TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY

A MEZIOBOROVÝCH

INŽENÝRSKÝCH STUDIÍ

DIPLOMOVÁ PRÁCE

Využití umělých neuronových sítí pro diagnostiku vad elektrických indukčních motorů

Liberec 2003 Jaroslav Langpaul

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA MECHATRONIKY
A MEZIOBOROVÝCH
INŽENÝRSKÝCH STUDIÍ

Studijní program : 2612T Elektrotechnika a informatika

Studijní obor : Mechatronika

Využití umělých neuronových sítí pro diagnostiku vad elektrických indukčních motorů

(Application artificial neural network for diagnosis of the induction motors faults)

Autor: Jaroslav Langpaul

Adresa: Jítrava 77
46353, Křižany

Vedoucí práce: Doc. Ing. Ivan Jaksch, CSc.

Počet

Stran	obrázků	Tabulek
42	24	6

V Liberci dne 13.01.2003

Anotace:

Cílem diplomové práce bylo zjistit možnost použití umělých neuronových sítí pro diagnostiku vad asynchronních motorů. Byly naměřeny průběhy proudů u jednotlivých motorů s předem definovanými vadami, které sloužily jako množina dat pro naučení sítě. Byly vytvořeny čtyři umělé neuronové sítě s různým předzpracováním dat. Vstupem do sítě je ve dvou případech frekvenční spektrum proudu a v dalších dvou spektrum magnitudy Parkovy transformace. K jejich vytvoření a natrénování byl použit matematický program MATLAB. Výsledkem jsou naučené sítě, která jsou schopny rozpoznat naučené vady motoru. Pro použití v praxi lze síť exportovat a použít samostatně (např. jako součást expertních systémů).

Abstract:

The aim of the diploma thesis was to find out the possibility of artificial neural networks use for diagnosis of the induction motor faults. There was measured time behavior of electrical current for each motor with predefined certain faults. These measured characteristics were used as set of data for neural network training. There were four neural networks created with different data preprocessing. Two of them used frequency spectrum of current and two others the magnitude spectrum of Park's transformation. To create and to train the neural networks was used mathematical program MATLAB. Product of work are trained artificial neural networks, which are able to diagnose learned motor faults. For practical use can the networks be exported and used independently on MATLAB (e.g. part of expert system).

Prohlášení o původnosti práce:

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a že jsem uvedl veškerou použitou literaturu.

V Liberci dne: 13.01.2003 Jaroslav Langpaul

Prohlášení k využívání výsledků DP:

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském zejména § 60 (školní dílo).

Beru na vědomí, že Technická univerzita v Liberci (TUL) má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **souhlasím** s případným užitím mé diplomové práce (prodej, zapůjčení, kopírování, apod.).

Jsem si vědoma toho, že: užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše). Diplomová práce je majetkem školy, s diplomovou prací nelze bez svolení školy disponovat.

Autor:
Jaroslav Langpaul

Adresa:
Jítrava 77
46353, Křižany

Podpis:

Datum:
13.01.2003

Obsah:

Obsah: ... 7
1. Asynchronní motor ... 9
1.1 Princip činnosti ... 9
1.2 Vady asynchronních motorů ... 10
1.2.1 Přerušené rotorové tyče ... 10
1.2.2 Dynamická excentricita ... 12
1.2.3 Zkratované závity na statorovém vinutí ... 12
2. Metody analýzy proudu ... 14
2.1 Rozklad signálů na frekvenční složky ... 14
2.1.1 Fourierovy řady ... 14
2.1.2 Fourierova transformace ... 14
2.1.3 Diskrétní Fourierova transformace ... 15
2.2 Použití Parkových vektorů ... 16
2.2.1 Parkova transformace ... 16
2.2.2 Užití Parkových vektorů pro diagnostiku vad ... 17
3. Umělé neuronové sítě ... 19
3.1 Úvod do neuronových sítí ... 19
3.2 Model neuronu ... 19
3.3 Přenosová funkce neuronu ... 21
3.4 Typy neuronových sítí ... 22
3.5 Učení neuronové sítě ... 25
3.6 Neuronová síť typu backpropagation ... 25
4. Experimentální část ... 29
4.1 Popis měřícího pracoviště ... 29
4.2 Měření na asynchronních motorech ... 30
4.2.1 Měření zkratovaných závitů na statorovém vinutí ... 30
4.2.2 Měření přerušených rotorových tyčí ... 32
4.3 Aplikace umělé neuronové sítě ... 33
4.3.1 Příprava dat a vytvoření umělé neuronové sítě ... 33
4.3.2 Umělá neuronová síť vytvořená v programu MATLAB® ... 34
4.3.3 Rozpoznávání závady rotoru pomocí FFT ... 37

4.3.4 Rozpoznávání závad rotoru a statoru pomocí FFT 40
4.4 Využití Parkových vektorů pomocí neuronových sítí 44
4.4.1 Využití spektra magnitudy 44
4.4.2 Využití význačných bodů ze spektra magnitudy 46
5. Závěr 50
Literatura: 51
Přílohy: 52

1. Asynchronní motor

1.1 Princip činnosti

Asynchronní motor je nejrozšířenější elektrický točivý stroj. Jeho konstrukce se skládá ze dvou částí, pevného statoru a pohyblivého rotoru.

Stator má magnetický obvod složený ze vzájemně izolovaných plechů a v jeho drážkách je uloženo trojfázové vinutí. Toto vinutí je vyvedeno na svorkovnici motoru.

Rotor je hřídel s magnetickým obvodem, který je také složen ze vzájemně izolovaných plechů a v jeho drážkách je odlita hliníková klec nakrátko, nebo je tam uloženo trojfázové vinutí vyvedené na tři kroužky.

Stator při provozu vytváří točivé magnetické pole otáčející se synchronní rychlostí

$$n_{SPM} = \frac{2 \cdot 60 \cdot f_{line}}{p} \quad [\text{min}^{-1}] \tag{1.1}$$

kde f_{line} ... frekvence napájecí sítě [Hz]

p ... počet pólů

n_{SPM} ... synchronní otáčky [min^{-1}]

Indukční čáry tohoto pole protínají vodiče rotoru. Na těch se indukuje napětí a v uzavřeném elektrickém obvodě vyvolá vznik proudu. Díky tomu rotor vytváří vlastní magnetické pole interagující s polem statoru. Vzniká síla umožňující roztočit rotor.

Rychlost otáčení rotoru je nižší než rychlost točivého magnetického pole. Při shodné rychlosti by se do rotoru přestalo indukovat napětí a síla na něj působící by přestala existovat.

$$n_{RPM} = \frac{2 \cdot 60 \cdot f_{line}}{p} \cdot (1 - s) \quad [\text{min}^{-1}] \tag{1.2}$$

$$s = \frac{n_{SPM} - n_{RPM}}{n_{SPM}} \tag{1.3}$$

kde n_{RPM} ... otáčky rotoru [min^{-1}]

s ... skluz

1.2 Vady asynchronních motorů

Vady asynchronních motorů nejvíce ovlivňují chování a životnost motorů Závady lze rozdělit do dvou hlavních skupin, mechanické a magnetické.[1]

Mechanické vady jsou poruchy strojního charakteru. Patří sem především

- nesouosost (osa rotoru není na ose otvoru)
- špatná vyváženost (hmota rotoru není homogenně rozložena kolem osy)
- vadná ložiska (zvýšený odpor při otáčení rotoru)

Magnetická závada se projeví narušením vazeb mezi statorem a rotorem. Ovlivní magnetické pole ve vzduchové mezeře motoru. Mezi magnetické vady patří

- statická excentricita
- dynamická excentricita
- přerušené či uvolněné rotorové tyče
- vyšší odpor rotorových tyčí
- přerušený rotorový věnec
- závity nakrátko ve vinutí statoru nebo rotoru
- zkratované nebo uvolněné rotorové či statorové plechy

Popíšeme si postupně projevy některých magnetických vad.

1.2.1 Přerušené rotorové tyče

Tato porucha se většinou vyskytuje u motorů s častým rozběhem u kterých dochází k většímu tepelnému namáhání díky vysokým rozběhovým proudům. Nebo u motorů s větším mechanickým zatížením. Je nutné diagnostikovat tuto poruchu, neboť má tendenci se rozšiřovat z důvodu vyššího namáhání ostatních tyčí.

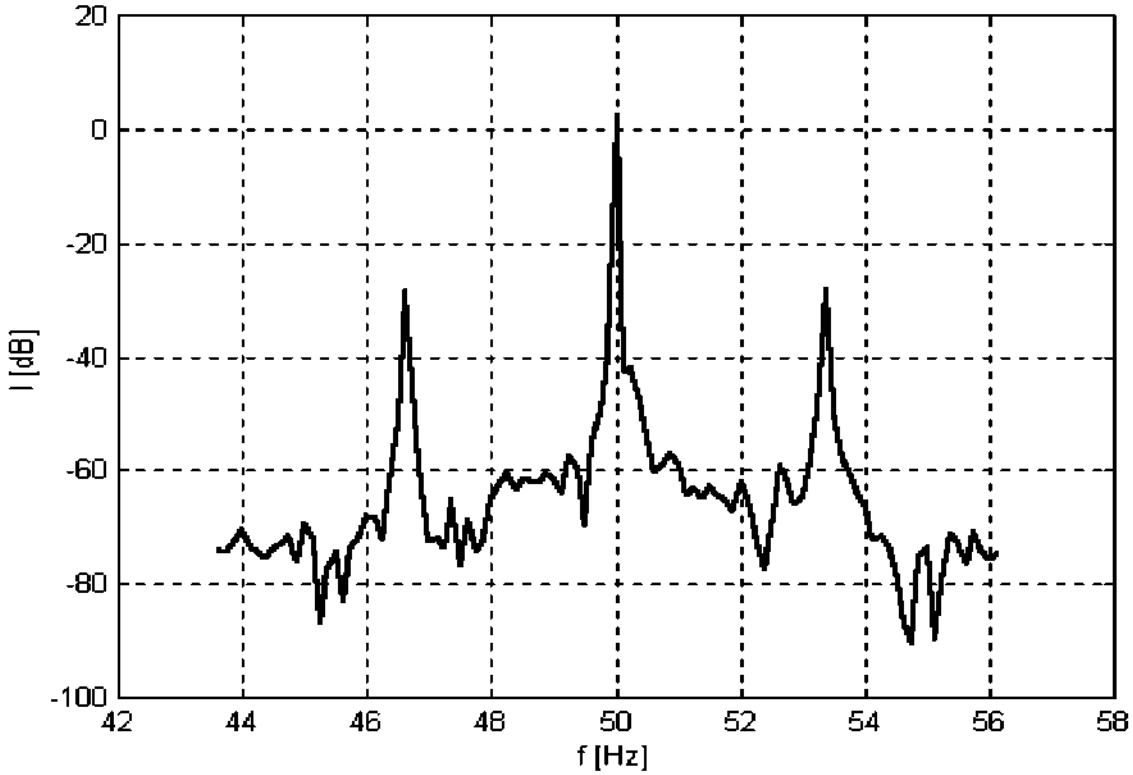
U takto postiženého rotoru dochází k narušení původně symetrického magnetického pole ve vzduchové mezeře. Projeví se jako amplitudová modulace, nosná frekvence je frekvence síťová a modulační má hodnotu dvojnásobku kmitočtu rotoru.

Deformace magnetického pole vyvolá vznik nevyvážené síly. Vibrace touto silou vyvolané se projeví ve spektru proudu jako dvě stejná postranní pásma u základní (síťové) frekvence.

$$f_{brs} = f_{line} \pm 2 \cdot s \cdot f_{line} \tag{1.4}$$

kde f_{brs} ... frekvence na které se projevuje přerušená rotorová tyč

Postranní pásma se vyskytují vždy, protože nelze vyrobit motor který by měl odpor všech rotorových tyčí úplně stejný. Velikost postranního pásma u motoru bez vad téměř zaniká v šumu (odstup okolo 60 dB). Při odstupu nižším než 50 dB se dá předpokládat přerušená rotorová tyč.



Obr1.1: Příklad FFT motoru s přerušenou rotorovou tyčí

Na *Obr 1.1* lze vidět odstup postranního pásma, který se pohybuje okolo 30 dB. Tento projev signalizuje přerušenou rotorovou tyč. Rozdíl mezi základní frekvencí a postranním pásmem je 3,3 Hz a nazýváme ho skluzová pólová frekvence ($f_{slip}=2*s*f_{line}$)

1.2.2 Dynamická excentricita

Dynamická excentricita je závada při níž se za provozu mění vzduchová mezera mezi statorem a rotorem. Vzniká především v motorech nemající přesně kruhový rotor, či je-li rotor po své délce prohnut. Může také vzniknout při poruše ložisek, ve kterých je uložen rotor a ten při otáčení mění svoji osu rotace.

U běžných motorů je velikost vzduchové mezery v řádech desetin milimetrů. V některých případech při této závadě může docházet odírání rotoru o stator. To má fatální následky pro chod motoru, často vede až k jeho úplnému zničení.

S dynamickou změnou vzduchové mezery se bude měnit magnetická indukce a projeví se, stejně jako u přerušených rotorových tyčí, vznikem postranních pásem. Účinky této poruchy se objevují modulací synchronních otáček frekvencí otáčení rotoru.

$$f_{de} = f_{line} \pm f_r \tag{1.5}$$

kde f_r ... frekvence otáčení rotoru

$$f_r = \frac{n_{RPM}}{60}$$

1.2.3 Zkratované závity na statorovém vinutí

Tato porucha vzniká porušením izolace mezi závity statorového vinutí. Nejdříve dochází k porušení izolace mezi dvěma závity a k jejich následnému zkratu. Díky vyššímu tepelnému namáhání dochází poškozování okolní izolace a ke zvětšování závady. Proto se hledají cesty k detekci této vady ještě v počátku vzniku .

Při zkratování několika závitů statorového vinutí dochází k narušení symetrie elektromagnetického pole. Vzhledem k tomu, že statorové vinutí je pevně uloženo v drážkách statoru, je tato chyba statická a nerotuje.

Tato vada se projevuje ve spektru proudu zvýšením amplitudy základní frekvence a jejích harmonických frekvencí

Změna amplitudy je malá a ze spektra proudu nelze s určitostí diagnostikovat tuto poruchu pro všechny motory. U malých motorů lze tuto závadu zjistit při plném napájení a při úplném zastavení motoru.

Existuje několik metod pro určení této závady, jedna z nich je i diagnostika pomocí Parkových vektorů, kde se projeví na dvojnásobku síťové frekvence. (kap 2.2) [2][3]

2. Metody analýzy proudu

2.1 Rozklad signálů na frekvenční složky

2.1.1 Fourierovy řady

Za signál považujeme každou proměnou jejíž hodnota je závislá na čase. Periodický signál se po určitých časových úsecích opakuje.

Základem analýzy signálu je věta pocházející od J.Fouriera. Podle ní je možné rozložit reálný periodický signál s periodou f (f=ω/2π) na řadu s harmonickými složkami:

$$x(t) = \sum_{k=0}^{\infty} c_k \cdot \cos(k \cdot \omega_0 \cdot t + \varphi_k) \qquad (2.1a)$$

ω_0 … základní frekvence

V literatuře [4] se ještě uvádějí jiné tvary této řady.

$$x(t) = a_0 + \sum_{k=1}^{\infty} [a_k \cos(k \cdot \omega_0 \cdot t) + b_k \sin(k \cdot \omega_0 \cdot t)] \qquad (2.1b)$$

$$x(t) = \sum_{k=0}^{\infty} X_k \cdot e^{j\omega_0 \cdot t} \qquad (2.1c)$$

Výpočet koeficientů a vztahy mezi nimi:

$$a_k = \int x(t) \cdot \cos(k \cdot \omega_0 \cdot t) dt \qquad X_k = \frac{1}{2}(a_k - jb_k)$$

$$b_k = \int x(t) \cdot \sin(k \cdot \omega_0 \cdot t) dt \qquad c_k = 2|X_k|$$

$$X_k = \int x(t) \cdot e^{-jk \cdot \omega_0 \cdot t} dt \qquad \varphi_k = arctg \frac{b_k}{a_k}$$

2.1.2 Fourierova transformace

Fourierova transformace umožňuje převést signál z časové roviny do roviny frekvenční. [4][5] Rozšiřuje oproti Fourierovým řadám analýzu i na signály s neperiodickým průběhem. Jediná podmínka je, aby signál měl konečnou energii.

Fourierova transformace je definována následovně:

Je-li funkce x(t) po částech spojitá na intervalu (-∞,∞) a také je-li její derivace na intervalu (-∞,∞) po částech spojitá. Dále musí platit:

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty$$

Pak pro všechna ω existuje integrál:

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \tag{2.2}$$

Funkce X(jω) představuje Fourierův obraz funkce x(t). Jsme schopni sestavit zpět původní signál pomocí zpětné Fourierovi transformace,

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega \tag{2.3}$$

2.1.3 Diskrétní Fourierova transformace

Při zpracování pomocí počítačů musíme pracovat s konečným počtem vzorků. Používají se tedy diskrétní průběhy. Signály v oblasti času a frekvence mají konečný počet hodnot N a při výpočtech se považují za periodické. Pro popis diskrétních signálů se používá upravených vztahů pro Fourierovu transformaci (2.2) a (2.3).

V diskrétní Fourierově transformaci (DFT) je integrál nahrazen sumou:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N} \cdot n \cdot k} \quad , \quad k = 0,1,2,\ldots, N-1 \tag{2.4}$$

Označíme-li čas měření T, počet vzorků N a periodu vzorkování Δt, pak odstup koeficientů je:

$$\Delta f = \frac{1}{T} = \frac{1}{N \cdot \Delta t}$$

Nepřímá (inverzní) Fourierova transformace je:

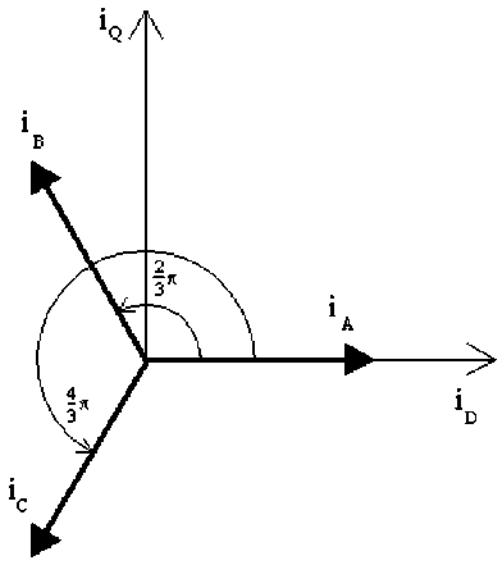
$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{j\frac{2\pi}{N} \cdot n \cdot k} \quad , \quad n = 0,1,2,\ldots, N-1 \tag{2.5}$$

Pro výpočet této transformace byly vypracovány efektivní algoritmy určené pro výpočet na číslicových počítačích. Říká se jim rychlá Fourierova transformace, zkratka je FFT (z angl. Fast Fourier Transformation). Díky vysoké výpočtové rychlosti se FFT stala velmi důležitým nástrojem v měřící technice.

2.2 Použití Parkových vektorů

2.2.1 Parkova transformace

Základem Parkovy transformace je převedení tří fázových proudů na dva, které vytvářejí rotující vektor. [2][3][6] Mějme tři fázové proudy i_A, i_B, i_C, které převedeme na proudy i_D, i_Q. (Obr 2.1)



Obr. 2.1: Převod na Parkův vektor.

Vektor je promítnut do roviny xy. Jeho reálná část je proud i_D a imaginární část je proud i_Q. Základní podmínkou je, aby amplitudy všech tří proudů (i_A, i_B, i_C) byly stejné. Pak převod na Parkův vektor je definován následně:

$$i_D = i_A + i_B \cos\left(\frac{2\pi}{3}\right) + i_C \sin\left(\frac{2\pi}{3}\right) = i_A - \frac{1}{2} i_B - \frac{1}{2} i_C \qquad (2.6)$$

$$i_Q = i_B \sin\left(\frac{4\pi}{3}\right) + i_C \sin\left(\frac{4\pi}{3}\right) = \frac{\sqrt{3}}{2} i_B - \frac{\sqrt{3}}{2} i_C \qquad (2.7)$$

Za ideálních podmínek čistě harmonických proudů a fázových posunů, kde

$$i_A = I_m \cdot \sin(\omega \cdot t)$$

$$i_B = I_m \cdot \sin\left(\omega \cdot t - \frac{2}{3}\pi\right)$$

$$i_C = I_m \cdot \sin\left(\omega \cdot t - \frac{4}{3}\pi\right)$$

jsou složky Parkova vektoru:

$$i_D = 1.5 \cdot I_m \sin(\omega \cdot t) \tag{2.8}$$

$$i_Q = -1.5 \cdot I_m \cos(\omega \cdot t) \tag{2.9}$$

Tento rotující vektor vytváří v ideálním případě kružnici se středem umístěným v počátku souřadnicového systému s poloměrem $1.5I_m$.

U rovnic (2.6),(2.7),(2.8) a (2.9) lze volit i jiné koeficienty. V literatuře [6] se vyskytuje varianta, kde jsou všechny koeficienty na pravých stranách rovnic vynásobeny $\sqrt{2}/\sqrt{3}$.

Parkovy vektory lze vyhodnocovat ve dvou oblastech, časové a frekvenční. V časové oblasti je výhodné sledovat Parkovy vektory zobrazené v rovině xy. Na tvaru a umístění orbity můžeme pozorovat projevy některých poruch.

Modulace ve spektrech proudu se projeví nadměrnou tloušťkou orbity nebo ztrátou jejího oblého tvaru. Pro lepší specifikaci poruch projevující se modulací používáme vyhodnocení spekter magnitudy:

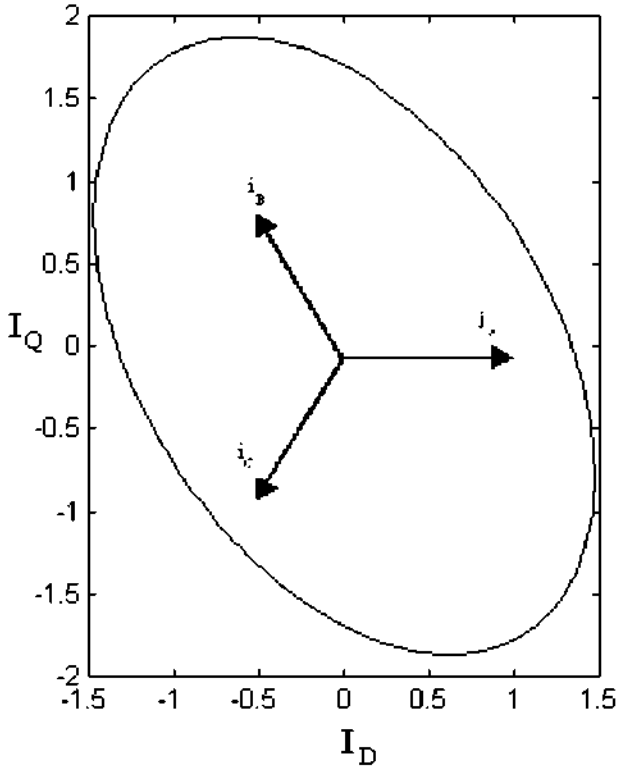
$$absP = \sqrt{i_D^2 + i_Q^2} \tag{2.10}$$

Spektrum magnitudy obsahuje v ideálním případě pouze složku na nulové frekvenci o velikosti $1.5I_m$.

2.2.2 Užití Parkových vektorů pro diagnostiku vad

Sledováním Parkových vektorů v časové rovině lze snadno detekovat chyby vznikající zkratovanými závity na statorovém vinutí. Na Obr 3.2 lze spatřit orbitu zploštělou do tvaru elipsy. Porovnání směru orbity s orientací proudů je zřejmé na které

fázi došlo ke zkratování závitů.Dále lze usuzovat, že při větším počtu zkratovaných závitů bude deformace orbity markantnější.



Obr. 2.2: Orbita při zkratovaných závitech na fázi i_B.

Sledováním frekvenční oblasti magnitudy (2.10) lze rozšířit diagnostiku motoru i o pozorování dalších závad.

Zkratované závity se ve spektru magnitudy projevují na frekvenci 100 Hz, kde je rozdíl mezi funkčním motorem a motorem se závadou snadno patrný.

Přerušené rotorové tyče se projeví ve spektru magnitudy na skluzové frekvenci $2*s*f_{line}$.

3. Umělé neuronové sítě

3.1 Úvod do neuronových sítí

Umělou neuronovou sítí myslíme strukturu určenou pro distribuované paralelní zpracování dat, která se skládá z určitého počtu vzájemně propojených výkonových prvků. Každý prvek může přijímat libovolně velký, ale konečný počet různých vstupních dat. Dále může na další výkonové prvky předávat konečný počet informací o stavu svého jediného výstupu. Každý výkonový prvek transformuje vstupní data na výstupní podle určité přenosové funkce. [7][8]

Uměla neuronová síť nalézá uplatnění v široké škále odvětví.[9] V lékařství můžeme jejím simulováním a sledováním chování prohlubovat znalosti o principech fungování nervových soustav živých organismů. V informačních oborech hraje velkou roli pro rozpoznávání vzorů z obrazu. V ostatních oborech se používá všude tam, kde je nutno zpracovávat neurčité, nepřesné či neúplné informace. Hlavní doménou použití je analýza komplikovaných neperiodických nebo kvaziperiodických signálů.

Názvem neuron je původně označen základní stavební prvek nervových soustav. V těchto soustavách je neuron živá buňka specializovaná na příjem, zpracování, uchování a přenos informace.

V umělých neuronových sítích se jako stavební prvek používá formální neuron (zkráceně neuron), což je zjednodušený matematický model neuronu živého organismu.

3.2 Model neuronu

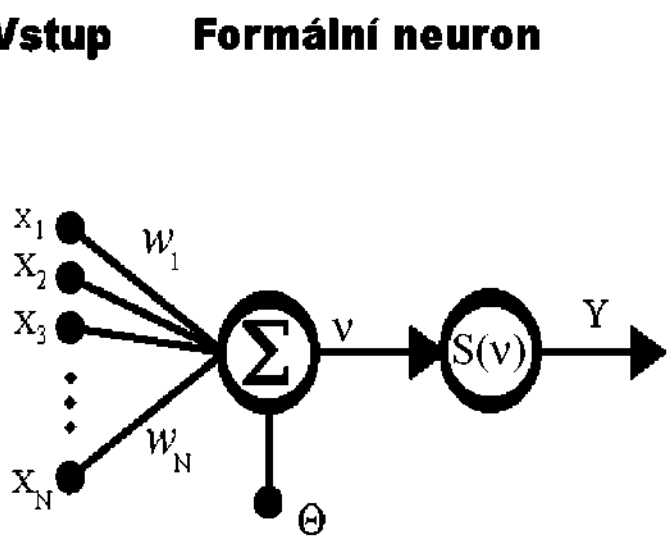
V literatuře je popsána řada modelů neuronů. Od těch velmi jednoduchých, řazených do první generace používající nespojité přenosové funkce (viz dále), až po velmi složité, popisující každý detail chování neuronu živého organismu. V praxi při vytváření neuronových sítí nejsme schopni úplně simulovat chování živé nervové soustavy i při nejkomplexnějších modelech neuronů, protože v nervové soustavě se vyskytují ještě další stavební prvky podílející se na přenosu informací nazývané gliové buňky a jejich funkce v biologické neuronové síti není ještě známa.

V umělých neuronových sítích se jako výkonový prvek používá formální neuron.

V praxi se nejčastěji používá model neuronu který popsali pánové McCulloch a Pitts. (*Obr 3.1.*)

kde:

x_i jsou vstupy neuronu
(aktivita vstupu)
w_i jsou synaptické váhy
Θ je práh
S(x) přenosová funkce neuronu
(též se nazývá aktivační)
Y je výstup neuronu



Obr. 3.1: Formální model neuronu.

Neuron zpracovává vstupní informace na výstupní podle vztahu:

$$v = \sum_{i=1}^{N} w_i x_i + \Theta \tag{3.1a}$$

$$Y = S(v) \tag{3.1b}$$

Velikost vah w_i vyjadřuje uložení zkušeností do neuronu. Čím vyšší hodnota, tím je daný vstup důležitější.

V biologickém neuronu práh Θ označuje prahovou hodnotu aktivace neuronu, a je-li $\Sigma w_i x_i$ menší než práh, pak je neuron v pasivním stavu. Z jiného úhlu pohledu se dá na práh chápat jako další váhový vstup pevně připojený na jednotkovou veličinu a přidává neuronu další stupeň volnosti.

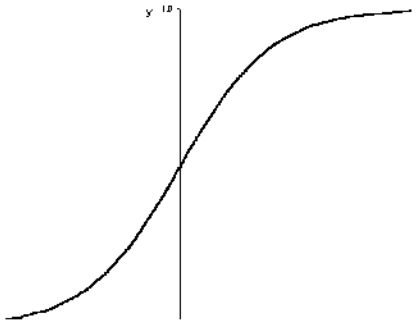
S překročením prahové hodnoty, výstup neuronu nelineárně roste a jeho průběh ovlivňuje přenosová funkce S(v).

Podle povahy vstupních (a výstupních) dat můžeme rozlišovat neurony binární a spojité.. Podle typu neuronu se použije vhodná přenosová funkce.

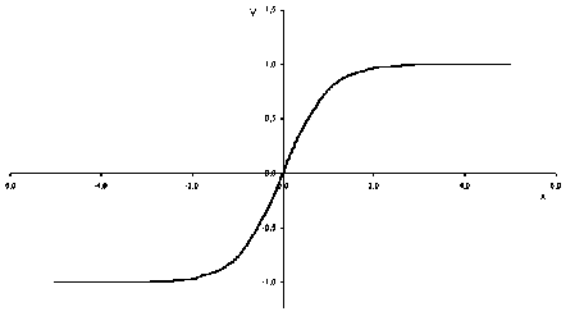
3.3 Přenosová funkce neuronu

Jednou z nejčastěji používaných přenosových funkcí je sigmoida (*Obr 3.2 a*). Její název je odvozen od tvaru průběhu, který připomínaná zploštělé písmeno S.
Na dalších obrázcích je ukázána řada dalších používaných aktivačních funkcí. Na *Obr 3.2 b)* je další z často používaných funkcí – hyperbolická tangenta. Na *Obr 3.2 c) a e)* je aktivační funkce binárních neuronů.

a) sigmoida



b) hyperbolická tangenta

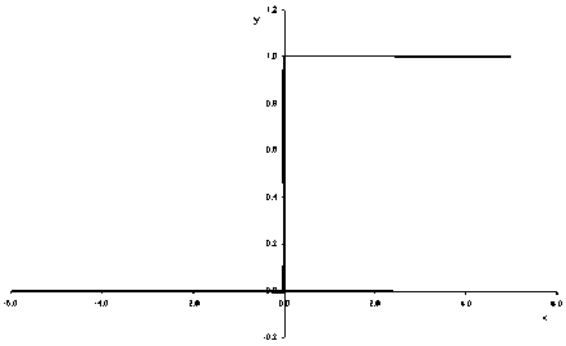


$$y(x) = \frac{1}{1 + e^{-k \cdot x}}$$

$$y(x) = \tanh(x)$$

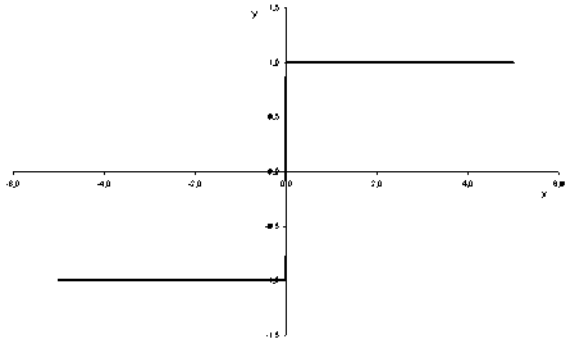
$$y(x) = \frac{2}{1 + e^{-k \cdot x}} - 1$$

c) skoková funkce



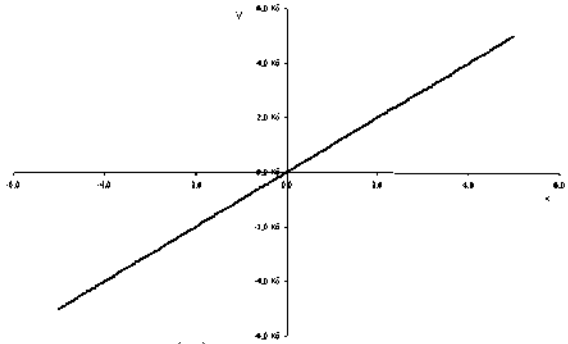
$$y(x) = 0 \quad x < 0$$
$$y(x) = 1 \quad x \geq 0$$

d) skoková funkce



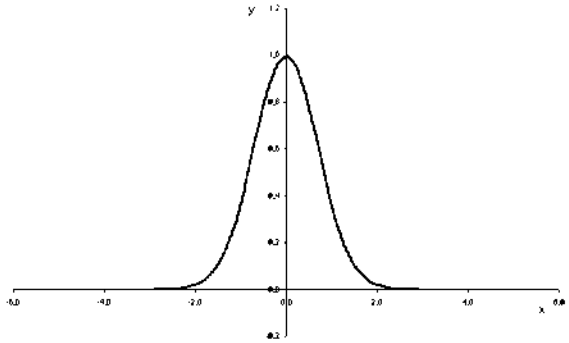
$$y(x) = -1 \quad x < 0$$
$$y(x) = 1 \quad x \geq 0$$

e) lineární funkce



$$y(x) = k \cdot x$$

f) funkce neuronů radiální báze



$$y(x) = e^{-k \cdot x^2}$$

Obr 3.2 Některé typy přenosových funkcí

3.4 Typy neuronových sítí

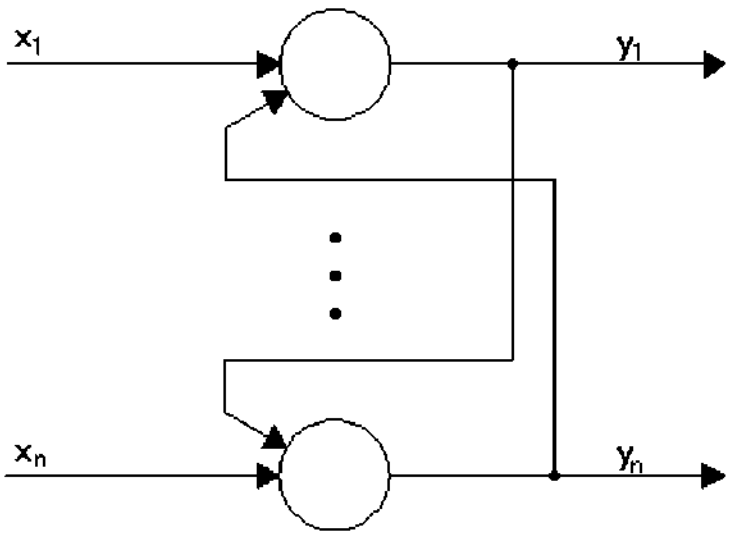
Neuronové sítě můžeme dělit podle několika různých hledisek. :

- podle typu zpracovávaných signálů: na digitální a analogové
- podle způsobu učení: na sítě s učitelem a bez něj
- podle topologie: na sítě rekurentní a vrstevnaté

Síť vytvoříme vzájemným propojením výkonových prvků..

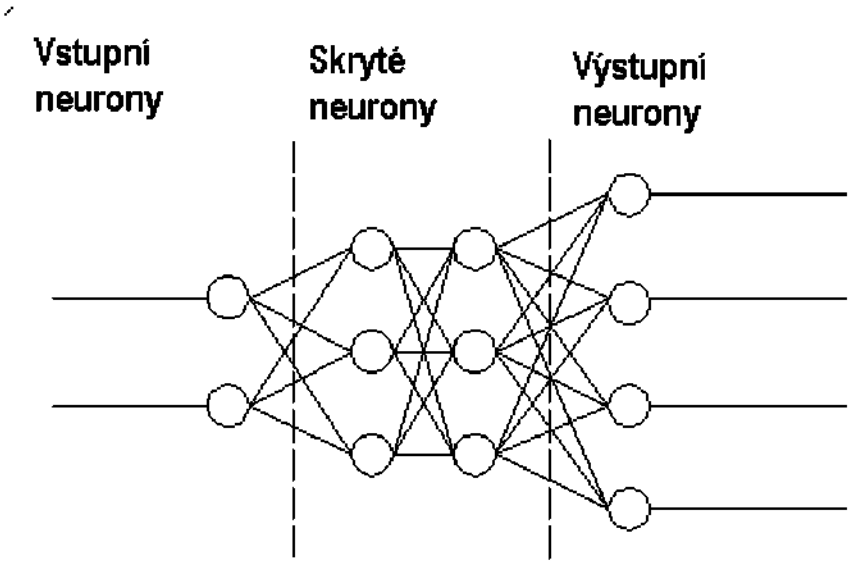
V rekurentních sítích je zavedena zpětná vazba, pomocí které vzniká značná časová závislost.Po příchodu signálu je určena výstupní hodnota a ta je zároveň znovu přivedena na vstup, který modifikuje. Při učení rekurentních sítí se vychází z určitého počátečního stavu sítě a jejích postupné změny probíhají dokud není dosaženo stabilního stavu.

Jedna z nejčastěji používaných rekurentních neuronových sítí je Hopfieldova síť [10] (*Obr. 3.3*) Každý neuron má propojení na všechny ostatní neurony v síti kromě sebe samotného.



Obr 3.3: Hopfieldova síť.

Ve vrstevnatých sítích se šíří signál od vstupu k výstupu. Neurony v jedné vrstvě nejsou mezi sebou pospojovány, ale neurony mezi vrstvami jsou pospojovány každý s každým. Také u nich není zavedena zpětná vazba. Neurony obsažené ve vrstevnaté neuronové síti lze rozdělit na tři druhy (*Obr 3.4*).



Obr 3.4: Obecná struktura vrstevnaté sítě.

Vstupní neurony — distribuují vnější signály do sítě

Skryté neurony — jejich vstupy a výstupy jsou propojeny pouze na neurony v síti

Výstupní neurony — výstupy těchto neuronů tvoří i výstup z celé sítě

Umělá neuronová síť jako celek provádí transformaci vstupních hodnot na vektor hodnot výstupních. Což lze vyjádřit vztahem:

$$\mathbf{Y} = T(\mathbf{X}) \tag{3.2}$$

V roce 1957 byl matematikem Andrejem Kolmogorovem uveřejněn teorém , z něhož vyplývalo, že jakoukoliv reálnou spojitou funkci na n-rozměrné krychli lze definovat jako:

$$f(x_1,\ldots,x_n)=\sum_{i=1}^{2n+1} g_i\left(\sum_{j=1}^{n} h_{ij}(x_j)\right) \tag{3.3}$$

g_i, h_{ij} jsou spojité funkce reálné proměnné

Prací dalších matematiků bylo dokázáno, že lze vystačit pouze s jednou funkcí g_i, kterou lze vyjádřit tvarem $\lambda_j h_i$, kde λ_j jsou konstanty. Z těchto tvrzení plyne, že pro aproximaci funkce *f* neuronovou sítí stačí, aby síť na vstupu měla n neuronů, jednu skrytou vrstvu s 2n+1 neurony a na výstupu m neuronů.. Bohužel tato informace nic nevypovídá o výběru nejvhodnější sítě pro daný problém.

3.5 Učení neuronové sítě

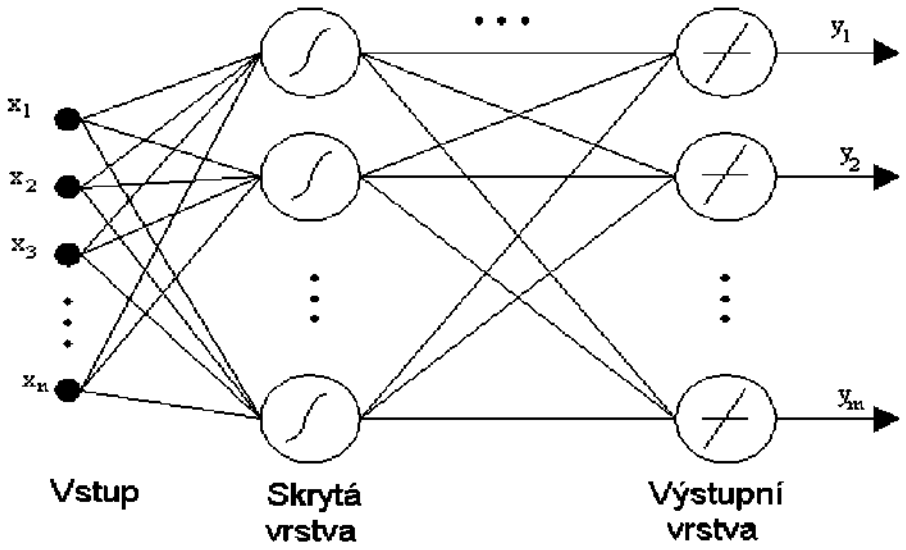
Učení neuronových sítí lze rozdělit na učení s učitelem a učení bez učitele.

Při učení s učitelem je nám při daném vstupním signálu znám odpovídající výstupní signál. Při procesu učení upravujeme parametry sítě tak, aby výstup sítě odpovídal požadovaným hodnotám v rámci určité přesnosti. Při učení se můžou upravovat nejen hodnoty vah a prahů, ale i struktura sítě nebo se mohou měnit parametry přenosových funkcí. Převážně se používá učení, které mění pouze velikosti vah a prahů.

U sítě s učením bez učitele se nepoužívá vzor k naučení sítě. Algoritmus učení je navržen tak, aby při předkládání vstupních dat hledala síť mezi nimi určitou závislost a „vytvářela si svůj vlastní názor na ně “.

3.6 Neuronová síť typu backpropagation

Sítě typu backpropagation (dále BPG) jsou vrstevnaté sítě s dopředným šířením signálu a se zpětným šířeným při učení. Jejich struktura (*Obr. 3.5*) odpovídá struktuře vrstevnaté sítě.



Obr. 3.5 Síť typu backpropagation

BPG sítě obvykle obsahují jednu nebo více skrytých vrstev neuronů s přenosovou funkcí sigmoidy nebo hyperbolické tangenty (*Obr 3.2* a) a b)). Dále následuje výstupní vrstva mající neurony s lineární funkcí (*Obr 3.2* e)).

Uvedené funkce jsou monotónně rostoucí a jsou diferencovatelné, což je v algoritmu BPG důležité, protože potřebujeme derivace přenosových funkcí.

Pravidlo učení BPG vychází z pravidla učení Widrow-Hoff (dále WH) používané v lineárních sítích.[11].

Při učení vycházíme ze součtu chyby čtverců sse (sum-squared error) a chybového vektoru **E** (3.4). Vektor se spočítá z rozdílu požadované hodnoty a výstupu ze sítě. Dále při učení využíváme krok učení lr, který funguje jako násobek změn vah a prahů. Jeho velikost ovlivňuje rychlost učení. Je-li ale příliš velký, vede učení k nestabilitě.

Odvodíme si pravidlo WH pro jednovrstvou lineární síť a poté provedeme modifikaci pro algoritmus BPG.

$$\mathbf{E} = \mathbf{Y}_p - \mathbf{Y}_s \tag{3.4}$$

E … chybový vektor

$\mathbf{Y_p}$… požadovaný výstup

$\mathbf{Y_s}$ … výstup ze sítě

Součet chyby čtverců je:

$$sse = \sum_{k=1}^{N} e(k)^2 = \sum_{k=1}^{N} \left(y_p(k) - y_s(k)\right)^2 \tag{3.5}$$

Pravidlo učení WH počítá malé změny vah a prahů ve směru snížení chyby neuronů. Tento směr je nalezen pomocí derivace součtu chyby čtverců k parametrům. Derivace chyby sse k vahám W(i,j) (vstup j neuronu i) pro jeden vstupní vektor **X** a cílový vektor $\mathbf{Y_p}$ je:

$$\frac{\partial sse}{\partial W(i,j)} = \frac{\partial}{\partial W(i,j)}\left(y_p(i) - y_s(i)\right)^2 = \frac{\partial}{\partial W(i,j)}\left(y_p(i) - \sum_{i=1}^{R} W(i,j)x(j) - \Theta(i)\right)^2 =$$
$$= -2 \cdot S(v)' \cdot x(j) = -2 \cdot e(i) \cdot x(j)$$

Konstantu –2 nahradíme krokem učení lr. Váhy a prahy se upraví následujícím způsobem:

$$\mathbf{W} = \mathbf{W} + \Delta\mathbf{W} \qquad \Delta\mathbf{W} = lr \cdot \mathbf{E} \cdot \mathbf{X}^T \tag{3.6a}$$

$$\Theta = \Theta + \Delta\Theta \qquad \Delta\Theta = lr \cdot \mathbf{E} \tag{3.6b}$$

Pro učení BPG je ještě nutno znát hodnoty v v každém neuronu vyskytující se v síti. Tato znalost je nutná pro určení derivací přenosových funkcí u každého neuronu. Z těchto derivací si vytvoříme matici **D** a nastavení vah a prahů se provede tímto způsobem:

$$\mathbf{W} = \mathbf{W} + \Delta\mathbf{W} \qquad \Delta\mathbf{W} = lr \cdot \mathbf{D} \cdot \mathbf{X}^T \tag{3.7a}$$

$$\Theta = \Theta + \Delta\Theta \qquad \Delta\Theta = lr \cdot \mathbf{D} \tag{3.7b}$$

Učení tímto způsobem je poměrně zdlouhavé a často se nemusí najít optimální parametry sítě. Učení může skončit v lokálních minimech a tak se hledají cesty, jak se tomuto problému vyhnout a jak urychlit učící proces. Používají se různé metody. Jedna z nich vylepšuje algoritmus BPG přidáním momentu nebo adaptivního kroku učení.

Učení s momentem umožňuje překonat lokální minima a dostat se do globálního. Adaptivní krok učení zrychluje trénink sítě. Zvětšuje se krok učení, dokud není změna chyby příliš velká, poté se snižuje, až je učení stabilní.

Dalšími metodami jsou aplikace standardních optimalizačních metod. Jedněmi z možností použití je Levenberg-Marquartova algoritmu nebo quasi-Newtonova algoritmu. [11][12]

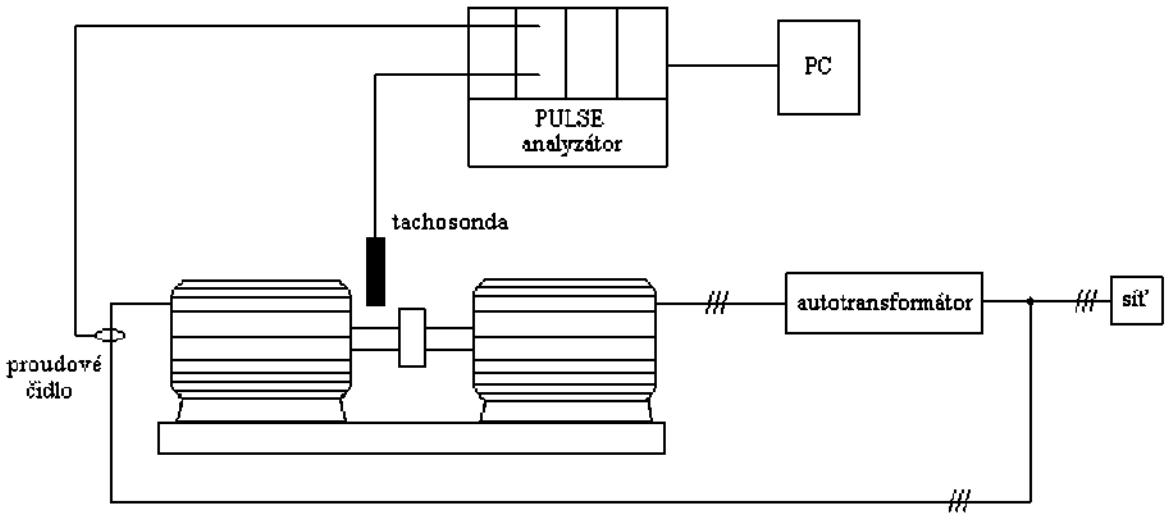
4. Experimentální část

4.1 Popis měřícího pracoviště

Hlavní částí měřícího pracoviště je multianalyzátor PULSE s kanály pro napětí, akustickou intenzitu, vibrace, hluky, generátor a tachosondu. Rozlišení rychlé Fourierovy transformace až 1 mHz a dynamické rozlišení až 80 dB. Proudy tekoucí statorovým vinutím jsou měřeny přípravkem obsahující 4 snímače proudu LTS 25-NP pracující na Hallově principu.

Pro experimenty jsou použity motory Siemens typu 1 LA7083-2, jmenovitý výkon 1.1 kW, 2 póly, jmenovité otáčky 2845 min^{-1}, jmenovitý proud 2.4 A. Firma Siemens dodala motory s definovanými vadami statoru a rotoru.

Motory jsou k přípravku přimontovány proti sobě (*Obr 4.1*), kde jeden motor slouží pro účely měření a druhý je použit jako zátěž.



Obr. 4.1: Měřící pracoviště.

Motory jsou k sobě spojeny pružnou spojkou přenášející moment mezi motory. Pružná spojka byla zvolena z důvodu snadnější manipulace. Indukční motory jsou k desce připevněny přes odpruženou destičku, aby nedocházelo k přenosu vibrací mezi motory.

Měření otáček se provádí pomocí tachosondy. Dále se měří tři fázové proudy. Multianalyzátor PULSE je schopen v reálném čase vyhodnocovat časové průběhy

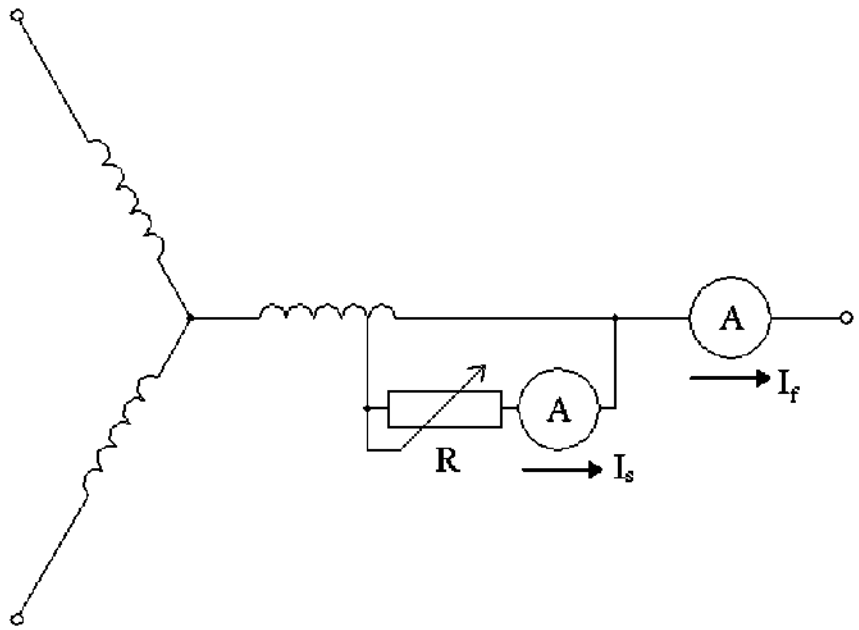
fázových proudů, efektivní hodnoty proudů, spektra proudů a otáčky hřídele motoru. Celkem lze vyhodnocovat až 16 veličin, které lze v různé kombinaci zobrazovat na monitoru počítače. S analyzátorem je dodána sada programů a lze vytvořit vlastní programy pro účely měření. Všechna měření, jejich případná transformace a následné zobrazení se provádí také v reálném čase.

Pro účely měření byl vyroben motor, který má na začátku fáze U vyvedeny odbočky po 2, 5 a 10 závitech. (viz příloha 1) Odbočky byly použity pro simulaci zkratu ve fázi. Pro další měření byl použit motor s poškozeným rotorem. Byly u něj přerušeny 2 rotorové tyče.

4.2 Měření na asynchronních motorech

4.2.1 Měření zkratovaných závitů na statorovém vinutí

První měření byla provedena u motoru s vyvedenými odbočkami na 2, 5 a 10 závitech a motoru bez poruchy. Simulace vývinu poruchy byla provedena zařazením proměnného odporu do zkratovaného vinutí.(*Obr 4.2*)



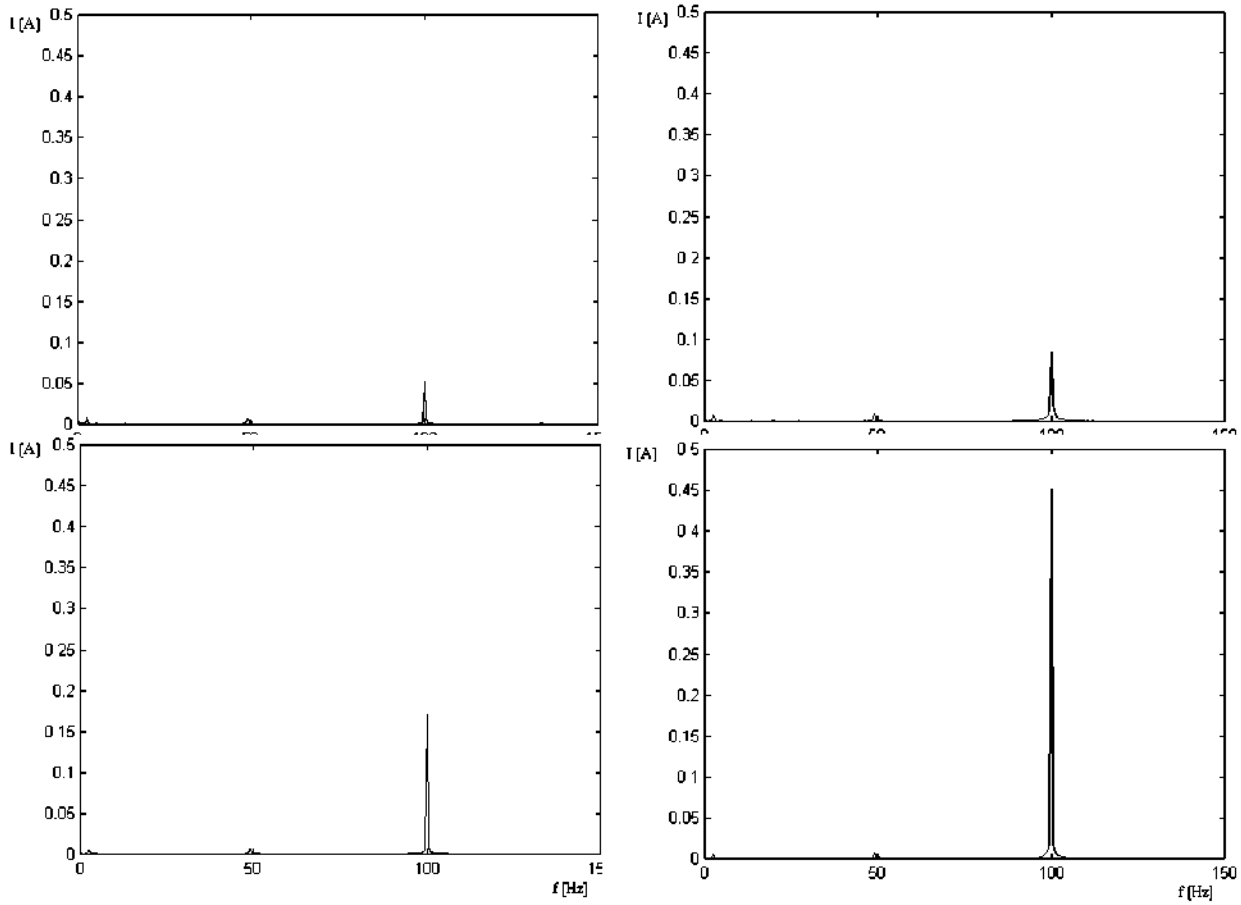
Obr. 4.2: Měření na statoru při simulaci vývinu poruchy.

Pro znatelný nárůst poruchy je hlavní měření při zkratovaných 10 závitech pro odpory 21.4Ω, 10.8Ω, 6.2Ω, 4.0Ω a 0Ω. Měření bylo prováděno pro motor bez zatížení, pro zátěž 25%, 50%, 75% a 85%. Průběhy proudů se dále snímaly pomocí analyzátoru PULSE a ukládaly se do počítače pro pozdější zpracování.

Tabulka 1:

zátěž	naprázdno	25%	50%	75%	85%	
If [A]	1,39	1,68	1,87	2,03	2,25	R=21,4Ω
Is [A]	0,65	0,65	0,64	0,64	0,64	
If [A]	1,36	1,63	1,81	1,97	2,19	R=10,8Ω
Is [A]	1,3	1,29	1,3	1,3	1,3	
If [A]	1,37	1,58	1,8	2,11	2,29	R=6,2Ω
Is [A]	2,5	2,48	2,48	2,48	2,48	
If [A]	1,31	1,55	1,7	1,96	2,24	R=4,0Ω
Is [A]	5,04	5,08	5,07	5,06	5,08	
If [A]	1,24	1,64	1,98	2,23	2,47	R=0Ω
Is [A]	18,8	19,3	19	19,1	18,6	

Z tabulky vyplývá nárůst zkratového proudu v souvislosti vývinem poruchy. Na *Obr. 4.3* je zobrazen nárůst poruchy ve spektru centrované magnitudy Parkových vektorů na frekvenci 100 Hz. (postupně pro 0z, 2z, 5z a 10z a 50% zátěž)



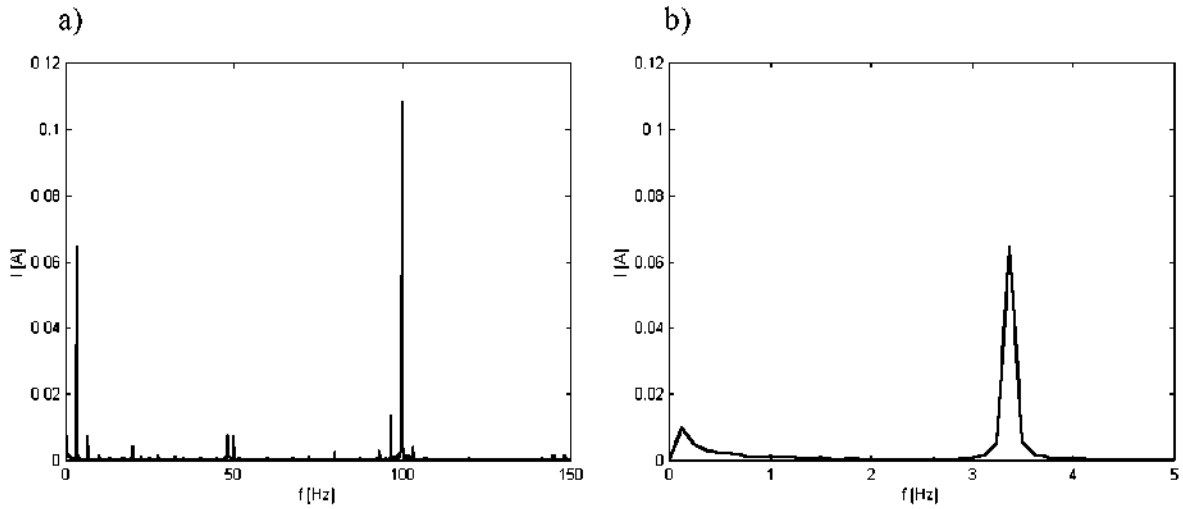
Obr. 4.3: Vývin poruchy a jeho projev v centrovaném spektru magnitudy

Z obrázku vyplývá, že pro menší počet zkratovaných závitů, je jejich diagnostika nejednoznačná. Zřetelný nárůst proudu je až při zkratovaných pěti závitech.

4.2.2 Měření přerušených rotorových tyčí

Pro měření přerušených rotorových tyčí se používá klasické FFT spektrum proudu a sledují se postranní pásma kolem síťové frekvence. (*Obr 1.1*)

Závady tohoto charakteru se pomocí Parkových vektorů zjišťují vznikem amplitudy na skluzové pólové frekvenci. (*Obr 4.4 a) a b)*)



Obr 4.4: Spektrum magnitudy při přerušených rotorových tyčí

4.3 Aplikace umělé neuronové sítě

4.3.1 Příprava dat a vytvoření umělé neuronové sítě

V praxi se pro diagnostiku používá změna okénka (čas měření T – kap 2.1.3) pro lepší specifikaci závady. Pro rotorové vady je výhodnější používat delší okénko (T > 4 s) než pro zjišťování statorových vad (0.2 s < T < 2 s). Pro zjišťování obou těchto vad je nutno zvolit kompromis, optimální se jeví T = 2 s.

Jedním z nejzákladnějších problémů kolem neuronových sítí je kromě zvolení vhodné neuronové sítě i kvalitní výběr učící množiny. Pro testování a učení neuronové sítě byly naměřeny tyto data:

- motor bez závad
- motor s přerušenými rotorovými tyčemi (2 přerušené rotorové tyče)
- motor se zkratovanými závity na statoru (2z, 5z a 10z)
- motor při simulaci vývinu poruchy (podle *Obr. 4.2* pro 2z, 5z a 10z)

Pro takto zvolená data odstup koeficientů frekvenčních složek FFT je $\Delta f = 0.5$ Hz. Podle projevů diagnostikovaných poruch lze usoudit, že frekvenční rozsah stačí do 150 Hz. Z toho vyplývá počet vstupních dat do neuronové sítě je max. 300 vzorků.

Vytvoří se neuronová síť se třemi vrstvami. V první vrstvě je počet neuronů roven počtu vzorků (např. 300 neuronů), ve druhé vrstvě je počet neuronů poloviční (150 neuronů) a ve třetí (výstupní) vrstvě je počet neuronů 2 až 3. Počet výstupních neuronů ovlivňuje specifikace rozlišovaných vad. Dále se síť redukuje na optimální počet neuronů se zpětnou kontrolou funkce sítě. Lze redukovat i vstupní data.

4.3.2 Umělá neuronová síť vytvořená v programu MATLAB®

Ve světě výpočetní techniky mají svou nezastupitelnou roli programy pro numerické výpočty. Jedním z nejrozšířenějších je univerzální programovací balík MATLAB od firmy The MathWorks, Inc.

Název MATLAB vznikl z MATrix LABoratory a byl původně určen jako knihovna matematických funkcí. Obsahuje co nejširší balík univerzálních nástrojů pro numerické výpočty. Pro některé problémy inženýrské praxe jsou k MATLABu přidány specializované algoritmy formou Toolboxů. Jako např. Neural Network Toolbox, nebo Frequency Domain Identification Toolbox.

K dispozici je MATLAB verze 5.3 (R11) s toolboxem Neural Network verze 3.0.1.

MATLAB umožňuje provádět technické výpočty zápisem, který je podoben programovacímu jazyku, ale i graficky pomocí nadstavby Simulink. Programování je objektového typu a při vytvoření neuronové sítě BPG jsou nejdůležitější tyto funkce a parametry [12]:

1) Vytvoření neuronové sítě

– Je-li trénovací vektor o velikosti RxS (řádky x sloupce), BPG síť vytvoříme inicializační funkcí **newff**.

Prototyp funkce je: **net = newff(PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF)**

PR ... matice Rx2 obsahující minimální a maximální hodnoty vstupu

[S1 S2...SNl] ... počet neuronů v jednotlivých vrstvách

{TF1 TF2...TFNl} ... přenosové funkce v jednotlivých vrstvách

BTF ... trénink neuronové sítě (standardně **trainlm**)

BLF ... nastavení vah a prahů neuronové sítě při učení (standardně **learngdm**)

PF ... funkce podle které se určuje celková chyba neuronové sítě (standardně **mse**)

2) Nastavení parametrů učení

— maximální počet kroků učení se nastaví parametrem **trainParam.epochs**

— maximální chyba učení se nastaví parametrem **trainParam.goal**

Při využití umělých neuronových sítí je důležité uvažovat čas strávený při výpočtech. Odezva výstupu neuronové sítě na vstup je poměrně rychlá. (řádově 10^{-2} – 10^{-1} s). Znatelně časově náročnější je trénink neuronové sítě. Využití pro praxi předpokládáme již jednou naučenou síť, která při předkládání naměřených dat diagnostikuje naučené vady.

Pro učení a následnou optimalizaci je klasický algoritmus BPG nevyhovující z důvodu jeho přílišné pomalosti. Nastavuje se parametrem **traingd**. Pokud jsou odpovídající možnosti počítače (hlavně z hlediska volné paměti) je doporučeno použití Levenberg-Marquardtova (dále LM) algoritmu. Nastavení je **trainlm**. Jeho paměťové nároky jdou snížit vhodným upravením parametru **trainParam.mem_reduc**. Nedostačuje-li volná paměť ani redukovanému LM učení, lze použít pružný BPG trénink (resilient BPG – parametrem **trainrp**). Struktura objektu neural network je uvedena v příloze 2.

Pro představu, v případě uvedeném v předchozí kapitole a počtu učených vzorů rovno pěti , klasický BPG algoritmus nebyl schopen natrénovat neuronovou síť po osmi minutách a byl ukončen krokem s číslem 6000. Pružný BPG trénink naučil síť rozpoznávat signál během pár sekund a stačilo mu na to méně než 70 kroků. A LM algoritmus je schopen síť naučit během 50-ti kroků.

U sítě, která se byla schopna naučit požadovaný vzor snižujeme počet neuronů do té doby, dokud je síť schopna se naučit předložené vzory. Snažíme v sítí snížit špatnou generalizaci způsobenou nadměrným počtem neuronů. Pro pochopení vlivu, příliš velké generalizace či naopak příliš malé, doporučuji prozkoumat příklady v MATLABu – demorb3 a demorb4.

3) Další parametry a vlastnosti umělé neuronové sítě

— **layers**

Tato vlastnost uchovává popis jednotlivých vrstev v síti

layers{i}.transferFcn. – změna přenosové vrstvy v i-té vrstvě

layers{i}.initFcn – změna inicializační funkce v i-té vrstvě

layers{i}.size – počet neuronů v i-té vrstvě

— **numInputs**

Popisuje a nastavuje vlastnosti každého z i vstupů.

numInputs{i}.range – nastavuje rozsah maximálních a minimálních hodnot vstupů

numInputs{i}.size – definuje počet prvků vstupující do neuronové sítě

— **outputs**

Parametr uchovávající vlastnosti každého z i výstupů.

outputs {i}.size – definuje počet prvků vystupující ze sítě

— **biases**

Parametr uchovávající vlastnosti každého prahu

biases {i}.initFcn – změna inicializační funkce i-tého prahu

biases {i}.learnFcn – definuje nastavení prahu při učení

biases {i}.size – definuje velikost vektoru prahů i-té vrstvy (pouze čtení)

— **inputWeights**

Parametr uchovávající vlastnosti vstupních vah.

inputWeights{i,j}.initFcn – změna inicializační funkce váhy i-té vrstvy j-tého vstupu

inputWeights{i,j}.learnFcn – definuje nastavení vah při učení

inputWeights{i,j}.size – definuje velikost matice i-té vrstvy j-tého vstupu (pouze čtení)

— **layerWeights**

layerWeights{i,j}.initFcn – změna inicializační funkce váhy i-té vrstvy z j-té vrstvy

layerWeights{i,j}.learnFcn – definuje nastavení vah při učení

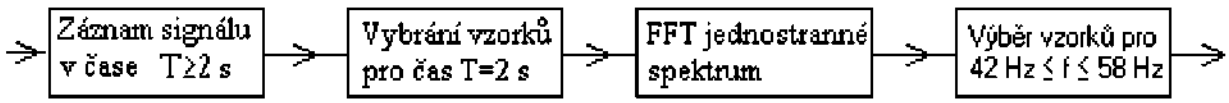
layerWeights{i,j}.size – definuje velikost matice i-té vrstvy z j-té vrstvy (pouze čtení)

4) Upravování parametrů umělé neuronové sítě

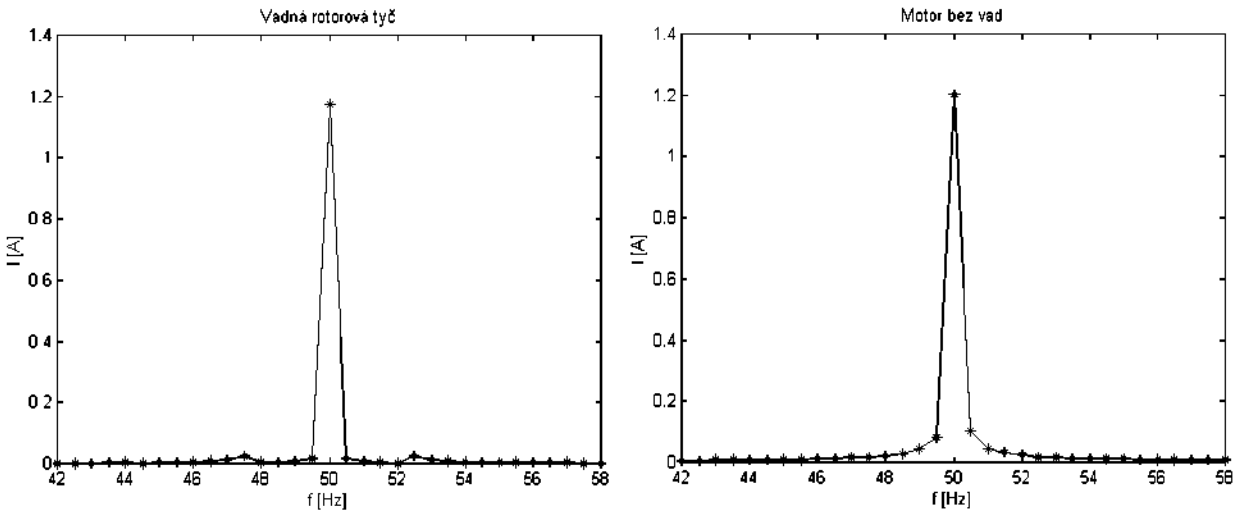
U již jednou naučené sítě lze upravovat parametry, kde se ještě mohou zlepšit přenosové funkce sítě. V rámci postprocesingu se může nastavit vhodnější výstupní přenosová funkce, která ušetří práci s reprezentováním dat, které dodala neuronová síť.

4.3.3 Rozpoznávání závady rotoru pomocí FFT

První síť byla vytvořena pro rozpoznávání závady rotoru (příloha 3). Trénovací množina sítě byla vytvořena z naměřených hodnot motoru bez vad a motoru s vadným rotorem. Schéma pro přípravu vzoru, který byl předán neuronové síti, je zobrazeno na *Obr. 4.5*. Na *Obr. 4.6* je zobrazen upravený signál.

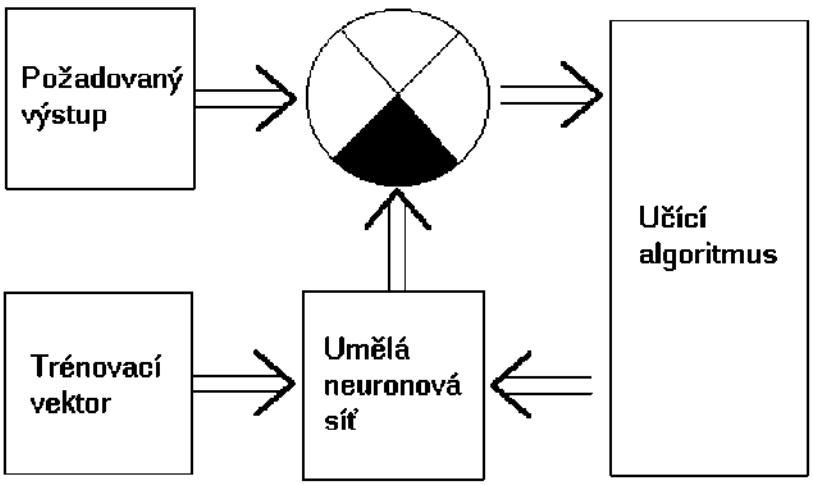


Obr. 4.5: Příprava vzoru pro síť zjišťující závadu rotoru.



Obr. 4.6: Upravený vzorek vstupující do neuronové sítě 1

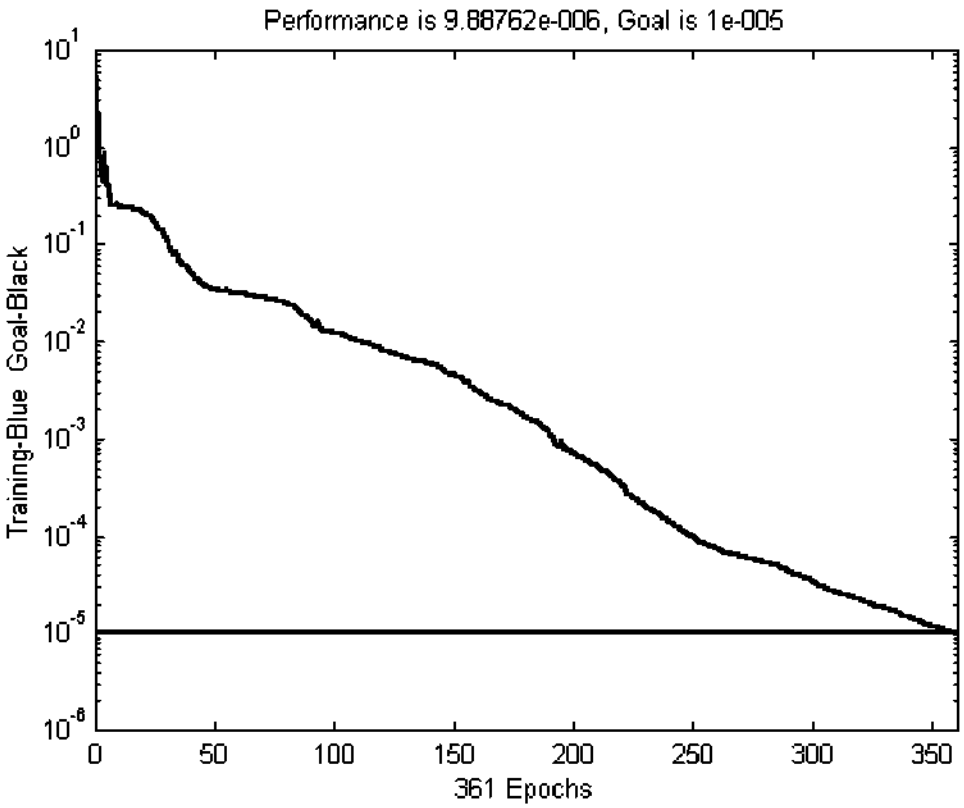
Tímto předzpracováním dodáme do sítě 33 hodnot frekvenčního spektra v rozmezí daném na *Obr. 4.5*. Pro natrénovaní neuronové sítě bylo použito 10 záznamů z měření, konkrétně fáze Ia pro všechna měřená zatížení u motoru s rotorovou vadou a motoru bez vad. Konfigurace sítě je 20 – 10 – 2. Učení probíhá podle následujícího schématu (*Obr. 4.7*).



Obr. 4.7: Schéma tréninku neuronové sítě

Trénovací vektor je v tomto případě matice o rozměrech 33 x 10. Výstupem ze sítě jsou dva neurony, z nichž první označuje je-li motor bez vad a druhý signalizuje poruchu na rotoru. Síť signalizuje příznak hodnotou blížící se jedné. Porovnáním obou výstupů můžeme usoudit výskyt poruchy.

Při trénování neuronové sítě se sleduje rychlost učení, respektive počet kroků učení a schopnost aproximace přenosové funkce neuronové sítě. Pro představu je zde uveden průběh chyby v závislosti na počtu kroků učícího algoritmu (*Obr. 4.8*).



Obr. 4.8: Průběh chyby při učení pomocí pružného BPG algoritmu

Pro testování byla použita všechna měření a výsledky jsou prezentovány v tabulce. Na začátku tabulky je uvedena specifikace chyby měřeného motoru. V lichých řádcích jsou hodnoty z výstupu signalizující dobrý stav motoru (A) a v sudých závadu na rotoru (B). Příznak se projeví, je-li hodnota výstupu blízká jedné (± 0,2). Z tabulek vyplývá, že pro jasné určení závady je nutno síti dodat data z více měření a výsledky poté zohlednit. U motoru se zkratovanými závity na statoru neuronová síť hlásí dobrý motor. To vyplývá z projevu poruchy, kde neuronová síť je natrénována na dynamickou poruchu a tento projev není schopna identifikovat. Schopnost sítě rozpoznávat je více než 90%. To je samozřejmě způsobeno zaměřením pouze na jednu vadu a nízkým počtem dat vstupujícím do sítě.

Tabulka 2:

dobrý motor:

A	0,8460	0,9313	0,2507	0,9916	0,9194	0,9078	0,9743	0,8763	1,0270	0,9743
B	0,1482	0,0649	0,7240	-0,0025	0,0606	0,0806	0,0399	0,1451	-0,0260	0,0221

přerušený rotor:

A	-0,1805	0,0767	-0,5061	0,1350	-0,3328	-0,0519	-0,1534	-0,1277	-0,0441	-0,0229
B	1,1911	0,9278	1,4649	0,8475	1,3280	1,0467	1,1680	1,1620	0,9940	0,9643

zkrat 2z:

A	0,9370	0,9040	1,0000	0,9980	0,9730	1,0500	0,9670	0,9010	0,9670	1,0700
B	0,0487	0,0928	-0,0122	0,0078	0,0328	-0,0457	0,0373	0,1090	0,0380	-0,0647
A	1,0700	1,0600	0,9790	0,9350	0,9800					
B	-0,0677	-0,0532	0,0107	0,0644	0,0058					

zkrat 5z:

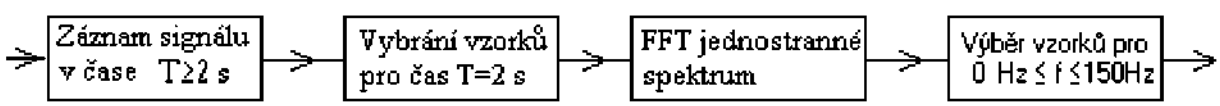
A	1,0700	0,8810	0,6280	0,9500	0,9350	0,8800	1,0200	0,9860	1,1000	0,9830
B	-0,0554	0,1170	0,3760	0,0715	0,0827	0,1370	-0,0111	0,0300	-0,0943	0,0269
A	1,0500	0,9810	0,8170	0,9020	0,8800					
B	-0,0488	0,0204	0,1790	0,0937	0,1200					

zkrat 10z:

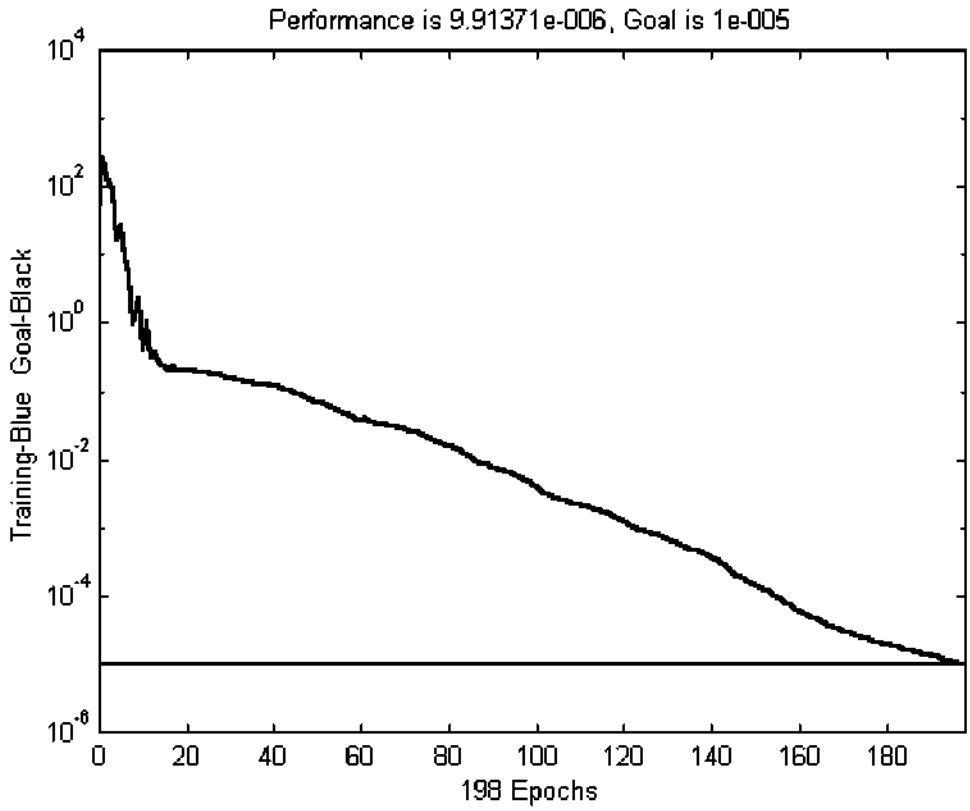
A	1,0600	1,0500	0,6910	0,9260	0,9820	0,9520	0,7800	0,9390	0,9290	0,9050
B	-0,0624	-0,0531	0,2910	0,0803	0,0327	0,0447	0,2220	0,0778	0,0858	0,0481
A	1,0200	0,9910	1,6200	1,2900	1,2500					
B	-0,0379	0,0061	-0,7960	-0,4020	-0,3510					

4.3.4 Rozpoznávání závad rotoru a statoru pomocí FFT

Druhá síť je modifikace z předchozí kapitole. Do trénovací množiny byla přidána měření provedená na motorech se zkratovanými závity na statoru a dále byla vstupní data do neuronové sítě rozšířena na 300 vzorků (*Obr. 4.9*).



Obr. 4.9: Příprava vzoru pro síť zjišťující závadu rotoru a statoru pomocí FFT



Obr. 4.10: Průběh chyby při tréninku sítě pomocí algoritmu pružný BPG.

V trénink byly použity hodnoty z fáze Ia a pro testování se použili záznamy fází Ib a Ic. Konfigurace umělé neuronové sítě je 140 – 60 – 3.

Síť má tři výstupy:

A – dobrý motor

B – přerušený rotor

C – zkratovaný stator

Tabulka 3:

dobrý motor:

A	1,0688	0,8315	1,3855	0,4522	1,4168	1,9220	0,6972	0,5484	1,0334	1,0154
B	-0,1044	0,0594	0,4977	0,1627	0,0382	0,1455	0,0996	0,0597	-0,0044	0,0040
C	-0,1164	0,2285	-0,8078	0,3498	-0,3694	-1,1969	0,2385	0,3963	0,0282	-0,0202

přerušený rotor:

A	1,3621	1,5907	0,4824	-0,1297	0,2146	-0,6399	0,3873	0,3654	1,5335	1,6708
B	0,7948	0,9376	1,2192	0,8501	1,0330	0,9292	1,0648	0,8590	0,8022	0,7275
C	-1,3968	-1,5434	-0,9437	0,0651	-0,3265	0,7546	-0,3873	-0,2362	-1,2698	-1,4175

zkrat 2z:

A	0,4187	0,3643	0,7856	0,0712	1,0882	0,7433	-0,0638	0,1142	1,3886	1,2630
B	-0,0247	-0,1168	0,1488	0,0902	-0,0207	-0,0396	0,1092	0,0331	0,0090	-0,0081
C	0,5957	0,6901	0,1272	0,9781	-0,0146	0,3828	1,0199	0,8809	-0,3544	-0,2294

zkrat 5z:

A	0,1549	0,1825	0,8425	0,6519	0,6507	0,8282	0,5968	0,4413	0,7735	0,8888
B	0,2304	0,2923	0,1383	0,3021	-0,1689	-0,1627	-0,0761	-0,0140	0,0502	0,0128
C	0,7303	0,5523	0,2223	0,3686	0,5746	0,5896	0,4213	0,5126	0,0556	-0,0038

zkrat 10z:

A	0,1995	0,3432	0,6039	0,4686	0,9543	0,2257	0,6311	0,6657	0,8480	1,1059
B	0,1924	-0,1309	-0,1478	0,0735	-0,0509	0,0648	0,0052	0,0417	-0,1149	-0,0222
C	0,8811	0,4670	0,6268	0,8135	0,1883	0,6823	0,4226	0,3277	0,2175	-0,1414

Na výsledcích je patrné, že úspěšnost rozpoznání závady není příliš vysoká. Pohybuje se kolem 30ti %. Důvodů této malé úspěšnosti může být několik:

Redundantní informace – vzorky vstupující do sítě obsahují spoustu údajů nevztahující se přímo k diagnostikovaným vadám.

Nesprávný výběr množiny – na množině, síti předkládané, se nedostatečně projevují příznaky důležité pro specifikaci závad.

Nevhodná konfigurace sítě – nesprávně zvolená velikost sítě ovlivňuje schopnost aproximace předkládaných dat.

Konfigurace sítě (tj. počet neuronů v jednotlivých vrstvách) se nastavuje v průběhu testování sítě.

FFT transfomace poskytuje o signálu velké množství informací, ale například zkrat na statoru se pomocí ní, zjistí velmi obtížně. Proto je vhodnější použít jinou metodu.

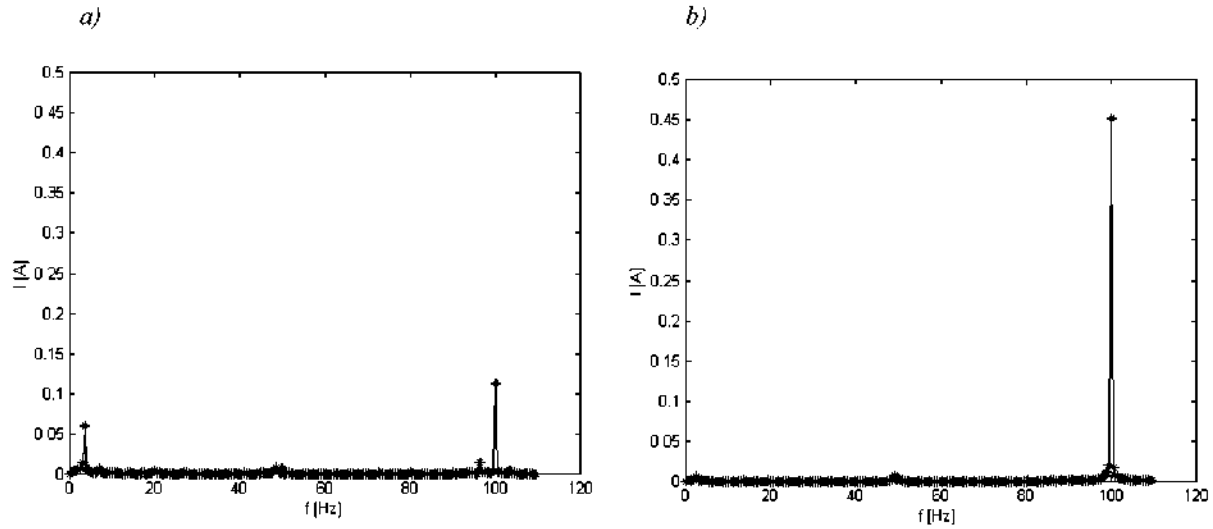
4.4 Využití Parkových vektorů pomocí neuronových sítí

4.4.1 Využití spektra magnitudy

Ve spektru magnitudy Parkovy transformace se výrazněji projevují definované vady (kap. 4.2.1 a kap. 4.2.2). To dává síti větší šanci rozpoznat závadu na motoru. Na *Obr. 4.11* je naznačeno schéma úpravy vzoru před předáním neuronové síti.

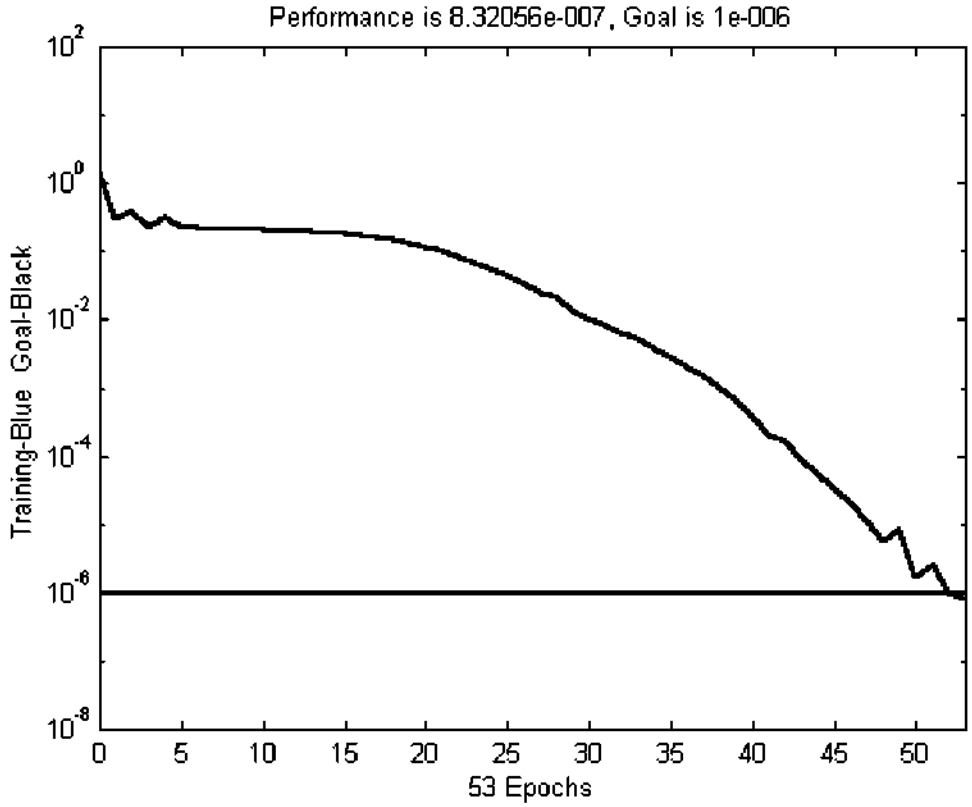
Obr. 4.11: Příprava vzoru pomocí spektra magnitudy Parkovy transformace

Po této úpravě vstupují do sítě data u nichž víme, že se vyskytuje vyšší amplituda na skluzové pólové frekvenci při rotorových vadách a znatelný nárůst amplitudy na frekvenci 100 Hz při statorové závadě. Příklad, jak mohou vypadat upravená data podle výše uvedeného schématu je na *Obr.4.12.*



Obr. 4.12: Upravený vzorek vstupující do neuronové sítě 3 (a-vadný rotor, b- vadný stator)

Na *Obr. 4.13* je zobrazen průběh tréninku umělé neuronové sítě.



Obr. 4.13: Průběh chyby sítě při tréninku pomocí algoritmu pružný BPG.

Byla vytvořena umělá neuronová síť s konfigurací 50 – 10 – 3. Počet vstupních dat je redukován na 240 vzorků. Programy, který vytvoří a natrénuje výše popsanou umělou neuronovou síť je uveden v příloze 4. Výsledky testování jsou uvedeny v následujících tabulkách v hodnotách při měření naprázdno, 25%, 50%, 75% a 85% zatížením.

Výstupy ze sítě jsou:

A – dobrý motor

B – vadný stator

C – vadný rotor

Tabulka 4:

dobrý motor:

A	1,0013	0,8074	0,4125	0,6955	0,9998
B	0,0009	0,1581	0,3288	0,4496	-0,0005
C	-0,0017	0,2342	0,5158	-0,0449	0,0002

zkrat 5z:

A	-0,1169	0,1520	-0,0399	-0,0013	0,5736
B	0,6759	0,8531	0,8764	0,9989	0,6063
C	0,4231	0,0626	0,3460	0,0000	-0,2798

přerušený rotor:

A	0,0008	-0,1252	0,0973	0,0009	0,1716
B	0,0005	-0,0518	-0,2969	0,0004	0,0336
C	1,0003	1,1576	1,2321	1,0004	0,9426

zkrat 10z:

A	-0,0002	-0,2911	-0,2853	-0,4585	-0,4237
B	1,0004	0,9858	0,9210	0,9280	0,6261
C	0,0004	0,3788	0,5178	0,5341	0,7848

zkrat 2z:

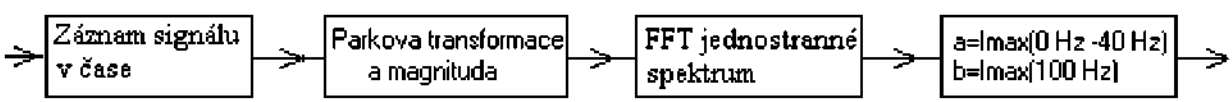
A	0,8118	-0,4869	0,0217	0,2157	0,7810
B	0,2632	0,6880	0,5280	0,6838	0,3413
C	-0,0602	0,7769	0,6083	0,1855	-0,2308

Z naměřených dat vyplývá, že umělá neuronová síť poměrně dobře diagnostikuje rotorové vady (výstup C je blízký jedné). Zkratované závity je síť schopna rozpoznat až od pěti závitů pro zatížení 25% – 75 %.

V datech dodávané síti je stále spousta redundantních informací. Pro zvýšení účinnosti rozpoznávání sítě lze data zredukovat na nejvýznačnější body.

4.4.2 Využití význačných bodů ze spektra magnitudy

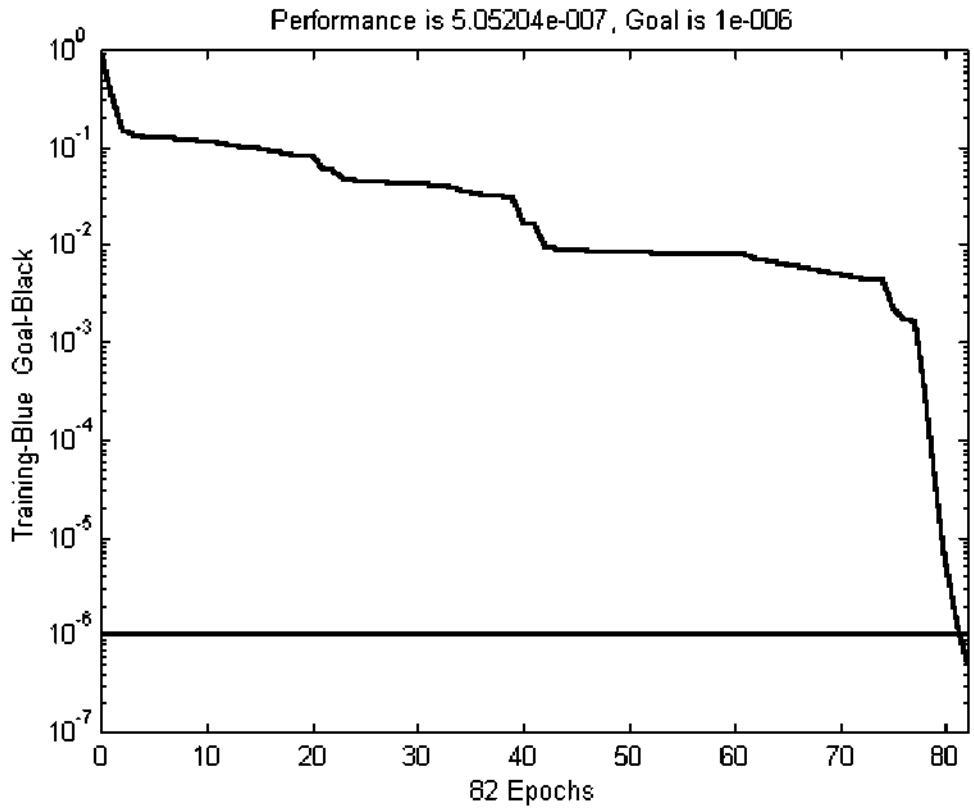
Ve zpracování dat podle *Obr. 4.12* se vychází ze spektra magnitudy. Síť má dva vstupy. Jako první vstup je maximální hodnota proudu v rozmezí 0 Hz < f [40 Hz. V tomto rozmezí se projevuje příznak přerušených rotorových tyčí. Druhým vstupem do sítě je velikost amplitudy na frekvenci 100 Hz.



Obr. 4.12: Příprava vzorku s využitím význačných projevů ve spektru magnitudy

Výhodou tohoto předzpracování je, že vzorkování proudu tekoucí motorem může být různé, dle potřeby. Dále je výhoda menší neuronové sítě mající jednu skrytou vrstvu a jednu výstupní. Konkrétně data zpracovává 5 neuronů (konfigurace 2 – 3).

Na *Obr. 4.13* je zobrazen průběh chyby při učení této neuronové sítě.



Obr. 4.13: Průběh chyby při tréninku umělé neuronové sítě.

U takto zredukovaných vstupních dat je obecně jednodušší vybrat vhodnou tréninkovou množinu. Síť se naučí nejpodstatnější příznaky a schopnost rozpoznání vady je vyšší. Pro trénink se použily hodnoty příznaků z šesti měření. Konkrétně měření dobrého motoru při zatížení 0% a 85%, s vadným rotorem při 0% a 75% a s vadným statorem při 10 závitech naprázdno a 5 závity při 75%.

Výsledky z testování jsou uvedeny v následující tabulce.

Výstupy ze sítě:

A – dobrý motor

B – vadný stator

C – vadný rotor

Tabulka 5:

dobrý motor:

A	0,9981	0,9997	0,9991	0,9985	0,9987
B	0,0000	0,0001	0,0006	0,0012	0,0013
C	0,0019	0,0002	0,0003	0,0003	0,0000

zkrat 2z:

A	0,9974	-0,0001	0,8375	0,1958	0,2874
B	0,0005	1,0000	0,1621	0,8041	0,7123
C	0,0020	0,0001	0,0003	0,0001	0,0002

přerušený rotor:

A	0,0014	-0,0045	-0,0001	-0,0003	-0,0024
B	0,0001	0,0044	0,0000	0,0002	0,0023
C	0,9985	1,0001	1,0001	1,0001	1,0001

zkrat 5z:

A	-0,5102	-0,0002	-0,0001	0,0003	-0,0001
B	0,9995	1,0000	1,0000	0,9996	1,0000
C	0,5106	0,0002	0,0001	0,0001	0,0001

zkrat 10z:

A	-0,0002	-0,0001	-0,0001	-0,0001	-0,0001
B	1,0000	1,0000	1,0000	1,0000	1,0000
C	0,0002	0,0001	0,0001	0,0001	0,0001

Umělá neuronová síť dobře rozpoznává motor bez vad a motor s přerušenými rotorovými tyčemi. U motoru se zkratovanými závity je schopna rozpoznat zkratovaných 10 závitů a 5 závitů. Pro menší počet zkratovaných závitů síť diagnostiku dobrý motor. To je zřejmě způsobeno malou změnou amplitudy proudu na frekvenci 100 Hz spektra magnitudy.

V následující tabulce jsou zobrazeny hodnoty z výstupu sítě při předložení měření uvedeném v kap. 4.1.2 (měření simulace vývinu poruchy).

Výstupy ze sítě:

A – dobrý motor

B – vadný stator

C – vadný rotor

Tabulka 6:

dobrý motor:

A	1,0000	1,0000	1,0000	0,9995	0,9984
B	0,0000	0,0000	0,0000	0,0000	0,0000
C	0,0000	0,0000	0,0000	0,0006	0,0016

zkrat 10z s odporem ve vinutí:

A	1,0000	0,9989	1,0000	0,9996	0,9994
B	0,0000	0,0000	0,0000	0,0000	0,0000
C	0,0000	0,0011	0,0000	0,0004	0,0006

Umělá neuronová síť není schopna rozpoznat zkrat na statoru. Změna amplitudy na frekvenci 100 Hz spektra magnitudy je malá a neuronová síť naměřená data vyhodnotí jako motor bez vad.

Pro praktické využití je nutno ještě síť naučit rozpoznávat další závady. To předpokládá ještě měření rychlosti otáček motoru. Vstupní vzorky pak budou přímo hodnoty amplitud důležitých frekvencí dodané díky znalosti otáček.

5. Závěr

Cílem diplomové práce bylo zjistit možnost použití umělých neuronových sítí pro diagnostiku vad asynchronních motorů. Byly naznačeny možné cesty, jak naučit umělou neuronovou síť diagnostikovat závady. V posledním případě se povedlo vytvořit účinnou síť, schopnou poměrně kvalitně rozpoznávat vady. Velkou výhodou umělých neuronových sítí, je jejich schopnost nalézat vztahy mezi vstupními signály a reagovat na ně.

Další prací na tomto tématu bude vytvoření univerzálnější sítě, schopné rozpoznávat vícero typů závad. Předpokládá větší výběr důležitých příznaků a složitější zpracování vzorků. Uplatní se především v měřících přístrojích pracujících v reálném čase a snímající otáčky motoru.

Zajímavým směrem v oblasti diagnostiky motoru se může ukázat použití samoučících neuronových sítí. Umělá neuronová síť, na základě dat naměřených za provozu, může být schopna odhadnout blížící se výskyt poruchy a oznámit předem kontrolu motoru před výskytem závady.

Literatura:

[1] Biloš, J.: Diagnostika asynchronních elektrických motorů., Výzkumná zpráva. Moravsko-slezské teplárny 1996.

[2] Jaksch, I.,Fuchs, P.: Metody diagnostiky rotorových vad asynchronních motorů. TD 2002 – Diagon 2002, Zlín 2002.

[3] Jaksch, I.,Fuchs, P.: Diagnostika statorových vad indukčních motorů pomocí Parkových vektorů, TD 2002 – Diagon 2002, Zlín 2002.

[4] Přikryl, P.: MVŠT. Numerické metod matematické analýzy, Praha, SNTL 1985

[5] Blaška, J.: Fourierova transformace a diskrétní Fourierova transformace - DFT. http://klokan.sh.cvut.cz/~blaska/kapitola1/headkap1.htm

[6] Cardoso, A.J.M., Mendes, A.M.S., Cruz, S.M.A.: The Park's Vector Approach: New Developments in On-Line Diagnosis of Electrical Machines, Power Electronics and Adjustable Speed Drives., IEEE SDEMPED, 1999

[7] Kadziolka, M., Pšenčík, O.: Neuronové sítě – Perceptron, http://www.mujweb.cz/www/_mak_/Perceptron/Perceptron.htm

[8] Sarle, W.S.: FAQ Neural networks, ftp://ftp.sas.com/pub/neural/FAQ.html

[9] Novák, M. a kol.: Umělé neuronové sítě. Teorie a aplikace. 1.vydání, C.H.BECK, Praha 1998

[10] Prokeš, M.: Hopfieldova síť. Základní model, http://service.felk.cvut.cz:8877/courses/36NAN/36NAN017

[11] Krejsa, J., Březina, T.: Neuronové modelování - návod ke cvičením

[12] Demuth, H.,Beale, M.: Neural network toolbox. For Use with MATLAB. Version 3, The MathWorks Inc. 1998

[13] Pavelka, J. a kol.: Elektrické pohony, ČVUT 1999

[14] Slavík, J.: EEG Workshop, Diplomová práce, VUT Brno 2002

Přílohy:

Příloha 1:

Přílohy:

Příloha 1

Příloha 2

Struktura objektu neural network v programu MATLAB

```
Neural Network object:
    architecture:
         numInputs: 1
         numLayers: 3
       biasConnect: [1; 1; 1]
      inputConnect: [1; 0; 0]
      layerConnect: [0 0 0; 1 0 0; 0 1 0]
     outputConnect: [0 0 1]
     targetConnect: [0 0 1]
        numOutputs: 1  (read-only)
        numTargets: 1  (read-only)
    numInputDelays: 0  (read-only)
    numLayerDelays: 0  (read-only)
    subobject structures:
            inputs: {1x1 cell} of inputs
            layers: {3x1 cell} of layers
           outputs: {1x3 cell} containing 1 output
           targets: {1x3 cell} containing 1 target
            biases: {3x1 cell} containing 3 biases
      inputWeights: {3x1 cell} containing 1 input weight
      layerWeights: {3x3 cell} containing 2 layer weights
    functions:
          adaptFcn: 'adaptwb'
           initFcn: 'initlay'
        performFcn: 'mse'
          trainFcn: 'trainrp'
    parameters:
        adaptParam: .passes
         initParam: (none)
      performParam: (none)
        trainParam: .epochs, .show, .goal, .time,
                    .min_grad, .max_fail, .delt_inc, .delt_dec,
                    .delta0, .deltamax
    weight and bias values:
                IW: {3x1 cell} containing 1 input weight matrix
                LW: {3x3 cell} containing 2 layer weight matrices
                 b: {3x1 cell} containing 3 bias vectors
    other:
          userdata: (user stuff)
```

Příloha 3

Program tréninku sítě pro diagnostiku rotorových vad

Progrotor.m

```
% Program pro vytvoreni trenovaci mnoziny a natrenovani neuronove site
% pro diagnostiku vad rotoru
echo on
clear all;
% pocatecni konstanty
load ladobry.txt;
load larotor.txt;

T=2;                    % delka mereni
N=1024;                 % kolik car pouzit pro FFT
n1=85;                  % parametry pro vyjmuti vzorku
n2=117;
dn=n2-n1+1;
konfigurace=[28 14 2];  % topologie site

% FFT transformace
Nd=size(ladobry);       % pocet vzorku
specd=zeros(N/2,Nd(2));
for i=1:1:Nd(2),
temp=fft(ladobry(:,i));     % udelat FFT
temp2=abs(temp(1:N/2)); % ziskej absolutni hodnotu jednostraneho spektra
specd(:,i)=temp2;
clear temp;
clear temp2;
end
Nr=size(larotor);
specr=zeros(N/2,Nr(2));
for i=1:1:Nr(2),
temp=fft(larotor(1:N,i)');
temp2=abs(temp(1:N/2))';
specr(:,i)=temp2;
clear temp;
clear temp2;
end
```

```
specd=specd/N;              % normalizace velikosti slozek spektra
specr=specr/N;

% vytvoreni trenovaci vzoru
trvzorek=zeros(dn,Nr(2)+Nd(2));
trvzorek(:,1:Nd(2))=specd(n1:n2,:);
trvzorek(:,Nd(2)+1:end)=specr(n1:n2,:);

% vytvoreni pozadovaneho vystupu
pozvystup=zeros(2,Nr(2)+Nd(2));
pozvystup(1,1:Nd(2))=1;
pozvystup(2,Nd(2)+1:end)=1;

% matice predpokladanych hodnot vstupu pro inicializaci neuronove site
Inicializace=zeros(dn,2);
Inicializace(:,2)=10;

% vytvoreni neuronove site a nastaveni parametru pro uceni
net = newff(Inicializace,konfigurace,{'tansig' 'tansig' 'purelin'},'trainrp');
net.trainParam.show = 30;
net.trainParam.epochs = 6000;
net.trainParam.goal = (0.00001);

%   Zacina trenink..prosim vyckejte...
net = train(net,trvzorek,pozvystup);

% Ulozeni neuronove site
save  net;
```

Program pro testování rotorových vad

test.m

```
% program pro testovani site urcene k diagnostice rotorovych vad
clear;

load net.mat;
load rtest.txt;
load dtest.txt;
load test2z.txt;
load test5z.txt;
```

```
load test10z.txt;

echo on

T=2;                    % delka mereni
N=1024;                 % kolik car pouzit pro FFT
n1=85;                  % parametry pro vyjmuti vzorku
n2=117;
dn=n2-n1+1;

% FFT transformace
Nr=size(rtest);         % pocet vzorku
specr=zeros(N/2,Nr(2));
for i=1:1:Nr(2),
temp=fft(rtest(:,i));   % udelat FFT
temp2=abs(temp(1:N/2)); % ziskej absolutni hodnotu jednostraneho spektra
specr(:,i)=temp2;
clear temp;
clear temp2;
end
specr=specr/N;

% vytvoreni testovaciho vzoru
vzorek=zeros(dn,Nr(2));
vzorek=specr(n1:n2,:);

a=sim(net,vzorek)

%vadny rotor
pause

Nr=size(dtest);         % pocet vzorku
specr=zeros(N/2,Nr(2));
for i=1:1:Nr(2),
temp=fft(dtest(:,i));   % udelat FFT
temp2=abs(temp(1:N/2)); % ziskej absolutni hodnotu jednostraneho spektra
specr(:,i)=temp2;
clear temp;
clear temp2;
end
```

```
specr=specr/N;

% vytvoreni testovaciho vzoru
vzorek=zeros(dn,Nr(2));
vzorek=specr(n1:n2,:);

a=sim(net,vzorek)
%dobry motor
pause

Nr=size(test2z);          % pocet vzorku
specr=zeros(N/2,Nr(2));
for i=1:1:Nr(2),
temp=fft(test2z(:,i));    % udelat FFT
temp2=abs(temp(1:N/2));   % ziskej absolutni hodnotu jednostraneho spektra
specr(:,i)=temp2;
clear temp;
clear temp2;
end
specr=specr/N;

% vytvoreni trenovaci vzoru
vzorek=zeros(dn,Nr(2));
vzorek=specr(n1:n2,:);

a=sim(net,vzorek)
%2z zkratovany
pause

Nr=size(test5z);          % pocet vzorku
specr=zeros(N/2,Nr(2));
for i=1:1:Nr(2),
temp=fft(test5z(:,i));    % udelat FFT
temp2=abs(temp(1:N/2));   % ziskej absolutni hodnotu jednostraneho spektra
specr(:,i)=temp2;
clear temp;
clear temp2;
end
specr=specr/N;
```

```
% vytvoreni trenovaci vzoru
vzorek=zeros(dn,Nr(2));
vzorek=specr(n1:n2,:);

a=sim(net,vzorek)
%5z zkratovano

Nr=size(test10z);          % pocet vzorku
specr=zeros(N/2,Nr(2));
for i=1:1:Nr(2),
temp=fft(test10z(:,i));    % udelat FFT
temp2=abs(temp(1:N/2));    % ziskej absolutni hodnotu jednostraneho spektra
specr(:,i)=temp2;
clear temp;
clear temp2;
end
specr=specr/N;

% vytvoreni trenovaci vzoru
vzorek=zeros(dn,Nr(2));
vzorek=specr(n1:n2,:);

a=sim(net,vzorek)
%10z zkratovano
```

Příloha 4

Program tréninku sítě pro diagnostiku motoru pomocí Parkovy transformace

Progpark1.m

```
% Program pro vytvoreni neuronove site pouzivajici Parkovy vektory
echo off;
clear all;
load Idnapr.txt;
load Id85.txt;
load I5z75.txt;
load I10znapr.txt;
load Irnapr.txt;
load Ir75.txt;

Np=6;          %pocet trenovacich vzoru
N=1024;        %pocet prvku ve vzoru
n1=300;        %pocet pouzitych prvku pro trenink z jednoho mereni
T=2;           %delka mereni
konfigurace=[50 10 3];% topologie site

echo on;
%parkova transformace a FFT centrovane magnitudy
%
echo off;
tempmer=[Idnapr,Id85,I5z75,I10znapr,Irnapr,Ir75];
temppark=zeros(N,Np);
j=1;
  for i=1:3:(3*Np),
    id=tempmer(:,i)-0.5*tempmer(:,i+1)-0.5*tempmer(:,i+2);   %id=ia-0.5*ib-0.5*ic;
       iq=sqrt(3)/2*(tempmer(:,i+1)-tempmer(:,i+2));         %iq=sqrt(3)/2*(ib-ic);
    mag=sqrt(id.^2+iq.^2);                                   %vypocet magnitudy
    mag=mag-mean(mag);                                       %centrovani magnitudy
    temppark(:,j)=mag;
    j=j+1;
  end
clear tempmer;
clear id;
clear iq;
```

```
clear mag;

spec=zeros(N/2,Np);
    for i=1:1:Np,
     temp=fft(temppark(:,i));          %fft transformace centrovane magnitudy
     spec(:,i)=abs(temp(1:N/2));       %absolutni hodnota jednostranneho spektra
  end
clear temp;
spec=spec/N;
echo on;
%Inicializace neuronove site
echo off;
trvzorek=spec(1:n1,:);
clear spec;
trspravne=zeros(3,Np);
trspravne(2,3:4)=1;    % vadny stator
trspravne(1,1:2)=1;    % dobry motor
trspravne(3,5:6)=1;    % vadny rotor
Inicializace=zeros(n1,2);
Inicializace(:,2)=10;
net = newff(Inicializace,konfigurace,{'logsig' 'logsig' 'purelin'},'trainrp', 'learngdm','mse');
net.trainParam.show = 30;
net.trainParam.epochs = 6000;
net.trainParam.goal = (0.000001);

echo on;
%   Zacina trenink..prosim vyckejte...
net = train(net,trvzorek,trspravne);
% Ulozeni neuronove site
save  net;
echo off;
```

Příloha 5

Program tréninku sítě využívající význačných příznaků ze spektra magnitudy Parkovy transfomace

Progpark1.m

```
% Program pro vytvoreni neuronove site pouzivajici priznaky ze spektra magnitudy
echo off;
clear all;
load Idnapr.txt;
load Id85.txt;
load I5z75.txt;
load I10znapr.txt;
load Irnapr.txt;
load Ir75.txt;

Np=6;          %pocet trenovacich vzoru
N=1024;        %pocet prvku ve vzoru
n1=80;         %pocet pouzitych prvku dulezitych pro rotorove vady
n2=101;        %dulezity prvek pro statorovou vadu
T=2;           %delka mereni
konfigurace=[2 3];  % topologie site

echo on;
%parkova transformace a FFT centrovane magnitudy
%
echo off;
tempmer=[Idnapr,Id85,I5z75,I10znapr,Irnapr,Ir75];
temppark=zeros(N,Np);
j=1;
  for i=1:3:(3*Np),
    id=tempmer(:,i)-0.5*tempmer(:,i+1)-0.5*tempmer(:,i+2);   %id=ia-0.5*ib-0.5*ic;
       iq=sqrt(3)/2*(tempmer(:,i+1)-tempmer(:,i+2));         %iq=sqrt(3)/2*(ib-ic);
    mag=sqrt(id.^2+iq.^2);                                   %vypocet magnitudy
    mag=mag-mean(mag);                                       %centrovani magnitudy
    temppark(:,j)=mag;
    j=j+1;
  end
clear tempmer;
clear id;
```

```
clear iq;
clear mag;

spec=zeros(N/2,Np);
    for i=1:1:Np,
     temp=fft(temppark(:,i));          %fft transformace centrovane magnitudy
     spec(:,i)=abs(temp(1:N/2));       %absolutni hodnota jednostranneho spektra
  end
clear temp;
spec=spec/N;
spec_fr=(0:1/T:N/(2*T)-1);

echo on;
%Inicializace neuronove site
echo off;
trvzorek=zeros(2,Np);
for i=1:1:Np
  a=max(spec(2:n1,i));
  trvzorek(1,i)=a;
end
trvzorek(2,:)=spec(201,:);
clear spec;
trspravne=zeros(3,Np);
trspravne(2,3:4)=1;    % vadny stator
trspravne(1,1:2)=1;    % dobry motor
trspravne(3,5:6)=1;    % vadny rotor
trspravne
trvzorek
Inicializace=zeros(2,2);
Inicializace(:,2)=2.5;
net = newff(Inicializace,konfigurace,{'logsig' 'purelin'},'trainlm', 'learngdm','mse');
net.trainParam.show = 30;
net.trainParam.epochs = 6000;
net.trainParam.goal = (0.000001);
echo on;
%   Zacina trenink..prosim vyckejte...
net = train(net,trvzorek,trspravne);
% Ulozeni neuronove site
save  net;
echo off;
```

Program pro testování rotorových vad

test1.m

```
% testovani neuronove site s predzpracovanim pomoci priznaku ze spektra magnitudy
echo off;
clear all;
load net.mat;
load ldnapr.txt;
load ld25.txt;
load ld50.txt;
load ld75.txt;
load ld85.txt;
load lrnapr.txt;
load lr25.txt;
load lr50.txt;
load lr75.txt;
load lr85.txt;
load l10znapr.txt;
load l10z25.txt;
load l10z50.txt;
load l10z75.txt;
load l10z85.txt;
load l2znapr.txt;
load l2z25.txt;
load l2z50.txt;
load l2z75.txt;
load l2z85.txt;
load l5znapr.txt;
load l5z25.txt;
load l5z50.txt;
load l5z75.txt;
load l5z85.txt;

Np=25;      %pocet trenovacich vzoru
N=1024;     %pocet prvku ve vzoru
n1=80;      %pocet pouzitych prvku dulezitych pro rotorove vady
n2=101;     %dulezity prvek pro statorovou vadu
T=2;        %delka mereni
echo on;
%parkova transformace a FFT centrovane magnitudy
```

```
echo off;
tempmer=[Idnapr,Id25,Id50,Id75,Id85,Irnapr,Ir25,Ir50,Ir75,Ir85,I10znapr,I10z25,I10z50,I10z75,I10z85,I
2znapr,I2z25,I2z50,I2z75,I2z85,I5znapr,I5z25,I5z50,I5z75,I5z85];
temppark=zeros(N,Np);
j=1;
  for i=1:3:(3*Np),
    id=tempmer(:,i)-0.5*tempmer(:,i+1)-0.5*tempmer(:,i+2);   %id=ia-0.5*ib-0.5*ic;
       iq=sqrt(3)/2*(tempmer(:,i+1)-tempmer(:,i+2));         %iq=sqrt(3)/2*(ib-ic);
    mag=sqrt(id.^2+iq.^2);                                   %vypocet magnitudy
    mag=mag-mean(mag);                                       %centrovani magnitudy
    temppark(:,j)=mag;
    j=j+1;
  end
clear tempmer;
clear id;
clear iq;
clear mag;
spec=zeros(N/2,Np);
   for i=1:1:Np,
    temp=fft(temppark(:,i));          %fft transformace centrovane magnitudy
    spec(:,i)=abs(temp(1:N/2));       %absolutni hodnota jednostranneho spektra
  end
clear temp;
spec=spec/N;
vzorek=zeros(2,Np);
for i=1:1:Np
  a=max(spec(2:n1,i));
  vzorek(1,i)=a;
end
vzorek(2,:)=spec(201,:);
vzorek
clear spec;
echo on;
%testovani
%radek 1 dobry motor
%radek 2 vadny stator
%radek 3 vadny rotor
%prvnich 5 dobry, 5 rotor, 1 5stator
a=sim(net,vzorek)
save vysledek.txt a -ascii;
```