



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MODEL VOZIDLA ŘÍZENÝ POČÍTAČEM

COMPUTER CONTROLLED MODEL OF VEHICLE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ DEINGRUBER

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2019

Zadání bakalářské práce



21973

Student: **Deingruber Ondřej**
Program: Informační technologie
Název: **Model vozidla řízený počítačem**
Computer Controlled Model of Vehicle
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte způsob řízení modelářských serv a elektrických motorů pro modely počítačem a jejich možné využití v modelu vozidla.
2. Vyberte vhodný typ vozidla využívajícího modelářská serva a motory pro modely a zvolte způsob jeho ovládání počítačem, přitom též vyberte vhodný (mikro)počítač a zvažte jeho další využití v modelu.
3. Popište a diskutujte možnosti vybraného modelu a možnosti jeho dálkového ovládání mobilním systémem, například telefonem, přes vhodné rozhraní.
4. Navržený model sestavte, implementujte jeho ovládání a demonstруйте jeho vlastnosti a možnosti.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 14. května 2019

Abstrakt

Tato bakalářská práce se zabývá řízením modelu vozidla pomocí modelářských serv a motorů. Cílem je navrhnout a sestrojít funkční model a na něm demonstrovat principy jeho řízení. Jako rozšíření bylo implementováno detekování vozidel a jejich soubojový systém. Bylo sestrojeno pásové vozidlo s dvěma stejnosměrnými motory a modelářským servem. Pro jejich ovládání byla zvolena kombinace jednodeskového počítače a mikrokontroléru. Pro detekci byla využita knihovna OpenCV v kombinaci s kamerou infračervenými diodami.

Abstract

This bachelor's thesis covers the topic of controlling models using servos and motors. The target was to design and construct a functional model and use it to demonstrate the principles of controlling models by computer. The work was extended by implementing vehicle detection program used for combat of the vehicles. A vehicle moving on tracks, controlled by two DC motors and one servo was constructed. For the control of the vehicle was selected a combination of single-board computer and microcontroller. For the purposes of detecting the vehicle OpenCV library paired with a camera and IR LEDs was used.

Klíčová slova

řízení modelu, modelářská serva, motory pro modely, mikropočítač, detekce objektu, OpenCV, infračervené diody

Keywords

model control, servos for models, motors for models, microcomputer, object detection, OpenCV, IR LEDs

Citace

DEINGRUBER, Ondřej. *Model vozidla řízený počítačem*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík

Model vozidla řízený počítačem

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Deingruber
22. května 2019

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce prof. Dr. Ing. Pavlu Zemčikovi za cenné rady, připomínky a vedení práce. Dále bych chtěl poděkovat své rodině za podporu a Antonínu Hurníkovi za rady při mechanické konstrukci modelu.

Obsah

1 Úvod	3
2 Shrnutí dosavadního stavu	4
2.1 Motory pro modely, modelářská serva a jejich ovládání	4
2.2 Detekce objektů	16
2.3 Dálkové ovládání	21
2.4 Výpočetní technika pro vestavěné systémy	23
3 Zhodnocení současného stavu a plán práce	27
3.1 Zhodnocení dosavadního stavu	27
3.2 Specifikace požadavků	27
3.3 Plán práce	28
4 Popis vlastní práce	29
4.1 Základní koncepce	29
4.2 Celkový popis	29
4.3 Platforma	30
4.4 Komunikační protokoly	37
4.5 Využití aplikace	38
4.6 Ověření funkčnosti a interpretace výsledků	40
5 Závěr práce	41
Literatura	42

Seznam obrázků

2.1	Stejnoseměrný motor	5
2.2	Schéma stejnosměrného motoru	6
2.3	Stator stejnosměrného motoru	7
2.4	Schéma ovládnání motoru 1	8
2.5	Schéma ovládnání motoru 2	8
2.6	H-můstek	9
2.7	Střídavý motor	10
2.8	Stator střídavého motoru	11
2.9	Klecová kotva střídavého motoru	12
2.10	Modelářské servo	13
2.11	Rozložené servo	14
2.12	Řídící signál serva	14
2.13	Analogový řídící obvod serva	15
2.14	RGB model	17
2.15	HSV model	18
2.16	Camera obscura	19
2.17	Geometrie modelu camera obscura	20
2.18	Kalmanův filtr	21
2.19	Komunikační protokol vysílačů	22
2.20	Mikrokontroler Intel 8049	23
2.21	Arduino UNO	24
2.22	Jednodeskový počítač	25
2.23	Raspberry PI 3B+	26
4.1	Sestrojený model tanku	30
4.2	Zjednodušené schéma elektroniky	31
4.3	Arduino Motor Shield	32
4.4	LED ovladač PCA9685	32
4.5	Binární filtrování	33
4.6	Detekce hran	34
4.7	Detekce vozidla	36

Kapitola 1

Úvod

Modely vozidel často využívají motory a modelářská serva. Ty jsou většinou řízeny jednoduchými analogovými nebo digitálními obvody. S vývojem počítačů – jejich miniaturizace, klesající spotřeby a rostoucího výkonu je čím dál tím výhodnější je využít pro daný účel. To umožňuje sestavení modelů, které mají funkcionalitu, kterou dříve nemohly disponovat. Modely se tak mohou autonomně pohybovat, pokročile zpracovávat informace, komunikovat s ostatními vozidly pomocí síťových protokolů nebo být dálkově ovládány přes internet. Práce je zaměřená na ovládání vozidla pomocí mikropočítačů a detekci objektů.

Na základě mého návrhu technického řešení byl sestaven model tanku, který je možné využít pro souboje pomocí detekčního systému. Pro pohon vozidla je využito 2 stejnosměrných motorů a modelářské servo. Detekci zajišťuje kamera detekující infračervené diody na vozidle, pomocí kterých zjišťuje jeho polohu metodou problému perspektivy n bodů. Pro ovládání je využito jednodeskového počítače a mikrokontroléru. Vozidlo je ovládáno pomocí síťové aplikace. Zde byly navrženy vlastní komunikační protokoly pro jeho ovládání. Bylo prakticky ověřeno, že sestavený model splňuje navržené vlastnosti.

Motivací pro tvorbu práce byla demonstrace praktického využití počítačů pro kontrolu modelů a možnosti které se nabízí při jejich využití.

Práce obsahuje informace potřebné pro pochopení principů ovládání vozidel pomocí počítačů a její rozšíření v podobě detekce ostatních vozidel za účelem soubojového systému. Ve shrnutí současného stavu je popsána konstrukce a řízení motorů pro modely a modelářských serv, počítače vhodné pro daný účel, základy zpracování obrazu a prostorové transformace využití v modelu. Popis vlastní práce obsahuje popis konstrukce vozidla a jeho technické možnosti. Popsány jsou funkční celky, využití aplikace, algoritmy a získané výsledky.

Kapitola 2

Shrnutí dosavadního stavu

Následující kapitola obsahuje shrnutí poznatků potřebných pro pochopení práce, nutně nemusí obsahovat všechny poznatky v dané problematice a nemá encyklopedický charakter, což je omezeno i maximálním možným obsahem práce. Je rozdělena do 4 podkategorií, do kterých práce zasahuje - modelářská serva, detekce objektů, komunikace a výpočetní technika pro vestavěné systémy.

2.1 Motory pro modely, modelářská serva a jejich ovládání

Následující kapitola se zabývá modelářskými servy, motory a způsoby jejich ovládaní pro účely tvorby dálkově ovládaných systémů.

Elektrické motory

Elektrické motory jsou zařízení, které přeměňují elektrickou energii na energii rotační. Většina elektrických motorů využívá pro svou funkci interakci mezi magnetickým a elektrickým polem. Vzniklé síly poté otáčí hřídel motoru. Elektrické motory mohou být napájeny stejnosměrným, nebo střídavým proudem, podle čehož také motory můžeme rozdělit do dvou základních kategorií, stejnosměrné a střídavé motory [15].

Stejnoseměrné motory

Je motor napájený stejnosměrným proudem. V této kapitole budou popsány klasické kartáčové motory. Stejnoseměrný motor využívá stejnosměrného proudu k tvorbě magnetické síly, která otáčí hřídel.



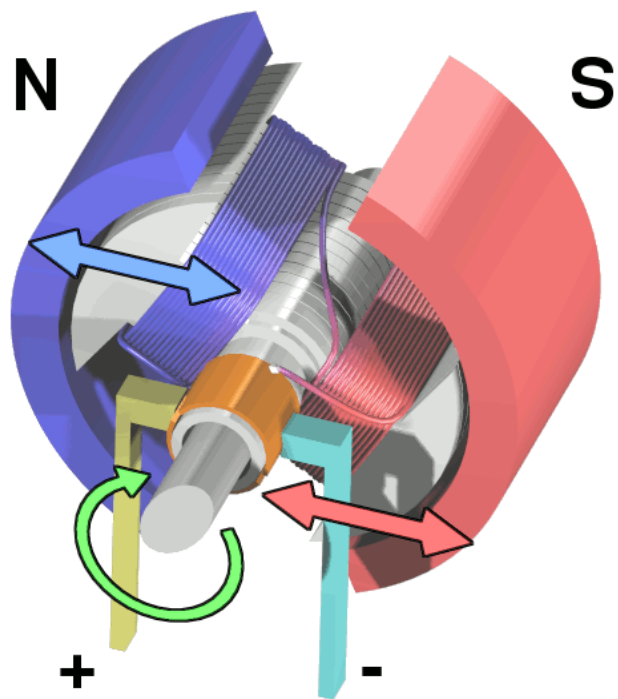
Obrázek 2.1: příklad stejnosměrného motoru^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:DC_Motor.jpg

Proud prochází přes dvě, nebo více cívek, které jsou připevněny na hřídel motoru a otáčejí se s ní. Tato soustava se nazývá rotor. Magnetická síla vytvořená cívkami je zesílena magneticky měkkou ocelí nebo ocelí s vysokým obsahem křemíku a působí na pole vytvořené stálými magnety uspořádanými okolo rotoru v pevné soustavě nazývané stator [15].

K cívkám je elektrický proud veden přes pár kartáčů, většinou vyrobených z grafitových sloučenin. Pružiny přitlačují kartáče na válec, který je rozdělený na části připojené na cívky. Válec je označován jako komutátor a funguje jako jednoduchý mechanický přepínač.

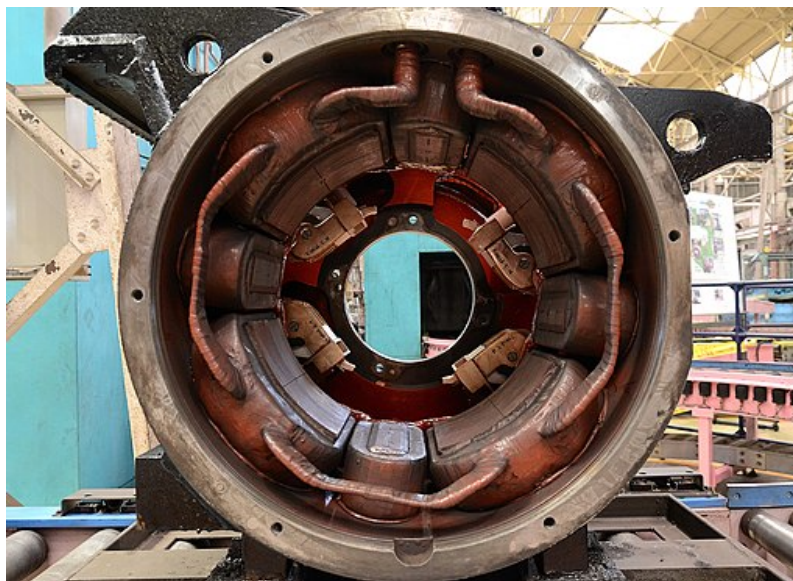
Pokud je na cívky přivedeno napětí, vytvoří magnetické pole. Díky komutátoru jsou vždy u sebe magnetické póly se stejnou polaritou, což způsobí pohyb motoru.



Obrázek 2.2: Schéma stejnosměrného motoru^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:Electric_motor_cycle_1.png

V praxi mají stejnosměrné motory tři nebo více cívek, které umožňují plynulejší provoz [14].



Obrázek 2.3: Cívky ve statoru stejnosměrném motoru ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:MT61_DC_motor_stator_and_brush.JPG

Pro ovládání stejnosměrných motorů je potřeba zajistit kontrolu směru otáčení a otáček. Pro správný směr otáčení je potřeba zajistit správnou polaritu napájecího napětí připojeného na svorky motoru. Pro kontrolu otáček máme více možností, rychlost otáčení motoru je možné ovládat kontrolou zdrojového napětí, změnou magnetického toku nebo změnou napětí cívek [13].

Budeme vycházet z rovnice rovnováhy napětí

$$U = U_i + R \times I$$

kde (U = svorkové napětí, U_i = indukované napětí při zatížení, R = odpor motoru, I = proud motoru, kotvy) a točivého momentu

$$M_t = K \times \Phi \times I_k$$

kde (M_t = točivý moment, K = konstanta motoru, Φ = magnetický tok buzení, I_k = proud kotvy)

Jako kotva je označena část motoru, ve které se indukuje elektrické napětí – vykonává práci. Ve stejnosměrném motoru se nejčastěji jedná o rotor.

Otáčkovou charakteristiku je možno vyjádřit následujícími způsoby:

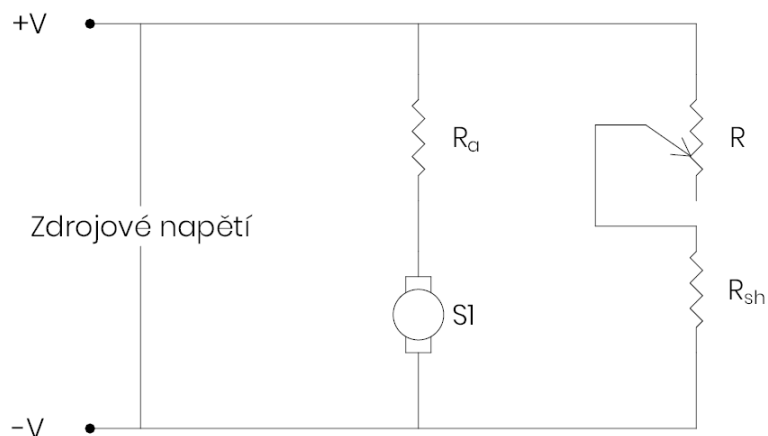
$$\begin{aligned} n &= f(I) - \text{jako závislost otáček na proudu} \\ n &= f(M) - \text{jako závislost otáček na momentu} \end{aligned} \quad (2.1)$$

Pro ustálený stav poté platí

$$n = (U - R_k \times I_k) / (K \times \Phi) \quad (2.2)$$

Z rovnice 2.2 vyplývá, že otáčky motoru jsou přímo úměrné zdrojovému napětí, nepřímo úměrné poklesu napětí v kotvě a nepřímo úměrné magnetickému toku procházejícím kotvou.

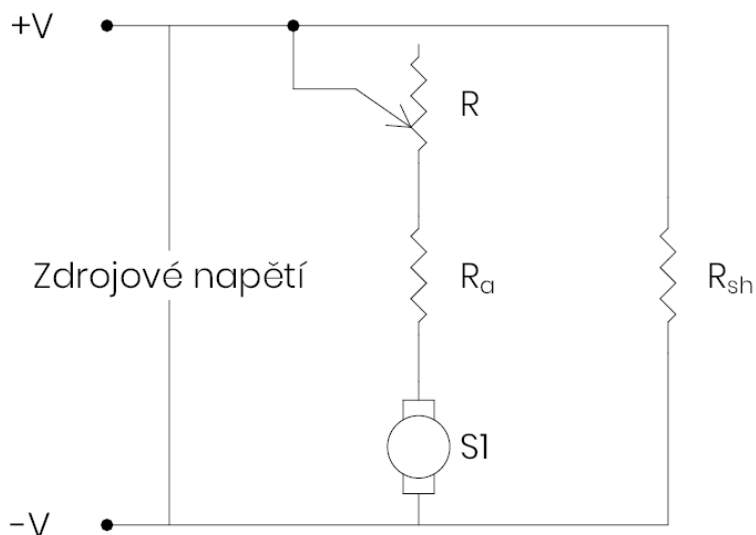
Magnetický tok je možné omezit omezením proudu proudícího přes cívku, pomocí paralelně umístěného rezistoru a potenciometru s vinutím rotoru viz obr. 2.4. Při zvýšení odporu potenciometru se sníží proud procházející přes vinutí. To způsobí zvýšení otáček.



Obrázek 2.4: Kontrola magnetického toku ^a

^apřevzato z: vlastní

Při kontrole napětí cívky rotoru je potenciometr sériově zapojený s motorem viz obr. 2.5. Při zvyšování odporu potenciometru se snižuje napětí na cívkách a tím se snižuje rychlost otáčení motoru.



Obrázek 2.5: Kontrola napětí cívky ^a

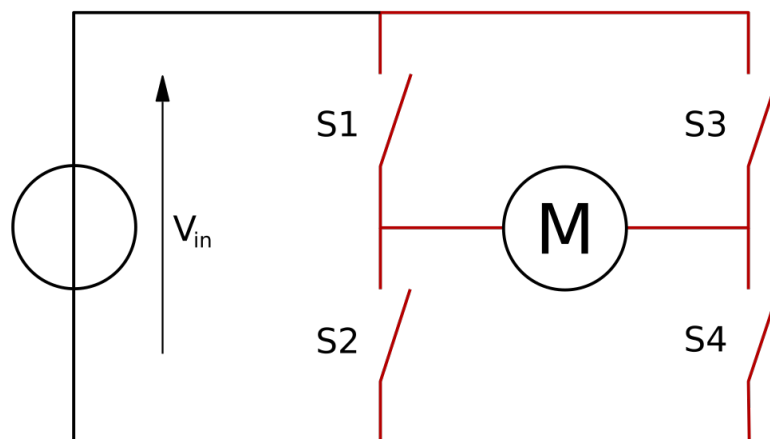
^apřevzato z: vlastní

Obě předchozí metody nemohou poskytnout kontrolu rychlosti motoru v úplném rozsahu. Navíc kontrolou magnetického toku je možno narušit komutaci a kontrolou odporu

rotoru vznikají velké energetické ztráty na rezistorech. Proto je nejčastěji využíváno jiné metody, kontroly otáček motoru, kdy kontrolujeme dodávané napětí.

Nejčastější metoda kontroly napětí je pomocí pulzní modulace. Motor je zapínán a vypínán v určitém poměru, tak aby simuloval různé napětí. Střída pulzně modulovaného signálu přibližně odpovídá poměru napětí signálu a efektivního napětí na motoru.

Pro ovládání směru otáčení je často využíváno H-můstku. H-můstek je elektronický obvod, který přepíná polaritu napětí přivedeného na motor. Je odvozen od typické grafické reprezentace takového obvodu viz obr. 2.6. Obsahuje 4 přepínače (mechanické nebo elektronické). Pokud je přepínač 1 a 4 uzavřen a přepínač 2 a 3 zavřen na motor bude přivedeno kladné napětí na motor. Pokud jsou uzavřeny přepínače 2 a 3 a otevřeny přepínače 1 a 4 je na motor přivedeno obrácené napětí, způsobující rotaci v opačném směru, než v předchozím případě. H-můstek lze zároveň využít jako brzdu a to přivedením stejného napětí na oba terminály motoru. Toho je možné docílit otevřením přepínačů 1 a 2 a uzavřením přepínačů 3 a 4 nebo uzavřením přepínačů 1 a 2 a uzavřením přepínačů 3 a 4 [14].



Obrázek 2.6: schéma H-můstku ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:H_bridge.svg

Střídavé motory

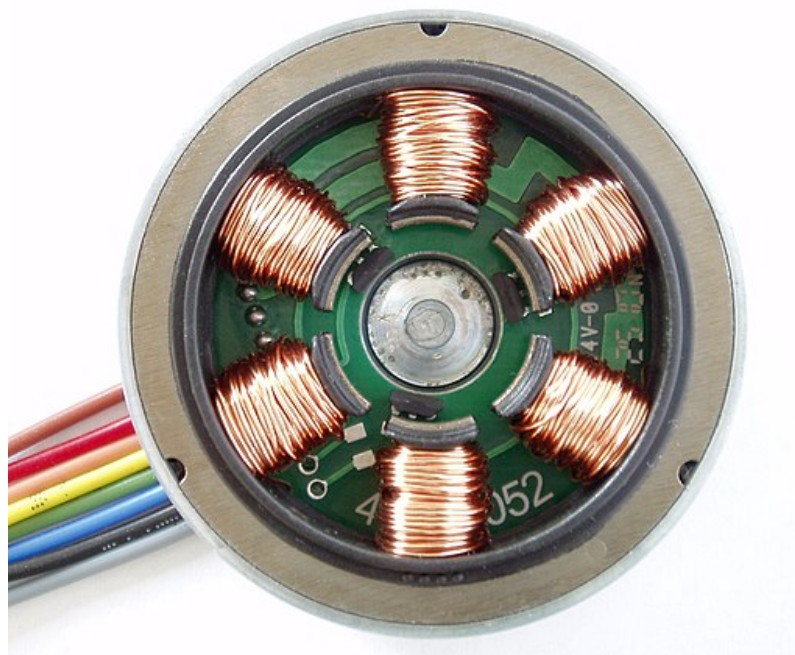
Jsou motory napájené střídavým proudem, to jak střídavým harmonickým tak pulzně modulovaným.



Obrázek 2.7: střídavý motor používaný v modelech ^a

^apřevzato z: vlastní

Motor se skládá ze dvou základních částí, rotoru a statoru. Rotor je pohybující se část, která předává mechanickou práci z motoru. Rotor obsahuje elektrické vodiče a kondenzátory, nebo permanentní magnety, které interagují s magnetickým polem statoru, čímž generují sílu otáčející motorem. Rotor většinou nebývá napojen na zbytek elektrického motoru a je napájen pouze indukčními proudy. Stator je pevná část motoru obsahující obvykle buď cívky, nebo permanentní magnety. Ty generují točivé magnetické pole otáčející rotorem. Stator můžeme vidět na obr. 2.8 [13][14].

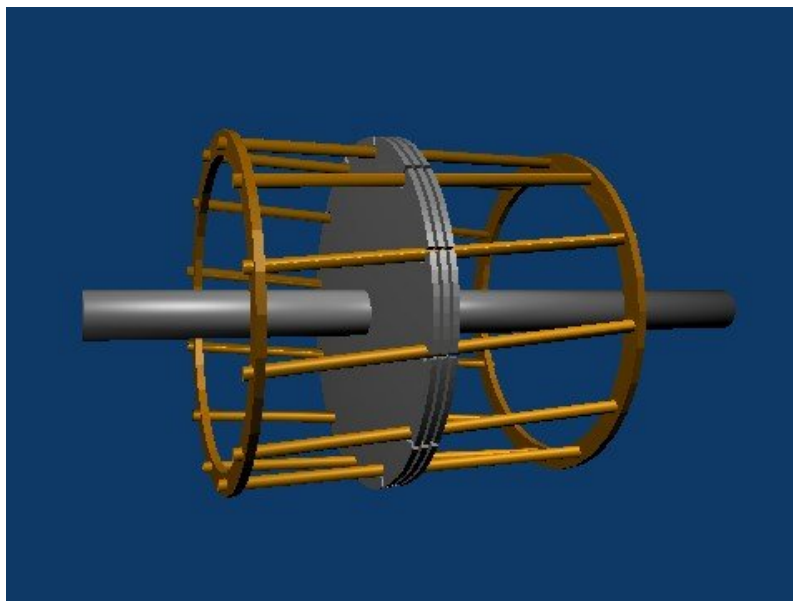


Obrázek 2.8: stator střídavého motoru ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:Stator_Winding_of_a_BLDC_Motor.jpg

Střídavé motory je možno rozdělit do dvou základních kategorií, synchronní a asynchronní.

U synchronních motorů se rotor snaží zachovat stejnou rychlost otáčení jako vektor magnetické indukce točivého pole, až do kritického momentu. Jejich výhody jsou konstantní otáčky při stabilní frekvenci napájení, hlavní nevýhody poté potřeba motor rozběhnout na provozní otáčky pomocí jiného stroje nebo pomocí asynchronního rozběhového vinutí [15].



Obrázek 2.9: Klecová kotva střídavého motoru ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:Squirrel_cage.jpg

Asynchronní motor se od synchronního liší především funkcí rotoru. Ten využívá pro své napájení elektromagnetickou indukci, proto ho lze označit za motor indukční. V rotoru umístěném v točivém poli motoru je indukován proud, který vyvolá magnetický tok. Ten interaguje s točivým magnetickým polem, což způsobuje otáčení rotoru. Kotva viz obr. 2.9 je buď klecová nebo kroužková. Bez zatížení je rychlost otáčení dána frekvencí otáčení magnetického pole.

$$n_s = \frac{f \times 60}{p} [\text{min}^{-1}]$$

kde f je frekvence otáčení magnetického pole a p počet pólových dvojic statoru.

Při zatížení je ale rychlost otáčení nižší než frekvence otáčení magnetického pole, to nazýváme skluz.

$$s = \frac{n_s - n}{n_s} \times 100 [\%]$$

kde n_s jsou synchronní otáčky magnetického pole a n mechanické otáčky rotoru.

Poté můžeme rychlost otáčení vyjádřit jako

$$n = \frac{f \times 60}{p} (1 - s) [\text{min}^{-1}]$$

kde f je frekvence otáčení točivého pole, s skluz a p počet pólových dvojic statoru [15].

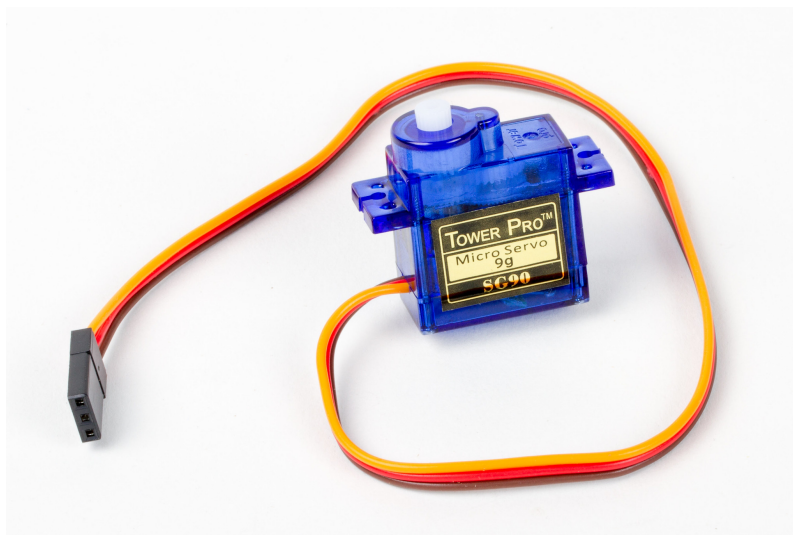
Pro řízení otáček střídavých motorů se používají dva hlavní přístupy, metody ovládání pomocí zpětné vazby a metody založené na modelech motorů [10].

Modelářská serva

Modelářská serva jsou motory opatřené vhodnými ovládacími prvky, které umožňují precizně ovládat natočení rotoru. Oproti ostatním motorům nejsou serva navržena pro volné

otáčení, ale jako zařízení pro nastavení otočení. Typicky jsou omezena mezi 120 až 360° rotace. Pro účely této bakalářské práce se zaměřím především na serva využívané především v modelech a robotech [14][13].

Skládá se z motoru, převodovky a řídicí elektroniky. Všechny prvky jsou většinou v kompaktním plastovém pouzdře. Běžné modelářské servo je na obrázku 2.10.



Obrázek 2.10: Modelářské servo ^a

^apřevzato z: <https://www.flickr.com/photos/oskay/22731304279>

Servo vykovává příkazy z externího ovladače, příkazy obsahují požadovaný úhel natočení. Po obdržení příkazu servo rychle změní úhel natočení a bude se snažit udržet svou polohu i přes působení externí otáčecí síly po celou dobu vysílání daného příkazu.

Pro modelářská serva je využíváno především stejnosměrných kartáčových i bezkartáčových motorů.

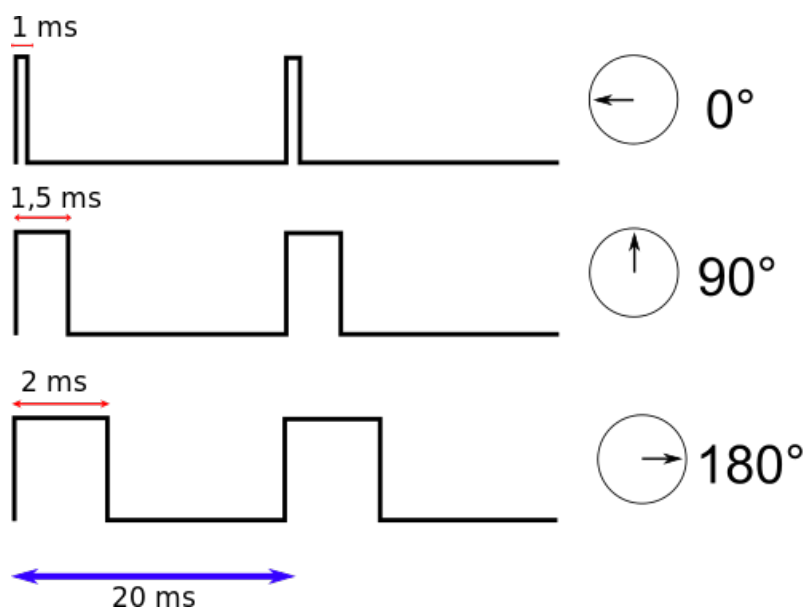
Pro vysoké otáčky a nízký kroutivý moment je většina elektrických motorů nevhodná pro využití v servomotorech, proto bývají převodovány převodovkami, které výrazně zlepšují jejich charakteristiky pro dané využití. Převodovky jsou konstruovány jak z plastů např. nylonu, tak i kovů a slitin pro vyšší pevnost. Pro většinu aplikací jsou nylonové převody dostačující a dokonce nabízí některé výhody, jako menší hmotnost, která pozitivně zlepšuje odezvu serva. Pro vyšší zatížení je, ale lepší využít serva kovové. Převodovka servomotoru viz obr. 2.11.



Obrázek 2.11: Rozložené servo ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:Exploded_Servo.jpg

Modelářská serva jsou řízena pomocí pulzně modulovaného signálu. Ten je typicky šířkově pulzně modulovaný s frekvencí 50 Hz. Šířka pulzu je 1 ms to 2 ms a určuje natočení serva. Šířka pulzu 1 ms znamená minimální natočení serva a 2 ms maximální natočení. Pulzy s šířkou mezi 1 ms a 2 ms jsou polohy mezi krajními polohami. Typický řídicí signál je na obrázku 2.12 [14].



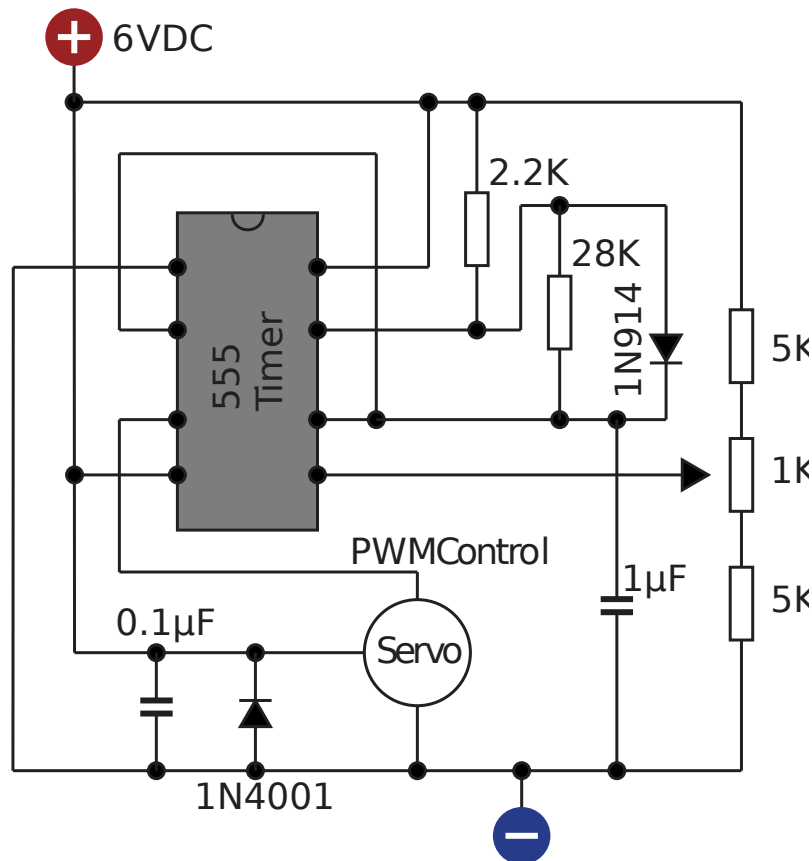
Obrázek 2.12: Řídicí signál serva ^a

^apřevzato z: <https://commons.wikimedia.org/w/index.php?title=File:TiemposServo.svg&oldid=156842428>

Řídicí část lze rozdělit na dva druhy, analogovou a digitální. Jsou rozlišovány podle druhu řídicí elektroniky. Analogová kontrolní část obsahuje potenciometr, časovač a kon-

denzátor. Digitální řízení obsahuje mikrokontrolér, který dokáže zlepšit některé charakteristiky serva. Například krouťivý moment při malých změnách výchylky, díky většímu počtu signálů schopných vyslat motoru.

Na obrázku 2.13 je schéma analogového řídicího obvodu servomotoru. Využívá časovač 555 s elektrickými prvky připojenými tak, aby vytvářely konstantní pulzní signál s frekvencí přibližně 48 Hz (perioda lehce větší než 20 ms). 1 μF kondenzátor připojený do obvodu je nabíjen přes 2.2 k Ω rezistor v sérii zapojený s diodou, kterou obchází 28 k Ω ohm nabíjecí čas kondenzátoru reprezentuje čas, po který je časovač v hodnotě "vysoká". Tento kondenzátor se vybití přes 28 k Ω rezistor, reprezentující stav časovače "nízká". 1 k Ω potenciometr, umístěný uvnitř motoru, zapojený sériově s 5 k Ω rezistory funguje jako dělič napětí. Toto napětí upravuje limity napětí nabití a vybití časovače. Otočení potenciometru prodlužuje, nebo zkracuje čas stavu "vysoká" bez změny frekvence. Minimální a maximální otočení je omezeno rozsahem potenciometru. Jelikož motor sdílí zdroj napětí s časovačem, obsahuje ochrannou diodu a kondenzátor mezi zdrojem napětí pro motor a zemí, pro potlačení šumu a zpětného proudu tvořeného motorem.



Obrázek 2.13: Analogový řídicí obvod serva ^a

^apřevzato z: vlastní

Pro řízení modelářských serv je využito analogových obvodů, mikrokontrolérů nebo počítačů schopných tvořit pulzně modulovaný signál. Pro tvorbu pulzně modulovaného signálu se využívají tři hlavní přístupy, hardwarově pomocí časovačů, DMA (direct memory access, česky přímý přístup do paměti) nebo softwarově pomocí systémových přerušení.

Využití časovačů ale není příliš vhodné, protože drobné rozdíly délek signálu způsobených povahou tvorby systémových přerušení můžou vnášet do signálu chybu způsobující třes serv.

2.2 Detekce objektů

Pro detekci objektů je používáno několik přístupů, využívají se techniky strojového učení nebo čistě algoritmické postupy.

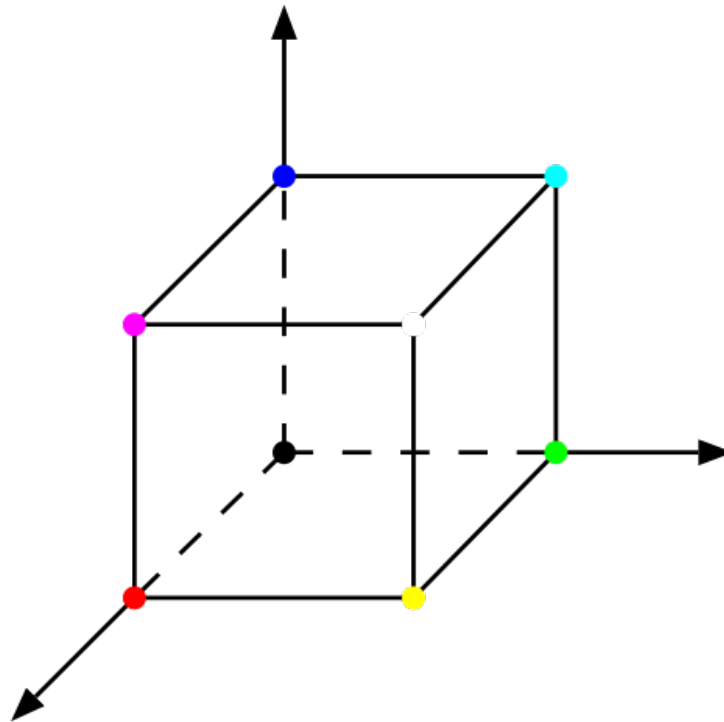
Pro algoritmickou detekci objektů je využíváno filtrování dat, detekce hran a algoritmy popisující geometrické vlastnosti detekovaných ploch.

Reprezentace dat a barevné modely

Data, s kterými se pracuje jsou matice. Každý prvek obsahuje hodnoty reprezentující barvu a jas daného bodu. Nejběžnějšími barevnými reprezentacemi při detekci objektů jsou odstíny šedi, nebo binární obrazy pro černobílé obrazy a RGB nebo HSV pro barevné obrazy.

V barevném modelu odstínů šedi je každý bod reprezentován jednou hodnotou určující intenzitu jasu černé barvy. Hodnoty mívají typicky velikost 8 nebo 16 bitů. Speciálním případem odstínů šedi je binární obraz, obsahující pouze 2 hodnoty, černou a bílou.

V modelu RGB je barevné spektrum rozděleno na 3 základní barvy, které jsou poté míchány pomocí aditivního modelu viz obr. 2.14. Název modelu název modelu pochází z prvních písmen anglického názvu barev červené, zelené a modré (Red, Green, Blue). Pro každou hodnotu se využívá 8, 16 nebo 24 bitů [9].



Obrázek 2.14: RGB model ^a

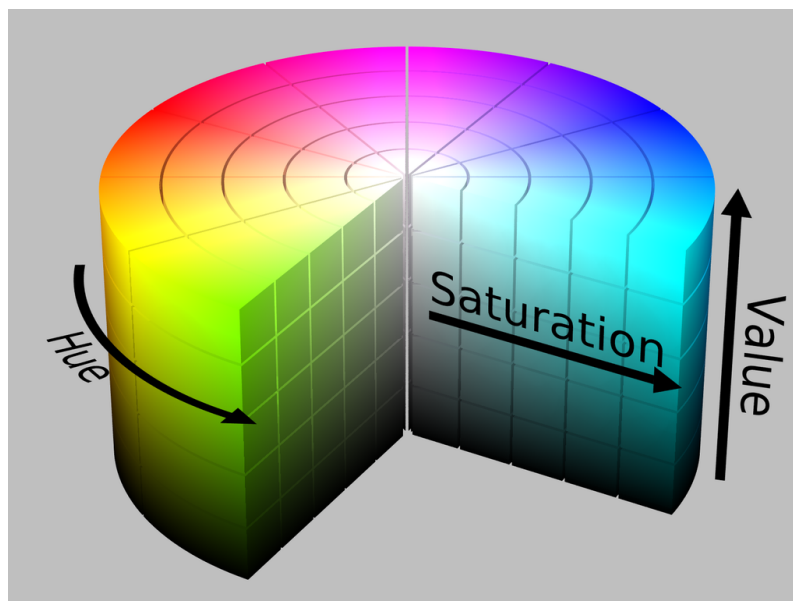
^apřevzato z: https://commons.wikimedia.org/wiki/File:Barevny_model_rgb.svg

Další z modelů je HSV (Hue, Saturation, Value), známý také jako HSB (Hue, Saturation, Brightness), je složený ze tří složek - odstínu, sytosti a intenzity jasu. Vizualizace modelu se nachází na obrázku 2.15.

Odstín určuje barvu bodu, hodnota je od 0 do 360° a je určena podle standardního barevného kruhu.

Sytost představuje množství šedi v poměru k odstínu, je určeno v procentech

Intenzita světlost nebo tmavost barvy



Obrázek 2.15: HSV model ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:HSV_color_solid_cylinder_saturation_gray.png

Barevné modely je možné mezi sebou převádět, následují nejvyužívanější převody modelů pro detekci objektů.

První je převod z modelu RGB na černobílý. Využívá se pro to metoda váženého průměru s experimentálně zjištěnými váhami, které se pohybují okolo 21% pro červenou, 71% pro zelenou a 8% pro modrou.

Pro převod na binární obraz z černobílého obrazu je nejčastěji využíván tresholding (česky prahování). Využívá se také lineární filtrování, to ale nemá takové využití pro zpracování obrazu. Je vytvořen práh, který určuje hraniční hodnotu, do které se bod považuje za černý a od které hodnoty už je považován za bílý. Pro převod z barevných obrazů je obraz nejčastěji buď převeden na černobílý a poté filtrován stejně. Také je možné využít filtrování v určitém rozsahu, jak černobílého obrazu, tak barevného [9].

Detekce shluků

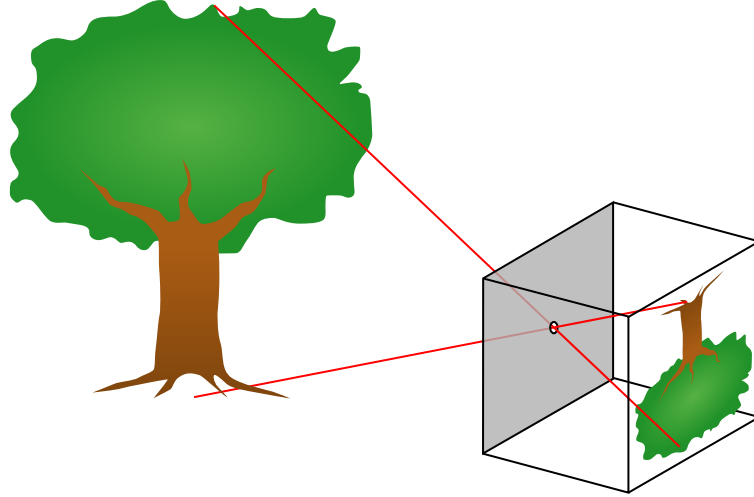
Jako shluk můžeme označit oblast, která má stejné nebo velmi podobné vlastnosti. Detekce shluků jsou metody zaměřené na detekování oblastí, které se odlišují od ostatních vlastnostmi jako jas nebo barva.

Zjištění orientace objektu

Pokud známe body objektu v obrazu a reálnou podobu objektu, řešíme často, jak je objekt orientován a kde ve scéně se nachází. Nastává tak problém perspektivy n bodů. To je problém odhadu polohy kalibrované kamery – 6 os volnosti (tří os rotace a trojrozměrného vektoru posunu), pokud známe 3D polohu bodů a jejich 2D projekci v obrazu [12].

Model Camera obscura

Pro řešení problému je potřeba popsat vlastnosti kamery, abychom znali mapování 3D souřadnicí reálného světa do 2D projekční roviny kamery.



Obrázek 2.16: Camera obscura ^a

^apřevzato z: <https://commons.wikimedia.org/wiki/File:Pinhole-camera.svg>

Nejvyužívanější je model Camera obscura, též dírková komora nebo pinhole. U modelu camera obscura zobrazujeme reálný obraz na plátno pomocí malého otvoru viz 2.16.

Uvažujme bod p v souřadnicích reálného světa $\{W : w_x, w_y, w_z, w_t\}$ a homogenní souřadnice p ve $\{W\}$ jsou $X_W = (w_x, w_y, w_z, 1)^T$ je možno vyjádřit model camera obscura jako rovnicí 2.3 [18].

$$m' = DK_0MX_W = \begin{pmatrix} \frac{f}{dx} & \frac{f \cot \theta}{dx} & u_0 & 0 \\ 0 & \frac{f}{f \sin \theta dx} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} X_W \quad (2.3)$$

kde

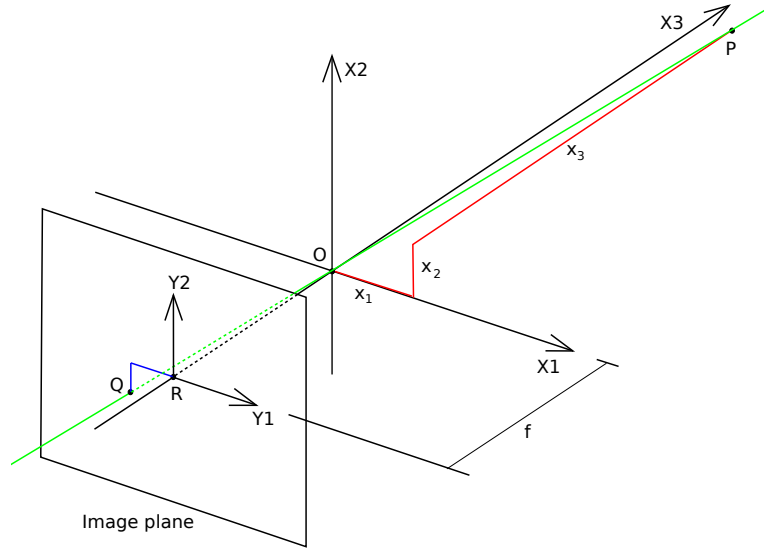
$$K_0 = \begin{pmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, D = \begin{pmatrix} \frac{1}{dx} & -\frac{\cot \theta}{dx} & 0 \\ 0 & \frac{1}{\sin \theta dx} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

a

$$M = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \quad (2.5)$$

Při převodu obrazu kamery na model camera obscura je provedeno následující. Souřadnice $p \in \{W\}$ jsou převedeny na souřadnice systému kamery pomocí matice M . Matice M obsahuje R , matici o rozměrech 3×3 , která reprezentuje otočení kamery a trojrozměrný vektor T , reprezentující její posunutí. Poté matice K_0 promítá bod p do roviny obrazu, kde f je ohnisková vzdálenost kamery a (x_0, y_0) je optický střed roviny. Nakonec D diskretizuje 2D

souřadnice v $\{I\}$ kde (x_0, y_0) reprezentuje velikost pixelu na senzoru kamery s dopadovým úhlem θ [12].



Obrázek 2.17: Geometrie modelu camera obscura ^a

^apřevzato z: <https://en.wikipedia.org/wiki/File:Pinhole.svg>

Matice $K = KD_0$ obsahuje všechny vnitřní parametry kamery, zatímco matice M obsahuje polohu kamery, což jsou vnější parametry. Tyto parametry lze pro kalibrovanou kameru odhadnout pomocí sady 3D bodů a jejich odpovídajících 2D projekcí v obrazu [17].

Problém perspektivy n bodů

Problém je definován následovně: Je-li dána skupina n 3D bodů v jejich reálných souřadnicích, jejich 2D projekce a také vnitřní parametry zkalibrované kamery, zjisti 6 parametrů pózy kamery – její rotaci a posunutí vůči souřadnicím reálného světa. To odpovídá perspektivnímu modelu pro kamery:

$$sp_c = K[R|T]p_w$$

kde $p_w = [x \ y \ z \ 1]^T$ jsou reálné homogenní souřadnice, $p_c = [u \ v \ 1]^T$ odpovídající homogenní souřadnice, K je matice vnitřních parametrů kamery (kde f_x a f_y jsou poměrné ohniskové vzdálenosti, γ je zkreslení, které je někdy považováno za nulové a (u_0, v_0) hlavní bod. s je měřítko pro bod obrazu, R a T jsou požadovanými 3D rotacemi a 3D posunutími kamery (externími parametry), které mají být spočítány. To vede na následující rovnici [18]:

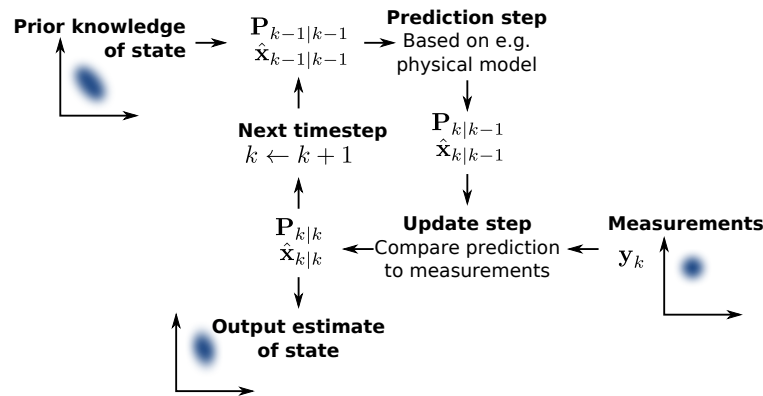
$$s = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.6)$$

Jelikož má kamera až 6 stupňů svobody v reálném světě, je potřeba získat alespoň 3 páry bodů. Většina dostupných řešení je použitelných pro situace, kde jsou známy více než 3 body, ale existují i řešení použitelná pro 3 body [12].

Většina metod počítá s tím, že data s kterými pracují neobsahují šum. Což může způsobit špatné aproximování parametrů. Proto je využíváno metody RANSAC (Random sample consensus, česky shoda náhodných vzorků), iterativní metoda pro odhad parametrů matematického modelu ze souboru pozorovaných dat, které obsahují prvky nepatřící do řešení, které nemají žádnou závislost na hodnotách odhadu [18].

Zpřesnění měření pomocí předchozích měření

Pokud máme řadu měření obsahujících statistický šum a další nepřesnosti, dokážeme je využít k zpřesnění následujícího měření. K řešení tohoto problému je často využíván Kalmanův filtr.



Obrázek 2.18: Kalmanův filtr ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:Basic_concept_of_Kalman_filtering.svg

Kalmanův filtr používá model systému, známé vstupy do systému a řadu po sobě jdoucích měření pro tvorbu odhadu dalšího stavu, který je přesnější než jedno samostatné měření ovlivněné šumem. K výpočtu odhadu stavu systému je využito váženého průměru, kde jsou váhy určeny z kovariance, odhadované míry nepřesnosti současného stavu systému. Výsledkem váženého průměru je nový odhad stavu, který leží mezi očekávaným a změřeným stavem a má lepší odhadovanou nepřesnost než kterýkoliv z nich. Proces využití Kalmanova filtru popisuje obr. 2.18 [16].

2.3 Dálkové ovládání

Kapitola obsahuje různé přístupy k dálkovému ovládání systémů.

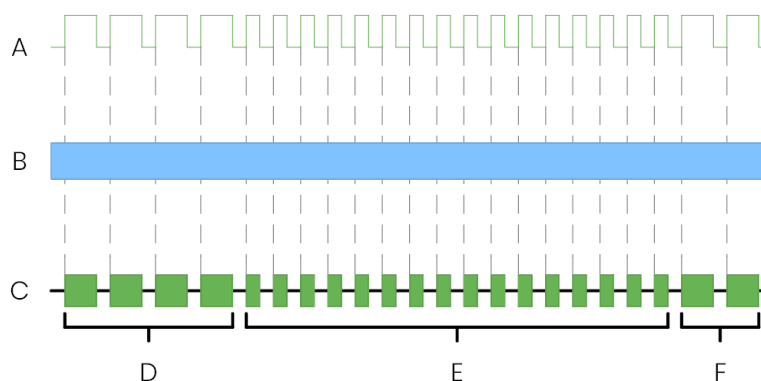
Pro bezdrátovou komunikaci je možné využít široké spektrum signálů především mechanického vlnění a elektromagnetických signálů. Existuje tendence využívat signály, které jsou nerozpoznatelné člověkem, světlo a zvuk.

Méně využívaná možnost je komunikace pomocí zvuku, např. ultrazvuk umožňující řízení zařízení i mimo přímou viditelnost, v praxi je omezeno poměrně vysokým rušením a akustickými jevy.

Nejběžněji využívaný typ komunikace. Je možno ji rozdělit podle způsobu přenosu dat na analogové a digitální.

Analogová komunikace

Běžně se využívají dva základní typy přenosu analogového signálu pomocí elektromagnetického pole, amplitudová a časová modulace. Signál se na vysílači převede do modulované podoby, přenesení se a na přijímači rozkóduje. Signál se interpretuje jako škála hodnot. Přenos je náchylný k vzniku chyb při rušení pásma. Je vhodný pro přenos spojitých signálů, které nemají vysokou náročnost na přesnost jako přenos zvuku, živého vysílání videa. Ale i v následujících oblastech se přechází na digitální vysílání pro vyšší kvalitu.



Obrázek 2.19: Komunikační protokol ^a

^apřevzato z: vlastní

U radiových ovladačů pro většinu hraček se používá především pásmo 27 a 49 MHz a frekvence 72 a 75 MHz pro radiové ovladače více sofistikovaných zařízení jako RC letadla. Signál pro dálkové ovládání může vypadat následovně. Vysílací pásmo se rozděluje do časových intervalů, kde každá část určuje určitou přenášenou veličinu, například obr. 2.19.

Digitální komunikace

Podobně jako v analogové komunikaci se využívá amplitudové a časové modulace. Signál je ale interpretován jako digitální data, to umožňuje implementovat komunikační protokoly, které zajišťují kvalitu přenosu.

Bluetooth

Je technologie pro bezdrátový přenos dat na krátkou vzdálenost mezi dvěma zařízeními na frekvenci 2,400 až 2,485 GHz. Typický dosah je do 10m pro zařízení třídy 2 a vyšší běžně využívané v mobilních telefonech. Je navržena primárně pro zařízení s malou energetickou spotřebou a krátkým dosahem [4].

Wi-Fi

Wi-Fi je bezdrátová technologie založená na standardech IEEE 802.11, které popisují komunikaci v bezdrátových místních sítích (WLAN). Využívá pásem na frekvenci 2,4, 3,65, 5, 5,9 a 60 GHz. K síti se zařízení připojují pomocí přístupových bodů (anglicky access point, AP). Dosah signálu je přibližně 20 m v budovách a více na otevřeném prostranství.

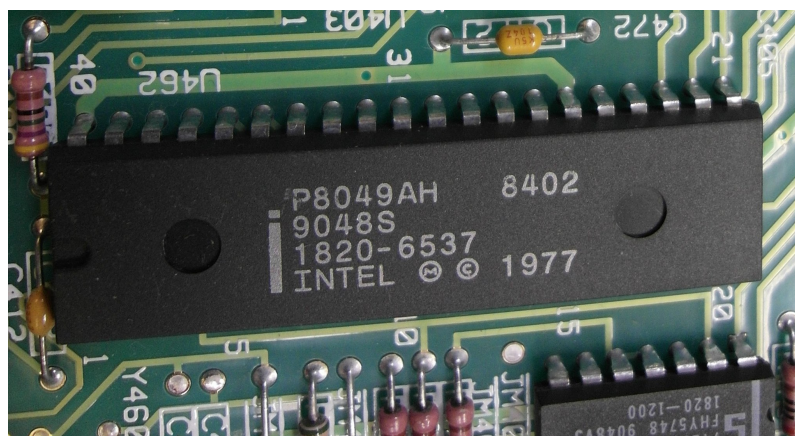
Využívá se především pro připojení zařízení k internetu nebo k vzájemnému bezdrátovému propojení více zařízení. Je koncipována pro rychlý přenos dat a nízké odezvy [1].

2.4 Výpočetní technika pro vestavěné systémy

Pro řízení modelů je nepraktické využívat osobní počítače z důvodu spotřeby elektrické energie a prostorových nároků. Pro účely řízení modelů vozidel počítačem lze využít několik přístupů s různými úrovněmi autonómnosti systému. Je možné využít buď mikrokontroléry nebo jednodeskové počítače. Klíčovými parametry pro daný řídicí systém jsou spotřeba elektrické energie, výpočetní výkon a dostupné výstupní rozhraní.

Využití mikrokontrolérů pro řízení modelů

Mikrokontrolér je malý počítač integrovaný na jeden plošný spoj. Obsahuje jedno nebo více procesorových jader, paměť a programovatelné vstupy a výstupy. Mezi běžné vstupy a výstupy se řadí paralelní, sériové porty, porty komunikačních sběrnic, A/D převodníky, D/A převodníky, PWM výstupy a vstupy pro zachycování času a počítání událostí. Jsou využívány především pro vestavěné systémy jako kalkulačky, vybavení domácnosti nebo počítačové periferie [11].



Obrázek 2.20: Mikrokontroler Intel 8049 ^a

^apřevzato z: https://commons.wikimedia.org/wiki/File:Intel_8049_Microcontroller.jpg

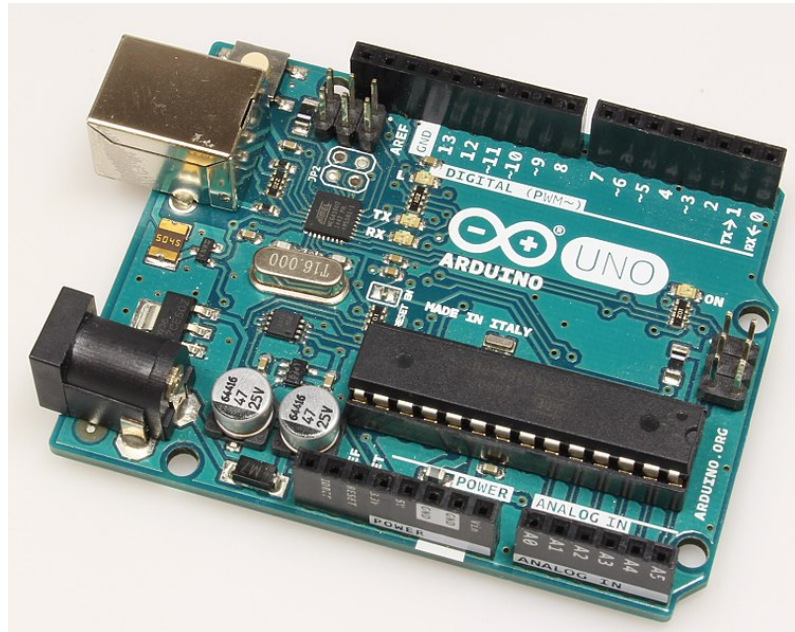
Pro využití ve vestavěných systémech je příhodná jejich velikost, spotřeba elektrické energie a velké množství dostupných vstupů a výstupů.

Často využívané mikrokontroléry pro ovládání modelů

V komunitě modelářů se využívají především mikrokontroléry z rodiny arduino a stm32. Především pro to, že jsou dostupné na prototypovacích deskách. Jsou cenově dostupné a existuje pro ně řada programovacích nástrojů. Dále pokračují příklady populárních mikrokontrolerů.

Arduino UNO je mikrokontrolér z rodiny Arduino. Arduino je open-source (česky s otevřeným zdrojovým kódem) elektronická platforma založená na jednoduše použitelném hard-

ware. Je to nečastěji využívaná deska z rodiny arduino. Obsahuje mikrokontrolér ATmega328P, 8-bitový mikrokontrolér založený na architektuře AVR. Má 14 programovatelných vstupů/výstupů, 6 z nich může být použito jako PWM výstupy, 6 analogových vstupů, 16 MHz křemíkový krystal USB port, napájecí konektor a ICSP konektor [2].



Obrázek 2.21: Arduino UNO ^a

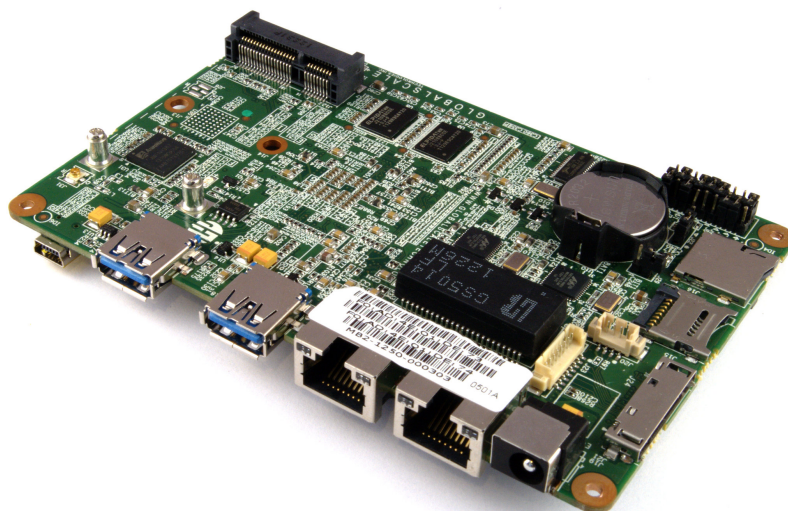
^apřevzato z: https://commons.wikimedia.org/wiki/File:Arduino_uno_r3_isometr.jpg

Pro programování je možné využít vývojové prostředí Arduino IDE a programovací jazyk Arduino, založený na frameworku Wiring.

Maple Mini je jedna z mnoha vývojových desek využívající mikrokontrolér STM32F103CB, který je často využívaná alternativa k deskám založených na mikrokontrolérech Atmel. Ten je postaven na 32-bitové architektuře ARM Cortex-M3 a poskytuje větší výkon než např. Arduino UNO. Deska obsahuje 34 vstupně/výstupních pinů, z toho 12 použitelných jako PWM výstupy a 9 analogových vstupů a USB port. Pro programování lze taktéž využít vývojové prostředí Arduino díky open-source komunitě podporující čipy STM32 [19].

Využití jednodeskových počítačů pro řízení modelů

Jednodeskový počítač je počítač obsahující všechny komponenty potřebující pro svou činnost na jednom plošném spoji. Plošný spoj tak obsahuje minimálně procesor, paměť a vstupy a výstupy. Jednodeskové počítače na rozdíl od osobních počítačů většinou nepoužívají expanzní sloty pro rozšíření nebo periferní funkce [11].



Obrázek 2.22: Jednodeskový počítač ^a

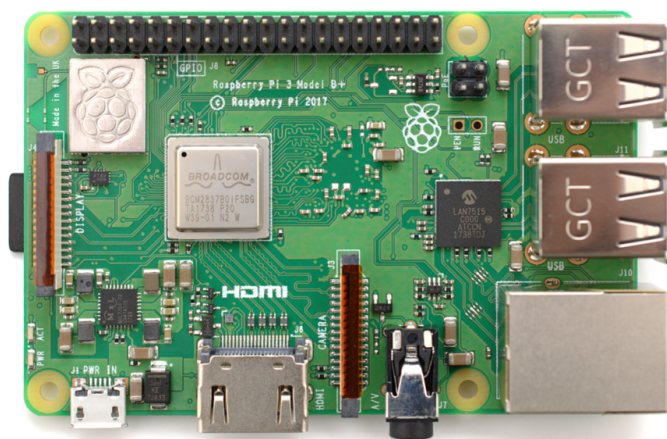
^apřevzato z: <https://www.flickr.com/photos/120586634@N05/14488887400>

Standardem v řízení modelů v modelářském prostředí jsou jednodeskové počítače vyráběné firmou ze Spojeného království Raspberry Pi Foundation. Jednodeskový počítač, původně určený do škol rozvojových zemí [5], daleko přesáhl očekávané využití a používá se dnes i v robotice.

Všechny generace počítačů Raspberry využívají SoC (system on chip, česky systém na čipu) Broadcom s procesorem kompatibilním s ARM architekturou a vestavěným grafickým procesorem. USB vstupy, grafický výstup, slot na SD kartu, zvukový výstup přes 3,5 mm jack a také programovatelné vstupně/výstupní piny.

Raspberry Pi 3B+

Nejnovější revize jednodeskového počítače Raspberry. Obsahuje 64-bitový SoC Broadcom BCM2837B0 s 4 jádry Coretex-A53 taktovanými na frekvenci 1,4 GHz, grafický procesor Broadcom VideoCore IV taktovaný na 250 MHz a 1 GB paměti LPDDR2 SDRAM. Data jsou ukládána na micro SD kartu s kapacitou až 32 GB. Konektivitu zajišťují 4 USB 2.0 porty, konektor RJ-45 a Wi-Fi adaptér podporující standardy b/g/n/ac pro připojení k síti. Video výstup zajišťuje HDMI konektor společně s 3,5 mm jack konektorem pro kompozitní video a DSI portem pro připojení displeje. Konektor jack je možné zároveň využít pro audiovýstup. Pro připojení periférií je možné využít 27 vstupně/výstupních pinů na 40 pinovém konektoru, všechny piny jsou schopny generovat PWM signál, 4 piny pomocí časovače, zbytek pomocí přerušení. Pro napájení lze využít mikro USB konektor [8].



Obrázek 2.23: Raspberry PI 3B+ ^a

^apřevzato z: [https://commons.wikimedia.org/wiki/File:Raspberry_Pi_3_B%2B_\(39906369025\).png](https://commons.wikimedia.org/wiki/File:Raspberry_Pi_3_B%2B_(39906369025).png)

Pro počítač je dostupná linuxová distribuce operačního systému Raspbian, vyvíjená komunitou. Je založená na Debianu. Kromě Raspbianu jsou dostupné další operační systémy jako Ubuntu Mate, Ubuntu Core, Ubuntu Server, Windows 10 IOT Core, OSMC, LibreElec, PiNet, RISC OS, Weather Station nebo IchigoJam RPi.

Kapitola 3

Zhodnocení současného stavu a plán práce

Následující kapitola obsahuje zhodnocení současného stavu techniky pro modely vozidel ovládané počítači, upřesnění specifikace požadavků a plán práce.

3.1 Zhodnocení dosavadního stavu

Pro řízení modelů, využívajících motorů a RC serv, pomocí počítačů můžeme v současnosti využít množství řešení, každé nabízející určité výhody a nevýhody. Můžeme vybírat ze dvou základních druhů počítačů pro kontrolu modelů – mikrokontrolérů a jednodeskových počítačů.

Mikrokontrolér 2.4 je jednoduché a energeticky nenáročné řešení pro kontrolu modelů, nabízející velké množství vstupů a výstupů, včetně PWM a výstupních sběrnic jako I2C pro připojení dalších ovládacích prvků. Další výhodou bývá velké množství rozšiřujících modulů s hotovými ovladači, což zjednodušuje implementaci a vývoj. Hlavní nevýhodou je jejich malý výkon, omezující možnosti aplikací. Nejčastěji jsou využívány pro modely, které nezpracovávají velké množství informací a fungují především jako vykonavatel pokynů vzdáleného řídicího systému. Cena mikrokontrolérů se pohybuje v řádu desítek korun za nejlevější čipy, kde je, ale třeba počítat s návrhem vlastního plošného spoje pro jeho osazení. Ceny nejpoužívanějších vývojových kitů se pohybují v jednotkách stokorun až po hranici tisíc korun [3].

Jednodeskové počítače 2.4 se pohybují na druhé straně spektra. Poměrně výkonné systémy nemají problém se zpracováváním videa v reálném čase nebo prováděním složitějších výpočtů. Jejich hlavním omezením je jejich větší energetická a prostorová náročnost a typicky omezenější možnosti analogových vstupů a výstupů společně s PWM. Cena nejlevnějších jednodeskových počítačů začíná na 150 korunách za nejméně výkonné desky, běžně využívaného Raspberry Pi 3b+ 2.4 se pohybuje okolo 1000 korun [7] a nejdražší jednodeskové počítače stojí tisíce korun [7].

3.2 Specifikace požadavků

Systém pro soubor vozidel byl navržen jako dvě samostatná vozidla, která jsou ovládána bezdrátově po síti pomocí ovládací aplikace. Vozidlo má umožnit detekci kamerou pro účel zjištění zásahu druhého vozidla a přenést danou informaci, aby na zásah mohlo adekvátně

reagovat. Model má být sestaven v měřítku 1:25 a má dosahovat reálné rychlosti vzhledem ke svému měřítku. Model má dosahovat výdrže na jedno nabití minimálně 15 minut.

3.3 Plán práce

Je potřeba zvolit výpočetní techniku pro daný model, to zahrnuje výběr počítače pro ovládní modelu, detekci druhého vozidla, komunikaci s ovladačem a vybrat vhodnou kameru. Pro dané komponenty navrhnout vhodné zapojení a napájení.

Na základě vybraných komponentů je potřeba navrhnout model vozidla schopný pojmout danou elektroniku. Vybrat pohon a další ovládací prvky jako servomotory.

Dalším krokem je sestavení modelu, osazení elektrickými komponenty a testování jeho vlastností.

Je potřeba navrhnout, implementovat a testovat ovládací aplikaci, komunikační protokoly, síťové aplikace a aplikaci pro detekci vozidla a samostatný souboj.

Kapitola 4

Popis vlastní práce

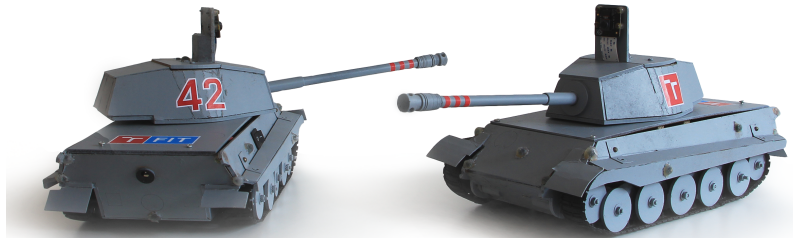
Následná kapitola popisuje vytvořené řešení, využitou techniku, ovládací aplikace, využitý algoritmy a hodnocení daného řešení.

4.1 Základní koncepce

Pásové vozidlo je koncipováno jako model tanku Tiger II. Pásky tanku jsou ovládány samostatnými motory, věž je ovládána pomocí serva, což umožňuje pohyb o 180°. Pro detekci vozidla jsou využity led diody a kamera bez infračerveného filtru. Diody jsou rozmístěny tak, že pokud není žádná část vozidla zakrytá je vidět vždy alespoň 5 diod. Každá dioda vysílá jedinečný kód, kód pro její jednoznačnou identifikaci. Pro ovládání motorů, serva a led diod byl zvolen mikrokontrolér. Ten je ovládán jednodeskovým počítačem, který zajišťuje také přijímání síťových příkazů, detekci vozidla a komunikaci s druhým vozidlem. Detekce je zajištěna pomocí nalezení led diod v obraze, zjištění jejich jedinečných kódů a jejich polohy pomocí problému perspektivy n bodů 2.2. Tank je ovládán pomocí terminálové síťové aplikace.

4.2 Celkový popis

Každý tank je složen z několika podčástí, které společně vykovávají určenou činnost. Obsahuje jednodeskový počítač s kamerou, mikrokontrolér, stejnosměrné motory, servo, generátor PWM signálů, ovladač motorů, baterii, převodníky napětí, modul pro kontrolu nabíjení a vybíjení baterií, 15 infračervených led diod a kontrolních aplikací. Pro pohyb vozidla je využito dvou stejnosměrných motorů, umožňujících pohyb pásů. Pro pohyb mířidel je využito serva otáčejícího věží tanku.



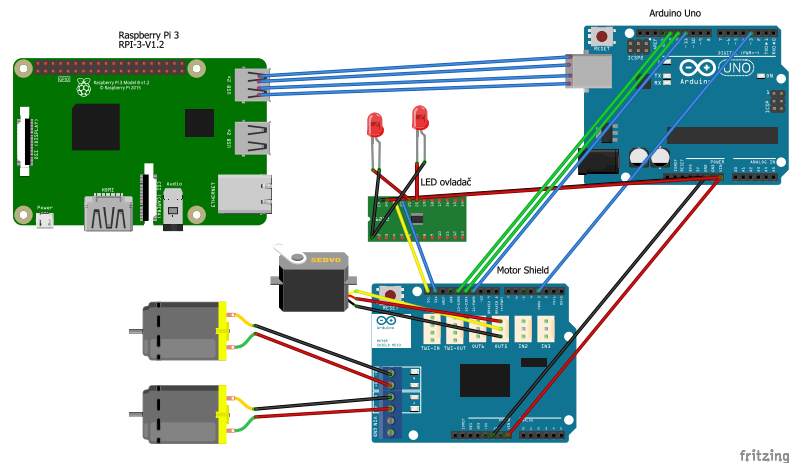
Obrázek 4.1: Sestrojený model tanku a jeho plánované využití pro vzájemné souboje^a

^apřevzato z: vlastní

Motory a serva jsou ovládány mikrokontrolérem, generujícím PWM signál pro serva a h-můstky ovládající motory. Pro účely detekce vozidlo využívá 15 infračervených diod ovládaných mikrokontrolérem, vysílajících unikátní kód. Mikrokontrolér je ovládán po sériové lince vestavěným systémem. Jednodeskový počítač má připojenou kameru bez infračerveného filtru pro detekci infračervených diod. A je ovládán aplikací server, která má na starost komunikaci s ovladačem, ovládání mikrokontroléru, komunikaci s druhým vozidlem a jeho detekci. Ovladač je terminálová aplikace, která po síti ovládá vestavěný systém.

4.3 Platforma

Jako základ platformy byl zvolen model tanku Tiger II, který byl vymodelován a vyřezán z dřevoláknitých desek MDF na laserové řezačce. Jako hlavní pohonná jednotka byly zvoleny 2 stejnosměrné motory s převodovkou. Pro jejich ovládání bylo využito H-můstku L289P na rozšiřujícím modulu Arduino Motor Shield pro mikrokontrolér Arduino UNO využívající portů 3 a 11 na mikrokontroléru pro ovládání rychlosti motorů a portů 12 a 13 pro kontrolu směru otáčení. Pro otáčení věže tanku bylo zvoleno servo s kovovými převody ovládané mikrokontrolérem Arduino UNO. Pro ovládání serva bylo využito portů 6, 5V a GND mikrokontroléru. Pro řízení infračervených led diod byl zvolen mikrokontrolér PCA9685 připojený přes sběrnici I2C k mikrokontroléru Arduino UNO pomocí portů SDA, SCL, 5V a GND. Kamera pro detekci led diod byla vybrána Raspberry Pi NoIR Camera Module V2 s čipem Sony IMX219 CCD připojená přes rozhraní CSI k vestavěnému systému Raspberry Pi 3B+. Vestavěný systém zpracovává zachycený obraz a zároveň zajišťuje síťovou komunikaci pomocí vestavěného Wi-Fi modulu a ovládá přes USB port po sériovém rozhraní mikrokontrolér Arduino UNO.



Obrázek 4.2: Zjednodušené schéma elektroniky ^a

^apřevzato z: vlastní

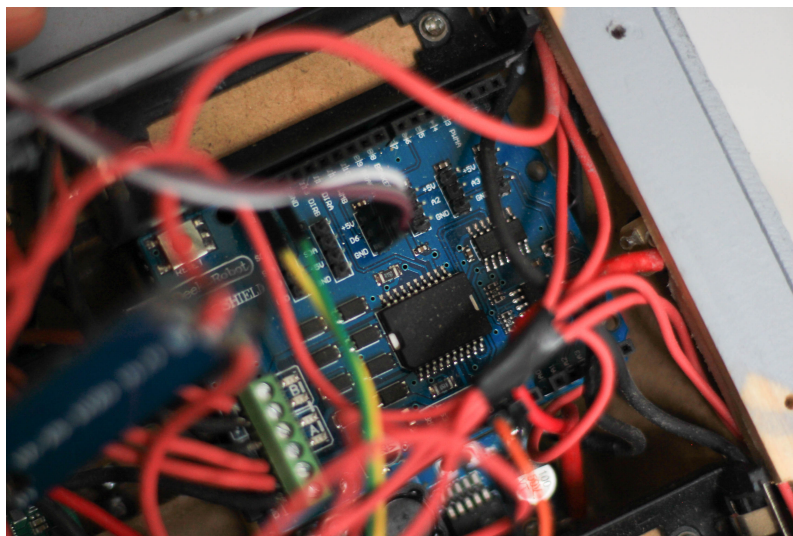
Napájení zajišťují dva bateriové články typu 18650 poskytující v sériovém zapojení nominální napětí 7,4V potřebné k napájení motorů a mikrokontroleru Arduino UNO, vestavěný systém Raspberry PI 3B+ a mikrokontroler PCA9685 je také napájen bateriemi pomocí převodníku napětí snižující napětí na 5V. Nabíjení a vybíjení baterií je chráněno ochranným modulem zabezpečujícím ochranu proti přebití a podbití baterií a ochranu proti zkratu. Pro nabíjení baterií je použit převodník regulující vstupní napětí 9V napájecího konektoru na 8,4V.

Řízení serv

Pro řízení serva bylo využito mikrokontroléru Arduino UNO a vestavěné knihovny Servo, která generuje PWM pulzy ovládající serva.

Řízení motorů

Pro řízení motorů bylo využito mikrokontroléru Arduino UNO a rozšiřujícího modulu Arduino Motor Shield obsahujícího h-můstek, řízený PWM pulzy z mikrokontroléru.



Obrázek 4.3: Arduino Motor Shield v modelu zapojený do mikrokontroléru Arduino UNO^b

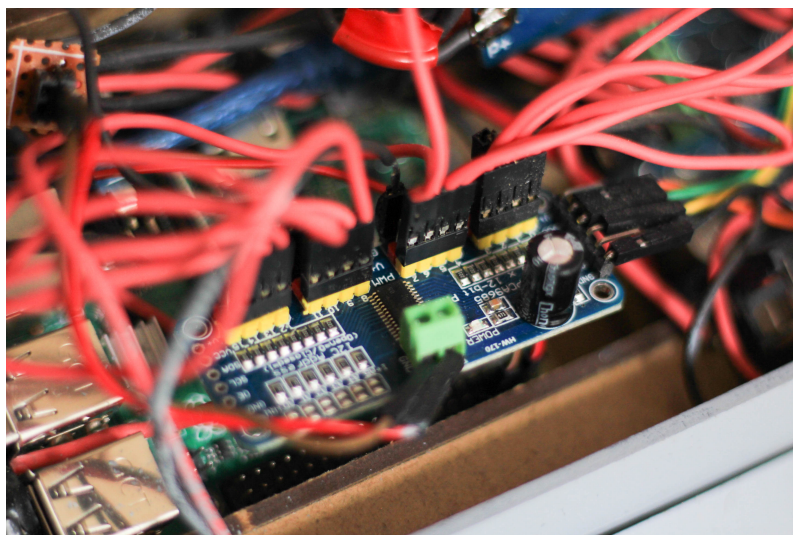
^apřevzato z: vlastní

^bpřevzato z: vlastní

Mikrokontrolér přepočítává přijaté pohybové příkazy na směry a výkony motorů a generuje příslušné signály pro ovládání Arduino Motor Shield.

Ovládání led diod

LED diody jsou ovládány mikročipem PCA9685 připojeným přes sběrnici I2C na mikrokontrolér. Mikročip je schopný generovat až 16 nezávislých PWM výstupů s rozlišením 12 bitů.



Obrázek 4.4: LED ovladač PCA9685 zapojený v modelu ^a

^apřevzato z: vlastní

Mikrokontrolér v pravidelných intervalech aktualizuje střihu PWM signálu, čímž mění jas led diod.

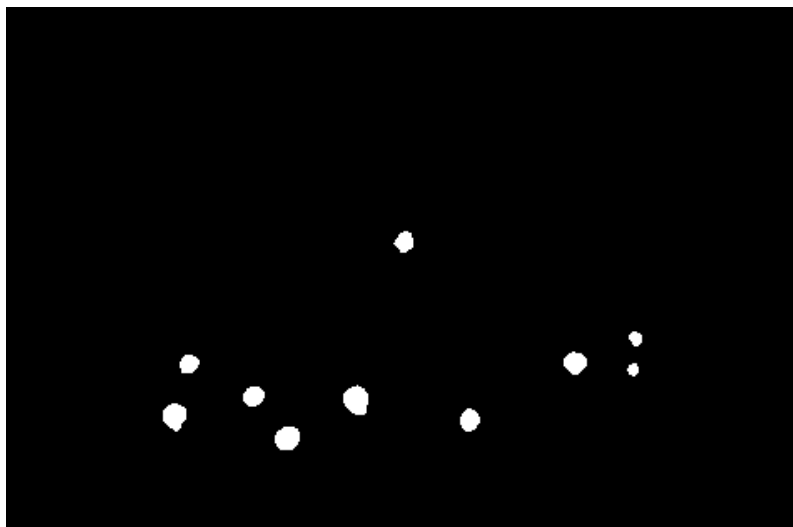
Detekce zásahů

Pro detekci zásahů je využito infračervených led diod umístěných na vozidle. Led diody jsou detekovány a identifikovány jejich jedinečnými kódy pomocí kamery bez infračerveného filtru. Pomocí pozic led diod je prostorovými transformacemi vypočítána poloha vozidla a následně určeno, jestli je výstřel zásahem nebo ne.

Algoritmus detekce led diody

Algoritmus detekce diody můžeme rozdělit do několika částí. Zpracování obrazu pro detekování shluků, detekce shluků, párování detekovaných shluků mezi obrazy a detekce kódu led diody.

Každý snímek je převeden na dichromatický. To je provedeno tak, že se snímek převede do HSV reprezentace a je škálován pro jasné bílé body. Pro hranici škálování byl vybrán libovolný odstín se sytostí maximálně 39 % a jasnem minimálně 0.6.



Obrázek 4.5: Binární filtrování ^a

^apřevzato z: vlastní

Pro detekci hran byl využit algoritmus Suzuki85 implementovaný v knihovně OpenCV. Poté byl pro každý shluk vypočítán jeho moment, pomocí kterého je stanoven jeho střed. Jsou filtrovány shluky příliš malé a příliš velké. To jsou shluky s plochou větší než 1000 pixelů nebo plochou menší než 5 pixelů. Poté jsou vyfiltrovány příliš nekulaté shluky, to jsou shluky s poměrem plochy shluku proti ploše koule s průměrem největšího průměru shluku menší než 0,4.



Obrázek 4.6: Detekce hran ^a

^apřevzato z: vlastní

Středy a plochy jsou porovnány s polohami bodů nalezených led diod v předchozích snímcích. Bod je přiřazen k nejbližší led diodě, vzdálené maximálně 25 pixelů. Diody, které se nepovedlo přiřadit k žádnému shluku jsou smazány, nakonec pro každý nepřirazený shluk je vytvořena nová led dioda.

Pro každou detekovanou diodu je spočítán její kód. Kódy jsou 14 bitové s alespoň 7 přechody 0, 1 nebo 1, 0 s hammingovou vzdáleností 3 pro jakoukoliv vzájemnou rotaci kódu. Takové kódy nám umožňují jejich jednoznačnou identifikaci pro začátek snímání v jakémkoliv časovém okamžiku. Každý bit kódu je snímán se vzorkováním 2 snímky na jeden bit. Kód je určován ze zaznamenaných velikostí shluků. Ukládá se vždy posledních 28 velikostí shluků. Je-li zaznamenáno všech 28 vzorků, je provedena detekce kódu, jinak je kód nastaven na 0. Při detekci kódu je potřeba prvně určit, jestli jsou bitové přechody mezi dvojicemi začínajícími na sudých nebo lichých indexech. To je provedeno porovnáním součtu absolutních hodnot rozdílů velikostí plochy ve dvojicích viz 4.1. Jsou zvoleny dvojice s menšími rozdíly. Poté následuje detekce kódu, kde je kód procházen po dvojicích a je porovnáván rozdíl mezi současným součtem ploch dvojice proti předchozímu součtu. Pokud je procentuální zvětšení plochy větší než daná hranice, jedná se o přechod nahoru, pokud je procentuální zvětšení plochy menší než daná hranice, jedná se o přechod dolů. Algoritmus detekce kódu funguje jako stavový automat. Začíná s předpokladem, že první hodnota je 1. Další hodnoty jsou také určeny jako 1, dokud není nalezen přechod dolů nebo nahoru. Pokud se jedná o přechod dolů, byl odhad první hodnoty správný a následující stav je kód ve stavu 0. Pokud se jedná o přechod nahoru, byl prvotní odhad špatný, hodnoty jsou korigovány a následující stav zůstává kód v hodnotě 1. Pokud je automat ve stavu kód v hodnotě nula následují 3 možnosti. Bude detekován přechod nahoru, poté přejde do stavu 1, nebude detekován přechod, automat zůstane ve stavu 0. Pokud bude detekován přechod dolů, byl současný stav špatně identifikován, opraví se poslední hodnota kódu a pokračuje se ve stavu 0. Ve stavu 1 následují také 3 možnosti. Je-li detekován přechod dolů je následující stav 0, pokud nebude detekována změna, zůstane stav 1 pokud je detekován přechod nahoru, je v detekovaném kódu chyba, je opravena poslední hodnota a pokračuje se ve stavu 1.

```

double _area[LED_CODE_WIDTH]; // array of sizes of led in time
int _it; // index of the oldest value

bool LED::CodeEven() {
    double oddSum = 0;
    int index = _it; // first index of the code
    for (int i = 0; i < LED_CODE_WIDTH;
        i+=2, index = (index + 2) % LED_CODE_WIDTH) {
        oddSum += abs(_area[index] - _area[(index+1) % LED_CODE_WIDTH]);
    }

    index = (_it + 1) % LED_CODE_WIDTH; // start on _it + 1
    double evenSum = 0;
    for (int i = 0; i < LED_CODE_WIDTH;
        i+=2, index = (index + 2) % LED_CODE_WIDTH) {
        evenSum += abs(_area[index] - _area[(index+1) % LED_CODE_WIDTH]);
    }

    if (oddSum < evenSum) {
        return true;
    }
    else {
        return false;
    }
}

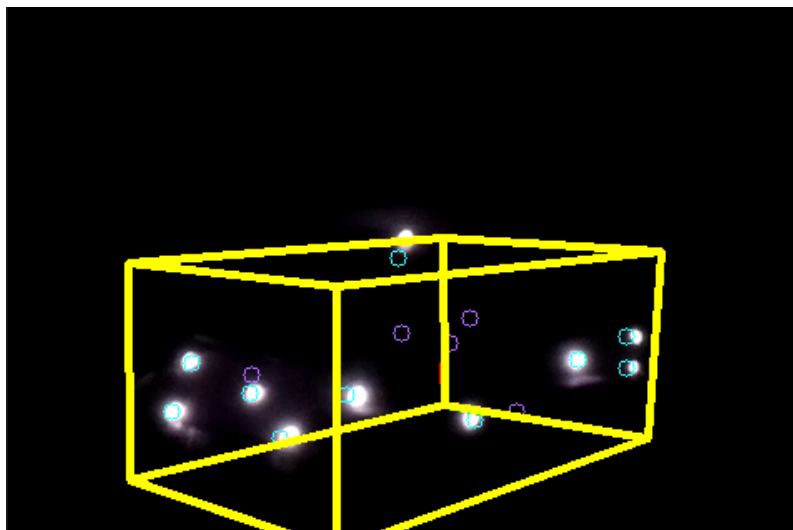
```

Výpis 4.1: Algoritmus zjišťující jestli kód diody začíná na sudém nebo lichém indexu

Následuje kontrola kódu, kód je normalizován na svou nejmenší hodnotu ze všech možných rotací, poté je porovnán s validními kódy. Pokud je hodnota detekovaného kódu validní je ponechána, pokud má hammingovu vzdálenost 1 s některým z ostatních kódů, tak je hodnota opravena. Pokud je hammingova vzdálenost 2, je detekovaný kód nastaven na 0. Poté je detekována jistota detekce. To je provedeno pomocí paměti na nejčastěji detekovaný kód a počítadlo počtu výskytu kódu. Při každé shodě detekovaného kódu s uloženým kódem je počítadlo navýšeno a to až do počtu délky kódu. Pokud se kód neshoduje, je počítadlo sníženo, pokud dosáhne nuly, je uložený kód aktualizován na nový. Pokud je počítadlo vyšší než 5, je kód led diody nastaven na uložený kód, jinak je kód led diody 0.

Algoritmus detekce polohy tanku

Pro detekci polohy tanku se používá aproximace polohy pomocí mapování bodů trojrozměrného modelu na dvourozměrné souřadnice určením translačního a rotačního vektoru (dále translace a rotace). To je provedeno pomocí řešení problému perspektivy n bodů (anglicky Perspective- n -Point problem) 2.2. Je využito iterativní metody Ransac 2.2, která je odolnější proti špatně identifikovaným bodům. Je využit algoritmus EPNP s optimalizací pomocí předchozích hodnot. Implementovaný v knihovně OpenCV. Následně je výsledek zpřesněn kalmanovým filtrem 2.2.



Obrázek 4.7: Detekce vozidla, vozidlo je ohraničeno kvádrem, detekované led diody modře, očekávané pozice led diod žlutě. ^b

^apřevzato z: vlastní

^bpřevzato z: vlastní

První krok detekce je filtrování vhodných bodů, jsou využity pouze body, které mají platný kód a pokud už byla určena předchozí póza, je provedena projekce bodů modelu pomocí předchozích parametrů translace a rotace a jsou porovnány předpokládané a současné souřadnice, body, které jsou příliš vzdálené od předpokládaných pozic jsou vyřazeny. Tím je zamezeno špatně identifikovaným kódům ovlivnit odhad polohy. Poté jsou vybrány všechny trojrozměrné body modelu náležící detekovaným kódům a je provedena estimace pózy algoritmem EPNP s 5 iteracemi, optimalizovaná metodou RANSAC. Získané souřadnice jsou optimalizovány Kalmanovým filtrem s 16 stavy.

Algoritmus detekce zásahu

Pokud je určená poloha vozidla, je možné určit, jestli je vozidlo při výstřelu zasaženo. Je zjištěno, jestli se tank nachází ve směru míření kanónu vozidla. Kanón vždy míří do středu obrazu kamery. Protože mají mířidla pouze jednu osu volnosti, je považována za směr míření plocha kolmá na plochu projekční roviny kamery procházející středem obrazu a středem delších stran obrazu. Kontrola zásahu je určena pomocí určení průsečíků roviny mířidel a hranolu ohraničující vozidlo. Pokud existuje průsečík, je určeno vozidlo při výstřelu jako zasažené.

Průsečíky jsou určeny pomocí projekce roviny a hranolu do plochy promítání kamery. Poté jsou jednotlivé přímky kontrolovány na průsečíky, jestli se protínají, jedná se o zásah. Pro kontrolu průsečíků dvou přímek byl použit algoritmus popsáný v [6].

Prostředky

Pro detekci led diod je využito knihovny OpenCV 4.0.1. Využitá kamera je Raspberry Pi NoIR Camera Module V2 s čipem Sony IMX219 CCD, který je schopný ve spojení s Raspberry PI 3B+ natáčet video se snímkovací frekvencí 90 snímků za vteřinu a rozlišení

640 na 480 pixelů. Pro zpracování videa byl využit vestavěný systém Raspberry PI 3B+, který v reálném čase video zpracovává.

4.4 Komunikační protokoly

Komunikace mezi serverem a mikrokontrolérem

Mezi serverem a mikrokontrolérem probíhá komunikace po sériové lince. Server posílá mikrokontroléru servence 5 bytů, 1. byt určuje typ příkazu, který má mikrokontrolér provést, význam 3. a 4. bytu je specifický podle typu příkazu. 4. a 5. byt jsou kontrolní byty obsahující hodnotu '\n' a '\0'.

Příkazy mikrokontroléru			
Kód příkazu	rozsah hodnot 1. příkazu	rozsah hodnot 2. příkazu	význam příkazu
M	(-100) - 100	(-128) - 127	příkaz pohybu, první hodnota určuje směr pohybu vůči ose vozidla, druhá hodnota určuje rychlost pohybu
T	(-90) - (90)	nedefinován	příkaz pro pohyb mířidel, 1. hodnota určuje úhel natočení mířidel vůči ose vozidla

Tabulka 4.1: Příkazy mikrokontroléru

Komunikace mezi ovladačem a vozidlem

Mezi ovladačem probíhá komunikace po síti pomocí socketů nad protokolem TCP na portu 48707. Kódování je velmi podobné jako při komunikaci serveru a mikrokontroléru. Obsahuje 4 byty. 1. byt určuje typ příkazu, význam 3. a 4. bytu je specifický podle typu příkazu. 4. byt je kontrolní znak '\0'. Jsou přenášeny stejné typy příkazy jako při komunikaci serveru a mikrokontroléru, ale navíc obsahuje následující příkazy.

Příkazy vozidlu			
Kód příkazu	rozsah hodnot 1. příkazu	rozsah hodnot 2. příkazu	význam příkazu
M	(-100) - 100	(-128) - 127	příkaz pohybu, první hodnota určuje směr pohybu vůči ose vozidla, druhá hodnota určuje rychlost pohybu
T	(-90) - (90)	nedefinován	příkaz pro pohyb mířidel, 1. hodnota určuje úhel natočení mířidel vůči ose vozidla
S	nedefinován	nedefinován	příkaz pro výstřel po druhém vozidlu
R	nedefinován	nedefinován	příkaz pro restart životů vozidla

Tabulka 4.2: Příkazy vozidlu

Komunikace mezi vozidly

Mezi vozidly probíhá komunikace, která zajišťuje soubojový systém. Jedná se jednoduchý komunikační protokol nad protokolem TCP na portu 48777, kde při detekování zásahu vozidla je odeslána zpráva, která obsahuje 5 bytů s následujícími hodnotami 'H', 0, 0, '\n' a '\0'.

4.5 Využití aplikace

Aplikace ovládající mikrokontrolér

Cílem aplikace je obsloužit požadavky vestavěného systému na ovládání pohybu a ovládání infračervených led diod.

Byla navržena jako server, který obsluhuje požadavky ovladače pomocí neblokujících funkcí, čímž zajišťuje paralelní obsluhu změny kódů led diod a pohybu vozidla.

Aplikace pracuje v nekonečné smyčce, první část smyčky obsluhuje dekodování a provádí příkazy přijaté po sériové lince. Druhá část nekonečné smyčky obsluhuje aktualizaci led diod.

Na začátku běhu programu se nastaví kódy led diod, čas první změny jasu infračervených led diod, je inicializována sériová linka a I2C rozhraní.

Každý cyklus se načtou ze sériové linky byty tak, aby jich celkově bylo načteno právě 5, pokud není ve vstupním zásobníku dostatek bytů k čtení, uloží se a zbytek bude načten v příštím běhu smyčky. Pokud se povede načíst 5 bytů, jsou byty dekodovány a podle typu příkazu provedeny konkrétní akce. Pokud jde o příkaz ovládající motory, jsou hodnoty rychlosti a úhlu pohybu přepočteny na rychlosti a směry pohybu samostatných motorů, které jsou pak vykonány. Jedná-li se o příkaz servu je daný úhel překódován pro potřeby knihovny Servo a příkaz vykonán. Pro aktualizaci led diod je kontrolováno jestli nastal čas na změnu jasu led diod. Pokud ano, je vypočítán další čas změny času a aktualizovány jas led diod přes I2C rozhraní. To je provedeno pomocí knihovnických funkcí knihovny Adafruit PWM Servo Driver.

Je využito knihoven Arduino Wire, Servo, Serial a knihovny Adafruit PWM Servo Driver. Aplikace byla implementována v jazyce Arduino.

Aplikace server

Cílem aplikace je obsloužit požadavky aplikace ovladač a zajistit komunikaci mezi vozidly pro zajištění soubojového systému.

Aplikace byla navržena jako dva iterativní servery, jeden pro komunikaci s ovladačem, druhý pro komunikaci s druhým vozidlem a detekční program. Všechny běží paralelně. Spolu komunikují pomocí globálních proměnných s výlučným přístupem.

Pro komunikaci mezi vlákny jsou využity globální proměnné HP, Shoot a Hit. Proměnná HP je číslo určující počet životů vozidla. Proměnná Shoot signalizuje vláknu starajícího se o detekci, že má zjistit jestli bylo nepřátelské vozidlo zasaženo a proměnná Hit signalizuje hlavní aplikaci, že nepřátelské vozidlo bylo zasaženo.

Při zapnutí aplikace je inicializováno spojení s mikrokontrolérem. Dále aplikace běží v nekonečné smyčce, kde v každém běhu čeká na připojení ovladače, poté spustí další vlákna obstarávající komunikaci s druhým vozidlem a detekci vozidla. Pokusí přečíst tolik bytů, aby mohla provést další příkaz. Pokud bylo načteno dostatek dat, dekóduje příkaz a je proveden. Pokud nebylo načteno dostatek dat, jsou data uloženy a zbytek příkazu je načten v dalším běhu smyčky. Obdržené příkazy se dělí do dvou kategorií. Příkazy pro ovládání

vozidla a příkazy pro souboje. Příkazy pro ovládání vozidla jsou dále přeposlány po sériové lince mikrokontroléru. Pokud se jedná o příkazy pro souboj vozidel, jsou nastaveny globální proměnné pro signalizaci kontroly zásahu nebo obnovení hry do počátečního stavu. Nakonec je zkontrolován stav globální proměnné HP, pokud proběhla změna je to odesláno ovladači aby mohl informovat uživatele.

Vlákno pro detekci vozidla běží v nekonečné smyčce, kde detekuje nepřátelské vozidlo pomocí algoritmu popsaného v 4.3, pokud je nastavena globální proměnná Shoot, je zkontrolováno jestli je vozidlo zasaženo. Pokud ano, je nastavena proměnná Hit, poté je proměnná Shoot nulována.

Vlákno pro komunikaci s druhým vozidlem čeká na zprávu o obdržení zásahu a poté sníží proměnnou HP. Pokud je nastavená proměnná Hit, posílá druhému vozidlu informaci o tom že bylo zasaženo a proměnnou Hit nuluje.

Aplikace byla implementována v jazyce C++. Pro síťovou komunikaci byly využity schránky, sériová komunikace byla implementována pomocí knihovny termios. Detekční algoritmus využívá knihovnu OpenCV.

Aplikace kontroler

Účelem aplikace je vytvořit ovladač, který bude pomocí síťového protokolu ovládat všechny funkce vozidla a prezentovat užitečné údaje o souboji s nepřátelským vozidlem. Síťová aplikace byla navržena jako klient připojující se na server, který je umístěný ve vozidle.

Ovládací klávesy	
Klávesa	Význam
w	pohyb dopředu
s	pohyb dozadu
a	otočení doleva na místě
d	otočení doprava na místě
q	pohyb šikmě doleva dopředu
e	pohyb šikmě doprava dopředu
z	pohyb šikmě doleva dozadu
c	pohyb šikmě doprava dozadu
k	otočení věže doleva
a	otočení věže doprava
p	výstřel
t	obnovení životů

Tabulka 4.3: Tabulka ovládacích kláves

Aplikace se připojí na specifikovaný server, poté nastaví vstupní terminál na nebufferovaný a čtení na neblokující. Poté probíhá nekonečná smyčka, ve které probíhá čtení ze vstupu. Pokud načtený znak odpovídá ovládací klávese je vyslán příkaz na server. Pokud se jedná o příkaz pohybu je nastaven časovač a pokud nebude do stanoveného intervalu načtena stejná klávesa nebo se neprovede změna pohybového příkazu, bude vozidlo zastaveno. Při každém průběhu smyčky probíhá kontrola, jestli server neposlal oznámení o zásahu vozidla a jeho zbývajících životech nebo jestli nebylo zasaženo nepřátelské vozidlo. Pokud ano je informace vypsána uživateli.

4.6 Ověření funkčnosti a interpretace výsledků

Ověření funkčnosti

Pro ověření funkčnosti byl vytvořený software nahrán do modelu a otestována jeho základní funkcionality. Pro testování pohyblivosti modelu byly vybrány různé povrchy a překážky.

Testování ovládání modelu bylo provedeno na skupině 10 lidí ve věkovém rozmezí 10 až 50 let. Ukázalo se, že ovládání je intuitivní pro část skupiny, která má zkušenosti s hraním počítačových her, pro ostatní ale také nebyl problém po krátké chvíli si na ovládání zvyknout. Dalším zjištěním bylo, že uživatelům vadí krátké přerušování na začátku pohybu vozidla při držení klávesy.

Testování souborového systému bylo provedeno pomocí dalšího Raspberry Pi, na které byl nahrán software, který imituje další vozidlo. Detekce byla vyhodnocena jako výborná na vzdálenost do 2 metrů a dostačující na vzdálenost 4m.

	odezva
ovládání vozidla	0,098 s
zaregistrování zásahu	0,198 s
obdržení zásahu	0,211 s

Tabulka 4.4: Měřené odezvy

Pro testování síťové komunikace byly měřeny odezvy pro ovládání vozidla, doba zaregistrování zásahu nepřátelským vozidlem a doba obdržení zásahu nepřátelským vozidlem. Na lokální síti byly naměřeny hodnoty viz tabulka 4.4.

Interpretace výsledků

Ovládání vozidla bylo po testování vyhodnoceno jako dostačující. Detekce vozidel byla vyhodnocena jako výborná pro vzdálenosti nižší než 2 metry a dostačující pro vzdálenost 5 metrů, což odpovídá zamýšlenému vnitřnímu využití. Odezva ovládání při síťové komunikaci byla vyhodnocena jako výborná, stejně jako informování o zásahu vozidla a doba oznámení zásahu druhému vozidlu.

Kapitola 5

Závěr práce

Cílem práce bylo navrhnout a sestavit funkční model vozidla, které bude dálkově ovládáno pomocí počítače. Tento cíl byl splněn.

Po prostudování dostupných materiálů týkajících se ovládání modelu pomocí modelářských serv a motorů pro modely byly nejdůležitější poznatky popsány v teoretické části, ze které jsem vycházel při návrhu modelu. Byly vybrány vhodné pohony a k ovládání byl vybrán jednodeskový počítač a mikrokontrolér. Jako rozšíření je vozidlo vybaveno kamerou bez infračerveného filtru a je na něm rozmístěno 15 infračervených led diod pro jeho detekci, čímž jsem demonstroval další využití počítače v modelu.

Na základě mého technického návrhu jsem sestrojil pásový model tanku Tiger II z dřevovláknitých desek MDF v měřítku 1:25, který dosahuje dobré pohyblivosti díky využití dvou nezávisle ovládaných pásů. Model umožňuje provádět souboje s jinými vozidly pomocí detekce vozidla na základě identifikace led diod umístěných na vozidle a estimace pózy pomocí problému perspektivy n bodů (anglicky Perspective-n-Point problem). Pro jeho ovládání byly vytvořeny síťové aplikace umožňující jeho bezdrátové ovládání po síti. Při konstrukci jsem využil poznatky zjištěné v teoretické části práce. Zvolené řešení bylo dostatečné pro zamýšlenou funkci. Vozidlo bylo prakticky testováno a soubojový systém byl simulován pomocí dalšího jednodeskového počítače. Dosažené výsledky byly vyhovující danému účelu. Kvůli časovému omezení práce nebylo dokončeno druhé identické vozidlo pro umožnění soubojů a pokročilejší soubojový systém.

Dalším pokračováním práce by mohlo být dokončení stavby druhého vozidla, vytvoření ovládací aplikace s grafickým uživatelským rozhraním, mechanické zdokonalení modelu tanku, jako kontrola elevace kanónu, zdokonalení bojového systému nebo aplikace pro mobilní telefon.

Literatura

- [1] IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, Dec 2016, doi:10.1109/IEEESTD.2016.7786995.
- [2] AG, A.: *Arduino Uno Rev3*. Arduino AG, [Online; navštíveno 21.04.2019].
URL <https://store.arduino.cc/arduino-uno-rev3>
- [3] AG, A.: *Boards & Modules - Arduino*. Arduino AG, [Online; navštíveno 18.04.2019].
URL <https://store.arduino.cc/arduino-genuino/boards-modules>
- [4] Bluetooth Special Interest Group: *Specification of the Bluetooth System*. Leden 2019, v5.1 [Online; navštíveno 08.05.2019].
URL
https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457080
- [5] Cellan-Jones, R.: *BBC - dot.Rory: A 15 pound computer to inspire young programmers*. BBC - dot.Rory, Květen 2011, [Online; navštíveno 03.05.2019].
URL https://www.bbc.co.uk/blogs/thereporters/rorycellanjones/2011/05/a_15_computer_to_inspire_young.html
- [6] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; aj.: *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009, ISBN 978-0-262-03384-8.
URL <http://mitpress.mit.edu/books/introduction-algorithms>
- [7] Foundation, R. P.: *Buy a Pi – Raspberry Pi*. Raspberry Pi Foundation, [Online; navštíveno 27.04.2019].
URL <https://www.raspberrypi.org/products/>
- [8] Foundation, R. P.: *Raspberry Pi Documentation*. Raspberry Pi Foundation, [Online; navštíveno 16.04.2019].
URL <https://www.raspberrypi.org/documentation/>
- [9] Gevers, T.; Gijzenij, A.; van de Weijer, J.; aj.: *Color in Computer Vision: Fundamentals and Applications*. Wiley Publishing, první vydání, 2012, ISBN 0470890843, 9780470890844.
- [10] Giri, F.: *AC electric motors control: advanced design techniques and applications*. Wiley, 2013, ISBN 978-1-118-33152-1.

- [11] Horowitz, P.; Hill, W.: *The Art of Electronics*. New York, NY, USA: Cambridge University Press, třetí vydání, 2015, ISBN 0521809266, 9780521809269.
- [12] Lu, X. X.: A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation. *Journal of Physics: Conference Series*, ročník 1087, sep 2018: str. 052009, doi:10.1088/1742-6596/1087/5/052009.
URL <https://doi.org/10.1088/1742-6596/1087/5/052009>
- [13] Petruzella, F. D.: *Electric motors and control systems*. New York: McGraw Hill education, second edition vydání, 2016, ISBN 9780073521824.
- [14] Platt, C.: *Encyclopedia of Electronic Components Volume 1: Resistors, Capacitors, Inductors, Switches, Encoders, Relays, Transistors*. O'Reilly Media, 2012, ISBN 9781449333850.
- [15] Roubíček, O.: *Elektrické motory a pohony: příručka techniky, volby a užití vybraných druhů*. BEN - technická literatura, 2004, ISBN 80-7300-092-X.
- [16] Sorenson, H. W.: *Kalman filtering : theory and applications*. IEEE Press selected reprint series, New York: IEEE Press, 1985, ISBN 0-7803-0421-7.
- [17] Stockman, G.; Shapiro, L. G.: *Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, první vydání, 2001, ISBN 0130307963.
- [18] Szeliski, R.: *Computer Vision : Algorithms and Applications*. Texts in computer science, London: Springer, 2010, ISBN 978-1-84882-934-3.
- [19] wiki, S.: *Maple Mini - STM32duino wiki*. STM32duino wiki, [Online; navštíveno 23.04.2019].
URL https://wiki.stm32duino.com/index.php?title=Maple_Mini