TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ŠTATISTICKÉ VÝPOČTY V MATLABE

Diplomová práca

2020 **Adam Fehér, Bc.**

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ŠTATISTICKÉ VÝPOČTY V MATLABE

Diplomová práca

Študijný program:	Priemyselná elektrotechnika
Študijný odbor:	Elektrotechnika
Školiace pracovisko:	Katedra teoretickej a priemyselnej elektrotechniky
Školiteľ:	doc. Ing. Milan Guzan, PhD.

2020 Košice **Adam Fehér, Bc.**

Abstrakt v SJ

Táto diplomová práca sa zaoberá štatistickou analýzou údajov meraných v čase. Pre účely štatistickej analýzy bol použitý program MATLAB, v ktorom bol vypracovaný plne funkčný projekt. Práca zároveň poukazuje na využitie programu MATLAB v oblasti údajovej vedy. Tento projekt je zložený z viacerých funkcii a aplikácii, ktoré umožňujú používateľovi efektívne prehliadať údaje z rôznych uhlov pohľadu, čím napomáha dosiahnuť používateľovi požadované výsledky.

Kľúčové slova v SJ

MATLAB, štatistika, údaje, údajová veda, meranie, analýza, elektrické napätie, teplota, vlhkosť, časový priebeh, aplikácia.

Abstrakt v AJ

This diploma thesis deals with the statistical analysis of data measured over time. For the purposes of statistical analysis, the MATLAB program was used, in which a fully functional project was developed. The work also points to the use of MATLAB in the field of data science. This project consists of several functions and applications that allow the user to effectively view the data from different angles, thus helping the user to achieve the desired results.

Kľúčové slova v AJ

MATLAB, statistic, data, data science, measurement, analysis, voltage, temperature, humidity, time series, application.

56952

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra teoretickej a priemyselnej elektrotechniky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Elektrotechnika**

Študijný program: **Priemyselná elektrotechnika**

Názov práce:

Štatistické výpočty v Matlabe

Statistical calculations in Matlab

Študent: **Bc. Adam Fehér**

Školiteľ: **doc. Ing. Milan Guzan, PhD.**

Školiace pracovisko: **Katedra teoretickej a priemyselnej elektrotechniky**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Oboznámiť sa s možnosťami Matlabu v oblasti štatistiky.
2. Vytvoriť a opísať postup pri vytváraní aplikácie na výpočet a zobrazenie štatistických parametrov pre merané sieťové napätie.
3. Výsledky prehľadne dokumetovať.
4. Vzájomne porovnať merania v jednotlivých dňoch.
5. Ilustrovať možnosti Matlabu aj pre prácu s inými nameranými údajmi.
6. Zdrojové kódy dôsledne komentovať.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 04.05.2020

Dátum zadania diplomovej práce: 31.10.2019

12.

.....

prof. Ing. Liberios Vokorokos, PhD.

dekan fakulty

TECHNICKÁ UNIVERZITA V KOŠICIACH FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval/a samostatne s použitím uvedenej odbornej literatúry.

Košice, 06. mája 2020 .....

vlastnoručný podpis

Poďakovanie

Chcel by som sa touto formou poďakovať svojmu vedúcemu práce doc. Ing. Milanovi Guzanovi, PhD. za jeho čas, pripomienky a odbornú pomoc pri riešení tejto záverečnej práce. Poďakovanie taktiež patrí aj mojej rodine a mojím blízkym za ich neustálu podporu a trpezlivosť.

Obsah

Zoznam obrázkov 9

Zoznam tabuliek 11

Zoznam symbolov a skratiek 12

Úvod 13

1. Vývojové prostredie MATLAB 14

1.1. Pracovná plocha 14

1.2. Údajové typy 16

1.3. Skripty 18

1.4. Funkcie 18

1.5. Aplikácie 20

2. Údajová veda 25

2.1. Zber údajov 25

2.2. ETL proces 26

2.3. Analýza údajov 27

2.4. Vizualizácia údajov 28

2.5. Postrehy 28

3. Organizácia projektu v MATLABe 29

3.1. Hardvérové a softvérové parametre zariadenia 29

3.2. Organizácia priečinkov projektu 30

3.3. Súčasný stav riešenia 30

4. Návrh riešenia pomocou jazyka MATLAB 32

4.1. Zber údajov 32

4.2. ETL proces 33

4.2.1. Funkcia *create_values_file* 33

4.2.2. Funkcia *create_values* 35

4.2.3. Funkcia *create_missing* 35

4.2.4. Funkcia *create_filled* 35

4.2.5. Funkcia *create_statistics* 37
4.2.6. Funkcie *etl_update* a *etl_append* 38
4.3. Analýza a vizualizácia údajov 40
4.3.1. Grafy štatistických parametrov 40
4.3.2. Štatistická analýza signálov 44
4.3.3. Signály meranej veličiny 47
4.3.4. Korelačné diagramy štatistických parametrov 48
4.3.5. Korelačná matica ľubovoľného štatistického parametra 49
4.3.6. Dvojrozmerné mapy výskytu hodnôt 50
4.3.7. Grafy rýchlej Fourierovej transformácie (FFT) signálov 51
4.4. Postrehy pri práci s aplikáciami 52
4.4.1. Meranie efektívnej hodnoty sieťového napätia 52
4.4.2. Meranie teploty a relatívnej vlhkosti 59
4.4.3. Meranie odporu meracieho kábla a napätia 61
Záver 62
Zoznam použitej literatúry 63
Prílohy 66

Zoznam obrázkov

Obr. 1 Náhľad programu MATLAB pri prvotnom spustení. 15
Obr. 2 Úvodné okno editora *App Designer* 21
Obr. 3 Okno úpravy dizajnu aplikácie editora *App Designer* 22
Obr. 4 Okno úpravy kódu aplikácie editora *App Designer* 23
Obr. 5 Ukážka editovateľnej časti kódu aplikácie 24
Obr. 6 Časová os procesov údajovej vedy 26
Obr. 7 Znázornenie vnútorných podprocesov ETL procesu 27
Obr. 8 Náhľad prvých 10 riadkov náhodného údajového súboru merania sieťového napätia 32
Obr. 9 Časová os fiktívneho údajového súboru 34
Obr. 10 Princíp získavania nových numerických hodnôt pomocou pohyblivého priebehu 36
Obr. 11 Okno aplikácie *ETL_Configuration* 39
Obr. 12 Okno spusteného procesu v aplikácii *ETL_Configuration* 40
Obr. 13 Ukážka prvej záložky aplikácie *Descriptive_Statistics* 41
Obr. 14 Pravá polovica ovládacieho panela 41
Obr. 15 Ľavá polovica ovládacieho panela 41
Obr. 16 Ukážka druhej záložky aplikácie *Descriptive_Statistics* 44
Obr. 17 Ukážka okna aplikácie *Statistics_Analysis* 45
Obr. 18 Ilustrácia závislosti hodnoty koeficientu Z od požadovanej percentuálnej zložky hodnôt. 46
Obr. 19 Ukážka okna aplikácie *Signals* 47
Obr. 20 Ukážka troch korelačných diagramov rozsahu hodnôt a smerodajnej odchýlky 48
Obr. 21 Ukážka korelačnej matice rozsahu hodnôt všetkých utorkov mesiaca október v roku 2019 49
Obr. 22 Ukážka troch 2D máp hodnôt sieťového napätia 50
Obr. 23 Ukážka 30-minútového signálu nameraných hodnôt sieťového napätia a jeho rýchlej Fourierovej transformácii 51
Obr. 24 Časový priebeh všetkých „čistých“ údajov sieťového napätia 53
Obr. 25 Analýza 10-minútových vzoriek dňa 1. 9. 2019 54
Obr. 26 Štatistická analýza vzorky dňa 13.11.2019 v čase od 19:10 do 19:20 55
Obr. 27 Štatistická analýza vzorky dňa 25.2.2019 v čase od 7:30 do 7:40 55
Obr. 28 Graf porovnania skutočnej (body grafu) a teoretickej (referenčná diagonála) percentuálnej zložky odhadovaných intervalov spoľahlivosti 56
Obr. 29 Časový priebeh signálu sieťového napätia dňa 1.1.2019 v čase od 15:00 do 16:00 57
Obr. 30 Časový priebeh signálu sieťového napätia dňa 16.11.2019 v čase od 8:00 do 9:00 57

Obr. 31 Časový priebeh signálu sieťového napätia dňa 3.5.2019 v čase od 12:00 do 13:00 57
Obr. 32 Časový priebeh signálu sieťového napätia dňa 29.10.2019 v čase od 21:00 do 22:00 57
Obr. 33 Korelačný diagram koeficientu špicatosti a medzikvartilového rozptylu hodnôt 10-minútových vzoriek dňa 23.2.2019 a 24.2.2019 58
Obr. 34 Korelačná matica signálov smerodajných odchýlok 60-minútových vzoriek všetkých sobôt v mesiaci november v roku 2019 58
Obr. 35 Dvojrozmerné mapy výskytu hodnôt 60-minútových vzoriek dňa 25.12.2019 59
Obr. 36 Vzor časových priebehov (hore) a prvej derivácie časových vektorov (dole) teploty (vľavo) a relatívnej vlhkosti vzduchu (vpravo) 60
Obr. 37 Ilustrácia údajov tabuľky s názvom *e* 61
Obr. 38 Ilustrácia údajov tabuľky s názvom *s* 61

Zoznam tabuliek

Tab. 1 Čiastočná špecifikácia zariadenia 29

Zoznam symbolov a skratiek

ans skrátka pre answer (odpoveď)

CDF skrátka pre Cumulative Distribution Function (kumulatívna distribučná funkcia)

CSV skrátka pre Comma Separated Values (čiarkou oddelené hodnoty)

DAQ skrátka pre Data Acquisition (zber údajov)

EDF skrátka pre European Data Format (Európsky formát údajov)

ETL skrátka pre Extract, Transform, Load (proces získavania, transformácie a načítania údajov)

GUI skrátka pre Graphical User Interface (grafické používateľské rozhranie)

IQR skrátka pre Interquartile Range (medzikvartilný rozsah)

JSON skrátka pre JavaScript Object Notation (objektový zápis jazyka JavaScript)

PDF skrátka pre Probability Density Function (funkcia hustoty pravdepodobnosti)

RDS skrátka pre Relative Standard Deviation (relatívna smerodajná odchýlka)

SD skrátka pre Standard Deviation (smerodajná odchýlka)

SEM skrátka pre Standard Error of the Mean (štandardná chyba aritmetického priemeru)

Úvod

V prvej kapitole tejto diplomovej práce je popísané vývojové prostredie programu MATLAB. Najprv sa čitateľ oboznámi s pracovnou plochou tohto programu, následne s údajovými typmi využívanými v tejto práci. V závere tejto kapitoly sa čitateľ oboznámi s tromi možnými spôsobmi uchovávania kódu zapísaného v programe MATLAB a zmysel týchto kódov v praxi. V druhej kapitole je popísaná údajová veda z teoretického hľadiska. Tento pojem sa stáva čoraz viac populárnejším pojmom v pracovnom prostredí, kde sa údajová veda často využíva za účelom dosiahnutia vyšších ziskov. Autor práce vo štvrtej kapitole dokazuje, že táto práca veľmi úzko súvisí z praktikami a postupmi údajovej vedy. Medzitým, t.j. v kapitole č. 3 sú zhrnuté organizačné záležitosti projektu, ktoré bolo potrebné pripraviť a zohľadniť ešte pred samotným vypracovaním plne funkčného projektu. V štvrtej kapitole je popísaný spomínaný projekt programu MATLAB, ktorý je rozdelený do štyroch po sebe idúcich fáz. Tieto fázy sú teoretický popísané v druhej kapitole tejto práce a poznatky z teórie sú v praxi aplikované a popísané v štvrtej kapitole práce. Ide o fázy zberu počiatočných údajov, proces ETL, fáza analýzy a vizualizácie spracovaných, resp. pripravených údajov a interpretácia výsledkov, resp. postrehov z údajov.

1. Vývojové prostredie MATLAB

V tejto časti práce je opísané vývojové prostredie programu MATLAB. V podkapitole č. 1.1 je stručne opísaná pracovná plocha programu MATLAB. Pretože existuje mnoho spôsobov uchovávania informácií v tomto programe, v podkapitole č. 1.2 sú spomenuté len tie údajové typy, ktoré sa v tejto práci využívajú. V podkapitolách č. 1.3 a 1.4 sú v stručnosti vysvetlené spôsoby, akými sa uchovávajú rozsiahlejšie a zložitejšie kódy, ich výhody, resp. nevýhody a rozdiely medzi nimi. V podkapitole č. 1.5 je opísaný štandardný postup pri tvorbe interaktívnych aplikácii pomocou programu MATLAB.

1.1. Pracovná plocha

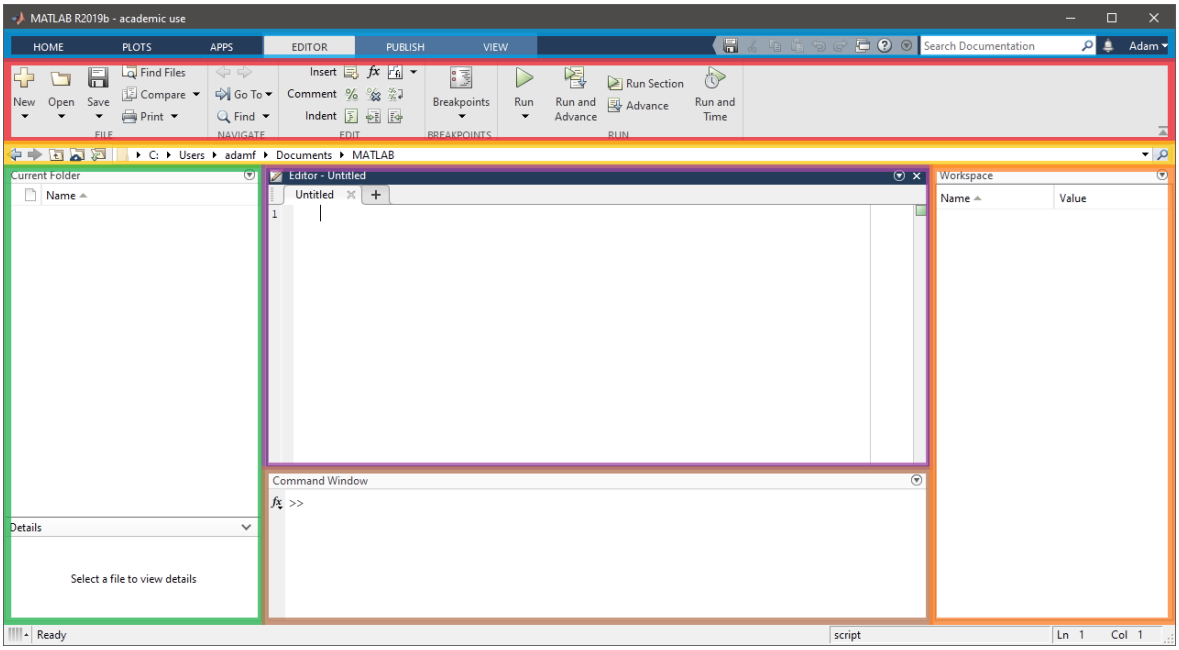
MATLAB® je programovacia platforma navrhnutá špeciálne pre inžinierov a vedcov. Jadrom MATLABu je jazyk MATLAB. Tento jazyk je založený na maticiach, čo umožňuje najprirodzenejšie vyjadrenie výpočtovej matematiky. Pomocou MATLABu môžeme analyzovať údaje, vyvíjať algoritmy, vytvárať modely a aplikácie. Jazyk, aplikácie a vstavané matematické funkcie nám umožňujú rýchlo preskúmať niekoľko prístupov a dospieť k riešeniu. MATLAB nám umožňuje presúvať naše nápady od výskumu po výrobu nasadzovaním do podnikových aplikácií a zabudovaných zariadení, ako aj integráciou so zabudovaným podprogramom Simulink®. MATLAB okrem iného ponúka aj tvorbu interaktívnych a profesionálnych aplikácii pomocou zabudovaného editora App Designer.

Táto práca bola vypracovaná autorom práce v programe MATLAB vo verzii R2019b. Niektoré súčasti tejto práce nemusia fungovať v starších verziách programu MATLAB. Spoločnosť The MathWorks, Inc. v spolupráci s Technickou Univerzitou v Košiciach poskytla bezplatné licencie pre študentov tejto univerzity, čo umožnilo autorovi vypracovať túto prácu legálnym spôsobom. Pri prvotnom spustení programu MATLAB sa používateľovi naskytne pohľad vyobrazený na Obr. 1.

Rozloženie jednotlivých komponentov programu je pomerne intuitívne. V hornej časti pozdĺž celého zobrazenia sa v časti zvýraznenej svetlomodrým obdĺžnikom na Obr. 1 nachádza hlavný panel programu MATLAB. Na tomto paneli sa v jeho pravej časti nachádza výber šiestich hlavných sekcií. Na základe tohto výberu sa používateľovi prispôsobí ponuka prvkov na paneli umiestnenom pod hlavným panelom, zvýraznenom v červenom obdĺžniku na Obr. 1. V ľavej časti hlavného panelu je umiestnená ponúka všeobecných nastavení programu MATLAB ako napr. autentifikácia používateľa, upozornenia o novinkách alebo vyhľadávací panel, ktorý používateľovi umožňuje prehľadávať podrobnú oficiálnu dokumentáciu [1] k jednotlivým metódam a funkciám.

Ako už bolo spomenuté, pod hlavným panelom sa nachádza panel prvkov. Sú tu umiestnené rôzne ovládacie prvky v závislosti od výberu sekcie na hlavnom paneli.

Pod panelom prvkov sa nachádza výber súčasného pracovného priečinka (z angl. Present Working Directory). To znamená, že príkazy, ktoré používateľ vykonáva sa viažu na tento priečinok, pokiaľ nie je používateľom explicitne zadefinovaný iný pracovný priečinok. Tento panel je na Obr. 1 zvýraznený žltým obdĺžnikom. Okrem súborov, ktoré sa nachádzajú v inštalačnej zložke programu a súborov, ktoré sa nachádzajú v súčasnom pracovnom priečinku, MATLAB nepozná žiadne iné súbory. Z tohto dôvodu je potrebné sa nastaviť do správneho pracovného priečinku. V prípade, že projekt je rozložený do viacerých priečinkov, je potrebné pridať všetky tieto priečinky do cesty.



Obr. 1 Náhľad programu MATLAB pri prvotnom spustení.

V ľavej časti pod výberom súčasnej pracovnej cesty sa nachádza prieskumník súborov (z angl. File Explorer). V tejto časti sa zobrazujú súbory a priečinky, ktoré sa nachádzajú v pracovnom priečinku. Z tohto miesta vie používateľ pridať viacero priečinkov do cesty. Prieskumník súborov je na Obr. 1 umiestnený v oblasti zeleného obdĺžnika. Označením a pravým kliknutím myši na priečinky, ktoré chce používateľ pridať do cesty zvolí možnosť *Add to Path > Selected Folder and Subfolders*. Používateľ týmto krokom zabezpečí, aby program MATLAB vedel, že existujú aj také priečinky a súbory, s ktorými používateľ chce ďalej pracovať. V dolnej časti tohto okna sa nachádza rýchly náhľad a to najmä pre označený obrázok alebo zoznam premenných, ktoré sú uložené v označenom súbore s príponou **.mat*.

Na pravej strane programu sa nachádza pracovná plocha MATLABu (z angl. Workspace), ktorá je na Obr. 1 umiestnená v oblasti oranžového obdĺžnika. Na tomto mieste sa nachádza zoznam všetkých

premenných, s ktorými používateľ momentálne pracuje. Tieto premenné sú uložené v dočasnej pamäti RAM. Pre trvalé uloženie premenných je potrebné uložiť tieto premenné do súboru s príponou **.mat*. Premenné pracovnej plochy sa dajú uložiť viacerými spôsobmi. Najjednoduchším spôsobom je označiť premenné, ktoré chce používateľ uložiť a pravým kliknutím myši na niektorú z označených premenných zvoliť možnosť *Save As...*, čím program vyzve používateľa zvoliť cestu a názov súboru s príponou **.mat*.

Uprostred dolnej časti programu MATLAB sa nachádza oblasť príkazového riadku (z angl. Command Window), ktorá je na Obr. 1 zvýraznená v oblasti hnedého obdĺžnika. Pomocou tohto okna komunikuje používateľ s programom. Slúži predovšetkým na spúšťanie jednoduchých príkazov ale je vhodný a nápomocný aj pri riešení chýb (z angl. Debugging) komplexnejších kódov. Pri tvorbe komplexnejších kódov je používanie príkazového riadku nepraktické, resp. menej praktické ako okno, ktoré sa v predvolenom zobrazení nachádza priamo nad príkazovým riadkom.

V okne, ktoré sa nachádza nad príkazovým riadkom, zvýrazneným v oblasti fialového obdĺžnika na Obr. 1, je otvorený prázdny skript, resp. funkcia s názvom *Untitled*, čo znamená, že súbor ešte nebol uložený. Pri pokuse o spustenie kódu, ktorý je zapísaný v tomto neuloženom súbore s názvom *Untitled*, program pred spustením vyzve používateľa o uloženie súboru pod ľubovoľným názvom bez použitia diakritiky. Uložený súbor, ktorý obsahuje kód programu MATLAB má príponu **.m* a v závislosti od štruktúry kódu sa tento súbor stáva buď skriptom alebo funkciou spustiteľnou v programe MATLAB. Bližšie detaily sú opísané v podkapitolách č. 1.3 a 1.4.

1.2. Údajové typy

Program MATLAB disponuje širokou škálou údajových typov. Z tohto dôvodu sa spomínajú len tie, ktoré sa priamo využívajú v projekte tejto práce. Tieto údajové typy sú kategorizované do niekoľkých skupín. Patrí tu viacero skupín ale tohto projektu sa týka len skupina numerických hodnôt, skupina znakov a reťazcov, skupina dátumov a časov, skupina údajových rámcov, bunkový a logický údajový typ.[2]

Najčastejšie využívanou skupinou údajových typov v tomto projekte sú čísla, resp. skupina numerických údajových typov. V predvolenom nastavení MATLAB ukladá všetky číselné hodnoty ako desatinné čísla s presnosťou dvoch desatinných miest. Tento predvolený typ a presnosť nie je možné zmeniť. Používateľovi je umožnené vybrať si, či chce ľubovoľné číslo alebo pole číselných hodnôt uložiť ako celé čísla alebo ako desatinné čísla. Údajové typy celočíselných hodnôt obsadzujú menej pamäte RAM ako údajový typ číselných hodnôt s desatinou čiarkou. Všetky číselné údajové typy podporujú základné operácie, ako sú napr. indexovanie, zmena tvaru a matematické

operácie.[3] V tomto projekte sa využíva len jeden z týchto numerických údajových typov a to údajový typ *double*, ktorý predstavuje desatinné číslo.

Ďalšia skupina údajových typov, ktorá sa v projekte využíva je skupina znakov a reťazcov. V princípe ide o spôsoby uchovania textových údajov do poli znakov a poli textových reťazcov. Polia znakov a textových reťazcov poskytujú úložisko pre textové údaje v MATLABe. Pole znakov uchováva postupnosť znakov, rovnako ako numerické pole uchováva postupnosť čísel. Typické použitie tohto údajového typu je ukladanie znakov, resp. vektory znakov. Pole textových reťazcov je údajový typ určený pre väčšie časti textu. Polia textových reťazcov poskytujú množinu funkcií pre prácu s textom ako aj s údajmi. Znak, resp. pole znakov *char* je v MATLABe definované pomocou apostrofov, zatiaľ čo textový reťazec, resp. pole textových reťazcov *string* je definované úvodzovkami.[4] V projekte sa využívajú obe údajové typy.

Do skupiny údajových typov zaoberajúcej sa dátumom a časom patria údajové typy dátum *datetime*, trvanie *duration* a kalendárne trvanie *calendarDuration*. Tieto údajové typy podporujú efektívne výpočty, porovnania a formátované zobrazenie dátumov a časov. S poliami týchto údajových typov sa dá pracovať rovnako ako s číselnými poliami. MATLAB umožňuje vykonávať operácie ako napr. pridávať, odčítavať, triediť, porovnávať, zreťaziť a vykresľovať hodnoty dátumu a času. Je možné tiež reprezentovať dátumy a časy ako číselné polia alebo ako text.[5] V projekte sa z tejto skupiny využívajú len údajové typy *datetime* a *duration*. Údajový typ *datetime* je schopný v sebe uchovávať dátum a čas s presnosťou na jednu milisekundu, zatiaľ čo údajový typ *duration* sa využíva predovšetkým na uchovávanie záznamu trvania, resp. plynutia času podobne ako stopky.

Skupina údajových rámcov (z angl. Data Frame) zahrnuje údajové typy ako tabuľka *table* [6], časová tabuľka *timetable* [7], a štruktúra *struct* [8]. Tabuľka *table* je údajový typ vhodný pre tabuľkové údaje, ktoré sa často ukladajú do stĺpcov v textovom súbore. Tabuľky pozostávajú z riadkov a stĺpcovo orientovaných premenných. Každá premenná v tabuľke môže mať odlišný typ údajov a inú veľkosť s tým obmedzením, že každá premenná musí mať rovnaký počet riadkov. Časová tabuľka *timetable* je veľmi podobný údajový typ ako tabuľka *table*. Líši sa len v tom, že označenia jej riadkov sú namiesto číselných hodnôt, ktoré znamenajú indexáciu sú v časovej tabuľke *timetable* nahradené časovou značkou (z angl. Timestamp). Pole štruktúry *struct* je údajový typ, ktorý zoskupuje súvisiace údaje pomocou údajových kontajnerov nazývaných polia. Každé pole môže obsahovať ľubovoľný typ údajov. Prístup k údajom, ktoré sú uložené v jednotlivých stĺpcoch, resp. poliach týchto údajových typov získavame cez bodku pomocou zápisu *nazovDatovehoRamca.nazovStlpca*. V projekte sa využívajú všetky tri údajové typy z tejto skupiny.

Ďalším údajovým typom, ktorý sa v projekte využíva je bunkový údajový typ *cell*. Bunka, resp. bunkové pole je údajový typ s indexovanými údajovými kontajnermi nazývanými bunky, kde každá bunka môže obsahovať akýkoľvek typ, veľkosť a počet dimenzií údajov. Bunkové polia obyčajne obsahujú buď zoznamy znakových vektorov rôznych dĺžok alebo zmesi textových reťazcov a čísiel alebo číselných polí rôznych veľkostí. Množiny buniek označujeme tak, že do klasických zátvoriek vkladáme indexy, resp. adresy poľa. K obsahu buniek môžeme pristupovať indexovaním pomocou množinových zátvoriek.

Posledný údajový typ, ktorý sa využíva v projekte je údajový typ logických hodnôt *logical*. Tento údajový typ môže nadobúdať len 2 hodnoty a to *true* alebo *false*. Využíva sa vo všeobecnosti pomerne často a nie len v tomto v projekte.

1.3. Skripty

Najjednoduchší typ programu MATLAB sa nazýva skript. Skript je súbor, ktorý obsahuje postupnosť niekoľkých riadkov príkazov programu MATLAB. Skript je možné spustiť viacerými spôsobmi. Jedným zo spôsobov je zadať jeho názov bez prípony **.m* do príkazového riadku. Druhý možný spôsob je spustiť ho pomocou zeleného tlačidla v tvare šípky s nápisom *Run* alebo stlačením klávesy F5. Tlačidlo *Run* sa nachádza v sekcií *EDITOR > RUN*. Spustením skriptu sa postupne vykonajú všetky príkazy od prvého až po posledný riadok zapísaný v skripte. V skripte môže byť zadefinovaná funkcia, ale musí byť definovaná na konci skriptu. Zadefinovaných funkcií môže byť v jednom skripte viacero. Čo je to funkcia je bližšie popísane v nasledujúcej kapitole.

1.4. Funkcie

Obsahom funkcie je sekvencia príkazov podobne ako v prípade skriptov. Zásadný rozdiel medzi skriptom a funkciou je ten, že funkcia môže prijímať vstupy a vracať výstupy. Aj skript aj funkcia používa príponu **.m*. S touto skutočnosťou súvisí aj ďalší zásadný rozdiel medzi nimi. K tomu, aby MATLAB považoval **.m* súbor ako funkciu je potrebné, aby sa v súbore nachádzala len funkcia, resp. viacero funkcii bez takých časti kódu, ktoré sa nachádzajú mimo funkcii. Ďalšia významná vlastnosť funkcii je tá, že funkcie musia byť definované v programovom súbore a nie na príkazovom riadku. Na definovanie funkcie používame niektorý z nasledujúcich zápisov:

```
% Zápis funkcie č.1
function vystup = nazov_funkcie(vstup)
% Obsah funkcie.
end

% Zápis funkcie č.2
function [vystup1, vystup2] = nazov_funkcie(vstup1, vstup2)
% Obsah funkcie.
end
```

```
% Zápis funkcie č.3
function varargout = nazov_funkcie(varargin)
% Obsah funkcie.
end
```

Všetky tri spôsoby zápisu definovania funkcie s názvom sú správne. V prvom prípade definujeme funkciu, ktorá má len jeden vstup s názvom *vstup* a len jeden výstup s názvom *vystup*. Druhý spôsob poukazuje na definíciu funkcie s viacerými vstupnými a výstupnými argumentmi funkcie. Pokiaľ je vstupných, resp. výstupných argumentov viacero, musia byť oddelené čiarkami. V prípade výstupných argumentov sa okrem iného musia nachádzať v hranatých zátvorkách. Pokiaľ nevieme odhadnúť koľko vstupných, resp. výstupných argumentov budeme potrebovať pre správny návrh funkcie, odporúča sa použiť tretí zápis. V tomto prípade ide o univerzálny zápis definovania počtu vstupných a výstupných argumentov pomocou kľúčových slov *varargin* pre vstupné argumenty a *varargout* pre výstupné argumenty. Medzi definovaním funkcie pomocou kľúčového slova *function* a kľúčovým slovom *end* definujeme sekvenciu príkazov, ktoré sa vykonajú pri zavolaní funkcie. Túto sekvenciu nazývame aj telom funkcie. V prípade tretieho zápisu vieme v tele funkcie pristupovať k argumentom pomocou zápisu *varargin{1}* pre prvý vstupný argument, *varargin{2}* pre druhý vstupný argument, *varargout{1}* pre prvý výstupný argument a podobne. V prípade prvých dvoch spôsobov zápisu definovania funkcie vieme pristupovať k argumentom priamo pomocou ich názvov. Všetky premenné, ktoré boli vytvorené v tele funkcie existujú len tam a po skončení funkcie sa vymažú z pamäte. Súbor s príponou **.m*, v ktorom je uložená len funkcia, resp. len skupina funkcii sa musí volať rovnako ako názov definovanej funkcie, resp. názov jednej z viacerých funkcii. V opačnom prípade funkciu nebude možne zavolať. Podobne ako skript, funkciu vieme zavolať viacerými spôsobmi. Najčastejšie používané spôsoby volania funkcie sú nasledovné:

```
[o1, o2] = nazov_funkcie(i1, i2);

o = nazov_funkcie(i);

nazov_funkcie(vstup);

nazov_funkcie;
```

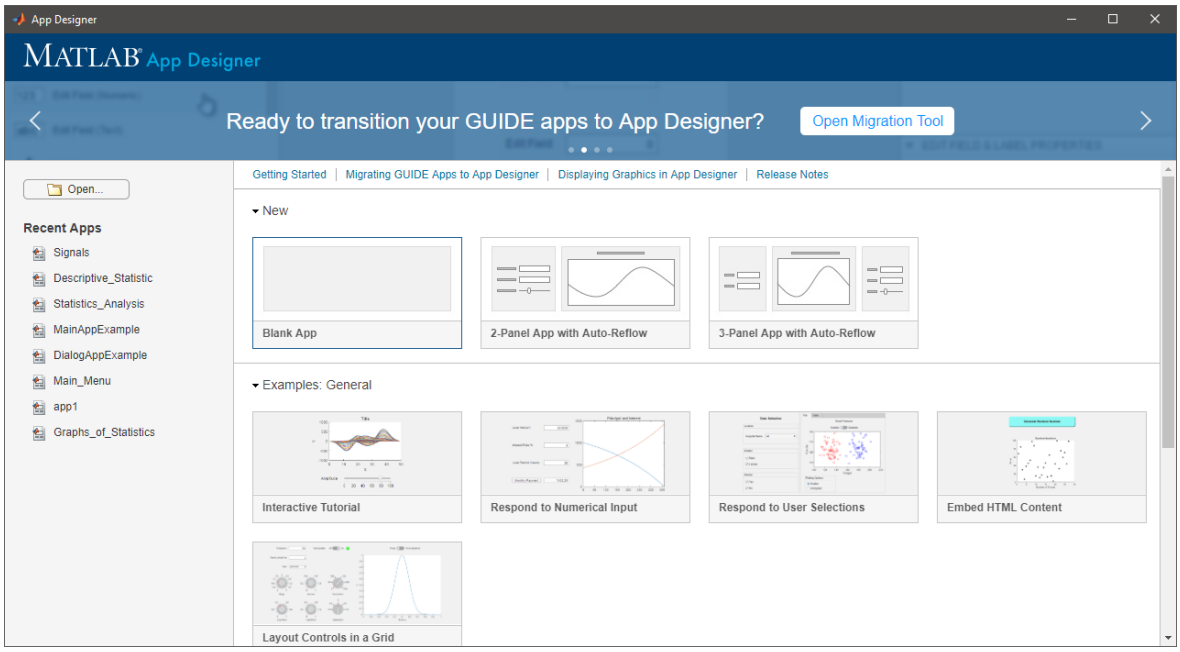
Opäť všetky štyri zápisy volania funkcie s názvom *nazov_funkcie* sú správne. V prvom riadku je znázornený príklad volania funkcie s dvomi vstupnými (*i1* a *i2*) a dvomi výstupnými (*o1* a *o2*) parametrami. Tu je potrebné poukázať na niekoľko skutočnosti. Pri definovaní funkcie nazývame vstupno-výstupné premenné argumentami, zatiaľ čo pri volaní funkcie ich nazývame parametrami. Názvy argumentov a parametrov môžu ale nemusia byť rovnaké, zatiaľ čo ich počet a poradie musí byť zachované. V druhom riadku je znázornený príklad s jedným vstupným a jedným výstupným parametrom. V treťom riadku je znázornený príklad bez výstupných parametrov a v štvrtom riadku

je znázornený príklad bez parametrov. Ak je v definícii funkcie definovaný aspoň jeden výstupný argument, tretí a štvrtý príklad vráti prvý výstupný argument do premennej (parametra) s názvom *ans*. Ide o automatický vygenerovanú premennú programom MATLAB vtedy, ak je zavolaná taká funkcia, ktorá má aspoň jeden výstupný argument a zároveň nebol špecifikovaný názov premennej, do ktorej sa má výstup tejto funkcie uložiť.

1.5. Aplikácie

Jednou zo súčastí programu MATLAB je editor na tvorbu aplikácií s názvom *App Designer*. Pomocou tohto editora je používateľ schopný vytvoriť interaktívnu aplikáciu podľa svojich predstáv. *App Designer* integruje dve základné úlohy vytvárania aplikácií - rozloženie vizuálnych komponentov grafického používateľského rozhrania (GUI) a programovania správania sa aplikácií. Je to odporúčané prostredie na vytváranie aplikácií v MATLABe.[9] Tento editor vieme otvoriť niekoľkými spôsobmi. Jedným z najjednoduchších spôsobov je napísať do príkazového riadku MATLABu príkaz *appdesigner*. Druhým spôsobom by bolo zvoliť na hlavnom paneli sekciu *APPS* a v časti *FILE* kliknúť na možnosť *Design App*.

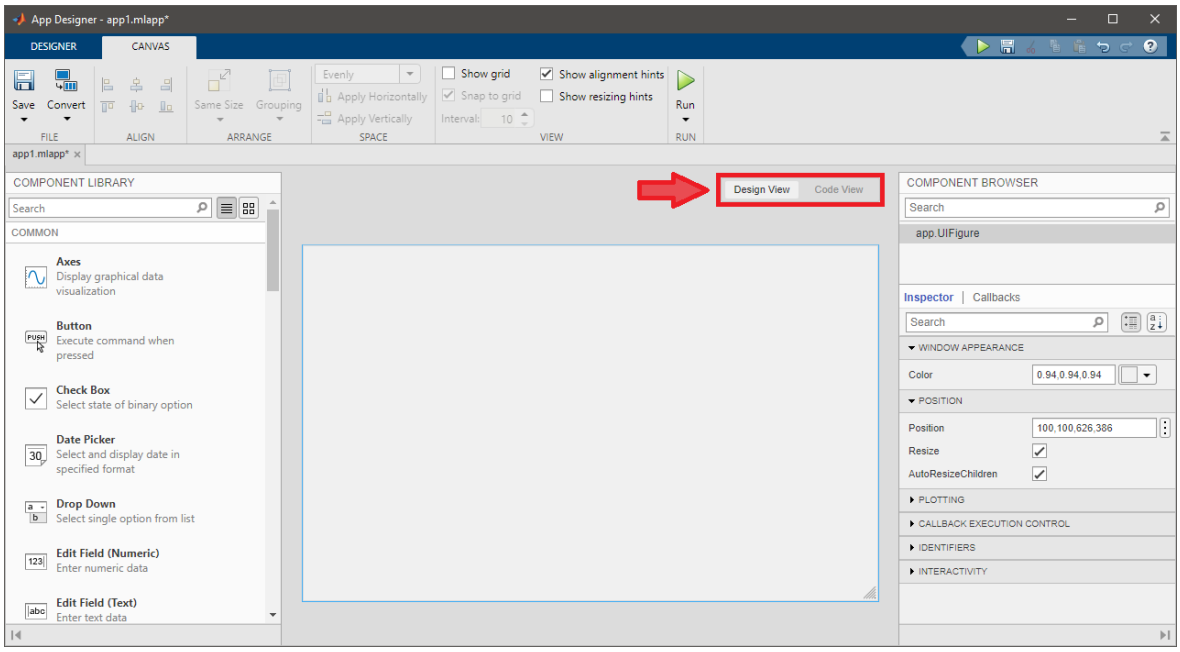
Po využití jedného z týchto spôsobov sa používateľovi otvorí úvodné okno editora vyobrazené na Obr. 2. Okno editora sa skladá z niekoľkých častí. V hornej modrej časti sa nachádza panel s novinkami. V ľavej časti úvodnej obrazovky sa nachádza zoznám nedávnych aplikácii, s ktorými používateľ nedávno pracoval v tomto editore a v hlavnej časti tohto okna, čiže napravo od zoznamu sa nachádza niekoľko predpripravených šablón aplikácii, ktoré by mohli po menších úpravách urýchliť používateľovi návrh jeho plánovanej aplikácie. Pre interpretáciu tvorby novej aplikácie bola z tohto výberu zvolená čistá aplikácia, čiže z angl. *Blank App*.



Obr. 2 Úvodné okno editora *App Designer*

Po kliknutí na možnosť *Blank App* sa používateľovi otvorí okno tvorby novej aplikácie znázornené na Obr. 3. Táto aplikácia automaticky nadobúda dočasný názov *app1* s príponou **.mlapp*, čo je prípona pre spustiteľné aplikácie editora *App Designer*. Ide o dočasný názov aplikácie, pretože táto aplikácia ešte nie je nikde uložená. Pred jej prvým spustením je potrebné aplikácii definovať názov a priečinok, v ktorom sa bude nachádzať, podobne ako v prípade skriptov a funkcií.

Editor *App Designer* sa skladá z dvoch základných režimov zobrazení. Prvý režim je režim úpravy dizajnu aplikácie označený angl. *Design View*. V závislosti od navoleného režimu zobrazenia sa menia aj prvky editora. Režim zobrazenia je možné prepínať v červenej oblasti znázornenej šípkou na Obr. 3. Režim *Design View* umožňuje upravovať vlastnosti jednotlivých objektov aplikácie ako napr. veľkosť, pozíciu a mnoho ďalších. V pravej hornej časti sa nachádza zoznam objektov, resp. komponentov aplikácie (z angl. Component Browser). V prázdnej aplikácii sa na začiatku nachádza len jeden objekt aplikácie a to návrhové plátno *UIFigure*, na ktoré pridáva používateľ komponenty z knižnice. Pri kliknutí na akýkoľvek objekt z tohto zoznamu sa zobrazia všetky dostupné vlastnosti tohto objektu a to priamo pod zoznamom objektov. Zároveň sa tento zvolený objekt zvýrazni v oblasti návrhu dizajnu aplikácie, ktorá sa nachádza uprostred editora. Po ľavej strane editora sa nachádza knižnica všetkých dostupných komponentov, ktoré môže používateľ vkladať do aplikácie. Tvorba aplikácie je pomerne intuitívna. Používateľ po presunutí komponentov na návrhové plátno a pomocou rád na zarovnanie objektov (z angl. alignment hints) získa presné rozloženie. Zobrazovanie týchto rád je pri základnom nastavení povolené. *App Designer* následne automaticky vygeneruje objektovo orientovaný kód, ktorý určuje rozloženie a dizajn aplikácie.



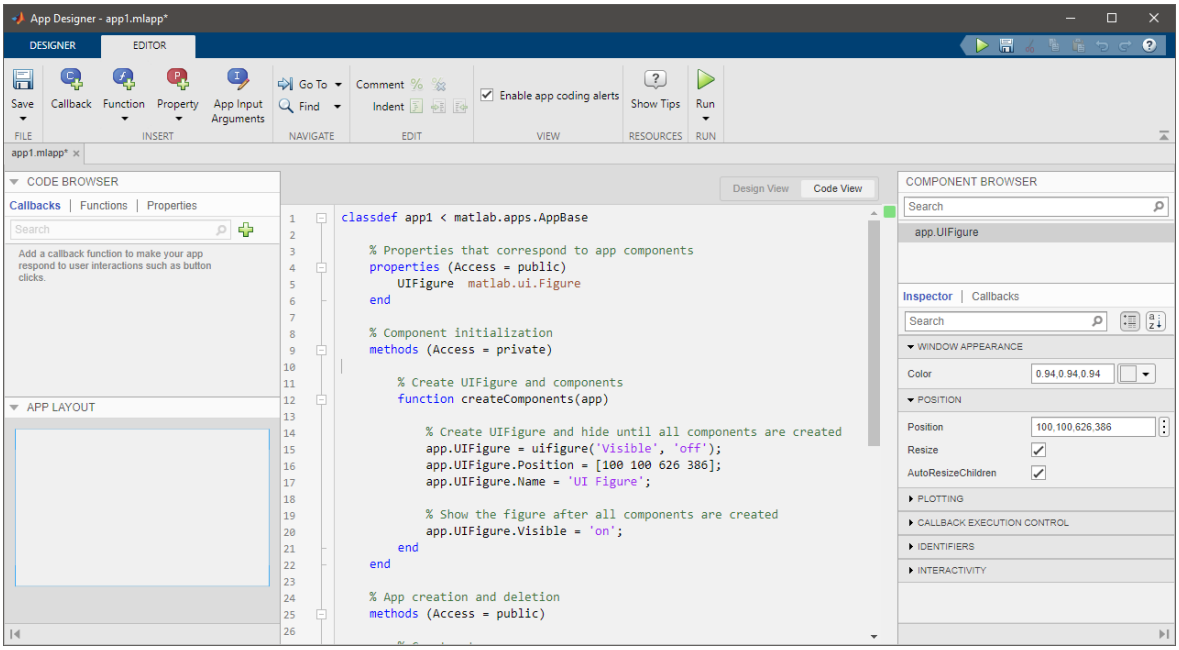
Obr. 3 Okno úpravy dizajnu aplikácie editora *App Designer*

Druhý režim zobrazenia s názvom *Code View* slúži na editáciu prepojení medzi jednotlivými komponentami aplikácie a ich správanie za určitých okolností. Ukážka okna v tomto režime je znázornená na Obr. 4. Na ľavej strane sa nachádza prehľad štruktúry kódu (z angl. Code Browser), ktorý sa nachádza uprostred obrazovky. V dolnej časti tohto panela sa nachádza rýchly náhľad dizajnovej časti aplikácie. Panel na pravej strane sa pri prepínaní medzi režimami nemení. Rovnako tak sa nemení ani hlavný panel umiestnený v hornej časti, ktorý plní podobnú funkciu ako v programe MATLAB. Každý kód aplikácie je v MATLABe rozdelený na 3 časti. Prvou skupinou sú takzvané funkcie spätných volaní (z angl. Callbacks). Funkcia spätného volania je taká funkcia, ktorá sa vykoná, keď používateľ interaguje s komponentom používateľského rozhrania v aplikácii. Väčšina dostupných komponentov môže mať aspoň jedno spätné volanie. Niektoré komponenty ako napríklad štítky a žiarovky, však nemajú žiadne spätné volania, pretože tieto komponenty zobrazujú iba informácie.[10] Druhú skupinu tvoria pomocné funkcie (z angl. Functions). Pomocné funkcie sú funkcie MATLABu, ktoré užívateľ definuje vo svojej aplikácii, aby ich mohol volať na rôznych miestach svojho kódu. Napríklad môže chcieť aktualizovať zobrazenie potom, čo zmení číslo v editovacom poli alebo vyberie položku z rozbaľovacieho zoznamu. Vytvorenie pomocnej funkcie mu umožňuje jednosmerné spoločné príkazy a vyhnúť sa nutnosti udržiavať nadbytočný kód.[11] V podstate spĺňajú rovnaký účel ako klasické funkcie opísane v kap. 1.4, avšak v editore *App Designer* na to majú vyhradené osobitné miesto v kóde. Poslednú skupinu tvoria vlastnosti (z angl. Properties). Použitie vlastností je najlepší spôsob zdieľania údajov v aplikácii, pretože vlastnosti sú prístupné všetkým pomocným funkciám a funkciám spätných volaní v aplikácii.[12] Všetky komponenty používateľského rozhrania sú tiež vlastnosťami, takže používateľ môže použiť túto

syntax na prístup a aktualizáciu komponentov používateľského rozhrania v rámci svojich spätných volaní:

```
app.Komponent.Vlastnost
```

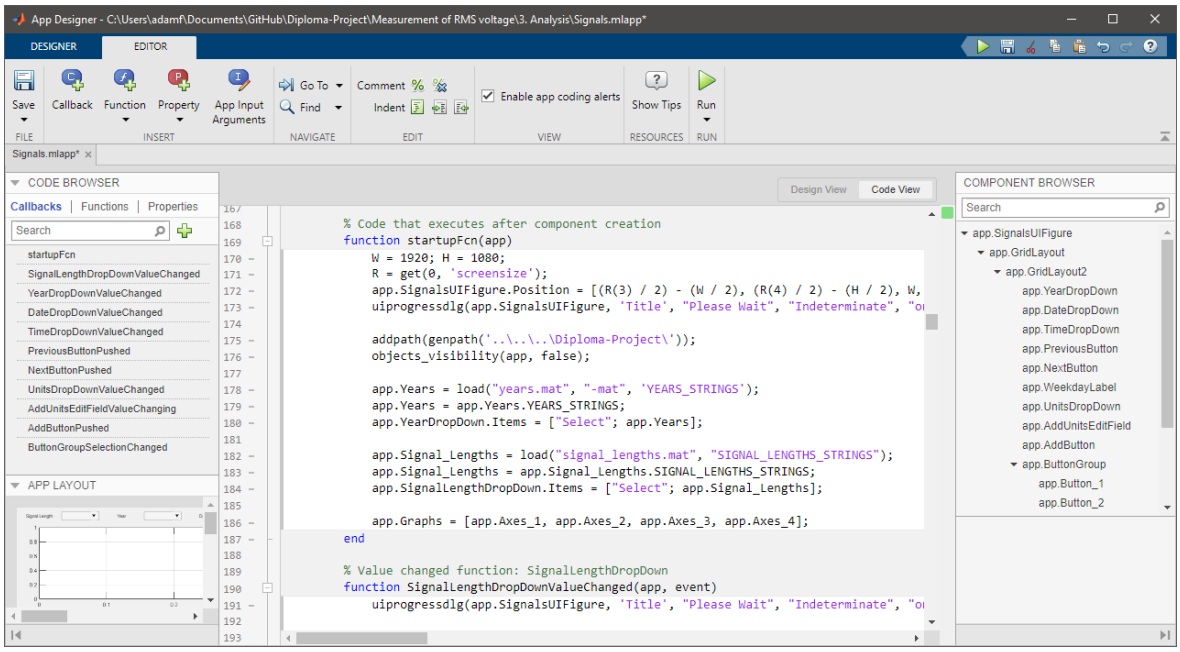
V editore *App Designer* sa dajú editovať len tieto 3 skupiny, resp. časti kódu. Ostatné časti kódu používateľ nemôže upraviť. Tieto časti kódu sú farebne odlíšené pozadím priamo v kóde režimu *Code View*.



Obr. 4 Okno úpravy kódu aplikácie editora *App Designer*

Používateľ môže upravovať len tie časti kódu, ktoré majú svetlo biele pozadie. K šedej časti kódu používateľ nemá prístup. Túto skutočnosť je dobré vidieť na Obr. 5. Z tohto dôvodu budú v tejto práci opísané len tie časti kódu, ktoré boli tvorené autorom tejto práce.

Aplikáciu spúšťame podobným spôsobom ako skript, čiže buď tlačidlom v tvare zelenej šípky s nápisom *Run* alebo vpísaním názvu uloženej aplikácie do príkazového riadku v programe MATLAB. Editor pri spúšťaní aplikácie nemusí byť otvorený. Z toho dôvodu je rýchlejšie spúšťať aplikácie z príkazového riadku. Nevýhodou MATLABu je skutočnosť, že aplikácia sa dá spustiť len vtedy, ak je otvorené okno MATLABu.



Obr. 5 Ukážka editovateľnej časti kódu aplikácie

2. Údajová veda

Údajová veda, resp. veda o údajoch (z angl. Data Science) je štúdium údajov. Zahŕňa vývoj metód zaznamenávania, ukladania a analýzy akéhokoľvek typu údajov s cieľom efektívne extrahovať užitočné informácie, resp. postrehy.[13] Cieľom vedy o údajoch je získať informácie a vedomosti z akéhokoľvek typu údajov - štruktúrovaných aj neštruktúrovaných. Údajová veda súvisí s počítačovou vedou, ale je to samostatná oblasť. Počítačová veda zahŕňa vytváranie programov a algoritmov na zaznamenávanie a spracovanie údajov, zatiaľ čo údajová veda pokrýva akýkoľvek druh analýzy údajov, ktorý môže alebo nemusí používať počítače. Údajová veda veľmi úzko súvisí s matematickou oblasťou, presnejšie štatistika, ktorá zahŕňa zber, organizáciu, analýzu a prezentáciu údajov. Z dôvodu veľkého množstva údajov, ktoré moderné spoločnosti a organizácie udržiavajú, sa veda o údajoch stala neoddeliteľnou súčasťou IT oblasti. Napríklad spoločnosť, ktorá má rádovo petabajty údajov o svojich zamestnancov alebo zákazníkov, môže aplikovať aspekty údajovej vedy na vývoj efektívnych spôsobov ukladania, správy a analýzy údajov. Spoločnosti tak môžu využívať vedecké metódy na vykonávanie rôznych testov a následné získavanie výsledkov, ktoré môžu poskytnúť zmysluplné informácie o ich používateľoch, resp. zamestnancoch.[14] V tejto kapitole budú popísané jednotlivé fázy a procesy, ktoré pri manipulácii údajov v údajovej vede prebiehajú. Zohľadňuje štandardný postup pri práci účastníkov údajovej vedy z teoretického hľadiska. Je vhodné poznamenať, že na týchto procesoch sa štandardne podieľajú viacerí zamestnanci spoločnosti, pričom každý z nich je špecializovaný na svoju konkrétnu činnosť práce.

2.1. Zber údajov

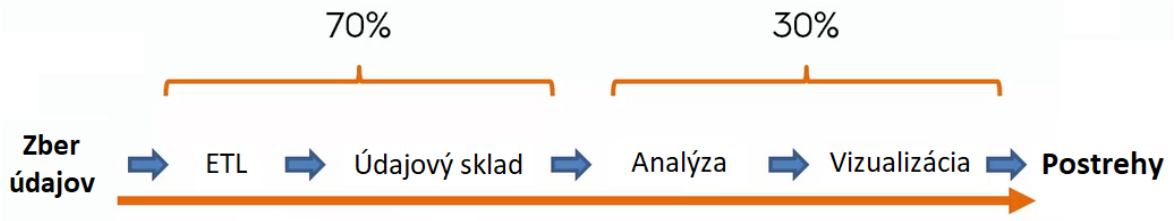
Nespracované údaje, resp. údaje bez štruktúry získavame pri prvom procese údajovej vedy a to zberu údajov. Zber údajov (z angl. Data Acquisition alebo skrátene DAQ) je proces merania elektrického alebo fyzikálneho javu, ako je napr. napätie, prúd, teplota, tlak alebo zvuk, pomocou počítača. Systém DAQ pozostáva zo senzorov, hardvéru na meranie DAQ a počítača s programovateľným softvérom. V porovnaní s tradičnými meracími systémami využívajú DAQ systémy založené na PC výkonnosť spracovania, produktivitu, zobrazovanie a možnosti pripojenia v štandardných počítačoch, ktoré poskytujú výkonnejšie, flexibilnejšie a nákladovo efektívnejšie riešenie merania.[15] Výsledkom procesu zberu údajov sú neštruktúrované údaje, ktoré sú uložené alebo sa ukladajú v prítomnom čase do údajových súborov na zariadenie s meracími senzormi. Väčšinou sa tieto údaje ukladajú do súborov vo formáte CSV (z angl. Comma Separated Values) alebo JSON (z angl. JavaScript Object Notation). Aj napriek tomu, že tieto formáty definujú určité štruktúry, neznamená to, že údaje majú štruktúru vhodnú na analytické účely. Tieto údaje majú svoje miesto v databáze a odtiaľ ďalej putujú do ďalšieho procesu s názvom ETL proces. Tento proces je popísaný v nasledujúcej kapitole.

2.2. ETL proces

Proces extrahovania údajov z viacerých zdrojových systémov, ich transformácie podľa obchodných potrieb a ich načítania do cieľovej databázy sa bežne nazýva ETL proces, čo znamená proces extrakcie (z angl. Extract), transformácie (z angl. Transform) a načítania (z angl. Load) údajov. Zatiaľ čo ETL sa zvyčajne vysvetľuje ako tri odlišné kroky, v skutočnosti to príliš zjednodušuje, pretože je to skutočne široký proces, ktorý si vyžaduje celý rad akcií.

Prvý nárast popularity pojmu ETL bol zaznamenaný v sedemdesiatych rokoch, keď spoločnosti a organizácie začali na ukladanie svojich informácií používať viac databáz. Rýchlo sa tento proces stal štandardnou metódou na získavanie údajov z rôznych zdrojov, ich transformáciu a načítanie do cieľa. Cieľom sa rozumie ďalší proces údajovej vedy a to je analýza údajov. O niekoľko desaťročí neskôr sa údajové sklady (z angl. Data Warehouses) stali ďalšou populárnou vecou. Údajové sklady poskytovali zreteľnú databázu, ktorá integrovala informácie z viacerých systémov. S cieľom prispôsobiť sa neustále sa meniacemu svetu digitálnych technológií v posledných rokoch sa počet údajových systémov, zdrojov a formátov exponenciálne zvýšil, ale potreba ETL procesov zostala rovnako dôležitá pre širšiu stratégiu integrácie údajov jednotlivých spoločnosti.[16]

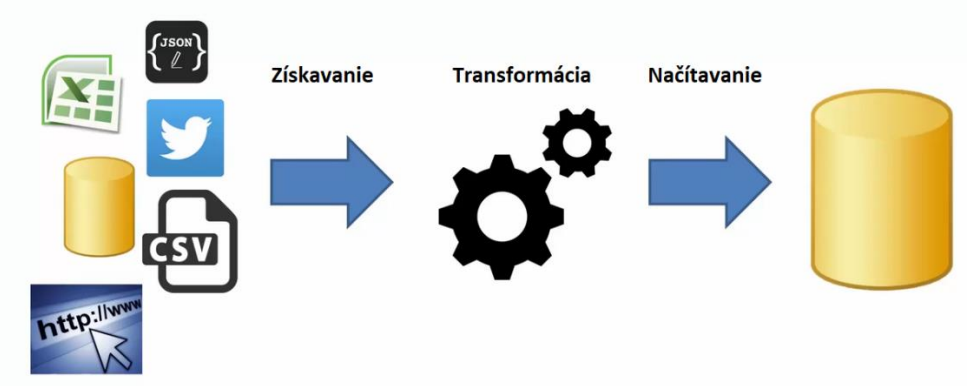
Odhaduje sa, že ETL proces za bežných okolností trvá približne 70 % celkového času údajovej vedy, z čoho vyplýva, že ide o pomerne náročný proces z hľadiska času. Tento čas je myslený od momentu, kedy boli údaje akýmkoľvek spôsobom získané až po extrakciu užitočných poznatkov a postrehov z týchto údajov. Táto skutočnosť je znázornená na Obr. 6.



Obr. 6 Časová os procesov údajovej vedy

Ako už bolo spomenuté, ETL proces sa skladá z troch hlavných krokov znázornených na Obr. 7. Prvým krokom v ETL procese je extrakcia údajov. Počas extrakcie sú údaje špecificky identifikované a potom prevzaté z mnohých rôznych miest, označovaných ako zdroj údajov. Zdrojom môže byť celý rad vecí ako sú súbory, tabuľky, databázové tabuľky, atď. Spravidla nie je možné presne určiť presnú podmnožinu záujmu, takže sa extrahuje viac údajov, ako je potrebné, aby sa zabezpečilo, že údajová veda pokrýva všetko potrebné. Objem extrahovaných údajov sa veľmi líši a závisí od požadovaných potrieb a požiadaviek. Niektoré extrakcie pozostávajú zo stoviek kilobajtov až po petabajty. Platí to aj pre časový odstup medzi dvoma po sebe idúcimi extrakciami. Niektoré sa môžu líšiť medzi dňami

alebo hodinami, prípadne prebieha extrakcia údajov v reálnom čase. Ďalším krokom v procese ETL je transformácia údajov. Po extrahovaní údajov je potrebné ich fyzicky preniesť do cieľového miesta a previesť do príslušného formátu. Táto transformácia údajov môže zahŕňať operácie, ako sú čistenie, spájanie a overovanie údajov alebo generovanie nových vypočítaných údajov na základe existujúcich hodnôt. Či už k transformácii dôjde v údajovom sklade alebo ešte skôr, existujú bežné ale aj pokročilé typy transformácie, ktoré zabezpečia prípravu štruktúrovaných údajov vhodných na analýzu. Základné transformácie zahŕňajú napr. čistenie, revízia formátu, reštrukturalizácia, deduplikácia a pod. Do skupiny pokročilých transformácii patria operácie ako napr. filtrovanie, spájanie, delenie, odvodenie, zhrnutie, integrácia a pod.



Obr. 7 Znázornenie vnútorných podprocesov ETL procesu

Posledný krok v procese ETL zahŕňa načítanie transformovaných údajov do cieľa. Týmto cieľom môže byť databáza alebo údajový sklad. Na načítanie údajov do skladu existujú dve primárne metódy a to úplné načítanie údajov a prírastkové načítavanie údajov. Metóda úplného načítania zahŕňa výpis všetkých údajov, ku ktorému dôjde pri prvom načítaní počiatočných údajov do údajového skladu. Na druhej strane k inkrementálnemu načítavaniu údajov dochádza v pravidelných intervaloch. Výsledkom tohto procesu sú štruktúrované údaje, vhodné na bezchybnú analýzu údajov.

2.3. Analýza údajov

Tento proces údajovej vedy zahrňuje analýzu prvotných údajov s cieľom vyvodiť z nich užitočné závery. Techniky analýzy údajov môžu odhaliť trendy a metriky, ktoré by sa inak stratili v obrovskom množstve nahromadených informácií. Tieto užitočné informácie sa potom môžu použiť na optimalizáciu určitých procesov na zvýšenie celkovej efektívnosti spoločnosti alebo systému. Vďaka analýze údajov vieme docieliť oveľa viac. Napríklad herné spoločnosti používajú techniky analýzy údajov na nastavenie rozpisov odmien pre hráčov za účelom udržať aktivitu čo najväčšieho

počtu hráčov v hre.[17] Súčasťou tohto procesu je zároveň zabezpečiť prostriedky a spôsoby, vďaka ktorým bude možné vykonať vizualizáciu údajov, ktoré podliehajú analýze údajov.

2.4. Vizualizácia údajov

Vizualizácia údajov je grafické znázornenie informácií a údajov. Pomocou vizuálnych prvkov ako sú grafy a mapy, poskytujú nástroje vizualizácie údajov spôsob, ako vidieť a porozumieť trendom, odľahlým hodnotám a vzorcom v údajoch. Vo svete veľkého objemu údajov sú nástroje a technológie vizualizácie údajov nevyhnutné na analyzovanie obrovského množstva informácií a prijímanie rozhodnutí na základe údajov.[18] Vizualizácia údajov veľmi úzko súvisí s ich analýzou a to najmä kvôli tomu, že už počas samotnej analýzy údajov sa nám ukazuje množstvo náhľadov vizualizácii údajov. Naše oči sú priťahované farbami a vzormi. Môžeme rýchlo identifikovať červený objekt na modrom pozadí alebo objekt tvaru štvorca medzi objektami v tvare kruhu. Keď sa človek pozerá na graf, rýchlo uvidí trendy a odľahlé hodnoty. Pri porovnaní s pohľadom na rozsiahlu tabuľku údajov môže byť vizualizácia jej obsahu ďaleko viac efektívnejšia.

2.5. Postrehy

Táto časť údajovej vedy nadväzuje na predchádzajúcu časť a veľmi úzko s ňou súvisí. Ide o finálnu fázu údajovej vedy, kedy spomedzi množstva vizualizácií vyberáme len tie najužitočnejšie pre účely definované v začiatočnej fáze údajovej vedy. Najčastejšie ide o dosiahnutie väčšieho zisku pre spoločnosť, prípadne optimalizáciu procesov, ktorá zabezpečí povedzme menšie náklady pre spoločnosť a teda nepriamo zvýši výnosy. Účel nemusí mať len materiálny charakter. Vďaka údajovej vede, respektíve vďaka výsledkom údajovej vedy vieme dospieť k zaujímavým zisteniam na základe ktorých sa vieme v budúcnosti lepšie rozhodovať.

3. Organizácia projektu v MATLABe

V tejto časti práce sú popísané organizačné záležitosti. V prvej podkapitole 3.1 sú popísané parametre zariadenia, resp. počítača, na ktorom bol projekt vyhotovený. V nasledujúcej podkapitole 3.2 je popísaná hierarchia priečinkov, ktorú tento projekt dodržiava. V poslednej podkapitole 3.3 je popísaný súčasný stav riešenia, resp. na základe čoho vychádza nápad pre vyhotovenie tohto projektu.

3.1. Hardvérové a softvérové parametre zariadenia

Na vypracovanie projektu v programe MATLAB bol použitý notebook značky Lenovo. Konkrétne bol použitý model IdeaPad 510-15ISK (80SR00AHCK)[19], ktorého čiastočná špecifikácia sa nachádza v Tab. 1. V tabuľke sa nachádzajú len tie hardvérové parametre, ktoré sa priamo týkajú, resp. vplývajú na rýchlosť funkcií a aplikácii tohoto projektu. Notebook bol rozšírený o SSD disk od spoločnosti Samsung, čo výrazne urýchlilo všetky procesy projektu. Ide o model 860 EVO SATA III 2.5inch SSD s kapacitou 500 GB.[20]

Tab. 1 Čiastočná špecifikácia zariadenia

Typ zariadenia	Notebook
Určenie	Multimediálny
Procesor (CPU)	Intel Core i5-6200U
Počet jadier CPU	2
Pamäť cache CPU	3 MB
Frekvencia CPU	2,3 – 2,8 GHz
Operačná pamäť (RAM)	8 GB
Typ pamäte RAM	DDR4 – 2133 MHz
Typ disku	HDD
Kapacita disku	1000 GB
Veľkosť displeja	15,6 ‘‘ (39,62 cm)
Rozlíšenie displeja	FHD 1920x1080 px
Pomer strán	16:9
Technológia displeja	ISP
Typ displeja	LCD
Grafická karta (GPU)	nVidia GeForce 940MX
Pamäť GPU	4 GB
Typ pamäte GPU	DDR3

Na tomto notebooku bol nainštalovaný operačný systém Windows 10 Home. Na notebook sa pravidelne inštalovali automatické aktualizácie pomocou funkcie Windows Update. Projekt bol vyhotovený v programe MATLAB verzie R2019b. Používanie akýchkoľvek častí projektu v iných verziách nemusí fungovať správne, prípadne nemusí fungovať vôbec.

3.2. Organizácia priečinkov projektu

Keďže sa tento projekt zaoberá údajovou vedou, je vhodné zabezpečiť aby jednotlivé časti projektu dodržiavali určitú hierarchiu. Štandardne sa projekt údajovej vedy delí na niekoľko fáz spomenutých v kap. 2. Rovnako tak bude aj tento projekt rozdelený na tieto fázy a to tak, že každej fáze projektu bude prislúchať osobitný adresár. Každý projekt údajovej vedy sa delí na 4 časti, resp. adresáre. V prvom adresári s názvom *1. Pôvodné údaje* sa nachádzajú prvotné údaje bez štruktúry získané najčastejšie meracím prístrojom s digitálnym výstupom. V tomto priečinku sa nachádza výstup prvej fázy údajovej vedy popísanej v kap. 2.1. V druhom priečinku tejto hierarchie s názvom *2. Pripravené údaje* sa budú nachádzať výstupy druhej fázy údajovej vedy popísanej v kap. 2.2. Vnútri tohto adresára sa nachádza ďalší adresár s názvom *ETL proces*, v ktorom sú uložené všetky funkcie a prostriedky programu MATLAB na vykonanie automatizovaného ETL procesu. V treťom priečinku s názvom *3. Analýza údajov* sa nachádzajú výstupy tretej a štvrtej fázy údajovej vedy, ktoré sú bližšie popísane v kap. 2.3 a 2.4. Keďže tieto fázy sa navzájom prelínajú, je vhodné ich zlúčiť do jedného spoločného adresára. V tomto adresári sa nachádza aj adresár s názvom *Chyby údajov*, ktorý sa ďalej delí na adresáre s názvami *Automaticky upravené riadky* a *Ručne upravené riadky*. Zmyslom tejto zložky je zaznamenávať akékoľvek zásahy a zmeny v údajoch, ktoré sa nachádzajú v zložke *2. Pripravené údaje*. Zmeny, ktoré nastali automaticky pri automatizovanom ETL procese budú automaticky vygenerované v príslušnej zložke po zbehnutí ETL procesu, t.j. *Automaticky upravené riadky*. Zmeny, ktoré vykoná používateľ ručne by mal z vlastnej iniciatívy zaznamenať nejakou vhodnou formou do zodpovedajúceho adresára, t.j. *Ručne upravené riadky.* V poslednej zložke hierarchie s názvom *4. Štatistiky údajov* sa nachádzajú len zaujímavé a potencionálne užitočné postrehy pri prehľadávaní vizuálne interpretovaných údajov z predchádzajúcej fázy údajovej vedy.

3.3. Súčasný stav riešenia

V roku 2018 bol autorom tejto diplomovej práce vyhotovený podobný projekt na spracovanie veľkého množstva údajov, avšak v oblasti programu MS Excel od spoločnosti Microsoft.[21] Tento projekt bol rozdelený do troch súborov spustiteľných v programe MS Excel. Oproti aktuálnemu návrhu projektu má pôvodné riešenie značné nevýhody. Jednou z najväčších nevýhod bola neschopnosť spracovať viac ako jeden milión riadkov a to práve vďaka

fixne definovanému limitu počtu riadkov v programe, ktorý nedovolí používateľovi prekročiť túto hranicu. Limit programu MS Excel je presne 1048576 riadkov [22], pričom celkovo riadkov získaných meraním bolo 150557655. Obísť sa to síce dalo využitím umiestnenia údajov do vedľajších voľných stĺpcov, avšak značne to komplikovalo prácu a najmä plynulosť spracovania takéhoto veľkého množstva údajov. Síce menej efektívnym spôsobom sme vyhotovením plne funkčného projektu zameraného na štatistickú analýzu rozsiahleho množstva údajov v programe MS Excel poukázali na skutočnosť, že aj takýmto pomerne známym programom vieme docieliť požadované výsledky. Pretože program MS Excel nie je určený na prácu s takýmto mohutným množstvom údajov, bolo potrebné zvoliť vhodnejší program pre túto problematiku. Aktuálny projekt v oblasti programovacieho jazyka MATLAB bol navrhnutý tak, aby vedel spracovávať nielen sieťové napätie ale aj iné údajové súbory meraných hodnôt s časovou značkou a to na základe postupností krokov opísaných v ďalších kapitolách tejto práce.

4. Návrh riešenia pomocou jazyka MATLAB

V tejto časti práce je podrobne popísaný význam všetkých časti projektu. Nachádza sa tu detailný postup jednotlivých fáz projektu, ktoré viedli k zaujímavým zisteniam, náhľadom a vizualizáciám. V Prílohe A sa nachádza celý predpripravený projekt spustiteľný v programe MATLAB.

4.1. Zber údajov

Táto fáza je ako jediná fáza riešená v tomto projekte externe. Tým je myslené najmä to, že autor tejto práce sa na tejto fáze vôbec nepodieľa. Napriek tomu je potrebné spomenúť princíp zberu údajov, keďže sa jedná o nevyhnutný proces údajovej vedy. Pri meraní efektívnej hodnoty sieťového elektrického napätia bol použitý stolový multimeter UNI-T UT803. Tento multimeter bol pripojený k stolovému počítaču cez komunikačný kanál RS-232. Na počítači, ktorý je spolu s multimetrom umiestnený na katedre teoretickej a priemyselnej elektrotechniky je nainštalovaný softvér, ktorého autorom je doc. Ing. Tibor Vince, PhD. Softvér bol navrhnutý tak aby fungoval v nepretržitej prevádzke a jeho úlohou je ukladať namerané efektívne hodnoty sieťového elektrického napätia každú sekundu spolu s časovými údajmi. Tieto údaje sú ukladané v riadkoch v údajovom súbore **.csv* formátu. Pri odčítavaní hodnôt elektrického odporu a elektrického prúdu na vodiči zapojenom v skrate bol použitý multimeter RIGOL DM 3068. Údaje síce boli tiež zaznamenávane v údajových súboroch formátu **.csv*, avšak nie prostredníctvom spomínaného softvéru. Meranie teploty a relatívnej vlhkosti vzduchu bolo uskutočnené skrz 8 kusov izbových teplomerov značky Sensirion SHT31 Gadget. Zozbierané údaje sú následne ukladané do textových súborov **.edf*. Keďže nasledujúce kapitoly, resp. funkcie a aplikácie sú zamerané hlavne na meranie efektívnej hodnoty sieťového elektrického napätia, budú vychádzať z formátu údajového **.csv* súboru, ktorý je možné vidieť na Obr. 8.

	1 Var1	2 Var2	3 Var3
1	'19.9.2014'	10:20:29.640	'228,1 V'
2	'19.9.2014'	10:20:30.640	'228,1 V'
3	'19.9.2014'	10:20:31.640	'228,4 V'
4	'19.9.2014'	10:20:32.640	'228,5 V'
5	'19.9.2014'	10:20:33.640	'228,4 V'
6	'19.9.2014'	10:20:34.656	'227,1 V'
7	'19.9.2014'	10:20:35.640	'227,1 V'
8	'19.9.2014'	10:20:36.640	'227,1 V'
9	'19.9.2014'	10:20:37.640	'228,3 V'
10	'19.9.2014'	10:20:38.640	'228,2 V'

Obr. 8 Náhľad prvých 10 riadkov náhodného údajového súboru merania sieťového napätia

Tento projekt je ale plne kompatibilný s akýmkoľvek typom merania, avšak musí byť zachovaný spomínaný formát údajového súboru. To znamená, že musí mať presne 3 stĺpce, pričom v prvom budú uvedené len dátumy, v druhom len časy a v treťom budú uvedené numerické hodnoty merania v danom čase daného dátumu.

4.2. ETL proces

Ako už bolo spomenuté v kap. 2.2, ETL proces je bezpochyby časovo najnáročnejší proces údajovej vedy. V závislosti od toho, ako dobre sú spracované zozbierané údaje po dokončení tohto procesu sa odvíja všetko ostatné, čo nasleduje po tomto procese. Z tohto dôvodu je preto veľmi dôležité tento proces nijakým spôsobom nepodceniť. To sa práve preukazuje na celkovom čase, ktorý je potrebne vymedziť na tvorbu automatizovaného ETL procesu. V tejto kapitole je popísaná významnosť jednotlivých ETL funkcii programu MATLAB. Kódy k týmto funkciám sú ďalej uvedené v Prílohe B. Funkcie *create_values*, *create_missing* a *create_statistics* vygenerujú po každom spustení textový súbor so záznamom v priečinku *ETL proces*. Rovnako tak v tom istom priečinku sú umiestnené aj všetky funkcie tohto procesu. Súbory sa začínajú slovom *log* (z angl. Záznam) a obsahom týchto textových súborov sú záznamy o dátume a čase, kedy bola funkcia zavolaná, aké vstupné parametre boli pri zavolaní zadefinované a ďalšie detailné informácie o priebehu daných funkcií.

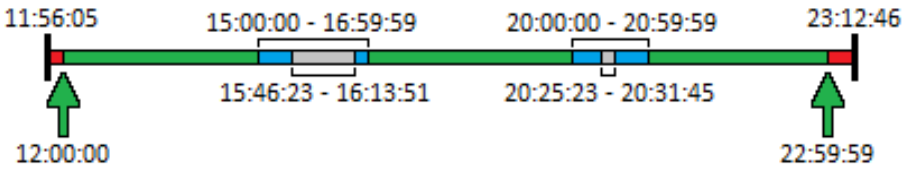
4.2.1. Funkcia *create_values_file*

Funkcia *create_values_file* je prvá funkcia ETL procesu, ktorá slúži na spracovanie jedného konkrétneho údajového **.csv* súboru zozbieraných údajov. Spracovanie zahŕňa niekoľko operácií pričom niektoré z nich musia byť vykonané v určitom poradí. Niektoré z týchto operácií sú nevyhnutné, niektoré viac-menej len optimalizujú rýchlosť programu a prehľadnosť údajov a niektoré sa aplikujú len za určitých podmienok. Sú to nasledujúce operácie:

- pridelenie názvov pre stĺpce (pretože po importe do MATLABu sa volajú *Var1*, *Var2* a *Var3*),
- zmena údajového typu údajov v stĺpcoch (pretože predvolený typ je textový reťazec),
- zaokrúhľovanie času na celé sekundy (ak je presnosť času lepšia, pretože nie je tak dôležitá),
- zlúčenie stĺpcov dátumu a času,
- odstránenie jednotiek zo stĺpca numerických hodnôt (ak sú uvedené),
- odstránenie duplicitných riadkov (ak náhodou sú takéto riadky),
- triedenie údajov na „čisté“ a vypustené.

Najdôležitejšou častou tejto funkcie je triedenie údajov na kontinuálne údaje bez jediného prerušenia (ďalej ako „čisté údaje“) a vypustené údaje. Toto triedenie prebieha za určitých podmienok. Cieľom tejto funkcie je zo zozbieraných údajov vyextrahovať len 1-hodinové intervaly

začínajúce od HH:00:00 a končiace do HH:59:59, pričom HH je hodina. Zároveň meranie v týchto hodinách nesmie byť nijakým spôsobom prerušené ani na jednu sekundu (pokiaľ je krokom merania jedna sekunda). Na Obr. 9 je znázornená časová os fiktívneho údajového súboru, ktorý slúži len na interpretáciu a lepšiu predstavu toho, ako táto funkcia funguje. Fiktívny údajový súbor obsahuje namerané údaje v čase od 11:56:05 do 23:12:46. Funkcia najprv vyhľadá začiatok najbližšej nikde neprerušenej hodiny merania a koniec poslednej nikde neprerušenej hodiny merania a následne vymaže tieto prebytočné hodnoty, ktoré sú na Obr. 9 znázornené červenou farbou. Miesta, ktoré sú na Obr. 9 znázornené sivou farbou sú časové úseky, kedy došlo k prerušeniu merania a teda všetky hodiny, ktorých sa toto prerušenie týka sú funkciou klasifikované ako vypustené údaje. Vypustené údaje sú na Obr. 9 znázornené modrou farbou a „čisté“ údaje sú znázornené zelenou farbou.



Obr. 9 Časová os fiktívneho údajového súboru

Funkcia má 3 vstupné argumenty. Funkcia očakáva, že prvý argument s názvom *PATH* bude predstavovať textový reťazec celej cesty vrátane názvu súboru s jeho **.csv* príponou. Za ním nasleduje druhý argument s názvom *STEP*, kde funkcia očakáva numerickú hodnotu, resp. prirodzené číslo, ktoré predstavuje počet sekúnd medzi jednotlivými meraniami. Posledný argument, ktorý očakáva funkcia je argument s názvom *UNITS*. Príklad spustenia tejto funkcie môže byť nasledovný:

```
[clean, excluded] = create_values_file("C:/Users/adamf/Desktop/subor1.csv", 1, "V");
```

Výstupom tejto funkcie sú 2 časové tabuľky. Prvá časová tabuľka je tabuľka s „čistými“ údajmi a druhá s vypustenými údajmi. Funkcia po dokončení všetkých operácii vypíše do oblasti príkazového riadku informácie o počte riadkov, resp. údajov merania. Príklad výpisu informácii po dokončení funkcie je nasledovný:

```
Vytváranie bolo dokončené.
Celkový počet údajov:           2591370
Počet vyčistených údajov:       2588400
Počet vylúčených údajov:        2930
Počet poškodených údajov:       40
```

Okrem kategórie „čistých“ a vylúčených údajov existujú ešte jedná kategorizácia údajov a to poškodené údaje. Funkcia vyhodnotí ako poškodený údaj každý taký riadok, v ktorom chýba buď dátum a čas alebo numerická hodnota veličiny alebo inak poškodený riadok, napr. duplicitný riadok.

4.2.2. Funkcia *create_values*

Táto funkcia je určená na spracovanie viacerých údajových súborov **.csv* formátu, ktoré sa nachádzajú v rovnakom priečinku. Funkcia je taktiež rozšírená o záznamník v podobe textového súboru s názvom *log_create_values.txt*, do ktorého sa ukladá všetko to, čo sa ocitne v okne príkazového riadku počas behu funkcie. Funkcia v princípe identifikuje všetky údajové súbory formátu **.csv* v zadanej ceste k priečinku a následne pre každý identifikovaný súbor zavolá funkciu *create_values_file* na spracovanie daného súboru. Funkcia *create_values* má identické vstupné a výstupné argumenty ako funkcia *create_values_file* s tým, že funkcia *create_values* má o jeden vstupný argument s názvom PROGRESS a jeden výstupný argument FILE_COUNT naviac. Oba tieto argumenty sú voliteľné, čo znamená, že nemusia byť vôbec uvedené pri volaní funkcie. Využívajú sa len pri volaní prostredníctvom aplikácie *ETL_Configuration*, ktorá je popísaná v kap. 4.2.6. Slúžia len na aktualizáciu dialógového okna aplikácie, ktoré je možné vidieť v kap. 4.2.6 na Obr. 12.

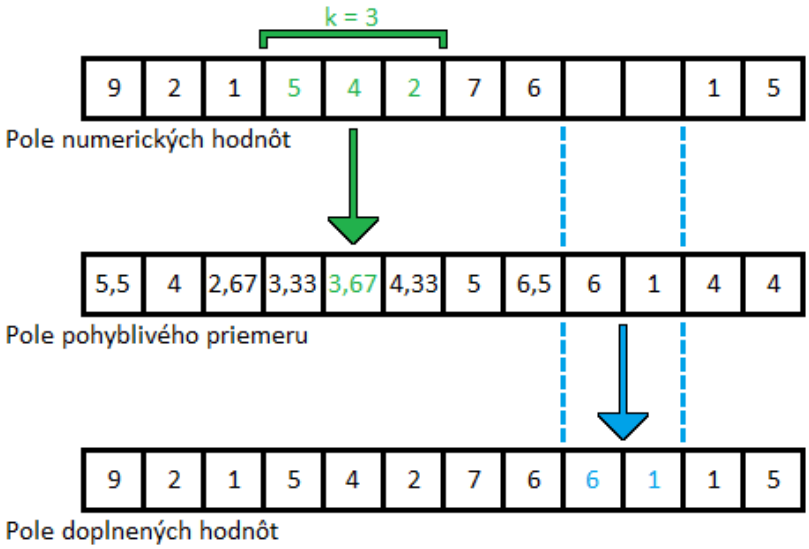
4.2.3. Funkcia *create_missing*

Funkcia *create_missing* slúži na identifikáciu prázdnych riadkov merania, resp. takých časových úsekov merania, pri ktorých došlo k prerušeniu. Keďže vychádzame z „čistých“ údajov, ktoré majú štruktúru 1-hodinových intervalov merania, chýbajúce časové úseky budú taktiež 1-hodinové úseky, resp. násobkami 1-hodinových intervalov. V tejto funkcii vystupujú 2 vstupné argumenty. Prvý vstupný argument s názvom CLEAN reprezentuje časovú tabuľku „čistých“ údajov. Druhý vstupný argument s názvom RECORD_STEP reprezentuje numerickú hodnotu, ktorá vyjadruje dĺžku krokov merania v sekundách. Táto dĺžka by mala byť identická s dĺžkou, ktorá bola zadaná pri tvorbe časovej tabuľky „čistých“ a vylúčených hodnôt pomocou funkcie *create_values*, resp. *create_values_file*. Výstupom tejto funkcie je tabuľka záznamov chýbajúcich časových úsekov merania, ktoré boli buď vypustené ETL procesom z analýzy údajov alebo neboli vôbec zaznamenané meracím prístrojom. Výstupná tabuľka má 3 stĺpce. Prvý stĺpec s názvom *Begin* vyjadruje začiatok časového intervalu chýbajúcich hodnôt. Druhý stĺpec s názvom *End* vyjadruje koniec časového intervalu chýbajúcich hodnôt. Tretí stĺpec s názvom *Hours* vyjadruje počet 1-hodinových intervalov, ktoré sa v tomto časovom rozmedzí nachádzajú. Keďže ide o presné násobky 1-hodinových časových intervalov, táto numerická hodnota bude vždy prirodzeným číslom.

4.2.4. Funkcia *create_filled*

Úlohou tejto funkcie je automaticky rekonštruovať chýbajúce časové intervaly, ktoré boli identifikované pomocou funkcie *create_missing*. V princípe táto funkcia príjme ako vstupné argumenty tabuľku chýbajúcich hodnôt s názvom MISSING, časovú tabuľku vypustených hodnôt s názvom EXCLUDED, šírku krokov merania v sekundách s názvom RECORD_STEP a hornú hranicu tolerancie automatického doplňovania s názvom TOLERANCE, ktorá udáva maximálny počet

poškodených hodín, ktoré budú touto funkciou doplnené. Hodnota argumentu RECORD_STEP by mala byť opäť totožná s hodnotou, ktorá bola zadaná pri tvorbe časových tabuliek „čistých“ a vypustených hodnôt pomocou funkcie *create_values*. Funkcia rozoznáva medzi 5 rôznymi scenármi, resp. príčinami toho, prečo v danej časovej oblasti meranie nastal výpadok. Následne táto funkcia podľa typu scenára automaticky reaguje doplnením údajov pomocou metódy pohyblivého priemeru (z angl. Moving Average). Pohyblivý priemer je technika na získanie celkovej predstavy o trendoch v množine údajov s časovou značkou. Je to vypočítaný priemer z akejkoľvek podskupiny numerických hodnôt, ktorá sa nazýva perióda (na Obr. 10 označená písmenom *k*).



Obr. 10 Princíp získavania nových numerických hodnôt pomocou pohyblivého priebehu

Pohyblivý priemer môžeme vypočítať pre akékoľvek časové obdobie, resp. periódu. Napríklad, ak máme údaje o meraniach za 24 hodín, môžeme vypočítať päťminútový pohyblivý priemer, štvorminútový pohyblivý priemer, trojminútový pohyblivý priemer a tak ďalej.[23] Šírka tejto periódy je udávaná v sekundách. Čím je šírka periódy pohyblivého priemeru menšia, tým viac sa nové numerické hodnoty výsledného signálu podobajú na okolité numerické hodnoty. Ak je definovaná šírka periódy príliš nízka, funkcia nemá dosah na hodnoty, ktoré sa nachádzajú uprostred úseku chýbajúcich hodnôt. Na druhej strane, ak je definovaná šírka periódy príliš vysoká, doplnené hodnoty sa prestanú podobať na okolité hodnoty pôvodného signálu a začnú na grafe skôr pripomínať lineárny priebeh. Z tohto dôvodu je pre každý časový úsek chýbajúcich hodnôt automaticky zvolená najmenšia prípustná perióda pohyblivého priemeru.

Existuje 5 najčastejších scenárov chýbajúcich hodnôt, ktoré boli identifikované autorom práce. V prvom scenári ide o prípad, kedy nebola v danom časovom intervale nameraná ani jedna hodnota. V takomto prípade nie je čo dopĺňať a úsek nových časových hodnôt ostáva prázdny.

V druhom scenári ide o prípad, kedy sa všetky namerané hodnoty rovnajú jednému a tomu istému číslu a zároveň nebolo meranie počas tohto časového úseku nikde prerušené. V takomto prípade sa do tabuľky MISSING zapíše k tomuto záznamu nereálna hodnota -1 do stĺpca s počtom chýbajúcich hodnôt (stĺpec s názvom NEW_Missing), ktoré slúži len na odlíšenie pre používateľa, že sa jedná o tento špecifický prípad. V treťom scenári ide o prípad podobný prípadu č. 2 avšak s tým rozdielom, že meraný signál nie je konštantný. Ide o špecifický prípad, kedy bol (pravdepodobne) bezchybný časový úsek vyhodnotený ako chybný. Keďže takýto časový úsek nemá zdanlivo žiadne chýbajúce riadky, funkcia nemá čo dopĺňať a posunie tento časový úsek do výstupnej časovej tabuľky tejto funkcie s názvom FILLED. Vo štvrtom scenári ide o špecifický prípad, kedy nastáva zmena zimného času na letný čas. Opačný prípad, kedy nastáva zmena letného času na zimný čas sa nemôže dostať do tabuľky MISSING, pretože do tejto tabuľky sa cez ETL proces dostanú len neúplné časové úseky. To znamená, že by tam musela byť zaznamenaná aspoň jedná hodnota v tom čase. Naopak v tomto prípade je vyhodnotený vždy 2-hodinový časový úsek, pretože posledná hodnota pred zmenou času sa zaznamená vždy o 02:00:01 a končí o 03:00:00. Vďaka posunu o jednu sekundu smerom dozadu sa vyhodnotil tento 1-hodinový časový úsek ako 2-hodinový časový úsek, čo napomáha funkcii identifikovať tento špecifický prípad. Zároveň to znamená, že počet chýbajúcich hodnôt (3600 riadkov) je rovný počtu aktuálnych meraní (prvý riadok + 3599 riadkov, ktoré nasledujú po jednohodinovej absencii hodnôt). Podobne ako v druhom prípade je tento špecifický prípad označený nereálnou hodnotou -2 v stĺpci s počtom chýbajúcich hodnôt, resp. riadkov. Posledný známy scenár je univerzálny scenár, ktorý platí pre všetky ostatné prípady, kedy došlo k inému typu poškodenia ako vyššie spomenutým štyrom typom poškodenia merania. Na takto poškodené časove úseky sa aplikuje metóda automatického dopĺňania chýbajúcich hodnôt pomocou pohyblivého priemeru spomenutá vyššie. Výstupom tejto funkcie sú 2 výstupné argumenty a to rozšírená tabuľka s názvom MISSING a časová tabuľka automaticky doplnených hodnôt s názvom FILLED.

4.2.5. Funkcia *create_statistics*

Funkcia *create_statistics* slúži na vytvorenie tabuľky štatistických parametrov, [24] ktoré sú vypočítané z určitých časových intervalov. Šírka týchto intervalov je určená druhým vstupným argumentom funkcie numerickou hodnotou, ktorá predstavuje počet minút intervalu. Prvým vstupným argumentom funkcie je časová tabuľka s „čistými“ údajmi. Výstupom tejto funkcie je rozsiahla tabuľka štatistických parametrov, ktorú tvorí 18 nasledujúcich stĺpcov:

- DateStart – Začiatok časového intervalu
- DateStop – Koniec časového intervalu
- Mean – Aritmetický priemer časového intervalu

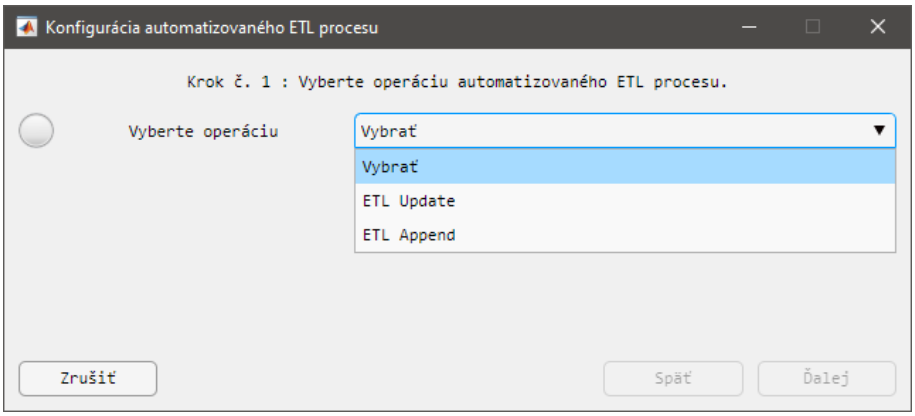
- Mode – Módus časového intervalu
- Median – Medián časového intervalu
- Minimum – Najnižšia nameraná hodnota časového intervalu
- Maximum – Najvyššia nameraná hodnota časového intervalu
- Range – Rozsah nameraných hodnôt časového intervalu
- IQR – Medzikvartilové rozpätie hodnôt časového intervalu
- Variance – Variabilita hodnôt časového intervalu
- SD – Smerodajná odchýlka časového intervalu
- RSD – Relatívna smerodajná odchýlka časového intervalu
- Skewness – Koeficient šikmosti hodnôt časového intervalu
- Kurtosis – Koeficient špicatosti hodnôt časového intervalu
- Equality – Smerodajná odchýlka vypočítaná z hodnôt aritmetického priemeru, modusu a mediánu hodnôt časového intervalu
- SEM – Štandardná chyba aritmetického priemeru hodnôt časového intervalu
- ProbableError - Pravdepodobná chyba aritmetického priemeru hodnôt časového intervalu
- AccuracyRate – Miera presnosti hodnôt časového intervalu

Vypočítané ale aj odvodené hodnoty štatistických parametrov sú v ďalších častiach projektu vizualizované v podobe niekoľkých grafov.

4.2.6. Funkcie *etl_update* a *etl_append*

Funkcie *etl_update* (z angl. Aktualizovať) a *etl_append* (z angl. Pripojiť) sú hlavnými funkciami tejto časti, resp. fázy projektu. Všetky vyššie spomenuté funkcie kap. 4.2 boli vytvorené za účelom zjednodušenia týchto dvoch funkcii, v ktorých sa budú vyššie spomenuté funkcie využívať. Obe funkcie sú si navzájom dosť podobné. Funkcia *etl_update* má za úlohu načítať a uložiť do súborov formátu **.mat* všetky výstupné údaje ETL procesu. V prípade, že už existuje niečo uložené tak funkcia tieto údaje prepíše. Sem patria časové tabuľky „čistých“, vypustených a doplnených údajov roztriedených podľa rokov, tabuľky štatistických parametrov roztriedené podľa rokov a tabuľka záznamov chýbajúcich časových údajov merania. S výnimkou tabuľky záznamov chýbajúcich časových údajov, ktorá je umiestnená v priečinku automaticky upravených riadkov, sú všetky údajové súbory umiestnené v priečinku prípravných údajov. Úlohou funkcie *etl_append* je načítať **.csv* súbory nových zozbieraných údajov a spojiť ich s aktuálne uloženými údajmi. Pre účely interaktívnejšieho a pohodlnejšieho spôsobu používania týchto dvoch funkcií bola vytvorená aplikácia s názvom *ETL_Configuration* znázornená na Obr. 11. Používanie aplikácie je

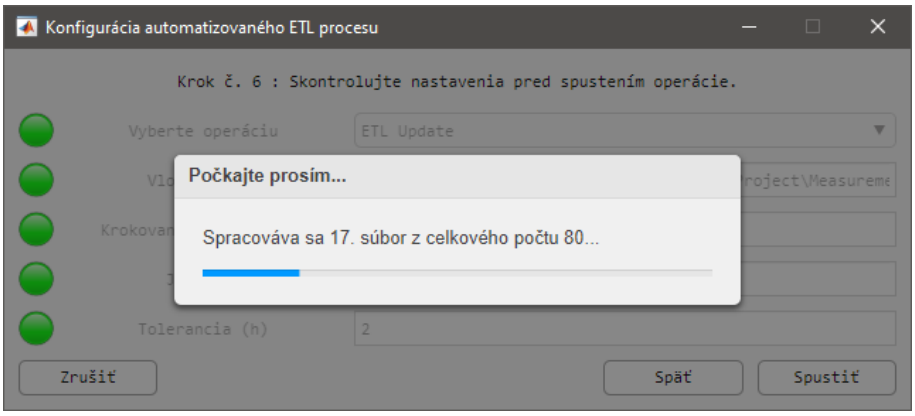
veľmi jednoduché a intuitívne pre používateľa. Aplikácia svojím návrhom pôsobí ako sprievodca pri inštalácii, v ktorom štandardne vystupujú tlačidla ako *Ďalej*, *Späť* a *Zrušiť*.



Obr. 11 Okno aplikácie *ETL_Configuration*

Postup prípravy údajov je rozdelený do šiestich krokov. Prvých 5 krokov sa týka výberu všetkých potrebných vstupných parametrov pre vyššie spomínané funkcie ETL procesu. V prvom kroku aplikácia požaduje od používateľa vybrať si, ktorú operáciu ETL procesu chce vykonať. V závislosti od tohto výberu je používateľ povinný v druhom kroku uviesť celú cestu k priečinku, v ktorom sa nachádzajú zozbierané údaje merania vo formáte **.csv*. V prípade, že si používateľ zvolil operáciu ETL Append, má aj možnosť uviesť cestu k súboru, to znamená celú cestu vrátane názvu a prípony konkrétneho **.csv* súboru. V treťom kroku používateľ uvedie numerickú hodnotu, ktorá reprezentuje konštantný časový odstup jednotlivých meraní v sekundách. Pokiaľ sa v stĺpci zozbieraných údajov s numerickými hodnotami nachádza aj jednotka, resp. akýkoľvek iný textový údaj podobne ako to je vidieť na Obr. 8, používateľ uvedie túto jednotku v štvrtom kroku. Posledný údaj, ktorý potrebuje uviesť používateľ je numerická hodnota, ktorá reprezentuje maximálny počet po sebe idúcich hodín, resp. hodinových úsekov. Všetky prerušené časove úseky, ktorých dĺžka v hodinách je menšia alebo rovná zadanej tolerancii budú automaticky doplnené.

V poslednom kroku č. 6 aplikácia požiada používateľa o kontrolu údajov pred spustením zvolenej ETL operácie. Následne tlačidlom *Spustiť* sa spustí operácia, ktorá v závislosti od množstva údajov vstupujúcich do operácie môže trvať aj niekoľko hodín. Počas behu operácie je zobrazené dialógové okno, ktoré informuje používateľa o konkrétne prebiehajúcom procese (Obr. 12). Po dokončení operácie sú pripravené všetky potrebné údaje pre analýzu údajov.



Obr. 12 Okno spusteného procesu v aplikácii *ETL_Configuration*

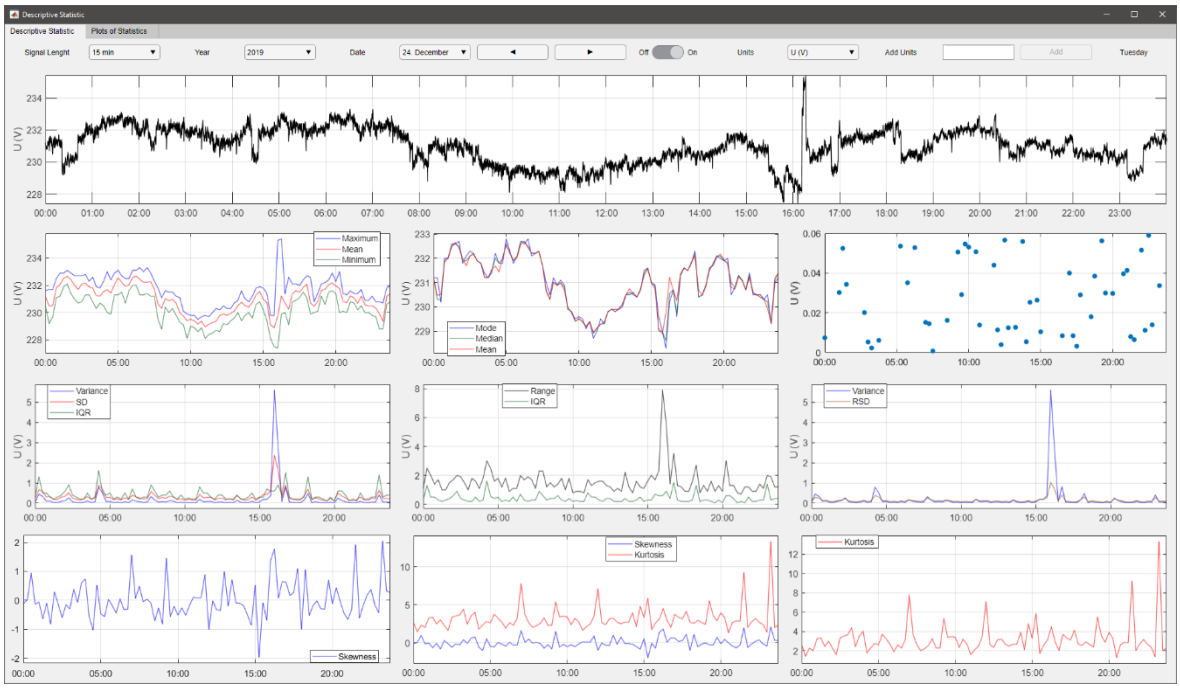
Po dokončení procesu je možné aplikáciu uzavrieť tlačidlom *Zrušiť*, alebo vrátiť sa naspäť, rovnakým postupom vykonať úpravy vo vstupných parametroch zvolenej operácie a spustiť ďalšiu operáciu ETL procesu.

4.3. Analýza a vizualizácia údajov

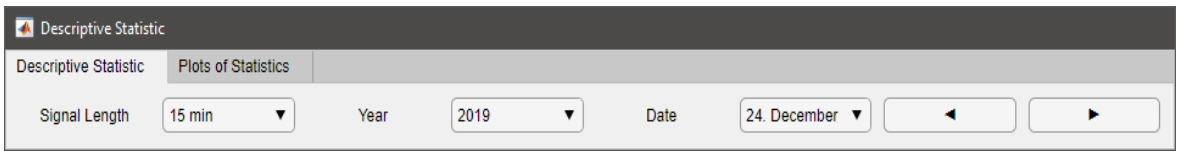
V tejto kapitole budú podrobne opísané aplikácie, ktoré boli vytvorené za účelom dokonalej analýzy našich spracovaných údajov. Vytvorené boli celkovo 3 aplikácie pomocou vývojového štúdia *App Designer*. Keďže analýza a vizualizácia údajov úzko navzájom súvisia a prelínajú sa, bude v tejto časti poukázané aj iné možnosti vizualizácie údajov prostredníctvom jednoduchých funkcií.

4.3.1. Grafy štatistických parametrov

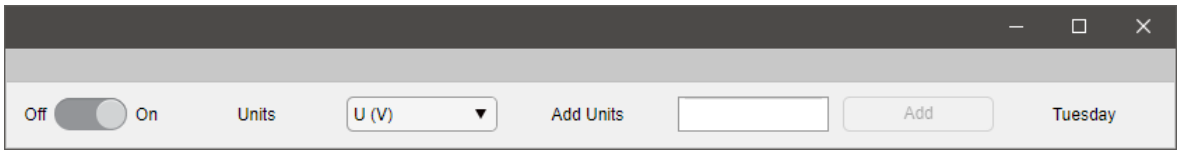
Prvá v poradí vytvorená aplikácia s názvom *Descriptive_Statistics.mlapp* slúži na vizualizáciu údajov meraného signálu celého zvoleného dňa a taktiež na vizualizáciu štatistických parametrov zvoleného dňa. Aplikácia je zložená z dvoch hlavných režimov zobrazení, ktoré sú oddelené záložkami a sú navzájom nezávislé od seba. V prvej záložke sú znázornené štatistické parametre popisnej štatistiky [25] a v druhej záložke sa nachádza zobrazenie niekoľkých ďalších štatistických parametrov, ktoré boli prevažne odvodené z parametrov vyobrazených v prvej záložke. Ukážka otvorenej aplikácie sa nachádza na obrázku nižšie. V prvej záložke s názvom *Descriptive Statistics* (z angl. Popisná štatistika), ako je možné vidieť na Obr. 13 je znázornený signál údajov meraného dňa, ktorý bol používateľom zvolený pomocou ovládacieho panela aplikácie. Tento ovládací panel je umiestnený v hornej časti aplikácie a bude veľmi podobný vo všetkých ďalších aplikáciách. Bližšie znázornený ovládací panel na Obr. 14 a Obr. 15 umožňuje používateľovi aplikácie zvoliť požadovaný deň merania a to pomocou rozbaľovacích zoznamov.



Obr. 13 Ukážka prvej záložky aplikácie *Descriptive_Statistics*



Obr. 14 Pravá polovica ovládacieho panela



Obr. 15 Ľavá polovica ovládacieho panela

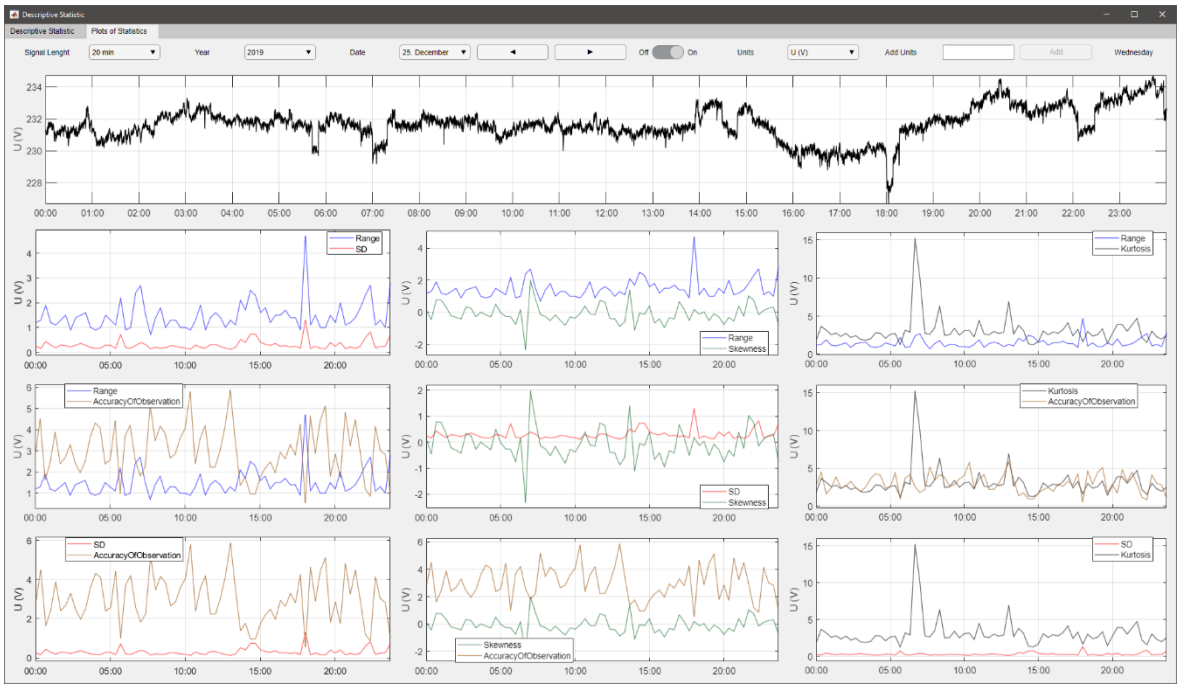
Tato aplikácia ale aj ďalšie aplikácie sú navrhnuté tak, aby boli čo najviac intuitívne pre používateľa. Ako prvý objekt ovládacieho panela zľava je rozbaľovací zoznam s názvom *Signal Length* (z angl. Dĺžka signálu), kde má používateľ na výber spomedzi viacerých možnosti zvoliť dĺžku signálu. Používateľ má k dispozícii na výber 10, 12, 15, 20, 30 a 60-minútové dĺžky signálov. Zvolený deň, resp. údaje namerané v ten deň sú následne rozdelené na krátke časové úseky, ktorých dĺžka je definovaná týmto rozbaľovacím zoznamom. Z týchto malých vzoriek sú vypočítané štatistické parametre, ktoré sú vykreslené v jednotlivých grafoch. Tento rozbaľovací zoznam je zároveň jediná vec, ktorú používateľ vidí pri spustení aplikácie. Ďalší rozbaľovací zoznam s názvom *Year* (z angl. Rok) sa zobrazí až keď si používateľ zvoli dĺžku signálov. Slúži na výber roku, v ktorom sa chystá používateľ analyzovať namerané údaje. Na výber má používateľ len tie roky, ktoré boli ETL procesom tohto projektu automaticky vyhodnotené ako „čisté“ údaje. Ďalším objektom je

rozbaľovací zoznam s názvom *Date* (z angl. Dátum) sa zobrazí až keď si používateľ vyberie rok. Používateľovi sa znovu zobrazia na výber len tie dni zvoleného roku, ktoré boli ETL procesom vyhodnotené ako „čisté“ údaje. Po zvolení ľubovoľného dňa sa používateľovi zobrazia všetky ostatné objekty aplikácie tak, ako sú ilustrované aj na Obr. 13. Za týmto rozbaľovacím zoznamom nasledujú tlačidla označené šípkami vľavo a vpravo. Tieto šípky slúžia na zmenu zobrazenia štatistickej analýzy údajov meraného dňa na predchádzajúci (šípka vľavo) alebo na nasledujúci (šípka vpravo) deň. V prípade, že sa používateľ dostane až na úplne prvý deň merania, šípka vľavo sa stane neaktívna, čo znamená, že sa na ňu nebude dať kliknúť. Podobne je aplikácia ošetrená voči chybovým hláseniam aj z druhej strany, čo znamená, že tlačidlo vpravo sa zablokuje vtedy, ak narazí používateľ na úplne posledný deň merania. Za týmito tlačidlami nasleduje prepínač s dvomi možnosťami: *On* a *Off*. Ak používateľ premiestni kurzor na tento prepínač, zobrazí sa mu po anglicky vysvetlivka *Show Legends*, čo v preklade znamená, že tento prepínač slúži na zobrazenie legiend k jednotlivým grafom. Pri spustení aplikácie sú legendy vypnuté, pretože občas sú krivky grafov rozvrhnuté na zobrazovacej ploche grafu tak, že zakrývajú krivky grafov. To je zároveň dôvod, prečo sa v tejto aplikácii nachádza tento prepínač. Prepínač pri nastavení do polohy *On* zobrazí legendy pre všetky grafy okrem grafu signálu a bodového grafu, ktorý nezobrazuje žiadny štatistický parameter, ale vzájomnú polohu troch štatistických parametrov. O tomto a ďalších grafoch bude bližšie popísané v ďalšej časti tejto kapitoly. Pri nastavení prepínača do polohy *Off* sa všetky legendy zakryjú. Za prepínačom nasleduje ďalší rozbaľovací zoznam s názvom *Units* (z angl. Jednotky). Tento zoznam umožňuje používateľovi prepnúť jednotky, v ktorých boli údaje namerané. Zoznam ponúka na výber len jednotky elektrického napätia *U* (V), jednotky teploty *T* (°C) a jednotky relatívnej vlhkostí vzduchu *RH* (%). V prípade, že chce používateľ analyzovať signály inej jednotky, resp. meranej veličiny udávanej v iných jednotkách, má možnosť rozšíriť zoznam jednotiek pomocou textového poľa s názvom *Add Units* (z angl. Pridať Jednotky), ktoré sa nachádza hneď vedľa zoznamu jednotiek. Zatlačením tlačidla *Add*, ktoré sa nachádza hneď vedľa textového poľa *Add Units* sa pridá jednotka zapísaná v tomto textovom poli do zoznamu jednotiek a používateľ má možnosť vybrať si túto pridanú jednotku zo zoznamu *Units*. Posledný objekt na ovládacom paneli je *Weekday* (z angl. Deň v týždni). Nachádza sa v pravom hornom rohu aplikácie a slúži len na informáciu používateľa o tom, ktorý deň v týždni je práve teraz zobrazený v aplikácii. Deň v týždni a dátumy sa zobrazuje len v anglickom jazyku, čo je zároveň dôvod toho, prečo sa autor práce rozhodol ponechať rozhranie aplikácie v anglickom jazyku.

Aplikácia zobrazuje v hornej oblasti pod ovládacím panelom po celej svojej šírke úplný signál zvoleného dňa. Ak ETL proces nevyhodnotil ako „čisté“ údaje všetky hodiny zvoleného dňa, tak chýbajúce hodiny uprostred „čistých“ údajov budú z analýzy vynechané a na grafe sa prejavia ako

rovné úsečky v chýbajúcom intervale hodín. Ak sa takéto chýbajúce hodiny vyskytujú od polnoci do určitej hodiny času dňa alebo od určitej hodiny dňa do polnoci nasledujúceho dňa, tak sa signál automaticky zobrazí tak, aby zamedzil plytvaniu prázdnej plochy z ľavej, resp. z pravej strany, ktorá by sa v opačnom prípade objavila na začiatku alebo na konci grafu signálu meraného dňa. Podobné správanie vykazujú aj grafy štatistických parametrov umiestnené pod signálom.

Grafy štatistických parametrov prvého okna s názvom *Descriptive Statistics* zobrazujú štatistické parametre vypočítané z po sebe idúcich vzoriek signálu, ktorých časový interval bol zvolená na začiatku. Grafy štatistických parametrov sú zobrazené v matici, ktorá má 3 stĺpce a 3 riadky. V prvom riadku sú vykreslené kombinácie vypočítaných štatistických parametrov popisnej štatistiky, ktoré popisujú centrálnu tendenciu rozdelení. Tu patria štatistické parametre minimum, maximum, modus, medián a aritmetický priemer. Posledný graf v tomto riadku je ako jediný graf bodový. Na tomto grafe sú vykreslené body, ktoré odzrkadľujú vzájomnú polohu aritmetického priemeru, modusu a mediánu. Tieto hodnoty boli vypočítané pomocou smerodajnej odchýlky uplatnenej na spomínanú trojicu štatistických parametrov, vďaka čomu vieme zistiť ako veľmi sú tieto parametre podobné, resp. rovnaké. Tento graf je taktiež orezaný z hornej strany, vďaka čomu nám ukáže len tie hodnoty, ktoré sú menšie ako 0,06 jednotiek. Je to z toho dôvodu, že nás budú zaujímať hlavne také vzorky signálov, ktoré majú takmer (ak nie úplne) rovnaké štatistické parametre centrálnej tendencie. V druhom riadku sú vykreslené kombinácie, ktoré popisujú variabilitu rozdelení. Tu patria štatistické parametre ako smerodajná odchýlka, relatívna smerodajná odchýlka, koeficient variácie, medzikvartilný rozsah a rozsah hodnôt. V treťom riadku sa nachádzajú kombinácie štatistických parametrov, ktoré popisujú tvar rozdelení (vzoriek). Sem patrí koeficient špicatosti a koeficient šikmosti. Všetky tieto koeficienty boli vypočítané z rozdelení, resp. vzoriek celkového signálu zvoleného dňa, a jednotlivé hodnoty týchto štatistických parametrov boli následne vykreslené na grafy. Grafy štatistických parametrov druhej záložky s názvom *Plots of Statistics* (z angl. Grafy Štatistík) ilustrované na Obr. 16 zobrazujú štatistické parametre vypočítané z po sebe idúcich vzoriek signálu, ktorých šírka bola zvolená na začiatku rovnako ako v prípade zobrazenia v prvej záložke.

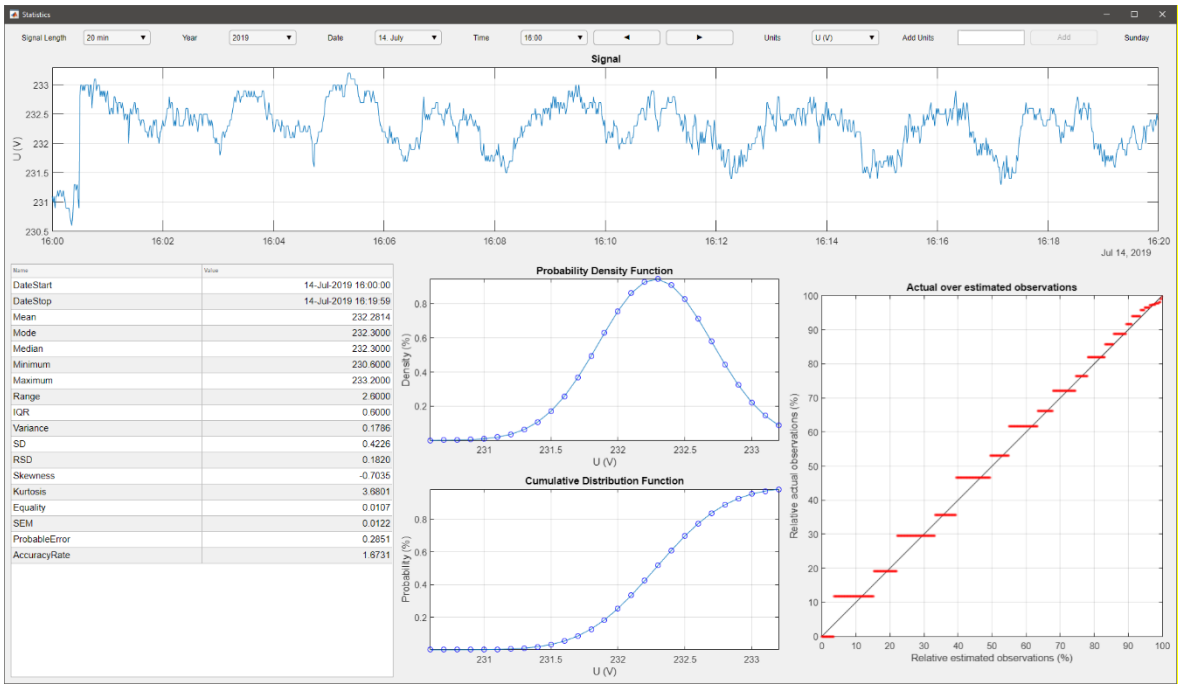


Obr. 16 Ukážka druhej záložky aplikácie *Descriptive_Statistics*

Na rozdiel od prvého okna sú v tomto okne zobrazené prevažne odvodené štatistické parametre. Tieto parametre boli odvodené z parametrov vyobrazených v prvom okne. Podobne ako v prvom okne sa v tomto okne pod grafom signálu nachádza matica grafov o veľkosti troch stĺpcov a troch riadkov. V každom grafe je vykreslená jedinečná dvojica výpočtom odvodených štatistických parametrov alebo inej doposiaľ nevykreslenej kombinácie štatistických parametrov z prvej záložky. V prvom riadku sa vľavo nachádza dvojica štatistických parametrov rozsah a smerodajná odchýlka, v strede dvojica rozsah a koeficient špicatostí a vpravo dvojica rozsah a koeficient špicatosti. V druhom riadku sa vľavo nachádza dvojica rozsah a miera presnosti, v strede dvojica smerodajná odchýlka a koeficient šikmosti a vpravo dvojica koeficient špicatosti a miera presnosti. V treťom riadku sa vľavo nachádza dvojica smerodajná odchýlka a miera presnosti, v strede dvojica koeficient šikmosti a miera presnosti a vpravo dvojica smerodajná odchýlka a koeficient špicatosti.

4.3.2. Štatistická analýza signálov

Táto časť analýzy údajov sa zaoberá bližším náhľadom na jednotlivé vzorky signálov meraného sieťového napätia. Prostredníctvom aplikácie s názvom *Statistics_Analysis* je používateľovi umožnené analyzovať nielen signály sieťového napätia ale prakticky všetky údaje, ktoré rešpektujú formát opísaný v kap. 4.1. K tomu aby táto aplikácia fungovala správne, sa musí v priečinku s pripravenými údajmi nachádzať výstup ETL procesu kap. 4.2.6. Na Obr. 17 je možné vidieť náhľad zobrazenia analýzy náhodného signálu merania sieťového napätia.

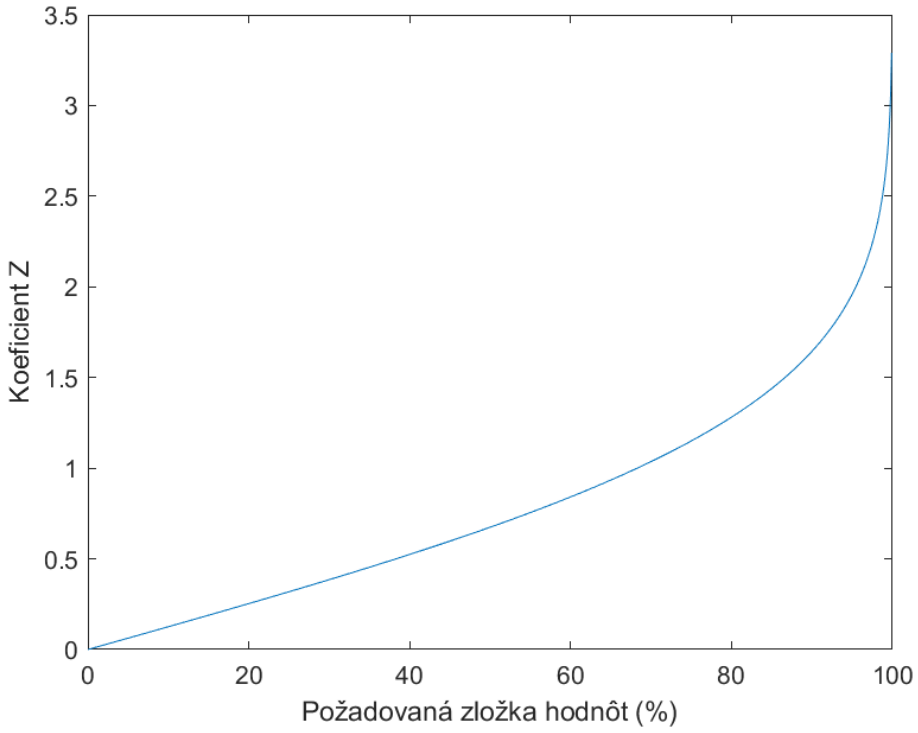


Obr. 17 Ukážka okna aplikácie *Statistics_Analysis*

Rozloženie komponentov tejto aplikácie je podobné ako v prípade prvej aplikácie určenej na ilustráciu štatistických parametrov vypočítaných z nameraných údajov. V hornej časti je pozdĺž celej aplikácie umiestnený hlavný ovládací panel. Tento panel má rovnaký účel ako v prípade prvej aplikácie. V tejto aplikácii pribudol na ovládacom paneli nový rozbaľovací list s názvom *Time* (z angl. Čas) a ubudol prepínač na voľbu viditeľnosti legiend jednotlivých grafov zobrazenia. Nový rozbaľovací list s názvom *Time* umožňuje používateľovi upresniť výber konkrétnej vzorky signálu meraného dňa, ktorá sa následne podrobí detailnej analýze. V oblasti pod ovládacím panelom sa nachádza ilustrácia časového priebehu zvolenej vzorky meraných údajov modrou krivkou označený názvom *Signal*. V ľavej dolnej časti zobrazenia je umiestnená tabuľka štatistických parametrov, ktoré boli vyextrahované z tabuľky štatistických parametrov pre práve zvolenú vzorku. Po pravej strane tabuľky sú umiestnené ilustrácie distribučných funkcií. Ilustrácia s názvom *Probability Density Function* (skrátene PDF) je funkciou hustoty pravdepodobnosti rozdelenia. Krivka tohto grafu ilustruje hustotu jednotlivých nameraných hodnôt analyzovaného signálu vzorky v zadanom časovom intervale. Pod týmto grafom je umiestnená kumulatívna distribučná funkcia s názvom *Cumulative Distribution Function* (skrátene CDF). Krivka tohto grafu ilustruje pravdepodobnosť toho, že náhodne zvolená hodnota v rozsahu nameraných hodnôt je menšia ako hodnota prislúchajúca nezávislej premennej. Ilustrácia umiestnená v pravej dolnej časti aplikácie s názvom *Actual over estimated observations* vznikla pomerne zložitým procesom.

Zo zvolenej vzorky nameraných údajov bol najprv vypočítaný aritmetický priemer a smerodajná odchýlka. Tieto hodnoty sa okrem iného nachádzajú umiestnené aj v tabuľke štatistických

parametrov, ktorá je umiestnená v ľavej dolnej časti aplikácie. V štatistike existuje taký koeficient, ktorým ak vynásobíme hodnotu smerodajnej odchýlky a túto hodnotu odčítame, resp. pripočítame k hodnote aritmetického priemeru, získavame ľavú a pravú hranicu tzv. intervalu spoľahlivosti. Ak zoberieme všetky namerané hodnoty vzorky, z ktorej bol vypočítaný aritmetický priemer a smerodajná odchýlka a zároveň sa nachádzajú uprostred tohto intervalu, zistíme, že sa v intervale nachádza len určitá percentuálna zložka celého súboru nameraných hodnôt. Vo všeobecnosti platí, že výsledná percentuálna zložka hodnôt je závislá od veľkosti tohto koeficientu, ktorý sa označuje v štatistike ako koeficient Z. Táto skutočnosť logicky musí platiť aj v opačnom zmysle (Obr. 18).

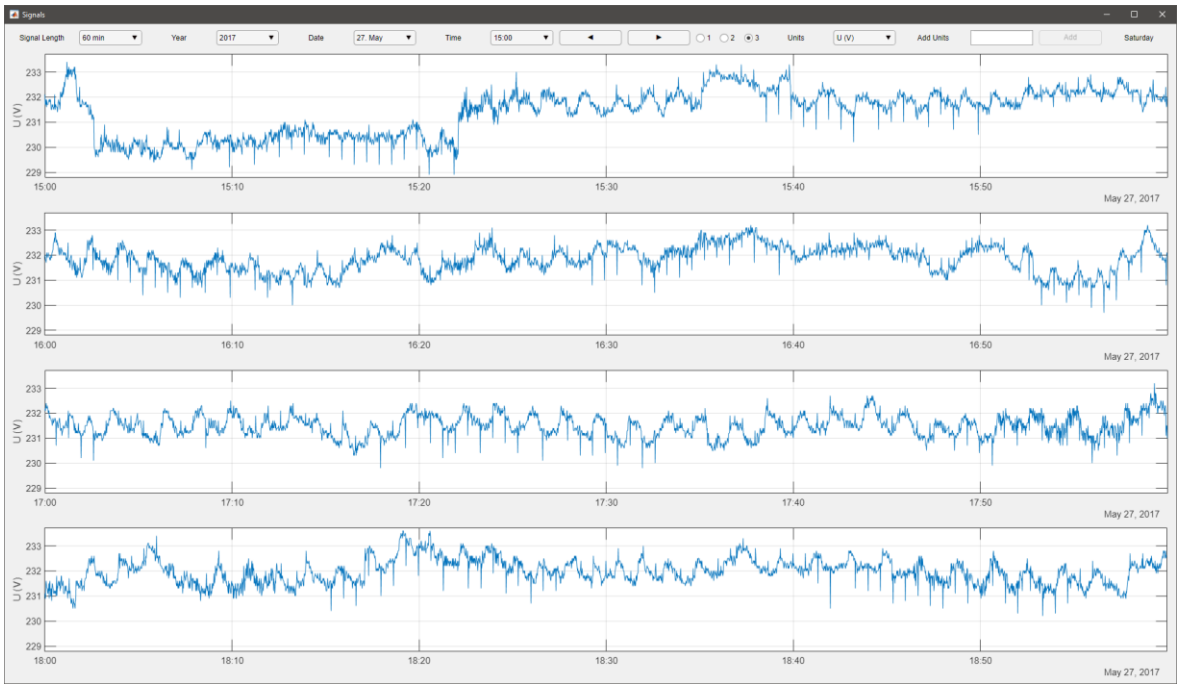


Obr. 18 Ilustrácia závislosti hodnoty koeficientu Z od požadovanej percentuálnej zložky hodnôt

To znamená, že vieme tento koeficient vypočítať na základe požadovanej percentuálnej zložky. Vtedy očakávame, že zvolením vypočítaného koeficientu Z získame také hranice intervalu spoľahlivosti, v ktorom bude ležať požadovaná percentuálna zložka všetkých nameraných hodnôt. Tento odhad intervalu spoľahlivosti funguje dokonalé v teoretickej oblasti štatistiky. V praxi tomu tak nemusí byť vždy a spravidla tomu tak nie je nikdy. Graf, ktorý je umiestnený v ľavom dolnom rohu aplikácie ilustruje túto skutočnosť. Nezávislou premennou tohto grafu je požadovaná percentuálna zložka hodnôt a závislou premennou je skutočná percentuálna zložka hodnôt, ktorá sa nachádza vo vypočítaných intervaloch spoľahlivosti. Čierna úsečka, ktorá je diagonálne vedená grafom je referenčná teoretická úsečka. Červené body grafu ilustrujú skutočný počet vyskytujúcich sa hodnôt v intervale pre každé požadované percento hodnôt.

4.3.3. Signály meranej veličiny

Zaujímavé postrehy sa dajú vyhľadať aj v samotných ilustráciách časových priebehov, resp. signálov nameraných údajov. V závislosti od tvaru signálu vieme niekedy skonštatovať, resp. odhadnúť možné príčiny tohto tvaru. Prostredníctvom interaktívnej aplikácie s názvom *Signals* má používateľ prehliadať signály nameraných údajov. V aplikácii je možné zobraziť v jeden čas až štyri po sebe idúce časové priebehy nameraných údajov o konštantnej dĺžke z hľadiska času. Vzor okna spustenej aplikácie pre 60-minútové intervaly je znázornený na Obr. 19.



Obr. 19 Ukážka okna aplikácie *Signals*

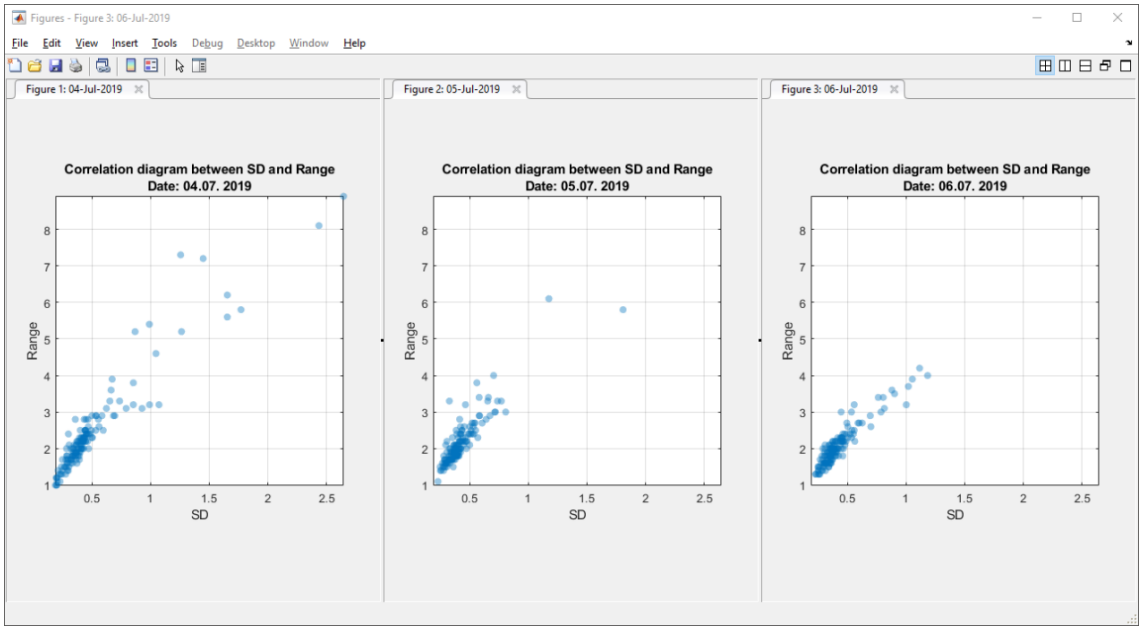
Opäť je v hornej časti aplikácie umiestnený hlavný ovládací panel, z ktorého vieme presne špecifikovať začiatok prvého časového priebehu. Oproti hlavnému panelu aplikácie *Statistics_Analysis*, opísanej v kap. 4.3.2, je hlavný panel tejto aplikácie rozšírený o výber spomedzi troch možností. Tieto možnosti ponúkajú používateľovi tri režimy zobrazenia časových priebehov. Zmenou týchto možností používateľ nariaďuje aplikácii zmeniť nastavenie hraníc zobrazenia vertikálnej osi pre jednotlivé grafy. Režimy zobrazenia sú síce v aplikácii kvôli kompaktnosti označené len číslami *1*, *2* a *3* ale ponúkajú vysvetlivku v anglickom jazyku pri presunutí kurzora na oblasť výberu týchto možností. Pri spustení aplikácie je automaticky nastavená ako predvolená prvá možnosť. Ak je zvolená prvá možnosť, vertikálna os každého ilustrovaného signálu začína od najmenšej nameranej hodnoty a končí na najväčšej nameranej hodnote dňa, ktorého sa daný časový priebeh týka. To znamená, že ak sú práve zobrazené všetky štyri signály toho istého dňa, potom budú mať všetky signály spoločné hranice vertikálnych osí. Ak je zvolená druhá možnosť, každý graf signálu má nastavenú svoju vlastnú minimálnu a maximálnu nameranú hodnotu ako

hranicu svojich vlastných vertikálnych osí. V prípade, že je zvolená tretia možnosť, všetky štyri grafy majú nastavený rovnaký začiatok a koniec vertikálnej osi, pričom spodná hranica je rovná najmenšej nameranej hodnote a horná hranica je rovná najväčšej nameranej hodnote zistenej z údajov práve zobrazených štyroch grafov signálov. Všetky 4 ilustrácie teda budú mať vždy spoločné hranice vertikálnych osí. Zmysel a funkčnosť ostatných komponentov hlavného ovládacieho panela je rovnaký ako v predchádzajúcich aplikáciách.

4.3.4. Korelačné diagramy štatistických parametrov

Funkcia *plot_correlation_diagram* poskytuje používateľovi zaujímavý náhľad na údaje z hľadiska štatistických parametrov. Táto vizualizácia ilustruje závislosť medzi dvomi štatistickými parametrami, ktoré boli vypočítané zo vzoriek údajov o určitej konštantnej dĺžke v priebehu určitého dňa a to spôsobom, ktorý je popísaný v kap. 4.2.5. Na Obr. 20 je možné vidieť výsledok funkcie, ktorý dostávame pri volaní funkcie s nasledujúcimi vstupnými parametrami funkcie:

```
plot_correlation_diagram(stats_10_2019, {'SD', 'Range'}, [4, 7], 3, 1);
```



Obr. 20 Ukážka troch korelačných diagramov rozsahu hodnôt a smerodajnej odchýlky

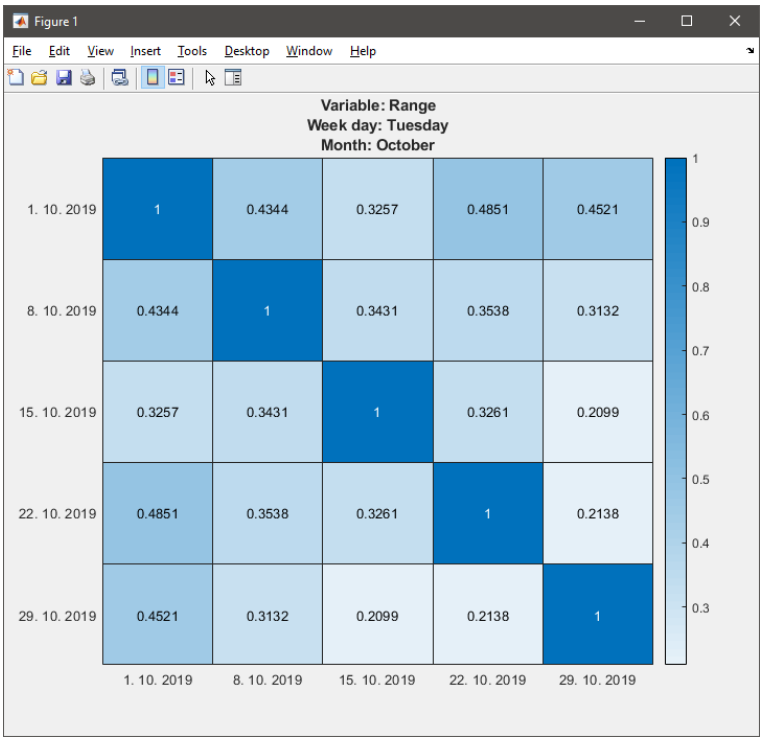
Funkcia má 5 vstupných argumentov. Prvým argumentom (`stats_10_2019`) je tabuľka štatistických parametrov, ktorú si môže používateľ zvoliť z ľubovoľného roka a s ľubovoľným vzorkovaním. Druhým argumentom (`{'SD', 'Range'}`) je bunka dvoch názvov stĺpcov tejto tabuľky, ktoré budú voči sebe navzájom graficky porovnané. Tretím argumentom (`[4, 7]`) je pole dvoch numerických hodnôt, pričom prvá hodnota predstavuje deň a druhá mesiac v roku. Od tohto dňa vrátane sa začnú vykresľovať bodové grafy, ktoré poznáme aj ako korelačné diagramy. Počet týchto diagramov je určený štvrtým vstupným argumentom (`3`), pričom najviac je možné vykresliť až 8 grafov súčasne.

Posledným argumentom (`1`) sa používateľ rozhoduje medzi dvomi režimami zobrazenia. Ak je zadaná *0* alebo *false*, každý graf má svoje vlastné hranice zobrazenia. Ak je parameter rovný číslu *1*, resp. *true*, hranice všetkých grafov sú rovnaké a to v závislosti od najväčšej zaznamenanej hodnote zvoleného dňa. Funkcia nemá žiadne výstupné parametre. Jej použitie je vhodné a užitočné pri skúmaní podobnosti dvoch ľubovoľných štatistických parametrov zvoleného dňa.

4.3.5. Korelačná matica ľubovoľného štatistického parametra

Kým prechádzajúca funkcia ilustruje podobnosť medzi hodnotami dvoch odlišných štatistických parametrov, funkcia s názvom *plot_correlation_weekdays* ilustruje podobnosť medzi viacerými rovnakými štatistickými parametrami. Výsledkom tejto funkcie je dvojdimenzionálna teplotná mapa (z angl. Heat Map), resp. korelačná matica štatistického parametra, ktorý bol vypočítaný z údajov rovnakých dní v týždni zvoleného mesiaca. Na Obr. 21 je uvedený príklad použitia tejto funkcie. Syntax príkazu pre vyššie ilustrovanú maticu je nasledovný:

```
plot_correlation_weekdays(stats_10_2019, 'Range', [2, 10]);
```



Obr. 21 Ukážka korelačnej matice rozsahu hodnôt všetkých utorkov mesiaca október v roku 2019

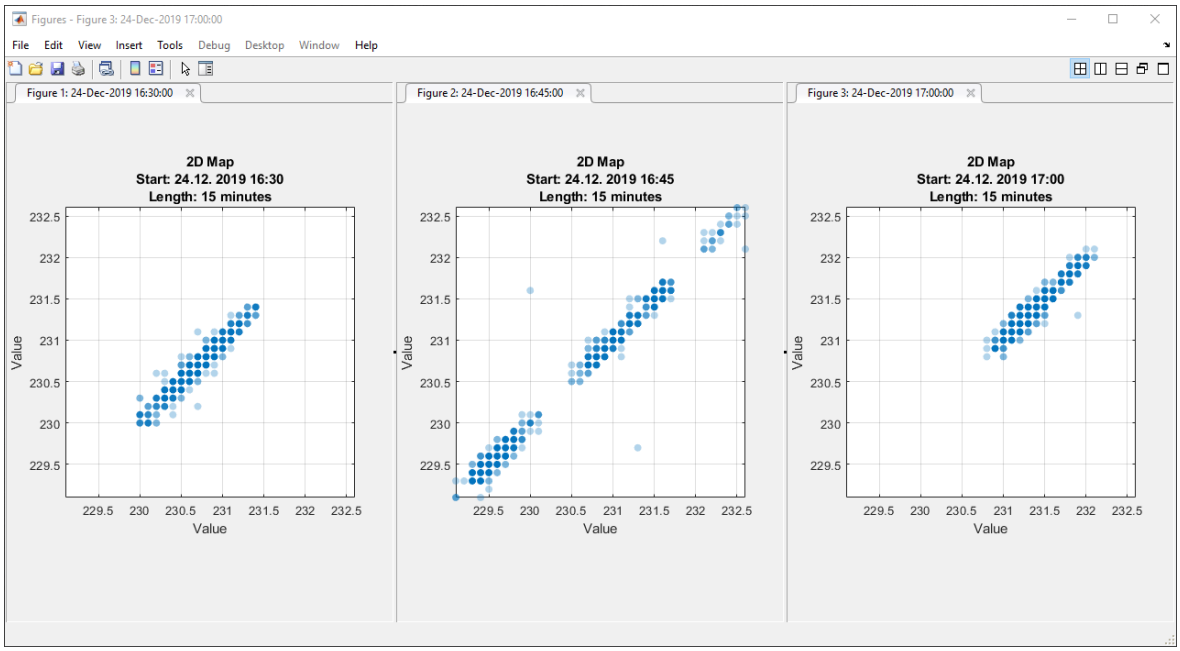
Funkcia *plot_correlation_weekdays* má tri vstupné argumenty. Aj v tomto prípade je prvým argumentom (`stats_10_2019`) tabuľka štatistických parametrov. Opäť platí, že šírka intervalu vzorkovania údajov a rok je uvedený v názve tejto tabuľky. Druhým vstupným argumentom (`'Range'`) používateľ vyberá štatistický parameter, resp. názov stĺpca tabuľky. Tretí argument (`[2,`

`10`]) je pole dvoch numerických hodnôt, kde prvá hodnota reprezentuje deň v týždni a druhá hodnota reprezentuje mesiac v roku. Bunky vypočítaných korelácií sú vyfarbené modrou farbou. Pri vyššej podobnosti signálov je intenzita odtieňu modrej farby vyššia, čím optický napomáha k rýchlejšej identifikácii navzájom podobných signálov rovnakého štatistického parametra nameranom v odlišnom čase.

4.3.6. Dvojrozmerné mapy výskytu hodnôt

Ďalšou funkciou, ktorá ponúka používateľovi zaujímavý spôsob, resp. uhol pohľadu na údaje je funkcia s názvom *plot_signal_2D_maps*. Táto funkcia ilustruje hodnoty ľubovoľne zvoleného časového úseku merania položené voči tým istým hodnotám posunutých o jeden riadok dopredu. Takýmto spôsobom sa nám zobrazí dvojrozmerná mapa výskytu hodnôt v zadanom časovom úseku, ktorá poukazuje na veľkosť zmeny signálu. Inými slovami tieto mapy ilustrujú netradičným spôsobom funkciu prvej derivácie signálu nameraných hodnôt. Na Obr. 22 je ilustrácia troch po sebe idúcich časových úsekov merania efektívnej hodnoty sieťového napätia. Táto ilustrácia je výsledkom nasledujúceho príkazu:

```
plot_signal_2D_maps(values_clean_2019, 15, [24, 12, 16, 30], 3, 1);
```



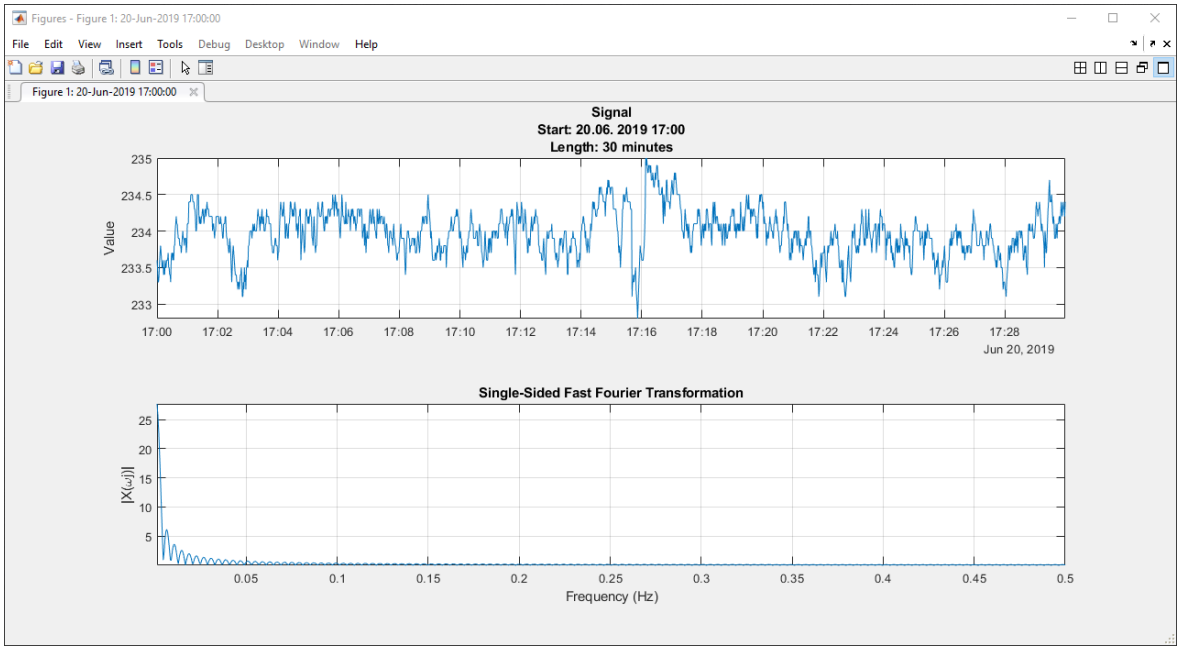
Obr. 22 Ukážka troch 2D máp hodnôt sieťového napätia

Funkcia má 5 vstupných parametrov. Prvým vstupným argumentom (`values_clean_2019`) je časová tabuľka „čistých údajov“ merania. Druhým vstupným argumentom (`15`) používateľ rozhoduje o konštantnej dĺžke časového intervalu, z ktorého budú vykreslené po sebe idúce dvojrozmerné mapy. Tretím argumentom (`[24, 12, 16, 30]`) určí používateľ deň a presný čas, od ktorého začne vykresľovanie prvej mapy. Štvrtým argumentom (`3`) určí počet máp, ktoré majú byť

vykreslené. Posledným argumentom (`1`) rozhoduje používateľ o tom, či majú mať všetky mapy totožné hranice osí zobrazenia hodnotou *1* alebo hodnotou *true*. V opačnom prípade budú mať jednotlivé mapy nastavené svoje vlastné hranice osí.

4.3.7. Grafy rýchlej Fourierovej transformácie (FFT) signálov

Pri analýze údajov s časovou značkou, resp. údajov nameraných v čase je v niektorých prípadoch vhodné prešetriť tieto údaje pomocou funkcie rýchlej Fourierovej transformácie.[26] Pomocou tejto vizualizácie vieme zistiť, či má nameraný signál periodický charakter pri niektorých konkrétnych frekvenciách.



Obr. 23 Ukážka 30-minútového signálu nameraných hodnôt sieťového napätia a jeho rýchlej Fourierovej transformácii

Na Obr. 23 je znázornený výstup funkcie s názvom *plot_signal_FFT*, ktorý je zložený z dvoch grafických prvkov. V hornej časti je umiestnený časový priebeh signálu a v dolnej časti je umiestnený výsledok funkcie rýchlej Fourierovej transformácie tohto signálu. Vykreslená je len jedna polovica úplného výsledku funkcie rýchlej Fourierovej transformácie a to z toho dôvodu, že obe polovice nadobúdajú identické hodnoty, pričom druhá polovica je súmerná podľa vertikálnej osi. V tomto konkrétnom prípade bol prešetrený signál merania efektívnej hodnoty sieťového napätia dňa 20. 6. 2019 v čase od 17:00 do 17:30. Z tejto ilustrácie vieme usúdiť, že meraný signál nevykazuje výrazný periodický charakter v žiadnej časti dostupného frekvenčného spektra. Tento kombinovaný graf bol vykreslený po vpísaní nasledujúceho príkazu do príkazového riadku programu MATLAB:

```
plot_signal_FFT(values_clean_2019, 30, 1, [20, 6, 17, 0], 1);
```

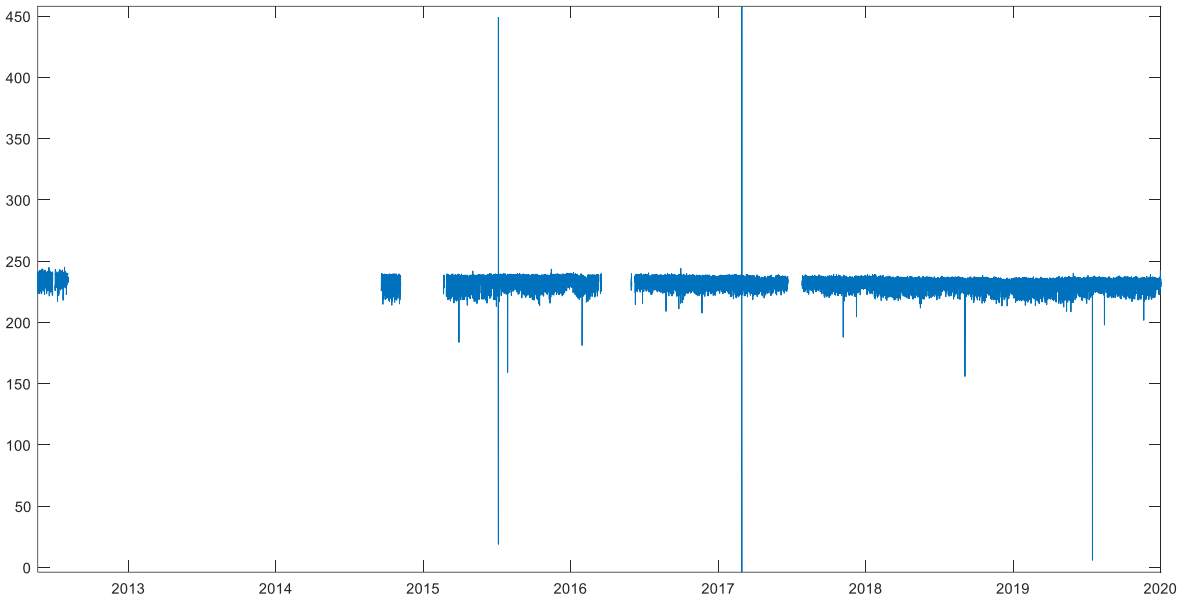
Pre docielenie tejto ilustrácie je potrebné uviesť pri volaní 5 vstupných parametrov. Prvým argumentom (`values_clean_2019`) je časová tabuľka „čistých“ údajov, druhým argumentom (`30`) je dĺžka časového intervalu, tretím argumentom (`1`) je frekvencia udávaná v jednotkách Hertz (Hz), v ktorej boli údaje namerané. Štvrtým argumentom (`[20, 6, 17, 0]`) je definovaný presný dátum a čas, od ktorého sa spustí vykresľovanie grafov s konštantnou dĺžkou časového intervalu uvedeného v druhom argumente tejto funkcie. Posledným argumentom (`1`) používateľ volí počet grafov, pričom maximálny počet vykreslených grafov je osem.

4.4. Postrehy pri práci s aplikáciami

V tejto kapitole návrhu riešenia, ktorá je zároveň poslednou častou, resp. fázou projektu budú spomedzi všetkých údajov vybraté len tie najzaujímavejšie postrehy, ktoré boli objavené pri prehľadávaní údajov počas analýzy vykonanej prostredníctvom funkcií a aplikácií spomenutých v kapitole č. 4.3.

4.4.1. Meranie efektívnej hodnoty sieťového napätia

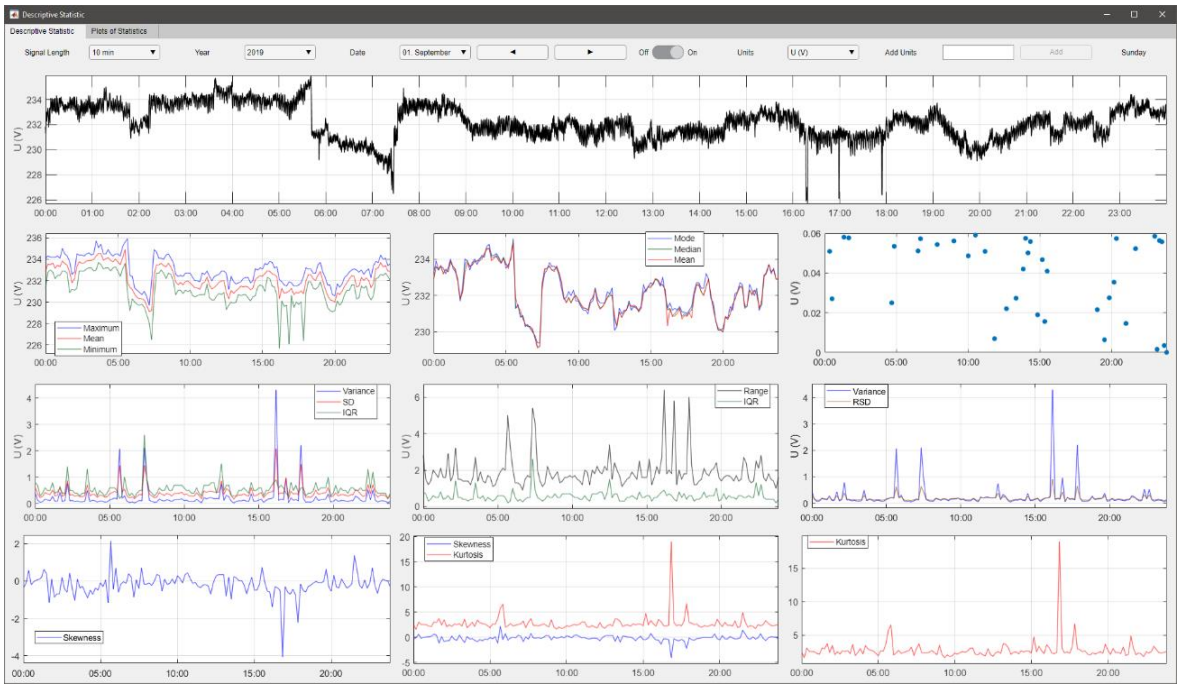
Keďže údajový súbor meraných hodnôt efektívnej hodnoty sieťového napätia s krokom jednej sekundy má začiatok v roku 2012 a siaha až do súčasnosti, nasledujúce postrehy boli skôr objavené náhodným prezeraním údajov autorom práce. Je takmer nemožné realizovať analýzu všetkých vzoriek tohoto rozsiahleho merania. Niektoré z týchto postrehov nie sú až tak úplne jedinečné prípady ako by sa očakávalo a teda je možné odpozorovať podobné prípady viackrát. Je to z toho dôvodu, že nameraných údajov je príliš veľa, čím sa zvyšuje šanca na výskyt neobvyklých situácií. Z tohto dôvodu budú v tejto kapitole odprezentované aj niektoré častejšie sa opakujúce postrehy autora práce. Keďže ide o meranie takej veličiny, kde sa očakáva konštantná hodnota, t.j. 230 V, [27] bude nás okrem vizualizácie všetkých „čistých“ údajov (Obr. 24) v tejto súvislosti zaujímať aj aritmetický priemer všetkých nameraných hodnôt.



Obr. 24 Časový priebeh všetkých „čistých“ údajov sieťového napätia

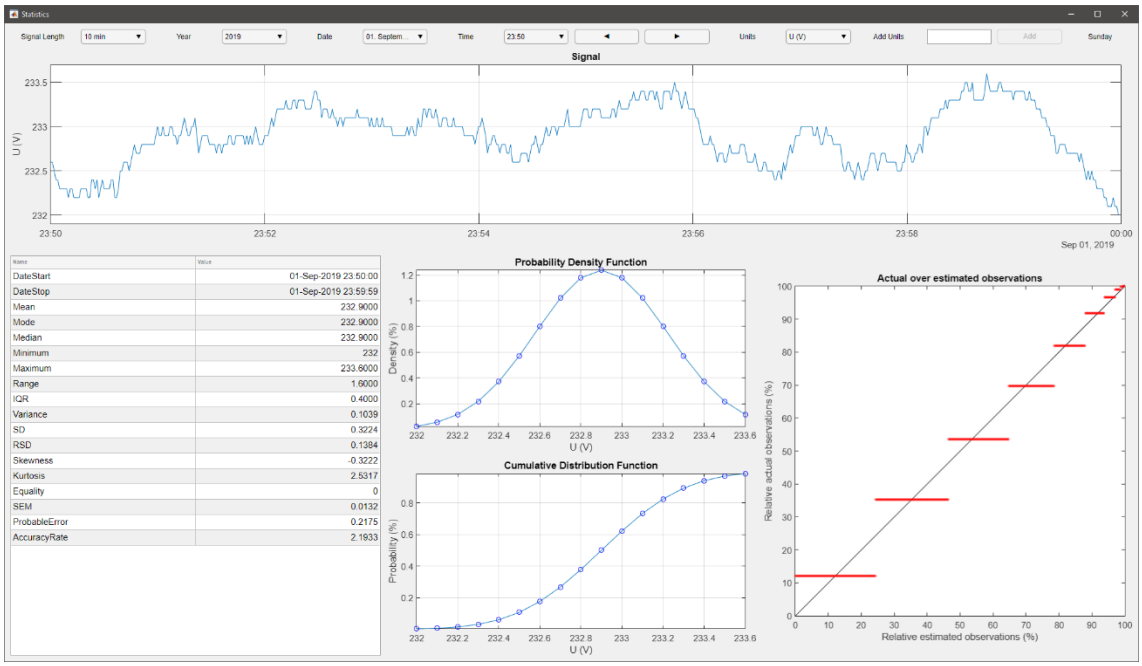
Aritmetický priemer tohto súboru je rovný 232,4 V, čo znamená, že odchýlka od referenčnej hodnoty je rovná 2,4 V. Na Obr. 24 je taktiež vidieť 5 väčších výchyliek a približne 10 stredne veľkých výchyliek, pričom jedná z nich siaha až do záporných hodnôt. Tieto výchylky boli s najvyššou pravdepodobnosťou zapríčinené krátkodobým výpadkom meracieho prístroja. Ide samozrejme o nereálne hodnoty sieťového napätia, ktoré boli zle odčítané meracím prístrojom a je vhodné ich manuálne opraviť.

Aplikácia *Descriptive_Statistics* (Obr. 25) poukazuje na to, že krivky jednotlivých grafov vykreslených v mriežke o veľkosti 3 x 3, ktoré sú vykreslené vo dvojiciach a trojiciach spolu navzájom silno korelujú. Hodnota tejto korelácie nie je nikde vyčíslená, avšak táto skutočnosť je opticky viditeľná. V prvom riadku mriežky je na konci umiestnená zaujímavá ilustrácia podobností troch koeficientov z ilustrácie umiestnenej uprostred riadku. Tieto body vznikli vypočítaním smerodajnej odchýlky kriviek aritmetického priemeru, modusu a mediánu, čím sme získali prehľad o tom, ako veľmi podobné sú si navzájom tieto hodnoty. Graf ilustruje skutočnosť, že z 10-minútových vzoriek dňa 1. 9. 2019 je vzorka, resp. rozdelenie začínajúce v čase od 23:50 do 00:00 ideálnym symetrickým rozdelením [28] údajov, pretože je na bodovom grafe najnižšie položeným bodom na vertikálnej osi grafu a jeho hodnota je rovná nule.



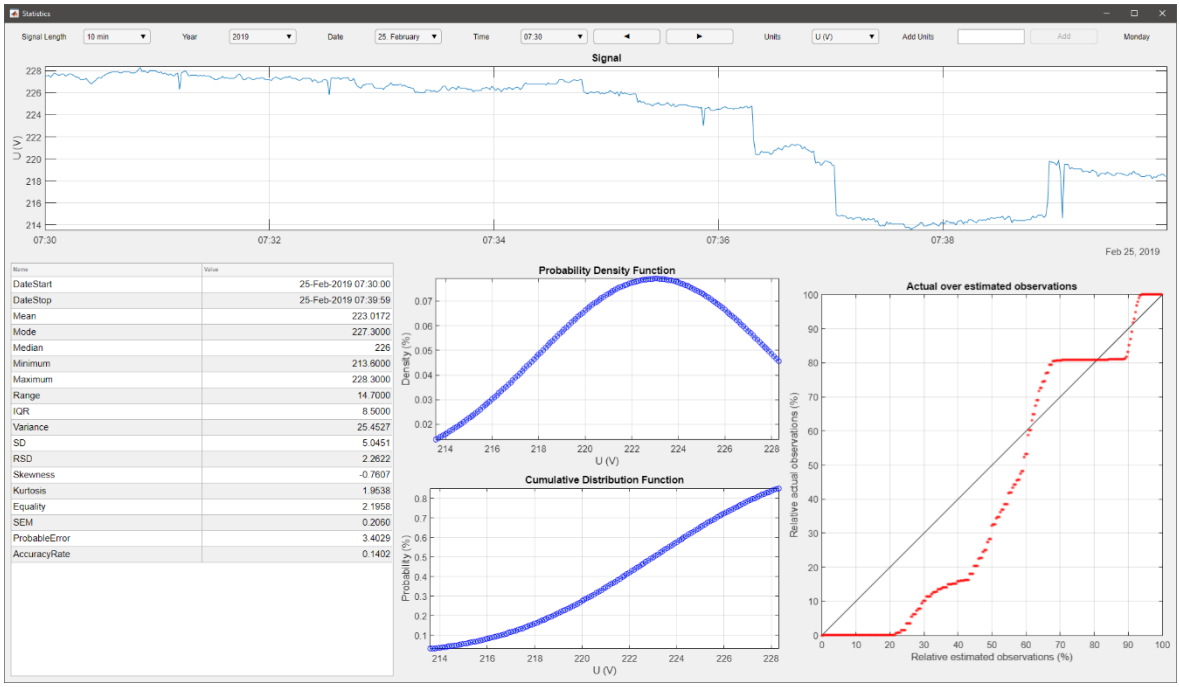
Obr. 25 Analýza 10-minútových vzoriek dňa 1. 9. 2019

Táto skutočnosť je priamo dokázaná pomocou hĺbkovej analýzy tohto 10-minútového intervalu merania v aplikácii *Statistics_Analysis*, kde je vidieť na ilustrácii funkcie hustoty pravdepodobnosti rozdelenia hodnôt (ilustrácia s názvom *Probability Density Function*), že rozdelenie je skutočne symetrické. Nemusí to byť pravidlom, ale jedným z postrehov autora je fakt, že rozdelenia s najvyššou symetriou boli práve tie, ktoré mali zároveň nízky rozsah meraných hodnôt. Meraná vzorka na Obr. 26 má v tabuľke uvedený rozsah hodnôt rovný 1,6 V. Pretože merací prístroj, ktorým boli tieto údaje merané má rozlíšenie 0,1 V, resp. bol nastavený v takom režime, znamená to, že v období tohto merania sa efektívna hodnota sieťového napätia obmieňala len medzi šestnástimi hodnotami. V ďalšej časti tejto podkapitoly bude ukázané, aký vplyv má veľkosť rozsahu hodnôt na výsledky štatistickej analýzy.



Obr. 26 Štatistická analýza vzorky dňa 13.11.2019 v čase od 19:10 do 19:20

Z dostupných meraných údajov v roku 2019 bola vybratá 10-minútová vzorka merania s najvyšším rozsahom hodnôt, ktorý je rovný 14,7 V (v tabuľke na Obr. 27). Stojí za povšimnutie skutočnosť, že hustota bodov, ktoré tvorili tmavo modré krivky ale aj červený bodový graf sa niekoľkonásobne zvýšila v porovnaní s 10-minútovou vzorkou analyzovanou na Obr. 26.

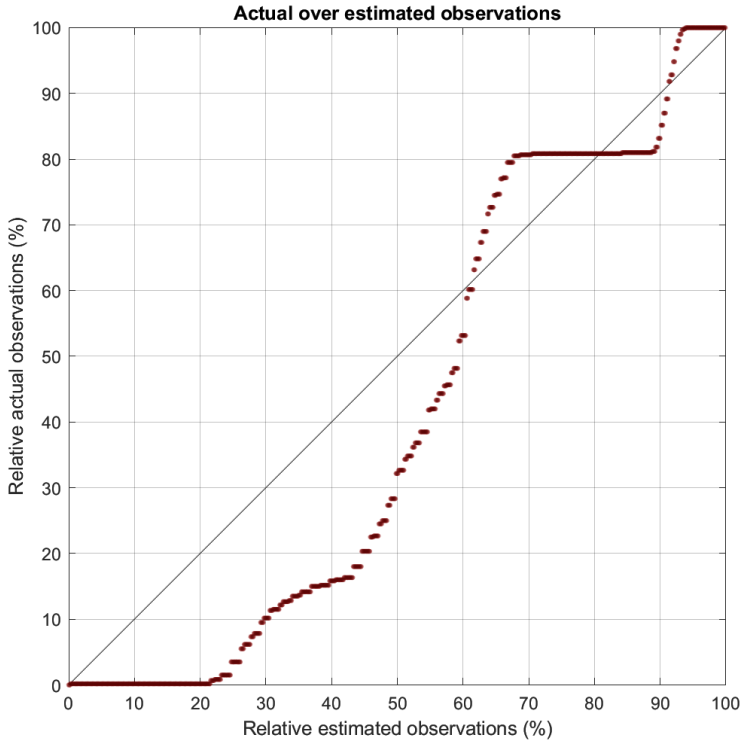


Obr. 27 Štatistická analýza vzorky dňa 25.2.2019 v čase od 7:30 do 7:40

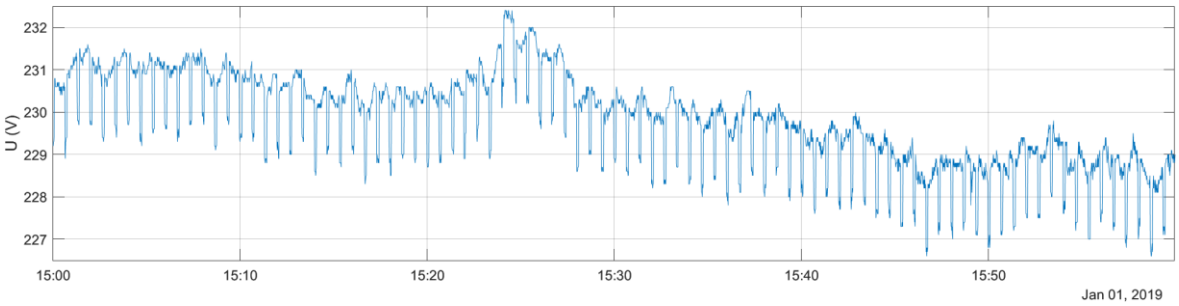
Graf nespojitých percentuálnych hodnôt odhadu intervalu spoľahlivosti (Obr. 28) je vďaka väčšiemu rozsahu hustejší ako na predchádzajúcej vzorke a je zároveň zložený z viacerých úrovni. Zatiaľ, čo

prvá vzorka dosahuje približne 9 úrovní, táto vzorka dosahuje takmer pri každom percentuálnom odhade intervalu spoľahlivosti iné úrovne. Z ilustrácie tejto vzorky sa dá skonštatovať, že odhad intervalu spoľahlivosti je skutočne spoľahlivý len od približne 60 do 80 percent a ďalej od 90 do 100 percent (body umiestnené nad referenčnou teoretickou krivkou). V ostatných percentuálnych odhadoch sa nevieme spoľahnúť na vypočítaný interval spoľahlivosti, čím tento graf poukazuje na rozdiely medzi teóriou a praxou aj keď na druhej strane je vhodné to prijať s rezervou, keďže ide o jedinečnú situáciu rozdelenia meraných hodnôt. V ďalšej časti tejto podkapitoly budú popísané niektoré postrehy autora z aplikácie *Signals*.

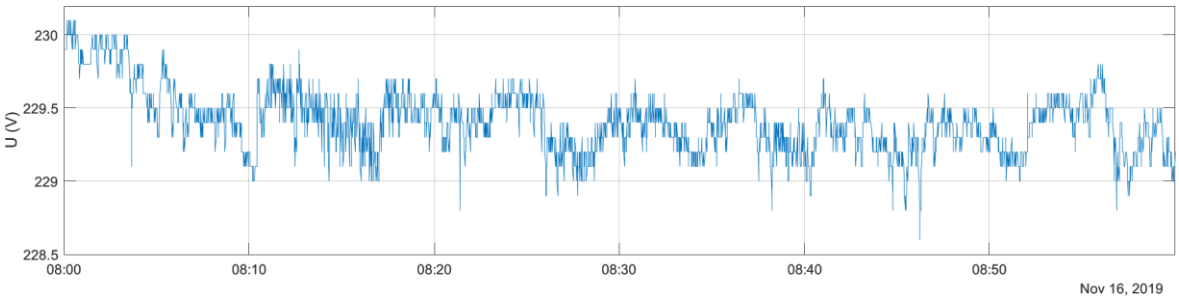
Aplikácia *Signals*, ktorá slúži na ilustráciu časových priebehov, resp. signálov meraných údajov je vhodná na analýzu tvaru signálu. Pri náhodnom prezeraní 60-minútových vzoriek údajov sa najčastejšie opakovali štyri typy tvarov signálov, ktoré sú ilustrované na Obr. 29, Obr. 30, Obr. 31 a Obr. 32. V prvom prípade sa konštantne ocitol približne každých 80 sekúnd výrazný pokles napätia, ktorý trval tiež konštantne okolo 10 sekúnd. V druhom prípade má signál relatívne hladký priebeh s miernymi výkyvmi. Tretí prípad je podobný prvému prípadu, ktorý pripomína skôr obdĺžnikový priebeh signálu a posledný typ tvaru signálu je kombináciou druhého a tretieho typu. Podobné typy sa opakovali aj v prípade kratšie zvolených vzoriek údajov.



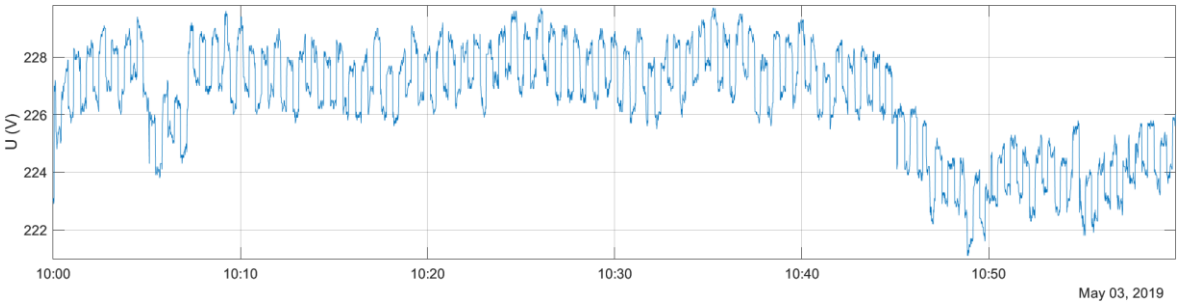
Obr. 28 Graf porovnania skutočnej (body grafu) a teoretickej (referenčná diagonála) percentuálnej zložky odhadovaných intervalov spoľahlivosti



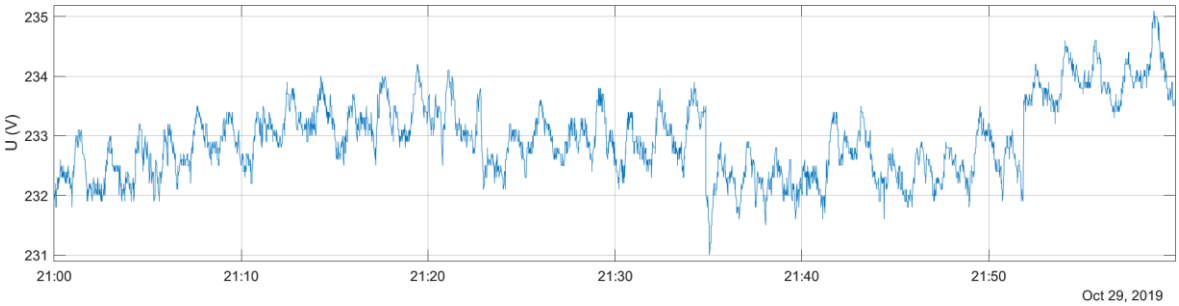
Obr. 29 Časový priebeh signálu sieťového napätia dňa 1.1.2019 v čase od 15:00 do 16:00



Obr. 30 Časový priebeh signálu sieťového napätia dňa 16.11.2019 v čase od 8:00 do 9:00

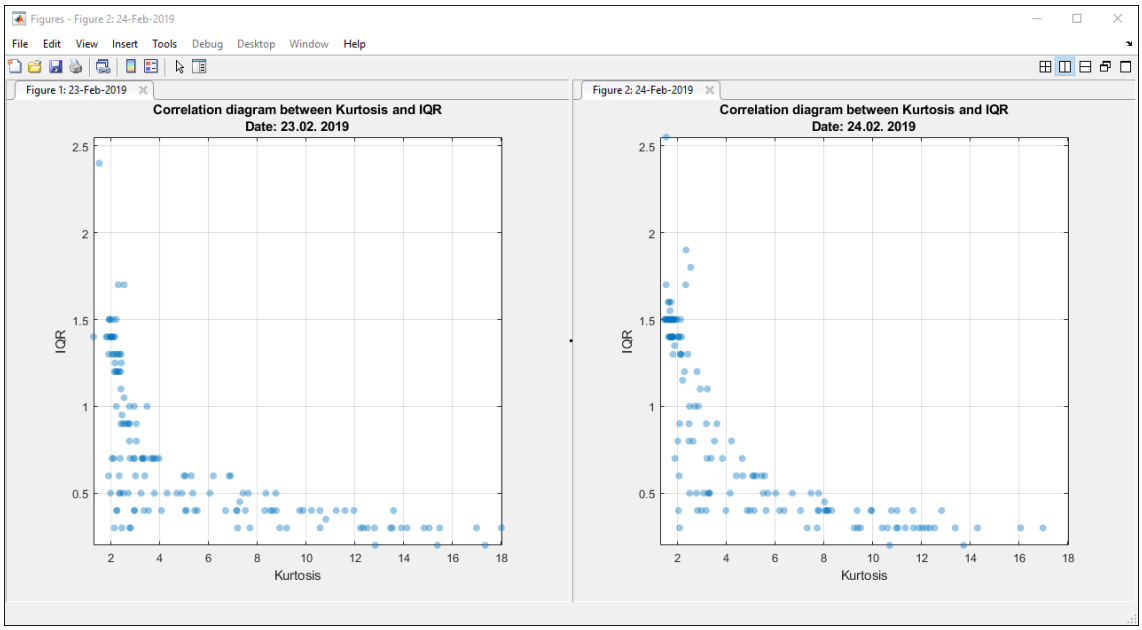


Obr. 31 Časový priebeh signálu sieťového napätia dňa 3.5.2019 v čase od 12:00 do 13:00



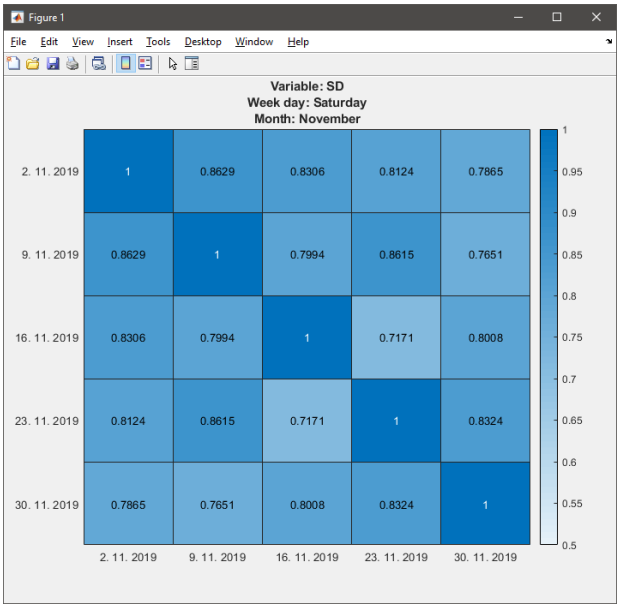
Obr. 32 Časový priebeh signálu sieťového napätia dňa 29.10.2019 v čase od 21:00 do 22:00

Zaujímavým postrehom pri náhodnom prehliadaní korelačných diagramov bolo porovnanie koeficientu špicatosti a medzikvartilového rozptylu 10-minútových vzoriek dňa 23.2.2019 a 24.2.2019, kedy sa body na oboch grafoch vyskytovali pod pomyselnou krivkou exponenciálneho rozkladu.[29] Podobne to vyzerá aj v prípade iných dní, avšak zrejme najlepšie to je vidieť na vyššie umiestnených ilustráciách na Obr. 33.



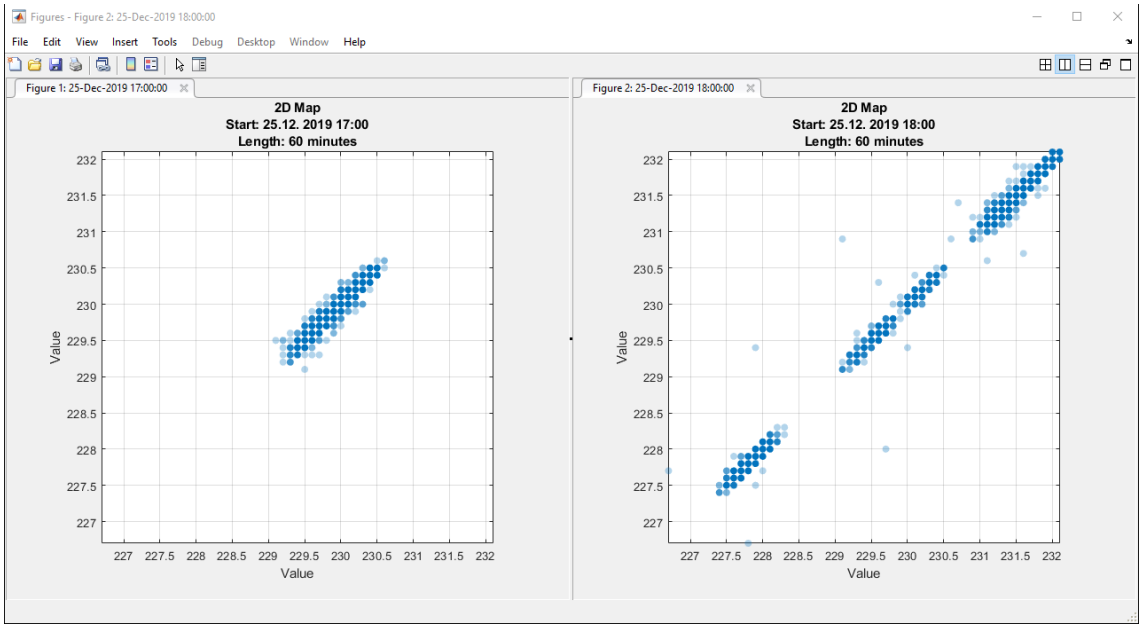
Obr. 33 Korelačný diagram koeficientu špicatosti a medzikvartilového rozptylu hodnôt 10-minútových vzoriek dňa 23.2.2019 a 24.2.2019

Pri náhodnom prehľadávaní korelačných matíc sa podarilo nájsť maticu s pomerne vysokými korelačnými hodnotami, ktoré boli vypočítané porovnávaním signálov smerodajnej odchýlky v mesiaci november v roku 2019 (Obr. 34). Tieto signály vznikli zlúčením vypočítaných smerodajných odchýlok zo 60-minútových vzoriek príslušných dní, ktoré sú ilustrované aj v aplikácii *Descriptive_Statistics*. Vysokú podobnosť týchto signálov možno odôvodniť tým, že meranie prebiehalo na katedre KTPE, ktorá rovnako ako ostatné školské katedry zvykne byť v sobotu zatvorená.



Obr. 34 Korelačná matica signálov smerodajných odchýlok 60-minútových vzoriek všetkých sobôt v mesiaci november v roku 2019

Ilustrácia výskytu hodnôt v prípade merania sieťového napätia poukazuje na veľkú stabilitu, resp. výraznú zmenu meraného signálu. Náhodným prehľadávaním údajov z roku 2019 sa podarilo nájsť dve po sebe idúce vzorky ilustrované na Obr. 35, kde bola vďaka ilustrácii objavená vysoká stabilita signálu (vľavo) a vysoká premenlivosť hodnôt signálu (vpravo). Body týchto máp sa prevažne zoskupujú v diagonálnej rovine mapy. Vyššia koncentrácia odľahlých bodov zároveň signalizuje nižšiu stabilitu meraného signálu.



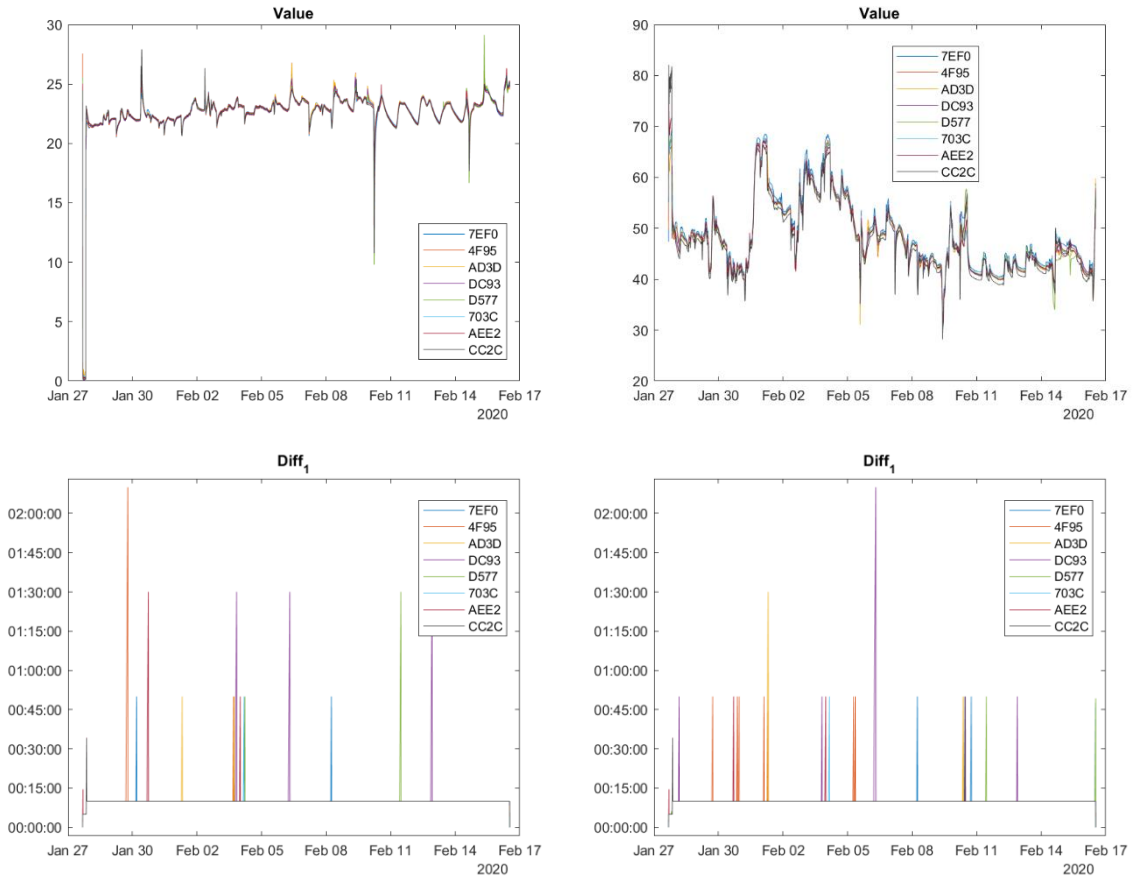
Obr. 35 Dvojrozmerné mapy výskytu hodnôt 60-minútových vzoriek dňa 25.12.2019

Ako už bolo spomenuté v kap. 4.3.7, aplikovanie funkcie rýchlej Fourierovej transformácie na údaje merania efektívnej hodnoty sieťového napätia nemá veľký význam. Ilustrácie sa pri rôznych vstupných parametroch funkcie *plot_signal_FFT* líšia od seba len minimálne, čo znamená, že pravdepodobne žiadna časová vzorka nemá periodický charakter. Keďže funkcia rovnako ako ostatné komponenty projektu sú plne kompatibilné s akýmkoľvek typom údajov s časovou značkou, funkcia má využitie pri analýze iných typov digitálnych signálov, ktoré nie sú v tejto práci k dispozícii.

4.4.2. Meranie teploty a relatívnej vlhkosti

Údaje tohto merania boli ukladané s rôzne nastavenou šírkou krokovania, čo značne komplikuje aplikovať tieto údaje do analýzy vyššie pripravených aplikácii a funkcií projektu. Podobným postupom ako je opísaný v kap. 4.2 prebiehal proces štrukturalizácie získaných údajov z teplomera. Poskytnuté údaje síce dostali štruktúru, avšak nie dostatočne vhodnú na analýzu. Spravidla neexistuje žiaden univerzálny ETL proces, ktorý je schopný spracovať akýkoľvek typ a štruktúru údajov. Preto bolo potrebné navrhnúť nový, automatizovaný ETL proces prispôsobený pre tento konkrétny formát údajov získaných z teplomerov. Funkcia, ktorou boli údaje spracované

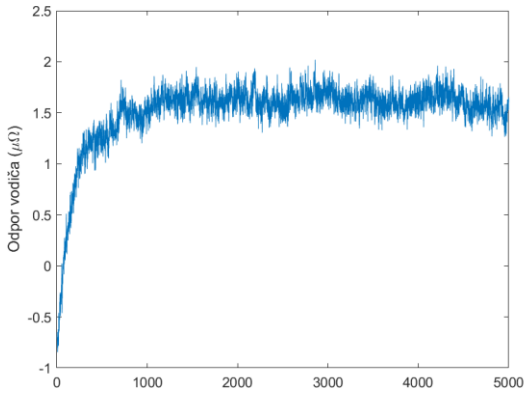
je uložená v zložke *ETL proces*, ktorá je umiestnená v zložke pripravených údajov a nesie názov *folder_processing*. Kód programu je zložený z jednej hlavnej funkcie a troch pomocných funkcií. Funkcia má jeden vstupný argument, ktorým je cesta k priečinku, v ktorom by sa mal vyskytovať aspoň jeden podpriečinok s textovými súbormi s príponou **.edf*. Správnym zavolaním tejto funkcie sa spracujú všetky údaje v každom podpriečinku zvolenej zložky. Najprv program MATLAB načíta zozbierané údaje do svojho pracovného prostredia a následne spracuje, vygeneruje a uloží tieto štruktúrované údaje vo formáte **.csv*. Následne vygeneruje grafy časových priebehov a prvej derivácie vektora časových značiek z údajov, ktoré boli zozbierané teplomermi, t.j. teplota a relatívna vlhkosť vzduchu (Obr. 36). Tieto obrázky sú uložené v podpriečinkoch vo formáte **.png* a rozlišujú sa podľa svojich pomenovaní. Časové priebehy všetkých teplomerov sú uložené pod názvom *Value_T* a pre hodnoty relatívnej vlhkosti vzduchu pod názvom *Value_RH*. Podobným spôsobom sú odlíšené prvé derivácie časových vektorov, t.j. *Diff_T* a *Diff_RH*. V každej ilustrácii je znázornená legenda, v ktorej sú uvedené posledné 4 znaky MAC adresy teplomera. V každom podpriečinku, na každej ilustrácii sú farby týchto kriviek prislúchajúcich ku konkrétnemu teplomeru vždy rovnaké.



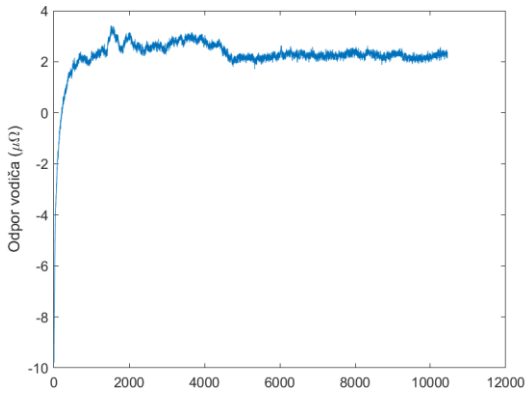
Obr. 36 Vzor časových priebehov (hore) a prvej derivácie časových vektorov (dole) teploty (vľavo) a relatívnej vlhkosti vzduchu (vpravo)

4.4.3. Meranie odporu meracieho kábla a napätia

Poskytnuté údaje k tomuto meraniu nebolo možné hlbšie analyticky spracovať vyššie spomenutými metódami, pretože meraným údajom neprislúchajú žiadne časové značky. Údajom aj napriek tomu bola dodaná aspoň provizórna štruktúra, keďže pôvodné údaje boli rozmiestnené do viacerých údajových súborov bez referenčného stĺpca, ktorý je v tomto prípade stĺpec s časovými značkami. Na účely implementácie tejto štruktúry bol vyhotovený skript s názvom *create_voltage_data*. Tento skript zlúči všetky súvisiace údajové súbory do jednej tabuľky, uloží ich do súboru *voltage.mat* do priečinka pripravených údajov. Údaje spolu navzájom súvisia vtedy, ak sú pomenované rovnakými písmenami (napr. *mmm0.csv*, *mmm1.csv* a *mmm2.csv*). Do tohto priečinka sa uložia aj časové priebehy týchto údajov (Obr. 37 a Obr. 38).



Obr. 37 Ilustrácia údajov tabuľky s názvom *e*



Obr. 38 Ilustrácia údajov tabuľky s názvom *s*

Keďže nie je známy čas, v ktorom tieto údaje boli merané, čas je nahradený vektorom s poradovým číslom. Z ilustrácií vieme posúdiť, že odpor vodiča na začiatku viacnásobného merania sa postupne zvýšil (jav zapríčinený zahrievaním multimetra) do oblasti hodnôt približne medzi 1 až 3 μΩ. Tieto hodnoty by mohli zodpovedať skutočným hodnotám odporu bežného vodiča.

Záver

Docieliť postrehy pri práci s údajmi vie byť niekedy tzv. behom na dlhú trať. Obzvlášť pokiaľ ide o návrh automatizovaného ETL procesu prispôsobeného na akýkoľvek typ údajov s časovou základňou. Správne navrhnúť a optimalizovať takýto ETL proces si vyžaduje niekoľko týždňov až mesiacov práce a tvorí odhadom až 70 % celkového času návrhu procesov údajovej vedy. ETL proces dáva štruktúru zozbieraným údajom, čím umožňuje ďalším procesom údajovej vedy efektívne a najmä bezproblémovo manipulovať s údajmi. Je vhodné poznamenať, že počet riadkov zozbieraných údajov pre meranie sieťového napätia je rovný 150557655. Po zbehnutí ETL procesu pre toto meranie sa zachovalo až 149122800 riadkov „čistých“ údajov, čo predstavuje len 0,95 % stratu poškodených alebo inak nevhodných údajov pre analýzu z celkového počtu údajov. Po návrhu tohto procesu nasledoval návrh softvéru na analýzu a vizualizáciu štruktúrovaných údajov. Navrhnuté aplikácie a funkcie pomáhajú používateľovi napr. identifikovať kvalitu stability sieťového napätia, skúmať príčiny prudkých zmien, analyzovať výsledky štatistických výpočtov pre jednotlivé meracie obdobia a pod. Návrh a realizácia tohto softvéru si taktiež vyžiadala niekoľko týždňov času a námahy. Ako bolo v teoretickej časti údajovej vedy spomínané, na podobných projektoch pracuje obvykle viacero ľudí v tíme. V tejto práci boli vypracované až 3 samostatné projekty údajovej vedy, pričom najviac pozornosti bolo venovanej efektívnej hodnote sieťového napätia. Pre údaje získané z teplomerov a údaje merania odporu vodiča boli vyhotovené samostatné funkcie na transformáciu údajov, čím sa výrazne zjednoduší používateľovi práca s údajmi. Navrhnutý projekt má ale aj svoje nedostatky. Keďže projekt slúži na spracovanie importovaných údajových súborov, znamená to, že nie je schopný spracovať údaje merané v reálnom čase. Taktiež plynulosť vypracovaných aplikácií by pravdepodobne mohla byť lepšia po optimalizácii niektorých častí kódov.

Zoznam použitej literatúry

[1]. MATLAB Documentation. [Online]. 1994-2020. [cit. 2020-03-19]. Dostupné na internete: <https://www.mathworks.com/help/matlab>.

[2]. Data Types - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-20]. Dostupné na internete: <https://www.mathworks.com/help/matlab/data-types.html>.

[3]. Numeric Types - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-20]. Dostupné na internete: <https://www.mathworks.com/help/matlab/numeric-types.html>.

[4]. Characters and Strings - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-20]. Dostupné na internete: <https://www.mathworks.com/help/matlab/characters-and-strings.html>.

[5]. Dates and Time - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-20]. Dostupné na internete: <https://www.mathworks.com/help/matlab/date-and-time-operations.html>.

[6]. Tables - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-20]. Dostupné na internete: <https://www.mathworks.com/help/matlab/tables.html>.

[7]. Timetables - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-20]. Dostupné na internete: <https://www.mathworks.com/help/matlab/timetables.html>.

[8]. Structures - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-20]. Dostupné na internete: <https://www.mathworks.com/help/matlab/structures.html>.

[9]. MATLAB App Designer - MATLAB. [Online]. 1994-2020. [cit. 2020-03-25]. Dostupné na internete: <https://www.mathworks.com/products/matlab/app-designer.html>.

[10]. Write Callbacks in App Designer - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-25]. Dostupné na internete: <https://www.mathworks.com/help/matlab/creating_guis/write-callbacks-for-gui-in-app-designer.html>.

[11]. Reuse Code Using Helper Functions - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-25]. Dostupné na internete: <https://www.mathworks.com/help/matlab/creating_guis/code-and-call-app-functions-in-app-designer.html>.

[12]. Share Data Within App Designer Apps - MATLAB & Simulink. [Online]. 1994-2020. [cit. 2020-03-25]. Dostupné na internete: <https://www.mathworks.com/help/matlab/creating_guis/share-data-across-callbacks-in-app-designer.html>.

[13]. GRUS, Joel. 2015. *Data Science from Scratch*, California : O'Reilly Media, 2015. 330 s. 978-1-491-90142-7.

[14]. Data Science Definition. [Online]. 2020. [cit. 2020-03-26]. Dostupné na internete: <https://techterms.com/definition/data_science>.

[15]. What is Data Acquisition? – National Instruments. [Online]. 2020. [cit. 2020-03-26]. Dostupné na internete: <http://www.ni.com/data-acquisition/what-is>.

[16]. What is ETL (Extract, Transform, Load)? ETL Explained – BMC Blogs. [Online]. 2017. [cit. 2020-03-26]. Dostupné na internete: <https://www.bmc.com/blogs/what-is-etl-extract-transform-load-etl-explained>.

[17]. Data Analytics Definition. [Online]. 2019. [cit. 2020-03-26]. Dostupné na internete: <https://www.investopedia.com/terms/d/data-analytics.asp>.

[18]. What is data visualization? A definition, examples, and resources. [Online]. 2003-2020. [cit. 2020-03-28]. Dostupné na internete: <https://www.tableau.com/learn/articles/data-visualization>.

[19]. LENOVO IdeaPad 510 15 (80SR00AHCK) | LENOVO-SHOP.SK. [Online]. 2013-2020. [cit. 2020-03-30]. Dostupné na internete: <https://www.lenovo-shop.sk/lenovo-ideapad-510-15-80sr00ahck#Technical_specification>.

[20]. 860 EVO SATA III 2.5inch SSD | MZ-76E500B/EU | Špecifikácia a popis | Samsung Slovenská republika. [Online]. 2020. [cit. 2020-03-30]. Dostupné na internete: <https://www.samsung.com/sk/memory-storage/860-evo-sata-3-2-5-ssd/MZ-76E500BEU>.

[21]. FEHÉR, Adam. 2018. *Štatistika v Exceli*, Košice : Technická Univerzita, 2018. 66 s.

[22]. How many sheets, rows, and columns can a spreadsheet have?. [Online]. 2020. [cit. 2020-03-28]. Dostupné na internete: <https://www.computerhope.com/issues/ch000357.htm>.

[23]. Moving Average: What it is and How to Calculate it - Statistics How To. [Online]. 2013. [cit. 2020-04-14]. Dostupné na internete: <https://www.statisticshowto.com/moving-average>.

[24]. MILLER, James. 2017. *Statistics for Data Science*, Birmingham: Packt Publishing, 2017. 286 s. 978-1-78829-067-8.

[25]. HOLCOMB, Zealure. 2017. *Fundamentals of Descriptive Statistics*, Abingdon : Routledge, 2017. 98 s. 978-1-884-58505-0.

[26]. SCHILLING, Robert. 2016. *Digital Signal Processing using MATLAB*, Boston : Cengage Learning, 2016. 800 s. 978-1-305-63519-7.

[27]. IEC - World Plugs: List view by potential. [Online]. 2020. [2020-04-26]. Dostupné na internete: <https://www.iec.ch/worldplugs/list_byelectricpotential.htm>

[28]. Relation Between Mean Median and Mode With Solved Example Questions. [Online]. 2020. [2020-04-26]. Dostupné na internete: <https://byjus.com/maths/relation-between-mean-median-and-mode>

[29]. Leike, A.: *Demonstration of the exponential decay law using beer froth*. V: European Journal of Physics roč. 23 / 2001, číslo 1, s. 21–26. doi: http://dx.doi.org/10.1088/0143-0807/23/1/304

Prílohy

Príloha A: CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe, spustiteľný projekt programu MATLAB, dodané namerané hodnoty potrebné pre štatistické vyhodnocovanie, výsledky štatistickej analýzy. (Z kapacitných dôvodom prenosu údajov do školskej knižnice boli odstránené pripravené údaje merania sieťového napätia po rok 2017 vrátane, tabuľka chýbajúcich hodnôt a všetky pôvodné údaje. Ďalej boli odstránené všetky pôvodné údaje merania teploty a relatívnej vlhkosti vzduchu a väčšia časť pripravených údajov)

Príloha B: Kódy projektu