

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STROJNÍ



DIPLOMOVÁ PRÁCE

Řízení synchronního servomotoru pomocí
mikroprocesorové platformy Texas Instruments

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kuřina** Jméno: **Jan** Osobní číslo: **439005**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Strojní inženýrství**
Studijní obor: **Mechatronika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Řízení synchronního servomotoru pomocí mikroprocesorové platformy Texas Instruments

Název diplomové práce anglicky:

Control of synchronous servomotor with microprocessor platform Texas Instruments

Pokyny pro vypracování:

1. Proveďte rešerši na téma řízení synchronních elektrických pohonů se zaměřením na využití observeru pro získání rychlostního signálu, případně dalších stavových veličin.
2. Sestavte simulační model vektorového řízení motoru
3. Implementujte řízení reálného motoru s využitím platformy Instaspín
4. Zhodnoťte kvalitu rychlostního signálu získaného z kvadratického a Instaspín observeru
5. Kriticky zhodnoťte dosažené výsledky

Seznam doporučené literatury:

- [1] R. Doncker, D.W.J. Pulle, A.Veltman: Advanced Electrical Drives, Springer Netherlands, 2011
- [2] G. Ellis: Control System Design Guide, Elsevier, 2012, 4th Edition
- [3] Valášek, M. a kol. Mechatronika, Praha : ČVUT 1995

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Martin Nečas, MSc., Ph.D., odbor mechaniky a mechatroniky FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **29.04.2019**

Termín odevzdání diplomové práce: **16.08.2019**

Platnost zadání diplomové práce: _____

Ing. Martin Nečas, MSc., Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Milan Růžička, CSc.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

_____ Datum převzetí zadání

_____ Podpis studenta

Anotační list

Jméno autora:	Bc. Jan Kuřina
Název diplomové práce:	Řízení synchronního servomotoru pomocí mikroprocesorové platformy Texas Instruments
Anglický název:	Control of synchronous servomotor with microprocessor platform Texas Instruments
Akademický rok:	2018/2019
Obor studia:	Mechatronika
Ústav/odbor:	Ústav mechaniky, biomechaniky a mechatroniky Odbor Mechaniky a mechatroniky
Vedoucí diplomové práce:	Ing. Martin Nečas, MSc., Ph.D.
Bibliografické údaje:	Počet stran: 65 Počet obrázků: 43 Počet příloh: 1
Klíčová slova:	Vektorové řízení, synchronní motor s permanentními magnety, Clarkové transformace, Parkova transformace, PMSM, bezsenzorové řízení
Keywords:	Vector control, synchronous motor with permanent magnets, Clark transformation, Park transformation, PMSM, sensorless control
Anotace:	Tato práce se zabývá problematikou vektorového řízení, jakožto jednou z nejpokročilejších metod řízení elektrických pohonů. V práci je popsán princip vektorového řízení synchronního stroje a princip získání rychlostního signálu pro řídicí algoritmus. V praktické části bylo realizováno vektorové řízení servomotoru na vývojové desce Texas Instruments a také porovnání signálu rychlosti, získaného z enkodéru a ze softwarového pozorovatele.
Abstract:	This thesis deals with problematics of field oriented control, as one of the most advanced ways how electric drives are controlled. In this thesis, there is a description of how field oriented control of synchronous engine works and principle of acquirement of speed signal for control algorithm. In the practical part a field oriented control of servomotor with Texas Instruments development board was achieved as well as comparison of speed signals acquired from quadrature encoder unit and from software observer.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a použil pouze podklady uvedené v příloženém seznamu literatury.

V Praze dne _____

podpis

Poděkování

Rád bych poděkoval vedoucímu své diplomové práce Ing. Martinu Nečasovi, MSc., Ph.D. za cenné rady a vstřícnost při řešení problémů, které se během plnění cílů diplomové práce objevily a také za poskytnutí potřebných součástí a zázemí v laboratoři. Také děkuji rodině a svým blízkým za podporu, pochopení a motivaci.

Obsah

1 Úvod	9
1.1 Cíle diplomové práce	10
1.2 Definice základních pojmů a zkratek	10
2 Teorie vektorového řízení synchronních elektrických pohonů s permanentními magnety	11
2.1 Skalární řízení	11
2.1.1 Oblast konstantního momentu	11
2.1.2 Oblast konstantního výkonu	11
2.2 Vektorové řízení - Field Oriented Control	12
2.2.1 Úvod	12
2.2.2 Pomocné transformace FOC algoritmu	12
2.2.3 Matematický model PMSM	14
2.2.4 Princip vektorového řízení	17
2.2.5 Funkční schéma FOC řízení	19
2.3 Získávání polohy a rychlosti motoru	19
2.3.1 Druhy bezsenzorových metod získávání polohy	20
2.3.2 Výhody metod bezsenzorového řízení	22
2.4 Simulační model vektorového řízení motoru	22
2.4.1 Ladění parametrů a průběhy veličin	23
3 Praktická část	30
3.1 Použité vybavení	30
3.1.1 Hardware	30
3.1.2 Software	33
3.2 Realizace řízení motoru	34
3.2.1 Identifikace parametrů motoru	34
3.2.2 Schéma technologie InstaSPIN a FAST	36
3.2.3 Řízení momentu	37
3.2.4 Ladění proudové regulační smyčky	38
3.2.5 Ladění rychlostní regulační smyčky	40
3.3 Porovnání rychlostních signálů	43
3.3.1 Kvadrurní enkodér	43
3.3.2 Testovací sestava	44
3.3.3 Zpracování signálu z kvadrurního enkodéru	44
3.3.4 Průběh srovnání	49
3.3.5 Závěr srovnání	61
4 Závěr	62
Seznam literatury	63
Seznam příloh na CD	65

Seznam obrázků

2.1	Transformace mezi systémy - převzato z [15]	12
2.2	Clarkové transformace - převzato z [15]	13
2.3	Parkova transformace - převzato z [15]	14
2.4	Úhel Θ mezi souřadnicovým systémem rotoru a statoru	15
2.5	Výsledný vektor proudu	18
2.6	Odchylka proudů - korekce	18
2.7	Funkční schéma FOC - převzato z http://www.olliv.eu	19
2.8	Model vektorového řízení	24
2.9	Model PMSM	25
2.10	Průběhy veličin při přechodovém ději	26
2.11	Detail průběhů veličin při přechodovém ději	27
2.12	Průběhy veličin při přechodovém ději 2	28
2.13	Průběhy veličin po ustálení	29
3.1	Servomotor SGMM-A1S312 - převzato z https://s3.amazonaws.com	30
3.2	Závislost momentu a otáček motoru (A - provozní stav, B - stav ve špičce) - převzato z [8]	31
3.3	Vývojová deska LAUNCHXL-F28069M C2000 - převzato z http://edgate.in	31
3.4	Vývojová deska LAUNCHXL-F28027F C2000 - převzato z http://www.tme.eu	32
3.5	BoosterPack BOOSTXL-DRV8305EVM - převzato z http://www.tme.eu	32
3.6	Spínaný napájecí zdroj - převzato z https://www.tme.eu	33
3.7	Funkční schéma technologie InstaSPIN a FAST - převzato z [14]	37
3.8	Funkční schéma - řízení momentu - převzato z [14]	38
3.9	Napojení proudového PI regulátoru na stator - převzato z [14]	39
3.10	Schéma paralelního PI regulátoru rychlosti - převzato z [14]	41
3.11	Kaskádové uspořádání rychlostní a proudové regulační smyčky a filtru - převzato z [14]	41
3.12	Signály z jednotlivých kanálů enkodéru, výsledný signál - převzato z [19]	43
3.13	Testovací sestava	44
3.14	Model v Simulinku	45
3.15	Subsystém Delay 10	45
3.16	Frekvenční spektrum rychlostního signálu při 1000 RPM	46
3.17	Nastavení filtru pro pracovní oblast okolo 1000 RPM	47
3.18	Subsystém Filtrace	47
3.19	Subsystém Offset	47
3.20	Srovnání přístupů 1	48
3.21	Srovnání přístupů 2	49
3.22	Průběhy: 200 RPM	50
3.23	Průběhy: 1000 RPM	51
3.24	Průběhy: 3000 RPM	52
3.25	Průběhy: 5000 RPM	53
3.26	Průběhy: Rampa 1	54
3.27	Detail: Rampa 1	55
3.28	Průběhy: Rampa 2	56
3.29	Detail: Rampa 2	56
3.30	Průběhy: Sekvence 1	57

3.31	Detail: Sekvence 1	58
3.32	Průběhy: Sekvence 2	59
3.33	Detail: Sekvence 2	59
3.34	Průběhy: Sekvence 3	60

1. Úvod

Elektrické pohony jsou dnes využívány v téměř všech odvětvích průmyslu, kde pohánějí různorodé stroje s odlišnými požadavky na výkon, pracovní charakteristiku, přesnost či zástavbový prostor. Pro většinu těchto pohonů se stále častěji volí stroje střídavé a to kvůli jejich jednoduché konstrukci a s tím spojené nenáročnosti na údržbu, vyšší účinnosti a také možnosti použití v náročnějších podmínkách. Stejnoseměrné motory jsou tedy postupně nahrazovány motory střídavými a to i v náročnějších aplikacích, které mají vyšší nároky na přesnost řízení. Na druhou stranu obtížnější řízení těchto motorů vyžaduje výkonné řídicí jednotky, které budou spolu s frekvenčními měniči realizovat pokročilé řízení střídavých motorů.

Dříve byly nejčastějším střídavým pohonem motory asynchronní a to hlavně kvůli velkému výkonu. Tyto motory byly a jsou používány v těžkém průmyslu, u kolejových vozidel a podobných aplikací. Méně často se objevují v těchto aplikacích také synchronní motory a na vzestupu je synchronní typ motoru s permanentními magnety. Nově vyráběné permanentní magnety ze slitin kovů vzácných zemin dosahují vyšší magnetické indukce, než běžný typ permanentního magnetu z feritu a díky tomu lze produkovat menší a kompaktnější synchronní motory než motory asynchronní či stejnosměrné při uvažování stejného výkonu. Tento fakt je jeden z důležitých parametrů servopohonů. [5]

Synchronní pohony s permanentními magnety se částečně dostávají i do drážních aplikací, kdy díky menším rozměrům mohou být zabudovány přímo do kola nápravy vozidla díky čemuž dojde k ušetření místa kolem nápravy. Toto řešení se označuje jako nábojový pohon a je hojně využíváno např. u nízkopodlažních tramvají. [5]

Metody řízení synchronních pohonů se stále vyvíjejí a dříve používané skalární řízení dnes v náročnějších aplikacích nahradilo vektorové řízení, jehož vývoj a zdokonalování je stále aktuální. Tento typ řízení ke své funkčnosti potřebuje snímač polohy (rychlosti) a ten se dříve získával převážně z mechanického snímače, který musel být přímo spojen s rotorem motoru. Tato skutečnost zvyšovala finanční náročnost řešení aplikací s použitím synchronního motoru s permanentními magnety, zvětšovala zástavbový prostor motoru a také nároky na údržbu tohoto snímače. Proto dalším stupněm ve vývoji vektorového řízení je vypuštění mechanického snímače a nahrazení jeho funkce softwarovým algoritmem.

Problematika získání signálu polohy a rychlosti softwarově a základní princip vektorového řízení synchronního motoru s permanentními magnety budou v této práci představeny a také demonstrovány na reálné sestavě řídicí jednotky a servomotoru.

1.1 Cíle diplomové práce

1. Proveďte rešerši na téma řízení synchronních elektrických pohonů se zaměřením na využití observeru pro získání rychlostního signálu, případně dalších stavových veličin.
2. Sestavte simulační model vektorového řízení motoru.
3. Implementujte řízení reálného motoru s využitím platformy Instaspín.
4. Zhodnoťte kvalitu rychlostního signálu získaného z kvadrurního enkodéru a Instaspín observeru.
5. Kriticky zhodnoťte dosažené výsledky.

1.2 Definice základních pojmů a zkratk

- **Sensorless control**

Sensorless control, nebo-li bezsenzorové řízení nevyužívá mechanický snímač polohy a aktuální poloha rotoru se získává ze softwarového algoritmu.

- **Sensored control**

Sensored control znamená řízení s využitím snímače polohy, často například kvadrurního enkodéru, který musí být mechanicky spojen s rotorem motoru a jehož signál se přivádí do řídicího algoritmu.

- **Pól pár**

Počet pólových párů, tedy počet cívek každé fáze. Tato mechanická vlastnost motoru má vliv na výpočet rychlosti otáčení.

- **PI regulátor**

Regulátor, který obsahuje proporcionální a integrační složku.

- **PMSM**

Permanent Magnet Synchronous Motor - synchronní motor s permanentními magnety. Často užívaný termín v cizojazyčné literatuře.

- **PWM**

Pulse Width Modulation - pulzně šířková modulace.

- **ePWM**

enhanced Pulse Width Modulator - modul pro generování pulzně šířkové modulace.

- **eQEP**

enhanced Quadrature Encoder Pulse - modul pro zpracování signálu z kvadrurního enkodéru.

- **FOC**

Field Orientation Control - metoda vektorového řízení.

- **SVM**

Space Vector Modulation - prostorově vektorová modulace, efektivnější modifikace PWM.

- **EMF**

Electromotive Force - elektromotorická síla.

2. Teorie vektorového řízení synchronních elektrických pohonů s permanentními magnety

2.1 Skalární řízení

Přestože se tato práce věnuje zejména vektorovému řízení synchronních motorů, skalární řízení je ještě stále v mnoha méně náročnějších aplikacích využíváno a proto bude krátce zmíněno. Frekvenčním řízením střídavých motorů bylo dosaženo téměř vlastností motorů stejnosměrných a očekává se stále větší rozšiřování jejich uplatnění z důvodu jednodušší konstrukce, absence komutátoru a nižších požadavků na údržbu.

Základní myšlenka skalárního řízení synchronního motoru je řízení synchronní rychlosti pomocí změny frekvence napájecího napětí statoru. Spolu se změnou frekvence napětí se však musí také měnit velikost napětí, aby poměr napětí a frekvence zůstal konstantní a nenastávala změna magnetického toku motorem. Tato jednoduchá metoda se vyznačuje schopností provozu bez snímače aktuální polohy či rychlosti rotoru. [4]

Skalární řízení tedy umožňuje jednoduše řídit otáčky střídavých motorů, ale zajišťuje požadovaný chod motoru pouze v ustáleném stavu. Tento způsob řízení totiž nezajistí přechod mezi dvěma rychlostmi přesně dle požadavků uživatele, jelikož pokročilé dynamické chování systému není možné tímto řízením dosáhnout. Nejčastěji rozlišujeme mezi dvěma způsoby řízení:

- Oblast konstantního momentu - řízení při konstantním statorovém toku ψ
- Oblast konstantního výkonu - řízení při konstantním statorovém napětí U

2.1.1 Oblast konstantního momentu

Pro tuto oblast platí $0 < f < f_0$, $\psi = konst.$, $I = konst.$, kde f_0 je frekvence, pro kterou byl motor navržen, ψ je magnetický tok státorem a I je proud procházející státorem. Řídící obvody zde tedy upravují napětí U podle frekvence f tak, aby bylo buzení motoru konstantní. Toto řízení se také nazývá frekvenčně-napěťové buzení. Činnost v oblasti konstantního momentu odpovídá stejnosměrnému stroji s konstantním buzením, který je řízen napětím na kotvě. [3]

2.1.2 Oblast konstantního výkonu

Pro tuto oblast platí $f > f_0$, $I = konst.$, $U = konst.$. Jelikož je napětí U možné zvyšovat pouze do určité hranice, po jejím dosažení již musí být jeho hodnota konstantní. Jestliže se však zvyšuje frekvence f a napětí U zůstává konstantní, musí dojít ke zmenšení velikosti magnetického toku ψ a spolu s tím se zmenší také moment vyvíjený motorem. Konstantní tedy zůstane pouze výkon, podle čeho je také tato oblast pojmenována. Tato oblast je obdobou stejnosměrného stroje s konstantním napětím na kotvě, který je řízen budícím magnetickým polem. [3]

2.2 Vektorové řízení - Field Oriented Control

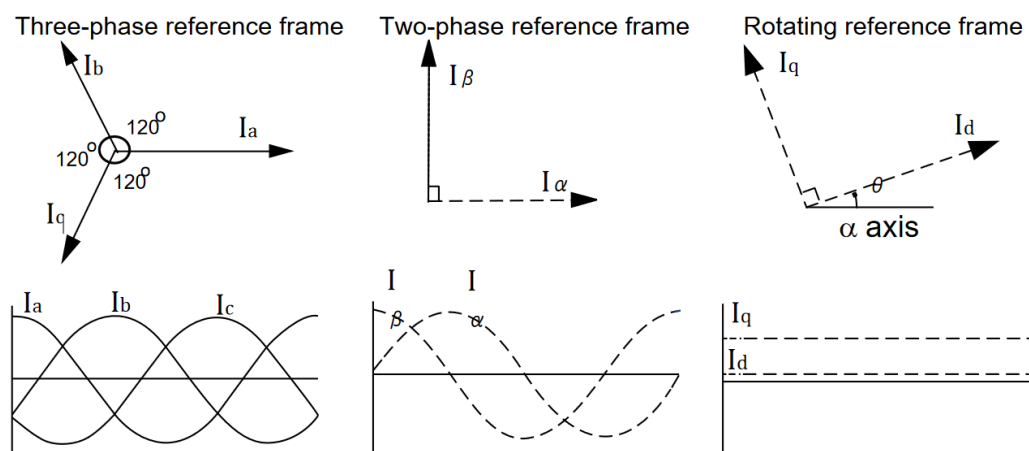
2.2.1 Úvod

Vektorové řízení, nebo-li Field Oriented Control, je v současné době nejpokročilejší metodou řízení střídavých motorů. Je možné zajistit optimální chod jak během, tak po skončení přechodného děje motoru. Jsme tedy schopni úspěšně řídit i dynamické chování systému.

Cílem vektorového řízení je od celkového statorového proudu separovat tokotvornou složku a momentotvornou složku. Odděleným řízením hodnot těchto dvou složek je možné získat maximální poměr momentu ku spotřebovanému proudu a tím minimalizovat celkovou spotřebu proudu k dosažení určitého momentu. Vektorové řízení tedy poskytuje kromě možnosti řídit dynamické chování systému také vysokou efektivitu využívání energie.

2.2.2 Pomocné transformace FOC algoritmu

Časově proměnné indukčnosti statoru a rotoru komplikují řízení dynamického chování motoru. Pro snížení náročnosti výpočtů řídicího algoritmu můžeme transformovat veličiny z třífázového systému do dvoudimenzionálního ortogonálního systému použitím transformace souřadnic. Tato transformace veličin tří fází do dvoudimenzionálního ortogonálního rotujícího referenčního systému se provede ve dvou krocích. Transformace ze souřadného systému tří fází statoru a, b a c do stacionárního systému $\alpha\beta$ je popsána Clarkovou transformací. Transformace ze stacionárního systému $\alpha\beta$ do rotujícího systému dq je popsána rotační Parkovou transformací. Znázornění transformace proudu ze systému abc do systému $\alpha\beta$ pomocí Clarkové transformace a poté do rotujícího systému dq pomocí Parkovy transformace je spolu s průběhy veličin na obrázku 2.1. [15]



Obrázek 2.1: Transformace mezi systémy - převzato z [15]

Clarkové transformace

Úkolem Clarkové transformace je převedení hodnot z třífázového systému do systému se dvěma ortogonálními osami (obr. 2.2). Matematický popis transformace vyjadřují rovnice 2.1 a 2.2.

$$I_{\alpha} = \frac{2}{3}(I_a) - \frac{1}{3}(I_b - I_c), \quad (2.1)$$

$$I_{\beta} = \frac{2}{\sqrt{3}}(I_b - I_c), \quad (2.2)$$

kde

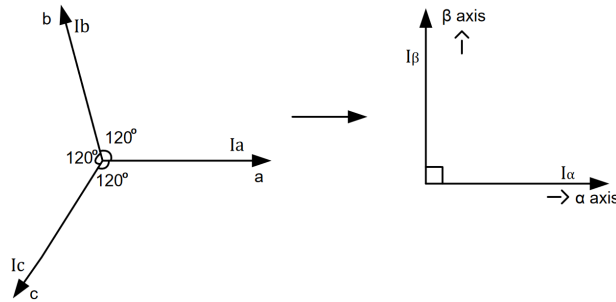
- I_a , I_b a I_c jsou veličiny vyjádřené v původním třífázovém systému,
- I_α a I_β jsou veličiny vyjádřené ve stacionárním ortogonálním dvoudimenzionálním systému.

Transformaci lze ještě dále zjednodušit, pokud využijeme principu superpozice a skutečnosti, že součet proudů v uzlu musí být roven nule, tedy $I_a + I_b + I_c = 0$.

Poté můžeme psát:

$$I_\alpha = I_a, \quad (2.3)$$

$$I_\beta = \frac{1}{\sqrt{3}}(I_a + 2I_b). \quad (2.4)$$



Obrázek 2.2: Clarkové transformace - převzato z [15]

Inverzní Clarkové transformace

Inverzní Clarkové transformace převádí veličiny ze stacionárního systému dvou os do systému se třemi fázemi. Platí pro ni vztahy 2.5, 2.6 a 2.7.

$$I_a = I_\alpha, \quad (2.5)$$

$$I_b = \frac{-I_\alpha + \sqrt{3} \cdot I_\beta}{2}, \quad (2.6)$$

$$I_c = \frac{-I_\alpha - \sqrt{3} \cdot I_\beta}{2}. \quad (2.7)$$

Parkova transformace

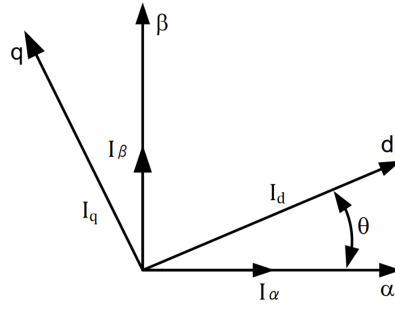
Parkova transformace (obr. 2.3) se používá pro transformaci ze stacionárního ortogonálního systému $\alpha\beta$ do rotujícího systému rotoru dq a platí pro ni rovnice 2.8 a 2.9.

$$I_d = I_\alpha \cdot \cos(\Theta) + I_\beta \cdot \sin(\Theta), \quad (2.8)$$

$$I_q = -I_\alpha \cdot \sin(\Theta) + I_\beta \cdot \cos(\Theta), \quad (2.9)$$

kde

- I_α a I_β jsou veličiny vyjádřené v původním stacionárním systému,
- I_d a I_q jsou veličiny vyjádřené v rotujícím systému rotoru,
- Θ je úhel natočení rotoru.



Obrázek 2.3: Parkova transformace - převzato z [15]

Inverzní Parkova transformace

Inverzní Parkova transformace převádí veličiny z rotujícího systému rotoru do stacionárního systému. Pro transformaci platí rovnice 2.10 a 2.11.

$$I_{\alpha} = I_d \cdot \cos(\Theta) - I_q \cdot \sin(\Theta), \quad (2.10)$$

$$I_{\beta} = I_d \cdot \sin(\Theta) + I_q \cdot \cos(\Theta). \quad (2.11)$$

2.2.3 Matematický model PMSM

Matematický model synchronního motoru s permanentními magnety může být popsán užitím teorie prostorových vektorů. Veličiny třífázového motoru jako napětí, proudy nebo magnetické toky jsou vyjádřeny ve formě komplexních prostorových vektorů. Takový model poskytuje výstup pro jakoukoli kombinaci vstupních napětí a proudů a to jak pro ustálený stav, tak pro dynamickou změnu. Komplexní prostorový vektor lze popsat užitím dvou ortogonálních os a třífázový motor lze pak uvažovat jako dvoufázový, což značně zjednodušuje počet rovnic a snižuje výpočetní náročnost řídicího algoritmu. [2]

Pod pojmem synchronního motoru s permanentními magnety rozumíme stroj s vinutími rozdělenými na 3 fáze, symetricky rozmístěnými po obvodu statoru a hladkou vzduchovou mezerou oddělené permanentní magnety na obvodu rotoru. Vztahy pro napětí na jednotlivých cívkách statoru můžeme poté popsat rovnicemi 2.12, 2.13 a 2.14. [2]

$$u_{sa} = R_s \cdot i_{sa} + \frac{d}{dt} \psi_a, \quad (2.12)$$

$$u_{sb} = R_s \cdot i_{sb} + \frac{d}{dt} \psi_b, \quad (2.13)$$

$$u_{sc} = R_s \cdot i_{sc} + \frac{d}{dt} \psi_c, \quad (2.14)$$

kde

- u_{sa} , u_{sb} a u_{sc} jsou hodnoty statorových napětí na fázích a , b a c ,
- i_{sa} , i_{sb} a i_{sc} jsou hodnoty statorových proudů na fázích a , b a c ,
- ψ_a , ψ_b a ψ_c jsou hodnoty celkových magnetických indukčních toků na fázích a , b a c ,
- R_s je odpor jednotlivých vinutí.

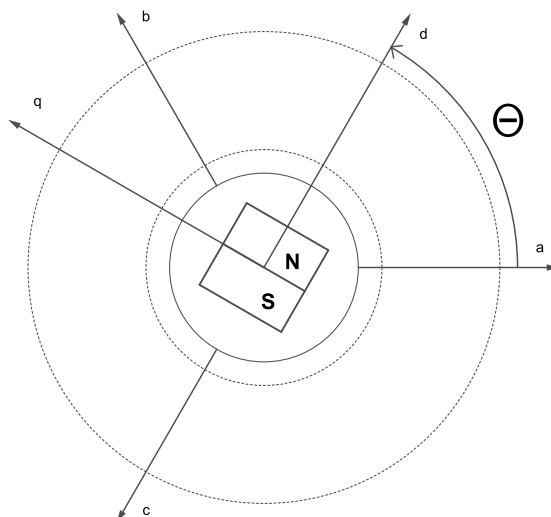
Na hodnoty celkových magnetických toků ve vinutích a , b a c mají vliv jednotlivé proudy procházející vinutími, indukčnosti vinutí a také permanentní magnety v rotoru. Maticová rovnice 2.15 popisuje tyto vztahy. [6]

$$\begin{bmatrix} \psi_{sa} \\ \psi_{sb} \\ \psi_{sc} \end{bmatrix} = \begin{bmatrix} L_a & L_{ab} & L_{ac} \\ L_{ba} & L_b & L_{bc} \\ L_{ca} & L_{cb} & L_c \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix}, \quad (2.15)$$

kde

- L_a , L_b a L_c jsou vlastní indukčnosti vinutí jednotlivých fází a , b a c ,
- L_{ab} , L_{ac} , L_{bc} a další jsou vzájemné indukčnosti mezi vinutími fází a , b a c ,
- ψ_{am} , ψ_{bm} a ψ_{cm} jsou hodnoty magnetických indukčních toků od permanentních magnetů na fázích a , b a c .

Maticové rovnice 2.16, 2.17 a 2.18 ukazují závislost zmíněných veličin na úhlu, který svírá vektor magnetické indukce rotoru a statoru. Tento úhel označíme jako Θ a je znázorněn na obr. 2.4.



Obrázek 2.4: Úhel Θ mezi souřadnicovým systémem rotoru a statoru

$$\begin{bmatrix} L_a \\ L_b \\ L_c \end{bmatrix} = L_s \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + L_m \begin{bmatrix} \cos(2\Theta) \\ \cos(2(\Theta - 2\pi/3)) \\ \cos(2(\Theta + 2\pi/3)) \end{bmatrix}, \quad (2.16)$$

$$\begin{bmatrix} L_{ab} \\ L_{bc} \\ L_{ca} \end{bmatrix} = \begin{bmatrix} L_{ba} \\ L_{cb} \\ L_{ac} \end{bmatrix} = -M_s \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - L_m \begin{bmatrix} \cos(2(\Theta + \pi/6)) \\ \cos(2(\Theta + \pi/6 - 2\pi/3)) \\ \cos(2(\Theta + \pi/6 + 2\pi/3)) \end{bmatrix}, \quad (2.17)$$

$$\begin{bmatrix} \psi_{am} \\ \psi_{bm} \\ \psi_{cm} \end{bmatrix} = \psi_m \begin{bmatrix} \cos(\Theta) \\ \cos(\Theta - 2\pi/3) \\ \cos(\Theta + 2\pi/3) \end{bmatrix}, \quad (2.18)$$

kde

- L_s je průměrná hodnota vlastní indukčnosti jednotlivých vinutí,

- L_m je amplituda měnící se hodnoty vlastní a vzájemné indukčnosti v závislosti na úhlu Θ ,
- M_s je průměrná hodnota vzájemné indukčnosti mezi vinutími a , b a c ,
- ψ_m je hodnota magnetického indukčního toku od permanentních magnetů.

Jelikož již známe potřebný matematický aparát z kapitoly 2.2.2, můžeme rovnice přepsat do systému $\alpha\beta$:

$$u_\alpha = Ri_\alpha + \frac{d}{dt}\psi_\alpha, \quad (2.19)$$

$$u_\beta = Ri_\beta + \frac{d}{dt}\psi_\beta. \quad (2.20)$$

Pro magnetické indukční toky v tomto systému platí:

$$\psi_\alpha = L_\alpha i_\alpha + \psi_m \cos(\Theta), \quad (2.21)$$

$$\psi_\beta = L_\beta i_\beta + \psi_m \sin(\Theta). \quad (2.22)$$

Po dosazení rovnic 2.21 a 2.22 do rovnic 2.19 a 2.20 dostaneme rovnice pro napětí:

$$u_\alpha = Ri_\alpha + L_\alpha \frac{di_\alpha}{dt} - \omega_e \psi_m \cos(\Theta), \quad (2.23)$$

$$u_\beta = Ri_\beta + L_\beta \frac{di_\beta}{dt} + \omega_e \psi_m \sin(\Theta), \quad (2.24)$$

kde ω_e je elektrická úhlová rychlost motoru a její vztah s mechanickou rychlostí ω_m je dán vztahem 2.25, přičemž P_p je počet pólových párů motoru. Je nutné si uvědomit, že odpor R a indukčnosti L_α a L_β jsou také přepočítány do nového souřadnicového systému. [17]

$$\omega_e = P_p \cdot \omega_m. \quad (2.25)$$

Elektromagnetický moment motoru T_e v $\alpha\beta$ souřadnicích je pak:

$$T_e = \frac{3}{2} \cdot P_p \cdot (\psi_\alpha \cdot i_\beta - \psi_\beta \cdot i_\alpha). \quad (2.26)$$

Pro úhlovou rychlost pak z pravidla momentové rovnováhy na hřídeli plyne vztah 2.27. [17]

$$\frac{d}{dt}\omega_e = \frac{P_p}{J} \left[\frac{3}{2} P_p (\psi_\alpha \cdot i_\beta - \psi_\beta \cdot i_\alpha) - T_L \right], \quad (2.27)$$

kde

- J je moment setrvačnosti,
- T_L je zátěžný moment.

Z předchozích rovnic jsme schopni realizovat model v souřadnicích $\alpha\beta$ a nyní zbývá model vyjádřit i v souřadnicích, které jsou funkcí natočení motoru Θ , v souřadnicích dq .

Pro napětí v systému dq platí:

$$u_d = Ri_d + \frac{d}{dt}\psi_d - \omega_e \psi_q, \quad (2.28)$$

$$u_q = Ri_q + \frac{d}{dt}\psi_q + \omega_e \psi_d, \quad (2.29)$$

kde i_d je tokotvorná složka proudu a i_q je momentotvorná složka proudu. Pro toky magnetické indukce platí:

$$\psi_d = L_d i_d + \psi_m, \quad (2.30)$$

$$\psi_q = L_q i_q. \quad (2.31)$$

Po dosazení rovnic 2.30 a 2.31 do rovnic 2.28 a 2.29 dostaneme výsledné napěťové rovnice v dq souřadnicích: [17]

$$u_d = Ri_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q, \quad (2.32)$$

$$u_q = Ri_q + L_q \frac{di_q}{dt} + \omega_e L_d i_d + \omega_e \psi_m. \quad (2.33)$$

Moment motoru T_e je pak dán rovnicí 2.34.

$$T_e = \frac{3}{2} P_p (\psi_d \cdot i_q - \psi_q \cdot i_d). \quad (2.34)$$

Jelikož u synchronních motorů s permanentními magnety bývá konstrukce ve většině případů (i u použitého reálného motoru) s hladkým rotorem, bývají indukčnosti L_q a L_d v na sebe kolmých směrech stejně velké, či zanedbatelně odlišné a lze tedy zjednodušeně psát:

$$T_e = \frac{3}{2} P_p [\psi_m + (L_d - L_q) i_d] i_q = \frac{3}{2} P_p \psi_m i_q. \quad (2.35)$$

Ze vztahu 2.35 je patrné, že moment je vytvářen momentotvornou složkou proudu i_q . Pro úhlovou elektrickou rychlost motoru v souřadnicích dq poté platí: [17]

$$\frac{d}{dt} \omega_e = \frac{P_p}{J} \left[\frac{3}{2} P_p \psi_m i_q - T_L \right]. \quad (2.36)$$

2.2.4 Princip vektorového řízení

Simulace ukázaly, že pokud vektor magnetické indukce statoru a vektor magnetické indukce rotoru svírají úhel $\delta = 0^\circ$, motor nevytváří žádný točivý moment. Opačný případ nastává při hodnotě $\delta = 90^\circ$, kde je hodnota vytvářeného točivého momentu maximalizována. Je to také limitní hodnota úhlu pro zachování stability. Pro nejvyšší efektivitu procesu je tedy výhodné udržovat tento úhel konstantní a to na nejefektivnější hodnotě $\delta = 90^\circ$.

Aby bylo možné vytvořit vektor magnetické indukce statoru kolmý na vektor magnetické indukce rotoru, musíme nejprve znát úhel Θ , tedy natočení rotoru vůči statoru. Metod získání tohoto úhlu je několik, buď pomocí mechanického snímače polohy, nebo pomocí pozorovatele, o čem bude v práci pojednáno později.

Máme-li již úhel Θ k dispozici, můžeme vytvářet ve vinutích statoru proud tak, aby byl udržován úhel δ mezi vektory 90° . Regulaci vyvíjeného momentu ovlivníme změnou amplitud třífázového proudu a rychlost otáčení ovlivníme frekvencí tohoto proudu.

Hodnota momentu zde závisí na počtu pólů motoru p , resp. počtu pólpárů P_p , konstantním magnetickým indukčním toku od permanentních magnetů v rotoru ψ_m a velikosti proudu i_q , tedy momentotvorné složky proudu (rovnice 2.37 ve tvaru s počtem pólů motoru oproti rovnici 2.35, kde se vyskytují pólpáry). Hlavní úloha řídicího algoritmu je tedy vytvářet pomocí proudů rotující magnetické pole ve statoru tak, aby byl udržován vektor magnetické indukce statoru kolmý k vektoru magnetické indukce rotoru. [1]

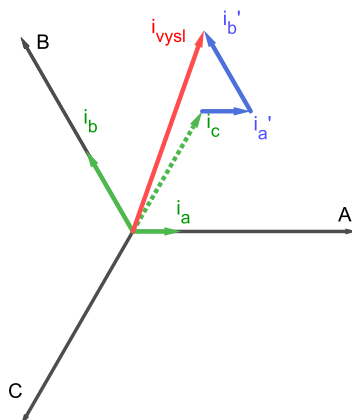
$$T_e = \frac{3}{4} p \psi_m i_q. \quad (2.37)$$

Princip řízení je přitom velmi podobný řízení stejnosměrného DC motoru, kde je vektorové řízení realizováno mechanicky skrze komutátor a kartáče. U střídavého motoru je princip podobný a pro realizaci řízení musí být podniknuty následující kroky: [1]

1. Změření proudu, který právě teď prochází motorem.
2. Porovnání měřeného proudu a požadovaného proudu a vytvoření odchylky.
3. Zesílení odchylky pomocí PI regulátoru a vytvoření korekce.

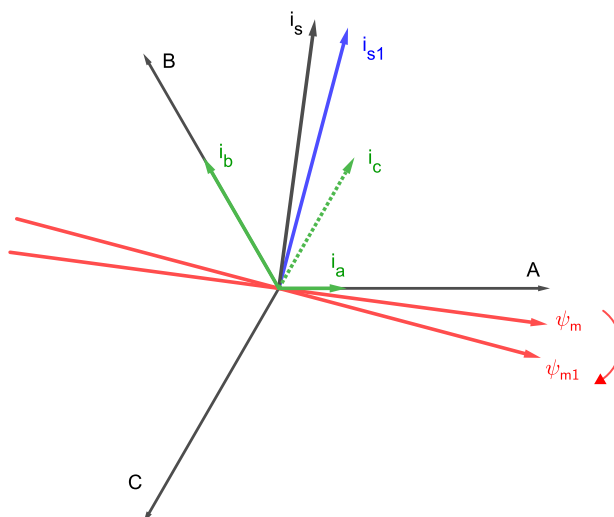
4. Převedení korekčního napětí pomocí PWM modulace.

První krok, tedy měření 3 proudů lze zjednodušit pomocí Kirchhoffova zákona, který říká, že součet proudů v uzlu se musí rovnat nule a lze tedy měřit pouze dva proudy a třetí dopočítávat. Z těchto tří proudů, jejichž orientaci známe, poté vytvoříme výsledný vektor znázorněný na obr. 2.5 červenou barvou.



Obrázek 2.5: Výsledný vektor proudu

Druhým krokem je zjištění odchylky vektorů proudu měřeného (černý vektor i_s na obr. 2.6) a proudu požadovaného (modrý vektor i_{s1} na obr. 2.6), abychom mohli aplikovat korekci a stále udržovali kolmý úhel s vektorem magnetické indukce rotoru, který se neustále mění (červené vektory ψ_m a ψ_{m1}). Třetí krok spočívá ve využití

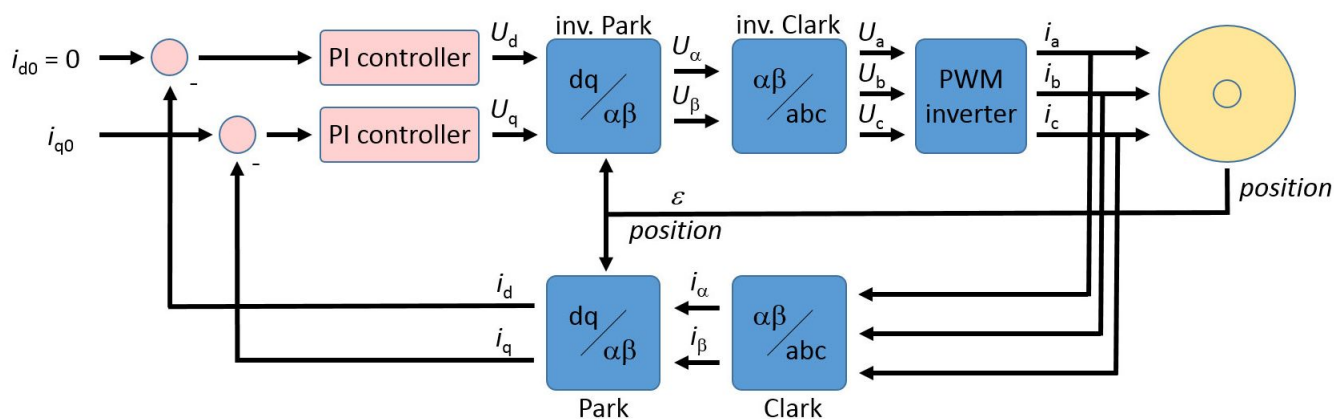


Obrázek 2.6: Odchylka proudů - korekce

dvou regulátorů s proporcionální a integrační složkou. Pouze dva regulátory jsou potřebné, jelikož se podařilo snížit počet veličin ze tří na dvě pomocí Clarkovy transformace, která převedla systém tří fází a , b a c na stacionární systém $\alpha\beta$. Výsledný vektor ve dvoudimenzionálním prostoru lze totiž definovat pouze pomocí dvou vektorů. Parkova transformace poté převádí veličiny ze stacionárního systému $\alpha\beta$, kde mají sinusový průběh, do rotujícího systému rotoru dq , kde jsou tyto veličiny konstantní funkcí. Fakt, že v systému dq , tedy rozdělení proudu na tokotvornou složku i_d a momentotvornou složku i_q , jsou průběhy těchto veličin konstantní značně

zjednodušuje výpočty řídicího algoritmu. Čtvrtý krok obstarává výsledný převod výstupů z PI regulátorů na signál pulzně šířkové modulace, který je poslán do výkonové elektroniky řízení, kde spíná výkonové tranzistory PWM modulu, který poskytuje napájení do jednotlivých vinutí a vytváří tak rotující magnetické pole.

2.2.5 Funkční schéma FOC řízení



Obrázek 2.7: Funkční schéma FOC - převzato z <http://www.oliw.eu>

Na obrázku 2.7 je základní funkční schéma algoritmu vektorového řízení, poskládané z jednotlivých bloků. Zleva do sumačních bloků pro výpočet odchylky měřeného a žádaného proudu, resp. jeho tokotvorné složky, kterou požadujeme rovnou nule $I_d = 0$ (to neplatí vždy, při odbuzování motoru může být žádána i nenulová hodnota z důvodu potlačení zpětné elektromagnetické síly) a momentotvorné složky I_q , která je hlavním ovládacím členem. Tyto odchylky jsou vstupem pro 2 PI regulátory, jejichž funkce je tyto odchylky zesílit tak, aby se vytvořil akční zásah a odchylky se co nejrychleji minimalizovaly. Následuje transformační blok inverzní Parkovy transformace, která z rotačního systému dq převede veličiny do stacionárního systému $\alpha\beta$. Pro tento převod je nutné znát pozici motoru ϵ , resp. úhel natočení rotoru Θ . Tento úhel je v použitém schématu získáván mechanickým snímačem polohy. Poté následuje transformační blok inverzní Clarkové transformace, jehož výstup jsou již tři veličiny pro tři fáze a , b a c . Tyto hodnoty se poté pomocí pulzně šířkové modulace převedou na spínací signály, které jsou posílány do výkonové elektroniky řízení. Zpětná vazba, tedy měření veličin přímo na motoru, je ve své původní podobě v systému třech fází a , b a c a proto je i na této cestě potřebné provést nejprve Clarkovou transformaci a poté Parkovu, aby mohly být výpočty korekcí co nejjednodušší, jelikož po transformacích se pracuje s konstantní funkcí, namísto se sinusovým průběhem. Další výhodou výpočtů s pozorovatelem otáčejícím se spolu s rotorem je ta, že nenastává problém ani při vysokých otáčkách, kdy by pozorovatel spjatý se státorem musel sbírat najednou velké množství informací o průběhu veličin a regulace by se tak stala velice obtížným úkolem.

2.3 Získávání polohy a rychlosti motoru

Jedním z klíčových úkolů při realizaci vektorového řízení je získání informace o aktuální poloze a rychlosti rotoru. Získání tohoto signálu může být realizováno skrze snímač, enkodér, který je mechanicky spojen s hřídelí motoru, ale takové řešení přináší složitější konstrukci a zvyšuje výrobní náklady. V této kapitole se tedy budeme zabývat ostatními způsoby, kterými lze tento signál získat a to výpočetními algoritmy využívajícími měřených veličin elektrických.

2.3.1 Druhy bezsensorových metod získávání polohy

Pod pojmem sensorless, nebo-li bezsensorové řízení, rozumíme eliminaci mechanických čidel polohy a rychlosti na rotoru, avšak stále je nutné některá měření zachovat, například měření proudu a napětí na statoru.

Bezsensorové metody řízení jsou založené na různých principech, avšak všechny je spojuje skutečnost, že nějakým způsobem zpracovávají měřená statorová napětí a také proudy, které statorovými vinutími procházejí. Základní rozdělení bezsensorových metod získávání polohy obsahuje tři kategorie: [10]

1. Umělá inteligence
2. Injekční metody
3. Metody, založené na matematickém modelu motoru

Umělá inteligence

Metody umělé inteligence jsou založené na bázi neuronových sítí a fuzzy logiky. Zatímco lidský mozek se skládá z mnohonásobně většího počtu neuronů, umělé neuronové sítě obsahují desítky až stovky neuronů. Princip trénování neuronových sítí pro toto uplatnění je podobný jako u jiných aplikací, kde je tato metoda použita. Neuronová síť si při trénování osvojuje vzorce chování daného systému pomocí sad vstupních a výstupních dat. Čím více těchto trénovacích dat poskytneme, tím bude výsledná reakce sítě přesnější. Výhodou neuronových sítí je schopnost se naučit chování z trénovacích dat a to si poté zapamatovat bez nutnosti použít klasickou paměť. Tato schopnost se ukrývá v pokročilé paralelní struktuře, která se nazývá asociativní paměť. Dalšími vlastnostmi neuronových sítí jsou nezávislost na výpadku menšího množství neuronů, robustnost při změně parametrů a přirozená eliminace šumu. [10]

Injekční metody

Injekční metody získávání informací o poloze rotoru využívají nesymetrie magnetického obvodu elektrického stroje. Tato vlastnost je většinou v matematických modelech zanedbávána. Čím větší je tato magnetická asymetrie, která zde znamená závislost indukce na poloze, tím lépe ji lze detekovat a poté využít k výpočtu polohy rotoru. Princip metody spočívá v aplikaci třífázového napětí o jiné frekvenci, než je základní frekvence stroje a toto napětí se označuje jako injektované napětí. Při aplikaci tohoto napětí vzniknou kromě základního proudu ještě další složky proudu a ty se poté po filtraci a demodulaci využijí k výpočtu polohy. Hlavní nevýhody těchto metod jsou přídavné ztráty, akustický hluk, zvlnění momentu a neuniverzálnost použití. Mezi výhody patří funkčnost i při velmi nízkých otáčkách a také schopnost určit polohu i při nulových otáčkách. [10]

Metody, založené na matematickém modelu motoru

Tyto metody vycházejí ze základního matematického modelu synchronního motoru s permanentními magnety, který nezná moment zatížení a musí ho dopočítávat z elektrických veličin. Matematický model je zjednodušený a zanedbává mnoho reálných vlastností elektrických strojů a proto jsou jeho použití a přesnost vždy omezené. Existuje více matematických modelů, přičemž každý model může systém popisovat tak, že se více blíží realitě pro různé účely a jiné pracovní oblasti. Matematické modely motorů slouží pro mnoho metod, například různé druhy pozorovatelů, Kalmanův filtr, či integrační metody.

Co se týče stavových veličin, pro výpočty polohy se používá především protinapětí a magnetických toků. Pro zjištění protinapětí se využívá pozorovatele a magnetické indukční toky se zjistí integračním přístupem. Jediná data, která jsou k dispozici, jsou statorové proudy a napětí pro tři fáze, jež jsou měřeny nejlépe přímo na svorkách stroje. Opět mohou být měřeny pouze dvě fáze a třetí hodnota dopočítávána. Tyto metody se označují jako adaptivní a využívají odchylky mezi referenčním modelem, což bývá většinou samotný elektrický stroj a matematickým modelem systému. Hodnoty z modelu, tedy pozorovatele systému jsou porovnávány se skutečně

naměřenými hodnotami a výsledné odchylky jsou používány k odhadu polohy a tedy k řízení. [10], [16], [20]
Stavový model PMS motoru lze vyjádřit jako

$$\dot{x} = Ax + Bu, \quad (2.38)$$

$$y = Cx. \quad (2.39)$$

kde

- x je sloupcový vektor stavových veličin,
- A je čtvercová matice dynamiky (stavu),
- u je sloupcový vektor vstupů,
- B je matice vstupu,
- y je sloupcový vektor výstupů,
- C je matice výstupu,
- Matice D , propojující vstupy přímo na výstupy se zde nevyskytuje.

Matice A a B jsou závislé na fyzikálních vlastnostech stroje, jako jsou např. indukčnosti a odpory ve vinutí. Napětí a proudy nemusí být, pro účely modelu, nutně vyjádřeny v soustavě rotoru a jako stavové veličiny zde mohou figurovat také indukovaná napětí. Matematický model pozorovatele lze vyjádřit podobně, ale je zde nutné zahrnout kompenzaci matice stavu a vstupu a přidat odhadovanou odchylku (rovnice 2.40 a 2.41). [16], [20]

$$\dot{\hat{x}} = A\hat{x} + Bu + K_e(y - \tilde{y}), \quad (2.40)$$

$$\tilde{y} = C\hat{x}. \quad (2.41)$$

Odhadovaný stav je tedy vyjádřen jako \hat{x} a odhadovaný výstup jako \tilde{y} . Matice K_e zde slouží jako faktor zesílení korekčního členu odchylky mezi měřeným a odhadovaným výstupem systému. Rozdíl mezi stavem systému a stavem pozorovatele lze vyjádřit rovnicí 2.42. [16], [20]

$$\dot{e} = Ax - A\hat{x} - K_e(Cx - C\hat{x}) = (A - K_eC)(x - \hat{x}). \quad (2.42)$$

Rovnice 2.42 se dá přepsat jako

$$\dot{e} = (A - K_eC)e, \quad (2.43)$$

kde

- e je odchylka pozorovatele.

Odchylka e závisí na vlastních číslech matice $(A - K_eC)$ a pokud je tato matice stabilní, bude se odchylka blížit k nule pro jakýkoli počáteční vektor odchylky $e(0)$. Způsob volby matice K_e tak, aby matice $(A - K_eC)$ měla požadovaný tvar lze dohledat v [16].

Základním dělením pozorovatelů je lze rozdělit na dva druhy:

- Pozorovatel plného řádu (Full order observer)
- Pozorovatel redukováného řádu (Reduced order observer)

Toto dělení závisí na tom, či je nutné pozorovat všechny stavové proměnné systému a tedy je nutné použít plný řád, nebo to není pro potřeby systému nutné a stačí sledovat pouze některé stavy a tedy využít pozorovatele menšího než plného řádu, který má ve srovnání s ním také menší nároky na výpočetní čas. [16]

Jiný typ dělení se řídí podle aplikace pozorovatele v řídicím schématu:

- Luenbergerův pozorovatel
- Pozorovatel redukováného řádu
- Pozorovatel v klouzavém režimu (Sliding Mode Observer), sledovací pozorovatel (Tracking Observer)
- Kalmanův filtr (Kalman Filter)

2.3.2 Výhody metod bezsensorového řízení

Mezi největší výhody užití bezsensorového řízení patří:[10]

- Zmenšení zástavbového prostoru elektrického stroje - zmenšení rozměrů a hmotnosti PMS motorů, které mají už z principu vysoký poměr výkonu k vlastní velikosti vynecháním snímače, který se většinou nachází za motorem a zvětšuje velikost motoru asi o jednu třetinu může usnadnit instalace v omezených prostorech.
- Snížení pořizovací ceny motoru - výpočet rychlosti stroje řídicím systémem klade důraz pouze na výkon tohoto řídicího systému, ty však pro tyto potřeby bývají dostačující v jejich standardním provedení a tedy dochází k finanční úspoře v řádu několika tisíců Kč na mechanických snímačích.
- Menší moment setrvačnosti rotoru - dynamika a možnosti její kontroly závisí na maximálním točivém momentu a momentu setrvačnosti stroje. Z tohoto důvodu tedy čím menší moment setrvačnosti stroj má, tím je dosahováno lepších dynamických vlastností motoru a mechanické snímače moment setrvačnosti zvětšují, což je nežádoucí.
- Vyšší robustnost a spolehlivost - čím méně částí je v systému použito, tím spolehlivější je systém, neboť může nastat méně poruch a například přerušení kontaktu se snímačem polohy by v případě klasického vektorového řízení vyřadilo celý systém.
- Vyšší přesnost měření rychlosti motoru - přesnost informace o poloze nebo rychlosti může být u bezsnímačových metod lepší a to zejména při nízkých otáčkách. Rychlost je dána počtem impulsů za určitý časový interval a pokud je při nízkých rychlostech frekvence těchto impulsů malá, klesá také s počtem měření i jeho přesnost. Při velmi nízkých otáčkách však dochází k nestabilitě některých bezsensorových metod získávání polohy rotoru. Toto neplatí u injekčních bezsensorových metod.
- Menší požadavky na údržbu - se snižujícím se počtem prvků se snižuje také pravděpodobnost poruchy a potřeba pravidelně kontrolovat správnou funkčnost těchto mechanických částí.

Naopak nevýhod bezsensorového řízení je mnohem méně a patří mezi ně například vyšší požadavky na výpočetní výkon řídicího systému, který ale pro potřeby klasického vektorového řízení bývá u běžně používaných mikroprocesorů více než dostatečný a tedy zbývá dostatek prostoru pro často složitý algoritmus výpočtu rychlosti a polohy. Bezsensorové metody s matematickým modelem mohou být při nízkých otáčkách nestabilní, nebo dokonce vůbec nefungují, ale PMS motory jsou většinou využívány ve vysokootáčkových režimech, takže tato nevýhoda nebývá pro většinu aplikací závažná. Naopak nevýhody injekčních metod, které předchozím problémem netrpí, spočívají ve zvýšeném hluku, zvlnění momentu a potřebě co nejvyšší změny indukčnosti se změnou úhlu natočení rotoru. Přesto jsou výhody bezsensorových metod zřejmě a jejich použití je stále častější, neboť i jejich vývoj jde stále kupředu. [10]

2.4 Simulační model vektorového řízení motoru

V této části práce bylo využito dostupného modelu vektorového řízení motoru v Simulinku z [21] a tento model byl přepracován a zjednodušen tak, aby odpovídal sestavě, která byla poté použita k praktickému experimentu.

Model, který je znázorněn na obrázku 2.8, je tvořen dvěma řídicími smyčkami, vnitřní smyčkou pro řízení momentu a vnější smyčkou pro řízení rychlosti. Model řízení obsahuje veškeré náležitosti vektorového řízení. Odchytky veličin jsou řídicím signálem pro PI regulátory a poté jsou výsledné regulované akční zásahy převáděny do požadovaných souřadnicových systémů pomocí Clarkové a Parkovy transformací a jejich inverzí. Pro získání polohy rotoru, potřebné pro Parkovy transformace, využívá model signál přímo vycházející z mechanického modelu PMSM motoru. Tento přístup simuluje přítomnost snímače polohy, tedy sensed řízení. Bezsniímačové řízení by znamenalo vytvořit spolehlivého pozorovatele systému, což je ale velmi sofistikovaný úkol, který není cílem této práce a proto bylo využito výše zmíněného sensed přístupu.

Na obrázku 2.9 lze vidět blok PMSM z celkového modelu, který představuje matematický model synchronního motoru s permanentními magnety. Tato část byla upravena pro použitý motor, který má velikosti indukčností rotoru v na sebe kolmých směrech v literatuře označované jako L_d a L_q stejné, nebo lišící se zanedbatelnou hodnotou. Vstupními parametry jsou zde moment setrvačnosti (J), odpor vinutí (rs), indukčnost rotoru (L_s), počet pólů (Pp) a magnetický tok od permanentních magnetů ($flux$). Model respektuje rovnice, popsané v části 2.2.3. Kromě řídicích napětí vstupuje do modelu také zátěžný moment (TL).

2.4.1 Ladění parametrů a průběhy veličin

Vstupní parametry mechanické části PMSM modelu byly použity stejné jako v praktické části 3.2, ale PI regulátory bylo potřebné naladit samostatně, nezávisle na fyzickém systému. Pro nezatížený stav byly konstanty regulátorů nastaveny po vyladění následně (pro přehlednost bylo použito stejné značení, jako v praktické části). Integrovaná a proporcionální zesílení proudového regulátoru:

$$Ki = 0.3,$$

$$Kp = 2.$$

Integrovaná a proporcionální zesílení rychlostního regulátoru:

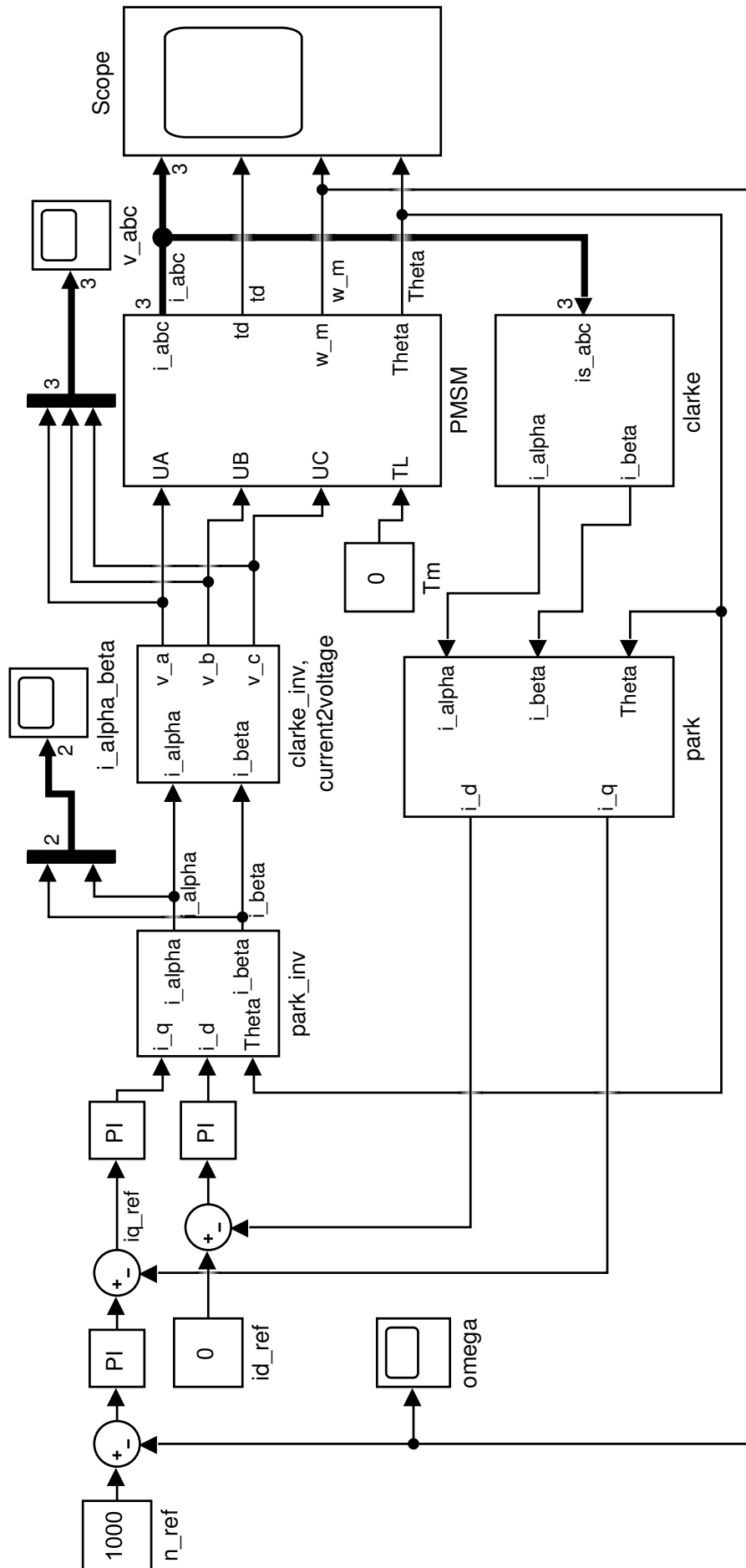
$$Ki_{speed} = 0.4,$$

$$Kp_{speed} = 1.8.$$

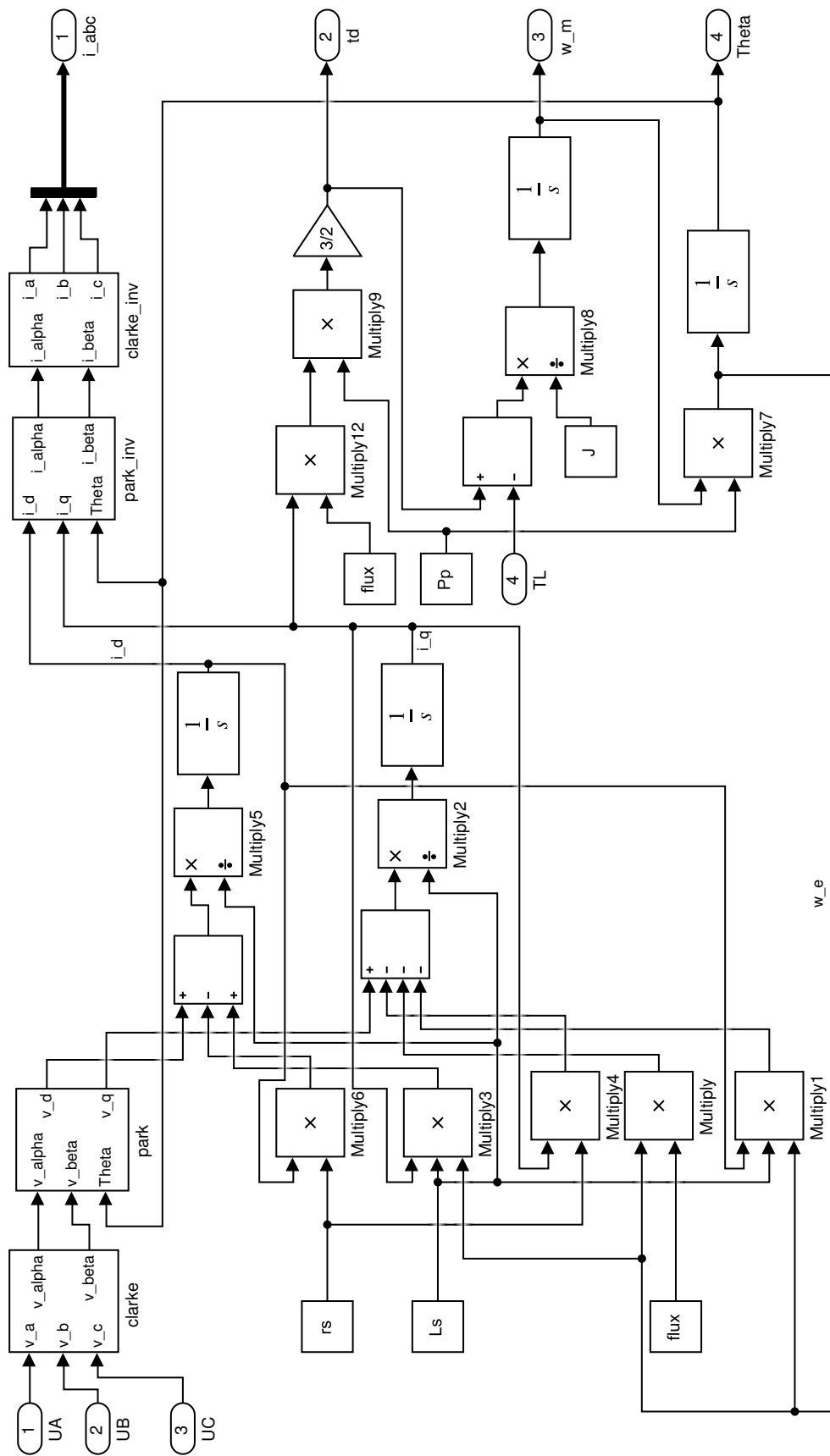
Po spuštění simulace bylo dosaženo dále popsaných průběhů sledovaných veličin. Přechodový děj simuloval rozběh motoru z nuly na 1000 RPM. Průběhy proudů v jednotlivých souřadnicových systémech a napětí na fázích jsou na obr. 2.10, přičemž detail těchto grafů je na obr. 2.11. Průběhy momentu na hřídeli a otáček v korelaci s proudem na fázích lze vidět na obr. 2.12. Po ustálení přechodového děje vypadaly průběhy veličin dle obr. 2.13. Model využívá obvyčejné pulzně šířkové modulace, což je vidět na sinusových průbězích veličin. Složitější modely mohou nabídnout také řízení s využitím SVM¹. Tento typ modulace poskytuje lepší efektivitu, podrobnosti lze nalézt v [22].

Model vektorového řízení motoru spolu s matematickým modelem PMSM motoru, realizovaný v prostředí Simulink poskytl ověření matematických vztahů a také průběhy řídicích a snímaných veličin. Sinusové průběhy napětí a proudů na fázích odpovídají očekávání a při startu motoru mají vysoké hodnoty amplitud. Po dosažení požadované rychlosti jsou již průběhy konstantní a značně menší, slouží už totiž pouze pro překonání momentu vytvořeného pasivními silami v motoru. Analogicky je tomu u momentu, který při rozběhu dosáhne pro dané parametry motoru velmi vysoké hodnoty a poté již klesá do běžných provozních hodnot. V grafech jsou také vidět průběhy proudů v systémech $\alpha\beta$ a dq , které odpovídají průběhům v části o transformacích (2.2.2).

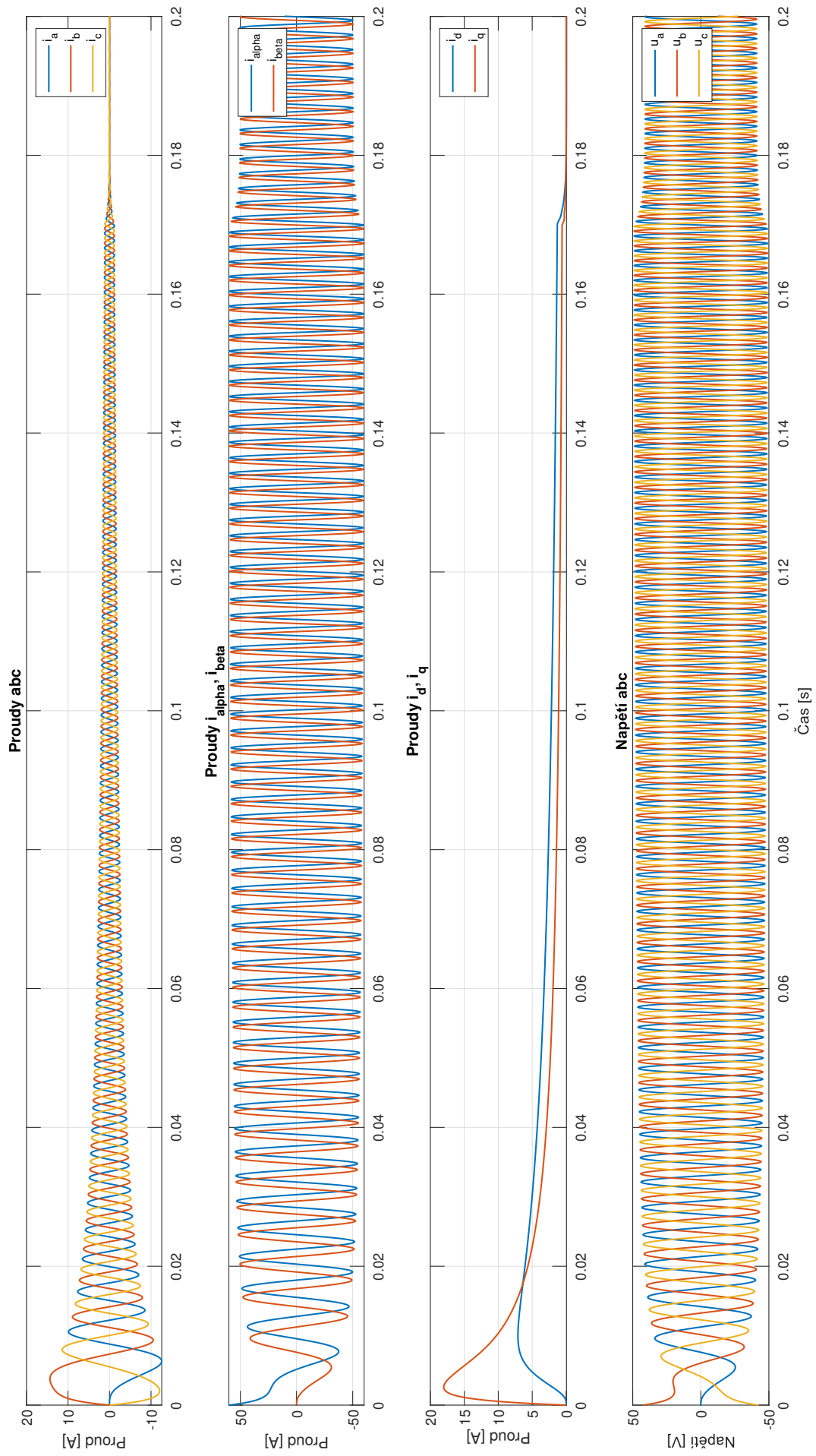
¹Vektorová modulace - Space Vector Modulation



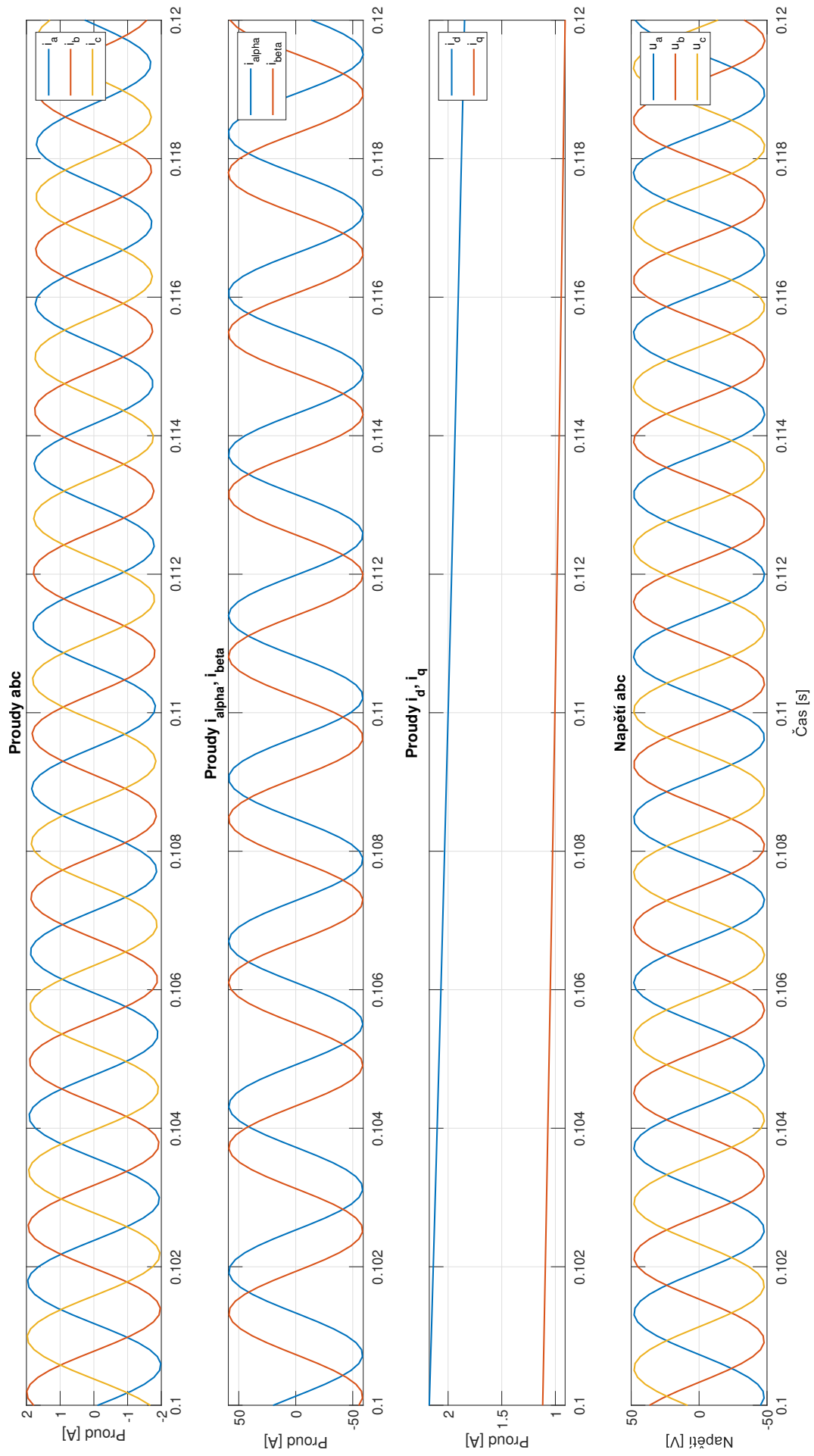
Obrázek 2.8: Model vektorového řízení



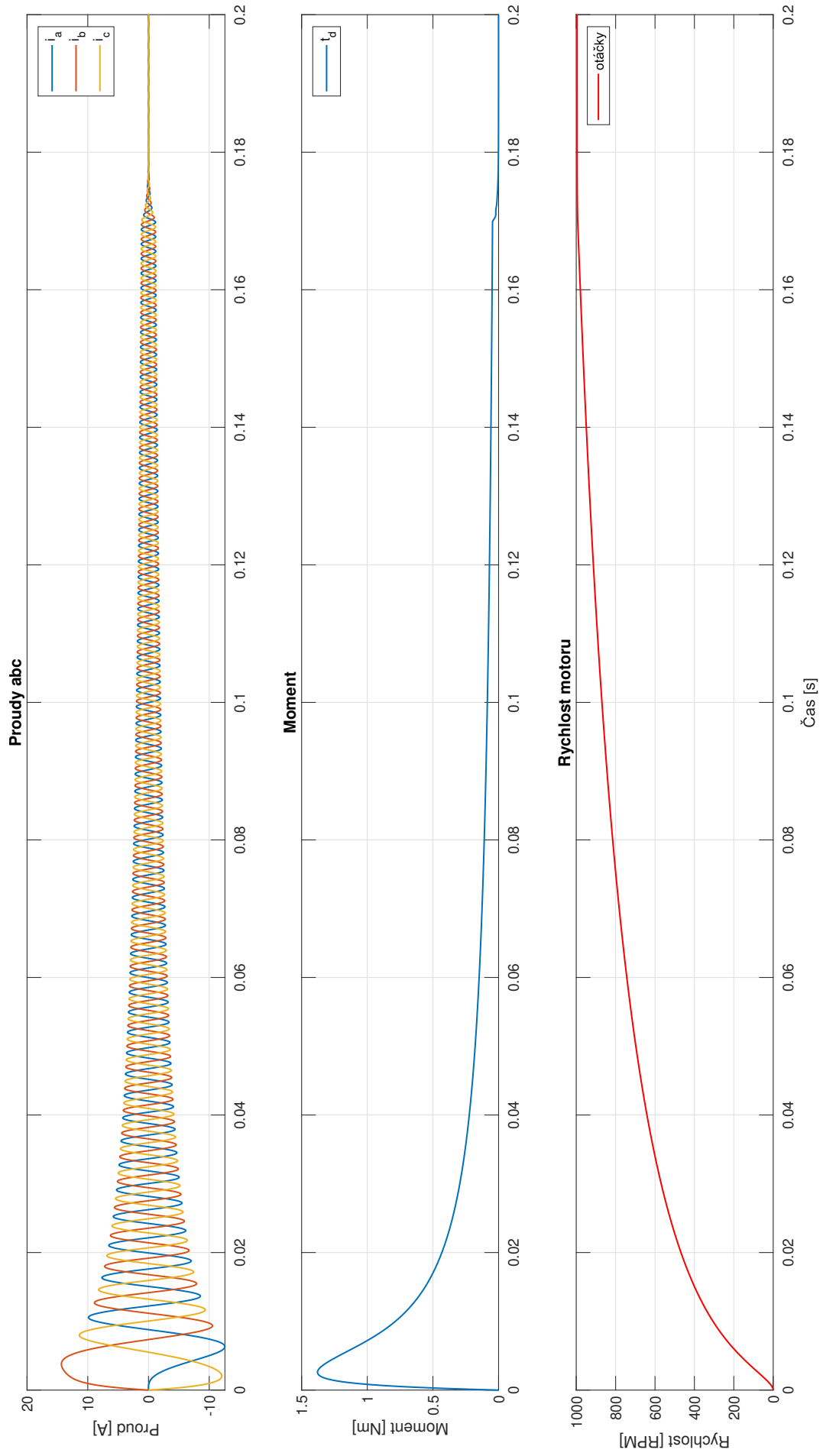
Obrázek 2.9: Model PMSM



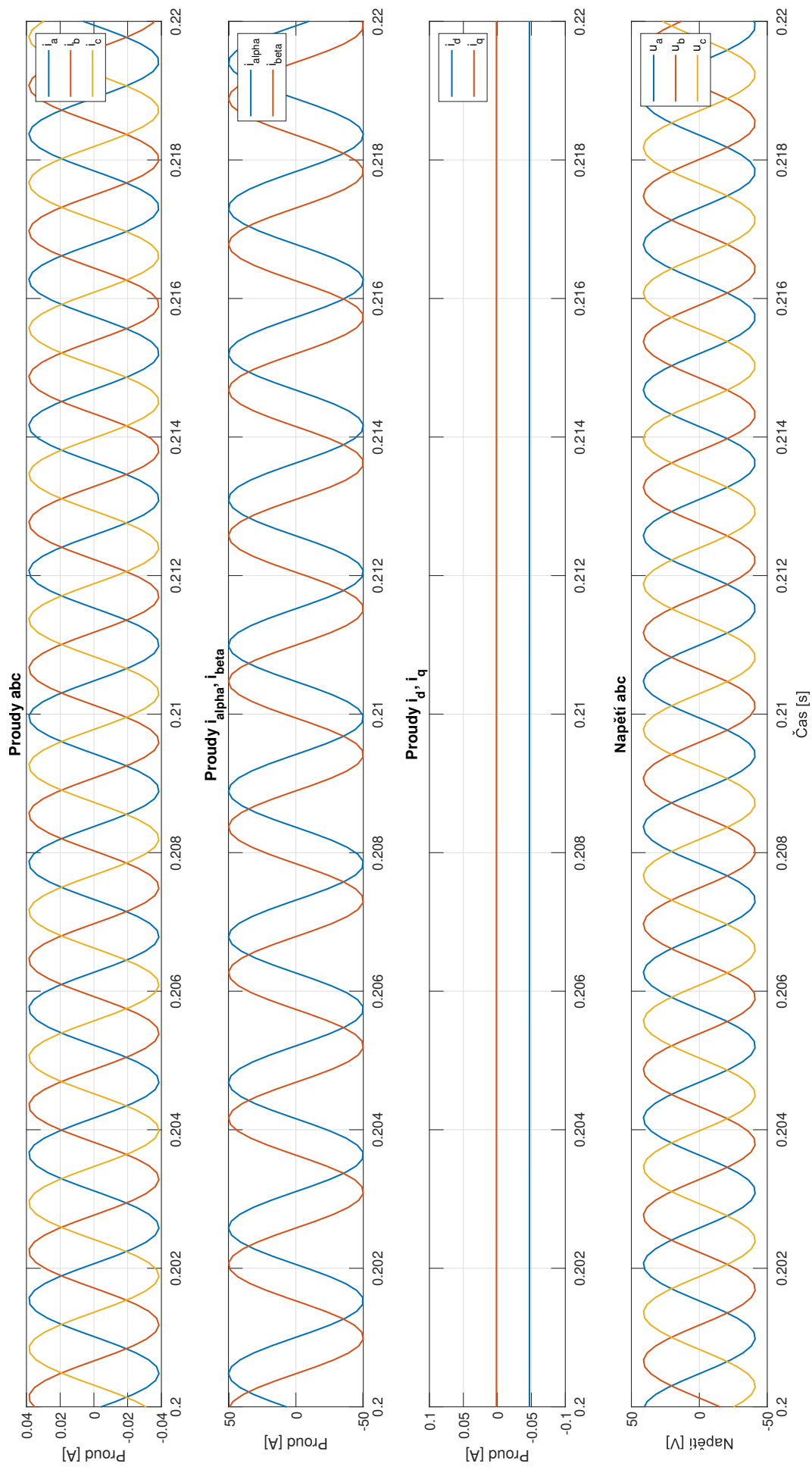
Obrázek 2.10: Průběhy veličin při přechodovém ději



Obrázek 2.11: Detail průběhů veličin při přechodovém ději



Obrázek 2.12: Příběhy veličin při přechodovém ději 2



Obrázek 2.13: Průběhy veličin po ustálení

3. Praktická část

3.1 Použité vybavení

3.1.1 Hardware

Synchronní servomotor

Pro aplikaci vektorového řízení byl zvolen servomotor Yaskawa SGMM-A1S312 (obr. 3.1), který disponuje inkrementálním snímačem polohy se základním rozlišením 2048 pulsů na otáčku, čemuž po kvadratickém vyhodnocení kanálů A a B odpovídá celkem 8192 vzájemně rozlišitelných poloh. Servomotor je konstruován jako motor s permanentními magnety, často také označován v literatuře jako PMSM. Ve špičce dokáže motor udržet maximální moment až tři sekundy a motor také poskytuje vysoký poměr vyvíjeného momentu k setrvačnosti na výstupu hřídele. Motor i enkodér je uzavřený v izolačním pouzdru s vlastním chlazením a stupněm krytí IP55, což znamená částečné krytí proti vniknutí prachových částic a ochranu proti tryskající vodě.

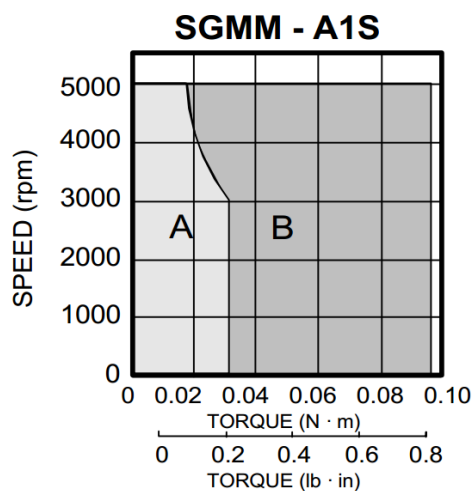
Tato řada SGMM servomotorů od firmy Yaskawa je navržena pro přesné řízení i při malých zátěžích. Motory se dodávají ve dvou velikostech s výkony 10 W a 20 W, přičemž pro realizaci praktické části byla použita verze o výkonu 10 W. Provozní otáčky motoru jsou 3000 RPM¹ a maximální otáčky jsou 5000 RPM. Konstantní moment, jež je motor schopen vyvíjet trvale činí 0,0318 Nm a ve špičce dokáže vyvinout až 0,0955 Nm. Provozní proud při 24 V činí 2,1 A a při špičce až 6 A. Momentová charakteristika je znázorněna na obr. 3.2.

Mezi mechanické vlastnosti motoru patří moment setrvačnosti, který výrobce uvádí jako $0,00354 \text{ kg} \cdot \text{m}^2 \cdot 10^{-4}$ a maximální moment setrvačnosti zatěžujícího zařízení $0,1064 \text{ kg} \cdot \text{m}^2 \cdot 10^{-4}$. Motor má tři elektrické póly.
[8]



Obrázek 3.1: Servomotor SGMM-A1S312 - převzato z <https://s3.amazonaws.com>

¹Z angličtiny "Revolutions Per Minute"- otáčky za minutu.

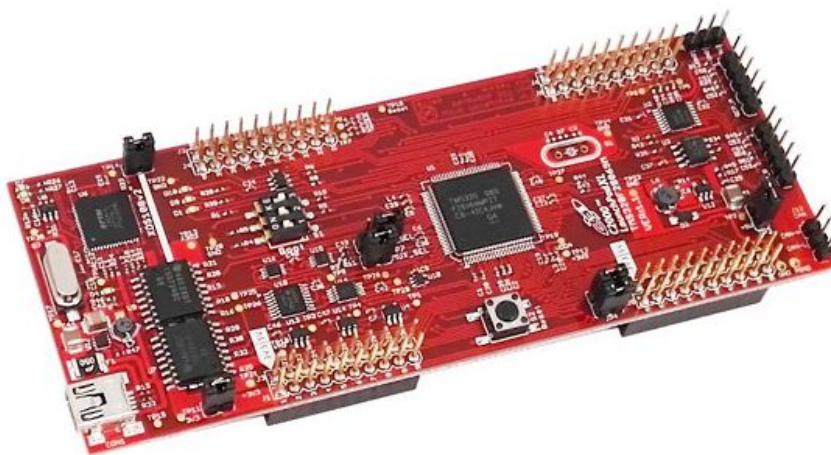


Obrázek 3.2: Závislost momentu a otáček motoru (A - provozní stav, B - stav ve špičce) - převzato z [8]

Řídicí mikroprocesor F28069M

Jako první řídicí platforma byla zvolena vývojová deska LAUNCHXL-F28069M C2000 (obr. 3.3) od firmy Texas Instruments. Tato vývojová deska mimo jiné disponuje technologiemi InstaSPIN-FOC a InstaSPIN-MOTION pro sensed i sensorless řízení motorů. Vývojová deska komunikuje s řídicím počítačem přes USB-JTAG rozhraní.

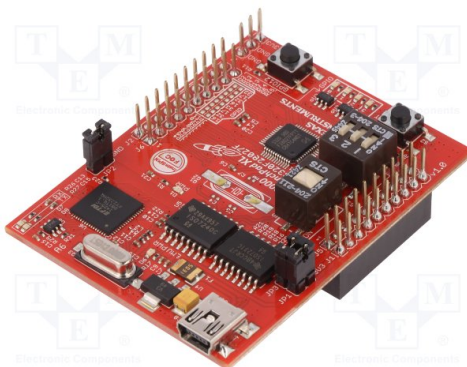
Vývojové zařízení je vybavené 90 MHz mikrokontrolerem TMS320F28069M, tlačítkem pro reset a dvěma sety pinů, na které lze připojit desky pro rozšíření funkčnosti, například výkonový PWM modul pro řízení motoru. Těmto rozšířením se v této oblasti říká tzv. "BoosterPack". K dispozici jsou také rozhraní pro dva enkodéry pro sensed řízení motorů, což bude využito pro připojení enkodéru z testovaného motoru SGMM-A1S312. LAUNCHXL-F28069M C2000 také umožňuje komunikaci přes CAN sběrnici. Napájení je zajišťováno přes USB, nebo lze odstraněním přemostujících hardwarových propojek povolit napájení z BoosterPacku. Napájecí napětí je 5 V. [7]



Obrázek 3.3: Vývojová deska LAUNCHXL-F28069M C2000 - převzato z <http://edgate.in>

Řídící mikroprocesor F28027F

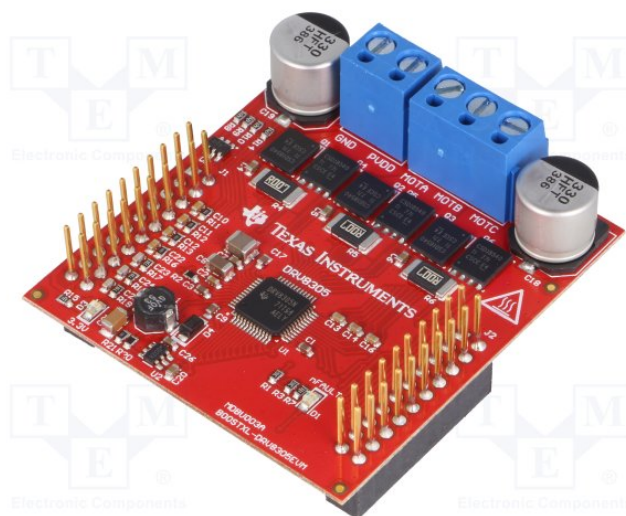
Jako druhá řídicí platforma byla použita deska LAUNCHXL-F28027F C2000 (obr. 3.4), také od firmy Texas Instruments. Tato deska disponuje technologií bezsenzorového řízení InstaSPIN-FOC, je vybavena mikrokontrolerem TMS320F28027F s frekvencí 60 MHz a jedním setem pinů pro připojení rozšiřujícího BoosterPacku. Ostatní vlastnosti jsou podobné jako u desky F28069M. Dva řídicí mikrokontrolery byly použity z potřeby realizovat bezsenzorové řízení motoru v jedné aplikaci (CCS Studio) a zároveň vyčítání dat z enkodéru v aplikaci druhé (Matlab). [18]



Obrázek 3.4: Vývojová deska LAUNCHXL-F28027F C2000 - převzato z <http://www.tme.eu>

BoosterPack

Jako BoosterPack, nebo-li rozšíření (nadvstavba) funkčnosti vývojové desky pro řízení motoru byl zvolen BOOSTXL-DRV8305EVM (obr. 3.5), což je 15 A modul pro napájení třífázových brushless motorů postavený na čipu DRV8305 a obsahuje 6 výkonových MOSFET tranzistorů. Modul dokáže měřit napětí a také proud na jednotlivých fázích motoru a umožňuje tak sensorless řízení motoru. Modul poskytuje řídicí jednotce na kterou se připojuje napájení 3,3 V a řídicí obvod je chráněn proti zkratu, tepelnému přetížení, proražení tranzistorů a příliš nízkému napětí a komunikuje pomocí SPI komunikace. BoosterPack vyžaduje externí napájení od 4,4 V do 45 V. Provozní hodnota výstupního proudu činí 15 A, ve špičce 20 A. [9]



Obrázek 3.5: BoosterPack BOOSTXL-DRV8305EVM - převzato z <http://www.tme.eu>

24 V zdroj

Pro napájení BoosterPacku, skrze něj i řídicí platformy a také samotného motoru byl použit průmyslový zdroj s výkonem 96 W, schopný dodat 5 A při 24 V.



Obrázek 3.6: Spínaný napájecí zdroj - převzato z <https://www.tme.eu>

3.1.2 Software

Motorware

Aplikace Motorware od Texas Instruments obsahuje balíček s dokumentací pro vývoj projektů založených na technologiích InstaSPIN, InstaMOTION a dalších. Součástí Motorware je také zvlášť spustitelná aplikace s přehledným uživatelským rozhraním pro identifikaci parametrů motoru, ladění regulátorů a celkově pro všechny náležitosti spojené s pokročilým řízením třífázových motorů. Univerzální aplikace Motorware pro práci s projekty ve zkompilem stavu, který je vygenerován kompilací projektu v CCS Studiu, se nazývá Motorware GUI. [11]

InstaSPIN technologie se dále dělí na InstaSPIN-FOC a InstaSPIN-MOTION.

- InstaSPIN-FOC umožňuje sensorless vektorové řízení motorů a také identifikaci neznámých parametrů motoru.
- InstaSPIN-MOTION staví na InstaSPIN technologii a rozšiřuje robustnost řízení co se týče řízení zrychlení, pozice a plánování pohybu.

CCS studio

Code Composer Studio je integrované vývojové prostředí od firmy Texas Instruments s podporou pro mikrokontrolery a vývojové desky od stejnojmenné firmy. Code Composer Studio obsahuje sadu nástrojů pro vývoj a ladění aplikací určených pro externí zařízení. Součástí je kompilátor z jazyku C/C++, editor zdrojového kódu, prostředí pro tvorbu a správu projektů, nástroj pro ladění kódu a hledání chyb a další. Vývojové prostředí umožňuje realizaci řízení krok po kroku a také real-time² změnu parametrů. [12]

²Zásah probíhá při spuštěném procesu, změny jsou prováděny okamžitě.

Simulink

Simulink je grafické programovací prostředí vyvíjené společností Mathworks a slouží pro modelování, simulaci a analýzu především dynamických systémů. Je využíván v úzkém propojení s programovacím prostředím Matlab a často využíván v aplikacích automatického řízení, nebo při digitálním zpracovávání signálu.

Simulink obsahuje také mnoho doplňujících knihoven, které umožňují komunikaci s hardwarovými vývojovými deskami a dalšími výrobky třetích stran. Mezi tyto se řadí například řada produktů od Arduina, RaspberryPi či Texas Instruments. Simulink byl v této práci využit pro získání signálu z kvadraturního enkodéru v propojení s vývojovou deskou Texas Instruments a také při simulaci vektorového řízení na virtuálním modelu.

3.2 Realizace řízení motoru

Pro realizaci řízení dříve zmíněného servomotoru bylo využito výše uvedených softwarů, kdy jako hlavní prostředí pro úpravu kódů sloužilo CCS Studio, kde lze aktivovat i testování v reálném čase a tedy měnit některé parametry pro účely ladění a testování chodu motoru. Pro opakované spouštění stejného kódu, přesněji z CCS Studia vygenerovaného zkompilevaného souboru byl využit software Motorware GUI, který disponuje přehlednějším uživatelským prostředím a rychlejší dostupností změn parametrů při testování chodu motoru. Bylo využito postupu a zdrojových kódů popsanych v jednotlivých laboratorních úlohách v [14].

3.2.1 Identifikace parametrů motoru

Než lze postoupit k samotné úloze řízení motoru, je nejprve nutné znát parametry motoru. Zde vystupující parametry, které v této kapitole budeme zjišťovat, jsou odpor jednotlivých vinutí statoru (R_s), indukčnost statoru (L_s), magnetický tok motorem ($Flux$) a napěťové a proudové offsety jednotlivých fází motoru. Parametry budou zjišťovány s pomocí technologie InstaSPIN jako průměr z deseti měření, jelikož identifikace parametrů vyjde pokaždé trochu jinak. Parametry, které budou zjištěny v této úvodní úloze budou poté použity pro veškerý následující postup, je tedy důležité parametry naidentifikovat co nejpřesněji. Hodnoty parametrů se poté vloží do souboru *user.h*, který obsahuje identifikační a později identifikované parametry.

Ještě před samotnou identifikací si připravíme parametry které jsou již nyní známé nebo je možné je určit výpočtem. Vložíme typ konstrukce motoru pod parametr *USER_MOTOR_TYPE* (pro náš případ *Pm* - motor s permanentními magnety), neboť je možné stejným softwarem řídit také například indukční motory, nebo motory s permanentními magnety uvnitř rotoru a další méně časté konstrukce. Počet pólpárů dosadíme do *USER_MOTOR_NUM_POLE_PAIRS* (pro námi použitý motor dosadíme hodnotu 3).

Za *USER_MOTOR_RES_EST_CURRENT*, což představuje konstantní proud, aplikovaný pro zjištění odporu vinutí statoru se dle empirických zkušeností doporučuje dosadit deset až dvacet procent provozní hodnoty proudu, tzn. proudu potřebného pro vyvolání provozního momentu. Pro motor SGMM-A1S312 byla zvolena hodnota 0.3 A. Při procesu identifikace je tato hodnota proudu také použita při roztáčení motoru v otevřené smyčce. Pokud se motor netočí během celého procesu identifikace, je doporučeno zvyšovat hodnotu tohoto parametru přírůstky po deseti procentech dokud není dosaženo plynulé rotace během celého identifikačního procesu. Univerzální pravidlo pro volbu hodnoty proudu pro výpočet indukčnosti statoru (parametr *USER_MOTOR_IND_EST_CURRENT*) je opět deset až dvacet procent provozního proudu, zde ovšem záporná hodnota proudu, tedy -0.3 A. Obecně neplatí, že pokud je potřeba zvýšit pro plynulý průběh rotace při identifikaci proud pro určení odporu statorového vinutí, je totéž potřeba i pro výpočet statorové indukčnosti. Parametr *USER_MOTOR_FLUX_EST_FREQ_Hz* udává frekvenci, s kterou je identifikována statorová indukčnost a tok motorem. Obecně pro synchronní motor s permanentními magnety je frekvence 20 Hz dostatečná pro hodnoty statorové indukčnosti pohybující se v desítkách mH a výše. Pro menší hodnoty je doporučována vyšší frekvence až do hodnoty 100 Hz. Při procesu identifikace bylo navyšování frekvence ukončeno na hodnotě 40 Hz. Ostatní parametry, které budou předmětem identifikace necháme nevyplněné (vložíme hodnotu

NULL).

Servomotor SGMM A1S312 se řadí mezi motory s velmi malou indukčností a proto je potřebné použít odpovídající zdrojový kód pro identifikaci parametrů. Laboratorní úloha má v postupu od Texas Instruments číslo 2c a mírně se liší od úlohy 2b, která slouží pro identifikaci motorů s vyššími indukčnostmi, obvykle také značně větších rozměrů. Kvůli tomuto zařazení bylo také nutné upravit další parametry, které jsou sěžejní pro řízení motoru. Pokud při identifikaci indukčnosti statoru motor ztrácí plynulost, točí se trhaně, či zcela zastaví, je možné řešení zvýšení frekvence vysílaných pulzů šířkové modulace, nebo-li PWM signálu. Frekvenci je také doporučeno zvyšovat tehdy, pokud má motor problém s vyvinutím dostatečného momentu pro roztočení při nízkých otáčkách. Parametr s touto hodnotou nalezneme také v souboru *user.h* pod označením *USER_PWM_FREQ_kHz*. Zatímco pro standardní hodnoty indukčnosti bývají hodnoty frekvencí pulzně šířkové modulace mezi 8 kHz a 30 kHz, pro velmi malé indukčnosti a vysoké otáčky se hodnoty mohou pohybovat mezi 45 kHz a 80 kHz. Postupným zvyšováním s přírůstkem po 5 kHz byla zjištěna hodnota dostačující frekvence pro použitý motor 65 kHz. Další parametr podléhající často nutné korekci je frekvence pro odhad konstanty $\frac{R}{L}$, kde se pohybujeme v rozsahu mezi 100 Hz a 300 Hz (parametr *USER_IQ_FULL_SCALE_FREQ_Hz*). Platí zde nepřímá úměra mezi touto frekvencí a velikostí indukčnosti statoru motoru a pro náš případ byla použita hodnota 300 Hz.

V následujících tabulkách je zaznamenán proces identifikace jednotlivých parametrů motoru z deseti měření a také výsledná průměrná hodnota, která byla poté použita pro další zdokonalování řízení. Tabulka 3.1 ukazuje identifikaci odporu vinutí statoru *Rs*, jeho indukčnosti *Ls* a toku motorem *Flux*. Záměrně je zde ponechané označení proměnných stejně jako v prostředí Code Composer Studio a Motorware. [13]

Měření	Rs [Ohm]	Ls [mH]	Flux [V/Hz]
1	0.9178	0.2338	0.01717
2	0.8972	0.2078	0.01726
3	0.9206	0.2160	0.01702
4	0.9492	0.2489	0.01694
5	0.9501	0.2710	0.01699
6	0.9201	0.2293	0.01707
7	0.9190	0.2253	0.01708
8	0.9114	0.2427	0.01721
9	0.9479	0.2519	0.01701
10	0.9345	0.2154	0.01702
Průměr	0.9267	0.2342	0.01708

Tabulka 3.1: Identifikace parametrů motoru

Tabulka 3.2 ukazuje identifikaci napěťových offsetů pro jednotlivé fáze motoru.

Měření	Napěťový offset fáze 1 [V]	Napěťový offset fáze 2 [V]	Napěťový offset fáze 3 [V]
1	0.49438	0.49331	0.49326
2	0.49436	0.49401	0.49332
3	0.49297	0.49181	0.49186
4	0.49336	0.49299	0.49222
5	0.49330	0.49293	0.49220
6	0.49335	0.49296	0.49226
7	0.49319	0.49282	0.49207
8	0.49332	0.49289	0.49216
9	0.49320	0.49284	0.49201
10	0.49323	0.49286	0.49208
Průměr	0.49347	0.49294	0.49234

Tabulka 3.2: Identifikace napěťových offsetů motoru

Tabulka 3.3 ukazuje identifikaci proudových offsetů pro jednotlivé fáze motoru.

Měření	Proudový offset fáze 1 [A]	Proudový offset fáze 2 [A]	Proudový offset fáze 3 [A]
1	1.00438	1.00391	1.00146
2	1.00466	1.00504	1.00175
3	1.00461	1.00398	1.00187
4	1.00482	1.00500	1.00200
5	1.00471	1.00503	1.00195
6	1.00510	1.00534	1.00201
7	1.00495	1.00527	1.00202
8	1.00472	1.00504	1.00203
9	1.00482	1.00514	1.00203
10	1.00486	1.00521	1.00203
Průměr	1.00476	1.00490	1.00191

Tabulka 3.3: Identifikace proudových offsetů motoru

Po úspěšné identifikaci parametrů motoru zbývá uložit nově získané parametry do souboru s nastavením *user.h*, aby poté již mohla být všechna tato data načtena při startu dalšího programu, kde bude jiný cíl a bylo by zbytečné provádět časově náročnou identifikaci parametrů znovu. Na počátku každého spuštění programu, po aktivaci technologie InstaSPIN, je spouštěna kalibrace napěťových a proudových offsetů. Jelikož během běžného chodu motoru a úloh řízení se tyto offsety znatelně nemění, můžeme i tuto část pro další úlohy přeskočit. Do souboru *user.h* tedy nakopírujeme nově zjištěné offsety a to jak napěťové, tak proudové pro všechny tři fáze motoru.

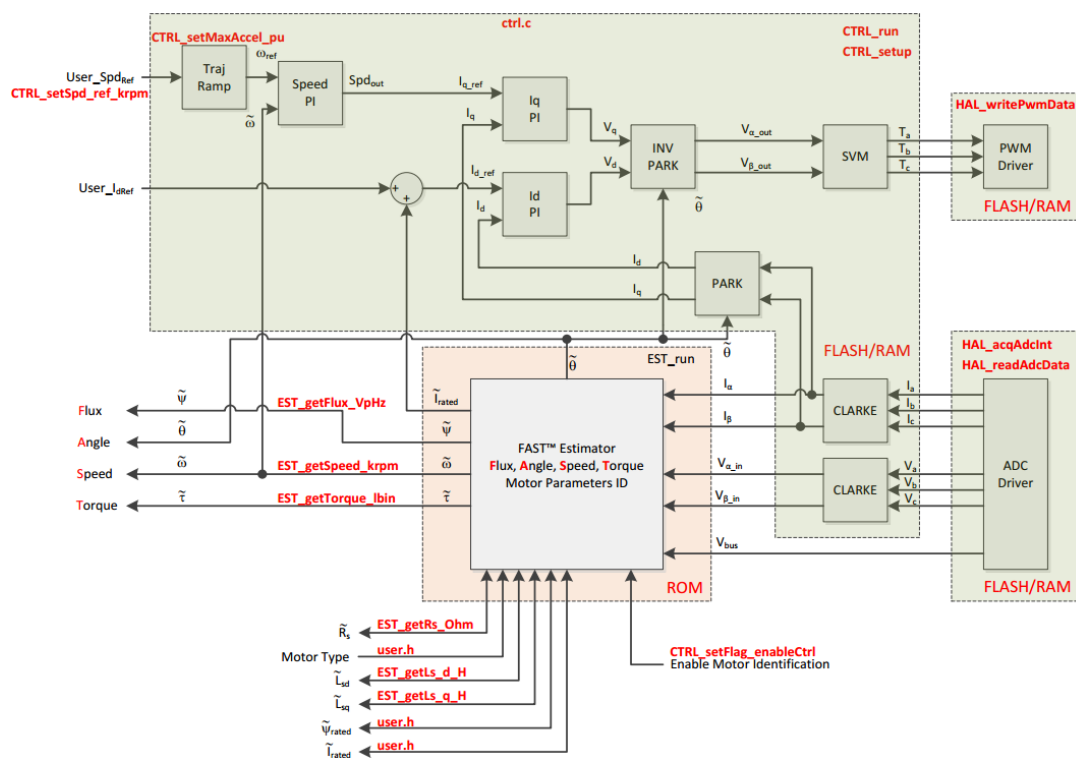
3.2.2 Schéma technologie InstaSPIN a FAST

Aby mohlo být uskutečněno sensorless řízení motoru, musí být aktuální poloha a rychlost rotoru dopočítávána softwarově. K tomu slouží například různé pozorovatele, jak bylo zmíněno v kapitole 2.3. Firma Texas Instruments vyvinula technologii InstaSPIN a s tím související FAST Estimator, jehož písmena značí veličiny, jež je tato technologie schopna dopočítat. Tyto veličiny jsou *Flux*, neboli tok magnetické indukce motorem, *Angle* značí okamžitý úhel, *Speed* aktuální rychlost rotoru a *Torque* je moment vytvářený motorem. Všechny

tyto veličiny jsou přitom získávány pouze z měření proudu a napětí na všech třech fázích motoru, což zajišťuje v našem případě BoosterPack a integrované analogově-digitální převodníky. Těchto šest hodnot v systému abc je poté dvěma transformacemi, Clarkové a Parkovou zredukováno na čtyři hodnoty v systému dq a z těch jsou poté tedy počítány tyto veličiny:

- Tok $\tilde{\Psi}$
- Úhel $\tilde{\theta}$
- Rychlost $\tilde{\omega}$
- Moment $\tilde{\tau}$

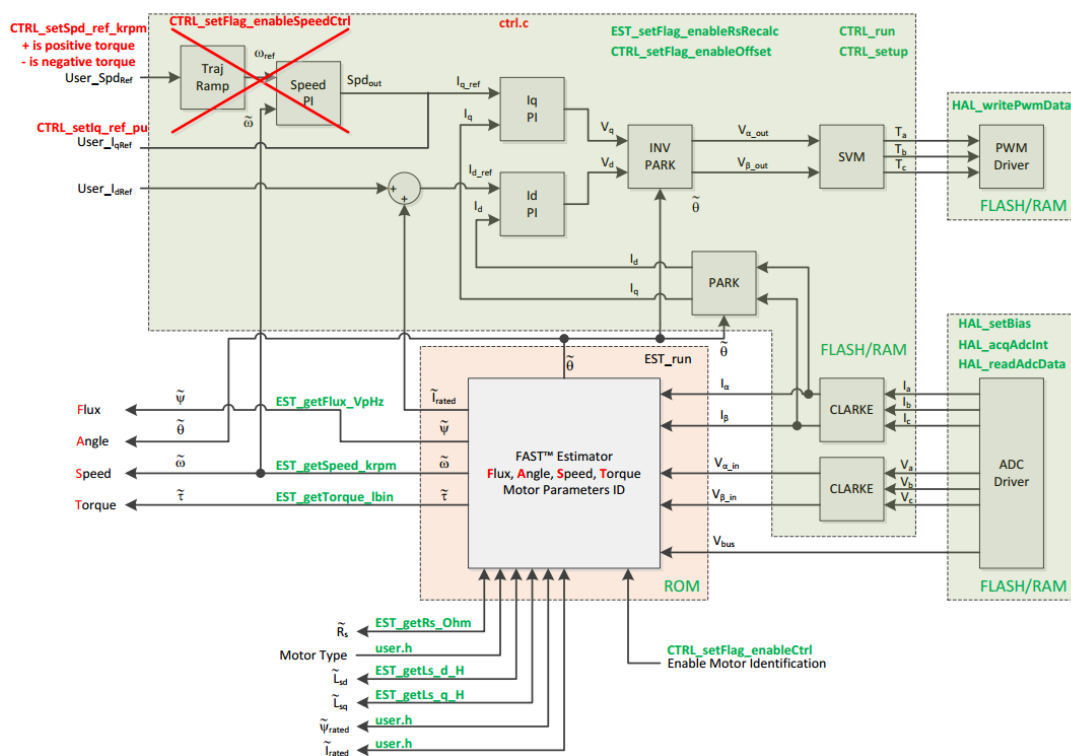
Funkční diagram technologií s jednotlivými bloky je na obrázku 3.7.



Obrázek 3.7: Funkční schéma technologie InstaSPIN a FAST - převzato z [14]

3.2.3 Řízení momentu

V této úloze se budeme zabývat čistě řízením dle požadovaného momentu. Technologie vektorového řízení (FOC - Field Oriented Control) je v jádru řízení momentu. Často však bývá vnořena do smyčky pro řízení rychlosti či polohy. Přidáváním dalších smyček pro řízení dalších veličin zvyšuje možnost vzniku nestabilního řízení. To ovšem neznamená, že při aplikaci pouze řízení momentu je stabilita zaručena - také zde by mělo být na odhad a sledování aktuálního úhlu rotoru technologií FAST nahlíženo kriticky. Pokud budeme mít dobře vyladěnou část řízení momentu, ale řízení rychlosti bude nepřesné či zašuměné, systém se jako celek nebude chovat dle požadavků uživatele. Přidání řízení rychlosti také zakrývá skutečné schopnosti základního systému a platí zde, že celková rychlost odezvy systému po přidání dalších řídicích smyček vždy klesá. Na obrázku 3.8 je vidět změna schématu vypuštěním rychlostní řídicí smyčky a uživatelský referenční vstup je poslán přímo do regulátoru proudu Iq , tedy momentotvorné části proudu. Uživatel v této úloze zadává pouze velikost referenčního



Obrázek 3.8: Funkční schéma - řízení momentu - převzato z [14]

proudu v ampérech do parametru $gMotorVars.IqRef_A$. Hodnota v parametru pro referenční rychlost zde nemá žádný význam, jelikož rychlostní smyčka je vyřazena ze zapojení, ale význam zde má znaménko v tomto parametru, které udává směr, kterým bude motor vyvíjet točivý moment. Pokud má dojít ke změně znaménka a tedy obrácenému chodu motoru, musí být nejprve dosaženo nulové rychlosti motoru.

Při realizaci této úlohy nejprve zvyšujeme proměnný parametr referenčního proudu I_q v přírůstcích po 0.1 A dokud se motor nezačne točit. Pokud nebude dosaženo potřebného proudu pro roztočení motoru, hřídel bude velmi pomalu oscilovat. Tato oscilace je způsobena skutečností, že technologie InstaSPIN obsahuje aktivovatelnou funkci, která umožňuje roztočení motoru s plným zátěžným momentem již od nulové rychlosti. Tato funkce³ může být vypnuta a poté bude dosaženo plynulejších přechodů v oblasti nulové rychlosti. Jelikož je řízení motoru provedeno jako řízení momentu a motor je bez zátěže, bude se hřídel motoru točit maximální možnou (napětím omezenou) rychlostí. [14]

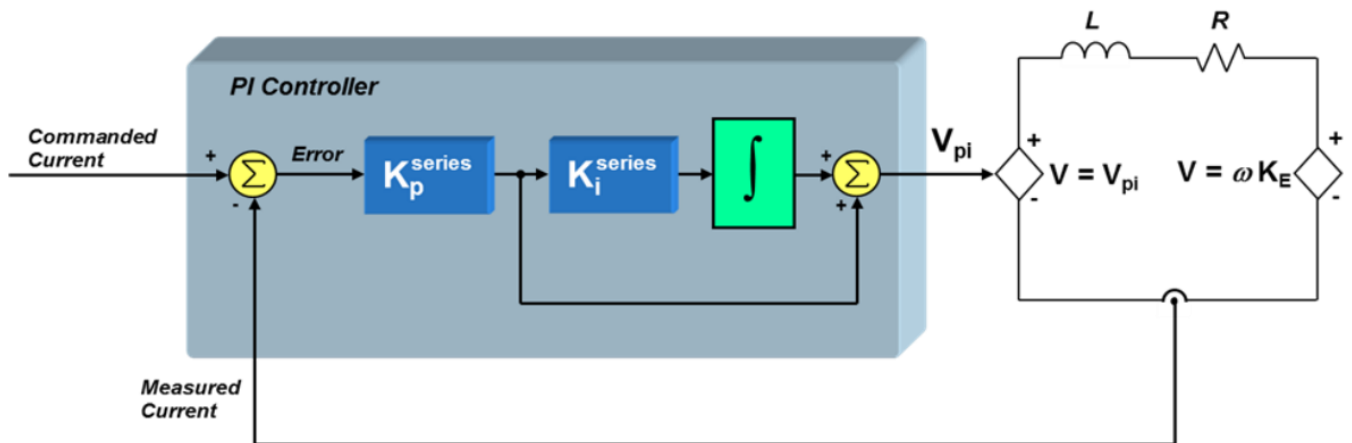
Při točící se hřídeli nyní můžeme vyzkoušet zastavení hřídele. Pokud byla hodnota referenčního proudu I_q nastavena pouze pro překonání počátečního tření a dalších pasivních účinků, bude motor vyvíjet pouze velmi malý moment. Zvyšováním referenční hodnoty můžeme zvyšovat také vyvíjený točivý moment a to až do hodnoty, která byla v souboru s uživatelskými parametry $user.h$ nastavena jako maximální hodnota proudu.

3.2.4 Ladění proudové regulační smyčky

Tato úloha se zabývá nastavováním proporcionálního zesílení K_p a integračního zesílení K_i proudového PI regulátoru. Po výpočtu těchto konstant budou použity v InstaSPIN algoritmu. Na obrázku 3.9 je topologie PI regulátoru proudu a také je zde zahrnut stator motoru. Pro odladění konstant tedy nejprve zkompilujeme projekt lab5a z [14], povolíme ladění v reálném čase a spustíme systém. Konstantu K_i získáme jako poměr R_s ku L_s dle 3.1.

$$K_i = \frac{R_s}{L_s} \quad (3.1)$$

³Funkce je v literatuře od Texas Instruments označována jako Force Angle



Obrázek 3.9: Napojení proudového PI regulátoru na stator - převzato z [14]

Dále si spočítáme periodu proudového regulátoru ze vztahu 3.2.

$$T_i = \frac{1}{PWM_Freq_kHz \cdot 1000} \cdot PWMvsISRtick \cdot ISRvsCTRLtick \cdot CTRLvsCURRENTtick, \quad (3.2)$$

kde:

- T_i je perioda proudového regulátoru,
- PWM_Freq_kHz je frekvence PWM signálu, definovaná v *user.h*,
- $PWMvsISRtick$ je vztah mezi frekvencí PWM signálu a frekvencí interruptů,
- $ISRvsCTRLtick$ je vztah mezi frekvencí interruptů a frekvencí řídicího mikroprocesoru,
- $CTRLvsCURRENTtick$ je vztah mezi frekvencí řídicího mikroprocesoru a frekvencí proudového regulátoru.

Pro přepočet na jednotky PU⁴, se kterými algoritmus pracuje, je nutné původní konstantu K_i přenásobit nově vypočtenou periodou proudového regulátoru dle vztahu 3.3. Konstanta K_i definuje přenosovou nulu PI regulátoru.

$$K_i(PU) = K_i \cdot T_i \quad (3.3)$$

Po aktivaci systému porovnáme námi vypočtenou hodnotu s hodnotou vypočtenou algoritmem, měly by být stejné.

Konstanta K_p definuje šířku pásma řízení. Je doporučeno nejprve zvolit konstantu K_p jako jednu dvacetinu frekvence proudového regulátoru. Základní hodnotu pak spočítáme dle 3.4.

$$K_p = L_s \cdot \frac{1}{20} \cdot \frac{1}{T_i}, \quad (3.4)$$

kde tuto hodnotu ještě přepočítáme na jednotky PU (rovnice 3.5).

$$K_p(PU) = K_p \cdot \frac{I_{fs}}{V_{fs}}, \quad (3.5)$$

kde

⁴Jednotky Per Unit jsou normalizované jednotky se kterými pracují použité regulátory.

- I_{fs} značí celý proudový rozsah (odpovídá parametru $USER_IQ_FULL_SCALE_CURRENT_A$ v $user.h$,
- V_{fs} značí celý napěťový rozsah (odpovídá parametru $USER_IQ_FULL_SCALE_VOLTAGE_V$ v $user.h$.

Takto nově vypočtenou hodnotu vložíme do příslušné proměnné Kp_Idq v algoritmu. Roztočíme motor, začneme s vypočtenou hodnotou a postupně zvyšujeme hodnotu Kp dokud motor nezačne vydávat vysoký pískající zvuk. V tomto okamžiku nastala nestabilita řídicí proudové smyčky a konstantu snížíme, aby byla zachována stabilita. [14]

Zatímco konstanta Ki vytváří nulu, která vyruší pól statoru a lze ji snadno spočítat, proporcionální zesílení Kp upravuje šířku pásma systému proudového regulátoru. Pokud je zavedeno i řízení rychlosti, Kp konstanta proudového regulátoru bude záviset na časové konstantě rychlostního regulátoru a její ladění je tedy vhodné provést až po zavedení rychlostního regulátoru.

Vypočtené hodnoty pro použitý motor a parametry jsou:

$$Ki = 4565.22,$$

$$Ki(PU) = 0.21,$$

$$Kp = Kp(PU) = 1.56 \quad (\text{při } \frac{1}{20} \text{ šířky pásma}),$$

$$Kp = Kp(PU) = 31.31 \quad (\text{při } \frac{20}{20} \text{ šířky pásma}),$$

$$Kp_{krit} = 5.9 \quad (\text{kritická hodnota, kdy nastala nestabilita}).$$

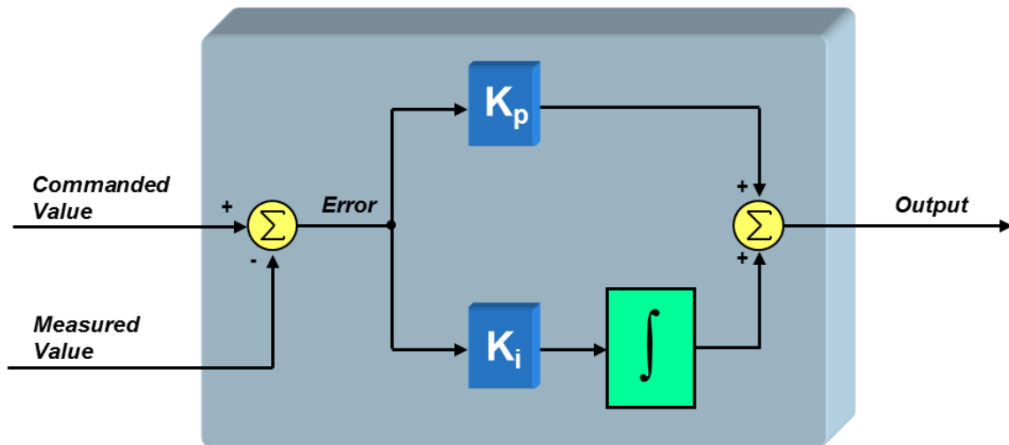
3.2.5 Ladění rychlostní regulační smyčky

Technologie InstaSPIN na vývojové desce od Texas Instruments poskytuje také standardní PI regulátor rychlosti. Proporcionální i integrační zesílení regulátoru jsou pro základní potřeby vypočteny automaticky a mohou sloužit jako startovací hodnoty. Pro lepší dynamické vlastnosti systému je však potřebné tyto hodnoty naladit a tím se budeme zabývat v této části.

Ladění rychlostního regulátoru je mnohem náročnější, než ladění proudového regulátoru. Jádrem rychlostního regulátoru je totiž mechanická část systému a tedy z této skutečnosti vycházejí také typické vlastnosti jako delší časové konstanty a fázová zpoždění, což může mít vliv na stabilitu systému. Nejdůležitější parametr pro úspěšné naladění regulátoru je moment setrvačnosti. Odtud nám vycházejí dva možné přístupy. První přístup volíme, pokud nemáme k dispozici žádné vlastnosti mechanického systému a tato metoda se označuje jako pokus a omyl. Druhý přístup předpokládá znalost mechanických vlastností a zesílení regulátoru jsou pak již dopočítány. [14]

Ladění konstant metodou pokus a omyl

Ladění konstant regulátoru metodou pokus a omyl je často nutností, protože neznáme moment setrvačnosti daného motoru. InstaSPIN technologie zde využívá paralelního uspořádání PI regulátoru pro ovládání rychlosti (obr. 3.10). Obecně platí, že zvyšováním integračního zesílení Ki zvyšujeme tuhost systému, podobně jako bychom zvyšovali tuhost pružiny. Tlumení systému je ovládáno proporcionálním zesílením Kp . Například pokud je konstanta Kp nastavena velmi nízko, konstanta Ki převezme kontrolu a motor se bude chovat jako pružina. Pokud přidáme zátěž, hřídel bude oscilovat a naopak zvětšením konstanty Kp zmenšíme oscilace. Nejprve tedy spustíme systém a možnost ovládání v reálném čase v CCS Studiu. Nejprve chceme, aby motor měl chování jako pružina, nastavíme tedy referenční otáčky na nulu a poté hřídeli rychle pootočíme o 90 stupňů a poté ji necháme volně točit zpět. Toto opakujeme a snižujeme zesílení Kp dokud se nám nepodaří utlumit oscilaci hřídele po jejím volném otočení do původní polohy. Konečná hodnota může být až stokrát menší, než původně navržená startovací hodnota. Čím více snižujeme hodnotu Kp , tím více se motor chová jako pružina. Pokud



Obrázek 3.10: Schéma paralelního PI regulátoru rychlosti - převzato z [14]

je hodnota K_i příliš vysoká, může být náročné otočit hřídel a proto snížíme hodnotu K_i , aby bylo dosaženo slabého odporu při otočení hřídelí. [14]

Pro ztlumení motoru zvyšujeme hodnotu K_p dokud nepomine pružinové chování hřídele. Zvyšování proporcionálního zesílení rychlostního regulátoru také zvyšuje ztlumení oscilací. U některých motorů tedy může být pro požadovaný účinek nastavena poměrně vysoká hodnota.

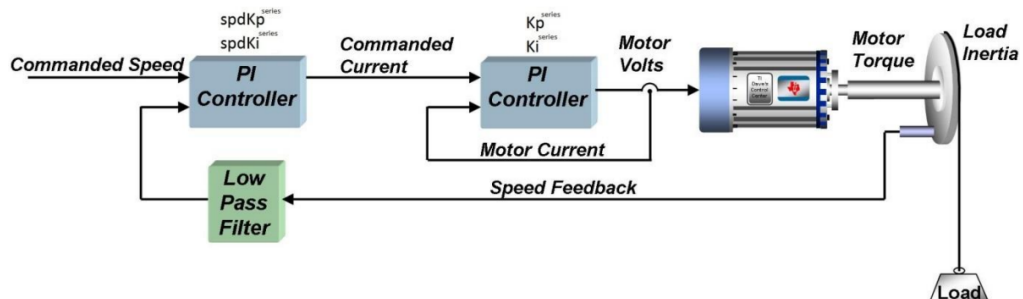
Nyní zbývá vyzkoušet nastavení tuhosti systému a to se provede zvyšováním integračního zesílení. Víme-li, že hodnota K_i zvyšuje konstantu pružnosti systému, v případě nestability tuto hodnotu snížíme, aby byl systém opět stabilizován. Naopak hodnota K_p zvyšuje tlumení systému a pro stabilizaci systému je možné její hodnotu zvýšit. Pomocí těchto znalostí a následném testování při běhu motoru při různých otáčkách a dynamických změnách bylo dosaženo konečných hodnot pro zmiňovaný motor:

$$K_{i_{speed}}(PU) = 0.0032,$$

$$K_{p_{speed}}(PU) = 0.14.$$

Ladění konstant výpočtem

U motoru SGMM-A1S312 však moment setrvačnosti známe a proto také vyzkoušíme metodu výpočtu zesílení regulátoru. Rychlostní signál od motoru musí být často filtrován, než je použitelný řídicím systémem. Zde je použitý jednoduchý filtr s dolní propustí. Na obrázku 3.11 je vidět kaskádové uspořádání regulačních smyček a filtr rychlostního signálu. Tento filtr, zejména tedy jeho rychlostní konstanta τ bude mít význam v dalších výpočtech zesílení. Periodu proudového regulátoru T_i jsme již definovali dříve, nyní musíme analogicky definovat



Obrázek 3.11: Kaskádové uspořádání rychlostní a proudové regulační smyčky a filtru - převzato z [14]

periodu rychlostního regulátoru jako

$$Tv = \frac{1}{PWM_Freq_kHz \cdot 1000} \cdot PWMvsISRtick \cdot ISRvsCTRLtick \cdot CTRLvsSPEEDtick,$$

kde

- Tv je perioda rychlostního regulátoru,
- $CTRLvsSPEEDtick$ je vztah mezi frekvencí řídicího mikroprocesoru a frekvencí rychlostního regulátoru.

Dále definujeme veličinu Vel_{fs} , která představuje plný rozsah frekvence v Hz a odpovídá hodnotě proměnné $USER_IQ_FULL_SCALE_FREQ_Hz$ v uživatelském nastavení v souboru *user.h*. Nyní definujeme tlumící faktor δ . Pro tuto úlohu bylo zvoleno $\delta = 4$ a $\tau = 1/200$ (časová konstanta rychlosti filtru rychlostního signálu). Ostatní použité symboly již byly uvedeny u ladění proudového regulátoru. [14]

Nejprve spočítáme hodnotu integračního zesílení rychlostního regulátoru, pro odlišení od proudových konstant označíme jako $spdKi$.

$$spdKi = \frac{1}{\delta \cdot \tau}. \quad (3.6)$$

Poté vypočteme konstantu K z parametrů motoru jako

$$K = \frac{3 \cdot P \cdot \lambda_r}{4 \cdot J}, \quad (3.7)$$

kde

- P je počet pólů motoru,
- λ_r je hodnota zpětné elektromagnetické síly (Back EMF),
- J je moment setrvačnosti motoru.

Nyní už můžeme vypočítat konstantu $spdKp$ dle 3.8.

$$spdKp = \frac{1}{\delta \cdot K \cdot \tau}. \quad (3.8)$$

Nesmíme však zapomenout, že tyto vzorce jsou odvozeny pro klasické sériové uspořádání složek regulátoru, ale zde bylo použito paralelní uspořádání dle obr. 3.10. Musíme proto hodnoty následně přepočítat dle 3.9 a 3.10.

$$spdKi^{par} = spdKi^{ser} \cdot spdKp^{par}, \quad (3.9)$$

$$spdKp^{par} = spdKp^{ser}. \quad (3.10)$$

Provedené výpočty však ještě nemohou být použity v digitálním PI regulátoru, neboť je ještě musíme převést na jednotky PU s využitím vztahů 3.11 a 3.12.

$$spdKi(PU) = spdKi^{par} \cdot Tv \cdot \frac{4\pi \cdot Vel_{fs}}{I_{fs} \cdot P}, \quad (3.11)$$

$$spdKp(PU) = spdKp^{par} \cdot \frac{4\pi \cdot Vel_{fs}}{I_{fs} \cdot P}. \quad (3.12)$$

Základní hodnoty regulačních zesílení, které byly nastavené automaticky byly:

$$spdKi(PU) = 0.00242,$$

$$spdKp(PU) = 0.525.$$

Hodnoty, které vyšly pomocí výše uvedených výpočtů:

$$spdKi(PU) = 8.22 \cdot 10^{-5},$$

$$spdKp(PU) = 0.019.$$

Z důvodu velmi malého momentu setrvačnosti motoru ($0.00354 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$), však výpočty nebylo dosaženo uspokojivých a v řídicím algoritmu funkčních hodnot a byly použity parametry naladěné první metodou:

$$\text{spdKi}(PU) = 0.0032,$$

$$\text{spdKp}(PU) = 0.14.$$

Nastavení zesílení regulátoru rychlosti tak, aby bylo dosaženo plynulého chodu motoru až do jeho maximálních otáček 5000 RPM bylo posledním krokem před samotným porovnáním rychlostních signálů z enkodéru a ze softwarového observeru. Vyladěním proudové i rychlostní regulační smyčky byly připraveny optimální podmínky pro další experiment.

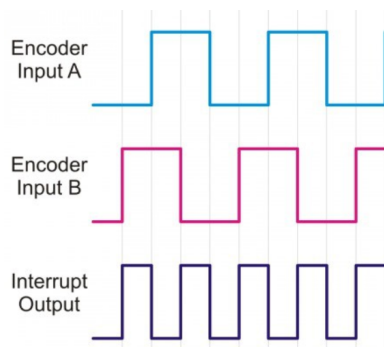
3.3 Porovnání rychlostních signálů

Jedním z cílů této práce je porovnání kvality rychlostního signálu z kvadraturního enkodéru a softwarového pozorovatele, který je součástí technologie pro vektorové řízení od Texas Instruments, již dříve zmiňované technologie InstaSPIN (resp. pozorovatel FAST).

Na kvalitě polohového, případně rychlostního signálu závisí z velké části chod motoru, neboť je zdrojem zpětné vazby pro výpočet odchylky, která slouží jako řídicí signál pro regulátory a také slouží pro výpočet transformací mezi souřadnicovými systémy. V aplikacích, kde je kladen velký důraz na přesnost dosahovaných otáček je tedy kladen důraz na kvalitní enkodér a zpracování signálu z něj, či propracovaný softwarový pozorovatel, který pracuje spolehlivě v pracovním prostoru dané aplikace.

3.3.1 Kvadraturní enkodér

Servomotor SGMM-A1S312 je vybaven inkrementálním kvadraturním enkodérem s 2048 pulzy na jednu otáčku⁵. Tento enkodér obsahuje tři kanály A, B a C, přičemž ke zjištění rychlosti a také směru otáčení stačí pouze dva kanály. Kvadraturní enkodér, také známý jako rotační enkodér, využívá posunu mezi signály, které přicházejí z kanálu A a B. Podle pořadí aktivace signálů přepnutím na logickou jedničku lze určit směr otáčení. Každý kanál vyše za jednu otáčku rotoru 2048 pulzů. Logika, která zpracovává a vyhodnocuje signály bývá často nastavena tak, že celkovou frekvenci pulzů udává jak náběžná, tak sestupná hrana obou signálů. Výsledný signál při měření rychlosti otáčení (celková frekvence pulzů) je znázorněn na obrázku 3.12 a pro dva kanály s 2048 pulzy na otáčku to znamená výslednou hodnotu 8192 pulzů na otáčku. [19]

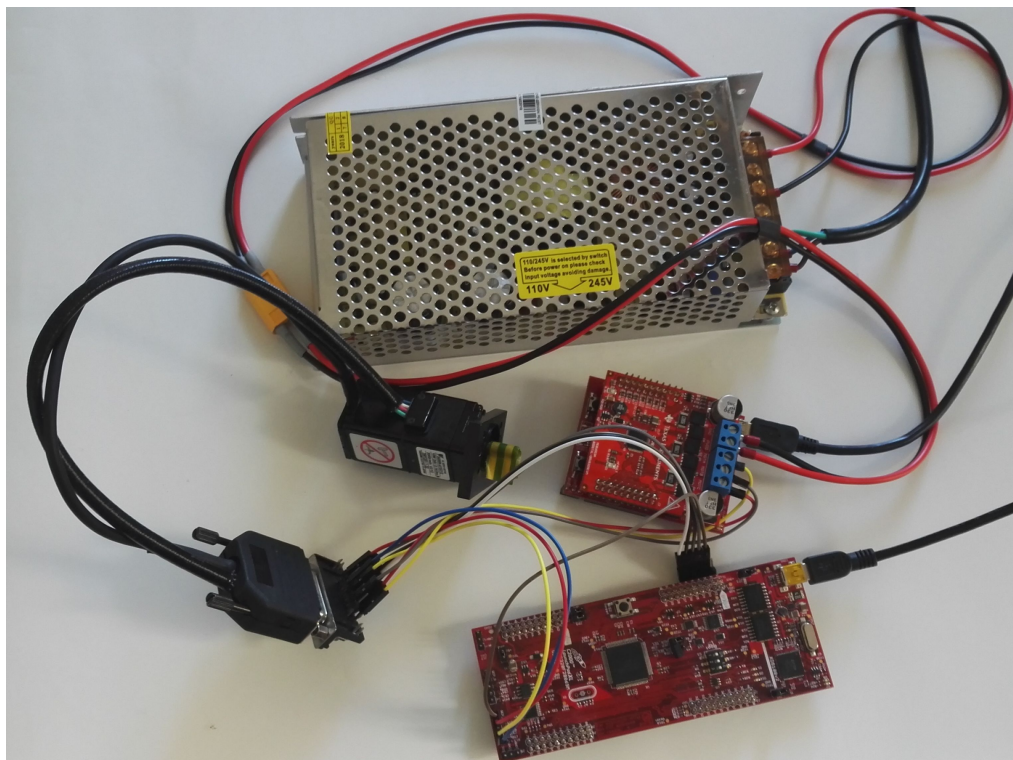


Obrázek 3.12: Signály z jednotlivých kanálů enkodéru, výsledný signál - převzato z [19]

⁵PPR - Pulse Per Revolution (pulzy na otáčku)

3.3.2 Testovací sestava

Na obrázku 3.13 je zobrazena sestava, která byla pro účely experimentu použita. Všechna technika je popsána v části 3.1 a bráno odshora je zde 24 V zdroj, servomotor SGMM-A1S312 s kvadraturním enkodérem, vývojová deska F28027F s Boosterpackem BOOSTXL-DRV8305EVM sloužící pro řízení motoru a vývojová deska F28069M sloužící pro příjem dat z kvadraturního enkodéru.



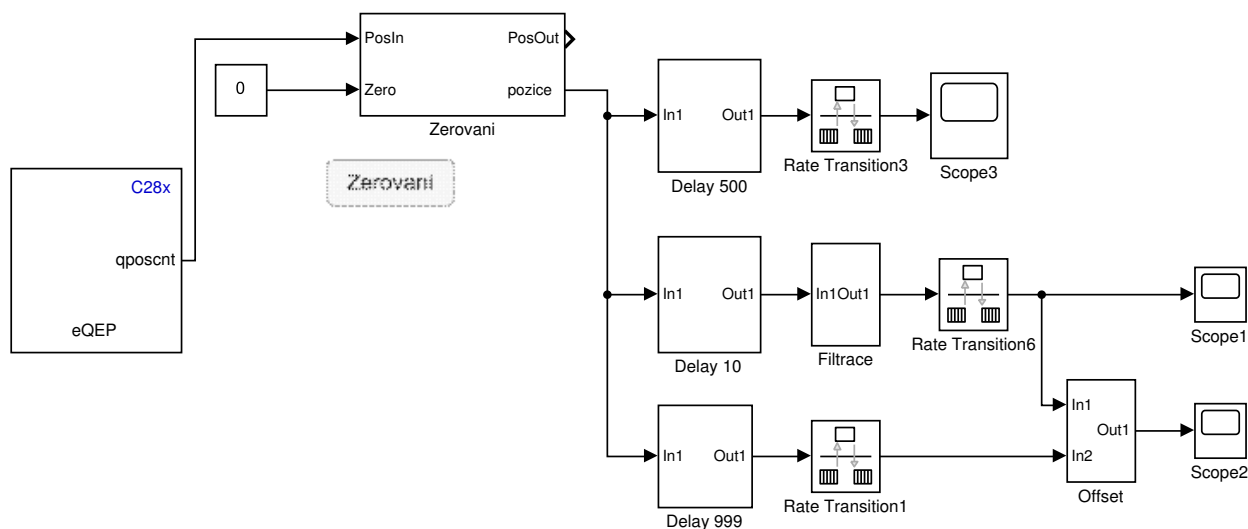
Obrázek 3.13: Testovací sestava

3.3.3 Zpracování signálu z kvadraturního enkodéru

Pro komunikaci s enkodérem byla využita vývojová deska F28069M, která disponuje potřebným rozhraním s možností připojení dvou kanálů z enkodéru, indexovacího signálu a napájení enkodéru. Data byla poté přivezena sériovou komunikací přes USB rozhraní do počítače a tam zpracována v prostředí Simulink (Matlab). Zde bylo vyzkoušeno několik přístupů zpracování signálu, který byl načítán pomocí modulu eQEP z knihovny pro podporu procesorů C2000.

Signál z bloku eQEP z této knihovny vysílá pulzy pro každou náběžnou i sestupnou hranu kanálu A i B, jak je popsáno v části 3.3.1, tedy 8192 pulzů na jednu otáčku a od toho se také odvíjí způsoby, jak tento signál polohy zpracovat na rychlostní signál. Model pro snímání pulzů pracoval na frekvenci 5000 Hz a způsobů pro výpočet rychlosti z polohy bylo využito několik, přičemž po srovnání způsobů byl poté vybrán ten nejvhodnější.

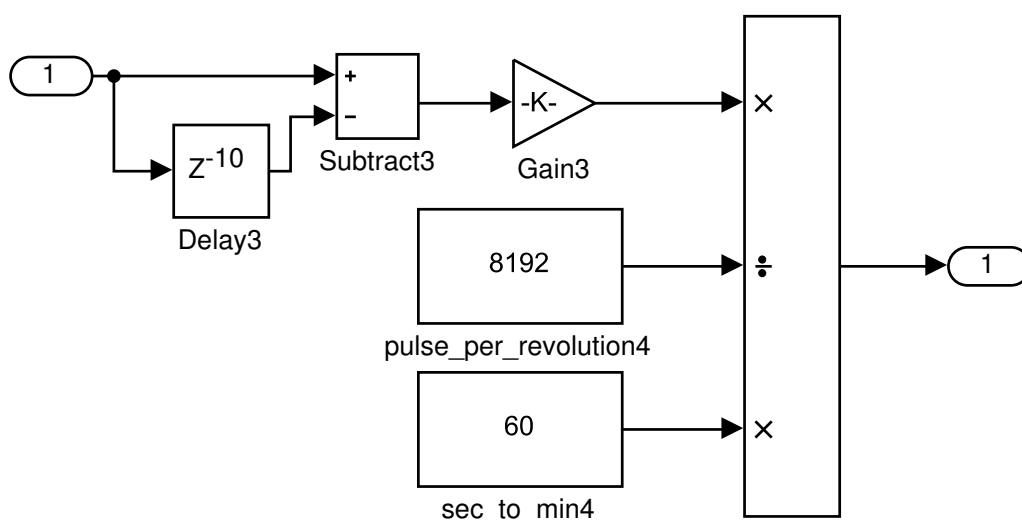
Při velmi náročných aplikacích je každé fázové zpoždění odhadu rychlosti vůči skutečnosti nežádoucí a usiluje se o jeho minimalizaci. Později použité metody filtrování signálu způsobují fázové zpoždění, ale vzhledem k tomu, že účel tohoto srovnání je pouze porovnat kvalitu, nebude zde na toto zpoždění kriticky nahlíženo. Na obrázku 3.14 je model v Simulinku s použitými přístupy. Bloky Rate Transition slouží pro změnu frekvence výpisu výsledků na dané cestě (pro vykreslování dat byla využita nižší frekvence, než pro vyčítání pulzů z enkodéru).



Obrázek 3.14: Model v Simulinku

Metoda difference

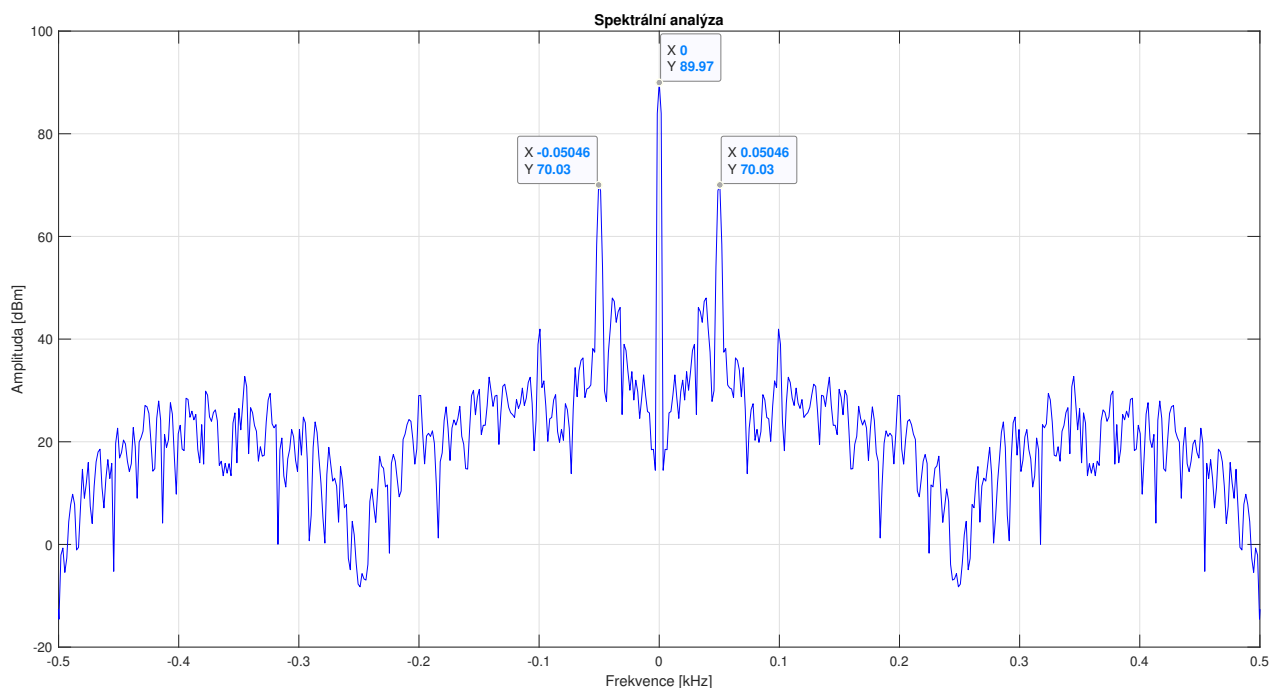
Jako základní metoda pro získání rychlostního signálu byla využita diference pulzů za určitý časový okamžik. Tento základní přístup poskytoval různě uspokojivé výsledky dle toho, jak velká diference byla použita. Spolu s větší použitou diferencí a zlepšující se kvalitou však klesá rychlost odezvy rychlostního signálu, což je vidět z jeho vyhlazení. Byla využita diference pulzů 10, 500 a 999. Poslední zmiňovaná však nebyla použita přímo, ale sloužila pro zpřesnění filtrovaných hodnot, což bude popsáno v další části. Podsystem "Delay 10" je zobrazený na obr. 3.15. Od aktuální hodnoty pulzů je odečtena hodnota pulzů před deseti časovými vzorky a tento výsledek je poté dělený právě deseti časovými vzorky. Výslednou hodnotu je ještě nutné vydělit celkovým počtem pulzů na otáčku, tedy hodnotou 8192. Dostáváme hodnotu otáčky za sekundu, přičemž pro žádané jednotky RPM, tedy otáčky za minutu musíme ještě přenásobit hodnotou 60. Pro diferenci 500 a 999 časových vzorků jsou použity analogicky subsystémy "Delay 500" a "Delay 999".



Obrázek 3.15: Subsystém Delay 10

Metoda filtrace

Druhá metoda se zakládala na použití filtru. Filtrován byl signál generovaný nejjednodušším způsobem pro získání rychlosti, diferencí deseti časových vzorků popsány v předchozí části. Pro implementaci filtru signálu byl využit blok "Filter Designer", který umožňuje nastavení velkého počtu různých typů filtrů. Pro správné nastavení filtru byla provedena spektrální analýza signálu, který byl získáván. Pro různé rychlosti se však frekvence signálu měnila a frekvence se pohybovaly od 0 Hz do 150 Hz při otáčkách 0 RPM až po provozní otáčky 3000 RPM. Pro hodnotu 1000 RPM lze na obrázku 3.16 vidět, že nosná frekvence signálu se pohybuje okolo 50 Hz. Z důvodu



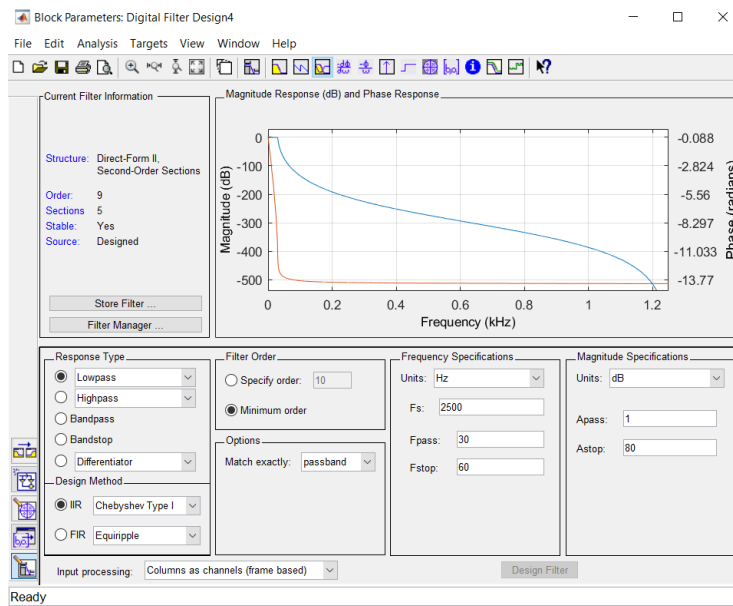
Obrázek 3.16: Frekvenční spektrum rychlostního signálu při 1000 RPM

odlišných frekvencí signálu při různých otáčkách byla použita funkce, která dle aktuální rychlosti zapojovala filtry, které byly navrženy pro funkci v dané pracovní oblasti. Rozsah otáček od nuly až po pracovní otáčky byl rozdělen na 5 oblastí a pro tyto oblasti byly navrženy filtry. Bylo vyzkoušeno několik typů filtrů a nejlépe signál filtroval IIR⁶ filtr typu "Chebyshev Type 1". Využitím informací o frekvenčních vlastnostech signálu byly tyto filtry nastaveny. Nastavení filtru pro pracovní oblast okolo 1000 RPM včetně amplitudové a fázové charakteristiky filtru lze vidět na obrázku 3.17. Subsystem pro přepínání filtrů je zobrazen na obrázku 3.18, kde vstup 1 je rychlostní signál z diference o hodnotě deseti časových oken.

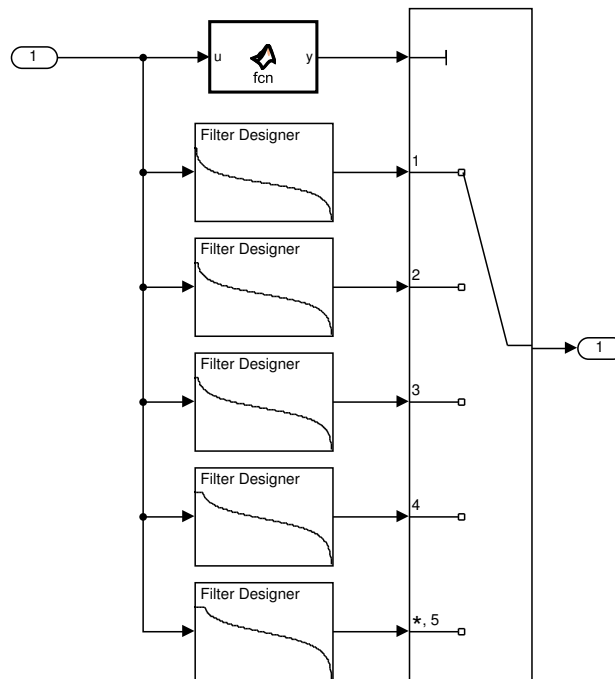
Metoda kombinace filtrace s diferencí

Třetí metoda se také zakládala na filtrování signálu. Bylo však využito také vysoké diference v hodnotě 999 časových okamžiků. Od teoreticky přesnější hodnoty z vysoké časové diference byla odečtena hodnota filtrovaného signálu a tato odchylka byla poté přičtena k původnímu filtrovanému signálu. Cílem bylo odstranit trvalou odchylku, nebo-li offset, který vznikl použitím filtrů. Subsystem lze vidět na obrázku 3.19, kde vstup 1 je filtrovaný signál a vstup 2 je signál ze subsystému "Delay 999".

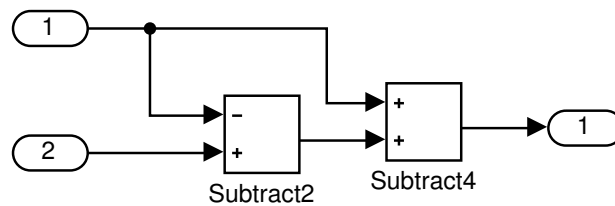
⁶Infinite Impulse Response - diskretní lineární filtr s nekonečnou impulzní odezvou



Obrázek 3.17: Nastavení filtru pro pracovní oblast okolo 1000 RPM



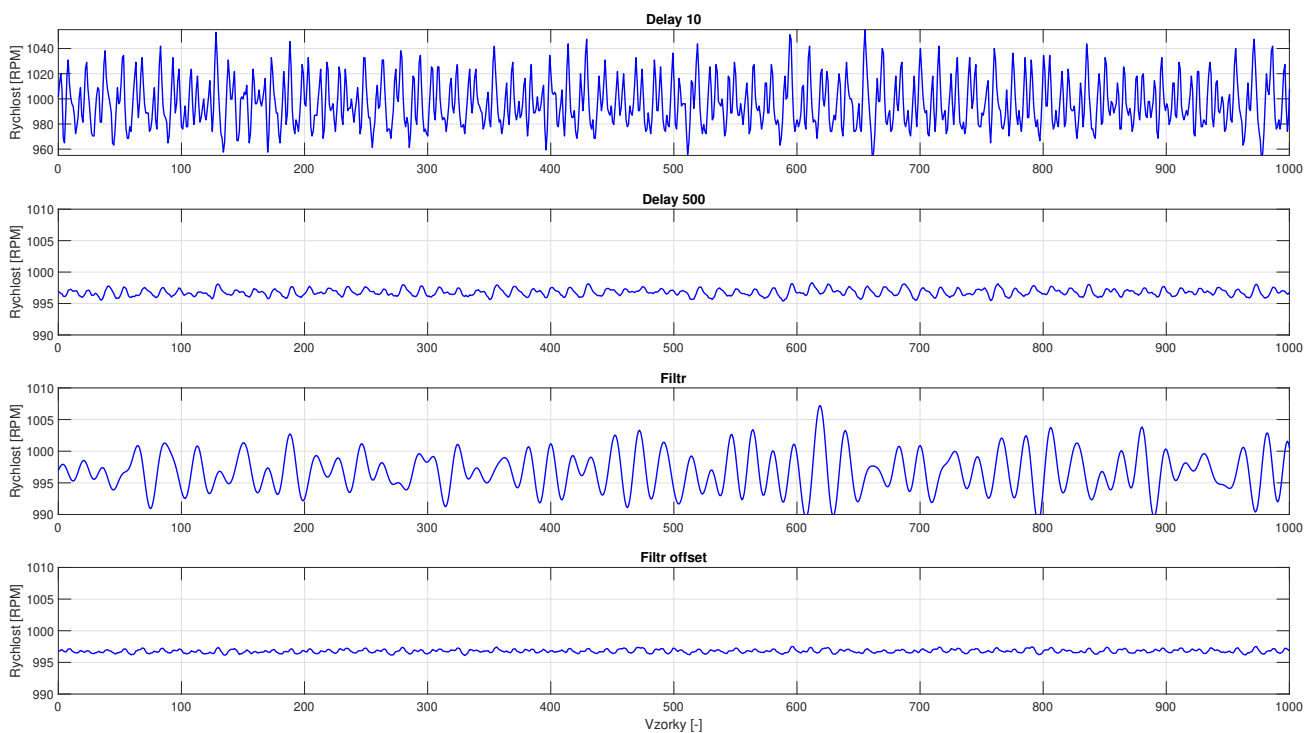
Obrázek 3.18: Subsystem Filtrace



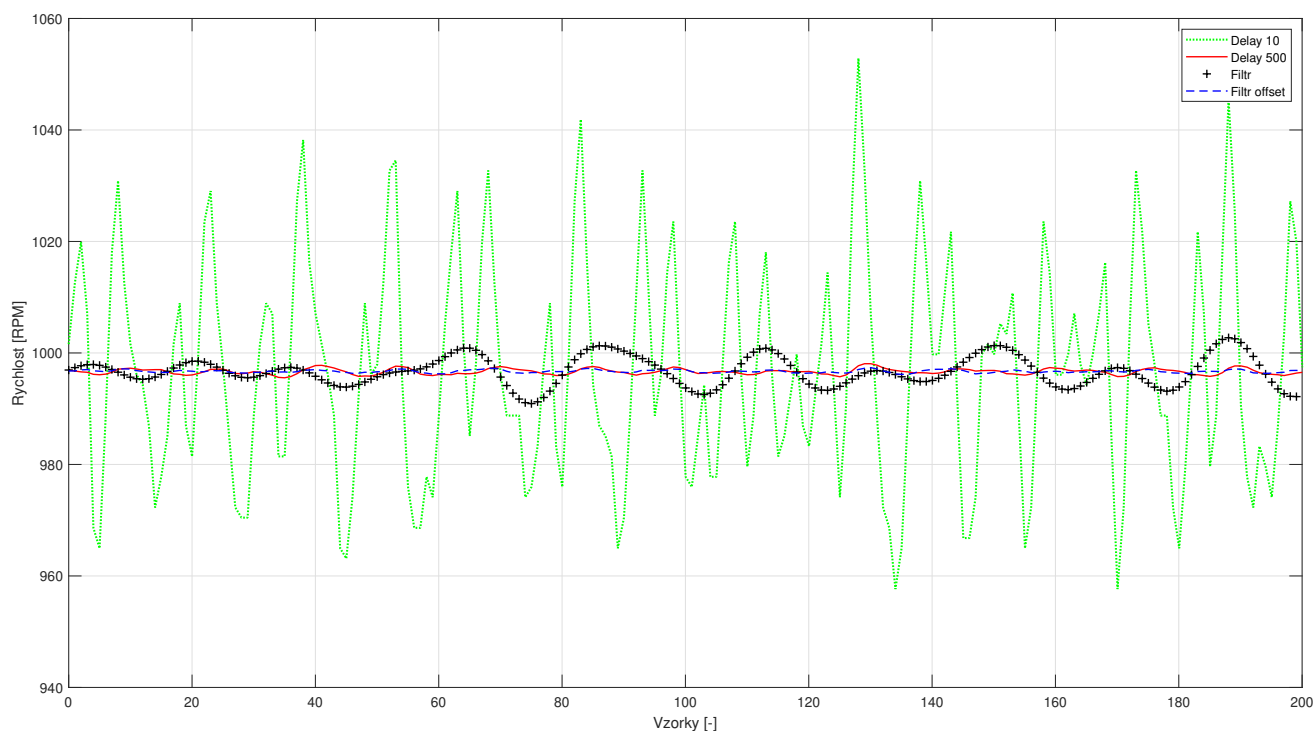
Obrázek 3.19: Subsystem Offset

Výběr metody

V této části zhodnotíme a porovnáme výše navržené metody zpracování signálu z kvadratického enkodéru. Na obrázku 3.20 je srovnání čtyř metod, difference s 10 a 500 časovými okny, filtrace a kombinace filtrace s diferencí za účelem odstranění offsetu. Srovnání přístupů je prováděno při otáčkách 1000 RPM. Pro zobrazení jednotlivých výkmitů není měřítko na ose y sjednocené pro všechny přístupy. Přímé srovnání je dostupné na obrázku 3.21, kde jsou signály zobrazeny současně.



Obrázek 3.20: Srovnání přístupů 1



Obrázek 3.21: Srovnání přístupů 2

Z výsledků srovnání je patrné, že nejmenší oscilace signálu bylo dosaženo pomocí metody difference s 500 časovými okny a také pomocí kombinace filtrace s diferencí. Samotná filtrace dosahuje také uspokojivých hodnot. Nejhlůře dle očekávání dopadla difference s 10 okny, kde úroveň filtrace, kterou tato metoda poskytuje, není dostatečná. Z obrázků je také patrné, že všechny signály jsou zatížené velmi malou trvalou odchylkou, která je však vzhledem ke své velikosti (méně jak 0.5% z celkové rychlosti) zanedbatelná.

Vzhledem k podobným výsledkům mezi dvěma nejlepšími signály byl zvolen přístup získání rychlostního signálu diferencí polohového signálu s 500 časovými okny a to kvůli nutnosti menšího výpočetního výkonu a zpracovávání signálu.

3.3.4 Průběh srovnání

Nyní již je připravena metoda pro zpracovávání informací z kvadratického enkodéru a je tedy možné přejít k samotnému porovnávání signálů z enkodéru se signály z InstaSPIN pozorovatele FAST. Hodnoty z pozorovatele lze vykreslovat do grafu přímo v reálném čase v CCS Studiu a hodnoty poté exportovat do souboru, kde s nimi lze dále pracovat.

Aby bylo porovnání signálů vypovídající, bude uskutečněno několik experimentů, kdy bude předepsána určitá hodnota rychlosti, náběžná rampa, či sekvence rychlostí. Také bude využito omezení akcelerace a provedení pokusů při různých hodnotách maximální akcelerace. Signály budou zobrazovány pod sebou, nejprve referenční signál a poté měřené signály společně v jednom grafu. Maximální frekvence zaznamenávaných dat pro real-time provoz projektu na mikroprocesorové platformě přes CCS Studio je jeden datový vzorek za 0.1 s, tedy frekvence dat je 10 Hz. Na tuto frekvenci bylo sjednoceno také vykreslování dat v Simulinku. Osy x u všech následujících grafů tedy zobrazují časové vzorky a pro převod do reálného času zde platí vztah 10 vzorků je jedna sekunda. U konstantních hodnot rychlosti bude také uvedena střední hodnota a směrodatná odchylka signálu. Signál ze softwarového pozorovatele bude označován jako "FAST" a signál z enkodéru jako "Enkoder".

Konstantní rychlost 200 RPM

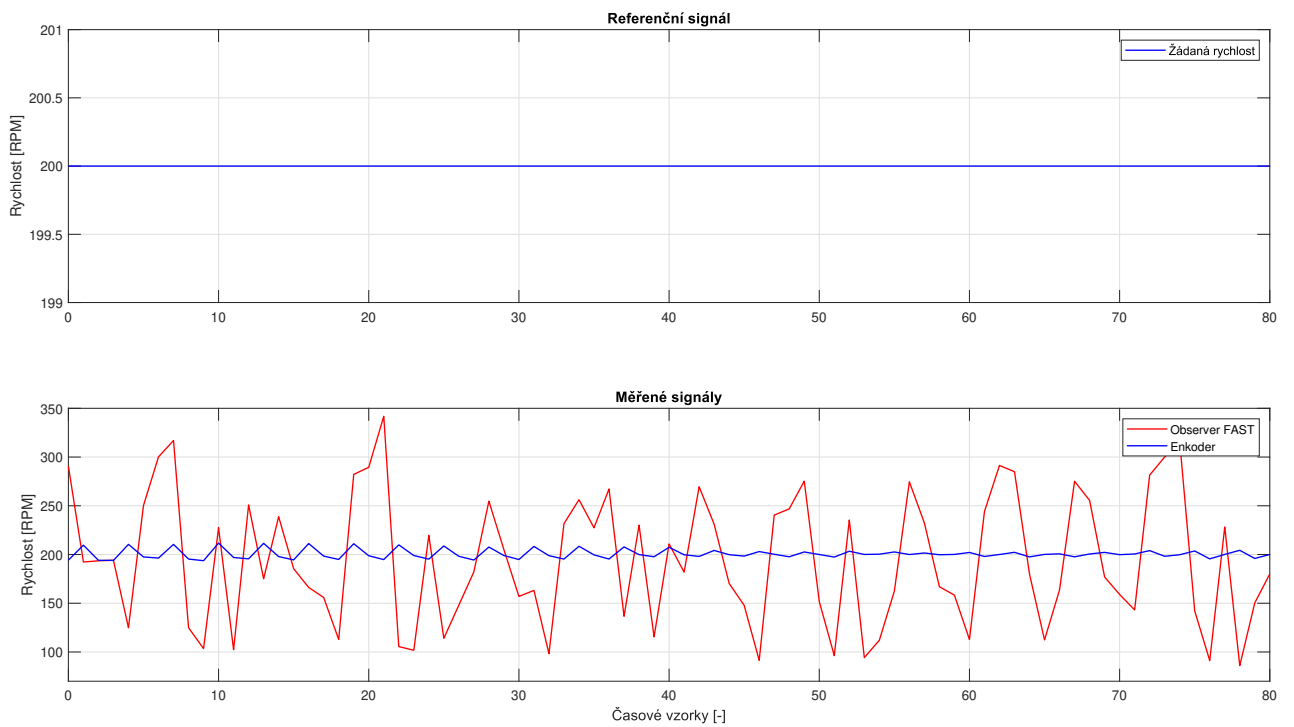
Na obrázku 3.22 je zobrazen průběh signálů při požadované rychlosti 200 RPM. Experiment dopadl dle očekávání, kdy při nízkých rychlostech má softwarový pozorovatel obvykle horší výsledky, ale jelikož motor není určený pro provoz v takto nízkých otáčkách, není to závažný nedostatek. Níže jsou uvedeny střední hodnoty signálů a směrodatné odchylky. Směrodatná odchylka signálu z pozorovatele činí 33.85 % celkové rychlosti. Střední hodnota signálu z enkodéru je zde naopak nově zatížena trvalou odchylkou, která činí 0.34 % celkové rychlosti.

$$\overline{FAST}(200RPM) = 194.5483 \text{ RPM}$$

$$\delta_{FAST}(200RPM) = 67.7278 \text{ RPM}$$

$$\overline{Enkoder}(200RPM) = 200.6212 \text{ RPM}$$

$$\delta_{Enkoder}(200RPM) = 4.9034 \text{ RPM}$$

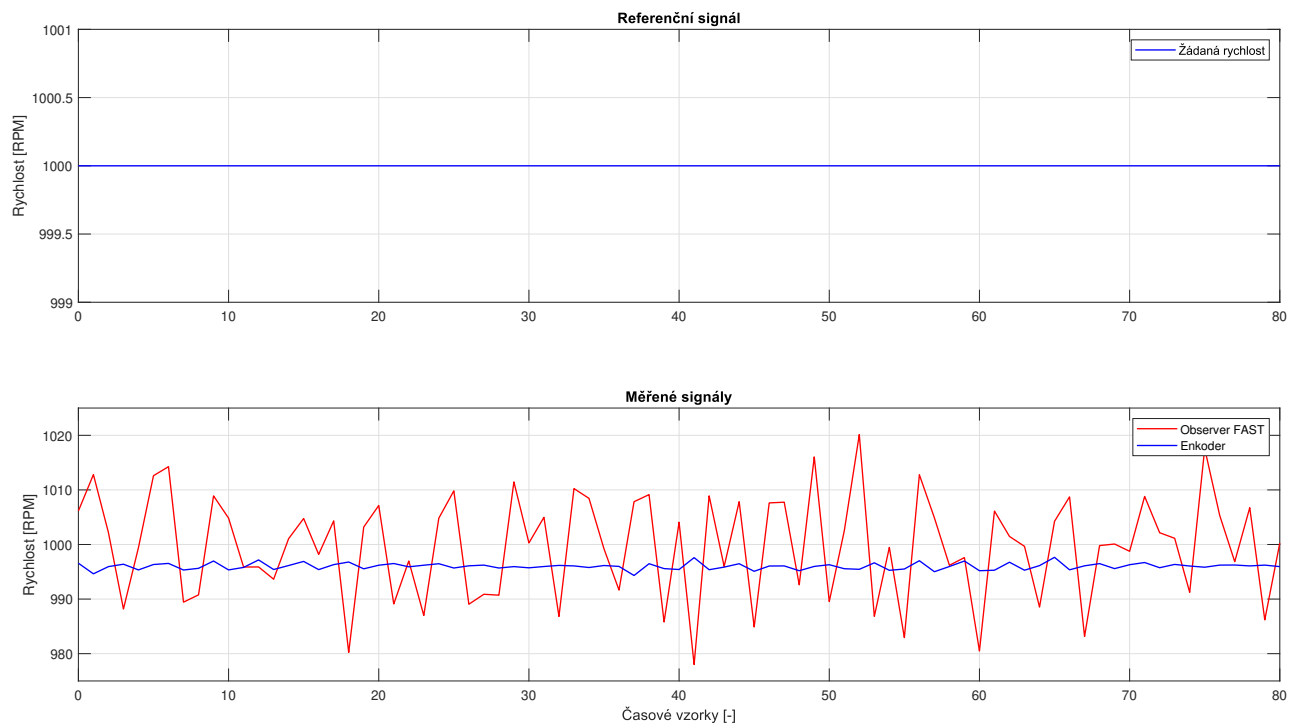


Obrázek 3.22: Průběhy: 200 RPM

Konstantní rychlost 1000 RPM

Na obrázku 3.23 je zobrazen průběh signálů při požadované rychlosti 1000 RPM. Rozdíl mezi směrodatnou odchylkou obou signálů je již výrazně menší, kdy směrodatná odchylka signálu z pozorovatele činí 0.97 % celkové rychlosti. Střední hodnota signálu z enkodéru je zatížena trvalou odchylkou, která činí 0.31 % celkové rychlosti, což bude komentováno při dalším měření. Níže jsou uvedeny střední hodnoty signálů a směrodatné odchylky.

$$\begin{aligned}\overline{FAST}(1000RPM) &= 999.6474 \text{ RPM} \\ \delta_{FAST}(1000RPM) &= 9.6686 \text{ RPM} \\ \overline{Enkoder}(1000RPM) &= 996.6474 \text{ RPM} \\ \delta_{Enkoder}(1000RPM) &= 0.6141 \text{ RPM}\end{aligned}$$

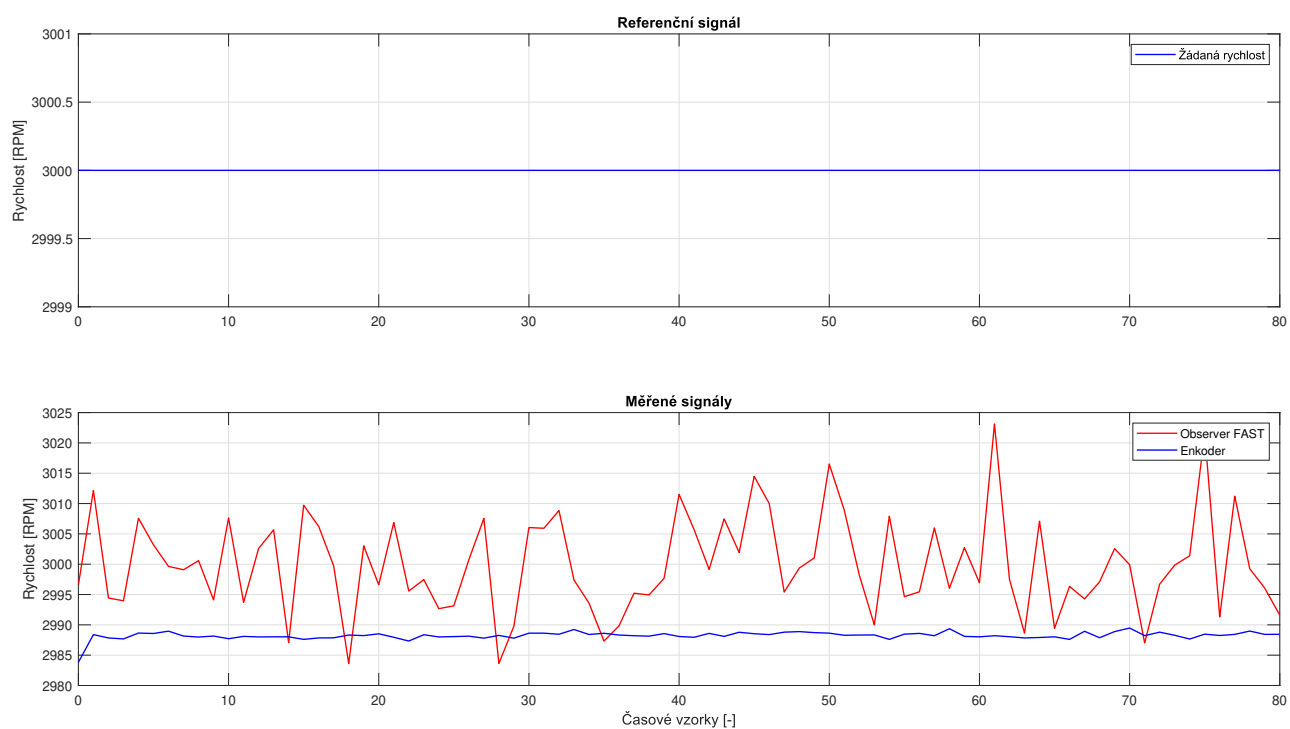


Obrázek 3.23: Průběhy: 1000 RPM

Konstantní rychlost 3000 RPM

Na obrázku 3.24 je zobrazen průběh signálů při požadované rychlosti 3000 RPM. Tato rychlost je uváděna u motoru jako provozní rychlost. Směrodatná odchylka signálu z pozorovatele je menší, než při předchozím experimentu a činí 0.27 % celkové rychlosti. Signál z enkodéru je zatížen trvalou odchylkou, která činí 0.4 % celkové rychlosti. Vzhledem k tomu, že signál z pozorovatele není zatížen významnou trvalou odchylkou, nepřesnost je vnášena při výpočtu rychlosti z pulzů polohy enkodéru. Níže jsou uvedeny střední hodnoty signálů a směrodatné odchylky.

$$\begin{aligned}\overline{FAST}(3000RPM) &= 2999.9 \text{ RPM} \\ \delta_{FAST}(3000RPM) &= 8.0619 \text{ RPM} \\ \overline{Enkoder}(3000RPM) &= 2988.2 \text{ RPM} \\ \delta_{Enkoder}(3000RPM) &= 0.6461 \text{ RPM}\end{aligned}$$



Obrázek 3.24: Průběhy: 3000 RPM

Konstantní rychlost 5000 RPM

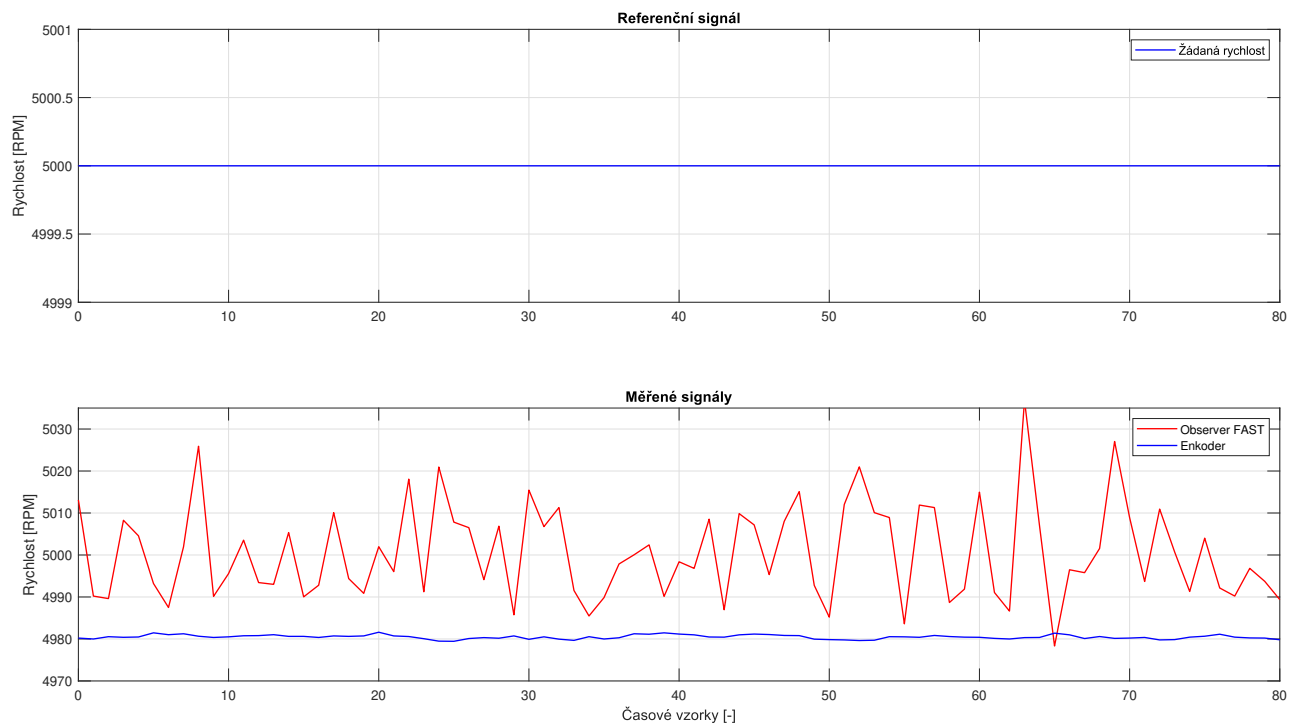
Na obrázku 3.25 je zobrazen průběh signálů při požadované rychlosti 5000 RPM. Tato rychlost je uváděna u motoru jako maximální rychlost. Směrodatná odchylka signálu z pozorovatele je opět dle očekávání menší, než při předchozím experimentu a činí 0.23 % celkové rychlosti. Signál z enkodéru je zatížen trvalou odchylkou, která činí 0.39 % celkové rychlosti. Níže jsou uvedeny střední hodnoty signálů a směrodatné odchylky.

$$\overline{FAST}(5000RPM) = 5000.6 \text{ RPM}$$

$$\delta_{FAST}(5000RPM) = 11.3745 \text{ RPM}$$

$$\overline{Enkoder}(5000RPM) = 4980.5 \text{ RPM}$$

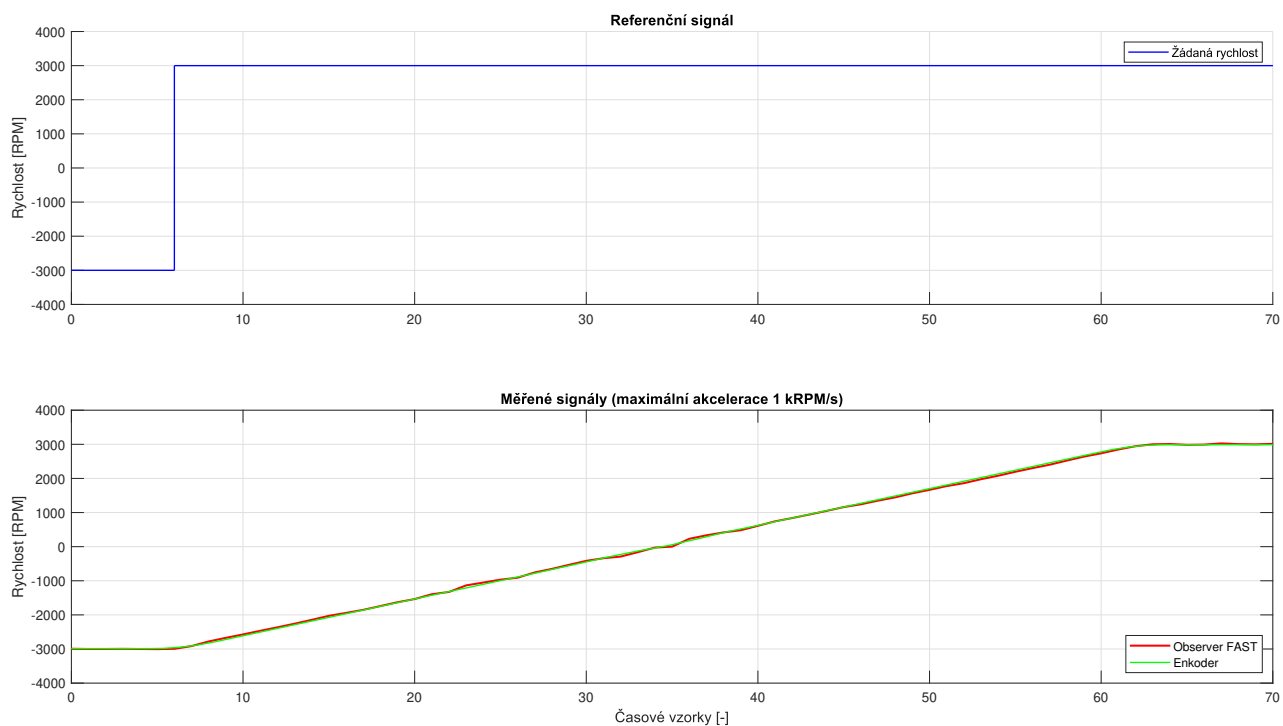
$$\delta_{Enkoder}(5000RPM) = 0.4828 \text{ RPM}$$



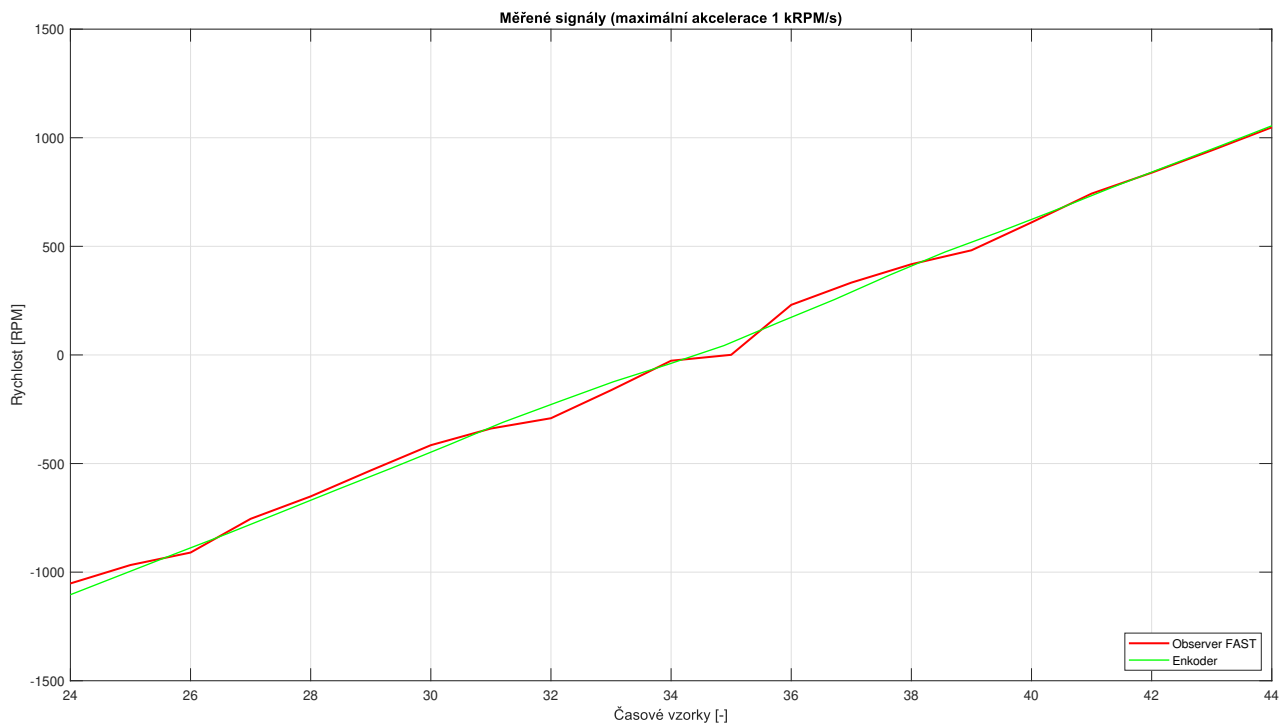
Obrázek 3.25: Průběhy: 5000 RPM

Rampa 1: -3000 RPM až 3000 RPM

Na obrázku 3.26 je zobrazen průběh signálů při skokové změně rychlosti z -3000 RPM na 3000 RPM. Požadavek na změnu rychlosti byl realizován jako náběžná rampa, kdy největší nepřesnosti nastaly při přechodu přes nulovou rychlost, tedy v oblasti velmi nízkých rychlostí a také nutnosti změnit směr rotace magnetického pole statoru. Motor má omezenou akceleraci a tento experiment byl prováděn při její hodnotě 1000 RPM/s. Na obrázku 3.27 je vidět detail průchodu přes nulovou rychlost a lze vidět větší zakolísání signálu u pozorovatele.



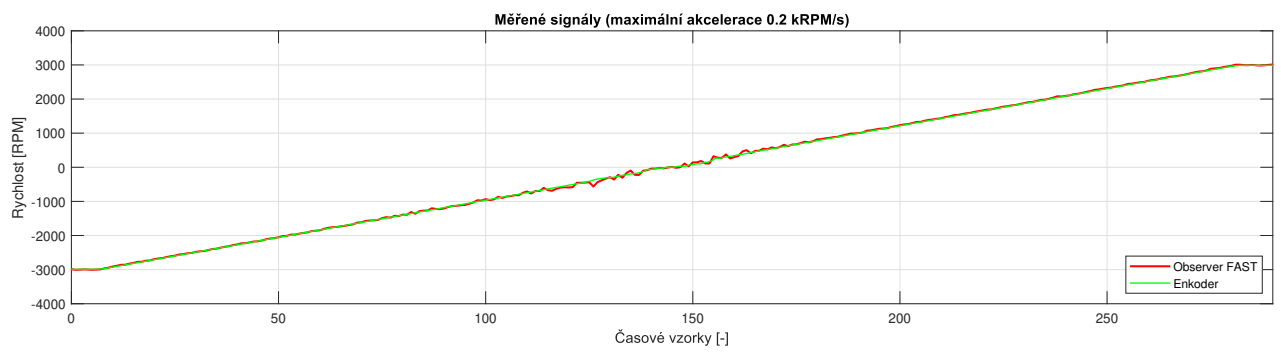
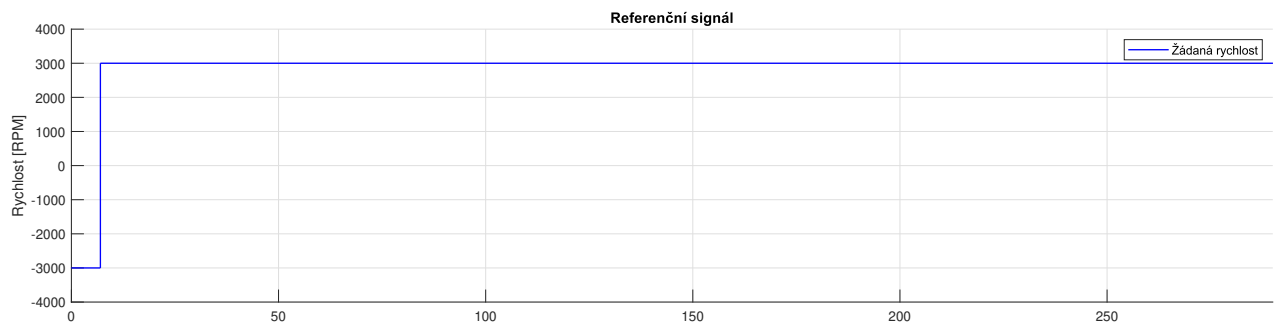
Obrázek 3.26: Průběhy: Rampa 1



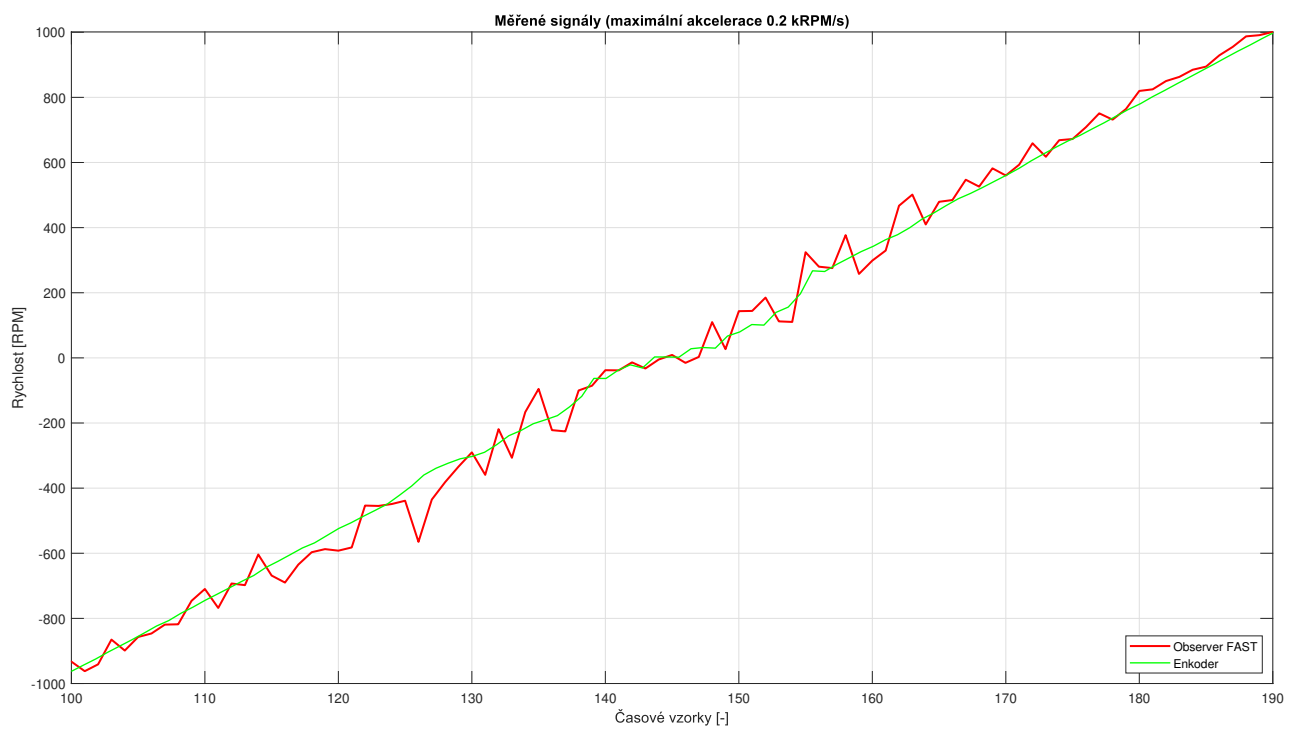
Obrázek 3.27: Detail: Rampa 1

Rampa 2: -3000 RPM až 3000 RPM

Na obrázku 3.28 je zobrazen průběh signálů opět při skokové změně rychlosti z -3000 RPM na 3000 RPM. Oproti předchozímu případu zde byla změněna pouze jedna skutečnost a to maximální akcelerace, jejíž hodnota činila 200 RPM/s. Na obrázku 3.29 je vidět detail průchodu přes nulovou rychlost a lze vidět řádově větší zakolísání signálu u pozorovatele než při předchozím případě, neboť motor v nižších otáčkách setrval v tomto případě déle. Velmi blízko změny směru otáčení, tedy okolo nulové rychlosti však jsou zřejmé výkmity také u signálu z enkodéru.



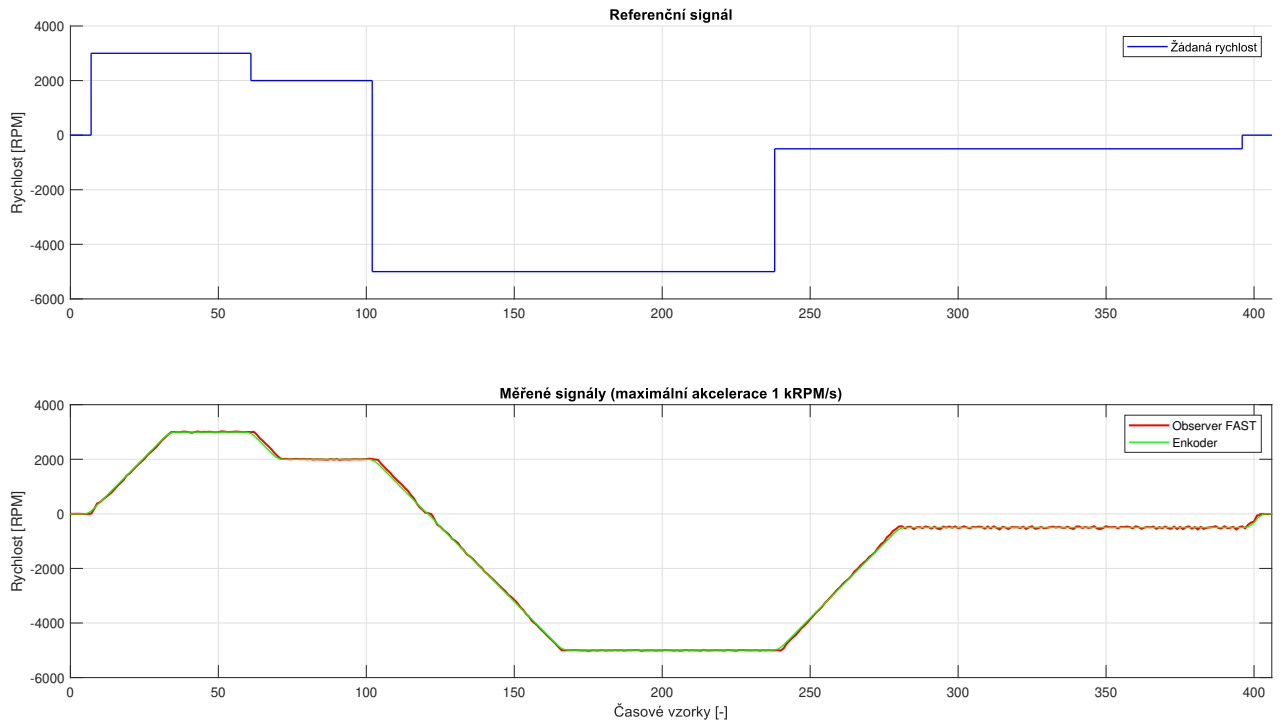
Obrázek 3.28: Průběhy: Rampa 2



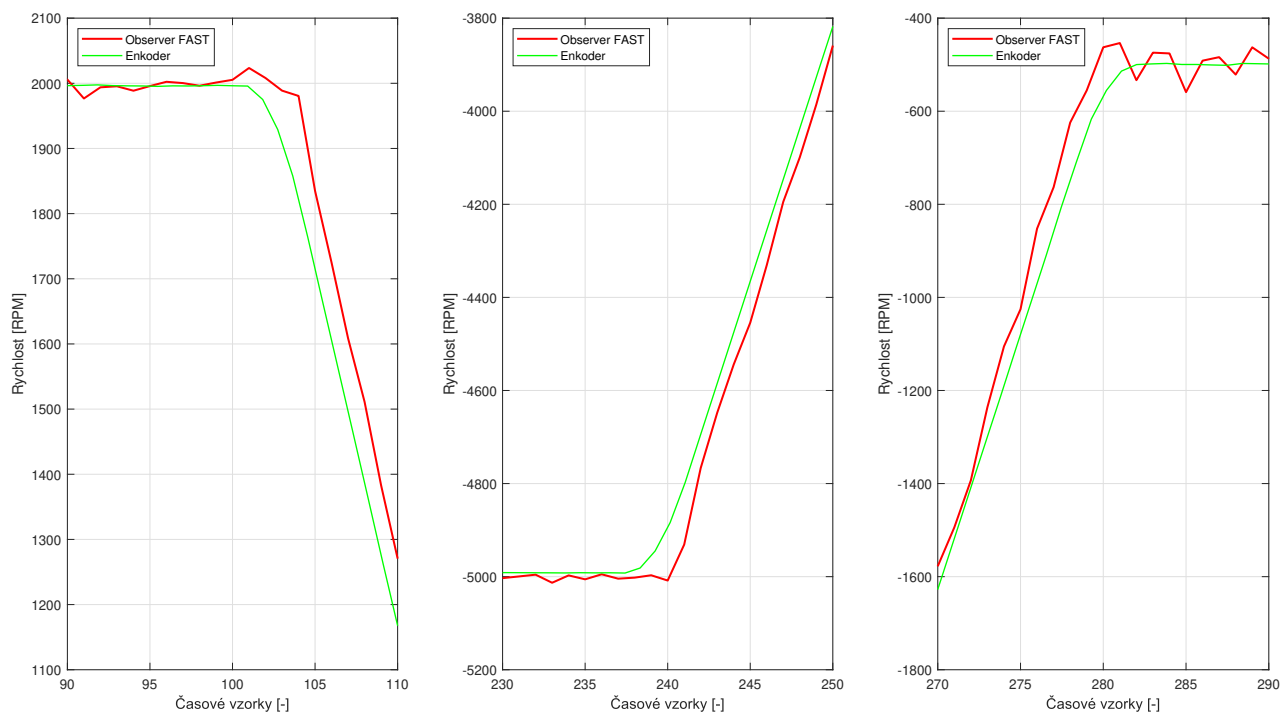
Obrázek 3.29: Detail: Rampa 2

Sekvence 1

V této části měl referenční signál podobu schodovité změny rychlostí. Sekvence probíhala v pořadí 0 RPM → 3000 RPM → 2000 RPM → -5000 RPM → -500 RPM → 0 RPM. Cílem této části bylo porovnat rychlost odezvy na změnu signálu. Výsledky lze vidět na obrázku 3.30, přičemž detail okamžiků, kdy nastala skoková změna referenčního signálu je na obrázku 3.31. Z detailu je patrné, že pozorovatel FAST má delší reakční dobu při změně rychlosti (při omezení akcelerace hodnotou 1 kRPM/s).



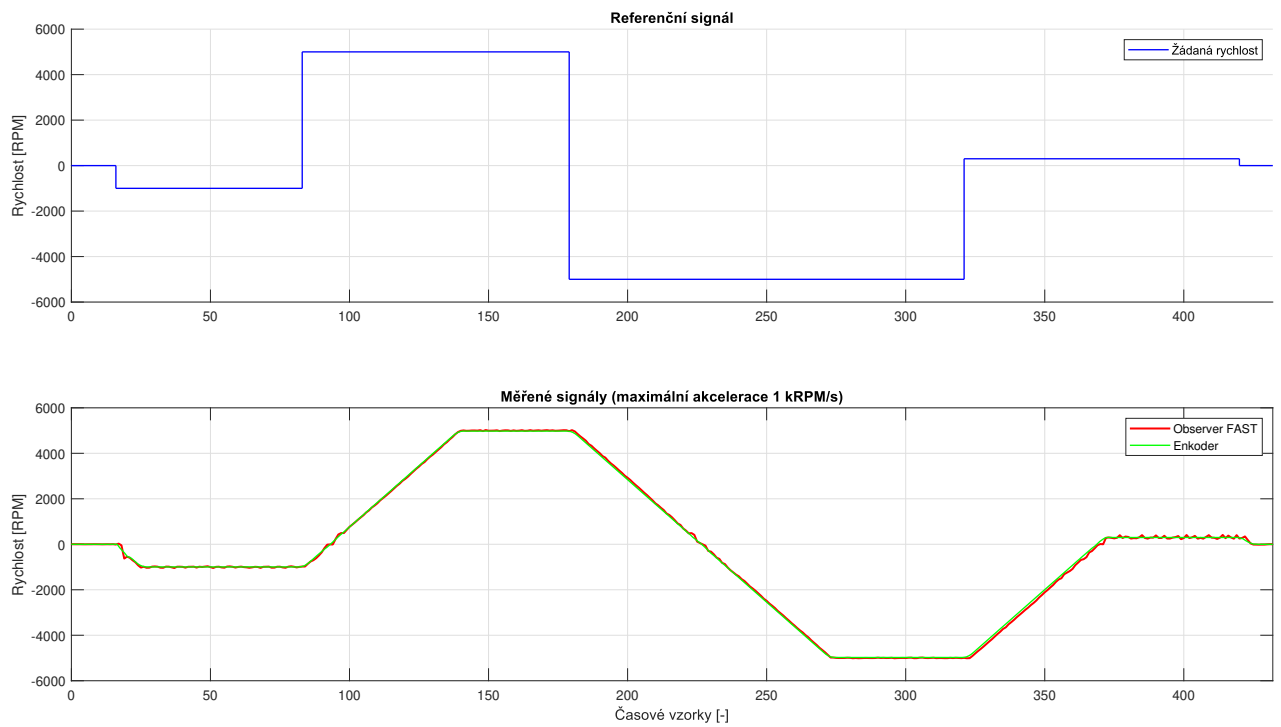
Obrázek 3.30: Průběhy: Sekvence 1



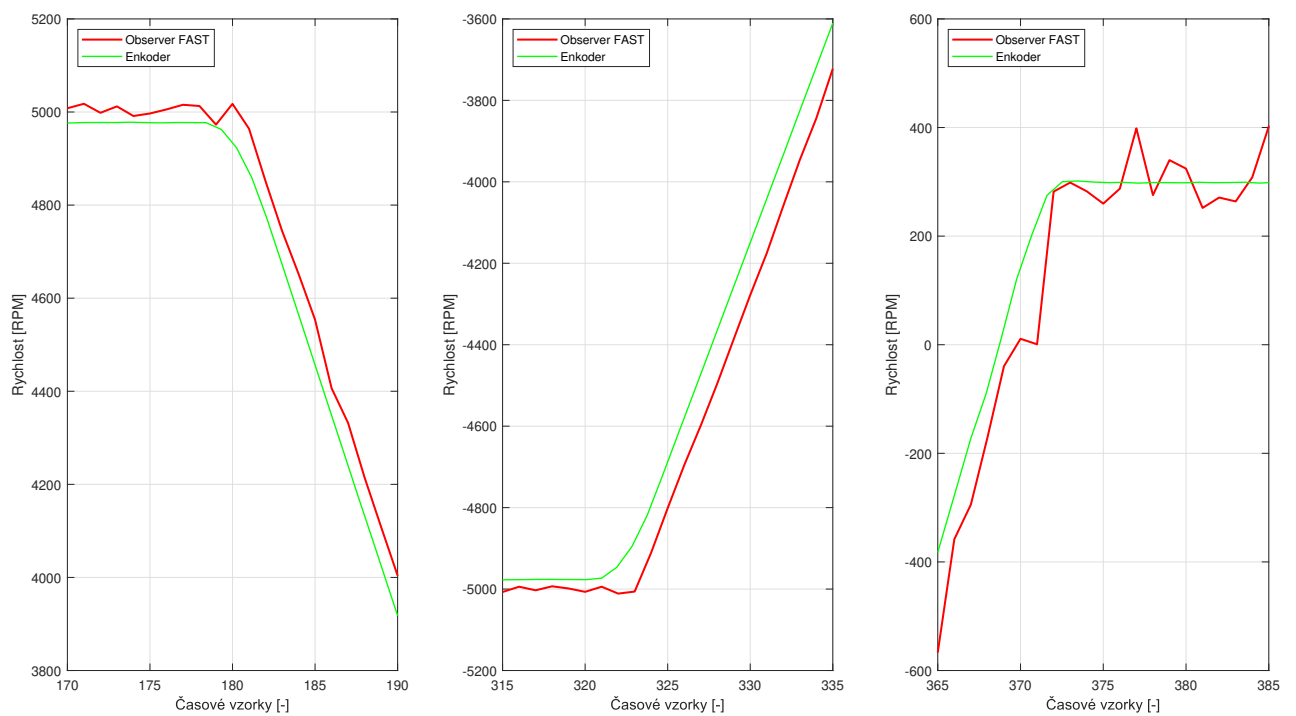
Obrázek 3.31: Detail: Sekvence 1

Sekvence 2

V této části měla sekvence rychlostí podobu $0 \text{ RPM} \rightarrow -1000 \text{ RPM} \rightarrow 5000 \text{ RPM} \rightarrow -5000 \text{ RPM} \rightarrow 300 \text{ RPM} \rightarrow 0 \text{ RPM}$. Také při této sekvenci docházelo k menšímu opoždění signálu z pozorovatele, oproti signálu z enkodéru, jak lze vidět na obrázku 3.32 a v detailu na 3.33 (opět je zde nutné zmínit omezení akcelerace hodnotou 1 kRPM/s).



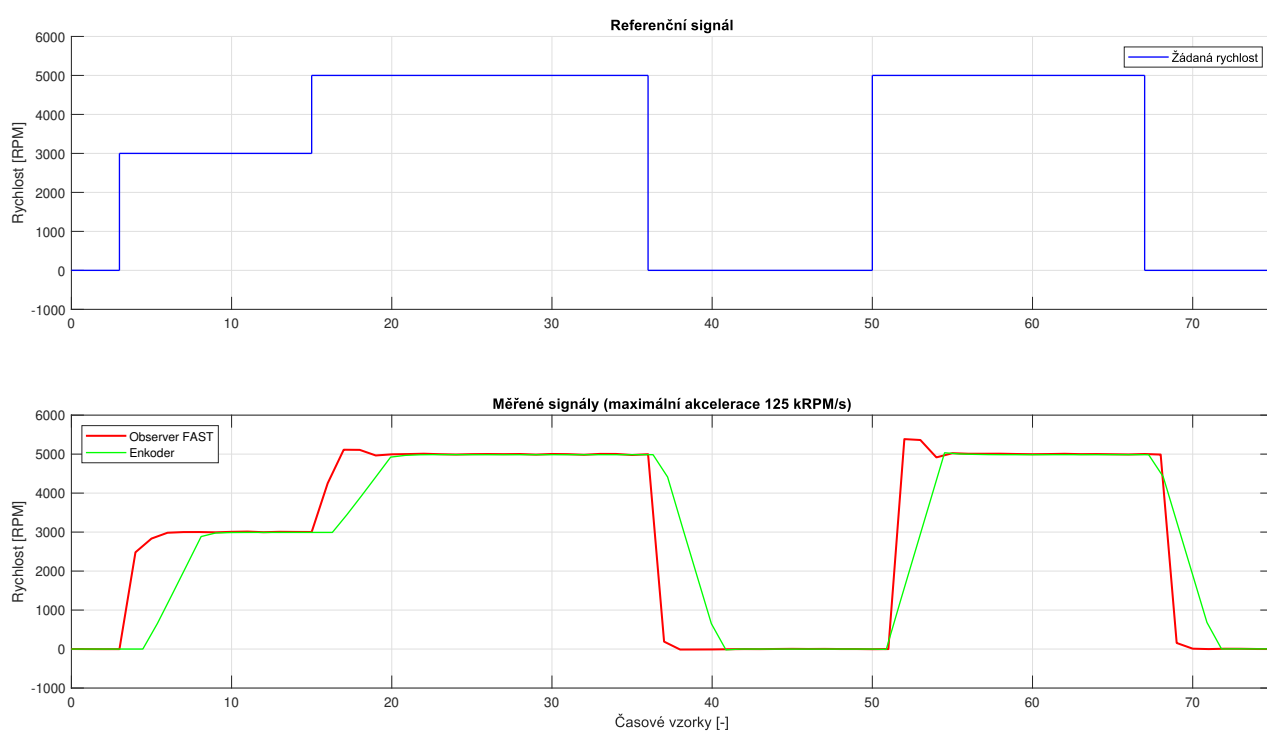
Obrázek 3.32: Průběhy: Sekvence 2



Obrázek 3.33: Detail: Sekvence 2

Sekvence 3

Poslední testování proběhlo při výrazně zvýšeném limitu maximální akcelerace, aby bylo dosaženo co nejrychlejších změn rychlosti a byly tak otestovány kritické podmínky pro získání signálu rychlosti. V této části měl referenční signál také podobu schodovité změny rychlostí a sekvence probíhala v pořadí 0 RPM → 3000 RPM → 5000 RPM → 0 RPM → 5000 RPM → 0 RPM. Tato sekvence proběhla za méně než 70 časových vzorků, tedy za méně než 7 sekund. Zpoždění signálu z enkodéru, které je způsobeno principem jeho získání se při omezení akcelerace výrazně neprojevuje. Při simulované nekonečné akceleraci, která byla nyní nastavena na programem dovolené maximum 125 kRPM/s a také velmi krátkém trvání sekvence je však již vidět zpoždění signálu z enkodéru vůči signálu z pozorovatele. Průběhy signálů a zmíněná fázová posunutí lze vidět na obrázku 3.34.



Obrázek 3.34: Průběhy: Sekvence 3

3.3.5 Závěr srovnání

V části testování bylo provedeno celkem devět porovnávacích průběhů. Nejprve byly porovnávány signály z kvadratického enkodéru a ze softwarového pozorovatele FAST při konstantní hodnotě rychlosti. Při rychlostech od 0 RPM až do provozní rychlosti 3000 RPM či maximální rychlosti 5000 RPM bylo zjištěno, že při nízkých rychlostech dominuje kvalita signálu, který je získáván z enkodéru vůči signálu z pozorovatele, který má velkou hodnotu směrodatné odchylky. Se zvyšujícími se otáčkami je kvalita signálu z pozorovatele řádově lepší a při provozních otáčkách je odchylka tohoto signálu menší než 0.5 %, což je přesnost naprosto dostačující. Naopak signál z enkodéru je zatížený trvalou odchylkou, která se se zvyšujícími otáčkami motoru zvětšuje, ale nepřekračuje 0.5 % celkové rychlosti. Vzhledem k tomu, že princip výpočtu je dělení pevného počtu pulzů uběhnutým časem mezi těmito pulzy, odchylka je vnášena nepřesností v časové informaci při tomto výpočtu.

V dalších dvou testech byl zkoumán přechod přes nulovou rychlost při různých hodnotách maximální povolené akcelerace motoru. Čím rychleji byla překonána nulová rychlost a ostatní nízké rychlosti v jejím okolí, tím byl průběh plynulejší. Obecně při nižších otáčkách byl nepřesnější signál z pozorovatele, ale také zde bylo vidět, že při velmi nízkých otáčkách, jmenovitě 200 RPM a nižších, byl velmi nepřesný také signál z enkodéru.

Poslední tři testovací sekvence byly provedeny se zaměřením na zpoždění signálů vůči sobě. Na začátku bylo uvedeno, že byla vybrána metoda pro výpočet rychlostního signálu z enkodéru z 500 polohových pulzů enkodéru. Při počtu 8192 pulzů na jednu otáčku to umožňuje výpočet rychlosti už při přibližně 6 % jedné otáčky, tedy zpoždění není značné, ale je nutné s ním počítat při volbě metody zpracování signálu. Získání signálu ze softwarového pozorovatele předchází velké množství kroků, mezi které patří změna napětí a proudu na vinutích statoru, následně převedení těchto měření analogově-digitálními převodníky a provedení výpočetního algoritmu. Vzhledem k těmto skutečnostem není při omezení akcelerace (případy Sekvence 1 a Sekvence 2) zpoždění výpočtu rychlosti z enkodéru viditelné, přičemž fázově zpožděný je zde signál z pozorovatele. Změna nastává v případě velmi strmé změny rychlosti, která simuluje provoz v dynamicky náročných aplikacích. V tomto případě se již zpoždění výpočtu signálu z enkodéru zvolenou metodou projevuje a rychlejší odezvu poskytuje softwarový pozorovatel. Z výsledků těchto srovnávacích testů lze vidět, že pro specifickou aplikaci může být dle funkčních potřeb vhodné použít jednu, či druhou technologii. Nelze totiž zcela univerzálně použít jednu technologii na všechny případy použití, neboť vždy bude při určitých podmínkách jedna technologie vhodnější a při jiných podmínkách bude tato technologie v porovnání s jinou méně vhodná. Pro správný výběr technologie je tedy potřebné zvážit požadavky na přesnost, typ a náročnost aplikace a také finanční možnosti při realizování projektu.

4. Závěr

V úvodní části této práce byla přiblížena problematika řízení elektrických pohonů, které jsou často využívány ve všech odvětvích průmyslu. Úvodní kapitola se věnovala důvodům, kvůli kterým jsou synchronní střídavé motory stále častěji volbou pro aplikace, kde dříve převládaly motory stejnosměrné, či střídavé asynchronní. Také zde bylo zmíněno časté uplatnění synchronních motorů s permanentními magnety v konkrétních aplikacích spolu s jejich výhodami, díky kterým nahradily dříve používané motory. Dále zde byly zmíněny metody řízení synchronních strojů a jejich vývoj a postupný přechod na plně softwarovou podobu. Druhá část úvodní kapitoly obsahuje cíle práce tak, jak jsou obsažené v zadání diplomové práce. Třetí část úvodní kapitoly pak definuje a vysvětluje některé pojmy, týkající se této problematiky, případně slouží pro přiřazení důležitých v anglické literatuře se vyskytujících pojmů k jejich českému ekvivalentu.

Druhá kapitola této práce se věnuje teoretickému principu řízení synchronních elektrických pohonů s permanentními magnety. Ve stručnosti je zmíněno skalární řízení, ale většina práce se zaměřuje na vektorové řízení, jehož vývoj je v této době, vzhledem ke stále klesající hodnotě výpočetního výkonu dominantní. Nejprve je popsán matematický aparát pro transformace mezi souřadnicovými systémy pomocí Clarkovy a Parkovy transformace a jejich inverzemi. Následuje matematický model synchronního motoru s permanentními magnety a vztahy, které pro něj platí v různých souřadnicových systémech. V další podkapitole je shrnut princip vektorového řízení jako takového a popis jednotlivého postupu, při jeho aplikaci. Další část druhé kapitoly přibližuje velmi obsáhlou část bezsensorového řízení. Jsou zde rozděleny a popsány metody pro získání polohového a rychlostního signálu. V poslední části této kapitoly je sestaven a naladěn model vektorového řízení se dvěma řídicími smyčkami v prostředí Simulink. Model byl co nejvíce optimalizován, aby se přiblížil skutečnému motoru, použitému v realizaci experimentu ve třetí kapitole.

Třetí kapitola nejprve představuje použité vybavení, především hardware a software, který byl použit při aplikování vektorového řízení. V této části bylo tedy realizováno řízení malého synchronního servomotoru s permanentními magnety a je zde uveden provedený postup včetně nastavování potřebných parametrů. Tento postup může sloužit jako návod pro další použití při realizaci řízení podobných motorů. V druhé části této kapitoly je samotný experiment, kdy po úspěšném nastavení metod pro získávání rychlostního signálu z enkodéru a softwarového pozorovatele byly tyto dva signály porovnány při několika různých nastaveních experimentu.

V návaznosti na výsledky, dosažené v této práci, je možné například realizovat další smyčku v řídicím algoritmu. Řízení polohy je dalším stupněm řízení, jehož cílem je umožnit přesné polohování hřídele. Využití mikroprocesorových plaftorem od firmy Texas Instruments pro tento účel vyžaduje typ, který podporuje kromě InstaSPIN technologie také technologii InstaMOTION. Je však nutné podotknout, že nastavení softwaru, který je k vývoji těchto technologií zapotřebí a vyladění systému jako celku ke správné funkčnosti je časově velmi náročné.

Závěrem může být řečeno, že cíle této práce, uvedené v první kapitole, byly splněny.

Seznam literatury

- [1] WILSON, Dave. Teaching Old Motors New Tricks - Part 2: Field Oriented Control of Permanent Magnet Synchronous Motors. [online]. [cit. 2019-07-13]. Dostupné z: <https://training.ti.com/teaching-old-motors-new-tricks-part-2-field-oriented-control-permanent-magnet-synchronous-motors>
- [2] Grasblum, Pavel a Prokop, Libor.: 3-Phase Synchronous Motor Vector Control Using 56F80x, 56F8100, or 56F8300 Device, [online]. [cit. 2019-07-13]. Dostupné z <https://cache.freescale.com/files/product/doc/AN1931.pdf>
- [3] RYDLO, P.: Řízení elektrických střídavých pohonů, vyd. Technická univerzita v Liberci, 2007. 129 s. , ISBN 978-80-7372-223-4
- [4] HOŘAVA, J.: Skalární versus vektorové řízení synchronních motorů. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2009. 42 s. Vedoucí bakalářské práce Ing. Vladimír Hubík.
- [5] Novák, J.: Uplatnění synchronních strojů v dopravní technice, [online]. [cit. 2019-07-13]. Dostupné z <http://www.odbornecasopisy.cz/elektro/casopis/tema/uplatneni-synchronnich-stroju-v-dopravni-technice-1-cast--13092>
- [6] The Mathworks, Inc. Permanent Magnet Synchronous Motor. [online]. [cit. 2019-07-13]. Dostupné z: <https://www.mathworks.com/help/physmod/sps/ref/pmsm.html>
- [7] LAUNCHXL-F28069M User's Guide [online]. 2019 [cit. 2019-06-15]. Dostupné z: <http://www.ti.com/lit/ug/sprui11b/sprui11b.pdf>
- [8] Sigma Mini Servo System Product Catalog [online]. 1999 [cit. 2019-06-15]. Dostupné z: https://www.galco.com/techdoc/yask/sgmm-als312_pg.pdf
- [9] BOOSTXL-DRV8305EVM User's Guide [online]. 2017 [cit. 2019-06-15]. Dostupné z: <http://www.ti.com/lit/pdf/slvuai8>
- [10] RECH, Pavel. Bezsenzorové řízení střídavého regulovaného pohonu se synchronním motorem s permanentními magnety. Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2010. Školitel disertační práce prof. Ing. Pavel Brandštetter, CSc.
- [11] MotorWare Software [online]. 2016 [cit. 2019-06-15]. Dostupné z: <http://www.ti.com/tool/MOTORWARE>
- [12] CCS User's Guide [online]. 2019 [cit. 2019-06-15]. Dostupné z: http://dev.ti.com/tirex/explore/node?node=ABOji8.duIdHhzsJkKch3A__FUz-xrs__LATEST
- [13] InstaSPIN-FOC and InstaSPIN-MOTION User's Guide [online]. 2017 [cit. 2019-06-15]. Dostupné z: <http://www.ti.com/lit/ug/spruhj1g/spruhj1g.pdf>
- [14] InstaSPIN Projects and Labs User's Guide [online]. 2014 [cit. 2019-06-16]. Dostupné z: https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/902/instaspin_5F00_labs.pdf
- [15] Park, Inverse Park and Clarke, Inverse Clarke Transformations MSS Software Implementations User Guide [online]. 2013 [cit. 2019-07-16]. Dostupné z: https://www.microsemi.com/document-portal/doc_view/132799-park-inverse-park-and-clarke-inverse-clarke-transformations-mss-software-implementation-user-guide
- [16] K. Ogata. Modern control engineering. 5th ed. Boston, Prentice Hall, 2010

- [17] MALIŠ, Martin. Algoritmy bezsnímačového řízení pohonu se synchronním motorem. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 110 s. Vedoucí práce doc. Ing. Pavel Václavek, Ph.D.
- [18] LAUNCHXL-F28027F Quick Start Guide [online]. 2013 [cit. 2019-07-21]. Dostupné z: <http://www.ti.com/lit/ml/sprz413/sprz413.pdf>
- [19] How to use a quadrature encoder [online]. [cit. 2019-07-21]. Dostupné z: <https://cdn.sparkfun.com/datasheets/Robotics/How%20to%20use%20a%20quadrature%20encoder.pdf>
- [20] Novák, Martin a Novák, Zdeněk a Chyský, Jan. Bezsenzorové řízení synchronních motorů s permanentními magnety [online]. 2013 [cit. 2019-07-21]. Dostupné z: https://www.researchgate.net/publication/317569939_Bezsenzorove_rizeni_synchronnich_motoru_s_permanentnimi_magnety
- [21] Siva Malla. Vector Control of Permanent Magnet Synchronous Motor (PMSM) [online]. 2012 [cit. 2019-07-21]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/39605-vector-control-of-permanent-magnet-synchronous-motor-pmsm>
- [22] WILSON, Dave. Teaching Old Motors New Tricks - Part 3: Space Vector Modulation, Field Weakening, d-q Axis Decoupling, and Operation of AC Induction Motors. [online]. [cit. 2019-07-13]. Dostupné z: <https://training.ti.com/teaching-old-motors-new-tricks-part-3-space-vector-modulation-field-weakening-d-q-axis-decoupling>

Seznam příloh na CD

Na přiloženém CD se nacházejí tyto adresáře:

- Programy
 - Obsahuje programy pro zpracování dat z enkodéru a simulační model řízení.
- PDF verze DP