



Fakulta elektrotechnická
Katedra teoretické elektrotechniky

Diplomová práce

Prediktivní řízení asynchronního motoru

Autor práce: Bc. David Jenší
Vedoucí práce: Ing. Jakub Talla, Ph.D.

Plzeň 2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David JENŠÍ**

Osobní číslo: **E17N0052P**

Studijní program: **N2612 Elektrotechnika a informatika**

Studijní obor: **Dopravní elektroinženýrství a autoelektronika**

Název tématu: **Prediktivní řízení asynchronního motoru**

Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

Tato diplomová práce se týká návrhu a implementace prediktivního řízení asynchronního motoru.

1. Stručně popište metodu prediktivního řízení s konečným počtem akčních zásahů FCS-MPC.
2. Navrhňte a odsimulujte regulaci rychlosti asynchronního motoru pomocí metody FCS-MPC.
3. Naprogramujte do mikroprocesoru TMS320F28377 navržené algoritmy.
4. Proveďte optimalizaci výpočtu pomocí paralelizace s využitím obou jader a CLA mikroprocesoru.
5. Zhodnoťte dosažené výsledky v diplomové práci.

Rozsah grafických prací: podle doporučení vedoucího

Rozsah kvalifikační práce: 40 - 60 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

Approximate predictive control of AC electric drives, dizertační práce, Štěpán Janouš

Vedoucí diplomové práce:

Ing. Jakub Talla, Ph.D.

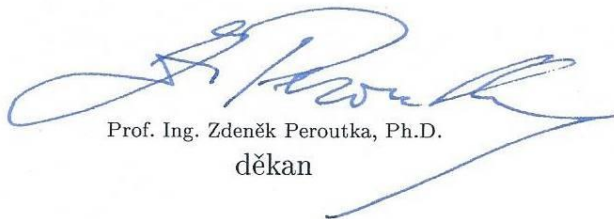
Regionální inovační centrum elektrotechniky

Datum zadání diplomové práce:


5. října 2018

Termín odevzdání diplomové práce:

30. května 2019



Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 5. října 2018

Abstrakt

Práce se zabývá prediktivním řízením asynchronního motoru metodou FCS-MPC (finite control set model predictive control). V úvodní části je obecně popsána problematika prediktivního řízení a více je rozebrána metoda FCS-MPC. Dále jsou popsány všechny hardwarové komponenty, které jsou použity v praktické části práce. Třetí kapitola se zabývá simulacemi metod FCS-MPC pro prediktivní regulaci proudu a otáček. V poslední části je popsána implementace programu v mikrokontroléru TMS320F28377D pro různé metody uvedené v třetí kapitole na jednom a na dvou jádrech společně s využitím koprocesorů CLA. Dále jsou v této kapitole uvedené výsledky měření získané z reálné aplikace metod FCS-MPC.

Klíčová slova

Prediktivní řízení, asynchronní motor, MPC, FCS-MPC, mikroprocesor, TMS23028377D, paralelní výpočty, CLA.

Abstract

The thesis deals with predictive control of asynchronous motor by FCS-MPC method (finite control set model predictive control). In the introduction part, there is described the issue of predictive control and there is discussed the FCS-MPC method. There are all the hardware components described and furthermore used in the practical part of the thesis. Simulations of FCS-MPC methods for predictive current and speed control are analyzed in the third part. The last part describes the implementation of the program in the TMS320F28377D microcontroller on one and two cores together with the use of CLA coprocessors for various methods mentioned in the third chapter. Furthermore, the results of measurements obtained from real application of FCS-MPC methods are presented in this chapter.

Keywords

Predictive control, induction motor, MPC, FCS-MPC, microcontroller, TMS23028377D, parallel computing, CLA.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

.....

podpis

V Plzni dne 29. května 2019

XXX

Poděkování

Tímto bych rád poděkoval panu Ing. Jakubovi Tallovi, Ph.D. za cenné profesionální rady, připomínky a metodické vedení práce.

Obsah

Úvod	11
1 Teorie	12
1.1 Řídicí systémy ve výkonové elektronice	12
1.2 Metody Prediktivního řízení	13
1.3 FCS-MPC ve výkonové elektronice	14
1.3.1 Prediktivní horizont	16
1.4 Měníč kmitočtu	17
1.5 Ztrátová funkce	18
1.6 Zpoždění výpočtů	19
1.6.1 Kompenzace zpoždění	20
1.7 Matematický model asynchronního motoru	21
1.7.1 Transformace souřadnic	22
1.7.2 Matematický model v dq	23
2 Hardware	25
2.1 Mikroprocesor	25
2.1.1 Floating Point Unit(FPU)	25
2.1.2 Trigonometric Math Unit(TMU)	26
2.1.3 Inter Processor Communication (IPC)	26
2.1.4 Global Shared RAM	27
2.1.5 IPC message RAM	27
2.1.6 IPC události a přerušení	27
2.2 Control Law Accelerator (CLA)	27
2.3 Periférie	28
2.3.1 Enhanced Pulse width modulation(ePWM)	28
2.3.2 Enhanced Quadrature Encoder Pulse(eQEP)	29
2.3.3 Analog to Digital Converter(ADC)	30
2.4 Měníč kmitočtu	31
2.5 Asynchronní motor 1AC 7070-4AB10Z	32
3 Simulace	33
3.1 PI regulace otáček s FCS-MPC regulátorem proudu pro jeden predikční horizont	34
3.2 PI regulace otáček s FCS-MPC regulátorem proudu pro jeden predikční horizont s modifikací	35

3.3	PI regulace otáček s FCS-MPC regulátorem proudu pro dva predikční horizonty	36
3.4	FCS-MPC regulace otáček pro jeden predikční horizont	38
3.5	FCS-MPC regulace otáček pro jeden predikční horizont s modifikací	39
3.6	FCS-MPC regulace otáček pro dva predikční horizonty	40
4	Implementace v procesoru	42
4.1	Popis programu	42
4.2	Výsledky měření	46
4.2.1	FCS-MPC proudu a PI regulátor otáček pro jeden predikční horizont	46
4.2.2	FCS-MPC proudu a PI regulátor otáček pro dva predikční horizonty	48
4.2.3	FCS-MPC otáček pro jeden predikční horizont	49
4.2.4	FCS-MPC otáček pro dva predikční horizonty	50
4.2.5	Doba výpočtu	51
5	Závěr	53

Seznam použitých symbolů a zkratek

FCS – MPC		Finite Control Set Model Predictive Control
CCS – MPC		Continuos control set Model Predictive Control
LUT		Look-up table
PH		Prediktivní horizont
J		Ztrátová funkce
IGBT		Insulated Gate Bipolar Transistor
MOSFET		Metal Oxide Semiconductor Field Effect Transistor
AS		Asynchronní motor
SOC		Start of conversation
EOC		End of conversation
CPU		Central processing unit(centrální procesorová jednotka)
S	$[-]$	Akční zásah
x	$[-]$	Obecná stavová proměnná
U_{CES}	$[V]$	Jmenovité napětí tranzistoru
I_C	$[A]$	Jmenovitý kolektorový proud tranzistoru
I_{CRM}	$[A]$	Maximální kolektorový proud tranzistoru
V_{GES}	$[V]$	Řídící napětí tranzistoru
T_j	$[^{\circ}C]$	Maximální teplota přechodu tranzistoru
P_N	$[W]$	Jmenovitý výkon
U_{sn}	$[V]$	Jmenovité napětí statoru
f_{sn}	$[Hz]$	Jmenovitá frekvence
p_p	$[-]$	Počet pólových dvojic
R_s	$[\Omega]$	Odpor statoru
R_r	$[\Omega]$	Odpor statoru
L_m	$[H]$	Magnetizační indukčnost
$L_{s\sigma}$	$[H]$	Rozptylová indukčnost statoru
$L_{r\sigma}$	$[H]$	Rozptylová indukčnosti rotoru
Ψ_{rd}	$[Wb]$	Magnetický tok rotoru
ω_1	$[rad.s^{-1}]$	Synchronní úhlová rychlost
ω_r	$[rad.s^{-1}]$	Rotorová úhlová rychlost
J	$[kg.m^2]$	Moment setrvačnosti
v	$[^{\circ}]$	Úhel natočení
M_m	$[Nm]$	Moment motoru
M_a	$[Nm]$	akcelerační moment

Úvod

V několika posledních desetiletích se řídicí systémy elektrických pohonů začali objevovat v široké oblasti všech možných aplikací průmyslu. Jako elektrické motory se v pohonech dnes používají téměř výhradně střídavé motory asynchronní nebo synchronní, stejnosměrné motory se objevují v praxi již jen ojediněle a to z důvodu vysoké pořizovací ceny a vyšších nároků na údržbu, avšak nevýhodou motorů střídavých je mnohem větší náročnost na řídicí systém. Jako řídicí systémy se používají hlavně konvenční metody založené na kaskádní struktuře PI regulátorů. V poslední době se začínají objevovat v akademických a vědeckých oblastech metody řízení založené na prediktivním přístupu.

Prediktivní řízení kvůli svým vlastnostem je velmi používanou metodou v různých průmyslových oblastech. V chemickém průmyslu se kvůli dlouhým časovým konstantám a tím nízkým nárokem na výpočetní výkon po procesorech používají prediktivní metody již od 70. let minulého století. Ve výkonové elektronice bylo dlouhou dobu použití predikčních metod omezeno vysokou výpočetní náročností pro vysokou spínací frekvenci, ale kvůli rychlému pokroku ve výpočetní technice, kde mikroprocesory v posledních letech vykazují velmi vysoký výkon za příznivou cenu, začínají se objevovat řídicí metody využívající právě prediktivní přístup.

Tato práce se zabývá konkrétní metodou prediktivního řízení založenou na modelu FCS-MPC(Finite Control Set Model Predictive Control).

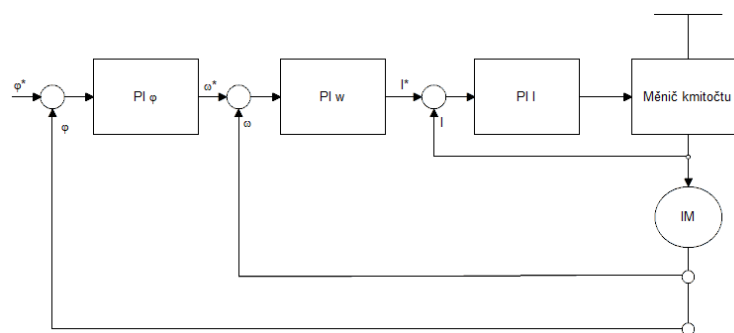
V první části je popsána obecná metoda prediktivního řízení MPC a následně podrobně popsána konkrétní metoda FCS-MPC. Ve druhé části jsou popsány všechny hardwarové části použité k praktické demonstraci prediktivní metody. Ve třetí části je provedena simulace metod FCS-MPC proudu a otáček, které jsou upravené do několika modifikací a jednotlivě porovnány. V poslední části je prakticky předvedena metoda FCS-MPC s využitím dvou výpočetních procesorových jader a dvou výpočetních koprocessorů CLA mikrokontroléru TMS320F28377D pro dvě metody FCS-MPC pro jeden a dva predikční kroky.

1 Teorie

1.1 Řídící systémy ve výkonové elektronice

První regulátory pro elektrické pohony se začali objevovat v polovině minulého století ve formě analogových obvodů. V 90. letech analogové obvody vystřídali diskretní regulátory ve formě mikroprocesorů, které jsou široce používané dodnes. Regulačních systémů elektrických pohonů existuje velké množství, pro střídavé stroje se nejčastěji používají lineární regulátory založené na kaskádní struktuře z PI/PID regulátorů, konkrétně se jedná o vektorové řízení FOC(field oriented control) a přímé řízení momentu DTC(direct torque control). Nežádoucí nelinearity motorů jsou kompenzovány pomocí před výpočty nebo feedforward kompenzací.

Důvod proč by se měli používat jiné metody řízení, když kaskádní zapojení PI regulátorů společně s dodatečnými výpočty nelinearit plně dostačují pro regulaci všech moderních elektrických motorů, je z rychlosti regulace podřízených smyček vzhledem k nadřazeným. Z důvodu dobrých regulačních vlastností by měl být podřízený regulátor 5-10x rychlejší, než nadřazený regulátor. Pokud se jedná o kaskádní regulaci se třemi PI regulátory, který se skládá z regulátoru polohy, regulátoru rychlosti a regulátoru proudu viz obr. 1.1. U takovéto struktury zapojení regulátorů, musí být regulátor proudu 50-100x rychlejší než regulátor polohy. Při požadavku na rychlou odezvu regulátoru polohy nemusí být proveditelná regulace proudu z hlediska dosažitelné rychlosti odezvy. Prediktivní řízení je značně výpočetně náročnější v porovnání s jednotlivými PI regulátory, ale v porovnání s celou kaskádní strukturou PI regulátorů může být výpočetní náročnost nižší.[1]



Obrázek 1.1: Blokové schéma tří regulátorů typu PI zapojené v kaskádní struktuře

Další nevýhodou kaskádní struktury je, že metody využívají pouze současný a díky integrátoru stav systému minulý, prediktivní metody využívají navíc stavy systému budoucí.

Prediktivní principy řízení lze přirovnat k lidskému chování při většině činnostech. Jako příklad lze vzít hraní šach, kde zkušený hráči předtím než udělají tah, tak se snaží

předvídat co nejvíce možných tahů co nejdál do budoucnosti. Pokud není blízko konci hry, je nemožné předvídat všechny možné tahy do konce hry, proto hráč vybere tah, ve kterém věří, že je nejlepší(optimální) pomocí predikce na co největším horizontu. Jakmile protivník provede tah, který je nepředvídatelný, hráč musí provést od znova všechny možné predikce co nejdál do budoucnosti.

1.2 Metody Prediktivního řízení

Ačkoli byla teorie MPC uvedena již v 60. letech minulého století. První aplikace se objevily v 70. letech v chemickém průmyslu, kde díky dlouhým časovým konstantám chemických procesů lze regulovat i pomocí pomalého procesoru. Naopak elektrické pohony, jenž disponují krátkými časovými konstantami se začali praktické aplikace objevovat až v 80. letech a to jen pro velké výkony a malé spínací frekvence.

Hlavním znakem všech metod prediktivního řízení je použití matematického modelu systému pro predikování chování řízených proměnných v budoucnu. Tyto informace jsou regulátorem použity k získání optimálního ovládání podle daných kritérií.

Metoda MPC není konkrétní metoda řízení, ale zahrnuje velkou oblast určitých metod prediktivního řízení. Všechny metody MPC využívají model systému k získání řídicího signálu, který dovede systém do požadovaného stavu. MPC vybírá ze vstupních sekvencí tu nejlepší přes celý budoucí horizont podle určitého kritéria. Nejlepší z první sekvence je aplikován na systém a celý proces je opakován každou periodu od znova. MPC používá matematický model systému, který umožňuje regulátoru predikovat dopad jednotlivých akčních zásahů.

rozdělení prediktivních řízení založených na modelu

Obecně lze rozdělit MPC metody používané ve výkonové elektronice na

1. Implicitní | explicitní
2. CCS-MPC | FCS-MPC

explicitní MPC řeší optimalizační problémy offline, tzn. před nasazením regulátoru v určité aplikaci je nejprve potřeba nasimulovat co nejvíce možných stavů systému a následně je uložit do paměti tzv. look up tabulky(LUT). U metod implicitních se optimalizace řeší online v průběhu regulace. Nevýhodou implicitních metod je vyšší výpočetní náročnost po procesoru, ale je schopen řídit systém s více stavovými proměnnými. U explicitních metod navíc může nastat problém s vysokým nárokem po velikosti paměti pro look-up tabulku. Dalším kritériem pro rozdělení může být podle použití modulátoru. MPC s nekonečným počtem akčních zásahů(CCS-MPC) využívá ke generování spínacích signálů

modulátor SV-PWM(Space Vector Pulse Width Modulation) a generuje tak spínací kombinace s konstantní frekvencí, které lze z pohledu modulátoru přirovnat k vektorovému řízení(FOC).

Druhým typem MPC je metoda s konečným počtem akčních zásahů FCS-MPC(Finite Control Set Model Predictive Control), která nevyžaduje modulátor a spínací kombinace může přicházet přímo na měnič kmitočtu v nezměněném tvaru(samozřejmě se započítáním mrtvých časů). Jak už z názvu vyplývá metoda FCS-MPC využívá výhody konečných počtu možných kombinací sepnutí měniče kmitočtu. Tento počet spínacích kombinací přímo určuje počet predikovaných stavů systému. Díky konečnému počtu sepnutí a dnes i dostupnosti velmi rychlých digitálních signálových procesorů DSP(digital signal processor) je možné použít implicitní metodu FCS-MPC na vysokých frekvencích, které jsou požadkem po moderních pohonech.

1.3 FCS-MPC ve výkonové elektronice

Cílem je výběr vhodného akčního zásahu $S(t_k)$, který bude řídit obecnou systémovou proměnnou $x(t_k)$ co nejlíže požadované proměnné $x^*(t_k)$ jako na obr. 1.2 kde měření, výpočty a řídicí zásah jsou provedeny okamžitě(prakticky nemožné). Jelikož počet akčních zásahů S_i je konečný počet $i=1\dots n$, lze je vyhodnocovat společně s měřenou proměnnou $x(t_k)$ pro všechny možné kombinace pomocí funkce

$$x_{pi}(t_{k+1}) = f_p \{x(t_k), S_i\}$$

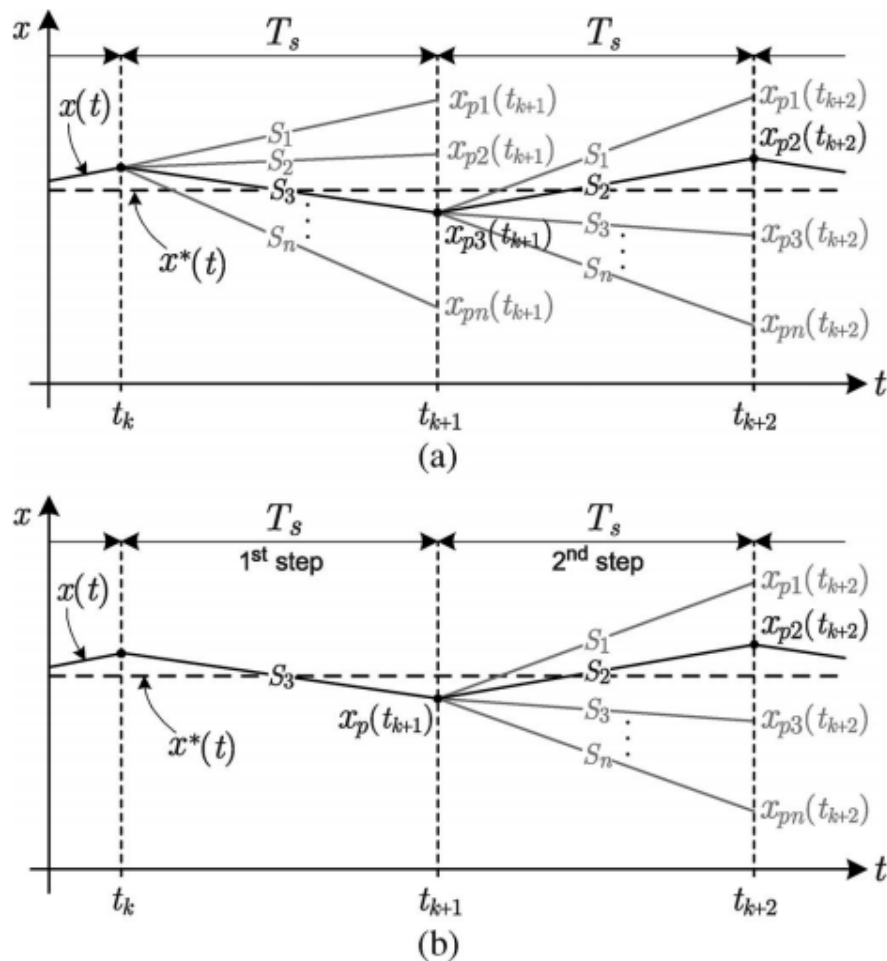
kde $i=1\dots n$, funkce f_p je získána z matematického modelu systému. K určení nejlepšího akčního zásahu slouží tzv. ztrátová nebo váhová funkce f_g ve tvaru

$$J_i(t_{k+1}) = f_g \{x^*(t_{k+1}), x_{pi}(t_{k+1})\}$$

Výsledkem je n rozdílných ztrátových hodnot, ze kterých se vybere minimální(optimální) a ta se aplikuje jako akční zásah pro měnič kmitočtu.

$$\min(J_i), \text{ pro } i = 1\dots n$$

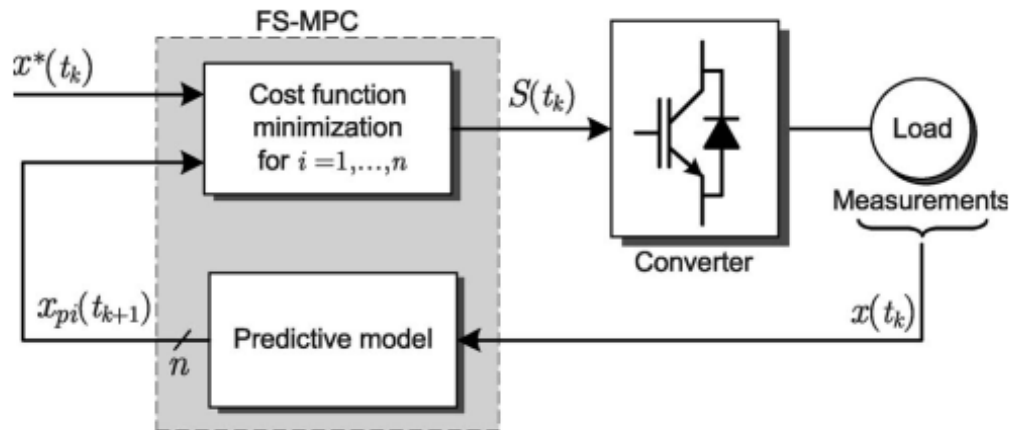
Jak je zobrazeno na obr. 1.2(a), kde predikovaná hodnota $x_{p3}(t_{k+1})$ je nejlíže referenční hodnotě $x^*(t_{k+1})$ a proto je jako akční zásah vybrána hodnota S_3 v čase t_k (obr.1.2(b)). A v další periodě je celý proces proveden od znova.



Obrázek 1.2: Predikční horizont metody FCS-MPC.[2]

Zatímco klasické způsoby řízení založené na kaskádních regulátorech nelinearity zanedbávají, prediktivní metody je přímo začleňují v regulaci současně s hlavními stavy systému. Další výhodou prediktivních metod je, že dokážou regulovat systémy typu MIMO (Multiple Input Multiple Output) stejně dobře jako SISO (Single Input Single Output) systémy.

Obecné schéma prediktivní metody FCS-MPC společně s měničem kmitočtu (Converter) a zátěží (Load) je na obr. 1.3. Měnič kmitočtu může být různých topologií s různým počtem fází (většinou dvou-úrovňový třífázový). Zátěž může představovat asynchronní nebo synchronní motor. Měřené veličiny $x(t_k)$ jsou použity v matematickém modelu k výpočtu predikčních hodnot $x(t_{k+1})$ pro každý z n možných spínacích kombinací. Všechny různé možné spínací kombinace pro dvou-úrovňový třífázový měnič kmitočtu, lze vyjádřit jako vektory napětí uvedené v tabulce na obr. 1.6. Tyto predikované hodnoty jsou porovnány s referenčními hodnotami a současně s tím mohou být provedena omezení a následně podle určitých podmínek vybrán optimální vektor napětí, který je následně generován ve formě spínací kombinace v měniči kmitočtu pro následující periodu.[2]

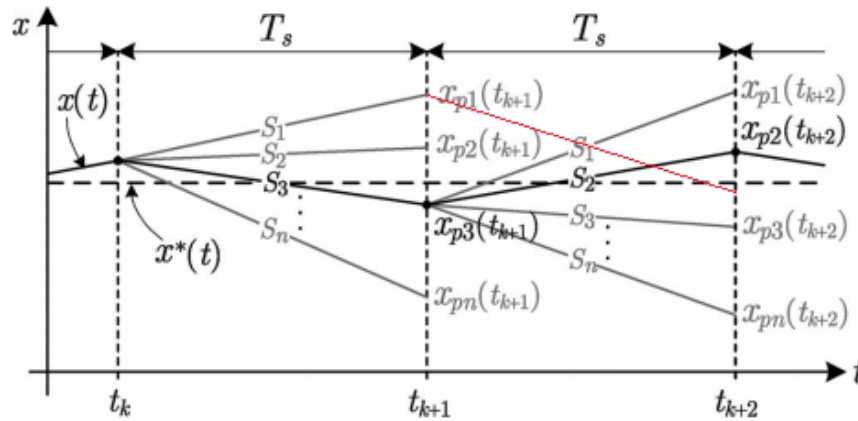


Obrázek 1.3: Obecné blokové schéma regulační metody FCS-MPC [2]

1.3.1 Prediktivní horizont

Výhodou metod MPC je možnost predikování dále než jen na jeden horizont do budoucnosti. Na obr. 1.4 je v čase $x(t_{k+1})$ vybrána jako optimální hodnota $x_{p3}(t_{k+1})$, ale pokud bude provedena predikce pro všechny hodnoty z t_{k+1} může vyjít optimálnější výsledek pro jiný vektor $x_{pn}(t_{k+1})$. Na obr. 1.4 je ukázána predikce v čase t_{k+2} pro nejlepší výslednou hodnotu $x_{p3}(t_{k+1})$ z času $x(t_{k+1})$, ale pokud se provedou predikce pro všechny hodnoty $x_{pn}(t_{k+1})$, může mít výsledná hodnota $x_{pn}(t_{k+2})$ lepší výsledek v porovnání s výsledky optimálního vektoru $x_{p3}(t_{k+1})$. Na obr. 1.4 je ukázán červeně výsledek nejlepší predikce hodnoty $x_{p1}(t_{k+1})$, kde výsledek hodnoty je blíže požadavku než jakákoliv predikce hodnoty $x_{p3}(t_{k+1})$.

Nevýhodou více predikčních kroků je ve výpočetní náročnosti, která roste exponenciálně s dalším predikčním krokem. Například u dvou-úrovňového třífázového napěťového střídače je k dispozici 7 vektorů pro první predikční horizont, tak v případě druhého predikčního horizontu je potřeba pro každý vektor z prvního predikčního horizontu vypočítat dalších 7 vektorů a tím je výsledný počet vektorů rovný 56. Výpočetní náročnost roste exponenciálně s každým dalším PH(predikčním horizontem).



Obrázek 1.4: Průběh predikovaných proměnných pro PH = 2 [2]

1.4 Měníč kmitočtu

Základním prvkem měniče je výkonový spínač, kterým je nejčastěji IGBT nebo MOSFET tranzistory. Nejjednodušší model výkonového spínače se uvažuje s ideálním spínáním ve dvou stavech: sepnutý a vypnutý. Proto je celkový počet možných spínacích stavů výkonového měniče kmitočtu roven počtu rozdílných kombinací dvou spínacích stavů každého spínače, ačkoli některé kombinace dají stejný výsledek, například kombinace, které způsobí na zátěži nulové napětí. Celkový počet sepnutí pro různé měniče kmitočtu je dán

$$N = x^y$$

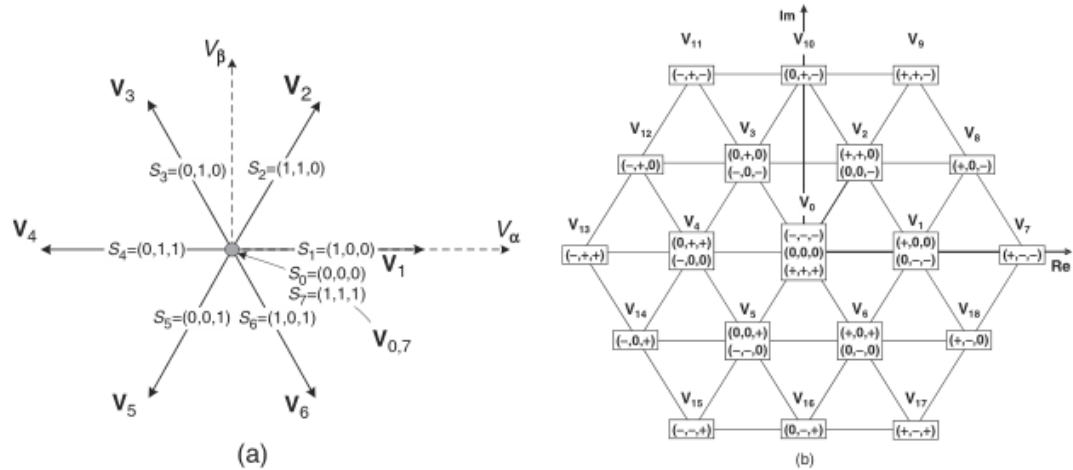
kde x je počet možných stavů v jednotlivých fázích a y je počet fází měniče. Například pro dvou-úrovňový třífázový měnič kmitočtu je celkový počet kombinací dán podle

$$N = 2^3 = 8$$

a pro tří-úrovňový třífázový měnič je celkový počet spínacích kombinací dán

$$N = 3^3 = 27$$

Na obr. 1.5 jsou vyznačeny prostorové vektory napětí pro třífázový (a)dvou-úrovňový a (b)tří-úrovňový měnič kmitočtu. Některé spínací kombinace generují stejný vektor napětí. U dvou-úrovňového měniče je 8 možných spínacích kombinací, ale pouze 7 vektorů generuje různé vektory napětí, dva vektory vytvářejí nulový vektor. Tří-úrovňový měnič vytváří 27 spínacích kombinací s 19 vektory o různé velikosti napětí.[3]



Obrázek 1.5: Prostorový vektor pro dvou-úrovňový(a) a tří-úrovňový(b) měnič kmitočtu

Výpočet jednotlivých prostorových vektorů je dán vztahem

$$\bar{x} = k \cdot (x_a + \bar{a} \cdot x_b + \bar{a}^2 \cdot x_c)$$

kde k je transformační konstanta a její hodnota se volí $k = \frac{2}{3}$ a hodnota $\bar{a} = e^{j \cdot 120^\circ}$

S_1	S_2	S_3	S_4	S_5	S_6	výsledný vektor napětí
1	0	0	1	0	1	$U_1 = \frac{2}{3} \cdot U_{dc}$
1	0	1	0	0	1	$U_2 = \frac{1}{3} \cdot U_{dc} + j \cdot \frac{\sqrt{3}}{3} \cdot U_{dc}$
0	1	1	0	0	1	$U_3 = -\frac{1}{3} \cdot U_{dc} + j \cdot \frac{\sqrt{3}}{3} \cdot U_{dc}$
0	1	1	0	1	0	$U_4 = -\frac{2}{3} \cdot U_{dc}$
0	1	0	1	1	0	$U_5 = -\frac{1}{3} \cdot U_{dc} - j \cdot \frac{\sqrt{3}}{3} \cdot U_{dc}$
1	0	0	1	1	0	$U_6 = \frac{1}{3} \cdot U_{dc} - j \cdot \frac{\sqrt{3}}{3} \cdot U_{dc}$
1	0	1	0	1	0	$U_7 = 0$

Obrázek 1.6: Kombinace sepnutí a výsledné vektory napětí

1.5 Ztrátová funkce

Ztrátová funkce je základním pilířem prediktivního řízení a vychází z porovnání prediktivních a referenčních proměnných. Jako příklad těchto proměnných může být proud, moment, výkon, rychlost atd. Ztrátová funkce je obecně zapsána jako rozdíl mezi predi-

kovanou a referenční proměnnou ve tvaru

$$J = \| x_{ref}(t_{k+1}) - x(t_{k+1}) \|$$

kde $x(t_{k+1})_{ref}$ je referenční proměnná a $x(t_{k+1})$ je predikovaná hodnota vypočítaná z matematického modelu pro určitou spínací kombinaci. Dvojitě závorky jsou v regulátoru nahrazeny absolutní hodnotou, kvadrátem nebo sumou rozdílů.

$$J = | x_{ref}(t_{k+1}) - x(t_{k+1}) |$$

$$J = (x_{ref}(t_{k+1}) - x(t_{k+1}))^2$$

$$J = \sum_{t_k}^{t_{k+1}} (x_{ref}(t_{k+1}) - x(t_{k+1}))$$

Absolutní rozdíl a rozdíl kvadrátů dává podobný výsledek chyby. Pokud jsou ve ztrátové funkci dva a více různých parametrů, vykazuje ztrátová funkce s kvadrátem rozdílů lepší výsledek. Ztrátová funkce v součtovém tvaru započítává rozdíl proměnných mezi časy $x(t_k)$ a $x(t_{k+1})$ a ne pouze aktuální predikovanou proměnnou $x(t_{k+1})$. Střední hodnota chyby součtové metody je v porovnání s předchozíma dvěma metodami menší a tím vede k vyšší přesnosti metody, nevýhodou je pak vyšší výpočetní náročnost.[1]

1.6 Zpoždění výpočtů

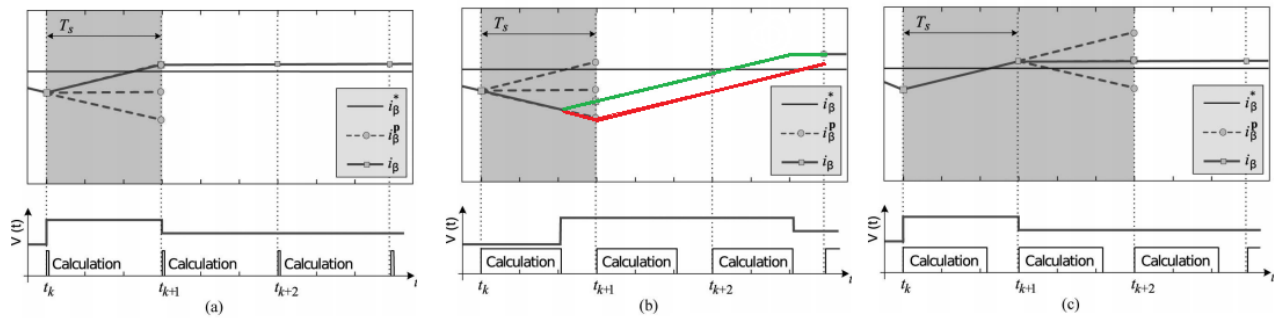
Prediktivní řízení vyžaduje velké množství výpočtů v krátké periodě, proto je třeba uvažovat časové zpoždění mezi měřením a akčním zásahem na měnič kmitočtu.

Metoda, která byla vysvětlena v kapitole 1.3 je metodou bez kompenzace zpoždění a tento proces lze popsat následující sekvencí

1. Měření hodnot proudů statoru
2. Výpočet predikčních hodnot pro t_{k+1}
3. Vyhodnocení ve ztrátové funkci pro všechny kombinace měniče kmitočtů
4. Výběr spínací kombinace s nejmenší ztrátovou funkcí
5. Aplikace vybrané spínací kombinace na měnič kmitočtu

Na obrázku 1.7 je průběh proudu I_β pro všechny spínací kombinace, které produkují 3 různé průběhy pro tento proud. Na obr. 1.7(a) je průběh se zanedbáním zpoždění výpočtů, kde je v čase t_k změřeno, vypočítáno a aplikována spínací kombinace s nejmenší ztrátovou funkcí pro čas t_{k+1} . V praxi není tento děj možné dosáhnout a vždy dochází k nějakému zpoždění aplikace vektoru t_{k+1} a pokud je výpočetní čas značný vzhledem k periodě

vzorkování dochází k aplikaci vhodné spínací kombinace až dlouhou po čase t_k obr. 1.7(b). Po dobu zpoždění je v měniči aplikována spínací kombinace z předchozí periody a jak je vidět z obr. 1.7(b) zelená křivka proudu se může vzdalovat od referenční hodnoty. Důsledkem zpoždění dochází ke zvětšení zvlnění proudu a jiných na proudu závislých parametrech. Druhou variantou, která se v mikroprocesorové technice používá a to pomocí tzv. shadow registrů, kde se aplikuje optimální vektor napětí až na začátku následující periody v čase t_{k+1} , na obr. 1.7(b) je označen tento průběh červenou křivkou.[3]



Obrázek 1.7: Průběhy predikovaných proměnných, kde okamžitý výpočet(a) skutečný výpočet(b) a skutečný výpočet s kompenzací(c)[3]

1.6.1 Kompenzace zpoždění

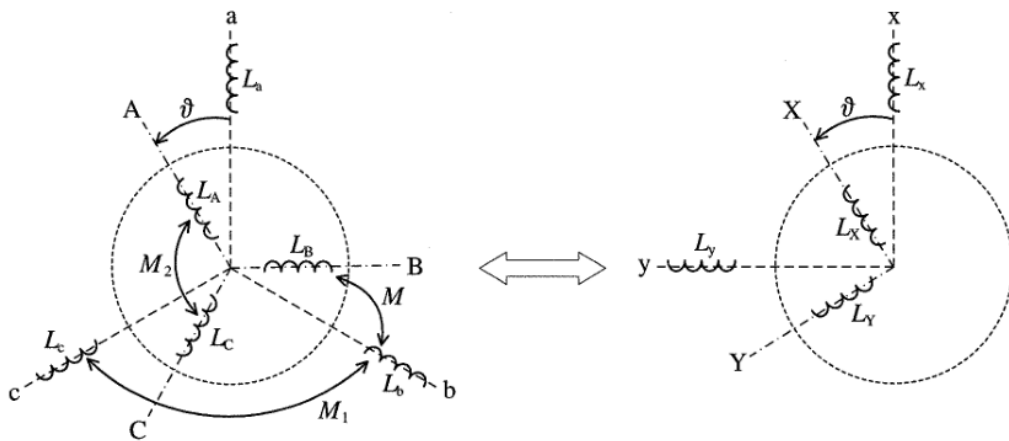
Eliminace zpoždění výpočtů, lze provést pomocí jednoduché úpravy, kde se vybraná spínací kombinace aplikuje v následující periodě na začátku podle následující sekvence

1. Měření hodnot proudů statoru
2. Aplikace spínací kombinace(vypočítaná v předchozím intervalu)
3. Výpočet pro kombinaci právě aplikovanou v měniči
4. Výpočet predikčních hodnot pro t_{k+2}
5. Vyhodnocení ve ztrátové funkci pro všechny kombinace měniče kmitočtů pro t_{k+2}
6. Výběr spínací kombinace s nejmenší ztrátovou funkcí.

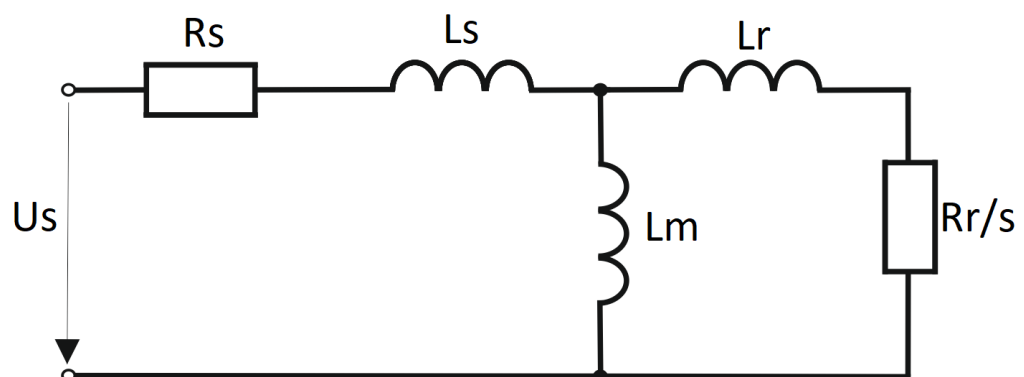
Průběh proudu I_β po provedení kompenzace je zobrazen na Obr. 1.7(c). V čase t_k je změřen proud a téměř současně aplikována spínací kombinace t_{k+1} , která byla v předěšlé periodě vypočítaná jako spínací kombinace pro t_{k+2} . Tato úprava způsobí zvýšení výpočetní náročnosti, kvůli výpočtu matematického modelu pro jeden spínací vektor, ale průběh proudu se může více vyhladit.

1.7 Matematický model asynchronního motoru

Asynchronní motor je tvořen dvěma vícefázovými, obvykle třífázovými systémy vinutí. Jeden systém je umístěn na statoru a druhý na rotoru AS motoru. Všechna vinutí jsou vzájemně vázána magnetickými indukčnostmi. Tyto vzájemné indukčnosti umístěné na statoru a na rotoru jsou časově proměnné z důvodu vzájemného otáčení rotoru vůči statoru. Z důvodu této složitosti se pro chování a analýzu stavu hledali způsoby, jak celý systém matematického popisu zjednodušit. Nejčastější způsob je transformace třífázového systému vinutí do systému dvou vzájemně kolmých vinutí. Pomocí transformace se nejen sníží počet fází systému ze tří na dvě, ale za vhodných okolností přejdou střídavé veličiny na veličiny stejnosměrné a některé parametry soustavy se změny ze střídavých na stejnosměrné.



Obrázek 1.8: Transformace třífázových systémů do dvoufázových systémů[4]



Obrázek 1.9: Náhradní schéma jedné fáze asynchronního motoru

Asynchronní motor lze popsat v ustáleném stavu dvěma rovnicemi pro napětí a dvěma

pro magnetický tok statoru a rotoru pro každou fázi

$$u_s = R_s \cdot i_{sa} + \frac{d\psi_s}{dt}$$

$$u_r = R_r \cdot i_{ra} + \frac{d\psi_r}{dt}$$

$$\Psi_s = L_s \cdot i_s + L_m \cdot i_r$$

$$\Psi_r = L_m \cdot i_s + L_r \cdot i_r$$

1.7.1 Transformace souřadnic

Vzájemné indukčnosti vinutí statoru a rotoru se mění v čase z důvodu otáčení rotoru vůči statoru. Těmto vzájemným indukčnostem vychází 6 rovnic pro napětí a 6 rovnic pro magnetický tok, které se mění v závislosti na úhlu otočení statoru vůči rotoru. Pro zjednodušení výpočtů se veličiny transformují do ortogonálních souřadnic.

Prvním typem transformace je transformace do systému pevně svázaného se státorem, která je označována jako transformace Clarkova nebo ve zkratce $\alpha\beta$. Reálná osa těchto souřadnic se označuje α a leží v ose vinutí "a" a imaginární osa se poté označuje β .

Transformační vztah z proudů i_a, i_b, i_c na proudy i_α, i_β je dán vztahem

$$i_\alpha = \frac{2}{3} \left(i_a - \frac{1}{2} i_b - \frac{1}{2} i_c \right)$$

$$i_\beta = \frac{2}{3} \left(\frac{\sqrt{3}}{2} i_b - \frac{\sqrt{3}}{2} i_c \right)$$

Nahrazení tří okamžitých skalárních veličin jednou vektorovou veličinou se nazývá tzv. prostorový vektor. Prostorový vektor lze použít nejen pro proudy, ale i pro jiné veličiny motoru jako je napětí nebo magnetické toky. Potom soustavu šesti napěťových rovnic lze nahradit dvěma vektorovými rovnicemi, jednu pro stator a druhou pro rotor. Tato transformace je vhodná pro sledování statorových veličin. U této transformace dojde ke snížení počtu fází systému na dvě, ale veličiny stále zůstávají střídavé.

Vektory popsané v Clarkově transformaci převedené do systému os $\alpha\beta$, které jsou pevně svázané se státorem lze snadno převést do jiného souřadnicového systému os, který rotuje libovolnou rychlostí vzhledem k původnímu systému os $\alpha\beta$. Tento systém obecně označíme jako systém x,y Rychlost otáčení systému x,y je ω_k . Pokud přiřadíme systém os $\alpha\beta$ do systému x,y je rychlost $\omega_k=0$.

Druhou transformací je transformace do systému os svázaných s rotorem. Systém se potom otáčí rychlostí $\omega_k = \omega_r$. Tato transformace je vhodná pro sledování rotorových veličin.

Třetí transformací je transformace do systému os rotujících rychlosti točivého magnetického pole motoru. Tento systém se značí jako Parkova transformace nebo ve zkratce dq. Systém dq se otáčí synchronní rychlostí, tedy $\omega_k = \omega_1$. Systém dq je vhodný pro sledování regulačních dějů.[4]

1.7.2 Matematický model v dq

Transformace proudů I_a, I_b, I_c do systému dq, kde se nejprve provede transformace do $\alpha\beta$ systému a následně transformace do systému dq pomocí transformace Parkovy

$$i_d = I_\alpha \cos(v) + I_\beta \sin(v)$$

$$i_q = -I_\alpha \sin(v) + I_\beta \cos(v)$$

kde v je úhel mezi magnetickými toky statoru a rotoru.

Pro výpočet jak aktuálních, tak hodnot budoucích v simulacích i v reálné aplikaci provedené v mikrokontroléru, je třeba provádět výpočty v diskretním čase. Pro převod do diskretního času byla zvolena Eulerova diskretizační metoda. Rovnice podle předešlého předpisu pro výpočet matematického modelu asynchronního motoru jsou následující

$$\frac{di_{sd}}{dt} = -k_1 \cdot i_{sd} + \omega_s \cdot i_{sq} + k_2 \cdot \frac{\psi_{rd}}{L_m} + k_4 \cdot u_{sd}$$

$$\frac{di_{sq}}{dt} = -\omega_s \cdot i_{sd} - k_1 \cdot i_{sq} - k_3 \cdot \omega_m \cdot \frac{\psi_{rd}}{L_m} + k_4 \cdot u_{sq}$$

$$\frac{d\Psi_{rd}}{dt} = \frac{L_m}{T_r} \cdot i_{sd} - \frac{1}{T_r} \cdot \Psi_{rd}$$

Konstanty z předchozích rovnic jsou vyjádřené následujícími tvary

$$k_1 = \frac{1}{\sigma \cdot T_s} + \frac{1 - \sigma}{\sigma \cdot T_r}$$

$$k_2 = \frac{1 - \sigma}{\sigma \cdot T_r}$$

$$k_3 = \frac{1}{\sigma \cdot L_s}$$

$$k_4 = \frac{1 - \sigma}{\sigma}$$

$$T_s = \frac{L_s}{R_s}$$

$$T_r = \frac{L_r}{R_r}$$

Souřadný systém dq je pevně svázán s vektorem magnetického toku, který leží v ose d a odpadá tím výpočet magnetického toku v ose q .

Rovnice pro akcelerační moment je dána

$$M_a = J \cdot \frac{d\omega}{dt} = M_m - M_z$$

kde zátěžný moment M_z se uvažuje nulový a moment motoru M_m je díky této úvaze dán následujícím tvarem

$$M_m = \frac{3}{2} \cdot pp \cdot \frac{L_m}{L_r} \cdot \Psi_{rd} \cdot I_{sq}$$

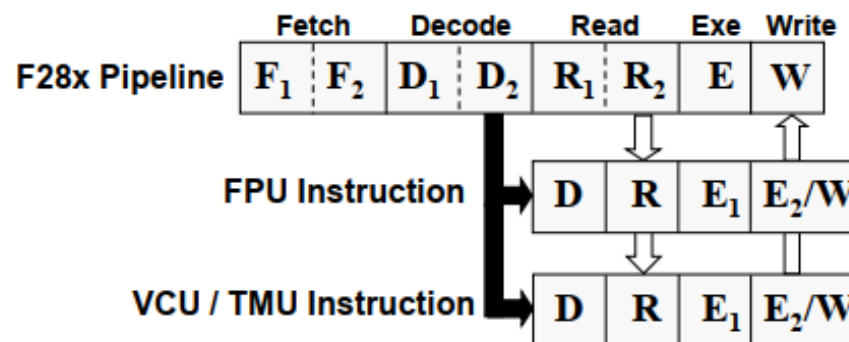
a nakonec lze odvodit požadovanou derivaci otáček podle

$$\frac{d\omega_m}{dt} = \frac{M_m}{J}$$

2 Hardware

2.1 Mikroprocesor

V této práci byly provedeny simulace pro metody prediktivního řízení pro jeden a dva predikční kroky. Pro jeden predikční krok není problém implementace na jakýkoliv mikroprocesor určený k regulaci, kde se vypočítává matematický model pouze pro 7 vektorů napětí (v případě dvou-úrovňového měniče). Problém nastává při implementaci řízení na dva predikční kroky, kde je potřeba vypočítat matematický model nejen pro sedm vektorů z prvního predikčního kroku, ale také dalších 48 vektorů pro druhý predikční krok. Celkem je třeba vypočítat 57 vektorů napětí pro prediktivní řízení s dvěma predikčními horizonty. Pro takovéto množství potřebných výpočtů v krátkém časovém intervalu (obvykle v rámci stovek us) je potřeba výkonný mikroprocesor. Pro tuto práci byl vybrán mikroprocesor od firmy Texas Instruments s označením TMS320F28377D. Mikroprocesor disponuje 2 CPU jádry s 200 MHz taktem. Každé z těchto jader obsahuje několik výpočetních akceleratorů a koprocesor CLA (Control Law Accelerator), které slouží ke zvýšení výpočetního výkonu jádra mikrokontroléru. Architektura paměti je modifikovaná Harvardská architektura a procesor je osmistupňový řetězec RISC, tato architektura na obr. 2.1.



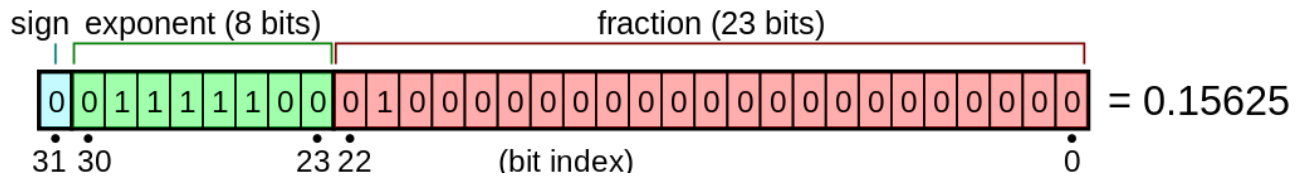
Obrázek 2.1: blokové schéma osmistupňového pipeline řetězce[5]

2.1.1 Floating Point Unit (FPU)

Jádro procesoru provádí výpočty pouze v pevné řádové čárce a mnoho výpočtů v programu je potřeba provádět v plovoucí řádové čárce, je proto k dispozici koprocesor FPU, který slouží k výpočtům v pohyblivé řádové čárce podle standardu IEEE 754 (formát single precision plovoucí řádové čárky). Díky FPU jednotce odpadá potřeba normování proměnných do pevné řádové čárky a tím nebezpečí vzniku chyb, jako je přetečení nebo podtečení.

Norma IEEE 754 pro základní přesnost "single" zabírá v paměti 32 bitů. Na obr 2.2 je rozložení jednotlivých bitů pro tento formát. Na nejvyšším je bit, který určuje znaménko

daného čísla(sign), 8 bitů označeny zeleně je exponent čísla a zbylých 23 bitů označených červeně je mantisa(fraction).



Obrázek 2.2: číslo 0.15625 reprezentované v single-precision IEEE 754-1985 floating-point number [6]

Vztah pro vyjádření čísla v jednoduché přesnosti je dán

$$X = -1^{zn} * 2^{(exp-127)} * 1.mantisa$$

Matematické operace, které lze pomocí FPU provádět jsou následující

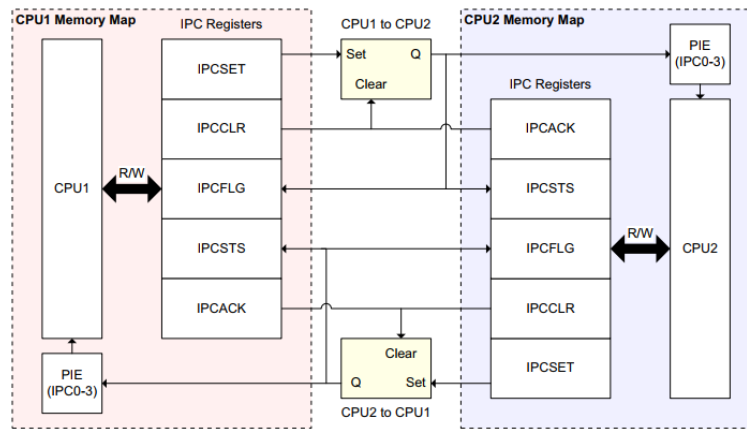
1. součet
2. rozdíl
3. součin
4. podíl
5. zbytek po dělení
6. druhá odmocnina
7. porovnání

2.1.2 Trigonometric Math Unit(TMU)

TMU je stejně jako FPU jednotka určená pro výpočty v pohyblivé řádové čárce, ale na rozdíl od FPU je určená pouze pro určité typy matematických operací. Mezi tyto operace patří sin, cos, tan, arctan, dělení, odmocnina a jiné. Pro tyto typy operací je mikroprocesor průměrně 1.4x násobně rychlejší než mikroprocesor bez TMU jednotky.

2.1.3 Inter Processor Communication (IPC)

Mezi dvěma jádry slouží komunikace IPC, kde existují 2 různé způsoby komunikace Global shared RAM a IPC Message RAM.



Obrázek 2.3: blokové schéma IPC komunikace [5]

2.1.4 Global Shared RAM

Zařízení TMS320F28377D obsahuje 16 bloků global shared RAM (GS0-GS15) o velikosti 4 kB, které jsou určeny pro použití jako paměť programu nebo dat, ale lze je nastavit i pro komunikaci mezi procesory. Po resetu jsou všechny bloky vlastněny CPU1, který může z těchto pamětí číst, zapisovat i konfigurovat, CPU2 může z těchto pamětí pouze číst. Jednotlivé bloky pamětí lze nakonfigurovat pomocí určitých registrů v CPU1 po resetu do vlastnictví CPU2, poté CPU2 může do těchto bloků i zapisovat.

2.1.5 IPC message RAM

Lepším způsobem jak mezi procesory komunikovat je pomocí IPC message RAM, která se skládá ze 2 paměťových bloků o velikosti 1kB. Na rozdíl od shared memory, tyto paměti jsou určeny pouze pro jeden směr a tento směr nemůže být žádným způsobem konfigurován. Paměť pojmenovaná jako CPU1 to CPU2, do které může pouze CPU1 a u pamětí CPU2 to CPU1 pouze CPU2 zapisovat. Číst mohou z obou pamětí oba procesory, ale pokud přistupují k jedné paměti obě jádra najednou má prioritu přístupu jádro s možností zápisu.

2.1.6 IPC události a přerušení

Přes IPC jednotku lze vyvolávat 32 IPC událostí z CPU1 do CPU2 a obráceně, tyto signály lze odeslat do Flag registru. Čtyři z těchto signálů (IPC0 až IPC3) lze nakonfigurovat pro generování přerušení na vzdáleném jádře.

2.2 Control Law Accelerator (CLA)

CLA je plně programovatelný nezávislý 32 bitový floating point hardwarový akcelerátor, určený k vykonávání matematických operací paralelně s CPU. Kód pro CLA se skládá

z inicializačního kódu a tasků. Tasky lze přirovnat k přerušení (ISR), ale jakmile se spustí task na CLA, lze vyvolat další task až po skončení právě probíhajícího tasku. Každý task může být spuštěn buď softwarově od CPU nebo pomocí přerušení od periférií.

Komunikace mezi CPU a CLA je prováděná pomocí dvou RAM pamětí. Jedna RAM paměť slouží k odesílání dat z CPU do CLA nazývá se CPUx to CPUx.CLA1, do této paměti může pouze CPU zapisovat. Druhá RAM paměť CPUx.CLA1 to CPU1 slouží k přenosu dat z CLA do CPU a pouze CLA do ní může zapisovat. CLA nemůže přímo komunikovat se vzdáleným CPU ani pomocí periférie DMA (direct memory acces).

CLA jednotka neobsahuje hardwarovou jednotku TMA pro zrychlení výpočtů goniometrických funkcí a tyto funkce se zde vypočítávají pomocí numerické metody Taylorových řad. Protože CLA jednotka nevyužívá stejný postup při výpočtu goniometrických funkcí, musí se v programu využívat knihovna CLAmath.h místo klasické math.h, která je určena pro výpočty v jádrech. Naopak při výpočtu klasických matematických floating point operací zabere výpočet u CLA pouze jeden takt, zatímco u CPU trvá stejná operace 2 takty, kde je potřeba nahrát operaci do FPU jednotky a až v následujícím taktu se provede FPU operace.

Programování CLA

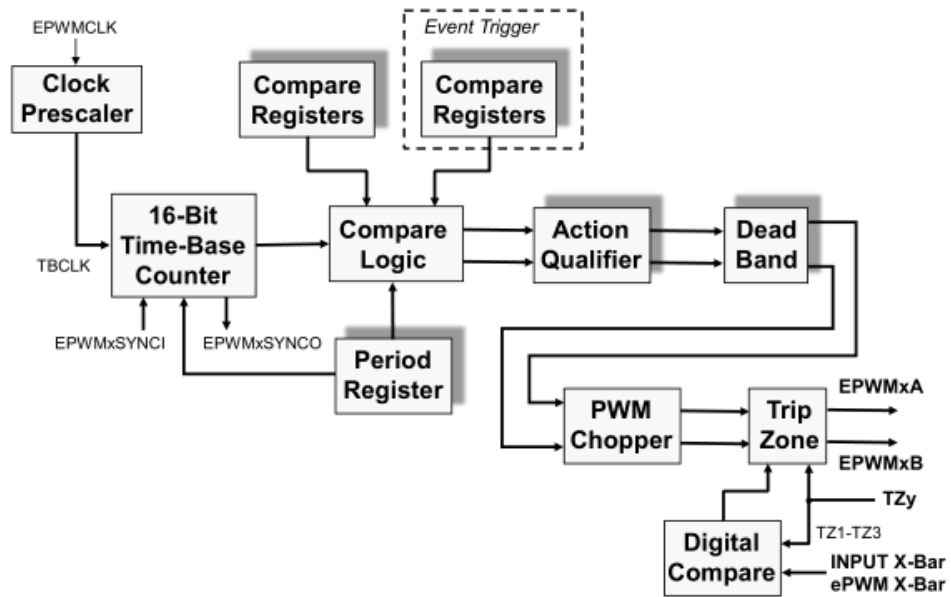
Pro programovou nebo datovou paměť slouží paměťové bloky LS1 až LS5 (každý blok 2K x 16b). Všechny LS paměti jsou po resetu na mapovány na jádro, ze kterého se nastaví bity registru LSxCLAPGM, jestli má být daný blok pro paměť programu (bit CLAPGMLSx = 1) nebo pro paměť data (CLAPGMLSx = 0). Po určení typu paměti se do daných pamětí nakopíruje program a data, poté se paměti nakonfigurují do vlastnictví CLA ko-procesoru přes registr LSxMSEL bitem CLAPGMLSx = 1.

Pro jednotlivé tasky je potřeba nastavit adresy funkcí přes registry MVECT1 - MVECT8. Adresa je 16 bitová a je uložena ve spodní části paměti.

2.3 Periférie

2.3.1 Enhanced Pulse width modulation (ePWM)

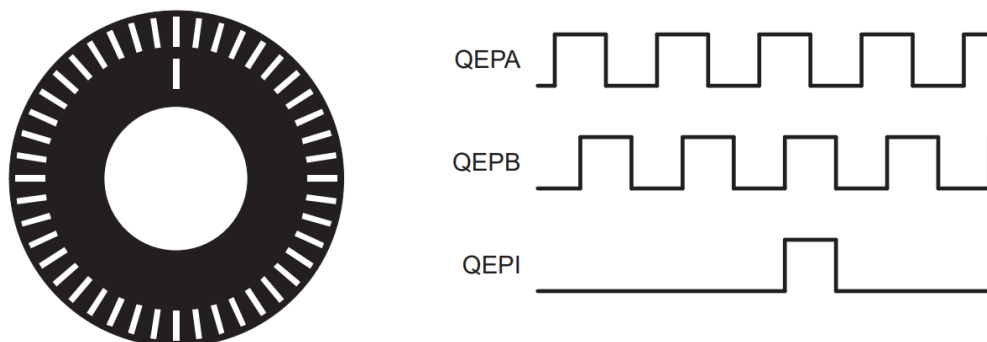
ePWM periférie je určena hlavně ke generování PWM signálů pro měnič kmitočtu. Metoda prediktivního řízení FCS-MPC, která je v této práci použita PWM modulaci nevyužívá. Tato periférie se zde používá k periodickému generování přerušení, kterým je dána perioda řízení FCS-MPC a Generátor mrtvých časů na obr. 2.4 Dead band. Dead band blok vkládá určitý čas mezi vypnutí jedné a zapnutí druhé spínací součástky v jedné větvi měniče kmitočtu, z důvodu zamezení zkratu právě mezi těmito součástkami.



Obrázek 2.4: blokové schéma PWM periférie[5]

2.3.2 Enhanced Quadrature Encoder Pulse(eQEP)

Jednotka eQEP je určená pro převod elektrických pulsů z lineárního nebo rotačního enkodéru pro získání pozice, směru nebo informace o rychlosti z rotačního stroje. Hlavní částí enkodéru je disk, který je připevněný k hřídeli motoru. Světlo vysílané světelným zdrojem většinou LED dioda umístěná před diskem prochází pouze průhlednými okénky v disku. Na druhé straně, než je světelný zdroj, je umístěn optický snímač, který tyto světelné signály převádí na elektrické impulsy ve tvaru obdélníků. Pro zjištění směru otáčení jsou zdroje světla dva, vzájemně posunuté o 90 stupňů QEPA a QEPB. Na disku je umístěno také průhledné okno, které se za otáčku vyskytne pouze jednou například na obr. 2.5 je umístěno blíže ke středu s průběhem vpravo označen jako QEPI. Tento signál slouží k synchronizaci absolutní pozice.



Obrázek 2.5: Optický disk enkodéru[5]

Každou periodou T se vyvolá přerušení QEP jednotky, ve kterém je přečtena a zpra-

cována hodnota z position registru podle

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T}$$

kde ΔX je počet napočítaných elektrický impulsů za periodu T .

Pro metody prediktivního řízení, nebo obecně metody využívající matematický model k řízení je potřeba měřit otáčky současně s ostatními veličinami v každé periodě (na rozdíl od regulátorů založených na kaskádní struktuře jak bylo popsáno v první kapitole). Pro měřené hodnoty otáček je v paměti vyhrazené pole obecně o velikosti N . Právě změřená hodnota $COUNT(t_k)$ je společně s nejstarší hodnotou uloženou v paměti $COUNT(t_{k-N})$ využity k vypočítání rychlosti motoru podle vztahu

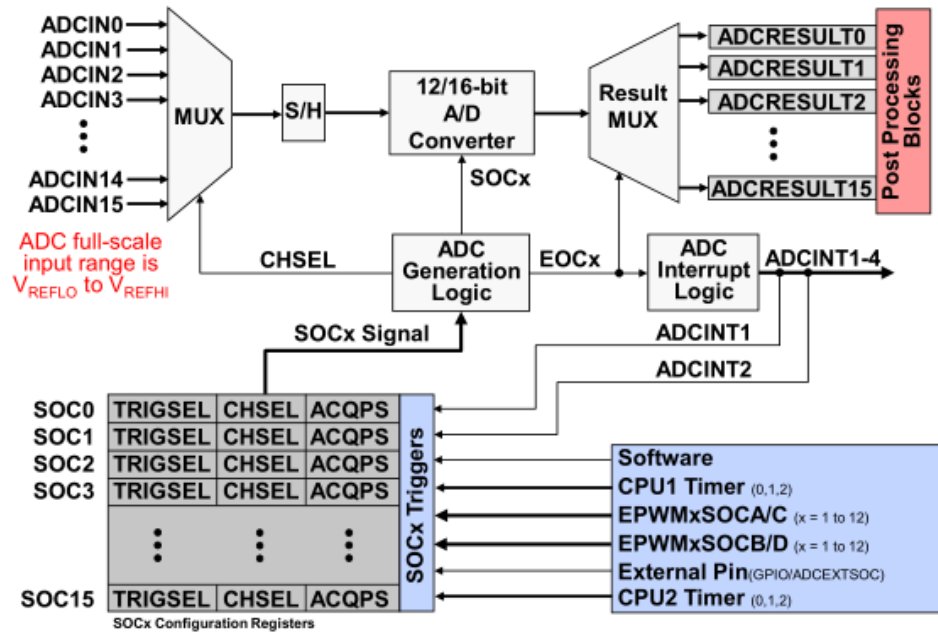
$$f(t_k) = \frac{4096.(COUNT(t_k) - COUNT(t_{k-N}))}{0.002}$$

a následně je nově změřená hodnota $COUNT(t_k)$ uložena místo poslední uložené hodnoty $COUNT(t_{k-N})$.

2.3.3 Analog to Digital Converter (ADC)

Pro výpočet parametrů motoru je potřeba změřit proudy statoru a napětí v meziobvodu. Pro tyto měření je mikrokontrolér vybaven AD převodníkem. AD převodník, který je součástí TMS320F28377D je typu s postupnou aproximací SAR (successive approximation) s volitelnou přesností buď 12 bitů nebo 16 bitů. Blokové schéma pro AD převodník je na obr. 2.6, kde jsou zobrazeny nejdůležitější části. Na vstup AD převodníku lze připojit až 16 zařízení, ale pouze na jednou vstupu dokáže AD převodník měřit, proto se multiplexorem (MUX) vybere jeden ze vstupů ADCIN0-ADCIN15, na kterém se změří hodnota napětí. V bloku S/H (sample and hold) se změří napětí a následně je udržováno po požadovaný čas, tento blok si lze jednoduše představit jako spínač s kondenzátorem. Pro převod analogové hodnoty do 12 bitové nebo 16 bitové digitální podoby slouží AD převodník označený na obr 2.6 jako 12/16 AD Converter. Výsledek z AD převodníku je následně pomocí multiplexoru Result MUX uložen do jednoho z předem určených registrů ADCRESULT 0-15. Start AD převodníku se provádí jedním z 16 SOC (start of conversion) bloků, které určují z jakého vstupu, jak dlouho se bude měřit a do jakého registru se uloží výsledek. SOC bloky lze nastavit do dvou různých režimů, Round Robin a High priority mód. V Round Robin módu nemá žádný SOC pevně danou prioritou, priorita je dána Round Robin ukazatelem, který pracuje v kruhovém módu a neustále provádí start SOC bloků do kola. Round Robin lze spustit také v tzv. Burst módu, kde jedním přerušením se postupně spustí všechny SOC nastavené v Round Robin. Bloky SOC lze přiřadit do módu vysoké priority, ve kterém při vyvolání mají větší prioritu než bloky v Round Robin. Pokud jsou spuštěny 2 bloky z high priority provede se SOC s nižším číslem.

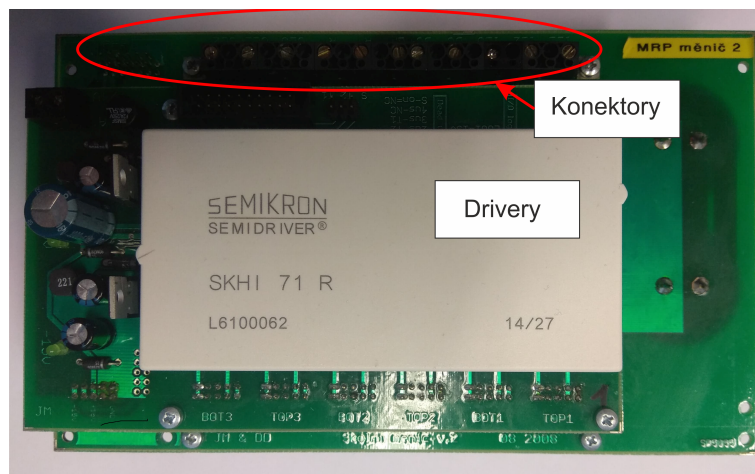
Každý SOC blok má svůj signál end-of-conversion(EOC), kterým lze vyvolat přerušení po ukončení měření.



Obrázek 2.6: blokové schéma AD převodníku [5]

2.4 Měnič kmitočtu

Měnič kmitočtu použitý v této práci je na obr. 2.7 s pohledem na horní část, kde se nachází výkonové konektory pro připojení elektrického motoru a driver od firmy Semikron SKHI 71 R. Výkonová část tohoto měniče kmitočtu je třífázový můstek s IGBT tranzistory od firmy Semikron SK20DGDL065ET s parametry uvedenými v tabulce 2.8.



Obrázek 2.7: Pohled shora na měnič kmitočtu použitého pro implementaci řídicí metody FCS-MPC

parametry	mezí hodnoty
U_{CES}	600 V
I_c	24 A
I_{CRM}	40 A
V_{GES}	± 20 V
T_j	$-40\text{ }^\circ\text{C} \dots +150\text{ }^\circ\text{C}$

Obrázek 2.8: Parametry IGBT prvků v třífázovém můstku Semikron SK20DGDL065ET

2.5 Asynchronní motor 1AC 7070-4AB10Z

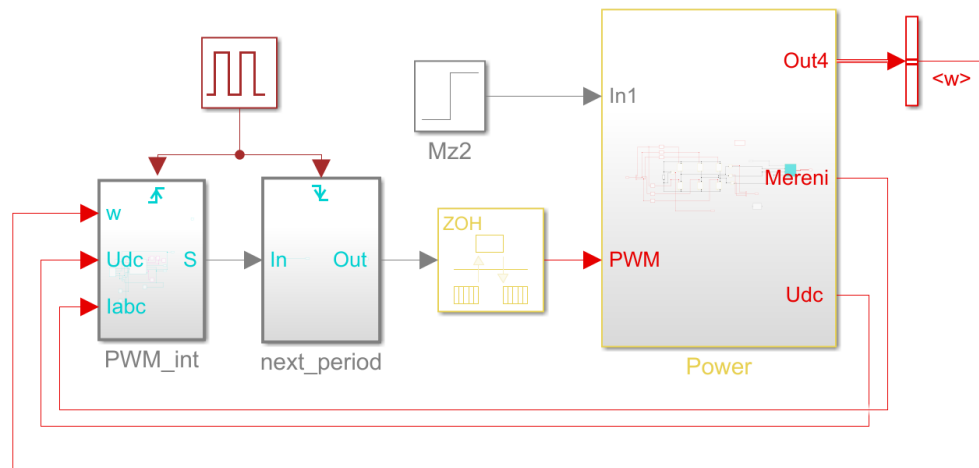
Asynchronní motor, na kterém byla testována metoda řízení FCS-MPC je o firmy Siemens 1AC 7070-4AB10Z. Parametry motoru jsou uvedeny v tabulce 2.7. Parametry uvedené v tabulce jsou jiné než uvedené výrobcem, z důvodu převinutí statorového vinutí pro školní účely.

štitkové údaje	hodnoty
P_N	250 W
U_{5N}	83 V
f_{sn}	50 Hz
p_p	2
R_s	1.86 Ω
R_r	1.53 Ω
L_m	0.033 H
L_{sc}	0.0053 H
L_{rc}	0.0043 H

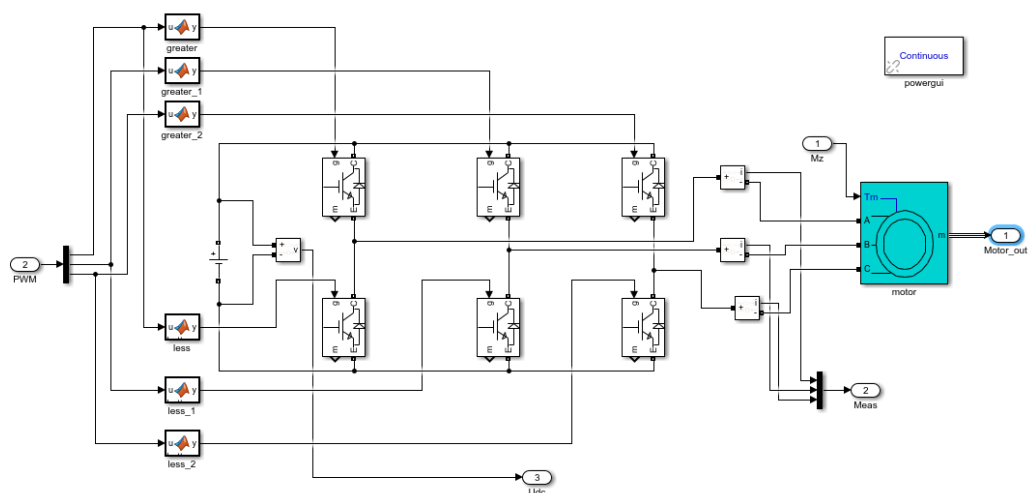
Obrázek 2.9: Parametry asynchronního motoru Siemens 1AC 7070-4AB10Z

3 Simulace

Pro metodu FCS-MPC budou v této kapitole byly provedeny simulace v simulačním prostředí Matlab/Simulink, pro dvě konkrétní metody FCS-MPC proudu a otáček. Jako regulátor otáček u prediktivního řízení proudu je PI regulátor. Blokové schéma zobrazené na obr. 3.1 je shodné pro všechny simulace provedené v této práci, kromě bloku PWMint, ve kterém se nachází různé metody a jejich modifikace. Bloky PWMint a nextperiod jsou společně spouštěny pulsním generátorem. Při náběžné hraně se spustí výpočty a regulace v bloku PWMint, a při doběžné hraně se převede výstup PWMint bloku na vstup bloku Power. Blok Power je složen z dvou-úrovňového třífázového měniče kmitočtu a asynchronního motoru (parametry uvedené v tabulce 2.9), vnitřek tohoto bloku je na obr. 3.2.



Obrázek 3.1: blokové schéma FCS-MPC metody v Simulinku



Obrázek 3.2: blokové schéma měniče kmitočtu s asynchronním motorem v prostředí Simulink

3.1 PI regulace otáček s FCS-MPC regulátorem proudu pro jeden predikční horizont

Tento regulátor vznikne kaskádním spojením prediktivního regulátoru proudu a nadřazeného regulátoru otáček typu proporčně-integrační (PI) regulátor. Požadovanými hodnotami jsou otáčky ω a proud statoru I_{sd} . Tento regulátor byl sestaven s požadavkem na maximální jmenovité otáčky a proto můžeme referenční hodnotu I_{sd} uvažovat v celé regulační oblasti za konstantu. Hodnota I_{sd} je dána jmenovitým tokem statoru podle požadovaného proudu

$$I_{sd}^* = \frac{\psi_m}{L_m} = \frac{0.1}{0.033} = 3.033A$$

Parametry pro PI regulátor proudu jsou dány

- $K_P = 1$
- $K_I = 1000$
- $I_{max} \pm 6A$

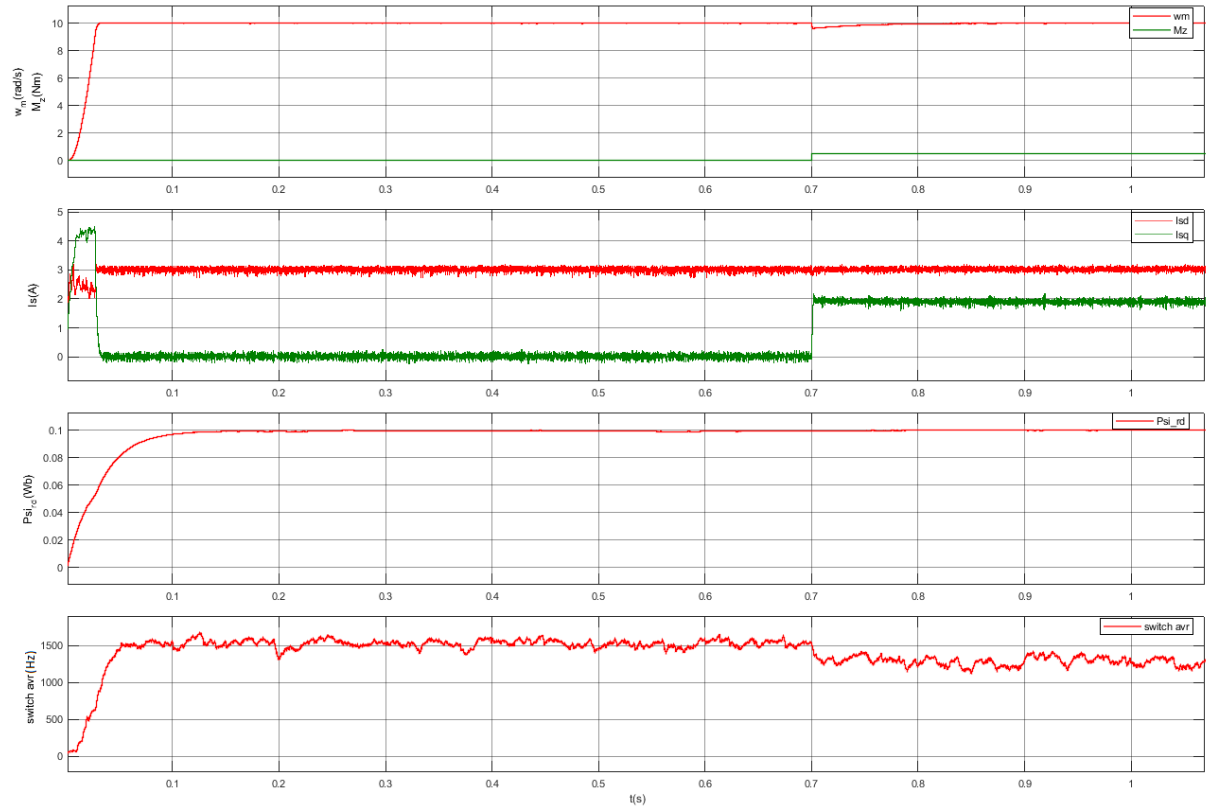
Požadavek na otáčky ω_m^* je v regulátoru otáček porovnáván se skutečnými otáčkami ω_m a výsledná odchylka je požadovaný proud $i_{sq}^*(t_k)$. Požadavek je zadán přímo uživatelem pomocí ovládacího rozhraní, případně může být zadáván nadřazeným regulátorem polohy. K omezení maximálního proudu se za regulátor otáček vkládá omezovač na $\pm 6A$. Cílem regulátoru proudu je minimalizovat rozdíl mezi měřenými a referenčními hodnotami proudu. Minimalizace je ve formě ztrátové funkce, která je vyjádřena v ortogonálních souřadnicích dq .

$$J = |i_d^*(t_{k+1}) - i_d(t_{k+1})| + |i_q^*(t_{k+1}) - i_q(t_{k+1})|$$

Hodnoty $i_d(t_{k+1})$ a $i_q(t_{k+1})$ jsou reálné a imaginární predikované proudy. Tyto hodnoty jsou získány z výpočtů matematického modelu asynchronního motoru. Referenční hodnoty jsou označeny hvězdičkou. Pro jednoduchost se předpokládá, že referenční hodnoty v čase t_{k+1} jsou stejné jako v čase t_k , z důvodu výstupního požadavku PI regulátoru, který neumí vytvářet predikované hodnoty.

$$i^*(t_{k+1}) = i^*(t_k)$$

Na obr. 3.3 jsou zobrazeny nasimulované průběhy $\omega_m, I_{sdq}, \Psi_{rd}$ a průměrná spínací frekvence (switch avr) pro regulační metodu FCS-MPC proudů na jeden predikční horizont. Průběhy jsou simulovány při rozběhu z $\omega_m = 0$ rad/s na požadovaných $\omega_m^* = 10$ rad/s a následně je v 0.7 vteřině proveden skok zátěžného momentu na hodnotu $M_z = 0.5$ Nm.



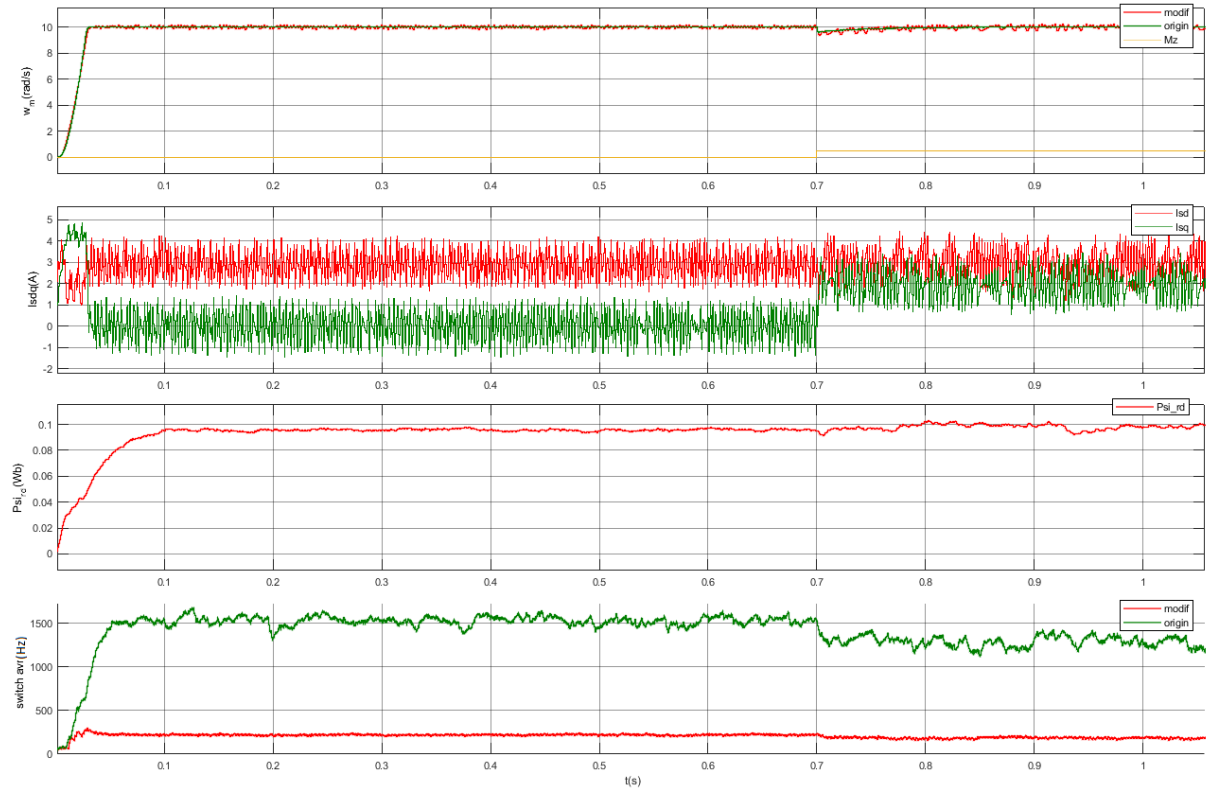
Obrázek 3.3: Průběh otáček ω_m , satorových proudů I_{sd}, I_{sq} , magnetického toku rotoru Ψ_{rd} a průměrná spínací frekvence switch avr

3.2 PI regulace otáček s FCS-MPC regulátorem proudu pro jeden predikční horizont s modifikací

Od předchozí metody se modifikovaná liší ve dvou úpravách. První je přidání kompenzace zpoždění, jenž byla popsána v kapitole 1.3.3 a druhou úpravou je snížení spínacích ztrát. Snížení spínacích ztrát se realizuje porovnáním aktuálně sepnutého vektoru napětí $U_{opt}(t_k)$ a predikovaného vektoru napětí $U_X(t_k)$ právě počítaného. Pokud se vektory nerovnájí je ve ztrátové funkci aktuálně počítaného vektoru $U_X(t_k)$ přičtena hodnota konstanty k_{switch} . Tímto se zvyšuje pravděpodobnost ponechání aktuálně aplikované kombinace a dojde ke snížení spínací frekvence (spínacích ztrát).

$$if(U_{opt}(t_{k+1}) \neq U_X(t_k)) \Rightarrow J(i) = J(i) + k_{switch}$$

Průběhy $\omega_m, I_{sdq}, \Psi_{rd}$ a průměrná spínací frekvence (switch avr) pro regulační metodu FCS-MPC pro jeden PH s úpravou pro snížení spínacích ztrát jsou na obr. 3.4. U průběhu spínacích ztrát je porovnání se stejnou regulační metodou jen bez úpravy pro snížení spínacích ztrát (na obr. 3.4 označen origin).



Obrázek 3.4: Průběh otáček ω_m , satorových proudů I_{sd}, I_{sq} , magnetického toku rotoru Ψ_{rd} a průměrná spínací frekvence switch avr

3.3 PI regulace otáček s FCS-MPC regulátorem proudu pro dva predikční horizonty

Simulace prediktivní regulace proudu typu FCS-MPC s výpočtem na dva predikční horizonty byla provedena bez eliminace spínacích ztrát.

Ztrátová funkce pro první predikční krok je ve tvaru

$$J(k+1) = (i_{sd}(t_{k+1}) - i_{sd}^*(t_k))^2 + (i_{sq}(t_{k+1}) - i_{sq}^*(t_k))^2$$

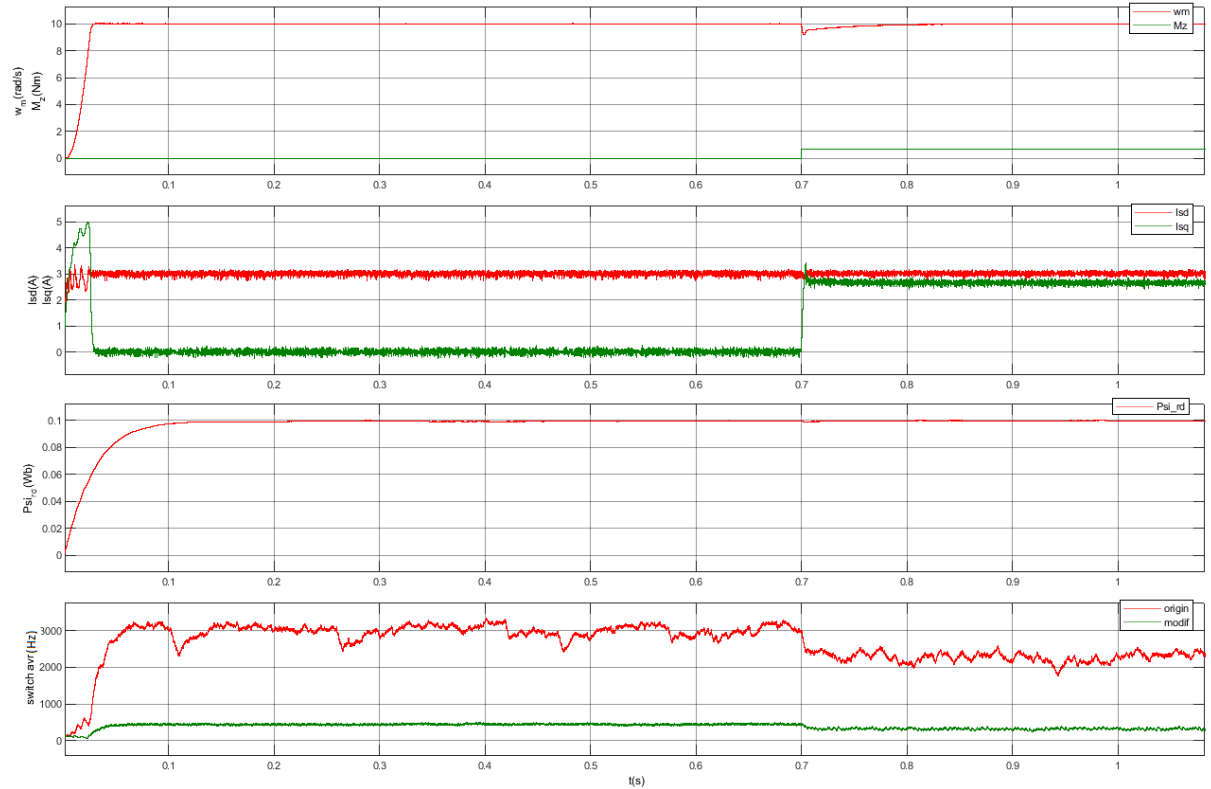
kde n_{sw} určuje změnu spínací kombinace. Ztrátová funkce pro druhý predikční krok je ve tvaru

$$J(k+2) = (i_{sd}(t_{k+2}) - i_{sd}^*(t_k))^2 + (i_{sq}(t_{k+2}) - i_{sq}^*(t_k))^2 + J(t_{k+1})$$

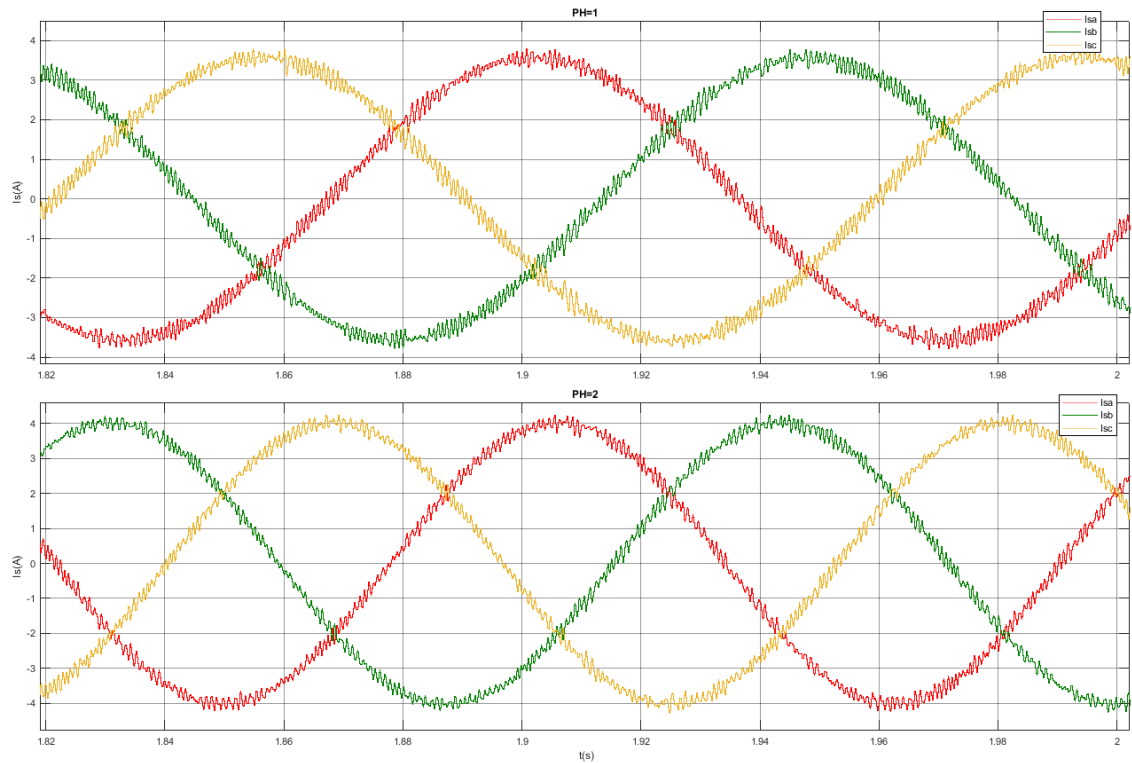
Na obr. 3.5 jsou zobrazeny nasimulované průběhy ω_m , I_{sdq} , Ψ_{rd} a průměrná spínací frekvence (switch avr) pro regulační metodu FCS-MPC proudu na dva predikční horizonty. Průběhy jsou nasimulovány při rozběhu z $\omega_m = 0$ rad/s na požadovaných $\omega_m^* = 10$ rad/s a následně je v 0.7 vteřině proveden skok zátěžného momentu na hodnotu $M_z = 0.5$ Nm. V průběhu spínacích ztrát je zobrazen průběh pro stejnou metodu s úpravou pro snížení spínacích ztrát (obr. 3.5 pod označením modif).

Zde uvedená metoda FCS-MPC dosahuje nejlepšího výsledku z uvedených metod FCS-

MPC proudů. Pro porovnání statorových proudů jsou na obr. 3.6 společně s metodou na dva PH průběhy statorových proudů pro metodu FCS-MPC na jeden PH, také bez úpravy pro eliminaci spínacích ztrát. Cenou za téměř vyhlazené průběhy proudů je vysoká spínací frekvence s hodnotou až kolem 3000 Hz.



Obrázek 3.5: Průběh otáček ω_m , statorových proudů I_{sd}, I_{sq} , magnetického toku rotoru Ψ_{rd} a průměrné změny spínání switch_avr pro metodu FCS-MPC proudů pro dva PH bez úprav pro snížení spínacích ztrát (origin) a s úpravou pro snížení spínacích ztrát



Obrázek 3.6: Průběhy statorových proudů metody FCS-MPC proudů bez modifikací pro PH=1(horní) a PH=2(dolní)

3.4 FCS-MPC regulace otáček pro jeden predikční horizont

Tato metoda je založená na čistě prediktivním regulátoru typu FCS-MPC, bez jakékoliv nadřazené regulace pomocí PI regulátorů ani jiné metody. Protože se eliminoval nadřazený regulátor, odstranila se tím i kaskádní struktura, která má tu nevýhodu, že vyžaduje, aby byl podřazený regulátor několikanásobně rychlejší než nadřazený. Ztrátová funkce v této metodě je ve tvaru

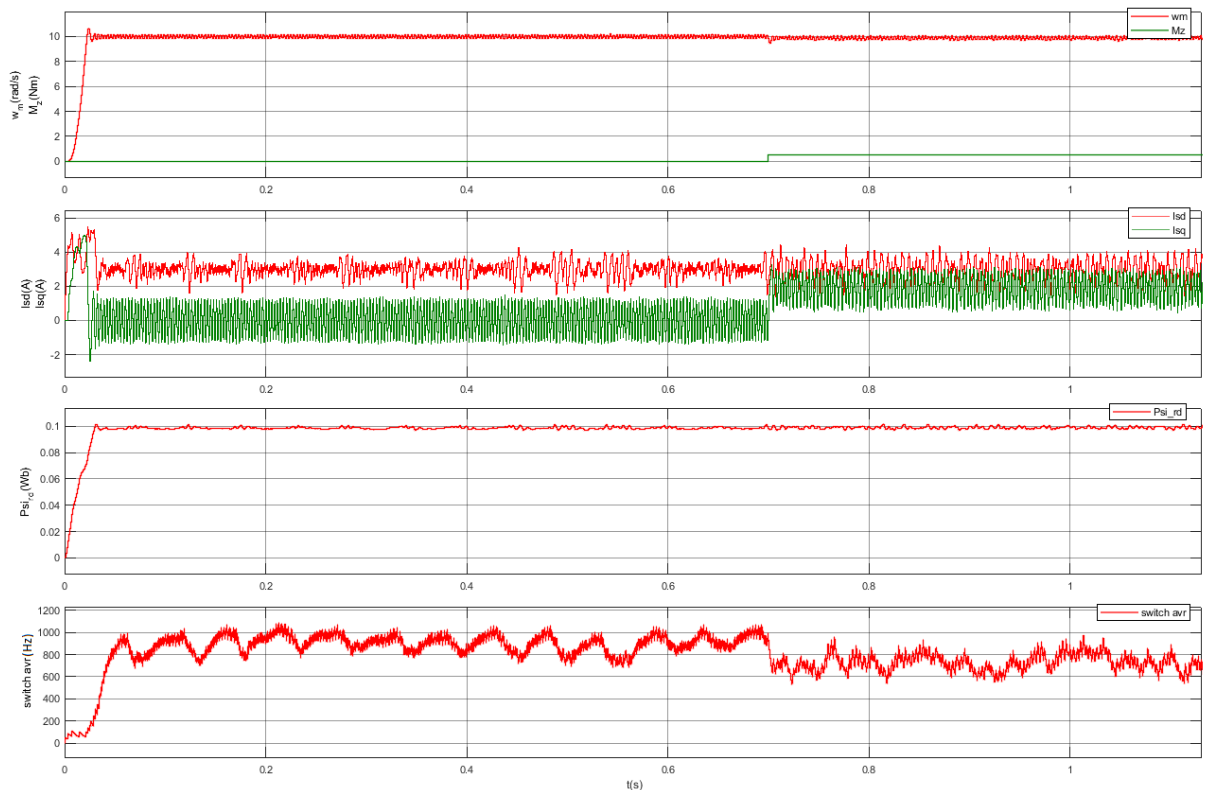
$$J = (\Psi_{rd}^*(t_k) - \Psi_{rd}(t_{k+1}))^2 + (\omega_m^*(t_k) - \omega_m(t_{k+1}))^2 \cdot \lambda_\omega$$

Ztrátová funkce se skládá z kvadrátu rozdílů magnetických toků požadovaných a skutečných a z kvadrátu rozdílů požadovaných a skutečných otáček. Bere se v úvahu, že u požadavků nedochází k okamžité změně, proto je plně dostačující zadávat požadavky jako současné t_k . Rozdíl otáček je vynásoben váhovým koeficientem λ_ω , z důvodu převedení otáček a toků na stejnou úroveň a nebo lze dát dané proměnné menší nebo větší důležitost. Dále je ve ztrátové funkci zahrnuto tvrdé omezení proudů I_{sd}, I_{sq} podle

$$if(\sqrt{I_{sd}(t_{k+1})^2 + I_{sq}(t_{k+1})^2} > I_{max}) \Rightarrow J(i) = J(i) + k_{IMAX}$$

kde při překročení maximální hodnoty proudu I_{max} se přičte ztrátové funkci J určitého vektoru napětí U_I taková konstanta k_{IMAX} , kterou se zajistí, že se daný vektor nevybere jako optimální.

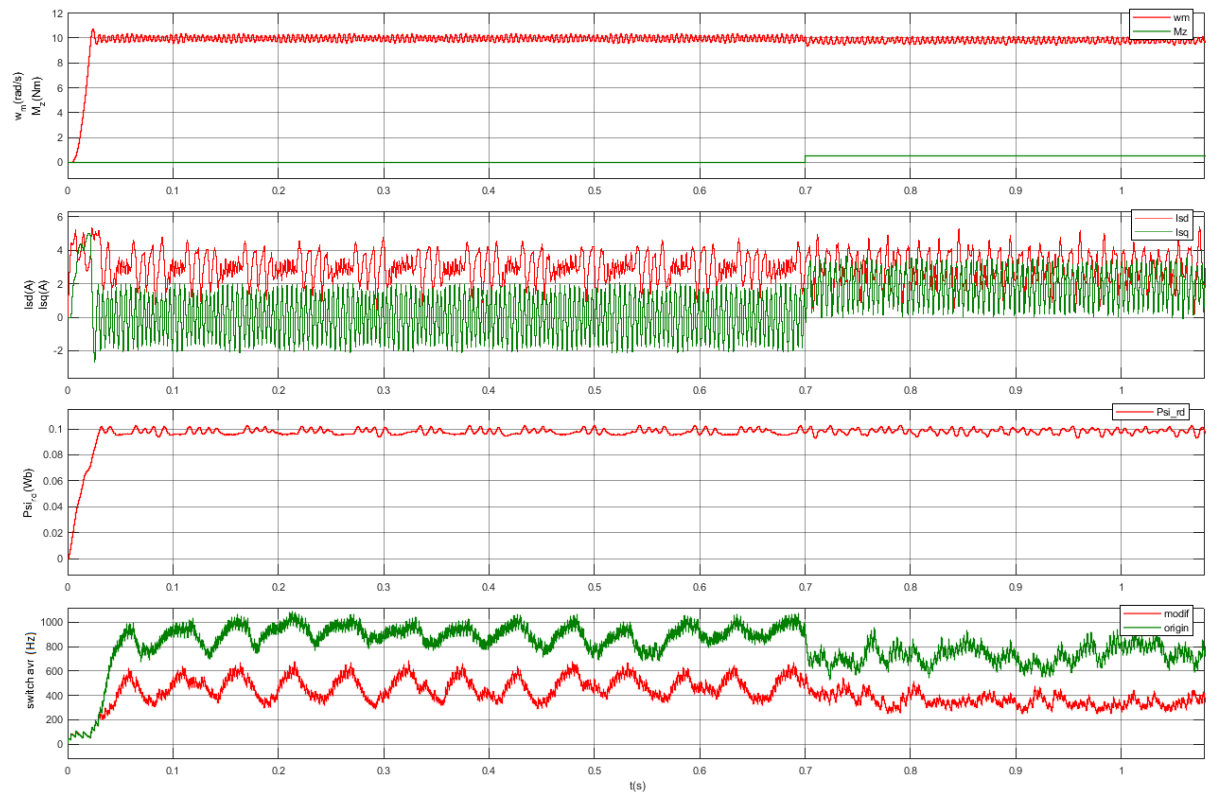
Na obr. 3.7 jsou zobrazeny nasimulované průběhy $\omega_m, I_{sdq}, \Psi_{rd}$ a průměrná spínací frekvence (switch avr) pro regulační metodu FCS-MPC otáček na jeden predikční horizont. Průběhy jsou nasimulovány při rozběhu z $\omega_m=0$ rad/s na požadovaných $\omega_m^* = 10$ rad/s a následně je v 0.7 sekundě proveden skok zátěžného momentu na hodnotu $M_z = 0.5$ Nm.



Obrázek 3.7: Průběh otáček ω_m a zátěžného momentu M_z , satorových proudů I_{sd} , I_{sq} , magnetického toku rotoru Ψ_{rd} a průměrné změny spínání $switch_{avr}$

3.5 FCS-MPC regulace otáček pro jeden predikční horizont s modifikací

Modifikace je ve smyslu snížení spínacích ztrát. Kompensace zpoždění není začleněna z důvodu zhoršení jak spínacích ztrát, tak zhoršení průběhu otáček. Na obr. 3.8 jsou proběhy $\omega_m, I_{sdq}, \Psi_{rd}$ a průměrná spínací frekvence (switch avr). Porovnání je provedeno s nemodifikovanou metodou stejného typu (obr. 3.8 označen jako origin) v průměrných spínacích frekvencích (switch avr).



Obrázek 3.8: Průběh otáček ω_m a zátěžného momentu M_z , statorových proudů I_{sd} , I_{sq} , magnetického toku rotoru Ψ_{rd} a průměrné změny spínání switch_{avr}

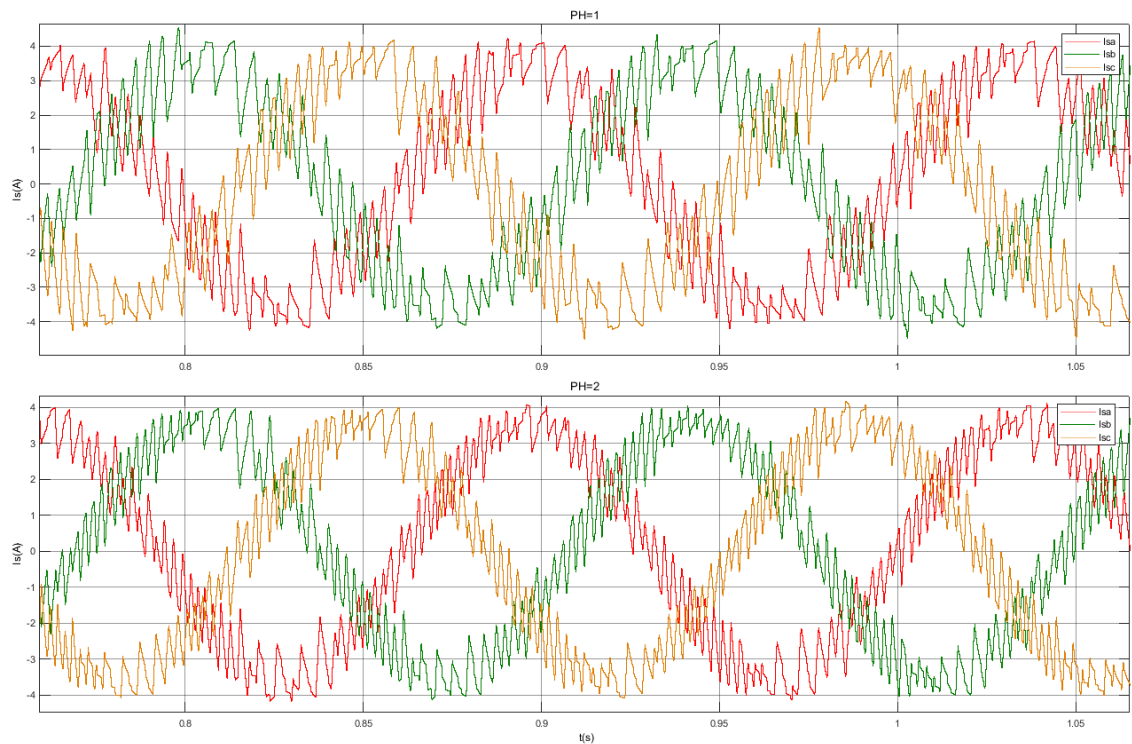
3.6 FCS-MPC regulace otáček pro dva predikční horizonty

Tato metoda byla simulována s úpravou pro snížení spínací frekvence. Průběhy $\omega_m, I_{sdq}, \Psi_{rd}$ na obr. 3.9 jsou simulovány při stejných podmínkách jako předchozí metody. Pro porovnání je průměrná spínací frekvence porovnávána se stejnou metodou na jeden PH (na obr. 3.9 označena PH=1), kde vychází spínací frekvence pro dva PH lépe jak v průměru, tak i v amplitudě.

Podobně jako u metod FCS-MPC proudů je i zde porovnání statorových proudů pro dvě metody FCS-MPC rychlosti bez úprav spínacích ztrát na obr. 3.10 pro jeden PH (horní) a dva PH (spodní). Metoda s dvěma PH vyšla lépe, ale v porovnání s metodami FCS-MPC proudů dosahují proudy této metody mnohem většího zvlnění.



Obrázek 3.9: Průběh otáček ω_m , satorových proudů I_{sd} , i_{sq} , magnetického toku rotoru Ψ_{rd} a průměrné změny spínání switch avr



Obrázek 3.10: Průběhy satorových proudů metod FCS-MPC otáček bez úprav pro snížení spínacích ztrát PH=1(horní) a PH=2(spodní)

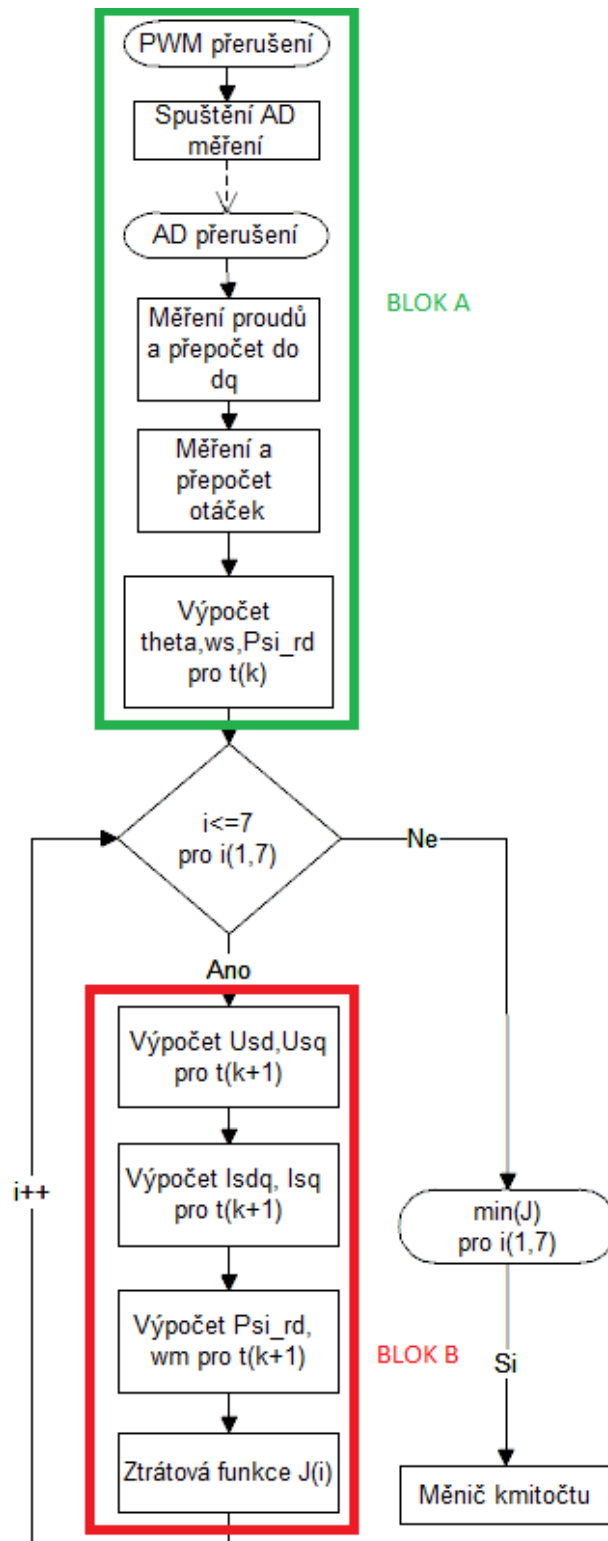
4 Implementace v procesoru

Tvorba aplikací pro mikrokontrolér TMS230F28377D se provádí ve vývojovém prostředí Code Composer Studio(CCS), které je určené pro mikroprocesory od Texas Instruments. Pro nahrání programu do mikrokontroléru pro dvě jádra se musí program rozdělit na dva projekty, pro každé jádro jeden, v této práci pod názvy `FCSMPC_cpu1` a `FCSMPC_cpu2`. Programy pro koprocesory CLA jsou umístěné v nadřazeném jádře CPU_x. Pro správné fungování programu je potřeba nastavit bloky paměti pro jednotlivá jádra a CLA. Nastavení se provádí v souboru `Ram_lnk_cpux.cmp`, kde oblast PAGE 0 určuje programovou paměť a PAGE 1 určuje paměť dat pro jednotlivé výpočetní jednotky. Při nahrávání programu do mikrokontroléru Linker určuje místo uložení podle předpisu v souboru `Ram_lnk_cpux.cmp`.

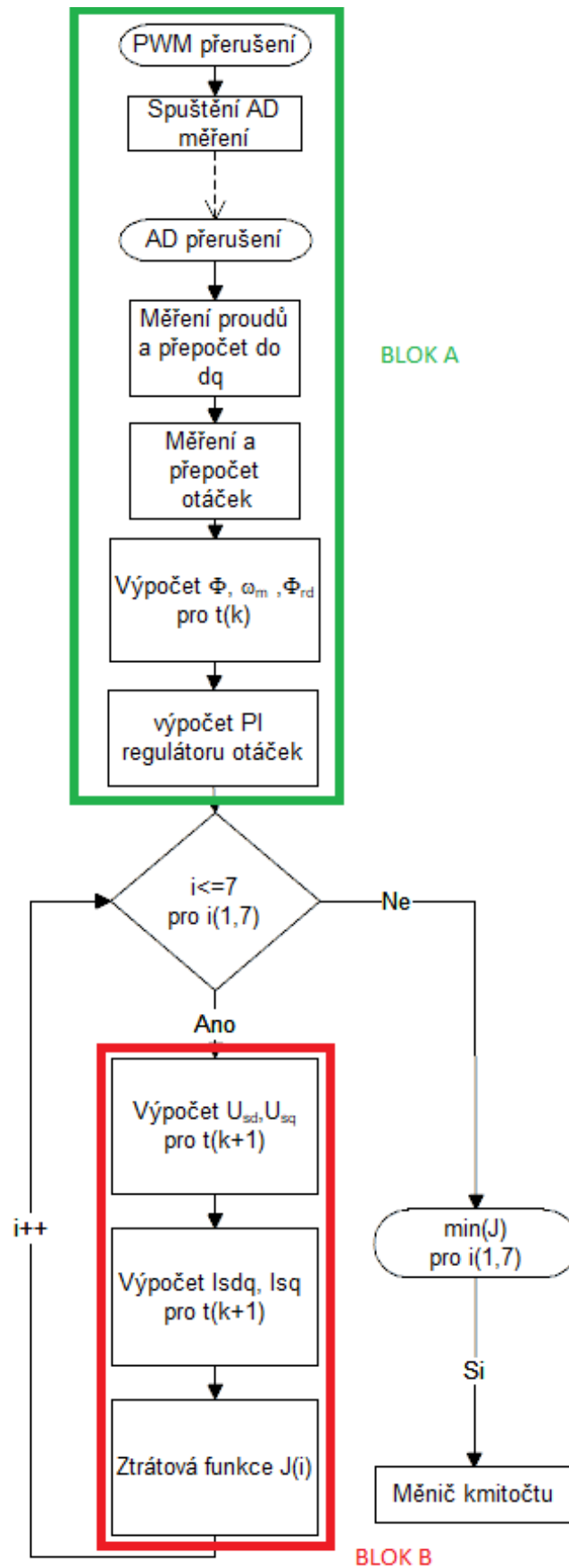
4.1 Popis programu

Pro metody s jedním predikčním horizontem se používá pouze jádro CPU1. Po nahrání programu do RAM paměti se nastaví hodiny, periferie a přerušení. Po ukončení inicializace se z CPU1 povolí přerušení pro PWM periférii. Přerušení od PWM jednotky se děje po určité periodě, jenž určuje spínací frekvenci měniče kmitočtu. V obsluze přerušení od PWM se spustí SOC AD převodníku. Po ukončení měření se vyvolá obsluha přerušení od AD převodníku, kde se převedou změřené hodnoty na dva statorové proudy a napětí v meziobvodu, dále se provede měření otáček a výpočet veličin motoru pro aktuální čas t_k . Tato část měření a výpočtů je na vyznačená na obr. 4.1 a 4.2 jako blok A. U predikční metody proudu na obr. 4.1 je navíc PI regulátor otáček.

Výpočet predikovaných hodnot v čase t_{k+1} pro sedm vektorů napětí se provádí v bloku B označeného na Obr. 4.1 a 4.2. Po těchto výpočtech je v bloku $\min(J)$ vybrán optimální spínací vektor pro následující periodu.



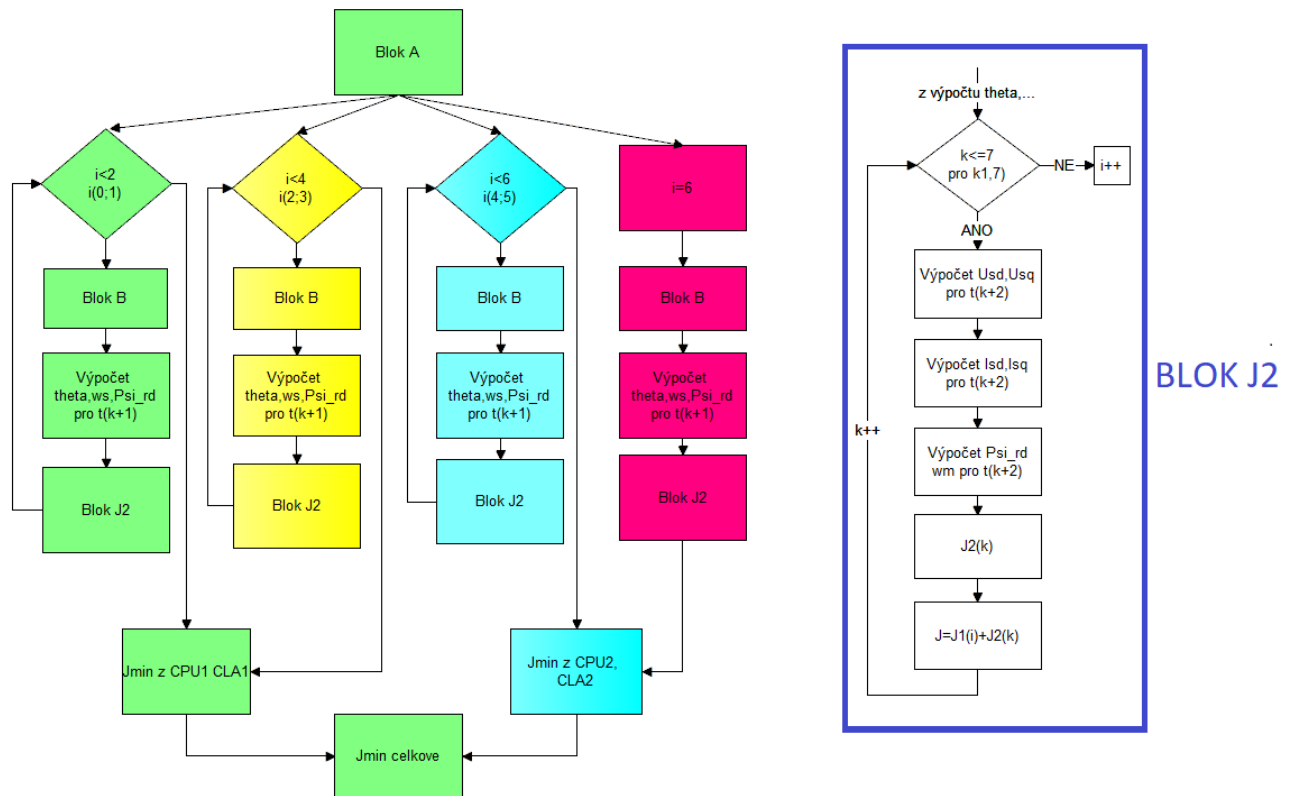
Obrázek 4.1: blokové schéma programu metody FCS-MPC otáček pro jeden predikční horizont



Obrázek 4.2: blokové schéma programu metody FCS-MPC proudu s PI regulátorem otáček pro jeden predikční horizont

Pro metody s dvěma predikčními horizonty se používá i druhé jádro CPU2 a oba koprocesory CLA, ale tyto jednotky slouží pouze pro zrychlení paralelních výpočtů a všechny ostatní výpočty a práce s perifériemi se provádí pomocí jádra CPU1. Po inicializaci CPU1, která je stejná jako u metod s jedním PH se nastaví paměť pro cla a uvolní se společná paměť určená pro CPU2. Po této inicializaci se nahraje program pro CPU2 do uvolněných pamětí, provede se inicializace CLA2 a nakonec se pošle status bit do CPU1 určující připravenost k provádění výpočtů.

Na obr. 4.3 (vlevo) je blokový diagram programu pro metodu FCS-MPC s prediktivním výpočtem na dva PH s barevným označením bloků pro jednotlivé výpočetní jednotky. CPU1(zelená) provádí měření a výpočty pro $t_{(k+1)}$, které jsou detailně zobrazeny na obr. 4.1 a 4.2. Vypočítané parametry jsou rozeslány do ostatních jednotek CLA1(žlutá), CPU2(modrá) a z té se následně odešlou do CLA2(červená), z důvodu nemožnosti přímo komunikovat mezi CPU1 a CLA2. Následně jsou ve všech jednotkách vypočítány matematické modely pro jeden až dva předem určené vektory napětí pro první PH, na obr. 4.3 v bloku B (na obr. 4.1 nebo 4.2 podle dané metody). Pro každý vektor v čase $t_{(k+1)}$, jsou vypočteny hodnoty v, ω_s, Ψ_{rd} potřebné pro výpočet hodnot v čase $t_{(k+2)}$, v bloku J2 na obr. 4.3(vpravo). V bloku J2 jsou provedeny výpočty pro všech 7 vektorů napětí v čase $t_{(k+2)}$. Celkový ztrátový vektor J je součtem ztrátové funkce J1 vypočítané v čase $t_{(k+1)}$ a ztrátové funkce J2 vypočítané v čase $t_{(k+2)}$. Z celkových 14 ztrátových funkcí je vybrán jeden ze dvou vektorů napětí pro čas $t_{(k+1)}$ v každé jednotce, kromě CLA2, kde se počítají hodnoty pouze pro jeden vektor. Z obou CLA jednotek jsou vybrané ztrátové funkce určitých vektorů napětí poslány do nadřazených CPU, kde se porovnají s optimální ztrátovou funkcí z jednotlivých CPU a na konec z jádra CPU2 je výsledek odeslán do jádra CPU1, kde je vybrán výsledný optimální vektor J_{min} , který je odeslán jako spínací kombinace do měniče kmitočtu a v případě kompenzace zpoždění je odeslán až začátkem následující periody.



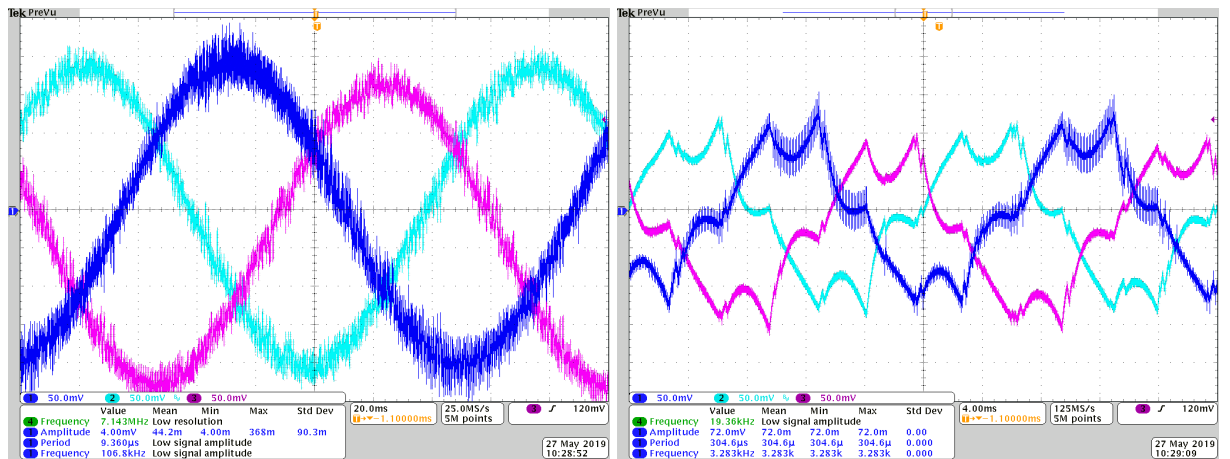
Obrázek 4.3: Blokové schéma(vlevo) rozdělení výpočetních operací mezi jednotky CPU1(zeleně), CPU2(modře), CLA1(žlutě) a CLA2(červeně) a detailní schéma bloku J2(vpravo)

4.2 Výsledky měření

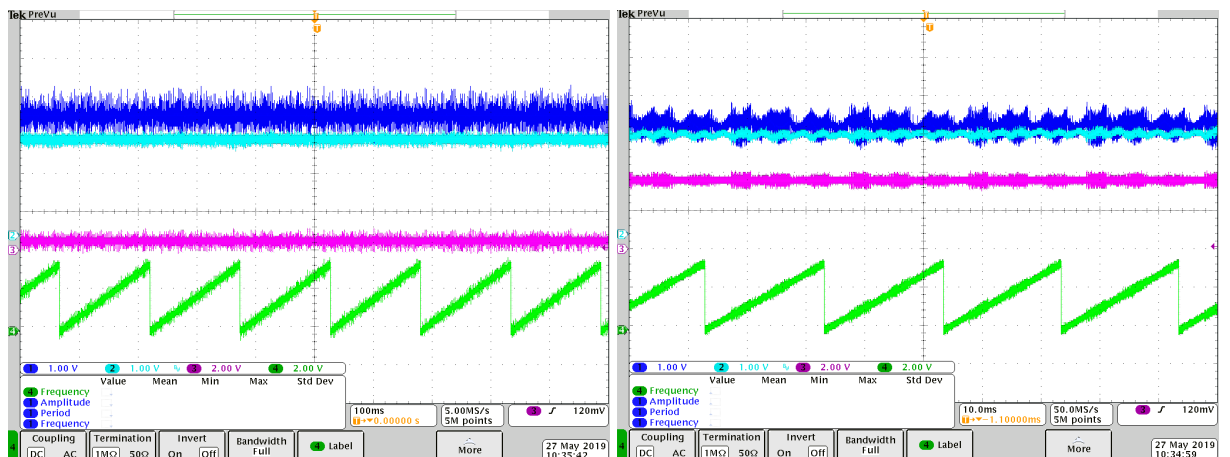
Měření probíhalo na motoru popsaném v kapitole 2.5 bez zátěže a se spínací frekvencí nastavenou na 10 kHz. Napájení zdroje bylo nastaveno na maximální napětí $U=30$ V s proudovým omezením na $I_{max} \pm 5A$. Parametry regulátorů jsou stejné jako v simulacích z třetí kapitoly.

4.2.1 FCS-MPC proudu a PI regulátor otáček pro jeden predikční horizont

Na obr. 4.4 jsou průběhy statorových proudů měřené při dvou různých otáčkách motoru. Při nižších otáčkách $\omega_m=20$ rad/s jsou proudy více vyhlazené, než proudy při $\omega_m=150$ rad/s. Proudů I_{sd}, I_{sq} na obr. 4.5 jsou téměř stejně zvlněné, stejně jako úhel v při různých otáčkách motoru.

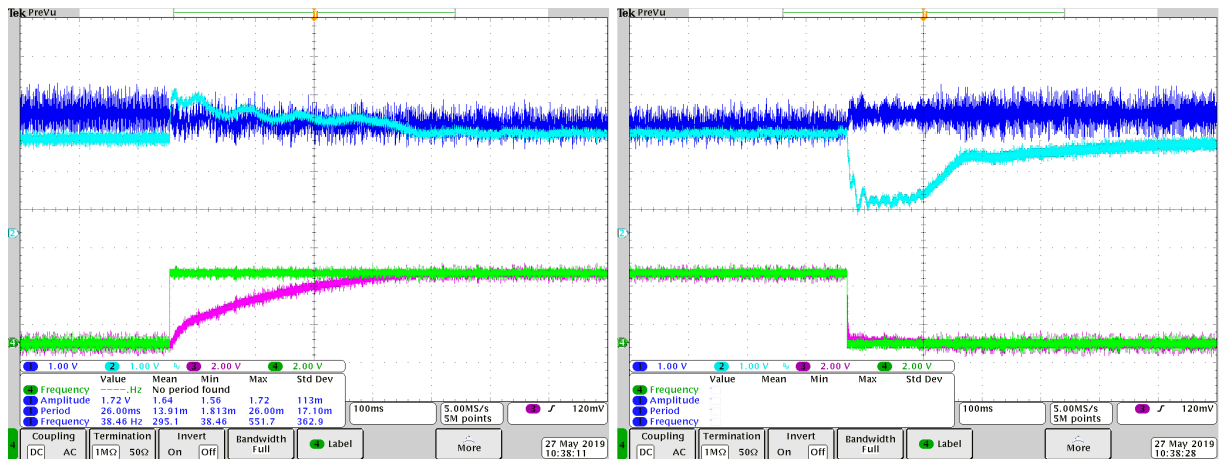


Obrázek 4.4: Průběhy satorových proudů I_{sa}, I_{sb}, I_{sc} při požadovaných otáčkách $\omega_m^* = 20$ rad/s (vlevo) a při požadovaných otáčkách $\omega_m^* = 150$ rad/s (vpravo)



Obrázek 4.5: Průběhy proudů I_{sd} (modrá) (3,2A/d), I_{sq} (purpurová) (3,2A/d), otáček ω_m (růžová) a úhel v při požadovaných otáčkách $\omega_m^* = 20$ rad/s (vlevo) a $\omega_m^* = 150$ rad/s (vpravo)

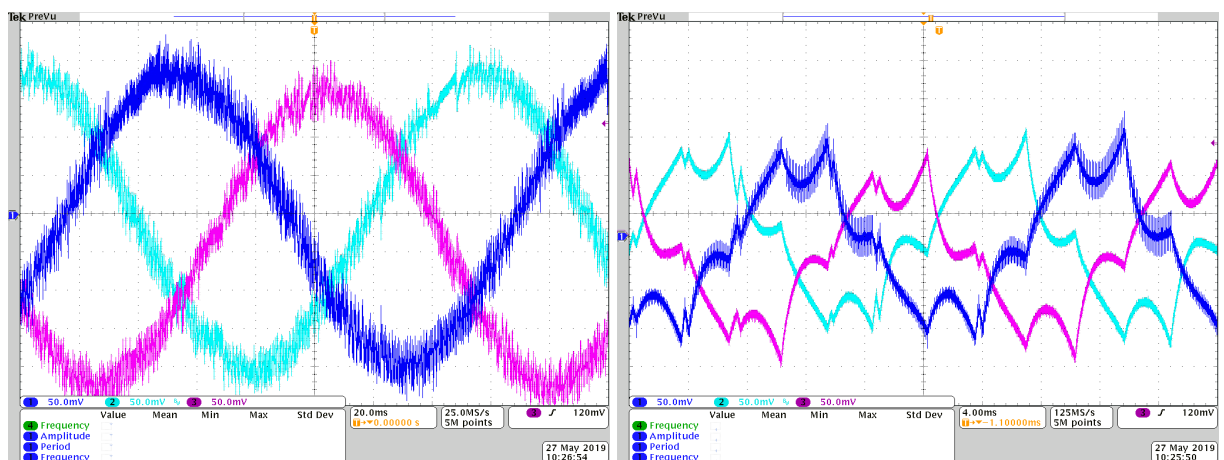
Na obr. 4.6 je zobrazen rozběh motoru při skoku požadavku otáček ω_m^* na 150 rad/s. Při požadavku dojde téměř okamžitě ke skoku proudu I_{sq} na hodnotu 4.5 A. Při dosažení požadovaných otáček se proud I_{sq} sníží na hodnotu 0.6 A. Při skoku požadavku ω_m^* na nulu dojde k poklesu proudu I_{sq} na hodnotu -5.5 A a tím otáčky poklesnou na nulovou hodnotu téměř okamžitě (obr. 4.6 vpravo).



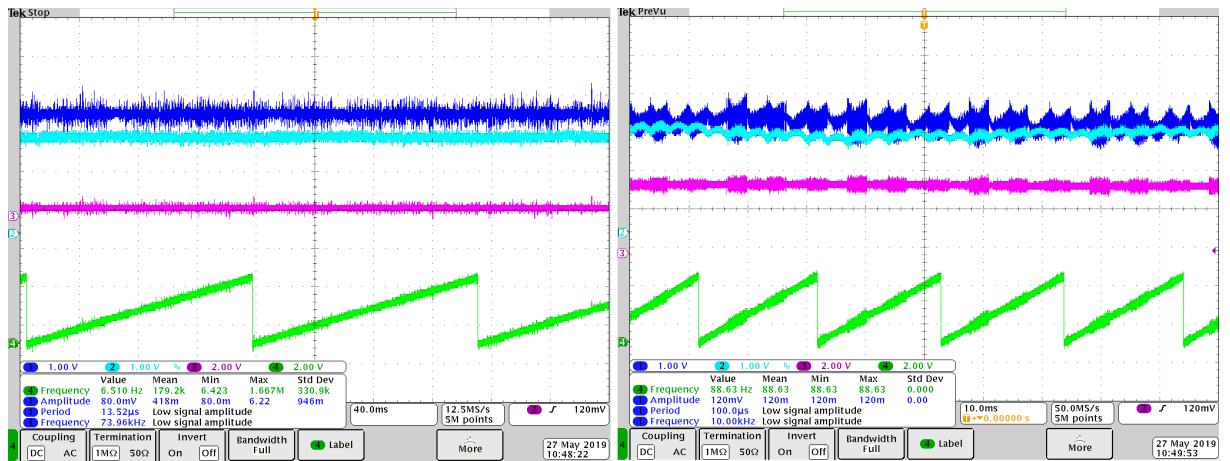
Obrázek 4.6: Průběh proudů I_{sd} (modrá) (3,2A/d), I_{sq} (purpurová) (3,2A/d), otáček ω_m (růžová) při rozběhu motoru na požadované otáčky $\omega_m^* = 150$ rad/s (vlevo) a následně brzdění na $\omega_m^* = 0$ rad/s (vpravo)

4.2.2 FCS-MPC proudu a PI regulátor otáček pro dva predikční horizonty

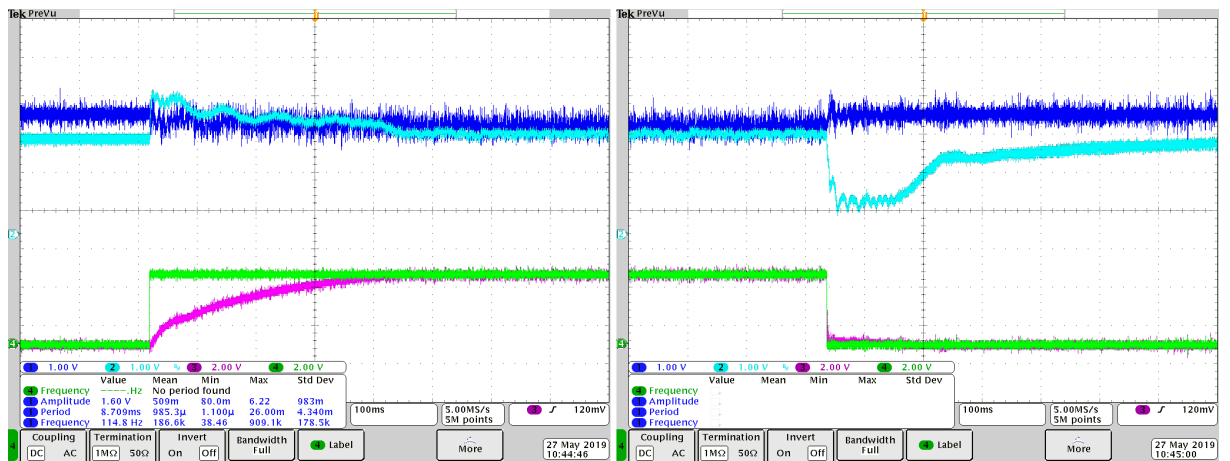
Statorové proudy jsou měřené za stejných podmínek jako u metody stejného typu s jedním PH uvedeným výše. Výsledné průběhy na obr. 4.7 jsou téměř shodné s průběhy pro jeden PH uvedené v 4.2.1.



Obrázek 4.7: Průběh statorových proudů I_{sa}, I_{sb}, I_{sc} při požadovaných otáčkách $\omega_m^* = 50$ rad/s (vlevo) a při požadovaných otáčkách $\omega_m^* = 150$ rad/s (vpravo)



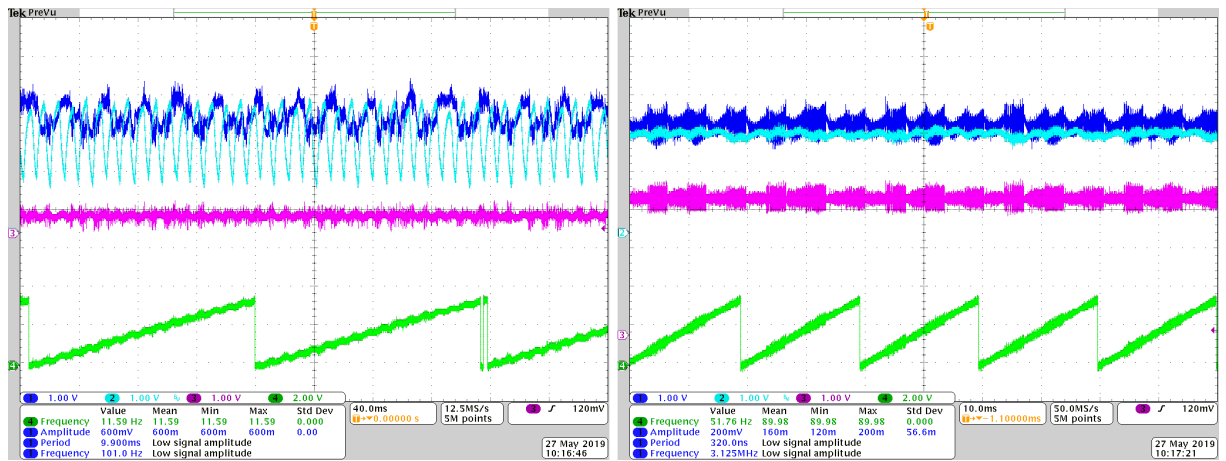
Obrázek 4.8: Průběh proudů I_{sd} (modrá)(3,2A/d), I_{sq} (purpurová)(3,2A/d), otáček ω_m (růžová) a úhel v při požadovaných otáčkách $\omega_m^* = 20$ rad/s (vlevo) a $\omega_m^* = 150$ rad/s (vpravo)



Obrázek 4.9: Průběh proudů I_{sd} (modrá)(3,2A/d), I_{sq} (purpurová)(3,2A/d), otáček ω_m (růžová) při rozběhu motoru na požadované otáčky $\omega_m^* = 150$ rad/s (vlevo) a následně brzdění na $\omega_m^* = 0$ rad/s vpravo

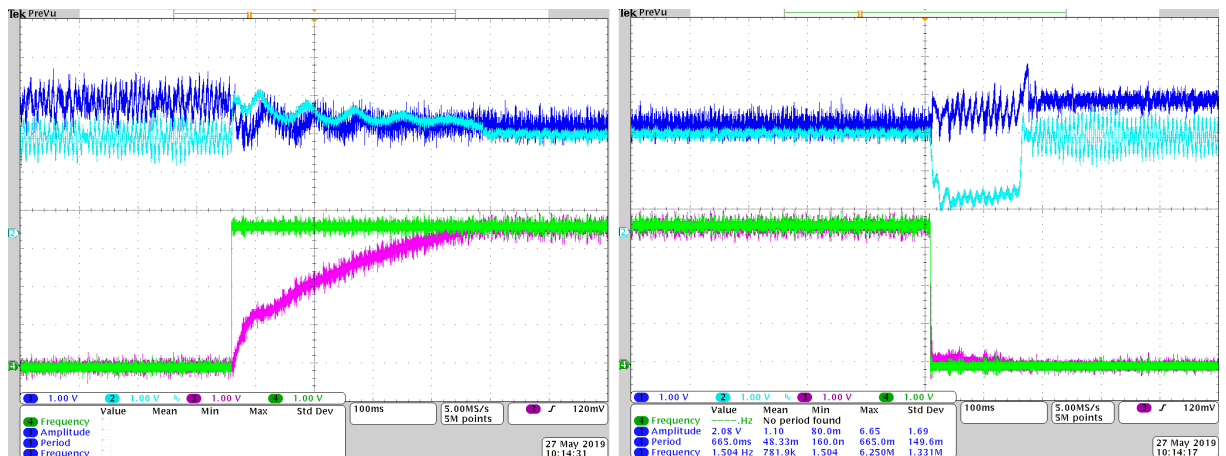
4.2.3 FCS-MPC otáček pro jeden predikční horizont

U regulační metody FCS-MPC otáček je při nižších otáčkách motoru vyšší zvlnění proudů I_{sd}, I_{sq} (obr. 4.10 vlevo), na rozdíl od proudů provozovaných na vyšších otáčkách (obr. 4.10 vpravo).



Obrázek 4.10: Průběh proudů I_{sd} (modrá)(3,2A/d) , I_{sq} (purpurová)(3,2A/d), otáček ω_m (růžová) a úhel v při požadovaných otáčkách $\omega_m^* = 20$ rad/s (vlevo) a $\omega_m^* = 150$ rad/s (vpravo)

Na obr. 4.11 (vlevo) se nejprve nachází motor ve stavu nulových otáček a proudy I_{sd} , I_{sq} jsou zvlněné s amplitudou kolem 1.5 A. Při skoku požadavku na otáčky, se začne motor roztáčet a u proudu I_{sq} se znatelně sníží zvlnění. Proud I_{sd} je stále dost zvlněný s amplitudou přibližně kolem 1.25 A . Při brzdění dojde k poklesu proudu I_{sq} na hodnotu kolem -6 A a po ustálení otáček na nulové hodnotě, dojde opět k rozkmitu proudů I_{sd} , I_{sq} .

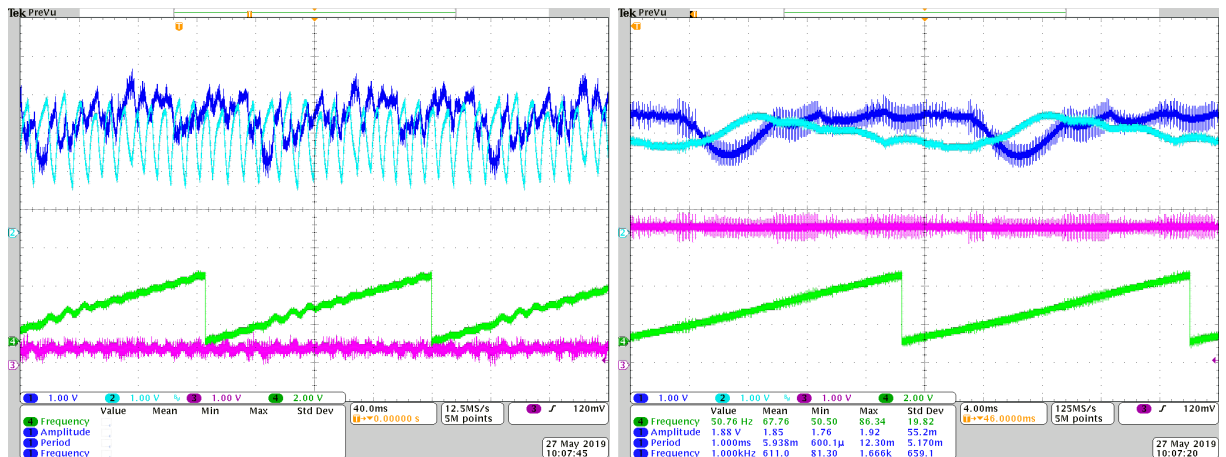


Obrázek 4.11: Průběh proudů I_{sd} (modrá)(3,2A/d), I_{sq} (purpurová)(3,2A/d), otáček ω_m (růžová) při rozběhu motoru na požadované otáčky(zelená) $\omega_m^* = 150$ rad/s (vlevo) a následně brzdění na $\omega_m^* = 0$ rad/s

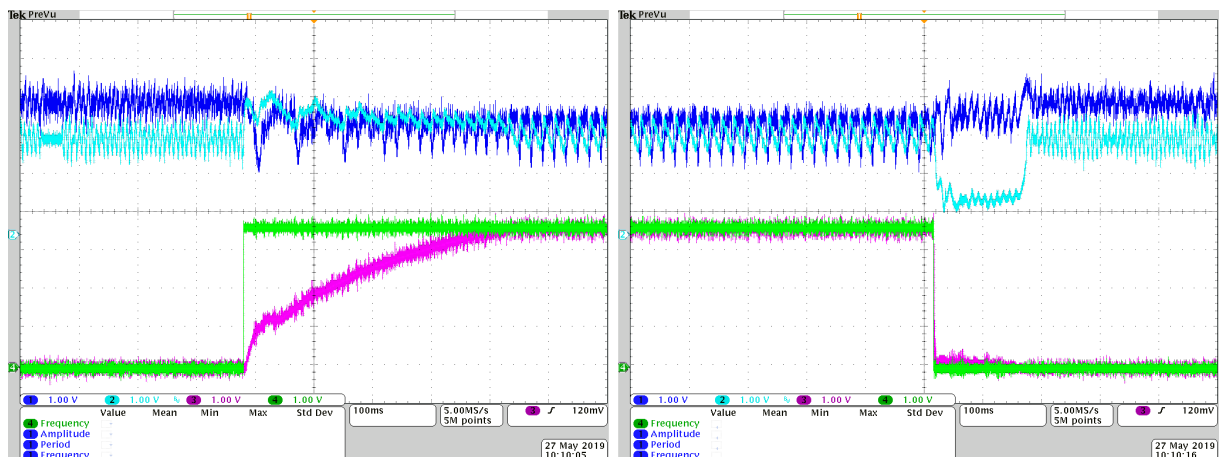
4.2.4 FCS-MPC otáček pro dva predikční horizonty

Stejně jako u prediktivní regulace proudů, je provedeno měření parametrů pro dva PH při stejných podmínkách jako pro jeden PH. Na obr. 4.12 jsou zaznamenány průběhy v ustáleném stavu při otáčkách $\omega_m^* = 20$ rad/s (obr.4.12 vlevo) a $\omega_m^* = 150$ rad/s (obr. 4.12 vpravo). Zaznamenané proudy na obr. 4.11 a 4.12 mají větší zvlnění v porovnání s proudy

na obr. 4.8 a 4.9 pro jeden PH.



Obrázek 4.12: Průběh proudů I_{sd} (modrá) (3,2A/d), I_{sq} (purpurová) (3,2A/d), otáček ω_m (růžová) a úhel v při požadovaných otáčkách $\omega_m^* = 20$ rad/s (vlevo) a $\omega_m^* = 150$ rad/s (vpravo)



Obrázek 4.13: Průběh proudů I_{sd} (modrá) (3,2A/d), I_{sq} (purpurová) (3,2A/d), otáček ω_m (růžová) při rozběhu motoru na požadované otáčky (zelená) $\omega_m^* = 150$ rad/s (vlevo) a následně brzdění na $\omega_m^* = 0$ rad/s (vpravo)

4.2.5 Doba výpočtu

Pro všechny čtyři metody testované na mikrokontroléru TMS320F28377D byla zaznamenána doba výpočtu jednotlivých metod obr. 4.14. V době výpočtu je zaznamenána od konce AD převodu až po aplikaci spínací kombinace na měnič kmitočtu. Metody s jedním predikčním horizontem jsou vypočítávané pouze na jádře CPU1 a u metod s dvěma PH se většina výpočtů provádí na všech 4 výpočetních jednotkách jak je zobrazeno na obr. 4.3. Nejnáročnější na výpočet jsou obecně řídicí metody FCS-MPC otáček, kde se v porovnání s metodou FCS-MPC proudu, vypočítávají navíc prediktivní hodnoty Ψ_{rd} a ω_m , které jsou použity jako parametry ve ztrátové funkci. Pro porovnání byla provedena

implementace metody FCS-MPC otáček pro dva PH pouze na jednom jádře, kde výpočty v jedné periodě trvaly dohromady 202,59 μs .

metoda řízení	doba výpočtu t(μs)
FCS-MPC proudu,PI regulátor otáček PH=1	24,90
FCS-MPC proudu,PI regulátor otáček PH=2	59,50
FCS-MPC otáček PH=1	32,10
FCS-MPC otáček PH=2	62,66
FCS-MPC otáček PH=2(pouze CPU1)	202,59

Obrázek 4.14: Tabulka doby výpočtů jednotlivých metod prediktivního řízení

5 Závěr

V první části práce byla obecně popsána teorie prediktivního řízení, konkrétně metoda FCS-MPC. Následně byly v této kapitole popsány různé transformace souřadnic a v souřadnicovém systému dq byl popsán matematický model pro asynchronní motor.

Ve druhé části byly popsány jednotlivé hardwarové komponenty použité v této práci. Větší část této kapitoly je věnována mikrokontroléru TMS320F28377D, konkrétně všem výpočetním jednotkám, komunikacím mezi nimi a perifériím potřebným k měření veličin a správnému generování výstupních signálů k ovládání měniče kmitočtu. Na konci druhé kapitoly jsou stručně uvedeny parametry měniče kmitočtu a asynchronního motoru, které jsou použity v praktické části.

Ve třetí části byly provedeny simulace pro prediktivní metody FCS-MPC otáček a proudů. Pro obě metody byly provedeny také simulace úprav pro snížení spínacích ztrát a kompenzace zpoždění. Dále byly všechny tyto metody jak bez úprav, tak s úpravami nasimulovány pro dva predikční horizonty. Z výsledných průběhů statorových proudů, vychází obecně mnohem lépe metody FCS-MPC proudů, konkrétně metoda pro dva predikční kroky bez úpravy pro snížení spínací frekvence. V porovnání dosažení požadovaných otáček, jsou rychlejší metody FCS-MPC otáček, kde požadovaných otáček dosáhly za 0.02 s, zatímco u metod FCS-MPC proudů byla dosažena požadovaná rychlost za 0.032 s. V porovnání spínacích ztrát vyšla nejlépe metoda FCS-MPC proudů pro jeden PH s úpravou pro snížení spínacích ztrát s průměrnou hodnotou sepnutí 215 Hz. U metod prediktivních metod otáček, vyšla nejlépe metoda na dva PH s průměrnou hodnotou sepnutí 275 Hz (průměrná spínací frekvence byla měřena v ustáleném stavu při otáčkách $\omega_m = 10$ rad/s).

V poslední kapitole byla popsána implementace regulačních metod FCS-MPC proudu a otáček v mikrokontroléru TMS320F28377D, jak pro jedno jádro, tak pro všechny čtyři výpočetní jednotky. Následně jsou uvedené výsledné průběhy měření při rozběhu, brzdění a v ustálených stavech motoru pro dvě metody FCS-MPC proudu a otáček pro jeden a dva predikční horizonty. Z měřených průběhů u prediktivních metod proudů, vychází lépe predikční metoda pro dva PH. U predikčních metod otáček vyšla naopak lépe metoda s jedním PH. Důvod horšího výsledku může být z důvodu nepřesného modelu motoru a tím nepřesného predikování na dva horizonty. Na závěr je tabulka porovnání doby výpočtu pro jednotlivé metody, ve které jsou uvedené čtyři měřené metody. Navíc je v tabulce zaznamenána predikční metoda otáček pro dva PH, počítaná na jednom jádře. Z tohoto výsledku je vidět 3.2x násobné zrychlení výpočtů díky optimalizaci pro dvě jádra CPU a dva koprocesory CLA oproti stejnému počtu výpočtů na jednom jádře.

Použití mikrokontroléru TMS320F28377D pro paralelní výpočty se jeví jako perspektivní, kde u metody FCS-MPC na dva PH došlo ke zrychlení výpočtů o 323 %. Použití metody FCS-MPC proudu na dva PH nepřináší velké výhody oproti stejné metodě na jeden PH, z důvodu velké výpočetní náročnosti a téměř stejným výsledným proběhům.

Seznam použité literatury

- [1] ARNE, L.; RAHUL, K.; KENNEL, R.; STOLZE, P.: Model-Based Predictive Control of Electric Drives. Munich, 2010
- [2] S. Kouro, P. Cortes, R. Vargas, U. Ammann and J. Rodriguez, "Model Predictive Control—A Simple and Powerful Method to Control Power Converters,"in IEEE Transactions on Industrial Electronics, vol. 56, no. 6, pp. 1826-1838, June 2009.
- [3] Rodríguez, J. , Pérez, M. A., Young, H. and Abu-Rub, H. (2014). Model Predictive Speed Control of Electrical Machines. In Power Electronics for Renewable Energy Systems, Transportation and Industrial Applications (eds H. Abu-Rub, M. Malinowski and K. Al-Haddad).
- [4] Koblíček, P.; Pavelka, J.: Elektrické pohony a jejich řízení. Fakulta Elektrotechnická ČVUT v Praze, 2016.
- [5] Schachter, Kenneth W: TMS320F2837xD Microcontroller Workshop. Texas Instrument, 2018.
- [6] Wikipedia, The Free Encyclopedia[online]:IEEE 754-1985
- [7] T. Kosan, J. Talla, S. Janous and V. Blahnik, "FPGA-Based Accelerator for Model Predictive Control of Induction Motor Drive,"2018 18th International Conference on Mechatronics - Mechatronika (ME), Brno, Czech Republic, 2018, pp. 1-6.
- [8] C. Bordons and C. Montero, "Basic Principles of MPC for Power Converters: Bridging the Gap Between Theory and Practice,"in IEEE Industrial Electronics Magazine, vol. 9, no. 3, pp. 31-43, Sept. 2015.
- [9] A. A. Ahmed, B. K. Koh and Y. I. Lee, "A Comparison of Finite Control Set and Continuous Control Set Model Predictive Control Schemes for Speed Control of Induction Motors,"in IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1334-1346, April 2018.
- [10] PHUNG QUANG, Nguyen a Jörg-Andreas DITTRICH. Vector Control of Three-Phase AC Machines: System Development in the Practice. Second Edition. Berlin: Springer, Berlin, Heidelberg, 2015. ISBN 978-3-662-46915-6.
- [11] Schachter, Kenneth W: TMS320F2837xD Microcontroller Workshop. Texas Instrument, 2018.
- [12] Texas Instrument Incorporated: TMS320F2837xD Microcontroller Workshop. Dallas, Texas, 2017.