



Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra Kybernetiky

---

**Analýza pohybových dat znakového jazyka**  
Sign Language Motion Capture Data Analysis

---

**BAKALÁŘSKÁ PRÁCE**

*Autor:*  
Ondřej VALACH

*Vedoucí práce:*  
Ing. Pavel JEDLIČKA

# ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2020/2021

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ondřej VALACH**  
Osobní číslo: **A19B0672P**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Inteligentní komunikace člověk – stroj**  
Téma práce: **Analýza pohybových dat znakového jazyka**  
Zadávací katedra: **Katedra kybernetiky**

### Zásady pro vypracování

- Seznamte se s daty pořízenými technologií Motion capture (Český znakový jazyk).
- Proveďte rešerši metod pro porovnání vektorů a interpolaci dat.
- Proveďte analýzu poskytnutých dat a porovnejte vybrané metody.
- Seznamte se s algoritmem DTW (Dynamic Time Warping) a metodou vzájemné korelace signálů.
- Vyberte vhodné metody pro práci s poskytnutými daty.

Rozsah bakalářské práce: **30-40 stránek A4**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

- MADISETTI, V.K., WILLIAMS, D.B.: The Digital Signal Processing Handbook. CRC Press, 1998. ISBN 0-8493-8572-5
- Alena Macurová, Poznáváme český znakový jazyk, Speciální pedagogika 11, 2001, č.2 a č.5. (Dostupné na [www.ruce.cz](http://www.ruce.cz))

Vedoucí bakalářské práce: **Ing. Pavel Jedlička**  
Výzkumný program 1

Datum zadání bakalářské práce: **9. listopadu 2020**  
Termín odevzdání bakalářské práce: **24. května 2021**



---

**Doc. Dr. Ing. Vlasta Radová**  
děkanka

---

**Prof. Ing. Josef Psutka, CSc.**  
vedoucí katedry

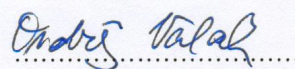
## Prohlášení

Překládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne

21. 5. 2021



Ondřej VALACH



## Poděkování

Chtěl bych poděkovat především vedoucímu práce Ing. Pavlu Jedličkovi za jeho podporu, cenné rady a trpělivost s řešením tématu a problémů při vývoji kódu a interpretaci výsledků. Také i za možnost tuto problematiku v rámci Bakalářské práce zpracovat. Dále bych chtěl poděkovat Ing. Miroslavu Jiříkovi, Ph.D. za návod a pomoc při zprovoznění komunikace dat na MetaCentrum a samotnému MetaCentru za možnost využít jejich rozsáhlé výpočetní infrastruktury.

Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

## Abstrakt

Znakový jazyk je jedním z nejdůležitějších komunikačních prostředků komunity neslyšících. Zprostředkování tlumočeného projevu není vždy dostupné. Jako řešení problému dostupnosti co největšího množství informací v jejich přirozené podobě pro Neslyšící se nabízí jeho automatická syntéza. Tato práce je součástí projektu, který se syntézou zabývá a pojednává o problému automatického určení podobnosti dvou projevů znakového jazyka. Během výpočtů je zpracováváno velké množství dat. Proto je potřeba měření této podobnosti plně automatizovat. I v případě, že projevy budou mít stejný význam, nemusí být stejně dlouhé. Je představen a testován algoritmus Dynamického borcení času, metody převzorkování dat a zjištění podobnosti, případně vzdálenosti dvou projevů ve znakovém jazyce. Cílem je získat kombinaci metod, jejíž výsledek by tuto vzdálenost, případně podobnost spolehlivě reprezentoval.

## Klíčová slova

Znakový jazyk, pohybová data, Dynamické borcení času, zpracování signálu, převzorkování, interpolace, korelace, vzdálenost, Python

## **Abstract**

Sign language is one of the most important communication ways for the deaf. Mediation of interpreted speech is not always available. As a solution to the problem of the availability of as much information as possible in its natural form for the Deaf, is its automatic synthesis. This thesis is part of a project, which deals with the synthesis and discusses the problem of automatically determining the similarity of two sign language signs. There is a large amount of data processed during the computations. This is the reason to fully automate the process. Even if two signs with the same meaning are tested, they will have probably different length. The thesis presents and tests the algorithm of Dynamic Time Warping, resampling methods, data interpolation and similarity detection, in the case of computing the distance between two time series. The aim is to obtain a combination of methods whose result would reliably represent this distance or similarity.

## **Keywords**

Sign Language, Motion Capture Data, Dynamic Time Warping, Signal Processing, Resampling, Interpolation, Correlation, Distance, Python

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Teoretická část</b>	<b>3</b>
2.1	Znakový jazyk . . . . .	3
2.2	Záznam dat . . . . .	5
2.3	Data . . . . .	9
2.4	Analýza dat . . . . .	11
2.5	Analyzované metody . . . . .	13
2.5.1	Algoritmus DTW . . . . .	13
2.5.2	Algoritmus SoftDTW . . . . .	17
2.5.3	Převzorkování . . . . .	18
	Fourierova transformace . . . . .	18
	Polynomiální interpolace . . . . .	19
2.5.4	Porovnání . . . . .	23
	Eukleidovská metrika . . . . .	23
	Hammingova metrika (Manhattanská) . . . . .	24
	Minkovského metrika . . . . .	25
	Čebyševova metrika . . . . .	25
	Mahalanobisova metrika . . . . .	25
	Canberrská metrika . . . . .	26
	Brayova-Curtisova metrika . . . . .	26
	Pearsonův korelační koeficient . . . . .	26
<b>3</b>	<b>Praktická část</b>	<b>27</b>
3.1	Validace použitých typů algoritmu . . . . .	27
3.2	Omezení dat . . . . .	28
3.3	Možnosti převzorkování . . . . .	32
3.3.1	Upsampling/downsampling . . . . .	32
3.3.2	Polynomiální interpolace . . . . .	33
3.3.3	Fourierova transformace . . . . .	35
3.4	DTW . . . . .	36
3.5	Testování metod . . . . .	36
3.5.1	Kombinace s lineární interpolací . . . . .	37
3.5.2	Kombinace s Fourierovou transformací . . . . .	38
3.5.3	Pearsonův korelační koeficient . . . . .	40
3.5.4	SoftDTW . . . . .	45
3.6	Výsledné zhodnocení . . . . .	46
3.7	Případová studie . . . . .	47
3.8	Výpočetní doba . . . . .	50
3.9	Pokračování ve výzkumu . . . . .	51
<b>4</b>	<b>Závěr</b>	<b>52</b>



# 1 Úvod

Katedra kybernetiky se již delší dobu zabývá syntézou řeči. Jejím cílem je převod psaného textu na výstup ve formě zvukové stopy, jedná se tedy o verbální složku komunikace. Neverbální forma komunikace, která je na rozdíl od zvuku vnímána vizuálně, má význam především pro komunitu neslyšících. Hlavním komunikačním prostředkem komunity neslyšících je znakový jazyk (dále ZJ). ZJ je založen na tvarech, pohybu a pozicích rukou v prostoru, proto se popisuje jako vizuálně motorický prostředek komunikace. Důležité jsou ovšem také pohyby hlavy, těla a mimika (více viz. kapitola 2.1). Stejně jako u řeči je i u ZJ možnost jeho syntézy. Tou se v tomto případě rozumí převod nějaké formy zápisu na výstup ve formě videa.

Samotná forma zápisu projevů v ZJ je problematická a přirozeným a také nejčastějším způsobem záznamu projevů v ZJ je videozáznam. Symbolický zápis ZJ je předmětem práce několika světových pracovišť, mezi nejrozsáhlejšími lze uvést například systém Ham-NoSys [1]. Dalšími notačními systémy jsou například Stokoeova notace [2] nebo systém SignWriting [3]. Jedna z možností syntézy ZJ je pomocí animovaného avatara [4]. Pohyb avatara je řízen kosterním modelem a daty pořízenými záznamem znakování, ta zajišťují přirozenost jeho pohybu. State-of-the-art technologie pro záznam pohybu je Motion Capture (také MoCap, viz. kapitola 2.2). Jejím výstupem jsou data obsahující informaci o trajektorii jednotlivých částí kosterního modelu během znakování (viz. kapitola 2.3). Přístup, který využívá tato data k syntéze ZJ se nazývá datově řízený přístup. Pravidlově řízený přístup, který v dnešní době převládá, funguje na základě symbolického popisu pohybů. Z něho je poté možno generovat animace, které odpovídají předpisům ZJ. Přístup ovšem ztrácí na přirozenosti pohybu, protože je generován z pevně daných pravidel a pohyby jsou velice strojové.

Hlavním požadavkem syntetizovaných pohybových dat je přirozenost výstupních pohybů avatara. Z toho důvodu je nutno přirozenost nějakým způsobem ohodnotit. Ohodnocení přirozenosti však nelze vyřešit pouze strojovou metodou, protože závisí na tom, jak je subjektivně vnímána člověkem. Důležitým předpokladem přirozenosti je podobnost syntetizovaných dat s daty reálnými, tuto podobnost lze již kvantitativně vyjádřit. Potřebujeme tedy objektivní metriku na nahrazení člověka pro účely strojového zpracování dat.

Tato práce se zabývá analýzou vhodných metod na porovnání dvou znaků ZJ reprezentovaných pohybovými daty. I v případě, kdy mají znaky stejný význam, se od sebe mohou mírně lišit svým provedením nebo rychlostí znakování. Problém porovnání nestejně dlouhých znaků řeší algoritmus Dynamického borcení času (také Dynamic Time Warping, dále DTW), který byl za tímto účelem přímo vyvinut (viz. kapitola 2.5.1).

Současným trendem strojového učení je použití neuronových sítí. Gradientní metoda zpětného šíření chyby, která je pro trénink neuronových sítí využívána, vyžaduje diferencovatelnou ztrátovou funkci. Algoritmus DTW tuto podmínku nesplňuje. Cílem práce je tedy najít jeho vhodnou diferencovatelnou alternativu. Tyto alternativy budou validovány s výsledky metody DTW.

Porovnání podobnosti dvou znaků bude většinou založeno na jejich vzdálenosti, pro zlepšení výsledku je v tomto případě požadována její minimalizace. Dalším možným způsobem vyjádření podobnosti může být vyjádření pomocí korelace, u které je naopak pro zlepšení výsledku vyžadována její maximalizace. Vzhledem k nesteré délce dvou porovnávaných znaků a tomu, že většina metod na porovnání signálů vyžaduje stejnou délku, je nutné data před využitím těchto metod připravit. Budou proto testovány různé metody převzorkování (více viz. kapitola 2.5.3).

Algoritmus DTW je jednou z metod schopných samostatně vyřešit problém porovnání nesteré dlouhých signálů. Jako další metoda může být uveden algoritmus SoftDTW, jehož rozdílem oproti DTW je modifikovaná ztrátová funkce s cílem její diferencovatelnosti (více viz. kapitola 2.5.2). Takto modifikovaný DTW algoritmus je v analýze práce testován také jako jedna z možných alternativ.

## 2 Teoretická část

### 2.1 Znakový jazyk

Jak již bylo zmíněno výše, ZJ je plnohodnotným komunikačním prostředkem komunity neslyšících. Je užíván jak Neslyšícími, tak i lidmi využívající ZJ na komunikaci s nimi. Ti, kteří se cítí být členy komunity neslyšících, se nazývají Neslyšící [5]. Na rozdíl od mluvených jazyků je ZJ vnímán vizuálně. Je založen na tvarech, pohybu a pozicích rukou v prostoru, proto se popisuje jako vizuálně motorický prostředek. Důležité jsou ovšem také pohyby hlavy, těla a mimika. Vývoj českého mluveného jazyka a českého znakového jazyka probíhal odděleně, z toho důvodu jsou gramatiky a slovníky rozdílné.



























Zde je nutno vysvětlit rozdíl mezi pojmy český znakový jazyk (dále ČZJ) a znakovaná čeština. Oboje patří pod pojem Znaková řeč. Primárním komunikačním prostředkem Neslyšících je ČZJ jako prostředek, který se časem vyvíjel. Odlišuje se od jiných národních ZJ hlavně zásobou znaků a gramatikou. Oproti tomu znakovaná čeština je uměle vytvořeným systémem vymyšleným slyšícími na komunikaci s Neslyšícími. Tento systém byl po dlouhou dobu mylně brán jako prestižnější než ČZJ. ČZJ se upravoval a přibližoval znakované češtině. Žádných takových změn ovšem není a nebylo zapotřebí a jako primární komunikační prostředek komunity neslyšících zůstal ČZJ [6].

Výrazové prostředky ZJ jsou velice blízké neverbální komunikaci a gestikulaci. První doložená zmínka o prstové abecedě pochází z doby života mnicha Beda Venerabilima, který žil z let 672-735 n.l. Mnich Bonaventura popsal v letech 1221-1274 prstovou abecedu s cílem komunikace mnichů mezi sebou, byli totiž zavázáni k trvalému mlčení. První spojení samotných neslyšících a ZJ pochází z let života Španěla Pedra de León (1508-1584), který využíval znalostí znaků, psaní a prstové abecedy mezi mnichy k výuce svých dvanácti neslyšících žáků v klášteře. Dokázal tím, že neslyšící jsou schopni číst, psát, myslet i mluvit.

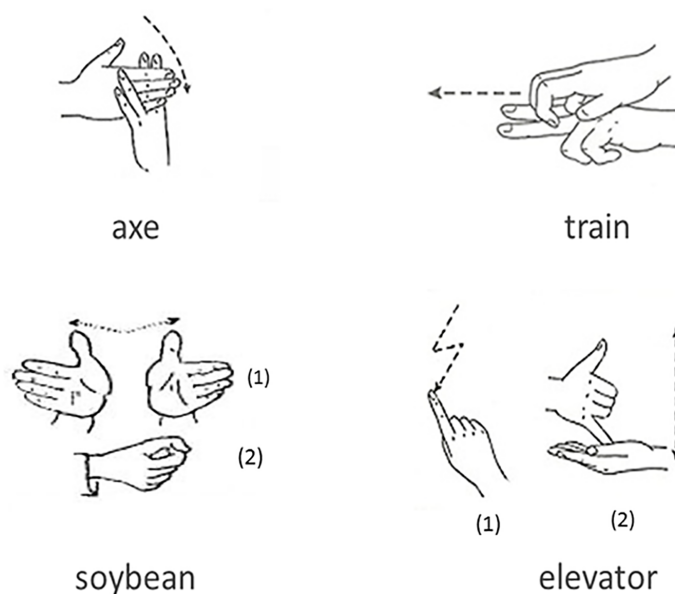
Až v průběhu 17. století začaly vznikat první knihy o prstové abecedě a jejím použití. Byla poprvé popsána a nakreslena jednoruční prstová abeceda. V anglické literatuře byla také popsána metoda odezírání. První knihu o znacích a prstové abecedě vydal v roce 1776 francouzský kněz Abbé l'Epée. Jeho dalšími zájmy byla výchova učitelů a neslyšících. Brzo začaly vznikat školy pro neslyšící a jeho výuce se začalo říkat francouzská metoda, která byla časem přenesena do USA. Tam se používá dodnes. Abbé l'Epée je díky těmto činům pokládán za takzvaného Otce neslyšících [7], [8].

Zde je nutno zmínit, že pojmy prstová abeceda a ZJ jsou dva rozdílné systémy komunikace vycházející z českého jazyka. Prstová abeceda dělí projevy na jednotlivá písmena, která jsou zobrazována formalizovaným postavením prstů, dlaně nebo obou dlaní rukou. Využívá se hlavně při potřebě hláskování, ta se vyskytuje často u cizích slov, odborných termínů apod. Důležitá je prstová abeceda také v případech komunikace hluchoslepých osob. Zde se využívá taktilní formy předávání informace (vzájemný kontakt jedné nebo obou rukou komunikujících). ZJ oproti prstové abecedě využívá gramatiky, která pracuje

se znaky převzatými z českého jazyka. Ty mají na rozdíl od prstové abecedy samostatný význam. Stejně jako u prstové abecedy lze tento aparát využít v taktilní formě jako komunikační prostředek pro hluchoslepé [9]. Příklady znaků prstové abecedy a ZJ jsou uvedeny na obrázcích č. 2.1 a 2.2. Nejpoužívanější notační systémy na oficiální zápis ZJ jsou uvedeny na obrázku č. 2.3.

 <b>A</b>	 <b>B</b>	 <b>C</b>	 <b>D</b>	 <b>E</b>	 <b>F</b>	 <b>G</b>	 <b>H</b>	 <b>I</b>
 <b>J</b>	 <b>K</b>	 <b>L</b>	 <b>M</b>	 <b>N</b>	 <b>O</b>	 <b>P</b>	 <b>Q</b>	 <b>R</b>
 <b>S</b>	 <b>T</b>	 <b>U</b>	 <b>V</b>	 <b>W</b>	 <b>X</b>	 <b>Y</b>	 <b>Z</b>	

Obrázek 2.1: Příklad zápisu prstové abecedy<sup>1</sup>



Obrázek 2.2: Příklad zápisu ZJ<sup>2</sup>

<sup>1</sup>Zdroj:[https://kopecek.pionyr.cz/nauc/prstova/sff\\_prst\\_m.2.png](https://kopecek.pionyr.cz/nauc/prstova/sff_prst_m.2.png)

<sup>2</sup>Zdroj:<https://journals.plos.org/plosone/article/figure?id=10.1371/journal.pone.0120943.g001>





data. Ta reprezentují všechny složky ZJ, tedy pohyb, orientaci (natočení oproti čelu), umístění objektu v prostoru, tvar ruky a mimiku. Pomocí těchto dat lze získat informaci o poloze a pohybu objektů z jakéhokoliv úhlu pohledu.

Metody používané pro záznam pohybu, jejichž výstupem jsou data ve 3D, mohou být rozděleny do pěti základních skupin:

- mechanické systémy
- elektromagnetické systémy
- akustické systémy
- inerciální systémy
- optické systémy

Mechanické systémy MoCap fungují na základě mechanického exoskeletonu (viz. obrázek č. 2.4), který je složen z pevných a flexibilních senzorů (goniometrů a potenciometrů). Tyto senzory jsou spojené s mechanickými kloubovými částmi a tak jsou schopny snímat momentální úhel natočení části těla. Tato data ovšem postrádají jakýkoliv záznam ohledně vztahu měřeného skeletu a jeho okolí, orientaci v prostoru, kontaktu částí těla a podobně. Zařízení je nutno kalibrovat na každého člověka samostatně, spočívá na povrchu těla, a velmi omezuje rozsah a množství pohybů.



Obrázek 2.4: Exoskelet<sup>4</sup>

---

<sup>4</sup>Zdroj:<https://baswaramursyid.wordpress.com/2015/01/20/motion-capture-ii/>

Data z elektromagnetických systémů MoCap jsou pořízena magnetickými přijímači umístěnými na povrchu těla (viz. obrázek č. 2.5). Díky relativnímu magnetickému toku cívek umístěných na generátoru magnetického pole vypočítávají orientaci a pozici v prostoru. Tento typ se často kombinuje s jinými systémy, například řeší problém systémů mechanických při ztrátě kontaktu nohou s podložkou. Dalším příkladem využití je elektromagnetický artikulograf. Ten umožňuje pořízení motion capture dat artikulátorů při vytváření řeči [13].



Obrázek 2.5: Elektromagnetický MoCap systém<sup>5</sup>

Akustické systémy fungují na principu emitorů zvuku umístěných na kloubech člověka. V blízkém okolí pohybujícího se objektu zachycují zvuk čidla (mikrofony). Emitor poté vysílá posloupnost vysokých frekvencí, které čidla zachytí a používají se k výpočtu souřadnic v trojrozměrném prostoru (inside-out systém). Existuje ještě druhý způsob realizace tohoto MoCap systému, a to výměna vysílačů a čidel. Zdroj zvuku tedy vychází z okolí a čidla jsou umístěna na člověku (outside-in).

Jedny z nejjednodušších, avšak méně variabilních systémů, jsou systémy inerciální. Využívají mnoho inerciálních zařízení jako gyroskopy, akcelerometry a magnetometry, které spočívají na snímaném objektu. Díky datům z těchto zařízení, tedy úhlům natočení, rotací ve třech osách a podobně, lze poté zjistit trajektorii pohybu objektu [14].

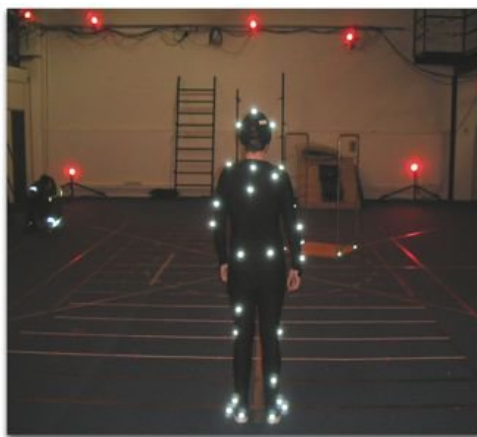
---

<sup>5</sup>Zdroj:<https://baswaramursyid.wordpress.com/2015/01/20/motion-capture-ii/>

Nejběžnější technikou jsou optické systémy MoCap, které vynikají flexibilitou a přesností. Využívají ke snímání polohy a pohybu objektu kamery. Existují jednodušší MoCap optická zařízení, která tvoří pouze kamery RGB. Ty se snaží zachytit např. rozdíly v barvách mezi oblečením a pozadím a tento objekt sledovat. Příkladem zařízení využívajícího kameru RGB je systém Kinect. Ten využívá RGB vstup pro rozpoznávání obličejů, pořizování záznamu aj. Současně s RGB snímáním zde probíhá snímání objektu 3D hloubkovým senzorem tvořeným dvěma částmi. Infračervený zdroj emituje záření, které se odráží od objektů ve scéně a dle doby návratu paprsků do IR VGA senzoru lze určit vzdálenost od snímače ve frekvenci 30 Hz. V kombinaci RGB 2D dat a hloubkové mapy z hloubkového senzoru lze vytvořit 3D náhled na celou scénu. Toto řešení není příliš přesné, avšak je cenově dostupné [15].

Pro složitější optické systémy se na sledovaný objekt umístí série značek (markerů) tvaru koule nebo polokoule. Užívají se dva druhy značek: reflexní a emitující světlo. Značky emitující světlo (aktivní) obsahují svítící LED diody. Dají se rozlišovat různými barvami, umístěním, jejich shlukováním na důležitějších místech aj. Reflexní značky (pasivní) jsou pokryty materiálem odrážejícím infračervené záření, které zachycují kamery s vhodně nastaveným prahem. Povrch objektu v tu chvíli nesmí mít podobné reflexní vlastnosti jako marker [16]. Výhodný je tento typ hlavně díky osvobození objektu od jakýchkoliv přívodů energie a odvodů dat kabelem. Značky je možno snímat pouhými RGB kamerami, nemusí být přítomna reflexní vrstva, a to jednoduše odlišením barvy markerů od povrchu snímaného objektu (např. snímání mimiky obličeje pomocí nakreslených markerů).

Optický MoCap systém s reflexními značkami byl použit na získání dat, jejichž analýza je v této práci prováděna. Jeho hlavní výhoda oproti ostatním systémům spočívá v přesnosti pořizovaných dat týkajících se ZJ. Znakující nebyl při znakování nijak omezen exoskeletem nebo většími snímači na těle a snímání probíhalo u všech značek současně. Problémem je ovšem zakrývání snímaných značek překážkou. Jedna značka musí být v každém okamžiku snímána minimálně dvěma kamerami pro spolehlivé určení její polohy. Příklad optického MoCap systému je uveden na obrázku č. 2.6.



Obrázek 2.6: Optický MoCap systém<sup>6</sup>



## 2.3 Data

Použitá data jsou tvořena projevy v ZJ. Data jsou tématicky omezena na předpovědi počasí. Nahrány byly repliky předpovědí počasí pocházející z archivu ČT, které byly součástí pořadu Zprávy v českém znakovém jazyce. Data tvoří celkově 36 záznamů napříč rokem o přibližné délce 30 sekund na jeden záznam. Každé 3 po sobě jdoucí předpovědi se z důvodu přes rok proměnlivého počasí týkají jednoho daného měsíce.

Předpovědi byly pořízeny optickým MoCap systémem s reflexními markery (viz. kapitola 2.2). Vše bylo nahráváno se snímkovací frekvencí 120 Hz pro umožnění dostatečně jemné analýzy. K dispozici je přibližně 18 minut plynulého záznamu, 176 různých znaků a 1336 jejich unikátních výskytů.

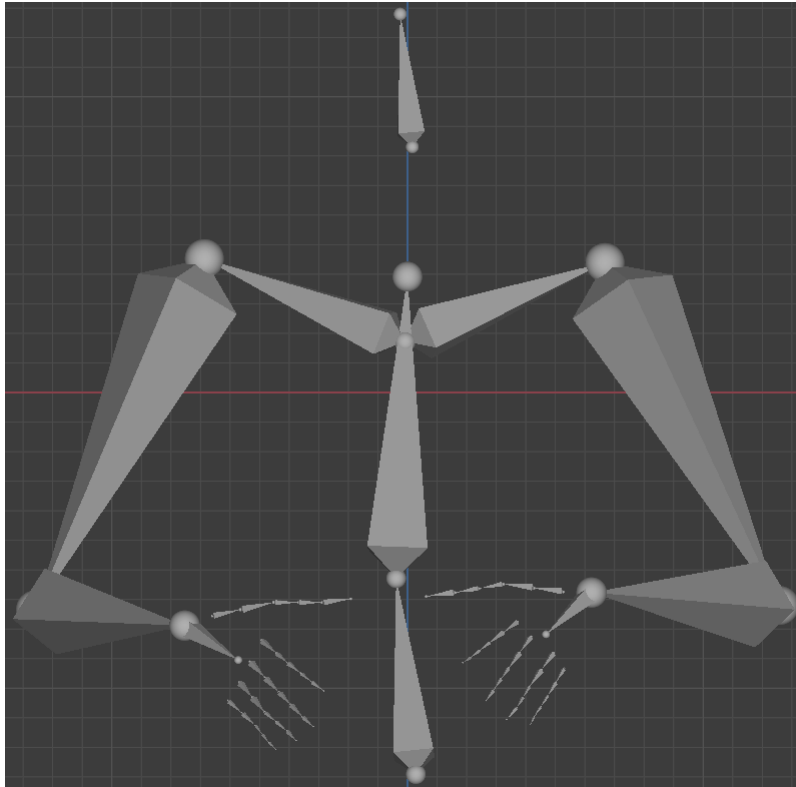
Zpracovávají se 2 typy těchto dat. Jedním typem jsou přímé repliky záznamů předpovědí počasí. Jednotlivé znaky jsou součástí širšího kontextu vět a jsou jím do určité míry ovlivňovány (např. začátek a konec znaku závisí na znacích okolních). Druhým typem je slovník obsahující znaky používané v daném kontextu. Tyto znaky začínají a končí v klidové poloze znakovacího, aby nebyly ovlivňovány znaky okolními. Vyskytují se i takové znaky, které v projevech použity nejsou (některé číselné hodnoty). Z důvodu omezení tématiky projevů a slovních knihoven je větší šance shody dvou významů znaků.

Byla zpracovávána pouze manuální složka dat, nemanuální složka totiž v našem případě nemění význam jednotlivých znaků, proto pro nás není důležitá (viz. kapitola 2.2).

Surová zpracovávaná data jsou trajektorie sledovaných markerů na těle znakovacího. Data je nutno navázat na kosterní model člověka (viz. obrázek č. 2.7) a minimalizovat tím možný šum. Šum může být následkem vzájemného pohybu těla znakovacího a jeho oblečení, na kterém jsou markery připevněny, nebo jakéhokoliv vnějšího vlivu. Proto by se mohlo bez fixace markerů na kosterní model zdát, že mezi jednotlivými snímky nahrávky dochází k deformaci kostí, jejich prodlužování a zkracování.

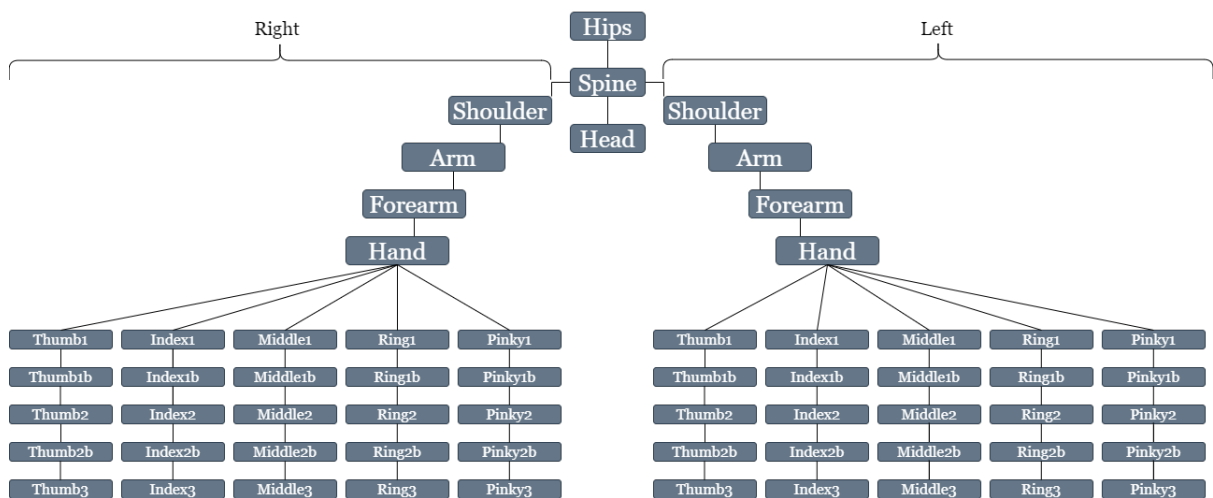
---

<sup>6</sup>Zdroj:[https://www.researchgate.net/publication/37243822\\_Motion\\_Planning\\_from\\_Digital\\_Actors\\_to\\_Humanoid\\_Robots/figures](https://www.researchgate.net/publication/37243822_Motion_Planning_from_Digital_Actors_to_Humanoid_Robots/figures)



Obrázek 2.7: Kosterní model v programu Blender

Po předzpracování jsou výstupem systému MoCap 3D angulární data jednotlivých částí těla Eukleidovského prostoru. Tím se rozumí úhlové natočení jednotlivých segmentů vůči těm, na které jsou napojeny (s počátkem v trupu). Na obrázku č. 2.8 je vykreslen jednoduchý stromový diagram pro reprezentaci jejich rozložení. Z tohoto formátu ovšem nelze přímo vyčíst informace o vztahu částí těla vůči sobě a jejich kontaktu. Proto data podstupují přepočítání na absolutní souřadnice částí těla (kloubů) v prostoru.

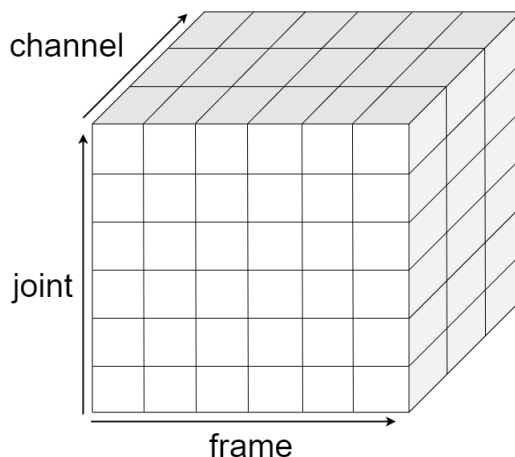


Obrázek 2.8: Graf stromové struktury částí těla

Výsledný zpracovávaný formát je tedy matice o třech dimenzích ve tvaru:

$$[frame, joint, channel].$$

První dvě dimenze reprezentují jednotlivé měřené části těla a postupně všechny 3 kanály Eukleidovského prostoru. Zbývající dimenze obsahuje informaci o počtu snímků daného projevu, ten je různý, odvíjející se od rychlosti znakování a délky znaku. Z dat lze tedy v jakémkoliv čase zjistit polohu jedné ze 61 testovaných částí těla v Eukleidovském prostoru (viz. obrázek č. 2.9).



Obrázek 2.9: Grafická ilustrace dat

## 2.4 Analýza dat

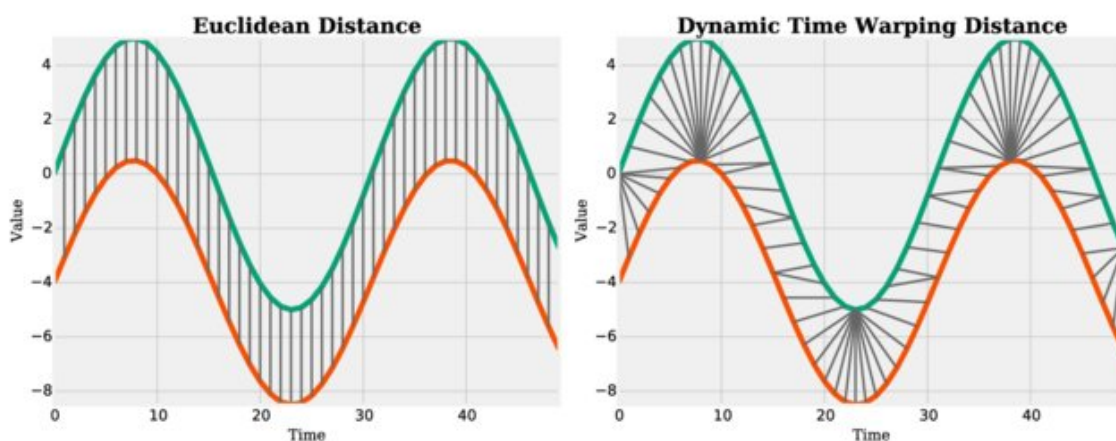
Nejlepším způsobem porovnání podobnosti dvou znaků je subjektivní ohodnocení člověkem. Jestliže je momentálně analyzovaným východiskem problému syntézy ZJ strojové učení neuronových sítí, je potřeba hodnocení automatickou metodou.

Jak již bylo zmíněno v kapitole 2.3, porovnávané projevy jsou trojdimenzionální matice, jejichž jedna z dimenzí reprezentuje čas (počet snímků). I přesto, že dva porovnávané projevy budou mít stejný význam v ZJ, je velice pravděpodobné, že budou mít v časové dimenzi různý počet snímků. V tabulce č. 1 je uvedeno několik užívaných významů znaků s jejich unikátními výskyty a jednotlivými délkami ve snímcích za sekundu. Je tedy potřeba zvolit na jejich porovnání vhodnou metodu.

Algoritmus Dynamického borcení času (anglicky Dynamic Time Warping, dále zkratka DTW) je využíván pro nelineární porovnání časových průběhů. Dokáže měřit podobnost při minimalizaci vlivu vzájemného posunu v čase nebo rychlosti zavedením tzv. bortivé funkce. Funguje na principu hledání ideálního zarovnání dvou vektorů tak, aby zajistil jejich minimální vzdálenost přizpůsobením buď jednoho vzorku druhému referenčnímu, nebo obou navzájem (viz. obrázek č. 2.10). Její výstupy budou tedy využity pro srovnání s ostatními metodami [4].

znak/ č. výskytu	Počet snímků				
	teplo	nekde	zitra	kolem	prset_1
1	58	91	29	52	79
2	53	77	16	41	57
3	65	82	24	52	49
4	48	43	25	39	39
5	72	48	24	33	87
6	63	36	34	52	43
7	58	48	19	68	76
8	86	86	15	56	63
<b>Průměr</b>	63	64	23	49	62

Tabulka 1: Tabulka délek unikátních výskytů znaků



Obrázek 2.10: Grafické znázornění lineárního a nelineárního porovnání signálů<sup>7</sup>

V poslední době jsou na syntézu často používané neuronové sítě se zpětným šířením chyby (algoritmus backpropagation) gradientní metodou. Proto musí být použita diferencovatelná ztrátová funkce. To DTW už z důvodu své algoritmické podstaty nesplňuje. Algoritmus je zároveň velice výpočetně náročný.

Alternativou k DTW algoritmu může být použití kombinace dvou procesů. Prvního za účelem převzorkování znaků na stejnou délku před vstupem do neuronové sítě. Druhého na zjištění jejich podobnosti. Převzorkování znaku na délku druhého může být realizováno například Furierovou transformací nebo interpolací signálu (viz. kapitola 2.5.3). Na zjištění podobnosti signálů je možno využít mnoho metrik, testovány jsou Eukleidova, Hammingova, Minkovského, Čebyševova, Mahalanobisova, Canberrská, Fréchetova a Brayova-Curtisova metrika nebo například korelace signálů. Všechny vybrané funkce byly systematicky procházeny a funkce a metriky kombinovány. Výsledkem analýzy kombinací by měla

<sup>7</sup>Zdroj:[https://www.researchgate.net/publication/283458699\\_Scalable\\_time\\_series\\_similarity\\_search\\_for\\_data\\_analytics/figures?lo=1](https://www.researchgate.net/publication/283458699_Scalable_time_series_similarity_search_for_data_analytics/figures?lo=1)



být taková kombinace, která se bude schopna co nejlépe přiblížit chování výsledků algoritmu DTW. Pro znaky stejného významu je požadována minimální vzdálenost, popřípadě maximální podobnost. Detaily pohybů, které v ZJ nemění význam, by měly hrát ve výsledné hodnotě minimální roli. Naopak malé rozdíly, které již způsobí změnu významu znaku, by měly být měřitelné. Rozdíly mezi nestejnými znaky by měly nabývat statisticky významných hodnot.

## 2.5 Analyzované metody

Cílem je porovnání dvou trojdimenzionálních vektorů s různou délkou v jedné z dimenzí. Data jsou tedy ve formátu 3D matic. Metody, které jsou schopny zpracovat vstup ve více dimenzích, jsou pro toto použití lepší. Pokud ho schopny zpracovat nejsou, je signál rozdělen dle částí těla a předkládán po jednotlivých kanálech (x,y,z).

Výsledky algoritmu DTW budou využity jako baseline [4]. Byl vytvořen za cílem porovnání rozdílně dlouhých signálů, proto by měl být v této situaci optimálním řešením. Je ovšem nutno získat diferencovatelné metriky porovnání signálů, DTW tento požadavek už z důvodu své algoritmické podstaty nespĺňuje. Proto jsou kombinovány funkce pro převzorkování signálu s funkcemi na jejich porovnání.

### 2.5.1 Algoritmus DTW

Algoritmus DTW funguje na základě dynamického programování. Dokáže měřit podobnost při minimalizaci vlivu vzájemného posunu v čase nebo rychlosti zavedením tzv. bortivé funkce. Odchylka dvou časových průběhů, jejichž vzdálenost je zjišťována, není většinou lineární. Bortivá funkce řeší tento problém přizpůsobením vzorků sobě navzájem tak, aby měly minimální vzdálenost viz. obrázek č. 2.10.

Výrazem vzorek jsou v tomto případě myšlena data pojící se k jednomu okamžiku. Mohou být ve formě samostatných skalárů, ale také vektorů s různým počtem dimenzí. V tomto případě je čas jednou ze tří dimenzí analyzovaných dat. Při zúžení časové dimenze na jeden snímek promluvy bude mít vzorek formát dvojdimenzionálního vektoru.

Při průběhu algoritmu probíhá analýza vzdálenosti vzorku testovaného s vzorkem referenčního časového průběhu a jeho sousedů. Mezi vzorky algoritmus umožňuje zjistit transformační cesty s nejmenší vzdáleností. Pomocí těch je dále jeden, nebo oba signály přizpůsobeny (lokálně zkráceny, prodlouženy).

Označme testovaný časový průběh  $X$  a referenční časový průběh  $Y$ . V matici  $A$  o rozměru  $I \times J$  je hledána optimální cesta ( $j = f(i)$ ), která v každém kroku minimalizuje hodnotu funkce  $D$ . Funkce  $D$  reprezentuje všechny možné vzdálenosti mezi vektory signálu  $X$  a  $Y$  a lze ji obecně zapsat jako

$$D(X, Y) = \sum_{i=1}^I d(x_i, y_{f(i)}) \quad (1)$$

kde  $d(x_i, y_{f(i)})$  je myšlena lokální vzdálenost mezi prvky signálů  $x_i$  a  $y_{f(i)}$ . Vzdálenost lze vypočítat jakoukoliv dostupnou metrikou, která je na formát vzorků vhodná. Ve většině případů je používána Eukleidovská metrika viz. kapitola 2.5.4.

Při zavedení obecné časové proměnné  $k$  a dále časových proměnných  $m$  a  $n$  vyjádřenými vztahy

$$\begin{aligned} k &= 1, \dots, K \\ m(k) &= i(k) \\ n(k) &= j(k) \end{aligned} \quad (2)$$

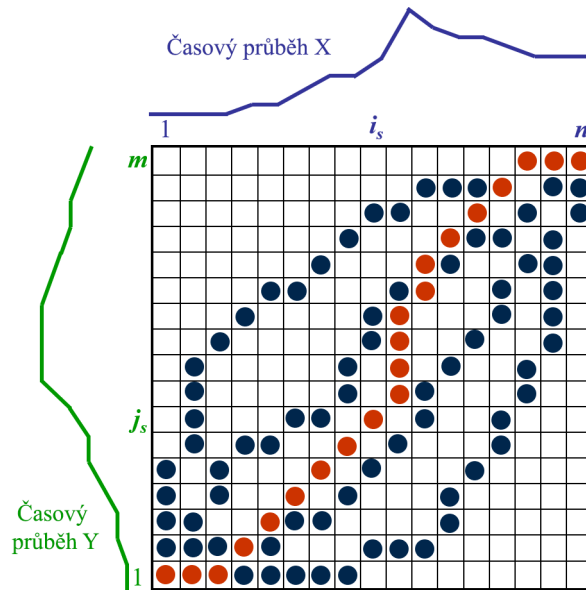
kde hodnota  $K$  reprezentuje délku obecné časové osy, můžeme definovat tato pravidla, pro správný průběh algoritmu DTW [17]:

- Okrajové podmínky:  $p_1 = (1, 1)$ ,  $p_k = (I, J)$
- Zákaz běhu pozpátku v čase
- Zákaz vynechávání, či přerušování časových úseků
- Zákaz mapování velmi malé části průběhu  $X$  do velmi velké části průběhu  $Y$
- Zákaz ignorování nepodobných rysů a pozastavování se pouze u podobných

Fumitada Itakura zformuloval tzv. podmínky spojitosti, nebo také Itakurovy podmínky [18]:

$$\begin{aligned} f(i) = j &\iff f(i-1) = j \wedge f(i-2) \neq j \\ &f(i-1) = j-1 \\ &f(i-1) = j-2 \end{aligned} \quad (3)$$

Podmínky redukuje možný počet transformačních cest algoritmu, i přesto by jich ovšem bylo velice mnoho. To řeší používané způsoby dynamického programování, díky kterým lze vyčíslit vzdálenosti cest již při jejich vytvoření. Následující grafické znázornění č. 2.11 poskytuje náhled na některé možné varianty průběhu algoritmu DTW [19].



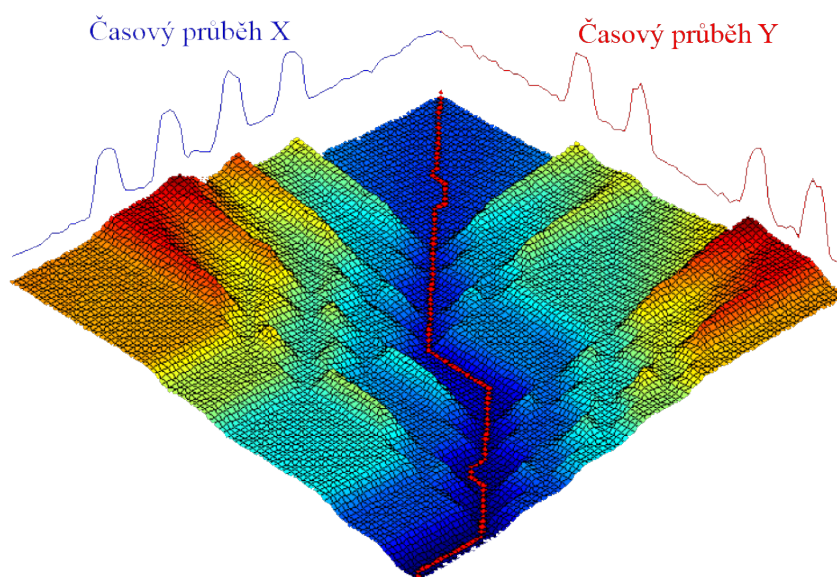
Obrázek 2.11: Grafické znázornění algoritmu DTW<sup>8</sup>

V matici o velikosti  $I \times J$  algoritmus začíná v počátečním bodě  $p_1 = (1, 1)$ . Hodnota v tomto bodě se rovná vzdálenosti vzorků  $x_1$  a  $y_1$ , tedy  $d(x_1, y_1)$  spočtena dle vybrané metriky. Následně probíhají iterace maticí, kde se hodnota na pozici  $[i, j]$  rovná minimu ze součtů:

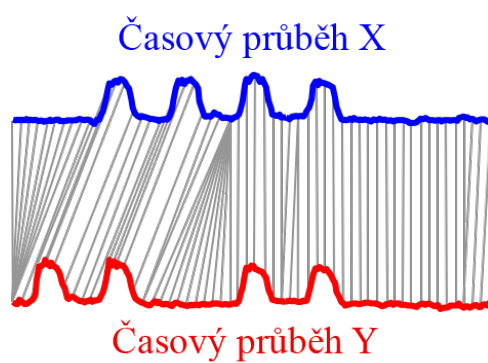
$$\begin{aligned}
 & d(x_i, y_j) + d(x_{i-1}, y_j) \\
 & d(x_i, y_j) + d(x_{i-1}, y_{j-1}) \\
 & d(x_i, y_j) + d(x_i, y_{j-1})
 \end{aligned} \tag{4}$$

Po poslední iteraci algoritmus končí v bodě  $p_k = (I, J)$ . Následuje vybrání optimální cesty (viz. obrázky č. 2.12 a 2.13) na základě spočtených hodnot se začátkem v bodě  $p_k = (I, J)$  [20].

<sup>8</sup>Zdroj:<http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWAlgorithm.ppt>



Obrázek 2.12: Vizuální vykreslení výsledku DTW<sup>9</sup>



Obrázek 2.13: Grafické znázornění výsledku DTW<sup>9</sup>

<sup>9</sup>Zdroj:<https://www.cs.unm.edu/~mueen/DTW.pdf>

## 2.5.2 Algoritmus SoftDTW

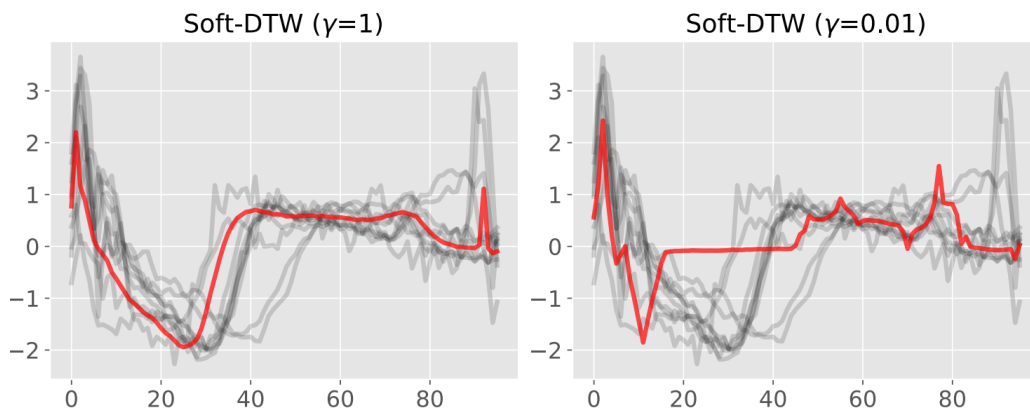
Vzhledem k problémům algoritmu DTW s diferencovatelností ztrátové funkce byl publikován roku 2017 článek pojednávající o modifikaci ztrátové funkce s cílem zajištění její diferencovatelnosti [21]. Ztrátová funkce nemodifikovaného DTW algoritmu při porovnávání dvou časových průběhů  $A$  a  $B$  vypadá následovně:

$$dtw_0(X, Y) = \min_{a \in A(i,j)} \langle A, \delta_{X,Y} \rangle, \quad (5)$$

kde  $A$  je ztrátová matice viz. kapitola 2.5.1. Tato ztrátová funkce je v softDTW verzi dle autorů M. Cuturi a M. Blondel modifikována následovně:

$$dtw_\gamma(X, Y) = -\gamma \log \sum_{a \in A(i,j)} e^{-\frac{\langle A, \delta_{X,Y} \rangle}{\gamma}}, \quad (6)$$

kde  $\gamma$  reprezentuje parametr, s jehož zvětšením dochází k vyhlazování ztrátové funkce [21]. Tento vliv je ukázán na grafu na obrázku č. 2.14:



Obrázek 2.14: Ztrátová funkce softDTW při  $\gamma=1$  a  $\gamma=0.01$ <sup>10</sup>

Díky vyhlazení je funkce vhodná pro optimalizaci algoritmu gradientní metodou. Vzhledem k tomu, že softDTW funguje na stejném principu jako DTW, je testováno samostatně a není za potřeby převzorkování, či porovnávacích metrik.

<sup>10</sup>Zdroj:<http://proceedings.mlr.press/v70/cuturi17a/cuturi17a.pdf>

### 2.5.3 Převzorkování

Prvním krokem k porovnání různě dlouhých signálů je převzorkování jednoho z porovnávaných tak, aby měly porovnávané signály stejnou délku. Převzorkování (anglicky resampling) umožňuje snižovat a zvyšovat lokální snímkovací frekvenci. Označme  $s(t)$  aproximační funkci reálné proměnné  $t$ , která dokáže na dostatečně velkém úseku lokálně aproximovat původní časový průběh. Po převzorkování jsou vzorky v časovém průběhu funkčními hodnotami aproximační funkce  $s(t)$  v daných časových okamžicích. Tedy pro okamžik  $t = 10$  je vzorek funkční hodnotou  $s(10)$ .

Jedním z problémů převzorkování signálu je validace správnosti chování výsledného časového průběhu. Dostupnou možností ověření správnosti aproximace je aplikování DTW algoritmu na převzorkovaný a původní časový průběh. Metoda, která dokáže poskytnout převzorkovaný signál s nejmenší DTW vzdáleností, bude vybrána a použita v analýze.

### Fourierova transformace

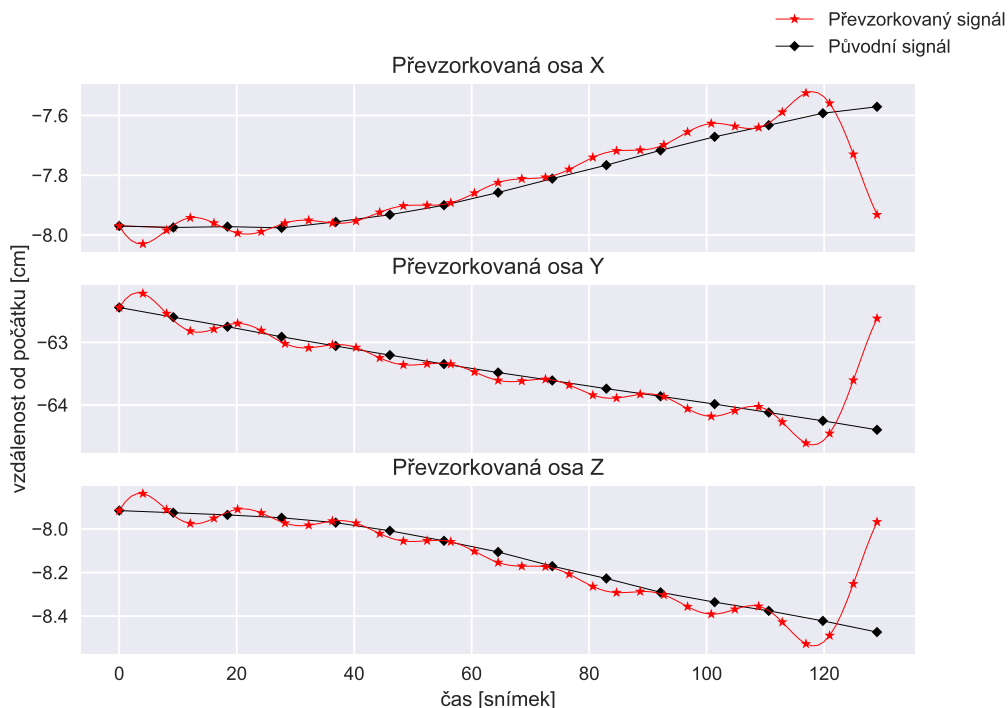
Aproximační funkce může být definována například transformací dle integrálního Fourierova vztahu:

$$S(w) = \int_{-\infty}^{\infty} s(t)e^{-i\omega t} dt \quad (7)$$

Funkci  $S(w)$  reálné proměnné  $w$  nazýváme Fourierovým obrazem funkce  $s(t)$ . Zmíněný vztah lze také definovat jako:

$$S(w) = F[s(t)] \quad (8)$$

Zde značí  $F$  Fourierovu transformaci veličiny  $s(t)$ . Z důvodu vstupu jako diskrétního signálu je nutno použít diskrétní Fourierovu transformaci. Funkce  $e^{-i\omega t}$  jako jeden z členů předpisu Fourierovy transformace je funkcí periodickou, periodický je tedy i výsledný Fourierův obraz  $S(w)$  [22]. Příklad chování signálu získaného Fourierovou transformací je uveden na grafu na obrázku č. 2.15.



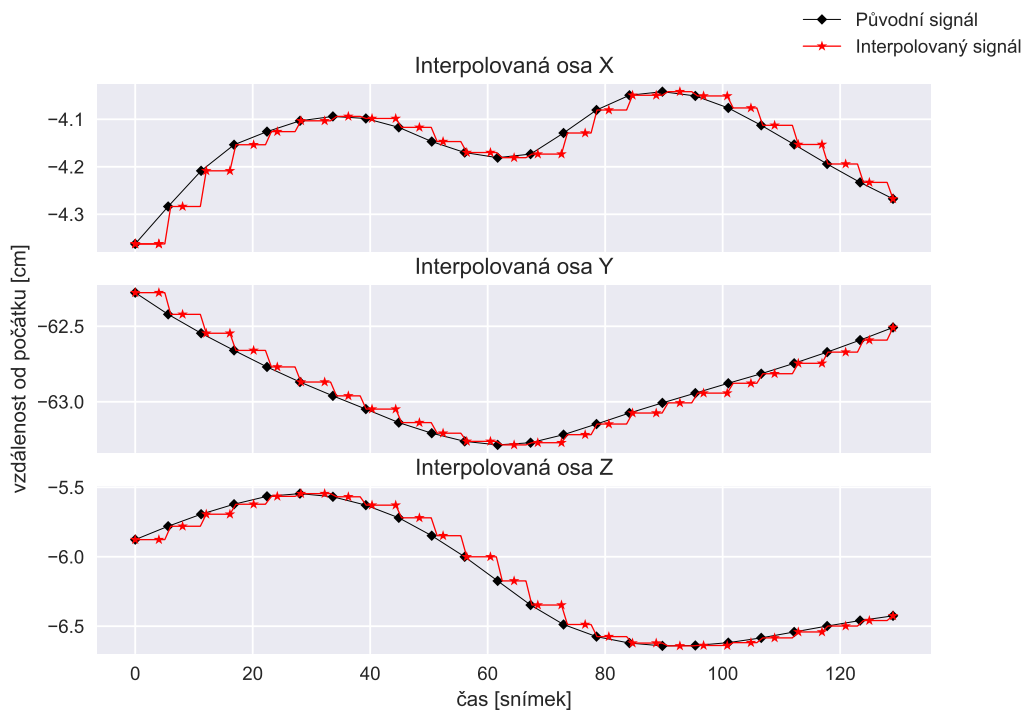
Obrázek 2.15: Příklad Fourierovy transformace

## Polynomiální interpolace

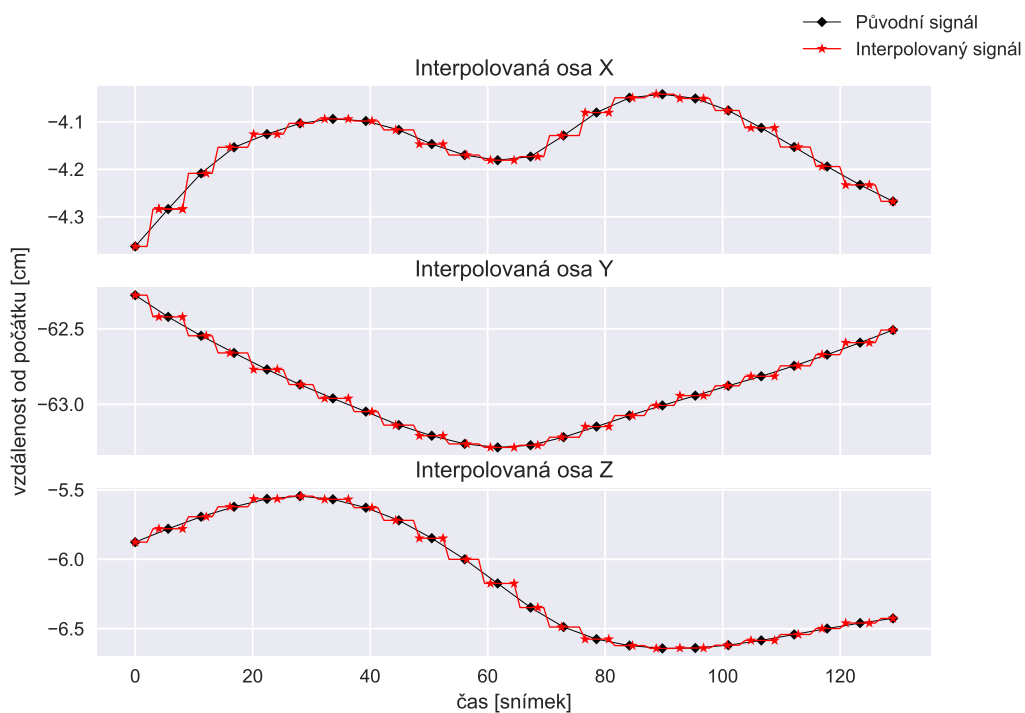
Další možností převzorkování časového průběhu je interpolace. Cílem metody je nalezení funkce  $y = g(x)$  k funkci  $y = f(x)$  (nebo v našem případě posloupnosti hodnot), která bude pro různé hodnoty  $x_1, \dots, x_n$  nabývat stejných funkčních hodnot. Tedy bude platit vztah  $f(x_i) = g(x_i)$  (popř.  $a_i = g(x_i)$ ).

Nejjednodušší interpolační metodou je například nahrazení funkční hodnotou prvního (druhého) referenčního bodu, nebo funkční hodnotou průsečíku na lineární průsečnici ve středu interpolovaného intervalu s osou X, která se nejvíce blíží původnímu signálu. Tyto interpolace jsou velice operačně jednoduché, ovšem postrádají na přesnosti (viz. obrázky č. 2.16, 2.17).





Obrázek 2.16: Příklad interpolace nahrazením funkční hodnotou  $f(x_1)$



Obrázek 2.17: Příklad interpolace nahrazením funkční hodnotou průsečíku

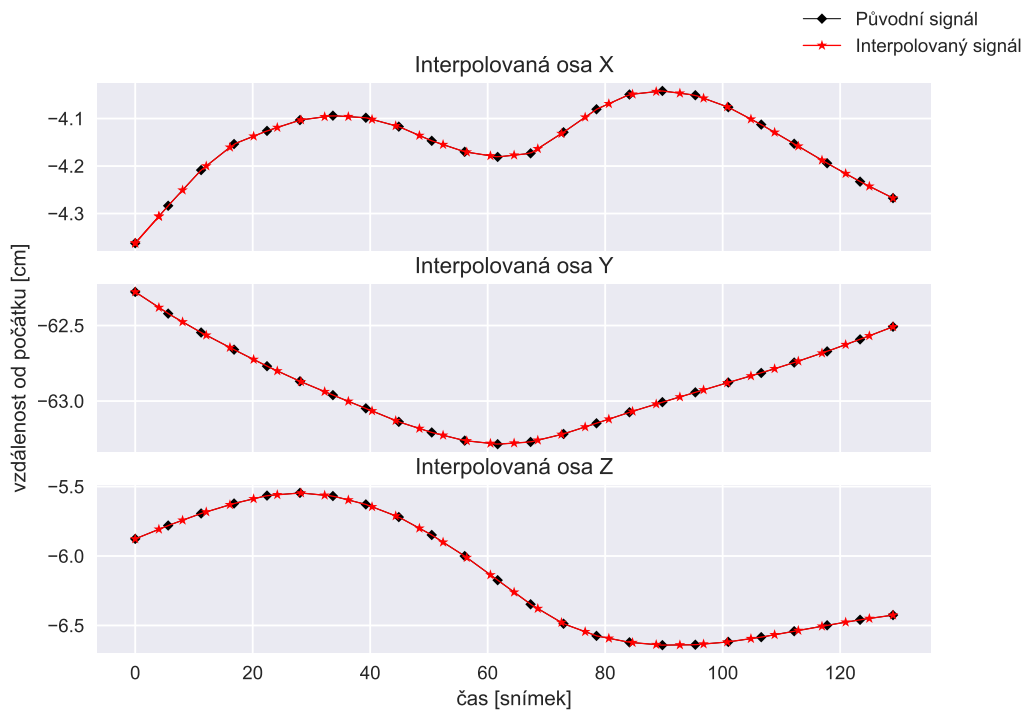
Při lineární interpolaci se využívá proložení lineární funkcí s předpisem:

$$y = ax + b \quad (9)$$

Pro dostatečně rozdílné souřadnice  $(k+1)$  bodů ovšem neexistuje funkce, která by procházela všemi body. Lineární funkce má pouze dva parametry  $a$  a  $b$ , může tedy spolehlivě procházet pouze dvěma body z  $(k+1)$  známých. Pro vyřešení jejího předpisu řešíme následující soustavu.

$$\begin{aligned} y_i &= a * x_i + b \\ y_{i+1} &= a * x_{i+1} + b \end{aligned} \quad (10)$$

Dvojicí  $[x_i, y_i], [x_{i+1}, y_{i+1}]$  je tedy definována jedna lineární funkce. Tato funkce je počítána pro každou dvojici po sobě jdoucích bodů. Výsledná aproximační funkce je poté lomená čára tvořena soustavou těchto úseček (viz. obrázek č. 2.18).



Obrázek 2.18: Příklad lineární interpolace

Interpolace polynomem funguje na základě předpisu polynomiální funkce ve tvaru

$$\sum_{n=1}^k a_n * x^n. \quad (11)$$

Pro polynomiální funkci stupně  $k$  existuje přesně  $(k+1)$  definujících koeficientů  $a_1, \dots, a_n$ . Pomocí  $(k+1)$  známých hodnot  $[x_i, y_i]$  je možno vyjádřit  $(k+1)$  lineárních rovnic o stejném

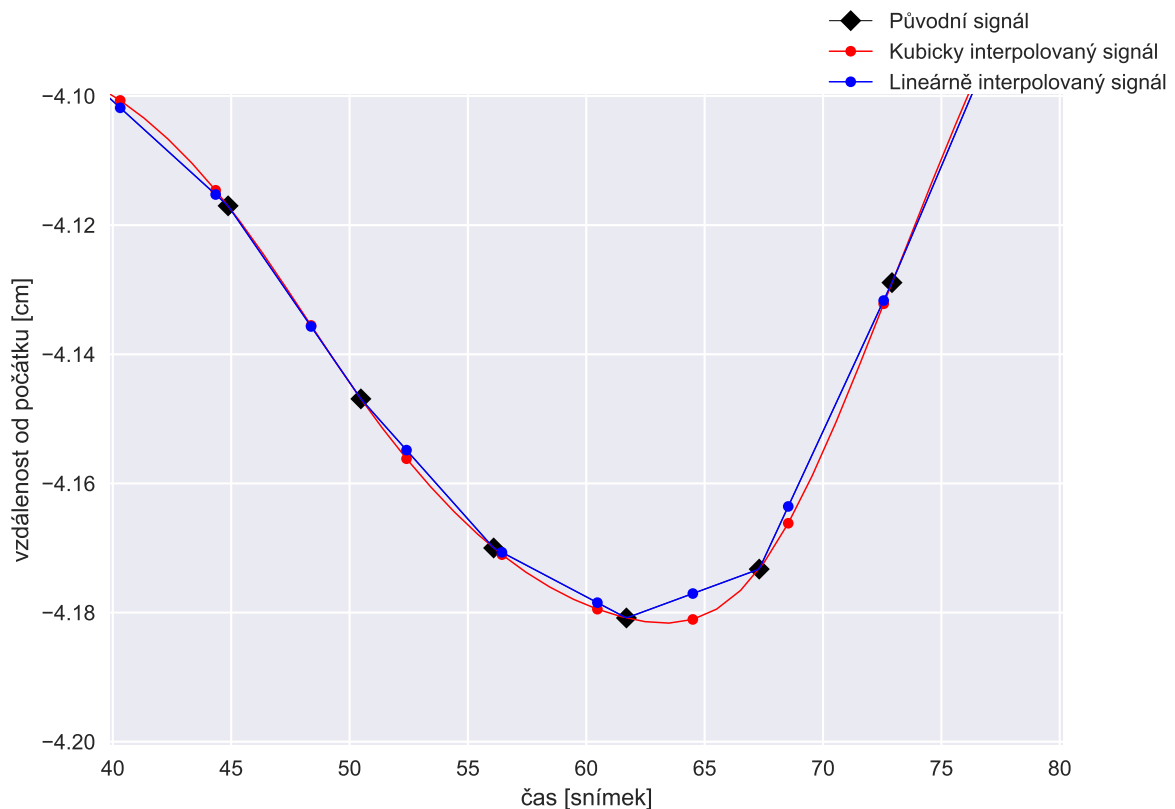
počtu neznámých koeficientů. Vyřešením jejich soustavy je poté (kromě případů se singularní maticí soustavy, v tu chvíli má soustava 0 nebo nekonečně mnoho řešení) jediný polynom stupně  $k$ , který dokáže chování posloupnosti daných bodů funkčně aproximovat.

Analogicky k lineární aproximaci lze provádět interpolaci vyšších řádů. Lze užít interpolaci kvadratickou, jejíž předpis vypadá takto:

$$y = a * x^2 + b * x + c \quad (12)$$

Grafem takovéto funkce je parabola, jejíž rovnice je určena přesně třemi body  $[x_i, y_i]$ ,  $[x_{i+1}, y_{i+1}]$ ,  $[x_{i+2}, y_{i+2}]$  [23, 24]. Řešena by v tomto případě byla soustava tří takových rovnic. Na stejném základě je možno provádět interpolaci kubickou aj.

Rozdíl mezi aproximací kubické a lineární interpolace (viz. obrázek č. 2.19) je znázorněn v nejostřejším úseku interpolovaného signálu z obrázku č. 2.18 mezi snímky 60 a 80 níže.



Obrázek 2.19: Detail rozdílu mezi lineární a kubickou interpolací

## 2.5.4 Porovnání

Po převzorkování jednou z uvedených metod následuje porovnání převzorkovaného a původního signálu s jiným významem znaku. V porovnávání je nutno zjistit vzdálenost přes všechny klouby ve všech časových okamžicích. Vzhledem k tomu, že dva porovnávané kanály mají již totožnou délku v časové dimenzi, je možno porovnávat dva vzorky náležící stejnému okamžiku. Výpočet je prvně proveden přes celou časovou dimenzi pro jednotlivé klouby, následně jsou sečteny výsledky všech testovaných kloubů. Tento postup používá většina níže popsaných metrik.

Metrika, která bude vyhodnocena jako nejlepší, by měla být schopna se co nejlépe přiblížit chování výsledků algoritmu DTW. Pro znaky stejného významu je požadována minimální vzdálenost, popřípadě maximální podobnost. Vzdálenost těchto dvou znaků by měla být vyhodnocena jako nedůležitá pro rozdíl ve významu. Stejně tak detaily pohybů, které v ZJ nemění význam, by měly hrát ve výsledné hodnotě vzdálenosti minimální roli.

Naopak malé rozdíly, které již způsobí změnu významu znaku, by měly být měřitelné. Měly by nabývat hodnot překračujících experimentálně určený práh, a tím poskytnout informaci o rozdílu významů daných signálů. Vzdálenost kompletně rozdílných znaků by měla nabývat významných hodnot.

Následující sekce bude pracovat s definicí  $u$  a  $v$  jako dvěma porovnávanými body v prostoru. Dále  $u_i$  a  $v_i$  reprezentují jejich souřadnice v  $i$ -té dimenzi 3D prostoru, tedy  $i = 1, \dots, n$ , kde  $n = 3$ . Označením  $X^T$  je myšlena transpozice  $n$ -dimenzionálního vektoru (popř. matice)  $X$ .

### Eukleidovská metrika

Vzdálenost dvou  $n$ -dimenzionálních časových průběhů  $u$  a  $v$  spočtena Eukleidovskou metrikou je definována vztahem:

$$D_E(u, v) = \left[ \sum_{i=1}^n (u_i - v_i)^2 \right]^{\frac{1}{2}} \quad (13)$$

Kvadrát rozdílu dvou vzorků znamená, že na větší vzdálenosti je kladen větší důraz. Geometrickou interpretací míst se stejnou Eukleidovskou metrikou od počátku  $x_0$  je kruh (popř. koule ve 3D prostoru).

Do vztahů na počítání vzdálenosti lze vždy zapojit určitý vyrovnávací faktor závisející na daných hodnotách, z nichž je vzdálenost počítána (např. střední hodnota  $\bar{x}$  nebo směrodatná odchylka  $s_x$ ) [25]. Takovýto druh transformace proměnných lze obecně zapsat pomocí matice váhových koeficientů  $C$  jako:

$$u = C^T * x, \quad (14)$$

kde jsou koeficienty transformační matice:

$$\begin{aligned} c_{ii} &= a_i, i = 1, \dots, m, \\ c_{ij} &= 0, i \neq j, \end{aligned} \tag{15}$$

kde  $m$  reprezentuje počet vzorků v signálu. Eukleidovskou metriku pro sloupcové vektory s použitím transformační váhové matice lze poté zapsat vztahem [25]:

$$D_{WE}(u, v) = [(u - v)^T * CC^T * (u - v)]^{\frac{1}{2}}, \tag{16}$$

### Hammingova metrika (Manhattanská)

Hammingova nebo Manhattanská metrika je definována vztahem:

$$D_H(u, v) = \sum_{i=1}^n |u_i - v_i|, \tag{17}$$

kde  $n$  je počet dimenzí vzorků  $u$  a  $v$ . Vztah je vytvořen linearizací Eukleidovské metriky. Díky tomu je snížena výpočetní náročnost a zároveň není dáována větší důležitost členům s větším rozdílem dílčích členů. Tato metrika je taktéž nazývána *city-block* [25]. Je totiž vzdáleností podobná vzdálenosti pohybu v pravoúhle zastavěném městském prostředí z místa  $A$  do místa  $B$  (viz. obrázek č. 2.20).



Obrázek 2.20: Grafické znázornění Eukleidovské a Hammingovy metriky<sup>11</sup>

<sup>11</sup>Zdroj:<https://aipo.computing.dcu.ie/2015-aipo-preliminary-round-problems>

## Minkovského metrika

Minkovského metrika zobecňuje předchozí dvě metriky a je definována vztahem:

$$D_M(u, v) = \left[ \sum_{i=1}^n |u_i - v_i|^p \right]^{\frac{1}{p}}, \quad (18)$$

kde  $n$  je počet dimenzí vzorků  $u$  a  $v$ . V tomto vztahu je použita obecná mocnina, z toho důvodu je tedy váha členů s větší vzdáleností obou vzorků nastavitelná vstupním určením parametru  $p$ . Čím větší vstupní parametr  $p$  bude, tím větší bude brán důraz na velké rozdíly vzorků [25].

## Čebyševova metrika

Čebyševova metrika je limitním případem Minkovského metriky. Je dána vztahem:

$$D_C(u, v) = \max_{\forall i} |u_i - v_i| \quad (19)$$

Limitní vztah Minkovského a Čebyševovy metriky je popsán níže:

$$D_C(u, v) = \lim_{p \rightarrow \infty} D_M(u, v), \quad (20)$$

kde  $p$  je parametr viz. kapitola 2.5.4. Tato metrika je používána při potřebě snížení výpočetní náročnosti dříve zmiňovaných metrik. Geometrické místo bodů od počátečního bodu  $x_0$  Čebyševovy vzdálenosti je čtverec (popř. krychle ve 3D prostoru) [25].

## Mahalanobisova metrika

Vztah pro Mahalanobisovu metriku vychází ze vztahu pro Eukleidovskou metriku s použitím váhové transformační matice. Místo této matice je ovšem použita inverzní kovarianční (disperzní) matice  $S^{-1}$  dvou testovaných sloupcových vektorů.

$$D_{MA}(u, v) = D_E(u, v) = [(u - v)^T * S^{-1} * (u - v)]^{\frac{1}{2}}, \quad (21)$$

kde je kovarianční matice určena vztahem:

$$S(u, v) = E((u - Eu)(v - Ev)^T), \quad (22)$$

kde  $E(x)$  je střední hodnota  $x$ . Je zde možnost použití korelační matice namísto kovarianční. Zde záleží na tom, jestli jsou využitelnější data obsahující jejich střední hodnotu, nebo nikoliv [25].

## Canberrská metrika

Canberrská metrika je tzv. relativizovanou verzí Hammingovy metriky a je definována vztahem:

$$D_C(u, v) = \sum_{i=1}^n \frac{|u_i - v_i|}{|u_i| + |v_i|}, \quad (23)$$

kde  $n$  je počet dimenzí vstupního vzorku  $u$  a  $v$ . Pokud jedna ze souřadnic je rovna nule, je roven nule celý zlomek bez ohledu na druhou hodnotu (to platí i v případě jsou-li nulové obě hodnoty). Pokud jsou oba vektory orientovány v blízkosti počátku souřadnicové soustavy, je tato metrika velice citlivá na malé změny. V opačném případě, pokud jsou hodnoty souřadnic velké (vektory jsou daleko od počátku), se její citlivost na malé změny snižuje [25].

## Brayova-Curtisova metrika

Brayova-Curtisova metrika je definována vztahem:

$$D_{BC}(u, v) = \frac{\sum_{i=1}^n |u_i - v_i|}{\sum_{i=1}^n (u_i + v_i)}, \quad (24)$$

kde  $n$  je počet dimenzí vzorků  $u$  a  $v$ . Dle G. M. Wojcika and M. Ważneho [26] dokáže při porovnání dvou časových průběhů Brayova-Curtisova metrika výrazněji zobrazit významnější rozdíly hodnot než Eukleidovská metrika. Křivky vykazují podobné chování, jsou ovšem strmější.

## Pearsonův korelační koeficient

V této kapitole jsou neznámými  $x$  a  $y$  definovány matice o velikosti channel x frame, kde channel vyjadřuje 3 kanály 3D prostoru a frame reprezentuje počet snímků analyzovaného signálu. Pearsonův korelační koeficient vyjadřuje míru lineární závislosti dvou vzorků  $x$  a  $y$  v intervalu  $\langle -1, 1 \rangle$ . Je definován tímto vztahem:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (25)$$

kde  $\bar{x}$  je střední hodnota elementů  $x$  a  $n$  je počet dimenzí vstupních vektorů.

## 3 Praktická část

Spouštěný testovací skript je psán v programu Python verze 3.8.5. Na testování metrik pro zjištění podobností znaků je použito funkcí z knihovny SciPy, která je častým řešením pro technické výpočty. Dále byly použity knihovny numpy [27], matplotlib [28], pandas [29], scikit-learn [30], similaritymeasures [31], dtaidistance [32], dtw-python [33] a soft-dtw [21].

### 3.1 Validace použitých typů algoritmu

Postup pro určení schopností vybraného algoritmu je poměrně výpočetně náročný, avšak principiálně jednoduchý. Pro každou testovanou verzi porovnávacího algoritmu je napočtena matice o velikosti  $N \times N$ , kde  $N$  je počet unikátních výskytů jednotlivých znaků (dále projevů) přes všechny nahrané projevy v ZJ. Tuto matici lze nazvat distanční maticí vybraného typu algoritmu, kde hodnota  $k$  na pozici  $[i, j]$  reprezentuje výsledek porovnání  $i$ -tého a  $j$ -tého projevu daným typem algoritmu.

Jednotlivé řádky a sloupce tedy náleží unikátním projevům a seřazením hodnot  $k$  v  $i$ -tém řádku vzestupně dostaneme od začátku seznam projevů, které byly vybraným algoritmem ohodnoceny jako  $i$ -tému referenčnímu projevu nejpodobnější. Mezi takto seřazenými projevy by poté měly minimální hodnoty  $k$  ideálně náležet projevům se stejným významem jako referenční  $i$ -tý projev. Prvním krokem samotné validace je výběr  $n$  počátečních (nejnižších) hodnot v daném řádku (dále  $n$ -oblasti). U každé takto vybrané hodnoty je porovnán význam projevu s referenčním (na dané řádce). Pokud je zjištěna shoda významu, projev řadíme do shodných (zelená oblast ve sloupci histogramu), pokud shoda nalezena není je projev zařazen do rozdílných (červená oblast ve sloupci histogramu).

Po zpracování všech řádek dané distanční matice pro zvolenou  $n$ -oblast jsou výsledky řádek rozděleny dle významu jejich referenčních projevů. Je zjištěn průměrný počet shod pro každý význam a následně průměrný počet shod přes všechny významy. Výsledné průměrné zastoupení projevů se stejným významem v  $n$ -oblasti je na výstupu ilustrováno procentuální měrou.

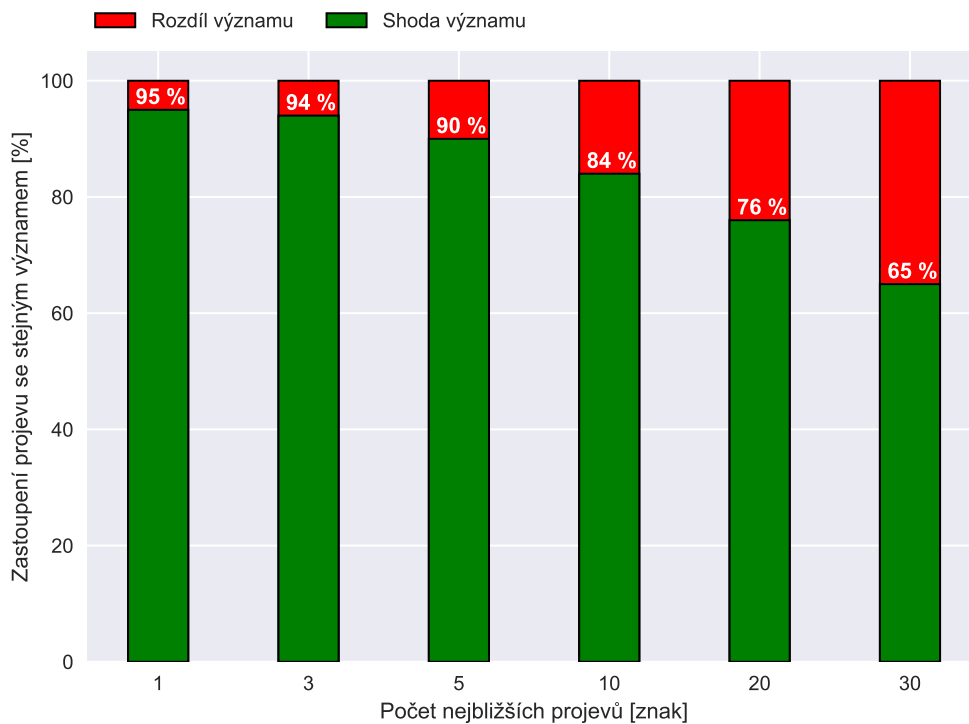
Výsledky dané  $n$ -oblasti z části ovlivňuje to, kolik instancí daného významu se v analyzovaných datech vyskytuje. Při analýze výsledků pro příliš velké  $n$ -oblasti je možné, že význam nebude mít v analyzovaných datech dostatečný počet instancí. Z toho důvodu bude do dané  $n$ -oblasti započítána chyba, kterou lze definováním její správně velikosti eliminovat. Další možností řešení tohoto problému je omezení analyzovaných dat pouze na významy, které jsou zastoupeny požadovaným počtem instancí. Proto je funkce na analýzu implementována tak, aby byla možnost tuto hodnotu (práh) stanovit a eliminovat tím nechtěnou a ve výsledku nic nevyovídající chybovost.



Příklad takového histogramu pro velikosti n-oblastí

- 1 (pouze shoda s nejbližším projevem)
- 3
- 5
- 10
- 20
- 30 (n-oblast větší než minimální počet instancí významu, možná větší chybovost)

je uveden na grafu na obrázku č. 3.1:



Obrázek 3.1: Ilustrační příklad reprezentace úspěšnosti algoritmu histogramem

### 3.2 Omezení dat

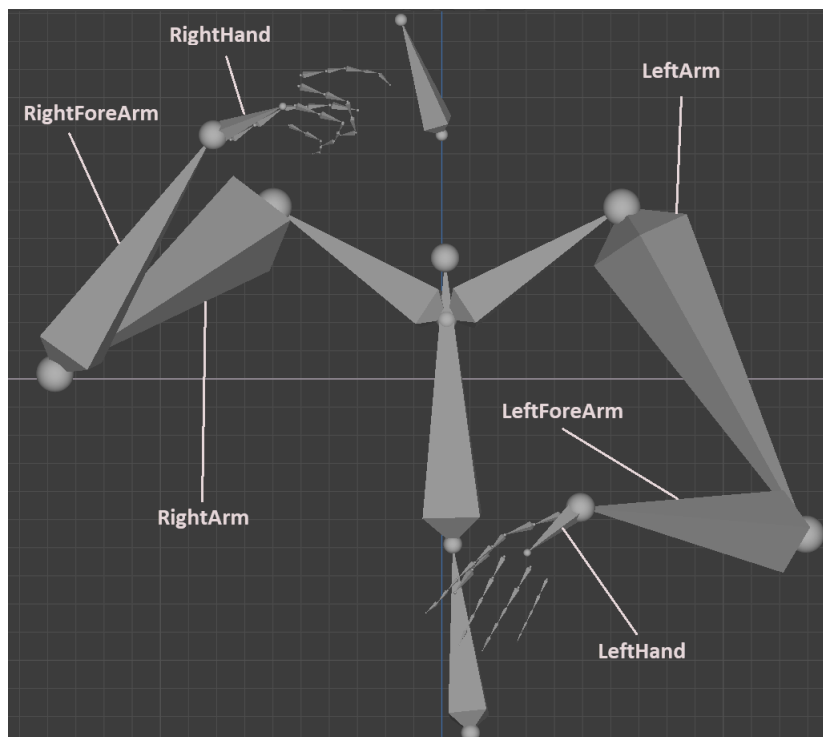
Úplná kostra, která je v datech sledována, je složena ze 61 částí viz. kapitola 2.3. V úloze omezení dat jsem zkoumal možnosti redukce komplexnosti kostry a vliv tohoto omezení na výsledky algoritmu DTW. Před omezením byla odečtením souřadnic páteře od všech dat eliminována možná chyba z důvodu rozdílné pozice znakujícího v prostoru. Tento krok má naprosto zásadní vliv na úspěšnost algoritmu, které kombinují převzorkování a porovnání vybranou metrikou. Rozdíly úspěšnosti po tomto přepočtení se pohybují v řádech desítek procent.

Vzhledem k tomu, že pohyb paží a prstů nese při znakování převážnou část významu [34] a bez kterého by pro pozorovatele bylo velice obtížně, ne-li nemožné význam projevu

pochopit, zahrnuje první analýza omezení dat na paže a tzv. tvar ruky (neboli handshape). Vynechány byly tedy hlava, ramena a páteř.

V dalším omezení kostry bylo provedeno omezení pouze na paže. Zahrnuty byly tedy tyto části (viz. obrázek č. 3.2):

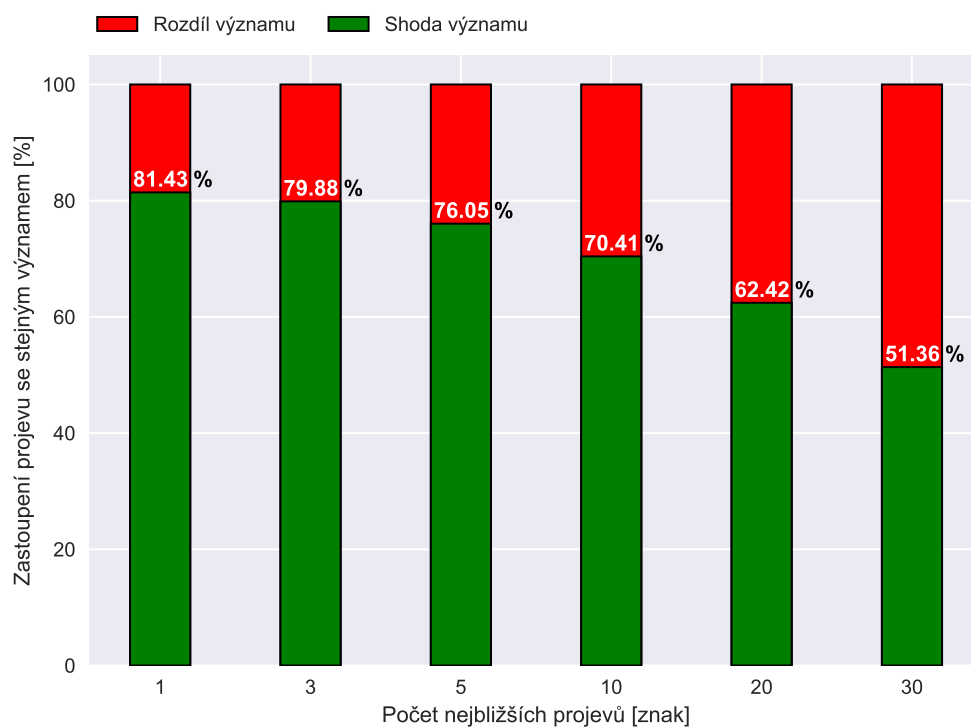
- RightArm (pravá pažní kost)
- RightForeArm (pravé předloktí)
- RightHand (pravá ruka)
- LeftArm (levá pažní kost)
- LeftForeArm (levé předloktí)
- LeftHand (levá ruka)



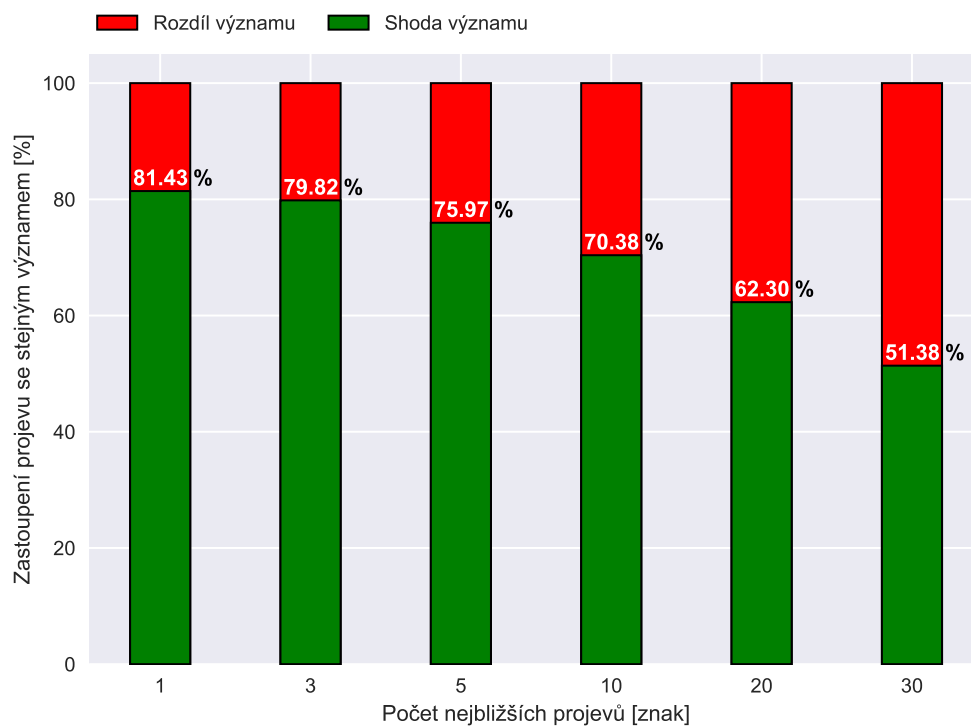
Obrázek 3.2: Vybrané klouby v programu Blender

Prsty během znakování tvoří tzv. tvar ruky. Vzhledem k tomu, že existují znaky ZJ, pro které tvar ruky nese převážnou část významu, měl by na těchto znacích být vidět jednoznačný rozdíl oproti předchozímu omezení. Jako příklad lze uvést číslovky 1-5, které jsou ukazovány pouze na prstech jedné ruky a tím pádem jsou na tvaru ruky (prstech) naprosto závislé.

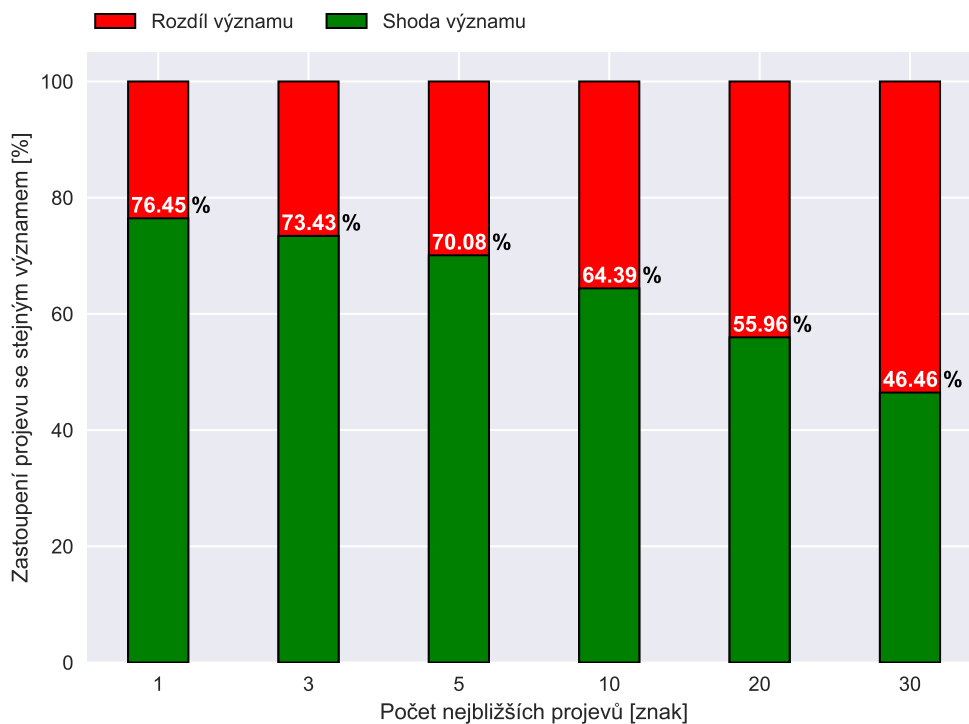
Výsledky analýzy vybraných omezení jsou uvedeny na následujících histogramech (viz. obrázky č. 3.3, 3.4, 3.5) v pořadí: úplná kostra, omezení na paže a prsty a omezení pouze na paže bez prstů.



Obrázek 3.3: Výsledky DTW (úplná kostra)



Obrázek 3.4: Výsledky DTW (omezení kostry na paže a prsty)



Obrázek 3.5: Výsledky DTW (omezení kostry pouze na paže)

Porovnáním výsledků jsou získány úspěšnosti, které pouze potvrzují majoritní význam rukou a prstů při znakování. Pokud z kostry použijeme pouze pohyb paží, rukou a prstů (viz. obrázek č. 3.4) lze pozorovat pokles úspěšnosti oproti testu na pohybech celé kostry (viz. obrázek č. 3.3) v řádu desetin procenta. Tento minimální rozdíl ve výsledcích může být způsoben pohybem specifickým pro daného znakovajícího. Pokud například podvědomě naklání hlavu na stranu nebo zvedá ramena při konkrétním pohybu rukou, tento podvědomý pohyb se ve stejných znacích vyskytuje opakovaně. Z toho důvodu může algoritmus najít závislost, která není podložena konkrétní částí významu projevu. To je ovšem hypotéza, jejíž ověření by vyžadovalo analýzu dat od více řečníků.

Při omezení kostry na variantu bez prstů (viz. obrázek č. 3.5) lze pozorovat pokles úspěšnosti oproti testu předchozího omezení v řádech jednotek procent. Lze tvrdit, že pouze pomocí těchto šesti vybraných částí těla (viz. obrázek č. 3.2) jsme schopni definovat a popisovat jednotlivé znaky ZJ a informace o zbylých částí těla nejsou v naší oblasti testování tolik významné. Rozdíl, který se vyskytuje mezi touto analýzou a analýzou celé kostry, je způsoben znaky závisujícími na tvaru ruky (prstech). V případě jeho vynechání mohou totiž významy jednotlivých znaků na něm závisujících splývat a zdát se totožné. Těchto znaků je ovšem v použitém datasetu velice málo a výslednou úspěšnost ovlivňují minimálně. Pokud by byla cílem redukce dat z důvodu snížení výpočetní doby algoritmu při zachování neporušené většiny významů projevů, tato možnost omezení kostry by byla ideální. Ovšem vzhledem k dostatku výpočetních prostředků a nejlepší zjištěné úspěšnosti

DTW na celé kostře budou následující testy prováděny bez omezení dat.

Toto téma se dá dále rozvinout analýzou experimentálního přiřazení vah jednotlivým částem těla odpovídající jejich důležitosti ve významu projevu. Nejvyšší váhy by náležely šesti částem těla z prvního omezení. Menší váhy by náležely tvaru ruky (prstům). Hlava, ramena a páteř by poté měly váhy zanedbatelné, protože jak bylo zjištěno, jejich význam je naprosto minimální až žádný.

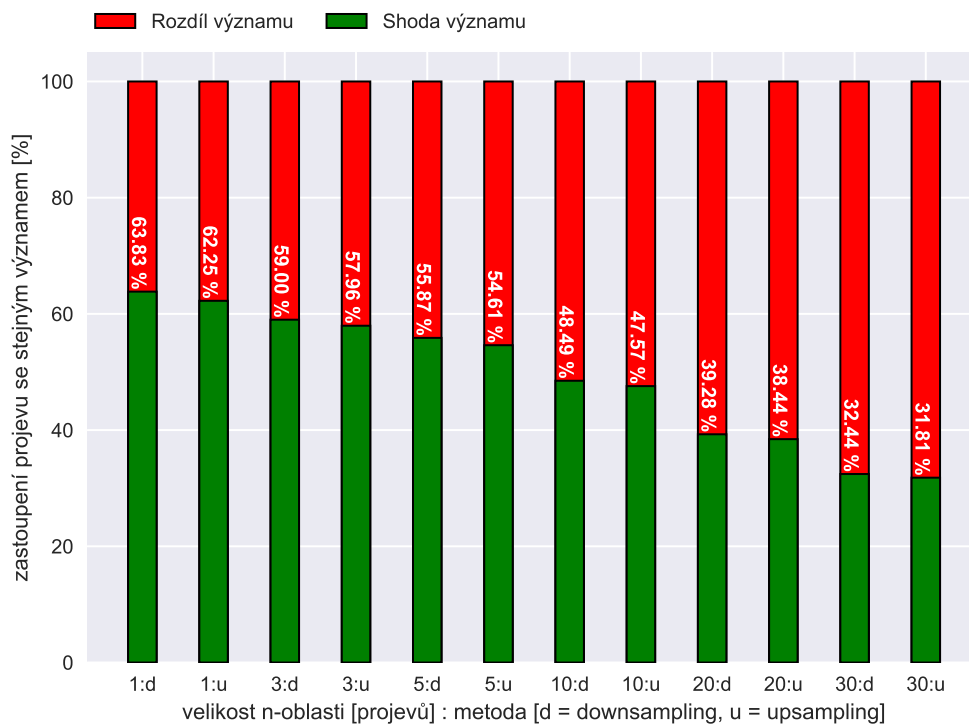
### 3.3 Možnosti převzorkování

Některé metody porovnání vyžadují jako vstup signály stejné délky, proto je nutné data na stejnou délku převzorkovat. Při ekvidistantním převzorkování probíhá zrychlení (tzv. upsampling) popřípadě zpomalení (tzv. downsampling) jednoho ze dvou porovnávaných projevů. Tato změna nemá vliv na význam znaku, může ale hrát roli v rozpoznání znaku člověkem nebo ve vnímané přirozenosti pohybu znakujícího.

Operaci převzorkování je možno provádět různými způsoby. Existuje několik metod převzorkování, které byly analyzovány (více viz. kapitoly 3.3.3, 3.3.2). Záleží také na výběru upravovaného signálu z dvojice porovnávaných, obzvlášť na jejich délce a výběru na jejím základě viz. kapitola 3.3.1.

#### 3.3.1 Upsampling/downsampling

Byly testovány dvě možnosti průběhu algoritmu. Těmi jsou downsampling (zrychlení znaku) neboli převzorkování delšího ze dvou porovnávaných signálů na počet snímků kratšího a upsampling (zpomalení znaku) neboli převzorkování kratšího signálu na počet snímků signálu delšího. Byly provedeny testy s těmito zvolenými variantami algoritmu: převzorkování lineární interpolací v kombinaci s porovnáním Eukleidovskou metrikou. K porovnání je uveden hromadný histogram (viz. obrázek č. 3.6) pro všechny 3 varianty).



Obrázek 3.6: Úspěšnost downsamplingu a upsamplingu

Ve všech případech zvolených n-oblastí dosáhla lepší výsledků varianta downsamplingu. Z toho důvodu bude používána v následujících analýzách a testech se samostatně prováděným převzorkováním.

### 3.3.2 Polynomiální interpolace

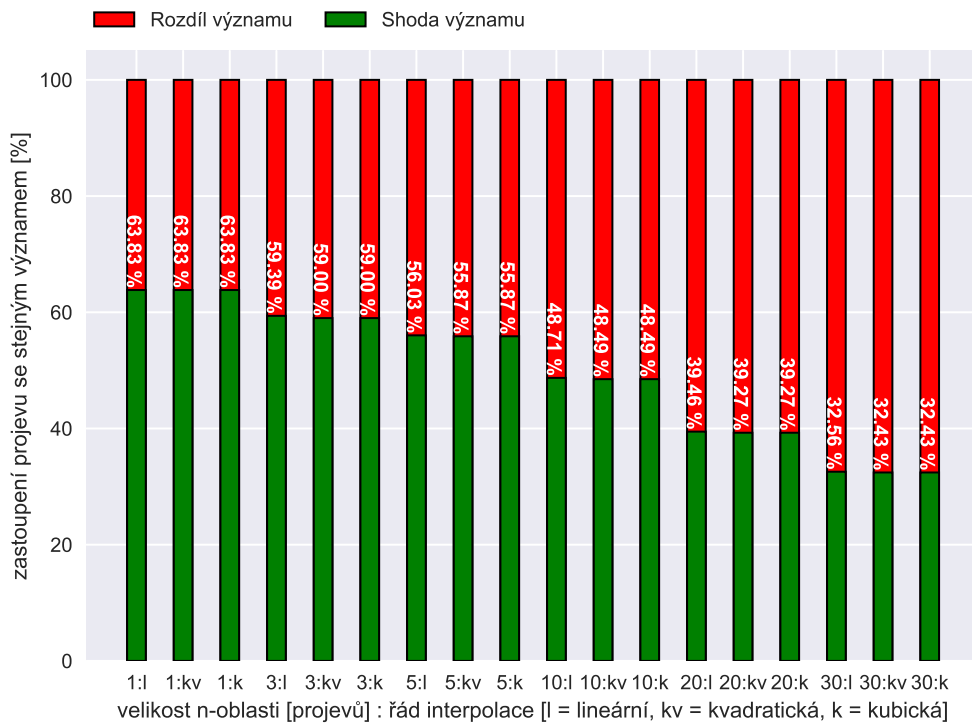
Interpolaci lze dělit dle řádu složitosti její polynomiální aproximační funkce viz. kapitola 2.5.3. V této části analýzy jsou porovnány různé řády aproximačních funkcí a jejich vliv na změnu úspěšnosti algoritmu. Výsledky rozdílných řádů mohou být reprezentovány validačními histogramy. Histogramy ovšem berou v potaz data až po porovnání vybranou metrikou. Výsledky tedy mohou být výběrem metriky ovlivněny a z toho důvodu nejsou pro interpolaci vhodné. Proto bude použit způsob validace výpočtem DTW vzdálenosti původního a interpolovaného signálu. Řád interpolace s nejmenší vypočtenou průměrnou DTW vzdáleností přes všechna data je vybrán jako optimální řešení a je využit v následujících analýzách.

Výsledky druhů interpolací jsou:

Řád	Průměrná vzdálenost
Lineární	1.09748
Kvadratická	1.09980
Kubická	1.09986

Tabulka 2: Tabulka ohodnocení řádů interpolací

K porovnání je uveden hromadný histogram pro všechny tři porovnávané řády:



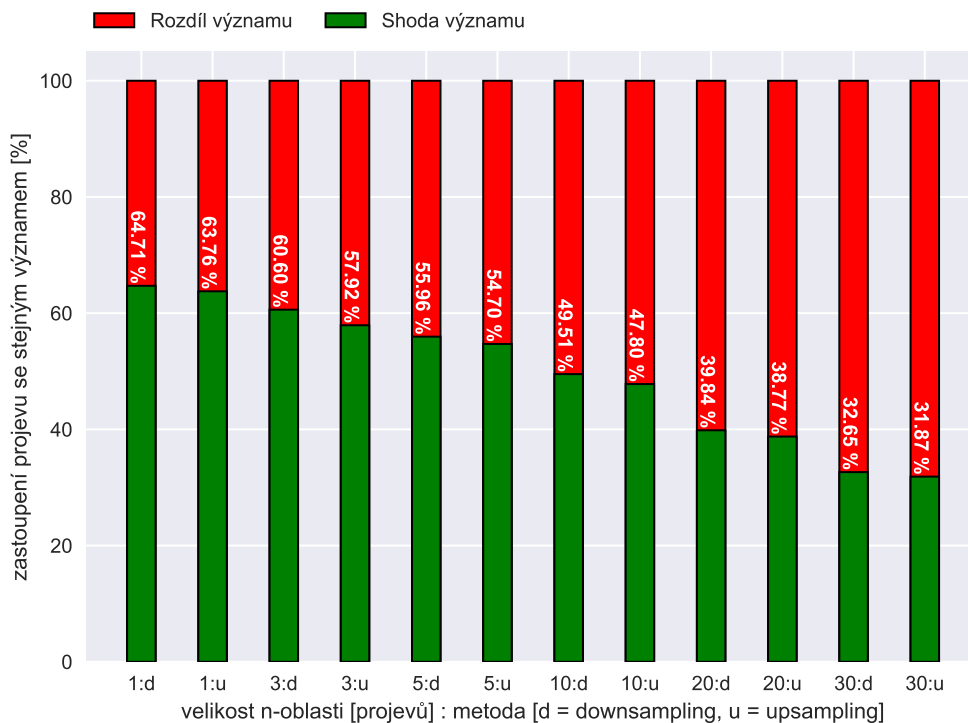
Obrázek 3.7: Výsledky řádů interpolace

Shodné chování výsledků v histogramu s výsledky druhého způsobu validace dává informaci o tom, že použití dané metriky a analýza výsledků pomocí histogramu (tedy až po její aplikaci) nemá na výsledky žádný vliv.

Dle získaných průměrných vzdáleností vypočtených algoritmem DTW mezi signálem před a po provedení dané interpolace bylo zjištěno, že všechny řády vykazují téměř stejné výsledky (viz. tabulka č. 2). Přesto poskytuje lineární interpolace i přes její největší jednoduchost pro tuto analýzu nejlepší výsledky. Její průměrná DTW vzdálenost mezi interpolovaným a původním signálem je nižší než vzdálenost kvadratické i kubické interpolace. Proto bude v následujících analýzách použita lineární aproximační funkce, která je zároveň výpočetně nejjednodušší.

### 3.3.3 Fourierova transformace

Úspěšnost Fourierovy transformace je z důvodu periodické podstaty aproximační funkce závislá na tom, zda je použit downsampling nebo upsampling. Použití downsamplingu dosahuje lepších výsledků z toho důvodu, že s menším množstvím funkčních hodnot aproximační funkce se dostáváme na nižší pravděpodobnost zavedení chyby vybráním funkční hodnoty při kmitu aproximační funkce. Oproti tomu se při upsamplingu vyskytují ve výsledném převzorkovaném signálu funkční hodnoty aproximační funkce s velkou hustotou. Z toho plyne větší pravděpodobnost aproximace chybných kmitů aproximační funkce a také větší chybovost. Na obrázku č. 3.8 je vidět porovnání úspěšnosti Fourierovy transformace při použití downsamplingu a upsamplingu. Počítáno bylo s vybranou Eukleidovskou metrikou.



Obrázek 3.8: Fourierova transformace - downsampling a upsampling

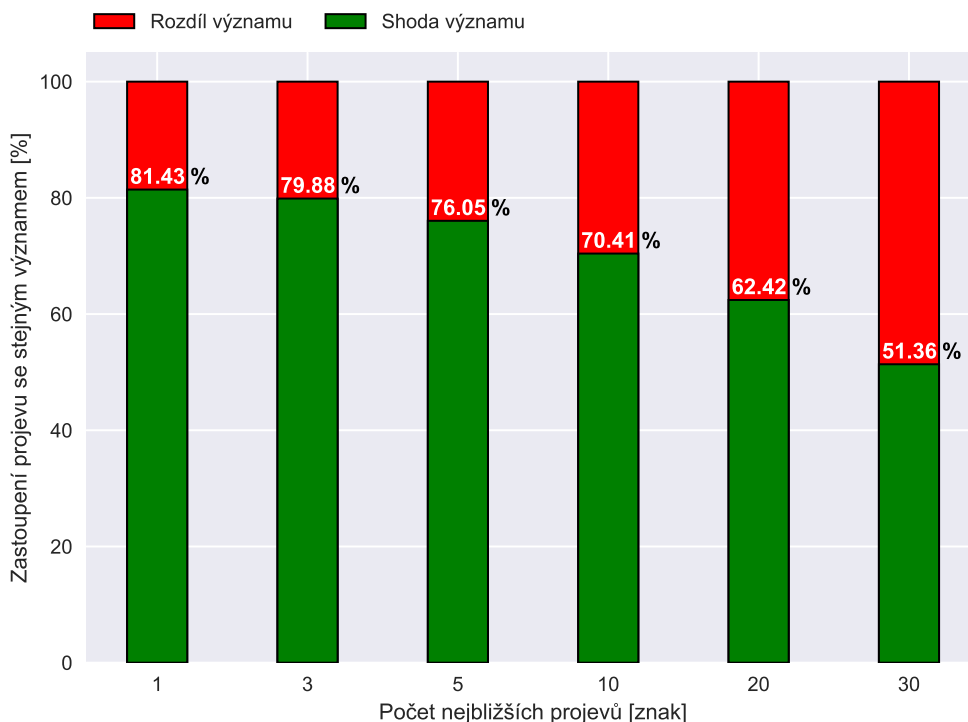
Z této analýzy byl získán podobný výsledek jako z analýzy předchozí kapitoly. Bylo zjištěno, že převzorkování signálu Fourierovou transformací je i přes její periodickou podstatu použitelné a lepší výsledky vykazuje při použití downsamplingu (převzorkování delšího signálu na délku kratšího). Tato metoda bude použita při následujících analýzách zahrnujících Fourierovu transformaci.



### 3.4 DTW

DTW algoritmus je navržený pro porovnání signálů rozdílné délky, proto jsou jeho výsledky zvoleny jako baseline analýz této práce (viz. kapitola 2.5.1). Dá se předpokládat, že úspěšnost jeho aplikace na tento problém bude poměrně vysoká.

Pro výpočet vzdálenosti DTW byla použita knihovna `dtaidistance` [32], která poskytuje možnost výpočtu na N-dimenzionálních vstupních datech optimalizovaným kódem v jazyce C. Jeho výpočetní rychlost je například při srovnání s knihovnou `dtw-python`, která je implementována pouze v jazyce python, přibližně desetkrát vyšší. Dle předchozích analýz jsou výsledky DTW algoritmu počítány z celé kostry znakových. Výsledky této metody jsou následující:



Obrázek 3.9: Výsledky algoritmu DTW

Zjištěné hodnoty tento předpoklad potvrzují. Tato úspěšnost bude nadále porovnávána s úspěšností ostatních typů algoritmů.

### 3.5 Testování metod

U obou vybraných technik převzorkování signálu (viz. kapitoly 3.3.2, 3.3.3) je vybrána metoda podání a zpracování dat, jejíž úspěšnost dosahuje nejvyšších hodnot. To znamená, že u obou vzorkovacích metod je využit `downsampling` a v případě interpolace je využita interpolace lineární. Takto zvolené metody převzorkování jsou v této kapitole kombinovány s možnostmi výpočtu vzdálenosti mezi dvěma signály pomocí metrik

představených v kapitole 2.5.4. V kombinaci s metrikami je tedy analyzována lineární interpolace a Fourierova transformace.

### 3.5.1 Kombinace s lineární interpolací

Výsledky kombinací metrik a lineární interpolace jsou zobrazeny v tabulce č. 3. Tabulka je seřazena dle výsledného souhrnného hodnocení  $K$  jednotlivých metrik. Hodnocení je definováno následovně:

$$K_{metrika} = K_1 + K_3 + K_5 + K_{10} + K_{20} + K_{30}$$

Jednotlivá partikulární hodnocení  $K_x$ , kde  $x$  nabývá jedné z hodnot 1, 3, 5, 10, 20, 30 jsou poté definována dle pořadí dané metriky pro danou  $n$ -oblast, tedy první nejlepší metrice pro  $n$ -oblast je přiřazen koeficient 1, druhé nejlepší 2 apod.

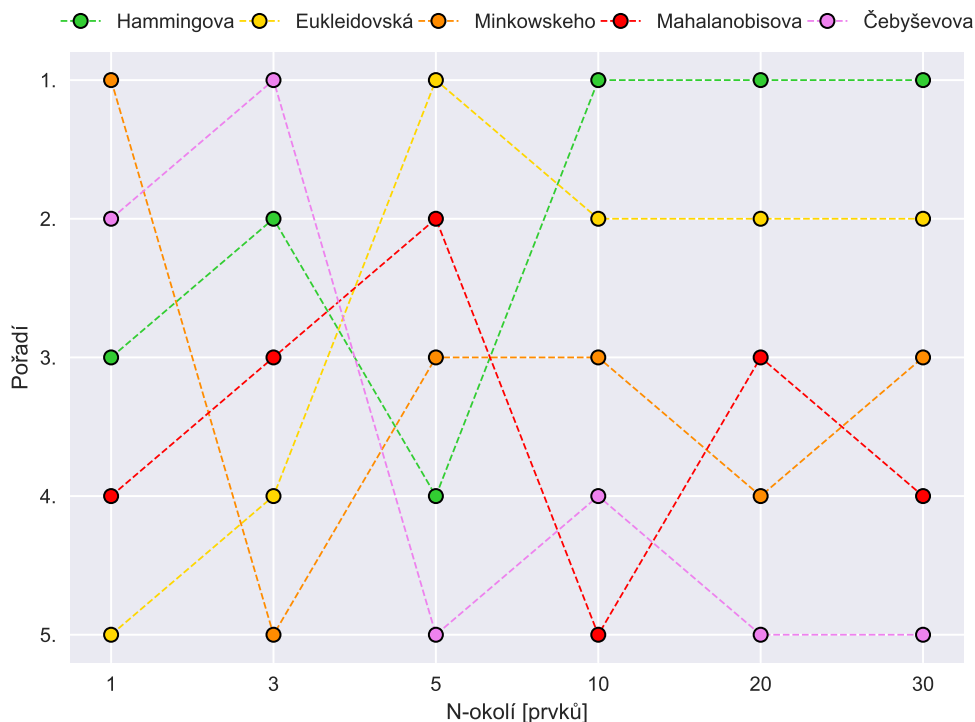
Již z definice každé metriky vyplývá různá citlivost na různé situace. Canberrská metrika je například nejcitlivější na malé změny souřadnic pokud se oba body nacházejí v počátku souřadnicového systému. Naopak pro velké změny souřadnic mezi porovnávanými body se její citlivost snižuje. Tato skutečnost odůvodňuje různé nejlepší metriky pro různé velikosti  $n$ -oblastí. Odstín podbarvení reprezentuje pořadí metriky v dané  $n$ -oblasti (v pořadí zelená, žlutá, oranžová, červená, fialová).

n-oblast [projevů]/ použitá metrika	Zastoupení projevu se stejným významem [%]						K
	1	3	5	10	20	30	
<b>Hammingova</b>	63,95	59,79	55,45	49,24	39,53	32,67	12
<b>Eukleidovská</b>	63,83	59,39	56,03	48,71	39,46	32,56	16
<b>Minkovského (p=3)</b>	64,31	59,25	55,63	48,66	39,24	32,44	19
<b>Mahalanobisova</b>	63,85	59,6	55,75	48,37	39,29	32,21	21
<b>Čebyševova</b>	64,26	59,98	55,12	48,38	38,8	31,88	22
<b>Brayova-Curtisova</b>	60,49	55,92	52,62	45,79	36,84	30,15	-
<b>Canberrská</b>	52,29	48,07	45,07	39,48	32,5	26,62	-

Tabulka 3: Výsledky metrik v kombinaci s lineární interpolací

Výsledky metrik jsou všechny poměrně srovnatelné vyjma Brayovy-Curtisovy a Canberrské, které vykazují celkově horší úspěšnost s řádem rozdílů v jednotkách procent. Vzhledem k rozdílnosti nejlepších metrik pro dané  $n$ -oblasti jsou výsledky poměrně vyrovnané a každá metrika je minimálně jednou v nějaké  $n$ -oblasti nejlepší. Pokud bychom ale museli vybrat dvě nejlepší, budou to Eukleidovská a Hammingova metrika.

Jejich měnící se pořadí na základě úspěšnosti je zobrazeno na obrázku č. 3.10.



Obrázek 3.10: Graf pořadí metrik při lineární interpolaci

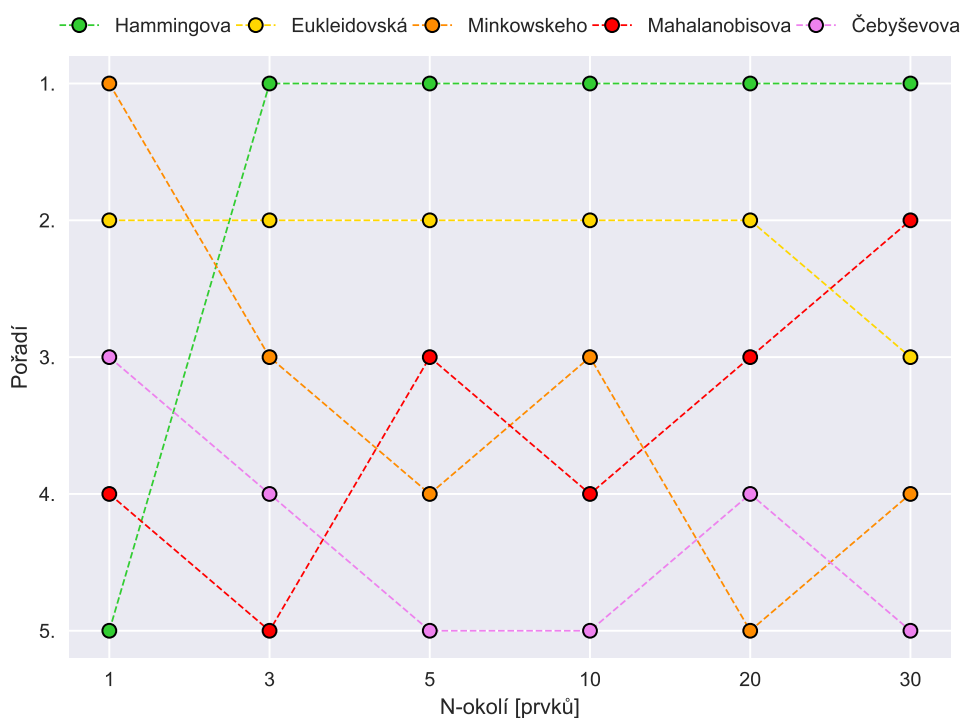
### 3.5.2 Kombinace s Fourierovou transformací

Výsledky kombinací všech metrik a Fourierovy transformace jsou zobrazeny v tabulce č. 4. Tabulka je seřazena dle hodnoty K popsaného v předchozí kapitole.

n-oblast [projevů]/ použitá metrika	Zastoupení projevu se stejným významem [%]						K
	1	3	5	10	20	30	
<b>Hammingova</b>	63,36	60,64	56,72	50,11	40,32	33,09	10
<b>Eukleidovská</b>	64,85	60,44	56,18	49,68	39,92	32,69	13
<b>Minkovského (p=3)</b>	65,03	60,25	55,92	49,31	39,6	32,58	19
<b>Mahalanobisova</b>	64,14	59,35	55,98	49,22	39,66	32,71	21
<b>Čebyševova</b>	64,82	60,17	55,81	49,05	39,63	32,14	26
<b>Brayova-Curtisova</b>	59,01	57,13	53,72	46,59	37,6	30,56	-
<b>Canberrská</b>	53,63	47,73	45,66	40,11	33,25	27,07	-

Tabulka 4: Výsledky metrik v kombinaci s Fourierovou transformací

Měnící se pořadí metrik na základě úspěšnosti je zobrazeno na obrázku č. 3.11.



Obrázek 3.11: Graf pořadí metrik při Fourierově transformaci

Při kombinaci analyzovaných metrik s Fourierovou transformací lze sledovat podobné výsledky jako u kombinace s lineární interpolací. Na úspěšnostech jednotlivých metrik lze ovšem pozorovat jejich hromadné zvýšení. Rozdíly jsou uvedeny v následující tabulce. Hodnota této tabulky na pozici  $[x, y_m]$  je získána jako  $[x_B - x_A, y_{mB} - y_{mA}]$ , kde A je tabulka úspěšností metrik po lineární interpolaci, B je tabulka úspěšností metrik po Fourierově transformaci a m značí řádek, kde se vyskytuje daná metrika. Kladné hodnoty tedy značí růst, záporné hodnoty naopak pokles úspěšnosti při změně metody převzorkování.

n-oblast [projevů]/ použitá metrika	Změna úspěšnosti [%]					
	1	3	5	10	20	30
<b>Eukleidovská</b>	1,02	1,05	0,15	0,97	0,46	0,13
<b>Čebyševova</b>	0,56	0,19	0,69	0,67	0,83	0,26
<b>Minkovského</b>	0,72	1	0,29	0,65	0,36	0,14
<b>Mahalanobisova</b>	0,29	-0,25	0,23	0,85	0,37	0,5
<b>Hammingova</b>	-0,59	0,85	1,27	0,87	0,79	0,42
<b>Brayova-Curtisova</b>	-1,48	1,21	1,1	0,8	0,76	0,41
<b>Canberrská</b>	1,34	-0,34	0,59	0,63	0,75	0,45

Tabulka 5: Změna úspěšnosti při použití Fourierovy transformace

S Fourierovou transformací jsou všechny metriky minimálně v pěti ze šesti testovaných n-oblastí úspěšnější.

Úspěšnosti jednotlivých metrik jsou vyjma Brayovy-Curtisovy a Canberrské opět poměrně vyrovnané, v popředí zůstává metrika Hammingova, na druhé místo se opět řadí metrika Eukleidovská, a to s poměrně velkým rozdílem oproti další Minkovskeho metrice. Jejich procentuální rozdíly se ovšem opět pohybují v jednotkách procent, proto nelze brát výběr nejlepších dle koeficientu  $K$  jako jednoznačné kritérium.

Nejúspěšnější kombinací převzorkování a porovnání signálů je tedy Fourierova transformace v kombinaci s jednou z pěti metrik, to jsou: Minkowskeho, Eukleidovská, Čebyševova, Mahalanobisova a Hammingova. Výsledky těchto kombinací budou porovnány s úspěšnými výsledky ostatních typů algoritmu.

### 3.5.3 Pearsonův korelační koeficient

Výpočet korelačních koeficientů obsahuje na rozdíl od ostatních metrik odečítání střední hodnoty daného signálu. Výsledná hodnota není získána odečítáním souřadnic bodů od sebe navzájem (jako u všech ostatních metrik). To je důvodem k uvedení metody i jejich výsledků samostatně v této kapitole.

n-oblast [projevů]/ metoda převzorkování	Zastoupení projevu se stejným významem [%]					
	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>20</b>	<b>30</b>
<b>Lineární interpolace</b>	72,2	66,52	61,9	53,46	41,82	32,9
<b>Fourierova transformace</b>	72,78	66,77	62,66	53,81	42,29	33,19

Tabulka 6: Výsledky Pearsonova korelačního koeficientu

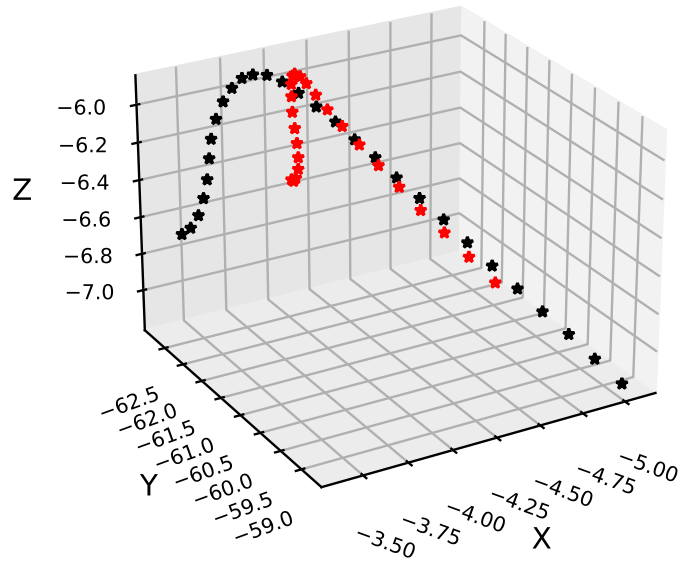
Pearsonův korelační koeficient jak v kombinaci s lineární interpolací, tak s Fourierovou transformací dosahuje nejlepších výsledků ze všech metrik. Podrobnější analýzou, jejíž průběh je popsán v dalším odstavci, byl zjištěn nejspíše majoritní důvod k dosažení této úspěšnosti.

Pro několik slov byly spočítány výsledky Eukleidovské metriky a Pearsonova korelačního koeficientu kombinovaných s Fourierovou transformací. U obou metrik bylo pro testovaná slova získáno 20 nejbližších projevů. Byly zpětně získány trajektorie, ze kterých byly vzdálenosti vypočteny. Následně byla pro tyto trajektorie vypočtena průměrná DTW vzdálenost od referenčního projevu pro obě metriky zvlášť. Výsledné hodnoty podpořily domněnku o důvodu úspěšnosti Pearsonova korelačního koeficientu.

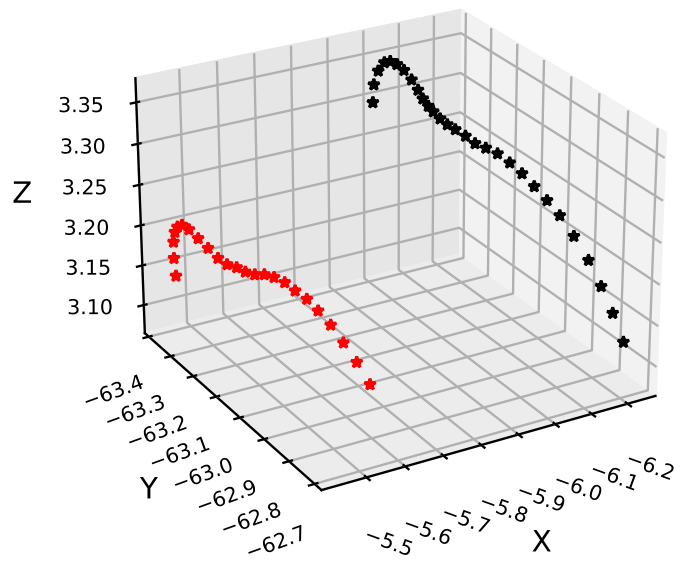
Průměrná DTW vzdálenost trajektorií nejbližších projevů získaných pomocí Eukleidovské vzdálenosti byla totiž téměř dvojnásobně menší, než DTW vzdálenost trajektorií po použití Pearsonova korelačního koeficientu. Pearsonův korelační koeficient tedy dokázal rozpoznat i trajektorie, byť významově stejné, avšak se vzdálenější lokací znakování. Příklady dvou trajektorií se stejným významem znakování, ale rozdílnou lokací znakování, jsou uvedeny na obrázcích č. 3.12 a 3.13.

- ★ 1. instance slova zítra
- ★ 2. instance slova zítra

### Pravé zápěstí



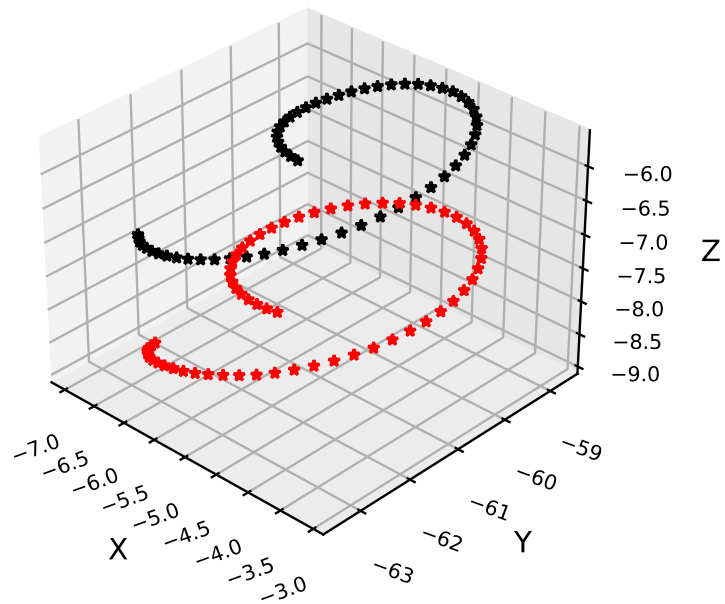
### Pravá pažní kost



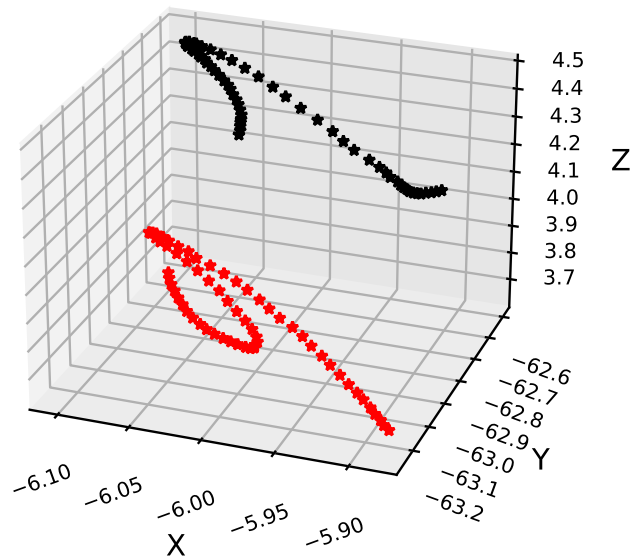
Obrázek 3.12: Trajektorie s posunem lokace znakování při znakování slova „zítra“

- ★ 1. instance slova teplo
- ★ 2. instance slova teplo

### Pravé zápěstí



### Pravá pažní kost



Obrázek 3.13: Trajektorie s posunem lokace znakování při znakování slova „teplo“

Důvodem tohoto jevu je skutečnost, že tato metrika oproti ostatním ignoruje důležitost posunu porovnávaných trajektorií vůči sobě navzájem (lokaci znakování). To vyplývá z její definice, kde je během výpočtu odečtena z celého signálu jeho střední hodnota (viz. kapitola 2.5.4). Díky tomu pracuje převážně s „dynamikou dat“ a dokáže poskytnout lepší výsledky porovnání.

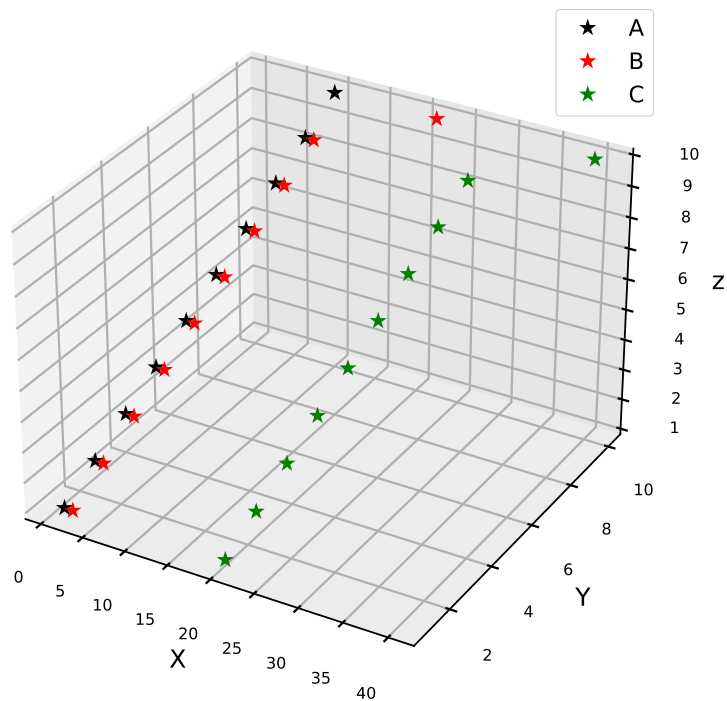
Pro demonstraci účinku posunu bylo provedeno porovnání Pearsonova korelačního koeficientu a Eukleidovské metriky na 3D umělých datech. Jejich souřadnice v prvním kanále jsou definovány listy A,B a C.

$$A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$$

$$B = [2, 3, 4, 5, 6, 7, 8, 9, 10, 22]$$

$$C = [20, 21, 22, 23, 24, 25, 26, 27, 28, 40]$$

Trend B a C je identický, rozdíl je pouze v posuvu na ose x, jak je zobrazeno na obrázku č. 3.14.



Obrázek 3.14: Testovací data



Po kalkulaci Pearsonova korelačního koeficientu a Eukleidovské metriky na dvojicích A,B a A,C lze prezentovat tyto výsledky:

metrika/ porovnávaná data	Výsledek dané metriky	
	<b>Eukleidovská metrika</b>	<b>Pearsonův kor. koef.</b>
<b>A,B</b>	2.1	0.85
<b>A,C</b>	20.1	0.85

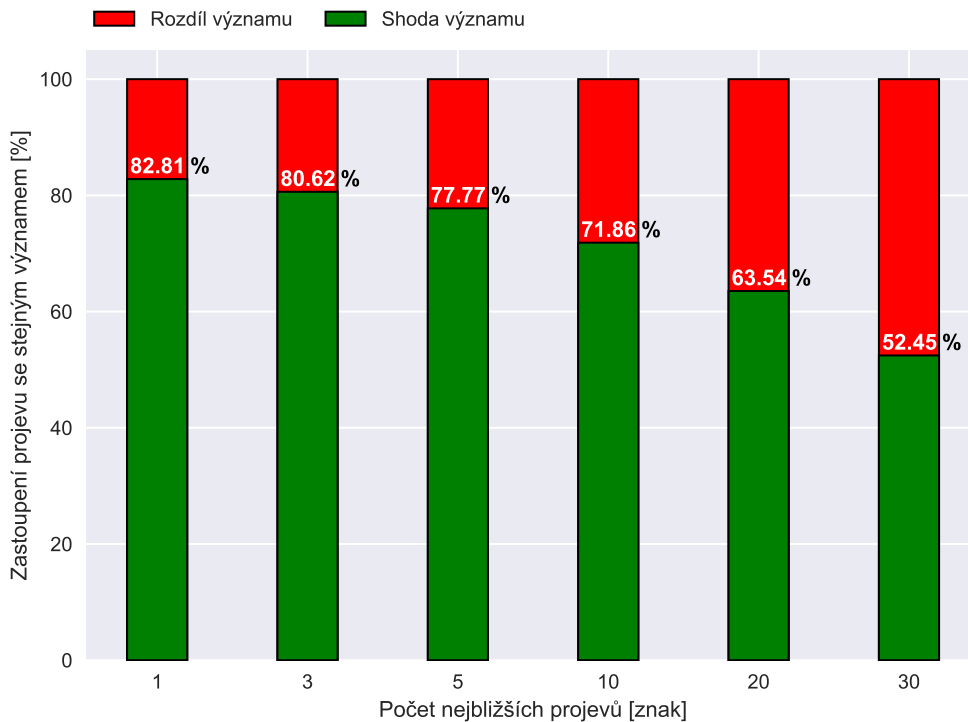
Tabulka 7: Změna Pearsonova kor. koef. a Eukleidovské metriky při posunu dat

Eukleidovská metrika počítá absolutní vzdálenost daných bodů signálu ve 3D prostoru. Oproti tomu Pearsonův korelační koeficient cílí díky odečítání střední hodnoty z celého datasetu více na jeho samotnou změnu v čase a vykazuje lepší výsledky při posunu dat. Z toho důvodu rozpozná lépe podobnost trajektorie části těla oproti druhé, která je umístěna jinde v prostoru, ovšem se stejným cíleným významem. Žádná z ostatních analyzovaných metrik neobsahuje ve své definici krok s cílem většího důrazu na porovnávání „dynamiky dat“ než jejich přímé polohy v prostoru, proto jsou výsledky při použití Pearsonova korelačního koeficientu lepší než všech ostatních kombinací převzorkování a porovnání.

Ignorování lokace znakování, na základě kterého tato metoda z definice funguje, může být ovšem také potenciálním problémem v ZJ. Pro dva projevy v ZJ, při kterých části těla vykonávají stejnou nebo podobnou trajektorii a jejichž význam se odvíjí až od lokace znakování, je možné, že budou po ignorování lokace vyhodnoceny jako projevy stejného významu. To může být jedním z potenciálních problémů při použití korelačních koeficientů, se kterým je nutno počítat.

### 3.5.4 SoftDTW

Algoritmus SoftDTW, jehož rozdíl oproti DTW spočívá v jinak definované a navíc diferencovatelné ztrátové funkci, má již díky své podstatě v DTW předpoklad pro vysokou úspěšnost analýzy. Na jeho implementaci v kódu bylo využito knihovny soft-dtw, jejímž autorem je autor samotné modifikace DTW do SoftDTW. Knihovna je dostupná na platformě GitHub [21]. Výsledky analýzy tohoto algoritmu zobrazuje následující graf.



Obrázek 3.15: Výsledky algoritmu SoftDTW

Algoritmus SoftDTW dosahuje zatím nejlepších výsledků ze všech testovaných typů algoritmů, zde se nabízí porovnání úspěšnosti DTW a SoftDTW, které je uvedeno v tabulce č. 8.

n-oblast [projevů]/ algoritmus	Zastoupení projevu se stejným významem [%]					
	1	3	5	10	20	30
<b>DTW</b>	81,43	79,88	76,05	70,41	62,42	51,36
<b>SoftDTW</b>	82,81	80,62	77,77	71,86	63,54	52,45

Tabulka 8: Výsledky algoritmů DTW a SoftDTW

SoftDTW dosahuje ve všech zvolených n-oblastech lepšího výsledku než vybraná baseline DTW. Tento malý rozdíl může být samozřejmě způsoben specifitou používaného datasetu. Proto lze tvrdit, že úspěšnost těchto dvou metod je spíše srovnatelná.

### 3.6 Výsledné zhodnocení

Zjištěné nejúspěšnější kombinace převzorkování a porovnání, Pearsonův korelační koeficient a SoftDTW jsou v tabulce č. 9 porovnány s výsledky DTW. Typy algoritmů v tabulce jsou opět seřazeny dle hodnoty K definované v kapitole 3.5.1.

n-oblast [projevů]/ typ algoritmu	Zastoupení projevu se stejným významem [%]						K
	1	3	5	10	20	30	
<b>SoftDTW</b>	82,81	80,62	77,77	71,86	63,54	52,45	6
<b>DTW</b>	81,43	79,88	76,05	70,41	62,42	51,36	12
<b>Pearsonův kor. koef.</b>	72,78	66,77	62,66	53,81	42,29	33,19	18
<b>Hammingova</b>	63,36	60,64	56,72	50,11	40,32	33,09	28
<b>Eukleidovská</b>	64,85	60,44	56,18	49,68	39,92	32,69	31
<b>Minkovského (p=3)</b>	65,03	60,25	55,92	49,31	39,6	32,58	37
<b>Mahalanobisova</b>	64,14	59,35	55,98	49,22	39,66	32,71	39
<b>Čebyševova</b>	64,82	60,17	55,81	49,05	39,63	32,14	44

Tabulka 9: Celkové srovnání nejlepších typů testovacího algoritmu

Nejvyšší úspěšnosti dosáhl algoritmus SoftDTW. Překonal dokonce i baseline (výsledky DTW) s rozdílem pohybujícím se kolem 1-2 %. Tento malý rozdíl může být ovšem způsoben specifičností použitého datasetu.

Druhý nejúspěšnější testovaný typ algoritmu je kombinace Fourierovy transformace a Pearsonova korelačního koeficientu. Dosahuje přibližně o 10-15 % nižší úspěšnosti než SoftDTW, ovšem stále lepších výsledků než ostatní testované metriky.

Ostatní kombinace metrik a Fourierovy transformace dosahují maximální úspěšnosti přibližně kolem 64 %. Tato úspěšnost nemůže být konkurencí pro úspěšnosti SoftDTW ani Pearsonova korelačního koeficientu, mohla by ovšem dosahovat lepších výsledků při případné modifikaci výpočtu (dále viz. kapitola 3.9).

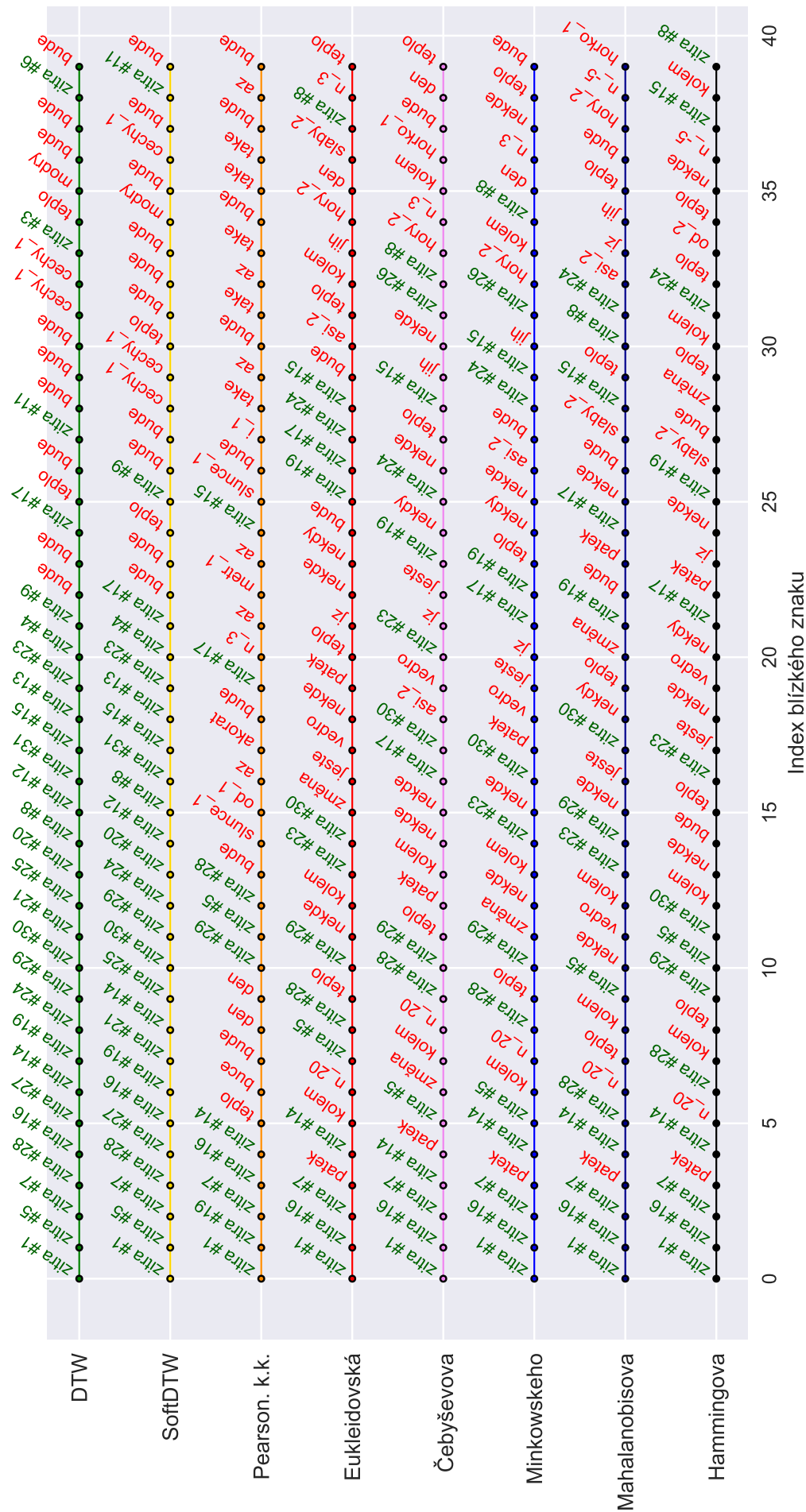
### 3.7 Případová studie

V této kapitole chci ukázat, jak vypadají výsledky analyzovaných typů algoritmu pro konkrétní testovaný projev.

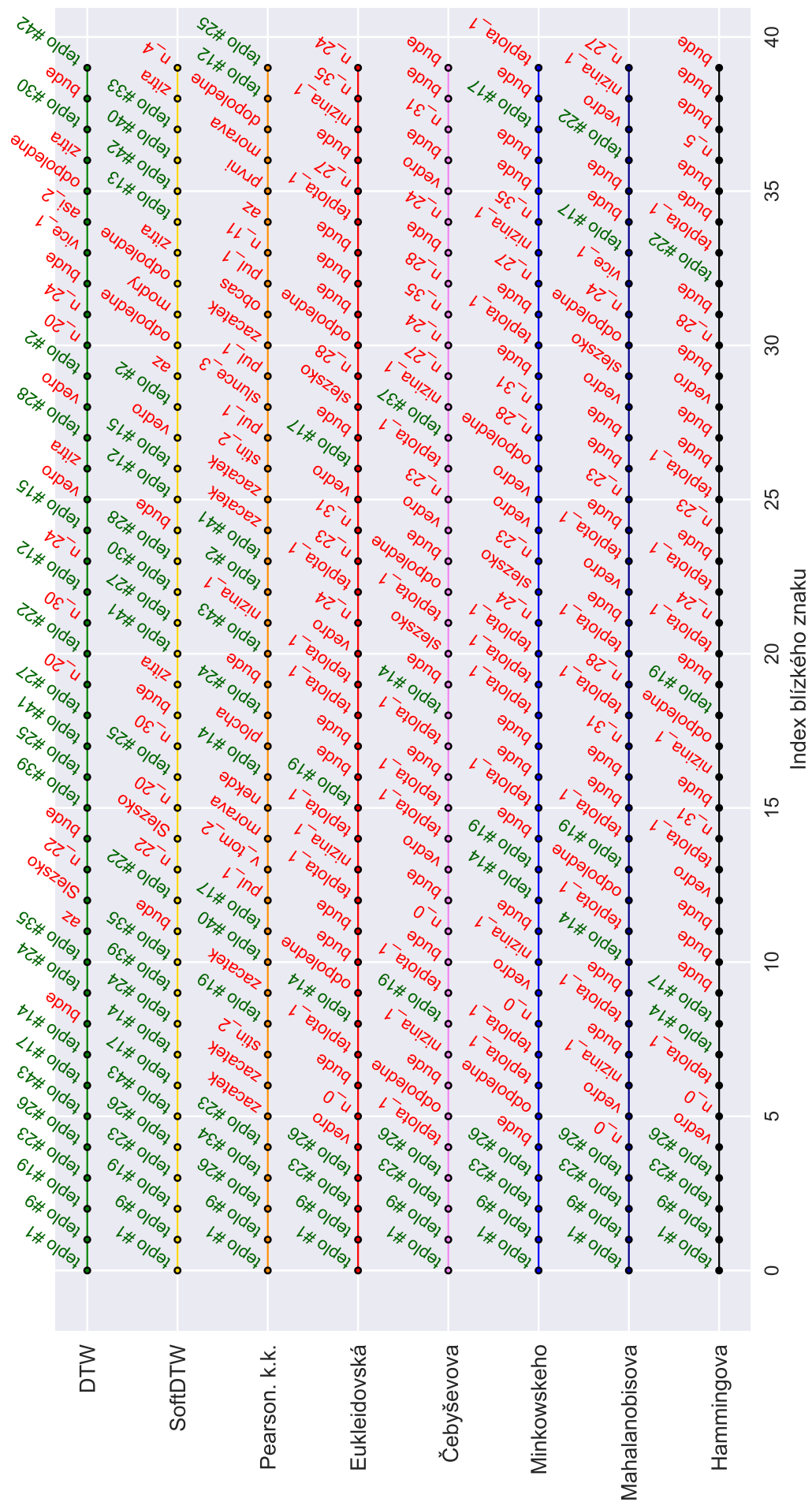
Pro analýzu byl vybrán znak „zítra“, který má v datasetu 32 instancí. Výsledných 40 znaků, které byly jednotlivými typy algoritmu vyhodnoceny jako nejbližší, jsou uvedeny na obrázku č. 3.16.

Graf na obrázku jednoznačně potvrzuje největší účinnost algoritmů DTW a SoftDTW a uvádí příklad větší úspěšnosti SoftDTW při jejich porovnání. Do čtyřiceti nejbližších slov bylo SoftDTW schopno zahrnout o tři instance znaku „zítra“ více než DTW. Následuje Pearsonův korelační koeficient, jehož úspěšnost je nižší a dosahuje přibližně 70 % oproti předchozím 80 % pro DTW a SoftDTW. Na grafu je zřejmý velký rozdíl mezi množstvím výskytů rozpoznávaných instancí slova „zítra“ oproti předchozím typům algoritmů. U dalších analyzovaných metrik se množství nadále zmenšuje a data jsou čím dál více „zašuměná“.

Byl proveden ještě jeden stejný test, tentokrát pro znak „teplo“ s počtem instancí 44. Jeho výsledky jsou na obrázku č. 3.17. Opět lze pozorovat velký rozdíl výsledků DTW a SoftDTW oproti ostatním metrikám. Znak „teplo“ má celkově horší úspěšnost rozpoznání než znak „zítra“ u všech typů algoritmu. Možných důvodů je více. Lze odhadnout, že určitý vliv mohlo mít natočení trajektorie znakování nebo celkově větší rozdíly v trajektoriích jednotlivých instancí tohoto znaku a nedostatečná stabilita znakování při vykonávání této trajektorie opakovaně.



Obrázek 3.16: Porovnání typů algoritmu na analýze slova „zítra“



Obrázek 3.17: Porovnání typů algoritmu na analýze slova „teplo“

### 3.8 Výpočetní doba

Nejdůležitějším kritériem při výběru typu algoritmu je samozřejmě jeho úspěšnost. Kromě úspěšnosti bude ovšem nutné přihlídnout při volbě algoritmu i k jeho výpočetní složitosti.

Výpočetní doba je v analýze této práce aspektem, který nebylo možné ignorovat. I přes moji snahu o optimalizaci všech možných částí kódu se test jednoho typu algoritmu na datech obsahujících přibližně 1000 projevů a zároveň 61 částí těla pohyboval v rozmezí výpočetní doby 4-40 hodin. Bylo tedy poměrně obtížné (někdy až nemožné) provést všechny testy analýzy na osobní pevné výpočetní stanici v rozumném čase. To bylo důvodem k využití vzdálených výpočetních stanic v Praze spravovaných společností CESNET pod názvem MetaCentrum. Díky tomuto přístupu k prakticky neomezenému množství programového a aplikačního vybavení byly výsledky získány bez problému. Výpočetní doba jednotlivých typů testovacích algoritmů byla ovšem i přesto poměrně vysoká. V tabulce č. 10 jsou uvedeny všechny testované typy algoritmů a jejich výsledná potřebná výpočetní doba.

Algoritmus	Výpočetní čas [h]
DTW	4,1
SoftDTW	11,26
Lineární interpolace a Eukleidovská metrika	21,94
Lineární interpolace a Hammingova metrika	32,56
Lineární interpolace a Minkovského metrika	11,27
Lineární interpolace a Čebyševova metrika	8,8
Lineární interpolace a Mahalanobisova metrika	38,2
Lineární interpolace a Canberrská metrika	21,71
Lineární interpolace a Brayova-Curtisova metrika	9,61
Lineární interpolace a Pearsonův korelační koeficient	15,23
Fourierova transformace a Eukleidovská metrika	4,86
Fourierova transformace a Hammingova metrika	7,54
Fourierova transformace a Minkovského metrika	4,21
Fourierova transformace a Čebyševova metrika	3,39
Fourierova transformace a Mahalanobisova metrika	26,19
Fourierova transformace a Canberrská metrika	4,02
Fourierova transformace a Brayova-Curtisova metrika	12,43
Fourierova transformace a Pearsonův korelační koeficient	8,37

Tabulka 10: Celkové srovnání výpočetní doby všech testovaných druhů algoritmu

Při případném výběru nejlepšího algoritmu z testovaných a zároveň nedůležitosti jeho výpočetní doby je tedy jednoznačnou volbou algoritmus SoftDTW, který ve výsledném porovnání dosáhl nejlepších výsledků. Pokud bychom ovšem potřebovali algoritmus s co nejnižší výpočetní dobou, avšak minimální ztrátou v úspěšnosti, je vhodným kompromisem algoritmus DTW. Jeho výpočetní doba je dvakrát až třikrát kratší a úspěšnost oproti

SoftDTW klesne pouze v jednotkách procent (největší pokles pro danou n-oblast je 1,75 %). Další použitelnou alternativou je algoritmus tvořený Fourierovou transformací v kombinaci s Pearsonovým korelačním koeficientem. Výpočetní doba je oproti SoftDTW přibližně dvoutřetinová a úspěšnost klesne přibližně o 10 % pro danou n-oblast. Ostatní testované druhy algoritmu již nepřinášejí žádnou výhodu oproti již zmíněným jak v rychlosti tak přesnosti.

### 3.9 Pokračování ve výzkumu

V celé práci jsou analyzována data ve formě absolutních souřadnic částí těla při znakování. Možnou modifikací analýzy by bylo použít stejné metody na původních angulárních datech, ze kterých byla absolutní přepočtena.

Vzhledem k velkému rozdílu úspěšnosti Pearsonova korelačního koeficientu od ostatních metrik na porovnání signálů a jeho odůvodnění vlastností ignorování posunu lokace znakování by při pokročilé analýze této problematiky určitě mělo význam vyzkoušet ruční ustřednění signálu před jeho porovnáním u ostatních metrik. Pracovalo by se poté více s „dynamikou dat“ a záleželo by na menších rozdílech souřadnic. Tuto hypotézu lze podpořit výsledky jediného testu kombinace Fourierovy transformace a Eukleidovské metriky, kde bylo ustřednění provedeno. V testu byly ustředněny všechny kanály všech částí těla jednotlivě při daném projevu. Výsledné úspěšnosti metody po této změně v přípravě dat se zvýšily a dosahovaly přibližně 64-68 %. Proto by mělo určitě smysl zkusit tento typ výpočtu i na ostatní metriky.

Další možnou variantou této analýzy je otestování získání dalších informací z porovnání signálů kombinováním výstupů z porovnávané „dynamiky dat“ získané ustředněním původních absolutních souřadnic. Absolutní souřadnice by mohly sloužit pouze k předání informace o lokaci znakování. Kombinací těchto dvou zdrojů by mohlo být dosaženo lepších výsledků než nyní, ne-li lepších, než kterých dosáhla metoda SoftDTW. To by změnilo výstup celé této analýzy.

Vzhledem k vysoké variabilitě dalších navrhovaných postupů a množství jejich kombinací již tyto myšlenky nebyly v práci realizovány.



## 4 Závěr

Cílem této práce bylo vybrat optimální metodu na porovnání dvou trajektorií částí těla. Ty byly získány nahráním projevů ZJ zařízením záznamu pohybu (také Motion Capture, MoCap). Výsledná metoda porovnání měla být schopna s co nejvyšší přesností rozlišit, zda dvě porovnávané trajektorie nesly původně stejný význam nebo zda významy těchto projevů byly rozdílné. Na zhodnocení metod byl využit algoritmus Dynamic Time Warping (dále DTW), který byl vyvinut za účelem porovnání dvou nestejně dlouhých signálů. Současným trendem strojového učení je použití neuronových sítí. Gradientní metody neuronové sítě vyžadují diferencovatelnou ztrátovou funkci, což DTW nespĺňuje. Diferencovatelnost byla tedy dalším kritériem pro výběr vhodné metody.

Práce byla řešena ve třech hlavních částech. První částí byla samotná rešerše použitelných metod pro porovnání vektorů a převzorkování dat. Dvě hlavní vybrané metody převzorkování vektorů na stejnou délku byly polynomiální interpolace různých řádů a Fourierova transformace. Dále bylo vybráno několik metod pro porovnání dvou signálů testovaných v následné analýze. Mimo testování samostatných metod na převzorkování a porovnání byla nalezena modifikace algoritmu DTW s pozměněnou definicí tak, aby byla diferencovatelná. Tato metoda s názvem SoftDTW byla zařazena do analýzy také. Teoretickou část práce tedy tvoří úvod do tématu a problematiky ZJ a dále teorie týkající se analyzovaných metod. Praktická část je tvořena jejich analýzou.

Všechny analýzy byly prováděny kódem psaným v jazyce Python a tvoří Praktickou část této práce. Byly analyzovány jednotlivé metody převzorkování a porovnání signálů. U metod převzorkování (dále metriky) byla testována úspěšnost různých způsobů předání dat do těchto algoritmů. Průběžné výsledky potvrdily, že převzorkování delších signálů na délku kratších (také downsampling) je úspěšnější, proto byla tato možnost nadále používána. Při testování různých řádů interpolace dat bylo zjištěno, že i přes očekávanou lepší schopnost aproximace časového průběhu vyššími řády aproximační polynomické funkce, dosáhla s malým rozdílem nejlepších výsledků interpolace lineární (maximální rozdíl lineární a kubické interpolace byl 0,22 % úspěšnosti pro danou n-oblast). Interpolace tohoto řádu je zároveň nejméně výpočetně náročná. Při analýze Fourierovy transformace bylo zjištěno, že dosahuje přibližně srovnatelných výsledků s lineární interpolací, proto byly nadále testovány obě varianty převzorkování.

Analýza metrik v kombinaci s vybranými metodami převzorkování potvrdila větší vhodnost Fourierovy transformace na data trajektorie znakového jazyka. Tato metoda je zároveň ještě méně výpočetně náročná. Sedm testovaných metrik se dostalo nejvýše na úspěšnost 60-64 % zastoupení projevů se shodným významem jako referenční ve vybrané n-oblasti. Pro srovnání lze uvést, že algoritmus DTW, který je použit jako baseline, dosahuje úspěšnosti kolem 80-81 %. Jako nejúspěšnější kombinace metod vyšla Fourierova transformace v kombinaci s Hammingovou metrikou s výsledky dosahujícími přibližně 61-64 %. O pár desetín procent horší byla změna z Hammingovy na Eukleidovskou metriku. Pearsonův korelační koeficient se dostal přibližně na 70% úspěšnost a překonal tím předešlé

metriky. Při bližší analýze jeho výpočtu bylo zjištěno, že toto zlepšení je způsobeno odečítáním střední hodnoty dat při výpočtu. Díky tomu dává větší důraz na „dynamiku“ znaku, než na umístění znaku v prostoru.

Výsledky algoritmu SoftDTW překonaly Pearsonův korelační koeficient a dostaly se na úspěšnost kolem 81-82 %. Překonaly dokonce i výsledky porovnávané metody DTW. Tento malý posun v řádu jednotek procent může být ovšem způsoben specifitou použitého datasetu, proto zatím nelze s jistotou tvrdit, že by SoftDTW bylo obecně schopno dosahovat lepších výsledků na jakýchkoliv datech. S přihlédnutím k výsledku Pearsonova korelačního koeficientu a jeho rozdílu oproti ostatním metrikám pouze v odečtení střední hodnoty z dat při výpočtu, lze tvrdit, že zapojení této operace do testů ostatních metrik by mohlo zvýšit jejich úspěšnost a poskytnout zajímavé výsledky.

Nakonec je nutno zmínit, že jediná metoda Canberrské metriky v kombinaci s Fourierovou transformací potřebovala menší výpočetní čas než algoritmus DTW (4,02 hod. oproti 4,1 hod. u DTW). Její úspěšnost se ovšem pohybuje kolem 50 %, proto nelze tuto metodu rozumně využít. Stále poměrně rychlou alternativou je kombinace Fourierovy transformace a Pearsonova korelačního koeficientu, jejíž výpočetní čas se pohyboval kolem 8,4 hodiny, ovšem úspěšnost dosahovala již hodnot kolem 70 %. Nejúspěšnější metodou je ale SoftDTW s výsledky okolo 81-82 %, její potřebný výpočetní čas byl přibližně 11,3 hodiny, tedy téměř trojnásobný oproti metodě DTW.

## Seznam obrázků

2.1	Příklad zápisu prstové abecedy . . . . .	4
2.2	Příklad zápisu ZJ . . . . .	4
2.3	Příklady zápisů ZJ v několika notačních systémech . . . . .	5
2.4	Exoskelet . . . . .	6
2.5	Elektromagnetický MoCap systém . . . . .	7
2.6	Optický MoCap systém . . . . .	8
2.7	Kosterní model . . . . .	10
2.8	Graf stromové struktury částí těla . . . . .	10
2.9	Grafická ilustrace dat . . . . .	11
2.10	Grafické znázornění lineárního a nelineárního porovnání signálů . . . . .	12
2.11	Grafické znázornění algoritmu DTW . . . . .	15
2.12	Vizuální vykreslení výsledku DTW . . . . .	16
2.13	Grafické znázornění výsledku DTW . . . . .	16
2.14	Ztrátová funkce softDTW při $\gamma=1$ a $\gamma=0.01$ . . . . .	17
2.15	Příklad Fourierovy transformace . . . . .	19
2.16	Příklad interpolace nahrazením funkční hodnotou $f(x_1)$ . . . . .	20
2.17	Příklad interpolace nahrazením funkční hodnotou průsečíku . . . . .	20
2.18	Příklad lineární interpolace . . . . .	21
2.19	Detail rozdílu mezi lineární a kubickou interpolací . . . . .	22
2.20	Grafické znázornění Eukleidovské a Hammingovy metriky . . . . .	24
3.1	Ilustrační příklad reprezentace úspěšnosti algoritmu histogramem . . . . .	28
3.2	Vybrané klouby v programu Blender . . . . .	29
3.3	Výsledky DTW (úplná kostra) . . . . .	30
3.4	Výsledky DTW (omezení kostry na paže a prsty) . . . . .	30
3.5	Výsledky DTW (omezení kostry pouze na paže) . . . . .	31
3.6	Úspěšnost downsamplingu a upsamplingu . . . . .	33
3.7	Výsledky řádů interpolace . . . . .	34
3.8	Fourierova transformace - downsampling a upsampling . . . . .	35
3.9	Výsledky algoritmu DTW . . . . .	36
3.10	Graf pořadí metrik při lineární interpolaci . . . . .	38
3.11	Graf pořadí metrik při Fourierově transformaci . . . . .	39
3.12	Trajektorie s posunem lokace znakování při znakování slova „zítra“ . . . . .	41
3.13	Trajektorie s posunem lokace znakování při znakování slova „teplo“ . . . . .	42
3.14	Testovací data . . . . .	43
3.15	Výsledky algoritmu SoftDTW . . . . .	45
3.16	Porovnání typů algoritmu na analýze slova „zítra“ . . . . .	48
3.17	Porovnání typů algoritmu na analýze slova „teplo“ . . . . .	49

## Seznam tabulek

1	Tabulka délek unikátních výskytů znaků . . . . .	12
2	Tabulka ohodnocení řádů interpolací . . . . .	34
3	Výsledky metrik v kombinaci s lineární interpolací . . . . .	37
4	Výsledky metrik v kombinaci s Fourierovou transformací . . . . .	38
5	Změna úspěšnosti při použití Fourierovy transformace . . . . .	39
6	Výsledky Pearsonova korelačního koeficientu . . . . .	40
7	Změna Pearsonova kor. koef. a Eukleidovské metriky při posunu dat . . . .	44
8	Výsledky algoritmů DTW a SoftDTW . . . . .	45
9	Celkové srovnání nejlepších typů testovacího algoritmu . . . . .	46
10	Celkové srovnání výpočetní doby všech testovaných druhů algoritmu . . . .	50

## Reference

- [1] T. Hanke, “Hamnosys-representing sign language data in language resources and language processing contexts,” in *LREC*, vol. 4, 2004, pp. 1–6.
- [2] J. Stokoe, William C., “Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf,” *The Journal of Deaf Studies and Deaf Education*, vol. 10, no. 1, pp. 3–37, 01 2005. [Online]. Available: <https://doi.org/10.1093/deafed/eni001>
- [3] J. Martin, “A linguistic comparison: Two notation systems for signed languages: Stokoe notation and sutton signwriting,” *Unpublished manuscript, Western Washington University*, p. 22, 2000.
- [4] P. Jedlička, Z. Krňoul, J. Kanis, and M. Železný, “Sign language motion capture dataset for data-driven synthesis,” in *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, E. Efthimiou, S.-E. Fotinea, T. Hanke, J. A. Hochgesang, J. Kristoffersen, and J. Mesch, Eds. Marseille, France: European Language Resources Association (ELRA), 2020, pp. 101–106. [Online]. Available: <https://www.sign-lang.uni-hamburg.de/lrec/lrec/pubs/20027.pdf>
- [5] J. Berke, “Self-identification in the deaf community,” 2010. [Online]. Available: <https://www.verywellhealth.com/deaf-culture-big-d-small-d-1046233>
- [6] K. Redlich, “Co je znakový jazyk, znakovaná čeština..” 2004. [Online]. Available: <http://ruce.cz/clanky/188-co-je-znakovy-jazyk-znakovana-cestina>
- [7] J. Fikejs, “Slavné osobnosti v historii neslyšících,” Nov 2005. [Online]. Available: <http://ruce.cz/clanky/16-slavne-osobnosti-v-historii-neslysicich>
- [8] L. Půlpánová, “Historie neslyšících,” Nov 2006. [Online]. Available: <http://ruce.cz/clanky/112-historie-neslysicich>
- [9] ČESKO, “Zákon ze dne 11. června 1998 o komunikačních systémech neslyšících a hluchoslepých osob,” Jun 1998.
- [10] A. Macurová and R. Zbořilová, *Jazyky v komunikaci neslyšících*. Charles University in Prague, Karolinum Press, 2019.
- [11] A. Macurová, “Jazyk a hluchota,” 2011. [Online]. Available: <http://sas.ujc.cas.cz/archiv.php?art=4006>
- [12] A. Macurová, “Proč a jak zapisovat znaky českého znakového jazyka,” Jun 2005. [Online]. Available: <http://ruce.cz/clanky/15-proc-a-jak-zapisovat-znaky-ceskeho-znakoveho-jazyka>

- [13] M. Matura, “3d elektromagnetický artikulograf,” *Studentská vědecká konference*, 2016.
- [14] B. Mursyid, “Motion capture ii,” Jan 2015. [Online]. Available: <https://baswaramursyid.wordpress.com/2015/01/20/motion-capture-ii/>
- [15] P. Stolař, “Snímání pohybu pomocí systému kinect,” 2013.
- [16] M. Vinkler, “Snímání a rekonstrukce pohybu postavy,” Ph.D. dissertation, Masarykova univerzita, Fakulta informatiky, 2009.
- [17] E. Tsiporkova, “Dynamic time warping algorithm for gene expression time series,” *Power point presentation. Retrieved September*, vol. 11, 2010.
- [18] M. N. Murty and V. S. Devi, *Introduction to pattern recognition and machine learning*. World Scientific, 2015, vol. 5.
- [19] M. Müller, “Dynamic time warping,” *Information retrieval for music and motion*, pp. 69–84, 2007.
- [20] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.
- [21] M. Cuturi and M. Blondel, “Soft-DTW: a differentiable loss function for time-series,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 894–903. [Online]. Available: <http://proceedings.mlr.press/v70/cuturi17a.html>
- [22] A. Klíč, K. Volka, and M. Dubcová, *Fourierova transformace s příklady z infračervené spektroskopie*. Vydavatelství VŠCHT v Praze, 2002.
- [23] V. Homola. [Online]. Available: <https://homel.vsb.cz/~hom50/SLBSTATE/IER/ESTIEXR.HTM>
- [24] C. Caruso and F. Quarta, “Interpolation methods comparison,” *Computers & Mathematics with Applications*, vol. 35, no. 12, pp. 109–126, 1998. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0898122198001011>
- [25] K. M. e. a. k. HOLČÍK Jiří, “Metriky pro určení vzdálenosti mezi dvěma vektory s kvalitativními hodnotami souřadnic,” *Matematická biologie: e-learningová učebnice*, 2015. [Online]. Available: <https://bit.ly/3foHHRF>
- [26] G. M. Wojcik and M. Ważny, “Bray-curtis metrics as measure of liquid state machine separation ability in function of connections density,” *Procedia Computer Science*, vol. 51, pp. 2979–2983, 2015, international Conference On Computational Science, ICCS 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050915018529>

- [27] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>
- [28] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [29] Wes McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference*, Stéfán van der Walt and Jarrod Millman, Eds., 2010, pp. 56 – 61.
- [30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [31] C. F. Jekel, G. Venter, M. P. Venter, N. Stander, and R. T. Haftka, “Similarity measures for identifying material parameters from hysteresis loops using inverse analysis,” *International Journal of Material Forming*, may 2019. [Online]. Available: <https://doi.org/10.1007/s12289-018-1421-8>
- [32] W. Meert, K. Hendrickx, and T. V. Craenendonck, “wannesm/dtaidistance v2.0.0,” Aug. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3981067>
- [33] T. Giorgino, “Computing and visualizing dynamic time warping alignments in r: The dtw package,” *Journal of Statistical Software*, vol. 24, 8 2009.
- [34] M. ANDRLOVÁ, “Znakový jazyk v obraze [online],” 2016 [cit. 2021-04-24]. [Online]. Available: AvailablefromWWW(<https://is.muni.cz/th/cpkwm/>)