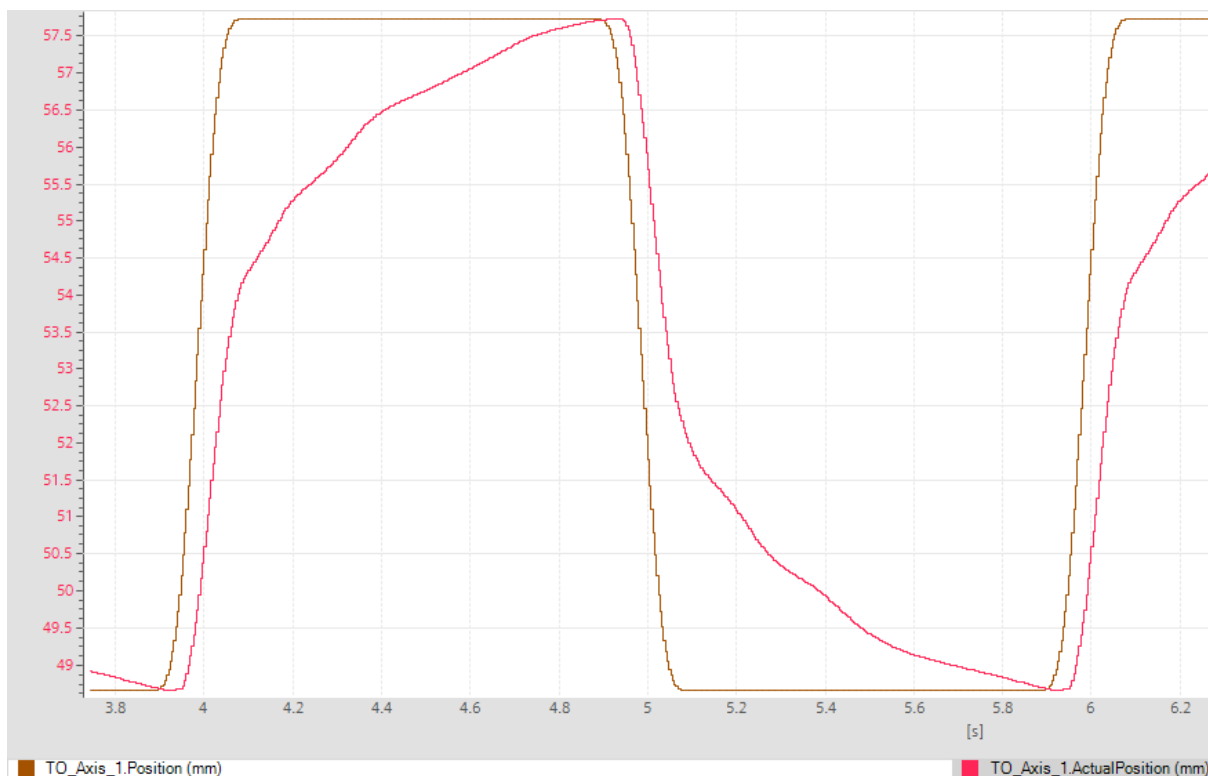
Slika 58. Odziv servo motora na referencu pozicije (1)

Za jednaki iznos pojačanja K_v , pojačan je utjecaj predupravljanja na 30% kako bi se vidio utjecaj na odziv.



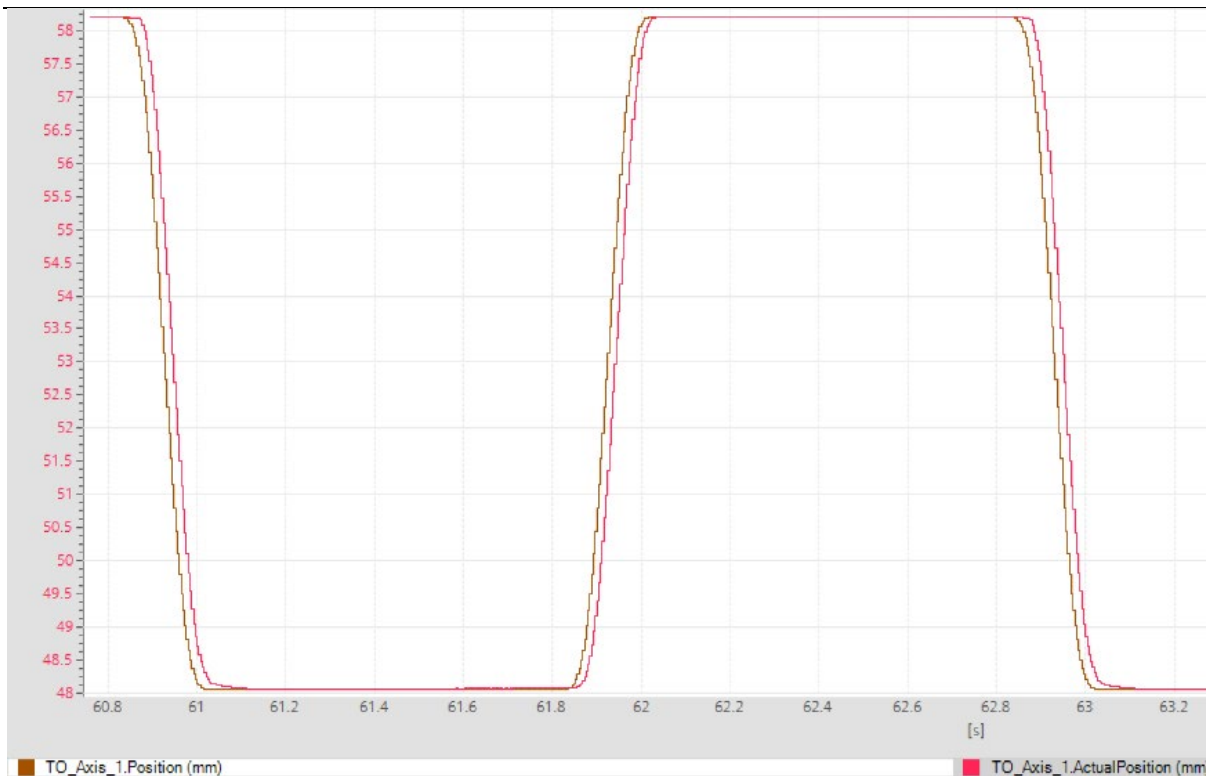
Slika 59. Odziv servo motora na referencu pozicije (2)

Na [Slika 59] je vidljivo kako je odziv i dalje netočan i s kašnjenjem, no u odnosu na odziv sa [Slika 58], može se primjetiti kako se kašnjenje već osjetljivo smanjilo. Na [Slika 60] je to još izrazitije vidljivo gdje je za isti iznos pojačanja, predupravljanje povećano na 50%. Može se primjetiti kako odziv počinje dobivati tzv. „oblik koljena“ što je i konačni cilj prilikom parametriranja regulacijske petlje pozicije.



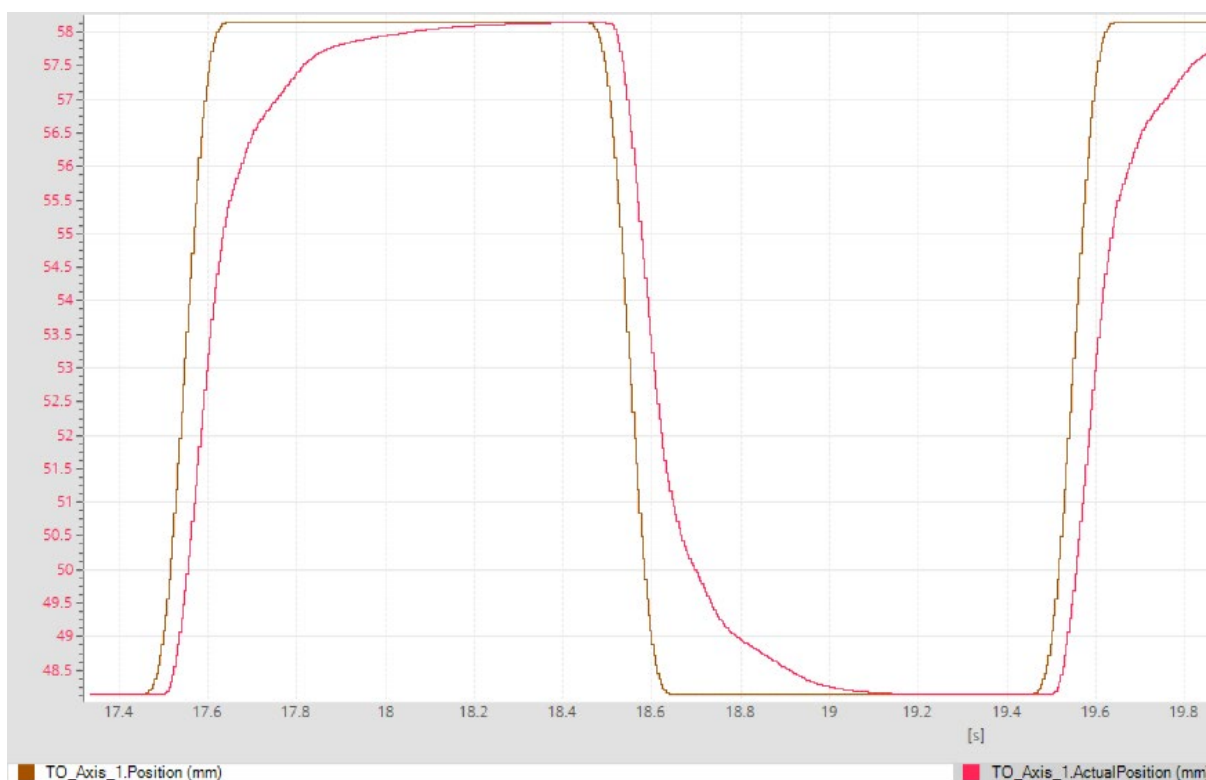
Slika 60. Odziv servo motora na referencu pozicije (3)

Povećanjem iznosa predupravljanja na 100% (s čime se u praksi i kreće u parametriranje pozicijske petlje), dobiva se točan odziv, bez nadvišenja i s minimalnim kašnjenjem. Odziv je vidljiv na [Slika 61].



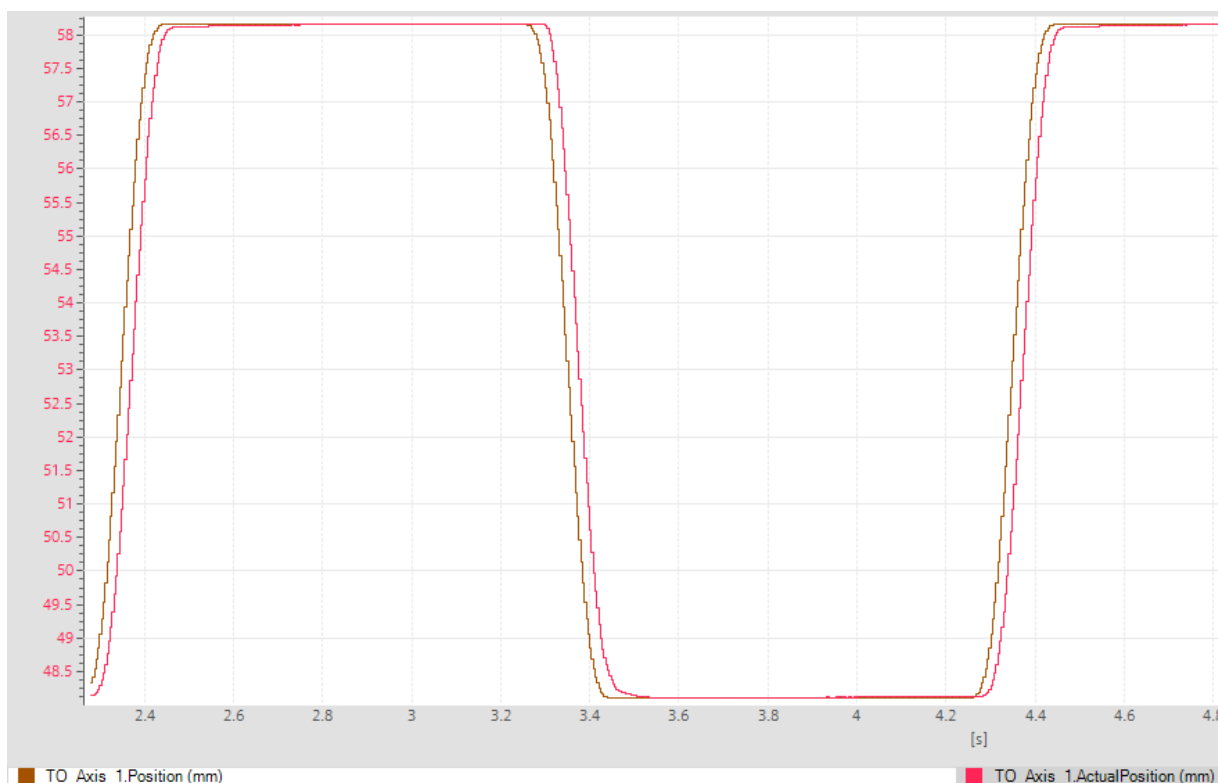
Slika 61. Odziv servo motora na referencu pozicije (4) – prihvatljiv odziv

S obzirom da kod odziva na [Slika 60] servo motor ne prati s točnošću željenu referencu, za isti iznos predupravljanja od 50%, povećan je iz pojačanja K_v , na iznos $K_v = 7$ 1/s.



Slika 62. Odziv servo motora na referencu pozicije (5)

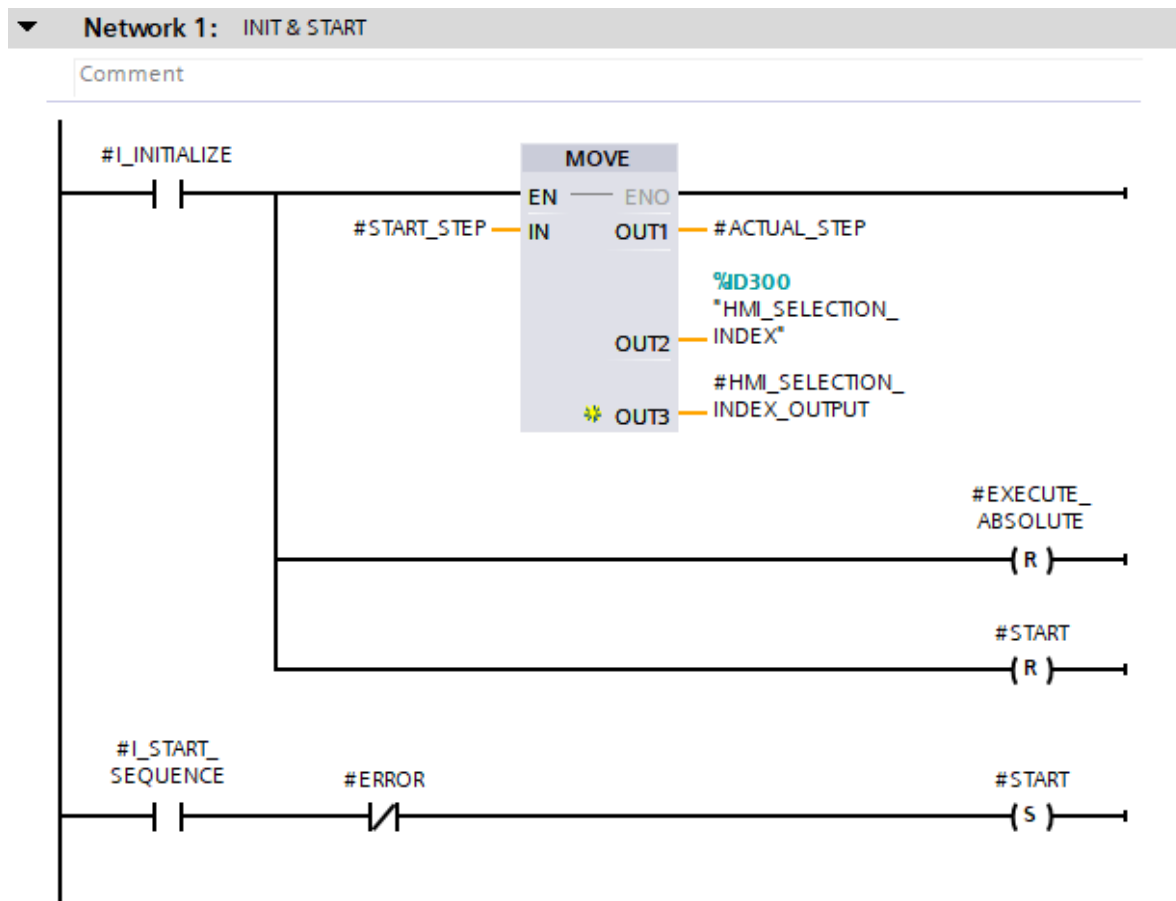
Na [Slika 62] može se vidjeti kako se servo motor počeo približavati zadanoj referenci te da u konačnici postiže iznos reference, no s kašnjenjem. Povećavanjem iznosa predupravljanja na 100% došlo je do prebačaja u odzivu, što kod pozicioniranja nikako nije poželjno (dolazi do udara). Stoga, kao što je i ranije navedeno, kompromisom između predupravljanja i pojačanja, s iznosima od 90% i $K_v = 7$ 1/s, ponovno se dobiva odziv izvrsnih karakteristika, kao i na [Slika 61]. S time se želi pokazati kako je moguće imati više varijantni rješenja prilikom podešavanja parametara s kojima se upravlja odzivom servo motora. Konačni odziv je vidljiv na [Slika 63].



Slika 63. Odziv servo motora na referencu pozicije (6) – prihvatljiv odziv

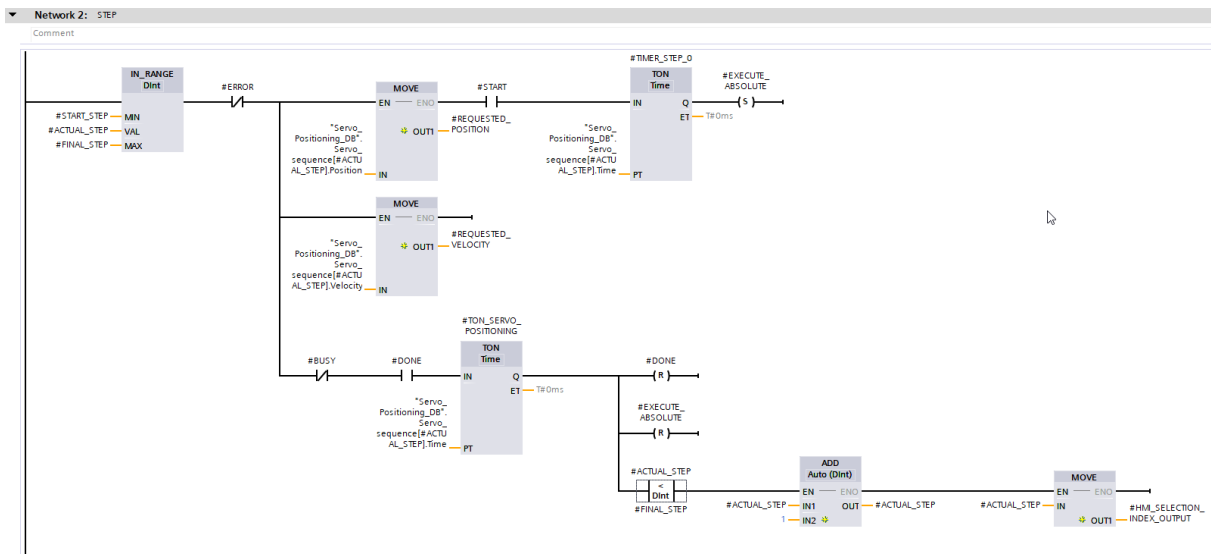
3.4.6. Izrada upravljačke sekvence za servo-pozicioniranje osi

Sljedeći zadatak unutar funkcijskog bloka za upravljanje radom servo motora je izrada recepta (upravljačke sekvence) za servo-pozicioniranje osi. Za navedeno se kreirao dodatni funkcijski blok SEQUENCER_FB kojeg se poziva unutar funkcijskog bloka SERVOMOTOR FB. Funkcijski blok SEQUENCER_FB je sačinjen od četiri dijela koji su prikazani na [Slika 64], [Slika 65], [Slika 68] i [Slika 69], a prikaz pozvanog funkcijskog bloka SEQUENCER_FB unutar funkcijskog bloka SERVOMOTOR FB vidljiv je na [Slika 70].

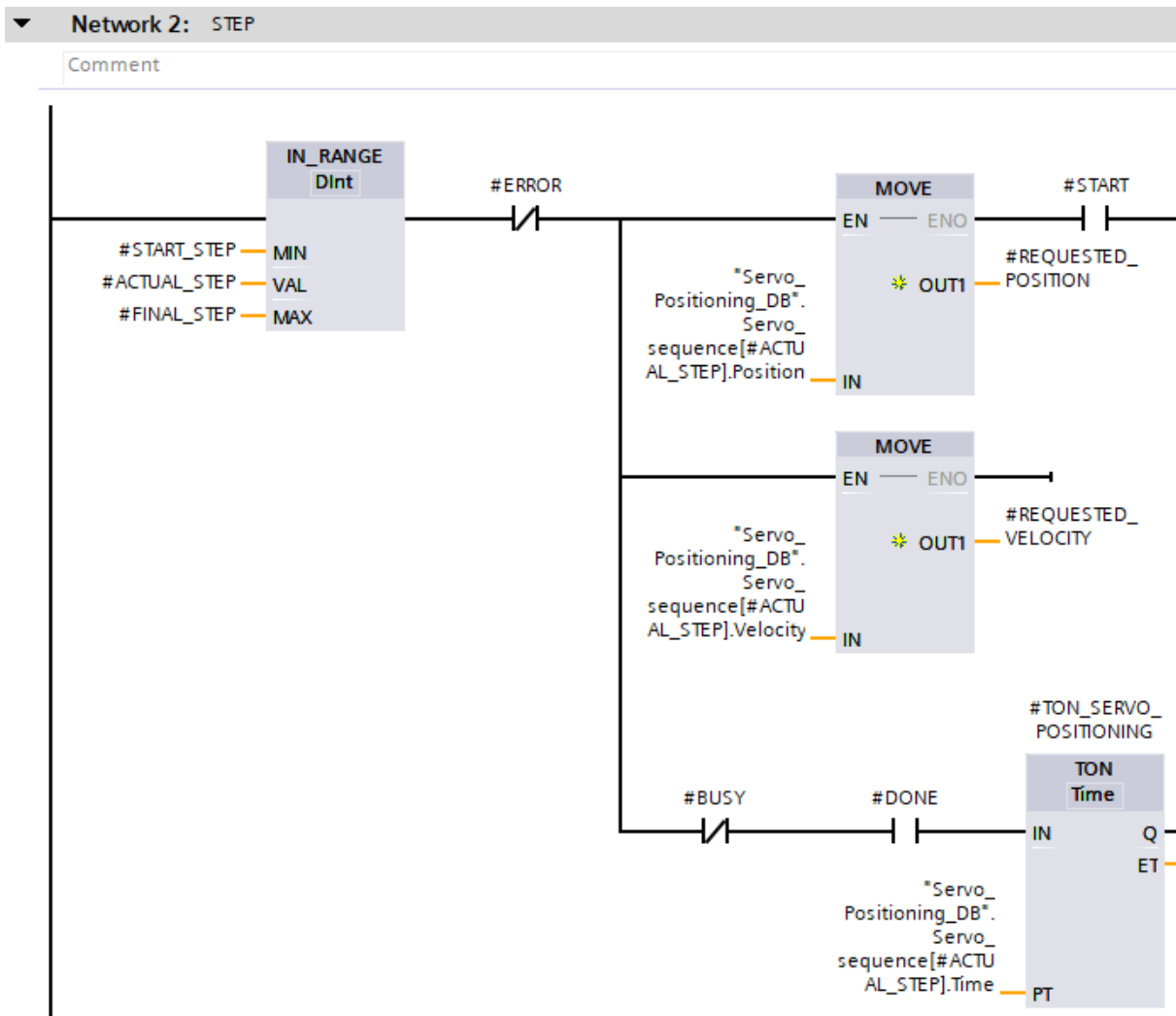


Slika 64. SEQUENCER_FB: Network1

Za početak, #I_INITIALIZE označava ulaz s kojim inicijaliziramo blok na željene početne vrijednosti. Na taj ulaz je spojen gumb čijim pritiskom se omogućava inicijalizacija. S blokom MOVE početni korak #START_STEP, koji je definiran od strane operatera na HMI uređaju, postavljamo i u varijable koje su na izlazu MOVE bloka. Također, inicijalizacijom se i upravljačke varijable #EXECUTE_ABSOLUTE i #START resetiraju na FALSE stanje. Sekvenca započinje kada nema nikakvih grešaka, a to se provjerava s varijablom #ERROR te kada je #I_START_SEQUENCE u ON stanju. Tada se upravljačka varijabla #START setira u ON stanje.

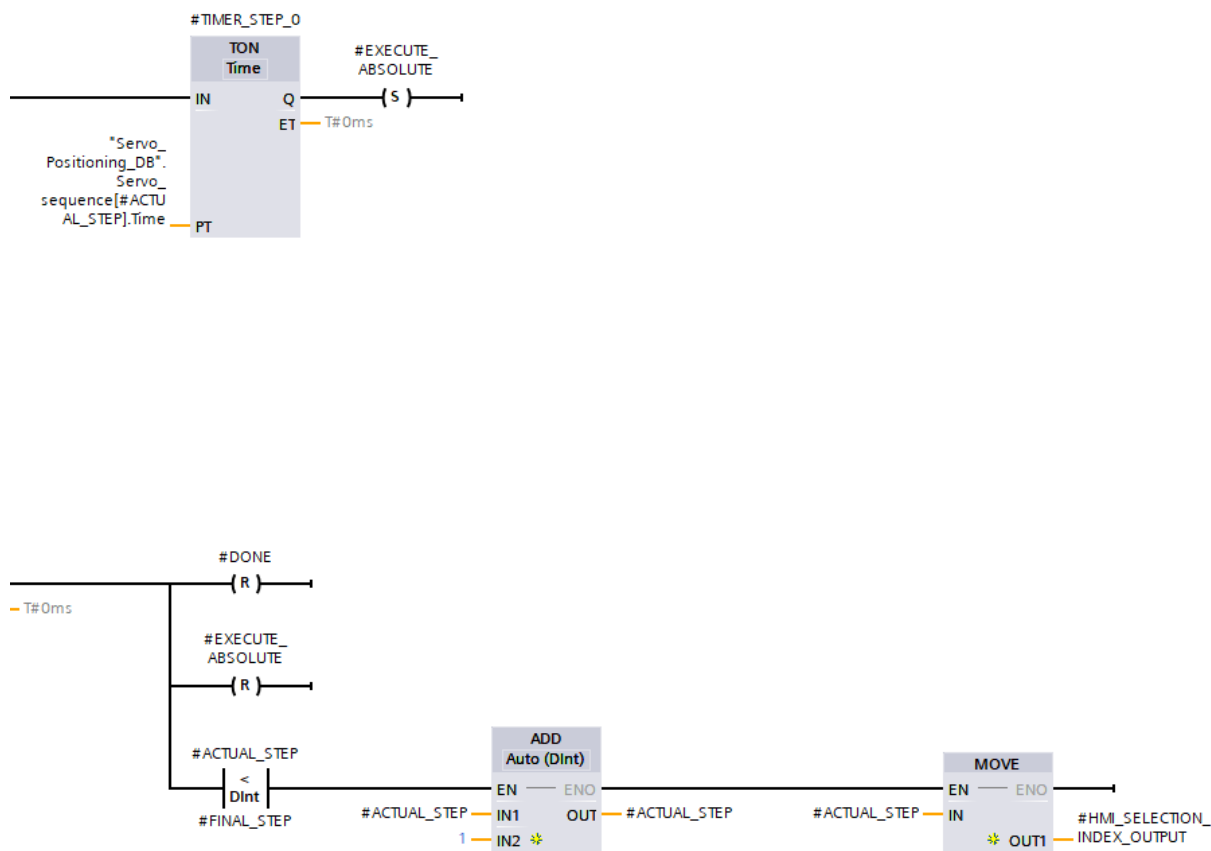


Slika 65. SEQUENCER_FB: Network2

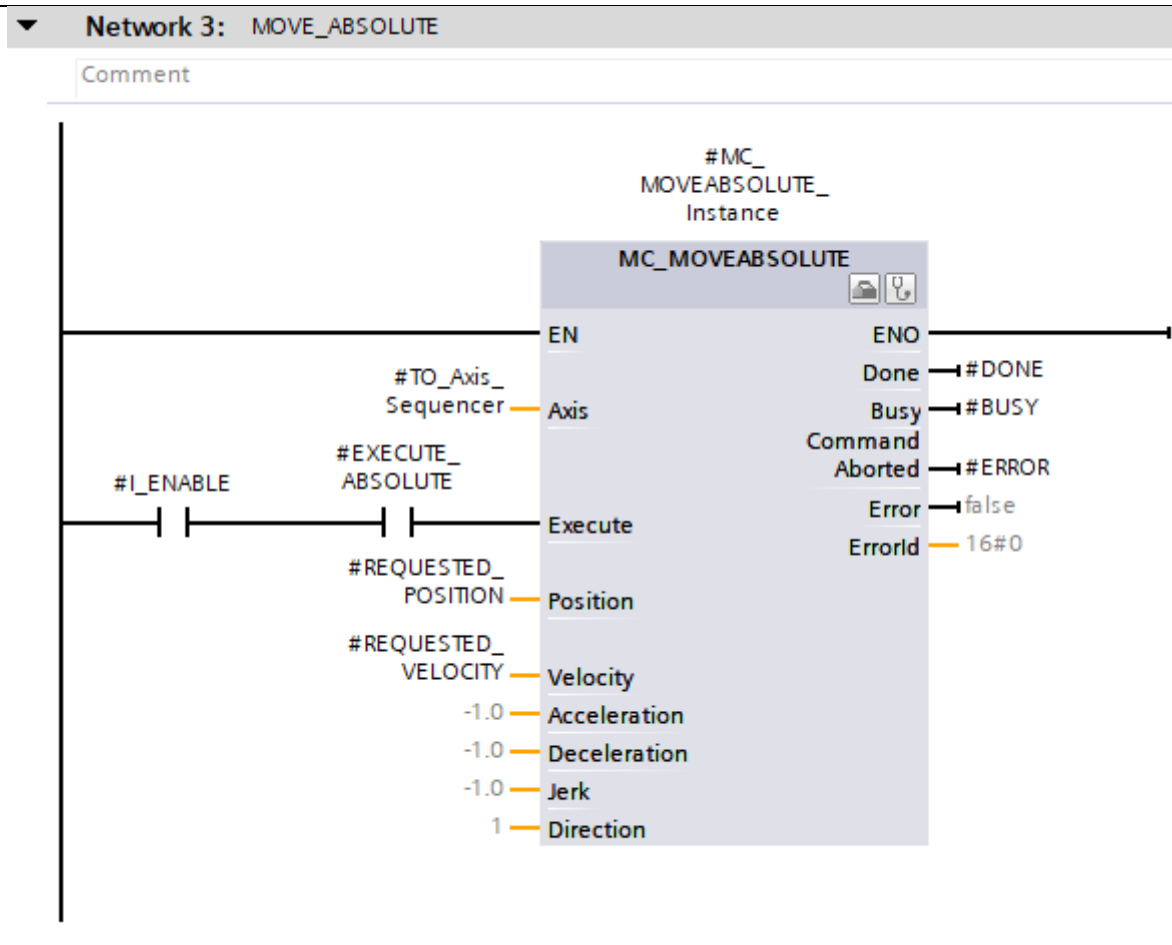


Slika 66. SEQUENCER_FB: Network2 – bliži prikaz (1)

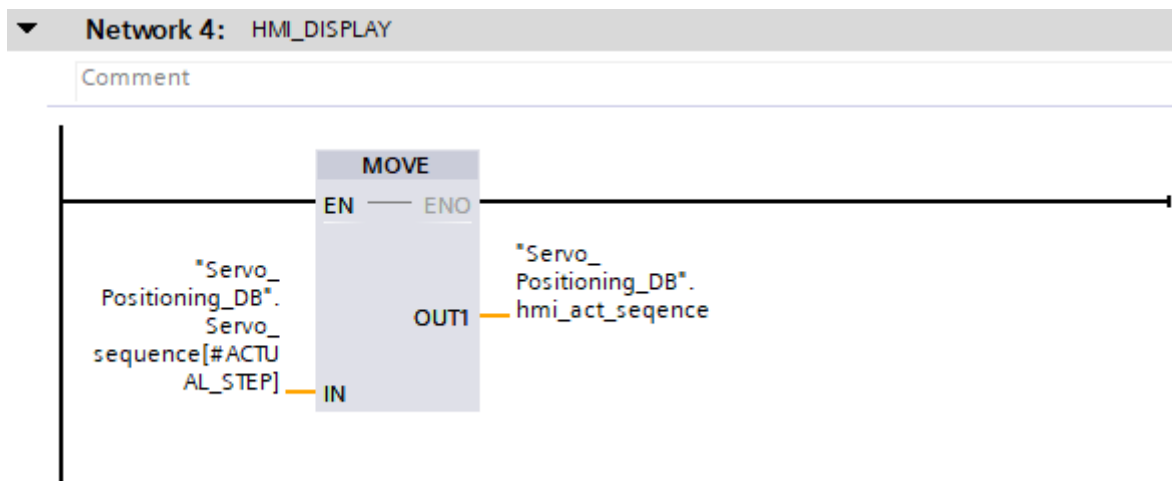
Na [Slika 66] može se vidjeti kako će sekvenca raditi u slučaju da je #ACTUAL_STEP unutar zadanog početnog i krajnjeg koraka, zatim da nema grešaka te da je #START u statusu ON. Vrijednosti za svaki korak dolaze iz podatkovnog bloka koji je prikazan na []. Jedna od tih vrijednosti je i vrijeme koje u bloku TON treba proći prije nego što se #EXECUTE_ABSOLUTE setira u status ON te zatim okida MC_MOVEABSOLUTE u slučaju da je #I_ENABLE također ON, kao što je vidljivo na [Slika 68]. Vidljivo je kako taj blok na svom izlazu ima #DONE kojim se koristi kako bi se resetirale bitne upravljačke varijable i povećao korak za 1. MC_MOVEABSOLUTE se koristi za odlazak u poziciju koja je definirana s #REQUESTED_POSITION.



Slika 67. SEQUENCER_FB: Network2 – bliži prikaz (2)



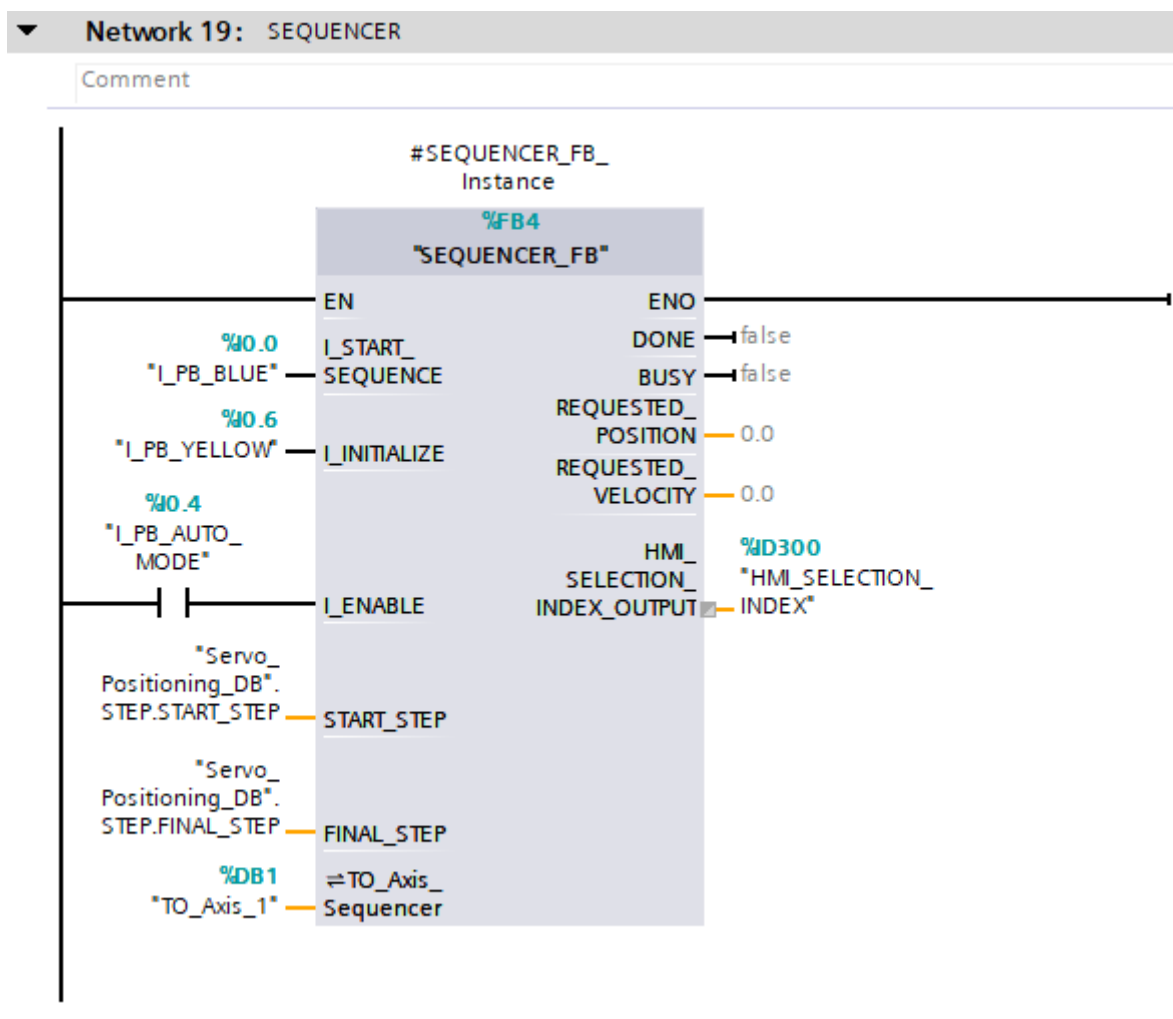
Slika 68. SEQUENCER_FB: Network3



Slika 69. SEQUENCER_FB: Network4

Blok MOVE se ovdje koristi kako bi vrijednosti mogli prikazivati na HMI uređaju na način da iz jednog podatkovnog bloka, za taj određeni korak, vrijednosti pomičemo u podatkovni blok koji tada sadrži vrijednosti trenutnog koraka izvođenja procesa. Podatkovni blokovi vidljivi su

na [Slika 73], a UDT (eng. User Defined Types) koji su korišteni sukladno ranijim objašnjenjima i prednostima korištenja istih u poglavlju [3.2] vidljivi su na [Slika 71] i [Slika 72].



Slika 70. SEQUENCER_FB funkcijski blok pozvan u SERVOMOTOR FB

Instanca ovog funkcijskog bloka je vidljiva na [Slika 70] gdje je vidljivo kako je sekvencu moguće izvoditi isključivo u slučaju ako je prekidač sa [Slika 31] u automatskom načinu rada. Na instanci su dodani ulazi za inicijalizaciju i start sekvence te ulazi koji definiraju željenu poziciju i brzinu za svaki korak izvođenja.

UDT_Servo_Positioning sadrži sve podatke za ispravnu sekvencu. S tim tipom podatka se kreirao „array“ od 0 do 9, što predstavlja 10 koraka izvođenja, a za svaki korak su definirane vrijednosti pozicije, brzine, funkcija koje se trebaju obaviti u tom koraku (npr. paljenje lampice i zatvaranje ventila) te vrijeme. Na [Slika 73] može se vidjeti i „hmi_act_sequence“ u koji se spremaju podaci za trenutni korak izvođenja te se prikazuju na HMI uređaju. Pod „STEP“ se nalaze podaci o početnom i krajnjem željenom koraku izvođenja koji je potrebno upisati na HMI uređaju.

| UDT_Servo_Positioning | | | |
|-----------------------|------------|-----------|---------------|
| | Name | Data type | Default value |
| 1 | Position | LReal | 0.0 |
| 2 | Velocity | LReal | 0.0 |
| 3 | Function_1 | Bool | false |
| 4 | Function_2 | Bool | false |
| 5 | Time | Time | T#0ms |

Slika 71. UDT_Servo_Positioning

| UDT_STEP | | | |
|----------|------------|-----------|---------------|
| | Name | Data type | Default value |
| 1 | START_STEP | DInt | 0 |
| 2 | FINAL_STEP | DInt | 0 |

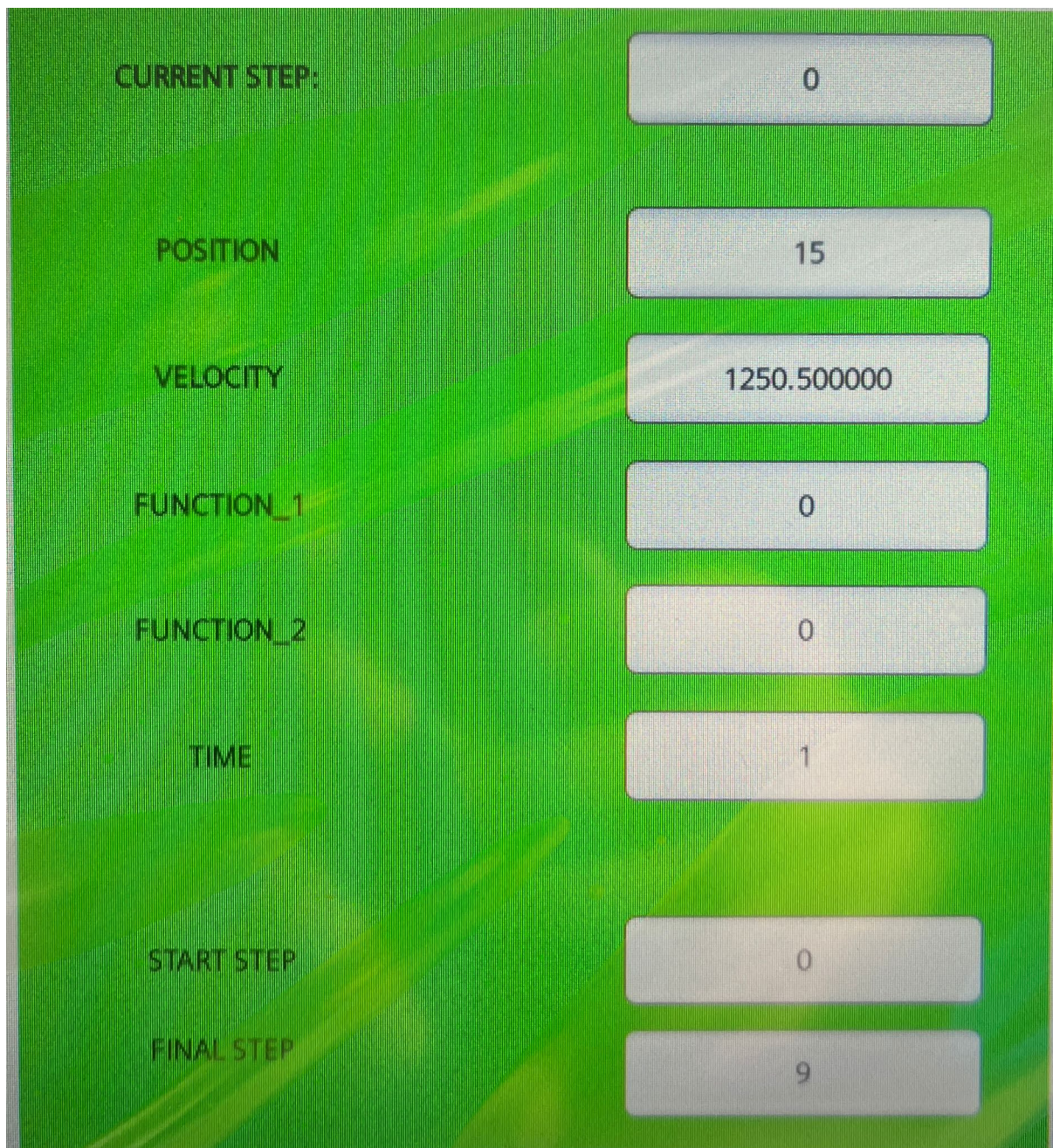
Slika 72. UDT_STEP

| | Name | Data type | Start value | Retain | Accessible f... | Writa... | Visible in ... |
|----|-------------------|--|-------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1 | Static | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | Servo_sequence | Array[0..9] of *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3 | Servo_sequence[0] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 4 | Position | LReal | 15.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 5 | Velocity | LReal | 1250.5 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 6 | Function_1 | Bool | 0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 7 | Function_2 | Bool | 0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 8 | Time | Time | T#15 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 9 | Servo_sequence[1] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 10 | Servo_sequence[2] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 11 | Servo_sequence[3] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 12 | Servo_sequence[4] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 13 | Servo_sequence[5] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 14 | Servo_sequence[6] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 15 | Servo_sequence[7] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 16 | Servo_sequence[8] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 17 | Servo_sequence[9] | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 18 | hmi_act_sequence | *UDT_Servo_Positioning* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 19 | Position | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 20 | Velocity | LReal | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 21 | Function_1 | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 22 | Function_2 | Bool | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 23 | Time | Time | T#0ms | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 24 | STEP | *UDT_STEP* | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 25 | START_STEP | DInt | 0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 26 | FINAL_STEP | DInt | 0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Slika 73. Servo_Positioning_DB podatkovni blok

Sljedeće dolazi kreiranje „Faceplate“ koji se redovito koristi u praksi unutar TIA Portal-a. Svrha korištenja „Faceplate“ je olakšavanje kreiranja zaslona na HMI uređaju za slučaj kontroliranja nekoliko strojeva koje moraju obavljati istu sekvencu. Dakle, ideja je da se napravi jedan „Faceplate“ koji se može vidjeti na [Slika 74] i zatim dodavanje onlikog broja koliko strojeva imamo za obavljanje iste sekvence. Za svaki stroj je potrebno u postavkama unutar programa

postaviti iz kojeg podatkovnog bloka zaprima informacije koje će se prikazivati na zaslonu HMI uređaja.

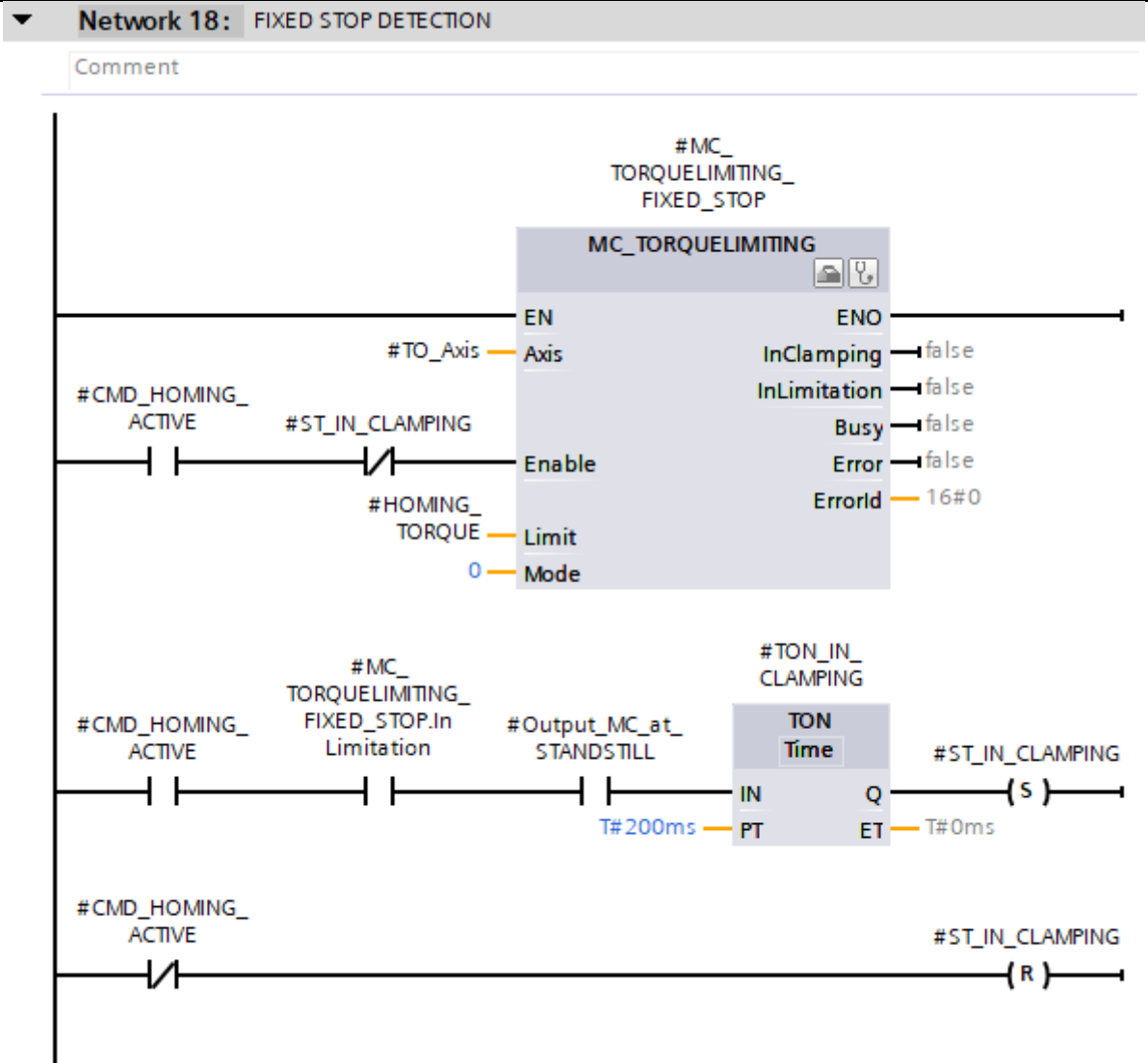


Slika 74. „Faceplate“ na HMI uređaju

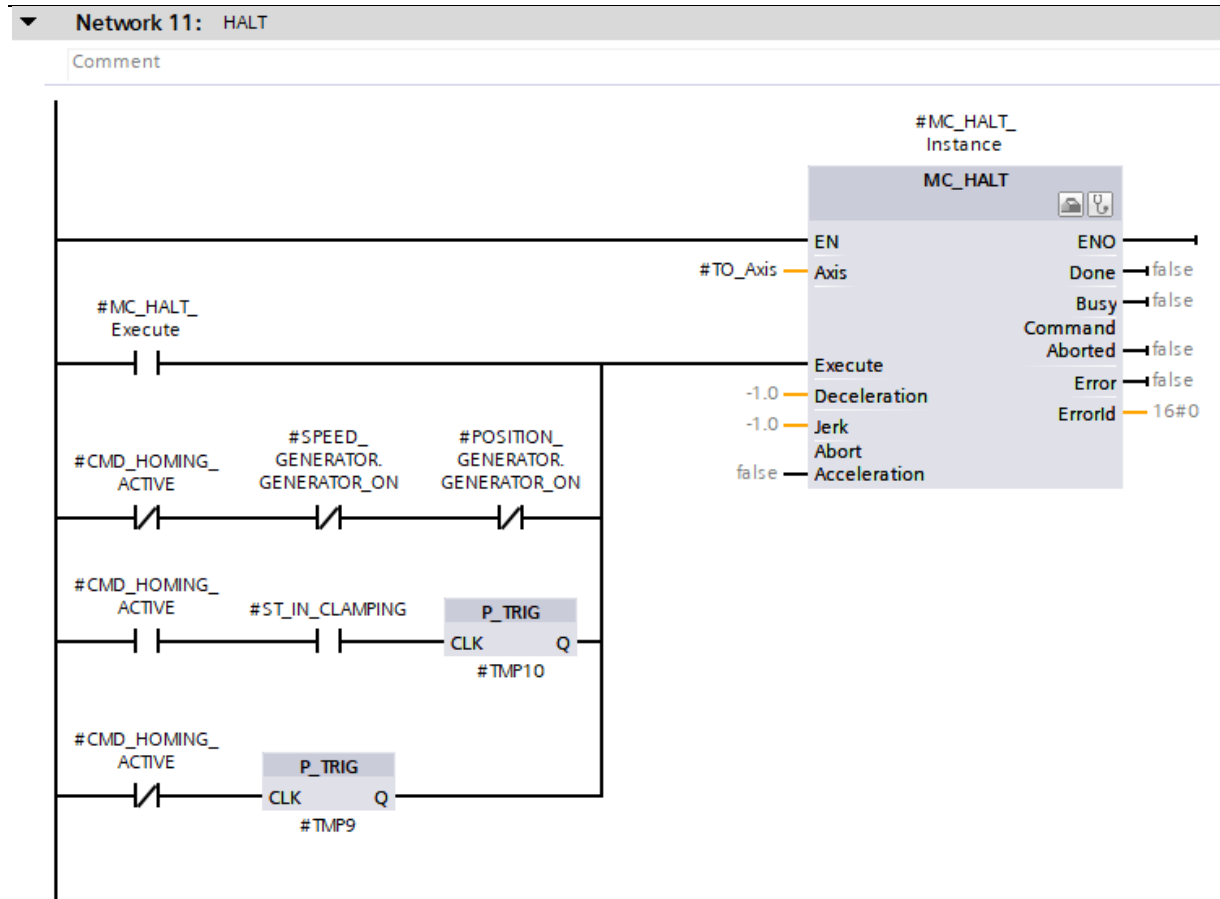
3.4.7. Povratak na početnu poziciju uz limit momenta servo motora

Zadnja opcija koja će se detaljnije opisati i da ju funkcijski blok SERVOMOTOR FB može obavljati je povratak na početnu poziciju uz limitiranje momenta servo motora. Za navedeno su potrebni blokovi MC_MOVEVELOCITY vidljiv na [Slika 38], MC_TORQUELIMITING, MC_HALT te MC_HOME. U situacijama kada s nekom kretnjom dođemo do pozicije u kojoj se više ne može dalje (recimo kraj neke vodilice), moment servo motora kreće rasti jer pokušava ostvariti kretnju koju ne može od mehaničke prepreke. Nakon prelaska postavljenog limita momenta, motor odlazi u željenu poziciju, a ovdje u radu u početnu.

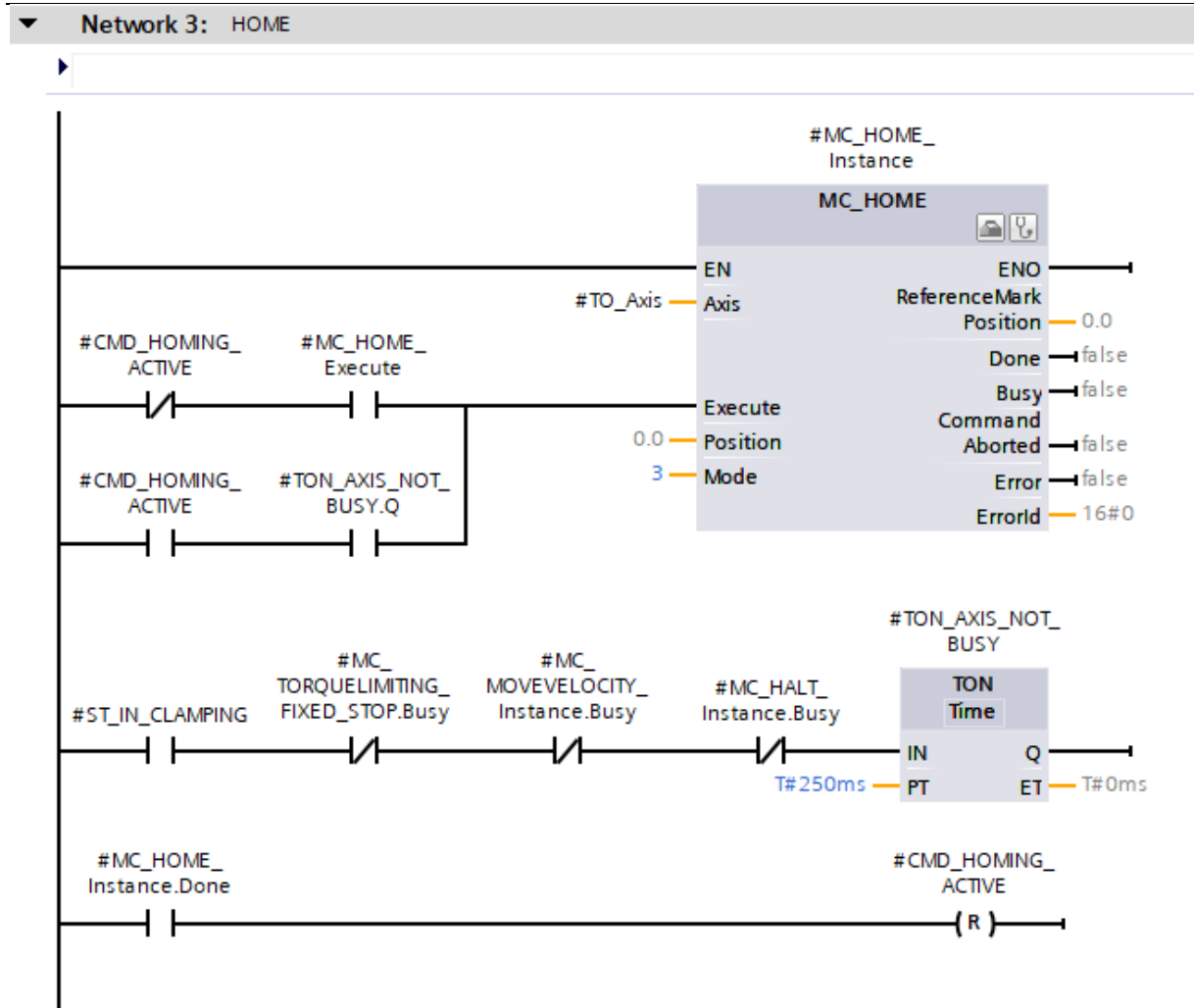
Za početak, prilikom postavljanja #CMD_HOMING_ACTIVE u TRUE servo motor kreće s radom brzinom koja je definirana u #FINAL_SPEED. Servo motor se okreće sve dok mu se moment ne poveća na iznos definiran u #HOMING_TORQUE. Kada se to dogodi, s #MC_TORQUELIMITING_FIXED_STOP.InLimitation dobiva se informacija da je servo motor ušao u limit momenta. Varijabla #Output_MC_at_STANDSTILL iz registara unutar regulatora postavlja se u TRUE u slučaju da servo-os miruje. Kada se ovi uvjeti zadovolje, nakon 200 ms će se setirati #ST_IN_CLAMPING te se pomoću bloka MC_HALT servo-os zaustavlja. S obzirom da je #ST_IN_CLAMPING i dalje u TRUE te blokovi MC_TORQUELIMITING, MC_MOVEVELOCITY i MC_HALT nisu u statusu BUSY (izvršeni su, ne odvijaju svoje operacije više), nakon 250 ms okida se #TON_AXIS_NOT_BUSY.Q koji dolazi od bloka TON. To omogućava da MC_HOME odradi povratak na početnu poziciju te nakon završetka s #MC_HOME_Instance.Done resetira se #CMD_HOMING_ACTIVE.



Slika 75. MC_TORQUELIMITING



Slika 76. MC_HALT



Slika 77. MC_HOME

4. ZAKLJUČAK

U ovom radu prikazano je upravljanje servo motorom tvrtke Siemens AG putem softvera TIA Portal V18. U prvom dijelu rada opisana je korištena oprema u stvarnom okruženju koja je omogućena od tvrtke HUNOR d.o.o u svrhu izrade završnog rada i način rada podređenih regulacijskih krugova struje, brzine i pozicije. Drugi dio rada koncentriran je na softverskom rješenju za upravljanje sa servo osi kroz korištenje funkcijskih blokova prema PLCOpen standardu, a sve unutar funkcijskog bloka koji je kreiran za univerzalno upravljanje s nekoliko servo osi po potrebi. Za početak je pružen uvid u način na koji će se kreirati funkcijski blok, kroz korištenje „User Defined Types“ (UDT) koji se u stvarnoj primjeni gotovo neizbježno koriste. Zatim je prikazano podešavanje tehnološkog objekta upravljanja putem kojeg upravljamo radom servo motora. Nadalje, detaljno su opisani dijelovi funkcijskog bloka s kojim se parametrira servo os te je prikazan i HMI uređaj s kojeg je korisnički moguće upravljati. Također, kroz nekoliko dijagrama prikazan je utjecaj parametara regulatora na konačni izgled odziva na skokovitu pobudu kako bi se dobio osjećaj i razumijevanje prilikom podešavanja parametara regulatora za neku stvarnu primjenu.

Za kraj, treba istaknuti kako bez kvalitetne programske podrške prilikom izrade projekta za upravljanje servo motorom, konačni proizvod nije potpun i dugoročno održiv. U bilo kojem trenutku se može dogoditi da su potrebne nadogradnje i preinake u projektu, a tada do izražaja dolazi način na koji se izradio funkcijski blok. Kroz ovaj završni rad pruženi su koncepti ispravnog kreiranja funkcijskog bloka te je isti ispitan na realnoj industrijskoj opremi.

LITERATURA

- [1] V. Ambrožič i P. Zajec, Električni servo pogoni, Zagreb: GRAPHIS, 2019.
- [2] K. Kamel i E. Kamel, Programmable Logic Controllers, McGraw-Hill Education, 2014.
- [3] SIEMENS AG, »S7-1500 System Manual,« SIEMENS AG, 2013.
- [4] SIEMENS AG, »HMI Unified devices System Manual« SIEMENS AG, 2020.
- [5] SIEMENS AG, »SIMATIC HMI MTP1500 Datasheet« SIEMENS AG, 2023.
- [6] ADD FURNACE CO., »SIMOTICS S-1FL6 servomotors« 2018.
- [7] SIEMENS AG, »SINAMICS V90, SIMOTICS S-1FL6« SIEMENS AG, 2017.
- [8] SIEMENS AG, »Information System - Help,« SIEMENS AG, 2022.
- [9] PLCopen: <https://plcopen.org/technical-activities/motion-control> (pristup: 19.07.2023.)