



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODERNÍ STRUKTURY ŘÍZENÍ SERVO-SYSTÉMŮ SE STŘÍDAVÝMI POHONY

MODERN ALGORITHMS OF AC SERVO-DRIVES CONTROL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN HRNČÁREK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PAVEL VÁCLAVEK, Ph.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Martin Hrnčárek

ID: 84358

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Moderní struktury řízení servosystémů se střídavými pohony

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s návrhem řídicích struktur servosystémů s 3-f asynchronními motory a 3-f synchronními motory s permanentním magnetem. Zaměřte se jak na tradiční algoritmy (vektorové řízení), tak i moderní algoritmy (DTC, využití zpětnovazební linearizace). V prostředí Matlab-Simulink vytvořte ucelenou knihovnu bloků pro simulace řídicích struktur. Vybrané algoritmy rovněž implementujte na procesoru řady Freescale 56F800E Hybrid Controller a proveďte jejich simulační ověření metodou "processor in the loop".

DOPORUČENÁ LITERATURA:

Caha, Z., Černý, M.: Elektrické pohony

Šubrt, J.: Elektrické regulační pohony

další dle průběžných pokynů vedoucího práce

Termín zadání: 8.2.2010

Termín odevzdání: 24.5.2010

Vedoucí práce: doc. Ing. Pavel Václavěk, Ph.D.

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Moderní struktury řízení servosystémů se střídavými pohony

Diplomová práce

Studijní obor: Kybernetika, automatizace a měření
Student: Bc. Martin Hrnčárek
Vedoucí práce: doc. Ing. Pavel Václavek, Ph.D.

Abstrakt:

Diplomová práce pojednává o moderních řídicích algoritmech asynchronního a synchronního motoru s permanentními magnety. Zaměřuje se na algoritmy vektorového řízení, přímého řízení momentu a zpětnovazební linearizace (vstup - výstup). Zabývá se také tvorbou knihovny algoritmů v prostředí Matlab – Simulink a její použití. V poslední části této práce je popsána implementace vybraných algoritmů na procesor řady Freescale 56F800E.

Klíčová slova:

Asynchronní motor, Synchronní motor, ASM, PMSM, Clarkova transformace, Parkova transformace, Vektorové řízení, Přímé řízení momentu, DTC, Zpětnovazební linearizace, Decoupling

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL, MEASUREMENT AND INSTRUMENTATION

Modern Algorithms of AC Servo-drives Control

Master's Thesis

Specialization of study: Cybernetics, Control and Measurement
Student: Bc. Martin Hrnčárek
Supervisor: doc. Ing. Pavel Václavěk, Ph.D.

Abstract:

This master thesis deals with the modern algorithms of the asynchronous induction machine and permanent magnet induction machine. It focuses at vector control, direct torque control and input-output feedback linearization. It also describes a library creation and its usages in the Matlab – Simulink environment. The final part of this thesis contains an implementation of chosen algorithms on the DSC Freescale 56F800E family.

Key words:

Asynchronous induction machine, Synchronous induction machine, ACIM, PMSM, Clark's transformation, Park's transformation, Vector control, Direct torque control, Feedback linearization, Decoupling

Bibliografická citace

HRNČÁREK, M. *Moderní struktury řízení servosystémů se střídavými pohony*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 93 s. Vedoucí diplomové práce doc. Ing. Pavel Václavek, Ph.D.

P r o h l á š e n í

„Prohlašuji, že svou diplomovou práci na téma Moderní struktury řízení servosystémů se střídavými pohony jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **24. května 2010**

.....

podpis autora

P o d ě k o v á n í

Děkuji svému vedoucímu doc. Ing. Pavlovi Václavkovi, Ph.D za konzultace, cenné rady a podnětné připomínky při vypracování této diplomové práce. Děkuji svým rodičům za jejich podporu a vytvoření ideálních podmínek ke studiu.

V Brně dne: **24. května 2010**

.....

podpis autora

OBSAH

1. ÚVOD	9
2. ASYNCHRONNÍ MOTOR (ASM)	11
2.1 MATEMATICKÝ MODEL ASYNCHRONNÍHO MOTORU	12
2.1.1 <i>Prostorový vektor</i>	14
2.1.2 <i>Clarkova transformace</i>	15
2.1.3 <i>Parkova transformace</i>	16
2.1.4 <i>Odvození modelu pro obecný souřadnicový systém</i>	17
2.2 PARAMETRY ASYNCHRONNÍHO MOTORU PRO SIMULACE	19
3. SYNCHRONNÍ MOTOR S PERMANENTNÍM MAGNETEM (PMSM).....	20
3.1 MATEMATICKÝ MODEL PMSM	20
3.1.1 <i>Matematický model PMSM v d-q a α-β souřadnicích</i>	21
3.2 PARAMETRY SYNCHRONNÍHO MOTORU PRO SIMULACE	23
4. VEKTOROVÉ ŘÍZENÍ	24
4.1 VEKTOROVÉ ŘÍZENÍ ASYNCHRONNÍHO MOTORU	24
4.1.1 <i>Výpočet magnetického toku</i>	25
4.1.2 <i>„Decoupling“ pro ASM</i>	26
4.1.3 <i>Návrh regulátorů</i>	27
4.1.4 <i>Napěťové, proudové omezení a „anti-windup“</i>	28
4.1.5 <i>Simulace vektorového řízení asynchronního motoru</i>	30
4.2 VEKTOROVÉ ŘÍZENÍ PMSM	33
4.2.1 <i>„Decoupling“ pro PMSM</i>	33
4.2.2 <i>Návrh regulátorů</i>	33
4.2.3 <i>Simulace vektorového řízení PMSM motoru</i>	35
5. PŘÍMÉ ŘÍZENÍ MOMENTU (DTC)	38
5.1 PŘÍMÉ ŘÍZENÍ MOMENTU ASYNCHRONNÍHO MOTORU	39
5.1.1 <i>Proudové omezení u DTC</i>	41
5.1.2 <i>Simulace přímého řízení momentu asynchronního motoru</i>	42
5.2 PŘÍMÉ ŘÍZENÍ MOMENTU PMSM	45
5.2.1 <i>Simulace přímého řízení momentu PMSM motoru</i>	45
6. ZPĚTNOVAZEBNÍ LINEARIZACE (VSTUP - VÝSTUP).....	47
6.1 ŘÍZENÍ ASYNCHRONNÍHO MOTORU POMOCÍ ZPĚTNOVAZEBNÍ LINEARIZACE	51
6.1.1 <i>Simulace řízení asynchronního motoru pomocí zpětnovazební linearizace</i>	56

6.2	ŘÍZENÍ PMSM MOTORU POMOCÍ ZPĚTNOVAZEBNÍ LINEARIZACE	59
6.2.1	<i>Proudové omezení</i>	63
6.2.2	<i>Simulace řízení PMSM pomocí zpětnovazební linearizace</i>	64
7.	KNIHOVNA ŘÍDICÍCH ALGORITMŮ PRO MATLAB - SIMULINK.....	67
7.1	VYTVOŘENÍ KNIHOVNY PRO MATLAB - SIMULINK	67
7.2	VYTVOŘENÍ KNIHOVNÍCH FUNKČNÍCH BLOKŮ.....	67
7.2.1	<i>„Mask Editor“</i>	68
7.3	TVORBA GRAFICKÉHO UŽIVATELSKÉHO ROZHRAŇÍ	69
7.4	NÁPOVĚDA MATLABU.....	69
7.5	INSTALACE KNIHOVNY.....	69
7.6	POPIS JEDNOLIVÝCH BLOKŮ KNIHOVNY - GUI.....	70
7.6.1	<i>Vektorové řízení ASM – GUI</i>	71
7.6.2	<i>Vektorové řízení PMSM – GUI</i>	72
7.6.3	<i>Přímé řízení momentu ASM a PMSM – GUI</i>	73
7.6.4	<i>Zpětnovazební linearizace ASM - GUI</i>	74
7.6.5	<i>Zpětnovazební linearizace PMSM - GUI</i>	75
8.	IMPLEMENTACE VYBRANÝCH ALGORITMŮ NA PROCESOR ŘADY FREESCALE 56F800E.....	76
8.1	PROCESOR MC56F8346 FIRMY FREESCALE	76
8.2	SFIO TOOLBOX.....	77
8.3	IMPLEMENTACE VEKTOROVÉHO ŘÍZENÍ PMSM	78
8.3.1	<i>Průběh vektorového řízení PMSM na procesoru</i>	80
8.4	IMPLEMENTACE ZPĚTNOVAZEBNÍ LINEARIZACE ŘÍZENÍ PMSM	82
8.4.1	<i>Průběh řízení PMSM pomocí zpětnovazební linearizace</i>	84
8.5	ŘÍZENÍ REÁLNÉHO PMSM POMOCÍ ZPĚTNOVAZEBNÍ LINEARIZACE.....	86
9.	ZÁVĚR.....	87
10.	SEZNAM POUŽITÉ LITERATURY.....	89
11.	SEZNAM ZKRATEK A POUŽITÝCH SYMBOLŮ	91
12.	SEZNAM PŘÍLOH	93

1. ÚVOD

Obliba střídavých pohonů na trhu průmyslové automatizace neustále stoupá, jelikož představují perspektivní směr ve vývoji elektrických regulačních pohonů zejména v oblasti úspory elektrické energie a snížení nákladů na údržbu. Mezi další výhodné vlastnosti patří jednoduchá konstrukce, robustnost, vysoká spolehlivost, dostupnost v širokém spektru výkonových řad a přijatelná pořizovací cena. Pro řízení střídavých regulačních pohonů je nutno aplikovat poměrně náročné řídicí algoritmy, které jsou nejčastěji odvozeny od matematického modelu motoru např. vektorové řízení, přímé řízení momentu atd. Model vzniká řešením soustavy diferenciálních rovnic, za předpokladu zjednodušujících podmínek, popisujících mechanické a elektromagnetické vlastnosti motoru. Avšak implementace těchto složitých algoritmů, díky technické úrovni výkonové a mikroprocesorové elektroniky v posledním desetiletí, nepředstavuje problém a umožňuje realizaci střídavých pohonů s regulací rychlosti, polohy, točivého momentu či jiné veličiny technologického procesu.

Splnění cílů této práce vyžaduje proniknout do tvorby matematických modelů ASM (kapitola 2.1.4), PMSM (kapitola 3.1.1) a vytvořit jejich modely v prostředí Matlab - Simulink. Po úspěšném vytvoření matematických modelů v Simulinku se může přistoupit k implementaci řídicích algoritmů pro vytvořené modely. Všechny algoritmy v této práci byly vytvořeny pro oba typy motorů a obsahují také grafické průběhy řízení pořízené při simulacích s patřičným komentářem. Prvním algoritmem je klasické vektorové řízení popsané v kapitole 4. Následující 5. kapitola popisuje algoritmus přímého řízení momentu označované zkratkou DTC. Poslední algoritmus, popsán v 6. kapitole, je založen na teorii zpětnovazební linearizace (vstup - výstup). Kapitola obsahuje přehled teoretických znalostí nutných k pochopení algoritmů a následnou aplikaci na ASM a PMSM. Řídicí algoritmy jsou značně složité s velkým množstvím nastavení, proto byl vytvořen pro každý algoritmus knihovní blok s grafickým uživatelským rozhraním (GUI), jenž velice zpřehledňuje, urychluje a usnadňuje použití algoritmů k simulacím. Tvorba knihovny je popsána společně

s popisem GUI v 7. kapitole. Veškeré simulace obsažené v kapitolách 4, 5, 6 byly prováděny pomocí knihovnic bloků popsaných v 7. kapitole. Posledním bodem zadání byla implementace vybraných algoritmů na procesor. Mezi vybrané algoritmy patřilo vektorové řízení a řízení pomocí zpětnovazební linearizace PMSM. Postup implementace na procesor společně s průběhy řízení procesorem a komentáři je popsán v 8. kapitole. Algoritmus zpětnovazební linearizace byl navíc vyzkoušen na reálném PMSM od firmy TG Drives (kapitola 8.5).

2. ASYNCHRONNÍ MOTOR (ASM)

Princip asynchronního motoru spočívá ve vytvoření točivého magnetického pole ve vzduchové mezeře mezi státorem a rotorem, které vzniká napájením trojce prostorově rozložených statorových vinutí harmonickým napětím vzájemně fázově posunutým o 120° . Koncový bod fázoru výsledného magnetického toku tak opisuje kružnici. Následná interakce točivého magnetického pole s rotorem indukuje v rotorovém vinutí napětí, následný průchozí proud pak v záběru s magnetickým tokem vyvolá točivý moment. Rychlost otáčení rotoru nemůže být shodná s rychlostí točivého magnetického pole (synchronní rychlost), protože při totožné rychlosti by se neindukovalo napětí v rotorovém vinutí. Z těchto důvodů je motor nazýván asynchronní. Konstrukce rotoru bývá nejčastěji tvořena tzv. s kotvou na krátko. Vinutí je provedeno formou hliníkových nebo měděných tyčí, které jsou umístěny v drážkách rotoru a na konci spojeny kruhem na krátko.

Příklad napájecího napětí statorových vinutí

$$\begin{aligned} u_a &= U_s \sin(\omega t), \\ u_b &= U_s \sin\left(\omega t - \frac{2}{3}\pi\right), \\ u_c &= U_s \sin\left(\omega t - \frac{4}{3}\pi\right), \end{aligned} \quad (2.1)$$

kde U_s je amplituda napájecího napětí, $\omega = 2\pi f$ je úhlová rychlost napájecího napětí (f je frekvence napájecího napětí) a t je čas.

Pro asynchronní motor s počtem pólových dvojic P_p vypadá vztah pro synchronní rychlost

$$\omega_s = \frac{\omega}{P_p} = \frac{2\pi f}{P_p}. \quad (2.2)$$

Rozdíl mezi otáčkami rotoru ω_m (mechanické otáčky) a synchronními otáčkami ω_s charakterizuje skluz

$$s = \frac{\omega_s - \omega_m}{\omega_s}, \quad (2.3)$$

který lze vyjádřit buď poměrově, nebo v procentech.

Mechanickou rychlost rotoru vyjádřená z (2.3) s pomocí (2.2) je

$$\omega_m = \frac{2\pi f}{P_p} (1 - s), \quad (2.4)$$

kde s je skluz.

Z rovnice (2.4) plyne, že rychlost rotoru se dá ovlivnit třemi způsoby. První možností je změna počtu pólových dvojic P_p . Jelikož počet pólových dvojic může být pouze celé číslo, dosáhne se pouze skokové změny otáčivé rychlosti. Další veličina, měnící otáčivou rychlost je skluz. První dvě uvedené možnosti se používají v starších a nenáročných aplikacích. Poslední a zároveň nejpoužívanější metoda využívá změnu frekvence napájecího napětí pomocí frekvenčních měničů [8].

2.1 MATEMATICKÝ MODEL ASYNCHRONNÍHO MOTORU

Matematický model asynchronního motoru vede na složitý systém nelineárních diferenciálních rovnic. Aby se dosáhlo zjednodušení a zároveň uspokojivě přesné prezentace modelu, musí se přijmout následující zjednodušující předpoklady [1, 2, 8]:

- napájení souměrným harmonickým napětím
- vinutí jednotlivých fází jsou symetricky rozložena v drážkách statoru a rotoru
- odpory a indukčnosti jednotlivých fází statoru i rotoru jsou shodné
- magnetizační charakteristika je lineární a rozložení magnetické indukce ve vzduchové mezeře je harmonické
- počet závitů vinutí rotoru a statoru je shodný
- ztráty v železe jsou zanedbatelné

Napěťové rovnice trojfázového motoru pro jednotlivá vinutí statoru $k = a, b, c$ a rotoru $k = A, B, C$ jsou

$$u_k = R_k I_k + \frac{d\psi_k}{dt}, \quad (2.5)$$

kde R_k je odpor pro jednotlivá vinutí statoru, I_k je proud pro jednotlivá vinutí a Ψ_k je magnetický spřažený tok jednotlivých vinutí.

Magnetický spřažený tok lze vyjádřit např. pro fázi a viz Obrázek 2.1

$$\begin{aligned} \psi_a = L_{Sa} I_{Sa} - \frac{M_S}{2} I_b - \frac{M_S}{2} I_c + M_{SR} \cos(\theta) I_A + M_{SR} \cos\left(\theta + \frac{2\pi}{3}\right) I_B \\ - M_{SR} \cos\left(\theta - \frac{2\pi}{3}\right) I_C, \end{aligned} \quad (2.6)$$

kde L_{Sa} je vlastní indukčnost statorového vinutí, M_S je vzájemná indukčnost statorového vinutí mezi fázemi a M_{SR} je vzájemná indukčnost mezi fázemi statorového a rotorového vinutí. Úhel natočení rotoru se vypočítá podle

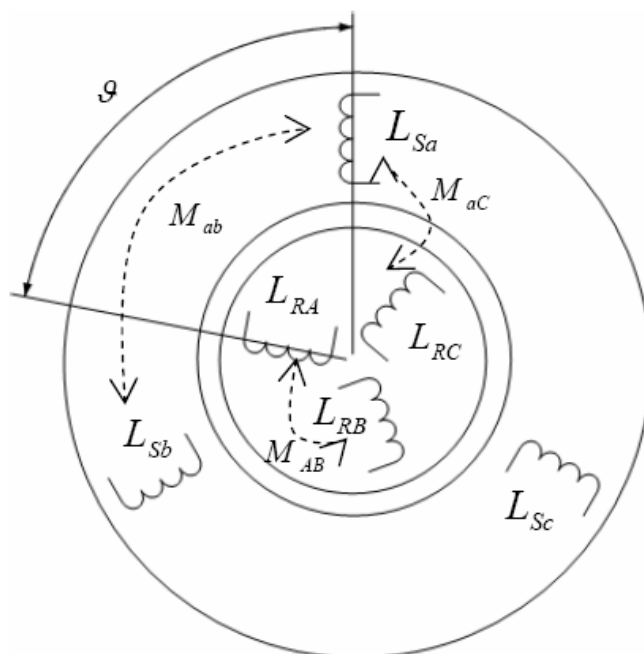
$$\theta = \int \omega_m dt. \quad (2.7)$$

Při bližším pohledu na výše uvedené rovnice zjistíme, že model v trojfázové soustavě by byl značně složitý a obsahoval velký počet rovnic. Detailní odvození matematického modelu pro trojfázovou soustavu je možno nalézt např. v [1, 2]. Pro zjednodušení a snížení počtu rovnic se nám nabízí možnost transformace trojfázové soustavy na ekvivalentní dvoufázovou soustavu pomocí prostorových vektorů (2.8), která matematicky vyjadřuje skutečnost, že prostorový komplexor je fázor, otáčející se úhlovou rychlostí ω . V dalším kroku musíme odstranit závislost změny magnetické vazby statorového a rotorového vinutí při otáčení rotoru. Tuto nežádoucí závislost je možné odstranit s využitím matematického aparátu Clarkovy a Parkovy transformace. Cílem transformace je nahlížet na děje probíhající ve statoru a rotoru ze společného souřadnicového systému, který svírá úhel θ (při ω) s osou statorového vinutí. Vznikne mezi nimi pevná magnetická vazba a děje statoru i rotoru se budou

promítat do ekvivalentního dvoufázového systému. Možnosti volby souřadnicových systémů jsou popsány v následující tabulce.

Úhlová rychlost ω_k	Osy	Vazba souř. systému na
0	α - β	stator
ω_m	d-q	rotor

Tabulka 2.1 Volba souřadnicového systému



Obrázek 2.1: Zjednodušené schéma asynchronního motoru [8]

2.1.1 Prostorový vektor

Změna úhlu mezi jednotlivými fázemi statoru a rotoru má vliv na velikost vzájemné indukčnosti. Použití prostorového vektoru, definovaného v komplexní rovině, umožňuje vypočítat okamžité hodnoty veličin trojfázového systému. Prostorový vektor např. statorového proudu je

$$\bar{i}_s = K (i_{Sa} + \bar{a} \cdot i_{Sb} + \bar{a}^2 \cdot i_{Sc}), \quad (2.8)$$

kde K je konstanta definovaná v (2.12) a \bar{a} je jednotkový vektor definovaný

$$\bar{a} = e^{\frac{j2\pi}{3}} = -\frac{1}{2} + \frac{\sqrt{3}}{2}j. \quad (2.9)$$

Pro jednotkový vektor platí

$$\bar{a}^2 = \bar{a}^{-1} = e^{\frac{j4\pi}{3}} = -\frac{1}{2} - \frac{\sqrt{3}}{2}j. \quad (2.10)$$

Výše uvedené odvození vychází z Eulerových vztahů

$$\begin{aligned} \cos \alpha + j \sin \alpha &= e^{j\alpha}, \\ \cos \alpha &= \frac{1}{2}(e^{j\alpha} + e^{-j\alpha}), \\ \sin \alpha &= \frac{1}{2j}(e^{j\alpha} - e^{-j\alpha}), \end{aligned} \quad (2.11)$$

Konstanta K je volitelná, nejvýhodnější je zvolit hodnotu $\frac{2}{3}$, potom můžeme prostorový vektor \bar{i}_s přepsat do tvaru

$$\bar{i}_s = \frac{2}{3}I_s \frac{3}{2}e^{j\omega_1 t} = I_s e^{j\omega_1 t}, \quad (2.12)$$

kde ω_1 je libovolná úhlová rychlost a platí předpoklad, že amplitudy všech proudů se jsou shodné ($i_a = i_b = i_c = i_m$).

2.1.2 Clarkova transformace

Umožňuje transformaci do stacionárních statorových souřadnic α - β . Na obrázku 2.2 je vidět grafické znázornění přechodu mezi trojfázovým a transformovaným dvoufázovým systémem.

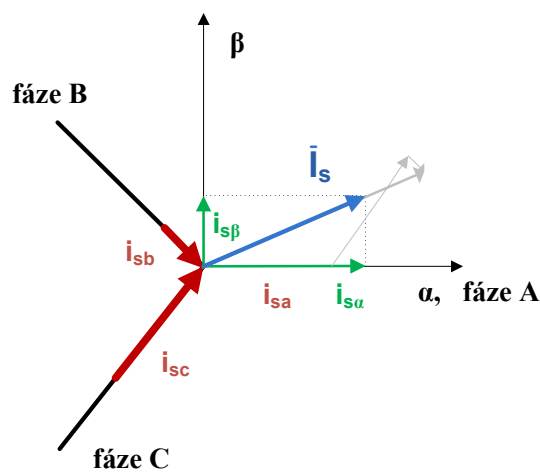
$$\overline{i_{S\alpha\beta}} = i_{S\alpha} + j i_{S\beta} = \frac{2}{3} \left(i_{Sa} - \frac{1}{2} i_{Sb} - \frac{1}{2} i_{Sc} + j \frac{\sqrt{3}}{2} i_{Sc} - j \frac{\sqrt{3}}{2} i_{Sb} \right) \quad (2.13)$$

Pokud platí symetrie $i_{Sa} + i_{Sb} + i_{Sc} = 0$ a $u_{Sa} + u_{Sb} + u_{Sc} = 0$ půjdou vyjádřit složky v α - β souřadnicích a Clarkova transformace bude vypadat

$$\begin{aligned} i_{S\alpha} &= i_{Sa}, \\ i_{S\beta} &= \frac{1}{\sqrt{3}}(i_{Sb} - i_{Sc}). \end{aligned} \quad (2.14)$$

Pro zpětnou Clarkovu transformaci platí

$$\begin{aligned} i_{sa} &= i_{s\alpha}, \\ i_{sb} &= -\frac{1}{2}i_{s\alpha} + i_{s\beta}, \\ i_{sc} &= -\frac{1}{2}i_{s\alpha} - i_{s\beta}. \end{aligned} \quad (2.15)$$



Obrázek 2.2: Clarkova transformace[3]

2.1.3 Parkova transformace

Pro implementaci řídicích algoritmů je nutné transformovat veškeré veličiny do stejného souřadnicového systému. Souřadnicový systém α - β není vhodný pro syntézu řídicích algoritmů, protože transformované hodnoty jsou závislé na čase a rychlosti. Parkova transformace vyjádří veličiny v rotujícím souřadném systému, rotujícím úhlovou rychlostí ω_s (systém d-q). Veličiny se po Parkově transformaci jeví jako stojící (stejnoseměrné). Pro výpočet Parkovy transformace je nutné nejprve převést trojfázový systém na dvoufázový pomocí Clarkovy transformace a až po té je možno vypočítat Parkovu transformaci

$$\begin{aligned} i_{sd} &= i_{s\alpha} \cos \theta + i_{s\beta} \sin \theta, \\ i_{sq} &= -i_{s\alpha} \sin \theta + i_{s\beta} \cos \theta, \end{aligned} \quad (2.16)$$

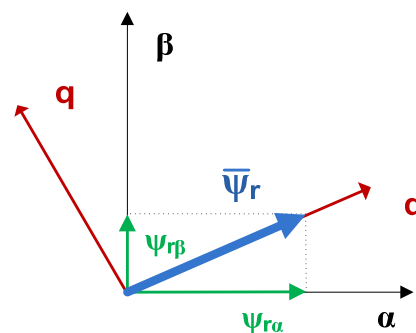
kde θ je úhel natočení rotoru.

Zpětná Parkova transformace

$$\begin{aligned} i_{s\alpha} &= i_{sd} \cos \theta - i_{sq} \sin \theta, \\ i_{s\beta} &= i_{sd} \sin \theta + i_{sq} \cos \theta. \end{aligned} \quad (2.17)$$

Úhel je možné vypočítat i pomocí složek rotorového toku. Výpočet je velice užitečný např. při implementaci na procesoru. Z obrázku 2.3 můžeme odvodit:

$$\begin{aligned} \psi_{Rd} &= \sqrt{\psi_{R\alpha}^2 + \psi_{R\beta}^2} & \sin \theta &= \frac{\psi_{R\beta}}{\psi_{Rd}} \\ \cos \theta &= \frac{\psi_{R\alpha}}{\psi_{Rd}} & \arctan &\left(\frac{\psi_{R\beta}}{\psi_{R\alpha}}\right) \end{aligned} \quad (2.18)$$



Obrázek 2.3: Parkova transformace[3]

2.1.4 Odvození modelu pro obecný souřadnicový systém

Rovnice modelu jsou uvedeny pro obecný souřadnicový systém, který je charakterizován parametrem ω_K (obecná úhlová rychlost). Reálnou osu symbolizuje dolní index u a imaginární osu v. Podrobnější odvození rovnic z trojfázového systému je možno najít např. v [1]. Napěťové rovnice statoru asynchronního motoru v obecném souřadnicovém systému jsou

$$\begin{aligned} u_{su} &= R_s i_{su} + \frac{d\psi_{su}}{dt} - \omega_K \Psi_{sv}, \\ u_{sv} &= R_s i_{sv} + \frac{d\psi_{sv}}{dt} + \omega_K \Psi_{su}, \end{aligned} \quad (2.19)$$

kde R_s je odpor statorového vinutí, Ψ_s je magnetický statorový rok a ω_K je obecná úhlová rychlost (volba souřadnicového systému).

Napět'ové rovnice rotoru v obecném souřadnicovém systému

$$\begin{aligned} 0 = u_{Ru} &= R_R i_{Ru} + \frac{d\psi_{Ru}}{dt} - (\omega_K - \omega)\psi_{Rv}, \\ 0 = u_{Rv} &= R_R i_{Rv} + \frac{d\psi_{Rv}}{dt} + (\omega_K - \omega)\psi_{Ru}, \end{aligned} \quad (2.20)$$

kde R_R je odpor rotorového vinutí, Ψ_R je magnetický rotorový tok a ω je úhlová rychlost. Rotorové napětí je nulové pouze u provedení motoru s kotvou nakrátko.

Pro složky magnetických toků statoru a rotoru platí

$$\begin{aligned} \psi_{Su} &= L_S i_{Su} + L_m i_{Ru}, \\ \psi_{Sv} &= L_S i_{Sv} + L_m i_{Rv}, \\ \psi_{Ru} &= L_R i_{Ru} + L_m i_{Su}, \\ \psi_{Rv} &= L_R i_{Rv} + L_m i_{Sv}, \end{aligned} \quad (2.21)$$

kde L_S a L_R jsou vlastní indukčnosti statoru a rotoru a L_m vzájemná indukčnost při $\theta = 0$.

Při tvorbě matematického modelu asynchronního motoru se vyjádřením veličin z (2.21) a následným dosazením do rovnic (2.19), (2.20) a (2.24) volit vnitřní (stavové) proměnné modelu [2]:

1. proudový model: stavovými proměnnými jsou proudy statoru a rotoru
2. tokový model: stavovými proměnnými jsou magnetické toky statoru a rotoru
3. proudově-tokový model: stavovými proměnnými jsou zde proudy i magnetické toky. (většinou statorový proud a rotorový magnetický tok)

Všechny jmenované modely jsou navzájem ekvivalentní a jsou řešením soustavy pěti (včetně (2.23)) diferenciálních rovnic 1. řádu.

Posledním krokem je vyjádřit elektromagnetickou rovnicí pro výpočet vnitřního momentu motoru

$$T_e = \frac{3}{2} P_p (i_{Sv} \psi_{Ru} - i_{Su} \psi_{Rv}) \quad (2.22)$$

a následně z pohybové rovnice vyjádřit otáčky

$$T_e - T_L = J \frac{d\omega_m}{dt}, \quad (2.23)$$

kde T_L je zátěžový moment, J celkový setrvačný moment rotoru a pro mechanické otáčky platí $\omega_m = \frac{\omega}{P_p}$.

Podle souboru rovnic (2.19) až (2.23) je možné vytvořit model 3f asynchronního motoru ve zvoleném souřadnicovém systému.

2.2 PARAMETRY ASYNCHRONNÍHO MOTORU PRO SIMULACE

Parametry asynchronního motoru použité pro simulace v této práci byly určeny na základě školního laboratorního reálného ASM od firmy Elektrik. V reálných podmínkách se parametry motoru mohou vlivem teploty měnit. Tabulka 2.3 zachycuje změnu parametrů v rozsahu $\mp 30\%$, které byly použity k testování robustnosti algoritmů.

Odpor statoru:	$R_S = 31 \text{ Ohm}$
Odpor rotoru:	$R_R = 28 \text{ Ohm}$
Indukčnost statoru:	$L_S = 1,537 \text{ H}$
Indukčnost rotoru:	$L_R = 1,537 \text{ H}$
Vzájemná indukčnost:	$L_m = 1,407 \text{ H}$
Moment setrvačnosti:	$J = 0.002 \text{ kg} \cdot \text{m}^2$
Počet pólových dvojic:	$P_p = 2$
Max. statorové napětí:	$U_{S \max} = 300 \text{ V}$
Max. statorový proud:	$I_{S \max} = 4 \text{ A}$

Tabulka 2.2: Parametry asynchronního motoru

Odpor statoru:	$R_S = 41 \text{ Ohm}$
Odpor rotoru:	$R_R = 22 \text{ Ohm}$
Indukčnost statoru:	$L_S = 1,567 \text{ H}$
Indukčnost rotoru:	$L_R = 1,500 \text{ H}$
Hlavní indukčnost:	$L_m = 1,407 \text{ H}$

Tabulka 2.3: Změna parametrů ASM

3. SYNCHRONNÍ MOTOR S PERMANENTNÍM MAGNETEM (PMSM)

Konstrukce statoru PMSM je jako u asynchronního motoru v provedení trojfázového vinutí uloženého v drážkách magnetického obvodu. Uvnitř statoru se nachází rotor, který na sobě nese permanentní magnety místo vinutí. Magnety se z počátku vyráběly z oceli, avšak v dnešní době se vyrábějí zejména ze speciálních slitin jako je nikl, slitiny kobaltu, ferity stroncia a bária atd. a vzácných zemin např. samarium-kobalt (SmCO_5 , SmCO_{17}) nebo neodým-bor-železo (NdBFe). Tyto nové materiály díky své vysoké magnetické indukčnosti umožňují zmenšit konstrukci motoru se zachováním výkonu. Při napájení statorového vinutí trojfázovým harmonickým napětím vzniká při průchodu proudu statorovým vinutím točivé magnetické pole, které je schopno roztočit rotor osazený permanentními magnety na rychlost shodnou se synchronní rychlostí točivého pole. V tom spočívá rozdíl mezi synchronním a asynchronním motorem. Synchronní motor se otáčí rychlostí ω .

3.1 MATEMATICKÝ MODEL PMSM

Odvození matematického modelu PMSM trojfázového do souřadnicového systému α - β , d-q probíhá pomocí Clarkovy a Parkovy transformace uvedené u asynchronního motoru v kapitolách 2.1.2 a 2.1.3. Napájení motoru je trojfázové harmonické viz (2.1). Pro matematický model PMSM platí následující zjednodušující předpoklady:

- napájení souměrným harmonickým napětím
- vinutí jednotlivých fází jsou symetricky rozložena v drážkách statoru
- odpory a indukčnosti jednotlivých fází statoru i rotoru jsou shodné
- magnetizační charakteristika je lineární a rozložení magnetické indukce ve vzduchové mezeře je harmonické
- ztráty v železe jsou zanedbatelné

- nulový vodič není připojen
- Tlumící vinutí na rotoru není provedeno a rovněž se zanedbávají tlumící účinky rotoru

3.1.1 Matematický model PMSM v d-q a α - β souřadnicích

Matematický model je výhodné vytvořit v souřadnicovém systému d-q spojeným s rotorem motoru, protože se odstraní závislosti na úhlu natočení statoru na poloze rotoru jako v případě modelu v souřadnicovém systému α - β . Pro napěťové rovnice statoru v systému d-q platí

$$\begin{aligned} u_{sd} &= R_S i_{sd} + \frac{d\psi_{rd}}{dt} - \omega \Psi_{rq}, \\ u_{sq} &= R_S i_{sq} + \frac{d\psi_{rq}}{dt} + \omega \Psi_{rd}, \end{aligned} \quad (3.1)$$

kde R_S je odpor statoru, Ψ_{rd} , Ψ_{rq} jsou magnetické toky a ω je elektrická rychlost $\omega = \omega_m P_p$.

Rovnice magnetických toků vypadají následovně

$$\begin{aligned} \psi_{rd} &= L_d i_{sd} + \psi_m, \\ \psi_{rq} &= L_q i_{sq}, \end{aligned} \quad (3.2)$$

kde L_d a L_q jsou indukčnosti v podélné a příčné ose a Ψ_m je vzájemný spřažený magnetický tok mezi státorem a rotorem.

Pro tvorbu modelu se dosadí (3.2) do (3.1)

$$\begin{aligned} u_{sd} &= R_S i_{sd} + L_d \frac{di_{sd}}{dt} - \omega L_q i_{sq}, \\ u_{sq} &= R_S i_{sq} + L_q \frac{di_{sq}}{dt} + \omega L_d i_{sd} + \omega \psi_m. \end{aligned} \quad (3.3)$$

Moment PMSM

$$T_e = \frac{3}{2} P_p (i_{sq} \psi_{rd} - i_{sd} \psi_{rq}) = \frac{3}{2} P_p [\psi_m + (L_d - L_q) i_{sd}] i_{sq}. \quad (3.4)$$

Pokud $L_d = L_q$ rovnice 3.4 se zjednoduší

$$T_e = \frac{3}{2} P_p \psi_m i_{sq}. \quad (3.5)$$

Tato rovnice definuje moment PMSM jako součin kolmé (momentové) složky prostorového vektoru statorového proudu a konstantního budícího magnetického toku, který je dán permanentními magnety.

Zbývá už jen pohybová rovnice:

Rovnice (3.1) až (4.5) tvoří model PMSM v souřadnicích d-q. Pro úplnost zbývá uvést rovnice v α - β souřadnicích, které však jsou závislé na úhlu natočení rotoru vůči statoru.

Pro magnetické toky v α - β platí

$$\begin{aligned} \psi_\alpha &= L_S i_{s\alpha} + \psi_m \cos \theta, \\ \psi_\beta &= L_S i_{s\beta} + \psi_m \sin \theta. \end{aligned} \quad (3.6)$$

Dosazením (3.6) do (3.1) se získají napěťové rovnice PMSM v α - β souřadnicích

$$\begin{aligned} u_{s\alpha} &= R_S i_{s\alpha} + L_S \frac{di_{s\alpha}}{dt} - \omega_e \psi_m \sin \theta, \\ u_{s\beta} &= R_S i_{s\beta} + \frac{di_{s\beta}}{dt} + \omega_e \psi_m \cos \theta. \end{aligned} \quad (3.7)$$

Pohybová rovnice (2.23) je stejná, proto zbývá jen uvést momentovou rovnici a je možné vytvořit model PMSM v souřadnicích α - β

$$T_e = \frac{3}{2} P_p (i_{s\alpha} \psi_{r\beta} - i_{s\beta} \psi_{r\alpha}). \quad (3.8)$$

3.2 PARAMETRY SYNCHRONNÍHO MOTORU PRO SIMULACE

Parametry motoru pro veškeré simulace řídicích algoritmů PMSM použité v této práci byly odvozeny od reálného motoru TGT2 – 0032 – 30 – 24 / TOPS1KX-1M firmy TG drives. Tabulka 3.2 zachycuje změnu parametrů v rozsahu $\pm 30\%$, které byly použity k testování robustnosti algoritmů.

Odpor statoru:	$R_S = 0,2915 \text{ Ohm}$
Indukčnost v podélné ose:	$L_d = 0,235 \text{ mH}$
Indukčnost v příčné ose:	$L_q = 0,235 \text{ mH}$
Magnetický tok magnetu:	$\psi_m = 0,01105 \text{ Wb}$
Moment setrvačnosti:	$J = 0,000008 \text{ kg} \cdot \text{m}^2$
Počet pólových dvojic:	$P_p = 3$
Maximální statorové napětí:	$U_{S \max} = 36,3 \text{ V}$
Maximální statorový proud:	$I_{S \max} = 8 \text{ A}$

Tabulka 3.1: Parametry PMSM

Odpor statoru:	$R_S = 0,36 \text{ Ohm}$
Indukčnost v podélné ose:	$L_d = 0,258 \text{ mH}$
Indukčnost v příčné ose:	$L_q = 0,258 \text{ mH}$
Magnetický tok magnetu:	$\psi_m = 0,0144 \text{ Wb}$

Tabulka 3.2: Změna parametrů PMSM

4. VEKTOROVÉ ŘÍZENÍ

4.1 VEKTOROVÉ ŘÍZENÍ ASYNCHRONNÍHO MOTORU

Vektorové řízení patří mezi nejpobulárnější řídicí algoritmy asynchronního. Jelikož algoritmus umožňuje nezávislé řízení magnetického toku a momentu, podobá se takto řízený motor svými vlastnostmi stejnosměrnému motoru. Algoritmus nepracuje s reálnými proměnnými stroje, ale transformovanými do souřadnicového systému daného polohou vektoru magnetického toku. Souřadnicový systém může být obecně orientován na magnetický hlavní, statorový nebo rotorový tok. Avšak pouze při orientaci na rotorový tok nezůstává vzájemná vazba mezi magnetickým tokem a momentem [1, 2]. Pro realizaci algoritmu vektorového řízení je získat informace o poloze a velikosti magnetického rotorového toku. Tyto informace se dají získat buď měřením, nebo použitím modelu magnetického toku (viz kapitola 4.1.1).

Při orientaci na rotorový tok je nutná transformace do d-q souřadnic, kde složka proudu i_{sd} ležící na reálné ose nazývá tokotvorná (4.1) a složka i_{sq} ležící na imaginární ose se nazývá momentotvorná (4.2).

Pro odvození následujících rovnic z matematického modelu v d-q souřadnicích musí platit, že složka Ψ_{Rq} a její derivace je nulová. Detailní odvození např. v [1, 2].

$$|\psi_R| = \psi_{Rd} = L_m i_{sd} \quad (4.1)$$

$$T_e = \frac{3}{2} P_p \frac{L_m}{L_R} \psi_{Rd} i_{sq} \quad (4.2)$$

Z rovnic (4.1) a (4.2) plyne, že při udržení konstantního rotorového toku Ψ_R dojde skutečně k oddělení řízení velikosti toku a momentu. Existují různé varianty a modifikace vektorového řízení. Na obrázku 4.1 je zobrazeno blokové schéma vektorového řízení realizovaného v této práci. Schéma obsahuje bloky transformace souřadnic pomocí Clarkovy, Carkovy transformace a jejich inverzní podoby. Model pro výpočet vektoru velikosti magnetického toku a polohy (proudový model viz kapitola 4.1.1). Dále blok obsahuje výhodnější výpočet úhlu natočení vektoru

magnetického toku pro Parkovu transformaci (kapitola 2.1.3). Čtveřici regulátorů, regulátor otáček (PID), který nastavuje referenční hodnotu proudu i_{sq} pro podřazený regulátor proudu. Tento regulátor (PI) na základě odchylky od referenční hodnoty proudu i_{sq} a její naměřené hodnoty vytváří složku napětí u_{sq} . Ve zbývající větvi je nasazen regulátor magnetického toku (PID), který vytváří referenční složku proudu i_{sd} pro další podřazený regulátor proudu. Tento regulátor (PI) opět na základě odchylky od referenční hodnoty proudu i_{sd} a její naměřené hodnoty vytváří složku napětí u_{sd} . Velmi důležitým blokem je „Decoupling“ (znamená „zrušení vazby“). Tento blok podle teorie zpětnovazební linearizace připočítává složky napětí křížených vazeb (viz kapitola 4.1.2). Výsledné napětí je přepočítáno pomocí Clarkovy a Parkovy transformace na trojfázové napětí, které je přivedeno na fáze statoru motoru. Simulace neobsahuje model PWM ani frekvenčního měniče.

4.1.1 Výpočet magnetického toku

1) Napěťový model

Efektivní využití je zejména v oblasti vysokých otáček, kde se může zanedbat vliv chyby statorového odporu při nízkém statorovém napětí a vysokých statorových proudech. Tento model má však jednu podstatnou negativní vlastnost, používá prostou integraci, což může způsobovat k nežádoucímu driftu integrátoru. Mezi výhody použití napěťového modelu patří nezávislost na měření otáček a odporu rotoru [5].

$$\psi_S = \int_0^t (u_s - R_S i_S) dt$$

$$\psi_R = \frac{L_R}{L_m} \int_0^t (u_s - R_S i_S) dt + \frac{L_m^2 - L_R L_S}{L_m} i_S$$
(4.3)

2) Proudový model

Tento model řeší problémy s driftem integrátoru, protože nepoužívá čistou integraci. Další nespornou výhodou je nezávislost na statorovém odporu, avšak při velkém skluzu je výpočet ovlivněn chybou rotorového odporu. Nevýhodou oproti napět'ovému modelu je nutná znalost mechanických otáček rotoru [5].

$$\frac{d\psi_R}{dt} = \frac{R_R L_m}{L_r} i_S - \psi_R \left(\frac{R_R}{L_R} - jPp\omega_m \right) \quad (4.4)$$

$$\psi_S = L_S i_S - \frac{L_m}{L_R} (\psi_R - L_m i_S)$$

3) Kombinovaný model

Tento model je kombinací napět'ového a proudového modelu. Výsledkem je odstranění vlivu driftu integrátoru a snížena citlivost na chyby parametrů, jelikož při pohledu na rovnici (4.5) je vidět výskyt citlivých parametrů jak v čitateli, tak ve jmenovateli výrazů. Tím dochází k „částečnému vykompenzování“ změny parametrů. Tento model je hojně používán v průmyslových aplikacích [5].

$$\frac{d\psi_R}{dt} = \frac{R_R L_m L_R}{L_m^2 R_R + L_R^2 R_S} u_S - \frac{L_R^2 R_S}{L_m^2 R_R + L_R^2 R_S} \left(\frac{R_R}{L_R} - j\omega_e \right) \psi_R \quad (4.5)$$

$$- \frac{R_R L_m (L_m^2 - L_R L_S)}{L_m^2 R_R + L_R^2 R_S} \frac{di_S}{dt}$$

4.1.2 „Decoupling“ pro ASM

Aby bylo dosaženo odděleného řízení magnetického toku, momentu je nutné oddělit a nezávisle řídit složky napětí u_{sd} a u_{sq} , která jsou tvořena proudy i_{sd} a i_{sq} . Napět'ové rovnice (2.19) a (2.20) obsahují obě složky proudů (křížené vazby). Blok „Decoupling“ umožňuje pomocí teorie zpětnovazební linearizace odstranit tyto křížené vazby resp. rozdělit jednotlivé rovnice na lineární a doplňující („křížné“) části. Tato úprava transformuje nelineární rovnice na lineární a tím umožníme řízení

pomocí PI, PID regulátorů místo složitých nelineárních regulátorů. Výstup z jednotlivých regulátorů napětí (PI) jsou tedy složky u_{sd}^{lin} a u_{sq}^{lin} . K těmto složkám se nakonec přičtou napětí u_{sd}^{dopln} a u_{sq}^{dopln} . Napěťové rovnice vypadají po úpravě následovně

$$\begin{aligned} u_{sd} &= u_{sd}^{lin} + u_{sd}^{dopln} = \left[K_R i_{sd} + K_L \frac{di_{sd}}{dt} \right] - \left[\omega_s K_L i_{sq} + \frac{\psi_{Rd} L_m}{L_R T_R} \right], \\ u_{sq} &= u_{sq}^{lin} + u_{sq}^{dopln} = \left[K_R i_{sq} + K_L \frac{di_{sq}}{dt} \right] + \left[\omega_s K_L i_{sd} + \frac{L_m \omega \psi_{Rd}}{L_R} \right], \end{aligned} \quad (4.6)$$

kde $K_R = R_S + \frac{L_m^2}{L_R^2} R_R$, $K_L = L_S - \frac{L_m^2}{L_r}$.

4.1.3 Návrh regulátorů

Výpočet regulátorů pro vektorové řízení vyplývá z principu rozvětvených regulačních obvodů s pomocnou veličinou (kaskádní regulace), kdy je systém rozdělen na vnější a vnitřní regulační smyčky. Žádané hodnoty pro vnitřní smyčky jsou vždy zadávané vnější, nadřazenou regulační smyčkou. Pro zjednodušení předpokládáme přenosy všech snímačů rovny jedné. Přenosová funkce v Laplaceově transformaci pro navrhnutí regulátoru vnitřní proudové smyčky (lineární část (4.6)) je

$$F_i(s) = \frac{1}{K_R + K_L s} = \frac{K_i}{T_1 s + 1}, \quad (4.7)$$

kde $T_1 = \frac{K_L}{K_R}$ a $K_i = \frac{1}{K_R}$. Přenosová funkce odpovídá setrvačnému članku 1. řádu. Pro řízení proudové smyčky je použit PI regulátor

$$R_i(s) = K_{PI} \frac{(T_1 s + 1)}{s}. \quad (4.8)$$

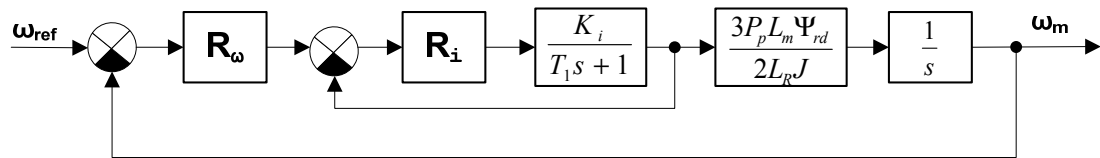
Přenos otevřené smyčky

$$F_0(s) = F_i(s) R_i(s) = \frac{K}{s}, \quad (4.9)$$

kde $K = K_{PI} K_i$. Přenos řízení bude vypadat

$$F_W(s) = F_i(s) R_i(s) = \frac{K}{s + K} = \frac{1}{\frac{1}{K} s + 1}. \quad (4.10)$$

Přenos řízení odpovídá přenosu setrvačného članku, jehož dynamiku lze nastavit zesílením PI regulátoru. Pro návrh regulátoru otáček se vyhází z obrázku 4.1 a rovnice (4.2).



Obrázek 4.1: Kaskádní zapojení otáčkového regulátoru asynchronního motoru

Soustava pro odvození otáčkového obsahuje podřízenou proudovou smyčku a otáčkovou rovnici

$$F_{\omega}(s) = \frac{1}{\frac{1}{K}s + 1} \frac{3P_p L \psi_{Rd}}{2L_R J} \frac{1}{s} = \frac{K_{\omega}}{s(\frac{1}{K}s + 1)} \quad (4.11)$$

kde $K_{\omega} = \frac{3P_p L \psi_{Rd}}{2L_R J}$ a ψ_{Rd} je konstanta podle (4.1). Jako otáčkový regulátor se může použít typ PI nebo PID.

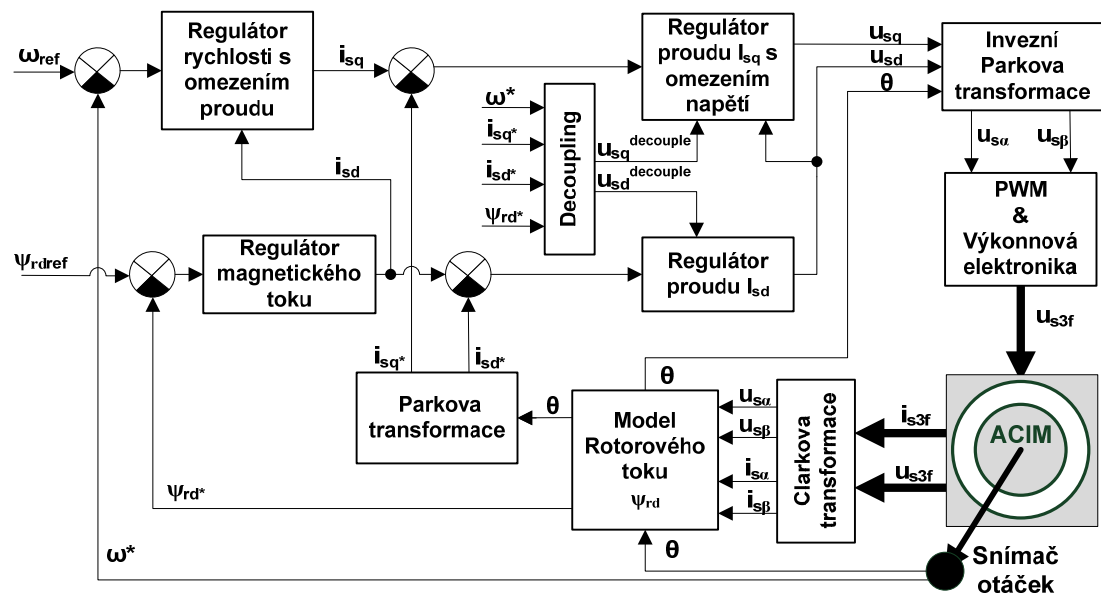
4.1.4 Napěťové, proudové omezení a „anti-windup“

V každém reálném systému existuje mez, nad kterou nemůže působení regulátoru na soustavu růst. Při řízení motorů se musí omezit maximální hodnotu napětí a proudů. Pokud se vloží za výstup regulátoru nelinearitu typu nasycení, omezí se sice hodnota výstupu regulátoru, avšak pokud je používán regulátor obsahující integrační složku, popřípadě sumační složku v podobě diskretních regulátorů, hodnota na těchto částech bude v případě nasycení dál narůstat. Tento fakt se velice nepříznivě projeví na kvalitě přechodového děje, protože při změně znaménka žádané hodnoty se musí hodnota na integrátoru nejprve „odintegrovat“. Řešení se nabízí v použití „anti-windup“ algoritmu (více v [4]). Algoritmus navržený v této práci řeší danou problematiku i s ohledem na to, že nepracujeme se skutečnou hodnotou napětí a proudu, nýbrž s jejich vektorovými složkami v d-q souřadnicích. Každý ze čtveřice regulátorů (Obrázek 4.1) pracuje s jednotlivou složkou napětí (PI) a proudu (PID). Regulátory magnetického toku a složky u_{sd} používají standardní „anti-windup“ algoritmus. Pokud je výstup z regulátoru vyšší než za nasycením, je tento rozdíl odčítán přes časovou konstantu od integrátoru. Regulátory otáček a u_{sq}

složky používají algoritmus stejného principu, avšak s modifikací vzhledem výskytu vektorových složek napětí a proudu. Abychom zjistili skutečnou hodnotu proudu a napětí, musíme vypočítat absolutní hodnotu vektorových složek např. pro napětí:

$$U_{max} = \sqrt{u_{sd}^2 + u_{sq}^2}. \text{ Z toho důvodu obsahuje regulátor otáček a regulátor } u_{sq}$$

složky místo standardního nasycení modifikovanou variantu, která ze znalosti maximálního dovoleného proudu, napětí a znalosti upřednostněné vektorové složky i_{sd} a u_{sd} vypočítá maximální dovolenou hodnotu zbylých složek i_{sq} a u_{sq} . K regulovanému napětí jsou přičteny i křížné složky pomocí bloku „Decoupling“. Tato modifikace umožňuje efektivně využít napěťové a proudové omezení společně s „anti-windup“ algoritmem ve vektorovém řízení.

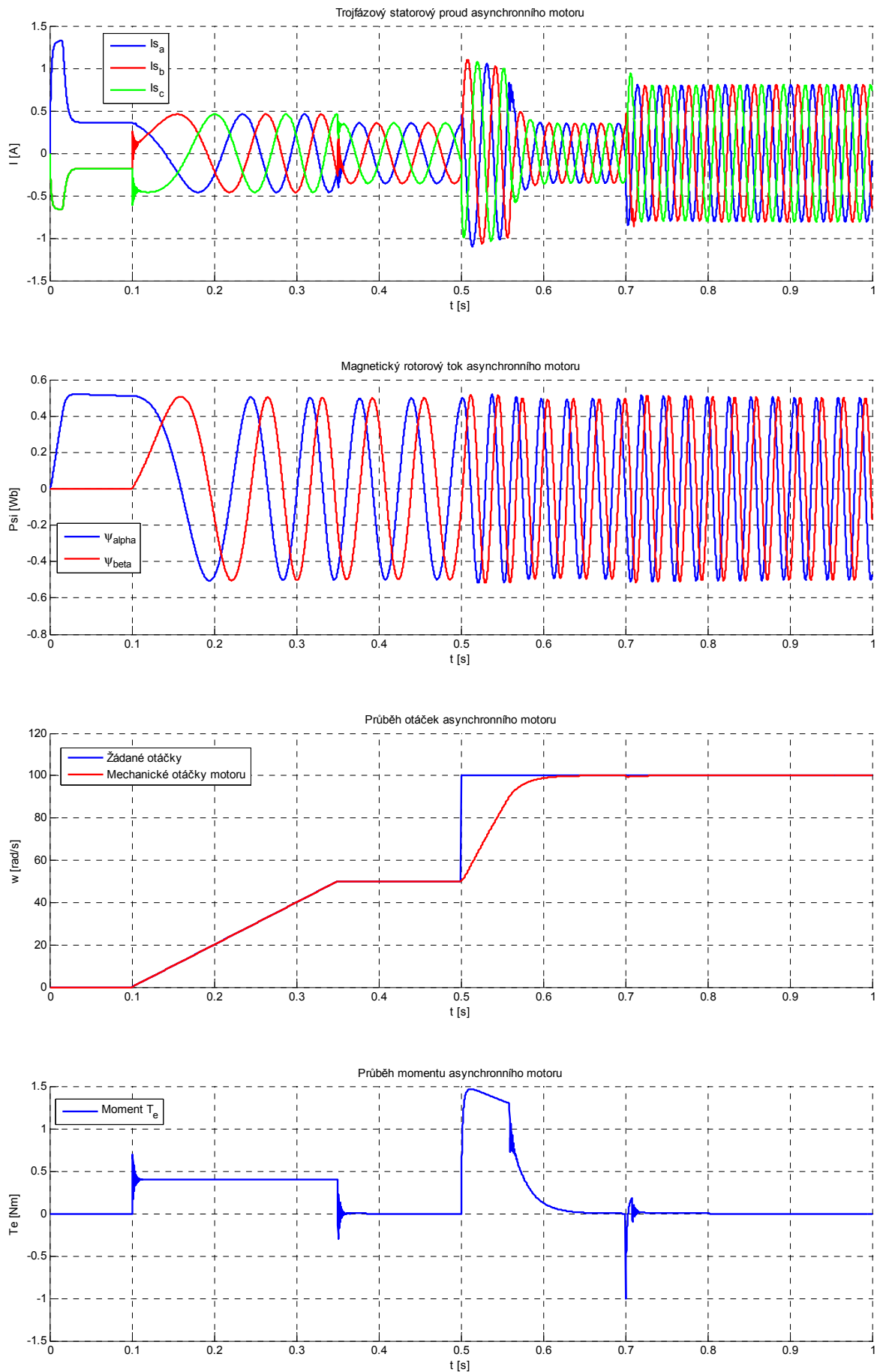


Obrázek 4.2: Blokové schéma vektorového řízení asynchronního motoru

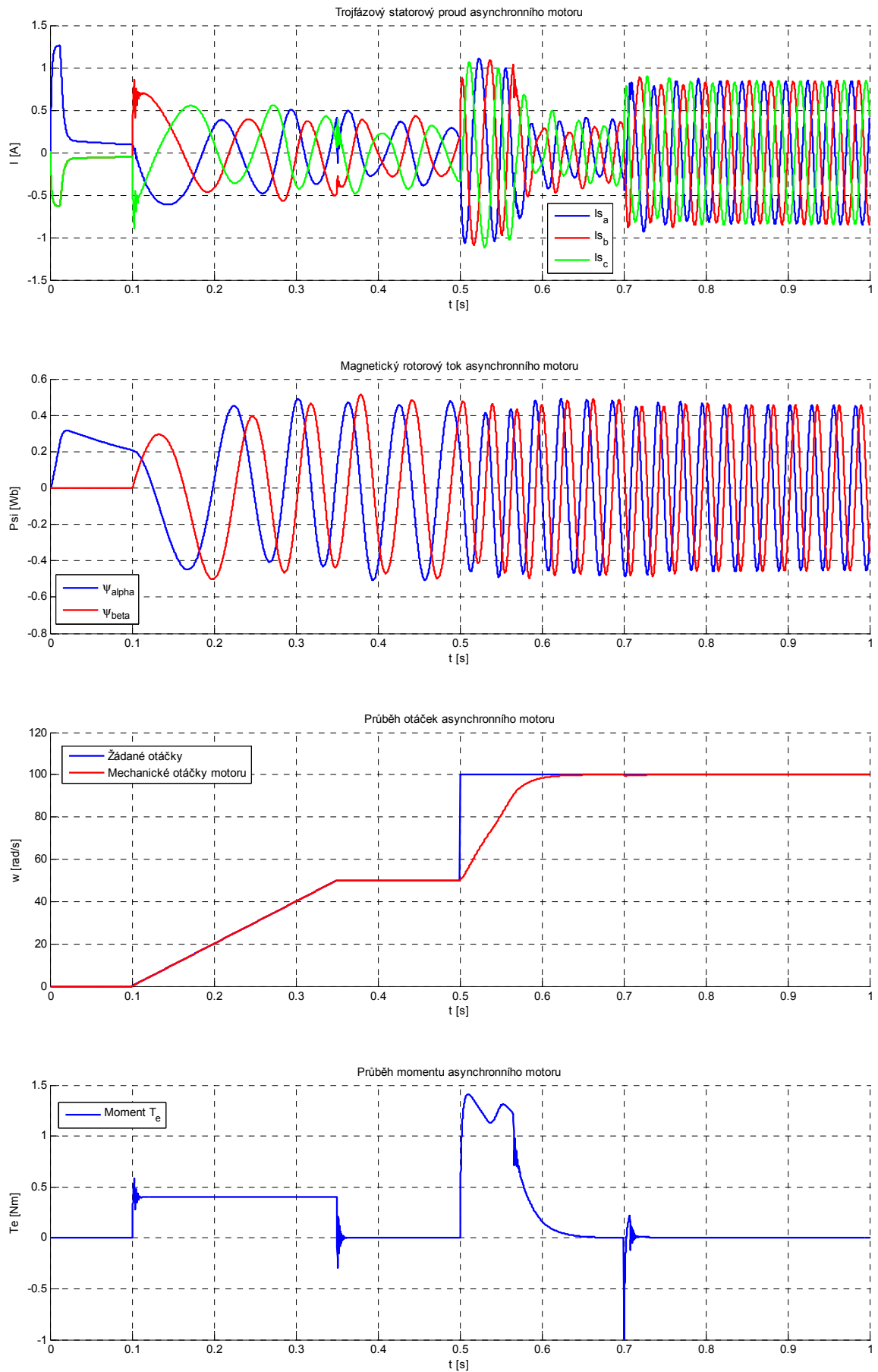
4.1.5 Simulace vektorového řízení asynchronního motoru

Obrázek 4.3 poskytuje náhled na průběh jednotlivých veličin během simulace. Nejprve byl nastaven magnetický tok na hodnotu 0,5 Wb a otáčky byly nastaveny na nulu až do času 0,1 s, kdy začal rozběh motoru po rampě na 50 rad/s. V čase 0,5 s následovala skoková změna otáček o 50 rad/s. Smysl vektorového řízení spočívá v odděleném řízení momentu a magnetického toku. Což potvrzuje průběh grafů v rozmezí 0 až 0,1 s. Budící složka magnetického toku má požadovanou hodnotu a způsobí nabuzení ASM. Od času 0,1 s je vidět vznik točivého magnetického pole a roztočení motoru na požadované otáčky. Poslední graf zobrazuje momentovou charakteristiku motoru. V čase 0,7 s byl motor zatížen (1 Nm), řídicí algoritmus okamžitě vyrovnává změnu s nepatrným zákmitem v otáčkách. Zatížení motoru má také za následek zvýšení statorového proudu.

Obrázek 4.4 zobrazuje průběh simulace při změně parametrů ASM podle tabulky 2.3. Simulace má stejný průběh jako v předchozím případě. Řízení otáček je prakticky beze změny. Patrný je však zdeformovaný průběh statorových proudů a nepatrný rozkmit magnetického toku. Algoritmus se také vyrovnal se zatížením motoru.



Obrázek 4.3: Simulace vektorového řízení ASM



Obrázek 4.4: Simulace vektorového řízení ASM – změna parametrů motoru

4.2 VEKTOROVÉ ŘÍZENÍ PMSM

Regulační struktura vektorového řízení pro PMSM je jednodušší než u asynchronního motoru, protože velikost magnetického toku, podílejícího se na vytvoření momentu, je dána magnety a není nutné zjišťovat velikost a polohu vektoru magnetického toku. Pro vektorové řízení PMSM se stačí orientovat pouze na polohu rotoru motoru, která se získá z jeho integrace rychlosti. Jak vyplývá z rovnice (3.5) moment PMSM je dán součinem složky prostorového vektoru i_q a konstantního budícího magnetického toku. Magnetický tok je dán pouze permanentními magnety. Z tohoto důvodu se reguluje žádaná složka budícího proudu I_d na nulu. Zapojení obsahuje trojici regulátorů, které fungují stejně jako u asynchronního motoru. Algoritmus „anti-windup“ je realizován na stejném principu, jako je popsán v kapitole 4.1.4. Jediným rozdílem je blok „Decoupling“, který se musí upravit vzhledem k rovnicím popisující matematický model PMSM. Blokové schéma znázorněné na obrázku 4.6 obsahuje také bloky realizující Clarkovu, Parkovu transformaci a její inverzní podobu.

4.2.1 „Decoupling“ pro PMSM

Blok má stejnou funkci jako u asynchronního motoru a tou je oddělit křížové vazby z napěťových rovnic PMSM. „Decoupling“ se vyjádří z rovnice (3.3) následnou úpravou

$$u_{sd} = u_{sd}^{lin} + u_{sd}^{dopl'n} = \left[R_s i_{sd} + L_d \frac{di_{sd}}{dt} \right] - [\omega L_q i_{sq}], \quad (4.12)$$

$$u_{sq} = u_{sq}^{lin} + u_{sq}^{dopl'n} = \left[R_s i_{sq} + L_q \frac{di_{sq}}{dt} \right] + [\omega(L_d i_{sd} + \psi_m)].$$

4.2.2 Návrh regulátorů

Návrh regulátorů pro PMSM vychází ze stejných principů popsaných v kapitole 4.1.3. Vnitřní proudové smyčky se odvodí z lineárních částí napěťových rovnic PMSM (4.12). Pro zajištění maximálního momentu PMSM se řídí budící složka proudu i_{sd} na nulovou hodnotu, protože rotor je tvořen permanentním

magnetem (konstantní magnetický tok) a není třeba jej budít. Soustava pro regulátor proudu i_{sd} je

$$F_{id}(s) = \frac{1}{R_s + L_d s} = \frac{K_i}{T_1 s + 1}, \quad (4.13)$$

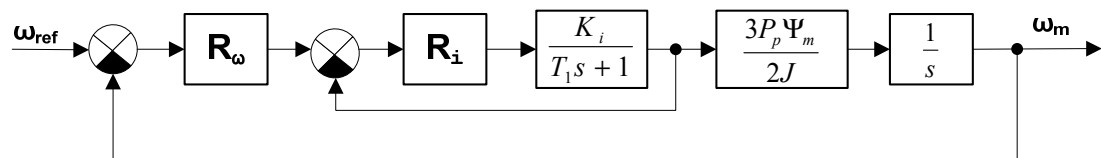
kde $T_1 = \frac{L_d}{R_s}$ a $K_i = \frac{1}{R_s}$. Zesílení regulátoru PI opět umožní nastavení dynamiky proudové smyčky jako v případě asynchronního motoru. Regulátor bude mít přenos

$$R_{id}(s) = K_{PI} \frac{(T_1 s + 1)}{s}. \quad (4.14)$$

Přenos řízení bude

$$F_W(s) = F_{id}(s)R_{id}(s) = \frac{K}{s + K} = \frac{1}{\frac{1}{K}s + 1}, \quad (4.15)$$

kde $K = K_{PI}K_i$. Regulátor proudu i_{sq} bude odvozen stejným způsobem jako R_{id} . Lišit se bude akorát časová konstanta T_1 (pokud $L_d \neq L_q$). Otáčkový regulátor se odvodí z obrázku 4.5.

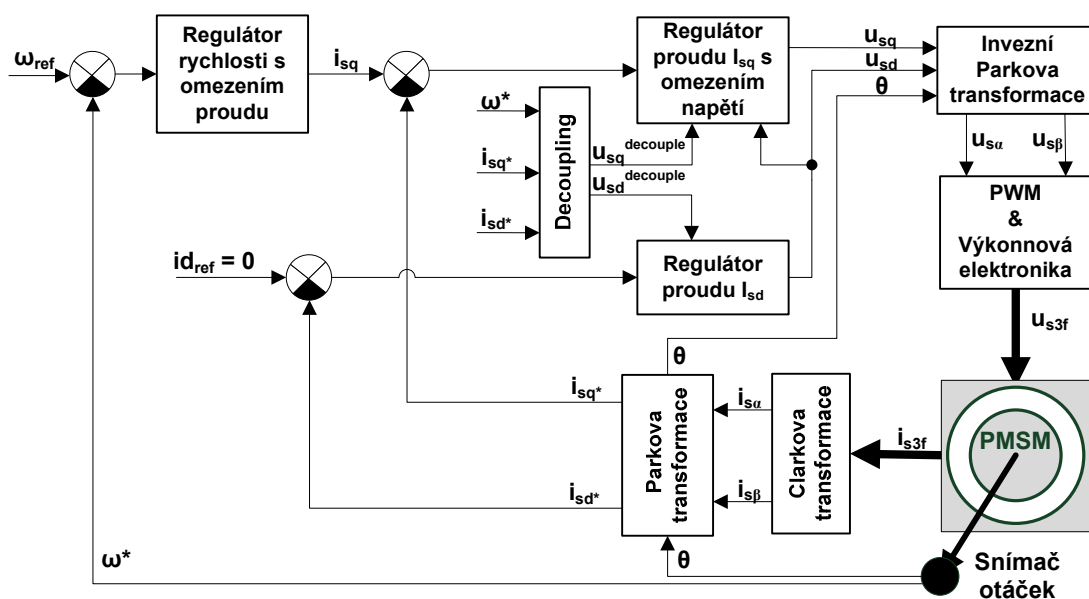


Obrázek 4.5: Kaskádní zapojení otáčkového regulátoru PMSM

Soustava pro otáčkový regulátor je

$$F_\omega(s) = \frac{1}{\frac{1}{K}s + 1} \frac{3P_p \psi_m}{2J} \frac{1}{s} = \frac{K_\omega}{s(\frac{1}{K}s + 1)}. \quad (4.16)$$

kde $K_\omega = \frac{3P_p \psi_m}{2J}$. Jako otáčkový regulátor se obvykle používá typ PI nebo PID.

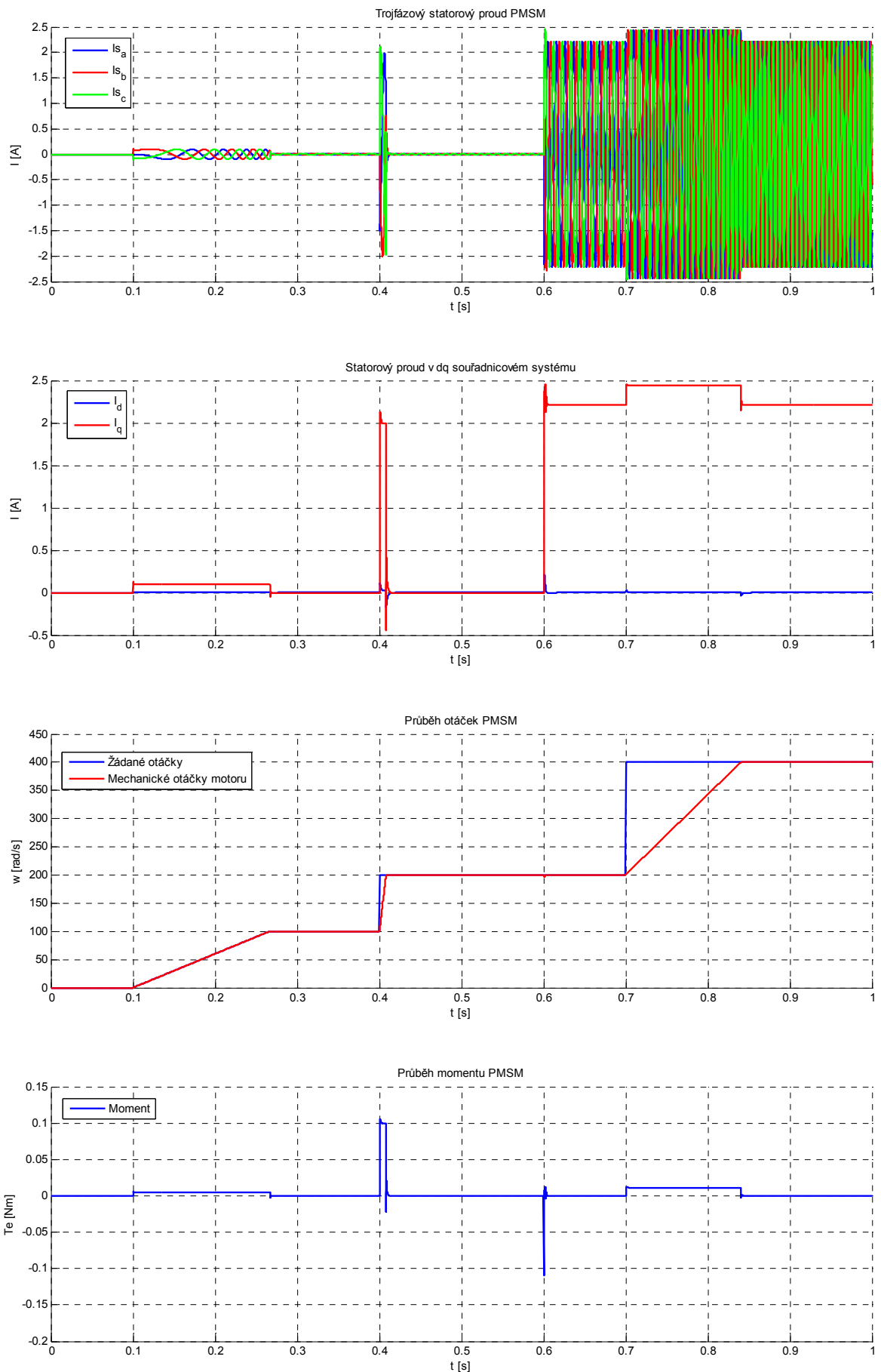


Obrázek 4.6: Blokové schéma vektorového řízení PMSM

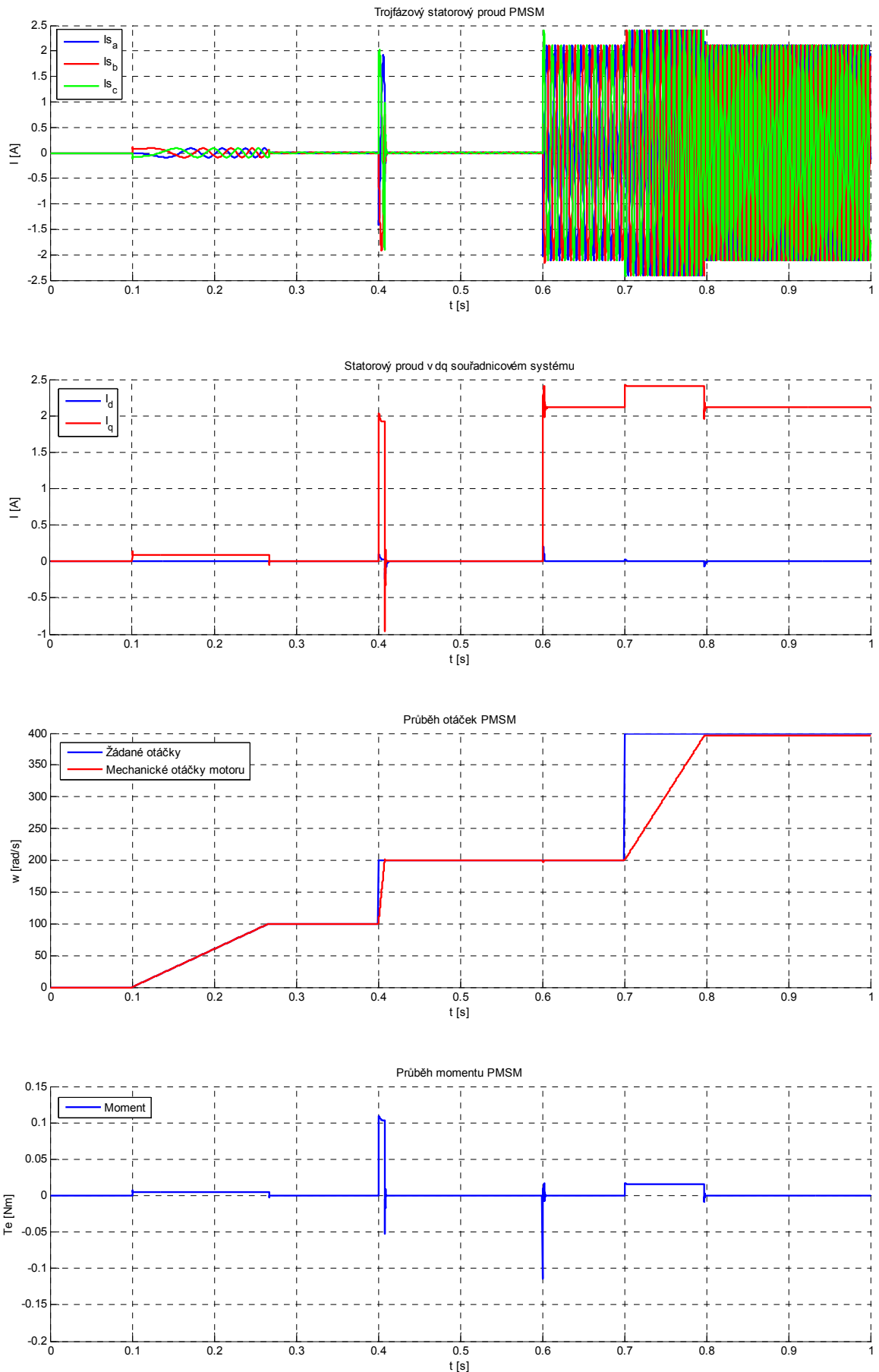
4.2.3 Simulace vektorového řízení PMSM motoru

Simulace vektorového řízení PMSM na obrázku 4.7 a 4.8 započala v čase 0.1 s rozběhem PMSM po rampě na 100 rad/s. Algoritmus reaguje na změnu žádané (0.4 s) o 100 rad/s s výbornou dynamikou. Následuje zatížení motoru (0,12 Nm), které je okamžitě vyrovnáno. Od 0,7 s je skoková změna o 200 rad/s. Otáčky rostou mnohem pomaleji než při předchozím skoku. Je to způsobeno proudovým omezením, které bylo nastaveno pro demonstraci na hodnotu 2,5 A. Jak je vidět proud se nad tuto hodnotu nedostane. PMSM pracuje s maximálním momentem, díky řízení proudu i_d na nulovou hodnotu. Průběh statorových proudů v d-q souřadnicích je na druhém grafu.

Obrázek 4.8 demonstruje průběh řízení PMSM při změně parametrů synchronního motoru podle tabulky 3.2. Změna parametrů se na průběhu regulace neprojevila.



Obrázek 4.7: Simulace vektorového řízení PMSM

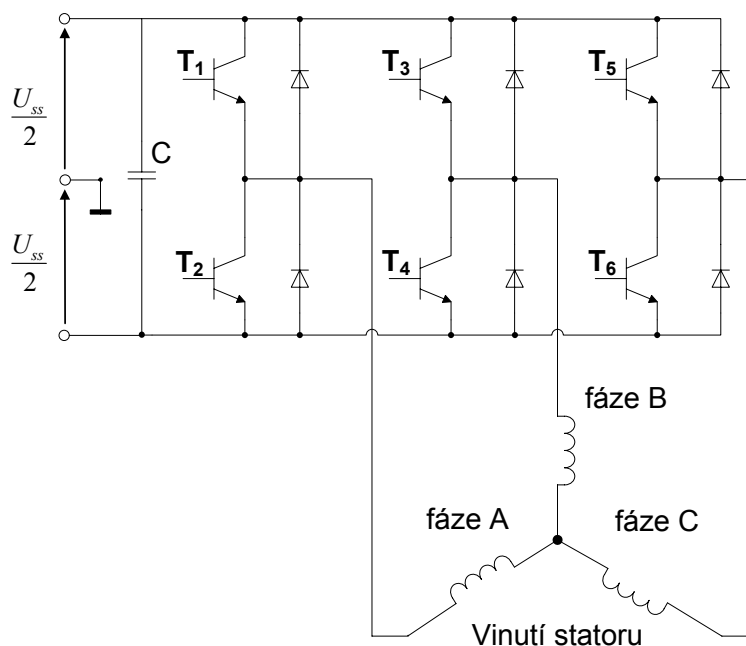


Obrázek 4.8: Simulace vektorového řízení PMSM - změna parametrů PMSM

5. PŘÍMÉ ŘÍZENÍ MOMENTU (DTC)

Použitím algoritmu vektorového řízení se dosáhne precizního řízení momentu motoru, avšak za cenu značných výpočetních nároků. Pokud je potřeba řídicí algoritmus implementovatelný na levnější, méně výkonné mikroprocesorové sestavy, nabízí se algoritmus přímého řízení momentu (Direct Torque Control - DTC).

Mezi hlavní výhody uváděného algoritmu patří výborné dynamické vlastnosti motoru, díky bleskové změně řízení momentu motoru. Výstupem algoritmu DTC je spínací kombinace výkonových prvků frekvenčního měniče.



Obrázek 5.1: Frekvenční měnič s napěťovým meziobvodem

Na jednotlivé fáze motoru jsou přivedeny napětí odpovídající jednotlivým spínacím kombinacím (viz Tabulka 5.1). Základní algoritmus DTC charakterizují následující vlastnosti [5, 13]:

- Přímé řízení magnetického toku (pomocí správných kombinací napěťových vektorů).
- Rychlá odezva na změnu momentu motoru, vysoká dynamika.
- Jednoduchá realizace algoritmu

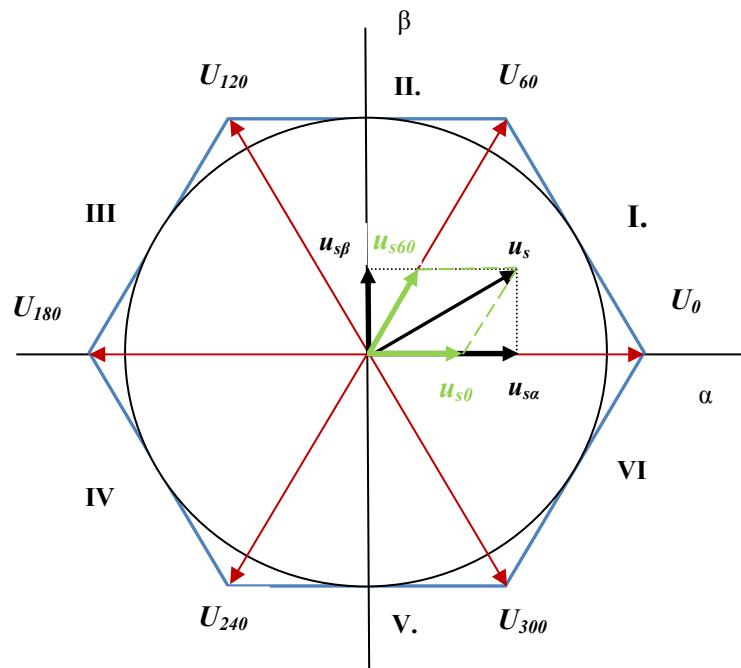
- Statorový proud obsahuje vyšší harmonické frekvence vyvolané spínáním frekvenčního měniče.
- Spínání může probíhat s frekvencí ve slyšitelném pásmu spektra a způsobovat hlučnost pohonu. Nelze odhadnout spínací ztráty výkonových prvků.

Vektor napětí	O_{000}	U_0	U_{60}	U_{120}	U_{180}	U_{240}	U_{300}	O_{111}
Sepnuté tranzistory	T_2, T_4, T_6	T_1, T_4, T_6	T_1, T_3, T_6	T_2, T_3, T_6	T_2, T_3, T_5	T_2, T_4, T_5	T_1, T_4, T_5	T_1, T_3, T_5
Spínací kombinace	[0,0,0]	[1,0,0]	[1,1,0]	[0,1,0]	[0,1,1]	[0,0,1]	[1,0,1]	[1,1,1]
u_A	0	$2/3 U_{ss}$	$1/3 U_{ss}$	$-1/3 U_{ss}$	$-2/3 U_{ss}$	$-1/3 U_{ss}$	$1/3 U_{ss}$	0
u_B	0	$-1/3 U_{ss}$	$1/3 U_{ss}$	$2/3 U_{ss}$	$1/3 U_{ss}$	$-1/3 U_{ss}$	$-2/3 U_{ss}$	0
u_C	0	$-1/3 U_{ss}$	$-2/3 U_{ss}$	$-1/3 U_{ss}$	$1/3 U_{ss}$	$2/3 U_{ss}$	$1/3 U_{ss}$	0

Tabulka 5.1 Spínací kombinace frekvenčního měniče

5.1 PŘÍMÉ ŘÍZENÍ MOMENTU ASYNCHRONNÍHO MOTORU

Princip řízení asynchronního motoru spočívá ve vytvoření točivého magnetického pole statoru pomocí spínání správné sekvence napěťových vektorů U_0 až U_{300} . Moment motoru závisí na úhlu mezi komplexory statorového ψ_s a rotorového magnetického toku ψ_r . Úhel mezi komplexory statorových toků, tedy moment motoru, můžeme měnit pomocí vhodně zvolených kombinací napěťových vektorů v závislosti na sektoru, ve kterém se nachází komplexor statorového magnetického toku ψ_s (Obrázek 5.2). Pokud se komplexor ψ_s bude nacházet v oblasti I. a bude se otáčet v kladném směru, tak přivedením napěťových vektorů U_{60} nebo U_{120} na stator motoru způsobíme zvýšení úhlu mezi komplexorem ψ_s a ψ_r a dojde ke zvýšení elektromechanického momentu. Naopak přivedení napěťových vektorů U_{240} nebo U_{300} způsobí zmenšení úhlu mezi komplexorem ψ_s a ψ_r a následnému poklesu elektromechanického momentu motoru [5, 13].



Obrázek 5.2: Grafická interpretace algoritmu DTC [5]

Řízení momentu je realizováno regulátorem s reléovou charakteristikou. Do regulátoru momentu vstupuje odchylka požadovaného momentu motoru T_{eref} od skutečného momentu motoru T_e . Požadovaný moment motoru T_{eref} se získá na výstupu regulátoru rychlosti z rozdílu požadovaných a měřených otáček motoru. Regulátor rychlosti se odvodí z rovnice (2.23). Typy regulátorů se obvykle volí typu P nebo PI. Moment motoru T_e se vypočítá ze znalosti satorových proudů a napětí v α - β souřadnicích podle rovnice

$$T_e = \frac{3}{2} P_p (\psi_{s\alpha} i_{s\beta} - \psi_{s\beta} i_{s\alpha}), \quad (5.1)$$

kde $\psi_{s\alpha}$ a $\psi_{s\beta}$ se vypočítá pomocí (4.3).

Výstup z reléového regulátoru momentu se řídí podle následujících pravidel

$$\delta_{T_e} = \begin{cases} 1 & T_e < T_{eref} - \varepsilon T_e \\ 0 & T_{eref} - \varepsilon T_e \leq T_e \leq T_{eref} + \varepsilon T_e, \\ -1 & T_e > T_{eref} + \varepsilon T_e \end{cases} \quad (5.2)$$

kde ε_{T_e} je pásmo necitlivosti. Pro řízení velikosti satorového toku je také použit regulátor s reléovou charakteristikou, jehož vstupní hodnotu tvoří odchylka

požadované velikosti statorového toku $|\Psi_{sref}|$ od skutečné velikosti magnetického toku $|\psi_s|$. Výstup z regulátoru se řídí pomocí následujících pravidel

$$\delta_{\psi_s} = \begin{cases} 1 & |\psi_s| < \psi_{sw} \\ 0 & |\psi_s| \geq \psi_{sw} \end{cases} \quad (5.3)$$

kde $|\psi_s|$ se vypočítá pomocí rovnice (2.19).

Z hlediska teorie nelineárních systémů by měla být použita reléová charakteristika s hysterezí, protože jinak může dojít k neřešitelnosti matematického modelu regulačního obvodu. Prakticky však je většinou použito relé bez hystereze, přičemž frekvence spínání je omezena periodou výpočtu algoritmu [5].

Poslední znalostí pro určení správné spínací kombinace frekvenčního měniče je sektor, ve kterém se nachází komplexor ψ_s . Sektor se vypočítá z úhlu (2.18)

$$\theta = \arctan\left(\frac{\psi_{s\beta}}{\psi_{s\alpha}}\right).$$

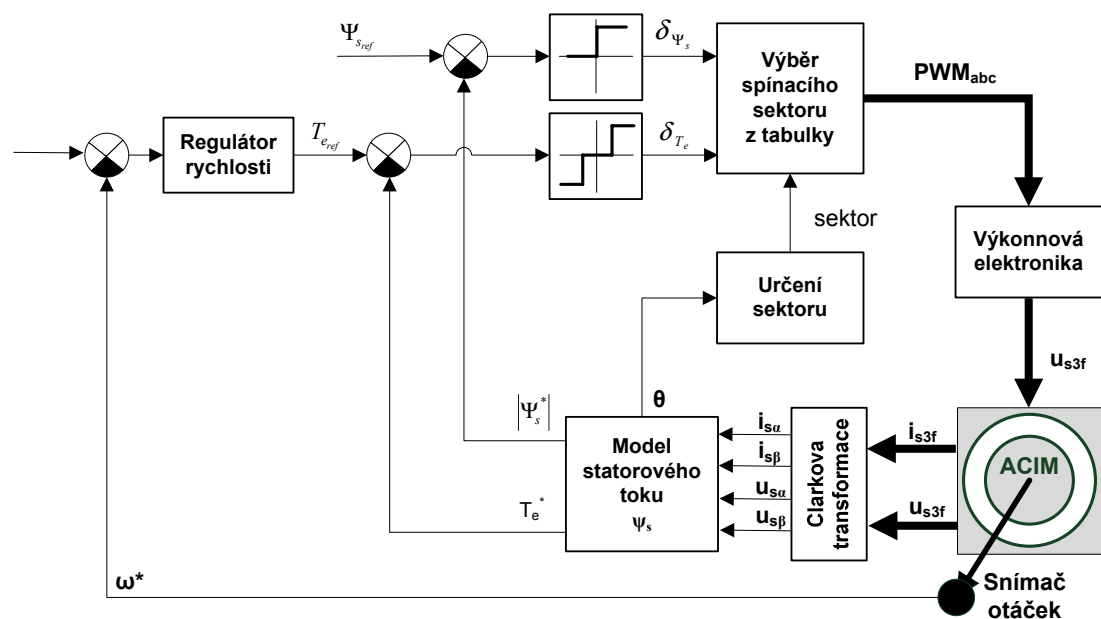
Spínací kombinaci pro frekvenční měnič můžeme určit z tabulky 5.2 podle výše uvedených výstupů z reléových regulátorů a sektoru výskytu ψ_s .

δ_{T_e}	δ_{ψ_s}	I.	II.	III.	IV.	V.	VI.
1	1	U ₆₀	U ₁₂₀	U ₁₈₀	U ₂₄₀	U ₃₀₀	U ₀
	0	U ₁₂₀	U ₁₈₀	U ₂₄₀	U ₃₀₀	U ₀	U ₆₀
0	1	O ₁₁₁	O ₁₁₁	O ₁₁₁	O ₁₁₁	O ₁₁₁	O ₁₁₁
	0	O ₀₀₀	O ₀₀₀	O ₀₀₀	O ₀₀₀	O ₀₀₀	O ₀₀₀
-1	1	U ₃₀₀	U ₀	U ₆₀	U ₁₂₀	U ₁₈₀	U ₂₄₀
	0	U ₂₄₀	U ₃₀₀	U ₀	U ₆₀	U ₁₂₀	U ₁₈₀

Tabulka 5.2: Pravidla spínání statorového napětí pro DTC algoritmus [5]

5.1.1 Proudové omezení u DTC

Proudové omezení je v této práci provedeno sepnutím nulového vektoru při detekci nadproudu. Výstup z reléového regulátoru momentu je nastaven na $\delta_{T_e} = 0$. Tento regulátor je nadřazen regulátoru magnetického toku (tabulka 5.2). Použití nulových vektorů O₀₀₀ nebo O₁₁₁ má za následek pokles statorového toku a brzdění rotoru v důsledku úbytku napětí na statorovém odporu způsobený protékajícím statorovým proudem [5, 6, 13].

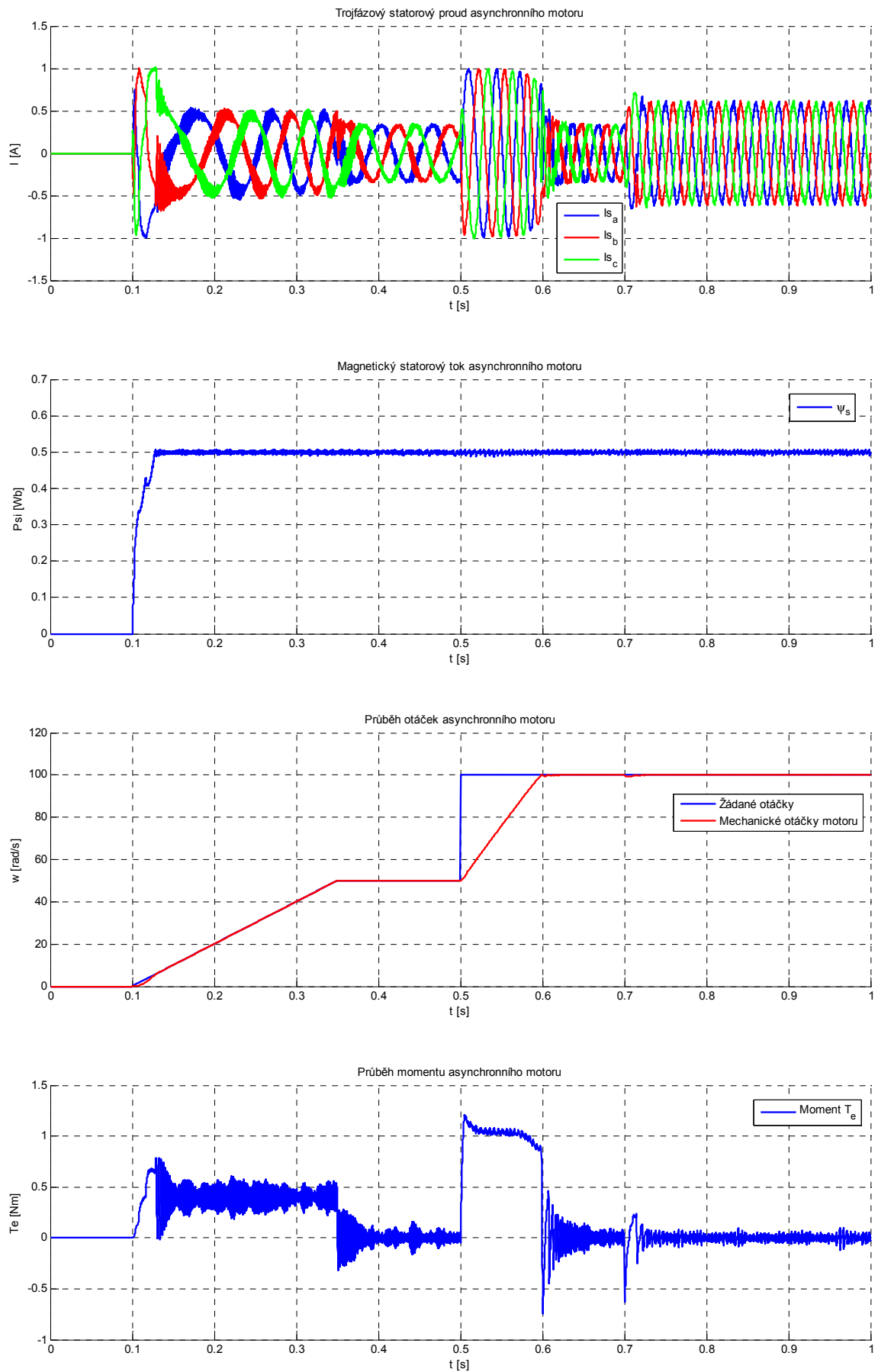


Obrázek 5.3: Blokové schéma regulačního obvodu s DTC algoritmem

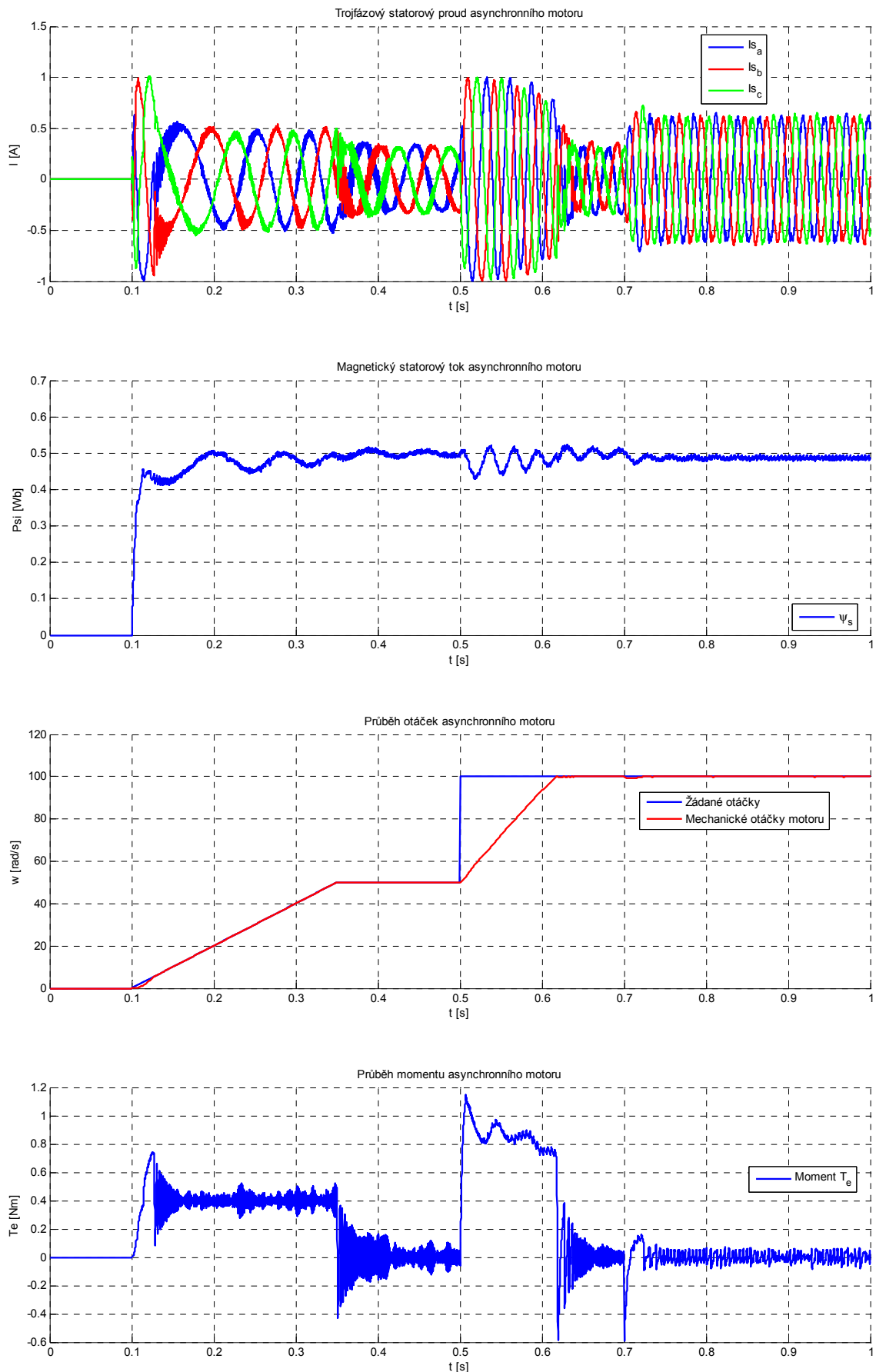
5.1.2 Simulace přímého řízení momentu asynchronního motoru

Průběh simulace je stejný jako u asynchronního motoru. Algoritmus DTC je charakteristický výbornou dynamikou, která je potvrzena simulací zobrazenou na obrázku 5.4. Odezva na změnu otáček je srovnatelná s vektorovým řízením. Oddělené řízení magnetického toku a momentu ukazuje druhý graf. Magnetický tok byl nastaven na hodnotu 0,5 Wb. Typický je zvlňný průběh momentu a proudu. Proudové omezení (kapitola 5.1.1) úspěšně limituje hodnotu proudu nastavenou pro demonstraci na 1 A.

Změna parametrů ASM měla za následek větší rozkmit magnetického toku, proudu a momentu. Proudové omezení a průběh otáček nejsou ovlivněny. Průběh simulace je na obrázku 5.5.



Obrázek 5.4: Simulace řízení ASM algoritmem DTC



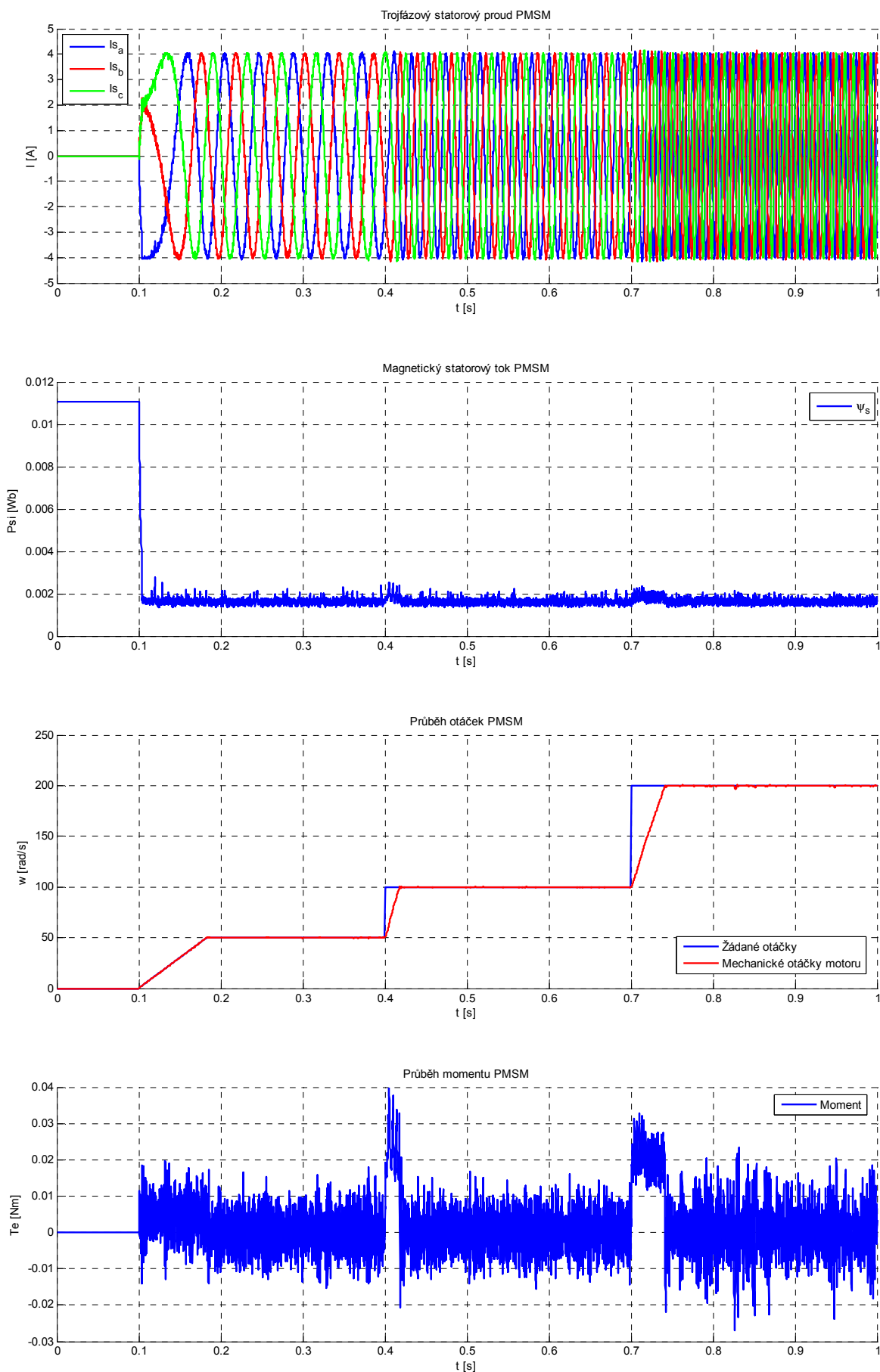
Obrázek 5.5: Simulace řízení ASM algoritmem DTC – změna parametrů motoru

5.2 PŘÍMÉ ŘÍZENÍ MOMENTU PMSM

Algoritmus přímého řízení momentu pro PMSM, použitý v této práci je prakticky totožný s popsaným algoritmem pro asynchronní motor.

5.2.1 Simulace přímého řízení momentu PMSM motoru

Na Obrázku 5.6 je zobrazen průběh jednotlivých veličin během simulace řízení PMSM motoru algoritmem DTC. Průběh statorového toku a momentu motoru má značně kmitavý charakter. Statorové proudy se neustále pohybují v proudovém omezení, které je nastaveno na 4 A. Rychlý nárůst proudu je způsoben díky velice malé časové konstantě motoru (zejména indukčnosti) vzhledem k velikosti napěťových vektorů aplikovaných na stator algoritmem DTC.



Obrázek 5.6: Simulace řízení PMSM pomocí DTC

6. ZPĚTNOVAZEBNÍ LINEARIZACE (VSTUP - VÝSTUP)

Cílem zpětnovazební linearizace je transformovat složitý nelineární systém na lineární, říditelný pomocí vhodných návrhových metod pro lineární systémy. Může probíhat na základě hledání vhodných transformací stavových proměnných a vstupů (linearizace vstup – stav), nebo systém linearizovat na základě popisu mezi jeho vstupem a výstupem (linearizace vstup – výstup).

Algoritmy popisované v této kapitole využívají k odvození algoritmu řízení linearizaci na základě vstupně-výstupních souvislostí, proto linearizace vstup – stav nebude dále podrobněji rozebírána. Více informací je možné nalézt např. v [5, 9].

Výpočet linearizace vstup – výstup probíhá opakovaným derivováním výstupu y u nelineárního systému s jedním vstupem a výstupem (SISO systém)

$$\begin{aligned}\dot{x} &= f(x) + g(x)u, \\ y &= h(x),\end{aligned}\tag{6.1}$$

až do okamžiku vzniku závislosti na vstupu u . Cílem je nahradit systém popsany rovnicí (6.1) řízený vstupní veličinou u , lineární náhradou se vstupem v

$$\begin{aligned}\dot{z} &= Az + Bv \\ \tilde{y} &= \tilde{h}(z).\end{aligned}\tag{6.2}$$

Lineární náhrada se volí jako libovolný stabilní lineární systém, avšak nejčastějším řešením je zvolit systém odpovídající sériově zapojeným integrátorům. Systémy (6.1) a (6.2) budou vstupně – výstupně ekvivalentní za předpokladu

$$\frac{d^i y}{dt^i} = \frac{d^i \tilde{y}}{dt^i} \quad i = 0, 1 \dots n,\tag{6.3}$$

což znamená, že při působení téhož vstupního signálu na oba systémy, bude mít výstupní signál stejný průběh u obou systémů. Výsledné vztahy linearizovaného systému se získají srovnáním derivací obou systémů.

Řešení zpětnovazebních linearizací je ve většině literatury popsáno pomocí matematického aparátu využívající Lieovy derivace.

Definice 6.1: Lieova derivace. Je dána hladká skalární funkce $h: \mathbb{R}^n \rightarrow \mathbb{R}$ a hladké vektorové pole $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$. Pak Lieova derivace $L_f h$ skalárního pole h vzhledem k vektorovému f je skalární funkce definovaná vztahem

$$L_f h = \nabla h \cdot f = \left(\frac{\partial h}{\partial x_1}, \dots, \frac{\partial h}{\partial x_n} \right) f(x) = \frac{\partial h}{\partial x} f(x), \quad (6.4)$$

kde ∇h je gradient funkce h (řádkový vektor).

Lieova derivace je směrová derivace skalárního pole h ve směru vektoru f .

Rekurzivně lze definovat vyšší derivace

$$\begin{aligned} L_f^2 h &= L_f(L_f h) = \nabla(L_f h) \cdot f & L_f^0 h &= h \\ L_f^i h &= L_f(L_f^{i-1} h) = \nabla(L_f^{i-1} h) \cdot f. \end{aligned} \quad (6.5)$$

Je-li g jiné vektorové pole, pak

$$L_g L_f h = \nabla(L_f h) \cdot g. \quad (6.6)$$

Pro nebužený dynamický systém $\dot{x} = f(x)$, $y = h(x)$ jsou derivace výstupu

$$\begin{aligned} \dot{y} &= \frac{\partial h}{\partial x} \dot{x} = \frac{\partial h}{\partial x} f(x) = L_f h \\ \ddot{y} &= \frac{\partial |L_f h|}{\partial x} \dot{x} = \frac{\partial |L_f h|}{\partial x} f(x) = L_f^2 h. \end{aligned} \quad (6.7)$$

Definice 6.1 byla převzata z [9].

Derivace výstupu y systému (6.1) s využitím Lieových derivací budou vypadat následovně

$$\begin{aligned} \dot{y} &= \nabla h \dot{x} = L_f h(x) + L_g h(x)u \\ \ddot{y} &= L_f^2 h(x) + L_g L_f h(x)u \\ &\vdots \\ &\vdots \\ &\vdots \end{aligned} \quad (6.8)$$

Derivace je opakována, dokud nebude splněna podmínka

$$L_g L_f^{r-1} h(x) \neq 0, \quad (6.9)$$

která znamená již zmíněnou závislost na vstupním signálu u .

Vstupní transformace linearizovaného systému je

$$u = \frac{1}{L_g L_f^{r-1} h} (-L_f^r + \vartheta), \quad (6.10)$$

kde r je relativní stupeň systému, počet derivací potřebný k tomu, aby se objevila závislost na vstupním signálu u . Relativní stupeň systému může být menší nebo roven počtu stavů původního systému. Pokud je relativní stupeň roven počtu stavů je linearizace vstup – výstup současně linearizací vstup – stav a systém je zcela lineární. V druhém případě bude obsahovat skrytou dynamiku, neboli bude obsahovat stavy, které jsou nepozorovatelné. Linearizace proběhne jen částečně. Má-li mít výsledek linearizace smysl, musí být skrytá dynamika stabilní. Vyšetření skryté dynamiky je značně komplikované a nebude v této práci rozebíráno, protože při aplikování v této práci ji není nutno vyšetřovat. U linearizace synchronního motoru se skrytá dynamika nevyskytuje a u asynchronního motoru je nepozorovatelný stav charakterizován úhlem natočení rotoru podle definice (6.33). Podrobnější informace je možné získat v [9, 4].

Při linearizaci matematických modelů synchronních a asynchronních motorů je nutné rozšířit teorii zpětnovazební linearizace ze SISO na MIMO systémy popsané rovnicí

$$\dot{x} = f(x) + \sum_{i=1}^m g_i(x) u_i \quad (6.11)$$

$$y_i = h_i(x) \quad i = 1, 2, \dots, n,$$

kde u_i jsou řídicí vstupy, y_i jsou výstupy, f a g jsou hladká vektorová pole a h_j je hladká skalární funkce. Při linearizaci MIMO postupujeme prakticky stejně jako u SISO systému. Nejprve je nutno provést Lieovy derivace výstupních funkcí dokud nebude splněna podmínka (6.9).

$$y_j^{r_j} = L_f^{r_j} h_j + \sum_{i=1}^m L_{g_i} L_f^{r_j-1} h_j(x) u_i, \quad (6.12)$$

kde $L_{g_i} L_f^{r_j-1} h_j(x) \neq 0$ alespoň pro jedno i a $\forall x \in R^n$ [9]. Pokud provedeme Lieovy derivace pro každou výstupní funkci dostaneme m rovnic, které můžeme zapsat v kompaktní formě

$$\begin{bmatrix} y_1^{(r_1)} \\ \dots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} h_1(x) \\ \dots \\ L_f^{r_m} h_m(x) \end{bmatrix} + \mathbf{D}(x) \begin{bmatrix} u_1 \\ \dots \\ u_m \end{bmatrix}, \quad (6.13)$$

kde $\mathbf{D}(x)$ je matice o rozměrech $m \times m$. Matice

$$\mathbf{D}(x) = \begin{pmatrix} L_{g_1} L_f^{r_1-1} h_1 & \dots & L_{g_m} L_f^{r_1-1} h_1 \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m & \dots & L_{g_m} L_f^{r_m-1} h_m \end{pmatrix} \quad (6.14)$$

musí být regulární v oblasti Ω [9]. Pokud je daná podmínka splněna, bude vstupní transformace platná

$$\begin{bmatrix} u_1 \\ \dots \\ u_m \end{bmatrix} = \mathbf{D}^{-1} \begin{bmatrix} -L_f^{r_1} h_1(x) + v_1 \\ \dots \\ -L_f^{r_m} h_m(x) + v_m \end{bmatrix}. \quad (6.15)$$

Mezi výstupy a novými vstupy \mathbf{v} jsou nyní lineární závislosti a navíc díky matici \mathbf{D} jsou odstraněny i interakce mezi původními vstupy a výstupy, což má nesmírnou výhodu, protože můžeme přistupovat k jednotlivým řídicím vstupům \mathbf{v} jako k samostatným SISO systémům. Matice \mathbf{D} je označována v anglické literatuře jako „**Decoupling matrix**“, matice odstranění křížných vazeb. Relativní stupeň je u linearizace MIMO systémů dán součtem jednotlivých stupňů pro jednotlivé výstupy

$$r = \sum_{j=1}^m r_j \leq n. \quad (6.16)$$

Pro precizní řízení střídavých strojů je nutné nezávisle řídit magnetický tok a moment motoru. Jak je patrné z rovnice (6.15) úspěšná aplikace zpětnovazební linearizace na nelineární MIMO systém umožní nejen použití algoritmů řízení aplikovatelných na lineární systémy, ale také oddělení vazeb mezi jeho vstupy a výstupy. Teorie zpětnovazební linearizace již byla zmíněna u algoritmů vektorového řízení k oddělení křížných vazeb od indukovaného napětí v kapitole 4.

Matematické modely asynchronních a synchronních strojů vedou i za předpokladu zjednodušujících podmínek na složité nelineární MIMO systémy. Další kapitoly se budou zabývat aplikací zpětnovazební linearizace vstup – stav na právě pro tyto motory.

6.1 ŘÍZENÍ ASYNCHRONNÍHO MOTORU POMOCÍ ZPĚTNOVAZEBNÍ LINEARIZACE

Matematický model 3f asynchronního motoru, odvozený za předpokladu zjednodušujících podmínek v kapitole 2.2.1, se musí pro aplikaci zpětnovazební linearizace upravit. Bude využíván proudově - tokový model v souřadnicích α - β ($\omega_K = 0$). Z rovnic (2.21) se vyjádří statorové toky a rotorové proudy a dosadí do (2.19-20). Po této úpravě budou stavové proměnné modelu jen statorové proudy a rotorové toky. Detailní odvození v [10, 11]. Získaný model můžeme zapsat v kompaktní formě pomocí rovnice (6.11) kde

$$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T = [\omega_m \ \Psi_{r\alpha} \ \Psi_{r\beta} \ i_{s\alpha} \ i_{s\beta}]^T, u = [u_\alpha \ u_\beta], \quad (6.17)$$

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \\ f_5(x) \end{bmatrix} = \begin{bmatrix} \mu(\Psi_{r\alpha} i_{s\beta} - \Psi_{r\beta} i_{s\alpha}) - \frac{Te}{J} \\ -\alpha\Psi_{r\alpha} - P_p\omega_m\Psi_{r\beta} + \alpha L_m i_{s\alpha} \\ -\alpha\Psi_{r\beta} + P_p\omega_m\Psi_{r\alpha} + \alpha L_m i_{s\beta} \\ \alpha\beta\Psi_{r\alpha} + P_p\beta\omega_m\Psi_{r\beta} - \gamma i_{s\alpha} \\ \alpha\beta\Psi_{r\beta} - P_p\beta\omega_m\Psi_{r\alpha} - \gamma i_{s\beta} \end{bmatrix}, \quad (6.18)$$

$$g_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \sigma L_s \end{bmatrix}, \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \sigma L_s \end{bmatrix}. \quad (6.19)$$

Pro zpřehlednění rovnic byly zavedeny konstanty [10]:

$$\begin{aligned} \sigma &= 1 - \left(\frac{L_m}{L_s L_r}\right), & \alpha &= \frac{R_r}{L_r}, & \beta &= \frac{L_m}{\sigma L_s L_r}, \\ \gamma &= \frac{L_m^2 R_s}{\sigma L_s L_r^2} + \frac{R_s}{\sigma L_s}, & \mu &= \frac{P_p}{J L_r}. \end{aligned} \quad (6.20)$$

Výstupní vektor, podle kterého bude matematický model linearizován, bývá nejčastěji druhá mocnina rotorového magnetického toku $\Psi_r^2 = \Psi_{r\alpha}^2 + \Psi_{r\beta}^2$ a poloha θ nebo rychlost asynchronního motoru ω_m . V této práci je zvoleno

$$y = \begin{bmatrix} h_1(x) \\ h_2(x) \end{bmatrix} = \begin{bmatrix} \omega_m \\ \Psi_{r\alpha}^2 + \Psi_{r\beta}^2 \end{bmatrix}. \quad (6.21)$$

Dalším krokem je výpočet Lieových derivací podle (6.12) dokud nebude splněna podmínka (6.9). První a druhá Lieova derivace výstupu otáček je

$$\begin{aligned} \dot{y}_1 &= \dot{h}_1(x) = L_f^1 h_1(x) + L_g h_1(x)u, \\ L_f^1 h_1(x) &= \mu(\Psi_{r\alpha} i_{s\beta} - \Psi_{r\beta} i_{s\alpha}) - \frac{Te}{J}, \\ L_g h_1(x) &= [0 \quad 0], \end{aligned} \quad (6.22)$$

$$\begin{aligned} \ddot{y}_1 &= \ddot{h}_1(x) = L_f^2 h_1(x) + L_g L_f h_1(x)u, \\ L_f^2 h_1(x) &= -\mu\beta P_p \omega_m (\Psi_{r\alpha}^2 + \Psi_{r\beta}^2) - \mu(\alpha + \gamma)(\Psi_{r\alpha} i_{s\beta} \\ &\quad - \Psi_{r\beta} i_{s\alpha}) - \mu P_p \omega_m (\Psi_{r\alpha} i_{s\alpha} + \Psi_{r\beta} i_{s\beta}), \\ L_g L_f h_1(x) &= \begin{bmatrix} -\frac{\mu}{\sigma L_s} \Psi_{r\beta} & \frac{\mu}{\sigma L_s} \Psi_{r\alpha} \end{bmatrix}. \end{aligned} \quad (6.23)$$

Jak je vidět při druhé Lieově derivaci výstupu y_1 je nalezena závislost na vstupu a podmínka (6.9) je splněna.

Nyní můžeme provést Lieovy derivace druhého výstupu y_2

$$\begin{aligned} \dot{y}_2 &= \dot{h}_2(x) = L_f^1 h_2(x) + L_g h_2(x)u, \\ L_f^1 h_2(x) &= \Psi_{r\alpha}^2 + \Psi_{r\beta}^2 \\ L_g h_2(x) &= 0. \end{aligned} \quad (6.24)$$

$$\begin{aligned} \ddot{y}_2 &= \ddot{h}_2(x) = L_f^2 h_2(x) + L_g L_f h_2(x)u, \\ L_f^2 h_2(x) &= (4\alpha^2 + 2\alpha^2 \beta L_m)(\Psi_{r\alpha}^2 + \Psi_{r\beta}^2) \\ &\quad + 2\alpha L_m P_p \omega_m (\Psi_{r\alpha} i_{s\beta} - \Psi_{r\beta} i_{s\alpha}) \\ &\quad - (6\alpha^2 L_m + 2\alpha\gamma L_m)(\Psi_{r\alpha} i_{s\alpha} + \Psi_{r\beta} i_{s\beta}) \\ &\quad + 2\alpha^2 L_m^2 (i_{s\alpha}^2 + i_{s\beta}^2), \\ L_g L_f h_2(x) &= \begin{bmatrix} \frac{2\alpha L_m}{\sigma L_s} \Psi_{r\alpha} & \frac{2\alpha L_m}{\sigma L_s} \Psi_{r\beta} \end{bmatrix}. \end{aligned} \quad (6.25)$$

Druhá Lieova derivace opět odhalila závislost na vstupu u . Linearizovaný přenos bude vypadat následovně

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= L_f^2 h_1 + L_{g1} L_f h_1 u_\alpha + L_{g2} L_f h_1 u_\beta \\ \dot{z}_3 &= z_4 \\ \dot{z}_4 &= L_f^2 h_2 + L_{g1} L_f h_2 u_\alpha + L_{g2} L_f h_2 u_\beta. \end{aligned} \quad (6.26)$$

Rovnice (6.26) se dá přepsat do maticového tvaru

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} L_f^2 h_1 \\ L_f^2 h_2 \end{bmatrix} + \mathbf{D}(\mathbf{x}) \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix}. \quad (6.27)$$

Matice odstranění křížných $\mathbf{D}(\mathbf{x})$ obsahovat členy

$$\begin{aligned} \mathbf{D}(\mathbf{x}) &= \begin{bmatrix} L_{g1} L_f h_1 & L_{g2} L_f h_1 \\ L_{g1} L_f h_2 & L_{g2} L_f h_2 \end{bmatrix} = \\ &= \begin{bmatrix} -\frac{\mu}{\sigma L_s} \Psi_{r\beta} & \frac{\mu}{\sigma L_s} \Psi_{r\alpha} \\ \frac{2\alpha L_m}{\sigma L_s} \Psi_{r\alpha} & \frac{2\alpha L_m}{\sigma L_s} \Psi_{r\beta} \end{bmatrix} \end{aligned} \quad (6.28)$$

a musí být regulární v oblasti Ω , aby bylo možné vypočítat její inverzi. Jak je vidět z rovnice (6.29), vyjadřující determinant matice \mathbf{D} , bude podmínka splněna pouze pro $\Psi^2 \neq 0$.

$$\det[\mathbf{D}] = -2 \frac{\alpha L_m \mu}{(\sigma L_s)^2} (\Psi_{r\alpha}^2 + \Psi_{r\beta}^2) = -2 \frac{\alpha L_m \mu}{(\sigma L_s)^2} \Psi^2. \quad (6.29)$$

Tato podmínka nebude splněna při rozběhu motoru, proto se musí v praxi zajistit prvotní nabuzení motoru a následné spuštění algoritmu. Po zajištění regulárnosti matice $\mathbf{D}(\mathbf{x})$ můžeme provést transformaci nových řídicích vstupů systému

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \mathbf{D}(\mathbf{x})^{-1} \begin{bmatrix} -L_f^2 h_1 + \mathbf{v}_1 \\ -L_f^2 h_2 + \mathbf{v}_2 \end{bmatrix}, \quad (6.30)$$

kde

$$\begin{cases} \mathbf{v}_1 = \dot{z}_1 \\ \mathbf{v}_2 = \dot{z}_3 \end{cases} \quad (6.31)$$

jsou nové řídicí vstupy systému. Rovnici (6.26) můžeme vzhledem k použité vstupní transformaci přepsat do tvaru

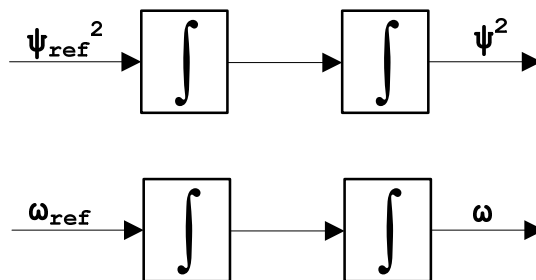
$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= v_1 \\ \dot{z}_3 &= z_4 \\ \dot{z}_4 &= v_2 \end{aligned} \quad (6.32)$$

Posledním krokem před návrhnutím řízení je vyšetření stability vnitřní dynamiky systému. Původní systém reprezentovaný rovnicí (6.18) je pátého řádu. Součet relativních stupňů systémů (6.32) podle rovnice (6.16) je roven čtyřem. Jelikož linearizovaný systém je nižšího řádu ($r < n$), bude obsahovat skrytou dynamiku a to konkrétně dynamiku 1. řádu. Stabilita vnitřní dynamiky nebude v tomto případě problém, nepozorovatelný stav je charakterizuje úhel natočení rotoru [10, 11] podle

$$z_5 = \arctan\left(\frac{\Psi_\beta}{\Psi_\alpha}\right), \quad (6.33)$$

$$z_5 = P_p z_1^1 + \frac{R_r}{P_p z_1^2} (J z_2^1 + T_e).$$

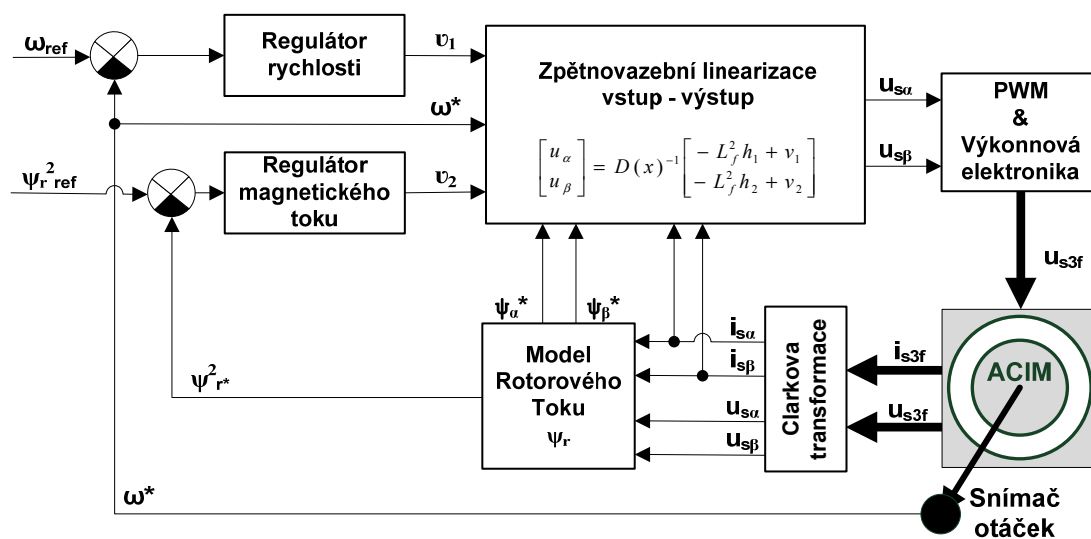
Po zajištění stability vnitřní dynamiky můžeme přistoupit k návrhování řízení. Původní nelineární systém 5. řádu se transformoval lineární systém 4. řádu (6.32). Výsledný systém umožňuje nezávislé řízení magnetického toku a otáček motoru. Soustava pro návržení obou regulačních smyček je tvořena dvěma integrátory viz obrázek 6.1.



Obrázek 6.1: Vstupně - výstupně linearizovaný systém

Regulátory použité v této práci jsou vzhledem k požadavku na nulovou ustálenou odchylku a robustnosti na změnu parametrů soustavy typu PID. Pro realizaci algoritmu musíme zajistit znalost složek statorových proudů a rotorového

toku nutné k výpočtu $D(x)$, $L_f^2 h_1$ a $L_f^2 h_2$ v rovnici (6.30). Pro výpočet složek rotorového toku možné použít modely popsané v kapitole 4.4.1. Blokové schéma řídicího algoritmu asynchronního motoru pomocí zpětnovazební linearizace vstup – výstup navrženého v této práci je na obrázku 6.2.

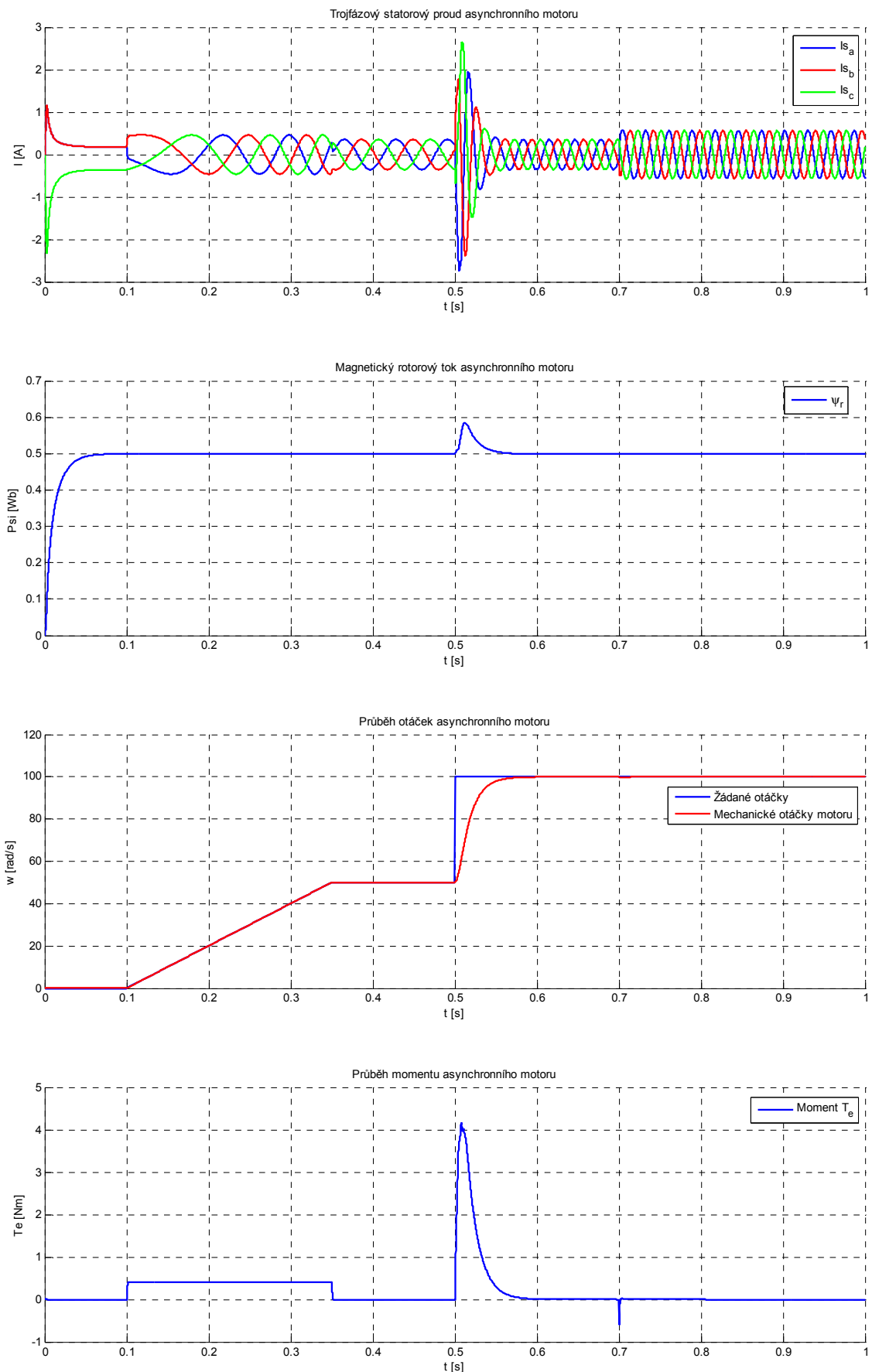


Obrázek 6.2: Blokové schéma řízení ASM pomocí zpětnovazební linearizace

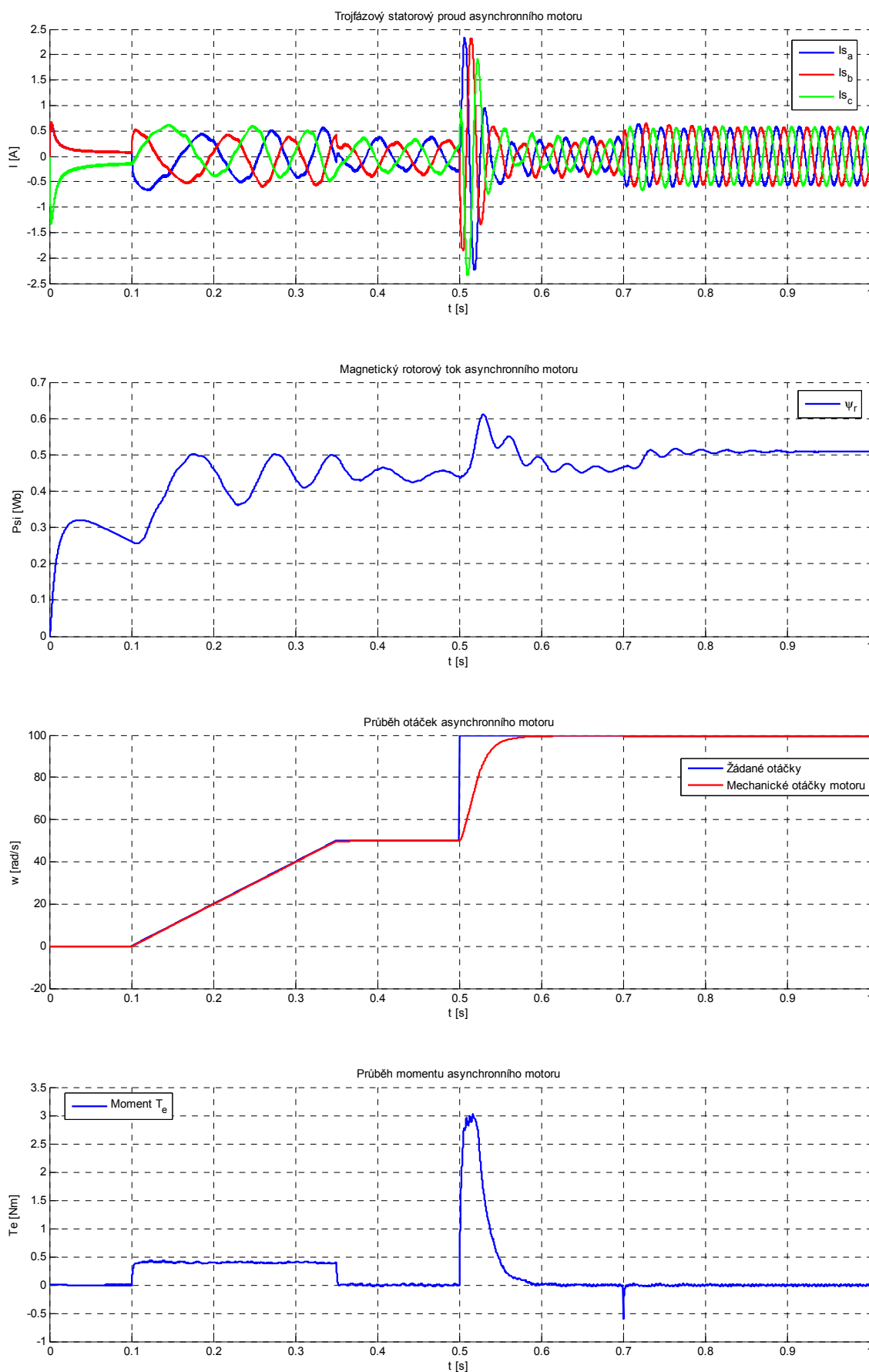
6.1.1 Simulace řízení asynchronního motoru pomocí zpětnovazební linearizace

Obrázek 6.3 poskytuje náhled na průběh jednotlivých veličin během simulace řízení pomocí zpětnovazební linearizace ASM. Nejprve byl nastaven magnetický tok na hodnotu 0,5 Wb a otáčky byly nastaveny na nulu až do času 0,1 s, kdy začal rozběh motoru po rampě na 50 rad/s. V čase 0,5 s následovala skoková změna otáček o 50 rad/s. Algoritmus reaguje na změny žádané hodnoty otáček s výbornou dynamikou. Vyrovnaní změny je rychlejší než u vektorového řízení nebo DTC. Rychlost je způsobená nepřítomností proudového omezení, která má za následek vyšší akční zásah a odpovídající zvýšení proudu. Zpětnovazební linearizace umožňuje nezávislé řízení otáček a rotorového magnetického toku. Na posledním grafu je vidět momentová charakteristika motoru. V čase 0,7 s byl motor zatížen (1 Nm), řídicí algoritmus okamžitě reaguje bez zákmitu v otáčkách.

Obrázek 6.4 zobrazuje průběh simulace zpětnovazební linearizace při změně parametrů ASM podle tabulky 2.3. Simulace má stejný průběh jako v předchozím případě. Změna parametrů způsobí rozkmit rotorového magnetického toku a zvlnění statorových proudů. Objevuje se také mírné zvlnění momentu. Průběh řízení otáček odpovídá průběhu na obrázku 6.3 až do okamžiku zatížení motoru. Otáčky po zatížení nepatrně klesnou.



Obrázek 6.3: Simulace řízení ASM pomocí zpětnovazební linearizace



Obrázek 6.4: Simulace řízení ASM pomocí zpětnovazební linearizace – změna parametrů motoru

6.2 ŘÍZENÍ PMSM MOTORU POMOCÍ ZPĚTNOVAZEBNÍ LINEARIZACE

Matematický model synchronního motoru je vhodné z důvodů popsaných v kapitole 3.1.1 v d-q souřadnicích spjatých s rotorem. Model synchronního motoru můžeme zapsat podle rovnice (6.11) jako

$$x = [x_1 \ x_2 \ x_3]^T = [i_d \ i_q \ \omega]^T, u = [u_\alpha \ u_\beta], \quad (6.34)$$

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{bmatrix} = \begin{bmatrix} a_1 x_1 + a_2 x_2 x_3 \\ b_1 x_2 + b_2 x_1 x_3 + b_3 x_3 \\ c_1 x_1 x_2 + c_2 x_2 \end{bmatrix}, \quad (6.35)$$

$$g_1 = \begin{bmatrix} 1 \\ L_d \\ 0 \\ 0 \end{bmatrix}, \quad g_2 = \begin{bmatrix} 0 \\ 1 \\ L_q \\ 0 \end{bmatrix}. \quad (6.36)$$

Pro zpřehlednění rovnic jsou zavedeny konstanty [12]:

$$\begin{aligned} a_1 &= -\frac{R_s}{L_d}, & a_2 &= \frac{P_p L_q}{L_d}, \\ b_1 &= -\frac{R_s}{L_q}, & b_2 &= -\frac{P_p L_d}{L_q}, & b_3 &= -\frac{P_p \psi_m}{L_q}, \\ c_1 &= \frac{3P_p(L_d - L_q)}{2J}, & c_2 &= \frac{3P_p \psi_m}{2J}. \end{aligned} \quad (6.37)$$

Prvním zvoleným výstupem je složka budícího proudu i_d , protože pro zajištění maximálního momentu synchronního motoru je potřeba řídit budící složku proudu i_d na nulovou hodnotu. Druhým výstupem je možno zvolit polohu, nebo rychlost synchronního motoru jako bylo zvoleno v této práci

$$y = \begin{bmatrix} h_1(x) \\ h_2(x) \end{bmatrix} = \begin{bmatrix} i_d \\ \omega \end{bmatrix}. \quad (6.38)$$

Dalším krokem je vypočet Lieových derivací podle (6.12) dokud nebude splněna podmínka (6.9). První a druhá lieova derivace výstupu proudu i_d je

$$\begin{aligned} \dot{y}_1 &= \dot{h}_1(x) = L_f^1 h_1(x) + L_g h_1(x)u, \\ L_f^1 h_1(x) &= \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 x_3 = \mathbf{f}_1(x), \\ L_g h_1(x) &= \begin{bmatrix} 1 \\ L_d \end{bmatrix}. \end{aligned} \quad (6.39)$$

V případě Lieovy derivace výstupu y_1 synchronního motoru je nalezena závislost na vstupu a podmínka (6.9) je splněna již při první derivaci.

Nyní můžeme provést Lieovy derivace druhého výstupu y_2

$$\begin{aligned} \dot{y}_2 &= \dot{h}_2(x) = L_f^1 h_2(x) + L_g h_2(x)u, \\ L_f^1 h_2(x) &= c_1 x_1 x_2 + c_2 x_2 = f_3(x), \\ L_g h_2(x) &= [0 \quad 0]. \end{aligned} \quad (6.40)$$

$$\begin{aligned} \ddot{y}_2 &= \ddot{h}_2(x) = L_f^2 h_2(x) + L_g L_f h_2(x)u, \\ L_f^2 h_2(x) &= \mathbf{c}_1 x_2 \mathbf{f}_1(x) + \mathbf{f}_2(x)(\mathbf{c}_2 + \mathbf{c}_1 x_1), \\ L_g L_f h_2(x) &= \begin{bmatrix} \mathbf{c}_1 x_2 & (\mathbf{c}_1 x_1 + \mathbf{c}_2) \\ L_d & L_q \end{bmatrix}. \end{aligned} \quad (6.41)$$

Druhá Lieova derivace opět odhalila závislost na vstupu u . Linearizovaný přenos bude vypadat následovně

$$\begin{aligned} \dot{z}_1 &= L_f^2 h_1 + L_{g1} L_f h_1 u_d \\ \dot{z}_2 &= z_3 \\ \dot{z}_3 &= L_f^2 h_2 + L_{g1} L_f h_2 u_d + L_{g2} L_f h_2 u_q. \end{aligned} \quad (6.42)$$

Rovnice (6.26) se dá přepsat do maticového tvaru

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} L_f^1 h_1 \\ L_f^2 h_2 \end{bmatrix} + \mathbf{D}(x) \begin{bmatrix} u_d \\ u_q \end{bmatrix}. \quad (6.43)$$

Matice odstranění křížných $\mathbf{D}(x)$ obsahovat členy

$$\mathbf{D}(x) = \begin{bmatrix} L_{g1} h_1 & L_{g2} h_1 \\ L_{g1} L_f h_2 & L_{g2} L_f h_2 \end{bmatrix} = \quad (6.44)$$

$$= \begin{bmatrix} \frac{1}{L_d} & 0 \\ \frac{c_1 x_2}{L_d} & \frac{(c_1 x_1 + c_2)}{L_q} \end{bmatrix}$$

a musí být regulární v oblasti Ω , aby bylo možné vypočítat její inverzi. Jak je vidět z rovnice (6.45), vyjadřující determinant matice D , podmínka nebude splněna pouze pro $c_1 x_1 = -c_2$. Prakticky však k této situaci nemůže dojít, jelikož proud i_d je regulován a nulovou hodnotu.

$$\det[D] = \frac{(c_1 x_1 + c_2)}{L_q L_d}. \quad (6.45)$$

Po zajištění regulárnosti matice $D(\mathbf{x})$ můžeme provést transformaci nových řídicích vstupů systému

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = D(\mathbf{x})^{-1} \begin{bmatrix} -L_f^1 h_1 + \mathbf{v}_1 \\ -L_f^2 h_2 + \mathbf{v}_2 \end{bmatrix}, \quad (6.46)$$

kde

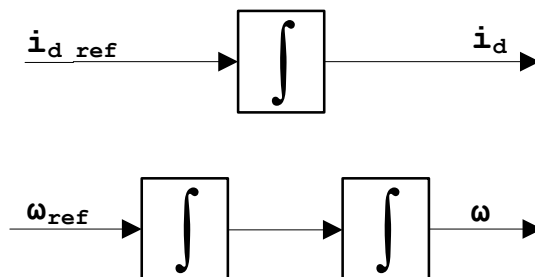
$$\begin{cases} \mathbf{v}_1 = \dot{z}_1 \\ \mathbf{v}_2 = \dot{z}_2 \end{cases} \quad (6.47)$$

jsou nové řídicí vstupy systému. Rovnici (6.26) můžeme vzhledem k použité vstupní transformaci přepsat do tvaru

$$\begin{cases} \dot{z}_1 = \mathbf{v}_1 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = \mathbf{v}_2 \end{cases} \quad (6.48)$$

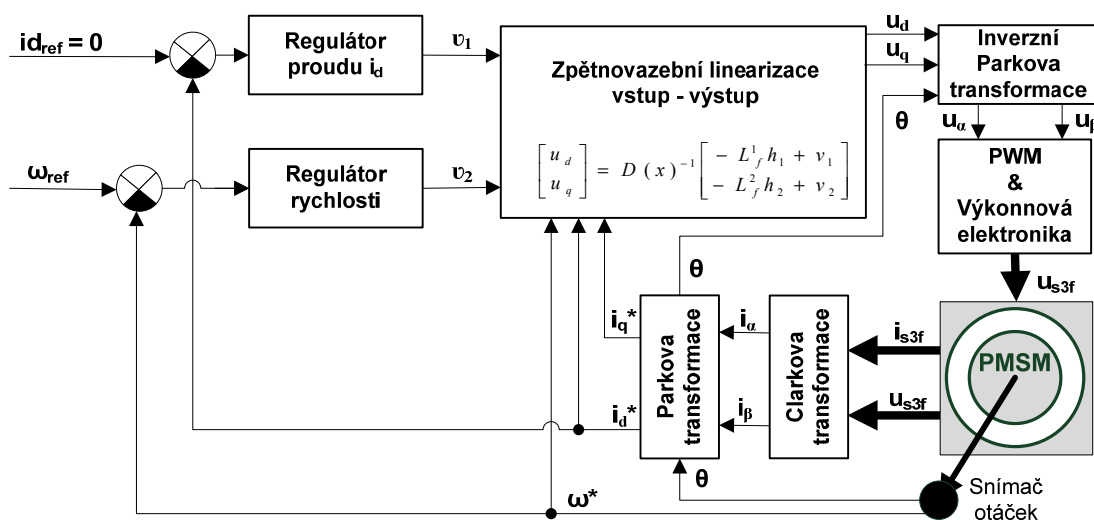
Posledním krokem před navrnutím řízení je vyšetření stability vnitřní dynamiky systému. Původní systém reprezentovaný rovnicí (6.35) je třetího řádu. Součet relativních stupňů systémů (6.48) podle rovnice (6.16) je roven třem. Jelikož linearizovaný systém je stejného řádu ($r = n$), nebude obsahovat skrytou dynamiku jako v případě asynchronního motoru.

Díky nepřítomnosti skryté dynamiky v systému (6.48), můžeme přistoupit k navrhování řízení. Výsledný systém zajistí řízení otáček PMSM s maximálním momentem, díky regulace proudu i_d na nulovou hodnotu. Řízení proudu i_d spočívá v řízení jediného integrátoru. Soustava pro návrh regulátoru otáček je reprezentována dvojicí integrátorů viz Obrázek 6.5



Obrázek 6.5 Vstupně-výstupně linearizovaný synchronní motor

Regulátory pro řízení synchronního motoru jsou z důvodu popsaných v kapitole 6.1 zvoleny jako PI pro proudovou smyčku a jako PID pro otáčkovou smyčku. Pro výpočet $D(x)$, $L_f^1 h_1$ a $L_f^2 h_2$ není potřeba vypočítávat další proměnné, jako tomu bylo u asynchronního motoru. Musíme však přidat Parkovu a Inverzní Parkovu transformaci proudů, napětí pro transformaci ze souřadnic α - β do d - q a naopak. Blokové schéma řídicího algoritmu zpětnovazební linearizace vstup – výstup pro PMSM navrženého v této práci je na obrázku 6.6.



Obrázek 6.6: Blokové schéma řízení PMSM pomocí zpětnovazební linearizace

6.2.1 Proudové omezení

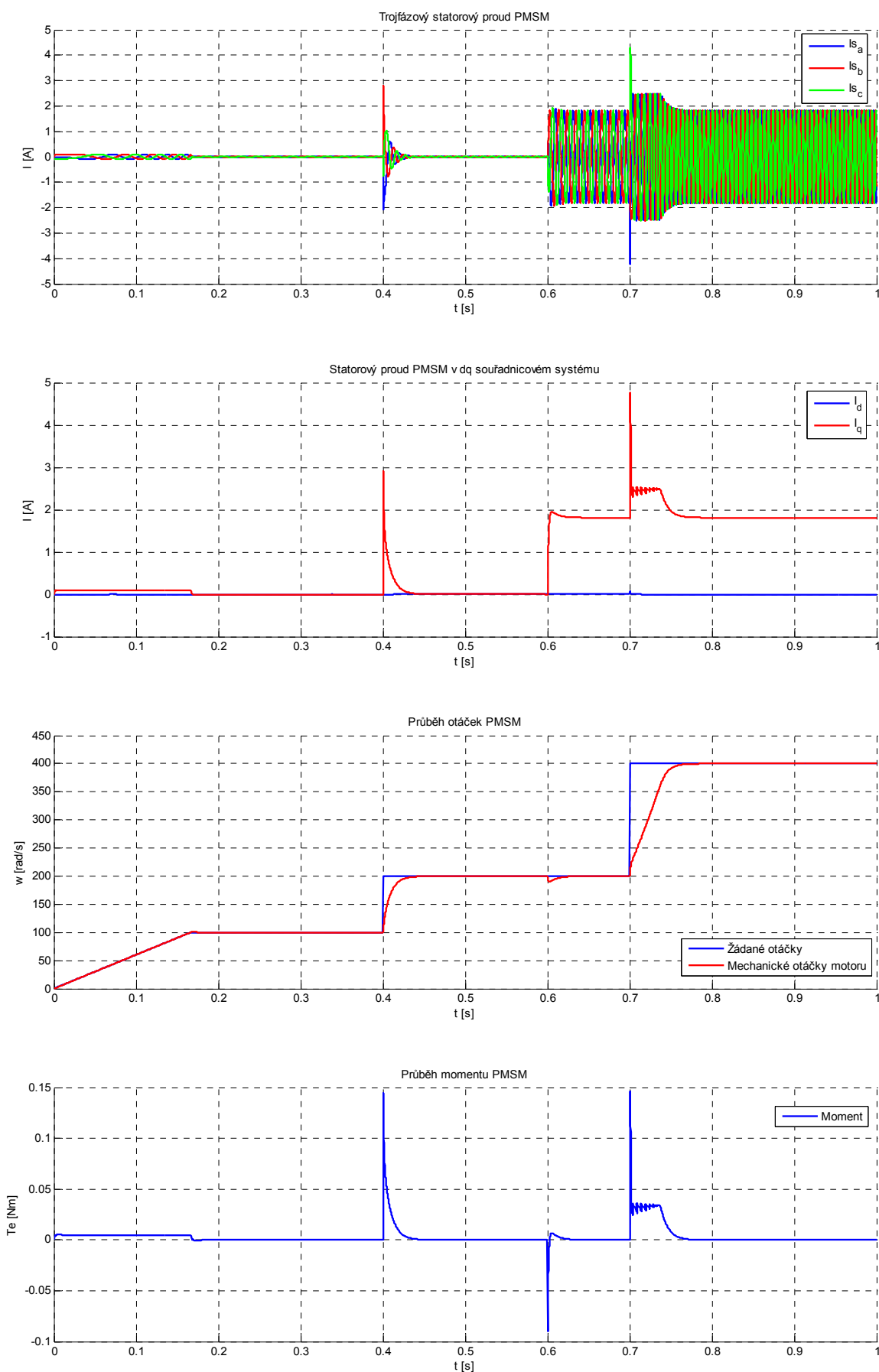
Algoritmus popisovaný v této kapitole byl otestován na reálném motoru firmy TG Drives (tabulka 3.1) popsané v kapitole 3.2. Aby mohl algoritmus rozumně fungovat na reálných motorech, musí obsahovat omezení proudu a napětí. Omezení napětí se dá jednoduše provést saturací z výpočtu absolutní hodnoty vektorových složek napětí u_d , u_q (viz kapitola 4.1.3), jelikož jsou k dispozici na výstupu algoritmu. Proudové omezení je však u toho algoritmu problém, neobsahuje podřízenou proudovou smyčku jako v případě vektorového řízení, nebo možnost řízení momentu (proudu) jako u DTC, protože je linearizován až na otáčkovou zpětnou vazbu.

Omezení proudu u zpětnovazební linearizace je provedeno změnou polarity akčního zásahu při detekci nadproudu, jenž odpovídá zrychlení, respektive zpomalení a má přímou vazbu na tvorbu výstupní složky napětí u_q (6.46). Nutností je také použití „anti - windup“ pro mezení růstu integrační složky v režimu nadproudu. Při simulacích v Simulinku se použitelnost proudového omezení ukázala jako funkční, avšak při výpočtu na procesoru a na reálném přípravku nebyla funkčnost ověřena. Docházelo k značnému rozkmitání proudu při působení proudového omezení, což by ve výsledku až tak nevadilo, neboť se jednalo o velice krátkou dobu. Největším problémem nemožnost zabránit počátečnímu překmitu proudu, který při velkých skokových změnách žádané hodnoty způsobovalo spuštění proudových ochran v motoru a jeho následné zastavení. Regulátory z důvodů omezení při implementaci (8.15) generují velký akční zásah, který má za následek velkou změnu vstupního napětí. Tato skoková změna napětí způsobí prudký nárůst proudu a následné spuštění proudových ochran motoru. Funkčnost proudového omezení nebyla prokázána, proto není zakreslena v blokovém schématu algoritmu (Obr. 6.6).

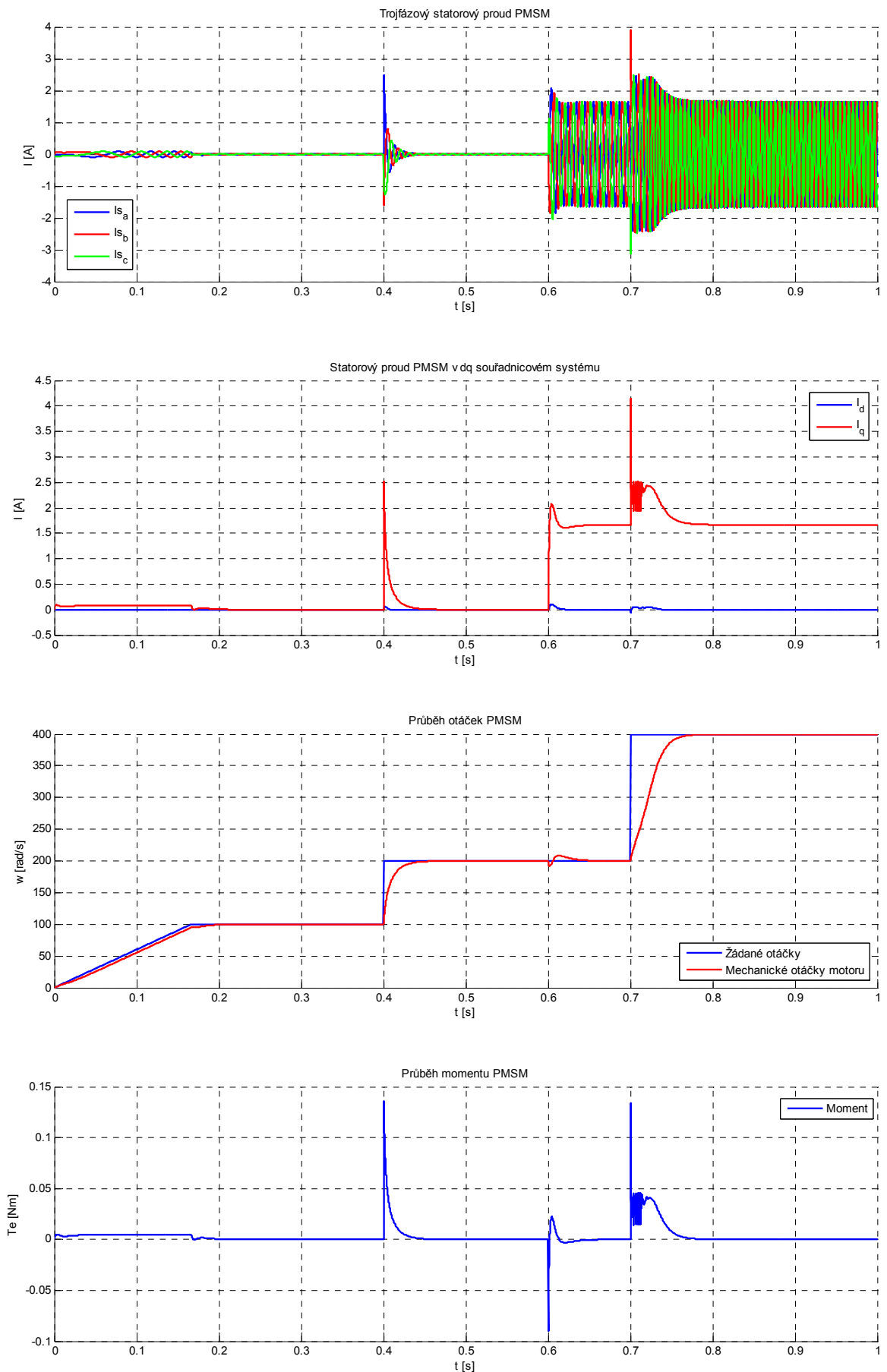
6.2.2 Simulace řízení PMSM pomocí zpětnovazební linearizace

Průběh řízení PMSM pomocí zpětnovazební linearizace je zachycena na obrázcích 6.7 a 6.8. Algoritmus umožňuje řízení PMSM s maximálním momentem, protože složka proudu i_d je řízena na nulovou hodnotu. V čase 0,6 s je motor zatížen 0,1 Nm. Zátěž způsobí patrný zákmit v otáčkách. Proudové omezení, popsané v kapitole 6.2.1, je nastaveno na hodnotu 2,5 A, jako v případě vektorového řízení. Zesílení otáčkového regulátoru bylo nastaveno na hodnotu ($K = 100000$) odpovídající nastavení při implementaci na procesoru (viz (8.15)). Velký akční zásah má za následek rychlý vzrůst napětí, respektive proudu a způsobuje velký počáteční překmit, kterému proudové omezení není schopno zabránit.

Obrázek 6.8 zachycuje průběh simulace při změně parametrů PMSM (tabulka 3.2). Změna parametrů se projevila vyšším rozkmitem otáček při zatížení motoru.



Obrázek 6.7: Simulace řízení PMSM pomocí zpětnovazební linearizace



Obrázek 6.8: Simulace řízení ASM pomocí zpětnovazební linearizace – změna parametrů motoru

7. KNIHOVNA ŘÍDICÍCH ALGORITMŮ PRO MATLAB - SIMULINK

Řídicí algoritmy pro řízení asynchronních a synchronních motorů popsaných v této práci obsahují velké množství nastavitelných konstant a možností jednotlivých variant provedení. Aby bylo možné tyto algoritmy v budoucnu dále efektivně využívat, je potřeba pro každý algoritmus vytvořit smysluplné uživatelské rozhraní, které ucelí a zpřehlední značné množství konstant a nastavení. Program Matlab - Simulink umožňuje vytvářet uživatelům vlastní knihovny algoritmů a implementovat je do Simulinku. Knihovnu si může každý uživatel přidat do svého Simulinku a využívat její funkce.

7.1 VYTVOŘENÍ KNIHOVNY PRO MATLAB - SIMULINK

Vytvoření knihovny v Matlabu je velice triviální záležitost díky přichystané m-funkci „*sblocks.m*“. V tomto „m-filu“ je nutno nadefinovat pouze jméno subsystému a soubor *.mdl, který obsahuje vytvořené funkční bloky. Všechny soubory včetně souborů potřebných k běhu funkčních bloků se nakopírují do adresáře a tento adresář se přidá do „*Matlab search path*“ (File→Set path...). Vytvořená knihovna je přidána do seznamu knihoven Simulinku při jeho opětovném spuštění.

7.2 VYTVOŘENÍ KNIHOVNÍCH FUNKČNÍCH BLOKŮ

K realizaci samotných funkčních bloků pro jednotlivé řídicí algoritmy je nejprve nutné zvolit vhodný blok subsystému, do kterého je možné vložit jednotlivé algoritmy pomocí blokové algebry. Jako nejvhodnější subsystém byl vybrán blok „Model“ (Ports & Subsystems). Tento blok funguje jako reference na další modely, proto je naprosto ideální volbou vzhledem ke složitosti jednotlivých algoritmů. Díky možnosti reference na libovolný počet modelů je možné pro každou variantu algoritmu vytvořit samostatné, přehledné regulační schéma. K libovolnému schématu

je samozřejmě přidána možnost nastavení aktuálních hodnot parametrů pro danou simulaci. Další nespornou výhodou je počet vstupních, výstupních portů z funkčního bloku odpovídající počtu portů aktuálního zvoleného modelu.

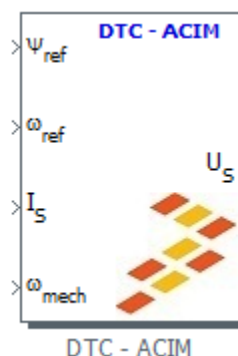
7.2.1 „Mask Editor“

Pokud potřebujeme specifikovat vzhled funkčního bloku, nastavit proměnné, které se mají předávat do obsaženého modelu, či vytvořit inicializační příkaz, musíme bloku vytvořit masku (obrázek 7.1). Masku lze snadno tvořit pomocí Mask Editoru (kliknout pravým tlačítkem na blok → Mask subsystem). Editor obsahuje čtyři záložky: Icon, Parameters, Initialization a Documentation.

Záložka Icon pomůže dodat blokům profesionální vzhled díky možnostem změn barev, zobrazování obrázků, kreslení, popisů portů, formátovaného textu atd. Za zmínku stojí možnost psát text pomocí LaTeX notace: `port_label(,input',1, '\psi_{ref}', 'texmode', 'on');` tento příkaz zobrazí u prvního vstupního portu symbol ψ_{ref} .

V záložce Parameters je možné definovat společně s krátkým popisem proměnné, jejichž hodnoty se budou předávat do těla bloku. Následným dvojklikem na funkční blok se zobrazí seznam proměnných společně s popisem.

Záložka Initialization umožňuje vkládat inicializační příkazy a poslední záložka umožňuje doplnit blok o textový popis masky.



Obrázek 7.1: Příklad masky – blok s algoritmem DTC

7.3 TVORBA GRAFICKÉHO UŽIVATELSKÉHO ROZHRAŇÍ

Jak již bylo zmíněno v kapitole 7.2.1, můžeme pomocí masky definovat a zadávat proměnné, které se budou předávat do těla bloku. Řídící algoritmy však obsahují velké množství proměnných a nedají se v možnostech masky přehledně rozčlenit a zobrazit. Masku použijeme na definici proměnných, které chceme předat do aktuálního modelu bloku a na inicializační příkaz, který ho vykoná. Na zadávání parametrů bylo vytvořeno samostatné grafické uživatelské rozhraní (GUI – „Graphical User Interface“) pomocí „Layout Editoru“. Tento editor umožňuje vkládat, rozmísťovat komponenty typu: tlačítko, textové pole, slider, panel, grafy adt. a naprogramovat jejich funkci. Jednotlivé hodnoty proměnných se dají předávat mezi maskou a GUI pomocí příkazů: „*get_param*“ a „*set_param*“. Samotné spuštění obstará „*OpenFcn callback*“ daného bloku pomocí příkazu „*run*“.

7.4 NÁPOVĚDA MATLABU

Výše popsané kapitoly popisují stručný postup pro vytvoření knihovny v programu Matlab - Simulink, jednotlivých knihovnických bloků (použitých v této práci), tvorbu masek a GUI. Podrobné informace k řešení problematiky z výše popsaných kapitol je možné najít v nápovědě Matlabu. Stačí vyhledat klíčová slova: „*Model*“, „*Mask Editor*“, „*guide*“, „*set_param*“, „*get_param*“, „*Callback*“, „*OpenFcn*“, „*run*“.

7.5 INSTALACE KNIHOVNY

Instalace knihovny je velmi jednoduchá, stačí obsah knihovny z příloženého DVD nakopírovat do libovolného adresáře na pevném disku. Zvolený adresář, včetně všech podadresářů, se přidá do „*Matlab search path*“ („*File*“ → „*Set path*“ → „*Add with subfolders...*“ → „*save*“). Po spuštění Simulinku je obsah knihovny k dispozici pod názvem „*Motor Control Toolbox*“. Knihovna obsahuje šestici bloků, přičemž každý blok reprezentuje konkrétní algoritmus viz kapitola 7.6.

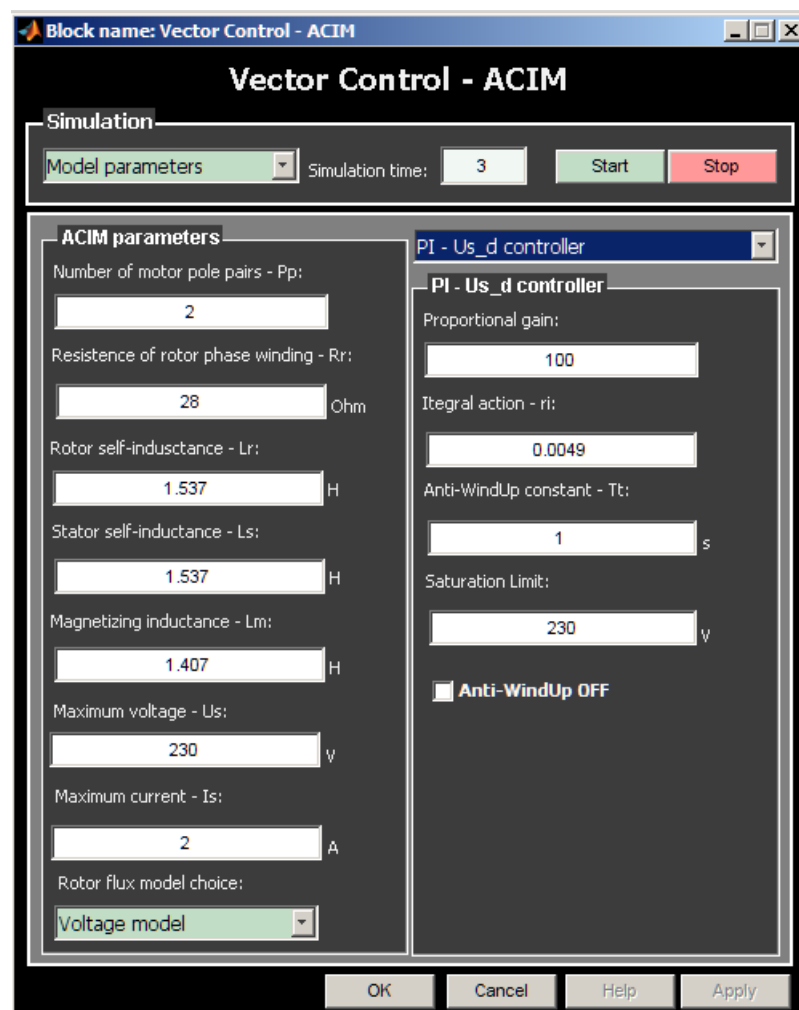
7.6 POPIS JEDNOLIVÝCH BLOKŮ KNIHOVNY - GUI

GUI se aktivuje dvojklikem levého tlačítka myši na jednotlivý blok. Možnosti nastavení se liší v závislosti na potřebách jednotlivých algoritmů, avšak všechny GUI mají společný panel „Simulation“. Tento panel obsahuje ovládací prvky pro nastavení délky času simulace („Simulation time“), dvojici tlačítek pro spuštění a zastavení simulace. Posledním prvkem v panelu „Simulation“ je roletové menu, které slouží k přepínání mezi nastavením parametrů algoritmů a vnitřním vstupem. Vnitřní vstup umožňuje rychle zadání žádané hodnoty otáček nebo magnetického toku, bez nutnosti generovat žádanou hodnotu pomocí bloků Simulniku. Hodnoty lze měnit i v průběhu simulace, proto lze snadno generovat např. skokovou změnu otáček.

Veškeré vstupy pro nastavování parametrů jsou ošetřeny proti zadání nekorektních vstupů a při jejich detekci je uživatel upozorněn a informován, v jakém formátu má vstup zadat. Při korektní změně parametrů je nutné změnu potvrdit. Uživatel je opět upozorněn změnou barvy editovaného parametru a aktivací tlačítka „Apply“, které po stisknutí uloží změny parametrů. Změnu lze také potvrdit tlačítkem „Ok“, který však standardně po uložení změny GUI ukončí. Tlačítko „Cancel“ neuloží změnu a ukončí GUI. Změny parametrů algoritmu lze opět zadávat i během simulace a simulovat např. změny parametrů motoru vlivem teploty.

7.6.1 Vektorové řízení ASM – GUI

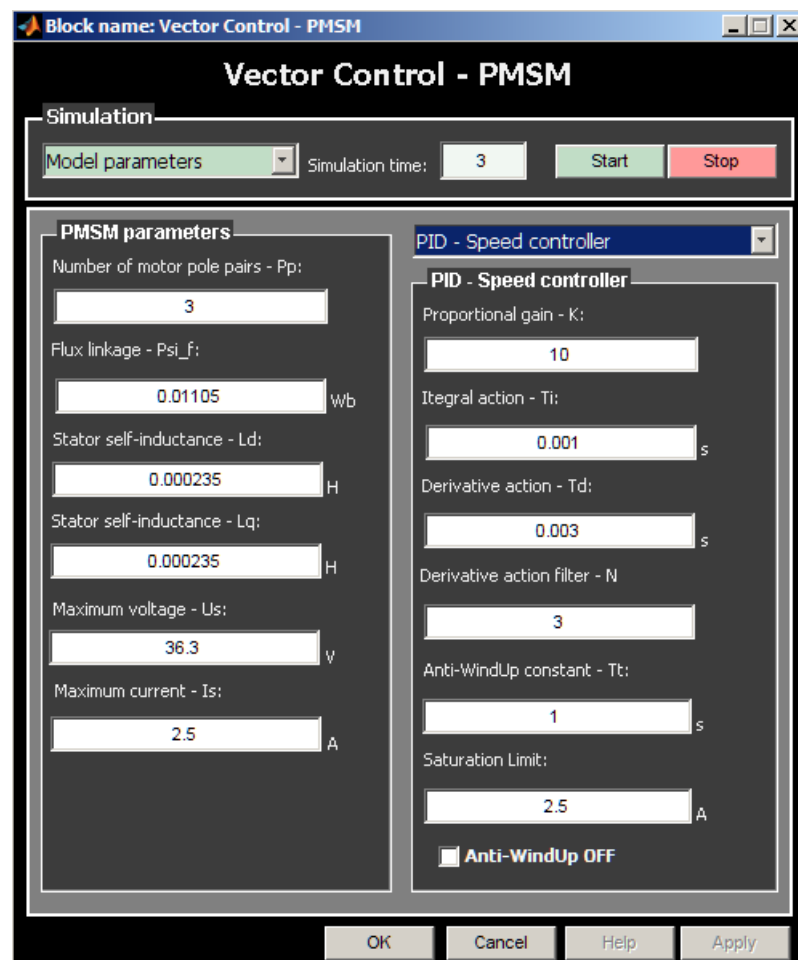
GUI tohoto algoritmu (Obrázek 7.2) obsahuje parametry motoru nutné pro výpočet „Decouplingu“, modelu rotorového toku (kapitola 4.1.1) a volbu samotného modelu rotorového toku reprezentovanou roletovým menu („Rotor Flux Model Choice“). Dále se nastavuje maximální statorové napětí a proud. Algoritmus vektorového řízení obsahuje čtyři regulátory, dva PI regulátory pro regulaci složek proudu i_{sd} a i_{sq} a dva PID regulátory k řízení otáček a magnetického toku. Mezi regulátory se přepíná pomocí roletového menu pod tlačítka „Start“ a „Stop“. Každý regulátor obsahuje nastavení konstant v závislosti na použitém typu, dále maximální výstup regulátoru a časovou konstantu pro anti-windup, který se dá kompletně vypnout zaškrtnutím políčka „Anti – windUp - Off“.



Obrázek 7.2: Vektorové řízení ASM - GUI

7.6.2 Vektorové řízení PMSM – GUI

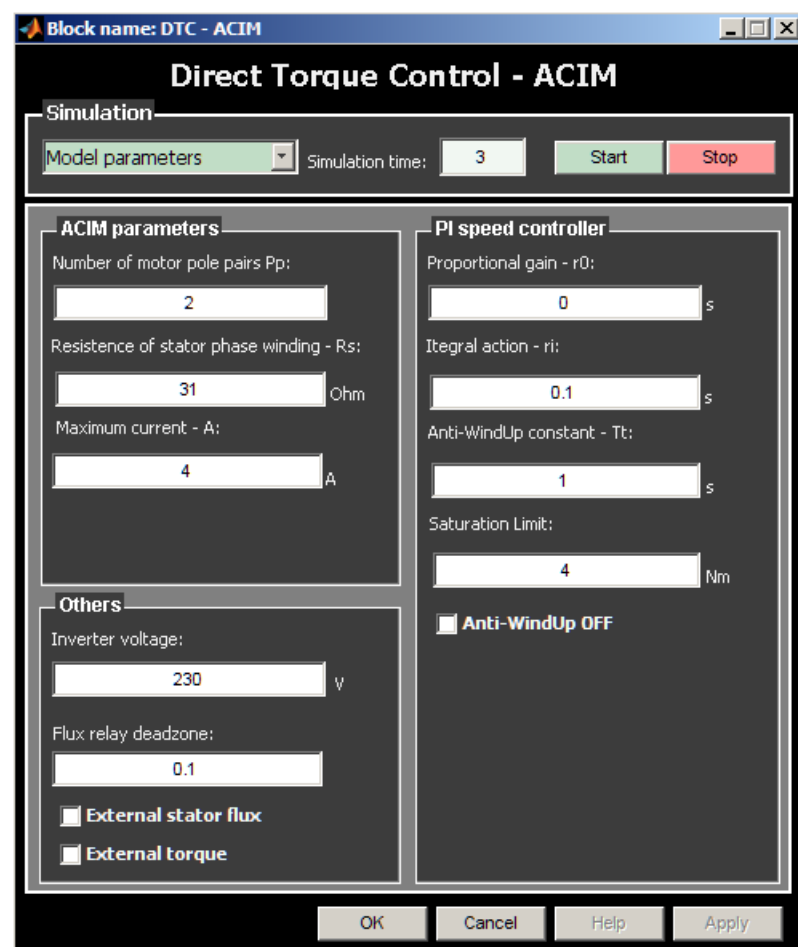
Vektorové řízení PMSM (Obrázek 7.3) obsahuje jen parametry motoru pro výpočet „Decouplingu“, nastavení maximálního statorového proudu a napětí. Roletové menu pod tlačítky „Start“ a „Stop“ obsahuje tři regulátory. Dva PI regulátory pro řízení složek proudu i_{sd} , i_{sq} a jeden PID regulátor k řízení otáčkové zpětné vazby. Nastavení konstant závisí opět na použitém typu regulátoru.



Obrázek 7.3: Vektorové řízení PMSM - GUI

7.6.3 Přímé řízení momentu ASM a PMSM – GUI

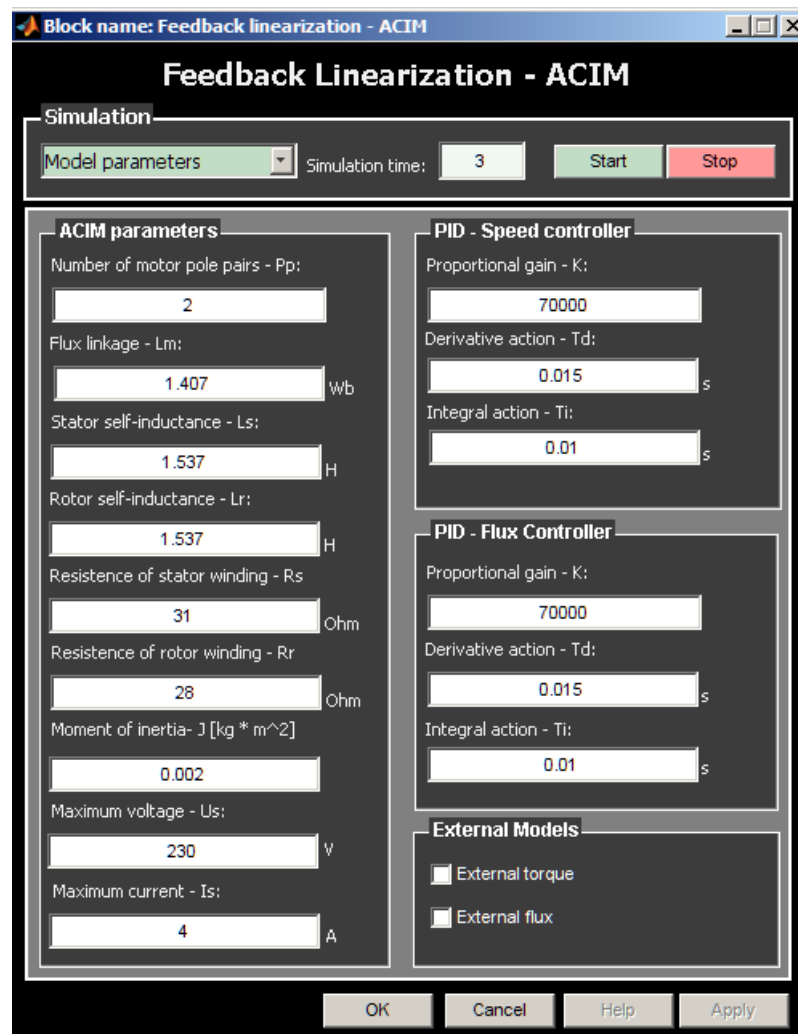
Obrázek 7.4 zobrazuje GUI použité u algoritmu přímého řízení momentu jak PMSM tak ASM. Samotný algoritmus DTC je velice jednoduchý. Nutné je nastavení dvou parametrů motoru k vypočítání napěťového modelu rotorového toku a momentu. Jeden otáčkový PI regulátoru. Nastavení maximálního proudu také nechybí. Blok umožňuje použití externího statorového toku a momentu. Po zaškrtnutí příslušného políčka („External stator flux“ nebo „External torque“) a potvrzení volby tlačítkem „Apply“ nebo „OK“ se objeví požadovaný vstup na bloku v Simulinku.



Obrázek 7.4: Přímé řízení momentu – PMSM, ASM

7.6.4 Zpětnovazební linearizace ASM - GUI

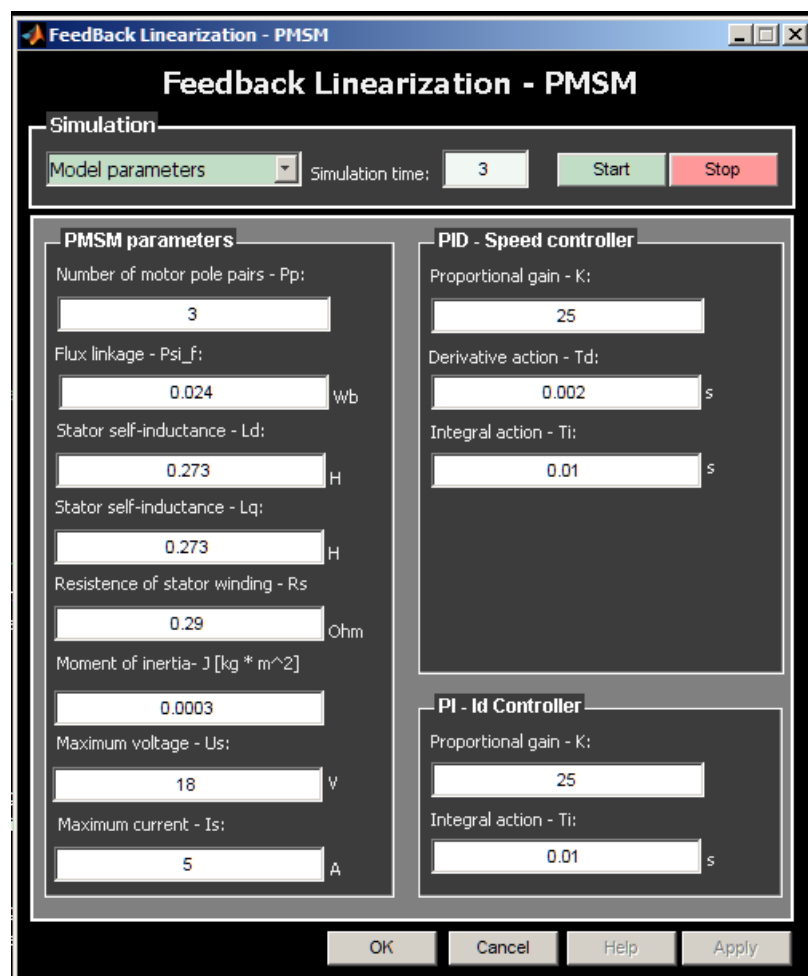
GUI na obrázku 7.5 odpovídá bloku zpětnovazební linearizace ASM. K výpočtu linearizace je nutné zadat parametry motoru nutné k řešení modelu rotorového toku. Dále je nutné nastavit konstanty dvou PID regulátoru a maximální statorové napětí. Nastavení maximálního proudu je zde pro budoucí implementaci proudového omezení. Tento blok umožňuje připojení externího modelu rotorového toku a momentu jako v kapitole 7.6.3



Obrázek 7.5: Zpětnovazební linearizace ASM - GUI

7.6.5 Zpětnovazební linearizace PMSM - GUI

Obrázek 7.6 popisuje poslední blok knihovny, jedná se o algoritmus zpětnovazební linearizace PMSM. Obsahuje nastavení konstant motoru pro výpočet linearizace, nastavení regulátoru otáček a regulátoru proudu i_{sd} . Nechybí ani nastavení maximálního napětí a proudového omezení popsané v kapitole 6.2.1.



Obrázek 7.6: Zpětnovazební linearizace PMSM - GUI

8. IMPLEMENTACE VYBRANÝCH ALGORITMŮ NA PROCESOR ŘADY FREESCALE 56F800E

Řízení pohonů probíhá v reálném procesu zejména pomocí signálových procesorů a programovatelných automatů. Implementace na takovéto zařízení přináší mnohá omezení a úskalí jako např. rušení, rozlišení AD/DA převodníku, nelinearity v obvodech, normalizace proměnných, omezený rozsah proměnných při výpočtech, měřeních atd. Implementované algoritmy mají v reálném procesu odlišné chování než při počítačových simulacích. Existují však prostředky, které umí reálné podmínky napodobit a rozšířit tak možnosti testování algoritmů.

Algoritmy vektorového řízení a zpětnovazební linearizace byly implementovány do procesoru MC56F8346 firmy Freescale a otestovány pomocí SFIO toolboxu pro Matlab - Simulink. Tento toolbox umí propojit vývojové desky s procesorem z rodiny 56F800E se Simulinkem pomocí sériové linky RS232. Je možné tak pohodlně otestovat řídicí algoritmy metodou „processor in the loop“. Testování spočívá v implementaci řídicího algoritmu do procesoru a následného propojení pomocí knihovnicí bloků SFIO toolboxu s modelem vytvořeným v simulinku. Do procesoru vstupují proměnné ze Simulinku, které jsou algoritmem v procesoru zpracovány a vypočítaný akční zásah je vrácen zpět do Simulinku.

8.1 PROCESOR MC56F8346 FIRMY FREESCALE

MC56F8346 je založený na výkonném jádře rodiny procesorů 56F800E. Jedná se o 16bitové DSC (Digital Signal Controller) založené na duální (modifikované) harvardské architektuře, kombinující vysoký výpočetní výkon klasických signálových procesorů (DSP) s výkonnými periferiemi klasických mikrokontrolérů. Mezi hlavní rysy patří [14]:

- výpočetní výkon 60 MIPS pro frekvenci jádra 60 Mhz

- 128 Kb flash, SCI, SPI, PWM, 12 bit ADC, FlexCan, Quad Timer, Quadrature Decoder, COP, 62 I/O pins...
- 16x16bitová paralelní MAC
- 36bitový akumulátor obsahující rozšiřující bity
- instrukční sada podporující DSP i funkce mikrokontroléru
- JTAG/OmCE rozhraní pro ladění programu
- Výkonný kompilátor jazyka C s podporou lokálních proměnných

Pokud chceme dosáhnout, co nejlepšího řízení pohonů musíme často složité algoritmy počítat v procesoru co nejefektivněji. K tomuto jsou v použitém DSC vytvořeny intristické funkce. Tyto funkce, napsané v assembleru zajišťují velmi rychlý výpočet v procesoru. Funkce jsou schopny počítat se 16 a 32bitovými proměnnými s desetinnou čárkou uložené v typu Frac16 nebo Frac32 [14]. Tento typ umožňuje provádět výpočty proměnných s plovoucí desetinnou čárkou, jako operace s celočíselným typem (integer) což urychluje dobu výpočtu. Rozsah typu Frac16 je od -2^{15} do 2^{15} a Frac32 od -2^{31} do 2^{31} . Při použití desetinné („fractional“) aritmetiky je nutné aby vstupní hodnoty byly normované a odpovídaly rozsahu od -1 do 1. Další optimalizaci umožňuje knihovna MCLIB, která obsahuje optimálně naprogramované funkce speciálně pro řízení motorů. [15].

8.2 SFIO TOOLBOX

Tento toolbox umožňuje uživateli přenést navržený algoritmus do reálného hardwaru a otestovat jej s modelem běžícím v Simulinku. Toolbox je na PC straně reprezentován knihovnou, která po nainstalování umožní uživateli využít knihovní funkce umožňující propojení Simulinku a reálného hardware. Knihovna obsahuje tyto bloky: SFIO interface, TrigBreak, AsynBreak.

SFIO interface je hlavním funkčním blokem SFIO toolboxu, skládá se ze dvou bloků umožňující nastavení komunikace a nadefinování vstupů, výstupů bloku. Bloky TrigBreak a AsynBreak slouží k ladění algoritmu. Na straně hardwaru je komunikace s konkrétním blokem umožněna pomocí knihovny „sfio.dll“ a dvojice souborů (zdrojový a hlavičkový) obsahující obsluhu přerušování od sériové linky a

zpracování vstupů a výstupů funkce. Zdrojový a hlavičkový soubor je možná psát ručně, nebo využít funkci „*sficodegen*“ a nechat si jejich obsah pohodlně vygenerovat.

SFIO toolbox je velice intuitivní a výkonný software k nadstandardnímu testování algoritmů pomocí procesoru před samotnou implementací do reálného výrobku.

8.3 IMPLEMENTACE VEKTOROVÉHO ŘÍZENÍ PMSM

Pro algoritmus popsaný v kapitole 4.2 můžeme použít hotové funkce z MCLIB a to Clarkovy, Parkovy transformace a jejich inverzní varianty, dále bude výhodné použití PI regulátory [15]. Decoupling v MCLIB není a proto musí být naprogramován. Proměnné k výpočtu algoritmu je nutné na vstupu normalizovat a na výstupu denormalizovat. Normy pro tyto veličiny byly určeny na základě reálného přípravku, na kterém se bude testovat algoritmus zpětnovazební linearizace PMSM popsaný v kapitole 8.4. Normy pro proud, napětí, a otáčky jsou v tabulce 8.1.

I_N	U_N	Ω_N
8 A	36,3 V	418,9 rad/s

Tabulka 8.1: Normy proudu, napětí a otáček

Vstupní veličiny jsou normalizovány podle příslušných norem podle

$$X_N = \frac{x}{X_{MAX}} N, \quad (8.1)$$

kde X_N je normalizovaná vstupní proměnná x , X_{MAX} je její příslušná norma a N reprezentuje rozsah typu použitého pro výpočty v procesoru \rightarrow pro Frac16 je $N = 2^{15}$ a pro Frac32 je $N = 2^{31}$. Denormalizace proměnných se provede podle

$$x = \frac{X_N}{N} X_{MAX}. \quad (8.2)$$

Pro implementaci algoritmu byly použity následující funkce z knihovny MCLIB [15]:

```
void MCLIB_ClarkTrfm (MC_2PhSyst *pAlphaBeta, MC_3PhSyst *p_abc);
// Clarkova transformace

void MCLIB_ClarkTrfmInv (MC_3PhSyst *p_abc, MC_2PhSyst *pAlphaBeta);
// Inverzní Clarkova transformace

Frac16 MCLIB_Sin (Frac16 x);
// sinus
Frac16 MCLIB_Cos (Frac16 x);
// cosinus

void MCLIB_ParkTrfm (MC_D-qSyst *pD-Q, MC_2PhSyst
*pAlphaBeta, MC_Angle *pSinCos); // Parkova transformace

void MCLIB_ParkTrfmInv (MC_2PhSyst *pAlphaBeta, MC_D-qSyst *pD-Q,
MC_Angle *pSinCos); // Inverzní Parkova transformace

Frac16 MCLIB_ControllerPI (Frac16 desiredValue, Frac16 easuredValue,
MC_PiParams *pParams); // PI regulátor
```

Při využití výše zmíněných funkcí, zbývá k dokončení algoritmu jen implementace Decouplingu (4.?). Lineární složky u_{sdN}^{lin} a u_{sqN}^{lin} jsou vypočítány PI regulátory, jejichž normalizované konstanty se přepočítají podle [15]. Výpočet „decouplingu“ ukazuje rovnice (8.3). Při výpočtu se nesmí zapomenout, že se pracuje s normalizovanými proměnnými a musí se přidat příslušné normy.

$$\begin{aligned} u_{sdN}^{doplN} &= -(i_{sqN} \omega_{eN}) K_{1N}, \\ u_{sqN}^{doplN} &= (i_{sdN} \omega_{eN}) K_{2N} + (\omega_{eN}) K_{3N}, \end{aligned} \quad (8.3)$$

kde

$$K_{1N} = \frac{\Omega_N L_q I_N}{U_N}, \quad K_{2N} = \frac{\Omega_N L_d I_N}{U_N}, \quad K_{3N} = \frac{\psi_m \Omega_N}{U_N} \quad (8.4)$$

Výsledné konstanty by měly mít z důvodu přesnosti výpočtu v desetinné aritmetice rozsah $\pm(1;0.5)$. Pokud konstanty vyjdou výrazně menší, je vhodné je pomocí bitových posunů zvětšit a po jejich vynásobení s příslušnou proměnnou zase pomocí bitových posunů zmenšit o příslušný počet bitů. Například pro

$$K_1 = \frac{\Omega_N L_q I_N}{U_N} = \frac{418,9 * 0.000235 * 8}{36,3} = 0,0217. \quad (8.5)$$

Konstanta K_1 vyšla mnohem menší, než je stanovený rozsah, proto ji posuneme bitově doleva o n bitů, abychom dosáhli požadované hodnoty (8.5). Bitový posun doleva způsobí vynásobení konstanty 2^n .

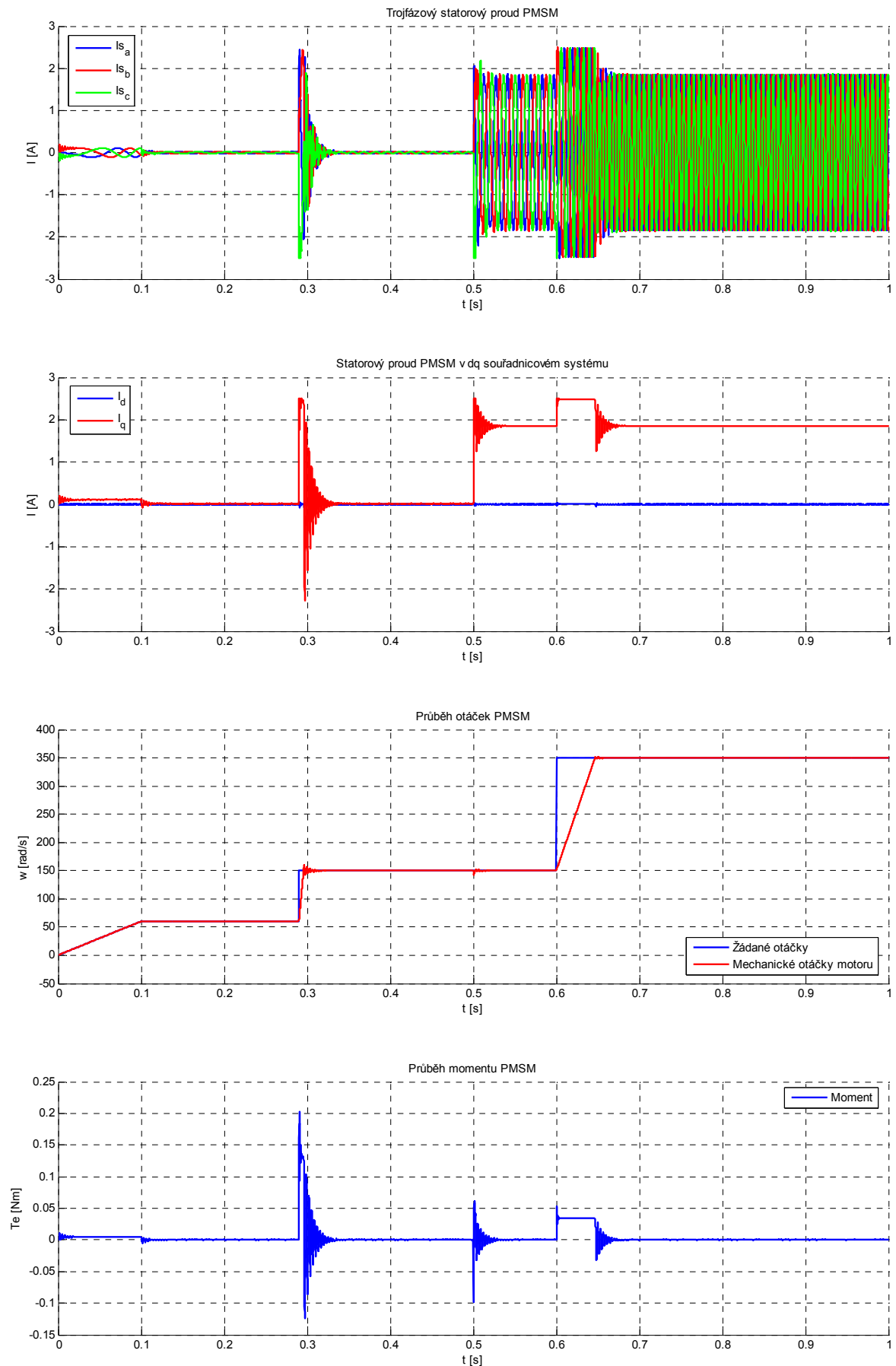
$$K_1 = 0.0217 * 2^5 = 0,6944 \quad (8.6)$$

Pokud konstanty vyjdou větší než jedna, musí se bitově posunout doprava. Bitový posun způsobí vynásobení konstanty 2^{-n} .

Při znalosti normovaných konstant se celý výpočet jednoduše provede pomocí intristických funkcí součtu, součinu a bitových posunů (mult, add, shl []). Zbytek algoritmu se dá snadno realizovat pomocí funkcí z MCLIB [15].

8.3.1 Průběh vektorového řízení PMSM na procesoru

Průběh simulace na obrázku 8.1 zachycuje vektorové řízení PMSM pomocí procesoru a se změněnými parametry motoru podle tabulky 3.2. Perioda vzorkování je 62,5 μ s. Vektorové řízení PMSM implementované na 16bitovém procesoru má prakticky stejný průběh jako při simulacích v Simulinku. Odezva na změnu otáček je velmi rychlá. Proudové omezení je nastaveno opět na hodnotu 2,5 A a proud se nad tuto hodnotu nedostane.



Obrázek 8.1: Vektorové řízení PMSM pomocí procesoru

8.4 IMPLEMENTACE ZPĚTNOVAZEBNÍ LINEARIZACE ŘÍZENÍ PMSM

Pro implementaci jsou využity opět funkce z knihovny MCLIB a to PI regulátor pro řízení proudu i_d a Clarkovy a Parkovy transformace. PSD regulátor pro řízení otáček a rovnice (6.46) realizující linearizaci motoru. Pro ověření algoritmu je použit PMSM se stejnou indukčností v příčné a podélné ose ($L_d = L_q$). Konstanta c_1 v rovnici (6.41) a (6.44) bude nulová, $\mathbf{D}(\mathbf{x})$ se zjednoduší na

$$\mathbf{D}(\mathbf{x}) = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{c_2}{L_q} \end{bmatrix}. \quad (8.7)$$

Výpočet inverzní matice $\mathbf{D}(\mathbf{x})^{-1}$ bude

$$\mathbf{D}(\mathbf{x})^{-1} = \begin{bmatrix} L_d & 0 \\ 0 & \frac{L_q}{c_2} \end{bmatrix}. \quad (8.8)$$

Rovnice (6.46) se upraví

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & \frac{L_q}{c_2} \end{bmatrix} \begin{bmatrix} -L_f^1 h_1 + v_1 \\ -L_f^2 h_2 + v_2 \end{bmatrix}, \quad (8.9)$$

$$\begin{bmatrix} u_d \\ u_q \end{bmatrix} = \begin{bmatrix} L_d(-L_f^1 h_1 + v_1) \\ \frac{L_q}{c_2}(-L_f^2 h_2 + v_2) \end{bmatrix}$$

Po přidání norem

$$\begin{bmatrix} u_{dN} \\ u_{qN} \end{bmatrix} = \begin{bmatrix} \frac{L_d}{U_N}(-L_f^1 h_1 + v_1) \\ \frac{L_q}{c_2 U_N}(-L_f^2 h_2 + v_2) \end{bmatrix} = \quad (8.10)$$

$$= \begin{bmatrix} \left(-\frac{L_d}{U_N} L_f^1 h_1 + \frac{L_d}{U_N} v_1 \right) \\ \left(-\frac{L_q}{c_2 U_N} L_f^2 h_2 + \frac{L_q}{c_2 U_N} v_2 \right) \end{bmatrix}.$$

$L_f^1 h_1$ (6.39) a $L_f^2 h_2$ (6.40) se po normalizaci změjí na

$$L_f^1 h_1(x) = i_{dN} \mathbf{A}_{1N} + i_{qN} \omega_N \mathbf{A}_{2N} \quad (8.11)$$

$$\begin{aligned} L_f^2 h_2(x) &= f_2(x) (c_2) \frac{L_q}{C_2 U_N} = \\ &= i_{qN} \mathbf{B}_{1N} + i_{dN} \omega_N \mathbf{B}_{2N} + \omega_N \mathbf{B}_{3N}, \end{aligned} \quad (8.12)$$

kde

$$\begin{aligned} \mathbf{A}_{1N} &= \frac{I_N L_d a_1}{U_N}, & \mathbf{A}_{2N} &= \frac{I_N \Omega_N L_d a_2}{U_N}, \\ \mathbf{B}_{1N} &= \frac{L_q b_1 I_N}{U_N}, & \mathbf{B}_{2N} &= \frac{b_2 L_q I_N \Omega_N}{U_N}, & \mathbf{B}_{3N} &= \frac{b_3 L_q \Omega_N}{U_N}. \end{aligned} \quad (8.13)$$

Výsledné konstanty následně při výpočtu v procesoru upravíme bitovým posunem, jako je popsáno v kapitole (8.2).

Otáčkový regulátor má za úkol řídit dva integrátory. PI regulátor z MCLIB je nahrazen PSD regulátorem. Akční zásah PSD regulátoru byl naprogramován podle

$$u(k) = K e(k) + K_I T_s \sum_{i=1}^k e(k) + \frac{K_D}{T_s} (e(k) - e(k-1)), \quad (8.14)$$

kde T_s je vzorkovací perioda, $K_I = K/T_I$ je integrační zesílení a $K_D = K * T_D$ je derivační zesílení a $e(k)$ je regulační odchylka. V rovnici (8.10) je vidět násobení obou nových řídicích vstupů (v_1, v_2) konstantami $\frac{L_q}{C_2 U_N}$, $\frac{L_d}{U_N}$. Tyto vstupy, respektive akční zásahy z regulátorů není možno vzhledem k hodnotě konstant (8.16) v 16bitové desetinné aritmetice vynásobit, proto se musí přidat ke konstantám regulátorů. Konstanty PSD regulátoru společně s normalizací a vynásobením $\frac{L_q}{C_2 U_N}$ jsou

$$u(k) = \frac{L_q K \Omega_N}{C_2 U_N} e(k) + \frac{L_q \Omega_N K_I T_s}{C_2 U_N} \sum_{i=1}^k e(k) + \frac{L_q \Omega_N K_D}{C_2 U_N T_s} (e(k) - e(k-1)). \quad (8.15)$$

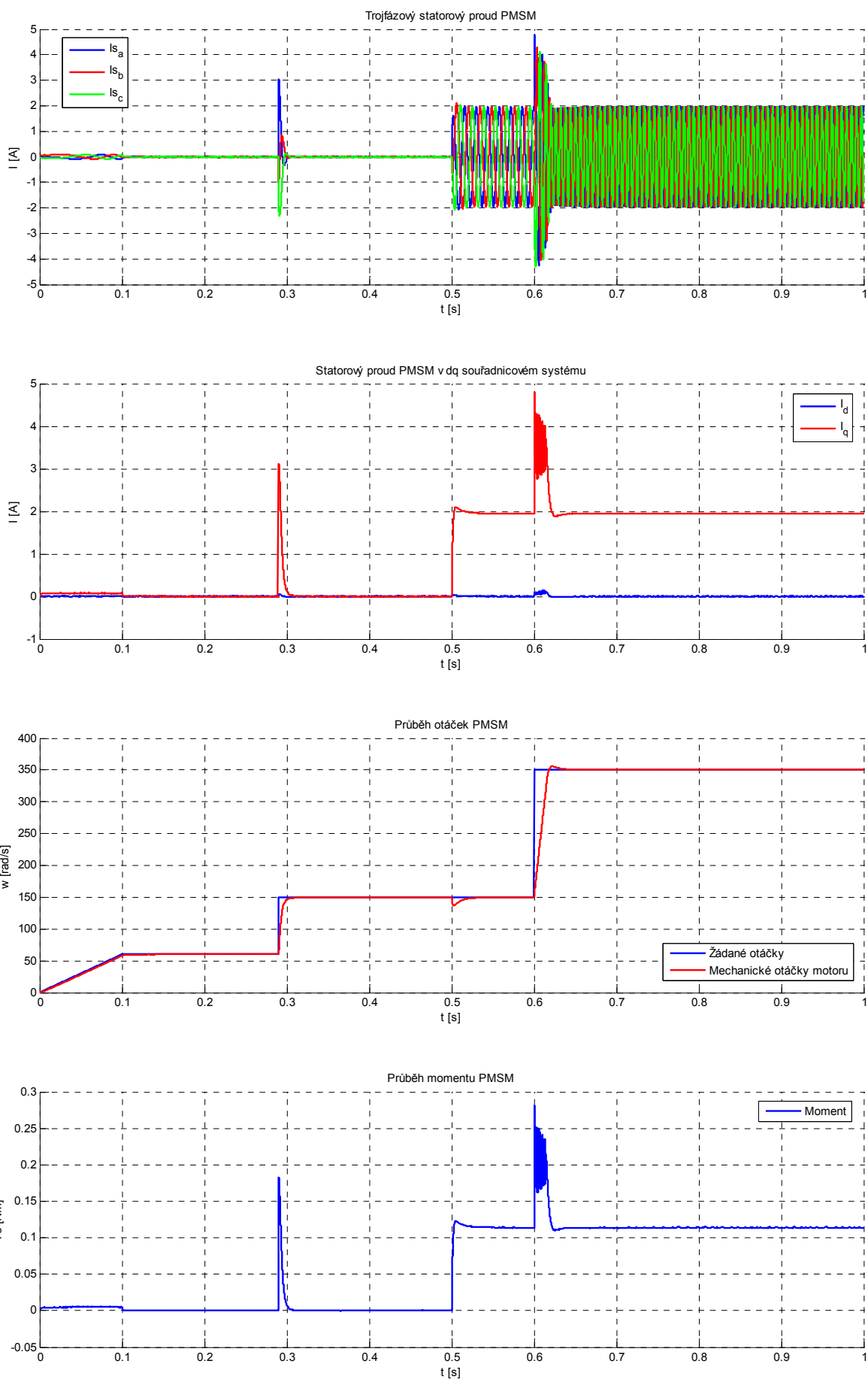
$$\begin{aligned} \frac{L_q}{C_2 U_N} &= \frac{0.000235}{621.5625 * 36,3} = 1.0415e^{-8} \\ \frac{L_d}{U_N} &= \frac{0.000235}{36,3} = 6.47e^{-6} \end{aligned} \quad (8.16)$$

Úpravou (8.15) zajistí realizovatelnost algoritmu, ale bude mít zásadní vliv na velikost konstant, zejména zesílení K , které bude muset být vzhledem k numerické hodnotě jmenovatele obrovské. Velké zesílení způsobuje velký akční zásah, který má za následek prudké změny napětí, respektive proudu a způsobuje nefunkčnost proudového omezení popsaného v kapitole 6.2.1.

8.4.1 Průběh řízení PMSM pomocí zpětnovazební linearizace

Průběh simulace na obrázku 8.2 zachycuje řízení PMSM pomocí zpětnovazební linearizace implementované na procesoru. Perioda vzorkování je $62,5 \mu\text{s}$. Průběh otáček má prakticky stejný průběh jako při simulacích v kapitole 8.3.1. Proudové omezení je nastaveno tentokrát na hodnotu 4 A. Kapitola 6.2.1 popisuje, že proudové omezení vlivem velkého akčního zásahu (rychlý nárůst proudu) není schopna zabránit počátečnímu překmitu proudu (kapitola 6.2.2). Druhý graf také zachycuje rozkmitání proudu při působení proudového omezení, které při simulacích v kapitole 6.2.2 nenastalo.

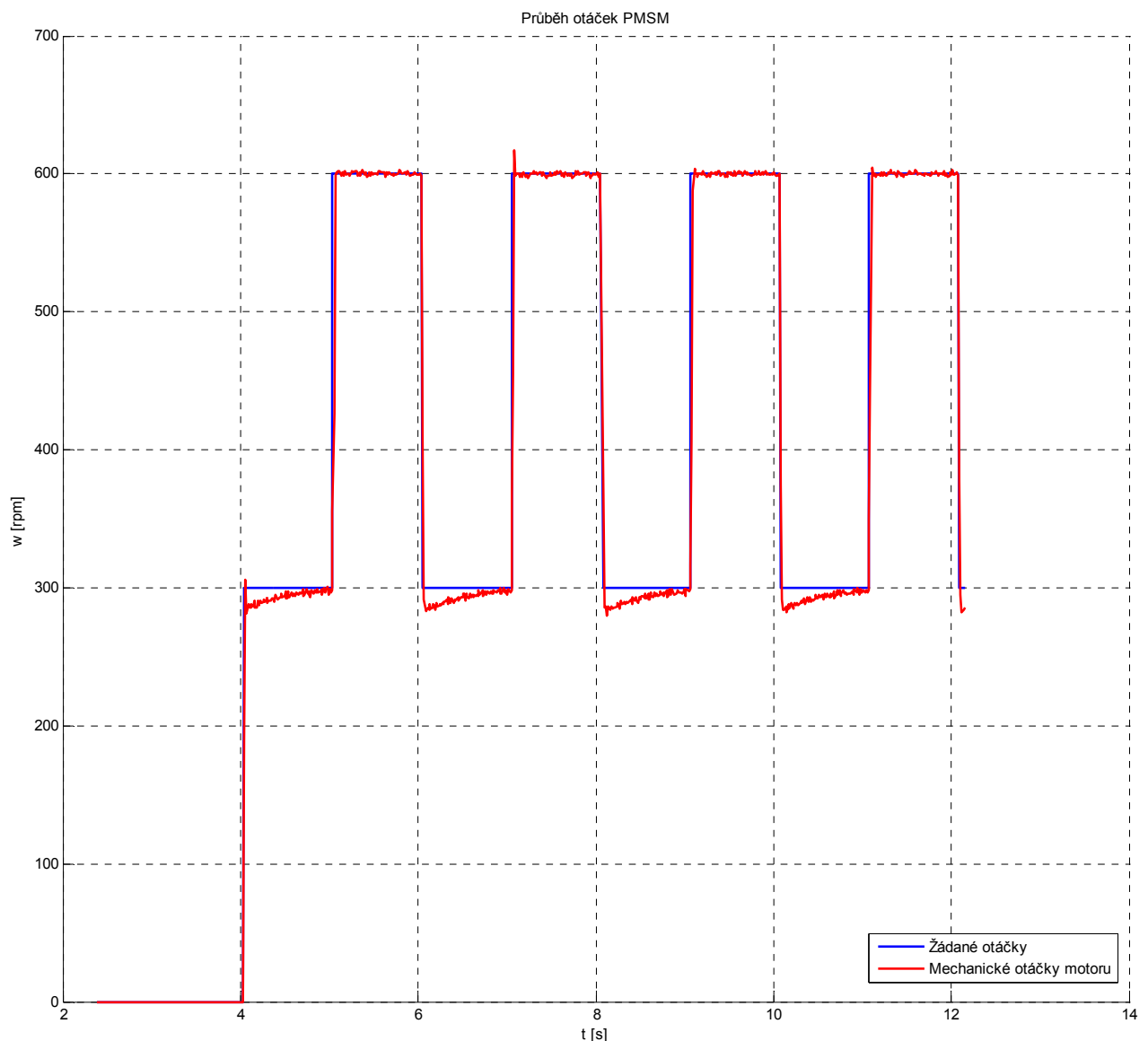
Průběh řízení pomocí zpětnovazební linearizace má srovnatelné dynamické vlastnosti s vektorovým řízením. Proudové omezení se však neukázalo jako plně funkční. Velký vliv na počáteční překmit bude mít malá časová konstanta PMSM. Při řízení pohonu s vyšší časovou konstantou by nárůst proudu nebyl tak rychlý a proudové omezení by mohlo fungovat i při velkém zesílení regulátorů vyplývající z úprav pro implementaci na procesor (8.15).



Obrázek 8.2: Řízení PMSM pomocí zpětnovazební linearizace na procesoru

8.5 ŘÍZENÍ REÁLNÉHO PMSM POMOCÍ ZPĚTNOVAZEBNÍ LINEARIZACE

Obrázek 8.3 zachycuje průběh řízení otáček reálného PMSM firmy TG Drives (kapitola 3.2). Tento průběh byl pořízen jako potvrzení funkčnosti algoritmu zpětnovazební linearizace použitého v této práci. Bohužel se nepodařilo naměřit vyšší změny žádané hodnoty než o 300 rpm z důvodu počátečního překmitu proudu, který spouštěl proudové ochrany motoru. Rychlý nárůst proudu zapříčinila velmi malá časová konstanta motoru. Při řízení motoru s vyšší časovou konstantou by byl nárůst proudu podstatně pomalejší a proudové omezení (kapitola 6.2.1) by pravděpodobně omezilo proud včas.



Obrázek 8.3: Průběh řízení reálného PMSM pomocí zpětnovazební linearizace

9. ZÁVĚR

Cílem této diplomové práce bylo vytvořit ucelenou knihovnu bloků pro simulace řídicích algoritmů asynchronních (ASM) a synchronních motorů s permanentním magnetem (PMSM). Dále vybrané řídicí algoritmy naimplementovat na procesor řady Freescale 56F800E Hybrid Controller a provést jejich simulaci metodou “processor in the loop“.

Všechny řídicí algoritmy asynchronního motoru (ASM), uváděné v této práci, prokázaly velice kvalitní dynamiku i možnost odděleného řízení magnetického toku a momentu motoru. Hlavní rozdíl spočívá zejména ve složitosti algoritmů z hlediska implementace a použitelnosti v praktických aplikacích (proudové omezení). Pokud je potřeba nižších nároků na výpočetní výkon, je vhodné použít algoritmus přímého řízení momentu (DTC). Algoritmus založený na zpětnovazební linearizaci (vstup - výstup) je velmi složitý a nenabízí jednoduše implementovatelné proudové omezení, ačkoli je jeho výkon srovnatelný s vektorovým řízením, kterým se dosáhlo nejpreciznějšího řízení ASM.

Algoritmy vektorového řízení a zpětnovazební linearizace (vstup - výstup) pro řízení synchronního motoru s permanentním magnetem (PMSM) jsou porovnány z hlediska řízení na procesoru. Hodnocení průběhů řízení vychází opět velmi vyrovnaně. Algoritmus zpětnovazební linearizace PMSM byl vyzkoušen na reálném motoru firmy TG Drives (kapitola 8.5). Při řízení reálného motoru byla implementace proudového omezení do řídicího algoritmu nezbytná (kapitola 6.2.1). Proudové omezení však nebylo schopno zabránit počátečnímu překmitu proudu a docházelo k aktivaci proudových ochran a následnému zastavení motoru. Prudký nárůst proudu má souvislost s velmi malou časovou konstantou použitého PMSM a velkým zesílením regulátoru, vycházejícího z implementace na procesoru (8.15). Použití vhodnějšího PMSM s vyšší časovou konstantou by zamezilo tak rychlému nárůstu proudu a proudové omezení by mělo zareagovat včas.

Vektorové řízení obsahuje podřízené proudové smyčky, pomocí kterých je možno proudové omezení provést efektivně a navíc je celý algoritmus jednodušší, snadněji implementovatelný na procesor (PMSM i ASM). Podřízené proudové

smyčky jsou realizovány pomocí bloku odstranění křížných vazeb „Decoupling“ (kapitola 4.1.2 a 4.2.1). Tento blok je vytvořen aplikací teorie zpětnovazební linearizace, proto lze konstatovat, že nejlepším řídicím algoritmem, uvedeným v této práci, je právě kombinace klasického vektorového řízení a zpětnovazební linearizace.

Pro využití algoritmů byla vytvořena knihovna v prostředí Matlab – Simulink (kapitola 7). Knihovna značně zpřehlední, usnadní a hlavně urychlí budoucí použití řídicích algoritmů v simulacích.

10. SEZNAM POUŽITÉ LITERATURY

- [1] Caha Z., Černý M.: Elektrické pohony, SNTL, Praha 1990, 298 s.
- [2] Šubrt, J.: Elektrické regulační pohony, VUT, Brno 1987, 180 s.
- [3] Lepka J, Stekl P.: 3-Phase AC Induction Motor Vector Control Using a 56F80x, 56F8100 or 56F8300 Device, [online]. [cit. 2009-3-25].
Dostupné z: < <http://www.freescale.com/files/product/doc/AN1930.pdf>>.
- [4] Šolc F., Václavek P., Vavřín P.: Řízení a regulace II, analýza a řízení lineárních systémů, FEKT VUT, Brno 2006, 228 s.
- [5] VÁCLAVEK, P. *Estimace rychlosti pro bezsnímačové řízení elektrických pohonů s asynchronními motory*, habilitační práce. Brno, VUTIUUM. 2006
- [6] Brandštetter, P: Střídavé regulační pohony - moderní způsoby řízení, FEI VŠB, Ostrava, 1999, 177 s.
- [7] Blaha P., Vavřín P.: Řízení a regulace 1, Základy regulace lineárních systémů – spojité a diskrétní; FEKT VUT, Brno, 212 s.
- [8] Skalický J.: Elektronické regulované pohony, FEKT, Brno 2007, 123 s.
- [9] Razím M., Štecha J.: Nelineární systémy, FEL ČVUT, Praha 1997, 201 s.
- [10] Marino R., Peresad S., Valigi P.: Adaptive Input – Output Linearizing Control of Induction Motors, IEEE Transactions on Automatic Control, February 1993, vol. 38, no. 2, s. 208 – 221.
- [11] Raumer T., Dion J. M., Dugard L., Thomas J. L., Valigi P.: Applied Nonlinear Control of an Induction Motor Using Digital Signal Processing, IEEE Transactions on Control Systems Technology, December 1994, vol. 2, no. 4, s. 327 – 355.
- [12] Rebouh S., Kaddouri A., Abdessemed R., Haddoun A.: Nonlinear Control by Input-Output Linearization Scheme for EV Permanent Magnet Synchronous Motor [online], URL: <http://ieeexplore.ieee.org>, VPPC 2007. IEEE, 10.1109/VPPC.2007.4544122, s. 185-90 [cit. 2010.5.3]
- [13] Takahashi I., Ohmori Y.: High-Performance Direct Torque Control of an Induction Motor, IEEE Transactions on Industry Applications, March/April 1989, vol. 25, no. 2, s. 257 – 624.

[14] Freescale Semiconductor, CodeWarrior™ Development Studio for Freescale™ 56800/E Digital Signal Controllers: MC56F8xxx/DSP5685x Family Targeting Manual, [online],[cit. 2010-3-12],

Dostupné z:

<http://cache.freescale.com/files/soft_dev_tools/doc/user_guide/Targeting_56800E.pdf>.

[15] Freescale Semiconductor, Motor Control Library for 56F83xx Motorola Hybrid Controllers, [online],[cit. 2010-3-12],

Dostupné z:

<http://cache.freescale.com/files/microcontrollers/doc/user_guide/56800E_MCLIB.pdf?fsrch=1>.

11. SEZNAM ZKRATEK A POUŽITÝCH SYMBOLŮ

U_s	amplituda napájecího napětí
u_a	napájecí napětí fáze a statoru
ω	elektrická úhlová rychlost
t	čas
f	frekvence
ω_s	synchronní rychlost
P_p	počet pólových dvojic
s	skluz
ω_m	mechanická rychlost
R_s	odpor statoru
R_r	odpor rotoru
L_s	indukčnost statoru
L_r	indukčnost rotoru
L_m	vzájemná indukčnost
$U_{s\ max}$	maximální statorové napětí
$I_{s\ max}$	maximální statorový proud
L_d	indukčnost v podélné ose
L_d	indukčnost v příčné ose
ψ_m	magnetický tok magnetu
J	moment setrvačnosti
θ	úhel rotoru
α - β	značení os pro Clarkovu transformaci
d - q	značení os pro Parkovu transformaci
\bar{l}_s	prostorový vektor
i_s	statorový proud
u_s	statorové napětí
ψ_{Rd}	absolutní hodnota rotorového toku
ψ_R	rotorový magnetický tok
ψ_S	statorový magnetický tok

T_e	elektrický moment motoru
T_L	zátěžový moment motoru
I_N	norma pro proud
U_N	norma napětí
Ω_N	norma otáček
X_N	normalizovaná vstupní proměnná
x	vstupní proměnná
X_{MAX}	příslušná norma veličiny
N	rozsah typu pro výpočty v procesoru při normalizaci
GUI	grafické uživatelské rozhraní (Graphical User Interface)

12. SEZNAM PŘÍLOH

Příloha č. 1: DVD s obsahem

- zdrojové soubory k simulacím na procesoru
- knihovna řídicích algoritmů pro Matlab – Simulink
- demonstrační simulace pomocí knihovny
- elektronická verze diplomové práce