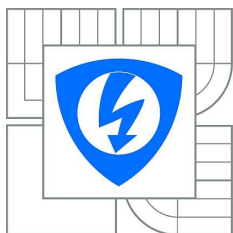


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ALGORITMY PREDIKTIVNÍHO ŘÍZENÍ ELEKTRICKÝCH POHONŮ

PREDICTIVE CONTROL ALGORITHMS OF ELECTRICAL DRIVES

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

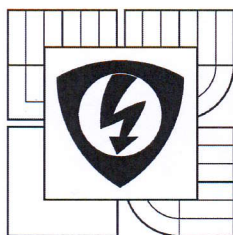
AUTOR PRÁCE
AUTHOR

Bc. ZBYNĚK MYNÁŘ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. PAVEL VÁCLAVEK, Ph.D.

BRNO, 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Zbyněk Mynář

Ročník: 2

ID: 125558

Akademický rok: 2013/14

NÁZEV TÉMATU:

Algoritmy prediktivního řízení elektrických pohonů

POKYNY PRO VYPRACOVÁNÍ:

V rámci práce budou studovány algoritmy prediktivního řízení střídavých pohonů s modelem. Zadání lze shrnout do následujících bodů:

1. Zpracujte rešerši týkající se algoritmů řízení střídavých pohonů se zaměřením na prediktivní algoritmy.
2. Simulačně ověřte vybrané existující algoritmy prediktivního řízení synchronního nebo asynchronního motoru.
3. Navrhněte a popište vlastní řešení prediktivního řídicího algoritmu.
4. Navržený algoritmus realizujte v prostředí Matlab-Simulink
5. Simulačně ověřte a vyhodnoťte vlastnosti navrženého algoritmu.

DOPORUČENÁ LITERATURA:

- [1] Caha, Z.; Černý, M. : Elektrické pohony, Praha, SNTL 1990.
- [2] G. C. Goodwin, M. Seron, and J. De Dona, Constrained control and estimation: an optimisation approach. Springer Verlag, 2005.
- [3] S. Bolognani, L. Peretti, and M. Zigliotto, "Design and Implementation of Model Predictive Control for Electrical Motor Drives," IEEE Trans. Ind. Electron., vol. 56, no. 6, pp. 1925–1936, Jun. 2009.

další dle pokynů vedoucího práce

Termín zadání: 10.2.2014

Termín odevzdání: 19.5.2014

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

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

předseda oborové rady

UPOZORNĚNÍ:

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



ABSTRAKT

Tato diplomová práce se zabývá algoritmy prediktivního řízení střídavých pohonů. V úvodní části se nachází shrnutí současného stavu teorie a dále popis a zařazení nejvýznamnějších prediktivních algoritmů. Zvláštní kapitola je věnována algoritmu lineárního prediktivního řízení založeného na modelu (lineární MPC). Hlavní přínos této práce pak spočívá v představení dvou nových prediktivních algoritmů pro řízení PMSM motoru, které vycházejí z lineárního MPC. První z těchto algoritmů vznikl s cílem minimalizace svých výpočetních nároků, zatímco u druhého algoritmu byla zavedena schopnost motor odbudit. Oba nové algoritmy, včetně lineárního MPC, byly simulovány v prostředí MATLAB-Simulink.

KLÍČOVÁ SLOVA

Prediktivní řízení, MPC, lineární LMPC, synchronní motor, PMSM

SUMMARY

This work deals with the predictive control algorithms of the AC drives. The introductory section contains summary of current state of theory and further description and classification of most significant predictive algorithms. A separate chapter is dedicated to linear model predictive control (linear MPC). The main contribution of this work is the introduction of two new predictive control algorithm for PMSM motor, both of which are based on linear MPC. The first of these algorithms has been created with the aim of minimizing its computational demands, while the second algorithm introduces the ability of field weakening. Both new algorithms and linear MPC were simulated in MATLAB-Simulink.

KEYWORDS

Predictive control, MPC, linear MPC, synchronous machine, PMSM

MYNÁŘ, Z. *Algoritmy prediktivního řízení elektrických pohonů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 89 s. Vedoucí diplomové práce byl doc. Ing. Pavel Václavek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Algoritmy prediktivního řízení elektrických pohonů“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Místo

.
(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce doc. Ing. Pavlovi Václavkovi, Ph.D, za jeho ochotu, trpělivost a cenné odborné rady, jež mi poskytl při vypracování této práce.

Místo

.

(podpis autora)

Seznam obrázků

1.1	Příklad struktury prediktivního regulátoru střídavého pohonu	16
1.2	Rozdělení algoritmů pro řízení pohonů dle funkčního principu [9] . . .	17
1.3	Princip deadbeat regulace proudu [9]	19
1.4	Blokové schéma deadbeat regulátoru proudu	20
1.5	Princip prediktivní regulace proudu založené na hysterezi [16]	21
1.6	Blokové schéma histerezního prediktivního regulátoru proudu [16] . .	21
1.7	Princip řízení statorového toku algoritmem DTC	23
1.8	Blokové schéma prediktivního algoritmu DTC	23
1.9	Způsob řízení momentu pomocí algoritmu DMTC	25
1.10	Princip řízení s pomocí DSPC algoritmu [22]	26
1.11	Strategie ustupujícího horizontu pro predikční horizont $N = 3$	28
1.12	Princip řídicího horizontu N_u [18]	30
1.13	Blokové schéma MPDTC algoritmu [11]	32
1.14	Blokové schéma řízení střídavého pohonu pomocí PCC algoritmu [28]	33
2.1	Obrazce tvořené vektory jednotkové délky dle a) $\mathcal{L}_1$ -normy, b) kvad- ratické a $\mathcal{L}_2$ -normy a c) $\mathcal{L}_\infty$ -normy v $\mathbb{R}^2$	41
2.2	Příklady účelových funkcí založených na a) $\mathcal{L}_1$ -normě, b) $\mathcal{L}_2$ -normě, c) $\mathcal{L}_\infty$ -normě a kvadratické normě	42
2.3	Rozdělení stavového prostoru s pomocí explicitního lineárního MPC .	48
2.4	Blokového schéma řízení PMSM motoru pomocí LMPC algoritmu . .	53
2.5	Napěťové omezení aproximované s pomocí osmiúhelníku	57
2.6	Proudové omezení s pomocí obdélníku	58
3.1	Blokové schéma řízení PMSM motoru LMPTC algoritmem	60
3.2	Blokové schéma rekonstruktoru rotorového magnetického toku	62
3.3	Interpretace omezení statorového napětí jako kružnice s pohyblivým středem v rovině virtuálního napětí	68
4.1	Srovnání průběhů časových závislostí úhlové rychlosti rotoru ω_m . . .	79
4.2	Srovnání průběhů statorového magnetického toku Ψ_{sd}	80
4.3	Srovnání průběhů časových závislostí momentu m	81
4.4	Průběh momentu a jeho reference u LMPTC algoritmu	81
4.5	Srovnání vektorů statorových a) napětí u_{sd} a u_{sd} a b) proudů i_{sd} . . .	82

Seznam tabulek

4.1	Parametry synchronního motoru s permanentními magnety použitého v simulaci [5]	71
4.2	Srovnání výpočetní náročnosti implicitních verzí algoritmů	84
4.3	Srovnání výpočetní náročnosti explicitních verzí algoritmů	84

Seznam zkratek a symbolů

Symbol	Popis	Jednotka
a	Úhlové zrychlení	$[\text{rad/s}^{-2}]$
$A(q^{-1})$	Výstupní polynom CARIMA modelu	$[-]$
$\mathbf{A}$	Matice zpětných vazeb diskrétního systému	$[-]$
$\mathbf{A}'$	Matice zpětných vazeb spojitého systému	$[-]$
$\tilde{\mathbf{A}}$	Rozšířená matice zpětných vazeb systému	$[-]$
$\mathbf{A}_{6i}$	Matice šestiúhelníkového omezení proudu	$[-]$
$\mathbf{A}_{6u}$	Matice šestiúhelníkového omezení napětí	$[-]$
$\mathbf{A}_8$	Matice osmiúhelníkového omezení	$[-]$
$\mathbf{A}_{xset}$	Matice omezení stavů	$[-]$
$\vec{b}_4$	Vektor obdélníkového omezení	$[-]$
$\vec{b}_{6i}$	Vektor šestiúhelníkového omezení proudu	$[-]$
$\vec{b}_{6u}$	Vektor šestiúhelníkového omezení napětí	$[-]$
$\vec{b}_8$	Vektor osmiúhelníkového omezení	$[-]$
$\vec{b}_{xset}$	Vektor definující obecné omezení stavů	$[-]$
$B(q^{-1})$	Vstupní polynom CARIMA modelu	$[-]$
$\mathbf{B}$	Matice vazeb diskrétního systému na vstup	$[-]$
$\mathbf{B}'$	Matice vazeb spojitého systému na vstup	$[-]$
$\tilde{\mathbf{B}}$	Rozšířená matice vazeb systému na vstup	$[-]$
c_j	Počet změn sepnutí měniče za danou spínací sekvencí j	$[-]$
$C(q^{-1})$	Polynom vstupu poruchy CARIMA modelu	$[-]$
$\mathbf{C}$	Matice vazeb výstupu na stav systému	$[-]$
$\mathbf{D}$	Matice vazeb vstupu na výstup systému	$[-]$
$\mathbf{E}$	Matice omezení změny akčního zásahu	$[-]$
e	Řídicí odchylka	$[-]$
e_s	Zpětné elektromotorické napětí	$[\text{V}]$
f	Frekvence	$[\text{Hz}]$
$\mathbf{G}$	Matice nucené odezvy GPC	$[-]$
$\mathbf{H}$	Hessova matice	$[-]$
H_y	Hranice hystereze DSPC algoritmu	$[\text{rad/s}]$

i_s	Proud statorovým vinutím	[A]
i_s^*	Proudová reference statorového proudu	[A]
I	Jednotková matice	[-]
J	Moment setrvačnosti	[kgm ²]
J_j	Účelová funkce MPDTC algoritmu	[-]
J_N	Účelová funkce	[-]
k	Pořadí vzorku	[-]
K_{INT}	Integrační zesílení	[-]
L	Matice omezení optimalizovaného vektoru vstupů	[-]
L_d	Indukčnost statorového vinutí v d -ose	[H]
L_f	Penalizace stavů a vstupů systému	[-]
L_q	Indukčnost statorového vinutí v q -ose	[H]
m	Točivý moment motoru	[Nm]
m^*	Žádaná hodnota moment motoru	[Nm]
m_A	Točivý moment při aktivním napěťovém vektoru	[Nm]
m_z	Zátěžný moment	[Nm]
m_Z	Točivý moment při nulovém statorovém napětí	[Nm]
n_a	Řád polynomu $A(q^{-1})$	[-]
n_b	Řád polynomu $B(q^{-1})$	[-]
n_c	Řád polynomu $C(q^{-1})$	[-]
n_j	Počet kroků extrapolace, jež se veličina nachází v mezích	[-]
n_u	Počet vstupů systému	[-]
n_x	Počet stavů systému	[-]
n_y	Počet výstupů systému	[-]
N	Délka predikčního horizontu	[-]
N_u	Délka řídicího horizontu	[-]
p	Laplaceův operátor	[-]
P_p	Počet pólových párů motoru	[-]
P	Matice penalizující konečný stav systému	[-]
q^{-1}	Operátor zpoždění o jeden krok	[-]
Q	Matice penalizující hodnoty stavů	[-]
Q[*]	Rozšířená matice penalizující hodnoty stavů	[-]
R	Matice penalizující hodnoty akčního zásahu	[-]

$\mathbf{R}^*$	Rozšířená matice penalizující hodnoty akčního zásahu	[-]
R_s	Odpor statorového vinutí	[Ω]
s	Spínací kombinace tranzistorů měniče	[-]
S	Spínací sekvence tranzistorů měniče	[-]
t_A	Doba po kterou působí moment m_A	[s]
t_Z	Doba po kterou působí moment m_Z	[s]
T_s	Vzorkovací perioda	[s]
$\vec{u}$	Vektor vstupů systému	[-]
u_{np}	Napětí neutrálního bodu	[V]
u_{N_u-1}	Optimalizovaný vektor budoucích vstupů	[-]
u_s	Statorové napětí	[V]
U_{DC}	Napětí meziobvodu	[V]
U_{max}	Maximální hodnota statorového napětí	[V]
$\vec{x}$	Vektor stavů systému	[-]
$\vec{x}'$	Rozšířený vektor stavů systému	[-]
$\vec{y}$	Vektor výstupů systému	[-]
$\vec{w}$	Vektor žádaných hodnot	[-]
$\mathbf{W}$	Matice omezení optimalizovaného vektoru vstupů	[-]
$\vec{y}_N$	Predikce výstupů systému	[-]
$\vec{\delta u}$	Vektor změn vstupů systému	[-]
μ	Koeficient penalizace odchylky	[-]
λ	Koeficient penalizace akčního zásahu	[-]
γ	Penalizační koeficient	[-]
ξ	Gaussův šum	[-]
ϑ_f	Úhel natočení rotorového magnetického toku	[rad]
ϑ_Ψ	Úhel rotorovým a statorovým magnetickým tokem	[rad]
ω_e	Elektrická úhlová rychlost	[rad/s]
ω_m	Mechanická rychlost rotoru	[rad/s]
ω_m^*	Žádaná mechanická rychlost rotoru	[rad/s]
ω_{mi}^*	Žádaná rychlost v včetně integrační složky	[rad/s]
ε	Uvolňující proměnná (slack variable)	[-]
Ψ_{PM}	Magnetický tok permanentního magnetu	[Vs]
Ψ_r	Rotorový magnetický tok	[Vs]

Ψ_s	Statorový magnetický tok	[Vs]
Γ	Matice prediktoru stavů	[-]
Γ_y	Matice prediktoru výstupu	[-]
Ω	Vstupní matice prediktoru stavů	[-]
Ω_y	Vstupní matice prediktoru výstupu	[-]

Význam dolních indexů

Zkratka	Význam
max	Maximální hodnota
min	Minimální hodnota
$\alpha, \beta, \alpha\beta$	Složky vektoru veličiny ve statorovém souřadném systému
d, q, dq	Složky vektoru veličiny v rotorovém souřadném systému
a, b, c, abc	Složky vektoru v jednotlivých fázích motoru

Seznam zkratk

Zkratka	Význam
ACIM	AC Induction Machine (asynchronní motor)
CARIMA	Controlled Autoregressive Integrated Moving Average
DMC	Dynamic Matrix Control
DMTC	Direct Mean Torque Control (přímé řízení střední hodnoty momentu)
DSC	Digital Signal Controller (digitální signálový kontrolér)
DSPC	Direct Speed Predictive Control (přímé prediktivní řízení rychlosti)
DTC	Direct Torque Control (přímé řízení momentu)
FOC	Field Oriented Control (vektorové řízení)
GPC	Generalized Predictive Control (obecné prediktivní řízení)
LMPC	Linear MPC (lineární MPC)
LMPCFW	Linear MPC with Filed Weakening (lineární MPC s odbuzováním)
LMPTC	Linear Model Predictive Torque Control (lineární MPC řízení momentu)
LQ	Linear Quadratic
MAC	Model Algorithmic Control (algoritmické řízení založené na modelu)

MIMO	Multi-Input, Multi-Output (systém s více vstupy a výstupy)
MPC	Model Predictive Control (prediktivní řízení založené na modelu)
MPDTC	Model Predictive Direct Torque Control (přímé prediktivní řízení momentu založené na modelu)
NLMPC	Nonlinear MPC (nelineární MPC)
PCC	Predictive Current Control (prediktivní řízení proudů)
PI	Proporcionálně Integrační regulátor
PID	Proporcionálně Integračně Derivační regulátor
PMSM	Permanent Magnet Synchronous Machine (synchronní motor s permanentními magnety)
PWM	Pulse Width Modulation (pulsně-šířková modulace)
QDMC	Quadratic Dynamic Matrix Control
RHPC	Receding Horizon Predictive Control (prediktivní řízení s ustupujícím horizontem)
SISO	Single-Input, single-Output (systém s jediným vstupem a výstupem)
SVM	Space Vector Modulation (modulace metodou prostorových vektorů)
SMPMSM	Surface Mounted PMSM (PMSM s magnety na povrchu rotoru)

Obsah

Úvod	14
1 Prediktivní řízení střídavých pohonů	15
1.1 Rozdělení prediktivních algoritmů	17
1.1.1 Deadbeat prediktivní regulace	18
1.1.2 Prediktivní řízení založené na hysterezi	20
1.1.3 Prediktivní řízení založené na trajektorii	22
1.1.4 Prediktivní řízení založené na modelu	26
2 Lineární prediktivní řízení založené na modelu	38
2.1 Popis lineárního prediktivního řízení založeného na modelu	38
2.1.1 Model systému	38
2.1.2 Účelová funkce	40
2.1.3 Omezení	41
2.2 Lineární MPC regulátor	44
2.2.1 Explicitní lineární MPC regulátor	47
2.2.2 Sledování reference s lineárním MPC	49
2.2.3 Nulová ustálená řídicí odchylka s lineárním MPC	50
2.2.4 Stabilita řízení s lineárním MPC	51
2.3 Lineární MPC pro řízení synchronního motoru s permanentními mag- nety	52
3 Vlastní algoritmy prediktivního řízení založeného na modelu	59
3.1 Lineární prediktivní řízení momentu založené na modelu	59
3.2 Lineární MPC s implementací odbuzování	65
4 Simulace navržených algoritmů	71
4.1 Návrh regulátoru lineárního prediktivního řízení založeného na modelu	72
4.2 Návrh regulátoru lineárního prediktivního řízení momentu založeného na modelu	74
4.3 Návrh regulátoru lineárního MPC s odbuzením	75
4.4 Výsledky simulací	78
Závěr	85
Přílohy	89

Úvod

Střídavé elektrické pohony, tedy především synchronní a asynchronní motory, jsou známy již od počátku éry elektrotechniky. Oba typy motorů jsou vyráběny v širokém rozsahu výkonů. Asynchronní motor vyniká svou spolehlivostí a synchronní motor s permanentními magnety (PMSM) pak svou účinností a výhodným poměrem mezi objemem a výkonem. V současnosti se v praxi nejčastěji pro jejich řízení používá vektorového řízení (FOC), s nímž je možno dosáhnout vynikajících výsledků i v aplikacích náročných na dynamiku.

Stále důležitější alternativou k FOC se však stávají prediktivní algoritmy řízení. Obvyklou výhodou těchto algoritmů je schopnost využít informace, jež máme o řízeném systému k získání, v uživatelem definovaném smyslu, optimálního akčního zásahu. Další výhody oproti FOC pak budou shrnuty v první kapitole této práce. V téže kapitole pak bude provedeno rozdělení jinak velmi rozmanitých algoritmů prediktivního řízení, a to na základě funkčního principu. V rámci tohoto přehledu pak budou popsány principy, výhody a nevýhody nejvýznamnější prediktivních algoritmů pro řízení elektropohonů.

Velmi významnou skupinou prediktivních algoritmů jsou ty založené na modelu (v překladu Model Predictive Control či MPC). Zvláště pak algoritmus lineárního MPC, který má nesporné výhody jednoduché aplikace pro MIMO systémy, snadné implementace omezení a - díky nedávnému pokroku v teorii- také možnosti výrazného snížení výpočetních nároků. Z těchto důvodů byla algoritmu, včetně popisu jeho nasazení pro synchronní motor s permanentními magnety, věnována celá druhá kapitola. [5], [12]

Hlavním cílem této práce byl návrh vlastního prediktivního algoritmu. Ve třetí kapitole tak budou představeny dva nové algoritmy pro řízení PMSM motoru. Oba byly postaveny na lineárním MPC algoritmu, což jim propůjčilo jeho nesporné výhody. První z algoritmů byl nazván Linear Model Predictive Torque Control (LMPTC). Jeho cílem je byla minimalizace svých výpočetních nároku. Druhým algoritmem je Linear MPC with Filed Weakening, či LMPCFW. Jak název napovídá, účelem bylo získání možnosti lineárním MPC řízený motor algoritmem odbudit.

Poslední, čtvrtá kapitola je věnována simulacím nově navržených algoritmů včetně vektorového řízení a klasické implementace lineárního MPC pro řízení PMSM motoru. Simulace byly provedeny v prostředí MATLAB-Simulink verze 7.14.0.739.

1. Prediktivní řízení střídavých pohonů

V současnosti jsou pro dosažení dobrých dynamických vlastností při řízení střídavých pohonů nejčastěji používány algoritmy vektorového řízení (FOC). K jejich rozšíření došlo v devadesátých letech minulého století v důsledku příchodu cenově dostupných a výkonných mikroprocesorů. Základem těchto algoritmů je převedení prostorových vektorů veličin motoru do dvoufázové rotující souřadné soustavy (podrobněji viz příloha A). Při vhodné volbě rychlosti rotace tohoto systému pak přestávají být takto transformované veličiny harmonické, čímž dojde k oddělení řízení rotorového magnetického toku a momentu motoru. Motor tak získává vlastnosti stejnosměrného motoru s cizím buzením.

K regulaci proudů a rychlosti jsou u FOC použity PI (případně PID) regulátory v kaskádní struktuře (viz příloha B). Tento přístup však vykazuje jistá omezení. Kaskádní struktura se vyznačuje tím, že podřízená smyčka musí být výrazně rychlejší než smyčka nadřazená. Mnohdy se udává zhruba až desetkrát kratší časová konstanta. Při řízení polohy to znamená až 100 krát rychlejší proudová smyčka, než polohová, což vede na extrémně krátkou vzorkovací periodu. Tato potřeba přitom nevyplývá z časových konstant systému, ale pouze z použití kaskádní struktury. [18]

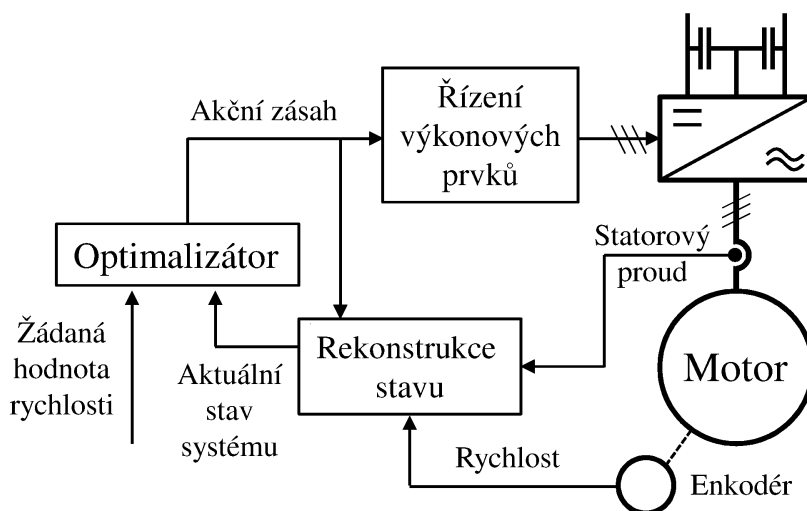
Nevýhodou klasických PID regulátorů je, že v sobě neobsahují znalosti o regulované soustavě a tudíž nemohou být využity. Tyto informace, například ve formě matematického modelu, jsou použity pouze při jejich syntéze. Během regulace jsou tak brány v úvahu pouze současné a skrze integrační složku i minulé hodnoty regulační odchylky. Analogicky si tak řízení s PID regulátorem můžeme představit jako řízení automobilu pouze s pomocí odrazu ze zpětných zrcátek. Tento přístup neodpovídá lidskému intuitivnímu řízení, jehož velmi důležitým prvkem je, na zkušenostech založená, předpověď budoucnosti. Prediktivní řízení bylo vytvořeno právě s myšlenkou do výpočtu akčního zásahu zahrnout i predikci budoucího chování systému. Tu lze získat z matematického modelu systému. Cílem tohoto přístupu je pak dosažení optimálního řízení a to v uživatelském zadaném smyslu. Jedná se o společnou společnou charakteristiku jinak velmi rozmanitých prediktivních algoritmů.

Tím, že PID regulátory nejsou úzce spjaty s matematickým modelem systému, získávají jistou robustnost vůči jeho nepřesnosti. Naplnění stále náročnějších a různorodějších požadavků, jež jsou dnes mnohdy na regulaci kladeny, se však stává poměrně obtížným. Často je pak nutno zavádět zvláštní řídicí vazby či modifikace PID regulátoru jako je například feedforward vazba pro omezení překmitu či filtrace

derivační složky pro omezení vlivu šumu. Mnohdy se tedy jedná o nesystémová řešení. Nastavení takového regulátoru se pak může pro osobu bez znalostí teorie regulace stát poněkud obtížným. Naproti tomu u mnoha prediktivních algoritmů je naprostá většina vlastností výsledného regulovaného systému nastavitelná na jediném místě, a to takzvané účelové funkci. Změna chování regulační smyčky při změně koeficientů v této funkci je přitom lehce předvídatelná. Nastavení takového prediktivního regulátoru je tak intuitivní a proveditelné pouhým laikem.

Další nevýhodou PID regulátoru je, že samotný dokáže regulovat pouze jedinou veličinu systému. Při regulaci MIMO systémů je tak nutné použít více samostatných PID regulátorů. Pokud pak některý z těchto regulátorů začne pracovat v omezení, musí být tato informace zohledněna v ostatních regulátorech, což mnohdy není jednoduše řešitelné. Vzhledem k tomu, že jsou v dnešní době akční členy často provozovány na hranicích svých možností, je právě jednoduchá a systematická implementace omezení u MIMO systémů důležitým faktorem. Zde se projeví výhoda některých prediktivních algoritmů, jež pro výpočet optimálního akčního zásahu musí uvažovat aktuální hodnoty všech stavů systému a zpracovat je v jediném regulátoru, který tak má k dispozici všechny potřebné informace. Nasazení těchto algoritmů pro MIMO systémy pak není o nic složitější než pro systémy typu SISO.

Typická struktura prediktivního algoritmu pro řízení střídavého pohonu je na obrázku 1.1. Blok rekonstrukce stavu má za úkol sestavit aktuální stavový vektor systému. Ten je spolu s žádanou hodnotou regulovaných veličin použit v bloku optimalizátoru k výpočtu optimální hodnoty akčního zásahu. Blok řízení výkonových prvků, který případně může obsahovat modulátor, pak vhodným způsobem řídí spínání tranzistorů v měniči, čímž vytváří požadované statorové napětí.



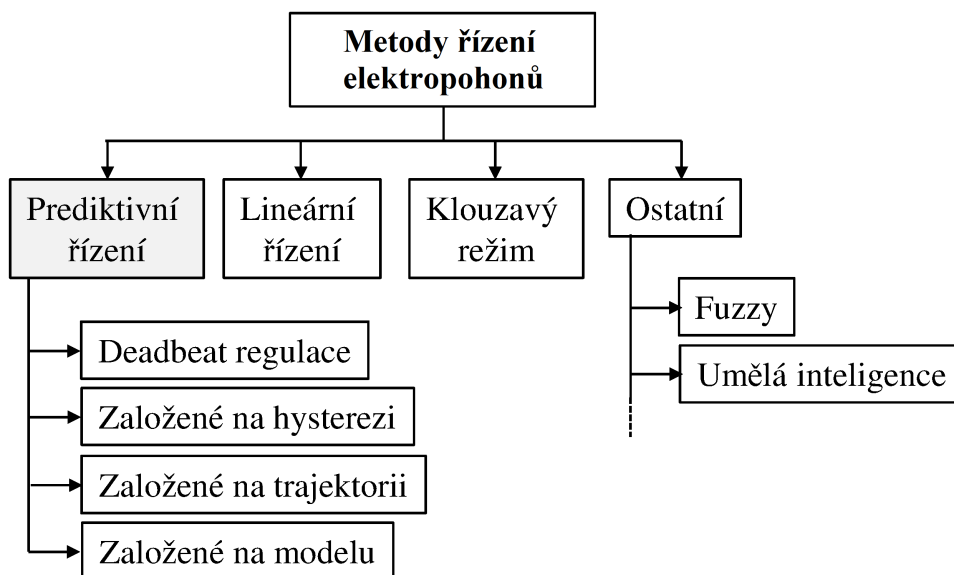
Obrázek 1.1: Příklad struktury prediktivního regulátoru střídavého pohonu

Ne všechny prediktivní algoritmy disponují všemi zmíněnými přednostmi oproti PID regulátorům. Těch dosahují spíše výpočetně složitější algoritmy, především ty založené na modelu (viz kapitola 1.1.4). Jejich společným nedostatkem je však vysoká výpočetní náročnost, která je, u tak rychlých systémů jako jsou ty elektromechanické, zásadní. Zatímco například u chemických soustav se díky dlouhým časovým konstantám nasazují výpočetně náročné prediktivní algoritmy běžně, pro řízení elektromotorů se v praxi užívají spíše jednodušší principy. Avšak díky rostoucímu výpočetnímu výkonu dnešních mikroprocesorů a pokrokům v teorii prediktivního řízení lze očekávat, že pro řízení elektropohonů budou nasazovány stále komplexnější prediktivní algoritmy, jež umožní dosáhnout náročných požadavků na řízení.

Od počátků vývoje prediktivních algoritmů v šedesátých letech minulého století bylo vyvinuto velké množství prediktivních algoritmů. Tato práce se však bude soustředit především na ty, jež lze využít pro řízení elektromotorů. V následujícím textu bude proveden souhrn nejvýznamnějších z těchto algoritmů, včetně jejich zařazení.

1.1. Rozdělení prediktivních algoritmů

Existuje velké množství prediktivních algoritmů a najít kategorie, jež by je jednoznačně rozdělili, je velmi obtížné. I tak představují z hlediska řízení pohonů pouze jedinou skupinu. Rozdělení řídicích metod můžeme nalézt na obrázku 1.2.



Obrázek 1.2: Rozdělení algoritmů pro řízení pohonů dle funkčního principu [9]

Do skupiny lineárního řízení spadá například hojně využívané vektorové řízení. Tento přístup vyžaduje přítomnost transformace souřadnic a modulátoru pro řízení spínání měniče (ať už je použita metoda pulsní šířkové modulace (PWM) či prostorových vektorů (SVM)). Řízení v klouzavém režimu pak vzniklo s důrazem na robustnost řízení. Toto řídicí schéma bere v potaz nelineární charakter měniče a nevyžaduje tak modulátor. Pokročilé řídicí metody jako jsou neuronové sítě či fuzzy regulátory jsou využívány jen okrajově. Je tomu především z důvodů jejich velmi vysokých výpočetních nároků. [18],[28],[9]

Tím se dostáváme ke skupině prediktivních algoritmů. Ty budou v rámci této práce rozděleny ze tří hledisek:

- Délka predikčního horizontu
- Způsob řízení měniče
 - S modulátorem
 - Bez modulátoru
- Funkční princip (viz obrázek 1.2)

Rozdělení algoritmů dle funkčního principu bude věnována pozornost v následující podkapitole. Co se týká způsobu řízení měniče, lze obecně říci, že modulátor je nutné použít u deadbeat regulace a prediktivních algoritmů založených na modelu. Použití modulátoru v důsledku znamená konstantní spínací frekvenci. Ta je upřednostňována především u řízení pohonů nižších výkonů. Rušení vznikající konstantní spínací frekvencí lze totiž snáze odfiltrovat. Naopak u vysokých výkonů jsou požadovány co nejnížší spínací ztráty a tedy co nejnížší spínací frekvence. Je tedy výhodnější použít přímé řízení měniče s proměnnou spínací frekvencí. Délka predikčního horizontu je u prediktivních algoritmů pro řízení elektropohonů různá (nejčastěji pouze jeden krok), zpravidla však konečná. LQ regulátory s nekonečným predikčním horizontem se pro řízení pohonů nehodí, neboť principiálně není možné zavést omezení veličin motoru při zachování optimality. [30]

1.1.1. Deadbeat prediktivní regulace

První deadbeat algoritmy byly představeny v sedmdesátých letech minulého století. Jejich základní výhodou je velmi rychlá odezva řízení. Díky této vlastnosti jsou deadbeat algoritmy v oblasti elektropohonů používány především k regulaci proudů. Existují i návrhy k jejich použití spolu s algoritmem přímého řízení momentu asynchronního motoru, kde nahrazují PI regulátor vytvářející momentovou referenci.

V následujícím textu však bude představen pouze základní princip deadbeat regulace proudu. [29], [9]

Dynamiku statorových proudů v rotující d, q souřadné soustavě a v diskretním čase lze popsat rovnicí

$$\vec{i}_{sdq}(k+1) = \mathbf{f}(\vec{i}_{sdq}(k), \omega_m(k)) + \mathbf{B}\vec{u}_{sdq}(k), \quad (1.1)$$

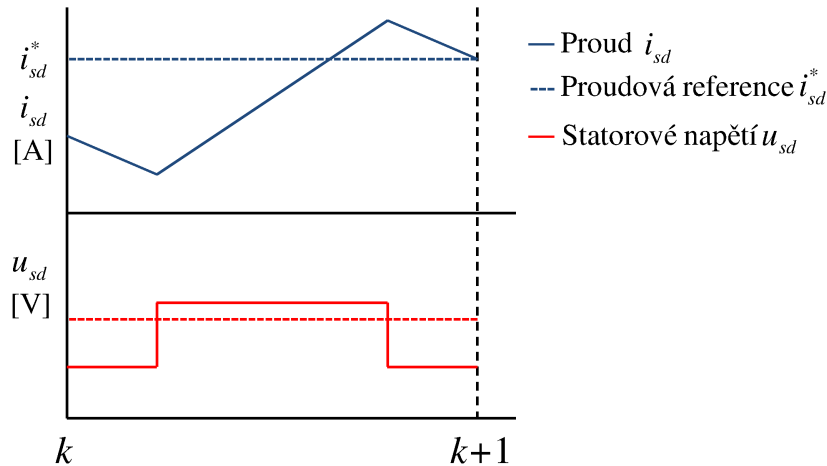
kde $\vec{i}_{sdq}$ je vektor statorových proudů, ω_m je mechanická úhlová rychlost, $\mathbf{f}(\vec{i}_{sdq}(k), \omega_m)$ je vektor dynamiky statorových proudů, $\mathbf{B} \in \mathbb{R}^{2 \times 2}$ je matice vstupů stavového popisu a $\vec{u}_{sdq}$ je vektor statorových napětí. V každém kroku je využito tohoto modelu systému k výpočtu akčního zásahu, tedy napětí, jež zajistí, že v následujícím kroku dosáhne regulovaný proud žádané hodnoty (viz případ regulace přímé složky proudu na obrázku 1.3). Tímto způsobem by mělo být dosaženo časově optimálního řízení. Naším požadavkem je tedy

$$\vec{i}_{sdq}(k+1) = \vec{i}_{sdq}^*(k), \quad (1.2)$$

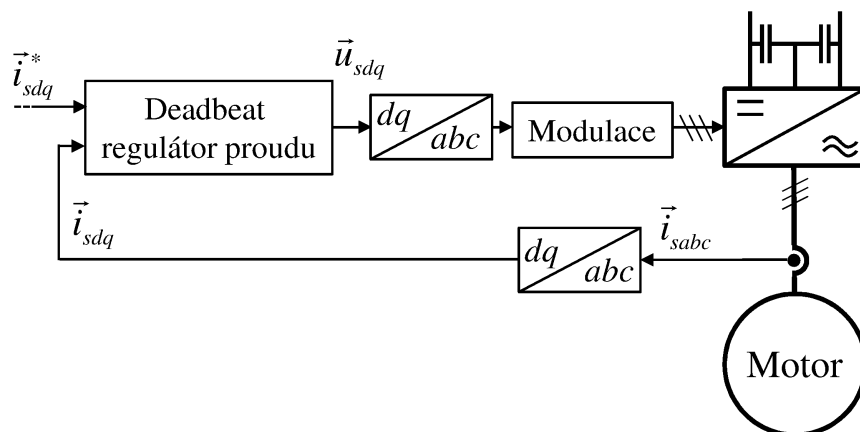
kde $\vec{i}_{sdq}^*$ je proudová reference. Pomocí vhodných úprav a dosazení do rovnice (1.1) dostáváme řídicí zákon

$$\vec{u}_{sdq}(k) = \mathbf{B}^{-1}\vec{i}_{sdq}^*(k) - \mathbf{B}^{-1}\mathbf{f}(\vec{i}_{sdq}(k|k-1), \omega_m(k)), \quad (1.3)$$

kde $\mathbf{B}^{-1}$ je inverze matice $\mathbf{B}$ a $\vec{i}_{sdq}(k|k-1)$ je predikce vektoru proudu. Mechanickou úhlovou rychlost ω_m není, vzhledem k její pomalé dynamice, třeba predikovat. Výsledné napětí je aplikováno na statorové vinutí motoru s pomocí modulátoru. Principiální blokové schéma deadbeat regulátoru proudu je na obrázku 1.4.



Obrázek 1.3: Princip deadbeat regulace proudu [9]



Obrázek 1.4: Blokové schéma deadbeat regulátoru proudu

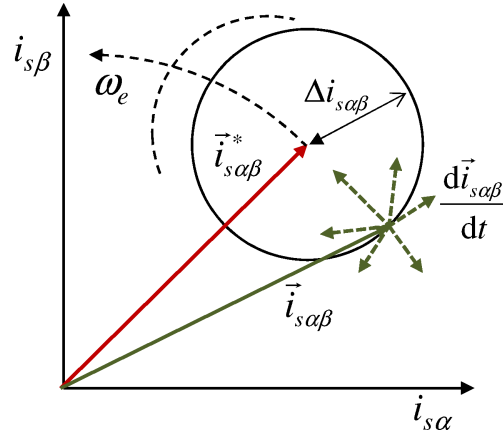
Bohužel tento algoritmus vykazuje i několik nevýhod. Díky své rychlosti se jedná o agresivní způsob regulace s vysokou citlivostí na šum. Algoritmus je rovněž silně závislý na přesnosti modelu dynamiky proudu. Jeho případná nepřesnost se může projevit nenulovou ustálenou odchylkou či v horším případě nestabilitou. Tento problém lze řešit úpravou deadbeat regulátoru na adaptivní algoritmus, přidáním integrátoru či s pomocí pozorovatele poruchy. V důsledku přítomnosti zpoždění způsobeného modulací a dobou výpočtu algoritmu může docházet k oscilacím. To lze řešit zavedením zpoždění již do modelu. Problematická je rovněž implementace napěťového omezení. Komplikovanější typ omezení, jež by dobře aproximoval kružnici napěťového omezení, totiž zavádí nechtěné vazby mezi regulátory přímé a kvadratické složky proudu. Obvykle se tak používá omezení typu nasycení pro každou složku zvlášť. Tyto úpravy však zvyšují výpočetní náročnost algoritmu. [23], [20], [2]

1.1.2. Prediktivní řízení založené na hysterezi

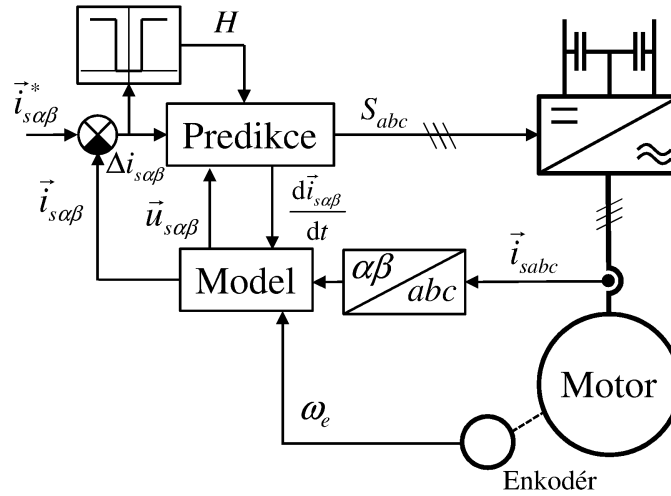
Cílem regulace s hysterezí je udržení regulované veličiny v definovaných mezích. Typickým zástupcem tohoto principu jsou především známé reléové regulátory (označované bang-bang). Právě jejich upravená verze byla vyvinuta pro regulaci proudů elektromotoru.

Tento algoritmus řídí spínání měniče přímo a nevyžaduje tak transformaci souřadnic. Princip regulace proudu je znázorněn na obrázku 1.5 a blokové schéma je na obrázku 1.6. Žádanou hodnotu představuje vektor $\vec{i}_{s\alpha\beta}^*$. Vzhledem k tomu, že regulace probíhá ve satorovém souřadném systému, dochází k rotaci tohoto vektoru elektrickou úhlovou rychlostí ω_e . Úkolem regulátoru je udržet vektor satorového proudu $\vec{i}_{s\alpha\beta}$ uvnitř kružnice, jež značí toleranci odchylky od reference $\Delta i_{s\alpha\beta}$. Jakmile

proudový vektor $\vec{i}_{s\alpha\beta}$ této hranice dosáhne, stane se signál H aktivní a musí dojít k přepnutí stavu měniče S_{abc} . Pro všechny stavy, do kterých je možno přepnout, je na základě modelu systému vypočtena trajektorie proudového vektoru s jednotkovou délkou predikčního horizontu. To je znázorněno pomocí zelených šipek ve směrech $d\vec{i}_{s\alpha\beta}/dt$. Ten z vektorů, který se dle predikce udrží ve vymezené oblasti nejdéle, je vybrán jako optimální a měnič je sepnut v odpovídající kombinaci. [16]



Obrázek 1.5: Princip prediktivní regulace proudu založené na hysterezi [16]



Obrázek 1.6: Blokové schéma histerezního prediktivního regulátoru proudu [16]

Tento přístup zajišťuje minimalizaci spínací frekvence a tím i spínacích ztrát na měniči, což je žádoucí zvláště u vysokých výkonů. Krom tohoto kritéria optimality existují modifikace tohoto algoritmu, jež například upřednostňují dosažení minimální hodnoty zvlnění momentu. [18]

Obecnou výhodou hysterezní regulace je, že není potřeba přesného modelu řízeného systému. Je však potřeba, aby regulátor reagoval dostatečně rychle na pře-

kročení mezí, v nichž se má regulovaná veličina nacházet. To však vyžaduje velmi krátkou vzorkovací periodu. Z tohoto pohledu je algoritmus spíše vhodný pro spojitou realizaci s pomocí operačních zesilovačů.

1.1.3. Prediktivní řízení založené na trajektorii

Strategie těchto prediktivních algoritmů spočívá v udržení stavu systému na předem stanovených trajektoriích uvnitř stavového prostoru. Právě díky znalosti trajektorií, po kterých se bude pohybovat stav systému, lze tyto algoritmy zařadit mezi prediktivní. Již z tohoto pohledu je jasné, že narozdíl od prediktivních algoritmů založených na hysterezi budou tyto algoritmy vyžadovat poměrně přesný model systému.

Algoritmus Direct Torque Control

Známy algoritmus Direct Torque Control (DTC) lze označit za kombinaci prediktivního řízení založeného na trajektorii a na hysterezi. Při vysvětlování jeho principu vyjdeme z diferenciální rovnice pro statorový magnetický tok $\vec{\Psi}_{s\alpha\beta}$, tedy

$$\frac{d\vec{\Psi}_{s\alpha\beta}}{dt} = \vec{u}_{s\alpha\beta} - R_s \vec{i}_{s\alpha\beta}, \quad (1.4)$$

kde $\vec{u}_{s\alpha\beta}$ je statorové napětí, R_s je statorový odpor a $\vec{i}_{s\alpha\beta}$ je statorový proud. Po zanedbání statorového odporu, což si u motorů vyšších výkonů můžeme dovolit, můžeme situaci zjednodušit, tedy

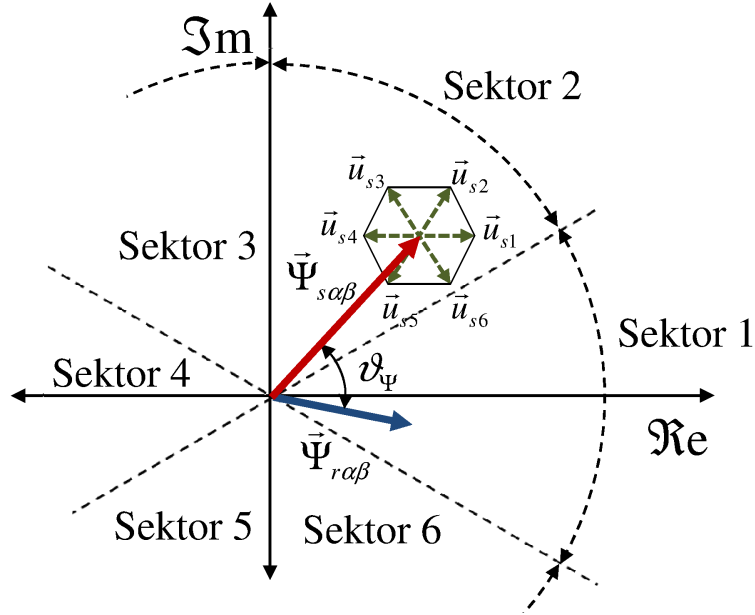
$$\Delta \vec{\Psi}_{s\alpha\beta}(t) = \vec{\Psi}_{s\alpha\beta}(t + T_s) - \vec{\Psi}_{s\alpha\beta}(t) \doteq \vec{u}_{s\alpha\beta} T_s, \quad (1.5)$$

kde T_s je vzorkovací perioda a t je čas. Lze tedy říci, že statorový magnetický tok je, více či méně, závislý pouze na zvoleném vektoru statorového napětí. Amplituda statorového magnetického toku je při regulaci držena blízko žádané hodnotě $|\Psi_{s\alpha\beta}^*|$. Vektor $\vec{\Psi}_{s\alpha\beta}$ je tak nucen opisovat kruhovou trajektorii (proto zařazení mezi algoritmy založené na trajektorii). Dále vyjdeme z momentové rovnice. Tu lze pro synchronní i pro asynchronní motor psát ve tvaru

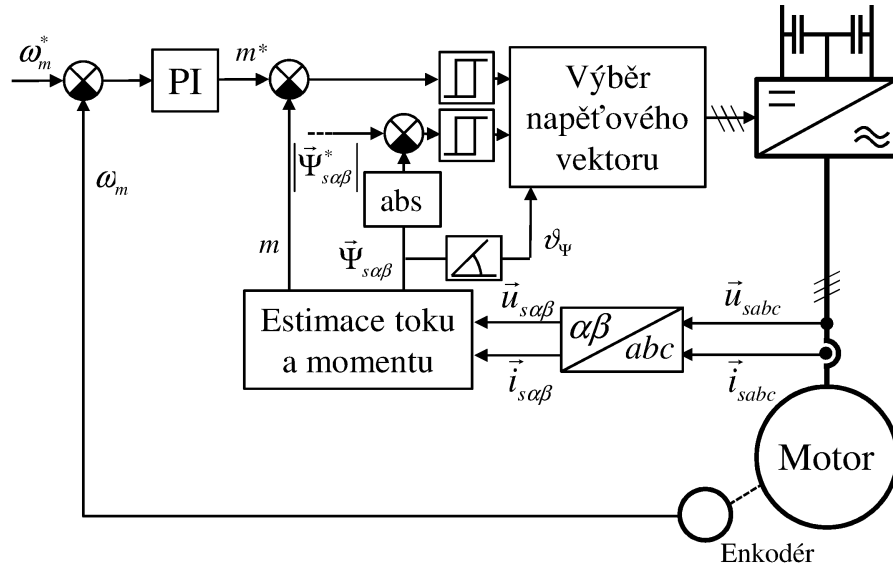
$$m = K_m \left| \vec{\Psi}_{s\alpha\beta} \right| \left| \vec{\Psi}_{r\alpha\beta} \right| \sin(\vartheta_\Psi), \quad (1.6)$$

kde K_m je konstanta závislá na typu motoru, $\vec{\Psi}_{r\alpha\beta}$ je rotorový magnetický tok a ϑ_Ψ je úhel mezi rotorovým a statorovým magnetickým tokem. Jelikož je dynamika rotorového toku pomalejší než toho statorového, lze rotorový tok v rovnici (1.6)

prohlásit za konstantu. Moment se tedy stává čistě funkcí úhlu mezi těmito toky. Velikost tohoto úhlu je tedy řídicí veličinou. [28]



Obrázek 1.7: Princip řízení satorového toku algoritmem DTC



Obrázek 1.8: Blokové schéma prediktivního algoritmu DTC

Tento algoritmus využívá přímé řízení měniče a neobsahuje tedy modulátor. U dvoustavového měniče pak lze v každém kroku aplikovat šest nenulových napětových vektorů $\vec{u}_{s1}, \vec{u}_{s2}, \dots, \vec{u}_{s6}$ a dva nulové. Vhodný vektor satorového napětí je vybrán z tabulky na základě informace o právě aktivním sektoru (viz obrázek 1.7) a potřeby zvýšit či snížit velikost momentu a satorového magnetického toku.

Výsledné blokové schéma je na obrázku 1.8. Otáčková regulační smyčka je obvykle realizována pomocí PI regulátoru. Ten na základě žádané hodnoty rychlosti ω_m^* a skutečné hodnoty rychlosti ω_m vytváří momentovou referenci m^* . Nejdůležitější výhodou DTC algoritmu je jeho vynikající dynamická odezva momentu. K tomu je však potřeba dosáhnout velmi krátké vzorkovací periody, a to až 25 μs , což může být problematické.

Algoritmus Direct Mean Torque Control

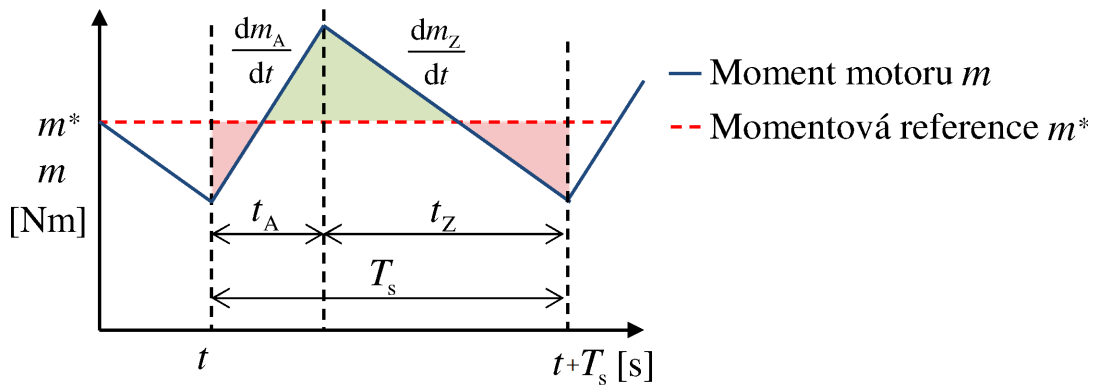
Jednou z mnoha modifikací DTC algoritmu je Direct Mean Torque Control (DMTC), který lze zařadit čistě mezi algoritmy založené na trajektorii. DMTC opět řídí měnič přímo, avšak spínací frekvence je konstantní a vzorkovací perioda tak nemusí být natolik nízká jako v případě DTC. To je také hlavní výhodou DMTC algoritmu.

Pakliže budeme uvažovat případ dvoustavového měniče, lze na stator motoru přivést jeden ze šesti nenulových (aktivních) vektorů nebo jeden ze dvou nulových. Stejně jako u DTC je vytvořena tabulka, kde je v závislosti na sektoru uveden vliv aktivních napěťových vektorů na moment a statorový magnetický tok. V každém kroku regulace jsou potom vybrány vhodné aktivní vektory a je pro ně vypočtena doba t_A . Jak je vidět z obrázku 1.9, je to čas, po který bude v rámci doby jednoho vzorku aktivní napěťový vektor přiveden na stator motoru. Predikční horizont má tedy jednotkovou délku. Po zbytek doby vzorkovací periody T_s , tedy po dobu t_Z , je na stator přiveden nulový napěťový vektor. Cílem je, aby bylo dosaženo požadované střední hodnoty momentu m^* , dle vztahu

$$m^* = \frac{m_A t_A + m_Z t_Z}{T_s}, \quad (1.7)$$

kde m_A je moment motoru při přivedení aktivního a m_Z při přivedení nulového napěťového vektoru. Pokud se tedy nacházíme v ustáleném stavu, měla by mít zelená a červená oblast na obrázku 1.9 stejnou plochu.

Hodnota statorového magnetického toku je řízena tak, že z vybraných aktivních vektorů, pro něž byla vypočtena doba t_A , je vybrán ten, jež způsobí menší odchýlení magnetického toku od jeho požadované kruhové trajektorie. Tato informace je získána z modelu statorového magnetického toku. Priorita je však dána řízení momentu. [21]

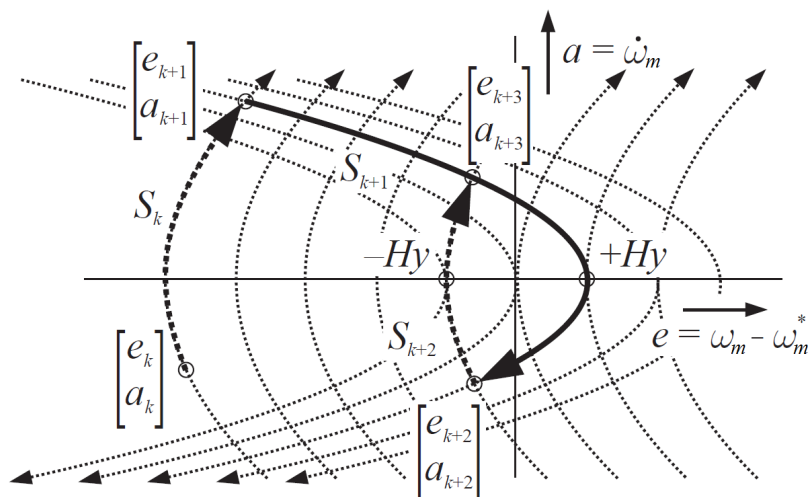


Obrázek 1.9: Způsob řízení momentu pomocí algoritmu DMTC

Algoritmus Direct Speed Predictive Control

Jako nejtypičtějšího zástupce prediktivních algoritmů založených na trajektorii uvedeme algoritmus Direct Speed Predictive Control (DSPC). Ani DSPC nemá modulátor a při řízení tak využívá celkem sedmi unikátních napěťových vektorů, jež lze vytvořit s pomocí dvoustavového třífázového měniče. To, jaký tyto napěťové vektory mají vliv na rychlost rotoru ω_m , můžeme zobrazit na obrázku 1.10, kde je na vodorovné ose vynesena regulační odchylka e a na svislé zrychlení a . Pokud uvažíme, že zátěžný moment a moment setrvačnosti jsou konstantní, budou trajektorie systému pro jednotlivé napěťové vektory nabývat tvar parabol. Pokud se například v čase k nachází stav systému v bodě $[e_k, a_k]$ (viz obrázek 1.10), je zvolen spínací stav měniče S_k , který zvyšuje moment systému a tím motor zrychluje. Ve chvíli, kdy je dosaženo bodu $[e_{k+1}, a_{k+1}]$ je zvolen spínací stav S_{k+1} , jež snižováním momentu přivede systém na trajektorii, která protíná bod $[+H_y, 0]$. Tohoto bodu je tak dosaženo v nejkratším možném čase. Cílem je tedy časově optimální řízení. Hodnota H_y označuje horní mez oblasti, v níž je držena žádaná hodnota. Po dosažení bodu $[e_{k+2}, a_{k+2}]$ se systém zacyklí na trajektorii mezi tímto bodem a bodem $[e_{k+3}, a_{k+3}]$ za použití spínacích stavů S_{k+1} a S_{k+3} . Z principu tedy není možno dosáhnout nulové ustálené odchylky, neboť by byla potřeba nekonečná spínací frekvence. Místo toho se v systému nachází hystereze.

Výhoda tohoto algoritmu je, že nevyžaduje kaskádní strukturu a chybí tak zde podřízená proudová smyčka vyžadující krátkou vzorkovací periodu. Nevýhodou je silná závislost kvality regulace na přesnosti modelu a poměrně vysoké výpočetní nároky.



Obrázek 1.10: Princip řízení s pomocí DSPC algoritmu [22]

1.1.4. Prediktivní řízení založené na modelu

První algoritmy prediktivního řízení založeného na modelu (v angličtině Model Predictive Control či MPC), byly představeny již v sedmdesátých letech minulého století. Jako jeden z mála typů prediktivní regulace se dočkal mnoha úspěchů na poli praktických aplikací. Krom akademické půdy jsou MPC algoritmy populární obzvláště v chemickém průmyslu, kde (díky dlouhým časovým konstantám) nepředstavují vysoké výpočetní nároky MPC algoritmů problém. Existuje však snaha rozšířit tyto algoritmy i do jiných oblastí. Řízení elektromotorů přitom není výjimkou. Motivace k tomu byla popsána v úvodu kapitoly 1. V souhrnu jsou výhody MPC algoritmů oproti PID regulátorům:

- Snadná implementace pro MIMO systémů a není nutné použít kaskádní řídicí struktury.
- Jednoduchý a systematický způsob zavedení omezení vstupů, výstupů a stavů systému.
- Nastavení regulátoru je intuitivní, zvládnutelné i člověkem bez hlubších znalostí teorie regulace.
- Přirozeně obsahuje možnost feedforward vazby vhodné ke kompenzaci měřených poruch.
- Některé algoritmy umožňují řízení systémů s neminimální fází či s dopravním zpožděním.

Ač se jedná o skutečně rozsáhlou rodinu algoritmů založených na rozmanitých principech, všechny mají společné následující znaky:

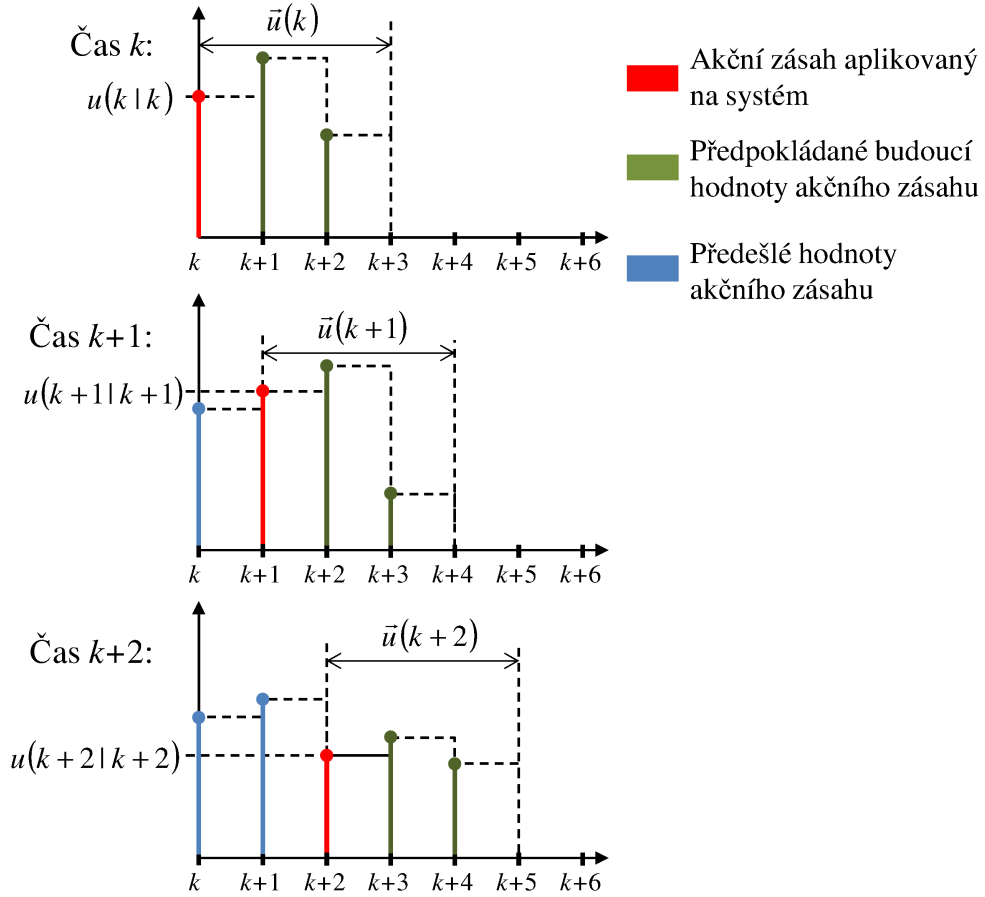
- Explicitně zadaný model sloužící k predikci stavů a výstupů systému a využití současných i minulých hodnot vstupů, výstupů a stavů systému.
- Výpočet optimální řídicí sekvence (akčního zásahu) skrze minimalizaci účelové funkce.
- Strategie ustupujícího horizontu.

Díky posledně jmenovanému se MPC algoritmy občas označují jako Receding Horizon Predictive Control (RHPC) algoritmy. Princip ustupujícího horizontu pro tři po sobě jdoucí kroky regulace s délkou predikčního horizontu tři, je naznačen na obrázku 1.11. V každém kroku regulace je vypočtena kompletní sekvence akčního zásahu $\vec{u}(k)$, jež by v rámci predikčního horizontu minimalizovala hodnotu účelové funkce. Pro tuto sekvenci můžeme psát

$$\vec{u}(k) = \{u(k|k), u(k+1|k), \dots, u(k+N-1|k)\}, \quad (1.8)$$

kde $u(k+i|k)$ znamená hodnota akčního zásahu v kroku $k+i$, vypočtená v čase k , kdy $i \in \mathbb{N}_0$ a N je délka predikčního horizontu. Z této sekvence (anglicky optimizer) se však na vstup systému pošle pouze první hodnota. Zbývající hodnoty akčního zásahu v optimalizační sekvenci zůstanou nepoužity, neboť v dalším kroku regulace se celý proces opakuje a je vypočítána nová sekvence. To se zdá být plýtváním výpočetního výkonu, nicméně pokud by byla využita kompletní sekvence, jednalo by se o řízení v otevřené smyčce. To by samozřejmě přineslo řadu nevýhod. Především by tím trpěla možnost kompenzace poruchy. Platí tedy, že strategie ustupujícího horizontu kombinuje výhody zpětnovazební regulace s optimální regulací v otevřené smyčce. [8], [9]

Tato metoda je přitom blízká lidskému způsobu řízení. Tento fakt lze demonstrovat na řízení automobilu. Z něj máme výhled pouze do omezené vzdálenosti (predikčního horizontu). Na základě našich zkušeností (modelu chování systému) a aktuální pozice a rychlosti (současný stav systému) pak kontinuálně stanovujeme natočení volantu (akční zásah). V každém okamžiku jsme přitom schopni reagovat na nečekané události (předpokládané budoucí hodnoty akčního zásahu zůstanou v takovém okamžiku nepoužity). A ačkoli sledujeme vyšší cíl přemístění se z bodu A do vzdáleného bodu B (minimalizace účelové funkce), svůj automobil řídíme dle informací z mnohem kratší vzdálenosti (predikčního horizontu).



Obrázek 1.11: Strategie ustupujícího horizontu pro predikční horizont $N = 3$

Účelová funkce, jejíž hodnota je v rámci predikčního horizontu N minimalizována, je definována v závislosti na daném algoritmu. Pro všechny MPC algoritmy principiálně platí

$$J_N(\vec{x}(k)) = P(\vec{x}(k+N|k)) + \sum_{i=k+1}^{k+N-1} L_f(\vec{x}(i|k), \vec{u}(i|k)), \quad (1.9)$$

kde P a L_f jsou penalizační funkce váhující predikované hodnoty stavů $\vec{x}(i|k)$ a optimalizující sekvence akčních zásahů $\vec{u}(i|k)$, kde $i \in \mathbb{N}_0$. Funkce P penalizuje stav na konci predikčního horizontu. Obě penalizační funkce obvykle mívají tvar $\mathfrak{L}_n$ -normy (bližší popis v následující kapitole). Pomocí účelové funkce, tedy zákona optimality pro náš regulátor, jsme schopni určit chování regulační smyčky. Nastavení je přitom intuitivní, neboť lze jednoduše předvídat odezvu v chování regulační smyčky v závislosti na změně parametrů v účelové funkci. Důležitou výhodou je rovněž možnost rozšířit stavový vektor o nové proměnné a zahrnout je tím do výpočtu akčního zásahu. Tímto způsobem můžeme jednoduše určit primární hledisko optimality

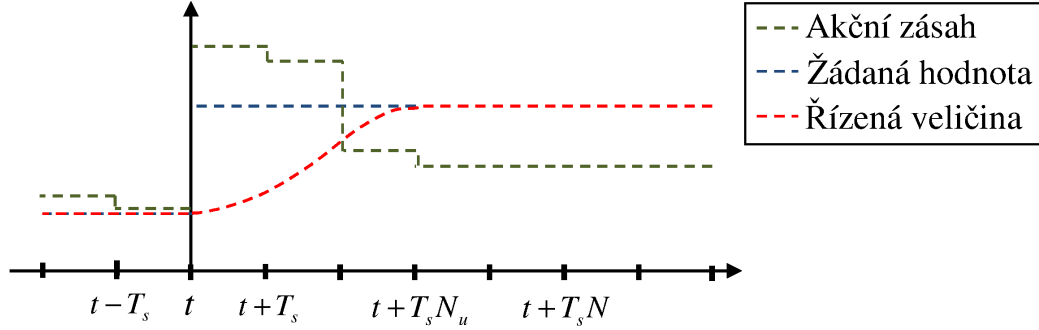
naší regulace. To může být časově optimální řízení, ale také například minimalizace ztrát v systému.

Značná část diverzity mezi MPC algoritmy je tvořena typem použitého modelu. Každý z typů modelů pak výslednému algoritmu propůjčuje své výhody i nevýhody. Přesnost modelu je pak důležitá pro výslednou kvalitu řízení. Adaptivní řízení s MPC algoritmy, jež by případné nepřesnosti kompenzovalo, je možné, ale mnohdy výrazně zvyšuje výpočetní náročnost. Fázi získávání modelu systému je tak třeba věnovat zvýšenou pozornost. Na akademické půdě se s oblibou používá stavový popis. Z něj vychází například algoritmus Linear Model Predictive Control. Stavový popis se hodí pro modelování lineárních MIMO systémů. Nevýhodou však je, že nelze jednoduše modelovat dopravní zpoždění (zvyšovala by se dimenze stavového prostoru). V praxi se tak spíše uplatňují jiné typy modelů. Například algoritmus General Predictive Control (GPC) využívá diskrétního operátorového přenosu. Tento typ modelu je již postaven na - v praxi často užívaném - modelování pomocí časových konstant, zesílení a dopravních zpoždění. Existují však i algoritmy postavené na skokové či impulzové odezvě jako modelu systému. Tyto modely mají výhodu, že je lze v praxi získat relativně snadno. Tyto modely využívá například algoritmus Dynamic Matrix Control (DMC), který je postaven na přechodové charakteristice nebo Model Algorithmic Control (MAC), jež používá impulzovou charakteristiku. Nakonec existují i algoritmy pro řízení nelineárních systémů, jako je Non-Linear MPC (NLMPC), který vychází z diferenčních rovnic systému. [8], [15], [30]

Pro lineární systémy lze predikci chování systému na základě modelu získat superpozicí volné a nucené odezvy. V případě volné odezvy se uvažuje nulová hodnota akčního zásahu od současného okamžiku dále s tím, že je počítáno s aktuálními, případně minulými stavy, vstupy a výstupy systému. U nucené odezvy je předpokládán nulový počáteční stav, ale budoucí hodnoty akčního zásahu jsou již obecně nenulové a jsou počítány s cílem minimalizace účelové funkce. To, jakým způsobem je do výpočtu zahrnuta minulost, závisí na použitém typu modelu. U algoritmů vycházejících ze stavového popisu je minulost uložena v aktuálním stavu systému, zatímco například algoritmy vycházející z operátorového přenosu musí ukládat minulé hodnoty vstupů a výstupů. [18]

Délka predikčního horizontu se obvykle nastavuje experimentálně. Spodní hranice je ovlivněna dynamikou systému, neboť predikční horizont by měl být dostatečně dlouhý, aby regulátor viděl možnost ji ovlivnit skrze akční zásah. Horní hranice je pak dána pouze výpočetními možnostmi hardwaru. Právě výpočetní náročnost je důležitým faktorem při volbě délky horizontu. Proto byl zaveden takzvaný řídicí horizont N_u , kdy platí $N_u \leq N$, s nímž lze dosáhnout výrazné úspory výpočetního

výkonu, neboť se snižuje počet stupňů volnosti. Po překročení N_u kroků se totiž předpokládá dosažení ustáleného stavu a akční zásah se pak nemění (viz obrázek 1.12).



Obrázek 1.12: Princip řídicího horizontu N_u [18]

Algoritmus Model Predictive Direct Torque Control

Mezi MPC algoritmy lze zařadit například Model Predictive Direct Torque Control (MPDTC). Jak název napovídá, jedná se o modifikovaný DTC algoritmus. I zde je statorový magnetický tok a moment držen v definovaných mezích pomocí hystereze. Byla však přidána filozofie RHC. Cílem, stejně jako u DTC algoritmu, bylo dosažení nízké spínací frekvence. [11], [24]

Blokové schéma řízení střídavého pohonu s MPDTC algoritmem je na obrázku 1.13. Stejně jako u DTC algoritmu i zde chybí modulátor. Algoritmus lze použít nejen pro dvoustavový, ale i třístavový trojfázový měnič, kdy je možno vytvořit celkem 27 vektorů statorového napětí. Zde však musí MPDTC algoritmus regulovat krom magnetického toku a momentu také potenciál neutrálního bodu u_{np} . Ten se nachází mezi filtračními kondenzátory stejnosměrného meziobvodu.

MPDTC algoritmus byl vytvořen pro řízení asynchronního motoru. Jeho model je popsán pomocí nelineárních diferenčních rovnic pro složky statorového a rotorového magnetického toku $\Psi_{s\alpha}$, $\Psi_{s\beta}$, $\Psi_{r\alpha}$ a $\Psi_{r\beta}$, které spolu s napětím neutrálního bodu u_{np} tvoří vektor stavů systému. K jejich modelování je zapotřebí znát hodnotu rotorových otáček ω_m a napětí meziobvodu U_{dc} . Regulované veličiny, tedy výstupy modelu systému, jsou moment motoru m , amplituda statorového magnetického toku $|\Psi_s|$ a napětí u_{np} .

Jak už bylo řečeno, cílem algoritmu je dosažení co nejnižší spínací frekvence. Průměrná spínací frekvence f na měniči lze popsat pomocí rovnice

$$f = \lim_{k \rightarrow \infty} \frac{1}{kT_s} \sum_{i=1}^k \|s(i) - s(i-1)\|_1, \quad (1.10)$$

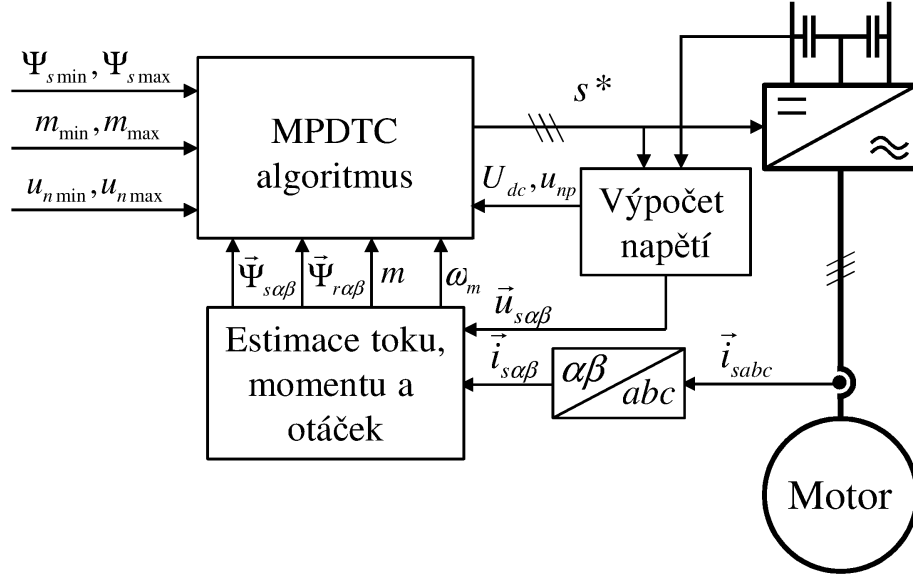
kde $s \in \{-1, 0, 1\}$ je spínací kombinace na měniči a T_s je vzorkovací perioda jejíž hodnota je obdobná jako u DTC algoritmu, tedy až 25 μs . U aplikací s velmi vysokými výkony, kde tento algoritmus nachází uplatnění, je průměrná spínací frekvence velmi nízká, někdy až v hodnotách 200-400 Hz. To by však při uvážení vztahu (1.10) vedlo na nutnost použití velmi dlouhého predikčního horizontu, což by znamenalo přílišné výpočetní nároky. Účelová funkce je tedy použita v poněkud jiné formě než je rovnice (1.10) tak, aby predikční horizont mohl být mnohem kratší. Algoritmus MPDTC pak lze shrnout do následujících kroků.

1. Na základě poslední spínací kombinace na měniči $s(k-1)$ je určeno všech $i \in \mathbb{N}$ použitelných (myšleno v tomto kroku) spínacích sekvencí $S_i(k) = [s_i(k), \dots, s_i(k+N-1)]$, kde N je délka predikčního horizontu.
2. Pro všechny tyto sekvenční jsou vypočteny trajektorie momentu m , amplitudy statorového magnetického toku $|\Psi_s|$ a napětí u_{np} v rámci následujících N kroků.
3. Je vybráno $j \leq i$ spínacích sekvencí, pro něž predikované trajektorie regulovaných veličin m , $|\Psi_s|$ a u_{np} nepřekročí v celé délce predikčního horizontu stanovené meze. Pro vhodné spínací sekvenční (řekněme jim kandidátské) tak dle predikcí musí platit $m(l) \in \langle m_{\min}, m_{\max} \rangle$, $|\Psi_s|(l) \in \langle |\Psi_s|_{\min}, |\Psi_s|_{\max} \rangle$ a $u_{np}(l) \in \langle u_{np\min}, u_{np\max} \rangle$, kde $l = [k, \dots, k+N-1]$, kde dolní index min značí dolní mez a max horní mez dané veličiny.
4. Pro všechny kandidátské spínací sekvenční jsou extrapolovány trajektorie m , $|\Psi_s|$ a u_{np} od kroku $k+N$ dále. Uvažuje se přitom konstantní spínací kombinace na celém horizontu extrapolace (konstantní statorové napětí). Délka horizontu extrapolace je omezena dosažením hodnoty veličiny její mezní hodnoty. Důležitá je pak informace o minimálním počtu kroků n_j , jež pro danou spínací sekvenční stráví všechny extrapolované veličiny uvnitř svých mezí, a to od současnosti k .
5. Pro každou spínací sekvenční je spočítána hodnota účelové funkce $J_j = c_j/n_j$, kde

$$c_j = \sum_{l=k}^{k+N-1} \|s_j(l) - s_j(l-1)\|_1 \quad (1.11)$$

je celkový počet změn sepnutí měniče za danou spínací sekvenční j .

6. Je vybrána optimální spínací sekvenční s minimální hodnotou účelové funkce J_j . Následně je, dle principů RHPC, z této optimální spínací sekvenční použita pouze první spínací kombinace (s^* na obrázku 1.13).



Obrázek 1.13: Blokové schéma MPDTC algoritmu [11]

Z předchozího popisu algoritmu je jasné, že s délkou predikčního horizontu prudce stoupá počet spínacích sekvencí, pro něž je nutno predikovat trajektorie. Tím pádem značně narůstá výpočetní náročnost algoritmu. Avšak i při volbě délky predikčního horizontu $N = 1$ dosahuje MPDTC algoritmus výrazného snížení spínací frekvence, a to díky extrapolaci trajektorií regulovaných veličin. Dle autorů pak lze dosáhnout až 50% snížení průměrné spínací frekvence oproti DTC algoritmu. Z něj je přitom zachována velmi rychlá odezva řízení momentu. Výhodou oproti DTC je rovněž lepší schopnost držet regulované veličiny uvnitř svých mezí.

Algoritmus Model Predictive Current Control

Dalším MPC algoritmem používaným při řízení elektromotorů je Predictive Current Control (PCC). Jedná se o spíše jednodušší prediktivní algoritmus. Existuje varianta algoritmu jako pro řízení v d, q , tak v α, β souřadné soustavě (zde představíme druhou variantu). Stejně jako MPDTC ani PCC nevyužívá modulace statorového napětí. Cílem algoritmu je minimalizace odchylky vektoru statorového proudu od jeho reference, tedy

$$J_N = \left| \vec{i}_{s\alpha\beta}^*(k+1|k) - \vec{i}_{s\alpha\beta}(k+1|k) \right|, \quad (1.12)$$

kde $\vec{i}_{s\alpha\beta}^*(k+1|k)$ je vektor proudové reference a $\vec{i}_{s\alpha\beta}(k+1|k)$ je vektor predikce statorového proudu v následujícím kroku. Predikční horizont má tedy jednotkovou délku. Pokud je vzorkovací perioda dostatečně krátká, lze situaci zjednodušit a nahradit predikci proudové reference její současnou hodnotou, tedy $\vec{i}_{s\alpha\beta}^*(k) \doteq \vec{i}_{s\alpha\beta}^*(k+1|k)$. Výsledné dopravní zpoždění je pak zanedbatelné. Blokové schéma řízení s PCC regu-

látořem je na obrázku 1.14. Algoritmus PCC regulátoru lze shrnout do následujících kroků:

1. Nejprve jsou stanoveny predikce proudů $\vec{i}_{s\alpha\beta}(k+1|k)$ pro všechny vektory statorového napětí. V případě dvoustavového třířázového měniče se jedná o sedm unikátních vektorů. Například pro ACIM lze predikci proudu stanovit pomocí rovnice

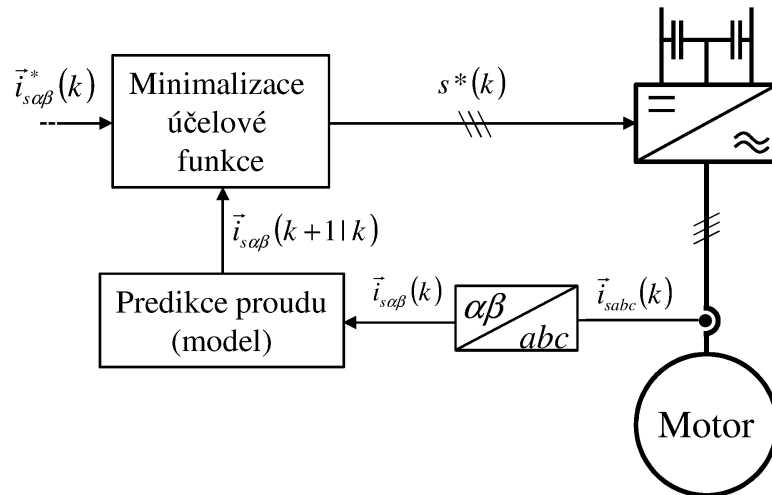
$$\vec{i}_{s\alpha\beta}(k+1|k) = \left(1 - \frac{T_s R_s}{L}\right) \vec{i}_{s\alpha\beta}(k) + \frac{T_s}{L} (\vec{u}_{s\alpha\beta}(k) - \vec{e}_{s\alpha\beta}(k|k)), \quad (1.13)$$

kde T_s je vzorkovací perioda, R_s je statorový odpor, L je statorová indukčnost, $\vec{u}_{s\alpha\beta}$ je jeden ze sedmi unikátních napěťových vektorů a $\vec{e}_{s\alpha\beta}$ je vektor zpětného elektromotorického napětí. To lze modelovat rovnicí

$$\vec{e}_{s\alpha\beta}(k-1|k) = \vec{u}_{s\alpha\beta}(k-1) - \frac{L}{T_s} \vec{i}_{s\alpha\beta}(k) - \left(R_s - \frac{L}{T_s}\right) \vec{i}_{s\alpha\beta}(k-1). \quad (1.14)$$

Současnou hodnotu $\vec{e}_{\alpha\beta}(k|k)$ potřebnou v rovnici (1.13) lze extrapolovat z jeho předchozích hodnot nebo v případě dostatečně krátké vzorkovací periody prohlásit $\vec{e}_{\alpha\beta}(k|k) \doteq \vec{e}_{s\alpha\beta}(k-1|k)$.

2. Pro každý unikátní vektor statorového napětí, respektive predikci proudu dle rovnice (1.13), je dle rovnice stanovena hodnota účelové funkce (1.12).
3. Napěťový vektor s nejmenší hodnotou účelové funkce J_N je považován za optimální. Jemu odpovídající spínací kombinace měniče s^* je aplikována na systém.



Obrázek 1.14: Blokové schéma řízení střídavého pohonu pomocí PCC algoritmu [28]

Největší výhodou PCC algoritmu je absence nutnosti modulátoru a Parkovy transformace (regulace probíhá ve statorových souřadnicích). Oproti jiným MPC algoritmům tedy vyniká nízkými výpočetními nároky. Nevýhodou je poměrně vysoké zvlnění proudu pro delší vzorkovací periody (cca 100 μs). Zkrácením vzorkovací periody až k hodnotám 25 μs sice zvlnění klesá, nicméně takto stoupá spínací frekvence měniče. [28]

Algoritmus Generalized Predictive Control

Dalším MPC algoritmem, který zde bude zmíněn je Generalized Predictive Control (GPC) algoritmus, který vznikl v roce 1987. Jak už název napovídá, jeho formulace je velmi obecná a algoritmus je tak možné nasadit v zásadě pro jakýkoli lineární systém. S výhodou se dá použít pro řízení MIMO systémů, systémů s dopravním zpožděním či s neminimální fází. Ačkoli se GPC při řízení elektropohonů používá jen okrajově, v jiných odvětvích se jedná o velmi populární algoritmus, který byl s úspěchem nasazen v mnoha průmyslových aplikacích. Proto zde bude uveden jeho princip alespoň v nejzákladnější podobě, tedy pro řízení SISO systému bez použití filtrů či měřených poruch. [8], [18]

Typ modelu, na němž je GPC založeno, se nazývá Controlled Autoregressive Integrated Moving Average (CARIMA), což je forma operátorového přenosu. Nejdříve zavedme polynomy

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_naq^{-na}, \quad (1.15)$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \dots + b_n bq^{-nb}, \quad (1.16)$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_n cq^{-nc}, \quad (1.17)$$

kde q^{-1} je operátor zpoždění o jeden krok (obecně platí $y(k)q^{-1} = y(k-1)$). Pak je CARIMA model definován jako

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + C(q^{-1})\frac{\xi(k)}{\Delta}, \quad (1.18)$$

kde $y(k)$ je výstup, $u(k)$ je vstup, $\xi(k)$ je šum vstupující do systému a člen $\Delta = 1 - q^{-1}$. Jestliže platí, že $\xi(k)$ je bílý šum, pak lze prohlásit $C = 1$. Jako další zavedme prediktor j -té budoucí hodnoty výstupu systému, ve tvaru

$$y(k+j|k) = E_j(q^{-1})B(q^{-1})\Delta u(k+j-1) + F_j(q^{-1})y(k), \quad (1.19)$$

kde polynomy $E_j(q^{-1}) = e_{j,0} + e_{j,1}q^{-1} \dots + e_{j,j-1}q^{-(j-1)}$ a $F_j(q^{-1}) = f_{j,0} + f_{j,1}q^{-1} \dots + f_{j,na}q^{-na}$ jsou získány řešením diofantické rovnice

$$1 = E_j(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}). \quad (1.20)$$

Zavedeme-li polynom $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$ a pro zjednodušení budeme předpokládat nulové dopravní zpoždění, pak můžeme psát vektor predikcí výstupu

$$\underbrace{\begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+N|k) \end{bmatrix}}_{\vec{y}_N} = \underbrace{\begin{bmatrix} G_1(q^{-1}) \\ G_2(q^{-1}) \\ \vdots \\ G_N(q^{-1}) \end{bmatrix}}_{\vec{G}(q^{-1})}^T \underbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix}}_{\vec{u}} + \underbrace{\begin{bmatrix} F_1(q^{-1}) \\ F_2(q^{-1}) \\ \vdots \\ F_N(q^{-1}) \end{bmatrix}}_{\vec{F}(q^{-1})} y(k), \quad (1.21)$$

kde N je délka predikčního horizontu. Jak bylo řečeno v úvodu kapitoly 1.1.4, lze predikci chování systému složit z takzvané nucené a volné odezvy. Nucená odezva počítá s minulými a současnými hodnotami vstupu a výstupu systému, které, jelikož je již nelze změnit, nejsou předmětem optimalizace. Rovnici (1.21) tedy rozdělíme následujícím způsobem

$$\vec{y}_N = \underbrace{\mathbf{G}\vec{u}}_{\text{nucená odezva}} + \underbrace{\vec{F}(q^{-1})y(k) + \vec{G}'(q^{-1})\Delta u(k-1)}_{\text{volná odezva}} = \mathbf{G}\vec{u} + \vec{f}, \quad (1.22)$$

kde rozdělením vektoru polynomů $\vec{G}(q^{-1})$ vznikla matice $\mathbf{G} \in \mathbb{R}^{N \times N}$, jež respektuje minulost a vektor polynomů $\vec{G}'(q^{-1})$ budoucnost. Tato rovnice již představuje výslednou podobu prediktoru.

Účelová funkce pak nabývá podoby kvadratického kritéria s penalizací řídicí odchylky a změny akčního zásahu, tedy

$$J = \sum_{j=1}^N \mu_j (y(k+j|k) - w(k+j))^2 + \sum_{j=1}^{N_u} \lambda_j (\Delta u(k+j-1))^2, \quad (1.23)$$

kde $w(k+j)$ je žádaná hodnota, μ_j a λ_j jsou penalizační koeficienty odchylky a změny akčního zásahu v j -tém kroku a N_u je řídicí horizont.

Nyní již zbývá odvodit způsob nalezení optimalizující sekvence akčního zásahu. Naším cílem je přitom minimalizace hodnoty účelové funkce (1.23). Pokud budeme

uvažovat zjednodušující předpoklady $\mu_j = 1$ a $\lambda_j = \lambda$ ve všech $k + j$ krocích, pak dosazením rovnice (1.22) do rovnice (1.23) získáme

$$\begin{aligned} J &= \left(\mathbf{G}\vec{u} + \vec{f} - \vec{w} \right)^T \left(\mathbf{G}\vec{u} + \vec{f} - \vec{w} \right) + \lambda \vec{u}^T \vec{u} = \\ &= \vec{u}^T \left(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I} \right) \vec{u} + 2\vec{u}^T \mathbf{G}^T \left(\vec{f} - \vec{w} \right) + \left(\vec{f} - \vec{w} \right)^T \left(\vec{f} - \vec{w} \right), \end{aligned} \quad (1.24)$$

kde $\vec{w} = [w(k+1), \dots, w(k+N)]^T$ je vektor budoucích hodnot reference. Optimum pak nalezneme skrze derivaci rovnice (1.24) podle vektoru $\vec{u}$, tedy

$$\frac{\partial J}{\partial \vec{u}} = 2 \left(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I} \right) \vec{u} + 2\mathbf{G}^T \left(\vec{f} - \vec{w} \right) \stackrel{!}{=} 0. \quad (1.25)$$

Řešením pro $\vec{u}$ získáme optimalizující sekvenci akčního zásahu, tedy

$$\vec{u} = \left(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I} \right)^{-1} \mathbf{G}^T \left(\vec{w} - \vec{f} \right). \quad (1.26)$$

Dle principu RHPC je jako akční zásah v kroku k použita první hodnota ze sekvence $\vec{u}$. Analytický způsob řešení je výpočetně relativně nenáročný, nicméně tímto způsobem není možné jednoduše zahrnout omezení vstupů či výstupů při zachování optimality řešení. Rovnici (1.24) sice lze řešit jako úlohu kvadratického programování (bude vysvětleno v další kapitole), nicméně takovýto způsob řešení je spojen s vyššími výpočetními nároky.

V oblasti elektromotorů byl GPC algoritmus použit buďto pro řízení statorových proudů, kde GPC oproti FOC nahrazuje oba proudové PI regulátory nebo jako náhrada otáčkového regulátoru. Při regulaci proudů je pro nižší výpočetní nároky nutno použít analyticky řešeného GPC, neboť proudová regulace vyžaduje krátkou vzorkovací periodu. Napěťové omezení je tak nutno řešit mimo regulátor, čímž nelze zaručit optimalitu akčního zásahu. Ve výsledku tak GPC oproti klasické metodě regulace proudů s PI regulátory nepřináší příliš výhod. Co se týká otáčkové regulace, je situace lepší. Algoritmus totiž dosahuje rychlejší odezvy a hlavně menšího překmitu. GPC algoritmu jako jediného společného regulátor proudů i otáček se nepoužívá, neboť pro popis motoru jako MIMO systému se operátorový přenos (a tedy i CARIMA model) nehodí.

Algoritmus Linear Model Predictive Control

Dalším algoritmem, který bude v této práci popsán je lineární prediktivní řízení založené na modelu (lineární MPC). Ten je v některých ohledech podobný GPC

algoritmu a to natolik, že jsou tyto algoritmy v literatuře často zaměňovány. Za GPC algoritmus bude v této práci považován ten, jež vychází z operátorového přenosu typu CARIMA. Lineární MPC pak bude vycházet ze stavového popisu. [3]

Lineární MPC je díky svým vlastnostem velmi perspektivní typ řízení (nejen) elektropohonů. Jeho výhodou je především možnost jednoduché implementace pro MIMO systémy a zahrnutí omezení stavů, vstupů a výstupů systému. Navíc se v nedávné době objevila možnost takzvaného explicitního řízení, jež dovoluje výrazně snížit výpočetní nároky. Z těchto důvodů bude lineární MPC algoritmu věnována vlastní kapitola. [30], [5]

2. Lineární prediktivní řízení založené na modelu

V této kapitole bude představen algoritmus lineárního prediktivního řízení založeného na modelu, v překladu Linear Model Predictive Control (lineární MPC). Tento algoritmus je poměrně populární, neboť umožňuje řízení MIMO systémů, systematický způsob implementace omezení vstupů, stavů a výstupů systému při zachování optimality řízení. Nevýhodou jsou potom vyšší výpočetní nároky, možnost nasazení pouze pro lineární systémy a (stejně jako u všech MPC algoritmů) závislost na přesnosti modelu systému. V oblasti elektropohonů však získání poměrně přesného modelu nebývá problém. Díky nedávným pokrokům v teorii prediktivního řízení se navíc daří lineární MPC algoritmy nasazovat se stále kratší vzorkovací periodou. Ve spojitosti s rostoucím dostupným výpočetním výkonem hardwaru se lineární MPC pro řízení elektromotorů jeví jako velmi perspektivní. Implementace omezení je totiž při řízení elektropohonů velmi důležitým aspektem řízení, který může zásadně ovlivnit výslednou kvalitu řízení. [28], [4], [5], [9]

V této kapitole bude podrobně vysvětleny základy lineárního MPC algoritmu. Nakonec pak bude představen lineární MPC algoritmus pro řízení PMSM motoru (pro odlišení tuto implementaci budeme nazývat zkratkou LMPC).

2.1. Popis lineárního prediktivního řízení založeného na modelu

Pro odvození lineárního MPC algoritmu je potřeba tří základních komponent. Jsou to účelová funkce, model systému a rovnice omezení vstupů, stavů a výstupů systému. Ty budou postupně představeny v této podkapitole. Nakonec bude proveden převod těchto tří soustav rovnic na úlohu kvadratického programování.

2.1.1. Model systému

Algoritmus lineárního MPC lze teoreticky postavit na libovolném typu lineárního modelu. Nejvýhodnější se však jeví stavový popis. Ten, oproti modelu ve formě impulzové, skokové odezvy a operátorového přenosu, umožňuje snadné modelování MIMO systémů. Nevýhoda v podobě obtížného modelování dopravního zpoždění nemá u elektropohonů takový význam.

Stavový popis časově invariantního spojitého lineárního systému lze popsat rovnicemi

$$\frac{d\vec{x}(t)}{dt} = \mathbf{A}'\vec{x}(t) + \mathbf{B}'\vec{u}(t), \quad (2.1)$$

$$\vec{y}(t) = \mathbf{C}\vec{x}(t) + \mathbf{D}\vec{u}(t), \quad (2.2)$$

kde $\vec{x} \in \mathbb{R}^{n_x}$ je vektor stavů, $\vec{u} \in \mathbb{R}^{n_u}$ je vektor vstupů systému, $\vec{y} \in \mathbb{R}^{n_y}$ je vektor výstupů, $\mathbf{A}' \in \mathbb{R}^{n_x \times n_x}$ je matice zpětných vazeb, $\mathbf{B}' \in \mathbb{R}^{n_x \times n_u}$ je výstupní matice, $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ je vstupní matice, $\mathbf{D} \in \mathbb{R}^{n_y \times n_u}$ je matice přímých vazeb vstupu na výstup a t značí spojitý čas. Pro převod na diskretní verzi použijeme Eulerovu diskretizaci, kdy je použita aproximace

$$\frac{d\vec{x}(t)}{dt} \doteq \frac{\vec{x}(k+1) - \vec{x}(k)}{T_s}, \quad (2.3)$$

kde T_s je vzorkovací perioda a k značí diskretní čas (pořadí vzorku). Tím získáme rovnice diskretního stavového popisu lineárního systému

$$\vec{x}(k+1|k) = (\mathbf{I} + T_s\mathbf{A}')\vec{x}(k) + T_s\mathbf{B}'\vec{u}(k) = \mathbf{A}\vec{x}(k) + \mathbf{B}\vec{u}(k), \quad (2.4)$$

$$\vec{y}(k|k) = \mathbf{C}\vec{x}(k) + \mathbf{D}\vec{u}(k), \quad (2.5)$$

kde $\mathbf{I} \in \mathbb{R}^{n_x \times n_x}$ je jednotková matice, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ a $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ jsou diskretizované matice zpětných vazeb a vstupů.

Na základě diskretního modelu nyní odvodíme prediktor stavů a výstupů. Pro predikci stavu v následujícím kroku $k+2$ dle rovnice (2.4) platí

$$\vec{x}(k+2|k) = \mathbf{A}\vec{x}(k+1|k) + \mathbf{B}\vec{u}(k+1) = \mathbf{A}^2\vec{x}(k) + \mathbf{A}\mathbf{B}\vec{u}(k) + \mathbf{B}\vec{u}(k+1). \quad (2.6)$$

Pro predikci vektoru stavů v j -tém kroku pak můžeme psát

$$\vec{x}(k+j|k) = \underbrace{\mathbf{A}^j\vec{x}(k)}_{\text{volná odezva}} + \underbrace{\sum_{i=0}^{j-1} \mathbf{A}^i\mathbf{B}\vec{u}(k+j-1-i)}_{\text{nucená odezva}}, \quad (2.7)$$

což stejně jako u GPC algoritmu můžeme interpretovat jako superpozici volné a nucené odezvy. Pro vektor predikcí stavů $\vec{x}_N(k)$ v rámci predikčního horizontu N můžeme psát

$$\underbrace{\begin{bmatrix} \vec{x}(k+1|k) \\ \vec{x}(k+2|k) \\ \vdots \\ \vec{x}(k+N|k) \end{bmatrix}}_{\vec{x}_N} = \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^N \end{bmatrix}}_{\mathbf{\Gamma}} \vec{x}(k) + \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix}}_{\mathbf{\Omega}} \underbrace{\begin{bmatrix} \vec{u}(k) \\ \vec{u}(k+1) \\ \vdots \\ \vec{u}(k+N-1) \end{bmatrix}}_{\vec{u}_{N-1}}, \quad (2.8)$$

kde $\mathbf{\Gamma} \in \mathbb{R}^{N \cdot n_x \times n_x}$ a $\mathbf{\Omega} \in \mathbb{R}^{N \cdot n_x \times N \cdot n_u}$ jsou stavové predikční matice.

Stejný postup můžeme aplikovat i pro predikci výstupu. Pro něj v rámci predikčního horizontu N dostáváme

$$\underbrace{\begin{bmatrix} \vec{y}(k|k) \\ \vec{y}(k+1|k) \\ \vdots \\ \vec{y}(k+N-1|k) \end{bmatrix}}_{\vec{y}_{N-1}} = \underbrace{\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{N-1} \end{bmatrix}}_{\mathbf{\Gamma}_y} \vec{x}(k) + \underbrace{\begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CB} & \mathbf{D} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N-2}\mathbf{B} & \mathbf{CA}^{N-3}\mathbf{B} & \cdots & \mathbf{D} \end{bmatrix}}_{\mathbf{\Omega}_y} \vec{u}_{N-1}, \quad (2.9)$$

kde $\mathbf{\Gamma}_y \in \mathbb{R}^{N \cdot n_y \times n_x}$ a $\mathbf{\Omega}_y \in \mathbb{R}^{N \cdot n_y \times N \cdot n_u}$ jsou výstupní predikční matice. [30], [5], [12]

2.1.2. Účelová funkce

Cílem všech MPC algoritmů je minimalizace skalární hodnoty účelové funkce. Právě skrze ni lze definovat cíle regulace a tedy i chování regulační smyčky. To se děje skrze nastavení jednotlivých koeficientů, kterými jsou násobeny veličiny, jež vystupují v účelové funkci. Právě velikostí těchto koeficientů jsme schopni určit relativní důležitost cílů regulace. [12], [30]

Účelová funkce může nabývat mnoha podob. Obecně však vždy musí být kladá a konvexní. Uvažujme například

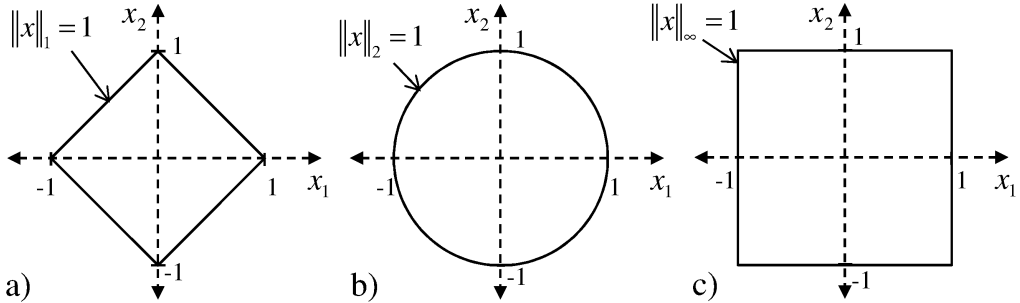
$$J_N = \sum_{i=0}^N \|\mathbf{Q}\vec{e}(k+i)\|_m + \sum_{i=0}^{N_u-1} \|\mathbf{R}\vec{u}(k+i)\|_m, \quad (2.10)$$

kde N je délka predikčního horizontu, N_u je délka řídicího horizontu, $\mathbf{Q} \in \mathbb{R}^{n_e \times n_e}$ je penalizační matice řídicí odchylky $\vec{e}(k) \in \mathbb{R}^{n_e}$, $\mathbf{R} \in \mathbb{R}^{n_u \times n_u}$ je penalizační matice akčního zásahu $\vec{u}(k) \in \mathbb{R}^{n_u}$ a m označuje typ použité $\mathcal{L}_m$ -normy. Zatímco první člen účelové funkce penalizuje velikost řídicí odchylky, což je jeden ze základních

požadavků na řízení, druhý penalizuje velikost akčního zásahu. Musí přitom platit $\mathbf{Q} \succeq 0$ a $\mathbf{R} \succ 0$. $\mathcal{L}_m$ -norma je pro obecnou veličinu v definována jako

$$\|v\|_m = \sqrt[m]{\sum_{\forall i} |v_i|^m}, \quad (2.11)$$

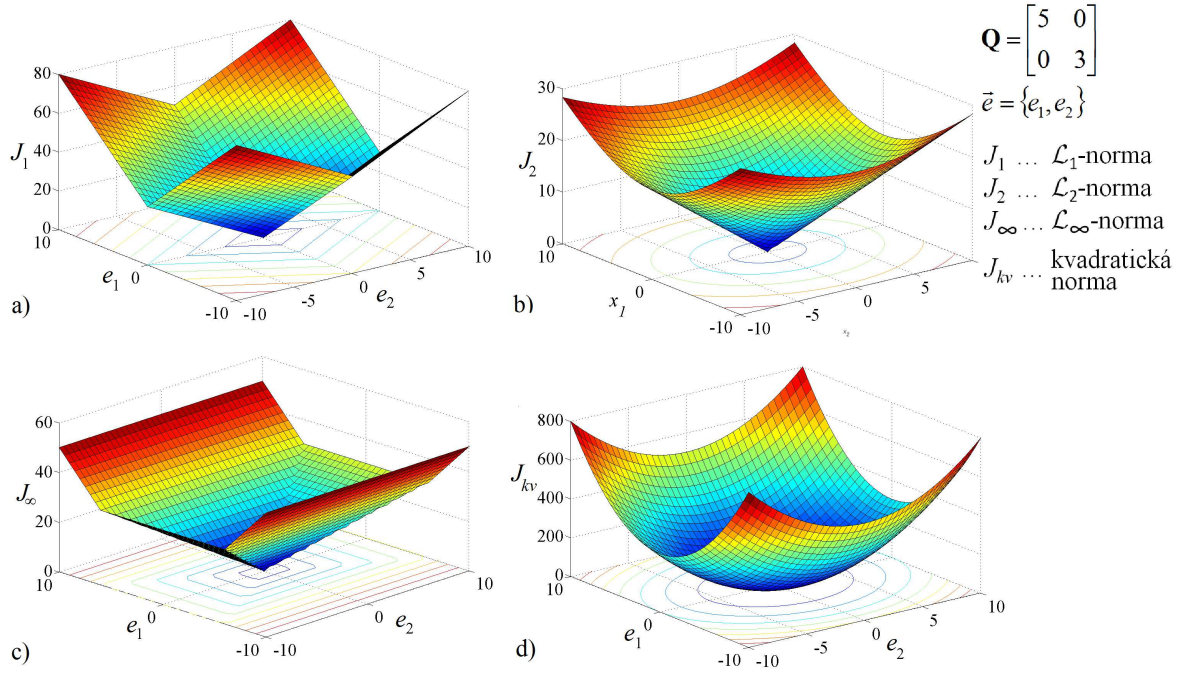
kde $m \in \mathbb{R}^+$. U lineárního MPC regulátoru se používá $\mathcal{L}_1$ -norma, $\mathcal{L}_\infty$ -norma, $\mathcal{L}_2$ -norma a kvadratická norma, která je velmi blízká $\mathcal{L}_2$ -normě, avšak ve vztahu (2.11) není použito odmocniny. První dvě ze jmenovaných norem vedou na úlohu optimalizace zvanou lineární programování, zatímco kvadratická norma vede na úlohu kvadratického programování (bude vysvětleno dále). Na obrázku 2.1 jsou obrazce, jež tvoří vektory jednotkové délky v $\mathbb{R}^2$. Důležité je povšimnout si, že pouze $\mathcal{L}_2$ -norma a kvadratická norma má hladký tvar. Ještě lépe je situace patrná z obrázku 2.2, kde je zobrazen příklad účelové funkce při $\vec{u}(k) = \vec{0}$ na celém predikčním horizontu. Pokud bychom totiž začali uvažovat omezení veličin, pak optimální bod by se pro $\mathcal{L}_1$ -normu a $\mathcal{L}_\infty$ -normu nacházel vždy v rohu výsledného geometrického útvaru. Lineární programování je tím sice (co do výpočtu) méně náročné, nicméně se výsledný regulátor stává oproti kvadratickému programování zbytečně agresivním a citlivým k nepřesnosti modelu a šumu. Svými vlastnostmi se pak lineární programování podobá deadbeat regulaci. [26]



Obrázek 2.1: Obrazce tvořené vektory jednotkové délky dle a) $\mathcal{L}_1$ -normy, b) kvadratické a $\mathcal{L}_2$ -normy a c) $\mathcal{L}_\infty$ -normy v $\mathbb{R}^2$

2.1.3. Omezení

Jak již bylo zmíněno v úvodu kapitoly, právě systematický způsob implementace omezení je nejdůležitější výhodou lineárního MPC algoritmu. Jedná se o velmi cennou vlastnost, neboť v praxi jsou akční členy mnohdy provozovány na hranicích



Obrázek 2.2: Příklady účelových funkcí založených na a) $\mathcal{L}_1$ -normě, b) $\mathcal{L}_2$ -normě, c) $\mathcal{L}_\infty$ -normě a kvadratické normě

možností. Regulátor se tedy nejčastěji musí umět vypořádat s nelinearitou typu nasycení vstupů $\vec{u}(k)$, stavů $\vec{x}(k)$ a výstupů systému $\vec{y}(k)$, tedy

$$\vec{u}(k) \in \langle \vec{u}_{\min}(k), \vec{u}_{\max}(k) \rangle, \quad (2.12)$$

$$\vec{x}(k) \in \langle \vec{x}_{\min}(k), \vec{x}_{\max}(k) \rangle, \quad (2.13)$$

$$\vec{y}(k) \in \langle \vec{y}_{\min}(k), \vec{y}_{\max}(k) \rangle, \quad (2.14)$$

kde $\vec{u}_{\max}(k)$, $\vec{x}_{\max}(k)$ a $\vec{y}_{\max}(k)$ značí horní hranici a $\vec{u}_{\min}(k)$, $\vec{x}_{\min}(k)$ a $\vec{y}_{\min}(k)$ zase dolní hranici. Často je pak výhodné zavést omezení na změnu akčního zásahu, tedy

$$\delta \vec{u}(k) \in \langle \vec{\delta u}_{\min}(k), \vec{\delta u}_{\max}(k) \rangle, \quad (2.15)$$

kde $\vec{\delta u}_{\max}(k)$ je maximální a $\vec{\delta u}_{\min}(k)$ je minimální změna akčního zásahu. [30], [12]

Omezení u lineárního MPC můžeme rozdělit do dvou kategorií

- Pevná omezení - v překladu hard constraints. Tato omezení nelze ze svého principu překročit. Zde mohou patřit omezení výstupu akčního členu či různá fyzikální omezení.
- Měkká omezení - v angličtině soft constraints. Tato omezení lze mírně překročit, nicméně je tato situace penalizována. Jedná se o různá uměle defi-

novaná omezení (například statorový proud motoru může na chvíli překročit stanovenou mez, aniž by to odporovalo fyzikálním podmínkám)

Pevná omezení se obvykle používají u vstupů systému. Naopak u stavů a výstupů systému je doporučeno používat měkká omezení. Narozdíl od vstupů systému, které definuje regulátor jsou stavy a výstupy systému předmětem měření a jsou tedy zatíženy šumem. Snadno by tak mohlo dojít k překročení pevného omezení, což by vedlo k neřešitelnosti (v překladu *infeasibility*) úlohy optimalizace.

Princip měkkých omezení můžeme ukázat na příkladu omezení maximální hodnoty výstupu systému, tedy

$$y(k) \leq y_{\max}(k) + \varepsilon(k), \quad (2.16)$$

kde $\varepsilon(k)$ je uvolňující proměnná (v překladu slack variable). Proměnná $\varepsilon(k)$ je následně penalizována v účelové funkci a to dostatečně velkým koeficientem. Samozřejmě musí platit, že pokud není omezení překročeno, tedy $y(k) \leq y_{\max}(k)$, tak jsou hodnoty $\varepsilon(k)$ nulové. Dojde tedy k rozšíření účelové funkce (2.10) o nový člen, čímž dostáváme

$$J'_N = J_N + \sum_{i=0}^N \|\mathbf{Q}_\varepsilon \tilde{\varepsilon}(k+i)\|_m, \quad (2.17)$$

kde m označuje použitou normu a $\mathbf{Q}_\varepsilon \succeq 0$ penalizační matici vektoru uvolňujících proměnných $\tilde{\varepsilon}(k)$. Ekvivalentní formulací je zavedení pevného omezení $\varepsilon(k) \leq y_{\max}(k)$ a penalizace rozdílu $y(k) - \varepsilon(k)$ uvnitř účelové funkce.

Z rovnice (2.17) je však patrné, že zavedení měkkých omezení zvyšuje počet stupňů volnosti úlohy optimalizace a tím i výpočetní nároky. Je sice možné zavádět uvolňující proměnné společně i pro více proměnných a tím výpočetní nároky mírnit, nicméně u tak časově náročných úloh, jako je regulace elektropohonů, se používají výhradně pevná omezení. Problém s případnou neřešitelností úlohy v případě překročení pevného omezení stavem či výstupem lze řešit odstraněním omezení na první krok predikce.

U lineárního MPC je možné definovat omezení obecněji, než je tomu u množin (2.12) až (2.15). Je pouze nutné, aby byly afinní funkcí vstupů, stavů a výstupů. Jelikož však lineární MPC povede na úlohu kvadratického programování je důležité, aby byla výsledná množina vzniklá definovanými omezeními konvexní. Platí přitom, že množina $S \subset \mathbb{R}^n$ je konvexní, pokud všechny body na úsečce spojující jakékoliv dva body množiny zároveň této množině příslušely.

2.2. Lineární MPC regulátor

V této podkapitole bude ukázán převod rovnic modelu systému, účelové funkce a omezení definovaných v podkapitolách 2.1.1 až 2.1.3 na úlohu kvadratického programování. Jejím řešením je pak získána optimalizující sekvence akčního zásahu. Nejprve uvažujme účelovou funkci ve tvaru kvadratické normy, tedy

$$J_N(k) = \frac{1}{2} \vec{x}(k+N)^T \mathbf{P} \vec{x}(k+N) + \frac{1}{2} \sum_{i=1}^{N-1} \vec{x}(k+i)^T \mathbf{Q} \vec{x}(k+i) + \frac{1}{2} \sum_{i=1}^{N_u} \vec{u}(k+i-1)^T \mathbf{R} \vec{u}(k+i-1), \quad (2.18)$$

kde $\vec{x}(k) \in \mathbb{R}^{n_x}$ je vektor stavů systému, $\vec{u}(k) \in \mathbb{R}^{n_u}$ je vektor vstupů systému, matice $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x}$ penalizuje vektor stavů, matice $\mathbf{R} \in \mathbb{R}^{n_u \times n_u}$ penalizuje akční zásah, $\mathbf{P} \in \mathbb{R}^{n_x \times n_x}$ penalizuje konečný stav, N je predikční horizont a N_u je řídicí horizont. Matice $\mathbf{Q} \succeq 0$, $\mathbf{P} \succeq 0$ a $\mathbf{R} \succ 0$ přitom musí být symetrické. V účelové funkci (2.18) sice nevystupuje odchylka, nicméně jak bude později ukázáno, sledování reference lze definovat i v rámci této definice. [5], [30], [12]

Pro převod na úlohu kvadratického programování nahradíme budoucí hodnoty stavů v této účelové funkci prediktorem definovaném v rovnici (2.8). Ještě před tím však upravíme rovnici prediktoru dle techniky blokování (v překladu *move blocking*). Tato metoda umožňuje podstatně snížit výpočetní nároky lineárního MPC algoritmu zavedením řídicího horizontu $N_u \leq N$. Akční zásah pak bude v čase $k+N_u$ až $k+N$ konstantní, což vede na snížení stupně volnosti při výpočtu optimalizující sekvence akčního zásahu. Můžeme tedy psát

$$\underbrace{\begin{bmatrix} \vec{u}(k) \\ \vec{u}(k+1) \\ \vdots \\ \vec{u}(k+N_u-1) \\ \vdots \\ \vec{u}(k+N-1) \end{bmatrix}}_{\vec{u}_{N-1}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \vec{u}(k) \\ \vec{u}(k+1) \\ \vdots \\ \vec{u}(k+N_u-1) \end{bmatrix}}_{\vec{u}_{N_u-1}}, \quad (2.19)$$

kde $\vec{u}_{N-1}$ a $\vec{u}_{N_u-1}$ jsou vektory budoucích vstupů, $\mathbf{I} \in \mathbb{R}^{n_u \times n_u}$ je jednotková matice a $\mathbf{M} \in \mathbb{R}^{N \cdot n_u \times N_u \cdot n_u}$ je takzvaná blokuující matice. Tím dostáváme stavový prediktor

$$\vec{x}_N = \mathbf{\Gamma} \vec{x}(k) + \mathbf{\Omega} \mathbf{M} \vec{u}_{N_u-1} = \mathbf{\Gamma} \vec{x}(k) + \mathbf{\Omega}_m \vec{u}_{N_u-1}, \quad (2.20)$$

kde $\mathbf{\Omega}_m \in \mathbb{R}^{N \cdot n_x \times N_u \cdot n_u}$ je matice, pro niž platí $\mathbf{\Omega}_m = \mathbf{\Omega} \mathbf{M}$.

Jako další zavedme diagonální matice

$$\mathbf{Q}^* = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P} \end{bmatrix}, \quad \mathbf{R}^* = \begin{bmatrix} \mathbf{R} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R} \end{bmatrix}, \quad (2.21)$$

kdy musí platit $\mathbf{Q}^* \succeq 0$ a $\mathbf{R}^* \succ 0$. Dosazením těchto matic a preditkoru (2.20) do účelové funkce (2.18) můžeme psát

$$J_N = \frac{1}{2} (\mathbf{\Gamma} \vec{x}(k) + \mathbf{\Omega}_m \vec{u}_{N_u-1})^T \mathbf{Q}^* (\mathbf{\Gamma} \vec{x}(k) + \mathbf{\Omega}_m \vec{u}_{N_u-1}) + \frac{1}{2} \vec{u}_{N_u-1}^T \mathbf{R}^* \vec{u}_{N_u-1}. \quad (2.22)$$

Úpravou této rovnice pak dostáváme

$$J_N = \frac{1}{2} \vec{u}_{N_u-1}^T \underbrace{(\mathbf{\Omega}_m^T \mathbf{Q}^* \mathbf{\Omega}_m + \mathbf{R}^*)}_{\mathbf{H}} \vec{u}_{N_u-1} + \underbrace{(\vec{x}(k)^T \mathbf{\Gamma}^T \mathbf{Q}^* \mathbf{\Omega}_m)}_{\vec{f}^T} \vec{u}_{N_u-1} + \underbrace{\vec{x}(k)^T (\mathbf{\Gamma}^T \mathbf{Q}^* \mathbf{\Gamma})}_{c} \vec{x}(k), \quad (2.23)$$

kde $\mathbf{H} \in \mathbb{R}^{N_u \cdot n_u \times N_u \cdot n_u}$, $\vec{f} \in \mathbb{R}^{N_u \cdot n_u}$ a $c \in \mathbb{R}$. Aby mohla rovnice (2.23) vystupovat jako část zadání kvadratického programování, musí být matice $\mathbf{H}$ pozitivně definitní, což je vzhledem k $\mathbf{Q}^* \succeq 0$ a $\mathbf{R}^* \succ 0$ vždy splněno.

Nyní již zbývá pouze vhodným způsobem definovat omezení. Jak již bylo řečeno v kapitole 2.1.3, je možné je definovat jako afinní funkce stavů, vstupů a výstupů systému. Zároveň musí být funkcí optimalizační sekvence $\vec{u}_{N_u-1}$. Pevná omezení (2.12) až (2.15) tak můžeme přepsat ve formě

$$\mathbf{L} \vec{u}_{N_u-1} \leq \mathbf{W}, \quad (2.24)$$

kde

$$\mathbf{L} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{E} \\ -\mathbf{E} \\ \mathbf{\Omega}_m \\ -\mathbf{\Omega}_m \\ \mathbf{\Omega}_y \mathbf{M} \\ -\mathbf{\Omega}_y \mathbf{M} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \vec{u}_{\max} \\ \vec{u}_{\min} \\ \vec{\delta u}_{\max} \\ \vec{\delta u}_{\min} \\ \vec{x}_{\max} + \mathbf{\Gamma} \vec{x}(k) \\ \vec{x}_{\min} - \mathbf{\Gamma} \vec{x}(k) \\ \vec{y}_{\max} + \mathbf{\Gamma}_y \vec{x}(k) \\ \vec{y}_{\min} - \mathbf{\Gamma}_y \vec{x}(k) \end{bmatrix}. \quad (2.25)$$

V rovnici (2.25) je $\mathbf{I} \in \mathbb{R}^{N_u \cdot n_u \times N_u \cdot n_u}$ jednotkovou maticí a matice $\mathbf{E} \in \mathbb{R}^{N_u \cdot n_u \times N_u \cdot n_u}$ je definována jako

$$\begin{bmatrix} \mathbf{I}_{N_u} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -\mathbf{I}_{N_u} & \mathbf{I}_{N_u} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I}_{N_u} & \mathbf{I}_{N_u} \end{bmatrix}, \quad (2.26)$$

kde $\mathbf{I} \in \mathbb{R}^{n_u \times n_u}$ je jednotková matice a matice $\mathbf{\Omega}_m, \mathbf{\Gamma}_m, \mathbf{\Omega}_y, \mathbf{\Gamma}_y$ i $\mathbf{M}$ byly definovány v rovnicích (2.8), (2.9) a (2.19).

Co se týká vektorů, z nichž je složena matice $\mathbf{W}$ v rovnici (2.25), pak platí

$$\vec{u}_{\max} = [\vec{u}_{\max}(k), \dots, \vec{u}_{\max}(k + N_u n_u)]^T, \quad (2.27)$$

$$\vec{u}_{\min} = [-\vec{u}_{\min}(k), \dots, -\vec{u}_{\min}(k + N_u n_u)]^T, \quad (2.28)$$

$$\vec{\delta u}_{\max} = [\vec{u}(k-1) + \vec{\delta u}_{\max}(k), \dots, \vec{\delta u}_{\max}(k + N_u n_u)]^T, \quad (2.29)$$

$$\vec{\delta u}_{\min} = [-\vec{u}(k-1) - \vec{\delta u}_{\min}(k), \dots, \vec{\delta u}_{\min}(k + N_u n_u)]^T, \quad (2.30)$$

$$\vec{x}_{\max} = [\vec{x}_{\max}(k), \dots, \vec{x}_{\max}(k + N n_x)]^T, \quad (2.31)$$

$$\vec{x}_{\min} = [-\vec{x}_{\min}(k), \dots, -\vec{x}_{\min}(k + N n_x)]^T, \quad (2.32)$$

$$\vec{y}_{\max} = [\vec{y}_{\max}(k), \dots, \vec{y}_{\max}(k + N n_y)]^T, \quad (2.33)$$

$$\vec{y}_{\min} = [-\vec{y}_{\min}(k), \dots, -\vec{y}_{\min}(k + N n_y)]^T, \quad (2.34)$$

kde $\vec{u}_{\max}(k), \vec{x}_{\max}(k), \vec{y}_{\max}(k), \vec{\delta u}_{\max}(k), \vec{u}_{\min}(k), \vec{x}_{\min}(k), \vec{y}_{\min}(k)$ a $\vec{\delta u}_{\min}(k)$ byly definovány jako hranice množin (2.12) až (2.15).

Tím se dostáváme k definici naší optimalizační úlohy kvadratického programování

$$\underset{\vec{u}_{N_u-1}}{\text{minimalizuj:}} \quad \frac{1}{2} \vec{u}_{N_u-1}^T \mathbf{H} \vec{u}_{N_u-1} + \vec{f}^T \vec{u}_{N_u-1}, \quad (2.35)$$

$$\text{při dodržení:} \quad \mathbf{L} \vec{u}_{N_u-1} \leq \mathbf{W}.$$

Můžeme si povšimnout, že skalár c z rovnice (2.23) zcela chybí. Na výsledek nemá vliv, protože není funkcí optimalizující sekvence $\vec{u}_{N_u-1}$.

Jelikož je nutné řešit optimalizační úlohu (2.35) v každém kroku regulace, je rychlost jejího výpočtu kritická. Dnešní moderní algoritmy pro řešení kvadratického programování (nazvěme je *solvery*) jsou obvykle založeny na metodě aktivní množiny (v překladu *active set*) nebo metodě vnitřního bodu (v angličtině *interior point*).

Metoda aktivní množiny využívá toho, že se optimum obvykle nachází na některé z hranic řešitelné množiny (takzvané *feasible set*). To znamená, že některá z ome-

zení jsou v optimu aktivní. Tato omezení tvoří pak aktivní množinu (*active set*). V každém kroku je do pracovní množiny aktivních omezení, se kterými je v tu chvíli zacházeno jako s rovnostními omezeními, přidáno či odebráno omezení. Následně je pro tuto pracovní množinu vypočtena úloha kvadratického programování s rovnostními omezeními. Výpočet této úlohy sice není triviální, nicméně počet aktivních omezení je obvykle nízký a tím i počet iterací tohoto algoritmu. Pokud se nacházíme v globálním minimu, pak algoritmus končí, jinak se iterativní proces opakuje a je přidáno či odebráno aktivní omezení. Nevýhodou algoritmu aktivních množin je, že v jeden okamžik lze přidat či odebrat pouze jedno aktivní omezení, což v určitých případech může vést k velkému počtu iterací.

Metoda vnitřního bodu, jinak známá jako metoda bariérových funkcí, počítá s modifikovanou úlohou (2.35) bez uvážení omezení. Ta jsou do výpočtu zahrnuta skrze takzvanou bariérovou funkci, která se přidá k zadání úlohy kvadratického programování. Bariérová funkce má obvykle tvar logaritmu, jehož velikost prudce narůstá v blízkosti omezení. Je tedy nemožné, aby se optimem stal bod mimo řešitelnou množinu (*feasible set*), odtud název metoda vnitřních bodů. Z toho rovněž plyne nutnost, aby hledání optima započalo uvnitř řešitelné množiny. Řešením úlohy kvadratického programování bez omezení pak lze získat pouze aproximaci původního řešení. To je způsobeno přítomností nelineárních bariérových funkcí. Ačkoli je každá iterace poměrně výpočetně náročná, jejich počet je obvykle nízký. [30]

2.2.1. Explicitní lineární MPC regulátor

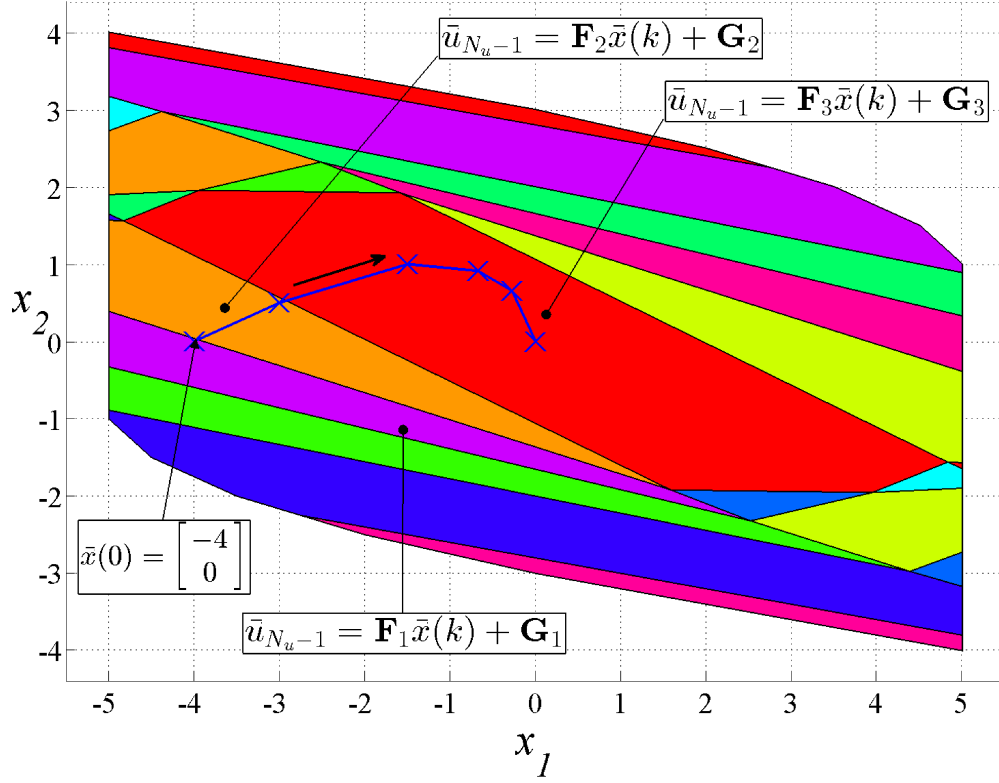
Lineární MPC řešené výše popsaným způsobem (takzvané implicitní lineární MPC) je pro dnešní běžně používané signálové procesory výpočetně příliš náročné. Tento fakt omezuje jeho nasazení na pomalé systémy, kde lze použít dlouhou vzorkovací periodu. Možné řešení přináší takzvané explicitní řízení, které umožňuje předpočítat část řídicí úlohy offline, ještě před samotnou implementací do procesu. [4], [5]

Lze ukázat, že lineární MPC regulátor lze vyjádřit množinou řídicích zákonů v podobě afinních funkcí stavů, respektive stavových zpětných vazeb. Každý z těchto řídicích zákonů pak přísluší některému z konečného počtu regionů uvnitř stavového prostoru, ve tvaru mnohostěnu. Pokud se aktuální stav nachází v regionu i , pak je systému zavedena zpětná vazba

$$\vec{u}_{N_u-1} = \mathbf{F}_i \vec{x}(k) + \mathbf{G}_i, \quad (2.36)$$

kde $\mathbf{F}_i$ a $\mathbf{G}_i$ jsou matice definující řídicí zákon. Právě v rozdělení stavového prostoru do regionů a nalezení příslušných řídicích zákonu spočívá offline část řešení

algoritmu. Při online výpočtu algoritmu pak pouze zbývá identifikovat aktivní region, v němž se nachází aktuální stav systému a následně vypočítat rovnici (2.36). Aktivní region lze nalézt s pomocí binárního stromu, kdy pro počet regionů N_r je potřeba pouze $\log(N_r)$ větvení. Bohužel se složitostí zadané úlohy stoupá i počet regionů, což omezuje možnosti nasazení explicitního řízení pro komplexní úlohy.



Obrázek 2.3: Rozdělení stavového prostoru s pomocí explicitního lineárního MPC

Na obrázku 2.3 je uveden příklad rozdělení stavového prostoru do regionů pro systém

$$\vec{x}(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \vec{x}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad (2.37)$$

kde byl regulátor nastaven

$$\mathbf{Q} = \mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = 1, \quad N = 5, \quad (2.38)$$

s omezeními

$$x_1 \in \langle -5, 5 \rangle, \quad x_2 \in \langle -5, 5 \rangle \quad u \in \langle -1, 1 \rangle. \quad (2.39)$$

Na obrázku je rovněž zanesena trajektorie stavu systému v prvních pěti krocích regulace pro případ $\vec{x}(0) = [-4, 0]^T$. Například v prvním kroku pak byla vygenerována optimalizační sekvence

$$\vec{u}_{N_u-1} = \mathbf{F}_1 \vec{x}(0) + \mathbf{G}_1 = \begin{bmatrix} 0,00 & 0,00 \\ 0,00 & 0,00 \\ -0,51 & -1,92 \\ -0,05 & -0,51 \\ 0,00 & 0,00 \end{bmatrix} \begin{bmatrix} -4 \\ 0 \end{bmatrix} + \begin{bmatrix} 1,00 \\ 1,00 \\ -2,17 \\ -0,53 \\ 0,00 \end{bmatrix} = \begin{bmatrix} 1,00 \\ 1,00 \\ -0,14 \\ -0,34 \\ 0,00 \end{bmatrix}, \quad (2.40)$$

kde podle principů RHPC bude použita pouze první hodnota z této sekvence (pro reálnou aplikaci tedy stačí použít první řádky matic $\mathbf{F}_1$ a $\mathbf{G}_1$). Je vidět, že po nalezení aktivního regionu je výpočet akčního zásahu (ve srovnání s úlohou (2.35)), poměrně triviální. [14]

2.2.2. Sledování reference s lineárním MPC

Tvar, ve kterém je uvedena účelová funkce (2.18), penalizuje pouze hodnoty stavů a akčního zásahu. Regulátor tak pouze uvede všechny penalizované stavy systému do počátku stavového prostoru. Aby bylo možno sledovat referenci, je nutno rozšířit vektor stavů o nové proměnné, tedy

$$\underbrace{\begin{bmatrix} \vec{x}(k+1|k) \\ \vec{y}(k|k) \\ \vec{w}(k+1|k) \end{bmatrix}}_{\vec{x}'(k+1|k)} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_y \end{bmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix} \vec{x}(k) \\ \vec{y}(k-1) \\ \vec{w}(k) \end{bmatrix}}_{\vec{x}'(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{D} \\ \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{B}}} \vec{u}(k), \quad (2.41)$$

kde $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ a $\mathbf{D}$ jsou matice stavového popisu dle (2.4) a (2.5), $\mathbf{I}_y \in \mathbb{R}^{n_y \times n_y}$ je jednotková matice, $\tilde{\mathbf{A}}$ a $\tilde{\mathbf{B}}$ jsou rozšířené matice stavového popisu a $\vec{w}(k) \in \mathbb{R}^{n_y}$ je vektor žádaných hodnot. Vektor výstupů systému $\vec{y}(k)$ je samozřejmě možno měřit, v tom případě se odpovídajícím způsobem změní matice $\tilde{\mathbf{A}}$. Rovněž si můžeme povšimnout, že vektor žádaných hodnot je považován za konstantu, čímž se předpokládá, že budoucí hodnoty reference nejsou známy, nebo se na predikčním horizontu příliš nemění. Pokud budoucí hodnoty reference známy jsou a je výhodné je do výpočtu zahrnout, pak je nutno odpovídajícím způsobem namodelovat jejich dynamiku. Nicméně stavový popis se k tomuto příliš nehodí (narozdíl od přenosového modelu u GPC). Krom toho se u elektropohonů používá velmi krátký predikční horizont, na kterém se žádaná hodnota příliš nezmění. Nyní, když je součástí nového

stavového vektoru $\vec{x}'(k)$ i vektor žádaných hodnot a výstupu systému, je možné penalizovat rozdíl mezi těmito stavy uvnitř účelové funkce (2.18). Matice penalizace stavů tak získá tvar

$$\mathbf{Q} = \begin{bmatrix} \gamma_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \gamma_e & -\gamma_e \\ \mathbf{0} & -\gamma_e & \gamma_e \end{bmatrix}, \quad (2.42)$$

kde $\gamma_x \in \mathbb{R}^{n_x \times n_x}$, $\gamma_x \succeq 0$ je matice koeficientů penalizující stavy systému a $\gamma_e \in \mathbb{R}^{n_y \times n_y}$, $\gamma_e \succeq 0$ je matice penalizující řídicí odchylku $\vec{e}(k+i) = \vec{w}(k+i) - \vec{y}(k+i)$. Ekvivalentní způsob je do modifikovaného vektoru stavů $\vec{x}'(k)$ zahrnout místo vektoru výstupu $\vec{y}(k)$ vektor odchylek $\vec{e}(k)$, tedy $\vec{x}'(k) = \begin{bmatrix} \vec{x}(k)^T, \vec{e}(k-1)^T, \vec{w}(k)^T \end{bmatrix}^T$ čímž získá rozšířená matice zpětných vazeb tvar

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & -\mathbf{I}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_y \end{bmatrix}. \quad (2.43)$$

Řídicí odchylku je pak možno v účelové funkci (2.18) penalizovat přímo.

2.2.3. Nulová ustálená řídicí odchylka s lineárním MPC

Ze způsobu, jakým byla v kapitole 2.2.2 zavedena schopnost regulátoru sledovat žádanou hodnotu je patrné, že k nulové řídicí odchylce bude řízený systém konvergovat v nekonečno. To je řešeno s pomocí inkrementální formulace lineárního MPC. Ta spočívá v úpravě rovnice (2.41) do tvaru

$$\underbrace{\begin{bmatrix} \vec{x}(k+1|k) \\ \vec{y}(k|k) \\ \vec{w}(k+1|k) \\ \vec{u}(k|k) \end{bmatrix}}_{\vec{x}'(k+1|k)} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} & \mathbf{0} & \mathbf{D} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_u \end{bmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix} \vec{x}(k) \\ \vec{y}(k-1) \\ \vec{w}(k) \\ \vec{u}(k-1) \end{bmatrix}}_{\vec{x}'(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{D} \\ \mathbf{0} \\ \mathbf{I}_u \end{bmatrix}}_{\tilde{\mathbf{B}}} \vec{\delta u}(k), \quad (2.44)$$

kde $\mathbf{I}_u \in \mathbb{R}^{n_u \times n_u}$ je jednotková matice a $\vec{\delta u}(k)$ je změna (diference) akčního zásahu. Rozšířením vektoru stavů o minulou hodnotu akčního zásahu byl do regulátoru v podstatě vřazen integrátor. Dokud totiž nebude dosaženo nulové ustálené řídicí odchylky, bude mít pomyslný vstup do systému $\vec{\delta u}(k)$ nenulovou hodnotu.

Výsledný akční zásah, jež bude aplikován na vstup systému, lze potom získat sumací předchozí hodnoty akčního zásahu a difference na výstupu regulátoru, tedy

$$\vec{u}(k) = \vec{u}(k-1) + \vec{\delta u}(k). \quad (2.45)$$

2.2.4. Stabilita řízení s lineárním MPC

Stabilita regulace je základním požadavkem na řízení. Díky konečné délce predikčního horizontu lineárního MPC algoritmu není stabilita, která je definována na nekonečně dlouhém horizontu, zaručena. Narozdíl od klasických metod řízení, přitom nelze stabilitu lineárního MPC snadno vyšetřit. Problémem je přítomnost omezení, které z lineárního MPC činí nelineární regulátor. A co více, jak bylo ukázáno v podkapitole 2.2.1, řídicí zákon se dá vyjádřit pomocí afinních rovnic (2.36) příslušných danému regionu stavového prostoru. Těchto regionů přitom může být velký počet. Aby byl potom systém stabilní, musí ke stabilitě vést všechny kombinace regionů (a tím i řídicích zákonů) ve stavovém prostoru. Počet těchto kombinací je vysoký i pro jednoduché problémy. [30]

V praxi však problémy se stabilitou lineárního MPC algoritmu nebývají a obvykle není potřeba žádných zvláštních úprav algoritmu. Přesto v případě potřeby je možno pro její zajištění použít kombinace dvou prostředků:

- Penalizace konečného stavu (*terminal cost*) v účelové funkci maticí $\mathbf{P}$.
- Omezení na konci predikčního horizontu (*terminal constraint set*). Těmito omezeními je definován region uvnitř stavového prostoru, pro nějž platí, že pokud do něj vstoupí stav systému zůstane již uvnitř.

Princip těchto prostředků naznačíme s pomocí Ljapunovovy funkce. Jako její kandidát se výborně hodí účelová funkce (2.18). Aby však Ljapunovovou funkcí skutečně byla, je potřeba dokázat, že se její hodnota s časem nezvětšuje, tedy musí platit $J_N(k+1) \leq J_N(k)$. Pokud v rovnici (2.18) předpokládáme $N_u = N$, pak pro diferenci $W_N(k) = J_N(k+1) - J_N(k)$ můžeme psát

$$W_N(k) \leq -\vec{x}(k+1)^T \mathbf{Q} \vec{x}(k+1) - \vec{u}(k)^T \mathbf{R} \vec{u}(k) - \vec{x}(k+N)^T \mathbf{P} \vec{x}(k+N) + \vec{u}(k+N)^T \mathbf{R} \vec{u}(k+N) + \vec{x}(k+N+1)^T \mathbf{P} \vec{x}(k+N+1). \quad (2.46)$$

Lze prohlásit, že účelová funkce (2.18) je Ljapunovovou funkcí, pokud je splněna podmínka

$$\vec{x}(k+N)^T \mathbf{P} \vec{x}(k+N) \geq \vec{u}(k+N)^T \mathbf{R} \vec{u}(k+N) + \vec{x}(k+N+1)^T \mathbf{P} \vec{x}(k+N+1). \quad (2.47)$$

Tuto podmínku lze splnit dvěma způsoby:

- Zavedením řídicí sekvence $\vec{u}(k+i) = \vec{0}$ pro $i \geq N$. Podmínka (2.47) potom vede na Ljapunovovu funkci

$$\mathbf{A}^T \mathbf{P} \mathbf{A} \geq \mathbf{P} \rightarrow W_N = \mathbf{A}^T \mathbf{P} \mathbf{A} - \mathbf{P} \leq \mathbf{0}. \quad (2.48)$$

Pokud je systém stabilní, pak pro zajištění stability stačí vhodně penalizovat stav na konci predikčního horizontu.

- Zavedením řídicí sekvence $\vec{u}(k+i) = -\mathbf{K}\vec{x}(k+i)$ pro $i \geq N$. Podmínka (2.47) pak vede na Riccatiho rovnici

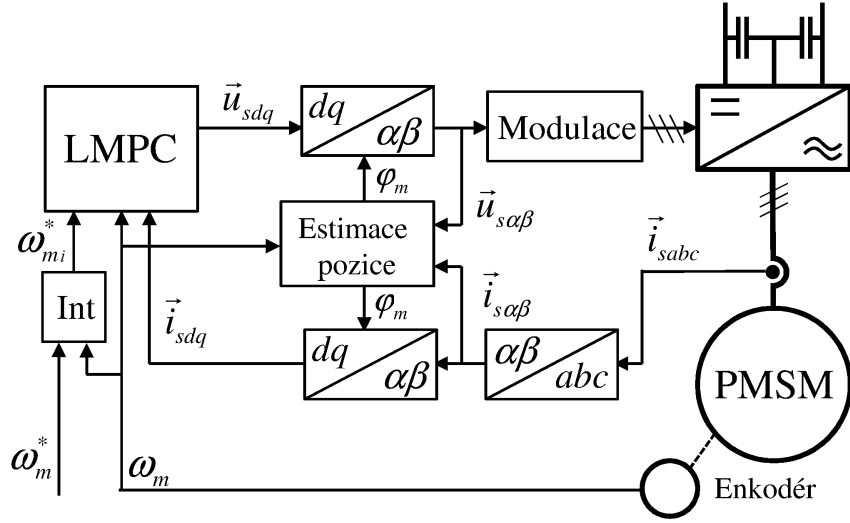
$$(\mathbf{A} - \mathbf{BK})^T \mathbf{P} (\mathbf{A} - \mathbf{BK}) + \mathbf{K}^T \mathbf{R} \mathbf{K} \leq \mathbf{P}, \quad (2.49)$$

kde $\mathbf{K}$ je řídicí zákon. Tento výsledek můžeme interpretovat tak, že na konci predikčního horizontu je třeba stav systému dovést do regionu uvnitř stavového prostoru, v němž takováto Riccatiho rovnice platí.

2.3. Lineární MPC pro řízení synchronního motoru s permanentními magnety

V této podkapitole bude ukázána aplikace lineárního MPC regulátoru pro řízení PMSM motoru (pro přehlednost tuto implementaci nazýváme zkratkou LMPC). Blokové schéma je na obrázku 2.4. Lze vidět, že algoritmus je krom řídicí části identický s algoritmem vektorového řízení. Je použito Clarkovy a Parkovy transformace a regulace tedy probíhá v d, q souřadné soustavě. Rovněž měnič je řízen s pomocí modulátoru. Rozdíl se nachází v řídicí části, kde byly všechny PI regulátory nahrazeny jediným blokem lineárního MPC algoritmu. Jeho vstupem je současný stav systému, tedy statorové proudy $\vec{i}_{sdq} = [i_{sd}, i_{sq}]^T$, rychlost rotoru ω_m a žádaná hodnota ω_m^* , respektive ω_{mi}^* . Hodnotu rychlosti lze získat buďto s pomocí snímače, nebo s pomocí estimátoru (to samé platí pro úhel natočení rotoru φ_m). Přímá složka statorového proudu i_{sd} je regulována na nulovou hodnotu, zatímco s pomocí q -složky je řízen moment i_{sq} (stejně jako u FOC). [5]

Model PMSM motoru byl odvozen v příloze A, kde rovnice (A.7) a (A.8) modelují dynamiku statorového proudu $\vec{i}_{sdq}$, (A.9) modeluje moment motoru a (A.10) představuje diferenciální rovnici rychlosti rotoru. Pro zjednodušení bude uvažován případ PMSM motor s magnety na povrchu rotoru (SMPMSM). Tím odpadá neline-



Obrázek 2.4: Blokového schéma řízení PMSM motoru pomocí LMPC algoritmu

ární reluktanční složka momentu. Rovněž bude zanedbáno tření a zátěžný moment. Přepsání rovnic modelu do tvaru stavového popisu brání nelinearita rovnic statového proudu. Tu tvoří násobek rotorových otáček ω_m a dané složky statového proudu i_{sd} nebo i_{sq} . U FOC se toto řeší metodou decoupling, tedy mimo regulátor. To je však u LMPC nevýhodné, neboť by se tím ztížila implementace napěťového omezení. Model je tedy pro použití v LMPC nutno vhodným způsobem linearizovat. Klasický způsob pomocí rozvoje do Tylorovy řady bohužel nelze použít, neboť nelze jednoduše zvolit pracovní bod, kolem kterého by linearizace probíhala. Zrovna tak adaptivní regulace, kdy by vhodným způsobem byly přizpůsobovány koeficienty v matici zpětných vazeb stavového popisu, není vhodná. Jednak by se tím podstatně zvýšily výpočetní nároky, neboť regulátor by bylo potřeba přepočítávat. Hlavně by však takto nebylo možno využít explicitního řízení, které může být pro nasazení na velmi rychlých systémech klíčové. Používaný způsob namísto toho spočívá v prohlášení nelineárních členů za měřené poruchy. Vektor stavů tak rozšíříme o dvě nové proměnné $\widehat{\omega_m i_{sd}}$ a $\widehat{\omega_m i_{sq}}$. Avšak vzhledem k tomu, že proud i_{sd} je regulován na nulovou hodnotu, bude člen $\widehat{\omega_m i_{sd}}$ pro svůj malý vliv zcela zanedbán. Tím dostáváme rozšířený stavový vektor

$$\vec{x}'(k) = \begin{bmatrix} i_{sd}(k), & i_{sq}(k), & \widehat{\omega_m i_{sq}}(k), & \omega_m(k) \end{bmatrix}^T. \quad (2.50)$$

Po provedení diskretizace dle Eulerovy metody tak dostáváme diskrétní model PMSM motoru

$$\vec{x}(k+1|k) = \underbrace{\begin{bmatrix} 1 - T_s \frac{R_s}{L_d} & 0 & T_s P_p \frac{L_q}{L_d} & 0 \\ 0 & 1 - T_s \frac{R_s}{L_q} & 0 & -T_s P_p \frac{\Psi_{MG}}{L_q} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{3}{2} T_s P_p \Psi_{PM} & 0 & 1 \end{bmatrix}}_{\mathbf{A}_{sm}} \vec{x}(k) + \underbrace{\begin{bmatrix} \frac{T_s}{L_d} & 0 \\ 0 & \frac{T_s}{L_q} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{B}_{sm}} \vec{u}(k), \quad (2.51)$$

kde T_s je vzorkovací perioda, R_s je statorový odpor, L_d a L_q jsou statorové indukčnosti, P_p je počet pólových párů, Ψ_{MG} je magnetický tok permanentního magnetu a $\vec{u}(k) = [u_{sd}, u_{sq}]^T$ je vektor vstupů, respektive statorových napětí. Můžeme si povšimnout, že měřená porucha $\widehat{\omega_m i_{sq}}$ je dle matice $\mathbf{A}_{sm}$ konstanta (na predikčním horizontu nemění svou hodnotu), díky čemuž bude výsledný akční zásah suboptimální. Dá se však předpokládat, že predikční horizont bude při řízení tak rychlého systému spíše kratší a hodnota $\widehat{\omega_m i_{sq}}$ by se za tuto dobu neměla příliš změnit. Tímto způsobem linearizace modelu však regulátor ztratil informaci o vztahu mezi velikostí proudů a měřenými poruchami. Ve výsledku tak lineární MPC regulátor není schopen motor odbudit.

Model (2.51) následně rozšíříme dle principů popsanych v podkapitole 2.2 tak, aby LMPC regulátor umožňoval sledování žádané hodnoty rotorové rychlosti s nulovou ustálenou odchylkou. Té však lze dosáhnout pouze v případě přesného modelu, respektive nepřítomnosti poruch (to v našem případě může být i zátěžný moment). Proto byl žádané hodnotě ω_m^* ve schématu na obrázku 2.4 vřazen blok Int. Ten do regulační smyčky přidává integrační složku, dle vztahu

$$\omega_{mi}^*(t) = \omega_m^*(t) + K_{INT} \int_0^t [\omega_m^*(\tau) - \omega_m(\tau)] d\tau, \quad (2.52)$$

kde ω_{mi}^* je žádaná hodnota rotorové rychlosti včetně integrované řídicí odchylky a K_{INT} je integrační zesílení. Tím dostáváme rozšířený vektor stavů

$$\vec{x}'(k) = \left[i_{sd}(k), i_{sq}(k), \widehat{\omega_m i_{sq}}(k), \omega_m(k), \omega_{mi}^*(k), u_{sd}(k-1), u_{sq}(k-1) \right]^T. \quad (2.53)$$

Výsledný model má potom tvar

$$\vec{x}'(k+1) = \underbrace{\begin{bmatrix} \mathbf{A}_{sm} & \mathbf{0} & \mathbf{B}_{sm} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_2 \end{bmatrix}}_{\tilde{\mathbf{A}}} \vec{x}'(k) + \underbrace{\begin{bmatrix} \mathbf{B}_{sm} \\ \mathbf{0} \\ \mathbf{I}_2 \end{bmatrix}}_{\tilde{\mathbf{B}}} \vec{\delta u}(k), \quad (2.54)$$

kde $\mathbf{I}_2 \in \mathbb{R}^2$ je jednotková matice a $\vec{\delta u}(k) = [\delta u_{sd}(k), \delta u_{sq}(k)]$ je vektor diferencí statorového napětí, respektive vstupů systému. Výsledný akční zásah je vypočten dle vztahů

$$u_{sd}(k) = u_{sd}(k-1) + \delta u_{sd}(k), \quad (2.55)$$

$$u_{sq}(k) = u_{sq}(k-1) + \delta u_{sq}(k). \quad (2.56)$$

Nyní je potřeba nastavit účelovou funkci. Vyjdeme-li z vektor stavů (2.53), pak budou penalizační matice dle definice (2.18) nabývat tvaru

$$\mathbf{Q} = \begin{bmatrix} \gamma_{i_{sd}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{i_{sq}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{\omega} & -\gamma_{\omega} & 0 & 0 \\ 0 & 0 & 0 & -\gamma_{\omega} & \gamma_{\omega} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \gamma_{u_{sd}} & 0 \\ 0 & \gamma_{u_{sq}} \end{bmatrix}, \quad (2.57)$$

kde koeficienty $\gamma_{i_{sd}} > 0$ a $\gamma_{i_{sq}} > 0$ penalizují hodnotu přímé a kvadratické složky statorového proudu a $\gamma_{\omega} > 0$ penalizuje rozdíl mezi žádanou a skutečnou hodnotou rychlosti rotoru. Co se týká penalizace konečného stavu můžeme pro jednoduchost nastavit $\mathbf{P} = \mathbf{Q}$ (vzásadě tím dojde k prodloužení predikčního horizontu).

Poslední část návrhu účelové funkce spočívá ve volbě délky predikčního a řídicího horizontu N a N_u . Bylo by výhodné, aby byl predikční horizont dostatečně dlouhý a pokryl tak dynamiku všech regulovaných veličin. Nicméně u rychlosti ω_m by tento požadavek vedl na extrémně dlouhý predikční horizont a z hlediska výpočetních nároků zcela nereálný. Minimální hodnota N vyplývá z modelu (2.51). Díky dopravnímu zpoždění na měniči nejsme schopni akčním zásahem okamžitě ovlivnit vstup systému. Až v kroku $k+1$ se změní statorové napětí, jež ovlivní proud v kroku $k+2$. Otáčky jsou potom řízeny až se zpožděním tří kroků. Minimální délka predikčního, ale i řídicího horizontu je tedy $N \geq N_u \geq 4$. Co se týká maximální délky, je

ovlivněna především možnostmi hardwaru na kterém bude algoritmus implementován. Vzhledem k náročnosti LMPC algoritmu však tato hodnota bude volena blízko svému minimu.

Nyní zbývá definovat omezení. Ta budou kladena pouze na satorové napětí a satorový proud. Snahou přitom bude napodobit kružnice proudového a napěťového omezení

$$\sqrt{u_{sd}^2(k) + u_{sq}^2(k)} \leq U_{\max} \quad \text{a} \quad \sqrt{i_{sd}^2(k) + i_{sq}^2(k)} \leq I_{\max}, \quad (2.58)$$

kde $U_{\max}$ a $I_{\max}$ je maximální amplituda satorového napětí a proudu. Problém je, že rovnice (2.58) jsou nelineární a bude je potřeba aproximovat. Omezení definujeme ve tvaru

$$\mathbf{A}_{xset} \vec{x}'(k) \leq \vec{b}_{xset}, \quad (2.59)$$

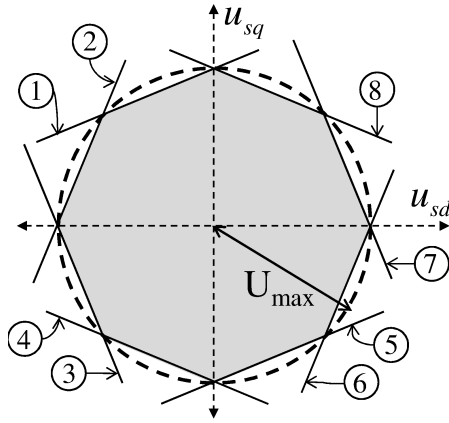
kde $\mathbf{A}_{xset}$ a $\vec{b}_{xset}$ jsou matice a vektor definující omezení stavů formou afinních funkcí stavů. Do tvaru (2.24) lze nerovnici (2.59) převést dosazením stavového prediktoru (2.20). Počet nerovnic omezení je věcí kompromisu mezi kvalitou aproximace rovnic (2.58) a výpočetními nároky, jež se s počtem omezení samozřejmě zvyšují. Obvykle se pak k aproximaci používá n -úhelníku.

Co se týká omezení napětí, je dostatečně dobré aproximace dosaženo při použití šestiúhelníku, případně osmiúhelníku, který můžeme popsat s pomocí soustavy nerovnic

$$\begin{array}{l} \textcircled{1} \rightarrow \\ \textcircled{2} \rightarrow \\ \textcircled{3} \rightarrow \\ \textcircled{4} \rightarrow \\ \textcircled{5} \rightarrow \\ \textcircled{6} \rightarrow \\ \textcircled{7} \rightarrow \\ \textcircled{8} \rightarrow \end{array} \underbrace{\begin{bmatrix} \sqrt{2}-1 & 1 \\ 1 & \sqrt{2}-1 \\ 1 & 1-\sqrt{2} \\ \sqrt{2}-1 & -1 \\ 1-\sqrt{2} & -1 \\ -1 & 1-\sqrt{2} \\ -1 & \sqrt{2}-1 \\ 1-\sqrt{2} & 1 \end{bmatrix}}_{\mathbf{A}_8} \begin{bmatrix} u_{sd}(k-1) \\ u_{sq}(k-1) \end{bmatrix} \leq \underbrace{\begin{bmatrix} U_{\max} \\ U_{\max} \\ U_{\max} \\ U_{\max} \\ U_{\max} \\ U_{\max} \\ U_{\max} \\ U_{\max} \end{bmatrix}}_{\vec{b}_8}. \quad (2.60)$$

Toto napěťové omezení je pak znázorněno na obrázku 2.5.

Co se týká proudového omezení, není potřeba natolik kvalitní aproximace jako v případě napětí, jelikož je přímá složka satorového proudu je regulována na nulu. Stačí tedy použít obdélníku, jež lze popsat pomocí sady nerovnic



Obrázek 2.5: Napěťové omezení aproximované s pomocí osmiúhelníku

$$\begin{aligned}
 \textcircled{1} &\rightarrow \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} i_{sd}(k) \\ i_{sq}(k) \end{bmatrix}}_{\mathbf{A}_4} \leq \underbrace{\begin{bmatrix} i_{sq \max} \\ i_{sd \max} \\ i_{sq \max} \\ i_{sd \max} \end{bmatrix}}_{\vec{b}_4}.
 \end{aligned} \tag{2.61}$$

Proudové omezení je znázorněno na obrázku 2.6.

Pro rovnici (2.59) tak platí

$$\begin{bmatrix} \mathbf{A}_4 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_8 \end{bmatrix} \vec{x}'(k) \leq \begin{bmatrix} \vec{b}_4 \\ \vec{b}_8 \end{bmatrix}. \tag{2.62}$$

Vzhledem k tomu, že u SMPMSM motoru je proud i_{sq} zcela zodpovědný za tvorbu momentu, bude parametr $i_{sq \max}$ volen co nejblíže maximální dovolené hodnotě proudu $I_{\max}$. Co se týká parametru $i_{sd \max}$, je volen tak, aby jeho překročení bylo málo pravděpodobné a zabránilo by se tak problémům s neřešitelností optimalizační úlohy.

Poněkud jiná situace by nastala u motoru s vyniklými póly, kde lze využít reaktanční složky momentu. Zde se potom parametry $i_{sd \max}$ a $i_{sq \max}$ volí například na základě křivky *max torque per ampere* (MTPA). Tu lze určit z rovnice (A.9), kde dosazením

$$|i_{sdq}| = \sqrt{i_{sd}^2 + i_{sq}^2}, \quad i_{sd} = -|i_{sdq}| \sin(\theta) \quad \text{a} \quad i_{sq} = |i_{sdq}| \cos(\theta), \tag{2.63}$$

dostáváme rovnici

$$m = \frac{3}{2} P_p \left(\Psi_{PM} |i_{sdq}| \cos(\theta) + (L_d - L_q) |i_{sdq}|^2 \sin(\theta) \cos(\theta) \right), \tag{2.64}$$

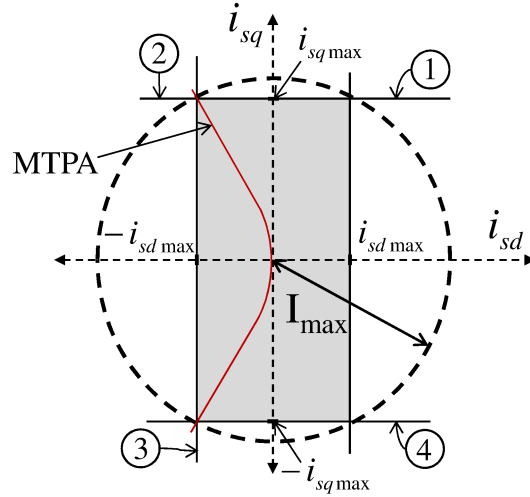
kde θ úhel natočení statorového proudu. Následnou derivací

$$\frac{dm}{dt} = \Psi_{PM} |i_{sdq}| \sin(\theta + (L_d - L_q) |i_{sdq}| (\sin^2(\theta) - \cos^2(\theta))) = 0 \quad (2.65)$$

a dosazením $\theta = \arctan\left(-\frac{i_{sd}}{i_{sq}}\right)$ lze získat vztah

$$i_{sd} + \frac{L_d - L_q}{\Psi_{PM}} (i_{sd}^2 - i_{sq}^2) = 0. \quad (2.66)$$

Rovnice (2.66) popisuje dvě křivky, nicméně maxima momentu je dosaženo jen pro křivku, jež splňuje podmínku $i_{sd} < (L_d - L_q)/2\Psi_{PM}$. Parametry $i_{sd\max}$ a $i_{sq\max}$ jsou pak voleny tak, aby se výsledný proudový vektor po této křivce mohl pohybovat v celé oblasti definované kružnicí proudového omezení. Aby se proudový vektor skutečně po této křivce pohyboval, je samozřejmě nutno LMPC regulátor vhodně upravit. MTPA křivka a odpovídající proudová omezení jsou znázorněna na obrázku 2.6. [6]



Obrázek 2.6: Proudové omezení s pomocí obdélníku

3. Vlastní algoritmy prediktivního řízení založeného na modelu

Hlavním cílem této práce byl vývoj vlastního, či modifikace již existujícího MPC algoritmu. Pozornost byla zaměřena na algoritmus lineárního MPC. Jeho výhody a důvody proč se jedná o perspektivní druh řízení byly shrnuty již v kapitole 2.

I přes použití explicitního řízení je LMPC pro dnešní běžné signálové procesory příliš výpočetně náročné. To brání rozšíření LMPC algoritmu v praxi řízení pohonů. A proto nalezení modifikace LMPC algoritmu pro řízení pohonů, jež by výpočetní nároky snížila bylo prvním cílem bádání v rámci této práce.

Druhým cílem bylo nalézt způsob, jak by LMPC algoritmus mohl získat schopnost motor odbudit za účelem dosažení vyšších otáček. Jak bylo zmíněno v předchozí kapitole, této schopnosti brání použitý způsob linearizace. U FOC se odbuzení dosahuje pomocí vnějších algoritmů odbuzování, což je způsob, který by u LMPC šel použít rovněž. Z hlediska optimality akčního zásahu a omezení by však bylo výhodnější, aby odbuzování bylo výsledkem optimalizační úlohy řešené v rámci jediného regulátoru.

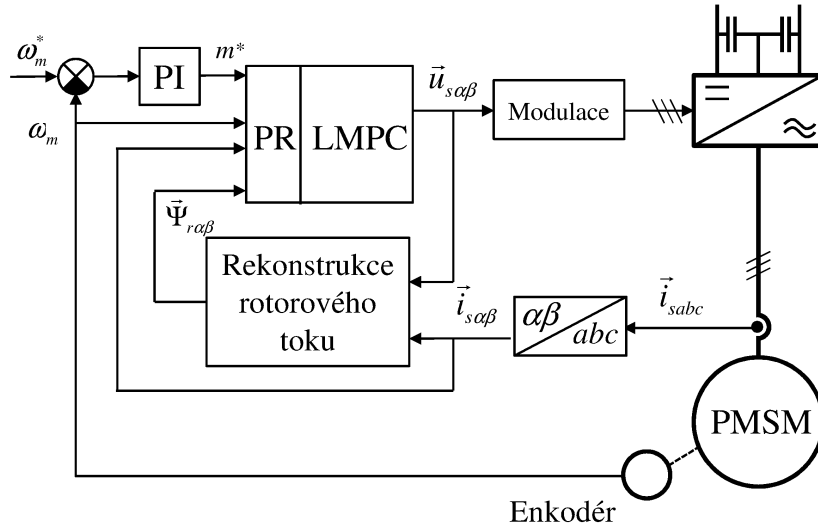
V této kapitole tak budou představeny dva nové typy algoritmů řízení pohonů, jež z lineárního MPC vycházejí. Jako první bude představen algoritmus, který pro účely této práce nazvěme lineární prediktivní řízení momentu založené na modelu (Linear Model Predictive Torque Control či LMPTC). Cílem při vývoji tohoto algoritmu bylo výslednou implementaci lineárního MPC pro PMSM co nejvíce zjednodušit a snížit tím výpočetní nároky algoritmu. Druhý algoritmus, jež zde bude popsán, nazvěme lineární MPC s odbuzováním (Linear Model Predictive Control with Field Weakening či LMPCFW). Jak už název napovídá, algoritmus je schopen PMSM motor odbudit.

Oba algoritmy budou odvozeny pro PMSM motor s magnety na povrchu rotoru, tak jako tomu bylo u LMPC v předchozí kapitole 2.3.

3.1. Lineární prediktivní řízení momentu založené na modelu

Blokové schéma LMPTC algoritmu se nachází na obrázku 3.1. Stejně jako u DTC (viz kapitola 1.1.3) jsou otáčky regulovány pomocí PI regulátoru, který na základě rozdílu mezi žádanou hodnotou rychlosti ω_m^* a skutečnou rychlostí ω_m určuje momentovou referenci m^* . Rozdíl se však nachází v podřízené smyčce. Blok PR (zkratka

pro Proudová Reference) vytváří žádanou hodnotu proudů, které jsou následně řízeny s pomocí lineárního MPC algoritmu. Narozdíl od běžné implementace LMPC, popsané v kapitole 2.3, regulace probíhá ve statorové α, β souřadné soustavě. Odpadá tedy použití Parkovy transformace, jež je spojena s výpočtem goniometrických funkcí sinus a cosinus elektrického úhlu natočení rotoru (viz příloha A). Jelikož jsou v rámci algoritmu rekonstruovány i složky rotorového magnetického toku $\vec{\Psi}_{r\alpha\beta}$, jedná se pouze o malou úsporu výpočetního výkonu. Narozdíl od ACIM se totiž u PMSM dá využít faktu, že amplituda rotorového toku je konstantní (vyhneme se tak operaci dělení).



Obrázek 3.1: Blokové schéma řízení PMSM motoru LMPTC algoritmem

Uveďme nyní rovnice popisující model PMSM motoru s magnety na povrchu ve statorové souřadné soustavě. Rovnice statorového magnetického toku $\vec{\Psi}_{s\alpha\beta} = [\Psi_{s\alpha}, \Psi_{s\beta}]^T$ jsou obdobou rovnic A.6, tedy

$$\Psi_{s\alpha} = Li_{s\alpha} + \Psi_{PM} \cos(P_p \varphi_m) = Li_{s\alpha} + \Psi_{r\alpha}, \quad (3.1)$$

$$\Psi_{s\beta} = Li_{s\beta} + \Psi_{PM} \sin(P_p \varphi_m) = Li_{s\beta} + \Psi_{r\beta}, \quad (3.2)$$

kde $L = L_d = L_q$ je statorová indukčnost, $i_{s\alpha}$ a $i_{s\beta}$ jsou α, β složky statorového proudu $\vec{i}_{s\alpha\beta}$, Ψ_{PM} je magnetický tok permanentního magnetu, φ_m je úhel natočení rotoru, P_p je počet pólových párů a $\Psi_{r\alpha}$ a $\Psi_{r\beta}$ jsou α, β složky rotorového magnetického toku $\vec{\Psi}_{r\alpha\beta}$. Dosazením rovnic (3.1) a (3.2) do rovnice (A.5) lze získat rovnici modelující chování statorového proudu, tedy

$$\frac{di_{s\alpha}}{dt} = -\frac{R_s}{L}i_{s\alpha} + \frac{P_p}{L}\omega_m\Psi_{r\beta} + \frac{1}{L}u_{s\alpha}, \quad (3.3)$$

$$\frac{di_{s\beta}}{dt} = -\frac{R_s}{L}i_{s\beta} + \frac{P_p}{L}\omega_m\Psi_{r\alpha} + \frac{1}{L}u_{s\beta}, \quad (3.4)$$

kde R_s je statorový odpor, ω_m je mechanická úhlová rychlost rotoru a $u_{s\alpha}$ a $u_{s\beta}$ jsou α , β složky statorového napětí $\vec{u}_{s\alpha\beta}$. Jako poslední zbývá odvodit rovnici momentu. Obdobně jako v rovnici (A.9) můžeme psát

$$m = \frac{3}{2}P_p(\vec{\Psi}_{s\alpha\beta} \times \vec{i}_{s\alpha\beta}) = \frac{3}{2}P_p(\Psi_{s\alpha}i_{s\beta} - \Psi_{s\beta}i_{s\alpha}) = \frac{3}{2}P_p(\Psi_{r\alpha}i_{s\beta} - \Psi_{r\beta}i_{s\alpha}). \quad (3.5)$$

Řízení ve statorových α , β souřadnicích však sebou přináší jistou potíž. Zaprvé, aby platily rovnice (3.1) až (3.4), musí se jednat o synchronní motor s permanentními magnety na povrchu. V opačném případě by statorová indukčnost L v osách α , β souřadné soustavy nebyla konstantní, což by znamenalo výraznou nelinearitu modelu.

Dále potřebujeme získat složky vektoru rotorového magnetického toku $\vec{\Psi}_{r\alpha\beta}$. Pro tento účel můžeme použít rekonstruktor, který vychází z rovnic (A.5), (3.1) a (3.2), kdy lze psát

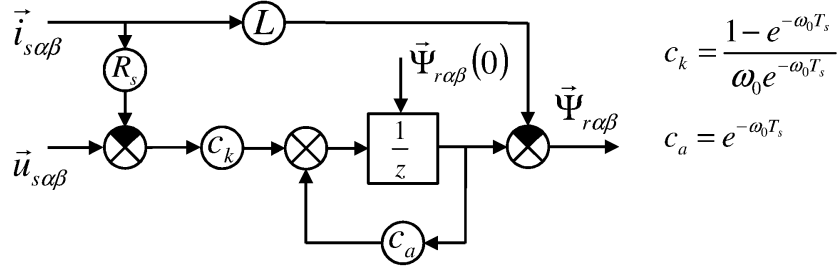
$$\vec{\Psi}_{r\alpha\beta}(t) = \int_0^t \left(\vec{u}_{s\alpha\beta}(\tau) - R_s \vec{i}_{s\alpha\beta}(\tau) \right) d\tau - L \vec{i}_{s\alpha\beta}(t). \quad (3.6)$$

Čistá integrace v této rovnici však přináší problém s jeho driftem. Problém lze odstranit aproximací integrátoru systémem prvního řádu, čímž dojde k utlumení stejnosměrné složky signálu. Takový systém můžeme popsat přenosem

$$F_i(p) = \frac{1}{p + \omega_0}, \quad (3.7)$$

kde ω_0 je mezní frekvence filtru a p je Laplaceův operátor. Použijeme-li Eulerovu metodu diskretizace, pak můžeme rekonstruktor rotorového magnetického toku (3.6) vyjádřit pomocí blokového schématu na obrázku 3.2. Jediným problémem je nyní určení počáteční hodnoty $\vec{\Psi}_{r\alpha\beta}(0)$. Zde lze použít metodu zarovnání či v překladu *alignment*, kdy je po startu řídicí aplikace rotor natočen do definované výchozí pozice. Toho je dosaženo aplikací vybraného vektoru statorového napětí po dostatečně dlouhou dobu.

Dalším problémem, který v α , β souřadné soustavě vyvstal, je nelinearita rovnice (3.5). Díky ní nelze moment regulovat přímo lineárním MPC algoritmem. Nahradit



Obrázek 3.2: Blokové schéma rekonstruktoru rotorového magnetického toku

oba nelineární členy v rovnici (3.5) měřenými poruchami (viz kapitola 2.3) není v tomto případě vhodné. Popsat dynamiku těchto měřených poruch by bylo velmi komplikované a pokud by byly modelovány jako konstanty, pak by se z modelu vytratila informace o dynamice momentu. Místo toho byl nalezen jiný způsob, kdy je v rámci bloku PR (viz obrázek 3.1) z žádané hodnoty momentu m^* vypočítána žádaná hodnota statorového proudu $\vec{i}_{s\alpha\beta}^* = [i_{s\alpha}^*, i_{s\beta}^*]^T$. O regulaci proudů se pak již stará lineární MPC.

Při výpočtu proudové reference $\vec{i}_{s\alpha\beta}^*$ se vyjde z momentové rovnice (3.5). Známé jsou v tuto chvíli složky rotorového magnetického toku $\vec{\Psi}_{r\alpha\beta}$ a žádaná hodnota momentu m^* . Z tohoto pohledu se jedná o rovnici se dvěma neznámými, což vede na nekonečně mnoho řešení. To lze řešit zavedením požadavku na kolmost rotorového magnetického toku a statorového proudu. Vyjádřeno formou skalárního součinu to znamená

$$\vec{\Psi}_{r\alpha\beta} \cdot \vec{i}_{s\alpha\beta}^* = \Psi_{r\alpha} i_{s\alpha}^* + \Psi_{r\beta} i_{s\beta}^* = 0 \rightarrow \Psi_{r\alpha} i_{s\alpha}^* = -\Psi_{r\beta} i_{s\beta}^*, \quad (3.8)$$

kdy se nabízí řešení

$$i_{s\alpha}^* = -h\Psi_{r\beta} \quad \text{a} \quad i_{s\beta}^* = h\Psi_{r\alpha}, \quad (3.9)$$

kde $h \in \mathbb{R}$ je libovolný skalár. Toto řešení zajistí kolmost vektorů $\vec{\Psi}_{r\alpha\beta}$ a $\vec{i}_{s\alpha\beta}^*$ (v případě dostatečně přesného sledování proudové reference skutečným proudovým vektorem i kolmost vektorů $\vec{\Psi}_{r\alpha\beta}$ a $\vec{i}_{s\alpha\beta}$). Nyní je potřeba zajistit, aby výsledná proudová reference vedla na požadovaný moment m^* . Dosazením řešení (3.9) do rovnice (3.5) dostáváme pro žádanou hodnotu momentu

$$m^* = \frac{3}{2}P_p (\Psi_{r\alpha} i_{s\beta}^* - \Psi_{r\beta} i_{s\alpha}^*) = \frac{3}{2}P_p (h\Psi_{r\alpha}^2 + h\Psi_{r\beta}^2) = \frac{3}{2}P_p \Psi_{PM}^2 h. \quad (3.10)$$

Proudovou referenci, jež povede na žádanou hodnotu momentu, tak získáme pomocí vztahů

$$i_{s\alpha}^* = -\frac{2}{3P_p \Psi_{PM}^2} \Psi_{r\beta} m^* \quad \text{a} \quad i_{s\beta}^* = \frac{2}{3P_p \Psi_{PM}^2} \Psi_{r\alpha} m^*. \quad (3.11)$$

Jednoznačnou výhodou tohoto řešení je jeho výpočetní nenáročnost, neboť je zapotřebí pouze operace násobení.

Ve srovnání se zmiňovaným DTC, kde se dle vztahu (1.6) moment řídí úhlem mezi statorovým a rotorovým magnetickým tokem, je u LMPTC úhel mezi $\vec{i}_{s\alpha\beta}$ a $\vec{\Psi}_{r\alpha\beta}$ držen konstantní. Řízení tak probíhá skrze amplitudu statorového proudu. Situace je tedy bližší FOC, kde se rovněž, díky rotující souřadné soustavě, drží pravý úhel mezi rotorovým magnetickým tokem q –složkou statorového proudu.

Nyní již pouze zbývá navrhnout lineární MPC regulátor proudu. V rovnicích statorových proudů (3.3) a (3.4) se nachází nelineární operace násobení rychlosti rotoru ω_m a rotorového magnetického toku $\vec{\Psi}_{r\alpha\beta}$. K linearizaci bude použita stejná metoda, jako v kapitole 2.3. Do stavového vektoru tak budou zavedeny měřené poruchy $\widehat{\omega_m \Psi_{r\alpha}}$ a $\widehat{\omega_m \Psi_{r\beta}}$. Aby bylo možno sledovat žádanou hodnotu proudů, musejí být do rozšířeného stavového vektoru rovněž přidány složky proudové reference $\vec{i}_{s\alpha\beta}^*$, tedy

$$\vec{x}'(k) = \left[i_{s\alpha}(k), i_{s\beta}(k), \widehat{\omega_m \Psi_{r\alpha}}(k), \widehat{\omega_m \Psi_{r\beta}}(k), i_{s\alpha}^*(k), i_{s\beta}^*(k) \right]. \quad (3.12)$$

Oproti LMPC (viz kapitola 2.3) má nyní stavový prostor o dimenzi méně, což samo o sobě vede na nižší výpočetní nároky. Použití lineárního MPC v inkrementální podobě (tedy zavedení integrátoru) zde nemá smysl, jelikož žádaná hodnota proudu není ustálená. Integrační složka by tak pouze zaváděla další zpoždění. Tím bohužel opadá možnost jednoduché penalizace změny akčního zásahu.

S použitím diskretizace pomocí Eulerovy metody pak můžeme psát

$$\vec{x}'(k+1|k) = \begin{bmatrix} 1 - T_s \frac{R_s}{L} & 0 & 0 & T_s \frac{P_p}{L} & 0 & 0 \\ 0 & 1 - T_s \frac{R_s}{L} & -T_s \frac{P_p}{L} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \vec{x}'(k) + \begin{bmatrix} \frac{T_s}{L} & 0 \\ 0 & \frac{T_s}{L} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \vec{u}_{s\alpha\beta}(k). \quad (3.13)$$

Dalším krokem v návrhu lineárního MPC regulátoru proudů je účelová funkce. Ta bude uvažována ve tvaru rovnice (2.18). Penalizační matice stavů a akčního zásahu budou mít tvar

$$\mathbf{Q} = \mathbf{P} = \begin{bmatrix} \gamma_{i_s} & 0 & 0 & 0 & -\gamma_s & 0 \\ 0 & \gamma_{i_s} & 0 & 0 & 0 & -\gamma_{i_s} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\gamma_{i_s} & 0 & 0 & 0 & \gamma_{i_s} & 0 \\ 0 & -\gamma_{i_s} & 0 & 0 & 0 & \gamma_{i_s} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \gamma_{u_s} & 0 \\ 0 & \gamma_{u_s} \end{bmatrix}, \quad (3.14)$$

kde $\gamma_{i_s} \geq 0$ je koeficient penalizující řídicí odchylku mezi složkami satorového proudu a jejich referencí a $\gamma_{u_s} \geq 0$ je koeficient penalizující velikost satorového napětí. Jelikož musí být matice $\mathbf{R}$ pozitivně definitní, je potřeba, aby byl koeficient γ_{u_s} větší než nula. Avšak vzhledem k tomu, že není důvodu velikost satorového napětí penalizovat, měla by být jeho hodnota volena dostatečně malá (relativně vůči parametru γ_{i_s} a maximální velikosti proudů a napětí).

Dle modelu (3.13) jsme schopni pomocí akčního zásahu ovlivnit proudy až v kroku $k+2$. Predikční horizont tedy musí mít velikost $N \geq 2$. O moc vyšší hodnotu se však volit nedoporučuje. Prvním důvodem je samozřejmě výpočetní náročnost. Druhým důvodem je, že rotorový magnetický $\vec{\Psi}_{r\alpha\beta}$ je (dle rovnic (3.1) a (3.2)) funkcí úhlu natočení φ_m . Ten se při vyšších rychlostech může během několika kroků výrazně změnit. Měřené poruchy $\widehat{\omega_m \Psi_{r\alpha}}$ a $\widehat{\omega_m \Psi_{r\beta}}$ jsou přitom modelovány jako konstanty, což by dlouhý predikční horizont zatížilo výraznou chybou.

Poslední částí návrhu lineárního MPC regulátoru proudů jsou omezení. Obdobně, jako v kapitole kapitole 2.3, i zde se budeme snažit aproximovat kružnice napěťového a proudového omezení

$$\sqrt{u_{s\alpha}^2(k) + u_{s\beta}^2(k)} \leq U_{\max} \quad \text{a} \quad \sqrt{i_{s\alpha}^2(k) + i_{s\beta}^2(k)} \leq I_{\max}, \quad (3.15)$$

kde $U_{\max}$ a $I_{\max}$ jsou maximální hodnoty amplitudy satorového napětí a proudů. Můžeme si povšimnout, že narozdíl od LMPC, kde byla omezení kladena pouze na stavy (viz rovnice (2.59)), zde budou kladena i na akční zásah. Obdobně tak bude platit

$$\mathbf{A}_{uset} \vec{u}(k) \leq \vec{b}_{uset} \quad (3.16)$$

kde $\mathbf{A}_{uset}$ a $\vec{b}_{uset}$ jsou matice a vektory definující omezení akčního zásahu.

Nevýhodou oproti LMPC je, že všechny veličiny motoru, tedy satorové napětí a proudy, jsou ve satorové souřadné soustavě harmonickými veličinami. Jinými slovy, vektory, na něž chceme klást omezení, rotují. Nepřesnost aproximace kružnic napěťo-

vého a proudového omezení (3.15) s pomocí n -úhelníku se zde bohužel projeví pulsací amplitud vektorů $\vec{u}_{s\alpha\beta}$ a $\vec{i}_{s\alpha\beta}$ v omezení. To se samozřejmě nepříznivě promítne i na momentu. Z tohoto důvodu je potřeba použít dostatečně dobrou aproximaci omezení u obou veličin, což však jde proti požadavku na nízké výpočetní nároky. Je na uživateli, jak přesnou aproximaci použije, nicméně jako dobrý kompromis se zdá být osmiúhelník, jež byl popsán v rovnici (2.60).

3.2. Lineární MPC s implementací odbuzování

Tento algoritmus upravuje LMPC pro řízení PMSM motoru (viz kapitola 2.3) tak, aby byla získána schopnost odbuzení motoru. U LMPC to není umožněno díky použitému způsobu linearizace. Nelineární členy v rovnicích statorového proudu (A.7) a (A.8), kde se mezi sebou násobí otáčky rotoru ω_m a složky statorového proudu $\vec{i}_{sdq}$, byly prohlášeny za měřené poruchy $\widehat{\omega_m i_{sd}}$ a $\widehat{\omega_m i_{sq}}$. U těchto členů pak byla následně zanedbána (s pomocí lineárních operací poměrně těžko popsatelná) dynamika. Tímto způsobem vznikl model PMSM motoru (2.51). Avšak díky tomu, že jsou tyto měřené poruchy modelovány jako konstanty, není lineární MPC regulátor schopen motor odbudit. Je totiž ztracena informace o možnosti působit magnetickým tokem, vytvořeným proudem i_{sd} , proti magnetickému toku permanentního magnetu Ψ_{PM} , což by vedlo ke snížení zpětného elektromotorického napětí. Za cenu nižšího maximálního momentu motoru by pak bylo dosaženo vyšších otáček. Snaha tuto informaci do modelu navrátit pak vedla k návrhu Linear Model Predictive Control with Field Weakening (LMPCFW) algoritmu.

Stejně jako v kapitole 2.3 i LMPCFW algoritmus bude odvozen pro synchronní motor s permanentními magnety s magnety na povrchu (platí $L_d \doteq L_q$). Díky tomu lze zanedbat reluktanční složku moment. Z důvodu obecnosti však v následujících rovnicích nebudeme indukčnosti L_d a L_q mezi sebou krátit. Pokud s pomocí Eulerovy metody diskretizujeme rovnice statorových proudů (A.7) a (A.8), pak můžeme psát

$$i_{sd}(k+1) = \left(1 - T_s \frac{R_s}{L_d}\right) i_{sd}(k) + T_s P_p \frac{L_q}{L_d} \omega_m(k) i_{sq}(k) + \frac{T_s}{L_d} u_{sd}(k), \quad (3.17)$$

$$i_{sq}(k+1) = \left(1 - T_s \frac{R_s}{L_q}\right) i_{sq}(k) - T_s P_p \left(\frac{L_d}{L_q} i_{sd}(k) + \frac{\Psi_{PM}}{L_q} \right) \omega_m(k) + \frac{T_s}{L_q} u_{sq}(k), \quad (3.18)$$

kde T_s je vzorkovací perioda, R_s je statorový odpor, L_d a L_q jsou indukčnosti dané ose, P_p je počet pólových párů, Ψ_{PM} je magnetické tok permanentního magnetu a u_{sd} , u_{sq} jsou složky statorového napětí $\vec{u}_{sdq}$. S cílem rovnici (3.17) a (3.18) linearizo-

vat byl nalezen způsob, kdy je satorové napětí $\vec{u}_{sdq}$ nahrazeno veličinou, jež můžeme nazvat virtuální satorové napětí $\vec{v}_{sdq}$. Pro satorové napětí tak zavedme rovnice

$$u_{sd}(k) = \underbrace{-L_q P_p (\omega_m(k) - \Omega_m) i_{sq}(k)}_{c_{sd}(k)} + v_{sd}(k), \quad (3.19)$$

$$u_{sq}(k) = \underbrace{L_q P_p (\omega_m(k) - \Omega_m) i_{sd}(k)}_{c_{sq}(k)} + v_{sq}(k), \quad (3.20)$$

kde $\Omega_m \in \mathbb{R}$ je konstanta rychlosti a c_d a c_q nazvěme přímým a kvadraturním korekčním členem. Po dosazení vztahů (3.19) a (3.20) do rovnic (3.17) a (3.18) pak dostáváme již lineární rovnice

$$i_{sd}(k+1) = \left(1 - T_s \frac{R_s}{L_d}\right) i_{sd}(k) + T_s P_p \frac{L_q}{L_d} \Omega_m i_{sq}(k) + \frac{T_s}{L_d} v_{sd}(k), \quad (3.21)$$

$$i_{sq}(k+1) = \left(1 - T_s \frac{R_s}{L_q}\right) i_{sq}(k) - T_s P_p \frac{L_d}{L_q} \Omega_m i_{sd}(k) - T_s P_p \frac{\Psi_{PM}}{L_q} \omega_m(k) + \frac{T_s}{L_q} v_{sq}(k). \quad (3.22)$$

Proměnná ω_m byla nahrazena konstantou Ω_m , čímž byla odstraněna nelineární operace násobení dvou proměnných. Z rovnic (3.21) a (3.22) přitom nezmizela informace o vztahu mezi přímou složkou proudu i_{sd} a zpětným elektromotorickým napětím. LMPCFW regulátor je tak nyní schopen motor odbudit.

Nyní však vyvstávají dvě otázky. První je, jakým způsobem zvolit konstantu rychlosti Ω_m a druhou otázkou je, jakým způsobem omezovat satorové napětí $\vec{u}_{sdq}$, které nyní v modelu satorových proudů (3.21) a (3.22) nevystupuje.

Co se týká prvního problému, tedy volby konstanty Ω_m , pak by její hodnota měla být volena co nejbližší aktuální rychlosti ω_m tak, aby byl model co nejpřesnější a výsledný akční zásah byl co nejbližší optimální hodnotě. Toho by šlo dosáhnout s použitím adaptivního lineárního MPC regulátoru. Model satorových proudů (3.21) a (3.22) by pak byl přizpůsobován (ať už pravidelně či při příliš velké odchylce mezi ω_m a Ω_m) tak, aby pokud možno platilo $\Omega_m \doteq \omega_m$. To by však vedlo k výraznému zvýšení výpočetních nároků. Tento způsob tedy není vhodný.

Opačným extrémem by byla volba Ω_m jako jediné hodnoty pro celý rozsah rychlosti ω_m . V takovém případě by byl model satorových proudů (3.21) a (3.22) zatížen chybou, úměrnou (mnohdy výraznému) rozdílu skutečné rychlosti ω_m a hodnoty Ω_m . Problém by pak nastal také v případě reverzace. Pokud by totiž měla konstanta Ω_m opačné znaménko než rychlost ω_m , pak by LMPCFW regulátor po dosažení na-

přetřetího omezení motor nabudil (hodnota Ψ_{sd} by byla zvýšena), což je nežádáný efekt.

Problém lze řešit použitím po částech afinního modelu systému (v překladu PieceWise Affine, či PWA systém). Je to systém, který se skládá z konečného počtu lokálních dynamik, každá platná v odpovídající oblasti R ve stavovém prostoru. Vyjdeme-li z definice diskrétního stavového popisu (2.4) a (2.5), pak můžeme PWA systém popsat jako

$$\left. \begin{aligned} \vec{x}(k+1|k) &= \mathbf{A}_i \vec{x}(k) + \mathbf{B}_i \vec{u}(k) + \vec{\rho}_i \\ \vec{y}(k|k) &= \mathbf{C}_i \vec{x}(k) + \mathbf{D}_i \vec{u}(k) + \vec{\sigma}_i \end{aligned} \right\} \text{pokud } \vec{x}(k) \in R_i, \quad (3.23)$$

kde $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ a $\mathbf{D}_i$ jsou matice a $\vec{\rho}_i$ a $\vec{\sigma}_i$ jsou vektory popisující i -tou dynamiku systému příslušnou oblasti R_i ve tvaru mnohostěnu. Rozměry vektorů vstupů, stavů a výstupů tedy zůstávají zachovány. Jediné, co se mění, jsou matice popisující dynamiku systému. Na základě PWA systému potom lze vytvořit implicitní, ale i explicitní lineární MPC regulátor. [27]

V našem případě tedy využijeme PWA systému, kdy pro danou oblast rychlosti $\omega_m \in \langle \omega_{m_i}, \omega_{m_{i+1}} \rangle$, vystupuje, v části dynamiky modelující statorové proudy dle rovnic (3.21) a (3.22), konstanta $\Omega_m \in \langle \omega_{m_i}, \omega_{m_{i+1}} \rangle$. Nutno podotknout, že s počtem oblastí $i \in \mathcal{I}$, do kterých bude stavový prostor rozdělen, prudce vzrůstají výpočetní nároky. Především u explicitního řízení je toto problémem, neboť během predikčního horizontu N se musí počítat s tím, že stav systému může projít všemi $\mathcal{I}^N$ variacemi modelovaných dynamik. Snadno se tak může stát, že výsledný regulátor se stane příliš složitým i při nízkém počtu oblastí $\mathcal{I}$. Počet $\mathcal{I}$ tedy musí být kompromisem mezi požadovanou mírou přesnosti modelu (tomu odpovídající optimalitě akčního zásahu) a výpočetními nároky.

Vzhledem k tomu, že si budou dynamiky v navazujících regionech našeho PWA systému velmi podobné (jediné, co se mění je konstanta Ω_m), a že je použit velmi krátký predikční horizont, lze uvážit alternativu použití $\mathcal{I}$ separátních regulátorů. Mezi těmito regulátory by se přepínalo v závislosti na rychlosti ω_m . Každý z těchto i regulátorů by pak byl navržený pro jednoduchý systém, v němž by figurovala konstanta Ω_{m_i} . Na jednu stranu by utrpěla optimalita akčního zásahu (predikce stavů by byly počítány v rámci jediného, aktuálně použitého systému), nicméně by výrazně poklesly výpočetní nároky (zvláště v případě explicitního řízení).

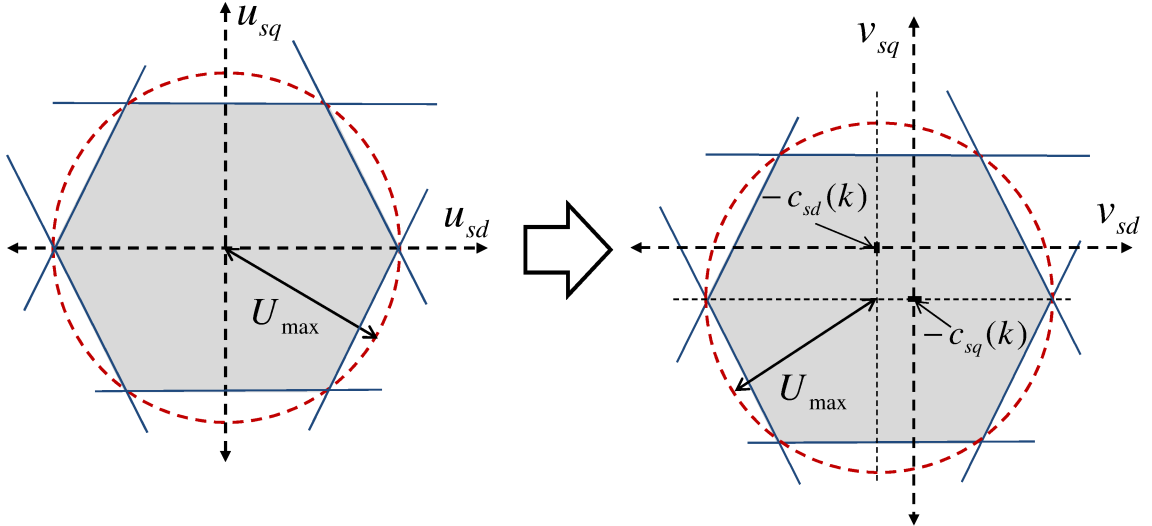
Nyní se věnujme zmíněnému problému omezení statorového napětí $\vec{u}_{sdq}$. V rovnicích (3.21) a (3.22) se totiž vyskytuje pouze virtuální akční zásah $\vec{v}_{sdq}$ a skutečný akční zásah v podobě $\vec{u}_{sdq}$ je nutno vypočítat mimo LMPCFW regulátor z rovnic

(3.19) a (3.20). Omezení $\vec{u}_{sdq}$ by tak sice šlo přesunout mimo LMPCFW regulátor, nicméně tento způsob by kompletně negoval hlavní výhodu MPC algoritmu v podobě optimálního (či alespoň suboptimálního) akčního zásahu v přítomnosti omezení.

Podařilo se však nalézt jiné řešení. Nejprve dosaďme vztahy (3.19) a (3.20) do rovnice kružnice napěťového omezení (2.58). Tím dostáváme

$$U_{\max} \leq \sqrt{(v_{sd}(k) + c_{sd}(k))^2 + (v_{sq}(k) + c_{sq}(k))^2}, \quad (3.24)$$

což se dá interpretovat jako rovnice kružnice v rovině virtuálních napětí v_{sd} a v_{sq} se středem v bodě $[-c_{sd}(k), -c_{sq}(k)]$. Pokud bychom uvažovali omezení pomocí šestiúhelníku, mohli bychom situaci znázornit tak, jako na obrázku 3.3. Na tomto obrázku vlevo vidíme požadovaný způsob omezení satorového napětí. Vpravo je potom odpovídající omezení virtuálního satorového napětí $\vec{v}_{sdq}$.



Obrázek 3.3: Interpretace omezení satorového napětí jako kružnice s pohyblivým středem v rovině virtuálního napětí

U LMPCFW regulátoru se u napěťového omezení budeme snažit aproximovat kružnici omezení virtuálního napětí s pohyblivým středem (3.24). Toho lze naštěstí dosáhnout pomocí lineárních operací. V našem případě s využitím vztahů (3.19) a (3.20) můžeme psát

$$\mathbf{A}_{6u} \vec{u}_{sdq}(k) = \mathbf{A}_{6u} (\vec{v}_{sdq}(k) + \vec{c}_{sdq}(k)) = \begin{bmatrix} \mathbf{A}_{6u} & \mathbf{A}_{6u} \end{bmatrix} \begin{bmatrix} \vec{v}_{sdq}(k) \\ \vec{c}_{sdq}(k) \end{bmatrix} \leq \vec{b}_6, \quad (3.25)$$

kde $\vec{c}_{sdq} = [c_{sd}, c_{sq}]$ a

$$\mathbf{A}_{6u} = \begin{bmatrix} -1 & -1 & 0 & 0 & 1 & 1 \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & -\frac{2}{\sqrt{3}} \end{bmatrix}^T, \quad (3.26)$$

$$\vec{b}_{6u} = \begin{bmatrix} U_{\max} & U_{\max} & U_{\max} & U_{\max} & U_{\max} & U_{\max} \end{bmatrix}, \quad (3.27)$$

kde $U_{\max}$ je poloměr kružnice napětového omezení. Korekční členy $\vec{c}_{sdq}$ však nyní musí být součástí stavového vektoru. Vzhledem k nelinearitě těchto členů bude jejich dynamika zanedbána a tak stejně, jako v kapitole 2.3, budou tyto členy prohlášeny za měřené poruchy.

Šestiúhelníkové omezení bylo zvoleno, jelikož se v d, q rovině satorového napětí nenachází v bodě $[0, U_{\max}]$ jeho vrchol (tak jako u osmiúhelníku). Díky krátkému predikčnímu horizontu by se totiž regulátor snažil umístit vektor satorového napětí do tohoto bodu (byla by zde největší amplituda satorového napětí největší). Tím by však nedošlo k odbuzení motoru, neboť by napětí u_{sd} bylo velmi malé.

Nyní se již dostáváme ke konstrukci výsledného modelu, z něž vychází LMP-CFW algoritmus. Při uvážení požadavku na sledování žádané hodnoty otáček ω_{mi}^* s nulovou ustálenou odchylkou, dostáváme rozšířený vektor stavů a vstupů

$$\begin{aligned} \vec{x}'(k) &= [i_{sd}(k), i_{sq}(k), \omega_m(k), \omega_{mi}^*(k), c_{sd}(k), c_{sq}(k), v_{sd}(k-1), v_{sq}(k-1)]^T, \\ \vec{u}(k) &= [v_{sd}(k), v_{sq}(k)]^T. \end{aligned} \quad (3.28)$$

Pro danou oblast rychlosti ω_m pak můžeme systém definovaný rovnicemi (3.23) popsat pomocí matic

$$\tilde{\mathbf{A}}_i = \underbrace{\begin{bmatrix} 1 - T_s \frac{R_s}{L_d} & T_s P_p \frac{L_q}{L_d} \Omega_{m_i} & 0 & 0 & 0 & 0 & \frac{T_s}{L_d} & 0 \\ -T_s P_p \frac{L_d}{L_q} \Omega_{m_i} & 1 - T_s \frac{R_s}{L_q} & -T_s P_p \frac{\Psi_{PM}}{L_q} & 0 & 0 & 0 & 0 & \frac{T_s}{L_d} \\ 0 & \frac{3}{2} T_s P_p \Psi_{PM} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{pro } \omega_m(k) \in \langle \omega_{m_i}, \omega_{m_{i+1}} \rangle}, \quad (3.29)$$

$$\tilde{\mathbf{B}}_i = \begin{bmatrix} \frac{T_s}{L_d} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{T_s}{L_d} & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \quad (3.30)$$

kde $i \in \mathcal{I}$ je počet oblastí s hranicemi ω_{m_i} a $\omega_{m_{i+1}}$, do nichž je ω_m rozdělena.

Účelová funkce je ve všech ohledech stejná, jako u LMPC algoritmu (viz kapitola 2.3) a nebude zde znovu probírána. Rovněž blokové schéma 2.4 bez větších změn odpovídá situaci s LMPCFW algoritmem.

Co se týká omezení proudů, pak je k aproximaci kružnice proudového omezení (2.58) možné použít obdélníku popsaneho rovnicí (2.61). Nicméně platí, že hodnota $i_{sd\max}$ bude oproti LMPC mnohem větší. Ve výsledku tak obdélník může přestat vyhovovat, neboť dojde ke zbytečnému omezení proudů i_{sq} (a tedy i maximálního momentu) nebo k překročení proudového omezení. Ač za cenu vyšších výpočetních nároků, je tedy lepší použít šestiúhelníku. Ten lze popsat pomocí soustavy rovnic

$$\underbrace{\begin{bmatrix} -\frac{1}{\sqrt{3}} & -1 \\ \frac{1}{\sqrt{3}} & -1 \\ \frac{2}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{3}} & 1 \\ -\frac{1}{\sqrt{3}} & 1 \\ -\frac{2}{\sqrt{3}} & 0 \end{bmatrix}}_{\mathbf{A}_{6i}} \begin{bmatrix} i_{sd}(k) \\ i_{sq}(k) \end{bmatrix} \leq \underbrace{\begin{bmatrix} I_{\max} \\ I_{\max} \\ I_{\max} \\ I_{\max} \\ I_{\max} \\ I_{\max} \end{bmatrix}}_{\tilde{b}_{6i}}, \quad (3.31)$$

kde $I_{\max}$ je poloměr kružnice proudového omezení.

4. Simulace navržených algoritmů

V této kapitole budou představeny výsledky simulací algoritmů popsanych v kapitole 3, tedy algoritmů LMPTC a LMPCFW pro řízení synchronního motoru s permanentními magnety. Spolu s nimi budou pro srovnání simulovány algoritmy LMPC (viz kapitola 2.3) a vektorového řízení (viz příloha B).

Cílem simulací bylo především zjistit obecné vlastnosti nově navržených algoritmů. Pozornost bude věnována především jejich chování algoritmů během omezení proudů a napětí. U LMPCFW algoritmu pak bude srovnáno chování implicitní a explicitní verze regulátoru, kdy implicitní verze bude postavena na PWA systému, zatímco explicitní bude složena z několika samostatných regulátorů tak, jak bylo naznačeno v kapitole 3.2.

Jako simulačního nástroje bylo použito prostředí MATLAB-Simulink R2012a 64-bit. Pro implementaci algoritmů lineárního MPC v prostředí MATLAB byl použit Multi-Parametric Toolbox (MPT) verze 3.0.17. Samotná optimalizační úloha kvadratického programování pak byla řešena s pomocí výkonného solveru GUROBI verze 5.6. [14], [13], [19]

Všechny algoritmy byly simulovány pro řízení trojpólového synchronního motoru s permanentními magnety na povrchu rotoru jehož parametry jsou v tabulce 4.1. Tento motor byl modelován pomocí spojitého modelu z knihovny Electromechanical Drives Toolbox. [25]

Parametr	Jednotka	Hodnota
R_s	$[\Omega]$	0,8
L_d	$[\text{mH}]$	6,5
L_q	$[\text{mH}]$	6,5
P_p	$[-]$	3
Ψ_{PM}	$[\text{Vs}]$	0,3
J	$[\text{kgm}^2]$	$0,8 \cdot 10^{-3}$
$U_{\max}$	$[\text{V}]$	150
$I_{\max}$	$[\text{A}]$	12

Tabulka 4.1: Parametry synchronního motoru s permanentními magnety použitého v simulaci [5]

Simulace byly v některých ohledech zjednodušeny. Byla například vynechána napěťová modulace, jelikož její vliv na výsledek byl považován za zanedbatelný. Všechny parametry motoru se během simulace nemění.

Co se týká společného nastavení, v prostředí MATLAB-Simulink byla použita metoda integrace ode3 a krok výpočtu byl nastaven pevně na hodnotu $0,1T_s$. Další specifické aspekty nastavení a návrhu algoritmů budou popsány v následujícím textu (krom algoritmu vektorového řízení, jehož nastavení je uvedeno v příloze C)

4.1. Návrh regulátoru lineárního prediktivního řízení založeného na modelu

Lineární MPC regulátor pro řízení synchronního motoru s permanentními magnety byl obecně popsán v kapitole 2.3. Jelikož je model dle rovnice (2.51), respektive (2.54) plně dán parametry motoru uvedených v tabulce 4.1, vše, co z návrhu LMPC regulátoru zbývá je nastavení účelové funkce (2.18) a omezení proudů a napětí.

Pro jednodušší nastavení účelové funkce, a rovněž z důvodu optimalizace počtu regionů explicitní verze regulátoru, je vhodné použít normalizace modelu. LMPC regulátor postavený na normalizovaném modelu je pak také mnohem snazší přenést na jiný PMSM motor. Relativní poměry koeficientů matic účelové funkce $\mathbf{Q}$, $\mathbf{P}$ a $\mathbf{R}$ se totiž vůči maximálním rozsahům stavů, vstupů a výstupů systému nezmění. Pro rozšířený vektor stavů (2.53) a vektor vstupů systému tak můžeme psát

$$\vec{x}(k) = \left[\frac{i_{sd}(k)}{I_{Nrm}}, \frac{i_{sq}(k)}{I_{Nrm}}, \frac{\widehat{\omega_m i_{sq}(k)}}{I_{Nrm} \Omega_{Nrm}}, \frac{\omega_m(k)}{\Omega_{Nrm}}, \frac{\omega_{mi}^*(k)}{\Omega_{Nrm}}, \frac{u_{sd}(k-1)}{U_{Nrm}}, \frac{u_{sq}(k-1)}{U_{Nrm}} \right], \quad (4.1)$$

$$\vec{\delta u}(k) = \left[\frac{\delta u_{sd}(k)}{U_{Nrm}}, \frac{\delta u_{sq}(k)}{U_{Nrm}} \right], \quad (4.2)$$

kde I_{Nrm} , Ω_{Nrm} , a U_{Nrm} jsou normy proudů, úhlové rychlosti rotoru a statorového napětí. Pokud bychom chtěli veličiny normalizovat do rozsahu $\langle -1; 1 \rangle$ (jako v našem případě), pak bychom tyto normy volili jako maximální dosažitelné hodnoty dané veličiny (tedy $U_{Nrm} = 150$ V, $I_{Nrm} = 12$ A a $\Omega_{Nrm} = 180$ rad/s). Norma Ω_{Nrm} byla zvolena na základě zjednodušeného výpočtu maximální hodnoty dosažitelných otáček $|\Omega_{m_{\max}}|$, které můžeme určit dle vztahu

$$|\Omega_{m_{\max}}| = \frac{U_{\max}}{P_p \Psi_{PM}} = \frac{150}{3 \cdot 0,3} \doteq \underline{\underline{167 \text{ rad/s}}}. \quad (4.3)$$

Je však třeba počítat s tím, že integrační složka reference rychlosti může výslednou požadovanou hodnotu otáček ω_{mi}^* zvýšit nad tuto hodnotu, a proto byla norma zvýšena na 180 rad/s.

Odpovídajícím způsobem pak normalizujeme rovnice i model (2.54), tedy

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 - T_s \frac{R_s}{L_d} & 0 & \frac{T_s P_p}{\Omega_{Nrm}} \frac{L_q}{L_d} & 0 & 0 & \frac{T_s}{L_d} \frac{U_{Nrm}}{I_{Nrm}} & 0 \\ 0 & 1 - T_s \frac{R_s}{L_q} & 0 & -T_s P_p \frac{\Psi_{PM}}{L_q} \frac{\Omega_{Nrm}}{I_{Nrm}} & 0 & 0 & \frac{T_s}{L_q} \frac{U_{Nrm}}{I_{Nrm}} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{2} T_s P_p \Psi_{PM} \frac{I_{Nrm}}{\Omega_{Nrm}} & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.4)$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} \frac{T_s}{L_d} \frac{U_{Nrm}}{I_{Nrm}} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{T_s}{L_q} \frac{U_{Nrm}}{I_{Nrm}} & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T. \quad (4.5)$$

Co se týká nastavení účelové funkce, byl zvolen poměrně krátký predikční a řídicí horizont $N = 4$ a $N_u = 2$. Přímá a kvadrurní složka satorového proudu i_{sd} a i_{sq} byla v maticích (2.57), penalizována koeficienty $\gamma_{i_{sd}} = 5$ a $\gamma_{i_{sq}} = 0,15$. Proud i_{sd} tak bude držen blízky nule, zatímco i_{sq} bude nulový pouze v případě, kdy není vyžadován moment. Rozdíl skutečné a žádané úhlové rychlosti byl penalizován hodnotou $\gamma_\omega = 200$. Sledování žádané hodnoty rychlosti je tedy prioritou regulátoru. Co se týká penalizace změny akčního zásahu $\vec{\delta u}(k)$, byly shodně nastaveny na hodnotu $\gamma_{u_{sd}} = \gamma_{u_{sq}} = 1 \cdot 10^{-3}$. Vyšší hodnoty by byly voleny v případě přítomnosti šumu v systému, což však není náš případ a následné zpomalení regulační smyčky není žádané. Konečný stav byl penalizován maticí $\mathbf{P} = \mathbf{Q}$ (vzhledem k tomu, že k problémům se stabilitou nedocházelo, nebylo potřeba volit jinak).

Jako omezení satorového napětí bylo použito osmiúhelníku (2.60), jež byl vepsán kružnici napěťového omezení o poloměru 150 V. U omezení satorového proudu pomocí obdélníku bylo potřeba zvolit parametry $i_{sd \max}$ a $i_{sq \max}$. Vzhledem k tomu, že proud i_{sd} bude držen blízko nule, může být zvolena maximální hodnota proudu $i_{sq \max} = I_{\max} = 12$ A. Z důvodu zajištění řešitelnosti optimalizační úlohy však nejde omezovat proud i_{sd} na nulu a je tedy zvolena hodnota $i_{sd \max} = 2,4$ A. Všechna omezení, pak bylo nutné normalizovat stejně, jako v případě modelu (4.4).

Jak bylo popsáno v kapitole 2.2, solvery úlohy kvadratického programování vyžadují, aby se počáteční stav systému vždy nacházel uvnitř omezeními definované oblasti. Vzhledem k tomu, že použitý model je pouze linearizací skutečného modelu, je výsledný akční zásah suboptimální. To může vést k situaci, kdy měřený stav, například satorový proud, překročí své omezení a úloha se tím stane neřešitelnou. Stejná situace samozřejmě může nastat i v důsledku přítomnosti poruch a šumu.

Aby se neřešitelnosti předešlo, nejsou na první krok predikčního horizontu kladena žádná omezení. Ta nebyla pro snížení výpočetní náročnosti kladena ani na konečný stav systému.

V případě explicitní verze algoritmu bylo pro redukci počtu regionů vhodné co nejvíce zmenšit oblast stavového prostoru, ve které je explicitní regulátor počítán. Toho bylo dosaženo přidáním jednoduchých omezení typu maximum a minimum pro všechny stavy systému. Tato omezení jsou volena tak, aby jich nikdy nebylo dosaženo, čili mírně větší než je norma daného stavu, čímž se předejde případným problémům s neřešitelností optimalizační úlohy.

Vzorkovací perioda T_s byla nastavena na hodnotu 200 μs . Dalo by se očekávat, že jelikož nebyla použita kaskádní řídicí struktura, bude tato hodnota vyšší. Bohužel díky nepřesnosti modelu (například způsob linearizace nebo použití Eulerovy metody diskretizace) se s délkou vzorkovací periody zvyšuje i chyba predikce. Tím by samozřejmě degradovala kvalita regulace.

Nakonec je nutno zvolit integračního zesílení K_{INT} v rovnici (2.52). Tato hodnota může výrazně ovlivnit výslednou dynamiku regulace rychlosti, neboť ovlivňuje velikost překmitu. Ten je sice nežádoucí, nicméně bez integrační složky by nebylo možno dosáhnout nulové ustálené odchylky v přítomnosti poruch (například zátěžného momentu) a dalších nepřesností v modelu. Uspokojivých výsledků bylo dosaženo s hodnotou $K_{INT} = 40$, kdy integrační složka byla saturována na hodnotu $\langle -10; 10 \rangle$ rad/s. Saturace je nutná k prevenci windup efektu, jež by nastal, pokud by byly požadovány otáčky vyšší než dosažitelné.

4.2. Návrh regulátoru lineárního prediktivního řízení momentu založeného na modelu

Algoritmus LMPTC byl obecně popsán v kapitole 3.1. Tento algoritmus využívá kaskádní regulační smyčky, kdy nadřízený regulátor tvoří otáčkový PI regulátor a v podřízené smyčce se nachází LMPC regulátor proudů.

Regulátor proudů je založen na modelu (3.13). Ten byl, stejně jako v kapitole 3.2, normalizován pro rozsah $\langle -1; 1 \rangle$. Příslušné normy pak byly voleny stejně jako u LMPC v kapitole 3.2. Vzhledem k jednoduchosti modelu byl zvolen velmi krátký predikční a řídicí horizont $N = N_u = 2$. Koeficienty penalizačních matic (3.14) byly nastaveny $\gamma_{i_s} = 100$ a $\gamma_{u_s} = 10^{-4}$. Poměrně vysoká hodnota koeficientu γ_{i_s} byla použita, jelikož je potřeba, aby vektor proudů $\vec{i}_{s\alpha\beta}$ svou referenci následoval dostatečně přesně a byl tak kolmý na vektor $\vec{\Psi}_{r\alpha\beta}$. To je nutné, aby byla zajištěna platnost vztahů (3.11). Penalizace konečného stavu byla opět zvolena $\mathbf{P} = \mathbf{Q}$.

Pro omezení proudů a napětí bylo v obou případech použito osmiúhelníků, které lze popsat obdobně jako v rovnici (2.60). Proudové omezení bylo vepsáno kružnici o poloměru 12 A a napěťové pak kružnici o poloměru 150 V (respektive jednotkové kružnici, jelikož používáme normalizovaný model systému). Aby byla zajištěna řešitelnost optimalizační úlohy, byla vynechána proudová omezení v prvním kroku predikčního horizontu.

Vzorkovací perioda byla zvolena $T_s = 200 \mu\text{s}$, což se ukázala jako postačující hodnota (i vzhledem k řízení v α, β souřadné soustavě)

Co se týká regulátoru otáček, bylo použito diskrétního PI regulátoru s přenosem

$$F_{PI}(z) = K_{PI} \left(1 + \frac{T_s}{T_I} \frac{z^{-1}}{1 - z^{-1}} \right), \quad (4.6)$$

kde K_{PI} je zesílení, T_I je integrační časová konstanta a z je operátor z-transformace. Vzhledem k tomu, že se v podřízené smyčce nachází LMPC regulátor, který je sám o sobě nelineární, není možné PI regulátor nastavit jinak než experimentálně. V našem případě byly použity hodnoty $K_{PI} = 1$ a $T_I = 0,01 \text{ s}$. Do PI regulátoru bylo rovněž implementováno omezení výstupu na rozmezí $\langle -16, 2; 16, 2 \rangle \text{ Nm}$, což odpovídá maximálnímu momentu motoru

$$|m_{\max}| = \frac{3}{2} J \Psi_{PM} I_{\max} = \frac{3}{2} \cdot 3 \cdot 0,3 \cdot 12 \doteq \underline{\underline{16,2 \text{ Nm}}}. \quad (4.7)$$

Vzorkovací perioda PI regulátoru byla, vzhledem k rychlosti dynamiky otáčkové smyčky, zvolena na hodnotu pětikrát vyšší než u zbytku algoritmu.

Nakonec je ještě potřeba zmínit nastavení rekonstruktoru rotorového magnetického toku $\vec{\Psi}_{r\alpha\beta}$, respektive nastavení mezní frekvence filtru ω_0 (viz přenos (3.7)). Dobrá aproximace čisté integrace a odstranění driftu integrátoru byla zajištěna při hodnotě $\omega_0 = 0,01 \text{ s}^{-1}$.

4.3. Návrh regulátoru lineárního MPC s odbuzením

Teorie LMPCFW algoritmu byla popsána v kapitole 3.2. Jedná se v zásadě o modifikaci LMPC a mnohá nastavení tak budou podobná, jako v kapitole 3.2.

Implicitní verze algoritmu LMPCFW vychází z PWA systému (3.23). Ten se v našem případě bude skládat ze tří lokálně platných dynamik, jež lze popsat pomocí matic (3.29). Každá z těchto dynamik pak přísluší části stavového prostoru, respektive oblasti rotorových otáček ω_m . Počet těchto dynamik je kompromisem

mezi aproximací rychlosti ω_m konstantou Ω_m a výpočetními nároky. Pro volbu oblastí ω_m však vypočteme teoretickou maximální hodnotu úhlové rychlosti rychlosti $|\omega_{m_{\max}}|$ s použitím odbuzování. Budeme-li uvažovat nulový zátěžný moment, pak v krajním případě při $u_{sq} = U_{\max}$ a $i_{sd} = -I_{\max}$ dostáváme v rovnovážném stavu přibližnou hodnotu

$$|\omega_{m_{\max}}| = \frac{U_{\max}}{P_p(\Psi_{PM} - L_d I_{\max})} = \frac{150}{3(0,3 - 6,5 \cdot 12 \cdot 10^{-3})} \doteq \underline{\underline{225 \text{ rad/s}}}. \quad (4.8)$$

S myšlenkou rovnoměrného rozdělení dosažitelné oblasti otáček se tak v našem případě bude PWA systém skládat ze tří dynamik, kdy v maticích (3.29) bude platit

$$\begin{aligned} \Omega_m &= 180 \text{ rad/s} \quad \text{pro } \omega_m > 80 \text{ rad/s}, \\ \Omega_m &= 0 \text{ rad/s} \quad \text{pro } \omega_m \in \langle -80, 80 \rangle \text{ rad/s}, \\ \Omega_m &= -180 \text{ rad/s} \quad \text{pro } \omega_m < -80 \text{ rad/s}. \end{aligned} \quad (4.9)$$

Stejně, jako v kapitole 3.2, i zde bude výsledný systém normalizován na rozsah stavů, vstupů a výstupů $\langle -1; 1 \rangle$. Bude k tomu použito norem $U_{Nrm} = 150 \text{ V}$, $I_{Nrm} = 12 \text{ A}$ a $\Omega_{Nrm} = 230 \text{ rad/s}$.

Predikční a řídicí horizont byl zvolen $N = 4$ a $N_u = 3$. Co se týká nastavení penalizačních matic (2.57), pak byla přímá a kvadrurní složka statorového proudu i_{sd} a i_{sq} penalizována koeficienty $\gamma_{i_{sd}} = 2 \cdot 10^{-3}$ a $\gamma_{i_{sq}} = 0,15$. Zatímco proud i_{sq} je penalizován stejně jako u LMPC algoritmu, tak na velikost proudu i_{sd} je brán mnohem menší zřetel tak, aby došlo pokud možno k nejvyššímu odbuzení motoru (to vyžaduje vysokou hodnotu i_{sd}). Rozdíl skutečné a žádané rychlosti a změny akčního zásahu $\vec{\delta u}(k)$ je pak penalizován stejně jako u LMPC, tedy $\gamma_\omega = 200$ a $\gamma_{u_{sd}} = \gamma_{u_{sq}} = 1 \cdot 10^{-3}$. Konečný stav byl penalizován maticí $\mathbf{P} = \mathbf{Q}$.

Co se týká omezení proudů, pak můžeme použít šestiúhelníku definovaného pomocí nerovnic (3.31). Maximální velikost přímé složky proudu je tak omezena na hodnotu $\sqrt{3}I_{\max}/2 = 10,4 \text{ A}$. Sice tak poklesne přibližná maximální rychlost na hodnotu $|\omega_{m_{\max}}| \doteq 215 \text{ rad/s}$, nicméně tím není omezen proud i_{sq} natolik, jako by tomu bylo u obdélníku.

V případě napětového omezení, je v kapitole 3.2 použito rovnoramenného šestiúhelníku. Ten bohužel výrazně omezuje maximální velikost napětí u_{sq} . Experimentálně přitom bylo zjištěno, že při maximální hodnotě odbuzovacího proudu i_{sd} nebylo využito veškeré napětí u_{sd} , jež šestiúhelníkové omezení dovoluje. To znamená, že maximální dosažená délka vektoru statorového napětí u_{sq} byla omezena více, než bylo

potřeba. Šestiúhelník napěťového omezení tak byl protažen o 10 % v obou směrech osy q . Matice (3.26) tak byla upravena do tvaru

$$\mathbf{A}_{6u} = \begin{bmatrix} -1 & -1 & 0 & 1 & 1 & 0 \\ -0,82 & 0,82 & 1,05 & 0,82 & -0,82 & -1,05 \end{bmatrix}^T. \quad (4.10)$$

Na první krok predikčního horizontu nebyla, pro zajištění řešitelnosti optimalizační úlohy, kladena žádná omezení. Všechna omezení pak byla normalizována, stejně jako PWA model systému.

Integrační složka v rychlostní referenci a vzorkovací perioda byly nastaveny stejně, jako u klasické implementace MPC v kapitole 3.2.

V případě explicitní implementace algoritmu bohužel není použití PWA systému dobře možné, neboť nárůst počtu regionů je významný. Jak bylo popsáno v kapitole 3.2, toto je možno obejít použitím (v našem případě) tří samostatných regulátorů. Každý z nich navržen pro systém s dynamikou (3.29), kde vystupuje konstanta (4.9). Mezi těmito regulátory se následně přepíná na základě hodnoty úhlové rychlosti rotoru.

Oproti LMPC má rozšířený stavový vektor LMPCFW algoritmu (3.2) vyšší dimenzi (co do počtu je zde o jeden stav více). Ukázalo se, že nárůst počtu regionů výsledného explicitního LMPCFW algoritmu je značný (každý z dílčích regulátorů obsahoval tisíce regionů) a bránil by případnému praktickému nasazení. Bylo proto rozhodnuto, že ze stavového vektoru bude vynechán kompenzační člen c_{sd} (napětí u_{sd} je však stále počítáno dle vztahu (3.19)). Následné zjednodušení napěťových omezení a snížení dimenze stavového prostoru pak výrazně redukovalo počet regionů. Pro ilustraci vlivu tohoto zanedbání uvažujme nejhorší možný případ, kdy při proudu $i_{sq} = I_{\max}$ dosáhneme maximálních otáček $|\omega_{m_{\max}}|$, tedy

$$c_{sd_{\max}} = -L_q P_p (|\omega_{m_{\max}}| - \Omega_m) I_{\max} = -6,5 \cdot 10^{-3} \cdot 3 (215 - 180) 12 = \underline{\underline{-5,85 \text{ V}}}. \quad (4.11)$$

Tato situace však nikdy nenastane, neboť v oblasti obuzování (při nichž je dosaženo maximálního rozdílu otáček ω_m a konstanty Ω_m) bude hodnota i_{sq} daleko menší, respektive nulová. Kompenzační člen c_{sd} tak bude mít v případě napěťového omezení (které jako jediné ovlivňuje) zanedbatelnou hodnotu.

Ostatní aspekty návrhu explicitního LMPCFW regulátoru (omezení, účelová funkce) jsou shodné jako u jeho implicitního protějšku.

4.4. Výsledky simulací

V této podkapitole budou představeny výsledky simulace řízení PMSM (viz tabulka 4.1), s použitím algoritmů vektorového řízení (FOC), LMPC a dvou nově navržených algoritmů LMPTC a LMPCFW. Veškeré nastavení těchto algoritmů bylo popsáno v předchozí podkapitole. U algoritmu LMPC a LMPTC je chování implicitní a explicitní verze regulátoru identické. Pouze u LMPCFW algoritmu dochází vlivem náhrady PWA systému třemi samostatnými regulátory k odchylkám v chování. Proto budou simulovány obě jeho verze.

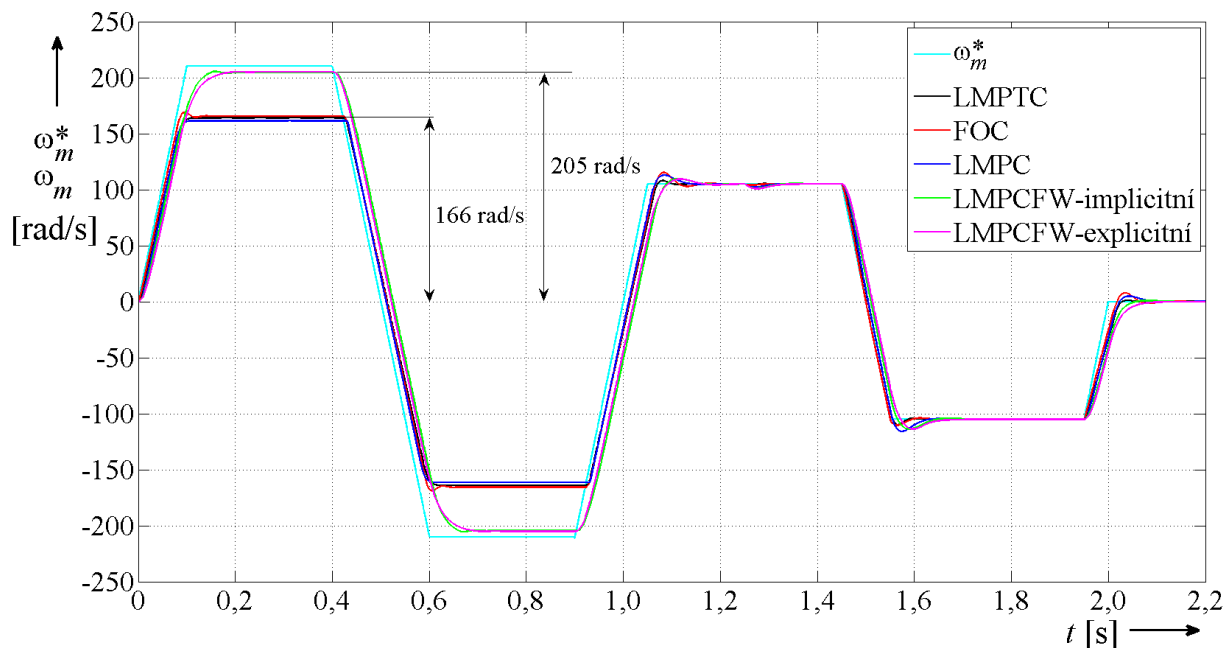
Zvolený průběh žádaných otáček ω_m^* je vidět v grafu na obrázku 4.1. V první sekundě byla žádaná hodnota nastavena tak, aby bylo testováno chování všech algoritmů jak v napěťovém, tak proudovém omezení. Napěťového omezení v časech přibližně $t \in \langle 0, 1; 0, 4 \rangle$ s a $t \in \langle 0, 6; 0, 9 \rangle$ s bylo dosaženo nastavením maximální žádané hodnoty rychlosti na hodnotu 210 rad/s, což se ukázala být hodnota, již žádný z regulátorů nemohl dosáhnout. K proudovému omezení by mělo docházet v časech $t \in \langle 0; 0, 1 \rangle$ s, $t \in \langle 0, 4; 0, 6 \rangle$ s a $t \in \langle 0, 9; 1, 05 \rangle$ s. Toho bylo dosaženo nastavením rychlosti nárůstu žádané hodnoty na $a = 2100 \text{ rad} \cdot \text{s}^{-2}$, přičemž maximálně lze dosáhnout

$$a_{\max} = \frac{|m_{\max}|}{J} = \frac{16,2}{8 \cdot 10^{-3}} = \underline{\underline{2025 \text{ rad} \cdot \text{s}^{-2}}}. \quad (4.12)$$

K omezení rychlosti nárůstu žádané hodnoty bylo přistoupeno, jelikož skoková změna žádané hodnoty způsobovala problémy s řešitelností u algoritmů založených na lineárním MPC. V praxi by bylo možno problém odstranit méně agresivním nastavením těchto regulátorů. V čase 1,25 s je pak na motor přiveden zátěžný moment $m_z = 3 \text{ Nm}$.

Srovnání průběhů úhlové rychlosti rotoru

Výsledné srovnání odezev úhlové rychlosti rotoru ω_m je na obrázku 4.1. Prvním závěrem, jež můžeme učinit, je, že oba navržené algoritmy LMPTC a LMPCFW jsou plně funkční. Algoritmus LMPCFW dosáhl hodnoty zhruba 205 rad/s, což je o téměř 40 rad/s více než u ostatních algoritmů bez odbuzování (nárůst zhruba 23,5 %). Z grafu rovněž vidíme, že všechny algoritmy bez problémů dokázaly kompenzovat poruchu v podobě zátěžného momentu.

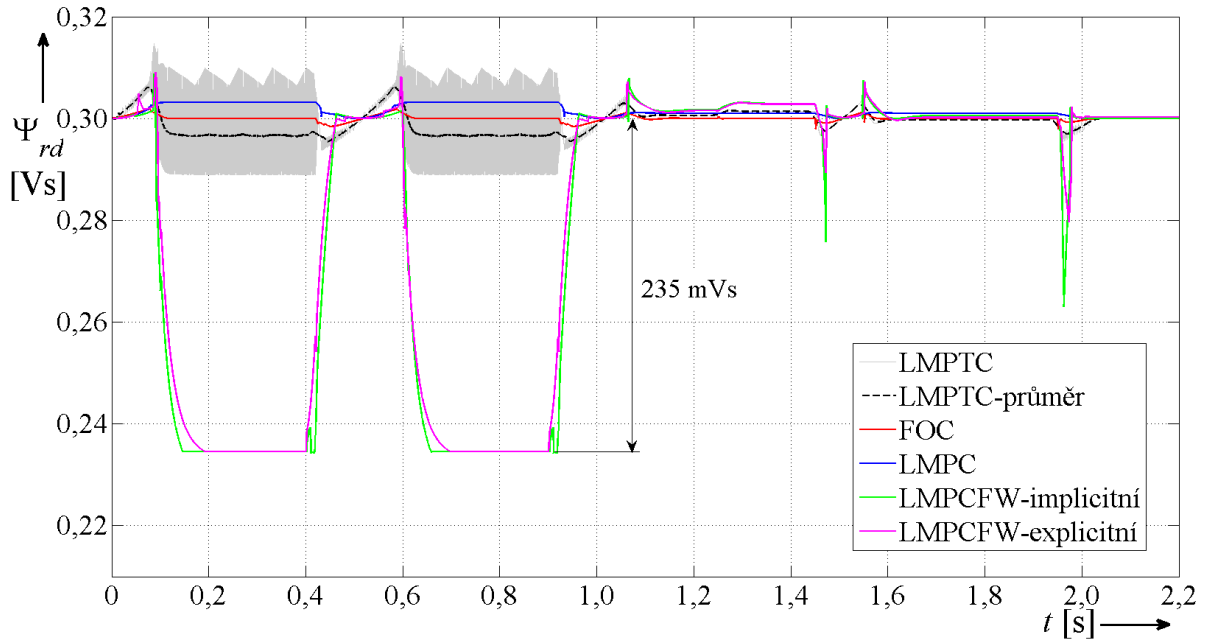


Obrázek 4.1: Srovnání průběhů časových závislostí úhlové rychlosti rotoru ω_m

Srovnání průběhů statorového magnetického toku

Na dalším obrázku 4.2, vidíme srovnání průběhů statorového magnetického toku Ψ_{sd} . Zde si můžeme povšimnout, že u LMPCFW algoritmu dochází ke snížení magnetického toku Ψ_{sd} (odbuzení) o 235 mVs v době napěťového omezení. Bohužel jsou také přítomny výraznější zákmity. To je způsobeno jednak nízkou penalizací proudu i_{sd} , působením integrační složky v rychlostní referenci, ale také přechodem mezi jednotlivými modely PWA systému (či jednotlivými regulátory u explicitní verze). V tomto případě se zřejmě jedná o daň za schopnost motor odbudit. Poměrně zajímavé je, že i přes učiněná zjednodušení jsou tyto zákmity menší u explicitní verze regulátoru.

Dalším poznatkem, jež můžeme učinit, je výrazné kmitání statorového toku u LMPTC algoritmu v době napěťového omezení. To je způsobeno řízením ve statorové α, β souřadné soustavě a aproximací napěťového a proudového omezení pomocí přímek, respektive osmiúhelníku. Rotující vektory proudu a napětí v omezení jsou nuceny tyto osmiúhelníky opisovat a nemají tak konstantní amplitudu. Bohužel, zde se jedná o vlastnost algoritmu.

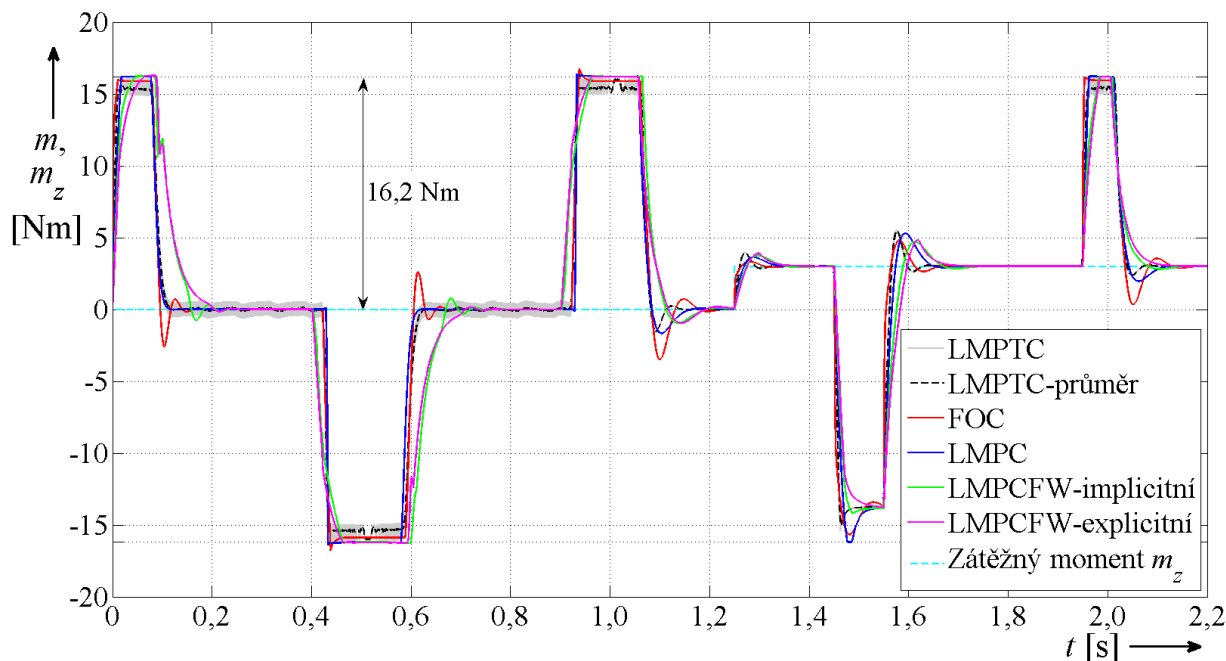


Obrázek 4.2: Srovnání průběhů statorového magnetického toku Ψ_{sd}

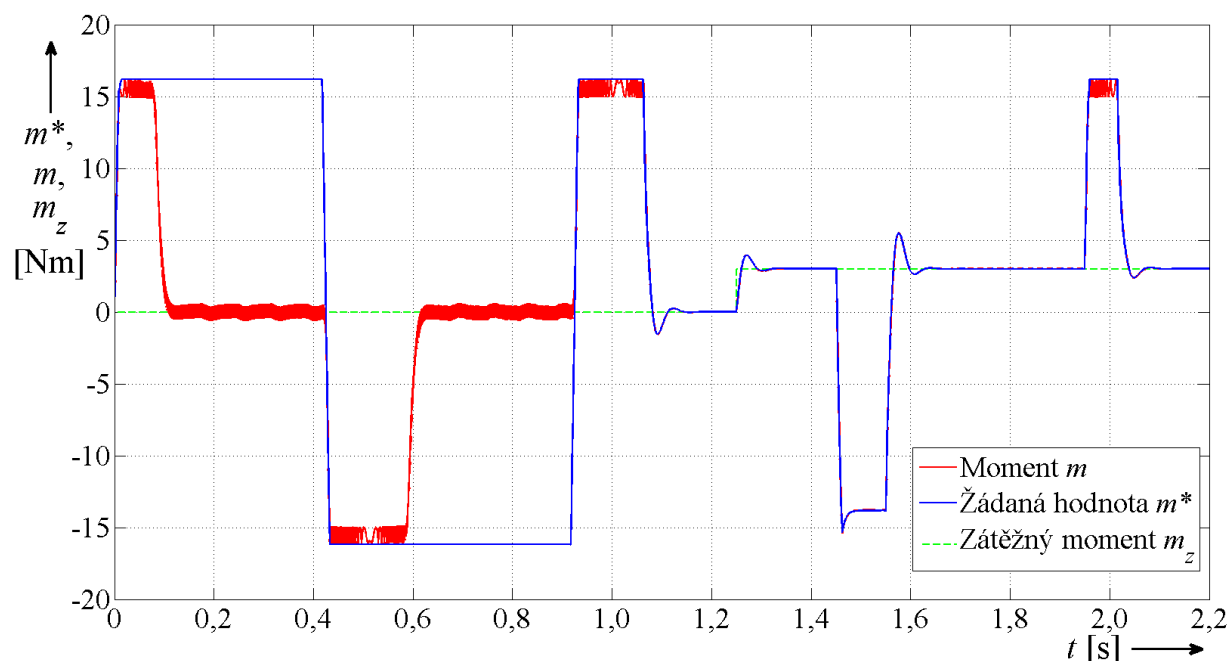
U klasické implementace LMPC algoritmu můžeme pozorovat drobné nežádoucí nabuzení. To je způsobeno nepřesností modelu, respektive použitím Eulerovy diskretizace. Při vyšších rychlostech rotoru ω_m se totiž během vzorkovací periody výrazně změní úhel natočení rotoru, a tím i rotorový magnetický tok. Regulátor není schopen tuto změnu kompenzovat, jelikož není obsažena v modelu. Výsledkem je nárůst proudu i_{sd} spolu s rostoucími otáčkami, což zvyšuje magnetický tok Ψ_{sd} . Tento závěr potvrzuje i fakt, že při nastavení periody výpočtu simulace na hodnotu T_s problém zmizí. Výsledný nárůst Ψ_{sd} je však zcela zanedbatelný. Nicméně oproti FOC, které skvěle dokáže držet hodnotu $i_{sd} = 0$ A (respektive $\Psi_{sd} = \Psi_{PM}$) se jedná o nevýhodu.

Srovnání průběhů točivého momentu

Průběh momentu vidíme na obrázku 4.3. U LMPTC algoritmu si můžeme povšimnout kmitání momentu v proudovém i napěťovém omezení. Jak již bylo vysvětleno, je to způsobeno aproximací kružnic omezení pomocí n -úhelníku. Důsledkem toho je, že maximálního momentu je dosaženo pouze špičkově a v průměru je maximální moment snížen zhruba o 1 Nm. V grafu je také dobře vidět, že LMPCFW algoritmus je schopen vytvářet moment déle než ostatní algoritmy. V důsledku odbuzování však klesá s otáčkami, neboť se již nenacházíme v oblasti konstantního momentu, ale v oblasti konstantního výkonu. To také můžeme pozorovat na obrázku 4.1, kde se v oblasti odbuzování snižuje rychlost nárůstu otáček.



Obrázek 4.3: Srovnání průběhů časových závislostí momentu m

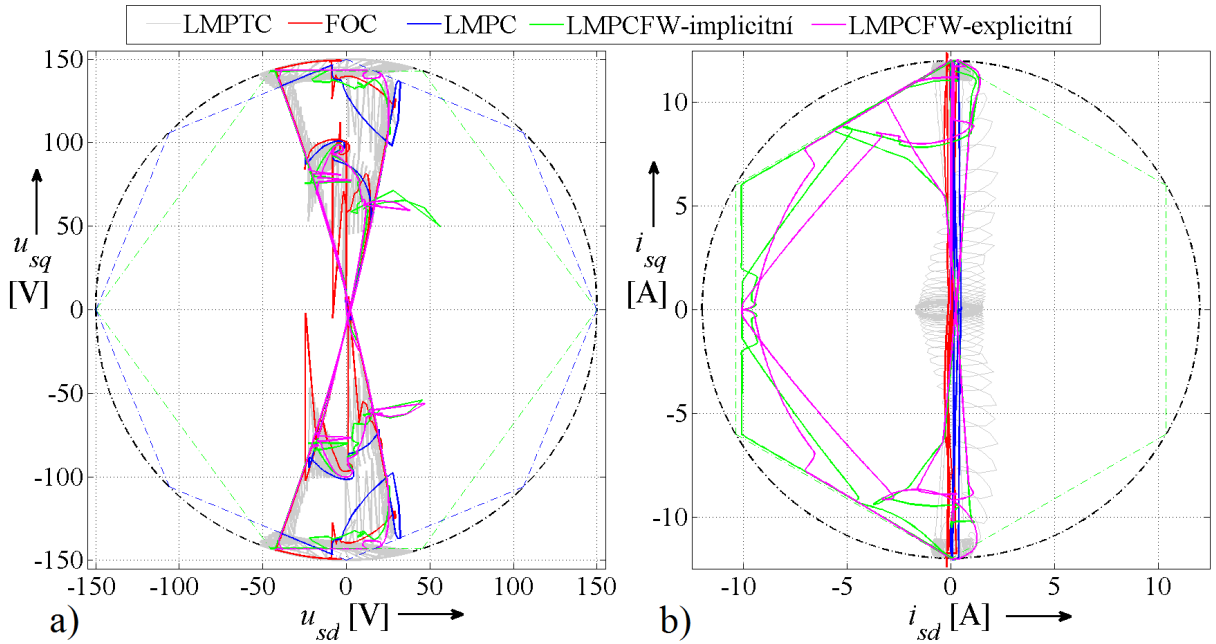


Obrázek 4.4: Průběh momentu a jeho reference u LMPTC algoritmu

Na obrázku 4.4 pak vidíme, jak je podřízený regulátor momentu u LMPTC algoritmu schopen sledovat svou referenci m^* . Během napěťového a proudového omezení dochází k již popisovanému kmitání. Pokud však k omezení nedochází, je reference - tvořená PI regulátorem - sledována s vysokou přesností. LMPTC algoritmus tak podobně jako DTC algoritmu vyniká velmi rychlou odezvou momentu.

Srovnání statorových napětí a proudů v d, q souřadné soustavě

V dalším grafu na obrázku 4.5 jsou vyneseny trajektorie vektorů a) statorového napětí a b) statorového proudu v d, q souřadné soustavě. Na obrázku si všimněme především dodržování hranic vymezených kružnicí proudového a napěťového omezení. Vidíme, že algoritmus FOC v napěťovém omezení dokázal nejlépe využít maximální délky vektoru statorového napětí (trajektorie kopíruje kružnici). To není překvapivé, neboť složky statorového napětí jsou u tohoto algoritmu omezovány přímo výpočtem z rovnice kružnice napěťového omezení (2.58). Napěťová omezení jsou však přesně dodržena všemi simulovanými algoritmy, včetně explicitní verze LMPCFW, kde byl v rovnicích napěťového omezení zanedbán kompenzační člen c_{sd} .



Obrázek 4.5: Srovnání vektorů statorových a) napětí u_{sd} a u_{sq} a b) proudů i_{sd} a i_{sq} v d, q souřadné soustavě

K mírnému překročení omezení proudů však došlo u všech algoritmů. U algoritmů založených na lineárním MPC se pravděpodobně jedná o důsledek nepřesnosti modelu, jež vzniká použitím Eulerovy diskretizace (stejný jev, jež způsobil mírné nabuzení u LMPC algoritmu na obrázku 4.2). Výsledkem je pak mírný nárůst proudu i_{sd} . Jelikož omezení proudů nebyla u tohoto algoritmu kladena na první krok predikčního horizontu, je optimalizační úloha dále řešitelná i po mírném překročení proudového omezení. U FOC je proudové omezení překročeno spíše v ose q . To lze vysvětlit tím, že v případě proudového omezení je patřičně limitována velikost reference proud i_{sq} na výstupu otáčkového PID regulátoru. Regulátor proudu

i_{sq} , však tuto referenci nemusí sledovat dostatečně přesně (v našem případě došlo k překmitu), což vede k pozorovanému porušení proudového omezení.

U LMPCFW algoritmu také můžeme pozorovat poměrně výrazný odbuzovací proud i_{sd} dosahující hodnoty až 10 A, což je mírně pod hodnotou teoretického maxima $\sqrt{3}I_{\max}/2 = 10,4$ A. Je to kvůli hodnotě koeficientu penalizace proudu i_{sd} , který nebyl volen nižší z důvodu výrazných zákrmitů proudu i_{sd} .

Z grafů lze rovněž vyčíst, že se LMPTC algoritmu poměrně dobře dařilo držet pravý úhel mezi vektorem statorového proudu a rotorového magnetického toku. Vektor statorového proudu se totiž pohyboval výhradně v okolí osy q .

Srovnání výpočetní náročnosti algoritmů založených na lineárním MPC

Výpočetní náročnost byla určena pomocí nástroje profiler v prostředí MATLAB-Simulink. Měřena byla pouze doba výpočtu jednoho volání t_{ct} samotné části lineárního MPC. Není tedy například započítána doba strávená výpočtem transformace souřadnic. Měření bylo provedeno na osobním počítači s frekvencí procesoru 2,5 GHz a s operačním systémem Windows 7. Výsledky je tedy třeba brát pouze orientačně, neboť mohly být zkresleny procesy běžícími na pozadí. V tabulce 4.2 jsou uvedeny zjištěné časy výpočtu pro implicitní verze regulátorů a v tabulce 4.3 pak pro jejich explicitní realizace. Zatímco implicitní lineární MPC byly realizovány pomocí s-funkce (tedy ve skriptovacím jazyce MATLAB), explicitní verze byly s pomocí MPT toolboxu převedeny do jazyka C. V obou tabulkách je rovněž uvedeno relativní srovnání výpočetního času s klasickou implementací LMPC regulátoru.

První, čeho si můžeme povšimnout, je řádový rozdíl ve výpočetní náročnosti implicitní a explicitní verze každého z regulátorů. To je částečně způsobeno formou realizace (vykonání programu v jazyce C v případě explicitního regulátoru je výrazně rychlejší), nicméně to dobře dokumentuje redukci výpočetní náročnosti u explicitního řízení. Rozdíly mezi implicitními MPC algoritmy se pohybují v řádech desítek procent. Dle očekávání je LMPTC algoritmus méně výpočetně náročný než klasické LMPC. U LMPCFW založeném na PWA systému je tomu naopak. Stejně poměry panují i u explicitních verzí regulátorů, pouze rozdíly jsou mnohem větší. Jenom 81 regionů u explicitního LMPTC algoritmu je velmi dobrým výsledkem a algoritmus by jistě šlo implementovat na dnešních DSC. Naopak u LMPCFW algoritmu by zřejmě bylo potřeba další optimalizace. Bez ní by bylo nutné implementovat data definující přes dva tisíce regionů.

Tabulka 4.2: Srovnání výpočetní náročnosti implicitních verzí algoritmů

Název algoritmu	t_{ct} [ms]	Nárůst oproti LMPC [%]
LMPC	3,30	0
LMPTC	2,87	-15
LPMCFW	4,11	24

Tabulka 4.3: Srovnání výpočetní náročnosti explicitních verzí algoritmů

Název algoritmu	Počet regionů	t_{ct} [μs]	Nárůst oproti LMPC [%]
LMPC	299	14,2	0
LMPTC	81	2,8	-80
LPMCFW*	781 759 781	63,2 60,8 58,9	424 408 395

*Výsledky jsou uvedeny pro dílčí regulátory v pořadí:
 $\omega_m > 80 \text{ rad/s}$ | $\omega_m \in \langle -80, 80 \rangle \text{ rad/s}$ | $\omega_m < -80 \text{ rad/s}$

Závěr

Tato práce je věnována teorii prediktivního řízení aplikovaného pro regulaci střídavých elektrických pohonů, tedy synchronního a asynchronního motoru. Obecné charakteristiky těchto algoritmů, stejně jako jejich zařazení uvnitř velkého množství řídicích algoritmů elektropohonů, jsou popsány v úvodu první kapitoly. Přestože jsou prediktivní algoritmy velmi různorodou skupinou, byly na základě funkčního principu rozděleny do čtyř skupin. Jedná se o algoritmy založené na hysterezi, trajektorii, modelu a deadbeat prediktivní algoritmy. Zbytek první kapitoly se pak věnoval popisu principu, výhodám a nevýhodám nejvýznamnějších algoritmů v každé z těchto kategorií.

V následující kapitole se pak dostalo pozornosti algoritmu lineárního prediktivního řízení založeného na modelu (lineárnímu MPC). Díky možnosti snadného nasazení pro řízení MIMO systémů a implementace omezení vstupů, stavů a výstupů systému se totiž jedná o velmi perspektivní algoritmus. U tak rychlých systémů jako jsou elektropohony je však nejdůležitější možnost explicitního řízení, která může vést k významné redukci jinak poměrně vysokých výpočetních nároků.

V druhé kapitole tak byl podrobně vysvětlen princip a základní stavební prvky lineárního MPC regulátoru. Především byl předveden převod účelové funkce, výchozího modelu systému a rovnic omezení na úlohu kvadratického programování, která je jádrem lineárního MPC regulátoru. Nakonec byla odvozena implementace lineárního MPC algoritmu pro řízení synchronního motoru s permanentními magnety (tuto implementaci nazvěme LMPC).

Třetí kapitola pak byla věnována vlastnímu přínosu práce, tedy vytvoření vlastního prediktivního algoritmu pro řízení střídavých pohonů. Nakonec se podařilo vytvořit dva algoritmy, kdy oba vycházejí z lineárního MPC, čímž si uchovaly jeho výhody. Oba algoritmy byly vytvořeny pro řízení PMSM motoru.

První z těchto algoritmů byl nazván Linear Model Predictive Torque Control (LMPTC). Tento algoritmus byl vytvořen s cílem dosáhnout co nejnižší výpočetní náročnosti (viz kapitola 3.1). Jeho blokové schéma je uvedeno na obrázku 3.1. V nadřazené smyčce se nachází PI regulátor otáček a v podřízené smyčce pak lineární MPC pro regulaci proudů, jejichž reference je získána výpočtem rovnic (3.11). Řízení probíhá v α, β souřadné soustavě, což odstraňuje potřebu Parkovy transformace a tím snižuje výpočetní nároky. Úspora je však poměrně malá, neboť součástí algoritmu je i rekonstrukce vektoru $\vec{\Psi}_{r\alpha\beta}$, s jehož pomocí lze u PMSM motoru Parkovu transformaci snadno vypočítat. Přesto byl koncept dále rozvíjen, neboť LMPTC by

se v rámci dalšího výzkumu mohlo podařit LMPTC aplikovat pro řízení asynchronního motoru, kde by již úspora byla výraznější.

Druhý algoritmus, který byl v rámci práce vytvořen, byl nazván Linear Model Predictive Control with Field Weakening (LMPCFW). Jak název napovídá, cílem bylo získání schopnosti algoritmu odbudit řízený motor, a tím dosáhnout vyšších otáček. Tato schopnost byla u LMPC ztracena díky způsobu linearizace modelu. Nakonec toho bylo docíleno použitím PWA modelu (3.23), který lze složit z několika dynamik, příslušných dané oblasti stavového prostoru (viz kapitola 3.2). Výsledná nepřesnost modelu byla kompenzována zavedením virtuálního napětí dle rovnic (3.19), (3.20) a následnou modifikací implementace napěťového omezení (viz obrázek 3.3).

Poslední, čtvrtá kapitola se již zabývala simulačním srovnáním algoritmů vektorového řízení, LMPC a nově navržených LMPTC a LMPCFW algoritmů při řízení PMSM motoru, jehož parametry se nacházejí v tabulce 4.1. U LMPCFW algoritmu byla simulována jak implicitní, tak explicitní verze regulátoru. Zatímco implicitní verzi bylo možné bez potíží postavit na PWA systému, explicitní regulátor by vedl na nereálně velké množství regionů. Explicitní verze tak byla realizována složením z několika samostatných lineárních MPC regulátorů.

Na obrázku 4.1, kde je vynesena průběh úhlové rychlosti rotoru ω_m , vidíme ověření funkčnosti obou navržených algoritmů. Zvláště pak LMPCFW algoritmu se podařilo motor odbudit a dosáhnout tak o 24 % vyšších otáček než ostatní algoritmy bez odbuzování. Z průběhu statorového magnetického toku Ψ_{sd} pak u LMPTC algoritmu můžeme pozorovat kmitání v oblasti napěťového a proudového omezení. To je bohužel nevyhnutelný důsledek řízení v α, β souřadné soustavě, kdy jsou vektory statorového proudu a napětí nuceny opisovat po částech lineární omezení. Výhodou LMPTC je potom dle obrázku 4.4 výborná odezva momentu (pokud se nenacházíme v omezení). Na obrázku 4.5 byly vyneseny trajektorie vektorů statorového proudu a napětí v d, q souřadné soustavě. Z grafu je možno vyčíst, že všechny MPC algoritmy byly schopny dodržet napěťová omezení. U proudových omezení již k mírnému porušení došlo u všech simulovaných algoritmů.

Nakonec byla srovnána výpočetní náročnost algoritmů založených na lineárním MPC (viz tabulky 4.2 a 4.3). Dle očekávání byl LMPCFW nejnáročnější. Jeho optimalizace tedy bude hlavním cílem dalšího vývoje tohoto algoritmu. Velmi dobrých výsledků (81 regionů u explicitní verze) však bylo dosaženo u LMPTC algoritmu.

V rámci dalšího výzkumu navržených algoritmů budou jejich explicitních verze implementovány na reálném hardwaru pro řízení PMSM motoru. Následně pak bude věnována pozornost vyšetření jejich robustnosti a další optimalizaci.

Literatura

- [1] ABU-RUB, H., et al. Predictive Current Control of Voltage-Source Inverters. *IEEE Transactions on Industrial Electronics*. 2004, vyd. 51, č. 3, s. 585-593. DOI: 10.1109/TIE.2004.825364.
- [2] ABDEL-RADY IBRAHIM MOHAMED, Yasser a Ehab F. EL-SAADANY. An Improved Deadbeat Current Control Scheme With a Novel Adaptive Self-Tuning Load Model for a Three-Phase PWM Voltage-Source Inverter. *IEEE Transactions on Industrial Electronics*. 2007, vyd. 54, č. 2, s. 747-759. DOI: 10.1109/TIE.2007.891767.
- [3] BELDA, Kvetoslav a David VOSMIK. Speed control of PMSM drives by generalized predictive algorithms. *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*. 2012, s. 2012-2017. DOI: 10.1109/IECON.2012.6388893.
- [4] BEMPORAD, A., et al. The explicit linear quadratic regulator for constrained systems. *Automatica*. 2002, č. 38, s. 3-20.
- [5] BOLOGNANI, S., et al. Design and Implementation of Model Predictive Control for Electrical Motor Drives. *Industrial Electronics, IEEE Transactions on*. Vyd. 56, s. 1925-1936.
- [6] BOLOGNANI, S., et al. Speed and current Model Predictive Control of an IPM synchronous motor drive. *IEEE International Electric Machines*. 2011, s. 1597-1602. DOI: 10.1109/IEMDC.2011.5994599.
- [7] CAHA, Zdeněk. *Elektrické pohony*. 1. vyd. Praha: SNTL, 1990, 359 s. ISBN 80-03-00418-7.
- [8] CAMACHO, E. a C. BORDONS. *Model predictive control*. Mnichov, 2003, 404 s. ISBN 1-85233-694-3.
- [9] CORTES, P., M.P. KAZMIERKOWSKI, R.M. KENNEL, D.E. QUEVEDO a J. RODRIGUEZ. Predictive Control in Power Electronics and Drives. *IEEE Transactions on Industrial Electronics*. 2008, vyd. 55, č. 12, s. 4312-4324. DOI: 10.1109/TIE.2008.2007480.
- [10] DEPENBROCK, M. Direct Self-Control (DSC) of inverter-fed induction machine, *IEEE Transactions on Industrial Electronics*., č. 3, s. 420-429.

- [11] GEYER, T., G. PAPAFOITOU a M. MORARI. Model Predictive Direct Torque Control—Part I: Concept, Algorithm, and Analysis. *IEEE Transactions on Industrial Electronics*. 2009, vyd. 56, č. 6, s. 1894-1905. DOI: 10.1109/TIE.2008.2007030.
- [12] GOODWIN, C. a M. SERON. *Constrained control and estimation: an optimization approach*. New York: Springer, 2005, 411 s. ISBN 18-523-3548-3.
- [13] Gurobi Optimization, Inc. GUROBI Optimizer Reference Manual. In: <Http://www.gurobi.com/documentation/5.6/reference-manual/refman.pdf> [online]. 2013 [cit. 2014-05-19].
- [14] HERCEG, M. a M. KVASNICA. Multi-Parametric Toolbox 3.0. *Proc. of the European Control Conference*. 2013, č. 1, s. 502-510. Dostupné z: <http://control.ee.ethz.ch/mpt>
- [15] HOLKAR, K. a L. WAGHMARE. An Overview of Model Predictive Control *International Journal of Control and Automation*. Vyd. 3 č. 4, prosinec, 2010
- [16] HOLTZ, J. a S. STADTFELT, A. predictive controller for the stator current vector of AC machines fed from a switched voltage source *Proc. IPEC*, Tokyo, Japan, 1983, s. 1665–1675.
- [17] LEPKA, J a P. STEKL. *3-Phase AC Induction Motor Vector Control Using a 56F8300 Device*. [online]. Freescale Semiconductor, 2005 [cit. 2014-01- 23]. Dostupné z: <http://cache.freescale.com/files/product/doc/AN1930.pdf>
- [18] LINDER, A., et al. *Model-Based Predictive Control of Electric Drives*. Goettingen, 2010, 256 s. ISBN-13 9783869553986
- [19] MathWorks, Inc. MATLAB version 7.14.0.739. In: <Http://www.mathworks.com> [online]. 2013 [cit. 2014-05-19].
- [20] MATTARELLI, P. An Improved Deadbeat Control for UPS Using Disturbance Observers. *IEEE Transactions on Industrial Electronics*. 2005, vyd. 52, č. 1. DOI: 10.1109/TIE.2004.837912.
- [21] MORALES-CAPORAL, Roberto a Mario PACAS. A Predictive Torque Control for the Synchronous Reluctance Machine Taking Into Account the Magnetic Cross Saturation. *IEEE Transactions on Industrial Electronics*. 2007, vyd. 54, č. 2, s. 1161-1167. DOI: 10.1109/TIE.2007.891783.

- [22] MUTSCHLER P. Verfahren zur direkten Regelung der Geschwindigkeit eines elektrischen Antriebs (Principle for Direct Control of the Speed of an Electric Drive), German Patent DE 196 35 981 C 2, 1996
- [23] OTTERSTEN, R. a J. SVENSSON. Vector Current Controlled Voltage Source Converter—Deadbeat Control and Saturation Strategies. *IEEE Transactions on Industrial Electronics*. 2002, vyd. 17, č. 2, s. 279-285.
- [24] PAPAFOTIOU, G., J. KLEY, K.G. PAPADOPOULOS, P. BOHREN a M. MORARI. Model Predictive Direct Torque Control—Part II: Implementation and Experimental Evaluation. *IEEE Transactions on Industrial Electronics*. 2009, vyd. 56, č. 6, s. 1906-1915. DOI: 10.1109/TIE.2008.2007032.
- [25] PETRUŠKA, L. *Simulace řídicích struktur elektromechanických systémů*. Diplomová práce. FEKT VUT v Brně, 2010, 90s.
- [26] RAO, C. a J. RAWLINGS. Linear programming and model predictive control. *Journal of Process Control*. 2000, č. 10, s. 283-289.
- [27] RANTZER, A. a M. JOHANSSON. Piecewise Linear Quadratic Optimal Control. *IEEE Transactions on Automatic Control*. 200, č. 4.
- [28] RODRIGUEZ, J. a P. CORTES. *Predictive control of power converters and electrical drives*. Chichester, 2012. ISBN 978-1119-96398-1.
- [29] STUMPER, J: *Flatness-based predictive and optimal control for electrical drives*. Mnichov, 2013, 199 s.
- [30] ŠTECHA, J. a J. PEKAŘ. Short Course on Model Predictive Control. In: [online]. 2010. Dostupné z: http://www.crr.vutbr.cz/system/files/brozura_07_1005.pdf

Seznam příloh

Příloha A: Matematický model synchronního motoru s permanentními magnety

Příloha B: Vektorové řízení synchronního motoru s permanentními magnety

Příloha C: Nastavení simulace algoritmu vektorového řízení

A. Matematický model synchronního motoru s permanentními magnety

V této kapitole bude odvozen matematický model synchronního motoru s permanentními magnety (PMSM). Ty vynikají především svou vysokou účinností a dobrým poměrem mezi objemem a výkonem motoru. Jejich nevýhodou oproti asynchronnímu motoru je především jejich cena vyplývající z použití drahých permanentních magnetů.

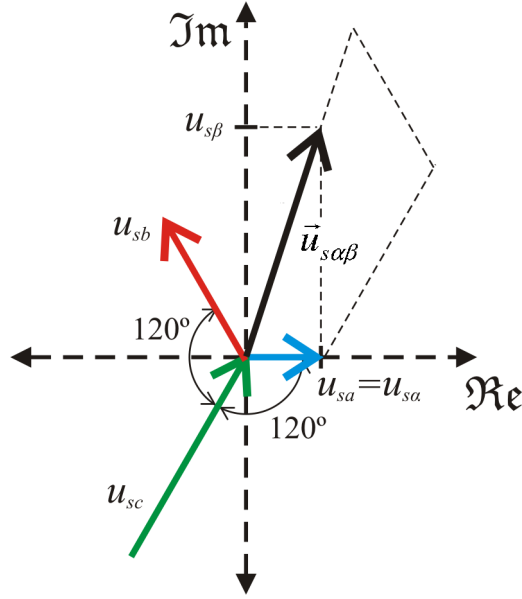
Statorové vinutí je třífázové, rozložené tak, aby při napájení třífázovým symetrickým napětím vzniklo rotující magnetické pole. To interaguje s magnetickým polem permanentních magnetů na rotoru, čímž vzniká točivý moment. Aby byl výsledný model dostatečně jednoduchý, bylo přijato několik zjednodušujících předpokladů:

- Ve smyslu rozložení vinutí statoru a rotoru a jejich odporů a indukčností je předpokládána třífázová symetrie motoru.
- Rozložení magnetického pole ve vzduchové mezeře je harmonické.
- Hodnoty indukčností a odporů vinutí je konstantní.
- Ztráty v železe jsou zanedbány.
- Motor je napájen třífázovým symetrickým napětím.

Pro odvození dostatečně jednoduchého a přesného matematického modelu PMSM motoru je vhodné vyjádřit prostorové vektory veličin motoru (například statorový proud či napětí) ve dvoufázové rotující souřadné soustavě. K tomu je nejprve potřeba za pomoci Clarkovy transformace přejít z přirozené, třífázové, stacionární soustavy a , b , c na dvoufázovou, statorovou soustavu α , β . To je znázorněno na obrázku A.1. Například pro transformaci statorových fázových napětí u_{sa} , u_{sb} a u_{sc} do α , β souřadnic pak platí

$$\vec{u}_{s\alpha\beta} = \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} = \frac{2}{3} \mathbf{C}_{trf} \begin{bmatrix} u_{sa} \\ u_{sb} \\ u_{sc} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} u_{sa} \\ u_{sb} \\ u_{sc} \end{bmatrix}, \quad (\text{A.1})$$

kde $\mathbf{C}_{trf}$ je transformační matice.



Obrázek A.1: Princip Clarkové transformace

Inverzní transformaci Clarkové pak lze provést pomocí vztahu

$$\begin{bmatrix} u_{sa} \\ u_{sb} \\ u_{sc} \end{bmatrix} = \mathbf{C}_{trf}^T \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix}. [29] \quad (\text{A.2})$$

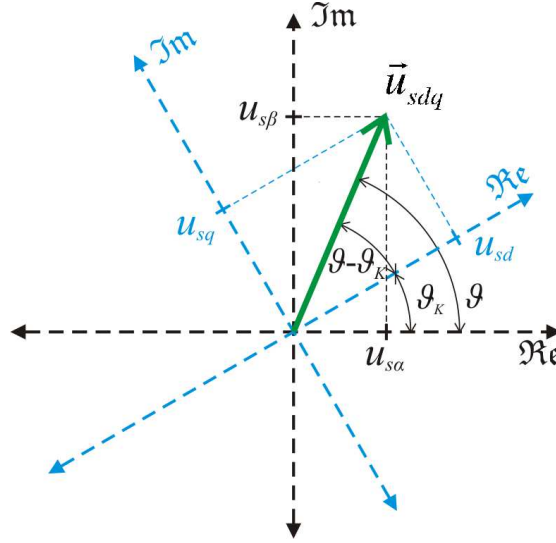
Pro převod z dvoufázové, stacionární souřadné soustavy α, β na soustavu d, q rotující úhlovou rychlostí ω_K je použito Parkovy transformace. Její princip je znázorněn na obrázku A.2. Transformace je provedena vynásobením vektoru veličiny komplexorem s jednotkovou amplitudou a požadovaným úhlem natočení ϑ_K . V praxi jsou pro výpočet d, q složek obvykle použity vztahy

$$\vec{u}_{sdq} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} = \mathbf{P}_{trf} \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} = \begin{bmatrix} \cos\vartheta_K & \sin\vartheta_K \\ -\sin\vartheta_K & \cos\vartheta_K \end{bmatrix} \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix}. \quad (\text{A.3})$$

Inverzní Parkovu transformaci pak lze provést

$$\begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} = \mathbf{P}_{trf}^T \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}. \quad (\text{A.4})$$

Úhel ϑ_K je volen tak, aby byl výsledný model co nejjednodušší, v našem případě tedy stejný jako úhel natočení rotoru ϑ_m . Tím přestanou být jednotlivé veličiny v d, q souřadné soustavě harmonickými veličinami a dojde k oddělení řízení toku a momentu. [7]



Obrázek A.2: Princip Parkovy transformace

Pro statorové napětí PMSM motoru můžeme psát vztah

$$\vec{u}_{s\alpha\beta} = R_s \vec{i}_{s\alpha\beta} + \frac{d}{dt} \vec{\Psi}_{s\alpha\beta}, \quad (\text{A.5})$$

kde R_s je odpor statorového vinutí, $\vec{i}_{s\alpha\beta}$ je vektor statorového proudu a $\vec{\Psi}_{s\alpha\beta}$ je vektor statorového magnetického toku. Jako další je nutno tuto rovnici převést do d, q souřadného systému. Složky statorového magnetického toku lze získat pomocí vztahů

$$\vec{\Psi}_{sdq} = \begin{bmatrix} \Psi_{sd} \\ \Psi_{sq} \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} \Psi_{PM} \\ 0 \end{bmatrix}, \quad (\text{A.6})$$

kde L_d a L_q jsou indukčnosti v přímém a kvadrurní ose a Ψ_{PM} je magnetický tok permanentního magnetu. Dosazením rovnice (A.6) do rovnice (A.5) pak pro statorové proudy dostáváme stavové vztahy

$$\frac{di_{sd}}{dt} = -R_s \frac{i_{sd}}{L_d} + P_p \omega_m \frac{L_q}{L_d} i_{sq} + \frac{1}{L_d} u_{sd}, \quad (\text{A.7})$$

$$\frac{di_{sq}}{dt} = -R_s \frac{i_{sq}}{L_q} - P_p \omega_m \left(\frac{L_d}{L_q} i_{sd} + \frac{\Psi_{PM}}{L_q} \right) + \frac{1}{L_q} u_{sq}, \quad (\text{A.8})$$

kde P_p je počet pólových párů. Momentová rovnice PMSM motoru je potom

$$m = \frac{3}{2} P_p (\vec{\Psi}_{sdq} \times \vec{i}_{sdq}) = \frac{3}{2} P_p \Psi_{PM} i_{sq} + \frac{3}{2} P_p (L_d - L_q) i_{sd} i_{sq}, \quad (\text{A.9})$$

kde člen $\frac{3}{2} P_p \Psi_{PM} i_{sq}$ značí elektromagnetickou složku momentu a člen $\frac{3}{2} P_p (L_d - L_q) i_{sd} i_{sq}$ značí reluktanční složku moment. Elektromagnetická složka je založena

na interakci mezi rotorovým magnetickým tokem a kvadratickou složkou statorového proudu. Reluktanční složka vzniká v důsledku snahy systému uvést se do stavu s minimálním množstvím potenciální energie. V případě PMSM motoru s povrchovými magnety (SMPMSM) platí $L_d \doteq L_q$ a reluktanční složku momentu lze zanedbat.

Jako poslední je nutno uvést mechanickou rovnici rotoru tedy

$$\frac{d\omega_m}{dt} = \frac{1}{J}(m - m_z), \quad (\text{A.10})$$

kde J je moment setrvačnosti a m_z je zátěžný model. Výsledný stavový model PMSM motoru je tvořen rovnicemi (A.7), (A.8) a (A.10). [7], [29]

B. Vektorové řízení synchronního motoru s permanentními magnety

Vektorové řízení (VOC) střídavých elektrických pohonů se začalo využívat v devadesátých letech minulého století. Jeho nasazení umožnil až příchod cenově dostupných a výkonných mikroprocesorů. V dnešní době již VOC nahrazuje často používané frekvenční řízení (také známo jako řízení V/Hz) a to z důvodu možnosti dosažení lepších dynamických vlastností výsledného pohonu. Jistou nevýhodou VOC je právě jeho vyšší výpočetní náročnost, která však v dnešní době výkonných DSC ztrácí na významu.

Významná část teorie vektorového řízení spočívá ve vyjádření modelu řízeného motoru v d, q rotující souřadné soustavě, kdy úhel natočení této soustavy je stejný jako u magnetického toku rotoru (viz příloha A). Tento úhel je pevně svázan s polohou rotoru a lze jej získat s pomocí snímače či pomocí estimátoru. [7], [17]

Z proudových rovnic (A.7) a (A.8) můžeme vyjádřit složky statorového napětí, tedy

$$u_{sd} = \left(L_d \frac{di_{sd}}{dt} + R_s i_{sd} \right) - P_p \omega_m L_q i_{sq} = u_{sdlin} + u_{sdnl}, \quad (\text{B.1})$$

$$u_{sq} = \left(L_q \frac{di_{sq}}{dt} + R_s i_{sq} \right) + P_p \omega_m (L_d i_{sd} - L_q \Psi_{PM}) = u_{sqlin} + u_{sqnl}, \quad (\text{B.2})$$

kde u_{sdlin} a u_{sqlin} jsou lineární složky a u_{sdnl} a u_{sqnl} jsou nelineární složky statorového napětí. Lineární složky tvoří akční zásah PI regulátorů proudů, jež jsou navrženy pro přenosy

$$F_{i_{sd}}(p) = \frac{I_{sd}(p)}{U_{sdlin}(p)} = \frac{1/R_s}{\tau_d p + 1}, \quad (\text{B.3})$$

$$F_{i_{sq}}(p) = \frac{I_{sq}(p)}{U_{sqlin}(p)} = \frac{1/R_s}{\tau_q p + 1}, \quad (\text{B.4})$$

kde $\tau_d = L_d/R_s$ a $\tau_q = L_q/R_s$ jsou časové konstanty, p je operátor Laplaceovy transformace a $U_{sdlin}(p), U_{sqlin}(p), I_{sd}(p)$ a $I_{sq}(p)$ jsou obrazy $u_{sdlin}, u_{sqlin}, i_{sd}$ a i_{sq} . Nelineární složky napětí lze brát jako poruchu, která bude odstraněna integrační

složkou proudových PI regulátorů. Lepší možností je však připočíst je k lineární složce přímo (takzvaný *decoupling*). Při napěťovém omezení dle rovnice

$$U_{\max} \geq \sqrt{u_{sd}^2 + u_{sq}^2}, \quad (\text{B.5})$$

kde $U_{\max}$ je maximální povolená velikost statorového napětí, se pak obvykle dává přednost d -složce napětí.

Při regulaci otáček se vychází z rovnice (A.10). Pokud zanedbáme zátěžný moment, jež můžeme brát jako poruchu a reluktanční složku momentu, která je za podmínky $L_d \doteq L_q$ (splněno u SMPMSM) či $i_{s,d} \rightarrow 0$ velmi malá, lze psát

$$\frac{d\omega_m}{dt} = \frac{3}{2} \frac{P_p}{J} \Psi_{PM} i_{sq}. \quad (\text{B.6})$$

Pro přenos mezi proudem a otáčkami motoru platí

$$F_{\omega_m}(p) = \frac{\Omega_m(p)}{I_{sq}(p)} = \frac{3}{2} \frac{P_p}{J} \Psi_{PM} = \textit{konst.}, \quad (\text{B.7})$$

kde $\Omega_m(p)$ je obraz úhlové rychlosti ω_m a $I_{sq}(p)$ je obraz statorového proudu i_{sq} . K regulaci otáček se používá kaskádní regulační struktura. Regulátor otáček, obvykle typu PID, je pak navržen pro přenos

$$F_{\omega_m}^*(p) = \frac{\Omega_m(p)}{U_{sq\text{lin}}(p)} = \frac{F_{PI i_{sq}}(p) F_{i_{sq}}(p)}{1 + F_{PI i_{sq}}(p) F_{i_{sq}}(p)} F_{\omega_m}(p), \quad (\text{B.8})$$

kde $F_{PI i_{sq}}(p)$ je přenos PI regulátoru kvadrurní složky proudu.

Přímá složka proudu je obvykle regulována na nulovou hodnotu. Jiná reference by mohla být použita v případě potřeby motor odbudít. [29]

Všechny přenosy, tedy rovnice (B.3), (B.4) a (B.8) jsou spojité a pro použití v číslicovém systému je nutno je diskretizovat.

C. Nastavení simulace algoritmu vektorového řízení

Simulace byla provedena se vzorkovací periodou $T_s = 200 \mu\text{s}$. Nebylo použito žádného algoritmu odbuzování, takže přímá složka statorového proudu i_{sd} byla regulována na nulu. Při regulaci proudů byly použity diskrétní PI regulátory s přenosem

$$F_{PI}(z) = K \left(1 + \frac{T_s}{T_I} \frac{z^{-1}}{1 - z^{-1}} \right),$$

kde K je zesílení, T_s je vzorkovací perioda, T_I je integrační časová konstanta a z je operátor z-transformace. Pro regulaci otáček bylo použito diskrétního PID regulátoru s filtrací derivační složky s přenosem

$$F_{PID}(z) = K \left(1 + \frac{T_s}{T_I} \frac{z^{-1}}{1 - z^{-1}} + N_f \frac{1 - z^{-1}}{1 - e^{\frac{-T_s}{T_D}} z^{-1}} \right),$$

kde T_D je derivační časová konstanta a N_f je filtrační koeficient. U všech regulátorů byla rovněž, jako součást omezení akčního zásahu, implementována anti-windup vazba. Její nastavení bylo provedeno skrze konstantu T_t . Veškeré nastavení regulátorů je pak uvedeno v následující tabulce A.1.

Typ regulátoru	Regulovaná veličina	K [—]	T_I [s]	T_D [s]	N_f [—]	T_t [s]
PID	ω_m	1	0,01	$1 \cdot 10^{-2}$	3	100
PI	i_{sd}	5	$8 \cdot 10^{-3}$	-	-	200
PI	i_{sq}	5	$8 \cdot 10^{-3}$	-	-	200

Tabulka C.1: Nastavení regulátorů při simulaci vektorového řízení řízení synchronního motoru s permanentními magnety