Generiranje referentnog signala S oblika

Blažević, Emil

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:578975

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-26**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij

GENERIRANJE REFERENTNOG SIGNALA S OBLIKA

Završni rad

Emil Blažević

Osijek, 2019.

Sadržaj

1. UVOD 1
2. UPRAVLJANJE BRZINOM 2
2.1. Linearna rampa 2
2.2. S rampa 3
3. PROGRAMABILNI LOGIČKI KONTROLER 5
3.1. Simatic Manager 6
3.1.1. Function block diagram (FBD) 7
3.1.2. Ladder logic (LAD) 7
3.1.3. Statement list (STL) 8
3.1.4. Structured control language (SCL) 8
3.1.5. Tipovi podataka 9
4. PROGRAMSKO RJEŠENJE ZA GENERIRANJE REFERENTNOG SIGNALA S OBLIKA 10
4.1. Generiranje referentnog signala S oblika 11
4.2. Kreiranje hardverske konfiguracije 13
4.3. Testiranje programa 14
5. CODESYS 17
5.1. Programski jezici 17
5.2. Rješenje problema generiranja referentnog signala S oblika 18
5.3. CODESYS VIZUALIZACIJA 19
6. ZAKLJUČAK 21
LITERATURA 22
SAŽETAK 23
ABSTRACT 24
ŽIVOTOPIS 25

1.UVOD

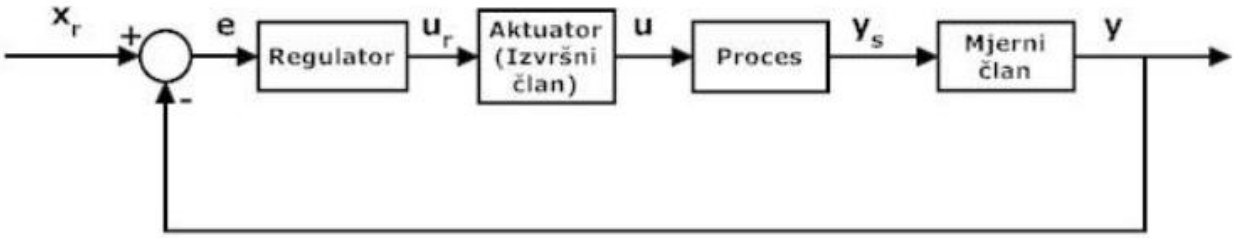
Računala su, razvojem, pronašla svoju primjenu u raznim granama industrije. Računala omogućavaju bolje i preciznije vođenje procesa i s time je ljudski faktor pogreške sveden na minimum. Sustavi upravljanja gibanjem imaju široki raspon primjene u industriji, kao što je u proizvodnji, transportu, u robotici itd. U sustavima upravljanja gibanjem, računala se primjenjuju za kontrolu brzine i položaja radi generiranja željenih pokreta koji mogu biti linearni ili kružni. Izazov u upravljanju gibanjem je kako postići precizno kretanje s minimalnim prebacivanjem brzine i položaja te minimalnim vibracijama. U ovom radu će biti prikazano programsko rješnje za generiranje S rampe.

Tema ovog rada je opisana u šest poglavlja. U prvom je dijelu dan uvod i opis zadatka. U drugom poglavlju je opisano upravljanje brzinom. U trećem poglavlju je dan kratki pregled programabilnog logičkog kontrolera (engl. Programmable Logic Controller - PLC) kao Simatic Manager. Programsko je rješenje prikazano i objašnjeno u četvrtom poglavlju, dok je vizualizacija u CoDeSys programskom okruženju opisana u petom poglavlju. Šesto poglavlje sadrži zaključak rada.

Zadatak završnog rada je izrada programa za generiranje referentnog signala S oblika koji uključuje mogućnost definiranja brze i spore izmjene brzine, maksimalnog i minimalnog iznosa brzine te trenutne točke rampe (ubrzanje, usporenje). Rad uključuje razvoj softvera i simulaciju u alatu Step7 i CoDeSys te vizualizaciju u alatu CoDeSys.

2. UPRAVLJANJE BRZINOM

Upravljanje brzine ubrzavanja i usporavanja motora je bitan faktor kada je potrebna velika brzina i velika preciznost. Promjena brzine se zove rampa. Složenost generiranja rampe ovisi o primjeni. Upotreba rampi u industriji je vrlo često, kako bi se dobili precizni pomaci u procesu. Pomaci mogu biti linearni ili rotacijski. Za kontrolu brzine se često upotrebljava zatvorena petlja te se u svakom trenutku zna trenutna brzina koja se uspoređuje s referentnom.



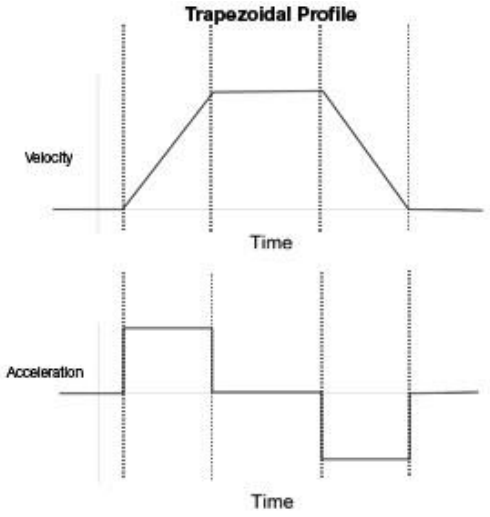
Slika 2.1. Regulacijski krug s povratnom vezom [1]

Napravit će se usporedba sa slikom 2.1. Aktuator bi bio neka vrsta motora. Proces kojim se upravlja je promjena brzine nekog sustava. Kada motor dobije signal iz regulatora, on se pokrene te s time mijenja brzinu sustava. Mjerni član mjeri trenutnu brzinu koja se uspoređuje s referentnom. Sve dok postoji odstupanje trenutne brzine od referentne, regulator će djelovati i procesirati signal prema svom algoritmu. Posljedica toga će biti ubrzavanje, usporavanje ili održavanje brzine. Na taj se način regulira brzina pomoću povratne veze.

2.1. Linearna rampa

Najčešće se kod upravljanje brzinom motora upotrebljava referentna veličina s linearnom promjenom brzine. Pri tome je akceleracija konstantna za vrijeme ubrzavanja i usporavanja te se mogu postići brzi pokreti. To znači da se na motor dovodi nazivni napon, i on će odmah pokušati dostići nazivnu brzinu. Sami motori posjeduju određenu tromost te se promjenom akceleracije na njih stvara opterećenje koje može utjecati na stabilnost sustava. Linearne rampe su sklone premašivanju pozicije, te vibracije koje se pojavljuju tijekom zaleta i zaustavljanja motora utječu na vrijeme koje je potrebno da stroj dođe na određenu poziciju sa željenom preciznošću.

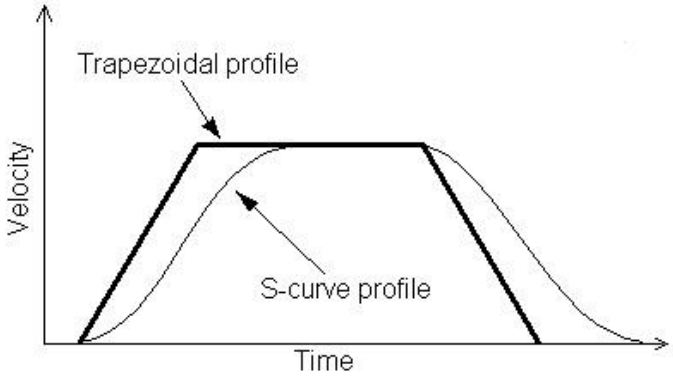
Na slici 2.2 je prikazan izgled linearne rampe koja ima 3 faze kretanja kao i promjena brzine i akceleracije u vremenu. Prva faza je vrijeme ubrzanja, druga je vrijeme konstantne brzine i treća faza je vrijeme usporavanja. Može se zaključiti da su najveća opterećenja na motoru prilikom pokretanja i zaustavljanja.



Slika 2.2. Linearna rampa [3]

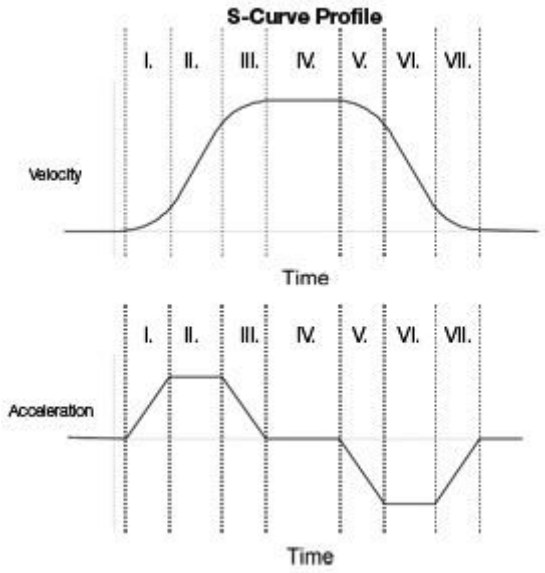
2.2. S rampa

S-rampa je dobila ime po svojem obliku koji podsjeća na slovo S. Oblik krivulje se dobije integriranjem linearne rampe koja ovdje predstavlja akceleraciju. Prednost S-rampe je ta što se vrijednost akceleracije može kontrolirati. Vrijednost napona na motoru se mijenja po S-rampi te se time dobiju manja opterećenja i minimalne vibracije za razliku od linearne rampe. Vrijeme koje je potrebno da se dostigne zadana brzina je veće nego kod linearne rampe, ali se zato dobiju nježniji pokreti prilikom ubrzavanja ili usporavanja. Na slici 2.3 može se vidjeti usporedba linearne rampe (Trapezoidal profile) i S rampe (S-curve profile).



Slika 2.3 Usporedba linearne i S rampe [4]

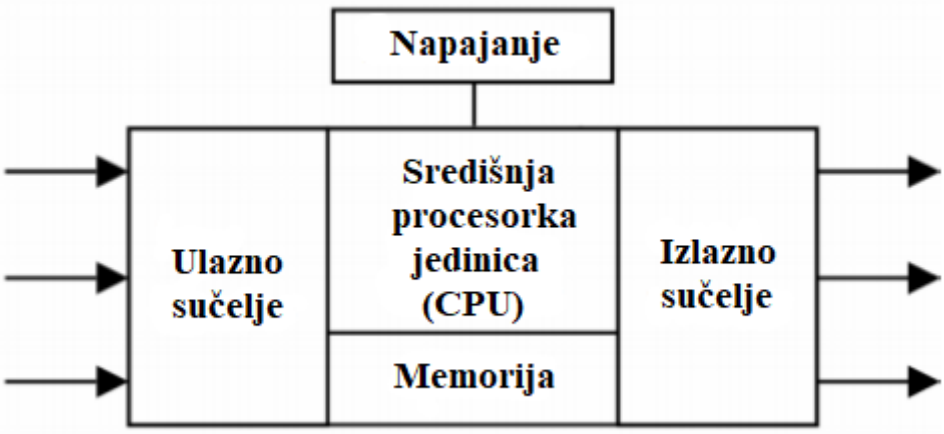
Na slici 2.4 je prikazan izgled S rampe (S-Curve profile) koja ima 7 fazi kretanja i promjena brzine (Velocity) i akceleracije (Acceleration) u vremenu (Time). Prva faza rampe je povećanje ubrzanja od 0 do maksimalne vrijednosti. U drugoj fazi, rampa ubrzava najvećim ubrzanjem sve dok ne mora početi usporavati. U trećoj fazi rampa usporava od maksimalne vrijednosti ubrzanja sve do 0. U četvrtoj fazi je brzina konstantna a akceleracija je 0. Preostale faze rampe su simetrične prvim tri fazama.



Slika 2.4 S rampa [3]

3. PROGRAMABILNI LOGIČKI KONTROLER

Kako bi se moglo upravljati nekim procesom, potreban je regulator koji pomoću neke zadane logike upravlja aktuatorima (električni motori, ventili, pumpe i dr.). Time se uzrokuje promjena u procesu i omogućava se regulacija sustava. Prije pojave programabilnih logičkih kontrolera u industriji, procesima se upravljalo preko sklopnika i releja. Te komponente su bile spojene žicama. Problem kod ovog načina upravljanja u složenim sustavima upravljanja je, ako se dogodi neka greška ili se dodaju nove komponente u sustav ili se želi promijenit logika upravljanja, to se moralo učiniti prespajanjem žica. Kod velikih i složenih sustava ovaj način upravljanja je bio loš a sustav neiskoristiv. Razvojem mikroprocesora i sve većim zahtjevima industrije uvedeno je procesno računalo u sustave upravljanja. Procesno računalo za upravljanje procesima naziva se programabilni logički kontroler ili skraćeno PLC. PLC-ovi su elektronički uređaji s funkcijama upravljanja spremljenim u memoriju upravljačke jedinice. Prednosti PLC-a su smanjenje dimenzija i potrošnja električne energije, pouzdanost i programirljivost, što im omogućuje široku primjenu u različitim postrojenjima u industriji. Prema slici 3.1 može se vidjeti da se PLC sastoji od nekoliko funkcijskih cjelina.



Slika 3.1. Funkcijske cjeline PLC-a [2]

PLC se sastoji od središnje procesorske jedinice (CPU) koja izvodi korisnički program. Memorija PLC-a služi za pohranu operacijskog sustava i korisničkog programa. Ulazni i izlazni moduli mogu biti digitalni i analogni. Analogni moduli mogu biti strujni ili naponski. Strujni analogni moduli upotrebljavaju se ako se signal mora slati na veće udaljenosti i raspon signala je od 4-20 mA. Naponski analogni moduli imaju raspon signala od 0-5 V, 0-10 V ili +/-10 V. Digitalni moduli

najčešće imaju 24 V DC. Kako bi PLC radio, potrebno mu je napajanje. PLC-u se mogu po potrebi dodavati razni moduli. Osnovni način rada PLC-a je cikličko izvođenje korisničkog programa (Slika 3.2).

Slika 3.2. Osnovni način rada PLC-a [2]

Za potrebe ovog rada upotrijebio se Simatic S7-400 PLC koji se sastoji od sljedećih modula:

- postolja
- napajanja
- procesora
- komunikacijskog procesora (služi za komunikaciju računala i PLC-a povezanih ethernet kablom)

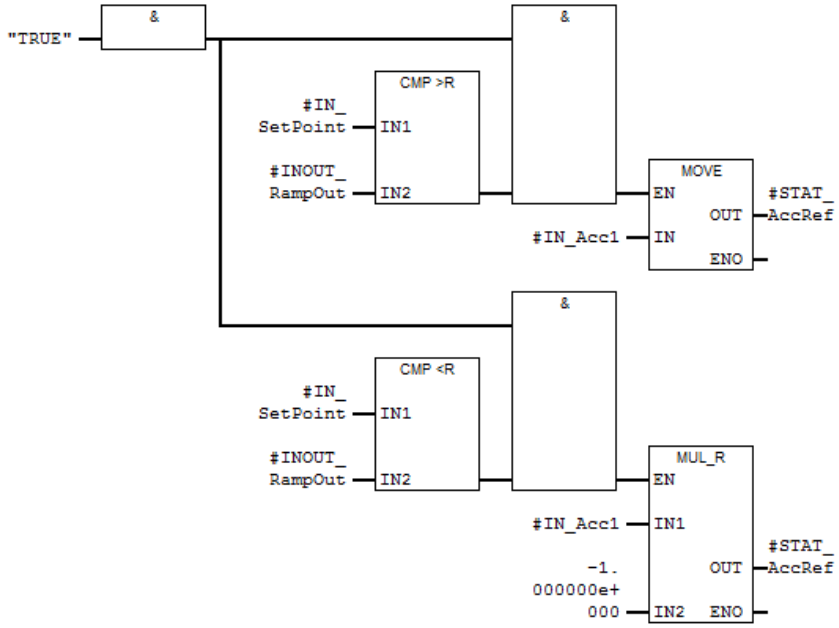
3.1. Simatic Manager

Simatic Manager je program kojeg je razvila tvrtka Siemens. Pomoću njega se može programirati i konfigurirati PLC. Program je zasnovan na IEC 61131-3 standardu. Programski jezici koji su podržani su :

- FBD (engl. Function Block Diagram)
- LAD (engl. Ladder Logic)
- STL (engl. Statement List)
- SCL (engl. Structured Control Language)

3.1.1. Function block diagram (FBD)

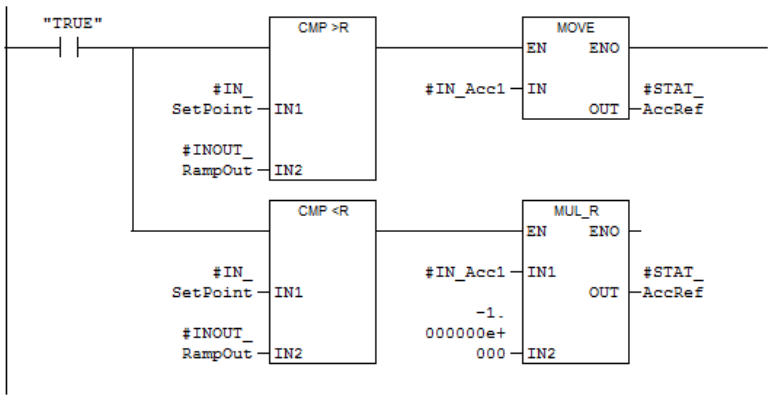
FBD pripada grafičkim programskim jezicima za programiranje PLC-a. Funkcije su opisane kao blokovi koji se sastoje od određenog broja ulaza i izlaza. Ulazi i izlazi su spojeni linijama. Pogodan je za praćenje toka signala.



Slika 3.3 FBD programski jezik

3.1.2. Ladder logic (LAD)

Najupotrebljivaniji jezik za programiranje PLC-ova je LAD. Program se izvršava od lijevo prema desno i odozgo prema dolje i tako ciklički. Programski jezik se zasniva na dijagramu strujnog kruga relejnih logičkih sklopova.



Slika 3.4 LAD programski jezik

3.1.3. Statement list (STL)

U STL programskom jeziku upotrijebljuju se naredbe tekstualnog oblika i najbliži je programerima. Svaka linija koda odgovara jednoj naredbi koje PLC izvršava. STL jezik je program niske razine, što znači da je razina apstrakcije mala između samog jezika i strojnog jezika.

```
      A     "TRUE"
      =     L     20.0
      A     L     20.0
      A(
      L     #IN_SetPoint
      L     #INOUT_RampOut
      >R
      )
      JNB   _001
      L     #IN_Acc1
      T     #STAT_AccRef
_001: NOP   0
      A     L     20.0
      A(
      L     #IN_SetPoint
      L     #INOUT_RampOut
      <R
      )
      JNB   _002
      L     #IN_Acc1
      L     -1.000000e+000
      *R
      T     #STAT_AccRef
_002: NOP   0
```

Slika 3.5. STL programski jezik

3.1.4. Structured control language (SCL)

Za razliku od STL-a, SCL je viši programski jezik. SCL sličan je pascal-u na kojem je i zasnovan. Na slici 3.6 je primjer koda napisan u SCL-u za logički sklop I.

```
FUNCTION AND
VAR_INPUT
    IN_1 : BOOL;
    IN_2 : BOOL;
END_VAR

VAR_OUTPUT
    OUTPUT : BOOL;
END_VAR

BEGIN
IF (IN_1 = TRUE AND IN_2 = TRUE) THEN
    OUTPUT := TRUE;
    END_IF;
END_FUNCTION
```

Slika 3.6. Logički blok I napisan u SCL-u

3.1.5. Tipovi podataka

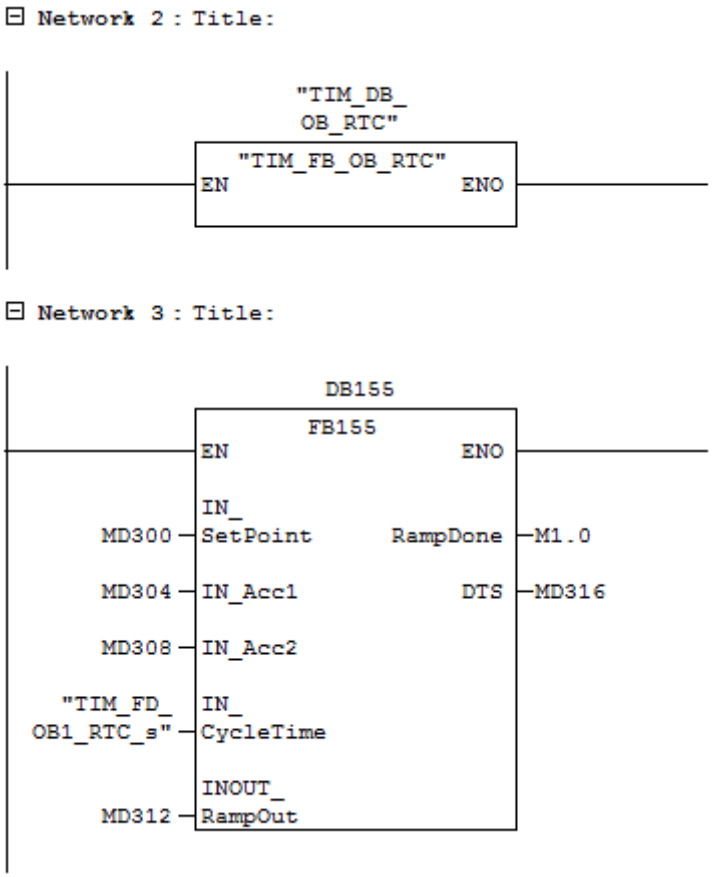
Tijekom programiranja PLC-a moraju se poštivati formati unosa podatka, inače program neće raditi. Svaki tip podatka je definiran određenom dužinom u bitovima koju zauzima u memoriji PLC-a. Osnovni tipovi podataka koji se upotrijebljuju su dani u tablici 3.7.

Vrsta Podatka i opis	Veličina koju podatak zauzima u bitovima	Dozvoljeni format
BOOL (bit)	1	Boolean
BYTE (byte)	8	Heksadecimalni broj
WORD (riječ)	16	BCD,binarni broj,heksadecimalni broj,ne pridruženi dekadski broj
DWORD (dupla riječ)	32	Heksadecimalni broj,binarni broj,ne pridruženi broj
INT (integer)	16	Dekadski broj
DINT (dupli integer)	32	Dekadski broj
REAL (broj s pomičnom točkom)	32	IEEE broj s pomičnom točkom
S5TIME(SIMATIC vrijeme)	16	S7 vrijeme u koracima od 10ms
TIME (IEC vrijeme)	32	IEC vrijeme u koracima od 1ms
DATE (IEC datum)	16	IEC vrijeme u koracima od 1 dan
TIME_OF_DAY (vrijeme)	32	Vrijeme u koracima od 1 ms
CHAR (znak)	8	ASCII znak

Tablica 3.7 Tipovi podataka

4. PROGRAMSKO RJEŠENJE ZA GENERIRANJE REFERENTNOG SIGNALA S OBLIKA

Program za generiranje S signala napisan je u LAD programskom jeziku u Siemensovom razvojnom okruženju Step7. Izgled funkcijskog bloka (FB155) je prikazan na Slici 4.1. i pridružen mu je podatkovni blok DB155 u kojem su svi podaci funkcijskog bloka. Program je napravljen u zasebnom projektu kako bi bio univerzalan te ne ovisi o hardveru PLC-a. Tako se program može upotrebljavati na bilo kojoj hardverskoj konfiguraciji PLC-a.



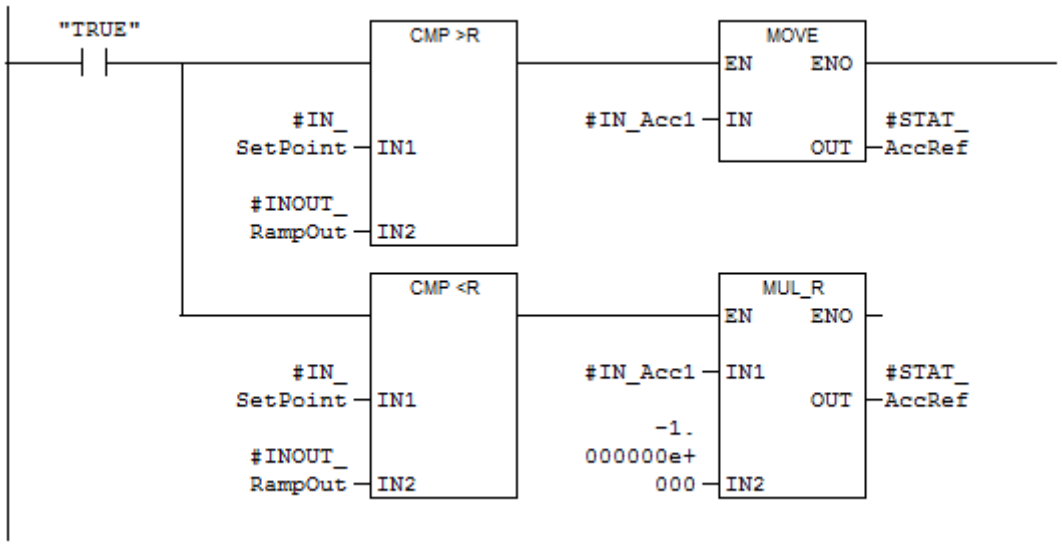
Slika 4.1. Izgled funkcijskog bloka za generiranje S signala

Kako bi se mogla mijenjati brzina rampe, potrebno je znati vrijeme izvođenja jednog ciklusa PLC-a. Upravo za to služi funkcijski blok „TIM_DB_OB_RTC“ (Slika 4.1.). Vrijeme izvođenja ciklusa PLC-a ovisi o veličini napisanog programa i traje onoliko koliko je potrebno za izvođenje svih funkcija u programu. U nekim procesima je bitno da se izvođenje ciklusa odvija u jednakim intervalima te se izvođenje jednog ciklusa može namjestiti na točno određeno vrijeme. U ovom radu, izvođenje jednog ciklusa nije definirano nego ovisi o veličini programa. Varijabla u koju se

sprema vrijeme izvođenja jednog ciklusa je „TIM_FD_OB1_RTC_s“. Vrijednosti koje su potrebne za generiranje rampe su „IN_SetPoint“, „IN_Acc1“, „IN_Acc2“ i ranije spomenuto vrijeme jednog ciklusa „IN_CycleTime“ koje se mijenja u svakom ciklusu PLC-a. „IN_SetPoint“ predstavlja maksimalnu brzinu do koje blok mora doći, „IN_Acc1“ je maksimalna akceleracija kojom smije ubrzavati, a „IN_Acc2“ je korak ubrzanja koji se dodaje u svakom ciklusu PLC-a. U varijable „MD300“, „MD304“ i „MD308“ se zapisuju navedene vrijednosti. „INOUT_RampOut“ predstavlja trenutnu brzinu. Zbog praktičnosti je upotrijebljen „INOUT“ ulaz/izlaz u funkcijski blok. Unutar bloka, „INOUT“ varijabla se može po potrebi mijenjati što nije slučaj s varijablom koja je definirana kao „OUT“. Izlaz iz bloka su „RampDone“ i „DTS“. Vrijednost izlaza „RampDone“ može biti „0“ ili „1“ i služi za signalizaciju je li postignuta zadana brzina ili nije. Izlaz „DTS“ (engl. Distance to stop) daje informaciju ako se u tom trenutku motor krene zaustavljati, kao i koliki mu je potreban put da se zaustavi dok brzina ne dođe na 0.

4.1. Generiranje referentnog signala S oblika

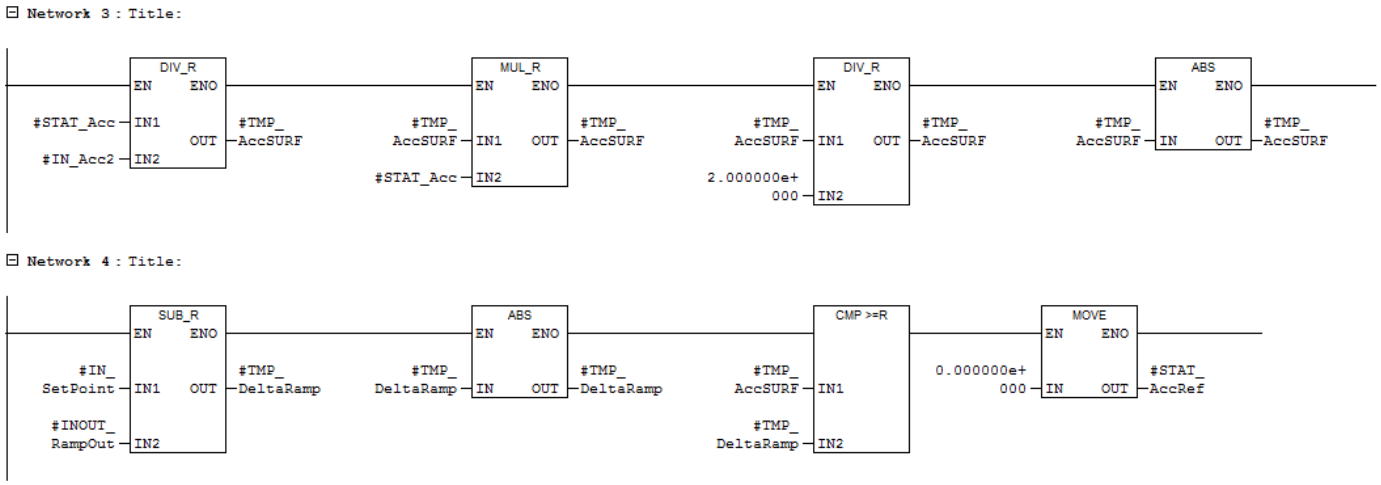
U svakom ciklusu PLC-a, program uspoređuje trenutnu brzinu sa zadanom. Ako je zadana brzina veća od trenutne, u varijablu „STAT_AccRef“ će se zapisati vrijednost varijable „IN_Acc1“. Ako je zadana brzina manja od trenutne, u varijablu „STAT_AccRef“ će se zapisati negativna vrijednost varijable „IN_Acc1“ (Slika 4.2).



Slika 4.2. Usporedba zadane i trenutne brzine

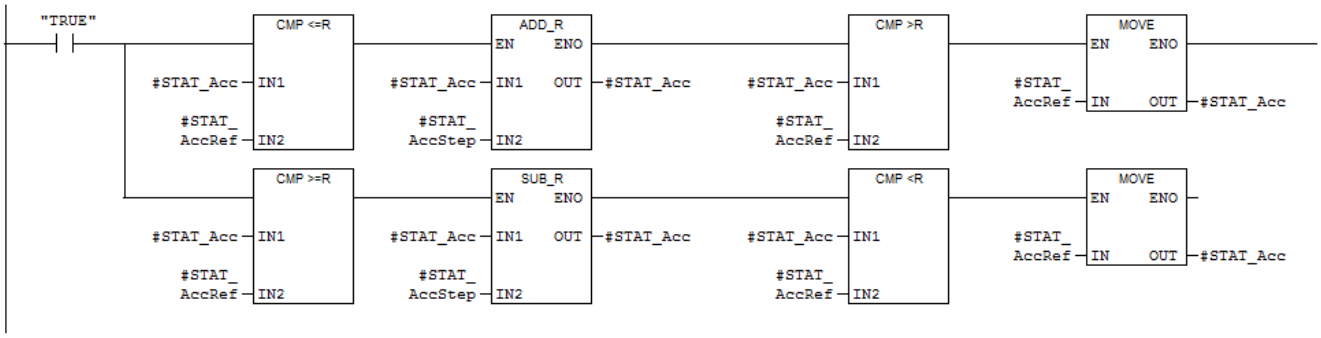
Kako bi program znao kada treba početi usporavati, u svakom ciklusu računa površinu ispod krivulje ubrzanja („TMP_AccSurf“) i razliku trenutne vrijednosti brzine i zadane brzine

(„TMP_DeltaRamp“). Ukoliko je površina veća ili jednaka od razlike brzine, u varijablu „STAT_AccRef“ se zapisuje vrijednost 0 (Slika 4.3).



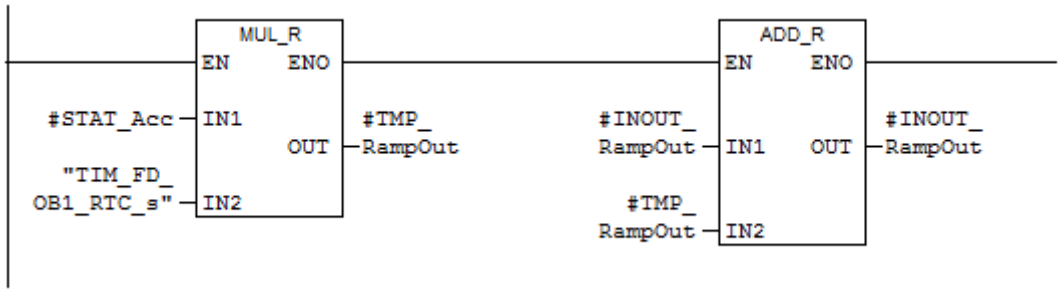
Slika 4.3. Računanje površine, razlike brzine i točke usporavanja

Zatim program uspoređuje dvije vrijednosti, „STAT_Acc“ koja predstavlja trenutnu akceleraciju i „STAT_AccRef“ koja predstavlja maksimalnu akceleraciju. Ovisno o vrijednostima pojedinih varijabla, program zbraja ili oduzima korak ubrzanja i upisuje novu vrijednost u varijablu „STAT_Acc“. Na kraju svake grane je dodan uvjet da vrijednost varijable „STAT_Acc“ ne može biti veća od „STAT_AccRef“ varijable. Time se je osiguralo da program ne može računati vrijednost akceleracije veću od maksimalne koja je zadana (Slika 4.4).



Slika 4.4. Računanje trenutne akceleracije

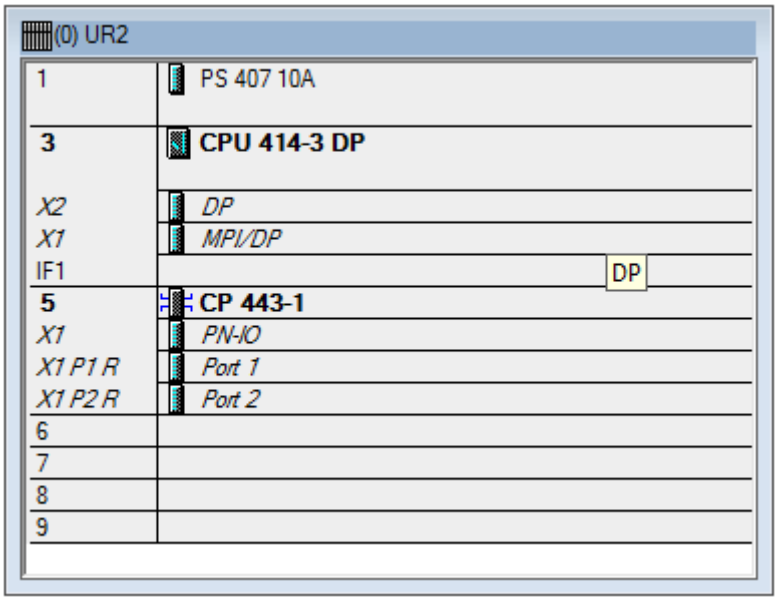
Slika 4.5 prikazuje izračun brzine u jednom ciklusu PLC-a „TMP_RampOut“ koja se zatim dodaje na trenutnu brzinu „INOUT_RampOut“.



Slika 4.5. Integriranje brzine

4.2. Kreiranje hardverske konfiguracije

Kako bi se mogao pokrenuti program na PLC-u, u Simatic Manager-u je potrebno napraviti konfiguraciju hardvera. Unutar prozora HW Config može se kreirati hardver koji mora biti identičan s hardverom kojim se raspolaže. Jedinstveni identifikacijski brojevi u HW Config prozoru se moraju podudarati sa stvarnim hardverom. Pojedini elementi se slažu u predviđeni dio prozora HW Config.



Slika 4.6. Hardverska konfiguracija PLC-a

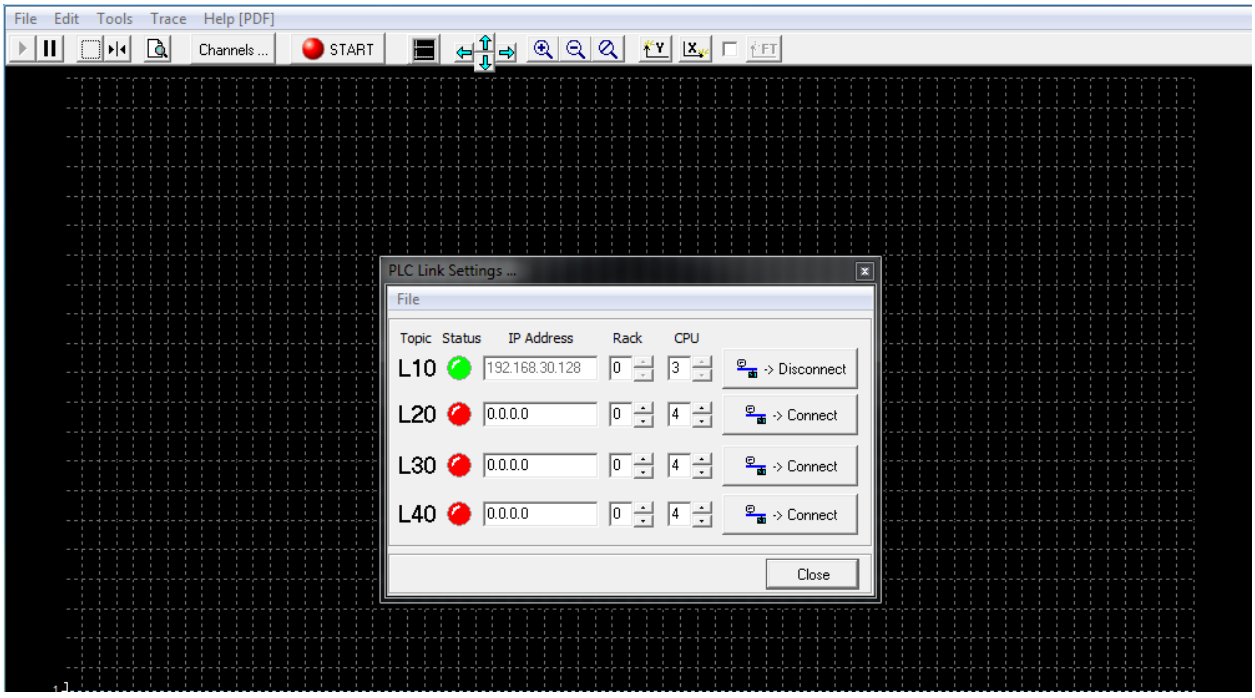
Na slici 4.6 se vide elementi PLC-a na kojem će se testirati program, a to su:

- Postolje UR2 (engl. Rack)
- napajanje 407 10A
- procesor 414-3 DP
- komunikacijski procesor 443-1.

Postolje PLC-a sadrži devet utora od kojih je pet popunjeno. Na prva dva se nalazi napajanje koje služi za pretvorbu izmjeničnog napona 230 V na 24 V i 5 V. Napajanje može imati jednu ili dvije baterije koje se upotrebljavaju kada nestane 230 V i služe kao zaštita za pohranu podataka. Procesor u navedenom PLC-u je iz serije 414-3 koji ima dva komunikacijska sučelja označena s X1 i X2 i zauzima 3. i 4. mjesto na postolju. Procesor je dodatno opremljen s 4 MB SRAM memorije. Na petom utoru se nalazi komunikacijski procesor koji ima 2 ethernet porta za komunikaciju s računalom.

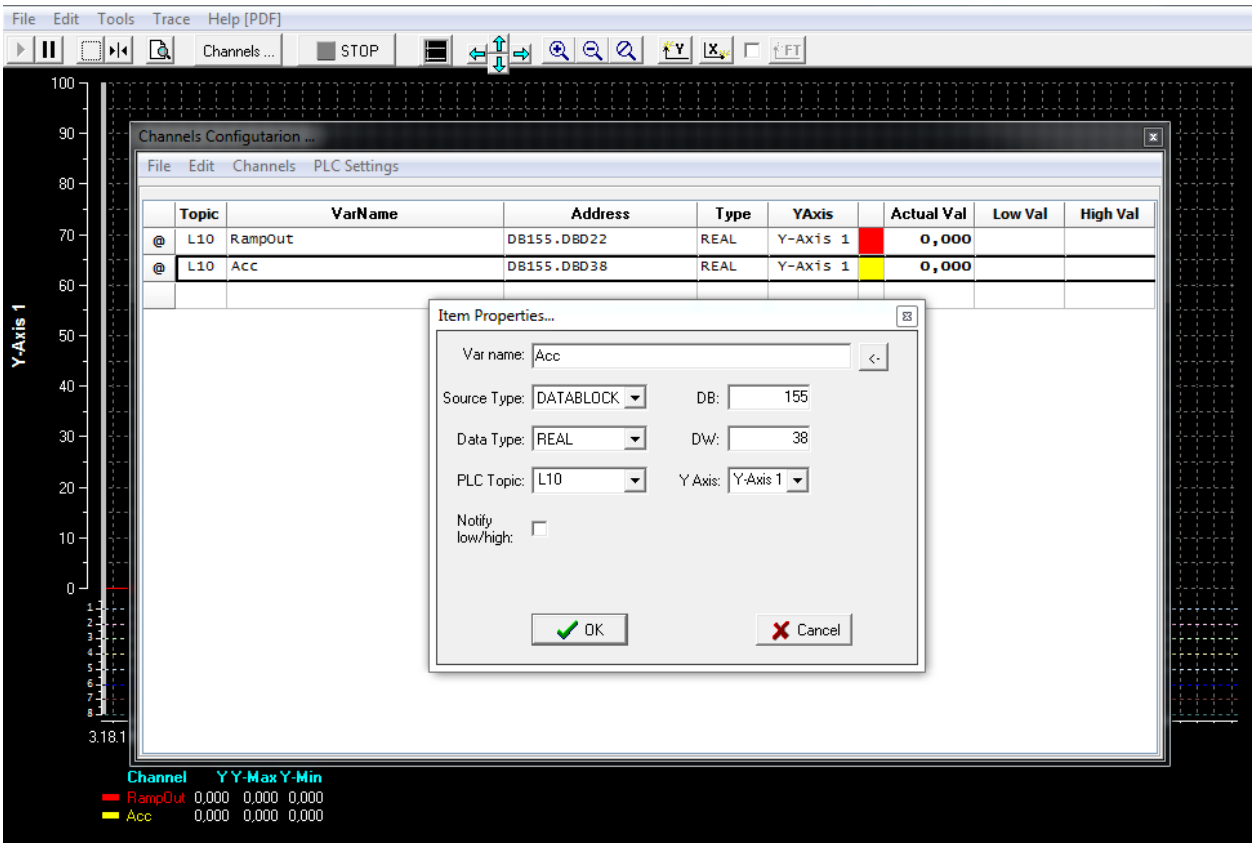
4.3. Testiranje programa

Testiranje programa se može vršiti nakon konfiguracije hardvera. Kako Simatic Manager ne nudi grafičku vizualizaciju, upotrijebio se program S7 Trend Values. Nakon pokretanja programa, pod izbornikom Edit treba odabrati PLC Settings. U novom prozoru treba upisati IP adresu PLC-a i kliknuti na Connect (Slika 4.7).



Slika 4.7. Uspostava komunikacije S7 Trend Values i PLC-a

Zatim je potrebno odabrati varijable koje se žele vizualizirati. U padajućem izborniku Edit zatim se odabire Channel Settings. U novom prozoru se upisuje ime varijable, broj podatkovnog bloka iz kojeg će se čitati podaci i adresu varijable u podatkovnom bloku (Slika 4.8).



Slika 4.8. Postavljane varijable

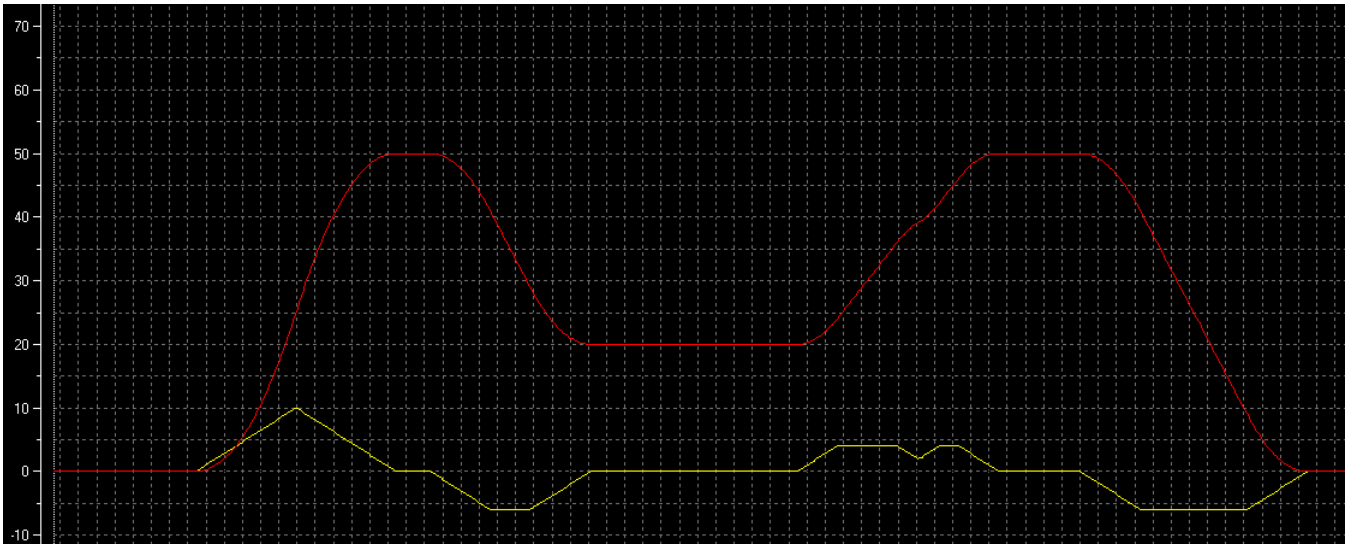
Kako bi se isprobala funkcionalnost funkcijskog bloka za generiranje referentnog signala S oblika, vrijednosti varijabla će se mijenjati preko VAT (engl. Variable Table) tablice. Iz tablice se mogu vidjeti vrijednosti varijabla gdje je „MD 300“ maksimalna dopuštena brzina, „MD 304“ maksimalna dopuštena akceleracija, „MD 308“ korak akceleracije, „MD 312“ trenutna brzina i „MD 32“ vrijeme izvođenja jednog ciklusa PLC-a u sekundama (Slika 4.9).

@Variable table1 ONLINE

		Address	Symbol	Display format	Status value	Modify value
1		MD 300		FLOATING_POINT	50.0	50.0
2		MD 304		FLOATING_POINT	6.0	6.0
3		MD 308		FLOATING_POINT	2.0	2.0
4		MD 312		FLOATING_POINT	35.9051	
5		MD 32	"TIM_FD_OB1_RTC_s"	FLOATING_POINT	0.01	
6						

Slika 4.9. VAT tablica

Kako se mijenjaju vrijednosti varijabli u VAT tablici, tako se može vidjeti promjena brzine i akceleracije na grafu u programu S7 Trend Values. Brzinu predstavlja crvena linija a akceleraciju žuta linija. Iz grafa se može vidjeti da realizirani funkcijski blok uspješno mijenja brzinu prema unesenim parametrima (Slika 4.10).



Slika 4.10. Prikaz brzine (crvena linija) i akceleracije (žuta linija)

5.CODESYS

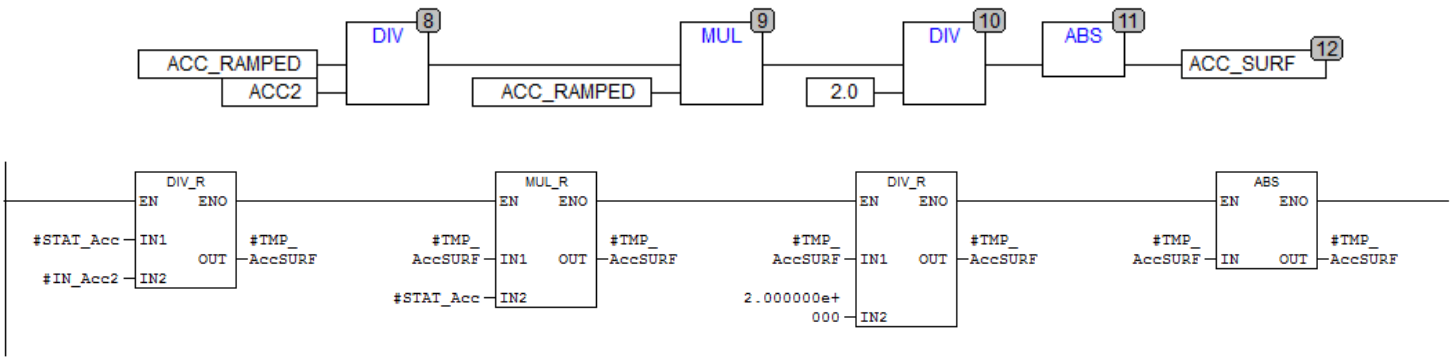
CoDeSys (engl. controller development system) kao i Simatic Manager služi za programiranje PLC-ova prema IEC 61131-3 industrijskom standardu. Program je razvila njemačka softverska tvrtka 3S-Smart Software Solutions. Softverski alat pokriva različite tehnologije u industrijskoj automatizaciji. CoDeSys program u kojem je napravljen ovaj rad je verzije 2.3 i može se besplatno preuzeti s interneta. Za razliku od Simatic Manager-a, u CoDeSys-u se može napraviti vizualizacija programskog rješenja.

5.1. Programski jezici

Kao i Simatic Manager, CoDeSys podržava sve programske jezike koji su definirani IEC 61131-3 industrijskim standardom.

To su: - IL (engl. instruction list)
- ST (engl. structured text)
- LAD (engl. ladder diagram)
- FBD (engl. function block diagram)
- SFC (engl. sequential function chart)

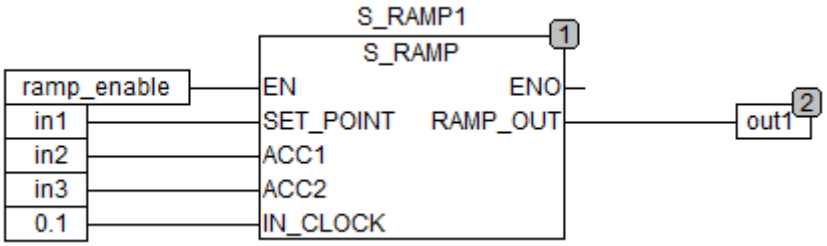
Dodatni programski jezik koji podržava CoDeSys, a nije definiram standardom je CFC (engl. Continuous Function Chart). Njegova prednost je ta što dopušta programiranje bez privremenih varijabli i blokovi se mogu pozivati neovisno o drugima.



Slika 5.1 Razlika CFC (Continuous Function Chart) i LAD (ladder diagram) jezika

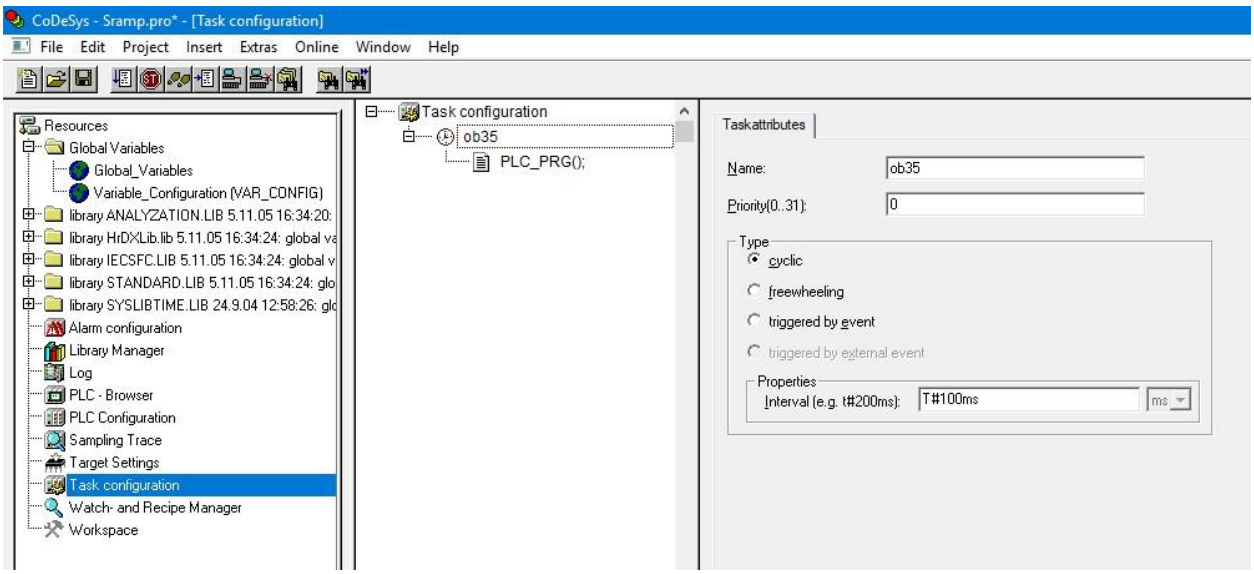
5.2. Rješenje problema generiranja referentnog signala S oblika

Programsko rješenje referentnog signala S oblika u CoDeSys-u je zasnovano na logici napisanoj u programu Simatic Manager. Odabrani programski jezik je Continuous Function Chart (CFC). Na slici 5.2. je prikazan izgled bloka napisanog u CFC jeziku.

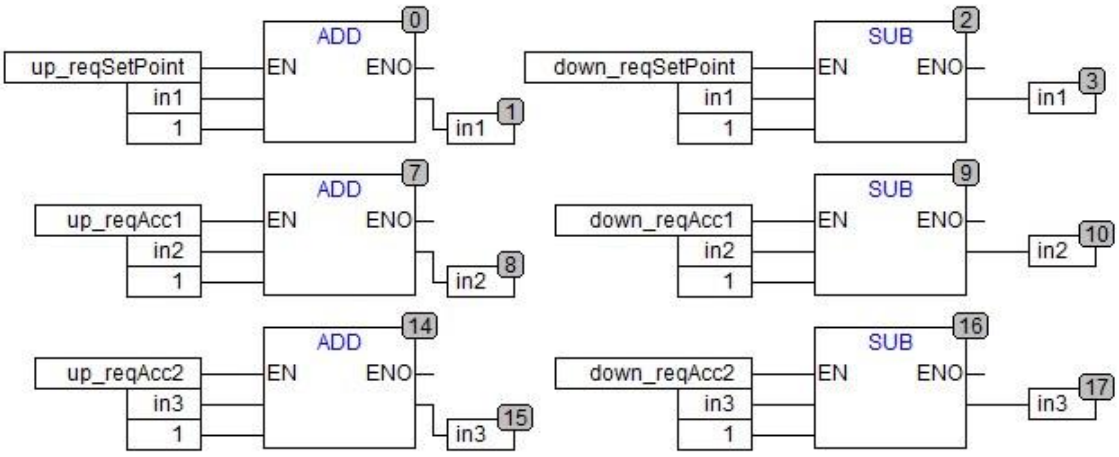


Slika 5.2. Izgled programskog bloka za S rampu u CODESYS programskom alatu

Za razliku od izvođenja programa na PLC-u, u CoDeSys-u se mora postaviti vrijeme izvođenja jednog ciklusa. Kada se napravi novi projekt, automatski se napravi POU (Program Organization Unit). POU služi za organiziranje programa isto kao i OB1 u Simatic Manageru i unutar njega moraju biti pozvani svi programi. Vrijeme izvođenja programa je postavljeno na 100 ms i ta vrijednost je upisana u blok S-rampe „IN_CLOCK“.



Slika 5.3. Namještanje izvođenja programa

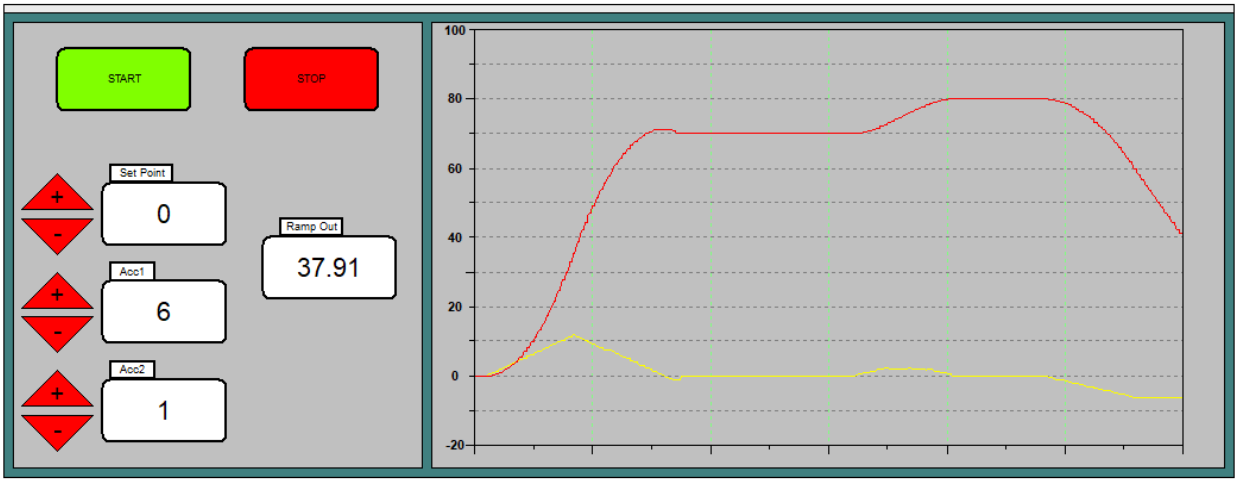


Slika 5.4. Mjenjanje parametara S rampe s HMI sučelja

Za izradu ovog programa upotrijebljen su dva funkcijska bloka, jedan za S-rampu i drugi za promjenu parametara rampe s jednostavnog HMI sučelja. Sa slike 5.4 se može vidjeti kako se može mijenjati vrijednost parametara s HMI sučelja. Kada dođe zahtjev „up_req“ ili „down_req“ za odgovarajući parametar, istom se dodaje ili oduzima 1 i nova vrijednost se upisuje u varijablu na izlazu iz bloka.

5.3. CODESYS VIZUALIZACIJA

Osim programiranja PLC-ova, CoDeSys nudi mogućnost vizualizacije programa odnosno izradu jednostavnog HMI (human machine interface) sučelja. HMI sučelje omogućava interakciju i prikaz bitnih parametara za neki automatizirani proces. Slika 5.5. prikazuje jednostavno sučelje za kontrolu S rampe.



Slika 5.5. CoDeSys vizualizacija

Sa slike 5.5 se mogu vidjeti dva gumba koji služe za pokretanje (START) i zaustavljanje (STOP) rampe. Također se vide 3 parametra koja operater može namjestiti po potrebi pritiskom na oznake + ili -, a to su maksimalna brzina (Set Point), maksimalna dopuštena akceleracija (Acc1) i korak akceleracije (Acc2). Ramp Out služi kao uvid u trenutnu brzinu. Na grafu se mogu pratiti krivulje brzine i akceleracije u realnom vremenu.

6. ZAKLJUČAK

Upravljanje brzinom u automatiziranim procesima je bitna zbog mogućnosti kontrole pomaka koji mogu biti brzi i precizni. Upravljanje procesom je moguća zbog PLC računala koja svoje zadaće mogu odrađivati u realnom vremenu. PLC-ovi su modularna i snažna računala koja su zbog svojih brojnih prednosti našla veliku primjenu u industriji. Za testiranje programa upotrijebljen je Siemensov PLC S7-400 koji je povezan s računalom ethernet kablom. Najčešće upotrebljavane rampe su linearne. Prednost S-rampe, za razliku od linearne, je ta što se vrijednost ubrzanja motora može mijenjati. Zbog toga se smanjuju udarna opterećenja zbog inercije sustava i potrošnja energije je manja. U ovom radu je napisan program za generiranje S signala. Jednostavno HMI sučelje je napravljeno u programskom alatu CoDeSys. HMI sučelje omogućava upravljanje parametrima S-rampe.

Programsko rješenje za generiranje S signala je napravljen u Simatic Manager-u i upotrebljen je ladder programski jezik.

LITERATURA

[1] Prof.dr.sc Nedeljko Perić, AUTOMATSKO UPRAVLJANJE, Zagreb, 2004.

[2] D. Slišković Procesna Automatizacija – predavanja, ETF, Osijek, 2009.

[3] https://www.pmdcorp.com/resources/get/s-curve-profiles-deep-dive-article
Datum pristupa linku 12.03.2019 u 12:08

[4] http://www.ni.com/product-documentation/4824/en/
Datum pristupa linku 12.03.2019 u 12:08

[5] Vinko Juranović, MAKETA POKRETNE TRAKE, Diplomski rad, Osijek 2017.

[6] https://www.scribd.com/doc/184536652/Simatic-s7-St-7pro1-Hrvatski
Datum pristupa linku 12.03.2019. u 12:08

[7] Danieli Automation, prezentacija - Level 1 basic software, CoDeSys programming basic Course

[8] Training Module A1-Siemens Simatic S7

SAŽETAK

U ovom radu je napravljen programski blok za generiranje referentnog signala S oblika. S-rampa je dobila ime po svojem obliku koji podsjeća na slovo S. Oblik krivulje se dobije integriranjem linearne rampe koja ovdje predstavlja akceleraciju. Prednost S-rampe je ta što se vrijednost akceleracije može kontrolirati. Upravljanje brzinom promjene je proces u kojem se brzina kontinuirano povećava ili smanjuje. Sustav upravljanja brzine je bitan u automatizaciji pogona kako bi se dobili precizni pomaci. U programu CoDeSys je izrađena jednostavna vizualizacija programa.

Ključne riječi: PLC, STEP7, CODESYS, S-krivulja

ABSTRACT

In this paper, a program block has been created to generate the reference S signal. The S-ramp is named because its shape resembles the letter S. The shape of the curve is obtained by integrating the linear ramp that represent acceleration. The advantage of the S-ramp is that the value of the acceleration can be controlled. Ramping is a process in which the speed continuously increases or decreases. The speed control system is essential in automation to get precise movements. A simple visualisation of the program was made in CoDeSys.

Keywords: PLC, STEP7, CODESYS, S-curve

ŽIVOTOPIS

Emil Blažević rođen je 31.8.1992. u Našicama. 2011. završio je matematičku gimnaziju „Isidora Kršnjavoga“ u Našicama. 2014. upisao je preddiplomski stručni studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer automatika. Stručnu praksu je odradio u tvrtki Danieli-Systec d.o.o.

Posjeduje vozačku dozvolu B kategorije, te se služi engleskim jezikom u govoru i pismu.