

Aplikace neuronových sítí v řízení střídavých regulovaných pohonů s asynchronním motorem

Application of neural networks in the control of induction motor drives

Jakub Bača

Disertační práce

Školitel: doc. Ing. Petr Palacký, Ph.D.

Ostrava, 2021

Poděkování

Rád bych na tomto místě poděkoval panu doc. Ing. Petru Palackému, Ph.D. a doc. Ing. Martinu Kuchařovi, Ph.D. za odborné vedení a cenné připomínky v průběhu studia. Dále děkuji Ing. Martinu Sobkovi, Ph.D. za pomoc s úpravami řídicího systému. A také kolegům a přátelům Danielu Kouřilovi a Petru Chamrádovi za pomoc a duševní podporu během společných let strávených doktorským studiem.

Abstrakt

Náplní disertační práce je problematika užití umělých neuronových sítí v oblasti řízení elektrických pohonů. Práce se konkrétně soustředí na použití neuronových sítí v systémech pro odhad stavových veličin asynchronního motoru. Byly realizovány čtyři systémy bezsensorového vektorového řízení, které využívají offline trénovanou dopřednou neuronovou síť. První řešení využívá neuronovou síť přímo k výpočtu mechanické úhlové rychlosti, rotorový tok je následně určen pomocí proudového modelu. Druhé řešení je založeno na použití pozorovatele rychlosti RF-MRAS, kde je neuronová síť použita na místě referenčního modelu a nahrazuje tak napěťový model, tím je odstraněn problém s otevřenou integrací. Největší pozornost byla věnována pozorovateli CB-MRAS. Byly navrženy dvě nové modifikace CB-MRAS s neuronovou sítí na místě proudového estimátoru. Experimentální výsledky ukazují zlepšení přesnosti a stability CB-MRAS v generátorickém režimu. Ověření bylo provedeno měřeními na experimentálním pohonu, který je vybaven 2,2 kW asynchronním motorem a řídicím systémem s digitálním signálovým kontrolérem TMS320F28335. Za účelem práce s neuronovými sítěmi byl řídicí systém rozšířen o komunikační rozhraní, které umožňuje sběr dat potřebných pro návrh a testování neuronových sítí. Pro všechny realizované metody experimentální výsledky ukazují vysokou míru přesnosti v oblasti nízkých otáček.

Hlavní přínosy disertační práce jsou následující:

- Vznik laboratorního stanoviště s asynchronním motorem pro práci s řídicími algoritmy využívajícími umělé neuronové sítě.
- Experimentální ověření čtyř metod bezsensorového vektorového řízení, které jsou založeny na použití umělé neuronové sítě.
- Zkušenosti s návrhem a trénováním umělých neuronových sítí pro dané aplikace.
- Návrh dvou modifikací pozorovatele CB-MRAS spočívajících v použití umělé neuronové sítě.

Klíčová slova

Asynchronní motor; vektorové řízení; bezsensorové řízení; estimace rychlosti; pozorovatel; umělá neuronová síť (ANN); sběr dat; adaptivní systém s referenčním modelem (MRAS); Rotor Flux MRAS (RF-MRAS); Current Based MRAS (CB-MRAS).

Abstract

This doctoral thesis deals with the use of artificial neural networks in the field of control of electric drives. In particular, the thesis focuses on the application of neural networks in systems intended for estimation of the state variables of an induction motor. Four sensorless vector control schemes have been implemented, in which an offline-trained feedforward neural network is utilized. The first solution uses a neural network which directly provides the estimated mechanical angular speed, the rotor flux is determined using the current model. The second solution is based on the use of the RF-MRAS speed observer, in this case, a neural network is used in the place of the reference model, it replaces the voltage model, thereby the problem of pure integration is eliminated. The main focus was on the CB-MRAS observer. Two new modifications of CB-MRAS with a neural network in the place of the current estimator have been proposed. The experimental results show an improvement in the accuracy and stability of CB-MRAS in the regenerating mode. The verification was performed employing an experimental drive equipped with a 2.2 kW induction motor and controlled by a control system which is based on the TMS320F28335 digital signal controller. In order to work with neural networks, the control system has been extended with a communication interface that allows the collection of data needed for designing and testing neural networks. For all implemented methods, the obtained results show a high level of accuracy in the low speed range.

The main contributions of the dissertation are as follows:

- Creation of a laboratory workplace with an induction motor for working with control algorithms based on artificial neural networks.
- Experimental verification of four methods of sensorless vector control, which are based on the use of an artificial neural network.
- Experience with the design and training of artificial neural networks for given applications.
- Design of two modifications of the CB-MRAS observer consisting in the use of an artificial neural network.

Key Words

Induction motor; field-oriented control; sensorless control; speed estimation; observer; Artificial Neural Network (ANN); data acquisition; Model Reference Adaptive System (MRAS); Rotor Flux MRAS (RF-MRAS); Current Based MRAS (CB-MRAS).

Obsah

Obsah.....	5
Seznam použitých symbolů a zkratek.....	7
Seznam obrázků	10
Seznam tabulek	13
Úvod	14
Cíle disertační práce	16
1 Neuronové sítě	17
1.1 Biologické neuronové sítě	18
1.2 Model umělého neuronu	19
1.2.1 Aktivační funkce	21
1.3 Druhy neuronových sítí	23
1.4 Učení neuronových sítí.....	27
1.4.1 Trénovací algoritmy.....	28
1.4.2 Metoda backpropagation.....	28
1.5 Aplikace umělých neuronových sítí.....	32
2 Bezsenzorové řízení – pozorovatelé MRAS	33
2.1 Vektorové řízení	33
2.2 Metody bezsenzorového řízení	36
2.3 Rotor Flux MRAS (RF-MRAS)	38
2.4 Current Based MRAS (CB-MRAS).....	40
2.5 Určení statorového napětí	42
3 Řešené aplikace neuronových sítí	43
3.1 Neuronová síť jako estimátor rychlosti (ANN-EST)	43
3.2 Náhrada napěťového modelu v pozorovateli RF-MRAS (ANNRF-MRAS).....	46
3.3 Náhrada proudového estimátoru v pozorovateli CB-MRAS (ANNCB-MRAS $\alpha\beta$)	48
3.4 CB-MRAS s výpočtem proudu v orientovaných souřadnicích (ANNCB-MRAS xy).....	51
4 Výsledky simulací.....	55
4.1 Estimátor ANN-EST	55
4.2 Pozorovatel ANNRF-MRAS	57
4.3 Pozorovatel ANNCB-MRAS $\alpha\beta$	58
4.4 Pozorovatel ANNCB-MRAS xy.....	60
5 Laboratorní stanoviště.....	63

6	Experimentální výsledky.....	66
6.1	Nastavení experimentu	66
6.1.1	Měření trénovacích dat	67
6.2	Estimátor ANN-EST	68
6.3	Pozorovatel ANNRF-MRAS	72
6.4	Pozorovatel ANNCB-MRAS $\alpha\beta$	75
6.5	Pozorovatel ANNCB-MRAS xy	82
7	Shrnutí experimentálních výsledků	87
	Závěr	88
	Literatura	89
	Publikační činnost.....	93
	Řešené projekty.....	95

Seznam použitých symbolů a zkratek

Vektory (fázory) jsou značeny tučně. Statorové veličiny jsou odlišeny dolním indexem 1, rotorové indexem 2. V souvislosti s různými vztažnými soustavami je k odlišení vektorů používán horní index. Horní index S značí vektor vyjádřený souřadnicemi ve statorovém souřadném systému (SSS), R v rotorovém souřadném systému (RSS) a O v orientovaném souřadném systému (OSS). Souřadnice vektoru jsou odlišeny dolními indexy $\alpha\beta, dq, xy$. Např. Ψ_2^S je vektor rotorového magnetického toku vyjádřený souřadnicemi ve statorové souřadné soustavě, $\Psi_{2\alpha}$ je jeho reálná složka a $\Psi_{2\beta}$ jeho imaginární složka.

Veličiny trojfázové soustavy jsou značeny dolními indexy abc .

Jmenovité hodnoty veličin stroje jsou značeny dolním indexem n .

Symbol	Jednotka	Popis
*		Žádaná hodnota veličiny (např. i_{1y}^*)
$\hat{}$		Estimovaná hodnota veličiny (např. \hat{i}_{1y})
$ $		Modul vektoru (např. $ \Psi_2 $)
Ψ	Wb	Spřažený magnetický tok
Ψ_2	Wb	Vektor rotorového magnetického toku
Ψ_2^S, Ψ_2^O	Wb	Vektor rotorového magnetického toku vyjádřený v SSS, OSS
$\Psi_{2\alpha}, \Psi_{2\beta}$	Wb	Reálná a imaginární složka vektoru rotorového toku v SSS
Ψ_{2x}, Ψ_{2y}	Wb	Reálná a imaginární složka vektoru rotorového toku v OSS
ε		Odchylka v MRAS, která je minimalizována adaptačním algoritmem
γ	rad	Poloha vektoru rotorového toku v SSS
θ	rad	Poloha rotoru v SSS
$\sigma, \sigma_s, \sigma_r$	–	Činitel rozptylu celkový, statoru, rotoru
ω	$\text{rad}\cdot\text{s}^{-1}$	Úhlová rychlost
ω_1	$\text{rad}\cdot\text{s}^{-1}$	Úhlová rychlost vektoru statorového proudu v SSS
ω_2	$\text{rad}\cdot\text{s}^{-1}$	Skluzová rychlost ($\omega_2 = \omega_1 - \omega_m$)
ω_{im}	$\text{rad}\cdot\text{s}^{-1}$	Úhlová rychlost vektoru rotorového toku v SSS
ω_m	$\text{rad}\cdot\text{s}^{-1}$	Úhlová rychlost rotoru v SSS, mechanická rychlost
ζ	rad	Poloha vektoru statorového proudu v SSS
b		Práh excitace (bias) neuronu
b_i		Práh excitace (bias) i -tého neuronu ve vrstvě neuronové sítě
E		Chybová funkce neuronové sítě
e		Chyba estimace
e		Přirozená exponenciální funkce
f_{vz}	Hz	Vzorkovací frekvence
G		Přenosová funkce
i_{1a}, i_{1b}, i_{1c}	A	Fázové proudy statoru
\mathbf{i}_1	A	Vektor statorového proudu
$\mathbf{i}_1^S, \mathbf{i}_1^R, \mathbf{i}_1^O$	A	Vektor statorového proudu vyjádřený v SSS, RSS, OSS

Seznam použitých symbolů a zkratek

$i_{1\alpha}, i_{1\beta}$	A	Reálná a imaginární složka vektoru statorového proudu v SSS
i_{1x}, i_{1y}	A	Reálná a imaginární složka vektoru statorového proudu v OSS
i_m	A	Vektor magnetizačního proudu ($i_m = \frac{\Psi_2}{L_h}$)
i_m^R	A	Vektor magnetizačního proudu vyjádřený v RSS
i_m	A	Velikost magnetizačního proudu ($i_m = \frac{ \Psi_2 }{L_h}$)
J	kg·m ²	Moment setrvačnosti
j	–	Imaginární jednotka
K_1, K_2, K_3	S, H ⁻¹ , S	Koeficienty mat. modelu proudového estimátoru
K_i	s ⁻¹	Integrační složka PI regulátoru v paralelním tvaru
K_P		Proporcionální složka PI regulátoru v paralelním tvaru
K_R		Zesílení PI regulátoru
k		Nezávislá proměnná při popisu diskrétního systému
L	H	Indukčnost
L_s, L_r	H	Statorová a rotorová indukčnost, parametry náhradního schématu AM
$L_{s\sigma}, L_{r\sigma}$	H	Rozptylová statorová a rotorová indukčnost, parametry náhr. sch. AM
L_h	H	Hlavní indukčnost, parametr náhradního schématu AM
M, m	Nm	Mechanický moment
M_n	Nm	Jmenovitý moment zátěže stroje
m_z	Nm	Moment zátěže
n	min ⁻¹	Otáčky
p	–	Počet pólů stroje
R	Ω	Elektrický odpor
R_s, R_r	Ω	Statorový a rotorový odpor, parametry náhradního schématu AM
R_{Fe}	Ω	Odpor v železe, parametr náhradního schématu AM
S		Aktivační potenciál neuronu (výsledek součtu vstupů neuronu)
s		Operátor Laplaceovy transformace
T_i	s	Časová konstanta mat. modelu proudového estimátoru
T_{vz}	s	Vzorkovací perioda
T_s	s	Statorová časová konstanta
T_r	s	Rotorová časová konstanta
T_R	s	Časová konstanta PI regulátoru
t	s	Čas
t_{DT}	s	Ochranná doba
u_1	V	Vektor statorového napětí
u_1^S, u_1^O	V	Vektor statorového napětí vyjádřený v SSS, OSS
$u_{1\alpha}, u_{1\beta}$	V	Reálná a imaginární složka vektoru statorového napětí v SSS
u_{1x}, u_{1y}	V	Reálná a imaginární složka vektoru statorového napětí v OSS
u_d	V	Napětí ve stejnosměrném meziobvodu nepřímého měniče kmitočtu
w		Váhový koeficient spojení mezi neurony
w_{ij}		Váhový koeficient spojení i -tého neuronu s j -tým neuronem předchozí vrstvy

Zkratka	Význam
$\alpha\beta$	Souřadnice ve statorovém souřadném systému (SSS)
AM	Asynchronní motor
ANN	Umělá neuronová síť (<i>Artificial Neural Network</i>)
ANNCB-MRAS	Pozorovatel CB-MRAS s ANN na místě proudového estimátoru
ANNCB-MRAS $\alpha\beta$	Pozorovatel ANNCB-MRAS s estimací proudu v SSS
ANNCB-MRAS xy	Pozorovatel ANNCB-MRAS s estimací proudu v OSS
ANN-EST	Umělá neuronová síť jako estimátor otáček
ANNRF-MRAS	Pozorovatel RF-MRAS s ANN na místě referenčního modelu
BVOV	Blok vyhodnocení orientujících veličin
BVPR	Blok vyhodnocení polohy rotoru
BZV	Blok zrušení vazby
CB-MRAS	Current-Based MRAS
DSC	Digitální signálový kontrolér (<i>Digital Signal Controller</i>)
DSP	Digitální signálový procesor (<i>Digital Signal Processor</i>)
dq	Souřadnice v rotorovém souřadném systému (RSS)
DT	Ochranná doba (<i>Dead Time</i>), prodleva mezi vypnutím jednoho a zapnutím druhého tranzistoru ve větvi střídače
ePWM	Periferie pro generování PWM (Enhanced Pulse Width Modulator)
eQEP	Periferie pro vyhodnocení výstupu inkrementálního snímače (Enhanced Quadrature Encoder Pulse)
IRC	Inkrementální snímač otáček
MAE	Střední absolutní odchylka (<i>Mean Absolute Error</i>)
ME	Střední odchylka (<i>Mean Error</i>)
MSE	Střední kvadratická odchylka (<i>Mean Squared Error</i>)
MK	Měnič kmitočtu
MRAS	Adaptivní systém s referenčním modelem (<i>Model Reference Adaptive System</i>)
OSS	Orientovaný souřadný systém (vůči SSS rotující rychlostí vektoru rotorového toku)
PWM	Pulzně šířková modulace (<i>Pulse Width Modulation</i>)
RF-MRAS	Rotor Flux MRAS
RMSE	Odmocnina střední kvadratické odchylky (<i>Root Mean Squared Error</i>)
RSS	Rotorový souřadný systém (vůči SSS rotující rychlostí rotoru)
ŘS	Řídicí systém
SSS	Statorový (stacionární) souřadný systém
TR	Transformátor
VŘ	Vektorové řízení
xy	Souřadnice v orientovaném souřadném systému (OSS)

Seznam obrázků

Obr. 1.1 Blokové schéma představující biologickou nervovou buňku [1].....	19
Obr. 1.2 Statický model umělého neuronu [1].....	20
Obr. 1.3 Podoba aktivačních funkcí (b je bias): a) sigmoida, b) radiální funkce, c) skoková funkce [1]	23
Obr. 1.4 Tansigmoidální funkce ($a = 2$).....	23
Obr. 1.5 Přímá dvouvrstvá neuronová síť 3-2-2 [8].....	25
Obr. 1.6 Struktura neuronové sítě estimátoru rychlosti ASM (8-22-1) [8]	25
Obr. 1.7 Rekurentní dvouvrstvá neuronová síť 3-2-2 [8]	26
Obr. 1.8 Struktura neuronové sítě pro adaptaci hodnoty statorového odporu R_s [34].....	26
Obr. 2.1 Souřadné soustavy pro matematický popis asynchronního motoru	33
Obr. 2.2 Blokové schéma vektorového řízení.....	34
Obr. 2.3 Obecná struktura pozorovatele MRAS.....	37
Obr. 2.4 Struktura pozorovatele RF-MRAS.....	38
Obr. 2.5 Struktura pozorovatele CB-MRAS	40
Obr. 3.1 Použitá topologie neuronové sítě estimátoru rychlosti ANN-EST (8-22-1).....	45
Obr. 3.2 Zjednodušený blokový diagram systému s estimátorem ANN-EST.....	45
Obr. 3.3 Struktura pozorovatele ANNRF-MRAS	47
Obr. 3.4 Použitá topologie neuronové sítě estimátoru rotorového toku (8-12-2)	47
Obr. 3.5 Zjednodušený blokový diagram systému s pozorovatelem ANNRF-MRAS.....	47
Obr. 3.6 Struktura pozorovatele ANNCB-MRAS s výpočtem proudu v SSS.....	50
Obr. 3.7 Použitá topologie neuronové sítě proudového estimátoru ANNCB-MRAS $\alpha\beta$ (10-24-2).....	50
Obr. 3.8 Zjednodušený blokový diagram systému s pozorovatelem ANNCB-MRAS $\alpha\beta$	50
Obr. 3.9 Vektorový diagram CB-MRAS.....	52
Obr. 3.10 Struktura pozorovatele ANNCB-MRAS s výpočtem proudu v OSS.....	54
Obr. 3.11 Použitá topologie neuronové sítě proudového estimátoru ANNCB-MRAS xy (4-4-1)	54
Obr. 3.12 Zjednodušený blokový diagram systému s pozorovatelem ANNCB-MRAS xy	54
Obr. 4.1 Simulace ANN-EST: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 200 a $\pm 50 \text{ min}^{-1}$, bez zátěže	55
Obr. 4.2 Simulace ANN-EST: měřená a estimovaná rychlost motoru, zatížení pro 200 min^{-1} , $20 \% M_n$	56
Obr. 4.3 Simulace ANN-EST: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , $20 \% M_n$	56
Obr. 4.4 Simulace ANNRF-MRAS: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 200 a $\pm 50 \text{ min}^{-1}$, bez zátěže	57
Obr. 4.5 Simulace ANNRF-MRAS: měřená a estimovaná rychlost motoru, zatížení pro 200 min^{-1} , $20 \% M_n$	57
Obr. 4.6 Simulace ANNRF-MRAS: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , $20 \% M_n$	58
Obr. 4.7 Simulace ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 300 a $\pm 50 \text{ min}^{-1}$, bez zátěže	59
Obr. 4.8 Simulace ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, zatížení pro 300 min^{-1} , $20 \% M_n$	59

Obr. 4.9 Simulace ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , $20 \% M_n$	60
Obr. 4.10 Simulace ANNCB-MRAS xy: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 300 a $\pm 50 \text{ min}^{-1}$, bez zátěže	61
Obr. 4.11 Simulace ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, zatížení pro 300 min^{-1} , $20 \% M_n$	61
Obr. 4.12 Simulace ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , $20 \% M_n$	62
Obr. 5.1 Blokové schéma experimentálního pohonu.....	64
Obr. 5.2 Fotografie experimentálního pohonu	64
Obr. 5.3 Uživatelské rozhraní v LabVIEW	65
Obr. 6.1 Trénovací data, ANN-EST.....	69
Obr. 6.2 Porovnání chybových ukazatelů pro různé rychlosti, ANN-EST	69
Obr. 6.3 ANN-EST: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 200 \text{ min}^{-1}$, ..	70
Obr. 6.4 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 200 \text{ min}^{-1}$, bez zátěže	70
Obr. 6.5 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 200 \text{ min}^{-1}$, bez zátěže.....	70
Obr. 6.6 ANN-EST: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže.....	71
Obr. 6.7 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže	71
Obr. 6.8 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže.....	71
Obr. 6.9 Trénovací data	72
Obr. 6.10 RF-MRAS: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti, $0, \pm 25, \pm 50, \pm 200 \text{ min}^{-1}$, bez zátěže	73
Obr. 6.11 RF-MRAS: porovnání rotorových toků v SSS pro skoky žádané rychlosti, bez zátěže	73
Obr. 6.12 RF-MRAS: průběh statorového proudu v OSS pro skoky žádané rychlosti, bez zátěže	73
Obr. 6.13 ANN-RFMRAS: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti, $0, \pm 25, \pm 50, \pm 200 \text{ min}^{-1}$, bez zátěže	74
Obr. 6.14 ANN-RFMRAS: porovnání rotorových toků v SSS pro skoky žádané rychlosti, bez zátěže....	74
Obr. 6.15 ANN-RFMRAS: průběh statorového proudu v OSS pro skoky žádané rychlosti, bez zátěže .	74
Obr. 6.16 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže	76
Obr. 6.17 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže	76
Obr. 6.18 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže.....	76
Obr. 6.19 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže	77
Obr. 6.20 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže	77
Obr. 6.21 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže.....	77
Obr. 6.22 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 300 min^{-1} , $20 \% M_n$	78
Obr. 6.23 Porovnání rotorových toků v SSS, přechod do gen. režimu, 300 min^{-1} , $20 \% M_n$	78
Obr. 6.24 Průběh statorového proudu v OSS, přechod do gen. režimu, 300 min^{-1} , $20 \% M_n$	78

Obr. 6.25 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 50 min ⁻¹ , 20 % M _n	79
Obr. 6.26 Porovnání rotorových toků v SSS, přechod do gen. režimu, 50 min ⁻¹ , 20 % M _n	79
Obr. 6.27 Průběh statorového proudu v OSS, přechod do gen. režimu, 50 min ⁻¹ , 20 % M _n	79
Obr. 6.28 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 300 min ⁻¹ , 40 % M _n	80
Obr. 6.29 ANNCB-MRAS $\alpha\beta$: průběh měřeného a estimovaného proudu v SSS, přechod do gen. režimu, 50 min ⁻¹ , 20 % M _n	80
Obr. 6.30 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru při 20 min ⁻¹ , 40 % M _n	81
Obr. 6.31 Porovnání rotorových toků v SSS při 20 min ⁻¹ , 40 % M _n	81
Obr. 6.32 Průběh statorového proudu v OSS při 20 min ⁻¹ , 40 % M _n	81
Obr. 6.33 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 300 min ⁻¹ , bez zátěže	83
Obr. 6.34 Porovnání rotorových toků v SSS pro skoky žádané rychlosti ± 300 min ⁻¹ , bez zátěže	83
Obr. 6.35 Průběh statorového proudu v OSS pro skoky žádané rychlosti ± 300 min ⁻¹ , bez zátěže.....	83
Obr. 6.36 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 50 min ⁻¹ , bez zátěže	84
Obr. 6.37 Porovnání rotorových toků v SSS pro skoky žádané rychlosti ± 50 min ⁻¹ , bez zátěže	84
Obr. 6.38 Průběh statorového proudu v OSS pro skoky žádané rychlosti ± 50 min ⁻¹ , bez zátěže.....	84
Obr. 6.39 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 300 min ⁻¹ , 40 % M _n	85
Obr. 6.40 Porovnání rotorových toků v SSS, přechod do gen. režimu, 300 min ⁻¹ , 40 % M _n	85
Obr. 6.41 Průběh statorového proudu v OSS, přechod do gen. režimu, 300 min ⁻¹ , 40 % M _n	85
Obr. 6.42 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 50 min ⁻¹ , 20 % M _n	86
Obr. 6.43 Porovnání rotorových toků v SSS, přechod do gen. režimu, 50 min ⁻¹ , 20 % M _n	86
Obr. 6.44 Průběh statorového proudu v OSS, přechod do gen. režimu, 50 min ⁻¹ , 20 % M _n	86

Seznam tabulek

Tab. 5.1 Jmenovité parametry asynchronního motoru.....	65
Tab. 5.2 Parametry laboratorního stanoviště	65
Tab. 6.1 Nastavení použité při měření	66
Tab. 6.2 Proměnné přenášené z řídicího systému	67
Tab. 6.3 Chybové ukazatele pro různé rychlosti, ANN-EST	69

Úvod

Součástí každého odvětví průmyslu i každodenního života jsou elektrické pohony, které tvoří ústřední prvek přeměny mezi elektrickou a mechanickou energií. V současnosti se ve strukturách regulovaných pohonů v průmyslu a dopravě nejvíce uplatňují střídavé stroje (převážně asynchronní), které jsou v této oblasti následovníky dříve používaných strojů stejnosměrných. Střídavé stroje jsou konstrukčně jednodušší, odolnější, spolehlivější a nenáročné na údržbu, mají také menší moment setrvačnosti a větší proudovou přetížitelnost. Díky absenci komutátoru jsou bez problémů použitelné i v agresivních nebo nebezpečných prostředích (např. prašných, výbušných).

Jednoduchost a robustnost konstrukce je však vykoupena složitějším způsobem řízení s nutností spojení s frekvenčními měniči v případě regulace pohonu za účelem dosažení vysoké dynamiky a efektivního provozu. Z tohoto důvodu bylo širší používání střídavých strojů umožněno až s rozvojem číslicové výpočetní techniky a výkonové elektroniky [I]. Vývoj v těchto dvou oblastech umožnil nasazení moderních algoritmů regulace střídavých elektrických pohonů, které dříve nebyly prakticky realizovatelné. Střídavé pohony se tak rozšířily i v oblasti regulovaných pohonů pro dynamicky náročné aplikace, které byly dříve doménou právě stejnosměrných pohonů.

V současnosti existují tři základní způsoby řízení střídavých strojů. Nejstarší známou metodou je *skalární řízení*, které se používá pouze pro dynamicky nenáročné pohony, které pracují především v ustáleném stavu. Dalším způsobem je *vektorové řízení*, které se vyznačuje velmi vysokou dynamikou a přesností veličin v ustálených stavech i během přechodných dějů. Poslední, nejmladší je *přímé řízení momentu*, tato metoda se rovněž vyznačuje vysokou dynamikou veličin [II][III]. Cílem vektorového řízení i přímého řízení momentu je dosáhnout toho, aby měl střídavý pohon vlastnosti srovnatelné se stejnosměrnými pohony. Základní myšlenkou těchto dvou metod je proto umožnění nezávislé regulace momentu a buzení (magnetického toku) motoru.

Realizace vektorového řízení spočívá v rozdělení vektoru statorového proudu na dvě vzájemně kolmé složky, na proud momentotvorný a proud tokotvorný, které je možné odděleně regulovat. Momentotvorná složka proudu umožňuje řídit točivý moment motoru a tokotvorná složka ovlivňuje magnetizaci stroje. Přímé řízení momentu je založeno na tom, že se reguluje přímo moment ve zvoleném tolerančním pásmu a pohyb koncového bodu vektoru statorového magnetického toku po zadané křivce. Moment je dán rychlostí rotace vektoru magnetického toku, magnetizace stroje velikostí tohoto vektoru.

Dalším stupněm vývoje těchto metod je tzv. bezsenzorové řízení, kdy není nutné použít snímač mechanické úhlové rychlosti (polohy), který je nejcitlivější částí pohonu. Absence snímače polohy odstraňuje problémy s mechanickým provedením motoru, snižuje cenu a vede k vyšší spolehlivosti a robustnosti pohonu. Odhad (estimaci) polohy či mechanické úhlové rychlosti rotoru lze určit ze změřených napětí a proudů motoru s použitím výpočetních systémů využívajících např. MRAS (*Model Reference Adaptive System*), Kalmanův filtr, Luenbergerova pozorovatele, časově-frekvenční analýzu nebo algoritmy z oblasti soft computingu, do které náleží fuzzy logika, genetické algoritmy a neuronové sítě.

Výpočetní metody využívající prvky umělé inteligence nalézají v oblasti elektrických pohonů uplatnění při řešení úloh, mezi které patří hlavně: řízení pohonů, modelování systémů, identifikace parametrů

systemu, pozorování zpětnovazebních veličin a diagnostika stavů zařízení. Výhodou těchto metod je, že nevyžadují znalost matematického modelu systému a také mohou být odolnější vůči chybám měření a změnám parametrů systému. Tyto metody často vycházejí z poznání procesů probíhajících u biologických systémů, které se snaží napodobit. Schopnosti živých organismů v oblasti řízení systémů mnohdy výrazně převyšují schopnosti technických systémů. Adaptace na měnící se vnější podmínky a zpracování sensorických signálů zatížených chybami a šumem, to vše bez použití explicitního modelu řízené soustavy, se staly oprávněně objektem podrobného výzkumu.

Umělé neuronové sítě jsou založeny na poznacích z neurofyzologie. Nejde o přesnou simulaci fungování biologické neuronové sítě, ale o identifikaci a použití principů, podle kterých biologické neuronové sítě fungují. Proto umělé neuronové sítě představují hrubý model biologických neuronových sítí. Umělé neuronové sítě dovolují modelování neznámé soustavy bez znalosti jejího matematického modelu. Schopnosti adaptace a generalizace umožňují vytvořit tento model na základě konečného počtu naměřených hodnot (trénovacích vzorů) pomocí vhodného trénovacího algoritmu, který postupnou změnou konfigurace parametrů sítě zajistí konvergenci neuronového modelu k modelu zvolené soustavy. Adaptace sítě může pokračovat i za běhu (online) a tím průběžně přizpůsobovat neuronovou síť nebo její část změnám v soustavě.

Tato práce se zaměřuje na problematiku použití umělých neuronových sítí v oblasti bezsensorového řízení asynchronního motoru – tzn. bez snímače otáček. První část práce se zabývá teorií umělých neuronových sítí. Je zde nastíněna myšlenková cesta, která vedla od biologické předlohy ke vzniku teorie umělých neuronových sítí. Dále byla věnována pozornost principu umělých neuronových sítí; jejich druhům, rozdělení, způsobům trénování. Podrobněji jsou rozebrány vícevrstvé dopředné neuronové sítě a základní algoritmus jejich adaptace. Dále jsou popsány metody RF-MRAS a CB-MRAS, protože jedním z cílů je prozkoumat možnosti zlepšení vlastností těchto pozorovatelů pomocí neuronových sítí. Další části práce se už zaměřují na konkrétní řešené aplikace neuronových sítí.

Cíle disertační práce

Disertační práce se zabývá realizací metod využívajících umělé neuronové sítě pro pozorování veličin asynchronního motoru bez použití snímače otáček. Hlavním cílem byla aplikace umělých neuronových sítí za účelem vylepšení vlastností pozorovatelů typu MRAS, zvláště pak pozorovatele CB-MRAS. Byly stanoveny tyto dílčí cíle:

- Seznámení se s principem umělých neuronových sítí a možnostmi jejich použití v oblasti elektrických pohonů.
- Návrh a trénování neuronových sítí pro vybrané aplikace.
- Simulační ověření vybraných aplikací v programech Matlab a Simulink.
- Návrh a sestavení laboratorního pohonu pro práci s řídicími algoritmy využívajícími neuronové sítě.
- Experimentální ověření vybraných aplikací řízení asynchronního motoru s využitím umělých neuronových sítí.

1 Neuronové sítě

Pojem umělé neuronové sítě (*Artificial Neural Networks – ANN*) byl odvozen na základě analogie s biologickými neuronovými sítěmi. Umělé neuronové sítě jsou založeny na poznatcích z neurofyzologie a představují hrubý model biologických neuronových sítí. Obsahují mnoho umělých neuronů (výpočetních jednotek) propojených mezi sebou prostřednictvím adaptivních spojení (váhy) do rozsáhlých paralelních struktur. Velmi důležitou vlastností je to, že vhodná ANN je schopná se naučit požadované zobrazení mezi vstupními a výstupními signály systému bez znalosti přesného matematického modelu daného systému. V tomto smyslu hovoříme o ANN jako o numerickém, trénovatelném, adaptivním estimátoru bez matematického modelu.

Neurony biologických sítí se dlouhým vývojem specializovaly na přijímání, zpracování, uchovávání, přenášení a využívání informací. Umělé neuronové sítě se do jisté míry pokouší napodobit funkci lidského mozku, který má schopnost se učit, dokáže paralelně zpracovávat informace a pracuje jako asociativní paměť závislostí mezi vektorem vstupních a vektorem výstupních proměnných [1].

Zjednodušeně můžeme hlavní rozdíl mezi neuronovými a konvenčními systémy vidět v tom, že konvenční systémy pracují sekvenčně, převážně podle přesného algoritmu a neuronové systémy pracují paralelně bez dopředu známého algoritmu. Jejich činnost je založena na procesu učení (trénování), při kterém se postupně adaptují k řešení dané úlohy. Pohlíží se na ně jako na „černou skříňku“, známe vstupy a výstupy, ale nepotřebujeme přesně rozumět tomu, co se děje uvnitř, aby byl získán požadovaný výstup.

Neuronové sítě mají řadu vlastností, které se dají výhodně využít v technické praxi. Mezi tyto vlastnosti patří mimo jiné schopnost aproximovat chování neznámého systému s potřebnou přesností, schopnost učení se z omezené množiny zadaných vzorů, schopnost zobecnění (generalizace, abstrakce), odolnost proti šumu a zkreslení ve vstupních datech. Velmi důležitá je schopnost zobecnění, protože to umožňuje, aby neuronová síť pracovala přesně i se vstupy, které nebyly součástí vzorů použitých při procesu učení.

Umělé neuronové sítě napodobují chování biologických neuronových sítí hlavně v těchto bodech:

- jsou tvořeny propojením neuronů
- znalosti jsou sbírány v procesu učení
- na ukládání těchto znalostí jsou využívána neuronová spojení
- informace jsou zpracovávány paralelně

Mezi vlastnosti umělých neuronových sítí patří:

- mají schopnost se učit neboli adaptovat k řešení dané úlohy
- jsou univerzální, dokážou realizovat libovolné zobrazení
- mají schopnost zobecnění
- jsou robustní, protože jsou schopné zobecnění a eliminace šumu
- jsou vhodné pro práci s neurčitými a nepřesnými informacemi

Nejrozšířenějším modelem z hlediska použití jsou především vícevrstvé neuronové sítě s adaptačním algoritmem zpětného šíření chyby (*Backpropagation*). Bylo dokázáno, že dopředná třívrstvá neuronová síť (s jednou skrytou vrstvou) se sigmoidou jako aktivační funkcí ve skryté a výstupní vrstvě je

univerzálním aproximátorem, tj. je schopna s požadovanou přesností aproximovat libovolnou spojitou funkci. Pokud máme funkci F specifikovanou pomocí množiny trénovacích vzorů, můžeme tedy s jistotou říct, že existuje taková neuronová síť, která funkci F s požadovanou přesností aproximuje. Neexistuje však obecný postup, jak takovou neuronovou síť navrhnout, z velké části záleží na konkrétní aplikaci. Dostupná pravidla jsou spíše založena na praktických zkušenostech než na teoreticky odvoditelných principech. Samotný návrh struktury sítě nebývá jednoduchý a bez předchozích zkušeností s danou aplikací je často jedinou cestou metoda pokus-omyl, kdy srovnáváme chování různých konfigurací neuronových sítí. Znalost matematického modelu systému je při návrhu výhodou. [13]

Následující kapitoly se věnují modelu umělého neuronu a druhům neuronových sítí, které jsou použitelné při modelování systémů. Popsaný model umělého neuronu s tansigmoidální aktivační funkcí je výchozí model, který je používán ve výpočetním prostředí *Neural Network Toolbox* v programu *Matlab*.

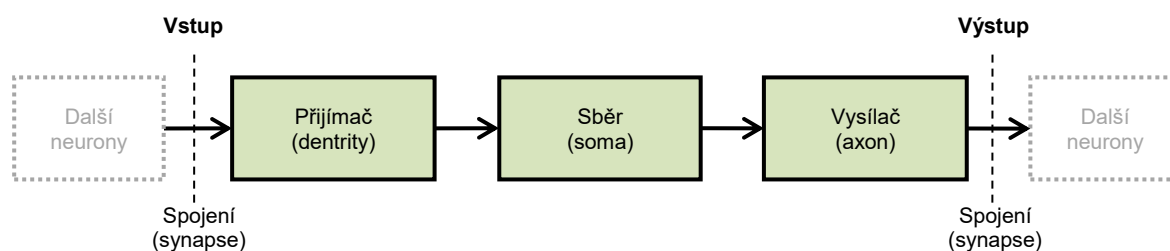
1.1 Biologické neuronové sítě

Biologické neurony (nervové buňky) jsou tvořeny třemi hlavními částmi – dendrity, tělem neuronu a axonem. S ostatními buňkami je neuron propojen prostřednictvím spojení zvaných synapse, kterými se váže na axony jiných neuronů. Takto propojené nervové buňky dohromady tvoří biologickou neuronovou síť (BNN).

Lidský mozek se skládá z více než stovky miliard buněk (neuronů), propojených vzájemně mezi sebou. Na každý neuron připadá přibližně 10^4 propojení s dalšími neurony, tj. celkové odhadované množství propojení v řádech 10^{15} , přičemž dva vzájemně propojené neurony mohou být vzdálené i více než jeden metr. Takovéto uskupení neuronů tvoří biologickou neuronovou síť, která poskytuje ohromný výpočetní výkon a paměť.

Na Obr. 1.1 je znázorněno základní blokové schéma reprezentující stavbu biologického neuronu. Schéma obsahuje tyto čtyři hlavní části:

1. **Vstupy** (přijímače informací) – tzv. dendrity (krátká vlákna), které přijímají informace od jiných neuronů.
2. **Sběr informací a nelineární transformace** – tělo buňky (soma), které sbírá, slučuje a transformuje příchozí informace od jiných neuronů.
3. **Výstup** (vysílač informací) – tzv. axon (jedno dlouhé nervové vlákno), kterým se přenáší informace do dalších neuronů.
4. **Spojení** – tzv. synapse (typický neuron má 10^3 až 10^5 synapsí). Spojovací bod axonu s dendritem jiného neuronu. Ve skutečnosti existují i další typy synaptických spojení: mezi dvěma axony, mezi dvěma dendrity a jiné; tyto však v základním modelu neuronu nejsou uvažovány.



Obr. 1.1 Blokové schéma představující biologickou nervovou buňku [1]

Dentry přivádějí do těla neuronu (soma) signály z jiných neuronů a tvoří tak vlastně vstupní část buňky. Na konci dentritu se nachází synapse, tedy místo, kde je dentrit připojen na axonové vlákno jiného neuronu. Úkolem synapse je předání elektrických impulzů mezi dvěma neurony. Dentrit není fyzicky přímo spojen s jinou buňkou (je od ní oddělen tzv. synaptickou mezerou), elektrické vzruchy vysílané axonovým vláknem jsou přeneseny z axonu na dentrit pomocí komplikovaných chemických procesů. Tento přenos zajišťují chemické látky nazývané neuropřenašeče (neurotransmitery). Přitom je zároveň upravena velikost vzruchu (zesílena nebo oslabena) a jeho smysl – vzruch může buňku excitovat nebo inhibovat. Tělo neuronu přijaté signály shromažďuje a v případě, že úroveň hromadění dosáhne vyšší hodnoty, než je tzv. práh excitace neuronu, je takto nashromážděný signál konvertován prostřednictvím nelineární aktivační transformace na nervové impulsy, které jsou vysílány podél axonového vlákna poskytujícího výstup neuronu (aktivace neuronu). Rychlost nervových impulzů závisí na mnoha faktorech, jako jsou velikost a délka axonu.

Mechanismy přenosu vzruchů i vlastní struktura a funkce různých neuronů a neuronových sítí jsou velmi komplikované. Dosavadní znalosti tedy zatím neumožňují ani vzdáleně popsat, jak přesně mozek funguje. Přesto vznikly i na základě velmi obecných a zjednodušujících poznatků modely umělých neuronových sítí, které v praxi prokazují jisté shodné prvky chování a vlastnosti, které pozorujeme u živých soustav.

1.2 Model umělého neuronu

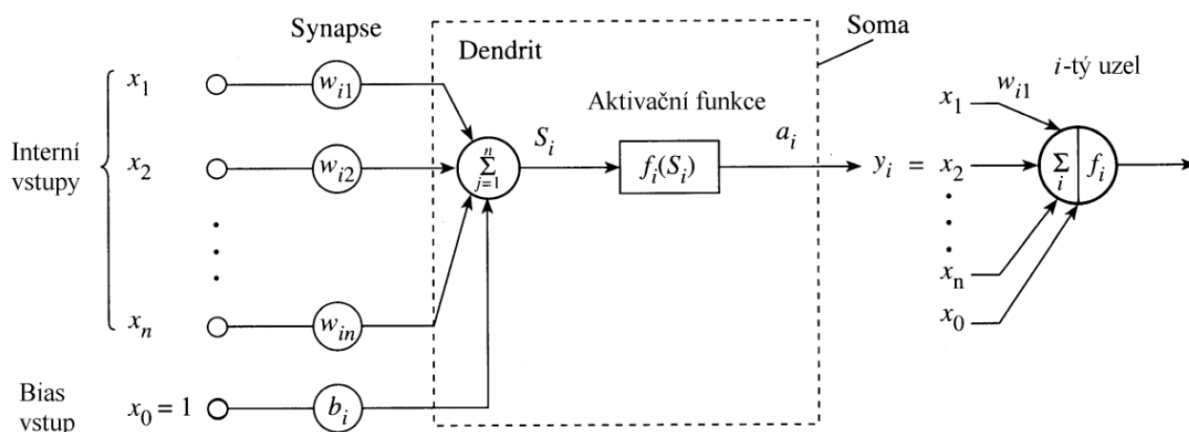
Základním prvkem (výpočetní jednotkou) umělé neuronové sítě je umělý neuron, který představuje matematický model jednotlivé buňky neuronové sítě. Souvislost mezi tímto matematickým modelem a biologickým neuronem je však více než volná. Umělý neuron nelze v žádném případě považovat ani za přibližný model biologického neuronu, protože ten slouží pouze jako inspirace pro vytvoření modelu umělého neuronu.

Obecně je umělý neuron vícevstupový, obvykle nelineární datazpracující prvek s váženými propojeními (synaptické váhy), který se snaží napodobit funkci biologického neuronu. Přestože je velmi složité najít přesný popis neuro-biologických procesů živých organismů, je možné obdržet poměrně jednoduchý model umělého neuronu, který může být implementován softwarově nebo hardwarově a dobře napodobuje chování skutečného neuronu.

Jednoduchý model biologického neuronu je možné chápat jako element, který přijímá signály (pomocí dentritů), zpracovává je (v těle buňky) a generuje odpovídající odezvu na svém výstupu

(prostřednictvím axonového vlákna). Na obrázku Obr. 1.2 je zobrazen model i -tého neuronu umělé neuronové sítě, skládá se z těchto čtyř hlavních částí:

1. **Vstup(y)**: externí nebo interní – příjem dat z vnějšího prostředí nebo z jiných neuronů
2. **Váhový sumátor**: provádí vážený součet, váhy zde zastupují funkci synapsí
3. **Aktivační funkce**: bývá někdy také označována jako přenosová funkce; ve většině případů se používá nelineární zobrazení, existují ale také modely, které používají lineární funkce
4. **Výstup(y)**: předávání dat dalším neuronům



Obr. 1.2 Statický model umělého neuronu [1]

Obecně vstupuje do i -tého neuronu n vstupů, které mohou být chápány jako členy n -rozměrného vektoru vstupních hodnot $\mathbf{x}_i = (x_1, \dots, x_n)$, který představuje výstupy jiných neuronů nebo vstupní signály zpracovávané neuronovou sítí. Vstupní hodnoty x_j mohou být obecně prvky různých množin hodnot. Mohou to být reálná čísla $x_j \in \mathbb{R}$, binární hodnoty $x_j \in \{0, 1\}$ nebo například intervaly, tedy $x_j = \langle a, b \rangle$, kde $a, b \in \mathbb{R}$. V řídicích aplikacích však neuronové sítě pracují většinou s reálnými čísly. Model neuronu můžeme zavést jako funkci:

$$y_i = f_i(\mathbf{x}_i) \quad \text{kde } y_i \in \mathbb{R}, \mathbf{x}_i \in \mathbb{R}^n \quad (1.1)$$

Protože se snažíme dosáhnout co nejjednodušší implementace, chceme, aby funkce f měla pouze jeden argument. Toho dosáhneme redukcí vstupů na jedinou hodnotu pomocí funkce p , toto vyjádříme přepsáním rovnice (1.1) do tvaru:

$$y_i = f_i(p(\mathbf{x}_i)) \quad (1.2)$$

Funkce f představuje aktivační funkci neuronu, prostřednictvím které se vykonává transformace vstupů na výstup. Výstup neuronu y je skalární veličinou.

Jako funkce p se nejčastěji použije vážený součet, kde je v souladu s biologickým vzorem každý vstup ohodnocen váhovými koeficienty w_{ij} nazývanými synaptické váhy, které mohou být excitační nebo inhibiční. Hodnoty těchto vah se nastavují během procesu učení. Pro i -tý neuron s n vstupy můžeme psát:

$$S_i = p(\mathbf{x}_i) = \sum_{j=1}^n w_{ij} x_{ij} \quad (1.3)$$

Funkční hodnotu funkce p nazýváme aktivační potenciál neuronu, jeho hodnota odpovídá míře excitace biologického neuronu. Neuron srovnává potenciál S_i s prahem excitace b_i , v případě překročení prahu excitace neuron aktivuje svůj výstup. Toto chování modeluje aktivační funkce neuronu ve tvaru:

$$f_i(S_i) = \begin{cases} 0 & \text{pro } S_i - b_i < 0 \\ 1 & \text{pro } S_i - b_i > 0 \end{cases} \quad (1.4)$$

Práh excitace b bývá označován také jako předpětí (*bias*).

Dalšího zjednodušení implementace neuronu dosáhneme tím, že vliv prahu excitace b_i vyjádříme přidáním dalšího vstupu x_0 , jehož hodnota je vždy rovna 1 a odpovídající váha $w_{i0} = b_i$. Vstupní vektor bude $\mathbf{x}_i = (1, x_1, \dots, x_n)$ a aktivační potenciál bude roven:

$$S_i = p(\mathbf{x}_i) = \sum_{j=1}^n w_{ij} x_{ij} + b_i \quad (1.5)$$

Výstup neuronu můžeme matematicky zapsat takto:

$$y_i = f_i \left(\sum_{j=1}^n w_{ij} x_{ij} + b_i \right) \quad (1.6)$$

Výše popsaný model umělého neuronu se skokovou aktivační funkcí je v literatuře označován jako perceptron. Tento pojem se však vyskytuje i v obecnějších souvislostech, kdy se nemusí jednat o neuronové sítě se skokovou aktivační funkcí, např. pojem *Multilayer Perceptron Networks* (MLP) označuje skupinu vícevrstevných neuronových sítí s libovolnou aktivační funkcí.

Jak je vidět, implementace umělého neuronu spočívá v operacích sčítání a násobení a ve výpočtu požadované aktivační funkce.

1.2.1 Aktivační funkce

Různé typy neuronových sítí se od sebe vzájemně liší ve dvou základních hlediscích. Kromě topologie neuronové sítě je klíčovým faktorem typ použitých neuronů, respektive jejich aktivační funkce. Mezi používané aktivační funkce patří např.:

- lineární funkce
- usměrněná lineární funkce (*Rectified Linear Unit* (ReLU))
- sigmoida (Obr. 1.3 a)
- radiální funkce (*Radial Basis Function*, RBF) (Obr. 1.3 b)
- skoková funkce (Obr. 1.3 c)
- hyperbolický tangens (tansigmoidální funkce) (Obr. 1.4)

Prostřednictvím aktivační funkce obvykle vnášíme do neuronových sítí nelinearitu, aby byl neuronový model schopen uskutečnit nelineární zobrazení mezi vstupem a výstupem. Neuronové modely s lineární aktivační funkcí mohou být použity jen pro modelování lineárních systémů. Při použití pouze lineární funkce by navíc neuronová síť nebyla schopna potlačit šumy, nebyla by robustní. [1]

Vhodná funkce by měla být nelineární a jednoduchá. Mnohé trénovací algoritmy jsou gradientní metody, které hledají kombinaci vah neuronů minimalizující chybu výstupu. Aktivační funkce pro neuronové sítě s adaptační metodou *backpropagation* musí mít následující vlastnosti: musí být spojitá, diferencovatelná a monotónně neklesající [13]. Historicky první použitou aktivační funkcí byl typ pevně ohraničené funkce (skoková funkce), která této podmínce nevyhovuje. Cílem bylo produkovat na výstupu neuronu pouze stavy ON a OFF. Tyto funkce nacházejí uplatnění především v klasifikačních (třídících) aplikacích. V oblasti modelování, řízení a diagnostiky jsou nejpoužívanější funkce asymptotické – Obr. 1.3 a), b) a Obr. 1.4.

Použití lineární funkce znamená pouze zkopírování hodnoty součtu na výstup:

$$f(S) = S \quad (1.7)$$

Pod pojem sigmoidální funkce spadá celá skupina asymptotických funkcí, které mají tvar písmene S. Sigmoida realizuje zobrazení $\mathbb{R} \mapsto (0, 1)$:

$$f(S) = \frac{1}{1 + e^{-aS}} \quad (1.8)$$

kde podle konstanty a , parametru strmosti, je možné měnit tvar této funkce. Sigmoidální funkce zahrnuje vlastnosti lineární i skokové (logické) funkce. S vyšší hodnotou a konverguje tvar sigmoidy ke skokové funkci perceptronu a pro $a \rightarrow \infty$ jsou obě funkce shodné. Z tohoto pohledu se na lineární a skokovou funkci můžeme dívat jako na limitní případy sigmoidální funkce.

Velkou výhodou sigmoidální funkce je fakt, že hodnotu její derivace můžeme určit přímo z její funkční hodnoty:

$$\frac{df(S)}{dS} = \frac{e^{-aS}}{(1 + e^{-aS})^2} = f(S)(1 - f(S)) \quad (1.9)$$

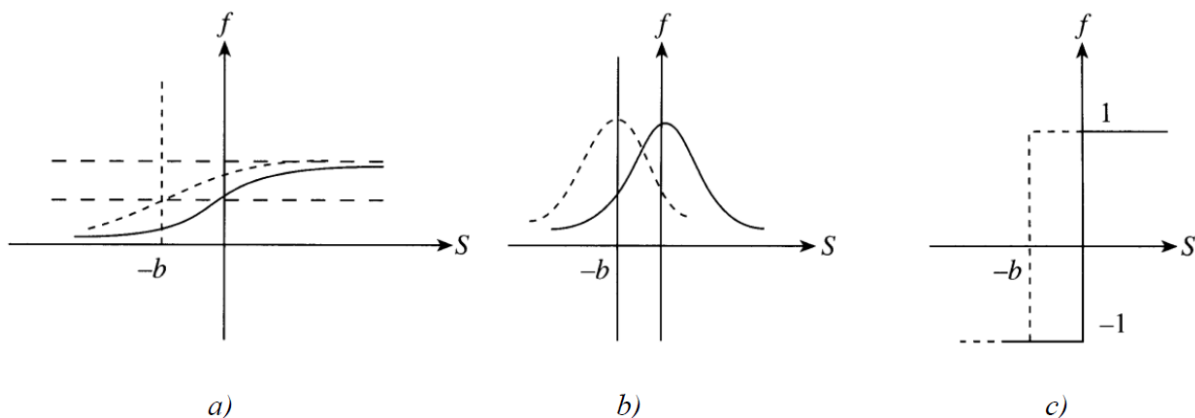
Často se používá symetrická varianta sigmoidy, hyperbolický tangens realizující zobrazení $\mathbb{R} \mapsto (-1, 1)$, která je definována jako:

$$F(S) = 2f(S) - 1 = \frac{1 - e^{-aS}}{1 + e^{-aS}} \quad (1.10)$$

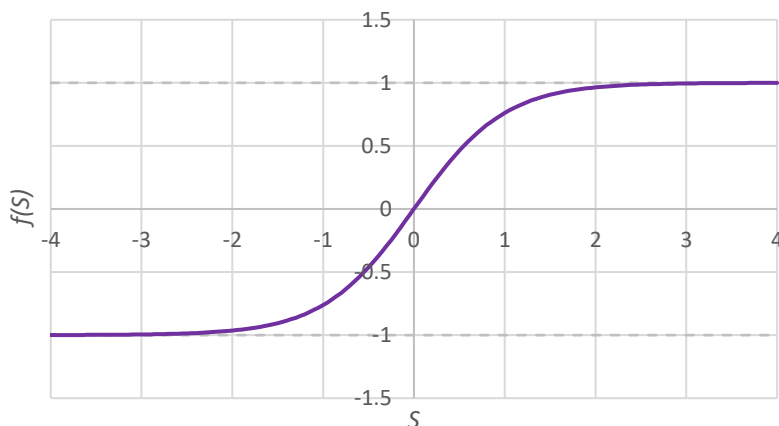
neboli

$$F(S) = \tanh\left(\frac{aS}{2}\right) \quad (1.11)$$

Tato funkce se také označuje jako tansigmoidální funkce (Obr. 1.4). Při zachování vlastností sigmoidy (1.8) obvykle poskytuje lepší výsledky.



Obr. 1.3 Podoba aktivačních funkcí (b je bias): a) sigmoida, b) radiální funkce, c) skoková funkce [1]



Obr. 1.4 Tansigmoidální funkce ($a = 2$)

1.3 Druhy neuronových sítí

Neuronové sítě můžeme rozlišovat podle vlastností použitých neuronů a struktury (topologie) neuronové sítě.

Podle vlastností použitých neuronů můžeme neuronové sítě dělit na základě dvou hledisek. Jednak dle použité aktivační funkce (skoková, RBF, lineární atd.), nebo podle homogenity sítě. Homogenní síť obsahuje všechny neurony stejného typu, naopak heterogenní síť je tvořena propojením neuronů různých typů lišících se použitou aktivační funkcí. Nejčastější jsou případy použití heterogenních sítí, ve kterých se používá specifických vlastností odlišných neuronů pro dosažení optimální funkce sítě. Při často bývá používána vícevrstvá síť s paralelními vrstvami, kde jsou u neuronů v tzv. skrytých vrstvách použity nelineární aktivační funkce a ve výstupní vrstvě je použita lineární aktivační funkce.

Jednotlivé neurony se mezi sebou vzájemně propojují tak, aby tvořily síť požadovaných vlastností. Neuronová síť má obvykle takovou strukturu, ve které jsou neurony uspořádány do skupin, které označujeme jako vrstvy a jsou řazeny za sebou. Tyto vrstvy jsou vzájemně propojeny, mezi neurony stejné vrstvy ale propojení nejsou. Můžeme hovořit o modelu s úplným propojením, kde je výstup každého neuronu předchozí vrstvy přiveden na vstupy všech neuronů následující vrstvy. V případě že některé ze vzájemných propojení chybí, nazývá se síť částečně propojená.

Rozlišujeme tři druhy vrstev:

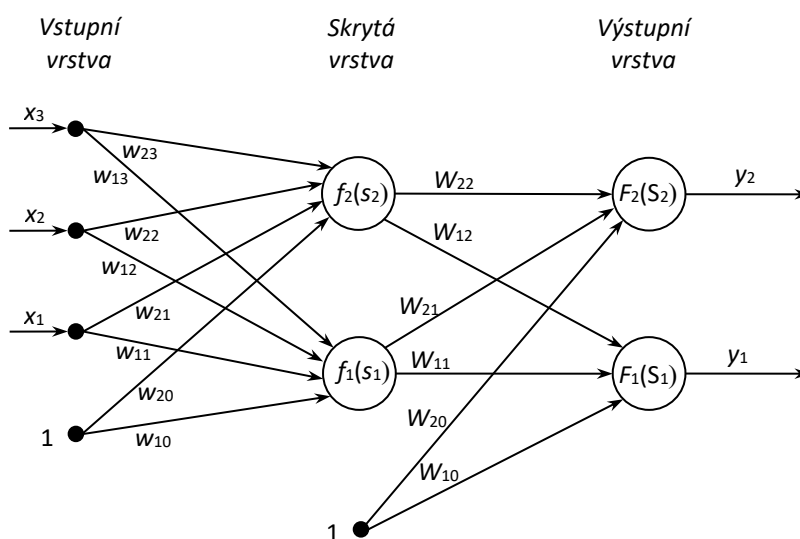
1. **Vstupní vrstva;** neurony této vrstvy neprovádí žádný výpočet, jsou to pouze uzly, od kterých se rozvádí vstupy neuronové sítě k neuronům ve skryté vrstvě, která se nazývá první skrytá vrstva.
2. **Skryté vrstvy;** podle použité aktivační funkce provádí transformaci vstupu na výstup, teoreticky může být jakýkoliv počet skrytých vrstev, ale tím narůstá náročnost procesu učení sítě. I sítě s jednou nebo dvěma skrytými vrstvami v mnoha případech poskytují dostatečnou přesnost, robustnost a schopnost zevšeobecnění.
3. **Výstupní vrstva;** zabezpečuje rozvod signálu z poslední skryté vrstvy a jejím výstupem je výstup celé neuronové sítě. Jako aktivační funkce mohou být použity lineární funkce, protože vícevrstvá síť je schopna reprezentovat nelineární systém použitím nelineárních aktivačních funkcí ve skryté vrstvě (skrytých vrstvách).

Neuronové sítě můžeme dělit na sítě bez zpětné vazby – přímá neuronová síť (*Feedforward Neural Network*) nebo sítě se zpětnou vazbou – rekurentní neuronová síť (*Recurrent Neural Network, RNN*). Neuronové sítě obsahující zpětné vazby mohou mít, na rozdíl od sítí přímých, krátkodobou paměť neboli vnitřní stav. Rekurentní sítě obsahují stavové neurony, které slouží jako paměť vnitřního stavu sítě. Výstup z předchozího výpočetního kroku je přiveden zpět na vstup stejného neuronu nebo i dalších neuronů ve stejné nebo předchozí vrstvě. Rekurentní sítě se často používají hlavně při modelování dynamických systémů, které obsahují stavové proměnné. Výstup takových systémů závisí nejen na aktuálním vstupu, ale i na jeho předchozích stavech. Přítomnost zpětné vazby má významný vliv na schopnost učení a výkonost sítě. Dopředné sítě můžeme také označit jako statické, rekurentní jako dynamické.

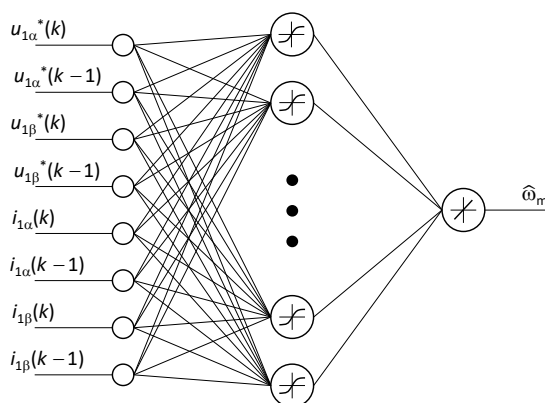
Často používaným typem neuronové sítě je dopředná síť s jednou nebo více skrytými vrstvami. Příklad struktury dopředné dvouvrstvé sítě je na Obr. 1.5. Označení 3-2-2 znamená, že má tři vstupy x (tři neurony ve vstupní vrstvě), dva neurony ve skryté vrstvě a dva výstupy y (dva neurony ve výstupní vrstvě). Jako w_{ij} jsou označeny váhové koeficienty (váhy) spojení mezi i -tým neuronem skryté vrstvy a j -tým vstupem. Jako W_{ij} jsou označeny váhy spojení mezi i -tým neuronem výstupní vrstvy a j -tým neuronem předchozí, skryté vrstvy. Jako f_i je označena aktivační funkce i -tého neuronu ve skryté vrstvě, ve skutečnosti se používá stejná aktivační funkce pro všechny neurony ve skryté vrstvě. F je výstupní aktivační funkce. Jako s_i a S_i je označen součet všech vstupů neuronu. Váhové koeficienty w_{i0} a W_{i0} představují biasy. Takováto síť je někdy v literatuře označována jako třívrstvá, což odpovídá faktu, že jsou zde tři vrstvy uzlů (vstupní, jedna skrytá a výstupní vrstva). V této síti jsou ale jen dvě skutečné vrstvy obsahující datazpracující neurony, ve vstupní vrstvě nejsou prováděny výpočty. Z tohoto pohledu je logičtější takovou síť nazývat jako dvouvrstvou, tedy že počet vrstev takovéto sítě je roven počtu skrytých vrstev plus jedna výstupní. [1]

Při konfiguraci sítě na Obr. 1.5 je dobrého výsledku při modelování nelineárního systému možné dosáhnout použitím nelineární aktivační funkce ve skryté vrstvě a lineární aktivační funkce ve výstupní vrstvě neuronové sítě. Ve srovnání s touto sítí, která má jednu skrytou vrstvu, může síť s více skrytými vrstvami poskytnout vyšší přesnost při menším počtu neuronů ve skrytých vrstvách. Použití většího počtu skrytých vrstev a neuronů může být omezeno výpočetní náročností.

Existují úlohy, při kterých jednomu vstupu může být přiřazeno více výstupů v závislosti na časovém kontextu – výstup závisí i na předchozích hodnotách vstupu. Při řešení těchto úloh se využívají neuronové sítě s časovým posunem (*Time Delay Neural Network, TDNN*), které mají na vstup přiveden kromě aktuálního vstupu (v čase t) i hodnoty vstupu z minulých výpočetních kroků (v čase $t - T_{vz}$, $t - 2T_{vz}$, ..., $t - NT_{vz}$). TDNN je možné trénovat klasickým algoritmem *backpropagation*, oproti dopředné neuronové síti bez časového posunu může síť s časovým posunem lépe pracovat s časovými průběhy a přesněji modelovat dynamické systémy. [1] S rostoucím počtem vzorků na vstupu se přesnost neuronové sítě zvyšuje, ale zároveň se zvyšuje také výpočetní náročnost. Příklad dopředné sítě s časovým posunem je na Obr. 1.6, kromě aktuálních vstupních hodnot (krok k) jsou na vstup přiváděny i hodnoty z předchozího kroku (krok $k - 1$).



Obr. 1.5 Přímá dvouvrstvá neuronová síť 3-2-2 [8]

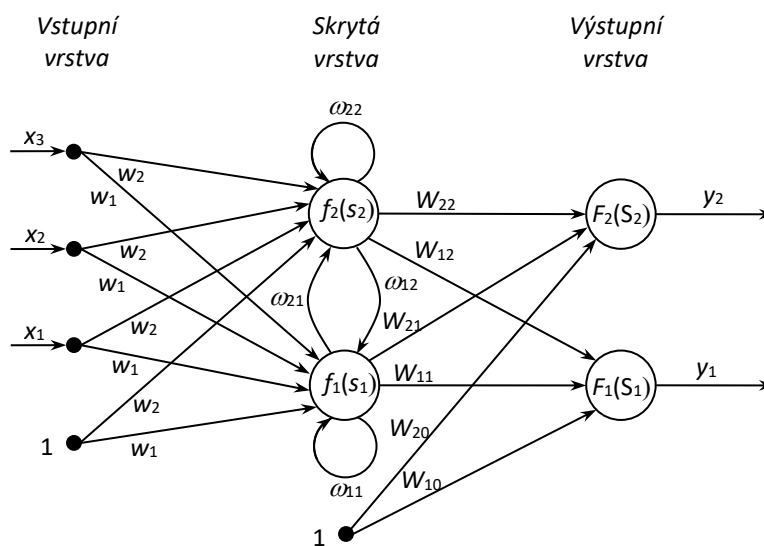


Obr. 1.6 Struktura neuronové sítě estimátoru rychlosti ASM (8-22-1) [8]

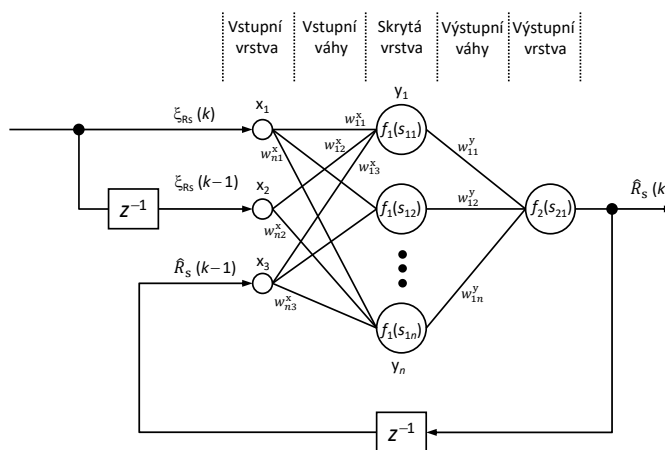
Příklad rekurentní neuronové sítě je na Obr. 1.7. Ve srovnání s Obr. 1.5 jsou zde navíc váhové koeficienty ω zpětnovazebního spojení z výstupu skryté vrstvy zpět na vstup. V literatuře se můžeme setkat s typy rekurentních neuronových sítí, které se označují jako Elmanova a Jordanova neuronová síť. Obě sítě mají ve vstupní vrstvě kromě vnějších vstupů x i další, stavové vstupy. U Elmanovy sítě je stavový vstup tvořen výstupy skryté vrstvy. Jordanova síť má vnější zpětnou vazbu, na vstup neuronové

sítě je přiveden výstup neuronů výstupní vrstvy. Také existuje Hopfieldova rekurentní neuronová síť, která je tvořena jednou vrstvou plně propojených neuronů, výstup každého neuronu je přiveden na vstup všech ostatních neuronů. Hopfieldova síť se používá např. při rozpoznávání obrazu.

Podle výše uvedeného by neuronová síť na Obr. 1.7 mohla být označena jako Elmanova rekurentní neuronová síť. Na Obr. 1.8 je příklad rekurentní neuronové sítě s vnější zpětnou vazbou, která byla použita pro adaptaci statorového odporu asynchronního motoru v systému MRAS (ξ vyjadřuje chybový signál vyhodnocený podle výstupů referenčního a adaptivního modelu).



Obr. 1.7 Rekurentní dvouvrstvá neuronová síť 3-2-2 [8]



Obr. 1.8 Struktura neuronové sítě pro adaptaci hodnoty statorového odporu R_s [34]

Rekurentní sítě je možné trénovat algoritmem zpětného šíření chyby v čase (*Backpropagation Through Time*, BPTT). Princip vychází z toho, že rekurentní síť lze rozvinutím v čase převést na síť nerekurentní.

Jedním z dalších typů rekurentní sítě, která je vhodná pro predikci časových řad a modelování dynamických systémů, je i neuronová síť typu NARX (*Nonlinear Autoregressive Exogenous Model*). Je to rekurentní neuronová síť s vnější zpětnou vazbou, jejíž výstup $y(k)$ je kromě aktuálního vstupu $x(k)$ závislý na N předchozích vstupech $x(k-1), \dots, x(k-N)$ a M předchozích výstupech $y(k-1), \dots, y(k-M)$, je zde časové zpoždění.

Jako RBF sítě se označují neuronové sítě, které ve skryté vrstvě používají radiální aktivační funkci. [14] Dalším typem jsou fuzzy neuronové sítě (*Adaptive Neuro-Fuzzy Inference System*, ANFIS), které kombinují principy neuronových sítí a fuzzy logiky. Výhodou je to, že mají dobře definovanou architekturu. [15][16]

Zajímavým typem neuronových sítí jsou také tzv. waveletové neuronové sítě (*Wavelet Neural Networks*, WNN), které kombinují teorii neuronových sítí a waveletové (vlnkové) transformace. Jako aktivační funkce je použit mateřský wavelet. [17][18]

Architektura neuronové sítě závisí na dané aplikaci a složitosti zobrazení, kterého má být dosaženo. Dopředné neuronové sítě s jednou skrytou vrstvou jsou univerzálními funkčními aproximátory, prakticky ale nevíme, kolik skrytých neuronů je potřeba použít pro rovnováhu mezi výpočetní náročností sítě a schopností sítě se naučit dané zobrazení s požadovanou přesností. Počet neuronů a spojení omezuje množství vzorů (složitost zobrazení), které může neuronová síť uchovávat. Každá skrytá vrstva může obsahovat jiný počet neuronů. Obecně lze říct, že při malém počtu neuronů ve skryté vrstvě se sice zkrátí doba výpočtu, ale klesá přesnost sítě. Naopak pokud je počet neuronů ve skryté vrstvě příliš velký, doba učení a výpočtu se neúměrně prodlouží a síť ztrácí schopnost zobecnění.

Počet vstupů (vstupních uzlů) a výstupů (výstupních neuronů) je samozřejmě dán řešenou úlohou. Počet vstupů odpovídá počtu veličin, které jsou uvažovány jako nezbytné pro správnou funkci neuronové sítě a počet výstupů je roven počtu výstupních veličin, které mají být určovány. Výhodou při návrhu neuronové sítě může být, pokud je znám matematický model systému.

1.4 Učení neuronových sítí

Jednou z nejvýznamnějších vlastností neuronových sítí je jejich schopnost učit se na základě podnětů z okolního prostředí a po dobu učení zlepšovat svou výkonnost. V kontextu neuronových sítí se často pojem učení definuje následovně: Učení je proces, ve kterém jsou parametry sítě adaptovány na základě stimulace okolním prostředím, kterým je síť obklopená. Typ učení je dán způsobem, kterým tato změna nastává.

V souvislosti s procesem učení neuronových sítí rozlišujeme dva pracovní stavy – adaptivní a aktivní.

V adaptivním režimu se mění konfigurace sítě podle požadavků dané úlohy. Adaptační proces může vést k modifikaci hodnot parametrů sítě (váhy a předpětí), ke změně aktivačních funkcí i ke změně topologie celé sítě.

V aktivním režimu se konfigurace sítě nemění. Neuronová síť realizuje určité zobrazení vstupních vektorů na výstupní vektory tak, jak se to naučila v adaptivním režimu.

K naučení sítě je potřebná jednak tzv. trénovací množina obsahující prvky popisující danou problematiku a dále pak metoda, která dokáže tyto vzory zafixovat v neuronové síti formou hodnot synaptických vah.

1.4.1 Trénovací algoritmy

Trénovací algoritmus je adaptivní metoda, jejíž pomocí se neuronová síť adaptuje tak, aby realizovala požadovanou funkci. Obecně může trénovací algoritmus modifikovat jakékoliv vlastnosti neuronové sítě, aby dosáhl požadovaného výsledku, včetně její struktury. Takový stupeň volnosti ovšem přináší zvýšené nároky na složitost algoritmu. Proto jsou základní adaptační algoritmy nejčastěji založené na tom, že vzhledem ke zvolenému kritériu nastavují optimálním způsobem váhy sítě.

Podle charakteru učicího procesu je možné trénovací algoritmy rozdělit do následujících skupin:

1. **Učení s učitelem** – kontrolované učení. Tento způsob učení se obvykle uskutečňuje pomocí vhodných trénovacích vzorů. Neuronové síti je předkládána množina příkladů požadovaného chování sítě – vstupy a jim odpovídající výstupy. Trénovací algoritmus pracuje v uzavřené smyčce, porovnává odezvu sítě na vstupní data s požadovaným výstupem a iteračně upravuje parametry sítě, aby minimalizoval chybu. Cílem tohoto procesu je, aby ANN na základě explicitní informace, kterou představují vzorové páry vstupů a odpovídajících výstupů, dosáhla takové konfigurace, na základě které dokáže realizovat požadované zobrazení. Učení s učitelem se velmi často využívá při identifikaci, diagnostice a modelování systémů.
2. **Učení bez učitele** – nekontrolované učení. Pro správnou funkci stačí ANN zadat vstupní vektory a ANN na základě svých interních pravidel sama vyhledá charakteristické znaky procesu, podle nichž upraví své nastavení. Tato metoda učení, která je v porovnání s předcházející časově mnohem náročnější, se nejčastěji používá u Kohenových a samoorganizujících sítí. Užívá se tehdy, když přesný výstup neuronové sítě není znám. Taková situace nastává například tehdy, kdy chceme identifikovat jednotlivé skupiny vstupních vektorů, ale nevíme, kolik se takových skupin ve vstupních datech může objevit. Obvykle se podobná zadání vyskytují v oblasti analýzy dat.
3. Na pomezí předcházejících způsobů učení se nacházejí způsoby ohodnoceného učení – známkového učení. Při tomto učení učitel poskytuje neuronové síti hodnocení její reakce na přicházející vstupy tak, aby proces učení vedl k žádanému cíli.

V některých případech lze použít pro adaptaci sítě stochastické metody, kdy jsou parametry sítě hledány postupnými testy náhodně zvolených hodnot. Typickými stochastickými algoritmy jsou např. genetické algoritmy.

Metody učení můžeme také rozdělit na *offline* a *online*. Při způsobu učení *offline* nezáleží na době potřebné k nastavení sítě. Neuronová síť je natrénována před svým použitím na základě předem zaznamenaných hodnot nebo na základě údajů z modelu systému. V průběhu provozu sítě již k adaptaci nedochází. Metody *online* naopak průběžně adaptují váhy neuronové sítě. Tím se síť může neustále přizpůsobovat případným změnám parametrů soustavy.

1.4.2 Metoda backpropagation

Podle zvoleného typu neuronové sítě volíme i algoritmus jejího učení. Nejrozšířenějším adaptačním algoritmem vícevrstevných přímých neuronových sítí je metoda *backpropagation* (BP), česky metoda *zpětného šíření chyby*, která umožňuje adaptaci neuronové sítě nad danou trénovací množinou. Popis metody *backpropagation* je převzat z [13].

Každý vzor trénovací množiny popisuje, jakým způsobem jsou excitovány neurony vstupní a výstupní vrstvy. Formálně můžeme za trénovací množinu \mathbf{T} považovat množinu prvků (vzorů), které jsou definovány uspořádanými dvojicemi vektorů následujícím způsobem:

$$\mathbf{T} = \{(\mathbf{x}_k, \mathbf{t}_k) | \mathbf{x}_k \in \{0, 1\}^n, \mathbf{t}_k \in \{0, 1\}^m, k = 1, \dots, q\} \quad (1.12)$$

kde q je počet vzorů trénovací množiny, \mathbf{x}_k vektor excitací vstupní vrstvy tvořené n neurony a \mathbf{t}_k vektor excitací výstupní vrstvy tvořené m neurony.

Samotný algoritmus obsahuje tři etapy: dopředné (*feedforward*) šíření vstupního signálu, zpětné šíření chyby a aktualizace váhových hodnot na spojeních.

Nejprve získáme odezvu sítě na vstupní podnět daný excitací neuronů vstupní (senzorické) vrstvy. Během dopředného šíření signálu obdrží každý neuron ve vstupní vrstvě vstupní signál (x_i) a zprostředkuje jeho přenos ke všem neuronům vnitřní vrstvy. Každý neuron ve vnitřní vrstvě vypočítá svou aktivaci (z_j) a pošle tento signál všem neuronům ve výstupní vrstvě. Každý neuron ve výstupní vrstvě vypočítá svou aktivaci (y_k), která odpovídá skutečnému k -tému výstupu neuronové sítě.

Během adaptace neuronové sítě metodou *backpropagation* jsou srovnávány vypočítané aktivity y_k s definovanými výstupními hodnotami t_k pro každý neuron ve výstupní vrstvě a pro každý vzor trénovací množiny. Na základě tohoto srovnání je definována chyba neuronové sítě, pro kterou je vypočítán faktor δ_k ($k = 1, \dots, m$), jenž odpovídá části chyby, která se šíří zpětně z neuronu Y_k ke všem neuronům předcházející vrstvy majícím s tímto neuronem definovaná spojení. Podobně lze definovat i faktor δ_j ($j = 1, \dots, p$), který je částí chyby šířené zpětně z neuronu Z_j ke všem neuronům vstupní vrstvy, jež mají s tímto neuronem definovaná spojení. Úprava váhových hodnot w_{kj} na spojeních mezi neurony vnitřní a výstupní vrstvy závisí na faktoru δ_j a aktivacích z_j neuronů Z_j ve vnitřní vrstvě. Úprava váhových hodnot v_{ji} na spojeních mezi neurony vstupní a vnitřní vrstvy závisí na faktoru δ_j a aktivacích x_i neuronů X_i ve vstupní vrstvě.

Chyba sítě $E(\mathbf{w})$ je definována jako součet parciálních chyb sítě $E_l(\mathbf{w})$ vzhledem k jednotlivým vzorům trénovací množiny a závisí na konfiguraci sítě \mathbf{w} jako:

$$E(\mathbf{w}) = \sum_{l=1}^q E_l(\mathbf{w}) \quad (1.13)$$

Parciální chyba $E_l(\mathbf{w})$ sítě pro l -tý tréninkový vzor ($l = 1, \dots, q$) je úměrná součtu mocnin odchylek skutečných hodnot výstupu sítě pro vstup l -tého vzoru trénovací množiny od požadovaných hodnot výstupů u tohoto vzoru jako:

$$E_l(\mathbf{w}) = \frac{1}{2} \sum_{k \in Y} (y_k - t_k)^2 \quad (1.14)$$

Cílem adaptace je minimalizace chyby sítě ve váhovém prostoru. Vzhledem k tomu, že chyba sítě přímo závisí na komplikované nelineární složené funkci vícevrstvé sítě, znamená tento cíl netriviální optimalizační problém.

Pro jeho řešení představuje v základním modelu metoda *backpropagation* nejjednodušší variantu gradientní metody. Chybová funkce určuje chybu sítě vzhledem k trénovací množině v závislosti na konfiguraci sítě. Při adaptaci sítě hledáme takovou konfiguraci, pro kterou je chybová funkce

minimální. Začneme s náhodně zvolenou konfigurací $\mathbf{w}^{(0)}$, kdy zřejmě bude odpovídající chyba sítě od požadované funkce velká.

Při adaptaci sestrojíme v tomto bodě $\mathbf{w}^{(0)}$ ke grafu chybové funkce tečný vektor (gradient):

$$\frac{\partial E}{\partial \mathbf{w}}(\mathbf{w}^{(0)}) \quad (1.15)$$

a posuneme se ve směru tohoto vektoru dolů o krok ε . Pro dostatečně malé ε tak získáme novou konfiguraci $\mathbf{w}^{(1)} = \mathbf{w}^{(0)} + \Delta \mathbf{w}^{(1)}$, pro kterou je chybová funkce menší než pro původní konfiguraci $\mathbf{w}^{(0)}$, tj. $E(\mathbf{w}^{(0)}) \geq E(\mathbf{w}^{(1)})$. Celý proces konstrukce tečného vektoru opakujeme pro $\mathbf{w}^{(1)}$ a získáme tak $\mathbf{w}^{(2)}$ takové, že $E(\mathbf{w}^{(1)}) \geq E(\mathbf{w}^{(2)})$ atd., až se limitně dostaneme do lokálního minima chybové funkce. Tato metoda vždy konverguje k nějakému lokálnímu minimu z libovolné počáteční konfigurace, není však zaručeno, že se tak stane v reálném čase. Obvykle je tento proces časově velmi náročný i pro malé topologie vícevrstvé sítě (desítky neuronů).

Hlavním problémem gradientní metody je, že nemůžeme stanovit, zda je nalezené minimum globální. Uvedený postup adaptace se v takovémto minimu zastaví (nulový gradient) a chyba sítě se již dále nesnižuje.

Pro třívrstvou neuronovou síť můžeme adaptační algoritmus *backpropagation* popsat takto:

Krok 0. Váhy a prahy (předpětí) jsou inicializovány malými náhodnými čísly. Přiřazení inicializační hodnoty koeficientu učení α , který určuje velikost změny vah.

Krok 1. Dokud není splněna podmínka ukončení výpočtu, opakovat kroky (2–9).

Krok 2. Pro každý pár vektorů trénovací množiny $\mathbf{s}:\mathbf{t}$ provádět kroky (3–9).

Feedforward:

Krok 3. Aktivovat vstupní neurony ($X_i, i = 1, \dots, n$) $x_i = s_i$.

Krok 4. Vypočítat vstupní hodnoty vnitřních neuronů ($Z_j, j = 1, \dots, p$):

$$z_in_j = \sum_{i=1}^p v_{ji} x_i + v_{j0}$$

Stanovení výstupních hodnot vnitřních neuronů $z_j = f(z_in_j)$.

Krok 5. Stanovení skutečných výstupních hodnot neuronové sítě ($Y_k, k = 1, \dots, m$):

$$y_in_k = \sum_{j=1}^m w_{kj} z_j + w_{k0} y_k = f(y_in_k)$$

Backpropagation:

Krok 6. Ke každému neuronu ve výstupní vrstvě ($Y_k, k = 1, \dots, m$) je z trénovací množiny přiřazena hodnota očekávaného výstupu. Dále je vypočteno:

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

které je součástí váhové korekce $\Delta w_{kj} = \alpha \delta_k z_j$ i korekce prahu $\Delta w_{k0} = \alpha \delta_k$.

Krok 7. Ke každému neuronu ve vnitřní vrstvě ($Z_j, j = 1, \dots, p$) je přiřazena sumace jeho „delta“ vstupů (tj. neuronů, které se nacházejí v následující vrstvě):

$$\delta_{in_j} = \sum_{k=1}^m w_{kj} \delta_k$$

Vynásobením získaných hodnot derivací jejich aktivační funkce obdržíme:

$$\delta_j = \delta_{in_j} f'(z_{in_k})$$

které je součástí váhové korekce $\Delta v_{ji} = \alpha \delta_j x_i$ i korekce prahu $\Delta v_{j0} = \alpha \delta_j$.

Aktualizace vah a prahů:

Krok 8. Každý neuron ve výstupní vrstvě ($Y_k, k = 1, \dots, m$) aktualizuje na svých spojeních váhové hodnoty včetně svého prahu ($j = 0, \dots, p$):

$$w_{kj}(\text{new}) = w_{kj}(\text{old}) + \Delta w_{kj}.$$

Každý neuron ve vnitřní vrstvě ($Z_j, j = 1, \dots, p$) aktualizuje na svých spojeních váhové hodnoty včetně svého prahu ($i = 0, \dots, n$):

$$v_{ki}(\text{new}) = v_{ki}(\text{old}) + \Delta v_{ki}.$$

Krok 9. Podmínka ukončení: Pokud již nenastávají žádné změny váhových hodnot nebo pokud již bylo vykonáno maximálně definované množství váhových změn, stop; jinak, pokračovat.

Účelem algoritmu je nalézt globální minimum funkce E . Mimo popsanou základní gradientní metodu existují i další, jako jsou metody cyklické změny parametrů a náhodné optimalizace. Pro praktické aplikace jsou základní metody BP velmi pomalé. Proto se používají různé modifikace algoritmu *backpropagation*, které konvergují 10 až 100krát rychleji. Každá z metod má svoje přednosti i nedostatky. Při trénování popsaných vícevrstevých dopředných neuronových sítí dosahuje nejlepších výsledků, co se týče přesnosti a rychlosti, Levenberg-Marquardtův algoritmus. [8][11]

Trénování neuronové sítě je v podstatě optimalizační úloha, nehledáme jediné správné řešení, hledáme přijatelné řešení z hlediska přesnosti a výpočetní náročnosti. Pro trénování sítí mohou být použity i optimalizační algoritmy z oblasti *soft computingu*, např. genetický algoritmus. Jejich výhodou je především schopnost se vyhnout lokálním minimům chybové funkce E a také rychlejší konvergence řešení. [3]

1.5 Aplikace umělých neuronových sítí

Při řešení úloh z oblasti elektrických pohonů se setkáváme s těmito základními problémy:

1. Modelování systému
2. Identifikace parametrů systému
3. Odhady a pozorování stavových veličin
4. Řízení pohonu
5. Diagnostika stavu zařízení

Ve všech zmíněných oblastech je možné aplikovat umělé neuronové sítě. Realizace uvedených problémů za použití klasických řídicích teorií je relativně jednoduchá jen pro lineární systémy, ale v případě nelineárních systémů, jakými jsou elektrické pohony, musíme při použití klasických metod často řešit mnoho problémů, a to jak na poli syntézy daného úkolu, tak i v oblasti praktické realizace. V klasických řídicích metodách je znalost regulovaného systému vyžadována ve tvaru algebraických a diferenciálních rovnic, které analyticky vyjadřují vztah mezi vstupy a výstupy. Tyto modely jsou často velmi složité, závisí na mnoha předpokladech a obsahují parametry, které jsou obtížně měřitelné nebo se značně mění během činnosti systému. Z uvedených důvodů se staly alternativou umělé neuronové sítě se svými charakteristickými vlastnostmi: schopnost aproximovat chování neznámého systému s potřebnou přesností, schopnost aproximací nelineárních funkcí, schopnost učení se požadovaného chování na základě dodaných trénovacích dat, schopnost zevšeobecnění, rychlost zpracování informací a odolnost proti šumu a zkreslení. K nejvíce používaným topologiím v řídicích aplikacích patří přímé vícevrstvé neuronové sítě. [8][11][19] – [30]

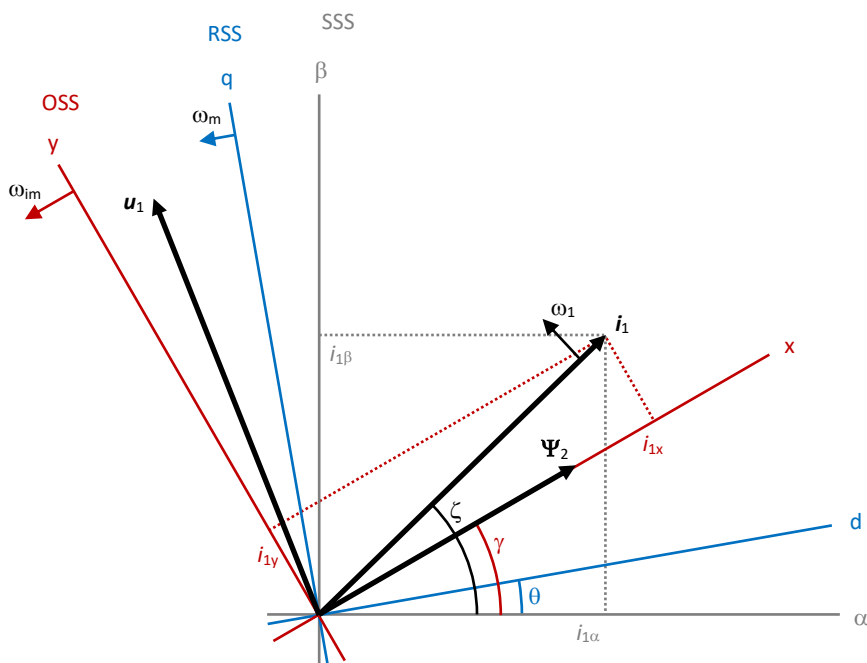
2 Bezsenzorové řízení – pozorovatelé MRAS

Jako bezsenzorové řízení se označuje regulace mechanické úhlové rychlosti střídavých strojů bez snímače otáček nebo polohy. Cílem je nalézt odhad (*estimaci*) polohy rotoru či mechanické úhlové rychlosti na základě jiných měřených veličin. Použití těchto metod vede k vyšší spolehlivosti pohonu, protože není potřeba použít snímač otáček nebo tyto metody mohou sloužit jako záložní systém pro případ poruchy snímače otáček. V případě napájení asynchronního motoru z nepřímého měniče kmitočtu máme k dispozici snímač napětí v meziobvodu měniče u_d a dva snímače fázových proudů i_{1a} a i_{1b} .

Jelikož se tato práce zabývá bezsenzorovým vektorovým řízením, je v následující kapitole stručně popsán algoritmus vektorového řízení asynchronního motoru. Následuje stručný přehled a rozdělení metod bezsenzorového řízení. Podrobněji budou rozebrány pozorovatelé RF-MRAS a CB-MRAS.

2.1 Vektorové řízení

V případě vektorového řízení asynchronního motoru je přesná informace o poloze rotoru stroje mimo rychlostní a polohovou regulaci důležitá také pro výpočet vektoru rotorového magnetického toku Ψ_2 , jehož velikost $|\Psi_2|$ nám dává informaci o buzení stroje a poloha γ slouží k vektorovému natočení veličin ze SSS do OSS a zpět (viz Obr. 2.1). Vektor statorového proudu i_1^0 má dvě složky, tokotvornou složku i_{1x} , jejíž velikost je v ustáleném stavu úměrná velikosti rotorového toku, a momentotvornou složku i_{1y} , jejíž velikost je úměrná velikosti momentu stroje. Tento rozklad statorového proudu umožňuje realizovat oddělenou regulaci momentu a buzení stroje. Proud i_{1x} se reguluje napětím u_{1x} a proud i_{1y} napětím u_{1y} . Chyba určení polohy vektoru magnetického toku vede k nedokonalému rozkladu na momentotvornou a tokotvornou složku a má nepříznivý vliv na dynamiku pohonu.



Obr. 2.1 Souřadné soustavy pro matematický popis asynchronního motoru

V kap. 5 je popsáno laboratorní stanoviště, na kterém byly získány experimentální výsledky. Algoritmus vektorového řízení, který byl implementován v řídicím systému, pracuje podle rovnic matematického modelu asynchronního motoru s kotvou nakrátko ve vztahné soustavě OSS, které jsou uvedeny níže, viz také vektorový diagram na Obr. 2.1. Blokové schéma vektorového řízení je na Obr. 2.2. Podrobnější popis je např. v [5][6].

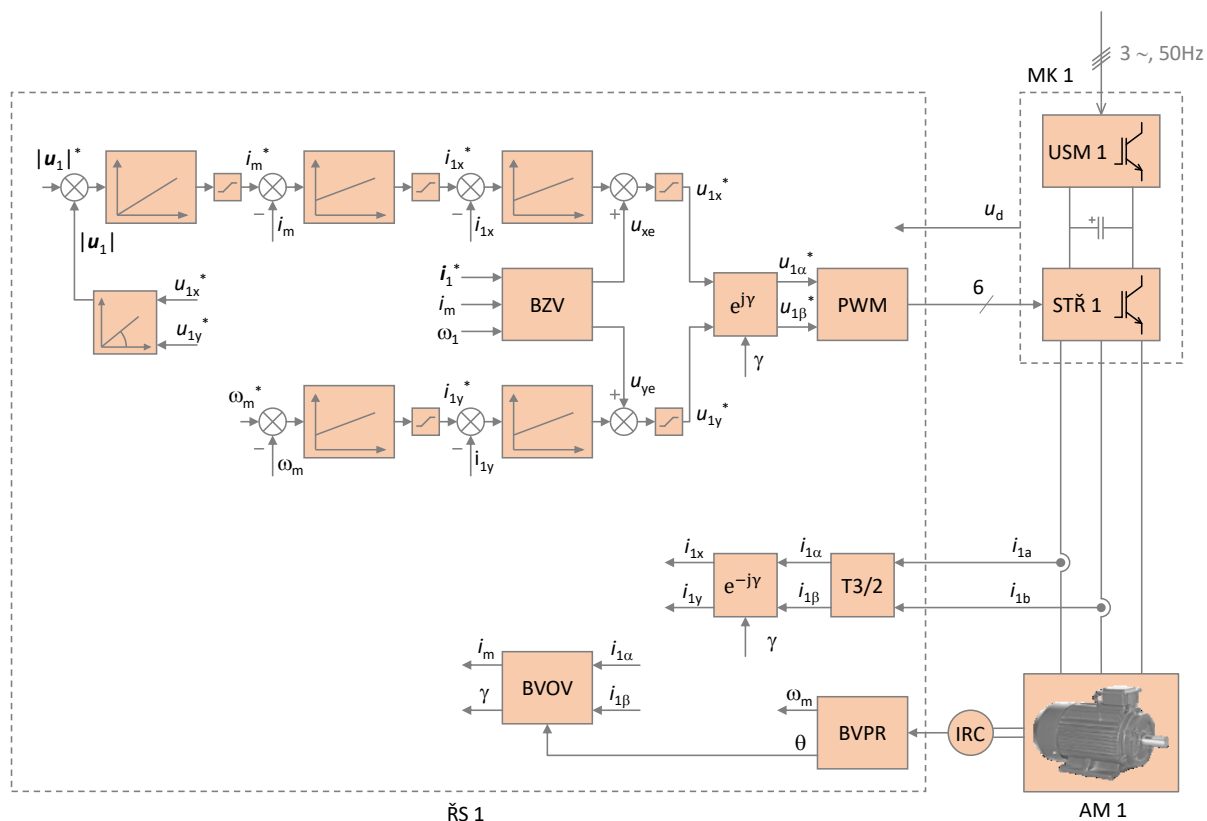
$$\sigma T_s \frac{di_{1x}}{dt} + i_{1x} = \frac{u_{1x}}{R_s} + \omega_{im} \sigma T_s i_{1y} - (1 - \sigma) T_s \frac{di_m}{dt} \quad (2.1)$$

$$\sigma T_s \frac{di_{1y}}{dt} + i_{1y} = \frac{u_{1y}}{R_s} - \omega_{im} \sigma T_s i_{1x} - (1 - \sigma) \omega_{im} T_s i_m \quad (2.2)$$

$$J \frac{d\omega_m}{dt} = \frac{3}{2} p \frac{L_h}{1 + \sigma_r} i_m i_{1y} - m_z \quad (2.3)$$

$$T_r \frac{di_m}{dt} + i_m = i_{1x} \quad i_m = \frac{|\Psi_2|}{L_h} \quad (2.4)$$

$$\omega_1 = \frac{d\zeta}{dt} \quad \omega_{im} = \frac{d\gamma}{dt} \quad \omega_m = \frac{d\omega}{dt} \quad (2.5)$$



Obr. 2.2 Blokové schéma vektorového řízení

Blok vyhodnocení polohy rotoru (BVPR) představuje periférii eQEP + SW ovladač pro měření rychlosti a polohy rotoru.

Blok vyhodnocení orientujících veličin (BVOV) je výpočet rovnice proudového modelu v RSS za účelem zjištění polohy a velikosti rotorového toku (resp. magnetizačního proudu):

$$T_r \frac{di_m^R}{dt} + i_m^R = i_1^R \quad (2.6)$$

Blok zrušení vazby (BZV) realizuje kompenzaci vzájemné vazby mezi složkami i_{1x} a i_{1y} a kompenzaci vlivu změny indukovaného napětí při změně otáček. Kompenzační signály u_{xe} a u_{ye} se přičítají k výstupům regulátorů proudu. Ruší se tím členy na pravé straně rovnic (2.1) a (2.2), čímž se tyto rovnice linearizují a zlepšují se tak chování v přechodných dějích.

Blok PWM představuje periférii ePWM + SW ovladač pro generování spínacích signálů.

V bezsenzorovém režimu jsou měřené otáčky ω_m nahrazeny estimovanými otáčkami $\hat{\omega}_m$ a výstup bloku BVOV je nahrazen estimovanými hodnotami \hat{i}_m a $\hat{\gamma}$ (přesněji $\sin \hat{\gamma}$ a $\cos \hat{\gamma}$).

2.2 Metody bezsensorového řízení

Existuje velké množství algoritmů, které se mohou uplatnit při odhadu stavových veličin nebo při identifikaci parametrů stroje. Metody bezsensorového řízení můžeme rozdělit následovně:

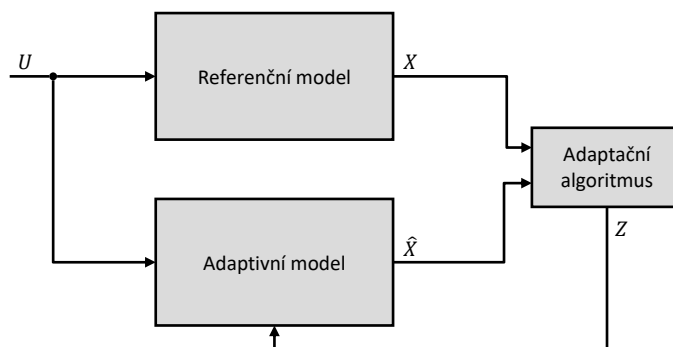
1. Metody s matematickým modelem stroje
 - a) použití matematického modelu v otevřené smyčce
 - b) použití adaptivních systémů s referenčním modelem (MRAS)
 - c) použití pozorovatelů s rekonstrukcí stavu
 - Kalmanův filtr
 - Luenbergerův pozorovatel
 - *Sliding-mode observer* (pozorovatel v klouzavém režimu)
2. Metody bez matematického modelu stroje
 - a) využití prostorové saturace statorových napětí
 - b) injektování napěťového nebo proudového signálu
 - c) měření drážkových harmonických
 - d) použití algoritmů z oblasti *soft computingu*
 - Umělé neuronové sítě
 - Genetické algoritmy
 - Fuzzy logika
 - *Support vector machines* (SVM)

Odhadci (*estimátory*) mohou být bez zpětné vazby (*Open-loop estimators*) nebo se zpětnou vazbou (*Closed-loop estimators*). Estimátory se zpětnou vazbou se označují také jako pozorovatelé. Rozdíl spočívá v tom, že pozorovatelé obsahují zpětnou vazbu zajišťující korekci odezvy estimátoru podle skutečného stavu systému. Do této skupiny patří i pozorovatelé MRAS (*Model Reference Adaptive System – Adaptivní systém s referenčním modelem*).

Princip pozorovatele MRAS je znázorněn na Obr. 2.3. V tomto systému je na základě dvou různých modelů určována stavová proměnná X . Pozorovatel se skládá z referenčního modelu, adaptivního modelu a adaptačního algoritmu. Oba modely mají společný vstup U . Referenční model vychází pouze z měřených veličin, zatímco do adaptivního modelu vstupuje i zpětnovazební signál Z představující estimovanou veličinu. Na základě výstupů obou modelů je vyhodnocen chybový signál ε . Úkolem adaptačního algoritmu je působit signálem Z tak, aby byl chybový signál minimalizován, tzn. aby odchylka mezi referenčním a adaptivním modelem byla nulová. Chybový signál může být minimalizován PI regulátorem. Minimalizovaný chybový signál ($\varepsilon = 0$) však nemusí znamenat, že odchylka bude nulová, tzn. že výstup obou modelů bude stejný. Díky zpětné vazbě je pozorovatel schopen částečně omezit vliv změn parametrů stroje.

Přesnost a vlastnosti pozorovatele ovlivňují tyto faktory:

- přesnost měření
- přesnost modelů
- volba veličiny X
- použitý adaptační algoritmus a jeho nastavení (např. nastavení PI regulátoru)



Obr. 2.3 Obecná struktura pozorovatele MRAS

Existuje několik druhů pozorovatelů MRAS, které se liší podle volby výstupní veličiny referenčního a adaptivního modelu [32]:

1. *Rotor Flux MRAS* (RF-MRAS) – rotorový tok
2. *Current Based MRAS* (CB-MRAS) – statorový proud
3. *Back Electromotive Force MRAS* (BEMF-MRAS) – vnitřní elektromotorické napětí
4. *Reactive Power MRAS* (Q-MRAS) – jalový výkon
5. *X-MRAS* – fiktivní veličina, $X = \mathbf{u}_1 \times \mathbf{i}_1$

Dále bylo navrženo nepřeberné množství systémů, které spočívají v modifikaci základních typů pozorovatele MRAS za účelem zlepšení přesnosti, zlepšení odolnosti vůči změnám parametrů stroje, dynamických vlastností a stability (např. [32] – [40]). Problematická bývá stabilita, např. pokud změna parametrů překročí určitou mez nebo při provozu v generátorickém režimu. Řeší se především algoritmy identifikace parametrů použitých v matematických modelech a různé formy adaptačního algoritmu.

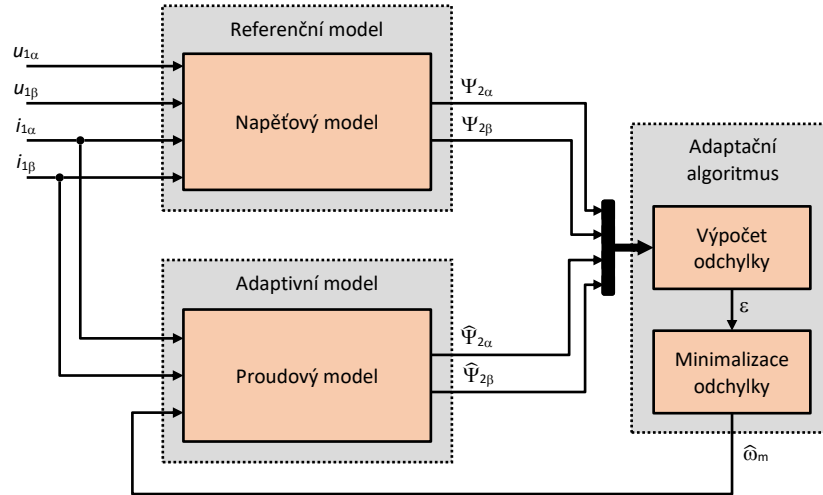
Mohou se uplatnit jak umělé neuronové sítě, tak i genetické algoritmy a fuzzy logika. Neuronové sítě mohou sloužit jako přesnější náhrada za matematický model nebo i jako náhrada za PI regulátor v adaptačním algoritmu. Genetické algoritmy jsou použitelné pro identifikaci parametrů, online adaptaci neuronové sítě nebo pro ladění parametrů PI regulátoru [IV]. Místo PI regulátoru může být použit regulátor založený na fuzzy logice. Také existují aplikace fuzzy-neuronových sítí, tyto sítě v sobě spojují vlastnosti fuzzy logiky a neuronových sítí.

Nesmíme také opomenout jeden zásadní problém, který je společný všem estimátorům, které jsou závislé na statorovém napětí. Zvláště v oblasti nízkých napájecích frekvencí je vzhledem k malé amplitudě obtížné získat přesnou hodnotu statorového napětí na svorkách motoru. Je proto potřeba se také zabývat metodami estimace statorového napětí.

Jelikož je práce z velké části zaměřena na pozorovatele RF-MRAS a CB-MRAS, je v následujících kapitolách uveden matematický popis těchto pozorovatelů.

2.3 Rotor Flux MRAS (RF-MRAS)

Struktura estimátoru mechanické úhlové rychlosti RF-MRAS je zobrazena na Obr. 2.4. Někdy se můžeme také setkat s označením *Reference Frame MRAS*. Adaptace je založena na porovnávání rotorových toků vypočítaných dvěma různými modely.



Obr. 2.4 Struktura pozorovatele RF-MRAS

Referenční model je založen na použití napěťového modelu pro rotorový tok, který je ve statorovém souřadném systému popsán touto rovnicí:

$$\Psi_2^S = \frac{L_r}{L_h} \left[\int (\mathbf{u}_1^S - R_s \mathbf{i}_1^S) dt - \frac{L_s L_r - L_h^2}{L_r} \mathbf{i}_1^S \right] \quad (2.7)$$

Pro použití v referenčním modelu je zapotřebí rovnici (2.7) vyjádřit ve složkovém tvaru:

$$\Psi_{2\alpha} = \frac{L_r}{L_h} \left[\int (u_{1\alpha} - R_s i_{1\alpha}) dt - \frac{L_s L_r - L_h^2}{L_r} i_{1\alpha} \right] \quad (2.8)$$

$$\Psi_{2\beta} = \frac{L_r}{L_h} \left[\int (u_{1\beta} - R_s i_{1\beta}) dt - \frac{L_s L_r - L_h^2}{L_r} i_{1\beta} \right] \quad (2.9)$$

Adaptivní model je založen na proudovém modelu pro rotorový tok, který je ve statorovém souřadném systému popsán takto:

$$\hat{\Psi}_2^S = \int \left[\left(j\hat{\omega}_m - \frac{1}{T_r} \right) \hat{\Psi}_2^S + \frac{1}{T_r} L_h \mathbf{i}_1^S \right] dt \quad (2.10)$$

Ve složkovém tvaru pak:

$$\hat{\Psi}_{2\alpha} = \int \left[-\frac{1}{T_r} \hat{\Psi}_{2\alpha} - \hat{\omega}_m \hat{\Psi}_{2\beta} + \frac{1}{T_r} L_h i_{1\alpha} \right] dt \quad (2.11)$$

$$\hat{\Psi}_{2\beta} = \int \left[-\frac{1}{T_r} \hat{\Psi}_{2\beta} + \hat{\omega}_m \hat{\Psi}_{2\alpha} + \frac{1}{T_r} L_h i_{1\beta} \right] dt \quad (2.12)$$

Pokud použijeme pro minimalizaci odchylky PI regulátor, je celý adaptační algoritmus vyjádřen dvěma rovnicemi:

$$\varepsilon = \hat{\Psi}_2^S \times \Psi_2^S = \hat{\Psi}_{2\alpha} \Psi_{2\beta} - \hat{\Psi}_{2\beta} \Psi_{2\alpha} \quad (2.13)$$

$$\hat{\omega}_m = K_P \varepsilon + K_I \int \varepsilon dt \quad (2.14)$$

Chybový signál ε vstupující do minimalizačního bloku je vypočítán vektorovým součinem obou toků, jeho velikost a znaménko proto závisí na úhlu, který tyto dva vektory svírají. Chybový signál bude nulový, pokud vektor toku vypočítaný referenčním modelem bude ležet v jedné přímce s vektorem toku vypočítaným adaptivním modelem. Rovnice (2.14) popisuje PI regulátor, jehož výstupem je estimovaná hodnota mechanické úhlové rychlosti $\hat{\omega}_m$.

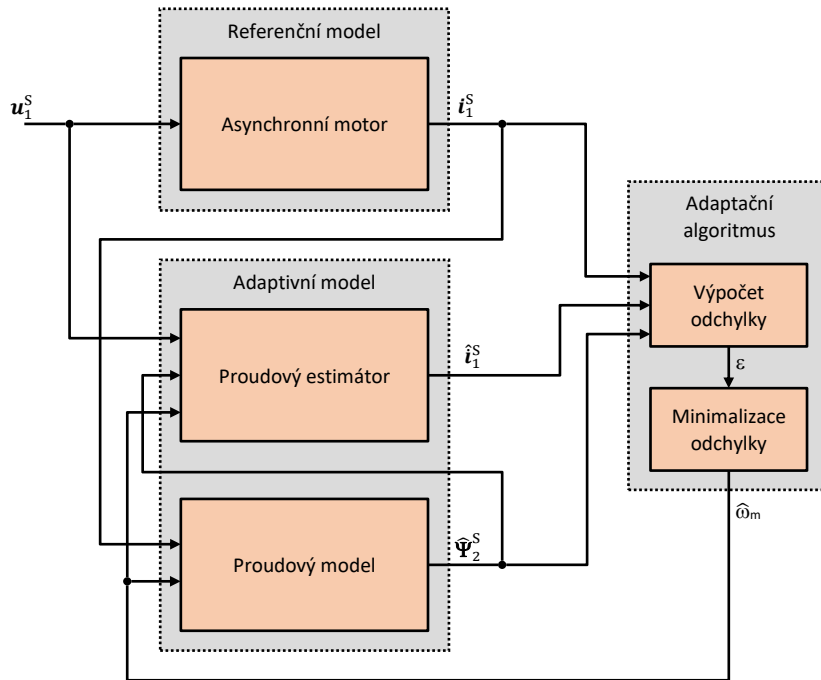
Hlavní slabinou je referenční model, který je nepřesný, protože je citlivý na chyby měření a na změny R_s . Integrátor v referenčním modelu způsobuje problémy zejména v oblasti velmi nízkých otáček (resp. frekvencí), kde se vzhledem k nízké amplitudě napětí více projevuje vliv offsetu a driftu měřených veličin a také chyba určení statorového napětí, což způsobuje značnou chybu ve výpočtu hodnoty referenčního rotorového toku. Při nízkých frekvencích je obtížné přesně určit statorové napětí na svorkách stroje, protože hodnota napětí je velmi nízká vzhledem k chybě měření.

Zlepšení vlastností pozorovatele je možné dosáhnout zavedením identifikace statorového odporu a rotorové časové konstanty. Také je možné použít kompenzaci offsetu a driftu. Hodně závisí i na způsobu určení napětí. Například v případě, že je statorové napětí získáváno výpočtem z napětí v meziobvodu u_d a referenčních signálů vstupujících do PWM modulátoru, je potřeba kompenzovat nelinearity měniče (ochranné doby, úbytky napětí).

Další z možností, jak zvýšit kvalitu pozorovatele, je využití algoritmů vycházejících z teorie umělých neuronových sítí. Rovnice referenčního modelu [39], adaptivního modelu [36][37], případně oba modely mohou být nahrazeny neuronovou sítí. To umožňuje značně snížit působení změn parametrů stroje a offsetu na přesnost estimace, protože neuronová síť vytváří přesnější model zahrnující nelinearity a neobsahuje integrátor. Další použití neuronové sítě je u této struktury možné v adaptačním mechanismu, kde může být použita pro nahrazení PI regulátoru [34].

2.4 Current Based MRAS (CB-MRAS)

Struktura estimátoru mechanické úhlové rychlosti CB-MRAS je zobrazena na Obr. 2.5. Princip je založen na porovnávání měřeného a vypočítaného statorového proudu.



Obr. 2.5 Struktura pozorovatele CB-MRAS

Referenčním modelem je přímo asynchronní motor, výstupem referenčního modelu je měřený proud \hat{i}_1^S . Adaptivní model je tvořen proudovým modelem, který slouží k výpočtu rotorového toku, a estimátorem proudu. Proudový model je popsán stejným způsobem jako u RF-MRAS (viz (2.10)). Proudový estimátor je popsán rovnicí pro vektor statorového proudu ve statorovém souřadném systému:

$$\hat{i}_1^S = \frac{1}{T_i} \int (K_1 u_1^S + K_2 \hat{\Psi}_2^S - j \hat{\omega}_m K_3 \hat{\Psi}_2^S - \hat{i}_1^S) dt \quad (2.15)$$

Ve složkovém tvaru pak:

$$\hat{i}_{1\alpha} = \frac{1}{T_i} \int [K_1 u_{1\alpha} + K_2 \hat{\Psi}_{2\alpha} + \hat{\omega}_m K_3 \hat{\Psi}_{2\beta} - \hat{i}_{1\alpha}] dt \quad (2.16)$$

$$\hat{i}_{1\beta} = \frac{1}{T_i} \int [K_1 u_{1\beta} + K_2 \hat{\Psi}_{2\beta} - \hat{\omega}_m K_3 \hat{\Psi}_{2\alpha} - \hat{i}_{1\beta}] dt \quad (2.17)$$

$$K_1 = \frac{L_r}{C_1 L_h}, \quad K_2 = \frac{L_h}{T_r L_r R_s + L_h^2}, \quad K_3 = \frac{1}{C_1}, \quad T_i = \frac{L_s L_r - L_h^2}{C_1 L_h}, \quad C_1 = \frac{L_r R_s}{L_h} + \frac{L_h}{T_r} \quad (2.18)$$

Pokud použijeme pro minimalizaci odchytky PI regulátor, je celý adaptační algoritmus vyjádřen těmito rovnicemi:

$$\varepsilon = (i_{1\alpha} - \hat{i}_{1\alpha})\hat{\Psi}_{2\beta} - (i_{1\beta} - \hat{i}_{1\beta})\hat{\Psi}_{2\alpha} \quad (2.19)$$

$$\hat{\omega}_m = K_P \varepsilon + K_I \int \varepsilon dt \quad (2.20)$$

Rovnici (2.19) lze interpretovat tak, že se jedná o vektorový součin rozdílu měřeného a estimovaného proudu s rotorovým tokem.

Výhodou je přesnost referenčního modelu. Problémy však způsobuje proudový estimátor, který je velmi závislý na parametrech stroje a zároveň na statorovém napětí [V][7]. Také se vyskytují problémy se stabilitou v generátorickém režimu. Zde se nabízí použití neuronové sítě na místě proudového estimátoru nebo proudového modelu. Příklad takovéto aplikace je v [41].

2.5 Určení statorového napětí

Pro všechny řešené aplikace byla použita stejná metoda estimace statorového napětí. Používá se žádaná hodnota statorového napětí, která je výstupem regulátorů proudu. V řídicím systému je k dispozici měřená hodnota u_d ze snímače napětí ve stejnosměrném meziobvodu měniče frekvence. Zároveň je pro řízení střídače použita digitální komparační PWM. Statorové napětí je vypočítáno podle hodnoty reference, která vstupuje do PWM modulátoru:

$$\hat{\mathbf{u}}_1 = \mathbf{u}_1^* = \mathbf{u}_{\text{ref}} \cdot \frac{u_d}{2}, \quad \mathbf{u}_{\text{ref}} \in \{\mathbf{x} \in \mathbb{C} \mid |\mathbf{x}| \leq 1\} \quad (2.21)$$

Použití měřené hodnoty u_d umožňuje kompenzovat kolísání napětí v meziobvodu. Při vyšších napětích v meziobvodu má však tato chyba, a chyba způsobená úbytky napětí, menší vliv, relativní chyba je malá. Při vyšších napětích v meziobvodu se naopak na chybě určení napětí více podílejí ochranné doby, protože pro generování stejné hodnoty statorového napětí jsou potřeba užší pulzy. Na kvalitu estimace statorového napětí může mít vliv i použitý druh spínacích prvků a kvalita budicích obvodů [VI][7].

Ochranná doba (*dead-time*) byla nastavena $t_{DT} = 1 \mu\text{s}$. Byla implementována jednoduchá metoda kompenzace ochranné doby, která spočívá v úpravě délky pulzů v závislosti na polaritě proudu v dané fázi střídače (popis převzat z [9]). Pro každou fázi je hodnota střídavy upravována připočítáním kompenzačního signálu \mathbf{u}_{comp} k hodnotě referenčního signálu \mathbf{u}_{ref} podle tohoto vztahu:

$$\mathbf{u}'_{\text{ref}} = \mathbf{u}_{\text{ref}} + \mathbf{u}_{\text{comp}} \quad (2.22)$$

$$u'_{k_{\text{ref}}} = u_{k_{\text{ref}}} + u_{k_{\text{comp}}} = u_{k_{\text{ref}}} + t_{DT} f_p \cdot \text{sgn}(i_k) \quad (2.23)$$

kde k je fáze a, b, c ; sgn je funkce signum a f_p je frekvence pulzů PWM (10 kHz).

V okolí průchodu fázového proudu nulou je potřeba provést úpravu, bez které se v průběhu proudu objevují „zuby“:

$$u'_{k_{\text{ref}}} = u_{k_{\text{ref}}} + \frac{i_k}{I_{\text{mez}}} t_{DT} f_p \cdot \text{sgn}(i_k) \quad (2.24)$$

Pokud je proud v okolí nuly, použije se vztah (2.24), jinak vztah (2.23). Zóna použití vztahu (2.24) okolo nuly je definována konstantou I_{mez} :

$$Z = \{i_k \mid -I_{\text{mez}} < i_k < I_{\text{mez}}\} \quad (2.25)$$

Hodnota I_{mez} byla určena experimentálně podle tvaru fázového proudu: 0.1 A pro $t_{DT} = 1 \mu\text{s}$ a 0,3 A pro $t_{DT} = 3,5 \mu\text{s}$ ($u_d = 200 \text{ V}$).

Kompenzace DT byla zavedena hlavně kvůli zlepšení chování pohonu (zkreslení proudu, zvlnění otáček). Vzhledem k tomu, že daná aplikace pracuje s neuronovou sítí, která je trénována podle měřené rychlosti, nemá kompenzace DT zásadní vliv na přesnost estimace. V trénovacích datech se používá napětí $\hat{\mathbf{u}}_1$, neuronová síť by měla být schopná chybu estimace napětí do jisté míry kompenzovat. Pokud se však po natrénování sítě změní nastavení DT nebo metoda kompenzace DT, je pro zachování co největší přesnosti potřeba vztah (2.21) upravit tak, aby hodnota odpovídala nastavení použitému při měření trénovacích vzorů (\mathbf{u}_{comp} – aktuální, $\mathbf{u}_{\text{comp_train}}$ – použito při měření tr. dat):

$$\hat{\mathbf{u}}_1 = (\mathbf{u}_{\text{ref}} - \mathbf{u}_{\text{comp}} + \mathbf{u}_{\text{comp_train}}) \cdot \frac{u_d}{2} \quad (2.26)$$

3 Řešené aplikace neuronových sítí

3.1 Neuronová síť jako estimátor rychlosti (ANN-EST)

Toto řešení bezsensorového systému s neuronovou sítí již bylo v minulosti na naší katedře zkoumáno [8][42]. Nová implementace tohoto řešení byla vytvořena, abychom zjistili, zda je možné dosáhnout lepších výsledků. Hlavním rozdílem je dostupnost přesnějšího systému pro měření trénovacích vzorů. Původní systém byl navíc při měření trénovacích vzorů omezen minimální vzorkovací periodou 5 ms, tím byla omezena i minimální perioda výpočtu neuronové sítě, protože se jedná o časově závislou neuronovou síť.

Estimace rychlosti je založena na použití dopředné neuronové sítě, která představuje model asynchronního motoru a která je natrénována tak, aby na svém výstupu poskytovala přímo mechanickou úhlovou rychlost asynchronního motoru. Síť je trénována offline, což znamená, že před použitím sítě nejprve musíme mít k dispozici sadu trénovacích vzorů, podle kterých se síť natrénuje, a že později během provozu není už síť dále adaptována. [VII][VIII]

Struktura použité neuronové sítě je zobrazena na Obr. 3.1. Jedná se o plně propojenou dopřednou neuronovou síť s osmi vstupy, jednou skrytou vrstvou a jedním výstupem. Počet neuronů ve skryté vrstvě byl 22, což je kompromis mezi přesností aproximace a výpočetní náročností. Pro takovouto neuronovou síť je možné použít označení 8-22-1. Větší počet neuronů zhoršuje schopnost zobecnění, tzn. nalezení obecné závislosti mezi vstupy a výstupy na základě omezeného počtu trénovacích vzorů. Stejně tak je pravděpodobnost nalezení obecné funkce nižší, pokud je použita nevhodná nebo příliš rozsáhlá sada trénovacích vzorů. Označení k a $k - 1$ vyjadřuje, že jsou vstupem neuronové sítě dva po sobě jdoucí vzorky, neuronová síť je tedy časově závislá a vzorkovací perioda vstupních dat musí být stejná jako vzorkovací perioda použitá při přípravě trénovacích vzorů. Ve skryté vrstvě je použita tansigmoidální funkce, výstupní neuron je lineární.

Funkce estimátoru v bezsensorovém systému s vektorovým řízením je znázorněna na Obr. 3.2. Hlavní částí systému je algoritmus vektorového řízení, který zajišťuje, že skutečná rychlost sleduje žádanou rychlost zadanou nadřazeným systémem (aplikace v LabVIEW). Blokové schéma vektorového řízení je na Obr. 2.2. Výstupem bloku vektorového řízení je šest signálů, které určují spínání tranzistorů ve frekvenčním měniči, který napájí asynchronní motor třífázovým napětím. Jsou měřeny tři veličiny pohonu: stejnosměrné napětí u_d v meziobvodu měniče a dva fázové proudy motoru i_a , i_b . Neuronové síti je na vstupu předkládán vektor žádané hodnoty statorového napětí \mathbf{u}_1^{S*} a vektor měřeného proudu \mathbf{i}_1^S , na základě těchto hodnot síť estimuje mechanickou úhlovou rychlost asynchronního motoru. V bezsensorovém režimu je neuronovou sítí estimovaná rychlost použita jako vstup skutečné rychlosti do regulátoru otáček, dále je integrací estimované rychlosti získána estimovaná poloha rotoru $\hat{\theta}$ a ta je zavedena do bloku BVOV, kde se určuje poloha vektoru rotorového toku a velikost magnetizačního proudu. Estimovaná hodnota magnetizačního proudu \hat{i}_m je zavedena do regulátoru magnetizačního proudu, estimovaná poloha rotorového toku $\hat{\gamma}$ je použita pro vektorové natočení ze statorového souřadného systému do orientovaného souřadného systému. Znalost polohy vektoru rotorového toku je u vektorového řízení potřebná pro rozklad statorového proudu na momentotvornou a tokotvornou složku, což umožňuje oddělenou regulaci momentu a buzení. Situace odpovídá skutečné

implementaci, na místě BVOV a integrátoru by samozřejmě mohl být použit i proudový model, jehož vstupem je mechanická úhlová rychlost.

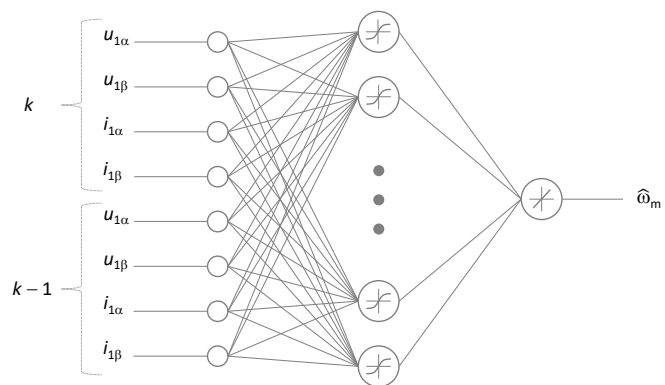
Pro trénování a testování sítě bylo použito prostředí *MATLAB* s využitím knihovny *Neural Network Toolbox*. Dále byla v jazyce C vytvořena implementace neuronové sítě, která byla zabudována do regulační struktury vektorového řízení v řídicím systému se signálovým kontrolérem TMS320F28335 od Texas Instruments.

Trénovací vzory popisují chování modelovaného systému, v tomto případě je modelovaným systémem pohon s asynchronním motorem. Jako vstupní data pro trénování neuronové sítě byly použity (viz Obr. 2.2): žádaná hodnota napětí ($u_{1\alpha}^*$, $u_{1\beta}^*$), měřený proud ($i_{1\alpha}$, $i_{1\beta}$) a měřená rychlost ω_m .

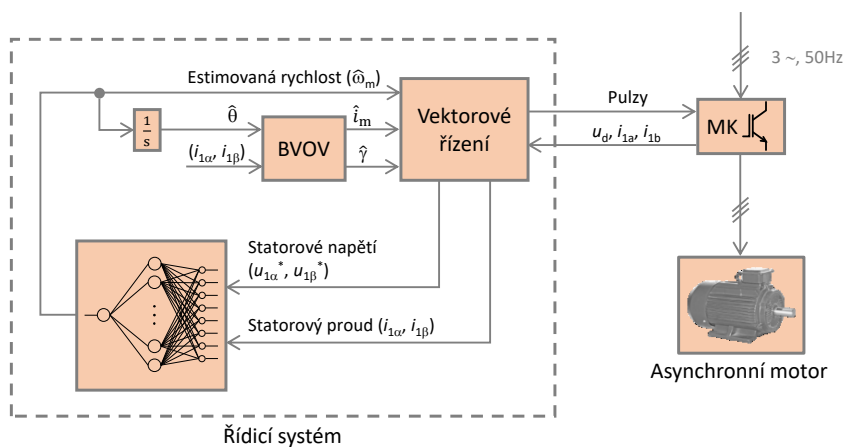
Při práci s tímto estimátorem bylo zjištěno, že neuronová síť je díky zobecnění schopna pracovat i pokud v množině trénovacích vzorů nejsou vzory změřené při zatížení (přesněji řečeno, je tam zatížení zastoupeno pouze v přechodných dějích při skocích rychlosti), přesnost při zatížení je však horší. Při pokusech o zlepšení přesnosti zahrnutím trénovacích vzorů změřených při různém zatížení se ukázalo, že se zvětšila minimální dosažitelná chyba při trénování a chování neuronové sítě se nezlepšilo. Navíc se vyskytovala ustálená odchylka estimovaných otáček. To může znamenat, že neuronová síť nemá na svém vstupu dostatek informací, k tomu, aby mohla estimovat otáčky i při změně zatížení. Jednou z možností by mohlo být zvýšení počtu vstupních vzorků, tím se ale výrazně zvyšuje výpočetní náročnost. Další možností by mohlo být přidání vstupu rotorového toku (což vyplývá z aplikace proudového estimátoru v CB-MRAS). V tomto případě je kromě zvýšení výpočetní náročnosti možné, že by došlo k ovlivnění stability systému, protože tento estimátor neobsahuje korekci výstupu podle skutečného stavu systému jako CB-MRAS. Rotorový tok je estimován na základě estimované rychlosti, čímž by se chyba estimace rychlosti mohla zesílit.

Experimentálně byla ověřována možnost zlepšení chování při změně zatížení přidáním vstupu momentotvorného proudu, čímž se struktura sítě změnila na 9-xx-1. Vliv chyby estimace rotorového toku je zde stále přítomen v podobě chyby při vektorovém natočení proudu. S touto topologií bylo možné dosáhnout srovnatelné chyby při trénování, jako u sítě 8-xx-1 trénované podle vzorů při jednom zatížení a experimentální výsledky také ukazovaly zlepšení. Bylo možné i snížit počet neuronů (9-13-1).

Zkušenosti s tímto estimátorem také ukazují, že lepších vlastností může být dosaženo použitím filtru na výstupu neuronové sítě.



Obr. 3.1 Použitá topologie neuronové sítě estimátoru rychlosti ANN-EST (8-22-1)



Obr. 3.2 Zjednodušený blokový diagram systému s estimátorem ANN-EST

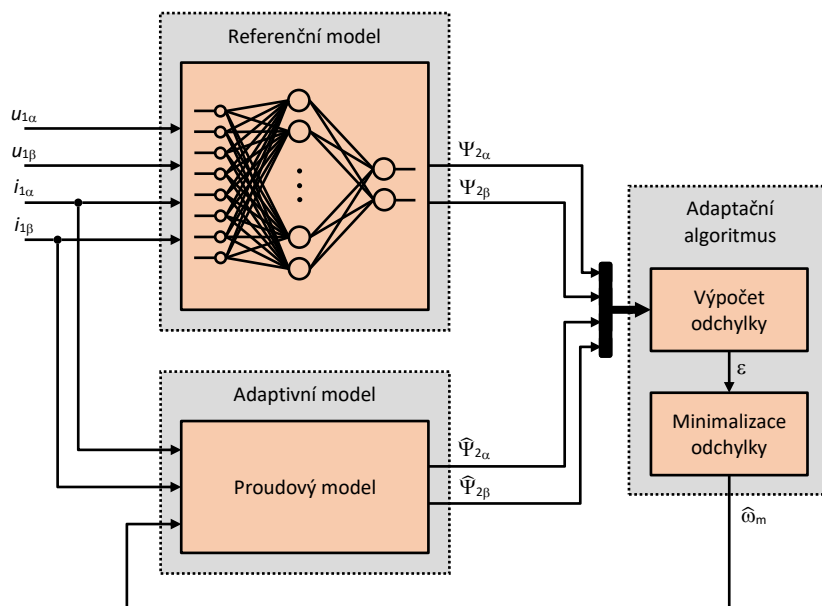
3.2 Náhrada napěťového modelu v pozorovateli RF-MRAS (ANNRF-MRAS)

Byly navrženy různé aplikace umělých neuronových sítí ve spojení s pozorovateli typu MRAS. Cílem je zlepšit vlastnosti těchto pozorovatelů. Toto řešení pozorovatele RF-MRAS s neuronovou sítí by mělo zlepšit vlastnosti pozorovatele v oblasti nízkých otáček. Neuronová síť je zde použita jako pozorovatel rotorového toku na místě referenčního modelu asynchronního motoru a nahrazuje tak napěťový model, čímž se odstraní z modelu čistá integrace a sníží se závislost na změnách statorového odporu. Neuronová síť také dokáže do jisté míry kompenzovat vliv chyby estimace statorového napětí. Použití neuronové sítě umožňuje zahrnout nelinearity pohonu. Podrobnější popis lze nalézt v [39] a 91[40][41]. [IX]

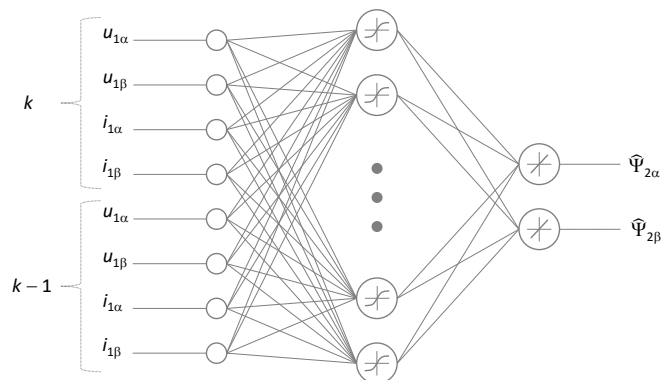
Struktura pozorovatele RF-MRAS s neuronovou sítí je znázorněna na Obr. 3.3. Na Obr. 3.5 je zjednodušené blokové schéma bezsenzorové regulační struktury s tímto pozorovatelem. Neuronové síti je na vstupu předkládán vektor žádané hodnoty statorového napětí u_1^{S*} a vektor měřeného proudu i_1^S , na základě těchto hodnot neuronová síť estimuje rotorový tok Ψ_2^S , který je výstupem referenčního modelu. Chybový signál je vyhodnocen podle rovnice (2.13). Pro minimalizaci chybového signálu je použit PI regulátor. V bezsenzorovém režimu je pozorovatelem estimovaná rychlost použita jako vstup skutečné rychlosti do regulátoru otáček. Výstup BVOV je nahrazen výstupem proudového modelu, který je adaptivním model pozorovatele. Modul vektoru rotorového toku je přes L_h přepočítán na magnetizační proud.

Struktura použité neuronové sítě je zobrazena na Obr. 3.4. Je to plně propojená dopředná neuronová síť s osmi vstupy, jednou skrytou vrstvou a dvěma výstupy, kterými jsou složky rotorového toku ve statorovém systému souřadnic. Byla použita síť 8-12-2, počet neuronů ve skryté vrstvě byl tedy 12. Při dalším zvyšování počtu neuronů skryté vrstvy se už výrazně nezlepšovala přesnost výstupu poskytovaného sítí. Označení k a $k - 1$ vyjadřuje, že jsou vstupem neuronové sítě dva po sobě jdoucí vzorky, to dělá neuronovou síť časově závislou. Vzorkovací perioda vstupních dat musí být stejná jako vzorkovací perioda použitá při přípravě trénovacích vzorů. Ve skryté vrstvě je použita tansigmoidální funkce, výstupní neuron je lineární.

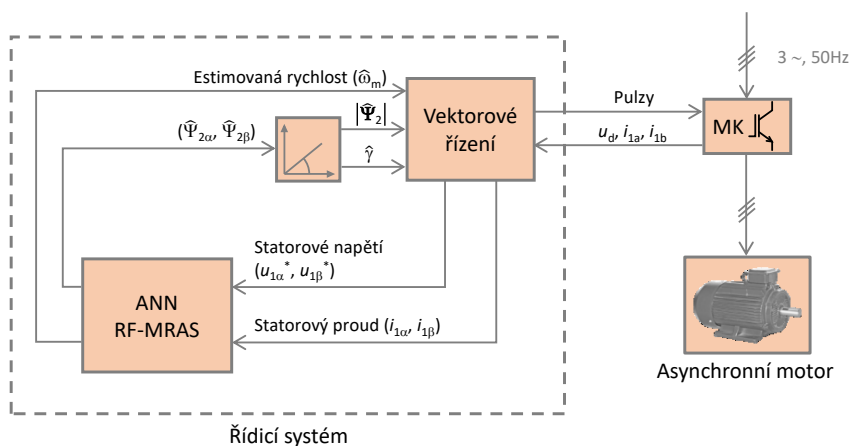
Jako vstupní data pro trénování neuronové sítě byly použity (viz Obr. 2.2): žádaná hodnota napětí $(u_{1\alpha}^*, u_{1\beta}^*)$, měřený proud $(i_{1\alpha}, i_{1\beta})$ a rotorový tok $(\Psi_{2\alpha}, \Psi_{2\beta})$ (přepočet z výstupu BVOV).



Obr. 3.3 Struktura pozorovatele ANNRF-MRAS



Obr. 3.4 Použitá topologie neuronové sítě estimátoru rotorového toku (8-12-2)



Obr. 3.5 Zjednodušený blokový diagram systému s pozorovatelem ANNRF-MRAS

3.3 Náhrada proudového estimátoru v pozorovateli CB-MRAS (ANN-CB-MRAS $\alpha\beta$)

Hlavní myšlenkou, která vedla k vytvoření tohoto pozorovatele, bylo to, že je matematický model proudového estimátoru ve skutečnosti nepřesný a hodně závislý na parametrech asynchronního motoru. Nepřesnost matematického modelu spočívá v tom, že je zjednodušující, zanedbává reálné vlastnosti asynchronního motoru. V oblasti nízkých otáček se navíc projevuje chyba estimace statorového napětí.

V [41] je použita na místě proudového estimátoru tzv. online neuronová síť, jak bylo ukázáno, použití této sítě přináší určité zlepšení ve srovnání se základním CB-MRAS. V rámci této práce byla tato metoda zkoušena, nicméně zlepšení není až tak výrazné. Hlavní rozdíl byl v tom, že pozorovatel byl schopen provozu se jmenovitými parametry (Tab. 5.1) jako samostatný blok i jako náhrada čidla otáček v bezsenzorovém režimu, základní CB-MRAS s těmito parametry vykazoval poměrně velkou chybu a nebyl schopen provozu v bezsenzorovém režimu. Navíc, jak autoři ukázali, i s online trénovanou sítí existují problémy v generátorickém režimu. Tyto problémy autoři vyřešili offline trénovanou neuronovou sítí, která nahradila proudový model. Použitá topologie sítě určovala rotorový tok z napětí a proudu (stejný model, který lze použít jako náhradu napěťového modelu v RF-MRAS), čímž byla odstraněna závislost na otáčkách. To má mít za následek zlepšení stability v generátorickém režimu.

Jako online trénovaná síť se většinou označuje výpočetní model neuronové sítě, jejíž struktura je odrazem matematického modelu. Nejedná se tedy o neuronovou síť v pravém slova smyslu, spíš jde o to, že se na danou výpočetní strukturu díváme z pohledu teorie neuronových sítí. Hlavní rozdíl je v použitém adaptačním algoritmu, jehož odvození vychází z metody *backpropagation*. Výhodou online trénované neuronové sítě je to, že se adaptuje v reálném čase, takže není potřeba předchozí fáze trénování. Další výhodou by mohlo být to, že pro trénování není potřeba znát měřené otáčky. Stále je zde však stejná závislost na parametrech stroje, protože se jedná v podstatě o diferenciální rovnice matematického modelu proudového estimátoru. Zlepšení pravděpodobně spočívá v použitém adaptačním algoritmu, který nahradil PI regulátor.

Řešení popisované v této kapitole se proto zaměřuje na použití offline trénované neuronové sítě pro vytvoření přesnějšího modelu proudového estimátoru. Struktura tohoto pozorovatele je znázorněna na Obr. 3.6. Rozdíl oproti základnímu CB-MRAS je v tom, že rovnice proudového estimátoru (2.16) a (2.17) jsou nahrazeny neuronovou sítí. Na Obr. 3.8 je znázorněno zapojení pozorovatele do systému s vektorovým řízením v bezsenzorovém režimu.

Návrh topologie vycházel ze zkušeností s předchozími aplikacemi a ze závislostí, které jsou dány matematickým modelem. Byly použity stejné vstupní, veličiny jako má matematický model. Nejlepší výsledky byly získány s neuronovou sítí v konfiguraci 10-24-2 (10 vstupů, jedna skrytá vrstva, 24 neuronů ve skryté vrstvě, 2 výstupní neurony), která je na Obr. 3.7. Byla použita síť s časovým zpožděním vstupů, to znamená, že na vstup jsou zároveň přiváděny dva po sobě jdoucí vzorky k a $k - 1$, to zlepšuje přesnost neuronové sítě. Toto časové zpoždění dělá neuronovou síť časově závislou, při implementaci takovéto sítě je potřeba dbát na to, aby vzorkovací perioda vstupních dat byla stejná jako vzorkovací perioda použitá při přípravě trénovacích vzorů. Dobré výsledky dávaly i síť 7-xx-2 (pouze vstupy toků časově zpožděné), 5-xx-2 (bez časového zpoždění). Ve skryté vrstvě je opět použita tansigmoidální aktivační funkce (Obr. 1.4), výstupní aktivační funkce je lineární. Byla vyzkoušena i

varianta dvou samostatných sítí, jedna síť estimovala složku α a druhá složku β , ale nebyl tam žádný přínos ve srovnání se sítěmi s dvěma výstupními neurony.

Jako vstupní data pro trénování neuronové sítě byly použity (viz Obr. 2.2): žádaná hodnota napětí $(u_{1\alpha}^*, u_{1\beta}^*)$, měřená rychlost ω_m , rotorový tok $(\Psi_{2\alpha}, \Psi_{2\beta})$ (přepočet z výstupu BVOV) a měřený proud $(i_{1\alpha}, i_{1\beta})$.

Cesta k fungujícímu pozorovateli byla o něco delší než u předchozích dvou aplikací. Při prvních pokusech pozorovatel s neuronovou sítí nefungoval, přestože při trénování byla chyba nízká a při použití mimo CB-MRAS (vstupy rychlosti a toku byly odvozeny z měřených otáček) neuronová síť estimovala proud s vysokou přesností. Ukázalo se, že pro použití v CB-MRAS musí být model obecnější, to pravděpodobně souvisí s tím, že má síť fungovat jako adaptivní model, to znamená, že je laděná estimovanými otáčkami. Hlavní problém byl v tom, že použitá data neobsahovala zatížení. Předchozí dva systémy byly v určitém rozsahu schopny stabilního provozu i v případě, že byla použita trénovací data změřená bez zatížení, v takovém případě se při zatížení zvyšovala chyba estimace.

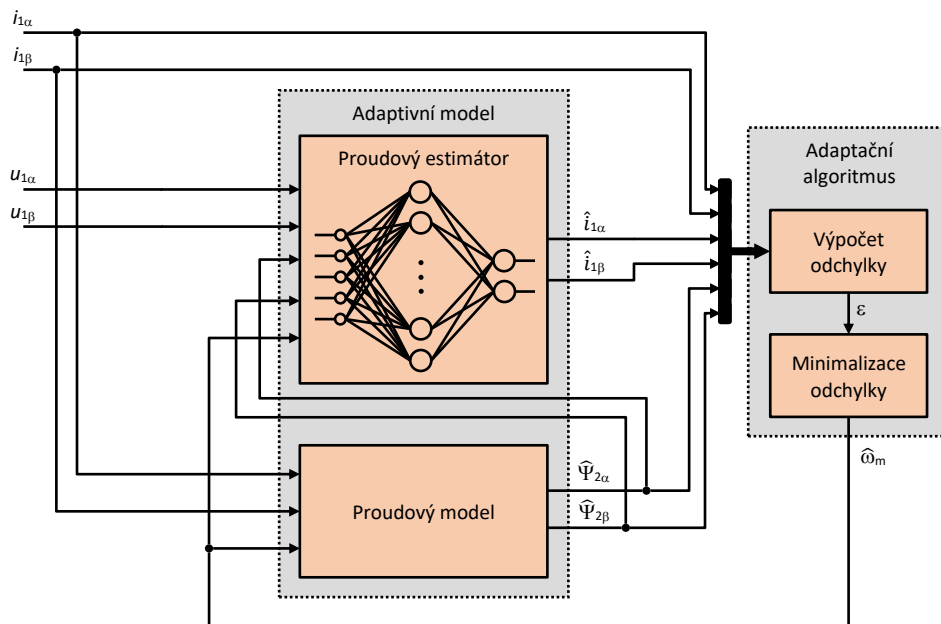
Z toho důvodu byly natrénované sítě později testovány i tak, že se v *Matlabu* simulovalo jejich chování při použití v CB-MRAS, vstupem pozorovatele byly časové průběhy získané při měření trénovacích dat. Počáteční podmínky integrátorů proudového modelu a PI regulátoru musely být nastaveny podle hodnot na počátku časových průběhů.

Při testování neuronových sítí v řídicím systému se ukázala jako užitečná funkce „Nulování CB-MRAS“, která spočívala v tom, že přes uživatelské rozhraní bylo možné ručně vyvolat inicializaci integrátorů proudového modelu a PI regulátoru hodnotami vyhodnocenými podle výstupu inkrementálního snímače. Teprve díky této funkci se podařilo tento pozorovatel poprvé zprovoznit. Umožnilo to totiž pracovat i se špatně natrénovanými neuronovými sítěmi, které způsobovaly nestabilitu například při nulové žádané rychlosti nebo při rozběhu. Bylo možné nejdříve provést rozběh a teprve poté přepnout do bezsenzorového režimu (zavazbení estimované rychlosti a rotorového toku do vektorového řízení). Mnohé sítě také fungovaly špatně při provozu pozorovatele v otevřené smyčce (větší chyba, nestabilita), ale v bezsenzorovém režimu byl pozorovatel stabilní.

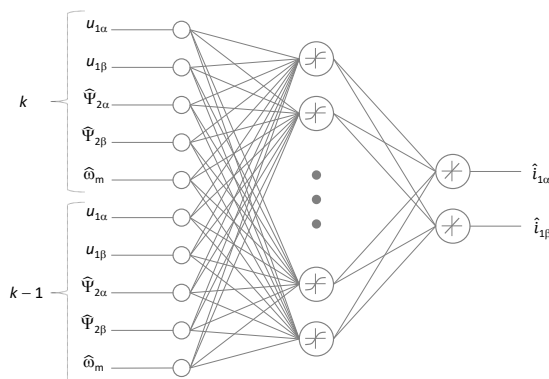
Dále se ukázalo, že je potřeba nastavit PI regulátor tak, aby byl dostatečně rychlý. Pro použitý experimentální pohon se osvědčilo nastavení $K_R = 1$ a $T_R = 0,5$ až 1 ms. Příliš velké zesílení a malá časová konstanta vede k zesílení šumu nebo rozkmitání. Příliš velká časová konstanta vede k nestabilitě při reverzaci.

Pokud má pozorovatel fungovat přesně i při změně buzení, musí to být zahrnuto v trénovacích datech. Pokud byly neuronové sítě trénovány na datech při různém zatížení, ale jen pro jednu hodnotu magnetizačního proudu ($i_m^* = 3$ A), projevovala se změna odbuzení zvýšením chyby estimace, při větší změně nestabilitou.

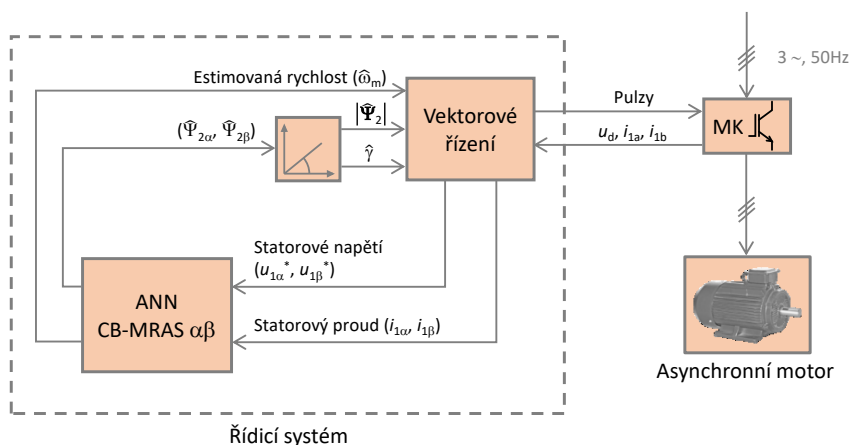
Později, s lépe sestavenými trénovacími vzory a s dokonalejším vyhodnocováním zobecnění neuronové sítě, se podařilo nedostatky překonat a natrénovat sítě, které zvládaly rozběh i v bezsenzorovém režimu. U takových sítí pak většinou nebyl velký rozdíl v přesnosti a stabilitě pozorovatele mezi provozem v otevřené smyčce a v bezsenzorovém režimu. A i pokud došlo například k nestabilitě při reverzaci vlivem špatného nastavení PI regulátoru, tak při změně žádaných otáček na původní hodnotu pozorovatel opět začal estimovat správně.



Obr. 3.6 Struktura pozorovatele ANNCB-MRAS s výpočtem proudu v SSS



Obr. 3.7 Použitá topologie neuronové sítě proudového estimátoru ANNCB-MRAS $\alpha\beta$ (10-24-2)



Obr. 3.8 Zjednodušený blokový diagram systému s pozorovatelem ANNCB-MRAS $\alpha\beta$

3.4 CB-MRAS s výpočtem proudu v orientovaných souřadnicích (ANN-CB-MRAS xy)

Princip tohoto pozorovatele je znázorněn na Obr. 3.10. Návrh vychází z vektorového natočení rovnice proudového estimátoru (2.15) do rotujícího souřadného systému orientovaného na rotorový tok, jehož poloha vůči stacionárnímu souřadnému systému je dána úhlem γ – viz Obr. 3.9. Rovnice proudového estimátoru lze upravit následovně, převedeme rovnici (2.15) na diferenciální rovnici:

$$T_i \frac{d\hat{i}_1^S}{dt} + \hat{i}_1^S = K_1 \mathbf{u}_1^S + K_2 \hat{\Psi}_2^S - j\hat{\omega}_m K_3 \hat{\Psi}_2^S \quad (3.1)$$

Vyjádříme veličiny ve statorovém souřadném systému pomocí veličin v orientovaném souřadném systému:

$$\mathbf{x}^S = \mathbf{x}^O e^{j\gamma} \quad (3.2)$$

$$T_i \frac{d(\hat{i}_1^O e^{j\gamma})}{dt} + \hat{i}_1^O e^{j\gamma} = K_1 \mathbf{u}_1^O e^{j\gamma} + K_2 \hat{\Psi}_2^O e^{j\gamma} - j\hat{\omega}_m K_3 \hat{\Psi}_2^O e^{j\gamma} \quad (3.3)$$

Provedeme derivaci v prvním členu na levé straně rovnice:

$$T_i \left(\frac{d\hat{i}_1^O}{dt} e^{j\gamma} + j\omega_{im} \hat{i}_1^O e^{j\gamma} \right) + \hat{i}_1^O e^{j\gamma} = K_1 \mathbf{u}_1^O e^{j\gamma} + K_2 \hat{\Psi}_2^O e^{j\gamma} - j\hat{\omega}_m K_3 \hat{\Psi}_2^O e^{j\gamma} \quad (3.4)$$

Po úpravě dostaneme:

$$T_i \frac{d\hat{i}_1^O}{dt} + \hat{i}_1^O = K_1 \mathbf{u}_1^O + K_2 \hat{\Psi}_2^O - j\hat{\omega}_m K_3 \hat{\Psi}_2^O - j\omega_{im} T_i \hat{i}_1^O \quad (3.5)$$

Ve složkovém tvaru pak:

$$T_i \frac{d\hat{i}_{1x}}{dt} + \hat{i}_{1x} = K_1 u_{1x} + K_2 \hat{\Psi}_{2x} + \hat{\omega}_m K_3 \hat{\Psi}_{2y} + \omega_{im} T_i \hat{i}_{1y} \quad (3.6)$$

$$T_i \frac{d\hat{i}_{1y}}{dt} + \hat{i}_{1y} = K_1 u_{1y} + K_2 \hat{\Psi}_{2y} - \hat{\omega}_m K_3 \hat{\Psi}_{2x} - \omega_{im} T_i \hat{i}_{1x} \quad (3.7)$$

Rovnice pro jednotlivé složky lze dále upravit (v OSS je složka Ψ_{2y} nulová):

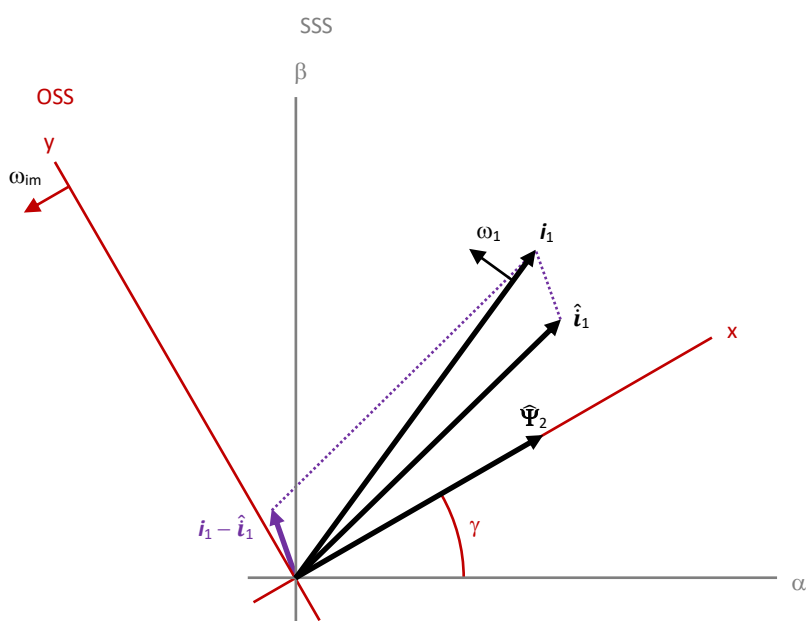
$$T_i \frac{d\hat{i}_{1x}}{dt} + \hat{i}_{1x} = K_1 u_{1x} + K_2 |\hat{\Psi}_2| + \omega_{im} T_i \hat{i}_{1y} \quad (3.8)$$

$$T_i \frac{d\hat{i}_{1y}}{dt} + \hat{i}_{1y} = K_1 u_{1y} - \hat{\omega}_m K_3 |\hat{\Psi}_2| - \omega_{im} T_i \hat{i}_{1x} \quad (3.9)$$

Výpočet chybového signálu ε je nezávislý na volbě souřadné soustavy, protože se jedná o výsledek vektorového součinu vektoru představujícího odchylku proudů $\hat{i}_1 - i_1$ a vektoru rotorového toku $\hat{\Psi}_2$. Na základě vztahu (2.19) jej lze proto chápat jako složku ε_z vektoru ε , který je kolmý k oběma souřadným soustavám, jeho velikost závisí pouze na velikosti obou vektorů a úhlu, který svírají. Jeho směr závisí na vzájemné poloze obou vektorů, pro situaci na Obr. 3.9 by bylo $\varepsilon < 0$. Rovnice pro výpočet chybového signálu vypadá takto:

$$\varepsilon = (\varepsilon_\alpha, \varepsilon_\beta, \varepsilon_z) = (\varepsilon_x, \varepsilon_y, \varepsilon_z) = (0, 0, \varepsilon) \quad (3.10)$$

$$\varepsilon = (i_{1x} - \hat{i}_{1x})\hat{\Psi}_{2y} - (i_{1y} - \hat{i}_{1y})\hat{\Psi}_{2x} = (\hat{i}_{1y} - i_{1y}) \cdot |\hat{\Psi}_2| \quad (3.11)$$



Obr. 3.9 Vektorový diagram CB-MRAS

Z rovnice (3.11) vyplývá, že stačí estimovat momentotvornou složku statorového proudu i_{1y} . To samé naznačují i rovnice (3.8) a (3.9), na mechanické úhlové rychlosti je závislá pouze rovnice (3.9). Díky tomu, a díky stejnosměrné povaze veličin v OSS, se návrh neuronové sítě zjednoduší.

Původní myšlenka byla vedena právě možným zjednodušením návrhu a trénování neuronové sítě při použití veličin v souřadném systému orientovaném na rotorový tok. První funkční varianta tohoto pozorovatele pracovala tak, že byly estimovány obě složky statorového proudu \hat{i}_{1x} a \hat{i}_{1y} . Provedlo se vektorové natočení vstupních veličin proudového estimátoru do OSS, byl proveden výpočet estimace neuronovou sítí a poté byl estimovaný vektor statorového proudu natočen zpět do SSS. Následně byl aplikován stejný algoritmus adaptace jako u pozorovatele ANNCB-MRAS $\alpha\beta$. Vstupní veličiny neuronové sítě byly stejné jako u sítě na Obr. 3.11. Pracovalo se se sítí 8-6-2, která měla časové zpoždění vstupů, později se ukázalo, že časové zpoždění nemá velký vliv na zlepšení přesnosti, a proto byl snížen počet vstupů na čtyři, což znamená, že struktura sítě byla 4-6-2.

Později se ukázalo, že estimace \hat{i}_{1x} a vektorové natočení zpět do SSS není nutné. Při trénování a testování neuronových sítí bylo pozorováno, že chyba estimace \hat{i}_{1x} je ve srovnání s chybou estimace \hat{i}_{1y} minimální. Také chyba estimace otáček měla na \hat{i}_{1x} minimální vliv. To vedlo k návrhu systému

s pozorovatelem, jehož struktura je zobrazena na Obr. 3.10. Je estimován pouze proud \hat{i}_{1y} , pro vektorové natočení vstupních veličin se používá poloha vektoru rotorového toku, který je výstupem proudového modelu. Použitá neuronová síť je Obr. 3.11. Na Obr. 3.8 je znázorněno zapojení pozorovatele do systému s vektorovým řízením v bezsensorovém režimu.

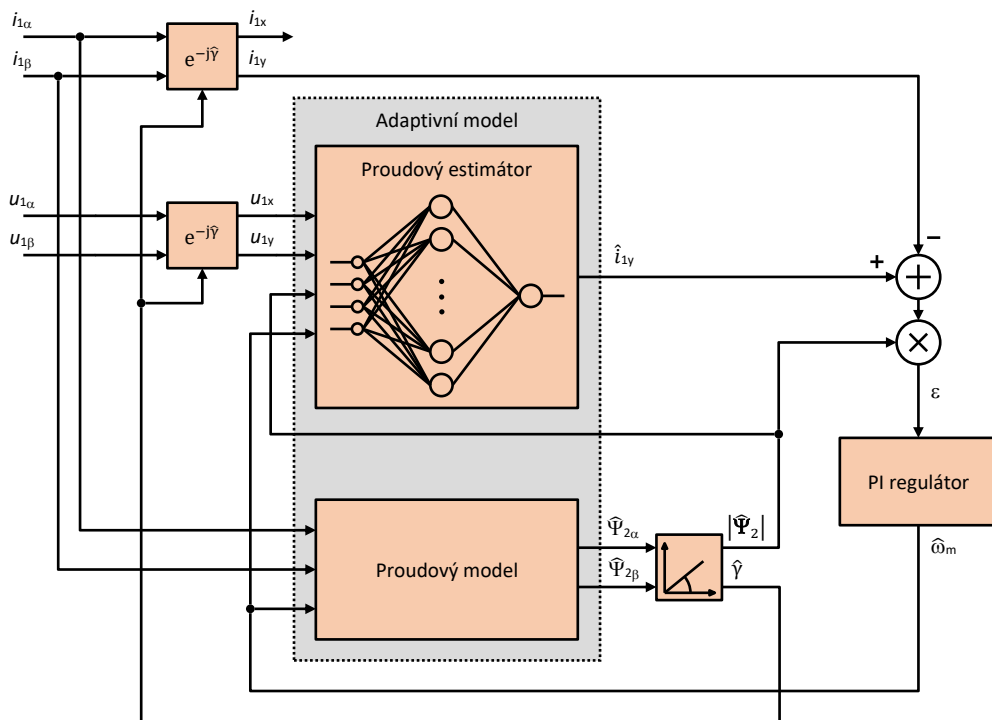
Experimentální výsledky byly získány se sítí 4-4-1, tato síť byla trénována podle dat, které byly změřeny při jedné hodnotě magnetizačního proudu. Neuronovou síť lze natrénovat i pro různé hodnoty buzení, to je zahrnuto prostřednictvím vstupu $|\hat{\Psi}_2|$. V takovém případě je potřeba zvýšit počet neuronů ve skryté vrstvě a přidat trénovací vzory změřené pro různé hodnoty magnetizačního proudu. Podařilo se natrénovat síť 4-18-1, která fungovala pro hodnoty magnetizačního proudu 2 až 3 A. Pro konstantní hodnotu buzení lze použít i síť 3-xx-1, vstupem jsou pouze napětí a otáčky, vstup rotorového toku je vynechán.

Jako vstupní data pro trénování neuronové sítě byly použity (viz Obr. 2.2): žádaná hodnota napětí (u_{1x}^*, u_{1y}^*) , měřená rychlost ω_m , modul rotorového toku $|\hat{\Psi}_2|$ (přepočten z výstupu BVOV) a měřený proud (i_{1x}, i_{1y}) .

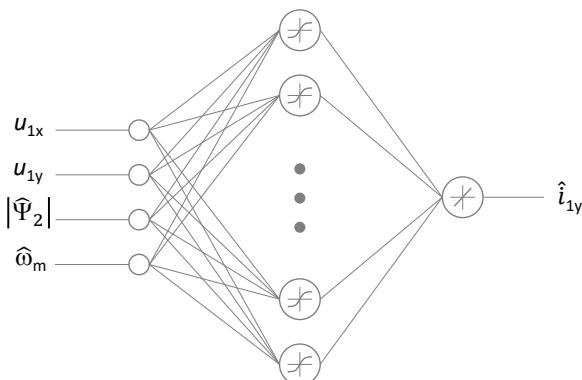
Chování při různém nastavení PI regulátoru bylo stejné jako u pozorovatele s výpočtem v $\alpha\beta$. Pro použitý experimentální pohon se osvědčilo nastavení $K_R = 1$ a $T_R = 0,5$ až 1 ms.

Při prvních testech s reálným systémem bylo pozorováno, že při provozu v otevřené smyčce byl pozorovatel nestabilní, ale v bezsensorovém režimu, po zavedení estimovaných veličin do vektorového řízení, fungoval naopak velmi dobře. Ukázalo se, že to bylo způsobeno tím, že v režimu se snímačem otáček nebyl pro vektorové natočení vstupů pozorovatele použit výstup proudového modelu pozorovatele, ale výstup BVOV.

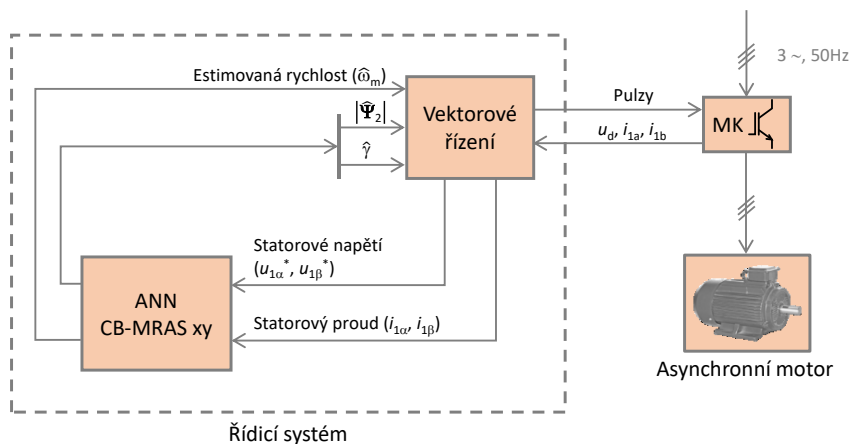
U tohoto pozorovatele byly provedeny i pokusy s rekurentní neuronovou sítí. Byla použita rekurentní síť s vnější zpětnou vazbou. Testovaná neuronová síť měla strukturu 5-4-1, pátým vstupem je výstup neuronové sítě z předchozího kroku $\hat{i}_{1y}(k-1)$. Dopředná neuronová síť je statická, rekurentní sítě jsou dynamické, mají vnitřní stav, výstup závisí na aktuálním vstupu a na stavu, který si pamatují z předchozího výpočetního kroku. Rekurentní sítě jsou proto schopny lépe modelovat dynamické děje. Podařilo se natrénovat funkční proudový estimátor, který dobře estimoval proud, výsledky pozorovatele s rekurentní sítí však nebyly tak dobré. Pozorovatel s rekurentní sítí měl problémy se stabilitou.



Obr. 3.10 Struktura pozorovatele ANNCB-MRAS s výpočtem proudu v OSS



Obr. 3.11 Použitá topologie neuronové sítě proudového estimátoru ANNCB-MRAS xy (4-4-1)



Obr. 3.12 Zjednodušený blokový diagram systému s pozorovatelem ANNCB-MRAS xy

4 Výsledky simulací

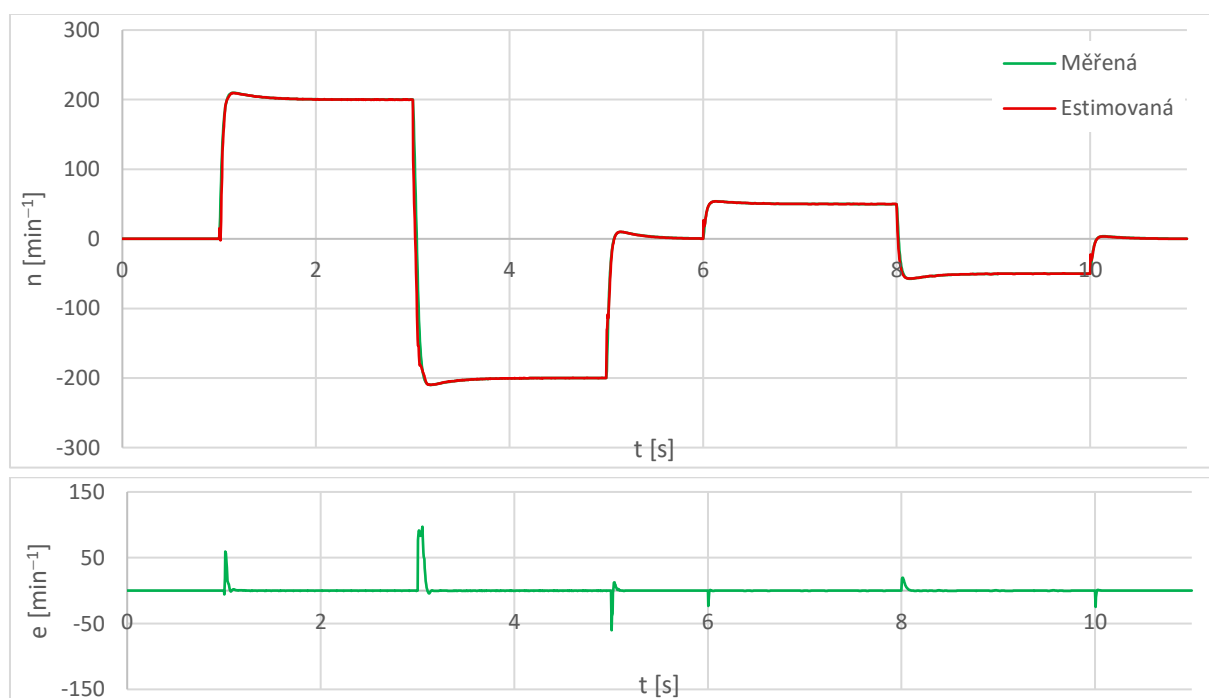
Použitý simulační model byl s matematickým modelem asynchronního motoru bez uvažování mechanických ztrát, ztrát v železe a nasycení magnetického obvodu. Také byl uvažován ideální napájecí zdroj bez modelování spínacích dějů. Simulované průběhy byly získány pro stejné rychlosti a se stejnou topologií neuronové sítě, jako průběhy získané měřením na skutečném pohonu. Také byla snaha dodržet stejný postup získání a přípravy trénovacích vzorů. Trénovací data byla získávána simulací chování pohonu v režimu se snímačem otáček. Byla uložena trénovací data pro zadanou sekvenci skoků rychlosti v rozsahu od -300 do $300 \text{ ot}\cdot\text{min}^{-1}$ a pro hodnoty momentu zátěže v rozsahu od -9 do 9 Nm .

Byly vykresleny průběhy veličin pro sekvenci skoků žádané rychlosti bez zatížení a pro konstantní žádanou rychlost se zatížením. Pro porovnání přesnosti estimace byly vykresleny průběhy odchylky měřených a estimovaných otáček:

$$e = n - \hat{n} \quad (4.1)$$

4.1 Estimátor ANN-EST

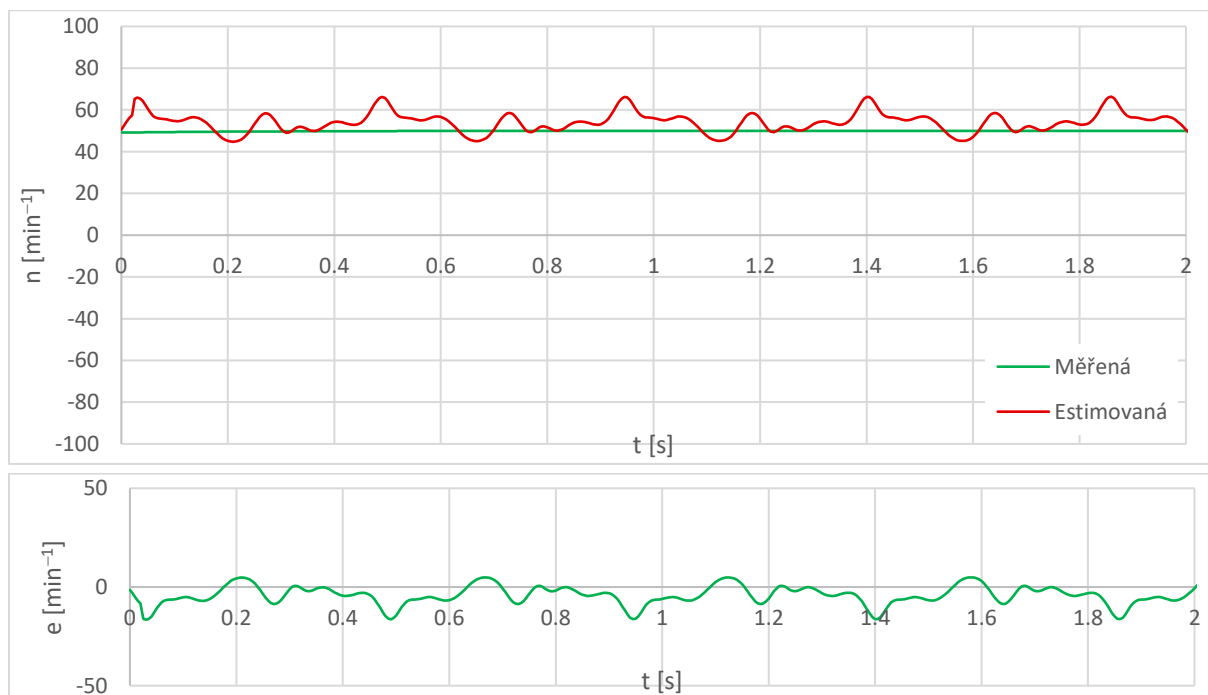
Na Obr. 4.1 je průběh měřené a estimované rychlosti pro skoky žádané rychlosti 50 a $200 \text{ ot}\cdot\text{min}^{-1}$. Na Obr. 4.2 je průběh pro $200 \text{ ot}\cdot\text{min}^{-1}$ a na Obr. 4.3 pro $50 \text{ ot}\cdot\text{min}^{-1}$ a 20% jmenovitého momentu zátěže. Byly použity trénovací vzory bez zatížení.



Obr. 4.1 Simulace ANN-EST: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 200 a $\pm 50 \text{ min}^{-1}$, bez zátěže



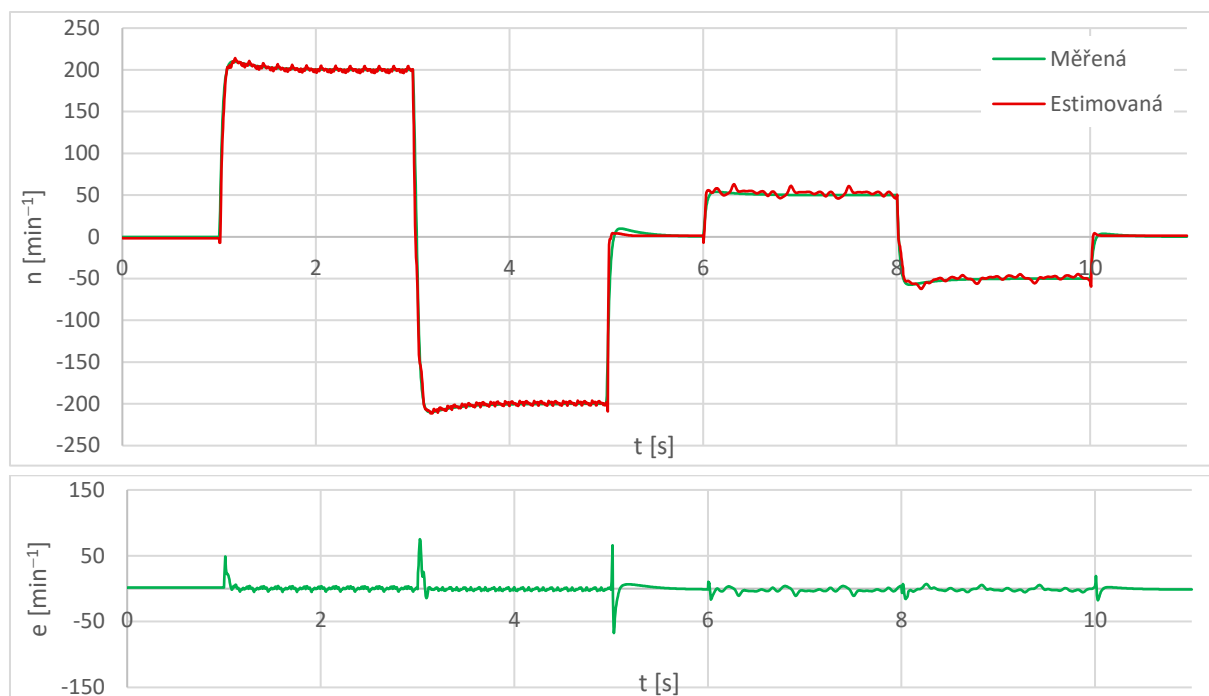
Obr. 4.2 Simulace ANN-EST: měřená a estimovaná rychlost motoru, zatížení pro 200 min^{-1} , $20 \% M_n$



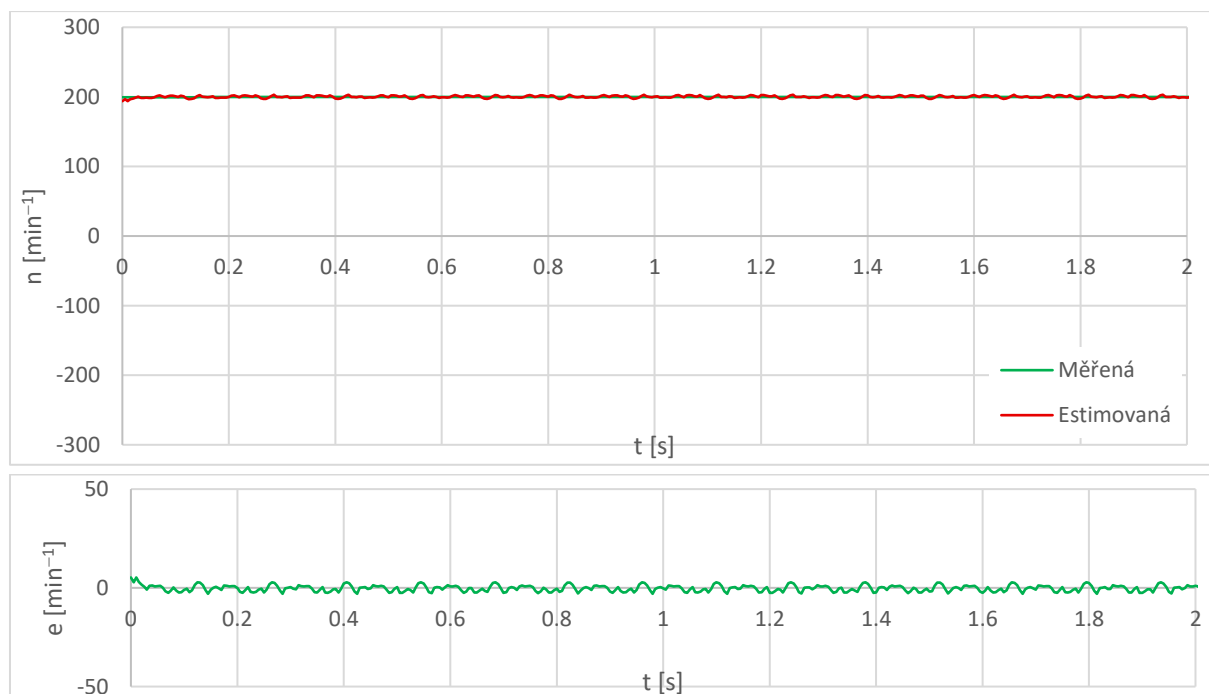
Obr. 4.3 Simulace ANN-EST: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , $20 \% M_n$

4.2 Pozorovatel ANNRF-MRAS

Na Obr. 4.4 je průběh měřené a estimované rychlosti pro skoky žádané rychlosti 50 a 200 ot·min⁻¹. Na Obr. 4.5 je průběh pro 200 ot·min⁻¹ a na Obr. 4.6 pro 50 ot·min⁻¹ a 20 % jmenovitého momentu zátěže.



Obr. 4.4 Simulace ANNRF-MRAS: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 200 a $\pm 50 \text{ min}^{-1}$, bez zátěže



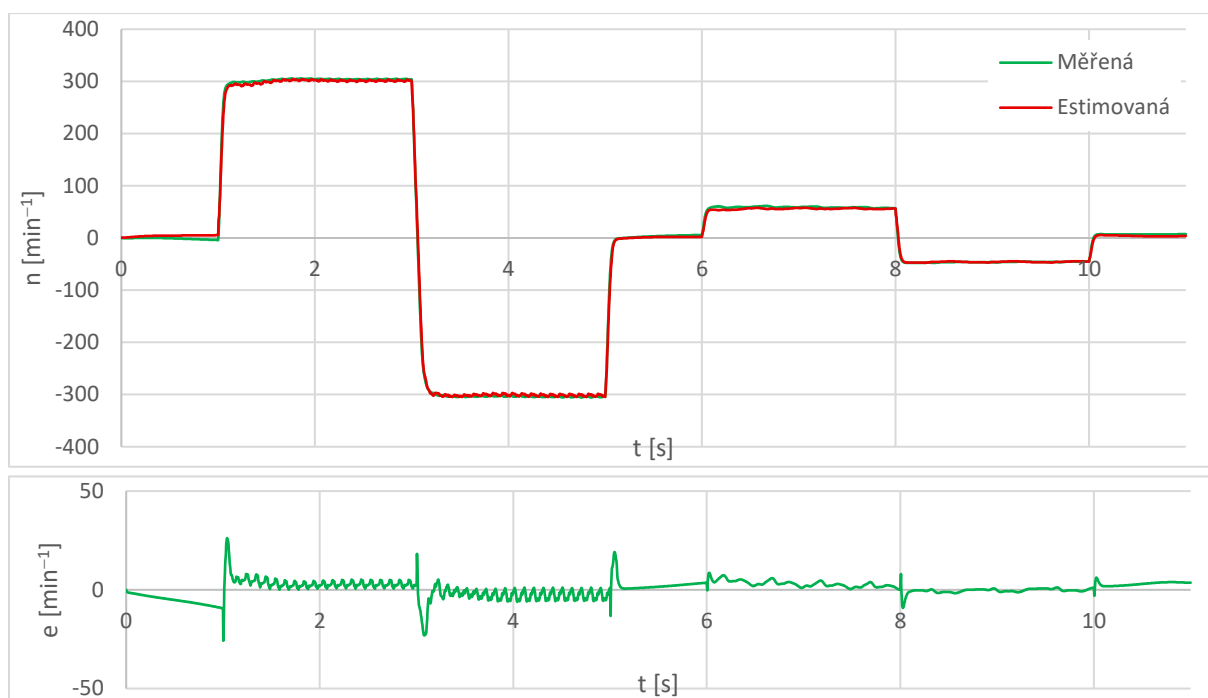
Obr. 4.5 Simulace ANNRF-MRAS: měřená a estimovaná rychlost motoru, zatížení pro 200 min^{-1} , 20 % M_n



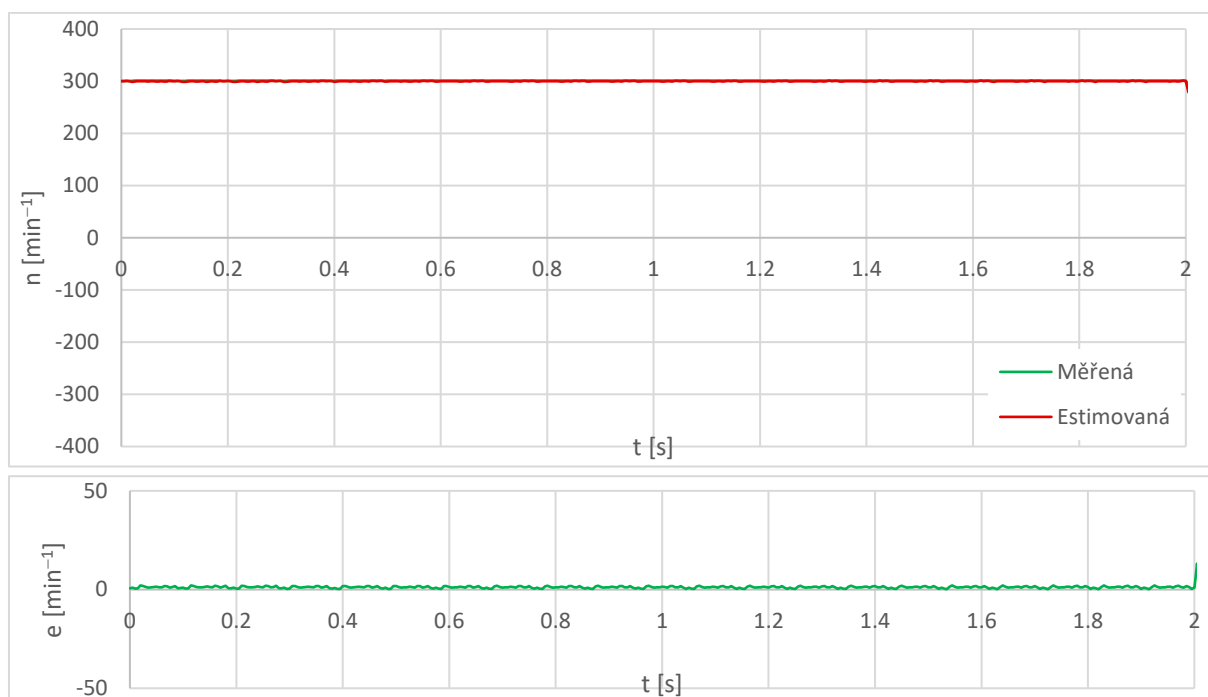
Obr. 4.6 Simulace ANNRF-MRAS: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , $20 \% M_n$

4.3 Pozorovatel ANNCF-MRAS $\alpha\beta$

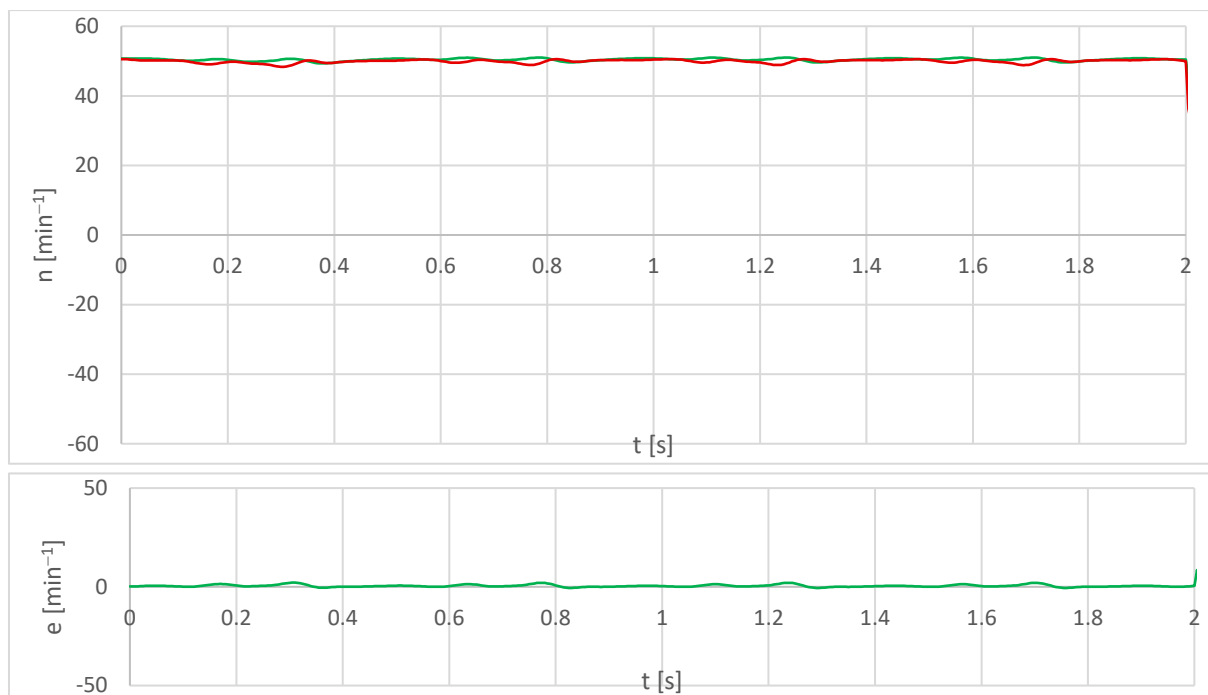
Simulované průběhy byly získány při stejném nastavení a se stejnou topologií sítě, jako průběhy získané měřením na skutečném pohonu (viz Tab. 6.1). Také byla snaha dodržet stejný postup získání a přípravy trénovacích vzorů. Byly použity trénovací vzory se zatížením. Obr. 4.7 obsahuje průběh měřené a estimované rychlosti pro sekvenci skoků žádané rychlosti $n^* = 0; 300; -300; 0; 50; -50; 0 \text{ ot} \cdot \text{min}^{-1}$ ($t = 0; 1; 3; 5; 6; 8; 10 \text{ s}$). Obr. 4.8 a Obr. 4.9 zobrazují průběhy při 20% jmenovitého zatížení pro žádanou rychlost 300 a $50 \text{ ot} \cdot \text{min}^{-1}$. Při srovnání průběhů si lze povšimnout, že přesnost je vyšší při zatížení.



Obr. 4.7 Simulace ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 300 a $\pm 50 \text{ min}^{-1}$, bez zátěže



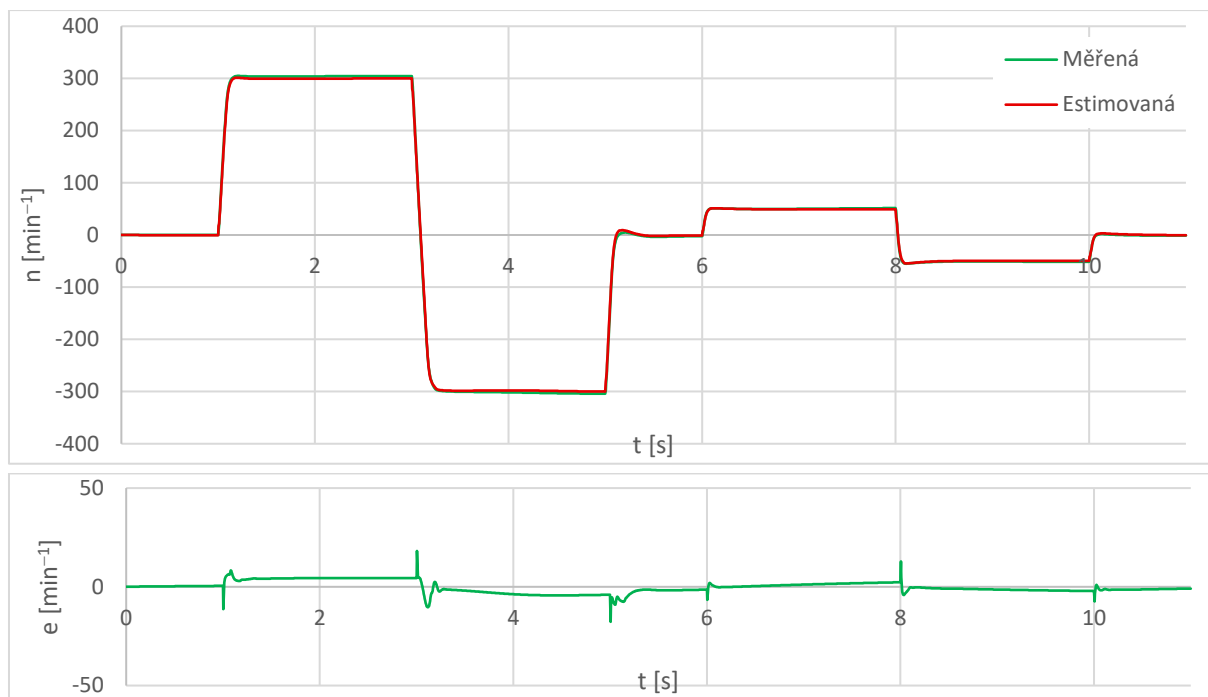
Obr. 4.8 Simulace ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, zatížení pro 300 min^{-1} , 20 % M_n



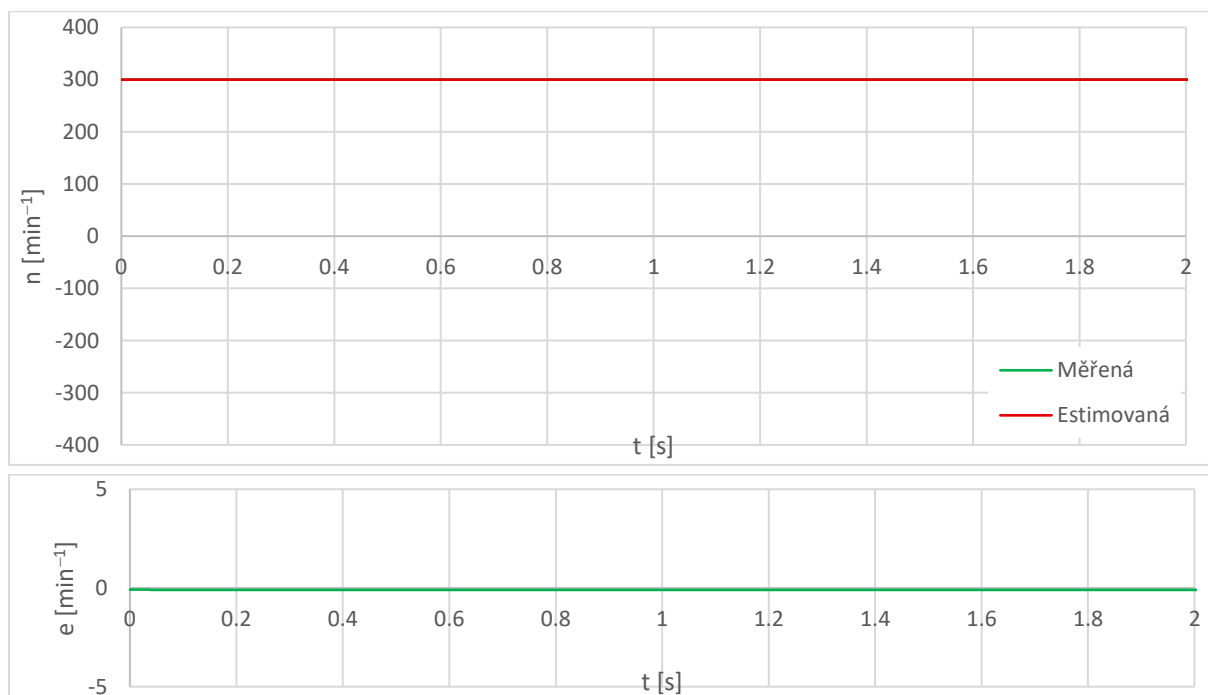
Obr. 4.9 Simulace ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , 20 % M_n

4.4 Pozorovatel ANNCB-MRAS xy

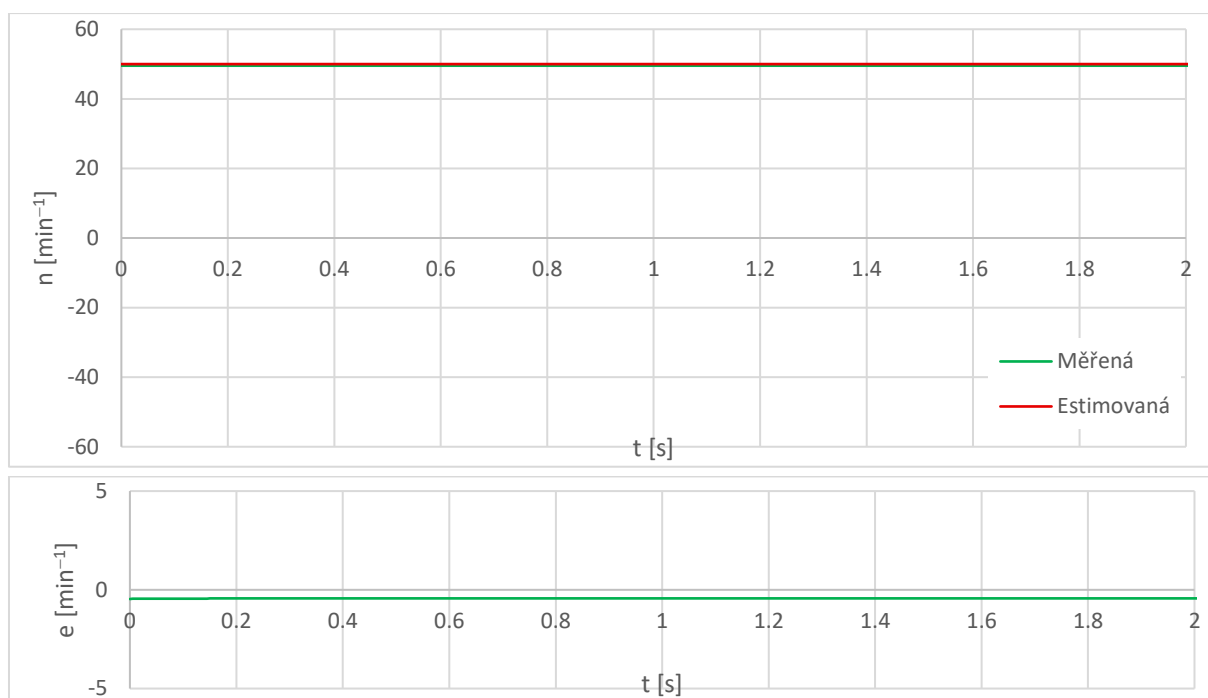
Simulované průběhy byly získány při stejném nastavení a se stejnou topologií sítě, jako průběhy získané měřením na skutečném pohonu (viz Tab. 6.1). Také byla snaha dodržet stejný postup získání a přípravy trénovacích vzorů. Byly použity trénovací vzory se zatížením. Obr. 4.10 obsahuje průběh měřené a estimované rychlosti pro sekvenci skoků žádané rychlosti $n^* = 0; 300; -300; 0; 50; -50; 0 \text{ ot} \cdot \text{min}^{-1}$ ($t = 0; 1; 3; 5; 6; 8; 10 \text{ s}$). Obr. 4.11 a Obr. 4.12 zobrazují průběhy při 20 % jmenovitého zatížení pro žádanou rychlost 300 a 50 $\text{ot} \cdot \text{min}^{-1}$. Ve srovnání s předchozími průběhy pozorovatele ANNCB-MRAS $\alpha\beta$ bylo se stejnými trénovacími vzory dosaženo vyšší přesnosti estimace rychlosti.



Obr. 4.10 Simulace ANNCB-MRAS xy: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti ± 300 a $\pm 50 \text{ min}^{-1}$, bez zátěže



Obr. 4.11 Simulace ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, zatížení pro 300 min^{-1} , 20 % M_n



Obr. 4.12 Simulace ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, zatížení pro 50 min^{-1} , $20 \% M_n$

5 Laboratorní stanoviště

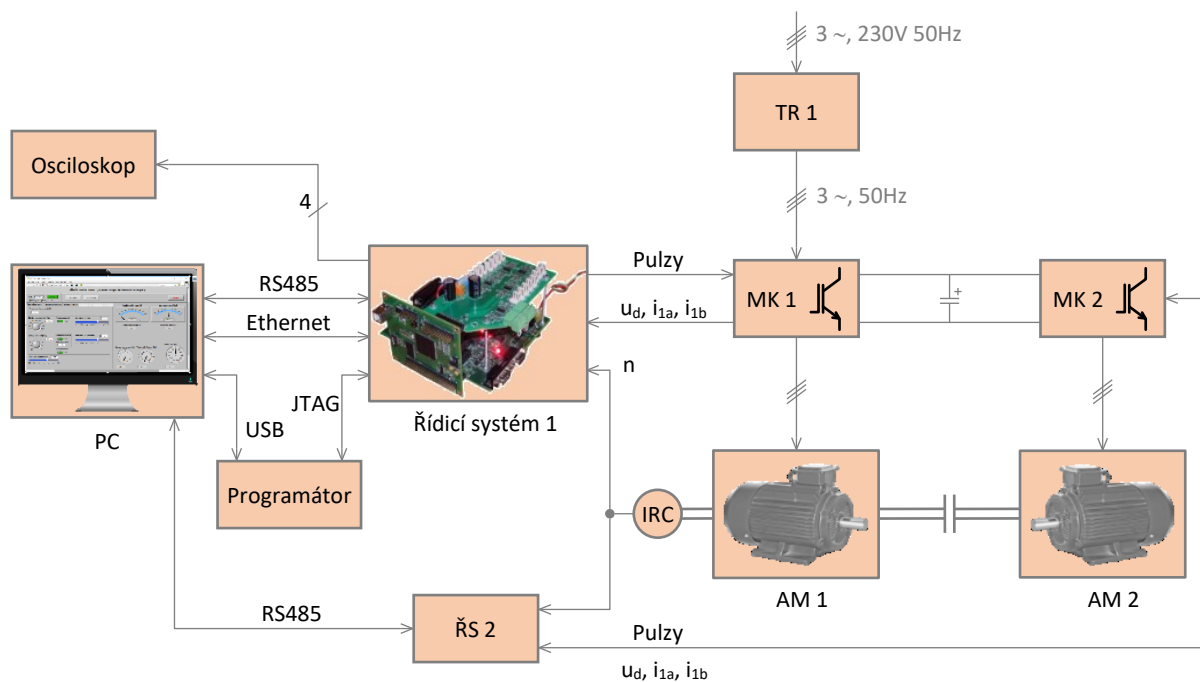
Simulační a experimentální výsledky byly získány pro pohon s asynchronním motorem *Celma Indukta Sg 100L-4A* o výkonu 2,2 kW, jehož parametry shrnuje Tab. 5.1. Blokové schéma laboratorního stanoviště je na Obr. 5.1, fotografie na Obr. 5.2, vybrané parametry laboratorního stanoviště jsou uvedeny v Tab. 5.2.

Laboratorní stanoviště je vybaveno motorovým soustrojím se dvěma mechanicky propojenými asynchronními motory stejného typu. Asynchronní motor 1 (AM1) je určen pro pohánění a je vybaven inkrementálním snímačem pro měření otáček a polohy. Asynchronní motor 2 (AM2) slouží jako zátěž a představuje pracovní mechanismus. Oba motory jsou napájeny z nepřímého měniče kmitočtu s napěťovým meziobvodem. Meziobvody měničů jsou propojeny, takže je brzdná energie motoru, který je právě v generátorickém režimu, odebírána druhým motorem. Napájecí vstup Měniče kmitočtu 1 (MK1) je připojen na výstup trojfázového autotransformátoru, aby bylo možné nastavovat velikost napětí v meziobvodu měniče. Každý měnič má svůj řídicí systém. V obou řídicích systémech je implementováno vektorové řízení (viz kap. 2.1). Dále je zde 4kanálový osciloskop, který je připojen k analogovému výstupu řídicího systému, je tak možné zobrazovat vnitřní proměnné řídicího systému. Pro ovládání řídicího systému slouží uživatelské rozhraní vytvořené v *LabVIEW* – Obr. 5.3.

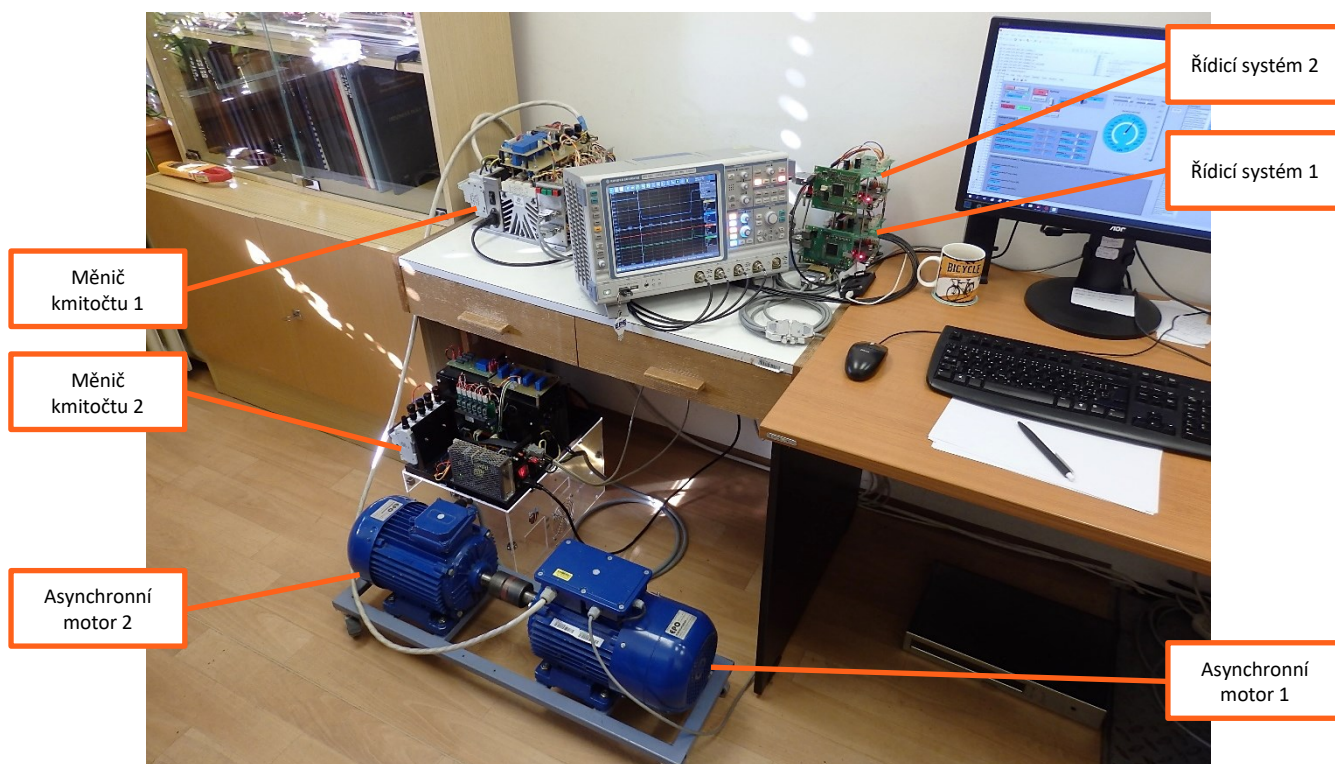
Použitý řídicí systém byl vyvinut na *Katedře elektroniky, VŠB – TUO*. Je určen pro řídicí aplikace z oblasti elektrických pohonů a výkonové elektroniky. Je vybaven digitálním signálovým kontrolérem *TMS320F28335* od *Texas Instruments*, který náleží do rodiny mikrokontrolérů C2000. Je to kontrolér podporující operace v plovoucí řádové čárce, který byl navržen pro použití ve vestavěných systémech reálného času, včetně elektrických pohonů. Mezi jeho periferie patří např. jednotka ePWM pro generování pulsně-šířkových modulací nebo eQEP pro vyhodnocování výstupu z inkrementálního snímače polohy. Taktovací frekvence je 150 MHz. [43]

V systému je využito přerušení od A/D převodníku, které má periodu $T_{vz} = 50 \mu s$ a je synchronizováno s čítačem PWM modulátoru. Toto přerušení má nejvyšší prioritu, je zde umístěn celý výpočet algoritmu vektorového řízení.

Pro účely práce s neuronovými sítěmi, byl vyvinut systém sběru dat, který je součástí řídicího systému a umožňuje odesílání vnitřních proměnných do počítače pro další zpracování. Je tak možné snadno změřit data potřebná pro natrénování neuronové sítě. Data se přenáší přes komunikační rozhraní Ethernet, prostřednictvím protokolu TCP/IP, a následně se ukládají do souboru pro další zpracování, např. v *Matlabu*. Komunikační rozhraní TCP/IP je na straně řídicího systému realizováno pomocí integrovaného obvodu *WIZnet W5300*. Vzorovací frekvence je nastavena na 20 kHz (dáno frekvencí výpočtu regulační smyčky), odesílá se deset 32bitových proměnných. [X]



Obr. 5.1 Blokové schéma experimentálního pohonu



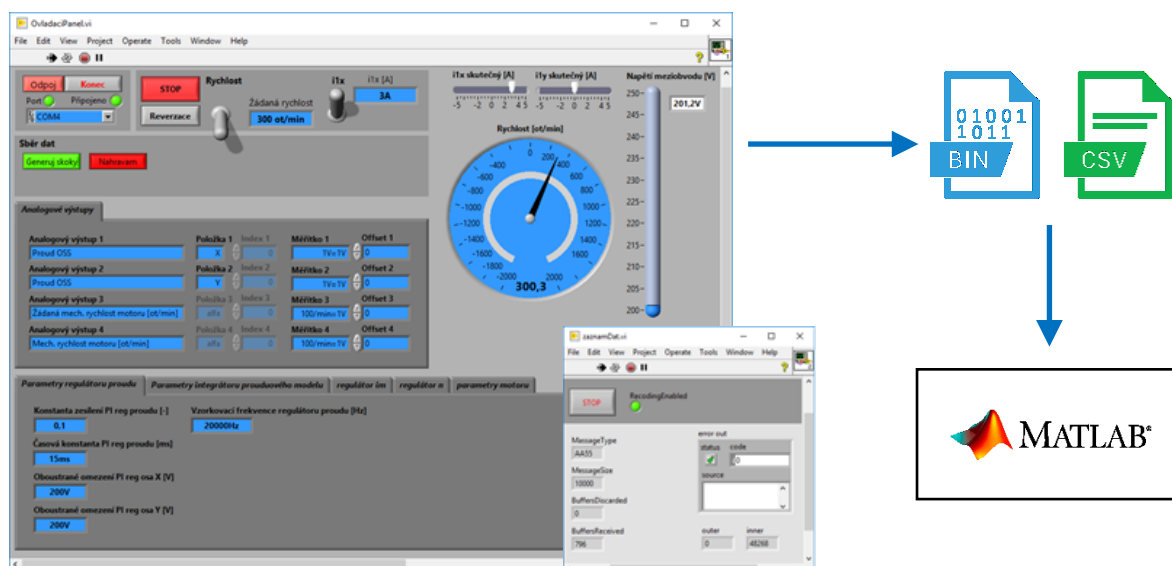
Obr. 5.2 Fotografie experimentálního pohonu

Tab. 5.1 Jmenovité parametry asynchronního motoru

Parametr	Hodnota	Parametr	Hodnota
Výkon	2,2 kW	R_s	2,78 Ω
Otáčky	1425 min^{-1}	R_r	2,84 Ω
Moment	14,7 Nm	L_s	318,9 mH
Napětí (Y/D)	400 / 230 V	L_r	318,1 mH
Proud (Y/D)	4,8 / 8,8 A	L_h	309 mH
Frekvence	50 Hz	R_{Fe}	1667 Ω
$\cos \varphi$	0,8	p	2

Tab. 5.2 Parametry laboratorního stanoviště

Parametr	Hodnota
Typ motoru	Sg 100L-4A
Snímače napětí	LEM LV 25-600
Snímače proudu	LEM LAH-25 NP
Snímač otáček	Heidenhain ERN 420 2048 01-03 (2048 imp/ot)
Moduly IGBT	SEMIKRON SKM 75GB 123D
Budiče tranzistorů	CONCEPT GSD106EI
Modulace	Komparační PWM
Frekvence PWM	10 kHz
Vzorkovací frekvence ADC	20 kHz
Typ procesoru	TMS320F28335
Osciloskop	Rohde&Schwartz RTE 1024



Obr. 5.3 Uživatelské rozhraní v LabVIEW

6 Experimentální výsledky

6.1 Nastavení experimentu

V kapitolách 6.2 až 6.5 jsou pro každou řešenou aplikaci uvedeny experimentální výsledky, které byly získány měřením na laboratorním pohonu popsaném v předchozí kapitole. Pokud není uvedeno jinak, bylo při měření použito nastavení podle Tab. 6.1. V matematických modelech byly uvažovány parametry asynchronního motoru podle Tab. 5.1. Jmenovitý momentotvorný proud je $i_{yn} = 5,5$ A (s uvažováním ztrát), magnetizační proud $i_{mn} = 3,1$ A.

Tab. 6.1 Nastavení použité při měření

Parametr	Hodnota
Napětí meziobvodu	$u_d = 200$ V
Buzení	$i_m^* = 3$ A
Regulátor otáček	$K_R = 0,3; T_R = 300$ ms;
Regulátor magn. proudu	$K_R = 1,3; T_R = 100$ ms;
Regulátory proudu	$K_R = 1,7; T_R = 4$ ms;
Regulátor ANNCB-MRAS	$K_R = 1; T_R = 1$ ms

Pokud bylo potřeba použít přepočítání mezi rotorovým tokem a magnetizačním proudem, byla rovněž použita hodnota hlavní indukčnosti L_h podle Tab. 5.1.

Čidla napětí a proudu byla zkalibrována podle změřených převodních charakteristik. Změřené převodní charakteristiky byly lineární regresí proloženy přímkou a získané koeficienty byly zadány do programu. Offsety proudových čidel se v programu měří automaticky při každém spuštění programu.

Vyhodnocování výstupu inkrementálního snímače pracovalo v kvadrantním režimu, rozlišení pro měření polohy bylo 4×2048 impulzů na otáčku, pro měření úhlové rychlosti 2048 imp/ot. V měřeném rozsahu otáček od 0 do 300 ot·min⁻¹ bylo pro určení úhlové rychlosti ω_m použito měření vzdálenosti mezi impulzy (měření periody). Jednotka eQEP má pro měření periody 16bitový čítač. Frekvence hodin čítače byla $150/4$ MHz, pro nejnižší rychlosti pak $150/128$ MHz. Poloha rotoru θ , používaná při výpočtu BVOV, byla odvozována přímo z aktuální hodnoty čítače impulzů inkrementálního snímače.

Řízení zátěže bylo realizováno zadáváním žádané hodnoty momentotvorného proudu i_{1y}^* , do druhého řídicího systému, regulační smyčka otáček byla vyřazena.

Rovnice PI regulátoru:

$$G(s) = K_R \left(\frac{1 + sT_R}{sT_R} \right) \quad (6.1)$$

6.1.1 Měření trénovacích dat

Trénovací data byla měřena v režimu vektorového řízení se snímačem otáček – viz schéma VŘ na Obr. 2.2. Seznam proměnných, které byly přenášeny do počítače, je uveden v Tab. 6.2. Data byla měřena se vzorkovací frekvencí $f_{VZ} = 20$ kHz. Takto změřená data sloužila nejen pro přípravu trénovacích vzorů, ale i pro další testování natrénovaných neuronových sítí (např. test chování proudového estimátoru jako součásti CB-MRAS).

Tab. 6.2 Proměnné přenášené z řídicího systému

	Symbol / jednotka	Popis
1	$u_{1\alpha}^*, u_{1\beta}^*$ [V]	Žádaná hodnota statorového napětí
2	$i_{1\alpha}, i_{1\beta}$ [A]	Měřený statorový proud
3	i_{1x}, i_{1y} [A]	Měřený statorový proud (natočení podle výstupu BVOV)
4	$\Psi_{1\alpha}, \Psi_{1\beta}$ [Wb]	Rotorový tok (výpočet BVOV + přepočít L_h)
5	ω_m [$\text{rad}\cdot\text{s}^{-1}$]	Měřená úhlová rychlost
6	ω_m^* [$\text{rad}\cdot\text{s}^{-1}$]	Žádaná hodnota úhlové rychlosti

Hodnoty č. 3 a 4 jsou z hlediska přípravy trénovacích dat nadbytečné, je možné je dopočítat. Hodnota žádané rychlosti byla užitečná pro třídění trénovacích vzorů.

6.2 Estimátor ANN-EST

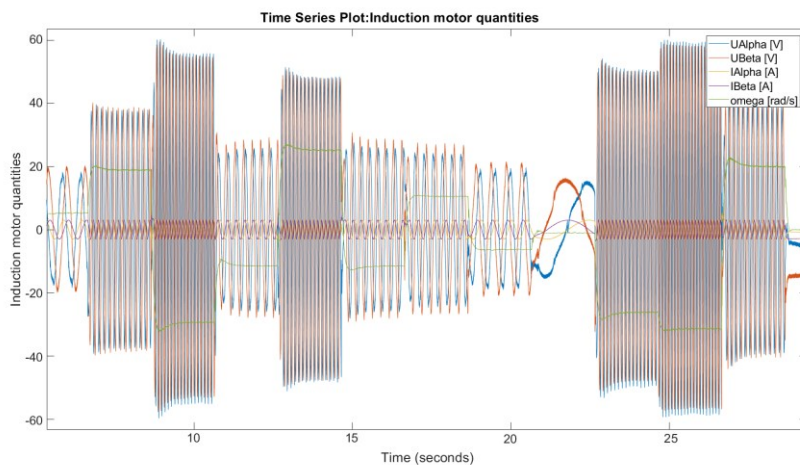
Celkem bylo změřeno 1200000 vzorků pro různé hodnoty rychlosti bez zátěže – Obr. 6.1, perioda vzorkování byla 50 μs . Z této kolekce bylo pro trénování náhodně vybráno 5000 trénovacích vzorů. Trénování bylo provedeno pomocí *Neural Network Toolbox*. Použitý trénovací algoritmus byl Levenberg-Marquardtův algoritmus zpětného šíření chyby, data byla rozdělena v poměru 60:30:10 na trénovací, validační a testovací data. Byla vybrána síť, která se zdála být nejlepší z hlediska odchylky mezi požadovaným a skutečným výstupem sítě po ukončení trénování. Proces trénování vybrané sítě byl ukončen ve 245. epoše a vyústil ve střední kvadratickou odchylku (*Mean Squared Error*, MSE) $63,11 \text{ min}^{-2}$.

Konstanty natrénované sítě byly exportovány do formátu pro použití v řídicím systému. Při experimentálním vyhodnocení, stejně tak jako při sběru dat, byly společné tyto podmínky: napětí v meziobvodu 200 V, žádaný magnetizační proud 3 A, vzorkovací frekvence 50 μs . Uvedené experimentální výsledky zobrazují funkci estimátoru v bezsenzorovém režimu. Odchylka rychlosti byla získána porovnáním výstupu neuronové sítě s rychlostí měřenou inkrementálním snímačem.

Z průběhu odchylky estimované rychlosti byly vypočítány tyto chybové ukazatele: střední odchylka (*Mean Error*, ME), střední absolutní odchylka (*Mean Absolute Error*, MAE), střední kvadratická odchylka (*Mean Squared Error*, MSE) a odmocnina střední kvadratické odchylky (*Root Mean Squared Error*, RMSE). Odchylky v ustáleném stavu pro různé hodnoty rychlosti jsou uvedeny v Tab. 6.3. V posledním řádku tabulky jsou pro srovnání hodnoty vypočítané pro trénovací vzory. Hodnota MSE je výrazně větší, protože trénovací vzory obsahovaly přechodné děje při skocích žádané rychlosti, kde je odchylka větší. Graficky jsou odchylky pro jednotlivé rychlosti porovnány na Obr. 6.2.

Na Obr. 6.3 až Obr. 6.5 jsou průběhy pro skoky žádané rychlosti $\pm 200 \text{ min}^{-1}$ bez zatížení. Na Obr. 6.3 je porovnání měřené a estimované rychlosti a jejich odchylka. Na Obr. 6.4 je porovnání „měřeného“ rotorového toku, který byl vypočítán podle měřené rychlosti a „estimovaného“ rotorového toku, který byl vypočítán podle estimované rychlosti. Na Obr. 6.5 jsou průběhy statorového proudu v OSS. Je zde porovnán proud získaný vektorovým natočením podle polohy „měřeného“ a podle polohy „estimovaného“ rotorového toku.

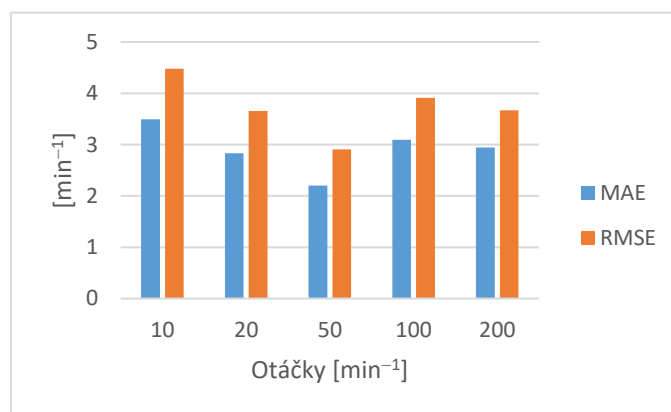
Na Obr. 6.6 až Obr. 6.8 jsou stejné průběhy pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$.



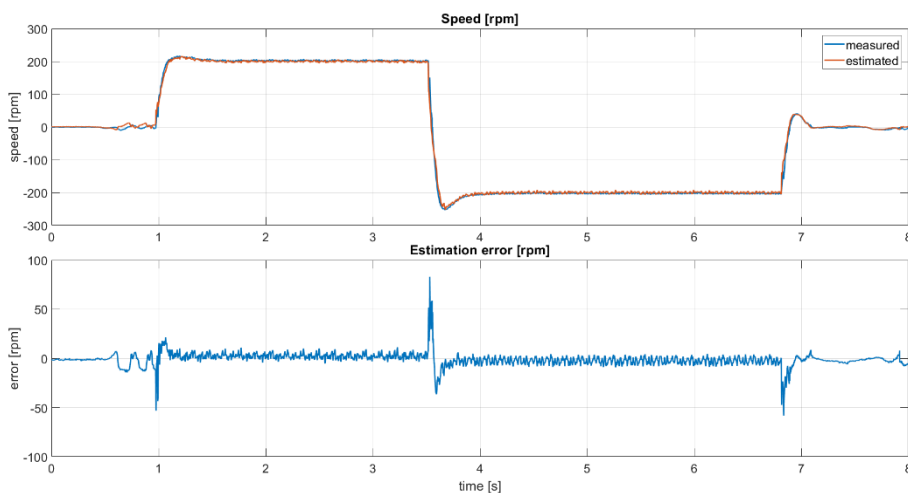
Obr. 6.1 Trénovací data, ANN-EST

Tab. 6.3 Chybové ukazatele pro různé rychlosti, ANN-EST

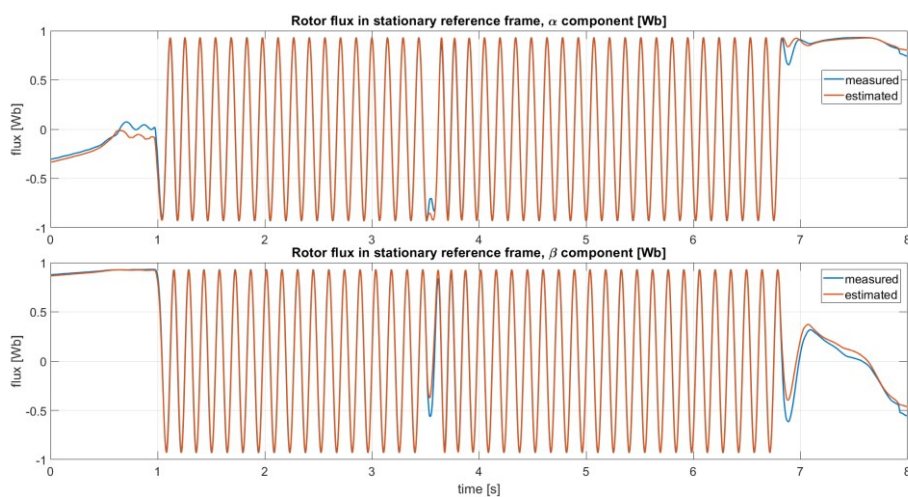
Dataset	ME	MAE	RMSE	ME	MAE	RMSE	MSE
-	[min ⁻¹]			[%]			[min ⁻²]
10 rpm	-3,23	4,12	5,37	32,31	41,16	53,71	28,84
-10 rpm	0,38	2,87	3,59	3,84	28,69	35,86	12,86
20 rpm	-2,24	3,14	4,13	11,18	15,70	20,63	17,02
-20 rpm	0,35	2,53	3,18	1,76	12,66	15,89	10,10
50 rpm	-1,25	2,06	2,58	2,51	4,13	5,16	6,65
-50 rpm	-1,88	2,34	3,23	3,77	4,68	6,46	10,42
100 rpm	-2,86	3,32	4,17	2,86	3,32	4,17	17,39
-100 rpm	1,86	2,87	3,65	1,86	2,87	3,65	13,34
200 rpm	2,35	2,72	3,48	1,17	1,36	1,74	12,10
-200 rpm	-2,45	3,17	3,85	1,23	1,59	1,92	14,79
Training	-	4,13	7,94	-	-	-	63,11



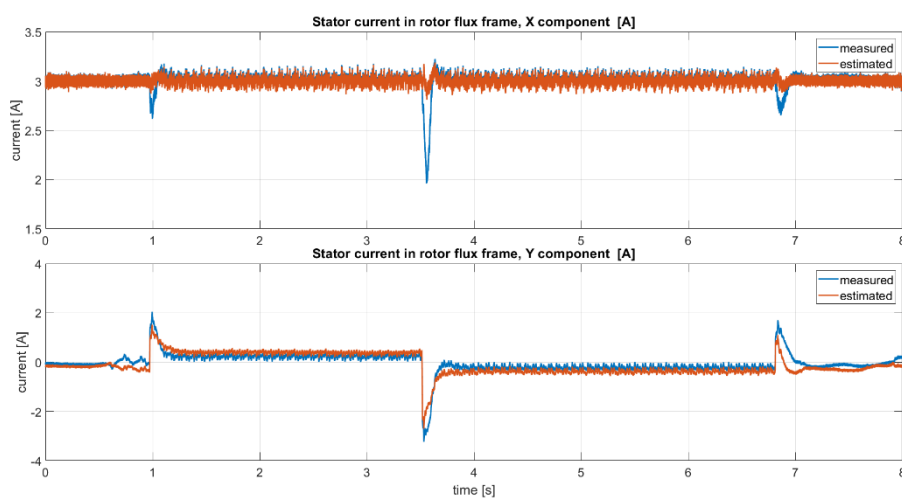
Obr. 6.2 Porovnání chybových ukazatelů pro různé rychlosti, ANN-EST



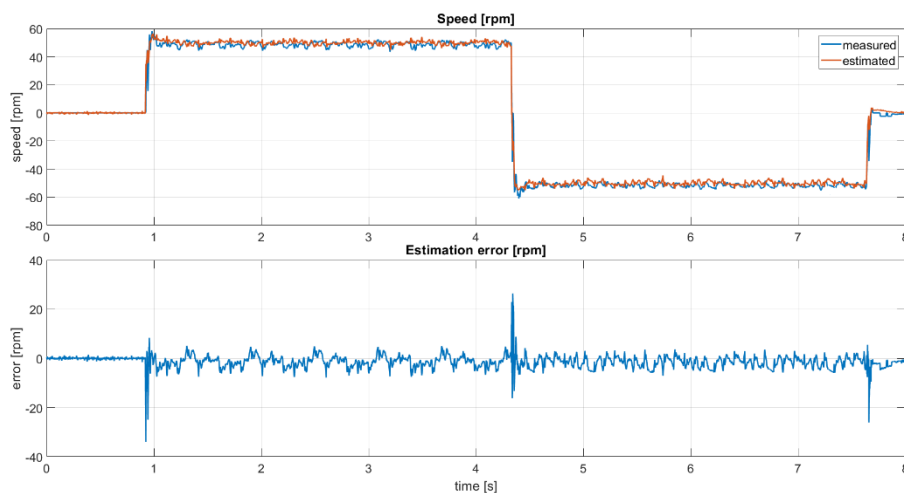
Obr. 6.3 ANN-EST: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 200 \text{ min}^{-1}$, bez zátěže



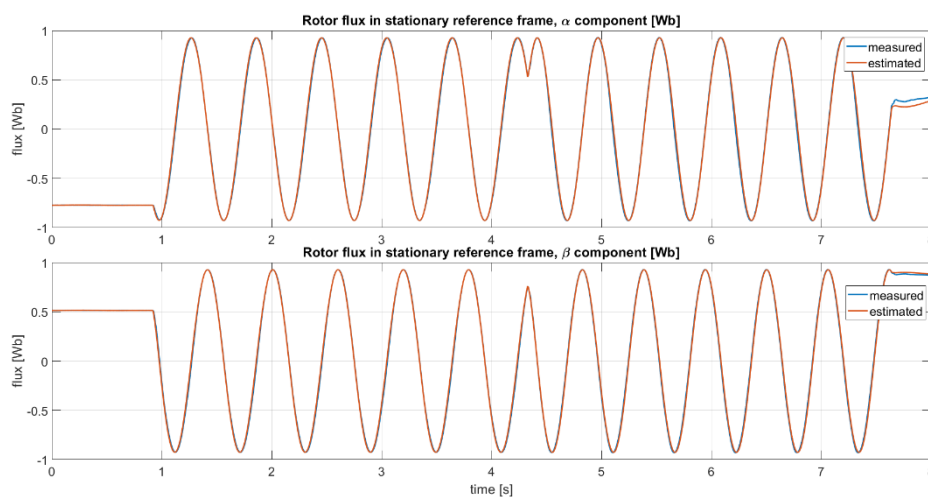
Obr. 6.4 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 200 \text{ min}^{-1}$, bez zátěže



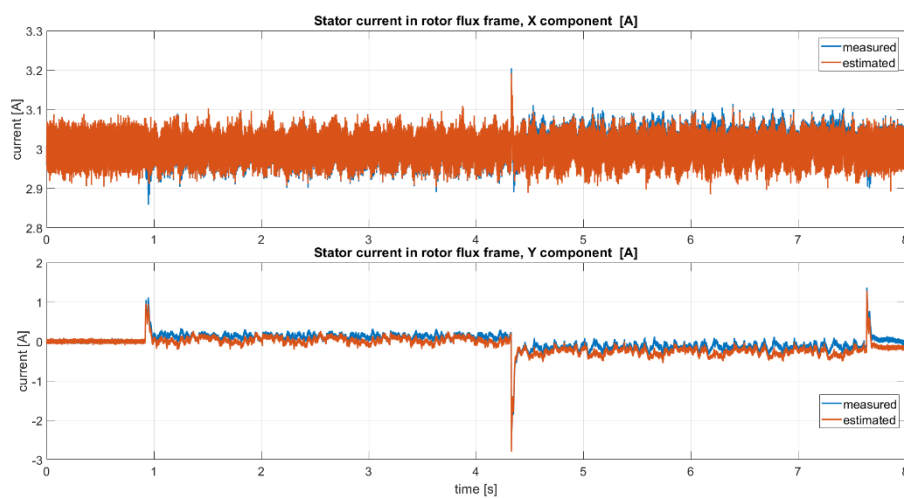
Obr. 6.5 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 200 \text{ min}^{-1}$, bez zátěže



Obr. 6.6 ANN-EST: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



Obr. 6.7 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



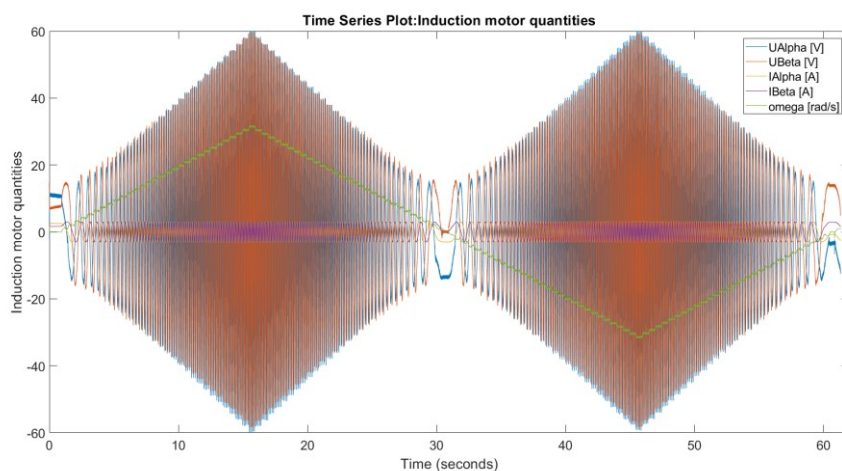
Obr. 6.8 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže

6.3 Pozorovatel ANNRF-MRAS

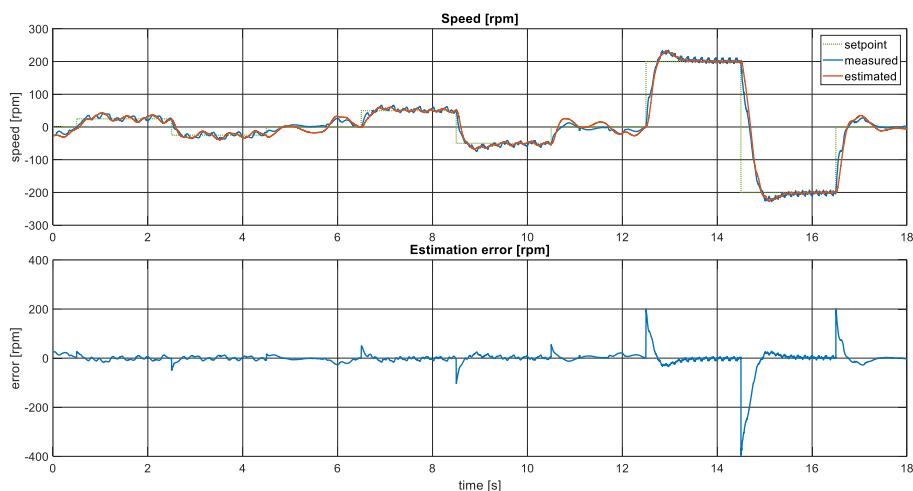
Celkem bylo změřeno 1220000 vzorků pro různé hodnoty rychlosti a bez zátěže – Obr. 6.9, perioda vzorkování byla $50 \mu\text{s}$. Z této kolekce bylo pro trénování náhodně vybráno 5000 trénovacích vzorů. Trénování bylo provedeno pomocí Neural Network Toolbox. Použitý trénovací algoritmus byl Levenberg-Marquardtův algoritmus zpětného šíření chyby, data byla rozdělena v poměru 60:30:10 na trénovací, validační a testovací data. Byla vybrána síť, která se zdála být nejlepší z hlediska odchylky mezi požadovaným a skutečným výstupem sítě po ukončení trénování. Proces trénování vybrané sítě byl ukončen ve 112. epoše a vyústil ve střední kvadratickou odchylku $4.7 \cdot 10^{-4} \text{ Wb}^2$.

Konstanty natrénované sítě byly exportovány do formátu pro použití v řídicím systému. Při experimentálním vyhodnocení, stejně tak jako při sběru dat, byly společné tyto podmínky: napětí v meziobvodu 200 V, žádaný magnetizační proud 3 A, vzorkovací frekvence $50 \mu\text{s}$. Uvedené experimentální výsledky zobrazují funkci pozorovatele v bezsenzorovém režimu.

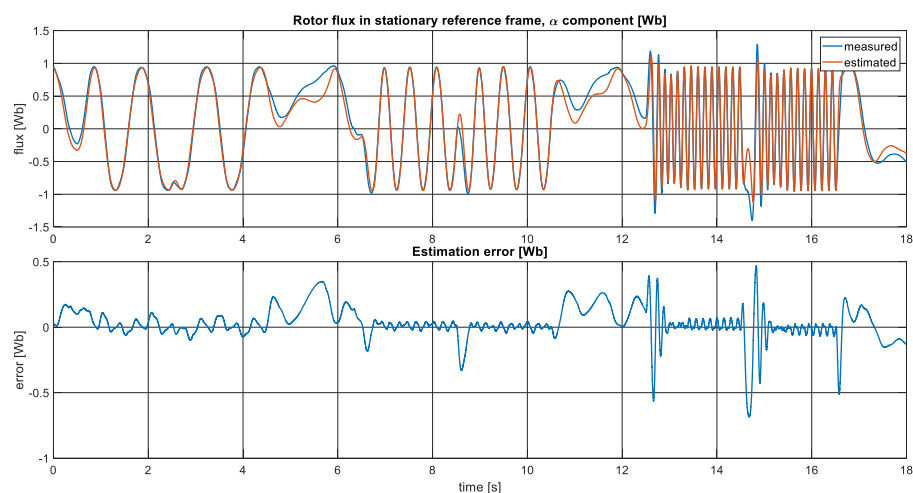
Význam veličin v grafech je stejný, jako u estimátoru ANN-EST. Obr. 6.10 až Obr. 6.12 zobrazují průběhy pro skoky žádané rychlosti ± 25 , ± 50 a $\pm 200 \text{ min}^{-1}$ změřené s pozorovatelem RF-MRAS (viz kap. 2.3), Obr. 6.13 až Obr. 6.15 pak průběhy pro pozorovatel RF-MRAS s neuronovou sítí. Nejvíce je zlepšení patrné v oblasti nulových a velmi nízkých otáček.



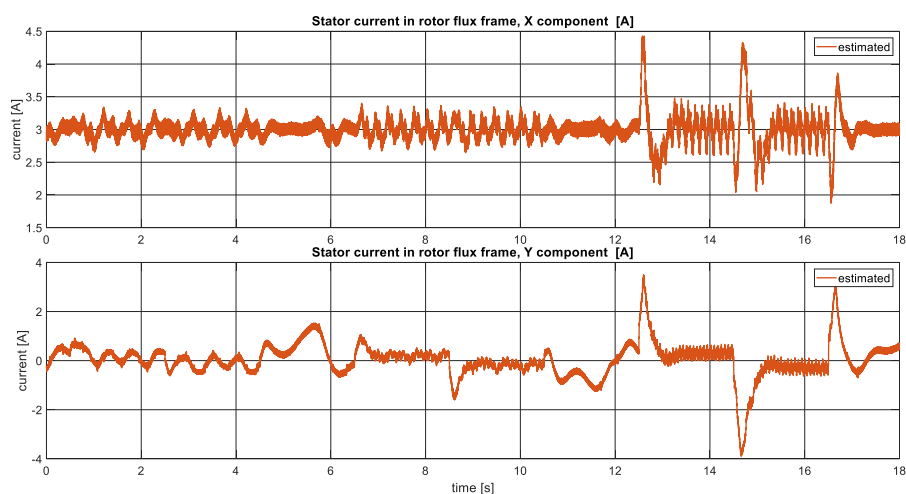
Obr. 6.9 Trénovací data



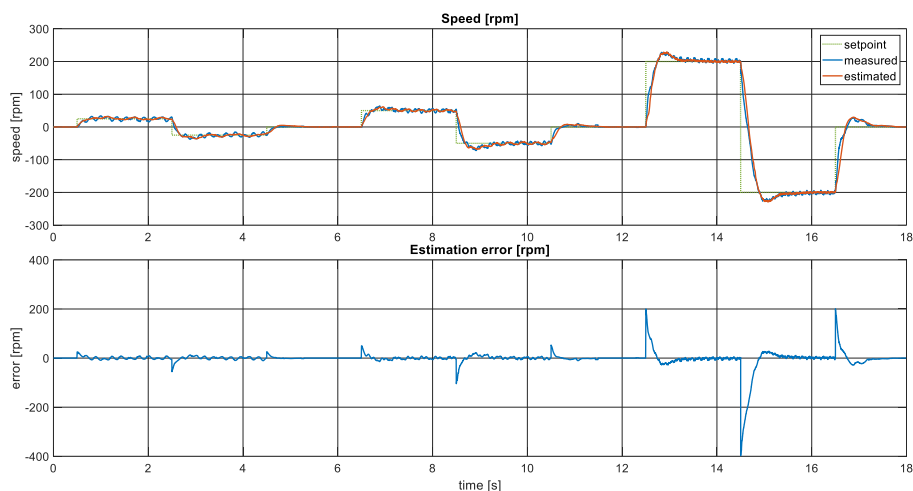
Obr. 6.10 RF-MRAS: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti, $0, \pm 25, \pm 50, \pm 200 \text{ min}^{-1}$, bez zátěže



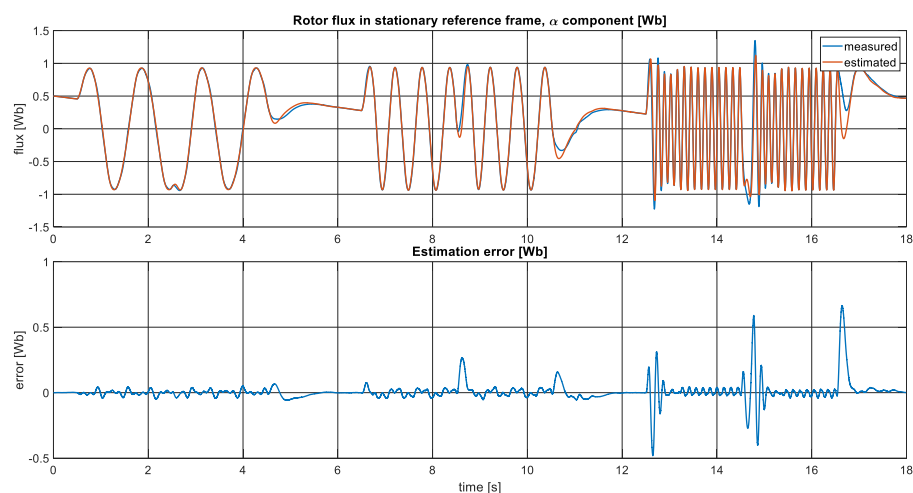
Obr. 6.11 RF-MRAS: porovnání rotorových toků v SSS pro skoky žádané rychlosti, bez zátěže



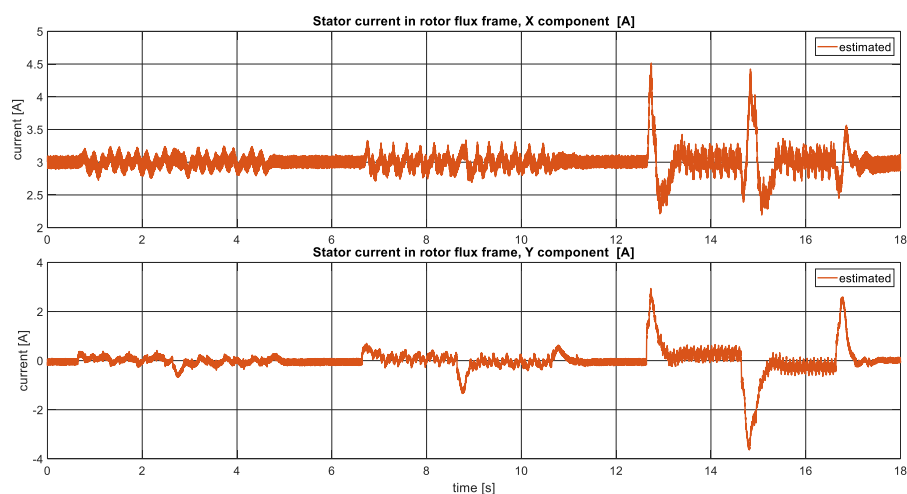
Obr. 6.12 RF-MRAS: průběh statorového proudu v OSS pro skoky žádané rychlosti, bez zátěže



Obr. 6.13 ANN-RFMRAS: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti, $0, \pm 25, \pm 50, \pm 200 \text{ min}^{-1}$, bez zátěže



Obr. 6.14 ANN-RFMRAS: porovnání rotorových toků v SSS pro skoky žádané rychlosti, bez zátěže



Obr. 6.15 ANN-RFMRAS: průběh statorového proudu v OSS pro skoky žádané rychlosti, bez zátěže

6.4 Pozorovatel ANNCB-MRAS $\alpha\beta$

Do řídicího systému bylo implementováno vektorové řízení s pozorovatelem ANNCB-MRAS $\alpha\beta$, který byl popsán v kap. 3.3. Byla použita topologie sítě podle Obr. 3.7 s 24 neurony ve skryté vrstvě (10-24-2).

Následující experimentální výsledky byly získány měřením při provozu v bezsenzorovém režimu, kdy byla estimovaná rychlost $\hat{\omega}_m$ použita jako vstup skutečných otáček ω_m a zároveň estimovaný rotorový tok $\hat{\Psi}_2^S$ byl použit pro výpočet orientujících veličin ($\sin \gamma$ a $\cos \gamma$) a skutečné hodnoty magnetizačního proudu i_m , byl tak nahrazen výstup BVOV – viz schéma vektorového řízení na Obr. 2.2.

Pro srovnání s pozorovatelem ANNCB-MRAS xy byly změřeny stejné průběhy veličin pro otáčky 50 a 300 ot·min⁻¹. Byl měřen rozběh, reverzace a zastavení s nezatíženým motorem a přechod do generátorického režimu s motorem zatíženým konstantním momentem.

Na Obr. 6.16 až Obr. 6.18 jsou zobrazeny průběhy při skocích žádané rychlosti, které byly měřeny s nezatíženým motorem pro 300 ot·min⁻¹ (sekvence skoků $n^* = 0; 300; -300; 0$ ot·min⁻¹). Na Obr. 6.16 je porovnání rychlosti měřené inkrementálním snímačem („měřená“) a estimované rychlosti $\hat{\omega}_m$ („estimovaná“). Spodní graf představuje odchylku mezi měřenou a estimovanou rychlostí. Obr. 6.17 zobrazuje porovnání α složky rotorového toku $\Psi_{2\alpha}$ („měřená“), který byl estimován s použitím měřené rychlosti, a rotorového toku $\hat{\Psi}_{2\alpha}$, který je výstupem proudového modelu v ANNCB-MRAS.

Obr. 6.18 zobrazuje tokotvorný a momentotvorný proud. Význam veličin v následujících grafech je stejný.

Na Obr. 6.19 až Obr. 6.21 jsou průběhy při skocích žádané rychlosti, které byly měřeny s nezatíženým motorem pro 50 ot·min⁻¹.

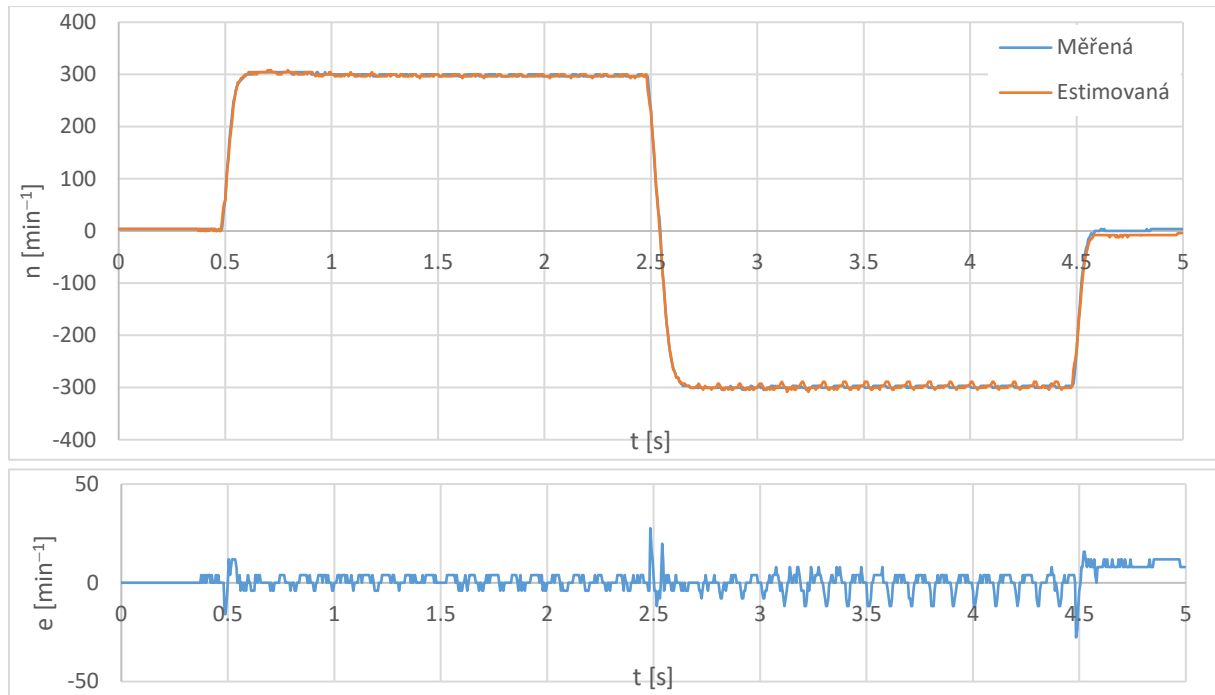
Grafy na Obr. 6.22 až Obr. 6.24 ukazují průběhy při přechodu z motorického do generátorického režimu a zpět pro skoky žádané rychlosti 300 min⁻¹ (sekvence skoků $n^* = 300; -300; 300$ ot·min⁻¹). Motor byl zatížen konstantním momentem, který odpovídal 20 % jmenovitého zatížení.

Na Obr. 6.25 až Obr. 6.27 jsou průběhy při přechodu z motorického do generátorického režimu a zpět pro skoky žádané rychlosti 50 ot·min⁻¹ a 20 % jmenovitého zatížení.

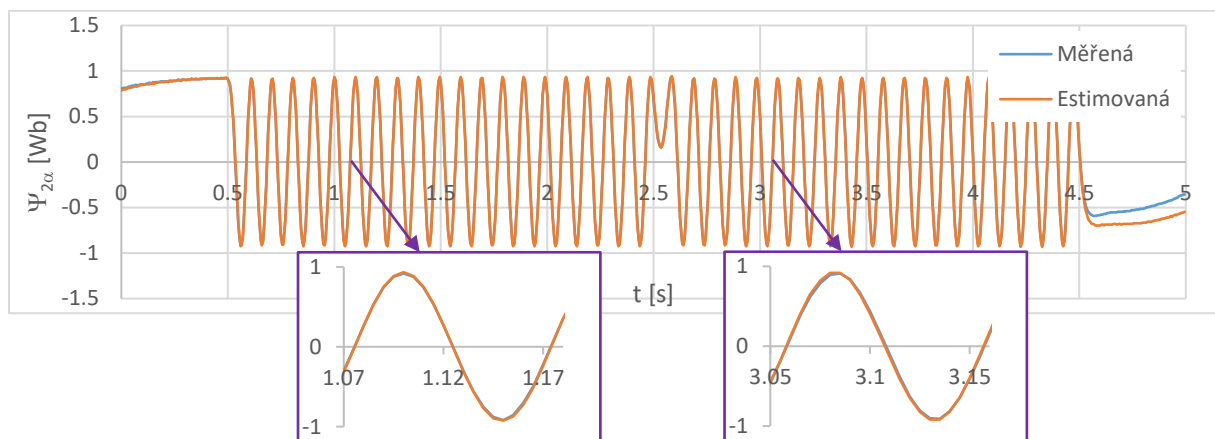
Na Obr. 6.28 je přechod do generátorického režimu pro 300 ot·min⁻¹ a 40 % jmenovitého zatížení. Zde už se v generátorickém režimu projevila větší chyba a nestabilita.

Na Obr. 6.29 jsou průběhy měřeného a estimovaného statorového proudu $i_{1\alpha}$ a $\hat{i}_{1\alpha}$ při přechodu do generátorického režimu pro skoky žádané rychlosti 50 ot·min⁻¹ a zřízení 20 % M_n .

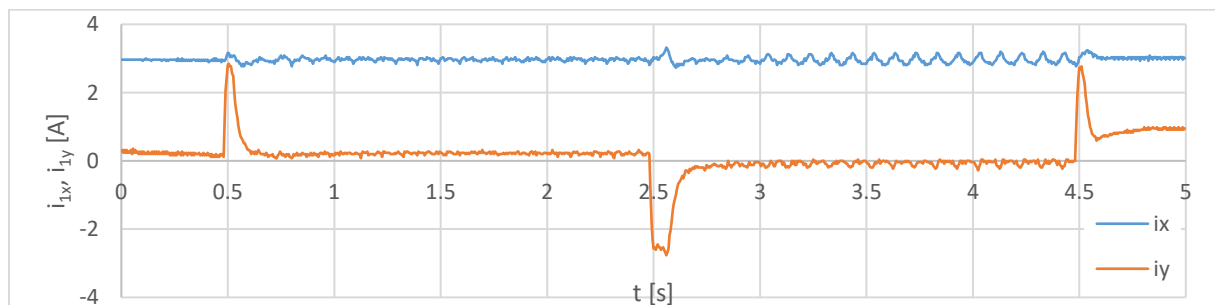
Poslední grafy na Obr. 6.30 až Obr. 6.32 ukazují průběhy při 20 ot·min⁻¹ a 40 % jmenovitého momentu zátěže.



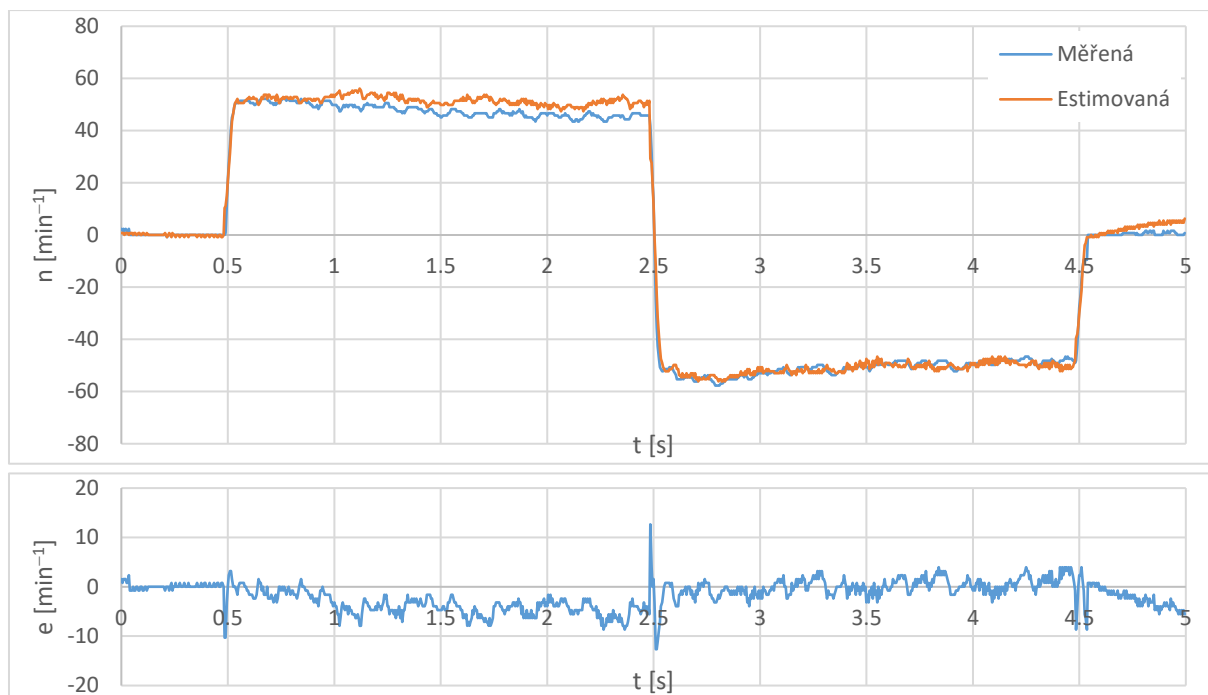
Obr. 6.16 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže



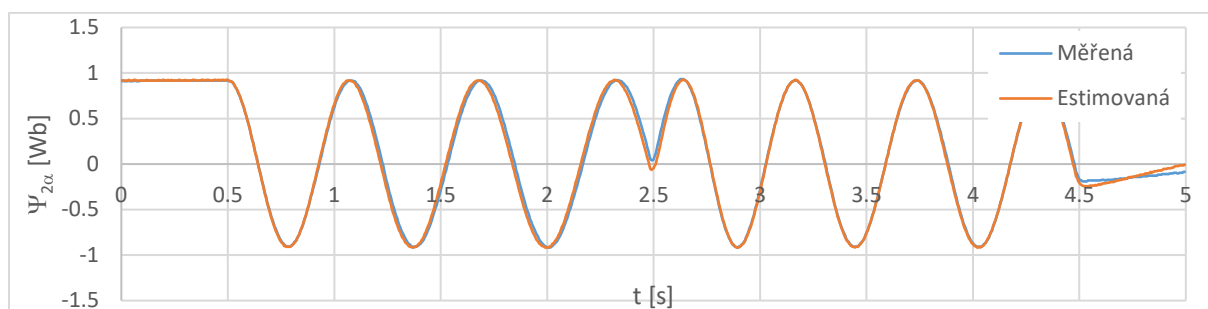
Obr. 6.17 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže



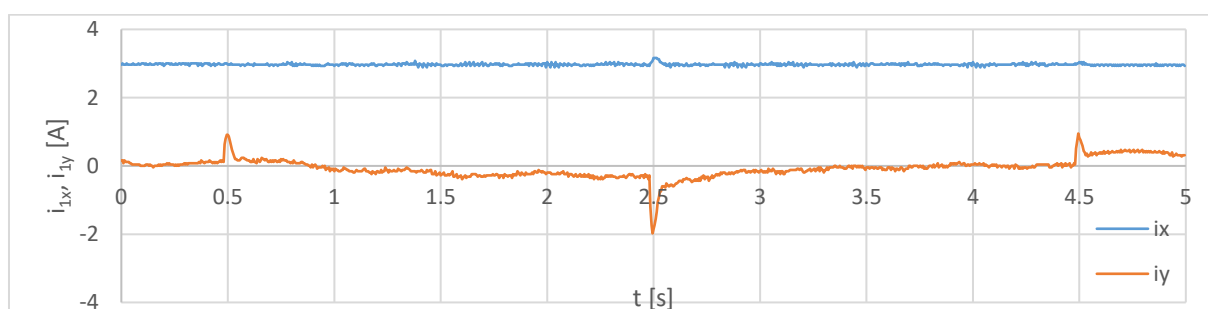
Obr. 6.18 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže



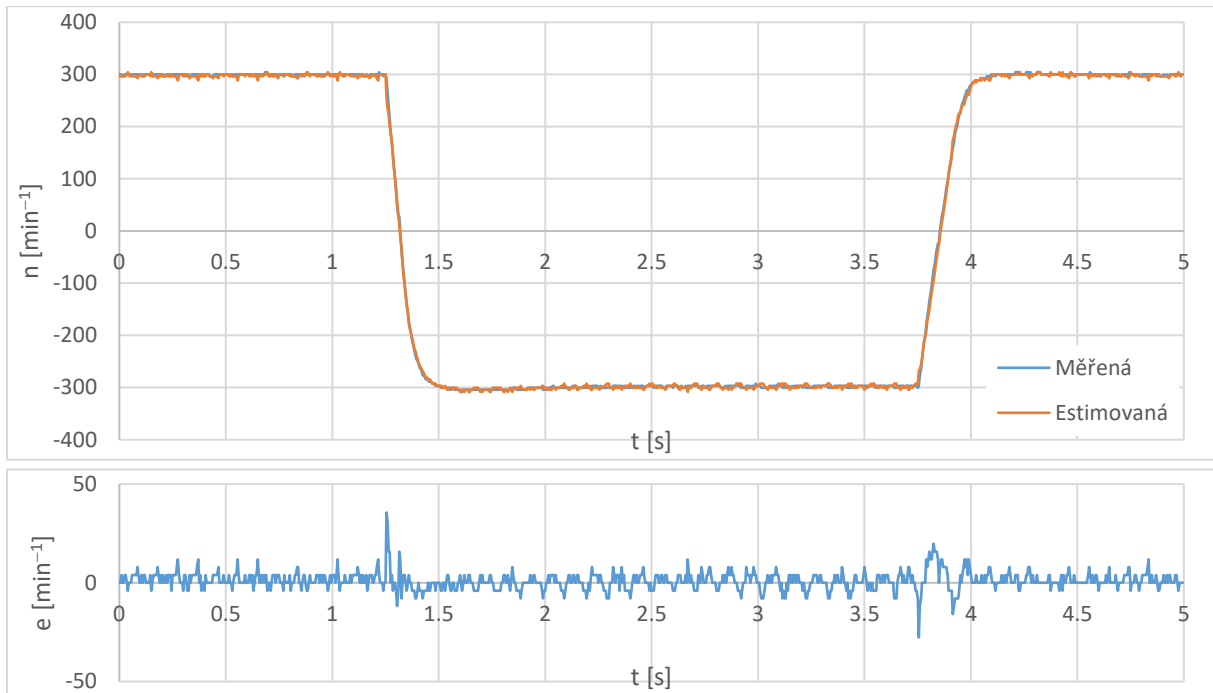
Obr. 6.19 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



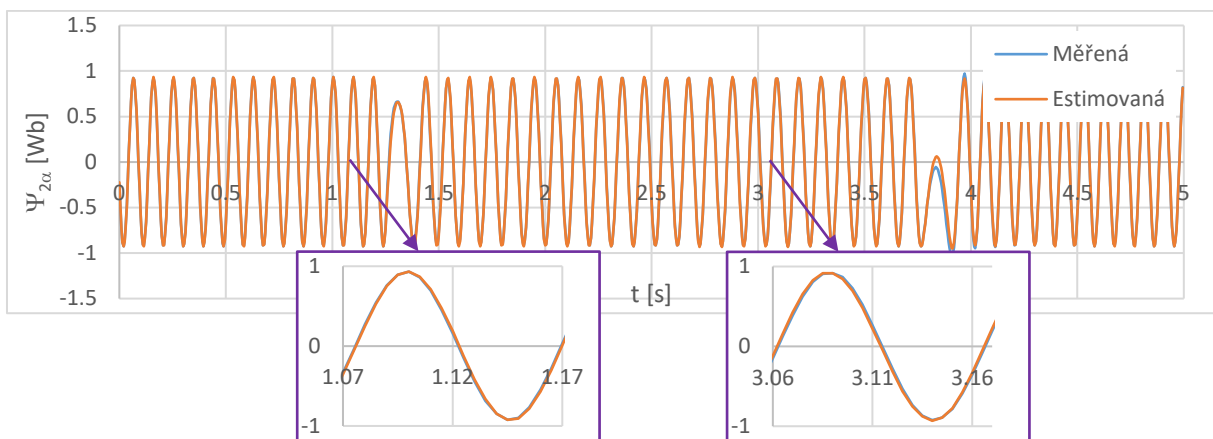
Obr. 6.20 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



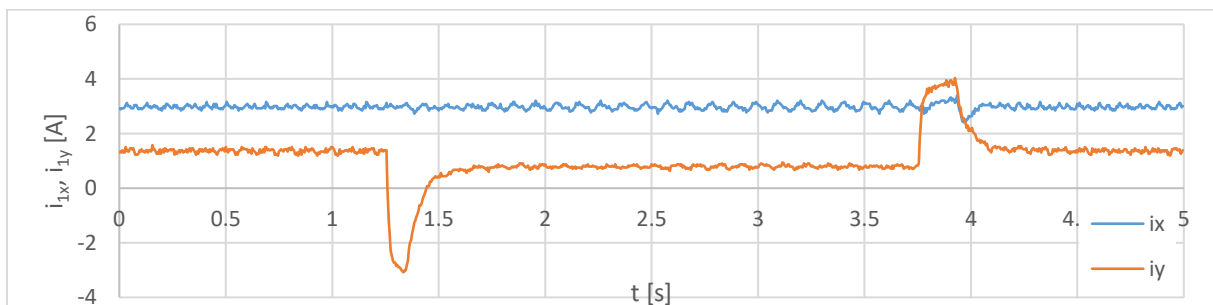
Obr. 6.21 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



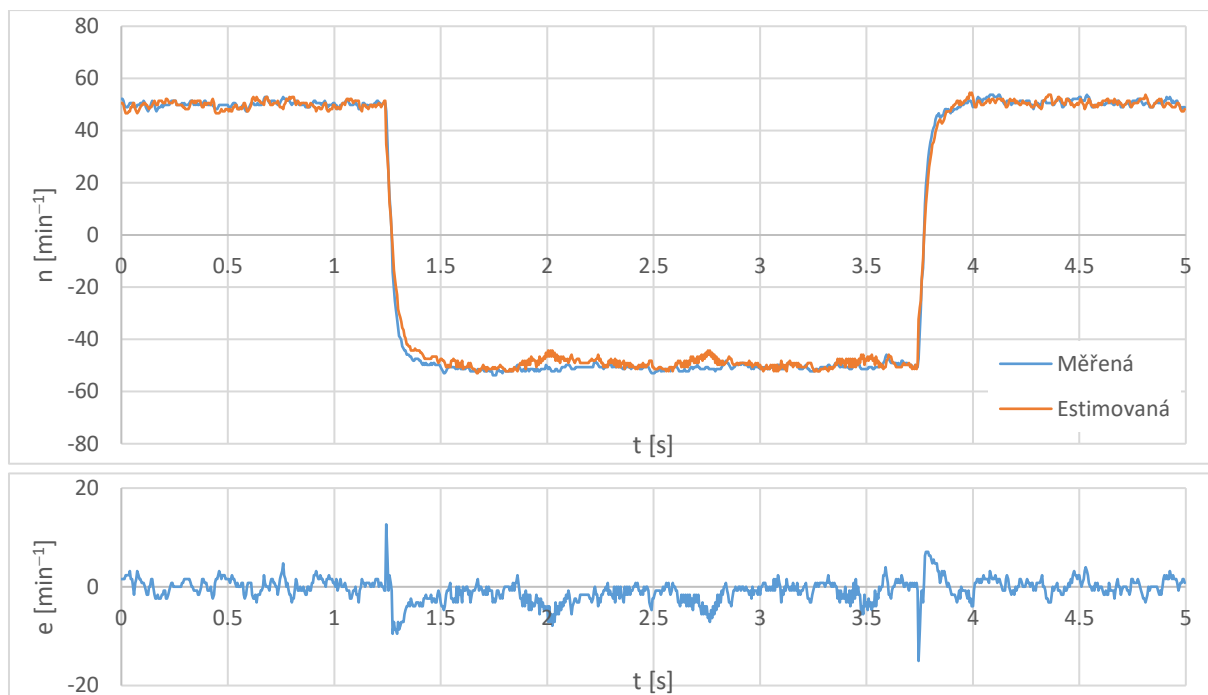
Obr. 6.22 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 300 min^{-1} , $20 \% M_n$



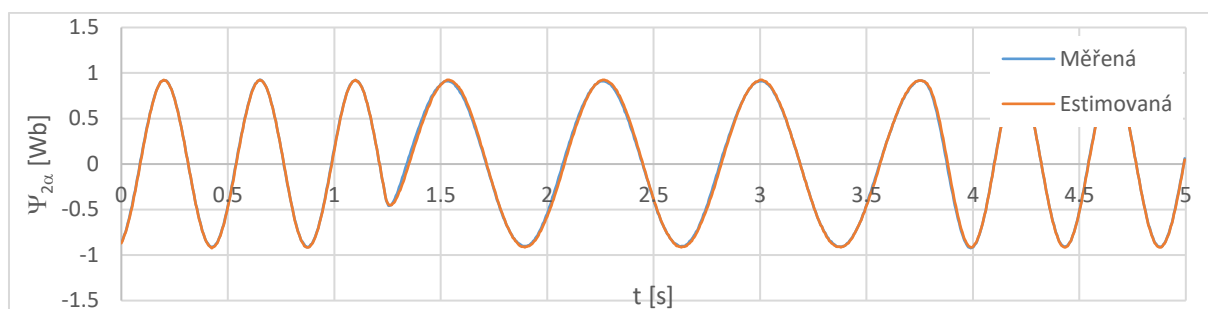
Obr. 6.23 Porovnání rotorových toků v SSS, přechod do gen. režimu, 300 min^{-1} , $20 \% M_n$



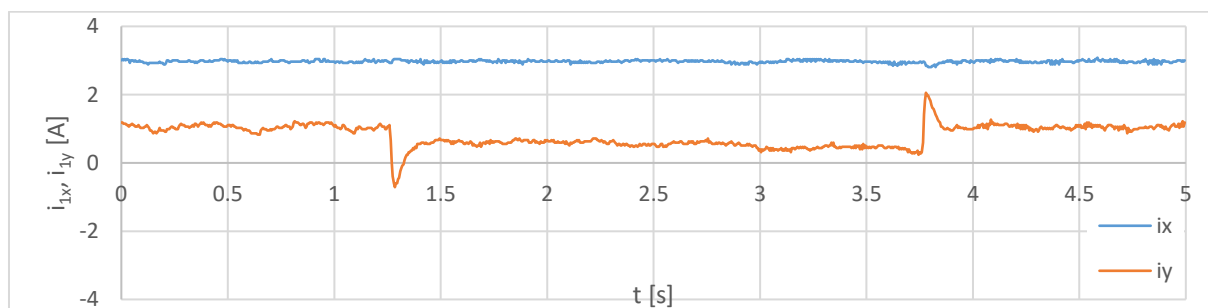
Obr. 6.24 Průběh statorového proudu v OSS, přechod do gen. režimu, 300 min^{-1} , $20 \% M_n$



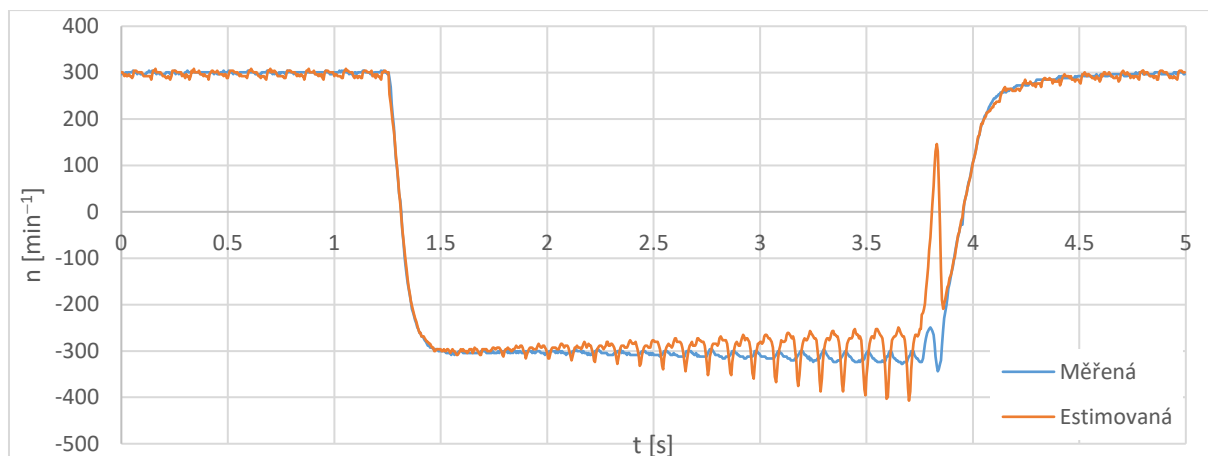
Obr. 6.25 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 50 min^{-1} , $20 \% M_n$



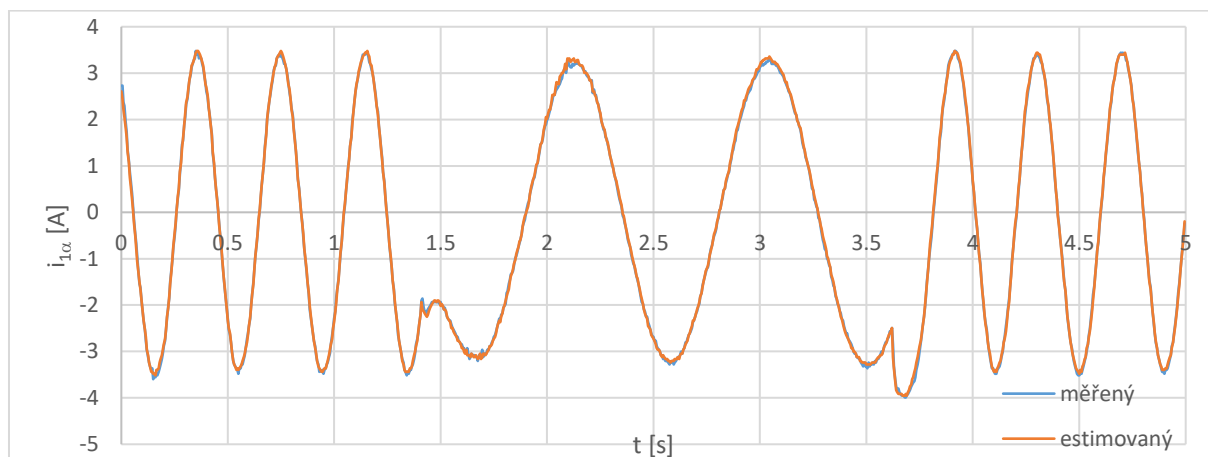
Obr. 6.26 Porovnání rotorových toků v SSS, přechod do gen. režimu, 50 min^{-1} , $20 \% M_n$



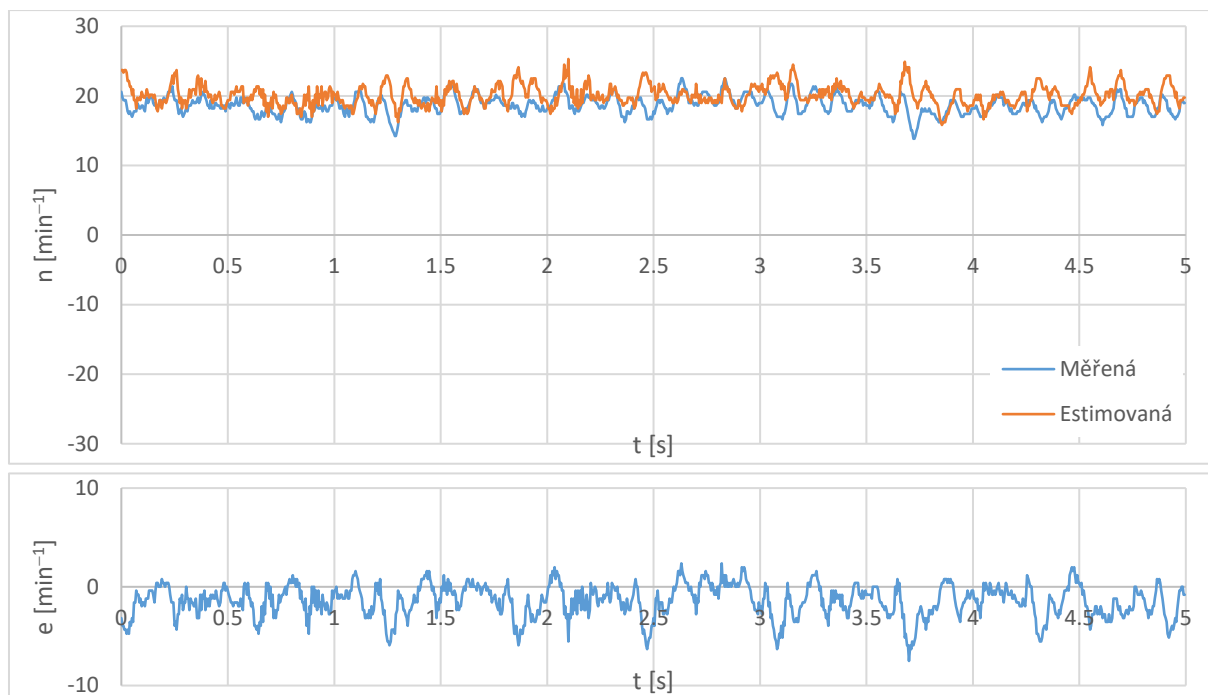
Obr. 6.27 Průběh statorového proudu v OSS, přechod do gen. režimu, 50 min^{-1} , $20 \% M_n$



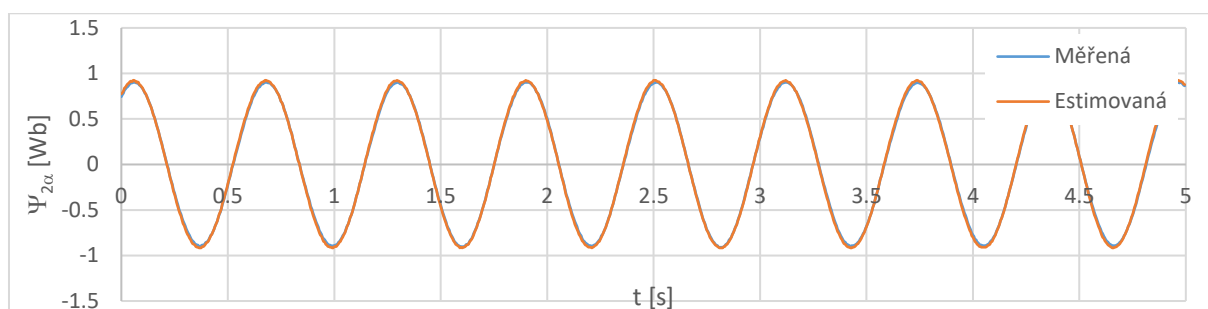
Obr. 6.28 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 300 min^{-1} , $40 \% M_n$



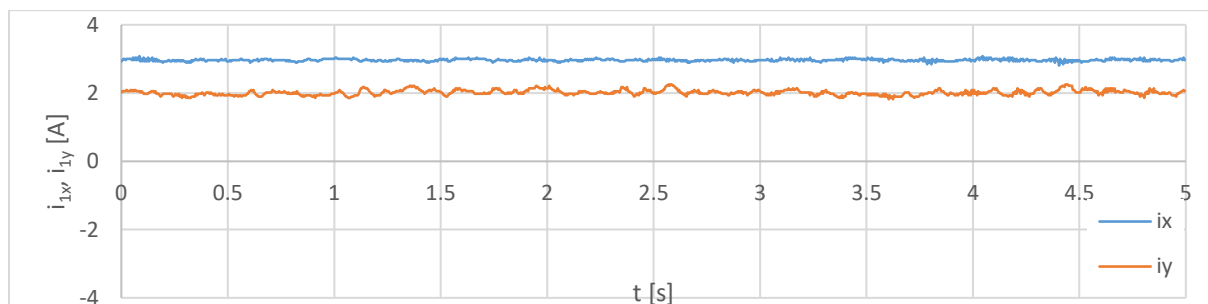
Obr. 6.29 ANNCB-MRAS $\alpha\beta$: průběh měřeného a estimovaného proudu v SSS, přechod do gen. režimu, 50 min^{-1} , $20 \% M_n$



Obr. 6.30 ANNCB-MRAS $\alpha\beta$: měřená a estimovaná rychlost motoru při 20 min^{-1} , $40 \% M_n$.



Obr. 6.31 Porovnání rotorových toků v SSS při 20 min^{-1} , $40 \% M_n$



Obr. 6.32 Průběh statorového proudu v OSS při 20 min^{-1} , $40 \% M_n$

6.5 Pozorovatel ANNCB-MRAS xy

Do řídicího systému bylo implementováno vektorové řízení s pozorovatelem ANNCB-MRAS xy, který byl popsán v kap.3.4.

Následující experimentální výsledky byly získány měřením při provozu v bezsenzorovém režimu, kdy byla estimovaná rychlost $\hat{\omega}_m$ použita jako vstup skutečných otáček ω_m a zároveň estimovaný rotorový tok $\hat{\Psi}_2^S$ byl použit pro výpočet orientujících veličin ($\sin \gamma$ a $\cos \gamma$) a skutečné hodnoty magnetizačního proudu i_m , byl tak nahrazen výstup BVOV – viz schéma vektorového řízení na Obr. 2.2.

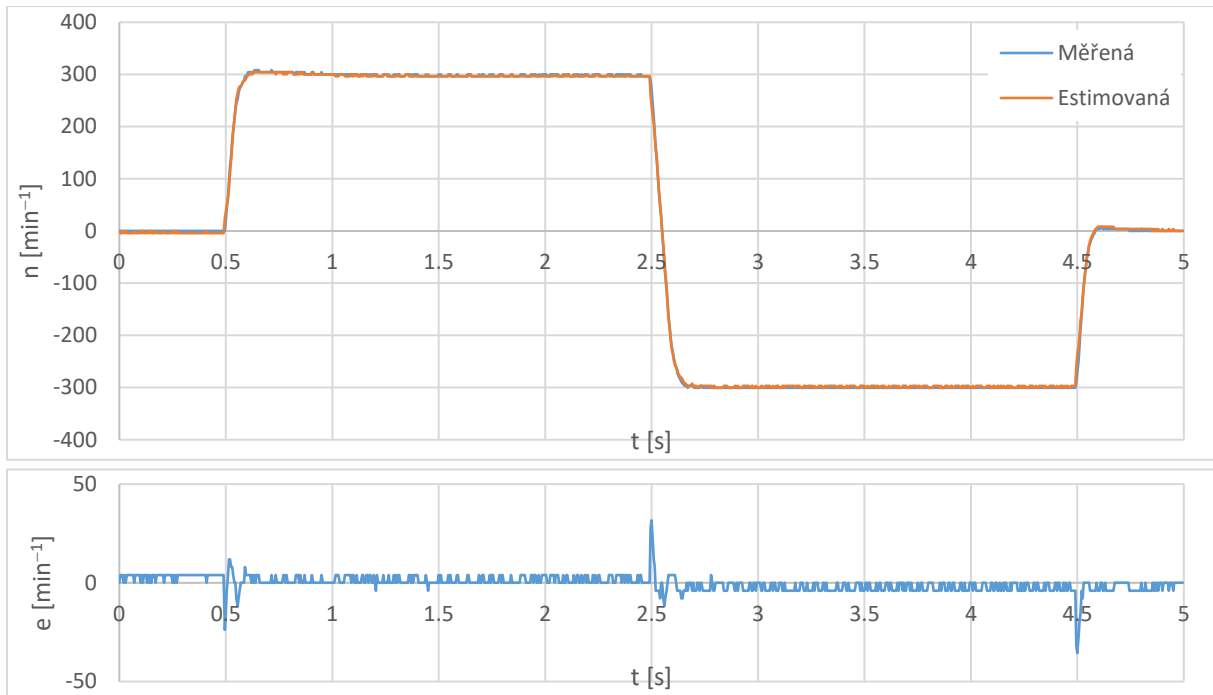
Pro porovnání byly měřeny stejné průběhy, jako s pozorovatelem ANNCB-MRAS $\alpha\beta$. Význam veličin v grafech je stejný jako v předchozí kap. 6.4.

Na Obr. 6.33 až Obr. 6.35 jsou zobrazeny průběhy při skocích žádané rychlosti, které byly měřeny s nezatíženým motorem pro $300 \text{ ot}\cdot\text{min}^{-1}$.

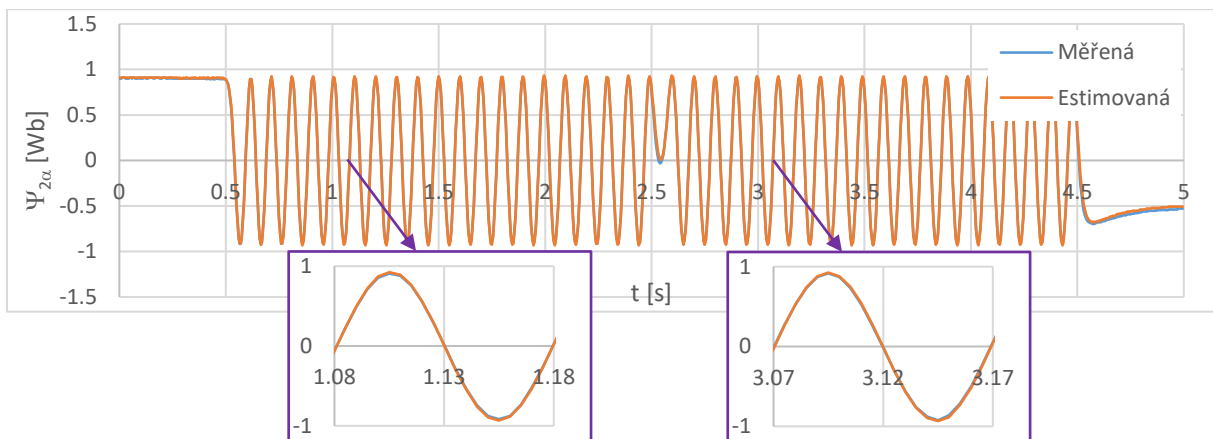
Na Obr. 6.36 až Obr. 6.38 jsou průběhy při skocích žádané rychlosti, které byly měřeny s nezatíženým motorem pro $50 \text{ ot}\cdot\text{min}^{-1}$.

Grafy na Obr. 6.39 až Obr. 6.41 ukazují průběhy při přechodu z motorického do generátorického režimu a zpět pro skoky žádané rychlosti 300 min^{-1} . Motor byl zatížen konstantním momentem, který odpovídal 40 % jmenovitého zatížení.

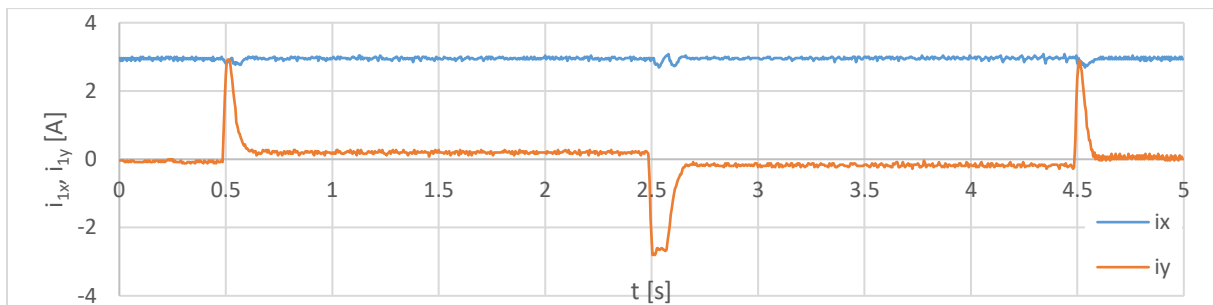
Na Obr. 6.42 až Obr. 6.44 jsou průběhy při přechodu z motorického do generátorického režimu a zpět pro skoky žádané rychlosti $50 \text{ ot}\cdot\text{min}^{-1}$ a 20 % jmenovitého zatížení.



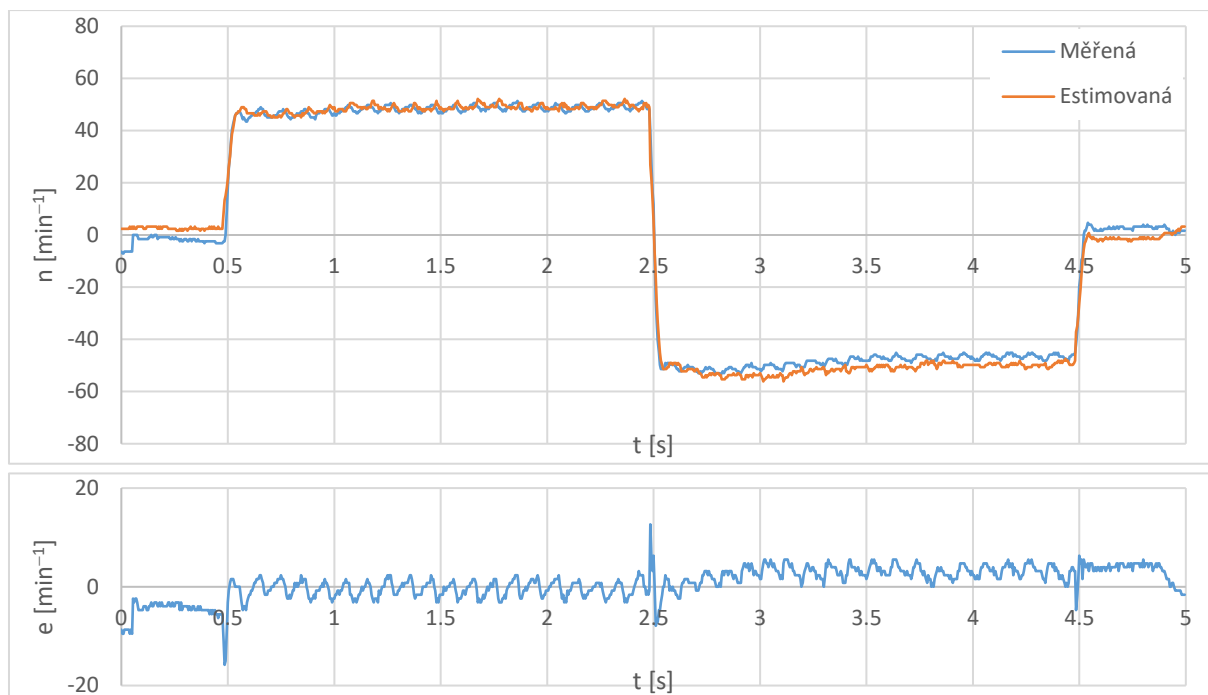
Obr. 6.33 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže



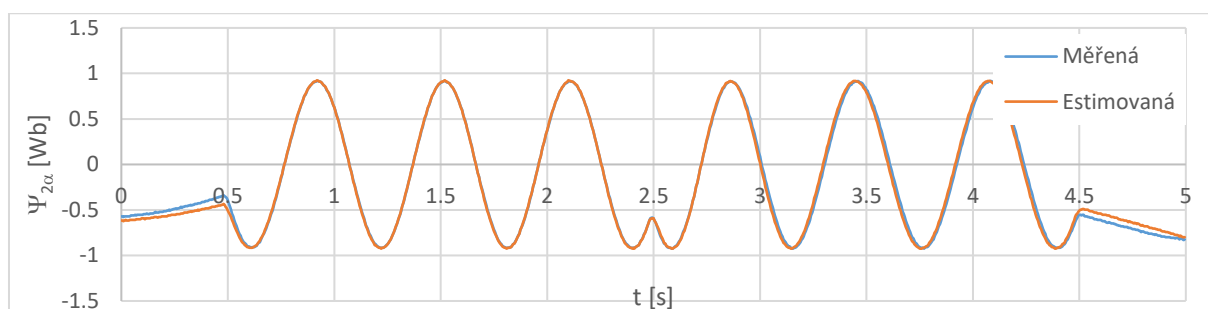
Obr. 6.34 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže



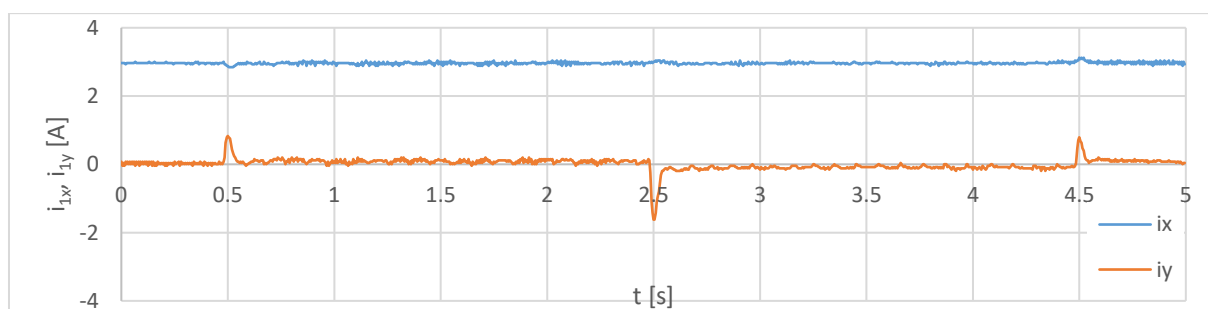
Obr. 6.35 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 300 \text{ min}^{-1}$, bez zátěže



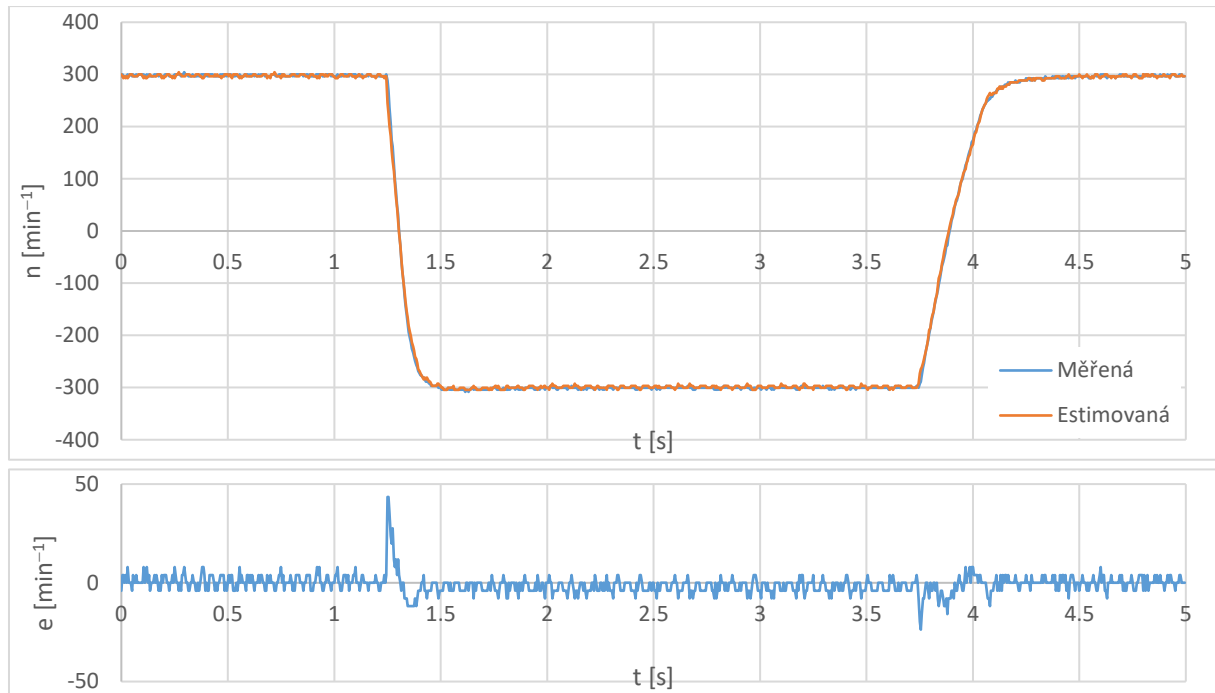
Obr. 6.36 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



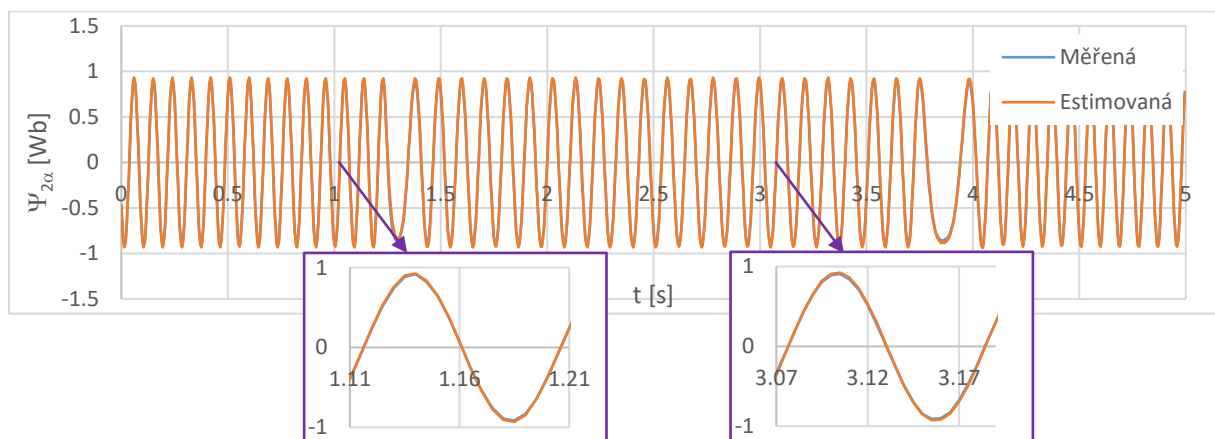
Obr. 6.37 Porovnání rotorových toků v SSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



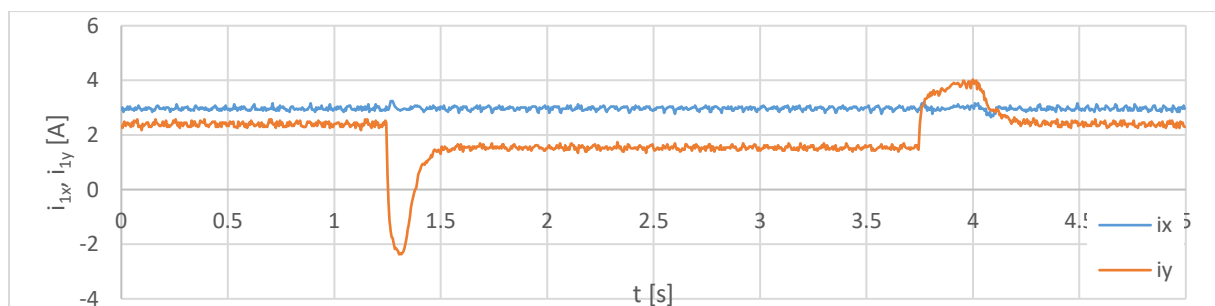
Obr. 6.38 Průběh statorového proudu v OSS pro skoky žádané rychlosti $\pm 50 \text{ min}^{-1}$, bez zátěže



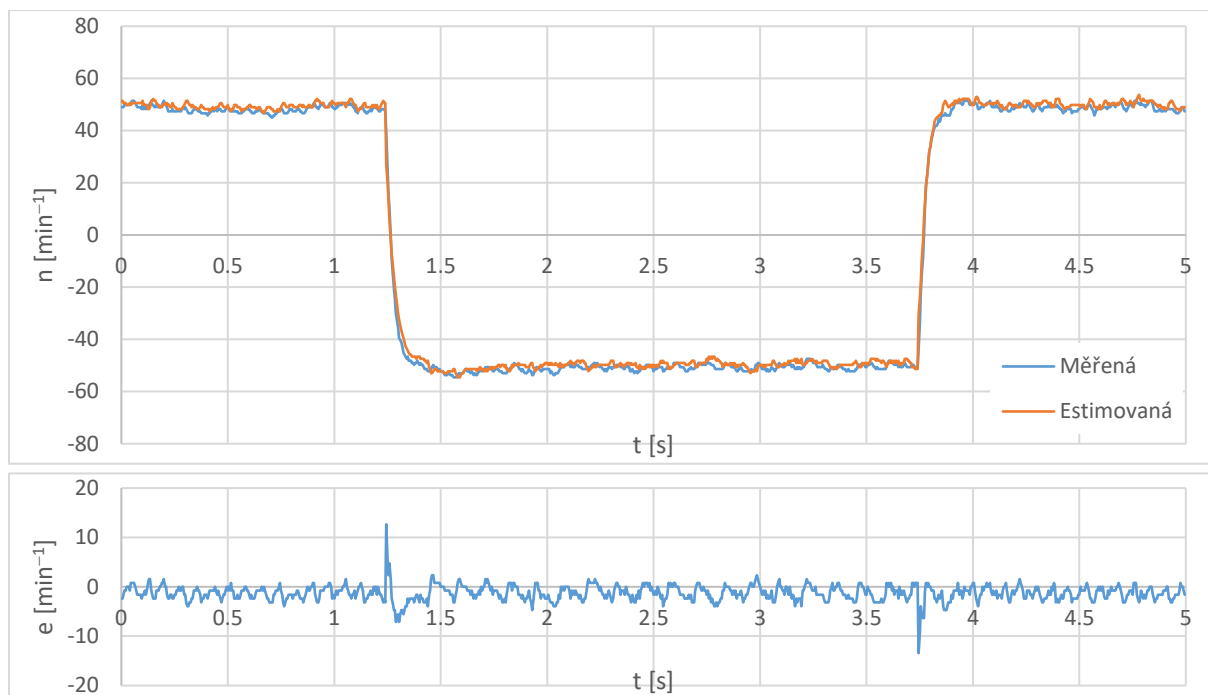
Obr. 6.39 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 300 min^{-1} , $40 \% M_n$



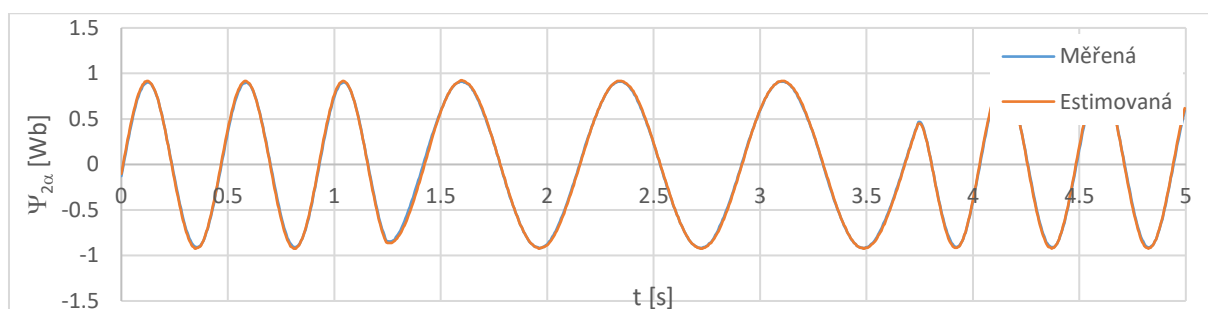
Obr. 6.40 Porovnání rotorových toků v SSS, přechod do gen. režimu, 300 min^{-1} , $40 \% M_n$



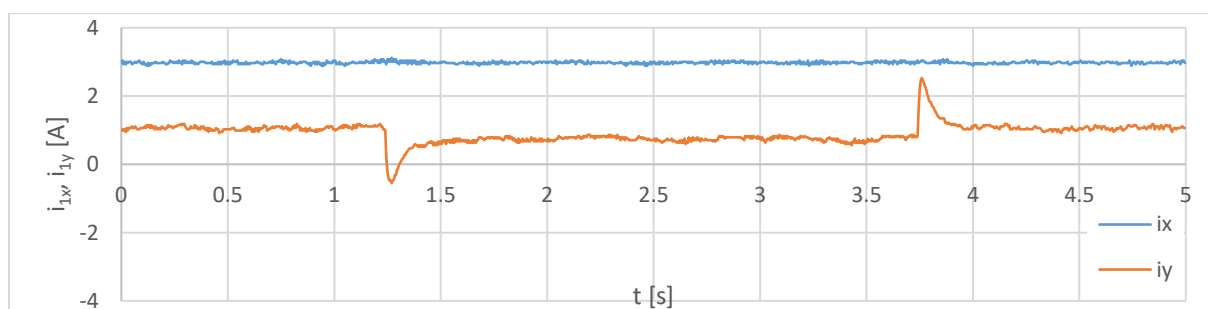
Obr. 6.41 Průběh statorového proudu v OSS, přechod do gen. režimu, 300 min^{-1} , $40 \% M_n$



Obr. 6.42 ANNCB-MRAS xy: měřená a estimovaná rychlost motoru, přechod do gen. režimu, 50 min^{-1} , $20 \% M_n$



Obr. 6.43 Porovnání rotorových toků v SSS, přechod do gen. režimu, 50 min^{-1} , $20 \% M_n$



Obr. 6.44 Průběh statorového proudu v OSS, přechod do gen. režimu, 50 min^{-1} , $20 \% M_n$

7 Shrnutí experimentálních výsledků

Ze srovnání všech čtyř metod podle zkušeností a experimentálních výsledků vychází nejlépe pozorovatel ANNCB-MRAS xy, následuje pozorovatel ANNCB-MRAS $\alpha\beta$ a neuronová síť ANN-EST, na posledním místě se umístil pozorovatel ANNRF-MRAS.

Při použití neuronové sítě na místě referenčního modelu pozorovatele RF-MRAS se potvrdilo zlepšení chování v oblasti nízkých otáček. Celkově však přesnost byla horší než u ostatních metod. Při rychlosti pod $50 \text{ ot}\cdot\text{min}^{-1}$ nebylo možné pohon zatížit větším momentem.

Estimátor ANN-EST byl mnohem přesnější a zvládal zatížení při rychlosti $20 \text{ ot}\cdot\text{min}^{-1}$, některé neuronové sítě dobře estimovaly i při $5 \text{ ot}\cdot\text{min}^{-1}$, v oblasti pod $50 \text{ ot}\cdot\text{min}^{-1}$ se však už více projevoval vliv změny parametrů. Zvláště při velmi nízkých otáčkách byl tento estimátor citlivější na změny teploty motoru než oba pozorovatelé ANNCB-MRAS.

Problematické u generátorického režimu při nízkých otáčkách je mimo jiné to, že se dostáváme do oblasti nižších frekvencí než při stejném zátěžném momentu v motorickém režimu. Výhodou pozorovatele CB-MRAS oproti RF-MRAS je to, že jako referenční veličina se používá přímo měřený statorový proud. S pozorovateli ANNCB-MRAS $\alpha\beta$ a xy bylo možné pohon zatížit více než pohon s ANNRF-MRAS. Pozorovatelé ANNCB-MRAS zvládaly zatížení až do $5 \text{ ot}\cdot\text{min}^{-1}$. Pozorovatel ANNCB-MRAS $\alpha\beta$ však nedosahoval tak dobrého výkonu v generátorickém režimu jako pozorovatel ANNCB-MRAS xy. Navíc pozorovatel ANNCB-MRAS xy fungoval lépe v generátorickém režimu i v případě, že to nebylo zahrnuto v trénovacích vzorech, tzn. že při trénování byly použity jen vzory změřené v motorickém režimu. To se podařilo pouze u této neuronové sítě.

Závěr

Náplní této práce je problematika použití umělých neuronových sítí ve spojení s elektrickými pohony. Konkrétně se práce zaměřuje na pohony s asynchronními motory.

Mezi nejdůležitější vlastnosti neuronových sítí patří schopnost naučit se teoreticky jakékoliv zobrazení mezi vstupem a výstupem bez znalosti matematického modelu systému a adaptovat se na měnící se podmínky. Jsou robustní, to znamená, že jsou méně citlivé na šum a zkreslení vstupních dat a jsou méně závislé na změnách parametrů systému. Pokud jsou neuronové sítě správně natrénované, mají také schopnost zevšeobecnění, díky tomu dávají správné výsledky i pokud se setkají se vstupy, které nebyly součástí trénovacích dat.

Důležitým kritériem při výběru vhodné neuronové sítě je nejen její přesnost, ale také výpočetní náročnost, ať už algoritmu adaptace, nebo výpočtu samotné neuronové sítě. Vysoká přesnost aproximace požadovaného chování bývá vykoupena velkou výpočetní náročností a naopak. Při praktických realizacích elektrických pohonů využívajících umělé neuronové sítě je většinou nutné algoritmy neuronových sítí implementovat do řídicích systémů běžících v reálném čase, které jsou dnes sice velmi výkonné, ale stále nás jejich výpočetní výkon limituje. S rostoucím výpočetním výkonem má toto hledisko menší váhu při rozhodování, jaký druh a jak rozsáhlou neuronovou síť použijeme.

Při simulacích a experimentech se ukázalo, že nízká chyba dosažená při offline trénování není zárukou dobré funkce neuronové sítě v uzavřené smyčce (ANN-EST) nebo v systému MRAS. Při práci s MRAS byly natrénované sítě navíc podrobovány testu chování v systému MRAS na změřených datech. Při hodnocení kvality natrénované sítě je potřeba vzít v úvahu také způsob, jakým se vyhodnocuje schopnost zobecnění sítě, tzn. nalezení obecné závislosti mezi vstupy a výstupy.

Byly navrženy dvě modifikace CB-MRAS s neuronovou sítí na místě proudového estimátoru. Přesnost obou je srovnatelná. Podle experimentálních výsledků se ANNCB-MRAS $\alpha\beta$ choval hůře v generátorickém režimu. Ze všech čtyř aplikací bylo dosaženo nejlepších výsledků se systémem ANNCB-MRAS xy . Ve srovnání s ANNCB-MRAS $\alpha\beta$ se podařilo mnohem jednodušeji natrénovat kvalitní proudový estimátor, s kterým pozorovatel dosahoval vysoké přesnosti. Rovněž v generátorickém režimu byly výsledky lepší. Bylo možné natrénovat síť, která zvládala generátorický režim při jmenovitém momentu zátěže.

Práce se zaměřila na oblast nízkých otáček. Bylo by možné pokračovat např. ověřením vlastností navržených metod pro celý rozsah otáček a pro režim odbuzení nad jmenovitými otáčkami. Zda by bylo možné natrénovat jednu neuronovou síť pro všechny režimy pohonu nebo by bylo potřeba natrénovat více sítí a přepínat mezi nimi v závislosti na aktuálním rozsahu. Také by mohly být otestovány další typy neuronových sítí nebo by mohl být místo použitého PI regulátoru otestován jiný adaptační mechanismus. Také by se mohla podrobněji vyhodnotit stabilita a odolnost vůči změnám parametrů.

Literatura

- [1] P. Vas, *Artificial-intelligence-based electrical machines and drives: application of fuzzy, neural, fuzzy-neural, and genetic-algorithm-based techniques.*, 1st ed. New York: Oxford University Press, 1999.
- [2] P. Vas, *Sensorless vector and direct torque control.* New York: Oxford University Press, 1998.
- [3] J. Heaton, *Introduction to neural networks for C#*, 2nd ed. St. Louis: Heaton Research, 2008.
- [4] S. S. Haykin, *Neural networks: a comprehensive foundation*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, c1999.
- [5] P. Brandštetter, *Střídavé regulační pohony: moderní způsoby řízení.* Ostrava: VŠB-Technická univerzita, 1999.
- [6] I. Neborák, *Modelování a simulace elektrických regulovaných pohonů.* Ostrava: Vysoká škola báňská - Technická univerzita, 2002.
- [7] P. Chamrád, "Optimalizace parametrů pozorovatelů typu MRAS pro bezsenzorové řízení asynchronního motoru", *Disertační práce*, Ostrava, 2019.
- [8] M. Kuchař, "Řídicí aplikace neuronových sítí v elektrických regulačních pohonech", *Disertační práce*, Ostrava, 2003.
- [9] M. Żelechowski, "Space Vector Modulated – Direct Torque Controlled (DTC – SVM) Inverter – Fed Induction Motor Drive", *Disertační práce*, Varšava – Polsko, 2005.
- [10] M. Bednář, "New Approach to Induction Motor Drive Sensorless MRAS Control", *Disertační práce*, Praha, 2014.
- [11] J. Žilková, "Aplikácia umelých neuronových sietí pri pozorovaní veličín ASM.", *Disertační práce*, Košice, 2000.
- [12] M. Skotnica, "Aplikace neuronových sítí v elektrických regulačních pohonech.", *Disertační práce*, Ostrava, 1999.
- [13] E. Volná, "Vícevrstvá neuronová síť jako univerzální aproximátor", *Automatizace*, vol. 52, no. 11, pp. 658-660, 2009.
- [14] P. Brandstetter and M. Kuchar, "Sensorless control of variable speed induction motor drive using RBF neural network", *Journal of Applied Logic*, vol. 24, pp. 97-108, 2017. doi:10.1016/j.jal.2016.11.017
- [15] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, "Recent developments of induction motor drives fault diagnosis using AI techniques", *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 994-1004. doi:10.1109/41.873207
- [16] R. N. Mishra, K. B. Mohanty, K. Thakre, and P. R. Sahu, "Design of a Simplified Neuro-Fuzzy-GA-based IM Drive Deploying Linearization Approach", in *2017 14th IEEE India Council International Conference (INDICON)*, 2017, pp. 1-6. doi:10.1109/INDICON.2017.8487488

- [17] A. K. Alexandridis and A. D. Zaprani, "Wavelet neural networks: A practical guide", *Neural Networks*, vol. 42, pp. 1-27, 2013. doi:10.1016/j.neunet.2013.01.008
- [18] Cheng-Zhi Cao, Mu-Ping Lu, and Xin Wang, "Speed estimation and stimulation of DTC system based on wavelet neural network", in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, 2003, pp. 755-759. doi:10.1109/ICMLC.2003.1259577
- [19] N. Pimkumwong and M. -S. Wang, "Online Speed Estimation Using Artificial Neural Network for Speed Sensorless Direct Torque Control of Induction Motor based on Constant V/F Control Technique", *Energies*, vol. 11, no. 8, 2018. doi:10.3390/en11082176
- [20] N. Torabi, V. M. Sundaram, and H. A. Toliyat, "On-line fault diagnosis of multi-phase drives using self-recurrent wavelet neural networks with adaptive learning rates", in *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2017, pp. 570-577. doi:10.1109/APEC.2017.7930751
- [21] H. Al-Behadili, M. E. Dubakh, and A. Mohsen Abbass, "Three-phase Induction Motor Speed Estimator Using Genetic Algorithm Based on Elman Recurrent Neural Network", in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2019, pp. 1-4. doi: 10.1109/ISMSIT.2019.8932948
- [22] L. M. Grzesiak, V. Meganck, J. Sobolewski, and B. Ufnalski, "Genetic Algorithm for Parameters Optimization of ANN-based Speed Controller", in *EUROCON 2007 - The International Conference on "Computer as a Tool"*, 2007, pp. 1700-1705. doi: 10.1109/EURCON.2007.4400689
- [23] X. Yang and X. Zheng, "Gradient Descent Algorithm-Based Adaptive NN Control Design for an Induction Motor", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-8. doi:10.1109/TSMC.2019.2894661
- [24] J. Wang, P. Fu, L. Zhang, R. X. Gao, and R. Zhao, "Multilevel Information Fusion for Induction Motor Fault Diagnosis", *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2139-2150, 2019. doi:10.3390/en11082176
- [25] R. Verma, V. Verma and C. Chakraborty, "ANN based sensorless vector controlled induction motor drive suitable for four quadrant operation", in *Proceedings of the 2014 IEEE Students' Technology Symposium*, 2014, pp. 182-187. doi:10.1109/TechSym.2014.6808043
- [26] A. Venkadesan, S. Himavathi, A. Muthuramalingam and K. Sedhuraman, "ANN based Flux Estimator for Rotor Resistance Estimation in Vector Controlled IM Drives", in *2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA)*, 2018, pp. 1-5. doi:10.1109/ICSCAN.2018.8541240
- [27] S. V. B. S. Reddy, B. Kumar, and D. Swaroop, "Investigations on Training Algorithms for Neural Networks Based Flux Estimator Used in Speed Estimation of Induction Motor", in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2019, pp. 1090-1094. doi:10.1109/SPIN.2019.8711623

- [28] P. Girovsky, J. Timko, J. Zilkova, and V. Fedak, "Neural estimators for shaft sensorless FOC control of induction motor", in Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010, 2010, p. -. doi:10.1109/EPEPEMC.2010.5606907
- [29] T. Orłowska-Kowalska and C. T. Kowalski, "Neural network application for flux and speed estimation in the sensorless induction motor drive", in ISIE '97 Proceeding of the IEEE International Symposium on Industrial Electronics, 1997, pp. 1253-1258. doi: 10.1109/ISIE.1997.648923
- [30] P. Vas, "Implementation of ANN-based sensorless induction motor drives", in 9th International Conference on Electrical Machines and Drives, 1999, pp. 329-333. doi:10.1049/cp:19991045
- [31] J. Holtz, "Sensorless control of induction motor drives", Proceedings of the IEEE, vol. 90, no. 8, pp. 1359-1394, 2002. doi:10.1109/JPROC.2002.800726
- [32] P. Syam, R. Kumar, S. Das, and A. K. Chattopadhyay, "Review on model reference adaptive system for sensorless vector control of induction motor drives", IET Electric Power Applications, vol. 9, no. 7, pp. 496-511, Aug. 2015. doi:10.1049/iet-epa.2014.0220
- [33] Li Yi and Han Mei, "Speed sensorless IM control system based on MRAS and NN flux observers", in 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 2010, pp. V5-163-V5-167. doi: 10.1109/ICACTE.2010.5579145
- [34] H. H. Vo, P. Brandstetter, C. S. T. Dong, and T. C. Tran, "Speed Estimators Using Stator Resistance Adaptation for Sensorless Induction Motor Drive", Advances in Electrical and Electronic Engineering, vol. 14, no. 3, pp. 267 - 273, Sep. 2016. doi:10.15598/aeec.v14i3.1732
- [35] A. Venkadesan, S. Himavathi, A. Muthuramalingam, "Novel SNC-NN-MRAS based speed estimator for sensor-less vector controlled IM drives", in World Academy of Science, Engineering and Technology, 2011, no. 51, pp. 1211-1216. doi:doi.org/10.5281/zenodo.1335432
- [36] A. Tiwari, B. Kumar, and Y. K. Chauhan, "ANN based RF-MRAS speed estimation of induction motor drive at low speed", in 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017, pp. 176-179. doi:10.1109/ICECA.2017.8212789
- [37] M. Cirrincione and M. Pucci, "An MRAS-Based Sensorless High-Performance Induction Motor Drive With a Predictive Adaptive Model", IEEE Transactions on Industrial Electronics, vol. 52, no. 2, pp. 532-551, 2005. doi:10.1109/TIE.2005.844247
- [38] M. Cirrincione, M. Pucci, G. Cirrincione, and G. -A. Capolino, "A new TLS based MRAS speed estimation with adaptive integration for high performance induction machine drives", in 38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003, 2003, pp. 140-151. doi: 10.1109/IAS.2003.1257496
- [39] S. M. Gadoue, D. Giaouris, and J. W. Finch, "Sensorless Control of Induction Motor Drives at Very Low and Zero Speeds Using Neural Network Flux Observers", IEEE Transactions on Industrial Electronics, vol. 56, no. 8, pp. 3029-3039, 2009. doi:10.1109/TIE.2009.2024665
- [40] S. M. Gadoue, D. Giaouris, and J. W. Finch, "Performance Evaluation of a Sensorless Induction Motor Drive at Very Low and Zero Speed Using a MRAS Speed Observer", in 2008 IEEE Region

- 10 and the Third international Conference on Industrial and Information Systems, 2008, pp. 1-6. doi:10.1109/ICIINFS.2008.4798413
- [41] S. M. Gadoue, D. Giaouris, and J. W. Finch, "A neural network based stator current MRAS observer for speed sensorless induction motor drives", in 2008 IEEE International Symposium on Industrial Electronics, 2008, pp. 650-655. doi:10.1109/ISIE.2008.4677079
- [42] M. Kuchar, P. Brandstetter, and M. Kaduch, "Sensorless induction motor drive with neural network", in 2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551), 2004, pp. 3301-3305. doi:10.1109/PESC.2004.1355058
- [43] A. Havel, M. Sobek, and P. Chamrad, "Control methods of modern systems utilizing accumulation of electrical energy", in 2017 18th International Scientific Conference on Electric Power Engineering (EPE), 2017, pp. 1-6. doi:10.1109/EPE.2017.7967338

Publikační činnost

- [I] D. Kouřil and J. Bača, "An ARM-based Embedded Microcontroller System Development Platform", in *WOFEX 2019: 17th annual workshop, Ostrava, 10th September 2019: proceedings of papers*, Ostrava: Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 2019, pp. 60-65. **(článek ve sborníku konference)**
<http://www.cs.vsb.cz/arg/worconf/wofex/proceedings/Wofex2019.pdf>
- [II] J. Baca, M. Kuchar, and P. Palacky, "Effect of Weighting Coefficients on Behavior of the DTC Method with Direct Calculation of Voltage Vector", in *AETA 2018 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*, Cham: Springer International Publishing, 2020, pp. 683-692. doi:10.1007/978-3-030-14907-9_66
(článek ve sborníku konference, Scopus, Q3(2018), SJR 0,134(2018))
https://doi.org/10.1007/978-3-030-14907-9_66
- [III] J. Bača, "An Implementation of the DTC Method with Direct Calculation of Voltage Vector", in *WOFEX 2018: 16th annual workshop, Ostrava, 10th September 2018: proceedings of papers*, Ostrava: Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 2018, pp. 57-62. **(článek ve sborníku konference)**
<http://www.cs.vsb.cz/arg/worconf/wofex/proceedings/Wofex2018.pdf>
- [IV] D. Kouřil and J. Bača, "Implementation of Genetic Algorithm in Control Structure of a DC Motor Drive", in *WOFEX 2020: 18th annual workshop, Ostrava, 8th September 2020: proceedings of papers*, Ostrava: Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 2020, pp. 50-55. **(článek ve sborníku konference)**
<http://dSPACE.vsb.cz/handle/10084/141796>
- [V] J. Bača, P. Chamrád, and D. Kouřil, "Sensorless Induction Motor Drive Based on CB-MRAS Observer", in *WOFEX 2018: 16th annual workshop, Ostrava, 10th September 2018: proceedings of papers*, Ostrava: Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 2018, pp. 63-68. **(článek ve sborníku konference)**
<http://www.cs.vsb.cz/arg/worconf/wofex/proceedings/Wofex2018.pdf>
- [VI] J. Strossa, V. Damec, M. Sobek, D. Kouril, and J. Baca, "High Frequency Multipurpose SiC MOSFET Driver", *Advances in Electrical and Electronic Engineering*, vol. 19, no. 3, pp. 203 - 211, Oct. 2021. doi:10.15598/aeer.v19i3.4159 **(článek v časopise, Scopus, Q3, SJR 0,225)**
<http://advances.utc.sk/index.php/AEEE/article/view/4159>
- [VII] J. Baca, D. Kouril, P. Palacky, and J. Strossa, "Induction motor drive with field-oriented control and speed estimation using feedforward neural network", in *2020 21st International Scientific Conference on Electric Power Engineering (EPE)*, 2020, pp. 1-6.
doi:10.1109/EPE51172.2020.9269215. **(příspěvek na konferenci IEEE)**
<https://ieeexplore.ieee.org/document/9269215>
- [VIII] J. Bača and D. Kouřil, "Implementation of Induction Motor Speed Estimator Using Feedforward Neural Network", in *WOFEX 2019: 17th annual workshop, Ostrava, 10th September 2019: proceedings of papers*, Ostrava: Faculty of Electrical Engineering and Computer Science, VSB –

- Technical University of Ostrava, 2019, pp. 48-53. **(článek ve sborníku konference)**
<http://www.cs.vsb.cz/arg/worconf/wofex/proceedings/Wofex2019.pdf>
- [IX] J. Bača and D. Kouřil, "Induction motor drive with RF-MRAS speed observer based on a neural network", in *WOFEX 2020: 18th annual workshop, Ostrava, 8th September 2020: proceedings of papers*, Ostrava: Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava, 2020, pp. 44-49. **(článek ve sborníku konference)**
<http://dspace.vsb.cz/handle/10084/141796>
- [X] D. Kouril, J. Baca, M. Sobek, M. Kuchar, and J. Strossa, "Data Acquisition System for the Modern Induction Motor Drive applications", in *2020 21st International Scientific Conference on Electric Power Engineering (EPE)*, 2020, pp. 1-5. doi:10.1109/EPE51172.2020.9269184
(příspěvek na konferenci IEEE)
<https://ieeexplore.ieee.org/document/9269184>

Řešené projekty

- SP2021/70 – Výzkum moderních metod řízení střídavých regulovaných pohonů, 2021, člen řešitelského týmu
- RPP2020/149 – Estimátor s umělou neuronovou sítí pro bezsenzorové řízení asynchronního motoru – inovace laboratorních úloh předmětu Elektrické regulované pohony III, 2020, odpovědný řešitel
- SP2020/128 – Výzkum a vývoj sofistikovaných metod řízení pro oblast elektrických regulovaných pohonů, 2020, člen řešitelského týmu
- RPP2019/63 – Inovace mikropočítačového řídicího systému pro potřeby výuky, 2019, člen řešitelského týmu
- SP2019/113 – Výzkum a vývoj moderních metod řízení v oblasti elektrických regulovaných pohonů, 2019, člen řešitelského týmu
- IRP/2018/227 – Modernizace stanoviště pro sběr dat v elektrických regulovaných pohonech, 2018, odpovědný řešitel
- SP2018/162 – Výzkum a vývoj pokročilých metod řízení elektrických regulovaných pohonů, 2018, člen řešitelského týmu
- SP2017/104 – Vývoj a implementace sofistikovaných algoritmů pro elektrické regulované pohony, 2017, člen řešitelského týmu