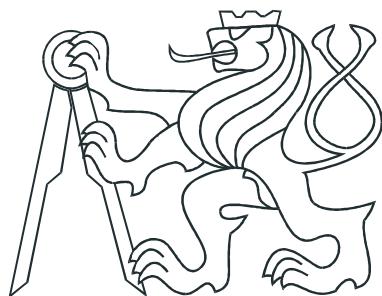


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

**Návrh řídicího systému pro autodráhové
autíčko pro soutěž Freescale Race Challenge**

Praha, 2013

Autor: Bc. Jaroslav Marek

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne 9.5. 2013

J. Novák
podpis

Poděkování

Především bych chtěl poděkovat vedoucímu diplomové práce Ing. Zdeňku Hurákoví, Ph.D., za jeho profesionální přístup, věcné připomínky, rady a také za jeho vstřícnost. Dále bych rád poděkoval svým spolužákům - především Ing. Danu Martincovi, kteří mi podali pomocnou ruku, kdykoliv jsem potřeboval, Ing. Vítu Hlinovskému, CSc., za zapůjčení autíčka, příslušné elektroniky a testovací dráhy a v neposlední řadě také své rodině a přítelkyni, kteří mi byli vždy oporou.

Abstrakt

Tato diplomová práce popisuje návrh řídicího systému pro standardní autodráhové autíčko určené pro jízdu na autodráhách Carrera Evolution. Práce se zabývá popisem použité elektroniky, matematickým modelováním dynamiky autíčka, až po samotné na-programování sběru dat, řídicí struktury a představení reinforcement learningu. O chod celého systému se stará mikrokontrolér MCF51JM64, který je umístěn uvnitř autíčka na osazené desce plošných spojů od společnosti Freescale Semiconductor, Rožnov pod Radhoštěm. Pohon autíčka zajišťuje stejnosměrný motor s cizím buzením, který je řízen prostřednictvím integrovaného H-můstku MC33931. K analýze průběhu jízdy byl využíván akcelerometr MMA7361L, gyro senzor LY530ALH a rychlostní senzor tvořený reflexním senzorem QRE113 a rozetou. Pro měření proudu bylo vytvořeno zapojení s diferenčním zesilovačem. Naměřená data je možné ukládat na μ SD kartu. Programování mikrokontroléru bylo zajištěno spojením počítače a mikrokontroléru prostřednictvím mini USB portu. Pro řízení rychlosti autíčka byla použita kaskádní struktura tvořená proudovou a rychlostní smyčkou. Dále je popsána příprava na soutěž Freescale Race Challenge 2012. Bylo analyzováno Q-učení, které patří do skupiny algoritmů reinforcement learningu a byl vyzkoušen jednoduchý učící algoritmus.

Abstract

This diploma thesis describes design of control system for standard slot car for Carrera Evolution racing tracks. The thesis deals with description of used electronic components, mathematical modeling of slot car dynamics, through programming of data acquisition, control structure and provides an introduction into reinforcement learning. The slot car is controlled by microcontroller MCF51JM64, which is situated on printed circuit board inside slot car and designed by Freescale Semiconductor, Rožnov pod Radhoštěm. The slot car is driven by DC motor, which is controlled through integrated H-bridge MC33931. Accelerometer MMA7361L, gyro sensor LY530ALH and velocity sensor which consists of reflexive sensor QRE113 and rosette were used for slot car's ride analysis. The circuit with differential amplifier was created for current measurement. Measured data can be saved on μ SD card. Programming of microcontroller is ensured by connecting computer and microcontroller through mini USB port. Cascade structure, which is composed of current and velocity loop was used for velocity control of slot car. Then preparation for Freescale Race Challenge 2012 is described. Q-learning was analyzed, which belongs to reinforcement learning algorithms and was tried simple learning algorithm.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jaroslav Marek**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Návrh řídicího systému pro autodráhové autíčko pro soutěž Freescale Race Challenge**

Pokyny pro vypracování:

- Navrhněte a implementujte řídicí systém pro autodráhové autíčko pro závod Freescale Race Challenge 2012. Zúčastněte se soutěže.
1. Jako výchozí hardwarový návrh použijte elektroniku poskytnutou organizátorem soutěže. V případě potřeby více či méně modifikujte. Zejména půjde o doplnění vhodnými senzory.
 2. Navrhněte vhodnou logickou strukturu řídicího systému (kaskádní vs. stavový regulátor, feedback vs. feedforward atd.).
 3. Modelujte dynamiku vozu a ověřte platnost modelu pomocí experimentů.
 4. Navrhněte regulátor(y) s použitím vhodných metod založených na modelu (model-based design), implementujte a ověřte v experimentech.
 5. Zúčastněte se soutěže Freescale Race Challenge 2012.

Seznam odborné literatury:

Dodá vedoucí práce.

Vedoucí: Ing. Zdeněk Hurák, Ph.D.

Platnost zadání: do konce letního semestru 2012/2013

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 2. 12. 2011

Obsah

Seznam obrázků	viii
Seznam tabulek	x
1 Úvod	1
2 Elektronika	2
2.1 Napájení desky plošných spojů	3
2.2 Mikrokontrolér MCF51JM64	4
2.3 Integrovaný H-můstek MC33931	4
2.4 Senzory	5
2.4.1 Akcelerometr MMA7361L	5
2.4.2 Reflexní senzor QRE1113	7
2.4.2.1 Měření rychlosti	8
2.4.3 Gyro senzor LY530ALH	10
2.4.4 Zapojení s diferenčním zesilovačem	13
2.4.5 Odometrie	16
2.4.6 Detektor křížení	18
2.5 Regulátor napětí LF33CV	19
2.6 Mini USB port a programování mikrokontroléru	20
2.7 μ SD karta	20
2.8 Celkové schéma	21
3 Matematický model	23
3.1 Identifikace systému	23
3.2 Identifikace konstant	25
3.2.1 Identifikace mechanické k_t a elektrické k_e konstanty	25
3.2.2 Identifikace tření	26

3.3	Návrh řízení	29
3.3.1	Návrh spojitéch regulátorů	29
3.3.2	Implementace regulátorů	31
4	Freescale Race Challenge	33
4.1	Analýza problému	34
4.1.1	Fyzikální popis průjezdu zatáčkou	35
4.1.2	Analýza dráhy	35
4.2	Předsoutěžní příprava	37
4.3	Mapování dráhy	37
4.4	Závodní režim	38
4.5	Program	39
4.6	Soutěž - univerzitní kolo	40
4.7	Vlastní soutěžní dráha	40
5	Reinforcement learning	44
5.1	Definice základních pojmu	45
5.1.1	Prostředí	46
5.1.2	Reinforcement funkce	46
5.1.3	Hodnotová funkce a další termíny	47
5.2	Temporal difference learning	48
5.3	Q-učení	49
5.4	Analýza Q-učení pro autíčko na dráze	50
5.5	Popis jednoduchého učícího algoritmu	51
5.5.1	Testování na dráze	51
6	Závěr	53
Literatura		59
A	Elektronické a simulinkové schéma	I
B	Obsah přiloženého CD	IV

Seznam obrázků

2.1	Osazená deska plošných spojů	3
2.2	Schéma napájení z kolejnic dráhy	4
2.3	Ukázka PWM signálu se střídou 50% a 25%	5
2.4	Znázornění principu měření akcelerometru	6
2.5	Schéma zapojení akcelerometru	6
2.6	Naměřené boční zrychlení v klidové poloze při přidržení autíčka	8
2.7	Naměřené boční zrychlení při jízdě na oválné dráze	9
2.8	Reflexní senzor QRE1113	9
2.9	Realizace rychlostního senzoru	10
2.10	Porovnání filtrovaného a nefiltrovaného naměřeného signálu rychlosti	11
2.11	Vnitřní stavba MEMS gyro senzoru	11
2.12	Schema zapojení desky s gyro senzorem	12
2.13	Senzor úhlové rychlosti	13
2.14	Naměřené průběhy z gyro senzoru s nefunkčním filtrem horní propust	14
2.15	Schéma zapojení pro měření proudu	14
2.16	Ukázka naměřeného průběhu proudu	15
2.17	Porovnání naměřených dat s referencí pro 10 kol jízdy	17
2.18	Porovnání naměřených dat s referencí pro 1 kolo jízdy	17
2.19	Schéma zapojení pro měření napětí z kolejnic dráhy	18
2.20	Naměřené průběhy napětí z kolejnic dráhy	19
2.21	Detekování křížení na dráze	19
2.22	Zapojení regulátoru podle doporučení z katalogového listu	20
2.23	Výsledné schéma elektronických úprav	21
2.24	Celkové měřící schéma	22
3.1	Simulinkové blokové schéma modelu autíčka	25
3.2	Závislost točivého momentu na proudu	26

3.3	Závislost třecího momentu na úhlové rychlosti	27
3.4	Porovnání naměřené a simulované rychlosti modelem autíčka	28
3.5	Kaskádní řídicí struktura tvořená proudovou a rychlostní smyčkou	29
3.6	Amplitudová frekvenční charakteristika řízených smyček	30
3.7	Odezva na skokovou změnu referenčního proudu	31
3.8	Odezva na skokové změny referenčního proudu	32
3.9	Odezva autíčka na skokové změny referenční rychlosti	32
4.1	Znázornění průjezdu zatáčkou	35
4.2	Díly dráhy Carrera Evolution	36
4.3	Naměřené boční zrychlení během mapovacího kola	38
4.4	Detekování křížení na dráze během mapovacího kola	39
4.5	Vývojový diagram programu	40
4.6	Fotografie složitější dráhy	41
4.7	Naměřená data na oválné dráze	42
4.8	Nastavované hodnoty činitele plnění <i>PWM</i> pro oválnou dráhu	42
4.9	Naměřená data na složitější dráze	43
4.10	Nastavované hodnoty činitele plnění <i>PWM</i> pro složitější dráhu	43
5.1	Rozhraní agent-prostředí	45
5.2	Schéma přechodů mezi stavy	46
5.3	Nastavované hodnoty činitele plnění <i>PWM</i> pro oválnou dráhu	52
5.4	Nastavované hodnoty činitele plnění <i>PWM</i> pro složitější dráhu	52
6.1	Fotografie upraveného autíčka	53
A.1	Schéma zapojení originální desky plošných spojů	II
A.2	Simulinkové blokové schéma kaskádní řídicí struktury autíčka	III

Seznam tabulek

2.1	Měření ujeté vzdálenosti na 10 kol ($l_{10} = 29400$ mm)	16
2.2	Měření ujeté vzdálenosti na 1 kolo ($l_1 = 2940$ mm)	17
3.1	Identifikované parametry matematického modelu autíčka	28
4.1	Experimentálně zjištěná data pro soutěžní algoritmus	37
6.1	Porovnání závodních časů na 10 kol na oválné dráze	54
6.2	Porovnání závodních časů na 10 kol na složitější dráze	54

Kapitola 1

Úvod

S autíčky na autodráze jsem si hrál už jako malý kluk. Volba tématu diplomové práce ”Návrh řídicího systému pro autodráhové autíčko pro soutěž Freescale Race Challenge mi umožnila vrátit se do dětských let, uplatnit znalosti nabyté během studia a přiučit se něčemu novému. Primárním cílem této diplomové práce byla příprava autíčka a následná účast v soutěži. Autíčko tedy musí být schopné projet neznámou dráhu v co nejkratším čase. Pod pojmem neznámá dráha máme na mysli, že tvar dráhy není předem známý, avšak víme, z jakých stavebních prvků může být sestavena. V tomto případě máme o neznámé dráze alespoň nějakou informaci a můžeme se na projetí takové neznámé dráhy snáze připravit.

Situace se stává zajímavější a zároveň složitější v okamžiku, kdy nemáme žádnou informaci o tom, z jakých stavebních prvků bude dráha sestavena. O dráze tedy nemáme žádnou informaci. V této chvíli můžeme sáhnout po metodě zvané reinforcement learning. Díky této technice učení by autíčko mělo být schopné se naučit efektivně jezdit na zcela neznámé dráze. Učení je založené na metodě pokus-omyl, kdy autíčko zkouší jet různou rychlosťí v jednotlivých úsecích trati. Příslušné situace jsou ohodnoceny a cílem je nalézt posloupnost rozhodnutí, která je nejcennější.

Příprava na soutěž zahrnovala jak přípravu elektroniky (hardwaru), detailněji popsané v Kapitole 2, tak matematické modelování dynamiky, vč. návrhu řízení, blíže rozebrané v Kapitole 3. Přípravě na soutěž a samotné soutěži je věnována Kapitola 4. Sekundárním cílem této diplomové práce bylo z analyzovat metodu reinforcement learningu, toto téma je rozpracováno v Kapitole 5.

Kapitola 2

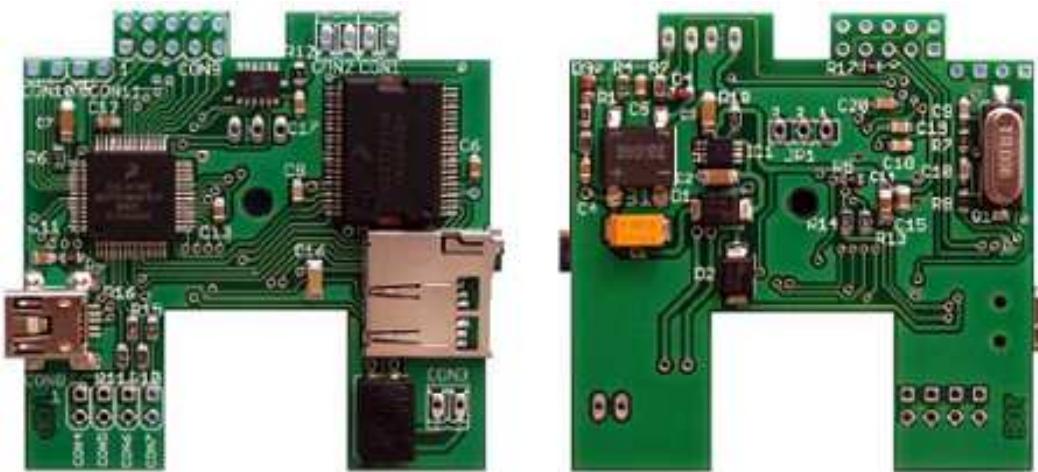
Elektronika

V této kapitole je popsána elektronika použitá v autodráhovém autíčku. Seznámíme se nejen s elektronikou, která byla k dispozici od pořadatele soutěže Freescale Race Challenge, ale také s elektronickými prvky, které byly k základní sestavě přidány. Společně s autíčkem byla od pořadatele soutěže k dispozici osazená deska plošných spojů (obr. 2.1, který byl převzat z (HW.CZ, 2010)) vyrobená společností Freescale Semiconductor, Rožnov pod Radhoštěm. Schéma zapojení základní desky je na obr. A.1 v Příloze A. Na desce se již nacházely následující důležité elektronické součástky:

- mikrokontrolér MCF51JM64 od společnosti Freescale Semiconductor
- integrovaný H-můstek MC33931
- akcelerometr MMA7361L
- slot pro μ SD kartu
- mini USB konektor pro připojení k PC pomocí USB kabelu
- stabilizátor napětí LP2950-D na 3,3V.

Prvotní myšlenkou bylo autíčko vybavit co možná největším počtem senzorů, díky kterým by bylo možné co nejpřesněji analyzovat průběh jízdy po dráze a poté autíčko řídit. Proto byl do autíčka přidán další regulátor napětí na 3,3 V LF33CV, aby stabilizátor napětí na osazené desce plošných spojů nebyl přetízen při napájení další elektroniky. Dále byl přidán reflexní senzor QRE1113 pro měření rychlosti, 2 gyro senzory LY530ALH pro měření úhlové rychlosti a zapojení s diferenčním zesilovačem pro měření proudu. Výsledné schéma všech elektronických úprav můžeme vidět na obr. 2.23. Počítalo se i se senzorem z

optické myši, ale z důvodu malého prostoru uvnitř autíčka a celkem velké desky plošných spojů, na které se senzor s podpůrnou elektronikou a soustavou čoček nacházel, nebylo možné jeho umístění do autíčka realizovat.

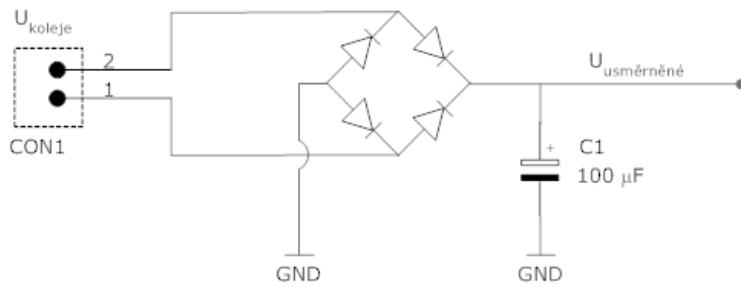


Obrázek 2.1: Osazená deska plošných spojů

2.1 Napájení desky plošných spojů

Deska může být napájena dvěma způsoby. První způsob napájení je přes mini USB konektor připojený k PC a poskytující stejnosměrné napětí 5 V. Tento způsob je určen výhradně k programování mikrokontroléru.

Druhou možností je napájet desku z kolejnic dráhy, čímž dojde ke spuštění uživatelské aplikace. Dráha je napájena laboratorním zdrojem 15 V. Napětí z dráhy je do autíčka přenášeno přes stírací kartáče. Aby se nemuselo řešit, jak se autíčko na dráhu umístí, nebo-li v jakém směru se bude pohybovat, nachází se za stíracími kartáči dvoucestný usměrňovač, který vstupní napětí usměrní na správnou polaritu. Tím zamezíme možnému přepólování elektroniky. Napětí za usměrňovačem je filtrováno kondenzátorem C1 a je nižší o úbytek napětí na dvou diodách, tedy přibližně o $2 \cdot 0,7$ V. Zapojení napájení z kolejnic dráhy je na obr. 2.2.



Obrázek 2.2: Schéma napájení z kolejnic dráhy

2.2 Mikrokontrolér MCF51JM64

Řídicí jednotkou celého autíčka je mikrokontrolér s 32-bitovým RISC jádrem V1 ColdFire. Patří do rodiny mikrokontrolérů označovaných "systems-on-chip" (SoCs), kdy se na jednom čipu integrují všechny komponenty a funkční bloky. Jedná se o variantu, která má 64 pinů. Mikrokontrolér je náplňen 3,3 V a pracovní frekvence procesorového jádra je nastavena na 48 MHz. Uživatelskou aplikaci je možné nahrát do Flash paměti o velikosti 64 kB. Pro autíčko pohybující se na autodráze je tato paměť výhodná, protože při výpadku napájení nedojde ke ztrátě uloženého programu. Pokud autíčko z dráhy odebereme nebo nám z ní vypadne, nemusíme program opětovně nahrávat. Operační paměť RAM má velikost 16 kB. Z celé řady funkčních bloků se nejvíce využívá 12-bitový A/D převodník pro měření analogového napětí. Maximální napětí, které může být na vstup A/D převodníku přivedeno je 3,3 V. Napětí změřené A/D převodníkem $U_{A/D}$ [V] se spočítá pomocí vztahu

$$U_{A/D} = ADC_{sample} \cdot \frac{3,3}{2^{12}}, \quad (2.1)$$

kde ADC_{sample} [-] je celé číslo získané z A/D převodníku. Dále se používají časovače TPM1 a TPM2 pro měření rychlosti a generování PWM signálu. Obvodové schéma zapojení mikrokontroléru je na obr. A.1 v Příloze A.

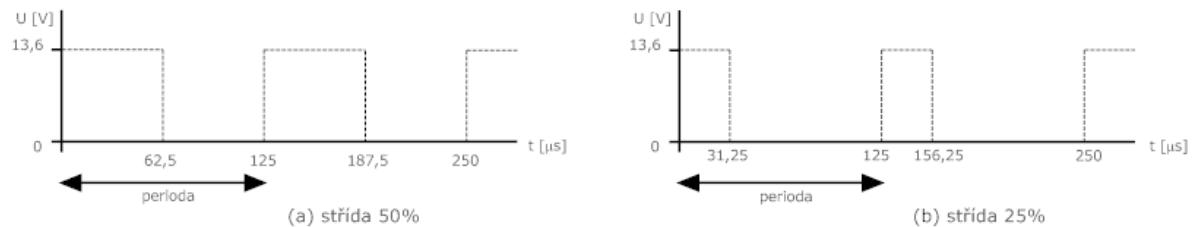
2.3 Integrovaný H-můstek MC33931

Číslicové řízení stejnosměrného motoru pomocí mikrokontroléru je možné díky integrovanému H-můstku, který umožňuje 4-kvadrantové řízení. Pro jízdu na autodráze se používá řízení motoru pouze v jednom směru (řízení v jednom kvadrantu), protože jízda

vzad se neuvažuje. Na H-můstek je přivedeno napětí 13,6 V. Rychlosť autíčka je řízena PWM signálem, který je generován mikrokontrolérem, na kterém byl použit nultý kanál PWM modulu číslo 2. Tento kanál byl nastaven tak, aby vytvářel signál s frekvencí 8 kHz. Perioda tím pádem činí 125 μ s. Činitel plnění lze programově ovládat v rozsahu 0 (0 V) až 6000 (13,6 V) a tomu odpovídá PWM signál se střídou 0% až 100%. Ukázka průběhu PWM signálu pro dvě různé střídy je na obr. 2.3. Přepočet mezi hodnotou PWM [-] a napětím U [V] je

$$PWM = 6000 \cdot \frac{U}{13,6}. \quad (2.2)$$

Obvodové schéma zapojení H-můstku je na obr. A.1 v Příloze A.



Obrázek 2.3: Ukázka PWM signálu se střídou 50% a 25%

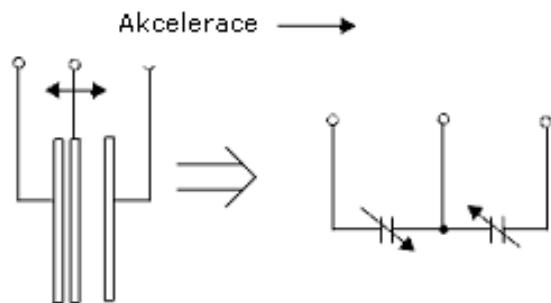
2.4 Senzory

Akcelerometr a měření napětí z dráhy bylo již na dodané desce plošných spojů hardwarově realizováno. Bylo třeba vyřešit pouze softwarovou stránku. Zbylé senzory byly do autíčka přidány a zprovozněny.

2.4.1 Akcelerometr MMA7361L

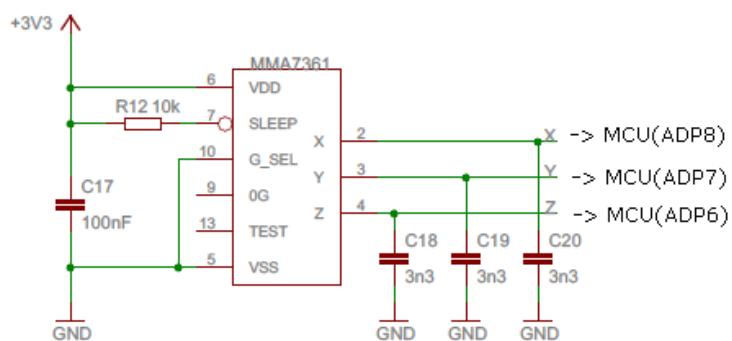
Pomocí tohoto senzoru jsme schopni změřit boční (odstředivé) zrychlení a z naměřených dat určit, zda autíčko projízdělo rovinkou nebo zatáčkou. V naší aplikaci hráje tedy velmi důležitou roli. Akcelerometr na osazené desce je založený na technologii MEMS a umožňuje měřit zrychlení ve všech třech osách. Měření zrychlení funguje na principu měření změny kapacity. Zařízení se skládá z kapacitního snímacího členu, který si můžeme představit jako pohyblivou desku připevněnou k hmotě na pružině, která se může pohybovat mezi dvěma pevnými deskami a tvorí tak dva kondenzátory. Jakmile se prostřední

deska pohybuje se zrychlením, tak se vzdálenost od jedné pevné desky zvětšuje a od druhé zmenšuje. Dochází ke změně vzdálenosti a tedy i ke změně kapacity. Příslušný obvod detekuje změnu kapacity a poskytuje výstupní napětí, které je úměrné zrychlení. Zjednodušené schéma principu měření je na obr. 2.4, který byl převzat z (SCOTT DANIELS, 2012).



Obrázek 2.4: Znázornění principu měření akcelerometru

Na výběr jsou dva režimy citlivosti, $S_H = 800 \text{ mV/g}$ a $S_L = 206 \text{ mV/g}$. Pro naše účely byla zvolena citlivost S_H připojením pinu g-select na zem. Tomu přísluší měřící rozsah $\pm 1,5 \text{ g}$, kde $g = 9,81 \text{ ms}^{-2}$ odpovídá gravitačnímu zrychlení. Akcelerometr ke svému chodu potřebuje napájení 3,3 V. Výstupy akcelerometru jsou přivedeny na kanály 12bit A/D převodníku mikrokontroléru ADP8, ADP7 a ADP6 (v pořadí osa X, Y a Z). Schéma zapojení akcelerometru je na obr. 2.5, který byl převzat z celkového schématu desky vyrobené společností Freescale Semiconductor.



Obrázek 2.5: Schéma zapojení akcelerometru

Časovač je nastaven tak, aby každou 1 ms vyvolal přerušení, během kterého se přečtu údaje z převodníků. Využívá se měření pouze v ose x, které odpovídá bočnímu (odstředivému)

zrychlení. Hodnotu odstředivého zrychlení změřené akcelerometrem dostaneme podle následujícího vztahu

$$a[\text{mm s}^{-2}] = \frac{(U_{A/D} - 1,65)}{S_H} \cdot g \cdot 1000 = 9,8795 \cdot (ADC_{sample} - 2048), \quad (2.3)$$

kde $U_{A/D}$ je napětí změřené A/D převodníkem podle (2.1). V katalogovém listu se lze dočíst, že hodnota 1,65 V odpovídá výstupnímu napětí v klidovém stavu, kdy na akcelerometr nepůsobí žádné zrychlení. Tuto hodnotu odečítáme od změřeného napětí, abychom získané odstředivé zrychlení měli symetrické podle osy x, kdy pravotočivým zatáčkám odpovídá záporné boční zrychlení a levotočivým kladné boční zrychlení. Kvůli driftu akcelerometru, kdy v klidové poloze na akcelerometr nepůsobí žádné zrychlení, se z akcelerometru vyčítá malá hodnota, kterou kompenzujeme odečtením této hodnoty.

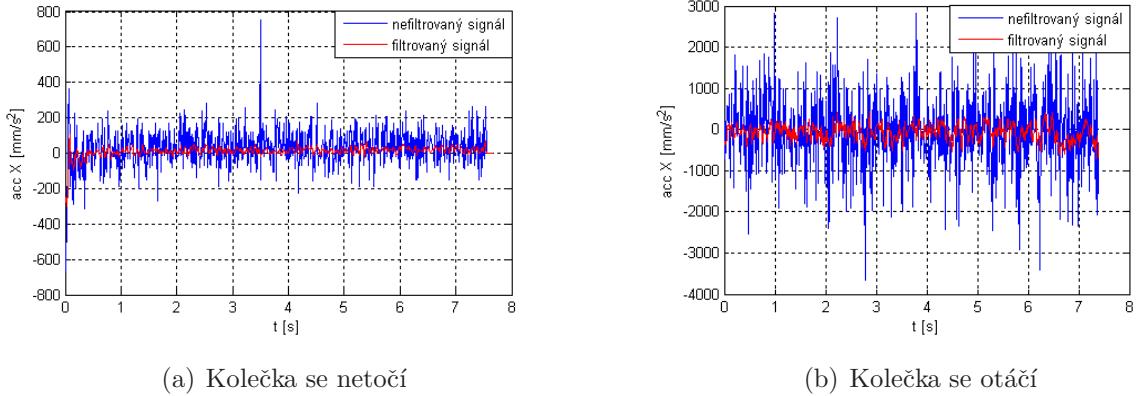
Při prvních měřeních bylo zjištěno, že výstupní signál z akcelerometru je značně zašuměný. Proto bylo potřeba naměřená data softwarově filtrovat. K filtraci byl zvolen IIR filtr typu dolní propust, jehož návrh byl proveden na naměřených datech z jízdy autíčka po oválné dráze. Návrh filtru byl proveden v prosředí Matlab. Diferenční rovnice popisující filtr má následující tvar:

$$y(k) = 0,02452 \cdot x(k) + 0,02452 \cdot x(k-1) - 0,95 \cdot y(k-1). \quad (2.4)$$

Na obr. 2.6(a) můžeme vidět naměřené boční zrychlení v klidové poloze, kdy se kola autíčka neotáčí. Na obr. 2.6(b) je naměřené boční zrychlení v neutrální poloze, kdy se kola otáčí. Otáčení kol způsobuje otřesy, které akcelerometr zachytává. Zde se ukazuje, že je nezbytné signál filtrovat. Na obr. 2.7 můžeme vidět boční zrychlení působící na autíčko při jízdě na oválné dráze. Z naměřených dat lze snadno rozpoznat, kdy se autíčko pohybovalo po rovince a kdy projízdělo zatáčkou. Dále můžeme vypozorovat, že se autíčko po oválné dráze pohybovalo po směru hodinových ručiček na základě záporného bočního zrychlení působící v zatáčkách.

2.4.2 Reflexní senzor QRE1113

Tento senzor byl vybrán vzhledem ke svým malým rozměrům a také proto, že má analogový výstup, který je možné vyčítat pomocí A/D převodníku mikrokontroléru. Obvodové schéma můžeme vidět na obr. 2.8(a), který byl převzat z (SPARKFUN ELECTRONICS, 2012b). Senzor je napájen 3,3 V a skládá se z IR vyzařující LED a fototranzistoru ciltivý na IR záření. V sérii s LED je rezistor o velikosti 100 Ω , který omezuje velikost



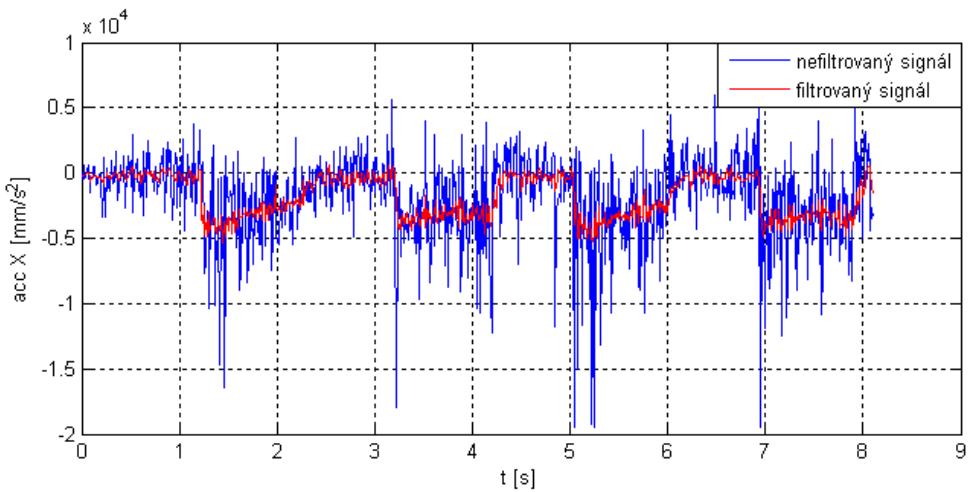
Obrázek 2.6: Naměřené boční zrychlení v klidové poloze při přidržení autíčka

proudů protékajícího touto diodou. K výstupu fototranzistoru je připojen pull-up rezistor o velikosti $10 \text{ k}\Omega$, který udržuje výstupní pin ve stavu "high" (log 1 - přibližně odpovídá napájecímu napětí), pokud se žádné záření neodrazí zpátky na fototranzistor. To nastává v případě černého povrchu. Pokud se ale světlo z LED odrazí zpátky na fototranzistor, tak výstupní napětí začne klesat. Čím více světla je fototranzistorem snímáno, tím nižší bude výstupní napětí. Nejnižší napětí odpovídá bílému povrchu.

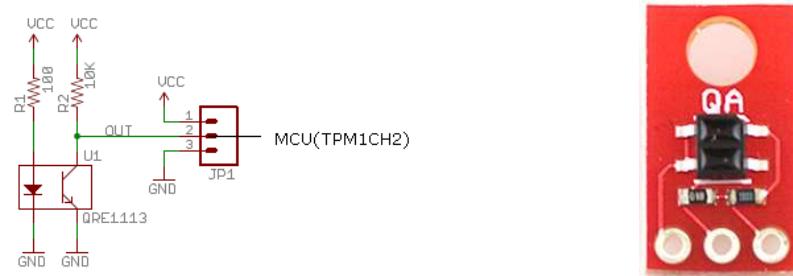
2.4.2.1 Měření rychlosti

Reflexní senzor je umístěn u zadní nápravy proti ozubenému převodovému kolečku, na kterém je nalepena rozeta - černobílý papír tvořící 8 segmentů, 4 bílé a 4 černé (obr. 2.9(a)). Jedná se tedy o variantu klasického rotačního enkodéru (obr. 2.9(b)), pomocí něhož můžeme měřit rychlosť. Během jednoho otočení kola jsou na výstupu fototranzistoru generovány 4 náběžné a 4 spádové hrany. Výstup fototranzistoru je připojen na kanál 2 modulu TPM1, který je nastaven do režimu "input-capture" pro zachytávání náběžných a spádových hran. To nám umožňuje detektovat rozhraní segmentů a měřit čas od předešlého detekovaného rozhraní. Měření rychlosti probíhá tak, že se měří čas mezi příchodem dvou stejných hran, tzn. mezi rozhraními černá-bílá a mezi bílá-černá. Poté, když známe poloměr kolečka $r = 9 \text{ mm}$, časovač běží na frekvenci $f = 3 \text{ MHz}$ a víme, že se rozeta skládá ze 4 bílých a 4 černých segmentů, tak jsme schopni spočítat rychlosť autíčka podle vztahu

$$v[\text{mm s}^{-1}] = \frac{2\pi r f}{4C_f} \cdot 1000 = \frac{42390000}{C_f}, \quad (2.5)$$



Obrázek 2.7: Naměřené boční zrychlení při jízdě na oválné dráze



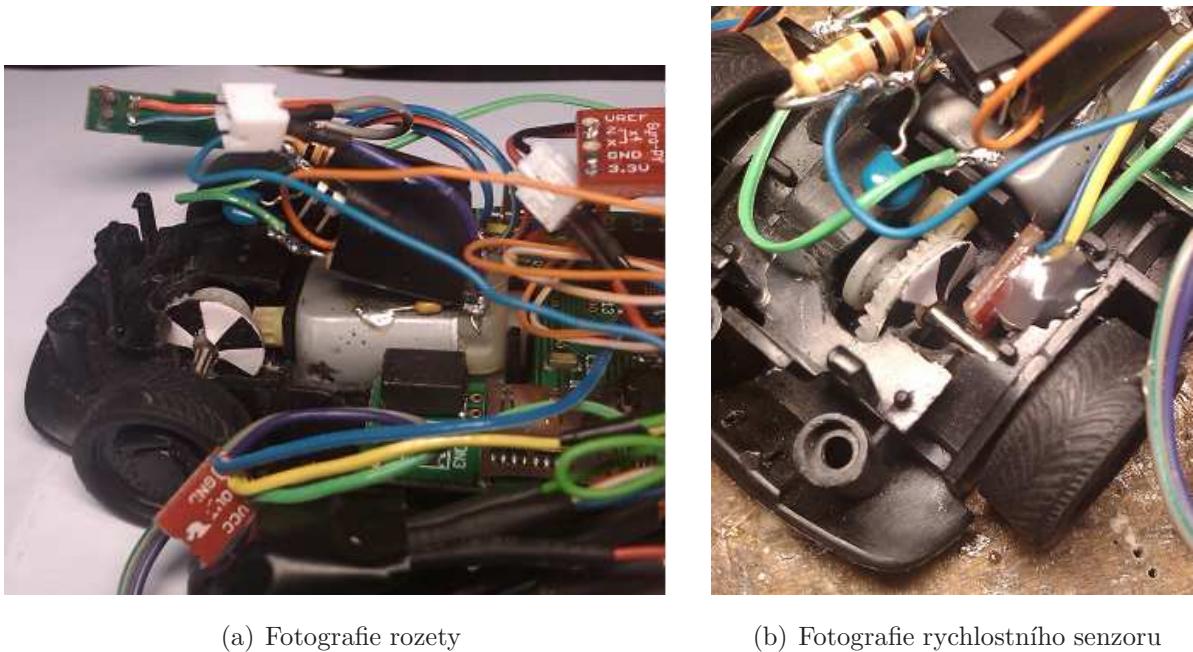
(a) Obvodové schéma reflexního senzoru

(b) Zvětšený obrázek senzoru

Obrázek 2.8: Reflexní senzor QRE1113

kde C_f [-] je počet pulsů od časovače mezi dvěma stejnými hranami. Rychlosť se měří v milimetrech za sekundu, aby se s ní mohlo počítat v pevné řádové čárce s rozumnou přesností. V plovoucí řádové čárce již procesor nestihá počítat potřebné operace. Pro minimalizaci chyb měření bylo implementováno omezení, které dovoluje maximální změnu rychlosti od předešlého měření o 200 mm s^{-1} .

Tento způsob měření rychlosti ovšem přináší i některé nevýhody. První nevýhodou je proměnná vzorkovací frekvence, která závisí na rychlosti, jakou se kola otáčí. To znamená, že čím pomaleji auto jede, tím pomaleji se rychlosť měří. Při rychlosti $v = 1000 \text{ m s}^{-1}$ je průměrná doba mezi každým měřením 8 ms. Další problém způsobuje nepřesné nalepení černobílého papíru na ozubený převod, což způsobuje, že výstupní signál je značně zašuměný. Signál proto musíme filtrovat. Nejlepších výsledků bylo dosaženo s filtrem



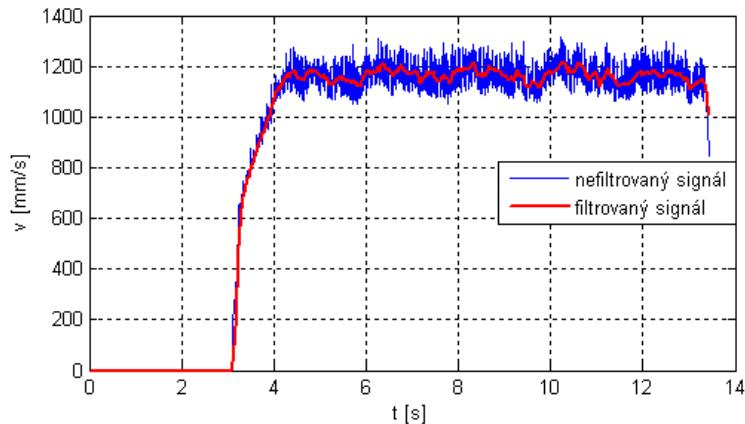
Obrázek 2.9: Realizace rychlostního senzoru

klouzavý průměr s délkou okna $N = 8$. Použití tohoto filtru ovšem zavádí do signálu určité zpoždění. Ukázka jak vypadá měřená rychlosť před a po filtraci je na obr. 2.10. V čase $t = 3$ s je na motor přivedeno napětí 6,8 V ($PWM = 3000$), autičko se rozjede a pohybuje se po kruhové dráze.

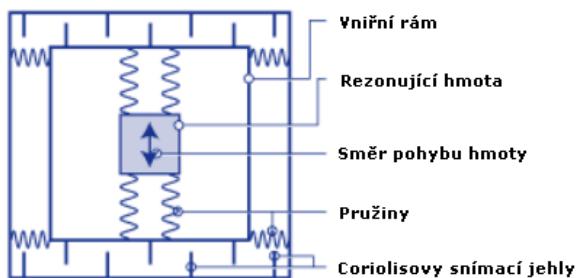
2.4.3 Gyro senzor LY530ALH

Gyro senzor je senzor, který se používá k měření úhlové rychlosti, která souvisí s rotačním pohybem. Existuje mnoho typů gyro senzorů, které se používají v širokém okruhu aplikací, přes použití v letadlech, závodních autech až po digitální fotoaparáty, smartphony nebo tablety. Námi vybraný senzor je založený, stejně jako akcelerometr, na technologii MEMS. MEMS gyro senzory využívají tzv. vibrující strukturu. Ta je tvořena vibrující hmotou, která je soustavou pružin připevněna k pouzdru tvořící vnitřní rám. Tento vnitřní rám je spojený s pevně umístěnou obvodovou deskou druhou soustavou pružin, která je kolmá na tu předešlou. Pro názornější představu máme k dispozici obr. 2.11, který byl převzat z (SENSORWIKI.ORG, 2012).

Hmota uvnitř pouzdra vibruje - pohybuje se ve směru první soustavy pružin. Jakákoli rotace systému způsobí, že bude hmota tlačena ve směru druhé soustavy pružin. Jak se



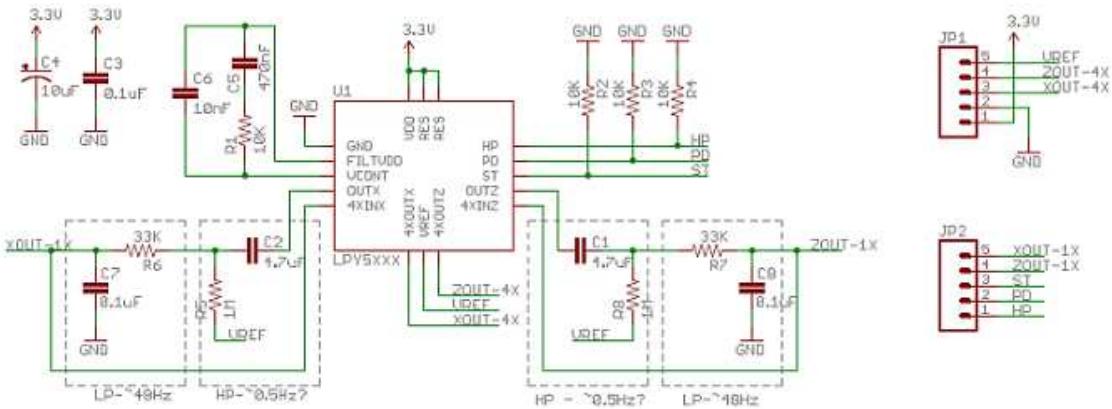
Obrázek 2.10: Porovnání filtrovaného a nefiltrovaného naměřeného signálu rychlosti



Obrázek 2.11: Vnitřní stavba MEMS gyro senzoru

hmota pohybuje od osy rotace, tak je tlačena kolmo v jednom směru a když se pohybuje zpět k ose otáčení, tak je tlačena do opačného směru a to vše v důsledku působení Coriolisovy síly. Coriolisova síla je detekována kapacitními snímacími jehlami, které jsou umístěny podél vnitřního rámu a vnější pevné konstrukce. Jak je hmota působením Coriolisovy síly tlačena, tak se snímací jehly k sobě přiblížují a vzniká kapacitní rozdíl. Když je hmota tlačena v opačném směru, dochází k přiblžení odlišné soustavy snímacích jehel. Senzor je tak schopen detektovat velikost a směr úhlové rychlosti.

Gyro senzor LY530ALH se nachází na zakoupené desce společně s dalšími elektronickými prvky. Na obr. 2.13(b), který byl převzat z (SPARKFUN ELECTRONICS, 2012a), můžeme vidět zmínovanou desku. Podle značení by se mohlo zdát, že je senzor dvouosý. Avšak je to klam. Ve skutečnosti se opravdu jedná o jednoosý gyro senzor. Obvodové schéma zapojení gyro senzoru na desce je na obr. 2.12, který byl převzat z (SPARKFUN ELEKTRONICS, 2013).



Obrázek 2.12: Schema zapojení desky s gyro senzorem

Výstupem senzoru je analogové napětí, které lze vyčítat pomocí A/D převodníku mikroprocesoru. Požadované napětí pro správnou činnost je stejné jako u předchozích dvou senzorů 3,3 V. K dispozici jsou dva režimy citlivosti $SoA = 3,33 \text{ mV}^\circ/\text{s}$ a $So = 0,83 \text{ mV}^\circ/\text{s}$. Nejprve byla zvolena vyšší citlivost SoA . Vybrané citlivosti odpovídá měřící rozsah $\pm 300^\circ/\text{s}$. Abychom dostali nějaká rozumná data, tak bylo třeba jet velmi nízkou rychlostí. Při trochu vyšších rychlostech se gyro senzor dostal do jakési saturace a naměřený signál byl velmi rozkmitaný. Proto byla citlivost změněna na So s měřicím rozsahem $\pm 1200^\circ/\text{s}$. S tímto rozsahem se už gyro senzor do saturace nedostával. Gyro senzor LY530ALH je schopen měřit úhlovou rychlost při otáčení okolo jedné osy, kolem osy kolmé na senzor - osa z. V ustálené pozici, když je úhlová rychlosť nulová, se na výstupu naměří hodnota $U_{out} = 1,23 \text{ V}$. V okamžiku, když senzor koná otáčivý pohyb, tak pro výstupní napětí platí

$$U_{out} = 1,23 + So \cdot \Omega, \quad (2.6)$$

kde $So = 0,83 \text{ mV}^\circ/\text{s}$ je námi zvolená citlivost a $\Omega[\text{°}/\text{s}]$ je úhlová rychlosť. Z rovnice (2.6) lze snadno určit, jaký bude vztah pro úhlovou rychlosť

$$\Omega[\text{°}/\text{s}] = \frac{U_{A/D} - 1,23}{So} \approx 1,2 \cdot (U_{A/D} \cdot 1000 - 1230), \quad (2.7)$$

kde $U_{A/D}$ je napětí změřené A/D převodníkem podle (2.1). Otáčení proti směru hodinových ručiček odpovídá kladné úhlové rychlosťi a v opačném směru záporné. Na obr. 2.13(a), který byl použit z (SPARKFUN ELEKTRONICS, 2012), je znázorněn směr citlivosti pro detekování kladné úhlové rychlosťi.

Do autíčka byly umístěny kolmo na sebe 2 gyro senzory. První z nich je ve vodorovné poloze rovnoběžně s dráhou a jeho výstup je přiveden na kanál ADP0 A/D převodníku.



(a) Směr detekování kladné úhlové rychlosti

(b) Zvětšený obrázek desky s gyro senzorem

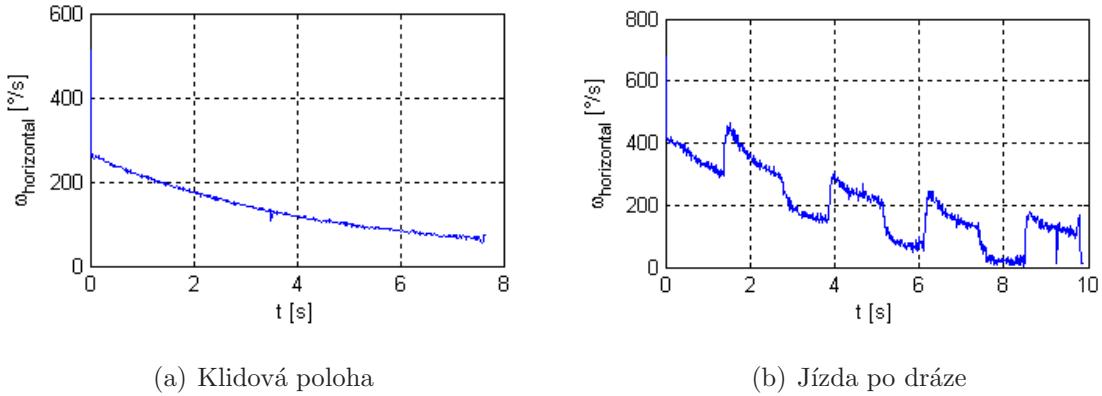
Obrázek 2.13: Senzor úhlové rychlosti

Předpokládalo se, že toto umístění senzoru bude sloužit k detekci případného smyku. S původním měřícím rozsahem SoA a při nízkých rychlostech se ukázalo, že gyro senzor bude dobrý pro obyčejnou detekci zatáčky. Naměřených průběh nebyl téměř zašuměný. Proto byla zvolena citlivost So s větším měřícím rozsahem. Už při prvních měřeních se však ukázalo, že naměřená data nejsou zcela v pořádku. V klidové poloze nebyla vyčítána konstantní hodnota, jak můžeme vidět na obr. 2.14(a) a při jízdě na oválné dráze se naměřená data pro stejné úseky lišily, jak můžeme vidět na obr. 2.14(b). Po hledání na stránkách výrobce bylo zjištěno, že se na zakoupené desce s gyro senzorem nachází nefunkční filtr horní propust, který způsobuje zmíněné problémy. Druhý gyro senzor je v kolmé poloze ke dráze, tedy i k prvnímu gyro senzoru. Jeho výstup je připojen na kanál ADP1 A/D převodníku. Stejným způsobem jako u akcelerometru je každou 1 ms vyvoláno přerušení, během kterého jsou přečteny údaje z převodníků.

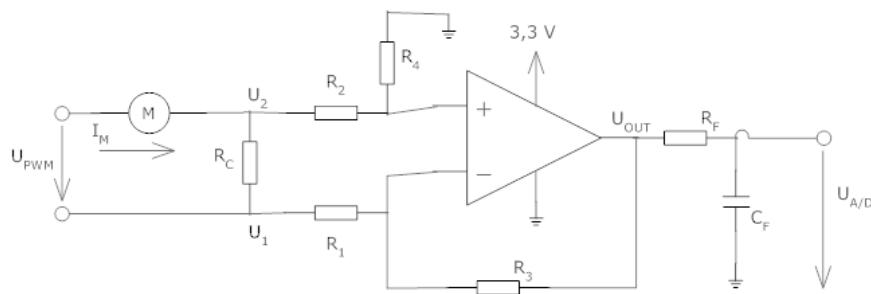
2.4.4 Zapojení s diferenčním zesilovačem

Měření proudu bylo původně realizováno na tzv. Feedback pinu H-můstku, který poskytuje 0,24% výstupního proudu tekoucího do motoru. Na pomocném rezistoru s odporem 270 Ω se A/D převodníkem změřilo napětí, které se poté přepočítalo na proud. Přímo na desce je již implementován analogový filtr prvního řádu typu dolní propust se zlomovou frekvencí 590 Hz, kterým se odfiltrují pulsy z H-můstku. Tento způsob měření však dával špatné výsledky.

Bylo tedy navrženo a realizováno zapojení podle obr. 2.15, kdy byl použit operační zesilovač typu LM358 s napájením 3,3 V. Princip měření proudu s tímto zapojením spočívá



Obrázek 2.14: Naměřené průběhy z gyro senzoru s nefunkčním filtrem horní propust



Obrázek 2.15: Schéma zapojení pro měření proudu

v tom, že do série s motorem byl zapojen měřící odpor R_c , na němž se pomocí diferenčního zesilovače měří napětí. Měřící odpor je tvořen paralelním spojení dvou rezistorů, kdy každý z rezistorů má odpor $220 \text{ m}\Omega$ a maximálním příkonem 1 W . Nízká hodnota odporu rezistorů byla zvolena proto, aby na nich docházelo k malému úbytku napětí. Předpokládáme, že motorem může protékat proud maximálně o velikosti přibližně $2,7 \text{ A}$. Z toho důvodu není použit jeden rezistor, který by se mohl zničit, ale jsou použity rezistory 2 zapojené paralelně, aby při případném maximálním proudu byl příkon na každém z nich do 1 W , což je splněno. Výsledkem paralelního spojení rezistorů je odpor o celkové hodnotě $R_c = 110 \text{ m}\Omega$, ke kterému je zapojen diferenční zesilovač. Jestliže má diferenční zesilovač opravdu zesilovat jen rozdílové napětí, tak musí být dodržena následující podmínka:

$$\frac{R_1}{R_2} = \frac{R_3}{R_4}. \quad (2.8)$$

To znamená, že dvojice odporů R_4 a R_2 musí být ve stejném poměru jako R_3 a R_1 .

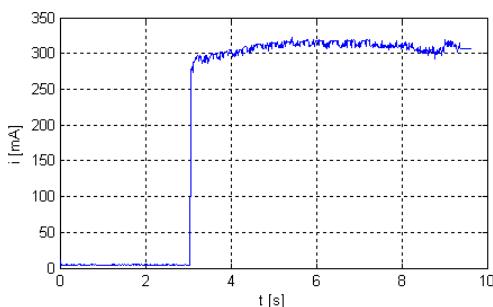
Pro výstupní napětí potom platí

$$U_{OUT} = \frac{R_3}{R_1} \cdot (U_2 - U_1), \quad (2.9)$$

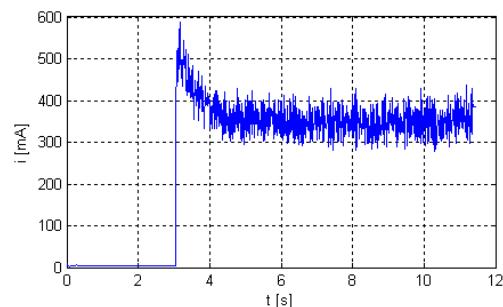
kde poměr $\frac{R_3}{R_1}$ je zesílení a $U_2 - U_1$ je napětí na měřícím odporu. Maximální úbytek napětí, ke kterému může na měřícím odporu docházet, je přibližně 0,3 V. Na vstupu A/D převodníku může být napětí maximálně o velikosti 3,3 V. Abychom se do tohoto limitu vešli, bylo zvoleno desetinásobné zesílení. Pro dosažení zvoleného zesílení byly vybrány následující hodnoty součástek $R_3 = 10 \text{ k}\Omega$ a $R_1 = 1 \text{ k}\Omega$. Zároveň musí platit $R_1 = R_2$ a $R_3 = R_4$. Výstup ze zesilovače byl opět filtrován analogovým filtrem prvního řádu typu dolní propust se zlomovou frekvencí 590 Hz. Hodnoty součástek použité pro analogový filtr jsou $R_F = 270 \Omega$ a $C_F = 1 \mu\text{F}$. Po filtraci je signál přiveden na kanál ADP2 A/D převodníku mikroprocesoru, kterým se změří napětí. To se poté přepočítá na proud podle vzorce

$$I[0, 1mA] = \frac{\frac{U_{A/D}}{10}}{R_c} \cdot 10000 = ADC_{sample} \cdot 7,3. \quad (2.10)$$

Proud se přepočítává na desetiny miliampér, aby se s ním mohlo počítat v pevné řádové čárce s rozumnou přesností. V plovoucí řádové čárce již procesor nestihá počítat potřebné operace. Ukázka naměřeného průběhu proudu s přidrženými kolečky je na obr. 2.16(a) a při jízdě na dráze na obr. 2.16(b). V obou případech je v čase $t = 3 \text{ s}$ je na motor přivedeno napětí. V prvním případě to je 3,17 V ($PWM = 1400$) a v druhém to je 6,8 V ($PWM = 3000$).



(a) Přidržená kolečka



(b) Jízda po dráze

Obrázek 2.16: Ukázka naměřeného průběhu proudu

2.4.5 Odometrie

Autíčko používá nejjednodušší způsob lokalizace a tím je odometrie. Na základě měření ujeté vzdálenosti a detekce křížení jsme schopni určit, kde se autíčko na dráze nachází. Pomocí reflexního senzoru, který se nachází u zadní nápravy proti ozubenému převodovému kolečku s nalepenou rozetou, můžeme měřit nejen rychlosť ale také ujetou vzdálenost. V přerušení, které se stará o měření rychlosti, se počítají detekovaná rozhraní, kterých je celkem 8. Podle počtu detekovaných rozhraní se pomocí vztahu (2.11) spočítá uražená vzdálenost s podle vztahu

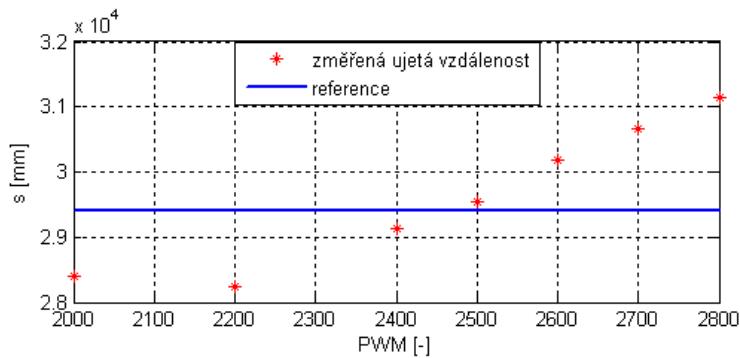
$$s[\text{mm}] = \frac{2\pi rn}{8}, \quad (2.11)$$

kde $r = 9 \text{ mm}$ je poloměr ozubeného převodového kolečka s nalepenou rozetou a n je počet detekovaných rozhraní.

Provedli jsme experimentální měření, jehož cílem bylo zjistit, zda naměřená ujetá vzdálenost bude totožná pro různé rychlosti. Předpokládali jsme, že tomu tak bude. Ovšem po vyhodnocení měření se ukázalo něco jiného. Měření probíhalo na oválné dráze, jejíž délka byla změřena krejčovským metrem a dávala tak správnou referenční hodnotu. Změřená délka jednoho kola byla $l_1 = 2940 \text{ mm}$ a tedy 10 kol má délku $l_{10} = 29400 \text{ mm}$. Měření bylo provedeno pro deset kol a pro kolo jedno. V tabulka 2.1 můžeme vidět naměřené hodnoty pro 10 kol a v tabulka 2.2 pro 1 kolo. V prvním sloupci jsou nastavované hodnoty činetele plnění PWM signálu, ve druhém sloupci můžeme vidět naměřenou ujetou vzdálenost a ve sloupci třetím se nachází odchylky od vzorové hodnoty určené podle vztahu $\Delta_{10} = |29400 - s_{10}|$ pro 10 kol a $\Delta_1 = |2940 - s_1|$ pro kolo jedno.

Tabulka 2.1: Měření ujeté vzdálenosti na 10 kol ($l_{10} = 29400 \text{ mm}$)

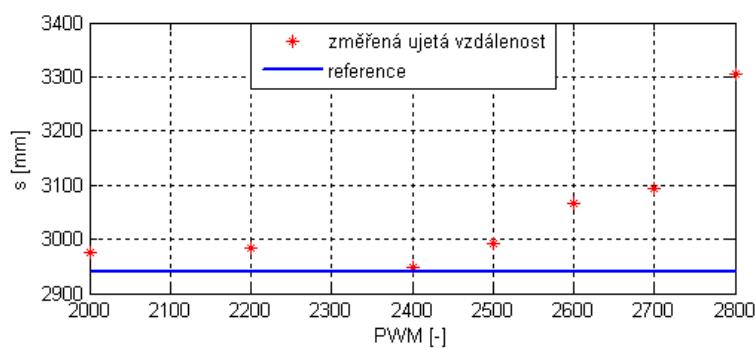
PWM_{10} [-]	s_{10} [mm]	Δ_{10} [mm]
2000	28410	990
2200	28230	1170
2400	29140	260
2500	29540	140
2600	30180	780
2700	30660	1260
2800	31140	1740



Obrázek 2.17: Porovnání naměřených dat s referencí pro 10 kol jízdy

Tabulka 2.2: Měření ujeté vzdálenosti na 1 kolo ($l_1 = 2940$ mm)

PWM ₁ [-]	s ₁ [mm]	Δ ₁ [mm]
2000	2975	35
2200	2985	45
2400	2949	9
2500	2993	53
2600	3065	125
2700	3094	154
2800	3305	365

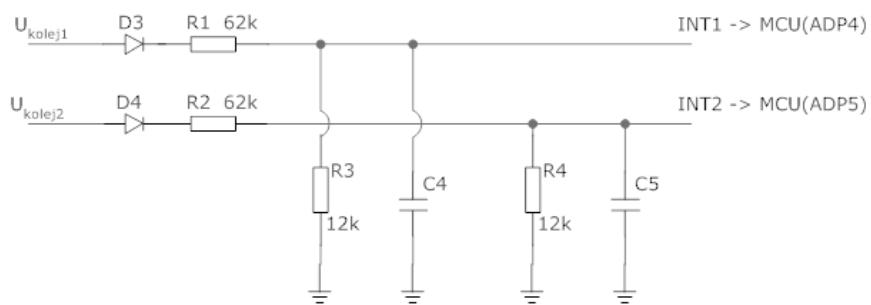


Obrázek 2.18: Porovnání naměřených dat s referencí pro 1 kolo jízdy

Při jízdě na 10 kol je patrné, že naměřené hodnoty se od referenční liší více než pro kolo jedno. Více se projeví chyby v měření způsobené při nižších rychlostech je pravděpodobně způsobené nedetekováním některých rozhraní u rozety. Při vyšších rychlostech většinou dochází na některých segmentech dráhy k prokluzu a tedy také k nepřesnému měření.

2.4.6 Detektor křížení

Dodaná deska plošných spojů obsahovala zapojení umožňující měřit napětí na stíracích kartáčích autíčka (obr. 2.19), což se dá využít pro detekci křížení na dráze. Při přejezdu přes křížení stírací kartáče ztratí na krátký okamžik kontakt s kolejnicemi dráhy, do kterých je přivedeno napájecí napětí. Protože jsme schopni měřit napětí z dráhy, tak naměříme i pokles napětí při přejezdu přes křížení a jsme tedy schopni ho detektovat.

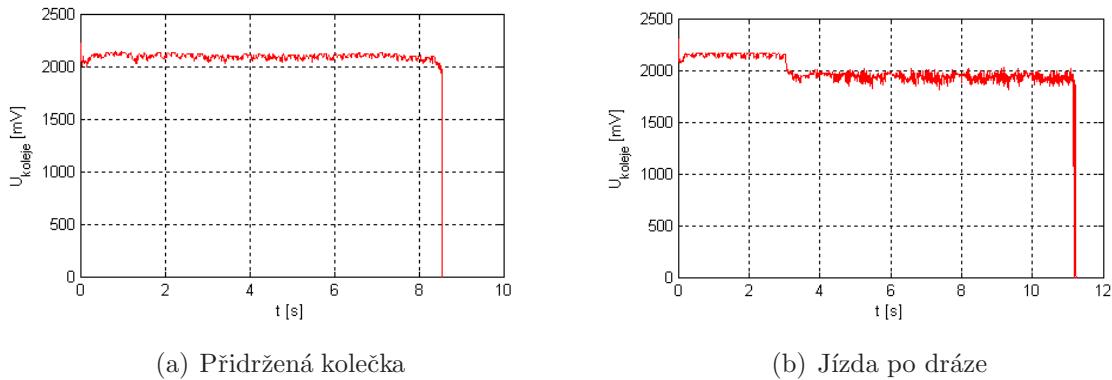


Obrázek 2.19: Schéma zapojení pro měření napětí z kolejnic dráhy

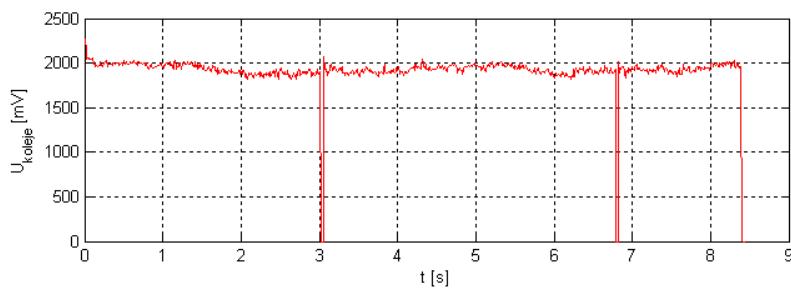
Napětí na stíracích kartáčích 15 V je po průchodu diodou zmenšeno děličem napětí s dělícím poměrem 0,16 na následující hodnotu

$$U_{koleje} = (15 - 0,7) \cdot 0,16 \approx 2,3V, \quad (2.12)$$

které se vejde do měřicího rozsahu A/D převodníku mikrokontroléru. Napětí z kolejky 1 je přivedeno na kanál ADP4 a napětí z kolejky 2 na kanál ADP5 A/D převodníku mikrokontroléru. Na obr. 2.20(a) můžeme vidět změřené napětí z dráhy, když je autíčko v klidovém stavu s přidrženými kolečky. Na obr. 2.20(b) je vidět rozdíl mezi změřeným napětím z kolejnic dráhy v klidovém stavu a při jízdě na dráze. V čase $t = 3$ s se autíčko rozjede. Zvýší se tak odběr proudu, které autíčko potřebuje pro svůj pohyb a můžeme tedy vidět pokles napětí. Dále si můžeme všimnout, že pokud se autíčko pohybuje po dráze, tak je průběh napětí více zarušený. To způsobují stírací kartáče, které jsou na koncích roztržené. Jsou v nedokonalém kontaktu s kolejnicemi dráhy a navíc při jízdě je velikost styčných ploch proměnlivá. Na obr. 2.21 je zaznamenán průběh napětí z kolejnic dráhy při přejezdu přes křížení. Podle zaznamenaných je vidět, že autíčko přejelo křížení dvakrát. Pokles napětí na konci měřených dat u všech třech měření je způsoben odebráním autíčka z dráhy.



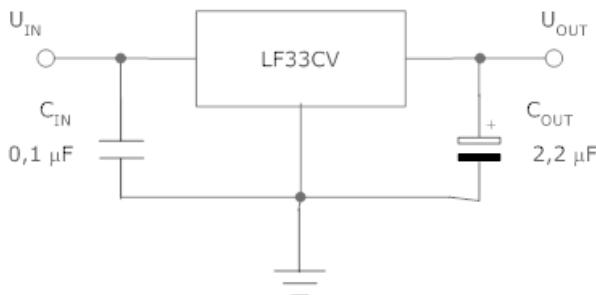
Obrázek 2.20: Naměřené průběhy napětí z kolejnic dráhy



Obrázek 2.21: Detekování krížení na dráze

2.5 Regulátor napětí LF33CV

K dodané elektronice byl dále přidán regulátor napětí LF33CV, který převádí napětí za dvoucestným usměrňovačem z hodnoty 13,6 V, které je příliš vysoké pro napájení senzorů a další elektroniky, na hodnotu 3,3 V. Snížené napětí slouží k napájení reflexního senzoru, dvou gyro senzorů a operačního zesilovače. Zapojení bylo provedeno podle doporučení v katalogovém listu, kdy mezi vstup regulátoru napětí a zem byl zapojen kondenzátor $C_{IN} = 0,1 \mu F$ a mezi výstup a zem elektrolytický kondenzátor $C_{OUT} = 2,2 \mu F$ obr. 2.22. Tento externí regulátor napětí byl přidán kvůli obavám z přetížení interního regulátoru napětí LP2950-D, který je umístěn na dodané desce plošných spojů a stará se například o napájení mikroprocesoru nebo akcelerometru. Více informací o tomto regulátoru napětí je možné získat zde (ON SEMICONDUCTOR, 2012).



Obrázek 2.22: Zapojení regulátoru podle doporučení z katalogového listu

2.6 Mini USB port a programování mikrokontroléru

Mikrokontrolér je možné propojit s počítačem prostřednictvím mini USB portu. V paměti mikrokontroléru se nachází předprogramovaná funkce USB Bootloader, která umožňuje nahrát aplikaci do flash paměti. Podmínkou spuštění Bootloaderu je napájení z USB. Při připojení mikrokontroléru pomocí USB kabelu se provede reset, inicializuje se Bootloader a počítač rozpozná velkokapacitní paměťové zařízení a přidá nový disk. Na něm se nachází prázdný soubor READY.TXT. Z naší aplikace se zkopíruje soubor .S19, který byl vytvořen v rámci komplilace. Bootloader tento soubor přečte a zapíše do flash paměti. Po odpojení USB kabelu a přiložení napájecího napětí z kolejnic dráhy, které musí být alespoň 12 V, se opět provede reset a autíčko přechází do aktivního režimu, ve kterém se spustí nahraná aplikace. Při nahrání nového souboru se ten předešlý přepíše.

Programování se provádí v jazyce C ve vývojovém prostředí Freescale CodeWarrior for Microcontrollers Special Suite, které je možné získat zdarma s omezením velikosti kódu na 64 kB. Přesně tolik má dodaný mikrokontrolér. Vycházelo se z projektu, který je dostupný na (hw.cz, 2011).

2.7 μSD karta

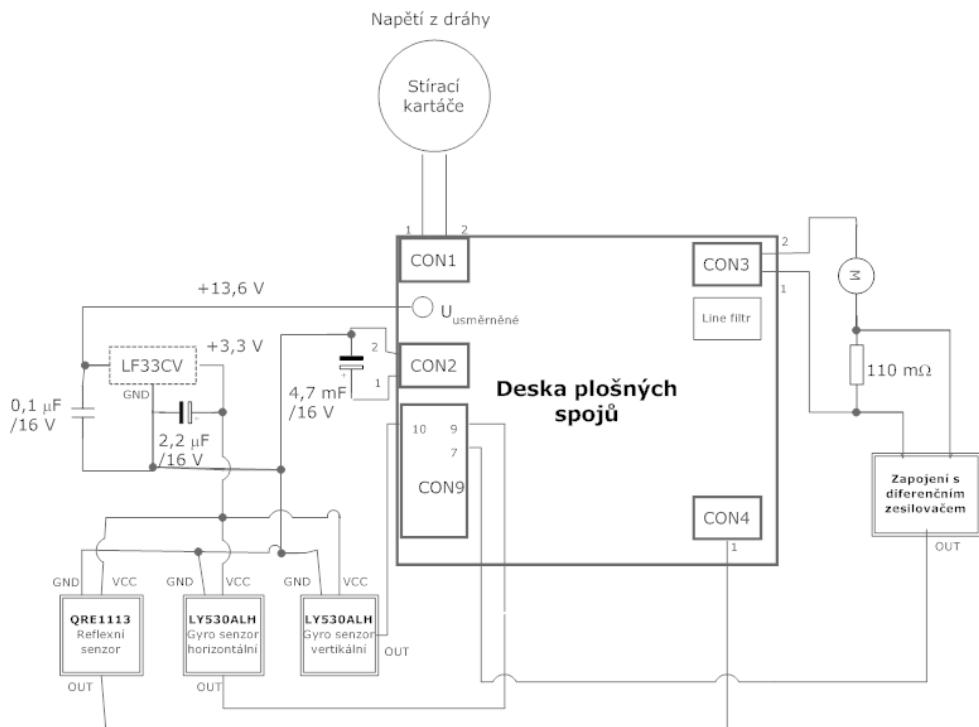
Aby bylo možné analyzovat průběh jízdy, tak je zapotřebí naměřená data někam ukládat. K tomu zde slouží μSD karta, která se umísťuje do slotu, který je napájen na dodané desce. Výhodou tohoto přenosného datového média jsou velmi malé rozměry a možnost komunikace po SPI sběrnici, která umožňuje synchronní sériovou komunikaci mezi mikrokontrolérem a μSD kartou. Jako master vystupuje mikrokontrolér, který generuje ho-

dinový signál. V pozici slave je μ SD karta.

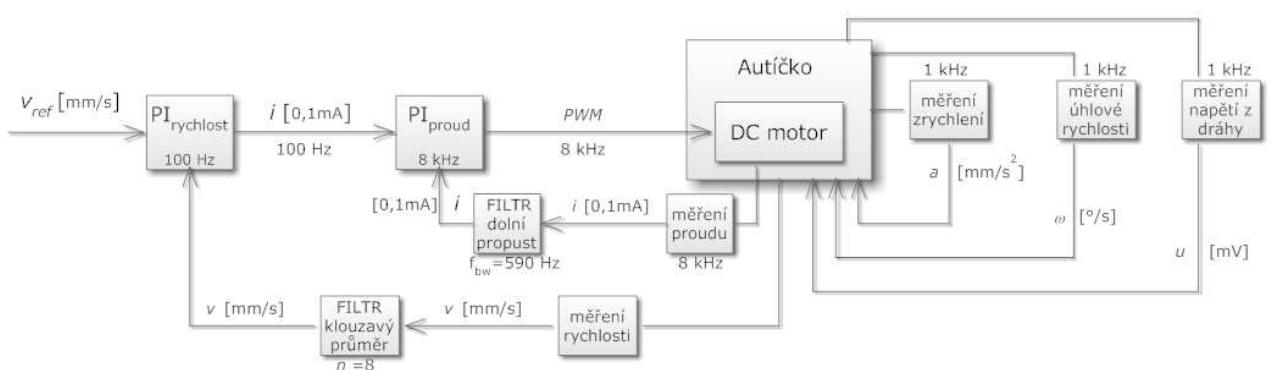
Řadič μ SD karty podporuje SD/SDHC a FAT/FAT32. Pro komunikaci byl použit modul FatFs generického FAT souborového systému, který je volně dostupný na (CHAN, 2012). Tento modul je napsaný v jazyce ANSI C a je kompletně oddělený od diskové I/O vrstvy. Je tedy nezávislý na použité hardwarové architektuře. Podporovány jsou soubory do velikosti 4 GB. Z řady funkcí, které tento modul nabízí, byly použity f_mount pro registraci diskového prostoru s nímž se má pracovat, f_open k vytvoření souboru, f_printf pro zapsání formátovaného stringu, f_close k zavření souboru, f_sync pro fyzický zápis dat na kartu.

2.8 Celkové schéma

V této sekci můžeme přehledně vidět celkové schéma elektronických úprav (obr. 2.23) a celkové měřící schéma (obr. 2.24).



Obrázek 2.23: Výsledné schéma elektronických úprav



Obrázek 2.24: Celkové měřící schéma

Kapitola 3

Matematický model

Autíčko pohybující se po autodráze si z hlediska dynamiky můžeme představit jako hmotný bod vykonávající posuvný pohyb. Autíčko pohání třípolový kartáčový stejnosměrný motor s permanentními magnety napájený napěťovým zdrojem. Vodící díl, který je připevněný k podvozku autíčka v přední části mezi koly, tvoří spojení s dráhou a umožňuje pohyb v přímém směru bez možnosti zatáčení předních kol. Může však dojít ke smyku zadních kol, na kterých je umístěný pohon. Do smyku se autíčko snadno dostane v zatačkách, kdy na něj působí odstředivá síla.

3.1 Identifikace systému

Jak již bylo zmíněno, model automobilu pohání stejnosměrný motor. Na základě 2. Kirchhoffova napěťového zákona a 2. Newtonova pohybového zákona jsme schopni stejnosměrný motor s cizím buzením napájený napěťovým zdrojem popsat dobře známými rovnicemi

$$L \frac{di(t)}{dt} = -Ri(t) - k_e \omega(t) + u(t), \quad (3.1)$$

$$J_{motor} \frac{d\omega(t)}{dt} = k_t i(t) - b_d \omega(t) - b_s sign(\omega(t)), \quad (3.2)$$

kde $u(t)$ [V] je vstupní napětí, $i(t)$ [A] je proud procházející vinutím kotvy rotoru, R [Ω] je odporník vinutí kotvy rotoru, L [H] je indukčnost vinutí kotvy rotoru, $\omega(t)$ [rad s^{-1}] je úhlová rychlosť hřídele motoru, J_{motor} [kg m^2] je moment setrvačnosti hřídele motoru bez zatížení, b_d [N m] je koeficient dynamického tření, b_s [N m] je koeficient statického

tření, k_e [$\text{V s}^{-1} \text{ rad}^{-1}$] je elektrická konstanta motoru a k_t [N m A^{-1}] je mechanická konstanta motoru.

Rychlosť pohybu modelu automobilu získáme pomocí vztahu

$$v(t) = \omega(t)rG_B, \quad (3.3)$$

kde r [m] je poloměr kola autíčka a G_B [-] je převodový poměr z motoru na kola autíčka. Výsledný moment setrvačnosti můžeme vyjádřit pomocí rovnice

$$J = J_{motor} + J_{auto} = J_{motor} + mr^2G_B^2, \quad (3.4)$$

kde m [kg] je hmotnost autíčka. Protože $J_{motor} \ll J_{auto}$ můžeme vztah (3.4) zjednodušit na $J \approx J_{auto}$. S využitím (3.3), při použití celkového momentu setrvačnosti J a zanedbáním statického tření b_s dostaneme rovnice pro model autíčka

$$L \frac{di(t)}{dt} = -Ri(t) - \frac{k_e}{rG_B} v(t) + u(t), \quad (3.5)$$

$$\frac{J}{rG_B} \frac{dv(t)}{dt} = k_t i(t) - \frac{b_d}{rG_B} v(t), \quad (3.6)$$

kde nám místo úhlové rychlosti otáčení hřídele $\omega(t)$ figuruje rychlosť $v(t)$, kterou se autíčko pohybuje vpřed. Využitím Laplaceovy transformace a následnými úpravami dostaneme rovnice do podoby

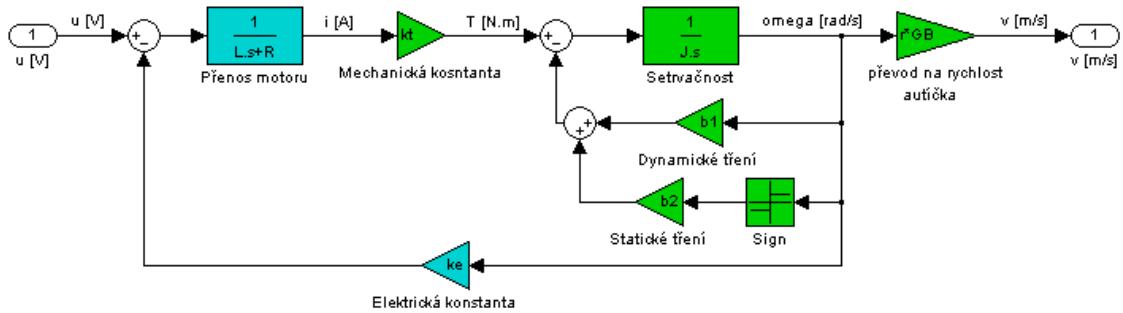
$$I(s) = \frac{-\frac{k_e}{rG_B}}{Ls + R} V(s) + \frac{1}{Ls + R} U(s), \quad (3.7)$$

$$V(s) = \frac{rG_B k_t}{Js + b_d} I(s). \quad (3.8)$$

Při dosazení rovnice (3.7) do (3.8) a následných úpravách můžeme určit přenos systému, kdy vstupem je napětí na motoru $u(t)$ a výstupem je rychlosť autíčka $v(t)$

$$G_{u \rightarrow v}(s) = \frac{V(s)}{U(s)} = \frac{rG_B k_t}{LJs^2 + (JR + b_d L)s + b_d R + k_e k_t}. \quad (3.9)$$

Simulinkové schéma modelu autíčka sestavené na základě rovnic (3.1) a (3.2) můžeme vidět na obr. 3.1.



Obrázek 3.1: Simulinkové blokové schéma modelu autíčka

3.2 Identifikace konstant

Po vytvoření matematického modelu autíčka bylo potřeba identifikovat hodnoty příslušných parametrů. Jejich hodnoty jsme určili buď výpočtem nebo měřením.

Odpor R a indukčnost L vinutí kotvy rotoru byly změřeny speciálním přístrojem, který umí měřit odpory, indukčnosti a kapacity na různých frekvencích. Hmotnost autíčka m byla určena pomocí laboratorní váhy. Poloměr kola r byl změřen pravítkem. Převodový poměr G_B jsme určili pozorováním, kdy jsme zjistili, že se motor otáčí třikrát rychleji než kola. Výsledný moment setrvačnosti J byl získán prostým dosazením do (3.4), kdy při předpokladu $J_{motor} \ll J_{auto}$ hodnotu J_{motor} zanedbáváme.

3.2.1 Identifikace mechanické k_t a elektrické k_e konstanty

Mechanická a elektrická konstanta jsou hodnotově ekvivalentní $k_t = k_e$. Provedli jsme měření k získání mechanické konstanty. Jakmile známe mechanickou konstantu, známe i elektrickou. Pro snadnější měření bylo potřeba vymontovat motorek z autíčka. Na jeho hřídelku bylo připevněno rameno ze stavebnice Merkur, jehož hmotnost a délku jsme znali a umístili jsme ho tak, aby volným koncem leželo na laboratorní váze s vysokou citlivostí pro rychlé zvážení. Motorek jsme připojili ke zdroji napětí, který jsme zatím nezapínali a nastavili na 2 V. Dále jsme připojili multimeter pro měření proudu tekoucího do motoru. Na krátkou dobu jsme zapnuli napájení motoru, který připevněným ramenem začal tlačit na váhu do zastavení. Odečetli jsme zobrazenou hmotnost a hodnotu proudu naměřenou multimetrem. Z naměřené hmotnosti jsme po odečtení hmotnosti ramene schopni určit sílu, kterou motor prostřednictvím ramene na váhu působí

$$F = mg, \quad (3.10)$$

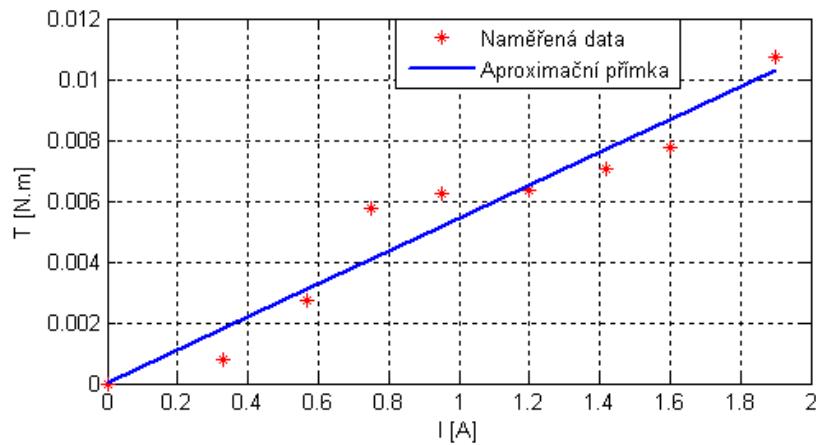
kde m [kg] je získaná hmotnost a $g = 9,81 \text{ m s}^{-2}$. Když známe sílu, kterou motor tlačí na váhu a délku ramene l [m], tak jsme schopni určit točivý moment motoru podle vztahu

$$T = Fl. \quad (3.11)$$

Hodnotu napětí jsme po krocích zvyšovali a v každém kroku z multimetru odečetli hodnotu proudu tekoucího do motoru a podle (3.10), (3.11) určili velikost točivého momentu. Točivý moment je s proudem spjat přes mechanickou konstantu motoru

$$T = k_t i. \quad (3.12)$$

Naměřená data jsme proložili přímkou podle obr. 3.2. Směrnice přímky odpovídá hledané mechanické konstantě.



Obrázek 3.2: Závislost točivého momentu na proudu

Tímto experimentem jsme také schopni určit hodnotu odporu vinutí kotvy rotoru R . Jelikož je motor při experimentu zastavený, tak zde nehráje roli indukované napětí. Napětí na motoru odpovídá napájecímu napětí. Protože známe hodnotu napájecího napětí nastavenou v jednotlivých krocích a měříme proud, tak jsme schopni dopočítat odpor podle vztahu

$$R = \frac{U}{I}. \quad (3.13)$$

3.2.2 Identifikace tření

Protože se autíčko pohybuje v těsném kontaktu s dráhou, vzniká jev, který se nazývá tření. V této práci tření modelujeme jako kombinaci Coulombova (statického) tření a

viskózního (dynamického) tření. Třecí točivý moment můžeme vyjádřit

$$T_f = b_d\omega + b_s, \quad (3.14)$$

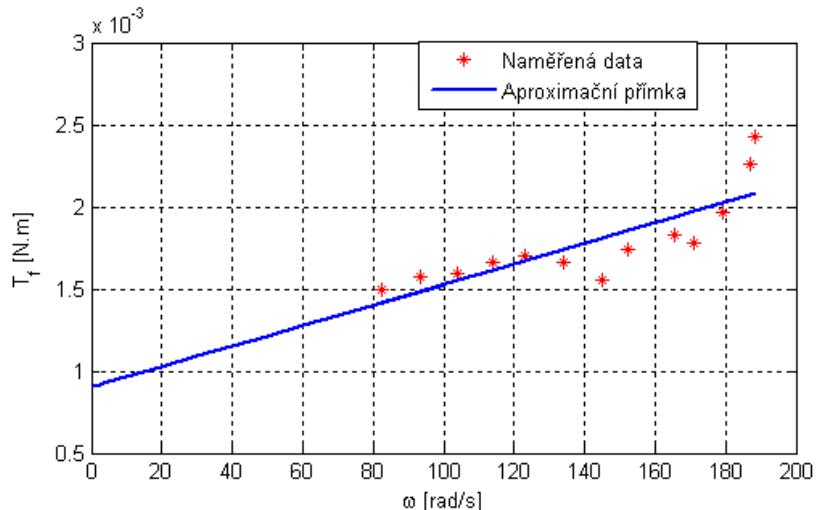
kde první člen reprezentuje dynamické tření a druhý statické. Měření probíhalo tak, že jsme na dráhu přivedli konstantní napětí a na ni umístili autíčko. Měřili jsme proud tekoucí motorem a rychlosť autíčka. Z naměřených dat jsme určili ustálenou hodnotu rychlosti a proudu. To jsme provedli pro různé hodnoty napětí na dráze. Rychlosť autíčka v jsme přepočítali na úhlovou rychlosť otáčení kol ω podle vztahu

$$\omega = \frac{v}{r}, \quad (3.15)$$

kde r je poloměr kola. Točivý moment vyvolaný třením se určí podle vztahu

$$T_f = k_t i. \quad (3.16)$$

Naměřená data jsme proložili přímkou, jak můžeme vidět na obr. 3.3 a určili jsme hodnoty koeficientů b_d a b_s .



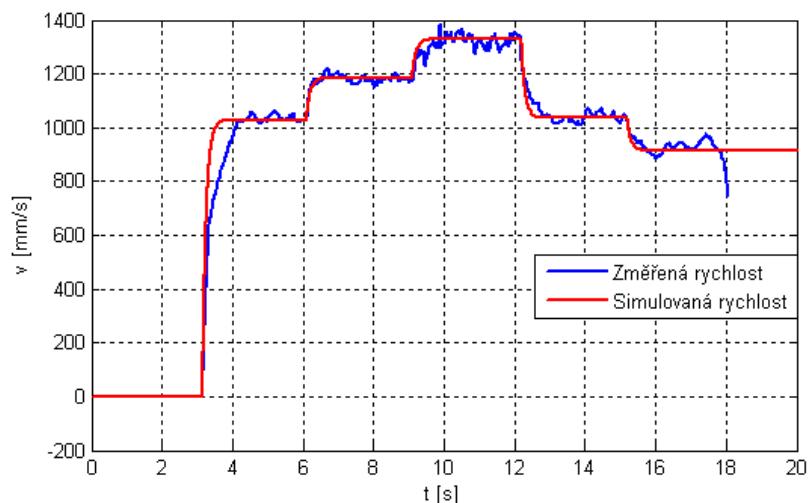
Obrázek 3.3: Závislost třecího momentu na úhlové rychlosti

V tabulce 3.1 jsou shrnuty hodnoty všech parametrů vyskytujících se v našem modelu.

Tabulka 3.1: Identifikované parametry matematického modelu autíčka

Parametr	Hodnota	Jednotky SI
R	5,14	Ω
L	1,38	mH
J	$1,2444 \cdot 10^{-6}$	kg m^2
k_t	0,0054	N m A^{-1}
k_e	0,0054	$\text{V s}^{-1} \text{rad}^{-1}$
b_d	$6,2615 \cdot 10^{-6}$	$\text{rad s}^{-1} \text{N m}$
b_s	$9,024 \cdot 10^{-4}$	N m
r	0,01	m
m	0,112	kg
G_B	$\frac{1}{3}$	-

Porovnání nelineárního modelu (3.1), (3.2) s naměřenou rychlosí na reálném autíčku pro skokové změny napětí je na obr. 3.4.

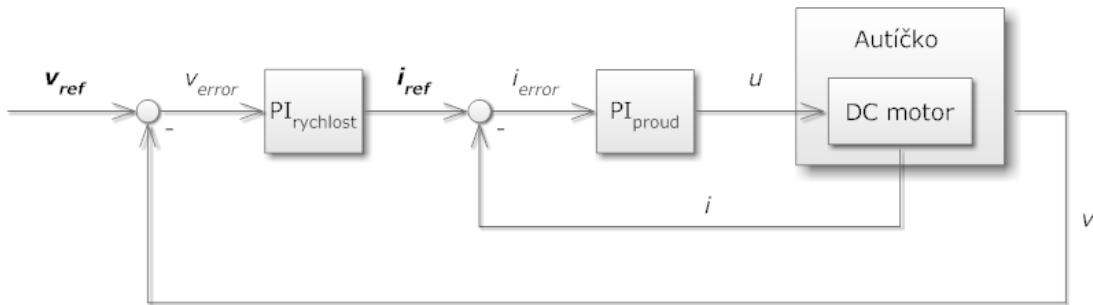


Obrázek 3.4: Porovnání naměřené a simulované rychlosti modelem autíčka

3.3 Návrh řízení

Pro řízení autíčka bylo zvoleno kaskádní uspořádání dvou PI regulátorů zobrazené na obr. 3.5. Detailnější zobrazení kaskádní řídící struktury, které se snaží co nejvíce přiblížit reálnému autíčku, je na obr. A.2 v Příloze A. Regulovanými smyčkami jsou proudová a rychlostní. Rychlostní regulátor ve vnější smyčce určuje referenci pro proudový regulátor ve smyčce vnitřní. Výhodou kaskádní řídící struktury je to, že rušení, které postihuje veličinu ve vnější smyčce, je regulováno rychleji a řízení je efektivnější.

Nejprve bylo navrženo řízení pro vnitřní smyčku a poté pro vnější. Cílem bylo navrhnut co nejlepší regulátory ve smyslu, aby doba náběhu na referenci byla co nejkratší a přitom byly dodrženy požadavky na zvolenou šířku pásma uzavřené smyčky. V prostředí Matlab byly navrženy spojité regulátory pro proudovou i rychlostní smyčku, které byly na sestaveném matematickém modelu autíčka vyzkoušeny. Spojité regulátory byly poté diskretizovány a implementovány do mikrokontroléru v reálném autíčku.



Obrázek 3.5: Kaskádní řídící struktura tvořená proudovou a rychlostní smyčkou

3.3.1 Návrh spojitých regulátorů

Při návrhu spojitého proudového regulátoru byla snaha, aby šířka pásma amplitudové frekvenční charakteristiky uzavřené smyčky byla co největší. Pro proudovou smyčku se zpravidla volí šířka pásma 1 kHz. Potřebovali jsme ale také dodržet, aby zvolená šířka pásma byla přibližně o dekádu nižší, než je vzorkovací frekvence, aby regulátor byl schopný dostatečně rychle reagovat na změnu pokynu. Vzorkovací frekvence proudového regulátoru je 8 kHz. Šířka pásma 1 kHz je tedy vyhovující. Přenosová funkce pro proudový

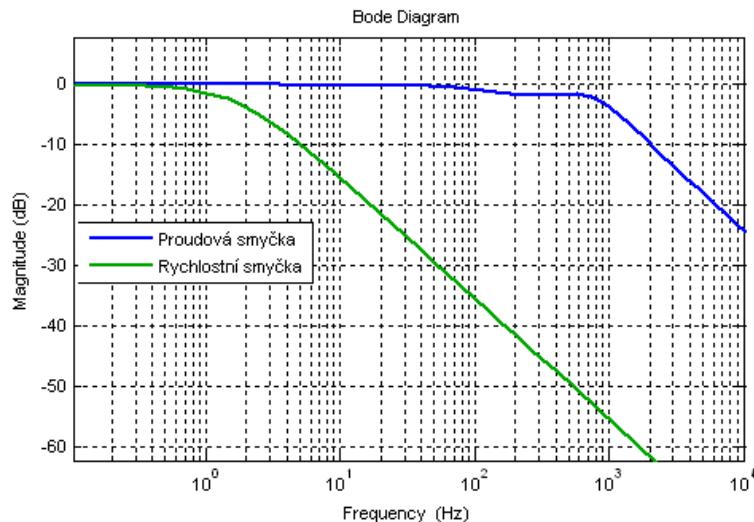
regulátor je

$$PI_{proud} = \frac{5,5s + 7500}{s}. \quad (3.17)$$

Při návrhu kaskádní řídící struktury musí být dodrženo, že vnitřní smyčka je rychlejší než vnější. V našem případě tedy proudová smyčka musí pracovat rychleji než rychlostní. Chceme, aby šířka pásma rychlostní smyčky byla menší než u proudové smyčky. Bereme-li v úvahu vzorkovací frekvenci rychlostního regulátoru, která je 100 Hz, tak požadovaná šířka pásma byla zvolena 10 Hz. Přenosová funkce rychlostního regulátoru má následující podobu

$$PI_{rychlost} = \frac{5s + 40}{s}. \quad (3.18)$$

Na obr. 3.6 můžeme vidět amplitudovou frekvenční charakteristiku proudové a rychlostní smyčky s navrženými PI regulátory. Požadavky na stanovené šířky pásma proudové a rychlostní smyčky byly splněny. Celkový přenos systému z referenční rychlosti na výstupní



Obrázek 3.6: Amplitudová frekvenční charakteristika řízených smyček

je následující

$$T_{v_{ref} \rightarrow v} = \frac{288237,6 \cdot (s + 1364) \cdot (s + 8)}{(s + 6885) \cdot (s + 749,4) \cdot (s + 72,63) \cdot (s + 8,391)}. \quad (3.19)$$

Jedná se o stabilní přenos, a tedy o stabilní systém. Můžeme vidět, že všechny póly mají zápornou reálnou část a leží tak v levé komplexní polorovině.

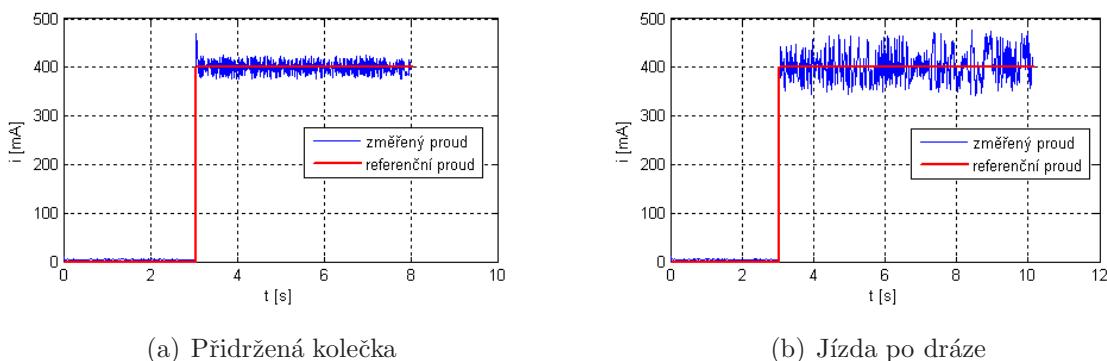
3.3.2 Implementace regulátorů

Spojité regulátory byly diskretizovány a implementovány do mikrokontroléru autíčka v jazyce C. Implementovaná diskrétní podoba jak proudového, tak rychlostního regulátoru vypadá následovně

$$u(t_n) = K_p \cdot e(t_n) + K_i T \cdot \sum_{k=0}^n e(t_k), \quad (3.20)$$

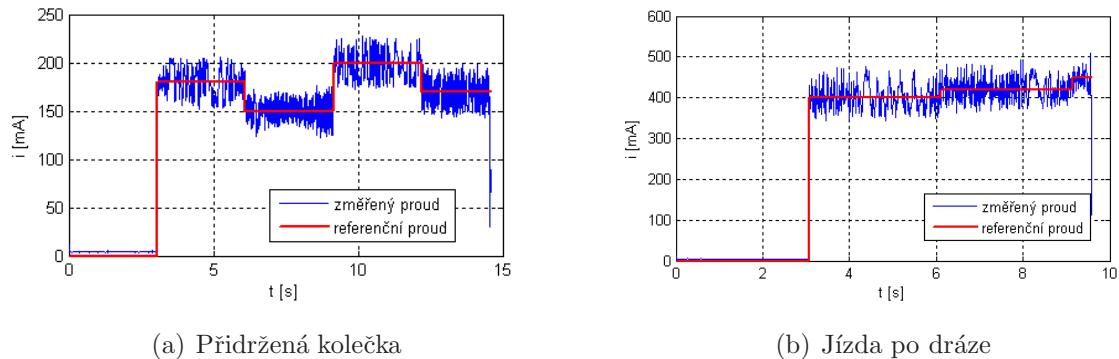
kde $u(t_n)$ je akční zásah v čase $t_n = nT$, kde n je diskrétní časový krok a T je perioda vzorkování. K_p , K_i jsou koeficienty nalezené při návrhu spojitych regulátorů - proporcionální a integrační konstanta. Chyba měření e je vypočtená jako rozdíl mezi naměřenou a referenční hodnotou. Implementované regulátory obsahují antiwindup opatření, neboli opatření pro případ saturace akčního zásahu, které zabrání tzv. unášení integrační složky. V případě proudového regulátoru, pokud dojde k překročení maximální nebo minimální meze akčního zásahu, je nastavena mezní hodnota a odečtena nebo přičtena malá hodnota od součtu chyb měření u integrační složky.

Chování proudové smyčky můžeme vidět na obr. 3.7. Proudový regulátor se chová velmi dobře, pokud jsou kolečka přidržená obr. 3.7(a). Situace se zhorší v okamžiku, kdy se autíčko pohybuje po dráze a kolečka se otáčí obr. 3.7(b). Důvodem, proč je průběh proudu více rozkmitaný, je pravděpodobně proměnlivý kontakt stíracích kartáčů s kolejnicemi dráhy. PI regulátor je schopný pouze částečné kompenzace. Odezva na skokové změny referenčního proudu je na obr. 3.8.

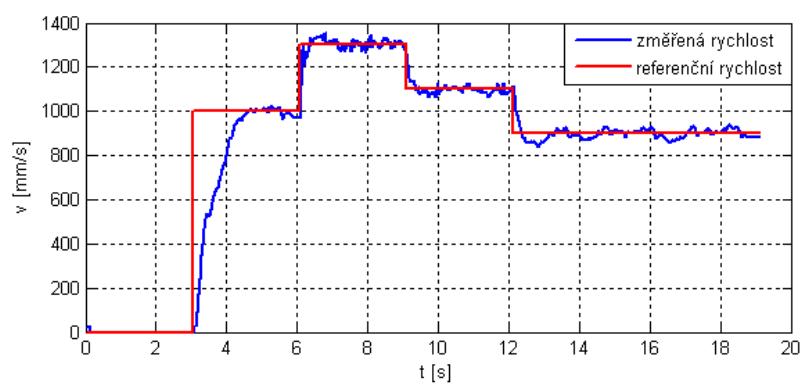


Obrázek 3.7: Odezva na skokovou změnu referenčního proudu

Chování rychlostní smyčky společně s proudovou smyčkou můžeme posoudit na obr. 3.9, kde můžeme vidět odezvu na skokové změny referenční rychlosti.



Obrázek 3.8: Odezva na skokové změny referenčního proudu



Obrázek 3.9: Odezva autíčka na skokové změny referenční rychlosti

Kapitola 4

Freescale Race Challenge

Soutěž byla vyhlášena společností Freescale Semiconductor ČR. Organizaci univerzitních kol měly letos prvním rokem na starosti jednotlivé školy. Na ČVUT se o organizaci staral Ing. Vít Hlinovský, CSc z Katedry elektrických pohonů a trakce FEL ČVUT. Freescale Semiconductor ČR pořádalo závěrečné ”Velké finále” v Rožnově pod Radhoštěm.

Podstatou soutěže bylo postavit a naprogramovat autíčko na autodráhu, které se rychle naučí samo jezdit po neznámé dráze a v co nejkratším čase ji projede.

Pravidla hlavního závodu měla následující podobu (převzato z (HW.CZ, 2011)):

- Každé auto závodí samostatně na čas.
- Měření času začíná prvním průjezdem časomírou a končí po 10 kolech.
- Pokud auto vypadne z dráhy, soutěžící jej smí znova nasadit.
- Závod se jede na 2 jízdy, v pravé a v levé dráze, součet časů obou jízd určuje výsledné pořadí.

Vlastnosti dráhy byly následující:

- Dráha hlavního závodu je účastníkům neznámá až do začátku závodu.
- Dráha se skládá z předepsaných dílů Carrera Evolution (všech nebo podmnožiny).
- Dráha je napájena laboratorním zdrojem 15 V.
- Krajnice a svodidla nejsou použity.

Auta musela splňovat tyto předpisy:

- Musí být použit originální podvozek, karoserie, vodící díl, motor a pneumatiky auta Carrera.
- Přítlačné magnety nejsou povoleny.
- Auto může být vybaveno jakoukoli elektronikou, ale pouze uvnitř karoserie.
- Váha auta nesmí překročit 125 g.
- Na autě smí být pouze jeden přepínač a jeden trimr.

V týmu se mnou byl kolega Ing. Dan Martinec.

4.1 Analýza problému

Zadanou úlohu je možné řešit několika způsoby. Nejjednoduší možností je nastavit takovou rychlosť, která je bezpečná pro všechny segmenty tratě.

Další možností je přímé řízení. Zatáčku si můžeme představit jako rušení působící na autíčko, když jí projízdí, a projevující se ve formě odstředivého zrychlení. Autíčko se musí po dráze pohybovat takovou maximální rychlostí, aby po detekování zatáčky zpomalilo na rychlosť, při které nevypadne z dráhy. Pokud by autíčko do zatáčky vjelo příliš rychle, tak by už nemuselo stihnout včas zpomalit na bezpečnou průjezdovou rychlosť. Informaci o bočním zrychlení totiž autíčko nedostává v předstihu, ale až v okamžiku, kdy se v zatáčce nachází. Žádaná rychlosť může být například generována tak, že nalezneme maximální rychlosť pro rovinu, z které vždy stihneme zpomalit na bezpečnou průjezdovou rychlosť pro nejmenší poloměr zatáčky. V zatáčkách tuto rychlosť pak zmenšíme přímo úměrně absolutní hodnotě zrychlení přenásobené empiricky nalezenou konstantou.

My jsme si pro řešení vybrali metodu, která je založená na predikci dráhy. Protože víme, z jakých segmentů může být dráha sestavena, tak pomocí mapovacího kola lze určit, o který typ segmentu se jedná a podle toho diskrétně upravovat rychlosť pro jednotlivé segmenty dráhy.

V následující kapitole si přiblížíme, jak vypadá průjezd zatáčkou z fyzikálního hlediska.

4.1.1 Fyzikální popis průjezdu zatáčkou

Autíčko se po dráze pohybuje v přímém směru. V okamžiku, kdy projíždí zatáčkou, můžeme pohyb charakterizovat jako pohyb po kružnici s poloměrem odpovídající poloměru zatáčky. Průjezd autíčka zatáčkou je zobrazen na obr. 4.1. Na autíčko v zatáčkách působí odstředivá síla

$$F_o = ma = m \frac{v^2}{R}, \quad (4.1)$$

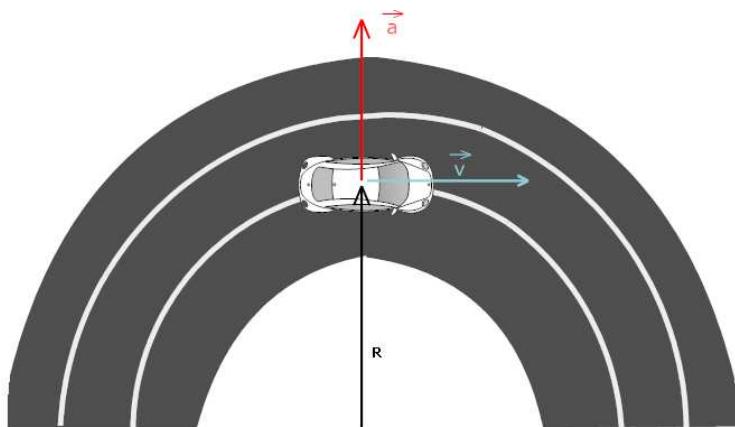
kde m je hmotnost autíčka, a je odstředivé zrychlení, v je rychlosť autíčka a R je poloměr zatáčky. Podmínkou pro udržení se v zatáčce je, aby odstředivá síla byla menší než síla třecí

$$F_t = f F_n, \quad (4.2)$$

kde f je součinitel tření a F_n je svislá tlaková síla (gravitační síla) působící na autíčka. Autíčko se v zatáčkách může pohybovat takovou rychlosťí, při které je tedy splněno

$$F_o < F_t. \quad (4.3)$$

Integrovaný akcelerometr na desce plošných spojů nám umožňuje měřit odstředivé zrychlení a , pomocí kterého můžeme snadno dopočítat odstředivou sílu podle (4.1).



Obrázek 4.1: Znázornění průjezdu zatáčkou

4.1.2 Analýza dráhy

Z pravidel soutěže víme, že se dráha bude skládat z předepsaných dílů Carrera Evolution (všech nebo podmnožiny). Dráha může být sestavena z následujících dílů: 15x standardní

rovinka, 1x napájecí díl, 2x 1/3 rovinka, 2x 1/4 rovinka, 2x křížení drah, 2x zatáčka 1/30, 13x zatáčka 1/60, 6x zatáčka 2/30. Jednotlivé díly dráhy můžeme vidět na obr. 4.2 Měli jsme jistotu, že se na dráze budou vždy vyskytovat: napájecí díl a dvě křížení, kdy autíčko přejíždí z vnitřní dráhy do vnější a opačně.



Obrázek 4.2: Díly dráhy Carrera Evolution

Významné díly potřebující větší analýzu jsou rovinky a křížení, protože těmito segmenty se nedá projet maximální rychlostí. Zatáčka 1/60 tvoří kruhovou výseč o úhlu 60°. Spojením tří kusů zatáček se vytvoří půl kruh a spojením šesti kusů celý kruh. Zatáčka s označením 1/30 je vlastně zatáčka 1/60 rozřízlá na půl. Spojením šesti kusů dostaneme půl kruh a spojením dvanácti kusů celý kruh. Zde je potřeba si všimnout, že obě zatáčky se liší pouze svojí délkou, ale poloměr mají stejný. Tím se nám o polovinu snižuje počet poloměrů zatáček pro klasifikaci. Dále máme k dispozici zatáčku s označením 2/30. Spojením šesti kusů této zatáčky dostaneme půl kruh a spojením dvanácti kusů celý kruh stejně jako u zatáčky 1/30. Avšak zatáčka 2/30 má větší poloměr. Poslední segment, kde by mohlo autíčko vypadnout z dráhy je křížení. Protože máme dráhu vnitřní a vnější, tak máme celkem 5 různých poloměrů zatáček (u křížení jeden poloměr - vnitřní a vnější dráha jsou ekvivalentní) a rovinku, které si označíme následovně - $R_{1/60_1/30_in}$, $R_{1/60_1/30_out}$, $R_{2/30_in}$, $R_{2/30_out}$, $R_{krizeni}$, $R_{rovinka}$. Index "in" je označení pro vnitřní a "out" pro vnější dráhu.

4.2 Předsoutěžní příprava

Pro odvozených pět poloměrů zatáček bylo třeba určit maximální bezpečné průjezdové rychlosti a zjistit jaké jim přísluší boční zrychlení potřebné pro klasifikaci během mapovacího kola. Podle (4.1) vidíme, že odstředivé boční zrychlení závisí nejen na poloměru zatáčky, ale také na rychlosti a to dokonce na kvadrátu. Experimentálně byla určena mapovací rychlosť, při které autíčko projede všemi segmenty dráhy, aniž by vypadlo a také aniž by šlo do smyku. Protože při smyku akcelerometr naměří špičku, která by zkreslovala pozdější klasifikaci. Zjištěná mapovací rychlosť $v_{map} = 1250 \text{ mm/s}$ ($PWM_{map} = 2800$). V tabulce 4.1 můžeme vidět, jaké boční zrychlení odpovídá jakému úseku tratě při průjezdu mapovací rychlostí a jaká je maximální bezpečná průjezdová rychlosť pro daný poloměr zatáčky. Ačkoliv je signál z akcelerometru filtrován, tak přesto je filtrovaný signál celkem rozkmitaný. Proto boční zrychlení, podle kterého se klasifikují jednotlivé segmenty tratě, je v podobě intervalu. A protože zatáčka může být pravotočivá nebo levotočivá, tak jsou to intervaly dva - záporný a kladný.

Tabulka 4.1: Experimentálně zjištěná data pro soutěžní algoritmus

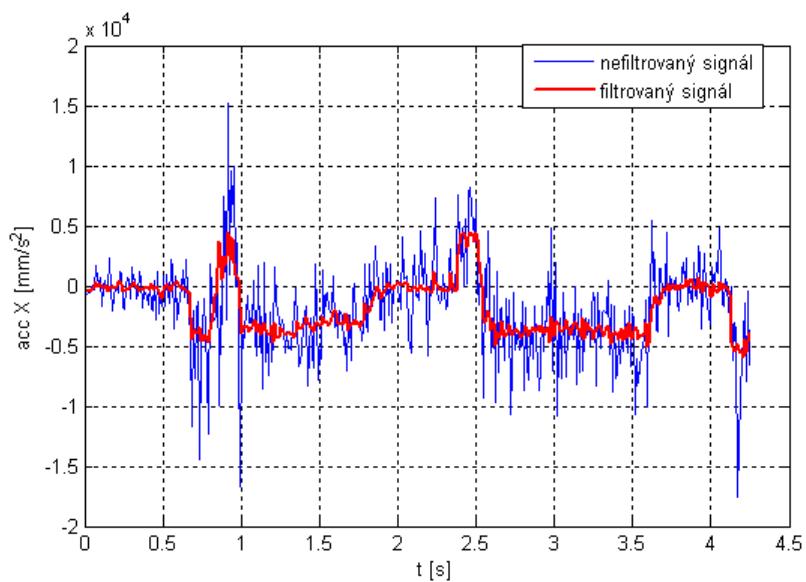
segment tratě	$a_{map} [\text{mm}/\text{s}^2]$	$PWM_{max} [-]$	$v_{max} [\text{mm/s}]$
$R_{1/60_1/30_in}$	$(a_{map} < -4750), (4750 < a_{map})$	3100	1500
$R_{1/60_1/30_out}$	$(-4750 < a_{map} < -4000), (4000 < a_{map} < 4750)$	3200	1600
$R_{krizeni}$	$(-4750 < a_{map} < -4000), (4000 < a_{map} < 4750)$	3200	1600
$R_{2/30_in}$	$(-4000 < a_{map} < -3400), (3400 < a_{map} < 4000)$	3500	2400
$R_{2/30_out}$	$(-3400 < a_{map} < -1500), (1500 < a_{map} < 3400)$	3400	2300
$R_{rovinka}$	$(-1500 < a_{map} < 1500)$	max	max

Maximální bezpečná průjezdová rychlosť pro daný poloměr zatáčky byla zjištěna tak, že byla postupně zvyšována dokud nedošlo k vypadnutí autíčka z dráhy. Stejným způsobem byla určena i maximální bezpečná hodnota PWM .

4.3 Mapování dráhy

Samotný závod začíná až po prvním průjezdu startem. To znamená, že je k dispozici jedno „neměřené“ kolo, kdy si autíčko konstantní mapovací rychlosť projede dráhu a změří boční zrychlení. Na základě naměřených dat autíčko určí o jakou kategorii zatáčky se jedná

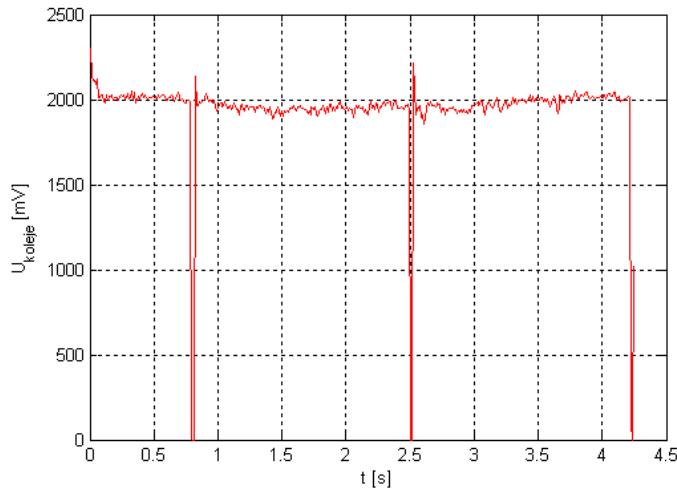
a přiřadí experimentálně určené maximální bezpečné průjezdové rychlosti. Na základě měření ujeté vzdálenosti je možné stanovit délku jednotlivých segmentů tratě a autíčko je tak schopné si vytvořit mapu. Ve vytvořené mapě bylo potřeba provést určitou korekci, aby autíčko nastavilo rychlosť pro danou kategorii zatáčky dříve, než se v té zatáčce objeví. Korekce byla určena experimentálně o hodnotě 100 mm. Naměřená data bočního zrychlení z mapovacího kola pro jednoduchou oválnou dráhu jsou na obr. 4.3. Na obr. 4.4 je znázorněno detekování křížení na dráze.



Obrázek 4.3: Naměřené boční zrychlení během mapovacího kola

4.4 Závodní režim

Aby autíčko poznalo, kdy má už jet ”naostro”, bylo potřeba zjistit, kdy projelo startem a začíná druhé kolo. Protože jsme v autíčku neměli zabudovaný senzor pro detekci startu, tak jsme se mohli spolehnout pouze na možnost detekování křížení na dráze. S jistotou jsme věděli, že na dráze budou 2 křížení. Křížení jsme detekovali na základě poklesu napětí na stíracích kartáčích autíčka. Detekovaná křížení jsme počítali a při třetí detekci křížení přešlo autíčko do závodního režimu a jelo podle vytvořené mapy. Tím se však trochu nabere časová ztráta, protože autíčko po průjezdu startem do druhého kola jelo mapovací rychlostí, dokud nedetekovalo křížení na dráze.

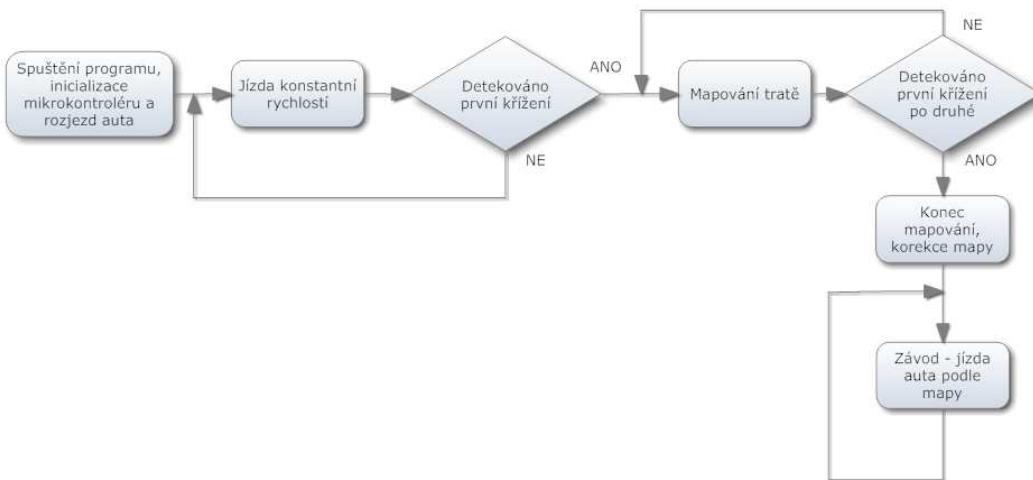


Obrázek 4.4: Detekování křížení na dráze během mapovacího kola

4.5 Program

Tato podkapitola obsahuje popis programu vytvořeného v jazyce C do mikrokontroléru autíčka.

Po umístění na dráhu se autíčko rozjede nastavenou mapovací rychlostí v_{map} (PWM_{map}). Proměnné *mapping* a *racing* jsou nastaveny na nulu. Autíčko jede po dráze a když detektuje první křížení tak proměnnou *mapping* nastaví na jedničku a přejde do mapovacího režimu. V tomto režimu je periodicky volána funkce *classifyAccX()*, která slouží pro klasifikaci jednotlivých segmentů tratě. Kategorie danému segmentu je přiřazena až poté, když naměřená hodnota bočního zrychlení spadá do stejného intervalu alespoň třikrát. Jakmile víme, o jakou kategorii segmentu se jedná, tak se do pole *trackMapVel* uloží maximální bezpečná průjezdová rychlosť pro klasifikovaný segment tratě a do pole *trackMapDist* se na stejnou pozici uloží délka klasifikovaného segmentu tratě. Takto je trať klasifikována do té doby, než je detekováno první křížení po druhé, nebo-li počet detekovaných křížení je roven třem. Při třetí detekci křížení je proměnná *mapping* nastavena na nulu, *racing* je nastaveno na jedničku a autíčko přechází do závodního režimu. Podle hodnot v poli *trackMapVel* nastavuje rychlosť pro klasifikované segmenty tratě. Nastavení rychlosti je vždy s předstihem, než autíčko do daného úseku vjede. Vývojový diagram programu můžeme vidět na obr. 4.5.



Obrázek 4.5: Vývojový diagram programu

4.6 Soutěž - univerzitní kolo

V univerzitním kole jsme obsadili 4. místo z šesti týmů. Postoupili jsme do "Velkého finále" v Rožnově pod Radhoštěm, kterého se však z důvodů nemoci ani jeden z nás nemohl zúčastnit. V období konání soutěže jsme měli problémy s ukládáním na μ SD kartu, a tak z průběhu soutěže nemáme bohužel žádná naměřená data. S postupem času se ukládání podařilo zprovoznit a tak byla sestavena vlastní "soutěžní" dráha a na ní data naměřena.

4.7 Vlastní soutěžní dráha

Jak již bylo řečeno, po zprovoznění ukládání byla sestavena vlastní dráha a při jízdě naměřena data. Soutěžní algoritmus byl vyzkoušen jak pro referenční rychlosti, tak pro referenční hodnoty PWM . Naměřená data byla získána pro referenční hodnoty PWM . V prvním kole se čas neměřil jako v soutěži a od dalšího kola byly spuštěny stopky. Měření bylo ukončeno po deseti kolech. Nejprve byl algoritmus vyzkoušen na klasickém oválu a pak na složitější dráze (obr. 4.6).

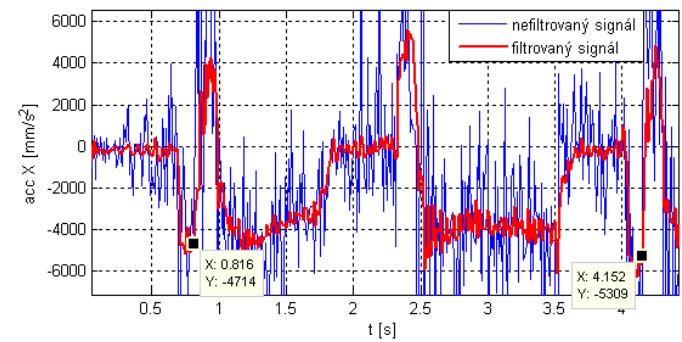
Nejprve se podíváme na oválnou dráhu. Na obr. 4.7(a) můžeme vidět naměřené boční zrychlení z mapovacího kola. Je označeno, kdy mapování začalo a kdy skončilo. Okamžiky detekování křížení ohraničující mapovací fázi můžeme vidět na obr. 4.7(b). Hodnoty PWM nastavované během "ostrého" závodu jsou na obr. 4.8. Jízda se soutěžním algorit-



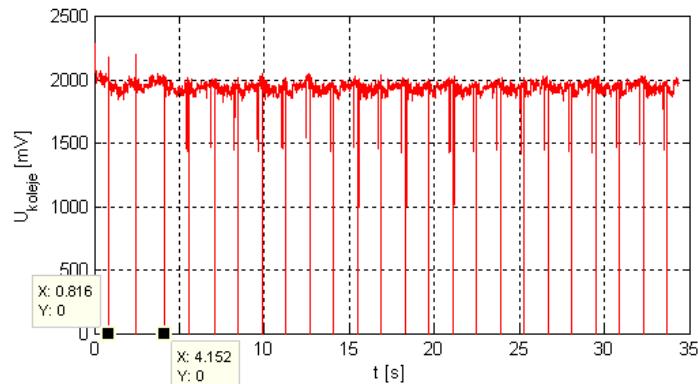
Obrázek 4.6: Fotografie složitější dráhy

mem byla několikrát opakována a průměrná doba projetí deseti kol byla 28,06 s. Pro porovnání s konstantní hodnotou $PWM = 3100$, což je maximální bezpečná průjezdová rychlosť pro nejmenší poloměr zatáčky, je průměrná doba projetí deseti kol 29,4 s. Se soutěžním algoritmem bylo tedy dosaženo lepšího výsledku.

Stejným způsobem byl soutěžní algoritmus otestován na složitější dráze, naměřena data a porovnán s konstantní hodnotou PWM . Získaná data můžeme vidět na obr. 4.9 a obr. 4.10. Průměrný čas projetí deseti kol byl v tomto případě se soutěžním algoritmem 49,51 s. Při konstantní hodnotě $PWM = 3100$ byla průměrná doba projetí deseti kol 49,16 s. Na složitější dráze soutěžní algoritmus před konstantní hodnotou nevyhrál. Těsná porážka byla způsobená častými smyky, ke kterým v případě soutěžního algoritmu docházelo. Pravděpodobně nedošlo ke správné klasifikaci segmentů tratě.

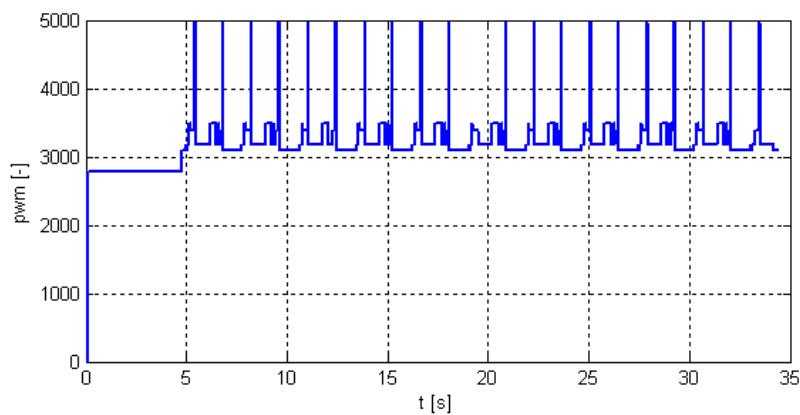


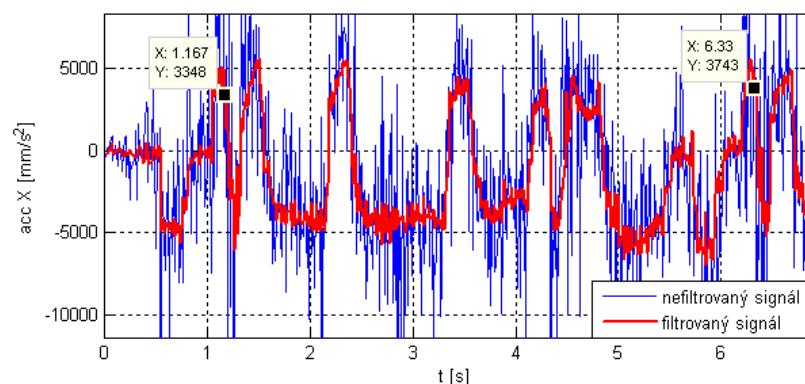
(a) Naměřené boční zrychlení v mapovacím kole



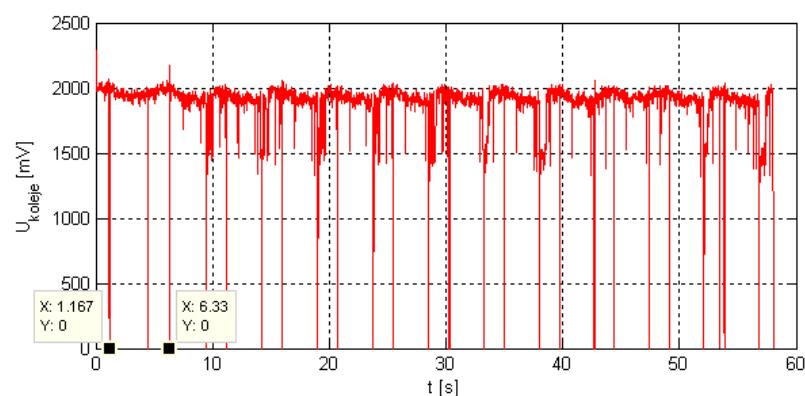
(b) Změřené napětí z dráhy

Obrázek 4.7: Naměřená data na oválné dráze

Obrázek 4.8: Nastavované hodnoty činitele plnění *PWM* pro oválnou dráhu

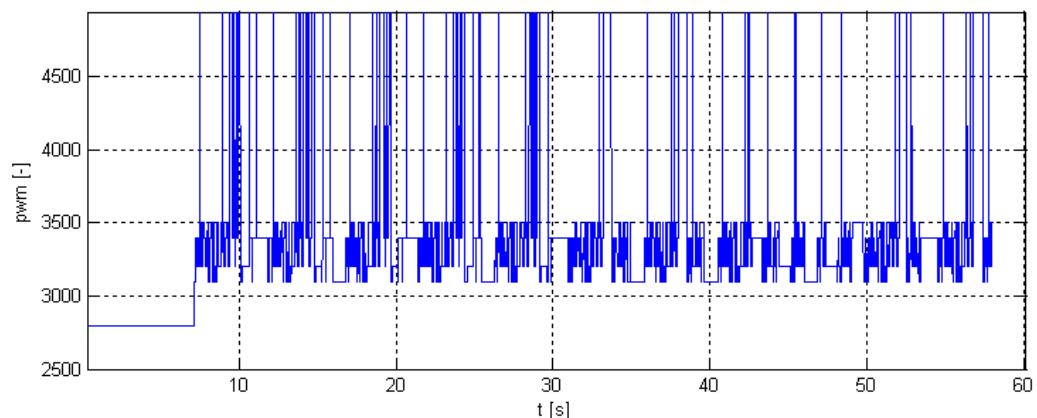


(a) Naměřené boční zrychlení v mapovacím kole



(b) Změřené napětí z dráhy

Obrázek 4.9: Naměřená data na složitější dráze

Obrázek 4.10: Nastavované hodnoty činitele plnění PWM pro složitější dráhu

Kapitola 5

Reinforcement learning

Sekundárním cílem diplomové práce bylo nastudovat nějakou metodu reinforcement learningu a pokusit se teoreticky rozebrat, jak by tato metoda šla aplikovat na naši soutěžní úlohu.

V této kapitole se čtenář seznámí s oblastí strojového učení, která v posledních letech nachází stále širší uplatnění a dostává se do "hledáčku" mnoha výzkumných pracovníků. Reinforcement v překladu znamená zesílení nebo posila. Volně do češtiny můžeme pojmem reinforcement learning přeložit jako zosilující učení, posilované učení, učení posilováním nebo zpětnovazební učení. Reinforcement learning, dále RL, se oblastech operačního výzkumu a řízení objevuje pod názvem approximované dynamické programování. RL je oblast strojového učení, která kombinuje dvě disciplíny - dynamické programování a supervised learning.

Dynamické programování je oblast matematiky, která se tradičně používá k řešení problémů optimalizace a řízení. Tradiční dynamické programování je však limitováno velikostí a složitostí problémů. Základní myšlenkou je rozklad problému na podproblémy. Podproblémy jsou řešeny a jejich řešení je uloženo pro případné další použití. V RL se prostředí modeluje jako Markovský rozhodovací proces a tedy mnoho RL algoritmů se díky této souvislosti vztahuje k technikám dynamického programování.

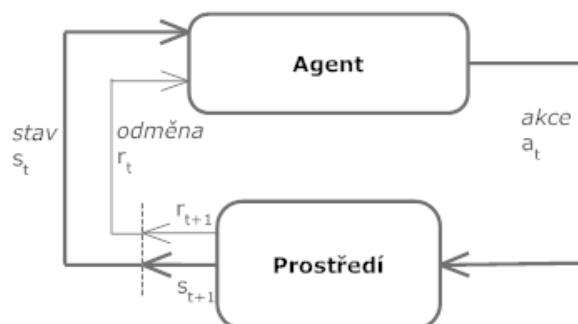
Supervised learning je obecná metoda pro trénování parametrizovaného funkčního approximátoru k reprezentaci funkcí. Supervised learning potřebuje vzorek vstupně-výstupních párů z funkce, která má být naučena. Jinými slovy je zapotřebí množina otázek a k nim množina správných odpovědí. Jedná se o tzv. učení s učitelem, kdy pro vstupní data je určen správný výstup. Můžeme si to představit na následujícím jednoduchém příkladu. Představme si, že chceme počítačový systém naučit, aby rozpoznał infračervený obrázek tanku. Pravděpodobně neznáme nejlepší způsob, jak počítač naprogramovat,

aby z mnoha infračervených obrázků vybral ty, na kterých je tank. Z rozsáhlé sbírky infračervených obrázků umíme určit, na kterém se tank nachází a na kterém ne. Supervised learning se podívá na všechny příklady s odpověďmi a naučí se, jak obecně tank rozpoznat. Bohužel v mnoha situacích neznáme správné odpovědi, které supervised learning potřebuje. Z těchto důvodů se v posledních letech objevil velký zájem o odlišný přístup, reinforcement learning. RL se odlišuje od standardního supervised learningu v tom, že nevyužívá vstupně-výstupní páry.

Kombinací dynamického programování a supervised learningu dostáváme výkonný systém strojového učení, který je schopen řešit mnoho dosud neřešitelných problémů. U RL je počítač jednoduše dán cíl, kterého má dosáhnout. Počítač se pomocí interakcí se svým prostředím, založených na metodě pokus-omyl naučí jak dosáhnout cíle.

5.1 Definice základních pojmu

Ve standardním RL modelu agent interahuje se svým prostředím v diskrétním čase $t = 0, 1, 2, \dots$. Interakce má takovou podobu, že v každém kroku t agent pozoruje - snímá prostředí a získá informaci o jeho stavu s_t . Na základě této informace vybere akci a_t , kterou v prostředí vykoná. Akce nějakým způsobem změní prostředí do jiného stavu s_{t+1} . Přechod můžeme zapsat následovně (s_t, a_t, s_{t+1}) . S přechodem do jiného stavu je spojená odměna r_{t+1} . Informace o změně je agentovi předána prostřednictvím skalárního reinforcement signálu. Cílem RL agentů je získat co největší odměnu (užitek). Interakce agenta s prostředím je znázorněna na obr. 5.1.



Obrázek 5.1: Rozhraní agent-prostředí

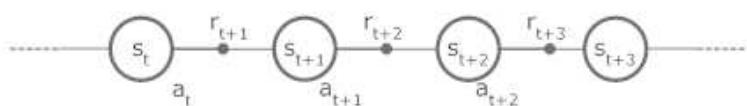
5.1.1 Prostředí

Prostředí musí být RL systémem alespoň částečně pozorovatelné. Jestliže RL systém umí v prostředí perfektně získat všechny informace, které by mohly ovlivnit výběr akce, potom RL systém vybírá akce založené na tzv. pravdivých stavech prostředí.

Jak již bylo řečeno, v RL se prostředí modeluje jako Markovský rozhodovací proces, který je definován následovně:

- S - množina stavů prostředí
- $s_t \in S$ - stav prostředí v čase t
- $A(s)$ - množina dostupných akcí ve stavu $s \in S$
- $a_t \in A(s_t)$ - akce provedená ve stavu s_t v čase t
- $P(s_t, s_{t+1}, a_t)$ - pravděpodobnost přechodu ze stavu s_t do stavu s_{t+1} při provedení akce a_t
- $R(s_t, s_{t+1}, a_t)$ - bezprostřední odměna při přechodu ze stavu s_t do stavu s_{t+1} při provedení akce a_t
- γ - srážkový faktor
- $t = 0, 1, 2, \dots$ - diskrétní čas

Pro názornost je k dispozici obr. 5.2.



Obrázek 5.2: Schéma přechodů mezi stavy

5.1.2 Reinforcement funkce

Jedná se o zobrazení z párů stav/akce na odměny. Tato funkce říká, jaká je odměna v závislosti na tom, v jakém stavu se prostředí nachází a jaká byla provedena akce. Po provedení akce v daném stavu RL agent obdrží odměnu, která je v podobě skalární hodnoty. RL agent se učí provádět akce, které maximalizují součet odměn obdržených, když

začínáme v nějakém počátečním stavu a pokračujeme do konečného stavu. Přestože mohou být definovány složité reinforcement funkce, existují 3 základní třídy často používaných reinforcement funkcí, které správně definují požadovaný cíl. Jsou to "Pure delayed reward", "Minimum time to goal" a "Games". Více o těchto funkčích je možné se dozvědět na (MANCE E. HARMON a STEPHANIE S. HARMON, 2006).

5.1.3 Hodnotová funkce a další termíny

Dříve než si objasníme pojem hodnotová funkce, tak je potřeba se seznámit s několika dalšími termíny.

Strategie (Policy) je definováno následovně

$$\pi : s \in S \rightarrow a \in A(s). \quad (5.1)$$

Jedná se tedy o zobrazení ze stavů na akce. Určuje, která akce by měla být provedena v jakém stavu.

Dalším pojmem je tzv. **výnos (return)**, který se také označuje jako celková snížená odměna. Definice je následující

$$\mathcal{R} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1}, \quad (5.2)$$

kde γ je srážkový faktor, $0 \leq \gamma \leq 1$. Srážkový faktor způsobuje, že bezprostřední odměna má větší důležitost než budoucí odměny. Používá se k exponenciálnímu snížení váhy odměn obdržených v budoucnosti.

Hodnota stavu je definována jako součet odměn obdržených, když začínáme v daném stavu a dodržujeme nějakou danou strategie do konečného stavu. Hodnota stavu tedy závisí na zvolené strategii. Definice hodnoty stavu pro danou strategii π je následující

$$V^\pi(s) = E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots | s_t = s, \pi\} = E\left\{\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} | s_t = s, \pi\right\}. \quad (5.3)$$

Jinými slovy hodnota stavu je očekávaný výnos.

Optimální strategie (Optimal policy) je zobrazení ze stavů na akce, které maximizuje (5.3) součet odměn obdržených, když začínáme v libovolném stavu a provádíme akce, dokud není dosaženo konečného stavu. Hodnota stavu s_t pro optimální strategii je součet odměn obdržených, když začínáme ve stavu s_t a provádíme optimální akce, dokud není dosaženo konečného stavu.

Hodnotová funkce (Value function) je definováno jako zobrazení ze stavů na hodnoty stavů

$$V^\pi : S \rightarrow \mathbb{R}. \quad (5.4)$$

Optimální hodnota, $V^*(s)$, stavu $s \in S$ je definována jako nejvyšší dosažitelný očekávaný výnos, když proces začíná ve stavu s

$$V^*(s) = \max_{\pi} V^\pi(s). \quad (5.5)$$

Funkce

$$V^* : X \rightarrow \mathbb{R} \quad (5.6)$$

se nazývá **optimální hodnotová funkce (optimal value function)**.

Primárním cílem učení je najít "správné" zobrazení ze stavů na hodnoty stavů. Jakmile je toto splněno, tak může být snadno určena optimální strategie. Obecně je $V(s_t)$ iniciализováno náhodnými hodnotami a neobsahuje žádnou informaci o optimální hodnotové funkci. Můžeme říci, že approximace optimální hodnotové funkce $V(s_t)$ v daném stavu se rovná správné hodnotě $V^*(s_t)$ plus nějaká chyba v approximaci

$$V(s_t) = V^*(s_t) + e(s_t), \quad (5.7)$$

kde $e(s_t)$ je chyba v approximaci hodnoty stavu v čase t .

5.2 Temporal difference learning

Dříve než se dostaneme k samotnému Q-učení, tak se seznámíme s technikou učení, z které Q-učení vychází. Temporal difference learning, TD učení patří mezi predikční algoritmy. Je to metoda učení, která se zabývá predikcí veličiny, která závisí na budoucích hodnotách daného signálu. V našem případě chceme odhadnout hodnotovou funkci, která závisí na budoucích hodnotách stavů. TD učení patří mezi inkrementální algoritmy, které dokáží opravit model, pokud obdrží nová data bez přepočítání celého modelu od začátku. Základní vlastností TD učení je *bootstrapping*. To znamená, že aktualizace hodnoty stavu používá odhad hodnoty následujícího stavu.

$\hat{V}_t(s)$ je odhad hodnoty stavu s_t v čase t ($\hat{V}_0 \equiv 0$). Nejjednodušší TD metoda TD(0) provádí v časovém kroku t následující výpočet

$$\delta_{t+1} = r_{t+1} + \gamma \hat{V}_t(s_{t+1}) - \hat{V}_t(s_t), \quad (5.8)$$

$$\hat{V}_{t+1}(s) = \hat{V}_t(s) + \alpha \delta_{t+1} \mathbb{I}_{\{s_t=s\}}, \quad (5.9)$$

kde r_{t+1} je bezprostřední odměna spojená s přechodem ze stavu s_t do stavu s_{t+1} , γ je srážkový faktor ($0 \leq \gamma \leq 1$), $\hat{V}_t(s_{t+1})$ je odhad hodnoty následujícího stavu, α je faktor učení ($0 \leq \alpha \leq 1$) a $\mathbb{I}_{\{s_t=s\}}$ se rovná 1, pokud je splněna podmínka ve složených závorkách. V jakémkoliv jiném případě je 0. Rovnice (5.8) se nazývá "temporal difference error" a odpovídá rozdílu mezi cílem predikce $r_{t+1} + \gamma \hat{V}_t(s_{t+1})$ a současnou predikcí $\hat{V}_t(s_t)$. Nejjednodušší způsob reprezentace hodnotové funkce je pomocí look-up tabulky, kde jednotlivé položky tabulky odpovídají odhadovaným hodnotám stavů. Aktualizační krok TD učení můžeme zapsat takto

$$\hat{V}(s_t) \leftarrow \hat{V}(s_t) + \alpha[r_{t+1} + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t)]. \quad (5.10)$$

5.3 Q-učení

Q-učení (Q-learning) se řadí do kategorie "Off-Policy" TD učení. Místo hledání zobrazení ze stavů na hodnoty stavů Q-učení hledá zobrazení z páru stav/akce na tzv. Q-hodnoty. Nehledáme tedy hodnotovou funkci ale Q-funkci ($Q^\pi : S \times A \rightarrow \mathbb{R}$). Q-hodnota je definována jako součet odměn obdržených, když provádíme asociovanou akci a potom sledujeme danou strategii

$$Q^\pi(s, a) = E\left\{\sum_{i=0}^{\infty} \gamma^i r_{t+i+1} \mid s_t = s, a_t = a, \pi\right\}. \quad (5.11)$$

Podobně jako $V^*(s)$ je optimální Q-hodnota páru stav/akce (s,a) definovaná jako maximální očekávaný výnos, pokud proces začíná ve stavu s a první zvolená akce je a

$$Q^*(s, a) = \max_\pi Q^\pi(s, a). \quad (5.12)$$

Funkce

$$Q^* : S \times A \rightarrow \mathbb{R} \quad (5.13)$$

se nazývá optimální Q-funkce.

Stejně jako hodnotová funkce, tak i Q-funkce může být implementována jako look-up tabulka. Aktualizace Q-hodnoty má následující podobu

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \quad (5.14)$$

Q-učení zaručuje, že optimální chování je nalezeno bez potřeby modelu prostředí

$$\lim_{t \rightarrow \infty} Q(s, a) \rightarrow Q^*(s, a). \quad (5.15)$$

5.4 Analýza Q-učení pro autíčko na dráze

V případě Q-učení je třeba nejprve stanovit cíl. Naším cílem je, aby autíčko projelo neznámou dráhu za co nejkratší dobu, aniž by vypadlo z dráhy. Protože se jedná o dynamický systém, tak stavem je dynamický stav autíčka. Stav si můžeme charakterizovat jako aktuální rychlosť autíčka a aktuální hodnotu bočního zrychlení. Dále je třeba stanovit diskrétní škálu těchto stavů. Otázkou je jak jemně zvolit krok mezi jednotlivými stavami, aby to mikrokontrolér stihl vyhodnotit a spočítat. Dále musíme stanovit, jaká bude reinforcement funkce. Protože se jedná o učení založené na metodě pokus-omyl, tak musíme učícímu agentovi - autíčku dát informaci, jaký stav v závislosti na vykonané akci je "špatný" a jaký "dobrý". Co si představit v naší situaci pod pojmem "špatné" a "dobré"? Jako špatný stav můžeme určitě považovat, když autíčko vypadne z dráhy. Protože ale vypadnutí z dráhy pro nás znamená skončení programu, tak by tento stav nebylo autíčko schopné vyhodnotit a při umístění zpět na dráhu by učení začínalo znovu. Proto si například "špatný" stav můžeme představit, když se hodnota odstředivého zrychlení přiblíží k nějaké maximální hodnotě, při které se autíčko ještě udrží na dráze. A při vykonání akce, která by způsobila přechod do tohoto stavu, bude odměna například -1 a ve všech ostatních případech 0. Akce, které autíčko může vykonávat, jsou zvýšení nebo snížení rychlosti. Při jízdě na dráze by mělo zkoušet vykonávat akce tak, aby postupně prošlo všemi diskrétně stanovenými stavami a při tom počítat Q - hodnoty pro všechny stavů a asociované akce podle (5.14). Finální fází učení je nalezení maximální posloupnosti Q-hodnot pro daný stav a asociovanou akci k určení optimální strategie řízení.

Praktická implementace Q-učení nebyla provedena. Pro začátek, co se týče učících algoritmů, byl naimplementován algoritmus popsaný v následující podkapitole.

5.5 Popis jednoduchého učícího algoritmu

Velmi jednoduchá verze učícího algoritmu byla implementována do mikrokontroléru autíčka v jazyce C. Podobně jako u soutěžního algoritmu předpokládáme, že se na dráze vždy vyškytují dvě křížení. To pro nás znamená, že s jistotou poznáme, kdy jedeme okruh znova. Nebyla modifikována rychlosť, ale byla upravována hodnota činitele plnění PWM , kterou lze jednoduše přepočítat na napětí nastavované na DC motoru.

Po umístění autíčka na dráhu se rozjede nastavenou konstantní hodnotou PWM . V podstatě je jedno, jaká to je hodnota, protože při detekování prvního křížení začne být tato hodnota "upravována". Zvolili jsme například $PWM = 3000$, pokud by prvnímu křížení nějaká zatáčka, tak aby nedošlo k vypadnutí autíčka z dráhy. Dále si připravíme pole, kde bude dráha rozdělena na úseky o námi zvolené délce. Zvolena byla délka úseku 80 mm. Byla zjištěna maximální hodnota bočního zrychlení pro nejmenší poloměr zatáčky při maximální bezpečné průjezdové hodnotě PWM . Hodnota tohoto zrychlení je přibližně $maxAcc = \pm 6000 \text{ mm/s}^2$. V každém úseku tratě, které autíčko projelo je hodnota PWM upravena do dalšího kola o hodnotu, která je určena následovně

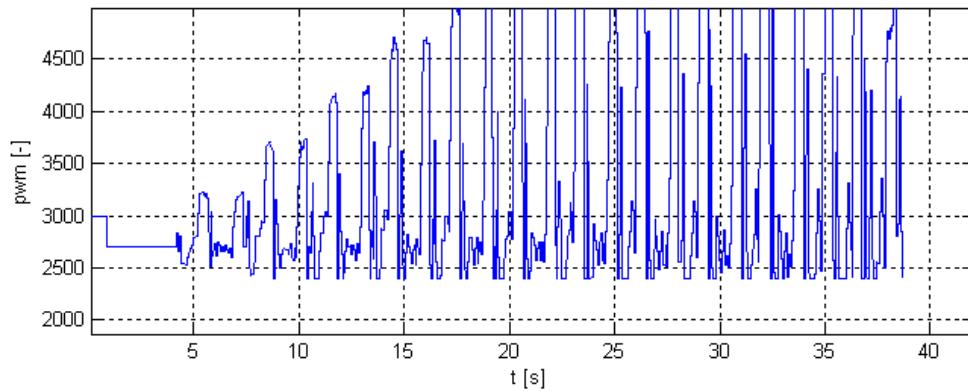
```
if(averageAccX > 0){
    modificationPWM = k*(maxAcc - averageAccX);
} else if(averageAccX <= 0){
    modificationPWM = k*(maxAcc + averageAccX);
}
```

k je experimentálně nalezený koeficient o hodnotě $k = 0,1$. Nová hodnota PWM pro ujetý úsek je dána součtem původní hodnoty PWM a $modificationPWM$. Pokud by nově nastavená hodnota přesahovala zvolené meze, tak je nastavena daná mez. V případě roviných úseků není boční zrychlení tak velké a tedy se rychlosť pořád zvyšuje až dosáhne horní meze. Aby nebyly velké rozdíly rychlosti mezi jednotlivými úseky, tak se měřené filtrované boční zrychlení průměruje. Tím se regulace stává o něco robustnější, ale za cenu trochu horšího celkového času. S jednotlivými hodnotami (např. délka úseku, koeficient k) je třeba experimentovat, pro optimální naladění.

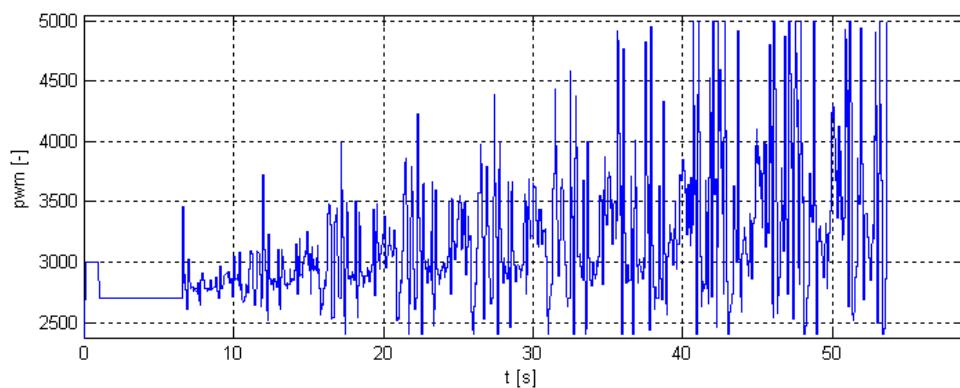
5.5.1 Testování na dráze

Učící algoritmus byl stejně jako soutěžní algoritmus vyzkoušen na oválné a složitější dráze. Na oválné dráze byla průměrná doba projetí deseti kol 30,7 s. Nastavované hodnoty činitele plnění PWM můžeme vidět na obr. 5.3. Na složitější dráze byla průměrná doba projetí deseti kol 51,76 s. a naměřená data jsou na obr. 5.4. Můžeme si všimnout, že na-

stavovaná hodnota PWM postupně konverguje tak, že na rovinných úsecích jede autíčko maximální mezní hodnotou PWM . Jak na oválné tak na složitější dráze je naměřený čas horší než v případě soutěžního algoritmu a konstantní hodnoty PWM .



Obrázek 5.3: Nastavované hodnoty činitele plnění PWM pro oválnou dráhu



Obrázek 5.4: Nastavované hodnoty činitele plnění PWM pro složitější dráhu

Kapitola 6

Závěr

Upravené autíčko (obr. 6.1) se podařilo zkompletovat po elektronické stránce tak, že je schopné měřit proud protékající motorem, svoji rychlosť, boční zrychlení působící na autíčko v zatačkách a napětí z dráhy. Gyro senzory umístěné v autíčku se nakonec nevyužívaly kvůli nesprávné funkčnosti způsobené porouchaným filtrem horní propust, který se nachází na zakoupené desce společně s gyro senzorem. Při testování programu byla velmi užitečná zadní světla. Bylo třeba řešit řadu problémů, které se v průběhu práce objevily. Dlouhou dobu se řešil problém s ukládáním naměřených dat na μ SD kartu. Nakonec se tento problém podařilo odstranit. Příčinou byla chyba ve zdrojovém kódu, kdy program uvázl v jednom z přerušení a nedostal se do přerušení starající se o ukládání dat. Dále se vyskytl problém při přejezdě přes křížení, kdy docházelo k výpadku programu. Kondenzátor o kapacitě 1 mF na CON2, který byl doporučen od pořadatele soutěže, musel být nahrazen silnějším o kapacitě 4,7 mF.



Obrázek 6.1: Fotografie upraveného autíčka

Byl sestaven model autíčka a úspěšně identifikovány příslušné parametry matematického modelu. Na základě modelu byla navržena kaskádní řídicí struktura dvou PI regulátorů pro proudovou a rychlostní smyčku, které byly v diskrétní podobě implementovány.

továny do reálného autíčka.

S kolegou Ing. Danem Martincem jsme připravovali autíčko na soutěž Freescale Race Challenge 2012. Algoritmus řízení fungoval v celku dobře. Jako nevýhoda se ukázala absence senzoru pro detekci startu. Díky tomu autíčko přecházelo do závodního režimu až při třetí detekci křížení. Další vylepšení by se mohlo týkat přidání záložního napájecího zdroje, aby při výpadku z dráhy nedošlo ke smazání vytvořené mapy. V soutěži jsme se umístili na 4. místě z šesti týmů. Postoupili jsme do "Velkého finále" v Rožnově pod Radhoštěm, kterého se však z důvodu nemoci nemohl ani jeden z nás zúčastnit. V době konání soutěže byly problémy s ukládáním dat na μ SD kartu, a tak byly data naměřeny ve škole v laboratoři, až když bylo ukládání funkční.

Seznámil jsem se s reinforcement learningem, konkrétně s algoritmem nazývaný Q-učení. Tento algoritmus může být použit pro autíčko, které by se mělo naučit jezdit na zcela neznámé dráze, tedy bez znalosti stavebních prvků, z kterých může být složena. Bylo analyzováno aplikování Q-učení na autíčko. Pro začátek byl na autíčku vyzkoušen jednoduchý učící algoritmus. Porovnání závodních časů soutěžního algoritmu, učícího algoritmu a konstantní hodnoty činitele PWM na deset kol pro oválnou dráhu je v tabulce 6.1 a pro složitější dráhu v tabulka 6.2. Videa z jízd je možné nalézt na přiloženém CD.

Tabulka 6.1: Porovnání závodních časů na 10 kol na oválné dráze

algoritmus	t [s]
soutěžní	28,06
učící	30,7
konst. PWM	29,4

Tabulka 6.2: Porovnání závodních časů na 10 kol na složitější dráze

algoritmus	t [s]
soutěžní	49,51
učící	51,76
konst. PWM	49,16

Vzniká zde možnost pro pokračování v práci na autíčku, kdy by se podařilo prakticky implementovat Q-učení nebo nějaký jiný učící algoritmus.

Pro vytváření schémat a dalších obrázků byl používán nástroj SmartDraw 2010. Pro modelování byl využit program Matlab se Simulinkem. Zdrojový kód byl psán ve vývojovém prostředí CodeWarrior.

Literatura

- ADVANCED VLSI DESIGN LAB (2012), MEMS and Signal Conditioning Group [online]. [cit. 2012-12-10], [⟨http://conf05.iitkgp.ac.in/avlsi/logf/news/MEMS.pdf⟩](http://conf05.iitkgp.ac.in/avlsi/logf/news/MEMS.pdf).
- ANDREW G. BARTO (2007), Temporal difference learning [online]. [cit. 2012-12-26], [⟨http://www.scholarpedia.org/article/Temporal_difference_learning⟩](http://www.scholarpedia.org/article/Temporal_difference_learning).
- ATMEL (2006), AVR221: Discrete PID controller [online]. [cit. 2013-04-25], [⟨http://www.atmel.com/images/doc2558.pdf⟩](http://www.atmel.com/images/doc2558.pdf).
- BARTÁK, R. (2012), Umělá inteligence 2 [online]. [cit. 2012-12-26], [⟨http://kti.mff.cuni.cz/~bartak/ui2/lectures/lecture12.pdf⟩](http://kti.mff.cuni.cz/~bartak/ui2/lectures/lecture12.pdf).
- CHAN (2012), The Electronic Lives Manufacturing [online]. [cit. 2012-12-10], [⟨http://elm-chan.org/fsw/ff/00index_e.html⟩](http://elm-chan.org/fsw/ff/00index_e.html).
- CHIASSON, J. (2005), *MODELING AND HIGH-PERFORMANCE CONTROL OF ELECTRIC MACHINES*, A John Wiley & Sons, Inc. ISBN 0-471-68449-X.
- CHUCK McMANIS (2006), H-Bridges: Theory and Practice [online]. [cit. 2012-12-10], [⟨http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/index.html⟩](http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/index.html).
- ELECTRONICS-TUTORIALS.WS (2012), Electronics Tutorial about Differential Operational Amplifiers [online]. [cit. 2012-12-10], [⟨http://www.electronics-tutorials.ws/opamp/opamp_5.html⟩](http://www.electronics-tutorials.ws/opamp/opamp_5.html).
- EN.WIKIPEDIA.ORG (2012), Temporal difference learning [online]. [cit. 2012-12-26], [⟨http://en.wikipedia.org/wiki/Temporal_difference_learning⟩](http://en.wikipedia.org/wiki/Temporal_difference_learning).
- GM ELECTRONIC (2012), LM358 [online]. [cit. 2012-12-10], [⟨http://www.gme.cz/operacni-zesilovace-bipolarni/lm358-p310-014/#dokumentace⟩](http://www.gme.cz/operacni-zesilovace-bipolarni/lm358-p310-014/#dokumentace).

HW.CZ (2010), FRC2011-SlotCarEquipmentCZ.pdf – návod k vybavení auta elektronikou a vývoji řídicího algoritmu [online]. [cit. 2012-12-10],
[⟨http://www.hw.cz/files/uploads/story/4314/frc2011-slotcarequipmentcz0.pdf⟩.](http://www.hw.cz/files/uploads/story/4314/frc2011-slotcarequipmentcz0.pdf)

HW.CZ (2011), Freescale Race Challenge 2012: Soutěž samořídících autíček na autodráhu pokračuje [online]. [cit. 2012-12-10],
[⟨http://www.hw.cz/teorie-a-praxe/mimochodem/freescale-race-challenge-2012-soutez-samoridicich-auticek-na-autodrahu⟩.](http://www.hw.cz/teorie-a-praxe/mimochodem/freescale-race-challenge-2012-soutez-samoridicich-auticek-na-autodrahu)

KACHROO, P a MELLODGE, P (2005), Mobile Robotic Car design [online]. [cit. 2012-12-26], [⟨http://books.google.cz/books?id=tY6ctARXqjYC&printsec=frontcover&hl=cs&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false⟩.](http://books.google.cz/books?id=tY6ctARXqjYC&printsec=frontcover&hl=cs&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

KRISHNAN, R. (2001), *Electric Motor Drives Modeling, Analysis, and Control*, Prentice Hall, Inc. ISBN 0-13-091014-7.

MANCE E. HARMON a STEPHANIE S. HARMON (2006), Reinforcement Learning: A Tutorial [online]. [cit. 2012-12-26],
[⟨http://www.nbu.bg/cogs/events/2000/Readings/Petrov/rltutorial.pdf⟩.](http://www.nbu.bg/cogs/events/2000/Readings/Petrov/rltutorial.pdf)

MARTINEC, D. (2012), Diploma thesis: Distributed control of platoons of racing slot cars[online]. [cit. 2012-11-20],
[⟨https://support.dce.felk.cvut.cz/mediawiki/images/7/75/Dp_2012_martinec_dan.pdf⟩.](https://support.dce.felk.cvut.cz/mediawiki/images/7/75/Dp_2012_martinec_dan.pdf)

ON SEMICONDUCTOR (2012), Low Power Low Dropout Voltage Regulator [online]. [cit. 2012-12-10], [⟨http://www.onsemi.com/pub_link/Collateral/LP2950-D.PDF⟩.](http://www.onsemi.com/pub_link/Collateral/LP2950-D.PDF)

PALKOVIČ, L. (2009), Článok: Autonómne riadené autíčko pre autodráhu[online]. [cit. 2012-11-20], [⟨Automatizácia a regulácia URPI ŠVOČ 2009⟩.](#)

PAVEL TIŠNOVSKÝ (2008), Externí sériové sběrnice SPI a I²C [online]. [cit. 2012-12-10],
[⟨http://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/⟩.](http://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/)

PENC, O. (2010), Bakalářská práce: Řízení modelu automobilu na autodráze[online]. [cit. 2012-11-20],
[⟨https://support.dce.felk.cvut.cz/mediawiki/images/6/6b/Bp_2011_penc_ondrej.pdf⟩.](https://support.dce.felk.cvut.cz/mediawiki/images/6/6b/Bp_2011_penc_ondrej.pdf)

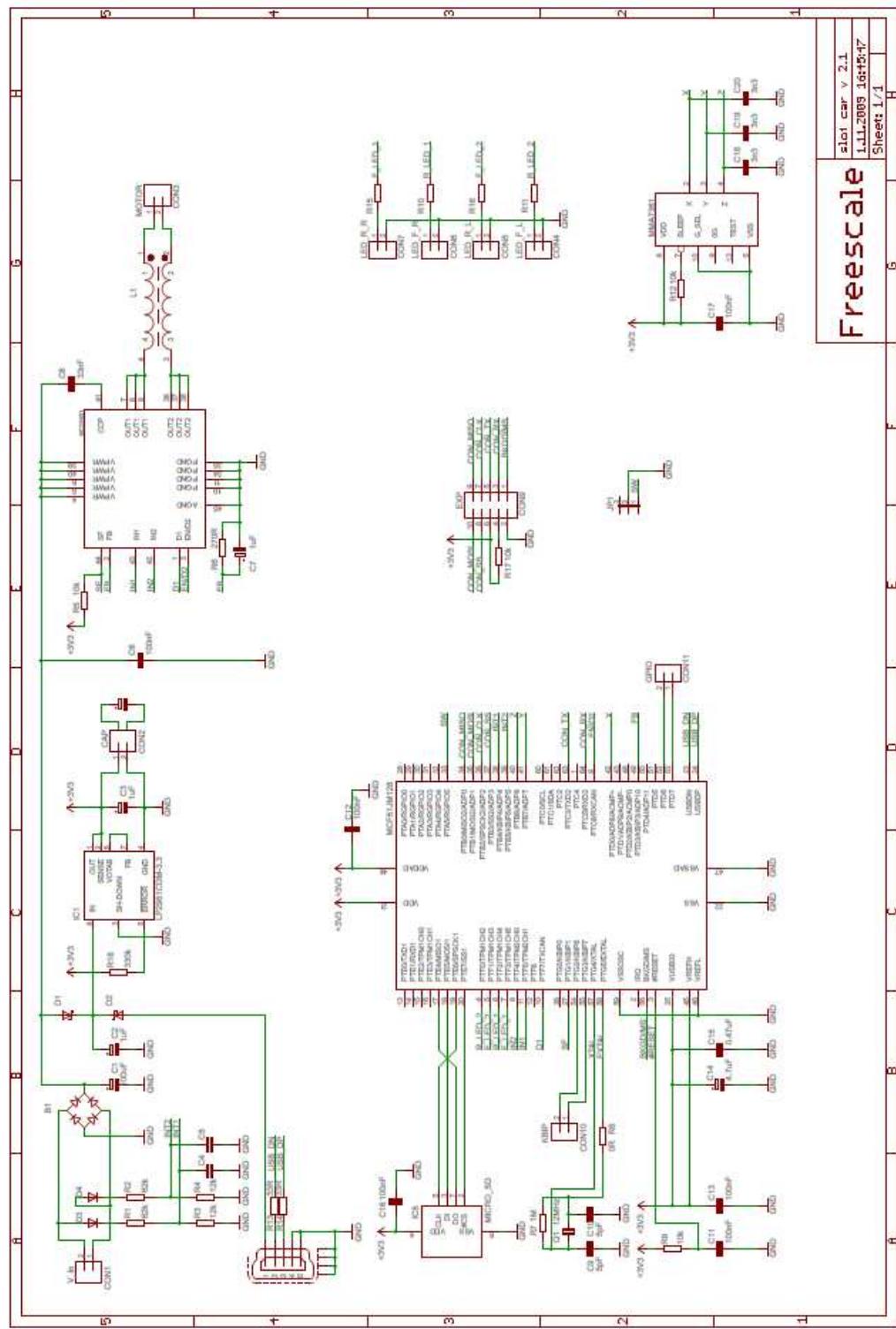
RICHARD S. SUTTON a ANDREW G. BARTO (1998), Reinforcement Learning: An Introduction [online]. [cit. 2012-12-26],
[⟨http://webdocs.cs.ualberta.ca/~sutton/book/the-book.html⟩.](http://webdocs.cs.ualberta.ca/~sutton/book/the-book.html)

- RSS.TUL.CZ (2011), Číslicové filtry (základy) [online]. [cit. 2012-12-10],
⟨http://www.rss.tul.cz/ftppub/cms/11_cisfiltry.pdf⟩.
- SALÁŠEK, J. (2012), Diplomová práce: Řídicí systém pro odlehčenou inerciálně stabilizovanou kamerovou hlavici [online]. [cit. 2012-11-20],
⟨https://support.dce.felk.cvut.cz/mediawiki/images/7/74/Dp_2012_salasek_jan.pdf⟩.
- SCOTT DANIELS (2012), ±1.5g, ±6g Three Axis Low-g Micromachined Accelerometer [online]. [cit. 2012-12-10],
⟨http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf⟩.
- SENSORWIKI.ORG (2012), Gyroscope [online]. [cit. 2012-12-10],
⟨<http://sensorwiki.org/doku.php/sensors/gyroscope>⟩.
- SOCIETY OF ROBOTS (2011), Arduino's AnalogWrite – Converting PWM to a Voltage [online]. [cit. 2012-12-10],
⟨<http://provideyourown.com/2011/analogwrite-convert-pwm-to-voltage/>⟩.
- SOCIETY OF ROBOTS (2012), SCHEMATICS - HOW H-BRIDGES WORK [online]. [cit. 2012-12-10], ⟨http://www.societyofrobots.com/schematics_h-bridgedes.shtml⟩.
- SPARKFUN ELECTRONICS (2012a), Gyro Breakout Board - LY530AL - 300°/s [online]. [cit. 2012-12-10], ⟨<https://www.sparkfun.com/products/10100>⟩.
- SPARKFUN ELECTRONICS (2012b), QRE1113 Line Sensor Breakout - Analog [online]. [cit. 2012-12-10], ⟨<http://www.sparkfun.com/datasheets/Robotics/QRE1113-Analog-Breakout-v11.pdf>⟩.
- SPARKFUN ELEKTRONICS (2012), Single Axis MEMs Gyro - LY530ALH [online]. [cit. 2012-12-10],
⟨<http://www.sparkfun.com/datasheets/Sensors/IMU/LY530ALH.pdf>⟩.
- SPARKFUN ELEKTRONICS (2013), ST-DualGyro-PY-Breakout-v10 datasheet [online]. [cit. 2012-12-10], ⟨<https://www.sparkfun.com/datasheets/Sensors/IMU/ST-DualGyro-PY-Breakout-v10.pdf>⟩.
- SZEPESVÁRI, C. (2009), Algorithms for Reinforcement Learning [online]. [cit. 2012-12-26], ⟨<http://www.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf>⟩.
- TU-CHEMNITZ.DE (2012), Temporal-Difference Learning [online]. [cit. 2012-12-26],
⟨http://www.tu-chemnitz.de/informatik/KI/scripts/ws0910/ml09_6.pdf⟩.

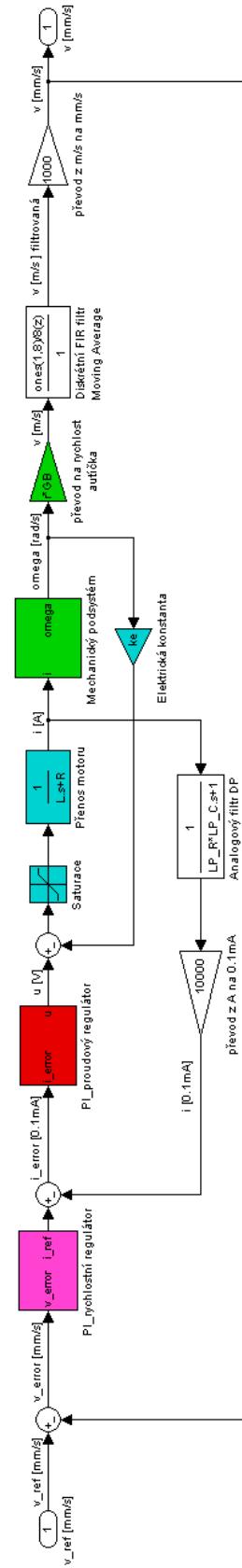
Z-MORAVEC.NET (2005), Diferenční zesilovač [online]. [cit. 2012-12-10],
[⟨http://www.z-moravec.net/ext_el/io/dif_oz.php?format=print⟩](http://www.z-moravec.net/ext_el/io/dif_oz.php?format=print).

Příloha A

Elektronické a simulinkové schéma



Obrázek A.1: Schéma zapojení originální desky plošných spojů



Obrázek A.2: Simulinkové blokové schéma kaskádní řídicí struktury autíčka

Příloha B

Obsah přiloženého CD

K této práci je přiloženo CD, na kterém jsou uloženy zdrojové kódy.

- Adresář CodeWarrior-projekt: Projekt ve vývojovém prostředí Codewarrior pro soutěžní a učící algoritmus
- Adresář Diplomová práce: Samotný text diplomové práce ve formátu .pdf
- Adresář Matlab: Zdrojové kódy
- Adresář Videa: Videa z jízd

(SCOTT DANIELS, 2012) (ADVANCED VLSI DESIGN LAB, 2012) (z-MORAVEC.NET, 2005) (SZEPESVÁRI, C., 2009) (MARTINEC, D., 2012) (ELECTRONICS-TUTORIALS.WS, 2012) (CHAN, 2012) (SPARKFUN ELEKTRONICS, 2012) (SPARKFUN ELECTRONICS, 2012a) (CHUCK McMANIS, 2006) (SOCIETY OF ROBOTS, 2012) (SOCIETY OF ROBOTS, 2011) (SPARKFUN ELECTRONICS, 2012b) (KACHROO, P a MELLODGE, P, 2005) (ON SEMICONDUCTOR, 2012) (GM ELECTRONIC, 2012) (PENC, O., 2010) (BARTÁK, R., 2012) (RICHARD S. SUTTON a ANDREW G. BARTO, 1998) (MANC E. HARMON a STEPHANIE S. HARMON, 2006) (SALÁŠEK, J., 2012) (SENSORWIKI.ORG, 2012) (PALKOVIČ, L., 2009) (PAVEL TiŠNOVSKÝ, 2008) (TU-CHEMNITZ.DE, 2012) (ANDREW G. BARTO, 2007) (HW.CZ, 2010) (HW.CZ, 2011) (EN.WIKIPEDIA.ORG, 2012) (KRISHNAN, R., 2001) (CHIASSON, J., 2005) (RSS.TUL.CZ, 2011) (SPARKFUN ELEKTRONICS, 2013) (ATMEL, 2006)