

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

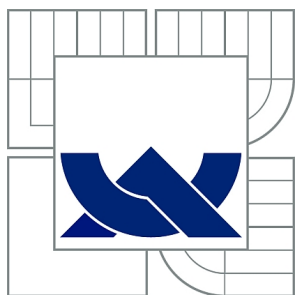
OPTIMALIZACE TRASY AUTONOMNÍHO VOZIDLA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

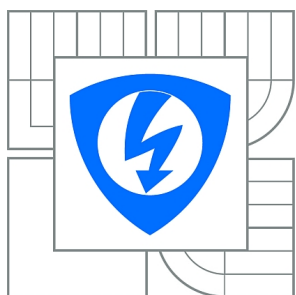
VOJTĚCH VLADYKA

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

OPTIMALIZACE TRASY AUTONOMNÍHO VOZIDLA

ADVANCED PATH OPTIMALISATION METHODS FOR AUTONOMOUS CAR MODEL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VOJTĚCH VLADYKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR PETYOVSÝ

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Vojtěch Vladyka

ID: 155259

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Optimalizace trasy autonomního vozidla

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a realizovat vhodný systém sebelokalizace pro model autonomního vozidla v předem neznámém prostoru (dráze). Na základě informace o umístění vozidla v již identifikovaném prostoru vhodně optimalizovat parametry pro další pohyb vozidla.

1. Seznamte se s hardwarovými a senzorickými možnostmi modelu autonomního vozidla.
2. Nastudujte základní principy sebelokalizace a mapování trasy použitelné pro danou aplikaci.
3. Navrhněte senzorické řešení pro určení budoucí trasy využívající vhodné senzory.
4. Při pohybu vozidla na trase získajte ze zvolených senzorů vhodnou množinu testovacích dat.
5. Navrhněte a realizujte nástroj pro záznam a zpětnou vizualizaci dat o pohybu vozidla v reálné scéně.
6. S pomocí zaznamenaných dat o pohybu realizujte algoritmy pro sebelokalizaci a optimalizaci parametrů pohybu vozidla na trase.
7. Implementujte zvolené algoritmy pro zpracování dat v reálném čase na dostupném hardwaru vozidla.
8. Zhodnoťte dosažené výsledky. Navrhněte další možné změny vedoucí ke zlepšení parametrů jízdy autonomního vozidla.

DOPORUČENÁ LITERATURA:

[1] EVERETT, H.,R.: Sensors for Mobile Robots theory and application, CRC Press 1995, ISBN 1568810482.

Termín zadání: 9.2.2015

Termín odevzdání: 25.5.2015

Vedoucí práce: Ing. Petr Petyovský

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

ABSTRAKT

Tato práce se zabývá návrhem konstrukce a řízení autonomního modelu automobilu. Hlavní koncepce je určena pravidly soutěže The Freescale Cup, pro kterou je model navržen.

KLÍČOVÁ SLOVA

robot, autonomní jízda, The Freescale Cup, zpracování obrazu,sebelokalizace, heuristiky

ABSTRACT

This paper is about design of mechanical and control solution of autonomous model car. Primary design points are given by rules of The Freescale Cup for which is model designed.

KEYWORDS

robot, autonomous drive, The Freescale Cup, image processing, selflocalization, heuristics

VLADYKA, Vojtěch *Optimalizace trasy autonomního vozidla*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2014. 52 s. Vedoucí práce byl Ing. Petr Petyovský

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Optimalizace trasy autonomního vozidla“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval Ing. Miloslavu Richterovi, Ph.D. (univerzitní koordinátor soutěže The Freescale Cup) za poskytnutí modelu, součástek a dráhy na testování. Dále bych rád poděkoval Ing. Petru Petyovskému za odborné konzultace a mnoho hodnotných tipů a postřehů.

Brno

.....

(podpis autora)

OBSAH

Úvod	9
1 O soutěži The Freescale Cup	10
1.1 Pravidla soutěže	10
2 Senzorické možnosti	11
2.1 Kamerová jednotka	11
2.1.1 Výstup dat z kamery	12
2.1.2 Možnosti umístění kamery	12
2.2 IR pár vysílač - snímač	14
2.2.1 Mechanická konstrukce snímače	14
2.2.2 Předzpracování signálu ze snímače	15
2.3 Hallova sonda	15
2.4 Akcelerometr	16
2.4.1 Použitý snímač	16
2.5 Magnetometr	17
2.6 Měření proudu motorem	18
3 Sebelokalizační možnosti	19
4 Mechanická konstrukce	20
5 Řízení	22
5.1 Řídící heuristiky	22
5.1.1 Rozhodovací struktura	23
5.1.2 Segment STRAIGHT	23
5.1.3 Segment TURN	24
5.1.4 Segment CROSS	26
5.2 Zpracování obrazu	26
5.2.1 Operátor detekce hran	27
5.2.2 Doplnkové filtrace	28
5.2.3 Segmentace obrazu	28
5.3 Řízení kamery	28
5.4 Řízení pohonů	29
5.4.1 Řízení pohonných motorů	30
5.4.2 Řízení servomotoru	30
5.5 Odesílání telemetrie pro ladění algoritmů řízení	31

6	Testovací data	32
6.1	Testovací vzorky dráhy	32
6.2	Metoda získávání testovacích dat	32
6.3	Popis formátu telemetrických dat	33
7	Koncepční řešení	35
7.1	Koncepce se dvěma kamerami	35
7.1.1	Řízení	36
7.1.2	Problémy	36
7.2	Koncepce s jednou kamerou v přední části mimo směr jízdy	36
7.2.1	Řízení	36
7.2.2	Problémy	36
7.3	Koncepce s jednou kamerou v přední části ve směru jízdy	37
7.3.1	Řízení	37
7.3.2	Problémy	38
7.4	Koncepce s jednou kamerou umístěnou v zadní části ve směru jízdy	38
7.4.1	Řízení	39
7.4.2	Problémy	39
8	Závěr	40
	Literatura	42
	Seznam symbolů, veličin a zkratk	44
	Seznam příloh	45
A	Řídicí struktura modelu vozidla	46
B	Dráha	47
C	Přílohy na DVD	50
D	Dokumentace k ladicím nástrojům	51
D.1	Návod k použití nástroje dbg_backend	51
D.2	Návod k použití nástroje parser	52

SEZNAM OBRÁZKŮ

2.1	Porovnání světelností ideální a reálné kamery	12
2.2	Umístění kamery nad přední nápravou (zdroj: http://fit.galileo.edu) .	13
2.3	Umístění kamery před přední nápravou (zdroj: [11])	13
2.4	Zjednodušená varianta zapojení odrazového snímače CNY70	15
2.5	Blokové schéma akcelerometru (zdroj [9])	16
2.6	Orientace akcelerometru vzhledem k modelu	17
4.1	Pohled z boku se zvýrazněnými senzory	20
5.1	Schematické znázornění detekce zatáčky při jízdě po rovině	24
5.2	Schematické znázornění heuristiky průjezdu opačných zatáček	25
5.3	Schematické znázornění heuristiky průjezdu stejných zatáček	26
5.4	Příklad původního a zpracovaného obrazu	27
5.5	Směry hledání okrajových čar v obraze	29
5.6	Průběh PWM signálu pro řízení servomotoru	30
7.1	Koncepce se dvěma kamerami	35
7.2	Koncepce s jednou kamerou vpředu	37
7.3	Koncepce s jednou kamerou vzadu	38
B.1	Díl zatáčka	47
B.2	Díl rovina	47
B.3	Díl křižovatka	48
B.4	Díl šikany	48
B.5	Díl start	49
B.6	Díl kopec	49

ÚVOD

V dnešní době je stále častější úloha automatického řízení mobilních robotů neboli AGV (Automated Guided Vehicle). Různé AGV vyžadují různé přístupy. V praxi používané automatické vozíky pro transport materiálu v průmyslových závodech řídící se magnetickou páskou. Projekt společnosti Google realizující plně automatický automobil schopný jízdy v provozu se řídí pomocí lidarů a skupiny kamer. Úloha řešená v této práci vyžaduje optické snímání dráhy a následné řízení podle nasnímaných dat v reálném čase. K tomu je zapotřebí obraz v reálném čase zpracovat a najít v něm vzory nezbytné pro rozpoznání dráhy.

Obsahem této práce je právě návrh řízení modelu v předem neznámém, ale definovaném prostoru. Konstrukce je limitovaná pravidly soutěže The Freescale Cup. Jedním z limitů je například právě ono optické snímání dráhy nebo řízení pomocí jediného mikrokontroléru.

Dráha je definovaná jako bílá plocha široká 600 mm s 17 mm širokými okrajovými hranami. Okolí dráhy není exaktně definováno. Dráha je složená ze známých dílů, kterých je konečný počet. Tento předpoklad je velice důležitý pro samotné řešení.

Důležitou částí práce je výběr senzorů. Vzhledem k požadavku soutěže na optické snímání dráhy je třeba vybrat kameru. V dnešní době jsou pro tuto úlohu přijatelné kamery s technologií CCD a CMOS. Důležitým parametrem bude samotné rozlišení a potažmo použití řádkové nebo matickové kamery. Dále je potřeba vybrat množinu podpůrných senzorů pro správnou navigaci po dráze.

Vzhledem k faktu, že úloha je zaměřená na mobilní robotiku, bylo by vhodné realizovat bezdrátový přenos dat z modelu vozidla do PC pro lepší možnosti ladění a návrhu algoritmů. Stejně tak by bylo vhodné mít možnost zobrazit data jak surová, tak zpracovaná. Protože je pravděpodobné, že bude potřeba provádět i statickou analýzu nasbíraných dat, bude třeba nástroj pro zpětnou vizualizaci obrazu. Návrh řídicích algoritmů bude jádrem celé práce, protože ty budou z největší části ovlivňovat výsledné chování modelu. Stejně tak zpracování obrazu a fúze dat pro řízení. Budou realizovány rozpoznávací heuristiky a na jejich základě se bude měnit řízení modelu vozidla v určitých situacích.

Implementace samotná bude v jazyce C na 32 bitovém mikrokontroléru od společnosti Freescale. Protože tato úloha je kritická na čas, bude třeba snažit se o optimalizaci algoritmů pro co nejrychlejší průběh. V ideálním případě bude čas zpracování algoritmů výrazně kratší než expoziční čas kamery.

Poslední částí je samotná konstrukce modelu vozidla, která je sice z části daná pravidly a neměnností podvozku, ale i přes to je stále mnoho míst pro úpravy. Nejvýraznějším místem pro změny je poloha kamery, kterou bude třeba experimentálně ověřit.

1 O SOUTĚŽI THE FREESCALE CUP

The Freescale Cup (TFC) je celosvětová soutěž, jejíž cílem je konstrukce a naprogramování autonomního závodního modelu automobilu, které má v co nejkratším čase projet zadanou dráhu.

Soutěž se opakuje každý rok již od roku 2003, kdy začala v Jižní Koreji. Od té doby se soutěž rozšířila do celosvětového měřítka se zapojením více než 500 univerzit [3].

1.1 Pravidla soutěže

Tato soutěž se pravidelně vyvíjí s rostoucí úrovní modelů. Letošní verze pravidel [8] obsahuje zásadní změnu oproti loňskému roku. Tou je změna definice dráhy z jedné čáry uprostřed na dvě čáry na krajích. Výčet dílů dráhy je v příloze B.

Jako klíčová konstrukční pravidla lze vybrat tato:

- použití podvozku od společnosti Freescale,
- použití jediného 32 bitového procesoru do společnosti Freescale,
- použití neprogramovatelných optických senzorů pro detekci dráhy,
- maximální počet senzorů je omezen na 12 párů vysílač - přijímač.

Klíčová pravidla týkající se závodní jízdy jsou tato:

- model nesmí opustit dráhu (vždy musí nejméně jedno kolo zůstat na povrchu dráhy),
- model nesmí být dálkově řízen ani nesmí mít vzdálené zpracování dat, veškeré výpočty musí provádět sám (viz. pravidlo jediného procesoru),
- do jízdy nesmí být nijak od startu do cíle zasahováno,
- model se musí sám za cílem zastavit.

Kompletní výčet pravidel je online na stránkách společnosti Freescale [8].

2 SENZORICKÉ MOŽNOSTI

Podle pravidel soutěže [8] jsou senzorické možnosti omezeny na 12 senzorů. Jako jeden senzor jsou brány následující možnosti:

- CCD kamera,
- IR pár vysílač - snímač,
- hallova sonda,
- akcelerometr,
- magnetometr,
- snímač proudu integrovaný v řadičích motorů.

Z tohoto seznamu je vidět cíl soutěže. Řízení podle dat z kamery a následného zpracování obrazu v reálném čase podpořené ostatními senzory.

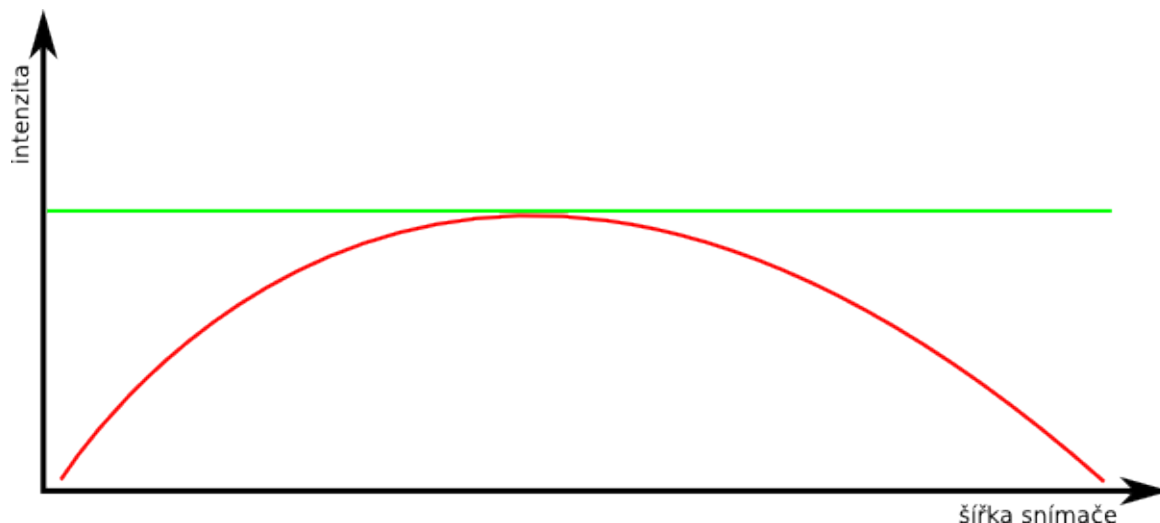
Dalším požadavkem je, že dráha musí být snímána opticky. Toto opatření je z důvodu možnosti sledování hrany dráhy taktilním senzorem.

Poslední omezení týkající se senzorů, je zákaz programovatelných senzorů. Výsledkem tohoto pravidla je zpracování všech dat pouze na jediném mikrokontroléru bez jakéhokoli programového předzpracování. Předzpracování diskrétní elektronikou je povoleno. Většina těchto senzorů a technik jejich zpracování a návrhu je popsána v knize *Sensors for mobile robots: theory and application* [1].

2.1 Kamerová jednotka

Kamera poskytovaná společností Freescale je osazena snímačem od společnosti TAOS, model TSL1401CL [5] a jedná se o řádkovou kameru s rozlišením 128x1 px s hradlovaným analogovým výstupem. Kamera je osazena optikou od společnosti Freescale s neznámým výrobcem a parametry. Tato optika má manuální ostření v široké škále ohniskových vzdáleností. Vyzkoušený rozsah ostření je od 150 mm do 400 mm. Nevýhodou této optiky je její špatná světelnost v okrajových oblastech.

Jak je vidět na obrázku 2.1, ideální průběh světelnosti (zelená) je přímka při záběru na rovnoměrně osvětlenou bílou plochu. Reálný průběh (červená) z dodávané optiky připomíná parabolu, která v krajích jde až do černé. To způsobuje ztrátu informace téměř na 20% šířky obrazu, což je při 128px širokém obrazu 26px, které jsou nepoužitelné. Z tohoto důvodu je zde možnost zlepšení při použití jiné optiky s lepšími parametry.



Obr. 2.1: Porovnání světelností ideální a reálné kamery

2.1.1 Výstup dat z kamery

Kamera nemá žádné vlastní předzesílení. Veškeré vyvedené piny vedou přímo na snímač. Ten má kromě napájení vyvedeny dva řídicí vstupy a jeden výstup.

- SI - Start Impulse - Spouštěcí signál. Po jeho přijetí začne senzor vybavovat data na výstup. Data jsou čtena následujících 18 hodinových tiků.
- CLK - Clock - Hodinový vstup přímo určuje rychlost vybavování dat na výstup. Lze dynamicky měnit.
- AO - Analog Out - Analogový výstup na který jsou vybavovány napětí jednotlivých pixelů rychlostí určenou hodinovým signálem.

Absence předzesilovače je ovšem poměrně nepříjemná z několika důvodů. Kamera pracuje v 3V3 logice, takže výstup má maximálně napětí 3,3V. To je hodnota dosažitelná při přesaturování. Reálné napětí je okolo 2V.

Signál z předzesilovače je přiveden na analogový vstup mikrokontroléru, kde je vzorkován a dále zpracováván.

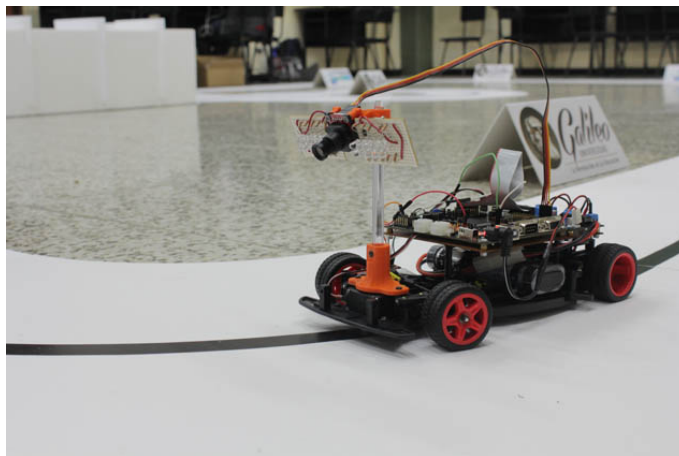
2.1.2 Možnosti umístění kamery

Kameru je možno umístit kamkoli na podvozek, pokud budou dodrženy rozměrové limity 250mm x 400mm x 305mm (viz pravidla TFC [8]). V materiálech k soutěži [11] jsou navržena dvě řešení:

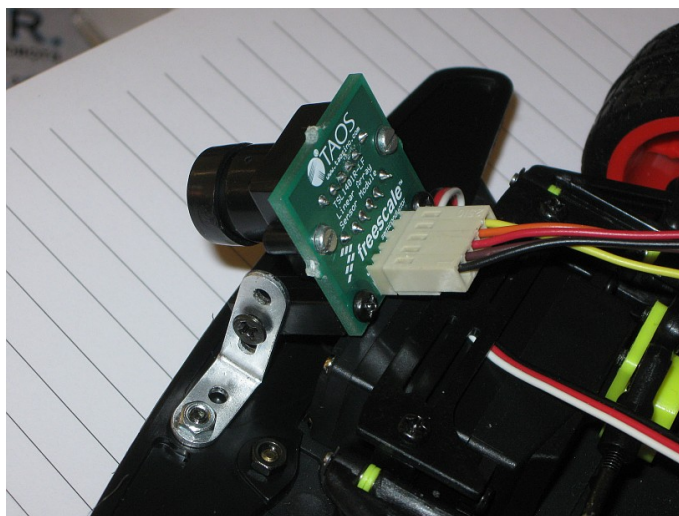
- kamera na předním nárazníku hledící vpřed, (viz obr. 2.3)
- kamera na sloupku nad přední nápravou hledící vpřed (viz obr. 2.2).

K tomuto navrženému řešení lze ještě přidat existující konstrukce:

- kamera na sloupku nad přední nápravou s ramenem před podvozek hledící dolů,
- kamera na sloupku v první třetině podvozku s kamerou hledící vpřed.



Obr. 2.2: Umístění kamery nad přední nápravou (zdroj: <http://fit.galileo.edu>)



Obr. 2.3: Umístění kamery před přední nápravou (zdroj: [11])

První řešení má několik problémů a prakticky žádné výhody. Tato konstrukce byla navržena pro jízdu po jedné čáře (viz pravidla TFC 2014 [12]) a pro jízdu se dvěma čárami se již nehodí. Její největší síla byla v ostrosti obrazu a minimu ruchů přímo před vozidlem. Nevýhodou byl krátký dohled a špatná zorná šířka. To je způsobeno malým úhlem mezi osou kamery a dráhou. Reálným konstrukčním problémem je vystavení kamery nárazům při selhání řízení.

Druhé řešení je výchozí pro většinu konstrukcí a odstraňuje jak problémy se zornou šířkou tak s dohledem. Při nárazu je kamera chráněná nárazníkem. Nevýhodou této konstrukce je její umístění nad nápravou. Toto místo je velice nestabilní a každý pohyb serva a následné mírné zatočení se projeví přímo uměrně uhnutím obrazu.

Třetí řešení je reálné řešení, které bylo navrženo pro jízdu po jedné čáře a jedná se o zdokonalení prvního řešení. Umístění kamery na L-rameno odstranilo problém s úhlem a s dohledem i zorným polem. Ovšem rozkmit je v této poloze tak velký, že řízení je při vyšších rychlostech již téměř nemožné (tým s tímto řešením nedokončil kvalifikaci).

Čtvrté řešení je nejčastěji používané a jedná se o zdokonalení druhé varianty. Umístění sloupku za servo snižuje kmitavost obrazu a zároveň zachovává kladné vlastnosti. Jediná nevýhoda je větší vzdálenost od dráhy a tudíž nižší zorný úhel při stejném dohledu nebo kratší dohled. Rozdíl několika centimetrů je ovšem při těchto rozměrech zanedbatelný. Toto řešení je používáno mnoho úspěšnými týmy ale nakonec nen9 použito i v naší konstrukci. Více o jednotlivých koncepcích v kapitole 7

2.2 IR pár vysílač - snímač

Tyto párové dvojice jsou použity obvykle pro sběr dat odometrie. Ovšem je možné je použít i pro sledování dráhy jako sekundární senzory.

2.2.1 Mechanická konstrukce snímače

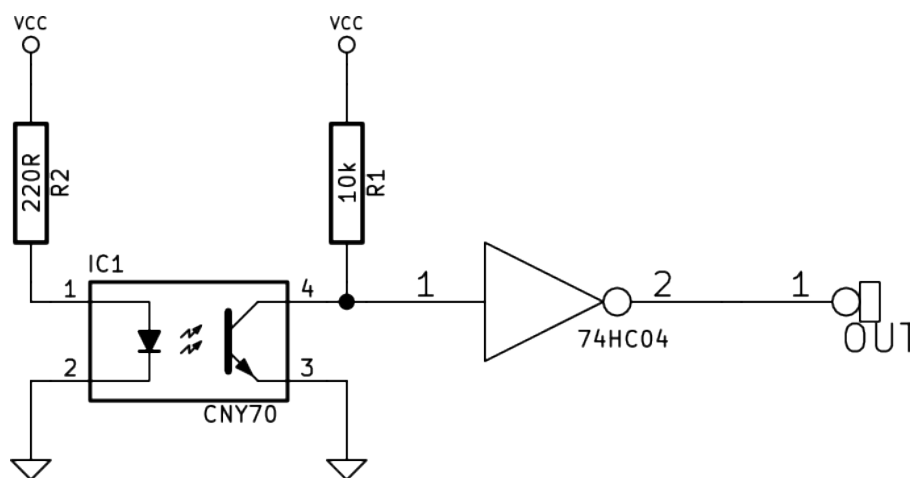
Konstrukce těchto snímačů bývá v podobě IR LED a IR tranzistoru. Obvykle bývají v jednom pouzdře oba prvky a pak se hodí jako odrazová čidla. Populární senzor použitý i námi je CNY70 [13]. Ten v pouzdře 5 x 5 mm obsahuje vysílač i přijímač bez přidané elektroniky. Konstrukce obsahuje i stínítko pro zamezení odrazům z okolí při dodržení požadované minimální vzdálenosti.

Jako odrazná plocha je použit odrazný terčik připevněný ke kolu nebo jiné rotující části. Odrazný terčik má na sobě střídavě odrazivé a neodrazivé plochy, nejčastěji reprezentované bílou a černou. Časté umístění terčíku bývá na kole na straně mezi podvozkem a kolem. Počet dílků potom určuje rozlišení snímače. Minimální šířka dílku je daná vzdáleností vysílače a snímače, respektive jeho rozlišovací schopností. Mnou navržené řešení je umístění odrazeného terčíku na pastorek motoru. Terčik je složen pouze z bílé a černé poloviny a poté jedna perioda odpovídá jedné otáčce motoru. Převodový poměr mezi motorem a kolem je 1:7,5, takže na jednu otáčku kola (a ujetou vzdálenost 15 cm) vychází 7,5 period, což je 15 detekovatelných hran. Výhoda tohoto řešení je snažší zakrytí senzoru před světlem a lepší ochránění před nárazy.

2.2.2 Předzpracování signálu ze snímače

Výstupem ze snímače je spínání tranzistoru. Ovšem při reálných podmínkách tranzistor není při černé plně zavřený a při bílé plně otevřený. Místo toho se přizavírá a pootevívá. Z tohoto důvodu je nutná předřazená elektronika.

Jednou možností je klopný obvod s nastavitelnou překlápěcí úrovní. Je třeba určit rozhodovací úroveň, pro kterou je tranzistor uzavřený a pro kterou otevřený. Takový signál je poté již snadno zpracovatelný mikrokontrolérem při jeho přivedení na digitální vstup. Druhou možností je předpoklad, že optotranzistor v zapojení se společným emitorem se při zastínění zavírá a tudíž stoupá výstupní napětí a obráceně. Při použití logického obvodu NOT jako tvarovače je možné vhodnou fyzickou konstrukcí zamezit odrazům a způsobit zavření tranzistoru natolik, aby se logický obvod překlátil. Jemným doladěním proudu diodou vysílače je možné nastavit výstupní napětí tranzistoru při sepnutí na hodnotu nutnou pro překlacení logického obvodu. Toto řešení není univerzálně použitelné, ale je velice výhodné pro tuto konkrétní aplikaci, kde je konstantní vzdálenost odrazného terčičku, který má konstantní parametry odrazivosti pro černou a bílou. Toto řešení je použité v konstrukci.

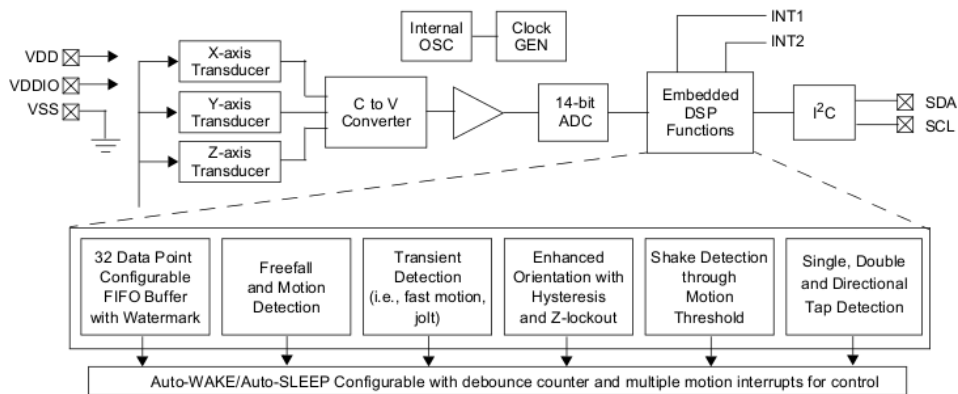


Obr. 2.4: Zjednodušená varianta zapojení odrazového snímače CNY70

2.3 Hallova sonda

Hallova sonda byla použita několika týmy v minulosti s různou úspěšností. Stejně jako pro IR snímač je nutná předřazená elektronika. Ovšem použití hallových sond nese více potíží.

Protože se jedná o snímače magnetického pole, jejich nasazení u motorů je téměř



Obr. 2.5: Blokové schéma akcelerometru (zdroj [9])

vyločené z důvodu magnetického rušení. Tudíž se dají použít pouze u předních kol. Další problém je umístění magnetu do ráfku kola a jeho protiváhy nebo druhého magnetu. Upevnění magnetů je problematické z důvodu vysokých přetížení, která se vyskytují na kolech při rázech na dílu s retardérem i při nárazu při selhání řízení. Obecně nepovažuji toto řešení za optimální.

2.4 Akcelerometr

Akcelerometr je základní komponenta pro určování polohy. V zemské gravitaci je pomocí něj možné učit absolutní hodnotu natočení.

Pro účely soutěže se hodí trojosý akcelerometr, protože pohyb vozidla je možný ve dvou rozměrech (jízda vpřed a vzad, zatáčení) a dráha obsahuje i převýšení v podobě mostu a retardérů (třetí rozměr).

Při vhodném umístění akcelerometru, kde osa X souhlasí s směrem jízdy vpřed, osa Y odpovídá bočnímu pohybu a osa Z odpovídá pohybu vzhůru, je snadné z takovýchto dat opatřených časovými údaji zpětně určit pohyb vozidla. V kombinaci s daty z odometrie je možné určit i polohu při smyku.

2.4.1 Použitý snímač

Nasazený senzor je součástí desky mikrokontroléru [4]. Jedná se o trojosý MEMS akcelerometr od společnosti Freescale, model Xtrinsic MMA8451Q [9] s rozsahem $+2g$, $+4g$ nebo $+8g$ se 14 bitovým výstupem komunikující po I²C sběrnici. Maximální frekvence vybavování hodnot je 800 Hz. Jeho poloha na desce odpovídá souřadnému systému, viz obrázek 2.6. Tento snímač obsahuje i integrované DSP

funkce, které nejsou použity. Vnitřní konstrukce je vidět z následujícího blokového schématu 2.5.



Obr. 2.6: Orientace akcelerometru vzhledem k modelu

2.5 Magnetometr

Magnetometr je zařízení schopné měřit intenzitu magnetického pole. Jednoosý magnetometr je kompas. Hlavní problém při použití magnetometru je nehomogenost magnetického pole, které je ovlivňované jak magneticky měkkými, tak tvrdými materiály a také magnetickými poli vytvářenými elektronikou. Výsledkem je obtížné zpracování dat, které obsahují velké množství šumu a nevalidních dat. Z tohoto důvodu je dobré magnetometr používat pouze jako doplňkový senzor a nepoužívat ho jako hlavní prvek navigace.

2.6 Měření proudu motorem

Měření proudu motorem je možné díky integrovanému řešení budiče motoru [10], které má jako zpětnou vazbu proud motorem. Výstup z můstku je proudový a jeho hodnota je 0.24% skutečného výstupního proudu. Na desce s můstkem [10] je však integrovaný převodník proudu na napětí s dolní propustí a výstup je tedy napěťový a zesílený pro efektivní využití rozsahu (maximální proud, který mohou můstky řídit je 6A, zkratový proud motoru je 3.8A přičemž normální běhový proud je do 1A). Převodní konstanta snímače proudu motorem je tedy **4,5 V/A**.

Těchto dat lze využít při známé zatěžovací charakteristice pro určení záběru pohonu. Při fúzi těchto dat s daty z akcelerometru a odometrie lze získat kompletní přehled o pohybu vozidla.

3 SEBELOKALIZAČNÍ MOŽNOSTI

Sebelokalizace je nezbytná pro navigaci v předem známém i neznámém prostoru. Do sebelokalizace spadá schopnost robotu prozkoumat okolí a na základě nasbíraných dat si vytvořit obraz, ze kterého vyvodí další pohyb. Použité sebelokalizační techniky mohou být jak taktilní tak proximitní. Pravidla soutěže ovšem vyžadují proximitní řešení, konkrétně optické.

Pro tuto konkrétní úlohu je sebelokalizace provedena následovně:

- Umístění na start (nebo jiný výchozí bod dráhy). O tomto bodě víme, že musí být z dílu rovina a musí být bílý s černými okraji o daných rozměrech. Dále víme, že vozidlo je na dráhu umístěno uprostřed a je natočeno ve směru jízdy.
- Řádková kamera [5] je namířena dopředu ve směru jízdy a je nastavena do známé vzdálenosti.

V tomto bodě vozidlo stojí a má známou polohu. Neznámý je další díl dráhy.

- Podle předem nadefinovaných pravidel se vozidlo rozjede vpřed. Podle dat z odometrie a akcelerometru víme, zda nastavená rychlost odpovídá skutečnosti (kontrola prokluzu), a tudíž víme i ujetou vzdálenost. Kamera snímá dráhu v známých intervalech a na základě těchto dat je možné skládat mapu dalšího dílu dráhy.
- Po ujetí definované vzdálenosti (konec známého dílu) a nasbírání dat z kamery bychom měli být schopni identifikovat další díl dráhy. Po jeho úspěšné identifikaci se nastaví jízdní parametry na jízdu po novém dílu. V reálném případě by měla identifikace proběhnout ještě před koncem starého dílu pro případné znovuidentifikování nového dílu jako potvrzení předchozí identifikace.
- Data z odometrie slouží pro kontrolu nastavených jízdních parametrů. Data z kamery slouží pro identifikaci dílů dráhy. Jako celek tato data tvoří množinu dat pro sebelokalizaci.

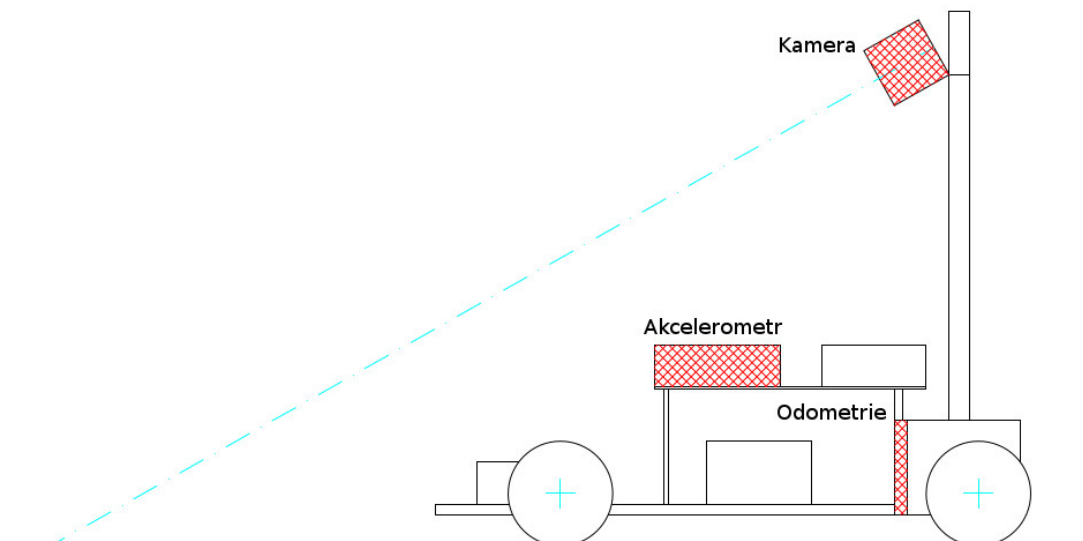
4 MECHANICKÁ KONSTRUKCE

Navržené řešení vychází z úspěšných konstrukcí použitých v minulých ročnících TFC. Mezi důležité parametry se řadí hmotnost, poloha těžiště a umístění kamery. Model sám váží asi 800g a s hotovou nadstavbou a elektronikou je výsledná hmotnost 1110g. Tato hmotnost je výrazně nižší oproti první verzi, která se účastnila kvalifikace na TFC 2014. Nižší hmotnost sebou nese výhodu v podobě lepší akcelerace a brzdění a lepší manévrovatelnosti (nižší vynášivost v zatáčkách). Zásadní nevýhoda je ovšem při jízdě přes retardéry, kde těžší model je méně náchylný na vypadnutí na této překážce. Všeobecně však nižší váha vychází lépe.

Dalším významným parametrem je poloha těžiště. Všechny vítězné modely mají těžiště nízko a uprostřed mezi nápravami. Originální model má těžiště blízko zadní nápravě. Pomocí manipulce s baterií se ovšem dá těžiště přesunout do středu mezi nápravami. Snížení těžiště je možné jen částečně oproti holému podvozku. Po přesunutí baterie již na modelu není místo kam umístit elektroniku, tudíž je umístěna v nadstavbě. Vliv této nadstavby je díky její nízké hmotnosti minimální. Největší vliv na polohu těžiště ve vertikální ose má pylon kamery. Ten je vyveden z sklolaminátové trubky. Toto řešení je lehké a robustní. První verze měla konstrukci ze stavebnice Merkur, které je několikanásobně těžší.

Posledním významným parametrem je poloha kamery. Ta je umístěna na pylonu asi 270 mm nad povrchem dráhy nad zadní nápravou. Zorné pole je v této variantě přibližně 520 mm od přední nápravy s šířkou 750 mm.

Podvozek samotný je dodávaný jako kit společností Freescale přímo pro účely sou-



Obr. 4.1: Pohled z boku se zvýrazněnými senzory

těže. Je vybavený dvěma nezávislými pohony na zadní nápravě a servomotorem pro zatáčení. Celý podvozek je odpružený.

Výslednou koncepci včetně všech nastaveb a senzorů je vidět na obrázku 4.1.

Model je vybaven dvěma bezkontaktními otáčkoměry, které jsou realizovány pomocí odrazových čidel CNY70 od firmy Vishay [13]. Jejich konstrukce je popsána v kapitole 2.2.

Dalším senzorem je akcelerometr, který je umístěn na desce mikrokontroléru. Jeho osy jsou vidět na obrázku 2.6.

5 ŘÍZENÍ

Řídicí algoritmus lze rozložit do několika částí (podle složitosti a významnosti):

- řídicí heuristiky,
- zpracování obrazu,
- řízení kamery,
- řízení pohonů,
- odesílání telemetrie.

Všechny tyto části tvoří program, který je zpracováván v MCU.

Jedním z nejdůležitějších parametrů je rychlost provádění programu. Celý program musí běžet v reálném čase. Na celé zpracování dat a rozhodnutí o dalším pohybu je čas přibližně 2 ms. To je při rychlosti jádra procesoru 50 MHz přibližně 100000 hodinových cyklů. Vzhledem k faktu, že tento procesor nemá implementovanou FPU, je snaha udržet všechny výpočty v pevné desetinné čárce, což významně zrychluje zpracování. Další významná charakteristika je použitý datový typ `unsigned char`. Důvod pro použití tohoto datového typu plyne z jeho velikosti a rozsahu. Vzhledem k rozlišení kamery 128x1 px mohou veškeré iterátory nad načteným vektorem být sedmibitové. Osmibitový char má tedy rezervu. Kvůli rychlosti a velikosti načítaných dat je ADC nastavené na rozsah 8 bitů. Výsledná hodnota je tedy opět uložitelná do datového typu `unsigned char`. Tato optimalizace ve fázi návrhu nám umožnila významné zmenšení ukládaných dat, kdy jeden uložený snímek z kamery má velikost pouhých 128B.

Celý řídicí řetězec je zachycen na blokovém schematu v příloze A.

5.1 Řídicí heuristiky

Tato část tvoří páteř celého řešení. Zde se vyhodnocují naměřená data a na jejich základě se rozhoduje o dalším pohybu.

Návrh heuristik byla nejobtížnější práce s mnoha úskalími. Prvním problémem je nutnost uvědomit si jak model 'vidí' dráhu. Na rozdíl od nás nevidí dvourozměrně (matice), ale pouze jednorozměrně (řádek), a onen chybějící druhý rozměr vzniká až samotným pohybem. Druhým problémem je potřeba uvědomit si, jak daleko před samotným modelem je zorné pole. To, že model detekuje v zorném poli zatačku znamená, že je asi 0,5 m před ní. A v zatačce samotné většinu času nedetekuje nic. Tyto dva problémy jsou zcela evidentně odlišné od toho, jak vidíme svět my, lidé. Jak uvedl jeden člen našeho týmu, nejpřesnější přirovnání situace řízení modelu k lidem by mohla být snaha „řídit závodní auto hledíce dalekohledem skrz tankový průzor.“

5.1.1 Rozhodovací struktura

Základní rozhodovací strukturou je stavový automat, který přepíná jednotlivé regulátory (metoda gain-scheduling). Jednotlivé stavy odpovídají dílům dráhy. Právě z tohoto důvodu byla zvolena tato tato struktura.

Stav automatu	Díl dráhy
STRAIGHT	Rovina, šikana
TURN	Zatáčka
CROSS	Křižovatka

Tab. 5.1: Tabulka stavů a dílů dráhy

5.1.2 Segment STRAIGHT

Segment STRAIGHT, neboli rovně, odpovídá výchozímu stavu. Podle pravidel je start na rovině a alespoň jeden následující díl bude též rovně. Tento stav jako jediný neměří ujetou vzdálenost pro potřebu vlastního řízení. Byl zde pro zatáčení použit jednoduchý proporcionální regulátor, který reaguje na odchylky polohy čar v obrazu. Tento regulátor má dva režimy:

- obě čáry v zorném poli,
- jedna čára v zorném poli.

Při stavu, že nebude vidět ani jedna čára není možné o řízení rozhodnout a de-facto se jedná o chybový stav. V tu chvíli je regulátor mimo provoz a natočení kol se nastaví na 0 (tj. jízda rovně).

Při stavu, kdy jsou obě čáry v zorném poli lze usoudit, že vozidlo je uprostřed dráhy a adekvátní jsou pouze jemné korekce.

Při stavu, kdy už je vidět jen jedna čára lze usoudit, že vozidlo se buď blíží ke kraji nebo je výrazně vyosené. Proto následuje akční zásah s větším zesílením.

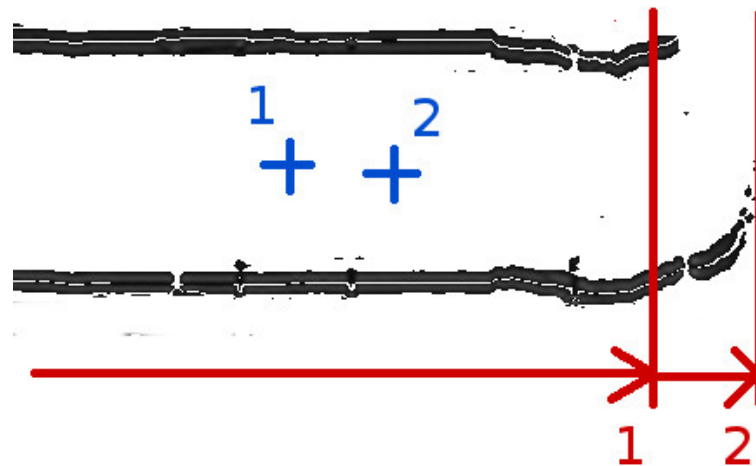
Rychlost je v tomto stavu udržována konstantní na hodnotě přibližně 70% maximální rychlosti. Dalším významným rysem tohoto stavu je způsob rozpoznávání dalšího stavu na základě vzorů. To vychází z výpočtu polohy čar za posledních 5 snímků, kdy se data kategorizují do třech tříd (vlevo, vpravo, střed) a pro každou třídu je uložen počet výskytů čáry v dané třídě. Tato data jsou poté tříděna do dalších tříd, které nahlížejí na data z pohledu postavení čar vůči sobě. Tyto třídy jsou následující:

- LILR - Line In Left and Right
- LILC - Line In Left and Center
- LIRC - Line In Right and Center
- LIC - Line In Center

- LIR - Line In Right
- LIL - Line in Left
- CROSS - Crossection

Do těchto tříd se třídí dle popisu (tj. do třídy LILR spadá pokud je čára vlevo a vpravo a ne uprostřed) s jedinou výjimkou a tou je CROSS. Do této třídy spadají oba zbývající extrémy, tj. žádné čáry anebo čára je všude.

Na základě těchto tříd se použitá heuristika rozhoduje, který díl následuje. To probíhá fúzí těchto tříd s ujetou vzdáleností. Pokaždé, když jsou v obrazu dvě čáry, ujetá vzdálenost je nuluje. Pokud ovšem je vidět pouze jedna čára, zkoumá se ujetá vzdálenost. Pokud je tato vzdálenost menší, než limit (20 cm), pokračuje se podle segmentu STRAIGHT. Ale ve chvíli, kdy ujetá vzdálenost překročí limit, přepne se segment na TURN se směrem opačným, než je poloha čáry. Tato heuristika byla naznačena na obrázku 5.1. Modré křížky označují polohu podvozku a červené svislé čáry zorné pole. V bodě 1 je spuštěno měření vzdálenosti a v bodě 2 je díl vyhodnocen jako zatáčka. Jak je vidět, je to stále s dostatečným předstihem. V tomto případě zatáčka vede doleva.



Obr. 5.1: Schematické znázornění detekce zatáčky při jízdě po rovině

Obdobným způsobem funguje rozpoznání i křižovatky. Tam se zkoumá opět ujetá vzdálenost oproti limitu 10cm a zároveň musí být splněna podmínka dvou po sobě jdoucích tříd CROSS.

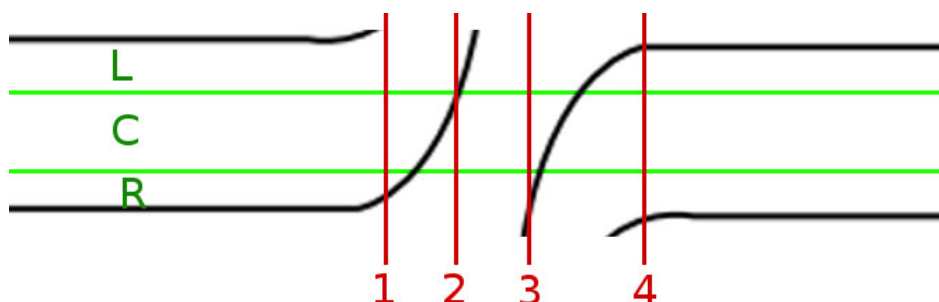
Díl šikana (CHICANE) není detekován a je projet jako rovina.

5.1.3 Segment TURN

Segment TURN (zatáčka) je jeden z nejsložitějších dílů celé dráhy z hlediska řízení. Kvůli sledování obou stran dráhy je zorné pole kamery relativně daleko od přední

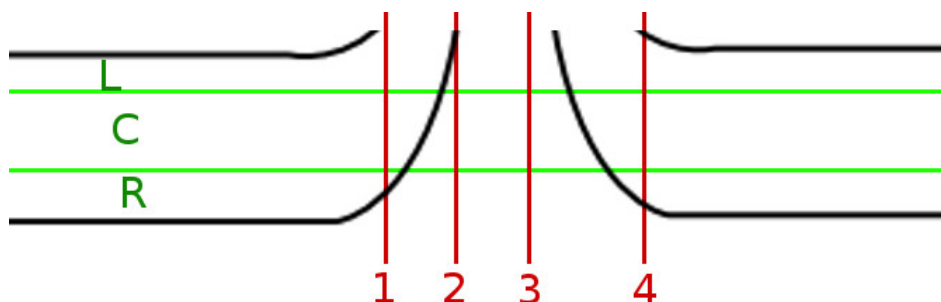
nápravy a z toho plyne, že není možné projet zatáčku proporcionálně, protože ve chvíli, kdy nastává zatáčka, je vozidlo ještě daleko od ní a následovalo by vyjetí z dráhy.

Řešením je detekce zatáčky (viz předchozí kapitola 5.1.2). Ta v stále stejném bodě určí zatáčku, takže je možné poměrně přesně odhadnout vzdálenost do začátku zatáčky. Tato vzdálenost je využita k brzdění. V tu chvíli již jede vozidlo 'naslepo'. Poté, co dorazí na patu zatáčky, přejde algoritmus do třífázového průjezdu zatáčkou. Zároveň si uloží současnou polohu čáry (kterou stále vidí, ale její přeběh je příliš rychlý pro přímé řízení). Poté nastaví kola do definované polohy, nastaví rychlost a podle odometrie projíždí jednotlivé segmenty zatáčky. Zatáčka byla rozsegmentována do třech částí z důvodu preciznějšího vektorování výkonu motorů a natočení kol. Po projetí posledního segmentu následuje rozhodnutí o dalším pohybu. K tomu je zapotřebí porovnat dříve uloženou polohu čáry ze začátku zatáčky s současnou polohou čáry. Pokud je poloha přibližně stejná, jedná se o další zatáčku ve stejném směru. Pokud je poloha opačná oproti středu, jedná se o zatáčku na opačnou stranu. Pokud jsou vidět čáry dvě, jedná se o rovinu. Pokud není vidět nic, jde pravděpodobně o křižovatku. O tu musí jít i po průjezdu třech stejných zatáček. Tato heuristika je nejlépe pochopitelná z obrázků 5.2 a 5.3.



Obr. 5.2: Schematické znázornění heuristiky průjezdu opačných zatáček

V bodě 1 je spuštěno měření vzdálenosti pro rozpoznání zatáčky při jízdě z roviny. Tato situace je stejná jako v kapitole 5.1.2. V bodě 2 je ukončeno měření a je rozhodnuto o směru zatáčky. Uloží se současná poloha čáry pro pozdější použití. Na konci první zatáčky (bod 3) je provedeno hledání čáry. Pokud je nalezena jedna čára a její poloha je opačná podle středu dráhy od polohy v bodě 2, jedná se o zatáčku do opačného směru. Pokud není nalezena žádná čára, je to zatáčka ve stejném směru a pokud jsou dvě čáry v krajních segmentech, jedná se o rovinu. V bodě 4 zjištění dalšího dílu za zatáčkou, v obou případech se jedná o rovinu.



Obr. 5.3: Schematické znázornění heuristiky průjezdu stejných zatáček

5.1.4 Segment CROSS

Posledním detekovatelným segmentem je křižovatka. Způsoby její detekce jsou převážně na základě ujeté vzdálenosti a zkoumání 'prázdnosti' obrazu. Samotný průjezd je opět 'naslepo'. V tomto segmentu se bohužel nedá podle ničeho orientovat a použité servo má jistou necitlivost, takže jízda rovně bez korekcí bývá velice nepřesná (při nejhorším přibližně 20cm na 1m ujeté vzdálenosti). Fakticky to znamená to, že po průjezdu křižovatky je reálná šance mylného rozpoznání zatáčky na rovině, protože algoritmus nemůže vědět, že nejede rovně, a ani neví na kterou stranu vybočil. Proto je předpoklad dalšího pohybu jízda rovně. To ovšem zvyšuje chybovost.

5.2 Zpracování obrazu

Zpracování obrazu je několikafázové a jsou při něm použity téměř výhradně nelineární filtry. Sejmутý snímek (vektor) o rozměru 128x1px započne svůj 'životní' cyklus uložením do bufferu posledních pěti snímků. Poté je ořezán zleva a zprava aby se odstranily problematické části obrazu (extrémně podsaturované okraje z důvodu nekvalitní optiky). Ořez je proveden přepsáním kritických částí na hodnotu posledního validního bodu. Takto připravená data jsou zpracována sobelovým operátorem (více dále v textu) pro detekci hran. Výsledek je ukládán v absolutní hodnotě. Tento obraz je stále v odstínech šedi. Poté je obraz filtrován a převeden do binární podoby. Poslední filtr provede sloučení dostatečně blízkých špiček do jedné a odstraní příliš úzké špičky. Tím končí zpracování obrazu jako takové. Takto zpracovaný obraz je dále segmentován na tři části (levá, pravá, střed) ve kterých jsou hledány špičky. Pokud je špička nalezena, je výsledek o její poloze uložen do příslušného bufferu. Tato data jsou poté použita v rozhodovacích heuristikách. Touto operací končí životní cyklus snímku.

5.2.1 Operátor detekce hran

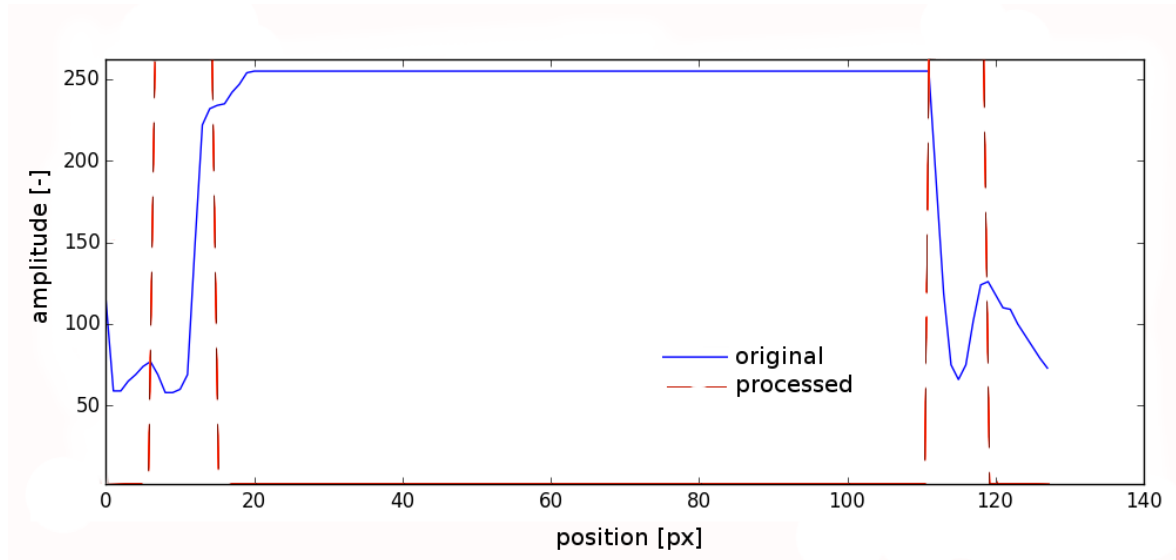
Hranový detektor je jedna ze základních filtrací. V této úloze je použita jednorozměrná varianta diskrétního diferenčního operátoru. Mezi jeho výhody lze zařadit jednoduchost a rychlost zpracování, která je daná jen šířkou operátoru, která je v tomto případě 5. Důvodem k použití širšího operátoru je vyšší odolnost vůči šumu. Tento operátor vychází z Sobelova operátoru a z Prewitt operátoru. Ovšem z důvodu přizpůsobení pro danou úlohu byl experimentálně modifikován.

Použitá konvoluční matice byla experimentálně nastavena na koeficienty v tabulce 5.2.1. Výsledkem takto zpracovaného obrazu je vektor, který v místech s hranou

-5	-3	0	3	5
----	----	---	---	---

Tab. 5.2: Použitá konvoluční matice

obsahuje špičku, jejíž výška odpovídá strmosti hrany v původním obrazu (viz 5.4). Zajímavý je samotný vstupní vektor který je právě kvůli hranovému detektoru lehce rozoostřený. Rozostření obrazu bylo provedeno rozostřením optiky kamery. Tato jednoduchá úprava přirozeně skryla šum a skutečné hrany rozšířila, byť na úkor strmosti. Díky tomu jsou výsledné hrany širší a snáze detekovatelné.



Obr. 5.4: Příklad původního a zpracovaného obrazu

5.2.2 Doplnkové filtrace

Filtrů je v zpracování celá řada. Pomineme-li prosté ořezání obrazu, tak první filtr, který pracuje s obrazem je operátor absolutní hodnoty, který každou zápornou amplitudu vynásobí hodnotou -1 a uloží.

Takto upravený obraz pokračuje do úrovněového filtru na odstranění šumu. Vzhledem k faktu, že obraz byl už při sejmutí normalizován pomocí úpravy saturace, je v tomto filtru pevně určený rozhodovací práh na hodnotě 200 (minimální hodnota jednoho pixelu je 0 a maximální hodnota je 255). Hodnoty, které jsou nižší než práh jsou přepsány na hodnotu 0, hodnoty které jsou nad prahem na jsou přepsány na hodnotu 255. Výsledkem je binární obraz s dostatečně strmými hranami.

Další filtr odstraňuje osamocené špičky a spojuje blízké špičky dohromady. Provedení této operace je opět přes stavový automat. Pokud najde špičku, zkontroluje existenci druhé špičky o dva pixely dále. Pokud ji nalezne, vyplní prostor mezi nimi hodnotou 255. Pokud ne, špičku nastaví na 0 a pokračuje na konci prohledávaného prostoru. Po průchodu tohoto filtru by měl vektor obsahovat pouze špičky s šířkou vyšší nebo rovnou třem. Tento mechanismus byl zaveden pro odstranění falešných špiček, které vznikají například na hraně dráhy, ale pro řízení znamenají falešné údaje.

5.2.3 Segmentace obrazu

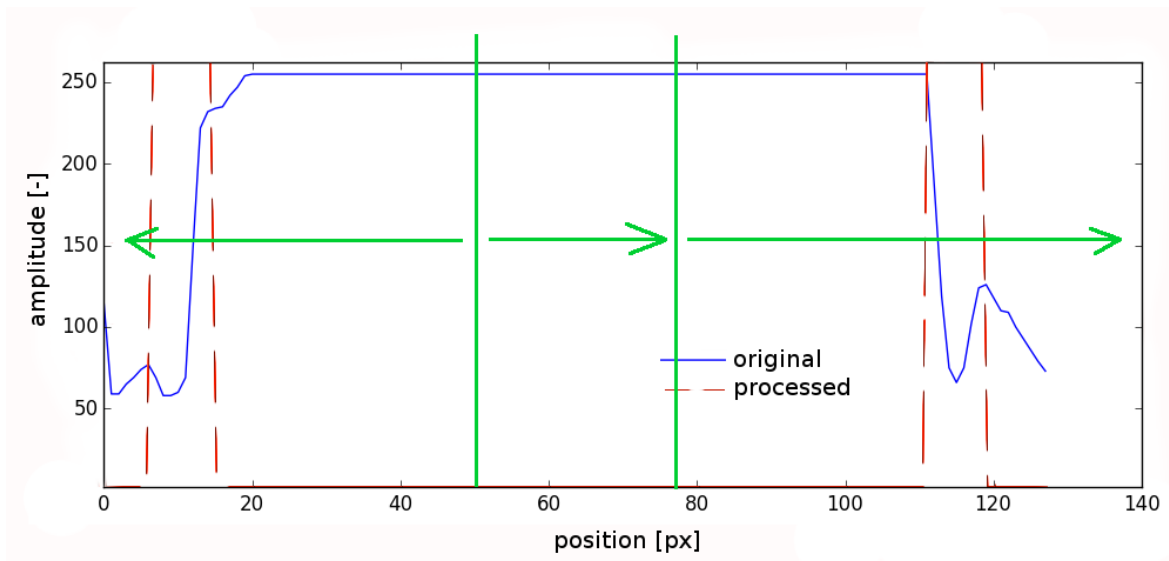
Předzpracovaný obraz je segmentován. Ta začíná rozdělením obrazu na tři části v bodech 50 a 78, což při šířce vektoru 128px znamená, že krajní oblasti jsou široké 50px a prostřední 28px.

Takto rozdělený obraz se poté od středu prohledává a poloha první nalezené hrany v dané oblasti se ukládá. Po prohledání celého vektoru (nebo nalezení hrany v každé oblasti) segmentace končí. Polohy čar jsou poté při rozhodování dále zpracovávány (viz kapitola 5.1)

5.3 Řízení kamery

Pro získání maxima informací z obrazu je nutné jej správně sejmut. Jednou z klíčových vlastností obrazu je saturace. Tu je možné opravit programově, ale za cenu ztráty informace nebo správným řízením kamery obraz saturovat již při snímání. Z hlediska efektivity a obtížnosti je snažší vhodně řídit kameru.

Saturace obrazu je daná dobou osvitů CCD čipu. Ten integruje po celou dobu příchodí náboj a naintegrované napětí se s příchodem startovacího impulsu uloží do výstupního bufferu, odkud je poté sériově po pixelech odesílán do mikrokontroléru.



Obr. 5.5: Směry hledání okrajových čar v obraze

Zároveň prvních 18 hodinových cyklů probíhá reset (vynulování) CCD čipu a po uplynutí této doby se začíná integrovat znovu.

Z této logiky vychází i řízení kamery. Protože doba samotného zpracovávání obrazu je řádově kratší než doba osvitů CCD čipu, lze na časování nahlížet z opačné strany, kdy doba vybavování dat z kamery určuje dobu saturace. Potom lze pomocí úpravy této frekvence upravovat saturaci obrazu.

V řešení je použité dynamické upravování saturace podle posledního snímku. Pokud je průměrná hodnota saturace nižší než práh, snímkovací frekvence se sníží, pokud je vyšší, snímkovací frekvence se zvýší. Prah byl zvolen 200. Při tomto prahu je střed obrazu přesaturovaný a okraje mají amplitudu alespoň 40. To je hodnota, ve které je již vidět nějaká změna. Snímková frekvence se může hýbat od 30 snímků za sekundu po 90 snímků za sekundu. Při rozptýleném denním světle a přisvitu zářivkami je snímková frekvence přibližně 70 snímků za vteřinu. Při umělém zářivkovém světle klesne na 40 snímků za vteřinu, což se pohybuje na hraně použitelnosti.

5.4 Řízení pohonů

Pohony na modelu jsou celkem tři (viz kapitola 4). Dva motory pro jízdu a jeden servomotor pro zatáčení. Pohonné motory jsou řízené přímo pomocí PWM a signálů určující smysl otáčení, servopohon je řízený pomocí PWM, která určuje polohu.

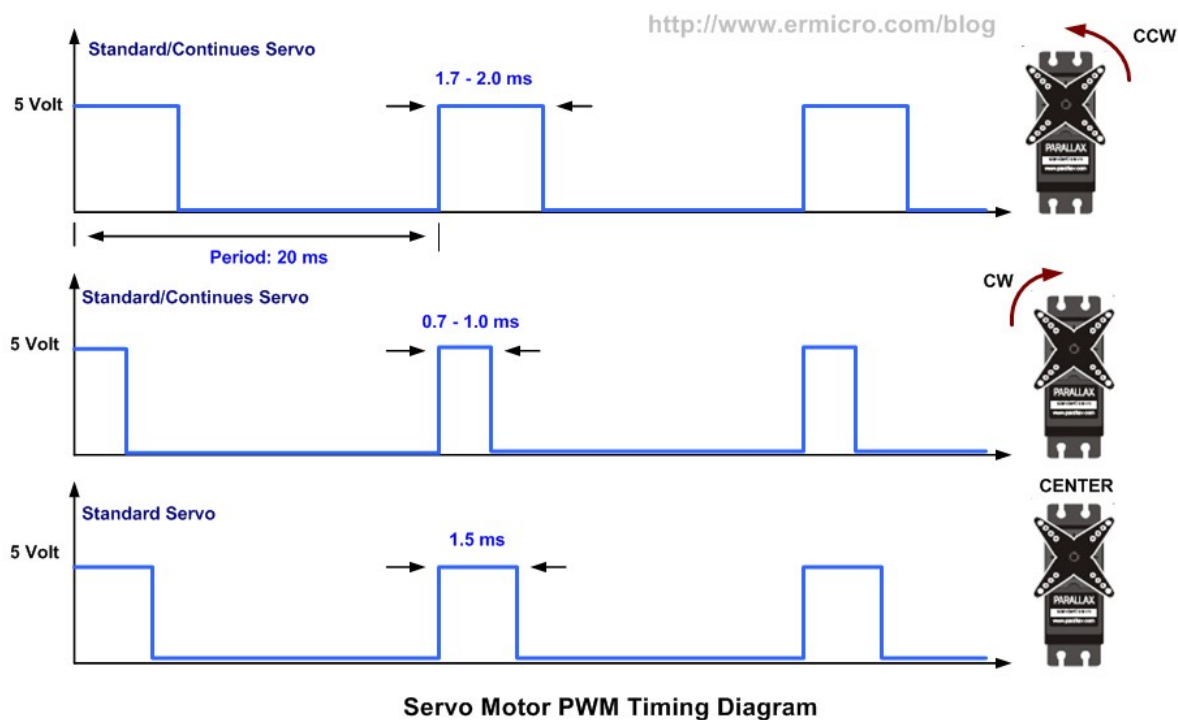
5.4.1 Řízení pohonných motorů

Pohonné motory jsou řízené přímo pomocí PWM. Frekvence PWM byla zvolena 13797 Hz. Tato hodnota byla zvolena na základě snahy odstranit rušení od těchto motorů. Proto byla zvolena hodnota, která není násobkem 50 Hz.

Řízení samotné z hlediska programu je rozložené do několika vrstev. První vrstva funguje jako rozhraní mezi řídicími algoritmy a řídicí logikou motorů. Algoritmus pomocí procedury nastaví požadavek na rychlost. Tento požadavek je po dokončení řídicí heuristiky zpracován. To znamená, že se ověří, zda jeho hodnota nepřesáhla limity a nastaví se příslušné linky pro jízdu vpřed nebo vzad. Poté se spočítá podle aktuální rychlosti a požadované rychlosti akční zásah. Pro regulaci rychlosti byl použit PS regulátor s koeficienty $P=1$ a $S=0,6$. Tyto parametry byly vzhledem k obtížnosti regulace soustavy určeny experimentálně. Tento výsledek je již použit pro samotný výpočet střídavy PWM, která se nastaví na výstup.

5.4.2 Řízení servomotoru

V konstrukci je použitý modelářský servomotor Futaba. Tyto servomotory jsou řízené pomocí PWM, kde už ale PWM určuje přímo polohu serva. Pro logiku serva není klíčová střída, ale délka kladného pulsu, která musí být v rozsahu 1 - 2 ms



Obr. 5.6: Průběh PWM signálu pro řízení servomotoru

(hodnota se mírně liší pro každého výrobce a typ serva, obvykle bývá rozsah menší). Doporučená perioda celého cyklu je 20 ms, což vede na frekvenci 50 Hz, která je použitá i v této konstrukci. Možným řešením by byla i konstantní střída a proměnlivá frekvence. Z hlediska programu a řízení periferie FlexTimer, která generuje na použitém mikrokontroléru PWM bylo snažší použití varianty s pevnou frekvencí a proměnlivou střídou. Toto řešení je vidět na obrázku 5.6 (zdroj [14]).

5.5 Odesílání telemetrie pro ladění algoritmů řízení

Metodika odesílání dat je nastíněna v kapitole 6.3. Zde bude popsána samotná implementace řešení.

Data fyzicky procházejí přes UART, na kterém je připojený bluetooth modul. Data z systému jsou odesílána buď na konci nebo na začátku zpracování, což určuje stav obrazu v kamerovém bufferu (před zpracováním a po zpracování).

Samotné odeslání je prosté překopírovávání požadovaných dat do výstupního bufferu, jejich oddělování oddělovacím znakem, spočítání CRC, které se přidá na konec rámce a obalení výsledku znaky, které určují začátek a konec rámce. Každý rámeček nese na konci před CRC číslo rámce v 8 bitovém formátu. Tato mechanika je zde pouze pro hlídání pořadí a ztrát rámečků na straně PC.

6 TESTOVACÍ DATA

Množina testovacích dat byla získána pomocí vnitřního telemetrického systému během projíždění testovacích vzorků dráhy. Právě tyto vzorky dráhy sloužily jako refernce pro funkčnost algoritmů a heuristik. Přenos dat je realizován pomocí technologie Bluetooth emulující seriovou linku s rychlostí 155200 baud. Tato data jsou následně logována a vizualizována na PC. Formát přenášených dat je popsán v kapitole 6.3.

6.1 Testovací vzorky dráhy

Vhodná množina testovacích vzorků dráhy je ta, která otestuje většinu (ideálně všechny) možnosti. Na základě zkušeností při ladění algoritmů při řešení byly rozpoznány 4 základní vzory:

- tři díly roviny - zatáčka - rovina,
- dva díly roviny - zatáčka - opačná zatáčka - rovina,
- dva díly roviny - zatáčka - stejná zatáčka - rovina,
- rovina - křižovatka - zatáčka.

Tyto varianty je třeba vyzkoušet pro jejich pravé a levé varianty. Tyto vzory vyzkouší jak rychlost reakce při vysoké rychlosti, tak schopnost regulace jízdy po rovině, průjezdu zatáčkou a rozpoznání zatáčky v různých polohách (z roviny, z křižovatky, ze zatáčky). Díky tomu je možné relativně rychle ověřit funkčnost algoritmů a heuristik. Zároveň tyto testovací vzory nejsou tak náročné na prostor jako celá dráha. Jednotlivé díly jsou popsány na stránkách soutěže [7].

6.2 Metoda získávání testovacích dat

Metoda získávání testovacích dat vznikla během vzniku algoritmů. Předpokladem této metody je dostatek adekvátně osvětleného prostoru. Samotný průběh je následující:

1. umístění modelu na začátek dráhy,
2. spuštění nástroje pro sběr dat (navázání spojení s modelem),
3. vizuální kontrola obrazu a kamery (jsou vidět obě čáry při poloze modelu uprostřed dráhy),
4. spuštění jízdy startovacím tlačítkem,
5. samotná jízda,
6. zachycení modelu na konci segmentu.

Výstupem této metody jsou naměřená data z jízdy daným segmentem. Z těchto dat lze vizualizovat obraz z kamery.

6.3 Popis formátu telemetrických dat

Během testovacích jízd je odesílána telemetrie v pravidelných intervalech. Odesílají se následující data:

- kompletní obraz z kamery (128B),
- spočítaná poloha středu čáry, poloha okrajových čar, (3B),
- data z akcelerometru pro všechny osy (pro každou osu 16bitové číslo), (6B),
- data z odometry (otáčky za vteřinu, průměrovaná data) (4B),
- nastavení motorů (požadovaná rychlost), nastavení serva, (3B),
- proudová zpětná vazba pro oba motory (2B).

Výsledný datový rámec má následující tvar:

BOF	data z kamery	EOP	...	EOP	číslo rámce	CRC CCIT	EOF
-----	---------------	-----	-----	-----	-------------	----------	-----

Datový rámec začíná znakem BOF (beginning of frame) a končí znakem EOF (end of frame). Jednotlivá data jsou od sebe oddělena oddělovacím znakem EOP (end of part). Na konci před znakem EOF je kontrolní součet CRC. Protože je objem přenášených dat stále stejný, není potřeba přenášet informaci o délce rámce. Znak EOF je zde pouze jako kontrola kompletnosti rámce.

Celkový objem dat, který je potřeba přenést je součet všech výše uvedených:

$$S_{data} = S_{cam} + S_{position} + S_{accelerometer} + S_{odometry} + S_{drive} + S_{currentfeedback}$$

$$S_{data} = 128 + 3 + 6 + 4 + 3 + 2 = 146B$$

kde $S_{název}$ jsou velikosti jednotlivých částí.

Celková velikost dat je 146B. Tato velikost je ještě navýšena o řídicí znaky a kontrolní mechanismy.

$$S_{frame} = S_{data} + parts \cdot S_{EOP} + S_{CRC} + S_{COUNT} + S_{BOF} + S_{EOF}$$

$$S_{frame} = 146 + 6 \cdot 1 + 2 + 1 + 1 + 1 = 157B$$

kde S_{frame} je celkový objem dat v jednom rámci, $parts$ je počet částí rámce. Jak je vidět, celková velikost rámce je 157B.

Data jsou odesílána pomocí technologie Bluetooth se standartní rychlostí přenosu 9600baud. Maximální počet rámců odeslaných touto rychlostí je následující:

$$speed_{BT} = \frac{9600b/s}{8} = 1200B/s$$

$$N = \frac{speed_{BT}}{S_{frame}} = \frac{1200}{156} = 7$$

kde N je maximální počet rámců přenesitelný při rychlosti 9600baud. Při této rychlosti by nebylo možné odesílat veškerá požadovaná data.

Při zvýšení rychlosti na 115200 baud je počet přenesitelných rámců následující:

$$speed_{BT115k} = \frac{115200b/s}{10} = 11520B/s$$

$$N = \frac{speed_{BT115k}}{S_{frame}} = \frac{11520}{157} = 73$$

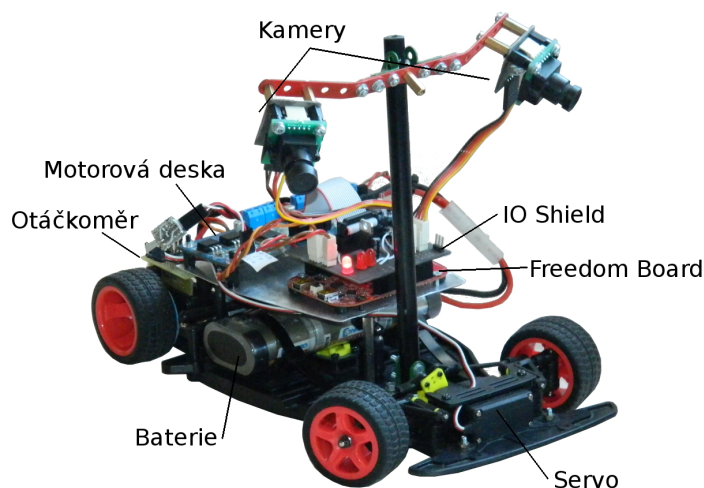
Tato rychlost již stačí pro odeslání všech dat. Reálný objem přenesených dat je maximálně poloviční, protože data jsou odesílána v dávkách a ne trvale.

7 KONCEPČNÍ ŘEŠENÍ

Během vývoje se koncepce řešení několikrát změnila. Důvody byly různé, od chyb v koncepci po zbytečnou složitost řešení. Veškeré změny se týkají výhradně polohy kamery a následného zpracování dat. Důvodem postupných změn bylo vždy objevení problému v současné koncepci, která neumožňovala (s rozumnou složitostí) dokončení. Většina koncepcí končila na problému 'dílu za zatáčkou'. Tento problém se týká rozpoznání dalšího dílu za dílem zatáčky. Průjezd zatáčkou nemá tak vysokou opakovatelnost jako jízda po rovině a z toho plyne naměření různých dat při výjezdu ze zatáčky. Druhým problémem je nízká snímková frekvence kamery a z toho plynoucí problémy s nedostatkem dat. Posledním problémem je nedefinovanost okolí dráhy. Dráha končí vnější hranou černé okrajové pásky a proto je nezbytné dokázat rozpoznat dráhu a okolí. Často totiž vznikají kontrastní vzory mimo dráhu a algoritmus je může vyhodnotit jako validní data.

7.1 Koncepce se dvěma kamerami

Tato koncepce byla zkoumána v rámci předmětu Robotika ve druhém ročníku. Obsahovala dvě kamery na pylonu v první třetině podvozku s tím, že každá kamera sledovala jednu stranu.



Obr. 7.1: Koncepce se dvěma kamerami

7.1.1 Řízení

Řízení bylo proporcionální podle polohy modelu. V této variantě nebylo implementováno žádné rozpoznávání dílů. Zpracování obrazu bylo řešeno dynamickým úrovňovým filtrem na detekci záporných špiček, které byly vyhodnocovány jako čáry. Z hlediska výkonosti a složitosti to bylo jednoduché a rychlé řešení, ovšem s špatnou robustností. Při vyšších rychlostech modelu již algoritmus nestačil a docházelo k častým výpadkům řízení.

7.1.2 Problémy

Jedním ze zásadních problémů byl poměr šířky zorného pole a předsazení pohledu. Z tohoto důvodu byly použity dvě kamery, tudíž se šířka zorného pole teoreticky zdvojnásobila při zachování předsazení pohledu.

Ve skutečnosti kvůli omezení šířky modelu musely být kamery relativně blízko sebe a zorná pole šikmo od směru jízdy. To zaneslo další nelinearitu do řízení a slepé pole doprostřed obrazu.

7.2 Koncepce s jednou kamerou v přední části mimo směr jízdy

Tato koncepce se pokusila omezit pouze na jednu kameru s zachováním koncepčního modelu z předchozí varianty. Jízda byla plánovaná mírně při kraji, sledovala by se jen jedna čára a podle ní by se řídila jízda.

Kamera byla na pylonu v první třetině a byla vyosená o přibližně 20° od směru jízdy.

7.2.1 Řízení

Řízení zůstalo téměř nezměněno od předchozí varianty. Bylo odstraněno řízení podle dvou čar a byly přepracovány jízdní parametry.

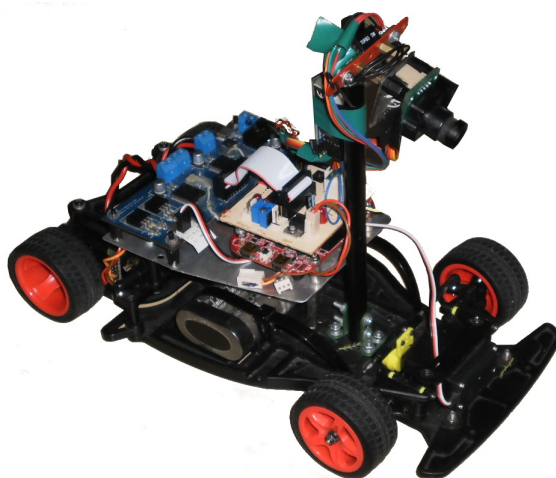
7.2.2 Problémy

Toto řešení v sobě opět neslo nelinearitu v podobě vyosení kamery, ale protože již tato nelinearita nebyla v konfliktu s druhou kamerou, nebyl důvod ji řešit. Problém nastal v případě zatáčky, respektive průjezdu při vnitřní straně. V tomto případě pravidelně docházelo k rychlému výpadku obrazu a následnému vyjetí z dráhy. To souviselo s vysokým přeběhem okrajové čáry a nízkou snímkovou frekvencí kamery. Protože nebylo implementováno rozpoznávání dílů dráhy, bylo od koncepce upuštěno.

7.3 Koncepce s jednou kamerou v přední části ve směru jízdy

Tato koncepce byla původně zamýšlena jako soutěžní koncepce. Vycházela z úspěšných konstrukcí minulých let a zdálo se, že obsahuje všechny klady.

Kamera byla umístěna na pylonu v první třetině (stejně jako tomu bylo v předchozí variantě), ale kamera byla jen jedna a směřovala ve směru jízdy. Pohled byl předsazen od zadní nápravy asi o 90 cm, což umožnilo obsáhnout celou šířku dráhy s jednou kamerou a bez deformace obrazu jako tomu bylo v předchozím případě.



Obr. 7.2: Koncepce s jednou kamerou vpředu

7.3.1 Řízení

První verze vycházely přímo z první koncepce, ale to bylo brzy opuštěno kvůli velkému předsazení obrazu.

Z toho vyplynula nutnost rozpoznávání dílů. Rovina je z hlediska rozpoznávání dílů jednouchý díl a není nutné speciální řízení. Proto bylo aplikováno proporcionální řízení se snahou udržet model uprostřed s minimálním kmitáním. Tento úkol nejlépe splnil P regulátor. Důležité bylo rozpoznat zatáčku, kterou bylo nutné projet po slepu: Vzhledem k faktu, že model detekoval zatáčku předčasně, detekce vycházela z ztráty jedné z okrajových čar po ujetou vzdálenost delší než nejdelší přijatelná doba výpadku obrazu při zákmitech řízení na rovině. Vzhledem k známé vzdálenosti při detekci zatáčky není problém dojet k začátku zatáčky a pak podle definovaného vzoru zatáčku projet.

Podobným způsobem byla řešena i křižovatka, kde se sledovaly obě okrajové čáry. Při jejich výpadku v přibližně stejnou dobu byla vyhodnocena křižovatka. Díl šikany byl řešen jako rovina.

7.3.2 Problémy

Jak bylo předesláno v úvodu kapitoly, problématické místo je detekce dílu za zatáčkou. Velké předsazení pohledu sebou nese zásadní problém. Ve chvíli, kdy model již vyjíždí ze zatáčky, není možné vidět následující zatáčku (nebo pouze její nevýznamný fragment v kraji obrazu). Proto byla snaha zatáčku detekovat dříve. Pokud byl vzor zatáček do 'S', druhá zatáčka se v jednu chvíli objevila v zorném poli. Pokud byl vzor do 'U', nic nebylo vidět po celou dobu a pokud byla rovina, na konci zatáčky se obraz objevil.

Toto řešení ovšem předpokládá definované okolí dráhy. Pokud totiž je kontrastní hrana mimo dráhu, algoritmus ji detekuje a mylně vyhodnotí další díl. Experimentálně byla zjištěna pravděpodobnost správného rozpoznání přibližně 50%.

7.4 Koncepte s jednou kamerou umístěnou v zadní části ve směru jízdy

Toto řešení se snaží o odstranění nedostatku s vysokým předsazením pohledu při zachování šířky zorného pole.



Obr. 7.3: Koncepte s jednou kamerou vzadu

Kamera je umístěna nad motorovým blokem na pylonu a směřuje ve směru jízdy. Pohled je předsazený přibližně o 70 cm od zadní nápravy.

7.4.1 Řízení

Z hlediska řízení se toto řešení nijak neliší od předchozí koncepce. Byla doplněna konstrukce pro rozpoznávání dalšího dílu za zatáčkou na základě statistiky polohy okrajových čar, ale výsledky nebyly průkazné.

7.4.2 Problémy

Naděje, že posunutí zorného pole vyřeší problém s rozpoznání dílu za zatáčkou se potvrdila. Zatáčka je sice vidět po celou dobu, ale kvůli špatné opakovatelnosti průjezdu zatáčkou není možné spolehlivě rozhodnout o dalším dílu. Úspěšnost je vyšší než v předchozím případě (přibližně 70% šance správného rozpoznání zatáček od sebe a asi 50% šance rozpoznání roviny za zatáčkou.) Tato koncepce byla zvolena jako finální i přes její nedostatky.

8 ZÁVĚR

Cílem práce bylo navrhnout a realizovat sebelokalizační systém pro model autonomního vozidla v předem neznámém prostoru a na jeho základě rozhodnout o dalším pohybu. Prvním krokem bylo seznámení se s hardwarovými a senzorickými možnostmi modelu. Na základě těchto znalostí bylo postupně navrženo několik konstrukčních řešení přičemž poslední verze je prezentovaná právě jako výsledek této práce.

Senzorické řešení bylo zvoleno na základě předchozích zkušeností a výsledků ostatních týmů [2]. Jako hlavní senzor pro sledování trasy byla zvolena řádková kamera s rozlišením 128x1 px. Pro měření ujeté vzdálenosti, a tudíž určení polohy na trati, byl navržen bezkontaktní optický snímač otáček založený na odrazovém optickém snímači CNY70. Bohužel jeho výsledky jsou zkreslovány rušením od pohonu, takže i po důkladné filtraci je měření ujeté vzdálenosti pouze orientační. Poslední zvoleným senzorem byl 3-osý akcelerometr. Ovšem data z něj nakonec nebyly použity pro řízení. Jejich místo by bylo v detekci kopce nebo retardérů a případně jako zpětná vazba o průjezdu zatáčkou.

Byl navržen systém a metodika sběru dat z jízd. Ten je realizován bezdrátově pomocí technologie Bluetooth a data z něj sloužily pro vizualizaci dat z modelu. Bez tohoto nástroje by bylo ladění algoritmů citelně složitější. Stejně tak nástroj pro zpětnou vizualizaci dat z kamery posloužil pro statickou analýzu jízd pro návrh heuristik pro řízení.

Samotné řídicí algoritmy byly navrženy na základě naměřených dat telemetrie a opakovaných pokusů průjezdu modelu po dráze. Na konci tohoto procesu stojí výsledek práce v podobě implementovaných algoritmů pro řízení založený na heuristikách a pevných předpokladech o pohybu. Samotné řízení je řešeno pomocí rozpoznávání dílů dráhy a na jejich základě přepínání parametrů regulátoru (metoda gain-scheduling). Bez těchto předpokladů by bylo výrazně složitější nebo v některých případech nemožné například určit výchozí polohu. Vzhledem k pevně daným dílům dráhy bylo možné určit jisté opakující se vzory, které posloužili jako výchozí bod pro vznik heuristik.

Implementace byla provedena na mikrokontroléru Freescale K20D50M z rodiny ARM Cortex M4 v jazyce C. Celý program je optimalizován pro úsporu paměti a se snahou minimalizovat použití výpočtů v plovoucí desetinné čárce, protože mikrokontrolér nemá implementovanou FPU. Výsledkem je kód, který do 2 ms rozhodne o dalším pohybu. Celková perioda včetně načítání dat z kamery trvá 15-25 ms. Tuto dobu již nelze zkrátit kvůli nezbytné expozici CCD snímače kamery.

Bylo navrženo řešení, které je schopné dráhu projet. Ovšem se špatnou opakovatelností která je způsobená částečně již starým a opotřebovaným podvozkem, částečně

rušením měření vzdálenosti. Pokud by se zlepšila opakovatelnost díky novému podvozku, celkové výsledky by se citelně zlepšily. Řídicí algoritmy by bylo možné dále optimalizovat a slučovat do méně průchodů. V současné době je algoritmus pro zachování vysoké granuly, která je výhodná pro ladění, rozdělen na jednotlivé filtry. Ale pro zvýšení rychlosti by bylo lepší sloučit některé filtry dohromady. Určitým zlepšením z hlediska metodiky ladění by mohlo být dálkové zastavení modelu a tím zabránit poškození nárazem. Dalším bodem, kde by bylo dobré zlepšení je samotná dráha, která v současné době již plně neodpovídá soutěžní dráze.

LITERATURA

- [1] EVERETT, H. *Sensors for mobile robots: theory and application*. Wellesley, Mass.: A.K. Peters, c1995, xiv, 528 p. ISBN 1568810482.
- [2] MOLČÁNY, Peter. *Systém sledování trasy pro autonomní vozidlo*. Brno, 2012/2013. Dostupné z: <https://www.vutbr.cz/studium/zaverecne-prace?zp_id=66304>. Bakalářská práce. VUT v Brně, UAMT. Vedoucí práce Ing. Petr Petyovský.
- [3] Freescale Cup Overview. FREESCALE SEMICONDUCTOR. *Freescale Community* [online]. 2012 [cit. 2015-01-13]. Dostupné z: <<https://community.freescale.com/docs/DOC-1011>>
- [4] FRDM-K20D50M: Freescale Freedom Development Platform for the Kinetis K20 USB MCUs. FREESCALE SEMICONDUCTOR. *Freescale Semiconductor* [online]. 2004-2014 [cit. 2014-04-10]. Dostupné z: <http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-K20D50M>
- [5] Line Scan Camera Use. *Line Scan Camera Use* [online]. 2012-07-12 [cit. 2014-04-10]. Dostupné z: <<https://community.freescale.com/docs/DOC-1030>>
- [6] Freescale Cup Motor Driver Board. FREESCALE SEMICONDUCTOR. *Freescale Cup Motor Driver Board* [online]. 2012-07-12 [cit. 2014-04-10]. Dostupné z: <<https://community.freescale.com/docs/DOC-1059>>
- [7] Freescale Cup Race Track Details. FREESCALE SEMICONDUCTOR. *Freescale Community* [online]. 2012-07-17, 2013-12-12 [cit. 2014-04-10]. Dostupné z: <<https://community.freescale.com/docs/DOC-1092>>
- [8] The Freescale Cup EMEA Rules 2015 v1.1. FREESCALE SEMICONDUCTOR. *Freescale Community* [online]. 2014 [cit. 2014-12-22]. Dostupné z: <<https://community.freescale.com/docs/DOC-101287>>
- [9] MMA8451Q 3-Axis, 14-bit/8-bit Digital Accelerometer. FREESCALE SEMICONDUCTOR. *Freescale* [online]. 2014 [cit. 2015-01-13]. Dostupné z: <http://cache.freescale.com/files/sensors/doc/data_sheet/MMA8451Q.pdf>
- [10] MC33931: 5.0 A Throttle Control H-Bridge. FREESCALE. *Freescale* [online]. 10/2012 [cit. 2014-05-01]. Dostupné z: <http://cache.freescale.com/files/analog/doc/data_sheet/MC33931.pdf>

- [11] DIY Camera Mounts. FREESCALE SEMICONDUCTOR. *Freescale Community* [online]. 2012 [cit. 2014-12-22]. Dostupné z: <<https://community.freescale.com/docs/D0C-1018>>
- [12] 2014 Freescale Cup Global Rules. FREESCALE SEMICONDUCTOR. *Freescale Community* [online]. 2013-12-04, 2013-12-13 [cit. 2014-04-10]. Dostupné z: <<https://community.freescale.com/docs/D0C-98806>>
- [13] Datasheet CNY70. *Vishay* [online]. 2014 [cit. 2014-12-22]. Dostupné z: <<http://www.vishay.com/docs/83751/cny70.pdf>>
- [14] Servo Motor PWM Timing Diagram. *Ermicro* [online]. 2015 [cit. 2015-05-12]. Dostupné z: <http://www.ermicro.com/blog/wp-content/uploads/2009/02/servo_01.jpg>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CCD Charge-coupled Device

V Volt

A Amper

Ah amper hodina

Hz Hertz

TFC The Freescale Cup

AD Analog-Digital Covertor - Analogově-Digitální převodník

Shield Obchodní název přídatných desek pro Arduino. Typickým znakem jsou 4 konektory pro připojení k Arduino.

PWM Pulse-Width modulation - Pulzně šířková modulace.

PIT Periodic Interrupt Timer - Periodický přerušovací časovač, modul procesoru

H High - Vysoká logická úroveň, Logická 1

L Low - Nízká logická úroveň, Logická 0

BOF Begining of frame - začátek datového rámce

EOF End of frame - konec datového rámce

EOP End of part - konec datové oblasti, oddělovač dat

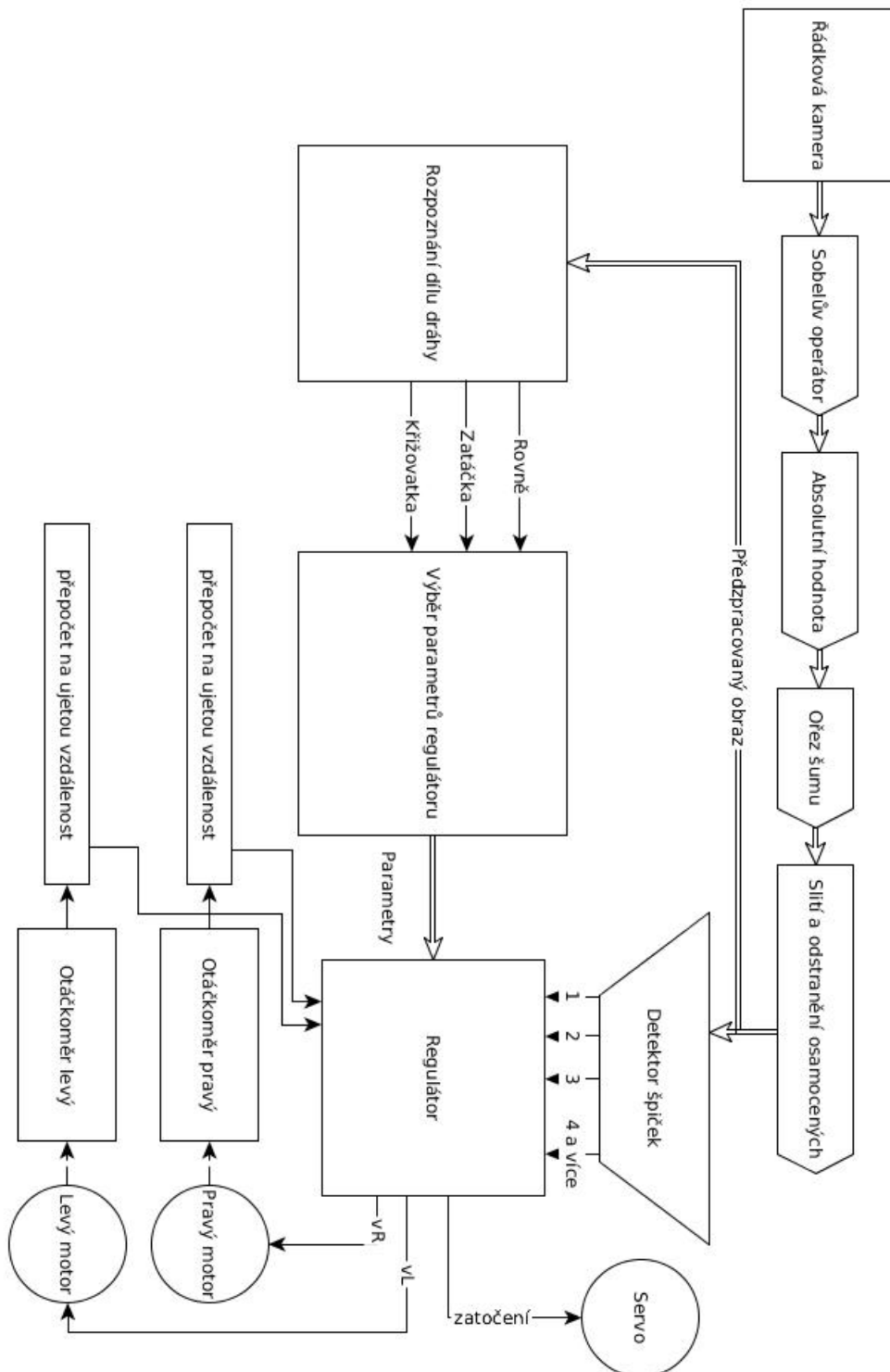
CRC Cyclical redundancy check - kontrolní součet

NOT Negace logické hodnoty

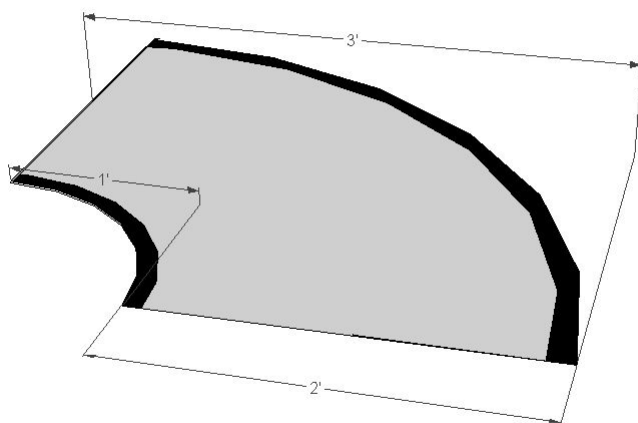
SEZNAM PŘÍLOH

A Řídicí struktura modelu vozidla	46
B Dráha	47
C Přílohy na DVD	50
D Dokumentace k ladicím nástrojům	51
D.1 Návod k použití nástroje dbg_backend	51
D.2 Návod k použití nástroje parser	52

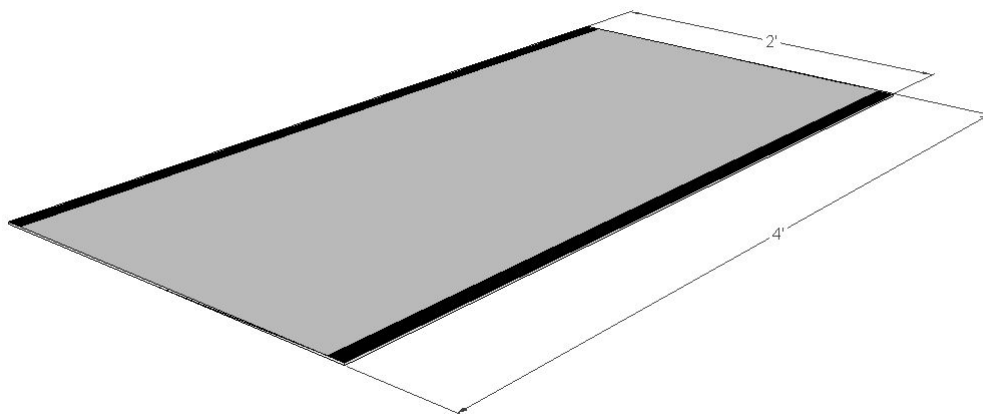
A ŘÍDICÍ STRUKTURA MODELU VOZIDLA



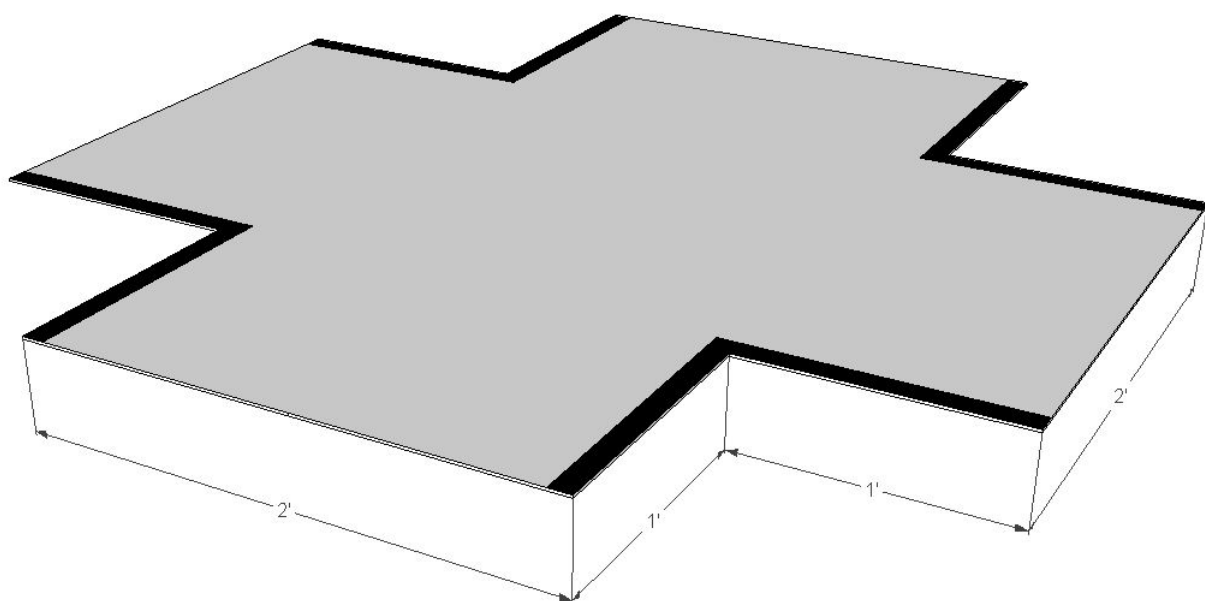
B DRÁHA



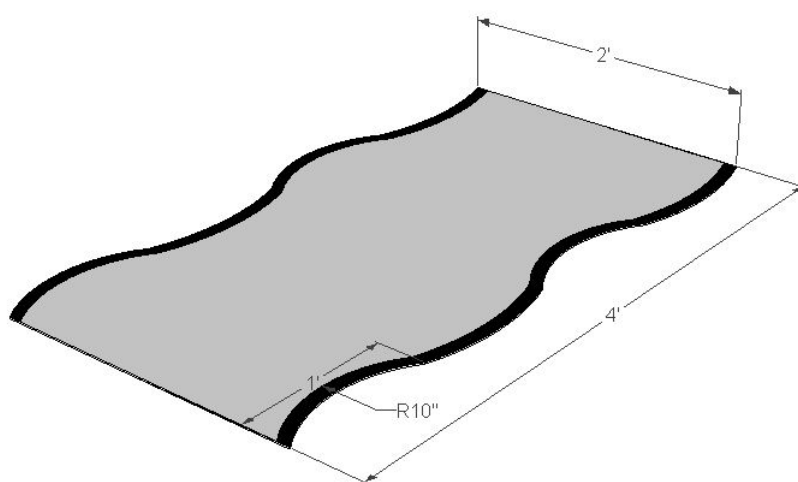
Obr. B.1: Díl zatáčka



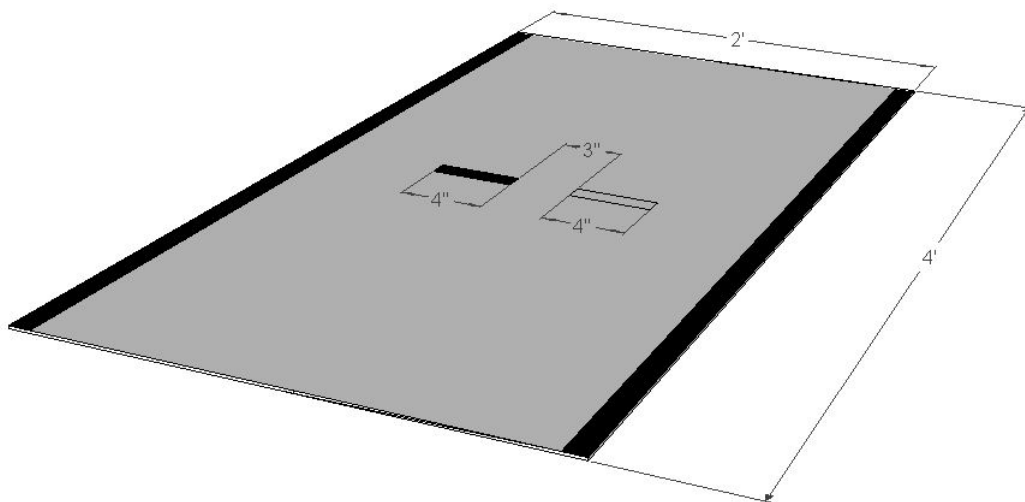
Obr. B.2: Díl rovina



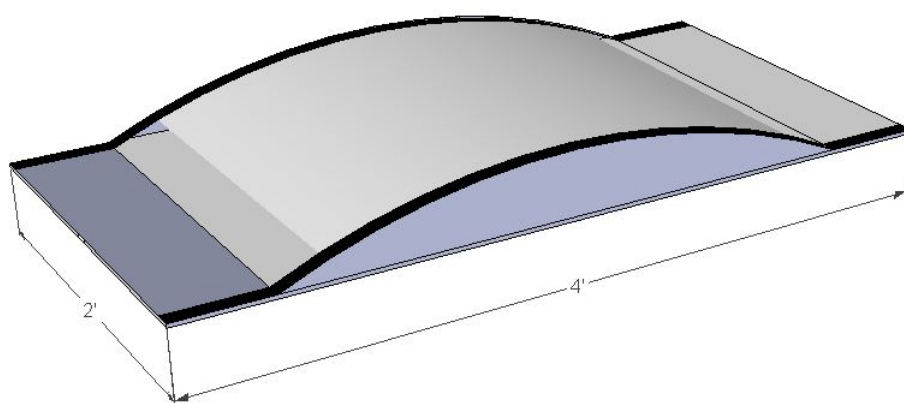
Obr. B.3: Díl křížovatka



Obr. B.4: Díl šikany



Obr. B.5: Díl start



Obr. B.6: Díl kopec

C PŘÍLOHY NA DVD

- bakalarska_prace_vojtech_vladyka.pdf - Tato práce
- PCB/ - Zdrojové soubory designu přídatné desky (shieldu) v EAGLE
- DOCS/FRDM-K20D50M_SCH.pdf - Schema Freedom Boardu
- DOCS/FRDM-K20D50MUM.pdf - Manuál k Freedom Boardu
- DOCS/K20P64M50SF0RM.pdf - Datasheet procesoru Kinetis K20
- DOCS/MOTOR_DRIVE_REV_A.pdf - Schema Motor Boardu
- FW/KDS_AROS - projektové soubory pro KDS 2
- SW/ - instalační balík KDS 2 pro OS Debian/Ubuntu
- TOOLS/dbg_backend/ - debugovací nástroj v jazyce Python. Více informací v příloze D.1.
- TOOLS/parser/ - nástroj pro parsování obrazových dat z přijatého logu v jazyce Python. Více informací v příloze D.2.
- FOTO - fotodokumentace

D DOKUMENTACE K LADICÍM NÁSTROJŮM

D.1 Návod k použití nástroje `dbg_backend`

Požadavky na PC:

- PC s OS Linux 2.8 a vyšší
- Python 2.7.x
- knihovny `pyBluez`, `matplotlib`
- `bluetooth` modul podporující Bluetooth 2.0
- běžící grafický server pro zobrazení grafické reprezentace dat

Tato aplikace je složena z několika modulů. Jádrem celé aplikace je modul `BT-Comm.py`, který zajišťuje samotnou komunikaci pomocí knihovny `pyBluez`. V této knihovně je kromě základních komunikačních metod implementovaná i metoda pro zpracovávání přijatých dat a jejich ukládání do pole. Tato metoda je zároveň schopna data ukládat do souboru pro další analýzu. Data jsou v čitelné podobě ve formátu `txt`.

Celá aplikace je navržena jako terminálová s rozšířením pro GUI pro zobrazování obrazu a průběhů akcelerometru. Jsou implementovány tyto příkazy:

- `d` - Zobraz data
- `l0` - Nastav zobrazovanou úroveň na 0 (normální obraz)
- `l1` - Nastav zobrazovanou úroveň na 1 (obraz zpracovaný hranovým detektorem)
- `e` - Zobraz statistiky chyb
- `sd` - Začni ukládat data
- `cd` - Konec ukládání dat
- `q` - Konec

Příkazy `l0` a `l1` způsobí odeslání příkazu do mikrokontroléru a ten přepne odesílaná data na normální nebo zpracovaná.

Pro spuštění zadejte v adresáři aplikace tento příkaz:

```
python main.py
```

D.2 Návod k použití nástroje parser

Požadavky na PC:

- PC s OS Linux 2.8 a vyšší
- Python 2.7.x
- knihovny PIL
- běžící grafický server pro zobrazení grafické reprezentace dat

Tato aplikace zkonvertuje přijatá obrazová data která byla uložena do logu do grafické podoby. Aplikace je plně automatická a nevyžaduje žádnou obsluhu ze strany uživatele. Vstupní data jsou vložena v podobě cesty k souboru. Výsledný obrázek v formátu bmp je uložený na stejné místo jako log pod stejným názvem.

Pro spuštění zadejte v adresáři aplikace tento příkaz:

```
python main.py NAZEV_LOGU.txt
```