



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta strojní

Ústav mechaniky, biomechaniky a mechatroniky

Rekonstrukce povrchu vozovky určená z dat senzorů mobilního telefonu

Reconstruction of the road surface determined from mobile phone sensor data

Bakalářská práce

Studijní program: Teoretický základ strojního inženýrství

Studijní obor: bez oboru

Vedoucí práce: Ing. Zdeněk Neusser, Ph.D.

Alexander Kosenko

Prohlášení

Prohlašuji, že tuto předloženou bakalářskou práci “ Rekonstrukce povrchu vozovky určená z dat senzorů mobilního telefonu” jsem vypracoval samostatně pod vedením vedoucího bakalářské práce Ing. Zdenka Neussera, Ph.D. a použil jsem pouze uvedené v seznamu podklady.

V Praze dne 2017

.....

Poděkování

Chtěl bych poděkovat svému vedoucímu práce panu Ing. Zdenku Neusserovi, Ph.D. za odborné vedení této bakalářské práce, vstřícnost a cenné rady na každé etapě její zpracování, trpělivost a věnovaný čas. Dál bych rad poděkoval panu Alexeji Sharovovi za pomoc v realizaci aplikaci pro platformu iOS.

Anotační list

Jméno autora:	Alexander Kosenko
Název BP:	Rekonstrukce povrchu vozovky určená z dat senzorů mobilního telefonu
Anglický název:	Reconstruction of the road surface determined from mobile phone sensor data
Akademický rok:	2016/2017
Ústav/Odbor:	Mechaniky a mechatroniky
Vedoucí BP:	Ing. Zdeněk Neusser, Ph.D.
Bibliografické údaje:	Počet stran: 32 Počet obrázků: 21 Počet tabulek: 2 Počet příloh: 3
Klíčová slova:	senzory, mobilní telefony, povrch vozovky, čtvrt auto, zpracování signálů
Keywords:	sensors, mobile phones, road surface, quarter car, signal processing

Abstrakt:

Tato bakalařská práce se zabývá rekonstrukcí povrchu vozovky z dat určených senzory mobilních telefonů. Teoretická část je věnována senzorům mobilních telefonů, možnostem získání a zpracování dat ze senzorů s přehledem matematických modelů dynamiky vozidla.

Praktická část se skládá z realizace mobilní aplikace, následné úpravy dat v prostředí Python a MATLAB a ve vytvoření modelu pro rekonstrukci povrchu vozovky v prostředí MATLAB-Simulink.

Abstrakt:

This bachelors thesis deals with the road surface reconstruction determined from mobile phone sensor data. Theoretical part is focused on mobile phone sensors, possibilities of data acquisition, data processing and overview of vehicle's suspension mathematical models.

The practical part consists of mobile application implementation, signal data processing in Python and Matlab and reconstruction of the road surface model in Matlab-Simulink based on measured data.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kosenko** Jméno: **Alexander** Osobní číslo: **437107**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Teoretický základ strojního inženýrství**
Studijní obor: **bez oboru**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Rekonstrukce povrchu vozovky určená z dat senzorů mobilního telefonu

Název bakalářské práce anglicky:

Reconstruction of the road surface determined from mobile phone sensor data

Pokyny pro vypracování:

- seznámte se se senzory používanými v mobilních telefonech
- proveďte zpracování dat získaných z telefonu
- na základě zpracovaných dat proveďte rekonstrukci povrchu vozovky

Seznam doporučené literatury:

- Valášek M., Bauma V., Šika Z.: Mechanika B. Nakladatelství ČVUT, Praha 2006
- Stejskal V., Valášek M.: Kinematics and Dynamics of Machinery. Marcel Dekker, Inc., New York 1996
- <http://www.mathworks.com> (on-line dokumentace prostředí Matlab)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Zdeněk Neusser Ph.D., ústav mechaniky, biomechaniky a mechatroniky FS

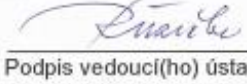
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:


Datum zadání bakalářské práce: **21.04.2017**

Termín odevzdání bakalářské práce: **18.08.2017**

Platnost zadání bakalářské práce:


Podpis vedoucí(ho) práce


Podpis vedoucí(ho) ústavu/katedry


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.

25.4.2017

Datum převzetí zadání


Podpis studenta

Obsah

Úvod.....	7
1. Sensory mobilních telefonů.....	8
1.1 Akcelerometr	9
1.2 Gyroskopy	10
1.3 Magnetometry	11
1.4 Vlastnosti MEMS senzorů	11
1.5 Vývoj MEMS senzoru	12
2 Získání a zpracování dat.....	13
2.1 Mobilní aplikace	13
2.2 Metody zpracování dat v mobilních aplikacích	15
2.2.1 Low-Pass	15
2.2.2 Sensor Fusion.....	15
2.2.3 Moving average	15
3. Stanovení povrchu vozovky	16
3.1 Systém odpružení vozidel	16
Druhy systému odpružení:.....	16
3.2 Modelování systému odpružení	17
3.2.1 Model čtvrt auta	17
3.2.2 Poloviční model	17
3.2.3 Model celého auta.....	18
4. Praktická část	18
4.1 Tvorba vlastní aplikace pro iOS.....	18
4.2 Python	20
4.3 Určení polohy smartphonu ve vozidle.....	21
4.4 Tvorba modelu pro rekonstrukci povrchu vozovky	21
5. Matlab a Simulink.....	23
6. Závěr	27
7. Seznam použité literatury	28
Seznam příloh	31

Úvod

Mobilní telefony jsou již dlouho pevnou součástí života v naší společnosti. Kromě zábavy a pomoci s organizací nesou i nové možnosti ke studiu a poznání světa kolem nás.

Výkon ve výpočtech je porovnatelný s počítači, které byli technicky vhodné pro měření před pár roky, toto a množství senzorů otevírá cestu k provedení různých výzkumů, k novým zážitkům a vše je dostupné jedním kliknutím kdekoli.

Motivací soustředění právě na rekonstrukci povrchu vozovky pro mě byla skutečnost, že realizace projektu by mohla být užitečná, protože dobrý stav vozovky je jedním z kritérií bezpečnosti silničního provozu.

Cíle práce:

- 1) Seznámení se senzory používanými v mobilních telefonech
- 2) Provedte zpracování dat získaných z telefonu
- 3) Na základě zpracovaných dat proveďte rekonstrukci povrchu vozovky

Předpokládáme známou a konkrétní polohu telefonu. Poloha telefonu v autě je předmětem zkoumání jiné práce.

V práci se uvažuje předpokladu, že při měření máme jenom jeden mobilní telefon, že nejsou k dispozici další senzory a předpokládáme známou a konkrétní polohu telefonu.

Částí práce je přehled různých modelů mobilních telefonů. Vůči široké nabídce, zaměření bylo stanoveno spíše na rozdělení podle mobilních operačních systémů, na kterých pracují. Konkrétně na Android a iOS, protože v současné době mají největší tržní podíl.

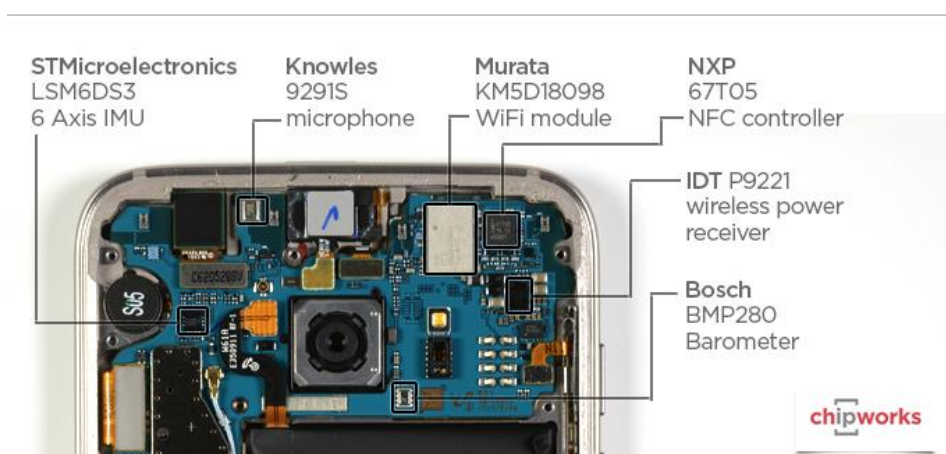
1. Senzory mobilních telefonů

Postupně zmenšení vnitřních elektronických součástí mobilu a zároveň zvětšení obrazovky a zánik doby mechanických klávesnic uvolnili prostor pro umístění dalších doplňků a *senzorů*.

Senzorem se rozumí součástka měřící nějakou fyzikální nebo veličinu. Senzory objevující v mobilních telefonech patří ke skupině *MEMS* sensorů.

Moderní mobilní telefony, které mají senzory a umožňují k nim přístup jsou náročnější mobilní zařízení, která využívá *mobilní operační systémy* a mají další název *smartphone*.

MEMS (z angl. Micro-Electro-Mechanical Systems) jsou zařízení, které sloučí mechanické a elektronické složky ve společném provedení mikročipu. MEMS zařízení může mít velikost řádových jednotek od mikrometru až do několika milimetrů. Kromě toho mají nízkou cenu, nízkou spotřebu a široké možnosti aplikace [1].



Obrázek 1 - Vnitřní provedení Galaxy S7 EDGE [2].

Existující metody s použitím mobilních zařízení zaměřené na stanovení povrchu vozovky se dá přiřadit mezi 2 skupiny.

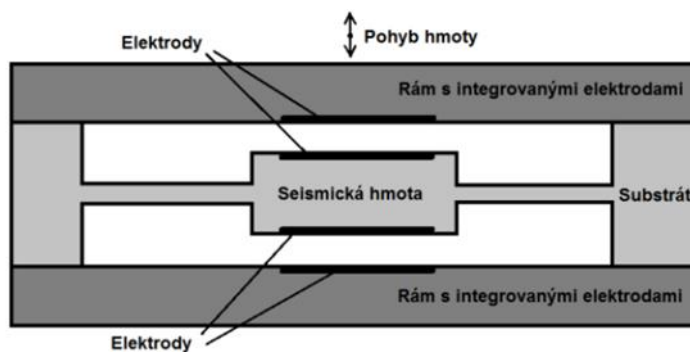
První je pořízení videozáznamu za jízdy s jeho následujícím zpracováním užitím různých algoritmu a programu. V této práci se tím nezabýváme, jelikož se vyžaduje hlubších znalostí z oboru počítačového programování a zpracování digitálních dat.

Druhá varianta se spočívá v detekci nerovnosti pomocí simulaci chování systému se známými mechanickými vlastnosti. Tuto metodou v práci se snažíme realizovat. Zde potřebujeme data ze senzorů, pomocí kterých se dokážeme určit polohu mobilu v prostoru. K nim patří akcelerometr, gyroskop magnetometr a modul GPS.

Modul GPS není senzorem ale přijímačem, nelze ho přímo použít k detekci nerovností vozovky, ale je nezbytným doplňkem pro praktickou cennost metody, jelikož může sloužit k stanovení pozice nerovnosti v reálném světě.

1.1 Akcelerometr

Akcelerometry mají několik druhů podle principu jejich fungování. V smartphonech se většinou používá kapacitní akcelerometry. Princip fungování v jedné ose je znázorněn na *Obrázku číslo 2*.



Obrázek 2 - Jednoosý kapacitní akcelerometr [3]

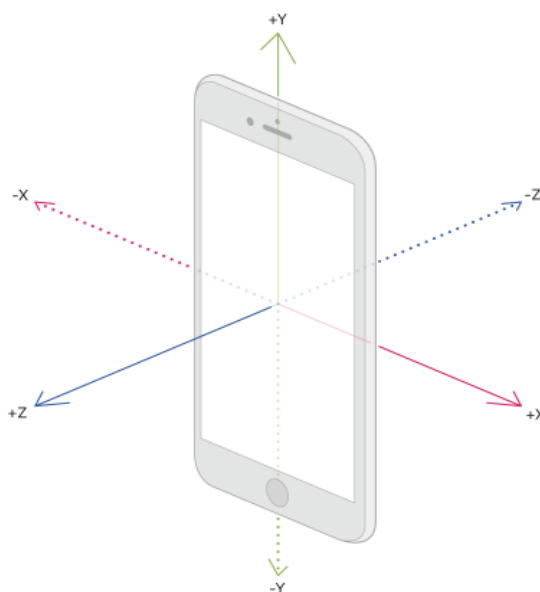
Při působení zrychlení na zařízení dojde k vychýlení seismické hmoty z rovnovážného stavu a ke změně kapacity mezi elektrodami integrovaného obvodu dle vzorečku (1) [1].

$$C = \frac{\epsilon_0 \cdot \epsilon_r \cdot S}{d} \quad (1)$$

Kde C je kapacita, ϵ_0 je permitivita ve vakuu, ϵ_r je relativní permitivita prostředí, S plocha elektrod, d vzdálenost mezi elektrody.

Elektronická část senzorů pozoruje tuto změnu a následně převede ji na hodnotu zrychlení v jednotkách tíhového zrychlení g .

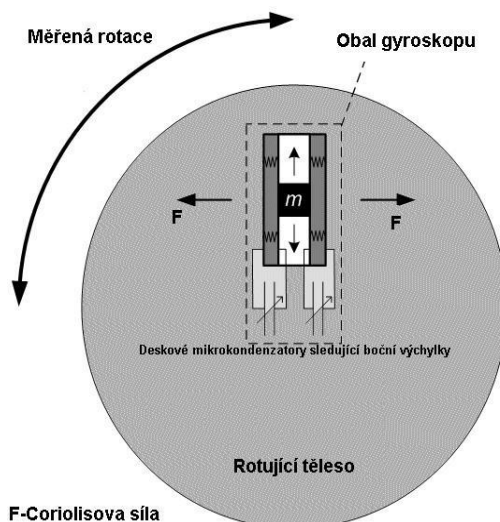
Akcelerometry v smartphonech určují směr vektorů zrychlení, používá se to pro stanovení, zda obrazovka mobilu je do země, nebo naopak, toto se používá pro automatické vypnutí, aplikací, a hlavně v hrách.



Obrázek 3 - Orientace tříosého akcelerometru a její smysly [4]

1.2 Gyroskopy

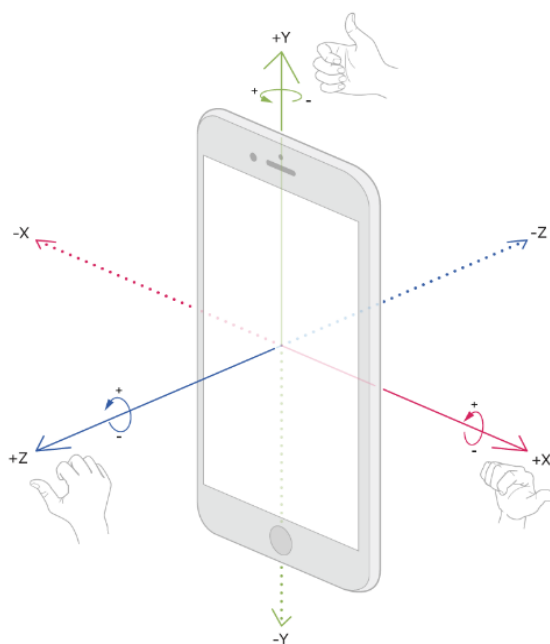
MEMS gyroskopy fungují na podobném principu jako vibrační gyroskopy, nemají žádnou rotující část a k měření úhlové rychlosti využívají Coriolisova efektu. [5]



Obrázek 4 - Orientace tříosého akcelerometru a její smysly [6]

Coriolisův efekt je jev dle vzorce (1.2), při kterém v rotujícím tělese se pohybuje těleso posuvným pohybem, Coriolisova síla F_c působí ve směru vektorového součinu obou rychlostí, její účinek je měřen a z něj je určena velikost rychlosti rotace-úhlové rychlosti ω , známe-li rychlost v posuvného pohybu tělesa, které se pohybuje na rotujícím tělese.

$$F_c = 2 \cdot m v \times \omega \quad (2)$$



Obrázek 5 - Orientace tříosého gyroskopu a její smysly [7]

1.3 Magnetometry

Magnetometr zaznamenává veškerá magnetická pole, která působí na osy telefonu. Je tvořen drátkem s procházejícím elektrickým proudem, který se vychýlí v závislosti na působení magnetického pole. Pomocí tohoto senzoru lze měřit směr magnetického severu, který se ale mírně liší od fyzického severu také využívají mapy pro určení. Výhodou magnetometru je jeho energetická nenáročnost oproti jiným sensorům, naopak nevýhodou je nepříliš velká přesnost a ovlivnitelnost vnějším rušením, například od jiných magnetu nebo elektromagnetických polí jiných zařízení jakým je například palubní počítač auta. V smartphonech používá jako elektronický kompas.

1.4 Vlastnosti MEMS senzorů

MEMS senzory dosáhli tak velkého rozsahu použití vůči svým rozměru v a nízké ceně. Jaký mají ostatní vlastnosti? To je velme složitá otázka, kterou nemá jednoznačnou odpověď, ovlivňují ji široká kombinace faktorů, které nelze jednoznačně vyznačit. Vybral jsem jen té, které mají cenu pro naši měření.

Measurement range (rozsah měření) – interval hodnot měřené veličiny který je schopen senzor zaznamenat, většinou tento parametr je nastavitelný.

Output data rate (vzorkovací frekvence) – počet vzorku za jednotku času, má přímý vliv na použitelnost dat z měření.

Sensitivity (citlivost) – změna výstupu v závislosti na změně vstupu.

Bias (offset)

V statické rovnovážné poloze senzor by měl ukázat nulovou hodnotu, v praxi tomu tak není, offsetem se nazývá odchylka naměřené hodnoty od reálné, je možné ji eliminovat, avšak její hodnota se může měnit:

- při každém zapnutí senzorů
- vlivem změny teploty
- vlivem změny napájecího napětí

Drift – tím se rozumí chybě vzniklé v následku náhodných změn hodnoty offsetu v průběhu měření a taky existenci šum.

Noise-Density (Šumová hustota) – efektivní hodnota šumu ve frekvenční oblasti signálu – čím menší tím lepší.

Noise (Šum) – náhodná hodnota znečišťující měření, u MEMS senzorů je to bílý Gaussův šum [1].

Temperature offset drift (teplotní změna offsetu) - změna hodnoty offsetu v závislosti na změně teploty.

Odezva ze spaní – čas který projde od obracení na senzor do jeho odezvy.

Drift je problémem u gyroskopu [5]. Například dle [8] chyba, během měření 3 minut v klidně poloze chyba se rovnala 4°. U akcelerometru větší problém dělá **offset**, ale ho lze snadno eliminovat kalibrací. [9]

1.5 Vývoj MEMS senzoru

Bylo celkem problematické najít nějaké informace ohledně konkrétních modelu senzorů z důvodu konkurence výrobců, ale dle [2], [10], [11]. Vybral jsem modely *Apple iPhone 5S* (podzim 2013), *Samsung Galaxy S7* (Járo 2016) a *Apple iPhone 6 SE* (podzim 2015), v závorkách je uveden čas objevu na trhu. Jelikož se mi nepodařilo zjistit modely magnetometrů, vynechal jsem jich v porovnání [12], [13], [14], [15] .

1) Záleží na vybraném nastavení

2) Výrobce uvádí 2g/4g pro akcelerometr a 250dps /500dps pro gyroskop jako nejlepší parametry k určení polohy

Vlastnost	Jednotky	iPhone 5S	S Galaxy S7	iPhone 6 SE
Measurement range	g	±2/±4/±8/±16	±2/±4/±8/±16	±2/±4/±8/±16
Output data rate	Hz	32-1000(1)	12,5/6664(1)	4/8000(1)
Sensitivity	mg/LSB	0,0625/0,125(2)	0,061/0,122(2)	0,0625/0,125(2)
Zero-Offset	mg	±95	±40	±60
Noise-Density	μg/√Hz	200	90/90	300
Noise	mg(RMS)	2	1,7/2,0	-
Temperature offset drift	mg/ °C	±2	±0,5	±1

Tabulka 1 - Porovnání vlastnosti akcelerometrů

Vlastnost	Jednotky	iPhone 5S	S Galaxy S7	iPhone 6 SE
Range Scale	°/s	±250/±500/±1000/±20	±125/±250/±500/±1000/	±250/±500/±1000/±2000
Output data rate	Hz	95/760(1)	12,5/6664(1)	4/8000(1)
Sensitivity	LSB(°/s)	0,1142/0,22(2)	0,061/0,122(2)	0,0625/0,125(2)
Zero-Offset	(°/s)	±95	±40	±60
Noise-Density	(°/s)/√Hz	0,03	0,07	0,01
Noise	(°/s) (RMS)	-	1,7/2,0	0,1
Temperature drift	°/s/°C	±0,03	±0,05	±0,24

Tabulka 2 - Porovnání vlastnosti gyroskopů

Dle zdrojů je vidět že MEMS senzory se dostanou do špičkových modelu smartphonu i po letech počátku výroby. Takže že hodnocení bylo o něco komplikovanější. Dle porovnání vlastnosti z Tabulek 1 a 2 je vidět, že v podstatě se zvětšila jenom vzorkovací frekvenci.

2 Získání a zpracování dat

2.1 Mobilní aplikace

Výhodou smartphonu je snadná možnost získání dat bez použití dalších zařízení, potřebujeme jenom vybrat a nainstalovat mobilní aplikaci. Seznámil jsem se s nabídkou aplikací pro *iOS* a vyzkoušel jsem kolem 10 produktu. S aplikacemi pro android jsem se seznámil ze specifikace vývojářů bylo zjištěno:

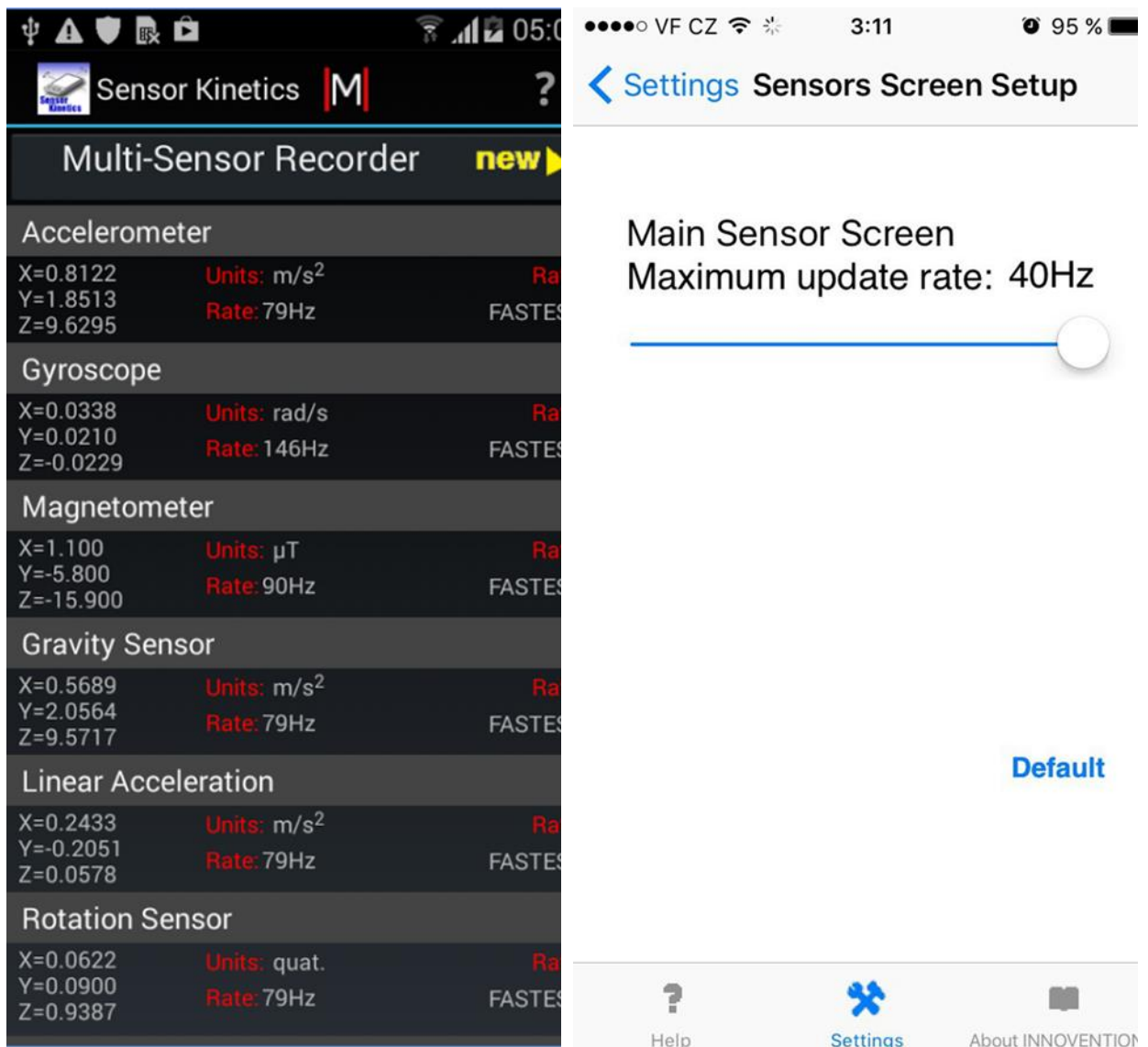
- produkty pro oba operační systémy poskytují *raw* neboli čistá data
- pro *iOS* neexistují žádné bezplatné produkty s možností importu dat,
- nedostatek informace o nastavení měření.

Na konec jsem vybral dvě vhodné aplikace, které jsou realizovány pro obě platformy, a to jsou

MATLAB Mobile a ***SensorKinetics***

Mobilní aplikace od *MATLAB* umožňuje jak sledování měření v reálném čase, tak i jejich uložení. Výhody jsou možnost zpracování dat okamžitě po měření, a existence verzí pro *iOS* a *Android*. Nevýhodou je skutečnost že studentská licence neumožní uložení záznamů, mezi další patří vliv výpočtů na kvalitu vzorkování, protože výkon mobilního zařízení bude obtíženo současně jak výpočty, tak i nahráním záznamů.

SensorKinetics je aplikace která dle jejích informací byla vyvíjena společností, jejíž zakladatelem je doktor zabývající právě problematikou senzorů. který se zabývá MEMS senzory. Na obrázcích 6 a 7 jsou snímky obrazovek z mého telefonu a z oficiálních stránek vývojáře. [16]



Obrázky 6 a 7 - Snímky z aplikace SensorKinetics pro Android (vpravo) [16] pro iOS (vlevo)

Po instalaci bylo zjištěno že maximální měřicí frekvence pro iOS je 100 Hz, i když maximální hodnota je 1000 Hz, což vyrazně omezuje možnosti použití mobilních telefonů pro veškerá měření, a dále že nejspíš je o software omezení od výrobce mobilu.

Další problém je nižší vzorkovací frekvence při současném měření z několika senzorů. Naopak mobilní telefony s operačním systémem Android nemá software omezení na vzorkovací frekvenci. [5]

2.2 Metody zpracování dat v mobilních aplikaci

Veškerá informace od vývojářů aplikaci a diskuzi na jejich formule, jak ohledně použitých metod filtraci byla, že buď vůbec není anebo je z těchto [17].

2.2.1 Low-Pass

Název může Vás, nejde tady o filtr typu *Dolní propust*, jelikož této 2 metody mají úplně různé principy. Princip této je založen na úpravě každé následující hodnotě měření s ohledem na hodnotu předcházející. Toto je označeno jako faktor filtrace a dle vzorečků.

$$a_i = f a_{i-1} + (1 - f) a_i \quad (3)$$

Kde a_i je vstupní hodnota a_{i-1} je předcházející hodnota, f je stupeň filtrace. Tato metoda je použitelná pro jednodušší aplikaci, avšak pro naši není vhodná.

2.2.2 Sensor Fusion

Fúze senzorů, tato metoda je buď kombinací dat z několika senzorů nebo to samé doplněné o informacemi od uživatele.

Příkladem takové implementaci je sloučení dat akcelerometru a magnetometrů jako jeden ze vstupů, druhým jsou data z akcelerometru. Užitím algoritmu známý jako Kálmánův filtr se eliminuje vliv driftu gyroskopu.

A to opravdu platí, tuto metodou se řídí i při navigaci kosmických letounů a letadel. U mobilních telefonů ale ne, data se zničí při objevu dalšího zrychlení, například odstředivého v zatáčce, Kálmánův filtr nemůže odlišit je od gravitačního zrychlení. [5]

2.2.3 Moving average

Klouzavý průměr – počítá se jako průměr hodnot stejného počtu za sebou jdoucích období dle vzorečku (4).

$$a_s = \frac{a_M + a_{M-1} + \dots + a_{M-(n-1)}}{n} = \frac{1}{n} \sum_{i=0}^{n-1} a_{M-i} \quad (4)$$

3. Stanovení povrchu vozovky

Překonání překážky tedy nerovnosti vozidla vede ke vzniku kmitavých pohybů přenášených do karoserie. Aby se zmenšit jejich negativní vliv existují různé systémy odpružení vozidel. Stanovení povrchu je možné dle sledování a následně zpracovávání kmitu, přinášené od vozovky do karoserie.

3.1 Systém odpružení vozidel

Systém odpružení je systém, který má za úkol zajistit bezpečnost jízdy, komfort cestujících a stabilní chování vozidla, a to tím že absorbuje kmitavé pohyby od náprav na karoserii a podvozek. Kromě toho odpružení výrazně zvyšuje životnost jednotlivých dílů. Ideální realizace toho systému vedla by plné absorpci kmitu a žádným kmitům karoserie vozidla bez ohledu na nerovnosti vozovky. [18]

Důležité pojmy:

Odpružené hmoty (z angl. Sprung mass) - karoserie, část náprav, a zavěšení.

Neodpružené hmoty (z angl. Unsprung mass) - kola, pneumatiky, část zavěšení.

Druhy systému odpružení:

- **Pasivní systémy**

Tento druh odpružení je nejstarší, skládá se ze tlumiče a pružiny. Hlavní výhody jsou jejich jednoduchá konstrukce, nízká cena, spolehlivost, a nízká energetická spotřeba pružiny se dá změnit v úzkém rozsahu a malá způsobí. Nastavení tlumiče oproti tomu má širší rozsah, avšak stále musí být kompromisní mezi bezpečností a komfortem cestujících. Pro obě varianty platí že se dá změnit jen před jízdou. [19], [18].

- **Semi-aktivní systémy**

Oproti pasivnímu systému tlumič má elektronickou řídicí jednotku, která mění nastavení tlumiče dle aktuální situační anebo manuální přepnutí řidičem. Výsledná bezpečnost je vyšší než u pasivního, přičemž je zachována relativně nízká cena. [19], [18], [20].

- **Aktivní systémy**

V těchto systémech tlumič a pružina buď úplně anebo částečně nahrazené zdrojem síly, ke překonání. Existuje různá varianty provedení, mezi ni patří: pneumatické systémy, hydraulické systémy, lineární elektromotory a další.

Jejich schopnost udržet kolo ve kontaktu s vozovkou a dosáhnout skoro nekmitající karoserii doplněna o skoro okamžitou reakci na kritické situace která je realizovatelná pro každou nápravu zvláště je neporovnatelně vyšší než u ostatních systému.

Na druhou stranu vyžadují složitý řídicí systém se spousta senzory a velkou energetickou spotřebu.

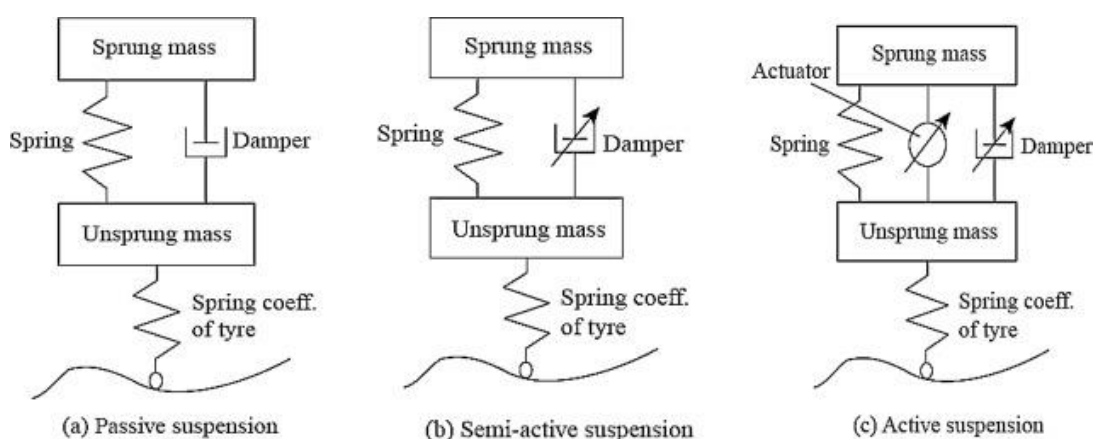
Dal jejich cena směřuje k volbě jen u dražších vozu. Složitost oprav, taky nepřispívá pro širší rozsah na trhu. [19], [18], [20]

3.2 Modelování systému odpružení

Dynamické chování vozidla je složitou problematikou, modely pro její jsou vždy kompromisem mezi náročností systému a spravedlivostí jeho popisu.

3.2.1 Model čtvrt auta

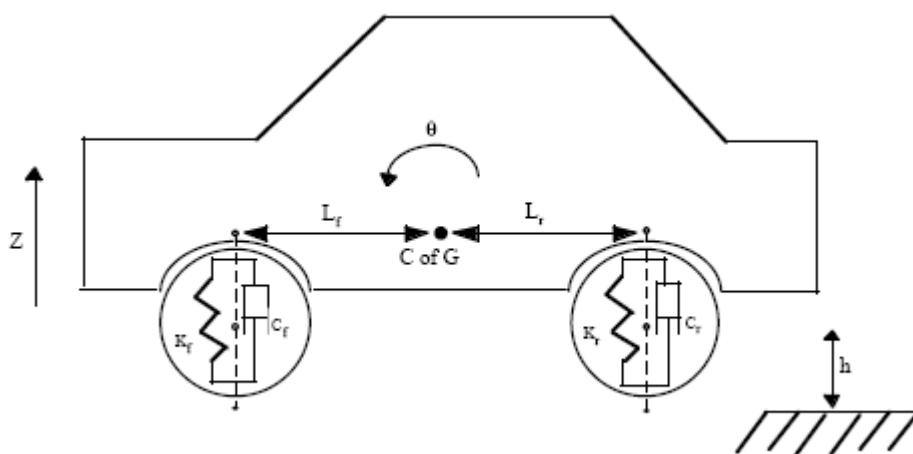
Je základním řešením pro modelování systému odpružení, pomocí kterého lze navrhnout lineární a nelineární verze modelu pasivního systému, které lze snadno upravit na model Semi-aktivního systému nebo na Aktivní variantu podle potřeb zadání, jak je to znázorněno na *Obrázku 8* [21].



Obrázek 8 - Modely čtvrt auta pro systémy odpružení a) pasivní b) semí-aktivní c) aktivní [20]

Výhodou je snadná realizace, která stačí pro modelování svislých pohybu vozidla. Nevýhodou je to že lze použít jen pro svislý směr pohybu. Taky je možné zjednodušit na jedno hmotovou verzi.

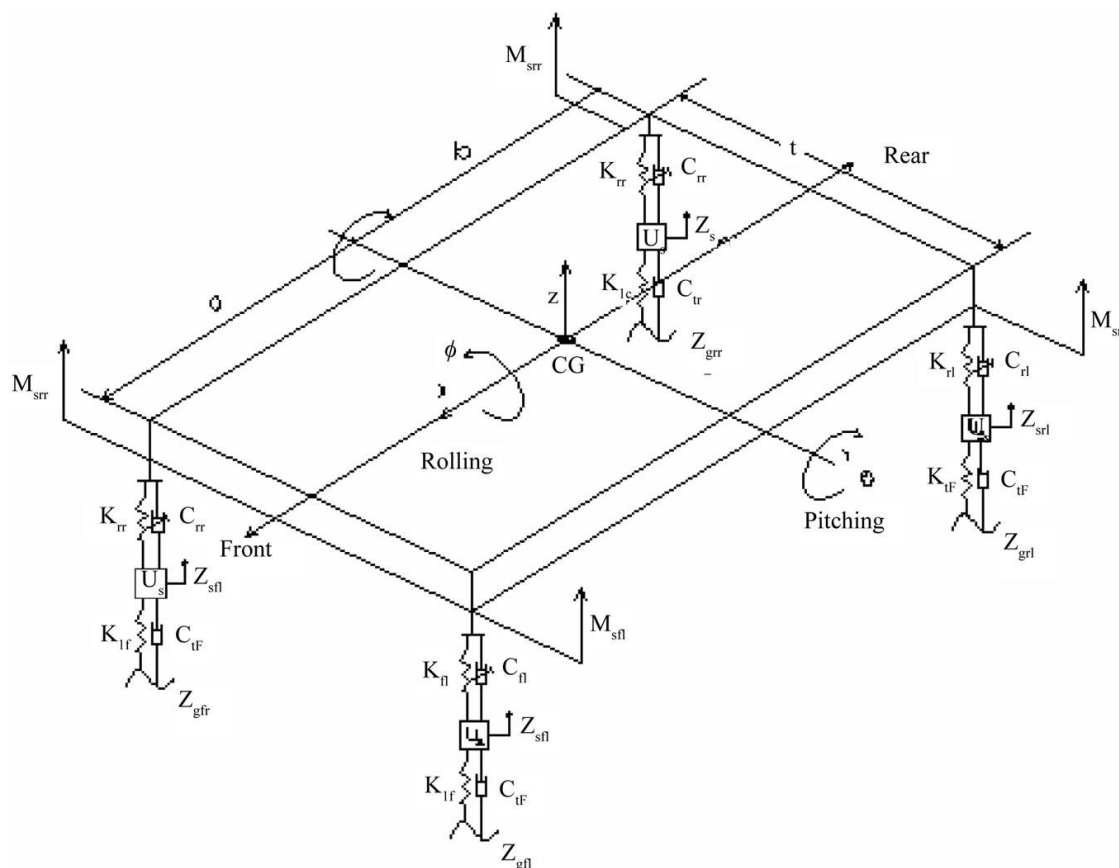
3.2.2 Poloviční model



Obrázek 9 - Poloviční model auta [22]

Při současné realizace dvou modelu čtvrt auta získáme poloviční model auta. Výsledky jsou bližší k reálnému chování, tím že se dá sledovat chování každé nápravy zvlášť
 Kromě svislého pohybu tady se zkoumá natočení kolem příčné osy anglicky pojem je **pitch**.
 K tomu je ale potřeba dimenzování geometrických vztahu mezi nápravami, a známa poloha těžiště.
 [22]

3.2.3 Model celého auta



Obrázek 10 - Model celého auta [21]

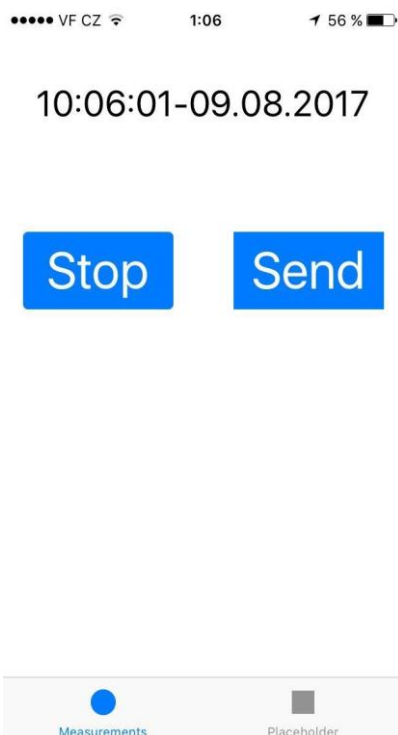
Tento model lze odvodit obdobně jako poloviční, tady volbou 2 polovičních modelu. Umožní sledování dynamického chování vozidla jak v svisle ose, tak i ve vodorovné rovině v soustavě úhlů *roll, pitch, yaw*.

4. Praktická část

4.1 Tvorba vlastní aplikace pro iOS

Z důvodu nespokojenosti s aplikacemi dostupnými pro *iOS*, bylo rozhodnuto o pokus tvorby vlastní aplikaci. Podmínkou pro tvorbu aplikaci pro *iOS* je počítač s operačním systémem *MacOS* na instalovány jen v počítačích od *Apple*. Proto jsem se obrátil jsem se na svého kamaráda, Alexie Sharova, který kromě přístupu ke svému počítači taky se výrazně podělil na vzniku této aplikaci.

Základní myšlenkou, kterou jsme se snažili při tvorbě aplikaci sledovat je co nejjednodušší program, aby co nejmíň ovlivnit výsledky a vytvořit jednoduché uživatelské rozhraní.



Obrázek 11 - Snímek z aplikace SensorRecord

Důležité pojmy:

- **IDE** (*Integrated Development Environment*) je vývojové prostředí je software slouží odlehčení práci programátorů, většinou je zaměřen na jeden programovací jazyk, má sadu nástrojů, editor kódu, kompilátor, debugger, může taky obsahovat nástroje pro tvorbu grafického uživatelského rozhraní
- **SDK** (*Software development kit*) je sada vývojových nástrojů pro tvorbu aplikaci pro určitou platformou kterou obvykle získáte od poskytovateli této platformy.
- **Framework** – tvořící základní strukturu aplikaci, obsahuje různé funkci, programy knihovny a nápovědy, výrazně usnadňující proces tvorby aplikace.
- **Databáze** – strukturovaný systém slouží k uložení souboru z možnosti přístupu a zpracování.

Jako vývojové prostředí byl vybrán *xCode* od společnosti *Apple Inc.*

Při tvorbě aplikaci byli použité frameworky *CoreFoundation*, *CoreMotion* a *CoreLocation* které jsou části *iOS SDK* od společnosti *Apple.Inc*

CoreFoundation je bázi budoucí aplikaci, umožňuje řízení základních funkcí operačního systému, zapisování dat, zpracovávání textu, provedení výpočtu časování, dělení a síťový přístup. *CoreMotion* je framework pro přístup k akcelerometru, gyroskopu, magnetometrů, barometru a algoritmu *DeviceMotion*, dle je uvedené že pracují na stejném principu, proto nebyl vybrán.

Poslední framework slouží k získání dat z GPS.

Vybrali jsme metodu *pull*, která vyžadují víc výpočtových výkonu, výsledné má menší vzorkovací frekvenci, ale nedochází k žádné ztrátě dat.

Při zkouškách se objevil problém: při současném měření z několika senzorů reálná frekvence měření byla mnohem menší než 100 Hz. Různé kombinace zapnutých senzorů ukázaly že jedinou variantou je GPS a akcelerometr.

```
func startMeasurement() {
    let sessionName = generateSessionName()
    setSessionLabel(sessionName: sessionName)
    showSessionLabel()
    recorder = SensorDataRecorder(
        sessionName: sessionName,
        gyrUpdateInterval: 0.01,
        accUpdateInterval: 0.01)
    recorder?.start()
}
```

Obrázek 12 – snímek části kódu aplikace *SensorRecord*

Je nutné zdůraznit že každá aktivita kromě vlastní práce se senzory ovlivňovala vzorkovací frekvenci, toho se lze částečně vyvarovat optimalizací, která však která vyžaduje větších zkušeností.

Důležitý je stručný popis toho, jak vlastně smartphone pracují z data:

- Časová měření se zapisuje v μs
- Jeden takt procesoru tvoří jednu operaci s jedním ze senzorů
- nutné zdůraznit že každá aktivita kromě vlastní práce se senzory ovlivňovala vzorkovací frekvenci, tomu se lze vyhnout optimalizací která vyžaduje větších zkušenosti

4.2 Python

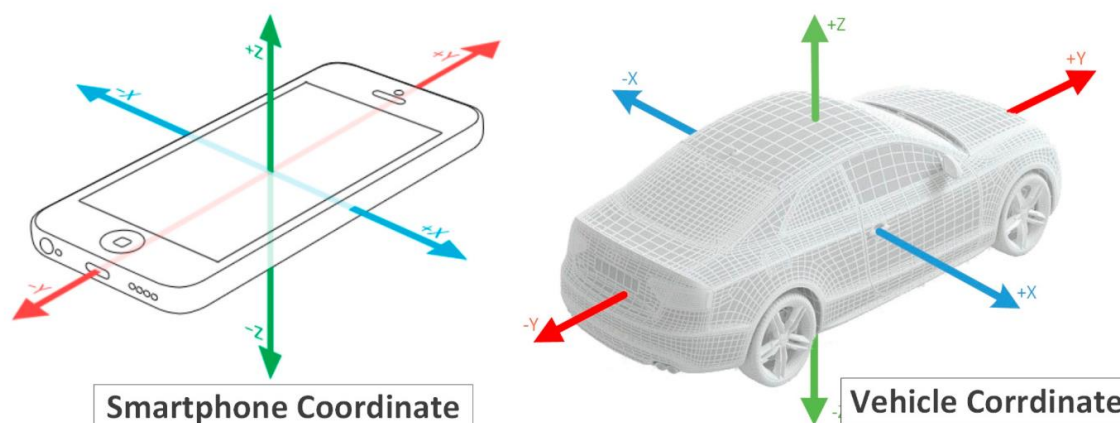
Skript v *Python* je určen pro předzpracování dat po aplikaci před vstupem do *Matlab*, jehož účelem je nahrání *csv* souboru s tím, aby dál převést čas měření v reálný čas. Dále musí dle původních verzi, rozdělit data jednotlivých senzorů do jednotlivých oblastí pro jednodušší další zpracování. Výsledkem je dva soubory *acc_prepared.csv* a *gps_prepared.csv*

4.3 Určení polohy smartphonu ve vozidle

Pro zvolený simulační model nejlepší umístění bude takové při kterémž jedna z os akcelerometru bude co nejmíň odlišná od gravitační osy zrychlení.

Dál by měla být co nejbliž k ose kol, a protože máme pouze jeden senzor, volím umístění někde uprostřed interiéru vozidla. Pro každé vozidlo to bude jinak, důležité je že stačí naznačit která z os akcelerometru pro daný případ bude svislá.

Pro vůz, na kterém se budou provádět měření jsem zvolil. Na níž jsem polepil pouzdro mobilu pomocí lepicí folie 3M, která se používá pro uchycení outdoorových kamer v sportovních aktivitách. Na obrázku je znázorněno umístění smartphonu ve vozidle, při kterém byli provedena měření.



Obrázek 13 - Souřadnicový systém mobilu ve vozidle [23]

Aby se eliminovat gravitační zrychlení je třeba užít vzorečků (5).

$$a_{\zeta} = a_{\zeta v} \cos(\beta) \tag{5}$$

a_{ζ} je kalibrovaná hodnota, $a_{\zeta v}$ je hodnota před kalibrací, β je úhel odchyly od svisle osy. Při našich měřeních úhel se rovna 8° , tato hodnota byla určena ze senzoru smartphonu.

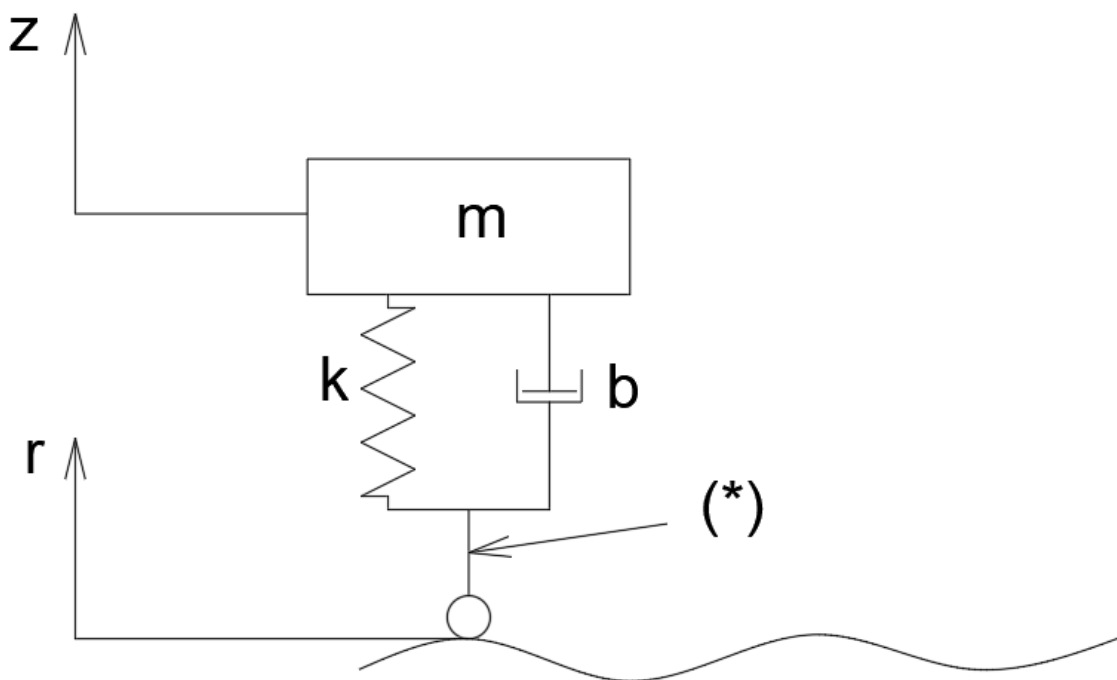
4.4 Tvorba modelu pro rekonstrukci povrchu vozovky

Ukázalo se, že pro náš účel je vhodně použít jen hodnoty z akcelerometru, a to z důvodu omezených zdrojů strojního času. Měření se provádělo dle předchozí (), zdrojem vzorků je zejména hodnoty zrychlení mobilního telefonu umístěného ve vozidle v dané poloze.

Měření se prováděla při konstantní rychlosti 30 km/h, doba měření 10 minut.

Po několika měřeních v okamžiku 150 sekund se měření zastavilo, důvodem k tomu nejspíš může být buď limita paměti mobilu nebo konflikt v aplikaci. Jelikož měření nebylo mou pozastavené, avšak na konci na e-mail jsem dostal dva soubory z měření.

Pro rekonstrukci povrchu vozovky byl vybrán jedno hmotový model čtvrt auta, jelikož se nemam k dispozici hodnoty zrychlení neodpuzené hmoty. Hodnoty a koeficienty používám ze [24]. [25]



Obrázek 14- Jednohmotový model čtvrtauta

Význam jednotlivých symbolů je následující:

- z poloha mobilního telefonu ve vozidle[m]
- r poloha vozovky[m]
- m hmotnost odpružené části automobilu[kg]
- k konstanta tuhosti pružiny[N/m]
- b konstanta tlumení tlumiče[Ns/m]
- (*) zanedbána ekvivalentní tuhost pro nahrazení pneumatiky [N/m]

Diferenciální rovnice druhého řádu popisující tento model je:

$$m\ddot{z} + b\dot{z} + kz = b\dot{r} + kr \quad (6)$$

Kde

- \ddot{z} zrychlení mobilního telefonu ve vozidle[m/s²]
- \dot{z} rychlost mobilního telefonu ve vozidle[m/s]
- \dot{r} rychlost vozovky[m/s]

Levou stranu rovnici (6) můžeme řešit postupné integraci:

$$m\ddot{z} + b\dot{z} + kz \quad (7)$$

Pravou stranou rovnici (6) tady převedeme na obrazovou rovnici povrchu vozovky:

$$Rhs = R(s)(bs + k) \quad (8)$$

Výsledná rovnice pro určení povrchu vozovky je součtem rovnic (7) a (8):

$$r(t) = (m\ddot{z} + b\dot{z} + kz) \frac{1}{bs + k} \quad (9)$$

5. Matlab a Simulink

Zpracování dat jsem provedl v prostředí **MATLAB** s doplňkem **Signal processing toolbox** a následně nadstavbě **SIMULINK**.

Skript **import1.m** nahrává data ze souboru, s přidělením do příslušných oblasti pro další zpracování. Dál se stanoví hodnoty koeficientu, převzaté od [24]. Vstupem do prostředí MATLAB je soubor **acc_prepared.csv** ve kterém jsou hodnoty zrychlení v ose **z**, zahrnuje zatím i tíhové zrychlení.

Hodnoty koeficientů převzaté od [24].

$$\begin{aligned} m &= 600[\text{kg}] \\ k &= 9600[\text{N/m}] \\ b &= 700[\text{Ns/m}] \end{aligned}$$

Skript **process.m** slouží k filtraci signálu tedy průběhu zrychlení v čase, s filtrací a komunikací s modely Simulink.

Lineární interpolace byla použita k tomu abychom získali konstantní časový krok, což je nezbytné nutné pro další zpracování signálu.

Pro filtraci signálu použijeme **nekausální** filtry typu **Dolní propust** a **Horní propust**. Proto použijeme **Rychlou Fourierovu Transformaci** užitím abychom mohli převést funkce získané po lineární interpolaci z časové do frekvenční oblasti.

Protože máme určitý časový úsek měření k jeho správnému znázornění ve frekvenční oblasti užitím **okna Kaissera**. [26]

Možné nastavení filtrů:

<i>PassbandFrequency</i>	<i>Frekvence propustného pásma [Hz]</i>
<i>StopbandFrequency</i>	<i>Frekvence nepropustného pásma [Hz]</i>
<i>PassbandRipple</i>	<i>Zvlnění propustného pásma [Hz]</i>
<i>Stopband Attenuation</i>	<i>Zeslabení nepřipustného pásma [-]</i>
<i>SampleRate</i>	<i>Vzorkovací frekvence [Hz]</i>

Dolní propust (z.*angl.***Low-pass filter**) filtr, který nepropouští signál o vysokých frekvencích [26].

```
lpFilt = designfilt('lowpassfir','PassbandFrequency',10, ...  
    'StopbandFrequency',12,'PassbandRipple',0.1, ...  
    'StopbandAttenuation',65,'DesignMethod','kaiserwin', ...  
    'SampleRate',1/T);  
DataFilt=filtfilt(lpFilt,vq');
```

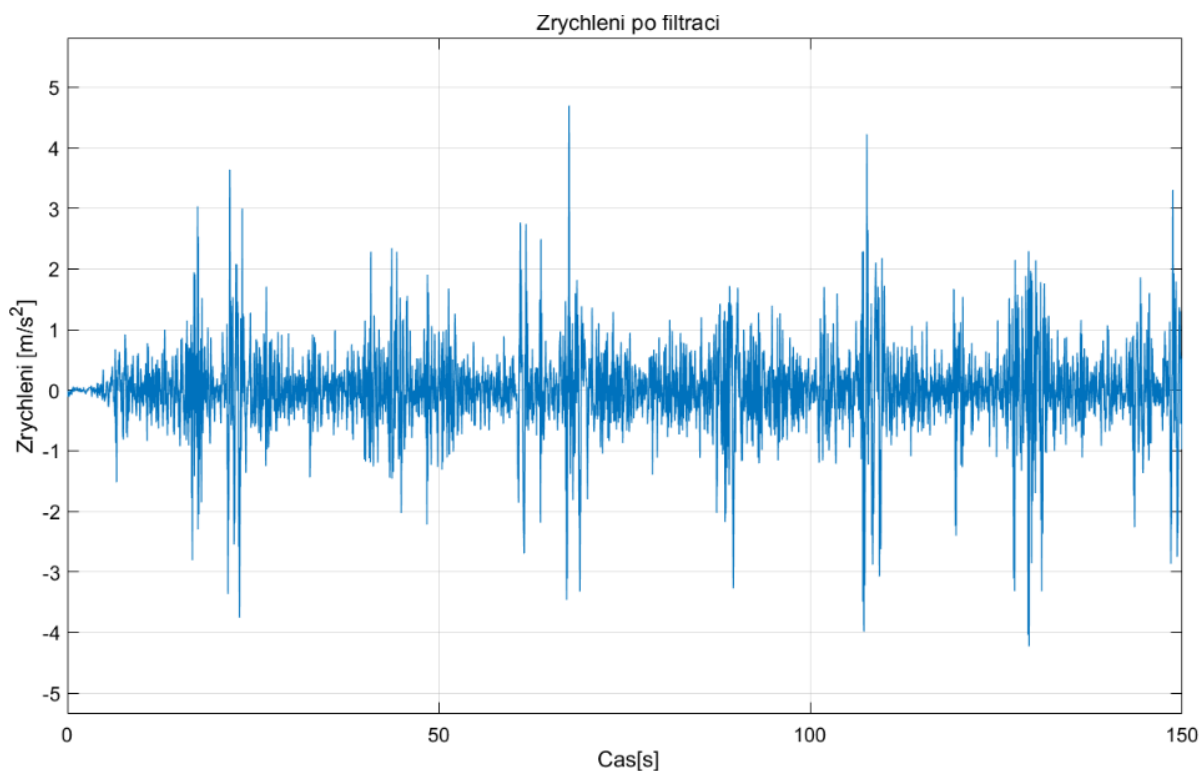
Obrázek 15 - Parametry nastavení filtru *Dolní propust*

Horní propust

Horní propust (z.*angl.***High-pass filter**) filtr, který nepropouští signál o nízkých frekvencích [26].

```
hpFilt = designfilt('highpassfir','PassbandFrequency',0.8, ...  
    'StopbandFrequency',0.03,'PassbandRipple',0.1, ...  
    'StopbandAttenuation',65,'DesignMethod','kaiserwin', ...  
    'SampleRate',1/T);  
DataFilt=filtfilt (hpFilt,DataFilt);  
  
[outXfilt,outYfilt]=fftZn(xq',DataFilt,1);
```

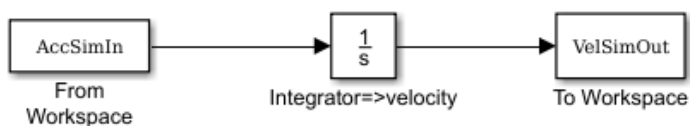
Obrázek 16 - Parametry nastavení filtru *Horní propust*



Obrázek 17 - Zrychlení po filtraci

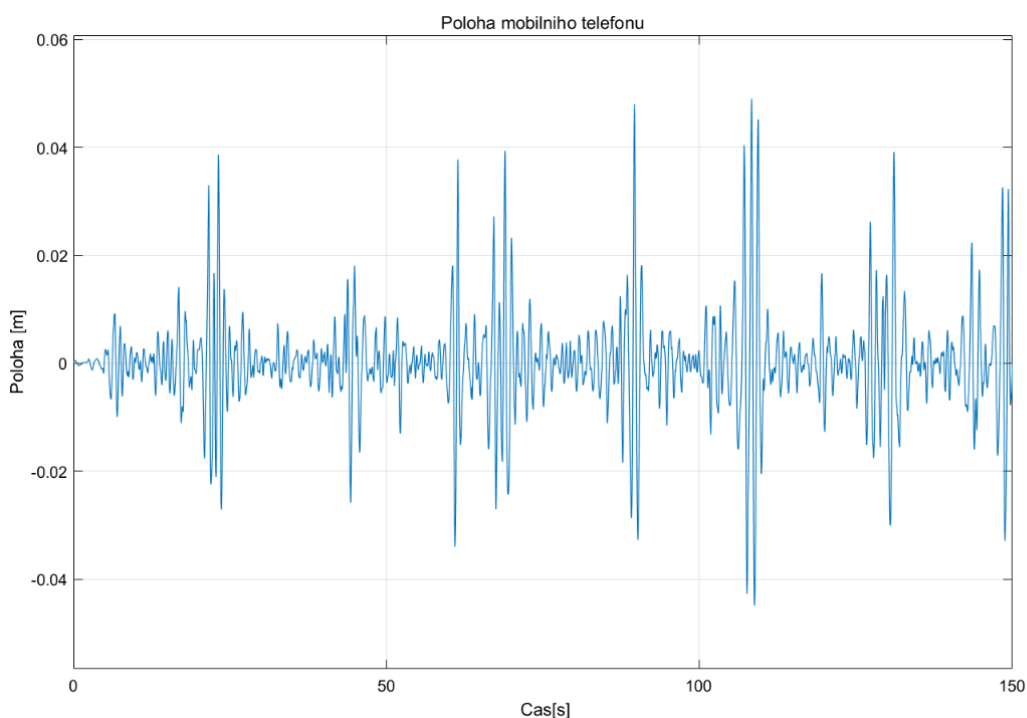
Užitím bloku *Integrátor* v prostředí *Simulink*, hodnoty zrychlení jsem integroval a následně odfiltroval. Stejným způsobem byly získané hodnoty polohy, k tomu jsem užil model *Acc2Vel.slx*

Výstupní hodnoty ze bloku po integraci byly zas odfiltrované užitím filtru *Horní propust*, nastavený pro filtraci frekvenci o *0.8 Hz*.



Obrázek 20 - Snímek modelu SIMULINKU *Acc2Vel.slx*

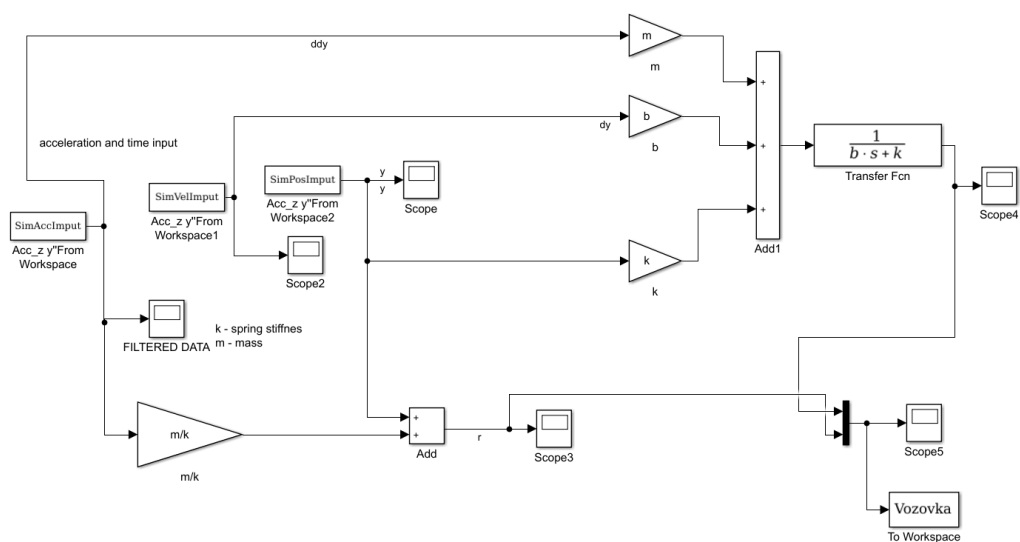
Výstupy jsou časový průběh rychlosti a polohy mobilu a pružené hmoty,



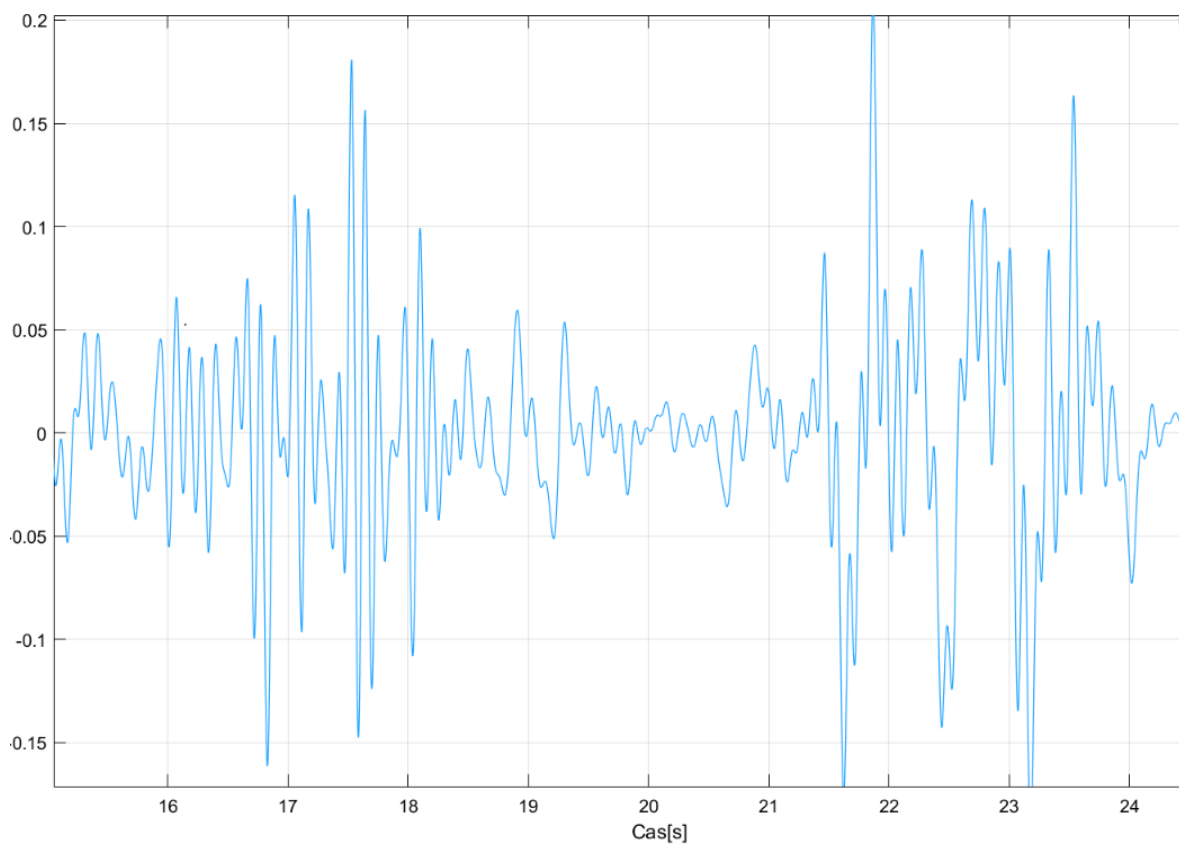
Obrázek 20 - Časový průběh polohy mobilu

Dalším krokem je tvorba modelu v nadstavbě *SIMULINK* vycházíme-li tady z rovnic (8),(9).

Vstupní signály modelů jsou hodnoty získané v předchozích krocích. Jsou násobené hodnotami příslušných koeficientů pro řešení rovnici (7). V tento okamžik je vyřešená rovnici (7). Dále celá tato rovnice je násobena přenosovou funkcí, blok *Transfer function*. Tímto získáme hledaný příběh profilu nerovnosti povrchu vozovky. Zbylé bloky v dolní části modelu byli užití pro předcházející verze řešení systému.



Obrázek 20 - Snímek modelu SIMULINKU *quarter_simplified.slx*



Obrázek 21 – Výřez z profilu povrchu vozovky (příloha 5)

Na Obrázku 21 je znázorněn průběh nerovnosti vozovky, ze kterého je vidět jak v okamžicích od 15 do 25 sekund, v rozsahu -0,2 až 0,2m pro ukázkou co lze identifikovat tímto modelem.

6. Závěr

V první části práce jsem se seznámil se senzory mobilních telefonů, které je možné použít k určení polohy v prostoru. Byli popsány jejich vlastnosti, které jsou důležité pro měření, provedeno porovnání změn těchto charakteristik.

Dále byl proveden průzkum aplikací, které slouží pro této účely definované omezení užití. Na konci byl popsán metodiky modelování kmitání vozidel.

V praktické části práce byl popsán proces tvorby aplikací pro mobilní operační systém iOS s naznačením nejdůležitějších pojmů a problémů vůči kterému jsme nechali pro další zpracování jen akcelerometr. Při měření mobilního telefonu ve vozidle byla do upřesněná jeho poloha. Z vybraného jednoduchého matematického modelu vozidla byl odvozen způsob získání profilu vozovky.

Dále byla popsána metodika zpracovávání dat v prostředí MATLAB a nadstavbě SIMULINK se stručným popisem použitých metod zpracování a filtraci signálu. Funkčnost algoritmu ukazuje získaný profil vozovky. Tento profil je zatížení máme příliš krátký dat. Funkčnost algoritmu ukazuje získaný průběh vozovky.

Dá se říct, že funkčnost je ověřena, pro lepší hodnocení bychom potřebovali provést delší měření, to by umožnilo lepší filtraci užitím *horní propusti*, realizace pro kalibraci *offsetu*.

Možnost praktického použití své práci vidím v možnosti vytvoření síťové aplikace pro mobilní zařízení, která by umožnila sledování kvality povrchu vozovky z dat uživatelů, pro další Zpracování těchto dat služby pro údržbu silnic. Avšak pro realizaci toho je třeba vytvořit algoritmu pro automatické hodnocení úseku, jeho zpracování pouze omezené množství dat, aby nedocházelo k zatížení sítě, což by umožnilo zmenšení nákladů na silniční správy a lepší sledování stavu silnic. Pro to by však bylo třeba realizaci aplikace pro konstantní měření, pro různé mobilní telefony, a různá vozidla, což vzhledem k stále nízké přesnosti senzorů a komplikaci ze strany výrobci mobilních telefonů není možné v blízké době.

Všechny cíle mé práce jsou splněné.

7. Seznam použité literatury

- [1] HUSÁK, Miroslav. *Mikrosenzory a mikroaktuátory. Vyd. 1. Praha: Academia, 2008. ISBN 978-80-200-1478-8.*
- [2] Galaxy S7 Edge *Teardown*. In: [Http://www.chipworks.com](http://www.chipworks.com) [online]. není: chipworks, 2016 [cit. 2017-08-18]. Dostupné z: <http://www.chipworks.com/about-chipworks/overview/blog/samsung-galaxy-s7-edge-teardown>
- [3] RICHTER, Jakub. *Monitorování pohybu akcelerometrem*. Praha, 2014.
- [4] Getting Raw Accelerometer Events. *Developer.apple.com* [online]. Kalifornie, USA: Apple Inc., 2017 [cit. 2017-08-18]. Dostupné z: https://developer.apple.com/documentation/coremotion/getting_raw_accelerometer_events
- [5] SUŠÁNEK, Filip. *Určení polohy objektu pomocí kombinace sensorů mobilního telefonu*. ČVUT, 2014. Bakalářská. ČVUT. Vedoucí práce Ing. Bc. Marek Neruda.
- [6] GyroMEMSschema.jpg. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2017-08-18]. Dostupné z: <https://commons.wikimedia.org/wiki/File:GyroMEMSschema.jpg>
- [7] Getting Raw Gyroscope Events. *Developer.apple.com* [online]. Kalifornie, USA: Apple Inc., 2017 [cit. 2017-08-18]. Dostupné z: https://developer.apple.com/documentation/coremotion/getting_raw_gyroscope_events
- [8] Solving problems of accelerometer,gyroscope,M7,compass and other iPhone 5S sensors [online]. Moskova: habrahabr.ru, 2013 [cit. 2017-08-20]. Dostupné z: <https://habrahabr.ru/post/196878/>
- [9] HUBR, IVO. MĚŘENÍ VZDÁLENOSTI POMOCÍ SENZORŮ MOBILNÍHO TELEFONU. Brno, 2013. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.
- [10] Comparing the InvenSense and *Bosch Accelerometers Found in the iPhone 6* [online]. Ottawa: Chipworks, 2014 [cit. 2017-08-20]. Dostupné z: <https://www.chipworks.com/about-chipworks/overview/blog/comparing-invensense-and-bosch-accelerometers-found-iphone-6>
- [11] Inside the iPhone 5s. *ChipWorks* [online]. Ottawa: TechInsights, 2013 [cit. 2017-08-18]. Dostupné z: <https://www.chipworks.com/about-chipworks/overview/blog/inside-the-iphone-5s>
- [12] MPU-6500 Product Specification *Revision 1.0*. In: [Https://invensense.com](https://invensense.com) [online]. CA USA: c-2, 2013 [cit. 2017-08-18]. Dostupné z: https://store.invensense.com/datasheets/invensense/MPU_6500_Rev1.0.pdf

- [13] *BMA 220 Datasheet*. In: [Www.mouser.com](http://www.mouser.com) [online]. Reutlingen: BOSCH Sensortech, 2011 [cit. 2017-08-18]. Dostupné z: <http://www.mouser.com/pdfdocs/BSTBMA220DS00308.PDF>
- [14] *L3GD20 Datasheet*. In: [Http://www.st.com/](http://www.st.com/) [online]. STMicroelectronics: STMicroelectronics, 2013 [cit. 2017-08-18]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/43/37/e3/06/b0/bf/48/bd/DM00036465.pdf/files/DM00036465.pdf/jcr:content/translations/en.DM00036465.pdf>
- [15] *LSM6DS3 Datasheet*. In: STMicroelectronics [online]. *STMicroelectronics*: STMicroelectronics, 2017 [cit. 2017-08-18]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/a3/f5/4f/ae/8e/44/41/d7/DM00133076.pdf/files/DM00133076.pdf/jcr:content/translations/en.DM00133076.pdf>
- [16] *Rotoview.com*. [Http://www.rotoview.com/index.htm](http://www.rotoview.com/index.htm) [online]. Houston TX 77099 USA: INNOVENTIONS, INC., 2017 [cit. 2017-08-18]. Dostupné z: <http://www.rotoview.com/index.htm>
- [17] *Stackoverflow* [online]. New York: Stackoverflow.com, 2017 [cit. 2017-08-18]. Dostupné z: <https://stackoverflow.com/>
- [18] KUKAČKA, Radek. *SIMULAČNÍ MODELOVÁNÍ TLUMIČE*. Brno, 2011.
- [19] *Odpružení vozidel*. Střední škola automobilní Kyjov, 2015. Střední škola automobilní Kyjov.
- [20] A universal suspension test rig for electrohydraulic active and passive automotive suspension system. *Alexandria Engineering Journal* [online]. 2017, 2017(2), 12 [cit. 2017-08-18]. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S1110016817300273>
- [21] Simulation Aspects of a Full-Car ATV Model Semi-Active Suspension. *Engineering Vol.4*. 2012, 2012(7), 4. DOI: DOI:10.4236/eng.2012.47050.
- [22] Half-car model. In: *Mathworks [online]*. Natick: Mathworks, 2017 [cit. 2017-08-18]. Dostupné z: https://www.mathworks.com/help/examples/simulink_product/win64/sldemo_susp_figure1.png
- [23] A Vehicle Steering Recognition System Based on Low-Cost Smartphone Sensors. *Sensors* [online]. 2017, 2017(173), 1 [cit. 2017-08-18]. Dostupné z: <http://www.mdpi.com/1424-8220/17/3/633/htm>
- [24] PASIVE SUSPENSION MODELING USING MATLAB, QUARTER CAR MODEL, IMPUT SIGNAL STEP TYPE. *TEHNOMUS - New Technologies and Products in Machine*

Manufacturing Technologies [online]. 2013, **2013**(45), 6 [cit. 2017-08-18]. Dostupné z: http://www.fim.usv.ro/conf_1/tehnomusjournal/pagini/journal2013/files/45.pdf

[25] VALÁŠEK, M., V. BAUMA a Šika Z. *Mechanika* B. 1. Praha: Nakladatelství ČVUT, 2006. ISBN 80-01-02919-0.

[26] Designing Low Pass FIR Filters [online]. *Germany: Mathworks*, 2016 [cit. 2017-08-20]. Dostupné z: https://www.mathworks.com/help/dsp/examples/designing-low-pass-fir-filters.html?searchHighlight=low%20pass%20filter&s_tid=doc_srchtile

Seznam příloh

- 1.SensorRecord.zip – složka s kódem a příslušné soubory aplikaci pro iOS 9 a vyšší
- 2.SensorRecord1.zip – zbylé soubory
- 3.PythonSkript.7z – složka se skriptem pro zpracování dat po aplikaci.
4. MatlabSkriptsModels.7z– soubory pro zpracování dat v Matlab a Simulink
- 5.RoadSurface.fig – výsledný příběh povrchu vozovky