

# Kosimulacijska analiza rada sinkronog motora sa permanentnim magnetima i izrazenom reluktancijom u stvarnom vremenu

---

**Knol, Kristian**

**Master's thesis / Diplomski rad**

**2022**

*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:200:035871>

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

*Download date / Datum preuzimanja:* **2024-11-27**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

Sveučilišni studij elektrotehnike

KOSIMULACIJSKA ANALIZA RADA SINKRONOG  
MOTORA S PERMANENTNIM MAGNETIMA I  
IZRAŽENOM RELUKTANCIJOM U STVARNOM  
VREMENU

Diplomski rad

Kristian Knol

Osijek, 2022.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit**

Osijek, 15.09.2022.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za diplomski ispit**

<b>Ime i prezime Pristupnika:</b>	Kristian Knol
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Elektrotehnika
<b>Mat. br. Pristupnika, godina upisa:</b>	D-1329, 12.10.2020.
<b>OIB studenta:</b>	30901863938
<b>Mentor:</b>	Izv. prof. dr. sc. Marinko Barukčić
<b>Sumentor:</b>	Dr. sc. Tin Benšić
<b>Sumentor iz tvrtke:</b>	
<b>Predsjednik Povjerenstva:</b>	Prof. dr. sc. Željko Hederić
<b>Član Povjerenstva 1:</b>	Dr. sc. Tin Benšić
<b>Član Povjerenstva 2:</b>	Prof. dr. sc. Denis Pelin
<b>Naslov diplomskog rada:</b>	Kosimulacijska analiza rada sinkronog motora sa permanentnim magnetima i izrazenom reluktancijom u stvarnom vremenu
<b>Znanstvena grana diplomskog rada:</b>	<b>Elektrostrojarstvo (zn. polje elektrotehnika)</b>
<b>Zadatak diplomskog rada:</b>	Proučiti teoriju sinkronih motora sa permanentnim magnetima i izrazenom reluktancijom. Izraditi simulacijski model spomenutog motora koristeći se OPAL-RT OP5031 simulatorom u stvarnom vremenu. Sintetizirati regulator za spomenuti stroj te izraditi simulacijski model tog regulatora u XIL (X-in the loop) formi. Izvršiti simulacije dinamičkih odziva reguliranog i nereguliranog pogona sinkronog stroja sa permanentnim magnetima i izrazenom reluktancijom. Analizirati dinamička stanja navedenog pogona. Sumentor: dr.sc.Tin Benšić Rezervirano za: Kristian Knol
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	15.09.2022.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 28.09.2022.

**Ime i prezime studenta:**

Kristian Knol

**Studij:**

Diplomski sveučilišni studij Elektrotehnika

**Mat. br. studenta, godina upisa:**

D-1329, 12.10.2020.

**Turnitin podudaranje [%]:**

0

Ovom izjavom izjavljujem da je rad pod nazivom: **Kosimulacijska analiza rada sinkronog motora sa permanentnim magnetima i izrazenom reluktancijom u stvarnom vremenu**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Marinko Barukčić

i sumentora Dr. sc. Tin Benšić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:



# Sadržaj

<b>1.</b>	<b>Uvod</b>	<b>1</b>
<b>2.</b>	<b>Sinkroni stroj s permanentnim magnetima</b>	<b>3</b>
2.1.	Troosni model sinkronog motora	4
2.2.	Dvoosna teorija	5
2.3.	Dvoosni model sinkronog motora	7
2.4.	Vektorsko upravljanje	8
2.4.1.	Regulator struje	9
2.4.2.	Regulator kutne brzine	11
2.4.3.	Efekt zaleta	12
<b>3.</b>	<b>Stvarnovremenska kosimulacija</b>	<b>17</b>
3.1.	OPAL-RT stvarnovremenski sustav	17
3.1.1.	OPAL OP5031 i OP5600 simulatori	17
3.1.2.	Priprema modela u Simulinku	19
3.1.3.	Komunikacija i skaliranje signala	20
3.1.4.	Kompajliranje matematičkog modela	22
3.1.5.	Kontrola simulacije u stvarnom vremenu	23
3.1.6.	Snimanje podataka u RT-Labu	24
3.2.	dSpace stvarnovremenski sustav	24
3.2.1.	MicroLabBox simulator	24
3.2.2.	Povezivanje MicroLabBoxa s računalom	25
3.2.3.	Priprema matematičkog modela u Simulinku	26
3.2.4.	Komunikacijski blokovi	28
3.2.5.	Skaliranje signala	31
3.2.6.	Kompajliranje matematičkog modela	32
3.2.7.	Control Desk	32
3.2.8.	Sustav za kontrolu simulacije u stvarnom vremenu	33
3.2.9.	Snimanje podataka u Control Desku	34
3.2.10.	Pokretanje simulacije	34
3.2.11.	Validacija dSpace stvarnovremenskog simulatora	35
<b>4.</b>	<b>Analiza rezultata kosimulacije</b>	<b>42</b>
4.1.	Zalet i terećenje	45
4.2.	Reverziranje	49
4.3.	Rasterećenje i zasutavljanje	51
4.4.	HMI sučelja	55
<b>5.</b>	<b>Zaključak</b>	<b>57</b>
	Literatura	58
	Sažetak	60
	Abstract	61
	Životopis	62

# 1. UVOD

Ranije su se sinkroni strojevi dominantno upotrebljavali za proizvodnju električne energije, dakle kao generatori dok se kao motori nisu upotrebljavali zbog složenosti upravljanja. Razvojem tehnologije proizvodnje trajnih magneta je omogućena zamjena uzbudnog namota na rotoru sinkronog motora s trajnim magnetima te se tako eliminiraju gubici u bakru rotora. Razvojem energetske elektronike omogućeno je složenije upravljanje sinkronim motorima koji se sve više primjenjuju u industrijskoj primjeni zbog preciznosti i pouzdanosti. Iako je još uvijek skuplja proizvodnja sinkronih motora s permanentnim magnetima u odnosu na asinkrone strojeve, ističu se nad njima zbog veće gustoće snage po jedinici mase, a eliminiranjem bakra u rotoru stroja se smanjuje masa čime se učinkovitost povećava[1, 2, 3].

Sve je veća primjena sinkronih motora s permanentnim magnetima u industriji, a posebno automobilskoj industriji. Zbog strogih uvjeta koje automobilska industrija zahtjeva potrebno je kako bi se smanjili troškovi simulirati model motora u raznim situacijama što omogućuju stvarnovremenski simulatori poput dSpace-a i OPAL-a[4]. Kako u svakoj industriji tako i u automobilskoj potrebno je upravljati motorom što se postiže upotrebom frekvencijskih pretvarača i regulatora. Ubrzani razvoj regulatora omogućava da se preko niza simulacija razvije sustav za upravljanje koji se najprije testira na modelu, a kasnije i na prototipu. Tijekom postupka razvoja stvarnovremenski simulator se upotrebljava kako bi se dio sustava (bio on digitalan ili stvarni sustav) povezo s digitalnom ostatkom sustava. Kod simulacije motora s regulatorom je regulator prototip, a motor je implementiran u stvarnovremenskom simulatoru, ovakvo simuliranje se još naziva *hardware-in-the-loop*, HIL[4]. Pomoću ovakve simulacije je moguće podvrgnuti motor određenim testovima koji su opasni za testiranje na prototipu, ali i generalno simulirati razne scenarije koji se ne mogu svi testirati na prototipu[4, 5, 6].

Tema ovog rada je kosimulacijska analiza vektorskog upravljanja sinkronim motorom s permanentnim magnetima. Izvesti će se model sinkronog motor s permanentnim magnetima temeljem kojeg će se izvesti model regulator za vektorsko upravljanje. Model motora će biti implementiran na OPAL OP5031 simulator dok će model regulatora biti implementiran na dSpace-ov MicroLabBox te će se povezati u kosimulaciju u HIL (*hardware-in-the-loop*) sustav s ciljem testiranja modela regulatora.

U drugom poglavlju je najprije postavljen uvod u sinkrone strojeve te su definirane vrste sinkronih strojeva s permanentnim magnetima. Potom je definiran troosni model stroja. Nadalje je definiran pojam prostornog vektora i izvedena transformacija kojom se troosne veličine transformiraju u dvoosne veličine, tzv Park-ova transformacija. Primjenom definirane transformacije se model stroja transformira iz troosnog u dvoosni model. Temeljem dvoosnog modela su parametrirani PI regulatori za vektorsko upravljanje. Detaljno je prikazan izvod unutarnjeg i vanjskog regulacijskog kruga. Uz parametriranje regulatora je opisan i efekt zaleta te je predloženo i implementirano rješenje kojim se taj efekt poništava.

U trećem poglavlju su najprije definirani osnovni pojmovi vezani uz kosimulaciju, pojam stvarnovremenosti, HIL sustav. Detaljno su opisani dSpace MicroLabBox i OPAL OP5031 simulatori. Opisan je postupak izrade modela u Simulinku, te su opisane sve nužne postavke kako bi model izrađen u Simulinku bio kompatibilan za simulatore. Nadalje su detaljno opisani programi dSpace-ov Control Desk i OPAL-ov RT-Lab. Opisan je postupak izrade sučelja za praćenje veličina tokom izvođenja simulacije (u Control Desk-u za dSpace-ov simulator i u LabVIEW-u za OPAL-ov RT-Lab), snimanje podataka te samo pokretanje i zaustavljanje simulacije na oba programa. Opisan je eksperiment koji je proveden na MicroLabBox-u s kako bi se utvrdilo svojstvo stvarnovremenosti (šalje li simulator signale u stvarnom vremenu) te je provedena analiza na dobivenim rezultatima.

Prelaskom na četvrto poglavlje je najprije dan detaljan opis eksperimenta, koji uključuje zalet, terećenje, reverziranje, rasterećenje i zaustavljanje. Opisano je na koji način su spojeni simulatori te su opisani su problemi koji su se pojavili za vrijeme namještanja kosimulacije te su predložena rješenja istih. Nakon eksperimenta je provedena detaljna analiza snimljenih signala podijeljena u potpoglavlja. Za kraj su prikazana i opisana sučelja za praćenje signala upotrijebljena u oba simulatora.

## 2. SINKRONI STROJ S PERMANENTNIM MAGNETIMA

Teorija o sinkronim strojevima u ovom diplomskom radu se temelji na knjigama [7, 3, 8, 9]. Kod rotacijskih strojeva sinkrona brzina  $\omega_s$  definira se kao kutna brzina kojom okretno magnetsko polje rotira u zračnom rasporu i ovisi o broju pari polova ( $p$ ) i o frekvenciji napajanja ( $f$ ):

$$\omega_s = \frac{2\pi f}{p} \quad (2-1)$$

Svaki rotacijski stroj ima dvije kutne brzine: električnu  $\omega_{el}$  i mehaničku  $\omega_{meh}$  za koje vrijedi:  $\omega_{el} = p\omega_{meh}$  te prethodno definirana sinkrona kutna brzina nekog stroja je ujedno mehanička brzina stroja. Ako je kutna brzina rotora  $\omega_r$  jednaka mehaničkoj kutnoj brzini,  $\omega_r = \omega_{meh}$  tada se taj rotacijski stroj zove sinkroni stroj. Prema Krishnanu [3] sinkroni stroj se sastoji od:

1. Statorskog namota
2. Rotor na kojem se nalazi uzbuda stroja
3. Paketa limova na statoru i rotoru
4. Osovine preko koje se prenosi energija
5. Kućišta koje sve djelove drži na jednom mjestu

Sinkroni strojevi se dijele prema vrsti uzbude koja se nalazi na rotoru stroja, pa tako postoje:

1. Sinkroni strojevi s vanjskom uzbudom (dobiva se pomoću istosmjerne struje)
2. Reluktantni strojevi
3. Sinkroni strojevi s permanentnim magnetima

U ovom će se radu pozornost obratiti na sinkrone motore s permanentnim magnetima i izraženom reluktancijom (eng. *permanent magnets synchronous machine, PMSM*).

Otpor magnetskog kruga, reluktancija,  $R_m$ , definirana kao omjer magnetomotorne sile i magnetskog toka i ovisi o geometriji i permeabilnosti materijala [10]. Reluktancija kod rotacijskih strojeva se odnosi na zračni raspor između statora i rotora i značajno varira u iznosu kad stroj ima rotor s izraženim polovima.

Izvedbe PMSM-a ovise o položaju magnetima u rotoru [3]:

1. Na obodu rotora s vanjske strane (eng. *surface mounted PMSM*). U ovoj izvedbi su magneti direktno izloženi zračnom rasporu što omogućuje najveću gustoću magnetskog toka u zračnom rasporu. Nedostaci ove vrste motora proizlaze iz izloženosti magnetima zračnom rasporu te bi se pri većim brzinama magneti mogli odvojiti od rotora. Iz tog razloga se ova vrsta motora primjenjuje za manje brzine. Ova vrsta motora ima variranje iznosa reluktancije duž oboda rotora.
2. Na obodu stroja s unutarnje strane (eng. *surface-insert PMSM*). Magnetima se postavljaju u utore na obodu rotora stroja te su zbog toga magnetima mehanički čvršći u odnosu na magnetima s vanjske strane oboda. Rotori ovakve vrste stroja su cilindrični i nemaju izraženu reluktanciju.

3. Unutar paketa limova rotora stroja. Ovakvi strojevi su robusni te kao i strojevi s magnetima s unutarnje strane oboda rotora su cilindrični i nemaju izraženu reluktanciju.

Strojevi pod 2. i 3. se primjenjuju u pogonima gdje je zahtjevana brzina veća zbog svoje mehaničke čvrstoće.

## 2.1. Troosni model sinkronog motora

Prilikom modeliranja nekog stroja potrebno je navesti pretpostvke koje vrijede u modelu. Za sinkroni motor s premanentnim magnetima se uvode sljedeće pretpostavke:

1. statorski namoti su simetrični i sinusoidalno raspodjeljeni
2. indukcija o položaju rotora ovisi sinusoidalno
3. zanemarije se magnetsko zasićenje i promjena parametara

Iz prve pretpostavke proizlazi simetričnost magnetskog toka što ga stvara stator u zračnom rasporu (broj faza na statoru ne utječe na simetričnost). Izvedba rotora stroja, točnije položaj magneta ne utječe na valni oblik indukcije, nego samo na njen iznos preko iznosa induktiviteta. Treća pretpostavka podrazumjeva da su svi parametri stroja nepromjenjivi, tako npr. temperatura ne utječe na trajni magnet. Naponske jednadžbe statora glase:

$$\mathbf{u}_{abc} = R_s \mathbf{i}_{abc} + \frac{d}{dt} \boldsymbol{\psi}_{abc} \quad (2-2)$$

Gdje je:  $\mathbf{u}_{abc} = [u_a \ u_b \ u_c]^T$  statorski napon,  $\mathbf{i}_{abc} = [i_a \ i_b \ i_c]^T$  statorska struja,  $R_s$  otpor statorskog namota jednak u sve tri faze, a  $\boldsymbol{\psi}_{abc} = [\psi_a \ \psi_b \ \psi_c]^T$  ulančani magnetski tok koji nastaje protjecanjem struje kroz statorske namote i toka što ga stvara magnet na rotoru:

$$\boldsymbol{\psi}_{abc} = \mathbf{L}_s \mathbf{i}_{abc} + \boldsymbol{\psi}_{Mabc} \quad (2-3)$$

Gdje je:  $\mathbf{L}_s$  matrica induktiviteta i glasi:

$$\mathbf{L}_s = \begin{bmatrix} L_{ls} + L_A + L_B \cos(2\theta_r) & -\frac{1}{2}L_A + L_B \cos 2(\theta_r - \frac{\pi}{3}) & -\frac{1}{2}L_A + L_B \cos 2(\theta_r + \frac{\pi}{3}) \\ -\frac{1}{2}L_A + L_B \cos 2(\theta_r - \frac{\pi}{3}) & L_{ls} + L_A + L_B \cos(2\theta_r - \frac{2\pi}{3}) & -\frac{1}{2}L_A + L_B \cos 2(\theta_r + \pi) \\ -\frac{1}{2}L_A + L_B \cos 2(\theta_r + \frac{\pi}{3}) & -\frac{1}{2}L_A + L_B \cos 2(\theta_r + \pi) & L_{ls} + L_A + L_B \cos(2\theta_r + \frac{2\pi}{3}) \end{bmatrix} \quad (2-4)$$

Gdje su:  $\theta_r$  - položaj rotora,  $L_{ls}$  - rasipni induktivitet, a  $L_A$  i  $L_B$  su induktiviteti faza dvofaznog sinkronog stroja temeljem kojeg je modeliran troosni model stroja koji je polazišna točka ovog diplomskog rada (detaljniji izvod se nalazi u knjizi [7]), njihovi iznosi su:

$$L_A = \left(\frac{N_s}{p}\right)^2 \pi \mu_0 r l \alpha_1 \quad (2-5)$$

$$L_B = \frac{1}{2} \left(\frac{N_s}{p}\right)^2 \pi \mu_0 r l \alpha_2 \quad (2-6)$$

gdje su:  $N_s$  broj namota jedne faze statora,  $\mu_0$  permeabilnost vakuuma,  $r$  polumjer stroja,  $l$  duljina stroja,  $\alpha_1$  najveći iznos zračnog raspora,  $\alpha_2$  najmanji iznos zračnog raspora.

Gibanjem rotora (promjenom kuta  $\theta_r$ ), magnetski tok trajnog magneta se ulačava s namotima faza statora:

$$\psi_{Mabc} = \psi_M \begin{bmatrix} \cos(\theta_r) \\ \cos(\theta_r - \frac{2\pi}{3}) \\ \cos(\theta_r + \frac{2\pi}{3}) \end{bmatrix} \quad (2-7)$$

Položaj (kut) rotora i brzina su povezani preko:

$$\frac{d\theta_r}{dt} = \omega_r \quad (2-8)$$

Pomnoži li se jednadžba (2-2) s lijeve strane s matricnom  $\mathbf{i}_{abc}^T$ , pri tome je važno naglasiti da je ulančani magnetski tok funkcija vremena i položaja i rastavlja se na zbroj funkcije tih varijabli:

$$\mathbf{i}_{abc}^T \mathbf{u}_{abc} = \mathbf{i}_{abc}^T R_s \mathbf{i}_{abc} + \mathbf{i}_{abc}^T \frac{\partial \psi_{abc}}{\partial t} + \mathbf{i}_{abc}^T \omega_r \frac{\partial \psi_{abc}}{\partial \theta_r} \quad (2-9)$$

dobivena jednadžba predstavlja bilancu snage. Gdje je  $\mathbf{i}_{abc}^T \mathbf{u}_{abc}$  električna snaga jednaka zbroju:  $\mathbf{i}_{abc}^T R_s \mathbf{i}_{abc}$  snage disipacije topline u bakru namota,  $\mathbf{i}_{abc}^T \frac{\partial \psi_{abc}}{\partial t}$  snage uskladištene u induktivitetima i  $\mathbf{i}_{abc}^T \omega_r \frac{\partial \psi_{abc}}{\partial \theta_r}$  mehaničke snage.

Tada se iz izraza za mehaničku snagu može izvesti izraz elektromagnetski moment stroja:

$$M_e = \frac{P_e}{\omega_r} = \mathbf{i}_{abc}^T \frac{\partial \psi_{abc}}{\partial \theta_r} \quad (2-10)$$

Primjenom drugog Newtonovg zakona za kružno gibanje povezuju se moment motora, moment tereta i moment ubrzanja:

$$M_u = J \frac{d\omega_r}{dt} = M_e - M_t \quad (2-11)$$

Gdje je:  $J$  - inercija motora,  $M_t$  - moment tereta, a  $M_u$  - moment ubrzanja.

Varijable izvedenog modela motora su ovisne o položaju rotora, a posljedično i o vremenu, što model čini nelinearnim vremenski promjenjivim. Kako bi se olakšala analiza motora potrebno je primijeniti transformaciju koja će model motora transformirati u vremenski nepromjenjivi model (ukloniti vremensku varijablu). Takva transformacija će se objasniti u sljedećem potpoglavlju.

## 2.2. Dvoosna teorija

1929. godine R.H.Park je osmislio transformaciju kojom se prelazi iz troosnog u dvoosni sustav[11]. Svrha ove transformacije je ukloniti induktivitete ovisne o položaju rotora, odnosno svesti model stroja na linearni, vremenski nepromjenjiv model. Opća Parkova transformacija transformira električne veličine (struje, napone, tokove) iz mirujućeg sustava u sustav koji rotira proizvoljnom brzinom. Jedan takav sustav prikazan je na slici 2.1.

Prostorni vektor  $\mathbf{x}$  koji može biti struja, napon ili tok može se zapisati pomoću trenutnih vrijednosti svih faza (faze su simetrične i pomaknute za  $\frac{2\pi}{3}$ ):

$$\mathbf{x}_s = \frac{2}{3}(x_a a_1 + x_b b_1 + x_c c_1) \quad (2-12)$$

Gdje je  $\frac{2}{3}$  faktor očuvanja amplitude,  $x_a$ ,  $x_b$  i  $x_c$  trenutne vrijednosti faznih veličina trofaznog

simetričnog sustava, a  $a_1, b_1, c_1$  su jedinični vektori prostornog vektora zapisanih pomoću  $dq$  komponenti, definirani kao[12]:

$$a_1 = \cos(\theta_r) - j \sin(\theta_r) \quad (2-13)$$

$$b_1 = \cos\left(\theta_r - \frac{2\pi}{3}\right) - j \sin\left(\theta_r - \frac{2\pi}{3}\right) \quad (2-14)$$

$$c_1 = \cos\left(\theta_r + \frac{2\pi}{3}\right) - j \sin\left(\theta_r + \frac{2\pi}{3}\right) \quad (2-15)$$

Ako se u jednadžbu (2-12) uvrste izrazi za jedinične vektore i ako se grupiraju realni i imaginarni dijelovi dobije se:

$$\mathbf{x} = \frac{2}{3} \left\{ x_a \cos(\theta_r) + x_b \cos\left(\theta_r - \frac{2\pi}{3}\right) + x_c \cos\left(\theta_r + \frac{2\pi}{3}\right) - j(x_a \sin(\theta_r) + x_b \sin\left(\theta_r - \frac{2\pi}{3}\right) + x_c \sin\left(\theta_r + \frac{2\pi}{3}\right)) \right\} \quad (2-16)$$

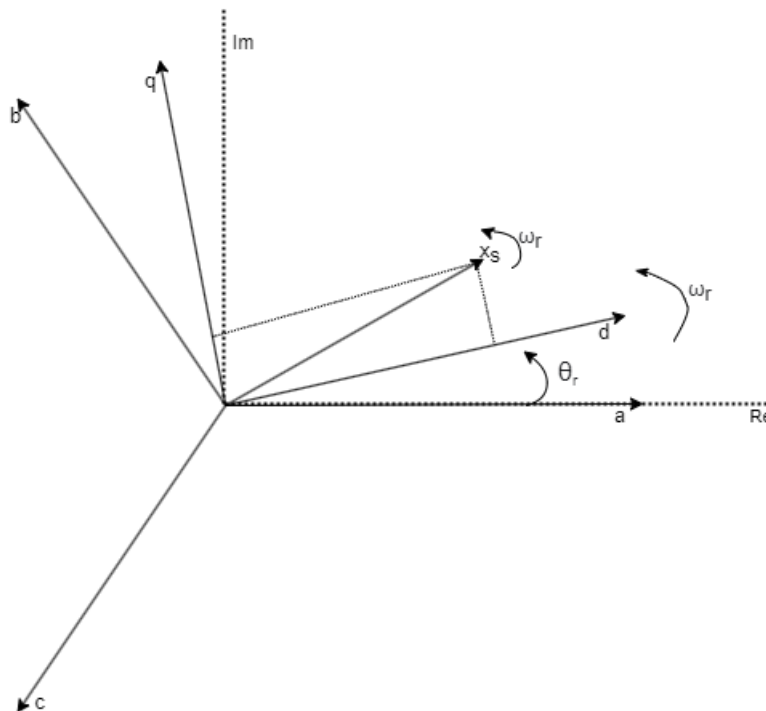
Zapiše li se jednadžba u matricnom obliku, dobije se jednadžba transformacije iz troosnog mirujućeg sustava u dvoosni rotirajući sustav:

$$\mathbf{x}_{dq} = \mathbf{K} \mathbf{x}_{abc} \quad (2-17)$$

gdje je:

$$\mathbf{K} = \frac{2}{3} \begin{bmatrix} \cos(\theta_r) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ -\sin(\theta_r) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \quad (2-18)$$

Jednadžba (2-17) još se naziva Parkova transformacija.



**SL. 2.1:** dvoosni rotirajući sustav u usporedbi s troosnim mirujućim sustavom

Jednadžba pomoću koje se iz dvoosnog sustava prelazi u troosni sustav glasi:

$$\mathbf{x}_{abc} = \mathbf{K}^{-1}\mathbf{x}_{dq} \quad (2-19)$$

Gdje matrica  $\mathbf{K}^{-1}$  glasi:

$$\mathbf{K}^{-1} = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & -\sin\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & -\sin\left(\theta_r + \frac{2\pi}{3}\right) \end{bmatrix} \quad (2-20)$$

Pomoću jednadžbi (2-17) i (2-19) omogućen je prijelaz iz jednog koordinatnog sustava u drugi pa se tako mogu modeli simulirati u jednom sustavu, a u drugom sustavu prikazivati mjerene veličine. Takva jedna transformacija će se primijeniti na jednadžbe PMSM-a u sljedećem potpoglavlju.

### 2.3. Dvoosni model sinkronog motora

Primjenom transformacije definirane jednadžbom (2-19) na jednadžbu (2-2) dobije se:

$$\mathbf{K}^{-1}\mathbf{u}_{dq} = R_s\mathbf{K}^{-1}\mathbf{i}_{dq} + \frac{d\mathbf{K}^{-1}\boldsymbol{\psi}_{dq}}{dt} \quad (2-21)$$

Derivacija u jednadžbi (2-21) je složena i raspisana glasi:

$$\frac{d\mathbf{K}^{-1}\boldsymbol{\psi}_{dq}}{dt} = \mathbf{K}^{-1}\frac{d\boldsymbol{\psi}_{dq}}{dt} + \boldsymbol{\psi}_{dq}\frac{d\mathbf{K}^{-1}}{dt} \quad (2-22)$$

Sljedeći korak je pomnožiti jednadžbu (2-21) s matricom  $\mathbf{K}$  s lijeve strane:

$$\mathbf{K}\mathbf{K}^{-1}\mathbf{u}_{dq} = \mathbf{K}R_s\mathbf{K}^{-1}\mathbf{i}_{dq} + \mathbf{K}\mathbf{K}^{-1}\frac{d}{dt}\boldsymbol{\psi}_{dq} + \mathbf{K}\frac{d\mathbf{K}^{-1}}{dt}\boldsymbol{\psi}_{dq} \quad (2-23)$$

Jednadžba se dalje pojednostavljuje. Vrijedi  $\mathbf{K}\mathbf{K}^{-1} = \mathbf{I}$ .

$$\mathbf{u}_{dq} = R_s\mathbf{i}_{dq} + \frac{d}{dt}\boldsymbol{\psi}_{dq} + \omega\mathbf{J}_r\boldsymbol{\psi}_{dq} \quad (2-24)$$

Gdje je  $\mathbf{J}_r$  rotacijska matrica i glasi:

$$\mathbf{J}_r = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (2-25)$$

Procedura je jednaka i za jednadžbu (2-3), primjenom jednadžbe transformacije (2-19), dobije se:

$$\mathbf{K}^{-1}\boldsymbol{\psi}_{dq} = \mathbf{L}_s\mathbf{K}^{-1}\mathbf{i}_{dq} + \mathbf{K}^{-1}\boldsymbol{\psi}_{Mdq} \quad (2-26)$$

Nakon množenja s matricom  $\mathbf{K}$  s lijeve strane i pojednostavljenja se dobije:

$$\boldsymbol{\psi}_{dq} = \mathbf{L}_{dq}\mathbf{i}_{dq} + \boldsymbol{\psi}_{Mdq} \quad (2-27)$$



gdje su:

$$\mathbf{L}_{dq} = \begin{bmatrix} L_{ls} + \frac{3}{2}(L_A + L_B) & 0 \\ 0 & L_{ls} + \frac{3}{2}(L_A - L_B) \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \quad (2-28)$$

$$\boldsymbol{\psi}_{Mdq} = \psi_M \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2-29)$$

Prema jednadžbi (2-29), referentna  $d$  os  $dq$  koordinatnog sustava je poravnata sa osi na kojoj su magneti i u tom položaju vrijedni,  $\theta_r = 0$ .

Prelaskom u dvoosni sustav mijenja se i izraz za moment motora:

$$M_e = (\mathbf{K}^{-1}\mathbf{i}_{dq})^T \left( \frac{\partial \mathbf{K}^{-1}}{\partial \theta_r} \boldsymbol{\psi}_{dq} + \frac{\partial \boldsymbol{\psi}_{dq}}{\partial \theta_r} \mathbf{K}^{-1} \right) \quad (2-30)$$

Drugi član u zagradi je nula jer magnetski tok u  $dq$  sustavu nije eksplcitno funkcija položaja (ovisnot toka o položaju je uklonjena Park-ovom transformacijom). Dalje se izraz pojednostavljuje:

$$M_e = \mathbf{i}_{dq}^T (\mathbf{K}^{-1})^T \frac{\partial \mathbf{K}^{-1}}{\partial \theta_r} \boldsymbol{\psi}_{dq} = \frac{3}{2} p (\mathbf{i}_{dq}^T \mathbf{J}_r \boldsymbol{\psi}_{dq}) \quad (2-31)$$

Nakon množenja matrica u zagradi dobije se:

$$M_e = \frac{3}{2} p (\psi_d i_q - \psi_q i_d) \quad (2-32)$$

Tokovi se zamjenjuju s pripadnim izrazima struje, induktiviteta i toka trajnog magneta prema jednadžbi (2-27) te se dobije konačni izraz za moment stroja u  $dq$  sustavu:

$$M_e = \frac{3}{2} p (\psi_M i_q - (L_q - L_d) i_d i_q) \quad (2-33)$$

Prvi član u zagradi je moment stroja uzrokovan trajnim magnetom i strujom dok drugi član predstavlja moment uzrokovan izraženom rekuktancijom. Dobiveni model predstavljen jednadžbama (2-24), (2-27), (2-28), (2-29) i (2-33) je model motora zapisan u rotorskom koordinatnom sustavu jer je to jedini sustav u kojem induktiviteti su u potpunosti neovisni o položaju rotora te je zato ovaj model linearan i vremenski nepromjenjiv. Temeljem ovog modela će se parametrirati regulatori za vektorsko upravljanje.

## 2.4. Vektorsko upravljanje

Ideja vektorskog upravljanja (eng. *Field oriented control, FOC*) se temelji na ideji zasebnog upravljanja magnetskim poljem i momentom stroja. Model PMSM-a izveden u potpoglavlju 2.3 je u rotorskom koordinatnom sustavu pa je za FOC potrebno poznavanje iznosa i položaja ulančanog magnetskog toka rotora, dok se samo vektorsko upravljanje PMSM-om realizira pomoću PI regulatora, no za to je potreban linearni model PMSM-a koji je izveden u 2.3. Kako momentna jednadžba (2-33) u svom izrazu ima struje obje osi potrebno je jednu struju postaviti na nulu. Struja  $i_d$  se postavi na nulu i na taj način se ukloni ovisnost momenta o dvije struje, a poništi se i utjecaj reluktantnog momenta. Upravljanjem strujom  $i_d$  se upravlja magnetskim poljem.

### 2.4.1 Regulator struje

Razdvajanjem jednadžbe (2-24) na  $d$  i  $q$  komponentne dobije se:

$$u_d = R_s i_d - \omega_r L_q i_q + \frac{d}{dt} L_d i_d \quad (2-34)$$

$$u_q = R_s i_q + \omega_r L_d i_d + \omega_r \psi_M + \frac{d}{dt} L_q i_q \quad (2-35)$$

Kako bi se izbjeglo rješavanje diferencijalnih jednadžbi uvodi se Laplaceova transformacija koja transformira funkcije iz vremenske domene u kompleksnu domenu[13]:

$$F(s) = \int_0^{\infty} f(t) e^{-st} dt \quad (2-36)$$

gdje je:  $F(s)$  funkcija u kompleksnoj domeni,  $f(t)$  funkcija u vremenskoj domeni,  $s = \sigma + j\omega$  kompleksna varijabla.

Posebno je važan teorem o deriviranju:

$$\mathcal{L}\left\{\frac{df(t)}{dt}\right\} = sF(s) \quad (2-37)$$

Pomoću jednadžbe (2-37) diferencijalne jednadžbe prelaze u algebarske jednadžbe. Jednadžbe (2-54) i (2-55) prelaskom u kompleksnu domenu glase:

$$U_d(s) = R_s I_d(s) - \omega_r L_q I_q(s) + s L_d I_d(s) \quad (2-38)$$

$$U_q(s) = R_s I_q(s) + \omega_r L_d I_d(s) + \omega_r \psi_M + s L_q I_q(s) \quad (2-39)$$

Daljnijim grupiranjem se dobije:

$$U_d(s) + E_d(s) = I_d(s)(R_s + s L_d) \quad (2-40)$$

$$U_q(s) + E_q(s) = I_q(s)(R_s + s L_q) \quad (2-41)$$

gdje su:

$$E_d(s) = -\omega_r L_q I_q(s) \quad (2-42)$$

$$E_q(s) = \omega_r L_d I_d(s) + \omega_r \psi_M \quad (2-43)$$

Protuelektromotne sile.

Ako se jednadžbe (2-40) i (2-41) zapišu kao ovisnost struje o napnu dobije se prijenosna funkcija pomoću koje se parametrira PI regulator za upravljanje strujom:

$$\frac{I_d(s)}{U_d(s) - E_d(s)} = \frac{1}{R_s + s T_{sd}} \quad (2-44)$$

$$\frac{I_q(s)}{U_q(s) - E_q(s)} = \frac{1}{R_s + s T_{sq}} \quad (2-45)$$

gdje su:  $T_{sd} = \frac{L_d}{R_s}$ ,  $T_{sq} = \frac{L_q}{R_s}$

Unutarnji regulacijski krug (regulacija struje) se sastoji od PI regulatora i prijenosne funkcije sustava, te prijenosna funkcija otvorenog regulacijskog kruga glasi (dalje u tekstu će se pokazati izvod za  $d$  komponentu, a postupak je identičan za  $q$  komponentu):

$$G_0(s) = \frac{1 + s T_D}{s T_i} \frac{1}{R_s + s T_{sd}} \quad (2-46)$$

PI regulator je parametriran po tehničkom optimumu koji nastoji poništiti dominantni pol sustava i tada vrijedni  $T_D = T_{sd}$  i daljnjim sređivanjem izraza se dobije:

$$G_0(s) = \frac{1}{sT_i R_s} \quad (2-47)$$

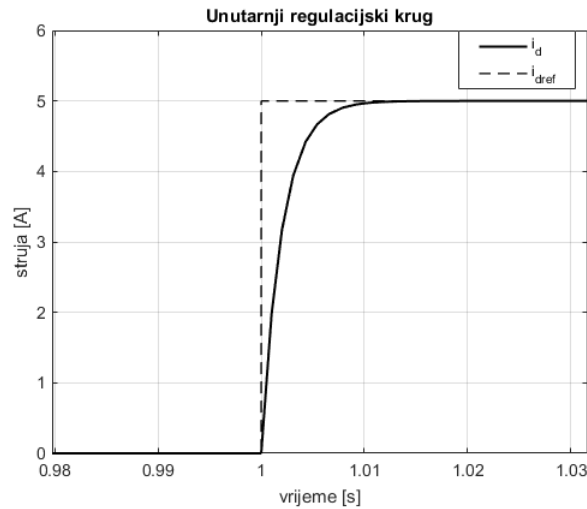
Regulator kao ulaznu veličinu ima razliku između referentne ( $I_d^*$ ) i mjerene vrijednosti struje ( $I_d$ ), pa se povratnom vezom struje postiže zatvoreni regulacijski sustav čija prijenosna funkcija nakon sređivanja glasi:

$$\frac{I_d}{I_d^*} = G_x(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{1}{1 + sT_i R_s} \quad (2-48)$$

Kako je ranije naglašeno,  $T_D = T_{sd} = \frac{L_d}{R_s}$ , dok je  $T_i$  po tehničkom optimumu jednak:

$$T_i = 2k_s T_x = 2 \frac{1}{R_s} T_x = \frac{2T_x}{R_s} \quad (2-49)$$

Gdje je  $T_x$  vremenska konstanta zatvorenog kruga i iznosi  $T_x = \frac{1}{\omega}$ . Slika 2.2 prikazuje odziv zatvorenog regulacijskog kruga na skokovitu promjenu referentne struje, koji je po funkciji oblika PT1 član.



**SL. 2.2:** odziv unutarnjeg regulacijskog kruga

U ovom radu je model PI regulatora izveden kao suma proporcionalnog i integralnog člana:

$$G_{PI}(s) = K_p + \frac{K_i}{s} = \frac{sK_p + k_i}{s} = \frac{\frac{K_p}{K_i}s + 1}{\frac{1}{K_i}s} \quad (2-50)$$

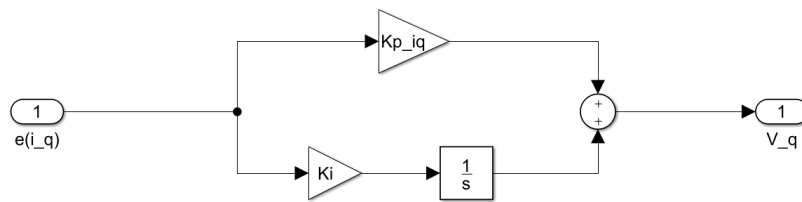
te vrijede jednakosti:  $\frac{K_p}{K_i} = T_D$  i  $T_i = \frac{1}{K_i}$ , a nakon sređivanja se dobiju izrazi za vrijednosti parametara PI regulatora (za  $d$  i  $q$  komponente):

$$K_{id} = K_{iq} = \frac{R_s}{2T_x} \quad (2-51)$$

$$K_{Pd} = \frac{L_d}{2T_x} \quad (2-52)$$

$$K_{Pq} = \frac{L_q}{2T_x} \quad (2-53)$$

Slika 2.3 prikazuje PI regulator realiziran u Simulinku



**SL. 2.3:** PI regulator

Dimenzija izlaznog signala PI regulatora za struju je napon ( $V_d$  i  $V_q$ ) te je potrebno uračunati utjecaj protuelektromotorne sile na napon motora pa se zato naponu kojeg regulator proračuna još doda iznos protuelektromotorne sile pa je napon koji se prosljeđuje motoru iznosa:

$$U_d = E_d + V_d = -\omega L_q I_q + (K_{Pd} + \frac{K_{id}}{s})(I_d^* - I_d) \quad (2-54)$$

$$U_q = E_q + V_q = \omega(L_d I_d + \psi_M) + (K_{Pq} + \frac{K_{iq}}{s})(I_q^* - I_q) \quad (2-55)$$

#### 2.4.2 Regulator kutne brzine

Regulator brzine se kaskadno spaja na regulator struje koji kontrolira iznos momenta stroja. Ranije je naglašeno da je postavljanjem referentne vrijednosti struje  $i_d$  na nulu postignuto upravljanje momentom PMSM-a samo  $i_q$  strujom.

Spajanjem prijenosna funkcija PI regulatora brzine stroja kaskadno sa funkcijom  $G_x$  unutarnjeg regulacijskog kruga dobije se otvoreni regulacijski sustav brzine:

$$G_0(s) = \frac{T_{D\omega}s + 1}{T_{i\omega}s} \frac{1}{1 + 2T_x s} \frac{1}{Js} \quad (2-56)$$

PI regulator brzine je parametriran po simetričnom optimumu zbog astatizma prvog reda (zbog integralnog djelovanja u sustavu). Po simetričnom optimumu su parametri PI regulatora iznosa:

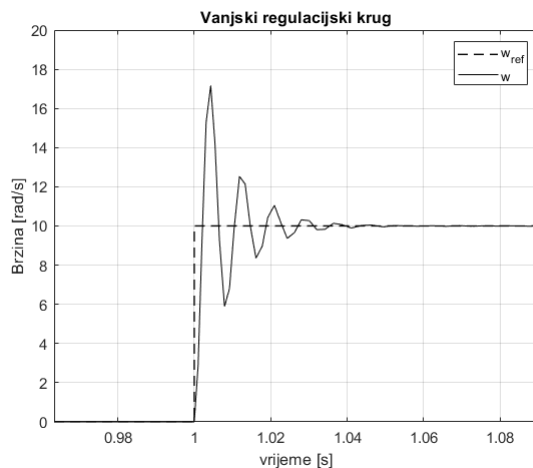
$$T_{D\omega} = 4T_x \quad (2-57)$$

$$T_{i\omega} = 8k_s T_x^2 = \frac{8T_x^2}{J} \quad (2-58)$$

Za ispravan rad regulatora brzine porebno je od otvorenog regulacijskog kruga (jednadžba (2-56)) napraviti zatvoreni regulacijski krug, čija je funkcija zapisana u jednadžbi (2-59)

$$G_x(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{4T_x s + 1}{16T_x^3 s^3 + 8T_x^2 s^2 + 4T_x s + 1} \quad (2-59)$$

čiji je odziv na skokovitu promjenu referentne brzine prikazan slikom 2.4



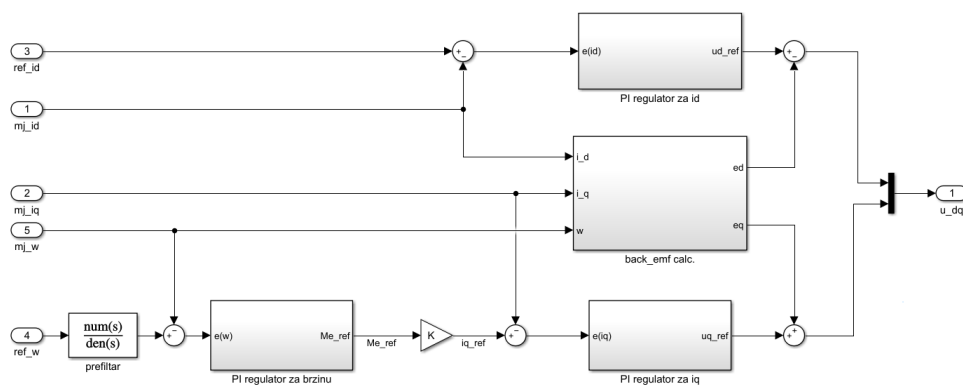
**SL. 2.4:** Odziv zatvorenog regulacijskog kruga brzine

PI regulator brzine je kao i regulatori brzine realiziran sumom proporcionalnog i integralnog člana (jednadžba (2-50)). Primjenom relacija:  $T_{D\omega} = \frac{K_p}{K_i}$  i  $T_{i\omega} = \frac{1}{K_i}$  dobiju se koeficijenti regulatora brzine:

$$K_{P\omega} = \frac{J}{2T_x} \quad (2-60)$$

$$K_{I\omega} = \frac{J}{8T_x^2} \quad (2-61)$$

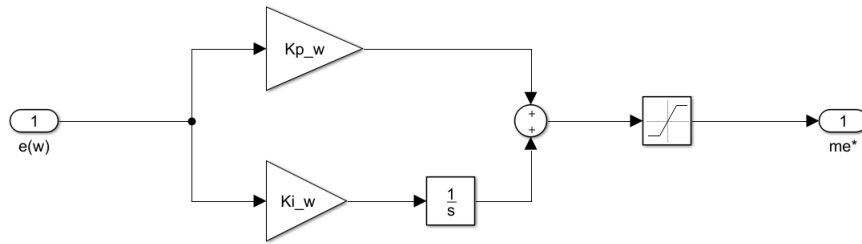
Konačni model regulatora za vektorsko upravljanje sadrži dva PI regulatora struje i PI regulatora brzine i blok za računanje protuelektromotorne sile, slika 2.5.



**SL. 2.5:** Model regulatora u Simulinku

### 2.4.3 Efekt zaleta

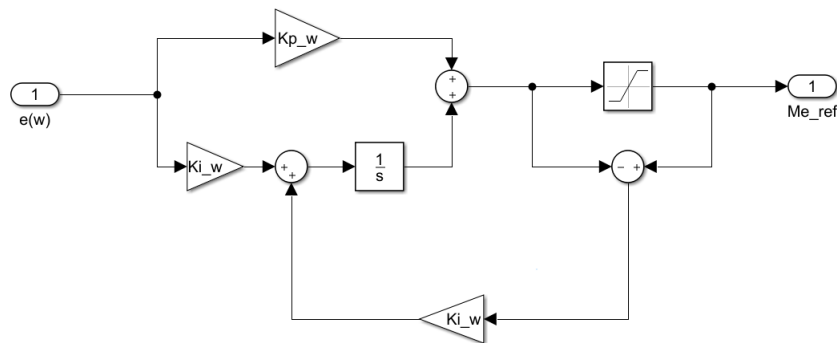
Prilikom parametriranja regulatora uvedeno je ograničenje iznosa kojeg regulator šalje jer u stvarnome sustavu aktuatori ne mogu dati više od maksimalne vrijednosti veličine (npr. struje, napona, momenta i sl.). U MATLAB/Simulinku je ograničenje uvedeno u obliku bloka *saturation*, slika 2.6.



**SL. 2.6:** Model regulatora sa zasićenjem

U slučaju kad je limitator na graničnoj vrijednosti, integrator regulatora i dalje integrira regulacijsko odstupanje i postiže vrijednost koja je veća od granične vrijednosti limitatora, pojava se zove "efekt zaleta", zasićenje integratora (eng. *integrator windup*). Kada se u nekom trenutku kasnije promjeni predznak regulacijskog odstupanja pojavljuje se određeno kašnjenje, jer integrator ima veću vrijednost od limitatora i to uzrokuje kašnjenje u odzivu [13]. Problemi koji proizlaze iz ove pojave mogu biti: preveliko odstupanje, predugo vrijeme ustaljivanja ili čak pojavu nestabilnosti u sustavu [14, 15].

Rješenje problema zasićenja integratora poznato je pod nazivom ograničenje izlazne veličine integratora (eng. *Anti-windup*) i postoje brojna rješenja. U radu je primijenjen tzv. postupak povratnog računanja (eng. *tracking back calculation*). Kad izlazni signal regulatora pređe granični iznos limitatora, formira se signal povratne veze od razlike izlaznog signala iz regulatora i limitatora. Taj signal se pribraja ulaznoj varijbli integralnog člana i tako smanjuje njegovo djelovanje. U slučaju kad je vrijednost manja od graničen vrijednosti onda nema signala povratne veze, odnosno signal povratne veze je vrijednosti nula. Rješenje je prikazano slikom 2.7.

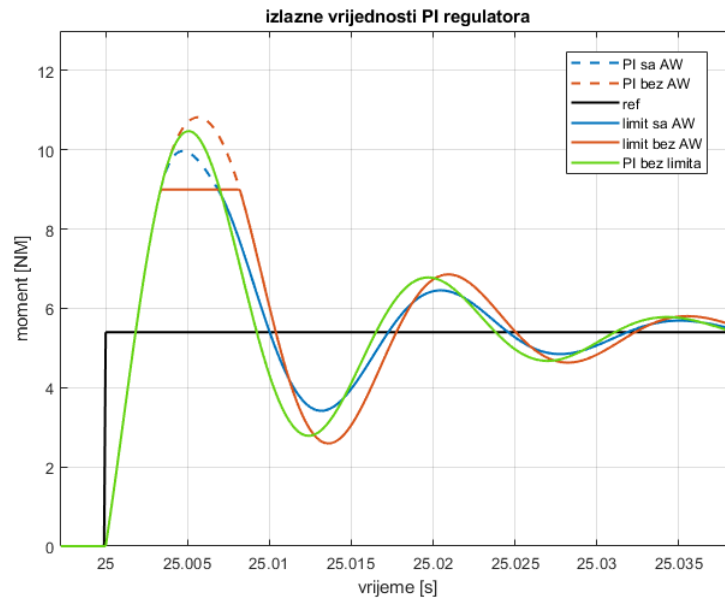


**SL. 2.7:** PI regulator s *anit-windupom*

Signal povratne veze je potrebno skalirati prije nego se proslijedi integratoru. Najčešće se pojačanje povratne veze postavi na iznos koji je jednak iznosu pojačanja integralnog člana regulatora, što je primijenjeno i u ovom diplomskom radu. Za slučaj da je vrijednost pojačanja manja od predložene tada nema utjecaja na integralni član, dok u slučaju veće vrijednosti pojačanja od predložene dolazi do prevelikog utjecaja na integralni član te se ne može postići referentna vrijednost.

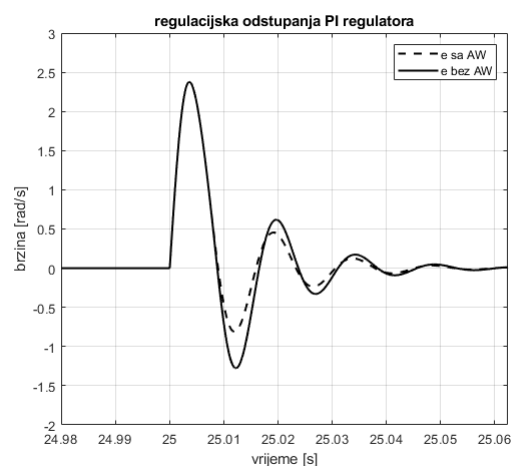
Na utjecaj *anti-windupa* najvećim djelom utječe limitator postavljen na izlazu iz regulatora. Na primjeru regulatora brzine će se pokazati utjecaj limitatora na *anti-windup* te sam utjecaj *anti-windupa*. Postaviti će se tri različita limita izlaznog momenta u Nm. Uspredivati će se izlazni signali iz regulatora i iz limitatora.

Najprije je iznos limita postavljen na najveći vrijednost od 9 Nm, a svaki sljedeći korak je iznos limita smanjen za 1 Nm, slika 2.8



SL. 2.8: Uspredba grafova pri limitu od 9 Nm

Za usporedbu je još na grafu dodan izlazni signal iz regulatora kad nema tehničkog ograničenja (bez limitatora). Vidljivo je kako *anti-windup* ne utječe značajnije na vrijeme ustaljivanja, ali utječe na iznos oscilacija, točnije smanjuje iznos oscilacija te na nagib krivulje. Primjenom *anti-windupa* se ne utječe samo na smanjenje integralnog djelovanja već i na regulacijsko odsutapanje, slika 2.9

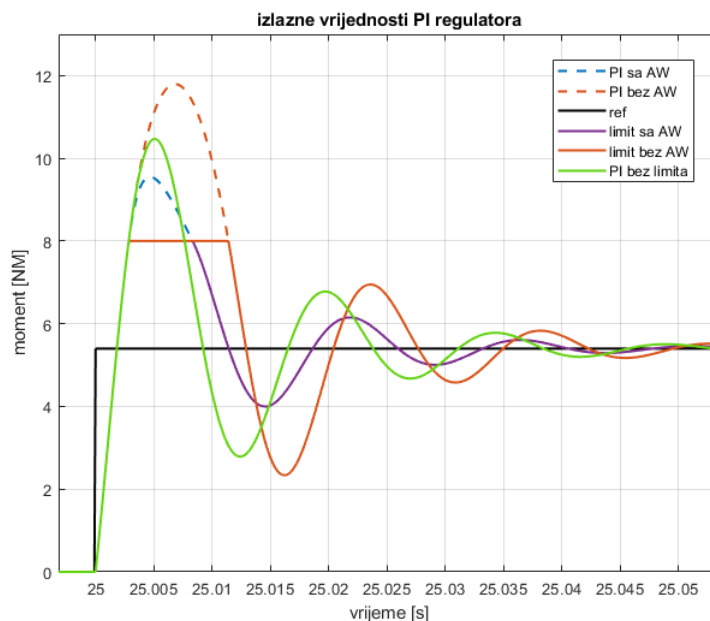


SL. 2.9: Regulacijska odstupanja pri limitu od 9 Nm

Korištenjem *anti-windupa* vrijednost izlaznog signala ranije padne ispod limita zbog smanjivanja utjecaja na integralno djelovanje, a time se i brže približava referentnoj vrijed-

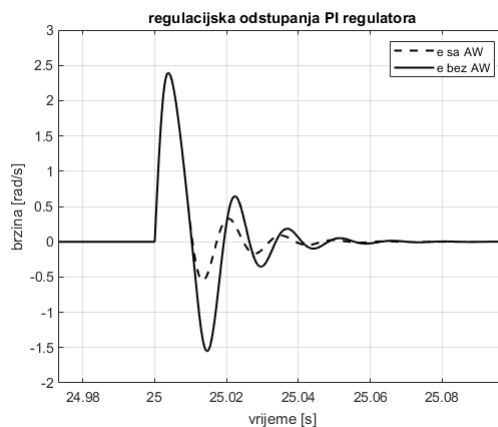
nosti što utječe na smanjenje regulacijskog odstupanja.

Daljnijim smanjivanjem limita se dobiju još veće razlike između izlaznih signala iz regulatora, slika 2.10.



**SL. 2.10:** Uspredba grafova pri limitu od 8 Nm

Vidljivo je kako kad nema *anti-windup* algoritma maksimalni iznos izlazne veličine PI regulatora poraste u odnosu na isti signal pri veće iznosu limita, što uzrokuje veće amplitude u oscilaciji. Promjene se pojave i kod regulacijskog odstupanja, slika 2.11

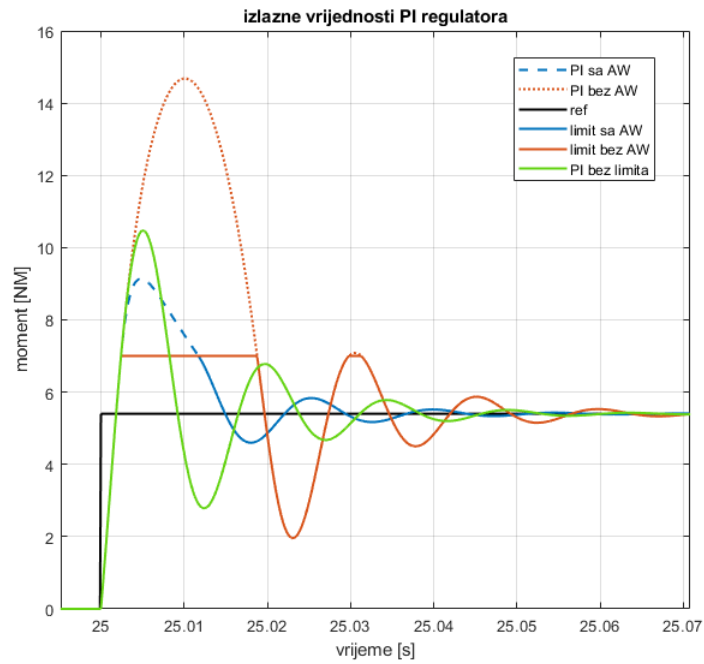


**SL. 2.11:** Regulacijska odstupanja pri limitu od 8 Nm

U odnosu na limit od 9 Nm, regulacijsko odstupanje bez *anti-windupa* je veće dok je regulacijsko odstupanje sa *anti-windupom* manje po iznosu.

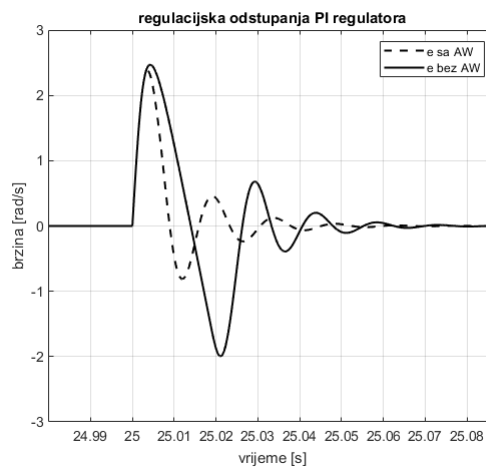
Daljnijim smanjenjem limita na 7 Nm dobije se, slika 2.12.





**SL. 2.12:** Uspredba grafova pri limitu od 7 Nm

Primjetan je značajni skok amplitude izlaznog signala PI regulatora bez *anti-windupa*, a i regulacijskog odstupanja prikazan slikom 2.13. Izlazni signal je ograničen na 7 Nm i zbog toga se sporije odvijaju promjene uzrokovane priključenjem tereta, što rezultira porastom regulacijskog odstupanja jer se ne utječe na integralno djelovanje. U konačnici se ustaljivanje odvija sporije, nego sa *anti-windupom*.



**SL. 2.13:** Regulacijska odstupanja pri limitu od 7 Nm

### 3. STVARNOVREMENSKA KOSIMULACIJA

Matematički modeli PMSM motora i regulatora za vektorsko upravljanje izvedeni u potpoglavljima 2.3 i 2.4 su modelirani u MATLAB/Simulinku prema MBD (eng. model-based design) metodi koja se sastoji od četiri glavna koraka:

1. izrada modela PMSM-a
2. analiziranje modela PMSM-a i sintetiziranje regulatora za brzinu vrtnje i struje motora
3. simuliranje PMSM-a i regulatora
4. puštanje regulatora u rad (ovaj korak nije odrađen u diplomskom radu jer se odnosi na stvarni regulator)

MBD metoda upotrebljava razne simulacije prilikom modeliranja sustava (PMSM-a) i regulatora za taj sustav. Osim simuliranja u Simulinku ova metoda obuhvaća još simulacije poput:

1. softver u petlji (eng. *software-in-the-loop*, *SIL*)
2. sustav u petlji (eng. *model-in-the-loop*, *MIL*)
3. hardver u petlji (eng. *hardware-in-the-loop*, *HIL*)

Ovaj diplomski rad upotrebljava dva različita simulatora (OPAL-ov OP5031 i dSpace-ov MicroLabBox o kojima će više biti riječi u potpoglavljima navest ref) i oni su povezani u stvarnovremenski kosimulacijski sustav u obliku HIL simulacije. Kosimulacija podrazumijeva paralelni rad dvaju povezanih simulatora s ciljem testiranja simulacijskog modela regulatora [16]. Izmjena signala se odvija u stvarnom vremenu (eng. *Real-Time*, *RT*), što znači da simulacijski model točno reproducira izlazne i reagira na ulazne veličine u isto vrijeme kad bi reagirao i stvarni sustav. Stvarnovremenske simulacije zbog uvjeta rada u stvarnom vremenu imaju određena ograničenja u svojim simulacijskim postavkama, poput diskretnog vremena i konstantan iznos vremenskog koraka (eng. *fixed-step size*) [17], jer stvarnovremenski simulator mora u zadanom vremenskom koraku odraditi matematičke proračune unutar zadanog vremenskog koraka [17, 18]. Postoje i stvarnovremenske simulacije koja imaju promjenjivi iznos koraka, ali ovaj rad se ograničava na konstantan iznos koraka. U HIL simulaciji su podsustavi implemetirani na različite simulatore povezani preko ulazno/izlaznih sučelja preko kojih se izmjenjuju signali. Jedina razlika između implementacije HIL-a u ovom radu i prave HIL simulacije je u regulatoru. U HIL simulaciji se primjenjuje stvarni regulator, dok se u radu primjenjuje model regulatora implementiran u stvarnovremenski simulator. Ova vrsta kosimulacije je najprije bila primjenjivana u zrakoplovnoj industriji, no u posljednje vrijeme se sve više primjenjuje u automobilskoj industriji, robotici, brodogradnji zbog svoje praktičnosti i mogućnosti ubrzanog razvoja te cijeni [19].

#### 3.1. OPAL-RT stvarnovremenski sustav

##### 3.1.1 OPAL OP5031 i OP5600 simulatori

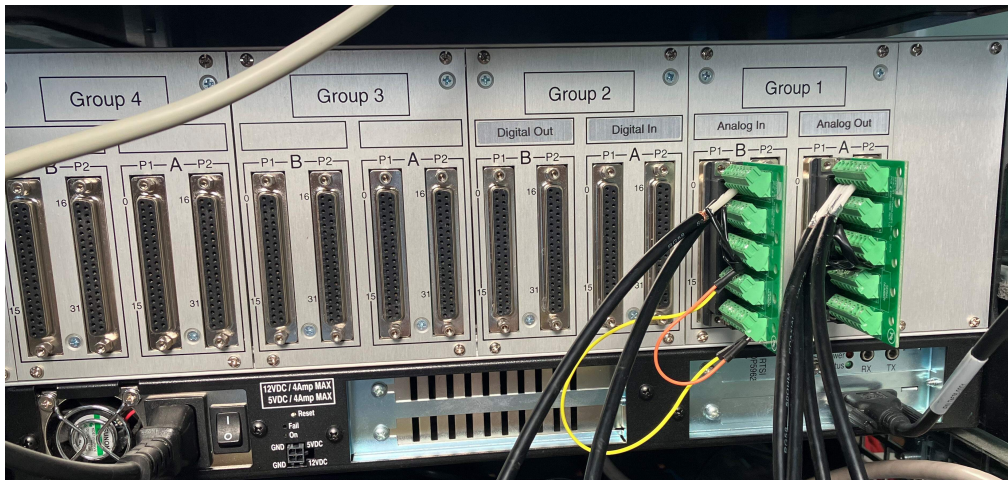
OPAL-RT-ov OP5031 je stvarnovremenski simulator. Ovakva vrsta simulatora se upotrebljava za stvarnovremensku simulaciju, kosimulaciju, komunikaciju i XIL (X-In-The-Loop)

simulacije. OP5031 je sustav koji sadrži FPGA (eng. *field-programable gate array*) kojeg je moguće reprogramirati po potrebi, za razliku od procesora čije se operacije ne mogu mijenjati te I/O (*input/output*) jedinice. OP5031 stvarnovremenski simulator je uparen s OP5600 simulacijskim sustavom koji omogućava HIL simulacije. OP5600 simulacijski sustav ima digitalne i analoge ulazno/izlazne priključke preko kojih se povezuje s drugom stvarnovremenskim simulatorima te je na taj način OP5600 svojevrsno proširenje OP5031 stvarnovremenskog simulatora. Slika 3.1 prikazuje oba OPAL simulatora (OP5031 je gornji dok je OP5600 donji simulator):



SL. 3.1: OP5031 i OP5600 simulatori

Povezivanje OP5600 stvarnovremenskog simulatora s drugim simulatorima se odrađuje pomoću DB37 konektora. OP5600 simulator se sastoji od 16 DB37 priključaka, podjeljenih u 4 grupe po 4 DB37 priključka. Unutar grupe su priključci podjeljeni u dvije podgrupe, A i B, svaki po dva DB37 priključka. Svaka podgrupa predstavlja ulazne ili izlazne signale. DB37 priključak može imati do 16 signala (pozitivni i negativni priključak svakog signala dok su ostali priključci ili nedostupni ili pod posebnim uvjetima dostupni). Ako se u podgrupi spajaju analogni signali tada podgrupa može maksimalno imati 16 signala dostupno (jedan DB37 priključak dok se drugi ne koristi), a u slučaju da se spajaju digitalni signali onda podgrupa može imati maksimalno 32 signala (upotrebljavaju se oba DB37 priključka), slika 3.2.



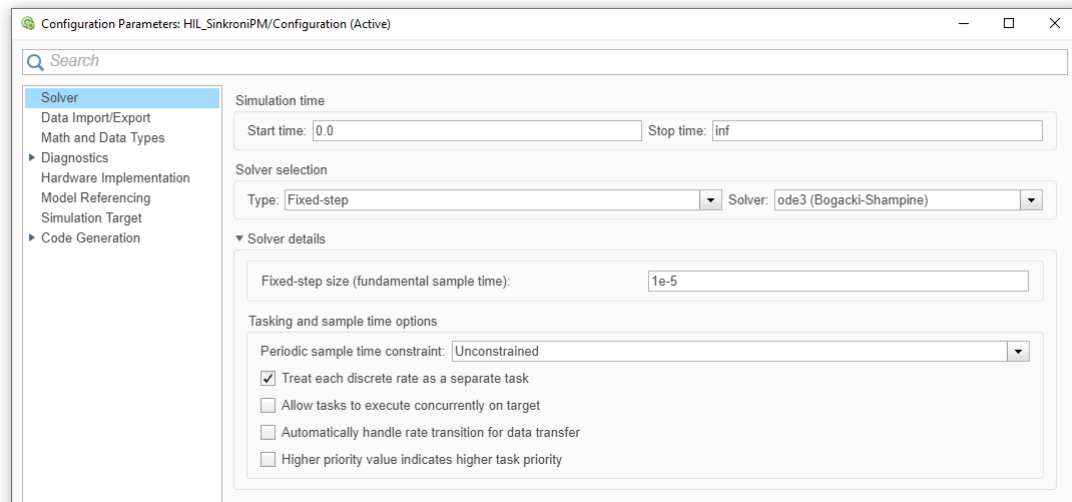
SL. 3.2: Rapodjela DB37 priključaka u OP5600 simulatoru

Način spajanja kabla na DB37 konektor je objašnjeno shematski i tablično u literaturi [20]. Ovakvo raspoređeni priključci se upotrebljavaju u simulacijske svrhe, točnije pomoću njih se šalju i dobivaju signali od drugih simulatora. Svi DB37 priključci, koji se nalaze na zadnjoj strani OP5600 simulatora su spojeni preko RJ45 kabla na ethernet priključke koji se nalaze s prednje strane OP5600 simulatora. Jedan DB37 priključak iz jedne podgrupe je spojen s 4 ethernet kanala (svaki kanal može primiti 4 signala od 16 koje dobije od DB37 priključka) tada se prespoji jedan od kanala na tzv. *monitoring connector* čiji priključci raspoređuju na 4 mini-BNC priključka preko kojih se spaja osciloskop (slika 3.1 prikazuje mini-BNC priključke u donjem desnom kutu, dok su ethernet priključci raspoređeni duž OP5600 simulatora). Ova procedura omogućuje mjerenje i poslanih i primljenih signala bez utjecaja na razmjenu signala između dva simulatora dok traje simulacija.

### 3.1.2 Priprema modela u Simulinku

Kako bi se moglo rukovati s OPAL-ovim simulatorima potrebno je na računalu imati RT-LAB program preko kojeg se odrađuju sve postavke i upravljanje za vrijeme simuliranja. RT-LAB program je povezan sa MATLAB/Simulinkom u kojem se uređuju i nadograđuju simulacijski modeli. MATLAB-u se pristupa iz RT-LAB-a jer su onda dostupni svi dodatni alati koje omogućuju simuliranje u OPAL-ovim simulatorima, između ostalog se proširuje biblioteka s blokovima za matematičke modele u Simulinku, grupe blokova poput: RT-EVENTS, RT-LAB I/O, RT-Drive i dr.. Otvaranjem Simulinka najprije se namještaju postavke simulacije, kako je ranije naglašeno diplomski rad se ograničava na stvarnovremensku kosimulaciju s konstantnim iznosom koraka, pa se tako i namjesti u postavkama, od *solvera* se odabere ode3(Bogacki-Shampine), moguće je odabrati i druge *solvere* pod uvjetom da je vremenski korak konstantan. Iznos koraka simulacije se odabere  $1e-5$  (0.00001) sekundi. U slučaju da se odabere duži korak od odabranog tada simulacija gubi na točnosti jer ima manje uzoraka (broj uzoraka je veći što je korak simulacije kraći) koji rekonstruiraju signale, a u slučaju da se odabere kraći vremenski korak onda se riskira da se proračuni unutar koraka neće provesti do kraja i ugroziti simulacija, točnije neće više slati podatke u stvarnom vremenu, slika 3.3





**SL. 3.3:** Postavke u Simulinku za OPAL simulator

Kako bi simulacijski model u Simulinku radio na RT-LAB-u za daljnje simuliranje potrebno je na glavnoj razini u Simulink-u (*root level*) stvoriti dva podsustava i samo oni trebaju biti na glavnoj razini sve ostalo od blokova treba biti u jednom od dva podsustava. Prvi podsustav je tzv. SM podsustav (eng. *subsystem master*, *SM*), u ovaj podsustav se implementira simulacijski model koji će se simulirati na OP5031 simulatoru. U ovaj podsustav se implementiraju ulazno/izlazna sučelja koja šalju/primaju signale od drugih simulatora/izvora za vrijeme simuliranja. Drugi podsustav je tzv. SC podsustav (eng. *subsystem console*, *SC*), ovaj podsustav se izvršava na računalu za vrijeme simulacije i predstavlja svojevrsno sučelje za korisnika.

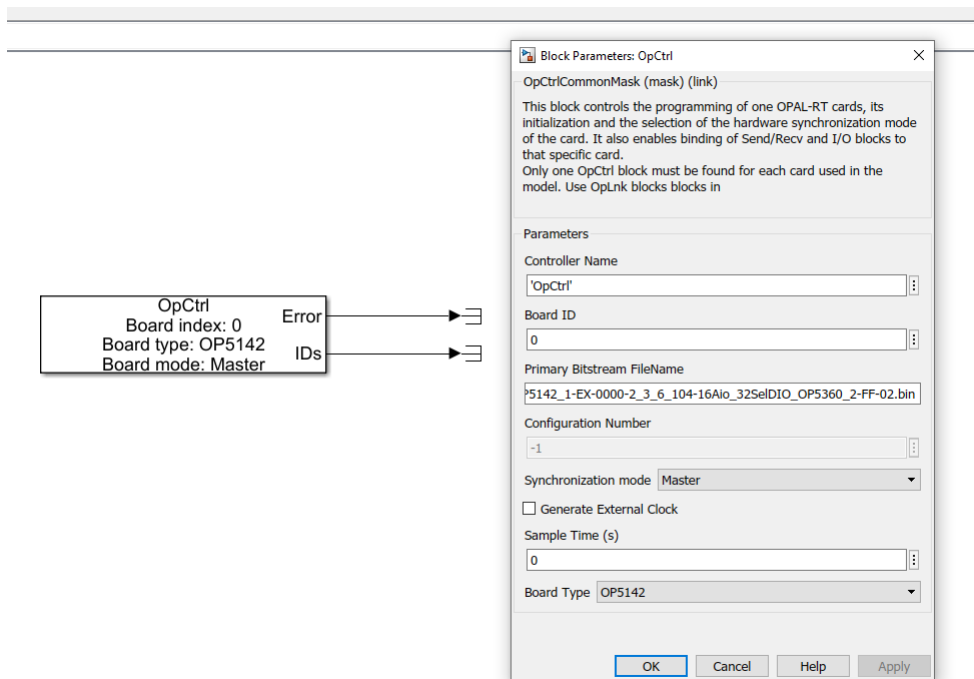
Oba podsustava moraju imati OpComm blok koji simulira stvarnovremensku komunikacijsku poveznicu. Svaki signal koji ulazi u podsustav iz drugog podsustava mora proći kroz OpComm blok prije nego se proslijedi dalje modelu. Signali koji prolaze su po vrsti podataka tipa *double*. Otvaranjem postavki OpComm bloka se unosi broj ulaznih signala.

Početne simulacije na OP5031 stvarnovremenskom simulatoru se odrade tako što se na SM podsustav implementira matematički model PMSM-a, dok se na SC implementira matematički model regulatora kako bi se utvrdila ispravnost matematičkog modela motora (kasnije se simulacijski model regulatora implementira na dSpace-ov MicroLabBox, ali o tome će više biti rečeno u poglavlju 4.). Implementiranjem matematičkih modela PMSM-a i regulatora odrade se *offline* simuliranje kako bi se utvrdilo ima li matematički model grešaka.

### 3.1.3 Komunikacija i skaliranje signala

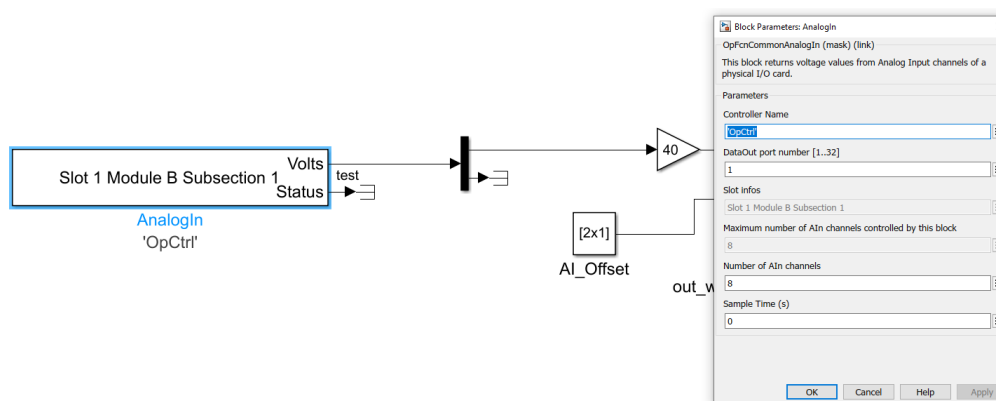
Utvrđivanjem ispravnosti se prelazi na implementiranje komunikacijskih blokova kako bi se mogla realizirati kosimulacija, točnije izmjena signala između dva stvarnovremenska simulatora. Komunikacijski blokovi se dijele na izlazne i ulazne, a po vrsti se dijele na: analogne, digitalne, PWM, enkoder, i dr.. Ovaj rad upotrebljava analogne ulazne i izlazne komunikacijske blokove u simulacijskom modelu i oni će se detaljnije objasniti, dok se kod ostalih vrste komunikacijskih blokova neće ulaziti u detaljniji opis.

Ako simulacijski model upotrebljava komunikacijske blokove, odnosno ako šalje i prima signale, onda se mora parametrirati blok *OpCtrl* u kojem se povezuju ulazno/izlazni blokovi sa ulazno/izlaznim sučeljima preko kojih izmjenjuju signali tijekom izvođenja kosimulacije i dodjeljuje mu se ime (*Controller name*) koje kasnije povezuje ulazne i izlazne blokove s komunikacijskim sučeljima koji se upotrebljavaju, slika 3.4



SL. 3.4: *OpCtrl* blok s postavkama

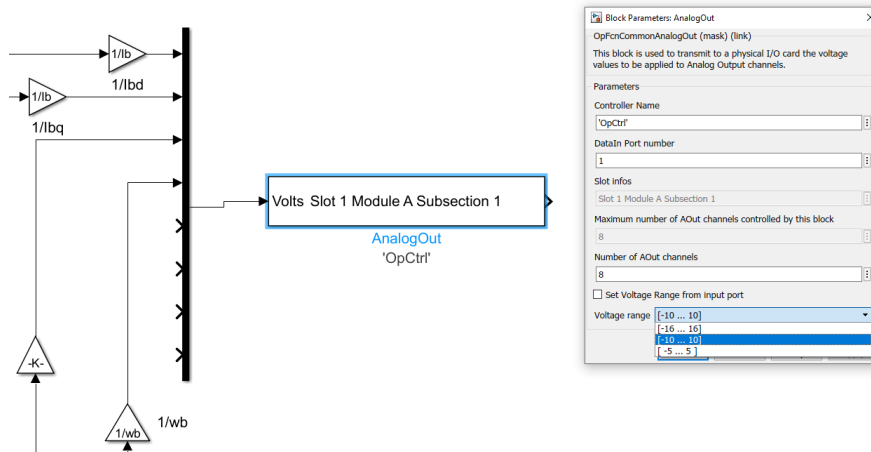
Primanje signala od drugih simulatora/izvora i prosljeđivanje dalje u simulacijski model u Simulinku se postiže blokom *AnalogIn*. Kako bi se signal primljen prosljedio simulacijskom modelu potrebno je u postavkama bloka pod *Controller Name* napisati isto ime koje je napisao u *OpCtrl* bloku, a pod *DataOut port number* se odabere broj (1 ili 2) koji predstavlja niz priključaka u DB37 priključku. Tako broj 1 predstavlja priključke (kanale) od 0 do 7, a broj 2 priključke od 8 do 15. Također je u postavkama bloka ispisana informacija koji DB37 priključak se koristi tako što je navedena grupa, podgrupa i priključak, ova informacija se ne može uređivati u tom bloku već ovisi o postavkama u *OpCtrl* bloku, gdje se odabire sučelje. Budući da *AnalogIn* blok prima 8 signala, te signale je potrebno razdvojiti kako bi svaki signal mogao biti prosljeđen gdje je potreban u matematičkm modelu, to se postiže upotrebom Simulink-ovog bloka *demux* koji *n*-dimenzijski signal razdvaja na *n* zasebnih signala. Slika 3.5 prikazuje *AnalogIn* blok sa postavkama, te prikaz skaliranja signala o čemu će više biti rečeno kasnije u tekstu.



SL. 3.5: *AnalogIn* blok s postavkama

Slanje podataka iz simulacijskog modela prema drugim simulatorima/mjernim uređajima

se postiže upotrebom bloka *AnalogOut*. Slično kao kod *AnalogIn* bloka, potrebno je u postavkama bloka *AnalogOut* napisati isto ime u *Controller Name* kao u *OpCtrl* kako bi se blok povezao s izlaznim sučeljem koje je određeno u *OpCtrl* i pod *DataIn port number* napisati 1 ili 2 ovisno o grupi priključaka koji će se koristiti. Potrebno je još namjestiti naponsku razinu, moguće je odabrati:  $\pm 16$  V,  $\pm 10$  V ili  $\pm 5$  V. Za Kosimulaciju u diplomskom radu je odabrano  $\pm 10$  V zbog dSpace-ovog MicroLabBox simulatora koji je namještan na  $\pm 10$  V, kako bi se olakšalo preračunavanje prilikom skaliranja signala. Odabrom grupe priključaka potrebno je bloku *AnalogOut* proslijediti 8 signala, što se postiže Simulink-ovim blokom *mux* koji povezuje  $n$  zasebnih signala u  $n$ -dimenzionalni signal, slika 3.6.



SL. 3.6: *AnalogOut* blok s postavkama

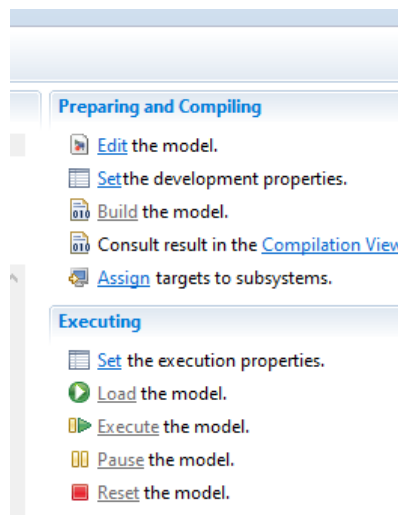
Prije prosljeđivanja *AnalogOut* bloku signale je, iz sigurnosnih razloga i zbog ograničenja OP5031 i OP5600 simulatora, potrebno skalirati kako bi napon bio unutar prethodno odabranih granica ( $\pm 10$  V). U opisu bloka *AnalogOut* stoji kako signal vrijednosti 10 u Simulinku je 10 V na priključku. Zato je potrebno signale čije su vrijednosti u Simulinku veće od 10 skalirati tako da se vrijednosti koja se šalje *AnalogOut* bloku bude u intervalu vrijednosti od -10 do 10. Tako se kutna brzina rotora stroja čija nazivna vrijednost iznosi 314 rad/s se skalira (podijeli) s brojem 35 upotrebom Simulink bloka *gain* i tako osigura da iznos signala koji predstavlja brzinu neće preći 10, a time i napon neće biti veći od 10 V. Signali koji se šalju OPAL-ovom simulatoru su u naponskom intervalu od -16 V do 16 V. Signali koje će OP5031 stvarnovremenski simulator primiti će biti poslani od dSpace-ovog MicroLabBox-a čiji je naponski interval od -10 V do 10 V pa će i skaliranje biti namješteno po tom naponskom intervalu. Budući da će na OP5031 stvarnovremenskom simulatoru u kosimulaciji biti implementiran matematički model PMSM-a koji će signale koji će predstavljati napon primiti od dSpace-ovog MicroLabBox-a, skaliranje će se objasniti na primjeru skaliranja napona. Primiti signali napona se najprije skaliraju (pomnože) pomoću bloka *gain*, no kako je na dSpace-ovom MicroLabBox-u vrijednost 1 u Simulinku ekvivalentna 10 V na priključnici, napon je skaliran s 400 V. Kako je već naglašeno 10 V na priključnici na OP5031 simulatoru je ekvivalentno vrijednosti 10 u Simulinku pa se iz tog razloga napon skalira sa 40 V, a ne 400 V.

### 3.1.4 Kompajliranje matematičkog modela

Nakon što se *offline* simulacijama utvrdi da je sve uredi sa matematičkim modelom i kad se uklone sve pogreške koje se pojave potrebno je vrijeme simulacije u postavkama namjestiti

na *inf* umjesto nekog broja, iz razloga što će pokretanjem simulacije u OP5031 simulatoru se simulacija izvršavati predviđeno vrijeme i potom ugaziti. Nakon što se implementira matematički model u Simulink, prelazi se na RT-Lab program odakle se dalje matematički model iz Simulinka priprema za stvarnovremensku kosimulaciju uz brojne mogućnosti poput uređenja HMI sučelja, snimanja podataka za rekonstrukciju signala, i dr..

Kako bi simulacijski model iz Simulinka mogao biti implementiran na OP5031 stvarnovremenski simulator potrebno ga je prevesti u C kod pomoću naredbe *Build*. Izrada C koda simulacijskog modela traje određeno vrijeme. U slučaju pojave pogreške, *Build* prestaje i ispisuje se pogreška, moguće pogreške su: *solver* nije konstantnog vremenskog koraka ili nije napisan iznos vremenskog koraka (u postavkama u Simulinku je iznos vremenskog koraka namješteno na *auto* odnosno program sam prilagođava iznos), ako komunikacijski blokovi nisu parametrirani pravilno i sl. Kad je *build* završen simulacijski model se može učitati na OP5031 stvarnovremenski simulator preko naredbe *load*. Nakon učitavanja, model se pokreće naredbom *execute*. Dok se simulacijski model izvršava moguće je model pauzirati naredbom *pause*, a ponovnim pokretanjem simulacija nastavlja gdje je stala (vrijednosti signala/varijabli su nepromjenjene), s druge strane pokretanjem naredbe *reset* se simuliranje zaustavlja, a vrijednosti signala/varijabli se prilikom ponovnog pokretanja vraćaju na početne vrijednosti, slika 3.7.



**SL. 3.7:** Naredbe za manipuliranje modelom u RT-Lab programu

### 3.1.5 Kontrola simulacije u stvarnom vremenu

RT-Lab program omogućuje praćenje i manipuliranje varijablama simulacijskog modela za vrijeme izvođenja simulacije putem HMI sučelja (*human machine interface, HMI*). HMI sučelje se uređuje preko LabVIEW programa. LabVIEW-u se kao i Simulinku pristupa preko RT-Lab-a, gdje se odabere verzija LabVIEW-a s kojom je RT-Lab povezan. HMI sučelje obuhvaća upotrebu grafova, numeričkih ispisa vrijednosti, numerički unos vrijednosti, tipke, lampice za signaliziranje i sl.. U RT-Lab sučelju pod *Project Explorer* izlistani svi važni dijelovi simulacijskog modela: blokovi upotrebljeni u Simulink-u, HMI sučelja za praćenje, snimanje podataka. Podjela blokova je po podsustavima u Simulinku, na glavnoj razini su izlistani SC i SM podsustavi, a unutar njih su izlistani blokovi i podsustavi koje sadrže. Važno je pravilno imenovati blokove i signale u Simulinku radi lakšeg snalaženja, a i iz razloga što ako se ne dodjeli ime bloku, dodjeli mu se automatsko ime koje generira Simulink.

Kako bi element HMI sučelja (npr. graf) mogao prikazivati vrijednosti (u slučaju grafa



valni oblik) neke varijable, potrebno je varijablu dodijeliti elementu na način da se varijabla "postavi" preko elementa sučelja (eng. *drag and drop*). Ovisno o vrsti elementa sučelja, za elemente koji ispisuju vrijednosti se mogu upotrijebiti signali koji izlaze iz blokova, dok kod elemenata za unos vrijednosti je potreban iznos koju blok ima (u ispisu varijable je navedeno pod *value*), npr. ako je potrebno manipulirati iznosom referentne vrijednosti kutne brzine rotora motora koja je postavljena u obliku bloka *constant* u Simulinku onda se ta vrijednost postavi u element HMI sučelja. Dodatne mogućnosti oko uređenja HMI sučelja su: promjena boja kako bi se postigao bolji kontrast, povećanje fonta brojeva i slova, uređivanje koordinatnih osi grafova. Pokretanje HMI sučelja je preko naredbe *run* unutar LabVIEW-a, kao i zaustavljanje, naredbom *stop*.

### 3.1.6 Snimanje podataka u RT-Labu

Pomoću RT-Lab-a je moguće i snimanje podataka varijabli simulacije radi kasnije rekonstrukcije grafova te je odrađeno pomoću *Recorder*-a. Unutar funkcije *recordera* se postavljaju pojedini *recorderi* te otvaranjem postavki svakog pojedinačno se otvara sučelje unutar kojeg se dodaju signali koji se žele snimiti. Signali se dodaju sa popisa Simulink blokova, slično kao za elemente HMI sučelja.

Definiraju se uvjeti pod kojima će se početi snimati uzorci, tzv. *trigger* koji se dodjeljuje nekom signalu koji se snima. Odabere se iznos pri kojem će snimanje početi (*trigger level*) i vrsta *triggera* koji može biti *edge* ili *level*, gdje *edge* snima nakon što *trigger* signal prijeđe zadani iznos, a *level* snima dok je *trigger* signal iznad zadane razine. Osim vrste *triggera* se određuje i polaritet koji može biti pozitivan ili negativan, gdje negativan kreće snimati kad je iznos *trigger* signala ispod zadana razine. Također važna veličina je vrijeme snimanja (u sekundama, milisekundama, minutama,...) i broj kadrova koji se snimaju (*frame number*), gdje broj kadrova predstavlja broj uzastopnih koraka bez gubitaka informacija.

Za primjer, ako se vrijeme snimanja stavi na 10 sekundi i na 2 kadra onda se dobiju dva uzastopna niza podataka gdje svaki traje 5 sekundi, no prilikom prelaska uz jednog kadra u drugi se riskira gubljenje podataka. Kako bi se izbjeglo gubljenje podataka broj kadrova se stavi na 1 i onda je sve vrijeme snimanja u jednom kadru. Opcijom *pre-trigger percentage* se odabire koliko dugo (postotak od vremena snimanja) unutar kadra će podataka biti snimljeno prije nego *trigger* varijabla pređe zadani iznos. Snimljeni podaci mogu biti spremljeni u OPREC, csv ili MAT formatu.

## 3.2. dSpace stvarnovremenski sustav

### 3.2.1 MicroLabBox simulator

DSpace-ov MicroLabBox je stvarnovremenski simulacijski sustav predviđen za laboratorijsku upotrebu, za istraživanje i razvoj.

Sastoji se od FPGA (*field programable gate array*) i I/O (*input/output*) jedinice za simulaciju i sadrži više od 100 različitih ulazno/izlaznih priključnica. MicroLabBox dolazi u tri različite varijante: BNC, Sub-D i tzv. *spring cage*. U ovom diplomskom radu je upotrebljena BNC varijanta MicroLabBox-a. BNC varijanta MicroLabBox-a se sastoji od sljedećih priključaka[21]:

1. 32 ulazna analogna BNC priključka, preko kojih MicroLabBox dobiva signale od drugih simulatora, izvora i sl.

2. 16 izlaznih analognih BNC priključaka, preko kojih MicroLabBox šalje signale prema drugim simulatorima, osciloskopu i sl.
3. dva 50-pinska digitalna ulazno/izlazna priključka
4. četiri 9-pinska digitalna ulazno/izlazna priključka (CAN priključak, 2 resolver priključka i RS232 (422/485) priključak)

Navedeni priključci služe za simulacijske svrhe, dodatno još sadrži priključke za spajanje s računalom: dva ethernet priključka, host PC priključak (također ethernet), dva USB priključka i priključke za senzore koji se mogu upotrebljavati u simulaciji. Također se na gornjoj strani uređaja nalazi pet LED lampica, gdje četiri svijetle kako bi se informiralo korisnika o određenom statusu vezano uz simulaciju i jedna LED lampica koja svijetli kad se MicroLabBox uključi. Kako je već navedeno, MicroLabBox se upotrebljava za istraživanje i razvoj u brojnim područjima od kojih su neka: upravljanje električnim strojevima, robotika, testiranje medicinskih uređaja. Slika 3.8 prikazuje MicroLabBox.



**SL. 3.8:** dSpace-ov MicroLabBox stvarnovremenski simulator

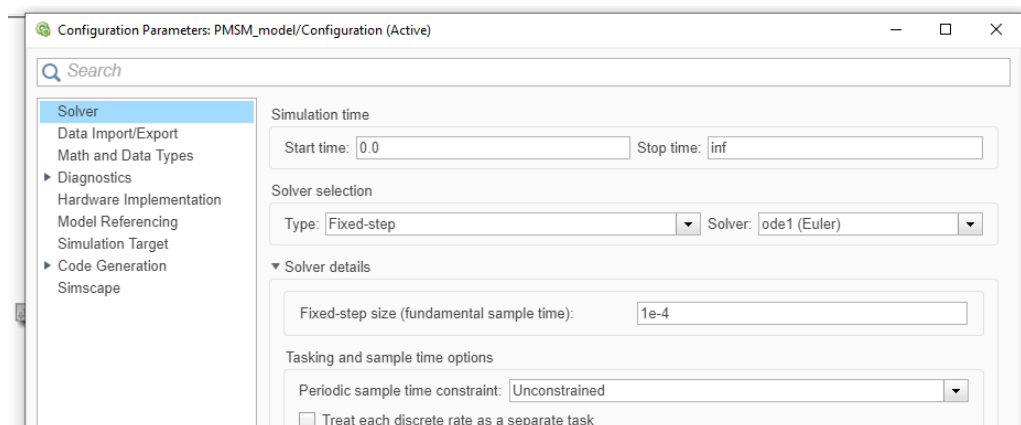
### 3.2.2 Povezivanje MicroLabBoxa s računalom

MicroLabBox se sa računalom povezuje preko IP adrese pomoću RJ45 kabla koji se spoji u host PC priključak, a u računalu u ethernet priključak, odabrana je network konekcija jer su računal i MicroLabBox spojeni na istu mrežu, a povezuju se preko switcha (druga opcija povezivanja je peer-to-peer konekcija koja direktno povezuje MicroLabBox i računalu). Kako bi se MicroLabBox mogao koristiti potrebno je osim valjane licence imati instalirane određene programe na računalu: MATLAB/Simulink programski paket, dSpace-ov Control Desk (uz Control Desk se dobije još programa, ali oni se nisu upotrebljavali u diplomskom radu). Nakon instaliranja programa potrebno je pomoću aktivne licence dekrptirati sve pakete za upotrebu (to uključuje: programske pakete, biblioteke, testne sustave i sl.). Uz dekrptiranje, potrebno je odabrati i MATLAB kojeg se želi integrirati u dSpace-ove programe, pomoću kojeg će se modelirati modeli za simuliranje u MicroLabBox-u, tako su se za potrebe ovog diplomskog softverski paketi povezali sa 2020b verzijom MATLAB-a. Povezivanje se radi kako bi se biblioteke dSpace-a koristiti u Simulinku.

### 3.2.3 Priprema matematičkog modela u Simulinku

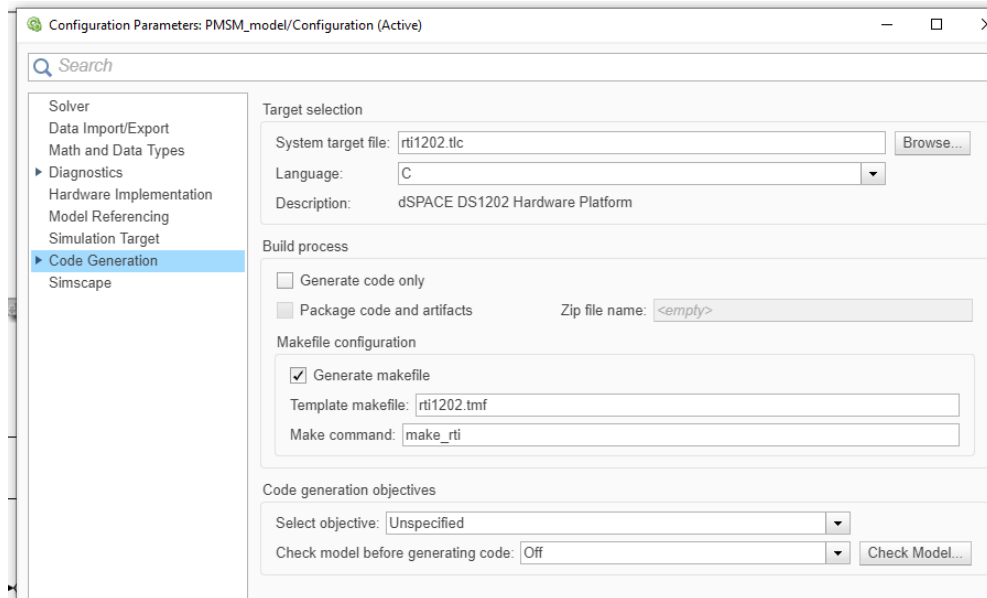
Pokretanjem MATLAB programa potrebno je odabrati stvarnovremensko sučelje (eng. *real time interface, RTI*), svaki dSpace-ov stvarnovremenski simulator ima dodijeljeni RTI pa je tako RTI za MicroLabBox RTI1202. RTI predstavlja poveznicu između MATLAB-a i dSpace-ovog hardvera te proširuje opcije Simulink Coder-a za implementiranje Simulink modela na stvarnovremenski dSpace-ov hardver.

Prilikom pokretanja Simulinka odabere se RTI1202 *template* kako bi se postavili uvjeti za razvoj modela za stvarnovremensku simulaciju. Prije izrade samog simulacijskog modela potrebno je namjestiti određene postavke simulacije, u *Configuration Parameters*. Najprije je potrebno odabrati *solver*. Kako je ranije napomenuto stvarnovremenska kosimulacija u diplomskom radu se ograničava na konstantan vremenski korak simulacije (*fixed-step*), pa je ta vrsta odabrana u Simulink postavkama, a iznos koraka je  $1e-4$  (0.0001) sekundi. Duži vremenski korak od odabranog iako brže odradi proračun ne daje precizna rješenja (na ukupnom vremenu simulacije bi bilo manje uzoraka pa su zato varijable/signalni manje precizni), dok je kraći vremenski korak precizniji, ali u slučaju zahtjevnijeg modela zahtjeva više vremena za proračun što može ugroziti stvarnovremensku simulaciju. *Solver* postavljen na ode1 (Euler) zbog brzog proračuna (moguća je upotreba i drugih *solvera*, ali pod uvjetom da je vremenski korak konstantan), slika 3.9.



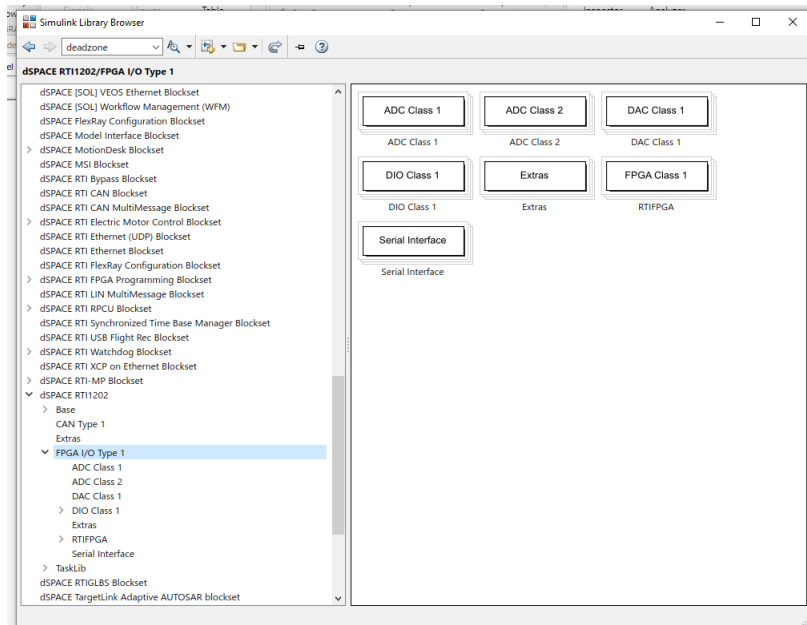
SL. 3.9: Postavke simulacijskog modela u Simulinku

Od dodatnih postavki, u *code generation* se odabire na koji će se hardver model iz Simulinka učitati, postavljeno je na ds1202. Namješta se programski jezik u kojeg će simulacijski model biti zapisan, moguće je odabrati programski jezik C ili C++, odabran je jezik C. Osim jezika odabire se tzv. *.tlc* (eng. *target language compiler, TLC*) datoteka, koja propisuje koji se softver koristi za generiranje C koda simulacijskog modela te je postavljeno na *rti1202.tlc* datoteku. Uz TLC se odabire i *makefile* datoteka koja određuje kompajler koji se upotrebljava za vrijeme generiranja C koda simulacijskog modela, odabrana je *rti1202.tmf* datoteka, slika 3.10. Navedene datoteke se mogu koristiti kad je odabran potrebni dSpace-ov RTI i dostupni su samo dok je dostupna aktivna licenca. Još je potrebno u *simulation target* u postavkama odznačiti *block reduction*, jer iako ubrzava proračun (ne odrađuje određene konverzije, ako signal ne vodi do nekog bloka za njega nema proračun i sl.) nije podržan za MicroLabBox-ov RTI.



**SL. 3.10:** Postavke programskog jezika za prijevod simulacijskog modela

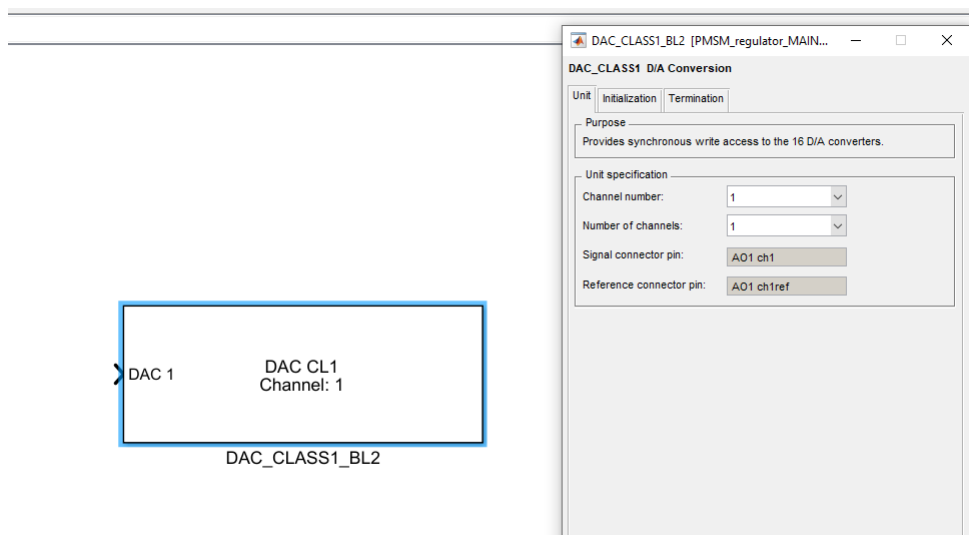
Na radnoj površini u Simulinku se prilikom otvaranja RTI1202 *template-a* pojavljuje RTI Data blok. Ovaj blok se upotrebljava kako bi se spremali interni podaci o modelu i mora biti na glavnoj razini (*root level*) Simulink modela i ne smije se brisati jer bi to moglo uzrokovati gubljenje podataka. U Simulinku se najprije implementira matematički model (najprije se implementirao matematički model PMSM-a za potrebe validacije MicroLabBox-a, o čemu više detalja ima kasnije u tekstu, a potom se implementirao matematički model regulatora za potrebe kosimulacije, i o ovome će također biti rečeno kasnije u tekstu). Nakon što se implementirao model slijedi simuliranje modela u Simulinku, tzv. *offline* simulacije, *offline* jer vrijeme dostave rezultata nije od presudne važnosti. Ovakve simulacije se rade kako bi se provjerila ispravnost matematičkog modela. Prilikom modeliranja u Simulinkovom *library-u* se osim standardnih blokova poput: pojačanja, integratora, sumatora i sl. mogu pronaći i dSpace-ovi blokovi. Ti blokovi su podijeljeni u kategorije, ovisno o svrsi koju imaju, pa tako postoje: blokovi koji predstavljaju modele električnih motora, CAN blokove, blokove za implementaciju senzora, komunikacijske blokove i druge, slika 3.11. U ovom radu je najvažnija bila grupa blokova za komunikaciju zbog izmjene signala između MicroLabBox-a i OP5031. Komunikacijski blokovi se kao i priključci na MicroLabBox-u dijele na analogne i digitalne.



SL. 3.11: Postavke programskog jezika za prijevod simulacijskog modela

### 3.2.4 Komunikacijski blokovi

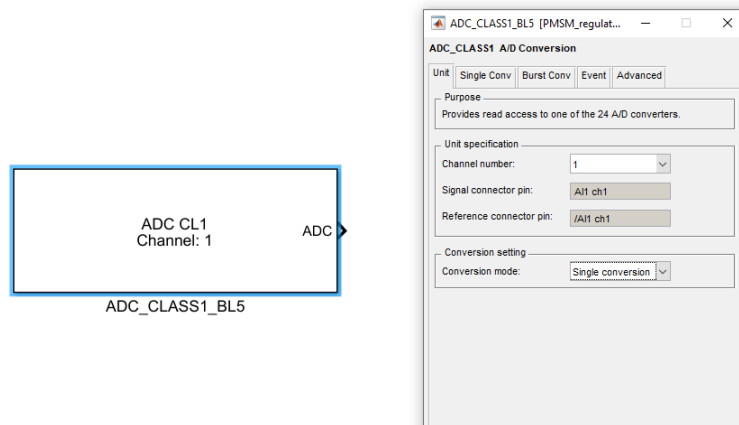
Ako određeni signal treba proslijediti na izlazne priključke MicroLabBox-a kako bi se mjerio signal na osciloskopu ili kako bi se poslao drugom simulatoru potrebna je upotreba DAC bloka (eng. *digital-analog converter*, *DAC*). Na DAC blok se spajaju signali koji su po tipu podataka double iako se mogu slati i signali drugih tipova podataka (npr. int, boolean i sl.). Nakon što se odabrani signal spoji na DAC blok, otvore se postavke bloka u kojem se odabire kanal (priključak) na kojeg signal treba proslijediti (*channel number*). Izlaznih priključaka ima 16 pa se odabire broj od 1 do 16, također je moguće grupirati signale pa se na isti DAC blok šalje više signala (*number of channels*), ali se pri tome odabire niz priključaka na koje će se signali proslijediti. Slika 3.12 prikazuje DAC blok sa prethodno opisanim postavkama.



SL. 3.12: DAC blok s postavkama

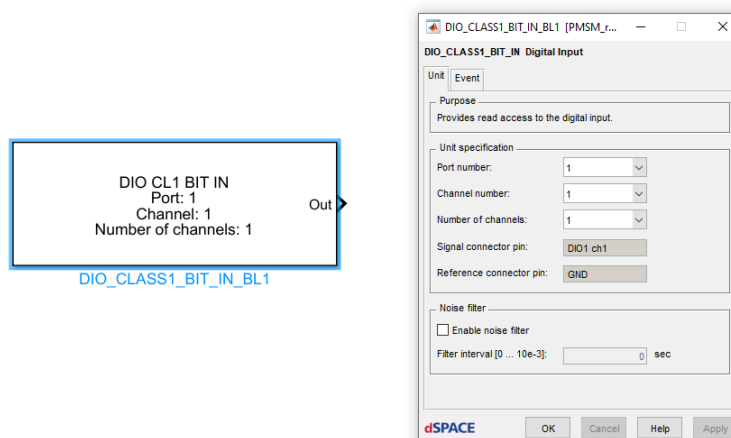
Signali koje simulirani model prima od izvora/drugih simulatora je potrebno uvesti

u model preko ADC bloka (*analog-digital converter*). Signal koji izlazi iz ADC bloka u Simulinku je po tipu podatka double. Kao i kod DAC bloka, otvaranjem postavki bloka potrebno je odabrati kanal (priključak) preko kojeg se prima signal. Ulaznih analognih priključaka ima 32 podijeljenih u dvije grupe: *class1* koji sadrži 24 priključka i *class2* koji sadrži 8 priključaka, razlika između ovih priključaka je u broju uzoraka koje snimaju gdje *class2* vrsta priključka snima 10 puta više uzoraka. Prilikom odabira DAC bloka odabire se i *class*, a u postavkama bloka kanal na kojeg će se signal spojiti i proslijediti dalje simuliranom modelu, slika 3.13



SL. 3.13: ADC blok s postavkama

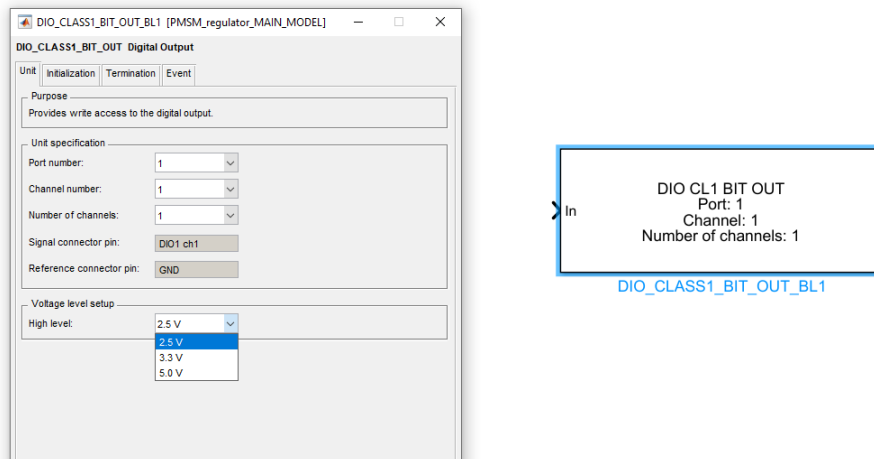
U slučaju da se za potrebe simulacije trebaju koristiti digitalni priključci, u Simulinku se implementiraju DIO BIT IN/OUT blokovi (eng. *digital input/output, DIO*), ovi blokovi primaju i šalju signale koji su boolean tip podatka pa mogu imati dva stanja 0 ili 1. Za slučaj da MicroLabBox dobiva digitalni signal na 50-pinski (ili 9-pinski) digitalni priključak tada se u Simulink model postavi DIO BIT IN blok, te se otvaranjem postavki bloka odabire: PORT *number* kojih ima tri, a unutar jednog PORT-a ima 16 kanala i potrebno je odabrati kanal koji će se koristiti, slika 3.14. Dok je kod BNC priključka referentna točka na plaštu priključka i svi BNC priključci ima svoje referentne točke koje su međusobno izolirane, kod digitalnog 50-pinskog (i kod 9-pinskog) priključka je za sve kanale referentna točka jednaka te se zato prilikom odabira kanala odabire samo priključak bez referentne točke.



SL. 3.14: DIO IN blok s postavkama

Procedura je identična za slanje digitalnih signala iz Simulink modela na digitalne

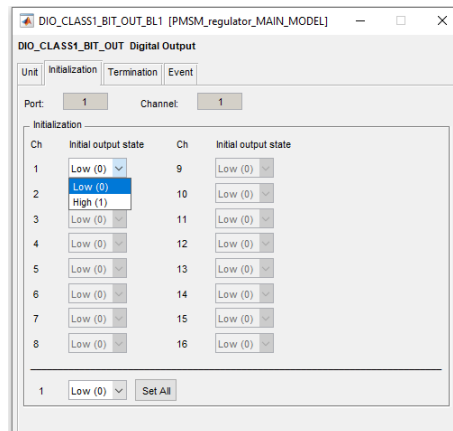
priključke. U Simulinku se upotrebljava blok DIO BIT OUT i u postavkama bloka se odabire PORT *number*, broj kanala unutar PORT-a te naponska razina koja može biti: 2.5 V, 3.3 V ili 5 V, slika 3.15



SL. 3.15: DIO OUT blok s osnovnim postavkama

Osim postavki kanala na kojeg će se slati signali i naponske razine potrebno je namjestiti još dvije važne postavke:

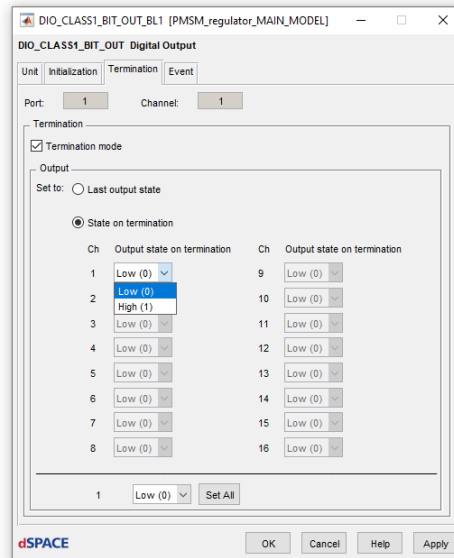
1. *initial output state* gdje se odabire na kojoj će naponskoj razini biti priključak u na početku: *High* (1) ili *Low* (0), gdje je 1 znači da na priključku ima napona, a 0 da nema napona. u slučaju da se ova postavka ne namjesti, postavljeno je na stanje *Low*.



SL. 3.16: Postavke početnog uvjeta DIO OUT bloka

2. *termination* gdje se odabire naponska razina na kojoj će ostati priključak nakon što se simulacija prekine. Postavkom kvačice na *termination mode* postoje dvije opcije uređivanja: zadržati posljednje stanje koje priključak ima prilikom zaustavljanja simulacije (*Last output state*) ili ručno namještanje gdje se odabire *High* ili *Low*. U slučaju da se ne stavi kvačica na *termination mode*, signal se vraća u stanje *Low*.





SL. 3.17: *termination* postavke DIO OUT bloka

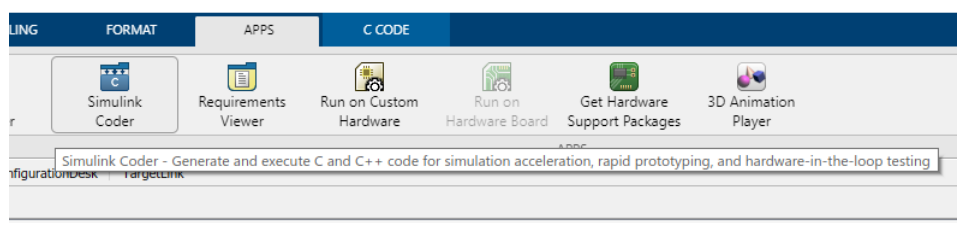
### 3.2.5 Skaliranje signala

Zbog ograničenih mogućnosti simulacijskog sustava i zbog sigurnosti osobe koja upravlja simulatorom napon je maksimalnog iznosa od 10 V. Iz tog razloga je potrebno signale koji će se proslijediti priključnicama skalirati kako ne bi prešli iznos od 10 V i tako oštetili simulator ili ugrozili sigurnost rukovatelja simulatorom. Kako je već ranije naglašeno prilikom namještanja signala koji se šalju na digitalne priključke odabire se naponska razina koja je manja od 10 V te nije potrebno skalirati jer signal je ili na 0 V (u slučaju logičke 0) ili odabrani napon (u slučaju logičkog 1) koji je manji od 10 V. Kod analognih signala signal u Simulinku nije ograničen na 0 ili 1 već može postići bilo koju vrijednost, a kod MicroLabBox-a za analogne signale vrijedi pravilo: signal vrijednosti 1 u Simulinku je 10 V na analognom priključku, odnosno MicroLabBox pojačava svaki signal 10 puta. Kako bi se signali koji se šalju iz Simulinka na analogne priključke mogli poslati bez da izgube svojstva (npr. valni oblik) potrebno ih je skalirati prije nego se proslijede DAC bloku. Najčešći način skaliranja je pomoću *gain* bloka koji se postavi prije ADC bloka u Simulinku sa svrhom spuštanja signala na interval vrijednosti od -1 do 1 kako bi napon na priključku bio od -10 V do 10 V. Kako će se koji signal skalirati ovisi o tome koju fizikalnu veličinu predstavlja, za primjer će se uzeti kutna brzina rotora motora mjerena u rad/s. Ako je nazivna kutna brzina rotora motora 314 rad/s i ako se ta kutna brzina za potrebe simulacije mora proslijediti na jedan od analognih priključaka mora se skalirati i to brojem koji je veći od 314, npr. 350 te se tako se osigura da će napon signala uvijek biti ispod 10 V jer brzina nikad neće prelaziti 314 rad/s (osim prilikom uključanja tereta, ali i tada ne prelazi više od 320 rad/s). Procedura skaliranja je jednaka za sve signale koji se šalju na neki od analognih priključaka. Za slučaj da se signal šalje drugom simulatoru gdje će utjecati na prilike u modelu potrebno je rekonstruirati signal skaliranjem istim iznosom s kojim je skaliran prije slanja na analogni priključak. U slučaju da MicroLabBox prima signal na jedan od svojih analognih ulaza, dodatno se skalira s 10 te se napon od 10 V svodi na napon od 1 V. Procedura rekonstruiranja signala u Simulink modelu se sastoji od upotrebe *gain* bloka spojenog s ADC blokom kako bi se primljeni signal pomnožio s istom vrijednošću s kojom je skaliran na drugom simulatoru te dodatno pomnožen s 10 kako bi se poništio utjecaj MicroLabBox-a.



### 3.2.6 Kompajliranje matematičkog modela

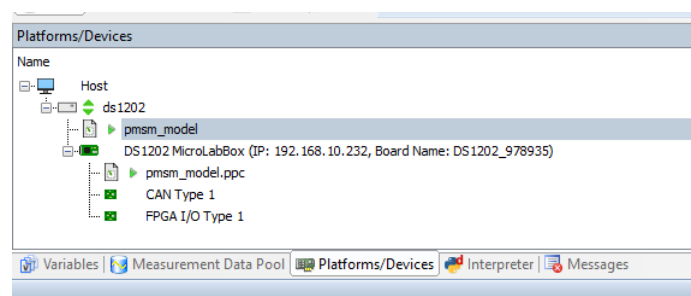
Namještanjem simulacijskog modela, skaliranjem i namještanjem komunikacijskih blokova potrebno je simulacijski model prevesti u jezik C kako bi se mogao pokrenuti na Control Desku. Također je potrebno promijeniti simulacijsko vrijeme na inf (beskonačno) jer u protivnom ako se ostavi vrijeme u sekundama onda se pokretanjem simulacije na MicroLabBox-u simulacija izvršava određeno vrijeme, nakon čega se zaustavlja. Kako bi se to izbjeglo te kako bi se zaustavljanje simulacije moglo izvesti ručno potrebno je vrijeme simulacije staviti na inf (prikazano na slici 3.9). Prevođenje simulacijskog modela u C kod se odrađuje pomoću Simulink-ovog alata Simulink Coder, ovaj proces se još naziva *build*.



SL. 3.18: Simulink Coder

### 3.2.7 Control Desk

Prevođenjem modela u C kod, stvaraju se datoteke od kojih je navažnija .sdf datoteka koja se otvara u Control Desk-u za daljnje simuliranje. Stvaranjem novog projekta u dSpace-ovom Control Desku potrebno je odabrati prethodno izgeneriranu sdf datoteku kako bi se učitao simulacijski model na MicroLabBox. Pokretanjem Control Deska potrebno je registrirati platformu (MicroLabBox je registriran kao ds1202) i uspostaviti komunikaciju s MicroLabBox-om. Jednom kad je uspostavljena komunikacija između Control Deska i MicroLabBox-a u popisu platformi je ispisana registrirana platforma kako prikazuje slika 3.19.



SL. 3.19: Registrirana platforma

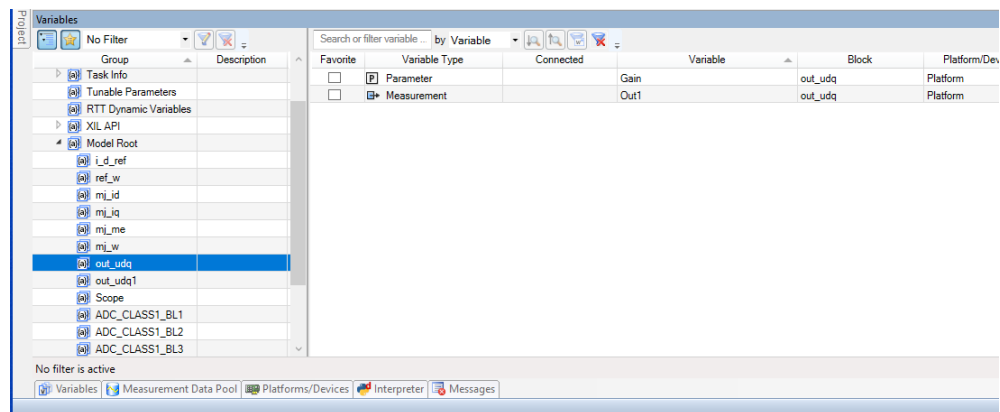
Control Desk je dSpace-ov program koji omogućava dodatno uređenje simulacije nakon što se model učita na MicroLabBox. Nakon otvaranja novog projekta na kojem se učita simulacijski model prethodno odrađen u Simulinku i nakon registriranja platforme i uspostave komunikacije otvara se sučelje u kojem se nalazi tzv *layout* na kojem se postavljaju grafovi, gumbi za pokretanje, zaustavljanje i sl., numerički ispis i unos vrijednosti vezanih uz simulacijski model, svjetlosne lampice za signaliziranje, u suštini HMI sučelje za kontrolu važnih veličina simulacijskog modela. Jedna od dodatnih mogućnosti koje Control Desk omogućava je automatiziranje eksperimenta, pomoću Python skripte u kojem se napišu komande se

automatski pokrene i izvrši testiranje na simulacijskom modelu, no to se nije upotrebljavalo u ovom radu. Control Desk također omogućava i namještanje snimanja podataka i nudi više opcije prikupljanja podataka.

### 3.2.8 Sustav za kontrolu simulacije u stvarnom vremenu

HMI je jedna od ugrađenih mogućnosti koje nudi Control Desk, kako bi se kontroliralo i manipuliralo važnim veličinama simulacijskog modela. HMI sučelje je kako već navedeno podijeljeno u *layout* kojih može biti više pa se promatrane veličine mogu grupirati. Najčešći elementi HMI sučelja uključuju grafove, numeričke ispise vrijednosti, numerički unos referentnih vrijednosti. Povezivanje simulacijskih veličina s elementima HMI-a će se objasniti na primjeru grafa. Otvaranjem novog projekta u Control Desk-u u kartici *Variables* se nalazi izlistanje svih blokova upotrebljenih u Simulinku prilikom izrade matematičkog modela. Najprije se izlistaju svi podsustavi koji se nalaze na glavnoj radnoj površini u Simulinku, a detaljnijim otvaranjem podsustava se izlistaju i blokovi unutar podsustava. Radi lakšeg snalaženja, važno je imenovati blokove ovisno o svrsi koju imaju te ih grupirati. U protivnom se blokovima dodjele generična imena koja Simulink dodjeljuje automatski. Jedan od pristupa imenovanju je da se one varijable (i blokovi) čije veličine će se mjeriti i koje su važnije za simulaciju se imenuju s određenim prefiksom kako bi prilikom izlistanja sve važne varijable bila na jednom mjestu (npr. staviti prefiks out). Odabirom bloka otvara se sučelje koje prikazuje sve podatke koje taj blok sadrži, tako npr. otvaranjem *gain* bloka prikaže se iznos pojačanja i izlazni signal dok blok integratora izlista izlaznu varijablu i početne uvjete (ako ih ima), itd. Primjer izlistanja blokova i podaci pojedinih blokova su prikazani slikom

3.20



SL. 3.20: Ispis Simulink blokova u Control Desku

Jedna od najvažnijih fizikalnih veličina za promatranje kod električnih rotacijskih strojeva je kutna brzina rotora. Kako bi se kutna brzina mogla prikazati na grafu potrebno je u popisu varijabli u Control Desku pronaći blok iz kojeg je izlazni signal brzina i u postavkama grafa odabrati tu varijablu, u postavkama grafa se još dodatno može namjestiti boja i debljina linije grafa, naziv osi i veličine koju graf predstavlja te kako će se skale na osima prilagoditi grafu (hoće li raspon osi biti nepromjenjiv, varijabla, hoće li biti vremenske osi svih grafova biti sinkronizirane i sl.). Slična je procedura za numerički ispis vrijednosti varijable, u postavkama numeričkog ispisa se odabere blok čiju izlaznu varijablu ispisuje. Kao i graf ima opcije prilagodbe ispisa, naziv variable koju predstavlja i sl.. Za razliku od ispisnih elemenata HMI-a koji prikazuju izlazne varijable blokova, kod elementa za numerički unos vrijednosti se odabere vrijednost koju blok posjeduje, npr. u slučaju bloka konstante se

odabere iznos konstante i njima se tijekom simulacije u HMI manipulira.

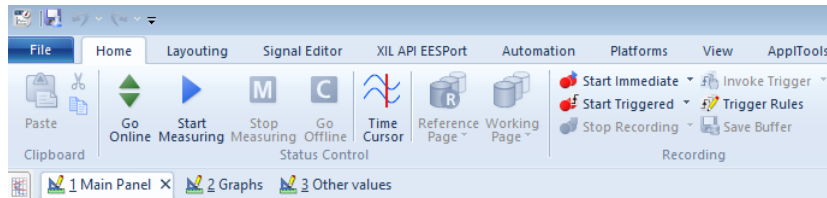
### 3.2.9 Snimanje podataka u Control Desku

Namješteni grafovi u HMI u Control Desk-u se mogu i snimiti, točnije njihovi podaci pomoću kojih se mogu ponovo rekonstruirati. Snimanje podataka se može postići na više načina jedan od njih je *trigger* (okidač) koji u trenutku kad dobije komadnu počinje snimanje podataka svih grafova koji su na HMI sučelju, moguće je i dodatne varijable snimati, ali one se moraju dodati zasebno. Komande koje *trigger* dobiva kako bi pokrenulo snimanje mogu biti: ručno pokretanje ili namještanjem signala koje kad se pređe određeni iznos zadane veličine pokreće snimanje. Prvi način počinje sa snimanjem kad se aktivira naredba *start immediate*, dakle tzv. ručno pokretanje snimanja i snima se sve dok se ne pritisne *stop recording*. Prednost ovog načina snimanja je u tome što se može pokrenuti prije nego se dogode bilo kakve promjene u prethodno pokrenutoj simulaciji i tako spriječiti gubitak informacija. Mana ovakvog načina snimanja je u tome što je kompletna simulacija u jednom nizu (jednom dokumentu) i ako je za daljnje potrebe analize potrebno razdvojiti događaje u simulaciji to se mora odraditi posebno, što oduzima vrijeme. Drugi način pokretanja snimanja pruža više opcija oko početka snimanja. Tako je moguće pokrenuti snimanje kad odabrana varijabla pređe određeni iznos (*POS EDGE*) ili kad padne ispod određene vrijednosti (*NEG EDGE*), može se namjestiti da umjesto neke vrijednosti se radi usporedba varijable na koju je namještanje snimanje s nekom varijablom iz simulacijskog modela. Osim varijable koja je postavljena kao uvjet za početak snimanja, postavlja se vrijeme snimanja, vrijeme koji se odnosi na snimljene varijable prije nego se aktivira *trigger*, tzv. *trigger delay*. Prednost ovog načina snimanja podataka je u tome što se namjesti prije pokretanja simulacije, te nakon završetka simulacije se samo snime datoteke na kojima su podaci, a nedostatak je u tome što može doći do gubljenja informacija, aktiviranja u pogrešnom trenutku. Ovakav način snimanja podataka je povoljan kod automatiziranog izvođenja simulacije.

Oba način pokretanja snimanja (ručno i pomoću *trigger* funkcije, slika 3.21) nakon snimanja spremaju svoje podatke u jednu od ponuđenih datoteka: .csv i budući da je dSpace umrežen s MATALB-om i .MAT datoteka.

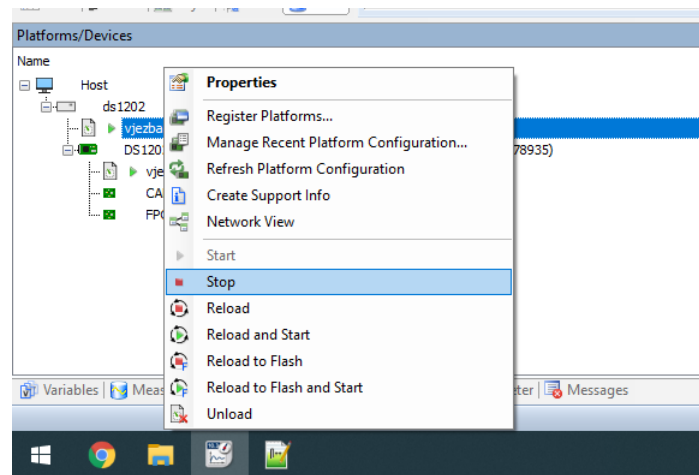
### 3.2.10 Pokretanje simulacije

Sljedeći korak je pokretanje simulacije. Kako je prethodno rečeno Control Desk mora imati uspostavljenu komunikaciju s MicroLabBox-om i imati učitani simulacijski model prethodno odrađen u Simulinku. Pokretanje simulacije preko Control Desk-a se odradi pomoću tipke *Go Online* i tada se model počinje simulirati na MicroLabBox-u o tog trenutka je moguće signale koji su prethodno bili namješteni da se šalju na jedan od digitalnih/analognih izlaza snimiti pomoću osciloskopa/drugog simulatora te isto tako primati signale od drugih izvora/simulatora. Pokretanjem simulacije preko *Go Online* naredbe se ne prikazuju signali na grafovima prethodno uređenim na HMI sučelju za to je potrebna *Start Measuring* naredba. Nakon što simulacija završi kako bi se prekinula komunikacija između Control Deska i MicroLabBox-a potrebna je naredba *Go Offline*, slika 3.21, međutim simulacijski model je i dalje aktivan te se na izlaznim priključcima i dalje mogu detektirati naponske razine koje su bile u trenutku prekidanja komunikacije što može biti opasno te se iz tog razloga mora i simulacijski model ugasiti.



SL. 3.21: Naredbe za pokretanje simulacije i snimanje podataka

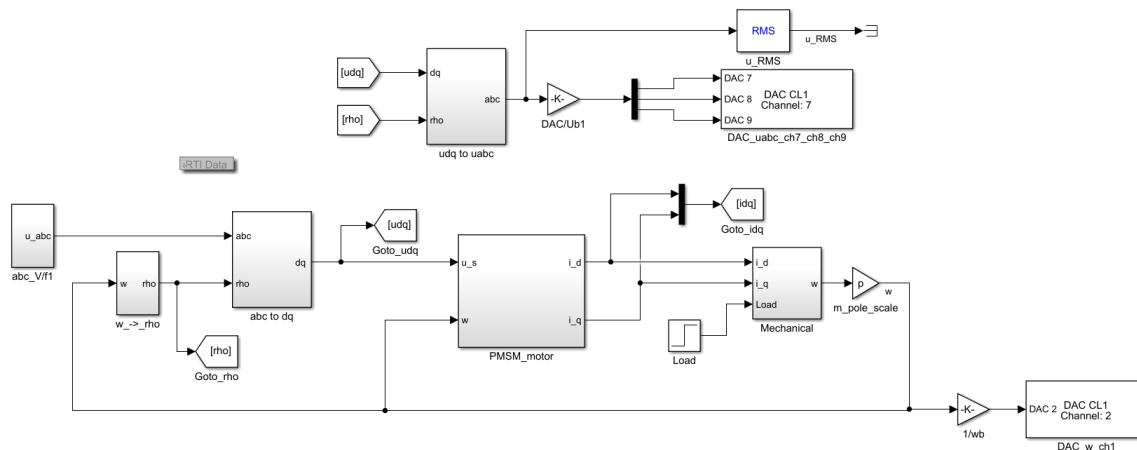
Gašenje simulacijskog modela na MicroLabBox-u se u Control Desku odradi kod popisa platformi, gdje se odabere ds1202 te desnim klikom miša se odabere naredba STOP čime se i simulacijski model zaustavlja, a kako bi ga se ponovo pokrenulo dovoljno je pritisnuti tipku *Go Online* i simulacijski model se opet pokreće na MicroLabBox-u s početnim uvjetima koji su definirani prilikom izrade modela dakle nema pamćenja stanja varijabli iz prethodnih simulacija, slika 3.22



SL. 3.22: Zasutavljanje simulacije na platformi

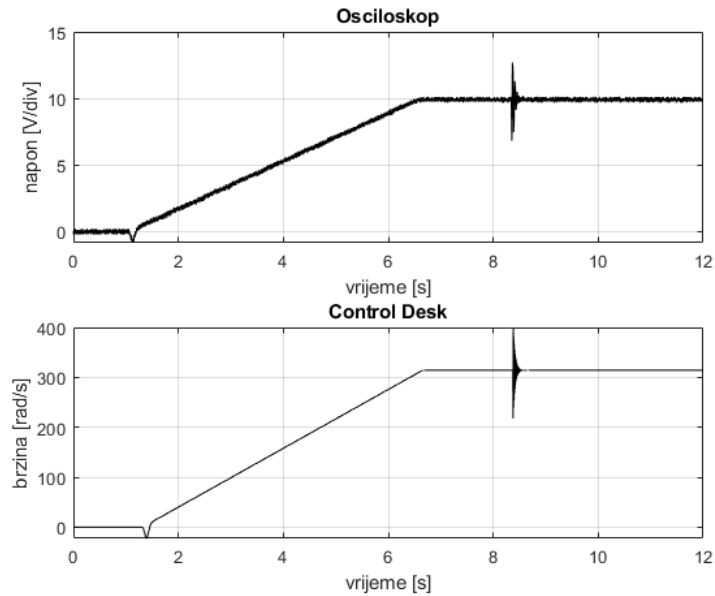
### 3.2.11 Validacija dSpace stvarnovremenskog simulatora

Nakon što su se dSpace-ovi programi instalirali na računalo i povezali s MATLAB-om, potrebno je proučavanje Control Desk okruženja i način rada MicroLabBox-a kao stvarnovremenskog simulatora preko jednostavnih eksperimenata u kojima su se izradili modeli u Simulinku u kojima se izvori valnih oblika (sinusnih za analogne i pulsni generatori za digitalne) signala šalju na priključke i snimali pomoću osciloskopa, a za kontrolu se u Control Desk-u namjesti snimanje grafa iste varijable u HMI sučelju. Kako je u ovom diplomskom riječ o PMSM motoru tako se u validaciji stvarnovremenskog simulatora MicroLabBox implementirao model temeljem jednadžbi iz potpoglavlja 2.3, no umjesto modela vektorskog upravljanja implementiran je model skalarnog upravljanja. Cilj validacije stvarnovremenskog simulatora MicroLabBox je utvrditi šalju li se signali na priključke u stvarnom vremenu. Matematički model u Simulinku se sastoji od dva bloka motora (električni i mehanički) trofazni naponski izvor u kojem se napon linearno povećava kao i frekvencija do nazivnih vrijednosti. Matematički model motora je dvoosni, a matematički model izvora napona je trofazni te je potrebna implementacija matematičkog modela transformacije iz troosnog u dvoosni sustav. Kako je dvoosni matematički model PMSM-a napisan u rotorski orijentiranom dvoosnom sustavu kut za transformaciju se dobiva iz integriranja električne kutne brzine rotora. Kompletan model skalarnog upravljanja PMSM-om je prikazan slikom 3.23.



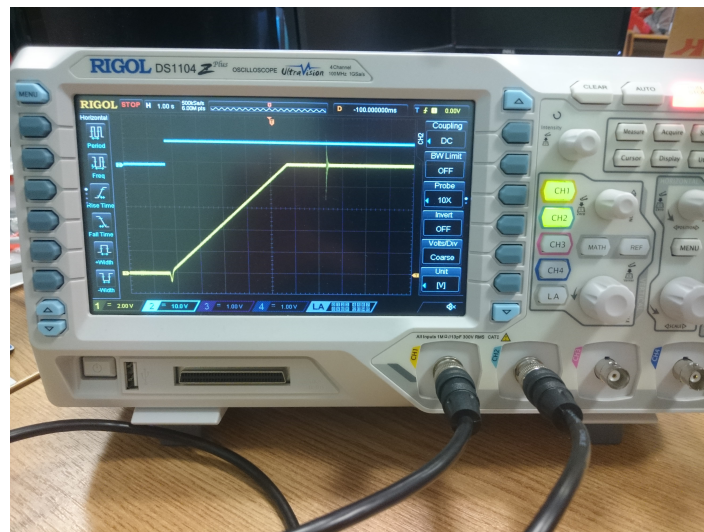
**SL. 3.23:** Simulink model PMSM-a sa skalarnim upravljanjem

Nakon implementacije matematičkog modela u Simulink, kako je već ranije objašnjeno, najprije se matematički model simulira u Simulinku (*offline* simulacija) kako bi se utvrdila ispravnost matematičkog modela te kako bi se ispravile greške koje se pojave jer se u protivnom ne može prijeći na izradu C koda simulacijskog modela. Prelaskom na Control Desk, implementira se simulacijski model prethodno generiran pomoću Simulink Coder-a na MicroLabBox. Namjesti se HMI sučelje kako bi se mogle pratiti važne veličine i kako bi se moglo utjecati na simulaciju (pokrenuti, zaustaviti i sl.). HMI se podjeli u tri kategorije (u tri *layouta*), gdje se na prvom nalaze grafovi brzine, napona i struja u dvoosnom sustavu, na drugom su grafovi troosnih veličina struja i napona uz numerički ispis vrijednosti dok se na trećem nalaze grafovi snaga (prividna, djelatna i jalova snaga) i graf momenta stroja. Snimanje podatka je pokrenuto u trenutku pokretanja simulacije, pritiskom tipke *start immediate*, a snimaju se samo grafovi koji se nalaze na HMI sučelju. Kako bi eksperiment validacije bio točan grafovi snimljeni u Control Desku se moraju usporediti sa grafovima signala koji se šalju na jedan od analognih priključaka na koje se spaji osciloskop. Osciloskop se preko RJ45 kabla povezao s računalom i otvoren je program digitalnog osciloskopa koji preko komunikacijskog kanala prikazuje iste valne oblike na računalu kao one što su na osciloskopu te se preko tog programa spremaju podaci (uzorci) signala kako bi se usporedili sa signalima iz Contro Deska. Sam eksperiment validacije se sastoji od zaleta motora do nazivne brzine linearnim povećavanjem napona i frekvencije do nazivnih vrijednosti, a potom se motor opteretio nazivnim teretom. Zalet je u HMI-u namješten na tipku čiji se pritiskom pokretao motor, a nakon zaleta se pritiskom druge tipke teretio motor. Nadalje su analizirani snimljeni valni oblici signala u Control Desku i na digitalnom osciloskopu.



**SL. 3.24:** Usporedba grafova brzine

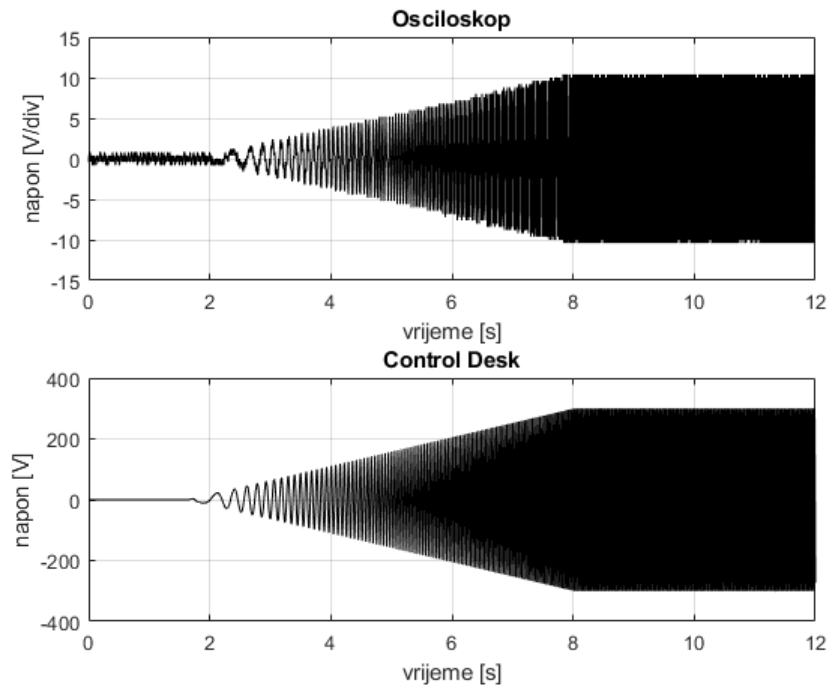
Slika 3.24 uspoređuje graf brzine sa digitalnog osciloskopa i Control Desk-a. Os y gornjeg grafa je mjerena u voltima jer su svi signali koje osciloskop prima naponski, a vrijednost od 10 V odgovara iznosu brzine od 314 rad/s. Na gornjem grafu se također vidi utjecaj okoline jer graf u trenutku konstantne brzine nema stalni iznos nego oscilira. Slika 3.25 prikazuje graf brzine snimaljen na osciloskopu.



**SL. 3.25:** Graf brzine snimaljen na osciloskopu

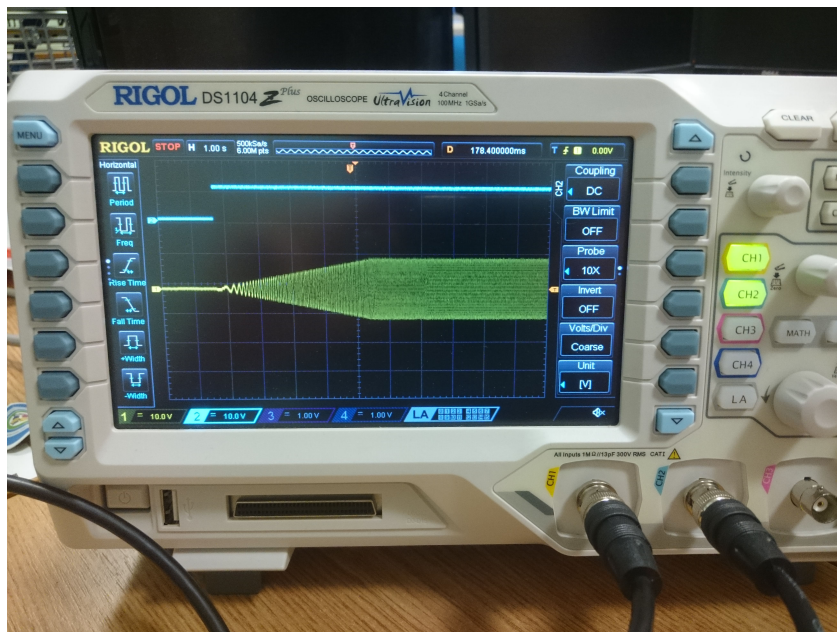
Plavi graf na slici 3.25 predstavlja tzv. *trigger* koji postiže vrijednost 1 kad se pritisne tipka start na Control Desku. Snimljen je kako bi se naznačio početak simulacije. Propad brzine koji se dogodi u trenutku pokretanja se dogodi jer se ne zna položaj magenta na rotoru pa se najprije poravna magnetsko polje magenta s poljem statora i onda se krene zaljetat.





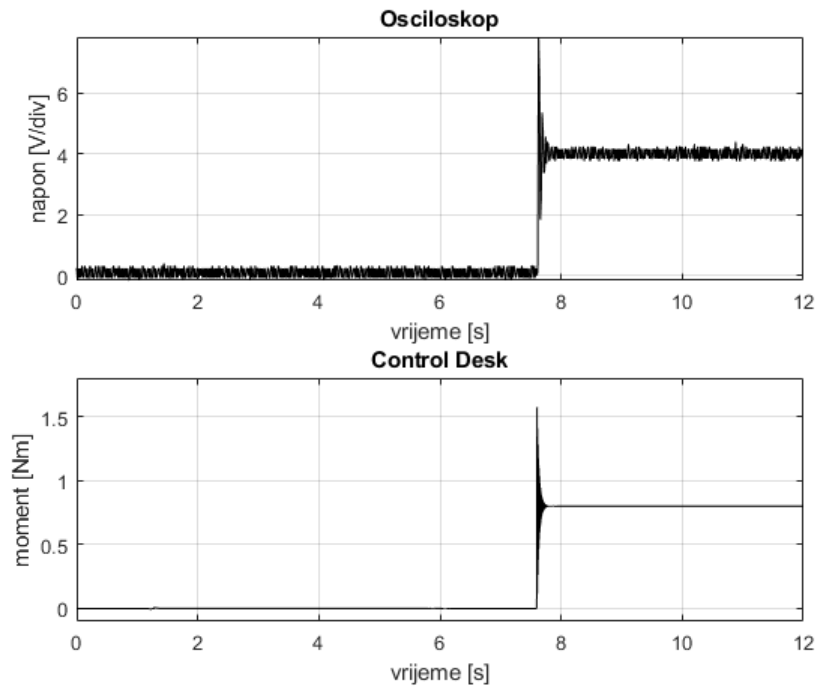
SL. 3.26: Usporedba grafova napona

Slika 3.26 uspoređuje grafove napona sa Control Desk-a i digitalnog osciloskopa. Snimljena je jedna faza (grafovi preostalih faza su identični po valnom obliku samo pomaknuti za  $\frac{2\pi}{3}$ ) trofaznog izvora kako bi se prikazalo linearno povećanje amplitude napona i frekvencije. Kao i kod brzine os y gornjeg grafa je mjerena u voltima koje je osciloskop dobio od MicroLabBox-a te 10 V na osciloskopu predstavlja 300 V u modelu.



SL. 3.27: Graf napona snimaljen na osciloskopu

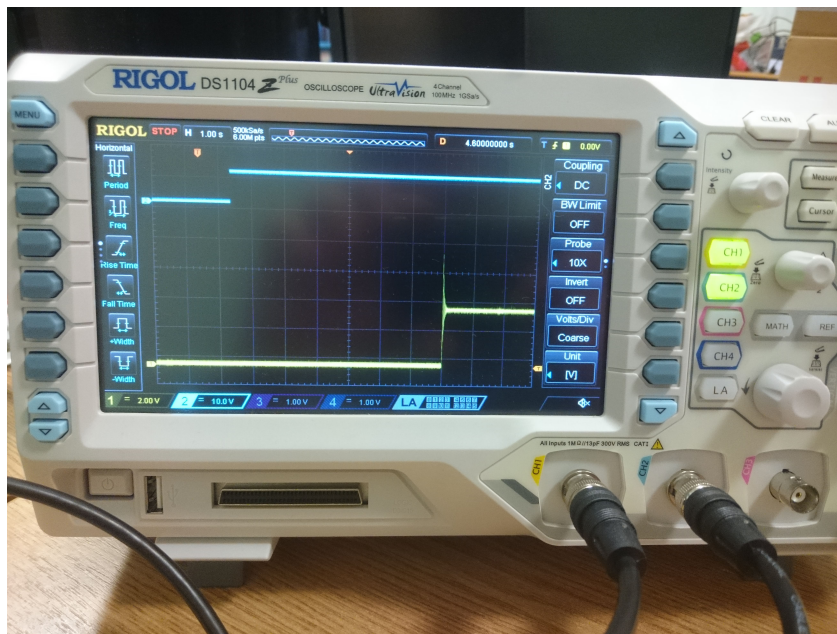
Kao i slika 3.25 tako i slika 3.27 osim grafa napona prikazuje i *trigger* kako bi se naznačio početak zaleta. Nakon što se motor zaletio opteretio se nazvinim teretom, slika 3.28.



SL. 3.28: Usporedba grafova momenta

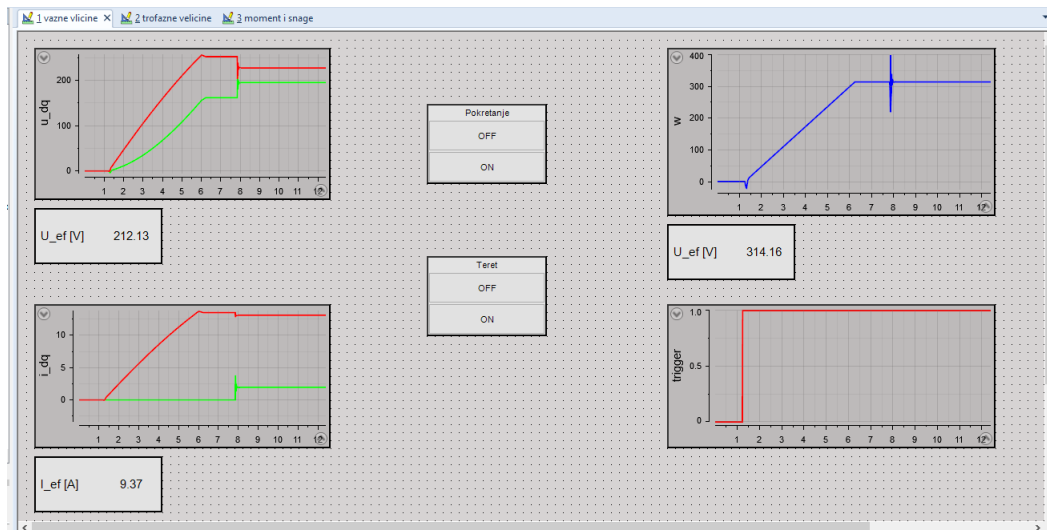
Vidljivo je kako se teret uključio oko 8 sekunde simulacije. Kao i kod grafova brzine i napona tako i kod momenta je gornji graf mjerjen u voltima, te 4 V odgovara momentu od 1 Nm.

Graf momenta snimljen na osciloskopu je prikazan na slici 3.29.



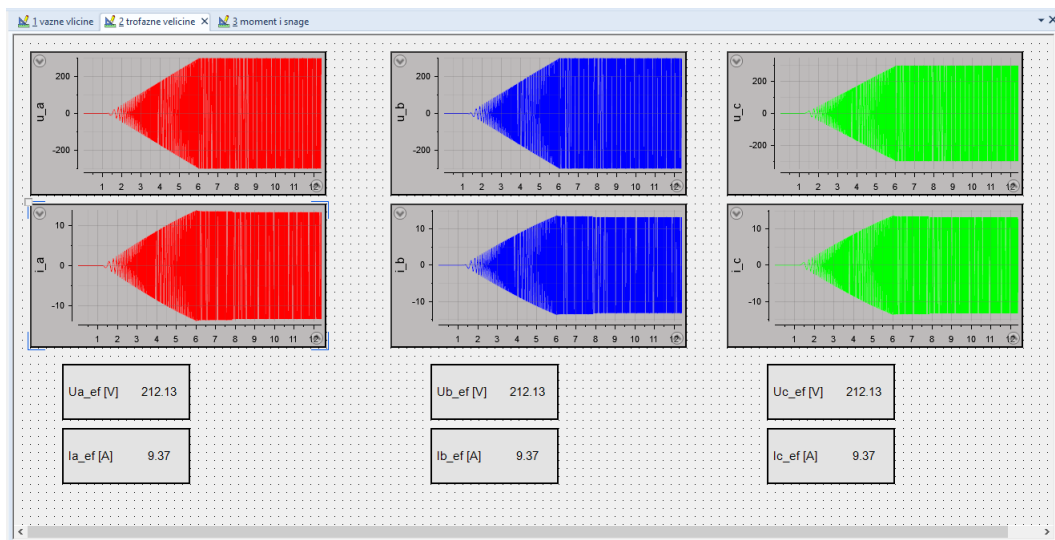
SL. 3.29: Graf momenta snimaljen na osciloskopu





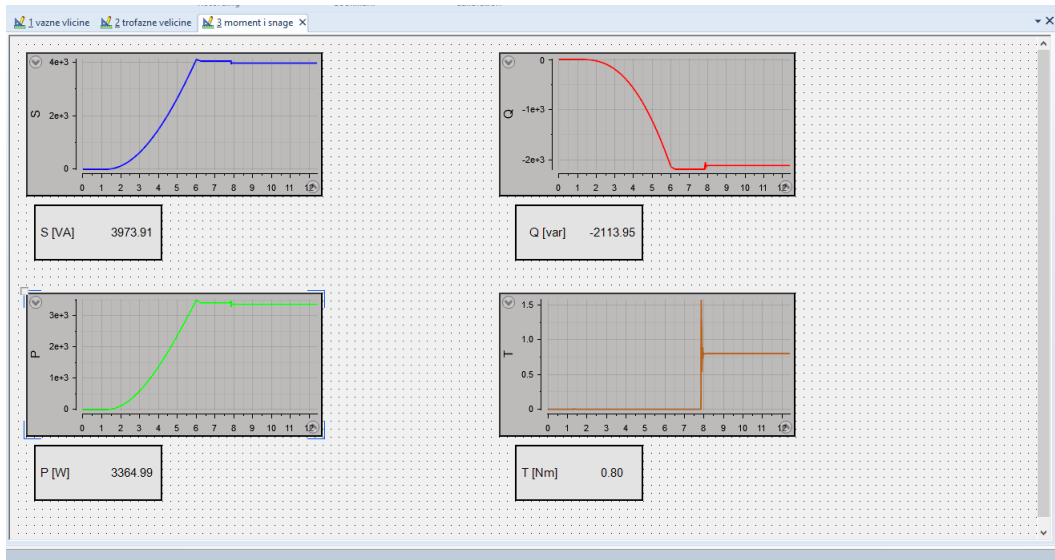
SL. 3.30: Prvo HMI sučelje

Kako slika 3.30 prikazuje, na prvom *layout-u* su prikazane najvažnije veličine motora, tipke za pokretanje i uključenje tereta i graf *trigger-a* kako bi se znalo kad je sustav pokrenut.



SL. 3.31: Druge HMI sučelje

Kako bi se mogla vidjeti promjena frekvencije prilikom zaleta, naponi i struje su preko inverzne Park-ove transformacije prikazani u troosnom sustavu. Struje i naponi svih triju faza su prikazani grafički te njihove efektivne vrijednosti su ispisane numerički, slika 3.31



**SL. 3.32:** Treće HMI sučelje

Osim električnih veličina, prikazan je graf momenta uz numerički ispis vrijednosti te sve tri snage (prividna, djelatna i jalova), slika 3.32. Temeljem eksperimenta je utvrđeno kako MicroLabBox proslijeđuje podatke na analogne izlaze u stvarnom vremenu.

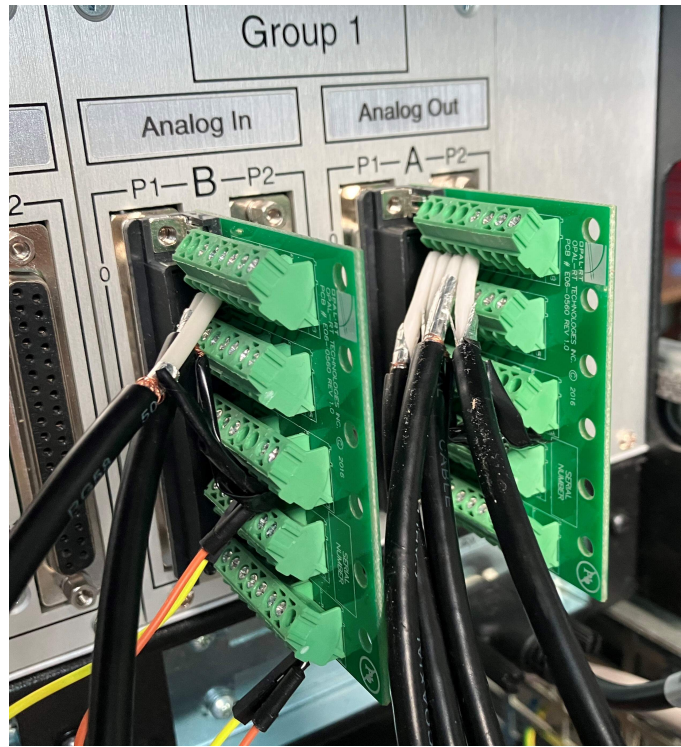
## 4. ANALIZA REZULTATA KOSIMULACIJE

Nakon validiranja dSpace-ovog MicroLabBox-a, započet je rad na kosimulaciji u stvarnom vremenu između dSpace-ovog MicroLabBox-a stvarnovremenskog simulatora sa OP5031 stvarnovremenskim simulatorom. Kako je riječ o vektorskom upravljanju PMSM motorom, simulacijski model regulatora je implementiran na dSpace-ov MicroLabBox, a simulacijski model PMSM-a je implementiran na OP5031 stvarnovremenski simulator, slika 4.1.



SL. 4.1: OP5031 i OP5600 povezani sa MicroLabBox-om u kosimulaciju

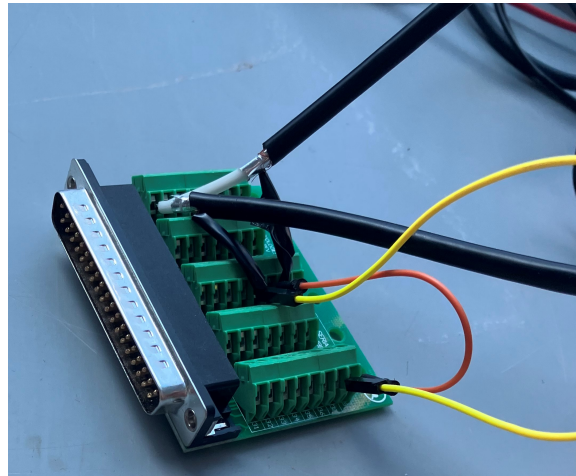
Matematički model regulatora temeljem primljenim vrijednosti struja ( $d$  i  $q$  komponente) i kutne brzine rotora računa napon koji je potom prosljeđen motoru. Kako bi ova kosimulacija bila HIL simulacija, simulacijski modeli se moraju povezati kablovima i to tako da dSpace-ov MicroLabBox simulator OP5031 simulatoru šalje dva signala ( $d$  i  $q$  komponente napona), a OP5031 simulator šalje dSpace-ovom MicroLabBox-u četiri signala ( $d$  i  $q$  komponente struje, kutnu brzinu i moment za mjerenje budući da ne treba za proračune u regulatoru). Kako su priključci na simulatorima različiti po vrsti (MicroLabBox za analogne signale koristi BNC priključke, a OP5031 DB37 priključke) upotrijebili su se BNC-BNC kablovi koji su se podijelili pa su od jednog BNC-BNC kabla nastala dva kabla s jedne strane BNC, a s druge strane izložene žice. Izloženi dio kabla se sastoji od žice kojom se prenosi signal i plašta, prema tablici iz OP5031 priručnika su se žice spojile na određene utore DB37 priključka. Slika 4.2 prikazuje ulazni DB37 priključak koji sadrži dva signala (dvije komponente napona) koje šalje MicroLabBox i DB37 priključak s četiri signala koje OPAL šalje MicroLabBox-u.



SL. 4.2: Ulazni i izlazni priključci OP5600 simulatora

Namjestili su se HMI sučelja za oba simulatora. U LabVIEW-u se namjestilo HMI sučelje za praćenje veličina motora, veličina poput: brzina, struje, naponi, moment motora. Navedene veličine su prikazane grafički, poseban graf za kutnu brzinu, struje, napone i moment stroja te dodatno numerički ispis vrijednosti istih veličina. Tijekom eksperimenta je predviđeno terećenje motora nazivnim teretom, te je zato dodan numerički unos vrijednosti tereta. HMI sučelje u Control Desk-u je napravljeno u tri *layout*-a. Prvi je na grafu prikazivao valni oblik kutne brzine stroja, numerički ispis svih veličina važnih za praćenje sustava (naponi, struje, brzina, moment), dodan je još numerički unos referentne vrijednosti kutne brzine u rad/s. Na drugom *layout*-u su postavljeni grafovi napona, struja i momenta dok su na trećem postavljeni grafovi regulacijskih odstupanja koja ulaze u regulator uz numerički ispis istih (tri su regulacijska odstupanja, jer se u regulator implementirana tri PI regulatora). Ovako spojeni simulatori s HMI sučeljima čine HIL testnu stanicu za testiranje simulacijskih modela regulatora. Redoslijed pokretanja simulatora u kosimulaciji je sljedeći: najprije se pokrene OP5031, simulator na kojem je simulacijski model PMSM motora, a onda MicroLabBox, simulator na kojem je simulacijski model regulatora. Pokrenuti simulator sa simulacijskim modelom motora iako ne dobiva napon od simulacijskog modela regulatora šalje regulatoru vrijednosti kutne brzine i struja. Dok je referentna vrijednost kutne brzine nula, nema regulacijskog odstupanja i simulacijski model motora miruje jer ne dobiva napon. Kad se na HMI promijeni referentna vrijednost kutne brzine, regulatori imaju regulacijsko odstupanje i temeljem tog odstupanja formiraju iznos napona koji se prosljeđuje simulacijskom modelu motora. Eksperiment koji je proveden u kosimulaciji se sastoji od zaleta motora do nazivne kutne brzine, potom slijedi terećenje nazivnim teretom, zatim se promijeni referentni iznos kutne brzine na iznos negativne nazivne brzine, rasterećenje motora i za kraj zaustavljanje motora. Prilikom prvog uspješnog pokretanja kosimulacije i zaleta motora pojavilo se odstupanje iznosa kutne brzine, točnije referentni iznos postavljen u Control Desk-ovom HMI-u je 314 rad/s, a u LabVIEW-ovom HMI-u je ispisana vrijednost od 317 rad/s. Kako u DB37 kartici svaki signal ima pozitivni i negativni priključak na kojeg se spaja, negativni priključci nisu na istom potencijalu već je svaki na svom potencijalu, dok

BNC priključci su odvojeni, ali uzemljeni (na istom potencijalu, 0 V). Ta razlika je uzrokovala odstupanje napona za 3 mV kad je napon trebao biti 0 V. Problem je riješen tako što su se svi negativni priključci DB37 kartice spojili na uzemljenu točku na kartici. Na slici 4.3 se negativni priključci spojeni preko kablova (narančastog i žutog) na uzemljenu točku.



**SL. 4.3:** Negativni priključci spojeni na uzemljenu točku

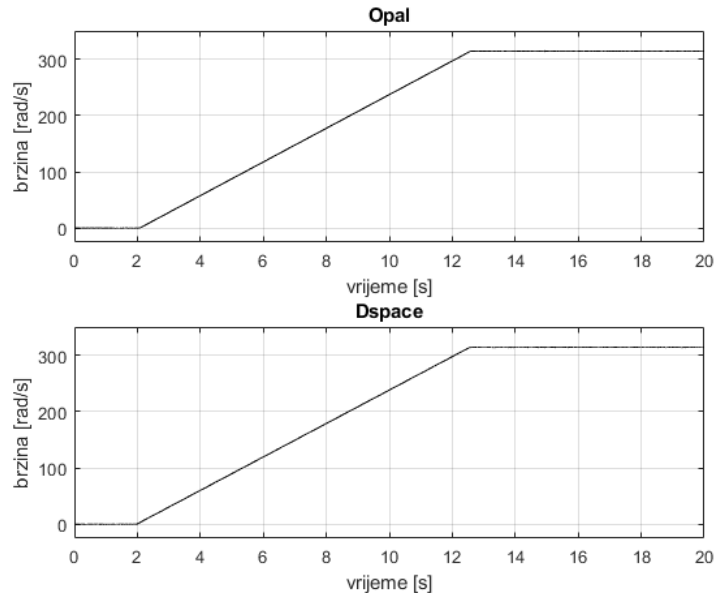
Promjene su napravljene i u Simulink modelu PMSM-a, tako što je dodan blok s naponom iznosa -3 mV kako bi se poništio utjecaj dobivenog signala koji ima odstupanje. Što se tiče snimanja podataka, snimanje u Control Desk-u je pokrenuto ručno, pritiskom naredbe *start immediate*, dok se u RT-Lab programu namjestilo tri snimanja: zalet i terećenje, reverziranje te rasterećenje i zaustavljanje. U sva tri snimka su se snimali iste varijable (brzina, oba napona, obje struje, moment stroja). Zalet s terećenjem je namješten na POZ EDGE, a iznos na 1 rad/s, dakle kad je kutna brzina motora prešla 1 rad/s pokrenuto je snimanje podataka uz uračunato 10 vremena prije početka snimanja. Zalet s terećenjem je trajao 20 sekundi. Reverziranje je namješteno na NEG EDGE i to kad je kutna brzina rotora pala ispod 310 rad/s uz uračunato vrijeme prije snimanja od 10%. Reverziranje je trajalo 25 sekundi. Rasterećenje i zaustavljanje motora je namješteno na POZ EDGE kad brzina pređe iznos od -310 rad/s uz prethodno rasterećen motor. Snimanje rasterećenja i zaustavljanja je namješteno na 25 sekundi uz uračunavanje 10% vremena prije početka snimanja Tablica 4.1 prikazuje podatke o motoru koji se koristio u kosimulaciji.

$R_s$	3.25 $\Omega$	$L_d$	18 mH
$L_q$	34 mH	$\Psi_M$	0.341 Wb
p	3	$M_n$	5.4 $N_m$
J	0.005 $kgm^2$	$n_n$	3000 o/min
P	1.7 kW	$U_n$	400 V

**Tab. 4.1:** Podaci o PMSM-u

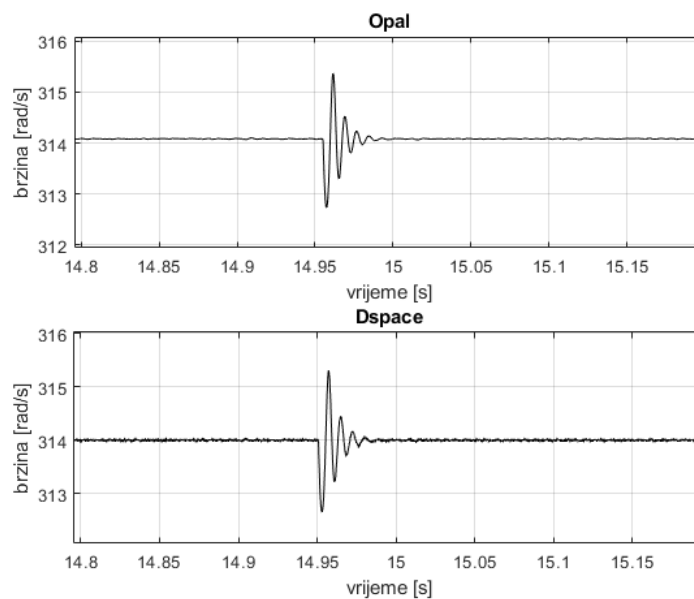
## 4.1. Zalet i terećenje

Najprije se motor zaletio do nazivne brzine, 3000 o/min odnosno 314 rad/s. Nakon što se zaletio i neko vrijeme se vrtio neopterećen nazivnom brzinom, uključio se teret nazivnom iznosa prema tablici 4.1. Temeljem podataka snimljenih na RT-Lab-u i Control Desku generirali su se grafovi.



**SL. 4.4:** brzina motora za vrijeme zaleta

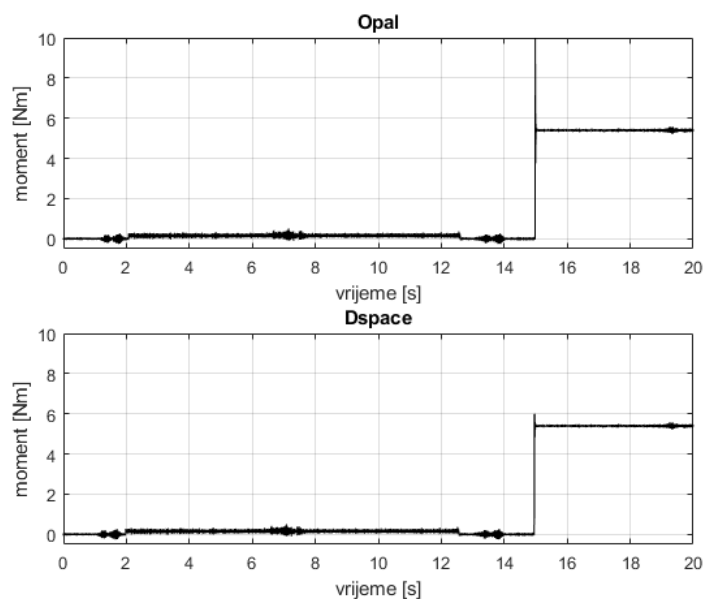
Slika 4.4 uspoređuje grafove brzina tokom zaleta. Zalet je počeo oko druge sekunde, u trenutku kad se na HMI sučelju Control Desk-a postavila referentna brzina na nazivnu. Nakon zaleta se oko 15. sekunde se uključio teret, slika 4.5.



**SL. 4.5:** brzina motora tokom uključivanja tereta

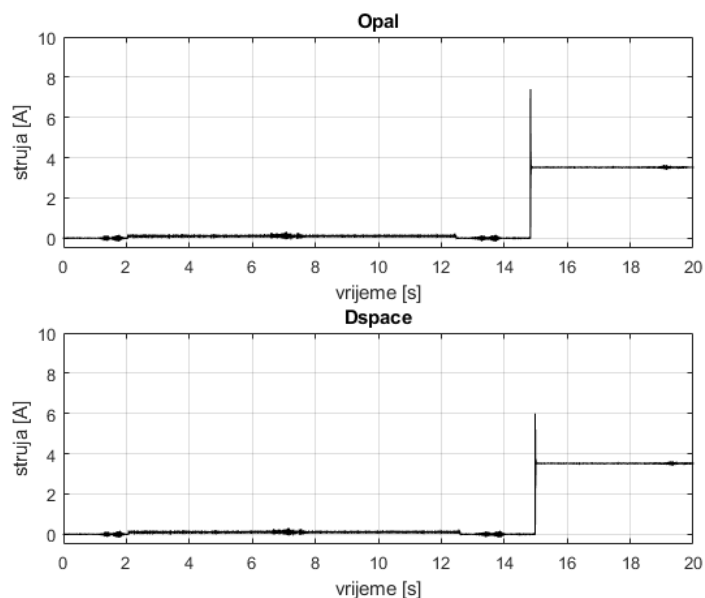


Vidi se kako prijelazna pojava traje kraće od 50 ms nakon čega se ustali na nazivnoj brzini. Oscilacije u iznosu brzine su oko 1 rad/s što je dokaz dobrog algoritma za upravljanje.

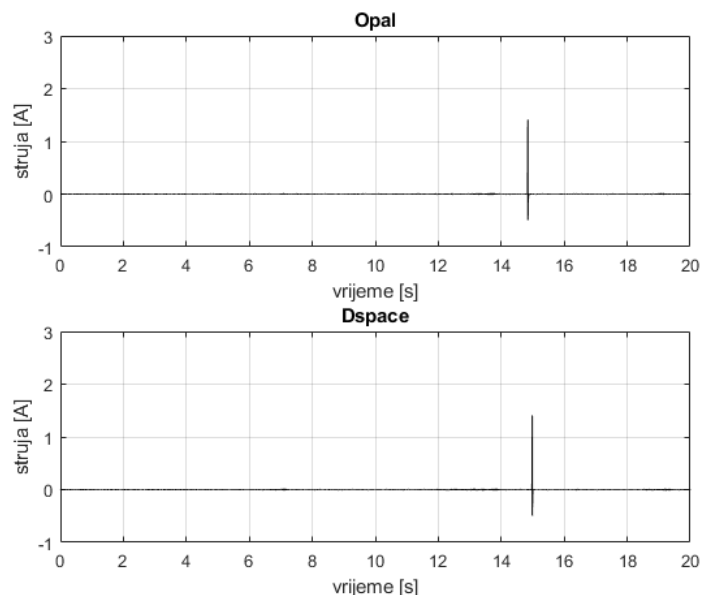


**SL. 4.6:** moment motora za vrijeme zaleta i uključenja tereta

Slika 4.6 prikazuje momen motora. Za vrijeme zaleta motor, prema jednadžbi 2-11, ima moment jednak dinamičkom momentu uzrokovanom promjeni brzine. Nakon zaleta moment motora je nula dok se ne uključi teret oko 15. sekunde, tada nakon prijelazne pojave je moment motora jednak momentu tereta, prema jednadžbi 2-11, jer nema promjene brzine pa su momenti jednaki. Budući da moment motora, prema jednadžbi 2-33, od struja ovisi o struji  $q$  komponente jer je referentna vrijednost struje  $d$  komponente namješšana na 0 A.

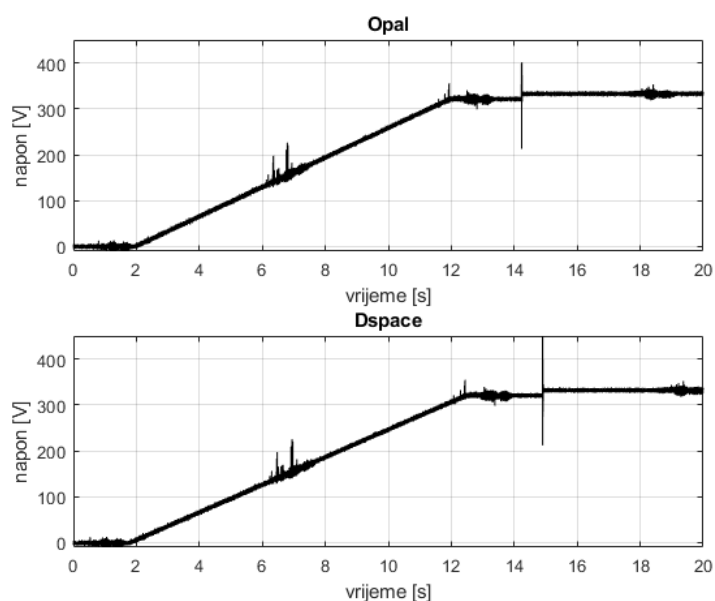


**SL. 4.7:** struja  $q$  komponente za vrijeme zaleta i uključenja tereta



**SL. 4.8:** struja  $d$  komponente za vrijeme zaleta i uključenja tereta

U trenutku uključenja tereta se ipak mijenja iznos  $d$  komponente struje, no brzo se kompenzira i vraća na referentnu vrijednost. Kad  $d$  komponenta struje nije nula tada se, prema jednadžbi 2-33, aktivira reluktanti moment stroja i produljuje se vrijeme prijelazne pojave uzrokovane pojavom tereta. Promjene struja preko regulatora stuje utječu na iznose napona.

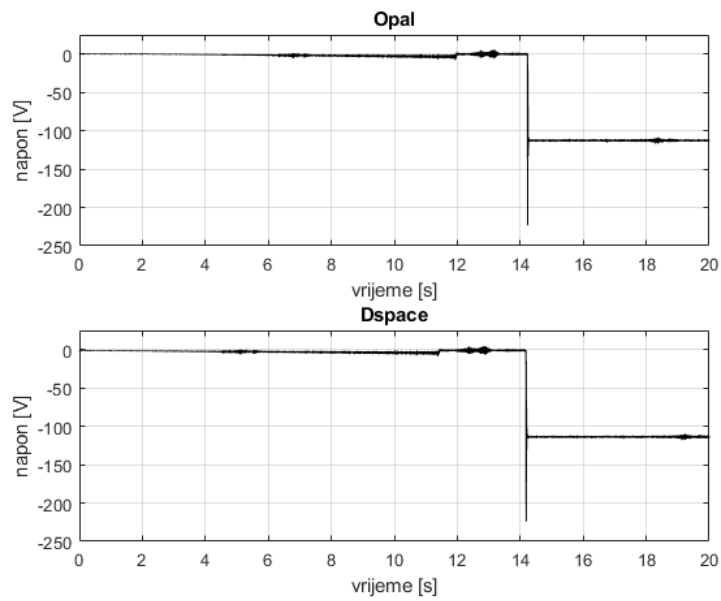


**SL. 4.9:** napon  $q$  komponente za vrijeme zaleta i uključenja tereta

Napon  $q$  komponente se linearno povećavao za vrijeme zaleta do iznos nešto većeg od 300 V, a nakon uključenja tereta iznos je dodatno porastao. Promjena iznosa  $q$  komponente napona je, prema jednadžbi 2-55, uzrokovana tokom trajnog magnet (  $d$  komponenta struje ne utječe jer je nula) i brzinom te reguliranjem napona temeljem regulacijskog odstupanja struje  $i_q$ . Na oba grafa slike 4.9 ima određenih oscilacija oko 6. sekunde te nakon zaleta,



što je posljedica utjecaja okoline, jer kako je ranije napomenuto između simulatora se slao skalirani signal čiji je maksimalni iznos 10 V pa je svaki utjecaj iz okoline utjecao na iznos i skalirao se zajedno sa željenim signalom.

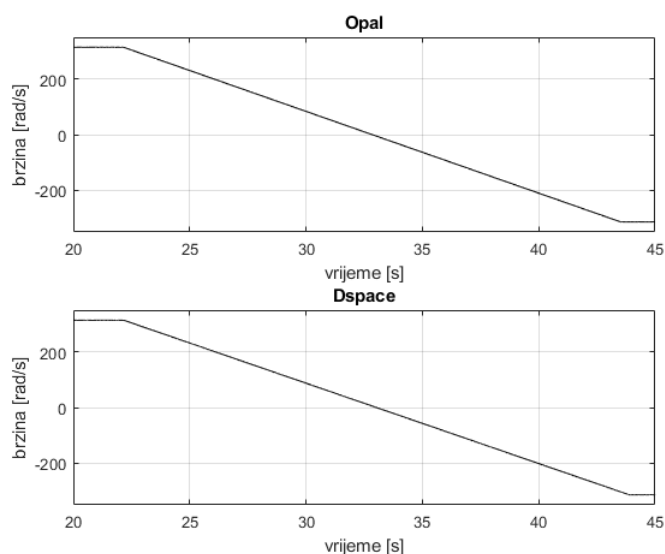


**SL. 4.10:** napon  $d$  komponente za vrijeme zaleta i uključenja tereta

Napon  $d$  komponente, slika 4.10 se za vrijeme zaleta smanjivao linerano što je, prema jedandžbi 2-54, posljedica  $q$  komponente struje koji umanjuje iznos napona  $d$ . Nakon zaleta napon je nula, a ponovnim uključenjem tereta, postiže napon od -110 V što je opet posljedica  $q$  komponente struje, koja je ovoga puta nazivnog iznosa.

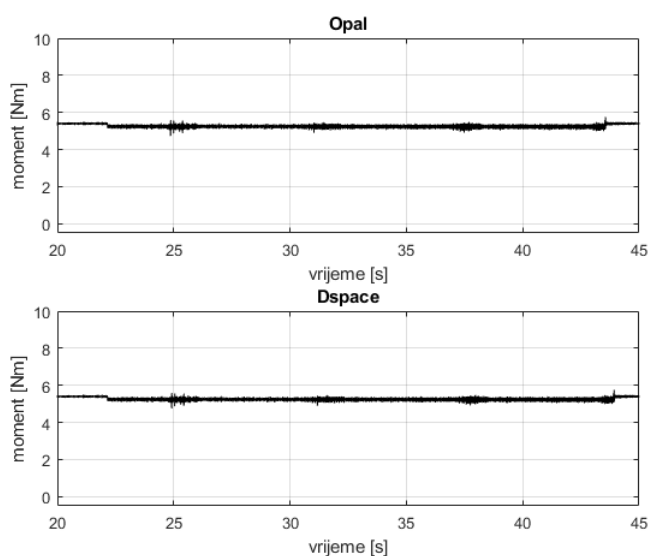
## 4.2. Reverziranje

Nakon što se motor neko vrijeme vrtio s nazivnim teretom, promijenio se iznos referentne brzine stroja na negativni iznos nazivne brzine, dakle na  $-314$  rad/s. Reverziranjem je motor prešao u generatorski režim rada budući da su moment i brzina suprotnih predznaka.



SL. 4.11: brzina motora za vrijeme reverziranja

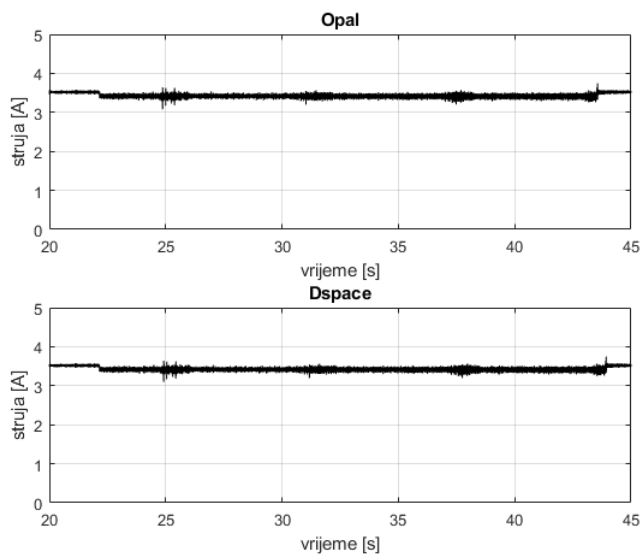
Reverziranje je trajalo oko 20 sekundi, 10 sekundi dok se motor zaustavio i 10 sekundi ubrzavanja do negativne brzine.



SL. 4.12: moment motora za vrijeme reverziranja

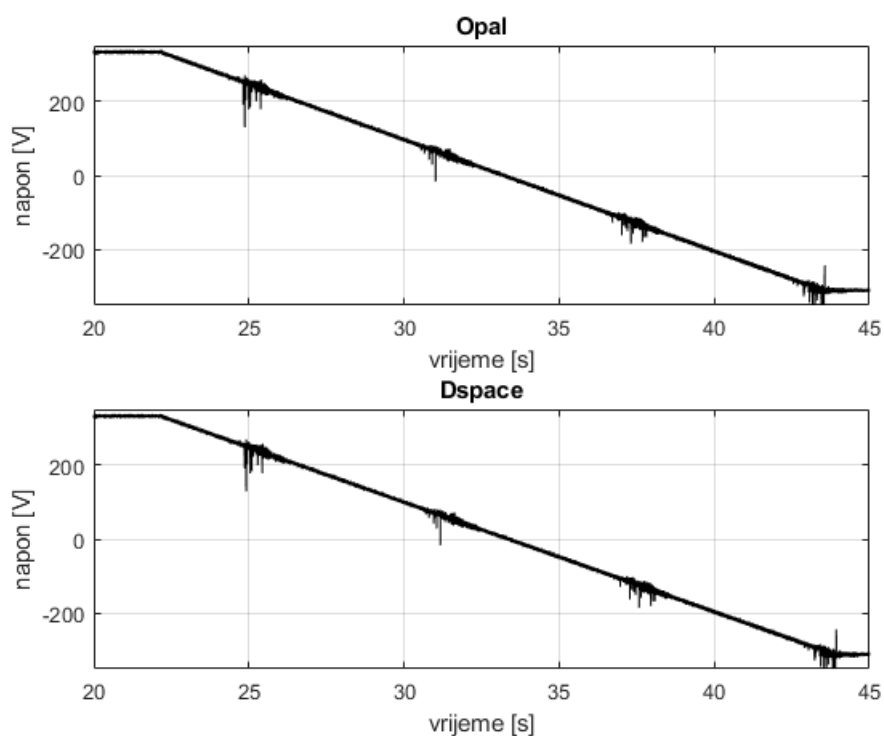
Prema slici 4.12 se vidi kako se iznos momenta motora za vrijeme reverziranja smanjio. Prema jednadžbi 2-11 moment ubrzavanja je negativnog iznosa (zbog negativne promjene brzine), a moment tereta je pozitivnog iznosa i to rezultira manjim momentom motora od momenta tereta, no nakon što se motor ustali na novoj brzini, moment motora se opet

izjednačava sa momentom tereta. Posljedično samnjenu momenta se i smanji  $q$  komponenta struje dok je  $d$  komponenta ostala nepromjenjena, slika 4.13:

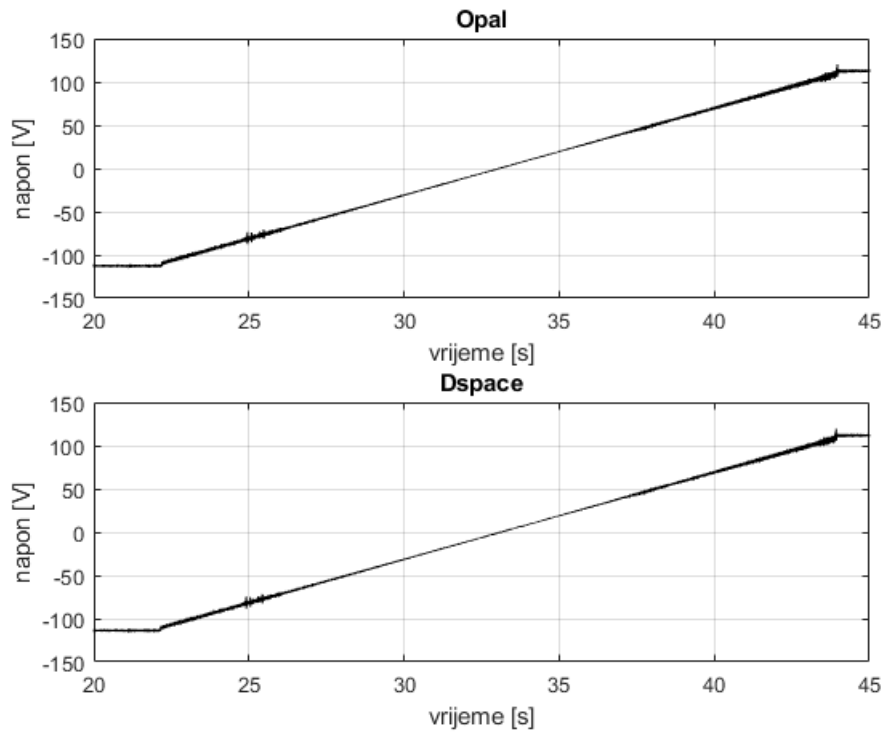


SL. 4.13:  $q$  komponenta struje za vrijeme reverziranja

Reverziranje brzine vrtnje značajnije utječe na iznose komponenti napona. Prema jednadžbama 2-55 i 2-54 zbog promjene predznaka brzine će se promijeniti predznaci napona, slika 4.14 i 4.15.



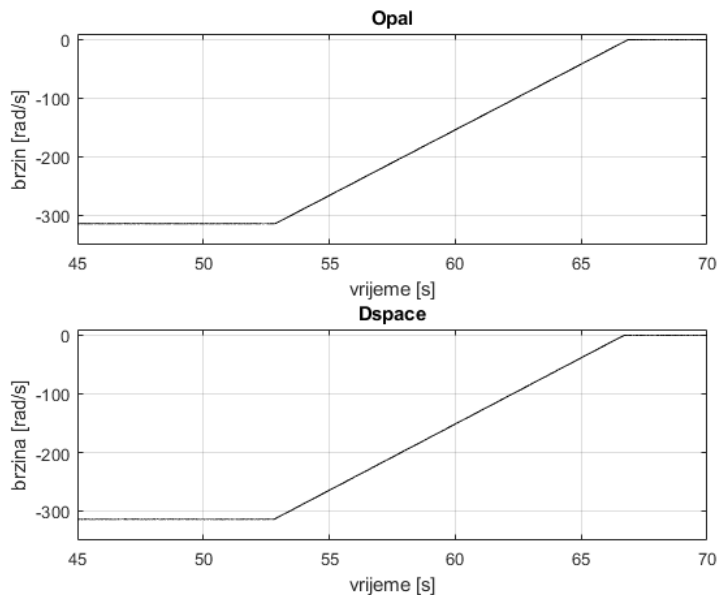
SL. 4.14:  $q$  komponenta napona za vrijeme reverziranja



SL. 4.15:  $d$  komponenta napona za vrijeme reverziranja

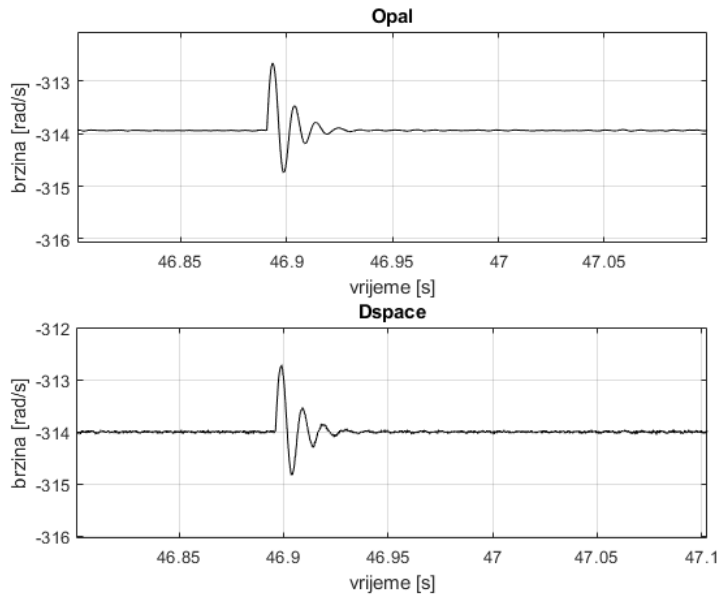
### 4.3. Rasterećenje i zasutavljanje

Nakon što se motor neko vrijeme vrtio na negativnoj nazivnoj brzini, rasteretio se i za kraj se zaustavio.



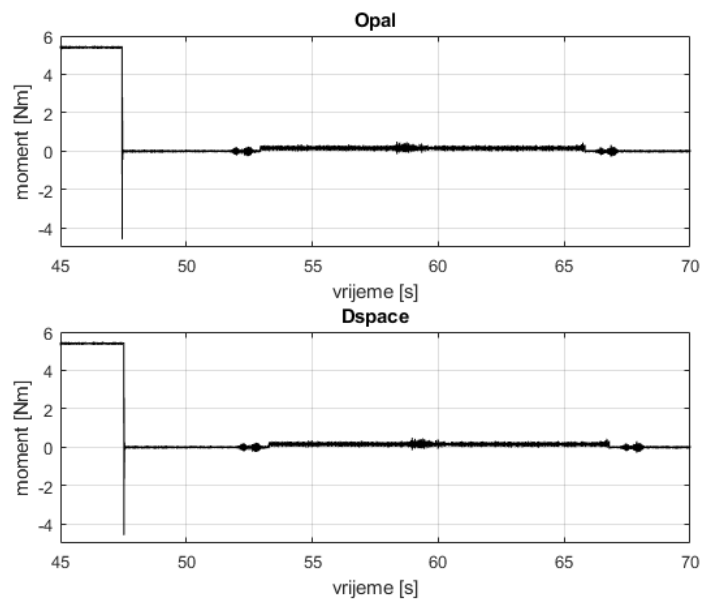
SL. 4.16: brzina motora za vrijeme reverziranja

Projenom referentne brzine na 0 rad/s se motor počeo zasutavljati, odnosno ubrzavati budući da je u generatorskom režimu rada, no prije toga se rasteretio.

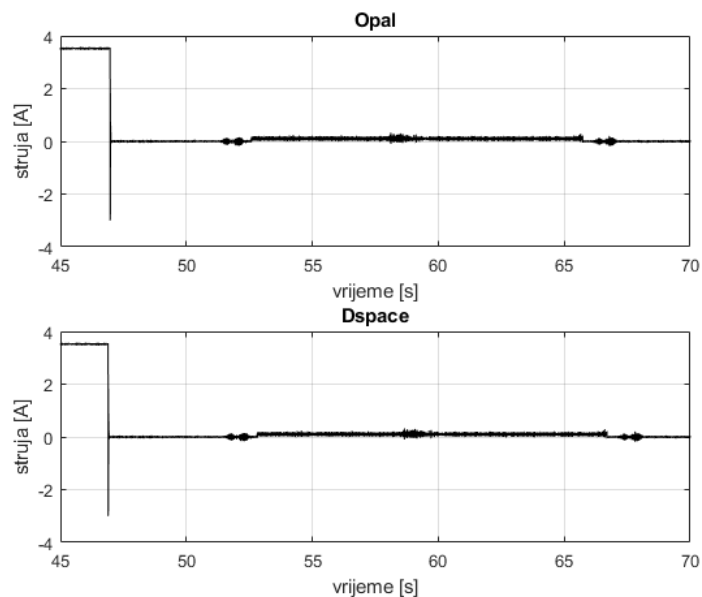


SL. 4.17: brzina motora za vrijeme rasterećenja

Isključenjem tereta, prema slici 4.17, motor krene ubrzavati, što je opet, prema jednadžbi 2-11, posljedica promjene brzine, ali ovoga puta pozitivne promjene što uzrokuje ubrzavanje, no kako je referentna brzine  $-314 \text{ rad/s}$  tako se izregulira i vrati na zadanu brzinu. Projenom tereta se mijenja moment motora, a posljedično i iznos struje motora, slika 4.18 i 4.19



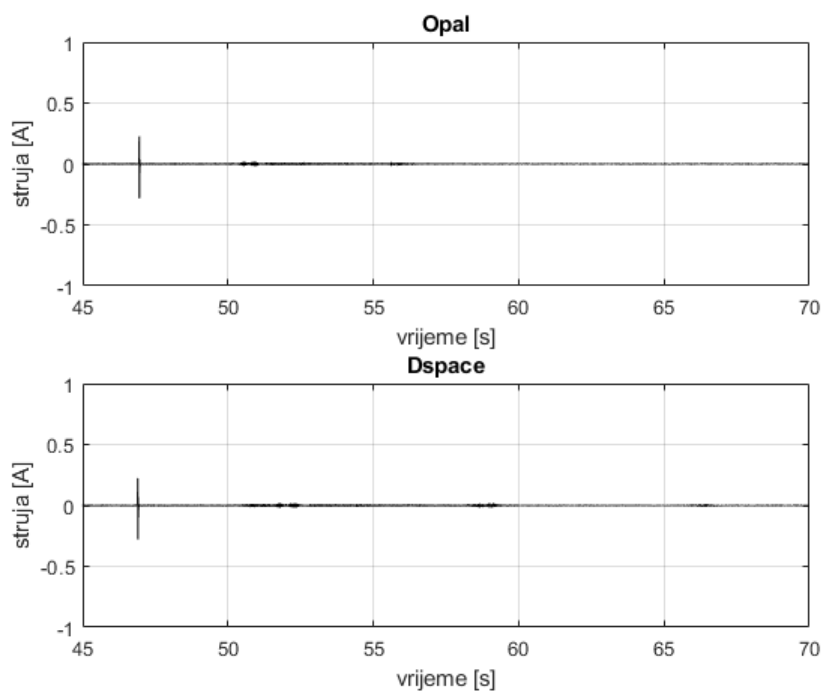
SL. 4.18: moment motora za vrijeme rasterećenja i zasutavljanja



**SL. 4.19:**  $q$  komponenta struje motora za vrijeme rasterećenja i zasutavljanja

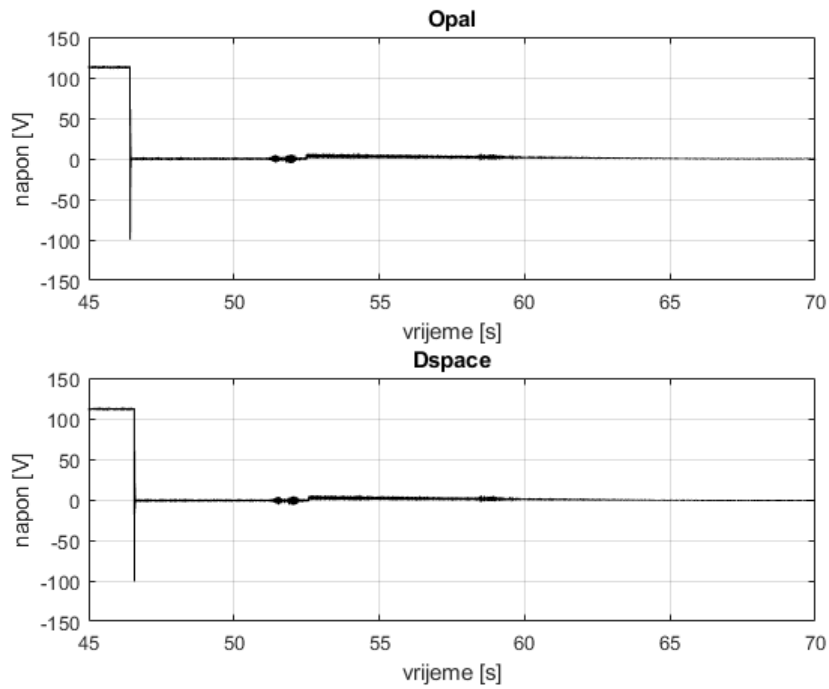
Nakon rasterećenja je nastupilo zaustavljanje motora. Za vrijeme zasutavljanja moment, a posljedično i  $i_q$  struja, su pozitivnog iznosa jer se motor ubrzava od negativne brzine do nule.

Struja  $i_d$  je za vrijeme rasterćenja, kao i za vrijeme terećenja, imala određeni iznos koji je variao od referentne, slika 4.20.

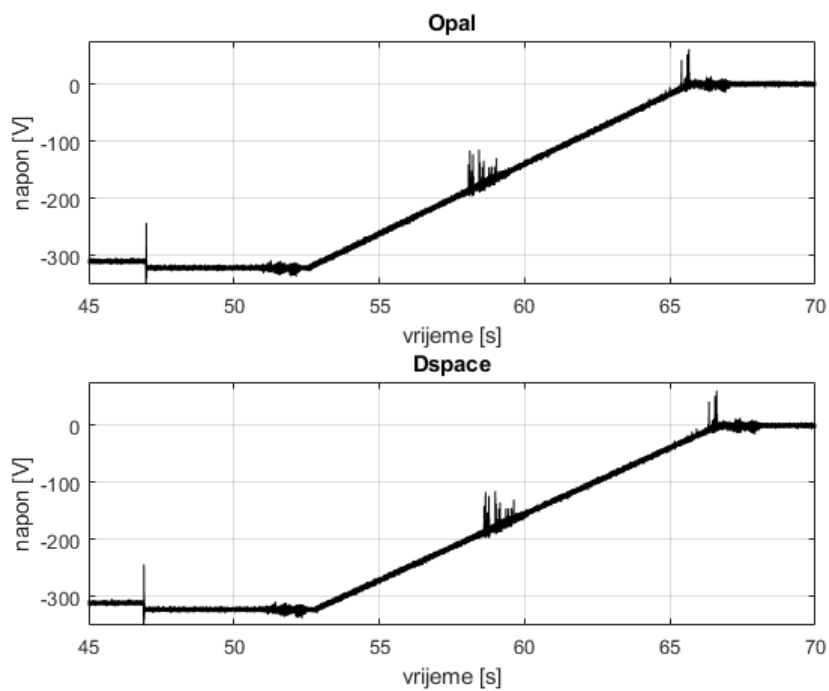


**SL. 4.20:**  $d$  komponenta struje motora za vrijeme rasterećenja i zasutavljanja

Suklando promjenama brzine i struja su se mijenjali i naponi, slika 4.21 i 4.22



SL. 4.21:  $d$  komponenta napona motora za vrijeme rasterećenja i zasutavljanja

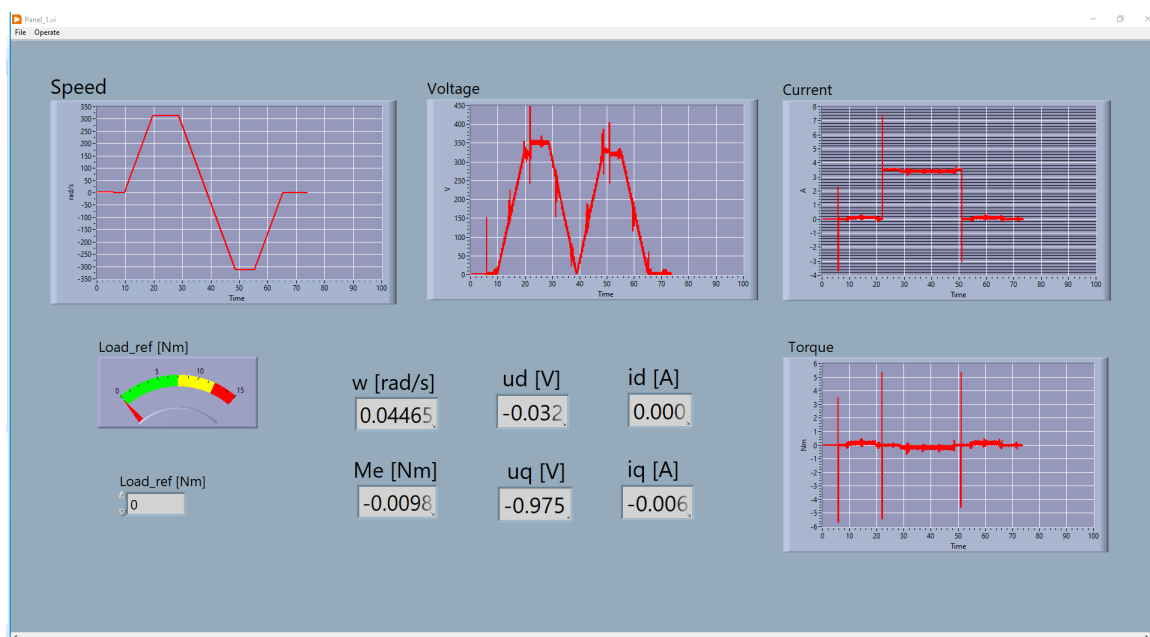


SL. 4.22:  $q$  komponenta napona motora za vrijeme rasterećenja i zasutavljanja

Napon  $u_d$  je isključenjem tereta pao na 0 V, a za vrijeme zasutavljanja je napon linearno rastao, ovoga puta je napon pozitivan jer je brzina negativan, pa prema jedandžbi (2-54) je napon pozitivan.

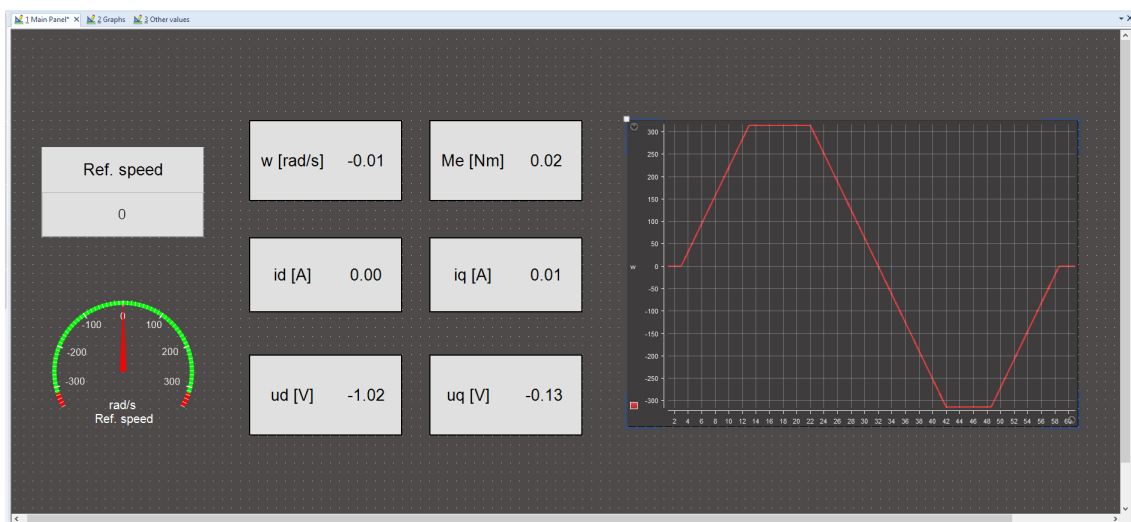
Napon  $u_q$  se sukladno promjeni brzine mijenjao, točnije povećavao iznos do 0 V.

## 4.4. HMI sučelja



SL. 4.23: HMI sučelje u LabVIEW-u

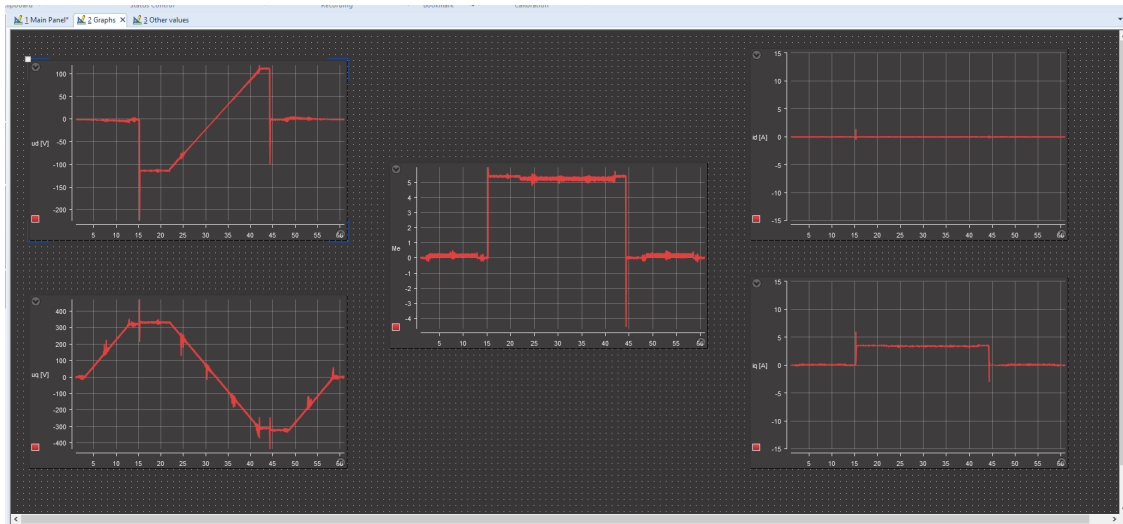
U HMI sučelje u LabVIEW-u su postavljene važne veličine za PMSM motor: naponi, struje, moment stroja, brzina. Sve navedene veličine su prikazane grafički i numerički ispis i dodan je numerički unos vrijednosti tereta.



SL. 4.24: Prvi *layout* u Control Desk-u

Prvi *layout* u Control Desku je glavni *layout* za kontrolu veličina na simulatoru na kojem je model regulatora. Sadrži numerički ispis veličina stroja: napona, struja, momenta, brzina i graf brzine. Na ovom sučelju se nalazi numerički unos referentne kutne brzine stroja. Prilikom pokretanja je namještena referentna vrijednost na nulu.





SL. 4.25: Drugi *layout* u Control Desk-u

Na drugom *layoutu* se nalaze grafovi svih veličina čije se numeričke vrijednosti nalaze na prvom *layoutu* (osim brzine koja je na prvom). Ove informacije su sekundarne, jer se isti grafovi već nalaze na LabVIEW-ovom HMI-u kao glavne informacije.



SL. 4.26: Treći *layout* u Control Desk-u

Ovaj *layout* grafički i numerički prikazuje vrijednosti regulacijskih odstupanja svakog regulatora. Prilikom ispravnog rada regulatora ove vrijednosti su nula osim za vrijeme poremećaja kad ima odstupanja koje se izregulira kako bi odstupanje bilo nula.

## 5. ZAKLJUČAK

Ovim diplomskim radom je razvijeno vektorsko upravljanje sinkronim motorom s permanentnim magnetima. Najprije je definiran sinkroni stroj (osnovne karakteristike) te je dan uvid u vrste sinkronih motora s permanentnim magnetima. Prelaskom na matematičko modeliranje stroja se najprije kreće od troosnog modela koji je svojevrsni temelj. Potom se uvodi transformacija kojom je omogućen prijelaz iz troosnog sustava u dvoosni sustav, tzv. Parkova transformacija. Definiranom transformacijom se troosni model motora transformirao u dvoosni model, koji služio kao temelj za parametiranje regulatora za vektorsko upravljanje. Detaljno je izveden unutarnji regulacijski krug za regulaciju struje. Regulator je izveden kao PI regulator parametriran prema tehničkom optimumu. Nakon parametriranja unutarnjeg regulacijskog kruga prelazi se na vanjski regulacijski krug također izveden PI regulatorom, ali je ovaj regulator parametriran po simetričnom optimumu (zbog integralnog djelovanja u mehaničkoj jednadžbi motora). Uvedena su ograničenja po pitanju izlaznih vrijednosti iz regulatora, a uvođenjem toga se mora i uračunati efekt zaleta. Nakon istaživanja mogućih metoda za poništavanje efekta zaleta odabrana je *tracking backing calculation* metoda te je implementirana u model regulatora.

Oba matematička modela, motora i regulatora, su implementirani u MATLAB/Simulink programski paket. Definiranjem osnovnih pojmova poput kosimulacije, HIL testiranja, stvarnovremenske simulacije i sl. se postavljaju temelji za kosimulaciju simulatora MicroLabBox i OP5031. Opisan je rad OP5031 i OP5600 simulatora (OP5600 je proširanje OP5031 simulatora), najprije izrada modela u Simulinku gdje su objašnjene postavke te skaliranje signala koji su poslani na/iz izlazno/ulazne priključke. Potom je objašnjen postupak kompajliranja matematičkog modela iz Simulinka u C kod. Opisan je postupak rada na RT-Lab programu. Detaljno je opisano snimanje podataka i naještanje HMI sučelja preko LabVIEW-a. Prelaskom na opis dSpace-ovog MicroLabBox-a, kao i kod opisa OPAL-ovih simulatora, najprije je opisan rad u simulinku uz postavke i komunikacijske blokova uz skaliranje. Također je opisan postupak kompajliranja matematičkog modela u C kod, nakon čega je opisan rad u Control Desk programu. Detaljno je opisan postupak izrade HMI sučelja te snimanje podataka. Kako je u trenutku pisanja ovog diplomskog rada dSpace-ov MicroLabBox bio novi simulator na fakultetu proveden je eksperiment validacije simulatora kako bi se utvrdilo šalje li signale u stvarnom vremenu. Matematički model koji je upotrijebljen za validaciju je sa skalarno upravljanje sinkronim motorom s permanentnim magnetima. MicroLabBox je signale proslijedio osciloskopu te na ja Control Desku napravljeno HMI sučelje sa snimanjem signala za usporedbu s osciloskopom.

Nakon validiranja se MicroLabBox i OP5031 povezuju u kosimulaciju, a budući da su povezani preko ulazno/izlaznih sučelja te postoji stvarna izmjena podatak povezani su u HIL sustav. Kosimulacijski eksperiment se sastoji od zaleta motora do nazivne brzine, potom terećenje, pa reverziranje do negativne nazivne brzine zatim rasterćenje i zaustavljanje. Za kraj je povedena analiza snimljenih signala.

Kako je u ovom diplomskom radu prilikom eksperimenta brzina stroja bila ograničena na nazivnu, daljnji razvoj bi uključivao i upravljanje strojem pri brzinama većim od nazivne te u tom slučaju referentna vrijednost struje  $d$  komponente treba biti negativna. Također je sljedeći korak u razvoju regulatora PHIL simulacija (eng. *power-hardware-in-the-loop*, *PHIL*), ali je za tu vrstu simulacije potreban sklop energetske elektronike te se radi s visokim iznosima napona.

## LITERATURA

- [1] B.K. Bose. *Modern Power Electronics and AC Drives*. Prentice Hall PTR, 2002.
- [2] C. M. Ong. *Dinamic simulation of electric machinery*, 1998. Prentice Hall PTR ,1998.
- [3] R. Krishnan. *Permanent Magnet Synchronous and Brushless DC Motor Drives*. CRC Press, 2010.
- [4] M. Harakawa, H. Yamasaki, T. Nagano, S. Abourida, C. Dufour, and J. Bélanger. *Real-time simulation of a complete pmsm drive at 10  $\mu$ s time step*. International Power Electronics Conference, 2005.
- [5] G. Luo, K. Song, W. Liu. *Permanent magnet synchronous motor field oriented control and hil simulation*. In em 2008 IEEE Vehicle Power and Propulsion Conference, 2008.
- [6] Eisenhawer de M. Fernandes , Denis R. Huller , Alexandre C. Oliveira , Welflen. R. N. Santos *Real-time simulator of interior permanent-magnet synchronous motor based on fpga devices*. In 2017 Brazilian Power Electronics Conference (COBEP), pages 1-6, 2017.
- [7] P. C. Krause, O. Wasynczuk, S. Sudhoff, and S. Pekarek. *Analysis Electric Machinery*. John Wiley & Sons, 2013.
- [8] Jan A. Melkebeek. *Electrical Machines and Drives*. Springer International Publishing, 2018.
- [9] J. Pyrhonen, T. Jokinen, and V. Hrabovcova. *Design of rotating electrical machines*. John Wiley & Sons, Ltd, 2008.
- [10] B: S. Guru and H. R. Hiziroglu. *Electric machinery and transformers*. Oxford University Press, 2001.
- [11] R. H. Park. *Two-reaction theory of synchronous machines generalized method of analysis-part i*. *Transactions of the American Institute of Electrical Engineers*, 48(3):716–727, jul 1929.
- [12] T. Benšić. *Analiza dinamičkih stanja sustava vlastite potrošnje hidroelektrane*. Diplomski rad, Sveučilište Josipa Jurja Strossmayera u Osijeku, 2014.
- [13] N. Perić. *Automatsko upravljanje, predavanja*. Fakultet elektrotehnike i računarstva Zagreb, 2004.
- [14] Y. Ishida K. Sakai. A design of an improved anti-windup control using a PI controller based on a pole placement method. *International Journal of Simulation: Systems, Science & Technology*, jan 2017.
- [15] H. B. Shin, X. L. Li, J. G. Park. *Comparison and evaluation of anti-windup pi controllers*. *Journal of Power Electronics*. Jan, 2011.
- [16] S. Sadjina, L. T. Kyllingstad, M. Rindarøy, S. Skjong, V. Æsøy, and E. Pedersen. *Distributed co-simulation of maritime systems and operations*. *Journal of Offshore Mechanics and Arctic Engineering*, 141(1), sep 2018.

- [17] P. Venne, J. Bélanger, and J. N. Paquin. *The what , where and why of real-time simulation*. OPAL-RT technologies, 2010.
- [18] P.J. Mosterman K. Popovici. *Real-Time Simulation Technologies: Principles, Methodologies, and Applications*. Taylor & Francis Ltd., 2017.
- [19] J. Nibert, M. Herniter, and Z. Chambers. *Model-based system design for MIL, SIL, and HIL*. World Electric Vehicle Journal, 5(4), 2012.
- [20] *OP5600 HILBOX USER GUIDE*, 2011.
- [21] *Dspace MicroLabBox Hardware Installation and Configuration*, 2021.

## SAŽETAK

U diplomskom radu je najprije opisan troosni model sinkronog motora s permanentnim magnetima. Potom je objašnjena dvoosna teorija kojom je omogućće prijelaz iz troosnog sustavu u dvoosni sustav, tzv. Park-ova transformacija. Nakon transformiranja troosnog modela sinkronog stroja s permanentnim magnetima u dvoosni sustav objašnjena je teorija vektorskog upravljanja motorom. Detaljno je izvedeno parametriranje PI regulatora struja i brzine. Također je opisan pojam efekta zaleta te je predloženo rješenje za poništavanje tog efekta. Modeli motora i regulatora za vektorsko upravljanje su implementirani u MATLAB/Simulinku. Potom su definirani osnovni pojmovi vezani za kosimulaciju, HIL testiranje. Detaljno je opisan rad dSpace-ovog simulatora MicroLabBox te je proveden eksperiment validacije. Također je detaljno opisan rad OPAL-ovih simulatora OP5031 i OP5600. Simulatori su se povezali u kosimulaciju te je proveden eksperiment testiranja vektorskog upravljanje, nakon čega je provedena analiza rezultata.

**Ključne riječi:** dSpace, kosimulacija, OPAL, sinkroni motor s permanentnim magnetima, vektorsko upravljanje

## ABSTRACT

### **Co-simulation analysis of permanent magnet synchronous motor with enhanced reluctance in real time**

This graduate thesis firstly describes a three-phase model of a synchronous motor with permanent magnets. Then the two-phase theory is explained, which enabled the transition from a three-phase system to a two-phase system, the so-called Park's transformation. After transforming the three-phase model of a synchronous machine with permanent magnets into a two-phase system, the theory of vector motor control is explained. The parameterization of the PI current and speed regulator is carried out in detail. The concept of the windup effect is also described and a solution for canceling this effect is proposed. Models of motor and controller are implemented in MATLAB/Simulink. Then the basic terms related to co-simulation, HIL testing are defined. The operation of dSpace's MicroLabBox simulator is described in detail and a validation experiment was performed. The operation of OPAL simulators OP5031 and OP5600 is also described in detail. The simulators were assembled in co-simulation and an experiment was conducted to test the vector control, after which the results were analyzed.

**Keywords:** co-simulation, dSpace, OPAL, permanent magnet synchronous motor, vector control

## ŽIVOTOPIS

Kristian Knol rođen je 11. svibnja 1998. godine u Osijeku. Završetkom osnovne škole Tenja 2013. godine upisuje III gimnaziju Osijek koji završava 2017. godine kada upisuje preddiplomski studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Tokom prediplomskog studija dobiva nagradu za uspješnost u studiranju. 2020. godine upisuje diplomski smijer industrijske elektroenergetike. Stručnu praksu na petoj godini odrađuje u tvrtki Danieli Systec u Osijeku. Tokom diplomskog studija dobiva nagradu za uspješnost u studiranju. Diplomski studij završava 2022. godine.