



**VYSOKÁ ŠKOLA  
CHEMICKO-TECHNOLOGICKÁ  
V PRAZE**

**Fakulta chemicko-inženýrská**

Ústav počítačové a řídicí techniky

**NÁZEV**

**DISERTAČNÍ PRÁCE**

AUTOR

**JAN VRBA**

ŠKOLITEL

**JAN MAREŠ**

ŠKOLITEL SPECIALISTA

**RATACHAN**

STUDIJNÍ PROGRAM

Chemické a procesní inženýrství (čtyřleté)

STUDIJNÍ OBOR

Technická kybernetika

ROK

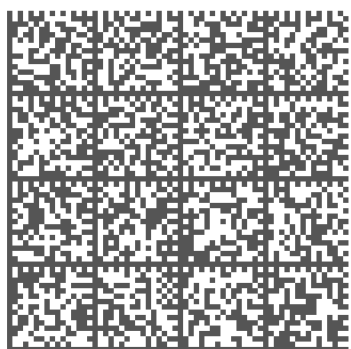
**2020**

**Faculty of Chemical Engineering**

Department of Computing and Control Engineering

NAME

**DISSERTATION**



AUTHOR	<b>JAN VRBA</b>
SUPERVISOR	<b>JAN MAREŠ</b>
SUPERVISOR SPECIALIST	<b>PANÍ SOBÍKOVÁ</b>
STUDY PROGRAMME	Chemical and Process Engineering
FIELD OF STUDY	Technical Cybernetics
YEAR	<b>2020</b>

Tato disertační práce byla vypracována na Ústavu počítačové a řídicí techniky v období OD-KDY – DOKDY.

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že na moji práci se vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon). Zejména se jedná o skutečnost, že Vysoká škola chemicko-technologická v Praze, popř. jiné vzdělávací zařízení, ve kterém jsem svou práci vypracoval, má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Pokud bych v budoucnu poskytl licenci o užití práce jinému subjektu, je Vysoká škola chemicko-technologická v Praze, popř. jiné vzdělávací zařízení, ve kterém jsem svou práci vypracoval, oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložil a to podle okolností až do jejich skutečné výše.

Souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne

.....

JMÉNO

## Poděkování

Rád bych poděkoval všem. #TODO





## **Souhrn**

Disertační práce se zabývá detekcí novosti, zejména pak algoritmem Extreme Seeking Entropy.  
#TODO

## **Klíčová slova**

*SLOVO* #TODO



## Summary

This dissertation deals with #TODO

## Keywords

*WORD* #TODO



# Obsah

<b>1</b>	<b>Přehled metod detekce novosti</b>	<b>21</b>
1.1	Algoritmus Learning Entropy . . . . .	21
1.2	Algoritmus Error and Learning Based Novelty Detection . . . . .	22
<b>2</b>	<b>Číslicové adaptivní filtry a algoritmy</b>	<b>25</b>
2.1	Adaptivní filtry . . . . .	27
2.1.1	Lineární FIR filtry . . . . .	27
2.1.2	Volterrovy filtry . . . . .	29
2.1.3	Polynomiální neuronové jednotky . . . . .	30
2.1.4	Fuzzy filtry . . . . .	31
2.2	Adaptivní algoritmy . . . . .	32
2.2.1	Algoritmy LMS a NLMS . . . . .	33
2.2.2	Algoritmus RLS . . . . .	35
2.2.3	Algoritmus Generalized Normalized Gradient Descent . . . . .	36
2.2.4	Gradient descent pro fuzzy filtry . . . . .	37
<b>3</b>	<b>Zobecněné Paretovo rozdělení</b>	<b>41</b>
3.1	Metoda Peak-over-Threshold . . . . .	41
3.2	Metody odhadu parametrů zobecněného Paretova rozdělení . . . . .	42
<b>4</b>	<b>Algoritmus Extreme Seeking Entropy</b>	<b>43</b>
<b>5</b>	<b>Výsledky experimentů</b>	<b>45</b>
5.1	Případové studie detekce novosti algoritmu Extreme Seeking Entropy . . . . .	45
5.1.1	Případová studie: chaotická časová řada Mackey-Glass a detekce pertubace	45
5.1.2	Případová studie: detekce změny rozptylu šumu v náhodném datovém toku . . . . .	46
5.1.3	Případová studie: detekce skokové změny parametrů generátoru signálu	48
5.1.4	Případová studie: detekce náhlé absence šumu . . . . .	48
5.1.5	Případová studie: detekce změny trendu . . . . .	48

5.1.6	Případová studie: detekce epilepsie v myším EEG . . . . .	48
5.1.7	Vyhodnocení úspěšnosti detekce skokové změny parametrů generátoru signálu . . . . .	49
5.2	Vyhodnocení úspěšnosti detekce změny trendu a evaluace ROC křivky . . . .	49
5.3	Vyhodnocení výpočetní náročnosti metod odhadu parametrů zobecněného Pare- tova rozdělení . . . . .	50
5.3.1	Motivace . . . . .	50
5.3.2	Specifikace experimentu . . . . .	50
5.3.3	Výsledky a diskuze . . . . .	54
5.4	Případová studie použití algoritmu Learning Entropy a adaptivního fuzzy filtru pro detekci změn stavů bioprocesu . . . . .	55
5.4.1	Popis bioprocesu a specifikace problému . . . . .	55
5.4.2	Experiment a zhodnocení . . . . .	57
<b>6</b>	<b>Závěr</b>	<b>61</b>
	<b>Publikace autora</b>	<b>63</b>
	<b>Literatura</b>	<b>65</b>
	<b>Příloha</b>	<b>67</b>

# Seznam použitých zkratek

LE	learning entropy
ELBND	error and learning based novelty detection
GEV	generalized extreme value
GNGD	generalized normalized gradient descend
NLMS	normalized least mean squares
POT	peak over threshold
LNU	linear neural unit
QNU	quadratic neural unit
SNR	signal-to-noise ratio
RLS	recursive least squares
ESE	extreme seeking entropy
#TODO	TODO





# Seznam symbolů

$\mathbb{N}$	množina přirozených čísel
$\mathbb{R}$	množina reálných čísel
$\mu$	rychlost učení
$\hat{y}$	výstup adaptivního filtru
$k$	diskrétní časový index
$e$	chyba predikce
$M_{ND}$	délka okna pro vyhodnocení změn adaptabilních parametrů
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$e$	chyba predikce
$\#TODO$	$\#TODO$



# Úvod

Tato disertační práce je věnována problematice využití adaptivních systémů při analýze dat. Vzhledem k exponenciálnímu celosvětovému nárůstu dat a ke zvyšování jejich variability roste i potřeba tato data analyzovat, kategorizovat a vytěžovat (TODO: nějaká citace k nárůstu dat). Analýzou dat rozumíme proces, kdy z nezpracovaných naměřených dat získáme nějakou interpretovatelnou informaci, s kterou pak lze dál pracovat. Jedna z možných důležitých interpretací nově získaných dat je, zda-li se nově získaná data nějakým zásadním způsobem odlišují od předchozích dat. Této problematice se věnuje obor detekce novosti, neboli anomálií, který spadá do oblasti vytěžování dat a strojového učení. Úspěšná detekce novosti pak může být využita k vícero účelům. Například k diagnostice sledovaného procesu, ke změně struktury nebo parametrů adaptivního modelu za účelem zlepšení predikce, z konkrétních aplikací pak k odhalení neoprávněného vniknutí do sítě nebo zneužití dat, v lékařství se detekce novosti používá k diagnostickým účelům, z průmyslových aplikací pak k detekci poruchy a monitoringu stavu strojů, senzorů, ev zpracování textových dat k detekci nových témat atd. Spektrum využití je velice široké.

V oblasti detekce novosti byla v posledních desetiletích intenzivního vývoje navržena celá řada algoritmů. Vzhledem k rostoucímu výpočetnímu výkonu a rozmanitosti analyzovaných dat rostla i potřeba nových algoritmů. Nové algoritmy typicky předčili ostatní algoritmy v rámci jedné aplikace, respektive v rámci jednoho typu dat. Doposud se však nepodařilo vytvořit algoritmus, který by ve všech, nebo alespoň ve významné části, oblastech použití předčil již publikované algoritmy. I proto vznikají v oblasti detekce novosti neustále nové přístupy, které navíc umožňují analyzovat nové typy dat.

Předkládaná disertační práce je členěna do pěti kapitol. První kapitola je věnována přehledu různých metod detekce novosti a obsahuje i oblasti jejich využití. Druhá kapitola obsahuje přehled adaptivních filtrů a metod, které byly v rámci práce použity. Třetí kapitola je věnována zobecněnému Paretovu rozdělení, které bylo použito v navrženém algoritmu detekce novosti. Čtvrtá kapitola obsahuje popis nově navrženého algoritmu nazvaném Extreme Seeking Entropy a představuje možnosti jeho použití v různých případech. Dále jsou zde výsledky tohoto algoritmu v porovnání s dalšími vybranými metodami adaptivní detekce novosti.



# 1 Přehled metod detekce novosti

Tato kapitola je věnována přehledu různých přístupů k detekci novosti, které se v posledních letech používají. Více pozornosti je věnováno algoritmům Learning Entropy (viz kap.1.1) a Error and Learning Based Novelty detection (viz kap. 1.2) které jsou v kapitole (XYZ) použity pro porovnání úspěšnosti detekce novosti.

## 1.1 Algoritmus Learning Entropy

Algoritmus Learning Entropy (LE) je algoritmem, který využívá adaptivních filtrů (obvykle LNU a HONU) k detekci novosti v datech [9, 10]. Autoři ukazují, že ne vždy, je novost v datech korelována s chybou predikce, a k detekci novosti navrhuji využívat přírůstky vah adaptivních filtrů  $\Delta \mathbf{w}$ , které jsou získány použitím gradientní metody a adaptací s každými novými daty. V první publikaci [10] je popsán multifraktální přístup, který využívá uspořádané množiny různých hodnot prahů

$$\boldsymbol{\alpha} = \{\alpha_1, \alpha_1, \dots, \alpha_{n_\alpha}\}, \alpha_1 > \alpha_2 > \dots > \alpha_{n_\alpha} \quad (1.1)$$

přičemž  $\forall i \in \{1, 2, \dots, n_\alpha\} : \alpha_i \in R$ . Míra novosti "Approximate Individual Sample Learning Entropy" je potom definovaná jako

$$E_A(k) = \frac{1}{n \cdot n_\alpha} \sum_{i=1}^n \sum_j^{n_\alpha} f(|\Delta w_i(k)|, \alpha_j) \quad (1.2)$$

kde funkce  $f$  je

$$f(\Delta w_i, \alpha_j) = \begin{cases} 1 & \text{if } |\Delta w_i(k)| > \alpha_j \cdot \overline{|\Delta w_i^M(k)|} \\ 0 & \text{if } |\Delta w_i(k)| \leq \alpha_j \cdot \overline{|\Delta w_i^M(k)|} \end{cases} \quad (1.3)$$

kde  $n$  je počet adaptivních parametrů filtru,  $n_\alpha$  je počet citlivostních prahů (respektive počet prvků množiny  $\boldsymbol{\alpha}$ ) a  $\overline{|\Delta w_i^M(k)|}$  průměrná hodnota přírůstku  $i$ -té adaptivní výhy v plovoucím okně délky  $M$ . Člen  $1/(n \cdot n_\alpha)$  normalizuje hodnotu AISLE takže  $E_A \in \langle 0, 1 \rangle$ . Toto hodnotu lze interpretovat tak že pokud se blíží 0, váhy se téměř nemění a z pohledu adaptace filtru se příliš neliší od minulých dat. Naopak vysoké hodnoty  $E_A$  znamenají, že dochází k velkým

změnám velkého množství adaptivních vah a data se nějakým způsobem liší od předchozích.

V publikaci [9] pak autoři navrhuji tzv. přímý algoritmus pro výpočet LE. Ta je vypočítána jako

$$LE(k) = \sum_{i=1}^n \max\{0, z(|\Delta w_i(k)|) - \beta\}; \quad (1.4)$$

kde funkce  $z$  je označovaná jako speciální  $z$ -score a je definovaná jako

$$z(|\Delta w_i(k)|) = \frac{|\Delta w_i(k)| - \overline{|\Delta \mathbf{w}_i^M(k-1)|}}{\sigma(|\Delta \mathbf{w}_i^M(k-1)|)} \quad (1.5)$$

kde

$$\overline{|\Delta \mathbf{w}_i^M(k-1)|} = \frac{\sum_{j=1}^M w_i(k-j-m)}{M} \quad (1.6)$$

je průměr hodnot přírůstku adaptivních vah, přičemž parametr  $M$  je délka plovoucího okna, z kterého je spočítán průměr přírůstků  $i$ -té váhy a parametr  $m$  je volitelný parametr pro data které vykazují periodicitu a  $\sigma(\sigma(|\Delta \mathbf{w}_i^M(k-1)|))$  je směrodatná odchylka těchto  $M$  přírůstků. Pro takto definovanou LE pak platí, že  $E(k) \in \langle 0, +\infty \rangle$ . Tento přístup umožňuje detekovat změny adaptivních parametrů, které jsou větší než jejich průměr zvětšený o  $\beta$  násobek jejich směrodatné odchylky a zároveň obchází nutnost výpočtu přes několik citlivostních prahů jako v případě 1.2. Pokud bychom chtěli detekovat i neobvykle malé změny adaptivních parametrů, navrhuji autoři vztah pro výpočet LE 1.4 modifikovat na tvar

$$E(k) = \sum_{i=1}^{n_w} z(|\Delta w_i(k)|); E \in R \quad (1.7)$$

který v případě negativních hodnot  $z(|\Delta w_i(k)|)$  nemusí poskytovat jasnou hranici mezi obvyklými a neobvyklými změnami adaptivních parametrů.

## 1.2 Algoritmus Error and Learning Based Novelty Detection

Algoritmus Error and Learning Based Novelty Detection (ELBND) [11, 12] je dalším z algoritmů detekce novosti, které využívají adaptivních filtrů, jejichž parametry jsou adaptovány vždy s novými daty. Každý naměřený vzorek dat je popsán vektorem hodnot, na základě přírůstku adaptivních vah a chyby filtru, definovaným jako

$$elbnd(k) = \Delta \mathbf{w} \cdot e(k) \quad (1.8)$$

který popisuje novost v daném vzorku dat. Skalární hodnotu popisující novost ve vzorku dat pak autoři navrhuji získat buď jako maximum z absolutních hodnot vektoru  $elbnd(k)$ , takže

skalární míra novosti  $ELBND(k)$  je definovaná jako

$$ELBND(k) = \max_{1 \leq i \leq n_w} |\Delta w_i(k) \cdot e(k)|. \quad (1.9)$$

Další možností je určit míru novosti jako součet absolutních hodnot jednotlivých přírůstků násobených chybou filtru, tedy

$$ELBND(k) = \sum_{i=1}^{n_w} |\Delta w_i(k) \cdot e(k)| \quad (1.10)$$

Uvedený algoritmus, využívající vztah 1.9, byl úspěšně použit k adaptivní klasifikaci EEG pacientů s demencí [13].





## 2 Číslicové adaptivní filtry a algoritmy

Tato kapitola je věnována diskretním adaptivním filtrům, které byly v průběhu práce na tématu dizertační práce použity (viz podkapitola 2.1 a algoritmům, které byli k jejich adaptaci použity (viz podkapitola 2.2). Z pohledu klasifikace filtrů dle impulsní charakteristiky se jedná o filtry s konečnou impulsní charakteristikou (viz podkapitola) i o filtry s nekonečnou impulsní charakteristikou (viz podkapitola). Z pohledu lineární závislosti adaptabilních parametrů potom na filtry lineární a nelineární v parametrech.

Obecně problém filtrace spočívá ve zpracování signálu filtrem tak, že ze signálu získáme nějakou užitečnou informaci[6].

Pro úplnost uveďme, že signálem rozumíme fyzikální veličinu, která se mění v čase, prostoru nebo v jakékoliv jiné nezávislé proměnné (obecně proměnných) [4]. V rámci dizertace jsou analyzovány pouze signály, které se mění v čase.

Jedno ze základních dělení signálů je rozdělení na signály spojitě a diskretní, přičemž signál může být spojitý, resp. diskretní v čase nebo amplitudě. Uvažujme signál spojitý v čase i amplitudě  $s(t)$ . Pro takový signál platí, že hodnota jeho amplitudy  $s(t) \in R$  a hodnota nezávislé proměnné  $t$  je z nějakého intervalu  $(t_1; t_2)$ , kde  $t_1 \in R$  a  $t_2 \in R$  (předkládaná dizertační práce se nezabývá komplexními signály). Signál spojitý v amplitudě a diskretní v čase vznikne vzorkováním původního signálu  $s(t)$  pomocí vzorkovací funkce ve tvaru

$$v(t) = \sum_{k=-\infty}^{\infty} \delta(t - k\Delta t) \quad (2.1)$$

kde  $\Delta t$  je vzorkovací perioda (uvažujeme konstantní vzorkovací periodu) a  $\delta$  je, z pohledu teorie distribucí, lineární funkcionál na prostoru testovacích funkcí  $\varphi$  (všech hladkých funkcí na  $R$  s kompaktním supportem, které mají požadovaný počet derivací) definovaný jako

$$\langle \delta, \varphi \rangle = \varphi(0) \quad (2.2)$$

pro každou testovací funkci  $\varphi$  [7]. Pro navzorkovaný signál tedy platí

$$s(k) = s(t) \cdot v(t) \quad (2.3)$$

a výsledkem je posloupnost vzorků  $s(k)$ . Signál spojitý v čase a diskretní v amplitudě získáme aplikací kvantizační funkce  $Q(s)$ , která převede hodnotu signálu na číslo z nějaké množiny přípustných hodnot, (typicky to bývá celé číslo, případně číslo s plovoucí desetinnou čárkou). Existuje celá řada kvantizačních algoritmů (více viz [8]), takže pro úplnost uvedeme pouze základní uniformní kvantizátor s velikostí kvantizačního kroku definovanou jako

$$q = \frac{s_{max} - s_{min}}{L} \quad (2.4)$$

kde  $L$  určuje počet intervalů o délce  $q$ . Potom kvantizovanou hodnotu můžeme určit jako

$$Q(s) = \left\lfloor s - \frac{s_{min}}{q} \right\rfloor q + \frac{q}{2} + s_{min} \quad (2.5)$$

Signál, který je diskretní v čase i amplitudě bývá označován jako digitální signál a právě těmito signály se předložená dizertační práce zabývá.

Signály dále můžeme rozdělit na skalární a vektorové. Vektorových (někdy též označovaných jako vícekanalový) signálem je například EEG. Některé systémy pro měření EEG využívají až 256 kanálů [5]. Takový signál můžeme v časovém okamžiku  $k$  reprezentovat 256ti dimenzionálním vektorem

$$\mathbf{s}(k) = [s_1(k), \dots, s_{256}(k)]^T \quad (2.6)$$

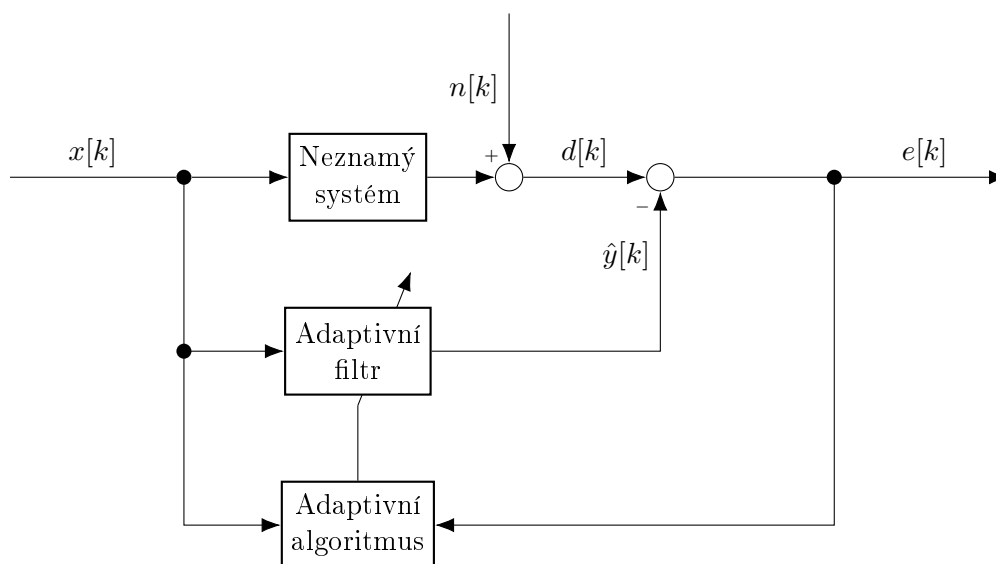
kde  $i$ -tá složka  $s_i(k)$  odpovídá signálu  $i$ -tého kanálu. Skalární signál (někdy též označovaný jako jednocanalový) je pak např. právě  $i$ -tá složka signálu  $\mathbf{s}(k)$ .

Z pohledu počtu nezávislých proměnných, jejichž funkcí lze signál vyjádřit lze rozlišovat mezi signály jednorozměrnými a vícerozměrnými. Jednorozměrným signálem je například záznam jednoho kanálu EEG, kde hodnota signálu  $s_i(k)$  je závislá pouze na čase. Vícerozměrným signálem může např. digitální obraz, jehož intenzita je funkcí souřadnic  $I(x, y)$ .

Další dělení signálu je na signály deterministické a náhodné. Pokud můžeme popsat signál pomocí explicitního matematického zápisu, tabulky nebo pravidlem tak, že pro každý časový okamžik získáme jednoznačnou hodnotu, jedná se o signál deterministický. V opačném případě jsou signály, jejichž hodnota lze určit pouze s nějakou pravděpodobností označovány jako náhodné (stochastické). Typickým příkladem náhodného signálu je např. bílý šum. Ze stochastických signálů jsou z pohledu této práce je zajímavé ještě dělení na ergodické a stacionární signály. Pro stacionární signály je jejich pravděpodobnostní popis nezávislý na počátku časové osy. Ergodické signály jsou pak signály jejichž statistické vlastnosti lze určit z libovolné, dostatečně velké, množiny dat a tyto vlastnosti se v čase nemění.

Filtr můžeme použít v následujících základních úlohách zpracování signálů:

- filtrace
- vyhlazování
- predikce



Obrázek 2.1: Schéma adaptivní filtrace

V rámci této práce byli všechny filtry využity k predikci, jejíž chyba  $e$  byla využita k adaptaci parametrů daného filtru (více viz podkapitola 2.2).

## 2.1 Adaptivní filtry

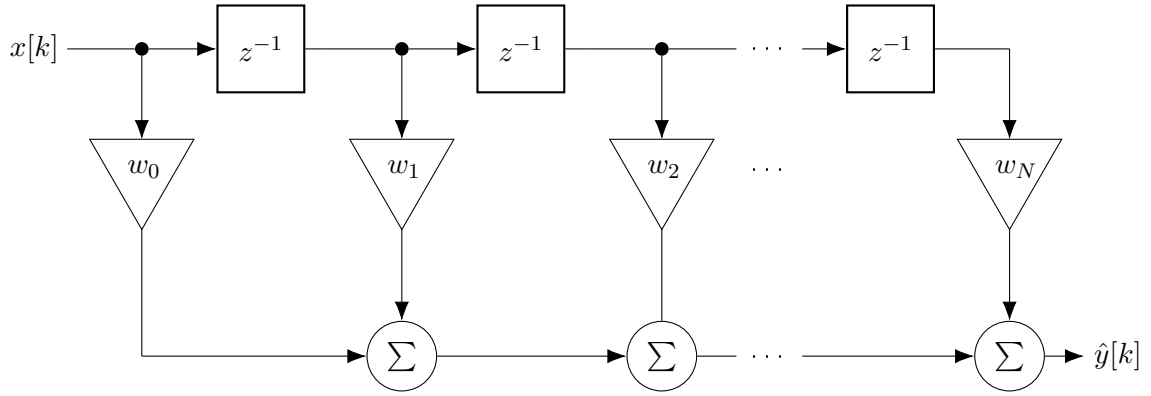
V této podkapitole jsou stručně popsány adaptivní filtry, které byly v rámci práce použity. FIR (finite impulse response) adaptivní filtry jsou popsány v podkapitole 2.1.1, Volterrové filtry v podkapitole 2.1.2 a adaptivní fuzzy filtry v podkapitole 2.1.4.

### 2.1.1 Lineární FIR filtry

Výstup FIR (finite-impulse-response) filtru  $\hat{y} \in \mathbb{R}$  v diskrétním čase  $k \in \mathbb{Z}$  je popsán rovnicí

$$\hat{y}[k] = \sum_{i=0}^N w_i \cdot x[k-i] \quad (2.7)$$

kde  $w_i \in \mathbb{R}$  hodnota  $i$ -tého koeficientu,  $x[k-i]$  je hodnota vstupu  $x \in \mathbb{R}$  posunutá o  $i$  hodnot v čase, hodnota  $N$  je řád filtru. Pokud jsou koeficienty FIR filtru v čase adaptovány, přejde



Obrázek 2.2: Blokový diagram FIR filtru

rovnice (2.7) do tvaru

$$\hat{y}[k] = \sum_{i=0}^N w_i[k] \cdot x[k-i] \quad (2.8)$$

kde člen  $w_i[k]$  reprezentuje hodnoty koeficientů filtru v diskrétním čase  $k$ . V následujícím textu, již uvažujeme, že veškeré adaptivní filtry mají v čase proměnné koeficienty  $w$ . Někdy se pro popis výstupu FIR filtru využívá operátoru konvoluce, potom pro výstup FIR filtru platí

$$\hat{y}[k] = h[k] * x[k] = x[k] * h[k] \quad (2.9)$$

Hodnoty  $h[k]$  v reprezentují impulzní odezvu filtru. Impulzní odezva filtru je definovaná jako

$$h[k] = \sum_{i=0}^N w_i \cdot \delta[k-i] = \begin{cases} w_i & 0 \leq k \leq N \\ 0 & k < 0 \vee k > N \end{cases} \quad (2.10)$$

Konečnost impulzní odezvy je dána konečným počtem  $N+1$  koeficientů filtru. Pokud koeficienty filtru splňují podmínku  $\forall i : w_i < \infty$ , potom je FIR filtr stabilní.

Uvedný filtr (viz rovnice (2.7)) lze zapsat ve tvaru

$$\hat{y}(k) = \mathbf{w}(k) \cdot \mathbf{x}(k) \quad (2.11)$$

kde  $\mathbf{w}(k)$  je vektor vah

$$\mathbf{w}(k) = [w_1, \dots, w_N] \quad (2.12)$$

a  $\mathbf{x}(k)$  je vstupní vektor

$$\mathbf{x}^T(k) = [x(k), x(k-1), \dots, x(k-N)] \quad (2.13)$$

### 2.1.2 Volterrovy filtry

Jeden z filtrů, který se používá k modelování nelineárních systémů je nelineární Voltérrův filtr (někdy uváděný pod označením Higher Order Neural Unit - HONU). Výstup  $\hat{y}[k]$  těchto filtrů, využívajících zkrácených Voltérových řad, je popsán rovnicí

$$\begin{aligned} \hat{y}[k] = w_0 + \sum_{i=0}^{N-1} w_1(i)x(k-i) + \sum_{i_1=0}^{N-1} \sum_{i_2=i_1}^{N-1} w_2(i_1, i_2)x(k-i_1)x(k-i_2) + \dots \\ \dots \sum_{i_1=0}^{N-1} \dots \sum_{i_p=i_{p-1}}^{N-1} w_p(i_1, \dots, i_p)x(k-i_1) \dots x(k-i_p) \end{aligned} \quad (2.14)$$

kde  $w_0$  je konstanta,  $\{w_j(i_1, \dots, i_j), 1 \leq j \leq p\}$  je množina koeficientů Volterrovo jader  $j$ -tého řádu a  $x(k)$  je vstupní signál. Filtr pracuje s pamětí  $N$  vzorků, parametr  $p$  určuje řád filtru. Analogický zápis využívající vektorové násobení

$$\hat{y}[k] = \mathbf{w} \cdot \mathbf{colx} \quad (2.15)$$

kde  $\mathbf{w}$  je uspořádaný vektor koeficientů Volterrovo jader a  $\mathbf{colx}$  uspořádaný vektor ve tvaru

$$\begin{aligned} \mathbf{colx} = [1, x[0], \dots, x[N-1], x[1] \cdot x[2], \dots, x[1] \cdot x[N-1], x[2]^2, x[2] \cdot x[3], \dots, \\ x[2] \cdot x[N-1], \dots, x[N-2] \cdot x[N-1], x[N-1]^2, \dots, \dots, x[N-1]^p]^T \end{aligned} \quad (2.16)$$

Určitým faktorem limitujícím použití Volterrovo filtrů je vysoký počet jejich parametrů, přičemž každé zvýšení počtu vzorků v paměti, nebo řádu filtru, výrazně zvýší počet jeho parametrů. Pro počet parametrů  $m$  Volterrova filtru s délkou paměti  $N$  a řádu  $p$  je dán jako

$$m(p, N) = \frac{(N+p)!}{N!p!} \quad (2.17)$$

kde  $N!$  je faktoriál  $N$  a  $p!$  je faktoriál  $p$ . Např. pro  $N = 3$  a  $p = 3$  má filtr 20 parametrů, pro  $N = 4$  a  $p = 3$  má 35 parametrů, pro  $N = 4$  a  $p = 4$  již 70 parametrů. Pro zpracování signálů v reálném čase je tedy využití Volterrových filtrů s velkou pamětí a vysokým řádem, vzhledem k vysokému počtu parametrů, komplikované. Voltérrův filtr druhého řádu bývá někdy označován Quadratic neural unit (QNU) a třetího řádu Cubic neural unit (CNU) [viz Ivo nebo IGI]. Voltérrův filtr prvního řádu je variantou standartního FIR filtru (viz kap.) v případě, že  $w_0 = 0$ . V případě, že  $w_0 \neq 0$ , je tento filtr typu IIR, tedy má nekonečnou impulsní charakteristiku.

### 2.1.3 Polynomiální neuronové jednotky

Jedním z typů nelineární filtrů, které jsou ale lineární v adaptivních parametrech jsou polynomiální neuronové jednotky, někdy též nazývané HONU (Higher Order Neural Units). Výstup HONU  $p$ -tého řádu je definován jako

$$\hat{y}(k) = \sum_{i_1=0}^n \sum_{i_2=i_1}^n \cdots \sum_{i_p=i_{p-1}}^n w_{i_1, i_2, \dots, i_p} x_{i_1} \cdot x_{i_2} \cdots x_{i_p} \quad (2.18)$$

přičemž člen  $x_{0, \dots, 0} = 1$  je označován jako bias,  $w_{i_1, i_2, \dots, i_p}$  jsou váhy a  $x_{i_j}$  je  $j$ -tý vstup. Uvedený filtr lze reprezentovat pomocí násobení vektorů, obdobně jako Volterrový filtry, jako

$$\hat{y}(k) = \mathbf{w} \cdot \mathbf{colx} \quad (2.19)$$

kde  $w$  je uspořádaný vektor adaptivních parametrů

$$\mathbf{w} = [w_{0, \dots, 0}, \dots, w_{n, \dots, n}] \quad (2.20)$$

a  $\mathbf{colx}$  je odpovídajícím způsobem uspořádaný vektor vstupů ve tvaru

$$\mathbf{colx} = [1, x_{0, \dots, 1}, \dots, x_{n, \dots, n}]^T \quad (2.21)$$

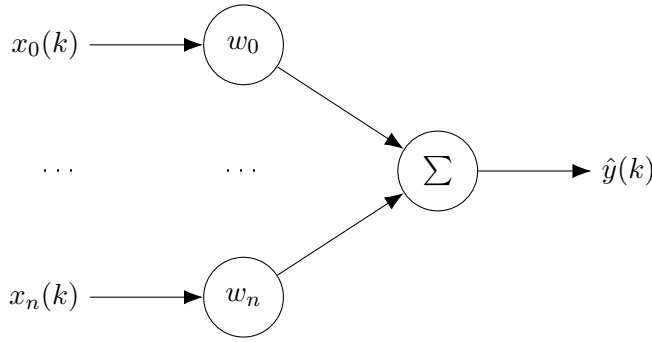
přičemž pokud vektor vstupů obsahuje časově spožděné vzorky vstupního signálu, jsou HONU identické s Volterrovými filtry. Pokud je vstupem  $n+1$ -dimenzionální vektor různých vstupů, tak je HONU kombinačním filtrem. HONU prvního řádu bývá označována jako lineární neuronová jednotka (LNU). Pokud jsou vstupem do LNU časově posunuté hodnoty vstupního signálu, je identická s klasickým FIR filtrem (viz kapitola 2.1.1), protože její výstup je definován jako

$$\hat{y}(k) = \sum_{i=0}^n w_i \cdot x(k-i) \quad (2.22)$$

Pokud je vstupem  $n+1$ -dimenzionální vektor  $n+1$  různých vstupů, potom je LNU klasickým lineárním kombinačním filtrem (viz obr. 2.3). Použijeme-li terminologii neuronových sítí, potom je lineárním neuronem, kde  $\mathbf{w}$  je vektor synaptických vah,  $\mathbf{x}$  je vstupní vektor a přenosová funkce tohoto neuronu realizuje lineární kombinaci.

$$\hat{y}(k) = \sum_{i=0}^n w_i \cdot x_i(k) \quad (2.23)$$

Často používanými HONU jsou jednotky druhého (quadratic neural unit - QNU) a třetího řádu (cubic neural unit - CNU). Podobně jako pro Volterrový filtry, i úskalím použití HONU vyššího řádu s velkou pamětí je velký počet parametrů.



Obrázek 2.3: LNU jako lineární kombinační filtr

#### 2.1.4 Fuzzy filtry

Jedním z rozšířených typů nelineárních adaptivních filtrů jsou filtry založené na fuzzy logice. (Fuzzy adaptive filters, with application to nonlinear channel equalization) V rámci práce byl použit fuzzy adaptivní filtr, tvořený Mamdaniho fuzzy systémem se součinným inferenčním mechanismem a defuzzifikací využívající metody těžiště typických hodnot. Báze pravidel tohoto fuzzy filtru je tvořena  $M$  pravidly, kdy  $l$ -té pravidlo je ve tvaru

$$Ru^l : IF x_1 is A_1^l and x_2 is A_2^l and \dots x_n is A_n^l THEN \hat{y} is B^l \quad (2.24)$$

kde  $A_i^l$  je  $i$ -tá množina ve vstupním prostoru  $U \subset R^n$ ,  $B^l$  je fuzzy množina ve výstupním prostoru  $V \subset R$  a  $x_i \in U$  a  $\hat{y} \in V$  jsou lingvistické proměnné. Fuzzy množiny ve výstupním prostoru jsou singletony, obsahují tedy jediný prvek  $x^*$  z univerza  $X$  a pro jejich funkce příslušnosti (charakteristické funkce) platí, že

$$\mu_B(x^*) = 1 \quad (2.25)$$

takže jádro této množiny je identické s jejím nosičem a obsahují pouze jeden stejný prvek.

$$ker(B) = supp(B) = \{1/x^*\} \quad (2.26)$$

Studovaný fuzzy systém využívá součinné konjukce a Larsenovy implikace. Tím se převede vyhodnocení implikace na několik součinů tak, že pro  $j$ -té pravidlo platí

$$\begin{aligned} [\mu_{A_1^j}(x_1) AND \mu_{A_2^j}(x_2) AND \dots AND \mu_{A_n^j}(x_n)] &\implies \mu_{B^j}(\hat{y}) = \\ &= \mu_{A_1^j}(x_1) \cdot \mu_{A_2^j}(x_2) \cdot \dots \cdot \mu_{A_n^j}(x_n) \cdot \mu_{B^j}(\hat{y}) \end{aligned} \quad (2.27)$$

kde  $\mu_{A_i^j}(x_i)$  je funkce příslušnosti k  $i$ -té množině  $j$ -tého pravidla ve vstupním prostoru a  $\mu_{B^j}$  je funkce příslušnosti ke množině ve výstupním prostoru.

Defuzzifikace převádí fuzzy do její reprezentace pomocí jediného čísla z množiny ostrých hodnot. Metodou těžiště typických hodnot se výstup fuzzy systému určí podle rovnice

$$y^* = \frac{\sum_{j=1}^M \bar{b}^j z_j}{\sum_{j=1}^M z_j} \quad (2.28)$$

kde  $\bar{b}^j$  je střed  $j$ -té fuzzy množiny reprezentující příslušné pravidlo a  $z_j$  je její váha ve smyslu hodnoty funkce příslušnosti tohoto pravidla.

Výstup uvedeného fuzzy systému je potom ve tvaru

$$\hat{y}(\mathbf{x}) = \frac{\sum_{j=1}^M \bar{b}^j [\prod_{i=1}^n \mu_i^j(x_i)]}{\sum_{j=1}^M [\prod_{i=1}^n \mu_i^j(x_i)]} \quad (2.29)$$

přičemž  $\mathbf{x}$  je vektor vstupů o délce  $n$

$$\mathbf{x} = [x_1, \dots, x_n] \quad (2.30)$$

a  $\bar{b}^j$  je střed výstupní množiny  $B^j$ , což je fuzzy množina  $j$ -tého pravidla.

Funkce filtru lze znázornit třívrstvou dopřednou sítí (viz Figure 2.4).

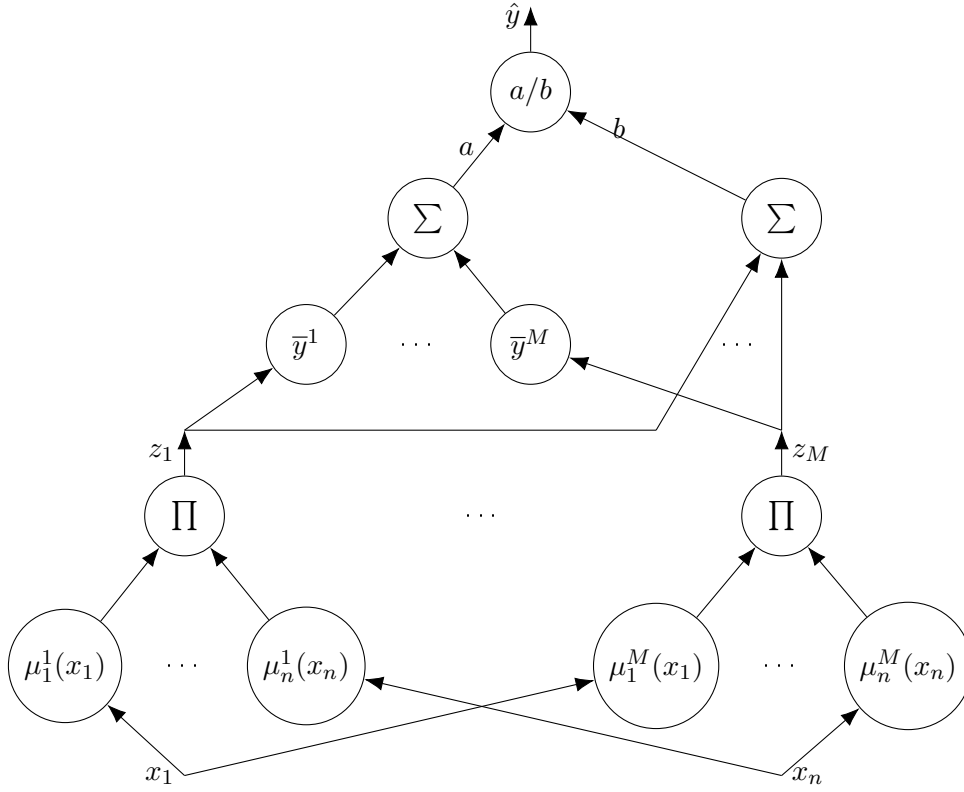
V první vrstvě jsou určeny váhy jednotlivých pravidel, tedy jsou vypočteny hodnoty  $z_j$ , kde  $j = 1, \dots, M$ . V druhé vrstvě jsou váhy jednak pronásobeny hodnotami středů a sečteny (výpočet  $a$ ), druhak jsou váhy jednotlivých pravidel sečteny ( $b$ ). Ve třetí vrstvě se pak vypočte výstup fuzzy systému jako  $\hat{y} = \frac{a}{b}$  (více viz podkapitola 2.2.4).

Nespornou výhodou uvedeného fuzzy adaptivního filtru je, že tvoří tzv. univerzální aproximátor. Při vhodně zvolené bázi pravidel a typu funkcí příslušnosti množin ve vstupním prostoru tak dokážet aproximovat libovolnou funkci s libovolně velkou přesností (více viz. [? ])

## 2.2 Adaptivní algoritmy

V této podkapitole jsou popsány adaptivní algoritmy, které byly v rámci dizertační práce vyzkoušeny. Jedná se o LMS a NLMS (viz podkap. 2.2.1), Generalized Normalized Gradient Descent (GNGD, viz podkap. 2.2.3) a algoritmus Gradient descent ve verzi pro adaptivní fuzzy filtry s Gausovskými funkcemi příslušnosti ve vstupním prostoru (viz podkap. 2.2.4).





Obrázek 2.4: Fuzzy filtr jako dopředná síť

### 2.2.1 Algoritmy LMS a NLMS

Při použití LMS algoritmu optimální hodnoty adaptivních parametrů filtru  $\mathbf{w}(k+1)$  minimalizují střední kvadratickou chybu predikce  $J(k)$  definovanou jako

$$J(k) = E[|e(k)|^2] \quad (2.31)$$

kde  $E[\cdot]$  značí střední (očekávanou) hodnotu, a  $e(k)$  je chyba predikce, definovaná jako rozdíl výstupu adaptivního filtru a skutečné hodnoty

$$e(k) = d(k) - \hat{y}(k). \quad (2.32)$$

Hodnoty adaptivních parametrů filtru jsou nalezeny gradientním algoritmem. S každými nově získanými daty jsou potom váhy filtru upraveny podle rovnice

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta \mathbf{w}(k) \quad (2.33)$$

přičemž pro LMS algoritmus je přírůstek vah definován jako

$$\Delta \mathbf{w}(k) = -\frac{\mu}{2} \nabla_{\mathbf{w}} E[|e(k)|^2] = \mu E[\mathbf{x}(k)e(k)] \quad (2.34)$$

kde  $\mu$  je rychlost učení (velikost kroku) ovlivňující rychlost konvergence algoritmu a  $\nabla$  je operátor nabra

$$\nabla = \left( \frac{\partial}{\partial w_1}, \dots, \frac{\partial}{\partial w_n} \right). \quad (2.35)$$

Protože jsou parametry filtru přepočítány s každými nově získanými daty (online), je možné nahradit očekávanou hodnotu  $E[\mathbf{x}(k)e(k)]$  hodnotou okamžitou. Výpočet nových hodnot parametrů adaptivního filtru tak přejde do tvaru

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{x}(k)e(k) \quad (2.36)$$

přičemž pro konvergenci a stabilitu algoritmu musí být splněna podmínka pro velikost rychlosti učení

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (2.37)$$

kde  $\lambda_{max}$  je největší vlastní číslo autokorelační matice

$$R_{xx} = E[\mathbf{x}(k)\mathbf{x}^T(k)]. \quad (2.38)$$

Pokud není podmínka splněna, pak je algoritmus nestabilní a hodnoty  $\mathbf{w}(k)$  divergují. Pokud je naopak velikost rychlosti učení  $\mu$  příliš malá, váhy konvergují pomalu.

Algoritmus NLMS řeší problém klasického LMS algoritmu, který v případě nevhodně škálovaného vstupu  $\mathbf{x}(k)$  ztrácí stabilitu. Tento problém je vyřešen normalizací vstupu. Velikost přírůstku vah je tedy

$$\Delta \mathbf{w}(k) = \mu \frac{e(k)\mathbf{x}(k)}{\mathbf{x}^T(k)\mathbf{x}(k)}. \quad (2.39)$$

Podmínka pro velikost rychlosti učení zajišťující stabilitu algoritmu je v případě, že vstupní signál  $x(k)$  není korelovaný s aditivním šumem

$$0 < \mu < 2. \quad (2.40)$$

Velikost optimální rychlosti učení je ovlivněna vlastnostmi aditivního šumu  $n(k)$  a v případě, že tento šum není korelovaný se vstupním signálem  $\mathbf{x}$  je dána jako

$$\mu_{optimal} = \frac{E[|d(k) - \hat{y}(k)|^2]}{E[|e(k)|^2]} \quad (2.41)$$

Problém nastane v případě, že vektor  $\mathbf{x}(k)$  je nulový. Z tohoto důvodu, se přidává do jmenova-

tele v rovnici 2.39 malá pozitivní konstanta  $\epsilon > 0$ , která řeší problém s potenciálním dělením nulou. Změna adaptivních vah filtru je v tomto případě dána jako

$$\Delta \mathbf{w}(k) = \mu \frac{e(k) \mathbf{x}(k)}{\mathbf{x}^T(k) \mathbf{x}(k) + \epsilon} \quad (2.42)$$

a podmínka pro velikost rychlosti učení přejde do tvaru

$$0 < \mu < 2 + \frac{\epsilon}{\mathbf{x}^T(k) \mathbf{x}(k)}. \quad (2.43)$$

### 2.2.2 Algoritmus RLS

Při použití rekurzivní metody nejmenších čtverců (RLS - recursive least squares) je odhad parametrů filtru získán na základě minimalizace kritériální funkce ve tvaru

$$J(k) = \sum_{j=j_1}^k \lambda^{k-j} e(j)^2 \quad (2.44)$$

která je exponenciálně váženým součtem chyb výstupu posledních  $k - j_1$  vzorků, přičemž parametr  $\lambda \in (0; 1)$  je tzv. faktor exponenciálního zapomínání. Chyba výstupu je pak definována jako

$$e(j) = d(j) - \hat{y}(j) \quad (2.45)$$

kde

$$\hat{y}(j) = \mathbf{w}^T(k) \mathbf{x}(j) \quad (2.46)$$

přičemž  $\mathbf{w}^T(k)$  je vektor adaptivních parametrů filtru

$$\mathbf{w}^T(k) = [w_0(k), w_1(k), \dots, w_n(k)] \quad (2.47)$$

a  $\mathbf{x}(k)$  je vstupní vektor v diskrétním časovém okamžiku  $k$  obsahující posledních  $n + 1$  vzorků a definovaný jako

$$\mathbf{x}(j) = [x(j), x(j-1), \dots, x(j-n)]^T \quad (2.48)$$

kde parametr  $n$  se označuje jako řád filtru. Z uvedené kritériální funkce 2.44 je tedy zřejmé, že historicky starší chyby výstupu mají exponenciálně klesající význam a vzorky, které jsou starší než  $k - j_1$  vzorků, nejsou pro odhad parametrů filtru použity. Pro hodnoty adaptivních parametrů, které minimalizují výše uvedenou kritériální funkci (2.44) v diskrétním časovém okamžiku  $k$  pak platí

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{P}(k) \mathbf{x}(k) [d(k) - \mathbf{x}^T(k) \mathbf{w}(k-1)] \quad (2.49)$$

kde matice  $\mathbf{P}(k)$  je

$$\mathbf{P}(k) = \frac{1}{\lambda} \left[ \mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{P}(k-1)}{\lambda + \mathbf{x}^T(k)\mathbf{P}(k-1)\mathbf{x}(k)} \right]. \quad (2.50)$$

Člen  $\mathbf{x}^T(k)\mathbf{w}(k-1)$  v rovnici (2.49) reprezentuje apriorní chybu filtru, která je vypočtena ještě před korekcí adaptivních vah. Kritérium, která je minimalizováno (viz rovnice (2.44)) ale obsahuje aposteriorní chyby filtru, která je vypočtena po korekci adaptivních vah (zde se nabízí určitá podobnost s Kalmanovým filtrem, více viz [3]). Dále poznamenejme, že  $\mathbf{P}(k)$  je inverzní maticí k vážené výběrové kovarianční matici  $\mathbf{R}_x(k)$  definované jako

$$\mathbf{R}_x(k) = \lambda^k \mathbf{R}_x(0) + \sum_{j=1}^k \lambda^{k-j} \mathbf{x}(j)\mathbf{x}^T(j) = \lambda \mathbf{R}_x(k-1) + \mathbf{x}(k)\mathbf{x}^T(k) \quad (2.51)$$

přičemž  $\mathbf{R}_x(0)$  je počáteční hodnota. V praxi se počáteční hodnota matice  $\mathbf{P}(0) = \mathbf{R}_x^{-1}(0)$  volí jako

$$\mathbf{P}(0) = \delta \cdot \mathbf{I} \quad (2.52)$$

kde  $\delta$  je dostatečně velká pozitivní konstanta a  $\mathbf{I}$  je jednotková matice (v některé literatuře nazývaná jako matice identity). Pro signály s vysokým poměrem výkon-šum se volí malé hodnoty  $\delta$ , pro signály s malým poměrem výkon-šum pak velké hodnoty  $\delta$ . Pokud je k dispozici apriorní informace o  $\sigma_x^2$  tedy varianci vstupního signálu  $x(k)$ , volí se hodnota konstanty podle [2] jako

$$\delta > 100\sigma_x^2. \quad (2.53)$$

Počáteční hodnota adaptivních parametrů se obvykle volí jako

$$\mathbf{w}(0) = 0. \quad (2.54)$$

### 2.2.3 Algoritmus Generalized Normalized Gradient Descent

Algoritmus Generalized Normalized Gradient Descent (GNGD) řeší problém případné pomalé konvergence algoritmu NLMS zavedením dalšího kompenzačního členu, který ovlivňuje velikost kroku při gradientní adaptaci. Nejprve uvažujme kvadratickou kritériální funkce ve tvaru

$$J(k) = \frac{1}{2} e^2(k) \quad (2.55)$$

a adaptaci parametrů ve tvaru (2.42). Malou pozitivní konstantu  $\epsilon$  nahradíme dalším adaptivním členem

$$\epsilon(k+1) = \epsilon(k) - \rho \nabla_{\epsilon(k-1)} J(k) \quad (2.56)$$

přičemž

$$\begin{aligned} \frac{\partial J(k)}{\partial \epsilon(k-1)} &= \frac{\partial J(k)}{\partial e(k)} \frac{\partial e(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \mathbf{x}(k)} \frac{\partial \mathbf{w}(k)}{\partial \eta(k-1)} \frac{\partial \eta(k-1)}{\partial \epsilon(k-1)} = \\ &= \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{(\mathbf{x}^T(k)\mathbf{x}(k) + \epsilon(k-1))^2}. \end{aligned} \quad (2.57)$$

Adaptace parametrů je tedy dána jako

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta(k)e(k)\mathbf{x}(k) \quad (2.58)$$

kde

$$\eta(k) = \frac{\mu}{\mathbf{x}^T(k)\mathbf{x}(k) + \epsilon(k)} \quad (2.59)$$

$$\epsilon(k) = \epsilon(k-1) - \rho\mu \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{(\mathbf{x}^T(k)\mathbf{x}(k) + \epsilon(k-1))^2} \quad (2.60)$$

přičemž parametr  $\rho$  je parametr adaptace velikosti kroku při spuštění algoritmu. Algoritmus GNGD je vhodný pro zpracování nelineárních a nestacionárních signálů.

## 2.2.4 Gradient descent pro fuzzy filtry

Při použití metody gradient descent pro adaptivní fuzzy filtry je potřeba nejdříve specifikovat strukturu filtru, tedy počet pravidel, množiny ve vstupním a výstupním prostoru, typ inferenční metody, fuzzifikace a defuzzifikace. Uvažujme fuzzy systém specifikovaný v podkapitole 2.1.4, v jehož vstupním prostoru  $U \subset R^n$  jsou Gaussovske funkce příslušnosti ve tvaru

$$\mu_{A_i^j}(x_i) = \exp\left[-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right] \quad (2.61)$$

kde  $\bar{x}_i^j$  je středem  $i$ -té vstupní množiny  $j$ -tého pravidla a  $\sigma_i^j$  je parametr, určující tvar, respektive šířku, této fuzzy množiny. Zobrazení, realizované výše specifikovaným fuzzy systémem je dáno rovnicí

$$\hat{y}(x) = \frac{\sum_{j=1}^M \bar{b}^j \left[ \prod_{i=1}^n \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right) \right]}{\sum_{j=1}^M \left[ \prod_{i=1}^n \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right) \right]} \quad (2.62)$$

přičemž parametr  $M$  určující počet množin je vzhledem k adaptaci parametrem fixním a parametry  $\bar{b}^j$ ,  $\sigma_i^j$  a  $\bar{x}_i^j$  jsou parametry, které se adaptují. Dále uvažujme množinu  $N$  dvojic vstup-výstup, kde  $N$  odpovídá počtu vzorků experimentu.

$$\{\mathbf{x}^d(k), d(k)\}, \quad k = 1, 2, \dots, N \quad (2.63)$$

Při použití algoritmu gradient descent pro fuzzy filtr v kontextu detekce novosti se vždy s nově naměřenými daty tento filtr adaptuje. Adaptace probíhá s každými daty na základě minimalizace kritériální funkce

$$J(k) = \frac{1}{2}[\hat{y}(\mathbf{x}(k)) - d(k)]^2 \quad (2.64)$$

kteřá je zvolená tak, aby měla právě jeden globální extrém, přičemž parametry jsou adaptovány dokud není dosaženo dostatečně malé chyby, nebo dokud není překročen předem stanovený maximální počet iterací  $q_{max}$ . Volba Gaussovských funkcí příslušnosti ve vstupním prostoru je výhodná z hlediska výpočtu derivace podle adaptabilních parametrů, neboť tato derivace existuje v každém bodě.

$$\bar{b}^j(q+1) = \bar{b}^j(q) - \mu \frac{\partial e}{\partial \bar{b}^j} \Big|_q = \bar{b}^j(q) - \mu \frac{\hat{y} - d(k)}{b(q)} z^j(q) \quad (2.65)$$

$$\begin{aligned} \bar{x}_i^j(q+1) &= \bar{x}_i^j(q) - \mu \frac{\partial e}{\partial \bar{x}_i^j} \Big|_q = \\ &= \bar{x}_i^j(q) - \mu \frac{\hat{y} - d(k)}{b(q)} [\bar{b}^j(q) - \hat{y}] z^j(q) \frac{2[x_i(k) - \bar{x}_i^j(q)]}{\sigma_i^{j2}(q)} \end{aligned} \quad (2.66)$$

$$\begin{aligned} \sigma_i^j(q+1) &= \sigma_i^j(q) - \mu \frac{\partial e}{\partial \sigma_i^j} \Big|_q = \\ &= \sigma_i^j(q) - \mu \frac{\hat{y} - d(k)}{b(q)} [\bar{b}^j(q) - \hat{y}] z^j(q) \frac{2[x_i(k) - \bar{x}_i^j(q)]^2}{\sigma_i^{j3}(q)} \end{aligned} \quad (2.67)$$

S každými nově získanými daty (v diskretním časovém okamžiku  $k$ ) jsou tedy v  $q$ -té iteraci hodnoty parametru  $\bar{b}^j$  vypočítány podle rovnice (2.65), parametru  $\bar{x}_i^j$  podle rovnice (2.66) a parametru  $\sigma_i^j$  podle rovnice (2.67). Parametr  $\mu$  je fixní a určuje velikost kroku. Algoritmus lze shrnout následujícími 5-ti kroky:

1. **Krok** Určení počtu pravidel a počáteční nastavení parametrů  $\bar{b}^j(0)$ ,  $\bar{x}_i^j(0)$ ,  $\sigma_i^j(0)$  a velikosti kroku  $\mu$ .
2. **Krok** Pro  $ktou$  dvojici  $(\mathbf{x}(k), d(k))$  v  $q$ -té iteraci jsou vypočteny hodnoty výstupních vrstev fuzzy systému (viz obr 2.4) podle rovnic (2.68), (2.69), (2.70) a (2.71).
3. **Krok** Výpočet nových hodnot parametrů  $\bar{b}^j(q+1)$ ,  $\bar{x}_i^j(q+1)$ ,  $\sigma_i^j(q+1)$  dle rovnic (2.65), (2.66), (2.67).
4. **Krok**  $q = q+1$  a opakování kroků 2. a 3. pro , dokud není dosaženo maximálního množství iterací  $q_{max}$  nebo požadované přesnosti  $\epsilon$ .

**5. Krok** Návrat do kroku 2. pro hodnotu  $k = k + 1$ , tedy s novou dvojicí dat  $(\mathbf{x}, d)$ .

Rovnice popisující výstup jednotlivých vrstev fuzzy systému (viz obr. 2.4) následují.

$$z^j = \prod_{i=1}^n \exp \left[ - \left( \frac{x_i(k) - \bar{x}_i^j(q)}{\sigma_i^j(q)} \right) \right] \quad (2.68)$$

$$b = \sum_{j=1}^M z^j \quad (2.69)$$

$$a = \sum_{j=1}^M \bar{b}^j(q) z^j \quad (2.70)$$

$$\hat{y} = \frac{a}{b} \quad (2.71)$$

Pro správnou funkci algoritmu je důležitá prvotní volba hodnot parametrů  $\bar{b}^j(0)$ ,  $\bar{x}_i^j(0)$ ,  $\sigma_i^j(0)$ . Náhodné hodnoty parametrů nejsou v případě použití tohoto algoritmu vhodné. Jedna z možných metod, jak vybrat hodnoty parametrů je využití prvních  $M$  dvojic vstup-výstup. Uvažujme množinu  $M$  dvojic vstup-výstup

$$\{\mathbf{x}(j), d(j)\}, j = 1, 2, \dots, M \quad (2.72)$$

kterou využijeme k počátečnímu nastavení parametrů následujícím způsobem.

$$\bar{b}^j(0) = d(j) \quad (2.73)$$

$$\bar{x}_i^j(0) = x_i(j) \quad (2.74)$$

$$\sigma_i^j(0) = \frac{[max(x_i^l : l = 1, 2, \dots, M)] - min(x_i^l : l = 1, 2, \dots, M)}{M} \quad (2.75)$$

Pro správnou funkci algoritmu je důležitá i volba velikosti kroku  $\mu$ , která se obvykle provádí experimentálně.





## 3 Zobecněné Paretovo rozdělení

#TODO

$$F_u(x) = P(X - u \leq x, X > u) = \frac{F(u+x) - F(u)}{1 - F(u)} \quad (3.1)$$

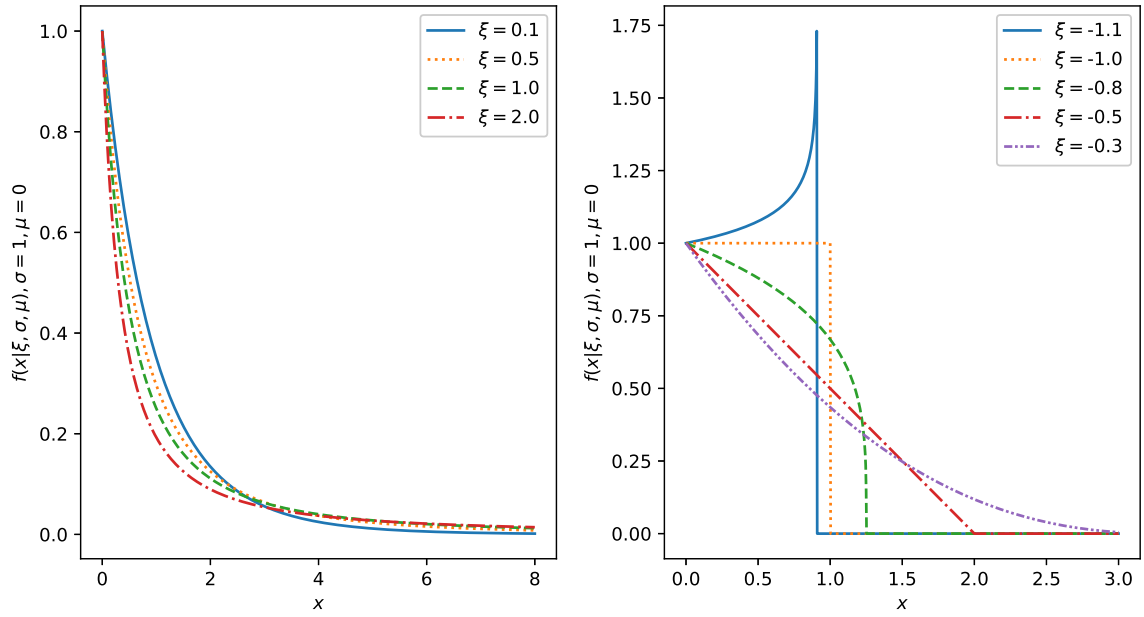
$$F_u(x) = P(X - u \leq x, X > u) = \frac{F(u+x) - F(u)}{1 - F(u)} \quad (3.2)$$

$$f_{(\xi, \mu, \sigma)}(x) = \begin{cases} \frac{1}{\sigma} \left( 1 + \frac{\xi(x-\mu)}{\sigma} \right)^{\left(-\frac{1}{\xi}-1\right)} & \text{for } \xi \neq 0, \\ \exp\left(-\frac{x-\mu}{\sigma}\right) & \text{for } \xi = 0. \end{cases} \quad (3.3)$$

$$F_{(\xi, \mu, \sigma)}(x) = \begin{cases} 1 - \left( 1 + \frac{\xi(x-\mu)}{\sigma} \right)^{-\frac{1}{\xi}} & \text{for } \xi \neq 0, \\ 1 - \exp\left(-\frac{x-\mu}{\sigma}\right) & \text{for } \xi = 0. \end{cases} \quad (3.4)$$

### 3.1 Metoda Peak-over-Threshold

Zásadním problémem při hledání parametrů GPD je volba vhodné hodnoty prahu  $z$ . Metody, které řeší uvedený problém jsou nazývány Peak-Over-Threshold (POT). V případě, že je hodnota prahu  $z$  příliš vysoká, mají parametry GPD velkou varianci, protože je existuje pouze příliš málo dat, které zvolenou prahovou hodnotu překročí. Pokud je hodnota prahu  $z$  naopak příliš nízká, není aproximace ocasu generujícího rozdělení spolehlivá. Z tohoto pohledu, je tedy vhodná volba hodnoty prahu  $z$  pro kvalitu fitu GPD zásadní. Existuje celá řada metod pro výběr hodnoty prahu (více viz [? ]). Některé z metod POT jsou však sami o sobě výpočetně náročné, případně poskytují výsledky, jež musí být vyhodnoceny expertem. Z těchto důvodů nejsou příliš vhodné pro algoritmus detekce novosti, který by vyhodnocoval data v reálném čase. Proto se zdá vhodné využít některé z metod "Rule of Thumb".



Obrázek 3.1: GPD probability density function with various parameters  $\xi$ , and fixed parameters  $\sigma = 1, \mu = 0$

[? ? ?] choice of threshold were used. Let  $l$  is the number of samples used for GPD fitting and  $n_s$  is total number of samples available:

$$l_1 = \lceil 0.1 \cdot n_s \rceil \quad (3.5)$$

$$l_2 = \lceil \sqrt{n_s} \rceil \quad (3.6)$$

$$l_3 = \left\lceil \frac{\sqrt[3]{n_s^2}}{\log(\log(n_s))} \right\rceil \quad (3.7)$$

Note, that we use the highest adaptive weight increments to estimate the GPD parameters. The **Peaks-over-threshold (POT)** method is crucial to decide, whether the  $|\Delta w_k(k)|$  belongs to set  $H_i$  or  $L_i$ . In section ?? there are results with different techniques of threshold choice.

## 3.2 Metody odhadu parametrů zobecněného Paretova rozdělení

#TODO

# 4 Algoritmus Extreme Seeking Entropy

V této kapitole je představen algoritmus pro detekci novosti nazvaný Extreme Seeking Entropy (ESE), který tvoří hlavní výsledek předkládané dizertační práce. Návržený algoritmus, vychází z předpokladu, že novost v datech se projeví neobvykle velkými přírůstky vah adaptivního filtru, který danou řadu dat modeluje. Cílem je tedy vyhodnotit neobvykle velké přírůstky vah adaptivního systém. Je tedy nejprve nutné určit nějaký práh  $z$ , podle kterého můžeme přírůstky adaptivních vah filtru rozdělit do dvou množin. Množinu, která bude obsahovat obsahovat přírůstky menší než je zvolený práh  $z$  označíme  $L$ . Přírůstek který je větší nebo roven hodnotě prahu  $z$  označíme  $H$ . Pozn. Volba hodnoty prahu přímo souvisí s volbou metody POT (viz kapitola 3). Uvažujme, že obě množiny existují pro každou adaptivní váhu, potom pro  $i$ -tou adaptivní váhu zvolíme práh  $z_i$  tak, že velikosti přírůstku této váhy náleží do jedné ze dvou množin, tak, že:

$$\forall |\Delta w_i| < z_i \in L_i \quad (4.1)$$

$$\forall |\Delta w_i| \geq z_i \in H_i \quad (4.2)$$

Vzhledem k výše zmíněnému předpokladu o velikosti změn vah adaptivního filtru a novosti v datech, uvažujme, že přírůstky náležející množině  $L_i$  pravděpodobně neobsahují informaci o novosti během adaptace a proto nebudou vyhodnocovány. Množina  $H_i$  by měla obsahovat přírůstky

Uvažujme, že máme dokonale nastavený adaptivní filtr, jehož chyba predikce  $e$  je nulová pro všechny vstupní hodnoty. Potom přírůstky vah tohoto filtru budou nulové. V případě, že dojde k nějaké změně v generátoru dat pro tento filtr, začne se filtr opět adaptovat což vyústí v nenulové změny adaptivních vah, které budou reflektovat novost způsobenou změnou vlastností daného generátoru.

$$ESE(|\Delta \mathbf{w}(k)|) = -\log \prod_{i=1}^n (1 - f_{cdf_i}(|\Delta w_i(k)|)) \quad (4.3)$$

kde

$$f_{cdf_i}(|\Delta w_i(k)|) = \tag{4.4}$$

## 5 Výsledky experimentů

V této kapitole jsou shrnuty výsledky, kterých bylo v rámci práce na tématu dizertace dosaženo. Jmenovitě jsou to výsledky navrženého algoritmu Extreme Seeking Entropy (viz kapitola 5.1) a potom použití algoritmu Learning Entropy pro adaptivní fuzzy filtr při detekci změn stavu bioprocesu (viz kapitola 5.4). Výsledky byly publikovány v [1].

### 5.1 Případové studie detekce novosti algoritmu Extreme Seeking Entropy

Obsah této kapitoly je shrnutím výsledků publikovaných v MDPI.

#### 5.1.1 Případová studie: chaotická časová řada Mackey-Glass a detekce pertubace

Tento experiment byl proveden pro porovnání s výsledky uvedenými v publikaci [10], která je první publikací o algoritmu LE (viz kapitola 1.1). Experiment spočívá v detekci pertubovaného vzorku v chaotické časové řadě, která je výsledkem řešení Mackey-Glassovy rovnice [20].

$$\frac{dy(t)}{dt} = \beta \cdot \frac{y(t - \tau)}{1 + y^\alpha(t - \tau)} - \gamma y(t) \quad (5.1)$$

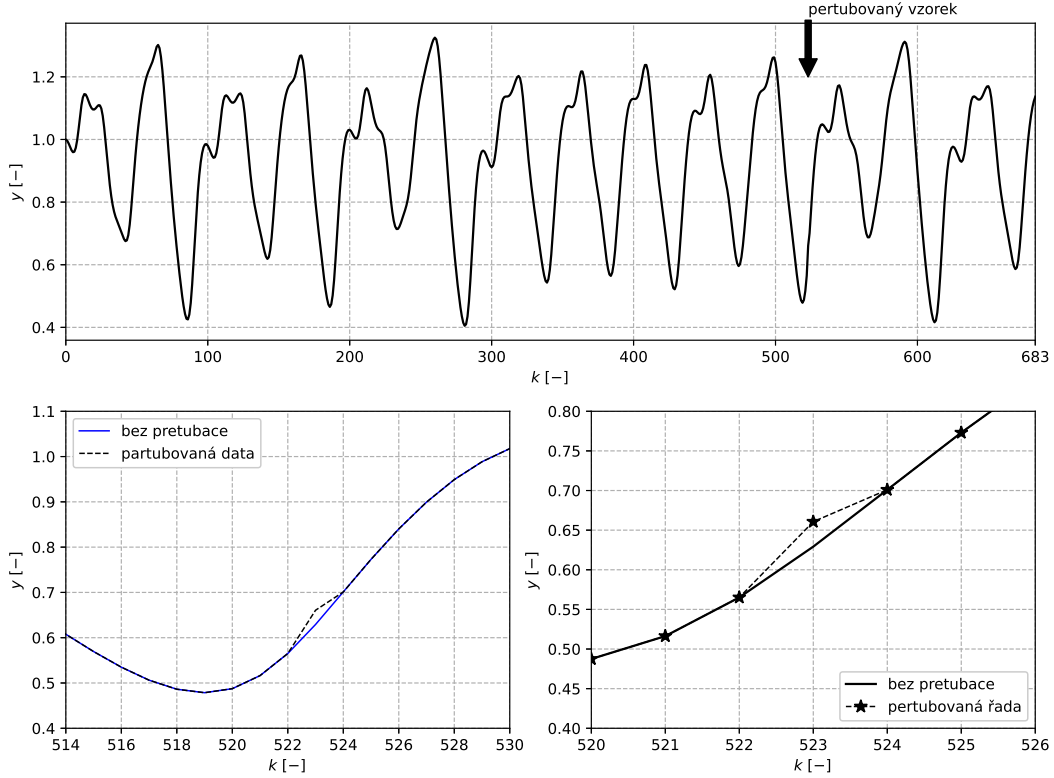
přičemž parametry  $\alpha = 10$ ,  $\beta = 0.2$ ,  $\gamma = 0.1$  and  $\tau = 17$  byly vybrány tak, aby řešením této rovnice byla chaotická časová řada. Celkem bylo vygenerováno 701 vzorků. Data v diskrétní časový okamžiku  $k = 523$  pak byly pertubovány podle následujícího předpisu.

$$y(523) = y(523) + 0.05 \cdot y(523) \quad (5.2)$$

Výsledná časová řada a detail pertubace je znázorněn na obrázku 5.1. Jako adaptivní filtr byl zvolen QNU (resp. Voltterrův filtr) jehož vstupem jsou 4 časově spožděné hodnoty časové řady  $y(k - 1)$ ,  $y(k - 2)$ ,  $y(k - 3)$ ,  $y(k - 4)$ . Hodnoty adaptivních vah tohoto filtru byly adaptovány algoritmem NLMS. Struktura filtru byla zvolena stejně jako v [10]. Rychlost učení během experimentu byla nastavena na  $\mu = 1$ . Metoda POT byla zvolena podle rovnice 3.5 a délka

okna pro detekci novosti algoritmem ESE byla nastavena na  $n_s = 300$ .

Výstup adaptivního filtru a chyba predikce jsou znázorněny na obrázku 5.2. Výsledky metod detekce novosti jsou zobrazeny na obrázku 5.3. Z obrázku je patrné, že globální maximum průběhu *ESE* odpovídá pertubovaným datům. Globální maximum metod *ELBND* a *LE* odpovídá vzorku u něhož byla největší chyba predikce. Hodnoty prahů algoritmu LE byly zvoleny  $\alpha = \{4, 5, 6, 7, 8, 9\}$  a délka okna byla nastavena na  $m = 30$ .



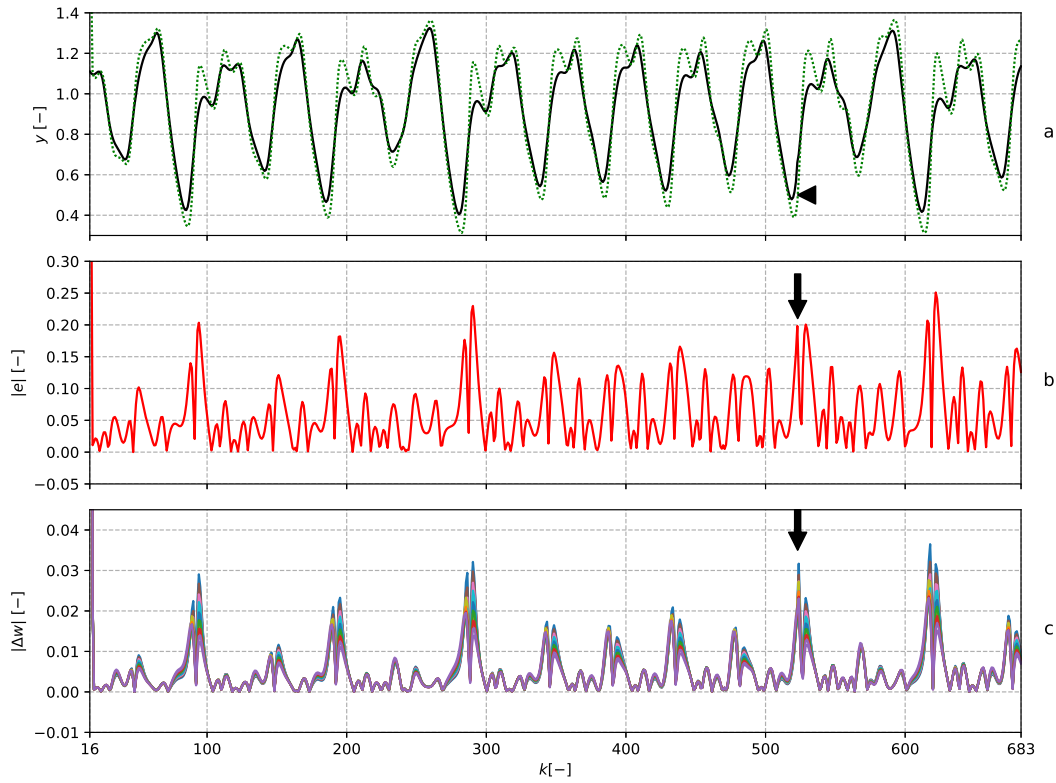
Obrázek 5.1: Horní graf zobrazuje celou datovou řadu. Spodní grafy zobrazují detail pertubovaného vzorku v diskrétní časový okamžik  $k = 523$ .

### 5.1.2 Případová studie: detekce změny rozptylu šumu v náhodném datovém toku

Tato případová studie je navržena na základě problému, který se vyskytuje v použití hybridních navigačních systémů využívajících GPS (Global Positioning System) senzory pro navigaci výpočtem [21]. Smyslem experimentu je demonstrovat možnost využití algoritmu ESE pro detekci změn rozptylu šumu v náhodných datech.

Uvažujme dva vstupy  $x_1(k)$  a  $x_2(k)$  a výstup generátoru signálu  $y(k)$  takový, že

$$y(k) = x_1(k) + x_2(k) + x_1(k) \cdot x_2(k) + v(k) \quad (5.3)$$



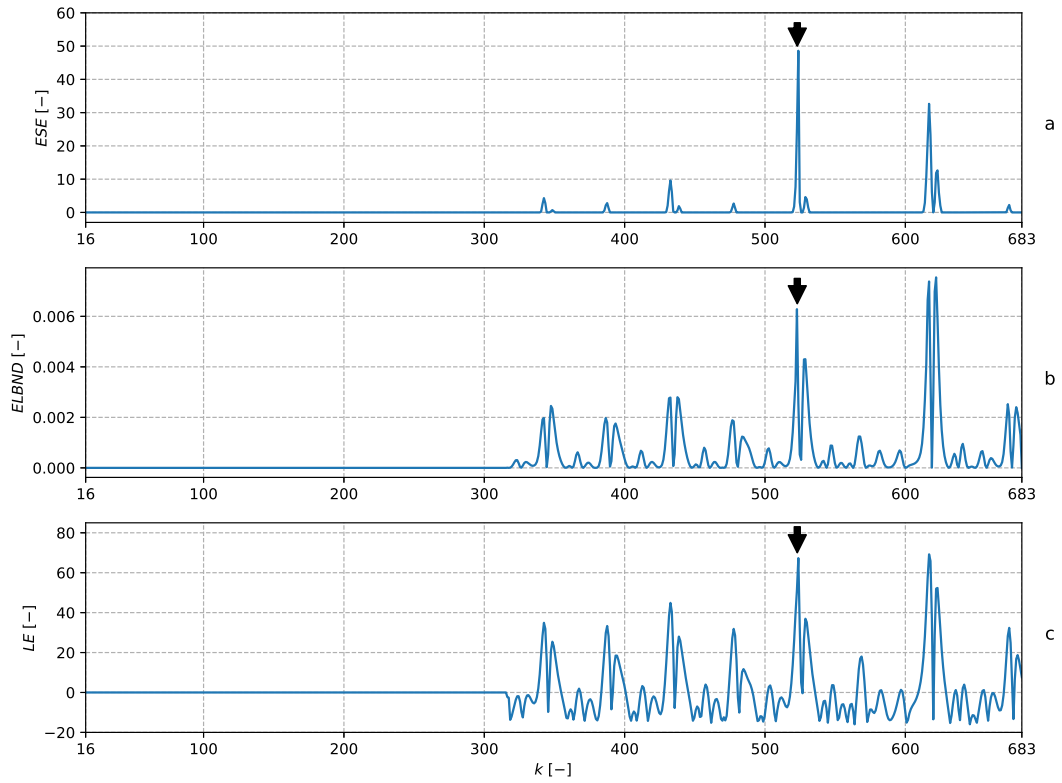
Obrázek 5.2: Graf (a) zobrazuje datovou řadu s pertubací (černá plná čára) a výstup adaptivního filtru (tečkovaná zelená čára). Pertubovaný vzorek je označen černou šipkou. Graf (b) zobrazuje velikost chyby predikce  $e$  (resp. její absolutní hodnotu). Na grafu (c) jsou znázorněny přírůstky adaptivních vah filtru (resp. absolutní hodnotu těchto přírůstků).

kde člen  $v(k)$  reprezentuje aditivní Gaussovský šum který je přidán k výstupu generátoru  $y(k)$ . Přidaný šum má nulovou střední hodnotu a směrodatnou odchylku  $\sigma_n = 0.1$ , takže  $v(k) \sim N(0,1)$ . Hodnoty vstupů jsou v každém diskrétním časovém okamžiku vybrány náhodně z rovnoměrného rozdělení na intervalu  $\langle 0, \rangle$ . V diskrétním časovém okamžiku  $k = 500$  dojde ke změně směrodatné odchylky šumu na hodnotu  $\sigma_n = 0.2$ ,  $v(k) \sim N(0,0.2)$ .

Adaptivní filtr v tomto experimentu byl QNU ve tvaru

$$\hat{y}(k) = w_1 \cdot x_1(k) + w_2 \cdot x_2(k) + w_3 \cdot x_1(k) \cdot x_2(k) \quad (5.4)$$

tak, že jeho struktura odpovídá struktuře generátoru signálu. Adaptivní parametry filtru byly adaptovány algoritmem GNGD. Rychlost učení byla nastavená jako  $\mu = 1$ . Metoda POT byla zvolena podle 3.6 a délka okna  $n_s = 500$ . Výsledky experimentu jsou zobrazeny na obrázku 5.4. Apriorní hodnoty parametrů GPD byly stanoveny na základě 500 vzorků, které nejsou v následujícím obrázku 5.4 zobrazeny. Globální maximum ESE odpovídá změně směrodatné odchylky šumu  $\sigma_n$ . Detekce pomocí algoritmů LE a ELBND je o několik vzorků opožděná.



Obrázek 5.3: Graf (a) zobrazuje hodnotu ESE. Prvních 300 vzorků je hodnota ESE nulová, protože délka okna pro vyhodnocování novosti  $n_s = 300$ . Graf (b) zobrazuje výsledky algoritmu ELBND. Prvních 300 výsledků ELBND je pro názornost vynecháno. Graf (c) zobrazuje výsledky algoritmu LE.

Pro výpočet LE byl použit vztah 1.7 a délka okna byla nastavena na  $M = 300$ .

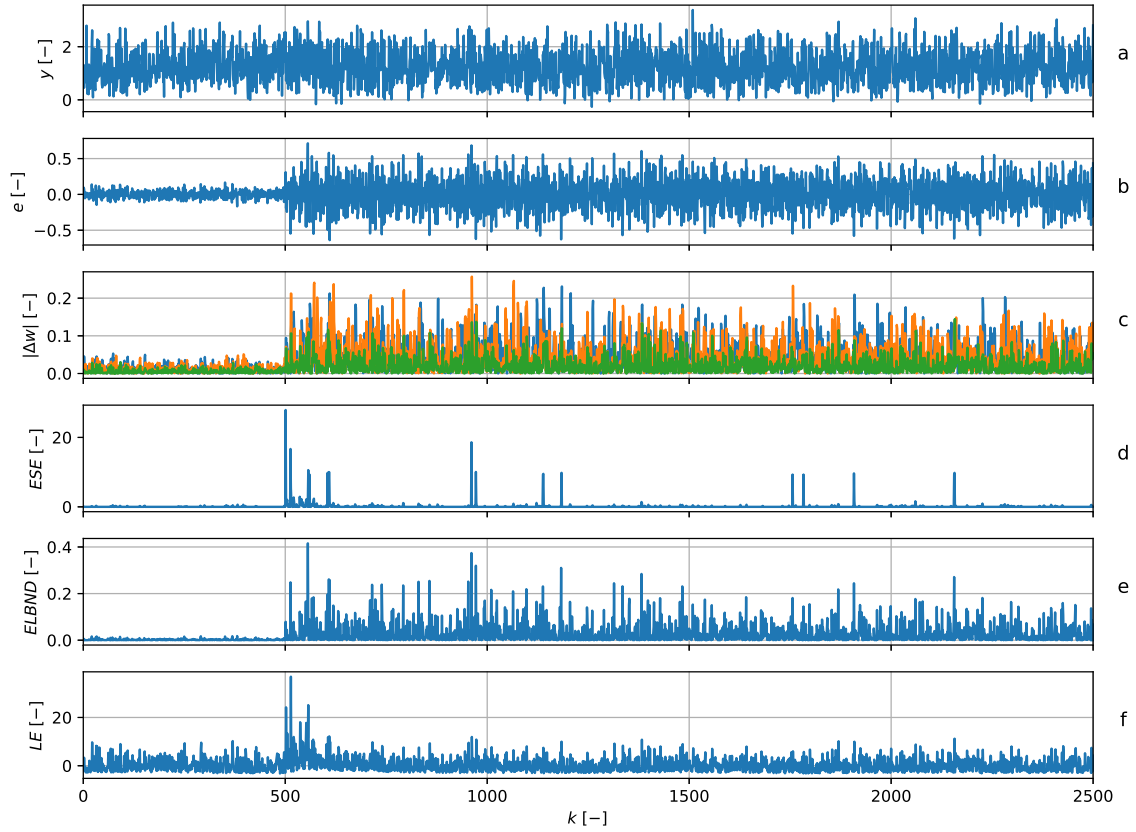
### 5.1.3 Případová studie: detekce skokové změny parametrů generátoru signálu

#TODO

### 5.1.4 Případová studie: detekce náhlé absence šumu

In this experiment it is shown, that the slightly reformulated algorithm can detect also deal with immediate decrease of learning effort. Assume, that instead of unusually high learning effort, we want to focus on unusually low learning effort. The only change in the proposed algorithm is, that we will use POT method to get  $l$  the smallest weight updates and based on those updates the parameters of GPDs are estimated. The scheme of this experiment is similar to previous. Assuming two inputs  $x_1(k)$  and  $x_2(k)$  and output  $y(k)$  that are related by the equation (??). But in this case, at discrete time index  $k = 500$ , the noise is removed,





Obrázek 5.4: Na grafu (a) jsou zobrazeny data z generátoru (modrá) a výstup z adaptivního filtru (zelená). Na grafu (b) je vynesena chyba filtru  $e(k)$ . Graf (c) zobrazuje velikosti přírůstků adaptivních parametrů filtru. Na grafu (d) jsou zobrazeny hodnoty ESE. V diskretní časový okamžik  $k = 500$  je patrný značný nárůst v ESE, který reflektuje změnu směrodatné odchylky aditivního šumu. Na dalších grafech (e) a (f) jsou zobrazeny výsledné hodnoty algoritmů ELBND a LE.

so the equation (??) for  $k \geq 500$  takes the form

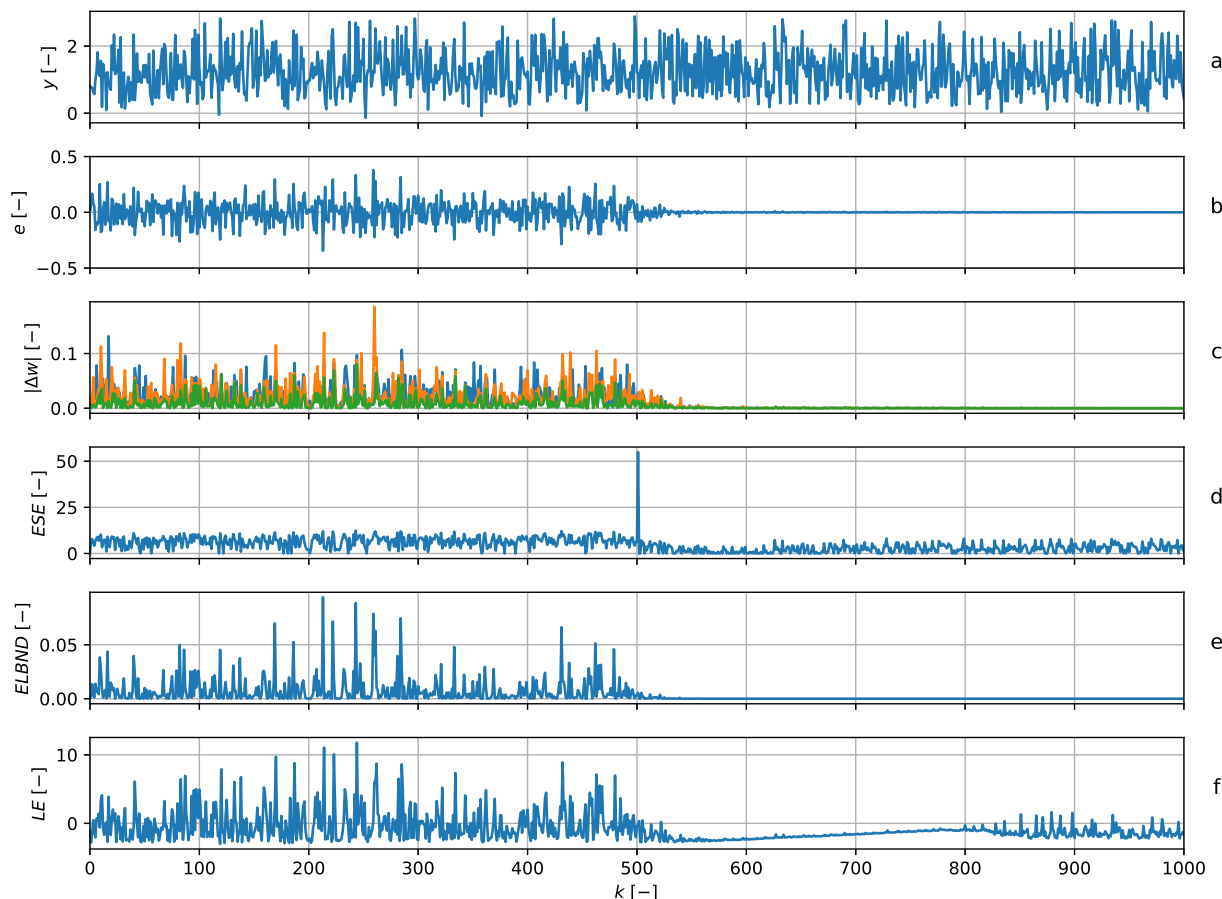
$$y(k) = x_1(k) + x_2(k) + x_1(k)x_2(k). \quad (5.5)$$

The QNU was chosen for obtained data processing. The number of inputs to QNU was set to  $n = 2$  so the inputs are

$$\mathbf{x} = [x_1(k-1), x_2(k-1)] \quad (5.6)$$

so the adaptive filter has totally 3 adaptive weights. **The structure of the adaptive filter was chosen to correspond with the structure of the signal generator.** The parameters are updated with every new obtained sample with the GNGD algorithm. The learning rate during the experiment was constantly set to  $\mu = 1$ . The POT method was chosen according to (3.5) with  $n_s = 500$ . The following ?? shows, that the peak in  $ESE$  corresponds with the noise

disappearance. The *LE* and *ELBND* methods failed to detect the noise disappearance. For *ELBND* this results can be expected as the values of the *ELBND* are high for high prediction error and high adaptive weight increments.



Obrázek 5.5: Noise disappearance detection. The graph (a) shows the data series (blue) and the output of the predictor (green). The graph (b) shows the value of predictors error. The graph (c) shows the absolute value of adaptable parameters increments. The graph (d) shows the *ESE* novelty score. For discrete time index  $k \geq 500$  the noise is removed from the signal which corresponds with peak in *ESE*. Graph (e) and (f) contains results of *ELBND* and *LE* methods.

### 5.1.5 Případová studie: detekce změny trendu

#TODO

### 5.1.6 Případová studie: detekce epilepsie v myším EEG

#TODO

### 5.1.7 Vyhodnocení úspěšnosti detekce skokové změny parametrů generátoru signálu

#TODO

## 5.2 Vyhodnocení úspěšnosti detekce změny trendu a evaluace ROC křivky

#TODO

## 5.3 Vyhodnocení výpočetní náročnosti metod odhadu parametrů zobecněného Paretova rozdělení

Výsledky v této podkapitole byli publikovány v (můj shit). Cílem bylo určit výpočetní čas výpočtu parametrů GPD v typické aplikaci pro použití algoritmu ESE, který byl, v tomto případě, testován ne experimentu detekce skokové změny parametrů generátoru signálu.

### 5.3.1 Motivace

Detekce novosti v reálném čase je úloha, která nalézá své uplatnění nejen v oblasti detekci a diagnostiky v průmyslových aplikacích [14], ale také např. v detekci narušení počítačových sítí [15] nebo v zabezpečovacích systémech [16]. Další oblastí uplatnění je např. mobilní robotika, která je specifická tím, že robot má k dispozici pouze limitovaný výpočetní výkon [17, 18]. Pro metody detekce novosti v reálném čase je tedy důležité, aby vynikali dostatečně nízkou výpočetní náročností. Z tohoto důvodu byli otestovány tři různé metody výpočtu parametrů GPD (viz kapitola 3, protože tento výpočet je z hlediska použití algoritmu ESE potenciálně limitující z hlediska využitelnosti v aplikacích detekce v reálném čase. Jmenovitě byli otestovány tyto metody: metoda maximální věrohodnosti (ML), metoda momentů (MOM) a metoda kvazi-maximální věrohodnosti (QML) (více viz kapitola 3). Výpočetní čas potřebný k určení parametrů GPD pomocí těchto metod byl vyhodnocen při experimentu, ve kterém dojde ke skokové změně parametrů generátoru signálu.

### 5.3.2 Specifikace experimentu

Vzhledem k povaze experimentu, který slouží k vyhodnocení výpočetní náročnosti různých metod určení parametrů GPD, a nikoliv k detekci novosti v nějakém komplexním procesu, byl zvolen jednoduchý lineární kombinační filtr (LNU), jehož výstup v diskrétním časovém okamžiku  $k$  je definován jako

$$\hat{y}(k) = w_1 \cdot x_1(k) + w_2 \cdot x_2(k) + w_3 \cdot x_3(k) \quad (5.7)$$

a tento filtr je adaptován algoritmem NLMS (viz kapitola 2.2.1), přičemž rychlost učení  $\mu$  byla nastavena jako  $\mu = 0.8$ .

Pro výstup generátoru signálu platí vztah

$$y(k) = x_1(k) + x_2(k) + x_3(k) + v(k) \quad (5.8)$$

pro všechny  $1 \leq k \leq 200$ . Člen  $v(k)$  reprezentuje aditivní gaussovský šum s nulovou střední hodnotou a směrodatnou odchylkou  $\sigma_{noise} = 0.1$ . V diskrétním časovém okamžiku  $k = 201$  dojde ke změně generátoru signálu a jeho výstup přejde do tvaru

$$y(k) = 0.7 \cdot x_1(k) + 1.2 \cdot x_1(k) + 1.1 \cdot x_1(k) + v(k) \quad (5.9)$$

pro  $201 \leq k \leq 400$ . Hodnota všech vstupů generátoru signálu je v každém časovém okamžiku  $k$  vybrána ze standardního rozdělení normálního rozdělení, takže  $i$ -tý vstup  $x_i \sim \mathcal{N}(0, 1)$ . Změna parametrů signálu byla vybrána tak, aby nedošlo ke změně střední hodnoty signálu  $y(k)$ .

Délka okna pro odhad parametrů GPD byl během experimentu nastavena na  $n_s = 1200$ . Metoda POT byla zvolena podle REF na 10%. Před experimentem bylo pořízení 1200 vzorků vygenerovaných generátorem signálu definovaným vztahem 5.5, na něž byla použita metoda POT, tak aby při experimentu v diskrétní časový okamžik  $k = 1$  byla hodnota ESE relevantní.

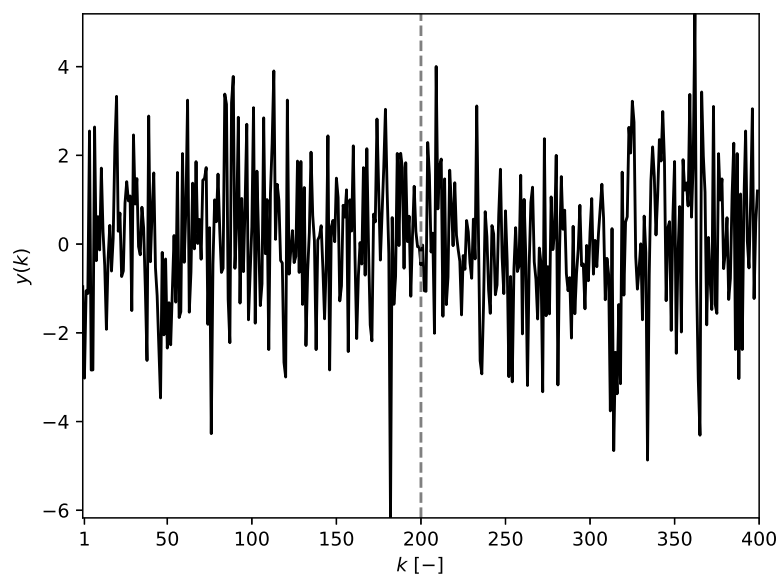
Experiment byl proveden na PC s procesorem Intel(R) Core(TM) i5-7400 se 4mi jádry s taktovací frekvencí 3001 MHz a operační paměť o velikosti 32 GB. Operační systém byl Windows 10 Pro, 64-bitová verze 10.0.18362. Kód byl napsán v Python 3.6.1 a byly použity knihovny Numpy 1.17.0 a Scipy 1.4.1. Pro

### 5.3.3 Výsledky a diskuze

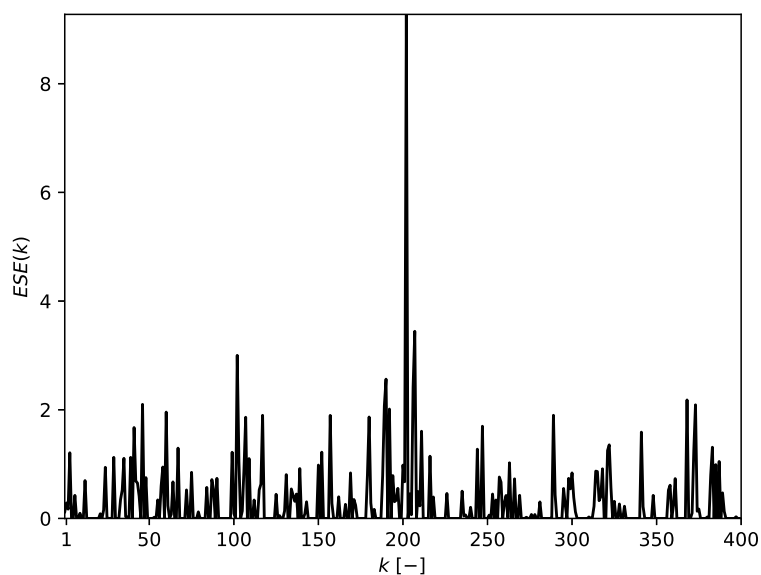
Průměrný čas výpočtu  $\bar{t}$  parametrů všech tří GPD (počet GPD odpovídá počtu adaptivních parametrů filtru) a odpovídající směrodatné odchylky  $\sigma_t$  jsou uvedeny v následující tabulce 5.1. Čas výpočtu je určený pro jeden experiment (400 vzorků).

Z uvedených výsledků je patrné, že nejrychlejší metoda je MOM. Podstatnou nevýhodou této metody pro využití v aplikacích, které vyhodnocují data v reálném čase je její omezení na hodnoty parametrů GPD (viz kapitola 3). Pokud parametry uvedené omezení nesplňují, vypočtené hodnoty nepřesné (resp. nesmyslné) a tedy nepoužitelné pro algoritmus ESE, který začne produkovat nepřesné výsledky. Z pohledu úlohy detekce novosti je diskutabilní, zda-li můžeme garantovat, že sledovaný proces po celou dobu bude splňovat uvedené omezení.

Metoda, jejíž výpočetní čas byl nejvyšší je ML, což je vzhledem k iterativnímu určení parametrů GPD očekávatelné. Nevýhoda použití této metody tkví v nemožnosti určit minimální resp. maximální počet iterací. Jednou z možností jak zrychlit nalezení parametrů je využití

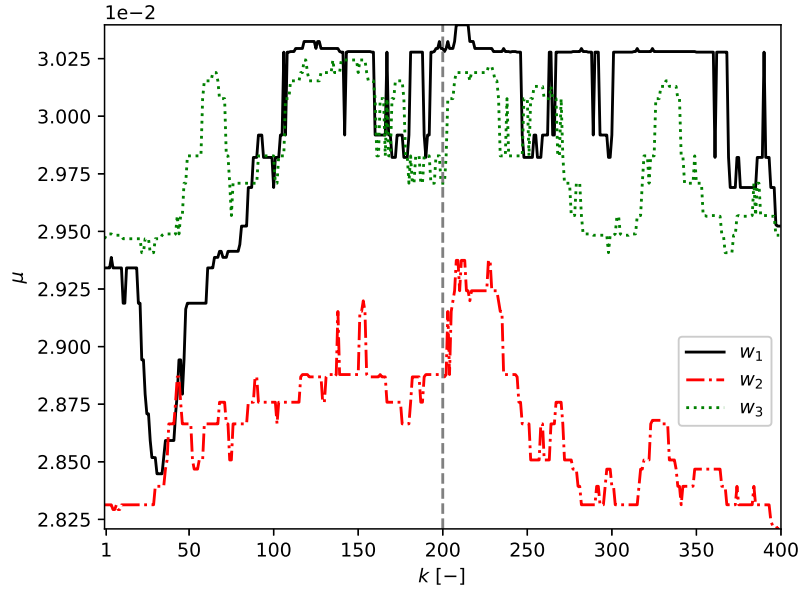


Obrázek 5.6: Výstup adaptivního filtru během experimentu. Skoková změna parametrů generátoru signálu je zvýrazněná svislou vodorovnou čarou v diskrétním časovém okamžiku  $k = 200$ .

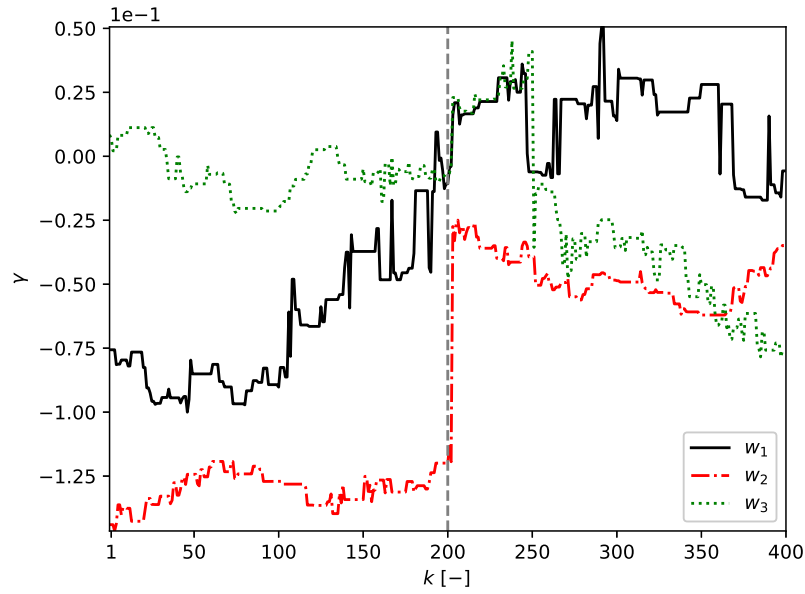


Obrázek 5.7: Hodnota ESE během experimentu. Globální maximum odpovídá změně parametrů generátoru signálu, resp. úspěšné detekci novosti.

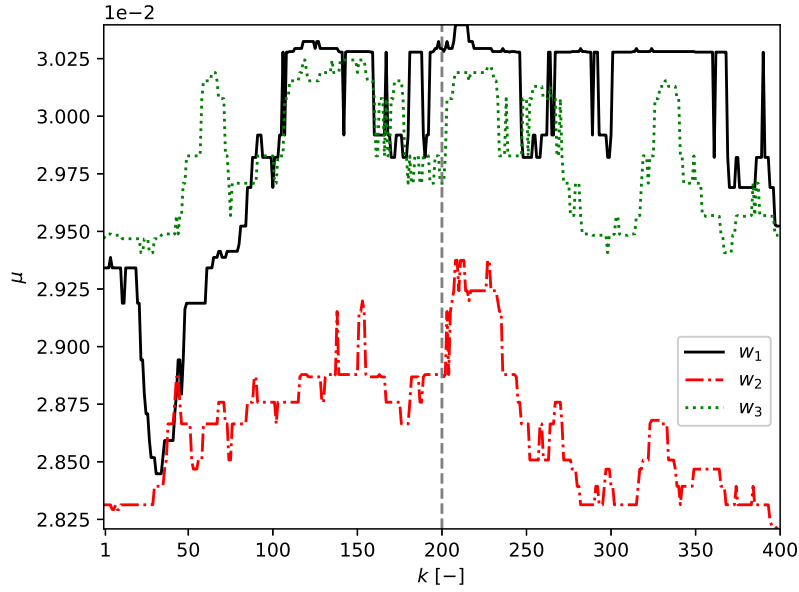
apriorní informace o hodnotách těchto parametrů. V rámci experimentu však byla využita pouze apriorní informace o parametru  $\mu$ , který odpovídá nejmenší hodnotě přírůstku vah,



Obrázek 5.8: Hodnota parametru  $\mu$  GPD pro všechny tři adaptivní váhy  $w_1, w_2, w_3$  během experimentu detekce změn parametrů generátoru signálu. Svislá čára v diskretním časovém okamžiku  $k = 200$  znázorňuje skokovou změnu parametrů generátoru signálu.



Obrázek 5.9: Hodnota parametru  $\gamma$  GPD pro všechny tři adaptivní váhy  $w_1, w_2, w_3$  během experimentu detekce změn parametrů generátoru signálu. Svislá čára v diskretním časovém okamžiku  $k = 200$  znázorňuje skokovou změnu parametrů generátoru signálu.



Obrázek 5.10: Hodnota parametru  $\sigma$  GPD pro všechny tři adaptivní váhy  $w_1$ ,  $w_2$ ,  $w_3$  během experimentu detekce změn parametrů generátoru signálu. Svislá čára v diskretním časovém okamžiku  $k = 200$  znázorňuje skokovou změnu parametrů generátoru signálu.

Tabulka 5.1: Tabulka průměrných časů výpočtu pro jednotlivé adaptivní váhy a odpovídajících směrodatných odchylek vybraných metod výpočtu parametrů GPD

	Metoda	$\bar{t}$ [ms]	$\sigma_t$ [ms]
$w_1$	ML	26.198	3.396
	QML	0.354	0.478
	MOM	<b>0.076</b>	0.264
$w_2$	ML	26.718	2.302
	QML	0.337	0.471
	MOM	<b>0.064</b>	0.244
$w_3$	ML	24.982	1.964
	QML	0.395	0.489
	MOM	<b>0.060</b>	0.238

které byli získány metodou POT aplikovanou na plovoucí okno délky  $n_s$ .

Dobrým kompromisem mezi výše uvedenými metodami je použití metody QML. Výpočetní čas této metody byl v uvedeném experimentu o dva řády kratší než ML a asi pětikrát delší než MOM. Pro každou z vyhodnocovaných vah byl kratší než  $500 \mu s$ .

Z provedeného experimentu je patrné, že použití algoritmu ESE pro aplikace v reálném čase je limitováno počtem parametrů filtru a rychlostí vzorkování monitorovaného procesu.

## 5.4 Případová studie použití algoritmu Learning Entropy a adaptivního fuzzy filtru pro detekci změn stavů bioproduktu

Cílem této studie je ověřit, že algoritmus LE, který byl doposud publikován s adaptivními filtry typu LNU a HONU, je možné použít i pro jiné typy adaptivních filtrů. Pro provedenou studii tak byl vybrán adaptivní fuzzy filtr (viz kapitola 2.1.4).

### 5.4.1 Popis bioproduktu a specifikace problému

Podle [1] je pro fermentační procesy, které probíhají v dávkovém režimu je podstatné, aby probíhalo správně dávkované živění substrátem. Pro tyto procesy je specifické, že se při nadměrných koncentracích stává substrát pro mikroorganismy toxickým a může dojít k tzv. přeživení a tím i zahubení těchto mikroorganismů. Naopak, v důsledku nedostatečného zábovení živinami může dojít k odumření kultivovaného organismu. Z tohoto pohledu se je tedy důležité v závislosti na koncentraci substrátu a stavu populace mikroorganismů měnit i strategii pro řízení procesu kultivace. Historicky byl stav bioproduktu klasifikován expertem, přičemž vyhodnocení bylo poměrně časově náročné a nebylo neobvyklé, že různí experti docházeli k rozdílným závěrům. Protože výnos fermentačního procesu je zásadním způsobem ovlivněn správnou klasifikací stavu ve kterém se právě nachází, bylo by vhodné klasifikaci automatizovat a pokud možno zvýšit její přesnost. Tomuto problému je právě věnována publikace [1], která řeší problém automatické klasifikace stavů bioproduktu kultivace bakterie *Pseudomonas putida* KT2442. V této publikaci je navržen komplexní algoritmus pro online klasifikaci stavů bioproduktu. Autoři zde rozlišují celkem tři stavy bioproduktu kultivace *Pseudomonas putida* KT2442, konkrétně:

1. normální živění
2. přeživení
3. nedoživení

Navržený algoritmus vyhodnocuje přísun vstupujících živin ( $F_m$ ), respektive substrátu, a změny a trendu rozpuštěného kyslíku ( $DO$ ), který je produkován bakteriemi *Pseudomonas putida* a pomocí hřebenové regrese (v literatuře se vyskytuje také pod názvem Tichonova regularizace) je určován vývoj populace bakterií respektive stav bioproduktu. Vzhledem k tomu, že modely vývoje populace pro jednotlivé stavy jsou různé, mohlo by v průběhu experimentu dojít i k podstatným změnám v adaptivním modelu v okamžicích změn stavů kultivace. Tyto změny se mohli projevit neobvykle velkými přírůstky adaptivních parametrů.

Protože algoritmus Learning Entropy využívá přírůstek adaptivních parametrů, mohl by být vhodným nástrojem pro detekci změn stavů bioproduktu. Předpokládáme, že tedy existuje souvislost mezi změnami stavu bioproduktu a nárůstem Learning Entropy. Přestože může být



proces kultivace bakterií *Pseudomonas Putidas* modelován různými a různě složitými modely, pro využití algoritmu Learning Entropy se jeví výhodné použít jednoduché prediktory nebo sledovače. Dosud publikované články využívali pro algoritmus LE pouze FIR filtry, případně Volterrovy filtry, které mají adaptivní parametry v lineární závislosti. V tomto experimentu je použit adaptivní fuzzy filtr, jehož struktura je specifikována v kapitole 2.1.4 a k jehož adaptaci byl použit algoritmus, který je uveden v kapitole 2.2.4.

Použitý adaptivní fuzzy filtr má 9 pravidel ( $M = 9$ ), jehož  $l$ -té pravidlo je ve tvaru

$$IF\ do(k-1)\ is\ A_1^l\ AND\ do(k-6)\ is\ A_2^l\ AND\ do(k-19)\ is\ A_3^l\ THEN\ do(k)\ is\ B^l\quad (5.10)$$

kde  $A_i^l$  je množina ve vstupním prostoru  $U \subset R^3$  a  $B^l$  je fuzzy množina ve výstupním prostoru  $V \subset R$ . V uvedeném pravidle jsou  $do(k-1)$ ,  $do(k-6)$ ,  $do(k-19)$  a  $do(k)$  lingvistické proměnné, které vyjadřují koncentraci rozpuštěného kyslíku  $do$  v diskretních časových okamžicích  $k$ ,  $k-1$ ,  $k-6$  respektive  $k-19$ , kde  $k$  je diskretní časový index. Vzhledem k tomu, že uvedený adaptivní fuzzy filtr používá Gaussovske funkce příslušnosti (viz kapitola 2.1.4 je zobrazení popisující jeho výstup ve tvaru

$$\hat{y}(\mathbf{x}(k)) = \frac{\sum_{j=1}^9 \bar{b}^j \left[ \prod_{i=1}^3 \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right) \right]}{\sum_{j=1}^9 \left[ \prod_{i=1}^3 \exp\left(-\left(\frac{x_i - \bar{x}_i^j}{\sigma_i^j}\right)^2\right) \right]} \quad (5.11)$$

kde vektor  $\mathbf{x}(k)$  je

$$\mathbf{x}(k) = [do(k-1), do(k-6), do(k-19)]. \quad (5.12)$$

Protože hodnota koncentrace rozpuštěného kyslíku je uváděna v procentech, platí pro všechna  $x_i \in \langle 0; 100 \rangle$ . K adaptování výše uvedeného filtru byl použit algoritmus gradient descent (viz kapitola 2.2.4. Maximální počet epoch byl stanoven na  $q_{max} = 100$  a požadovaná chyba predikce mezi výstupem adaptivního filtru a naměřenými daty na  $\epsilon = 0,001$ . Rychlost učení byla během experimentů nastavena na  $\mu = 1$ .

Pro vyhodnocení novosti byla použita přímá verze algoritmu LE, takže

$$E(k) = \max\{0, \sum_{j=1}^9 z(|\Delta \bar{b}^j|) - \beta\} \quad (5.13)$$

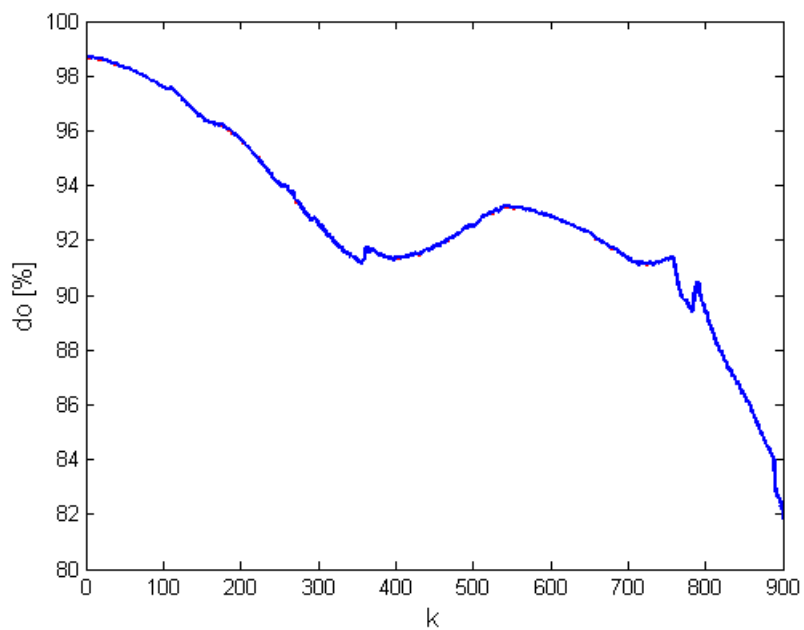
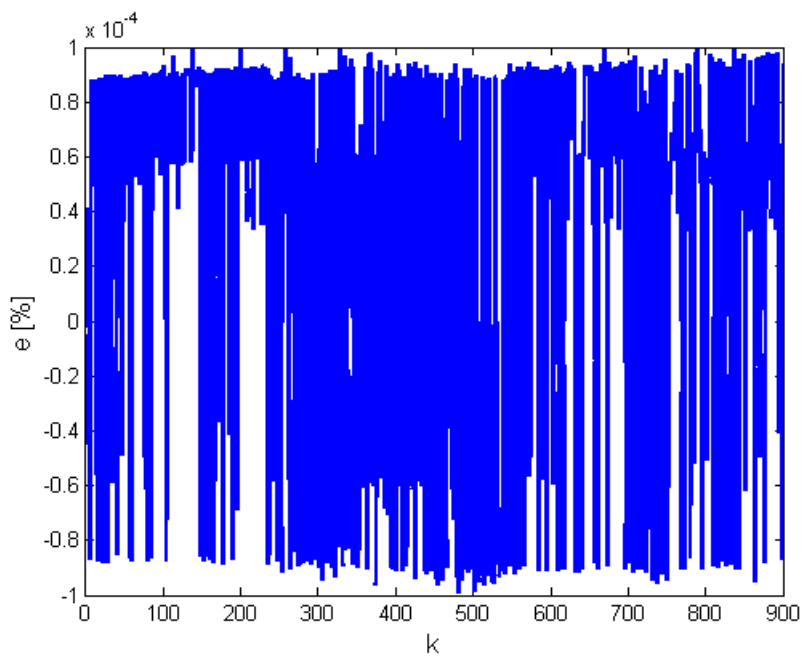
kde funkce  $z$  je dána rovnicí (XY) a  $\beta$  je citlivostní parametr. Pro vyhodnocení novosti jsou tedy použity změny polohy středů množin ve výstupním prostoru, nikoliv změny parametrů fuzzy množin ve vstupním prostoru.

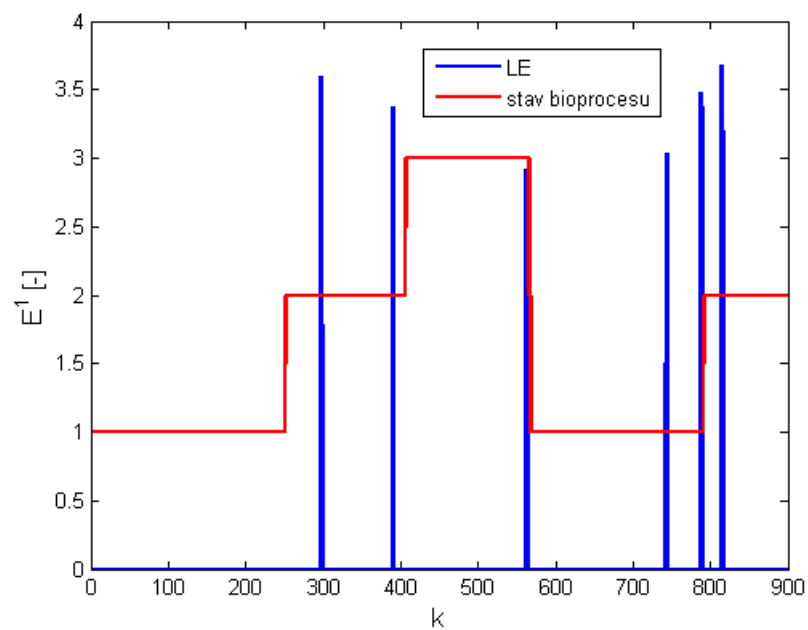
### 5.4.2 Experiment a zhodnocení

Experiment s použitím algoritmu LE a adaptivního fuzzy filtru byl uskutečněn na datech z kultivace bakterie *Pseudomonas Putida*, který byl uskutečněn na Ústavu počítačové a řídicí techniky VŠCHT Praha. Celkem byly zpracovávány hodnoty ze dvou kultivací. Přestože bylo během experimentu měřena sada různých veličin (např. teplota, pH, atd.), osvědčil se pro použití LE signál rozpustěného kyslíku  $do$  [%]. Během experimentu bylo použito vzorkování  $T = 1 \text{ min}$ , což je vzhledem k rychlosti celého procesu dostatečně rychlé vzorkování. Počáteční nastavení parametrů adaptivního fuzzy filtru bylo provedeno tak, jak je popsáno v kapitole 2.2.4. Podstatný vliv na výsledek detekce změn stavu bioprocesu měla volba délky okna pro vyhodnocení změn adaptabilních parametrů filtru  $M_{ND}$ . Na základě experimentů s různými délkami byla nakonec zvolena délka okna  $M_{ND} = 20$ .

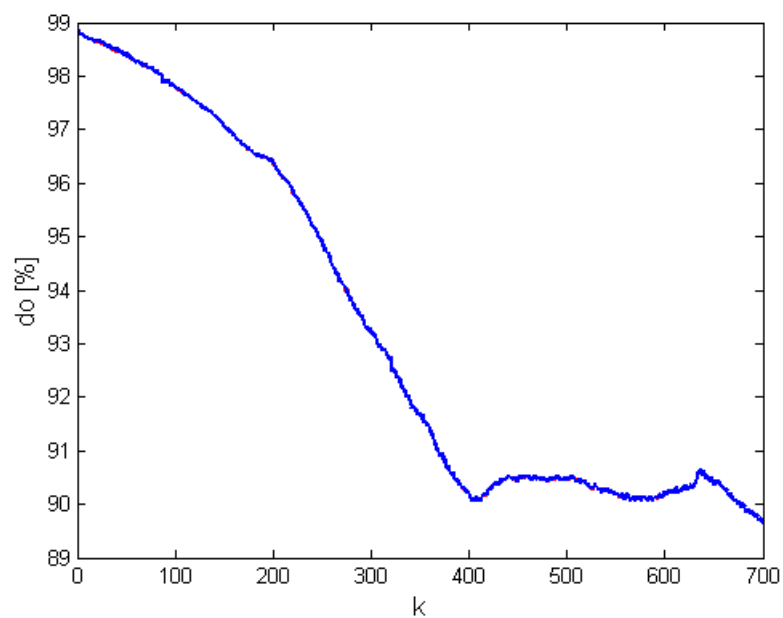
Na následujících obrázcích je znázorněn průběh signálu  $do$  během první kultivace (viz obrázek 5.10), chyba predikce (viz obrázek 5.11) a odpovídající hodnoty  $LE$  společně se stavy bioprocesu (viz obrázek 5.14). Význam stavů bioprocesu znázorněných na obrázku 5.11 respektive obrázku 5.15 jsou: 1 - nedoživení, 2 - živení, 3 - přeživení. Aby byli detekovány všechny změny stavu bioprocesů, byla stanovena hodnota parametru  $\beta = 2,58$  (viz rovnice 5.11).

Data z druhé kultivace byla použita k ověření správného nastavení parametru  $\beta$ . Obrázek 5.13 zobrazuje průběh signálu  $do$  během druhého experimentu. Chyba predikce adaptivního fuzzy filtru je zobrazena na obrázku 5.12. Obrázek 5.15 zobrazuje stavy bioprocesu během kultivace a odpovídající hodnoty  $LE$ . S tímto nastavením parametru  $\beta$  se podařilo detekovat pouze tři změny bioprocesu. Nicméně pro jiné hodnoty délky okna  $M_{ND}$  a hodnoty parametru  $\beta$  se tyto změny detekovat podařilo. Je tedy zřejmé, že pro praktické použití je správná volba obou parametrů zásadní. Vzhledem k malému množství dat a časové náročnosti kultivace nebylo možné použít nějakou validační metodu.

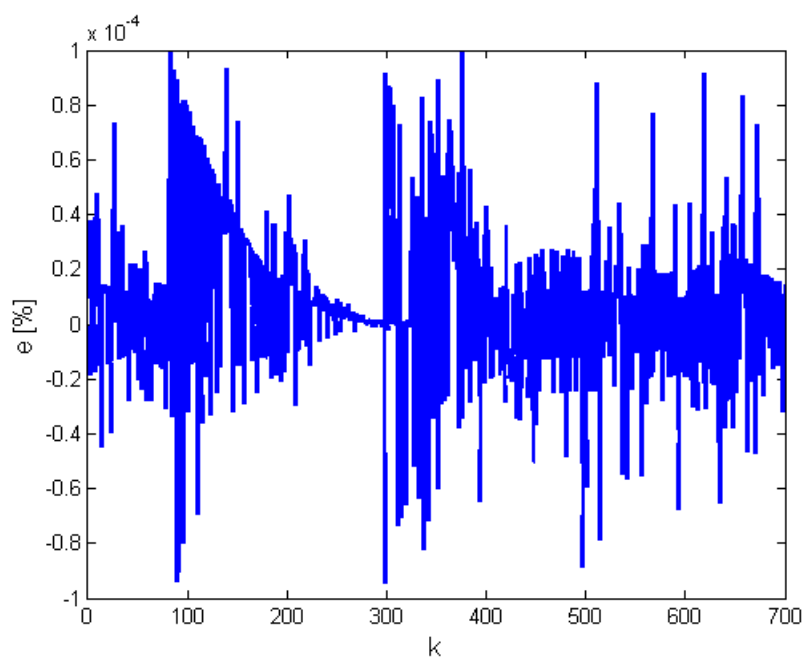
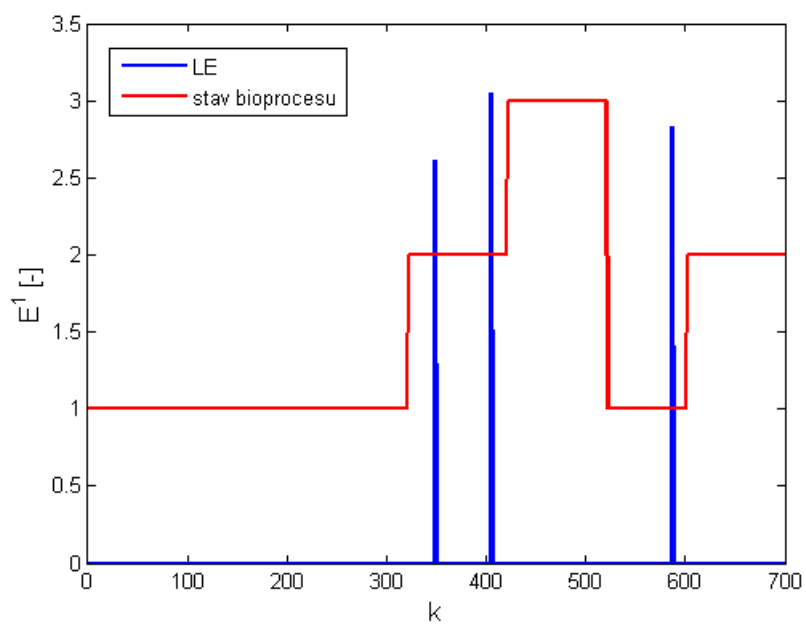
Obrázek 5.11: Průběh signálu  $do$  během první kultivaceObrázek 5.12: Chyba predikce  $e$  během první kultivace



Obrázek 5.13: Stav bioprocesu a hodnota  $LE$  první kultivace



Obrázek 5.14: Průběh signálu  $do$  během druhé kultivace

Obrázek 5.15: Chyba predikce  $e$  během druhé kultivaceObrázek 5.16: Stav bioprocesu a hodnota  $LE$  během druhé kultivace



## 6 Závěr

#TODO





# Publikace autora

- [1] VRBA, Jan; MAREŠ, Jan. Introduction to Extreme Seeking Entropy. Entropy, 2020, 22.1: 93.



# Literatura

- [1] MAREŠ, Jan, et al. Process state classification of fed-batch fermentation based on process variables analysis. *Biochemical Engineering Journal*, 2016, 112: 178-185.
- [2] ROWELL, D. 2.161 Signal Processing: Continuous and Discrete, Fall 2008. 2008.
- [3] MANDIC, Danilo P.; KANNA, Sithan; CONSTANTINIDES, Anthony G. On the intrinsic relationship between the least mean square and Kalman filters [Lecture Notes]. *IEEE Signal Processing Magazine*, 2015, 32.6: 117-122.
- [4] PROAKIS, John G. a Dimitris G. MANOLAKIS. Digital signal processing: principles, algorithms, and applications. 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 1996. ISBN 0133737624.
- [5] OOSTENVELD, Robert; PRAAMSTRA, Peter. The five percent electrode system for high-resolution EEG and ERP measurements. *Clinical neurophysiology*, 2001, 112.4: 713-719.
- [6] HAYKIN, Simon S. Adaptive filter theory. Fifth edition. vyd. Upper Saddle River, New Jersey: Pearson, 2014. ISBN 978-0-13-267145-3.
- [7] STRICHARTZ, Robert S. A guide to distribution theory and Fourier transforms. World Scientific Publishing Company, 2003.
- [8] GERSHO, Allen; GRAY, Robert M. Vector quantization and signal compression. Springer Science & Business Media, 2012.
- [9] BUKOVSKY, Ivo; KINSNER, Witold; HOMMA, Noriyasu. Learning Entropy as a Learning-Based Information Concept. *Entropy*, 2019, 21.2: 166.
- [10] BUKOVSKY, Ivo. Learning entropy: Multiscale measure for incremental learning. *Entropy*, 2013, 15.10: 4159-4187.
- [11] CEJNEK, Matous; BUKOVSKY, Ivo. Concept drift robust adaptive novelty detection for data streams. *Neurocomputing*, 2018, 309: 46-53.

- [12] CEJNEK, Matous; BUKOVSKY, Ivo. Influence of type and level of noise on the performance of an adaptive novelty detector. In: 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI\* CC). IEEE, 2017. p. 373-377.
- [13] CEJNEK, Matous; BUKOVSKY, Ivo; VYSATA, Oldrich. Adaptive classification of EEG for dementia diagnosis. In: 2015 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM). IEEE, 2015. p. 1-5.
- [14] GERTLER, Janos. Fault detection and diagnosis in engineering systems. CRC press, 1998.
- [15] YU, Kangqing, et al. Real-time Outlier Detection over Streaming Data. In: 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2019. p. 125-132.
- [16] RAMEZANI, Ramin; ANGELOV, Plamen; ZHOU, Xiaowei. A fast approach to novelty detection in video streams using recursive density estimation. In: 2008 4th International IEEE Conference Intelligent Systems. IEEE, 2008. p. 14-2-14-7.
- [17] MARS LAND, Stephen; NEHMZOW, Ulrich; SHAPIRO, Jonathan. On-line novelty detection for autonomous mobile robots. *Robotics and Autonomous Systems*, 2005, 51.2-3: 191-206.
- [18] NEHMZOW, Ulrich, et al. Novelty detection as an intrinsic motivation for cumulative learning robots. In: *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, Berlin, Heidelberg, 2013. p. 185-207.
- [19] SCARROTT, Carl; MACDONALD, Anna. A review of extreme value threshold estimation and uncertainty quantification. *REVSTAT–Statistical Journal*, 2012, 10.1: 33-60.
- [20] MACKEY, Michael C.; GLASS, Leon. Oscillation and chaos in physiological control systems. *Science*, 1977, 197.4300: 287-289.
- [21] SPANGENBERG, Mariana, et al. Detection of variance changes and mean value jumps in measurement noise for multipath mitigation in urban navigation. *Navigation*, 2010, 57.1: 35-52.

# Příloha