

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-104378-80074

**ANALÝZA KMITANIA NA ŠTVRTINOVOM FYZIKÁLNYM
MODELI PRUŽENIA AUTOMOBILU**

DIPLOMOVÁ PRÁCA

2021

Bc. Stanislav Révay

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-104378-80074

**ANALÝZA KMITANIA NA ŠTVRTINOVOM FYZIKÁLNYM
MODELI PRUŽENIA AUTOMOBILU
DIPLOMOVÁ PRÁCA**

Študijný program:	Aplikovaná mechatronika a elektromobilita
Študijný odbor:	kybernetika
Školiace pracovisko:	Ústav automobilovej mechatroniky
Vedúci diplomovej práce:	doc. Ing. Vladimír Goga, PhD.

Bratislava 2021

Bc. Stanislav Révay



ZADANIE DIPLOMOVEJ PRÁCE

Autor práce: Bc. Stanislav Révay
Študijný program: aplikovaná mechatronika a elektromobilita
Študijný odbor: kybernetika
Evidenčné číslo: FEI-104378-80074
ID študenta: 80074
Vedúci práce: doc. Ing. Vladimír Goga, PhD.
Miesto vypracovania: Ústav automobilovej mechatroniky

Názov práce: **Analýza kmitania na štvrtinovom fyzikálnom modeli pruženia automobilu**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania: Cieľom práce je zostaviť štvrtinový fyzikálny model pruženia automobilu s možnosťou jeho kinematického budenia od vozovky. Model bude osadený snímačmi zrýchlenia pre analýzu kmitania automobilu pri prejazde nerovnosťami.
Úlohy:
1. Navrhnete a zostrojíte geometrický model pruženia.
2. Definujete spôsob budenia modelu.
3. Aplikujete snímače zrýchlenia a merací reťazec.
4. Analyzujete kmitanie modelu a porovnáajte výsledky s matematickým modelom.

Dátum zadania: 15. 02. 2021

Dátum odovzdania: 14. 05. 2021

Bc. Stanislav Révay
študent

prof. Ing. Vladimír Kutiš, PhD.
vedúci pracoviska

prof. Ing. Vladimír Kutiš, PhD.
garant študijného programu

Čestne vyhlasujem, že som diplomovú prácu „*Analýza kmitania na štvrtinovom fyzikálnom modeli pruženia automobilu*“ vypracoval samostatne na základe konzultácií a s použitím uvedenej literatúry.

V Bratislave, máj 2021

.....

Stanislav Révay

Pod'akovanie

Rád by som sa pod'akoval vedúcemu práce doc. Ing. Vladimírovi Gogovi, PhD., za prejavenú dôveru a rady počas tvorby modelu a práce ako celku. Moje osobitné pod'akovanie patrí aj môjmu otcovi, ktorého rady boli dôležitou súčasťou vytvárania fyzikálneho modelu.

ANOTÁCIA DIPLOMOVEJ PRÁCE

Slovenská technická univerzita v Bratislave
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný odbor: kybernetika

Študijný program: Aplikovaná mechatronika a elektromobilita

Autor: Bc. Stanislav Révay

Diplomová práca: Analýza kmitania na štvrtinovom fyzikálnom modeli pruženia automobilu

Vedúci práce: doc. Ing. Vladimír Goga, PhD.

Mesiac a rok odovzdania: máj 2021

Kľúčové slová: lichobežníková náprava, geometrický model, matematický model, Matlab, 3D

Cieľom práce bolo zostrojiť fyzikálny štvrtinový model pruženia automobilu, na ktorom by bolo možné analyzovať kmitanie odpruženej a neodpruženej hmoty pri jeho vhodnom buzení od nerovnosti vozovky. Práca pozostáva z 3D konštrukčného návrhu v CAD programe Fusion 360 a následnej praktickej realizácii fyzického modelu pruženia lichobežníkovej nápravy spolu s vhodným systémom kinematického budenia. Na vybrané časti nápravy boli inštalované snímače zrýchlenia, ktorých signál sa spracoval za pomoci mikrokontroléra a sériovou komunikáciou sa prenášali do počítača. Na vyhodnotenie meraní bolo vytvorené používateľské prostredie v programe Matlab. Vytvorený bol taktiež simulačný model nápravy, ktorého dynamická analýza bolo riešená v CAE programe MSC.Adams a jej výsledky možno porovnať s experimentálne nameranými dátami z fyzikálneho modelu. Oba modely môžu poslúžiť ako výučbová pomôcka pre študijné programy zamerané na oblasť kybernetiky a mechatroniky.

ABSTRACT OF THE DIPLOMA THESIS

Slovak University of Technology in Bratislava

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Branch:	Cybernetics
Study Programme:	Applied mechatronics and electromobility
Author:	Bc. Stanislav Révay
Bachelor Thesis:	Vibration analysis on quarter physical model of the car suspension
Supervisor:	doc. Ing. Vladimír Goga, PhD.
Submitted:	May, 2021
Keywords:	double-wishbone suspension, geometric model, mathematical model, Matlab, 3D

The aim of the work was to construct a physical quarter model of the car's suspension, on which it would be possible to analyze the oscillation of the sprung and unsprung mass with its appropriate excitation from the unevenness of the road. The work consists of a 3D structural design in the CAD program Fusion 360 and the subsequent practical implementation of a physical model of the double-wishbone suspension together with a suitable kinematic excitation system. Acceleration sensors were installed on selected parts of the axle, the signal of which was processed by means of a microcontroller and transmitted to the computer by serial communication. A user interface in the Matlab program was created to evaluate the measurements. A simulation model of the axle was also created, the dynamic analysis of which was solved in the CAE program MSC.Adams and its results can be compared with experimentally measured data from the physical model. Both models can serve as a teaching aid for study programs focused on cybernetics and mechatronics.

Obsah

ÚVOD.....	13
1 SYSTÉM PRUŽENIA AUTOMOBILU	14
1.1 PASÍVNE SYSTÉMY PRUŽENIA	14
1.2 SEMIAKTÍVNE SYSTÉMY PRUŽENIA	14
1.3 AKTÍVNE SYSTÉMY PRUŽENIA.....	15
1.4 DELENIE NÁPRAV NA ZÁKLADE TYPU KONŠTRUKCIE	15
1.4.1 Tuhé nápravy.....	15
1.4.2 Výkyvné nápravy.....	18
2 NÁVRH A ŠPECIFIKÁCIA ZADANIA.....	22
2.1 CIEĽ PROJEKTU	22
2.2 ŠPECIFIKÁCIA A ŠTRUKTÚRA PROJEKTU	23
2.3 ŠTVRTINOVÝ MATEMATICKÝ MODEL VOZIDLA	24
3 NÁVRH A ZHOTOVENIE MODELU.....	26
3.1 DEFINOVANIE VZHĽADU FYZIKÁLNEHO MODELU	26
3.2 ZHOTOVENIE MODELU	27
3.2.1 Nosná konštrukcia fyzikálneho modelu	28
3.2.2 Model nápravy – Lichobežníková náprava	30
3.2.3 Hnacia sústava.....	36
3.2.4 Koleso a estetické prvky modelu.....	38
4 ELEKTRONIKA SYSTÉMU	39
4.1 SCHÉMA ZAPOJENIA	39
4.2 RIADENIE	40
4.2.1 Mikropočítač Maker UNO.....	40
4.2.2 Zapojenie	41
4.3 HNACIA SÚSTAVA	42
4.4 SNÍMAČE ZRÝCHLENIA	46
5 RIADIACI PROGRAM.....	47
5.1 PROGRAM MIKROPOČÍTAČA MAKER UNO	47
5.1.1 Premenné a funkcia „Setup“.....	47
5.1.2 Sériová komunikácia s GUI	48
5.1.3 Proces merania	50
5.2 PROGRAM GRAFICKÉHO POUŽÍVATEĽSKÉHO ROZHRAŇIA (MATLAB)	53

5.2.1	Premenné programu GUI.....	54
5.2.2	Popis aplikácie	55
6	POROVNANIE VÝSLEDKOV SIMULÁCIE A REÁLNEHO MODELU.....	60
6.1	SIMULÁCIA.....	60
6.2	POROVNANIE VÝSLEDKOV.....	62
	ZÁVER.....	65
7	BIBLIOGRAFIA	67
	PRÍLOHY.....	I
Príloha A:	POUŽIVATEĽSKÝ MANUÁL	II
Príloha B:	PROGRAM APLIKÁCIE RIADENIA A MIKROPOČÍTAČA	VIII
Príloha C:	TECHNICKÁ DOKUMENTÁCIA	XXII
Príloha D:	FOTODOKUMENTÁCIA.....	XL
Príloha E:	ŠTRUKTÚRA REPOZITÁRA	XLV

Zoznam použitých skratiek a označení

μ	micro, 10^{-6}
ms	milisekunda 10^{-3} s
V	volt, základná jednotka napätia v sústave SI
CAE	Computer-aided Engineering (Počítačom podporené inžinierstvo)
GUI	Graphical User Interface (Grafické používateľské rozhranie)
DOF	Degrees of Freedom (Stupne voľnosti)
CNC	Computer Numeric Controll
CAD	Computer-aided Design (Počítačom podporený dizajn)
RC	Remote Control (Diaľkové ovládanie)
CSV	Comma Separated Values (Hodnoty oddelené čiarkou)
OS	Operačný systém
px	pixel

Úvod

Automobilový priemysel a doprava vo všeobecnosti je jedným z najrýchlejšie sa rozvíjajúcich segmentov priemyslu. Aktuálnym trendom je narastajúci počet elektromobilov a znižovanie výroby dopravných prostriedkov so spaľovacím motorom. Typ pohonu je však len jedným z dôležitých prvkov dopravného prostriedku. V oblasti automobilov je rovnako dôležitým prvkom aj sústava tlmenia nerovností vozovky, ktorej hlavnými úlohami sú - zlepšenie komfortu cestujúcich, lepšia priľnavosť pneumatík k vozovke, riaditeľnosť vozidla a rovnako aj predĺženie životnosti mechanických a elektrických dielov automobilu a to znížením otrasov, ktorých následkom by mohlo dôjsť k poškodeniu dielov.

V súčasnosti existuje niekoľko typov odpružení automobilu. Jednotlivé typy sú navzájom odlišné a prinášajú rôzne výhody a nevýhody. Väčšinou zdieľajú spoločný cieľ, ktorým je dobrá ovládateľnosť vozidla a komfort jeho posádky. Rovnako dôležitými faktormi sú náklady na výrobu, priestor, ktorý odpruženie vo vozidle zaberá, komplikovanosť spojov ako aj hmotnosť vozidla. Rôzne typy odpruženia rôzne reagujú na nerovnosti na vozovke. Preto je dôležitým faktorom aj terén, ktorým vozidlo prechádza. Terénne vozidlá disponujú odlišným typom odpruženia ako vozidlá športové, či vozidlá vyššej triedy.

Cieľom tejto práce bolo vytvorenie fyzikálneho modelu pruženia automobilu, ktorého vlastnosti pruženia by bolo možné sledovať pomocou senzorov a vhodným spôsobom zaznamenávať a interpretovať. Pre účely tejto práce sme si vybrali lichobežníkovú nápravu, ktorá sa využíva vo vozidlách vyššej triedy. Takáto náprava má dobré vlastnosti pruženia a spoľahlivosti, avšak pre jej zložitejšiu výrobu a tým pádom aj vyššiu cenu sa inštaluje do áut vyššej triedy. Vďaka pozitívnym vlastnostiam v oblasti stability je táto náprava vhodná na vytvorenie fyzikálneho modelu. Vytvorený model sme porovnávali s matematickým modelom vytvoreným pomocou programu Fusion 360, ktorého vlastnosti sme následne simulovali pomocou programu MSC Adams. Vlastnosti oboch vytvorených modelov (matematického aj fyzikálneho) sme navzájom porovnali. Pre vhodnú interpretáciu rozdielov jednotlivých modelov sme vytvorili grafické prostredie, ktoré sme použili na ovládanie fyzikálneho modelu a zároveň na zobrazenie výsledkov v podobe grafov oboch modelov.

1 Systém pruženia automobilu

Systém pruženia automobilov má za úlohu tmiť nerovnosti vozovky za účelom zvýšenia jazdného komfortu a ovládateľnosti vozidla. V neposlednom rade zvyšuje aj bezpečnosť jazdy a to udržiavaním stáleho kontaktu medzi pneumatikou a vozovkou. Okrem najčastejšie používaného pasívneho pruženia sa využívajú aj semiaktívne či aktívne systémy pruženia, ktorých výroba je náročnejšia, avšak prinášajú viacero výhod.

1.1 Pasívne systémy pruženia

Najčastejšie používaným systémom pruženia je pasívne pruženie skladajúce sa z pružiny a tlmiča. Vlastnosti sústavy pruženia sú charakterizované fyzikálnymi veličinami koeficientu tuhosti pružiny a koeficientu tlmenia tlmiča. Následné jazdné vlastnosti vozidla sú závislé práve od týchto parametrov, ktorých výber je založený na matematických výpočtoch dynamiky vozidla. Vo vozidlách sa používajú:

- vinuté,
- listové,
- torzné,

pružiny v kombinácii s hydraulickým tlmičom. Tuhosť pružiny a koeficient tlmenia tlmiča sú konštantnými vlastnosťami. Pružina slúži na udržiavanie kontaktu medzi vozidlom a vozovkou za účelom zabezpečenia ovládateľnosti automobilu. Na druhej strane, úlohou tlmiča je tlmenie kmitania pružiny za účelom dosiahnutia jazdného komfortu.

1.2 Semiaktívne systémy pruženia

Semiaktívne systémy pruženia disponujú možnosťou zmeny účinku tlmenia vertikálneho kmitania. Táto zmena je buď manuálna alebo elektronická. Semiaktívne systémy umožňujú rýchle prepínanie tlmenia alebo pruženia. Každý bod charakteristiky je možné regulovať veľmi rýchlo a to s dobou menej ako 10 ms. Semiaktívne pruženie ponúka možnosť nastavenia si pruženia, no neponúka možnosť zmeny nastavenia počas jazdy a nedokáže vyvinúť silové účinky.

1.3 Aktívne systémy pruženia

Na rozdiel od semiaktívnych systémov pruženia, aktívne systémy odpruženia dokážu vyvinúť silové pôsobenie. Používajú pohonný systém umiestnený na každom kolese, ktorý umožňuje manipuláciu s karosériou, čím udržuje zaťaženie kola na takmer statickej hodnote. V porovnaní so semiaktívnym systémom pruženia má však aktívne vyššiu spotrebu energie. [1] V súčasnosti rozoznávame tri základné princípy aktívneho pruženia:

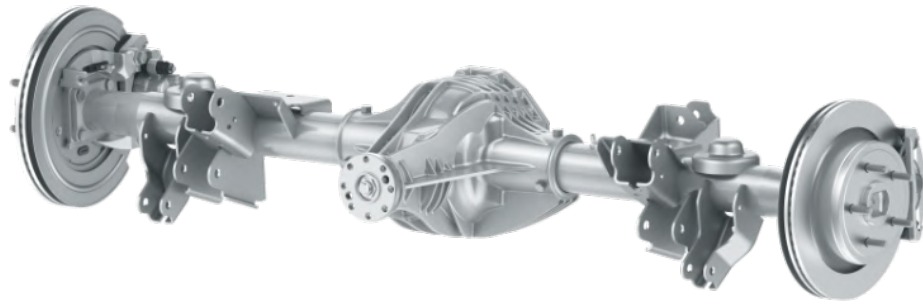
- Lotus: využíva 4 veľmi rýchle hydraulické valce zásobované pomocnou energiou
 - ◊ medzi hydraulickým valcom a karosériou je snímač zaťaženia kola
 - ◊ snímač zásobuje regulačný okruh a udržuje statické zaťaženie kola
- Williams: hydropneumatický systém odpruženia
 - ◊ pohyby karosérie sú riadené reguláciou oleja v obvode systému
- Horvat: pneumatický systém odpruženia
 - ◊ pohyby karosérie sú riadené reguláciou tlaku vzduchu v obvode systému [1]

1.4 Delenie náprav na základe typu konštrukcie

1.4.1 Tuhé nápravy

Jedným z prvých typov náprav boli tuhé nápravy resp., závislé zavesenie kolies. V súčasnosti sa tento typ náprav vyskytuje vo vozidlách s vyššou hmotnosťou ako sú nákladné vozidlá a autobusy a rovnako sa tuhá náprava používa aj v terénnych vozidlách. Obe kolesá tuhej nápravy sú navzájom pevne prepojené, pneumatika je v kontakte s vozovkou v celej šírke behúňa a pri prepružení sa nemení rozchod kolies ani vzájomné postavenie. Tuhá náprava je odpružená buď listovými perami alebo vinutými pružinami. Listové perá sú ukotvené priamo na karosérii alebo ráme vozidla okrem pruženia zabezpečujú aj vedenie nápravy. V prípade vinutých pružín je potrebné použiť dodatočné priečne a pozdĺžne vedenie, keďže neprenesú takmer žiadne priečne sily na rozdiel od listových pier. [2]

Nevýhodami tuhej nápravy sú napríklad: horší komfort na nerovnom povrchu, horšie jazdné vlastnosti pri rýchlejšej jazde a menej presné vedenie kola.

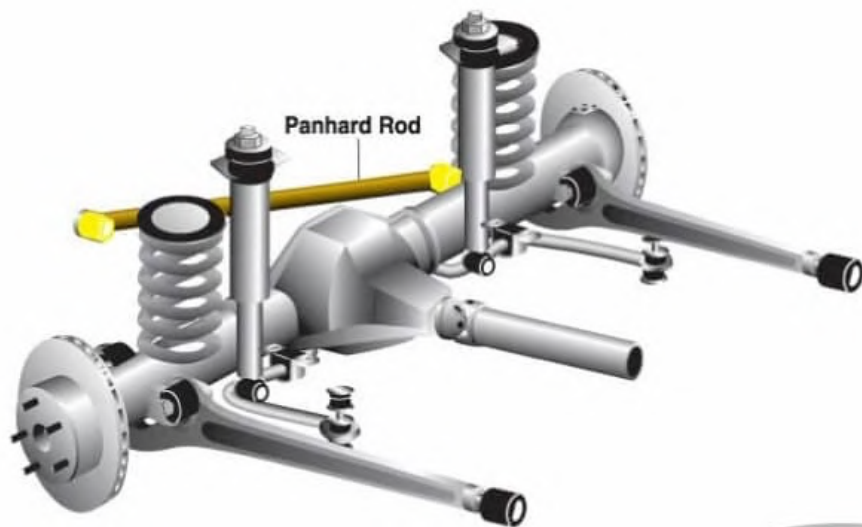


Obr. 1 Tuhá náprava [3]

Panhardská tyč

Panhardská tyč využíva na zabezpečenie vedenia nápravy v priečnom smere priečne ložené rameno spájajúce most nápravy s rámom alebo karosériou vozidla. Nevýhodou tejto konštrukcie je bočný posuv nápravy voči vozidlu počas prepruženia, čo má za následok zhoršenie jazdného komfortu. Nevýhodu je do istej miery možné eliminovať predĺžením konštrukcie a takmer vodorovným uložením panhardskej tyče. [2]

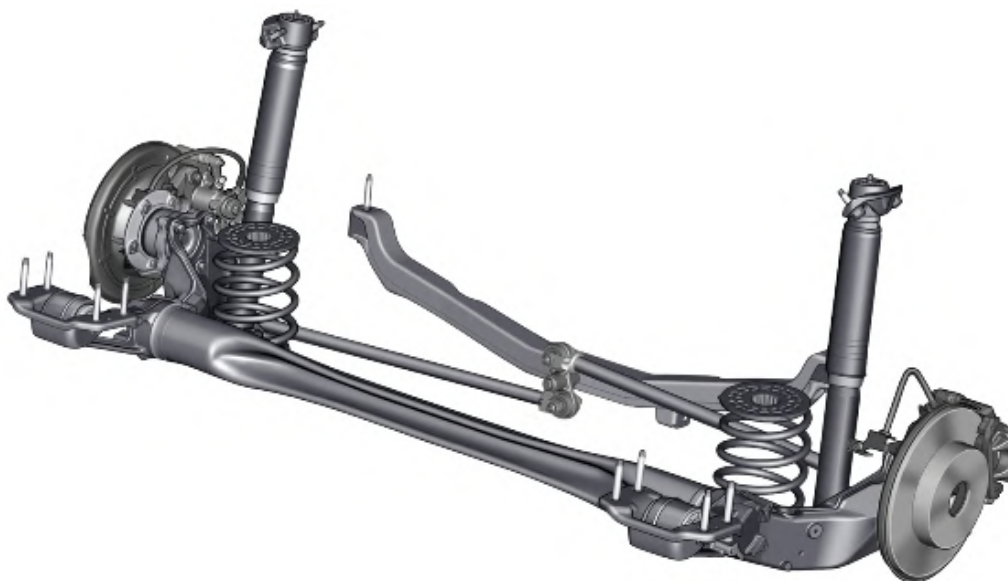
Nápravu s panhardskou tyčou sme znázornili na Obr. 2.



Obr. 2 Panhardská tyč [4]

Wattov priamovod

Vynálezcom Wattovho priamovodu je James Watt. Tento typ nápravy tvoria dve ramená rovnakej dĺžky. Stred priamovodu je tvorený otočným členom uloženým na moste nápravy, ktorý je spojený s karosériou alebo rámom vozidla. Toto spojenie zabezpečuje tuhé, priečne vedenie nápravy a súčasne eliminuje bočný pohyb prejavujúci sa v prípade použitia panhardskej tyče. [2]

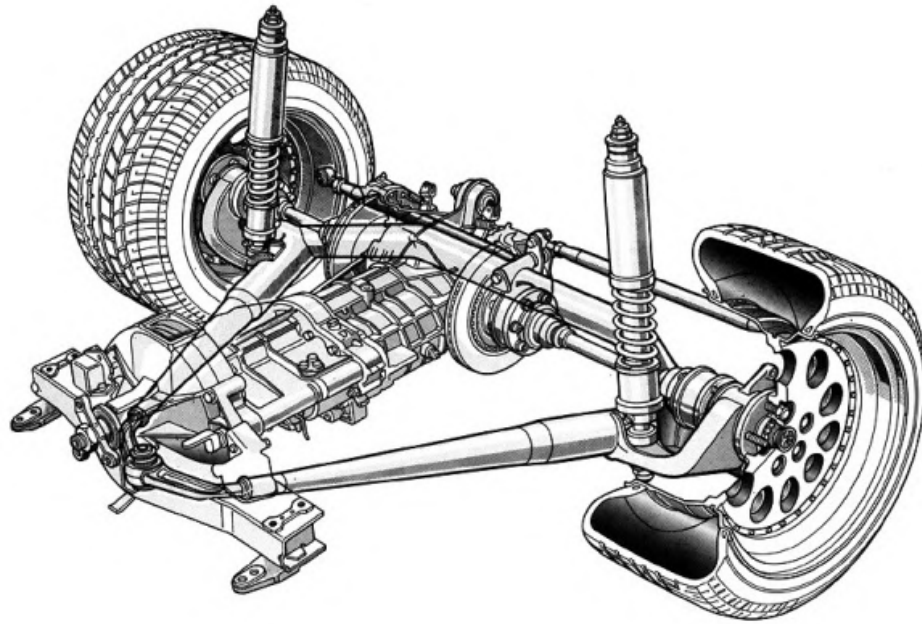


Obr. 3 Wattov priamovod - Opel Astra [5]

De-Dion

Náprava nesie meno po grófovi De Dion, ktorý ju prvýkrát použil v roku 1896. Dnes sa náprava používa v osobných a športových vozidlách. Kolesá nápravy sú spojené tuhým mostom, ktorý je vedený Wattovým priamovodom alebo Panhardskou tyčou, čím sú zachytené priečne sily. Pozdĺžne vedenie náprava je zabezpečené dvojicou šikmých ramien. Rozvodovka je uložená na karosérii alebo ráme vozidla a krútiaci moment je prenesený na kolesá pomocou kĺbových hriadeľov s premenlivou dĺžkou. [2]

Náprava De-Dion znižuje veľkosť neodpružených hmôt hnacej tuhej nápravy. Oproti tuhej náprave dochádza k miernemu zlepšeniu jazdných vlastností a komfortu.



Obr. 4 Náprava De-Dion [6]

1.4.2 Výkyvné nápravy

Výkyvné nápravy, označované aj ako nezávislé zavesenie kolies, disponujú viacerými pozitívnymi vlastnosťami ako napríklad:

- Lepší komfort jazdy a jazdné vlastnosti,
- Menšia hmotnosť neodpružených častí,
- Dostatok miesta pre motor a iné konštrukčné časti vozidla

Prinášajú so sebou aj viacero nevýhod:

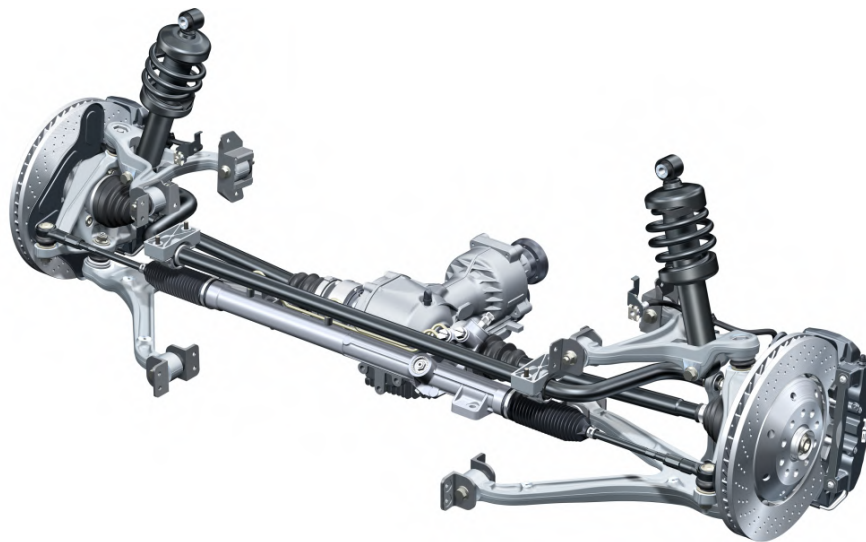
- Náročnejšia konštrukcia, z čoho vyplýva aj drahšia výroba,
- Rýchle opotrebovanie

Lichobežníková náprava

Lichobežníková náprava je tvorená horným a spodným priečnym trojuholníkovým ramenom, ktoré v priemete do zvislej roviny tvoria lichobežník. Ramená sú uchytené buď k nápravnici, rámu vozidla alebo v niektorých prípadoch aj k rozvodovke. Spodné rameno je väčšinou robustnejšie z dôvodu prenášania zvislých síl. Ramená sú uložené v gumových puzdrách, pružiny väčšinou bývajú uložené na spodnom ramene. Pri prepružení dochádza k zmene odklonu kolesa, zbiehavosti a tiež rozchodu kolies, čo má nepriaznivý vplyv na jazdné vlastnosti vozidla. Pre elimináciu tohto javu je dôležitá optimálna konštrukcia ramien a tiež

nastavenie geometrie. Ramená by mali byť rovnobežne uložené, aby bod klopenia kola ležal vo väčšej vzdialenosti od kola. Toto riešenie znižuje odklon kola a zmenu rozchodu kolies pri prepružení. Nevýhodou je však posunutý stred klopenia nápravy do roviny vozovky, čo má nepriaznivý vplyv na polohu osi klopenia vozidla. V praxi sú ramená rôznej dĺžky, čím sa mení uhol, ktorý zvierajú pri prepružení kola. Tým sa mení aj poloha okamžitého bodu klopenia kola a poloha stredu klopenia nápravy. [2]

Lichobežníková náprava pri správnej konštrukcii a geometrii zabezpečuje veľmi dobré vedenie kola a tým pádom veľmi dobré jazdné vlastnosti vozidla. Nevýhodou je však pomerne zložitá konštrukcia a vyššie výrobné náklady. Z toho dôvodu sa využíva v športových vozidlách a vo vozidlách strednej a vyššej triedy. Lichobežníkovú nápravu je možné použiť ako prednú hnanú aj hnaciu nápravu alebo aj zadnú hnanú aj hnaciu nápravu. [2]



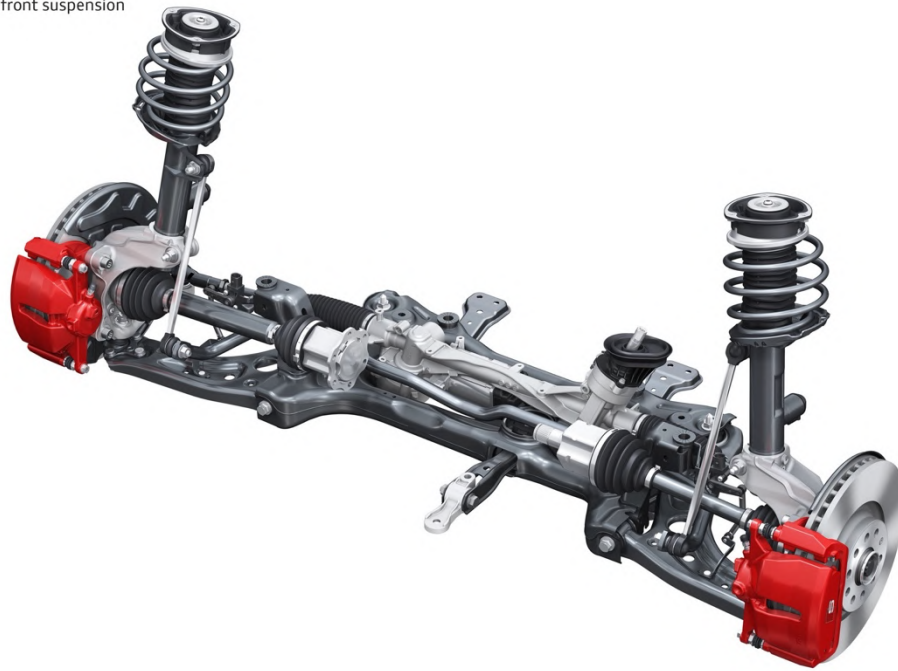
Obr. 5 Lichobežníková náprava - Audi R8 [7]

Mc Pherson

Náprava Mc Pherson je odvodená z lichobežníkovej nápravy, kde vrchné rameno je nahradené posuvným vedením. Vrchná časť je tak podstatne kompaktnjšia, čím vzniká viac priestoru pre ostatné komponenty vozidla. Na prednej riadiacej náprave musí horné uloženie tlmiča umožniť natáčanie. Aby sa pri otáčaní nekrútila aj vinutá pružina, býva vrchný koniec pružiny uložený pomocou valivého ložiska. Pružina je uložená na plášti tlmiča, tým pádom nie je posuvné vedenie zaťažované zvislými silami a nevzniká nadmerné trenie v ložisku. K zvýšenému treniu

dochádza od momentov priečných a pozdĺžnych síl pri zrýchľovaní, brzdení a zatáčaní. Tento jav sa eliminuje pomocou šikmého uloženia pružiny, gumovým uložením horného ložiska a tiež robustnejšou konštrukciou. Nežiaducim javom je odklon kolesa pri prepružení čím sa zhoršujú jazdné vlastnosti a komfort jazdy. Výhodou nápravy McPherson je jednoduchá a finančne nenáročná konštrukcia s minimálnym počtom dielov. Uplatnenie je vo vozidlách strednej triedy. Nápravu je možné použiť ako prednú hnanú aj hnaciu alebo zadnú hnanú aj hnaciu. [2]

McPherson front suspension
11/20



Obr. 6 Náprava McPherson - Audi SQ2 [8]

Náprava typu Multilink

Zavesenie multi-link výrazne zlepšilo vlastnosti priečne uloženého torzne pružného nosníka, z ktorého konštrukcia vychádza. Multilink využíva na spojenie zadných kolies ocelový nosník s prierezom v tvare prevráteného písmena U, ktorý je veľmi tuhý na ohyb a zároveň poddajný na krútenie. Nosník je pozdĺžne uchytený párom vodiacich ramien a na svojich vonkajších koncoch je zvisle uchytený vinutými pružinami s tlmičmi a špeciálne tvarovaným zvislým rameno. Mechanizmus Scott-Russel zahŕňa priečne rameno a riadiacu tyč. Podobne ako Panhardská tyč spája aj priečne rameno a torzne pružný nosník so skeletom karosérie. Zabezpečuje priečne pevné upevnenie, čo umožňuje aby vlečené ramená boli čo najtenšie. Na rozdiel od Panhardskej tyče sa priečne rameno vozidla neotáča v pevnom bode na torzne pružnom nosníku. Je pripevnené cez špeciálne púzdro, ktoré je vertikálne tuhé, ale stranovo poddajné. Kratšia riadacia tyč spája priečne rameno a torzný nosník vo vnútri vonkajšieho

púzdra. Keď sa os torzného nosníka dvíha a klesá vzhľadom ku karosérii, pracuje mechanizmus ako Panhardská tyč. Zadné kolesá sa voči karosérii pohybujú len vertikálne, bez rozdielu medzi zatáčaním vpravo alebo vľavo. Spojenie tiež umožňuje veľmi malý pohyb medzi stredom otáčania a ťažiskom, keď sa náprava zdvíha alebo klesá. Nápravu multilink je možné použiť ako prednú hnanú aj hnaciu alebo zadnú hnanú nápravu. [2]

Mehrlenker-Hinterachse
Multi-link rear suspension
03/20



Obr. 7 Náprava Multilink - Audi A3 Sportback [9]

2 Návrh a špecifikácia zadania

Prvým krokom k správne popisu práce je jednoznačné vymedzenie bodov, ktoré boli prioritnými prvkami jej vypracovania. V tejto kapitole je popísané, čím sa práca zaoberala a aké problémy rieši.

2.1 Cieľ projektu

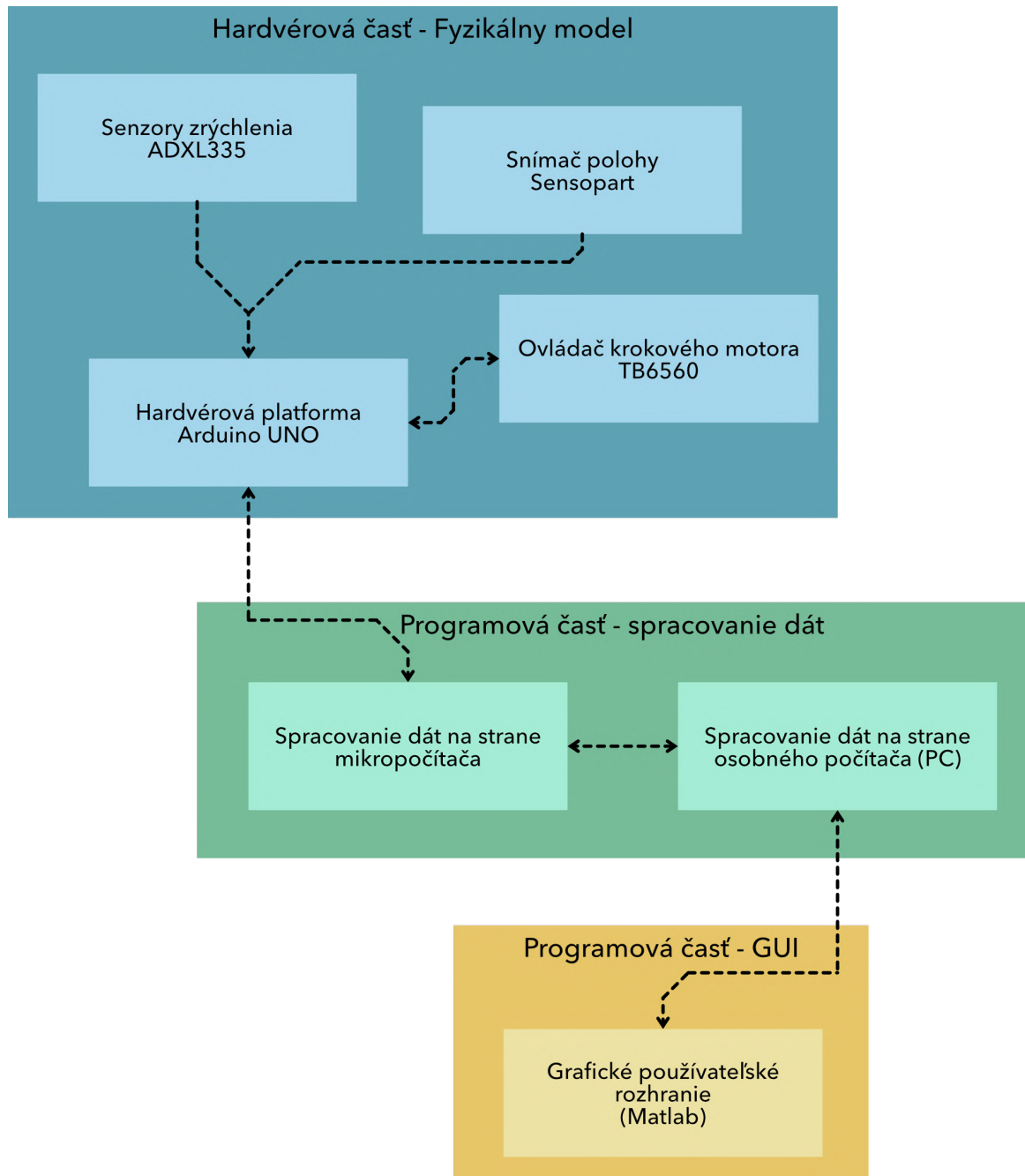
Projekt sa zaoberá celým životným cyklom geometrického modelu pruženia automobilu od jeho vzniku až po meranie fyzikálnej veličiny zrýchlenia komponentov geometrického modelu. Hlavným cieľom je možnosť merania zrýchlenia na dvoch miestach modelu a to v blízkosti kolesa, ktoré je budené vozovkou a na tlmenej odpruženej časti predstavujúcej automobil. Následne by malo byť možné údaje z reálneho geometrického modelu porovnať s údajmi, ktoré poskytuje matematický model vytvorený počítačovým CAE systémom. Očakávaným výsledkom je znázorniť podobnosť fyzikálneho a geometrického modelu prostredníctvom grafov zrýchlenia odpruženej a neodpruženej hmoty na oboch modeloch. Meranie a zobrazovanie nameraných dát by malo byť dostupné prostredníctvom grafického používateľského prostredia (ďalej len GUI). Očakávanou úlohou GUI je jednoduché ovládanie fyzikálneho modelu použitím tlačidiel, vstupných polí a informačných panelov sprostredkujúcich aktuálny stav fyzikálneho modelu a sériového pripojenia.

Fyzikálny model by mal byť vyrobený z materiálov, ktoré majú dostatočnú pevnosť pre zabezpečenie stabilného chodu modelu aj pri záťaži. Rovnako sa očakáva nenáročná obrábatelnosť použitých materiálov nakoľko by výroba modelu mala prebiehať v dielni so základným dielenským vybavením. Fyzikálny model by sa mal čo v najväčšej možnej miere svojimi vlastnosťami priblížiť reálnemu pruženiu automobilu, aby boli namerané údaje dôveryhodné.

Na riadenie fyzikálneho modelu sa využije mikrokontrolér alebo mikropočítač s analógovými vstupmi, ktoré budú schopné prijímať analógové hodnoty zo snímačov zrýchlenia. Na budenie vozovky sa využije vhodný typ motora, v prípade potreby aj s prevodovkou, riadený mikropočítačom.

2.2 Špecifikácia a štruktúra projektu

Pre lepšie pochopenie štruktúry projektu sme projekt rozdelili na niekoľko logických častí. Vďaka rozdeleniu projektu do logickej štruktúry sme sa počas jeho vypracovávania dokázali lepšie orientovať a tvorba projektu bola prehľadnejšia.



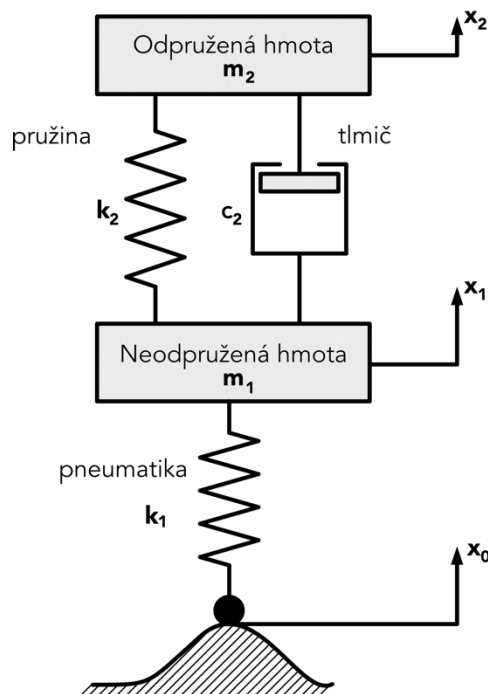
Obr. 8 Štruktúra projektu

Štruktúra projektu definuje tri prioritné množiny oblastí riešenia, ktorých spojením sa doieli kompletne riešenie projektu ako celku. Sú to oblasti:

- *Hardvérová časť – fyzikálny model* definujúca elektronickú hardvérovú časť modelu zabezpečujúcu chod modelu a zber dát.
- *Programová časť – spracovanie dát* je oblasť obsahujúca programovú časť mikroprocesora, ktorý číta dáta zo snímačov zrýchlenia a ďalej ich spracováva a sprostredkuje v správnom formáte pre softvér nachádzajúci sa v osobnom počítači používateľa. Ten má za úlohu dáta spracovať a preniesť do programovej časti GUI.
- *Programová časť – GUI* je rozhraním človeka a stroja, z priemyselnej oblasti nazývajúca sa aj Human Machine Interface. Umožňuje riadiť všetky predošlé oblasti a zobrazuje výsledok meraní človeku prirodzeným spôsobom prostredníctvom grafov.

2.3 Štvrtinový matematický model vozidla

Štvrtinový matematický model vozidla zobrazený na Obr. 9 je približuje matematické vzťahy medzi neodpruženou a odpruženou hmotou, tlmičom, pružinou a pneumatikou. Každé z vymenovaných telies sa podieľa na výslednom správaní sa nápravy vozidla.



Obr. 9 Matematický model pruženia automobilu

Odpruženou hmotou rozumieme rám alebo karosériu vozidla. K rámu vozidla je pripevnená tlmič s pružinou. Neodpružená hmota je uloženie kolesa a všetky diely nachádzajúce sa medzi tlmičom s pružinou a kolesom. Medzi samotnou vozovkou a pružením automobilu sa nachádza ešte pneumatika, ktorá na základe svojich vlastností tvorí ešte jednu vrstvu pruženia.

Označenia v schéme:

- x_0 – poloha vozovky
- x_1 – poloha neodpruženej hmoty
- x_2 – poloha odpruženej hmoty (vozidla)
- k_1 – konštanta tuhosti pneumatiky
- k_2 – konštanta tuhosti pružiny nápravy
- c_2 – konštanta tlmenia tlmiča nápravy
- m_1 – hmotnosť neodpruženej hmoty
- m_2 – hmotnosť odpruženej hmoty

Pre uvedené veličiny a matematický model pruženia automobilu platí pohybová rovnica:

$$m_2 \ddot{x}_2 + c_2(\dot{x}_2 - \dot{x}_1) + k_2(x_2 - x_1) = 0 \quad (1)$$

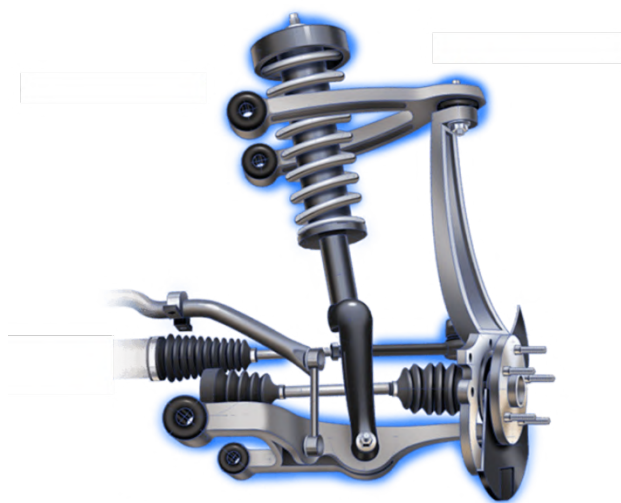
$$m_1 \ddot{x}_1 + c_2(\dot{x}_1 - \dot{x}_2) + k_2(x_1 - x_2) + k_1(x_1 - x) = 0 \quad (2)$$

3 Návrh a zhotovenie modelu

Zhotovenému výsledku v podobe fyzikálneho modelu predchádzalo zbieranie inšpirácií, kreslenie návrhov vzhľadu modelu a vyhľadávanie použiteľných materiálov vzhľadom na dostupné možnosti odpruženia.

3.1 Definovanie vzhľadu fyzikálneho modelu

Vytvorenie výsledného vzhľadu sa odvíjalo od nazbieraných inšpirácií ako aj od kreslenia niekoľkých návrhov. Dôležitým míľnikom bol aj fakt, že sme matematický model lichobežníkovej nápravy využívali aj v experimentoch, ktoré boli súčasťou zadania k predmetu CAEMS – CAE mechatronických systémov. Hlavnou inšpiráciou bol Obr. 10, na ktorom sú vhodne znázornené prvky nápravy ako aj ich vzájomné pospájanie do jedného systému.

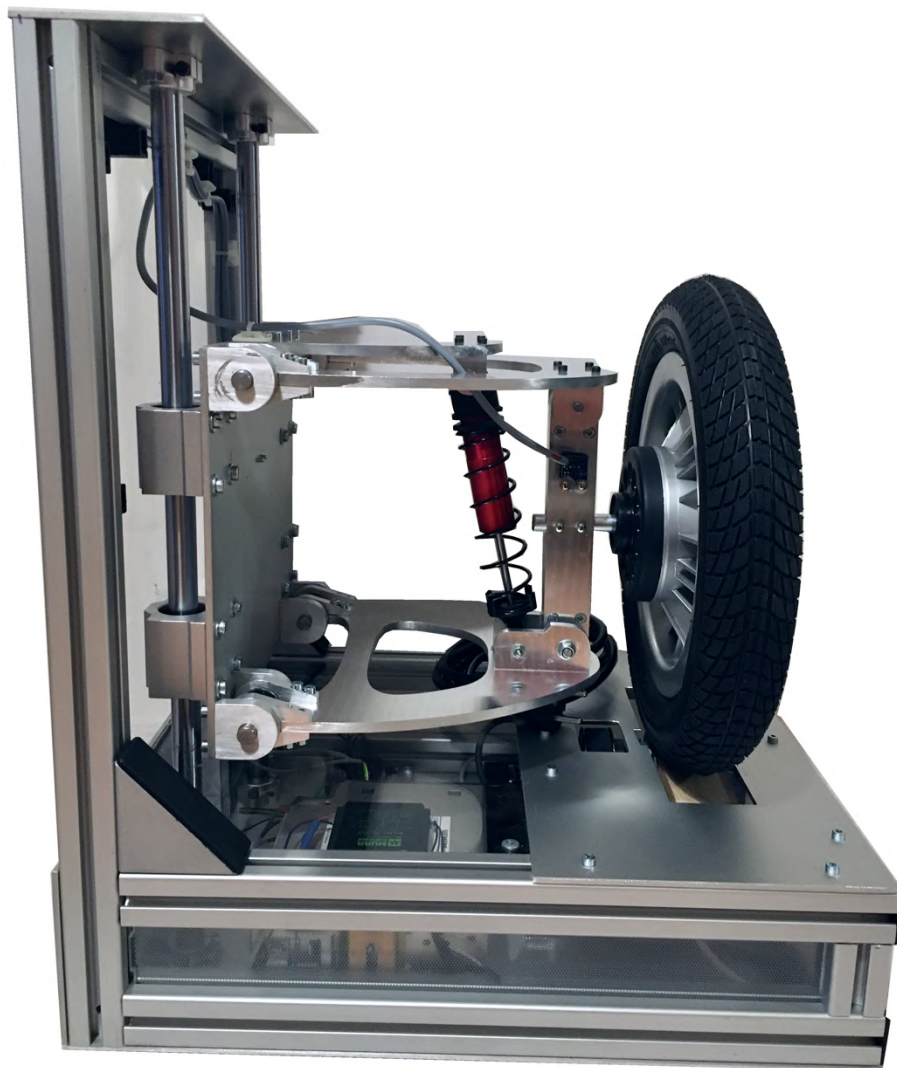


Obr. 10 Lichobežníková náprava - inšpirácia [10]

Uvažovaným rozdielom medzi reálnou lichobežníkovou nápravou a nami vytvoreným modelom bolo zníženie počtu smerov otáčania sa kolesa. Nakoľko sme predpokladali, že fyzikálny model zobrazujúci len jednu štvrtinu automobilu, nepotrebuje natáčanie kolies do smeru jazdy, rozhodli sme sa, že znížime počet stupňov voľnosti kolesa. Samotný vzhľad nám do výraznej miery ovplyvnil tlmič s pružinou, ktorý sme vyberali na základe dostupnosti a približnej odhadovanej hmotnosti systému a to približne 2 kg.

3.2 Zhotovenie modelu

Návrh fyzikálneho modelu vychádzajúceho z predstavy o jeho vzhľade opísanej v predchádzajúcej kapitole sme chceli v určitej miere prispôbiť laboratórnym podmienkam, v ktorých bude tento fyzikálny model umiestnený. Na Obr. 11 môžeme vidieť nami vytvorený fyzikálny model. Hlavnú nosnú konštrukciu tvoria hliníkové prefabrikované profily s lineárnymi nepodopretými tyčami. Na týchto je umiestnená lichobežníková náprava, ktorej vertikálny pohyb je zabezpečený práve lineárnym vedením s klznými vozíkmi. Klzné vozíky a pevne pripevnená hliníková doska predstavujú karosériu vozidla, čiže odpruženú hmotu. Na druhej strane koleso a časť držiaca koleso predstavuje neodpruženú hmotu. Na oboch hmotách sa nachádza snímač zrýchlenia, pomocou ktorého vyhodnocujeme vlastnosti sústavy.



Obr. 11 Fyzikálny model

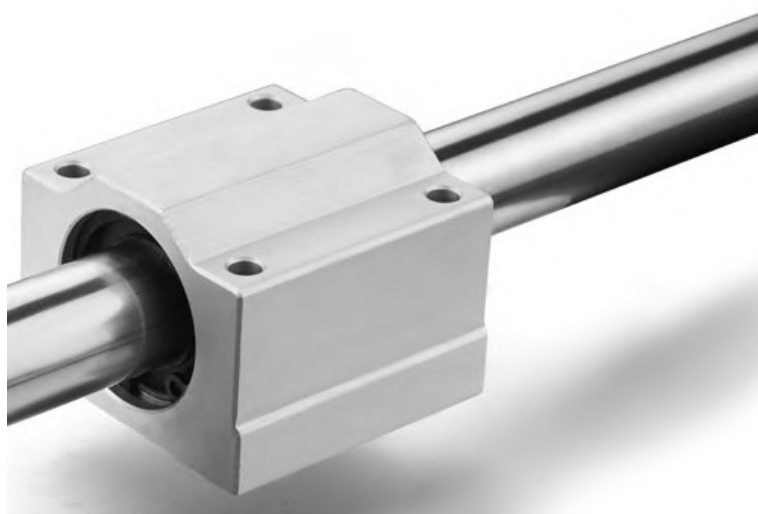
3.2.1 Nosná konštrukcia fyzikálneho modelu

Je potrebné, aby bola lichobežníková náprava spojená so stojanom tak, aby jej bol umožnený vertikálny pohyb. Tento pohyb vyvolaný ako reakcia na zmenu polohy kolesa symbolizuje zmenu polohy časti nápravy automobilu a teda náklon automobilu vplyvom nerovností vozovky. Na obmedzenie pohybu modelu sme sa rozhodli použiť guľatinové nepodopreté brúsené tyče, ktorých vlastnosti sa nachádzajú v Tabuľka 1.

Materiál	Akosť	Povrchová úprava	Tolerancia	Tvrdosť
ocel'	C45	Brúsená a povrchovo kalená	g6	HRC 58 ~ 62

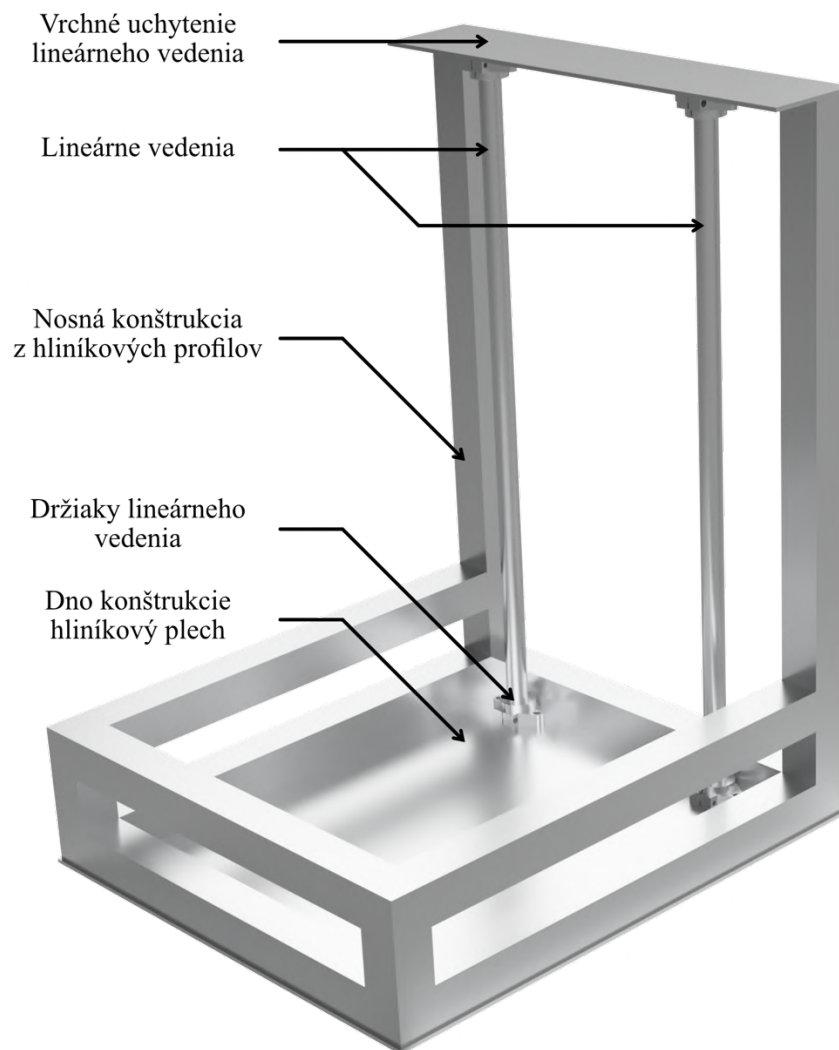
Tabuľka 1. Vlastnosti guľatinovej tyče

Aby bol spôsob uchytenia kolísajúceho modelu nápravy stabilný a v rovnovážnej polohe, je potrebné použiť dve guľatinové tyče, ku ktorým bude odpružená hmota modelu uchytená tak, aby bola schopná pohybu vo vertikálnom smere (v smere gravitačného zrýchlenia). Vertikálny klzavý pohyb po nepodopretých tyčiach sme zabezpečili vozíkmi pre nepodopreté tyče. V lineárnych klzných vozíkoch sa nachádzajú ložiská znižujúce trenie pri pohybe po lineárnom vedení. Lineárne vedenia s klznými vozíkmi sa využívajú napríklad ako pohybová sústava CNC strojov. Pre naše účely sme sa rozhodli použiť nepodopreté tyče priemeru 16mm s dĺžkou 500 mm. K týmto tyčiam sme z internetového obchodu cncprislusenstvo.sk objednali aj klzné vozíky pre lineárne vedenia a držiaky nepodopretých tyčí, vďaka ktorým bolo možné lineárne vedenia pevne spojiť so zvyškom nosnej konštrukcie fyzikálneho modelu. Pre lepšiu predstavu sme sa rozhodli doplniť Obr. 12 lineárneho vedenia a klzného vozíka.



Obr. 12 Lineárne vedenie s vozíkom a držiakom [11]

Hlavná nosná konštrukcia stojana je vytvorená z hliníkových profilov „combi“. Hliníkové profily majú vysokú pevnosť a vhodné spracovanie pre lepšie uchytenie zvyšných komponentov modelu. Počítali sme s možnosťou umiestnenia elektroniky a budenia od vozovky v spodnej časti stojanu a z toho dôvodu sme v dolnej časti stojanu vytvorili priestor s výškou 100 mm zdvojením konštrukčnej podstavy. Samotné dno konštrukcie stojanu je tvorené hliníkovým plechom s hrúbkou 2 mm. Hlavná nosná konštrukcia zobrazená na Obr. 13 má za úlohu niesť váhu celého fyzikálneho modelu a preto bolo pre nás dôležité, aby bola pevná a stabilná. Spomenutý obrázok obsahuje aj označenie jednotlivých komponentov konštrukcie a hlavné rozmery konštrukcie. Na spájanie hliníkových profilov sa využíva centrálna spojka uchytená skrutkou M5 k stredu profilu. Prepojenie hliníkových profilov a hliníkových plechov, prípadne ostatných dielov spojených s konštrukciou je vytvorené prostredníctvom matíc vložených v drážkach hliníkových profilov.



Obr. 13 Nosná konštrukcia fyzikálneho modelu

V tabuľke (Tabuľka 2) sme popísali základné vlastnosti použitých hliníkových profilov.

Hmotnosť [kg/m]	Prierezový modul [cm ³]	Moment zotrvačnosti [cm ⁴]	Drážka [mm]	Rozmery prierezu [mm]
0,92	W _x = 1,98 W _y = 1,92	I _x = 3,02 I _y = 2,88	6	30x30

Tabuľka 2 Vlastnosti hliníkových profilov

3.2.2 Model nápravy – Lichobežníková náprava

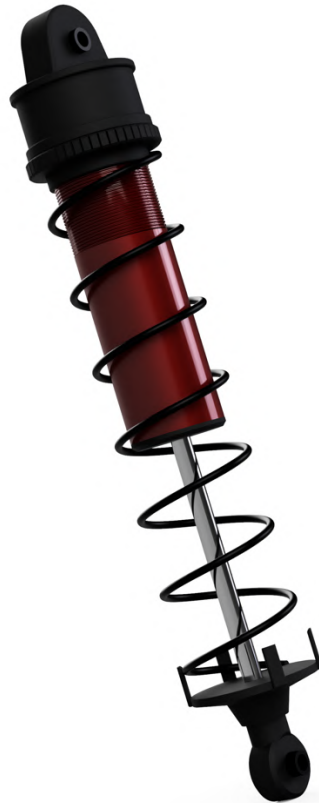
Ako sme už spomínali, na základe dohody s vedúcim práce sa táto diplomová práca venuje zhotoveniu štvrtinového fyzikálneho modelu lichobežníkovej nápravy automobilu. Pracovali sme na vytvorení modelu systému odpruženia nachádzajúcim sa za jedným kolesom automobilu. Nami vytvorená nosná konštrukcia umožňuje realizovať pohyb celej sústavy modelu lichobežníkovej nápravy. S týmto faktom sme už počítali pri návrhu a zhotovení lichobežníkovej nápravy. Vieme, že tento typ musí obsahovať nasledovné komponenty:

- Horné rameno
- Spodné rameno
- Uchytenie tlmiča s pružinou
- Tlmič s pružinou
- Uchytenie kolesa
- Odpružená hmota

Na základe týchto požiadaviek a už spomínanej inšpirácie sme pomocou CAD programu Fusion 360 navrhli model lichobežníkovej nápravy. Počas navrhovania sme museli klásť dôraz na hmotnosť celej nápravy. Hmotnosť celej nápravy nás zaujímala z dôvodu špecifikácie tlmiča s pružinou. Počas počiatočných príprav na tvorbu diplomovej práce sme sa pokúšali nájsť vhodný tlmič pre náš model. Pri hľadaní tlmiča s pružinou sme vzhľadom na vlastnosti pripravených materiálov (hliník ALU 7075) odhadovali hmotnosť sústavy pruženia na približne 2 kg. Odhadovaná hmotnosť vznikla na základe určenia približnej veľkosti celého modelu a to 500x400 mm. Túto veľkosť sme si určili, aby bolo možné s modelom manipulovať, prípadne, aby mohol model slúžiť v budúcnosti ako učebná pomôcka.

Na základe približných parametrov sme začali s vyhľadávaním správneho tlmiča s pružinou. Zistili sme, že použitie akejkoľvek pružiny z oblasti bicyklov, či už detských alebo normálnej veľkosti, nie je možné. Nie je to možné z dôvodu, že takáto pružina nesie váhu človeka.

V prípade dieťaťa ide o váhu v rozmedzí 15 – 35 kg, no v prípade dospeljej osoby je táto váha už nad 35 kg. Tieto hodnoty rádovo prekračujú nami vytýčené podmienky vlastností pružiny a tlmiča. Ďalšou možnosťou boli pružiny z oblasti modelov áut a RC modelov. V tejto oblasti sme vyhládávali nadštandardne veľký tlmič s pružinou, nakoľko modeli áut málokedy dosahujú veľkých rozmerov a hmotností. V tejto oblasti sme našli olejový tlmič a pružiny od spoločnosti ARRMA, ktorá vyrába RC modely áut. Nami nájdený tlmič s pružinou sme znázornili na Obr. 14.



Obr. 14 Pružina s tlmičom

Tento tlmič s pružinou sa využíva v RC modeloch áut ARRMA Kraton na zadnej náprave tohto modelu. Pre jeho vhodné vlastnosti opísané v tabuľke (Tabuľka 3) sme sa rozhodli využiť tento tlmič v našej práci. Je možné ho zakúpiť aj ako náhradný diel, čo proces jeho zakúpenia uľahčilo.

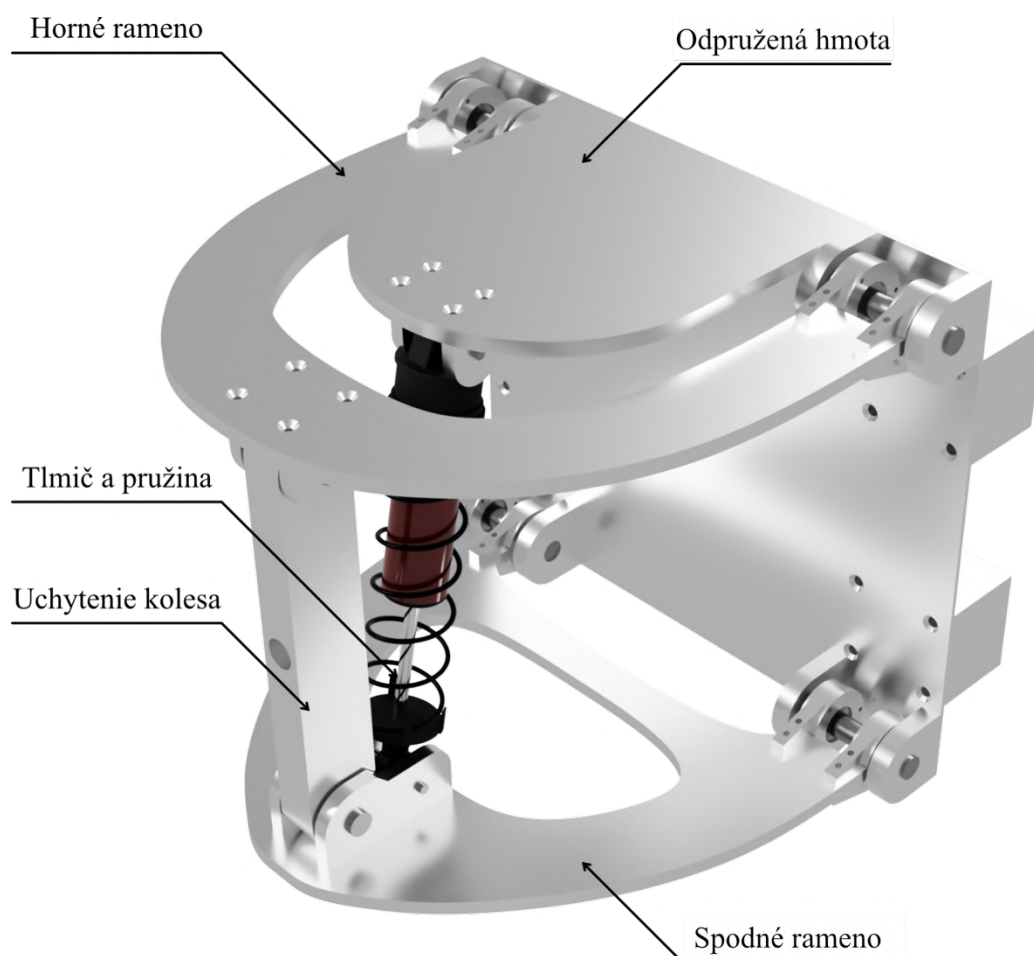
Upevnenie tlmiča je možné realizovať predpripravenými bodmi uchytenia na oboch koncoch tlmiča, ktoré sú navyše otáčavé do všetkých smerov. To znamená, že nie je potrebné, aby sme uchytenie tlmiča realizovali ložiskami po oboch jeho koncoch. Pre zníženie trenia sme na oba otočné body uchytenia naniesli plastické mazivo, ktoré zabezpečuje lepšie klzné vlastnosti súčiastky.

Dĺžka pružiny [mm]	Vnútny priemer [mm]	Vonkajší priemer [mm]	Tuhosť [N/cm]	Dĺžka tlmiča [mm]
130	27	32	22,1	190

Tabuľka 3 Vlastnosti tlmiča a pružiny

Najdôležitejšími parametrami sú tuhosť pružiny a tlmenie tlmiča. Zatiaľ čo prvý parameter výrobca udáva, druhý parameter sme museli zistiť na základe porovnania simulácie s výsledkami fyzikálneho modelu. Tuhosť pružiny 22,1 N/cm znamená, že sa dĺžka pružiny deformuje (skrúti) o dĺžku 1cm pri každej záťaži 2,2 kg. Z toho vyplýva, že ak požadujeme približnú hmotnosť našej sústavy 2 kg, tak pri stave pokoja sústavy bude deformácia stlačením pružiny približne 1 cm.

Po dodaní tlmičov sme vytvorili návrh modelu lichobežníkovej nápravy na základe poznatkov o fyzikálnych vlastnostiach pružiny. Pri vytváraní návrhu sme dbali na hmotnosti jednotlivých dielov. Výsledný návrh modelu nápravy je zobrazený na Obr. 15.



Obr. 15 3D model lichobežníkovej nápravy

Horné rameno

Je jedným z dvoch ramien, ktoré sú základnou charakteristikou lichobežníkovej nápravy. K hornému ramenu sa upevňuje komponent uchytenia kolesa. V reálnom automobile je tu guľové uloženie, ktoré umožňuje kolesu pohyb vo všetkých smeroch. Nakoľko sme nášmu modelu odobrali natáčanie kolesa do smeru jazdy, stačí nám otáčavý pohyb kolesa. Na druhom konci je horné rameno ložiskami pripevnené k pohyblivej časti. Ložiská opäť zabezpečujú otáčavý pohyb horného ramena, ktoré sa vychyluje vplyvom nerovností na vozovke, ktoré spôsobujú vychyľovanie kolesa. Horné rameno sa skladá zo samostatného kusu hliníka ALU 7075 hrubého 6 mm. Samotný tvar horného ramena sme vyrobili predvrtaním dier tak, že tieto diery kopírovali tvar ramena. Potom sme pílkou na železo a pilníkom odstránili zvyšné spoje medzi ramenom a zvyškom materiálu. Po oddelení ramena od zvyšku materiálu sme rameno obrúsili pilníkom, čím sme dosiahli jeho výsledný tvar. Takmer všetky diely vznikali rovnakým spôsobom, preto spôsob výroby už nebudeme pri týchto dieloch opakovať. V časti pre uchytenie kolesa je k ramenu kolmo priskrutkovaná dvojica hliníkových častí z rovnakého materiálu ako je rameno. Dvojica hliníkových častí slúži na spojenie horného ramena a uchytenia kolesa. V strede týchto prvkov je vyvrtaná diera s priemerom 8 mm, ktorou prechádza skrutka, ktorou je uchytenie kolesa spojené s dvojicou prvkov a teda aj s horným ramenom.

Spodné rameno

Náprotivným prvkom k hornému ramenu je spodné rameno. Plní rovnakú funkciu ako horné rameno s jedným zásadným rozdielom. K dolnému ramenu je upevnená pružina s tlmičom. Pružina s tlmičom je pripevnená cez svoj otočný bod pripevnenia, ktorý jej dovoľuje pohyb do všetkých smerov. Nakoľko je konštrukcia nášho modelu obmedzená len na vertikálny smer pohybu, nepriamo tak obmedzujeme pohyb aj tlmiču. Tlmič ako aj uchytenie kolesa je k dolnému ramenu pripevnené pomocou dvojice hliníkových komponentov, ktoré sú priskrutkované k dolnému ramenu. Tieto komponenty sú obdobou prvkov, ktoré sme spomínali pri hornom ramene. Tie majú však iný tvar práve kvôli spájaniu aj tlmiča s dolným ramenom.. Tvar týchto komponentov je upravený tak, aby nedochádzalo k stretu s tlmičom, čo by spôsobovalo zastavenie chodu nápravy.

Uchytenie kolesa

Je spájajúcim prvkom horného a dolného ramena. Ako sme už spomínali, je pripevnené tak, aby umožňovalo pohyb vo vertikálnom smere tým, že využíva ložiskové spojenie ramien s pohyblivou časťou mechanizmu. V strede uchytenia kolesa je vytvorená diera s priemerom 12 mm, do ktorej bude pevne uložený hriadeľ kolesa. Komponent uchytenia kolesa sme vyrobili z troch hliníkových dosiek s hrúbkou 6 mm. Celý diel uchytenia kolesa má hrúbku 18 mm a šírku 30 mm. Hriadeľ kolesa s priemerom 12 mm je vsunutý do diery uprostred uchytenia a dotlačený k stene uchytenia kolesa dvoma skrutkami M3. Takýmto spôsobom sme zaistili hriadeľ kolesa a znemožnili sme jeho samovoľný pohyb.

Odpružená hmota

Odpružená hmota spája ramená lichobežníkovej nápravy. Je ekvivalentom štvrtiny automobilu. Z tejto časti taktiež vyčnieva doskové rameno, ktoré slúži na upevnenie druhej polovice tlmiča. Toto doskové rameno je k pohyblivej časti pripevnené hliníkovým L-profilom. Hliníkový L-profil má rovnakú dĺžku ako prvok upevnenia tlmiča a to 140 mm. K tomuto prvku je rovnako pripevnený pomocou štyroch skrutiek.


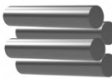


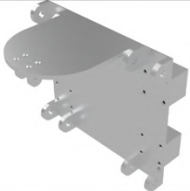
Hlavným komponentom odpruženej hmoty je hliníková doska hrúbky 4 mm. Na túto dosku sú zo zadnej strany priskrutkované klzné vozíky ktoré slúžia na pohyb po lineárnom vedení. K hliníkovej zadnej doske sú tiež z prednej strany pevne priskrutkované diely, ktoré vytvárajú otáčavý spoj s ramenami. Tieto diely sú podobné dielom, ktoré sme spomínali pri hornom aj dolnom ramene nápravy. Považujeme za vhodné priblížiť vzhľad dielov pomocou Obr. 16, ktorý znázorňuje diely prichytenia horného ramena nápravy k uchyteniu kolesa.



Obr. 16 Diel spájajúci pevné a otáčavé prvky

Hmotnosti komponentov

Geometrický a matematický model vytvárame vo vzájomnej interakcii. Nie je možné, aby sme v dielni so základnou výbavou dokázali vytvoriť geometrický model s presnosťou na mikrometre. Preto sme po vyrobení konštrukčných dielov lichobežníkovej nápravy zmerali hmotnosť každého jedného dielu. Tieto hmotnosti sme použili pri simulácii matematického modelu. Takto sme docieli presnejšiu matematickú podobnosť oboch modelov a výsledky simulácie sme dokázali viac priblížiť výsledkom geometrického modelu. Hmotnosti konštrukčných dielov sme zaznamenali do tabuľky (Tabuľka 4).

Názov	Obrázok	Hmotnosť
Uchytenie kolesa		0,270 [kg]
Čap (4x)		0,160 [kg]
Horné rameno		0,420 [kg]
Dolné rameno		0,540 [kg]
Odpružená hmota (automobil)		1,3 [kg]

Tabuľka 4 Hmotnosti konštrukčných prvkov lichobežníkovej nápravy

3.2.3 Hnacia sústava

Jednou z úloh zadania diplomovej práce je aj definovať spôsob budenia od vozovky. Pre náš model sme sa rozhodli budenie od vozovky definovať ako zjazd automobilu z obrubníka. Rozhodli sme sa tak z dôvodu, že pri zjazde automobilu z obrubníka vzniká veľký nárok na sústavu pruženia automobilu vyrovnáť takto spôsobenú nerovnosť. Z toho vyplýva možnosť pozorovať najbadateľnejšie zmeny zrýchlenia pri porovnaní zrýchlenia kolesa a zrýchlenia pohyblivej časti modelu. Naším úsilím bolo vytvoriť čo najdlhšiu dráhu voľného pádu kolesa. V predošlej časti sme si určili výšku základne nosnej konštrukcie 100 mm. Do tejto časti je potrebné inštalovať hnaciu sústavu, ktorá dokáže vytvoriť čo najväčšiu skokovú zmenu polohy kolesa. Definovali sme si niekoľko vlastností hnacej sústavy (budenia od vozovky), ktoré od nášho modelu vyžadujeme:

- a) Meranie má byť opakovateľné bez zásahu a zbytočnej prípravy
- b) Je potrebné, aby sústava fungovala bez fyzického pričinenia obsluhy
- c) Sústava musí zvládnuť zdvih modelu lichobežníkovej nápravy
- d) Voľný pád kolesa a nápravy nesmie sústavu poškodiť alebo zhoršiť jej vlastnosti
- e) Hnacia sústava má zabezpečiť čo najväčšiu skokovú zmenu polohy kolesa

Pre splnenia prvej podmienky opakovateľnosti merania sme sa rozhodli, že bude vhodné, ak ako budenie použijeme elektromotor, ktorého výstupom je otáčavý pohyb na hriadeľ. Čiastočne tým docielime aj vyriešenie druhej podmienky, keďže vhodným zostrojením pohybovej sústavy dokážeme vytvoriť systém, ktorý na svoj chod nepotrebuje pričinenie obsluhy. Takto zabezpečíme ochranu zdravia pri práci. Obsluha nebude musieť zasahovať do chodu stroja, čím sa znižuje riziko vzniku zranení. Aby sústava zvládla zdvih modelu, musí byť elektromotor dostatočne výkonný. Dôležitou podmienkou je podmienka d). V prípade, že by voľný pád kolesa poškodil hnaciu sústavu alebo by zhoršil jej vlastnosti, ďalšie uskutočnené merania by neboli korektné a výsledné hodnoty by boli skreslené. K poškodeniu by mohlo dôjsť v prípade, ak by koleso pravidelným voľným pádom dopadalo priamo na hriadeľ elektromotora. Vlastnosti elektromotora by sa pri tomto neštandardnom pohybe zhoršovali nepravidelným opotrebovaním ložísk vo vnútri motora a následným vychýlením hriadeľa. Magnetické pole cievok motora by vzhľadom k novej polohe hriadeľa zhoršili krútiaci moment, čo by nepriaznivo ovplyvnilo chod hnacej sústavy, prípadne by hnaciu sústavu trvale poškodilo. Preto sme sa rozhodli, že hriadeľ elektromotora spojíme remenicami a remeňom s hriadeľom, ktorý bude schopný znášať pravidelné zaťaženie vo forme pádu kolesa. Zvolili

sme si dve remenice s rôznym počtom zubov. Remenica na hriadeli elektromotora má 12 zubov a remenica na druhom hriadeli mimo motora má 24 zubov. Takto zvolené remenice zabezpečia prenos rýchlosti a krútiaceho momentu motora v pomere 1:2. To znamená, že sme vytvorili systém s prevodom, kde za cenu nižšej rýchlosti na kontaktnom hriadeli dosiahneme dvojnásobný krútiaci moment. Vzhľadom k hmotnosti modelu nápravy je dôležité disponovať čo najväčším možným krútiacim momentom.

Samotný zdvih kola a celého modelu nápravy zabezpečíme vačkou. Vačka ako mechanická súčiastka zabezpečí premenu otočného pohybu hriadeľa na vratný priamočiary pohyb. Táto súčiastka je v mechanických systémoch často využívaná práve pre svoju funkciu zmeny rotačného pohybu na priamočiary. Celú nami vytvorenú hnaciu sústavu sme znázornili na nasledovnom Obr. 17.



Obr. 17 Hnacia sústava fyzikálneho modelu

Na obrázku môžeme vidieť celý systém budenia. Ako hlavný hriadeľ sme použili oceľovú tyč s priemerom 12mm, ktorá má na oboch koncoch priemer zmenšený na 10 mm. Priemer na koncoch hriadeľa sme zmenšili z dôvodu uloženia hriadeľa do ložísk, ktorými znížime trenie pri rotačnom pohybe. V strede hriadeľa je umiestnená vačka. Tú sme vytvorili na základe našich výpočtov a simulácií tak, aby sa zmestila do podstavy nosnej konštrukcie a zároveň zabezpečila čo najväčšiu zmenu polohy kola. Efektívna hodnota zmeny polohy kola je tak 70 mm. Na hlavnom hriadeli sa ešte nachádza spomínaná remenica s počtom zubov 24. Obe

remenicie ako aj remeň sme zakúpili. Remeň má dĺžku 225 mm. Pri spracovaní jednotlivých rozmerov sme došli k záveru, že hlavný hriadeľ a hriadeľ motora musia byť vo vzájomnej vzdialenosti 67,5 mm, aby bol remeň dostatočne napnutý a nedochádzalo k jeho prešmykovaní.

3.2.4 Koleso a estetické prvky modelu

Posledným kusom potrebným na správny chod fyzikálneho modelu je koleso. Vzhľadom na rozmery modelu lichobežníkovej nápravy bolo našou podmienkou, aby bol priemer kolesa najmenej 300 mm. Ideálna (no nie povinná) hrúbka kolesa sa pohybuje v rozmedzí 25 – 50 mm. Aby sme čo najpresnejšie kopírovali reálny automobil, ideálne by bolo koleso s možnosťou jeho hustenia vzduchom. Všetky vlastnosti dokonale spĺňalo koleso predávané ako náhradný diel ku kočíkom. Koleso, ktoré sa nám podarilo získať má priemer 300 mm, šírku 50 mm a je možné pneumatiku hustiť vzduchom. Taktiež má hriadeľ, ktorý je možné pevne uchytiť, keďže koleso obsahuje ložiská. Potrebným úkonom bolo skrátenie hriadeľa na 50 mm, pretože jeho pôvodná dĺžka zasahovala do pružiny a znemožňovala chod modelu.



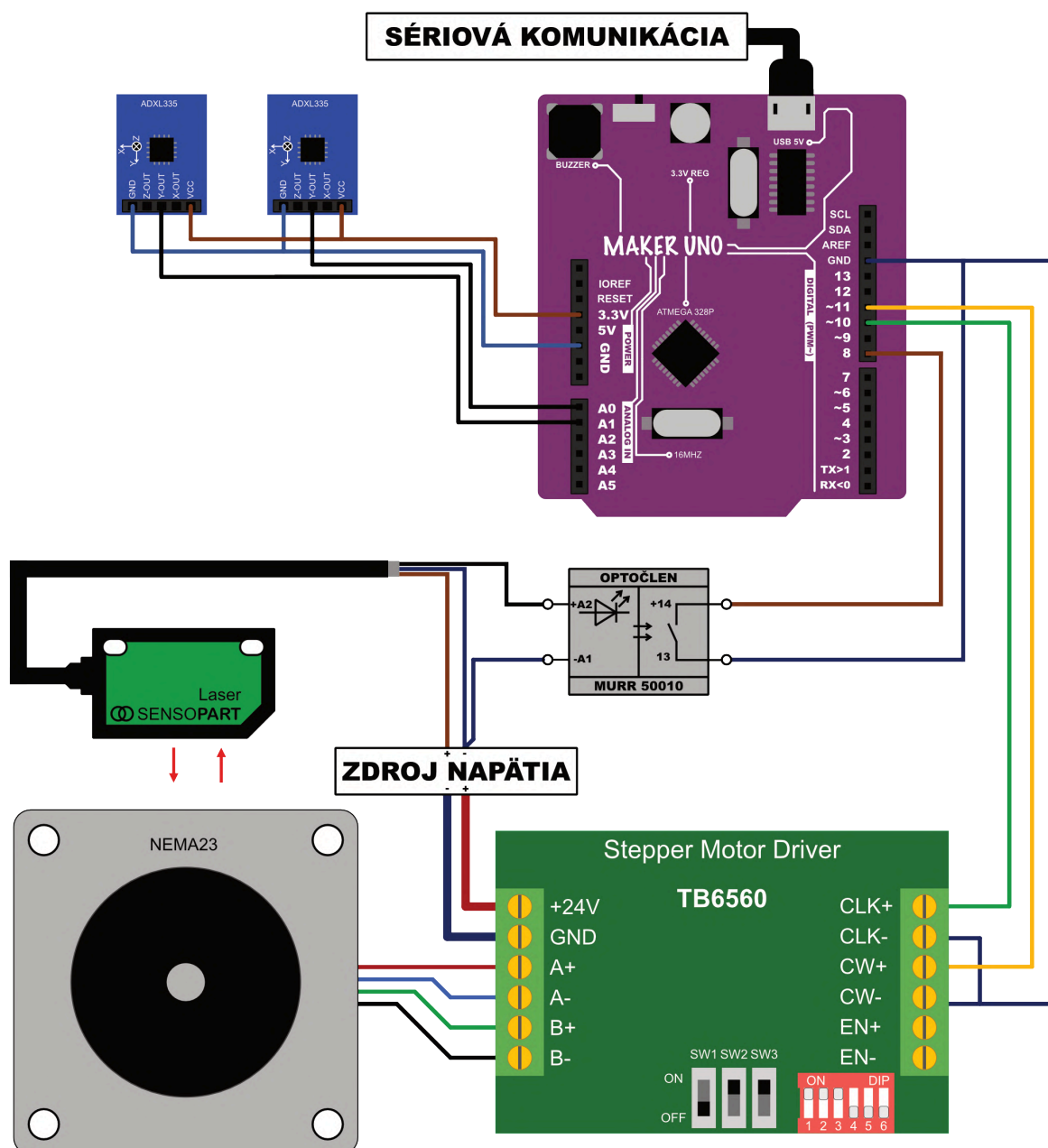
Obr. 18 Koleso

Estetickými prvkami modelu označujeme hliníkové a plexisklové kryty, ktoré dotvárajú výsledný vzhľad modelu, nemajú vplyv na funkčnosť modelu a zabezpečujú elektronické prvky pred nežiaducim dotykom obsluhy. Hliníkovým krytom sme zabezpečili hnaciu sústavu a konektor pre sieťovú prípojku 230 V AC. Ostatnú elektroniku a zadnú stenu nosnej konštrukcie sme zakryli plexisklom, aby bolo možné vidieť komponenty modelu aj elektroniky.

4 Elektronika systému

Jedným z našich cieľov bolo vytvoriť systém ovládania modelu schopný pracovať v uzavretej slučke bez zbytočného zásahu ľudí. Zvolili sme princíp ovládania modelu mikropočítačom ku ktorému sú pripojené zvyšné komponenty ovládania a merania.

4.1 Schéma zapojenia



Obr. 19 Schéma zapojenia elektroniky

Schéma zapojenia exaktne definuje jednotlivé elektronické komponenty a ich vzájomné prepojenie, ktoré sme v riadení a meraní fyzikálneho modelu použili. Riadenie modelu je možné rozdeliť do troch logických celkov a tým sú:

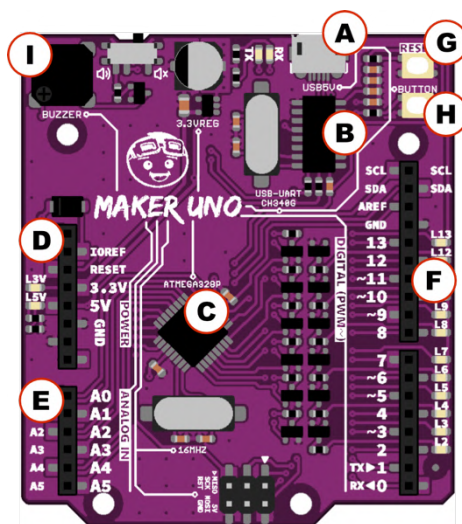
- *Riadenie*, ktoré riadi nasledovné dva celky a zároveň komunikuje s osobným počítačom
- *Ovládanie pohonu*, ktoré má za úlohu simulovať budenie od vozovky
- *Meranie zrýchlenia*, ktoré je realizované dvoma snímačmi zrýchlenia

4.2 Riadenie

Riadenie fyzikálneho modelu má za úlohu ovládať elektronické komponenty modelu a prijímať dáta zo snímačov zrýchlenia. Druhou nie menej podstatnou úlohou riadenia je odosielanie spracovaných údajov snímačov zrýchlenia do počítača, kde dochádza k ich druhotnému spracovaniu a vizualizácii. Ako hlavnú riadiacu jednotku sme si zvolili mikropočítač Arduino UNO, resp., jeho klon Cytron Maker UNO. Tento jednodoskový mikropočítač je vhodným pre náš projekt hlavne kvôli prítomnosti analógových vstupov, ktoré sme využili na získavanie analógových signálov zo snímačov zrýchlenia.

4.2.1 Mikropočítač Maker UNO

V prvom rade je vhodné, aby sme si opísali základné prvky mikropočítača Cytron Maker UNO. Pre prehľadnosť popisu sme vytvorili tabuľku (Tabuľka 5) s vysvetlivkami k Obr. 20.



Obr. 20 Cytron Maker UNO [15]

Pozícia	Vysvetlivka
A	USB Micro B konektor slúžiaci ako napájanie dosky a sériová komunikácia
B	USB-Sériový IC konvertor CH340G. Konvertuje USB dáta na sériové dáta.
C	Hlavný čip ATMEGA328P
D	Konektory IOREF, RESET, napájanie 3,3V a 5V ako aj konektor uzemnenia
E	Konektory vstupných analógových údajov
F	Vstupno-výstupné digitálne konektory a konektory pre komunikáciu
G	Reset tlačidlo
H	Programovateľné tlačidlo pripojené ku konektoru 2 (ako input_pullup)
I	Bzučiak pripojený ku konektoru 8

Tabuľka 5 Popis prvkov mikropočítača Maker UNO

Micro USB konektor využívame v našom projekte na napájanie modulu Maker UNO a zároveň na prenos dát sériovou linkou medzi počítačom a jednodoskovým mikropočítačom. Veľmi dobrou vlastnosťou mikropočítača Maker UNO sú LED indikátory pri každom digitálnom vstupno-výstupnom porte. Indikujú stav daného portu aj bez programovania. Dva digitálne porty využívame ako výstupné na riadenie krokového motora, respektíve ovládača krokového motora TB6560. Jeden digitálny port sme nastavili ako INPUT_PULLUP a pomocou tohto portu sledujeme stav laserového snímača. Komponent ovládača motora a laserový snímač polohy zdieľajú spoločné uzemnenie. V prípade analógových portov využívame dva analógové vstupy na prijímanie hodnôt zo snímačov zrýchlenia. Snímače zrýchlenia napájame napätím 3,3 V a uzemnenie snímačov je napojené na uzemnenie pri analógových vstupoch.

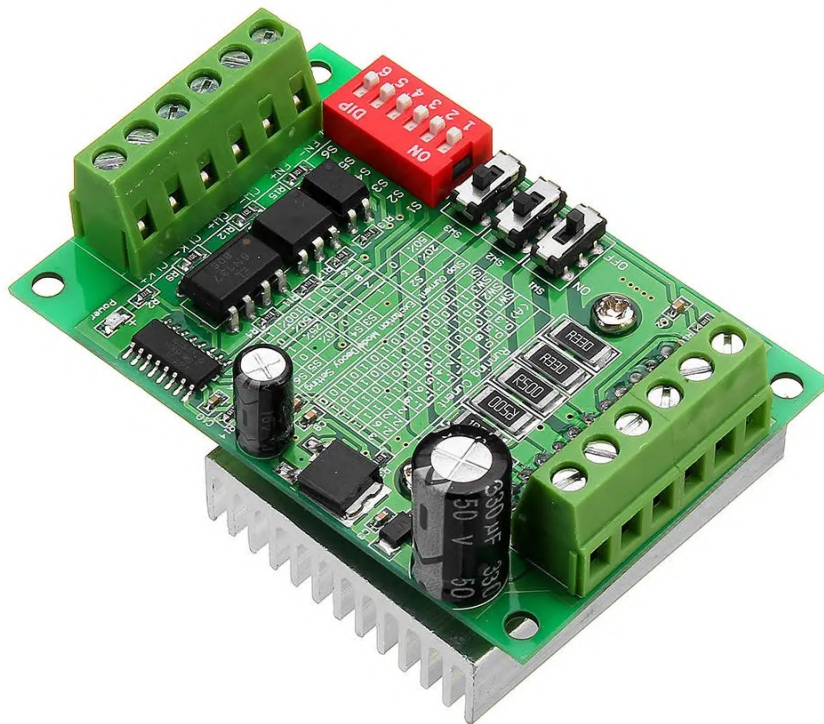
4.2.2 Zapojenie

Port	Popis
A0	Analógový vstup pre hodnoty zo snímača zrýchlenia odpruženej hmoty
A1	Analógový vstup pre hodnoty zo snímača zrýchlenia neodpruženej hmoty
8	Digitálny vstup na sledovanie stavu laserového snímača polohy
10	Digitálny výstup udávajúci takt krokovania pre krokového motora
11	Digitálny výstup smeru krokového motora

Tabuľka 6 Zapojenie portov mikropočítača Maker UNO

4.3 Hnacia sústava

Hnaciú sústavu fyzikálneho modelu tvorí hneď niekoľko elektronických komponentov. Prvým je ovládač krokového motora TB6560. TB6560 je samotný čip, ktorý je hlavným riadiacim čipom krokového motora. My sme zakúpili už hotovú dosku plošných spojov, ktorá nesie rovnaké označenie TB6560.



Obr. 21 Ovládač krokového motora TB6560 [12]

Ovládač TB6560 sme si vybrali kvôli jeho prúdovému rozsahu. Dokáže dodávať nominálny prúd až do 3 A a špička prúdu môže dosiahnuť hodnotu maximálne 3,5 A. Ovládač disponuje niekoľkými prepínačmi ako je možné vidieť na Obr. 21. Prepínače slúžia na nastavenie vlastností ovládača. Nastavuje sa napríklad maximálny výstupný prúd alebo presnosť krokového motora prerozdelením krokov. Samotný čip sa nachádza na dolnej strane ovládača a je na ňom umiestnený chladič. Na jednej strane ovládača sú svorky pre vstupné signály a na druhej sú výstupy, ktoré sa pripájajú priamo k cievkam krokového motora. Nastavenia výstupného prúdu dostupné zmenou prepínačov sme uviedli v tabuľke (Tabuľka 7).

Výstupný prúd [A]	Stav prepínačov			
	SW1	SW2	SW3	S1
0,3	0	0	1	1
0,5	0	0	1	0
0,8	0	1	0	1
1	0	1	0	0
1,1	0	1	1	1
1,2	1	0	0	1
1,4	0	1	1	0
1,5	1	0	1	1
1,6	1	0	0	0
1,9	1	1	0	1
2	1	0	1	0
2,2	1	1	1	1
2,6	1	1	0	0
3	1	1	1	0

Tabuľka 7 Výstupný prúd TB6560

Ovládač krokového motora ponúka široké spektrum možností nastavenia výstupného prúdu. My sme použili najvyššiu hodnotu 3 A, pretože maximálny prúd, ktorý dokáže náš krokový motor spotrebovať je 2,8 A. Ďalšie nastavenie, ktoré máme k dispozícii je nastavenie režimu budenia alebo aj rozdelenie krokov. Potom je tu nastavenie prúdu zastavenia, schopnosti brzdenia krokového motora a nastavenie tlmenia odozvy tranzistorov v čipe TB6560.

Režim budenia			Nastavenie tlmenia			Zastavovací prúd	
	S3	S4	[%]	S5	S6	[%]	S2
1	0	0	0	0	0	20	1
2	1	0	25	1	0	50	0
8	1	1	50	0	1		
16	0	1	100	1	1		

Tabuľka 8 Nastavenia režimu budenia, tlmenia a prúdu zastavenie TB6560

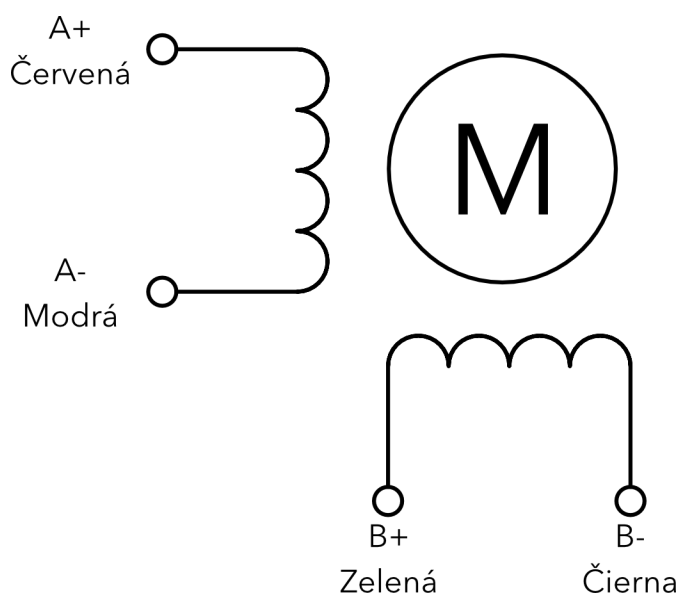
Nami zvolené nastavenia sme vyznačili oranžovou výplňou prislúchajúceho riadka tabuľky.

Ovládač krokového motora riadi krokový motor NEMA 23. Motor s parametrami uvedenými v tabuľke (Tabuľka 9) vyhovuje nášmu riešeniu, pretože disponuje veľkým výstupným prúdom a krútiacim momentom na hriadeľi.

NEMA23 - OK57STH56-2804A	
Veličina	Hodnota
Uhol kroku	1,8 [°]
Maximálny prúd	2,8 [A]
Odpor	0,9 [Ω]
Indukcia	2,5 [mH]
Hmotnosť	0,68 [kg]
Počet fáz	2

Tabuľka 9 Parametre krokového motora NEMA23


Motor sa pripája k ovládaču TB6560 pomocou štyroch vodičov. Vodiče sa k ovládaču pripájajú na výstupné svorky označené ako A+,A-,B+,B-. Schéma zapojenia na základe farieb vodičov sa nachádza na Obr. 22.



Obr. 22 Zapojenie vodičov krokového motora NEMA23

Ovládač motora zabezpečí po správnom nastavení plynulý chod krokového motora. Výsledný krútiaci moment, ktorý hnacia sústava dodáva je 3,6 N/m, čiže dvojnásobok krútiaceho momentu krokového motora. Krútiaci moment sme zvýšili už spomínaným prevodom 2:1. Chod krokového motora je zabezpečený vysielaním signálov z mikropočítača Maker UNO.

Naším cieľom bolo dosiahnuť, aby sa krokový motor počas vytlačania lichobežníkovej nápravy na pomyselný obrubník otáčal pomaly a v prípade, keď sa bude koleso nachádzať v najvyššom bode je potrebné rýchle otočenie. Rýchle otočenie musí uvoľniť priestor pod kolesom, aby sa docielil voľný pád geometrického modelu bez spomaľovania pádu dobiehajúcou vačkou. Na identifikáciu polohy vačky sme sa rozhodli použiť laserový snímač polohy Sensopart FT20-RLH. Tento senzor má možnosť nastavenia „Naučenia sa“ meranej vzdialenosti. Táto vlastnosť bola pre nás dôležitá, nakoľko sme sa snažili veľkosť modelu minimalizovať a vzdialenosť medzi umiestnením senzora a nosnou konštrukciou je len 100 mm. Schopnosť naučiť sa snímanú vzdialenosť sa nastavuje pomocou jediného tlačidla na senzore. Návod na nastavenie sa nachádza na zadnej strane senzora. Pre naučenie senzora je potrebné ho umiestniť v požadovanej vzdialenosti od snímaného predmetu, po dobu 3 sekúnd držať tlačidlo na senzore zatlačené, po pustení tlačidla sa musí snímaný predmet odobrať zo zorného poľa snímača a potvrdiť operáciu stlačením a podržaním tlačidla po dobu jednej sekundy. Rovnako sme postupovali pri nastavovaní vzdialenosti snímania vačky. Snímač sme uložili na časť uchytenia krokového motora tak, aby v prípade, keď bude model nápravy v najvyššom bode, zaznamenal vačku a odoslaním signálu do mikropočítača uviedol chod motora do rýchleho režimu

Veličina	Hodnota	Ilustratívny obrázok
Vzdialenosť snímania	30 . . . 100 [mm]	
Typ svetla	Laser, červený 650 [nm]	
Operačné napätie	10 . . . 30 [VDC]	
Výstupný prúd	100 [mA]	
Čas reakcie	500 [μs]	

Tabuľka 10 Parametre laserového senzora SENSOPART FT20-RLHD [13]

V tabuľke (Tabuľka 10) sme uviedli základné parametre laserového senzora Sensopart. Na napájanie snímača sme využili 24 V DC z napájacieho zdroja, ktorým napájame aj ovládač krokového motora TB6560. Aby sme mohli prepojiť výstup snímača s mikropočítačom, bolo potrebné medzi tieto dva komponenty umiestniť optočlen. Optočlenom sme zabezpečili galvanické oddelenie obvodov snímača a mikropočítača, čím sme ochránili mikropočítač pred zničením vplyvom vyššieho napätia na jeho svorkách. Optočlen sme zaradili do obvodu aj z dôvodu, že na vstup mikropočítača môžeme priviesť maximálne 5 V DC. Vstupný port mikropočítača nastavený ako INPUT_PULLUP sme priviedli na kladný pól optočlena

a záporný pól sme uzemnili. Ak snímač zaznamenal vačku v nami požadovanej polohe, zopnutím kontaktu aktivoval optočlen a došlo k uzemneniu výstupného portu mikropočítača, čím sme na jeho vstupe dosiahli hodnotu logickej jednotky.

4.4 Snímače zrýchlenia

Jedným z bodov zadania bolo použiť snímače zrýchlenia na vytvorenie záznamu o správaní sa geometrického modelu. Snímače zrýchlenia sú vhodné na meranie parametrov modelu aj vďaka svojej veľkosti. Rozhodli sme sa pre snímače zrýchlenia s analógovým výstupom nameraných hodnôt a to ADXL335. Tieto snímače snímajú zrýchlenie v troch osiach X, Y, Z. Každý snímač nesie označenie smeru osí. Jeden snímač zaznamenáva zrýchlenie neodpruženej hmoty a je umiestnený na uchytení kolesa. Druhý snímač zaznamenáva zrýchlenia odpruženej hmoty a je umiestnený na hlavnej doske medzi klznými vozíkmi. Snímače sme natočili tak, aby bola os snímačov Y v smere gravitačného zrýchlenia. Je dôležité, aby bol geometrický model počas merania umiestnený v rovnovážnej polohe. Na snímač je potrebné pripojiť napájacie napätie 3,3 V DC ako aj uzemnenie GND. Zo snímača využívame len jeden výstupný port pre os Y, ktorým snímač vysiela aktuálne zrýchlenie v smere osi Y. Výstupný rozsah je daný napájacím napätím a to 0 – 3,3 V DC. Tomuto rozsahu prislúcha aj typický rozsah merania zrýchlenia $\pm 3.6 \text{ g}$ ($\sim 35 \text{ m/s}^2$). Absolútnym maximom, ktorý je možné týmto snímačom zrýchlenia merať je $\pm 10 \text{ g}$, pri tejto hodnote však môže dôjsť k trvalému poškodeniu snímačov.

5 Riadiaci program

Riadiaci program tvoria dva samostatné celky. Prvou časťou je program bežiaci v mikropočítači a druhou časťou je program grafického používateľského rozhrania.

5.1 Program mikropočítača Maker UNO

Riadiaci program bežiaci kontinuálne v mikropočítači Maker UNO sme vytvorili tak, aby ovládal všetky komponenty elektroniky. To znamená, že týmto programom ovládame chod krokového motora, sledujeme stav laserového snímača polohy a zbierame analógové hodnoty snímačov zrýchlenia. Ďalšou úlohou mikropočítača Maker UNO je komunikácia s používateľským prostredím nachádzajúcim sa v osobnom počítači.

5.1.1 Premenné a funkcia „Setup“

Premenné programu mikropočítača sme sa rozhodli prezentovať formou tabuľky (Tabuľka 11).

Názov	Dátový typ	Popis	Začiatočná hodnota
serialCode	CHAR	Uchovanie prijatej hodnoty	„0“
serialCodeInt	INT	Celočíselná prijatá hodnota	0
motorSpeed	INT	Rýchlosť otáčok motora	1600
distance	INT	Prejdená vzdialenosť (krok)	0
i	INT	Iteračná premenná	0
Z1_axis	CONST INT	Priradenie analógového vstupu	A0
Z2_axis	CONST INT	Priradenie analógového vstupu	A1
z1_adc_value	INT	Analógová hodnota snímača	0
z2_adc_value	INT	Analógová hodnota snímača	0
z1_g_value	FLOAT	Hodnota snímača grav. zrých.	0
z2_g_value	FLOAT	Hodnota snímača grav. zrých.	0
dobaMerania	INT	Dĺžka merania	400
zmenaDobyMerania	BOOL	Premenná zmeny doby merania	false

Tabuľka 11 Premenné programu mikropočítača

Vo funkcii *Setup()* nastavujeme stavy portov (vstup, výstup). Digitálny port č.8 nastavený ako INPUT_PULLUP používame na sledovanie stavu laserového snímača polohy vačky. Porty

č.10 a č.11 sú porty na ovládanie chodu krokového motora (krok, smer). Vo funkcii *Setup()* sme nastavili aj šírku prenosového pásma (Baud Rate) sériovej komunikácie. Pôvodná nami nastavená hodnota bola 9600 bd. Sledovaním dĺžky času odoslania dát meraní sme ale zistili, že šírka pásma 9600 bd je nedostačujúca a spôsobuje pribrzdzenie chodu programu o 11 ms. Takéto omeškanie je neprijateľné, nakoľko pri tomto omeškaní nie je možné skĺbiť chod krokového motora s meraním. Preto sme rozšírili šírku pásma na 230400 Baud Rate. Pri takejto šírke pásma trvá odoslanie dát merania len približne 750 μ s. Ide o desaťnásobné zníženie doby odoslania údajov sériovou linkou. Zápis požadovanej šírky prenosového pásma a súčasné spustenie sériovej komunikácie sa vykonáva jedným príkazom *Serial.begin(230400)*.

5.1.2 Sériová komunikácia s GUI

Sériová komunikácia zabezpečuje prenos dát z merania a ovládanie fyzikálneho modelu prostredníctvom GUI. Mikropočítač v hlavnej nekonečnej slučke kontroluje dostupnosť dát na kanáli sériovej komunikácie. Aby bola sériová komunikácia medzi osobným počítačom a mikropočítačom spoľahlivá, vytvorili sme si určitú komunikačnú štruktúru, ktorá podlieha každému prenosu dát, týkajúcich sa nastavenia parametrov alebo spúšťania merania. Táto štruktúra spočíva v potvrdení komunikácie mikropočítačom. Nasledujúci Obr. 23 znázorňuje kód sériovej komunikácie v mikropočítači. Príkazom *Serial.available()* zisťujeme dostupnosť dát v zásobníku sériovej komunikácie. Ak mikropočítač prijal hodnotu cez sériovú komunikáciu, splní sa podmienka a nasleduje načítanie prijatej informácie do premennej *serialCode*, ktorá je dátového typu CHAR. Príkazom *Serial.read()* čítame informáciu zo zásobníka ako BYTE. Z toho dôvodu sme zvolili typ premennej CHAR. Rozhodli sme sa, že prijímané a odosielané hodnoty sériovou komunikáciou v rámci ovládania a nastavenia mikropočítača budú len celočíselné hodnoty. Preto sme vedeli, že na získanie hodnoty dátového typu INT (celé číslo) z typu CHAR dostaneme odčítaním znaku „0“. Týmto krokom a následným formálnym pretypovaním hodnoty na INT dostávame z BYTE-u celočíselnú hodnotu INT. Na základe zistenia hodnoty, ktorá bola načítaná zo sériovej komunikácie sme vedeli rozhodnúť, akú operáciu má mikropočítač vykonať. Vytvorili sme štyri podmienky, kde každá ma za úlohu iné správanie mikropočítača. Podmienky slúžia na potvrdenie pripojenia sériovej komunikácie, spustenie kontinuálneho merania, spustenie merania so zastavením pred voľným pádom nápravy a na zmenu doby merania. Prijatá informácia vždy spĺňa len jednu z podmienok.

```

//Nastavenie a spustanie merania seriovou komunikaciou
if (Serial.available()) {
  serialCode = Serial.read();
  serialCodeInt = (int)(serialCode-'0');
  if(serialCodeInt == 1) {
    Serial.print(1); //Odosielam cislo je pre potvrdenie prijatia prikazu
  }

  if(serialCodeInt == 2) { //Kontinualne meranie bez prerusenia
    ContinualMeasure();
  }

  if(serialCodeInt == 3){ //Meranie s ustalenim sa hodnot pred padom koleasa
    dobaMerania = dobaMerania + 100;
    StepMeasure();
  }

  if(zmenaDobyMerania == true){
    dobaMerania = serialCodeInt*100;
    zmenaDobyMerania = false;
    Serial.print(serialCodeInt); //Odosielam cislo je pre potvrdenie prijatia prikazu
  }

  if(serialCodeInt == 0){
    zmenaDobyMerania = true;
    Serial.print(0); //Odosielam cislo je pre potvrdenie prijatia prikazu
  }
}

```

Obr. 23 Sériová komunikácia na strane mikropočítača

Niektorým celočíselným hodnotám sme takto priradili ich funkciu. Tabuľka 12 obsahuje rozpis funkcií a ich priradenie celočíselným hodnotám.

Hodnota	Popis	Reakcia
0	Zmena doby merania, po príchode hodnoty 0 sa očakáva ešte príchod času merania	Odoslanie „0“
1	Začiatok sériovej komunikácie	Odoslanie „1“
2	Spustenie kontinuálneho merania	Spustenie merania
3	Spustenie merania s čakaním (STEP)	Spustenie merania
4 - 9	Prijatie doby merania (očakávané po „0“)	Odoslanie prijatej hodnoty

Tabuľka 12 Tabuľka hodnôt sériovej komunikácie

5.1.3 Proces merania

Proces merania tvoria dve skupiny operácií, ktoré sú navzájom prepojené a ich prepojenie je nevyhnutné.

Prvá skupina obsahuje prípravu merania vo forme nastavenia polohy geometrického modelu na najvyšší bod vačky. Táto operácia je vykonaná chodom krokového motora bez zaznamenávania údajov merania. Obsluhu prvej operácie má na starosti funkcia *preparePosition()*. Obsah tejto funkcie sme zobrazili na Obr. 24.

```
void preparePosition() {  
  
    if(digitalRead(sensor) == 0){ //Odchod z Home pozicie  
        while (digitalRead(sensor) == 0) { // otacam sa pomaly, kym neodidem zo snimaca  
            digitalWrite(stepPin, HIGH);  
            delayMicroseconds(motorSpeed);  
            digitalWrite(stepPin, LOW);  
            delayMicroseconds(motorSpeed);  
        }  
    }  
    while (digitalRead(sensor) == 1) { // otacam sa pomaly, kym nenarazim na snimac  
        digitalWrite(stepPin, HIGH);  
        delayMicroseconds(motorSpeed);  
        digitalWrite(stepPin, LOW);  
        delayMicroseconds(motorSpeed);  
    }  
    distance = 0;  
}
```

Obr. 24 Funkcia prípravy merania - preparePosition()

Funkcia sa skladá z dvoch cyklov WHILE. Prvý cyklus slúži na odchod z pozície „Home“. Pozíciu „Home“ sme definovali ako pozíciu, v ktorej sa model nachádza po zapnutí. Na vačku tlačí váha geometrického modelu a vo vypnutom stave systému sa vypne aj brzda motora, čo spôsobí otočenie vačky do polohy, kedy je koleso modelu najbližšie k hriadeľu. V tejto polohe sa však špička vačky nachádza v priestore pred laserovým snímačom polohy a snímač aktivuje vstup na mikropočítači. Preto tento stav evidujeme ako „domovská pozícia“ a prvý cyklus funkcie je spôsobom na opustenie domovskej pozície. Ak sa už vačka nenachádza v priestore pred snímačom polohy, tak program prejde do druhého cyklu WHILE. V tomto cykle sa otáča hriadeľom krokového motora, kým snímač polohy opäť neaktivuje vstup mikropočítača. Teraz ale vieme, že aktivovaný vstup je signálom dosiahnutia maximálnej výšky geometrického modelu a je pripravený na simuláciu zjazdu kolesa z obrubníka, čo v prípade modelu predstavuje pád na rovnú hranu vačky.

Kroky krokového motora generujeme jednoduchým zapínaním a vypínaním digitálneho výstupu. Medzi zapnutie a vypnutie výstupu sme vložili funkciu *delayMicroseconds()*, ktorej funkciou je vytvoriť čakanie v mikrosekundách dĺžky definovanej v zátvorke tejto funkcie.

Po dokončení prvej operácie nastavenia polohy vačky nasleduje druhá operácia. Úlohou druhej operácie je pustiť geometrický model do voľného pádu a súčasne merať zrýchlenia na neodpruženej a odpruženej hmote. Druhá operácia je rozdelená do dvoch funkcií merania. Dve funkcie merania sú si navzájom podobné. Jedna funkcia *ContinualMeasure()* predstavuje meranie zrýchlenia za kontinuálneho chodu vačky. Druhá funkcia *StepMeasure()* má za úlohu počkať niekoľko milisekúnd pred spustením geometrického modelu do voľného pádu no v tomto čase už prebieha meranie. Pre vzájomnú podobnosť funkcií sme sa rozhodli uviesť obrázok funkcie *StepMeasure()* Obr. 25, keďže vychádza z funkcie *ContinualMeasure()* s doplnením čakania na ustálenej hodnote zrýchlenia.

```
void StepMeasure(){
  preparePosition();
  //Čakanie na ustálenej hodnote
  i = 0;
  while(i < 200){ //Doba trvania Serial.print je cca 750us + 250us = 1ms
    measure();
    delayMicroseconds(250);
    i++;
  }
  i = 0;

  while (distance <= 100) { // nasledne sa otacam rychlo a umoznim kolesu volny pad
    digitalWrite(stepPin, HIGH);
    measure();
    digitalWrite(stepPin, LOW);
    measure();
    distance = distance + 1;
  }

  while ( i < dobaMerania) {
    measure();
    i++;
    delayMicroseconds(250);
  }
  i = 0;
}
```

Obr. 25 Funkcia merania zrýchlenia - StepMeasure()

Začiatok tejto funkcie obsahuje už spomínanú prípravu na meranie. Po príprave nasleduje čakanie na ustálenej hodnote zrýchlenia po dobu 200 ms. Cyklus čakania na ustálenej hodnote obsahuje už funkciu *measure()*, ktorá zabezpečuje meranie analógových hodnôt zrýchlenia.

Rovnako obsahuje aj čakanie 250 μs po každom meraní. Po skončení čakania na ustálenej hodnote prichádza cyklus, ktorý obsahuje súbežné meranie hodnôt zrýchlenia ako aj spustenie geometrického modelu do voľného pádu a to rýchlym otočením vačky z pozície najvyššieho bodu geometrického modelu do domovskej pozície. Rýchlym otočením vačky sme chceli dosiahnuť, aby vačka nebránila voľnému pádu modelu, čím by mohla ovplyvniť dáta merania. Po otočení vačky do domovskej pozície pokračuje už len meranie zrýchlenia a to po dobu merania, ktorá je preddefinovaná, ale používateľ má možnosť si dobu merania upraviť. V cykle merania sa nachádza volanie funkcie merania *measure()* a čakanie 250 μs . Kombinácia funkcie merania s čakaním vychádza už zo spomínaného problému, kedy sme meraním zistili, že načítanie analógovej hodnoty zo snímačov, prepočet tejto hodnoty a následné odoslanie oboch hodnôt cez sériovú komunikáciu trvá približne 750 μs . Rozhodli sme sa pre vzorkovanie 1ms. To znamená, že samotným odoslaním dát program stojí 75% času nami určenej vzorkovacej frekvencie, čiže zvyšných 25% času sme doplnili čakaním v mikrosekundách. Funkcia merania zrýchlenia *measure()* sa nachádza v oboch funkciách *StepMeasure()* ako aj *ContinualMeasure()*. Táto funkcia priamo obsahuje čítanie hodnôt zrýchlenia zo snímačov zrýchlenia analógovými vstupmi mikropočítača. Funkciu *measure()* sme zobrazili na Obr. 26.

```
void measure(){
    z1_adc_value = analogRead(Z1_axis);
    z2_adc_value = analogRead(Z2_axis);

    //Acceleration in G
    z1_g_value = -((float)z1_adc_value - 328)/67*9.81;
    z2_g_value = -((float)z2_adc_value - 326)/67*9.81;
    Serial.println(String( String(z1_g_value) + " " + String(z2_g_value)));
}
```

Obr. 26 Funkcia merania hodnôt zrýchlenia - *measure()*

Prvé riadky funkcie obsahujú samotné čítanie hodnôt z analógových vstupov A0 a A1 mikropočítača. Analógové hodnoty však nepredstavujú žiadnu fyzikálnu veličinu a preto sme museli tieto hodnoty prepočítať na hodnoty zrýchlenia s prihliadnutím na gravitačné zrýchlenie Zeme, ktoré na tieto snímače pôsobí aj, keď sú v pokoji. Pri uložení osi snímača kolmo na Zem sme na vstupe mikropočítača čítali hodnotu 395. Hodnota 395 pre nás predstavovala hodnotu gravitačného zrýchlenia 9,81 m/s^2 . Snímač sme následne uložili do polohy, kedy bola os Y vo vodorovnej polohe, aby sme dosiahli hodnotu gravitačného zrýchlenia voči tejto osi 0 m/s^2 . Za týchto podmienok sme na vstupe mikropočítača čítali hodnotu 328. To znamená, že hodnota 1g

je rozdielom hodnôt $395 - 328 = 67$. Aby sme dostali hodnotu zrýchlenia vytvorili sme rovnicu prepočtu hodnôt každého snímača na hodnotu zrýchlenia s uvažovaním gravitačného zrýchlenia.

$$z_{1_g} = \frac{z_{1_adc} - 328}{67} 9,81 \quad (3)$$

$$z_{2_g} = \frac{z_{2_adc} - 326}{67} 9,81 \quad (4)$$

Rovnica (3) znázorňuje prepočet hodnôt pre prvý snímač a rovnica (4) pre druhý snímač. Hodnoty zrýchlenia snímačov sa mierne líšili, preto sme museli vytvoriť rovnicu pre každý snímač zvlášť. Po prepočte sú hodnoty zrýchlení odosielané sériovou komunikáciou do osobného počítača na ďalšie spracovanie. Na odoslanie sme využili príkaz *Serial.println()*, a v ňom konverziu hodnôt typu FLOAT na STRING. Hodnoty odosielame ako reťazec v jednom riadku oddelené medzerou.

5.2 Program grafického používateľského rozhrania (Matlab)

Na vytvorenie programu GUI na ovládanie geometrického modelu nápravy sme sa rozhodli využiť multiplatformový matematicky zameraný nástroj Matlab. Jeho najsilnejšou výhodou je práve možnosť spúšťať aplikácie Matlabu na akomkoľvek operačnom systéme. Podmienkou je buď nainštalovaný Matlab alebo vytvorenie samostatnej aplikácie pomocou Matlab prekladača (Compiler). Compiler aplikáciu skompiluje a aplikácia je ďalej šíriteľná a inštaluje sa spolu s Runtime-om Matlabu. Runtime je možné stiahnuť z internetu pri inštalácii aplikácie alebo je možné ho vložiť do balíka inštalčných súborov aplikácie. Poslednou možnosťou ako poskytnúť aplikáciu používateľom je zabaliť ju ako aplikáciu v Matlabe. Používateľ má potom možnosť si aplikáciu nainštalovať ako súčasť jeho Matlabu a zobrazí sa mu v tabulátore „Apps“ medzi ostatnými aplikáciami.

My sme zvolili práve možnosť aplikácie vytvorenej ako Toolbox Matlabu. Podmienkou pre jej chod je už nainštalovaný Matlab v počítači spolu s tzv. Curve Fitting Toolbox-om. Našej aplikácii sme definovali niekoľko úloh:

- vytvorenie sériového spojenia zvoleným sériovým portom,
- prijatie nameraných hodnôt cez sériový komunikačný port,
- zobrazenie nameraných hodnôt prostredníctvom grafu,
- možnosť zmeniť dobu merania,
- export dát do súboru.

Aplikáciu sme programovali cez dizajnér aplikácií (App Designer). Dizajnér je vytvorený tak, aby programátor vkladal na okno aplikácie tlačidlá, textové polia a iné. Celý zdrojový kód aplikácie nie je programátorovi prístupný. Prístupné sú len volania vytvorené k tlačidlám alebo iným prvkom ovládania prostredia. Programátor má možnosť vytvoriť si vlastné globálne premenné, funkcie a *startup()* funkciu spúšťanú pri spustení aplikácie.

5.2.1 Premenné programu GUI

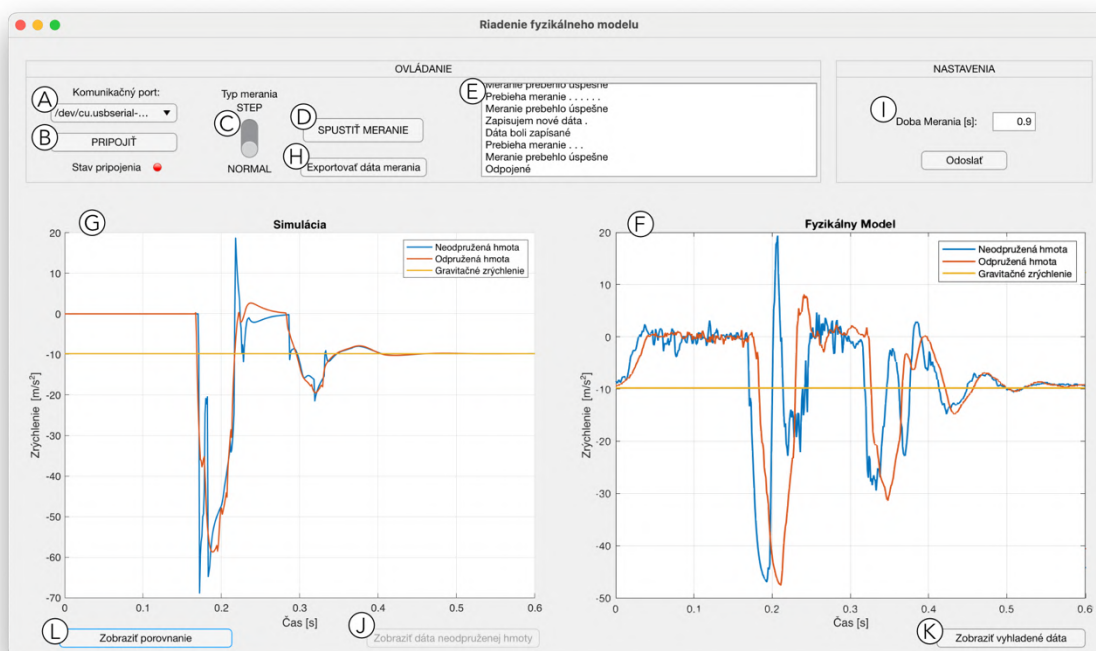
V nasledujúcej tabuľke (Tabuľka 13) sú uvedené premenné, ktoré sme v aplikácii riadenia fyzikálneho modelu použili.

Názov	Dátový typ	Popis	Začiatková hodnota
pickedSerialPort	STRING	Zvolený sériový port počítača	„“
status	STRING	Aktuálny stav systému	„“
message	STRING	Dáta prijaté cez sériový port	„“
timeout	INT	Doba čakania sériového spojenia	10
dobaMerania	INT	Čas merania fyzikálneho modelu	5
measureType	CHAR	Typ merania (STEP, NORMAL)	,2‘
device	OBJECT	Zariadenie sériového portu	null
buttonWasPressed	BOOL	Prístupnosť tlačidla pripojenia	false
connected	BOOL	Stav zariadenia sériového portu	false
values	[DOUBLE]	Usporiadané dáta merania	zeros(100,2)
i	INT	Iteračná premenná	0
simulationData_Car	[DOUBLE]	Dáta merania simulácie	null
simulationData_Wheel	[DOUBLE]	Dáta meranie simulácie	null
simDataVisibility	BOOL	Zobrazenie dát simulácie	true
smValues_tlmene	[DOUBLE]	Vyhladená krivka merania	null
smValues_netlmene	[DOUBLE]	Vyhladená krivka merania	null
smoothDataVisible	BOOL	Zobrazenie vyhladenej krivky	false
AccWasMeasured	BOOL	Stav uskutočnenia merania	false

Tabuľka 13 Premenné aplikácie riadenia fyzikálneho modelu

5.2.2 Popis aplikácie

Na uvedenom Obr. 27 sa nachádza okno grafického používateľského rozhrania aplikácie s abecedným označením prvkov.



Obr. 27 Okno aplikácie riadenia fyzikálneho modelu

Okno aplikácie má rozmer 1280x720 px. Snažili sme sa zvoliť najväčší možný rozmer s prihliadnutím na aktuálnu priemernú veľkosť rozlíšenia monitorov. Veľkosť okna aplikácie je možné prispôbovať. Ovládacie a zobrazovacie prvky aplikácie sme v okne rozložili do troch častí – ovládanie aplikácie, nastavenie a zobrazenie výsledkov. Pri usporiadaní častí sme sa snažili ich uložiť na základe pravidla čítania a písania textu, čiže zľava doprava, zhora nadol. Používateľ postupuje v aplikácii prirodzeným spôsobom až k výsledkom meraní v podobe grafov.

Vysvetlenie prvkov ovládania:

A. Rozbaľovacie menu komunikačného portu

Prvým ovládacím prvkom zľava je voľba komunikačného sériového portu. Používateľ má po rozbalení menu možnosť si vybrať len z dostupných komunikačných portov jeho zariadenia. Týmto spôsobom eliminujeme chyby, ktoré môžu vzniknúť pri manuálnom písaní komunikačných portov používateľmi. Z hľadiska kódu predstavuje výber dostupných komunikačných portov len jeden riadok kódu. Použili sme funkciu `serialportlist(„available“)`.

Príkaz zistí aktuálne dostupné komunikačné porty počítača. Dostupné porty potom vypíšeme ako položky rozbaľovacieho menu. Po kliknutí používateľa na konkrétny komunikačný port sa názov portu uloží do premennej *pickedSerialPort* ako reťazec.

B. Tlačidlo vytvorenia pripojenia

Akonáhle si používateľ zvolil komunikačný port, stlačí tlačidlo „PRIPOJIŤ“. Funkcia obsluhujúca toto tlačidlo má niekoľko dôležitých úloh. Po zatlačení dôjde ku kontrole voľby sériového portu. Ak používateľ nezvolil sériový port proces pripájania sa neuskutoční. Druhou podmienkou je, aby bola hodnota premennej *buttonWasPressed = false*. To znamená, že nie je pripojené žiadne zariadenie a na tlačidle sa zobrazuje nápis „PRIPOJIŤ“. Ak sú obe predošlé podmienky splnené, tak program najskôr vyčistí premennú *device* a následne v bloku *try – catch* vytvára sériové spojenie a na tlačidle sa zobrazí nápis „ODPOJIŤ“. Blok sme využili kvôli možným zlyháním operácie pripájania sa. Ak sa zariadenie úspešne pripojí, tak vykonávame kontrolu odoslaním znaku ‚1‘ sériovou komunikáciou do mikropočítača a očakávame návrat rovnakého znaku. Používateľovi sa počas tohto procesu zobrazuje informácia „Kontrola pripojenia“. Ak sa odoslaný znak vráti naspäť, znamená to pre nás potvrdenie pripojenia a zobrazí sa informácia „Spojenie aktívne“.

Tlačidlo vytvorenia spojenia sa používa aj pri uzatváraní spojenia. Používateľ tlačidlo zatlačí a vyvolá tak zmazanie obsahu objektu *device*. Na tlačidle sa opäť objaví nápis „PRIPOJIŤ“ a systém je pripravený na opätovné pripojenie hardvéru cez sériový port.

C. Prepínač typu merania

Na základe dohody sme vytvorili dva typy merania, ako už bolo spomínané pri opise programu bežiacom v mikropočítači. Typ merania si používateľ volí prepínačom. Typ merania „NORMAL“ je kontinuálnym meraním bez zastavenia chodu vačky. Typ merania STEP vytvorí čakanie dĺžky 200 ms v najvyššej polohe fyzikálneho modelu pred spustením do voľného pádu. Prepínač v programe zmení znak, ktorý sa odosiela do mikropočítača. Znak ‚2‘ znamená kontinuálne meranie a znak ‚3‘ pre mikropočítač zas znamená, že pôjde o meranie s čakaním.

D. Tlačidlo „Spustiť meranie“

Tlačidlom sa spúšťa proces merania na základe prepínačom zvoleného typu merania. Vo funkcií volania obsluhy tlačidla sa nachádza aj spracovanie dát z merania. Vyhodnotenie stavu prepínača prebieha príkazom *switch*. Pred odoslaním príkazu merania do mikropočítača sa kontroluje, či je zariadenie pripojené. V prípade, ak je zariadenie pripojené, tak príkazom *flush(device)* vyčistíme zásobník sériovej linky a príkazom *wirte(device, measureType, ‚char‘)*

odošleme príkaz na spustenie merania, kde *device* je objekt nášho mikropočítača, *measureType* je prepínačom zvolený typ merania a *,char‘* je dátový typ odosielanej informácie. Po odoslaní príkazu sa používateľovi zobrazí informácia „Prebieha meranie“, kde sa každú sekundu zobrazí o jednu bodku viac za informáciou až po dobu merania. Dáta z merania sa zapisujú do premennej *message*. Do premennej sa zapisujú ako návratová hodnota funkcie *waitForResponse()*. Funkcia má za úlohu sledovať dostupnosť dát v zásobníku sériovej komunikácie. Funkcia má stanovený čas čakania premennou *timeout*. V prípade, že sú dáta dostupné pred uplynutím časovej doby, tak sa hodnoty zapisujú do premennej *message*. Dáta v premennej *message* sa potom rozdelia príkazmi *splitlines* a *split* a následne sa konvertujú do typu DOUBLE. Po konverzii sa zobrazia na grafe.

E. Zobrazenie aktuálneho stavu

Používateľ má po celú dobu používania možnosť vidieť aktuálny aj historický vývoj jeho operácií v textovej oblasti. Textová oblasť zobrazuje používateľovi všetky informácie od nadviazania spojenia až po ukončenie aplikácie. Počas programovania sme vytvorili funkciu, ktorá má za úlohu odosielať aktuálny stav po interakcii s ovládacími prvkami aplikácie.

F. Graf meraní geometrického modelu

Na grafe s názvom „Fyzikálny model“ sa zobrazujú výsledky merania modelu po každom meraní. Graf sa po každom meraní prepíše novými hodnotami a zobrazia sa nové krivky merania. Na grafe sa zobrazujú tri krivky: krivka zrýchlenia neodpruženej hmoty, krivka zrýchlenia odpruženej hmoty a konštantné gravitačné zrýchlenie.

G. Graf výsledkov simulácie matematického modelu

Na druhom grafe s názvom „Simulácia“ sú zobrazené výsledky simulácie. Graf výsledkov simulácie matematického modelu sme vložili do aplikácie, aby bolo možné reálny a matematický model navzájom porovnať.

H. Export nameraných hodnôt

Pod tlačidlo spustenia merania sme vložili tlačidlo „Export dát merania“. Po kliknutí na toto tlačidlo sa používateľovi zobrazí okno znázornené na Obr. 28. V okne sa nachádza vysvetlivka usporiadania dát merania do stĺpcov, tlačidlo na odoslanie údajov do „Workspace“ a tlačidlo na exportovanie dát do súboru. Tlačidlom „Odoslať do Workspace“ sa údaje pošlú do pracovného priestoru Matlabu, kde je možné s nimi ďalej pracovať. Kliknutím na tlačidlo „Exportovať do súboru“ sa otvorí okno, v ktorom si používateľ zvolí miesto uloženia a názov súboru. Dáta sa exportujú ako CSV súbor. V prípade, že používateľ nechce exportovať dáta, okno exportu dát zatvorí tlačidlom „Zatvor“. Okno exportu sa otvára zmenou jeho parametra „Visibility“. Na

odoslanie dát do pracovného priestoru sme využili funkciu *assignin()* a na export dát do súboru sme použili funkciu *uiputfile()*, ktorej výhodou je, že vyvoláva aj otvorenie okna pre voľbu miesta uloženia súboru a názvu súboru.



Obr. 28 Okno exportu dát do súboru alebo pracovného prostredia

I. Zmena doby merania

Vytvorili sme možnosť zmeny dĺžky merania v časovom intervale 0,4 – 0,9 s. Používateľ zmení dobu merania vpísaním požadovanej dĺžky merania do textového poľa. Textové pole sme upravili tak, aby doň bolo možné písať len numerické hodnoty v spomínanom intervale. Preto sa nemôže stať, že by používateľ napísal a odoslal nesprávnu hodnotu. Po vpísaní požadovanej hodnoty používateľ túto hodnotu zapíše do mikropočítača stlačením tlačidla „Odoslať“. Operácia zapísania nového času merania do mikropočítača je v kóde rozdelená na dve časti. Okamžite po stlačení tlačidla „Odoslať“ sa najprv posieľa do mikropočítača znak ,0‘. Mikropočítač znak rozozná a ako nasledujúcu informáciu očakáva dobu merania. Svoju pripravenosť na načítanie doby merania mikropočítač potvrdí tak, že do aplikácie riadenia fyzikálneho modelu odošle rovnako znak ,0‘. Ak v aplikácii prijme rovnaký znak ako sme odoslali, tak sa odošle hodnota doby merania. Túto hodnotu opäť mikropočítač potvrdzuje spätným odoslaním rovnakej hodnoty. Ak sa odoslaná a prijatá hodnota zhodujú, znamená to, že zmena dĺžky merania je zapísaná. Týmto spôsobom odoslania doby merania sme sa snažili zabrániť chybám v komunikácii.

J. Prepínač zobrazenia dát simulácie

Graf simulácie obsahuje prepínač, ktorý slúži na prepínanie zobrazených kriviek medzi krivkou neodpruženej hmoty a krivkou odpruženej hmoty.

K. Prepínač vyhladenia kriviek

Graf zobrazenia meraných dát na fyzikálnom modeli zas obsahuje prepínač, ktorým si používateľ zobrazí vyhladenejšie krivky merania. Prepínač sme k tomuto grafu pridali z dôvodu, že snímače zrýchlenia sú veľmi citlivé a krivka meraní môže byť ťažšie čitateľná pri výkyvoch hodnôt zrýchlenia. Tlačidlom sa spustí funkcia *smooth()* nachádzajúca sa v *Curve Fitting Toolbox*. Funkcia spôsobí vyhladenie krivky a zmenšenie rozptylu dát.

L. Prepínač zobrazenia porovnania a simulácie

Tlačidlo „Zobraziť porovnanie“ slúži na prepínanie obsahu grafu simulácie. Primárne sú na grafe simulácie zobrazené krivky z údajov simulácie. Prepnutím tlačidla je však možné zobraziť krivky buď odpruženej alebo neodpruženej hmoty simulácie a meraní súčasne. Toto zobrazenie slúži pre jednoznačnejšiu viditeľnosť reálnych a simulačných kriviek a ich vzájomné porovnanie.

Kompletný program aplikácie používateľského prostredia a programu mikropočítača sa nachádza v prílohe (Príloha B:).

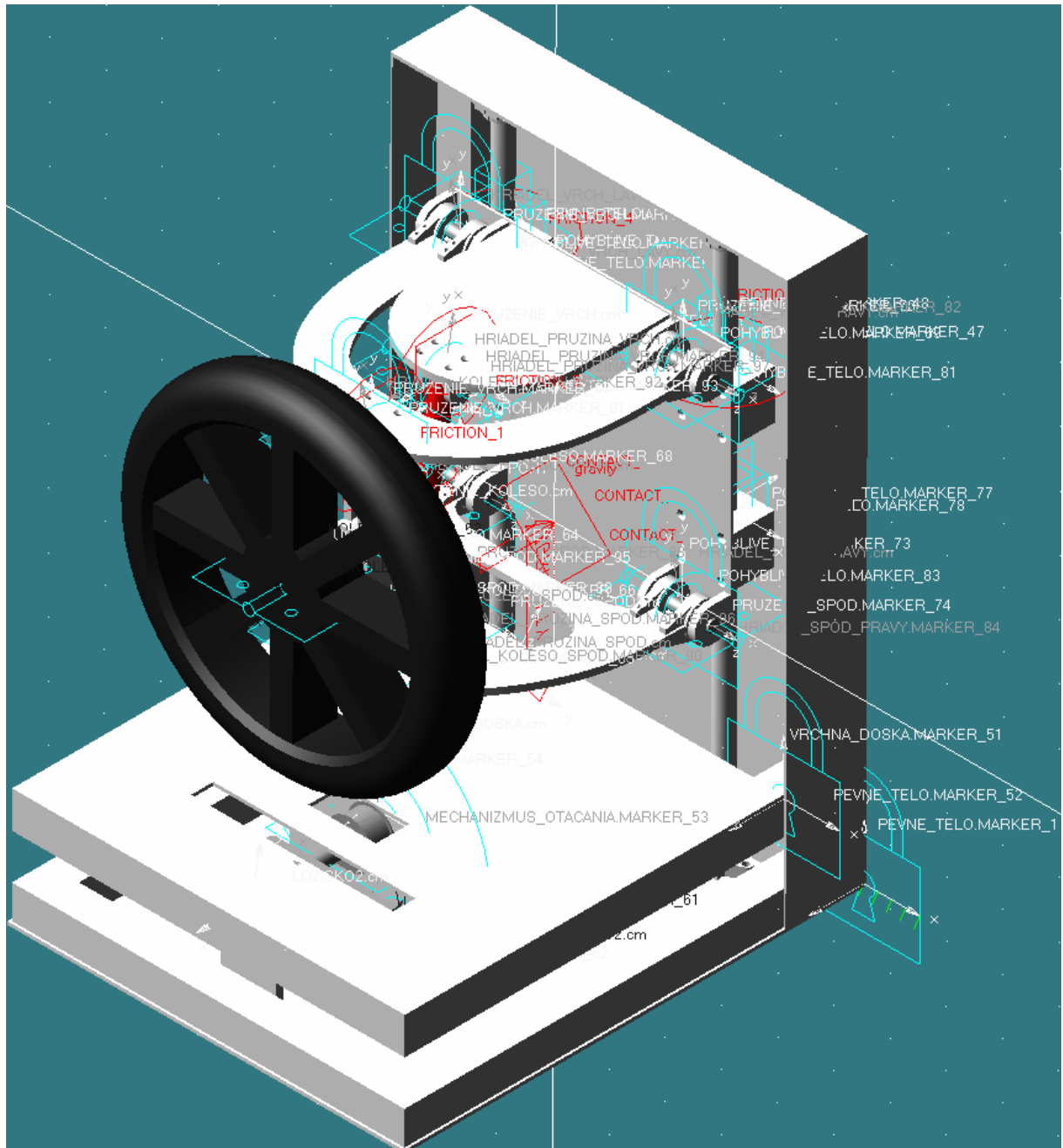
6 Porovnanie výsledkov simulácie a reálneho modelu

6.1 Simulácia

Simuláciu a výpočet zrýchlenia matematického modelu sme vykonali pomocou programu MSC Adams. Do programu sme importovali model vytvorený nástrojom Fusion 360. Pre presnejšie výsledky simulácie sme hmotnosti jednotlivých dielov lichobežníkovej nápravy vložili ako vstup používateľa na základe spomínanej tabuľky hmotností (Tabuľka 4). V programe MSC Adams sme vytvorili spojenia medzi komponentami. Využili sme pri tom posuvné a otočné spojenia. Posuvné spojenie sme aplikovali medzi odpruženú hmotu a nosnú konštrukciu modelu vo vertikálnom smere, čím sme simulovali kĺzavý pohyb kĺzných vozíkov na lineárnom vedení. Ako pri prepojení odpruženej hmoty s nosnou konštrukciou, tak aj v prípade ďalších spojov sme tieto spoje realizovali na základe stupňov voľnosti fyzikálneho, reálne zostrojeného modelu. Výsledný simulačný model, ktorý sme vytvorili, je zobrazený na Obr. 29. Nosnú konštrukciu sme museli ukotviť k zemi pevným uzamknutím. Pevné spojenie sme vytvorili aj medzi perami ktoré sú uložené v dieloch vykonávajúcich otáčavé pohyby ako napríklad ramená lichobežníkovej nápravy. Perá pevne spojené s odpruženou hmotou sme potom prepojili s ramenami nápravy otáčavým spojom tak, že os otáčania prechádza rovnobežne s horizontálnymi osami pier. Na druhej strane sú ramená navzájom prepojené uložením kolesa. Uloženie kolesa je s ramenami prepojené rovnakým spôsobom ako ramená s odpruženou hmotou a to pomocou pier zabezpečujúcich otáčavý pohyb. K uloženiu kolesa je potom pripevnené koleso prostredníctvom hriadeľa, ktorým koleso disponuje od výroby. Tento hriadeľ sme k uloženiu pripevnili otáčavým spojením, čím sme kolesu umožnili vykonávať rotačný pohyb okolo svojej osi. Spodné rameno nápravy je tlmičom spojené s odpruženou hmotou. Tlmič s pružinou sme nevkladali do programu MSC Adams ako model, ale sme ich vytvorili už ako objekt programu MSC Adams. Spravili sme tak z dôvodu lepšieho nastavenia parametrov tlmiča a pružiny. Posledným potrebným spojením bol kontakt medzi kolesom a podložkou resp., nosnou konštrukciou fyzikálneho modelu, na ktorú koleso dopadá. Tento spôsob prepojenia sme vytvorili spojením typu „*Contact*“. Ide o spojenie objektov kontaktnými plochami. To znamená, že koleso sa cez plochu nosnej konštrukcie neprepadne a nie je potrebné žiadne pevné prepojenie plochy kolesa s plochou nosnej konštrukcie.

Aby sme sa výsledkami simulácie viac priblížili reálnym výsledkom fyzikálneho modelu, vložili sme ku kĺznym vozíkom lineárneho vedenia konštanty odporu proti pohybu po

lineárnom vedení. Výrobcom udávaný odpor je menší ako 0,05. Rozhodli sme sa preto pre statický odpor (pri začiatku pohybu) o hodnote 0,03 a dynamický odpor 0,024. Ako je vidieť na Obr. 29, podobný odpor (v programe označený ako „Friction“) sme vložili aj pre otáčavé časti uloženia kolesa. Tu sme si zvolili ľubovoľné hodnoty a to 0,028 ako statický odpor a 0,024 ako dynamický odpor.



Obr. 29 Simulačný model v programe MSC Adams

6.2 Porovnanie výsledkov

Po dokončení fyzikálneho modelu a simulačného modelu je možné porovnať výsledky simulácie s reálnym modelom. Spustenie reálneho modelu pruženia a následné zobrazenie údajov merania na grafe prinieslo zistenie, že bol evidentný rozdiel medzi výsledkami simulácie a reálneho modelu. Pozorovaním fyzikálneho modelu sme zistili, že celý model po dopade na nosnú konštrukciu vyskočí naspäť nad úroveň konštrukcie o 1 cm. Vyskočenie modelu je vyvolané pružnosťou pneumatiky, ktorá sa pri dopade zachová ako pružina. Schopnosť odrazivosti pneumatiky vieme vyjadriť koeficientom reštitúcie. Koeficient reštitúcie reálneho kolesa sme určili experimentálne tak, že sme merali výšku odrazenia sa kolesa od pevného povrchu a hodnotu výšky výskoku a výšky spustenia sme vložili do vzorca. Výpočet koeficientu reštitúcie sme uviedli v rovnici (5).

$$k_r = \sqrt{\frac{h}{H}} = \sqrt{\frac{0,14}{0,23}} = 0,78 \quad (5)$$

Rovnica obsahuje:

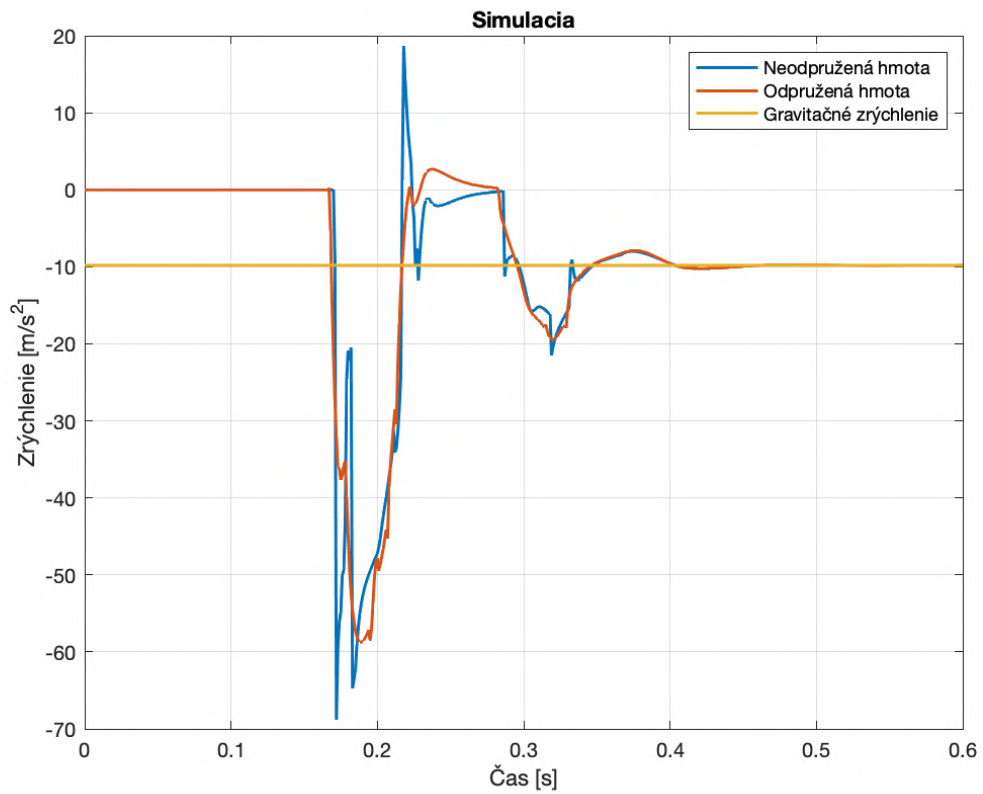
- h – výška odrazenie sa kolesa,
- H – počiatočná výška kolesa,
- k_r – koeficient reštitúcie

Výsledok pokusu sme zapísali v programe MSC Adams do kontaktu „Contact“, v ktorom sa nachádza pole pre zápis koeficientu reštitúcie. Po spustení simulácie sa však sústava pruženia pri prvom dopade na nosnú konštrukciu správala ako pevne spojený kus a pruženie sa prejavilo až pri páde nápravy z odrazenej výšky. Z toho dôvodu sme parametre tlmenia pneumatiky a vniknutia do konštrukcie nastavili opäť experimentálnym spôsobom. Hodnoty sme uviedli v tabuľke (Tabuľka 14).

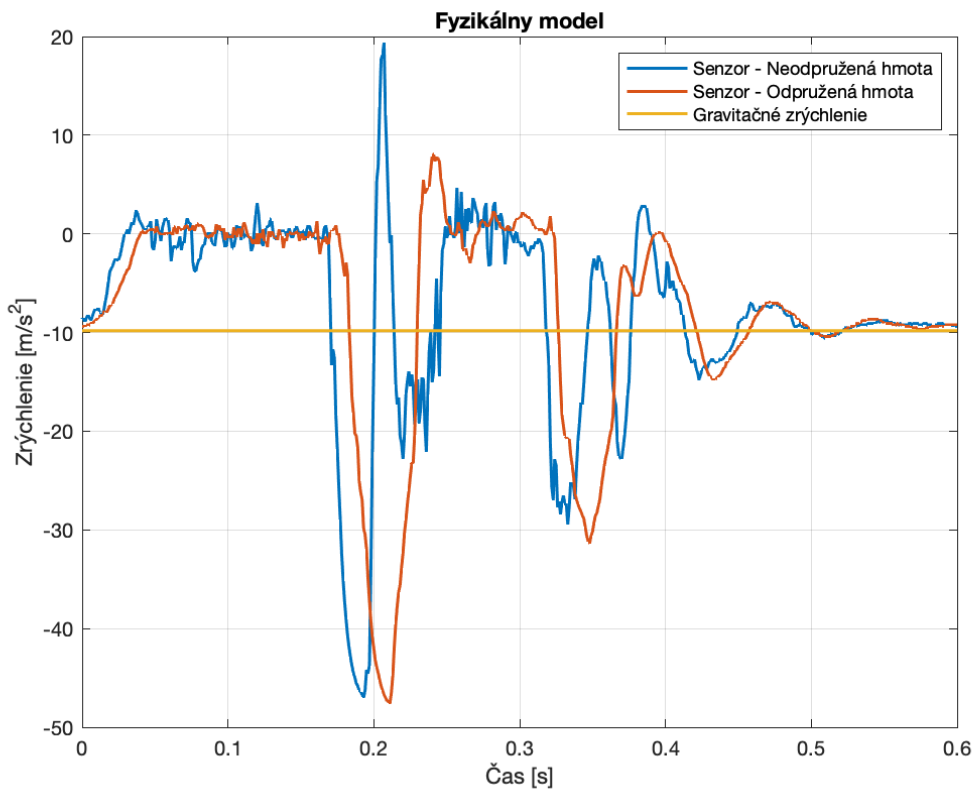
Tuhosť	Exponent sily	Tlmenie	Hĺbka prieniku
1,15E+07	2,2	80,0	1,0E-04

Tabuľka 14 Hodnoty prepojenia kolesa a konštrukcie "Contact"

Aby sme sa priblížili výsledkom reálneho modelu, museli sme upraviť aj tuhosť pružiny. Aj napriek tuhosti udanej výrobcom 2210 N/m, sme stanovili výslednú tuhosť pružiny až na 15000 N/m. Koeficient tlmenia tmiča sme určili 300 Ns/m. Pri týchto nastaveniach sa graf krivky zrýchlenia odpruženej a neodpruženej hmoty podobal krivkám vytvoreným na základe merania fyzikálneho modelu. Grafy sú znázornené na Obr. 30 a Obr. 31.



Obr. 30 Graf výsledkov simulácie



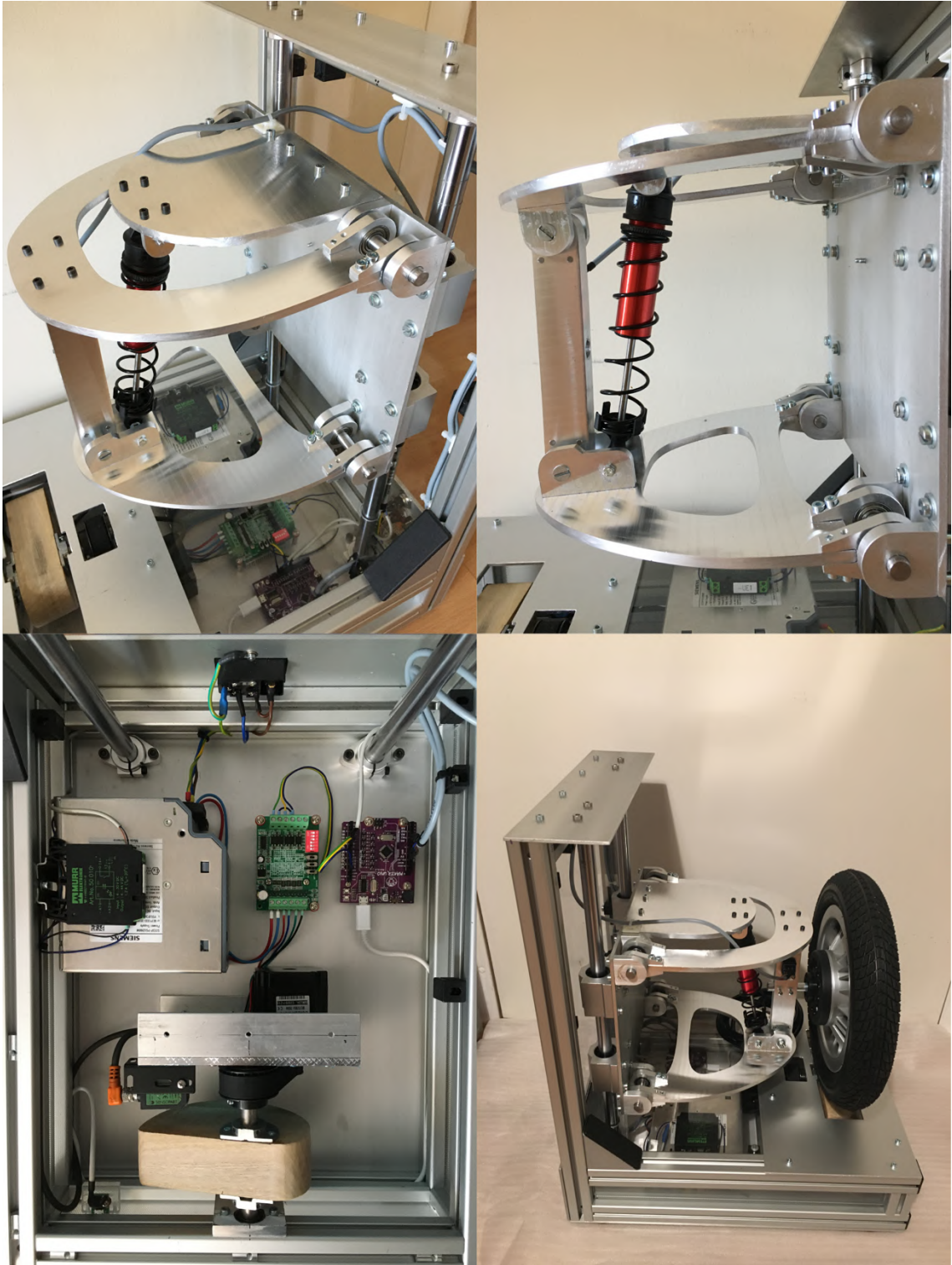
Obr. 31 Graf výsledkov fyzikálneho modelu

Na grafoch môžeme vidieť správanie sa modelov. Prvých približne 160 ms je merané zrýchlenie 0 m/s^2 . To je spôsobené voľným pádom sústavy, keďže počas voľného pádu je zrýchlenie padajúcich telies rovné gravitačnému zrýchleniu. Následne dochádza k dosiahnutiu nosnej konštrukcie systému, čo je na grafoch znázornené špičkovým zrýchlením. V prípade simulácie je špička zrýchlenia neodpruženej hmoty približne 68 m/s^2 , zatiaľ čo fyzikálny model dosiahne maximálne zrýchlenie približne 46 m/s^2 . Vplyvom pruženia pneumatiky celá sústava vyskočí nad povrch nosnej konštrukcie a preto je možné na grafoch pozorovať opäť voľný pád po ktorom dochádza opäť k druhej zápornej špičke zrýchlenia. Rozdiel medzi odpruženou a neodpruženou hmotou je viditeľný po dosiahnutí prvej zápornej špičky zrýchlenia. Odpružená hmota dosiahne dvakrát menšie zrýchlenie ako neodpružená hmota, čo je následkom pruženia a tlmenia.

Záver

Úlohou diplomovej práce bolo zostrojiť funkčný štvrtinový model pruženia automobilu a následne na ňom vykonať analýzu kmitania. Vytvoreniu fyzikálneho modelu predchádzalo vytvorenie počítačového CAD modelu programom Fusion 360. Tento počítačový 3D model sme vytvorili na základe podobnosti s reálnym modelom pruženia. Na základe dohody s vedúcim práce sme si zvolili lichobežníkovú nápravu pre jej stabilnú konštrukciu. Vytvorený 3D počítačový model sme vytvárali na základe vyhľadanej dostupnej pružiny s tlmičom. Po otestovaní funkčnosti v programe MSC Adams sme vytvorili reálny model lichobežníkovej nápravy automobilu. Ako materiál na výrobu sme si zvolili zliatinu hliníka ALU 7075, ktorá má dobré vlastnosti z hľadiska pevnosti a je dobre obrábatel'ná. Nosnú konštrukciu sme vyrobili z hliníkových profilov. Odpruženú hmotu sme s nosnou konštrukciou spojili pomocou lineárneho vedenia a klznými vozíkmi. Týmto spojením sme umožnili odpruženej hmote vertikálny pohyb, ktorý simuluje pohyb automobilu. K odpruženej hmote sú pripevnené ramená lichobežníkovej nápravy a uloženie tlmiča. K ramenám je pripevnené uloženie kolesa. Pohyb kolesa je obmedzený len na otáčanie okolo vlastnej osi a na posuvný pohyb vo vertikálnom smere. Pohyb sme obmedzili z dôvodu, že pohyb zatáčania kolesa nie je pre hľadanie fyzikálnych vlastností modelu potrebný.

Úlohou práce bolo k fyzikálnemu modelu zostrojiť aj budenie od vozovky a meranie zrýchlenia odpruženej a neodpruženej hmoty. Budenie od vozovky sme vyriešili pomocou krokového motora, ktorý sme prepojili s otočným hriadeľom remeňovým prevodom s pomerom 2:1. Na hriadeľi sa nachádza vačka, ktorou dvíhame nápravu do výšky 70 mm a následne je z tejto výšky spustená, čím simulujeme zjazd automobilu z obrubníka. Meranie prebiehalo pomocou snímačov zrýchlenia ADXL335, ktoré sme pripojili k mikropočítaču Maker UNO. Analógovými vstupmi mikropočítača sme zbierali hodnoty zrýchlení zo snímačov, konvertovali ich a posielali ďalej sériovou komunikáciou do počítača s grafickým používateľským rozhraním. GUI sme vytvorili programom Matlab ako samostatnú aplikáciu, ktorá však pre svoj chod potrebuje program Matlab a knižnicu „*Curve Fitting Toolbox*“. GUI umožňuje používateľom pohodlné ovládanie fyzikálneho modelu s okamžitým zobrazením nameraných hodnôt prostredníctvom grafov. Vlastnosti fyzikálneho modelu sme porovnali s výsledkami simulácie vytvorenej programom MSC Adams. Výsledné krivky simulácie a merania sa takmer zhodujú aj z dôvodu upravenia parametrov matematického modelu.



Obr. 32 Koláž fyzikálneho modelu

7 Bibliografia

- [1] **Baroš, Eduard.** *Srovnání semi-aktivních a konvenčních systému odpružení vozidel z pohledu jízdního komfortu.* Brno : Vysoké učení technické v Brně, 2015.
- [2] **Janco, Marcel.** autorubrik. [Online] 30. August 2015. <http://www.autorubik.sk/clanky/naprawy-osobnych-vozidiel/>.
- [3] **Products for Cars.** *Driven Solid Beam Axle.* [Online] 2021. https://www.zf.com/products/en/cars/products_34560.html.
- [4] **Suspension.com.** Suspension. [Online] 2021. <https://www.suspension.com/w83100>.
- [5] **Rabatin, Ondrej.** Webnoviny. [Online] 15. Január 2010. <https://www.webnoviny.sk/autolike/prva-jazda-opel-astra/>.
- [6] **Sajdl, Jan.** autolexicon. [Online] 2020. <https://www.autolexicon.net/cs/articles/naprava-de-dion/>.
- [7] **Audi.** Audi Technology Portal. *Double-Wishbone suspension.* [Online] Január 2007. <https://www.audi-technology-portal.de/en/chassis/wheel-suspension-steering/front-suspension>.
- [8] **Audi AG.** Audi MediaCenter. [Online] 16. November 2020. <https://www.audi-mediacyenter.com/en/photos/detail/audi-sq2-97138>.
- [9] —. Audi MediaCenter. [Online] 11. Marec 2020. <https://www.audi-mediacyenter.com/en/photos/detail/audi-a3-sportback-90663>.
- [10] **Moog.** Moog. *Is double wishbone suspension best for your car?* [Online] 2021.
- [11] **CNC príslušenstvo.** CNC príslušenstvo. [Online] 2020. <https://www.cncprislusenstvo.sk/ako-na-stavbu-cnc-stroja/linearne-vedenia/>.
- [12] **Techfun.** Techfun.sk. [Online] 2020. https://techfun.sk/produkt/motor-driver-tb6560-3a-pre-krokovce-motorceky/?gclid=Cj0KCQjwppSEBhCGARIsANIs4p7i7IZB5Tf4FSCEihgNcS9N9rct-XLv-kJpEzkUb9D6FnRjX-TGrp4aAlWfEALw_wcB.
- [13] **Industrial Electronics, Inc.** ieknox. [Online] 2020. http://www.ieknox.com/catalog/index.php?main_page=product_info&cPath=4121_4126&products_id=55673.

[14] **Madhu Kodati, S. Bandyopadhyay.** Semantic Scholar. [Online] 2013. <https://www.semanticscholar.org/paper/Kinematic-Analysis-of-the-Double-Wishbone-System-Kodati-Bandyopadhyay/95a62bb07dd5767796371be9ae68205208ed38c5>.

[15] **Bhd, Cytron Technologies Sdn.** *Maker-UNO User's Manual*. Penang, Malaysia : Lorong Industri Impian, 2017.

Prílohy

- Príloha A: Používateľský manuál
- Príloha B: Program aplikácie riadenia a mikropočítača
- Príloha C: Technická dokumentácia
- Príloha D: Fotodokumentácia
- Príloha E: Štruktúra repozitára

Príloha A: Používateľský manuál

Obsluha zariadenia

UPOZORNENIE!

Zariadenie obsluhujte opatrne v rámci zásad bezpečnosti pri práci. Je zakázané odnímať kryty na zariadení a dotýkať sa elektronických častí zariadenia. Za behu merania dodržujte bezpečnú vzdialenosť od zariadenia aspoň 50 cm. Počas behu merania je zakázané so zariadením manuálne manipulovať.

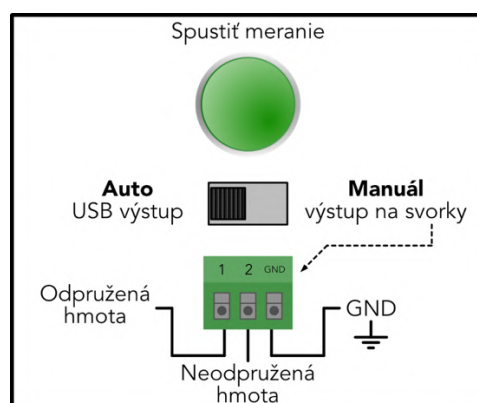
Nastavenie a pripojenie zariadenia

Pre vykonanie presného merania je potrebné, aby bolo zariadenie umiestnené na svojom podstavci vo vodorovnej polohe. Zariadenie by sa nemalo nakláňať a malo by byť položené na stabilnej podložke (prípadne na podlahe).

Zariadenie sa pripája k elektrickej sieti (230 V AC, 50 Hz) počítačovým káblom, ktorý musí obsahovať uzemňovací kolík. Sieťový kábel sa pripája do konektora na zadnej stene zariadenia. Po pripojení kábla je možné zariadenia zapnúť červeným prepínačom. Medzi prepínačom a zásuvkou sieťového kábla sa nachádza 6 A sklená poistka.

Manuálne spustenie merania

Zariadenie disponuje možnosťou manuálneho spustenia behu programu merania. Ovládací panel manuálneho behu programu sa nachádza na zadnej strane zariadenia a je možné ho vidieť na Obr.1 prílohy (Príloha A).



Obr1. Panel manuálneho spustenia merania.

Pred manuálnym spustením merania je potrebné najskôr zapojiť výstup zo svoriek do zariadenia, ktorým chcete analógové hodnoty snímačov zrýchlenia získavať. Po zapojení

prepnete prepínač do polohy „Manuál výstup na svorky“. Ak je prepínač v tejto polohe, tak sa analógové hodnoty budú odosielať na výstupné svorky. Meranie spustíte stlačením tlačidla „Spustiť meranie“. Spustením tlačidla sa spustí celý chod merania s čakaním pred spustením nápravy z najvyššej polohy. **Aj v prípade manuálneho merania je potrebné zapojiť USB port nachádzajúci sa na prednej strane aspoň na napájací zdroj 5 V DC.**

Automatické meranie

Celý systém je navrhnutý predovšetkým ako automatický systém merania ovládaný aplikáciou riadenia fyzikálneho modelu.

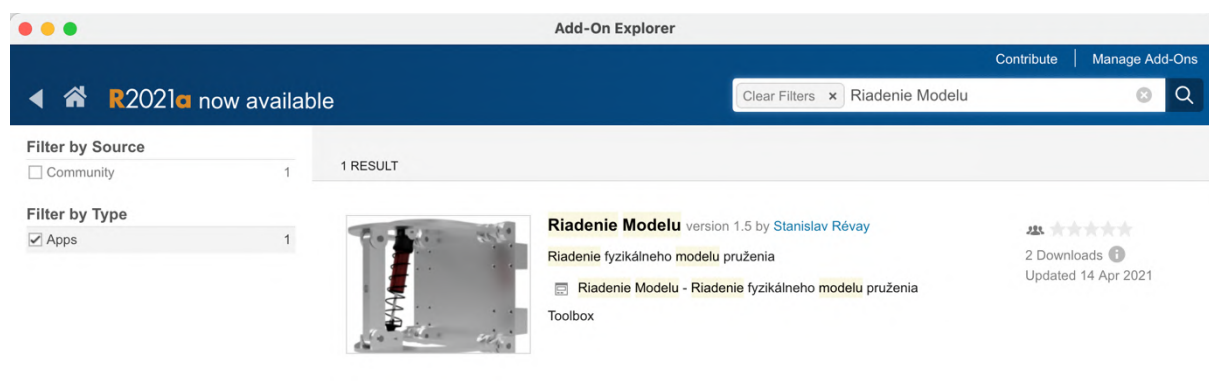
Systémové požiadavky

Aplikáciu riadenia modelu je možné nainštalovať na akýkoľvek operačný systém. Je nevyhnutné, aby bola na počítači nainštalovaná aplikácia **Matlab** verzie **R2020b** a vyššej. Softvér Matlab musí obsahovať balík Curve Fitting Toolbox.

Druhou podmienkou je dostupnosť komunikačných sériových portov. Pre správnu komunikáciu fyzikálneho modelu s počítačom je potrebné mať nainštalovaný ovládač CH340G (len pre OS Windows). Ovládač je dostupný z online zdroja na stránke arduino.eu spolu s návodom na inštaláciu: <https://www.arduined.eu/ch340-windows-10-driver-download/>.

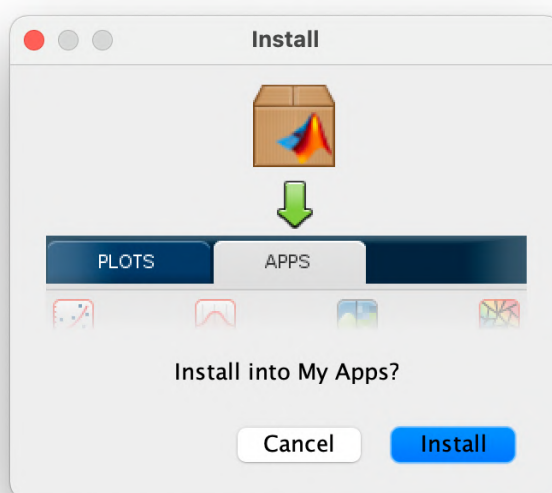
Inštalácia aplikácie Riadenie Modelu

Po spustení programu Matlab sa preklikneme do záložky „Apps“. V tejto záložke sa nachádza v ľavom hornom rohu tlačidlo „Get More Apps“. Stlačením tohto tlačidla otvoríme prehliadač aplikácií Matlabu – Add-On Explorer a do poľa vyhľadávania napíšeme „Riadenie Modelu“ tak, ako je zobrazené na Obr.2 tejto prílohy.



Obr. 2 Vyhľadanie aplikácie v prehliadači aplikácií Add-On Explorer

Otvoríme záložku aplikácie a klikneme na tlačidlo „Add“. Zobrazí sa licenčná zmluva, na tejto klikneme na tlačidlo potvrdenia „Agree“. Po potvrdení sa aplikácia stiahne a namiesto tlačidla „Add“ sa zobrazia tlačidlá „Open folder“ a „Manage“. Stlačíme tlačidlo „Open folder“ a otvorí sa nám v adresárovej časti Matlabu adresár s inštalátorom aplikácie s názvom „Riadenie Modelu.mlappinstall“. Klikneme na tento súbor a zobrazí sa nám výzva na inštalovanie aplikácie (Obr. 3).

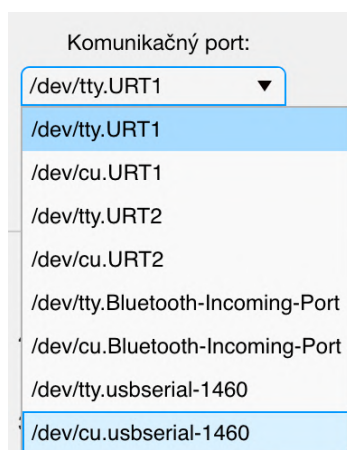


Obr. 3 Výzva na inštaláciu aplikácie

Aplikáciu nainštalujeme stlačením tlačidla „Install“. Aplikácia sa po inštalácii zobrazí v záložke aplikácií v sekcii „My Apps“. Aplikáciu spustíme kliknutím na jej ikonu.

Ovládanie aplikácie

Prvým krokom je pripojenie sa sériovým komunikačným portom k fyzikálnemu modelu. Pred tým je dôležité prepojiť počítač a fyzikálny model káblom. USB Micro B konektor kábla pripojíme do konektora na prednej strane fyzikálneho modelu. Prvou indikáciou pripojenia je rozblíkavie LED na mikropočítači Maker UNO. Ak sme už prepojili fyzikálny model a počítač káblom vytvoríme sériové pripojenie. Najskôr si v aplikácii zvolíme sériový komunikačný port pomocou rozbaľovacieho menu s názvom „Komunikačný port“. V prípade používania OS Windows je názov sériového portu v tvare „COMx“, kde x je číslo komunikačného portu. V prípade, ak aplikácia beží na MAC OS, tak je tvar názvu sériového portu „/dev/cu.usbserial-xxxx“, kde xxxx je štvormiestne číslo portu. Príklad zobrazenia rozbaľovacieho menu sme zobrazili na Obr. 4.



Obr. 4 Rozbaľovacie menu komunikačného portu.

V prípade používania operačného systému MAC OS vidíme, že je k dispozícii ešte jeden rovnaký port s rozdielom, že náš je označený ako „cu“ zatiaľ čo druhý je označený ako „tty“. Je dôležité si zvoliť port označený ako „cu“, pretože ten poskytuje lepšie možnosti komunikácie. Po zvolení správneho komunikačného portu vytvoríme pripojenie stlačením tlačidla „PRIPOJIŤ“, ktoré sa nachádza hneď pod rozbaľovacím menu komunikačného portu. Stav pripojenia môžeme sledovať na indikátore pripojenia umiestneným pod tlačidlom „PRIPOJIŤ“. Na obrázku (Obr. 5) je zobrazený celý panel ovládania.



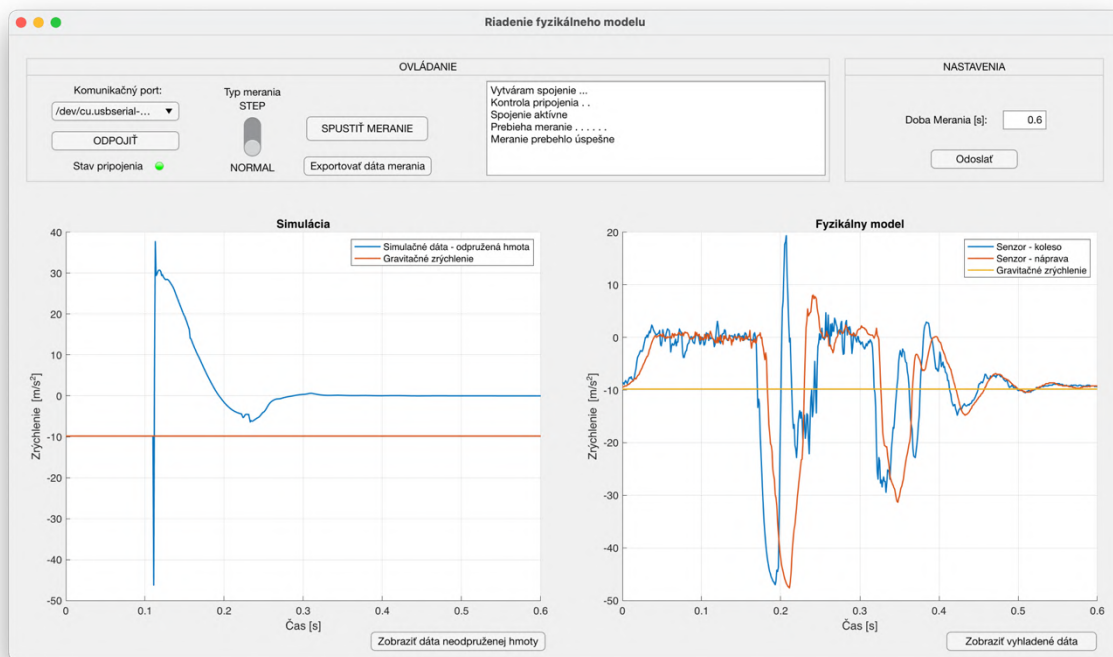
Obr. 5 Panel ovládania aplikácie

Po vytvorení spojenia medzi počítačom a fyzikálnym modelom sa farba indikátora zmení na zelenú a v informačnom okne sa zobrazí informácia „Spojenie aktívne“.

Teraz je možné pristúpiť k spusteniu merania. Aplikácia ponúka možnosť kedykoľvek si zmeniť typ merania. K dispozícii sú dva typy merania:

- **STEP** – meranie s čakaním 200 ms v najvyššej polohe nápravy pre ustálenie hodnôt snímačov zrýchlenia
- **NORMAL** – kontinuálny priebeh merania bez čakania.


Meranie sa spúšťa tlačidlom „SPUSTIŤ MERANIE“. Po spustení vykoná fyzikálny model meranie a po ukončení sa výsledky zobrazia na grafe merania, ktorý je znázornený obrázkom (Obr. 6), na ktorom sa nachádza celé okno aplikácie.



Obr. 6 Okno aplikácie Riadenie fyzikálneho modelu.

Graf „Fyzikálny model“ na pravej strane aplikácie zobrazuje výsledky posledného merania. Na grafe sú farebne rozlíšené krivky merania odpruženej hmoty, neodpruženej hmoty a konštantná krivka gravitačného zrýchlenia. Výsledky merania je možné porovnať s výsledkami simulácie, ktoré sa nachádzajú na grafe „Simulácia“ na ľavej strane aplikácie. Graf simulácie má možnosť prepínania medzi krivkou odpruženej a neodpruženej hmoty. Graf fyzikálneho modelu zas ponúka prepínanie medzi originálnymi hodnotami a vyhladenou krivkou, ktorá ponúka prehľadnejšie zobrazenie kriviek elimináciou rozptylu hodnôt merania. Aplikácia ponúka aj možnosť zmeny doby merania, ktorú je možné meniť na paneli „NASTAVENIA“, ktorý je znázornený na Obr. 7. Dobu merania je možné meniť v rámci intervalu 0,4 – 0,9 sekundy. Optimálna doba merania pri kontinuálnom meraní je 0,6 sekundy. V prípade prepnutia typu merania na STEP (meranie s čakaním) sa automaticky doba merania na grafe predĺži o 200 ms. Zmena hodnoty spočíva v prepísaní doby merania a následným odoslaním tejto hodnoty fyzikálnemu modelu stlačením tlačidla „Odoslať“. Na informačnom

okne sa zobrazí stav odoslania a zápisu novej doby merania. Každé ďalšie merania po zmene doby merania bude vykonané už s novou dĺžkou merania.



NASTAVENIA

Doba Merania [s]:

Odoslať

Obr. 7 Nastavenie doby merania

Po ukončení meraní je vhodné pred zatvorením aplikácie najskôr ukončiť sériovú komunikáciu stlačením tlačidla „ODPOJIŤ“. Po zobrazení informácie „Odpojené“ na informačnom okne je možné aplikáciu vypnúť zatvorením okna aplikácie.

Príloha B: Program aplikácie riadenia a mikropočítača

Aplikácia riadenia fyzikálneho modelu

```
classdef DP_App_exported < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    SuspensionModelAppmatlab.ui.Figure
    ControllPanelmatlab.ui.container.Panel
    KomunikantportDropDownLabelmatlab.ui.control.Label
    SerialPortListmatlab.ui.control.DropDown
    ConnectButtonmatlab.ui.control.Button
    StartMeasurementmatlab.ui.control.Button
    StavpripojeniaLampLabelmatlab.ui.control.Label
    ConnectionStatusLampmatlab.ui.control.Lamp
    StatusAreamatlab.ui.control.TextArea
    TypmeraniaSwitchLabelmatlab.ui.control.Label
    MeasureTypeSwitchmatlab.ui.control.Switch
    OpenExportPanelmatlab.ui.control.Button
    SettingsPanelmatlab.ui.container.Panel
    DobaMeraniaLabelmatlab.ui.control.Label
    MeasTimeEditFieldmatlab.ui.control.NumericEditField
    SendMeasTimeButtonmatlab.ui.control.Button
    SimDataPlotButtonmatlab.ui.control.Button
    SmoothDataButtonmatlab.ui.control.Button
    ExportPanelmatlab.ui.container.Panel
    Imagematlab.ui.control.Image
    UsporiadanieDtLabelmatlab.ui.control.Label
    ExportCSVPanelmatlab.ui.control.Button
    SendToWorkspaceButtonmatlab.ui.control.Button
    CloseButtonmatlab.ui.control.Button
    UIAxesmatlab.ui.control.UIAxes
    UIAxes_2matlab.ui.control.UIAxes
end

properties (Access = public)
    pickedSerialPort = ""; % Description
    status = "";
    message = "";
    timeout = 10;
    dobaMerania = 5;
    measureType = '2';
    device;
    buttonWasPressed = false;
    connected = false;
    values = (zeros(100,2));
    i = 0;
    simulationData_Car;
    simulationData_Wheel;
    simDataVisibility = true; %true = tlmena sustava, false %= netlmerna sustava
    smValues_tlmene;
    smValues_netlmene;
    smoothDataVisible = false;
    AccWasMeasured = false;
    t;
    porovnanieDat = false;
end
```

```

methods (Access = private)

function mes = waitForResponse(app)
    for l = 0:1:app.timeout
        if app.device.NumBytesAvailable > 0
            break;
        end
        pause(1);
    end
    mes = read(app.device,app.device.NumBytesAvailable,'string');
end

function WriteMessage(app, mes)
    app.i = app.i + 1;
    app.status(app.i,:) = mes;
    app.StatusArea.Value = app.status;
    pause(0.1);
    scroll(app.StatusArea, 'bottom');
end

end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
    app.simulationData_Wheel = load('neodAY.mat');
    app.simulationData_Car20 = load('odAY20.mat');
    app.simulationData_Car = load('odAY.mat');
    app.t = load('t.mat');
    set(app.UIAxes,'XLim',[0 0.6]);
    plot(app.UIAxes,(app.t.t)',app.simulationData_Wheel.neodAY,
        (app.t.t)',app.simulationData_Car20.odAY20, [0 length((app.t.t))],
        [-9.81 -9.81], 'LineWidth', 1.5);
    legend(app.UIAxes, "Neodpružená hmota", "Odpružená hmota", "Gravitačné
    zrýchlenie");
end

% Drop down opening function: SerialPortList
function SerialPortListOpening(app, event)
    app.SerialPortList.Items = serialportlist("available");
end

% Value changed function: SerialPortList
function SerialPortListValueChanged(app, event)
    app.pickedSerialPort = app.SerialPortList.Value;
end

% Button pushed function: ConnectButton
function ConnectButtonPushed(app, event)
    app.pickedSerialPort = app.SerialPortList.Value;
    if app.pickedSerialPort ~= "" %Kontrolujem, ci pouzivatel zvolil seriovy
    port
        if app.buttonWasPressed == false
            clear app.device;
            delete(app.device);
            app.ConnectButton.Enable = false;
        end
    end
end

```

```

app.timeout = 10;%Cas pokial cakam na potvrdenie pripojenie
(odoslanie a prijatie cisla 1 ako byte)
app.WriteMessage("Vytváram spojenie ...");
try
app.device = serialport(app.pickedSerialPort,230400);
catch ME
warning('Connection problem!');
app.WriteMessage("Problém s pripojením, skontrolujte
zariadenie");
app.connected = false;
app.ConnectButton.Enable = true;
rethrow(ME);
end

app.WriteMessage("Kontrola pripojenia");
write(app.device, '1', 'char');
pause(1);

for j = 1:app.timeout
if app.device.NumBytesAvailable == 0
write(app.device, '1', 'char');
app.status(app.i,:) = app.status(app.i,:) + ".";
app.StatusArea.Value = app.status;
pause(1);
else
break;
end
end

if app.device.NumBytesAvailable ~= 0
if read(app.device,app.device.NumBytesAvailable, 'char') == '1' %
1-ka ako BYTE je 49
app.ConnectionStatusLamp.Color = 'g';
app.WriteMessage("Spojenie aktívne");
app.ConnectButton.Text = 'ODPOJIŤ';
app.ConnectButton.Enable = true;
app.buttonWasPressed = true;
app.connected = true;
end
else
app.WriteMessage("Vyskytla sa chyba");
app.ConnectButton.Enable = true;
app.ConnectButton.Text = 'PRIPOJIŤ';
app.connected = false;
end
else
clear app.device
delete(app.device);
app.ConnectButton.Text = 'PRIPOJIŤ';
app.buttonWasPressed = false;
app.connected = false;
app.ConnectionStatusLamp.Color = 'r';
app.ConnectButton.Enable = true;
app.WriteMessage("Odpojené");
end
else
app.WriteMessage("Zvoľte komunikačný port");
end

end

```

```

% Button pushed function: StartMeasurement
function StartMeasurementPushed(app, event)
    swValue = app.MeasureTypeSwitch.Value;
    XlimPlot = app.MeasTimeEditField.Value;

    switch swValue
        case 'NORMAL'
            app.measureType = '2';
            XlimPlot = app.MeasTimeEditField.Value;
        case 'STEP'
            app.measureType = '3';
            XlimPlot = app.MeasTimeEditField.Value + 0.2;
        otherwise
            app.measureType = '2';
    end

    end

    if app.connected == true
        flush(app.device);
        write(app.device, app.measureType, 'char');
        app.StartMeasurement.Enable = false;
        app.WriteMessage("Prebieha meranie ");
        for k = 0:1:app.dobaMerania
            app.status(app.i,:) = app.status(app.i,:) + ". ";
            app.StatusArea.Value = app.status;
            pause(1);
        end

        app.message = app.waitForResponse();
        app.values = splitlines(app.message);
        app.values(end) = [];
        app.values = split(app.values);
        app.values = str2double(app.values);
        app.WriteMessage("Meranie prebehlo Úspešne");
        app.AccWasMeasured = true;
        app.StartMeasurement.Enable = true;
        app.smValues_tlmene = smooth(app.values(:,2));
        app.smValues_netlmene = smooth(app.values(:,1));
        set(app.UIAxes_2, 'XLim', [0 XlimPlot]);

        plot(app.UIAxes_2, (0:0.001:((length(app.values)-1)/1000))', app.
            values(:,1), (0:0.001:((length(app.values)-1)/1000))', app.values
            (:,2), [0 ((length(app.values)-1)/1000)], [-9.81 -9.81],
            'LineWidth', 1.5);
        legend(app.UIAxes_2, "Senzor-neodpruženáhmota", "Senzor-odpružená
            hmota", "Gravitačné zrýchlenie");

    else
        app.WriteMessage("Spojenie je neaktívne");
        app.StartMeasurement.Enable = true;
    end

end

% Button pushed function: SendMeasTimeButton
function SendMeasTimeButtonPushed(app, event)
    if app.connected == true
        write(app.device, '0', 'char'); %Nastavenie doby merania
        app.WriteMessage("Zapisujem nové dáta ");
        for j = 1:app.timeout

```

```

        if app.device.NumBytesAvailable == 0
            app.status(app.i,:) = app.status(app.i,:) + ". ";
            app.StatusArea.Value = app.status;
            pause(1);
        else
            break;
        end
    end
end
if app.device.NumBytesAvailable ~= 0
    if read(app.device,app.device.NumBytesAvailable, 'char') == '0' %
1-ka ako BYTE je 49
        valueToSend = (app.MeasTimeEditField.Value-0.2)*10;
        disp(int2str(valueToSend));
        app.dobaMerania = app.MeasTimeEditField.Value + 2;
        write(app.device, int2str(valueToSend), 'char');
        for j = 1:app.timeout
            if app.device.NumBytesAvailable == 0
                app.status(app.i,:) = app.status(app.i,:) + ". ";
                app.StatusArea.Value = app.status;
                pause(1);
            else
                break;
            end
        end
        if app.device.NumBytesAvailable ~= 0
            if read(app.device,app.device.NumBytesAvailable, 'char') ==
int2str(valueToSend)
                app.WriteMessage("Dáta boli zapísané");
            else
                app.WriteMessage("Vyskytla sa chyba");
            end
        end
        end
        else
            app.WriteMessage("Vyskytla sa chyba");
        end
    else
        app.WriteMessage("Spojenie je neaktívne");
    end
end

% Button pushed function: SimDataPlotButton
function SimDataPlotButtonPushed(app, event)
    if app.simDataVisibility

        plot(app.UIAxes, (0:0.001:((length(app.simulationData_Wheel.E1)-
1)/1000)), app.simulationData_Wheel.E1, [0
((length(app.simulationData_Wheel.E1)-1)/1000)], [-9.81 -9.81],
'LineWidth', 1.5);
        legend(app.UIAxes, "Simulačné dáta - Neodpružená hmota", "Gravitačné
zrýchlenie");
        app.simDataVisibility = false;
        app.SimDataPlotButton.Text = "Zobrazíť dáta odpruženej hmoty";
    else

        plot(app.UIAxes, (0:0.001:((length(app.simulationData_Car.E00)-1
)/1000)), app.simulationData_Car.E00, [0
((length(app.simulationData_Car.E00)-1)/1000)], [-9.81 -9.81],
'LineWidth', 1.5);
    end
end

```

```

        legend(app.UIAxes, "Simulačné dáta - Odpružená hmota", "Gravitačné
        zrýchlenie");
        app.simDataVisibility = true;
        app.SimDataPlotButton.Text = "Zobrazit' dáta neodpruženej hmoty";
    end
end

% Button pushed function: SmoothDataButton
function SmoothDataButtonPushed(app, event)
    if app.smoothDataVisible

        plot(app.UIAxes_2, (0:0.001:((length(app.values)-1)/1000))', app.
        values(:,1), (0:0.001:((length(app.values)-1)/1000))', app.values
        (:,2), [0 ((length(app.values)-1)/1000)], [-9.81 -9.81],
        'LineWidth', 1.5)
        legend(app.UIAxes_2, "Senzor-Neodpružená hmota", "Senzor-Odpružená
        hmota", "Gravitačné zrýchlenie");
        app.smoothDataVisible = false;
        app.SmoothDataButton.Text = "Zobrazit' vyhladené dáta";

    else

        plot(app.UIAxes_2, (0:0.001:((length(app.values)-1)/1000))', app.
        smValues_netlmene, (0:0.001:((length(app.values)-1)/1000))', app.
        smValues_tlmene, [0 ((length(app.values)-1)/1000)], [-9.81 -9.81],
        'LineWidth', 1.5)
        legend(app.UIAxes_2, "Senzor-Neodpružená hmota", "Senzor-Odpružená
        hmota", "Gravitačné zrýchlenie");
        app.smoothDataVisible = true;
        app.SmoothDataButton.Text = "Zobrazit' originálne dáta";

    end
end

% Button pushed function: ExportCSVPanel
function EportCSVtoFile(app, event)
    if app.AccWasMeasured
        [file,path] = uiputfile('Meranie.csv');
        filename = fullfile(path,file);
        valuesToExport = [(0:0.001:((length(app.values)-1)/1000))' app.values];
        csvwrite(filename, valuesToExport);
    else
        app.WriteMessage("Dáta na export nie sú k dispozícii, vykonajte meranie");
    end
end

% Button pushed function: OpenExportPanel
function OpenExportPanelButtonPushed(app, event)
    app.ExportPanel.Visible = true;
end

% Button pushed function: CloseButton
function CloseButtonPushed(app, event)
    app.ExportPanel.Visible = false;
end

% Button pushed function: SendToWorkspaceButton
function SendToWorkspaceButtonPushed(app, event)
    assignin('base', 'DataMerania',
    [(0:0.001:((length(app.values)-1)/1000))' app.values])
end
end

```

```

function CompareButtonPushed(app, event)
    app.porovnanieDat = ~app.porovnanieDat;
    if app.porovnanieDat
        app.SimDataPlotButton.Enable = true;
        plot(app.UIAxes, (app.t.t)', app.simulationData_Car.odAY,
            (0:0.001:((length(app.values)-1)/1000))', app.values(:,2), [0
            length((app.t.t)')], [-9.81 -9.81], 'LineWidth', 1.5);
        legend(app.UIAxes, "Odpružená hmota - simulácia", "Odpružená hmota
        - meranie", "Gravitačné zrýchlenie");
        app.simDataVisibility = true;
        app.SimDataPlotButton.Text = "Zobrazit' dáta neodpruženej hmoty";
        app.CompareButton.Text = "Zobrazit' dáta simulácie";

    else
        app.SimDataPlotButton.Enable = false;
        set(app.UIAxes, 'XLim', [0 0.6]);
        plot(app.UIAxes, (app.t.t)', app.simulationData_Wheel.neodAY,
            (app.t.t)', app.simulationData_Car20.odAY20, [0
            length((app.t.t)')], [-9.81 -9.81], 'LineWidth', 1.5);
        legend(app.UIAxes, "Neodpružená hmota", "Odpružená hmota",
            "Gravitačné zrýchlenie");
        app.CompareButton.Text = "Zobrazit' porovnanie";

    end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create SuspensionModelApp and hide until all components are created
app.SuspensionModelApp = uifigure('Visible', 'off');
app.SuspensionModelApp.Position = [100 100 1280 720];
app.SuspensionModelApp.Name = 'Riadenie fyzikálneho modelu';
app.SuspensionModelApp.Icon = 'Ikona.png';

% Create ControllPanel
app.ControllPanel = uipanel(app.SuspensionModelApp);
app.ControllPanel.TitlePosition = 'centertop';
app.ControllPanel.Title = 'OVLÁDANIE';
app.ControllPanel.Position = [21 552 924 142];

% Create KomunikanportDropDownLabel
app.KomunikanportDropDownLabel = uilabel(app.ControllPanel);
app.KomunikanportDropDownLabel.HorizontalAlignment = 'right';
app.KomunikanportDropDownLabel.Position = [49 94 107 22];
app.KomunikanportDropDownLabel.Text = 'Komunikačný port: ';

% Create SerialPortList
app.SerialPortList = uidropdown(app.ControllPanel);
app.SerialPortList.Items = {};
app.SerialPortList.DropDownOpeningFcn = createCallbackFcn(app,
@SerialPortListOpening, true);
app.SerialPortList.ValueChangedFcn = createCallbackFcn(app,
@SerialPortListValueChanged, true);
app.SerialPortList.Position = [29 70 147 22];
app.SerialPortList.Value = {};

% Create ConnectButton
app.ConnectButton = uibutton(app.ControllPanel, 'push');

```



```

app.ConnectButton.ButtonPushedFcn = createCallbackFcn(app,
@ConnectButtonPushed, true);
app.ConnectButton.Position = [29 36 147 22];
app.ConnectButton.Text = 'PRIPOJITĚ';

% Create StartMeasurement
app.StartMeasurement = uibutton(app.ControllPanel, 'push');
app.StartMeasurement.ButtonPushedFcn = createCallbackFcn(app,
@StartMeasurementPushed, true);
app.StartMeasurement.Position = [322 47 140 28];
app.StartMeasurement.Text = 'SPUSTITĚ MERANIE';

% Create StavpripojeniaLampLabel
app.StavpripojeniaLampLabel = uilabel(app.ControllPanel);
app.StavpripojeniaLampLabel.HorizontalAlignment = 'right';
app.StavpripojeniaLampLabel.Position = [48 7 86 22];
app.StavpripojeniaLampLabel.Text = 'Stav pripojenia';

% Create ConnectionStatusLamp
app.ConnectionStatusLamp = uilamp(app.ControllPanel);
app.ConnectionStatusLamp.Position = [148 13 10 10];
app.ConnectionStatusLamp.Color = [1 0 0];

% Create StatusArea
app.StatusArea = uitextarea(app.ControllPanel);
app.StatusArea.Position = [529 7 390 109];

% Create TypmeraniaSwitchLabel
app.TypmeraniaSwitchLabel = uilabel(app.ControllPanel);
app.TypmeraniaSwitchLabel.HorizontalAlignment = 'center';
app.TypmeraniaSwitchLabel.Position = [225 92 70 22];
app.TypmeraniaSwitchLabel.Text = 'Typ merania';

% Create MeasureTypeSwitch
app.MeasureTypeSwitch = uiswitch(app.ControllPanel, 'slider');
app.MeasureTypeSwitch.Items = {'NORMAL', 'STEP'};
app.MeasureTypeSwitch.Orientation = 'vertical';
app.MeasureTypeSwitch.Tooltip = {'Voľba typumerania: ' '- STEP - meranie s čakáním
pred zjazdom z obrubníka'; '- NORMAL - kontinuálne meranie'};
app.MeasureTypeSwitch.Position = [250 29 20 45];
app.MeasureTypeSwitch.Value = 'NORMAL';

% Create OpenExportPanel
app.OpenExportPanel = uibutton(app.ControllPanel, 'push');
app.OpenExportPanel.ButtonPushedFcn = createCallbackFcn(app,
@OpenExportPanelButtonPushed, true);
app.OpenExportPanel.Position = [319 7 147 22];
app.OpenExportPanel.Text = 'Exportovať dáta merania';

% Create SettingsPanel
app.SettingsPanel = uipanel(app.SuspensionModelApp);
app.SettingsPanel.TitlePosition = 'centertop';
app.SettingsPanel.Title = 'NASTAVENIA';
app.SettingsPanel.Position = [962 552 298 142];

% Create DobaMeraniasLabel
app.DobaMeraniasLabel = uilabel(app.SettingsPanel);
app.DobaMeraniasLabel.HorizontalAlignment = 'right';
app.DobaMeraniasLabel.Position = [64 60 100 22];
app.DobaMeraniasLabel.Text = 'Doba Merania [s]:';

```

```

% Create MeasTimeEditField
app.MeasTimeEditField = uicontrol(app.SettingsPanel, 'numeric');
app.MeasTimeEditField.Limits = [0.3 1];
app.MeasTimeEditField.ValueDisplayFormat = '%5.1f';
app.MeasTimeEditField.Position = [182 60 50 22];
app.MeasTimeEditField.Value = 0.6;

% Create SendMeasTimeButton
app.SendMeasTimeButton = uicontrol(app.SettingsPanel, 'push');
app.SendMeasTimeButton.ButtonPushedFcn = createCallbackFcn(app,
@SendMeasTimeButtonPushed, true);
app.SendMeasTimeButton.Position = [99 15 100 22];
app.SendMeasTimeButton.Text = 'Odoslať';

% Create SimDataPlotButton
app.SimDataPlotButton = uicontrol(app.SuspensionModelApp, 'push');
app.SimDataPlotButton.ButtonPushedFcn = createCallbackFcn(app,
@SimDataPlotButtonPushed, true);
app.SimDataPlotButton.Position = [417 13 201 22];
app.SimDataPlotButton.Text = 'Zobraziť dáta neodpruženej hmoty';

% Create SmoothDataButton
app.SmoothDataButton = uicontrol(app.SuspensionModelApp, 'push');
app.SmoothDataButton.ButtonPushedFcn = createCallbackFcn(app,
@SmoothDataButtonPushed, true);
app.SmoothDataButton.Position = [1079 13 173 22];
app.SmoothDataButton.Text = 'Zobraziť vyhladené dáta';

% Create ExportPanel
app.ExportPanel = uipanel(app.SuspensionModelApp);
app.ExportPanel.TitlePosition = 'centertop';
app.ExportPanel.Title = 'Export';
app.ExportPanel.Visible = 'off';
app.ExportPanel.BackgroundColor = [0.9412 0.9412 0.9412];
app.ExportPanel.Position = [377 250 527 223];

% Create Image
app.Image = uicontrol(app.ExportPanel);
app.Image.Position = [15 75 497 100];
app.Image.ImageSource = 'MatlabVysvetlivkaCSV.png';

% Create UsporiadanieDtLabel
app.UsporiadanieDtLabel = uicontrol(app.ExportPanel);
app.UsporiadanieDtLabel.Position = [15 174 97 22];
app.UsporiadanieDtLabel.Text = 'Usporiadanie dát';

% Create ExportCSVPanel
app.ExportCSVPanel = uicontrol(app.ExportPanel, 'push');
app.ExportCSVPanel.ButtonPushedFcn = createCallbackFcn(app, @EportCSVtoFile,
true);
app.ExportCSVPanel.Position = [173 37 132 22];
app.ExportCSVPanel.Text = 'Exportovať do súboru';

% Create SendToWorkspaceButton
app.SendToWorkspaceButton = uicontrol(app.ExportPanel, 'push');
app.SendToWorkspaceButton.ButtonPushedFcn = createCallbackFcn(app,
@SendToWorkspaceButtonPushed, true);
app.SendToWorkspaceButton.Position = [15 37 148 22];
app.SendToWorkspaceButton.Text = 'Odoslať do "Workspace"';

```

```

% Create CloseButton
app.CloseButton = uibutton(app.ExportPanel, 'push');
app.CloseButton.ButtonPushedFcn = createCallbackFcn(app, @CloseButtonPushed,
true);
app.CloseButton.Position = [422 5 100 22];
app.CloseButton.Text = 'Zatvor';

% Create UIAxes
app.UIAxes = uiaxes(app.SuspensionModelApp);
title(app.UIAxes, 'Simulácia')
xlabel(app.UIAxes, 'Čas [s]')
ylabel(app.UIAxes, 'Zrýchlenie [m/s^2]')
zlabel(app.UIAxes, 'Z')
app.UIAxes.PlotBoxAspectRatio = [1.29009433962264 1 1];
app.UIAxes.XGrid = 'on';
app.UIAxes.YGrid = 'on';
app.UIAxes.Position = [21 34 600 480];

% Create UIAxes_2
app.UIAxes_2 = uiaxes(app.SuspensionModelApp);
title(app.UIAxes_2, 'Fyzikálny model')
xlabel(app.UIAxes_2, 'Čas [s]')
ylabel(app.UIAxes_2, 'Zrýchlenie [m/s^2]')
zlabel(app.UIAxes_2, 'Z')
app.UIAxes_2.PlotBoxAspectRatio = [1.29009433962264 1 1];
app.UIAxes_2.XGrid = 'on';
app.UIAxes_2.YGrid = 'on';
app.UIAxes_2.Position = [661 34 600 480];

% Show the figure after all components are created
app.SuspensionModelApp.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = DP_App_exported

runningApp = getRunningApp(app);

% Check for running singleton app
if isempty(runningApp)

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.SuspensionModelApp)

% Execute the startup function
runStartupFcn(app, @startupFcn)
else

% Focus the running singleton app
figure(runningApp.SuspensionModelApp)

```

```
app = runningApp;
end

if nargin == 0
clear app
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.SuspensionModelApp)
end
end
end
```

Program aplikácie mikropočítača

```
#define stepsPerRevolution 400
#define dirPin 11 //CW+
#define stepPin 10 //CLK+
#define sensor 8 //senzor polohy
#define prepinac 12 //Prepinac manualneho modu

//Premenne na ovladanie motora
char serialCode = '0'; //Byte!!!
int serialCodeInt = 0; //Integer
int motorSpeed = 1600;
int distance = 0;

//Premenne na ovladanie (meranie) akcelerometrov
int i = 0;
const int Z1_axis = A0;
const int Z2_axis = A1;
int z1_adc_value = 0;
int z2_adc_value = 0;
float z1_g_value = 0;
float z2_g_value = 0;
int dobaMerania = 400;
bool zmenaDobyMerania = false;

void setup() {

    pinMode(9,OUTPUT);//Mam zapojeny aj tento PIN,
    digitalWrite(9,LOW);//preto tam dam pre istotu 0V
    pinMode(sensor,INPUT_PULLUP); //INPUT_PULLUP kvoli ruseniam
    pinMode(prepinac,INPUT_PULLUP);
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
    digitalWrite(dirPin, HIGH); //urcenie smeru otacania

    Serial.begin(230400); //POZOR NA BaudRate
}

void loop() {

    //Nastavenie a spustanie merania seriovou komunikaciou
    if (Serial.available()) {
        serialCode = Serial.read();
        serialCodeInt = (int)(serialCode-'0');
        if(serialCodeInt == 1) {
            Serial.print(1);//Odosielam cislo je pre potvrdenie prijatia
            prikazu
        }

        if(serialCodeInt == 2) { //Kontinualne meranie bez prerusenia
            ContinualMeasure();
        }

        if(serialCodeInt == 3){ //Meranie s ustalenim sa hodnot pred padom
        koleasa
            dobaMerania = dobaMerania + 100;
            StepMeasure();
        }

        if(zmenaDobyMerania == true){
            dobaMerania = serialCodeInt*100;
            zmenaDobyMerania = false;
        }
    }
}
```

```

        Serial.print(serialCodeInt); //Odosielam cislo je pre
        potvrdenie prijatia prikazu
    }

    if(serialCodeInt == 0){
        zmenaDobyMerania = true;
        Serial.print(0); //Odosielam cislo je pre potvrdenie prijatia
        prikazu
    }
}

//Spustenie merania tlacidlom
if(digitalRead(prepinac) == 0){
    delay(30);
    if(digitalRead(prepinac) == 0){
        dobaMerania = dobaMerania + 100;
        StepMeasure();
    }
}
}

void ContinualMeasure() {
    preparePosition(); //Priprava polohy vacky
    //Zaciatok meranie... namiesto delayMicroseconds(motorSpeed*0.4) davam
    meranie, ktore pri motorSpeed = 1600 trva rovnako dlho
    measure();
    delayMicroseconds(250);
    while (distance <= 100) { // nasledne sa otacam rychlo a umoznim kolesu
    volny pad
        digitalWrite(stepPin, HIGH);
        measure();
        digitalWrite(stepPin, LOW);
        measure();
        distance = distance + 1;
    }

    while ( i < dobaMerania) {
        measure();
        i++;
        delayMicroseconds(250);
    }
    i = 0;
}

void StepMeasure(){
    preparePosition();
    //Cakanie na ustalenej hodnote
    i = 0;
    while(i < 200){ //Doba trvania Serial.print je cca 750us + 250us = 1ms
        measure();
        delayMicroseconds(250);
        i++;
    }
    i = 0;

    while (distance <= 100) { // nasledne sa otacam rychlo a umoznim kolesu
    volny pad
        digitalWrite(stepPin, HIGH);
        measure();
        digitalWrite(stepPin, LOW);
        measure();
        distance = distance + 1;
    }
}

```

```

    }

    while ( i < dobaMerania) {
        measure();
        i++;
        delayMicroseconds(250);
    }
    i = 0;
}

void preparePosition() {

    if(digitalRead(sensor) == 0){ //Odchod z Home pozicie
        while (digitalRead(sensor) == 0) { // otacam sa pomaly, kym neodidem
            zo snimaca
            digitalWrite(stepPin, HIGH);
            delayMicroseconds(motorSpeed);
            digitalWrite(stepPin, LOW);
            delayMicroseconds(motorSpeed);
        }
    }
    while (digitalRead(sensor) == 1) { // otacam sa pomaly, kym nenarazim na
        snimac
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(motorSpeed);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(motorSpeed);
    }
    distance = 0;
}

void measure(){
    z1_adc_value = analogRead(Z1_axis);
    z2_adc_value = analogRead(Z2_axis);

    //Acceleration in G
    z1_g_value = -((float)z1_adc_value - 328)/67*9.81;//Meranim som zistil,
    zeacc9.81naosiZdosiahnemprihodnote397aacc0prihodnote330...397-330=67
    z2_g_value = -((float)z2_adc_value - 326)/67*9.81;
    Serial.println(String( String(z1_g_value) + " " + String(z2_g_value)));
}

```

Príloha C: Technická dokumentácia

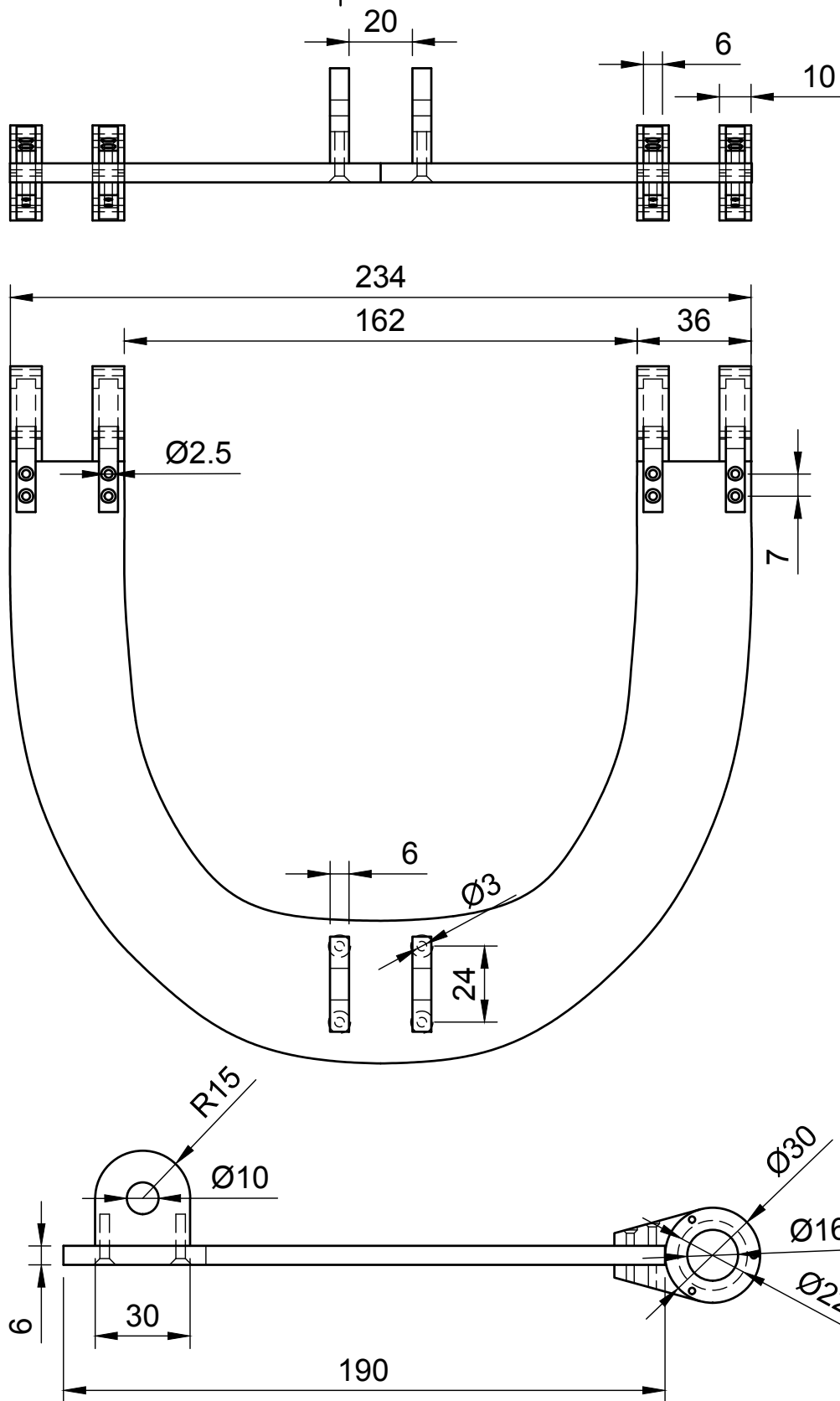
Analýza kmitania na štvrtinovom fyzikálnom modeli pruženia automobilu

Technická dokumentácia
k diplomovej práci

Autor práce: Bc. Stanislav Révay

Vedúci práce: doc. Ing. Vladimír Goga, PhD.

2020/2021



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu

Horné rameno nápravy

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-HRN

DÁTUM VYHOTOVENIA:
24.09.2020

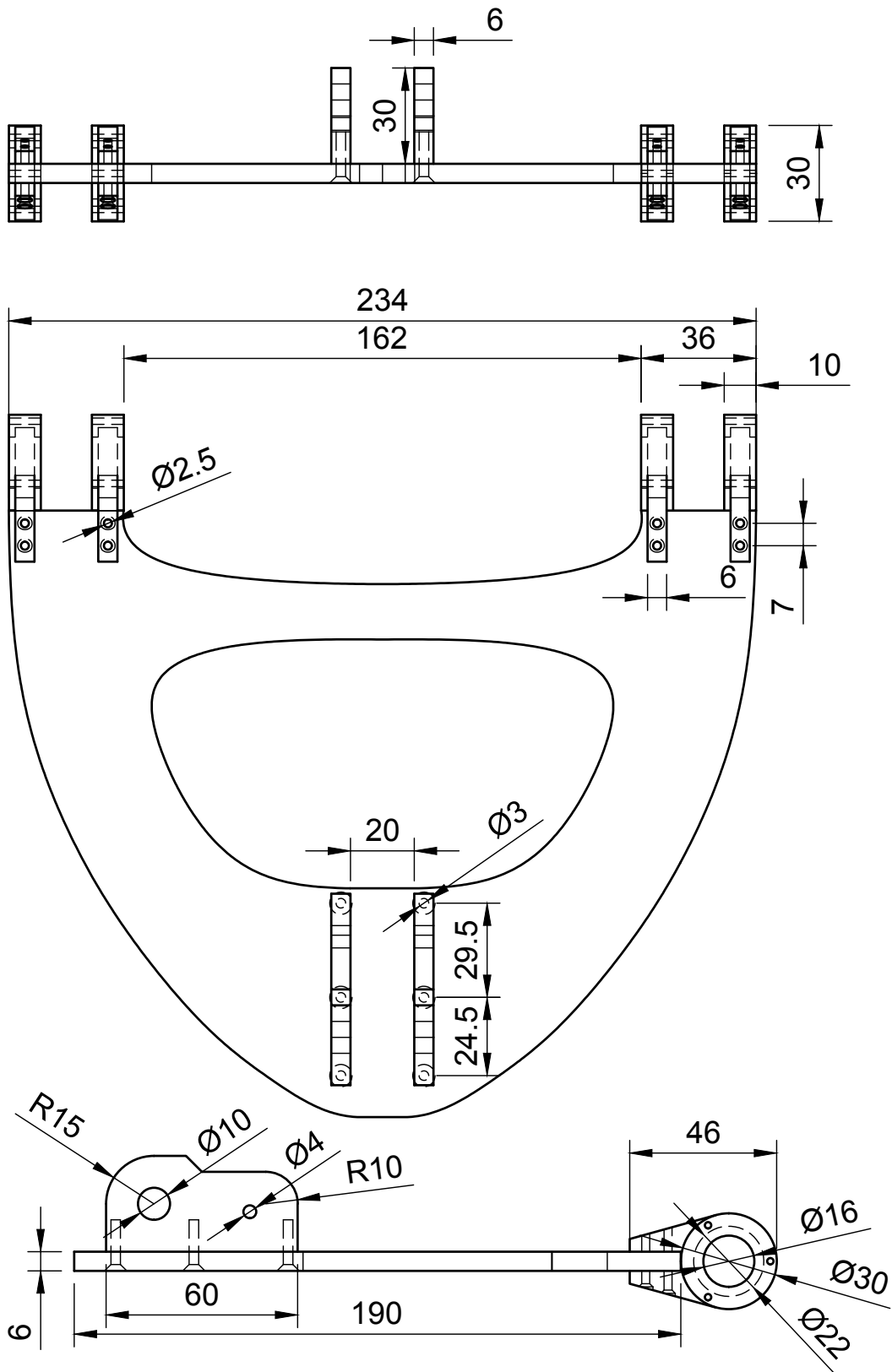
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:2

HMOTNOSŤ:
0,42 kg

MATERIÁL:
ALU 7075

LIST Č.:
1/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu

Dolné rameno nápravy

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-DRN

DÁTUM VYHOTOVENIA:
24.09.2020

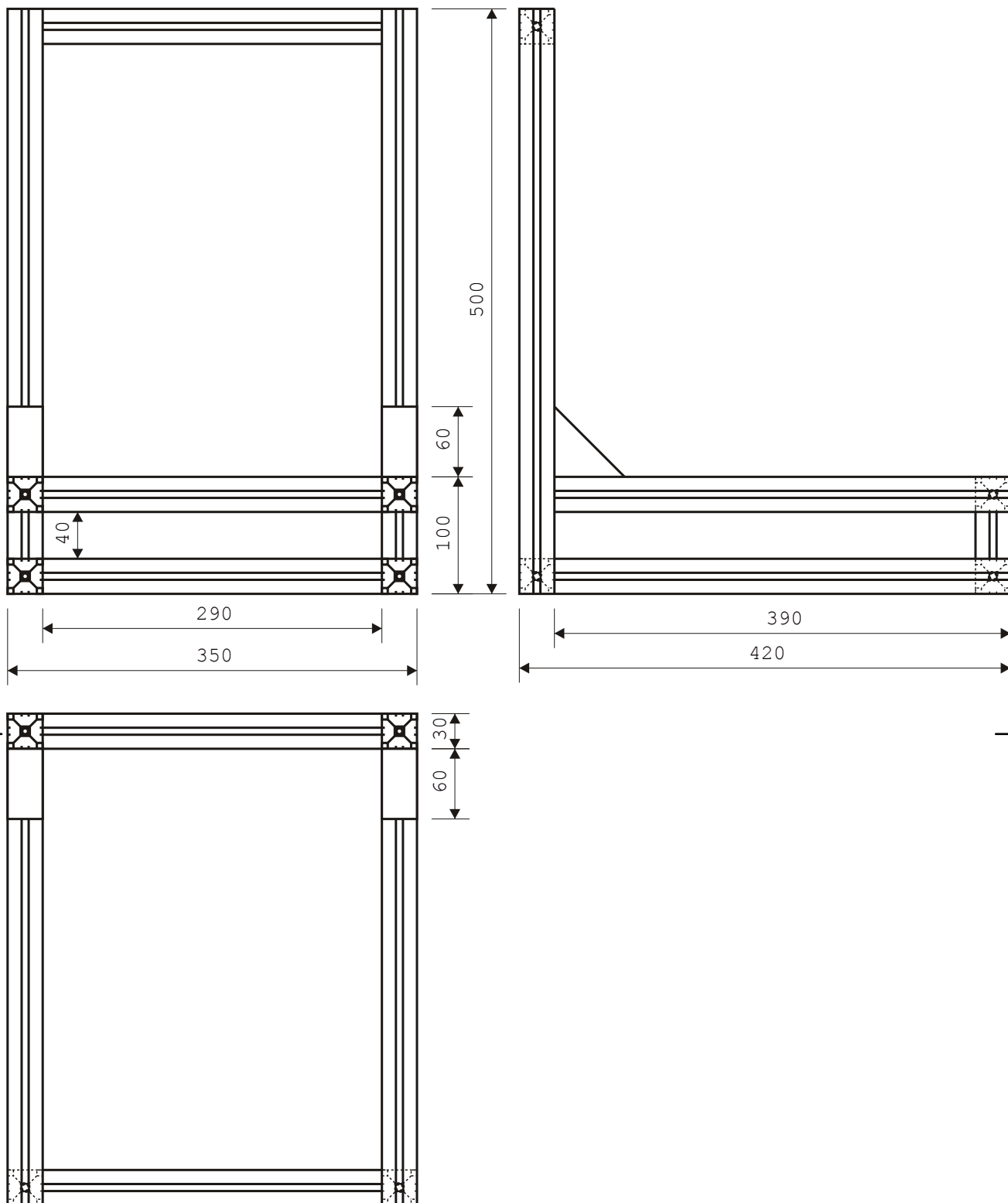
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:2

HMOTNOSŤ:
0,54kg

MATERIÁL:
ALU 7075

LIST Č.:
2/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Nosná konštrukcia

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-NKA

DÁTUM VYHOTOVENIA:
24.09.2020

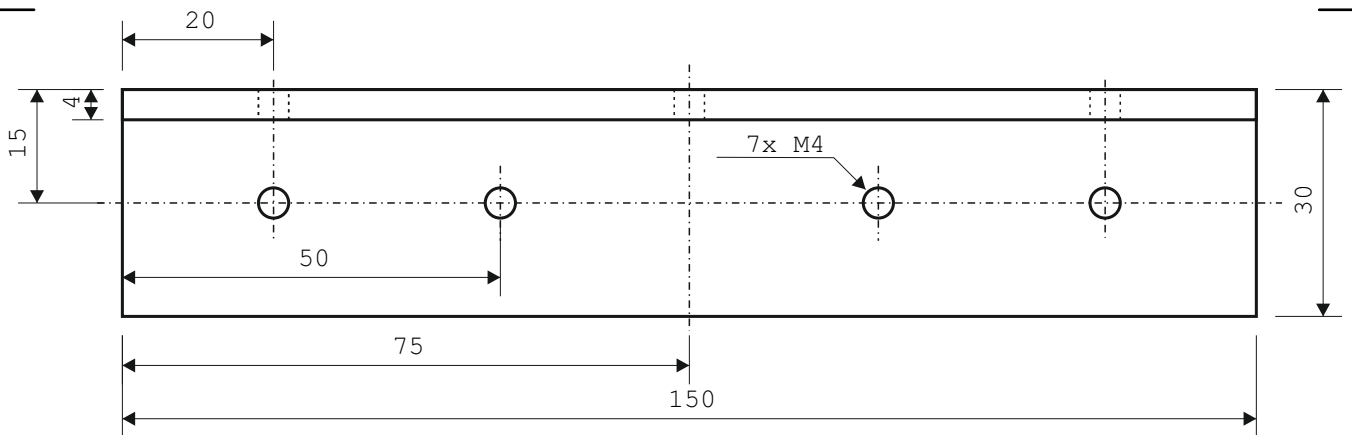
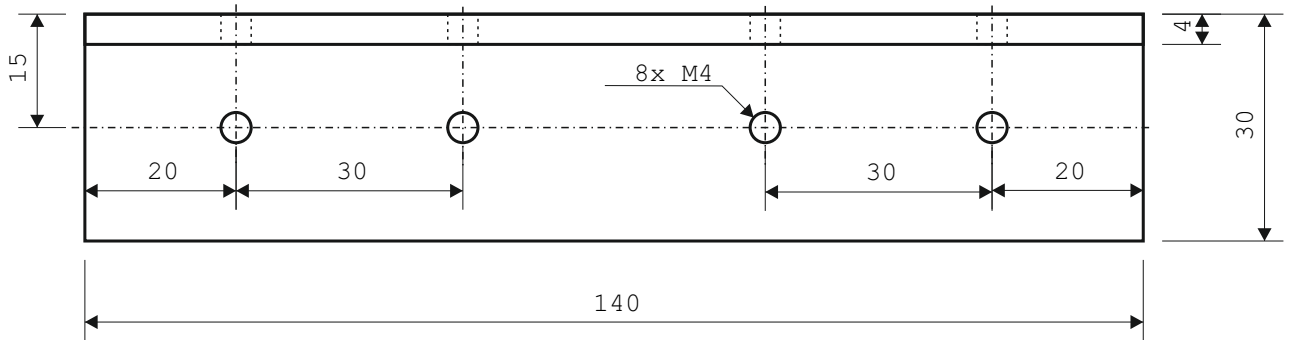
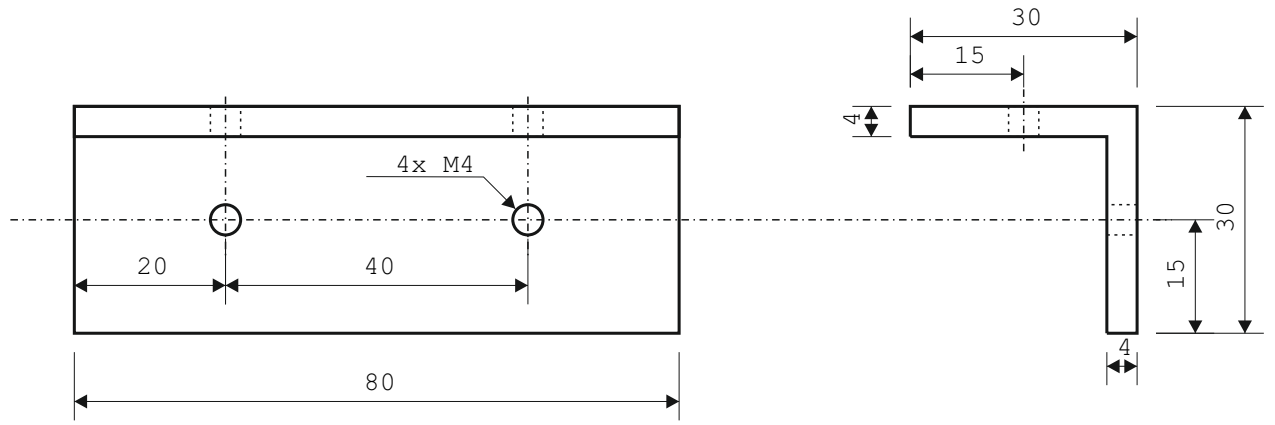
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:5

HMOTNOSŤ:
3,496kg

MATERIÁL:
ALU 7075

LIST Č.:
3/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

**Fyzikálny model pruženia automobilu
L-profily**

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-LPR

DÁTUM VYHOTOVENIA:
24.09.2020

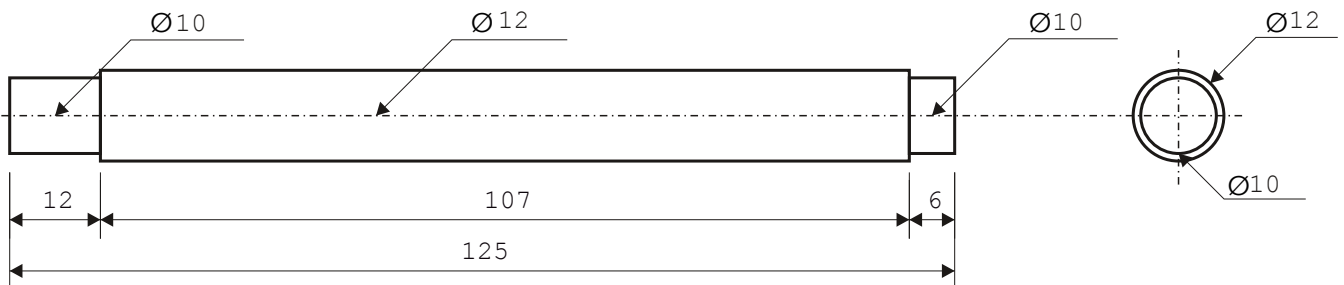
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:1

HMOTNOSŤ:
0,24kg

MATERIÁL:
AL

LIST Č.:
4/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

**Fyzikálny model pruženia automobilu
Hriadel'**

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-HR

DÁTUM VYHOTOVENIA:
24.09.2020

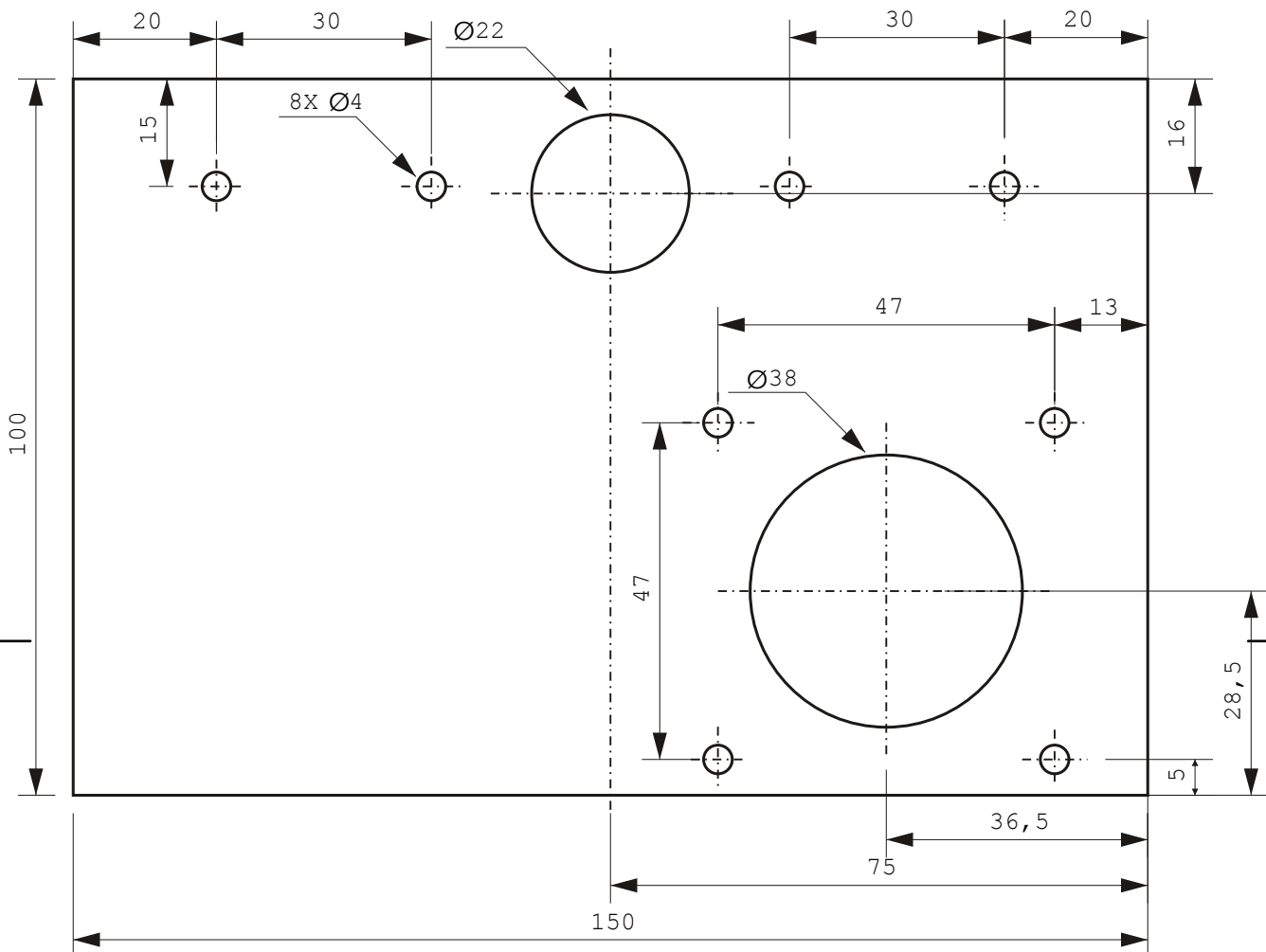
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:1

HMOTNOSŤ:
0,11kg

MATERIÁL:
Oceľ

LIST Č.:
5/18



Hrúbka materiálu 6 mm



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Uchytenie hnacej sústavy

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-UHSUS

DÁTUM VYHOTOVENIA:
24.09.2020

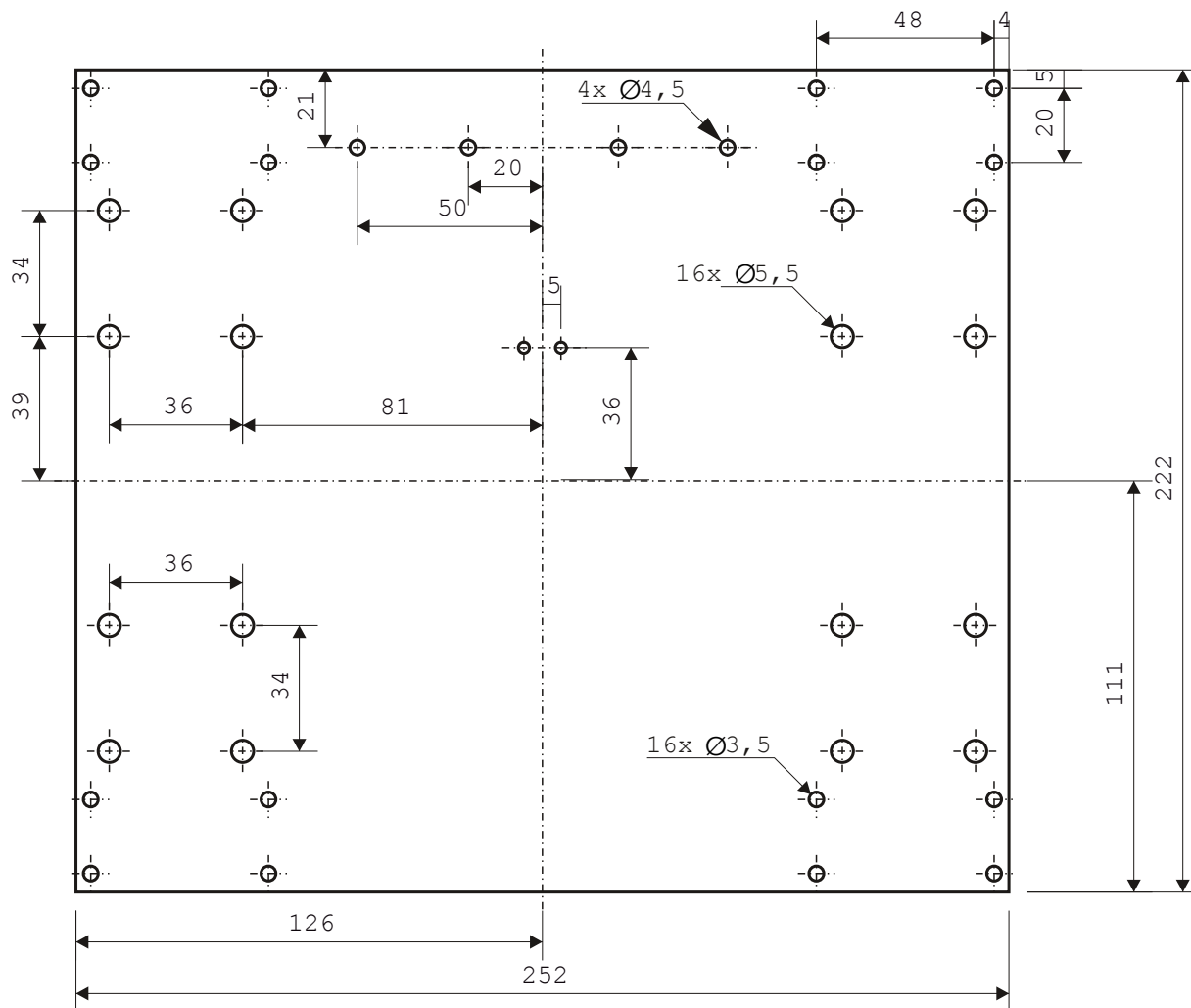
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:1

HMOTNOSŤ:
0,18 kg

MATERIÁL:
ALU 7075

LIST Č.:
6/18



Hrúbka materiálu 3 mm



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Zadná doska

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-ZD

DÁTUM VYHOTOVENIA:
24.09.2020

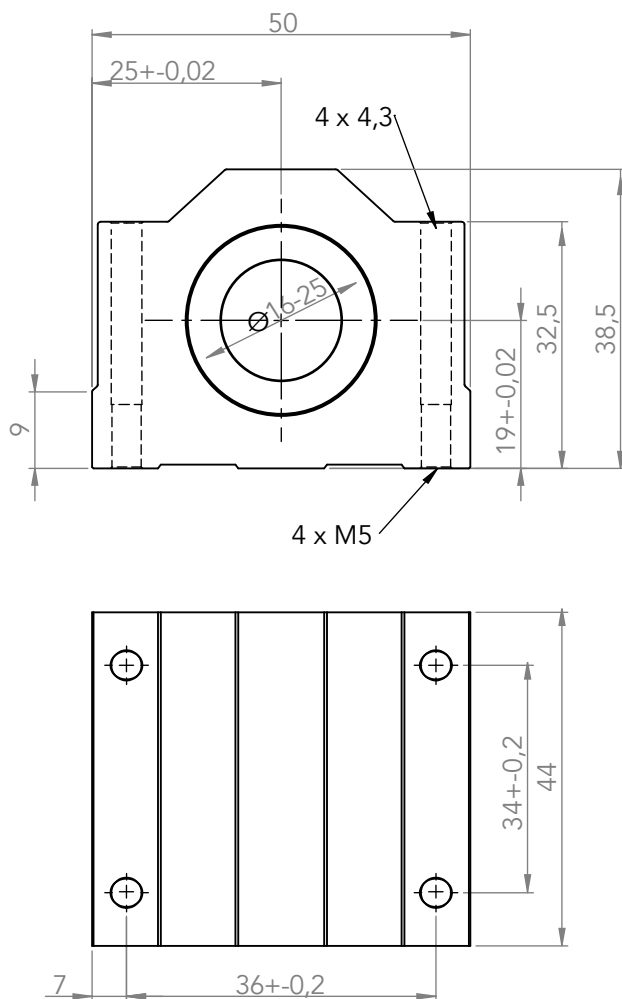
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:2

HMOTNOSŤ:
0,64 kg

MATERIÁL:
ALU 7075

LIST Č.:
8/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Vozík lineárneho vedenia

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-VLV

DÁTUM VYHOTOVENIA:
24.09.2020

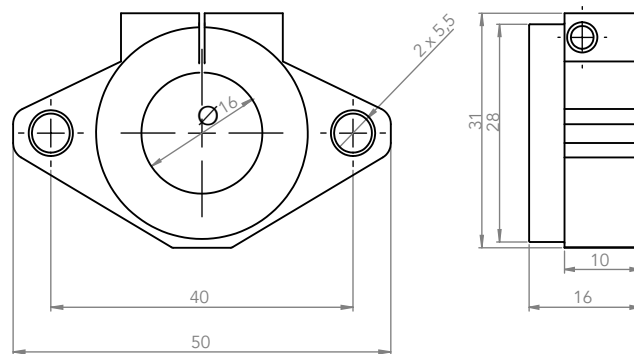
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:1

HMOTNOSŤ:
0,18 kg

MATERIÁL:
AL

LIST Č.:
9/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
 FAKULTA ELEKTROTECHNIKY
 A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Držiak lineárneho vedenia

VYPRACOVAL:
 Bc. Stanislav Révay

FORMÁT:
 A4

ČÍSLO VÝKRESU:
 FM-DLV

DÁTUM VYHOTOVENIA:
 24.09.2020

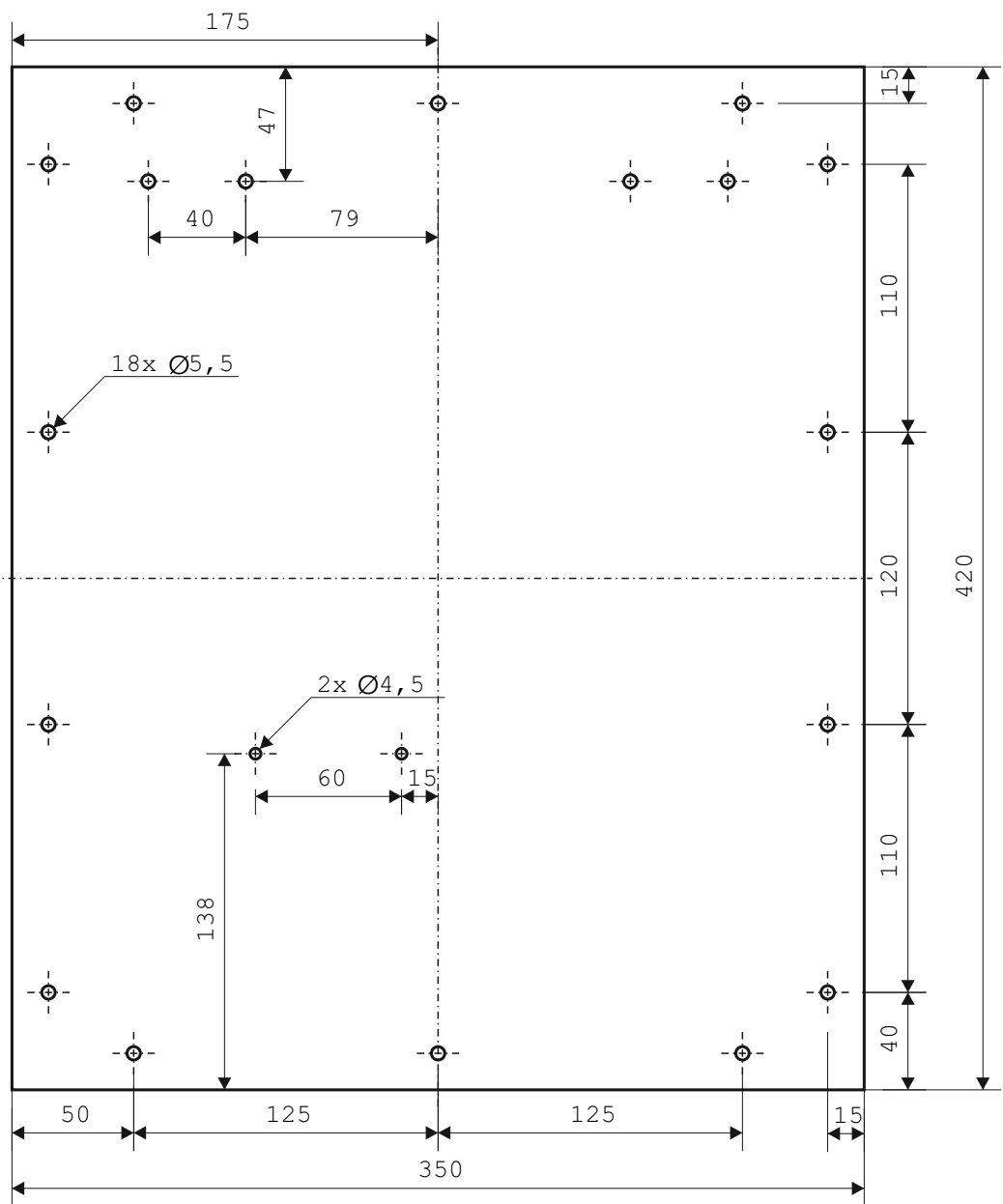
SCHVÁLIL:
 doc. Ing. Vladimír Goga, PhD.

MIERKA:
 1:1

HMOTNOSŤ:
 0,027 kg

MATERIÁL:
 AL

LIST Č.:
 10/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Základová doska

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-ZDOS

DÁTUM VYHOTOVENIA:
24.09.2020

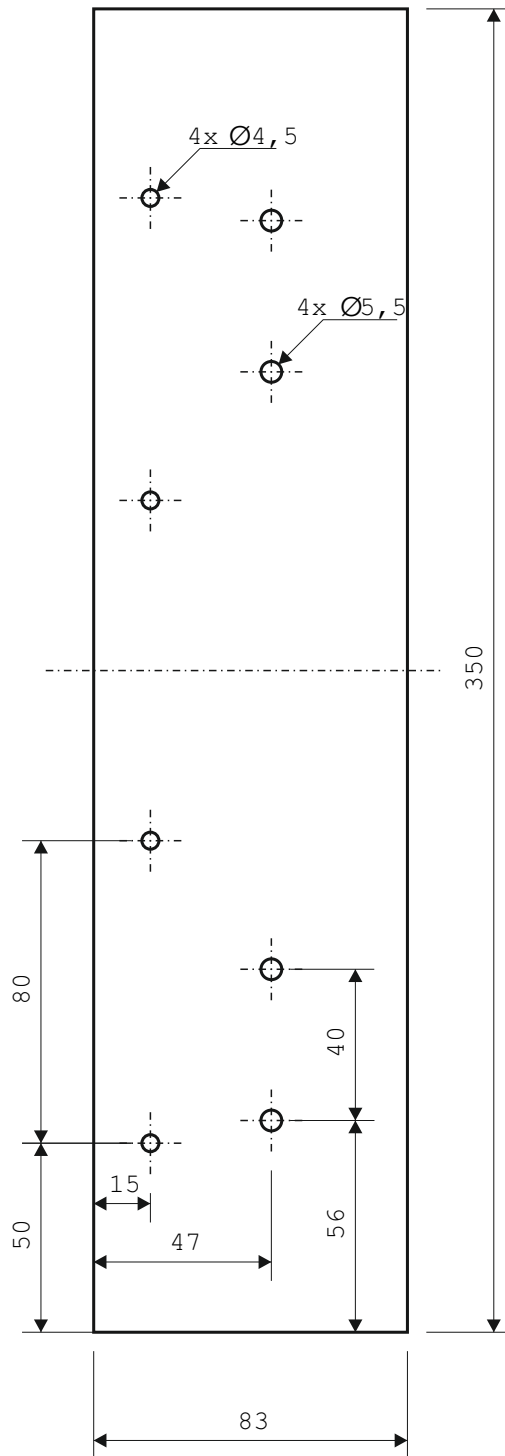
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:3

HMOTNOSŤ:
0,79 kg

MATERIÁL:
AL

LIST Č.:
11/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Čelná doska

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-CDOS

DÁTUM VYHOTOVENIA:
24.09.2020

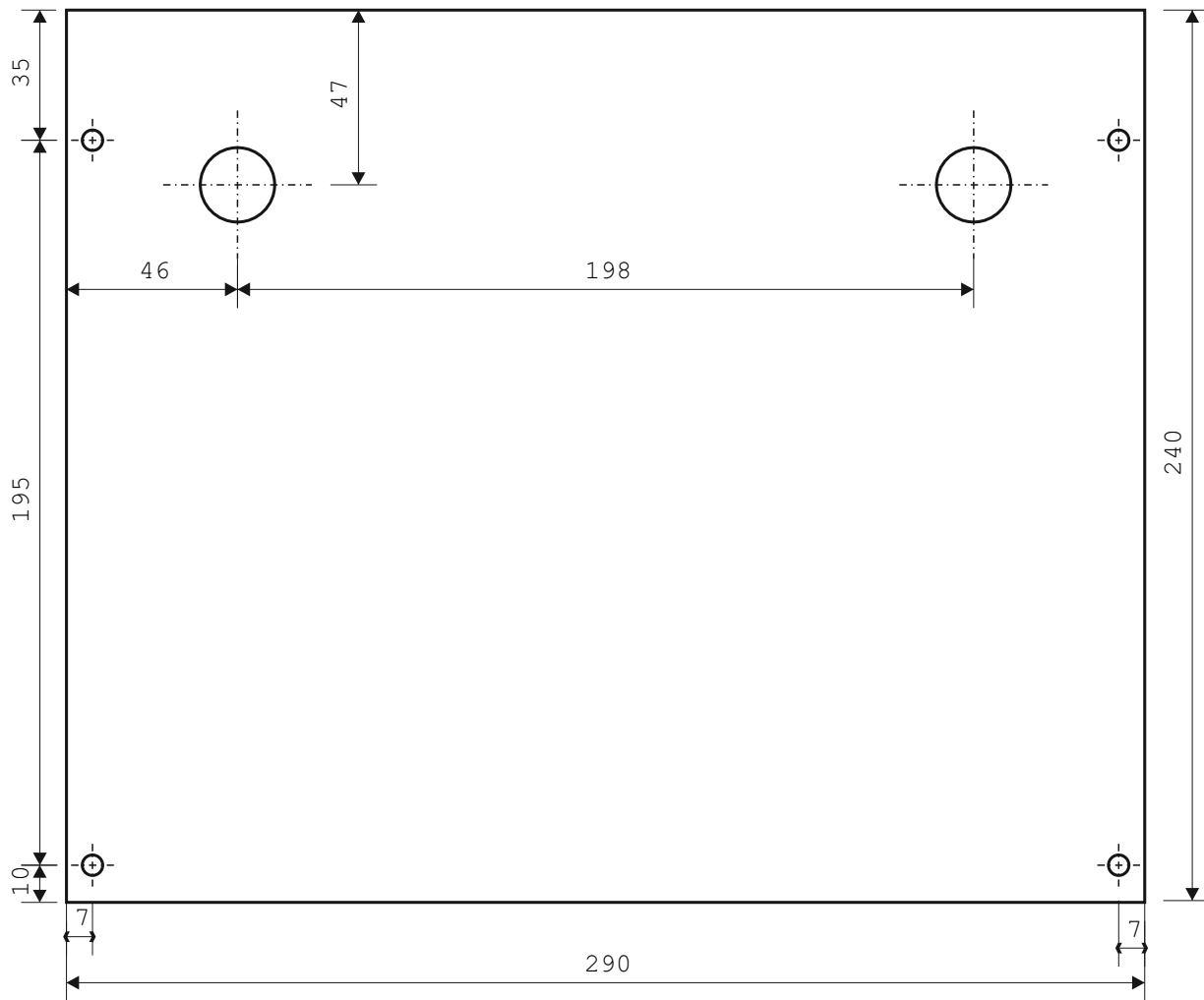
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:2

HMOTNOSŤ:
0,2kg

MATERIÁL:
AL

LIST Č.:
12/18



Hrúbka materiálu 4mm



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

**Fyzikálny model pruženia automobilu
Priehľadný kryt elektroniky**

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-KEPLEX

DÁTUM VYHOTOVENIA:
24.09.2020

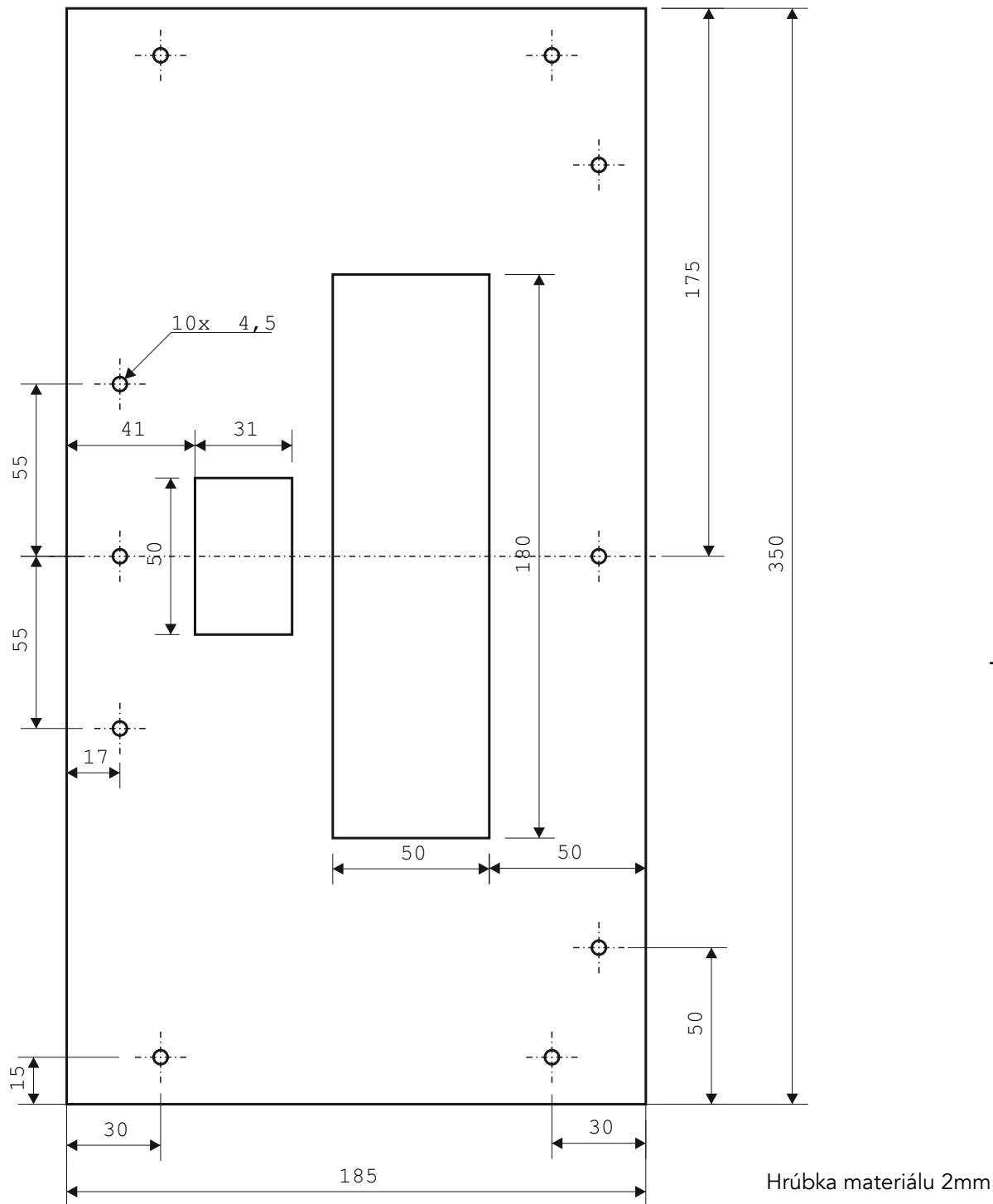
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:2

HMOTNOSŤ:
0,1kg

MATERIÁL:
Plexisklo

LIST Č.:
13/18



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Kryt hnacej sústavy

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-KHS

DÁTUM VYHOTOVENIA:
24.09.2020

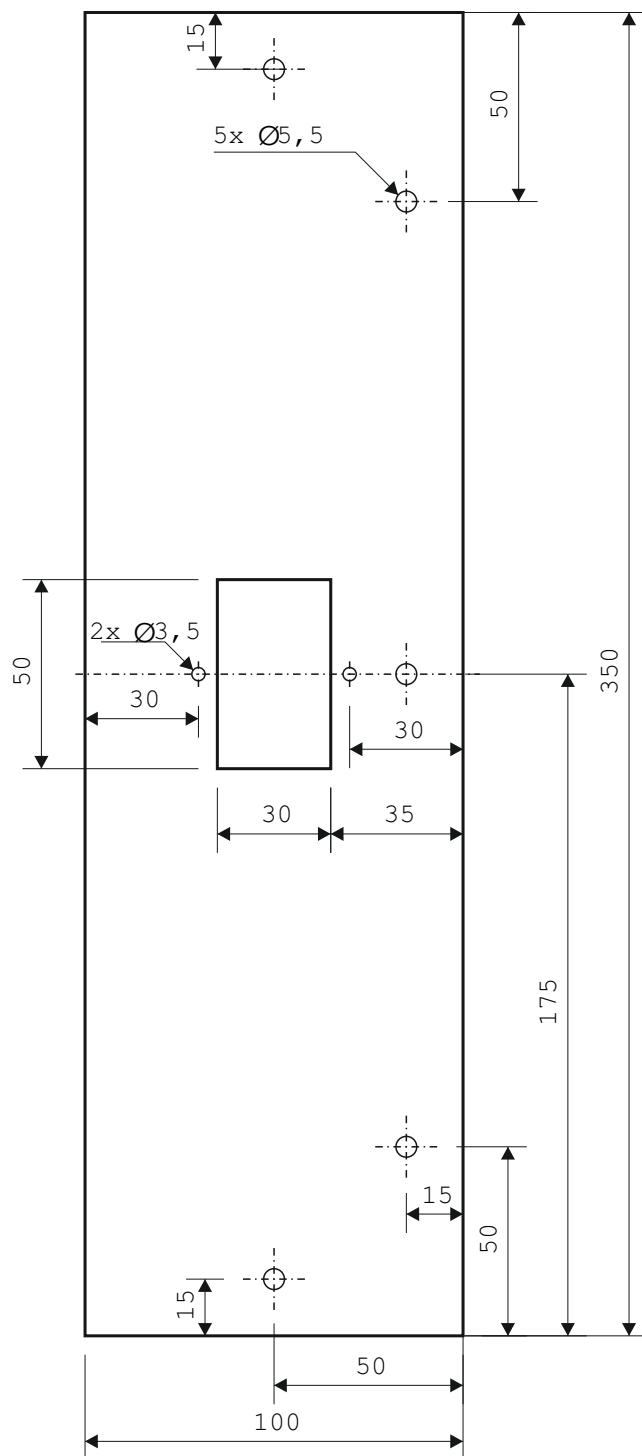
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:2

HMOTNOSŤ:
0,3kg

MATERIÁL:
AL

LIST Č.:
14/18



Hrúbka materiálu 2mm



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Zadný kryt - Al

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-ZKAL

DÁTUM VYHOTOVENIA:
24.09.2020

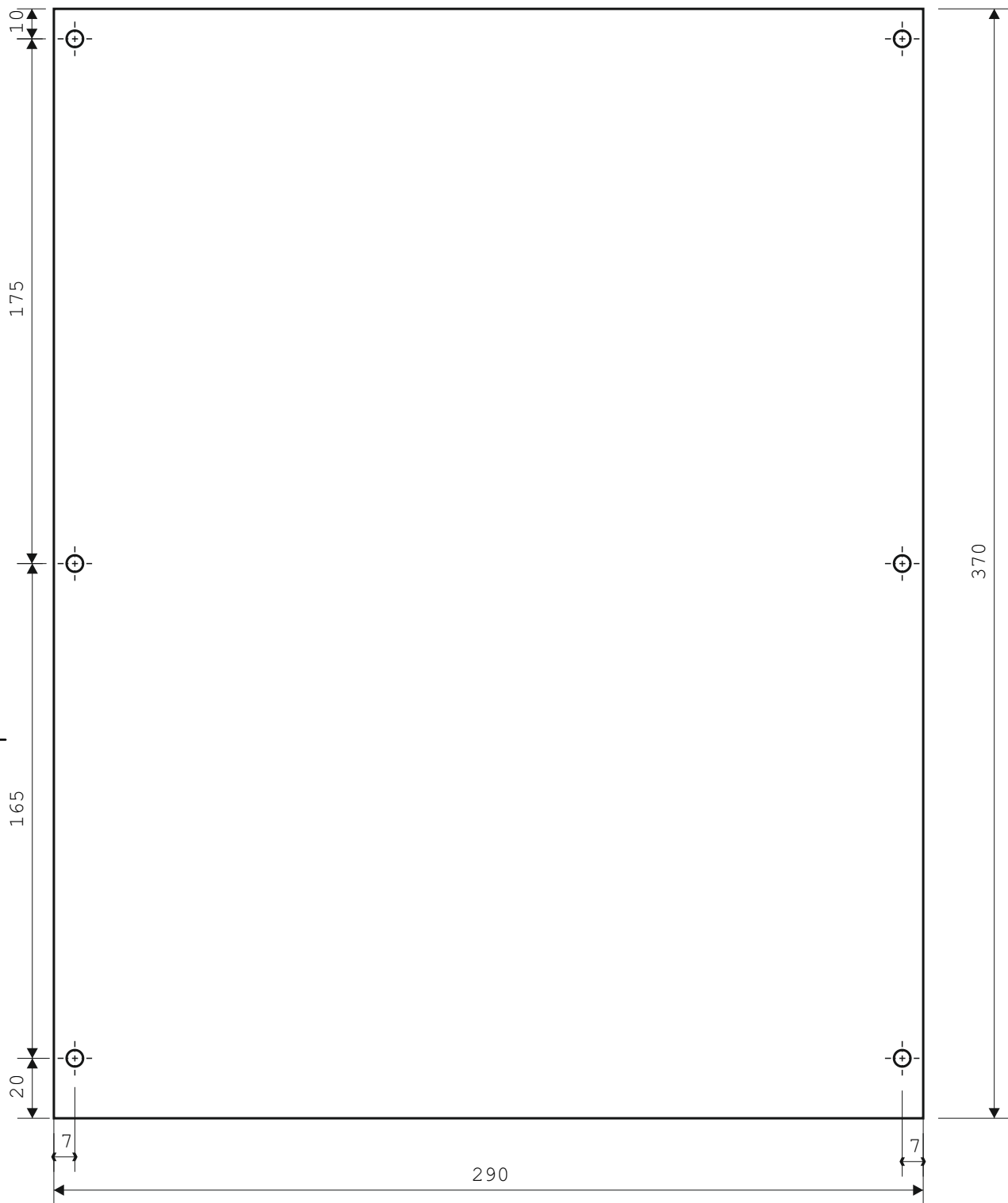
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:2

HMOTNOSŤ:
0,18kg

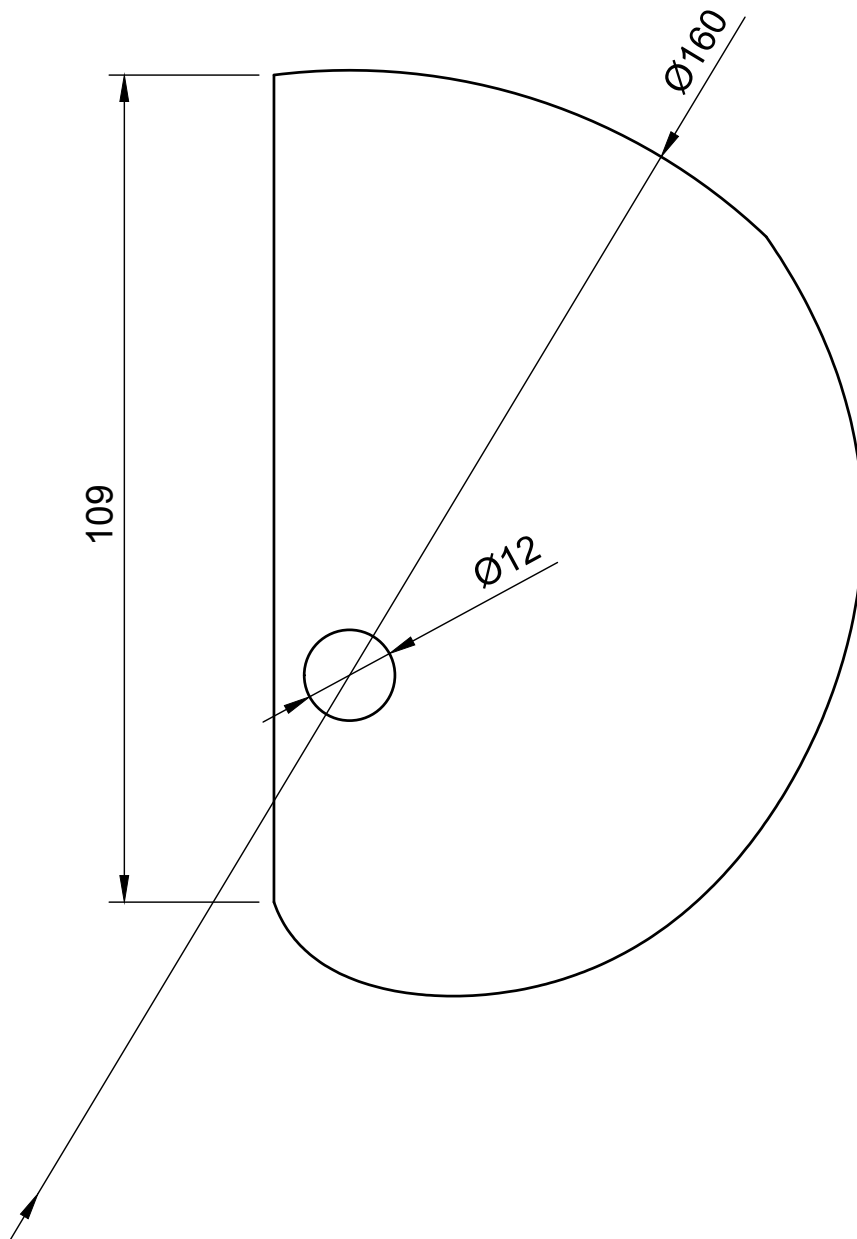
MATERIÁL:
AL

LIST Č.:
15/18



Hrúbka materiálu 4mm

 <p>SLOVENSKÁ TECHNICKÁ UNIVERZITA FAKULTA ELEKTROTECHNIKY A INFORMATIKY</p>	<p>ÚSTAV AUTOMOBILOVEJ MECHATRONIKY</p>			
	<p>NÁZOV VÝKRESU: Fyzikálny model pruženia automobilu Zadný kryt - Plexisklo</p>			
<p>VYPRACOVAL: Bc. Stanislav Révay</p>	<p>FORMÁT: A4</p>	<p>ČÍSLO VÝKRESU: FM-ZKPLEX</p>	<p>DÁTUM VYHOTOVENIA: 24.09.2020</p>	
<p>SCHVÁLIL: doc. Ing. Vladimír Goga, PhD.</p>	<p>MIERKA: 1:2</p>	<p>HMOTNOSŤ: 0,16kg</p>	<p>MATERIÁL: Plexisklo</p>	<p>LIST Č.: 16/18</p>



Hrúbka materiálu 40mm



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Vačka

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-VAC

DÁTUM VYHOTOVENIA:
24.09.2020

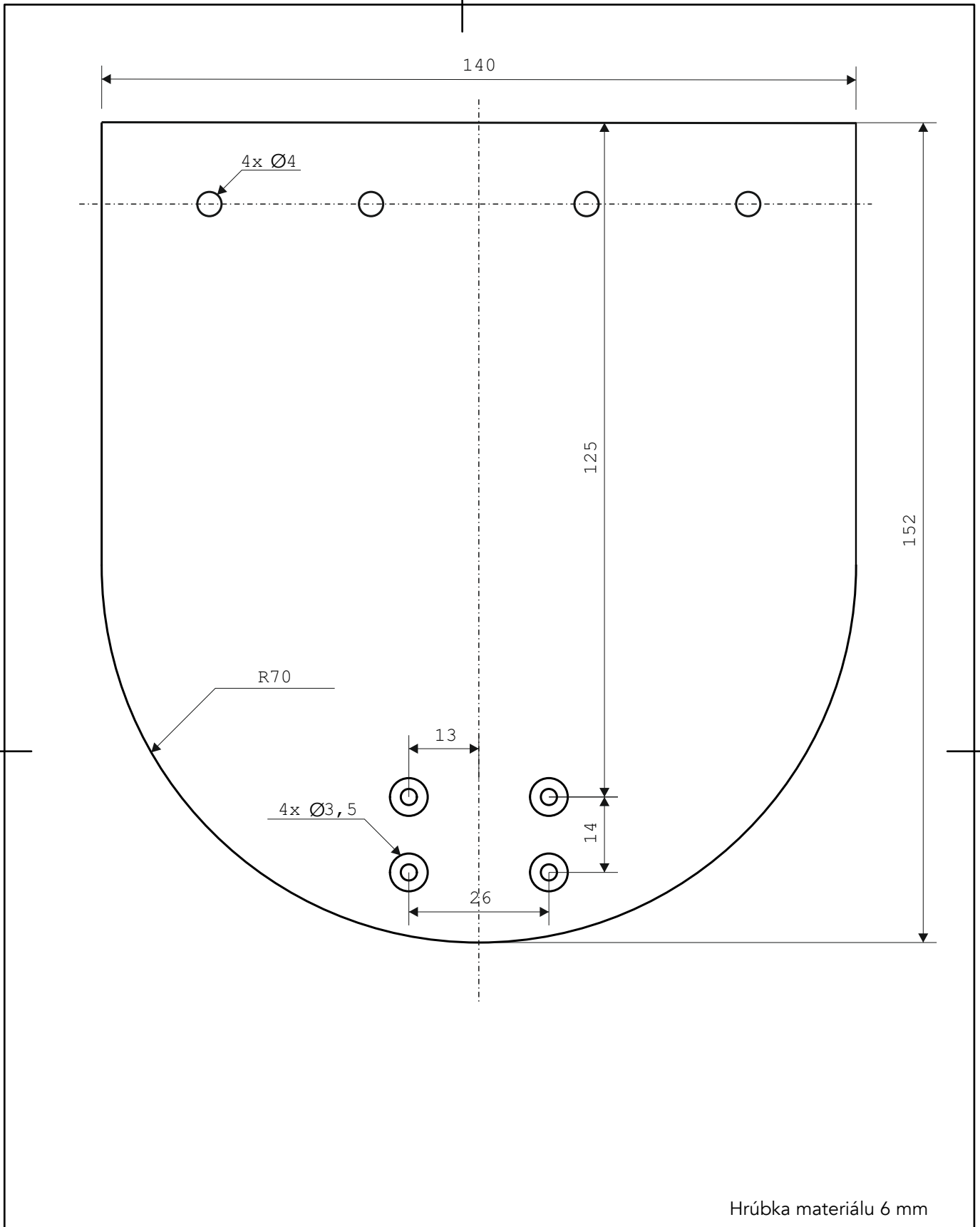
SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

MIERKA:
1:1

HMOTNOSŤ:
0,54kg

MATERIÁL:
Orechové drevo

LIST Č.:
17/18



Hrúbka materiálu 6 mm



SLOVENSKÁ TECHNICKÁ UNIVERZITA
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

ÚSTAV AUTOMOBILOVEJ MECHATRONIKY

NÁZOV VÝKRESU:

Fyzikálny model pruženia automobilu
Uloženie tlmíča

VYPRACOVAL:
Bc. Stanislav Révay

FORMÁT:
A4

ČÍSLO VÝKRESU:
FM-UT

DÁTUM VYHOTOVENIA:
24.09.2020

SCHVÁLIL:
doc. Ing. Vladimír Goga, PhD.

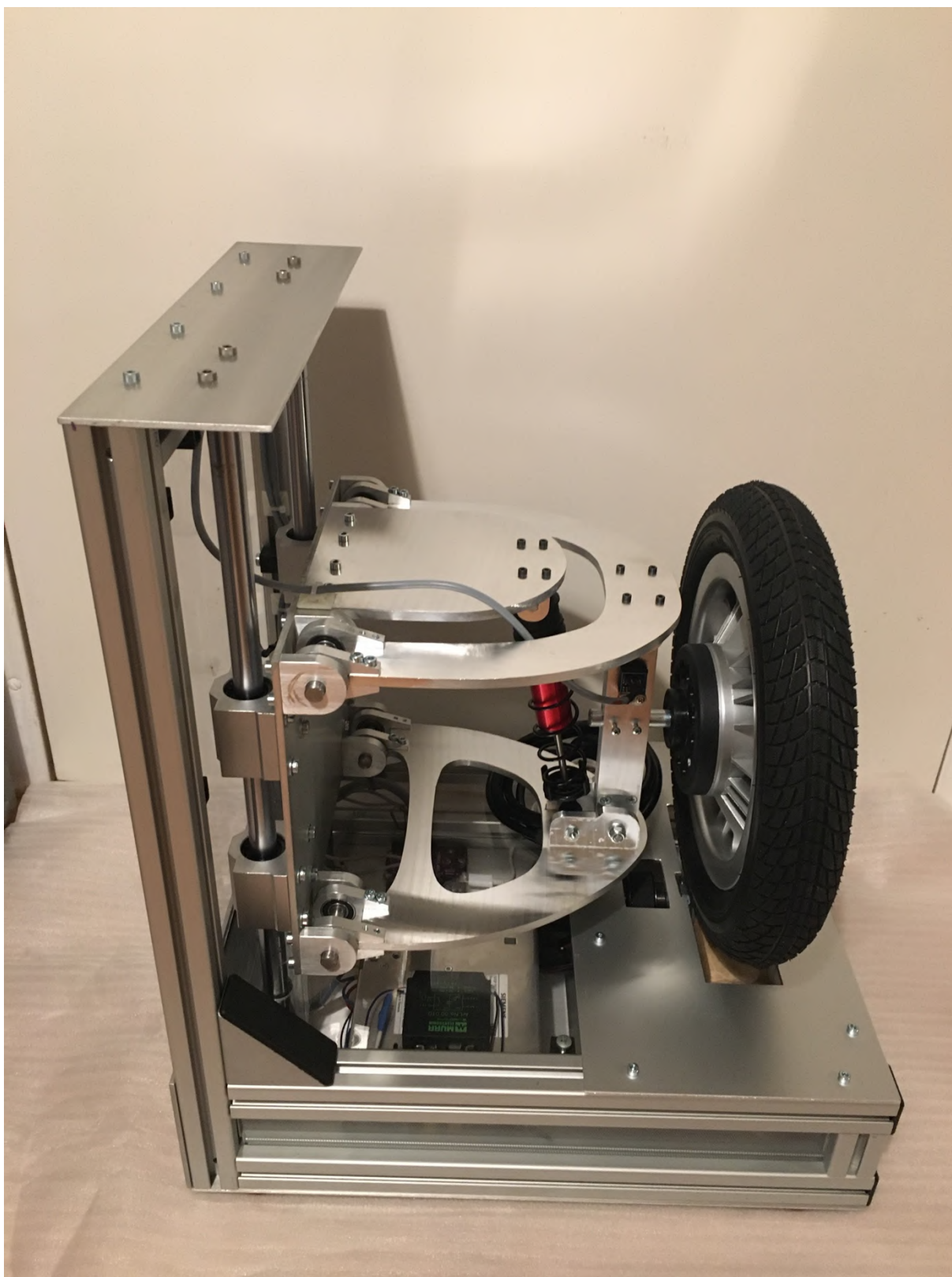
MIERKA:
1:1

HMOTNOSŤ:
0,64 kg

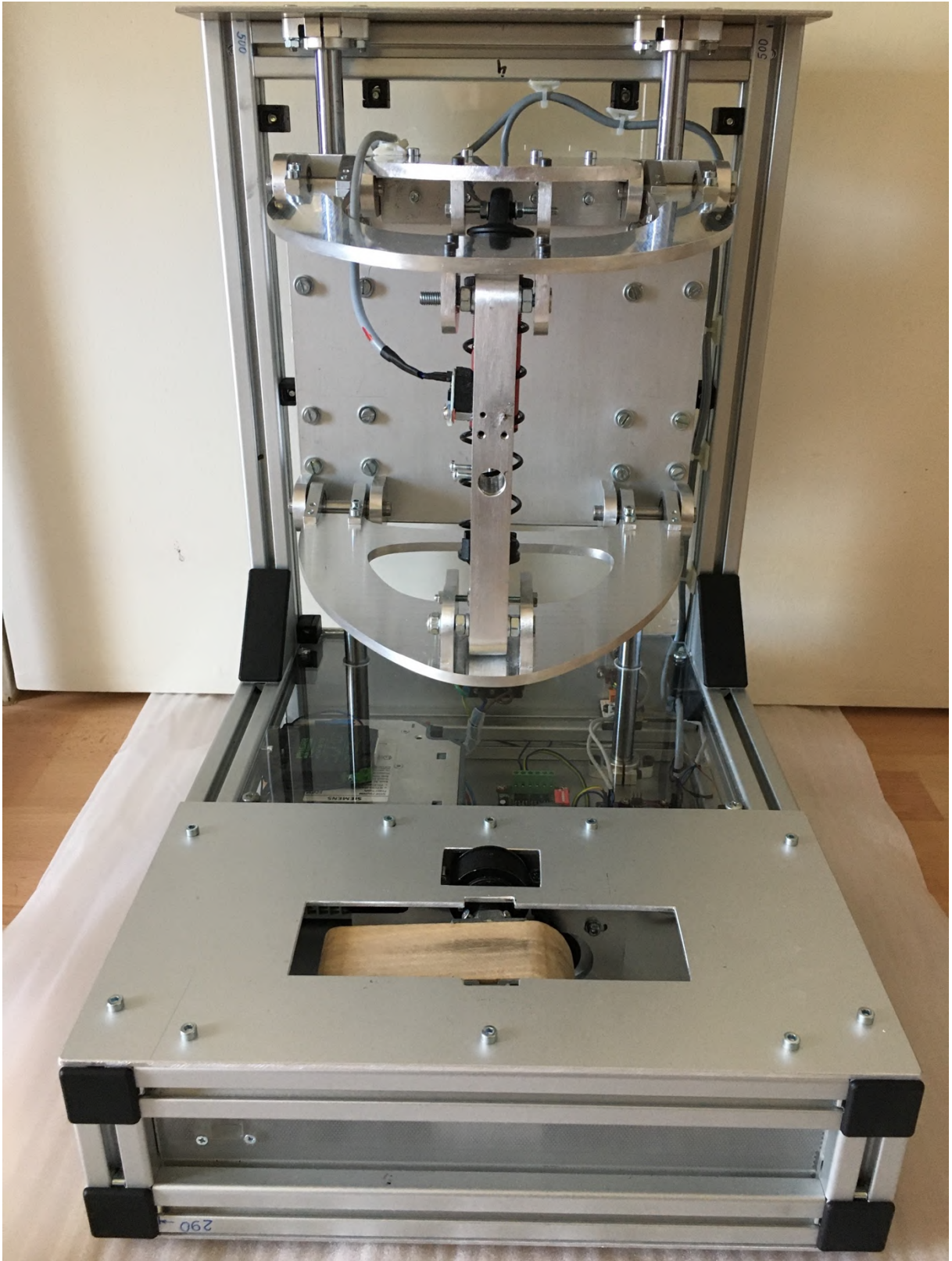
MATERIÁL:
ALU 7075

LIST Č.:
18/18

Príloha D: Fotodokumentácia



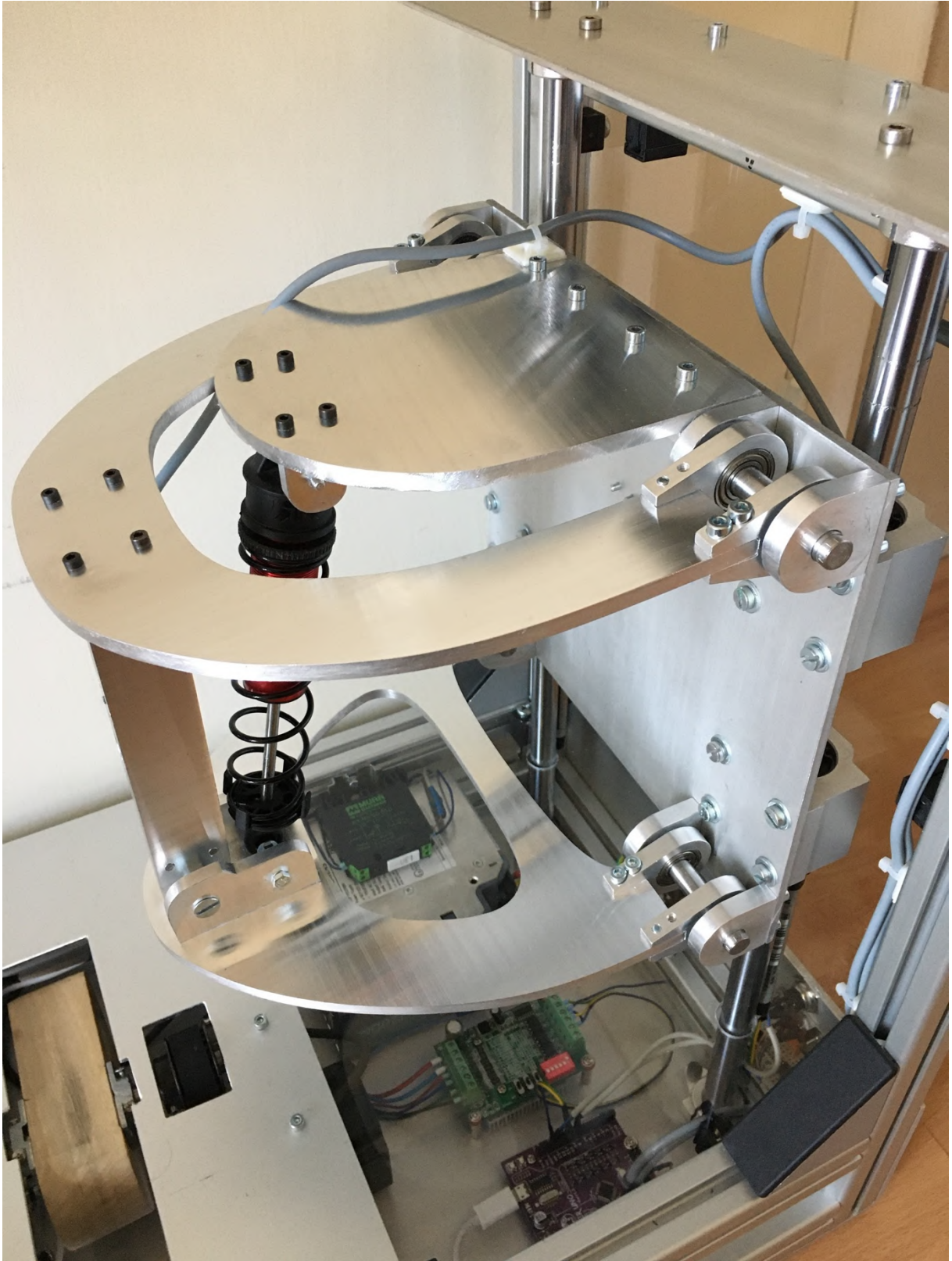
Obr. 1 Fyzikálny model – bočný pohľad



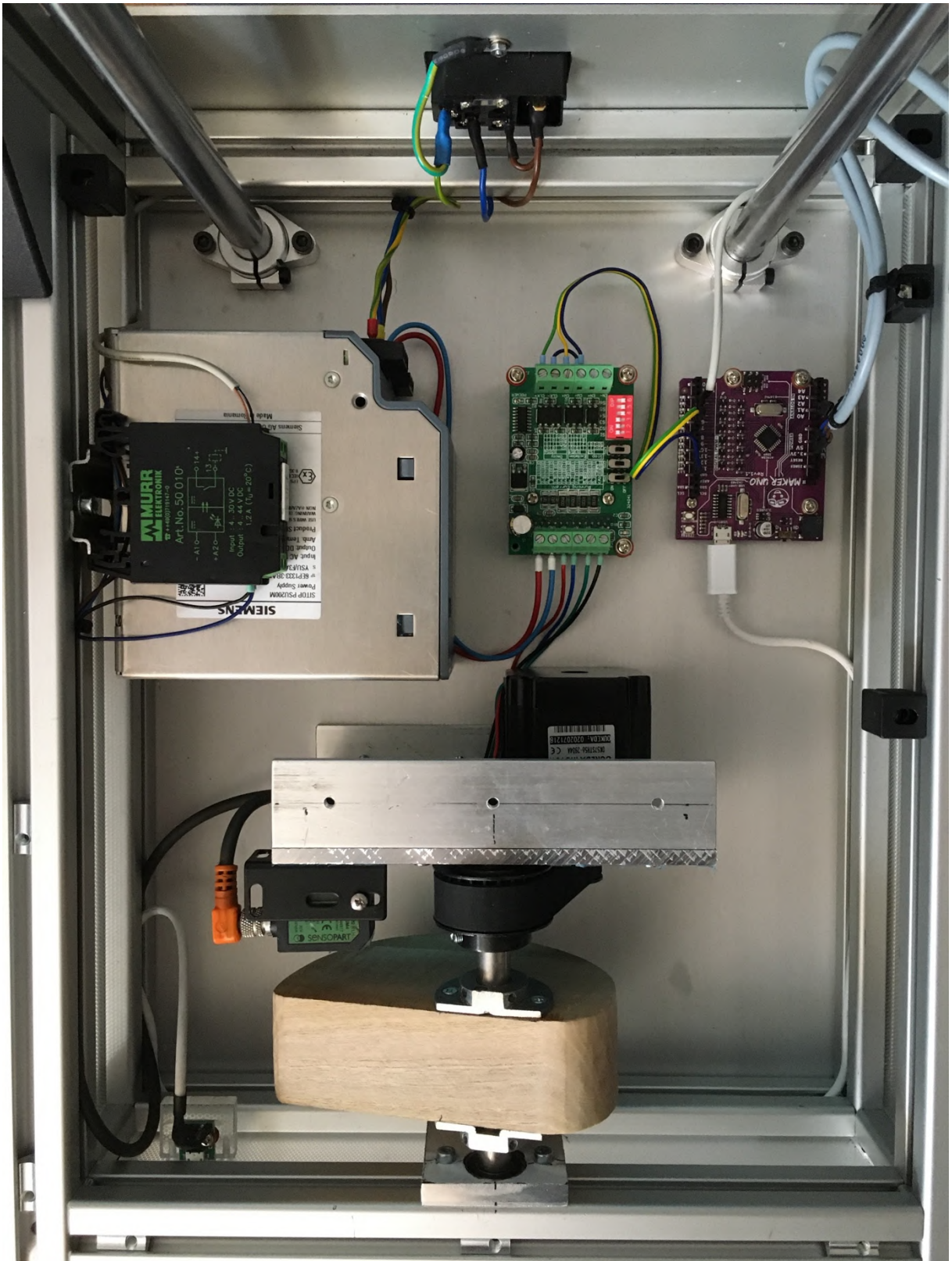
Obr. 2 Fyzikálny model – čelný pohľad bez kolesa



Obr. 3 Lichobežníková náprava – bočný pohľad bez kola



Obr. 4 Lichobežníková náprava – perspektívny pohľad bez kolesa



Obr. 5 Elektronika a budenie

Príloha E:Štruktúra repozitára

Adresa repozitára: <https://github.com/stanislavrevay/DP.git>

- Dokumentacia/ - Dokumentácia k diplomovej práci
 - DP_Revay.docx - Diplomová práca vo formáte .docx
 - DP_Revay.pdf - Diplomová práca vo formáte .pdf
 - DP_TechnickaDokumentacia - Technické výkresy
- Foto_a_Video/... - Fotografie a videá modelu
- 3DModel/ - Počítačový model nápravy
 - DP_3DModel_OBJ.obj - 3D model formátu .obj
 - DP_3DModel_STEP.step - 3D model formátu .step
 - DP_3DModel_STL.stl - 3D model formátu .stl
 - DP_Sim.bin - Súbor programu MSC Adams
- CH341SER/... - Inštalátor ovládača CH341SER
- DP_ArduinoProg/ - Priečinok s programom mikropočítača
 - DP_ArduinoProg.ino - Program mikropočítača
- MatlabApp/ - Súbory GUI aplikácie
 - DP_App.mlapp - Zdrojový súbor aplikácie GUI
 - DP_pohyblivaCast.png - Titulný obrázok GUI
 - neodAY.mat - Vektor simulačných dát
 - odAY.mat - Vektor simulačných dát
 - odAY20.mat - Vektor simulačných dát
 - t.mat - Vektor simulačných dát
 - Ikona.png - Ikona GUI
 - Riadenie Modelu.mlappinstal - Inštalátor GUI aplikácie