

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

BEZSNÍMAČOVÉ ŘÍZENÍ POHONU SE SYNCHRONNÍM  
MOTOREM

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

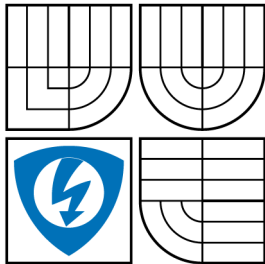
AUTOR PRÁCE  
AUTHOR

MIROSLAV GRAF

BRNO 2009



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

## BEZSNÍMAČOVÉ ŘÍZENÍ POHONU SE SYNCHRONNÍM MOTOREM

PMSM DRIVE SENSORLESS CONTROL

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

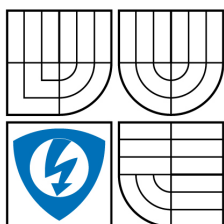
AUTOR PRÁCE  
AUTHOR

MIROSLAV GRAF

VEDOUcí PRÁCE  
SUPERVISOR

DOC. ING. PAVEL VÁCLAVEK, PH.D.

BRNO 2009



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí  
techniky

# Bakalářská práce

bakalářský studijní obor  
Automatizační a měřicí technika

**Student:** Miroslav Graf  
**Ročník:** 3

**ID:** 73039  
**Akademický rok:** 2008/2009

## NÁZEV TÉMATU:

**Bezsnímačové řízení pohonu se synchronním motorem**

## POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s modelováním a řízením synchronního motoru s permanentními magnety. V prostředí Matlab-Simulink připravte model pohonu a vektorového řízení. Navrhněte algoritmus bezsnímačového řízení založený na rozšířeném Kalmanově filtru a struktuře MRAS. Na základě posouzení výpočetní složitosti implementujte vybraný algoritmus na procesoru Freescale 56F8346 a proveďte jeho simulační ověření metodou "target in the loop".

## DOPORUČENÁ LITERATURA:

Caha, Z., Čemý, M.: Elektrické pohony  
Šubrt, J.: Elektrické regulační pohony  
další dle průběžných pokynů vedoucího práce

**Termín zadání:** 9.2.2009

**Termín odevzdání:** 1.6.2009

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

**prof. Ing. Pavel Jura, CSc.**  
*předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

## **ABSTRAKT**

Tato práce se zabývá bezsnímačovým řízením synchronních motorů s permanentními magnety na povrchu rotoru a jejich implementací na DSP procesor. Bezsímačové řízení probíhá přes rekonstruktor stavů, který je realizován jako Model Reference Adaptive System a Kalmanův filtr.

## **KLÍČOVÁ SLOVA**

synchronní motor, kaskádní řízení, bezsnímačové řízení, zpětnovazební linearizace, decoupling, Kalmanův filtr, MRAS, Model Reference Adaptive System, normalizace, DSP, Freescale

## **ABSTRACT**

This thesis concerns about sensorless control of surface permanent-magnet synchronous motor and their implementation to a DSP processor. Sensorless control is executed by a stage reconstructor which is programmed as Model Reference Adaptive System and Kalman Filter.

## **KEYWORDS**

synchronous drive, SPMSM, concatenated control, sensorless control, feedback linearization, decoupling, Kalman Filter, MRAS, Model Reference Adaptive System, normalization, DSP, Freescale

GRAF, M. *Bezsímačové řízení pohonu se synchronním motorem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 44 s. Vedoucí bakalářské práce doc. Ing. Pavel Václavek, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Bezsímačové řízení pohonu se synchronním motorem“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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 .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce mé bakalářské práce panu doc. Ing. Pavlu Václavkovi PhD. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych poděkoval spolubydlícím Bc. Janu Kubíčkoví a Bc. Karlu Piwkoví za programátorskou podporu a formulaci poděkování.

V Brně dne .....

.....

(podpis autora)

## OBSAH

<b>1</b>	<b>Modelování a vektorové řízení synchronního motoru</b>	<b>10</b>
1.1	Souřadnicový systém . . . . .	10
1.1.1	Souřadnicové systémy používané u motorů . . . . .	10
1.1.2	Převody z trojfázové soustavy na dvoufázovou . . . . .	11
1.2	Synchronní motor (SPMSM) . . . . .	11
1.2.1	Model synchronního motoru v rotorových souřadnicích . . . . .	11
1.2.2	Zapojení motoru do trojfázové soustavy . . . . .	12
1.2.3	Model synchronního motoru ve statorových souřadnicích . . . . .	12
1.3	Snímačové řízení . . . . .	14
1.3.1	Vektorové řízení . . . . .	14
1.3.2	Zpětnovazební linearizace - decoupling . . . . .	14
1.3.3	Řízení proudu . . . . .	16
1.3.4	Řízení otáček . . . . .	16
1.3.5	Wind-up jevy vznikající v regulační smyčce . . . . .	18
<b>2</b>	<b>Bezsímačové řízení</b>	<b>20</b>
2.1	Základní princip . . . . .	20
2.1.1	Pozorovatelnost a rekonstruovatelnost synchronního motoru . . . . .	20
2.1.2	Rekonstruktory stavů . . . . .	20
2.1.3	Diskretizace systému . . . . .	21
2.1.4	Zařazení rekonstruktoru do zpětné vazby . . . . .	21
2.2	Model Reference Adaptive System . . . . .	23
2.2.1	Adaptační a referenční model . . . . .	23
2.2.2	Výpočet chyby . . . . .	24
2.2.3	Odhad polohy a otáček . . . . .	24
2.3	Kalmanův filtr . . . . .	27
2.3.1	Predikce . . . . .	27
2.3.2	Korekce . . . . .	28
2.3.3	Nastavení Kalmanova filtru . . . . .	28
<b>3</b>	<b>Implementace řídicího algoritmu</b>	<b>31</b>
3.1	Freescale 56F8346 . . . . .	31
3.1.1	Normalizace veličin . . . . .	31
3.1.2	Počítání koeficientů regulátorů . . . . .	32
3.1.3	MRAS v mikroprocesoru . . . . .	32
3.1.4	Normalizace matic Kalmanova filtru . . . . .	33
3.1.5	Optimalizace Kalmanova filtru . . . . .	34

3.2	Simulace v prostředí Matlab Simulink . . . . .	36
<b>4</b>	<b>Výsledky</b>	<b>40</b>
<b>5</b>	<b>Závěr</b>	<b>42</b>
	<b>Reference</b>	<b>43</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>44</b>

## SEZNAM OBRÁZKŮ

1.1	Model synchronního motoru v trojfázové soustavě . . . . .	12
1.2	Zapojení decouplingu . . . . .	15
1.3	Zapojení regulátorů do linearizovaného modelu . . . . .	17
1.4	Odezva na jednotkový skok otáček . . . . .	18
2.1	Zapojení rekonstruktoru do soustavy . . . . .	22
2.2	Schéma MRAS . . . . .	25
2.3	Rekonstrukce otáček pomocí MRAS . . . . .	26
2.4	Rekonstrukce polohy pomocí MRAS . . . . .	26
2.5	Rekonstrukce otáček pomocí Kalmanova filtru . . . . .	29
2.6	Rekonstrukce polohy pomocí Kalmanova filtru . . . . .	30
3.1	Zapojení mikroprocesoru do regulační smyčky . . . . .	36
3.2	Odezva na skok otáček(MRAS)-otáčky . . . . .	37
3.3	Odezva na skok otáček(MRAS)-poloha . . . . .	37
3.4	Odezva na skok otáček(Kalmanův filtr)-otáčky . . . . .	38
3.5	Odezva na skok otáček(Kalmanův filtr)-poloha . . . . .	38
3.6	Odezva na skok otáček při zašumění vstupů(Kalmanův filtr)-otáčky . . .	39
3.7	Odezva na skok otáček při zašumění vstupů(Kalmanův filtr)-poloha . . .	39

## ÚVOD

Synchronní motory mají čím dál větší uplatnění v technické praxi a postupně nahrazují stejnosměrné motory díky svým lepším technickým vlastnostem. Na řízení stejnosměrný motorů je ale nutné znát polohu a úhlovou rychlost rotoru. Zvláště u malých pohonů značnou část ceny snímače mechanických veličin. Použití snímače není jediný způsob jak zjistit polohu a úhlovou rychlost rotoru. Pokud použijeme stavový rekonstruktor, jsme také schopni zjistit polohu a úhlovou rychlost rotoru.

Na úvod této práce se seznámíme se souřadnicovými systémy motorů a vzájemným převodem mezi nimi. V další kapitole následuje stavový popis synchronního motoru a vytvoření modelu motoru. Pokud máme vytvořený model, můžeme se zabývat řízením, nejprve snímačovým řízením. Ukážeme si linearizaci synchronního motoru a kaskádní regulaci.

Hlavní náplní této práce je bezsnímačové řízení synchronních motorů s permanentními magnety na povrchu. V práci je popsáno vytvoření rekonstruktoru stavů, Kalmanova filtru a Model Reference Adaptive System, a jejich porovnání. Dále se práce zabývá implementací obou algoritmů na mikroprocesor Freescale 56F8346, který neumí počítat s pohyblivou řádovou čárkou.

# 1 MODELOVÁNÍ A VEKTOROVÉ ŘÍZENÍ SYNCHRONNÍHO MOTORU

Tato kapitola se zabývá modelováním synchronního motoru s permanentními magnety na povrchu rotoru a snímačovým řízením.

## 1.1 Souřadnicový systém

Pro porozumění motorů obecně je nutné se zavést několik souřadnicových systému a vzájemný převod mezi nimi. Nejdůležitější je rotorový a statorový souřadnicový systém.

### 1.1.1 Souřadnicové systémy používané u motorů

#### Statorový souřadnicový systém

Statorový souřadnicový systém je pevně spjat na stator a vektor magnetického pole se otáčí uvnitř úhlovou rychlostí  $\omega$ . Podle názvu os se někdy také značí  $\alpha - \beta$ . Pokud si představíme statorový souřadnicový systém jako imaginární plochu,  $\alpha$  má význam reálné osy a  $\beta$  má význam imaginární osy.

#### Rotorový souřadnicový systém

Souřadnicový systém pevně vázaný na rotor. V tomto souřadnicovém systému jsou dvě složky: složka  $d$  a složka  $q$ . Složka  $d$  směřuje ve směru rotoru a způsobuje magnetické ztráty, zatímco složka  $q$  směřuje kolmo na rotor a vytváří elektromagnetickou sílu a tím i moment rotoru.

#### Převody mezi rotorovým a statorovým souřadnicovým systémem

Rotorový souřadnicový systém je otočen od statorového toku o natočení rotoru  $\varphi$  a převod do rotorových souřadnic je pevně dán vztahem:

$$\begin{aligned}i_d &= i_\alpha \cos\varphi + i_\beta \sin\varphi \\i_q &= i_\beta \cos\varphi - i_\alpha \sin\varphi\end{aligned}\tag{1.1}$$

a zpětně do  $\alpha - \beta$  souřadnic :

$$\begin{aligned}i_\alpha &= i_d \cos\varphi - i_q \sin\varphi \\i_\beta &= i_q \cos\varphi + i_d \sin\varphi\end{aligned}\tag{1.2}$$

### 1.1.2 Převody z trojfázové soustavy na dvoufázovou

Abychom mohli pracovat s motorem ve dvoufázové soustavě, musíme ji nejdříve převést z trojfázové. Trojfázová soustava má pro matematické operace jeden zbytečný rozměr navíc a tím by nám zbytečně komplikovala výpočty.

Převod dvoufázové soustavy na trojfázovou:

$$\begin{aligned} i_U &= i_\alpha \\ i_V &= -\frac{1}{2}i_\alpha + \frac{\sqrt{3}}{2}i_\beta \\ i_W &= -\frac{1}{2}i_\alpha - \frac{\sqrt{3}}{2}i_\beta \end{aligned} \quad (1.3)$$

Převod trojfázové soustavy na dvoufázovou:

$$\begin{aligned} i_\alpha &= \frac{2}{3}i_U - \frac{1}{3}i_V - \frac{1}{2}i_W \\ i_\beta &= \frac{1}{\sqrt{3}}i_V - \frac{1}{\sqrt{3}}i_W \end{aligned} \quad (1.4)$$

## 1.2 Synchronní motor (SPMSM)

Synchronní motory se narozdíl od stejnosměrným a asynchronním motorům lépe přizpůsobují zátěžovému momentu. Rotor je od vektoru statorového magnetického toku posunut o zátěžový úhel  $\beta$ . Pokud zátěžový úhel překročí maximální úhel  $\beta_{max}$ , motor vyjde ze synchronního režimu a jeho otáčky klesnou.

Budicí vinutí trojfázového synchronního motoru může být realizováno mnoha způsoby. My se zaměříme na motory s permanentními magnety na povrchu rotoru. Mají na rozdíl od synchronních motorů s buzením konstantní magnetický tok vyvolaný permanentním magnetem na povrchu. Od motoru s permanentními magnety uvnitř rotoru se liší stejnými velikostmi indukčností v rotorových souřadnicích v obou osách.

### 1.2.1 Model synchronního motoru v rotorových souřadnicích

Pro jednodušší matematické vyjádření dějů v synchronním motoru použijeme model v rotorových souřadnicích[1].

Stavový model synchronního motoru s permanentními magnety na povrchu rotoru je dán následujícími rovnicemi:

$$\begin{aligned}\frac{di_d}{dt} &= \frac{u_d}{L} + \omega i_q - \frac{R}{L} i_d \\ \frac{di_q}{dt} &= \frac{u_q}{L} - \omega i_d - \frac{R}{L} i_q - \frac{K_E}{L} \omega \\ \frac{d\omega}{dt} &= \frac{3}{2J} P_p^2 K_E i_q \\ \frac{d\varphi}{dt} &= \omega\end{aligned}\quad (1.5)$$

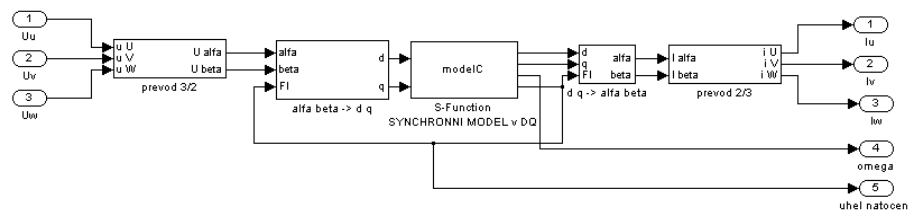
kde  $K_E$  je magnetická konstanta motoru a  $\omega$  jsou elektrické otáčky motoru. Převod mezi elektrickými a mechanickými otáčkami je dán vztahem:

$$\omega_m = \frac{\omega_{el}}{P_p} \quad (1.6)$$

Z diferenciálních rovnic je přímo vidět, že moment vůbec nezávisí na d-složce proudu a je přímo úměrný q-složce proudu.

## 1.2.2 Zapojení motoru do trojfázové soustavy

Abychom vytvořili model synchronního motoru v trojfázové soustavě, musíme ho nejprve převést z d – q souřadnic. Proto před motor předřadíme blok s převodem z trojfázové soustavy na dvofázovou a následně převod ze statorových souřadnic na rotorové. Zapojení bude vypadat jako na obrázku:



Obrázek 1.1: Model synchronního motoru v trojfázové soustavě

## 1.2.3 Model synchronního motoru ve statorových souřadnicích

Abychom mohli modelovat synchronní motor v rotorových souřadnicích, musíme znát přesnou polohu rotoru. Poloha rotoru je ale jedna z rekonstruovaných stavů a proto je modelování náchylné na chyby v rekonstrukci. K modelování ve statorových souřadnicích polohu rotoru znát nemusíme a proto je tento model odolnější proti chybám při rekonstrukci otáček a polohy rotoru.

Vydeme z rovnic 1.2 a rovnice derivujeme podle času:

$$\frac{di_{\alpha}}{dt} = \frac{di_d}{dt} \cos\varphi + i_d (-\sin\varphi) \omega - \frac{di_q}{dt} \sin\varphi - i_q (\cos\varphi) \omega \quad (1.7)$$

$\frac{di_d}{dt}$  a  $\frac{di_q}{dt}$  známe z diferenciálních rovnic 1.5.

$$\begin{aligned} \frac{di_\alpha}{dt} = & \left( \frac{u_d}{L} + \omega i_q - \frac{R}{L} i_d \right) \cos\varphi + i_d (-\sin\varphi) \omega \\ & - \left( \frac{u_q}{L} - \omega i_d - \frac{R}{L} i_q - \frac{K_E}{L} \omega \right) \sin\varphi - i_q (\cos\varphi) \omega \end{aligned} \quad (1.8)$$

$i_d$  a  $i_q$  dosadíme z rovnice 1.1

$$\begin{aligned} \frac{di_\alpha}{dt} = & \left[ \frac{u_\alpha \cos\varphi + u_\beta \sin\varphi}{L} + \omega (i_\beta \cos\varphi - i_\alpha \sin\varphi) - \frac{R}{L} (i_\alpha \cos\varphi + i_\beta \sin\varphi) \right] \cos\varphi \\ & - (i_\alpha \cos\varphi + i_\beta \sin\varphi) \omega \sin\varphi \\ & - \left[ \frac{u_\beta \cos\varphi - u_\alpha \sin\varphi}{L} - \omega (i_\alpha \cos\varphi + i_\beta \sin\varphi) - \frac{R}{L} (i_\beta \cos\varphi - i_\alpha \sin\varphi) - \frac{K_E}{L} \omega \right] \sin\varphi \\ & - (i_\beta \cos\varphi - i_\alpha \sin\varphi) \omega \cos\varphi \end{aligned} \quad (1.9)$$

To stejné uděláme i pro  $\beta$  složku, výrazy upravíme a dostaneme se k novým diferenciálním rovnicím:

$$\begin{aligned} \frac{di_\alpha}{dt} &= \frac{-Ri_\alpha + K_E \omega \sin\varphi + u_\alpha}{L} \\ \frac{di_\beta}{dt} &= \frac{-Ri_\beta - K_E \omega \cos\varphi + u_\beta}{L} \\ \frac{d\omega}{dt} &= \frac{3}{2J} p_p^2 K_E (i_\beta \cos\varphi - i_\alpha \sin\varphi) \\ \frac{d\varphi}{dt} &= \omega \end{aligned} \quad (1.10)$$

## 1.3 Snímačové řízení

Dříve než budeme motor řídit bezsnímačově, podíváme se na snímačové řízení motoru.

### 1.3.1 Vektorové řízení

Abychom mohli odděleně řídit otáčky a magnetický tok, řídí se synchronní motory vektorově. To znamená, že řízení probíhá v rotorový souřadnicích a reguluje se zvlášť  $d$  a  $q$ -složka proudu.  $d$ -složka proudu je pro nás nežádoucí, protože zvětšuje magnetický tok a tím i ztráty v rotoru. Proto se  $d$ -složka reguluje na nulovou hodnotu.

Naproti tomu  $q$ -složka proudu nám ovlivňuje moment a tím i otáčky. Pro řízení  $q$ -složky proudu se používá kaskádní regulace. Jako první se zapojuje smyčka pro regulaci proudu a poté smyčka na regulaci otáček.

### 1.3.2 Zpětnovazební linearizace - decoupling

Možností řízení nelineárním systémem je mnoho. Můžeme použít metody řízení pro nelineární systémy nebo systém linearizovat. Jako řešení se nabízí zpětnovazební linearizace. Ta udělá z nelineárního systému lineární. Aby byla zpětnovazební linearizace funkční, musíme ale znát stavy systému.

Decoupling je druh zpětnovazební linearizace. Pokud se podíváme na diferenciální rovnice synchronního motoru v  $d-q$  souřadnicích 1.5,  $d$ -složka proudu a  $q$ -složka proudu se navzájem ovlivňují. Decoupling zabráňuje vzájemnému ovlivňování regulátorů.

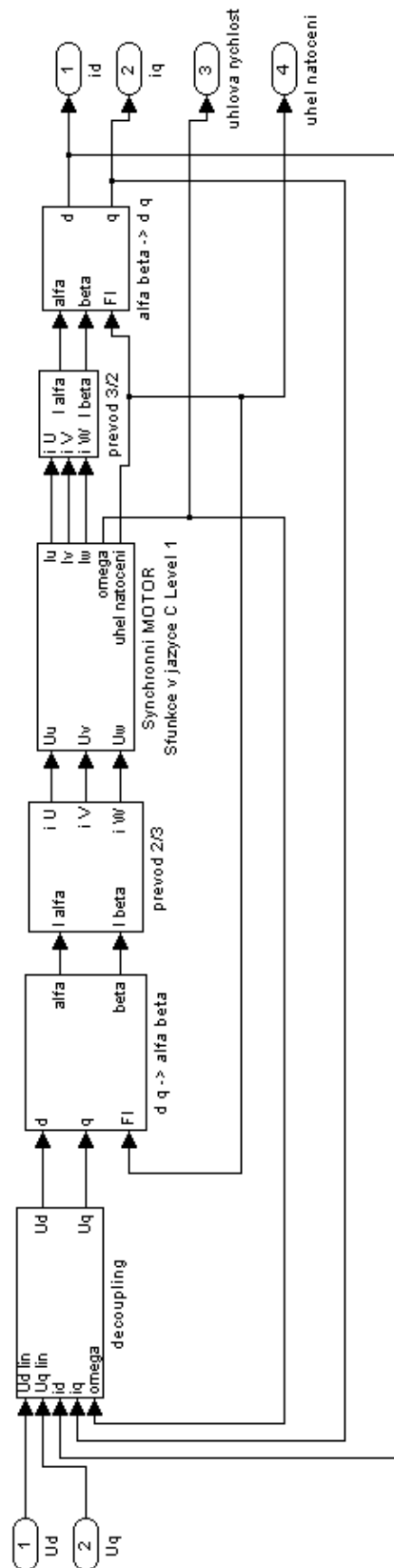
V rovnici najdeme takové  $u_{dlin}$  aby platilo  $u_d = u_{dlin} + u_{ddec}$ ,  $u_{dlin}$  bude lineární vstup do našeho linearizovaného modelu a  $u_{ddec}$  bude doplněk, aby platila zadaná rovnice.

Vzorce pro decoupling jsou:

$$\begin{aligned} u_d &= u_{dlin} - L\omega i_q \\ u_q &= u_{qlin} + L\omega i_d + \omega K_E \end{aligned} \quad (1.11)$$

A tím se dostaneme k novému linearizovanému systému:

$$\begin{aligned} \frac{di_d}{dt} &= \frac{1}{L}u_{dlin} - \frac{R}{L}i_d \\ \frac{di_q}{dt} &= \frac{1}{L}u_{qlin} - \frac{R}{L}i_q \\ \frac{d\omega}{dt} &= \frac{3}{2J}p_p^2 K_E i_q \\ \frac{d\varphi}{dt} &= \omega \end{aligned} \quad (1.12)$$



Obrázek 1.2: Zapojení decouplingu

### 1.3.3 Řízení proudu

Jak jsme si uvedli kapitole 1.3.1, d-složka proudu je pro nás nežádoucí, protože nám způsobuje jenom elektromagnetické ztráty a nemá žádný vliv na otáčky a moment motoru. Proto tuto složku regulujeme na nulu. Nulová hodnota d-složky proudu je také jednou s podmínek vektorového řízení. Naopak q-složka proudu je pro nás žádoucí a proto na smyčku s řízením q-složky proudu navazuje smyčka s řízením otáček.

Po provedení decouplingu se nám zruší křížená vazba a tudíž d-složka proudu již neovlivňuje q-složku proudu a jednotlivé regulace se navzájem neovlivňují. Aplikuje Laplaceovu transformaci na linearizované rovnice:

$$\begin{aligned} pI_d &= \frac{1}{L} U_{dlin} - \frac{R}{L} I_d \\ pI_q &= \frac{1}{L} U_{qlin} - \frac{R}{L} I_q \\ p\Omega &= \frac{3}{2J} p_p^2 K_E I_q \\ p\Phi &= \Omega \end{aligned} \quad (1.13)$$

Odtud již není složité odvodit přenos soustavy:

$$F_s(p) = \frac{\frac{1}{R}}{\frac{L}{R}p + 1} \quad (1.14)$$

Nyní hledáme regulátor pro tuto soustavu. Postačí nám PI regulátor a odtud odvodíme přenos řízení proudové smyčky:

$$F_{Ri}(p) = K_{Ri} \frac{Tp + 1}{p} \quad (1.15a)$$

$$T = \frac{L}{R} \quad (1.15b)$$

$$F_{Oi} = \frac{K_{Ri}}{R} \frac{1}{p} \quad (1.15c)$$

$$F_{Wi} = \frac{1}{\frac{R}{K_{Ri}}p + 1} \quad (1.15d)$$

Řízení q-složky proudu je zcela ekvivalentní s řízením d-složky proudu. Rozdílným nastavením zesílení regulátorů se dá nastavit různě rychlá odezva na jednotkový skok.

### 1.3.4 Řízení otáček

Řízení otáček je nadřazená smyčka nad proudovou smyčkou. Známe přenos řízení q-složky proudu (vzorec 1.15d) a známe přenos mezi proudem a otáčkami (viz. linearizované diferenciální rovnice 1.13) a odtud spočteme přenos soustavy:

$$F_s(p) = \frac{3}{2J} p_p^2 K_E \frac{1}{p \left( \frac{R}{K_{Ri}} p + 1 \right)} \quad (1.16)$$

Pro regulaci otáček použijeme PID regulátor a odtud odvodíme přenos otevřené smyčky:

$$F_{Rw}(p) = K_{Rw} \frac{(T_1 p + 1)(T_2 p + 1)}{p(\varepsilon p + 1)} \quad (1.17a)$$

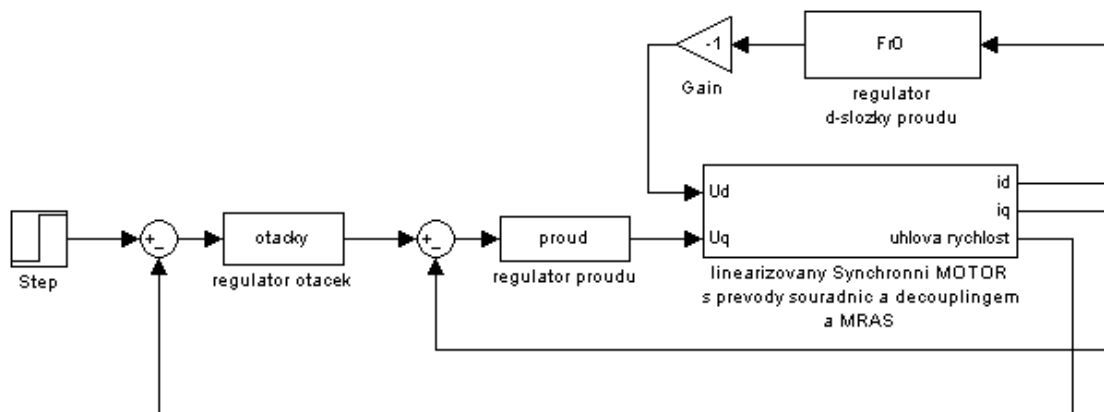
$$T_1 = \frac{R}{K_{Ri}} \quad (1.17b)$$

$$F_{Ow} = K_{Rw} \frac{3}{2J} p_p^2 K_E \frac{(T_2 p + 1)}{p^2 (\varepsilon p + 1)} \quad (1.17c)$$

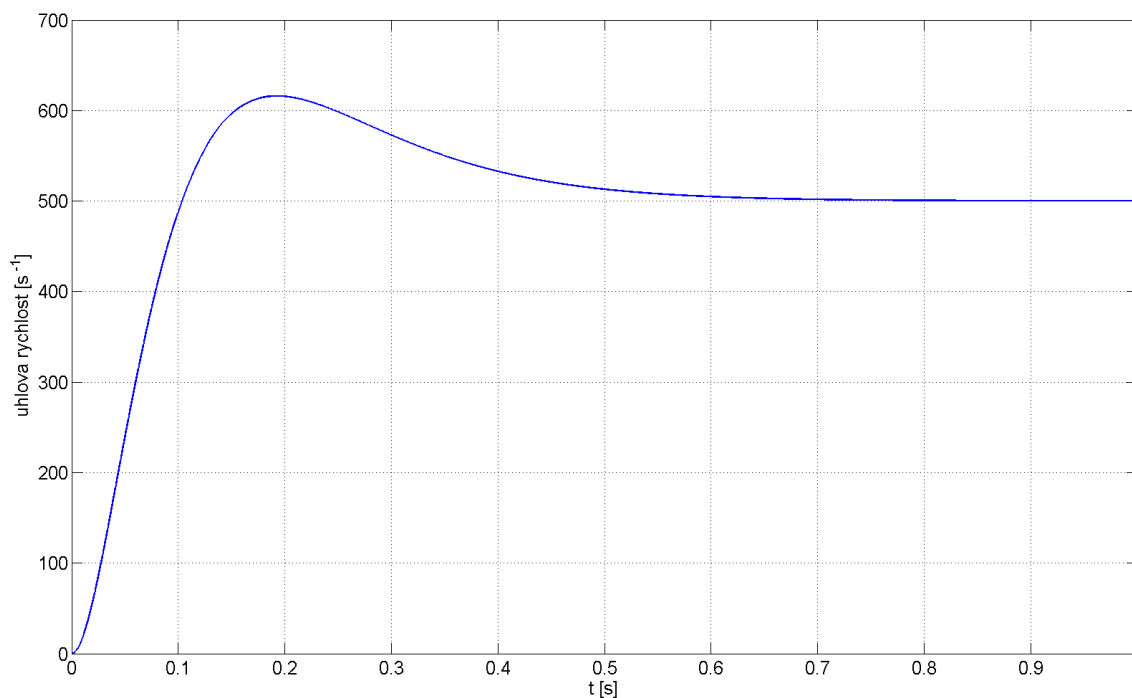
Nyní máme přenos otevřené smyčky a z otevřené smyčky nalezneme hodnoty  $T_2$  a  $K_{Rw}$ . Abychom měli co nejrychlejší regulaci, zvolíme  $T_2$  podle požadované rychlosti regulace a odtud spočítáme  $\varepsilon$  jako jednu desetinu  $T_2$  a tím zajistíme sklon -20dB po jednu dekádu. Nyní nastavíme zesílení tak, ať přenos v bodě  $T_2$  je 10dB. Dojdeme k následujícímu závěru:

$$K_{Rw} = 10^{0,5 - \log K_{sw} - 2 \log T_2} \quad (1.18)$$

$$\text{kde } K_{sw} = \frac{3}{2J} p_p^2 K_E$$



Obrázek 1.3: Zapojení regulátorů do linearizovaného modelu



Obrázek 1.4: Odezva na jednotkový skok otáček ( $\omega_{el} = 500 \text{ ot min}^{-1}$ )

### 1.3.5 Wind-up jevy vznikající v regulační smyčce

Vlastní Wind-up jevy jsou ošetřeny v každém regulátoru zvlášť, ale ve vektorovém řízení se používá regulace s pomocnou regulovanou veličinou a mohou vzniknout přesycení ve vnitřních smyčkách, které zhorší stabilitu regulační smyčky. V soustavě se synchronním motorem existují tři takovéto místa - decoupling, regulátor proudu a převod napětí na výstup z mikroprocesoru.

Regulátory pracují s linearizovaným napětím. Decoupling k tomu přičte decouplingové napětí (viz. rovnice 1.11) a právě při této operaci může dojít k přetečení hodnoty nad stanovenou normu. Tuto informaci je nutno předat regulátor proudu, aby se nesnažil dále zvyšovat akční zásah.

Další přesycení může vzniknout na regulátoru proudu. Podobně jako u decouplingu, může dojít k nasycení regulátoru proudu, akční zásah na motor nebude adekvátní tomu, co požadoval regulátor otáček a dojde opět k Wind-up jevu.

Poslední místo je při převodu napětí na výstup z PWM (pulsní šířkový modulátor). Na výstupu z decouplingu je rotorové napětí v d-q souřadnicích. Toto je nutno převést do  $\alpha - \beta$  souřadnic a následně na napětí na jednotlivých fázích. Takto vypočítané napětí může přesáhnout možnosti D/A převodníku a dojde opět k přesycení.

Způsoby, jakým lze Wind-up jevům omezit je mnoho. Ideální je kontrolovat vnitřní stavové proměnné regulátorů a pokud dojde k nasycení, předat tuto informaci ostatním

členům v regulační smyčce. Další jednodušší možnost je zavést omezení maximálních otáček regulátoru a tím se Wind-up jevům úplně vyhnout.

## 2 BEZSNÍMAČOVÉ ŘÍZENÍ

V této kapitole se zaměříme na bezsnímačové řízení. Odebereme snímač polohy a snímač otáček a pokusíme se pomocí rekonstruktoru tyto stavy odhadnout a řídit.

### 2.1 Základní princip

Nejdříve si rozebereme principy bezsnímačového řízení a převedeme model do diskrétní podoby.

#### 2.1.1 Pozorovatelnost a rekonstruovatelnost synchronního motoru

K tomu, abychom mohli rekonstruovat stavy, musí být systém pozorovatelný a rekonstruovatelný[3]. Pozorovatelnost znamená, že jsme schopni zjistit počátek odkud systém vyšel, a rekonstruovatelnost, že jsme schopni zjistit aktuální stav.

Důkaz pozorovatelnosti a rekonstruovatelnosti sahá nad rámec této práce[7], předpokládáme, že systém je za nenulových otáček pozorovatelný a rekonstruovatelný.

#### 2.1.2 Rekonstruktory stavů

Pokud nejsme schopni některé stavy měřit, musíme je rekonstruovat. K tomu slouží rekonstruktory stavů. Tato práce se zabývá dvěma rekonstruktory stavů. Jedním z nich je Model Reference Adaptive System(MRAS) a další je Kalmanův filtr.

MRAS funguje na principu dvou modelů, z nichž jeden je referenční a druhý je adaptační. Referenční model není závislý na rekonstruované veličině a adaptační model naopak je. Provode se porovnání obou modelů a výpočet chyby. Chyba slouží jako vstup do regulátoru, který upravuje adaptační model a tím zajišťuje rekonstrukci.

Kalmanův filtr je druh identického pozorovatele a má v sobě model systému a porovnává výstup z modelu a systému. Je velmi obecně použitelný a rychle konverguje k přesné rekonstrukci. Abychom mohli rovnou napsat diskrétního identického pozorovatele, musíme napsat diskrétní model soustavy. V kapitole 1.2 jsme si uvedli modely v rotorových a statorových souřadnicích. Pokud bychom napsali model v rotorových souřadnicích, nebyli bychom plně schopni rekonstruovat stavy systému, protože k získání rotorových souřadnic je nutné znát úhel natočení motoru, což je stav, který neznáme - pouze rekonstruujeme. Rekonstruktor stavů není nekonečně přesný a proto je volba  $d - q$  modelu nevhodná.

Naproti tomu model ve statorových souřadnicích nepotřebuje znát úhel natočení rotoru a k převodu z trojfázové na dvoufázovou soustavu není nutné znát žádný ze stavů. Proto naprogramujeme identického pozorovatele s modelem ve statorových souřadnicích.

### 2.1.3 Diskretizace systému

Identický pozorovatel musí mít v sobě model rekonstruovaného systému. Algoritmus se implementuje na mikroprocesor, proto musíme model diskretizovat. Diskretizaci provedeme pomocí spočtení numerické derivace.

Na to existuje více způsobů, ale nám bude stačit Eulerova metoda a to z důvodu relativní početní složitosti Kalmanova filtru, aby byl výsledný algoritmus realizovatelný i na pomalejších procesorech. Jako případné zlepšení modelu se dá Eulerova metoda počítat vícekrát na vzorkovací interval a tím se výrazně zpřesní. To ale vede ke zpomalení algoritmu.

Obecná rovnice pro takto distretizovaný systém:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k) \quad (2.1a)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + T_s \frac{d\mathbf{x}}{dt} \quad (2.1b)$$

kde  $T_s$  je perioda vzorkování pro Eulerovu metodu

Diskretizovaný model synchronního motoru ve statorových souřadnicích pomocí Eulerovy metody:

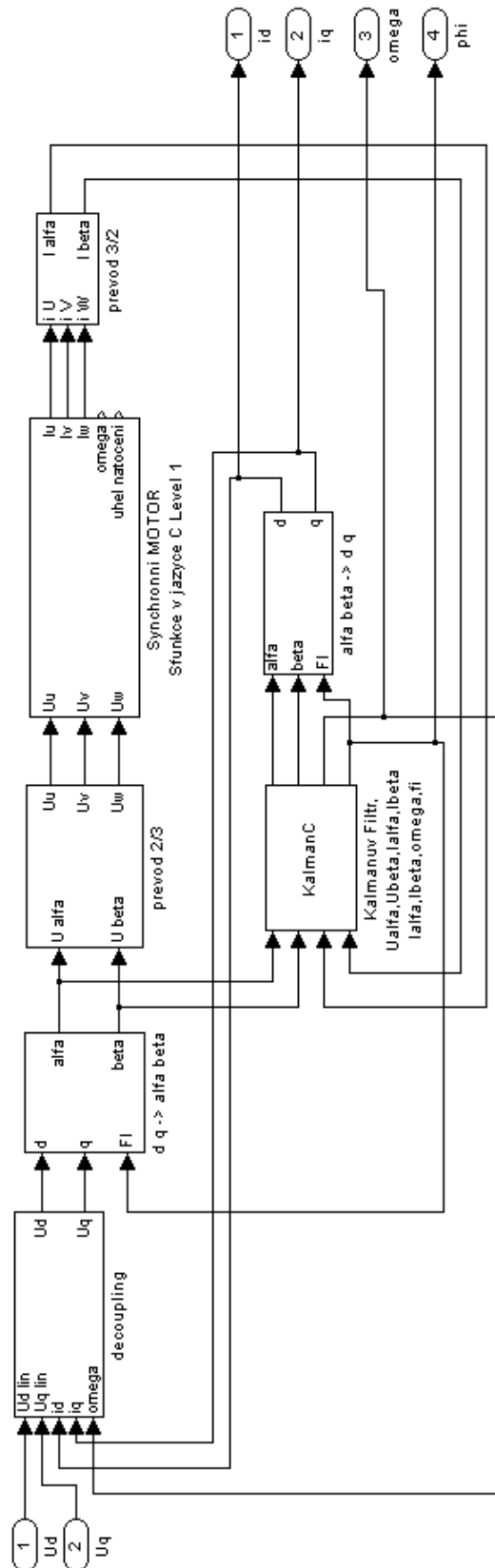
$$\begin{aligned} i_{\alpha k+1} &= i_{\alpha k} - \frac{RT_s}{L} i_{\alpha k} + \frac{T_s K_E \omega_k}{L} \sin \varphi_k + \frac{T_s}{L} u_{\alpha k} \\ i_{\beta k+1} &= i_{\beta k} - \frac{RT_s}{L} i_{\beta k} - \frac{T_s K_E \omega_k}{L} \cos \varphi_k + \frac{T_s}{L} u_{\beta k} \\ \omega_{k+1} &= \omega_k + \frac{3T_s P_p^2 K_E}{2J} i_{\beta k} \cos \varphi_k - \frac{3T_s P_p^2 K_E}{2J} i_{\alpha k} \sin \varphi_k \\ \varphi_{k+1} &= \varphi_k + T_s \omega_k \end{aligned} \quad (2.2)$$

### 2.1.4 Zařazení rekonstruktoru do zpětné vazby

Správně nastavený rekonstruktor odhaduje stavy, ale vrací je se zpožděním. To vadí decouplingu, který je na tuto vlastnost náchylný a přestává fungovat, jak by měl.

Funkce decouplingu sice nesplňuje svůj účel, ale odstranění tohoto bloku by bylo vážnou chybou, neboť kříženou vazbu alespoň částečně omezuje. Ze simulací se dá vyčíst, že složky jsou navzájem proporcionalně provázány. Abychom se vyrovnali s tímto problémem, musí se regulátory nastavit tak, aby regulace jedné složky byla alespoň dvakrát rychlejší než druhá. Regulátory se tak přestanou navzájem přetahovat.

Problém decouplingu při zapojení ve zpětné vazbě netkví jenom v neplnění své funkce zrušení křížené vazby, ale také, že mění charakter soustavy. I když bychom předpokládali, že obě složky budou mít po linearizaci integrační charakter, především q-složka tento charakter nemá. Při jednotkové skoku na q-složce napětí má odezva proudu derivační charakter a odezva otáček charakter setrvačného članku prvního řádu. Proto se musí regulátory znovu nastavit s ohledem na tyto skutečnosti.



Obrázek 2.1: Zapojení rekonstruktoru(zde Kalmanova filtru) do soustavy

## 2.2 Model Reference Adaptive System

Model Reference Adaptive System funguje na principu dvou modelů, z nichž jeden je závislý na rekonstruované veličině a druhý je nezávislý na rekonstruované veličině[6,7]. V případě synchronních motorů se volí magnetický tok. Jako referenční model slouží napěťový model a jako adaptační slouží proudový model.

### 2.2.1 Adaptační a referenční model

Jako adaptační model se používá proudový model magnetického toku a je závislý na natočení rotoru.

$$\begin{aligned}\Psi_{i\alpha} &= Li_{\alpha} + K_E \cos\phi_{re} \\ \Psi_{i\beta} &= Li_{\beta} + K_E \sin\phi_{re}\end{aligned}\quad (2.3)$$

Referenční model u synchronních motorů je volen jako napěťový model. Vychází se z vyjádření derivace magnetického toku:

$$\begin{aligned}\frac{d\Psi_{u\alpha}}{dt} &= u_{\alpha} - Ri_{\alpha} \\ \frac{d\Psi_{u\beta}}{dt} &= u_{\beta} - Ri_{\beta}\end{aligned}\quad (2.4)$$

odtud odvodíme spojitý napěťový model:

$$\begin{aligned}\Psi_{u\alpha} &= \int (u_{\alpha} - Ri_{\alpha}) dt \\ \Psi_{u\beta} &= \int (u_{\beta} - Ri_{\beta}) dt\end{aligned}\quad (2.5)$$

nahradíme integrál sumačním členem a dostaneme diskrétní napěťový model

$$\begin{aligned}\Psi_{u\alpha k+1} &= \Psi_{u\alpha k} + T_{vz} (u_{\alpha k} - Ri_{\alpha k}) \\ \Psi_{u\beta k+1} &= \Psi_{u\beta k} + T_{vz} (u_{\beta k} - Ri_{\beta k})\end{aligned}\quad (2.6)$$

Na začátku řízení se musí napěťový model správně nastavit. Pokud budeme předpokládat, že rotor je natočen o nulový úhel (rotor je natočen ve směru osy  $\alpha$ ), nastavíme hodnotu  $\Psi_{u\alpha 0}$  na  $K_E$  a  $\Psi_{u\beta 0}$  se nastaví na nulu. V napěťovém modelu se vyskytuje sumace což způsobí také sumaci chyby. Také hodnota  $K_e$  není většinou přesná. Proto se musí napěťový model ještě dále reguluje pomocí dvou PI regulátorů, kde jako vstup do regulátorů slouží odchylka magnetických toků z napěťového a z adaptačního modelu.

$$\begin{aligned}\Delta\Psi_{\alpha} &= \Psi_{i\alpha} - \Psi_{u\alpha} \\ \Delta\Psi_{\beta} &= \Psi_{i\beta} - \Psi_{u\beta}\end{aligned}\quad (2.7)$$

Jako regulátory stačí dva obyčejné PI regulátory pro každou složku  $\Psi$  s přenosem:

$$F_{\Psi} = \frac{k_1 p + k_2}{p} \quad (2.8)$$

Výstupem je potom korekce  $\delta_{\Psi}$ :

$$\begin{aligned} \delta_{\Psi} &= F_{\Psi} \Delta \Psi \\ \Psi_{u \text{ opravene}} &= \Psi_{uk} + T_{vz} \delta_{\Psi} \end{aligned} \quad (2.9)$$

## 2.2.2 Výpočet chyby

Abychom mohli reagovat na změnu polohy, musíme umět vyjádřit chybu  $\Delta \Phi_{re}$  odhadu polohy. Pokud statorový prostor převedeme do imaginární roviny, bude souřadnice  $\alpha$  odpovídat reálné ose a souřadnice  $\beta$  imaginární ose. Známe vektor  $\Psi_u$  a  $\Psi_i$ . Pokud chceme nalézt úhel mezi dvěma vektory použijeme vzorec:

$$\sin \Delta \Phi_{re} = \frac{\Im\{\Psi_u \bar{\Psi}_i\}}{|\Psi_u \bar{\Psi}_i|} \quad (2.10)$$

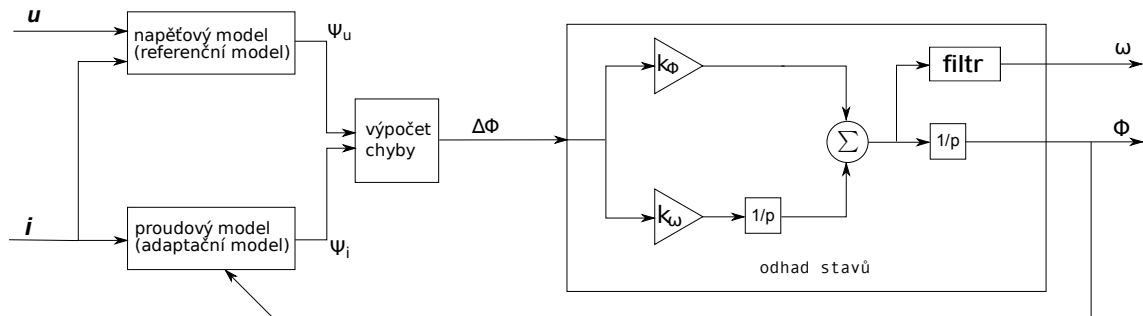
Pro velmi malé úhly můžeme zavést  $\sin \alpha \approx \alpha$  a odtud nalezneme vzorec:

$$\Delta \Phi_{re} \approx \frac{\Im\{\Psi_u \bar{\Psi}_i\}}{|\Psi_u \bar{\Psi}_i|} \quad (2.11a)$$

$$\Delta \Phi_{re} \approx \frac{\Psi_{u\beta} \Psi_{i\alpha} - \Psi_{u\alpha} \Psi_{i\beta}}{\sqrt{(\Psi_{u\alpha}^2 + \Psi_{u\beta}^2)(\Psi_{i\alpha}^2 + \Psi_{i\beta}^2)}} \quad (2.11b)$$

## 2.2.3 Odhad polohy a otáček

Motoru, který se točí konstantní úhlovou rychlostí  $\omega$ , lineárně narůstá úhel natočení. Pro případ MRAS musíme regulovat polohu, která v nejlepším případě lineárně narůstá. Pokud řídicí veličina lineárně narůstá a my ji chceme regulovat, musí být v regulační smyčce minimálně dva integrátory. Pro regulaci úhlu natočení použijeme PI regulátor a k němu sériově zapojený integrátor.



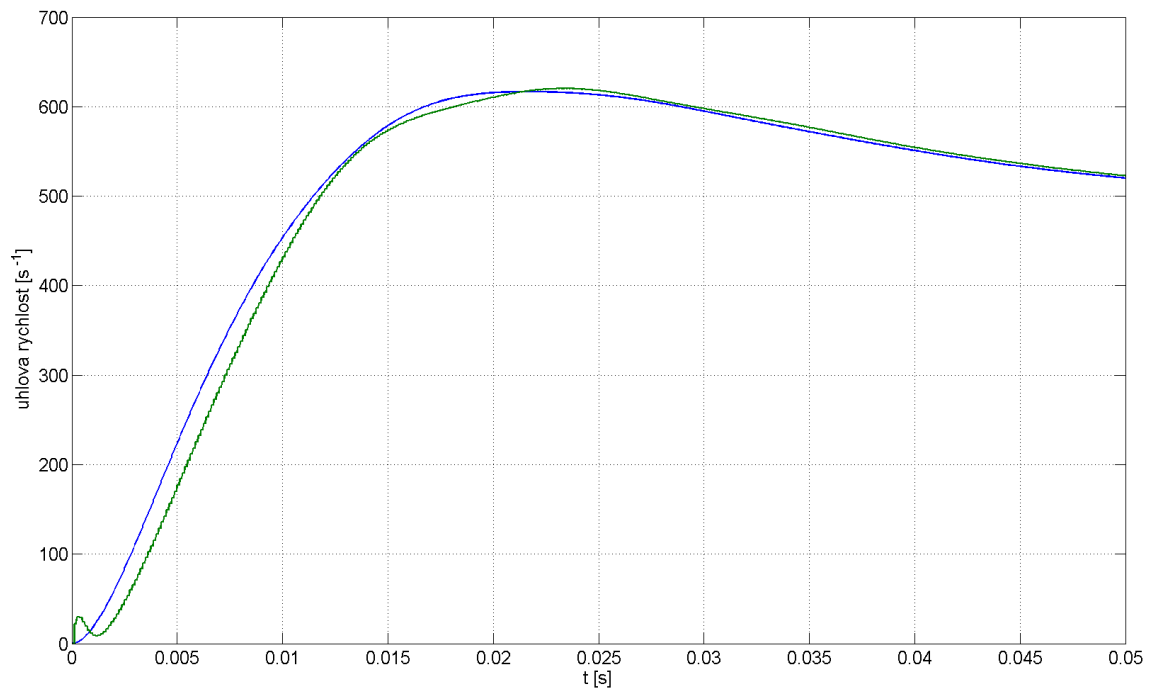
Obrázek 2.2: Schéma MRAS

Další důležitá představa je, že úhel natočení je integrálem otáček, proto nemusíme otáčky rekonstruovat. Otáčky jsou v tomto případě akční veličina a tudíž mohou kmitat, proto je pro lepší rekonstrukci odfiltrujeme signál filtrem prvního řádu.

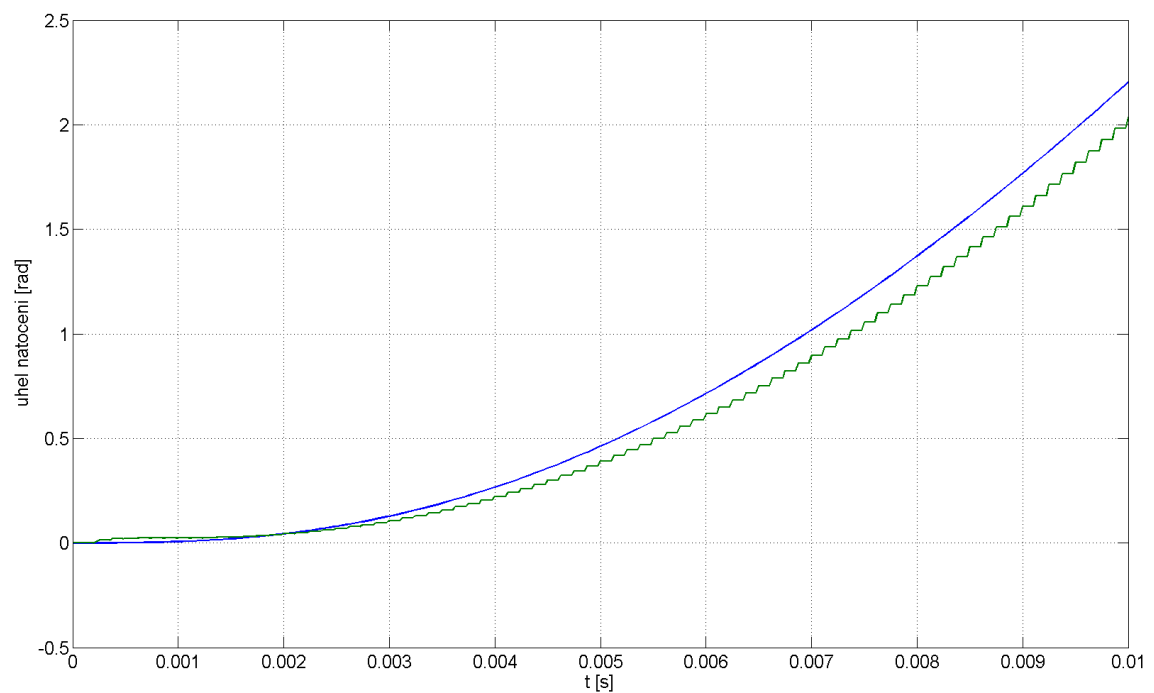
Výsledný přenos regulátoru bude:

$$F_{\Phi_{re}}(p) = \frac{\Phi_{re}}{\Delta\Phi_{re}} = \frac{k_{\Phi}p + k_{\omega}}{p} \frac{1}{p} \quad (2.12)$$

Na následujícím obrázku je ukázka rekonstrukce otáček pomocí MRAS. Zelená barva znázorňuje rekonstruované otáčky. Na začátku rekonstrukce dochází ke zpoždění o dvě periody vzorkování (dva sériově zapojené integrační členy) a regulátor se snaží toto zpoždění dohnat. To ovšem způsobí překmit na akční veličině, což jsou v tomto případě rekonstruované otáčky. Překmit omezíme filtrem prvního řádu, ovšem za cenu zhoršení fázové charakteristiky.



Obrázek 2.3: Rekonstrukce otáček pomocí MRAS(modrá-skutečné otáčky, zelená-rekonstruované otáčky)



Obrázek 2.4: Rekonstrukce polohy pomocí MRAS(modrá-skutečná poloha, zelená-rekonstruovaná poloha)

## 2.3 Kalmanův filtr

Kalmanův filtr je známý svou obecnou použitelností[5]. Rovnou se zaměříme na rozšířený Kalmanův filtr v diskretní podobě. Algoritmus Kalmanova filtru lze rozdělit na dvě části: predikci a korekci. Predikce slouží k předpovězení stavů a korekce slouží k opravení současných stavů.

Předtím, než se zaměříme na vlastní Kalmanův filtr, musíme definovat pár veličin:

$$e_k^- = x_k - \hat{x}_k^- \dots \text{znázorňuje chybu předpovězeného stavu od skutečného stavu}$$

$$e_k = x_k - \hat{x}_k \dots \text{znázorňuje chybu opraveného stavu od skutečného stavu}$$

$$\mathbf{P}_k^- = \mathbf{E} \left[ e_k^- e_k^{-T} \right] \dots \text{kovarianční matice chyb predikovaných stavů}$$

$$\mathbf{P}_k = \mathbf{E} \left[ e_k e_k^T \right] \dots \text{kovarianční matice chyb opravených stavů}$$

$$\mathbf{K}_k \dots \text{Kalmanovo zesílení (je také matice)}$$

### 2.3.1 Predikce

Predikce slouží k předpovězení stavů, které přijdou v následujícím kroku. Ke správné predikci je nutné znát současné stavy  $\mathbf{x}_{k-1}$ , také kovarianční matici opravených chyb a hodnotu vstupu. V první části predikce Kalmanův filtr předpoví následující stavy podle diskretizovaného modelu 2.2. Jediný problém vytváří numerické počítání derivace. Eulerova metoda patří k těm méně přesným. Pro zpřesnění spočteme derivaci na více menších kroků, nezvýší to tolik nároky na výpočet, ale výrazně ho zpřesní.

V dalším kroku se upraví kovarianční matice chyb predikovaných stavů podle vzorce:

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q} \quad (2.13)$$

kde matice  $\mathbf{Q}$  je parametr a určuje jak přesný je model v Kalmanově filtru a matice  $\mathbf{A}_k$  je Jacobiho matice:

$$\mathbf{A}_{k[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}} (\hat{x}_{k-1}, u_k) \quad (2.14)$$

Matici  $\mathbf{A}_k$  dostaneme poderivováním diskretního modelu synchronního motoru (vzorec 2.2):

$$\mathbf{K}_{\text{sub}} = \frac{3K_E P_p^2 T_s}{2J} \quad (2.15)$$

$$\mathbf{A}_k = \begin{bmatrix} 1 - \frac{RT_s}{L} & 0 & \frac{K_E T_s}{L} \sin \varphi_k & \frac{K_E T_s}{L} \omega_k \cos \varphi_k \\ 0 & 1 - \frac{RT_s}{L} & -\frac{K_E T_s}{L} \cos \varphi_k & \frac{K_E T_s}{L} \omega_k \sin \varphi_k \\ -K_{\text{sub}} \sin \varphi_k & K_{\text{sub}} \cos \varphi_k & 1 & K_{\text{sub}} (-i_{\beta k} \sin \varphi_k - i_{\alpha k} \cos \varphi_k) \\ 0 & 0 & T_s & 1 \end{bmatrix} \quad (2.16)$$

a vektor  $\hat{x}_k$  znázorňuje vektor stavů.

$$\hat{x}_k = \begin{bmatrix} i_{dk} \\ i_{qk} \\ \omega_k \\ \varphi_k \end{bmatrix} \quad (2.17)$$

Dále se provede predikce následujících stavů podle diskrétního modelu (vzorec 2.2):

$$\begin{aligned} i_{\alpha k+1} &= i_{\alpha k} - \frac{RT_s}{L} i_{\alpha k} + \frac{T_s K_E \omega_k}{L} \sin \varphi_k + \frac{T_s}{L} u_{\alpha k} \\ i_{\beta k+1} &= i_{\beta k} - \frac{RT_s}{L} i_{\beta k} - \frac{T_s K_E \omega_k}{L} \cos \varphi_k + \frac{T_s}{L} u_{\beta k} \\ \omega_{k+1} &= \omega_k + \frac{3T_s p_p^2 K_E}{2J} i_{\beta k} \cos \varphi_k - \frac{3T_s p_p^2 K_E}{2J} i_{\alpha k} \sin \varphi_k \\ \varphi_{k+1} &= \varphi_k + T_s \omega_k \end{aligned} \quad (2.18)$$

### 2.3.2 Korekce

Korekce slouží k opravení predikovaných stavů. Kalmanův filtr si na počátku změří vstupy a výstupy ze systému a vypočítá Kalmanovo zesílení. Pomocí Kalmanova zesílení opraví predikované stavy a kovarianční matici chyb opravených stavů:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \left( \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left( \mathbf{z}_k - h \left( \hat{\mathbf{x}}_k^- \right) \right) \\ \mathbf{P}_k &= \left( \mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^- \end{aligned} \quad (2.19)$$

kde  $(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-))$  značí rozdíl mezi tím, co filtr očekával, že naměří, a tím, co skutečně naměřil. Matice  $\mathbf{R}$  je parametr a určuje jak přesné je měření a matice  $\mathbf{H}_k$  má stejný význam jako matice  $\mathbf{C}$  ve spojitých systémech.

Jak je vidět z rovnic 2.19, Kalmanovo zesílení nám „opravuje“ stavy podle toho, jak moc se filtr v predikci zmýlil. Matice  $\mathbf{H}_k$  nám říká, které stavy se měří. V našem případě bude matice mít tvar:

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.20)$$

### 2.3.3 Nastavení Kalmanova filtru

Jak je vidět z rovnic 2.13 a 2.19, tak se filtr řídí sám bez cizího zásahu. Jediné parametry, které se nastavují, jsou matice chyb  $\mathbf{R}$  a  $\mathbf{Q}$  a počáteční kovarianční matice chyb opravených stavů  $\mathbf{P}_0$  v čase nula. Při bližším studiu Kalmanova filtru bychom přišli na to,

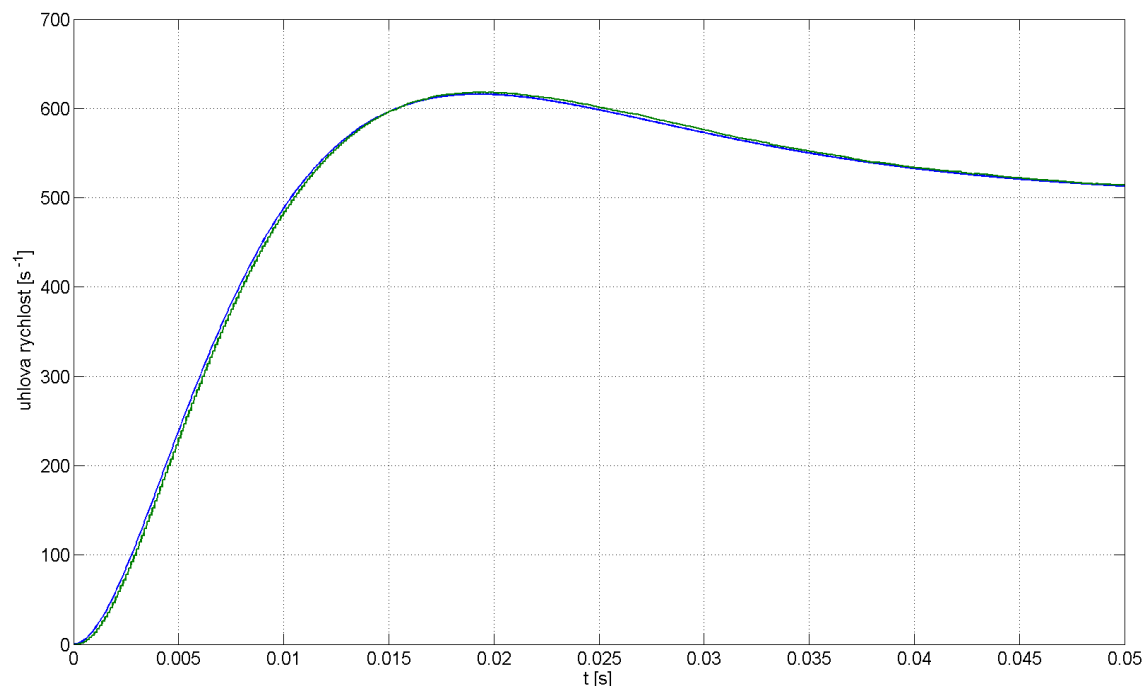
že volba  $P_0$  není pro chod celého filtru kritická, neboť filtr má schopnost velmi rychle konvergovat a během několika kroků si obě kovarianční matice chyb optimalizuje.

Mnohem důležitější je volba matic  $R$  a  $Q$ , které nám určují, jak přesný je model a jak přesné je měření. Volba zbytečně malých hodnot v matici  $R$  způsobí, že filtr až příliš věří hodnotám a pokud je v soustavě šum, filtr začne „zmatkovat“ a nepracuje správně. Naopak při volbě příliš malých hodnot v matici  $Q$  věří modelu a při poruše bude stále věřit modelu.

Větší hodnoty říkají, že filtr daným hodnotám nevěří. Ideální nastavení je nutno empiricky upravit podle konkrétního motoru.

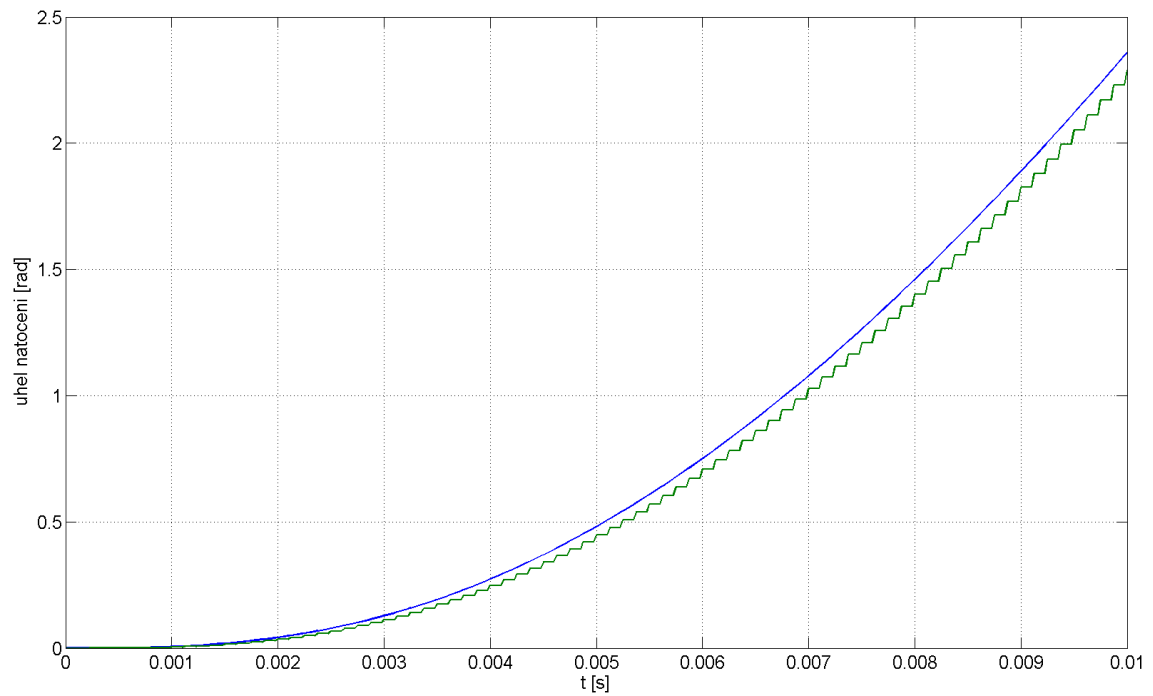
Při správném nastavení Kalmanova filtru je bezsnímačové řízení téměř plnohodnotným nahrazením snímačového řízení. Při vyšších otáčkách ale může dojít k tomu, že Kalmanův filtr nebude mít dostatečné množství informací pro rekonstrukci stavů a proto přestane rekonstruovat stavy a celý systém se zhroutl. Protože regulátor způsobuje překmit, nemůže regulátor nikdy regulovat až na maximální možné otáčky, kde Kalmanův filtr ještě funguje. Proto nejvyšší možné otáčky, které jsme schopni regulovat, jsou přibližně 75% maximálních otáček, které je Kalmanův filtr schopen zrekonstruovat.

Pokud bychom toto nedodrželi, může dojít k tomu, že motor překročí maximální otáčky a Kalmanův filtr přestane rekonstruovat stavy. Tím ztratíme jakékoliv informace o systému a nebudeme ho schopni dále regulovat.



Obrázek 2.5: Rekonstrukce otáček pomocí Kalmanova filtru se přidaným šumem 0.3A (modrá-skutečné otáčky, zelená-rekonstruované otáčky)

Na obrázku je ukázka rekonstrukce pomocí Kalmanova filtru, kde zelená barva ukazuje rekonstruované otáčky. Oproti MRAS, nedochází k žádnému překmitu a je také vidět značná odolnost proti šumu (šum zde tvoří téměř 10% signálu).



Obrázek 2.6: Rekonstrukce polohy pomocí Kalmanova filtru se přidaným šumem 0.3A (modrá-skutečná poloha, zelená-rekonstruovaná poloha)

## 3 IMPLEMENTACE ŘÍDICÍHO ALGORITMU

V této kapitole se budeme zabývat implementací na Freescale 56F8346 a problémy, které při implementaci vznikají.

### 3.1 Freescale 56F8346

Freescale 56F8346 je 16 bitový mikroprocesor o frekvenci 60MHz se znaky signálových procesor(DSP). Umí v jednom taktu násobit a přičíst hodnotu k akumulátoru. Rychle provádí bitový posun a bitovou rotaci. K mikroprocesoru je připojen PWM modul, A/D a D/A převodníky, paralelní port SPP pro připojení ke stolnímu počítači.

Jeho velkou nevýhodou je, že není vybaven jednotkou pro počítání s čísly s pohyblivou řádovou čárkou. ALU umí pouze násobit dvě čísla o pevné řádové čárce pokud jsou obě čísla v intervalu od minus jedné do plus jedné. To vede na normalizaci všech proměnných do hodnot menších než jedna.

Mikroprocesor použijeme k ověření funkčnosti a implementovatelnosti algoritmů na DSP pomocí metody „Target in the loop“. Na mikroprocesoru spustíme řídicí algoritmus a propojíme s počítačem pomocí paralelního portu. Na počítači spustíme matlab s modelem synchronního motoru a SFIO toolbox, který zajišťuje komunikaci s DSP.

#### 3.1.1 Normalizace veličin

Nepřítomnost FPU (Floating Point Unit) na mikroprocesoru nám nedovoluje pracovat s čísly v běžném formátu. Všechny vstupy, výstupy a konstanty regulátorů musíme normalizovat[9]. Budeme vycházet ze stavby 56F8346 podle vlastností periférií zvolíme normy na proudy a napětí.

Přehled použitých norem:

$U_{\max}$	16 V
$I_{\max}$	19,18 A
$\omega_{\max}$	$200 \pi$
$\varphi_{\max}$	$\pi$
$\Psi_{\max}$	30 mWb

Norma napětí je dána děličem na DC-busu. Ta je nastavena na 16 V a počítá a s rezervou na přepětí. Z důvodu přehlednosti je volena jedna společná norma pro všechna napětí. Norma proudu je dána schopností A/D převodníku na měření proudu. Při  $\omega = 200\pi$  bude synchronní motor indukovat asi 7V fázové napětí, což je 12V sdružené napětí a to je horní hranice D/A převodníků. Norma úhlu natočení je dána rozsahem kružnice. Abychom mohli obsáhnout celou kružnici, potřebujeme úhel  $2\pi$ , což je velikost intervalu od  $-\pi$

do  $\pi$ . Normování magnetického toku není nezbytné, ale magnetický tok se po celou dobu regulace pohybuje ve velmi malých hodnotách, proto je vhodné vytvořit normu i pro magnetický tok. Pokud používáme motor s  $K_E = 17mWb$ , bude norma  $30mWb$  dostačující.

### 3.1.2 Počítání koeficientů regulátorů

Všechny veličiny v regulátoru musí být v rozmezí minus jedna až plus jedna. Aby se zajistil norma na vstup i výstupu do regulátorů, musí se upravit všechny koeficienty regulátorů. Ty ovšem mohou vyjít jako čísla mnohem větší než jedna nebo naopak mnohem menší. To by způsobilo velké nepřesnosti. Proto musí být všechny koeficienty uloženy ve formátu hodnoty a bitového offsetu. Hodnota nesmí být menší než 0,5 a větší než 1. Bitový offset si můžeme jednoduše představit jako násobení mocninami čísla dvě.

Regulátory se musí před každým implementováním upravit a přepočítat. Zde je uveden přepočet diskrétního PI regulátoru použitého v řídicím algoritmu:

$$F_{PI}(p) = K \frac{T_p + 1}{p} \quad (3.1a)$$

$$y(k) = KT * u(k) + y(k-1) + KT_s u(k) \quad (3.1b)$$

$$y_n(k) \text{MAX}_{\text{vystup}} = KT \text{MAX}_{\text{vstup}} u_n(k) + \text{MAX}_{\text{vystup}} y_n(k-1) + KT_s \text{MAX}_{\text{vstup}} u_n(k) \quad (3.1c)$$

$$y_n(k) = KT \frac{\text{MAX}_{\text{vstup}}}{\text{MAX}_{\text{vystup}}} u_n(k) + y_n(k-1) + KT_s \frac{\text{MAX}_{\text{vstup}}}{\text{MAX}_{\text{vystup}}} u_n(k) \quad (3.1d)$$

$$P = KT \frac{\text{MAX}_{\text{vstup}}}{\text{MAX}_{\text{vystup}}}, I = KT_s \frac{\text{MAX}_{\text{vstup}}}{\text{MAX}_{\text{vystup}}} \quad (3.1e)$$

$$y_n(k) = P u_n(k) + y_n(k-1) + I u_n(k) \quad (3.1f)$$

kde  $y_n$  a  $u_n$  jsou normalizované vstupy a výstupy regulátorů I a P jsou parametry regulátoru. Po spočtení koeficientů P a I se ještě musí upravit do hodnoty v rozmezí minus jedna plus jedna a dvojkového offsetu.

### 3.1.3 MRAS v mikroprocesoru

Pokud se pozorně podíváme na strukturu MRAS, všimneme si že se skládá z výpočtu magnetických toků, chyby úhlu, PI regulátoru, sumačního členu a filtru. Všechny tyto části se musí normalizovat podle zavedených norem, stejně jako jsme provedli u PI regulátoru.

Jediný problém, na který narazíme, je normalizace výpočtu chyby. Ve výpočtu chyby se objevuje druhá odmocnina a také dělení, což jsou velmi náročné operace (viz. vzorec 2.11). Proto musíme vzorec zjednodušit tak, aby nebylo použito dělení!

Vektory  $\Psi_u$  a  $\Psi_i$  rotují kolem osy motoru ovlivňují výpočet chyby. Pokud nebude motor zatížen a zanedbáme tření, budou mít oba vektory po celou dobu regulace stejnou absolutní hodnotu. Pokud je absolutní hodnota součinu rovna součinu absolutních hodnot, tak můžeme absolutní hodnoty ve jmenovateli zlomku nahradit magnetickou konstantou motoru  $K_E$ . Dostaneme se k novému vyjádření chyby odhadu polohy:

$$\Delta\Phi_{re} \approx \frac{\Psi_{u\beta}\Psi_{i\alpha} - \Psi_{u\alpha}\Psi_{i\beta}}{K_E^2} \quad (3.2)$$

### 3.1.4 Normalizace matic Kalmanova filtru

V Kalmanově filtru je potřeba uchovávat informace v podobě matic i mezi cykly. Také model soustavy se musí znormalizovat, aby s ním mohl mikroprocesor pracovat. Vezmeme diskretizované stavové rovnice 2.2 a znormalizujeme podle zvolených norem. Vyjdou nám nové rovnice pro model systému:

$$\begin{aligned} i_{\alpha k+1} &= i_{\alpha k} - \frac{RT_s}{L}i_{\alpha k} + \frac{\omega_{\max}}{I_{\max}} \frac{T_s K_E \omega_k}{L} \sin(\varphi_{\max} \varphi_k) + \frac{U_{\max}}{I_{\max}} \frac{T_s}{L} u_{\alpha k} \\ i_{\beta k+1} &= i_{\beta k} - \frac{RT_s}{L}i_{\beta k} - \frac{\omega_{\max}}{I_{\max}} \frac{T_s K_E \omega_k}{L} \cos(\varphi_{\max} \varphi_k) + \frac{U_{\max}}{I_{\max}} \frac{T_s}{L} u_{\beta k} \\ \omega_{k+1} &= \omega_k + \frac{I_{\max}}{\omega_{\max}} \frac{3T_s p_p^2 K_E}{2J} i_{\beta k} \cos(\varphi_{\max} \varphi_k) - \frac{I_{\max}}{\omega_{\max}} \frac{3T_s p_p^2 K_E}{2J} i_{\alpha k} \sin(\varphi_{\max} \varphi_k) \\ \varphi_{k+1} &= \varphi_k + \frac{\omega_{\max}}{\varphi_{\max}} T_s \omega_k \end{aligned} \quad (3.3)$$

Nyní máme normalizovaný model synchronního motoru a žádná z veličin nemůže překročit číslo jedna. Normalizace modelu vede ke změnám počítání všech matic. Důležité je nalézt matice, které potřebují normalizace. Předtím, než vypočteme matici  $A_k$  zavedeme substituce

$$\begin{aligned} K_{1\text{sub}} &= \frac{K_E T_s \omega_{\max}}{L I_{\max}} \\ K_{2\text{sub}} &= \frac{3K_E p_p^2 T_s I_{\max}}{2J \omega_{\max}} \\ K_{\sin} &= \sin(\varphi_{\max} \varphi_k) \\ K_{\cos} &= \cos(\varphi_{\max} \varphi_k) \end{aligned} \quad (3.4)$$

matice  $A_k$  je potom rovna:

$$A_k = \begin{bmatrix} 1 - \frac{RT_s}{L} & 0 & K_{1sub}K_{sin} & K_{1sub}\varphi_{max}\omega_k K_{cos} \\ 0 & 1 - \frac{RT_s}{L} & -K_{1sub}K_{cos} & K_{1sub}\varphi_{max}\omega_k K_{sin} \\ -K_{2sub}K_{sin} & K_{2sub}K_{cos} & 1 & K_{2sub}\varphi_{max}(-i\beta_k K_{sin} - i\alpha_k K_{cos}) \\ 0 & 0 & \frac{\omega_{max}T_s}{\varphi_{max}} & 1 \end{bmatrix} \quad (3.5)$$

Ve dvou prvcích se může matice pohybovat mimo rozmezí minus jedna plus jedna. Nicméně normalizování dvou prvků v matici by způsobilo značné znepráhlednění programu a nezpůsobilo výrazné zlepšení efektivity. Proto je lepší zavést společnou normu pro celou matici. Volba normy nesmí být nahodilá a musí vycházet ze struktury procesoru. Ideální je volit normy jako násobky dvou, protože se číslo neuchovává jako norma, ale jako binární offset. Operace bitového posunu je rychlá a proto nezpomaluje řídicí algoritmus.

Matice  $P_k$  a  $P'_k$  jsou kovariantní matice chyb predikovaných a opravených stavů. Protože je zajištěno, že hodnoty normalizovaných stavů budou vždy menší než jedna, musí být už z principu kovariantní matice také menší než jedna.

Další důležitá úvaha je v nastavení Kalmanova filtru a matic  $Q_k$  a  $R_k$ . Matice  $R_k$  určuje nepřesnost měření stavů, v našem případě proudu. Normu proudu ale známe, proto musíme celou matici vydělit normou proudu a uložit v normalizované formě. Stejnou úvahu provedeme na matici  $Q_k$ , která uvádí nepřesnosti modelu. Pro zjednodušení budeme volit diagonální matici a budeme uvažovat, že hodnoty v diagonále jsou vždy nepřesnosti daného stavu. Každý stav má ale jinou normu a proto se musí matice vydělit normou daného stavu.

Poslední a nejkomplicovanější je uchovávaní Kalmanova zesílení  $K_k$ . Pro Kalmanovo zesílení nelze určit jednu normu po celou dobu programu, protože se během chodu dynamicky mění. Pokud se podíváme na výpočet Kalmanova zesílení (viz. vzorec 2.19), všimneme si počítání inverzní matice. Výpočet inverzní matice je možné spočítat pomocí adjungovaných matic a determinantu, kde determinant je vždy menší než jedna. Pokud bychom dělili matici v tomto okamžiku determinantem, mohli bychom dostat číslo větší než jedna! Proto uložíme determinant jako parametr společný pro celou matici a tím zajistíme, že matice nepřeteče.

### 3.1.5 Optimalizace Kalmanova filtru

Rekonstrukce stavů pomocí Kalmanova filtru dává vysoké nároky na výpočetní výkon mikroprocesoru, který bohužel nemáme k dispozici. Proto je nutné celý algoritmus zoptimalizovat.

První problém je s výpočty, kde se objevuje  $K_k$ . Matice  $K_k$  je uložena ve formě prvků matice a determinantu inverzní matice. Abychom dostali skutečné hodnoty museli bychom celou matici vydělit determinantem, což by bylo osm operací dělení. Dělení je zakázaná operace, ale počítání s determinantem se nelze vyhnout.

Jak jsme si již uvedli o normalizaci, nejvhodnější je volit si normy jako násobky čísla dvě, protože násobní mocninami dvou se dá realizovat pomocí bitového posunu a bitový posun je nesrovnatelně rychlejší než dělení. Proto provedeme aproximaci determinantu na nejbližší násobek dvou a tím získáme offset pro celou matici.

Tato operace změní velikost Kalmanova zesílení, ale pokud se podíváme na jeho význam, tak si všimneme, že násobí rozdíl naměřené hodnoty a modelu a opravuje odhadované stavy. Pokud se Kalmanovo zesílení uměle zmenší (pouze v přípustných mezích, nelze zmenšit trvale!!), dojde k omezení korekce stavů a tím i větší víře v model, ale vlastní funkce zůstane zachována.

I po této optimalizaci je algoritmus pomalý a není schopen v požadovaném čase rekonstruovat stavy. Při provedení analýzy celého filtru si všimneme konstantních matic a prvků matic, které se po celou dobu rekonstrukce nemění. Dále výpočet adjungovaných matic je také výpočetně náročný. Proto konstantní matice  $H_k$  není v algoritmu uchovávána. Je schovaná ve změně výpočetních procesů ve filtru.

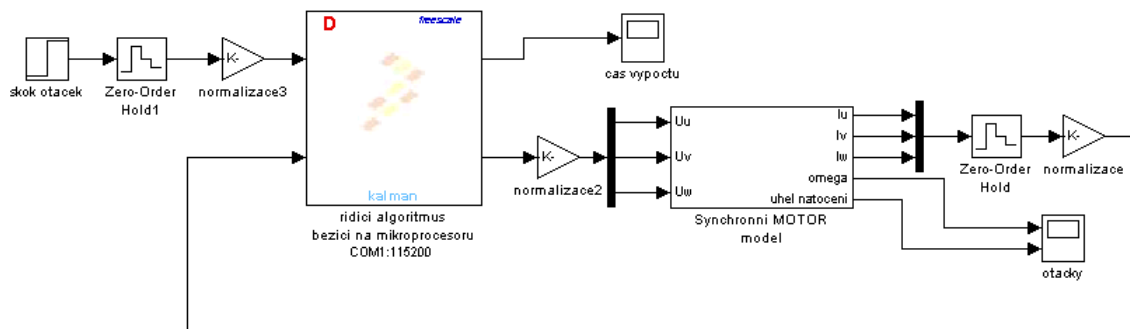
Těmito operacemi jsme schopni výrazně zrychlit rekonstrukci. Rekonstrukce však bude stále pomalá z důvodu násobení matic. Mikroprocesor nejvíce zatěžují podmíněné skoky a také cykly. K násobení matice se používá tři do sebe vnořených cyklů a podmínka nejnitřnějšího cyklu při násobení matic se prování  $4^3$ -krát, což neuvěřitelně zpomaluje výpočet. Procesor potom stráví 80% procesorového času počítáním cyklů a pouhých 20% času skutečným výpočtem.

Řešení tohoto problému vede k assembleru a hardwarovým cyklům, které jsou v DSP velmi rychlou operací. Jiným méně efektivním způsobem řešení je unlooping (rozmotání cyklů). Unlooping povede k výraznému znehlednění programu a téměř nemožnému debugging, proto je nutné program nejdříve odladit a potom teprve provést unlooping. Také dojde k výraznému zvětšení zdrojového kódu.

Pokud provedeme náhradu dělení a rozmotání všech cyklů, klesne výpočetní čas na minimum a procesor je schopen provést rekonstrukci včetně regulace za méně než  $125\mu s$ . Za tento čas je již možné motor řídit.

## 3.2 Simulace v prostředí Matlab Simulink

Ověření funkčnosti algoritmů provedeme metodou „target in the loop“. Řídicí algoritmus implementujeme na mikroprocesor a v matlabu spustíme toolbox SFIO. Model motoru běží jako S-funkce v Matlabu. Důležité je správně nastavit vstupní a výstupní formáty dat. Před vstupem do mikroprocesoru musí být všechny veličiny normalizovány a stejně tak i z mikroprocesoru vystupují normalizované veličiny.

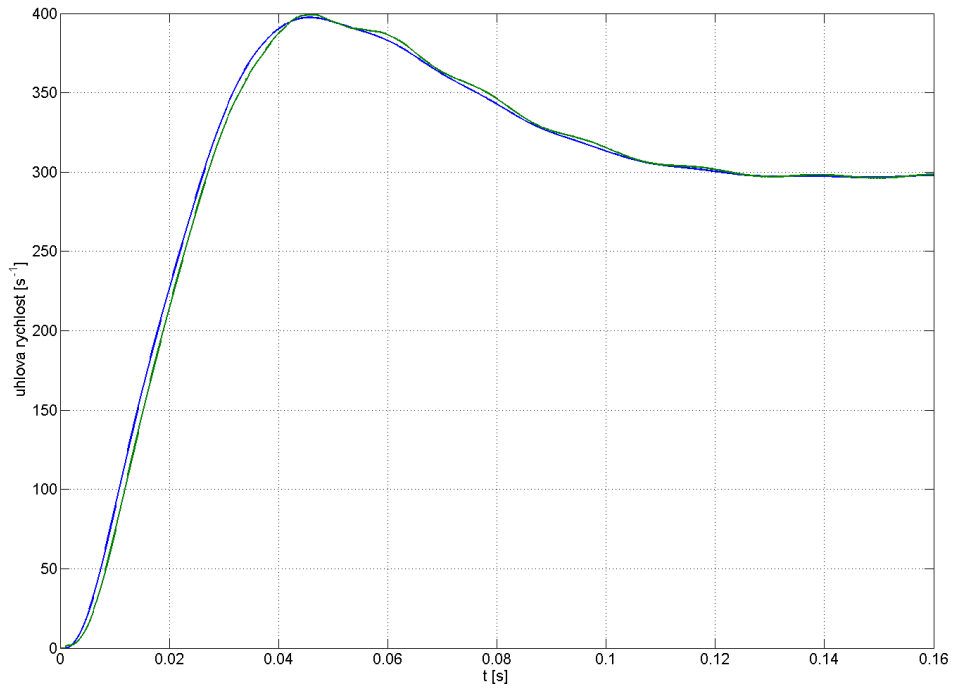


Obrázek 3.1: Zapojení mikroprocesoru do regulační smyčky(metoda target in the loop)

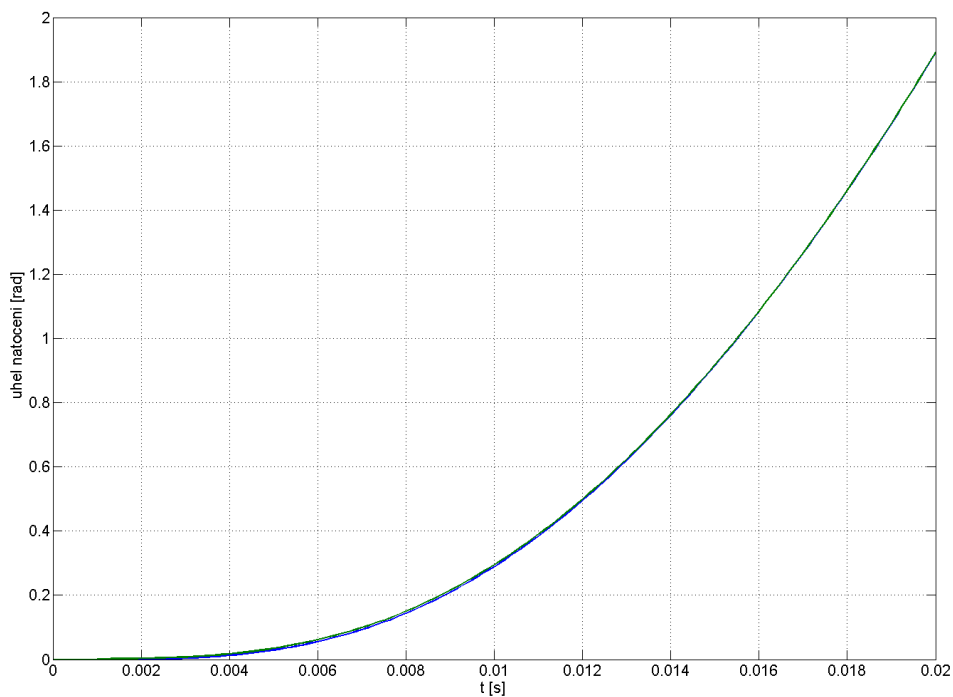
Na obrázku 3.2 je odezva na skok otáček, kdy celý řídicí algoritmus je implementován na mikroprocesoru. Rekonstrukce otáček je provedena pomocí MRAS. Nepřesnosti v rekonstrukci způsobují zvlnění přechodové charakteristiky.

Na obrázku 3.4 je rekonstrukce prováděna pomocí Kalmanova filtru. Kalmanův filtr se jeví jako kvalitnější rekonstruktor a nedochází k takovému zvlnění přechodové charakteristiky jako u MRAS. Po přidání šumu 0.3A(obrázek 3.6) je Kalmanův filtr stále schopen kvalitně rekonstruovat a řídit motor.

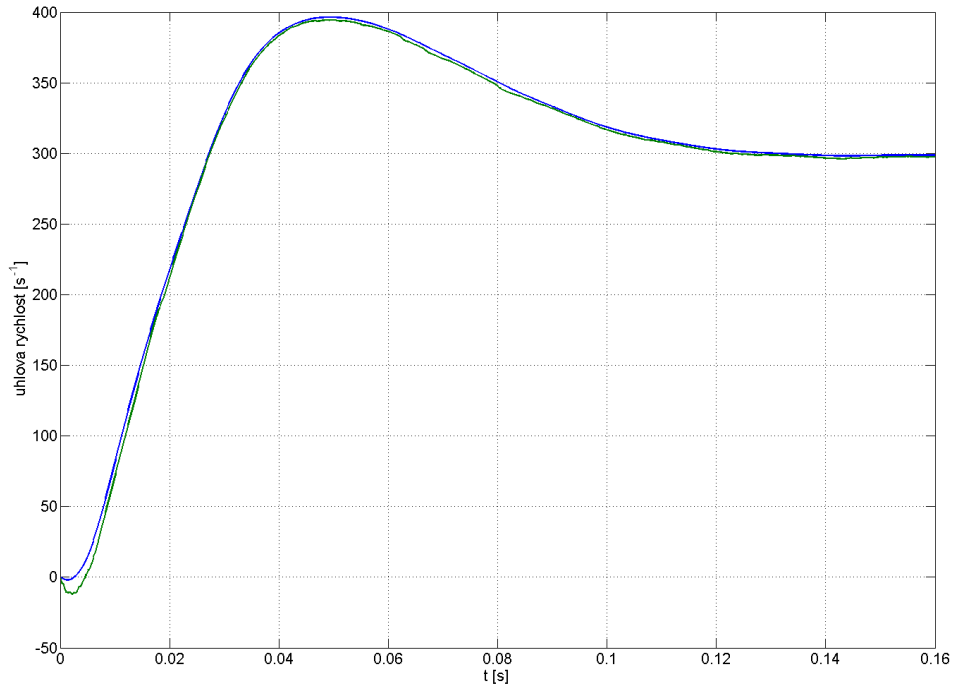
Pokud se podíváme obrázky 3.4 a 3.5 pozorněji, všimneme si, že zpočátku je odhad polohy a otáček velmi nepřesný. Synchronní motor je za otáček blízkým nule nepozorovatelný[7]. Důkaz vychází z teoremu pozorovatelnosti a sahá nad rámec této práce.



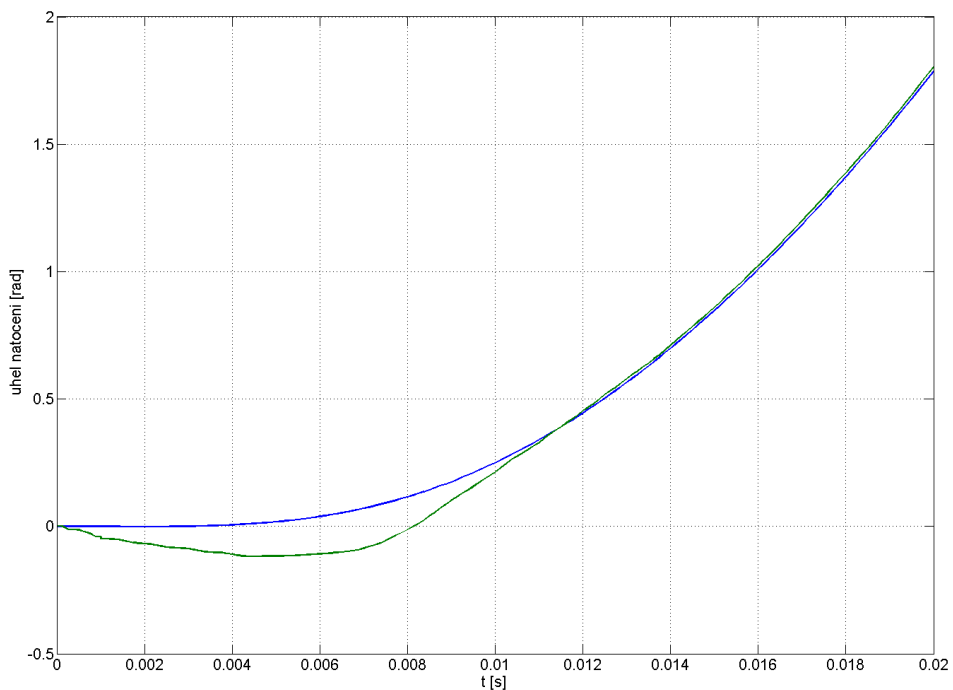
Obrázek 3.2: Odezva na skok otáček při použití Model Reference Adaptive System(modrá-skutečné otáčky, zelená-rekonstruované otáčky)



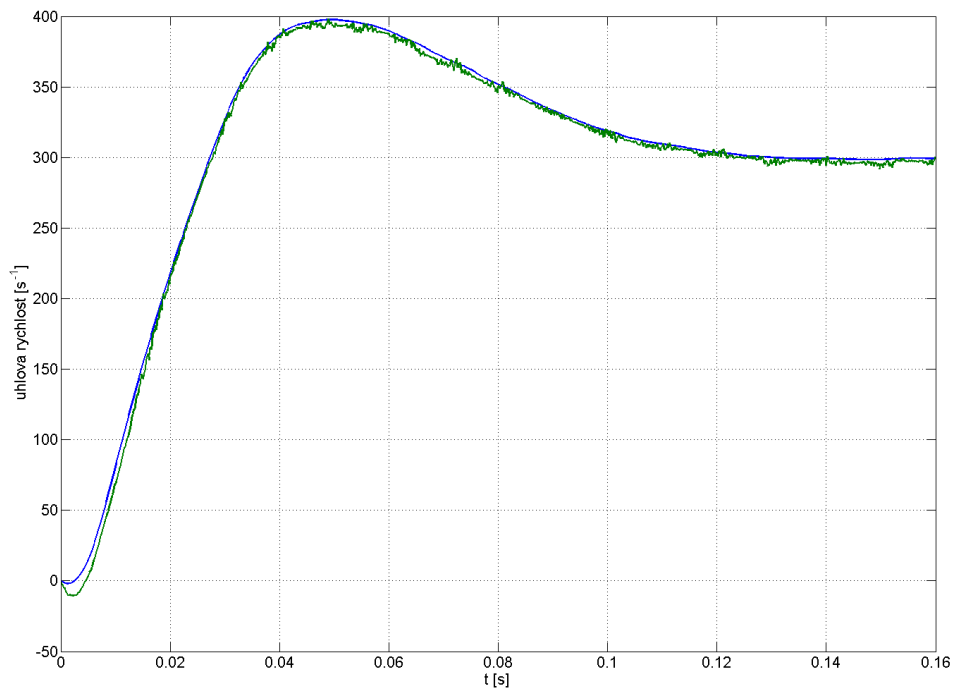
Obrázek 3.3: Odezva na skok otáček při použití Model Reference Adaptive System(modrá-skutečná poloha, zelená-rekonstruovaná poloha)



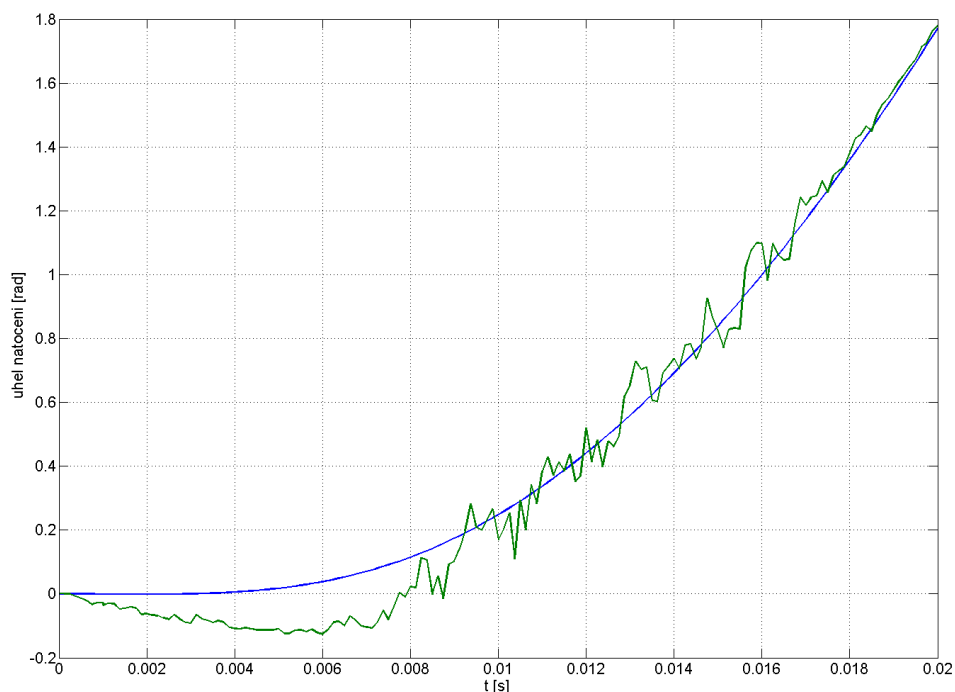
Obrázek 3.4: Odezva na skok otáček při použití Kalmanova filtru(modrá-skutečné otáčky, zelená-rekonstruované otáčky)



Obrázek 3.5: Odezva na skok otáček při použití Kalmanova filtru(modrá-skutečná poloha, zelená-rekonstruovaná poloha)



Obrázek 3.6: Odezva na skok otáček po přidání šumu (0.3A) a rekonstrukce pomocí Kalmanova filtru(modrá-skutečné otáčky, zelená-rekonstruované otáčky)



Obrázek 3.7: Odezva na skok otáček po přidání šumu (0.3A) a rekonstrukce pomocí Kalmanova filtru(modrá-skutečná poloha, zelená-rekonstruovaná poloha)

## 4 VÝSLEDKY

Parametry modelu motoru, který sloužil k simulacím:

$$L = 0.2 \text{ mH}$$

$$R = 0.275 \Omega$$

$$J = 10^{-4} \text{ kg} \cdot \text{m}^2$$

$$p_p = 3$$

$$K_E = 17.1 \text{ mWb}$$

### Snímačové řízení

Pro snímačové řízení synchronních motorů je nejvýhodnější použít vektorové řízení. Před regulátory je nutné použít zpětnovazební linearizaci pro zrušení křížových vazeb. Aby docházelo k co nejmenším ztrátám v železe, reguluje se d-složka proudu na nulu a q-složka proudu je zapojena v kaskádní regulaci.

Pro regulátory proudu stačí PI regulátory. Pro regulátor otáček musí být PID regulátor, podařilo se nám regulovat otáčky s překmitem přibližně 25 procent (viz obrázek 1.4).

### Bezsnímačové řízení

Při rekonstrukci stavů dochází ke zpoždění o jednu periodu a proto není rekonstrukce zcela aktuální. To má vliv hlavně na decoupling, který není tak přesný jako u snímačového řízení. Proto se musí přenastavit celá regulace. Regulátory jsou stejného typu jako u snímačového řízení, ale mají jiné parametry. Ty se musí nalézt až po nastavení rekonstruktoru.

Hlavní problém bude přetahování regulátorů d a q složky proudu. Abychom předešli přetahování regulátorů a splnili podmínky vektorového řízení, musí být regulátor d-složky proudu alespoň dvakrát rychlejší než regulátor q-složky proudu.

Odezva na jednotkový skok byla malinko horší i s větším překmitem (viz obrázek 3.4), ale řízení bylo funkční. Bezsnímačové řízení je závislé na rekonstruktoru stavů, který je realizován disktrétně. Pro příliš velkých otáčkách přestává filtr stíhat rekonstruovat stavy a tím se zhroutí celá regulace. Motor nesmí nikdy během řízení překročit tyto maximální otáčky. Kalmanův filtr přestává rekonstruovat při  $6300 \text{ ot min}^{-1}$  ( $\omega_{el} = 2000 \text{ s}^{-1}$ ), proto nejvyšší možné otáčky, které jsme byli schopni regulovat, byli přibližně  $4500 \text{ ot min}^{-1}$  ( $\omega_{el} = 1500 \text{ s}^{-1}$ ). MRAS přestává rekonstruovat asi při  $\omega_{el} = 1800 \text{ s}^{-1}$

## MRAS

MRAS funguje na principu PI regulátorů a využívá jejich výhod i nevýhod. Na začátku rekonstrukce dojde ke zpoždění o dvě periody a při snaze regulátoru dohnat rekonstrukci dojde k výraznému překmitu na rekonstrukci otáček. Tento překmit se dá zmenšit zapojením filtru za cenu zhoršení fázové charakteristiky.

Hlavní výhodou MRAS je jednoduchost algoritmu a také jeho rychlost. Je implementovatelný i na pomalejší procesory. Hlavní nevýhodou je kvalita rekonstrukce.

## Kalmanův filtr

U Kalmanova filtru je zcela zásadní správné nastavení matic  $R$  a  $Q$ , kde určujeme, jestli věříme více modelu nebo měření. I při správném volbě matic vlivem zpoždění o jednu periodu přestává zpětnovazební linearizace plnit svůj účel. Její vyřazení z regulace je však nevhodné.

Kalmanův filtr je velmi kvalitní rekonstruktor a přestává rekonstruovat při relativně vysokých otáčkách. Bohužel se musí po celou dobu regulace zajistit, aby neztratil kontakt se skutečnou hodnotou. Pokud rozšířený Kalmanův filtr přestane rekonstruovat, není již schopen se vrátit a je nutné motor zastavit.

Hlavní nevýhodou je komplikovanost algoritmu. Algoritmus je složitý a velmi výpočetně náročný, proto je nutné vhodně zvolit výpočetní jednotku.

## Porovnání rekonstruktorů a implementace na mikroprocesor

Obecně ukázal Kalmanův filtr kvalitnější rekonstrukci bez překmitů a fázových zpoždění. Bohužel jeho složitost zvyšuje nároky buď to na programátora(assembly) nebo na výpočetní jednotku. MRAS naopak je naopak jednodušší a je možné ho použít i na pomalejší procesory.

My jsme implementovali oba algoritmy na Freescale 56F8346, který neumí počítat s pohyblivou řádovou čárkou. Tento problém je řešitelný normalizací a počítáním s pevnou řádovou čárkou. Může se zdát, že jde o komplikovanou operaci, ale protože DSP jsou přizpůsobené na bitové posunu, nejde o žádné výrazné zpomalení.

Největší zpomalení způsobovalo dělení a také cykly. Dělení je v řízení zakázaná operace proto jsme jej aproximovali násobením konstantou. Pokud nebylo možné použít konstantu, našli jsme nejbližší bitový posun. Cykly jsme rozepsali do jednotlivých operací. Jednalo se sic o výrazné znehlednění kódu, ale bylo zcela nutné pro splnění vzorkovacích požadavků.

## 5 ZÁVĚR

V předložené práci bylo demonstrováno řízení synchronního motoru bez snímačů mechanických veličin. Na rekonstrukci byli použity Kalmanův filtr a MRAS. Obecně bylo bezsnímačové řízení méně přesné a použitelné pouze pro jednodušší aplikace.

Na mikroprocesor Freescale 56F8346 byli implementovány oba rekonstruktory a pomocí metody target in the loop byla ověřena jejich funkčnost. Kalmanův filtr se jevil jako kvalitnější rekonstruktor, hlavní nevýhodou byla jeho složitost.

Předpokládám, že práce na toto téma bude pokračovat. Rád bych se zaměřil na srovnání s dalšími rekonstruktory a použitelnost rekonstruktorů pro další typy motorů.

## REFERENCE

- [1] CAHA, Z.; ČERNÝ, M. *Elektrické pohony*. Praha: SNTL, 1990
- [2] ŠUBRT, J. *Elektrické regulační pohony II*. Brno: Vysoké učení technické v Brně, 1987
- [3] ŠTECHA, J.; HAVLENA, V. *Teorie dynamických systémů*. Praha: České vysoké učení technické v Praze, 2005
- [4] MAYBECK, P. *Stochastic models, estimation, and control*. New York: ACADEMIC PRESS, 1979
- [5] WELCH, G.; BISHOP, G. *An Introduction to the Kalman Filter*. Los Angeles: SIGGRAPH, 2001
- [6] ESKOLA, M.; TUUSA, H. *Comparison of MRAS and Novel Simple Method for Position Estimation in PMSM Drives*. Power Electronics Specialist Conference, 2003. PESC '03. 2003 IEEE 34th Annual, s. 550 - 555
- [7] VÁCLAVEK, P.; BLAHA, P. *Synchronous Machine Drive Observability Analysis and Sensorless Control Design*. Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International, s. 265 - 270
- [8] *MISRA-C:2004 - Guidelines for the use of the C language in critical systems*. Warwickshire: MIRA Ltd., 2004. 106 s.
- [9] ŠOLC, F.; VÁCLAVEK, P. *Modelování a simulace*. Brno: Ústav automatizace a měřicí techniky VUT Brno, 2009
- [10] *DSP56800E Quick Start User's Manual*. Denver: FREESCALE SEMICONDUCTOR Inc., 2007. 1068 s.
- [11] *DSP56800E 16-Bit Digital Signal Processor Core*. Denver: FREESCALE SEMICONDUCTOR Inc., 2001. 706 s.
- [12] *CodeWarrior™ Development Studio for Freescale™ 56800/E Digital Signal Controllers: MC56F8xxx/DSP5685x Family Targeting Manual*. Denver: FREESCALE SEMICONDUCTOR Inc., 2007. 360 s.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

SPMSM surface permanent-magnet synchronous motor

MRAS Model Reference Adaptive System

DSP digital signal processor

$i$  okamžitá hodnota proudu

$u$  okamžitá hodnota napětí

J moment setrvačnosti

M moment

$p_p$  počet pólových dvojic

R odpor

L indukčnost

T časová konstanta

$T_s$  perioda vzorkování

$\alpha$  úhel obecně

$\varphi$  úhel natočení

$\Psi$  fázor spřaženého toku

$K_E$  magnetická konstanta motoru

$\omega$  úhlová rychlost

$\omega_{el}$  úhlová rychlost točivého magnetického pole

$\omega_m$  úhlová rychlost rotoru