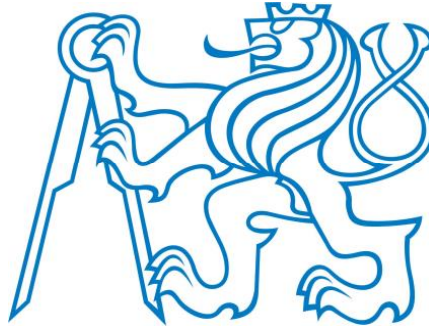


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta strojní – Ústav automobilů, spalovacích motorů a kolejových vozidel




BAKALÁŘSKÁ PRÁCE

**IDENTIFIKACE JÍZDNÍCH STAVŮ
OVLIVŇUJÍCÍCH OPOTŘEBENÍ
SDÍLENÉHO VOZIDLA**

IDENTIFICATION OF SHARED VEHICLE WEAR RELATED DRIVING STATES

Prohlašuji, že jsem tuto práci vypracoval samostatně s použitím literárních zdrojů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů.

Datum: **13.7.2022**


.....
Podpis

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Miletín** Jméno: **Michal** Osobní číslo: **492408**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav automobilů, spalovacích motorů a kolejových vozidel**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Identifikace jízdních stavů ovlivňujících opotřebení sdíleného vozidla

Název bakalářské práce anglicky:

Identification of shared vehicle wear related driving states

Pokyny pro vypracování:

Definujte jízdní stavy vozidla, proveďte jejich parametrický popis a simulujte je prostřednictvím vhodného výpočetního SW. Určete, jaká negativa může který stav obnášet z hlediska opotřebení nebo poškození vozidla. Rozšiřte simulaci o vybrané negativní vlivy (např. přejezd velké nerovnosti, nebo špatné ovládání spojkového pedálu), identifikujte veličiny, které jsou stavem nejvíce ovlivněny a navrhnete principy jejich zpracování pro využití v HW jednotce vozidla projektu sdílených vozidel Uniqway. Při řešení spolupracujte se studentským týmem, který projekt realizuje. Určete hraniční situace pro detekci vybraných negativních vlivů.

Seznam doporučené literatury:

Jméno a pracoviště vedoucí(ho) bakalářské práce:

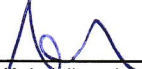
Ing. Václav Jirovský, Ph.D. 16123


Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

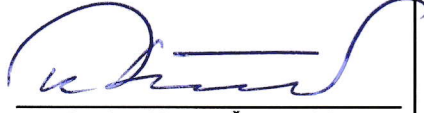
Datum zadání bakalářské práce: **22.04.2022**

Termín odevzdání bakalářské práce: **13.07.2022**

Platnost zadání bakalářské práce: _____


Ing. Václav Jirovský, Ph.D.
podpis vedoucí(ho) práce


doc. Ing. Oldřich Vítek, Ph.D.
podpis vedoucí(ho) ústavu/katedry


doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.


Datum převzetí zadání


Podpis studenta

Anotace

Bakalářská práce se věnuje problematice sledování opotřebenosti vozidla carsharingu a to konkrétně z hlediska detekce událostí, které vozidlo poškozují. Práce popisuje princip a funkčnost softwaru, navrženého pro analýzu dat získaných ze simulačního prostředí. Software se detailně věnuje dvěma případům, při kterých se vozidlo opotřebovává, a to detekcí průjezdu skrze výmol nebo příčný práh a měření zacházení se spojkou u vozidla s manuální převodovkou. Při volbě sledovaných parametrů je kladen důraz na hardwarové možnosti vozidel služby Uniqway.

Klíčová slova: opotřebenost vozidla, simulace jízdy, vyhodnocení dat, spojka, výmol

Abstract

The bachelor's thesis is devoted to the issue of monitoring the wear and tear of a carsharing vehicle, specifically from the point of view of detecting events that damage the vehicle. The work describes the principle and functionality of the software designed for data evaluation based on data obtained from the simulation environment. Finally, the software deals in detail with two cases in which the vehicle wears out. Firstly, detection of passing through a pothole or a speed bump, and secondly measurement of clutch handling in a vehicle with a manual transmission. When choosing the monitored parameters, emphasis is placed on the hardware capabilities of Uniqway vehicles.

Keywords: vehicle wear and tear, drive simulation, data evaluation, clutch, pothole

Obsah

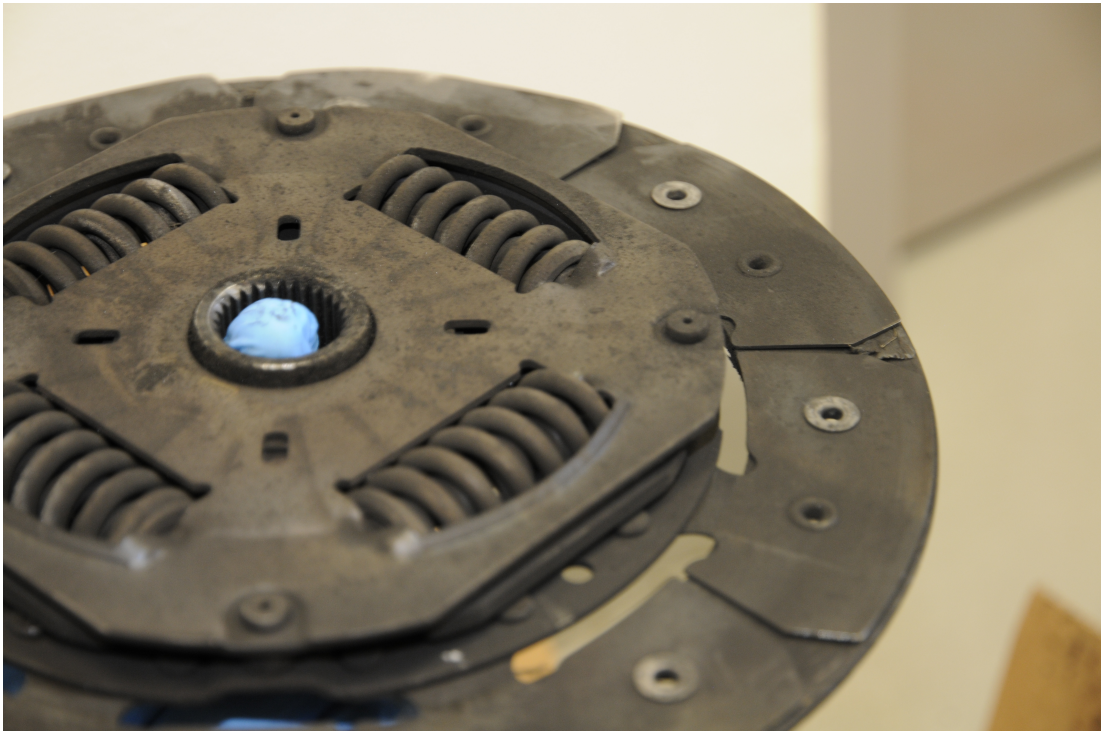
1	Úvod	7
2	Carsharing	8
2.1	Historie	8
2.2	Uniqway	8
2.2.1	Motivace pro sledování opotřeбенí vozidel	8
3	Rešerše podobných projektů	9
4	Jízdní extrém	10
4.1	Definice pojmu	10
5	Simulace v softwaru IPG CarMaker	10
5.1	Důležitost simulace	10
5.2	Popis CarMakeru	11
5.3	Simulační prostředí	11
5.3.1	Hlavní GUI	11
	Horní pás	12
	Prostřední pás	12
	Spodní pás	12
5.3.2	Režimy simulace	13
5.3.3	Manévr	13
5.3.4	Řidič	13
	Varianty řidiče	14
5.3.5	Scénář/Silnice	15
	Popis editoru	15
	Vrchní pás ikon	16
	Levá postranní lišta ikon	16
	Pravá postranní karta - Parameters/Object list	16
	Grafický náhled	17
	Spodní okno - Height profile/Messages	17
5.3.6	Vozidlo	17
	Generátor vozidla	17
5.3.7	Pneumatiky	18
5.3.8	IPG Movie	19
5.3.9	IPG Control	20
	Volba měřených veličin a vzorkovací frekvence	21
	Export dat	21
5.4	Řešení konkrétní situace v simulaci	21
5.4.1	Zvolený testovací úsek	21
5.4.2	Zvolené vozidlo a pneumatiky	23

5.4.3 Zvolený řidič	23
5.4.4 Sledované veličiny	23
5.4.5 Export dat pro Matlab	24
6 Matlab	24
6.1 Důvod použití	24
6.2 Hierarchie programu	25
6.3 Import dat ze simulace	25
7 Sledování průjezdů výmoly	26
7.1 Vysvětlení programu v softwaru Matlab	27
7.2 Výstup ze softwaru	31
8 Sledování zacházení se spojkou	34
8.1 Vysvětlení programu v softwaru Matlab	35
8.2 Výstup ze softwaru	38
9 Závěr	40
Seznam použitých značek a symbolů	42
Seznam použité literatury a zdrojů	43
Seznam použitého SW	45
Seznam příloh	46

1 Úvod

Carsharing se v dnešní době dostává do širokého povědomí, a kromě nespočtu výhod s sebou přináší i nové problémy, které musíme v automobilovém průmyslu řešit. Jedním z nich je značný nárůst opotřebení vozidel. V této bakalářské práci se budeme zabývat možnostmi identifikace situací, ve kterých k opotřebení dochází, takzvanými jízdními extrémami. Motivací pro zabývání se touto tematikou je snaha o získání nástroje, který by pomohl získat lepší porozumění z jakého důvodu k opotřebení dochází.

Abychom mohli jednoduše projít mnoho nápadů a dosáhnout tak porozumění situací, použijeme k identifikaci simulačního prostředí. Díky možnostem simulace jsme schopni flexibilně zkoušet veliké množství nových testů z pohodlí kanceláře bez nákladného testování v reálných vozech a bez potřeby použití nákladné měřicí techniky. Po provedení simulace se můžeme rozhodnout z velikého množství senzorů v simulovaném voze, jak budeme s daty nakládat a jaké senzory budeme při detekci v reálném autě pro vyhodnocení jízdních extrémů potřebovat. V poslední řadě se zaměříme software, který se bude specializovat na vyhodnocení dat, které ze simulace získáme. Tento vyhodnocovací nástroj bude navržen tak, aby ho bylo možné následně implementovat do reálného vozu a sledovat charakteristiku uživatelů carsharingové služby Uniqway.



Obr. 1: Opotřebená spojková lamela vozu Uniqway

2 Carsharing

Carsharing je moderní služba, která umožňuje lidem používat automobil, aniž by jej museli vlastnit a starat se o něj, zároveň díky moderním komunikačním technologiím umožňuje většímu množství lidí flexibilně a efektivně využívat menší množství automobilů. [3]

2.1 Historie

První zmínky o carsharingu se objevují již na počátku 50. let 20. století, a to konkrétně o společnosti zvané Selbstfahrgemeinschaft z švýcarského Curychu. Důvodem byla hlavně ekonomická situace, protože většina obyvatel si zkrátka automobil nemohlo dovolit. Prvním carsharingem v jakési moderní podobě se stala v roce 1987 společnost pod dnešním názvem Mobility Car sharing Switzerland. [5]

2.2 Uniqway

Uniqway je první a dosud jediný studentský carsharing v České republice. Je vyvíjen a zároveň používán studenty zejména v Praze, ale v poslední době se rozšiřuje i do dalších měst a od roku 2019 je možné s auty vyjet i do zahraničí. Na projektu se podílí tři univerzity a to Vysoká škola ekonomická v Praze, Česká zemědělská univerzita v Praze a České vysoké učení technické v Praze ve spolupráci s ŠKODA AUTO DigiLab s.r.o.. Pro studenty Uniqway představuje kromě způsobu přepravy i příležitost získat zkušenosti a rozvíjet své znalosti a dovednosti v oblasti dopravy například prostřednictvím workshopů nebo akcí a soutěží. [4]

2.2.1 Motivace pro sledování opotřebení vozidel

Není žádným překvapením, že vozidla v oblasti carsharingu jsou opotřebovávána mnohem rychleji a často i jinak než vozidla v osobním vlastnictví. U automobilů v carsharingu Uniqway často dochází k vysokému opotřebení dílů, se kterými u aut v osobním vlastnictví nebývá s opotřebením problém, jako například spojka převodovka, tlumiče, a jiné. Proto se nabízí otázka zjistit, jaké jsou bližší charakteristiky opotřebení. Jedna z otázek, kterou si pokládáme je, zda největší poškození na vozidlech způsobují spíše jedinci nebo jestli je na vině spíše celková myšlenka carsharingu a fakt, že lidé nejsou na vozidla zvyklí a opotřebovávají je spíše rovnoměrně všichni více, než by se tomu tak dělo za normálního provozu v rukou jednotlivce.

Proto je důležité nastavit určité podmínky a faktory podle kterých lze odlišit řidiče, kteří se chovají k vozidlu šetrně a řidiče kteří jezdí bezohledně. Díky těmto podmínkám se následně můžeme pokusit na základě nich rozřadit řidiče do kategorií, které by nám umožnili sledovat opotřebení vozidla po určitých typech jízdy.

3 Rešerše podobných projektů

V rámci rešerše za účelem prozkoumání výzkumných projektů, které se zabývaly nebo zabývají podobnými projekty jsem dospěl k závěru, že je tato práce poměrně unikátní z hlediska pojetí problematiky využití dat a neobvyklých veličin v kontextu vyhodnocení situací, které opotřebovávají vozidlo. Nepodařilo se mi ani dohledat nějaký výzkum klasifikace řidičů z hlediska opotřebení vozidla.

Existují práce, které se zabývají opotřebením vozidel v carsharingu. Na základě kterých jsem se rozhodl při výběru komponentů, u kterých jsem se rozhodl sledovat opotřebení.

V carsharingu Uniqway mezi nejběžnější opotřebení ve stanoveném horizontu patří opotřebení opotřebení brzd (brzdových destiček, kotoučů a mechanismu bubnů), opotřebení spojky (lamela, spojkové ložisko a přitlačný talíř) a opotřebení pneumatik. Práce zároveň shledává, že k zrychlenému opotřebením dochází při vysokých teplotách, k čemuž u spojky a brzd dochází při nevhodném používání zejména z hlediska tření, které trvá nepřiměřeně dlouho.[9]

Dále existují studie, které se zabývají predikcí opotřebení dílů automobilů. Jedna z prací nastiňuje sledování zacházení se spojkou pomocí senzoru polohy spojkového pedálu, otáček motoru a rychlosti vozidla. Ke konkrétnímu způsobu řešení se, ale práce dále neubírá a pouze konstatuje, že na základě těchto parametrů by bylo vyhodnocení velmi složité.[9]

V jiné studii se autor zabývá použitím Bayesových algoritmů pro získání prediktivního modelu opotřebení na základě dat ze sledování vibrací systému. Tato varianta potřebuje zvláštní senzory a zároveň musí být zkalibrována na každý vůz zvláště, což by pro carsharing Uniqway nebylo úplně ideální, protože služba využívá různé modely aut a v rámci jedné modelové řady různé motorizace. Z tohoto důvodu by vzniklo pravděpodobně více než 5 kombinací, na které by musel být systém konfigurován. Zároveň tato varianta pouze predikuje selhání jednotlivých komponentů, ale nesleduje průběh opotřebení na základě zacházení jednotlivých řidičů.[11]

Další z oblastí, kterou se zabývám, je sledování průjezdu výmoly. V této oblasti sleduji veliký trend v předvídání průjezdu a snahu o vyhodnocování ještě před událostí, aby mohlo vozidlo reagovat automaticky a upravit například nastavení tlumičů, aby byl průjezd hladší nebo rozsvítit kontrolku ve vozidle, aby byl řidič schopen se včas výmolu vyhnout sám. Tato vyhodnocení probíhají na základě strojového učení a snímání silnic kamerami.[10]

Jako poslední z podobných projektů, a dokonce z oblasti carsharingu jsem narazil na detekci nárazu vozidla koly na obrubník. Tato situace se sice liší od průjezdu výmolem, ale

využívá údaje z akcelerometru pro vyhodnocení situace. Způsob, kterým data vyhodnocuje, ale spočívá pouze ve vyhodnocení velikých maximálních výchylek zrychlení, ke kterým dochází při nárazu.[8]

4 Jízdní extrém

4.1 Definice pojmu

Jízdním extrémem se v této bakalářské práci se rozumí: nakládání s vozidlem způsobem, který není nutný pro dostání se ze začátku na konec cesty z hlediska převážně opotřebení, poničení nebo dokonce celkového zničení vozidla. Jízdní extrém je tedy objektivně pojmenovatelná charakteristika jízdy, která převážně negativně ovlivňuje stav sdíleného vozidla. Při zkoumání jednotlivých jízdních extrémů se můžeme zaměřit na jejich vliv při degradaci jednotlivých komponentů automobilu. Jízdní extrémy je tedy možné rozdělit, na jízdu, která způsobuje opotřebení a na jízdu nebezpečnou, u které je zvýšené riziko poničení nebo dokonce zničení automobilu. Z hlediska opotřebení můžeme ještě dělit do dvou hlavních kategorií a to na opotřebení dílů, u kterých se opotřebení předpokládá a jsou na voze zpravidla připravené k výměně - tak zvané díly spotřební a na opotřebení dílů, u kterých výrobce předpokládá životnost shodnou s životností samotného vozidla - tak zvané díly stálé. U stálých dílů je výrobcem obvykle kladen menší důraz na jednoduchost výměny, cenu servisu, a dokonce se setkáváme i s díly které jsou zcela neopravitelné nebo nevyměnitelné, ať už z ekonomického či fyzického hlediska.

Příklady jízdních extrémů z hlediska opotřebení spotřebních dílů: Opotřebení brzdového ústrojí z důvodu nadměrného zahřívání nebo pneumatik z důvodu agresivní jízdy.

Příklady jízdních extrémů z hlediska opotřebení stálých dílů: Opotřebení podvozku kvůli průjezdu nerovnostmi, motoru a jeho součástí například jízdou v extrémních otáčkách, ať už nízkých či vysokých nebo například interiéru.

5 Simulace v softwaru IPG CarMaker

5.1 Důležitost simulace

Simulace je pro identifikaci parametrů, kterými je možné rozpoznat jízdní extrémy klíčová. Počet parametrů, které můžeme v simulaci sledovat je obrovský a jejich sledování je prakticky bezplatné. Vyhodnocení, jestli měřená hodnota reaguje na extrémy provádíme v prostředí, kde si jízdní extrém izolujeme od ostatních problémů, které sledovat nechceme. Tato výhoda, může být samozřejmě ji nevýhodou, když měříme v izolovaném prostředí, tak

je jednoduché přehlédnout vlivy, které do senzorů, které si vybereme budou za normálního provozu, přinášet šum. Proto je vhodné simulovat i v prostředí, které simuluje reálné trasy, po kterých se automobil běžně pohybuje.

Pokud bychom chtěli při návrhu senzorů pro identifikaci jízdních extrémů simulaci vynechat, tak bychom museli vědět, jaké veličiny chceme měřit a jaké senzory k tomu použijeme. Tento přístup bez simulace musí být často doprovázen výrobou prototypů, které mohou být velmi nákladné a často je potřeba je měnit, čímž se náklady dále zvyšují. V případě, že nevíme, jaké senzory a jak je chceme použít je bez simulace velmi obtížné na ně bez jakýchkoli zkušeností přijít. Naopak v simulaci můžeme v rozumné míře sledovat veličin a čidel prakticky neomezené množství a nakonec se teprve rozhodnout.

5.2 Popis CarMakeru

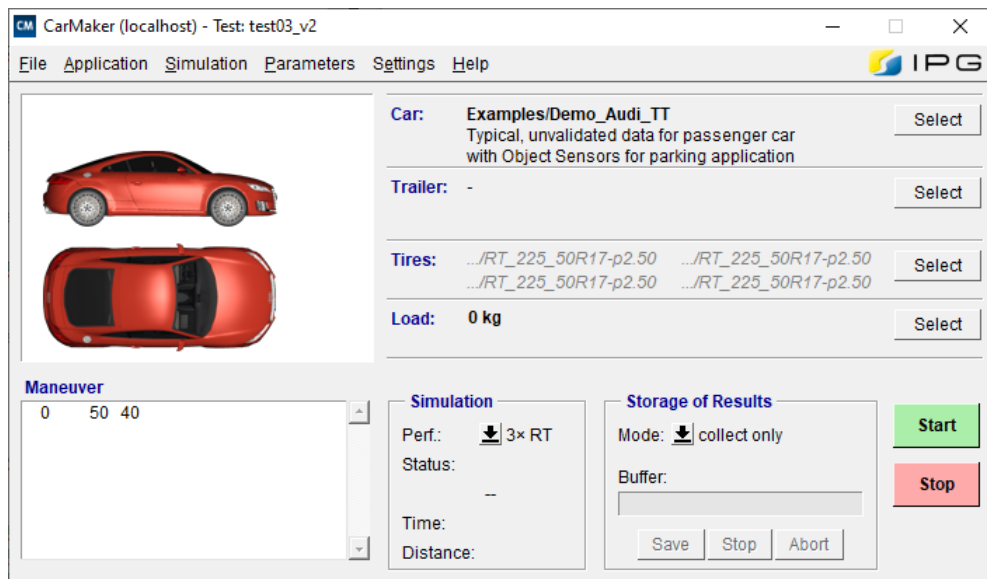
CarMaker je simulační software německé společnosti IPG Automotive GmbH, který se v praxi automobilového průmyslu používá zejména pro vývoj autonomních systémů automobilů jako jsou například systémy ABS, ESP a další. Dále se také používá pro simulaci automobilů v obvyklých testech, jako je například test zrychlení a brzdění na různých površích nebo testy simulující reálné zkoušky, které se v automobilovém průmyslu už dlouho používají (například takzvaný Losí test). Software umožňuje zadefinovat trasu, zvolený manévr, vozidlo pomocí parametrického modelu a jeho pneumatiky, vlastnosti řidiče, a dokonce i provoz ostatních vozidel.

5.3 Simulační prostředí

Simulační prostředí softwaru IPG CarMakeru 10.0 se skládá z několika oken s různými nástroji a jedním oknem hlavním, které celou simulaci spravuje. Všechna společně vytváří celek, kterým je IPG CarMaker.

5.3.1 Hlavní GUI

Celé ovládání a simulace virtuálních vozidel v prostředí CarMaker se provádí přes hlavní GUI CarMaker. Okno působí jednoduchým dojmem, ale nabízí opravdu nespočet možností, jak simulaci konfigurovat.



Obr. 2: Hlavní GUI IPG CarMaker 10.0

Horní pás Většina podrobných nastavení simulace se nachází v horním pásu. Je z něj mimo jiné možnost otevřít všechny nástroje CarMakeru jako je například IPG Movie, IPG Control, Scenario/road editor, editor vozidla a mnoho dalších. Pro detailní porozumění CarMakeru je zde také dostupný velmi podrobný návod v záložce Help.

Prostřední pás V prostředním páse dochází k volbě vozidla a jeho parametrů a příslušenství pro daný test. Mezi možnostmi, které volíme patří: Vozidlo, příves, pneumatiky s ráfky, a nakonec můžeme do soupravy přidat nějaké břemeno do libovolné pozice.

Spodní pás Ve spodní části jsou tři podskupiny. Maneuver, Simulation a Storage of Results. Samozřejmě nechybí ani zelené tlačítko Start, které celou simulaci spouští a červené tlačítko stop, které simulaci zastavuje.

Manuever Sekce Maneuver nabízí náhled na pokyny, které jsou naplánované pro vykonání simulačním řidičem. Dvojklikem do oblasti jsme automaticky přesměrováni do editoru manévru.

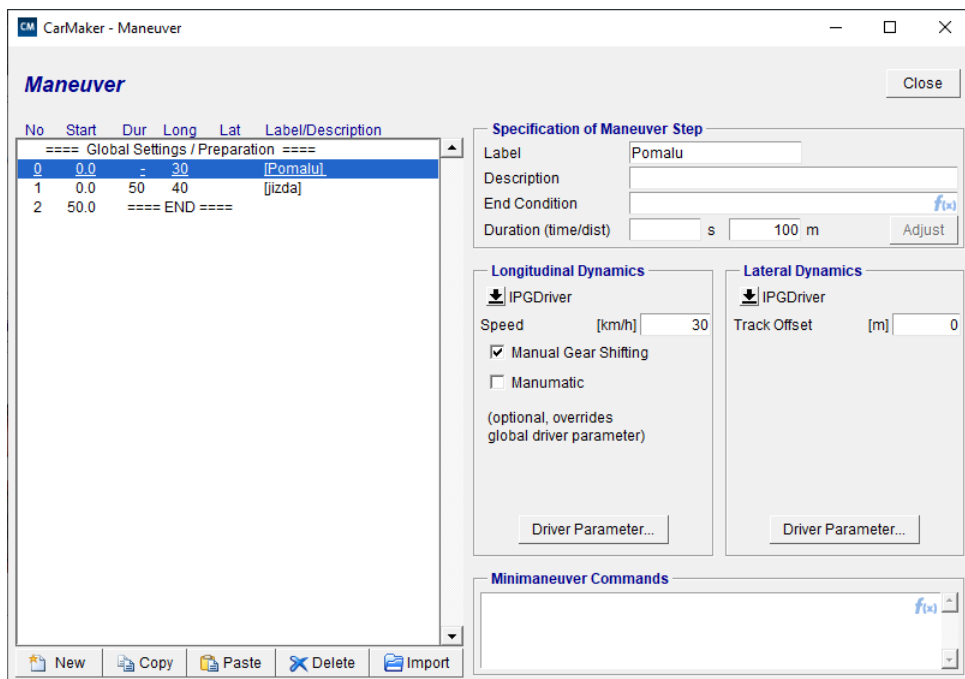
Simulation Prostřední sekce Simulation umožňuje volbu, jak rychle bude simulace probíhat. Simulaci je možno spustit v reálném čase, zrychleně anebo co nejrychleji, což je režim, kdy simulace dojde do konce prakticky okamžitě. Samozřejmě vše se může lišit v závislosti na výkonu počítače a složitost testu. Sekce Simulace dále slouží pro monitorování průběhu aktuálně probíhající simulace. Můžeme se zde dozvědět stav simulace, jak dlouho probíhá a jak daleko zkušební vozidlo došlo.

Storage of results Sekce Storage of results, umožňuje vytvářet jakýsi záznam o simulaci. Toto může být velmi užitečné pro ukládání výsledků při automatizovaném testu. Pokud bych například posílal různá auta různou rychlostí losím testem, tak by mi software mohl automaticky ukládat na disk záznam o tom, jestli vozidlo test zvládlo a s jak velkým úspěchem.

5.3.2 Režimy simulace

5.3.3 Manévr

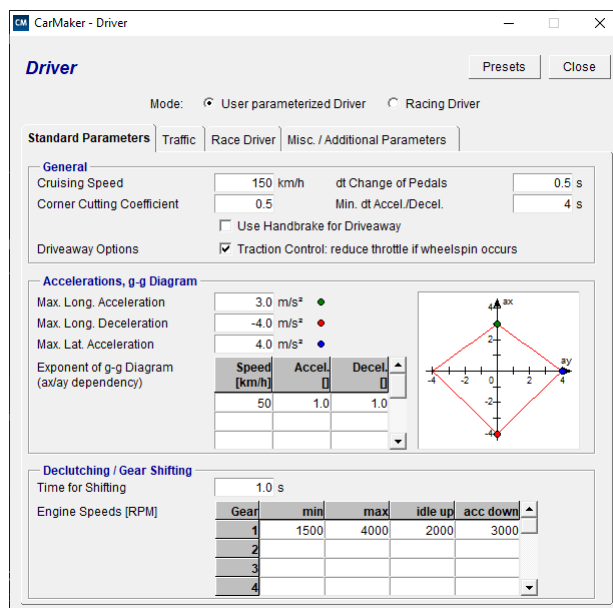
Jako manévr se rozumí soubor pokynů, které dáváme testovacímu řidiči. Například jed' 100 metrů rychlostí 30 kilometrů za hodinu a poté zrychli na 40 kilometrů za hodinu a jed' 50 sekund jako na obr. 3.



Obr. 3: Konfigurace manévru

5.3.4 Řidič

IPGDriver je nástroj, který zajišťuje následování vytyčené trasy (otáčení volantu) a ovládání pohonného ústrojí (pedály, řazení, ruční brzda) na dané trati.



Obr. 4: Konfigurace řidiče

IPGDriver vám umožňuje regulovat, jak lidský řidič ovládá vozidlo a jeho funkce. Tyto činnosti zahrnují řízení, brzdění, polohu plynu, řazení a chod spojky. Avšak je možné používat IPGDriver pouze k ovládání kurzu, a ne rychlosti nebo naopak. Kompletní seznam shrnující vlastnosti ovládané řidičem:

- volba jízdní dráhy v rámci hranic jízdního pruhu (řezání zatáček)
- řízení
- volba rychlosti jízdy podle kurzu a chování vozidla
- poloha pedál plynu a brzdy a také spojkového

pedálu ve spojení s výběrem převodového poměru na základě rychlosti vozidla

- reakce na cizí objekty a další účastníky provozu

Varianty řidiče CarMaker obsahuje dvě verze IPGDriver: uživatelsky parametrizovatelného řidiče a závodního pilota. U uživatelsky parametrizovaného řidiče můžeme provést jeho přizpůsobení tak, aby znal definované limity vozidla na základě předem nastavené sekvence manévrů a na základě nich se přizpůsobil danému manévru. Druhou verzí je takzvaný závodní pilot. Navíc oproti uživatelsky parametrizovanému

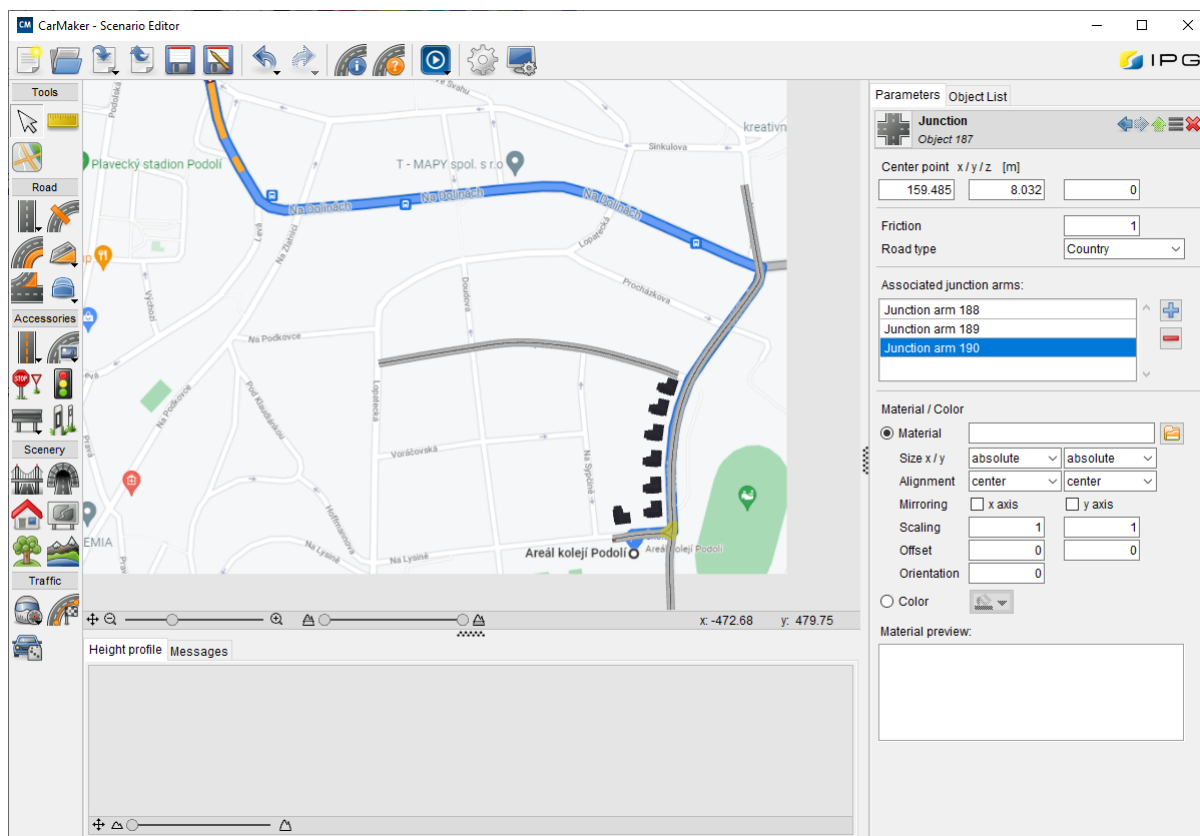
jezdci, umožňuje optimalizovat čas na kolo automaticky, avšak pouze za předpokladu, že jako limit průjezdu považujeme schopnosti vozidla a nikoli dovednost řidiče. Simulace posléze automaticky provede optimalizaci řidiče a jeho parametrů pro dosažení optimálního průjezdu.

5.3.5 Scénář/Silnice

V CarMakeru je Scenario Editor grafické uživatelské rozhraní (GUI), které umožňuje vytváření vysoce komplexních silničních sítí pro simulaci a řízení vozidel. Je přístupný přes CarMaker GUI kliknutím na nabídku Parametry a výběrem Scenario/Road. Editor scénářů je front endem stavby silnic (uživatelským prostředím se kterým pracujeme), zatímco IPGRoad je back endový modul neboli softwarová knihovna používaná k budování silnic a scénérií v CarMakeru na základě uživatelsky vytvořeného scénáře.

Editor scénářů navíc není určen pouze k navrhování a budování silničních sítí, ale také k usnadnění definování tras pro testování vozidla v provozu. Dále jej lze použít pro snadné definování velkého množství dopravních prostředků díky zabudovanému generátoru provozu.[1]

Popis editoru Všechny trasy vytvořené pomocí Editoru scénářů se ukládají buď jako samostatné soubory ve formátu CarMaker InfoFile s příponou .rd5 nebo jako část celého programu. Editor umožňuje import dat trasy z cizích prostředí jako je například Google Earth nebo TomTom maps.



Obr. 5: Editor cesty

Vrchní pás ikon Vrchní pás ikon (dle Obr.5) umožňuje načíst soubory silnic, uložit je a upravit obecná nastavení scénáře.[1]

Levá postranní lišta ikon Levá postranní lišta (dle Obr.5) poskytuje přístup ke všem nástrojům potřebným pro budování silniční sítě a přidávání/úpravu specifických atributů silnice.[1]

Pravá postranní karta - Parameters/Object list Pravý postranní panel (dle Obr.5) pokrývá dvě funkce: Jako první je karta Parameters určená pro konfiguraci podrobností vybraného silničního prvku. V horní části jsou speciální tlačítka umožňující uživateli přepínat mezi předchozím a následujícím objektem, přejít na nadřazený prvek, konfigurovat podrobnosti, jsou-li k dispozici, nebo odstranit prvek. Níže pak samotný prvek konfiguruje. Jednotlivé možnosti jsou různé pro jednotlivé typy prvků.

Druhá záložka v poli dle obr.5 nazvaná Object list obsahuje seznam všech prvků současného scénáře v hierarchickém pořadí podle času přidání. Nadřazené a podřízené objekty jsou ve stromovém zobrazení. Výběr položky ve stromové struktuře vede k výběru příslušného objektu. Za názvem prvku následuje v závorkách jeho vnitřní ID. Stejně ID je také zobrazeno v záhlaví karty Parametry objektu. ID lze také použít v poli Hledat.

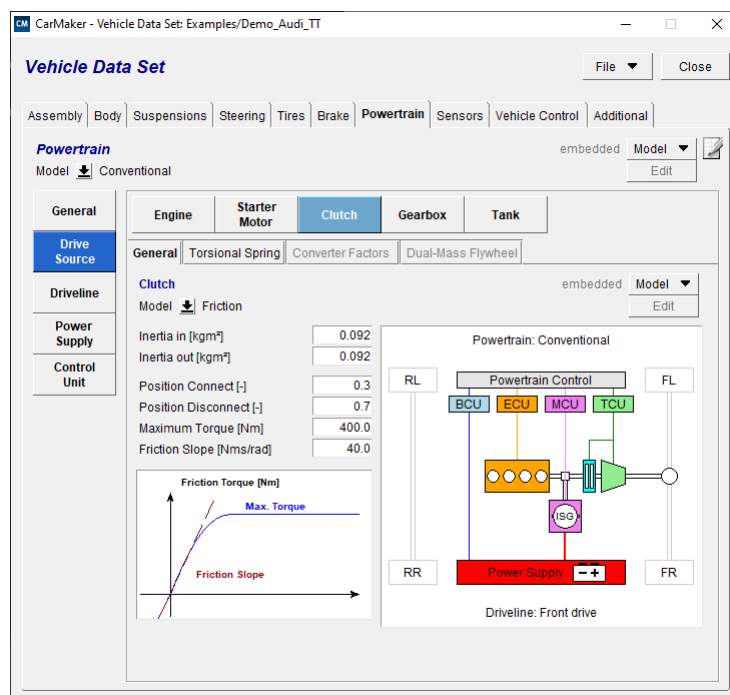
Kontextová nabídka přístupná kliknutím pravým tlačítkem myši na stromové zobrazení poskytuje možnost sbalení/rozbalení všech prvků stromu nebo podstromu [1]

Grafický náhled Prostřední okno (Obr. 5) zobrazuje 2D pohled na půdorys vytvořené silniční sítě. V nastavení je k dispozici několik možností přizpůsobení zobrazení.[1]

Spodní okno - Height profile/Messages Nejspodnější část okna (Obr. 5) obsahuje dvě záložky: Height profile nabízí boční pohled podél výškového profilu daného segmentu. V části Messages se zaznamenávají varování a chyby v případě chybných nebo nevhodných parametrů spolu s obecnými informacemi. Zprávy je možné třídit použitím filtru.[1]

5.3.6 Vozidlo

Volba vozidla je nezbytnou součástí testovacího scénáře. V Softwaru CarMaker je dostupné velké množství předvytvořených vozidel, ze kterých je možnost volit a následně upravit jejich některé parametry. Následně existuje možnost vozidlo generovat anebo ho zadefinovat rovnou kompletně všemi parametry, které CarMaker nabízí.



Obr. 6: Menu konfigurace vozidla

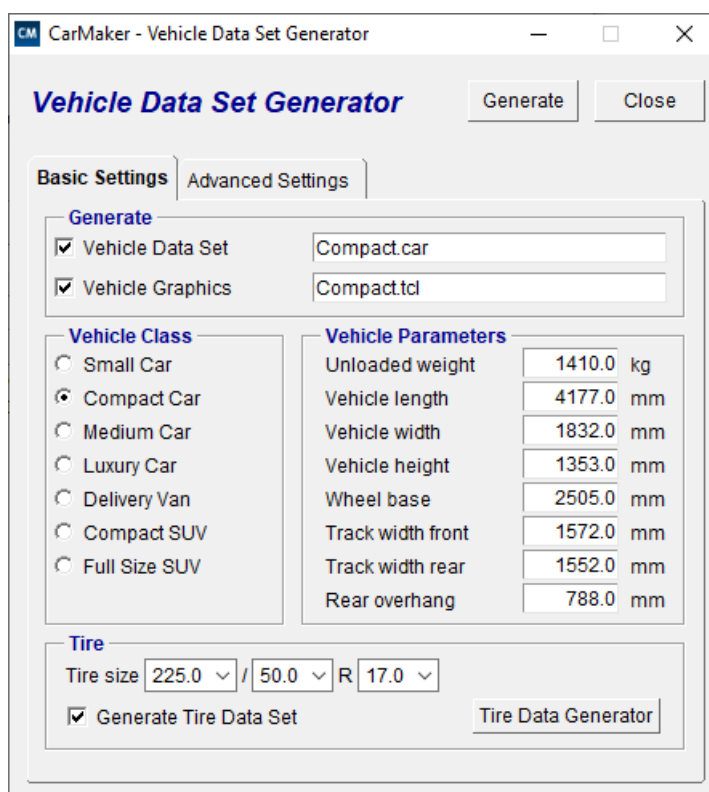
Generátor vozidla Generátor datové sady vozidla umožňuje uživateli vytvořit celou řadu dat o vozidle na základě relativně malého množství základních informací o vozidle. To je užitečné z několika důvodů:

- V případě, že uživatel nemá přístup k podrobným informacím o vozidle, které jsou nutné pro simulaci.

- Pokud je nutné pouze průměrné vozidlo určité třídy za účelem testu různých modulů, které mohou být shodné pro více modelů, jako jsou asistenční systémy řidiče.

- Jako výchozí bod pro parametrizaci datové sady vozidla, na kterou posléze navážeme změnou pouze některých údajů, které nás zajímají a není pak nutné přesně definovat ostatní méně důležité parametry vozidla.

Navzdory skutečnosti, že generátor sady dat o vozidle vytváří kompletní a plně funkční sadu dat, nevytváří ověřenou virtuální parametrizaci vozidla a to znamená, že se mohou výsledky simulace značně lišit od skutečnosti. Proto je nutné data získaná ze simulace validovat.

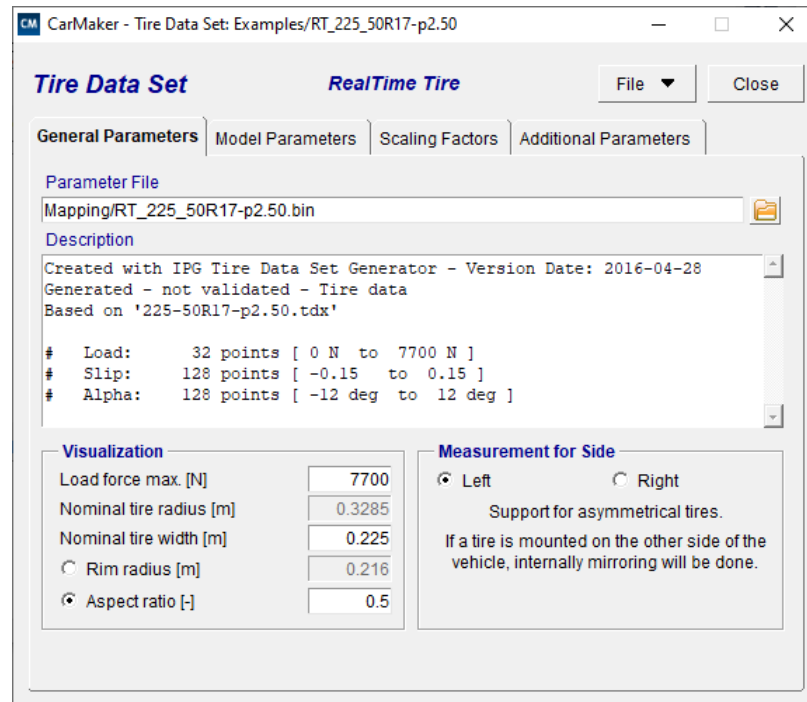


Obr. 7: Generátor vozidla

5.3.7 Pneumatiky

Volba pneumatik je věc, která není součástí volby vozidla. Volba pneumatik je tedy nezávislá na volbě vozidla a my jsme schopni zvolit různé obutí pro stejné vozidlo pouhou změnou konfigurace. Tato volba by nám mohla například přizpůsobit simulaci pro jízdu s

vozidlem na zimních pneumatikách, které se z pravidla liší u osobních automobilů svými rozměry, a to zejména užší šířkou a vyšším profilem s použitím menších ráfků.



Obr. 8: Nastavení pneumatik

Na obr.8 je zobrazen editor pneumatik. Kromě rozměrů můžeme definovat i jejich chování ať už v kinematice, jejich tuhost, vlnění v rychlosti nebo jejich únosnost a další.

5.3.8 IPG Movie

Pomocí utility IPGMovie se vytváří 3D pohled na silniční síť. Z IPGMovie nevychází, žádné zkoumání ani žádná data, ale je to skvělý nástroj například pro rychlou vizuální identifikaci, zda simulace probíhá podle představ a může být užitečným nástrojem pro rychlé vyřazení datové sady. V IPGMovie je například jednoduché poznat, když jde auto do hodin nebo když projíždí trasu špatným směrem. Nástroj zároveň poskytuje pohled na základní údaje o vozidle jako je rychlost, otáčky motoru a zařazený rychlostní stupeň. Vizualizaci je možné konfigurovat například změnou grafického stylu zobrazení nebo změnou zobrazovaných dat.

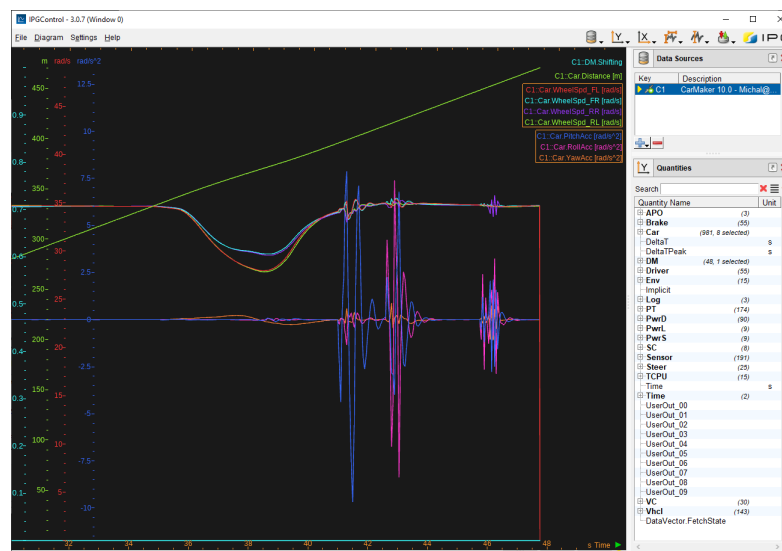


Obr. 9: IPGMovie

Na Obr. 9 je navíc k budíkům navíc vidět zóna snímání předních a zadních parkovacích senzorů a sloupcové grafy zatížení na jednotlivých kolech.

5.3.9 IPG Control

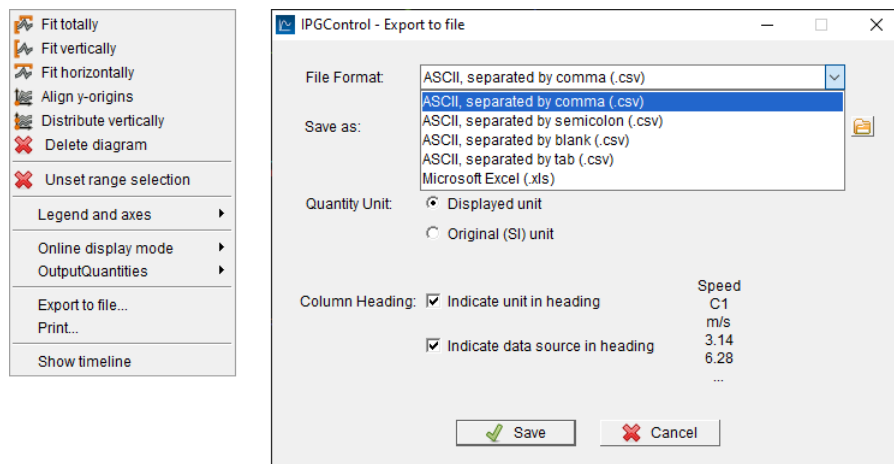
Nástroj pro vizualizaci a analýzu. IPGControl lze použít na zobrazení vybraných výstupních veličin v reálném čase, exportovat soubory dat simulace, zavést data z vybraných veličin do grafů a pro další analýzu.



Obr. 10: IPGControl

Volba měřených veličin a vzorkovací frekvence Pro správné získání dat je nejprve nutné zvolit měřené veličiny z dostupného seznamu (Pravá strana obr. 10). Veličiny zvolíme dvojklikem a zda se to zdařilo poznáme například podle legendy v levé části hlavního okna. Následně je ještě před spuštěním simulace nutné nastavit vzorkovací frekvenci se kterou data snímáme, která je společná pro všechna data, jelikož jí není možné změnit po provedení testu a s její změnou obdobně jako i se změnou sledovaných veličin je tedy nutné simulaci opakovat.

Export dat Po získání naměřených dat ze simulace je klíčové ze softwaru CarMaker data exportovat. To je možné po kliknutí pravým tlačítkem myši kamkoliv na graf. Následně si ještě volíme destinaci, kam se mají data uložit a jeden ze dvou formátů - .csv (s podporou různých oddělovačů) nebo přímo .xls pro Microsoft Excel.



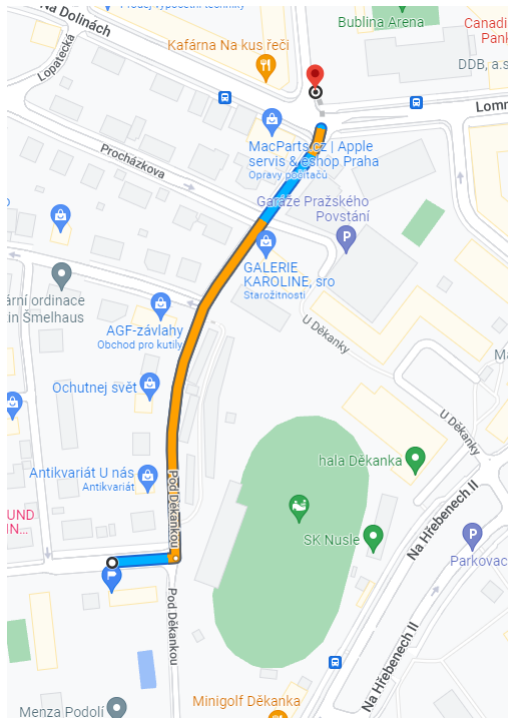
Obr. 11: Export dat

5.4 Řešení konkrétní situace v simulaci

Nyní si popíšeme konkrétní řešení provedených testů. Volba prostředí je podřízena dvěma prováděným vyhodnocením situací, o kterých hovoříme v pozdější části práce.

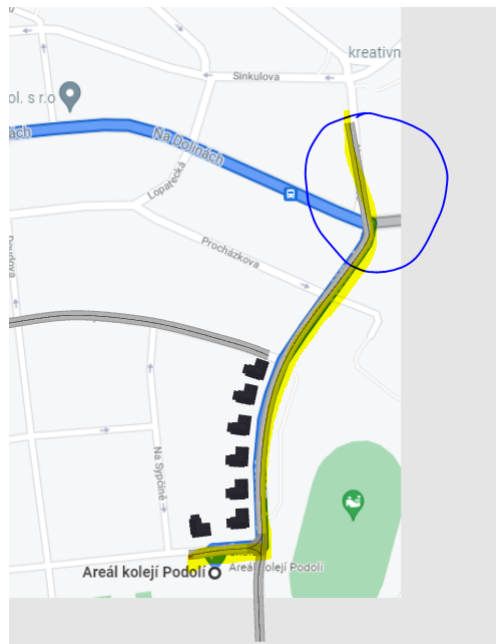
5.4.1 Zvolený testovací úsek

Pro testovací účely jsem zvolil krátký úsek silnice u Podolských kolejí ČVUT viz obr.12 a to čistě z důvodu příhodné lokace, kterou podrobně znám a vím, jaká úskalí by mohla skrývat. Tento úsek testovací trasy je v simulaci čistě za účelem získání šumu na čidlech z reálných situací, do kterých se vozidlo při provozu dostává. Data o cestě jsou do simulačního prostředí importována z prostředí Google Earth.



Obr. 12: Trasa v Google Maps

Úsek silnice použitý v simulačním prostředí (vyznačen žlutě na obr. 13) je doplněna o fiktivní rovný úsek (vyznačen modrou kružnicí na obr.13) na kterém je příčný práh, díra a úsek na kterém jsou místo asfaltu dlažební kostky. Rozměr prahu je podle normy TP-085 z roku 1996.[2]



Obr. 13: Trasa v IPG CarMakeru

Co se týče variant projetí testu, tak jsem se zaměřil na změnu rychlosti, kterou vozidlo projíždí segmentem a na konfiguraci volby převodových poměrů.

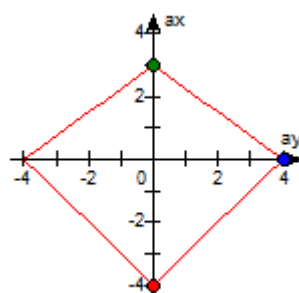
5.4.2 Zvolené vozidlo a pneumatiky

Jako vozidlo pro sledování jsem použil předvytvořenou simulovanou verzi automobilu Audi TT z důvodu nejpodobnějších rozměrů z dostupné knihovny s vozidly ve vozovém parku společnosti Uniqway a zároveň je taky součástí koncernu VAG, a tudíž sdílí spoustu dílů s vozy Škoda. Avšak do předvytvořeného vozu jsem zasáhl, a to změnou rozložení pohonu a to změnou na pouze přední náhon z původní 4WD jednotky Haldex, kterou vozidlo Audi TT v této verzi disponuje. Systém Haldex je sice dostupný i pro vozy značky Škoda, ale v carsharingu Uniqway jím žádné auto nedisponuje, a proto jsem zvolil verzi s předním náhonem.

Jako model pneumatik jsem zvolil jeden z rozměrů používaný v carsharingu Uniqway a to přesně 225/50 R17 na všech čtyřech kolech.

5.4.3 Zvolený řidič

Řidič je zvolený podle předvolby běžně agresivního řidiče, kterou nabízí software IPG CarMaker. Znamená to, že řidič jede s maximálním podélným zrychlením 3 m/s^2 a zpomalením 4 m/s^2 . Maximální příčné zrychlení je pak také 4 m/s^2 . Kombinace se podélného a příčného zrychlení se řeší podle g-g Diagramu (viz obr. 14) definicí křivky. Pro účel testu jsem nechal extrémy spojené přímkou, což je výchozí hodnota. Na ose Y je zrychlení podélné a na ose x je zrychlení příčné.



Obr. 14: g-g diagram

5.4.4 Sledované veličiny

Ze simulace exportujeme 11 hodnot, a to rychlost otáčení každého ze čtyřech kol vozidla v rad/s , čas, ujetou vzdálenost v m , otáčky motoru v rad/s , zrychlení podle všech třech os v rad/s^2 a pravděpodobnost řazení. Pro samotné vyhodnocení dat používáme pouze zrychlení ve směru úhlu sklonu, čas, otáčky motoru a rychlosti kol, které by ale mohly být nahrazeny

údajem o rychlosti vozidla přímo z jednotky. Sledování všech čtyřech kol nám dává výhodu při volbě své vlastní metody výpočtu rychlosti vozidla.

5.4.5 Export dat pro Matlab

Pro export jsem zvolil formát .csv oddělený čárkami. Důvodem byla jednoduchost exportu dat do software Matlab, protože ten je schopný importovat data rovnou bez jakéhokoliv zásahu jedním příkazem a vytvořit z nich zpět matici naměřených hodnot.

Volba vzorkovací frekvence je složitější. Na straně schopností softwaru CarMaker jsme sice omezeni až na 1000 Hz (Teoreticky možno navýšit na 10 000 Hz), ale jednotka, která je k dispozici v automobilech carsharingu není ani zdaleka schopná načítat data touto obrovskou rychlostí. Ve skutečnosti jsme schopni se u ní přiblížit k rychlosti čtení necelých 10 Hz. Proto je použití takto vysokých hodnot pro výstup ze simulace zbytečné. Zároveň si ale necháváme jistý prostor pro snižování frekvence uměle v softwaru Matlab a proto jsem nakonec zvolil vzorkovací frekvenci 20 Hz, kterou můžeme jednoduše snížit na polovinu. Z gyroskopu s akcelerometrem, který je k dispozici pro Uniqway je možné číst i rychleji než 20 Hz, ale 20 Hz se prokázalo jako dostačující pro vyhodnocení průjezdu výmolem, a proto není nutné frekvenci zvyšovat a vytvářet data, která nemají žádný přínos a jen zabírají prostor v paměti a zpomalují vyhodnocení simulace. Zároveň není potřeba žádná aproximace dat mezi sebou, protože každá hodnota z jednotky má svou příslušnou hodnotu zrychlení a naopak.

6 Matlab

Matlab (ze slov "matrix laboratory") je software a programovací jazyk čtvrté generace vyvíjený společností MathWorks určený pro vědeckotechnické výpočty. Umožňuje počítání s maticemi vykreslování grafů (2D a 3D), algoritmicizaci, analýzu dat a vytváření aplikací, rozhraní aj. [7]

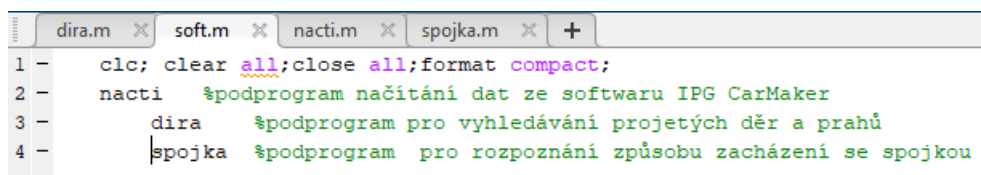
Software byl vytvořen pro verzi Matlab R2021a.

6.1 Důvod použití

Důvodem použití softwaru Matlab pro vyhodnocení dat ze simulace je fakt, že software patří mezi světovou špičku ve vyhodnocování vědeckých dat a jeho použití je velmi uživatelsky příznivé a relativně nenáročné na naučení.[7]

6.2 Hierarchie programu

Software, vytvořený pro zpracování dat je rozdělen do několika podprogramů spouštěných hlavním programem *soft.m* viz obr. 15. Důvodem je hlavně zjednodušení uživatelského ovládání a možnosti tvorby případných nástaveb anebo dělání úprav bez zásahu do ostatních částí programu. Zároveň nám rozdělení programu umožňuje využít práce více programátorů, kteří by mohli pracovat nezávisle na sobě při tvorbě nebo úpravě programů pro vyhodnocování stylu jízdy. Dosavadními částmi programu jsou tři podprogramy: *Nacti.m*, *Dira.m* a *Spojka.m*. Jak už názvy napovídají první z programů se stará o import dat z prostředí IPG CarMaker, druhý o rozpoznávání projetých výmolů a příčných prahů a poslední o sledování řidičova zacházení se spojkou.

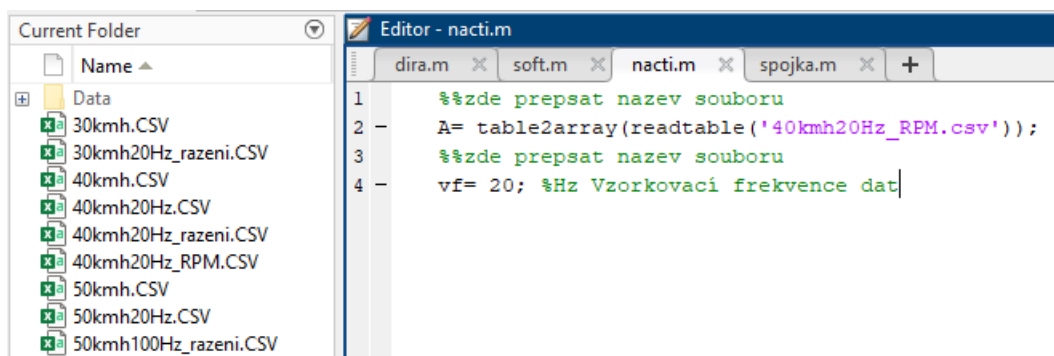


```
1 - clc; clear all; close all; format compact;
2 - nacti %podprogram načítání dat ze softwaru IPG CarMaker
3 -     dira %podprogram pro vyhledávání projetých děr a prahů
4 -     spojka %podprogram pro rozpoznání způsobu zacházení se spojkou
```

Obr. 15: Hlavní rozcestí programu

6.3 Import dat ze simulace

Na obr. 16 vidíme, že software Matlab je perfektně připraven na import dat ze souborů .csv. Stačí nám tedy jedna funkce *table2array*, která je jeho součástí. Program pak automaticky rozpozná tabulku a převede ji na matici a to tak, že sám zvolí dimenze matice, a dokonce i automaticky vyřadí první řádky ve kterých jsou zapsané názvy veličin a jednotky (zachová pouze číselné hodnoty). Celou načtenou matici přiřadíme pod proměnnou *A*, pro další operace. V neposlední řadě vyplníme vzorkovací frekvenci dat. Je pravda, že by bylo možné jí lehce vypočítat podle času a počtu naměřených hodnot, ale není nutné přidávat na komplexitě programu a zároveň se dostat do komplikací v případných chyb v zaokrouhlování při větším počtu naměřených hodnot. Tuto hodnotu není potřeba vypočítávat, když jí známe.



```
1 %%zde prepsat nazev souboru
2 A= table2array(readtable('40kmh20Hz_RPM.csv'));
3 %%zde prepsat nazev souboru
4 vf= 20; %Hz Vzorkovací frekvence dat
```

Obr. 16: Načítání dat do prostředí Matlab

7 Sledování průjezdů výmoly

Vyhledávání výmolů, příčných prahů a jiných nedokonalostí vozovky je poměrně komplikovaná disciplína, jelikož v osobních autech značky Škoda, které jsou k dispozici v carsharingu nejsou přítomny žádné senzory, které by výmoly přímo detekovaly jako vlastní veličinu. Z tohoto důvodu je nutné vyhledávat výmoly pomocí předvídání chování jiných senzorů při průjezdu výmolem a následné vyhodnocení, zda je chování podobné navržené modelové situaci. Vyladění potřebné míry podobnosti s modelovou situací je možné provést experimentální metodou, a to konkrétně nasbíráním dat z cesty o definovaném profilu a s definovaným počtem výmolů. Následně je možné upravit myšlenou charakteristiku chování senzoru ve výmolu tak, aby lépe odpovídala reálným datům. Toto vylepšení vyhledávacího vzorového výmolu s největší pravděpodobností povede i k vylepšení přesnosti vyhodnocovacího algoritmu.

Pro vyhodnocování dat senzorem, který není určen přímo pro měření dané veličiny bohužel platí, že mezi daty, které naměříme při průjezdu výmolem jsou i data ze situací, které nás v tuto chvíli nezajímají - takzvaný šum. Na první pohled by se mohlo zdát, že nám šum nevadí, ale je velmi důležité, aby nedošlo ke zmatení algoritmu, který by jako výmoly mohl označit i data jiná. Z tohoto důvodu musí být vzorový výmol, v datech pokud možno jedinečným identifikátorem.

K vyhledávání výmolů v silnici na základě dat dostupných z jednotky sdíleného osobního automobilu jsem použil gyroskop s akcelerometrem, a to konkrétně údaj o zrychlení ve směru úhlu sklonu vozidla. Z dat tohoto senzoru je možné rozlišit výmoly a odlišit je od šumu převážně z důvodu vyšší amplitudy. Mezi úskalí této metody bohužel patří, že malé výmoly nebo výmoly projeté nízkou rychlostí jsou z veliké části pohlceny tlumiči, z důvodu umístění akcelerometru uvnitř v těžišti, nikoli jako součásti neodpružené hmoty, a proto se touto metodou nedají detekovat. Tuto skutečnost je, ale možné zanedbat, protože se soustředíme na veliké výmoly, které by mohly poškozovat vozidlo, které se na naměřených datech projevovaly budou.

Jako identifikátor výmolu jsem zvolil "dvojitě zhoupnutí". Jedná se o stav, kdy se na krátkém časovém intervalu vystřídají dvě a dvě vysoké amplitudy zrychlení v opačném směru.

Příložený detekční software prochází průběh hodnot zrychlení. Na celém intervalu hledá hodnoty, které převyšují stanovenou výchytku. Druhá část programu následně prochází průběh a hledá shluky míst, kde dochází ke dvojitému zhoupnutí. Program tyto místa vypíše, sdělí nám rychlost průjezdu přes výmol a vyznačí ho do grafu.

7.1 Vysvětlení programu v softwaru Matlab

Následující část proběhne soustavou komentářů ke kompletnímu zdrojovému kódu na obrázcích 17-23, které jsou řazené chronologicky tak, jak jsou seřazeny v programu.

```
1 %% Definice konstant
2 - prah_h=2.3; %horní práh rozpoznání zrychlení|
3 - prah_d=-2.3; %dolní práh rozpoznání zrychlení
4 - cas_tolerance=1; %s % čas jak dlouho může trvat průjezd výmolem
5 - polomer_kola=656.8/2/1000; % poloměr kola se zanedbáním jeho zmenšení v důsledku deformace zatížení
6 - Rychlost=(A(:,4)+A(:,5)+A(:,6)+A(:,7))./4; %rychlost jednotlivých kol
7 - pr_Rychlost=mean(Rychlost); %průměrné otáčky kola
8
```

Obr. 17: Definice konstant

Obrázek 17 V první části programu definujeme, konstanty, které jsou použité v programu. Jako první definujeme dolní a horní prahovou hodnotu výchyly zrychlení, kterou program detekuje. Experimentálním způsobem jsem dospěl k tomu, že ideální hodnoty pro zajištění správné funkčnosti programu jsou 2.3g v obou dvou směrech. Je však nutné podotknout, že se tato hodnota může u různých vozidel a pozic gyroskopu s akcelerometrem lišit a tím pádem být zároveň využita pro doladění programu, aby správně fungoval pro konkrétní vozidlo. Dále definujeme proměnnou *cas.tolerance*, kterou používáme na určení nejdelší možné doby průjezdu výmolem. Tato konstanta by mohla pro upřesnění být do budoucna nahrazena opravdovým výpočtem na základě rychlosti vozidla, ale to s sebou přináší další komplikace, protože by bylo nutné vyhodnotit, jak dlouho po přejetí výmolu se vozidlo houpe, což by byla s největší pravděpodobností funkční závislost na rychlosti průjezdu. Poslední tři proměnné vedou na výpočet rychlosti vozidla. Nejdříve si vypočteme poloměr kola, zde je nutno podotknout, že zanedbáváme deformaci pneumatiky zatížením. Data jsou z toho důvodu drobně zkreslená. Rychlost otáčení kol vozidla nakonec počítáme jako průměrnou hodnotu úhlové rychlosti všech čtyřech kol.

```
10 %% Graf rychlosti vozidla
11 - t=A(:,11); %aktuální cas
12 - hold on
13 - tiledlayout(2,1)
14 - nexttile
15 - plot(t,Rychlost.*polomer_kola.*3.6);
16 - legend('Rychlost [km/h]')
17 - xlabel('t [s]')
18 - hold off
19
```

Obr. 18: Graf rychlosti v závislosti na čase

Obrázek 18 V druhé části programu se zaměříme na tvorbu grafu aktuální rychlosti. Aby to bylo možné definujeme časovou proměnnou *t* jako jedenáctý sloupec matice *A* exportované ze simulace. Graf aktuální rychlosti získáme jednoduchým vynásobením průměrné úhlové rychlosti poloměrem kola a převedením z m/s na km/h vynásobením 3.6 v závislosti na čase *t*. Nakonec ještě popíšeme osy a vytvoříme legendu. Příkaz *hold* zaručuje

vložení všech věcí do jednoho grafu. Graf rychlosti vozidla je jedním ze dvou grafů v jednom okně. Aby tomu tak bylo používáme příkaz *tiledlayout*. Mezi jednotlivými grafy v okně se přepínáme příkazem *nexttile*.

```

20 %% Graf vyhodnocení výmolů
21 - nexttile
22 - hold on
23 - xlabel('t [s]')
24 - plot(t,A(:,2));
25 |

```

Obr. 19: Začátek grafu projetých výmolů

Obrázek 19 V další části začínáme se samotným vyhodnocení výmolů/prahů. Jako první krok se přepneme do druhé poloviny okna s grafem, ve kterém si popíšeme osu x a zaneseme do něj data zrychlení ve směru úhlu sklonu.

```

26 %% hlednani děr Definice boolovských matic které vyhodnocují zrychlení v porovnání s prahem
27 - ADh= A(:,2)>prah_h;
28 - ADd= A(:,2)<prah_d;
29
30 %% Čištění řady od opakujících se 1
31 - h=1;
32 - s=1;
33 - while h<size(ADh,1)-1
34 -     if ADh(h)==1
35 -         while true
36 -             if ADh(h+s)==1
37 -                 ADh(h+s)=0;
38 -                 s=s+1;
39 -             else
40 -                 s=1;
41 -                 break
42 -             end
43 -         end
44 -     end
45 -     h=h+1;
46 - end
47 - i=1;
48 - u=1;
49 - while i<size(ADd,1)-1
50 -     if ADd(i)==1
51 -         while true
52 -             if ADd(i+u)==1
53 -                 ADd(i+u)=0;
54 -                 u=u+1;
55 -             else
56 -                 u=1;
57 -                 break
58 -             end
59 -         end
60 -     end
61 -     i=i+1;
62 - end
63 - plot(t,ADh)
64 - plot(t,ADd)
65 - legend('Pitch acc','Zhoupnutí kladné','Zhoupnutí záporné')

```

Obr. 20: Hledání extrémů a úprava hodnot

Obrázek 20 Ve čtvrté části, probíhá vyhodnocení dat zrychlení. Zdefinujeme si matice pravdivostních hodnot ADh a ADd . Hodnota je 1 (pravda) vždy, když hodnota přesahuje

jednu z prahových hodnot a 0 (nepravda), pokud se pohybuje mezi nimi. Naměřená data ale nevyhovují principu vyhodnocení. Z toho důvodu používáme while cykly. Protože nás zajímá vyhodnocení počtu výchylek, potřebujeme upravit jednotlivé matice, tak aby měli vždy jen jednu hodnotu 1 (pravda) pro každou výchylku. Pro lepší představu situace si předvedeme funkci cyklů na příkladu.

Máme soustavu dat: 0 1 1 1 0 0 1 1 0

Stejná soustava po úpravě: 0 1 0 0 0 1 0 0

Tuto úpravu provedeme pro obě matice pravdivostních hodnot ADh a ADd . Proto používáme 2 stejné cykly. Nakonec promítneme obě upravené matice do grafu a přidáme legendu.

```

66     %% Vyhodnocení dvojitého zhoupnutí
67 -   pocet_der_100pr=0;
68 -   pocet_der_75pr=0;
69 -   AVh=zeros(1,ceil(cas_tolerance*vf));
70 -   AVd=zeros(1,ceil(cas_tolerance*vf));
71 -   d=1;
72     %% hledání děr a prahů
73 -   while d<size(A,1)-ceil(cas_tolerance*vf)
74 -       AVh=ADh(d:d+ceil(cas_tolerance*vf)-1);
75 -       AVd=ADd(d:d+ceil(cas_tolerance*vf)-1);
76 -       if and(and(sum(AVh)>=2,sum(AVd)>=2),AVh(1)==1)%or(AVd(1)==1,AVh(1)==1)
77 -           pocet_der_100pr=pocet_der_100pr+1;
78 -           cas=d/vf;
79 -           fprintf('Nalezena diera v case %u sekund v rychlosti %d km/h \n',ceil(cas),ceil(Rychlost(d)*polomer_kola*3.6))
80 -           plot(d/vf,0, 'r+');
81
82
83 -           d=d+ceil(cas_tolerance*vf);
84 -           end
85 -           if and(and(sum(AVh)>=2,sum(AVd)>=2),AVd(1)==1)
86 -               pocet_der_100pr=pocet_der_100pr+1;
87 -               cas=d/vf;
88 -               fprintf('Nalezen prah v case %u sekund v rychlosti %d km/h \n',ceil(cas),ceil(Rychlost(d)*polomer_kola*3.6))
89 -               plot(d/vf,0, 'b+');
90
91 -               d=d+ceil(cas_tolerance*vf);
92 -               end
93 -           end
94 -           d=d+1;
95 -       end
96 -   c=1;

```

Obr. 21: Vyhledávání děr a prahů

Obrázek 21 V páté části kódu se zaměřujeme na samotné vyhledání dvojitého zhoupnutí. Jako vyhodnocení nám nestačí jednoduchá suma výchylek, a to z několika důvodů. Je důležité, jak jsou výchylky rozmístěné oproti sobě, což nás zbavuje šumu a zároveň musíme dbát na to, abychom nedetekovali stejný výmol vícekrát. Proto si nejprve stanovíme na základě předpokládané doby trvání průjezdu přes výmol/práh a vzorkovací frekvence počet hodnot, který uvažujeme pro naše vyhodnocení. Vytvoříme tedy matice AVh a AVd , které obsahují tento počet hodnot. Nyní si můžeme vyhodnocení představit jako posuv matice AV maticí AD . O posuv se stará while cyklus. Jedna iterace cyklu vždy funguje tak, že použijeme matici AV z předchozí iterace, vymažeme první hodnotu všechny hodnoty

následně posuneme o jedno místo vlevo a poslední prázdné místo nahradíme dalším číslem v pořadí z matice AD . Takto vytvoříme matici AVh i AVd . Jako výmol nebo práh situaci označíme, jestliže v daném časovém intervalu je jak v matici AVh , tak i v matici AVd dvakrát hodnota 1 (pravda), takzvané dvojité zhoupanutí. Abychom upřesnili pozici výmolu, přidáváme ještě další podmínku a tou je skutečnost, že první hodnota v matici AVh nebo AVd musí být 1 (pravda). Díky tomuto upřesnění je taky možné situace oddělit na průjezd dírou a průjezd přes práh. Pokud první je první 1 (pravdivá hodnota) v matici AVh jedná se o překročení horního prahu - jev charakteristický pro průjezd výmolem. Pokud je první 1 (pravda) u AVd jedná se o průjezd přes příčný práh. Jako poslední věc musíme zajistit, abychom nedetekovali stejný práh nebo výmol vícekrát. O to se postaráme tak, že vždy když detekujeme jedno z nich, tak následující matice AVh a AVd vytvoříme kompletně nové z dat, které jsou v maticích AD posunuté o rozměr matice AV (nebereme žádnou z hodnot stávající matice AV znovu v potaz). Tato skutečnost má bohužel svoje úskalí. Mohlo by se stát, že pokud by řidič přejel dva výmoly v krátkém časovém intervalu (v rámci velikosti proměnné `cas.tolerance`), tak by se dva výmoly počítaly jako jeden. Je to ale zvoleno takto, protože alternativní řešení by mohlo znamenat, že bychom jeden výmol počítali jako výmolů více, což by bylo nepřijatelné z hlediska férovosti, kdybychom podle detekce chtěli odvozovat nějaké postihy pro řidiče vozidla. Jako poslední detail této části kódu stojí za zmínku fakt, že pokud zjistíme průjezd výmolem či přes práh, tak software vypíše, v jakém čase a v jaké rychlosti k němu došlo a zaznačí místo křížkem do grafu. Modrým pro práh a červeným pro výmol.

```

97   %% Hledání situací kdy možná došlo ke dvojitému zhoupanutí, ale z nějakého důvodu jsou k dispozici pouze 3 vrcholky zrychlení
98   while c<size(A,1)-ceil(cas_tolerance*vf)
99       AVh=ADh(c:c+ceil(cas_tolerance*vf)-1);
100      AVd=ADd(c:c+ceil(cas_tolerance*vf)-1);
101      if sum(AVh)+sum(AVd)>=3
102          pocet_der_75pr=pocet_der_75pr+1;
103          c=c+ceil(cas_tolerance*vf);
104
105      end
106      c=c+1;
107   end

```

Obr. 22: Případy k zamyšlení

Obrázek 22 Šestá část se zabývá problematikou situací, kdy z požadovaných 4 výchylek detekujeme pouze tři. Tento nástroj slouží zejména k manuální kontrole dat. Na základě něj je možné ladit detekci změnou prahů. Pokud víme, k jaké situaci doopravdy došlo anebo například zjistíme, že nějaká data, které jsme považovali za šum se téměř projevila jako práh nebo výmol. Software ještě vypisuje, kolik takových situací nastalo jako rozdíl počtu těchto hodnot a hodnot opravdu vyhodnocených děr a výmolů.

```

108 %% výpis výsledků
109 - legend('Pitch acc','Zhoupnutí kladné','Zhoupnutí záporné')
110 - disp("Pocet der/prahu")
111 - disp(pocet_der_100pr)
112 - $pocet_der_75pr=pocet_der_75pr-pocet_der_100pr
113 - hold off

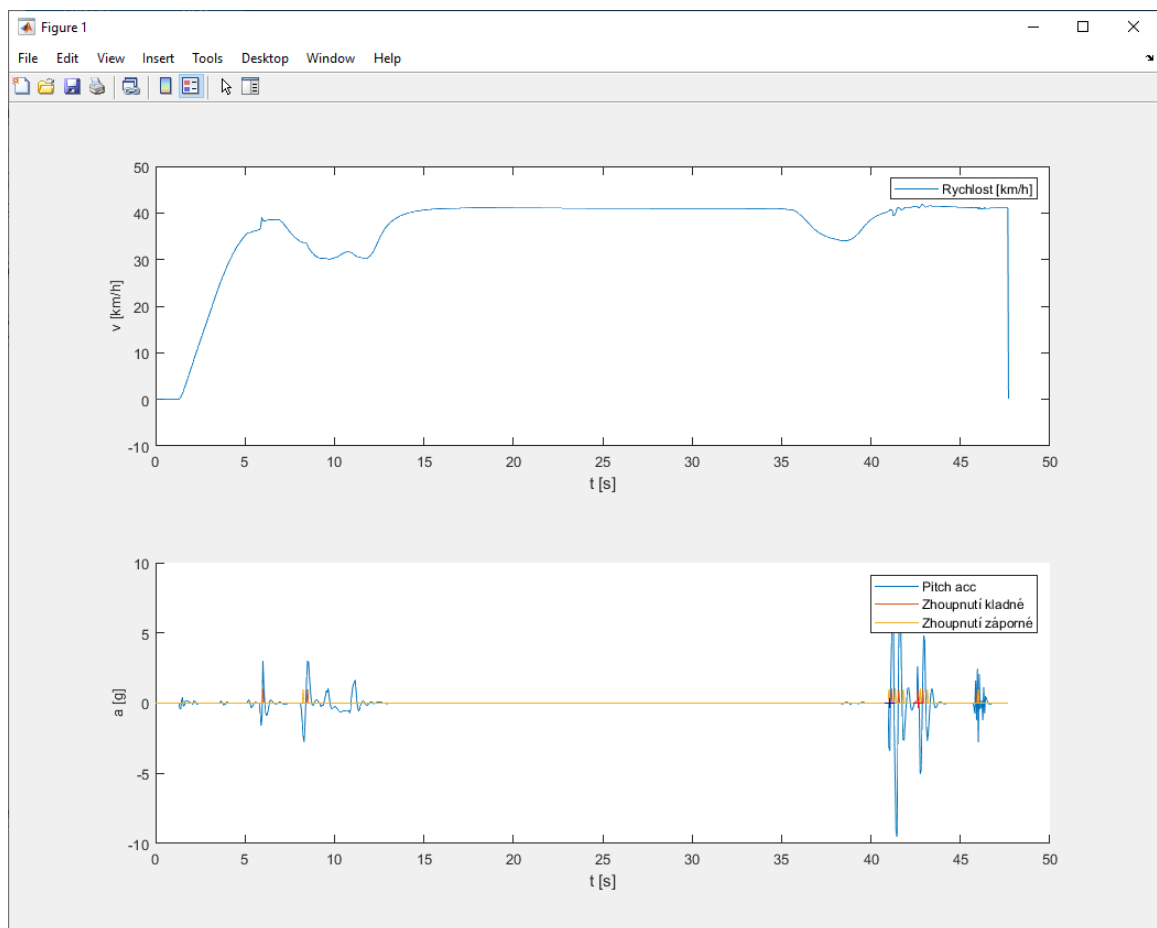
```

Obr. 23: Výpis výsledků

Obrázek 23 V poslední části programu tvoříme legendu ke grafu a vypisujeme celkový počet projetých děr/výmolů.

7.2 Výstup ze softwaru

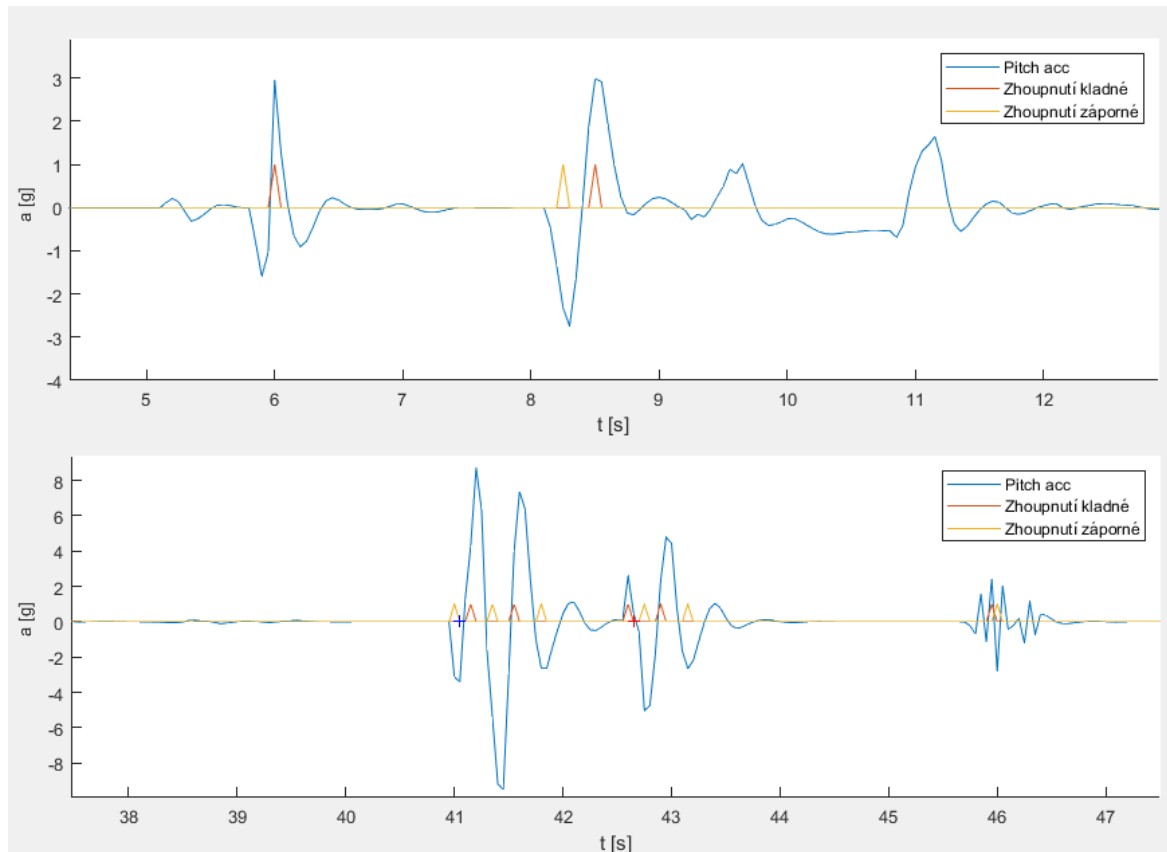
Výstupem ze softwaru jsou dva grafy a textový výstup o průjezdu.



Obr. 24: Graf průjezdu 40 km/h

Vše si nejprve ukážeme na průjezdu definovaným testovacím úsekem rychlostí 40 km/h. Jako první se podívejme na graf (Obr.24). Ve vrchní části vidíme rychlostní profil cesty. Řidič se snaží na měřeném úseku držet rychlost 40 km/h, ale zároveň se chová jako opravdový řidič a dodržuje stanovená pravidla, která jsme definovali, a tudíž rychlost kolísá v závislosti na terénu. Na druhém grafu jsou viditelná data ze zkoumání průjezdu přes prahy/výmoly.

Modře je na grafu vyznačená neupravená hodnota zrychlení přímo ze simulace. Červeně je znázorněna hodnota nad-prahové výchylky nad horní hranici a žlutě pod hranici dolní. Červenými křížky jsou vyznačeny průjezdy výmolem a modře průjezdy přes příčný práh. Je možné sledovat, že při rychlosti 40 km/h funguje program podle našich představ. Software detekuje oba extrémy, výmol i práh.



Obr. 25: Graf průjezdu 40 km/h - DETAIL

Na obr. 25 je zobrazen detail z obr.24, kde je k dispozici lepší pohled na spodní graf vyhodnocení průjezdu. Na horním detailu je vidět, jak do dat přispívá šum z průjezdu reálnou trasou bez výmolů nebo příčných prahů. Nad-prahové výchylky se zde nacházejí, ale nejsou organizované a nelze hovořit o jasně viditelném dvojitém zhoupnutí. Zatímco na spodním detailu na uměle vytvořené oblasti s výmolem a příčným prahem je charakteristické dvojitě zhoupnutí jasně viditelné. Okolo 46. sekundy dále pozorujeme šum od průjezdu přes dlažební kostky, u kterého by se dalo hovořit o jisté pravidelnosti, ale naopak výchylky zde nedosahují zdaleka stejné výšky. Nutno navíc podotknout že nerovnost dlažebních kostek je volena vyšší, než se kterou se v běžném provozu setkáváme. V neposlední řadě si můžeme prohlédnout, že výchylky od díry a prahu začínají akcelerací na druhou stranu a z toho důvodu je možné je odlišit.

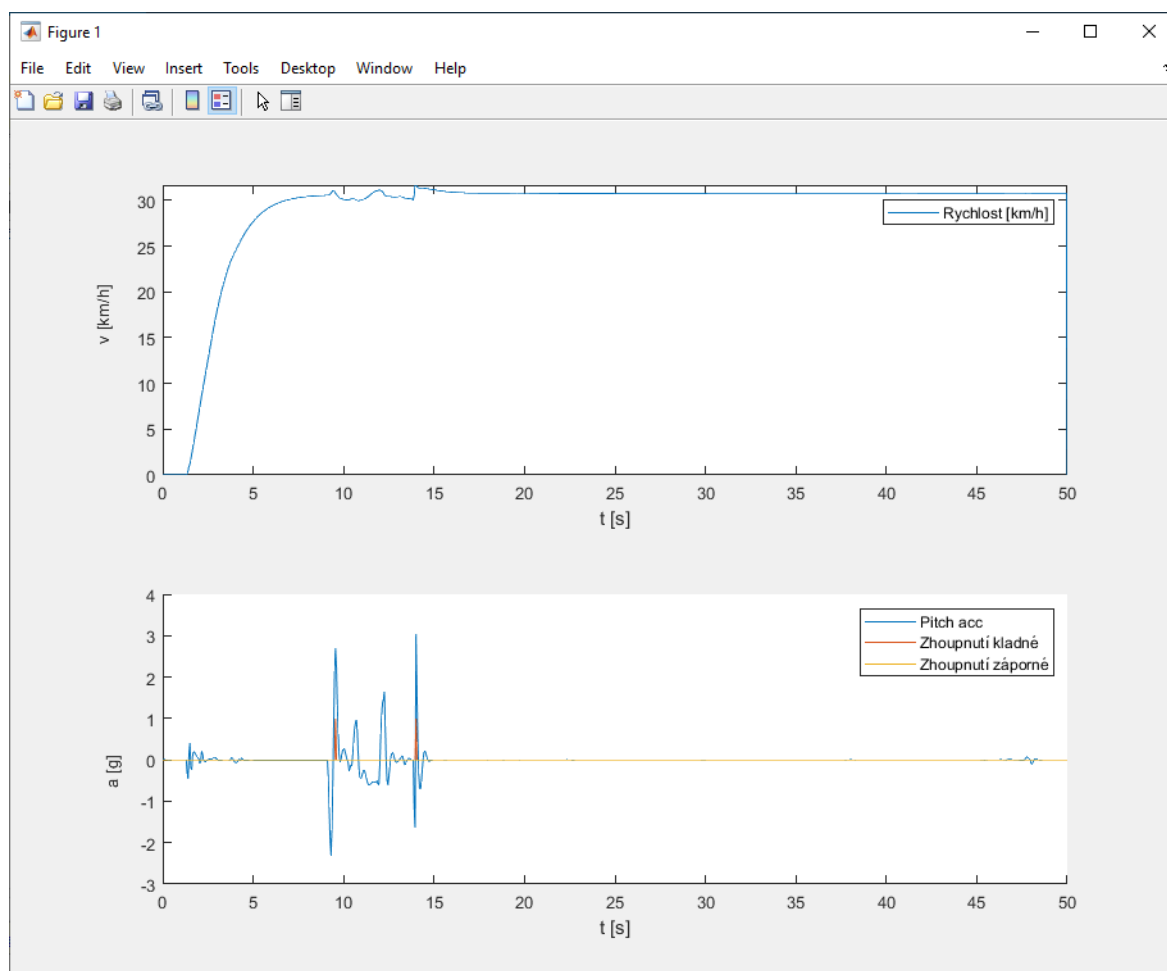

```

Nalezen prah v case 42 sekund v rychlosti 41 km/h
Nalezena dira v case 43 sekund v rychlosti 42 km/h
Pocet der/prahu
      2
fx >>

```

Obr. 26: Textový výstup průjezdu 40 km/h

Na obr. 26, je pro ukázkou textový výstup z vyhodnocení průjezdu v rychlosti 40 km/h. Software vypisuje jednotlivé události, a nakonec vypíše i celkový součet vyhodnocených událostí.



Obr. 27: Grafy při průjezdu rychlostí 30 km/h

Jak už bylo naznačeno dříve existuje určité úskalí této metody vyhodnocení průjezdů a tím není nic jiného než průjezd přes práh/výmol pomalou rychlostí z důvodu dobrého technického řešení moderních automobilů, které jsou schopné v nižších rychlostech dvojitě zhoupnutí od průjezdu velmi dobře zatlumit. Na obr. 27 z průjezdu identickou situací rychlostí pouze 30 km/h je viditelné, že v místě, kde jsme při průjezdu vyšší rychlostí na obr. 24 pozorovali data o zrychlení jsou výchylky opravdu nízké, a proto z nich není možné průjezdy detekovat. Na druhou stranu by se tato situace dala označit za vhodný průjezd,

protože je viditelné, že byl proveden v rámci schopností vozidla, a i opotřebení z takového průjezdu by mělo být mnohem nižší.

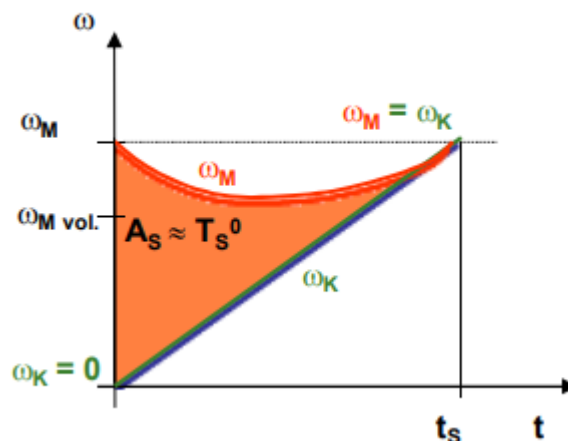
8 Sledování zacházení se spojkou

Druhou problematikou, kterou jsem zvolil jako vhodnou pro sledování je zacházení se spojkovým pedálem. Podobně jako u sledování průjezdů přes výmoly/prahy v automobilu není k dispozici žádný senzor kterým bychom mohli přímo sledovat pohyb pedálu nebo ještě lépe dění přímo na samotné spojce. I toto řešení se nabízí. Bylo by například možné použít dva otáčkoměry, jeden na vstupním hřídeli převodovky a druhým na setrvačnicku nebo například teplotní čidlo na samotném kotouči spojky anebo například senzory polohy a zatížení spojkového pedálu. Oproti použití stávajících senzorů se ale jedná o nákladná řešení, a to zejména kvůli tomu, že by musela být instalována do každého vozidla ve vozovém parku a ještě se nabízí argument, zda v té situaci není ekonomičtější se zaměřit například na zlevnění samotné výměny spojky.

Motivací pro sledování zacházení se spojkou v prostředí carsharingu, je zejména skutečnost, že u moderních aut, které společnost Uniqway používá je spojka považovaná za součást, která by bez výměny měla mít stejnou životnost jako vozidlo samotné. Avšak v prostředí vozového parku Uniqway je běžné, že odchází. Nabízí se tedy otázka, co stojí za tímto nadměrným opotřebením. Existuje několik teorií, ze kterých vyzvednu dvě nejpravděpodobnější, mezi kterými ale není bez vhodné validace možné rozhodnout. Rozdíl mezi těmito teoriemi spočívá zejména v tom, jestli je vysoké opotřebení problémem jednotlivců nebo spíše problémem u většiny uživatelů služby Uniqway. První z variant by mohla být, že existuje nízké procento lidí, které se k vozům chová bezdůvodně bezohledně anebo jednoduše nemají natolik dobré řídičské schopnosti, aby s vozidly mohli adekvátně zacházet. Jedním z důvodů by mohla být registrace do služby Uniqway, při které není nutné nijak prakticky dokázat řídičské dovednosti jinak než předložením platného řídičského průkazu. Druhá z teorií, které se nabízí by mohla být, že uživatelé, kteří si automobily půjčují mají většinou nízké zkušenosti s jízdou. Ať už je to z důvodu, že sami auto nevlastní nebo aby řídili správně, potřebují delší dobu pro přizpůsobení na řízení jiného vozidla, než jsou zvyklí, která se jim ale při průměrné délce výpůjčky pod dvacet kilometrů nedostává a je to tedy spíše skupinový problém většiny uživatelů.

Aby bylo toto možné vytvořit potřebujeme nástroj, který by umožňoval zacházení se spojkou sledovat. Pro pochopení je nezbytné stanovit, jak se spojka opotřebovává. Spojka v automobilu se spalovacím motorem umožňuje plynulé řazení rychlostních stupňů bez potřeby vyrovnávat otáčky setrvačnicku motoru a vstupního hřídele převodovky (například takzvaným mezi-plynem) a zároveň umožňuje rozjezd zastaveného vozidla,

protože spalovací motor běží nejnižší v takzvaných volnoběžných otáčkách (Nevytváří moment při nulových otáčkách) jak je tomu například u elektromobilů. Přenos momentu je zajištěn třecími segmenty. Protože jde o tření, tak kromě přenosu momentu se značná část energie přeměňuje na teplo, které zahřívá spojkovou lamelu, kde z důvodu vysokých teplot dochází k nadměrnému opotřebení třecích segmentů. Celou problematiku perfektně vystihuje obr.28, kde jsou červenou čarou vyznačeny otáčky motoru a zelenou čarou vyznačeny otáčky vstupního hřídele převodovky. Oranžová plocha pak představuje teplo, které vzniká ztrátami.[6]



Obr. 28: Graf rozjezdu vozidla[6]

Navrhovaný způsob řešení sleduje dvě veličiny. Otáčky motoru a rychlost vozidla. Otáčky motoru čteme opět z jednotky ve vozidle a pro rychlost používáme stejné hodnoty jako z měření průjezdu přes výmoly/prahy. Podle známých převodových poměrů přepočítáváme otáčky kol na otáčky na vstupním hřídeli převodovky a porovnáváme je s otáčkami motoru. Zde se nabízí ještě diskuse, která kola použít pro výpočet rychlosti. Pokud bychom chtěli sledovat čistě prokluz spojkové lamely, tak je potřeba sledovat rychlost pouze kol hnaných. Pokud bychom stav protáčených kol označovali též za stav, který opotřebovává vozidlo bylo by naopak nutné měřit rychlost z nápravy která hnaná není. Pro účely předvedení principu měření a ulehčení programové složitosti používáme pro simulaci údaj o průměrné rychlosti z předchozího měření, tudíž průměr ze všech čtyřech kol vozidla. Pokud by bylo potřeba sledovat jeden ze zmíněných případů, je úprava programu velmi jednoduchá.

8.1 Vysvětlení programu v softwaru Matlab

Následující část opět proběhne soustavou komentářů ke kompletnímu zdrojovému kódu na obrázcích 29-34, které jsou řazené chronologicky tak, jak jsou seřazeny v programu.

```

1 %% data
2 - ot=A(:,10).*(60/(2*pi)); %otáčky motoru v ot/min
3 - tol=0.01; %% tolerance rozpoznání zařazené rychlosti
4 - ot_rad=A(:,10);
5 - převod_převodovka=[2.93;1.792;1.185;0.829;0.623;0.495]; %%převodové stupně ze softwaru carmaker
6 - převod_diff=[5.732]; %% převod na diferenciálu

```

Obr. 29: Přiřazení dat a definice konstant

Obrázek 29 V první části programu definujeme určitá data, která jsou nutná pro výpočty obdobně jako u předchozího programu. Z matice A dvakrát načteme sloupec 10, který obsahuje data o otáčkách motoru. Jednou je načteme rovnou v rad/s a jednou je převedeme na ot/min do proměnné ot . Dále zadefinujeme konstanty: matici převodových poměrů převodovky, převodový poměr diferenciálu a toleranci odchytky zařazeného převodového stupně. Proměnné $Rychlost$ a t přebíráme z předchozí části programu.

```

7 %% tvorba grafů
8 - figure
9 - hold on
10 - tiledlayout(5,1)
11 - nexttile
12 - plot(t,ot);
13 - legend('Otáčky motoru [ot/min]')

```

Obr. 30: Inicializace tvorby grafu

Obrázek 30 V druhé části programu začínáme tvorbu okna, které je složeno z pěti grafů. Do prvního grafu v okně následně zobrazíme otáčky motoru v závislosti na čase.

```

14 %% Rychlost vozidla v jednotlivých rychlostních stupních
15 - nexttile
16 - plot(t,Rychlost);
17 - legend('Rychlost [km/h]')
18 - převod_vysledny=převod_převodovka.*převod_diff./3.661;
19 - rychlost1=ot_rad.*polomer_kola./převod_vysledny(1);
20 - rychlost2=ot_rad.*polomer_kola./převod_vysledny(2);
21 - rychlost3=ot_rad.*polomer_kola./převod_vysledny(3);
22 - rychlost4=ot_rad.*polomer_kola./převod_vysledny(4);
23 - rychlost5=ot_rad.*polomer_kola./převod_vysledny(5);
24 - rychlost6=ot_rad.*polomer_kola./převod_vysledny(6);
25 - nexttile
26 - hold on
27 - plot(t,rychlost1);
28 - plot(t,rychlost2);
29 - plot(t,rychlost3);
30 - plot(t,rychlost4);
31 - plot(t,rychlost5);
32 - plot(t,rychlost6);
33 - legend('Rychlostní stupeň 1 [km/h]', 'Rychlostní stupeň 2 [km/h]', 'Rychlostní stupeň 3 [km/h]'
34 - , 'Rychlostní stupeň 4 [km/h]', 'Rychlostní stupeň 5 [km/h]', 'Rychlostní stupeň 6 [km/h]');

```

Obr. 31: Výpočet fiktivních rychlostí

Obrázek 31 Další část programu se zabývá vytvořením fiktivních rychlostí vozidla. Na základě znalosti otáček motoru, převodových stupňů a poloměru pneumatiky vozidla vypočítáme jakou rychlostí by se vozidlo pohybovalo ve všech rychlostních stupních. Tato část je vytvořená pro vozidlo se šestistupňovou převodovkou, ale připravená na jednoduchou

změnu počtu stupňů. Do druhé části grafu je vyvedena závislost skutečné rychlosti vozidla na čase a do třetí části jsou vyvedeny fiktivní rychlosti ve všech šesti rychlostních stupních.

```

34     %% výpočet odchylek od rychlosti v jednotlivých převodových stupních
35 -   rychlost_v=[rychlost1 rychlost2 rychlost3 rychlost4 rychlost5 rychlost6];
36 -   matice_odchylek=rychlost_v./Rychlost;
37 -   prevod=matice_odchylek.*(matice_odchylek>=(1-tol));
38

```

Obr. 32: Tvorba matice odchylek rychlosti

Obrázek 32 Část popsaná na obr. 32 obsahuje jednu ze stěžejních myšlenek celého principu vyhodnocení zacházení se spojkou. Nejprve si vytvoříme jednu matici, do které sloučíme fiktivní rychlosti ve všech rychlostních stupních a následně jednotlivé sloupce vydělíme rychlostí skutečnou. Tím nám vznikne matice odchylek, ze které už poznáme, že číslo sloupce, ve kterém je hodnota přibližně 1 je náš požadovaný rychlostní stupeň. Nyní je potřeba ještě tuto matici vyhodnotit a vyřešit případy s odchylkou. Začneme s vyřešením jedné z chyb prostředí Matlab a to jsou operace s nulou. Vytvoříme tedy matici *prevod*, ve které všem číslům, která jsou hodně nízká (menší než proměnná *tol*) přiřadíme hodnotu 0 a zároveň přiřadíme nulovou hodnotu číslům menším než $1 - tol$, protože to jsou převody, které nemohou být zařazeny. V tomto případě by totiž otáčky motoru byly nižší než otáčky převodovky. Tato situace ale může nastat při jízdě nebo stání s vyřazenou rychlostí. S detekcí vyřazené rychlosti se však vypořádáme, až v další části.

```

39     %% Rozpoznání zařazené rychlosti a výpočet odchylky otáček
40 -   mozna_rychlost=(prevod>=0+tol).*(prevod<=300);
41 -   zarazena_rychlost=zeros(size(mozna_rychlost,1),1);
42 -   k=1;
43 -   r=1;
44 -   odchylka=zeros(size(mozna_rychlost,1),1);
45 -   while k <= (size(mozna_rychlost,1))
46 -       while true
47 -
48 -           if mozna_rychlost(k,r)==true
49 -               zarazena_rychlost(k)=r;
50 -               break
51 -           end
52 -           r=r+1;
53 -           if r > (size(mozna_rychlost,2))
54 -               zarazena_rychlost(k)=0;
55 -               %disp('Neutrál');
56 -               break
57 -           end
58 -
59 -       end
60 -
61 -       if zarazena_rychlost(k)==0
62 -           odchylka(k)=0;
63 -       else
64 -           odchylka(k)=matice_odchylek(k, zarazena_rychlost(k))-1;
65 -       end
66 -       r=1;
67 -       k=k+1;
68 -   end

```

Obr. 33: Vyhodnocení zařazené rychlosti a odchylky otáček

Obrázek 33 Pátá část kódu se zabývá tvorbou dvou matic. Matice s hodnotou zařazeného rychlostního stupně 1-6 anebo 0 pro neutrální. A maticí *odchylka*, která vyjadřuje procentuální odchylku otáček vstupní hřídele převodovky a motoru. Odrazíme se od matice *prevod*. Na základě ní vytvoříme matici *mozna.rychlost* která je pravdivostní maticí řídicí se podmínkou, že pokud je hodnota prvku matice *prevod* v intervalu $< (0 + tol); 300 >$, tak je hodnota prvku na stejné pozici v matici *mozna.rychlost* rovna 1 (pravda). Pokud je mimo interval, tak je hodnota rovna 0 (nepravda). V cyklu while vyhodnocujeme, jaký rychlostní stupeň je zařazen. Výsledný rychlostní stupeň je roven v pořadí prvnímu sloupci s hodnotou 1 (pravda). Pokud je ve všech sloupcích hodnota 0 (nepravda), znamená to, že je zařazený neutrální. Hodnoty o zařazeném stupni zapíšeme do matice *zarazena.rychlost*. Jako výslednou odchylku do matice *odchylka* zaznamenáme hodnotu z matice odchylek odpovídající danému zařazenému převodovému stupni.

```

69 - %% zanesení výsledků do grafu
70 - nexttile
71 - plot(t, zarazena_rychlost)
72 - legend('Zařazený rychlostní stupeň [-]')
73 - nexttile
74 - plot(t, odchylka.*10)
75 - legend('Odchylka otáček [%]')
76 - xlabel('t [s]')

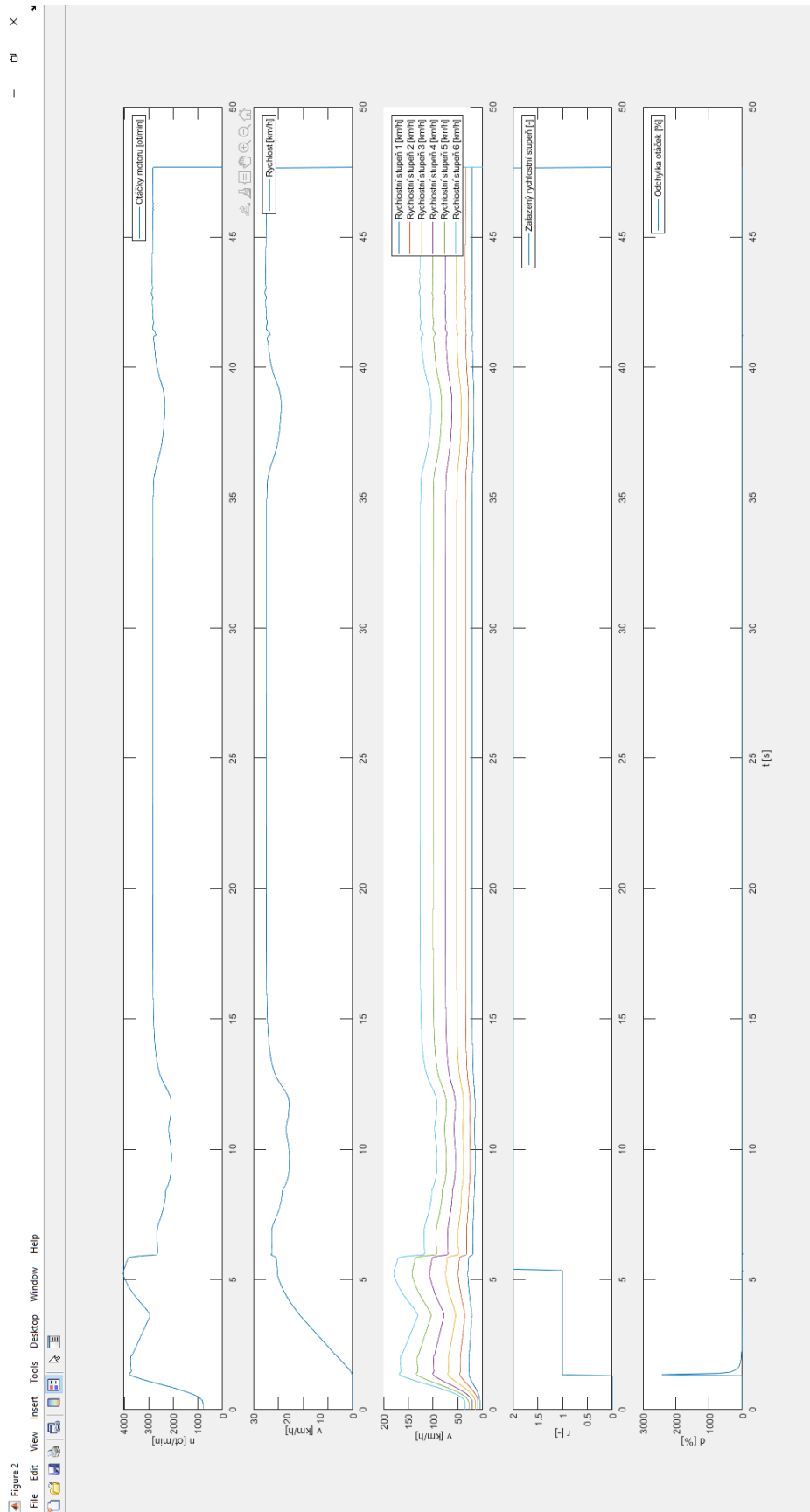
```

Obr. 34: Zanesení výsledků do grafů

Obrázek 34 V poslední části tvoříme poslední dva grafy. První z nich ukazuje závislost zařazeného rychlostního stupně na čase a druhý závislost odchylky otáček na čase.

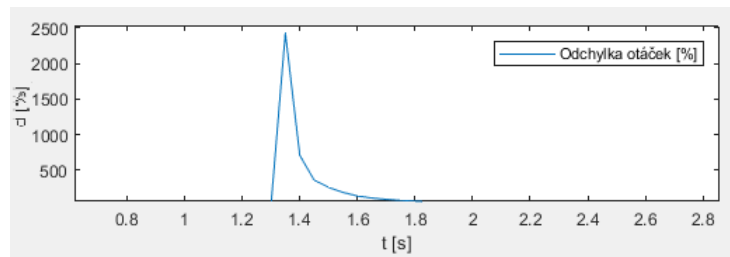
8.2 Výstup ze softwaru

Výstupem ze softwaru je okno (viz obr. 35), které obsahuje 5 grafů.



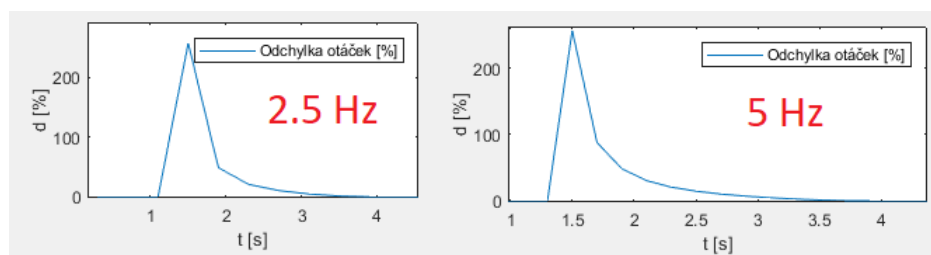
Obr. 35: Výstup z vyhodnocení zacházení se spojkou

Na prvním grafu jsou zobrazeny otáčky motoru v závislosti na čase a na druhém grafu je zobrazena rychlost vozidla opět v závislosti na čase. Třetí graf zobrazuje všech šest fiktivních rychlostí v závislosti na čase. Na čtvrtém grafu vidíme jednotlivé zařazené rychlostní stupně a to tak, jak je vyhodnotil software v závislosti na čase. Jako hlavní výstup považuji graf pátý, který je možné si prohlédnout detailněji na obr. 36. Graf vyjadřuje rozdíl otáček na setrvačnicku a otáček na vstupním hřídeli převodovky. To je údaj, který představuje vyjádření prokluzu spojky. Také zde můžeme navíc sledovat podobnost závislosti s obrázkem 28.



Obr. 36: Detail odchylnosti otáček motoru a vstupního hřídele převodovky

Na obr.37, ještě můžeme porovnat stejný graf jako na obr.36, ale s použitím nižší vzorkovací frekvence při získávání dat ze simulace. Toto je pro nás zajímavé z důvodu snížení nákladů na vyhodnocovací jednotku ve voze, která je limitovaná přenosovou rychlostí portu OBD2. Namísto původních 20 Hz ukazujeme vyhodnocení dat při snížení vzorkovací frekvence na 5 respektive 2.5 Hz. Při vzorkovací frekvenci 2.5 Hz už bohužel pozorujeme značnou ztrátu na kvalitě dat, která se například projevuje pomalejší reakcí na začátku situace a proto dosahuje nižší amplitudy, a proto se při použití jednotky budeme muset snažit o optimalizaci získání frekvence alespoň 5 Hz nebo ideálně ještě více. Z grafů by se mohlo zdát, že samotný rozjezd trvá déle při nižší vzorkovací frekvenci, což je ale způsobeno různým rozsahem hodnot na ose Y. Na všech grafech sledujeme identickou situaci, která ve všech případech trvá stejně dlouho.



Obr. 37: Porovnání vzorkovací frekvence

9 Závěr

V bakalářské práci jsem parametricky popisoval a sledoval možné jízdní extrémy, ke kterým dochází při jízdě carsharingových vozidel a mají vliv na opotřebení vozidla. Pomocí

simulačního prostředí IPG CarMaker 10.0 jsem vytvořil testovací situaci, která byla navržena na základě dat o skutečném úseku silnice. Který jsem získal pomocí software Google Earth a následně modifikoval přidáním kontrolovaného testovacího úseku s izolovanými překážkami, které vyvolávají jízdní extrémy. Výstupem z práce je software v prostředí Matlab, který přebírá naměřená data ze simulace a vyhodnocuje průjezd testovacího vozidla testovacím úsekem. Software se zaměřuje převážně na problematiku průjezdu vozidla po nerovnostech, kterými mohou být například výmoly nebo příčné prahy a následně problematikou zacházení řidiče se spojkou při jízdě ve voze s manuální převodovkou.

Zjistil jsem, že je možné sledovat a vyhodnocovat průjezdy skrze výmoly/prahy a zacházení se spojkou a to s použitím snímání dat z OBD2 portu v automobilech carsharingu doplněných o data z gyroskopu s akcelerometrem, kterými vozidla vozového parku společnosti Uniqway disponují. Data z portu OBD2 budou muset být snímány s frekvencí vyšší než 2.5 Hz, aby byly dostatečně detailní pro vyhodnocení dat o užívání. Bakalářská práce je připravena na pokračování v návrhu speciálního softwaru pro vyhodnocení vlastností řidičů přímo ve vozidlech Uniqway. Je jí tedy možné použít jako východisko pro získání povědomí o struktuře řidičů, kteří službu používají a na základě dat o nich odpovědět na otázky ohledně řešitelnosti problematiky opotřebení vozidel ve vozovém parku společnosti Uniqway. Vyhodnocení by následně mohlo vést k opatřením, které by opotřebení automobilů značně snížily. Vyhodnocovací software bude implementován do aplikace Uniqway, která se stará a vyhodnocení jízdy vozidel.

Seznam použitých značek a symbolů

OBD2 On-board diagnostics

GUI Graphical user interface

ID Identifikace, identita

ABS Anti-lock braking system

ESP Electronic Stability Programme

Obr. Obrázek

aj. a jiné

4WD Pohon všech čtyř kol

VAG Volkswagen koncern

Seznam použité literatury a zdrojů

- [1] IPG Automotive Group. *CarMaker User's Guide*. Karlsruhe, Německo: IPG Automotive Team, 2021. Dostupné v software IPG CarMaker 10.0
- [2] TP 85. Zpomalovací prahy. Technické normy. Stavební fakulta ČVUT v Praze - Katedra silničních staveb: MD ČR, září 1996, 40 str. s účinností od 1. října 1996
- [3] BATULKOVÁ, Anna. Car-sharing – Snižování počtu automobilů ve městech. Praha, 2008. 98 str. Diplomová práce. Fakulta dopravní, České vysoké učení technické v Praze, 2008.
- [4] Carsharing Uniqway, O nás [online]. ŠKODA AUTO DigiLab s.r.o., ©2022. [cit. 8.7.2022]. Dostupné z: <https://uniqway.cz/about>
- [5] BRITTON, E.: Carsharing 2000: Sustainable Transport's Missing Link [online]. EcoLogica Ltd., Lancaster, 2000. 317 s. [cit. 2017-04-17]. Dostupné z: <http://www.ecoplan.org/library/wt5-3.pdf>. ISSN 1352-7614.
- [6] Kolář, J. : Teoretické základy konstrukce kolejových vozidel , skriptum, 1. vyd., Nakladatelství ČVUT -Česká technika, 2009. , 276 s. , ISBN 978-80-01-04262-5
- [7] Dušek, F.: MATLAB a SIMULINK – úvod do používání. Univerzita Pardubice 2000, 147 s., ISBN 80-7194-273-1
- [8] VALACH, Ondrej. Analýzy opotrebenia vozidiel prevádzkovaných v službách carsharingu. Praha, 2021. 46 str. Bakalářská práce. Fakulta strojní, České vysoké učení technické v Praze, 2021.
- [9] BRICH, Jan. Možnosti predikce opotřebení vozidla provozovaného v systému car sharing Praha, 2018. 53 str. Bakalářská práce. Fakulta strojní, České vysoké učení technické v Praze, 2018.
- [10] RODE, Sudarshan S., et al. Pothole Detection and Warning System using Wireless Sensor Networks. Embed. Real-Time Syst. Lab. Indian Inst. Technol. Bombay, 2007.
- [11] CHAKRAPANI, Ganjikunta; SUGUMARAN, V. Health monitoring of automotive clutch system by using Bayes algorithms. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021. p. 012028.

Seznam obrázků, grafů a tabulek

Seznam obrázků

Obrázek 1	Opotřebená spojková lamela vozu Uniqway	7
Obrázek 2	Hlavní GUI IPG CarMaker 10.0	12
Obrázek 3	Konfigurace manévru	13
Obrázek 4	Konfigurace řidiče	14
Obrázek 5	Editor cesty	16
Obrázek 6	Menu konfigurace vozidla	17
Obrázek 7	Generátor vozidla	18
Obrázek 8	Nastavení pneumatik	19
Obrázek 9	IPGMovie	20
Obrázek 10	IPGControl	20
Obrázek 11	Export dat	21
Obrázek 12	Trasa v Google Maps	22
Obrázek 13	Trasa v IPG CarMakeru	22
Obrázek 14	g-g diagram	23
Obrázek 15	Hlavní rozcestí programu	25
Obrázek 16	Načítání dat do prostředí Matlab	25
Obrázek 17	Definice konstant	27
Obrázek 18	Graf rychlosti v závislosti na čase	27
Obrázek 19	Začátek grafu projetých výmolů	28
Obrázek 20	Hledání extrémů a úprava hodnot	28
Obrázek 21	Vyhledávání děr a prahů	29
Obrázek 22	Případy k zamyšlení	30
Obrázek 23	Výpis výsledků	31
Obrázek 24	Graf průjezdu 40 km/h	31
Obrázek 25	Graf průjezdu 40 km/h - DETAIL	32
Obrázek 26	Textový výstup průjezdu 40 km/h	33
Obrázek 27	Grafy při průjezdu rychlostí 30 km/h	33
Obrázek 28	Graf rozjezdu vozidla[6]	35
Obrázek 29	Přiřazení dat a definice konstant	36
Obrázek 30	Inicializace tvorby grafu	36
Obrázek 31	Výpočet fiktivních rychlostí	36
Obrázek 32	Tvorba matice odchylek rychlosti	37
Obrázek 33	Vyhodnocení zařazené rychlosti a odchylky otáček	37
Obrázek 34	Zanesení výsledků do grafů	38
Obrázek 35	Výstup z vyhodnocení zacházení se spojkou	39

Obrázek 36	Detail odchylky otáček motoru a vstupního hřídele převodovky	40
Obrázek 37	Porovnání vzorkovací frekvence	40

Seznam tabulek

Seznam použitého SW

- Texmaker
- MiKTeX (L^AT_EX)
- Overleaf
- Matlab
- IPG CarMaker 10.0
- Visual Studio Code
- Google Maps
- Google Earth

Seznam příloh

Příloha 1: CD s daty a softwarem