ROBD dokumetace

Verze: 1.0

ROBD je základní balíček pro on-board diagnostiku kolejových vozidel. Obsahuje funkce pro diagnostiku vady kol (ploché kolo), stability podvozku a jízdního komfortu vozidel.

Obsah

1 Identifikace výsledku 5
2 Verze dokumentu 6
3 Obecný popis SW knihovny ROBD 7
3.1 Definice vstupních dat 7
3.2 Nahrání balíčku 8
3.3 Externí knihovny 9
3.4 Nutný externí SW potřebný pro funkci 10
4 Práce s client knihovnou (**ROBD.client_io**) 11
4.1 Třída pro načtení csv (**ROBD.client_io.CSV**) 11
4.1.1 Metoda **load** 11
4.2 Třída pro přípravu dat (**ROBD.client_io.PRE**) 13
4.2.1 Metoda **cut_one_sensor** 14
4.2.2 Metoda **cut_few_sensors** 14
4.3 Třída pro vykreslení grafů (**ROBD.client_io.PLOT**) 15
4.3.1 Metoda **stab** 15
4.3.2 Metoda **comf** 16
4.4 Třída pro uložený vyhodnocených dat (**ROBD.client_io.SAVE**) 17
4.4.1 Metoda **json** 17
5 Vady kol (**ROBD.flat**) 18
5.1 Metoda CALC_FLAT_AKF a CALC_FLAT_AKF_PLOT 18
5.2 Balík FLAT_FTR11 20
5.2.1 Tréning 21
5.2.2 Detekce z DB 21
5.2.3 Detekce ze signálu 21
5.2.4 Formát výstupu detekce 22
5.3 Balík FLAT_FTR38 22
6 Stabilita (**ROBD.stab**) 23
6.1 Konfigurace pro vyhodnocení stability (**stab_config.ini**) 23
6.2 Nahrání dílčího balíčku (**import**) 23
6.3 Trendová diagnostika – statistika – (**ROBD.stab.Trend**) 23
6.3.1 Metoda **CALC_few_sensors** 24

6.3.2 Metoda **CALC_one_sensor** ... 24
6.3.3 Metoda **CALC_one_block** ... 25
6.3.4 Omezení vyhodnocení polohou a rychlostí jízdy ... 26
6.4 Varovná diagnostika – upozornění – (**ROBD.stab.Warn**) ... 26
6.4.1 Metoda **CALC_few_sensors** ... 27
6.4.2 Metoda **CALC_one_sensor** ... 27
6.4.3 Metoda **CALC_one_block** ... 28
6.4.4 Příklad použíti třídy **ROBD.stab** ... 28
7 Komfort (**ROBD.comf**) ... 30
7.1 Konfigurace pro vyhodnocení stability (**comf_config.ini**) ... 30
7.2 Nahrání dílčího balíčku ... 30
7.3 Vyhodnocení komfortu – (**ROBD.comf.Trend**) ... 31
7.3.1 Metoda **CALC_one_sensor** ... 31
7.3.2 Metoda CALC_one_block ... 32
7.3.3 Příklad použíti třídy **ROBD.comf.Trend** ... 33
8 ROBD Desktop Client ... 34
8.1 Instalace ... 34
8.2 Konfigurace ... 34
8.3 Ovládací prvky ... 35
8.4 Příklad exportů ... 38
1 Měřící řetězec ... 2
2 Umístění sensorů na vozidle ... 2
3 Vzorkovací frekvence a parametry digitalizace dat ... 4
4 Sběr dat ... 5
4.1 Aktivace měření ... 5
4.2 Přenos dat ... 5
4.3 Archivace dat ... 6
1 Technické parametry databáze ... 2
1.1 Verze databáze ... 2
1.2 Nastavení přístupů ... 2
2 Struktura databází a tabulek ... 2
2.1 Tabulka PREFIX_DATA ... 3

2.2 Tabulka PREFIX_META 3
2.3 Tabulka PREFIX_NODE 3
2.4 Tabulka PREFIX_STAT 4
2.5 Tabulka PREFIX_CPB 4
2.6 Tabulka PREFIX_FLAT_POSTFIX 5
2.7 Tabulka PREFIX_STAB 5
2.8 Tabulka PREFIX_COMF 6
2.9 Tabulka PREFIX_MARK_XXXX 6

1 Identifikace výsledku

Řešeno v projektu	TAČR CL 01000243
Název projektu	On board diagnostika pojezdu kolejových vozidel
Název výsledku	SW nástroj pro diagnostiku pojezdu kolejových vozidel
Období řešení	01. 01. 2023 – 30. 6. 2026
Hlavní příjemce	Univerzita Pardubice
Další příjemce	ŠKODA TRANSPORTATION a.s.
Číslo výsledku	CL01000243-V2
Typ výsledku	R - software
Datum dosažení výsledku	03/2026
Původci výsledku	50 % - Univerzita Pardubice 50 % - ŠKODA TRANSPORTATION a.s.
Možnosti využití	K využití je potřeba vždy získat licenci, kontaktní osoba: Ing. Jakub Vágner, Ph.D., Univerzita Pardubice, Studentská 95, 532 10 Pardubice jakub.vagner@upce.cz

2 Verze dokumentu

Tento dokument je platný pro následující verze knihoven a funkcí:

Datum poslední úpravy dokumentu	27. 07. 2026
Verze dokumentu = verze SW ROBD	1.0
Verze FLAT_AKF	1.08
Verze FLAT_FTR11	1.03
Verze FLAT_FTR38	1.03
Verze stabilita	1.02
Verze komfort	1.03
Verze ROBD Desktop Client	1.7.7

3 Obecný popis SW knihovny ROBD

Tato dokumentace popisuje jednotlivé funkce SW balíku ROBD. Jedná se o algoritmy pro detekci vad na jízdní ploše kola, hodnocení stability vozidla a jízdního komfortu. Vyhodnocení se provádí na signálech naměřených onboard diagnostikou dle popsané metodiky. U algoritmů se předpokládá trojí použití:

- **Client:** knihovna se využije k vyhodnocení dříve naměřených dat v desktopové aplikaci na stolním počítači, obvykle za účelem analýzy dat za účelem výzkumu.
- **Server:** knihovna se použije pro automatické vyhodnocení dat na serveru, výsledky tvoří časovou řadu pro trendovou diagnostiku.
- **On-board:** knihovnu lze aplikovat při online vyhodnocení přímo na vozidle, příchozí data jsou zpracovávána ihned a lze tak realizovat výstražnou diagnostiku.

Celá knihovna je realizována v jazyce Python, což definuje způsob její implementace. Knihovnu však lze spouštět z externích programů, což je využíváno zejména u přístupů on-desktop a on-server. Uživatelsky spustitelné části jsou realizovány v jazyce FPC ve vývojovém prostředí Lazarus IDE.

Pro varianty client a server je nutné použít vždy databázi. V případě client je databáze provozována na lokálním počítači, v případě server je provozována na serveru. Struktura obou databáze je v obou případech stejná, tedy pro verzi client lze přistupovat jak k vlastní (lokální), tak vzdálené (serverové) databázi.

3.1 Definice vstupních dat

Vstupní data lze do databáze vkládat v různých formátech, pro každý formát je nutné použít příslušný skript pro import nebo si vytvořit vlastní skript. Data ale musí I tak splňovat následující požadavky:

- Umístění snímačů je definováno v **příloze 1** tohoto dokumentu;
- musí být splněny požadavky na kvalitu sbíraných dat pro jednotlivé kanály popsané v **příloze 1**;
- pro import do databáze jsou připraveny funkce pro načtení dat ve formátu CSV (oddělený středníkem, desetinný oddělovač tečka);
- pro import jiných formátů dat musí být data načítána a ukládána do databáze tak, aby byly splněny podmínky dle popisu struktury databáze (**příloha 2**);
- při onboard použití se předpokládá integrace knihovny přímo do výpočetního SW, tedy data jsou předána jako array (popis u příslušných funkcí).

3.2 Automatizace výpočtu

Pro serverovou verzi se předpokládá automatizace celého procesu. Níže je uvedeno schéma, které popisuje tok dat a nutnost zásahu uživatelů. Hlavním tokem je online vstup do vstupní tabulky -> automatizovaný solver -> výstup do výstupní tabulky -> vizualizace.

Samotná vizualizace není předmětem popisovaného SW, ale lze s výhodou použít již zavedený systém nebo univerzální nástroj GRAFANA. Výstupní tabulka může sloužit jako vstup do nadřazeného IT systému.

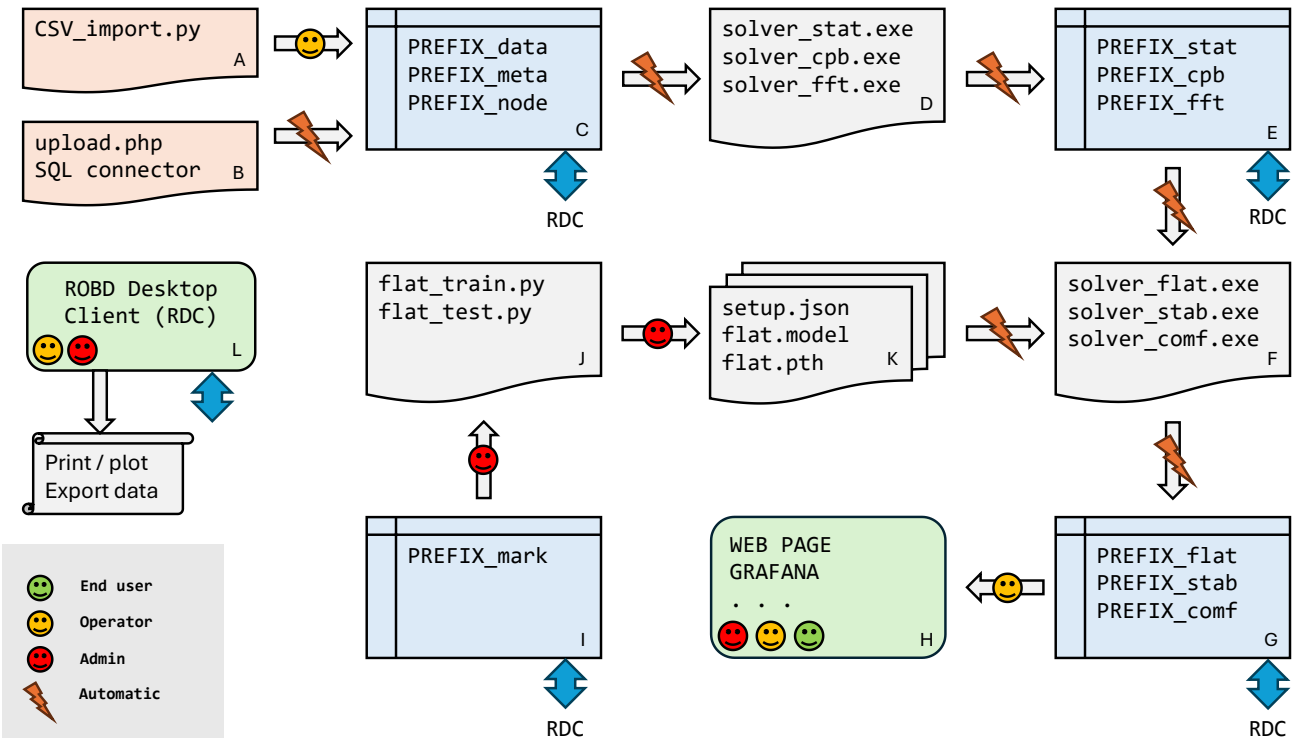
„End user“ – zaměstnanec údržby, koncový uživatel výsledků.

„Operátor“ – uživatel, který dohlíží na chod systému, importuje data, exportuje data, může provést kategorizace dat pro nové učení

„admin“ – uživatel, který spravuje tabulky, udržuje systém a má právo provést nové učení. Tento uživatel musí mít znalost na úrovní nejen procesu vyhodnocení dat, ale i úpravy kódu skriptů a konfiguračních souborů.

Popsaný systém lze s výhodou provozovat v cloudovém prostředí, případně ve virtuálním kontejneru.

Definice databáze je v příloze 1.



3.4 Nahrání balíčku

```
import ROBD
```

Nahrání dílčího balíčků:

```
from ROBD import jmeno_dílčího_balíčku
```

Jméno dílčího balíčku	*Popis obsahu a funkce*
client_io	Slouží k importu, rozdělení csv dat a uložení výstupudo json.
flat_AKF	Obsahuje funkce pro jednoduchou detekci vad na jízdní ploše kola
flat_FTR11	Obsahuje funkce pro trénování, analýzu a detekci vad na jízdní ploše kola pomocí strojového učení z vektoru statistických hodnot.
flat_FTR38	Obsahuje funkce pro trénování, analýzu a detekci vad na jízdní ploše kola pomocí strojového učení z vektoru statistických hodnot a také frekvenční oblasti.
stab	Obsahuje funkce pro vyhodnocení trendových parametrů pro sledování stavu stability vozidla (kmitání rámu podvozku). Funkce pro vyhodnocení varovné diagnostiky pro hlášení aktuálního stavu stability vozidla.
comf	Funkce pro vyhodnocení trendových parametrů komfortu pro sledování vývoje komfortních vlastností vozidla.

3.5 Externí knihovny

Pro potřeby vyhodnocení jsou použity tyto externí a interní python knihovny:

Knihovny	*Použití*	*Popis*
SciPy	comf	https://scipy.org/ Knihovna pro práci se signály a jejich filtrací
NumPy	stab, comf, flat	https://numpy.org/ Knihovna pro matematické vypočty a operace
pandas	flat, client_io	https://pandas.pydata.org/ Knihovna pro práci s datovými strukturami
os	stab, flat, client_io	Standardní modul pythonu pro práci s adresářovou strukturou
datetime	stab, flat, client_io	Standardní modul pythonu pro práci s formáty datumů

json	stab, flat, client_io	Standardní modul pythonu pro práci s json formátem dat
matplotlib.pyplot	client_io, stab	Podpůrná knihovna pro vykreslení grafických výstupů. Pro přímý výpočet není potřeba ji použít.

3.6 Nutný externí SW potřebný pro funkci

Pro funkci celého systému je nutné mít nainstalovaný následující SW. Jedná se zejména o zabezpečení funkcí souvisejících s databází.

Název	*Použití*	*Odkaz*
XAMPP	- Databáze MariaDB - Web server pro běh PHP - PHPMYADMIN pro správu databáze	https://www.apachefriends.org/
Python	- Běh skriptů pro vyhodnocení a import dat - Automatické zpracování databáze	https://www.python.org/
GRAFANA	- **Jedná se pouze o doporučený nástroj** - pro vizualizaci a verifikaci výsledků	https://grafana.com/

Externí knihovny ani externí programy nutné pro fungování SW nejsou součástí distribuovaného balíku a je nutné si je doinstalovat zvlášť v souladu s jejich licenčními podmínkami.

4 Práce s client knihovnou (**ROBD.client_io**)

Knihovna slouží pro práci s naměřenými daty uloženými na stolním počítači. Obsahuje třídy:

- **CSV** – načtení dat z csv souboru (4.1)
- **PRE** – příprava dat před vyhodnocením (4.2)
- **PLOT** – vykreslení výsledků do grafů (4.3)
- **SAVE** – uložení vysledků diagnostiky (4.4)

Příklad importu knihovny:

```
import ROBD.client_io
nebo
from ROBD import client_io
```

4.1 Třída pro načtení csv (**ROBD.client_io.CSV**)

Třída slouží k načtení csv souborů, který má definovaný formát (viz obr. 1). První řádek obsahuje identifikaci snímače a vzorkovací frekvenci. Druhý řádek názvy proměnných v tabulce a od třetího řádku dále jsou uvedena konkrétní hodnoty proměnných. Jejich pořadí i oddělovač lze v rámci načtení dat definovat, není tedy závazné.

	A	B	C	D	E	F	G
1	94-54-153	022-u_HV11_B814_Y; fvz = 209 Hz					
2	datetime;time [s];V [km/h];LAT;LON;HV11_R1_Y [m/s2]						
3	2025-07-21 06:29:14.486;0.0000;4.5;49.347248;16.417814;-0.068700						
4	2025-07-21 06:29:14.490;0.0048;4.5;49.347248;16.417814;0.206000						
5	2025-07-21 06:29:14.495;0.0096;4.5;49.347248;16.417814;0.019600						
6	2025-07-21 06:29:14.500;0.0144;4.5;49.347248;16.417814;-0.107900						
7	2025-07-21 06:29:14.505;0.0191;4.5;49.347248;16.417814;-0.127500						
8	2025-07-21 06:29:14.510;0.0239;4.5;49.347248;16.417814;0.058900						
9	2025-07-21 06:29:14.514;0.0287;4.5;49.347248;16.417814;-0.107900						

Obr. 1 Ukázka csv souboru pro načtení dat

Příklad načtení třídy:

```
from ROBD.client_io import CSV
```

4.1.1 Metoda **load**

Třída obsahuje jednu zásadní metodu **load**, která obsahuje algoritmus pro načtení dat. Pomocí této metody je možné načítat také data uložená jako soubor formátu txt.

Příklad syntaxe:

```
loaded_data = CSV.load(filename, columns_order, separator)
```

Vstupy		
filename	string	Cesta k *.csv nebo *.txt souboru s daty
columns_order	dictionary	Uživatel může definovat pořádí sloupců v *.csv souboru. Jsou ale konkrétně definované klíče (key) pro slovník. Vice viz
separator	string	Definuje oddělovač použitý v *.csv souboru
Výstup		
loaded_data	dictionary	Slovník obsahuje položku *meta,* kde jsou uložena data o snímači a položku *data*, která obsahuje načtené vektory dat podle columns_order.

Columns_order – pořadí sloupců

Jedná se o konkrétně definovaný slovník, kde číslo u jednotlivých klíču označuje pořadí sloupce v *.csv souboru. Pokud není zadáno uživatelem, použije se toto základní nastavení:

```
columns_order = {
   “abstime“ : 1,
   “speed“   : 2,
   “lat“     : 3,
   “lon“     : 4,
   “acc“     : 5}
```

Klíče „lat“ a „lon“ slouží ke čtení GPS polohy vozidla. Ta je důležitá při trendové diagnostice stability či komfortu, kdy je vhodné porovnávat výsledky ze shodné lokality. Pro varovnou diagnostiku stability nemají význam.

Pokud tato data o poloze nejsou dostupná, následné vyhodnocení proběhne bez závislosti na poloze vozidla, tedy budou analyzována všechna načtená dat.

Separator – oddělovač dat

Je možné zvolit jiný oddělovač dat, který je ve vstupní *.csv souboru použit. Pokud není hodnota uvedena, je použito základní nastavení:

```
separator = “;“
```

Příklad použití

Příklad použití pro načtení dat ze dvou sensorů “B1” a “B2” do proměné **loaded_data**:

```
from ROBD.client_io import CSV
columns_order = {“abstime“: 1, “speed“: 3, “acc“: 6}
sensors = [‘train1_wagon1_B814_Y’, ‘train1_wagon1_B824_Y’]
loaded_data = {}
for sen in sensors:
      filename = “data/”+sen+”.csv”
      loaded_data[sen] = CSV.load(filename, columns_order)
```

Pro vyhodnocení komfortu je nutné znát příčný i svislý směr zrychlení, proto je třeba tímto způsobem načítat data sice z jednoho snímače, ale ve dvou směrech, například tedy definovat:

```
sensors = [‘train1_wagon1_B825_Y’, ‘train1_wagon1_B825_Z’]
```

4.2 Třída pro přípravu dat (**ROBD.client_io.PRE**)

Třída PRE slouží k rozdělení načtených dat již popsanou funkcí **CSV.load**. Obsahuje metody pro rozdělení pouze dat z jednoho snímače, ale také z více snímačů zároveň pokud je metodou load načteno více dat najedou (viz příklad použití výše).

Příklad načtení třídy:

```
from ROBD.client_io import PRE
```

4.2.1 Metoda **cut_one_sensor**

Tato metoda zpracuje načtená data zmíněnou metodou **ROBD.client_io.CSV.load**. Data jsou rozdělena do stejně velkých bloků jejich délka je stanovena v konfiguračním souboru (pro stabilitu typicky 512 vzorků dat a pro komfort 1024 vzorků dat). V těchto blocích je zachován celkový průběh signálů zrychlení. Rychlost jízdy a GPS souřadnice jsou stanoveny z hodnot na začátku blocku.

Základní syntaxe:

```
cutted_data = PRE.cut_one_sensor(loaded_data, typ)
```

Vstupy		
loaded_data	dictionary	Slovník obsahující informace o snímači (klíč *meta*) a data (klíč *data*). Konkrétně jde o výstup z metody **ROBD.client_io.CSV.load** (viz 4.1.1).
typ	string	Uvádí se **typ="stab"** pro přípravu dat pro vyhodnocení stability nebo **typ="comf"** pro přípravu dat pro vyhodnocení komfortu.
Výstup		
cutted_data	list	Seznam, ve kterém je jedna položka typu slovník (dict), kde jsou načteny data o snímači převzaté z načtených souborů a také definovaná část signálu zrychlení.

Příklad použití:

```
filepath = way+"train1_wagon1_B814_Y_20100721_0727.csv"
loaded_data = CSV.load(filepath, columns_order)
cutted_data = PRE.cut_one_sensor(loaded_data, typ="stab")
```

V případě použití této metody pro přípravu dat na vyhodnocení **komfortu** jsou ve výstupním souboru uvedeny dva vektory dat pro příčný a svislý směr, a to z jednoho snímače.

4.2.2 Metoda **cut_few_sensors**

Tuto metodu lze využít pokud je metodou **CSV.load** načteno více signálů, resp. signály s více snímačů (viz **Příklad použití** v 4.1.1). Vstupní data rozdělí podle klíče (key) a pro každý spustí metodu **cut_one_sensor.**

Základní syntaxe:

```
cutted_data = PRE.cut_few_sensors(loaded_data, typ)
```

Parametry **loaded_data** a **cutted_data** jsou formátu slovník (dictionary), kdy jednotlivé klíče připadají jednotlivým snímačům. Uvnitř je poté seznam (list) rozdělených dat stejně v metodě pro jeden snímač (viz předchozí podkapitola). Parametr **typ** (formátu string) zůstává stejný.

4.3 Třída pro vykreslení grafů (**ROBD.client_io.PLOT**)

Třída sloužící pro vykreslení a uložení grafů dlouhodobé trendové diagnostiky stability a komfortu.

Příklad načtení třídy:

```
from ROBD.client_io import PLOT
```

4.3.1 Metoda **stab**

Metoda pro zobrazení a uložení grafu dlouhodobých trendů stability vozidla. V grafu jsou zobrazeny základní ukazalete (parametry).

Základní syntaxe:

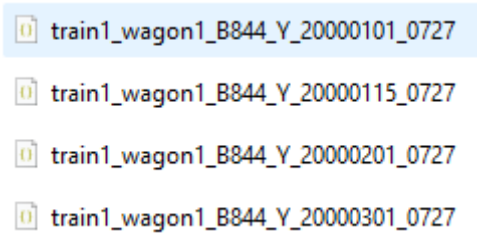
```
PLOT.stab(sensor, path, show_plot)
```

Vstupy		
sensor	string	Název snímače, jehož data mají být zobrazena, resp. označení adresáře.
path	string	Cesta k adresáři se soubory. Pokud není zadána, tak platí **path = “export/stability - trend/”**
show_plot	boolean	**True** zobrazí graf, **False** nezobrazí graf. V obou případech dojde k uložení grafu v png.
Výstup		
Výstupem je pouze vytvoření adresářů a souborů s výsledky diagnostiky.		

Příklad použíti:

```
client_io.PLOT.stab("train1_wagon1_B844_Y", path=”export/stability trend”,
                    show_plot=True)
```

V adresáři *export**stability - warning*\ *train1_wagon1_B844_Y* jsou uložena data viz obr. 2. Načítány jsou všechny soubory, které odpovádají identifikaci snímače.



Obr. 2 Příklad názvů souborů pro dlouhodobé trendy

Uvedené soubory a jejich názvy se tvoří automaticky použitím třídy **SAVE**, která je popsání dále.

4.3.2 Metoda **comf**

Metoda pro zobrazení a uložení grafu dlouhodobých trendů komfortu vozidla. V grafu jsou zobrazeny základní ukazalete (parametry).

Základní syntaxe:

```
PLOT.comf(sensor, path, show_plot)
```

Vstupy		
sensor	string	Název snímače, jehož data mají být zobrazena, resp. označení adresáře s daty.
path	string	Cesta k adresáři se soubory. Pokud není zadána, tak platí **path = “export/comfort/”**
show_plot	boolean	**True** zobrazí graf, **False** nezobrazí graf. V obou případech dojde k uložení grafu v png.
Výstup		
Výstupem je pouze vytvoření adresářů a souborů s výsledky diagnostiky.		

Příklad použití:

```
client_io.PLOT.comf("train1_wagon1_B825", path=”export/comfort”, show_plot=True)
```

V tomto případně není u označení snímače uveden směr, jelikož pro vyhodnocení komfortu je analyzování více směrů zrychlení zároveň. Cesta *export/comfort/ train1_wagon1_B825* tedy vede do adresáře s veškerými výslednými soubory k tomuto snímači. Uvedené soubory a jejich názvy se tvoří automaticky použitím třídy **SAVE**, která je popsání dále.

4.4 Třída pro uložený vyhodnocených dat (**ROBD.client_io.SAVE**)

Třída **SAVE** slouží k uložení výsledků diagnostiky do adresářů a souborů v počítači. Obsahuje pouze jednu metodu **json** pro uložení dat v textovém formátu .json.

Příklad načtení třídy:

```
from ROBD.client_io import SAVE
```

4.4.1 Metoda **json**

Metoda pracuje s výsledky diagnostiky, ve který je definována informace typu diagnostiky (*stability – trend*, *stability – warning* nebo *comfort*). Podle tohoto typu metoda pozná s jakým formátem dat pracuje a podle toho zvolí jejich zpracování a formát uložení – zavolá další dílčí metody.

Základní syntaxe:

```
SAVE.json(export_data, folder_name)
```

Vstupy		
export_data	dictionary	Slovník obsahující informace o typu diagnostiky, vozidel, snímači a výsledné hodnoty sledovaných parametrů. Konkrtění obsah se liší podle typu diagnostiky.
folder_name	string	Lze definovat složku, do které budou data uložena. Pokud není definováno, tak **folder_name = “export”**.
Výstup		
Výstupem je pouze vytvoření adresářů a souborů s výsledky diagnositky.		

Dílčí výsledky uložené touto metodou slouží k následnémů vykreslování a analýze dlouhodobých trendů.

5 Vady kol (**ROBD.flat**)

Balík funkcí, které umožňují detekci vad na jízdní ploše kol pomocí několika druhů přístupů vyhodnocení:

- CALC_FLAT_AKF – jednoduchá detekce vad,
- FLAT_FTR11 – detekce pomocí strojového učení v časové oblasti,
- FLAT_FTR38 – detekce pomocí strojového učení v časové i frekvenční oblasti.

5.1 Metoda CALC_FLAT_AKF a CALC_FLAT_AKF_PLOT

Jedná se přímo o metodu obsahující vyhodnocení jednoho bloku dat z jednoho snímače. Délka a vzorkovací frekvence bloku dat je daná parametrem v rámci konfigurace detekce (v souboru **flat_config_AKF.ini**).

Základní syntaxe:

```
ouput = CALC_FLAT_AKF(block, speed, Dk)
```

Vstupem je signál, ale také odhad rychlosti a průměru kola. Tyto odhady by měly být v rozmezí 0,75 až 1,5 násobku skutečné hodnoty.

Vstupy		
block	block	Blok dat obsahují signál zrychlení. Blok je definován právě délkou tohoto signálu.
speed	float	Odhad rychlosti. Pokud není rychlost známa, zadat 0.
Dk	float	Odhad průměru kola, pokud není znám, pak se zadá 0.
Výstup		
output	dictionary	Slovník výsledků pro analyzovaná data. Obsahuje výsledky: - **Flat**: hodnocení závažnosti ploch na kolech v rozmezí 0 až 1. - **Peaks**: hodnocení přítomnosti rázů z koleje (kladné číslo).

Základní syntaxe pro grafický výstup:

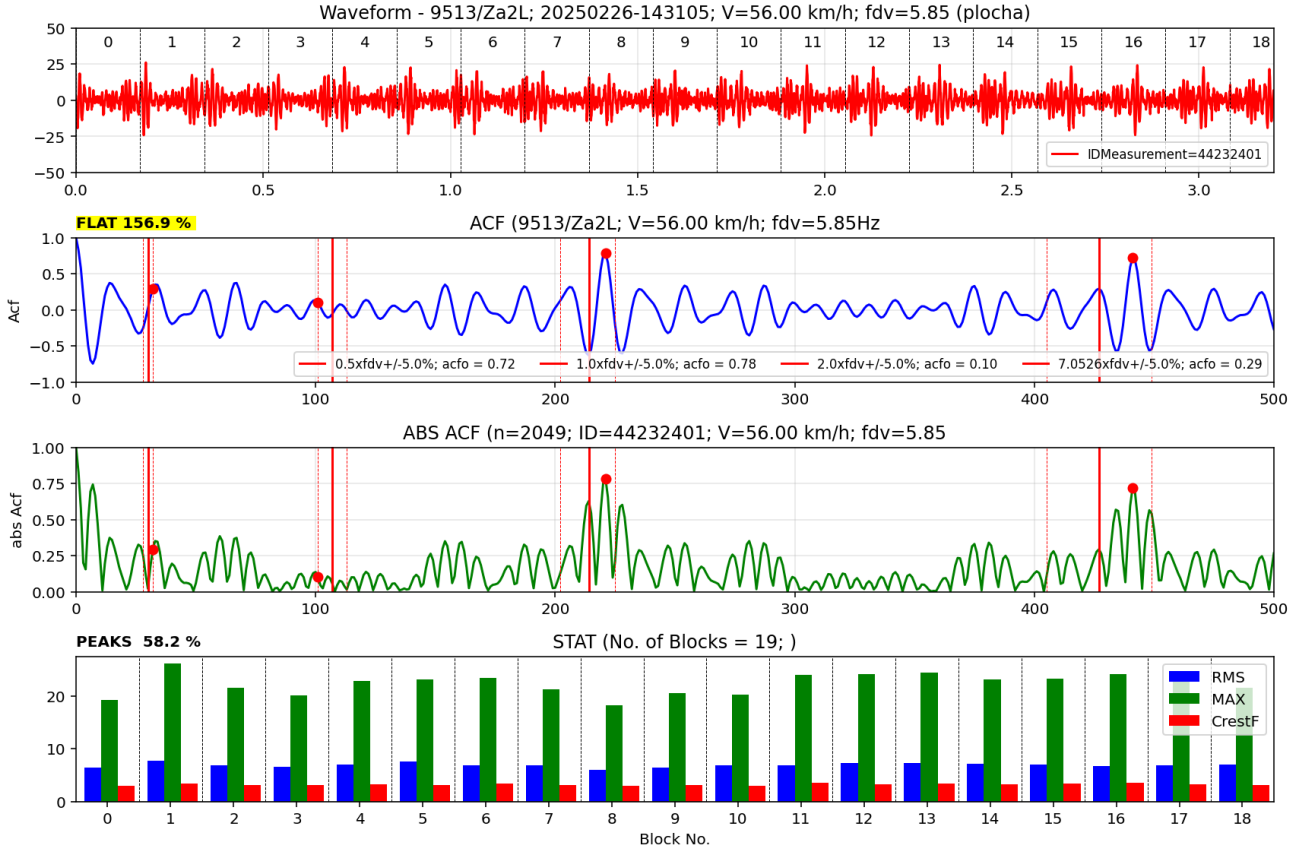
```
ouput = CALC_FLAT_AKF_PLOT(block, speed, Dk, file)
```

Výpočet a výstup funkce je stejný jako u **CALC_FLAT_AKF** s tím rozdílem, že parametr **file** udává, kam se má uložit grafický výstup výpočtu.

Vstupy		
block	block	Blok dat obsahují signál zrychlení. Blok je definován právě délkou tohoto signálu.
speed	float	Odhad rychlosti. Pokud není rychlost známa, zadat 0.
Dk	float	Odhad průměru kola, pokud není znám, pak se zadá 0.
file	String	Grafický výstup výpočtu ve formátu PNG.

Výstup		
output	dictionary	Slovník výsledků pro analyzovaná data. Obsahuje výsledky: - **Flat**: hodnocení závažnosti ploch na kolech v rozmezí 0 až 1. - **Peaks**: hodnocení přítomnosti rázů z koleje (kladné číslo).

Příklad grafického výstupu:



5.2 Balík FLAT_FTR11

FLAT_FTR11 obsahuje balík funkcí pro strojové učení s velikostí vstupního vektoru o 11 prvcích. Primárně se jedná o hodnocení dat z tabulky **PREFIX_STAT** (viz příloha 2).

Definice modelu je uložena ve formátu JSON v souboru ***.model** s následujícími parametry:

```
{ "type": "model config flat_FTR11",
  "cnt": 11,
  "cnt1": 11,
  "speed": 0.5,
  "eps": 0.001,
  "epochs": 100000,
  "batch": 1000,
  "normFile": "normal_10000u.norm",
  "feature_names": ["min","max","mean","peak","rms","std","swn","kur","cf","ipf","ene"]
}
```

Hodnoty cnt a cnt1 určují velikost skrytých vrstev. Parametr speed pak rychlost učení. Zadává se také počet epoch a maximální počet batch po kterých se tréning ukončí.

Důležitým parametrem je odkaz na soubor ***.norm**, který definuje parametry pro normalizaci vstupů.

```
{
  "type": "minmax",
  "features": ["min","max","mean","peak","rms","std","swn","kur","cf","ipf","ene"],
  "params": {
    "min": [-100.0, 0.0,  0.0, 0.0,  0.0,  0.0, -1.0,   0.0,  0.0,  0.0, 0.0],
    "max": [    -0, 100, 0.01, 100, 15.0, 15.0,  1.0, 100.0, 20.0, 40.0, 5e5]
  }
}
```

Pro výpočet je potřeba také definovat vstupní a výstupní tabuku databáze. Pro definici solveru složí soubor ***.solver**:

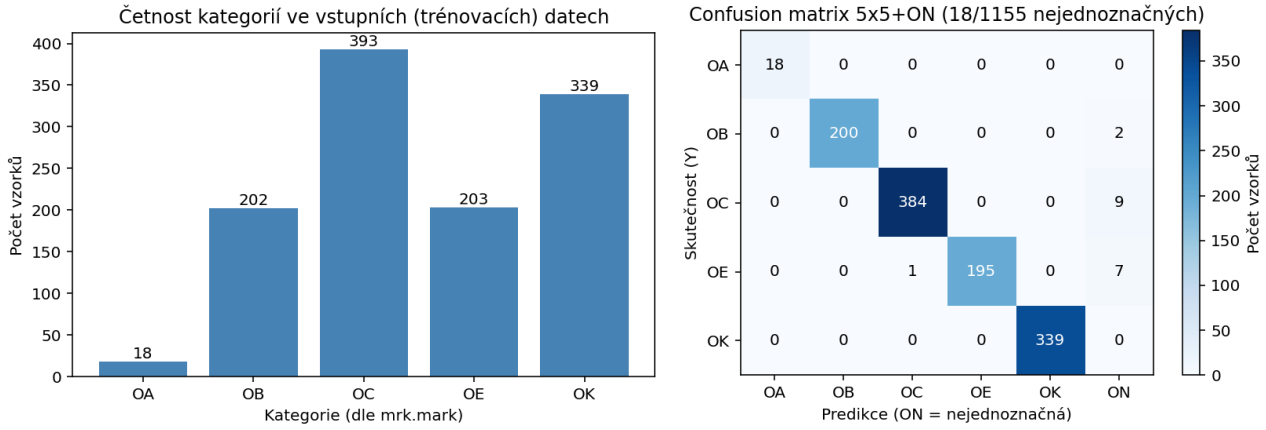
```
{
  "type": "solver config flat_FTR11",              název modelu - typ
  "modelname": "model5_FTR_11",                    prefix souborů
  "modeldir": "Tmodels",                           název drasáře relativně skriptu
  "limit": 10000,                                  limit pro batch
  "tabin": "skf21mt_stat",                         vstupní tabulka
  "lasti": 2815132,                                poslední načtena hodnota (index i)
  "tabout": "skf21mt_ftr11",                       výstupní tabulka
  "colout": "flats"                                výstupní sloupec
}
```

5.2.1 Tréning

Pro trénink modelu se použije skript **Train_FTR11.py:**

Vstupy		
file	string	Název souboru ***.solver**, další soubory jsou načteny pomocí „**modelname**“ z adresáře „**modeldir**“. Adresář musí obsahovat soubory ***.model** a ***.norm**.
Výstup		
***.pth**	file	Výsledkem učení je soubor ***.pth** v adresáři „**modeldir**“
***_train_log.csv**	file	Log průběhu učení
***_matrix.png**	file	Confusion matrix výsledku učení
***_input.png**	file	Četnosti jednotlivých kategorií vstupů
***_wrong.csv**	file	Seznam IDmeasurements, které nebyly správně klasifikovány.

Ukázka výstupních souborů:



5.2.2 Detekce z DB

Detekce pomocí naučeného modelu se provede pomocí skriptu **Detect_FTR11.py**, kdy jediným vstupním parametrem je počet detekcí v jednom běhu. Skript načte zadaný počet záznamů od posledního detekovaného („lasti“ ze souboru *.solver), provede detekci a výsledek uloží do výstupní tabulky („tabout“ ze souboru *.solver). Po provedení se aktualizuje hodnota „lasti“ pro další běh. Skript nemá žádný grafický výstup. Všechna data vstupují a vystupují z databáze.

Vstupem je ale soubor *.solver a v něm odkazované soubory *.norm a *.model v příslušném adresáři.

Pro výpočet je nutné, aby byl model natrénovaný, tedy aby adresář obsahoval soubor *.pth !

5.2.3 Detekce ze signálu

Detekce pomocí naučeného modelu se provede pomocí skriptu **Detect_raw_FTR11.py**, kdy vstupem je odkaz na soubor *.solver a vstupní data. Výstupem je hodnota příznaku poruchy v podobě vektoru hodnot tak, jak by se uložily v tabulce „tabout“.

5.2.4 Formát výstupu detekce

Detekce určí pravděpodobnost nálezu jednotlivých kategorií poruch. Finální výsledek pro vizualizaci a další zpracování je uložen v hodnotě „result“. Ostatní parametry jsou pouze informativní pro kontrolu a případnou analýzu v případě nevalidního výsledku. V ideálním případě se předpokládá, že bude detekován pouze jeden z výstupů OA, OB, OC, OE, OK jako dominantní. Pokud se tak stane, je výsledek validní. Pokud ne, pak je výsledek neurčitý a výstup ON je nastaven na hodnotu 1 – takový výsledek by se měl ignorovat.

```
Result                              výsledná detekce (hodnota 0 až 5)
OA, OB, OC, OE, OK,                 výstupy NN pro jednotlivé kategorie (hodnoty 0 až 1)
ON                                  0 – validní výstup, 1 – vadný výstup
```

5.3 Balík FLAT_FTR38

FLAT_FTR38 obsahuje balík funkcí pro strojové učení s velikostí vstupního vektoru o 38 prvcích. Primárně se jedná o hodnocení dat z tabulky **PREFIX_STAT** ale navíc z tabulky **PREFIX_CPB** (viz příloha 2). Všechny názvy souborů a jejich použití je stejné jako u FTR11. Odlišností je velikost vstupního vektoru. Prvních 11 prvků je stejných, vektor je ale rozšířen o 27 prvků z vektoru tabulky CPB.

Definice modelu je uložena ve formátu JSON v souboru ***.model** s následujícími parametry:

```
{ "type": "model config flat_FTR38",
  "cnt": 38,
  "cnt1": 38,
  "speed": 0.5,
  "eps": 0.001,
  "epochs": 100000,
  "batch": 1000,
  "normFile": "normal_10000u.norm",
  "feature_names": ["min","max","mean","peak","rms","std","swn","kur","cf","ipf","ene"]
}
```

Názvy skriptů jsou tedy obdobné: **Train_FTR38.py**, **Detect_FTR38.py**, **Detect_raw_FTR38.py.**

Výstupní vektor je stejný jako u FTR11 a je ukládán do zadané tabulky.

6 Stabilita (**ROBD.stab**)

Balíček definuje funkce pro vyhodnocení stability jízdy vozidla. Cílem je diagnostikovat stav vozidla z pohledu stability, ale také mít možnost okamžitě zasáhnout v případě překročení limitů stability. Balíček obsahuje třídy:

- **Trend** – pro vyhodnocení trendových parametrů pro sledování stavu stability vozidla (6.3)
- **Warn** – pro zaznamenávání aktuálního hlášení stavu stability vozidla (6.4)

Jsou stanoveny 3 ukazatele komfortu, které jsou vyhodnocovány z hodnot příčného zrychlení na rámu podvozku a v jednom případě i na aktuální rychlosti jízdy vozidla:

- **ratio** [-] – poměr aktuálně spočítané RMS hodnoty příčného zrychlení vůči jeho limitní hodnotě
- **prob** [-] – pravděpodobnost nestability, která zahrnuje do výpočtu rychlost jízdy vozidla, kdy limitní případ je nestabilní jízda při zvolené maximální rychlosti jízdy
- **pred** [s] – prediktivní ukazatel, která pracuje s určitým počtem výsledků z minulých úseků a tvoří lineární aproximaci, pomocí které odhaduje čas do nestability

6.1 Konfigurace pro vyhodnocení stability (**stab_config.ini**)

Jedná se o samostatný soubor *stab_config.ini*, který je místěn do podadresáře *config*. Obsahuje základní konfigurační parametry pro vyhodnocení stability jízdy vozidla. Je rozdělen na bloky: GENERAL, ALARMS, PREDICTION, FILTER, NOTES.

6.2 Nahrání dílčího balíčku (**import**)

```
import ROBD.stab
nebo
from ROBD import stab
```

6.3 Trendová diagnostika – statistika – (**ROBD.stab.Trend**)

Výstupem trendové diagnostiky je slovník (dictionary), který obsahuje informativní data o vozidle a snímači, které přebírá ze vstupních dat, a také hodnoty charakteristických parametrů pro určení stavu stability.

Trendová diagnostika je zabalena ve třídě **Trend**, která obsahuje dílčí methody pro vyhodnocení:

- **CALC_few_sensors**
- **CALC_one_sensor**
- **CALC_one_block**

Příklad načtení třídy:

```
from ROBD.stab import Trend
```

6.3.1 Metoda **CALC_few_sensors**

Metoda pro souběžné vyhodnocení signálů z více snímačů. Jedná se o cyklické procházení a vyhodnocení dat ze snímačů, které lze samostatně vyhodnotit metodou **CALC_one_sensor** (6.3.2).

Základní syntaxe:

```
ouput = stab.Trend.CALC_few_sensors(cutted_data, rms_lim, GPS)
```

Metoda pouze rozdělí vstupní proměnnou **cutted_data** na data z jednotlivých snímačů, které pak samostatně vyhodnotí. Výsledky pak spojí do jednoho výstupu **output**.

Vstupy		
cutted_data	dictionary	Slovník dat získaný pomocí např. metody **PRE.cut_few_sensors** třídy **Client** (4.2.2).
rms_lim	float	Limitní RMS hodnota pro signál zrychlení, kdy při jejím překročení mluvíme o nestabilitě, např. ***rms_lim=5.5***. Může být závislá na parametrech podvozku, a tedy rozdílná pro jednotlivé snímače.
GPS	list	Oblast zájmu pro vyhodnocení. Čtyři hodnoty polohy GPS. Více viz 0.
Výstup		
output	dictionary	Slovník sledovaných statistických parametrů pro každý snímač. Tento výstup lze použít pro další trendovou dlouhodobou diagnostiku. Obsahuje také zakladní údaje o vozidle a snímači, které přebírá z **cutted_data**.

6.3.2 Metoda **CALC_one_sensor**

Metoda postupně prochází jednotlivé bloky dat v proměnné **cutted_data**. Bloky jsou již předem definované při načítání dat. Ty lze načíst metodou **PRE.cut_one_sensor** třídy **Client**. Další popis vstupů a výstupu je uveden již v 6.3.1 s tím, že zde je proměnná **cutted_data** formátu list.

Pro samotné vyhodnocení sledovaných parametrů volá metoda dílčí metodu **CALC_one_block** (6.3.3), která vyhodnocuje konkrétní bloky data.

Základní syntaxe:

```
ouput = stab.Trend.CALC_one_sensor(cutted_data, rms_lim, GPS)
```

6.3.3 Metoda **CALC_one_block**

Jedná se přímo o metodu obsahující vyhodnocení jednoho bloku dat z daného snímače. Velikost bloku dat je daná parametrem v rámci konfigurace stability (v souboru **stab_config.ini**, 6.1).

Metoda pracuje s proměnou memory, která slouží především pro vyhodnocení parametru nazvaného ***pred***, pro prediktivní diagnostiku. Tento parametr pro své vhodnocení potřebuje několik posledních hodnot vybraných veličin. Jejich počet je definován také v rámci konfigurace stability (v souboru *stab_config.ini*, 6.1).

Základní syntaxe:

```
ouput, memory = stab.Trend.CALC_one_block(block, memory, rms_lim, GPS)
```

Vstupy		
block	dictionary	Slovník dat obsahují informace o vozidle, snímači a data představující signál zrychlení. Blok je definován právě délkou tohoto signálu.
memory	dictionary	Obsahuje parametr *info* (list), který slouží pro záznam informace, zda byly splněny podmínky rychlosti a GPS. Dále obsahuje parametr *pred* (dict), který ukládá potřebné veličiny pro vyhodnocení prediktivní diagnostiky.
rms_lim	float	Limitní RMS hodnota pro signál zrychlení, kdy při jejím překročení mluvíme o nestabilitě, např. ***rms_lim=5.5***. Může být závislá na parametrech podvozku, a tedy rozdílná pro jednotlivé snímače.
GPS	list	Oblast zájmu pro vyhodnocení. Čtyři hodnoty polohy GPS. Více viz 0.
Výstup		
output	dictionary	Slovník sledovaných statistických parametrů pro daný blok dat (**block**).
memory	dictionary	Po proběhnutí metody jsou do proměnné memory přidány informace, které budou vstupem pro následující blok vyhodnocení.

6.3.4 Omezení vyhodnocení polohou a rychlostí jízdy

V rámci vyhodnocení je ověřována podmínka minimální rychlosti jízdy (**v_lim**) pro spuštění vyhodnocení a může být sledována poloha vozidla, pokud existují data o GPS poloze a je definována oblast zájmu.

GPS – oblast zájmu pro vyhodnocení trendové diagnostiky stability

Uživatel u trendové diagnostiky může definovat oblast zájmu pomocí souřadnic GPS. Formát parametrů je list, který má čtyři hodnoty: *[min(latitude), min(longitude), max(latitude), max(longitude)]*, které jsou zadané jako float.

Příklad:

```
GPS = [49.0923, 16.599, 49.1247, 16.610]
```

Pokud chceme vyhodnotit všechny nahraná data a není definována oblast zájmu je parametr GPS nezadávat, potom se automaticky definuje GPS oblast s hodnotami:

```
GPS = [0, 0, 0, 0]
```

v_lim – limitní rychlost pro spuštění vyhodnocení

Vyhodnocení stability jízdy vozidla se standardně vyhodnocuje pro vysoké rychlosti a v nízkých rychlostech postrádá význam. Proto je výpočtů zaveden parametr **v_lim**, který představuje limitní rychlost. Pokud je rychlost jízdy vyšší než tento limit, dojde ke spuštění výpočtů a záznamu výsledků. Hodnotu **v_lim** lze nastavit v konfiguraci (v souboru **stab_config.ini**, 6.1).

Rychlost **v_lim** by však neměla být nižší než 120 km/h, jelikož při nižších rychlostech může dojít k ovlivnění parametru *prob*, který je součástí vyhodnocení.

6.4 Varovná diagnostika – upozornění – (**ROBD.stab.Warn**)

Varovná diagnostika v každém dignostikovaném bloku dat ověřuje, zda bylo dosaženo nějaké z definovaných hladin, resp. hodnot ukazatelů. Pokud ano, tak vypíše hlášení např.:

“*2000-08-01 07:25:44 | B814 | prob = 0.54 - Zhoršené jízdní vlastnosti - hrozí neklidný chod při jízdě maximální rychlostí*”

Mezní hodnoty pro vypisování hlášení I samotné znění těchto slovních hlášek je možné upravovat v konfiguračním souboru *stab_config.ini* (6.1).

Varovná diagnostika je zabalena ve třídě **Warn**, která obsahuje dílčí methody pro vyhodnocení se shodnými názvy jako trendová diagnostika:

- **CALC_few_sensors**
- **CALC_one_sensor**
- **CALC_one_block**

Oproti trendové diagnostice zde není hlídána GPS poloha a spuštění diagnostiky je podmíněno pouze dosažením požadované rychlosti, která lze nastavit v konfiguračním souboru.

Příklad načtení třídy:

```
from ROBD.stab import Warn
```

6.4.1 Metoda **CALC_few_sensors**

Metoda pro souběžné vyhodnocení signálů z více snímačů. Jedná se o cyklické procházení a vyhodnocení dat ze snímačů, které lze samostatně vyhodnotit metodou **CALC_one_sensor** (6.4.2).

Základní syntaxe:

```
ouput = stab.Warn.CALC_few_sensors(cutted_data, rms_lim)
```

Metoda pouze rozdělí vstupní proměnnou **cutted_data** na data z jednotlivých snímačů, které pak samostatně vyhodnotí. Výsledky pak spojí do jednoho výstupu **output**.

Vstupy		
cutted_data	dictionary	Slovník dat získaný pomocí např. metody **PRE.cut_few_sensors** třídy **Client**.
rms_lim	float	Limitní RMS hodnota pro signál zrychlení, kdy při jejím překročení mluvíme o nestabilitě, např. ***rms_lim=5.5***. Může být závislá na parametrech podvozku, a tedy rozdílná pro jednotlivé snímače.
Výstup		
output	dictionary	Slovník obsahující varovná hlášení pro každý snímač. Hlášení jsou uložena v promněnné **notes** jako seznam (list) jednotlivých hlášení. Seznam zůstane prázdný, pokud nejsou překročeny sledované hodnoty ukazatelů. Obsahuje také zakladní údaje o vozidle a snímači, které přebírá z **cutted_data**.

6.4.2 Metoda **CALC_one_sensor**

Metoda postupně prochází jednotlivé bloky dat v proměnné **cutted_data**. Bloky jsou již předem definované při načítání dat. Ty lze načíst metodou **PRE.cut_one_sensor** třídy **Client**. Další popis vstupů a výstupu je uveden již v 6.4.1 s tím, že zde je vstupní proměnná **cutted_data** formátu list.

Pro samotné vyhodnocení sledovaných parametrů volá metoda dílčí metodu **CALC_one_block** (6.4.3), která vyhodnocuje konkrétní bloky data.

Základní syntaxe:

```
ouput = stab.Warn.CALC_one_sensor(cutted_data, rms_lim)
```

6.4.3 Metoda **CALC_one_block**

Jedná se přímo o metodu obsahující vyhodnocení jednoho bloku dat z daného snímače. Velikost bloku dat je daná parametrem v rámci konfigurace stability (v souboru **stab_config.ini**, 6.1).

Metoda pracuje s proměnou memory, která slouží především pro vyhodnocení parametru nazvaného ***pred***, pro prediktivní diagnostiku. Tento parametr pro své vhodnocení potřebuje několik posledních hodnot vybraných veličin. Jejich počet je definován také v rámci konfigurace stability (v souboru *stab_config.ini*, 6.1).

Základní syntaxe:

```
ouput, memory = stab.Warn.CALC_one_block(block, memory, rms_lim, GPS)
```

Vstupy		
block	dictionary	Slovník dat obsahují informace o vozidle, snímači a data představující signál zrychlení. Blok je definován právě délkou tohoto signálu.
memory	dictionary	Obsahuje parametr *info* (list), který slouží pro záznam informace, zda byly splněny podmínky rychlosti. Dále obsahuje parametr *pred* (dict), který ukládá potřebné veličiny pro vyhodnocení prediktivní diagnostiky.
rms_lim	float	Limitní RMS hodnota pro signál zrychlení, kdy při jejím překročení mluvíme o nestabilitě, např. ***rms_lim=5.5***. Může být závislá na parametrech podvozku, a tedy rozdílná pro jednotlivé snímače.
Výstup		
output	dictionary	Slovník s informace o daném bloku a seznamem varovných hlašení, která jsou do seznamu přidána, pokud dojde k překročení limitů sledovaných hodnot. Příklad takového hlášení je uveden v úvodu této podkapitoly 6.4.
memory	dictionary	Po proběhnutí metody jsou do proměnné memory přidány informace, které budou vstupem pro následující blok vyhodnocení.

6.4.4 Příklad použiti třídy **ROBD.stab**

V rámci toto příkladu jsou načtena dat z jednoho snímače pomoc metody **CSV.load**, a dále zpracována pomocí **PRE.cut_one_sensor**. Samotné vyhodnocení komfortu je provedeno metodou **stab.Trend.CALC_one_sensor** a **stab.Warn.CALC_one_sensor**. Výsledky jsou následně uložena metodou **SAVE.json**.

```
from ROBD.client_io import CSV, PRE, SAVE
from ROBD import stab as st

stab_filename = "wagon1_B814_Y_20200701_0727.csv"
columns_order = {"abstime": 1, "speed": 2, "lat": 3, "lon": 4, "acc": 5}
GPS = [49.0850, 16.100, 49.1000, 16.500]
rms_lim = 5.5

stab_loaded = CSV.load(stab_filename, columns_order)
stab_cutted = PRE.cut_one_sensor(stab_loaded, typ="stab")
stab_trend_result = st.Trend.CALC_one_sensor(stab_cutted, rms_lim, GPS)
stab_warn_result = st.Warn.CALC_one_sensor(stab_cutted, rms_lim)

SAVE.json(stab_trend_result)
SAVE.json(stab_warn_result)
```

7 Komfort (**ROBD.comf**)

Funkce pro vyhodnocení trendových parametrů komfortu pro sledování vývoje komfortních vlastností vozidla.

Hodnocení je založeno na parametrech definovaných normou ČSN EN 12299 s vědomím toho, že v běžném provozu nelze splnit všechny její požadavky. Důležitým prvkem této normy je nastavení váhové filtrace signálu. Tyto filtry jsou z normy převzaty a použity. Pro potřeby vyhodnocení je nutné analyzovat signály zrychlení na skříni vozidla ve svislém a příčném směru. Norma pracuje i s podélným směrem zrychlení, které ale není v této aplikaci používáno. Jelikož je tedy potřeba analyzovat 2 signály zároveň, tak již od začátku algoritmu je s těmito dvěma signály pracováno současně.

Balíček pro vyhodnocení komfortu obsahuje pouze jednu třídu:

- **Trend** – pro vyhodnocení základní charakteristik komfortu v definovaném úseku, které lze následně využít pro dlouhodobou trendovou diagnostiku (0)

Sledované parametry jsou normalizované hodnoty ukazatelů komfortu (hodnoty 0 až 1) pro jednotlivé směry a jejich vektorový součet:

- **comfort_y** – pro příčný směr
- **comfort_z** – pro svislý směr
- **comfort_yz** – pro kombinaci příčného a podélného směru

Sledováním vývoje těchto parametrů lze usuzovat na zhoršení či zlepšení jízdního komfortu, kdy hranice 1 znamení již nepohodlnou jízdu podle zmíněné normy výše. Jelikož ale nejsou dodrženy všechny požadavky normy, jde o orientační škálování výsledků.

7.1 Konfigurace pro vyhodnocení stability (**comf_config.ini**)

Jedná se o samostatný soubor *comf_config.ini*, který je místěn do podadresáře *config*. Obsahuje základní konfigurační parametry pro vyhodnocení komfortu jízdy vozidla. Je rozdělen na bloky: GENERAL, LIMITS, FILTER.

7.2 Nahrání dílčího balíčku

```
import ROBD.comf

nebo

from ROBD import comf
```

7.3 Vyhodnocení komfortu – (**ROBD.comf.Trend**)

Třída **Trend** obsahuje metody pro vyhodnocení dvou signálů zrychlení zároveň, což je pro hodnocení komfortu nezbytné. Třída má dvě základní metody:

- **CALC_one_sensor**
- **CALC_one_block**

Příklad načtení třídy:

```
from ROBD.comf import Trend
```

7.3.1 Metoda **CALC_one_sensor**

Metoda provádní vyhodnocení dat z jednoho snímače, což v tomto případě třídy pro komfort znamená vyhodnocení dvou signálů zrychlení. Bloky jsou již předem definované při načítání dat. Ty lze načíst metodou **PRE.cut_one_sensor** třídy **Client**, kde musí být definován typ jako *"comf"*, což zajistí právě zpracování dvou signálů. Vstupní proměnná v této metodě musí obsahovat načtená data obou signálů.

Základní syntaxe:

```
output = comf.Trend.CALC_one_sensor(cutted_data, GPS)
```

Vstupy		
cutted_data	dictionary	Slovník dat obsahují informace o vozidle, snímači a data představující oba signály zrychlení. Blok je definován právě délkou tohoto signálu.
GPS	list	Oblast zájmu pro vyhodnocení. Čtyři hodnoty polohy GPS. Více viz 0.
Výstup		
output	dict	Slovník sledovaných parametrů pro komfort jízdy vozidla. Tento výstup lze použít pro další trendovou dlouhodobou diagnostiku. Obsahuje také zakladní údaje o vozidle a snímači, které přebírá z **cutted_data**.

Samotné vyhodnocení signálů zrychlení a výpočet statistické hodnoty je provedeno dílčí metodou **CALC_one_block** (7.3.2). Sledované paramtery komfortu jsou vyhodnoceny až pro skupinu bloků v této metodě. Vypsané sledované parametry v proměnné **output** jsou popsány v úvodu této kapitoly.

7.3.2 Metoda CALC_one_block

Metoda slouží přímo pro vyhodnocení signálů zrychlení pro jeden definovaný block dat. Velikost tohoto bloku dat je stanovena v souboru *comf_config.ini*, podle kterého jsou data rozdělována. V rámci této metody jsou ověřovány podmínky rychlosti jízdy a GPS polohy. Pokud jsou splněny průběhne výpočet – fitrace dat a výpočet statistické hodnoty.

Základní syntaxe:

```
output = comf.Trend.CALC_one_block(block, GPS)
```

Vstupy		
block	dictionary	Slovník dat obsahují informace o vozidle, snímači a data představující signál zrychlení. Blok je definován právě délkou tohoto signálu.
GPS	list	Oblast zájmu pro vyhodnocení. Čtyři hodnoty polohy GPS. Více viz 0.
Výstup		
output	dict	Slovník obsahující statistické hodnoty pro filtrovaný signál zrychlení v příčném a svislém směru. Tento výstup je nutné dále zpracovat metodou **Trend.CALC_one_sensor**, který dopočítává sledované parametry komfortu.

7.3.3 Příklad použití třídy **ROBD.comf.Trend**

V rámci toto příkladu jsou načtena dat ze dvou souborů pomoc metody **CSV.load**, a dále zpracována pomocí **PRE.cut_one_sensor**. Samotné vyhodnocení komfortu je provedeno metodou **comf.Trend.CALC_one_sensor** a výsledky jsou následně uložena metodou **SAVE.json**.

```
from ROBD.client_io import CSV, PRE, SAVE
from ROBD import comf as cf

comf_filenames = [“wagon1_B845_Y_20200701_0727.csv”, “wagon1_B845_Z_20200701_0727.csv”]
columns_order = {“abstime”: 1, “speed”: 2, “lat”: 3, “lon”: 4, “acc”: 6}
GPS = [49.0850, 16.100, 49.1000, 16.500]

comf_loaded = {}
for filename in comf_filenames:
      comf_loaded[filename] = CSV.load(filename, columns_order)
comf_cutted = PRE.cut_one_sensor(comf_loaded, typ=”comf”)
comf_result = cf.Trend.CALC_one_sensor(comf_cutted, GPS)
SAVE.json(comf_result)
```

8 ROBD Desktop Client

ROBD Desktop Client je desktopová aplikace určena primárně pro servisní účely. Tato aplikace umožňuje:

- prohlížet primární data,
- výpočet a vizualizace FFT, CPB a filtrace signálu,
- vizualizaci výsledků,
- výběr a filtraci dat,
- export grafů (*.png) a primárních dat (*.csv),
- tvorbu, administraci a vizualizaci tabulek s příznaky,
- vizualizace trendu: statistiky, FFT, CPB,
- vizualizace X-Y grafů dvou veličin metadat a výsledků,
- vizualizace primárních dat do polárního grafu dle otáček kola,

8.1 Instalace

Jedná se o spustitelný soubor *.exe pro operační systém Windows 10 a 11 64bit. Program lze dodat také ve verzi pro Linux (testovány distribuce Debian a Ubuntu).

Program pro svou funkci potřebuje další soubory:

- **ROBD_desktop_client.ini** pro konfiguraci programu
- **exportNODE.csv** – buffer pro rychlejší načítání struktury sensorů
- **libmysql.dll** – knihovna pro přístup k databázi (není součástí distribuce SW)

Výše uvedené soubory musí být ve stejném adresáři jako spustitelný soubor.

8.2 Konfigurace

Základní konfigurace se provádí v souboru **setup.ini** v následujících kategoriích. Jedná se o nastavení programu, přístupu k databázi a také k definici exportu grafů.

```
[setup]
exportdata=1              # při exportu grafů exportovat i data
dataname=data             # postfix pro datové soubory exportu
exportdir=EXPORT1         # adresář pro export výsledků
server=127.0.0.1          # IP adresa serveru s databází

[domains]                 # domény diagnostiky: alias=název_tabulky
18Ev=18ev
21Mt=skf21mt
11Mt=11mt
8171M=8171m


[TimeChart]
en=1                      # zapnout export
w=1000                    # šířka v pixelech
h=300                     # výška v pixelech
```

```
y0=-30                  # min. rozsah Y osy
y1=30                   # max. rozsah Y osy
name=signal             # postfix souboru PNG

[TrendChart]
. . .                   # stejné jako v předchozím
name=trend              # postfix souboru PNG

[FFTChart]
. . .                   # stejné jako v předchozím
name=fft                # postfix souboru PNG
```

8.3 Ovládací prvky

Program obsahuje hlavní okno (**main**), okno pro grafy (**viewform**) a okno pro tabelární výpis výsledků (**tabform**). Při používání je doporučeno použít rozlišení obrazovky 1920 x 1080 nebo 1920 x 1200, při používání **viewform** je vhodné použít pracovní plochu se dvěma monitory.

Hlavní okno **main** je rozděleno do čtyř hlavních sekcí:

a) Nástrojová lišta (nahoře)
b) Filtr databáze (vlevo)
c) Panel metadat (vpravo)
d) Zobrazovací část (uprostřed)

Všechny panely jsou mezi sebou rozděleny pohyblivými přepážkami, tedy je možné je upravit podle aktuální potřeby uživatele a podle aktuálního rozlišení monitoru.

Vysvětlení jednotlivých funkcí (čísla odrážek odpovídají pozicím ve výše uvedeném obrázku):

1) Výběr databáze (diagnostické domény).
2) Připojení/odpojení od databáze (**connect** = odpojeno, **-OK-** = připojeno)
3) Filtr vlaků/vozidel/sensorů – lze vybrat více položek, při výběru vlaku se zobrazí pouze existující vozidla a sensory, záznamy jsou načteny z databáze (funkce DISTINCT), tedy jsou zobrazeny pouze existující položky. Při zprovozňování diagnostiky tak nemusí být výčet vozidel kompletní.
4) Filtry pro výběr záznamů:
    a. Filtr rychlosti a časového období (po kliknutí na datum se objeví kalendář pro výběr, vybere se datum "od" a následně datum "do"); filtr je aktivní pouze při zatržení příslušného checkboxu.
    b. Limit udává maximální počet načítaných položek, čím vyšší limit, tím delší čas při načítání. Filtr limit je nadřazen ostatním filtrům, protože omezuje počet výsledků.
    c. *Order* udává směr třídění záznamů, kdy "/\" = od nejmenšího, "\/" = od největšího; následně lze vybrat, která z položek je tříděná (*meta.MeasDate, meta.i, meta.IDMeasurement, meta.speed, meta.IDnode, stat.rms, stat.max, stat.min*)
    d. Filter sentence: jedná se o část databázového dotazu, který se složí z výše uvedených filtrů. Tento text lze upravovat, při čtení z databáze se použije toto pole. (pokud jsou zvoleny výše uvedené filtry, toto pole se průběžně upravuje, pokud je text v poli smazán, tak se při čtení neaplikuje žádný filtr). Lze tak vytvořit uživatelský filtr, například: *wagon = 7001 AND wheel = "Za1L"* nebo *meta.IDMeasurement = 42856121*. Názvy tabulek jsou uváděny bez prefixu, názvy sloupců odpovídají tabulkám (viz příloha 1).
    e. READ DB – tímto tlačítkem se odešle dotaz. Při získaní dat se vypíše jejich seznam v panelu (5). Pokud se mají načíst zároveň výsledky stability, komfortu a ploch, je nutné použít příslušné zatržítko. To umožňuje zrychlit načítání v případě, že není potřeba načítat výsledky.
5) Seznam nalezených záznamů. Každý záznam je ve formátu: IDmeasurement, date, time, wagon, bogie, sensor, speed; při dvojkliku na záznam se vykreslí jeho grafy a zobrazí metadata v pravém panelu. Pokud jsou načteny také výsledky, zobrazí se také výsledky.
6) FFT vybraného záznamu
7) CPB/AKF vybraného záznamu
8) Waveform vybraného záznamu
9) Pokud je vybráno více záznamů v seznamu (5), zvolen parametr hodnocení v seznamu (13) a spuštěn výpočet tlačítkem „Plot Trend", pak je v tomto grafu vykreslen trend těchto hodnot v čase a podle jednotlivých kol.
10) Values – vstupní parametry pro výpočet a výsledky pro vybraný záznam, Setup – nastavení programu, Log – záznam chodu programu.
11) Save – možnost uložení logu pro případnou diagnostiku.
12) Pole pro definici „postfix" názvů souborů při exportu.

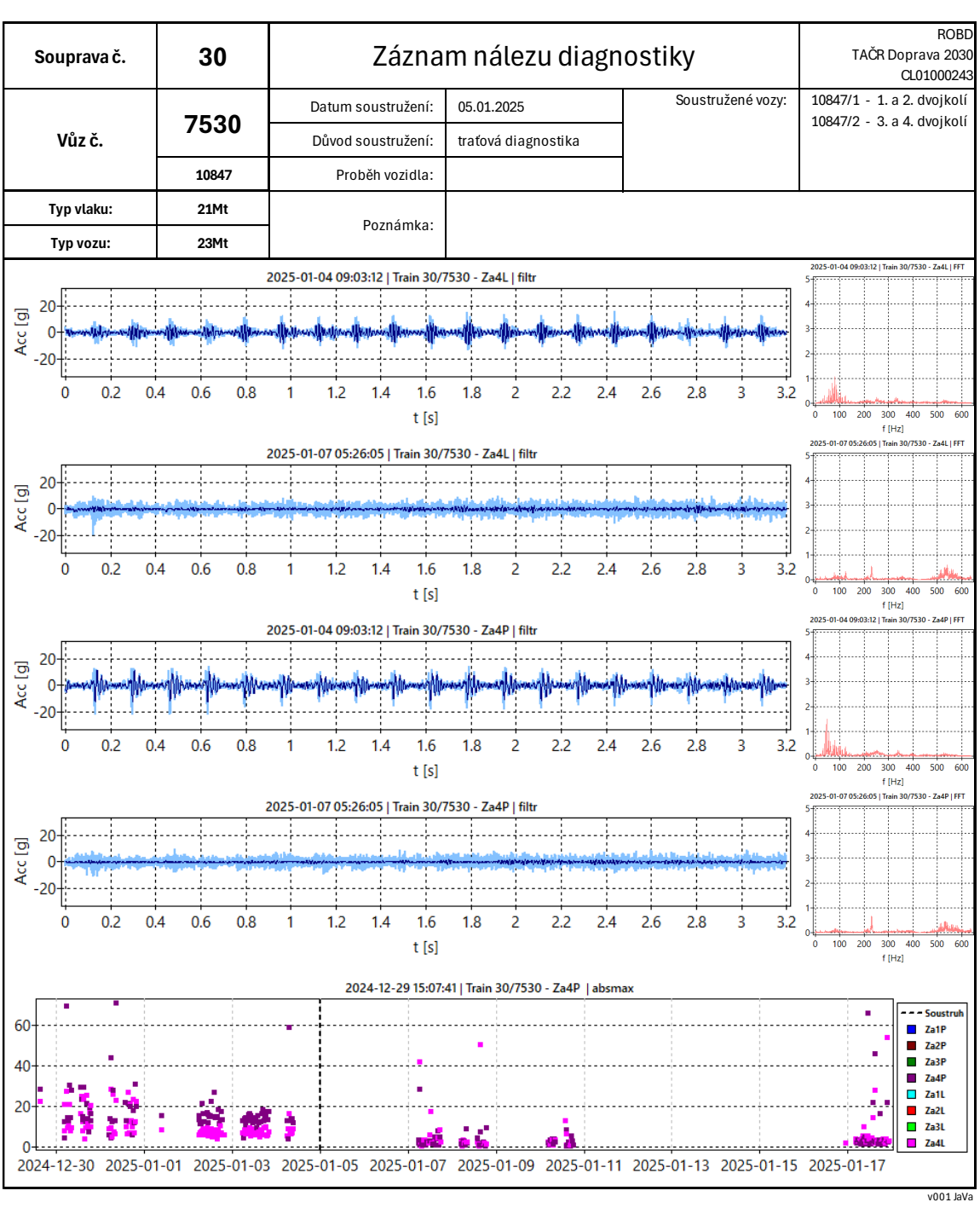
13) Nastavení vykreslení trendů.
14) Nastavení pro vykreslení polárních grafu – ruční korekce fáze
15) Volba tabulky markerů; LoadTab – načte do seznamu (5) všechny ID z marker tabulky; STAT – zobrazí statistiku marker tabulky v podobě četností jednotlivých příznaků.
16) Zadání markerů do zvolené tabulky a pro vybraný záznam.
17) Vizuální náhled na výsledky detekce (pokud ve výstupních tabulkách existuje). Číselně také v panelu (10) – values.
18) Nastavení uživatelského markeru do trendu (9).

8.4 Příklad exportu

Jednotlivé grafy lze exportovat a tvořit tak reporty pro jednotlivé případy poruch. Ty lze následně sdílet a diskutovat mezi zaměstnanci údržby. Níže jsou vedeny některé příklady.

Příklad trendu pro kola 4P a 4L vozu 7530. Spodní graf zobrazuje trend statistické hodnoty „max“ a svislá čárkovaná čára znázorňuje datum údržbového zásahu – reprofilaci.

Čtyři grafy pak zobrazují náhled waveform signálů před a po zásahu. Vpravo jsou uvedené FFT spektra příslušných signálů (lze zaměnit za CPB spektrum) nebo polární graf.



Příloha č.1

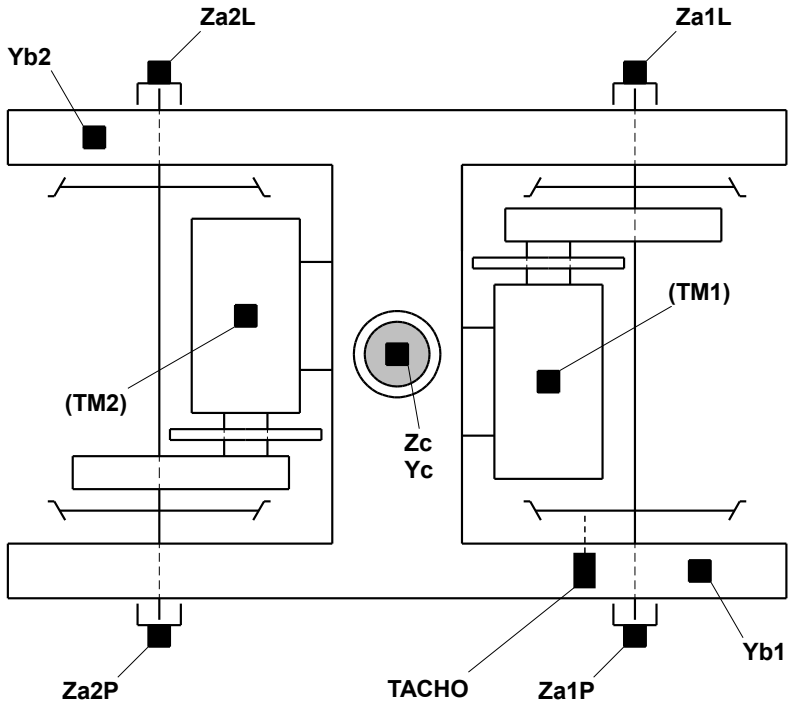
DEFINICE KONFIGURACE MĚŘÍCÍHO ŘETĚZCE

1 Měřící řetězec

Měření probíhá za účelem diagnostiky vad v pojezdu kolejového vozidla. Aby byla diagnostiky funkční, je nutné dodržet určitá pravidla při sběru dat. Při nedodržení minimálních požadavků na sběr dat nelze zaručit, že bude diagnostika funkční.

2 Umístění sensorů na vozidle

Následující obr.1popisuje generický návrh umístění sensorů. Jedná se o základní konfiguraci, kterou lze rozšiřovat a modifikovat podle účelu diagnostiky.



Obr. 3 Generický návrh umístění snímačů.

Rozdělení sensorů podle účelu diagnostiky je následující:

Označení	*Účel diagnostiky*	*Poznámka*
Za1L, Za1P, Za2L, Za2P ZaNS: N = č.nápravy S = strana	**Svislé snímače zrychlení (osa Z) umístěné na ložiskových skříních**, které slouží pro detekci vad na jízdních plochách kol. Pro účely pokročilé diagnostiky lze přidat také osu Y, která může složit pro diagnostiku trati z vozidla.	Příklad zobrazuje pouze dvě nápravy, při použití na reálném vozidle se doporučuje číslovat sensory 1 až 4 pro jednotlivé nápravy na vozidle, v případě jednotek se doporučuje rozdělit jednotku na jednotlivé vozy a ty číslovat dle reálného číslování na voze.
Yb1, Yb2 YbN	**Příčné snímače zrychlení (osa Y) umístěné na rámu podvozku**, který slouží pro hodnocení stability jízdy vozidla.	Předpokládá se, že snímač bude umístěn na podélníku rámu podvozku a to co nejdále od osy

N = č. přilehlé nápravy		podvozku v podélném i příčném směru. Umístění je také možné na čelníku podvozku.
Zc1, Yc1 ZcX, YcX X = č. podvozku	**Svislé a příčné snímače umístěné na skříni vozidla** (osy Z a Y), které slouží zejména pro hodnocení komfortu jízdy vozidla.	Předpokládá se umístění co nejblíže středu podvozku a co nejblíže podlaze vozidla. Této polohy však obvykle nelze přesně dosahnout, proto je potřeba navrhnout alternativní umístění.
TACHO	**Měření rychlosti vozidla** slouží pro určení očekávaných frekvencí vibrací rotačních částí. Dále pro eliminaci vlivu rychlosti na výsledek diagnostiky.	Měření lze provést přímo a nebo převzetím hodnoty z již existujících systémů (sběrnice vozidla). Výpočet rychlosti z GNSS systémů se nedoporučuje z důvodu nestability a nepřesnosti.
GNSS	**Určování polohy vozidla** pro účely třídění dat do referenčních úseků a případné eliminace negativních vlivů tratě.	Tento systém nelze použít u podzemních drah a v místě tunelů. K lokalizaci vozidel metra do jednotlivých mezistaničních úseků lze použít LTE/5G signál nebo nadřazené systémy vozidla. Systém určování polohy vozidla není zcela nutný pro diagnostiku, ale přináší další možnosti pro verifikaci a kontrolu trendových dat.

Označení snímačů je pouze doporučené. Zejména v případech, kdy se přebírají data z již existujících diagnostických systémů nebo měřicích zařízení, je vhodné použít již zavedené značení.

3 Vzorkovací frekvence a parametry digitalizace dat

Každý měřený kanál musí splňovat požadavky na kvalitu signálu dle Tab.1, jedná se o minimální (resp. doporučené) hodnoty. Požadavky jsou závislé na: umístění snímače (ložiska, rám, skříň), rychlosti jízdy vozidla, následném způsobu zpracování, účelem diagnostiky, parametrech vozidla. Nelze tak stanovit jednoznačné univerzální parametry. Níže jsou uvedeny parametry pro vozidla provozované do 80 km/h (regionální tratě, tramvaje a metro) a vozidla do 160 km/h (hlavní tratě a koridory). Pro vyšší a nižší rychlosti je potřeba parametry upravit, a to nejlépe na základě prvotních ~~průzkumných~~ naměrů.

Tab. 1 Přehled rozmístění sensorů pro výchozí návrh ROBD

Kanál	**Směr měření**	**Min. rozsah pro V_{max}**	**f_{vz}, ADC pro V_{max}**	**Popis**
Za1L	Z	± 200 g; 80 km/h ± 500 g; 160 km/h	1 kHz@16 bit 80 km/h 5 kHz@24 bit 160 km/h	Svislé zrychlení ložisková skříň 1L
Za1P				Svislé zrychlení ložisková skříň 1P
Za2L				Svislé zrychlení ložisková skříň 2L
Za2P				Svislé zrychlení ložisková skříň 2P
Yb1	Y	± 16 g	200 Hz@16 bit	Příčné zrychlení rámu nad 1. nápravou
Yb2	Y			Příčné zrychlení rámu nad 2. nápravou
Zc	Z	± 16 g	200 Hz@16 bit	Svislé zrychlení na skříni nad podvozkem
Yc	Y	± 16 g	200 Hz@16 bit	Příčné zrychlení na skříni nad podvozkem
TM1 [1)]	X/Z	± 32 g	5 kHz@16 bit 80 km/h [1)] 10 kHz@24 bit 160 km/h [1)]	Vibrace trakčního motoru 1. nápravy
TM2 [1)]				Vibrace trakčního motoru 2. nápravy

4 Sběr dat

4.1 Aktivace měření

Provozovatel systému musí zabezpečit vhodný management měření. Triggerování měření lze provádět na základě:

- rychlosti vozidla,
- lokality dle polohy z GNSS,
- časového intervalu,
- kilometrického intervalu.

Podmínky lze kombinovat a vytvořit navzájemné podmínky, např.:

- aktivace v určité lokalitě v případě, že od posledního měření proběhl daný časový úsek,
- aktivace v referenčním úseku při dosažení minimální rychlosti,
- aktivace při zadaných číslech vlaků (dle GVD) a dané lokalitě.

Délka záznamu může být opět omezena splněním určitých podmínek. Ukončení měření lze podmínit např.:

- délkou záznamu na zadanou hodnotu času nebo dráhy,
- pokles rychlosti pod minimální hodnotu,
- opuštění referenčního úseku.

Způsob aktivace měření neovlivňuje diagnostiku výstražnou, ovlivněna je však diagnostika trendová, protože je více citlivá na porovnání dat za stejných podmínek.

Při návrhu managementu měření se musí zohlednit předpokládaný objem ukládaných a přenášených dat.

4.2 Přenos dat

Při sběru dat se musí dbát nejen na správný záznam signálů, ale také na správný záznam metadat. Společně se signálem ze snímačů zrychlení je nutné zaznamenat:

- identifikace sensoru (NODE ID),
- čas pořízení záznamu (nejlépe jednotně UTC+0 převzatý s GNNS, LTE nebo timeserveru),
- rychlost (počáteční / průměrná / průběh rychlosti),
- místo pořízení (dle GNSS, LTE, dle systému vozidla, průběžně),
- vzorkovací frekvence (pokud nelze zabezpečit přesnou, tak průměrná),
- scale faktor (pokud je uplatňován).

Způsob uložení signálových dat a metadat musí být proveden tak, aby se signálová data a metadata přenášela a archivovala neoddělitelně.

Pokud je to možné, doporučuje se synchronní způsob záznamu dat alespoň na úrovni jednoho podvozku. Není to podmínkou diagnostikování, ale přináší to další možnosti při výzkumu dat.

Identifikace sensorů je při vizualizaci dat rozklíčována podle ID NODE a identifikační tabulky **PREFIX_NODE** tak, že je každému sensoru přiřazena uživatelsky čitelná identifikace v úrovni: jednotka - vozidlo – podvozek – náprava – strana. Případně případně jednotka – vozidlo – náprava - agregát (motor, převodovka apod.).

> Systém značení sensorů musí musí být nastaven tak, aby identifikační číslo sensoru bylo unikátní v celém systému a nedošlo tak k záměně.

> Identifikační tabulku je potřeba vytvořit pro každý typ vozidla zvlášť. Při diagnostice vlotily vozidel s nejednotnou konfigurací se doporučuje tyto rozdělit do jednotlivých domén a diagnostiku provádět odděleně.

4.3 Archivace dat

Samotná data se ukládají v tabulce **PREFIX_DATA** a **PREFIX_META**. Struktura dat je popsán a v popisu databáze. Archivace dat může probíhat několika způsoby:

1) **manuální import** ze souborů naměřených při ad-hoc měření,
2) **automatický import** z lokálního adresáře (automatizovaný způsob dle bodu 1),
3) **online** přenosem přímo do databáze.

Předpokládá se, že primárně bude využívána varianta 3 ve verzi server. Pro desktop verzi se předpokládají varianty 1 a 2. Varianty však lze kombinovat různě, tedy manuální upload je možný i pro server.

> Způsob online přenosu dat z vozidla do databáze si musí zabezpečit provozovatel systému sám, protože se jedná o proces závislý na koncovém zařízení a samotném přenosu. S výhodou lze také využít některé z Cloudových služeb.

Příloha č.2

DEFINICE STRUKTURY DATABÁZE

1 Technické parametry databáze

Pro chod systému se využívá databáze MariaDB (https://mariadb.org/), která uchovává jak vstupní signálová data, metadata, identifikace sensorů atd., ale také mezivýsledky vypočtené ze vstupních signálů. Po provedení diagnostiky pak jsou uloženy do databáze také výstupy, které lze s výhodou rovnou vizualizovat.

1.1 Verze databáze

SW prostředky byly navrženy pro níže uvedenou verzi SW, což je minimální konfigurací pro spuštění a administrativu databáze. Nižší verze zřejmě mohou fungovat, ale nelze to zaručit.

Databázový server

Typ serveru: MariaDB

Verze serveru: 10.4.32-MariaDB - mariadb.org binary distribution

Verze protokolu: 10

Znaková sada: UTF-8 Unicode (utf8mb4)

Webový server

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12

Verze databázového klienta: libmysql - mysqlnd 8.2.12

Verze PHP: 8.2.12

phpMyAdmin

Informace o verzi: 5.2.1, nejnovější stabilní verze: 5.2.3

1.2 Nastavení přístupů

Pro přístup se doporučuje zřídit tyto typy účtů:

Admin – pro správu databáze, maximální práva přístupu.

Operator – uživatel oprávněný nastavovat provozní parametry a udržovat strojové učení.

Viewer – uživatele a programy umožňující čtení (koncový uživatel).

Automat – automatické skripty pro výpočet výsledků a mezivýsledků.

Pro přístup **admin** a **automat** nastavit pouze lokální přístup.

2 Struktura databází a tabulek

Databázová struktura je při instalaci vytvořená dodaným SQL skriptem, není tedy potřeba ji vytvářet ručně. Před instalací je potřeba změnit pouze hodnotu **PREFIX**, která určuje název diagnostické domény (název typu vozidla, název provozu, lokalita apod.).

Níže je vysvětlena funkce jednotlivých tabulek s tím, že se jedná o základní konfiguraci, kterou lze upravit nebo rozšířit. Při napojení na existující databázi lze přejmenovat některé názvy. Zachovány však musí být sloupce používané k indexaci. Do tabulek lze přidat další sloupce, které nesou další metadata, základní sloupce však nelze ubírat.

2.1 Tabulka PREFIX_DATA

Tabulka naměřených dat, přiřazení metadat je provedeno pomocí sloupce **IDMeasurement**. Jedná se o vstupní tabulku signálů. Propojení na tabulku META je realizováno skrze LEFT JOIN, tedy bude zahnut jen takový signál, pro který existují metadata v tabulce META. Sloupec RawData obsahuje binární data signálu zakódována do formátu BASE64. V případě formátu čísel Int16 nebo Int32 se uplatňuje po dekódování ScaleFactor. Pokud jsou zakódovány hodnoty float, pak se ScaleFactor neuplatňuje.

i	IDMeasurement	DataType	ScaleFactor	RawData
5182712	48493768	2	0.005903655	qwCSADMAYgCb/xUBZf/HAJEAIQD4AdcA4gEDAQ4AIP+y/rL+lf...
5182711	48493767	2	0.005902364	u/+A/hgB9P9+Apb/PAByAV3/mQCp/uf/IQBq/h0AfgA0AOn+fP...
5182710	48493766	2	0.005907761	xv5D/68AUwDU//AALwAyAeX/6ABnAdcAzAFtAOgBgwA6/5X/C/...
5182709	48493765	2	0.005908929	GgDB//sA+P9cAFUA0v5jAEH+jP+o/lsB1f9GANkBPgEFAUX/wf...
5182615	48493382	2	0.005902864	ewIUAgcB1P9q/s7+ev41/2P+KwCkA94AbAJUAXQA8/+D/QH/Mg...

2.2 Tabulka PREFIX_META

Tabulka obsahuje metadata, která se vážou k pořízení záznamu signálu v tabulce DATA. Jde například o čas měření, rychlost vozidla, GPS polohu, čas vložení záznamu, apod. Tuto tabulku lze rozšiřovat o další metadata, ale musí být zachovány mandatorní sloupce IDMeasurement a IDNode pro propojení dalších tabulek.

i	IDMeasurement	IDNode	MeasDate	speed	lat	lon	Datetime
1	1	220003	2025-07-21 06:29:14	4.5	49.3472	16.4178	2026-06-13 13:56:56
2	2	220003	2025-07-21 06:29:16	4.5	49.3472	16.4178	2026-06-13 13:56:56
3	3	220003	2025-07-21 06:29:19	9	49.3472	16.4178	2026-06-13 13:56:56
4	4	220003	2025-07-21 06:29:21	13.8	49.3472	16.4179	2026-06-13 13:56:56
5	5	220003	2025-07-21 06:29:24	23.3	49.3471	16.418	2026-06-13 13:56:56
6	6	220003	2025-07-21 06:29:26	31.3	49.3469	16.4182	2026-06-13 13:56:56

2.3 Tabulka PREFIX_NODE

Tabulka je propojená na tabulku META pomocí sloupcu **meta.IDNode** a **node.ID**. Tato tabulka opět umožňuje variabilitu zadání. Mandatorní je sloupec ID a sloupce: **train, wagon, wheel,**

sensor. Tyto informace slouží k popisu a zatřízení dat ve vlaku. Sloupec **parents** není povinný, ale lze jej použít při vizualizaci stromové struktury vlaku, protože definuje ID nadřazeného prvku.

i	ID	train	trains	bogie	wagon	wheel	sensor	parents
2	100	1	76	1	9001	Yb1	Ch9 Bogie high	82
3	101	1	76	1	9001	Zc1	Ch10 Body high	82
4	112	1	76	1	9001	Za1L	Ch1 Wheel high	82
5	113	1	76	1	9001	Za1P	Ch3 Wheel high	82
6	114	1	76	1	9001	Za2L	Ch5 Wheel high	82
7	115	1	76	1	9001	Za2P	Ch7 Wheel high	82
8	134	1	76	2	9001	Yb2	Ch9 Bogie high	116

2.4 Tabulka PREFIX_STAT

Jedná se o tabulku mezivýsledků, která se počítá automaticky na serveru a obsahuje statistické charakteristiky příslušného vzorku signálu. Propojení na signálová data zabezpečuje sloupec IDMeasurement.

i	IDMeasurement	min minimum	max maximum	mean str.h.	peak peak od str.h	rms rms bez str.h	std stand.dev.	swn skewness	kur kurtosis	cf crest factor	ipf impulse factor	ene energie
3079188	48493768	-6.09848	7.92271	0.00044681	7.92271	1.49428	1.49428	0.139507	4.38978	5.30201	6.86399	9145.9
3079187	48493767	-14.4903	11.4329	0.00406364	14.4903	1.83439	1.83439	-0.178392	8.01488	7.89925	11.0693	13783
3079186	48493766	-7.44378	7.27245	0.00519381	7.44378	1.65739	1.65739	-0.0162623	3.37575	4.49125	5.66948	11251.5
3079185	48493765	-16.9291	15.7591	0.00281886	16.9291	1.83051	1.83051	0.173134	9.08857	9.2483	12.373	13724.7
3079184	48493382	-9.86959	8.77166	0.00484075	9.86959	2.25086	2.25085	0.0144979	3.38414	4.38481	5.57055	20751.8
3079183	48493381	-9.55122	9.74626	0.00173589	9.74626	2.62915	2.62915	-0.0627983	3.51552	3.707	4.77108	28313.4
3079182	48493380	-9.69771	9.87489	0.00579068	9.87489	2.95437	2.95436	0.0000421286	2.51155	3.34247	4.04223	35751.1

2.5 Tabulka PREFIX_CPB

Jedná se o tabulku mezivýsledků vypočtených automaticky. Propojení je provedeno skrze IDMeasurement a sloupec CPB obsahuje textový CSV řetězec (čísla oddělená středníkem). Počet hodnot se může lišit s ohledem na typ CPB (1-okt, 1/3-okt, 1/12-okt), primárně je použito 1/3-okt CPB, tedy každý záznam obsahuje 10 hodnot pro centrální frekvence: 0 až 500 Hz

i	IDMeasurement	CPB
1	43042565	0.332575;0.431100;0.830748;1.710226;1.254434;0.693...
2	43042566	0.173862;0.186123;1.351843;1.055188;2.004900;0.405...
3	43042567	0.115625;0.539161;1.937501;0.231059;1.995281;0.938...
4	43042568	0.240056;0.435930;1.373848;1.171424;1.426285;0.546...
5	43042571	0.050948;0.260307;0.819141;1.225256;1.797330;1.566...

2.6 Tabulka PREFIX_FLAT_POSTFIX

Obsahuje výsledky diagnostiky detekce plochých kol. Podle použité metody je název tabulky upraven a lze vytvářet také různé označení pomocí POSTFIX. Pro jednoduchost se může jednat o indexování pomocí čísla. Tento přístup lze využít zejména při prvním výzkumném ohledání prvních nájezdů a nebo ve fázi trénování v případě strojového učení. Ve všech případech má ale tabulka stejnou strukturu povinných sloupců: **i, IDMeasurement, value, note**. Přitom vyplnění sloupce note není povinné a nevstupuje do žádného výpočtu. Jedná se o sloupec poznámek nebo jej lze použít pro uložení metadat.

i	IDMeasurement	value	note
1	44206681	0.25	*NULL*
2	44206691	0.75	*NULL*
3	44206721	0.5	*NULL*
4	44206731	0	*NULL*
5	44206750	1	*NULL*

U výsledku tohoto typu se předpokládá následující interpretace hodnot:

1.00 = Priorita A / 0.75 = Priorita B / 0.50 = Priorita C / 0.25 = Priorita E/ 0.00 = Priorita OK

Rozdělení hodnot 0.00 až 1.00 nahrazuje slovní nebo písmenné označení záměrně tak, aby bylo možné tyto hodnoty vizualizovat v dashboardech. Tento přístup umožňuje použití jak spojité, tak diskrétní způsoby interpretace výsledků.

2.7 Tabulka PREFIX_STAB

Výsledky výpočtu stability se ukládají do tabulky **PREFIX_STAB**, která je párovaána na původní data pomocí ID Measurement. Výsledkem výpočtu jsou hodnoty **ratio**. p**rob** a **pred**, které se ukládjí do stejně pojmenovaných sloupců. Slovní komentář výsledku diagnostiky se pak přiřazuje až při prezentaci výsledků. Tabulka obsahuje ještě sloupec META, kde jsou uloženy doplňující informace ve formátu JSON. Metadata k signálu jsou propojeny pomocí IDmeasurement do ostatních tabulek.

i	IDMeasurement	ratio	prob	pred
1	44206681	0.01245	0.01	0.125
2	44206691	0.07511	0.2	0.1254
3	44206721	0.9542	0.8044	15.24
4	44206731	0.9123	0.7587	2.548
5	44206750	0.00124	0.054	0.154

2.8 Tabulka PREFIX_COMF

Výsledky výpočtu stability se ukládají do tabulky **PREFIX_STAB**, která je párovaána na původní data pomocí ID Measurement. Výsledkem výpočtu jsou hodnoty **ratio**. **prob** a **pred**, které se ukládjí do stejně pojmenovaných sloupců. Slovní komentář výsledku diagnostiky se pak přiřazuje až při prezentaci výsledků. Tabulka obsahuje ještě sloupec META, kde mohou být uloženy doplňující informace ve formátu JSON. Metadata k signálu jsou propojeny pomocí IDmeasurement do ostatních tabulek.

i	IDMeasurement	comfZ	comfY	comfYZ	META
1	3564	0.12	0.23	0.16	*NULL*
2	3565	0.14	0.22	0.17	*NULL*
3	3566	0.11	0.25	0.19	*NULL*
4	3567	0.11	0.25	0.18	*NULL*

2.9 Tabulka PREFIX_MARK_XXXX

Tabulka PREFIX_MARK_XXXX sloužípro uchování kategorizace signálů uživatelem. Primárně se jedná o tabulku použitelnou pro detekci ploch, ale může šloužit I pro jiné poruchy a jiné signály. Základním principem je přiřadit každému IDmeasurement příslušnu značku MARK (zde hodnoty 0 až 1), které vyjadřuje závažnost poruchy. Mimo to obsahuje tabulka index a poznámku. Při uložení se zaznamená také datum uložení. Hodnoty MARK se použijí pro trénování strojového učení. Lze je ale použít I jako zpětnou vazbu uživatelů údržby.

Název tabulky má také postfix XXXX, který se čísluje od 0000 do 9999. Data lze do tabulky plnit ručně, externím programem, importem z CSV nebo pomocí programu ROBD_client, kde lze marky přidávat přímo při prohlížení záznamů.

i ▲ 1	IDMeasurement	mark	note	insertdate
1	44008046	1		2026-06-04 14:01:11
2	44009118	1		2026-06-04 14:01:31
3	44009763	1		2026-06-04 14:01:36
4	44011465	1		2026-06-04 14:01:37
5	44012145	1		2026-06-04 14:01:39
6	44012820	1		2026-06-04 14:01:40
7	44013573	1		2026-06-04 14:01:41
8	44014315	0.25		2026-06-04 14:01:43
9	44015002	0.25		2026-06-04 14:01:47
10	44015493	0.5		2026-06-04 14:01:48
11	44016023	0.25		2026-06-04 14:01:50
12	44017173	0.5		2026-06-04 14:01:52