

## **PROHLÁŠENÍ**

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 16. 8. 2013

.....

## **Poděkování**

Děkuji Ing. Miroslavu Flídrovi, Ph.D. za vstřícnost a poskytnutí odborné pomoci při vedení této práce.

Dále děkuji nadačnímu fondu REGI Base, obchodům CykloNěmčik, Ramala, beznamahy.cz a v neposlední řadě přátelům za pomoc při stavbě asistenčního systému.

# Anotace

Tato bakalářská práce se zabývá návrhem asistenčního systému pro invalidní vozík, který bude možné snadno montovat a demontovat na stávající obyčejný invalidní vozík, a to takovým způsobem, aby byla snížena fyzická námaha při tlačení tohoto vozíku. Nejprve byla navržena a vyrobena konstrukce asistenčního systému, v dalším kroku byl proveden návrh a výroba elektroniky tohoto asistenčního systému (baterie, motor, řídicí jednotka a další elektronika). V poslední části této bakalářské práce bylo navrženo řízení asistenčního systému – odvozením a regulací modelu a následně vytvořením programu, který se bude o řízení starat.

**Klíčová slova:** Arduino, asistenční systém, Circuit Simulator, časovač, elektrické krytí, H-můstek, Hallova sonda, invalidní vozík, model, Lagrangeova rovnice, PID, plošné spoje, pulzně šířková modulace, regulátor, řídicí systém, Simulink, tlakový senzor.

# Abstract

This thesis describes the design of an assistance system for wheelchair that can be easily assembled and disassembled into the existing ordinary wheelchair in such a way as to reduce physical exertion when pushing the wheelchair. First the construction of the assistance system was designed and manufactured. Second the design and manufacturing of electronic assistance system (battery, motor, controller and other electronics) was carried out. Last the design of system model, derivation of control law and the control software was presented.

**Keywords:** Arduino, assistance system, Circuit Simulator, timer, electric protection, H-bridge, Hall sensor, wheelchair, model, Lagrange equations, PID, printed circuit board, pulse width modulation, controller, control system, Simulink, pressure sensor.

# Obsah

1.Úvod .....	8
2.Konstrukce .....	9
2.1.Materiál .....	9
2.2.Kostra konstrukce .....	9
2.3.Uchycení pohonu .....	11
2.4.Uchycení baterie .....	14
2.5.Uchycení k rukojeti .....	15
2.6.Kolo.....	16
2.7.Ozubené kolo .....	17
2.8.Disk s magnety .....	17
3.Elektronika .....	19
3.1.Baterie .....	19
3.2.Motor.....	20
3.3.Pomocný obvod.....	20
3.4.H-Můstek .....	21
3.5.Senzory.....	22
3.5.1.Tlakový senzor .....	22
3.5.2.Halova sonda .....	23
3.6.Volba součástek .....	24
3.6.1.H-Můstek .....	24
3.6.2.Pomocný obvod .....	24
3.7.Návrh a výroba plošného spoje.....	25
3.8.Řídící jednotka .....	27
3.8.1.Úvod .....	27
3.8.2.Řídící jednotka asistenčního systému .....	27
3.8.3.Časovače .....	29
3.8.4.Pulzně šířková modulace .....	29
3.9.Elektrické krytí.....	31
3.9.1.Stupeň krytí – IP .....	31
3.9.2.Krytí elektrických částí asistenčního systému.....	31
3.10.Kompletní propojení .....	32

4.Model.....	34
4.1.Odvozování modelu.....	34
4.2.Model v Simulinku .....	36
4.3.Množina stabilizujících regulátorů .....	36
4.4.Nastavení parametrů regulátoru.....	39
4.5.Kalibrace tlakových senzorů .....	46
4.6.Kalibrace Hallových sond .....	50
5.Program .....	51
5.1.Úvod .....	51
5.2.Knihovny - Arduino.....	52
5.2.1.Knihovna PID .....	52
5.2.2.Knihovna Queue .....	53
5.3.Vstupy.....	53
5.4.Výstupy.....	53
5.5.Časovače.....	54
6.Testy .....	56
6.1.Nasazení asistenčního systému na invalidní vozík .....	56
6.2.Měření rychlosti .....	57
6.3.Jízda (regulace) .....	57
7.Závěr.....	61
8.Literatura a zdroje.....	62

## Seznam výkresů

Výkres 1 – Kostra konstrukce .....	9
Výkres 2 – Svařená kostra konstrukce .....	10
Výkres 3 – Válečky (Píst).....	11
Výkres 4 – Upevnění asistenčního systému na vozík .....	12
Výkres 5 – Úchyty vozíku .....	12
Výkres 6 – Pružina .....	13
Výkres 7 – Závlačka .....	13
Výkres 8 – Ložiskový domeček .....	13
Výkres 9 – Upevňovací kroužky .....	14
Výkres 10 – Úchyt baterie .....	14
Výkres 11 – Zarážky na baterii .....	15
Výkres 12 – Úchyty rukojeti .....	15
Výkres 13 – Pant rychloupínáků .....	16
Výkres 14 – Kolo.....	16
Výkres 15 – Disk na magnety .....	18
Výkres 16 – Tlakové senzory v rukojeti.....	23
Výkres 17 – Umístění Hallových sond a magnetů .....	24
Výkres 18 – Plošný spoj H-můstku .....	26
Výkres 19 – Plošný spoj pomocného obvodu .....	26
Výkres 20 – Hmota na nakloněné rovině .....	34
Výkres 21 – Vzdálenost magnetů podle aktivace Hallových sond .....	50
Výkres 22 – Nasazování asistenčního systému na vozík .....	56

## Seznam schémat

Schéma 1 – Pomocný obvod .....	21
Schéma 2 - Schéma výkonového spínače (H-můstku) .....	21
Schéma 3 – PWM v Simulinku .....	29
Schéma 4 – Kompletní zapojení.....	33
Schéma 5 – Rovnice hmoty na nakloněné rovině v Simulinku.....	36
Schéma 6 – Metoda stabilizujících regulátorů .....	40
Schéma 7 – Systém s regulátorem v Simulinku .....	43

Schéma 8 – H-můstek nasimulovaný v programu Circuit Simulator .....	57
Schéma 9 – Model systému v Simulinku .....	58

## Seznam obrázků

Obrázek 1 – Řídící jednotka .....	28
Obrázek 2 - Okno programu Arduino .....	51

## Seznam tabulek

Tabulka 1 - Měření doby chůze .....	17
Tabulka 2 – Vzdálenosti magnetů na kole podle sepnutí Hallových sond .....	50
Tabulka 3 – Přepočítání vzdálenosti magnetů na disku podle sepnutí Hallových sond .....	50
Tabulka 4 – Nastavení předděliček .....	55

## Seznam grafů

Grafy 1 – Rychlé PWM .....	30
Grafy 2 – Fázově korektní PWM .....	30
Grafy 3 – GMK systému .....	42
Grafy 4 – Odezva systému s regulátorem na jednotkový skok .....	43
Grafy 5 – Akční zásahy systému s regulátorem na jednotkový skok .....	44
Grafy 6 - Odezva systému s regulátorem na jednotkový skok s chybou .....	45
Grafy 7 - Detail odezvy systému s regulátorem na jednotkový skok s chybou .....	45
Grafy 8 – Odezva tlakových senzorů v rukojeti na zatížení .....	46
Grafy 9 – Částečné proložení odezvy tlakových senzorů v rukojeti na zatížení přímkami .....	47
Grafy 10 – Proložení odezvy tlakových senzorů v rukojeti na zatížení úsečkami a přenesení do jedné přímky .....	48
Grafy 11 – Kvanta Arduina v přepočítání napětí na požadovanou hodnotu .....	49
Grafy 12 – Odezva tlakových senzorů v rukojeti na přepočtenou požadovanou hodnotu ..	49
Grafy 13 – Odezva systému na jednotkový skok a poruchu .....	58
Grafy 14 – Porucha přivedená na systém .....	59
Grafy 15 – Akční zásahy regulátoru .....	59

# 1. Úvod

Pro elektrické invalidní vozíky existuje již mnoho typů pohonů. Problémem většiny těchto vozíků je skutečnost, že jsou ovládány pomocí člověka sedícího na vozíku. Pokud však na něm sedí osoba, která nemůže ovládat vozík, bývá jedinou možností koupit obyčejný vozík bez jakéhokoliv pohonu.

Dalším problémem je váha elektrického invalidního vozíku, neboť pohony jsou buď permanentně spojeny s vozíkem, nebo lze jednoduše vyjmout pouze některé části tohoto elektrického invalidního vozíku. S tím spojený je problém možnosti a jednoduchosti montáže a demontáže celého systému invalidního vozíku například při převozu automobilem. Použitelné jsou totiž v tomto případě pouze speciálně upravené dodávky se zdvihovými plošinami či nájezdy. Kvůli jejich vysoké ceně jsou však lidé, kteří jsou upoutáni na elektrický invalidní vozík, omezeni dojezdovou vzdáleností, kterou umožňují baterie.

Problémem elektrických invalidních vozíků je také jejich vysoká pořizovací cena.

Hlavním cílem této bakalářské práce je tak navrhnout asistenční systém pro invalidní vozík, který bude možné snadno montovat a demontovat na stávající obyčejný invalidní vozík, a to takovým způsobem, aby byla snížena fyzická námaha při tlačení tohoto vozíku.

Ke splnění stanovení hlavního cíle předkládané bakalářské práce povedou tři dílčí cíle:

1. navrhnout a vyrobit základní konstrukci asistenčního systému,
2. navrhnout a vyrobit elektroniku tohoto asistenčního systému – baterii, motor, řídicí jednotku a další elektroniku,
3. navrhnout řízení asistenčního systému – odvozením a regulací modelu a následně vytvořením programu, který se bude o řízení starat.



## 2. Konstrukce

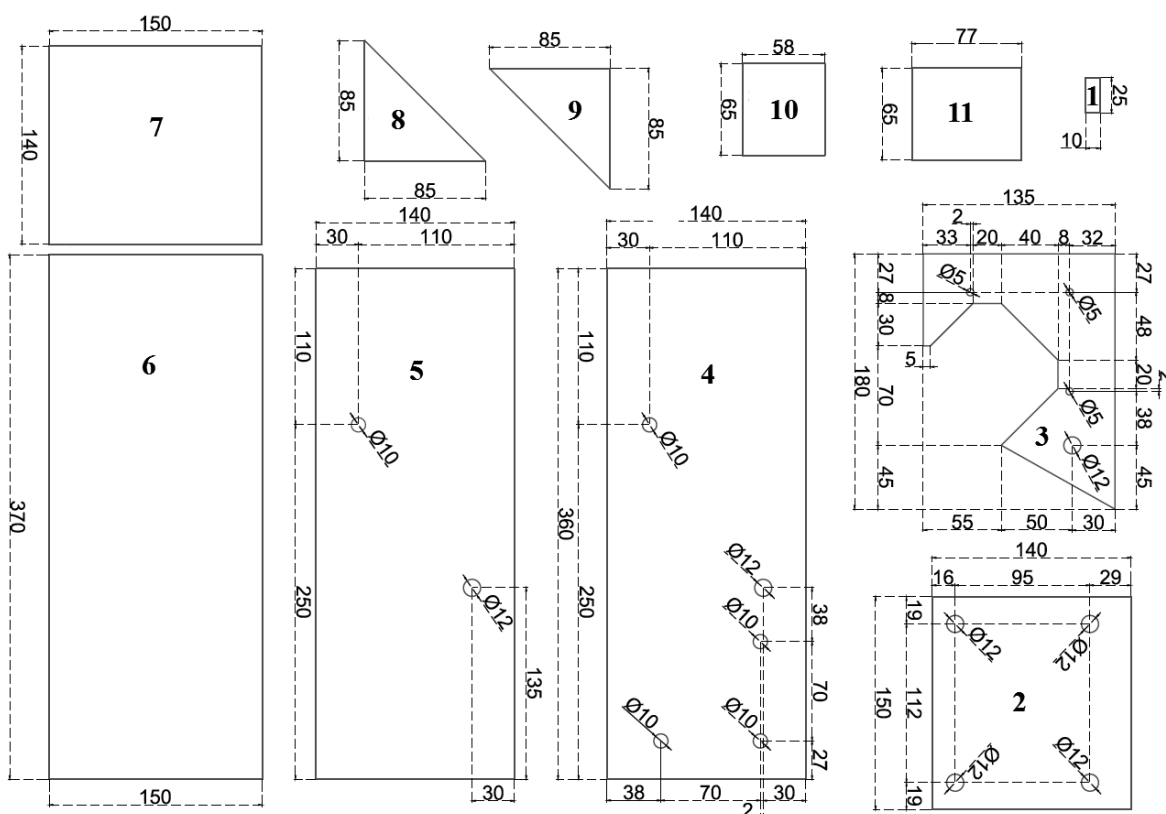
Tato kapitola se zabývá návrhem a výrobou jednotlivých částí konstrukce asistenčního systému tak, aby bylo možno jej snadno připevnit zezadu pod invalidní vozík.

### 2.1. Materiál

Největší prioritou při stavbě a návrhu konstrukce asistenčního systému byla celková hmotnost asistenčního systému. Proto bylo rozhodnuto, že většina dílů se nechá vyrobit z hliníkových plechů či trubek. Pouze nosné části či tyče budou vyrobeny ze železa, aby se používáním materiál nezkroutil či neohnul.

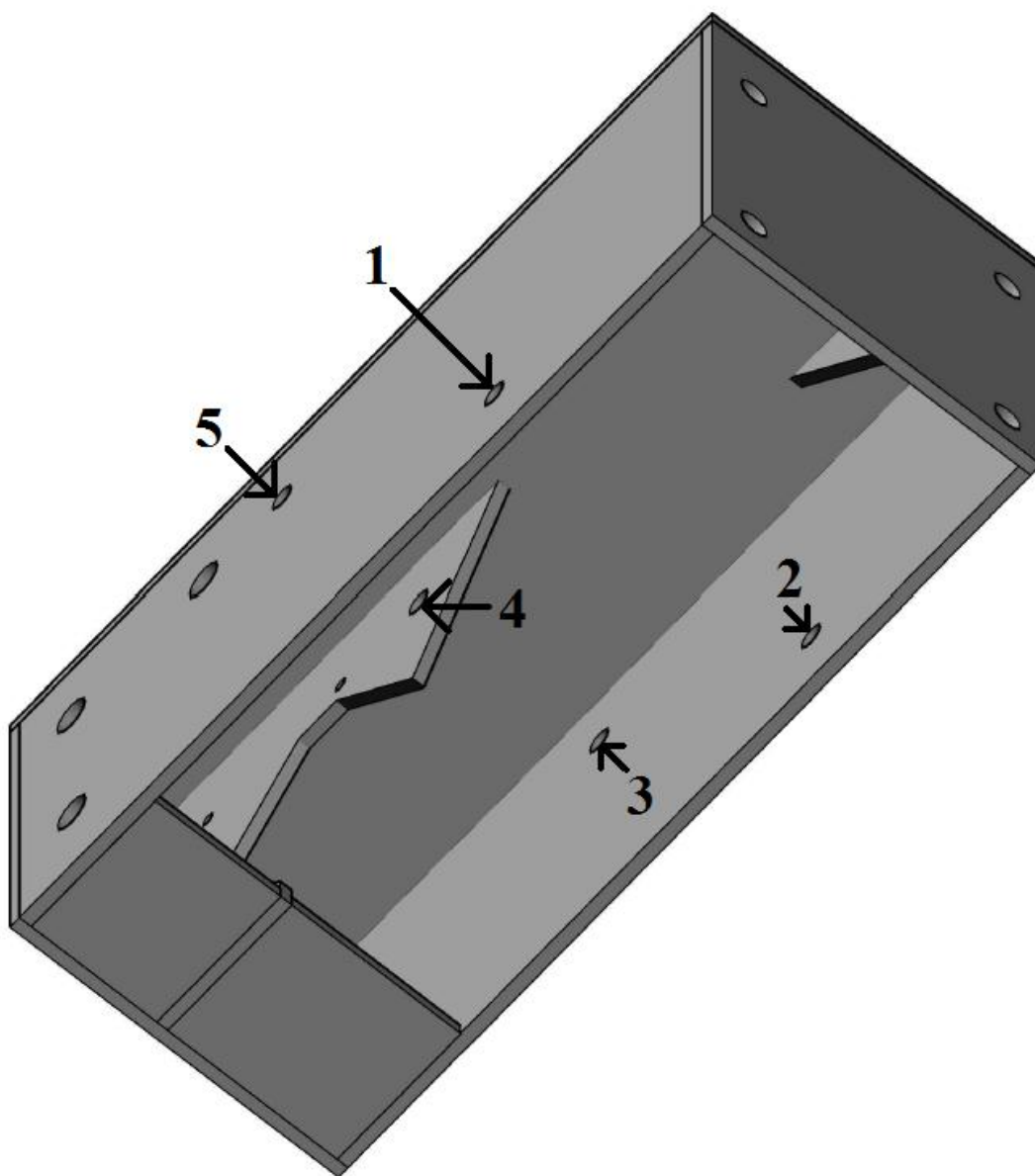
### 2.2. Kostra konstrukce

Bylo zapotřebí vytvořit jakousi nosnou konstrukci pro pohon a řídicí elektroniku tvořící asistenční systém, která by se zasouvala pod invalidní vozík. Tato konstrukce je tvořena díly, které byly navrženy pomocí programu AutoCAD a jsou souhrnně vyobrazeny na výkresu 1.



Výkres 1 – Kostra konstrukce  
Zdroj: vlastní zpracování

Díl označený číslem 6 tvořil stropu kostry. Díly 5 a 4 tvořily boky, díl 2 byla zad' a díl 7 příd'. Díly 8 a 9 podpíraly díl 2 a strop tak, aby nedocházelo k ohybu těchto dvou dílů. Díl 3 má za úkol podepírat příd' tak, aby se neohýbala ke stropu. Na tento díl se také přišrouboval motor do třech otvorů o průměru 5 mm. Tyto šrouby jsou přišroubovány přes tři otvory o průměru 10mm na bočnici 4. Přes 12mm otvory v díle 3, 4 a 5 byla protažena jedna tyč na uchycení. A v 12mm otvorech v díle 2 byly ložiskové domečky, ve kterých byla další uchycovací tyč, jež bude podrobněji zmíněna v další kapitole. Díl 10 a 11 jsou na udržení dílu 3 od dílů 4 a 5. Díl 1 je přivařen na bočnici 4 v místě, kde se bude otáčet kolo, a bude sloužit k uchycení senzorů. Svařená hliníková kostra, je vidět na výkresu 2.

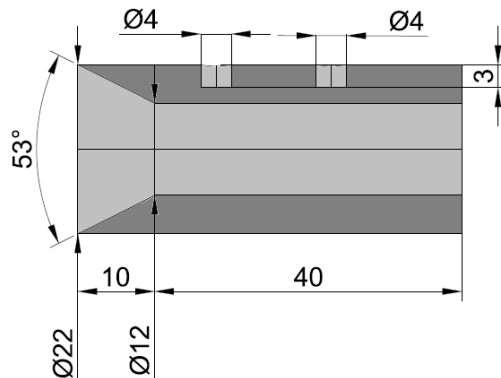


Výkres 2 – Svařená kostra konstrukce  
Zdroj: vlastní zpracování

### 2.3. Uchycení pohonu

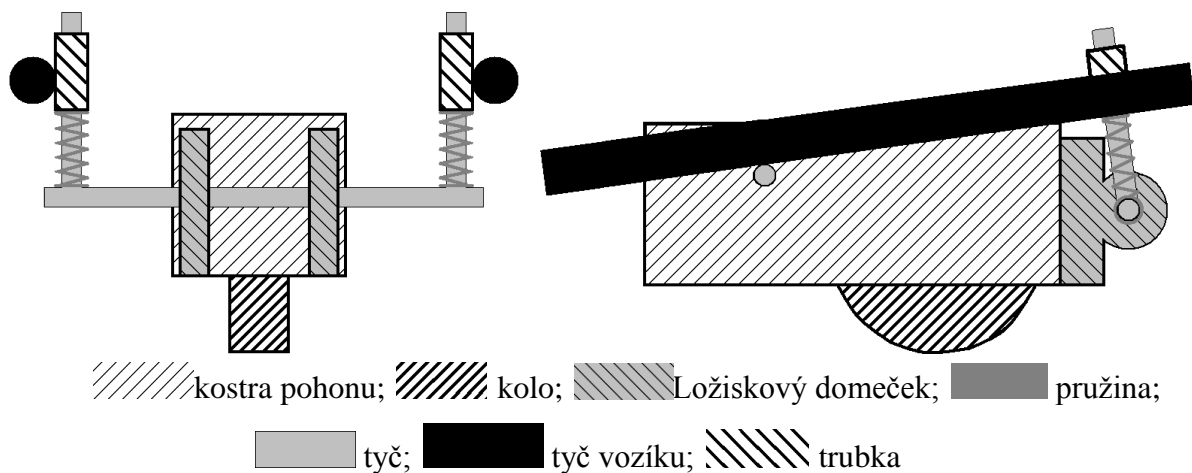
Tato část se zabývá problémem uchycení konstrukce asistenčního systému, která byla navržena v předchozí části.

Jako první opěra byla použita tyč, která se pouze opírá o tyče invalidního vozíku. Tato tyč byla v zasunutá v přední polovině kostry v díle 2, 3, 4 (viz předchozí kapitola) přes díry 3, 4 a 5 (viz předchozí kapitola). V otvorech 1 a 2 (viz předchozí kapitola) bude tyč, na které bude kolo, jež bude poháněno motorem. Toto kolo ale musí být během jízdy přitlačováno k zemi. Jelikož se bude vozík pohybovat po nerovnostech, je nutné použít nucený přítlak. Aby bylo k vozovce přitlačováno kolo, musí být zadní část (díl 3 – viz předchozí kapitola) přitlačována k zemi. Proto bylo zapotřebí vytvořit píst s pružinou, který bude mít tento úkol. Na trubce vozíku, o kterou se opírá první z uchycovacích tyčí, jsou úchyty, jež jsou kolmé k zemi a jež se využívají k upevnění pomocných koleček z důvodu, aby se vozík nepřevrátil. Jelikož ale tyto úchyty nebyly využívány, byla tato pomocná kolečka odstraněna, a byly do nich vyrobeny ocelové válečky s vnějším průměrem takovým, aby se přesně vešel do úchyty, a vnitřním průměrem 12 mm, který se na jedné straně rozšiřuje do ztracena.



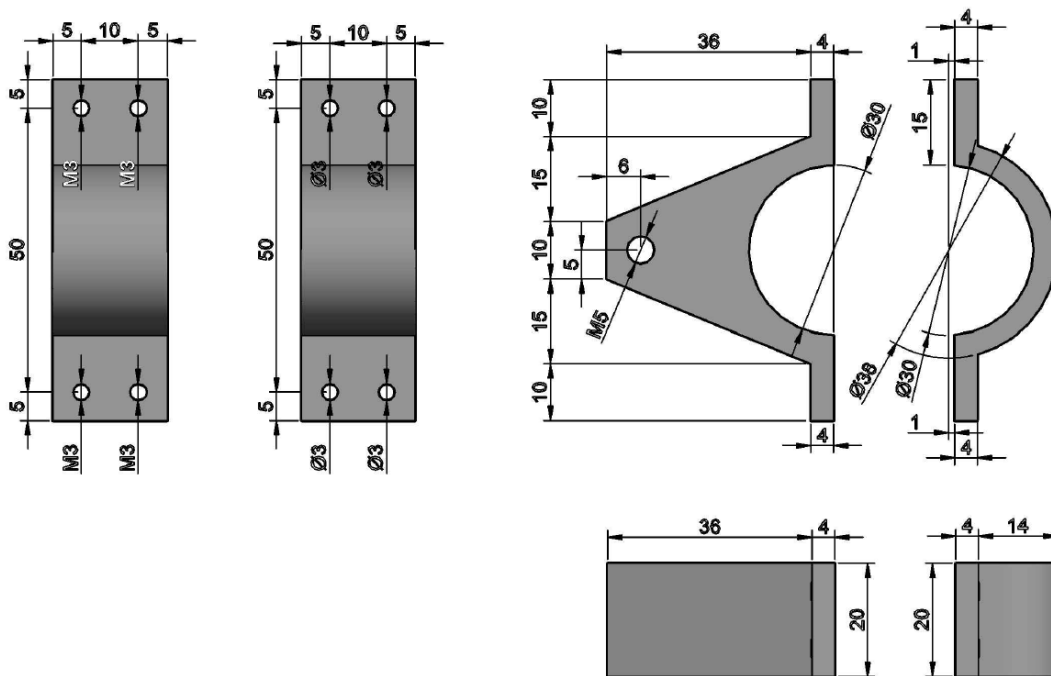
Výkres 3 – Válečky (Píst)  
Zdroj: vlastní zpracování

Do těchto trubek vyobrazených na výkresu 3 musely přijít ještě tyče o průměru 12 mm, které by byly připevněny k další tyči, jež byla přes ložiskové domečky připevněna k zadní části kostry. Tyto tyče musely být tak dlouhé, aby po nasazení na vozík byly z trubek vidět. Spolu s trubkou budou tyto tyče tvořit píst. Na nich také musejí být pružiny, které přitlačují pohon k zemi.



Výkres 4 – Upevnění asistenčního systému na vozík  
Zdroj: vlastní zpracování

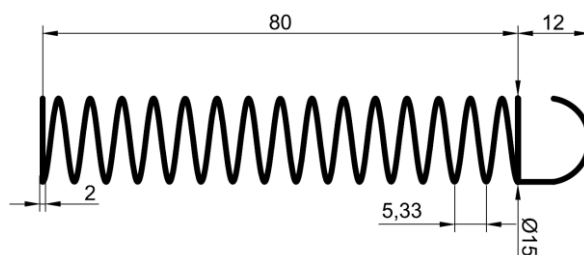
Na výkresu 4 je vidět, že pokud by se vozík naklonil dozadu, kdyby bylo potřeba překonat nějaký obrubník či patník, tak by se přední opěrná trubka neopírala o tyč a navíc by druhá opěrná tyč vypadla z trubek. Aby se nám první opěrná tyč stále opírala o trubku vozíku a aby šlo stále celý mechanismus nasadit ze zadní části vozíku, byly na trubku vozíku vyrobeny úchyty, do kterých se trubka zasunula. Tyto úchyty jsou vidět na výkresu 5.



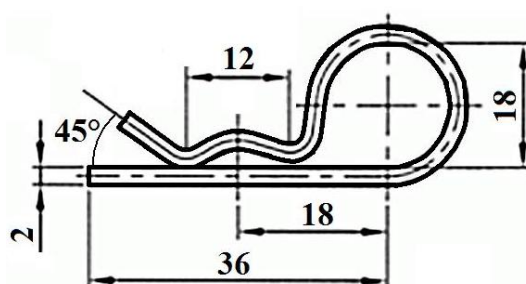
Výkres 5 – Úchyty vozíku  
Zdroj: vlastní zpracování

Do této části úchytu se ještě zašroubovaly tyče do otvorů se závitem M5. Na této tyči poté ležela opěrná tyč, když se vozík naklonil vzad.

Druhým problémem byla možnost, aby druhá opěrná tyč nevypadávala z trubek. Proto byla do trubek vyvrtaná dírka na závlačky, které se nechaly vyrábět spolu s pružinami. Jsou vyobrazeny na výkresech 6 a 7.

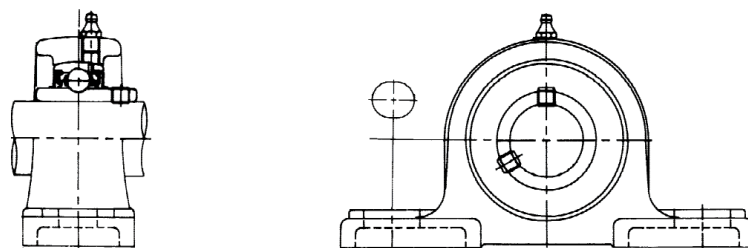


Výkres 6 – Pružina  
Zdroj: vlastní zpracování



Výkres 7 – Závlačka  
Zdroj: vlastní zpracování

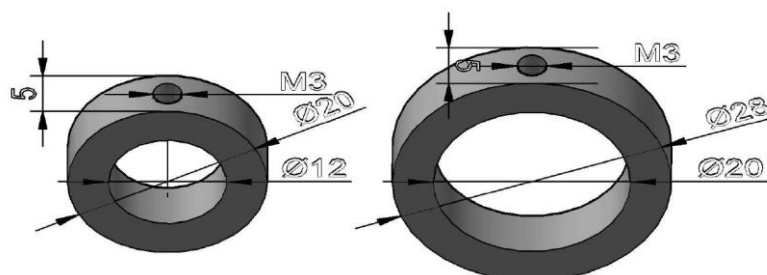
Po těchto úpravách se bude pohon naklánět kolem pevné osy, která má osu otáčení totožnou s první opěrnou tyčí vpředu. A druhá opěrná tyč se bude otáčet v domečkovém ložisku, jež je k nahlédnutí na výkresu 8.



Výkres 8 – Ložiskový domeček  
Zdroj: <http://zbozi.arkov.cz/i/99745-ucp-201-kr-lozisko-upinaci-se-stojatym-litinovym-domeckem-korea.html>

Dále se musela upevnit první opěrná tyč do kostry. Toho se docílilo kroužkem, který se nasadil na tyč a přes příslušnou dírku se závitem pomocí šroubku přitáhnul k tyči. Na tento kroužek se nechal vyrobit ještě jeden kroužek, který větší průměr prvního kroužku, a to z toho důvodu, aby se takto zvětšený kroužek nasadil opět na první opěrnou tyč a fungovali jako zarážky mezi opěrnou tyčí a tyčí invalidního vozíku, na kterém byly

úchyty, do kterých se první opěrná trubka zasunula. Tyto kroužky jsou vyobrazeny na výkresu 9.

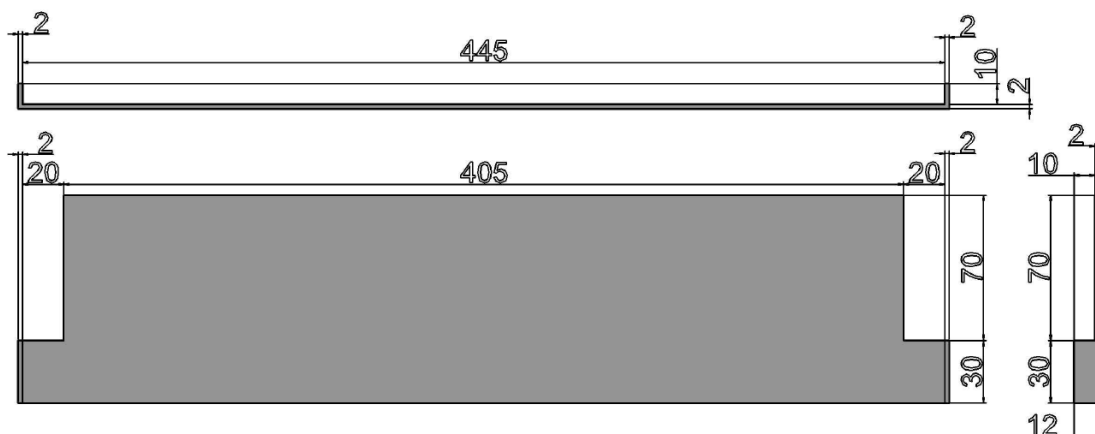


Výkres 9 – Upevňovací kroužky  
Zdroj: vlastní zpracování

## 2.4. Uchycení baterie

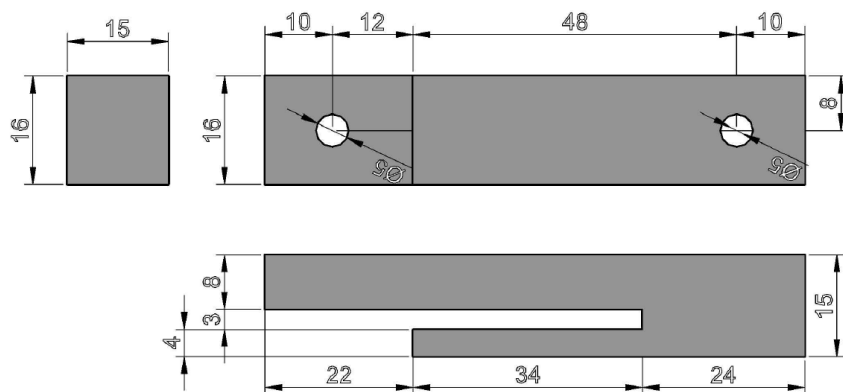
Tato část se zabývá problémem, jakým způsobem a na jaké místo připevnit baterii, která bude zdrojem asistenčního systému.

Baterie měla už od výroby na sobě kolejnici, po které se měla zasunout do doku. Toho bylo využito, byl proto vyroben hliníkový plech velký jako baterie, který se nasadil do těchto kolejnic a na konci měl „pacičky“. Tento plech je vidět na výkresu 10.



Výkres 10 – Úchyt baterie  
Zdroj: vlastní zpracování

Tyto „pacičky“ se zasouvaly do zářezek, které byly přišroubované k tyčím vozíku, jež jsou částí opěradla z druhé strany, než se člověk sedící na vozíku opírá bedry. Tyto zářezky jsou také vyrobené z hliníku a jsou vidět na výkresu 11.

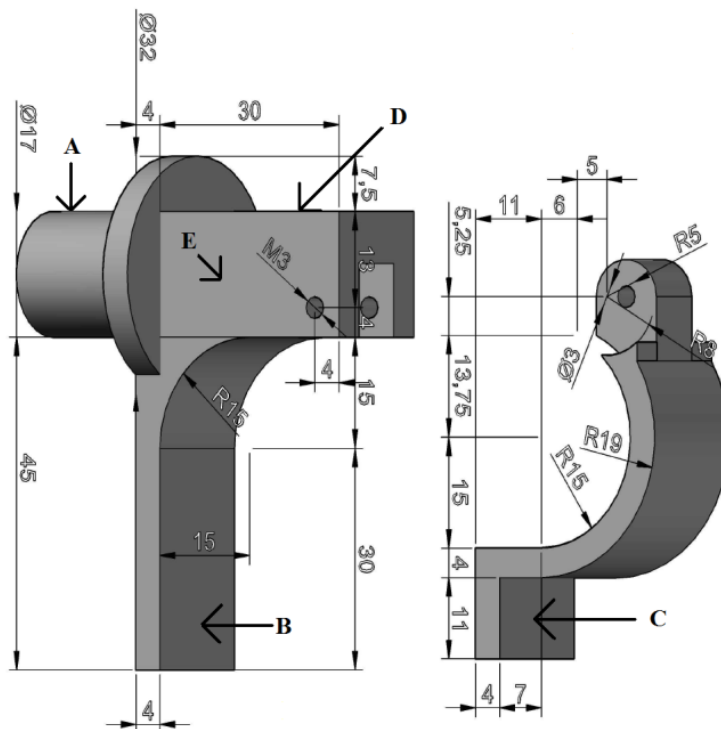


Výkres 11 – Zarážky na baterii  
Zdroj: vlastní zpracování

## 2.5. Uchycení k rukojeti

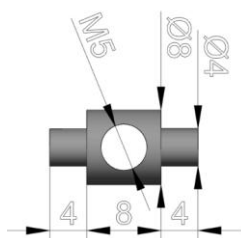
Tato část se zabývá problémem, jakým způsobem a kam připevnit k rukojetím vozíku senzor, který bude měřit sílu tlačení vozíku.

Na rukojeti musí být uchycena hliníková trubka, která je součástí senzoru, na který se tlačí při tlačení vozíku. Tento úchyt musí jít snadno nasadit i sundat. Proto byl navržen tento držák, který je vidět na výkresu 12.



Výkres 12 – Úchyty rukojeti  
Zdroj: vlastní zpracování

Tyto dva úchyty se v místě A nasadily na hliníkovou trubku a přišroubovaly se k ní. V místě C byla vyříznuta díra, do které se vložil díl vyobrazený ve výkresu 13.



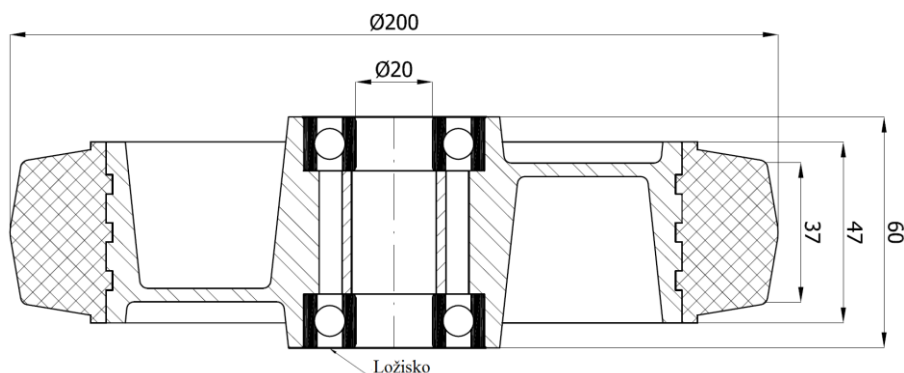
Výkres 13 – Pant rychloupínáků  
Zdroj: vlastní zpracování

Do závitu byl zašroubován rychloupínák, který se využívá v horských kolech k utažení středu kola do rámu kola. V místě B byla vytvořena štěrbina, aby se do ní mohl rychloupínák zasadit a utáhnout dva díly rukojeti k sobě. Na druhé straně byly díly spojeny pomocí šroubu M3. U jedné rukojeti byla tyč, na kterou ukazuje písmeno A (na výkresu 12), do této tyč byl vyvrtán otvor kvůli vodičům, které povedou ke snímačům. Tyto vodiče povedou s vodičem, který povede ke spínači, v místě E. Spínač bude shora vložený do úchyty v místě D, kde bude komůrka pro páčkový spínač. Z úchyty bude tedy vidět pouze páčka spínače.

## 2.6. Kolo

Tato část se zabývá problémem, jaké kolo vybrat pro pohon asistenčního systému.

Kolo, na které se bude přenášet točivý moment motoru řetězem, by mělo být co nejmenší, aby nepřidávalo hmotnost pohonné jednotce. Na druhou stranu však musí být dostatečně velké, aby se na něj dalo přidělat ozubené kolo. Kolo také musí mít pěticipý disk, aby se na něj dalo přidělat ozubené kolo s pěti otvory. Proto se použilo kolo A50GS s obvodem 0,62 m, hliníkovým diskem kvůli jeho hmotnost, kuličkovým ložiskem kvůli nejdelší životnosti a také kvůli šedé pryži, aby toto kolo nedělalo na světlých površích šmouhy. Toto kolo je vyobrazeno na výkresu 14.



Výkres 14 – Kolo  
Zdroj: Technická dokumentace společnosti ZABI CZECH s.r.o.



## 2.7. Ozubené kolo

Tato část se zabývá problémem, jaké rozměry a počet zubů by mělo ozubené kolo, které bude přenášet krouticí moment z motoru.

Šířka zubu byla určena tím, že na hřídeli motoru bylo již přiděleno kolečko s devíti zuby a šířka zubu byla 3 mm. Počet otvorů na přišroubování se vyrábí převážně v počtu pět v této šířce zubů. Pro stanovení počtu zubů na ozubené kolečko bylo hlavní určit, jakou maximální rychlostí vozík má jezdit. Není totiž potřeba velké rychlosti, ale spíše síly celého pohonu. Proto bylo sérií měření určeno, jak rychle se bude s vozíkem pohybovat, a následně byly tyto hodnoty zapsány do tabulky 1.

Měření doby trvání přejezdu vozíkem vzdálenosti 5 metrů maximální přiměřenou rychlostí při tlačení vozíku											
Měření	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	Průměr
Doba trvání [s]	4,9	4,8	4,7	5,0	4,7	4,8	4,9	4,7	4,8	4,9	4,82

Tabulka 1 - Měření doby chůze

Zdroj: vlastní zpracování

Experimentálním měřením zjistilo, že se vozíkem bude jezdit maximální rychlostí kolem 4,82 s na 5 m, což je přibližně 1,04 m/s neboli 62,24 m/min. Muselo se proto přepočítat, kolik je to zubů.

Musí se vzít v potaz, že na motoru je také převodovka a také se muselo zjistit, jak velký obvod má kolo. Z toho lze odvodit rovnici 2.1

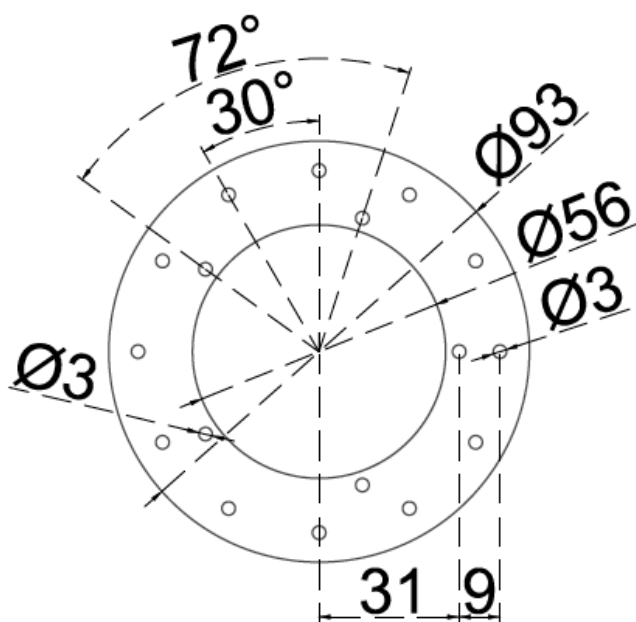
$$X_2 = \frac{n \cdot X_1 \cdot o}{p \cdot v} = \frac{3000 \cdot 9 \cdot 0,62}{7,18 \cdot 62,24} \doteq 37,46, \quad (2.1)$$

kde  $n$  je počet otáček motoru za minutu,  $X_1$  je počet zubů na hřídeli motoru,  $o$  je obvod hnacího kola v metrech,  $p$  je převodní poměr převodovky na motoru a  $v$  je požadovaná rychlost vozíku v metrech za minutu. Vybrán proto byl nejbližší počet zubů, který se vyrábí, a to je 36. Toto ozubené kolo nám umožní jezdit s vozíkem maximální rychlostí 1,079 m/s, což je 3,885 km/h.

## 2.8. Disk s magnety

Tato část se zabývá částí konstrukce, díky níž bude možné měřit rychlost asistenčního systému.

Na kolo bylo přiděláno nejen ozubené kolo, ale také disk s magnety. Díky znalosti vzdálenosti magnetů od sebe bude možno přesně vypočítat rychlost vozíku, pokud budeme znát dobu, jakou trvalo otočení o tuto vzdálenost. Tento disk je vyobrazen na výkresu 15.



Výkres 15 – Disk na magnety

Zdroj: vlastní zpracování

## 3. Elektronika

Tato kapitola se zabývá návrhem, výrobou a výběrem elektroniky asistenčního systému. V jednotlivých částech se tato kapitola zabývá napájením, motorem, plošnými spoji, senzory, řídicí jednotkou a samotným propojením těchto jednotlivých částí.

### 3.1. Baterie

Tato část se zabývá problémem, jakou zvolit baterii pro napájení asistenčního systému.

Při výběru baterie bylo hlavní se rozhodnout, jakou kapacitu má mít tato baterie. Vzhledem k tomu, že tento pohon má být určen na procházky, měla by baterka vydržet 2 až 3 hodiny. Vzhledem k tomu, že motor průměrně pojede přibližně na třetinový výkon, je nejlépe zvolena baterie kolem 10 Ah. Dalším velmi důležitým krokem bylo, jaký druh baterie vybrat. První z nich byla olověná baterie, která je velice levná. Stojí kolem 2500 Kč, avšak při kapacitě kolem 10 Ah a napětí 36 V se její váha dostane na hodnotu převyšující 11 kg. Další nevýhodou je počet cyklů nabíjení, které se u olověné baterie pohybuje kolem 500 cyklů. Nabíjecí proud se nejlépe pohybuje u této baterie kolem pětiny kapacity, což odpovídá v našem konkrétním případě nabíjení okolo sedmi hodin.

Z těchto důvodů byla zvolena dvanáctičládková baterie  $\text{LiFePO}_4$ , která patří mezi nejnovější typ baterií. Tato baterie je netoxická, oproti Li-ion baterii má větší vybíjecí proud a také nehrozí exploze. Baterie  $\text{LiFePO}_4$  je v hliníkovém obalu s plastovým madlem, má kapacitu 10 Ah, vybíjecí proud je až 40 A. Ten je však omezen vyměnitelnou pojistkou na 25 A. Dále tato baterie má v sobě elektroniku, která kontroluje nabíjení, vybíjení a stav kapacity baterie, kterou zobrazuje na čtyřstupňové stupnici. Tato baterie má tři velké výhody oproti olověné baterii. Cyklů nabíjení má až 1000, což je dvojnásobek oproti olověné baterii. Hmotnost je necelých 4,5 kg, což je 2,5krát méně než hmotnost olověné baterie. Poslední výhodou je rychlost nabíjení, která může být až polovinou kapacity baterie. To znamená, že se baterie nabije přibližně za 3 hodiny. Jedinou nevýhodou této baterie je cena, jež dosahuje k hodnotě kolem 8500 Kč, což je 3,4krát více než cena olověné baterie.

## 3.2. Motor

Tato část se zabývá problémem, jaký zvolit motor pro napájení navrhovaného asistenčního systému.

Při výběru motoru se rozhodovalo mezi bezkartáčovým motorem v náboji kola či klasickým kartáčovým motorem, který by dále přes řetěz převáděl svůj kroutící moment na přídatné kolo. Vzhledem k tomu, že jako řídicí jednotka se zvolila platforma Arduino, která pracuje s pulzně šířkovou modulací, byl z výběru vyřazen bezkartáčový motor v náboji kola, neboť všechny tyto motory se řídí frekvenční modulací a nikoliv pulzně šířkovou modulací. Také je tento motor velice těžký, protože váží kolem 7 kg.

Vybrán byl proto klasický kartáčový motor. Dále se muselo zvolit, jaký výkon a na jaké napětí bude tento motor. Vzhledem k tomu, že proud musel být snížen na co nejmenší hodnotu z toho důvodu, aby nemusely být používány vodiče s příliš velkým průměrem od baterie k motoru, zvolilo se napětí 36 V. Výkon motoru byl zvolen na doporučení výrobce elektrokol a pohonu elektrických vozíků kolem 400 W. Posledním parametrem, který rozhodoval o výběru motoru, byly otáčky motoru, které byly zvoleny co nejnižší, neboť není potřeba u invalidního vozíku dosahovat vyšších rychlostí.

Používán je proto kartáčový motor MY1018Z s permanentními magnety, který má napájecí napětí 36 V a výkon 450 W. Maximální špičkový proud motorem je 16 A. Tento motor dosahuje 3000 otáček za minutu, které jsou díky převodovce o převodním poměru 79:11 sníženy na 417,7 otáček za minutu. Tento motor je určen pro pohon kol, takže je hřídel zakončena ozubeným kolečkem s devíti zuby o šířce 3 mm a roztečí 12,7 mm. Kryt převodovky a celého motoru je vyroben z hliníkové slitiny, což pomohlo snížit hmotnost motoru na 2174 g, což je více než třikrát nižší než u motoru v náboji kola.

## 3.3. Pomocný obvod

Tato část se zabývá problémem výroby plošného spoje, který napájí řídicí jednotku a senzory.

Pomocný obvod je obvod, který obsahuje spínaný zdroj, který snižuje napětí z baterie pro napájení řídicího obvodu. Také je zde vyvedeno napájení, zem a výstupy senzorů, které jsou používány jako děliče napětí. Také je tu vyvedena signalizace napětí pomocí luminiscenční diody, která signalizuje napětí na výstupu spínaného zdroje. Dále jsou zde další součástky, které jsou nutné pro chod a stabilitu spínaného zdroje. Schéma tohoto pomocného obvodu je vyobrazeno na schématu 1.

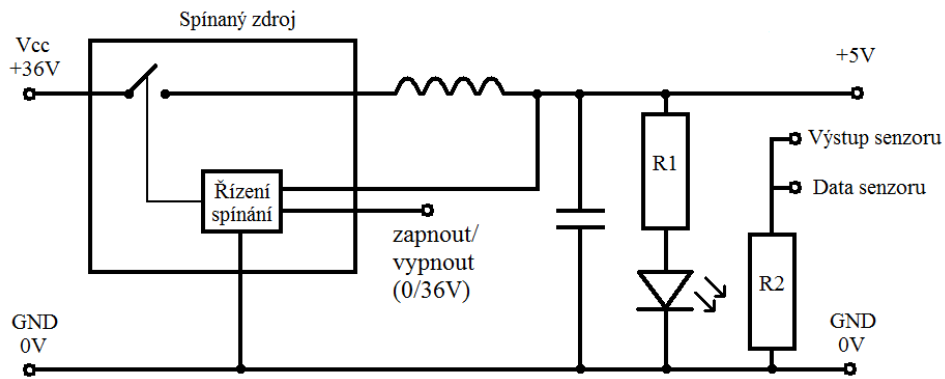


Schéma 1 – Pomocný obvod

Zdroj: vlastní zpracování

### 3.4. H-můstek

Tato část se zabývá problémem výroby plošného spoje, který bude řízen řídicím systémem a bude spínat proud z baterie do motoru.

H-můstek je výkonový spínač, který může spínat motor v obou směrech, ale také může motor brzdit. Při návrhu obvodu bylo potřeba oddělit výkonovou část (motor) od nízko výkonového řídicího obvodu. Proto byla použita pro galvanické oddělení optoelektronická součástka zvaná optron. Tato součástka je dvojhran, který má na vstupu luminiscenční diodu a na výstupu fototranzistor či fotodiodu. Vstup s výstupem je tak propojen pouze světlem. H-můstek je vyobrazen na schématu 2.

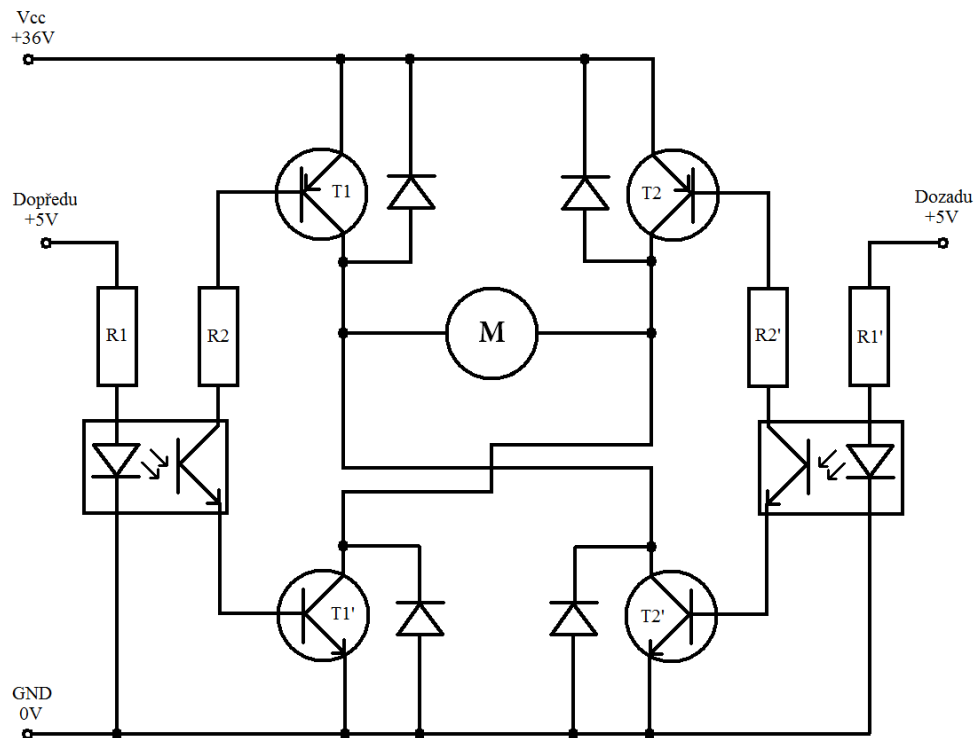


Schéma 2 - Schéma výkonového spínače (H-můstku)

Zdroj: vlastní zpracování

### **3.5. Senzory**

Tato část se zabývá problémem výběru senzorů, které jsou zapotřebí pro tento asistenční systém.

Při výběru senzorů se rozhodlo, že se bude používat tlakový senzor pro snímání síly působící na rukojeť vozíku. Pro měření rychlosti vozíku bude použita Hallova sonda, která bude reagovat na magnety přišroubované na disk s magnety.

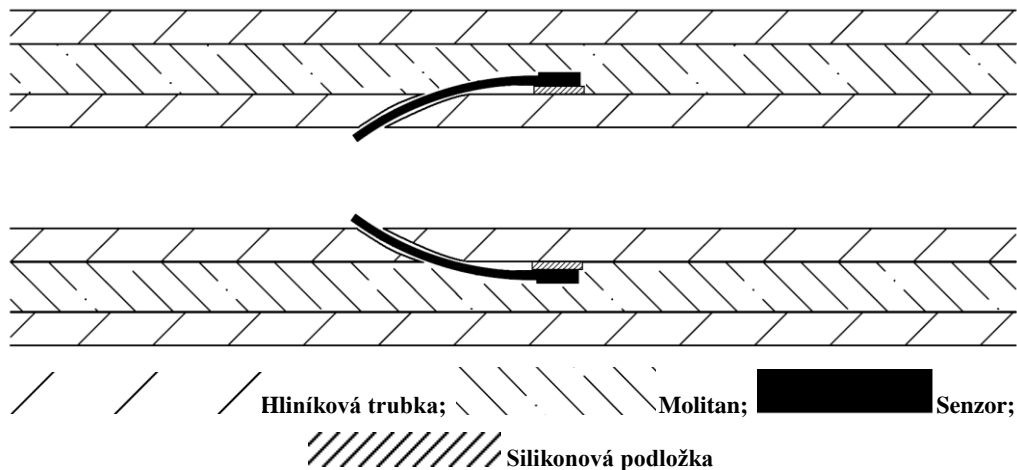
#### **3.6.1. Tlakový senzor**

Tlakový senzor bude zajišťovat, aby člověk, který tlačí vozík, nemusel manuálně měnit výkon pohonu, ale aby stačilo malé síly k tlačení vozíku a zbytek síly byl doplňován pohonem asistenčního systému. To znamená, že člověk tlačící vozík bude tlačít invalidní vozík stejnou silou, ať bude na invalidním vozíku sedět člověk libovolné váhy nebo bude člověk tlačít tento vozík do kopce či z kopce.

Jako tlakový senzor byl vybrán senzor FSR-149NS, který měří tlak od 0,2 do 100 N, jmenovité napětí na senzoru je od 1 V do 5 V a velikost měřicí plošky je 7,62 mm. Senzor se skládá ze dvou vrstev polyesteru, na kterých jsou nanесeny vodivé vrstvy, mezi kterými je vodivý polymer. Tento vodivý polymer je vyroben z organického materiálu, který mění svůj odpor v závislosti na tlaku.

Tento senzor byl umístěn na hliníkové trubce tak, že tlaková ploška byla na vnější straně trubky a zbytek senzoru s přívodními kabely se protáhl skrz vyvrtaný otvor do vnitřku trubky. Na měrnou plošku tlakového senzoru se připevnila silikonová podložka, která způsobí, že při stlačení se stlačí přímo měrná ploška. Trubka se senzorem se následně obalila tenkým a měkkým molitanem. Dále se vzala další hliníková trubka s větším průměrem a nasadila se přes molitan, který byl na vnitřní trubce. Poté byl celý tento mechanismus připevněn pomocí držáku na rukojeti vozíku tak, aby byl senzor přímo na místě, kde se bude vnější trubka přitlačovat k vnitřní kvůli tomu, aby se stlačoval snímač co nejvíce a nejlépe. Tlakový senzor je také samozřejmě i na opačné straně vnitřní hliníkové trubky tak, aby mohl snímat i přitažení trubky k sobě od vozíku.

Umístění tlakových senzorů mezi trubkami je vidět na výkresu 16.



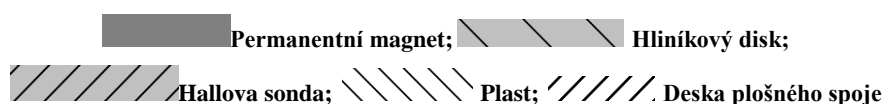
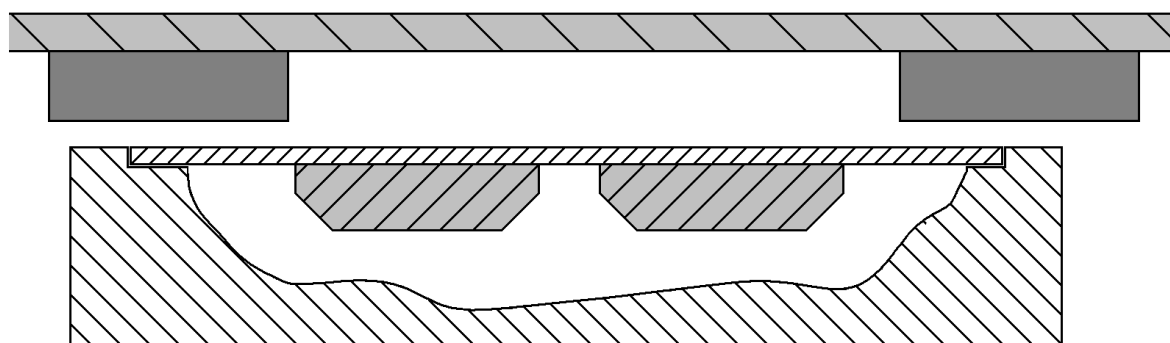
Výkres 16 – Tlakové senzory v rukojeti

Zdroj: vlastní zpracování

### 3.6.2. Hallova sonda

Dalším senzorem je Hallova sonda TLE4905L. Je to Hallova sonda, která v sobě obsahuje Schmittův klopný obvod, který diskretizuje hodnotu napětí na výstupním pinu této Hallovy sondy na hodnotu napětí přivedenou na výstupní pin nebo na nulový potenciál přiveden na uzemňovacím pinu Hallovy sondy do země. Tento senzor má podporované napájecí napětí od 3,8 V do 24 V. Tato sonda využívá takzvaného Hallova efektu, který byl objeven v kovových materiálech. Hallův efekt je způsoben Lorentzovou silou, která způsobuje, že pokud přes kovovou či polovodičovou destičku prochází elektrický proud a kolmo k ní prochází magnetický tok, tak kolmo k elektrickému proudu i k magnetickému toku naměříme napětí úměrně velké procházejícímu proudu a magnetickému toku.

Tento senzor je umístěn u kola pohonné jednotky, na kterém je připevněn hliníkový disk, na němž je připevněno dvanáct permanentních magnetů. Hallovy sondy jsou u kola umístěny hned dvě kousek od sebe, aby bylo možné určit, zda se kolo pohonné jednotky a celý vozík pohybují dopředu či dozadu, a to podle toho, v jakém sledu senzory zaznamenají průchod magnetu kolem sebe. Hallovy sondy jsou připájeny na destičce plošného spoje ze strany mědi. Touto stranou jsou také připevněny do plastu tak, aby byla menší šance, že se mechanicky poškodí. Umístění Hallových sond je vidět na výkresu 17.



Výkres 17 – Umístění Hallových sond a magnetů

Zdroj: vlastní zpracování

### 3.6. Volba součástek

Tato část se zabývá tím, jaké součástky pro jednotlivé plošné spoje zvolit tak, aby fungovali co nejlépe a nejefektivněji.

#### 3.6.1. H-můstek

Při volbě součástek u výkonového spínače (H-můstku) byly zvoleny tranzistory MJ11016 a MJ11015, které jsou v takzvaném Darlingtonovo zapojení. Toto zapojení bylo zvoleno proto, aby byly opravdu (pokud bude potřeba) tyto tranzistory zcela otevřené. Vybrané tranzistory se předdimenzovaly tak, aby se zbytečně neohřívaly při průchodu proudu. Tyto tranzistory měly v sobě také již diodu, která je chrání proti napěťovým rázům při změně otáčení či zastavování motoru. Ve schématu (schéma 2) se odpory R1 a R1' vybraly tak, aby nastavily daný proud a napětí na luminiscenční diodě v optronu. Odpory R2 a R2' se nastavily tak, aby proud přes ně byl co největší pro otevření tranzistorů T1, T1', T2 a T2', ale aby nepřesáhl maximální proud povolený fototranzistorem v optronu.

#### 3.6.2. Pomocný obvod

U pomocného obvodu (schéma 1) se jako spínaný regulátor napětí zvolil integrovaný obvod LM2575T-05, který spíná až +45 V velké vstupní napětí a díky zpětné vazbě toto napětí sníží na +5 V. Také se pomocí připojení napětí na jeden vstup integrovaného obvodu dá aktivovat takzvaný „standby“ mód, který způsobí, že integrovaný obvod nedodává do obvodu žádný proud. Výstupní spínané napětí dále procházelo přes cívku kvůli tomu, aby se integrovaný obvod nedostal do rezonance, protože tím by

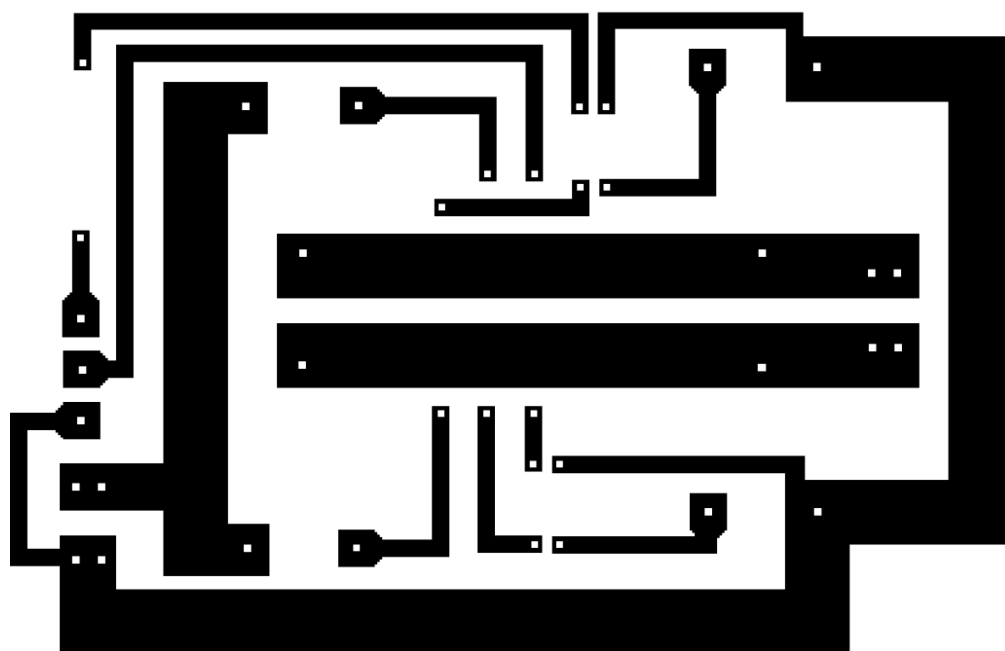


došlo k jeho zničení. Velikost indukčnosti cívky byla zvolena dle manuálu podle toho, jaký bude proudový odběr ze spínaného zdroje. Za touto cívkou se napětí vyhladilo kondenzátorem a vracelo se skrz zpětnou vazbu do integrovaného obvodu kvůli regulaci napětí. Také se toto napětí použilo pro napájení řídicí jednotky a senzory. Na výstup spínaného regulátoru se přidala luminiscenční dioda s ochranným odporem R1, který omezoval proud na diodě, aby bylo vidět, zda integrovaný obvod dodává do obvodu proud. Odpor R2 byl zvolen tak, aby měl na pinu data senzoru co největší rozsah napětí od 0 V do +5 V. To se mění podle toho, jak velký je odpor na odporovém snímači, který má jednu nožičku připojenou na pinu +5 V a druhou na pinu výstupu senzoru.

### **3.7. Návrh a výroba plošného spoje**

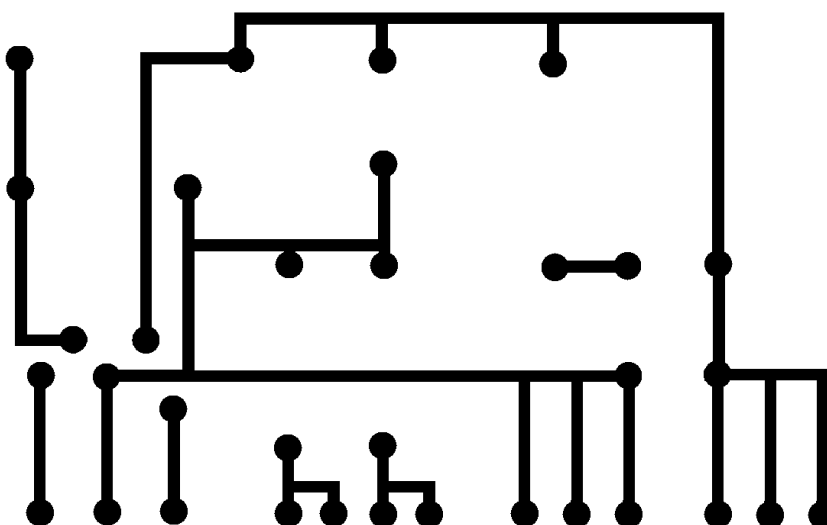
Tato část se zabývá návrhem a výrobou plošných spojů, které jsou zapotřebí v asistenčním systému.

Plošný spoj byl navržen a následně vytvořen za použití grafického programu Malování. Nejdříve se nakoupily jednotlivé součástky, aby se vědělo, jak jsou velké, a dle toho se mohl navrhnout plošný spoj tak, aby byl co nejmenší. Výsledný obrázek namalovaný v Malování byl následně vytištěn na laserové tiskárně na teflonový nepřilnavý papír. To je velice důležité, neboť teflonová tiskárna využívá ke svému černobílému tisku saze, které jsou nanесeny na papír a následně zažehleny do papíru. Jelikož ale nebyl používán k tisku obyčejný papír, ale teflonový, toner se tak nezažehlí, ale pouze se na papír nanese. Výkres plošných spojů H-můstku a pomocného obvodu jsou vidět na výkresu 18 a 19.



Výkres 18 – Plošný spoj H-můstku

Zdroj: vlastní zpracování



Výkres 19 – Plošný spoj pomocného obvodu

Zdroj: vlastní zpracování

Následně se tento teflonový papír natištěnou stranou přiložil k odmaštěné destičce plošného spoje, která je tvořena destičkou cuprexitu, jež má na obou či jako v našem případě na jedné straně nanesenou měděnou vrstvu o šířce 35  $\mu\text{m}$ . Potištěná strana teflonového papíru přiložená na stranu mědi se samolepicí páskou se přilepí k destičce tak, aby se vzhledem k ní nepohybovala. Papír se poté přezehlí obyčejnou žehličkou tak, aby se toner z laserové tiskárny nahřál a následně přenesl na měděnou destičku plošného spoje. Po přezehlení se odlepí teflonový papír a lihovým fixem se dobarví ta místa, na která se

toner nepřenesl dobře nebo vůbec. Destička se poté stranou mědi položila na hladinu leptacího roztoku chloridu železitého a nechá se odleptat měď, která nebyla překrytá tonerem či lihovým fixem. Po odleptání se z destičky omyl toner a lihový fix tak, aby byla vidět zbývající měď. Do míst, kde měla být nožička součástky, se důlčikem udělaly prohlubně, které se odvrtaly tak, aby se do nich přesně vešla nožička dané součástky. Součástky se následně vkládaly z druhé strany, než byla měď, do příslušných děr a ze strany mědi se tyto součástky připájely k vodivým cestám plošného spoje. Vodivé cesty se dále celé pocínovaly tak, aby nedocházelo ke korozi mědi a také proto, aby se zvětšil průřez vodivé dráhy, která se tak při průchodu proudem tolik nezahřívá.

### **3.8. Řídící jednotka**

Tato část se zabývá zvolenou řídicí jednotkou, která bude sloužit k řízení asistenčního systému.

#### **3.8.1. Úvod**

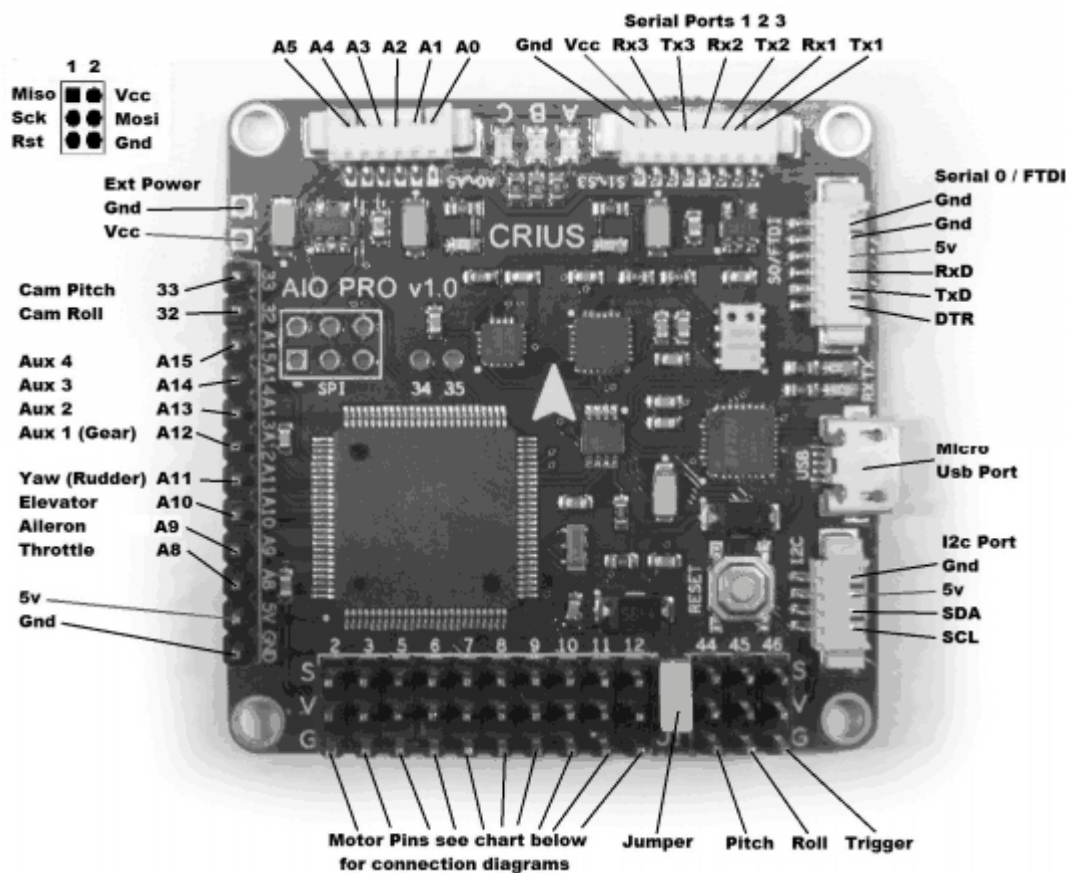
Jako řídicí jednotka byla použita open-source platforma Arduino založená na mikrokontroleru ATmega firmy Atmel. Tato jednotka bude číst hodnoty ze snímačů a dle toho bude zasahovat akčními zásahy do pohybu vozíku za účelem zjednodušit tlačení vozíku po nakloněné či nerovné rovině. Tato platforma má mnoho základních rozšíření, ať už se jedná o senzory, motory či zobrazovací jednotky. Samotný mikrokontroler obsahuje v sobě dle typu časovače, čítače, analogově-digitální převodník, pulsně-šířkový převodník a mnoho dalšího.

V tomto případě jde o osmibitový CMOS mikrokontroler typu ATmega 2560 postavený na RISC architektuře, který je taktován na 16 MHz. Tento mikroprocesor má flash paměť velkou 256 kB a paměť RAM velkou 8 kB. EEPROM je velká 4 kB. Dále obsahuje dva osmibitové čítače/časovače s oddělenou předděličkou a porovnávacím režimem, k tomu má dále čtyři šestnáctibitové čítače/časovače s oddělenou předděličkou, komparátorem a režimem snímání, čtyři osmibitové pulsně-šířkové výstupy, dvanáct šestnáctibitových pulsně-šířkových výstupů, čítač reálného času a vnitřní kalibrovaný oscilátor. Také obsahuje čtrnáct desetibitových analogově digitálních převodníků.

#### **3.8.2. Řídící jednotka asistenčního systému**

Tato část se zabývá konkrétním řídicím obvodem použitým pro řízení asistenčního systému.

Čip řídicího systému mikrokontroler ATmega 2560, byl osazený do řídicí jednotky zvané CRIUS ALL IN ONE PRO v1.1 Multi Rotor Flight Controller. Tato jednotka je vidět na obrázku 1.



Obrázek 1 – Řídicí jednotka

Zdroj: [http://www.rc-timer.com/download/Crius\\_AIOP\\_Manual\\_MWC.pdf](http://www.rc-timer.com/download/Crius_AIOP_Manual_MWC.pdf)

Zde je vyvedeno sedmnáct pulsně-širokových výstupů pro motory, které lze využít jako standardní digitální vstupní piny. Dále je zde vyvedeno 14 desetibitových analogových vstupů a čtyři dvojice pinů TX a RX pro komunikaci s počítači či jinými hardwary. Také jsou zde vyvedeny piny pro komunikaci po sběrnici I<sup>2</sup>C, která je tvořena pomocí dvou pinů, z nichž jeden je SCL, což je vlastně zdroj hodinového signálu, a druhý pin SDA, po kterém se přenášejí data. Na této sběrnici I<sup>2</sup>C je možné připojit až 128 zařízení pomocí pouze těchto dvou vodičů. Na destičce řídicí jednotky jsou také tři signalizující diody na digitálních výstupech a také jsou zde tři další měřicí čipy. Jeden z nich je šestiosý pohybový snímač MPU6050, který je složen z gyroskopu a akcelerometru. Další je tříosý magnetometr HMC5883L a poslední je čip MS5611-01BA01, který má v sobě teploměr a tlakoměr. Destička také obsahuje hardwarové tlačítko pro restartování programu běžícího v mikroprocesoru.

### 3.8.3. Časovače

Tato část se zabývá časovači, které jsou použity při řízení asistenčního systému.

Časovače jsou vlastně části hardwaru vestavěné přímo v čipu Atmel, který využívá platforma Arduino. Slouží k měření času či určitých časových úseků. Časovače v čipu Atmel využívají jako základní takt krystal, který je na destičce plošného spoje Arduino, na kterém je i čip Atmel. V tomto případě mluvíme o čipu Atmel 2560 a krystalu o frekvenci 16 Mhz. Tato frekvence se dá pomocí předděliček, které jdou nastavit na dělicí hodnotu 1, 8, 64, 256 a 1024, dále dělit libovolnými děličkami. Vybraný časovač, kterých má Atmel 2560 šest, lze nastavit na několik módů. Například na mód, při kterém může číst pouze pulzy z krystalu, dále také na mód CTC, který vyčistí hodnotu časovače, pokud dosáhne určité hodnoty. Pak také na mód PWM neboli pulsně šířkové modulace, který funguje ve dvou dalších módech jako rychlá pulsně šířková modulace a fázově korektní pulsně šířková modulace.

### 3.8.4. Pulsně šířková modulace

Pulsně šířková modulace je druh pulzní modulace, při které se snažíme maximální výkon pouštět v určitých pulzech tak, aby bylo docíleno požadované střední hodnoty. Tuto modulaci využívá Arduino jako digitálně-analogový převodník pro zápis hodnot na jednotlivé piny v prezentaci napětí od 0 V do 5 V. Arduino využívá dvou druhů PWM. Jedno z nich je takzvané fast PWM neboli rychlé PWM, které má jako referenční signál pilový signál. Tento referenční signál se porovnává s požadovanou hodnotou, pokud je požadovaná hodnota větší, je na výstupu maximální možný výkon. Pokud je však požadovaná hodnota nižší než referenční signál, je na výstupu nulový výkon. Druhým typem PWM modulace je fázově korektní. Tato modulace používá jako referenční signál trojúhelníkový referenční signál. Obě tyto modulace byly pro názornost nasimulovány softwarovým nástrojem Simulink. Výsledky těchto simulací jsou uvedeny v grafech 1 a 2. Schéma namodelované modulace je vidět na schématu 3.

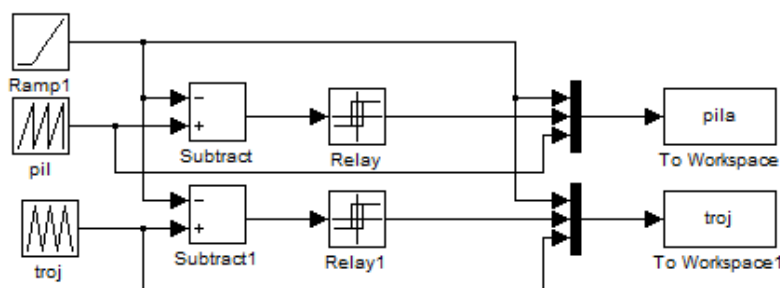
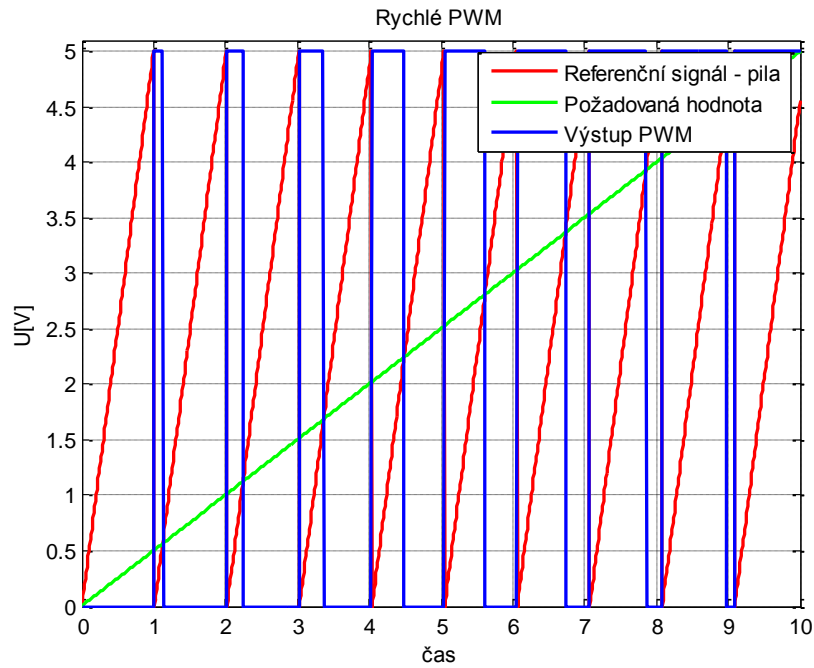
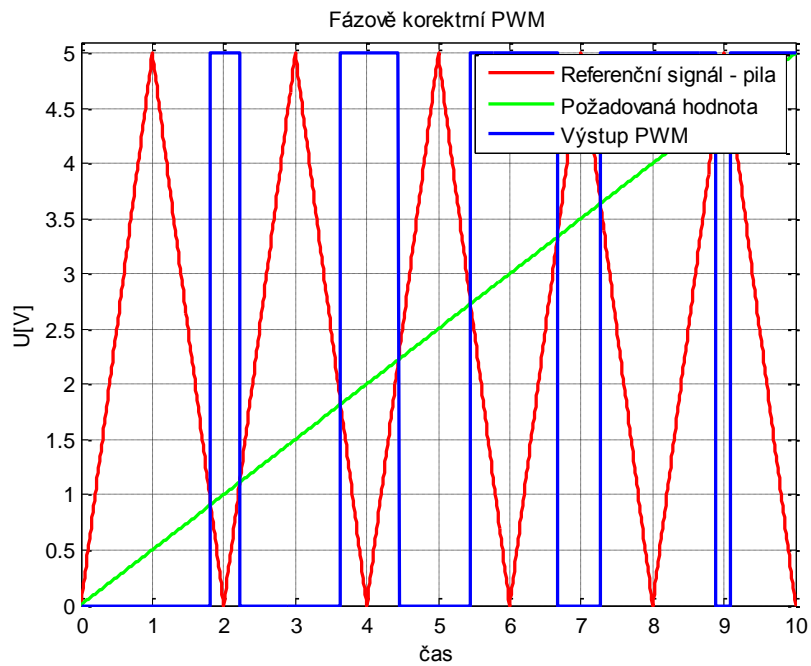


Schéma 3 – PWM v Simulinku

Zdroj: vlastní zpracování



**Grafy 1 – Rychlé PWM**  
Zdroj: vlastní zpracování



**Grafy 2 – Fázově korektní PWM**  
Zdroj: vlastní zpracování

Jak je vidět, fázově korektní PWM má o polovinu nižší frekvenci akčních zásahů oproti rychlému PWM.

### **3.9. Elektrické krytí**

Tato část se zabývá krytím elektrických částí kvůli tomu, aby nedošlo ke zkratu působením vody na elektroniku asistenčního systému.

#### **3.9.1. Stupeň krytí – IP**

Stupeň krytí IP určuje, jakou má dané zařízení či box pro zařízení odolnost proti vodě či proti vniknutí cizího předmětu k elektronice. Toto krytí se skládá ze dvou čísel, přičemž první z nich určuje odolnost zařízení či boxu proti vniknutí cizího předmětu. Toto číslo je v rozmezí od 0 (žádné krytí) až po 6 (zařízení či box je úplně chráněno proti vniknutí prachu k elektronice). Druhé číslo určuje odolnost proti vodě. Toto číslo je v rozmezí 0 (zařízení či box není vůbec kryté proti vodě) až po číslo 8 (box či zařízení může být trvale ponořeno pod vodou).

#### **3.9.2. Krytí elektrických částí asistenčního systému**

Tato část se zabývá konkrétním krytím elektrických součástí asistenčního systému.

Jelikož je tato pohonná jednotka určena k venkovnímu používání, musí být ochráněna především elektronika proti vodě. Proto byla v tomto případě použita krabička GEWISS GW44206 s krytím IP56, což znamená, že chrání elektroniku částečně proti vniknutí prachu i proti vniknutí vody (podle definice normy proti vodě mířící 12,5mm tryskou ve všech úhlech při průtoku 100 litrů za minutu při tlaku 100 kN/m<sup>2</sup> po 3 minuty ze vzdálenosti 3 metry). V této krabici byla uložena řídicí jednotka, H-můstek a také pomocný obvod. Jelikož musely být do krabičky přivedeny také kabely z baterie, musely být použity průchodky do krabičky. Aby nebylo sníženo krytí elektroniky, byla použita průchodka KSS F0603AG-12SR, která při vložení kabelu do průchodky a zašroubování kabel obklopí měkkou gumou tak, aby nemohlo nic vniknout do krabičky. Kabel vedoucí přes průchodka z krabice do baterie musel být připojen k baterii přes odpojitelný konektor, aby bylo možné nasadit a sundat baterii zvlášť. Proto byl použit konektor Superseal, který má krytí IP67, což znamená, že má úplné krytí proti vniknutí prachu a podle definice normy proti ponoření do vody je chráněn na 30 minut do hloubky 1 metr. V rukojeti je také páčkový spínač P-KNX1, kterým se přepínají různé módy programu v řídicí jednotce. Toto tlačítko muselo být také zakryto. To bylo docíleno gumovou čepičkou, která výborně zaizoluje páčkové tlačítko proti vniknutí prachu či vodě. Hallové sondy byly velice pevně přišroubovány do plastu a následně zalepeny tak, aby byly naprosto vodo- i prachotěsné. Signalizační diody byly vyvedeny přes speciální průchodky pro luminiscenční diody, takže

byly také výborně utěsněny. Baterie je už od výrobce výborně zaizolována gumovými těsněními.

### **3.10. Kompletní propojení**

Tato část se zabývá kompletním propojením všech elektrických částí asistenčního systému.

Kompletní propojení zahrnovalo jak propojení všech elektrických součástí k sobě, tak i upevnění na konstrukci pohonného systému. Krabice, která kryla řídicí jednotku, H-můstek a pomocný obvod, byla pomocí čtyř šroubů připevněna na konstrukci pohonného systému. Kabel vedoucí k baterii byl mezi průchodky v krabici propojen pomocí Supersealového konektoru. Kabel vedoucí přes průchodku k tlakovým snímačům a spínači byl kroucený kabel, který se běžně používá k pevným telefonům. Tento kroucený kabel zajišťoval, aby kabel byl napnutý, nikam se nezaplétal a při sundání se jeho délka zmenší, a nebude tak zabírat příliš prostoru. Konektory, kterými byly spínače a snímače připojeny k řídicí jednotce, byl počítačový konektor s roztečí 2,54 mm. Tento konektor byl zvolen z toho důvodu, že řídicí jednotka CRIUS ALL IN ONE PRO v1.1 Multi Rotor Flight Controller měla piny vyvedené pomocí tohoto konektoru. Tímto konektorem bylo také připojeno i napájení řídicí jednotky. Vodič z baterie k H-můstku byl k baterii a H-můstku připojen pomocí fastoru FH2250 6,3 x 0,8 mm. Kabel vedoucí z motoru byl k H-můstku připojen pomocí svorkovnice PTR AK100/2DS-5.0. Touto svorkovnicí byly také připevněny vodiče, které otevírají optron a tím i následně tranzistory, které spouští motor. Vodiče byly také v krabici připevněny pomocí samolepící kabelové příchytky KSS 0515-4J-S. Kompletní propojení je vidět na schématu 4.



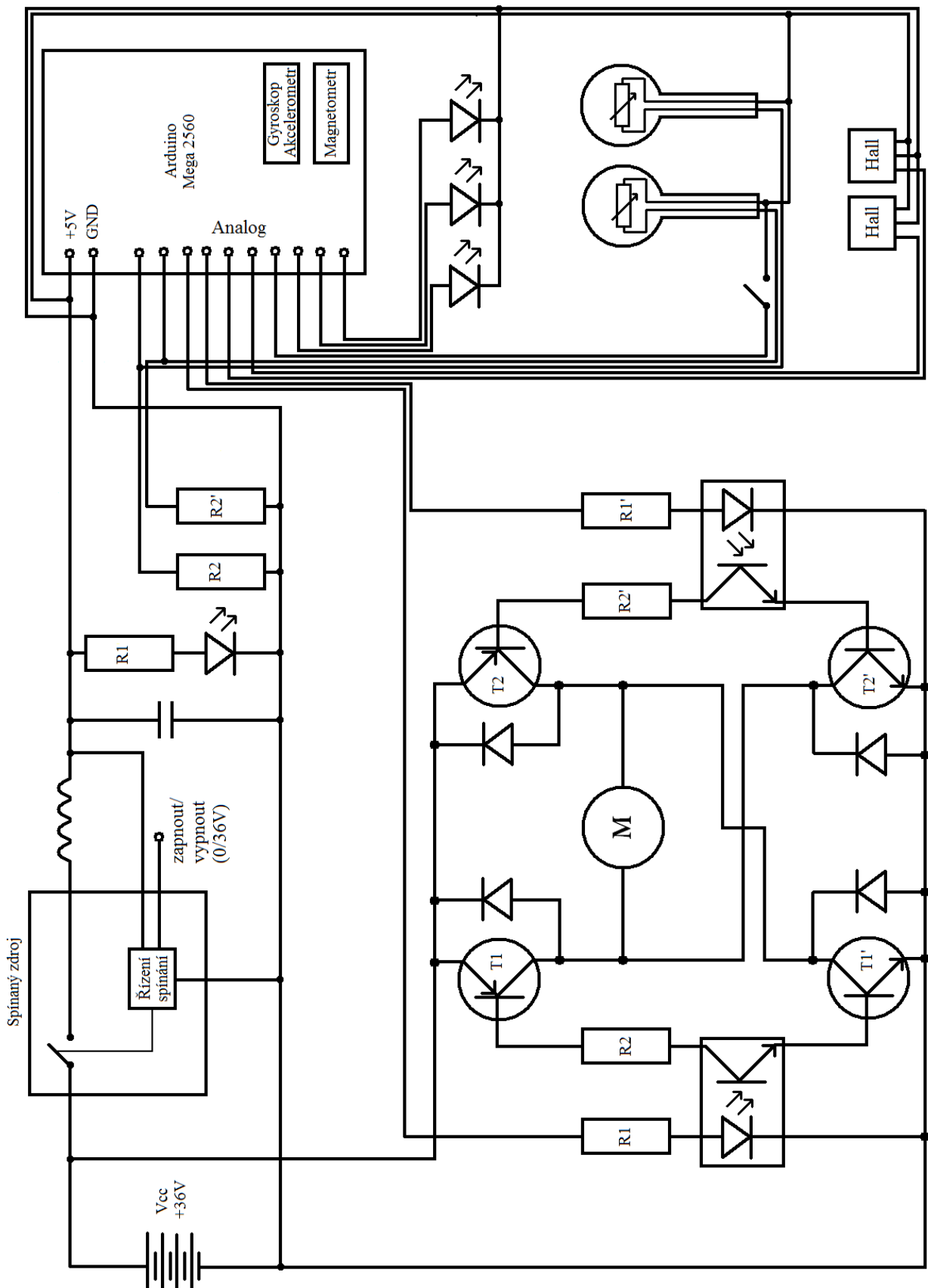


Schéma 4 – Kompletní zapojení

Zdroj: vlastní zpracování

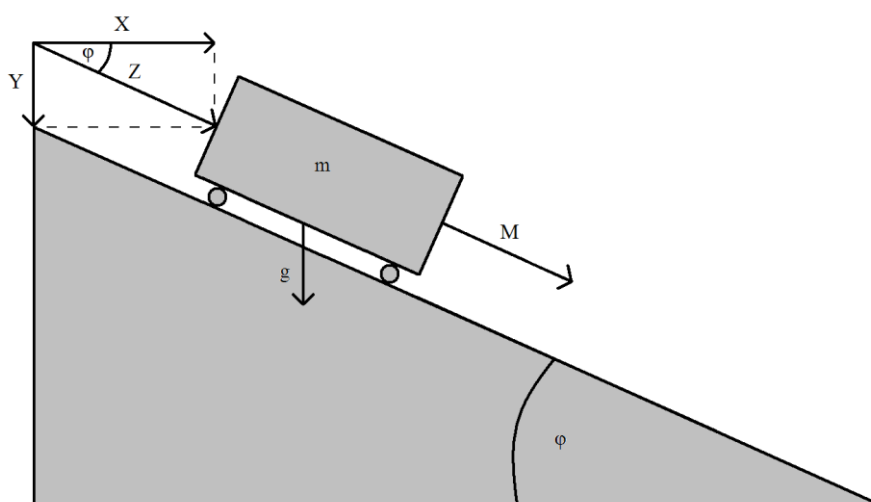
## 4. Model

Tato kapitola se zabývá odvozováním modelu popisující danou problematiku. V dalších částech poté nalezením a nastavením stabilního regulátoru.

### 4.1. Odvozování modelu

Tato část se zabývá odvozováním rovnice, která popisuje pohyb invalidního vozíku s asistenčním systémem. K jejímu zpracování posloužila prezentace Mechanické systémy (Schlegel 2007).

Výkres popisující hmotu na nakloněné rovině je vidět na výkresu 20.



$m$  – hmotnost       $M$  – síla motoru       $\varphi$  – úhel nakloněné plochy  
 $g$  – gravitační zrychlení       $X$  – vektor posunu v ose  $x$        $Y$  – vektor posunu v ose  $y$   
 $Z$  – vektor posunu rovnoběžný s nakloněnou rovinou

Výkres 20 – Hmotu na nakloněné rovině

Zdroj: vlastní zpracování

Pomocí posunu rovnoběžného s nakloněnou rovinou a úhlem sklonu nakloněné roviny vyjádříme posuny v osách  $X$  a  $Y$ :

$$Y = -Z * \cos \varphi \qquad X = Z * \sin \varphi \qquad (4.1)$$

Neboť hledáme změny v těchto osách, tyto rovnice derivujeme podle  $\varphi$ :

$$Y' = Z' * \sin \varphi \qquad X' = Z' * \cos \varphi \qquad (4.2)$$

Nyní pomocí Pythagorovy věty lze vyjádřit změnu pouze v rovnoběžném směru s nakloněnou rovinou:

$$\begin{aligned} Z'^2 = X'^2 + Y'^2 &= (Z' * \cos \varphi)^2 + (Z' * \sin \varphi)^2 = Z'^2((\cos \varphi)^2 + (\sin \varphi)^2) \\ &= Z'^2(1) \end{aligned} \quad (4.3)$$

Rovnice popisující posun hmotného bodu po nakloněné rovině byly odvozovány pomocí Lagrangeových rovnic. Za prvé tedy byla určena kinetická energie T:

$$T = \frac{1}{2} * m * (X'^2 + Y'^2) = \frac{1}{2} * m * Z'^2 \quad (4.4)$$

Za druhé byla určena potenciální energie V:

$$V = -g * m * Z * \sin \varphi \quad (4.5)$$

Za třetí byl určen takzvaný Lagrangian L, který je určen jako kinetická energie mínus potenciální energie:

$$L = \frac{1}{2} * m * Z'^2 + g * m * Z * \sin \varphi \quad (4.6)$$

Za čtvrté byly určeny takzvané zobecněné souřadnice  $q_i$ . V tomto konkrétním případě to byla souřadnice Z.

Za páté byl Lagrangian derivován podle všech zobecněných souřadnic:

$$\frac{\partial L}{\partial q_i} = -g * m * \sin \varphi \quad (4.7)$$

Za šesté byl Lagrangian derivován podle všech derivací zobecněných souřadnic:

$$\frac{\partial L}{\partial q_i'} = m * Z' \quad (4.8)$$

Za sedmé byla derivace Lagrangianu z minulého kroku opět derivována, v tomto případě však podle času:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial q_i'} \right) = m * Z'' \quad (4.9)$$

Za osmé byla sestavená Lagrangeova rovnice:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial q_i'} \right) - \frac{\partial L}{\partial q_i} = m * Z'' + g * m * \sin \varphi \quad (4.10)$$

Za deváté se rovnice z předchozího kroku rovná takzvaným zobecněným silám. V našem případě byla síla  $M$ , tato síla je dána silou dodávanou elektromotorem:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i'} \right) - \frac{\partial L}{\partial q_i} = m * Z'' + g * m * \sin \varphi = M \quad (4.11)$$

## 4.2. Model v Simulinku

V této části se model, který byl odvozen v předchozím kroku, namodeloval pomocí Simulinku, který je součástí Matlabu.

Rovnice, které byly odvozeny v předchozí části, byly ještě upraveny tak, aby na jedné straně rovnice bylo zrychlení a na druhé straně zbytek:

$$m * Z'' + g * m * \sin \varphi = M \quad (4.12)$$

$$Z'' = \frac{M}{m} - g * \sin \varphi \quad (4.12)$$

Schéma namodelování rovnic v Simulinku je vidět na schématu 5.

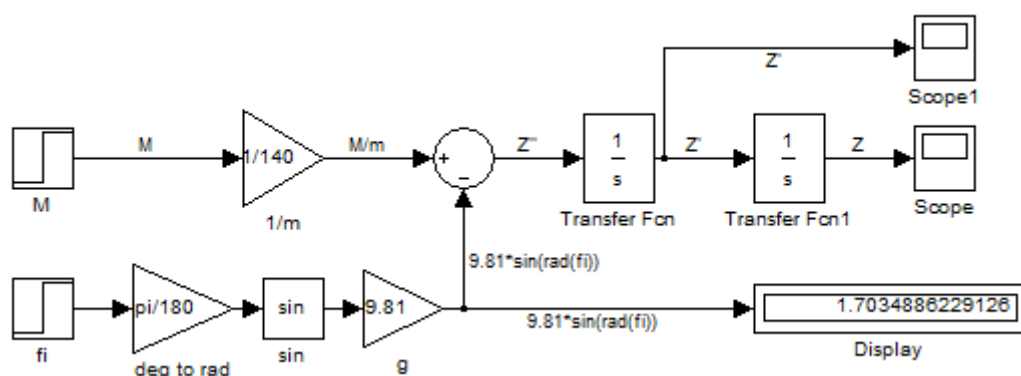


Schéma 5 – Rovnice hmoty na nakloněné rovině v Simulinku

Zdroj: vlastní zpracování

## 4.3. Množina stabilizujících regulátorů

Tato část se zabývá skutečností, že pokud chceme regulovat systém, musíme nejdříve určit množinu stabilizujících regulátorů. Pro následující text byla zdrojem skriptu Lineární systémy (Melichar 2011).

Tyto množiny stabilizujících regulátorů využívají inverzní model řízeného systému. Teorie počítá s tím, že pokud uzavřeme řízený systém s proporcionálním regulátorem a zesílení tohoto regulátoru se bude limitně blížit k plus nekonečnu, dá se regulační obvod popsat tak, že regulátor se změní v přímovazební regulátor k řízenému systému, který má v sobě inverzní přenos řízeného systému.

Obě tyto struktury se dají popsat pomocí vstupně výstupního přenosu:

$$F_{y,w}(p) = R(p) * F_s(p) = \frac{K * F_s(p)}{1 + K * F_s(p)} \quad (4.13)$$

$$R(p) = \frac{K}{1 + K * F_s(p)}, \quad (4.14)$$

kde  $R$  je přímovazební regulátor,  $F_s(p)$  je řízený systém a  $K$  je zesílení proporcionálního regulátoru. Dá se také jednoduše dokázat, že pro  $K$  blížíící se k nekonečnu se  $R$  bude rovnat inverzi řízeného systému:

$$\lim_{K \rightarrow \infty} R(p) = \lim_{K \rightarrow \infty} \frac{K}{1 + K * F_s(p)} = \frac{1}{F_s(p)} \quad (4.15)$$

Při  $K$  blížíící se k nekonečnu by byl vstupně výstupní přenos posán takto:

$$F_{y,w}(p) = \frac{1}{F_s(p)} * F_s(p) = 1 \quad (4.16)$$

To by znamenalo, že by tento systém měl nejen jednotkové zesílení na všech frekvencích, ale výstupní harmonické poruchy by měli oproti tomu nulové zesílení.

Realizace tohoto tvrzení by byla ale nemožná nejen vůli zesílení  $K$  blížíícímu se k nekonečnu, ale také kvůli tomu, že by musela existovat realizovatelná inverze systému.

Proto se přímovazební regulátor realizuje jako součin inverze systému a takzvaného tvarovacího filtru  $F_Q(p)$ , který z přímovazebního regulátoru udělá realizovatelnou ryzí funkci:

$$R = \frac{F_Q(p)}{F_s(p)} \quad (4.17)$$

Pro tento přímovazební regulátor bude vstupně-výstupní přenos vypadat takto:

$$F_{y,w}(p) = R(p) * F_s(p) = \frac{F_Q(p) * F_s(p)}{F_s(p)} = F_Q(p) \quad (4.18)$$

Z tohoto přenosu je vidět, že výsledný přenos uzavřené smyčky bude roven tvarovacímu filtru. Z předchozích vztahů se dá také určit jednoduchý převod přímovazebního a zpětného regulátoru:

$$R(p) = \frac{F_R(p)}{1 + F_R(p) * F_s(p)} \quad (4.19)$$

$$F_R(p) = \frac{R(p)}{1 - R(p) * F_s(p)} \quad (4.19)$$

Pro náš případ je řízený systém, pokud budeme chtít regulovat rychlost, popsán takto:

$$F_S(p) = \frac{1}{m \cdot p} \quad (4.20)$$

Přímovazební regulátor bude tedy popsán pomocí tvarovacího filtru a inverze systému takto:

$$R = \frac{F_Q(p)}{F_S(p)} = F_Q(p) \cdot p \cdot m \quad (4.21)$$

Neboť musí tento přenos být ryzí, tvarující filtr bude zvolen takto:

$$F_Q(p) = \frac{1}{\alpha \cdot p + 1} \quad (4.22)$$

Přímovazební regulátor poté bude vypadat takto:

$$R = \frac{p \cdot m}{\alpha \cdot p + 1} \quad (4.23)$$

Uzavřený systém bude tedy určen parametrem  $\alpha$  vypadat takto:

$$F_{y,w}(p) = R(p) \cdot F_S(p) = \frac{F_Q(p) \cdot F_S(p)}{F_S(p)} = F_Q(p) = \frac{1}{\alpha \cdot p + 1} \quad (4.24)$$

Množina parametrizujících zpětnovazebních regulátorů bude tedy vypadat takto:

$$F_R(p) = \frac{R(p)}{1 - R(p) \cdot F_S(p)} = \frac{\frac{p \cdot m}{\alpha \cdot p + 1}}{1 - \frac{p \cdot m}{\alpha \cdot p + 1} \cdot \frac{1}{p \cdot m}} = \frac{p \cdot m}{(\alpha \cdot p + 1) - 1} = \frac{m}{\alpha} \quad (4.25)$$

Pro náš případ by tedy postačoval proporcionalní regulátor se zesílením  $K = m \cdot \alpha^{-1}$ .

Po přepočtu přímovazebního regulátoru na zpětnovazební se přenos uzavřené smyčky bude rovnat:

$$F_{y,w}(p) = \frac{F_R(p) \cdot F_S(p)}{1 + F_R(p) \cdot F_S(p)} = \frac{\frac{m}{\alpha} \cdot \frac{1}{p \cdot m}}{1 + \frac{m}{\alpha} \cdot \frac{1}{p \cdot m}} = \frac{1}{\alpha \cdot p + 1} \quad (4.26)$$

V našem případě bude působit konstantní porucha na akční zásahy regulátoru. Bylo zapotřebí, aby se tato porucha limitně blížila k nule. Proto jsme si vyjádřili přenos poruchy, která působí na akční zásahy, na výstup, který vypadá takto:

$$F_{y,d}(p) = \frac{F_S(p)}{1 + F_R(p) \cdot F_S(p)} = \frac{\frac{1}{p \cdot m}}{1 + \frac{m}{\alpha} \cdot \frac{1}{p \cdot m}} = \frac{\alpha}{\alpha \cdot p + 1} \quad (4.27)$$

To znamená, že pro  $t$  blíží se k nekonečnu by měla porucha hodnotu:

$$\lim_{t \rightarrow \infty} d(t) = \lim_{p \rightarrow 0} F_{y,d}(p) = \lim_{p \rightarrow 0} \frac{\frac{\alpha}{m}}{\alpha * p + 1} = \frac{\alpha}{m} \quad (4.28)$$

V tomto případě byla porucha v nekonečnu násobkem  $\alpha * m^{-1}$  neboli  $K^{-1}$ . Bylo ale požadováno, aby se porucha limitně blížila k nule. To znamená, že byl k proporcionálnímu regulátoru přidán ještě astatismus pomocí integrátoru, který by měl tento problém vyřešit. Zpětnovazební regulátor se tedy změnil na:

$$F_R(p) = K_0 \left( 1 + \frac{1}{T_I * p} \right) = K + \frac{K_I}{p} = \frac{K * p + K_I}{p} \quad (4.29)$$

Podívali jsme se tedy na to, jestli se pro  $t$  blíží se k nekonečnu bude porucha blížit k nule.

$$\lim_{t \rightarrow \infty} d(t) = \lim_{p \rightarrow 0} F_{y,d}(p) = \lim_{p \rightarrow 0} \frac{F_S(p)}{1 + F_R(p) * F_S(p)} = \lim_{p \rightarrow 0} \frac{\frac{1}{p * m}}{1 + \frac{K * p + K_I}{p} * \frac{1}{p * m}} = 0 \quad (4.30)$$

Přenos uzavřené smyčky tedy bude vypadat takto:

$$F_U(p) = \frac{F_R(p) * F_S(p)}{1 + F_R(p) * F_S(p)} = \frac{\frac{K * p + K_I}{p} * \frac{1}{p * m}}{1 + \frac{K * p + K_I}{p} * \frac{1}{p * m}} = \frac{\frac{K * p + K_I}{m * p + m}}{p^2 + \frac{K * p + K_I}{m}} \quad (4.31)$$

Neboť vidíme, že systém má nulu, což znamená, že nemůžeme použít kritérium ITAE. Bylo rozhodnuto, že se použije metoda GMK. Jelikož chceme mít možnost „pohybovat“ s nulami systému, místo PI regulátoru zvolíme PID regulátor. Uzavřený systém bude vypadat takto:

$$F_U(p) = \frac{F_R(p) * F_S(p)}{1 + F_R(p) * F_S(p)} = \frac{\frac{K_D * p^2 + K * p + K_I}{p} * \frac{1}{p * m}}{1 + \frac{K_D * p^2 + K * p + K_I}{p} * \frac{1}{p * m}} = \frac{\frac{K_D * p^2 + K * p + K_I}{m * p^2 + \frac{K * p + K_I}{m}}}{(1 + K_D) p^2 + \frac{K * p + K_I}{m}} \quad (4.32)$$

#### 4.4. Nastavení parametrů regulátoru

Tato část se zabývá tím, jak nastavit parametry regulátoru tak, aby byl stabilní a efektivně reguloval.

Parametr  $T_I$  byl určen experimentálně a následně bylo pomocí metody GMK zjištěno, pro jaké  $K_0$  bude systém stabilní a jak moc blízko bude nestabilitě. Metoda GMK vykresluje polohu pólů otevřeného systému v závislosti na jednom reálném parametru. V tomto případě byl tento parametr jako proporcionální zesílení  $K_0$  v otevřené smyčce. Samozřejmě se dá také pomocí algebry blokových schémat zesílení  $K_0$  přesunout

v uzavřené smyčce do zpětné vazby a před požadovanou hodnotu. Přesuny blokových schémat namodelované pomocí Simulinku jsou vidět na schématu 6.

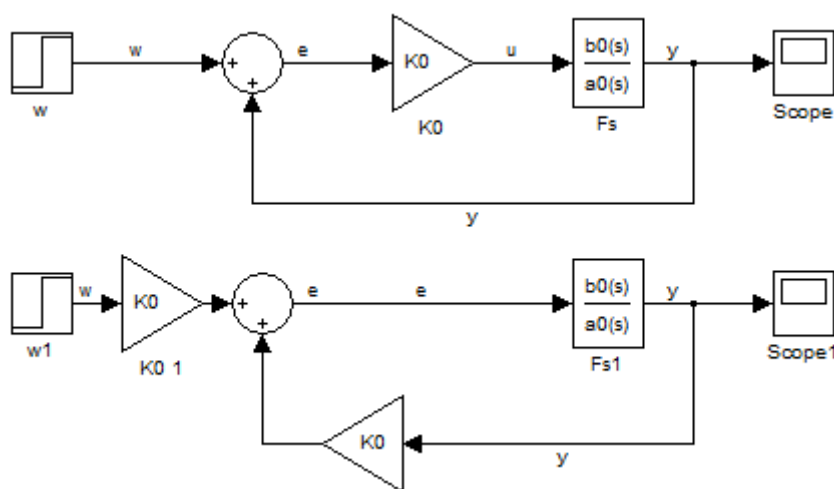


Schéma 6 – Metoda stabilizujících regulátorů

Zdroj: vlastní zpracování

Je tedy vidět, že pokud bude zesílení nula, systém nebude mít žádný vstup a žádnou zpětnou vazbu, systém bude tedy vypadat takto:

$$\frac{F_o(p)}{K_0 \rightarrow 0} = \frac{K_0 \cdot b_0}{a_0} \Rightarrow a_0 = 0 \quad (4.33)$$

GMK budou tedy představovat póly systému. Naopak pokud se zesílení bude blížit nekonečnu, bude systém vypadat takto:

$$\frac{F_o(p)}{K_0 \rightarrow \infty} = \frac{K_0 \cdot b_0}{a_0} \Rightarrow b_0 = 0 \quad (4.34)$$

GMK budou tedy představovat nuly systému. Dá se dokázat, že systém se bude postupně se zvětšujícím se zesílením přesouvat z pólů do nul systému. Existuje také několik základních pravidel a rovnic, pomocí kterých se dá odvodit tvar GMK.

V našem případě se budeme snažit zjistit, jak vypadá GMK, pokud  $T_I = 10$  a  $T_D = 5$ .

Za prvé musíme zjistit nuly a póly systému:

$$\begin{aligned} F_o(p) &= F_R(p) * F_s(p) = K_0 \left( 1 + \frac{1}{T_I * p} + T_D * p \right) * \frac{1}{p * m} = \\ &= K_0 \left( \frac{T_D * T_I * p^2 + T_I * p + 1}{m * T_I * p^2} \right) = K_0 \left( \frac{50 * p^2 + 10 * p + 1}{1400 * p^2} \right) \end{aligned} \quad (4.35)$$



$$50 * p^2 + 10 * p + 1 = 0 \Rightarrow z_{1,2} = -0.1 \pm 0.1i$$

$$10 * p^2 * 140 = 0 \Rightarrow p_{1,2} = 0 \quad (4.36)$$

Za druhé se budeme snažit zjistit, kde asymptoty GMK protínají reálnou osu:

$$q = \frac{\sum_i^n p_i + \sum_i^m z_i}{n-o} = \frac{-0.2}{0} = \nexists, \quad (4.37)$$

kde n určuje počet pólů systému a o počet nul systému. Kdyby existovaly, bylo by zapotřebí ještě zjistit, jaký úhel svírají asymptoty GMK s reálnou osu pomocí následujícího vzorce:

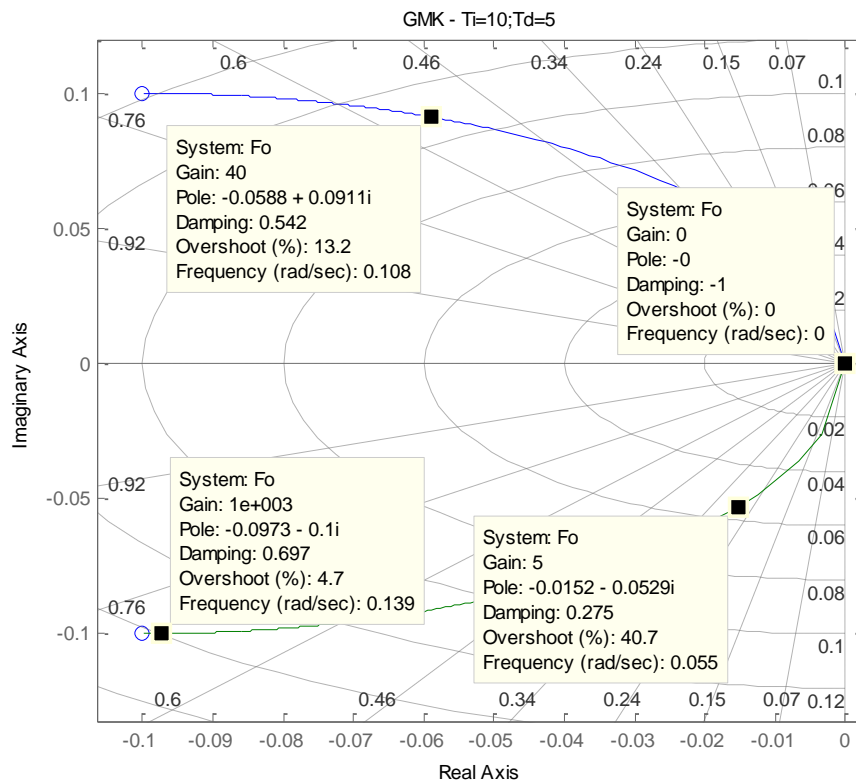
$$\alpha_1 = \frac{\pi - (l-1) * 2 * \pi}{n-o} \quad (4.38)$$

Za třetí je důležité vědět, že GMK probíhá po reálné ose vlevo od lichého počtu nul a pólů.

Za čtvrté je potřeba vědět, kde GMK opouští či přichází na reálnou osu. To se vypočte podle následující rovnice:

$$0 = \sum_i^n \frac{1}{p-p_i} - \sum_j^n \frac{1}{p-z_j} = \frac{2}{p} - \frac{2}{p+0.1} = \frac{2*p+0.2-2*p}{p*(p+0.1)} = \frac{0.2}{p*(p+0.1)} \Rightarrow p = \emptyset \quad (4.39)$$

Toto nám stačí pro náš systém vědět, aby šlo vidět, jak vypadá GMK. Pokud se systém následně vykreslí pomocí Matlabu, je vidět, že systém probíhá podle třetího kroku od pólů systému při zesílení nula do nul systému při zesílení blízcímu se nekonečnu. Graf GMK je vidět na grafu 3.



**Grafy 3 – GMK systému**

**Zdroj: vlastní zpracování**

Z grafu GMK je vidět, že pokud budeme zesílení  $K_0$  volit kladné bez nuly, bude systém vždy stabilní.

V Simulinku se tedy sestavil pomocí bloků uzavřený systém, který má v otevřené zpětné vazbě proporcionalně-integračně-derivační zesilovač s naším systémem. Porucha v podobě nakloněné roviny byla nulová. Také ale musíme vyřešit derivační složku, neboť v tomto případě je ideální a nerealizovatelná. Musíme tedy přenos v rovnici 4.40:

$$F_D(p) = T_D * p \quad (4.40)$$

nahradit ryzím přenosem v rovnici 4.41:

$$F_D(p) = \frac{T_D * p}{\tau * p + 1} \quad (4.41)$$

$\tau$  musí být volit menší než  $T_D$ , aby měl přenos derivační charakter.  $\tau$  se volí většinou v závislosti na  $T_D$ . Obvykle se  $\tau$  volí jako

$$\tau = \frac{T_D}{(3 \div 20)} \quad (4.42)$$

V našem případě bylo zvoleno  $\tau$  jako desetina  $T_D$ .

Schéma namodelovaného systému s regulátorem a zpětnou vazbou je vidět na schématu 7.

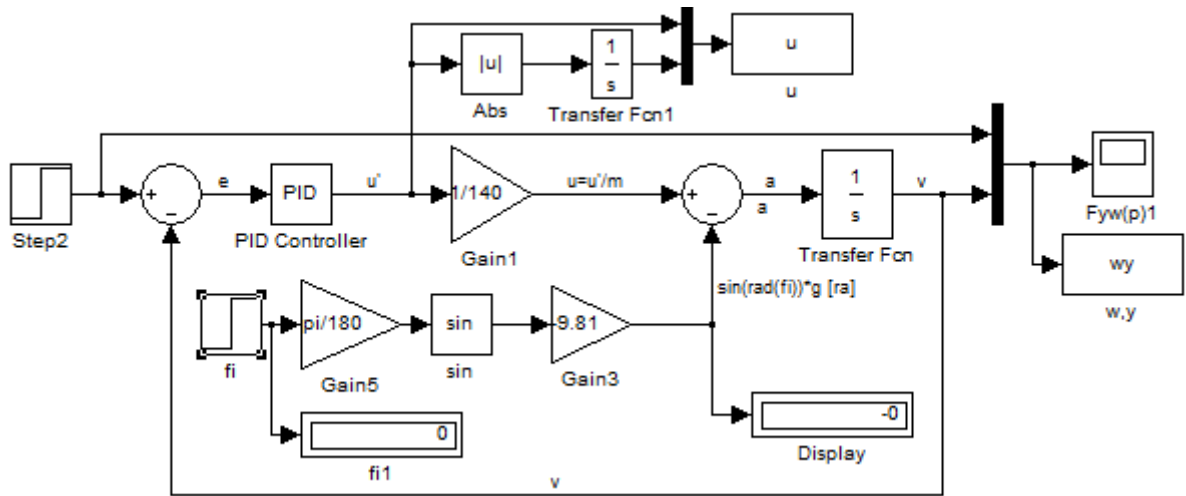
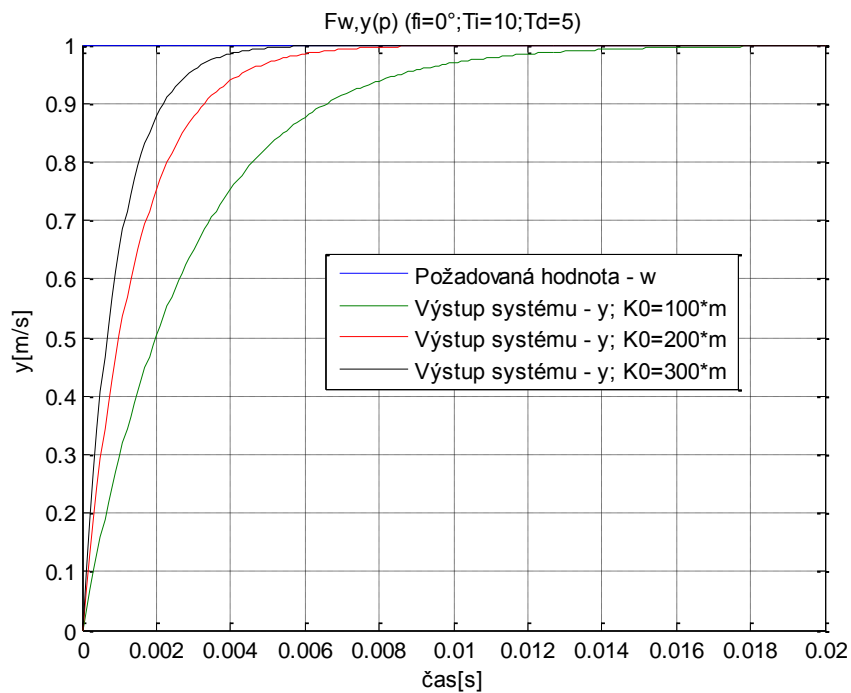


Schéma 7 – Systém s regulátorem v Simulinku

Zdroj: vlastní zpracování

Výstupy ze schématu 7 jsou vidět v grafech 4 a 5.

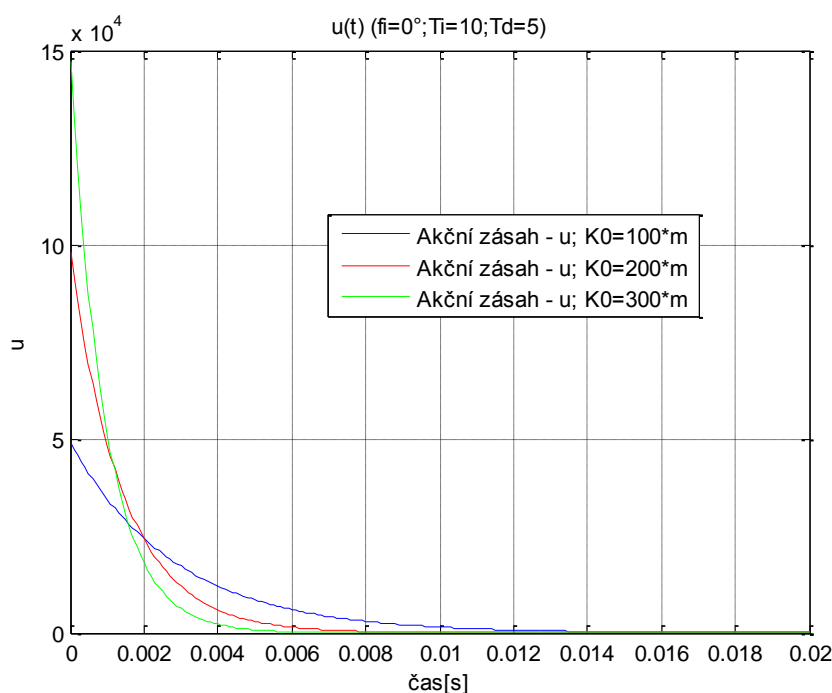


Grafy 4 – Odezva systému s regulátorem na jednotkový skok

Zdroj: vlastní zpracování

Jak je vidět, čím větší budeme volit konstantu  $K_0$ , tím rychleji systém naběhne na požadovanou hodnotu nehledě na to, že větší zesílení bude potlačovat konstantní

poruchy vyvolané jízdou po nakloněné rovině. Podívejme se ale na akční zásahy, které budou v podobě motoru přenášet svoji sílu na pohyb vpřed. A tím pádem čím větší budou, tím větší bude odběr energie z baterie.

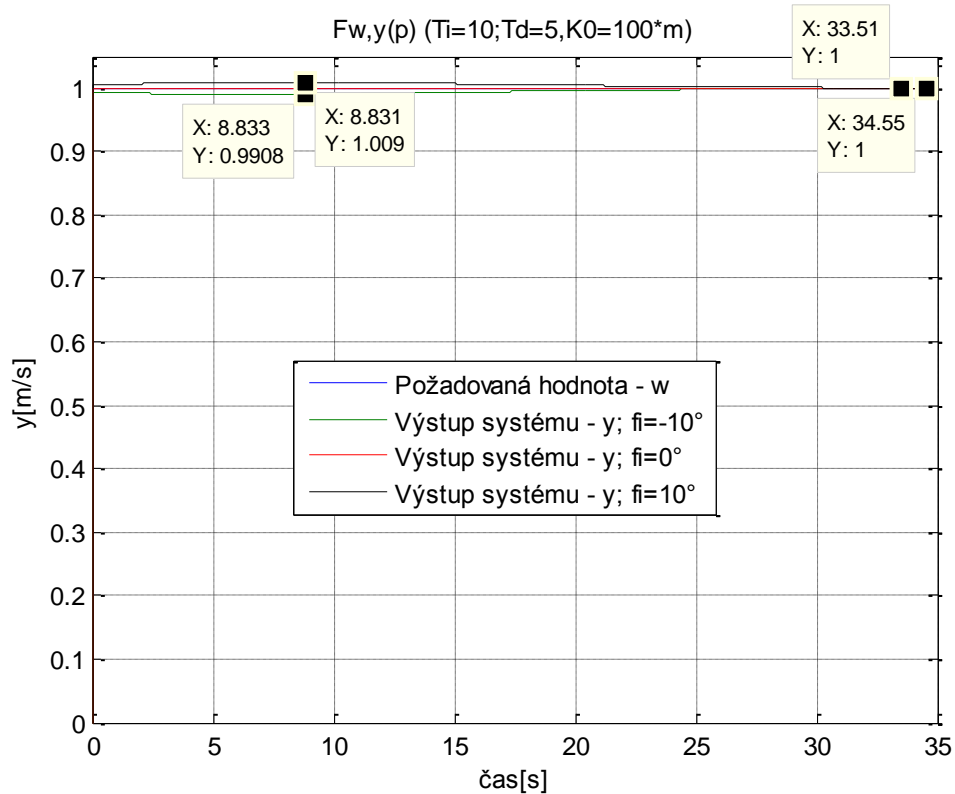


Grafy 5 – Akční zásahy systému s regulátorem na jednotkový skok

Zdroj: vlastní zpracování

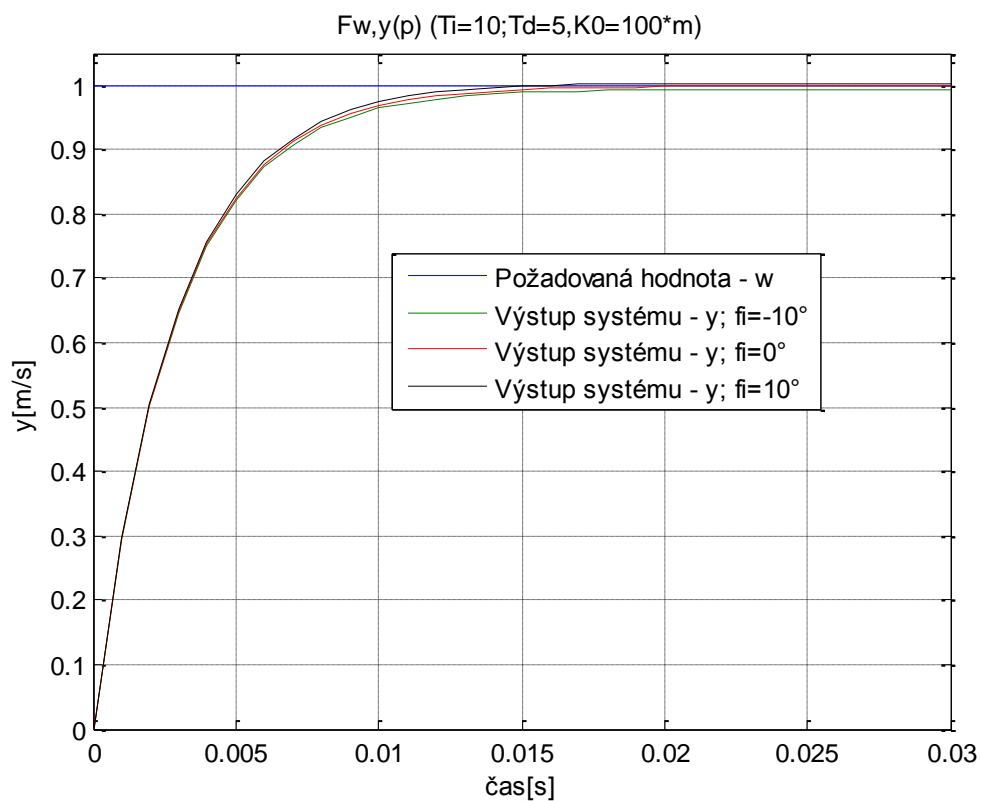
Jak je vidět, bylo už možné před testem říct, jaké řízení je nejvýhodnější pro úsporu baterie. Pro nejmenší zesílení  $K_0=100 \cdot m$  (což znamená, že systému to bude trvat déle, než naběhne na požadovanou hodnotu), bude celková odebraná energie menší než u zesílení větších než  $K_0=100 \cdot m$ . Přesně celková odebraná energie pro zesílení  $K_0=100 \cdot m$  byla rovna 139,7424, pro zesílení  $K_0=200 \cdot m$  celková odebraná energie vzrostla na 139,9344 a pro zesílení  $K_0=300 \cdot m$  opět vzrostla celková odebraná energie na 139,9564. Nesmíme zapomínat, že tento reálný systém bude ovšem mít meze v akčních zásazích, neboť motor má omezený výkon.

Na závěr bylo odzkoušeno, zda bude systém dobře regulovat chybu prezentující velikost úhlu nakloněné roviny pro úhly  $-10^\circ$ ,  $0^\circ$  a  $10^\circ$ . Odezva na tyto chyby je vidět v grafu 6 a 7.



**Grafy 6 - Odezva systému s regulátorem na jednotkový skok s chybou**

Zdroj: vlastní zpracování



**Grafy 7 - Detail odezvy systému s regulátorem na jednotkový skok s chybou**

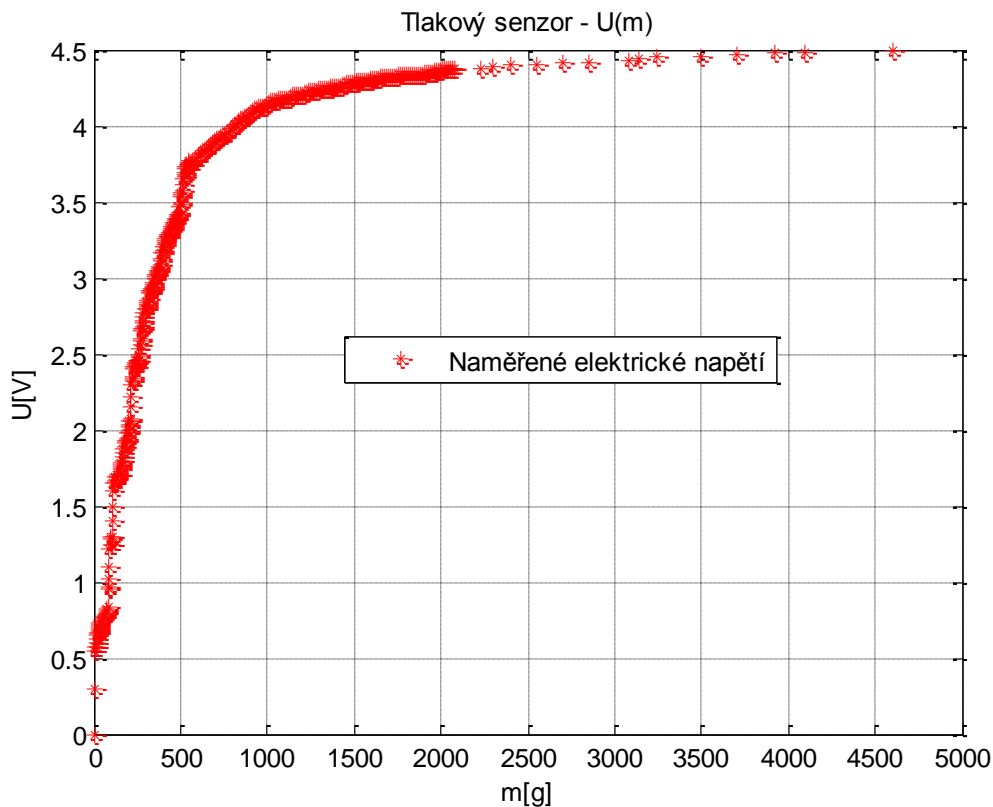
Zdroj: vlastní zpracování

Jak je vidět z grafů 6 a 7, systém naběhne na požadovanou hodnotu velice rychle. Jak pro  $\varphi$  rovno  $-10^\circ$ , tak pro  $\varphi$  rovno  $10^\circ$  systém nenaběhl na přesně požadovanou hodnotu, to až v čase přes 30 s. Do té doby byl ale systém na velice pěkné odchylce pod jedno procento. Maximálně byla tato procentuální odchylka na hodnotě 0,9 % pro  $\varphi = 10^\circ$  a pro  $\varphi = -10^\circ$  byla tato odchylka rovna 0,92 %.

#### 4.5. Kalibrace tlakových senzorů

Tato část se zabývá kalibrací tlakových senzorů tak, aby se potlačila jejich nelinearita.

Tlakový senzor v rukojeti byl použit tak, aby měřil sílu působící na rukojeť neboli sílu, kterou člověk musí vykonat k tlačení vozíku kupředu. Jelikož se ale při prvotním odzkoušení rukojeti ukázalo, že napětí, které je určeno silou působící na rukojeť, je popsáno nějakou funkcí. Proto bylo zapotřebí odměřit závislost síly působící na rukojeť a napětí na děliči. Pro přesné určení zatížení na rukojeť se jako zátěž určila voda. Tato voda byla pomocí injekční stříkačky přidávána do plastových lahví zavěšených na rukojeti tak, aby tlačily na tlakový senzor v rukojeti. Odezva senzorů na zatížení pomocí vody je vidět v grafu 8.

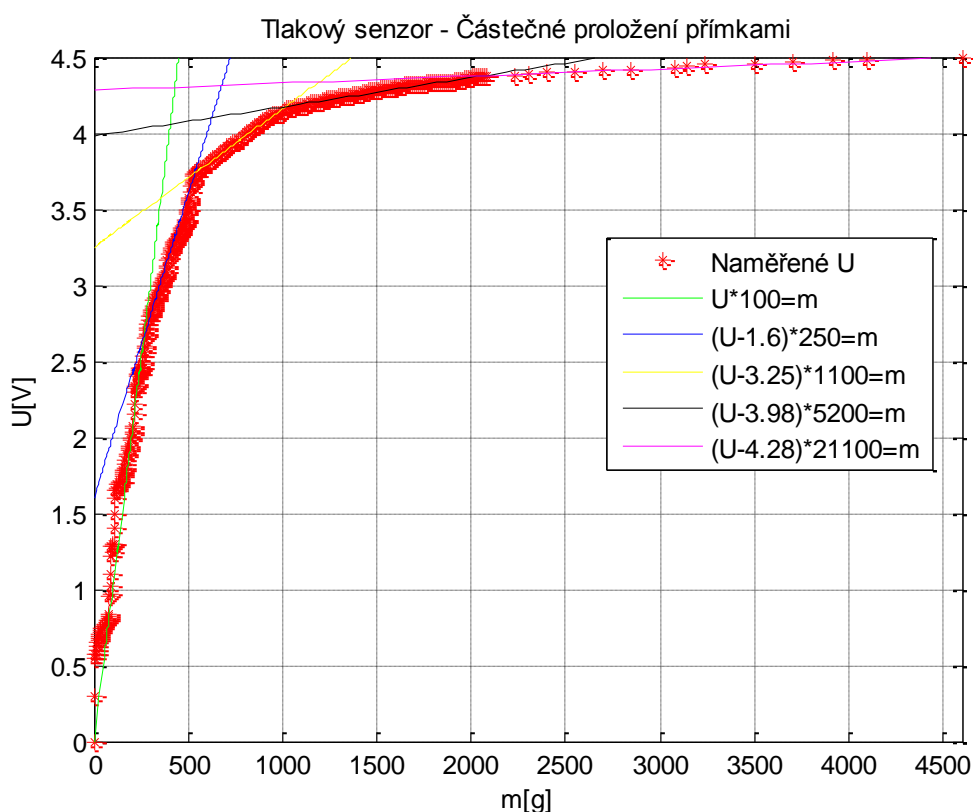


Grafy 8 – Odezva tlakových senzorů v rukojeti na zatížení

Zdroj: vlastní zpracování

Jako nulovou hodnotu na ose x, kam byla vynášena hmotnost, byla zvolena hmotnost těsně před tím, než se objevilo napětí na napěťovém děliči. Dále byla přidávána voda, z počátku po 2 g a poté se tato hodnota zvětšovala tak, aby byl odměřený „kloub co nejpřesněji“.

Pokud bychom nechali senzory takto a jako požadovanou hodnotu bychom používaly násobek napětí U, vozík by se choval velice nepřírozně. Bude proto zapotřebí najít funkci popisující vztah U(m), aby byly následně hodnoty přenásobeny inverzní funkcí tak, aby vznikla lineární závislost působící síly na rukojeť prezentovaná napětím na děliči a požadovanou hodnotou rychlosti. Jelikož ale inverzní funkce byla popsána vysokým stupněm polynomu, bylo rozhodnuto, že se křivka po částech proloží úsečkami, u kterých je následně velice jednoduše vypočitatelná inverze. Nejdříve byly nalezeny přímky, které byly prokládány po částech naměřenými hodnotami. Toto proložení přímkami je vidět v grafu 9.

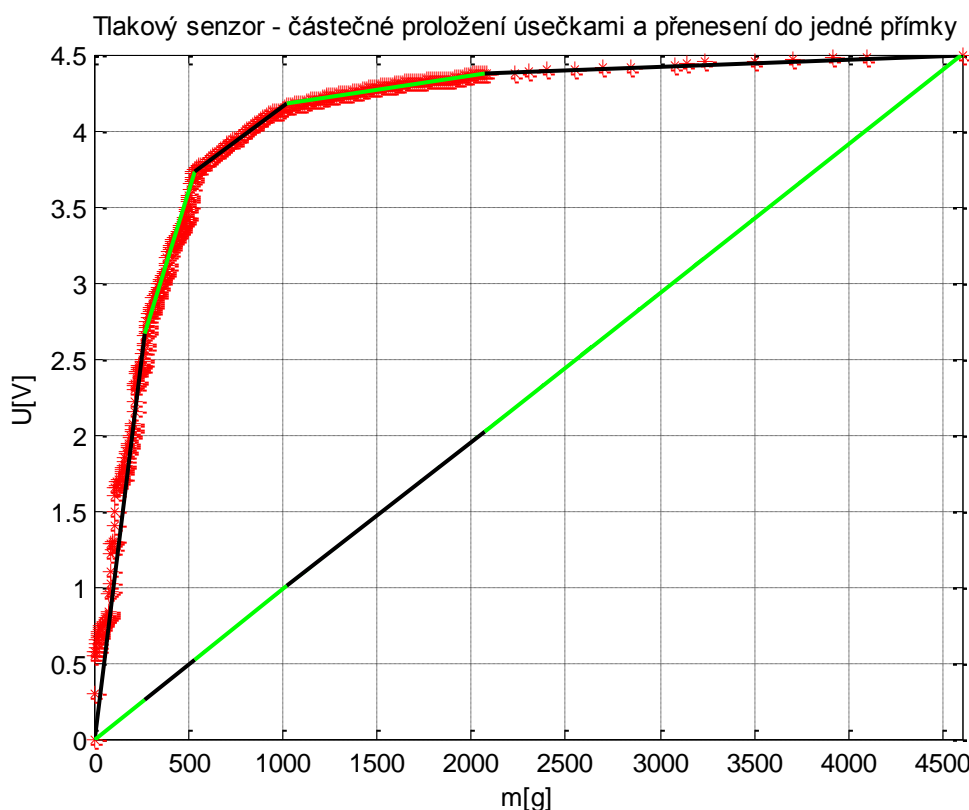


Grafy 9 – Částečné proložení odezvy tlakových senzorů v rukojeti na zatížení přímkami

Zdroj: vlastní zpracování

Místa, ve kterých se sousedící přímky střetly, byla určena jako začátky a konce úseček. U přímky  $U \cdot 100 = m$  a  $(U - 1.6) \cdot 250 = m$  to byl bod, v němž napětí nabývalo na hodnotě 2,665 V. U následující dvojice  $(U - 1.6) \cdot 250 = m$  a  $(U - 3.25) \cdot 1100 = m$  to bylo

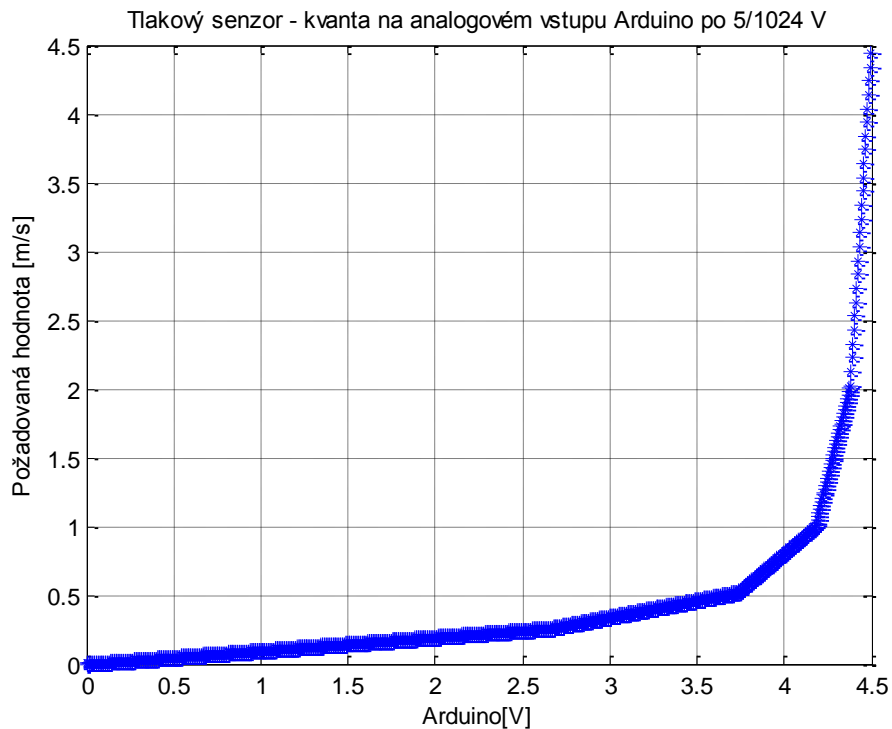
3,736 V. U další dvojice  $(U-3.25)*1100=m$  a  $(U-3.98)*5200=m$  to bylo 4,18 V a u poslední dvojice  $(U-3.98)*5200=m$  a  $(U-4.28)*21100=m$  to bylo 4,38 V. Od nuly přes tyto hranice byly nalezeny úsečky, které se následně přenásobily tak, aby byly součástí jedné přímky, která následně bude popisovat vztah mezi silou prezentovanou napětím a požadovanou hodnotou. Toto proložení úsečkami a přenásobení do jedné přímky je vidět v grafu 10.



**Grafy 10 – Proložení odezvy tlakových senzorů v rukojeti na zatížení úsečkami a přenesení do jedné přímky**  
Zdroj: vlastní zpracování

První úsečka od 0 V do 2,665 V se vydělila 10,222. Další úsečka od 2,665 V do 3,736 V se vydělila hodnotou 4,0888 a navíc bylo odečteno -0,391. Následující úsečka od napětí 3,736V po 4,18 se vydělila 0,9293 a odečetlo se 3,4976. Další od 4,18 V do 4,38 V byla vydělena 0,195 a bylo odečteno 20,4355. Poslední úsečka od 4,38 V do 4,5 V byla vydělena 0,04845 a odečteno bylo 88,3764. Tato přímka se dále celá přenásobí tak, aby výstup odpovídal požadované hodnotě rychlosti. Jak je vidět z grafu, požadovaná hodnota bude citlivější v počátku, což by však nemělo dělat problémy. Jako pokus bylo vyzkoušeno, jak by vypadala odezva systému na lineárně vzrůstající napětí při kvantování do 1024 úrovní, které má Arduino od 0 do 5 V na analogovém vstupu na požadovanou hodnotu. Toto kvantování hodnot napětí je vidět v grafu 11.

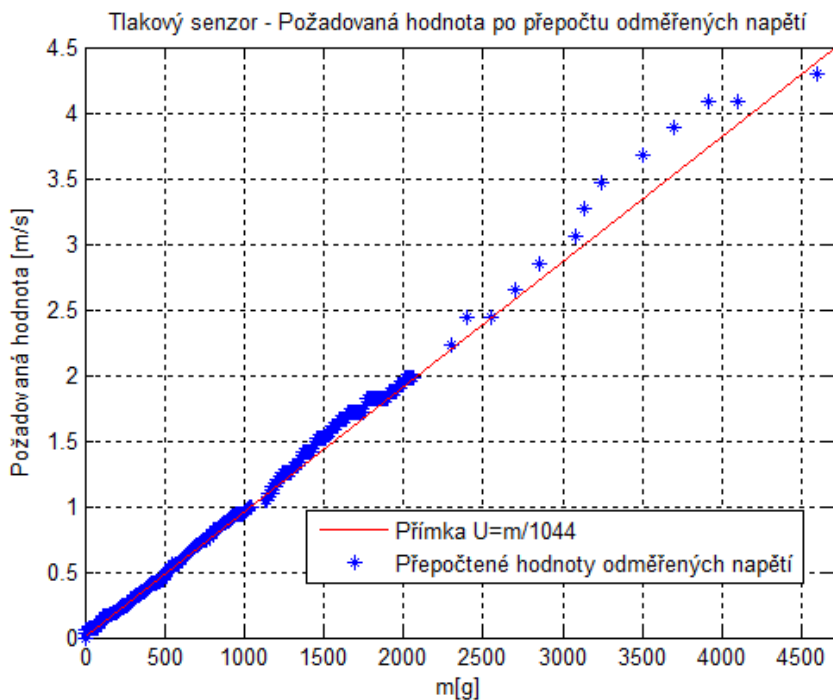




Grafy 11 – Kvanta Arduina v přepočítání napětí na požadovanou hodnotu

Zdroj: vlastní zpracování

Dále bylo odzkoušeno, jak by vypadala odezva systému na lineárně vzrůstající tlak ku přepočtu počátečních vstupních dat napětí na požadovanou hodnotu. Tato odezva je vidět v grafu 12.



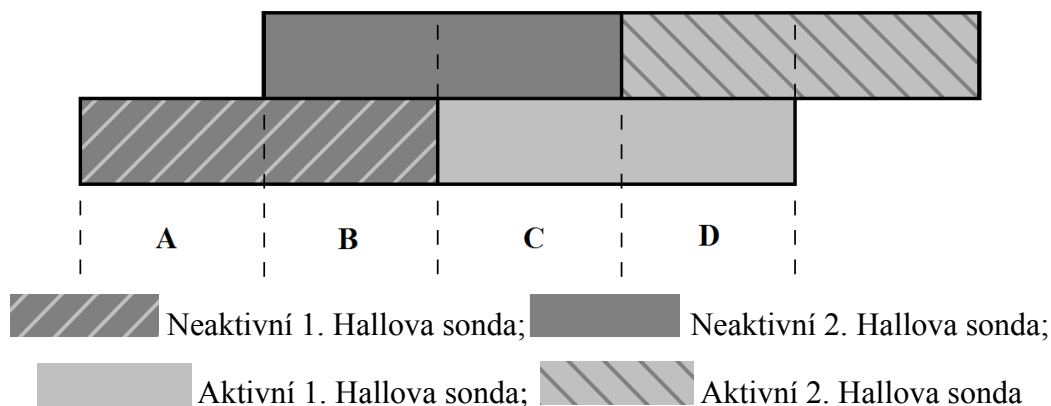
Grafy 12 – Odezva tlakových senzorů v rukojeti na přepočtenou požadovanou hodnotu

Zdroj: vlastní zpracování

## 4.6. Kalibrace Hallových sond

Tato část se zabývá tím, jak pomocí Hallových sond a permanentních magnetů přidělaných na kolo měřit rychlost otáčejícího se kola.

Zobrazení úseku aktivace a deaktivace Hallových sond je vidět na výkresu 21.



Výkres 21 – Vzdálenost magnetů podle aktivace Hallových sond

Zdroj: vlastní zpracování

Byly odměřeny vzdálenosti intervalů A, B, C a D na celém kole. Tyto intervaly se měřily na největším poloměru, a to na obvodu kola. Hodnoty těchto intervalů jsou vidět v tabulce 2.

													Průměr
A[mm]	9	11	6	17	14	10	15	20	16	17	13	14	13,50
B[mm]	19	19	18	16	8	17	13	12	15	15	18	20	15,83
C[mm]	13	11	10	10	10	10	11	10	13	11	11	10	10,83
D[mm]	13	14	9	12	15	14	12	7	7	11	8	13	11,25

Tabulka 2 – Vzdálenosti magnetů na kole podle sepnutí Hallových sond

Zdroj: vlastní zpracování

Tyto odměřené hodnoty se musely přepočítat tak, aby odpovídaly vzdálenostem v místě, kde se měří Hallovou sondou magnety přidělané na disk. Jelikož byla hodnota obvodu kola a disku známá, tyto hodnoty se jednoduše přepočítaly. Tyto přepočtené hodnoty jsou vidět v tabulce 3.

													Průměr	Suma
A[mm]	3,7	4,5	2,5	6,9	5,7	4,1	6,1	8,2	6,5	6,9	5,3	5,7	5,51	16,41
B[mm]	7,8	7,8	7,4	6,5	3,3	6,9	5,3	4,9	6,1	6,1	7,4	8,2	6,47	
C[mm]	5,3	4,5	4,1	4,1	4,1	4,1	4,5	4,1	5,3	4,5	4,5	4,1	4,42	
D[mm]	5,3	5,7	3,7	4,9	6,1	5,7	4,9	2,9	2,9	4,5	3,3	5,3	4,59	

Tabulka 3 – Přepočet vzdáleností magnetů na disku podle sepnutí Hallových sond

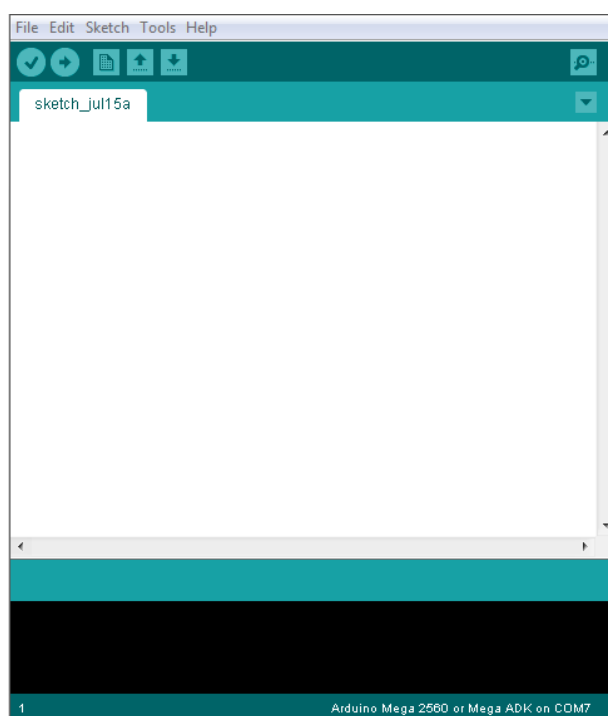
Zdroj: vlastní zpracování







## 5. Program

Tato část se zabývá částmi řídicího programu, který se bude starat o měření výstupů senzorů a také akčních zásahů.

### 5.1. Úvod

Program se píše pomocí jazyka syntakticky podobnému jazyku C a C++. Tento program se pro Arduino píše v jednoduchém textovém editoru s názvem Arduino 1.0.5. Tento editor ukládá své soubory pod příponou .ino. Obsahuje lištu s tlačítky, místo, kde se píše program, místo, kde se vypisují informace o chybách, a průběhy nahrávání a validity a stavovou lištu, kde je napsán typ Arduina a rozhraní, pomocí kterého je připojen k počítači (Program Arduino 1.0.5). Úvodní okno programu je vidět na obrázku 2.



-  Tlačítko pro validace programu
-  Tlačítko pro nahrání programu do Arduina
-  Tlačítko pro otevření nového listu
-  Tlačítko pro otevření vytvořeného programu
-  Tlačítko pro uložení vytvořeného programu
-  Tlačítko pro sledování sériové komunikace

Obrázek 2 - Okno programu Arduino

Zdroj: Program Arduino 1.0.5

## 5.2. Knihovny – Arduino

Tato část se zabývá knihovnami, které se importují do Arduina a svojí funkcí ulehčují práci při programování.

Knihovny jsou softwarové balíčky, které se importují do Arduina. Tento balíček je tvořen souborem s příponou .h, který obsahuje vlastní kód knihovny, a souborem .c nebo .cpp, který obsahuje předpisy jednotlivých tříd a funkcí. Pomocí těchto knihoven můžeme například jednoduše používat výpisy na displeje nebo pomocí nich zapisovat hodnoty do vnitřní EEPROM paměti nebo na SD kartu. U asistenčního systému bylo zapotřebí knihovny „PID\_v1“ a „Queue“. Knihovny se naimportují do programu pomocí příkazů:

```
#include < PID_v1.h>  
#include < Queue.h>
```

### 5.2.1. Knihovna PID

Tato část se zabývá knihovnou PID, která simuluje chování PID regulátoru v programovém prostředí Arduino.

Tato knihovna simuluje chování proporcionálně integračně derivačního zesilovače. Tato knihovna obsahuje vstupy „SETPOINT“, na který se zapisuje požadovaná hodnota regulace, a „INPUT“, na který se zapisuje skutečná hodnota. Také obsahuje jeden výstup „OUTPUT“, ze kterého lze číst akční zásahy. V této knihovně jde samozřejmě nastavit zesílení  $K_P$ ,  $K_I$  a  $K_D$ . Dále jde také nastavit rozmezí hodnot výstupu a čas vzorků PID `SetSampleTime`, pomocí kterých se vypočítává akční zásah integračního a derivačního zesilovače PID pomocí rovnic 5.1 a 5.2:

$$U_D = \frac{e(t) - e(t - \text{SetSampleTime})}{\text{SetSampleTime}} * K_D \quad (5.1)$$

$$U_I = \sum_{i=0}^t [e(t)] * K_I ; t \% \text{SetSampleTime} = !0 \quad (5.2)$$

Výsledný akční zásah se poté tedy vypočte pomocí vzorce 5.3:

$$U = e(t) * K_P + U_I(t) + U_D(t) \quad (5.3)$$

(Knihovna Arduino-PID 1.1.0)

### 5.2.2. Knihovna Queue

Tato část se zabývá knihovnou Queue, která simuluje chování multitaskingu v programovém prostředí Arduino.

Tato knihovna je vytvořena jako jednoduchý plánovač. Je určen pro programy, u kterých potřebujeme sesbírat určitý počet vstupních dat a poté je vyhodnocovat. Pomocí tohoto plánovače úloh lze jednoduše nastavit, aby požadavky na vykonání úkonu, které jsou potřeba, se do vykonávací fronty přidávaly častěji, než náročnější úkony, které trvají delší časový okamžik. V této knihovně se dá nastavit, po jak dlouhé době se mají jednotlivé úkony přidávat do vykonávací fronty. Dále se dá zadávat, po jak dlouhé době se má fronta kontrolovat, zda v ní není nový požadavek na vykonání úkonu (Knihovna AVRQueue).

### 5.3. Vstupy

Tato část se zabývá tím, jaké jsou přivedeny vstupy na řídicí jednotku Arduino.

Jako vstupy řídicí jednotky budou dva druhy senzorů a jeden spínač. První senzory budou Hallovy sondy, díky kterým se bude měřit rychlost vozíku. Druhé senzory, budou tlakové senzory v rukojeti. V této rukojeti bude také spínač, který bude přepínat určité módy programu. Senzory tlaku budou připojeny na desetibitovém analogovém vstupu pomocí děliče napětí se 100kΩ rezistory. Hodnoty z těchto snímačů se dále softwarově přezásobují pomocí korekcí, které byly počítány v kapitole 4.5. Výstupy z Hallových senzorů byly připojeny na digitální vstupy, na které bylo vnitřně přivedeno napětí 5 V pomocí příkazu `digitalWrite(„číslo pinu“,1);`. Toto napětí poté Hallova sonda svedla na zem nebo ne, podle toho, zda byl magnet v blízkosti Hallovy sondy. Tlačítko bylo přivedeno také na digitální vstup.

### 5.4. Výstupy

Tato část se zabývá tím, jaké jsou vyvedeny výstupy na řídicí jednotce Arduino.

Jako výstupy řídicí jednotky budou vstupy do H-můstku, který bude spínat motor. Tento výstup je fázově korektní pulzně-šířkový výstup, který je kvantován do osmi bitů. Další výstupy budou výstupy na dvě luminiscenční diody, které budou signalizovat chod programu. Tyto luminiscenční diody byly také připojeny na pulzně-šířkové výstupy, aby bylo možné nastavit napětí na luminiscenčních diodách.

## 5.5. Časovače

Tato část se zabývá časovači, které jsou používány v řídicí jednotce Arduino.

V případě navrhovaného asistenčního systému budou jistě potřeba časovače, které by spočítaly, jak dlouho trvá, než se přesune magnet nad Hallovu sondu ku předchozímu magnetu. A díky znalosti vzdálenosti magnetů lze vypočítat, jak rychle se pohonná jednotka a tím pádem i celý vozík pohybuje. Jelikož jsou od sebe magnety vzdáleny přibližně 16,6 mm, rychlost vypočítáme jako vzdálenost děleno čas. Maximální rychlost, kterou předpokládáme, že vozík pojede, je 2 m/s, což znamená, že náš úsek jedné vzdálenosti magnetů ujede za 8,3 ms. Proto musíme nastavit časovač tak, aby za tuto dobu nasčítal dostatek pulzů k přesnému změření rychlosti. Byl zvolen časovač 5, na kterém se nastavila předdělička na hodnotu 8 a dělička na hodnotu 20, čímž jsme dostali pulzy o periodě 10  $\mu$ s. To by pro náš případ maximální odhadované rychlosti znamenalo, že by bylo napočítáno 830 pulzů. Pro průměrnou rychlost 1 m/s by to bylo 1660 pulzů.

```
TCCR5A=0; // Nastaví registry časovače 5 na čtení a zápis
TCCR5B=0; // Povolení časovače 5 pro čítání
TCNT5=0; // Nastaví aktuální čas v časovače 5 na 0
OCR5A=20; // 16Mh/8=>2MHz/20=10 dělička
TCCR5B |= (1<<WGM52); //nové spuštění CTC mode
TCCR5B |= (1 << CS51); //16Mh/8=2MHz předdělička
TIMSK5 |= (1<<OCIE5A); //umožní časovači porovnat přerušeni

ISR(TIMER5_COMPA_vect){ //metoda načítává hodnoty pulzů
časovače 5
    time=time+1;
}
```

Dalším problémem byla frekvence PWM. Při zápisu pomocí příkazu `analogWrite(„číslo pinu“, „střední hodnota PWM (0÷255)“)` byla na základní hodnotě 488,28 Hz. Tato frekvence byla totiž dosažena tím, že se signál z krystalu vydělil 64 předděličkou a dále se vydělil 256 děličkou, což znamená 976,5625 Hz. Tato hodnota se ale ještě vydělí dvěma, neboť Arduino využívá pro analogový zápis fázově korektní PWM. Tato frekvence ale budila v elektromotoru vlastní kmity motoru, které následně vydávaly hlasitý pískavý zvuk. Tato frekvence se proto musela zvednout na vyšší frekvenci. Bylo rozhodnuto, že se použije další vyšší frekvence, neboť jsou dané určité velikosti dělení předděličky. Předdělička byla tedy nastavena na hodnotu 1 a následně byla nastavena na hodnotu 8, čímž se docílilo frekvence  $3906,25 \text{ Hz} = 16 \text{ MHz} / 256 / 8 / 2$

```
TCCR2B &= ~7; // nastaví předděličku na PWM na 1, která se
dále využívá při analogWrite()
TCCR2B |= 2; //změna PWM prekvence pro piny 9 a 10 na
3906.25Hz
```

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{IO}/(\text{No prescaling})$
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge
1	1	1	External clock source on T0 pin. Clock on rising edge

**Tabulka 4 – Nastavení předděliček**

**Zdroj:** <https://docs.google.com/viewer?url=http://www.atmel.com/Images/doc2549.pdf>

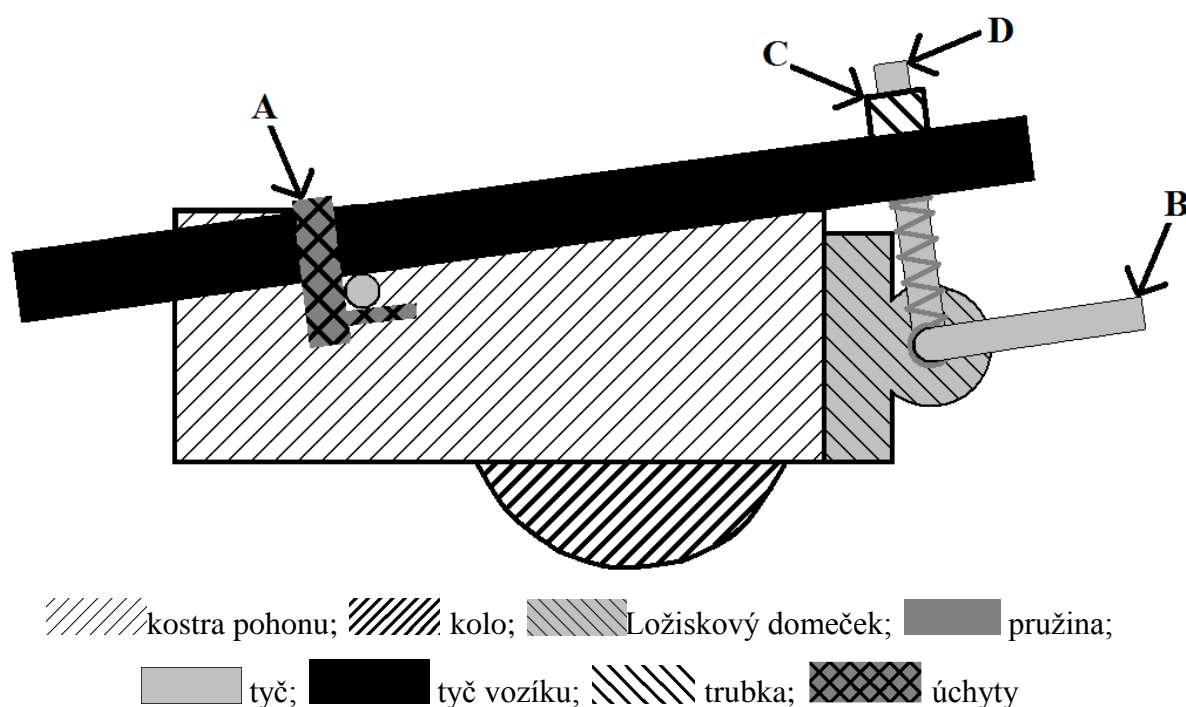
Tato tabulka 4 určuje, jakým číslem bude dělit předdělička. Na počátku jsou všechny tři registry na hodnotě 0. Pokud třeba chceme nastavit hodnotu předděličky na 1024 u časovače 3, musíme do registrů CS32 a CS30 nastavit hodnotu 1. A to buď příkazy  $TCCR3B \mid= (1 \ll CS30)$  a  $TCCR3B \mid= (1 \ll CS32)$  pro každý registr zvlášť, nebo jedním příkazem  $TCCR3B \mid= 5$ . Číslo 5 se totiž v binární vyjádření píše jako 101, kde každá číslice je hodnota jednoho ze tří registrů CSXX.

## 6. Testy

V této kapitole bylo odzkoušeno, jak se asistenční systém bude nasazovat, jak asistenční systém měří rychlost, a na závěr, jak dobře reguluje jízdu vozíku.

### 6.1. Nasazení asistenčního systému na invalidní vozík

Asistenční systém se do vozíku zkoušel nasazovat mnoha způsoby. Mnoho z nich však bylo nepraktických a člověk by se při nich ušpinil. Výkres konečného umístění asistenčního systému na vozíku je vidět na výkresu 22.



Výkres 22 – Nasazování asistenčního systému na vozík

Zdroj: vlastní zpracování

Nejvýhodnější možností se jeví způsob, při kterém člověk jako první nastrčí přední opěrnou tyč do úchyty A. Poté ve vzpřímené poloze zvedne vozík za jeho madla a nohou si jednoduše přizvedne držák B tak, aby byla tyč D nad trubicí C. Poté se vozík pouze položí na zem a tyč D s trubicí C utvoří jakýsi píst. Do dílu D na místo, kam ukazuje šipka, se poté zastrčí závlačky tak, aby se části pístu nerozpojily. Dále se rukojeť se senzory pouze připevní na rukojeti vozíku. Jako poslední krok se připevní baterie na záda vozíku a spojí se konektor vedoucí z baterie.



## 6.2. Měření rychlosti

Tato část pojednává o tom, jak dobře asistenční systém pozná rychlost jízdy a směr pohybu invalidního vozíku.

Test měření rychlosti proběhl velice dobře. Rychlost se měřila na mm za sekundu, a pokud byla rychlost nižší než 16,41 milimetrů za sekundu, což odpovídalo tomu, že se jednu sekundu nezměřila žádná rychlost, zaokrouhlila se rychlost na 0 mm/s. To bylo kvůli tomu, aby cyklus měření rychlosti nezdržoval běh programu.

## 6.3. Jízda (regulace)

Tato část se zabývá testem regulace asistenčního systému.

Při prvním testu jízdy prázdného vozíku došlo k nehodě a byly proraženy dva tranzistory. Jelikož proudově a napětově to měly tranzistory zvládnout, předpokládá se, že tranzistory to neunesly výkonově. Proto byl obvod nasimulován v programu Circuit Simulator, jak je vidět na schématu 8.

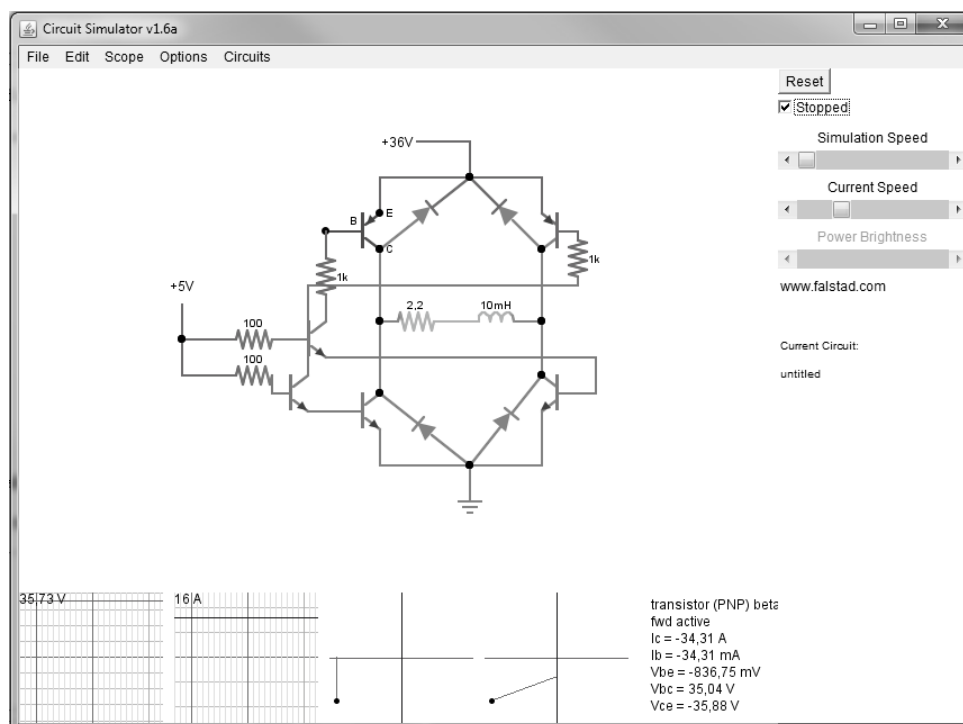


Schéma 8 – H-můstek nasimulovaný v programu Circuit Simulator

Zdroj: Program Circuit Simulator v1.6a

Bylo zjištěno, že pokud by se na oba dva vstupy H-můstku přivedlo napětí, tak by se obě větve H-můstku otevřely a na tranzistorech by se ztrácel výkon, který by byl větší než 1000 W. Tuto hodnotu ovšem tranzistory nevydržely, neboť sou dělány na ztrátový výkon 200 W. V programu, který spíná H-můstek, byl tento problém vyřešen, takže není

možné, aby byly sepnuty obě dvě větve najednou. Je však možné, že pokud došlo k náhlé změně sepnutí větví, mohla větev zůstat ještě okamžik otevřená díky parazitní kapacitě (Circuit Simulator v1.6a).

Proto bylo řízení otestováno na modelu, který byl namodelován pomocí softwarového prostředku Simulink a je vidět na schématu 9.

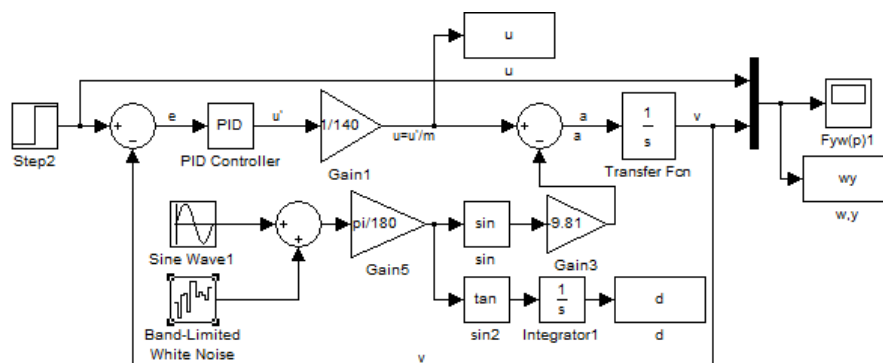
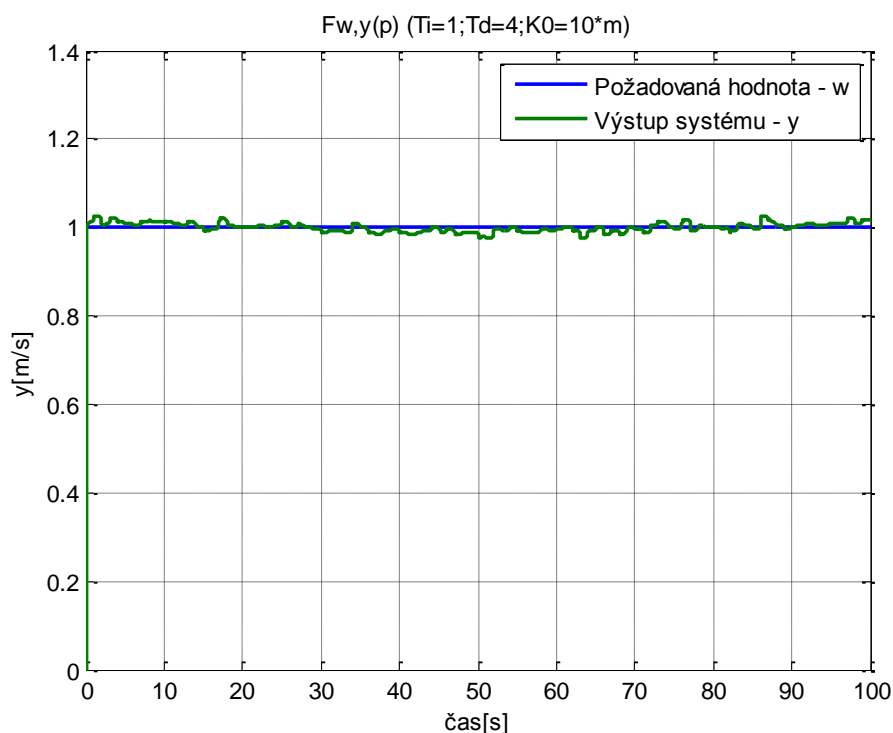


Schéma 9 – Model systému v Simulinku

Zdroj: vlastní zpracování

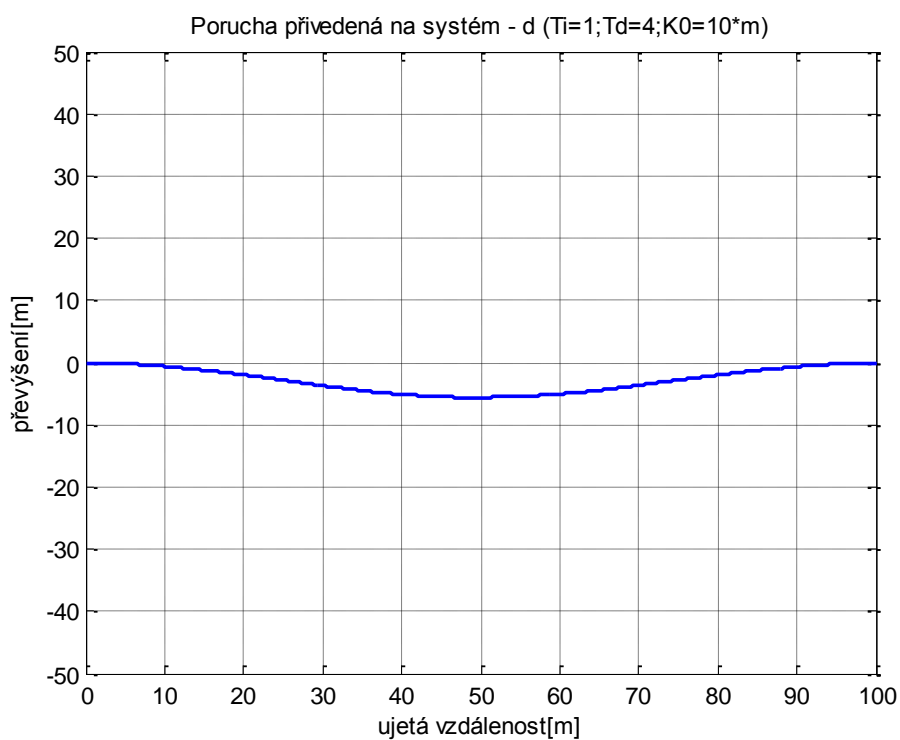
Na tento systém byl na vstup přiveden jednotkový skok a jako porucha byl přiveden součet sinusový signál a bílého šumu, který prezentoval úhel nakloněné roviny, po které asistenční systém s invalidním vozíkem bude jezdit. Odezvu systému na jednotkový skok s poruchou je vidět na grafu 13.



Grafy 13 – Odezva systému na jednotkový skok a poruchu

Zdroj: vlastní zpracování

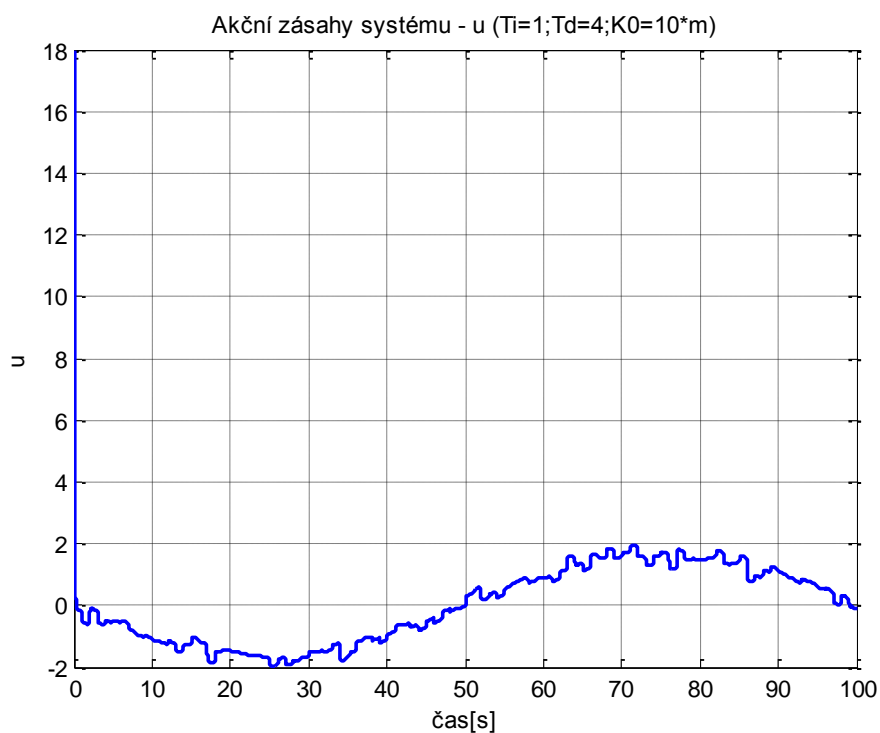
Průběh poruchy přivedený na systém je vidět na grafu 14.



Grafy 14 – Porucha přivedená na systému

Zdroj: vlastní zpracování

Akční zásahy regulátoru jsou vidět na grafu 15.



Grafy 15 – Akční zásahy regulátoru

Zdroj: vlastní zpracování

Jak je vidět z grafů, systém díky regulátoru naběhne na požadovanou hodnotu velice rychle. Poruchy, které byly přivedeny na systém, byly také velice dobře odregulovány. Nesmíme ale zapomínat, že při návrhu modelu bylo zanedbáno tření a výkonové omezení motoru. Z toho vyplývá, že reálný systém by „nenaběhl“ na požadovanou hodnotu tak rychle jako při simulacích, jak už bylo zmíněno v části 4.4.

## 7. Závěr

V úvodu této bakalářské práce bylo stanoveno několik dílčích cílů. Prvním z nich bylo navrhnout konstrukci asistenčního systému. Tento cíl byl splněn, navržená konstrukce se ukázala jako efektivně navržená, přestože byla vyrobená pouze jako jediná.

Druhým dílčím cílem bylo navržení a výroba elektroniky. Motor se ukázal jako dostatečně silný na rozpořádání člověka sedícího na invalidním vozíku. Také navržení plošného spoje pomocného obvodu bylo správné, neboť napájení a děliče napětí na tomto spoji fungovaly dobře. I plošný spoj H-můstek byl navržen správně, jelikož dokázal spínat motor. Také druhý dílčí cíl této práce tak byl splněn.

Posledním dílčím cílem byl návrh řízení asistenčního systému. Byl odvozen model pomocí Lagrangeových rovnic. Na tento model byl následně nalezen regulátor a nastaveny jeho parametry. Simulací uzavřeného systému s regulátorem bylo potvrzeno, že parametry regulátoru byly zvoleny správně. Úspěšně byla také odzkoušena rukojeť s tlakovými senzory a Hallovy sondy, které za pomoci programu v Arduinu měřily rychlost a směr otáčení velice dobře. Řídící systém Arduino fungoval při simulacích regulace také správně. Třetí dílčí cíl této práce tak byl splněn.

Jediným problémem, který se v průběhu vypracování této bakalářské práce vyskytl, byla skutečnost, že při závěrečném testu byla proražena jedna větev plošného spoje H-můstku. Nebylo tedy možné odzkoušet plně funkčnost řízení na prototypu asistenčního systému. Zbývá tedy vyřešit pouze tento problém proražení tranzistorů. Řešením může být programová změna, pokud se ukáže jaké nedostatečné, bude muset být navržen a vytvořen jiný H-můstek.

I přes vzniklý problém se podařilo hlavní cíl této bakalářské práce i všechny její dílčí cíle splnit.

## 8. Literatura a zdroje

Melichar, J. 2011. Lineární systémy. Plzeň: Západočeská univerzita, 117 s.

Schlegel, M. 2007. Mechanické systémy [online, cit. 20. 3. 2013]. Dostupné z WWW: <[https://portal.zcu.cz/wps/PA\\_Courseware/DownloadDokumentu?id=5849](https://portal.zcu.cz/wps/PA_Courseware/DownloadDokumentu?id=5849)>.

Program Arduino 1.0.5. Dostupné z WWW: <<http://arduino.cc/en/Main/Software>>.

Program Circuit Simulator v1.6a. Dostupné z WWW: <<http://www.falstad.com/circuit/>>.

Knihovna Arduino-PID 1.1.0. Dostupné z WWW: <<https://github.com/br3ttb/Arduino-PID-Library/>>.

Knihovna AVRQueue. Dostupné z WWW: <<https://github.com/Zuph/AVRQueue>>.