

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

# **Metody zpracování ERP signálů**



Zadání

# Prohlášení

---

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 17. 5. 2012

Pavel Petrman

## **Poděkování**

---

Rád bych na tomto místě poděkoval především vedoucímu práce Ing. Pavlu Mautnerovi, PhD., za jeho vstřícnost a pomoc nad obvyklý rámec. Dále děkuji svému zaměstnavateli, společnosti Ústav jaderného výzkumu Řež, a.s. (především mému vedoucímu, Ing. Ladislavu Horáčkovi, CSc.), za shovívavost a vstřícnost v souvislosti se zpracováním této práce, a za umožnění nahlížení do odborné literatury, která by mi jinak zůstala nedostupná. Děkuji.

# Abstrakt

---

## *Methods for ERP signal processing*

This master's thesis addresses the subject of processing of recorded event-related potential (ERP) data for the purposes of expert interpretation, namely the issue of averaging.

Starting with a brief introduction to ERP averaging, the introductory part lists selected methods available that are used for processing of raw ERP data to achieve better results of averaging with discussion of feasibility of each method for purposes of this thesis. The method selected for implementation as a part of the thesis work, the Dynamic Time Warping method, is described in detail in following section with emphasis on modifications made to the method that greatly improve its usability and results. Follow a brief comment on selected implementation details and a commented summary of achieved results.

The Dynamic Time Warping method, as promising as may sound in cited papers, proved to be of a very limited benefit when applied to general recording data that has not been processed in order to achieve good results. In this thesis an improved version of this method is proposed that improve results considerably and an implementation of this improved method is provided as a plug-in module for EEGLAB.

# Obsah

---

1	Úvod.....	1
1.1	Požadavky neuroinformatické skupiny KIV.....	1
1.2	Vysvětlení pojmů .....	2
2	Problematika vyhodnocení záznamů evokovaných potenciálů .....	4
2.1	Evokované potenciály .....	4
2.2	Frekvenční doména .....	7
2.3	Průměrování a grand average .....	8
2.4	Vybavení laboratoře .....	10
3	Přehled zkoumaných metod.....	12
3.1	Woodyova metoda .....	12
3.2	Metody detekce lokálních extrémů.....	14
3.3	Analýza nezávislých komponent .....	15
3.4	Metody založené na minimalizaci určené metriky .....	16
3.5	Metoda nejdelší společné podposloupnosti .....	16
4	Metoda DTW.....	19
4.1	Základní popis metody DTW.....	19
4.1.1	Vyhazení vzorových dat.....	21
4.2	Ohraničení a určující parametry.....	22
4.2.1	Váhy složek metriky.....	22
4.2.2	Parametr omezení délky kroku ve směru osy (point drift constraint) .....	23
4.2.3	Parametr omezení vzdálenosti hodnoty mapovací funkce od diagonály (diagonal drift constraint).....	24
4.2.4	Parametr intenzity vyhlazení vzorových dat (smoothing amount) ....	25
4.3	Algoritmus .....	26
5	Implementace.....	28
5.1	Volba programovacího jazyka a prostředí.....	28
5.1.1	Samostatný program .....	28
5.1.2	Zásuvný modul do programu JERPA .....	29
5.1.3	Zásuvný modul do programu EEGLAB .....	29
5.2	Implementace zásuvného modulu .....	31
5.2.1	Specifika jazyka a programu EEGLAB.....	31
6	Testování na experimentálních datech.....	34
6.1	Programový test.....	35
6.1.1	Paměťové nároky.....	36
6.1.2	Časová náročnost výpočtu.....	37
6.2	Funkční testování .....	38
6.2.1	Korekce nelineárního posunu dvojice průběhů signálů .....	39
6.2.2	Korekce nelineárního posunu dvojice vyhlazených průběhů signálů .....	42
6.3	Průměrování rozsáhlé datové sady .....	44
7	Zhodnocení dosažených výsledků.....	46
8	Závěr .....	47
9	Zdroje .....	49

# 1 Úvod

---

Tato diplomová práce popisuje implementaci algoritmu *Dynamic Time Warping* pro korekci časové latence jednotlivých komponent záznamu evokovaných potenciálů. Hlavní motivací práce je zlepšit výsledky vyhodnocení záznamů ERP dat poskytnutím lepší metody průměrování, díky které budou lépe zobrazeny signifikantní elementy v zaznamenaných datech.

V úvodu práce je nastíněna problematika vyhodnocování záznamů evokovaných potenciálů a objasněna motivace pro korekci proměnlivé časové latence jednotlivých komponent, popsány známé metody korekce proměnlivé latence a standardní programové prostředky pro zpracovávání záznamů EEG. Po teoretickém úvodu je objasněna volba metody *Dynamic Time Warping*. Metoda byla oproti popisu v literatuře upravena tak, že poskytuje dobré výsledky i pro data s malým odstupem signálu od šumu, a implementována jako zásuvný modul programu *EEGLAB*. Následují implementační detaily a popis funkcí programu. Závěr práce tvoří zhodnocení dosažených výsledků s popisem vlivu jednotlivých parametrů na charakter výsledku.

## 1.1 Požadavky neuroinformatické skupiny KIV

Neuroinformatická skupina Katedry informatiky a výpočetní techniky Fakulty aplikovaných Věd Západočeské univerzity v Plzni se dlouhodobě zabývá výzkumem v oblasti elektroencefalografie, evokovaných potenciálů a neuroinformatiky. Činnosti skupiny zahrnují návrhy a provádění experimentů vlastních nebo ve spolupráci s neurologií Fakultní nemocnice Plzeň a dalšími akademickými institucemi, vývoj softwaru pro provádění a vyhodnocování experimentů a vývoj algoritmů pro automatické zpracování zaznamenaných dat.

Požadavkem skupiny byla analýza dostupných algoritmů pro zlepšení výsledků průměrování a vytváření souhrnných průměrů záznamů pokusů v oblasti

evokovaných potenciálů. Na základě rozboru má být vybrán algoritmus s předpoklady největšího přínosu pro vyhodnocování pokusů, ve kterých se zaznamenávají reakce na dva nebo více různých typů stimulů a je třeba popsat soulad nebo rozdíl mezi reakcemi na dané stimuly. Při pokusech se zaznamenávají data z tří až jednadvaceti elektrod na skalpu. Implementace má být určena výhradně pro vyhodnocení člověkem pomocí vhodného grafického rozhraní a rozhodující charakteristikou hodnocení je zvýraznění detailů průběhu odezvy. Rychlost zpracování a prezentace výsledku, nároky na paměť a další výpočetní prostředky musejí odpovídat použití. Program tedy musí výsledek prezentovat v takovém čase, aby umožnil pohodlné zpracování záznamů pokusů člověkem. Musí být schopen běhu na běžném stolním nebo přenosném počítači. Výstup by měl být obrazový s možností využití ve výzkumných zprávách a člancích. Dále by program měl umožňovat dávkové zpracování dat.

### 1.2 Vysvětlení pojmů

Následující pojmy jsou vysvětleny pouze pro účely této práce, popisy tedy nemusejí být úplné ani univerzálně platné.

*Stimul* lze pro účely této práce popsat jako okamžik v časové ose záznamu, kdy nastal podnět pro evokovaný potenciál. Tento okamžik se spolu s informací o typu stimulu ukládá do záznamu EEG dat a při extrakci *epoch* je zpravidla umístován do času 0.

*Epocha* je časové období, ve kterém je zaznamenáván průběh evokovaného potenciálu. V této práci je délka epochy zpravidla 3 vteřiny, *stimul* je prezentován v 1. vteřině. Časová osa grafu proto začíná v okamžiku -1 000 ms a končí okamžikem 2 000 ms.

*Snímek* je záznam jedné epochy z jedné elektrody. Jestliže datový soubor obsahuje záznam 32 elektrod a 10 *epoch*, skládá se z 32 x 10, tedy 320 *snímků*.

*Vzorek* představuje číselné vyjádření intenzity zaznamenaného signálu v daném okamžiku (zpravidla kvantizovaná hodnota po převodu analogového signálu



na číslicový). Je-li snímek dlouhý 3 vteřiny a frekvence snímkování (angl. *sampling rate*) 128 Hz, obsahuje *snímek* 384 *vzorků*.

*Komponenta* je signifikantní časový úsek průběhu signálu v epoše, často charakterizovaný lokálním minimem nebo maximem. *Subkomponenta* je pak výrazná nebo charakteristická část *komponenty*. Detailní vysvětlení tohoto pojmu poskytuje následující kapitola.

*Časový průběh (signálu)* v této práci označuje průběh amplitudy signálu ve snímku. Matematicky odpovídá funkci určené výčtem hodnot.

*Vzor (vzorová data)* je snímek, na který je metodou *DTW* mapován *časový průběh*. Zpravidla je pro celý soubor dat použit jeden *vzor*, který je určen buď jako náhodně nebo cíleně vybraný *snímek*, nebo jako aritmetický průměr hodnot všech *snímků* v každém bodě.

*Mapování* je výsledek činnosti algoritmu metody *DTW*, při které je *časový průběh* dynamicky mapován na časovou osu. Časový průběh je smršťován nebo natahován tak, aby byla minimalizována celková odchylka *vzoru* a *časového průběhu*.

*DTW, Dynamic Time Warping*. V literatuře dosud není zaveden český název; slovo *warping* v rovině označuje deplanaci. Termín *warping function* bude dále v textu označován českým *mapovací funkce*, příp. *mapování* s poznámkou, že české označení není úplně ekvivalentní anglickému.

*Grand average* je terminus technicus pro souhrnný průměr dat ze všech souvisejících souborů (např. záznamů deseti osob, se kterými je prováděn týž pokus).

## 2 Problematika vyhodnocení záznamů evokovaných potenciálů

---

Tato kapitola poskytuje základ nezbytný pro objasnění možností a funkce metody a možností splnění požadavků neuroinformatické skupiny KIV.

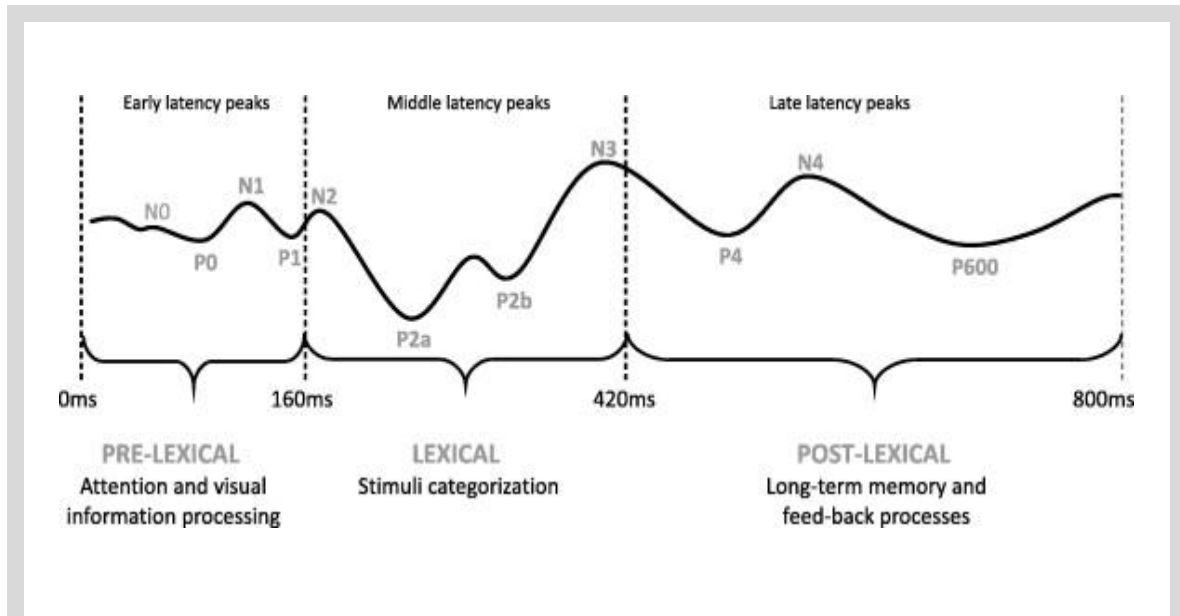
### 2.1 Evokované potenciály

Činnost nervových buněk (přenos vzruchu z buňky na buňku) se projevuje potenciálem o amplitudě až -70 mV. Lékařská disciplína zabývající se záznamem a analýzou těchto projevů je elektroencefalografie (zkr. *EEG*). V klinické praxi V lékařských aplikacích se lze setkat s příkládáním elektrod na obnažený mozek nebo jiná nervová zakončení, neuroinformatika pracuje zpravidla se záznamem pořízeným na pokožce hlavy, kde dosahuje amplituda změřitelného potenciálu 10  $\mu$ V až 100  $\mu$ V [1].

EEG signál může být reprezentován dvěma způsoby, buď jako soubor dipólů, kdy je signál zobrazen jako rozdíl v potenciálu mezi dvěma elektrodami, nebo jako referenční model, kdy jsou hodnoty potenciálu na všech elektrodách vztaženy k referenční nule. V této práci je standardem druhá možnost, tedy referenční model.

Evokované potenciály (angl. *event-related potentials*, zkr. *ERP*) jsou měřitelné elektrické projevy činnosti mozku, která je vyvolaná vnitřním nebo vnějším podnětem, *stimulem*. Jedná se tedy o měřitelnou odezvu mozku, která je přímým následkem vnímání [5]. Evokované potenciály se z elektroencefalogramu vyhodnocují v časové doméně – vlastnosti daného evokovaného potenciálu ukazují lokální amplitudy průběhu velikosti potenciálu obou polarit v určených latencích od prezentace stimulu. Jednotlivá lokální maxima a minima evokovaných potenciálů se nazývají komponenty a jsou označovány kombinací písmene a čísla. Písmeno označuje polaritu komponenty (P pozitivní a N negativní komponentu)

a číslo označuje buď latenci amplitudy, nebo její pořadí (např. N1 označuje první lokální negativní amplitudu, P300 pak pozitivní amplitudu s latencí 300 ms; viz Obrázek 1). V následujícím textu budou komponenty označovány zpravidla pořadím, tedy N1, P3 apod.



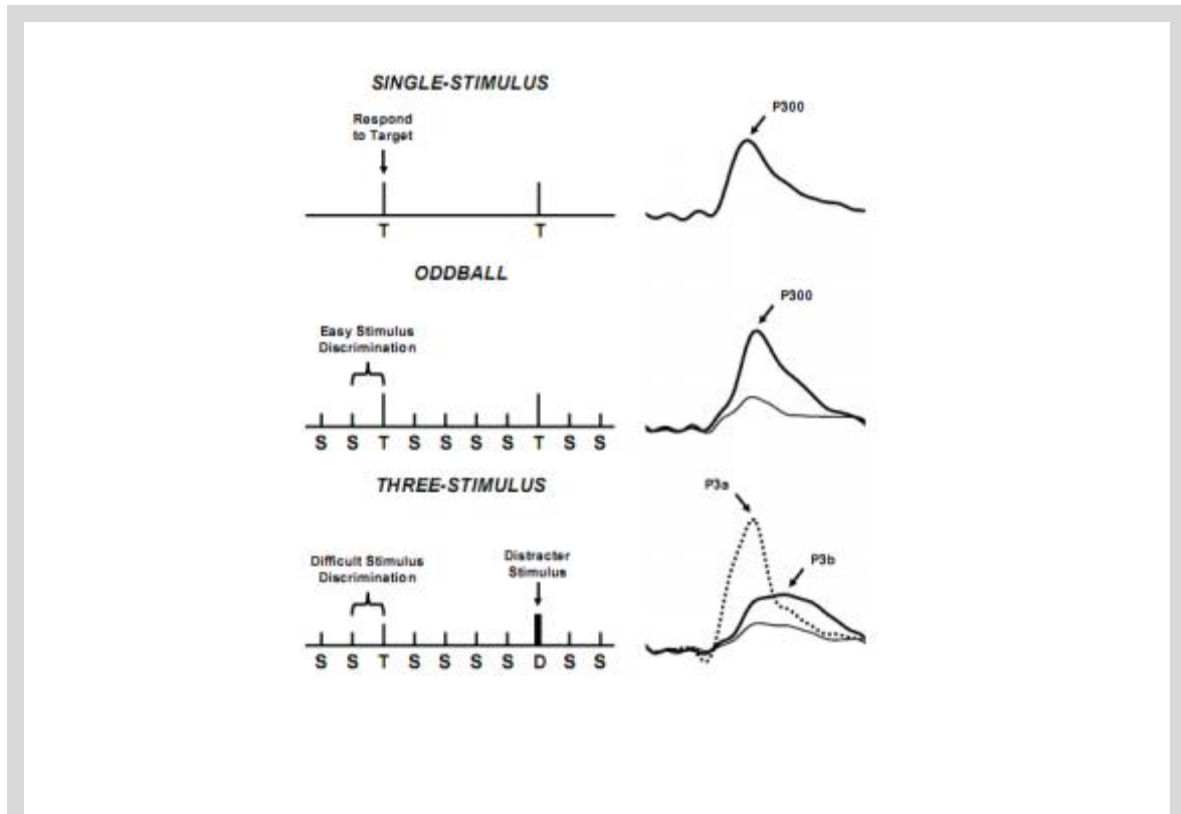
Obrázek 1: schéma časového průběhu významných evokovaných potenciálů. Ilustrace naznačuje číslování významných komponent vizuálně vyvolaných evokovaných potenciálů a jejich umístění na časové ose. Do 160 ms se objevují předlexikální reakce (na obr. *pre-lexical*) ukazující pozornost a příjem vizuální informace, od 160 ms do 420 ms lexikální reakce (*lexical*), kterými se projevuje prvotní myšlenkové zpracování stimulu, od 420 ms dále se projevují pozdější činnosti mozku spojené se stimulem, např. paměťové procesy nebo zpětná vazba vyvolaná stimulem (*post-lexical*). Ze schématu je patrné, že zájmových komponent může být v jedné epoše několik (podle aplikace) a mohou se nacházet v různých časových oblastech epochy.

*Poznámka:* Z historických důvodů je v literatuře na svislé ose záznamů EEG obrácená polarita, tedy záporné hodnoty amplitudy jsou nad vodorovnou osou a kladné pod ní.

Obrázek je převzat z [5].

Velmi často zkoumanou komponentou je P3, komponenta s výraznou kladnou amplitudou vrcholící přibližně 300 ms a více po prezentaci stimulu. Latence a průběh této komponenty se mění v závislosti na komplexitě vjemu – platí, že komplexnější stimuly způsobují protažení vlny P3 a tedy i opoždění vrcholu amplitudy. Vlastnosti komponenty P3 (přítomnost, velikost, topografie a latence)

se v neurologii užívají jako metrika kognitivních funkcí v rozhodovacím procesu. Nejpatrnější je v parietální oblasti a skládá se ze dvou částí označovaných P3a a P3b (viz Obrázek 2).



Obrázek 2: subkomponenty vlny P3. Ilustrace ukazuje průběh komponenty P3 v závislosti na způsobu prezentace stimulu (prezentace stimulu samotného, prezentace stimulu mezi jinými stimuly – tzv. *odball*, prezentace stimulu se záměrným následným rušením pozornosti). Ilustrace naznačuje důležitost tvaru a časového umístění komponenty pro vyhodnocení. Obrázek je převzat z [8].

Tyto subkomponenty se objevují nezávisle na sobě a mají odlišný význam – P3a vzniká ve frontálních mechanismech zpracování stimulů během zpracování úlohy, kdežto P3b je způsobená temporálně-parietální aktivitou spojenou s pozorností a je často spojována s následným zpracováním vzpomínek. P3a je proto také nazývána *non-target vlnou* (tedy vlnou nezávislou na vztahu subjektu ke stimulu) a P3b pak *target vlnou* (tedy vlnou ukazující na vzpomínání a další nekognitivní zpracování stimulu) [8].

Ačkoli je komponenta P3 při většině pokusů nejvýraznější a nejlépe vyhodnotitelnou komponentou (často tvoří základ vyhodnocení), pro mnoho neklinických výzkumných pokusů je třeba vyhodnotit všechny komponenty včetně pozdních, tedy celý průběh epochy až do latence 1000 ms až 2000 ms. Detekce a vyhodnocení komponenty P3 se těší velké pozornosti akademické a lékařské obce a je k dispozici mnoho metod a algoritmů pro různé přístupy k vyhodnocení. Metod pro pokročilé průměrování celých epoch s variabilní latencí jednotlivých komponent není k dispozici mnoho, k těm ověřeným pak není k dispozici dostupná implementace.

## 2.2 Frekvenční doména

Elektrická aktivita probíhající najednou ve velkém množství nervových buněk má periodický průběh s frekvencí jednotek Hz. Tyto projevy se nazývají EEG aktivita, EEG vlny nebo EEG rytmy (angl. *wave*, *brain rhythm*). Zkoumání záznamů EEG ve frekvenční doméně je mimo předmět této práce. Tyto projevy zde jsou vyjmenovány, protože mohou ovlivnit záznam evokovaných potenciálů. Označují se názvy řeckých písmen postupně podle frekvenčního pásma, ve kterém se projevují.

Aktivita alfa se projevuje ve frekvenčním pásmu 8 Hz až 13 Hz, většinou se sinusovým průběhem a amplitudou do 50  $\mu\text{V}$ . Při činnostech vyžadujících soustředění se tato aktivita neprojevuje. Aktivita beta se projevuje v pásmu 14 Hz až 26 Hz (některá literatura uvádí vyšší horní hranici) a amplitudou do 30  $\mu\text{V}$  především ve frontálních a centrálních oblastech skalpu. Tato aktivita je zjištěna zpravidla při soustředěném myšlení, ustupuje při motorické aktivitě nebo taktilní stimulaci. Aktivita gama se nejvýrazněji projevuje v pásmu 30 Hz až 45 Hz, ovšem s velmi nízkými amplitudami. Její zjištění často slouží jako součást průkazu některých nervových onemocnění. Aktivita delta má velmi nízké frekvence, zpravidla 1 Hz až 4 Hz a velmi vysokou amplitudu, až 200  $\mu\text{V}$ . Pro vyhodnocení evokovaných potenciálů ovšem většinou neznamena komplikaci, protože je zjišťována pouze u kojenců a osob s fyziologickým poškozením mozku. Literatura

jmenuje i další aktivity s periodickým projevem, ty ovšem nemají přímý vliv na vyhodnocení pokusů neuroinformatickou skupinou KIV.

### 2.3 Průměrování a grand average

Při vyhodnocování pokusů zaměřených na evokované potenciály se velmi často porovnávají záznamy reakcí mozku na dva nebo více různých typů stimulů. Příkladem může být úloha počítání sudých čísel, kdy jsou subjektu na obrazovce promítána různá čísla, a subjekt má za úkol počítat, kolik sudých čísel bylo zobrazeno. Záznam EEG je v průběhu pokusu doplňován značkami, díky kterým lze jednotlivé typy stimulů rozlišit při následném zpracování. V praxi je počet zobrazení každého stimulu v řádu desítek až stovek, zaznamenávají se minimálně tři elektrody v podélné ose skalpu (nadčelní – *frontální*, temenní - *centrální*, a týlní - *okcipitální*; zkr. *Fz*, *Cz*, *Oz*), maximálně pak plná konfigurace 21 elektrod rozmístěných po celém skalpu.

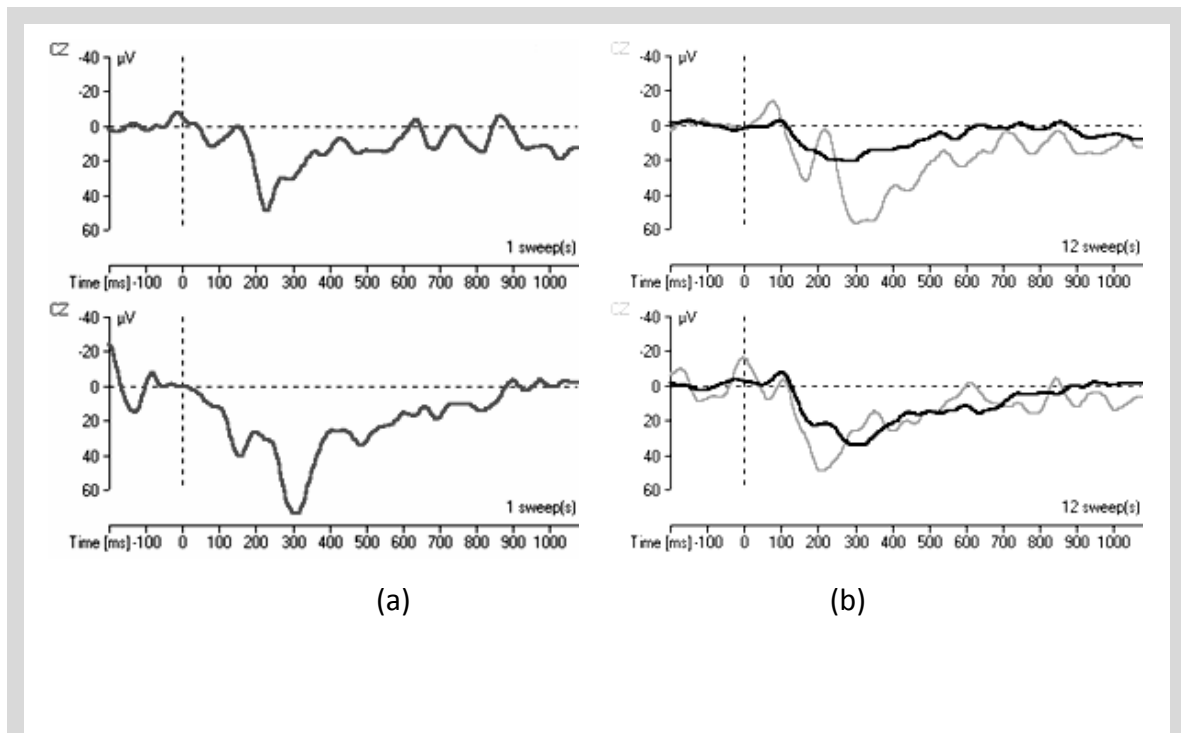
Rozdíly mezi reakcemi na jednotlivé typy stimulů se hledají v průměrech vypočtených ze všech záznamů opatřených značkou daného typu stimulu (průměruje se hodnota každého vzorku ze všech záznamů), případně v průměrech ze všech elektrod a všech záznamů daného typu stimulu i napříč několika pokusy nebo měřeními (souhrnný průměr, angl. *grand average*).

Z různých důvodů se časový průběh elektrického projevu mozkové činnosti liší jak v záznamech z různých elektrod (především kvůli zapojení různých částí mozku v různém okamžiku vyhodnocování stimulu, nestejněměrné šíření vzruchu po povrchu skalpu apod.), tak v záznamech pořízených v jiném čase v průběhu pokusu (kdy dochází k posunu reakcí delších latencí např. z důvodu menšího soustředění, únavy nebo přivyknutí úloze).

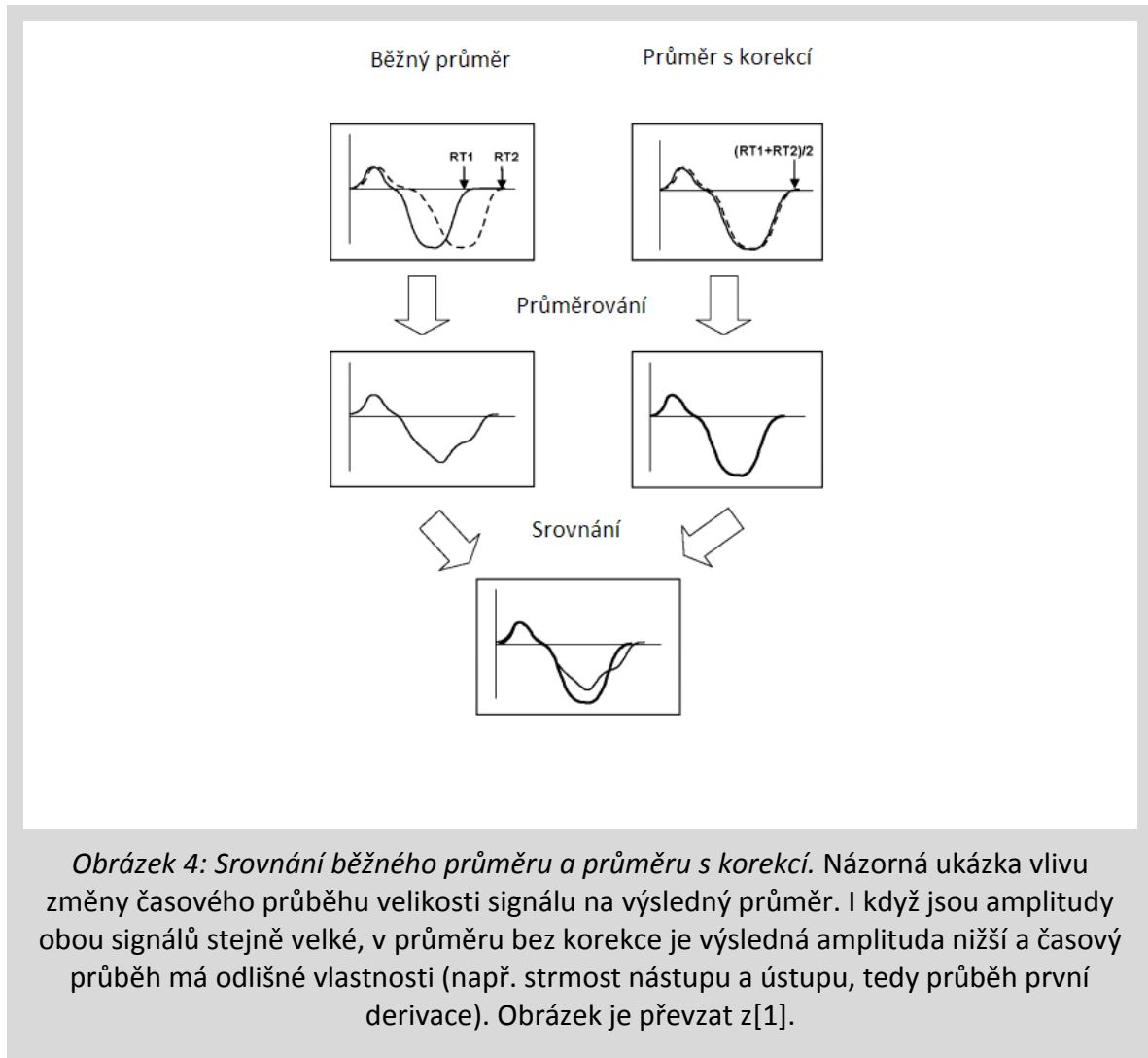
Vlivem různého časového průběhu špiček signálu pak dochází k tomu, že data po průměrování nevykazují takové vlastnosti, jaké by bylo možné vyčíst z každého individuálního snímku. Především se průměrováním komponent byt' s minimálně

posunutou latencí ztrácejí důležité detaily, které mohou ovlivnit vyhodnocení. Problém ilustrují následující obrázky (viz Obrázek 3).

Proto vyvstává potřeba pokročilého sjednocení dat z více snímků tak, aby ve výsledném průměru nezakly významné detaily průběhu velikosti signálu, především lokální maxima, minima a strmosti nástupu a ústupu komponent (viz Obrázek 4).



Obrázek 3: (a) první snímek pokusu, (b) dvanáctý snímek pokusu. Obrázky ukazují, jak dochází ke zkreslení výsledného signálu průměrováním snímků s nesterajným časovým průběhem velikosti potenciálu. Na obrázku (b) představuje světlá čára aktuální snímek, tmavá pak průměr ze všech 12 snímků (ty nejsou součástí obrázku). Obrázky jsou převzaty z [1].



## 2.4 Vybavení laboratoře

Neuroinformatická laboratoř KIV je vybavena pro provádění různorodých experimentů především v oblasti pozornosti a činnosti mozku v automobilovém provozu, ovládání počítače mozkiem (angl. *brain computer interface, BCI*) a výzkum a pokusy z oblasti evokovaných potenciálů a reakcí na různorodé stimuly. Primárním cílem této práce je rozšířit sadu nástrojů laboratoře pro vyhodnocení zaznamenaných dat, výběr algoritmu a implementačních detailů se tedy odvíjí i od vybavení laboratoře a standardního postupu vyhodnocování naměřených dat.

Záznamový řetězec tvoří zařízení od společnosti *Brain Products GmbH* se sídlem v Mnichově. Záznamový řetězec je tvořen čepicí s elektrodami (jsou k dispozici



čepice různých provedení a velikostí, s různým počtem elektrod) a zesilovačem *Brainvision Brainamp DC* s bateriovým zdrojem *Brainvision Powerpack* (zesilovač je napájen bateriovým zdrojem, aby bylo odstraněno rušení, které proniká napájecím zdrojem z rozvodu střídavého napětí). Samotná záznam a digitalizace dat je provedena prostřednictvím záznamového softwaru *Brainvision Recorder*. Pro pokusy s textovými, obrazovými a zvukovými stimuly je používán program *Presenation* od kalifornské společnosti *Neurobehavioral Systems*, synchronizace záznamu dat při těchto pokusech zajištěna zařízením *Brainvision Syncbox*.

Pro vyhodnocení zaznamenaných dat má katedra k dispozici licenci programu *Brainvision Analyzer*. Studenti pracující na projektech a kvalifikačních pracích využívají často program *EEGLAB*, který je vyvíjen v centru *Swartz Center for Computational Neuroscience* na *University of California San Diego*. Program *EEGLAB* je k dispozici zdarma a je implementován jako toolbox v prostředí *Matlab*. Studenti a zaměstnanci katedry mají k dispozici *Matlab* v několika univerzitních a katedrálních laboratořích, pro studenty je rovněž k dispozici cenově dostupná licence pro užívání tohoto prostředí ve vlastních počítačích. *EEGLAB* bude detailně popsán v dalším textu.

## 3 Přehled zkoumaných metod

---

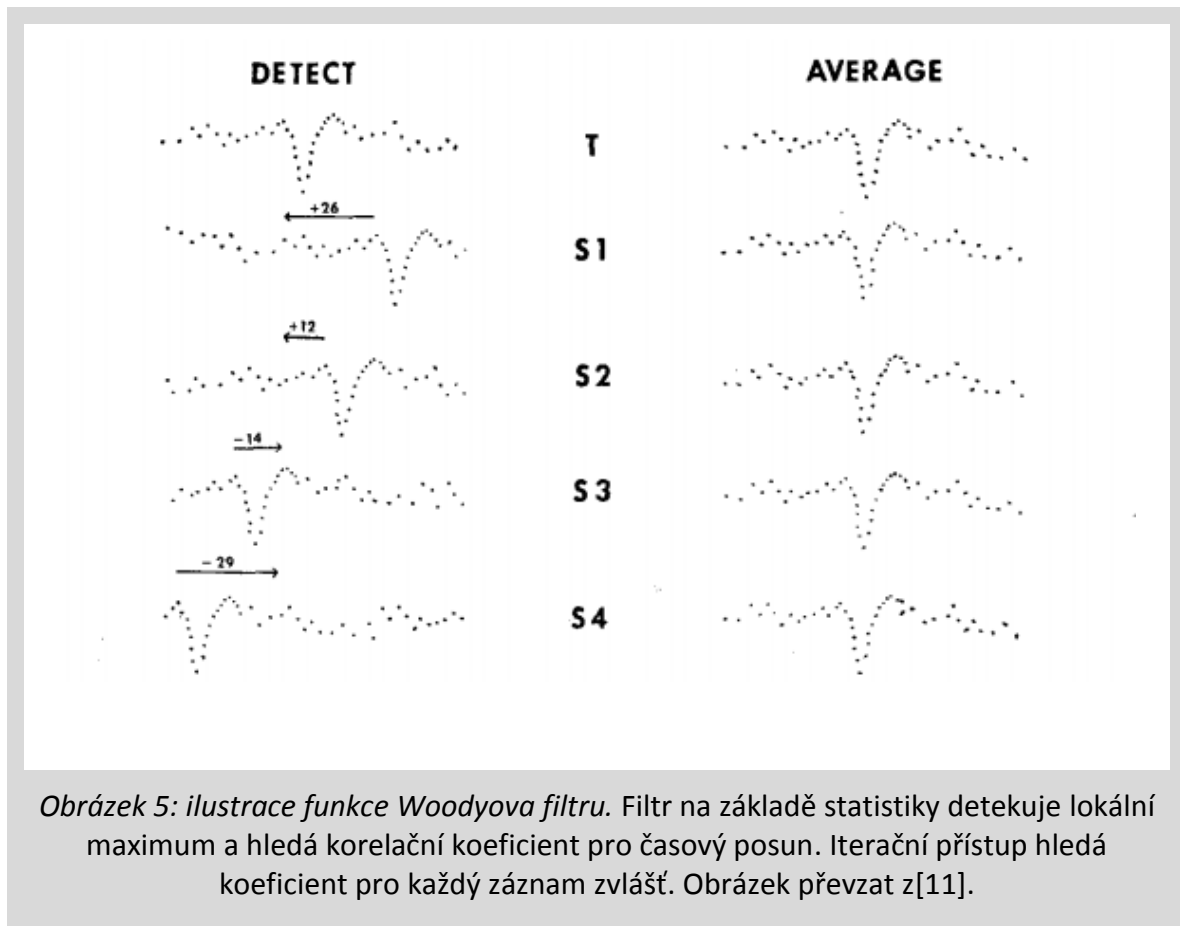
Rychlý rozvoj neuroinformatiky a *BCI* přinesl požadavky na vývoj a implementaci algoritmů a metod pro ruční nebo automatické potlačení šumu, detekci a posouzení vlastností jednotlivých komponent; a dalšího zpracování záznamů EEG a evokovaných potenciálů. V oblasti zpracování záznamů analogového signálu lze nalézt širokou škálu metod a algoritmů, z nichž některé již byly popsány nebo implementovány pro účely vyhodnocení záznamů EEG. Tyto algoritmy a metody lze rozdělit na automatické, parametrické a manuální, případně podle velikosti nebo charakteru změn v signálu po aplikaci těchto algoritmů. I když některé níže jmenované algoritmy nejsou určeny pro úpravu signálu pro ruční vyhodnocení, vzhledem k jejich charakteru byly vzaty v úvahu při hledání nejvhodnější metody.

### 3.1 Woodyova metoda

Woodyův adaptivní filtr (1967, Charles D. Woody, [11]) je první široce využívaná metoda pro zlepšení výsledků průměrování digitalizovaného záznamu EEG. Metoda umožňuje detekovat neuroelektrický signál s různou latencí a separovat jej od šumu na základě korelačních a statistických metod [11]. Použití korelační formule

$$\Phi = \frac{1}{T} \int_0^T f(t)g(t - \Delta\tau)dt \quad (1)$$

kde  $T$  je celková délka  $f(t)$  a  $g(t)$  v čase,  $f(t)$  a  $g(t)$  průběh signálů a  $\tau$  časový posun  $g(t)$ , umožňuje nalezení lineární závislosti  $f(t)$  a  $g(t)$  a jejich vzájemné posunutí v čase. Metoda je založena na postupném porovnání párů časových průběhů signálu. Jeden časový průběh signálu je určen jako řídicí a ostatní průběhy jsou k němu zarovnávané (viz Obrázek 5).

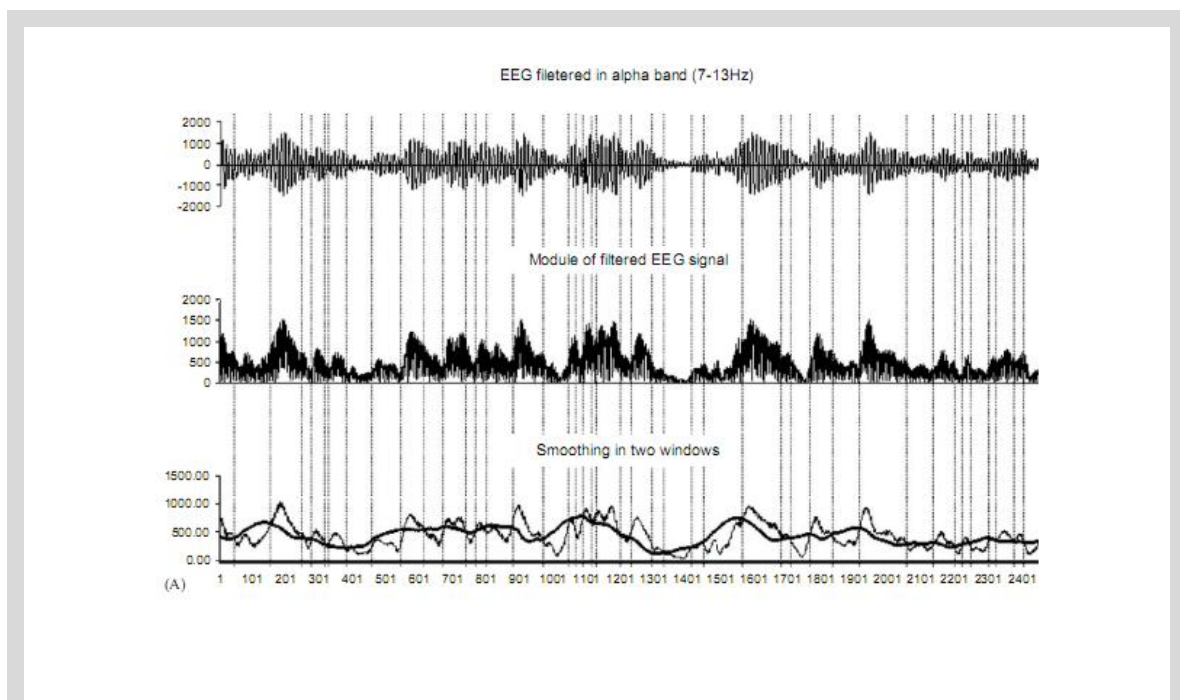


Woodyova metoda se jevila jako vhodný kandidát a též byla navrhována k implementaci neuroinformatickou skupinou. Při prvotních pokusech s uměle připravenými zkušebními daty a implementacích v jazyce C se ukázal problém nalezení  $\tau$  jako dobře řešitelný, přímočarý a výpočetně nenáročný. Filtr ovšem poskytuje dobré výsledky pouze při vysoké úrovni lineární závislosti jednotlivých záznamů, a umožňuje zlepšení výsledků průměrování vždy pro jednu vybranou komponentu. V případě výrazných časových posunů jednotlivých komponent vůči sobě však filtr nedokáže zlepšit výsledky ani u jedné signifikantní komponenty, výsledky může naopak zhoršit.

Woodyova metoda použití adaptivního filtru je omezena volbou jedné časové korekce pevné pro celý průběh epochy, a proto nesplňuje požadavek na zlepšení popisu celkové reakce na rozdílné stimuly.

### 3.2 Metody detekce lokálních extrémů

Metoda hledání lokálních extrémů využívá celou řadu algoritmů (především statistických) pro vyhledání posloupnosti lokálních maxim a minim a následné zarovnání těch, které vykazují definované společné znaky. Základem všech metod je pokus o automatickou identifikaci lokálních extrémů, jejich seskupování a následné zarovnávání segmentů mezi extrémy (Aunon et al., 1981, [12]; Acir, 2012, [13]; Vincent, 1992 [14];). Oproti Woodyovu filtru umožňují tyto metody nelineární korekci, a tedy jsou vhodné pro použití na celou dobu epochy.



Obrázek 6: ilustrace hledání tzv. statisticky inflexních bodů v časovém průběhu signálu.

Obrázek ilustruje segmentaci signálu pomocí průměrování se dvěma různými velikostmi klouzavého okénka. Jednotlivé segmenty začínají a končí v okamžiku, kdy se dva průměrované signály protnou. V jednotlivých segmentech se pak hledá celkové maximum nebo minimum (podle poměru průměrné hodnoty obou klouzavých průměrů v daném segmentu). Obrázek převzat z [1].

V rámci této práce byly zkoumány různé metody detekce lokálních extrémů a jejich vhodnost pro úlohu nelineární korekce latencí. Dále byly provedeny experimenty se segmentací na základě tzv. statisticky inflexních bodů, která signál rozděljuje na významové segmenty a v každém segmentu je pak nalezeno absolutní maximum

nebo minimum, které slouží pro korekci. Tyto statisticky inflexní body byly hledány pomocí klouzavého průměrování se dvěma velikostmi okénka, jak ilustruje Obrázek 6.

Hlavní nevýhodou všech metod založených na hledání lokálních extrémů jsou vysoké nároky na nastavení parametrů hledání. Parametry je třeba volit znovu pro každý soubor záznamů, s ohledem na cíl pokusu a na základě znalosti zaznamenaných dat. Další nevýhodou je linearita mapování dat mezi zarovnanými extrémy. Díky tomu nesplňují požadavky na zvýraznění detailů jednotlivých komponent a možnost dávkového zpracování.

### 3.3 Analýza nezávislých komponent

Analýza nezávislých komponent (angl. *independent component analysis, ICA*) je pokročilá matematická metoda pro hledání signifikantních součástí záznamu a jejich oddělení od šumu. Původně byla vyvinuta pro odlišení významové části vícekanálového zvukového záznamu od okolního šumu a nevýznamových součástí. Metoda *ICA* je již implementovaná v programech *JERPA* i *EEGLAB*, a proto by bylo neúčelné ji implementovat v rámci této práce. Je hojně využívána především pro odstranění šumu a vyhledání jednotlivých významných komponent v záznamech EEG, v tomto výčtu je uvedena především z důvodů vyjmenování jejich nevýhod pro účely potřeb neuroinformatické skupiny EEG a srovnání s metodou vybranou pro implementaci. Existuje několik způsobů implementace, některé byly provedeny i v rámci neuroinformatické skupiny KIV (např. [6]).

Využití metody *ICA* pro úlohu průměrování dat brání dvě zásadní úskalí. Především se metoda snaží minimalizovat informaci společnou všem vzorkům záznamu, a proto pracuje vzhledem k úloze průměrování nedeterministicky. Metoda je velmi vhodná pro nalezení signifikantních specifik signálu, vzhledem k úloze průměrování se chová nedeterministicky: „Nevýhodou lokalizace zdroje je, že dopředný model a statistické odhady mohou vést k chybným výsledkům. ... Zvyšování počtu dipólů rapidně snižuje stabilitu řešení a různé počáteční odhady umístění zdroje a jeho orientace mohou přinést velmi odlišné výsledky“ [9].

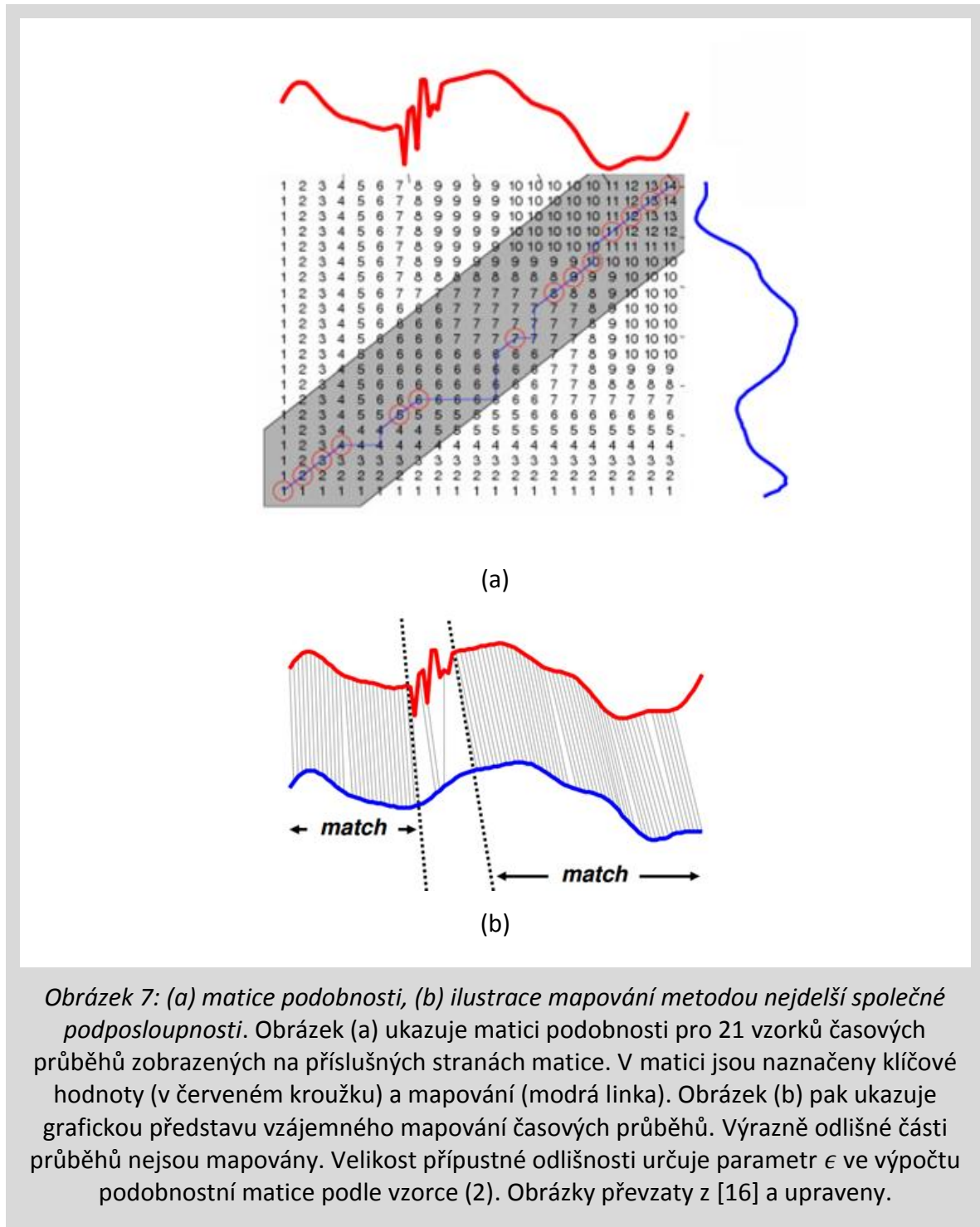
### 3.4 Metody založené na minimalizaci určené metriky

Dalšími zkoumanými metodami byly metody používané v úlohách umělé inteligence, zejména rozpoznávání mluvené řeči. Při rozpoznávání vysloveného výrazu je třeba jeho záznam porovnat se vzorem. Základem dvou zkoumaných metod je snaha o zajištění maximálního souladu lokálních extrémů strmostí dvou časových průběhů před vlastním porovnáváním (např. víceslovný výraz může být vyřčen s jinými důrazy nebo pomlkami, přesto vykazuje podobný průběh), z tohoto důvodu byl u následujících metod dobrý předpoklad využitelnosti pro úlohu zlepšení průměrování dat.

### 3.5 Metoda nejdelší společné podposloupnosti

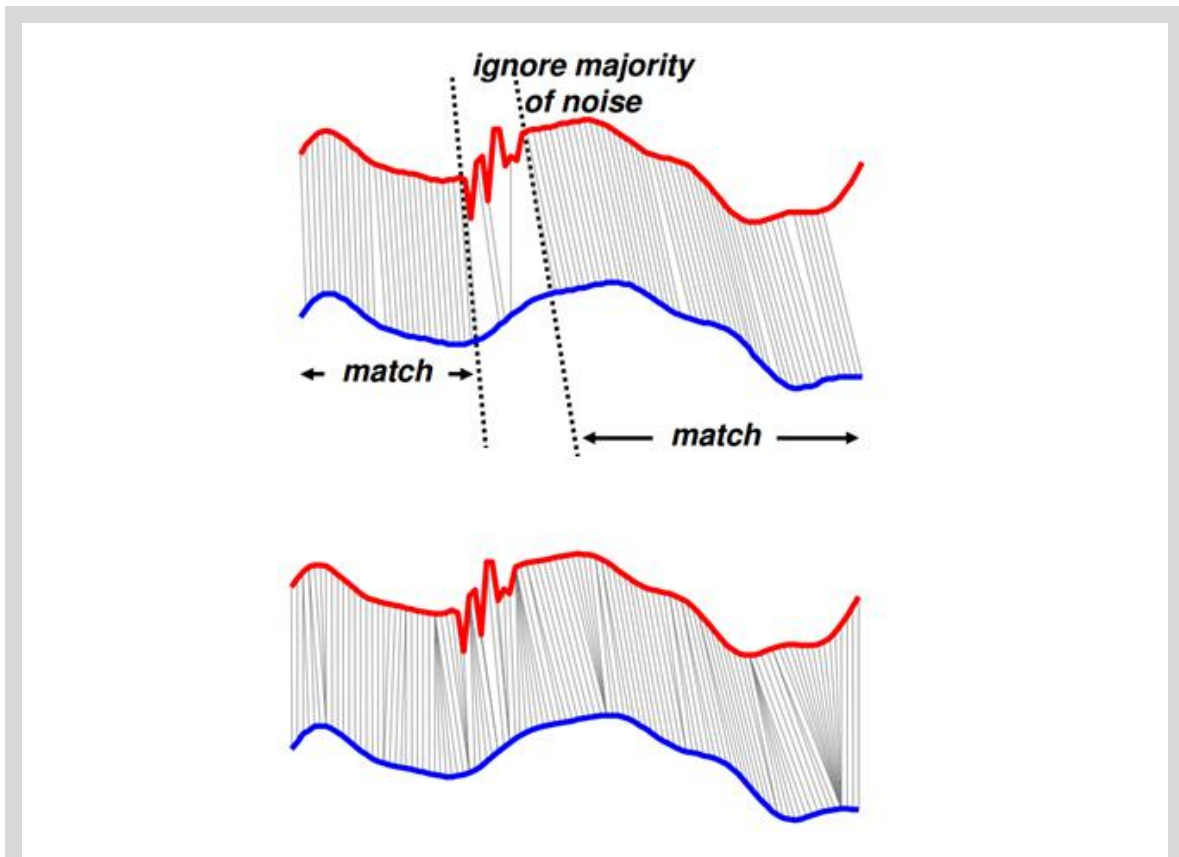
Jak napovídá její název, metoda hledání nejdelší společné podposloupnosti je založená na hledání částí časového průběhu signálu, které jsou společné oběma vstupním průběhům. Aby byla zajištěna časová kontinuita, lze metodu implementovat jako mapovací funkci, kdy je průběh jednoho vstupu vzorem a druhý průběh je mapován na společnou časovou osu na základě matice podobnosti *LCSS* (z angl. *longest common subsequence similarity matrix*, viz vzorec (2) pro výpočet hodnot matice). Tento přístup pomocí dynamického programování zajišťuje časovou kontinuitu mapování a lze díky němu mapovat i průběhy s různou základnou (angl. *baseline*). Mapování ilustruje Obrázek 7.

$$LCSS[i, j] = \begin{cases} 0, & i = 0 \text{ nebo } j = 0 \\ 1 + LCSS[i - 1, j - 1], & |a_{i,k} - b_{j,k}| < \epsilon \\ \max(LCSS[i - 1, j], LCSS[i, j - 1]), & \text{v ostatních případech} \end{cases} \quad (2)$$



Tato metoda je velmi vhodná pro zpracování dat zatížených šumem, protože pomocí správné volby parametru  $\epsilon$  umožňuje průběžně šum odstranit nebo ignorovat. Potřeba dobré volby parametru  $\epsilon$  však znesnadňuje použití této metody pro dávková zpracování.

Metody zpracování nahrávek mluvené řeči však nabízejí ještě jednu metodu, která nepřipouští ignorování žádných částí časového průběhu, a sice metodu dynamického mapování dat na časovou osu, angl. *dynamic time warping* (odtud zkratka *DTW*; viz Obrázek 8). Vzhledem k možnosti úplného mapování byla metoda DTW zvolena pro implementaci v rámci této práce. Pojednává o ní následující samostatná kapitola.



Obrázek 8: Grafické znázornění rozdílu mezi mapováním časových os metodou nejdelší společné podposloupnosti a metodou DTW. Zatímco metoda nejdelší podposloupnosti může na základě volby parametru  $\epsilon$  ignorovat určité části časového průběhu (zde zvýrazněno), metoda DTW mapuje celý průběh. Obrázek převzat [16] a upraven.



## 4 Metoda DTW

---

Metoda *dynamic time warping* je založena na hledání společných tendencí a lokálních extrémů mezi vzorem a snímkem nezávisle na rozdílech časového průběhu vzoru a snímku. Původně byl algoritmus vyvinut pro *pattern-matching* úlohy v rozpoznávání mluvené řeči. Článek [3] popisuje algoritmus jako proceduru, která lokálně prodlužuje nebo zkracuje časovou osu za účelem zmenšení zkreslení, vzniklého normálními morfologickými rozdíly v časových základnách srovnávaných křivek.

### 4.1 Základní popis metody DTW

Uvažujme dva časové průběhy signálu: vzor  $v(i)$ ,  $i = 1, 2, \dots, I$  a snímek  $s(j)$ ,  $j = 1, 2, \dots, J$ , kde  $i$  a  $j$  jsou digitalizované časové základny vzoru  $v$  a snímku  $s$ . Cílem DTW je nalezení optimálního mapování os  $i$  a  $j$  na společnou časovou osu  $k$  tak, že je minimalizován rozdíl mezi oběma průběhy (podle zadané metriky). Mapovací funkce  $F$  je definovaná jako řada bodů  $c(k) = (i(k), j(k))$  v rovině  $(ij)$ . Každý bod funkce  $F$  udává pozici shody metriky bodu  $i(k)$  na časové ose  $v(i)$  a bodu  $j(k)$  na časové ose  $s(j)$  a reprezentuje optimální mapování těchto dvou časových os, tedy mapování s minimální sumou vzdáleností jednotlivých bodů podle zadané metriky. Metrika vzdálenosti bodů  $d(v(i), s(j))$  je určena funkcí (3):

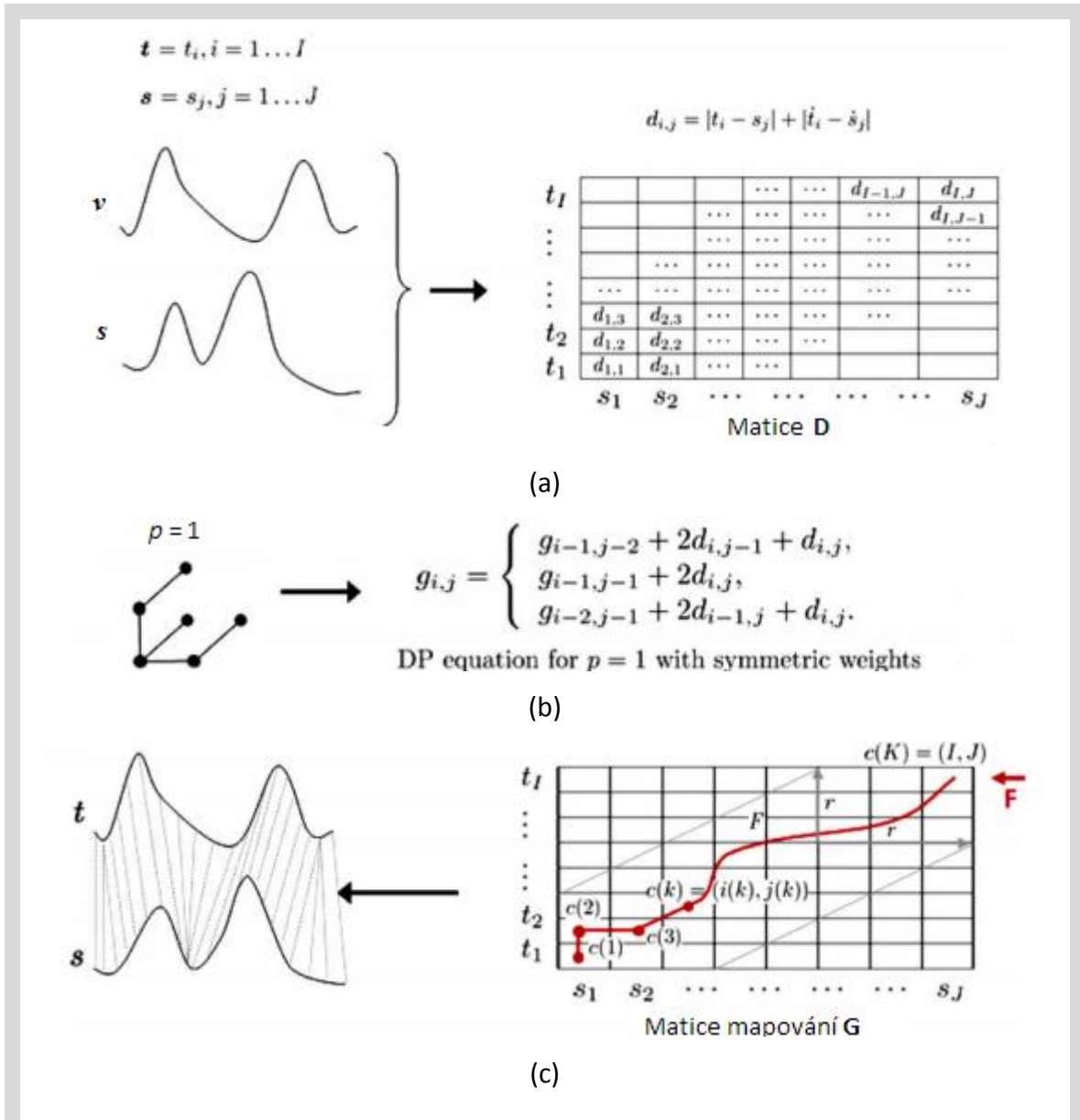
$$d(i, j) = k|v(i) - s(j)| + l|\dot{v}(i) - \dot{s}(j)|, \quad (3)$$

kde  $k$  a  $l$  určují váhu jednotlivých složek (hodnoty úrovně signálu v bodě a numerické hodnoty první derivace v bodě).

Z distanční matice  $D$  je vypočtena matice sumárních vzdáleností  $G$ :

$$g(i, j) = \min_k \sum_{k=1}^{K'} d(i(k), j(k)), \quad (4)$$

kteřá určuje minimální sumu vzdáleností jednotlivých bodů tak, aby bylo prvních  $i$  bodů průběhu namapováno na  $j$  bodů vzoru. Mapovací funkce je nalezena tak, že se v matici  $G$  hledá taková cesta mezi krajními body (levý dolní a pravý horní roh, tedy první a poslední bod průběhu), že suma vzdáleností na této cestě je minimální.



Obrázek 9: (a) výpočet distanční matice (angl. dissimilarity matrix), (b) výpočet volby kroku, (c) ukázka mapování.

Obrázek ilustruje průběh činnosti metody DTW, detailní popis následuje na další straně. Obrázek je převzat z [5].

V případě lineárního mapování (pro které by byla vhodná například výše zmíněná Woodyova metoda), výsledné mapování by bylo  $i(k) = \lfloor j(k) \rfloor$ . Protože je obecně  $F$  nelineární, musí být určeno takové ohraničení vlastností  $F$ , aby mapování dávalo dobré výsledky. Princip metody ilustruje Obrázek 9.

Obrázek 9 (a) ukazuje výpočet matice nepodobnosti mezi vzorem  $v$  a snímkem  $s$  (název  $D$  pochází z angl. *Difference matrix*). Řádky matice představují jednotlivé vzorkované hodnoty vzoru a sloupce pak rozdíl mezi hodnotou příslušného vzorku vzoru  $v$  a hodnotou daného vzorku snímku  $s$  podle určené metriky  $d_{ij}$ . Kompletní matice tak popisuje rozdíl mezi každou hodnotou vzorku a každou hodnotou snímku. Při zadání globálního parametru  $r$  (délka okénka) není třeba vypočítávat všechny hodnoty v matici, ale jen hodnoty do vzdálenosti  $r$  od diagonály. Bez omezení parametrem  $r$  je pro každou matici nepodobnosti třeba vyhradit  $ij$  ( $= i^2$ ) paměti, matice však po dokončení mapování není dále užitečná, a proto stačí paměť pro celý výpočet vyhradit pouze jednou. Obrázek (b) popisuje možné kroky z bodu při hodnotě parametru  $p = 1$  graficky s příslušným výpočtem hodnoty. Obrázek (c) znázorňuje průběh mapovací funkce  $F$  (ten je určen maticí mapování  $G$ ) s naznačením grafického významu parametru  $r$  (délka okénka). V levé části je pak naznačen grafický význam vlastního mapování.

Poznámka: v obrázku je umístěn počátek (souřadnice  $0,0$ ) v levém dolním okraji.

#### 4.1.1 Vyhlazení vzorových dat

Při testování metody se ukázalo, že z dat s nízkým poměrem signálu k šumu nebo z dat rychlým kolísáním signálu v průběhu epochy je velmi obtížné získat dobré výsledky, a pro tato data je určení vhodné kombinace parametrů časově velmi náročné. Protože však metoda poskytovala velmi dobré výsledky pro data s hladkým průběhem, bylo zapotřebí najít způsob, jak umožnit získání podobně dobrých výsledků i pro data s ostře zvlněným průběhem. Do postupu metody byl po zkouškách různých úprav vložen krok vyhlazení zdrojových dat, který výrazně zlepšil výsledky pro zašuměná a nehladká data.

Vyhlazení dat pomocí průměrování klouzavým okénkem je metoda používaná pro detekci lokálních extrémů. Myšlenka využití této metody jako jednoho z kroků

metody *DTW* je prostá: použít mapování časových os pro hladké obrazy nehladkých dat. Výpočet mapovací funkce probíhá na datech vyhlazených do zvolené úrovně a po určení průběhu mapovací funkce jsou mapována nevyhlazená data. Zavedením tohoto kroku byla výrazně zvýšena numerická stabilita metody a testování ukázalo, že dobré výsledky lze získat zpravidla pro velký rozsah nastavení parametru úrovně vyhlazení. Při příliš vysoké úrovni vyhlazení však hrozí slití nebo úplné odstranění signifikantních komponent časového průběhu.

## 4.2 Ohraničení a určující parametry

Základním požadavkem při průměrování ERP dat je, aby se shodovaly počáteční a koncové body  $v(i)$  a  $s(j)$ . Druhým požadavkem je spojitost funkcí  $i(k)$  a  $j(k)$  a jejich neostrá monotónnost (mapovací funkce musí být neklesající) z bodu (1,1) do bodu (I,J). Ostatní vlastnosti mapovací funkce nejsou předurčeny a proto je lze měnit volbou hodnot jednotlivých parametrů.

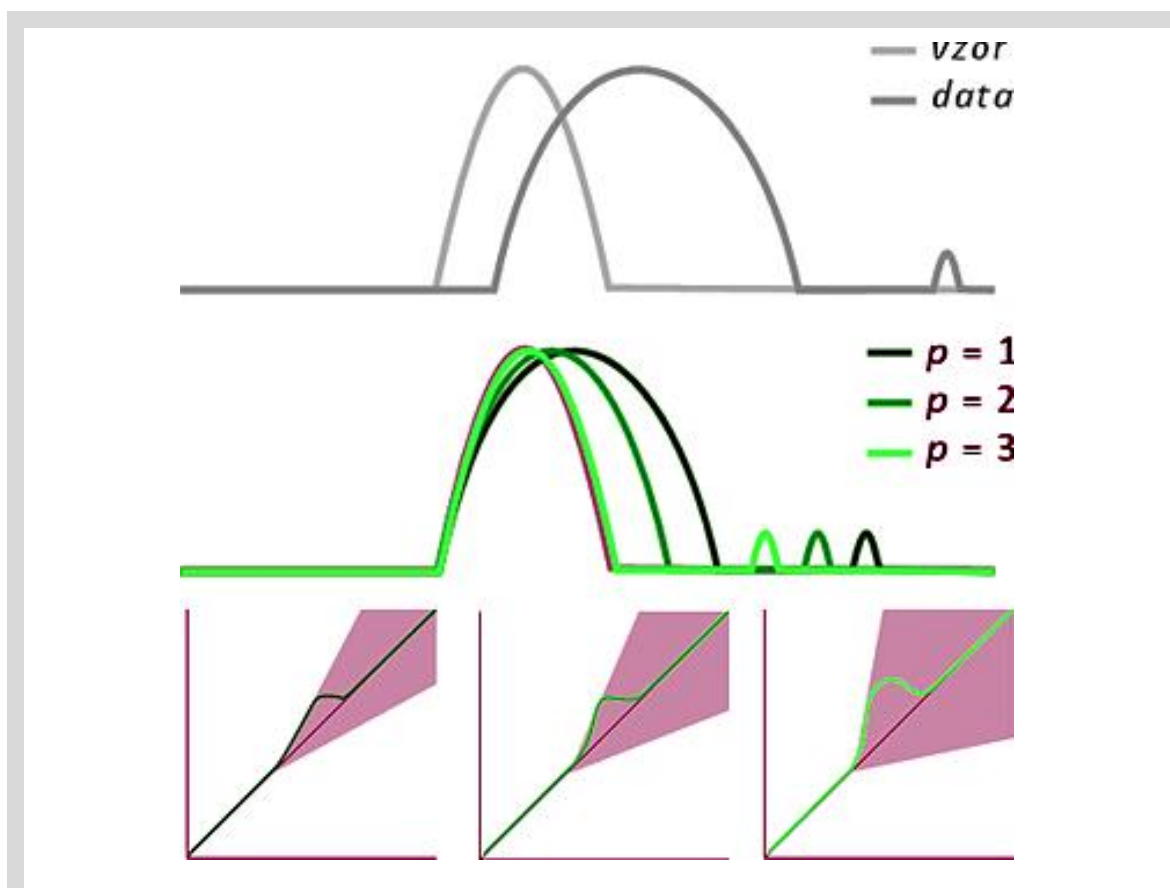
### 4.2.1 Váhy složek metriky

Optimální mapovací funkce minimalizuje vzdálenosti mezi dvěma časovými řadami, proto je třeba věnovat zvláštní pozornost stanovení metriky. Ta by měla odrážet ty vlastnosti dat, které chceme při průměrování zohledňovat, tedy v praxi různou váhu hodnot amplitudy (koeficient  $k$ ) a strmost (čili první derivace, koeficient  $l$ ) ve vzorci pro výpočet metriky (3). Ačkoli tyto koeficienty nejsou parametrem v pravém slova smyslu, jejich volba může výslednou podobu mapovací funkce ovlivnit. Literatura uvádí zpravidla hodnoty  $k = 0,5$ ,  $l = 0,5$  nebo  $k = 1$ ,  $l = 0$  (ignorování hodnoty 1. derivace). V jednom případě byl popsán experiment prováděný se třemi kombinacemi parametrů a výběrem té, která poskytuje nejlepší výsledky, viz [4].

Základní nastavení v implementaci metody je  $k = 0,5$  a  $l = 0,5$ ; doporučený rozsah je  $k = 0 \dots 1$  a  $l = 0 \dots 1$ , vždy tak aby  $k + l = 1$ .

#### 4.2.2 Parametr omezení délky kroku ve směru osy (point drift constraint)

V [5] je zaveden parametr  $p$  (angl. *point drift constraint*, omezení velikosti unášení v bodě), který určuje lokální omezení maximální délky expanze nebo kontrakce časové osy. Tento parametr je definován jako poměr mezi počtem kroků v diagonálním směru a počtem kroků v podélném nebo svislém směru (přesněji kolik kroků vodorovně nebo svisle je možné projít, než je povinně zařazen diagonální krok), a tak určuje možnou velikost elasticity mapování časových os. Vliv velikosti hodnoty parametru  $p$  ilustruje Obrázek 10.



Obrázek 10: ilustrace vlivu parametru *point drift constraint*.

Obrázek ilustruje vliv parametru  $p$  (*point drift constraint*) na mapování časových os. Čím vyšší je hodnota parametru, tím větší je elasticita mapování. Grafy ve spodní řádce ilustrují maximální přípustnou strmost průběhu mapovací funkce. Příliš vysoká hodnota parametru může způsobit nestabilitu (viz testy).

V případě takové cesty maticí, která by ukazovala na požadavek namapování více hodnot do jednoho bodu (posloupnost více kroků v jednom směru, až do délky

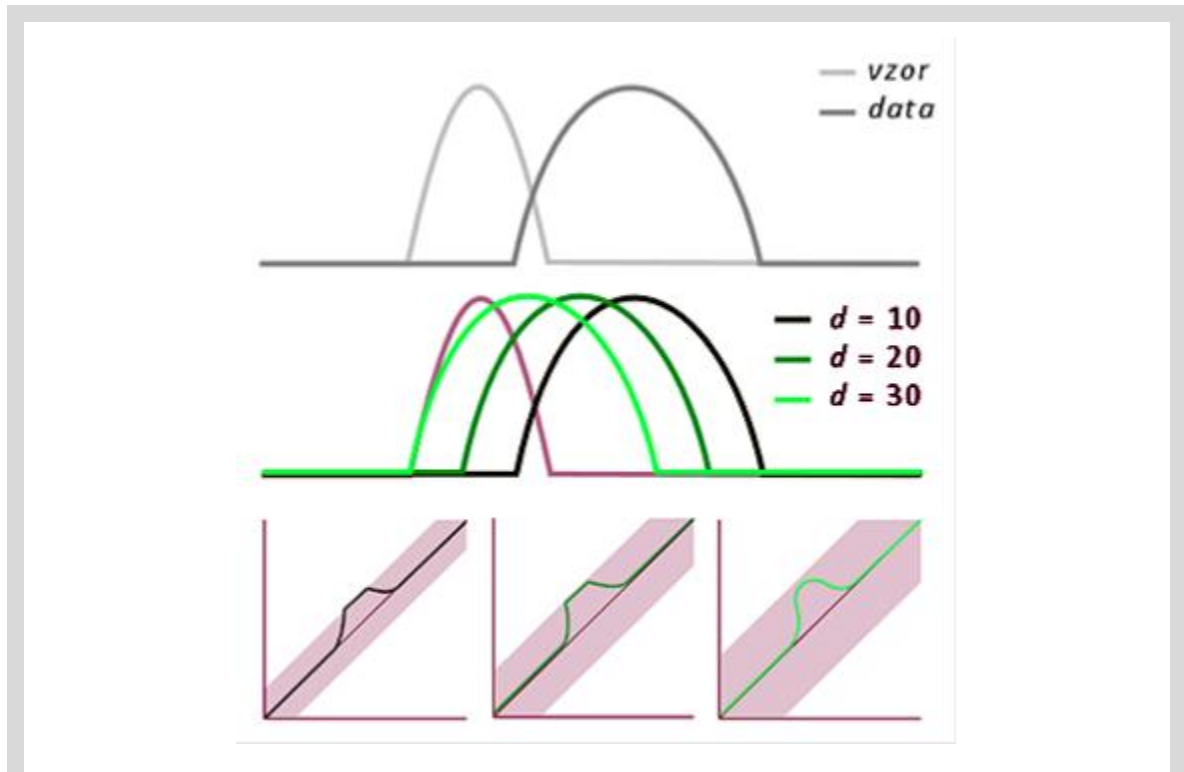
určené parametrem  $p$ ) je třeba určit způsob namapování těchto hodnot. Literatura se touto otázkou zabývá okrajově, používá se buď aritmetický průměr všech hodnot, vážený průměr s předem danou vahou jednotlivých bodů, případně nejvyšší nebo nejnižší hodnota.

V literatuře je obvyklá hodnota  $p = 1$ , příp.  $p = 2$ , vyšší hodnota se nedoporučuje. V implementaci metody je základním nastavením  $p = 1$ . Hodnota  $p = 0$  způsobí lineární mapování, tedy mapování dat na sebe a ne na vzor. Kapitola 6 ukazuje možnou nestabilitu při volbě příliš vysoké hodnoty parametru  $p$ .

#### **4.2.3 Parametr omezení vzdálenosti hodnoty mapovací funkce od diagonály (diagonal drift constraint)**

Dalším možným ohraničením volnosti mapování je parametr  $d$  (angl. *diagonal drift constraint*, omezení vzdálenosti hodnot mapovací funkce od diagonály), který slouží jako globální ohraničení hodnoty posunu hodnoty mapovací funkce od diagonály v rovině  $(i,j)$ . Důvodem pro zavedení parametru  $d$  je zřejmá skutečnost, že některé body v rovině  $(i,j)$  jsou navzájem vzdáleny příliš na to, aby jejich vzájemné namapování mohlo dát dobrý výsledek. Grafickou představu parametru  $r$  ukazuje Obrázek 11.

Při vhodné volbě parametru *point drift constraint* a *smoothing amount* dat lze volit hodnotu parametru *diagonal drift constraint* vysokou (v kapitole 6 byl použit limit o velikosti  $2/3$  délky epochy bez negativního vlivu na výsledek). Omezení velikosti tohoto parametru je vhodné především pro zrychlení výpočtu (lze násobně zkrátit délku výpočtu) a při hledání optimálních hodnot ostatních parametrů (především parametru *point drift constraint* a *smoothing amount*).



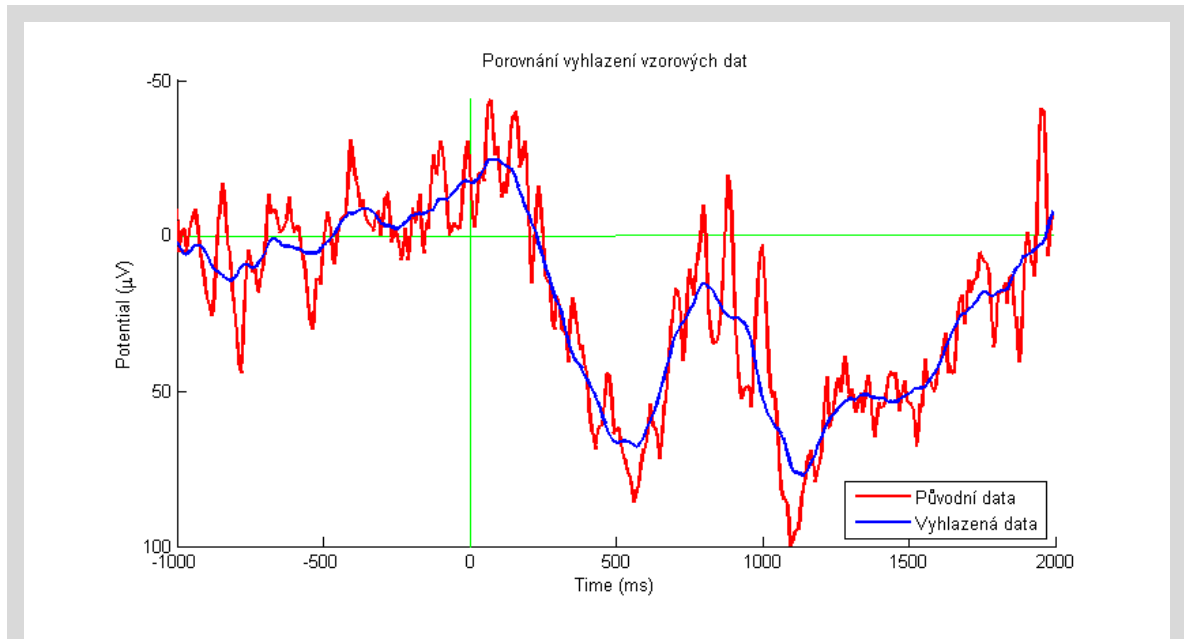
Obrázek 11: ilustrace vlivu parametru *diagonal drift constraint*.

Obrázek ilustruje vliv parametru  $d$  (*diagonal drift constraint*) na mapování časových os.

Čím vyšší je hodnota parametru, tím větší je přípustný posun celých spojených částí průběhu. Grafy ve spodní řádce ilustrují odchylku od diagonálního průběhu mapovací funkce. Příliš vysoká hodnota parametru může způsobit nestabilitu (viz testy).

#### 4.2.4 Parametr intenzity vyhlazení vzorových dat (smoothing amount)

Vyhlazením dat pomocí klouzavého průměru s vhodnou velikostí průměrovacího okénka je možné potlačit šum a zvýraznit signifikantní komponenty epochy. I když jsou následně mapována nevyhlazená data, distanční matice a mapovací funkce jsou vypočteny na základě vyhlazených dat, která zvýrazňují signifikantní extrémny. Velikost průměrovacího okénka je třeba volit s ohledem na míru detailu, která je od metody vyžadována, a podle poměru signálu a šumu (pro příklad viz Obrázek 12).



Obrázek 12: ilustrace vyhlazení zdrojových dat pomocí klouzavého průměru.

Červená linka (*Původní data*) ukazuje záznam tak, jak byl pořízen, modrá linka data vyhlazená klouzavým průměrem s velikostí okénka 35 snímků, tedy přibližně 10% délky celého snímku. Takto vyhlazená data zvýrazňují signifikantní komponenty v epoše a umožňují přesnější namapování nevyhlazených dat.

Vhodná hodnota tohoto parametru záleží na různých vlastnostech signálu (odstup signálu od šumu, periodičita šumu, hladkost průběhu signálu) a nelze jednoznačně doporučit univerzální přesnou hodnotu nebo rozsah hodnot. V rámci práce byly dosaženy dobré výsledky s velikostí okénka 5% až 25% celkové délky snímku pro zašuměná data a bez vyhlazování pro data s hladkým průběhem.

### 4.3 Algoritmus

1. Nejprve se stanoví vzor, na který se budou mapovat data. Vzor lze vytvořit synteticky, průměrem všech dat, nebo náhodným výběrem snímku. Volba vzoru může ovlivnit získaná výsledná data v závislosti na volbě parametrů  $p$  a  $r$ : pokud jsou jednotlivé lokální extrémy snímků příliš vzdáleny od lokálních extrémů vzoru a hodnoty parametrů příliš nízké, budou získané průměrné hodnoty nižší vlivem omezení průběhu funkce  $F$ .



2. Vytvoří se vyhlazená vzorová data.
3. Aktuální snímek je normalizován a jsou vypočteny numerické hodnoty první derivace v každém bodě.
4. Podle zvolené metriky se vypočítá distanční matice pro vzor a aktuální snímek z vyhlazených vzorových dat.
5. Za první mapovací bod se označí ( $i = 1, j = 1$ ) a následuje cyklus.
6. V každém kroku cyklu:
  - a. Je-li dosaženo okrajové hodnoty určené některým z omezujících parametrů, inkrementuje se  $i$  a  $j$  (cestou v matici se pokračuje diagonálně).  
Bylo-li předchozí pokračování v podélném směru, provede se příslušný přepočet hodnoty mapované do předchozího bodu.
  - b. Je-li dosaženo maximální hodnoty  $i$  nebo  $j$  inkrementuje se  $j$ , resp.  $i$ .  
Bylo-li dosaženo maximálních hodnot  $i$  i  $j$ , cyklus končí.
  - c. V ostatních případech se vybírá další směr dalšího kroku jako souseď s nejnižší hodnotou a tak, aby byly splněny požadavky na souvislost a monotónnost. Podle vybraného směru se buď inkrementuje  $i$  (podélný krok) nebo  $j$  (svislý krok) nebo obě proměnné (diagonální krok). Je-li vybrán podélný směr, je třeba uložit do pomocné proměnné (příp. pole) hodnotu k následnému přepočtu. V případě nerozhodnosti se v zájmu stability výpočtu preferuje diagonální krok.
7. Pokud je ve frontě na zpracování další snímek, uloží se aktuální vypočtené hodnoty a pokračuje se bodem 2, jinak se pokračuje bodem 8.
8. Vypočítá se aritmetický průměr ze všech namapovaných snímků.
9. Konec.

Popsaný algoritmus zahrnuje pouze vlastní zpracování dat, příprava (filtrování, stanovení základní úrovně signálu apod.) ani detaily prezentace výsledků nejsou záležitostí této práce.

## 5 Implementace

---

Tato část se věnuje druhé části druhého bodu zadání práce: implementace metody ve vhodně zvoleném jazyce. Vzhledem k požadavkům na implementaci se volba jazyka a implementačních detailů neomezila jen na jazyk samotný, ale byly uvažovány i vlastnosti různých prostředí, do kterých bylo možné implementovanou metodu začlenit. Implementace jako součást výpočetního prostředí pro vyhodnocování EEG dat s sebou nese výhody již připravených datových a prezentačních vrstev, bylo však třeba pečlivě uvážit i nevýhody a vlastnosti prostředí, která nejsou přímo spojena s implementací.

### 5.1 Volba programovacího jazyka a prostředí

Prvním krokem implementace je výběr vhodného programovacího jazyka, případně výpočetního prostředí. V této části práce budou popsány přednosti a nevýhody tří možných implementací: samostatného programu v jazyku Java nebo C, zásuvného modulu do programu *JERPA*, a zásuvného modulu do programu *EEGLAB*. Jednotlivé možnosti se výrazně liší především v možnostech datové vrstvy a možnostech prezentace výsledků, volba byla učiněna také s ohledem na možnou výraznou minimalizaci výsledného objemu nového zdrojového kódu.

#### 5.1.1 Samostatný program

Implementace metody jako samostatného programu nabízí největší volnost při implementaci a možnost největší výkonové optimalizace. Jako samostatné programy v jazyce C byly provedeny první zkušební implementace pro ověření základní funkčnosti metody. Tyto implementace byly velmi výkonné, především díky velmi dobrým možnostem paralelizace (lze paralelně provádět mapování jednotlivých snímků velkých datových souborů a vyvažovat zatížení jednotlivých vláken systémem farmer-worker).

Pro vlastní implementaci kompletní navržené metody včetně klouzavého průměru bylo upuštěno od implementace jako samostatného programu především z důvodů komplikovanosti načítání dat a zobrazení výsledků a náročné implementace těchto zásadních součástí. I při implementaci v jazyce Java a využití knihovny *JUIGLE* [6], která by mohla výrazně zjednodušit program, by nemohly být splněny požadavky neuroinformatické skupiny KIV na pohodlnost zpracování. Ukázalo se, že zapojení samostatného programu do řetězu činností při vyhodnocování dat by bylo na překážku efektivnosti a uživatelské přívětivosti celého řešení.

### 5.1.2 Zásuvný modul do programu JERPA

*JERPA* (angl. *Java Event-Related Potential Analysis*, [6]) je program pro vyhodnocování evokovaných potenciálů vyvíjený neuroinformatickou skupinou KIV v rámci výzkumných aktivit. Vychází z programu *JERP Studio*, který na katedře vyvinuli mj. kolegové Řondík, Soukal a Kučera. *JERPA* je program moderní architektury s důrazem na možnost dlouhodobého vývoje a údržby, jako součást rozsáhlé diplomové práce jej vytvořil kolega Souhrada v roce 2010, a je stále udržován a vyvíjen [6].

Program je dobře vybaven pro základní zobrazování a manipulaci s daty i pro vkládání zásuvných modulů. Implementace v jazyce Java umožňuje paralelizaci výpočtu.

Hlavními nevýhodami byly omezené možnosti pokročilého zobrazování výsledků (např. víc průběhů v jednom grafu), komplikovanost implementace zásuvného modulu oproti programu *EEGLAB* a malá rozšířenost programu. Ačkoli je *JERPA* živým a udržovaným programem, bylo po konzultacích rozhodnuto neimplementovat metodu DTW jako jeho zásuvný modul, a to především kvůli převažujícím výhodám následující možnosti.

### 5.1.3 Zásuvný modul do programu EEGLAB

Program *EEGLAB* je vyvíjen v centru *Swartz Center for Computational Neuroscience* na *University of California San Diego*, je k dispozici zdarma a je implementován jako toolbox v prostředí *Matlab*. Má velmi dobrou podporu zásuvných modulů,

je k dispozici velmi podrobná dokumentace a prostředí *Matlab* je optimalizované pro maticové výpočty na velmi rozsáhlých datech.

Implementace jako zásuvný modul pro *EEGLAB* má řadu nevýhod. Hlavní nevýhodou je nutnost získání licence pro *Matlab*, ve kterém se *EEGLAB* spouští jako tzv. *toolbox* (*EEGLAB* samotný je k dispozici zcela zdarma). Znamená to také, že pro běh vlastního *EEGLABu* je třeba nejdříve spustit *Matlab*, který zabere v paměti několik set MiB. Další nevýhodou je obtížná paralelizace, která je spojena s dalšími licenčními náklady na paralelizační *toolbox*. Poslední z řady nevýhod je nutnost implementovat alespoň část zásuvného modulu v proprietárním jazyce *Matlabu* (*Matlab* samotný umí využívat knihovny napsané v jazyce Java, ale minimálně grafické a datové rozhraní pro *EEGLAB* musí být implementováno jako tzv. *m-file*, tedy v jazyce *Matlabu*).

*EEGLAB* je ve vědecké komunitě standardem pro zpracovávání EEG záznamů v rámci výzkumu a vývoje. Je pro něj k dispozici řada zásuvných modulů a ve své základní formě implementuje i metodu *ICA* popsanou výše. Je tedy plnohodnotnou alternativou programu *JERPA*. Katedra má k dispozici licence *Matlabu* pro své zaměstnance i studenty, výrobce programu nabízí studentům licence za dostupné ceny. *EEGLAB* je navržen pro velmi snadné vytváření zásuvných modulů, nabízí dokonce podporu pro grafické rozhraní modulů. Všechna data (proměnné programu) jsou k dispozici ve výchozím prostředí *Matlabu*, takže je možné s nimi provádět i operace, které nejsou v programu implementované. *EEGLAB* je většinou psaný v nativním jazyce (zmíněné soubory *m-file*), a proto je možné využívat všechny jeho části a efektivně zapojit zásuvný modul do řetězce činností při zpracování dat. Pro detailní popis viz následující kapitolu.

I přes výše vyjmenované nevýhody byla zvolena možnost implementace jako zásuvného modulu programu *EEGLAB*. Následuje popis implementačních detailů.

## 5.2 Implementace zásuvného modulu

*Matlab* umožňuje použití vlastního jazyka a několika dalších jazyků (jazyk Java na vysoké úrovni propojení, jazyky C, C++ a Fortran na úrovni volání procedur). Výhodou použití proprietárního jazyka *Matlabu* je především možnost přímého přístupu ke všem datovým strukturám bez potřeby konverze a možnost velmi dobré a přehledné integrace zásuvného modulu. Především vícerozměrná pole a specifický způsob předávání argumentů a volání funkcí (viz dále). I v případě implementace v jiném jazyce musí být propojovací vrstva napsaná v jazyce *Matlabu*. Dvě hlavní nevýhody použití vlastního jazyka jsou jeho specifická, spojená s neobvyklými programátorskými zvyklostmi, a již zmíněná komplikovaná paralelizace (program v jiných jazycích lze paralelizovat prostředky jazyka bez potřeby paralelizačního nástroje *Matlabu*). Vlastnosti jazyka a obvyklé postupy, nutné pro rychlou orientaci ve zdrojovém kódu vytvořeného zásuvného modulu, jsou popsány v následujícím odstavci.

### 5.2.1 Specifika jazyka a programu EEGLAB

Jazyk *Matlabu* je optimalizovaný pro rychlé úpravy programu a výpočty nad maticemi (reprezentace maticí se používá i pro pole, znakové řetězce apod.) Programový kód je překládán technikou *Adaptive Just in Time*, uživatel proto pracuje pouze se zdrojovým kódem a aplikace zajišťuje překlad v okamžiku zavolání funkce. Funkce jsou ukládány každá v jednom souboru s příponou *.m* a názvem souboru shodným s názvem funkce.

#### *Zpětné volání*

Obvyklým způsobem volání funkce je tzv. *callback*, tedy zpětné volání funkce pomocí interpretace textu jako programového kódu. Tento přístup umožňuje např. velmi efektivní sestavování konstrukcí zachycení výjimek, kdy je možné připravit unifikované chování prostředí pro všechna volání bez potřeby kaskádování funkcí, viz následující příklad v pseudokódu:

```
try_strings := "volání před funkcí";  
catch_strings := "volání pro zachycení a zpracování výjimky";  
function_call := "zavolání funkce s parametry";
```

```
callback_text = [ try_strings, function_call, catch_strings ];  
eval(callback_text);
```

Příklad ukazuje, jakým způsobem lze sestavit text příkazu pro zpětné volání. Proměnná `callback_text` obsahuje sloučený znakový řetězec, obsahující znakové řetězce se všemi voláními zapsanými v použitých proměnných (konkatenace řetězce pomocí uložení do pole je obvyklý způsob, který vychází z výše popsané obvyklé reprezentace znakových řetězců). Tato proměnná je pak předána funkci `eval`, která zajišťuje interpretaci znakového řetězce jako zdrojového kódu.

Celý program *EEGLAB* je postaven na systému zpětných volání, a implementace zásuvného modulu se touto praxí řídí. Tento způsob poskytuje velmi dobré možnosti pro dynamické programování, ovšem přináší s sebou problém obtížného ladění programu – případná chyba vždy nastane při volání funkce `eval`. Pro ladění lze použít další úroveň stejné techniky tak, že se do vyhodnocovaného řetězce vloží bloky pro zachycení a zpracování výjimky. Všechny datové sady *EEGLABu* udržují historii volaných příkazů, a pro jednoduchou aktualizaci do této historie stačí zkopírovat znakový řetězec předávaný funkci `eval`.

### *Integrace zásuvného modulu do programu EEGLAB*

Integrace zásuvného modulu je snadná a přímočará. Do podadresáře *plugins* v pracovním adresáři programu *EEGLAB* se umístí adresář s názvem zásuvného modulu, a do tohoto adresáře se uloží soubor s názvem *eegplugin\_<název\_modulu>*. Tento soubor už může obsahovat cokoli a je volán při spouštění *EEGLABu*. Obvyklým obsahem je vložení potřebných tlačítek a dalších položek do menu grafického rozhraní a vytvoření potřebných zpětných volání. Funkce s grafickým rozhraním (zpravidla první volané po stisknutí příslušného tlačítka v menu) je obvyklé pojmenovávat *pop\_<název\_funkce>*, předponou *pop\_* totiž v prostředí *EEGLABu* začínají názvy všech funkcí s grafickým rozhraním.

Všechny funkce volané zpětným voláním z menu musejí vracet dvě hodnoty (přesněji pole dvou hodnot): *EEG*, obsahující pracovní datovou sadu, a *LASTCOM*, obsahující text naposledy použitých volání.

### Grafické rozhraní

*Matlab* poskytuje prostředky k vytváření jednoduchých uživatelských rozhraní a prezentací grafů časových průběhů. *EEGLAB* tyto možnosti dále upravuje tak, že je možné navrhnout uživatelské rozhraní v unifikované podobě s celým programem. Grafické rozhraní se definuje dvěma třírozměrnými poli, jedno určuje geometrické rozvržení prvků na obrazovce a druhé popisuje samotné prvky (text, textové editační pole, zatrhávací políčko, tlačítko apod.) U všech prvků je možné definovat zpětné volání, to je zpravidla použito jen u stisknutelných tlačítek.

Rozmístění prvků je možné určit pouze poměrově (např. tlačítko v první třetině druhého řádku prvků grafického rozhraní) po jednotlivých řádcích. Poměrné rozdělení se automaticky přizpůsobuje velikosti okna, a při nastavení příliš malé velikosti okna uživatelem může dojít ke zkrácení popisků nebo textů prvků.

0,25	0,5	0,25
Nadpis		Tlačítko
Příliš dlouhý te...	Delší text se do prostoru vejde	Tlačítko
1		0,5
1	1	1

*Obrázek 13: Ilustrace geometrického rozdělení prvků grafického rozhraní.*

První řádek ukazuje poměrné rozmístění prvků, následující dva řádky (na šedém pozadí) způsob, jak jsou do jednotlivých částí řádku umísťovány prvky a jak je s prvky naloženo, když přesahují vymezený prostor. Čtvrtý a pátý řádek ukazují, jak jsou rozmístěny jednotlivé prvky při nedodržení součtu udaných poměrů na řádek.

Zásuvný modul implementuje kromě metody *DTW* i možnost vytváření souhrnných průměrů. Ačkoli je tato možnost součástí zásuvného modulu *ERPLAB*, její využití pro tuto práci se ukázalo jako nepraktické vzhledem k nekompatibilitě datových struktur.

Podrobné informace o instalaci a použití zásuvného modulu jsou uvedeny v příloze *Uživatelská příručka*.

## 6 Testování na experimentálních datech

---

Pro zhodnocení implementované metody a její praktické využitelnosti byly navrženy následující testy:

- *Programový test.* Testování rychlosti zpracování, paměťových nároků a dalších technických vlastností programu.
- *Funkční testování.* Testování funkčnosti programu a zpracování vybraných úloh na experimentálních datech. Pro funkční testování byly navrženy následující úlohy.
  - *Korekce konstantního posunu dvojice průběhů signálů.*
  - *Korekce nelineárního posunu dvojice průběhů signálů.*
  - *Korekce nelineárního posunu dvojice vyhlazených průběhů signálů.*
  - *Průměrování rozsáhlé datové sady.*

Vzhledem k tomu, že implementace metody se liší od metod popsanych v literatuře především v přístupu k určení mapovací křivky, bude se testování zaměřovat především na parametr určující vlastnosti této křivky. Výše jmenované testy byly zvoleny z toho důvodu, že na nich lze nejlépe popsat vliv parametru vyhlazování (velikost vyhlazovacího okénka, *smoothing*) na výsledná data.

Soubor experimentálních dat použitý pro testování je anonymní nahrávka s 32 kanály, 153 epochami a vzorkováním 128 Hz. Epochy začínají 1 sekundu před stimulem a končí 2 sekundy po stimulu. Testování využití metody pro porovnání reakcí na dva různé stimuly nebylo provedeno vzhledem k tomu, že v průběhu testování nebyla prokázána vhodnost použití metody pro dávkové použití. Testování se proto soustředilo na hledání vhodných použití metody a popis vlivu parametrů na její fungování. Všechny testy byly provedeny s parametrem odchylky od diagonály (*diagonal drift*)  $d = 200$  a stejnou vahou hodnoty úrovně signálu a její první derivace v daném vzorku ( $k = 0,5$ ;  $l = 0,5$ ). Odchylka od diagonály byla nastavena na dostatečně vysokou úroveň, aby mohl být ukázán vliv volby



parametru vyhlazování. Stejná váha hodnoty úrovně signálu a první derivace v bodě je popisována v literatuře buď jako jediná používaná nebo jako jedna z použitých kombinací, a při testování bylo s touto kombinací dosaženo nejlepších výsledků.

### 6.1 Programový test

Testování bylo provedeno na dvou stolních počítačích lišících se konfigurací tak, aby bylo možné odhadnout výkonnost programu na běžně používaných stolních počítačích. Pro vývoj a testování byl použit program *Matlab* ve verzi 7.7.0 (obchodní označení R2008b) a program *EEGLAB* ve verzi 10.2.5.8b (verze označená jako připravovaná pro dlouhodobou podporu).

Konfigurace testovacích sestav je následující:

- 1) *Počítač 1*: procesor Intel i5 750 se proměnnou frekvencí jádra 2,66 GHz až 3,2 GHz, velikost paměti 8 GB, pevný disk s proměnnou rychlostí otáčení 5 400 až 7200 ot.min<sup>-1</sup>, operační systém Microsoft® Windows®<sup>1</sup> 7 Professional (64 bitová verze). Index uživatelských zkušeností<sup>2</sup> pro procesor: 7,3; pro paměť: 7,5 a pro primární pevný disk: 5,9.
- 2) *Počítač 2*: procesor Intel T2400 s frekvencí jádra 1,83 GHz, velikost paměti 3 GB, pevný disk s pevnou rychlostí otáčení 5 400 ot.min<sup>-1</sup>, operační systém Microsoft® Windows® 7 Professional (32 bitová verze). Index uživatelských zkušeností pro procesor: 4,3; pro paměť: 4,7 a pro primární pevný disk: 4,3.

---

<sup>1</sup> Jméno výrobce a název operačního systému jsou registrované ochranné známky společnosti Microsoft Corporation, sídlící v USA. Obě použité verze systému byly laskavě poskytnuty společností Microsoft pro studijní účely.

<sup>2</sup> Jednotná metoda hodnocení výkonnosti počítače v systémech Microsoft Windows od roku vydání 2006. Vliv jednotlivých součástí počítače se hodnotí na stupnici 1 (malý výkon) do 7,9 (vysoký výkon, tato hodnota se postupně zvyšuje spolu s technologickým pokrokem ve vývoji součástek).

Pro testování byly využity následující dvě datové sady:

- 1) *Datová sada 1*, obsahující záznam 32 kanálů, 153 epoch a 384 vzorků v každé epoše. Epochy v sadě trvají 3 sekundy a vzorkovací frekvence je 128 Hz. Velikost souboru datové sady je 7,8 MB.
- 2) *Datová sada 2*, obsahující 3 kanály, 68 epoch a 3 000 vzorků v každé epoše. Epochy v sadě trvají rovněž 3 sekundy a vzorkovací frekvence je 1 kHz. Velikost souboru datové sady je 2,7 MB.

### 6.1.1 Paměťové nároky

Samotné spuštění prostředí *Matlab* vyžaduje 330 až 360 MB paměti, spuštěný toolbox *EEGLAB* přidává dalších 10 až 30 MB. Celkem spuštěný program využívá zpravidla 340 až 380 MB operační paměti. Různý objem využití paměti je dán různou konfigurací programu, spuštěnými toolboxy, a také prediktivním načítáním těch součástí, jejichž využití program považuje za pravděpodobné. Načtení *datové sady 1* si vyžádá dalších 33 MB paměti, načtení *datové sady 2* pak 16 MB.

*Tabulka 1: využití paměti*

Použitá datová sada	Datová sada 1	Datová sada 2
Krok algoritmu		
Výpočet derivací a příprava dat	143 MB	2,3 MB
Výpočet distanční matice	143 MB	2,3 MB
Nalezení optimální cesty maticí mapování	216 MB	3,5 MB
Dokončení mapování zdrojových dat	216 MB	3,5 MB

Tabulka ilustruje špičkové hodnoty využití paměti při zpracování dat metodou DTW. Pro výpočty jsou sestaveny 3 matice o velikosti  $m \times m$ , kde  $m$  je délka snímku. Hodnoty v tabulce představují pouze paměť obsazenou zásuvným modulem, pro skutečné obsazení je třeba připočítat zmíněné hodnoty obsazení paměti prostředím a programem. Hodnoty jsou uvedeny kumulativně, poslední řádek tedy ukazuje nejvyšší dosaženou hodnotu.

Tabulka 1 ukazuje špičkové hodnoty využití paměti samotným zásuvným modulem. Vzhledem k sekvenčnímu způsobu zpracování dat je možné tyto hodnoty označit za konečné pro celý výpočet (vždy po dokončení výpočtu jednoho snímku je paměť uvolněna, teprve pak se vypočítává další snímek). Hodnoty v tabulce dokládají odhadovanou kvadratickou paměťovou složitost. *Datová sada 1* obsahuje velmi podrobné data snímku (3 000 vzorků), a v takovém případě vyžaduje maximálně řád stovek MB paměti.

### 6.1.2 Časová náročnost výpočtu

Následující tabulky (tabulka 2, tabulka 3) ukazují srovnání času výpočtu mapování jednoho snímku v závislosti na zvolené hodnotě parametru  $d$  (přípustná vzdálenost funkční hodnoty mapovací funkce od diagonály). Z tabulek je patrné, že časová náročnost strmě roste s počtem vzorků na snímek. Vyšší vzorkovací frekvence totiž znamená i požadavek na vyšší hodnotu argumentu  $d$ , aby byla zachována stejná relativní vzdálenost přípustného mapování.

Především při testu na *počítači č. 2* bylo dosaženo nepraktických časů výpočtu při výpočtu na datové sadě s velkým počtem vzorků nebo snímků (s parametrem  $d = 200$  vychází výsledný čas výpočtu přes 20 minut pro *datovou sadu 1* nebo 53 minut pro *datovou sadu 2*). Vhodným řešením je snížení vzorkovací frekvence pomocí nativní funkce *EEGLABu*.

Tabulka 2: čas výpočtu na jeden snímek, počítač 1

Použitý počítač		Počítač 1					
Datová sada		Datová sada 1			Datová sada 2		
Krok	Hodnota $d$	50	100	200	50	100	200
Výpočet derivací a příprava dat		<b>0,04</b>	<b>0,04</b>	<b>0,04</b>	<b>0,18</b>	<b>0,21</b>	<b>0,20</b>
Výpočet distanční matice		<b>0,09</b>	<b>0,12</b>	<b>0,19</b>	<b>0,60</b>	<b>0,98</b>	<b>1,68</b>
Nalezení optimálního mapování		<b>0,10</b>	<b>0,14</b>	<b>0,20</b>	<b>0,68</b>	<b>1,06</b>	<b>1,76</b>
Dokončení mapování zdrojových dat		<b>0,10</b>	<b>0,14</b>	<b>0,20</b>	<b>0,68</b>	<b>1,06</b>	<b>1,76</b>

Tabulka ukazuje srovnání časů výpočtu pro jednotlivé datové sady a pro různou volbu parametru  $d$  (přípustná vzdálenost funkční hodnoty mapovací funkce od diagonály).

Všechny časy jsou uvedeny ve vteřinách.

*Tabulka 3: čas výpočtu na jeden snímek, počítač 2*

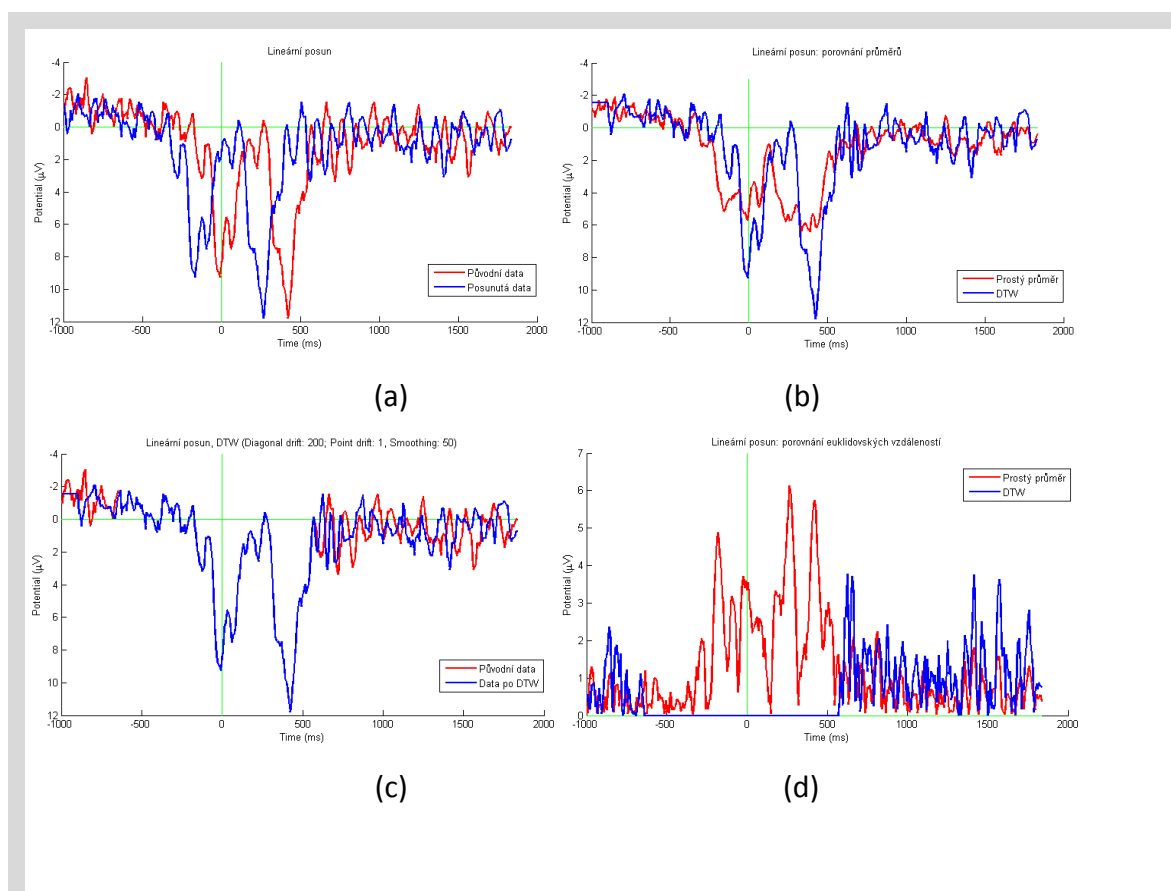
Použitý počítač		Počítač 2					
Datová sada		Datová sada 1			Datová sada 2		
Krok	Hodnota $d$	50	100	200	50	100	200
Výpočet derivací a příprava dat		<b>0,08</b>	<b>0,08</b>	<b>0,08</b>	<b>0,62</b>	<b>0,62</b>	<b>0,62</b>
Výpočet distanční matice		<b>0,26</b>	<b>0,42</b>	<b>0,59</b>	<b>1,96</b>	<b>3,35</b>	<b>5,89</b>
Nalezení optimálního mapování		<b>0,29</b>	<b>0,48</b>	<b>0,65</b>	<b>2,20</b>	<b>3,57</b>	<b>6,14</b>
Dokončení mapování zdrojových dat		<b>0,29</b>	<b>0,48</b>	<b>0,65</b>	<b>2,20</b>	<b>3,57</b>	<b>6,14</b>

Tabulka ukazuje srovnání časů výpočtu pro jednotlivé datové sady a pro různou volbu parametru  $d$  (přípustná vzdálenost funkční hodnoty mapovací funkce od diagonály).  
Všechny časy jsou uvedeny ve vteřinách.

## 6.2 Funkční testování

Prvním testem pro ověření očekávaného chování metody DTW byla zkouška korekce lineárního posunu. Pro test byl použit souhrnný průměr experimentálního datového souboru s vzájemným posunutím o 156 ms. Při časovém rozdílu v latencích vrcholů komponent větším, než perioda vrcholů šumu, nebyla metoda úspěšná bez vyhlazování vzorových dat. Především kvůli šumu, který měl velký vliv na počátek průběhu mapovací funkce.

S vyhlazením přes 50 vzorků byla metoda schopna spolehlivě korigovat rozdíl v latenci u výrazných amplitud signálu (špičky s latencí 0 a 400 ms).



**Obrázek 14:** Test funkčnosti programu na dvojici průběhů s konstantním posunem. Obrázek (a) ukazuje dvojici průběhů. Jedná se o totožný signál s posunem o 20 vzorků (156 ms). Obrázek (b) ilustruje pokles amplitudy průměru obou průběhů, (c) shodnost signifikantní části průběhu po aplikaci metody DTW s vyhlazováním přes 50 snímků. Obrázek (d) popisuje euklidovskou vzdálenost mezi průměrem posunutých dat a dat po úpravě metodou DTW.

### 6.2.1 Korekce nelineárního posunu dvojice průběhů signálů

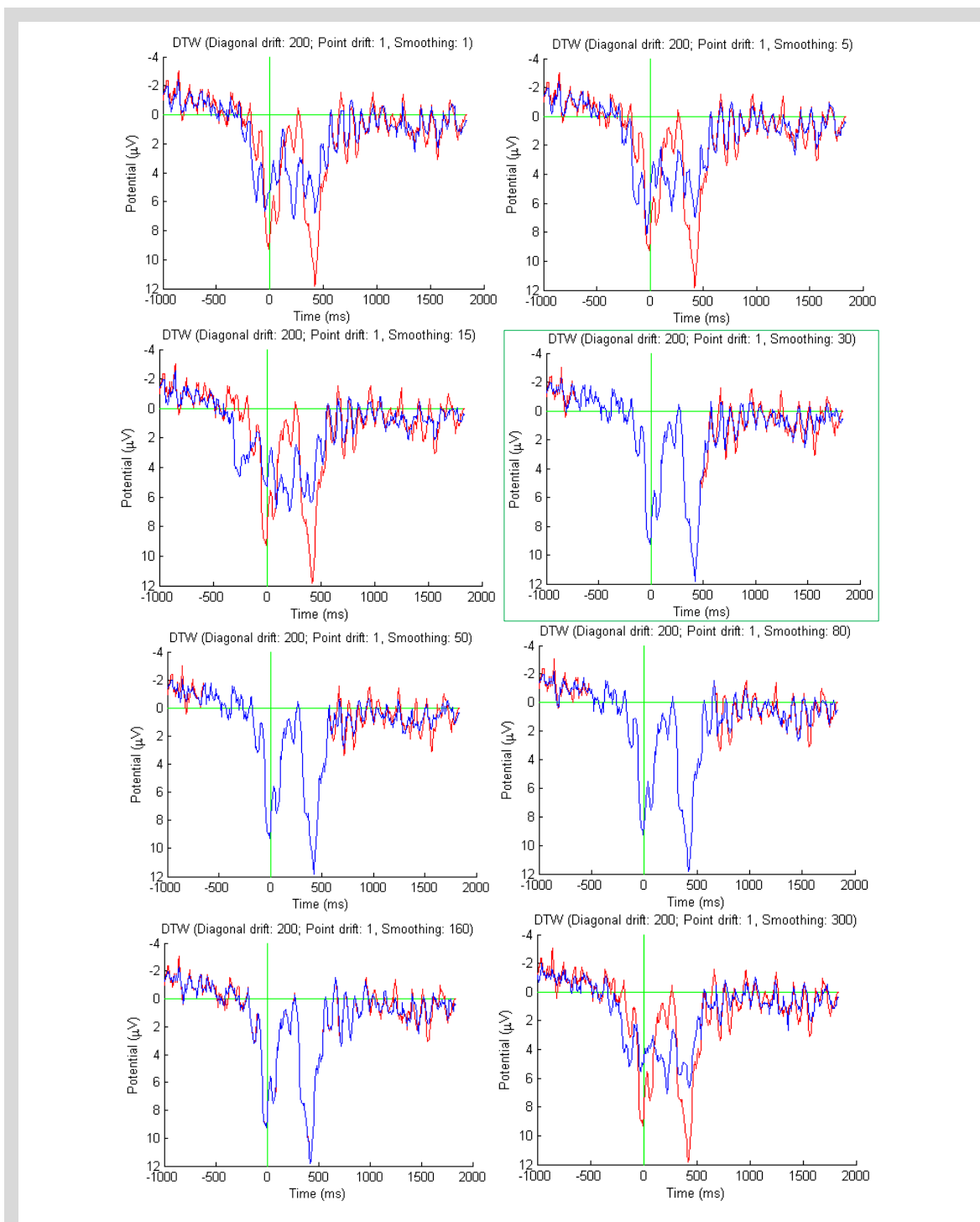
Tento test odpovídá požadavkům na volbu metody a zároveň dobře ilustruje vhodnost zavedení parametru vyhlazování vzorových dat. Bylo testováno několik různých kombinací průběhů signálu, dále bude popsán reprezentativní vzorek, který nejlépe ukazuje výsledky metody v závislosti na volbě parametrů.

Experimentální data byla namapována na časovou osu se sinusovým průběhem změny latence podle vzorce

$$\text{warping}(i) = i + 50 \sin\left(\frac{i - \text{start}}{80}\right), \quad (5)$$

kde proměnná *start* měla hodnotu 100. Při namapování dat pomocí tohoto vzorce došlo k žádanému nelineárnímu posunu a zároveň byla data uspořádána tak, že se zároveň kryla náběžná hrana obou komponent a v jiném časovém okamžiku se přesně kryje náběžná hrana jedné komponenty s opadovou hranou druhé (situaci ilustruje Obrázek 14). Tato data byla použita pro testování metody bez vyhlazování vzorových dat, přičemž nebylo dosaženo uspokojivého výsledku s žádnou testovanou kombinací parametrů (délka podélného nebo svislého kroku  $p = 1$  až 15, poměr vah hodnoty a první derivace s krokem 0,1).

Při použití vyhlazování vzorových dat bylo dosaženo uspokojivého výsledku v rozsahu parametru vyhlazování 30 až 160 snímků (velikost okénka 230 ms až 1250 ms). Takto vysoký parametr umožnil dobré namapování obou signifikantních komponent. Při největší z těchto hodnot bylo dosaženo výrazné shody v rozsahu od -150 ms do 1000 ms. Tuto velmi dobrou shodu je ovšem třeba připsat identičnosti výchozích dat, kdy metoda správně korigovala nelineární posun signifikantních amplitud i jejich okolí. Při porovnávání dvou odlišných průběhů lze očekávat dobré výsledky na dolní hranici uvedeného rozsahu. Na obrázku Obrázek 15 je ukázáno mapování lokálních výkyvů průběhu signálu po celé délce epochy právě s parametrem vyhlazování 30 vzorků. Hodnota parametr vyhlazení 300 téměř odpovídá třem čtvrtinám délky celé epochy, a proto s takto vysokým parametrem nejsou získány uspokojivé výsledky.



**Obrázek 15:** Vliv volby velikosti vyhlazovacího okénka na výsledek.

Grafy (zleva shora) zobrazují výsledky mapování pomocí metody DTW s velikostí vyhlazovacího okénka po řadě 1, 5, 15, 30, 50, 80, 160 a 300 vzorků. Nejlepšího výsledku bylo dosaženo s velikostí 30 (v zeleném rámečku), dobré výsledky byly získány s velikostí okénka až do 160 vzorků. Za pozornost stojí především první graf, který reprezentuje metodu DTW bez použití vyhlazování vzorových dat. V tomto případě metoda nedosáhla dobrých výsledků.

### 6.2.2 Korekce nelineárního posunu dvojice vyhlazených průběhů signálů

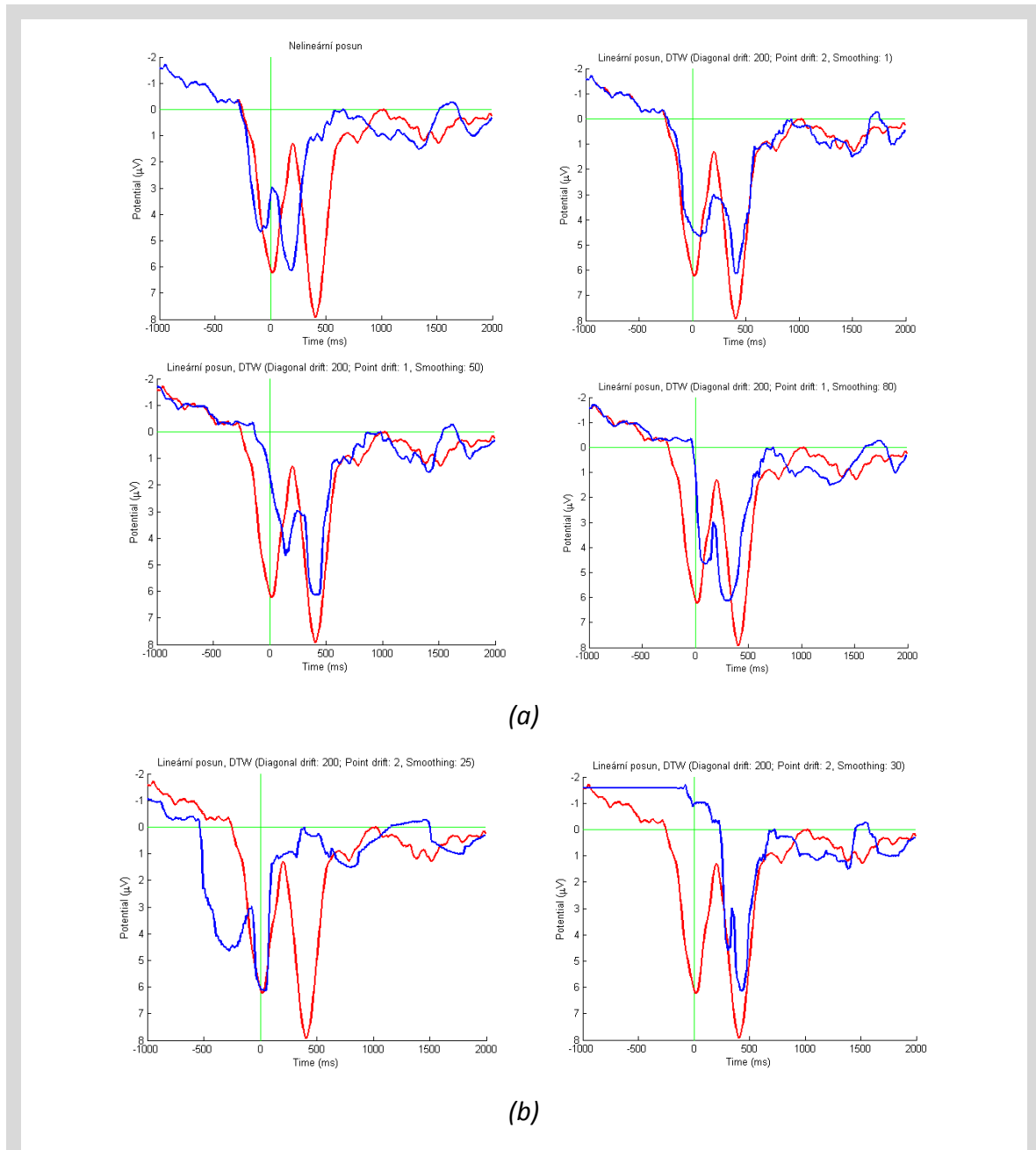
Tento test odpovídá předchozímu, byla použita stejná experimentální data, která byla pro potřeby testu vyhlazena tak, aby byl odstraněn šum ve frekvenční doméně. Pro vyhlazení byla použita velikost okénka 25.

Metoda byla opět testována s různými hodnotami velikosti okénka pro vyhlazení vzorových dat. Bez vyhlazení data přibližně odpovídají datům předchozího testu s velikostí okénka 30, se kterou bylo dosaženo nejlepších výsledků. V tomto případě byl testován vliv dvou parametrů: velikost testovacího okénka a délka vertikálního nebo horizontálního kroku.

Vzhledem k tomu, že data v tomto testu nebyla zatížena periodickým šumem (byl odstraněn vyhlazením), dosáhla metoda nejlepších výsledků bez použití vyhlazení vzorových dat. Vyhlazením došlo ke splnutí vrcholů obou komponent a tím i k zhoršení výsledků.

V testu byl dále předveden vliv zvýšení povolené délky vertikálního nebo horizontálního kroku (*point drift*) z 1 na 2. Zvýšení tohoto parametru při příliš vysoké hodnotě parametru vyhlazování může způsobit závažnou deformaci dat a nestabilní chování metody (kdy malá změna jednoho parametru způsobí nepřiměřeně velkou změnu ve výsledku). Tuto nestabilitu ilustruje Obrázek 16(b).





**Obrázek 16: Vliv volby parametru vyhlazování na výsledek.**

Obrázek (a) ilustruje zhoršení výsledků při volbě příliš velkého okénka pro vyhlazování na datech, která nejsou zatížena šumem. Graf vlevo nahoře ukazuje zdrojová data, vpravo nahoře výsledek metody DTW bez použití vyhlazení. Grafy v dolní řádce zobrazují výsledek při použití vyhlazení (velikost okénka 50, resp. 80).

Obrázek (b) ilustruje možnou nestabilitu metody, pokud jsou všechny tři řídicí parametry nastaveny na příliš vysokou hodnotu. Pro mapování byly použity velmi blízké hodnoty velikosti okénka pro vyhlazení (25, 30), přesto bylo určeno zcela odlišné mapování. Dlouhá vodorovná linka v grafu vpravo ukazuje na příliš vysokou hodnotu povolené odchylky od diagonály.

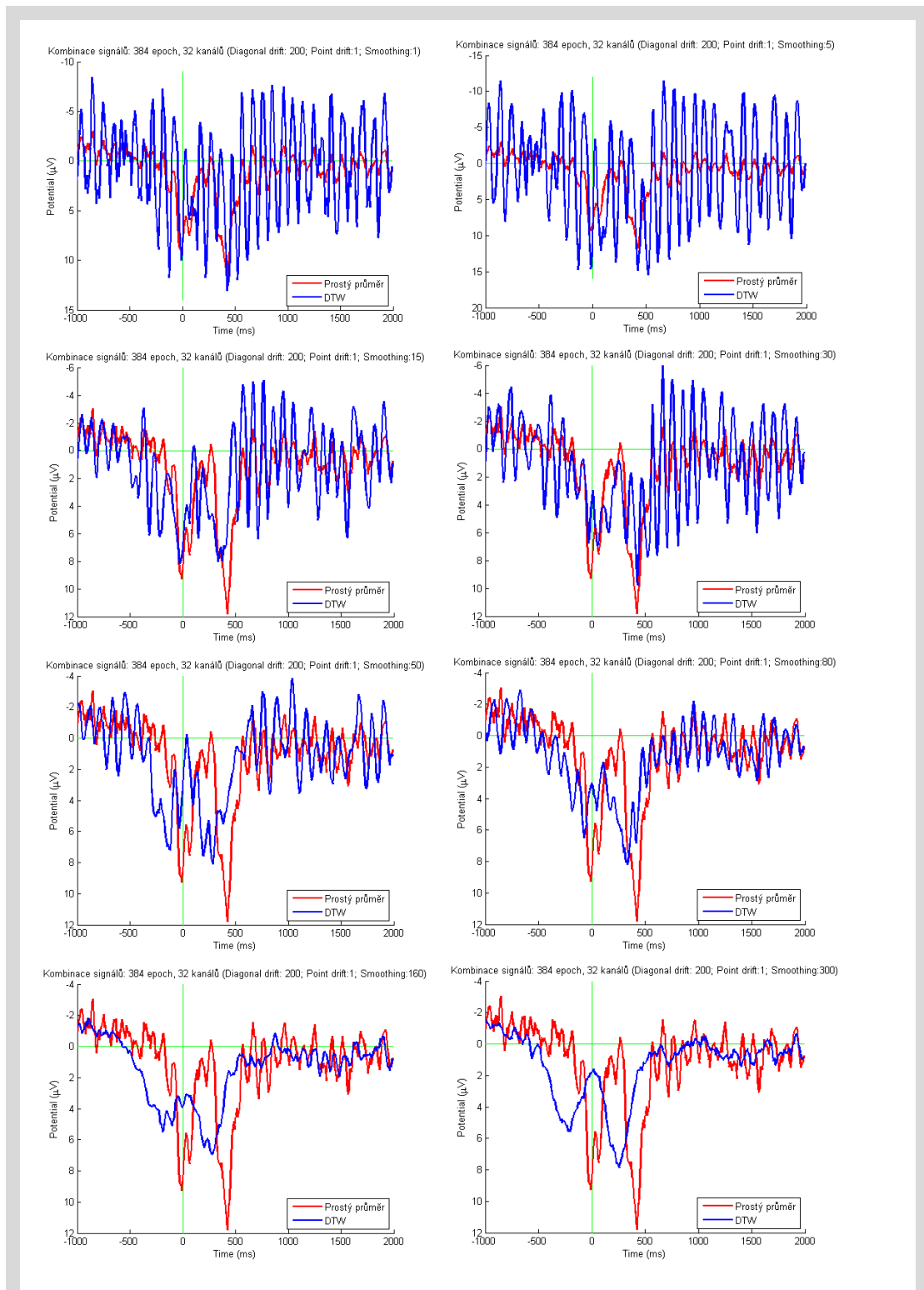
### 6.3 Průměrování rozsáhlé datové sady

Dávkové zpracování souborů s velkým množstvím naměřených dat bylo hlavním zamýšleným použitím metody. Při testování byly zjištěny dvě zásadní překážky pro toto využití: zpracování rozsáhlých datových souborů je časově velmi náročné, a metoda má tendenci zvýrazňovat šum ve frekvenční doméně. S prvním úskalím je spojen další problém: hledání vhodné kombinace parametrů. I při kvalifikovaném odhadu rozsahu je hledání zdlouhavé, protože zvolit vhodnou kombinaci je obtížnější s rostoucím množstvím dat.

Základní vlastností metody je snaha o korekci rozdílu latence na základě váhové funkce, která v různé míře zohledňuje amplitudu a její změnu (derivaci) v každém bodě. Během testování na rozsáhlém souboru bylo zjištěno, že vyšší váha derivace způsobuje vyšší citlivost na periodický šum, ovšem ani při úplném potlačení vlivu derivace na mapování docházelo ke zvýraznění periodického šumu.

Při použití metody DTW na dvojici průběhů bylo vhodným řešením problematického mapování dat zasažených šumem vyhlazení vzorových dat. Jak je patrné z výše uvedených příkladů, periodický šum byl zvýrazněn právě při volbě kombinace parametrů takových, že bylo docíleno dobrého namapování signifikantních komponent (viz Obrázek 17). Vzhledem ke skutečnosti, že periodický šum se zpravidla vyskytuje ve všech zaznamenaných datech, na rozdíl od evokovaných potenciálů, výsledná data mohou být metodou DTW úplně znehodnocena, což je v přímém rozporu s předpokladem zlepšení výsledků průměrování dat.

Tato vlastnost metody může být i prakticky využita, především pro detekci periodického šumu, případně pro jeho potlačení. Stimuly zpravidla nebývají se šumem synchronní, a proto tento šum v průměrovaných datech zaniká. Nabízí se tedy možnost využít metodu DTW obráceně: nejprve zvýraznit šum v souhrnném průměru, poté na tento průměr namapovat jednotlivé vzorky, a následně šum potlačit filtrem nebo odečtem normalizovaného šumu. Tato možnost je mimo rámec této práce a proto nebyla testována.



**Obrázek 17: Výsledky průměrování velkého počtu snímků s jedním nastavením parametrů pro celou sadu.**

Pokud jsou data zasažena periodickým šumem, má metoda DTW tendenci zvýrazňovat právě tento šum, a tedy snížit poměr signálu a šumu (což je v rozporu s požadavky na zlepšení výsledku průměrování dat).

## 7 Zhodnocení dosažených výsledků

---

Testování metody ukázalo následující zásadní skutečnosti:

- Při vhodné volbě parametrů je metodou DTW možno dosáhnout dobrých výsledků při korekci proměnlivého rozdílu časové latence dvou signálů.
- Vhodnou úpravou vzorových dat (vyhlazení s volenou velikostí průměrovacího okénka) lze dosáhnout dobrých výsledků i pro data obsahující šum a data, pro která je použití metody bez úpravy vzorových dat nevhodné.
- Výsledky metody jsou velmi závislé na vhodné volbě parametrů, především na velikosti průměrovacího okénka.
- Dávkové použití metody se jeví jako vhodné pouze pro menší počet snímků nebo pro data s vysokým poměrem signálu k šumu. U rozsáhlých souborů dat má metoda tendenci zvýrazňovat šum ve frekvenční doméně.

Použití metody bylo v literatuře popsáno vždy jako prostředek k zlepšení výsledků průměrování dat v konkrétní dané úloze, proto lze předpokládat, že průměrovaná data byla dostatečně homogenní a připravená pro zpracování s ohledem na danou úlohu (např. filtrování s velmi malým přípuštěným pásmem apod.) Pokud data neobsahují šum ve frekvenční doméně, lze vhodnou volbou parametrů dosáhnout dobrých výsledků. Parametr velikosti vyhlazovacího okénka pro vzorová data má největší vliv na výsledky metody, jeho volba závisí na vzorkovací frekvenci a požadavcích na výsledná data.

[14] popisuje získání dobrých výsledků při kombinaci metody DTW spolu s ručním určováním signifikantních vrcholů. Taková kombinace supluje volbu parametrů a na základě provedených testů lze odhadnout, že by měla pro průměrování přínos, ovšem její značná nevýhoda je nemožnost použití pro dávkové zpracování (uživatel musí ručně zvolit vrcholy).

Při testování možnosti využití metody DTW na rozsáhlejší soubory vznikla myšlenka využití této metody pro detekci a potlačení periodického šumu.

## 8 Závěr

---

Tato diplomová práce popisuje implementaci metody *Dynamic Time Warping* (*DTW*) jako prostředku pro zpracování záznamů evokovaných potenciálů (*ERP*). V úvodu poskytuje nezbytný teoretický základ pro pochopení požadavků na metodu, poté vyjmenovává metody popsáné v dostupné literatuře a shrnuje jejich možný přínos, předností a úskalí. Následuje obhajoba volby metody *DTW* a její detailní popis. Součástí popisu metody je i autorská úprava, která umožňuje výrazné zvýšení efektivity a použitelnosti metody. V dalších částech jsou popsány implementační detaily včetně volby vhodného programovacího jazyka a výpočetního prostředí, výsledky provedených programových a funkčních testů, na závěr je zhodnocen přínos metody a popsány vhodné možnosti využití. K práci je přiložena uživatelská a programátorská příručka zásuvného modulu programu *EEGLAB*, který byl vytvořen jako součást této práce. Struktura práce je přizpůsobena zadání tak, aby bylo možné snadno posoudit splnění jednotlivých požadavků.

Metoda implementovaná podle popisu v literatuře neposkytovala dobré výsledky, a nespĺnila očekávání založená na popisu dosažených výsledků. Na základě testů a dalšího zkoumání byla metoda upravena tak, že dokáže poskytnout přínosné výsledky na výrazně širší populaci vstupních dat. Ačkoli se při testech původní i upravené metody nepodařilo prokázat jednoznačný přínos metody *DTW* při zpracování velkých souborů dat, metoda byla shledána přínosnou pro menší a homogenní soubory záznamů nezátížených příliš velkým nebo periodickým šumem.

Pro implementaci byla zvolena forma zásuvného modulu programu *EEGLAB* v proprietárním jazyce prostředí *Matlab*. Tato volba znamenala zvýšené nároky v počáteční fázi implementace (studium nového jazyka, programovacích zvyklostí a programové struktury *EEGLABu*). Tyto vysoké nároky byly vyváženy přímočarostí implementace a faktem, že bylo třeba implementovat pouze metodu

samotnou a prostředky pro prezentaci výsledku. Vytvořený zásuvný modul dává uživateli kromě možnosti aplikace metody samotné i další možnosti průměrování, které v základním programu nejsou k dispozici.

## 9 Zdroje

---

- [1] SANEI, Saeid; Chambers, J.A. *EEG Signal Processing*. Chichester: John Wiley & Sons, Ltd, 2007. 313 stran. ISBN 13978-0-470-02581-9.
- [2] KAYSER. *ERP Averaging* (internetový zdroj). Akt. 29. 11. 2001 [cit. 10. 2. 2012]  
URL:<<http://psychophysiology.cpmc.columbia.edu/mmedia/General/auditoryERPaveraging.html>>
- [3] GIBBONS, Henning, STAHL Jutta, Response-time corrected averaging of event-related potentials. *Clinical Neurophysiology*, č. 118, roč. 2007, s. 197 – 208.
- [4] WANG, Konming, BEGLEITER Henri, PORJESZ Bernice, Warp-averaging event-related potentials. *Clinical Neurophysiology*, č. 112, roč. 2001, s. 1917 – 1924.
- [5] ASSECONDI, Sara et al, Automated identification of ERP peaks through Dynamic Time Warping: An application to developmental dyslexia. *Clinical Neurophysiology*, č. 120, roč. 2009, s. 1819 – 1827.
- [6] SVOBODA, Jaroslav. *Metody zpracování evokovaných potenciálů*. Plzeň, 2008. 59 stran. Diplomová práce, KIV FAV ZČU, vedoucí práce Pavel Mautner.
- [7] ŘONDÍK, Tomáš. *Metody zpracování ERP signálů*. Plzeň, 2010. 57 stran. Diplomová práce, KIV FAV ZČU, vedoucí práce Pavel Mautner.
- [8] SOUHRADA, Václav. *Integrace metod zpracování signálu do ERP softwaru*. Plzeň, 2010. 129 stran. Diplomová práce, KIV FAV ZČU, vedoucí práce Roman Mouček.
- [9] POLICH, John, Updating P300: An integrative theory of P3a and P3b. *Clinical Neurophysiology*, č. 118, roč. 2007, s. 2128–2148.

- [10] GROPPE, David M et al., Independent Component Analysis of Event-Related Potentials. *COGNITIVE SCIENCE ONLINE*, č. 6, roč. 2008, s. 1 – 44.
- [11] WOODY, Charles D., Characterization of an adaptive filter for the analysis of variable latency neuroelectric signals. *Medical and Biological Engineering*, č. 5, roč. 1967, s. 539 – 553.
- [12] AUNON, Jorge I., MCGILLEM Clare D., On the classification of single evoked potentials using a quadratic classifier. *Computer Programs in Biomedicine*, č. 14, roč. 1982, s. 29-40.
- [13] ACIR, Nurettin, Estimation of brainstem auditory evoked potentials using a nonlinear adaptive filtering algorithm. *NEURAL COMPUTING & APPLICATIONS*, 2012, s. 1 – 9.
- [14] VINCENT Alex, Methods for improving the signal-to-noise ratio of endogenous-evoked potentials, *INTEGRATIVE PHYSIOLOGICAL AND BEHAVIORAL SCIENCE*, č. 27, roč. 1992, s. 54-65.
- [15] KAPLAN, A.Y. et al, Nonstationary nature of the brain activity as revealed by EEG/MEG: Methodological, practical and conceptual challenges. *Signal Processing*, č. 85, roč. 2005, s. 2190-2212.
- [16] VLACHOS, Michalis. *A Practical Time-Series Tutorial with MATLAB*, příspěvek na konferenci *16th European Conference on Machine Learning & 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 3. – 7. 10. 2005, Porto, Portugalsko.  
URL:<[http://alumni.cs.ucr.edu/~mvlachos/PKDD05/PKDD05\\_Handout.pdf](http://alumni.cs.ucr.edu/~mvlachos/PKDD05/PKDD05_Handout.pdf)>



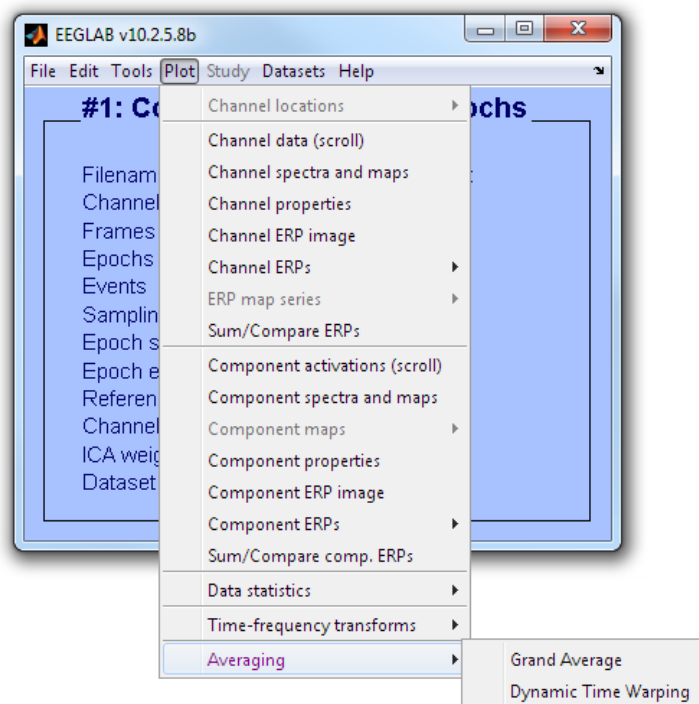
## **Příloha: uživatelská příručka**

---

## Instalace a spuštění programu

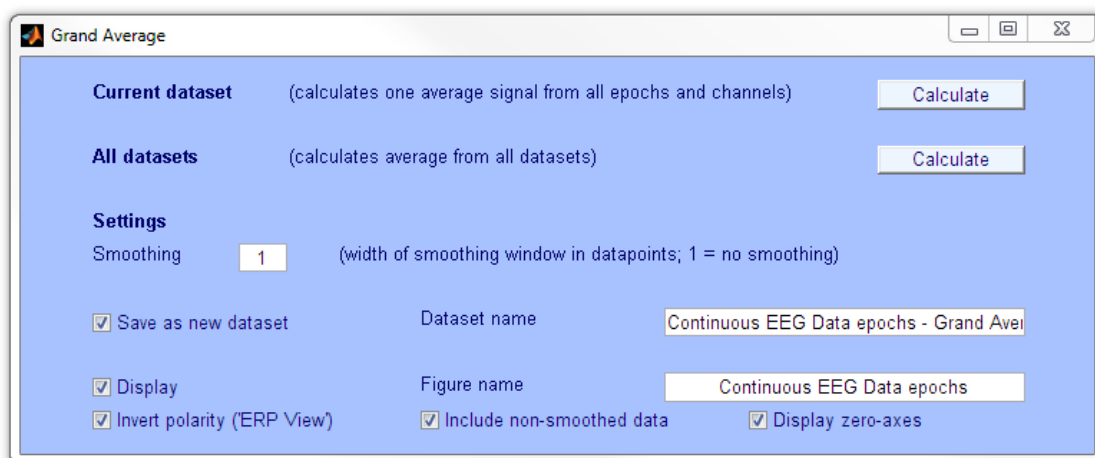
---

Zásuvný modul se instaluje prostým nahráním adresáře se zásuvným modulem (*plugin\_advanced\_averaging*, na přiloženém CD v adresáři *Program*) do podadresáře *plugins* v pracovním adresáři programu *EEGLAB*. Po spuštění programu a otevření alespoň jednoho *datasetu* je k dispozici nabídka *Averaging* v menu *Plot*, která obsahuje tlačítka *Grand average* a *Dynamic Time Warping*.



## Grand Average

---



Po stisknutí tlačítka *Grand Average* se otevře okno s ovládacími prvky pro volby parametrů a spuštění výpočtu. V dolní části je možné zvolit parametr úrovně vyhlazení (viz odstavec 4.2.4 diplomové práce), a volby zpracování výsledku:

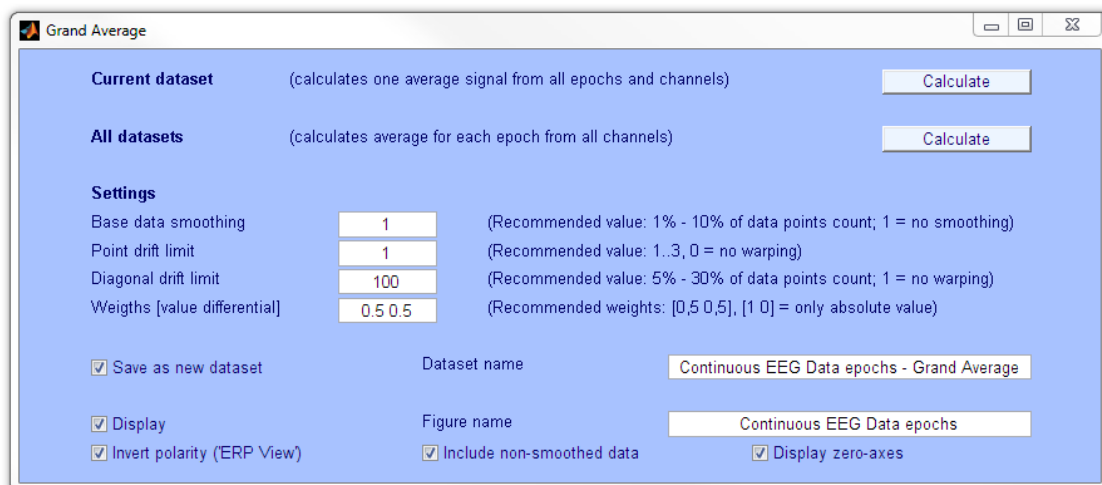
- *Save as new dataset* uloží vypočtený průměr jako nový *dataset* pod názvem, který je zadáný v textovém poli napravo od volby.
- *Display* zobrazí vypočítaný souhrnný průměr v okně, podle dalších nastavení (*Invert polarity* obrátí smysl svislé osy tak, jak je běžné při vyhodnocování ERP; *Include non-smoothed data* vloží do grafu i data bez vyhlazení tak, že jsou oba průběhy zobrazeny najednou; *Display zero-axes* do grafu vloží kříž soustavy souřadnic s počátkem v nule).

Je možné provést výpočet jak pro současný vybraný dataset (tlačítko *Calculate* na řádku *Current dataset*), tak pro všechny datasety nahrané do programu (řádek *All datasets*).

## Dynamic Time Warping

---

Po stisknutí tlačítka *Dynamic Time Warping* se otevře obdobné okno, které navíc umožňuje nastavení parametrů pro algoritmus metody DTW. Pro vysvětlení významu jednotlivých parametrů viz kapitolu 4.2 diplomové práce.



## **Příklad výpočtu průměru souhrnných průměrů metodou DTW**

---

Použití zásuvného modulu pro modelovou úlohu výpočtu souhrnných průměrů několika záznamů a jejich průměrování metodou DTW popisuje následující postup:

1. Otevření příslušných datasetů v programu EEGLAB (*File > Load existing dataset*).
2. Převzorkování na stejnou vzorkovací frekvenci (*Tools > change sampling rate*).
3. Segmentace dat na epochy (*Tools > extract epochs*). Doporučuje se nastavit minimální nebo nulový čas před stimulem, protože velmi rozdílné průběhy před zájmovou částí nahrávky mohou snížit efektivitu metody DTW. Pro každý zájmový stimul je třeba provést segmentaci zvlášť.
4. Vytvoření souhrnného průměru pro všechny datasety (*Plot > Averaging > Grand Average*, pak tlačítko *Calculate* u položky *All datasets*), s nastavením *Save as new dataset* (výsledný dataset bude automaticky uložen a vybrán).
5. Výpočet souhrnného průměru metodou DTW (*Plot > Averaging > Dynamic Time Warping*, pak tlačítko *Calculate* u položky *Current dataset*).
6. Pro výpočet souhrnného průměru pro další stimul znovu od kroku 3.