



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A  
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**METODY ZPRACOVÁNÍ VELKÝCH DAT Z EXPERIMENTŮ  
A SIMULACÍ DYNAMICKÝCH SOUSTAV**

PROCESSING OF BIG DATA FROM MEASUREMENTS AND SIMULATIONS OF DYNAMIC SYSTEMS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Anton Ďuriš**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Filip Kšica**

**BRNO 2019**

# Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky  
Student: **Anton Ďuriš**  
Studijní program: Aplikované vědy v inženýrství  
Studijní obor: Mechatronika  
Vedoucí práce: **Ing. Filip Kšica**  
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## **Metody zpracování velkých dat z experimentů a simulací dynamických soustav**

### **Stručná charakteristika problematiky úkolu:**

Výstupem měření a simulací prováděných na moderních technických zařízeních je zpravidla velké množství dat, jejichž zpracování skýtá značný problém z hlediska potřebného času a výpočetního výkonu. Cílem této práce je shrnout a demonstrovat dostupné metody zpracování souborů velkých dat z měření či simulací a obecné přístupy pro jejich analýzu z hlediska požadovaných vlastností a projevů dynamických soustav.

### **Cíle bakalářské práce:**

1. Rešerše metod pro zpracování Velkých dat v dostupných softwarových nástrojích.
2. Aplikace zvoleného přístupu pro zpracování konkrétních Velkých dat z experimentů a simulací technických soustav s cílem určit jejich dynamické vlastnosti.
3. Zhodnocení jednotlivých přístupů pro práci s Velkými daty vzhledem k potřebám praxe.

### **Seznam doporučené literatury:**

WILLINGHAM, D, 2015. BIG DATA ANALYSIS & ANALYTICS WITH MATLAB ® Pre-Press Release 23-Oct-2015 11 : 00. 9–12.

HATCH, Michael R., 2001. Vibration Simulation Using MATLAB and ANSYS [online]. ISBN 1584882050.

PIERSOL, Allan G. a Thomas L. PAEZ, 2010. Harris' Shock and Vibration Handbook [online]. 6. B.m.: McGraw Hill.

TAO, Fei, Jiangfeng CHENG, Qinglin QI, Meng ZHANG, He ZHANG a Fangyuan SUI, 2018. Digital twin-driven product design, manufacturing and service with big data. International Journal of Advanced Manufacturing Technology [online]. B.m.: The International Journal of Advanced Manufacturing Technology, 94(9–12), 3563–3576.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## Abstrakt

Táto práca sa zaoberá spracovaním Veľkých dát z moderných technických zariadení. Spracovanie Veľkých dát je časovo náročný a zložitý proces, vyžadujúci výkonný hardvér, avšak existujú metódy, vďaka ktorým je spracovanie možné aj na osobných počítačoch. V teoretickej časti práce sú zhrnuté dostupné metódy pre prácu s Veľkými dátami, a funkcie, ktoré sú integrované v softvéroch a programovacích jazykoch pre prácu s nimi. Zameriame sa na súborový formát TDMS, jeho vlastnosti a možnosti otvorenia v softvéroch a programovacích jazykoch a porovnáme ich. Ďalej sú uvedené metódy na preprocessing dát, ktoré urýchľujú ich spracovanie a analýzu. V praktickej časti sme vytvorili grafické rozhranie v prostredí Matlab pre prácu so súborovým formátom TDMS, v ktorom sú implementované metódy preprocessingu. Cieľom tejto práce je zefektívniť prácu s týmto súborovým formátom za použitia osobného počítača.

## Abstract

This thesis is focused on Big data processing from modern technical devices. Big data processing is a time-consuming and complex process requiring powerful hardware, however, there are some methods which enable Big data processing even on a personal computer. In the theoretical part of this thesis, available methods and functions for Big data processing, which are integrated within software and programming languages, are summarized. We focus on TDMS file format, its features and opening options in software and programming languages and their comparison. Then, data preprocessing methods accelerating processing and analysis are listed. In the practical part of this thesis, a graphical user interface in Matlab environment for TDMS file format processing with implemented data preprocessing methods is created. The aim of this thesis is to make processing with this file format more efficient by using a personal computer.

## Klíčové slová

Veľké dáta, digitálne spracovanie dát, súborový formát TDMS, grafické používateľské rozhranie, Fourierova transformácia, Matlab, Python

## Keywords

Big data, digital signal processing, TDMS file format, graphical user interface, Fourier transform, Matlab, Python

ŘURÍŠ, A. *Metody zpracování velkých dat z experimentů a simulací dynamických soustav*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2019. 41 s. Vedoucí Ing. Filip Kšica.

Prehlasujem, že táto práca je mojim dielom, ktorú som spracoval samostatne pod vedením  
Ing. Filipa Kšicu s použitím literatúry uvedenej v zdrojoch literatúry na konci práce.

Anton Ďuriš

Ďakujem vedúcemu bakalárskej práce pánovi Ing. Filipovi Kšicovi za poskytnutie odbornej pomoci, skúseností a cenných rád pri tvorbe tejto práce. Ďakujem tiež mojej rodine za podporu počas štúdia.

Anton Ďuriš

# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>  | <b>3</b>  |
| <b>2</b> | <b>Veľké dáta</b>  | <b>4</b>  |
| 2.1      | Definícia pojmu Veľkých dát . . . . .                                  | 4         |
| 2.2      | Charakteristické vlastnosti Veľkých dát . . . . .                      | 5         |
| <b>3</b> | <b>Zdroje Veľkých dát v priemysle</b>                                  | <b>6</b>  |
| 3.1      | Delenie zdrojov dát . . . . .  | 6         |
| 3.2      | Najväčšie zdroje dát . . . . .   | 7         |
| 3.2.1    | Dáta pre virtuálne dvojčá . . . . .                                    | 7         |
| 3.2.2    | Dáta zo senzorov . . . . .   | 8         |
| <b>4</b> | <b>Metódy pre spracovanie Veľkých dát pomocou počítačového clustra</b> | <b>10</b> |
| 4.1      | Hadoop . . . . .   | 10        |
| 4.2      | Apache Spark . . . . .   | 10        |
| <b>5</b> | <b>Spracovanie Veľkých dát<br/>v programovacích jazykoch</b>           | <b>11</b> |
| 5.1      | Matlab . . . . .   | 11        |
| 5.1.1    | Datastore . . . . .  | 11        |
| 5.1.2    | MapReduce . . . . .  | 11        |
| 5.2      | Python – PySpark . . . . .   | 12        |
| <b>6</b> | <b>Súborový formát TDMS</b>  | <b>13</b> |
| 6.1      | Štruktúra formátu TDMS . . . . .                                       | 13        |
| 6.2      | Otvorenie formátu TDMS . . . . .                                       | 14        |
| 6.2.1    | Microsoft Excel a OpenOffice . . . . .                                 | 14        |
| 6.2.2    | Matlab . . . . .   | 14        |
| 6.2.3    | Python . . . . .   | 14        |
| 6.3      | Porovnanie otvorenia súborov v jazyku Matlab<br>a Python . . . . .     | 15        |
| 6.4      | Obmedzenia softvérov . . . . .   | 16        |
| 6.4.1    | Veľkosť TDMS súboru pre rôzne časy mernia . . . . .                    | 16        |
| <b>7</b> | <b>Preprocessing dát</b>   | <b>17</b> |
| 7.1      | Vykreslenie dát . . . . .  | 17        |
| 7.2      | Zmenšenie dát . . . . .  | 19        |
| 7.3      | Filtrovanie dát . . . . .  | 19        |
| 7.4      | Transformácia dát . . . . .  | 20        |
| 7.4.1    | Zero padding . . . . .   | 21        |
| 7.4.2    | Windowing . . . . .  | 23        |
| 7.5      | Konverzia dát . . . . .  | 24        |

|           |  |           |
|-----------|--|-----------|
| <b>8</b>  | <b>Grafické rozhranie pre preprocessing dát</b>      | <b>25</b> |
| 8.1       | Popis oblastí grafického rozhrania . . . . .         | 26        |
| 8.1.1     | Hlavné menu . . . . .                                | 27        |
| 8.1.2     | Menu s funkciami pre vybratý kanál . . . . .         | 28        |
| 8.1.3     | Výber dát . . . . .                                  | 29        |
| 8.1.4     | Funkcie a export viacerých meraní . . . . .          | 30        |
| <b>9</b>  | <b>Analýza experimentálnych dát</b>                  | <b>31</b> |
| 9.1       | Overenie budiacej frekvencie . . . . .               | 32        |
| 9.2       | Výpočet tlmenia . . . . .                            | 33        |
| <b>10</b> | <b>Možnosti budúceho vývoja grafického rozhrania</b> | <b>36</b> |
| <b>11</b> | <b>Záver</b>   | <b>37</b> |
| <b>12</b> | <b>Prílohy</b>                                       | <b>41</b> |



# 1. Úvod

V dnešnej dobe, plnej moderných technických zariadení a zautomatizovaných procesov, ktoré je nutné monitorovať, nastáva problém so spracovaním dát, ktoré pri týchto procesoch vznikajú. Vďaka stále zrýchľujúcim sa technickým pokrokom klesá cena a veľkosť senzorov, ktoré slúžia na ich monitorovanie. S počtom týchto senzorov narastá aj veľkosť dát, ktoré je nutné spracovať a analyzovať. Tieto dáta môžeme nazvať Veľkými dátami. Tento pojem je nutné správne definovať a predstaviť si, pretože v prípade Veľkých dát nejde primárne o ich veľkosť, ale o výpočetnú náročnosť na ich spracovanie. Hranica, kedy tieto dáta môžeme nazvať Veľkými dátami, nie je určená.

Problém so spracovaním sa často rieši pomocou počítačových clustrov<sup>1</sup>, pretože spracovanie na osobnom počítači by zabralo dlhší čas, a výsledky tohoto spracovania by už nemuseli byť aktuálne. Metódy, ktoré sa používajú na paralelizáciu na clustroch, sú v dnešnej dobe už pomerne dobre odladené a integrované v programovacích jazykoch. V tejto práci sa zameriame na metódy Hadoop a Apache Spark. Jeden z hlavných problémov pri týchto metódach je samotný prístup ku clustrom. Takéto počítače sú drahé, a ak aj máme ku nim prístup, musíme transportovať väčšie množstvo dát na tento cluster.

Cieľom tejto práce je zefektívnenie spracovania Veľkých dát bez prístupu k počítačovému clustru. Pre naplnenie tohoto cieľa bude vytvorený softvér pre ľahšiu prácu s dátami. Veľké dáta, s ktorými pracujeme, pochádzajú zo senzorov, ktorých výstup sa ukladá pomocou meracej karty a softvéru Signal Express, od spoločnosti National Instruments(NI), do súborového formátu TDMS<sup>2</sup>. S týmto komplexným formátom je možné pracovať so softvérom od spoločnosti NI, ale pre väčšiu slobodu pri spracovaní týchto dát vznikli knižnice, kde je možné s formátom pracovať aj v iných prostrediach a programovacích jazykoch. My sme si vybrali programovacie prostredie Matlab, kde na načítaných dátach môžeme jednoducho vykonávať preprocessing dát a ich analýzu. Vytvorený program súbory tohoto typu načíta, spracuje a uloží do pracovného priestoru softvéru Matlab.

Prvá časť práce sa zaoberá všeobecným popisom Veľkých dát. Popíšeme zdroje výskytu Veľkých dát v priemysle a dostupné metódy pre ich spracovanie na počítačovom clustri. V závere tejto časti popíšeme integrovanie týchto metód v programovacích jazykoch Matlab a Python.

V druhej časti sa zameriame na súborový formát TDMS, s ktorým pracujeme v praktickej časti tejto práce. Tento súborový formát slúži k prehľadnému uloženiu veľkého množstva dát zo senzorov. Popíšeme možnosti otvorenia vo viacerých softvéroch s ich obmedzeniami a porovnáme dobu spracovania Veľkých dát medzi softvérmi. Ďalej si popíšeme základné oblasti preprocessingu, ako napríklad zmenšenie, filtrovanie, transformáciu do frekvenčnej oblasti a konverziu do iného formátu.

Posledná časť je zameraná na implementáciu poznatkov z prvých dvoch častí. Za účelom zjednodušenia práce s väčším počtom meraní, vytvoríme vyššie spomenuté grafické prostredie, ktoré tieto dáta roztriedi, umožní na nich vykonať preprocessing a uloží do pracovného priestoru. Pre overenie funkčnosti tohoto rozhrania spracujeme dáta z merania na nosníku a budeme určovať ich dynamické vlastnosti.

---

<sup>1</sup>zoskupenie počítačov pracujúcich navzájom, ktoré sa javia ako jeden počítač

<sup>2</sup>Technical Data Management Streaming

## 2. Veľké dáta

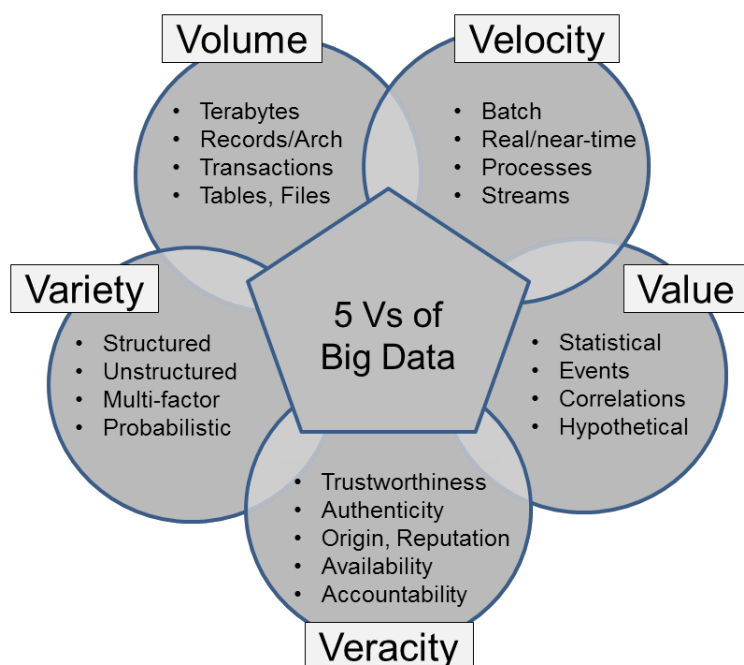
### 2.1. Definícia pojmu Veľkých dát

Ako prví pojem Veľké dáta použili Michael Cox a David Ellsworth, odkazujúc na používanie väčších objemov dát pre vizualizáciu. Existuje viac definícií na popísanie tohoto pojmu, ale najznámejšia pochádza od spoločnosti IBM, ktorá hovorí, že Veľké dáta by mali byť charakterizované aspoň jedným, alebo všetkými z troch „V“: <sup>[14]</sup>

- *volume* (objem) – odkazuje na väčší objem dát získaných zo zdrojov každý moment
- *velocity* (rýchlosť) – rýchlosť, akou sa dáta generujú alebo analyzujú
- *variety* (rôznosť) – popisuje používanie rôznych dátových typov a zdrojov, z ktorých sú dáta získavané

Ostatné definície sa líšia od tej vyššie uvedenej len pridaním ďalších vlastností, presnejšie pridaním jedného alebo viacerých „V“, ktoré sú charakteristické pre Veľké dáta. Na obrázku 2.1 sú zobrazené tieto „V“ spolu s predchádzajúcimi. Sú nimi napríklad: <sup>[3]</sup>

- *veracity* (pravdivosť) – vyjadruje, koľko šumu sa v dátach nachádza, ktorý ich znehodnocuje
- *value* (cena) – cena nákladov pre analýzu a úschovu dát, ale tiež aj cena, za ktorú môžu byť dáta poskytované iným spoločnostiam



Obr. 2.1: Diagram vlastností Veľkých dát <sup>[3]</sup>

## 2.2. Charakteristické vlastnosti Velkých dát

Velké dáta označujú obrovský objem dát, ktorý je veľmi náročný na spracovanie použitím tradičných databázových a softvérových techník, a potrebujú oveľa viac času na ich analýzu. Pod tradičnými technikami si môžeme predstaviť spracovanie dát na osobnom počítači, napríklad spracovanie tabuľky v softvéri Microsoft Office. Velké dáta si na rozdiel od tradičných techník požadujú iný prístup, ktorý aj pre ich zložitosť umožňuje systematické spracovanie. Tieto dáta väčšinou pochádzajú z monitorovania technických zariadení, pokiaľ neberieme ohľad na webové dáta, ktoré sú viacnásobne väčšie. Pojem Velké dáta označuje niekoľko vlastností v dátových súboroch a technológii na ich spracovanie. Z týchto vlastností by mali Velké dáta obsahovať tieto dve: <sup>[1]</sup>

- Sú priveľké na to, aby ich spracovanie bolo praktické na osobnom počítači. Namiesto toho používame pre spracovanie počítačový cluster. Dôraz kladieme na to, aby bol problém škálovateľný, a bol distribuovaný v clustri ľubovoľnej veľkosti s rôznymi časťami analýzy, ktoré bežia paralelne. Nie je však nikde uvedené, kedy sú Big data naozaj „veľké“.
- Tieto dáta sú často neštruktúrované. Toto označenie neznamena, že tieto dáta nemajú žiadnu štruktúru, ale nie je ich možné uložiť do tradičných databáz, ako napríklad SQL<sup>1</sup>. Neštruktúrované dáta by mohli byť napríklad obrázky, PDF<sup>2</sup> dokumenty, HTML<sup>3</sup> dokumenty, Excel dokumenty, neusporiadané do jednotlivých riadkov a stĺpcov a denníkové súbory. Tradičné databázy a pevná štruktúra dát, ktoré obsahujú, zaisťuje veľmi optimalizovaný výkon pri ich spracovaní. V oblasti Big data ale pracujeme prevažne s dátami rôznych formátov, a preto musíme byť schopní tieto dáta spracovať aj menej konvenčnými spôsobmi.

V túto chvíľu je technológia pre spracovanie Velkých dát zatiaľ nová a musí sa vysporiadať s neštruktúrovanými dátami, čo má za následok, že niekedy je pomalšia ako donedávna používané technológie, je ťažšie nastaviteľná a zaberie viac času. Preto je najlepšie používať nástroje pre Velké dáta spolu s tradičnými technikami. Napríklad, na počítačovom clustri získame medzivýsledky pomocou nástrojov pre Velké dáta, ktoré potom spracujeme pomocou tradičných techník na počítači. Tieto medzivýsledky zaberú menej pamäte ako vstupné dáta, a preto je možné spracovanie na bežnom počítači. <sup>[1]</sup>

---

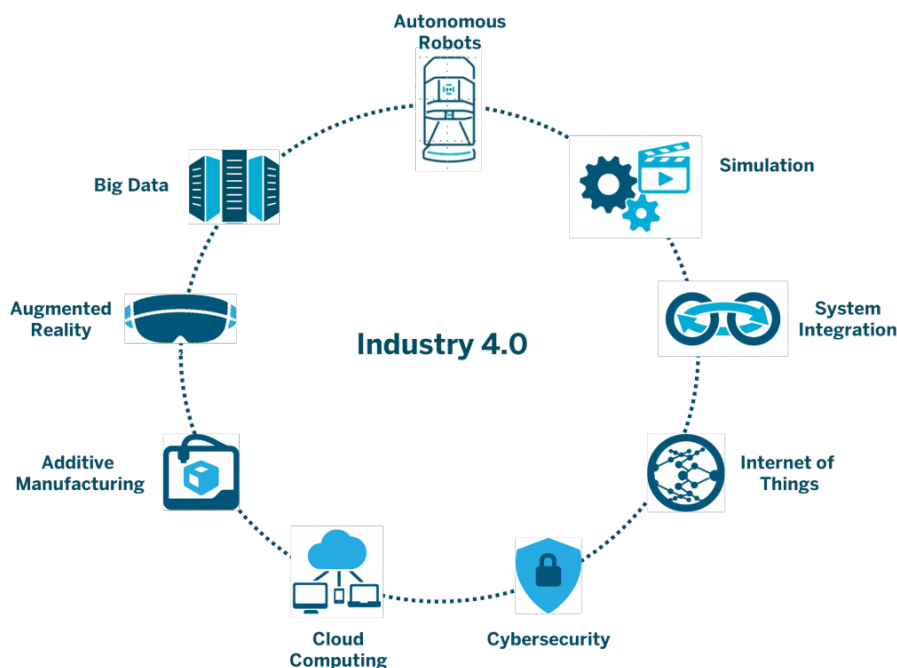
<sup>1</sup>Structured Query Language – databázový súborový formát

<sup>2</sup>Portable Document Format – súborový formát pre ukladanie dokumentov nezávislý na softvéri

<sup>3</sup>Hypertext Markup Language – značkovací jazyk určený na vytváranie webových stránok

### 3. Zdroje Veľkých dát v priemysle

S rýchlym rozvojom technológií, nových možností internetu, IoT<sup>1</sup>, Veľkých dát, počítačových clustrov, umelej inteligencie a výrobných technológií, sa začala aj štvrtá priemyselná revolúcia, tiež známa pod pojmom Industry 4.0, ktorej hlavným trendom je automatizovanie a výmena dát pri výrobných procesoch. Časti, ktoré Industry 4.0 obsahuje, sú zobrazené na obrázku 3.1. Tieto prenesené dáta, ktoré sú veľmi dôležité pre monitorovanie procesov, sú dostatočne veľké nato, aby sme ich mohli nazývať Veľkými dátami, a pochádzajú väčšinou zo senzorov na monitorovanom objekte. Okrem dát zo senzorov sa generujú dáta aj z rôznych prístrojov, logistických vozidiel, ľudí, výrobných hál a dáta zo sledovania výrobných procesov, ako napríklad efektívnosť pri výrobe, environmentálne znečistenie, energetická spotreba a iné.



Obr. 3.1: Časti, ktoré sú charakteristické pre Industry 4.0 [11]

#### 3.1. Delenie zdrojov dát

Zdroje dát môžeme rozdeliť do týchto podskupín: [17]

1. Rôzne prístrojové dáta – prepojenie podnikových systémov a prístrojov za pomoci *Cyber-Physical System* (CPS). CPS umožní pripojenie senzorov, ovládačov, videokamier a RFID<sup>2</sup> čítačiek do internetovej siete, čo umožní ich vzájomnú komunikáciu. Metódy pre spracovanie a analýzu Veľkých dát spracujú tieto dáta z prístrojov, buď lokálne, na týchto prístrojoch, alebo na serveri. Výsledky z analýzy týchto dát pomáhajú ľuďom a prístrojom pri optimalizácii výroby a výbere vhodného rozhodnutia.

<sup>1</sup>Internet of Things – sieť fyzických zariadení a každodenných objektov vybavených elektronikou, ktorá umožňuje ich prepojenie a výmenu dát

<sup>2</sup>Radio-frequency identification – identifikačný prvok, pracujúci vo vysokofrekvenčnom pásme

2. **Dáta životného cyklu produktu** – ide o proces od prijatia objednávky po vývoz finálneho produktu. Zahŕňa to dizajn, výrobu, testovanie, predaj a údržbu produktu. Pri všetkých procesoch sú zbierané, prenášané a spracované rôzne dáta, ktoré sú potrebné pre zostavenie životného cyklu produktu a následne jeho správu pre vylepšovanie výrobných procesov a finálneho produktu. Po zozbieraní dát z výrobných procesov sa začnú zbierať dáta z vložených senzorov na výrobku pri testovaní a používaní, ktoré sú použité na optimalizáciu výrobného procesu. Ďalšie obrovské množstvá dát sa produkujú pri vzájomnej interakcii zákazníka, spoločnosti a trhom. Ich spracovanie dáva možnosť spotrebiteľom podieľať sa na analýze po dopyte produktu, jeho dizajne a podobne.
3. **Dáta o prevádzke podniku** – obsahujú informácie o organizačnej štruktúre, riadení podniku, výrobe, zariadeniach, predaji, kontrole kvality, plánoch, elektronickom obchode a iné dáta. Tieto dáta a analýza dodávateľského reťazca vedú k efektívnosti a zníženiu nákladov a jeho redukovaniu a optimalizovaniu. Pri použití systému energetického spravovania sa môže optimalizovať spotreba energie pri výrobnom procese v reálnom čase. Taktiež pri použití dát o predaji a dodávateľských dát sa môže dynamicky upravovať optimálna produkcia a zásoby.
4. **Výrobný hodnotový reťazec** – obsahuje dáta od zákazníkov, partnerov, dodávateľov a iné. V každom odvetví priemyslu je konkurencia, a byť najlepším znamená pochopiť technologický vývoj, výrobu, predaj, logistiku a všetky ostatné súťažné faktory. Analýzou dát z tohoto reťazca môžu podniky nadobudnúť nový uhol pohľadu na tento reťazec a strategicky ho využiť.
5. **Dáta z externej spolupráce** – ide o dáta z oblastí ekonómie, priemyslu, obchodu a podobne. Analýza Veľkých dát v ekonómii a pri výskume trhu je kľúčovým prostriedkom pre rozhodovanie vo vedení podniku. V prípade nejakej externej zmeny, ktorá by mohla mať zlý dopad na podnik, môže tieto dáta využiť a tak predísť následkom, ktoré by mohli mať negatívny vplyv na funkčnosť podniku. Zamestnanci, od vedenia až po robotníkov, začali od podnikov dostávať informácie, zručnosti a nástroje, ktoré môžu využívať pre čo najlepšie a včasné rozhodnutia.

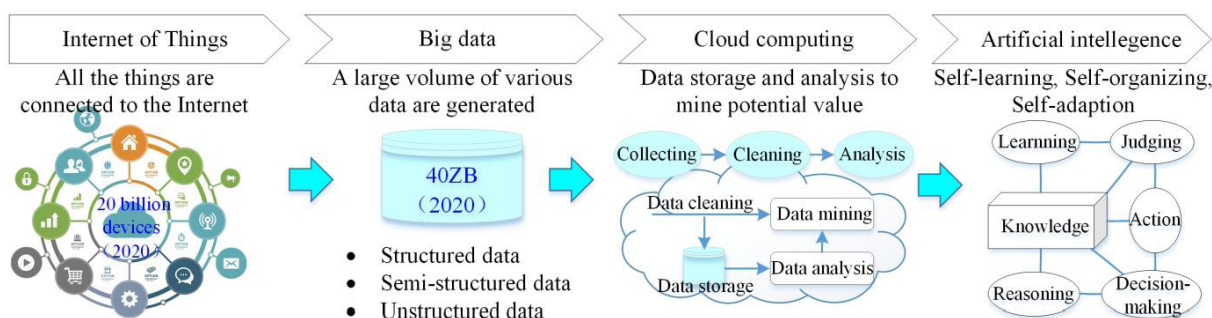
## 3.2. Najväčšie zdroje dát

Medzi najväčšie zdroje z predchádzajúceho delenia patria prvé dva, a to rôzne prístrojové dáta a dáta životného cyklu produktu. Ďalším významným zdrojom sú dáta, ktoré vznikajú pri monitorovaní prístrojov pomocou rôznych senzorov. Tieto dáta sa zaraďujú medzi rôzne prístrojové dáta z predchádzajúceho delenia.

### 3.2.1. Dáta pre virtuálne dvojča

Oba najväčšie zdroje využíva metóda virtuálneho dvojčata. Je novým trendom v priemysle a tvorí veľkú časť Veľkých dát v priemysle. Metóda virtuálneho dvojčata spočíva vo vzájomnej komunikácii virtuálneho a fyzického modelu. Dáta pre virtuálne dvojča získavame z rôznych IoT senzorov, historických dát procesu, alebo dát získaných pomocou umelej inteligencie a zbierame ich počas rôznych štádií životných cyklov produktu, aby

sme mohli simulovať fyzický proces. Pri tejto komunikácii vznikajú obrovské objemy dát, ktoré slúžia na zefektívnenie výroby. Tieto dáta sa väčšinou analyzujú na počítačových clustroch, pričom sa hľadajú chyby vo výrobnom procese, ich dôsledok na chod procesu a prípadné riešenia pre odstránenie tejto chyby. Ich hlavnou úlohou je optimalizácia výrobného procesu s cieľom minimalizovať personálne, časové a predovšetkým finančné investície do výroby pri súčasnej maximalizácii jej ziskovosti. Tento proces je zobrazený na obrázku 3.2. Virtuálne modely by mali reagovať na dáta zo senzorov, a tak predpovedať a analyzovať dynamické zmeny, zatiaľ čo fyzické objekty budú reagovať na zmeny podľa optimalizovaného modelu zo simulácie. <sup>[15]</sup>



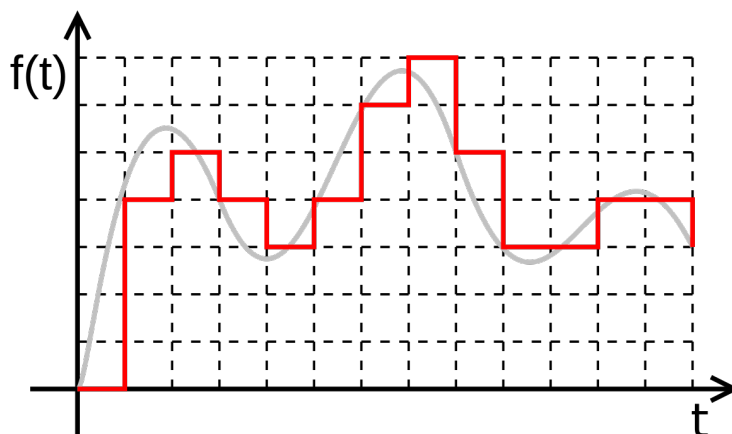
Obr. 3.2: Spracovanie veľkých dát <sup>[15]</sup>

### 3.2.2. Dáta zo senzorov

Pri automatizovaní rôznych procesov je nutné, aby boli stále monitorované a pri prípadnom zlyhaní nedošlo k škode. Tento problém je riešený pomocou senzorov umiestnených všade, kde by mohol nastať problém. Veľkosť dát z týchto senzorov ovplyvňujú najmä tieto faktory:

- Počet a typ senzorov – v dnešnej dobe používame veľké množstvo senzorov, ktoré monitorujú každý vykonávaný dej. Takýmto príkladom môžu byť napríklad veľké výrobné linky. Čím viac senzorov použijeme, tým sa zväčšuje veľkosť dát. Medzi typy senzorov môžeme spomenúť tie, ktoré sa líšia počtom výstupov. Sensory s jedným výstupom môžu byť napríklad na meranie teploty. Sensory s dvoma a viacerými výstupmi môžu zaznamenávať napríklad zrýchlenie v rôznych osách. Počet výstupov teda zvyšuje veľkosť dát.
- Doba monitorovania – niektoré procesy je nutné monitorovať kontinuálne, pričom vzniká obrovské množstvo dát, ktoré musia byť spracované po častiach. Takýmto príkladom môže byť monitorovanie chodu výrobnéj linky s nepretržitou prevádzkou. Pri procesoch, ktoré sa vykonávajú sekvenčne určitý čas, stačí monitorovanie len v určitých udalostiach, čím sa výrazne zníži veľkosť dát.
- Rozlíšenie senzorov – pri výbere senzorov pre dané monitorovanie zohľadňujeme, s akým minimálnym rozlíšením je potrebné tieto výsledky spracovať. Týmto obmedzíme, aby namerané dáta nemali zbytočne veľké rozlíšenie, ktoré im nepridávajú hodnotu, a zmenšia veľkosť nameraných dát. Pri analógových senzoroch namerané dáta prevádzame do digitálnych pomocou analógovo-digitálneho prevodníka, ktorý

môže mať rôzne rozlišovacie vlastnosti. Výstupom z prevodníka je binárny kód, ktorý môže mať rôznu dĺžku, napríklad 8,16-bitovú a podobne. Dĺžka tohoto kódu teda závisí na rozlíšení prevodníka. Výberom vhodného prevodníka pre dané monitorovanie sa namerané dáta orežú podľa veľkosti binárneho kódu, čím tiež zmenšíme veľkosť pôvodných dát zo senzora. Takýmto spôsobom vznikne nespojitý signál, zobrazený na obrázku 3.3. Pre jasnejšiu vizualizáciu sa body často prekládajú krivkou. Ak meráme dáta, ktoré sú určené pre výskum, musia byť čo najpresnejšie. Pri takýchto meraniach používame senzory s čo najväčším rozlíšením.



Obr. 3.3: Vzorkovanie digitálneho signálu <sup>[18]</sup>

- Vzorkovacia frekvencia – vyjadruje počet meraní, ktoré je senzor schopný vykonať za jednotku času. To je dané napríklad typom meranej veličiny, kde veličiny, pri ktorých predpokladáme dynamickú zmenu ich hodnôt, meriame s vyššou vzorkovacou frekvenciou. Pod dynamickou zmenou si môžeme predstaviť napríklad odozvu na impulz. Táto frekvencia býva v tisíckach vzorkov za sekundu, a tak stačí pár sekúnd merania a vzniknuté súbory budú veľmi veľké. Naopak pre veličiny, pri ktorých nepredpokladáme dynamickú zmenu, stačí merať s malou vzorkovacou frekvenciou. Tá môže byť v desiatkách vzorkov za minútu. Takýto dej môže byť napríklad otepľovanie stroja.

## 4. Metódy pre spracovanie Veľkých dát pomocou počítačového clustra

V tejto časti si popíšeme dve základné metódy pre spracovanie Veľkých dát pomocou počítačového clustra. Hlavnou výhodou tejto metódy je rýchlejší výpočetný čas pri extrémne veľkých dátach, ktorý by na osobnom počítači trval dlho, a ich výsledky by už nemuseli byť aktuálne. Opisované metódy sú už veľmi dobre integrované v programovacích jazykoch, a ich používanie nie je náročné. Nevýhodou môže byť len fakt, že prístup ku výkonným počítačovým clustrom nie je jednoduchý, a pokiaľ aj máme prístup k takémuto počítaču, je to väčšinou vzdialený prístup, pri ktorom nastáva problém s transportom obrovských dát k nemu.

### 4.1. Hadoop

Hneď po tom, ako Google publikoval informácie o metóde *MapReduce*, programovacieho modelu pre spracovanie veľkého množstva dát v paralelnom a distribuovanom algoritme na počítačovom clustri, začala moderná éra pre Veľké dáta. Tejto publikácii venoval pozornosť inžinier menom Doug Cutting, ktorý spravil voľne stiahnuteľnú, open-source implementáciu tejto metódy, napísanú v jazyku Java a nazval ju *Hadoop*. Táto implementácia mala veľký úspech a stala sa synonymom pre pojem Big data. *Hadoop* sa skladá z týchto dvoch častí <sup>[1]</sup>:

- *Hadoop Distributed File System* (HDFS) – umožňuje ukladať a manipulovať s dátami na počítačovom clustri. Stará sa o to, aké dáta sú uložené na ktorom uzle, a uchováva kópie dát pre prípad výpadku nejakého uzla.
- *MapReduce* framework – číta dáta z HDFS, paralelne ich spracuje a výstup zapíše naspäť do HDFS.

### 4.2. Apache Spark

Pri vývoji softvérovej knižnice sa vývojári snažia, aby ich produkt bol podporovaný viacerými programovacími jazykmi, bol menej závislý na hardvéri, a viac priateľský k pamäti počítača. Presne s takouto ideou bol spravený aj *Apache Spark*. *Apache Spark* je aktuálne vedúca metóda pre spracovanie Veľkých dát na počítačovom clustri, ktorá dokáže komunikovať s HDFS, a nahradila tradičný *Hadoop*. Je efektívnejšia pri spájaní niekoľkých operácií, no hlavne je jednoduchšia na používanie. Je to knižnica, ktorá je kompatibilná s programovacími jazykmi ako Python, Scala, R alebo Java. Spark je napísaný v jazyku Scala, takže bude najrýchlejšie komunikovať s týmto jazykom. <sup>[1]</sup>



# 5. Spracovanie Veľkých dát v programovacích jazykoch

## 5.1. Matlab

Existujú situácie, pri ktorých nie je možné veľké dáta spracovať naraz, ale môžeme ich rozdeliť na viacero častí a tie analyzovať jednotlivo. Výsledky z jednotlivých častí môžu byť následne jednoducho spojené. Takýmto spôsobom môžeme opakovane prechádzať rôzne súbory, načítať časti, o ktoré sa zaujímate, analyzovať ich, uložiť ich výsledky a opakovat tento cyklus. Tento proces je jednoducho a účinne umožnený v softvéri Matlab pomocou funkcie *datastore*. Nie všetky algoritmy sú dizajnované tak, aby spracovali veľký objem dát v rozumnom časovom období alebo množstve pamäte. *MapReduce* dokáže tento problém riešiť pomocou paralelizácie na viacerých jadrách, procesoroch alebo počítačových clustroch, a tak redukovať výpočetný čas.

### 5.1.1. Datastore

*Datastore* je funkcia na čítanie jedného, alebo viacerých súborov a dát. Slúži ako úschovňa pre dáta rovnakej štruktúry a formátu. Napríklad, každý súbor v *datastore* musí obsahovať prvky toho istého typu, v rovnakom poradí a rovnakým oddelovačom. *Datastore* je užitočný, keď: <sup>[19]</sup>

- Súbory sú priveľké, aby sa zmestili do pamäte. *Datastore* umožní čítať a analyzovať dáta zo súboru v menších častiach, ktoré sa zmestia do pamäte.
- Každý súbor z kolekcie má ľubovoľný názov a nie je nutné, aby súbory mali názvy v nejakom vzore alebo poradí.
- Nie je potrebné, aby boli všetky dáta načítané v pamäti, čo výrazne ovplyvní celkový čas na výpočet.

### 5.1.2. MapReduce

*MapReduce* je programovacia technika pre analyzovanie dát, ktoré sú väčšie ako pamäť počítača, a preto používa *datastore*, funkciu pre spracovanie dát v menších častiach, ktoré sa zmestia do pamäte. Každá časť dát musí prejsť fázou mapovania (map), ktorá dáta formátuje (filtruje a zoradí) pre ďalší proces. Tieto prechodné dáta postupujú do redukčnej (reduce) fázy, z ktorej dostaneme finálne výsledky. Tieto fázy sú vyjadrené pomocou mapovacích a redukčných funkcií, ktoré sú hlavnými vstupmi do *MapReduce*. Existujú nekonečné kombinácie mapovacích a redukčných funkcií, a preto je táto technika flexibilná a výkonná. Výhodou je možnosť výpočtu aj na osobnom počítači za účelom testovania. <sup>[19]</sup>

## 5.2. Python – PySpark

PySpark je rozhranie, ktoré importujeme do programu napísaného v jazyku Python, ktoré nám zabezpečí komunikáciu s počítačovým clustrom. Dáta, s ktorými pracujeme, sú reprezentované ako RDD (Resilient Distributed Dataset), čo je len zbierka objektov pre čítanie. Tieto objekty môžu byť textovými reťazcami, celočíselnými dátovými typmi a inými. Objekty sú distribuované cez rôzne uzly v počítačovom clustri, pričom sa nestaráme, kde sa aké nachádzajú. Obsah v RDD je nemenný, a pokiaľ ho potrebujeme zmeniť, musíme vytvoriť nové RDD. Existujú dva typy RDD – párované a nepárované. Párované RDD sa skladá z dvoch prvkov, kde prvý označuje poradie a druhý danú hodnotu. To využijeme, ak potrebujeme spojiť viacero RDD, alebo potrebujeme separovať prvky z RDD s rovnakými vlastnosťami. Pokiaľ potrebujeme v RDD vykonať operácie, ktoré pôsobia na každý prvok zvlášť, bez ohľadu na jeho poradie, potrebujeme nepárované RDD. Existujú dva typy PySpark operácií <sup>[1]</sup>:

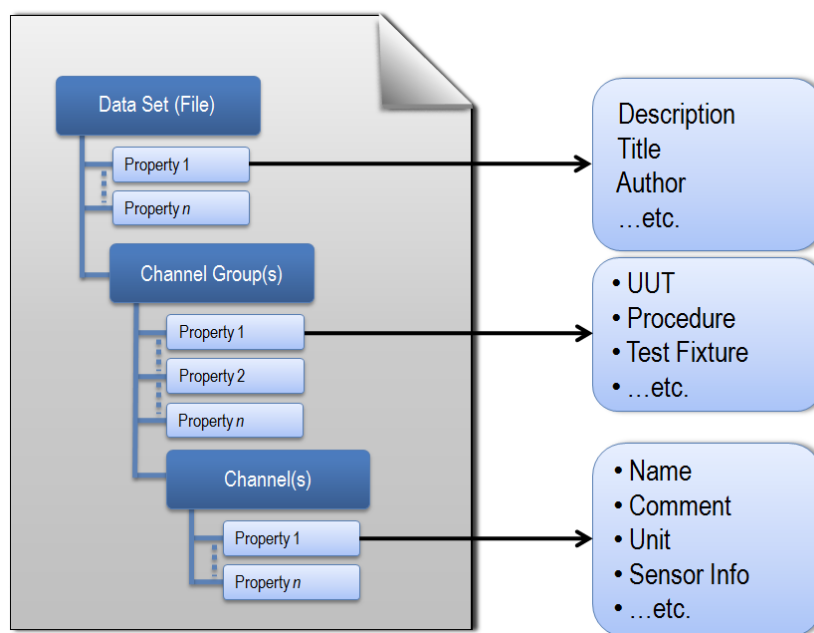
- Volanie metód na SparkContext, triedu, ktorá komunikuje so Spark clustrom, ktorá vytvorí RDD. To môže byť vytvorené napríklad z dát na našom počítači.
- Volanie metód na RDD, ktoré vrátia nové RDD s nejakou zmenou, alebo výsledok operácie na tomto RDD.

## 6. Súborový formát TDMS

V tejto práci sa budeme zaoberať dátami, ktoré sú vo formáte TDMS<sup>1</sup> (\*.tdms). Tento formát vyvinula spoločnosť National Instruments pre ich produkty za účelom prehľadnosti dát, ktoré môžu mať obrovskú veľkosť. V našom prípade tieto dáta získavame z meracej karty a softvéru Signal Express od tohoto výrobcu. Tieto dáta sú špecifické ich objemom, pretože sú zbierané s veľkou vzorkovacou frekvenciou, a tak už pár sekúnd merania vytvorí súbory, pri ktorých vzniká problém s časom ich spracovania. Ich veľkosť nie je taká, ako napríklad pri súboroch internetových gigantov, kde tieto dáta majú stovky terabajtov, ale predsa považujeme tieto súbory za Veľké dáta. Najväčší problém predstavuje získanie dát z TDMS súborov. Nie každý softvér dokáže pracovať s týmto formátom, buď pre hardvérové obmedzenia, alebo pre slabú podporu formátu v danom softvéri. Preto je vhodné tieto dáta konvertovať do iných formátov, ktoré budú kompatibilnejšie s viacerými softvérmi.

### 6.1. Štruktúra formátu TDMS

Tento súborový formát sa skladá z troch hladín, zobrazených na obrázku 6.1, ktorých poradie je nasledovné – súbor, skupina a kanál. Súbor môže obsahovať neobmedzený počet skupín, a každá skupina neobmedzený počet kanálov. Takáto štruktúra umožňuje používateľovi organizáciu dát a lepšie im rozumieť. Napríklad môžeme mať viacero kanálov, ktoré zodpovedajú rôznym zložkám senzora, prípadne rôznym typom meraní. [7]



Obr. 6.1: Štruktúra formátu TDMS [7]

<sup>1</sup>Technical Data Management Streaming

## 6.2. Otvorenie formátu TDMS

Jedným z hlavných problémov pri práci s formátom TDMS je jeho kompatibilita so softvéromi, ktoré dokážu z neho čítať dáta. Pri použití softvéru od spoločnosti NI, je práca s dátami vyladená, ale nie je používateľsky prívetivá. Existuje však viacero spôsobov, ako sa dostať pomerne ľahko k dátam, no s pár obmedzeniami, ktoré si popíšeme nižšie.

### 6.2.1. Microsoft Excel a OpenOffice

Tieto tabuľkové editory, ktoré má väčšina počítačov a dokáže s nimi pracovať väčší počet ľudí, sú ideálnym nástrojom pre prezeranie dát. Preto NI vydal voľne stiahnuteľný doplnok do týchto dvoch editorov, ktorý má slúžiť hlavne pre rýchle prezeranie malých dát bez nutnosti softvéru od NI. Pri oboch editoroch narazíme na problém s obmedzeným počtom riadkov, ktoré dokážu prečítať. Práca s dátami TDMS pomocou tohoto tabuľkového editora je pomalá, a pri väčších meraniach je nutné súbory opätovne načítať. [8]

### 6.2.2. Matlab

Pri použití softvéru Matlab sú tiež k dispozícii nástroje, ktoré dokážu načítať dáta do pracovného priestoru softvéru a následne s nimi pracovať. Túto možnosť môžeme považovať za veľmi účinnú, pretože môžeme dáta spracovávať priamo ďalej, bez nutnosti ich ukladať do nášho požadovaného formátu, a opätovne načítať. Nástroje na čítanie dát sú voľne stiahnuteľné zo stránok NI, ale sú celkom komplikované. Stiahnutý ZIP súbor obsahuje príkladné súbory ako čítať TDM a TDMS súbory v softvéri Matlab. Tento nástroj však pracuje za pomoci súboru formátu dll<sup>2</sup>, ktorý je nutné samostatne načítať, a načítané dáta v pracovnom priestore nie sú prehľadne zobrazené. Pre tento problém vznikla funkcia s názvom TDMS Reader, ktorá pracuje bez použitia dll súboru, a je voľne stiahnuteľná na MathWorks File Exchange. Túto funkciu budeme využívať vo vytvorenom grafickom prostredí, popísaného v kapitole 8, a pre porovnanie rýchlosti. [5]

### 6.2.3. Python

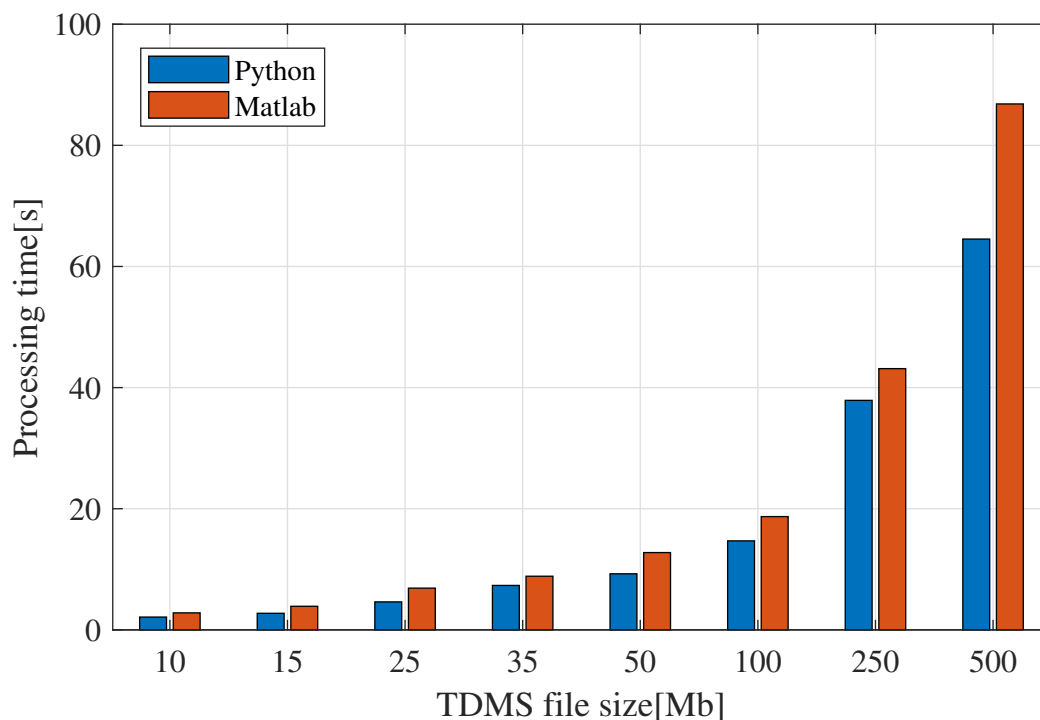
Ďalšou možnosťou je čítanie súborového formátu TDMS pomocou python balíka npTDMS, ktorý je postavený na balíku NumPy. [4] Získané dáta sú uložené v numpy poli. NumPy je najzákladnejším balíkom pre vedecké výpočty v jazyku Python, ktorý umožňuje vytvárať veľké, viac dimenzionálne polia, prípadne matice a obsahuje zložitejšie matematické funkcie, ktoré dokáže aplikovať na tieto polia. Táto metóda čítania dát pomocou jazyka Python by mala byť najrýchlejšia z dôvodu absencie grafického prostredia, ale z pohľadu náročnosti na používanie najzložitejšia, pretože kladie väčšie nároky na programovacie schopnosti. [16]

---

<sup>2</sup>Dynamic-link library – dynamicky spojená knižnica používaná v operačnom systéme Microsoft Windows

## 6.3. Porovnanie otvorenia súborov v jazyku Matlab a Python

Samotné načítanie súboru nie je vhodné merať, a preto budeme merať čas konverzie súboru do formátu csv<sup>3</sup>. Dvadsať krát meriame toto otvorenie a zaznamenávame časy, z ktorých priemernú hodnotu vynesieme do grafu. Odchýlka od priemernej hodnoty je v tomto prípade zanedbateľná a nie je ju potrebné zahrnúť v grafe. Výsledky preto porovnávame v stĺpcovom grafe na obrázku 6.2, ktorý prehľadne zobrazuje namerané časy. Pri väčších súboroch spracovaných v prostredí Matlab môžeme vidieť, že čas v závislosti na veľkosti súborov približne lineárne narastá. Napríklad čas výpočtu pre 250 MB súbor je približne 40 sekúnd, a pre dvojnásobne väčší súbor 80 sekúnd. Na grafe môžeme vidieť prevahu načítania súboru v jazyku Python, ktorý je približne o 25 percent rýchlejší pri najväčšom súbore, ako načítanie toho istého súboru v jazyku Matlab. Táto prevaha je spôsobená jednoduchosťou jazyka bez grafického rozhrania.



Obr. 6.2: Porovnanie rýchlosti otvorenia súboru TDMS v jazyku Matlab a Python

Veľký podiel na rýchlosti načítania má hardvér, na ktorom sme meranie prevádzali. V našom prípade sme výpočty vykonávali na osobnom počítači s operačným systémom Windows s týmito parametrami:

- Procesor – Intel(R) Core(TM) i5-3320M 2.60GHz
- Pamäť RAM – 4 GB DDR3
- Pevný disk – ADATA SU800 R/W 410/390 [MB/s]

<sup>3</sup>Comma-separated values – textový formát, ktorý používa čiarku pre oddelenie hodnôt

## 6.4. Obmedzenia softvérov

Každý z popísaných programov má svoje medze. Najviac to je vidieť pri tabulkovom editore Microsoft Excel, ktorý na takto veľké dáta nie je stavaný, a dokáže zobrazíť maximálne údaje z 1,048,576 vzorkov. Pri ďalších dvoch softvéroch záleží na veľkosti pamäte RAM. V softvéri Matlab s pamäťou RAM o veľkosť 4GB je možné prečítať TDMS súbor s približne 500 miliónmi vzorkov, ktoré sú uložené na jednotlivých riadkoch. Pri jazyku Python sme nenarazili na obmedzenia, z dôvodu „malých“ testovacích súborov.

### 6.4.1. Veľkosť TDMS súboru pre rôzne časy merania

V tejto sekcii spravíme jednoduchý výpočet veľkosti dát pre rôzne časy merania. Budeme vychádzať z nameraného súboru o veľkosti 503 MB, ktorý bol meraný 20 minút a 7 sekúnd so vzorkovacou frekvenciou 51200 vzorkov za sekundu. Takýto súbor obsahuje celkovo 61798400 vzoriek. Minúta merania vychádza približne na 25 MB. V nasledujúcej tabuľke sú uvedené výpočty veľkosti súborov a počet vzoriek pre rôzne dĺžky merania.

| Čas merania[min]                | 1    | 30    | 60     | 120    | 360     | 720     | 1440    |
|---------------------------------|------|-------|--------|--------|---------|---------|---------|
| Veľkosť TDMS[MB]                | 25   | 750   | 1500   | 3000   | 9000    | 18000   | 36000   |
| Počet vzoriek( $1 \cdot 10^3$ ) | 3072 | 92160 | 184320 | 368640 | 1105920 | 2211840 | 4423680 |

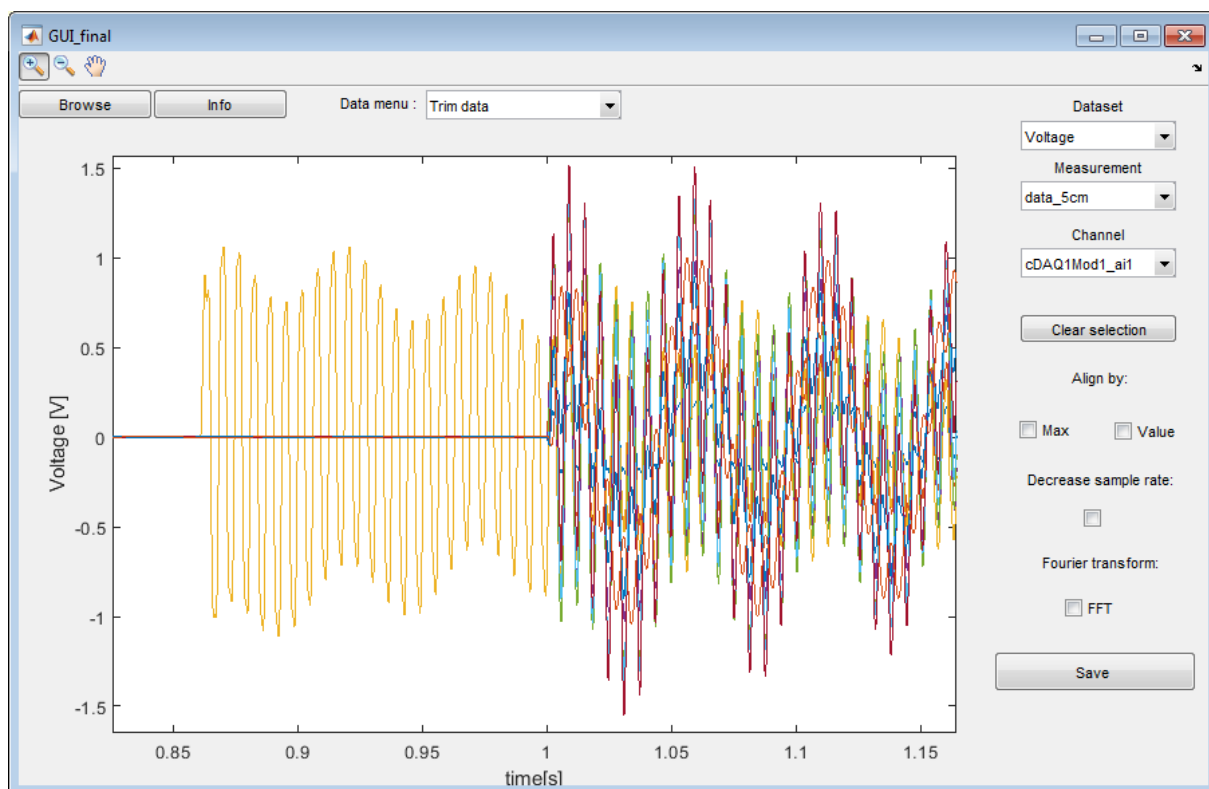
Po porovnaní predchádzajúcich obmedzení môžeme vidieť, že osobný počítač s vyššie uvedenými parametrami by zvládol prácu so súbormi TDMS, ktorých dĺžka merania bola o málo väčšia ako dve hodiny.

## 7. Preprocessing dát

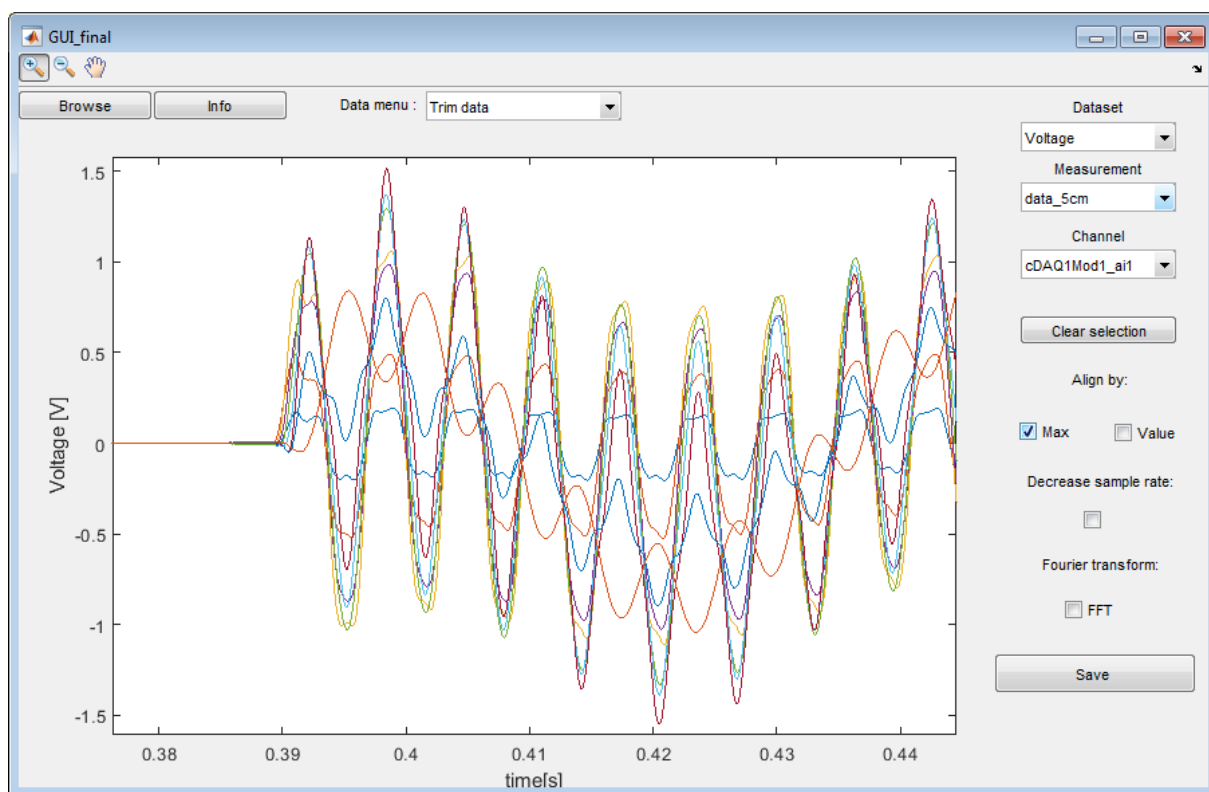
Monitorovanie pomocou senzorov dnes obnáša zložitejší proces. Týchto senzorov používame veľké množstvo, pričom spracovanie dát z nich zaberie obrovské množstvo času. Tomu sa snažíme predísť, a vyberáme z dát len dôležité informácie, aby sme nemuseli spracovávať všetky dáta. Hlavným cieľom preprocessingu dát je zrýchlenie a efektívnejšie spracovanie dát. Preprocessing sa môže vykonať ešte pred uložením dát do súboru na mikrokontroleri alebo inom zariadení. To môže byť napríklad nejaký druh filtrácie signálu. Po tom, ako získame prístup k nameraným dátam, prichádza na rad ich spracovanie, ktoré by malo používateľovi pomôcť zjednodušiť prácu s dátami, vybrať z nich oblasti záujmu a vykonať na nich napríklad filtrovanie, prevzorkovanie, alebo transformáciu do frekvenčnej oblasti. V prípade formátov, ktoré sú náročnejšie na otvorenie, je vhodné konvertovať tieto dáta do iných formátov, ktoré budú kompatibilnejšie s viacerými softvérmi na ich analýzu.

### 7.1. Vykreslenie dát

Po vykreslení dát získame vizuálnu informáciu o nameraných dátach. Dáta je vhodné vykresliť hlavne v prípade, pokiaľ je na nich zreteľne vidieť neželaný jav, a tak môžeme celé meranie odstrániť. Pri porovnávaní dát z viacerých meraní je vhodné dáta zarovnať, aby sme na nich mohli lepšie pozorovať zmeny. Jedna z možností je dáta zarovnať podľa maximálnej hodnoty impulzu, alebo podľa ľubovoľnej hodnoty impulzu, ktorú ako prvú dáta dosiahnu. Na obrázku 7.1 sú nezarovnané dáta. Kanál vykreslený žltou farbou začína zreteľne skôr, ako ostatné kanály. Tieto dáta môžeme zarovnať podľa hodnoty napätia, ktorú tieto dáta ako prvú dosiahnu. Takto zarovnané dáta sú zobrazené na obrázku 7.2.



Obr. 7.1: Nezarovnané dáta z viacerých meraní

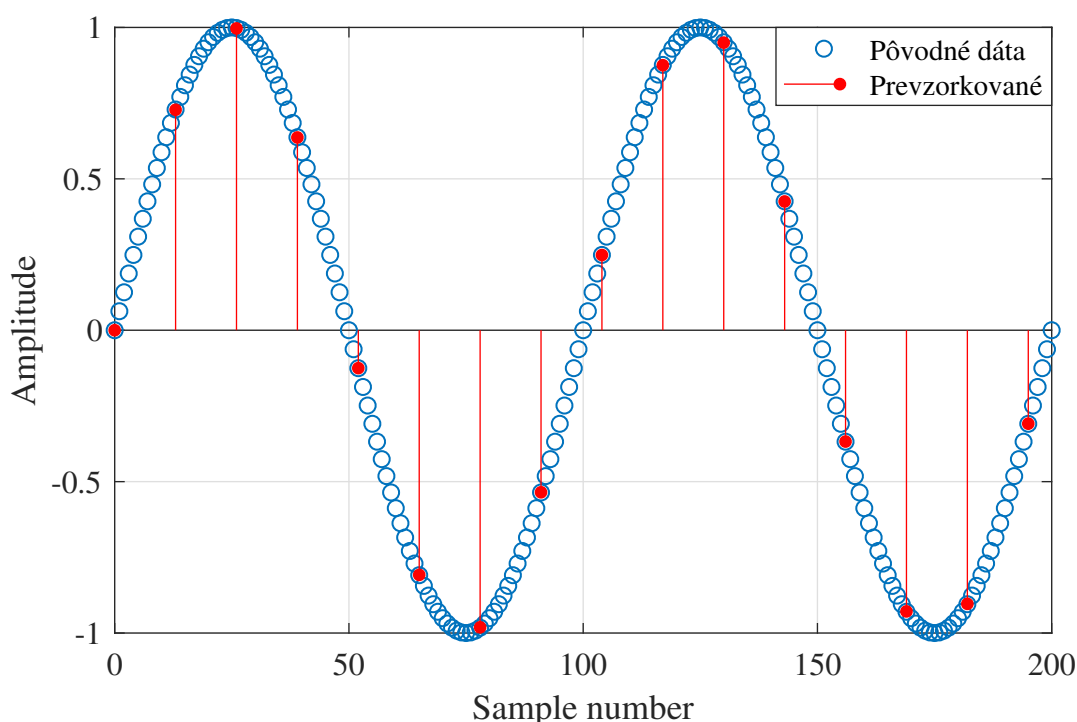


Obr. 7.2: Zarovnané dáta z viacerých meraní



## 7.2. Zmenšenie dát

V tejto časti popíšeme pár možností, ktorými môžeme zmenšiť dáta. Jednou z nich je zmenšenie počtu vzoriek dát. Nami používané dáta majú veľkú vzorkovaciu frekvenciu pre čo naj dôverihodnejšie zachytenie zmien meranej veličiny. Vhodným znížením vzorkovacej frekvencie môžeme výrazne zmenšiť veľkosť dát bez väčšej zmeny na vykreslených dátach. V softvéri Matlab používame na zmenšenie vzorkovacej frekvencie funkciu *decimate* <sup>[12]</sup>. Ukážka pôvodných dát a dát so zmenšenou vzorkovacou frekvenciou je zobrazená na obrázku 7.3. Ďalšou z možností je selekcia dát v oblasti záujmu, pri ktorej pracujeme len s vybranými dátami. Pri zmene dátového typu môžeme tiež zmenšiť veľkosť dát. Napríklad pri zmene z typu *double* to *single* ušetríme až polovicu veľkosti pôvodných dát.

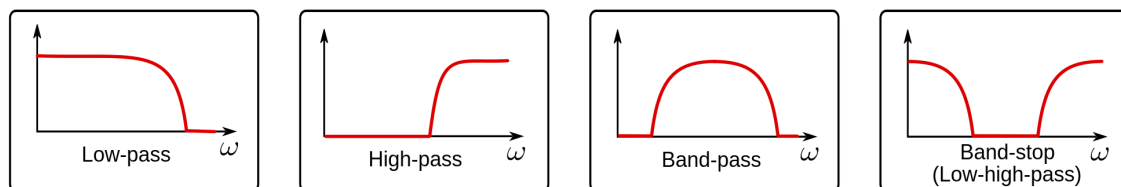


Obr. 7.3: Znázornenie dát po zmene vzorkovacej frekvencie

## 7.3. Filtrovanie dát

Filtrovaním dát dosiahneme zmenšenie ich veľkosti, odstránenie nechcených frekvencií, vyhladenie zobrazených dát a iné. Snažíme sa, aby po filtrácii dát ostala užitočná informácia, ktorú potrebujeme pre nasledujúcu analýzu. Filtrovanie môže byť vykonané aj externe, napríklad na mikroprocesori, ešte pred samotným spracovaním na počítači. Základné filtrovanie dát môžeme vykonať pomocou nasledujúcich druhov filtrov, ktoré sú zobrazené na obrázku 7.4. Pre filtrovanie frekvencií od najnižšej až po určitú maximálnu frekvenciu, používame filter typu dolný priepust (Low-pass). Takýto filter môžeme využiť pri eliminácii vysokofrekvenčného šumu. Naopak, ak sa zaujímate len o frekvencie nad určitou hodnotou, použijeme filter typu horný priepust (High-pass). Pokiaľ chceme filtrovať len určité frekvencie, o ktoré sa zaujímate, použijeme filter typu pásmový prie-

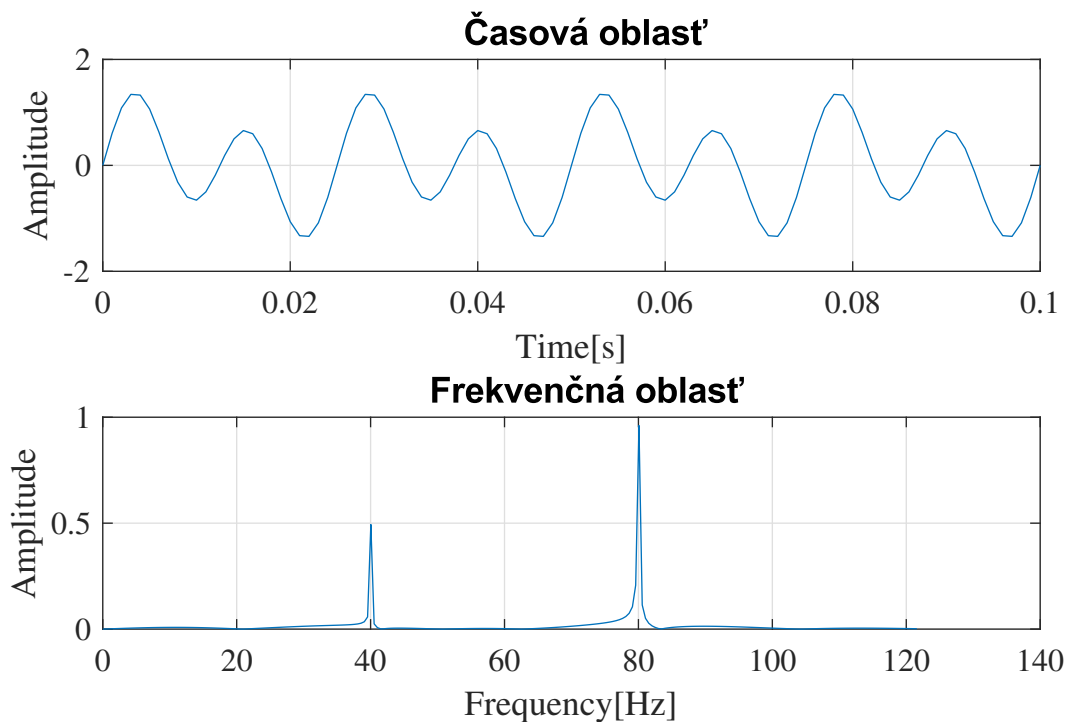
pust (Band-pass). Tento filter môže byť vytvorený zo spojenia filtra typu dolný priepust a horný priepust. Ak sa v signáli vyskytujú parazitné frekvencie, použijeme pásmovú zadrž (Band-stop), ktorá vybrané pásmo frekvencií tlmí. Všetky tieto filtre sa snažia frekvencie, o ktoré máme záujem, prepúšťať s minimálnym útlmom. Takto prefiltrované dáta je možné lepšie spracovať a zabrániť skresľovaniu nechceným šumom.



Obr. 7.4: Základné typy filtrov [2]

## 7.4. Transformácia dát

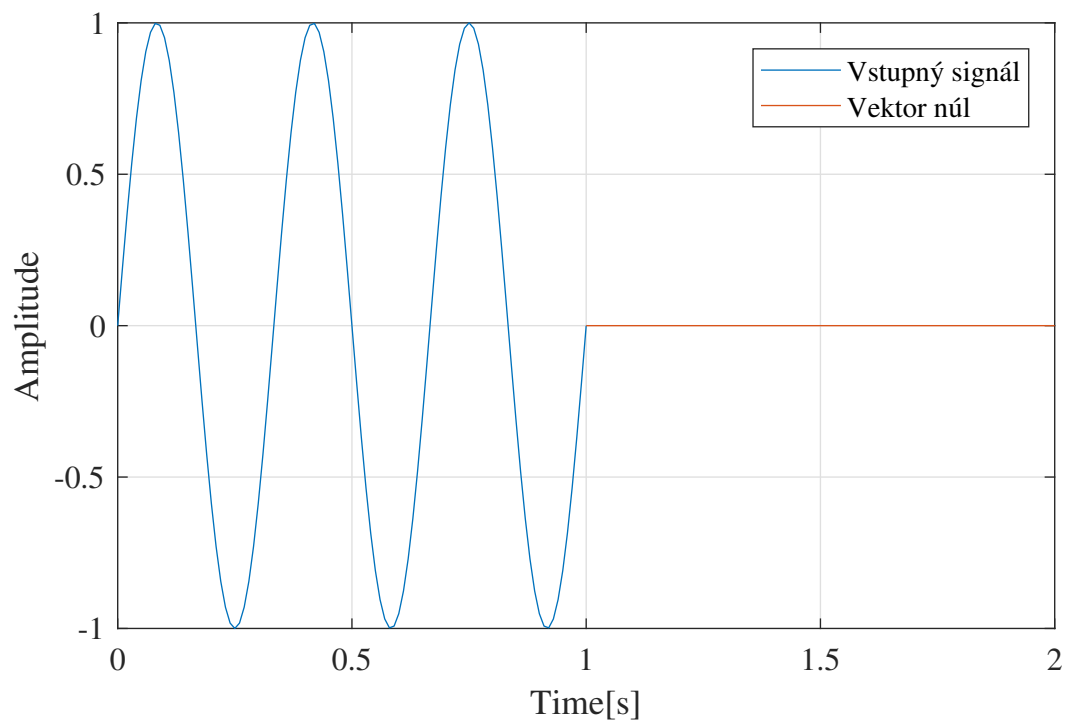
Jednou z metód, ktorú budeme na dátach aplikovať je Fourierova transformácia, ktorá transformuje dáta z časovej oblasti do oblasti frekvenčnej. Táto transformácia sa často využíva v priemysle, vede a matematike. Dĺžka dát, s ktorými pracujeme je veľká, preto pre získanie diskretnej Fourierovej transformácie(DFT) používame FFT(Fast Fourier Transform), čo je rýchlejší algoritmus pre počítanie DFT. Tento algoritmus publikovali v roku 1965 Cooley a Tukey, čo viedlo k rozvoju digitálneho spracovania dát. Výsledok Fourierovej transformácie je frekvenčná a fázová zložka. Väčšinou sa o fázovú zložku nezaujíname, a stačia nám informácie z frekvenčnej zložky. Výhoda tejto transformácie je tiež v zmene veľkosti dát, ak sa zaujíname len o určité frekvencie. Po transformovaní signálu do frekvenčnej oblasti vyrežeme úsek frekvencií, o ktoré máme záujem, a pracujeme ďalej s orezaným signálom. Na obrázku 7.5 je zobrazený signál v časovej oblasti a po jeho transformovaní do frekvenčnej oblasti. Signál je zložený zo súčtu dvoch sínusoid, kde pri prvej je amplitúda 0.5 a frekvencia 40 Hz, a pri druhej amplitúda 1 a frekvencia 80 Hz. Tieto frekvencie môžeme vidieť na horizontálnej osi v zobrazení tohoto signálu vo frekvenčnej oblasti. Po transformácii sa amplitúdy týchto sínusoid nemenia. [6]



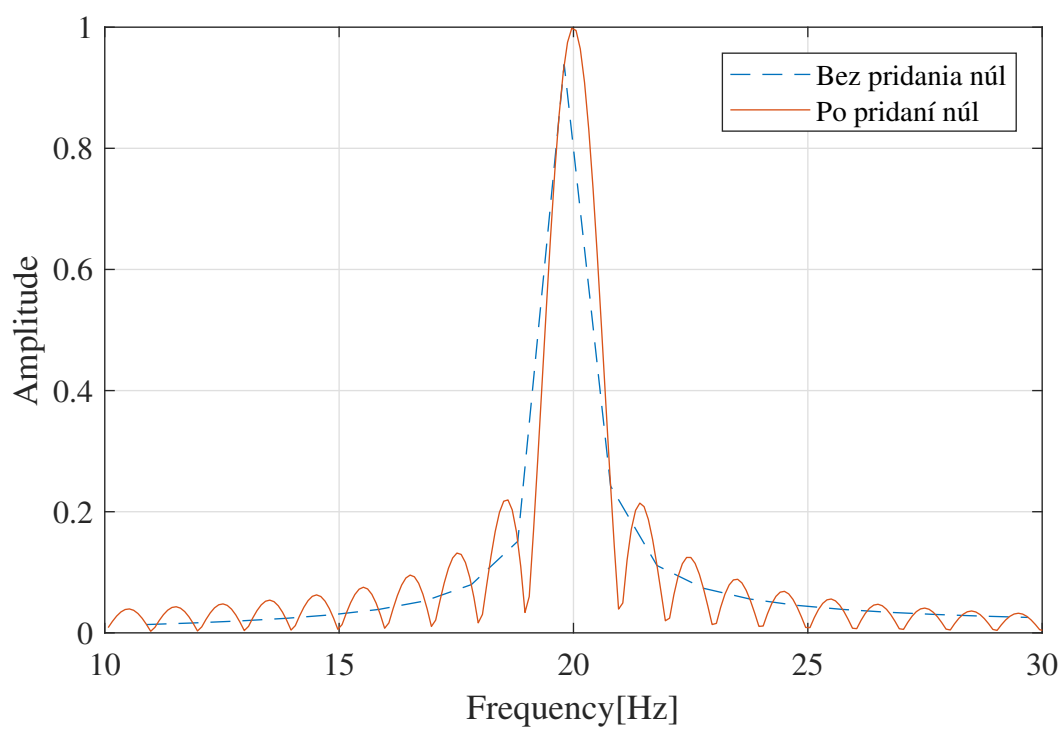
Obr. 7.5: Signál v časovej a frekvenčnej oblasti

#### 7.4.1. Zero padding

Metóda zero padding spočíva v rozšírení signálu o nuly, dáta bez informácie, čím sa zväčší dĺžka signálu. Dĺžka signálu je dôležitá pre frekvenčné rozlíšenie po transformovaní signálu do frekvenčnej oblasti. Rozlíšenie na frekvenčnej osi závisí na vzorkovacej frekvencii a dĺžke signálu, ktoré sú priamo úmerné. Toto rozlíšenie vyjadríme ako  $\frac{2 \cdot F_s}{L}$ , kde  $F_s$  je vzorkovacia frekvencia a  $L$  dĺžka signálu. Tiež je výhodné, ak je dĺžka signálu mocninou čísla dva. Vtedy je FFT algoritmus najúčinnnejší a ovplyvňuje rýchlosť transformácie. Na obrázku 7.6 je zobrazený signál s rozšírením o vektor 500 núl. Fourierova transformácia signálu bez pridania núl a po ich pridání je zobrazená na obrázku 7.7. Táto metóda môže spôsobiť mierne rozkývanie signálu po transformácii okolo maximálnej hodnoty, pokiaľ rozšírime signál o priveľa núl. Po pridání núl sa dostaneme k presnejším výsledkom, ktoré nie sú skreslené slabým frekvenčným rozlíšením.



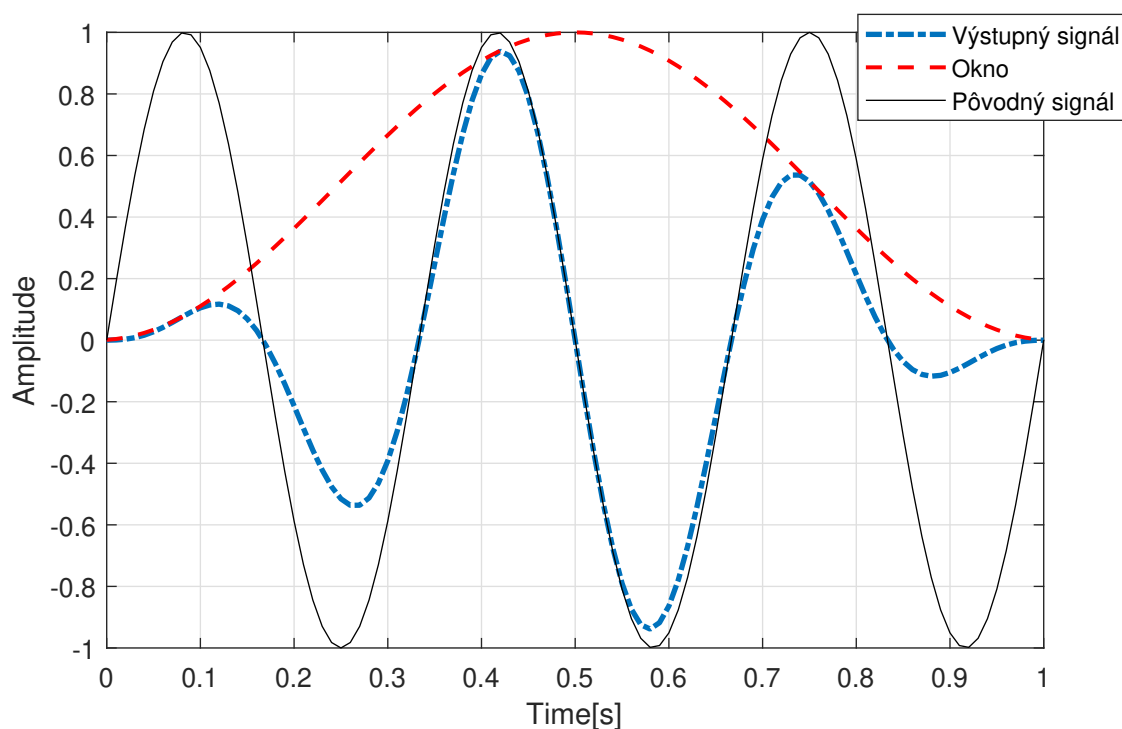
Obr. 7.6: Zero padding



Obr. 7.7: Porovnanie transformácie do frekvenčnej oblasti bez a s pridaním núl

### 7.4.2. Windowing

Pred transformovaním signálu do frekvenčnej oblasti pomocou Fourierovej transformácie je vhodné dáta upraviť pomocou metódy s názvom windowing, ktorá zabezpečí spojitosť dát na začiatku a konci pri ich rozširovaní počas transformácie. Ak je signál periodický, a počet periód, ktoré sa zmestia do celkovej dĺžky signálu je celým číslom, nie je potrebné vykonávať túto metódu. Pre signál, kedy nie je počet periód celým číslom, nastáva problém v podobe výskytu vyššej frekvencie vo Fourierovej transformácii, ktorá sa v pôvodnom signále nevyskytuje. Zjednodušene povedané, že nastáva nespojitosť pri spájaní týchto signálov. Existuje niekoľko druhov „okien“, ale najpoužívanejšie sú Hanning window a Hamming window. Hanning window je krivka, ktorá je na koncoch nulová, a jej maximum je v jej strede, okolo ktorého je symetrická. Toto zabezpečí celkovú elimináciu nespojitosti signálu po vynásobení týmto oknom. Krivka Hamming window nemá na koncoch nulové hodnoty, preto po vynásobení týmto oknom ostane na signále čiastočná nespojitosť. Týmto oknom však veľmi dobre odstránime najbližší vrchol po maximálnom. Vynásobenie signálu pomocou Hanning window je zobrazené na obrázku 7.8. [9]



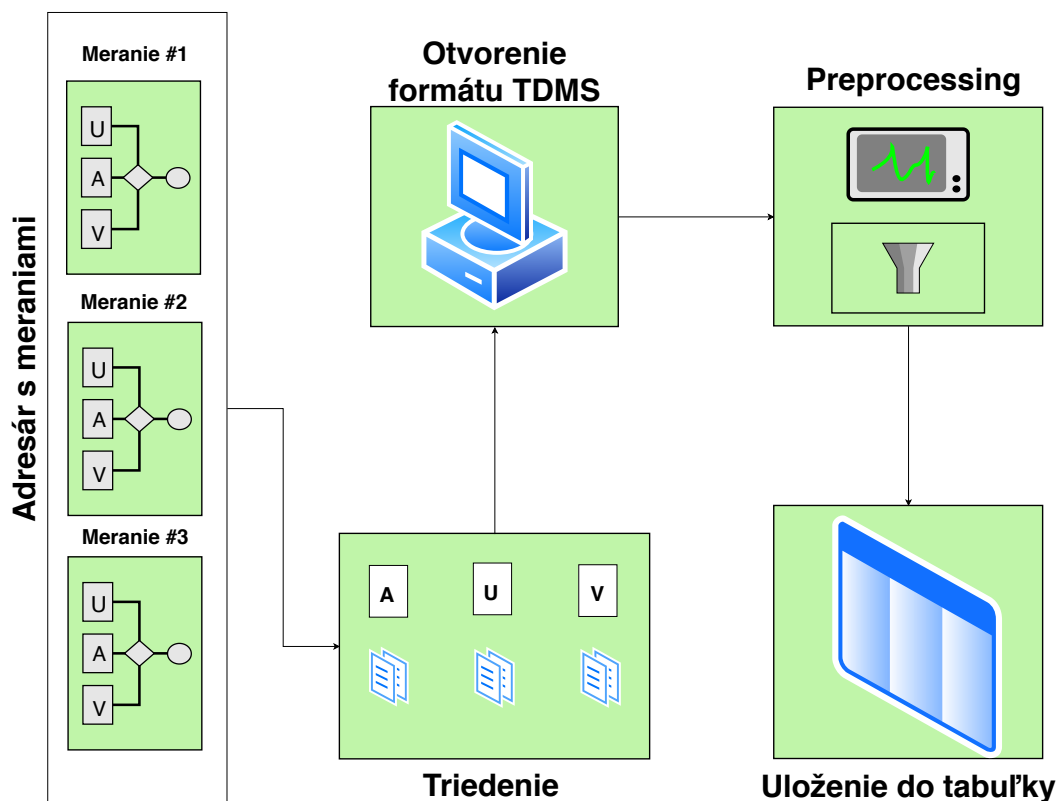
Obr. 7.8: Windowing

## 7.5. Konverzia dát

Poslednou časťou pri preprocessingu dát je možnosť používateľovi uložiť dáta v zvolenom formáte, ktorý by mal byť jednoduchší pre opätovné otvorenie a kompatibilný s viacerými softvérmi na analýzu dát. Vhodným výberom formátu môžeme ovplyvniť rýchlosť nasledujúcej analýzy. Možností na uloženie do iných formátov je veľa, ale najjednoduchším a najzákladnejším formátom je textový súbor (\*.txt,\*.csv). Textový súbor dokážeme bez problémov otvoriť vo všetkých druhoch softvérov pre preprocessing. V našom prípade v softvéri Matlab sú dáta dočasne uložené v pracovnom priestore v podobe vektorov a matic, a tak máme možnosť tieto dáta uložiť do rôznych formátov, podľa ďalšieho spracovania. V prípade, že máme viacero meraní a kanálov a dáta sú uložené v štruktúre, môžeme štruktúru uložiť v binárnom formáte (\*.mat) alebo uložiť separátne polia štruktúry do textových súborov.

## 8. Grafické rozhranie pre preprocessing dát

Hlavnou myšlienkou pre tvorbu grafického rozhrania je zjednodušenie a urýchlenie preprocessingu dát. Snažíme sa do tohoto rozhrania implementovať jednotlivé funkcie z predchádzajúcej kapitoly, aby nemuseli byť vykonávané separátne, ale aby boli vykonané na dátach podľa nastavení v grafickom rozhraní. Ďalší problém, ktorý rieši toto rozhranie, je počet súborov, ktoré chceme analyzovať. V dnešnej dobe sa pri experimentoch používa obrovské množstvo senzorov. V našom prípade sa dáta zo senzorov ukladajú do formátu TDMS, ktorý vzniká softvérom Signal Express. Pre každý senzor vznikne samostatný TDMS súbor. Pokiaľ by sme mali každý súbor načítať separátne a vykonať na ňom všetky potrebné funkcie, zabralo by nám to veľmi veľa času. Tento problém je riešený pomocou rozhrania tak, že v cykle načítava všetky zvolené súbory, a podľa nastavení vykoná na dátach zvolené úpravy. Ďalšou výhodou je triedenie súborov pri viacerých meraniach, kedy je vstupom do tohoto rozhrania adresár, ktorý obsahuje podadresáre z jednotlivých meraní. V každom podadresári sa nachádzajú súbory formátu TDMS z použitých senzorov pri meraní. Grafické rozhranie zjednotí tieto súbory podľa typu senzora z každého merania a vykreslí pre ich porovnanie. Následne je možné tieto dáta upraviť podľa nasledujúcich nastavení a uložiť do prehľadnej tabuľky. Pre lepšie porozumenie tohoto problému je tento postup zobrazený v diagrame na obrázku 8.1.

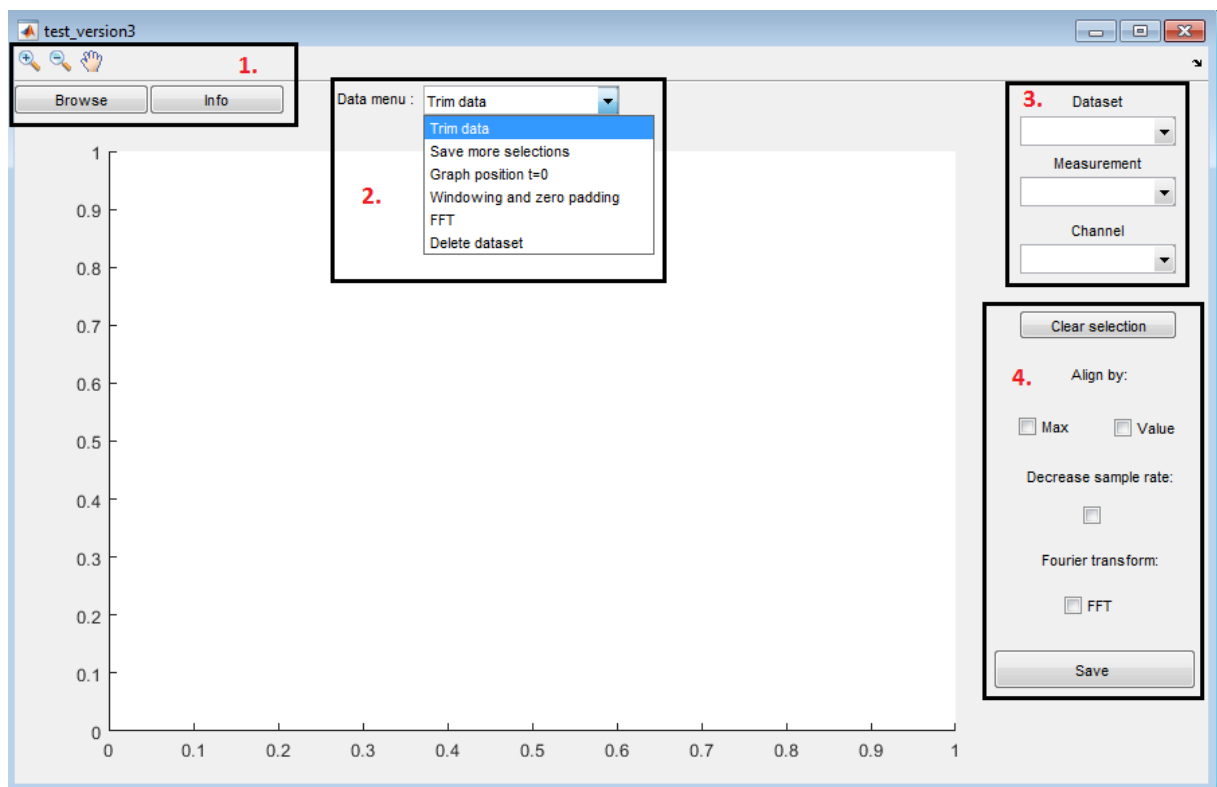


Obr. 8.1: Diagram algoritmu grafického rozhrania

Prvý blok znázorňuje vstupný adresár do grafického rozhrania. Podbloky Meranie #1-3 znázorňujú podadresáre z jednotlivých meraní so súbormi U,A,V, ktoré sú vo formáte TDMS. V ďalšom bloku je znázornené triedenie podľa typu merania, kde A môže predstavovať meranie zrýchlenia. To je zabezpečené názvom súboru, a tak je nutné, aby súbory rovnakého typu merania mali ten istý názov v každom podadresári. Z každého podadresára sa vyberú súbory s rovnakým názvom, a vytvoria prehľadnú štruktúru delenú podľa tohoto názvu. Takto roztriedené súbory sa postupne načítajú, spracujú podľa našich nastavení preprocessingu a uložia do tabuľky v pracovnom priestore programu Matlab. Pre funkčnosť tohoto rozhrania je nutná knižnica TDMS Reader dostupná na Mathworks File Exchange. [5]

## 8.1. Popis oblastí grafického rozhrania

Grafické rozhranie môžeme rozdeliť do štyroch hlavných oblastí, ktoré sú zobrazené na obrázku 8.2. V každej časti popíšeme jej funkcie.



Obr. 8.2: Oblasti v grafickom rozhraní

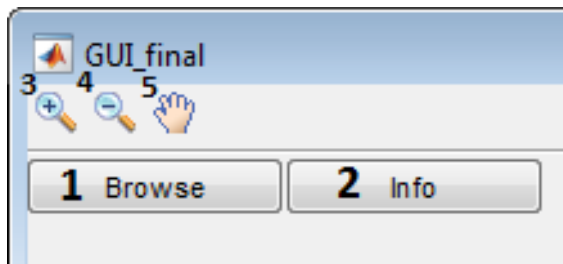
Sú to oblasti:

1. Hlavné menu.
2. Menu s funkciami pre vybratý kanál.
3. Výber dát.
4. Funkcie a export viacerých meraní.



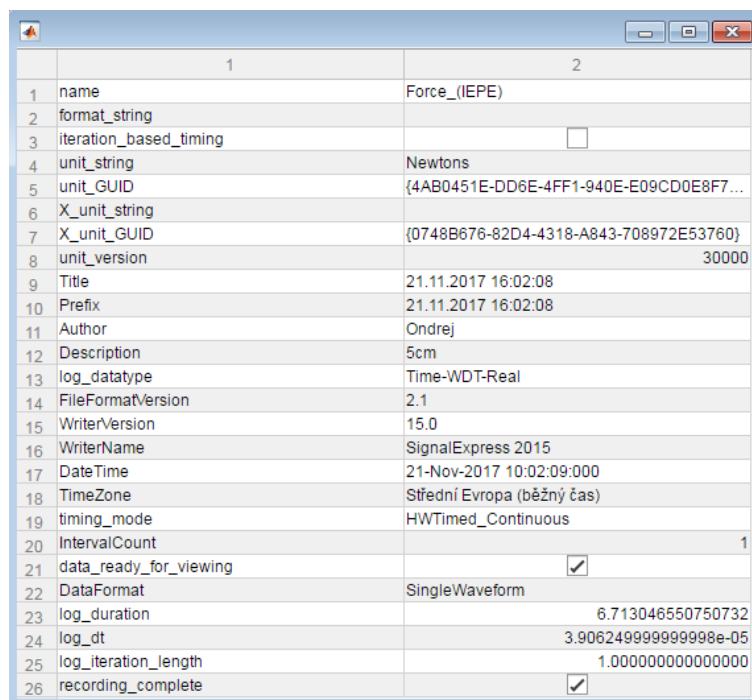
### 8.1.1. Hlavné menu

Táto oblasť slúži na počítačový výber dát a zobrazenie informácií o meraní. Tieto funkcie používame väčšinou len pre úvodný výber TDMS súborov, bez ktorého grafické rozhranie nefunguje. Ďalej sa tu nachádzajú nástroje na priblíženie a pohyb v oblasti grafu. Rozloženie tejto oblasti je zobrazené na obrázku 8.3.



Obr. 8.3: Rozloženie v prvej oblasti

1. Otvorí okno pre výber adresára. Ten by mal obsahovať podadresáre jednotlivých meraní. Súbor v podadresároch musia mať rovnaké názvy podľa typu merania.
2. Zobrazenie informácií o vykreslenom meraní v novom okne. Obsahuje všetky informácie z načítaného TDMS súboru. Príklad týchto informácií je na obrázku 8.4.



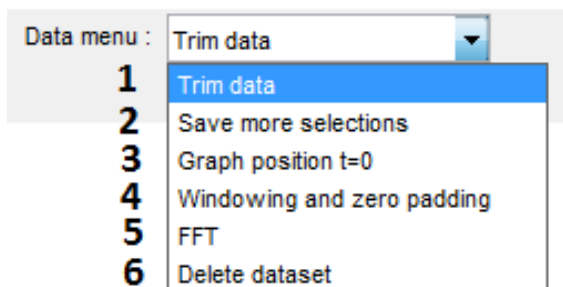
|    | 1                      | 2                                       |
|----|------------------------|---|
| 1  | name                   | Force_(IEPE)                            |
| 2  | format_string          |   |
| 3  | iteration_based_timing | <input type="checkbox"/>                |
| 4  | unit_string            | Newtons                                 |
| 5  | unit_GUID              | {4AB0451E-DD6E-4FF1-940E-E09CD0E8F7...} |
| 6  | X_unit_string          |   |
| 7  | X_unit_GUID            | {0748B676-82D4-4318-A843-708972E53760}  |
| 8  | unit_version           | 30000                                   |
| 9  | Title                  | 21.11.2017 16:02:08                     |
| 10 | Prefix                 | 21.11.2017 16:02:08                     |
| 11 | Author                 | Ondrej                                  |
| 12 | Description            | 5cm                                     |
| 13 | log_datatype           | Time-WDT-Real                           |
| 14 | FileFormatVersion      | 2.1                                     |
| 15 | WriterVersion          | 15.0                                    |
| 16 | WriterName             | SignalExpress 2015                      |
| 17 | DateTime               | 21-Nov-2017 10:02:09:000                |
| 18 | TimeZone               | Střední Evropa (běžný čas)              |
| 19 | timing_mode            | HWTimed_Continuous                      |
| 20 | IntervalCount          | 1                                       |
| 21 | data_ready_for_viewing | <input checked="" type="checkbox"/>     |
| 22 | DataFormat             | SingleWaveform                          |
| 23 | log_duration           | 6.713046550750732                       |
| 24 | log_dt                 | 3.906249999999998e-05                   |
| 25 | log_iteration_length   | 1.0000000000000000                      |
| 26 | recording_complete     | <input checked="" type="checkbox"/>     |

Obr. 8.4: Okno s informáciami z TDMS súboru

3. Priblíženie vykreslených dát.
4. Vzdialenie vykreslených dát.
5. Pohyb v okne vykreslených dát.

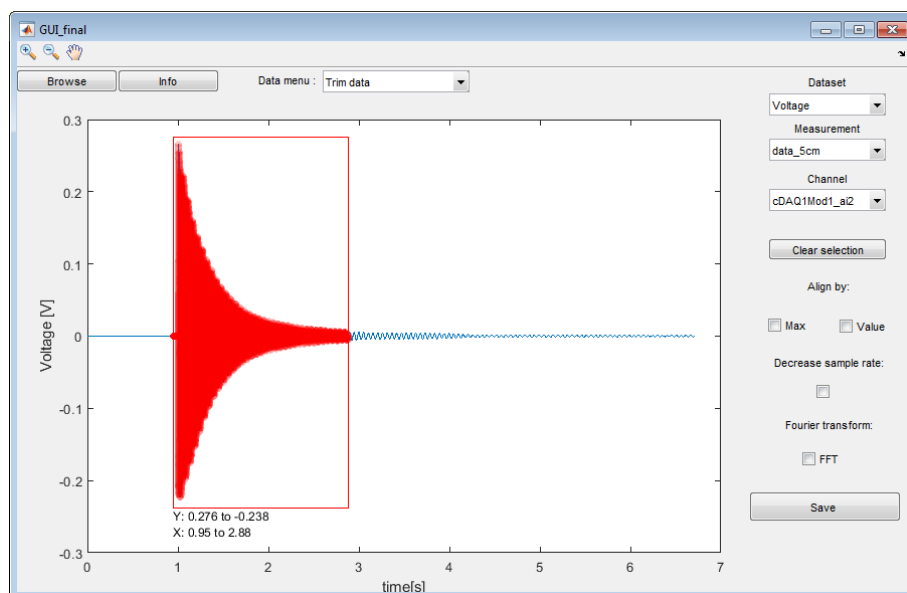
### 8.1.2. Menu s funkciami pre vybratý kanál

V tejto oblasti sa nachádzajú funkcie pre prácu s dátami len v prípade, že máme vybrané meranie a kanál. Popis funkcií zobrazených na obrázku 8.5 je rozpísaný nižšie.



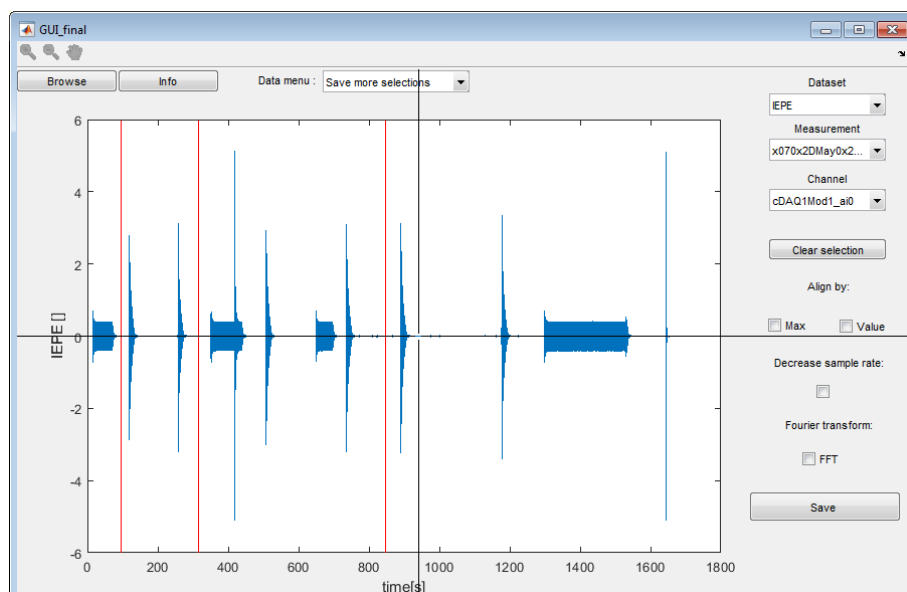
Obr. 8.5: Rozloženie v druhej oblasti

1. Orezanie dát pri výbere jedného merania a kanála. Orezané dáta sa uložia do pracovného priestoru softvéru Matlab a vykreslia v grafickom rozhraní. Po výbere je nutné stlačiť klávesu ENTER. Názorná ukážka je na obrázku 8.6.



Obr. 8.6: Ukážka orezania dát

2. Orezanie viacerých sekcií na vybratom kanáli. Sekcie vyberáme postupným klikaním kurzora na vykreslených dátach, ktoré sa označia vertikálnou červenou čiarou. Tento postup znázorňuje obrázok 8.7. Vybrané sekcie uložíme do štruktúry pomocou klávesy ENTER.



Obr. 8.7: Orežanie viacerých sekcií

3. Úprava časovej osi orezaných dát so začiatkom v 0 sekúnd.
4. Aplikovanie okna a rozšírenie o nuly pri vybranom meraní a kanáli. Počet núl je možné nastaviť v okne po vybraní tejto funkcie. Okno je vytvorené pomocou funkcie hanning. <sup>[13]</sup>
5. Vytvorenie Fourierovej transformácie pre vybrané meranie a kanál. Transformácia sa zobrazí v novom grafickom okne.
6. Zmazanie vybraného merania so všetkými kanálmi. Pri výbere merania nie je nutné vybrať aj kanál. Priečinko s meraním sa nezmaže.

### 8.1.3. Výber dát

V tejto časti grafického rozhrania vyberáme dáta, s ktorými budeme pracovať. Rozloženie výberu je zobrazené na obrázku 8.8 s popisom jednotlivých polí pod obrázkom.

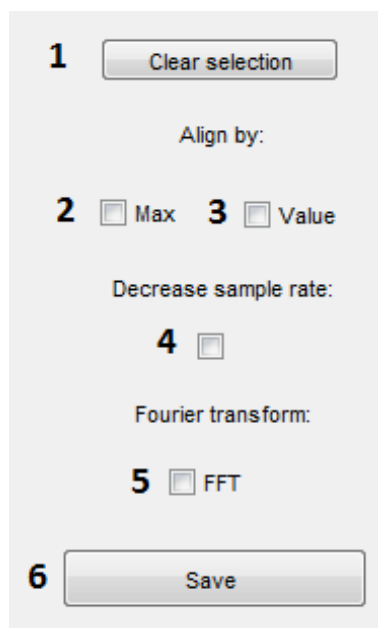
Obr. 8.8: Rozloženie v tretej oblasti

1. Výber typu merania. Vždy sa vyberajú neprevzorkované dáta. Pokiaľ má vybraný typ merania viacero kanálov, zobrazí sa okno s výberom kanálu. Vykreslia sa všetky merania tohoto typu.

2. Výber merania. Zobrazia sa kanály v nasledujúcom okne.
3. Výber kanála. Vybraný kanál sa vykreslí v grafickom priestore.

#### 8.1.4. Funkcie a export viacerých meraní

Táto oblasť slúži na export všetkých načítaných dát do prehľadnej tabuľky. Pred exportom je možné dáta upraviť alebo pridať ku nim ich Fourierovu transformáciu. Rozloženie týchto úprav je na obrázku 8.9 s popisom pod ním.



Obr. 8.9: Rozloženie v štvrtej oblasti

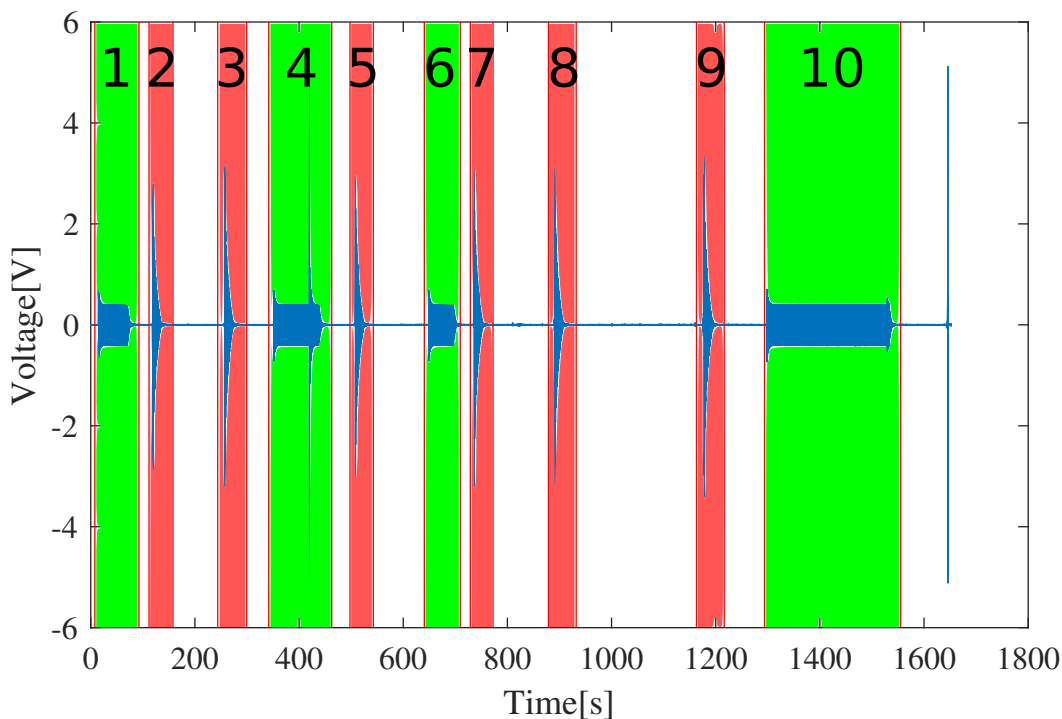
1. Zmazanie výberu pre zarovnanie, prevzorkovanie a Fourierovu transformáciu.
2. Zarovnanie podľa maximálnej hodnoty s výberom orezania pred a po maximálnej hodnote. Zarovnáva sa podľa vybraného typu merania. Po zarovnaní sa merania zobrazia v grafickej oblasti.
3. Zarovnanie podľa prvej dosiahnutej hodnoty s výberom orezania pred a po tejto hodnote. Zarovnáva sa podľa vybraného typu merania. Po zarovnaní sa merania zobrazia v grafickej oblasti.
4. Prevzorkovanie dát pomocou funkcie *decimate*. Faktor prevzorkovania je možné vybrať v okne po zvolení tejto možnosti.
5. Vytvorenie Fourierovej transformácie. Po vybraní tejto možnosti sa zobrazí okno s výberom typov meraní, na ktorých sa má transformácia vykonať, a s možnosťou výberu frekvenčného maxima transformácie.
6. Uloženie dát s úpravami, alebo bez nich, do prehľadnej tabuľky.

## 9. Analýza experimentálnych dát

Overenie funkčnosti grafického rozhrania budeme testovať na dátach, ktoré pochádzajú z experimentálnych meraní. Dáta sú namerané na nosníku, ktorý sme náhodne budili frekvenčným generátorom a rázovým kladivkom. Tieto budenia budeme nazývať deje. Dáta z experimentu sú zobrazené na obrázku 9.1, na ktorom sú označené úseky týchto dejov. Pomocou grafického rozhrania tieto dáta rozrežeme na časti podľa druhu deja, ktoré uložíme do pracovného priestoru softvéru Matlab. Z uložených dát budeme určovať ich dynamické charakteristiky.

Pre lepšiu predstavu výskytu takýchto dejov v priemysle, môžeme dáta z experimentu prirovnať k dátam z monitorovania obrábacieho stroja počas jeho prevádzky. Prvý z dejov je harmonické budenie pomocou frekvenčného generátora, ktoré môže na obrábacom stroji znázorňovať rotačný pohyb motora vretena, alebo motora pracovnej plochy stola. Pri tomto deji môžeme určovať frekvenciu otáčania motora, z ktorej monitorujeme napríklad jeho rýchlosť. Ďalším prípadom by mohli byť vibrácie stroja, ktoré vznikajú pri rotačnom pohybe motora, pri ktorých môžeme určiť ich frekvenciu. Druhý dej je odozva na impulz z rázového kladivka, ktorá môže znázorňovať náraz do stola počas prevádzky. Pri takomto deji pozorujeme tlmenie, ktoré určuje, ako rýchlo sa dokáže rozvibrovanie stola po náraze ustáliť.

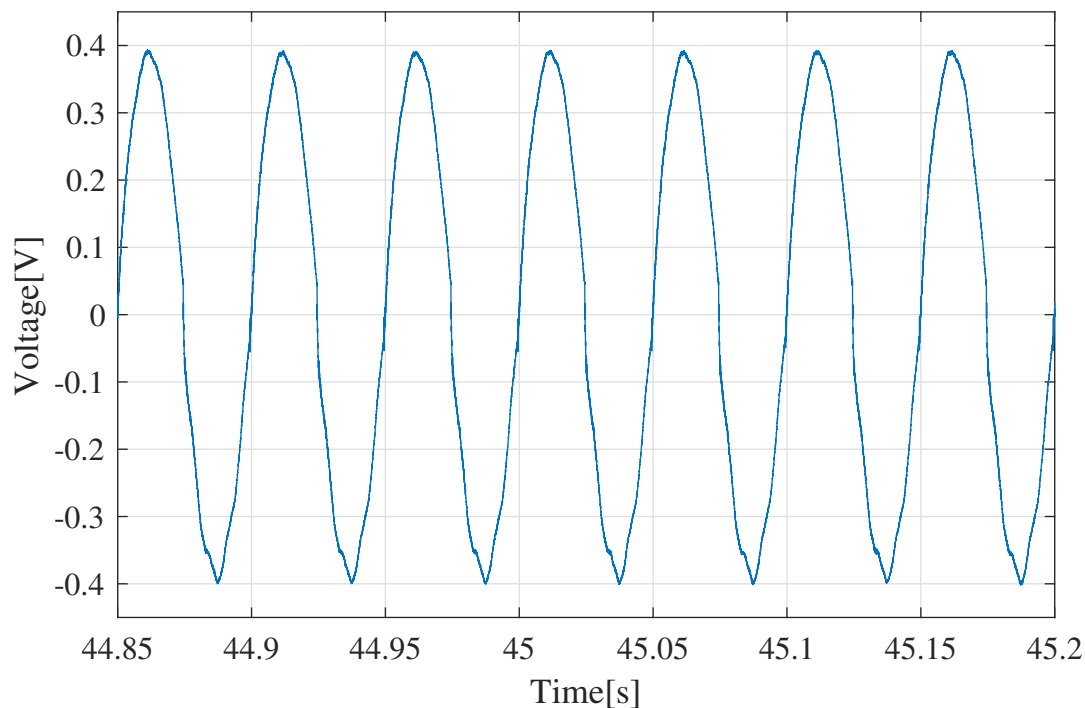
Analýzou týchto signálov môžeme dostať lepšiu predstavu o chovaní stroja, a v prípade neželaných vibrácií ich eliminovať alebo rýchlejšie utlmiť. Pre prvý dej určíme budiacu frekvenciu generátora pomocou transformácie do frekvenčnej oblasti, a pre druhý dej veľkosť tlmenia z rovnice tlmiacej krivky. Tieto hodnoty porovnáme z predchádzajúcimi meraniami.



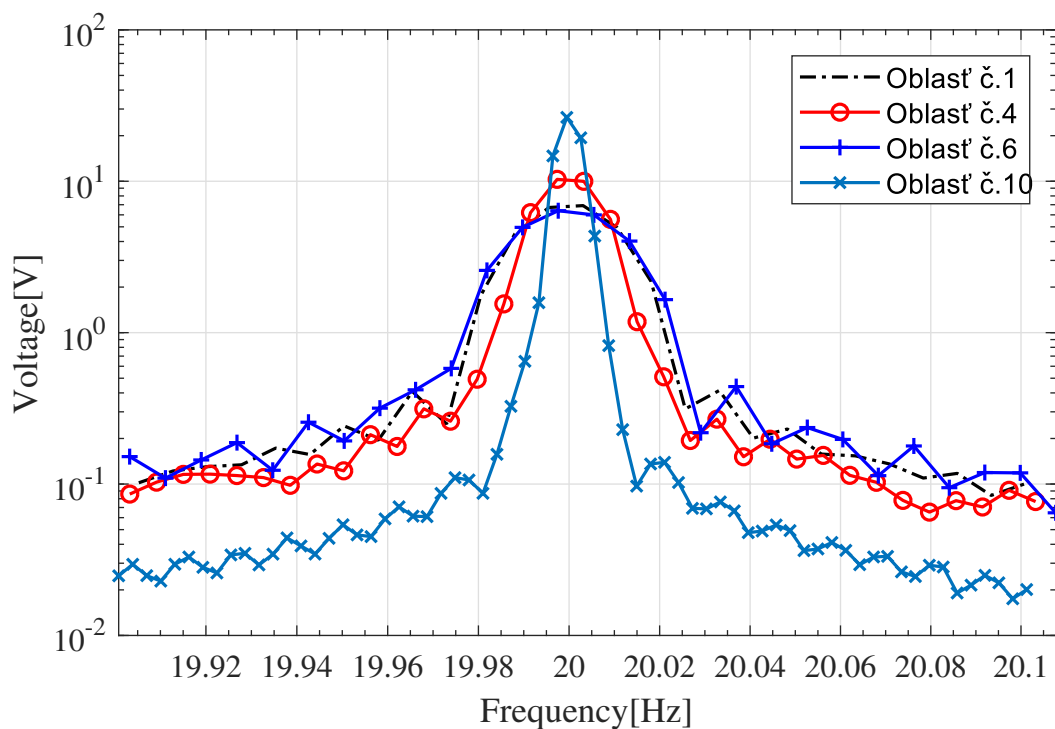
Obr. 9.1: Výber dát na analýzu

## 9.1. Overenie budiacej frekvencie

Oblasti, ktoré sú na obrázku 9.1 ohraničené s číslom 1, 4, 6 a 10 sú harmonicky budené frekvenčným generátorom. Generátor sme nastavili na frekvenciu 20 Hz, pričom sa budeme snažiť túto frekvenciu overiť pomocou analyzovania signálu. Ukážka priblíženej prvej oblasti je zobrazená na obrázku 9.2. Tieto oblasti analyzujeme pomocou transformácie do frekvenčnej oblasti, ktorej výsledky sú zobrazené na obrázku 9.3. Použili sme logaritmickú stupnicu pre detailnejšie zobrazenie. Na každej z týchto oblastí sme potvrdili budiacu frekvenciu generátora, ktorá predstavuje vrchol kriviek na tomto obrázku.



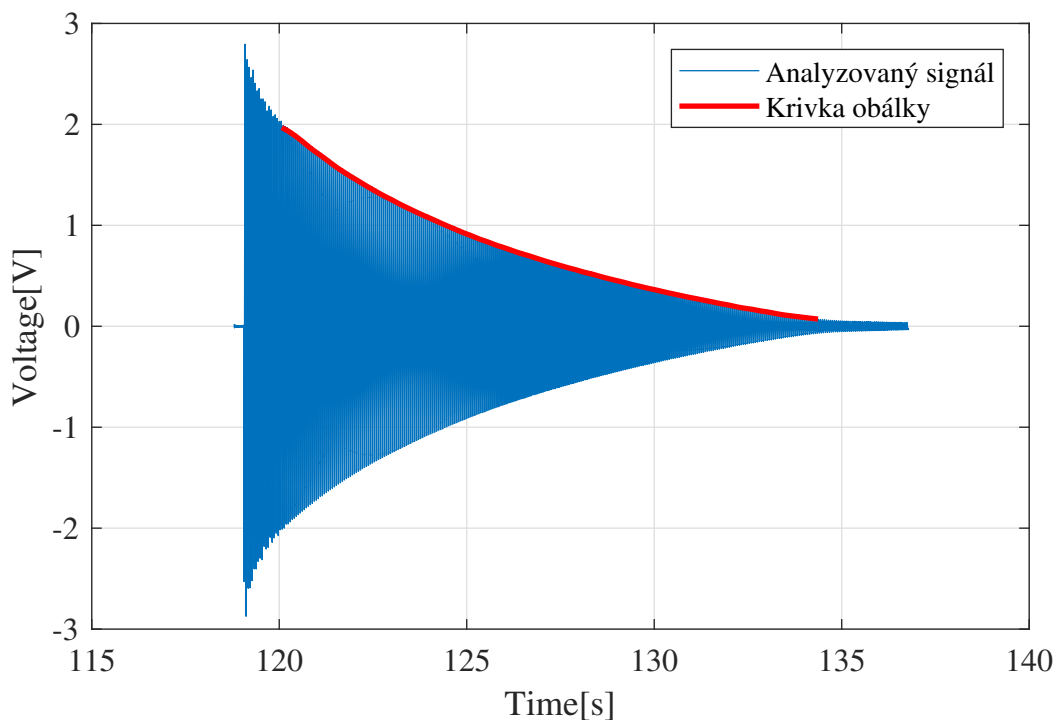
Obr. 9.2: Ukážka priblíženej prvej oblasti z obrázku 9.1



Obr. 9.3: Časť frekvenčnej oblasti analyzovaného signálu

## 9.2. Výpočet tlmenia

Ostatné oblasti sú budené rázovým kladivkom, na ktorých môžeme pozorovať tlmenie. To určíme buď pomocou logaritmického dekrementu, ktoré je menej presné a náročnejšie na výpočet, alebo pomocou rovnice exponenciálnej krivky tlmenia, ktorú získame po preložení vrcholov tlmeného signálu. V našom prípade budeme pre výpočet tlmenia používať rovnicu exponenciálnej krivky tlmenia. Pred preložením vrcholov krivkou sme dali dáta do absolútnej hodnoty, a vybrali časť signálu po ustálení, kúsok za jeho maximálnou hodnotou. Táto krivka s analyzovaným signálom je zobrazená na obrázku 9.4.



Obr. 9.4: Krivka obálky analyzovaného signálu

Rovnica tejto krivky je vyjadrená vzťahom <sup>[10]</sup>

$$f(t) = A \cdot e^{-\zeta \cdot \omega_0 \cdot t}$$

Pre určenie rovnice krivky na našich analyzovaných dátach používame v Matlabe toolbox cftool, ktorý hľadá parametre na rovnici exponenciálnej krivky

$$f(t) = A \cdot e^{-B \cdot t}$$

Parameter  $B$  reprezentuje súčiniteľ doznievania  $\delta$ , ktorý môžeme vyjadriť ako

$$B = \delta = \zeta \cdot \omega_0$$

z ktorého vyjadríme tlmenie  $\zeta$  ako

$$\zeta = \frac{\delta}{\omega_0}$$

kde  $\omega_0$  je netlmená vlastná frekvencia. V našom prípade túto frekvenciu získame pomocou Fourierovej transformácie, pričom je nutné previesť jednotky z  $Hz$  na  $rad \cdot s^{-1}$  pomocou vzťahu

$$\omega_0 = 2 \cdot \pi \cdot f_0$$

Výsledný vzorec na výpočet tlmenia potom bude

$$\zeta = \frac{\delta}{2 \cdot \pi \cdot f_0}$$

Konštanty a vypočítané tlmenie pre každú oblasť sú uvedené v nasledujúcej tabuľke.



| Výpočet tlmenia |                            |           |                      |
|-----------------|----------------------------|-----------|----------------------|
| Oblasť          | $\delta[rad \cdot s^{-1}]$ | $f_0[Hz]$ | $\zeta[-]$           |
| 2               | 0.1668                     | 18.71     | $1.42 \cdot 10^{-3}$ |
| 3               | 0.1644                     | 18.71     | $1.40 \cdot 10^{-3}$ |
| 4               | 0.1602                     | 18.51     | $1.38 \cdot 10^{-3}$ |
| 5               | 0.1615                     | 18.69     | $1.38 \cdot 10^{-3}$ |
| 7               | 0.1621                     | 18.70     | $1.38 \cdot 10^{-3}$ |
| 8               | 0.1648                     | 18.71     | $1.40 \cdot 10^{-3}$ |
| 9               | 0.1648                     | 18.69     | $1.40 \cdot 10^{-3}$ |

Na základe predchádzajúcich meraní na nosníku vychádza toto tlmenie v tisícinách. K týmto hodnotám sa nám podarilo priblížiť, preto toto meranie môžeme brať za správne. Priemerná hodnota vypočítaného tlmenia nám vyšla  $1.4 \cdot 10^{-3}$ .

## 10. Možnosti budúceho vývoja grafického rozhrania

Grafické rozhranie sme testovali na pár meraniach, pre ktoré fungovalo podľa očakávaní. Pri načítaní dát z iných meraní sme narazili na problém s názvami jednotlivých meraní, kde sa nachádzali znaky, ktoré nemôžu obsahovať názvy premenných v softvéri Matlab. Tento problém by bolo vhodné odstrániť. Taktiež by bolo vhodné pre väčšie dáta pridať možnosť načítania dát pomocou funkcie *datastore*. My sme túto funkciu nepotrebovali, z dôvodu veľkosti dát, ktoré sa zmestili do pamäte. Ďalej by sa mohla pridať možnosť načítania iných formátov pre flexibilnejšie využitie grafického rozhrania.

# 11. Záver

Táto bakalárska práca sa zaoberala riešením problému spracovania Veľkých dát, ktoré vznikajú pri monitorovaní moderných technických zariadení. Na začiatku sú zhrnuté základné vlastnosti Veľkých dát, ich výskyt v priemysle a metódy pre ich spracovanie na počítačovom clustri a v samotných programovacích jazykoch. Predpokladáme, že Veľkých dát bude stále pribúdať, a tento problém bude musieť byť riešený novými metódami, alebo výkonnejším hardvérom, pokiaľ budeme potrebovať výsledky v dostupnejšom čase.

Ďalšia časť práce bola venovaná súborovému formátu TDMS. Tento komplexný formát, ktorý slúži pre prehľadnú úschovu a prezeranie dát zo senzorov, získavame z meracích kariet a softvéru Signal Express od spoločnosti NI. Existuje softvér od tejto spoločnosti pre prácu s týmto formátom, avšak pre väčšiu slobodu pri výbere a následný preprocessing boli uvedené iné dostupné metódy pre jeho otvorenie v rôznych softvéroch a programovacích jazykoch. Na týchto metódach boli porovnané rýchlosti načítania v závislosti na veľkosti súboru v jazykoch Matlab a Python. Výsledky dopadli podľa očakávania, pričom programovací jazyk Python si s načítaním a uložením dát do textového dokumentu poradil pri najväčších súboroch približne o 25 percent rýchlejšie. Ďalej sú popísané možnosti preprocessingu dát po ich otvorení, ako napríklad zmenšenie veľkosti, filtrovanie, transformovanie a konverzia dát do iných formátov. Tieto možnosti preprocessingu uľahčujú prácu s dátami, nakoľko sa snažia zachovať ich pôvodnú informačnú hodnotu.

V poslednej časti boli implementované poznatky o Veľkých dátach, súborovom formáte TDMS a preprocessingu dát do grafického rozhrania vytvoreného v programovacom prostredí Matlab, ktoré spracováva dáta z viacerých meraní a výrazne ušetrí čas, pretože nie je potrebné spracovávať tieto merania samostatne. Výstupom z grafického rozhrania je prehľadná tabuľka dát zo všetkých meraní, na ktorých môže byť vykonaný preprocessing podľa preferencie používateľa a možnosť doplnenia tabuľky o transformáciu do frekvenčnej oblasti. Pre nedostupnosť počítačového clustra sa nám nepodarilo implementovať metódy pre prácu s dátami na clustri do grafického rozhrania, a tak porovnať s konvenčnými. Vytvorené grafické prostredie zefektívnilo prácu s formátom TDMS a tým bol cieľ práce naplnený. Autor práce sa zamyslel aj nad možnosťami možného vylepšenia a vývoja vytvoreného grafického prostredia. Tieto možnosti môžu byť predmetom nasledujúcich prác, ktoré ich môžu vyriešiť. Súčasnú riešenie predstavuje základ, ktorý je možné ďalej rozširovať.

Pre testovanie funkčnosti tohoto rozhrania boli orezané časti záujmu na dlhšom meraní, na ktorých sme určovali dynamické vlastnosti. Toto meranie bolo budené frekvenčným generátorom na frekvenciu 20 Hz a rázovým kladivkom v rôznych momentoch, pre ktoré sme sa snažili získať ich dynamické vlastnosti. Pre všetky úseky budené frekvenčným generátorom sa nám zo signálu podarilo získať jeho budiacu frekvenciu. Na ostatných úsekoch sme pozorovali tlmenie, ktorého hodnoty sa približujú k tým z predchádzajúcich meraní. Hodnota tlmenia vyšla približne  $\zeta = 1.4 \cdot 10^{-3}$ .

# Literatúra

- [1] Cady, F.: Big Data. In *The Data Science Handbook*, Hoboken, NJ, USA: John Wiley and Sons, Inc., 2017, ISBN 9781119092940, s. 185–201.
- [2] Creations, N.: Arduino Tutorial: Simple High-pass, Band-pass and Band-stop Filtering. <https://www.norwegiancreations.com/2016/03/arduino-tutorial-simple-high-pass-band-pass-and-band-stop-filtering/>, Mar 2016, [Online; accessed 18-May-2019].
- [3] Demchenko, Y.; Grosso, P.; de Laat, C.; aj.: Addressing big data issues in Scientific Data Infrastructure. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, May 2013, s. 48–55, doi:10.1109/CTS.2013.6567203.
- [4] Developers, N.: NumPy. <https://www.numpy.org/>, [Online; accessed 20-April-2019].
- [5] Hokanson, J.: TDMS Reader. <https://www.mathworks.com/matlabcentral/fileexchange/30023-tdms-reader>, 2019, MATLAB Central File Exchange. Retrieved May 22, 2019.
- [6] Ingle, V. K.: *Digital signal processing using MATLAB*. Pacific Grove: Brooks/Cole, 2000, ISBN 0-534-37174-4.
- [7] Instruments, N.: The NI TDMS File Format. <https://www.ni.com/product-documentation/3727/en/>, [Online; accessed 8-April-2019].
- [8] Instruments, N.: TDM Excel Add-In for Microsoft Excel Download. <http://www.ni.com/example/27944/en/>, [Online; accessed 19-May-2019].
- [9] Instruments, N.: Understanding FFTs and Windowing. <http://www.ni.com/cs-cz/innovations/white-papers/06/understanding-ffts-and-windowing.html>, [Online; accessed 10-May-2019].
- [10] Kšica, F.: Modelování dynamických vlastností a chování technických soustav. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 113 s. Vedoucí diplomové práce doc. Ing. Zdeněk Hadaš, Ph.D.
- [11] LlamaZOO: What is a Digital Twin? (Plus 3 industries which already benefit). <https://www.llamazoo.com/what-is-a-digital-twin/>, Jul 2018, [Online; accessed 15-May-2019].
- [12] Mathworks: decimate. <https://www.mathworks.com/help/signal/ref/decimate.html>, [Online; accessed 25-April-2019].
- [13] Mathworks: Hann (Hanning) window. <https://www.mathworks.com/help/signal/ref/hann.html>, [Online; accessed 8-April-2019].
- [14] O’Leary, D. E.: Artificial intelligence and big data. *IEEE Intelligent Systems*, ročník 28, č. 2, 2013: s. 96–99.

- [15] Qi, Q.; Tao, F.: Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *Access, IEEE*, ročník 6, 2018: s. 3585–3593, ISSN 2169-3536.
- [16] Reeve, A.: Welcome to npTDMS's documentation. <https://nptdms.readthedocs.io/en/latest/>.
- [17] Wang, J.; Zhang, W.; Shi, Y.; aj.: Industrial Big Data Analytics: Challenges, Methodologies, and Applications. *arXiv preprint arXiv:1807.01016*, 2018.
- [18] Wikipedia contributors: Digital signal (signal processing) — Wikipedia, The Free Encyclopedia. [https://en.wikipedia.org/w/index.php?title=Digital\\_signal\\_\(signal\\_processing\)&oldid=876690105](https://en.wikipedia.org/w/index.php?title=Digital_signal_(signal_processing)&oldid=876690105), 2019, [Online; accessed 15-May-2019].
- [19] Willingham, D.: Big Data Analysis and Analytics with MATLAB. In *Proceedings, 15th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 2015): Melbourne, Australia, October 17-23, 2015*, 2015, str. WED3O05, doi:10.18429/JACoW-ICALEPCS2015-WED3O05.

# Zoznam obrázkov

|     |   |    |
|-----|---|----|
| 2.1 | Diagram vlastností Veľkých dát <sup>[3]</sup> . . . . .                     | 4  |
| 3.1 | Časti, ktoré sú charakteristické pre Industry 4.0 <sup>[11]</sup> . . . . . | 6  |
| 3.2 | Spracovanie veľkých dát <sup>[15]</sup> . . . . .                           | 8  |
| 3.3 | Vzorkovanie digitálneho signálu <sup>[18]</sup> . . . . .                   | 9  |
| 6.1 | Štruktúra formátu TDMS <sup>[7]</sup> . . . . .                             | 13 |
| 6.2 | Porovnanie rýchlosti otvorenia súboru TDMS v jazyku Matlab a Python .       | 15 |
| 7.1 | Nezarovnané dáta z viacerých meraní . . . . .                               | 18 |
| 7.2 | Zarovnané dáta z viacerých meraní . . . . .                                 | 18 |
| 7.3 | Znázornenie dát po zmene vzorkovacej frekvencie . . . . .                   | 19 |
| 7.4 | Základné typy filtrov <sup>[2]</sup> . . . . .                              | 20 |
| 7.5 | Signál v časovej a frekvenčnej oblasti . . . . .                            | 21 |
| 7.6 | Zero padding . . . . .  | 22 |
| 7.7 | Porovnanie transformácie do frekvenčnej oblasti bez a s pridaním núl . . .  | 22 |
| 7.8 | Windowing . . . . .   | 23 |
| 8.1 | Diagram algoritmu grafického rozhrania . . . . .                            | 25 |
| 8.2 | Oblasti v grafickom rozhraní . . . . .                                      | 26 |
| 8.3 | Rozloženie v prvej oblasti . . . . .  | 27 |
| 8.4 | Okno s informáciami z TDMS súboru . . . . .                                 | 27 |
| 8.5 | Rozloženie v druhej oblasti . . . . .                                       | 28 |
| 8.6 | Ukážka orezania dát . . . . .   | 28 |
| 8.7 | Orezanie viacerých sekcií . . . . .   | 29 |
| 8.8 | Rozloženie v tretej oblasti . . . . .                                       | 29 |
| 8.9 | Rozloženie v štvrtej oblasti . . . . .                                      | 30 |
| 9.1 | Výber dát na analýzu . . . . .  | 31 |
| 9.2 | Ukážka priblíženej prvej oblasti z obrázku 9.1 . . . . .                    | 32 |
| 9.3 | Časť frekvenčnej oblasti analyzovaného signálu . . . . .                    | 33 |
| 9.4 | Krivka obálky analyzovaného signálu . . . . .                               | 34 |

## 12. Prílohy

Priložené CD obsahuje prílohy podľa nasledujúcej štruktúry:

CD-ZIP

- GUI\_final.m -- kód funkcií vytvoreného grafického prostredia
- GUI\_final.fig -- kód rozloženia vytvoreného grafického prostredia
- TDMS\_speedtest.m -- kód v jazyku Matlab pre porovnanie rýchlosti otvorenia
- TDMS\_speedtest.py -- kód v jazyku Python pre porovnanie rýchlosti otvorenia
- 2019\_BP\_Duris\_Anton\_184197.pdf -- elektronická verzia bakalárskej práce