

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Autonomní sledování člověka mobilním robotem

Martin Novotný

Vedoucí: Ing. Karel Košnar, Ph.D.
Obor: Kybernetika a robotika
Studijní program: Kybernetika a robotika
Květen 2019

Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Karlu Košnarovi, Ph.D. za podnětné připomínky, užitečné a věcné rady při práci na tomto projektu.

Prohlášení

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

V Praze, dne

.....
podpis autora práce

Abstrakt

Obsahem této práce je seznámit se s metodami lokalizace objektů pomocí kamery, k čemuž byl využit lokalizační systém detekující známou značku z obrazu. Dále je úkolem seznámit se s metodami pro rekonstrukci trajektorie. K tomu je nutné filtrovat naměřená data či využít proložení naměřených bodů polynomem. Následně je cílem vytvořit řídicí smyčku pro ovládní robotu tak, aby byl schopen projet vytvořenou cestu. Celý algoritmus je implementován na reálném robotu a jeho funkčnost experimentálně ověřena.

Klíčová slova: sledování člověka, navigace pomocí kamery, rekonstrukce trajektorie

Vedoucí: Ing. Karel Košnar, Ph.D.

Abstract

The subject of this work is to acquaint with a localization methods using a monocular camera. To do so a localization system detecting known marker is used. Additional task is to acquaint with a trajectory reconstruction methods such as filtering of measured data or useage of splines to recover missing sections of the trajectory. The goal is to create a control loop for the robot, so it can follow the human along constructed path. Whole algorithm is implemented on the real robot and its functionality is experimentaly verified.

Keywords: human following, camera-vision-based navigation, trajectory reconstruction

Title translation: Autonomous Human-following robot

Obsah

1 Úvod	1	3.1.2 Vizualní lokalizace	14
1.1 Definice úlohy	2	3.1.3 Sjednocení soustav	15
1.2 State of the art	3	3.2 Filtrace naměřených dat	15
1.2.1 Sledování člověka pomocí infračervené kamery	3	3.2.1 FIR filtr	15
1.2.2 Sledování pomocí detekce člověka v obraze	4	3.2.2 Kalmanův filtr	16
1.2.3 Sledování člověka pomocí laserového senzoru	5	3.3 Křivky	17
1.2.4 Přirozené chování robotů	5	3.3.1 Polynom prvního řádu	17
2 Vizualní sledování objektů	7	3.3.2 Polynom třetího řádu	18
2.1 Kalibrace kamery	7	3.3.3 Hermitovské křivky	18
2.2 Detekce značek v obraze	9	3.4 Follow the carrot	20
2.2.1 Příklad principu detekce značky Apriltag	10	3.5 PID regulátor	21
2.3 Detekce osoby v obraze	10	4 Implementace	23
3 Sledování trajektorie	13	4.1 Turtlebot 2 se systémem ROS	23
3.1 Lokalizace	13	4.2 Kamera	24
3.1.1 Lokalizace pomocí odometrie	13	4.3 Testování vlastností značek	25
		4.4 Rekonstrukce cesty	27
		4.5 Metoda sledování cesty	27
		4.6 Řídící smyčka algoritmu	28

5	Vyhodnocení reálných pokusů	31
5.1	Ostrá zatáčka	32
5.1.1	Kamera s obyčejným objektivem	32
5.1.2	Kamera s širokoúhlým objektivem	34
5.2	Sledování delší trajektorie	36
5.2.1	Kamera s obyčejným objektivem	38
5.2.2	Kamera s širokoúhlým objektivem	39
5.3	Test vlivu četnosti detekce na sledování	41
6	Závěr	43
6.1	Možnosti rozšíření práce	45
	Literatura	47
	Zadání práce	51

Obrázky

1.1 Robot sledující člověka pomocí kamery.	3	2.7 Zleva původní snímek, detekované osoby, model s postavami na detekovaných pozicích. Převzato z [23].	12
1.2 Snímání polohy objektu s pomocí infračervených diod a kamery. Převzato z [11].	4	3.1 Princip odhadu změny polohy senzoru z po sobě následujících snímků. Převzato z [20].	14
1.3 Detekce ramen a vzoru na oblečení. Převzato z [18].	4	3.2 Schéma FIR filtru. Převzato z [37].	16
1.4 Laserová data z detekce nohou. Převzato z [10].	5	3.3 Proložení dvou bodů p_0 a p_1 kubickou Hermitovskou křivkou s tečnami v_0 a v_1 . Převzato z [14].	19
1.5 Robot se sociálními vlastnostmi. Převzato z [16].	6	3.4 Kubické Hermitovy polynomy H_{0-3}^3 . Převzato z [14].	20
2.1 Model dírkové kamery.	8	3.5 Follow the carrot algoritmus. Převzato z [25].	20
2.2 Příklad zkřivení obrazu kamery. Převzato z [12].	8	3.6 Schéma systému řízeného PID regulátorem. Převzato z [27].	21
2.3 Různé typy značek, které lze využít k určení pozice.	9	4.1 Rozměry robotu Turtlebot. Převzato z [30].	23
2.4 Kroky při detekci Apriltag značky. Převzato z [26].	10	4.2 Použitá kamera s dvěma typy objektivů.	24
2.5 Detekce postavy v obraze (a). Odstranění pozadí (b - c). Přiřazení pravděpodobností pro části těla. Čím světlejší, tím větší pravděpodobnost, že pixel patří k části těla. Převzato z [13].	11	4.3 Trajektorie značek při testování jejich vlastností.	26
2.6 Tvorba 3D mřížky pro určení pozice. Převzato z [23].	11	4.4 Průběh rychlosti v závislosti na vzdálenosti od sledované osoby. ...	28

4.5 Schéma řídicí smyčky algoritmu se všemi částmi.	29
5.1 Turtlebot 2 s kamerou Basler. ...	31
5.2 Porovnání skutečných pozic a zkreslených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě [0, 0]. Pokus: ostrá zatáčka, obyčejný objektiv.	33
5.3 Porovnání skutečných pozic a zkreslených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě [0, 0]. Pokus: ostrá zatáčka, širokoúhlý objektiv.	35
5.4 Porovnání skutečných pozic a zkreslených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě [0, 0]. Pokus: delší trajektorie, obyčejný objektiv.	38
5.5 Porovnání skutečných pozic a zkreslených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě [0, 0]. Pokus: delší trajektorie, širokoúhlý objektiv.	40
5.6 Porovnání skutečných pozic a zkreslených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě [0, 0]. Pokus: vliv četnosti detekcí.	42

Tabulky

4.1 Základní parametry použité kamery.	24
4.2 Parametry získané při kalibraci kamery.	25
4.3 Vypočítané úhly záběru kamery s použitými objektivy.	25
4.4 Parametry klíčových vlastností značek.	26
5.1 Vypočítané odchylky k jednotlivým grafům na obrázku 5.2.	34
5.2 Vypočítané odchylky k jednotlivým grafům na obrázku 5.3.	36
5.3 Vypočítané odchylky k jednotlivým grafům na obrázku 5.4.	38
5.4 Vypočítané odchylky k jednotlivým grafům na obrázku 5.5.	39
5.5 Vypočítané odchylky k jednotlivým grafům na obrázku 5.6.	42

Kapitola 1

Úvod

V současné době, kdy výrobní procesy v továrnách zažívají velké změny ve formě digitalizace a automatizace, někdy nazývané Průmysl 4.0, je nutné do těchto procesů implementovat složitější a sofistikovanější technologie. Ty je nutné spojit do komplexních systémů, které budou schopné zajistit maximální koordinaci s minimální časovou náročností. Jednou z možností, jak se přiblížit tomuto cíli, je zajistit lepší spolupráci mezi lidmi a roboty. Příkladem může být převoz materiálů, které jsou příliš těžké na přemístění pro člověka. V takovém případě lze využít systému, kde robot umí rozpoznat člověka, je schopen ho následovat po požadované trajektorii a plnit jeho povely tak, že dojde k přesunu nákladu na požadované místo. Aby byl takový systém co nejdostupnější, je nutné použít co nejdostupnější komponenty. Proto lze pro určení pozice využít obyčejné kamery a z jejího obrazu získat informaci o poloze či povelch.

Tato práce se konkrétně zabývá vytvořením algoritmu pro řízení mobilního pozemního robotu, jehož úkolem je sledovat člověka po stejné trajektorii. Cílem je vytvořit řešení, které pro určení polohy člověka využije běžně dostupné kamery. Jelikož v reálných podmínkách může snadno dojít ke ztrátě osoby ze záběru, je nutné, aby vytvořený algoritmus uměl zajistit sledování trajektorie i s neznámými úseky. Využití pro takový systém lze nalézt nejenom v průmyslové oblasti při přesunu materiálů, ale i v životě běžných lidí, kteří potřebují asistenci, například při přenosu těžkých břemen. Další uplatnění lze nalézt i v armádní oblasti, kdy je při přesunu nutné přemístit větší množství vybavení, které může být velmi těžké.

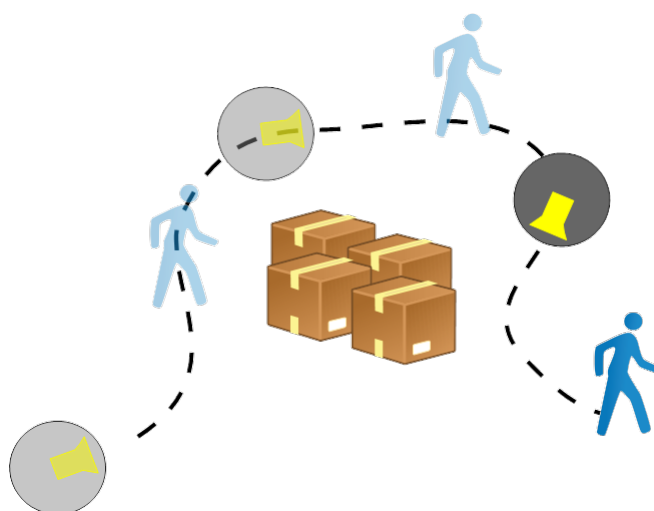
V rámci úvodní kapitoly této práce je uvedeno několik metod použitých pro sledování člověka mobilním robotem. Každá z těchto metod využívá jiného principu lokalizace člověka. V první kapitole je uveden princip detekce speciálních značek a princip detekce člověka z obrazu kamery. Ve druhé kapitole jsou uvedeny metody pro rekonstrukci trajektorie a metody, které zajistí sledování této zrekonstruované cesty. V další kapitole je rozebrána samotná implementace na reálném robotu a v následující kapitole jsou představeny výsledky experimentů s reálným robotem.

1.1 Definice úlohy

Zadáním této diplomové práce je vytvořit řídicí algoritmus pro mobilního pozemního robota, jehož úkolem je následovat člověka po stejné trajektorii. Problém lze rozdělit do několika podúloh. Těmi jsou například určení polohy člověka vzhledem k robotu, rekonstrukce trajektorie z naměřených pozic a řídicí smyčka zajišťující průjezd robota po vytvořené trajektorii.

Pro určení polohy člověka lze využít mnohá zařízení, například rozmítaný laserový dálkoměr, kameru, termokameru nebo GPS. Je žádoucí, aby použitá zařízení byla co možná nejdostupnější. Jelikož laserové dálkoměry a termokamery mohou být poměrně nákladné, zdá se vhodnější použití kamery nebo GPS. Pokud by plánované prostředí pohybu robota bylo pouze na otevřených prostranstvích, lze dobře využít GPS. Ovšem v případě, kdy chceme zajistit co největší funkčnost, v co nejvíce rozmanitém prostředí, včetně budov a skladů, jež způsobují špatné fungování GPS, je vhodnější použít k lokalizaci kameru. K určení polohy pomocí kamery lze využít speciálních značek známé velikosti a tvaru, které se následně detekují v obraze. Podle velikosti je možné určit vzdálenost této značky a díky tomu i polohu. Pro detekci člověka lze také využít učící algoritmy s využitím neuronových sítí. Ovšem u tohoto přístupu je složitější určení vzdálenosti.

Při použití kteréhokoli výše zmíněného zařízení obsahují naměřené polohy odchylku od skutečné polohy. To může být způsobeno šumem senzorů či nepřesnostmi v konstrukci robota. Pro co největší potlačení těchto nežádoucích vlivů a pro získání co nejpřesnější trajektorie lze využít filtry, jako jsou například Kalmanův filtr, nebo FIR filtr. V reálných podmínkách může nastat a také často nastává situace, kdy se na nějaký kratší čas osoba ztratí ze záběru kamery. Pozice na neznámém úseku lze dopočítat například proložením polynomem určitého řádu, obvykle 3. řádu. Je také možné využít prediktor, jako je například Kalmanův filtr, který po nějakou dobu od zmizení ze záběru dokáže předpovídat polohu.



Obrázek 1.1: Robot sledující člověka pomocí kamery.

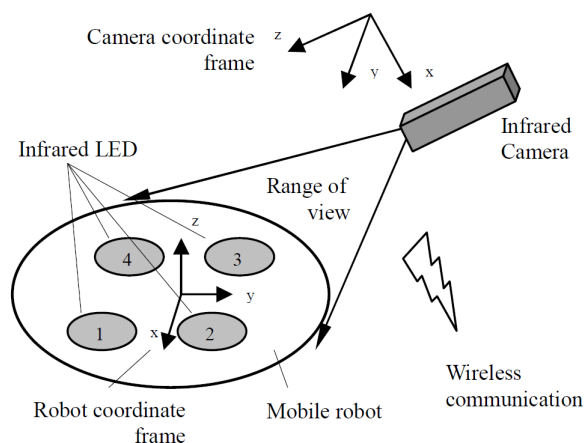
Po nalezení správné trajektorie je nutné vytvořit řídicí smyčku, která robotu přiřadí správnou rychlost tak, aby robot jel co možná nejpřesněji. K tomu lze využít kombinace různých algoritmů a metod. Například „Follow the carrot“, PID regulátor, či virtuální pružinu mezi robotem a člověkem.

■ 1.2 State of the art

■ 1.2.1 Sledování člověka pomocí infračervené kamery

Jeden z možných přístupů pro určování polohy objektu je za pomoci infračervené kamery. Na sledovaný objekt je připevněna sada infračervených diod (IR-LED) do určitého vzoru. V případě, že kamera vidí celou sadu diod, je možné určit relativní polohu vzhledem ke kameře. Z naměřených relativních poloh lze zrekonstruovat cestu vytvořenou člověkem, kterou robot může sledovat. Tento přístup byl použit v článku [11].

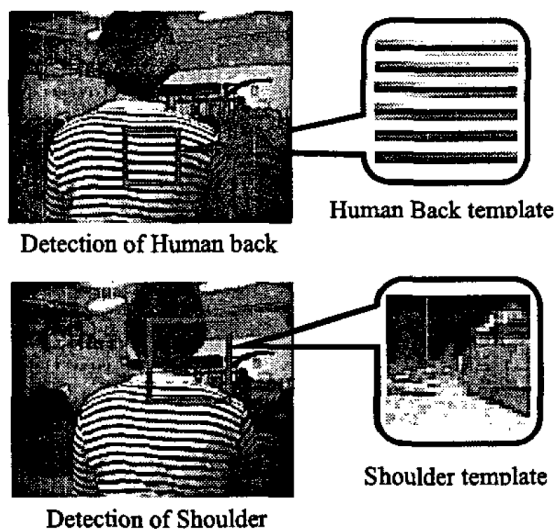
Na robot byly připevněny čtyři IR-LED. Infračervená kamera je připevněna na záda osoby, kterou má robot sledovat. Jakmile kamera vidí všechny čtyři diody, dojde k určení polohy a orientace kamery v souřadnicové soustavě robotu. Následně je vytvořeno virtuální spojení mezi člověkem a robotem, které řídí pohyb robotu.



Obrázek 1.2: Snímání polohy objektu s pomocí infračervených diod a kamery. Převzato z [11].

1.2.2 Sledování pomocí detekce člověka v obraze

Pro sledování člověka můžeme využít kameru, v jejímž obraze lze detekovat osobu. Ovšem určit přesně polohu člověka ke kameře je složité. Jelikož se poloha člověka a kamery neustále mění, mohou se i výrazně měnit světelné podmínky. To může zhoršit kvalitu detekce a správné určení velikosti osoby a následně i vzdálenosti.



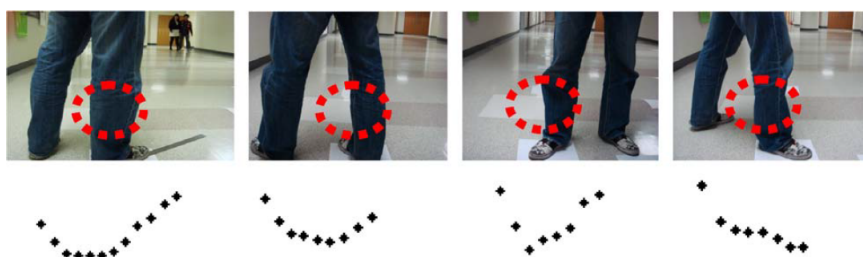
Obrázek 1.3: Detekce ramen a vzoru na oblečení. Převzato z [18].

Důležitou částí je schopnost nejprve detekovat člověka v obraze. Již existují poměrně spolehlivé metody například pro detekce obličeje. Ovšem tato metoda není vhodná pro problém následování člověka, protože člověk by musel být neustále natočený obličejem k robotu a tudíž chodit pozpátku. Vhodnější by bylo, kdyby robot dokázal detekovat člověka zezadu. Detekce by mohla

využívat například vzor na oblečení a tvar ramen. Nejprve by bylo nutné, aby se detektor naučil rozpoznávat vzor a tvar ramen, následně by dokázal tento vzor vyhledat v obraze a podle velikosti vzoru určit polohu sledované osoby. Tato metoda byla použita v článku [18]. Detekcí člověka v obraze se dále zabývá kapitola 2.3.

1.2.3 Sledování člověka pomocí laserového senzoru

Pro sledování pohybu člověka je také v neposlední řadě možné využít rozmítaný laserový dálkoměr umístěný ve výšce lýtky. Tvar lidské nohy připomíná přibližně elipsu, ovšem jak je vidět z obrázku 1.4, není vhodné uvažovat o tvaru jako o konkrétním geometrickém tvaru a ten hledat v laserových datech. Osoba může mít na sobě například volné kalhoty, které by způsobily velkou odchylku od hledaného geometrického tvaru.



Obrázek 1.4: Laserová data z detekce nohou. Převzato z [10].

Z nasbíraného souboru dat lze hned odstranit body ležící v dlouhých přímkách, které odpovídají například zdem. Zbývající body je následně potřeba rozdělit do shluků dle nějaké zvolené hraniční vzdálenosti, čímž získáme první potenciální detekce nohou. Body, které leží mimo shluky, můžeme odstranit. Dále je potřeba z nalezených shluků najít ty, které patří jedné osobě, tedy pravou a levou nohu. K tomu lze využít informace o vzdálenosti obou nohou během kroku z analýzy chůze. Poloha člověka je předpokládána uprostřed, mezi příslušnými polohami nohou. Pro zlepšení detekce lze přidat odhad budoucí polohy člověka na základě známé přibližné délky kroku. Tato metoda byla rozpracována a experimentálně otestována v článku [10], dále modifikovaná verze byla použita v článku [32].

1.2.4 Přirozené chování robotů

Při návrhu algoritmu pro sledování člověka robotem by měl být brán ohled nejen na spolehlivé metody detekce a řízení, ale také na přirozenost po-

hybu a společensky akceptovatelné chování, aby člověk, se kterým má robot spolupracovat, nebyl jeho chováním frustrován či omezován.



(a) : Robot sledující člověka v určité vzdálenosti.



(b) : Obraz s modelem obličeje a informativním textem.

Obrázek 1.5: Robot se sociálními vlastnostmi. Převzato z [16].

Nejdůležitějším aspektem je bezpečnost osob v okolí robotu. Ten se musí vždy chovat takovým způsobem, aby za žádných okolností neohrozil člověka. Neoddělitelně k tomuto patří spolehlivost, díky které robot zůstává užitečný a neobtěžuje osoby kolem sebe svými chybovými stavy. Zároveň by jeho systém měl být navržen tak, aby při selhání nedošlo k ohrožení okolí. Dále by měl robot dodržovat rozumnou vzdálenost od osoby, kterou následuje, aby nenarušoval její osobní prostor. V neposlední řadě by se měl chovat předvídatelným způsobem, ideálně stejně, jako by na jeho místě byl opravdový člověk. Pro lepší interakci s okolím může být robot vybaven systémem pro převod textu na zvuk nebo alespoň obrazovkou s textem, aby měl uživatel informace o stavu robotu. Sociálními aspekty se více zabývá článek [16].

V článku [16] použity dva typy metod pro sledování člověka. U první je sledování řízeno natočením robotu vždy směrem k pozici člověka. Společně s jednoduchou metodou pro vyhýbání se překážkám je robot schopen obstojně sledovat člověka například skrz dveře či bez problému míjet rohy zdí při ostrých zatačkách. Druhou metodou je co nejpřesnější sledování vytvořené cesty, které je založené na metodě „Pure pursuit“ (čistě pronásledování). Obě metody byly následně otestovány několika lidmi, kteří měli zhodnotit chování robotu, například zda jeho chování splňuje očekávání, či jak byl pohyb přirozený. Dle vyhodnocení hlasování účastníků očekávání více splnil první algoritmus řízený natočením, zároveň byl vyhodnocen i jako více přirozený. Vzdálenost při jízdě i při zastavení byla hodnocena podobně pro obě metody.

Kapitola 2

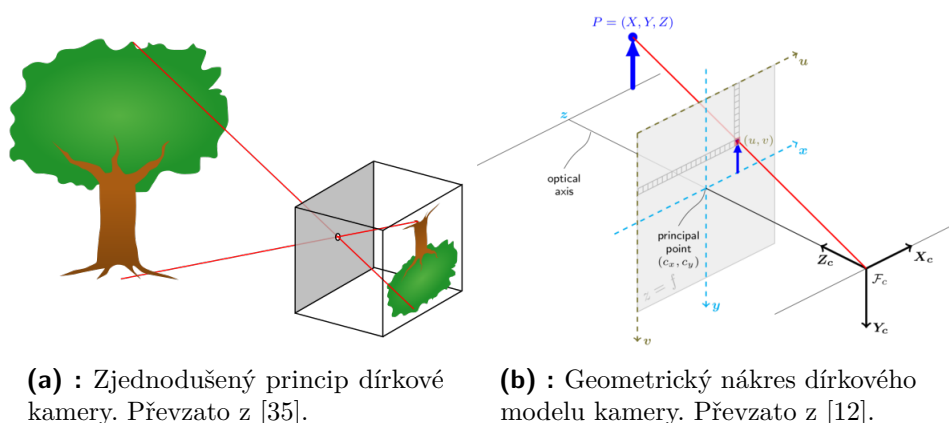
Vizuální sledování objektů

K určení polohy objektu v prostoru lze použít běžně dostupnou kameru. V zachyceném obrazu je možné detekovat požadovaný objekt buď nepřímo, pomocí speciální značky umístěné na sledovaném objektu, nebo přímo, pomocí strojového učení s využitím například neuronových sítí.

2.1 Kalibrace kamery

Před detekcí objektů z obrazu kamery je nutné zjistit hodnoty charakteristických parametrů. Těmi jsou například ohnisková vzdálenost či přesný optický střed. Na kameru lze připevnit různé objektivy, které mají různé vlastnosti a různé chyby a mohou způsobovat různé deformace obrazu. Nesprávné parametry kamery či deformovaný obraz mohou následně zapříčinit nepřesný nebo chybný výstup detekce, nebo kompletní selhání detekce. Z tohoto důvodu je vhodné před použitím kamery provést kalibraci.

Kalibrací lze získat nejenom vnitřní parametry kamery, ale také koeficienty pro potlačení deformace obrazu. Při běžné kalibraci se předpokládá dírkový model kamery. Tím je určen vztah mezi bodem ve 3D světě a jeho obrazem na 2D ploše. Předpokladem modelu je, že otvor pro vnik světla je pouze jediný bod a nejsou použity žádné čočky k zaostření světla.



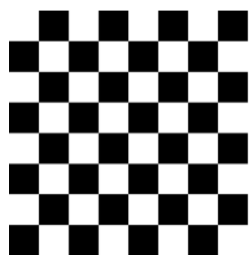
Obrázek 2.1: Model dírkové kamery.

Základní zobrazovací rovnici [12] lze zapsat ve tvaru

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (2.1)$$

$$su = K(R|t)x, \quad (2.2)$$

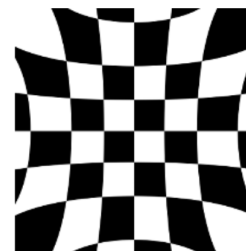
kde x je poloha zobrazovaného 3D bodu, u je 2D poloha zobrazovaného bodu, K je matice kamery obsahující vnitřní parametry, s je měřítko a $R|t$ je matice obsahující vnější parametry, které určují polohu a natočení.



(a) : Obraz bez radiálního zkřivení.



(b) : Pozitivní radiální zkřivení.



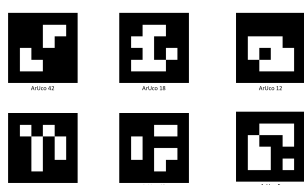
(c) : Negativní radiální zkřivení.

Obrázek 2.2: Příklad zkřivení obrazu kamery. Převzato z [12].

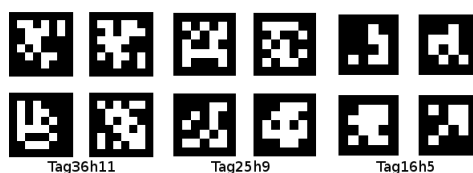
Kalibrací je možné odhalit parametry zkřivení obrazu [12]. Zkřivení mohou být dvojího typu. Prvním je radiální, které může být popsáno až 6 parametry k_{1-6} . Druhým je tangenciální, které je určeno dvěma parametry $l_{1,2}$. Radiální zkřivení lze rozlišit pozitivní, které má typicky $k_1 > 0$, a negativní, které má typicky $k_1 < 0$. Po nalezení parametrů zkřivení provedením kalibrace je možné toto zkřivení potlačit.

2.2 Detekce značek v obraze

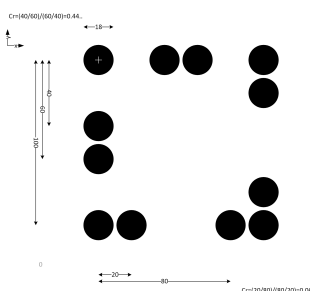
Pro nepřímé měření polohy nějakého objektu lze na objekt umístit speciální značku známé velikosti a tvaru, kterou dokážeme detekovat v obraze kamery. Díky známé velikosti značky jsme schopni určit její vzdálenost od kamery a tudíž i polohu.



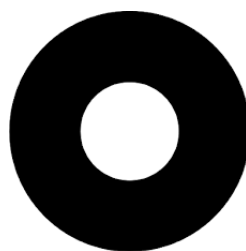
(a) : Aruco značky. Pře-
vzato z [19].



(b) : Apriltag značky. Pře-
vzato z [3].



(c) : Pi-tag značka. Pře-
vzato z [28].



(d) : Whycon značka. Pře-
vzato z [24].

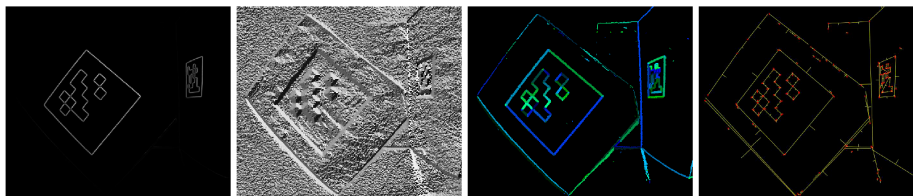
Obrázek 2.3: Různé typy značek, které lze využít k určení pozice.

Značek existuje mnoho druhů. Každá z nich je vhodná pro jiný typ problému. Některé jsou vhodné pro určování malých vzdáleností (v řádu desítek centimetrů), ale s vyšší přesností, jiné jsou vhodné pro detekci na větší vzdálenosti (v řádu metrů) s nižší přesností. Některé značky jsou vzájemně nerozlišitelné, některé jsou unikátní a lze do nich zakódovat nějakou informaci, například číslo. Různé značky jsou detekovány s různou rychlostí. Některé mohou být detekovány v řádu desítek Hz a jiné pouze v jednotkách Hz.

Příkladem nerozlišitelné značky může být například kruhová značka Whycon [22], kterou lze detekovat na vzdálenost v řádu metrů až desítek metrů. Příkladem rozlišitelných značek může být značka Pi-tag [9], pro kterou lze rozlišit různé konfigurace podle rozmístění černých teček na jednotlivých stranách. Jiným příkladem jsou značky Aruco [15], [31], nebo Apriltag [26], [33], které se liší například použitým kódováním bitového pole uvnitř značky.

2.2.1 Příklad principu detekce značky Apriltag

Systém pro detekci značek Apriltag [26] je složen ze dvou hlavních částí. První částí je samotný detektor, který v obraze hledá potenciální kandidáty značek, a kódovací systém, který z načtených bitů čte informaci a kontroluje správnost značky.

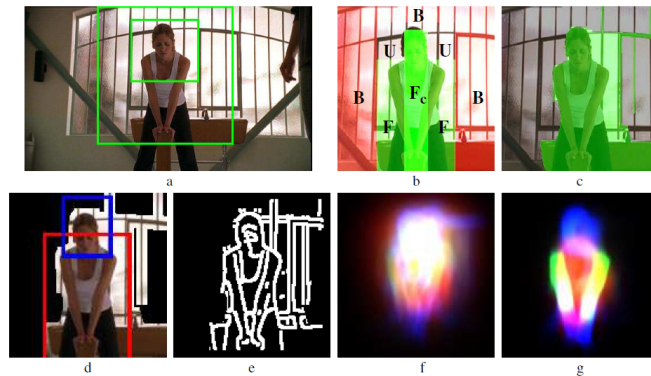


Obrázek 2.4: Kroky při detekci Apriltag značky. Převzato z [26].

Detektor nejprve zjistí velikost a směr gradientu v každém pixelu snímku. Pixely, které mají podobnou velikost a směr gradientu jsou následně přiřazeny do shluků. Na každý shluk je pomocí metody nejmenších čtverců sesazena přímka. Každé přímce je přiřazen směr tak, že tmavé segmenty jsou nalevo a světlé napravo. Následně je pomocí metody prohledávání do hloubky v přímkách hledán čtyřstranný útvar tak, aby směr přímek byl proti směru hodinových ručiček. Nakonec je třeba přechít bity uložené v nalezené značce a pomocí kódovacího systému ověřit, zda je nalezená značka správná či nikoli.

2.3 Detekce osoby v obraze

Detekovat osoby na snímcích s neznámou scénou, osvětlením, s neznámým zvětšením, či s libovolným vhladem osob je velmi obtížný proces. Detekci je nutné rozdělit do několika kroků. Nejprve je potřeba najít přibližnou polohu osoby na snímku. Toho lze dosáhnout například využitím detekce v posuvném okně. Dále je vhodné odstranit co největší část pozadí, ve kterém se postava nenachází. To může výrazně snížit oblast prohledávání. Následně ve snímku hledáme jednotlivé části těla: hlavu, tělo, předloktí, paže, atd. Každému pixelu se přiřadí pravděpodobnost, zda náleží dané části těla. Z přiřazených pravděpodobností je následně možné vyvodit, zda se jedná o osobu. Tento princip hledání částí těla byl použit v článku [13] a podobná metoda v článku [7].

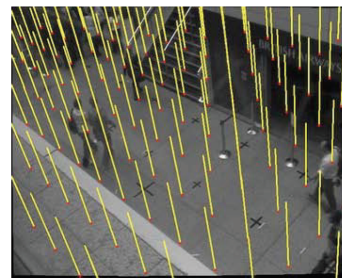


Obrázek 2.5: Detekce postavy v obraze (a). Odstranění pozadí (b - c). Přiřazení pravděpodobností pro části těla. Čím světlejší, tím větší pravděpodobnost, že pixel patří k části těla. Převzato z [13].

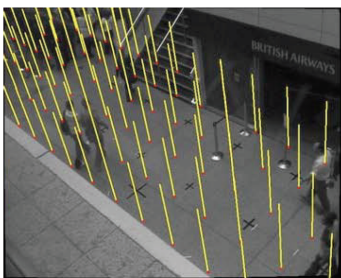
Dále lze detekci člověka z obrazu rozšířit o určení polohy ve 3D prostoru. Pokud známe pozici a natočení kamery, což by v případě většiny monitorovacích systémů mělo být známo, lze hledání objektu převést na prohledávání 3D prostoru místo 2D prostoru obrázku. Nejprve je potřeba vytvořit 3D mřížku, která bude pokrývat všechny možné pozice objektu ve scéně (obrázek 2.6b, 2.6c).



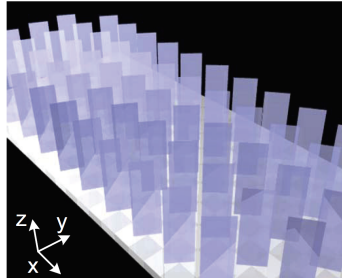
(a) : Originální snímek.



(b) : Automaticky generovaná mřížka.



(c) : Mřížka generovaná ve známé scéně.

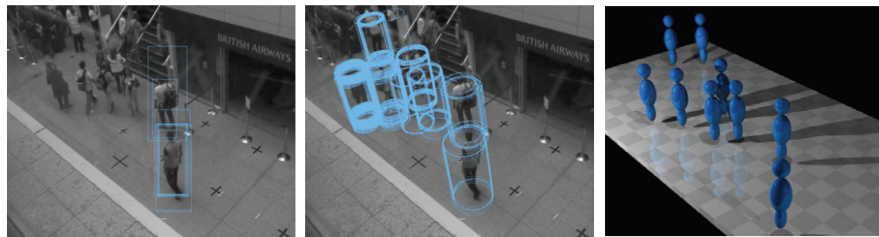


(d) : Natočení obrázku v každém bodě mřížky z pohledu kamery.

Obrázek 2.6: Tvorba 3D mřížky pro určení pozice. Převzato z [23].

V každém bodě mřížky se vytvoří příslušný výřez obrázku, který aproximuje vzhled scény. Rovina obrázku je kolmá na spojnici daného bodu mřížky a pozice kamery (obrázek 2.6d). Pokud je ve výřezu detekována postava, lze

následně z pozice obrázku v mřížce odhadnout pozici v prostoru. Tato metoda byla použita v článku [23].



Obrázek 2.7: Zleva původní snímek, detekované osoby, model s postavami na detekovaných pozicích. Převzato z [23].

Kapitola 3

Sledování trajektorie

3.1 Lokalizace

K tomu, aby robot mohl sledovat nějakou konkrétní trajektorii, je nutné, aby znal svou vlastní polohu. Za tímto účelem by bylo možné využít GPS. Ovšem v případě, že se robot bude pohybovat v libovolném prostředí, včetně uzavřených prostorů, nelze se na GPS příliš spoléhat. Proto je nutné využít jiných lokalizačních metod, například pomocí odometrie nebo metody SLAM.

3.1.1 Lokalizace pomocí odometrie

Odometrie je přímočarý systém lokalizace, který využívá informace z podvozku robotu. V případě diferenciálního podvozku se údaje o poloze dají považovat za spolehlivé, protože by nemělo docházet ke smýkání kol během jízdy. Ovšem v případě smykového podvozku jsou údaje velice nepřesné a je potřeba využít jiné metody pro lokalizaci.

Výpočet polohy dané kinematickým modelem pro dvoukolový diferenciální podvozek robotu [21] lze zapsat ve tvaru

$$x_{k+1} = x_k + D_c \cos(\Theta),$$

$$y_{k+1} = y_k + D_c \sin(\Theta),$$

$$\Theta_{k+1} = \Theta_k + \frac{D_r - D_l}{L}, \quad (3.1)$$

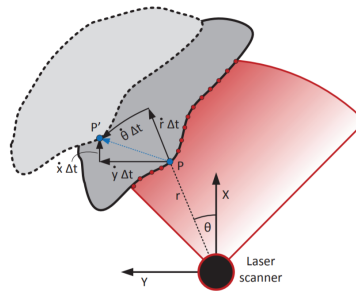
kde x , y jsou souřadnice polohy, Θ je natočení robotu, D_c , D_r , D_l jsou posuny robotu a pravého a levého kola, k představuje aktuální časový krok s konstantním vzorkovacím intervalem Δt , L je vzdálenost mezi koly. Tento kinematický model je založen pouze na prvním řádu Taylorova rozvoje

$$x(t + \Delta t) = x(t) + \dot{x}\Delta t. \quad (3.2)$$

Zahrnutím členu vyššího řádu Taylorova rozvoje do kinematického modelu dojde ke zpřesnění výpočtu polohy a orientace robotu.

3.1.2 Vizuální lokalizace

Získání informace o poloze robotu z jeho podvozku je jednoduchý a přímočarý. Ovšem tato metoda je velmi náchylná k chybám. Důvodem může být neustále narůstající chyba, kterou při delší jízdě již nelze zanedbat, nebo riziko prokluzu kola na nerovnostech povrchu. Pro potlačení těchto vlivů a zvýšení přesnosti lze využít tzv. vizuální odometrii [38], [20].



Obrázek 3.1: Princip odhadu změny polohy senzoru z po sobě následujících snímků. Převzato z [20].

Jedná se o metodu využívající senzoru, který je schopný snímat okolní prostředí, příkladem může být stereokamera, či laserový dálkoměr. Hlavní princip metody spočívá v porovnání po sobě jdoucích snímků, u kamery se porovnávají pozice příznaků v obraze, u laseru polohy bodů skenu. Následně dojde k nalezení korespondujících bodů a k odhadu jejich vzájemného posunu od posledního snímku tak, aby byl minimalizován rozdíl mezi snímky. Nalezený posun odpovídá pohybu senzoru mezi příslušnými snímky a tedy i pohybu robotu, na kterém je senzor připevněn.

3.1.3 Sjednocení soustav

Při určování pozice člověka za pomoci kamery jsou získané údaje o poloze relativní vzhledem k umístění kamery. Ta je připevněna pevně na robotu, jehož poloha se dynamicky mění a tím se mění i souřadná soustava, vzhledem ke které je získávána informace o poloze. Pro vytvoření požadované trajektorie, kterou má robot sledovat, je nutné převést naměřenou pozici člověka do stejné souřadné soustavy, v jaké se pohybuje robot. Toho lze dosáhnout následující transformací souřadnic

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_r \\ y_r \end{bmatrix}, \quad (3.3)$$

kde x_g, y_g je poloha člověka v souřadnicové soustavě, ve které se pohybuje robot, Θ je aktuální orientace robotu, x_r, y_r je aktuální poloha robotu, x, y je naměřená relativní pozice člověka vzhledem ke kameře.

3.2 Filtrace naměřených dat

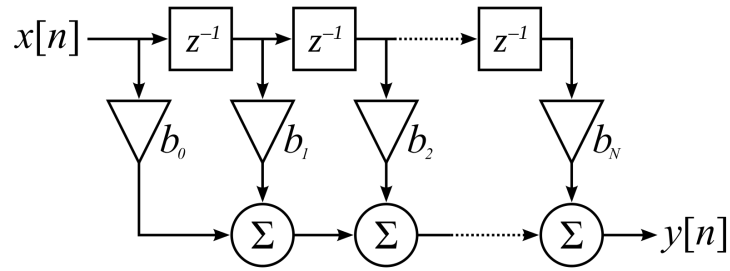
Při měření jakýchkoli dat je nutné předem počítat s nedokonalostmi reálných měřicích zařízení či přenosových médií. Tyto nedokonalosti, kterými může být například šum senzoru nebo občasná chybná měření, ovlivňují nasbírané hodnoty. K získání kvalitnějších hodnot lze na naměřená data aplikovat filtrovací metody, které dokáží částečně odstranit vliv nedokonalostí senzorů.

3.2.1 FIR filtr

Název FIR filtru vznikl jako zkratka z anglického názvu „finite impulse response“ [34], což znamená konečná impulzní odezva a je odvozen od způsobu výpočtu výstupu. Ten se spočítá jako suma konečného počtu vážených vstupů a to minulých, přítomných a případně i budoucích. Obecný předpis pro FIR filtr je

$$y[n] = \sum_{k=-M_1}^{M_2} b_k x[n-k], \quad (3.4)$$

kde y je výstup, n představuje časový okamžik diskrétního systému, M_1, M_2 nabývají konečných hodnot a v případě $M_1 = 0$ se jedná se o kauzální filtr, x je vstup, b_k je hodnota impulsní odezvy FIR filtru v k -tém okamžiku.



Obrázek 3.2: Schéma FIR filtru. Převzato z [37].

V případě rekonstrukce cesty z naměřených pozic člověka, není zásadní vysoká přesnost měření, ale spíše hladkost vytvořené trajektorie. K tomu lze použít speciální typ FIR filtru „moving average filter“, neboli klouzavý průměr. Jeho předpis lze napsat ve tvaru

$$y[n] = \frac{1}{N+1}(x[n] + x[n-1] + \dots + x[n-N]), \quad (3.5)$$

kde pro kauzální filtr je výstup počítán jako průměrná hodnota aktuálního vstupu a N posledních vstupů, tzn. koeficienty b_k z rovnice 3.4 mají pro všechny vstupy stejnou hodnotu.

3.2.2 Kalmanův filtr

Kalmanův filtr [6] je matematický aparát sloužící k filtraci signálů obsahující šum. Výhodou je, že dokáže pracovat v časové oblasti oproti jiným typům filtrů, které pracují ve frekvenční oblasti. Díky této vlastnosti odpadá výpočetně i implementačně náročná Fourierova transformace. Další výhodou je, že není nutné předem znát parametry rušení.

Pro výpočet je nutný stavový model systému, který se skládá ze dvou částí. Z modelu dynamiky stavu, který určuje závislost příštího stavu na aktuálním

$$x_n = Ax_{n-1} + Bu_n + w_n, \quad (3.6)$$

kde x reprezentuje stavy, u jsou vstupy modelu, w je bílý šum a A a B jsou stavové matice. Druhou částí je model výstupu, který určuje závislost aktuálního výstupu na aktuálním stavu

$$y_n = Cx_n + Du_n + v_n, \quad (3.7)$$

kde y je aktuální hodnota výstupu, v je bílý šum a C a D jsou stavové matice.

Kalmanův filtr je složen ze dvou hlavních částí. První částí je filtrace, jejímž úkolem je odhad středních hodnot prvků stavového vektoru. Kovarianční matice je zde dána rozptylem jednotlivých prvků stavového vektoru a udává přesnost, s jakou budou určeny prvky stavového vektoru. Druhou částí je predikce, jejímž úkolem je získat odhad středních hodnot stavových proměnných a hodnoty kovarianční matice pro další cyklus.

Rovnice Kalmanova filtru [17] jsou následující:

$$\hat{x}(t+1|t) = A\hat{x}(t|t) + Bu(t), \quad (3.8)$$

$$P(t+1|t) = AP(t|t)A^T + Q, \quad (3.9)$$

$$\hat{x}(t|t) = \hat{x}(t|t-1) + L(t)(y(t) - C\hat{x}(t|t-1) - Du(t)), \quad (3.10)$$

$$L(t) = P(t|t-1)C^T(CP(t|t-1)C^T + R)^{-1}, \quad (3.11)$$

$$P(t|t) = P(t|t-1) - P(t|t-1)C^T(CP(t|t-1)C^T + R)^{-1}CP(t|t-1). \quad (3.12)$$

Rovnice 3.8, 3.9 představují predikční část a rovnice 3.10, 3.11, 3.12 představují filtrační část. Matice A , B , C , D jsou stavové matice, \hat{x} je odhad stavu, Q , R jsou kovarianční matice modelu a měření, L je Kalmanovo zesílení, u je vstup do systému, y je výstup systému, P je kovarianční matice odhadu stavů.

■ 3.3 Křivky

V reálném světě lze velmi obtížně zajistit kontinuální proud informace o poloze sledovaného objektu. V případě použití kamery může objekt velmi snadno zmizet ze zorného pole. Například při ostrém úhlu zatočení, či když se objekt dostane za neprůhlednou překážku. V takových a podobných případech je nutné chybějící úsek trajektorie doplnit. To lze zajistit proložením tohoto úseku polynomem.

■ 3.3.1 Polynom prvního řádu

Nejjednodušší případ rekonstrukce chybějící části trajektorie je při proložení dvou bodů polynomem prvního řádu. Přímka zajistí nejkratší spojnicí dvou bodů. Ovšem nezařídí hladké navázání na předchozí trajektorii. Její rovnice má následující tvar

$$y = ax + b. \quad (3.13)$$

V případě že jsou známy dva body, je jednoduché dopočítat oba neznámé koeficienty a , b . Následně lze díky předpisu dopočítat libovolné množství bodů ležících na této přímce. Lze také využít informace o sklonu přímky vzhledem ke kladné poloose x z koeficientu a .

$$a = \operatorname{tg}(\alpha). \quad (3.14)$$

3.3.2 Polynom třetího řádu

Pro přirozenější a hladší pohyb robota lze chybějící úsek trajektorie nahradit polynomem třetího řádu. Kubická rovnice má následující předpis

$$y = ax^3 + bx^2 + cx + d. \quad (3.15)$$

Při hledání koeficientů a , b , c , d je nutné mít čtyři omezení, která zajistí unikátní řešení soustavy rovnic. První dvě jsou daná podmínkou, že polynom prochází dvěma požadovanými body. Pro získání dalších dvou omezení je nutné zvolit takové dva body, aby polynom získal požadovaný tvar.

Z výše zmíněných podmínek dostaneme soustavu 4 rovnic se 4 neznámými koeficienty a , b , c , d

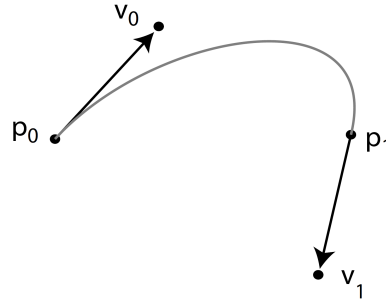
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \sim Y = Xa, \quad (3.16)$$

kde x_1, y_1, x_2, y_2 jsou krajní body a x_3, y_3, x_4, y_4 jsou body určující tvar hledané křivky. Koeficienty lze dopočítat z rovnice 3.16 vyjádřením neznámé a

$$a = X^{-1}Y. \quad (3.17)$$

3.3.3 Hermitovské křivky

Speciálním typem polynomů třetího řádu, které jsou často používány pro proložení dvou bodů, jsou tzv. Hermitovské křivky [14],[8]. Jejich hlavní výhodou oproti obyčejnému polynomu třetího řádu je, že jsou dány pouze dvěma krajními body a dvěma směrovými vektory v těchto bodech, nikoli čtyřmi body jako je tomu u obyčejného polynomu třetího řádu. Díky této vlastnosti je možné spojit libovolný počet Hermitovských křivek a získat tak hladkou trajektorii.



Obrázek 3.3: Proložení dvou bodů p_0 a p_1 kubickou Hermitovskou křivkou s tečnami v_0 a v_1 . Převzato z [14].

Obecný předpis Hermitovské funkce lze napsat v následujícím tvaru

$$p(x) = H_0^3(t)p_0 + H_1^3(t)v_0 + H_2^3(t)v_1 + H_3^3(t)p_1, \quad (3.18)$$

kde p_0, p_1 je počáteční, resp. koncový bod a v_0, v_1 jsou tečny v počátečním a koncovém bodě. $H_i^3, i = 0, 1, 2, 3$ jsou kubické Hermitovy polynomy, které mají následující tvar

$$H_0^3(t) = 2t^3 - 3t^2 + 1, \quad (3.19)$$

$$H_1^3(t) = t^3 - 2t^2 + t, \quad (3.20)$$

$$H_2^3(t) = t^3 - t^2, \quad (3.21)$$

$$H_3^3(t) = -2t^3 + 3t^2, \quad (3.22)$$

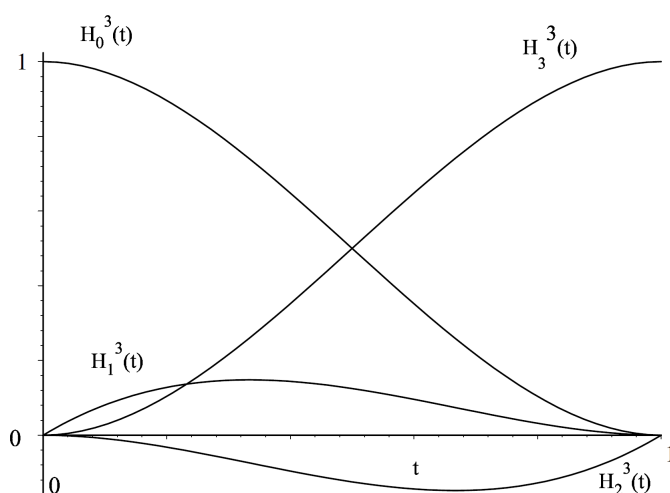
kde $t \in \langle 0, 1 \rangle$ tak, že v počátečním bodě p_0 je $t = 0$ a v koncovém bodě p_1 je $t = 1$. Jednotlivé kubické Hermitovy polynomy jsou zobrazeny na obrázku 3.4. Pro interpolaci na libovolném intervalu (x_k, x_{k+1}) je potřeba namapovat tento interval na jednotkový. Toho lze dosáhnout následujícím předpisem

$$t = \frac{x - x_k}{x_{k+1} - x_k}. \quad (3.23)$$

Pro snazší nalezení tečných vektorů v_0, v_1 lze využít podmnožinu Hermitovských křivek, tzv. „Cardinal splines“. Tato podmnožina křivek využívá k výpočtu tečných vektorů přímo řídicí body. To zjednodušuje práci s Hermitovskými křivkami. Předpis pro výpočet tečného vektoru v bodě p_k je

$$v_k = a(p_{k+1} - p_{k-1}), \quad (3.24)$$

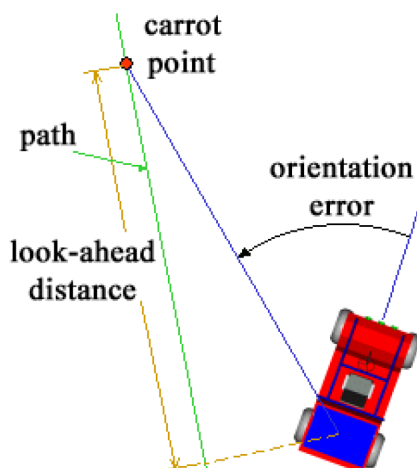
kde $a \in \langle 0, 1 \rangle$ určuje křivost křivky.



Obrázek 3.4: Kubické Hermitovy polynomy H_{0-3}^3 . Převzato z [14].

3.4 Follow the carrot

Algoritmus [25] je založen na jednoduchém principu, kdy je vybrán cílový bod z vytvořené cesty, ke kterému se robot následně nasměruje. Až se robot přiblíží k vybranému bodu s požadovanou přesností, tak algoritmus vybere nový cílový bod, ke kterému bude robot směřovat.



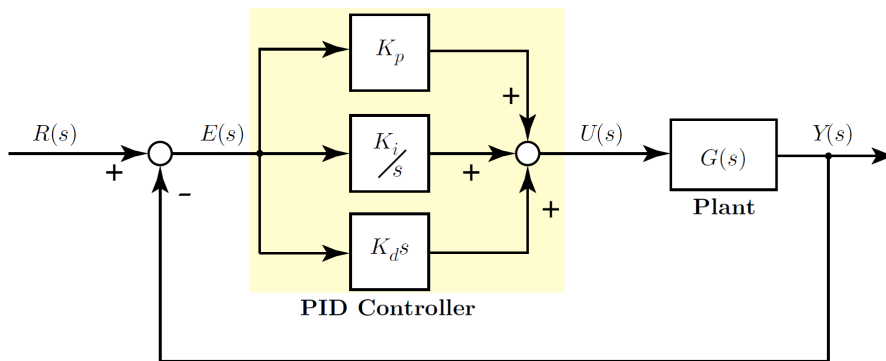
Obrázek 3.5: Follow the carrot algoritmus. Převzato z [25].

Nejdůležitějším parametrem je odchylka natočení, což je úhel mezi přímkou procházející polohou robotu a cílovým bodem a aktuálním natočením robotu. Následně se tuto odchylku snažíme minimalizovat.

Nevýhodou této metody je, že robot má tendence zkracovat zatáčky, což je způsobeno okamžitými změnami natočení při přiřazení nového cílového bodu. Další nevýhodou je riziko oscilace při jízdě podél cesty při vyšších rychlostech. Pro přesnější sledování, než poskytuje „follow the carot“ algoritmus, lze využít MPC řízení (Model Predictive Control), které je schopné nalézt optimální řešení pro přesný průjezd cesty. Ovšem náročnost implementace i výpočetní náročnost jsou vyšší.

3.5 PID regulátor

K řízení rychlostí robotu není vhodné používat pevné hodnoty rychlostí. Pro kvalitnější a přesnější jízdu je nutné dynamicky měnit rychlosti tak, aby se robot co nejméně odchyloval od naplánované cesty. K tomu se dá využít regulátoru typu PID.



Obrázek 3.6: Schéma systému řízeného PID regulátorem. Převzato z [27].

PID regulátor [4] je složen ze tří částí: proporcionální složka (P), integrační složka (I) a derivační složka (D), od kterých je odvozen jeho název. Jedná se o zpětnovazební regulátor, který počítá odchylku mezi naměřenou hodnotou a požadovanou hodnotou a tuto odchylku se snaží minimalizovat. Proporcionální složka je závislá na aktuální odchylce, integrační složka závisí na naakumulovaných minulých odchylkách a derivační složka závisí na aktuální změně odchylky. Rovnici výstupu regulátoru lze zapsat

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de}{dt}, \quad (3.25)$$

kde u je výstupem regulátoru, který je zároveň vstupem do systému, e je odchylka, K_p , K_i , K_d jsou regulační konstanty.

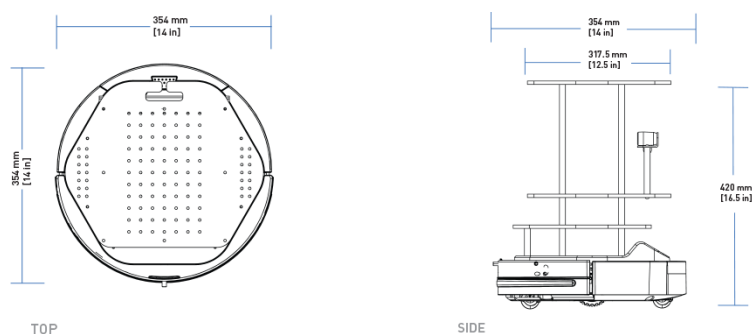
V některých případech může být vhodnější použít jen některé části regulátoru, jako například pouze P složku. Lze také použít kombinace částí, například PI nebo PD, v případě, když chceme zaručit stabilitu a přesnost, či rychlejší start.

Kapitola 4

Implementace

4.1 Turtlebot 2 se systémem ROS

Pro realizaci projektu byl jako testovací platforma zvolen robot Turtlebot 2 [30]. Robot byl vyvinut pro výukové a výzkumné účely. Díky jeho malé velikosti je vhodný pro testování v menších prostorech. Turtlebot je schopen pohybovat se rychlostí až 0.65 m/s a otáčet se rychlostí až 180 °/s. Je schopen uvést až 5 kg, přičemž sám váží 6.3 kg. Robot funguje na systému ROS a byl vyvíjen ve spolupráci přímo s vývojáři tohoto systému.



(a) : Rozměry těla robota Turtlebot shora.

(b) : Výška a šířka podpůrné konstrukce pro senzory.

Obrázek 4.1: Rozměry robota Turtlebot. Převzato z [30].

ROS (Robot Operating System) [5] není klasický operační systém, ale spíše komunikační nadstavba, která dokáže zajistit nízkouúrovňové ovládání připojených zařízení, či komunikaci mezi spuštěnými procesy. Každý proces

běží v tzv. uzlu (Node) a každý uzel může přijímat data z různých senzorů a z jiných uzlů. Uzel také může posílat příkazy do různých připojených zařízení.

4.2 Kamera

Pro získání obrazu byla použita kamera Basler daA1280-54um s charakteristickými vlastnostmi uvedenými v tabulce 4.1.

Parametry kamery			
Rozlišení	1280 px x 960 px	Velikost pixelu	3.75 μm x 3.75 μm
Velikost senzoru	4.8 mm x 3.6 mm	FPS	54

Tabulka 4.1: Základní parametry použité kamery.

Ke kameře byly připevněny postupně dva různé objektivy: Computar s úhlem záběru 63.9° x 49.1° (horizontální x vertikální) a širokoúhlý TV Lens.



(a) : Kamera Basler. (b) : Běžný objektiv. (c) : Širokoúhlý objektiv.

Obrázek 4.2: Použitá kamera s dvěma typy objektivů.

Před použitím pro detekci bylo nutné kameru zkalibrovat individuálně s oběma objektivy. Pro kalibraci obyčejného objektivu byl použit ROS-balíček *camera_calibration* [2]. Avšak širokoúhlý objektiv nelze příliš dobře kalibrovat běžnými metodami, proto bylo třeba použít sofistikovanější *Apriltag kalibraci* [1]. Hodnoty nalezené kalibrací jsou uvedeny v tabulce 4.2.

Po provedení kalibrace dojde k úpravě velikosti záběru kamery, jelikož se příliš deformované části na okraji obrazu oříznou. Nový úhel záběru lze vypočítat z parametrů získaných při kalibraci společně s informacemi o senzoru podle následujícího vzorce

$$\alpha = 2\arctan\left(\frac{d}{2f}\right), \quad (4.1)$$

Parametry kalibrace		
Parametr	Objektiv Computar	Objektiv TV Lens
Ohnisková vzdálenost f_x	1085.68 [px]	580.84 [px]
Ohnisková vzdálenost f_y	1083.25 [px]	581.04 [px]
Hlavní bod c_x	627.21 [px]	623.17 [px]
Hlavní bod c_y	492.92 [px]	476.36 [px]
Parametr zkreslení k_1	-0.213810	-0.280926
Parametr zkreslení k_2	0.144074	0.075348
Parametr zkreslení t_1	-0.000775	0.000509
Parametr zkreslení t_2	-0.001739	-0.000741
Parametr zkreslení k_3	0	-0.008695

Tabulka 4.2: Parametry získané při kalibraci kamery.

kde α je úhel záběru, d je velikost senzoru v milimetrech a f je ohnisková vzdálenost v milimetrech. Ohniskovou vzdálenost je nutné nejprve přepočítat z pixelů na milimetry, což lze pomocí následujícího předpisu

$$f = \frac{f_p}{pm}, \quad (4.2)$$

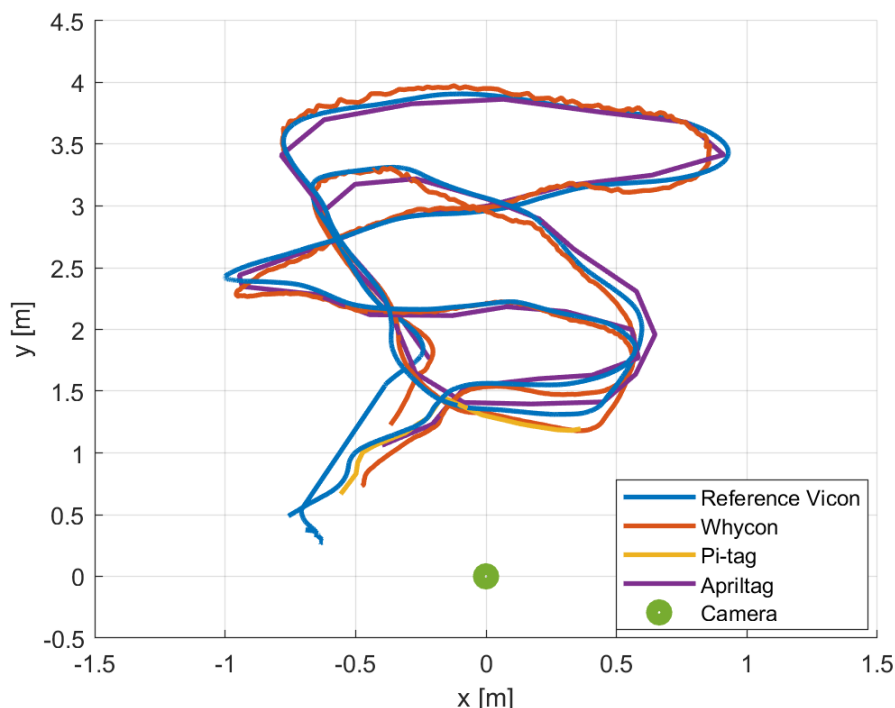
kde f_p je ohnisková vzdálenost v pixelech, pm je počet pixelů na milimetr. Vypočtené úhly záběru jsou uvedené v tabulce 4.3.

Úhly záběru kamery		
Objektiv	Horizontální úhel [°]	Vertikální úhel [°]
Computar	61.0380	47.7973
TV Lens	95.5486	79.1206

Tabulka 4.3: Vypočítané úhly záběru kamery s použitými objektivy.

4.3 Testování vlastností značek

Při výběru vhodné značky pro zadaný problém byly pro otestování vlastností zvoleny 3 druhy značek: Apriltag, Pi-tag a Whycon. Značky byly umístěny na pevnou podložku tak, aby nedocházelo k jejich deformaci, která by zhoršila kvalitu detekce. Jako referenční systém byl použit velice přesný a rychlý lokalizační systém Vicon, který k lokalizaci využívá shluk reflexních bodů. Jeden shluk byl umístěn ke kameře a druhý na podložku ke značkám, tudíž po celou dobu měření byla zajištěna přesná lokalizace vzhledem ke kameře, následně použitá jako referenční, a konstantní transformace mezi referenčním systémem a značkami.



Obrázek 4.3: Trajektorie značek při testování jejich vlastností.

Z obrázku 4.3 je na první pohled vidět, že značka Pi-tag byla detekována pouze v oblasti přibližně do 1.5 m, tudíž není příliš vhodná. Apriltag a Whycon byly detekovatelné bez problémů i ve vzdálenosti 4 m.

Test vlastností značek		
Značka	Detekce [Hz]	Průměrná odchylka [m]
Whycon	25.38	0.073
Apriltag	1.38	0.293

Tabulka 4.4: Parametry klíčových vlastností značek.

Jak je vidět z tabulky 4.4 Apriltag byl detekován s výrazně nižší frekvencí než Whycon, což způsobilo i vyšší odchylku, protože do odchylky je nutné započítat také posun za dobu zpracování snímku, protože v okamžiku, kdy dojde ke získání informace o poloze značky, se značka již v tomto místě nenachází. Dle výsledků pokusu byla pro další experimenty vybrána značka Whycon.

V průběhu dalších experimentů byly zaznamenány tendence detektoru značky Whycon chybně lokalizovat značku v obraze. V několika případech došlo k záměně značky s podobnými tvary, například s podobnými písmeny na krabici, či s kruhovou zásuvkou, apod.

4.4 Rekonstrukce cesty

Pro lokalizaci byla použita odometrie robotu Turtlebot 2, která je dostatečně přesná vzhledem k délce experimentů prováděných v laboratoři. Dále je při rekonstrukci trajektorie nutné počítat s úseky, během kterých nebyl sledovaný objekt v záběru kamery. V takových případech je nutné chybějící úsek nahradit. K tomu byly použity Hermitovy křivky dle kapitoly 3.3.3, jejichž směr v hraničních bodech byl dán rovnicí 3.24, kde parametr a byl nastaven na hodnotu 0.5.

Při měření polohy pomocí kamery obsahují naměřené pozice různé nepřesnosti a šum. K co největšímu potlačení těchto nežádoucích vlivů a k vyhlazení trajektorie lze využít filtraci pomocí FIR filtru dle kapitoly 3.2.1. Konkrétně byl použit klouzavý průměr („running average filter“) dle rovnice 3.5. Konkrétní tvar rovnice byl

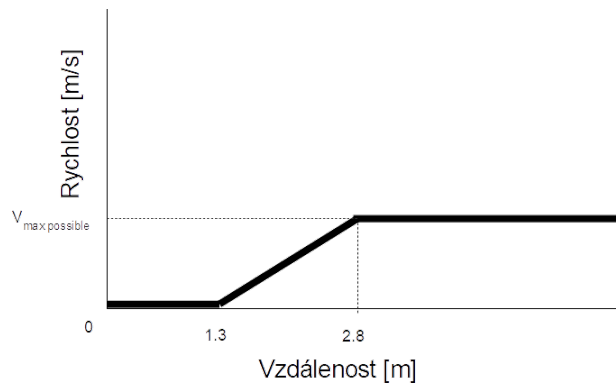
$$y_{fir7}[n] = \frac{1}{7}(x[n] + x[n-1] + \dots + x[n-6]). \quad (4.3)$$

V případě ztráty sledovaného objektu ze záběru kamery, je nutné při znovunalezení objektu zkontrolovat vzdálenost bodů, které jsou vstupem do filtru. V případě, že by body byly příliš vzdálené došlo by k nežádoucímu velkému zkreslení výstupního bodu. Proto je nutné v takovém případě vypnout filtraci. Znovu zapnuta je v případě, kdy je počet vstupních bodů $n \geq 3$ a jejich vzdálenost není příliš velká.

4.5 Metoda sledování cesty

Pro sledování vytvořené cesty byla zvolena jednodušší přímočará metoda založená na metodě „Follow the carrot“. Celá cesta byla uložena ve formě sekvence bodů, ze kterých byl cílový bod vybrán dle nastavené vzdálenosti, se kterou by se měl nacházet před robotem.

Po vybrání cílového bodu byl zjištěn směr, kterým se má robot vydat. Pro natočení správným směrem byl použit PI regulátor s hodnotou proporcionální složky $k_p = 2.5$ a integrační složky $k_i = 0.04$. Výstupem regulátoru je úhlová rychlost robotu ω . Díky známému poloměru robotu $r_r = 0.115$ m



Obrázek 4.4: Průběh rychlosti v závislosti na vzdálenosti od sledované osoby.

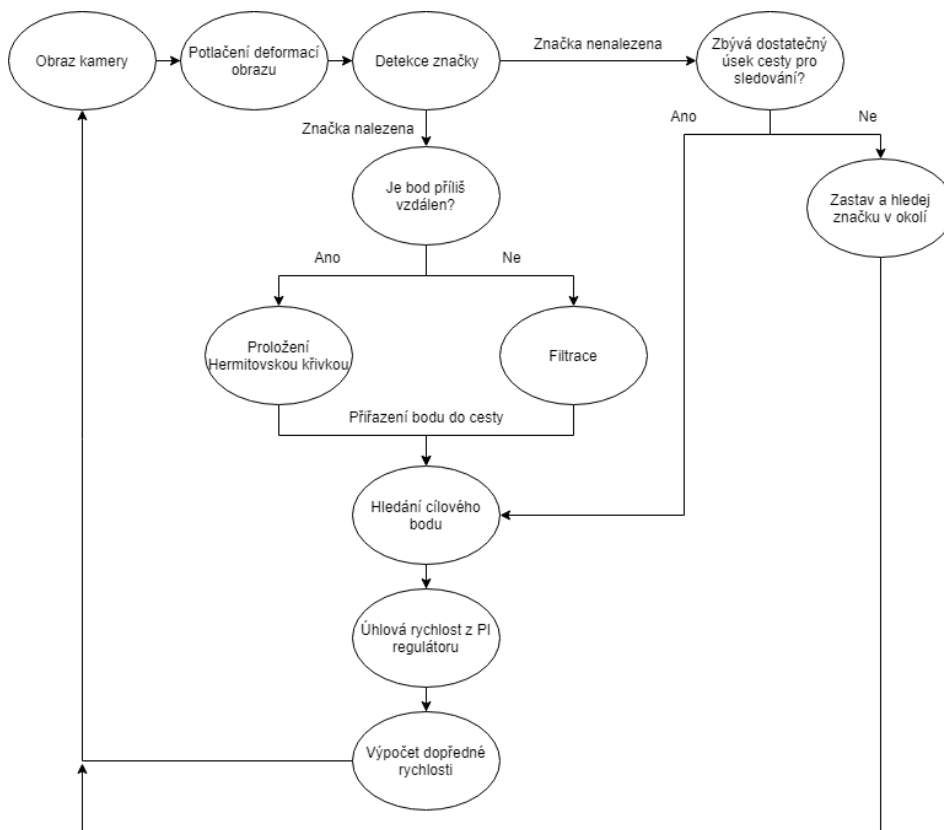
a známé maximální rychlosti robotu $v_{r,max} = 0.65$ m/s, lze dopočítat zbývající možnou dopřednou rychlost $v_{max\,possible}$ tak, aby se robot stále pohyboval s úhlovou rychlostí ω . Rovnice pro výpočet této rychlosti je

$$v_{max\,possible} = v_{r,max} - |\omega r_r|. \quad (4.4)$$

Robot se začne pohybovat po vytvořené trajektorii, pokud je jeho vzdálenost od posledního bodu trajektorie větší než 1.3 m. Jeho dopředná rychlost je určena dle vzdálenosti od tohoto hraničního bodu a mění se lineárně tak, že ve vzdálenosti 1.5 m od hraničního bodu pro sledování dosáhne robot maximální možné dopředné rychlosti $v_{max\,possible}$. Průběh rychlosti je ukázán na obrázku 4.4.

4.6 Řídící smyčka algoritmu

Celý řídicí algoritmus byl implementován v c++ jazyce a využívá systému ROS. Projekt je rozdělen do několika částí (balíčků dle ROS systému), kde každá má na starost určitou část algoritmu, jako jsou například snímání obrazu, detekce značky, nebo výpočet a přiřazení rychlostí robotu. Jednotlivé části si vzájemně posílají zprávy s daty, které následně zpracují a informaci předají dále. Celé řízení běží v nekonečné smyčce s periodou 0.1 s.

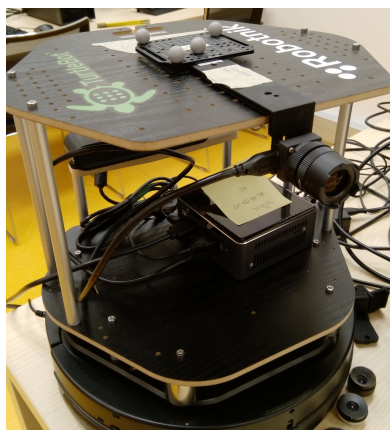


Obrázek 4.5: Schéma řídicí smyčky algoritmu se všemi částmi.

Kapitola 5

Vyhodnocení reálných pokusů

Celý algoritmus popsaný v předchozích kapitolách byl otestován při reálných pokusech, při nichž byl použit robot Turtlebot 2 s kamerou Basler. Pokusy byly provedeny se dvěma různými objektivy, z nichž jeden byl širokoúhlý. Pro určení polohy člověka byla použita značka Whycon. Jako referenční lokalizační systém byl použit velice přesný lokalizační systém Vicon. Podrobnější informace o použitém hardwaru a softwaru jsou uvedeny v předchozí kapitole Implementace 4.

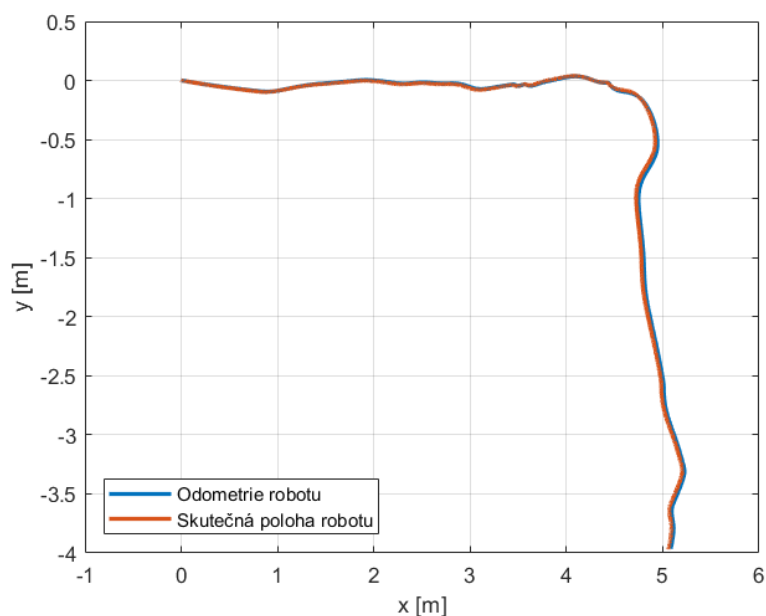


Obrázek 5.1: Turtlebot 2 s kamerou Basler.

5.1 Ostrá zatáčka

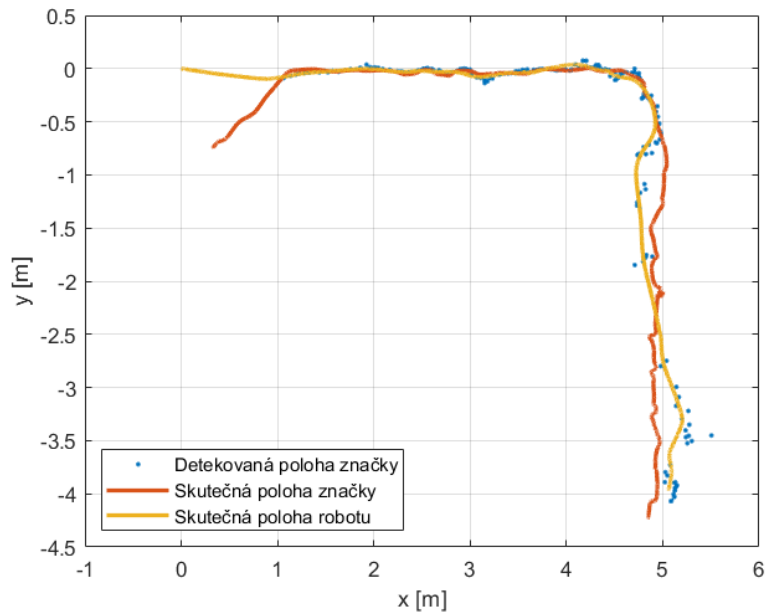
Prvním z pokusů bylo sledování, při kterém člověk prudce odbočil a vytvořil tak přibližně pravoúhlou zatáčku. Cílem tohoto pokusu bylo zjistit, jakým způsobem se změní přesnost sledování robotu při ztrátě značky z obrazu kamery a jak bude ovlivněno celkové chování.

5.1.1 Kamera s obyčejným objektivem

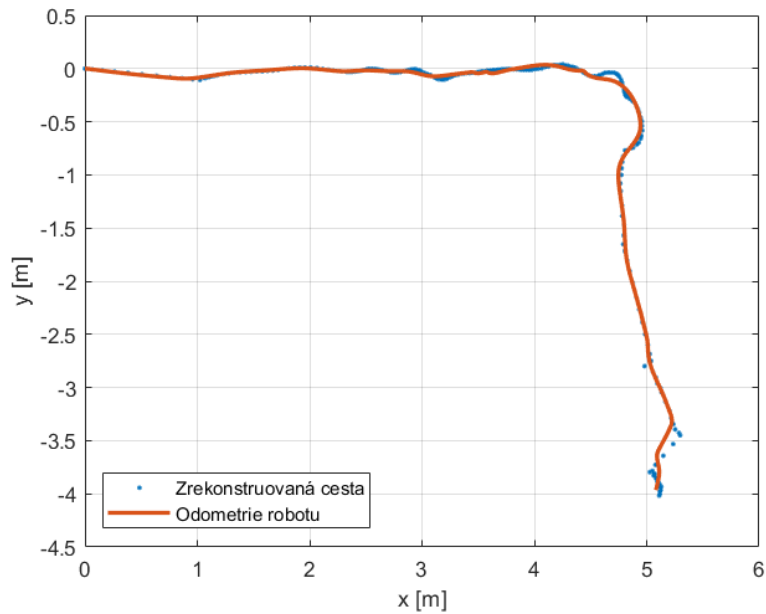


(a) : Porovnání odometrie robotu a jeho skutečné polohy pro ostrou zatáčku s obyčejným objektivem.

Jak je vidět z grafu 5.2a odometrie, kterou robot používá k lokalizaci, odpovídá poměrně přesně reálné poloze. Její odchylky od reálné polohy se pohybovali v rozmezí do několika jednotek centimetrů. Z grafu 5.2c lze vidět, že robot také poměrně přesně sleduje vytvořenou cestu. Jeho odchylka od sledované trajektorie se v průměru pohybovala okolo 2 cm. Ovšem z grafu 5.2b lze vidět, že ve druhé polovině cesty, kde robot ztratil značku ze záběru, došlo k odchylce skutečné polohy robotu až 30 cm od skutečné polohy značky. To je způsobeno nepřesnými detekcemi poloh značky, jak lze v grafu vidět.



(b) : Porovnání detekované polohy značky, její skutečné polohy a skutečné polohy robotu pro ostrou zatáčku s obyčejným objektivem.



(c) : Porovnání odometrie a sledované cesty pro ostrou zatáčku s obyčejným objektivem.

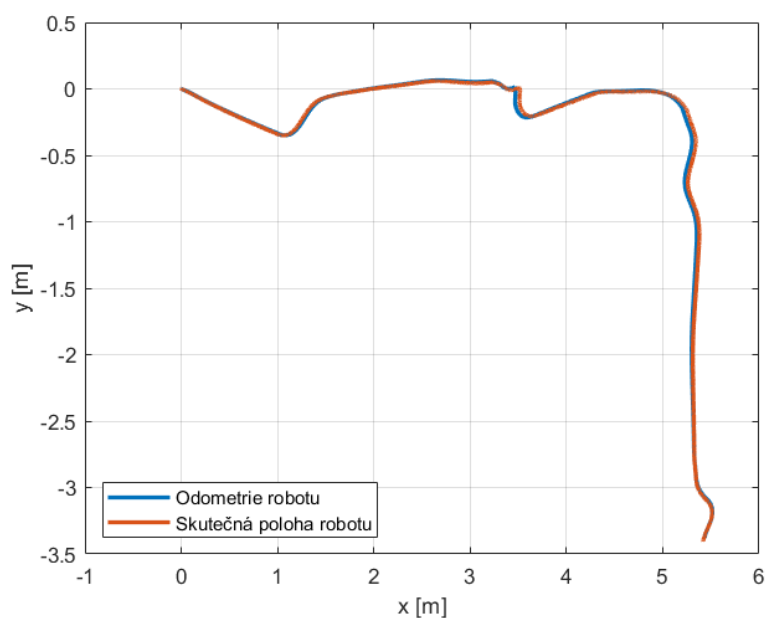
Obrázek 5.2: Porovnání skutečných pozic a zkrešených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě $[0, 0]$. Pokus: ostrá zatáčka, obyčejný objektiv.

Odchylky - ostrá zatáčka - obyčejný objektiv		
	Průměrná [m]	Maximální [m]
Odchylka odometrie a skutečné polohy robotu	0.010	0.041
Odchylka detekované polohy a skutečné polohy značky	0.124	0.539
Odchylka skutečných poloh značky a robotu	0.089	0.301
Odchylka odometrie robotu a vytvořené cesty	0.019	0.111

Tabulka 5.1: Vypočítané odchylky k jednotlivým grafům na obrázku 5.2.

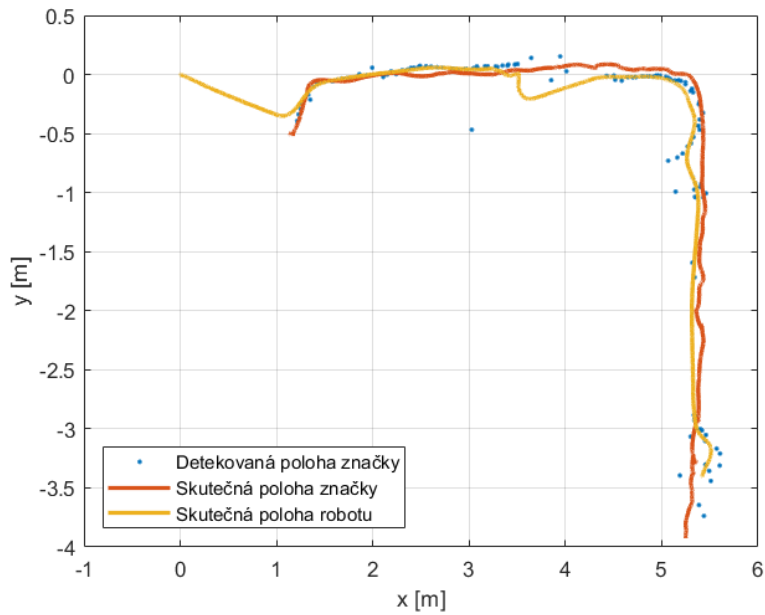
Průměrná odchylka detekované polohy značky od skutečné polohy se pohybovala okolo 12 cm, přičemž maximální odchylka byla až 54 cm. K nárůstu průměrné odchylky došlo především ve druhé části trajektorie. Nepřesnosti byly pravděpodobně způsobeny detekcí značky v krajní části obrazu kamery, kde může být obraz pokrivený nedokonalou kalibrací, čímž dojde ke zkreslení údajů o poloze.

5.1.2 Kamera s širokoúhlým objektivem

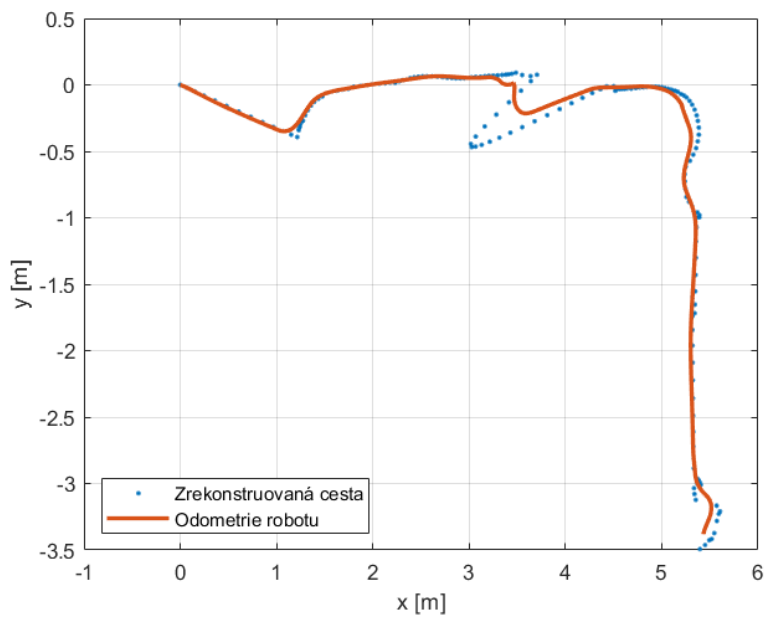


(a) : Porovnání odometrie robotu a jeho skutečné polohy pro ostrou zatáčku s širokoúhlým objektivem.

Jak je vidět z grafu 5.3a odometrie přibližně kopíruje skutečnou pozici.



(b) : Porovnání detekované polohy značky, její skutečné polohy a skutečné polohy robotu pro ostrou zatáčku s širokoúhlým objektivem..



(c) : Porovnání odometrie a sledované cesty pro ostrou zatáčku s širokoúhlým objektivem.

Obrázek 5.3: Porovnání skutečných pozic a zkrešených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě $[0, 0]$. Pokus: ostrá zatáčka, širokoúhlý objektiv.

Odchytky - ostrá zatačka - širokoúhlý objektiv		
	Průměrná [m]	Maximální [m]
Odchytky odometrie a skutečné polohy robotu	0.030	0.064
Odchytky detekované polohy a skutečné polohy značky	0.094	0.474
Odchytky skutečných poloh značky a robotu	0.076	0.226
Odchytky odometrie robotu a vytvořené cesty	0.068	0.521

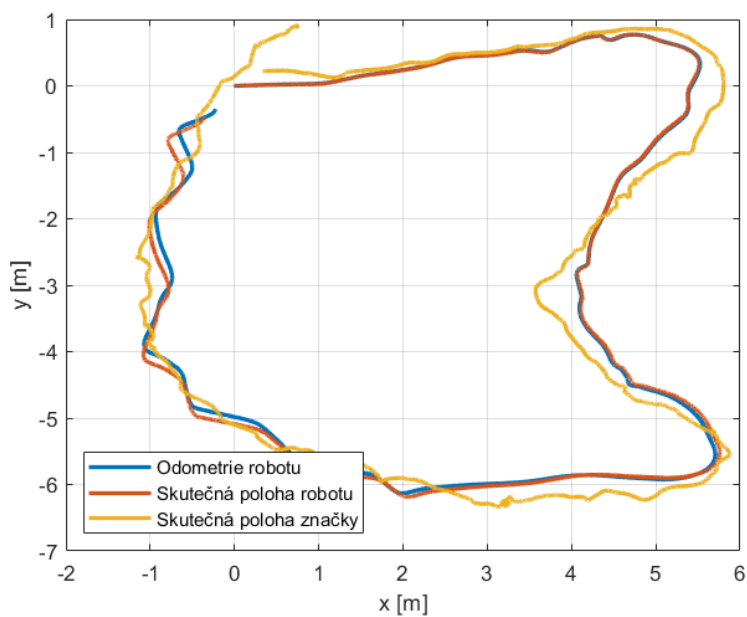
Tabulka 5.2: Vypočítané odchytky k jednotlivým grafům na obrázku 5.3.

Z grafu 5.3c lze vidět, že robot cestu sleduje opět poměrně přesně, ačkoli maximální odchytky dosahuje hodnoty 52 cm. Tato vysoká odchytky byla způsobena metodou, kterou si robot vybírá aktuální cílový bod. Nejedná se tedy o nepřesnost sledování, ale o žádoucí efekt metody výběru aktuálního cílového bodu. Z grafu 5.3b je navíc vidět, že pozice tohoto bodu byla pravděpodobně špatně detekována a zároveň jeho poloha splnila podmínky pro vypnutí filtrace popsané v kapitole 4.4 ve třetím odstavci. V grafu 5.3b jsou dále ukázány skutečné pozice robotu a značky, jejich vzájemné odchytky jsou mírně menší než v případě prvního objektivu.

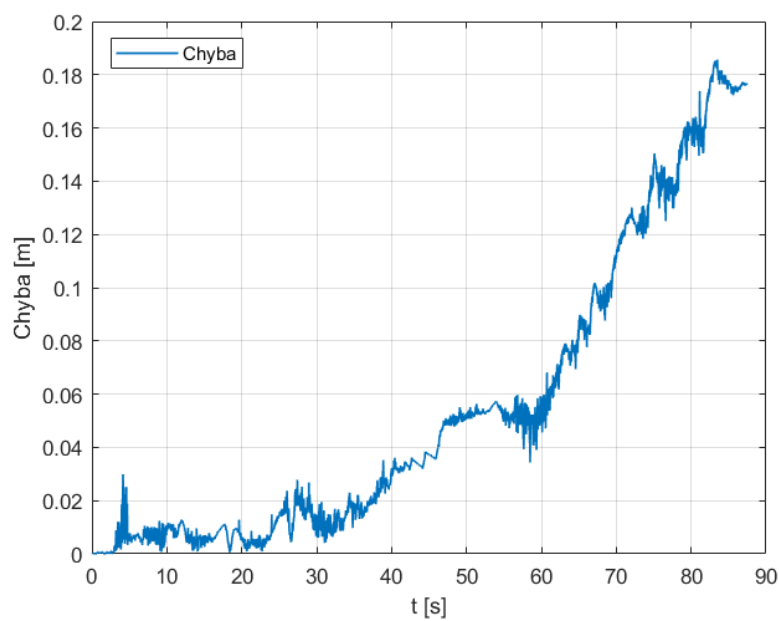
Při porovnání hodnot odchylek obou objektivů je vidět, že odchytky pro detekovanou polohu značky a skutečnou polohu značky jsou o něco nižší u širokoúhlého objektivu, stejně tak jsou nižší odchytky skutečných poloh robotu a značky. Ovšem z grafů, a také z pozorování při experimentu, je vidět, že širokoúhlý objektiv je více náchylný ke špatným detekcím, které způsobují odchýlení od skutečné trajektorie pohybu značky.

5.2 Sledování delší trajektorie

Druhým pokusem bylo sledování člověka po delší obecné trajektorii. Cílem bylo zjistit, jaký vliv na přesnost bude mít delší sledování cesty, při kterém by se měly výrazněji projevit kumulované chyby.



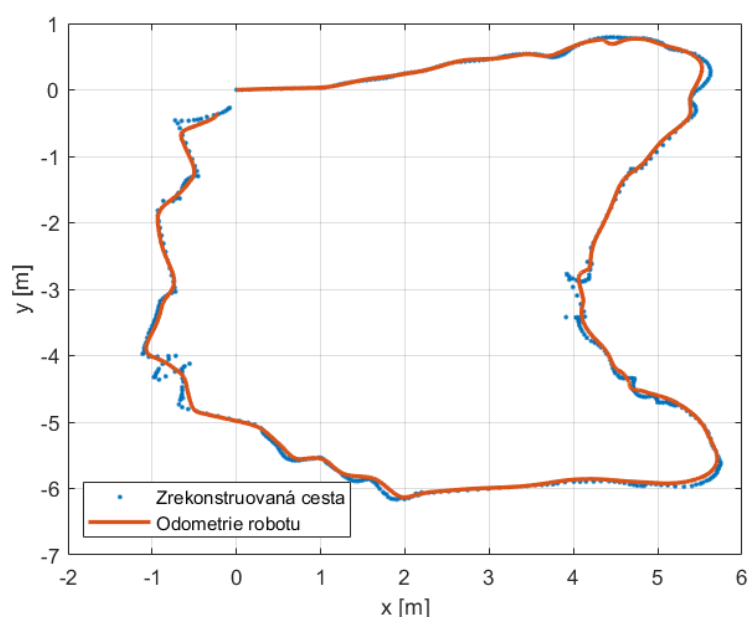
(a) : Porovnání odometrie robotu, jeho skutečné polohy a detekované polohy značky pro delší cestu s obyčejným objektivem.



(b) : Průběh odchylky odometrie od skutečné pozice robotu pro delší cestu s obyčejným objektivem.

5.2.1 Kamera s obyčejným objektivem

Z grafu 5.4a lze vidět, že v první polovině cesty si trajektorie pohybu robota měřená odometrií i skutečná poloha měřená systémem Vicon poměrně přesně odpovídají. Ovšem ve druhé části je již zřetelně vidět, jak se údaje o poloze začínají lišit. Průměrná chyba byla 5.5 cm a maximální chyba se dostala téměř až na 19 cm. Průběh nárůstu chyby lze vidět v grafu 5.4b. Z grafu 5.4c je možné vidět, že uloženou cestu robot sleduje velmi přesně bez ohledu na její délku. Z grafu 5.4a je vidět, že skutečné polohy robotu a značky přibližně



(c) : Porovnání odometrie a sledované cesty pro delší cestu s obyčejným objektivem.

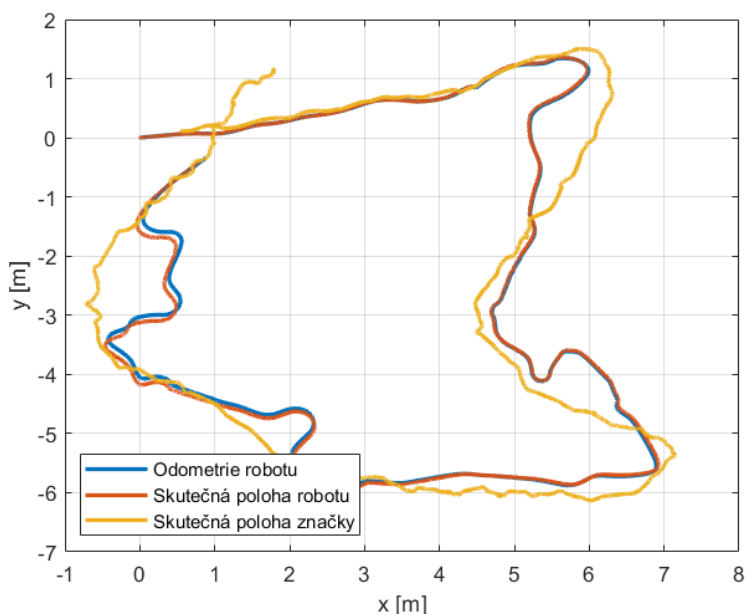
Obrázek 5.4: Porovnání skutečných pozic a zkreslených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě $[0, 0]$. Pokus: delší trajektorie, obyčejný objektiv.

Odchylky - delší trajektorie - obyčejný objektiv		
	Průměrná [m]	Maximální [m]
Odchylka odometrie a skutečné polohy robotu	0.055	0.186
Odchylka skutečných poloh značky a robotu	0.143	0.536
Odchylka odometrie robotu a vytvořené cesty	0.039	0.259

Tabulka 5.3: Vypočítané odchylky k jednotlivým grafům na obrázku 5.4.

kopírují stejnou trajektorii, ovšem jejich odchylka je již větší než v předchozím pokusu. Průměrná odchylka byla 14 cm a maximální 53 cm.

5.2.2 Kamera s širokoúhlým objektivem

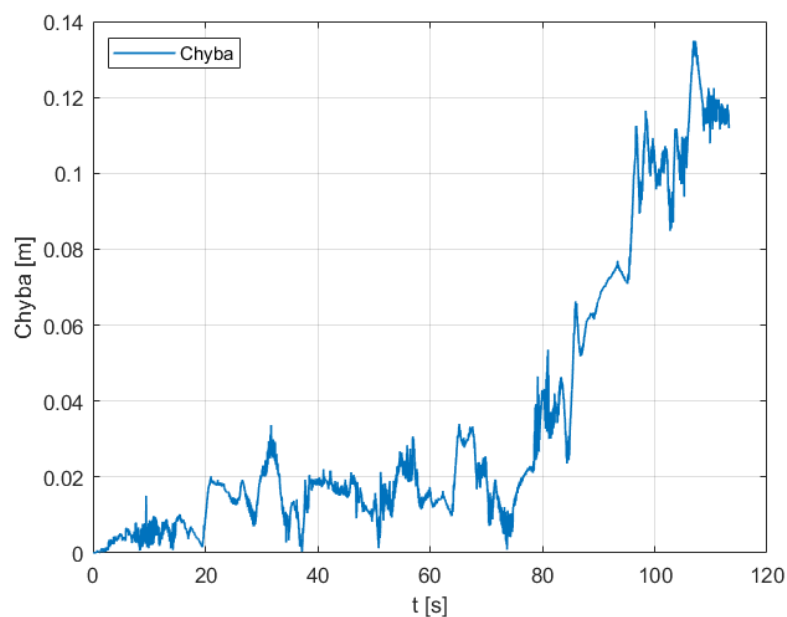


(a) : Porovnání odometrie robotu, jeho skutečné polohy a detekované polohy značky pro delší cestu s širokoúhlým objektivem.

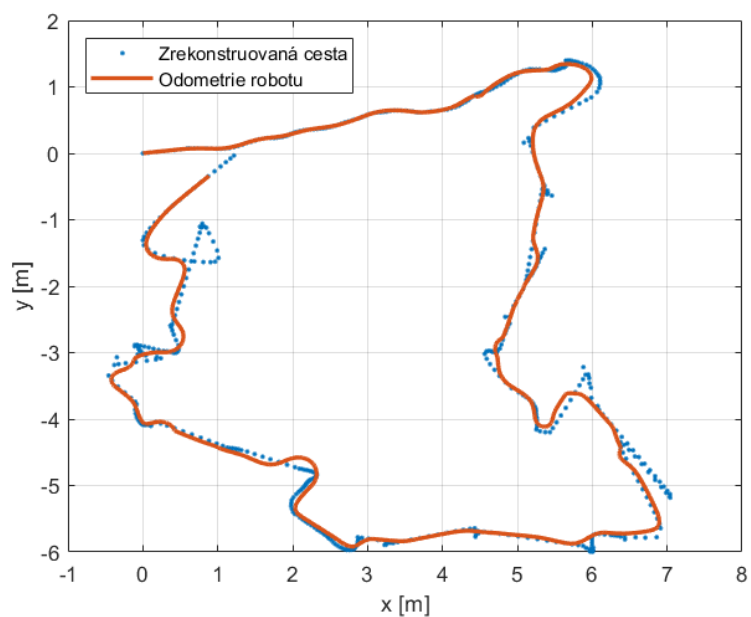
Odchylky - delší trajektorie - širokoúhlý objektiv		
	Průměrná [m]	Maximální [m]
Odchylka odometrie a skutečné polohy robotu	0.035	0.135
Odchylka skutečných poloh značky a robotu	0.246	0.931
Odchylka odometrie robotu a vytvořené cesty	0.069	0.569

Tabulka 5.4: Vypočítané odchylky k jednotlivým grafům na obrázku 5.5.

Z grafu 5.5a lze opět vidět, že po ujetí delšího úseku dochází k nárůstu odchylky odometrie robotu a skutečné pozice. Průběh nárůstu lze vidět v grafu 5.5b. Z grafu 5.5c je možné vidět, že vytvořenou cestu sleduje poměrně přesně. Maximální odchylka byla 57 cm, což bylo způsobeno přeskočením úseku trajektorie vhodnou volbou cílového bodu. Z grafu 5.5c lze dále vidět, že trajektorie vytvořené pohybem robotu a značky jsou podobné, ovšem v pohybu robota jsou vidět náhlé výchylky, které byly pravděpodobně způsobeny špatnou detekcí značky.



(b) : Průběh odchylky odometrie od skutečné pozice robotu pro delší cestu s širokoúhlým objektivem.



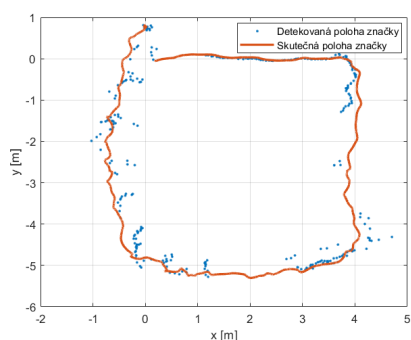
(c) : Porovnání odometrie a sledované cesty pro delší cestu s širokoúhlým objektivem.

Obrázek 5.5: Porovnání skutečných pozic a zkrešených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě $[0, 0]$. Pokus: delší trajektorie, širokoúhlý objektiv.

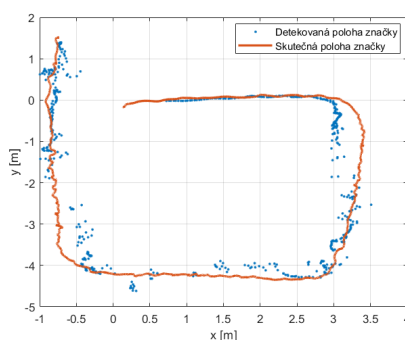
Při tomto pokusu byly odchylky u širokoúhlého objektivu větší než u obyčejného. Z grafů je vidět, že se robot několikrát náhle odchýlil od trajektorie pohybu značky. K těmto vychýlením pravděpodobně došlo špatnou detekcí značky a zároveň byly pravděpodobně důvodem zvýšení hodnot odchylek. Z grafů a z pozorování při experimentu se chování robotu s obyčejným objektivem jevílo jako stabilnější a předvídatelnější oproti širokoúhlému, který se několikrát náhle odchýlil od skutečné trajektorie značky.

5.3 Test vlivu četnosti detekce na sledování

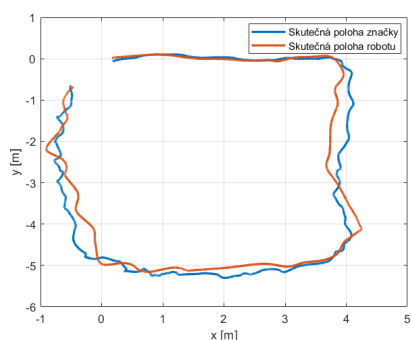
V prvním případě se osoba se značkou pohybuje bez omezení. V druhém případě se snaží pohybovat pouze v případě, že je robot natočen jejím směrem tak, aby byl pohyb značky co nejvíce vidět v záběru kamery. Cílem tohoto experimentu je zjistit, zda zvýšený počet detekcí a trajektorie s méně neznámými úseky bude mít vliv na přesnost sledování.



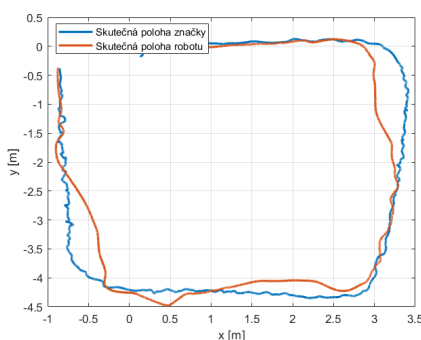
(a) : Normální pohyb osoby se značkou. Počet detekcí je přibližně poloviční oproti 5.6b.



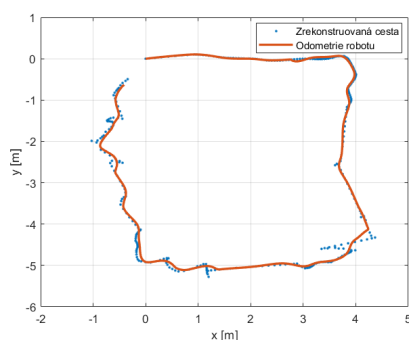
(b) : Pohyb osoby tak, aby kamera zaznamenala co největší část jejího pohybu.



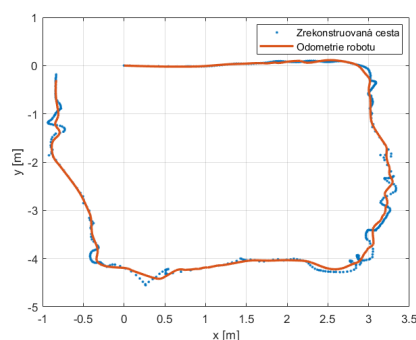
(c) : Porovnání skutečných poloh robotu a značky pro normální detekci.



(d) : Porovnání skutečných poloh robotu a značky pro zvýšenou detekci.



(e) : Porovnání odometrie robotu a vytvořené cesty pro normální detekci.



(f) : Porovnání odometrie robotu a vytvořené cesty pro zvýšenou detekci.

Obrázek 5.6: Porovnání skutečných pozic a zkrešených pozic, se kterými pracuje robot. Počáteční pozice robotu byla v bodě $[0, 0]$. Pokus: vliv četnosti detekcí.

Odchylky - obyčejný pohyb člověka		
	Průměrná [m]	Maximální [m]
Odchylka detekované polohy a skutečné polohy značky (obr. a)	0.212	0.622
Odchylka skutečných poloh značky a robotu (obr. c)	0.089	0.353
Odchylka odometrie robotu a vytvořené cesty (obr. e)	0.043	0.328
Odchylky - pohyb se zvýšenou detekcí		
Odchylka detekované polohy a skutečné polohy značky (obr. b)	0.198	0.429
Odchylka skutečných poloh značky a robotu (obr. d)	0.118	0.407
Odchylka odometrie robotu a vytvořené cesty (obr. f)	0.034	0.180

Tabulka 5.5: Vypočítané odchylky k jednotlivým grafům na obrázku 5.6.

Počet vzorků detekovaných poloh značky byl v druhém případě přibližně dvakrát větší. Z grafů 5.6a, 5.6b lze vidět, že v obou případech je mnoho detekovaných poloh nepřesných. Z tabulky odchylek 5.5 lze vidět, že jejich hodnoty jsou přibližně stejné v obou případech. Dále z grafů 5.6e, 5.6f je vidět, že robot zrekonstruovanou cestu sledoval poměrně přesně v obou případech, odchylky se opět pohybovaly v podobných hodnotách. Odchylky skutečných poloh robotu a značky měly podobné hodnoty v obou případech. Ovšem v prvním případě byly odchylky mírně menší. Tuto skutečnost lze spíše vyhodnotit jako náhodný jev, který byl způsobený mírně odlišnou trajektorií, což mohlo způsobit více detekcí v krajní části obrazu kamery, kde může docházet k většímu pokřivení obrazu a tudíž k většímu zkrešení detekovaných poloh.

Kapitola 6

Závěr

V první kapitole je rozebrán problém sledování člověka mobilním robotem. Jsou zde uvedeny nutné základní funkce, které musí algoritmus pro sledování obsahovat. Dále je zde uvedeno několik obecně využívaných metod pro sledování člověka. Zároveň je zde uvedena kapitola věnující se sociálním aspektům spolupráce člověka a robotu.

Druhá kapitola se věnuje možným metodám vizuálního sledování objektů. Je zde uvedena kapitola věnující se kalibraci kamery, kterou je nezbytné provést, aby došlo k potlačení deformací obrazu a tím ke zkreslení údajů o poloze. Jsou zde uvedeny metody pro nepřímé sledování polohy s pomocí speciálních značek připevněných na sledovaný objekt, včetně podrobnějšího popisu principu jedné z nich. Dále je zde ukázána metoda pro nalezení osob v obraze kamery a metoda pro následné určení polohy nalezených osob.

Ve třetí kapitole jsou rozebrány metody nutné pro rekonstrukci cesty, kterou má mobilní robot sledovat. Získané informace z kamery o relativní poloze značky je nutné převést do globálních souřadnic. K tomu je nutné znát polohu robotu v globálních souřadnicích, kterou lze získat metodou lokalizace, jakou je například odometrie robotu. Dále kapitola obsahuje popis metody filtrace naměřených dat, které umožňují potlačit šum a nepřesnosti, ale také vyhladit vytvořenou cestu. V případě reálného použití může dojít ke ztrátě sledovaného objektu ze záběru kamery. Z toho důvodu je nutné nahrazovat chybějící kusy cesty. K tomuto účelu zde byla použita podmnožina polynomů třetího řádu, tzv. Hermitovy křivky. Nakonec byla uvedena metoda pro sledování cesty „Follow the carrot“ a princip PID regulátoru.

Ve čtvrté kapitole byla popsána celá implementace algoritmu, včetně konkrétních použitých hodnot a metod. Celý algoritmus byl vytvořen na laboratorním robotu Turtlebot 2 a algoritmus zkompletován v systému ROS v jazyce c++. K vizuálnímu sledování byla využita kamera Basler se dvěma různými různými objektivy, pro které zde byly uvedeny kalibrační hodnoty. Dále zde byly uvedeny výsledky porovnání třech značek: Pi-tag, Apriltag a Whycon, díky kterým byla pro další pokusy vybrána značka Whycon. Byly zde popsány konkrétní parametry a metody použité pro filtrování dat, rekonstrukci cesty, či její sledování.

V poslední kapitole byly uvedeny výsledky několika experimentů. Cílem prvních dvou bylo porovnat chování pro dva různé objektivy, z nichž jeden byl širokoúhlý a druhý obyčejný. Třetí pokus měl za úkol zjistit jak bude sledování ovlivněno počtem detekovaných poloh značky.

Z prvních dvou experimentů je vidět, že metoda sledování vytvořené trajektorie fungovala obstojně ve všech případech. Průměrná odchylka odometrie robotu od vytvořené cesty se pohybovala do několika centimetrů. Ovšem z grafů odometrie a skutečné pozice je vidět, že s narůstající délkou jízdy robotu dochází také k nárůstu nepřesnosti odometrie. To má mimo jiné za následek také zkreslení vytvořené cesty, protože globální poloha značky je dána polohou robotu a její relativní polohou ke kameře. Z pozorování při experimentu lze vyvodit, že s obyčejným objektivem se robot choval předvídatelněji, ovšem častěji ztrácel značku ze záběru. S širokoúhlým objektivem byly odchylky robotu mírně nižší, ovšem častěji se projevovaly nesprávné detekce z důvodu méně kvalitního obrazu v krajních částech snímků, algoritmus tak působil méně stabilně. Z výsledků obou pokusů je tedy patrné, že největší chybu způsobovaly špatné detekce poloh značky.

Z výsledků třetího pokusu je vidět, že dvojnásobný počet detekovaných poloh neměl zásadní vliv na funkcionalitu algoritmu. Dá se usuzovat, že pokud má algoritmus přiměřený minimální počet detekovaných poloh, jejich další navýšení zásadním způsobem neovlivní funkci sledování. Minimální množství poloh pro spolehlivé sledování nelze určit přesně, ale je nutné vyvodit z podmínek, ve kterých má být algoritmus implementován. Především jsou rozhodující plánovaná rychlost robotu a velikost a členitost prostředí.

6.1 Možnosti rozšíření práce

Dle výsledků experimentů je pro zlepšení funkčnosti algoritmu zásadní zpřesnit detekci polohy značky. Prvním možností je zvětšit úhel záběru kamery oproti obyčejnému objektivu, ale zároveň je vhodné vyhnout se širokoúhlému objektivu, který se obtížně kalibruje, čímž se zároveň snižuje úhel záběru a obraz je méně kvalitní. Řešením může být více obyčejných kamer připevněných na robot, či využití jednotky Pan-tilt s připevněnou kamerou, která umožní její natočení libovolným směrem nezávisle na natočení robotu. Další možností je vyzkoušení jiných detektorů značek, které mohou mít přesnější detekci a být méně náchylné ke špatným detekcím.

Dalším problémem, který způsobuje nepřesnosti ve vytvořené cestě oproti skutečnému pohybu značky, je narůstající nepřesnost lokalizace robotu zajišťovaná odometrií. Pro odstranění narůstající nepřesnosti je možné použít sofistikovanější algoritmus, například vizuální odometrii s využitím laserového dálkoměru.

Z experimentů je vidět, že metoda vytvořená pro sledování cesty je i přes svou relativní jednoduchost schopna poměrně přesného sledování cesty. Pro zlepšení lze využít sofistikovanějšího algoritmu jakým je například MPC (Model predictive control), který dokáže na stanoveném horizontu hodnot nalézt optimální řešení pro sledování.

S ohledem na bezpečnost by při použití v reálných podmínkách měl robot umět detekovat neočekávané překážky, například osoba procházející mezi člověkem a robotem, a adekvátním způsobem na ně reagovat. K tomu je nutné robot opatřit vhodným senzorem, kterým může být například laserový dálkoměr. Ten lze současně využít i pro zpřesnění lokalizace.



Literatura

- [1] Camera suite [online]. https://april.eecs.umich.edu/wiki/Camera_suite. [cit. 20.5.2019].
- [2] Camera_calibration [online]. http://wiki.ros.org/camera_calibration. [cit. 20.5.2019].
- [3] Fiducial marker nodelet [online]. <https://hackaday.io/project/12373-bunnybot/log/41970-fiducial-marker-nodelet>. [cit. 20.5.2019].
- [4] Pid controller [online]. <http://engineering.ju.edu.jo/Laboratories/07-PID%20Controller.pdf>. [cit. 20.5.2019].
- [5] Robot operating system [online]. <http://wiki.ros.org/ROS/Introduction>. [cit. 20.5.2019].
- [6] Stavový model a Kalmanův filtr [online]. <https://www.fd.cvut.cz/personal/provipav/Stochastika/Materialy-test4/Stav.pdf>, 2013. [cit. 20.5.2019].
- [7] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 623–630. IEEE, 2010.
- [8] Jim Armstrong. Hermite curves [online]. <http://algorithmist.net/docs/hermite.pdf>, November 2005. [cit. 20.5.2019].
- [9] Filippo Bergamasco, Andrea Albarelli, and Andrea Torsello. Pi-tag: A fast image-space marker design based on projective invariants. *Machine Vision and Applications*, 24, 08 2013.

- [10] Woojin Chung, Hoyeon Kim, Yoonkyu Yoo, Chang-Bae Moon, and Jooyoung Park. The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder. *IEEE transactions on industrial electronics*, 59(8):3156–3166, 2012.
- [11] Quoc Khanh Dang and Young Soo Suh. Human-following robot using infrared camera. In *2011 11th International Conference on Control, Automation and Systems*, pages 1054–1058. IEEE, 2011.
- [12] OpenCV documentation. Camera calibration and 3d reconstruction [online]. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html. [cit. 20.5.2019].
- [13] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [14] David L. Finn. Ma 323 geometric modelling course notes: Day 08 cubic curves and cubic hermite interpolation [online]. <https://www.rose-hulman.edu/~finn/CCLI/Notes/day08.pdf>, December 2004. [cit. 20.5.2019].
- [15] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco Madrid-Cuevas, and Manuel Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47:2280–2292, 06 2014.
- [16] Rachel Gockley, Jodi Forlizzi, and Reid Simmons. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 17–24. ACM, 2007.
- [17] Vladimír Havlena. Estimation, filtering, and detection. Czech Technical University Lecture, EFD lecture notes, 2019.
- [18] Nobutaka Hirai and Hiroshi Mizoguchi. Visual tracking of human back and shoulder for person following robot. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 1, pages 527–532. IEEE, 2003.
- [19] Laurent Itti. Demo aruco [online]. <http://jevois.org/moddoc/DemoArUco/modinfo.html>. [cit. 20.5.2019].
- [20] Mariano Jaimez, Javier G Monroy, and Javier Gonzalez-Jimenez. Planar odometry from a radial laser scanner. a range flow-based approach. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4479–4485. IEEE, 2016.

- [21] Abhishek Jha and Manoj Kumar. Two wheels differential type odometry for mobile robots. In *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, pages 1–5. IEEE, 2014.
- [22] Tomáš Krajník, Matias Nitsche, Jan Faigl, Tom Duckett, Marta Mejail, and Libor Přeučil. External localization system for mobile robotics. 12 2013.
- [23] Yuan Li, Bo Wu, and Ram Nevatia. Human detection by searching in 3d space using camera and scene knowledge. In *2008 19th International Conference on Pattern Recognition*, pages 1–5. IEEE, 2008.
- [24] Peter Lightbody, Tomáš Krajník, and Marc Hanheide. A versatile high-performance visual fiducial marker detection system with scalable identity encoding. In *Proceedings of the Symposium on Applied Computing*, pages 276–282. ACM, 2017.
- [25] Martin Lundgren. Path tracking for a miniature robot. *Masters, Department of Computer Science, University of Umea*, page 9, 2003.
- [26] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011.
- [27] Robert Paz. The design of the pid controller. 01 2001.
- [28] Matthew Petroff. Pi-tag-detector [online]. <https://github.com/mpetroff/pi-tag-detector>. [cit. 20.5.2019].
- [29] Nils Pipenbrinck. Hermite curve interpolation [online]. <https://www.cubic.org/docs/hermite.htm>, March 1998. [cit. 20.5.2019].
- [30] Clearpath Robotics. Turtlebot 2 [online]. <https://www.clearpathrobotics.com/turtlebot-2-open-source-robot/>. [cit. 20.5.2019].
- [31] Francisco Romero Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76, 06 2018.
- [32] Elin Anna Topp and Henrik I Christensen. Tracking for following and passing persons. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2321–2327. IEEE, 2005.
- [33] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.
- [34] M. Wickert. Fir filters [online]. http://www.eas.uccs.edu/~mwickert/ece2610/lecture_notes/ece2610_chap5.pdf. [cit. 20.5.2019].

- [35] Wikipedia contributors. Pinhole camera model — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Pinhole_camera_model&oldid=871297383, 2018. [Online; accessed 15-April-2019].
- [36] Wikipedia contributors. Cubic hermite spline — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Cubic_Hermite_spline&oldid=891722897, 2019. [Online; accessed 13-April-2019].
- [37] Wikipedia contributors. Finite impulse response — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Finite_impulse_response&oldid=890030121, 2019. [Online; accessed 26-April-2019].
- [38] Ji Zhang and Sanjiv Singh. Laser–visual–inertial odometry and mapping with high robustness and low drift. *Journal of Field Robotics*, 35(8):1242–1264, 2018.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novotný** Jméno: **Martin** Osobní číslo: **434769**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Kybernetika a robotika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Autonomní sledování člověka mobilním robotem

Název diplomové práce anglicky:

Autonomous Human-following robot

Pokyny pro vypracování:

1. Seznamte se s metodami vizuálního sledování objektů a člověka
2. Seznamte se s metodami sledování trajektorie mobilním robotem
3. Navrhněte metody rekonstrukce trajektorie sledovaného člověka
4. Implementujte metodu sledování člověka mobilním robotem
5. Vyhodnoťte vlastnosti implementované metody

Seznam doporučené literatury:

1. Human-following robot using infrared camera, Quoc Khanh Dang ; Young Soo Suh, International Conference on Control, Automation and Systems, 2011
2. Visual tracking of human back and shoulder for person following robot, N. Hirai ; H. Mizoguchi, International Conference on Advanced Intelligent Mechatronics, 2003
3. Natural person-following behavior for social robots. Rachel Gockley, Jodi Forlizzi, and Reid Simmons, International conference on Human-robot interaction 2007

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Karel Košnar, Ph.D., inteligentní a mobilní robotika CIIRC

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **14.02.2019**

Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **20.09.2020**

Ing. Karel Košnar, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta