

VŠB – Technická univerzita Ostrava

Fakulta strojní

Katedra robotiky

Senzorický a navigační subsystém pro mobilní robot

Sensor and navigation subsystem for mobile robot

Student:

Bc. Petr Oščádal

Vedoucí diplomové práce:

Ing. Zdenko Bobovský, PhD.

Ostrava 2019

## Zadání diplomové práce

Student:

**Bc. Petr Oščádal**

Studijní program:

N2301 Strojní inženýrství

Studijní obor:

2301T013 Robotika

Téma:

Senzorický a navigační subsystém pro mobilní robot  
Sensor and Navigation Subsystem for Mobile Robot

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Analýza současněho stavu .
2. Specifikace požadavkového listu.
3. Lokomoční subsystém.
4. Senzorický subsystém.
5. Navigační subsystém.
6. Softwarová část operátor.
7. Softwarová část simulační model.
8. Práci též doložte v elektronické podobě ve formátu MS WORD.

Seznam doporučené odborné literatury:

1. Smrček, L. Kárník, Robotika - Servisné roboty. Navrhovanie – Konštrukcia – Riešenia. Edícia vedeckej a odbornej literatúry - SJF TUKE, Košice 2008.
2. Jens Christian Andersen. Mobile Robot Navigation. PhD thesis. September 2007, [online] <http://orbit.dtu.dk/files/4724729/oersted-dtu2868.pdf>
3. Stachniss Cyrill. Spatial Modeling and Robot Navigation. 2009. [online] <http://www2.informatik.uni-freiburg.de/~stachnis/pdf/stachniss-habil.pdf>
4. Ng James. An Analysis of Mobile Robot Navigation Algorithm in Unknown Environments. 2010. [online] <http://robotics.ee.uwa.edu.au/theses/2010-Navigation-Ng-PhD.pdf>
5. Kirby Rachel. Social Robot Navigation. 2010. [online] [https://ri.cmu.edu/pub\\_files/2010/5/rk\\_thesis.pdf](https://ri.cmu.edu/pub_files/2010/5/rk_thesis.pdf)
6. V-REP User Manual. Coppelia Robotics V-REP: Create. Compose. Simulate. Any Robot. [online]. Dostupné z: <http://www.coppeliarobotics.com/helpFiles/>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Zdenko Bobovský, PhD.**

Datum zadání: 21.12.2018

Datum odevzdání: 20.05.2019

---

prof. Dr. Ing. Petr Novák  
vedoucí katedry



---

prof. Ing. Ivo Hlavatý, Ph.D.  
děkan fakulty

## **Místopřísežné prohlášení studenta**

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě 20.5.2019

.....

podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§ 35 odst. 3).
- souhlasím s tím, že diplomová práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užití své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě 20.5.2019

.....

podpis

Jméno a příjmení autora práce: Bc. Petr Oščádal

Adresa trvalého pobytu autora práce: Pěňčín č.p. 8, okr. Prostějov

## **ANOTACE DIPLOMOVÉ PRÁCE**

Diplomová práce se zabývá analýzou robotu, prostředí a simulačním programem. Z této analýzy je stanoven požadavkový list. Následně jsou vytvořeny simulační modely robotu a prostředí. Je navrhnout senzorický subsystém, který umožňuje detekci překážek a lokalizaci robotu. K systému je v závěru vytvořena řídicí aplikace pro operátora.

## **ANNOTATION OF MASTER THESIS**

The thesis deals with the analysis of robot, surroundings and simulation program. The requirement sheet is based on this analysis. Then were created simulation models of robot and surroundings. A sensory subsystem is designed to detect obstacles and locate the robot. At the end is created a control application for the operator for the system.

## **Poděkování**

Především bych rád poděkoval panu Ing. Zdenku Bobovskému, PhD. za poskytnuté konzultace při přípravě diplomové práce a předávání velice užitečných zkušeností v oblasti programování. Dále bych chtěl poděkovat katedře robotiky, za poskytnutou výpočetní techniku, která umožnila kvalitnější zpracování práce. Také bych rád poděkoval Bc. Markétě Crhové za poskytnuté rady a připomínky ohledně formální stránky práce.

Diplomová práce prezentuje znalosti nabyté v rámci řešení projektů: „Digitální dvojčata robotických systémů a jejich verifikace“, reg. č.: „SP2019/69“, „Adaptabilita robotických systémů“, reg. č.: „SP2018/86“ a „Výzkum a vývoj robotických systémů“, reg. č.: „SP2017/143“ financovaných ze státního rozpočtu České republiky, na kterých jsem se v průběhu magisterského studia podílel.

# Obsah

<b>Seznam obrázků</b> .....	<b>11</b>
<b>Seznam tabulek</b> .....	<b>14</b>
<b>Slovník termínů</b> .....	<b>15</b>
<b>Úvod</b> .....	<b>16</b>
<b>1 Analýza</b> .....	<b>17</b>
1.1 Robot Odin .....	17
1.2 Prostředí .....	19
1.3 V-Rep .....	21
1.3.1 Fyzikální výpočtové modely .....	21
1.3.2 Senzory ve V-Repu .....	22
1.3.3 Modul pro plánování trajektorie .....	24
1.3.4 OMPL .....	24
1.4 Lokalizace .....	25
1.4.1 Relativní lokalizace .....	25
1.4.2 Absolutní lokalizace .....	25
1.4.3 Lokalizace pomocí referenčních bodů .....	26
1.5 Emgu CV .....	26
<b>2 Specifikace požadavkového listu</b> .....	<b>27</b>
2.1 Požadavky .....	27
<b>3 Simulační modely</b> .....	<b>28</b>
3.1 Lokomoční subsystém .....	28
3.1.1 Popis dynamického modelu .....	29
3.2 Simulační prostředí .....	30
3.2.1 Popis dynamického modelu .....	30
<b>4 Senzorický subsystém</b> .....	<b>31</b>
4.1 Kolizní subsystém .....	31
4.1.1 Sonary .....	31
4.1.2 Lidary .....	32
4.2 Lokalizační subsystém .....	32
4.2.1 Varianta A .....	33
4.2.2 Varianta B .....	34



4.3	Zhodnocení variant.....	35
4.4	Hardwarové komponenty .....	36
4.4.1	Sonar .....	37
4.4.2	Lidar.....	38
4.4.3	Kamera.....	39
4.5	Popis senzorických modelů .....	40
4.5.1	Simulační sonar.....	40
4.5.2	Simulační lidar .....	41
4.5.3	Simulační kamera.....	42
<b>5</b>	<b>Řídicí systém .....</b>	<b>44</b>
5.1	Aplikace 1.....	45
5.2	Aplikace 2.....	45
5.2.1	Komunikace .....	45
5.2.2	Logika .....	46
5.2.3	Podvozek.....	47
5.2.4	PathPlanning .....	47
<b>6</b>	<b>Navigační subsystém .....</b>	<b>48</b>
6.1	Lokalizace.....	48
6.2	Plánování trajektorie.....	50
6.3	Globální trajektorie.....	51
6.3.1	Nastavení algoritmu ve V-Repu.....	51
6.4	Lokální trajektorie .....	52
6.4.1	Nastavení algoritmu ve V-Repu.....	53
6.5	Simulace výpočtu trajektorie .....	54
<b>7</b>	<b>Softwarová část operátora.....</b>	<b>55</b>
7.1	Struktura aplikace .....	55
7.2	GUI rozhraní.....	55
7.3	Komunikace.....	57
<b>8</b>	<b>Softwarová část simulačního modelu .....</b>	<b>58</b>
8.1	Komunikace s nadřazeným systémem.....	58
<b>9</b>	<b>Ukázky simulace.....</b>	<b>59</b>
<b>10</b>	<b>Testování systému .....</b>	<b>64</b>
	<b>Závěr .....</b>	<b>65</b>

<b>Seznam použité literatury .....</b>	<b>66</b>
<b>Přílohy.....</b>	<b>68</b>

---

## Seznam obrázků

Obr. 1	Robot Odín s nástavbou Schunk .....	17
Obr. 2	Řídící aplikace robotu .....	18
Obr. 3	3D model robotu .....	18
Obr. 4	Plány prostorů katedry robotiky .....	19
Obr. 5	Detail chodby .....	20
Obr. 6	Prostředí V-Rep .....	21
Obr. 7	Typy detekovacích výsečí [9] .....	23
Obr. 8	Vision senzory [11] .....	23
Obr. 9	Ukázka detekce Aruco tagu .....	26
Obr. 10	Robot Odín ve V-Repu .....	28
Obr. 11	Dynamický model robotu .....	29
Obr. 12	Simulace Mecanum kola .....	29
Obr. 13	Vizuální model prostoru .....	30
Obr. 14	Dynamický model prostoru .....	30
Obr. 15	Rozmístění sonarů .....	31
Obr. 16	Rozmístění lidarů .....	32
Obr. 17	Aruco tagy .....	32
Obr. 18	Rozmístění Aruco kódů varianta A - schéma .....	33
Obr. 19	Rozmístění Aruco kódů varianta A - simulace .....	33
Obr. 20	Rozmístění Aruco kódů varianta B - schéma .....	34
Obr. 21	Rozmístění Aruco kódů varianta B - simulace .....	34
Obr. 22	Umístění kamer .....	35
Obr. 23	Celkový senzorický subsystém .....	36
Obr. 24	Sonar SRF08 [4] .....	37
Obr. 25	Hokuyo URG-04LX-UG01 [3] .....	38
Obr. 26	Kamera Logitech [7] .....	39
Obr. 27	Simulace sonarů .....	40
Obr. 28	Parametry sonaru .....	40
Obr. 29	Simulace lidarů .....	41

---

Obr. 30	Parametry lidarů.....	41
Obr. 31	Simulace kamer.....	42
Obr. 32	Parametry simulační kamery .....	43
Obr. 33	Struktura řídicího systému .....	44
Obr. 34	Schéma sítě .....	44
Obr. 35	Blokové schéma aplikace 2.....	45
Obr. 36	Blokové schéma odesílání dat.....	46
Obr. 37	Schéma objektu Logika .....	47
Obr. 38	Blokové schéma navigačního subsystému.....	48
Obr. 39	Souřadnicové systémy výpočtu pozice robotu.....	49
Obr. 40	Schéma výpočtových parametrů .....	49
Obr. 41	Blokové schéma lokalizačního subsystému.....	50
Obr. 42	Schéma plánování trajektorie .....	50
Obr. 43	Nastavení globální trajektorie.....	51
Obr. 44	Nastavení parametrů globální trajektorie .....	52
Obr. 45	Nastavení lokální trajektorie.....	53
Obr. 46	Nastavení parametrů lokální trajektorie .....	54
Obr. 47	Matematický model .....	54
Obr. 48	Simulace systému .....	54
Obr. 49	Rozhraní operátora .....	55
Obr. 50	Nastavování režimů .....	56
Obr. 51	Nastavování kamer .....	56
Obr. 52	Nastavování pozic.....	56
Obr. 53	Graf rychlosti pohonů .....	56
Obr. 54	Přeposílání objektů .....	57
Obr. 55	Streamování dat z V-Repu [10].....	58
Obr. 56	Simulace bez překážky .....	59
Obr. 57	Výpočetní model - bez překážky na krátké vzdálenosti .....	59
Obr. 58	Výpočetní model - bez překážky na velké vzdálenosti .....	60
Obr. 59	Simulace s překážkou na krátké vzdálenosti .....	61
Obr. 60	Výpočetní model – s překážkou na krátké vzdálenosti .....	61

---

---

Obr. 61	Simulace s překážkami na velké vzdálenosti .....	62
Obr. 62	Výpočetní model – s překážkami na velké vzdálenosti.....	62
Obr. 63	Simulace s překážkami na velké vzdálenosti .....	63
Obr. 64	Výpočetní model – s překážkami na velké vzdálenosti.....	63
Obr. 65	Testování lokalizace .....	64

**Seznam tabulek**

Tab. 1	Parametry robotu Odin.....	19
Tab. 2	Parametry prostoru.....	20
Tab. 3	Parametry požadavků.....	27
Tab. 4	Počty využitých senzorů .....	36
Tab. 5	Parametry reálného senzoru SRF08.....	37
Tab. 6	Parametry reálného senzoru Hokuyo URG-04LX-UG01.....	38
Tab. 7	Parametry kamery Logitech Pro Stream Webcam C922 PRO .....	39

## **Slovník termínů**

**OMPL** jsou knihovny pro plánování trajektorie.

**OpenSorce** je projekt, který je vyvíjen komunitou a je volně dostupný na internetu.

**Serializace** je funkce k vytvoření dat typu string, ve kterém je popsán určený objekt.

**Deserializace** je reverzní funkcí serializace. Z dat typu string vytvoří objekt.

**Wrapper** je zaobalení softwarové knihovny. Umožňuje volat funkce z externí knihovny.

## Úvod

Servisní robotika je specifická tím, že z velké části vyvíjí speciální roboty, které jsou jedinečné a určené pro konkrétní servisní úlohy. Proto jsou tyto roboty unikátní a jsou navrhovány pro konkrétní požadavky. Tyto požadavky vyplývají z vykonávané činnosti robotu, prostředí ve kterém se robot pohybuje, požadovaného stupně autonomie a dalších. Jelikož je návrh robotického systému velice složitý podílí se na něm tým lidí. Každý člen týmu pak má za úkol návrh dílčích subsystémů robotu. Proto se tato práce zabývá pouze sensorickým a navigačním subsystémem. Tento systém bude aplikován na již reálný robot.

Práce se zabývá návrhem sensorického a navigačního subsystému mobilního robotu. Cílem práce je navrhnout systém, který umožní mobilnímu robotu autonomní jízdu v předem definovaném prostředí s dynamicky se měnícími překážkami.

Navrhovaný subsystém vychází z analýzy robotu, pro který je systém určený. Dále je provedena analýza prostředí, ve kterém se má robot pohybovat. Následně je vytvořen simulační model robotu s reálně nastavenými parametry vyplývajícími z detailního 3D modelu. Rovněž je vytvořen simulační model prostředí, které vyplývá z návrhu možných prostor katedry robotiky.

Práce se dále zabývá rozšířením sensorického subsystému, který je rozdělen na detekci překážek a lokalizaci robotu. Data ze sensorů jsou dále zpracovávána pro výpočet aktuální polohy a vektoru pohybu do operátorem zadané pozice. Operátor nastavuje parametry robotu pomocí serveru, který komunikuje s robotem. Systém umožňuje řídit robot manuálně nebo automaticky.



# 1 Analýza

Kapitola se zabývá analýzou robotu, pro který bude navržen senzorický a navigační subsystém. Rovněž bude v kapitole provedena analýza prostředí, ve kterém se robot bude pohybovat. Dále budou popsány vlastnosti a možnosti simulačního nástroje, který bude využit k následným simulacím. V závěru kapitoly jsou popsány druhy lokalizací, které umožňují určit polohu robotu.

## 1.1 Robot Odin

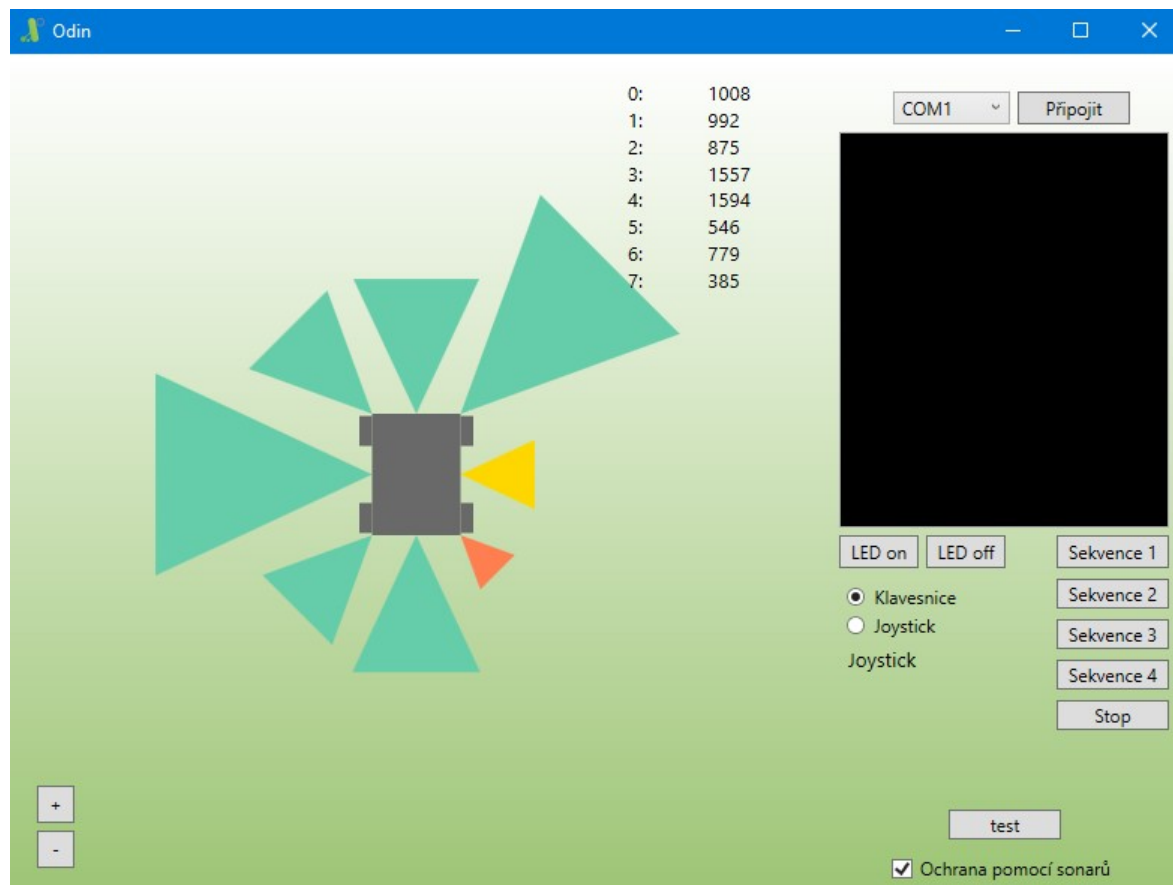
Robot je transportní platforma na podvozku se všesměrovými koly pro logistická centra (viz Obr. 1). Na systému jsou aplikována kola typu Mecanum, které mají po obvodu válečky pootočené o 45°. Každé kolo má vlastní pohon Maxon o výkonu 90 W, který je řízen pomocí řídicí jednotky Epos 50/5. Kola jsou uložena v ložiskových domcích na rámu, tvořených z Item profilů. Pomocí 8 sonarů se může vyhýbat překážkám ze všech směrů. Řídicí jednotka je zde Netduino 2 +, která sbírá data ze senzorů a posílá je pomocí routeru přes WI-FI do řídicího počítače. Zásah do řízení od operátora následně netduino přepošle do řídicích jednotek motorů pomocí sběrnice CAN.



Obr. 1 Robot Odin s nástavbou Schunk

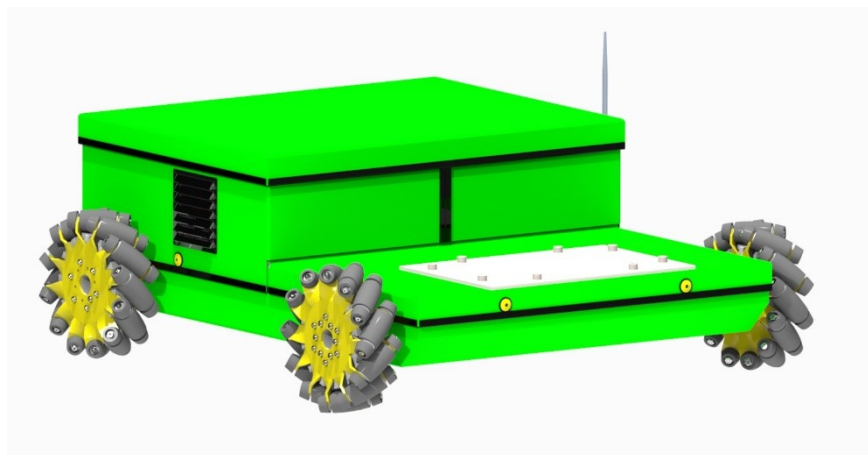
Robot nedisponuje žádnými autonomními funkcemi. Operátor má k dispozici pouze informace o vzdálenosti překážek od senzorů, které jsou vykreslovány operátorovy do řídicí aplikace (viz Obr. 2).

Aplikace (viz Obr. 2) umožňuje ovládat robota pomocí klávesnice nebo joysticku. Dále je v aplikaci možné využít předdefinované sekvence, které demonstrují pohyb po kružnici nebo čtverci.



Obr. 2 Řídící aplikace robota

K tomuto robotu je vytvořen důkladný 3D model, který byl vytvořen v modeláři Creo (viz Obr. 3). Tento robot byl navržen v rámci diplomové práce [3], díky čemuž je k dispozici jeho detailní model, podle kterého byl robot sestaven.



Obr. 3 3D model robota

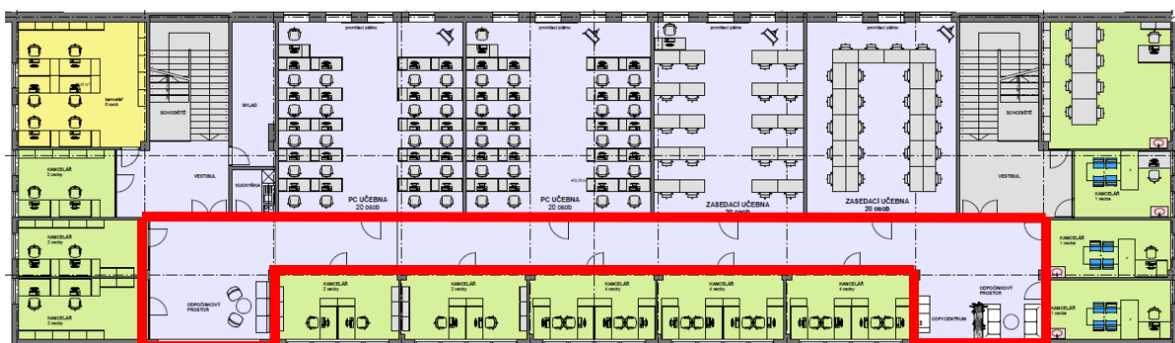
Reálný robot disponuje základními parametry, které jsou shrnuty v Tab. 1. Tyto parametry budou následně využity při vytváření simulačního modelu.

Tab. 1 Parametry robotu Odin

Hmotnost	45 [Kg]
Maximální rychlost	2.5 [m/s]
Rozměry	315x750x800 [mm]
Světlá výška	45 [mm]
Výkon motorů	4 x 90 [W]
Max. moment pohonů	7.5 Nm
Napájecí napětí	24 [V]
Kapacita akumulátoru	20 [Ah]
Nosnost	226 Kg
Odpružení	-
Kola	4 x Mecanum 10"

## 1.2 Prostředí

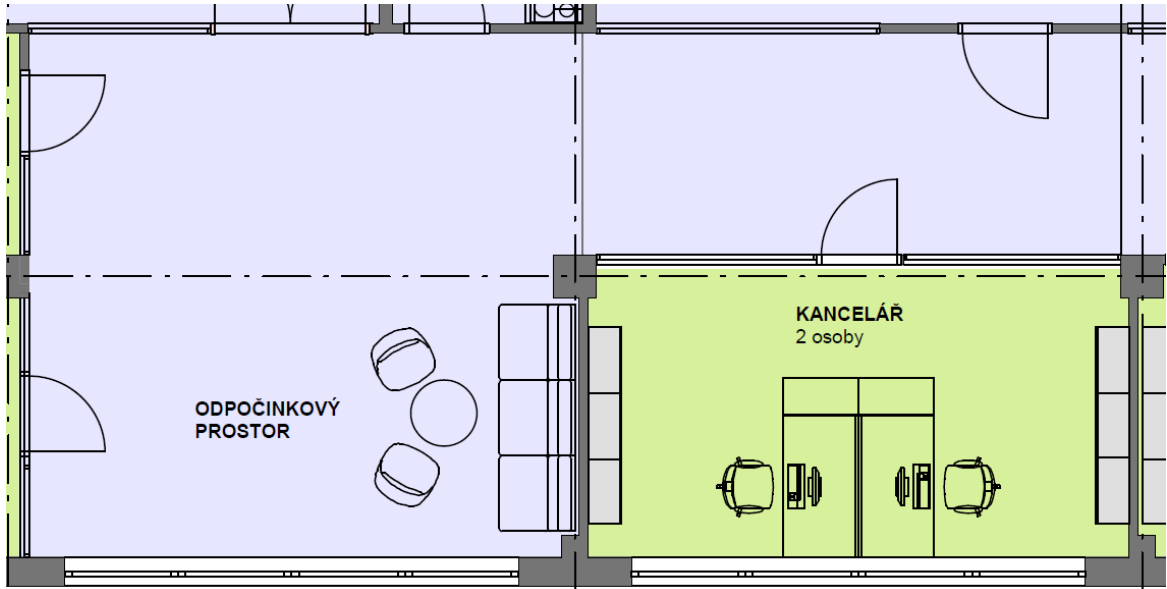
Robot je určen do nově navrhovaných prostor katedry robotiky. Prostory se budou skládat z učeben, kanceláří a chodby. I přes to, že je robot navržen tak, aby mohl projet dveřmi, bude jeho prostor k pohybu prozatím omezen pouze na prostor chodby (viz Obr. 4).



Obr. 4 Plány prostorů katedry robotiky

Z výkresů, které slouží jako plány budoucích prostor katedry není zřejmé, z jakého materiálu bude povrch země vytvořen. Jelikož má robot všesměrová kola, bude pracováno s předpokladem, že povrch bude hladký, tedy příznivý pro pohyb robotu. Požadavek hladkého povrchu bude navržen do dalšího návrhu prostoru.

V nových prostorách se bude nacházet několik druhů materiálu stěn. Základní materiály budou tvořit čiré sklo, mléčné sklo, dřevo a obložení (viz Obr. 5). Většina povrchů stěn bude statická, kromě dveří, které se budou otevírat. Na chodbě se mohou nacházet různé překážky (krabice, vozíky, apod.) jako jsou například krabice, vozíky apod. Převážně v době mezi vyučujícími hodinami se v těchto prostorách bude pohybovat mnoho osob.



Obr. 5 Detail chodby

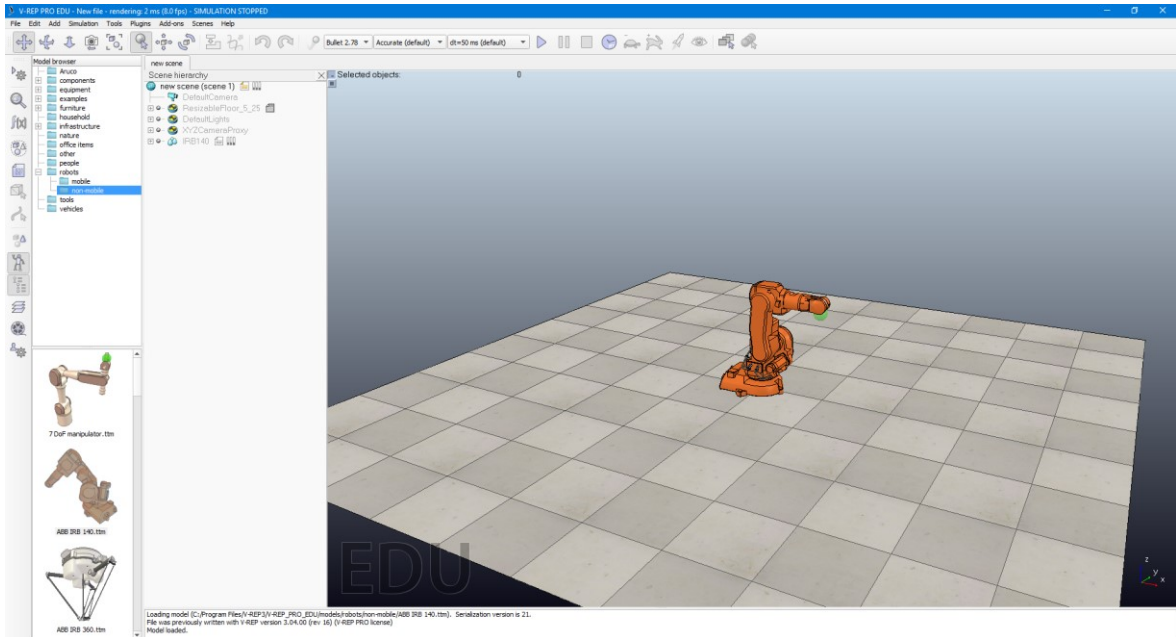
Z výkresu byly vyčteny základní parametry prostoru, který byl definován pro pohyb robotu. Parametry, které nejsou zřetelné z výkresu, jsou nadefinovány jako požadavek. Tyto parametry jsou přehledně zaznamenány v Tab. 2.

Tab. 2 Parametry prostoru

Délka	43 [m]
Šířka	7 [m]
Nejužší prostor	1.2 [m]
Překážky	Dynamické/Statické
Povrch stěn	Sklo/mléčné sklo/dřevo
Povrch podlahy	Hladký

## 1.3 V-Rep

V-Rep je prostředí (viz Obr. 6) ve kterém lze simulovat robotická zařízení. Je založen na architektuře distribuovaného řízení: každý objekt (model) může být individuálně řízený prostřednictvím vloženého skriptu, pluginu, uzlu ROS nebo BlueZero, vzdáleného API klienta nebo vlastního řízení. Tímto je V-Rep velmi univerzální a ideální simulační prostředí [21].



Obr. 6 Prostředí V-Rep

### 1.3.1 Fyzikální výpočtové modely

Jelikož je simulování fyzikálních jevů velice náročné, disponuje V-Rep čtyřmi výpočtovými modely, které se liší především v přesnosti, rychlosti výpočtů nebo podporou různých vlastností. Tyto modely lze libovolně přepínat před spuštěním simulace.

**Typy výpočtových modulů:**

- Bullet knihovna [1]

Tato knihovna disponuje výpočtovým modelem pro 3D kolizi, dynamikou tuhého tělesa a dynamikou měkkého tělesa (tuto funkci zatím V-Rep nepodporuje). Tato knihovna se především využívá ve hrách a při vizuálních efektech ve filmech.

- Vortex dynamika [20]

Knihovna vytváří simulace fyziky s vysokou věrností. Vortex nabízí výpočty na skutečných parametrech pro velké množství fyzikálních vlastností. Díky tomu jsou vytvořeny realistické a přesné simulace. Tato knihovna se využívá ve vysoce přesných a výkonných průmyslových výzkumných aplikacích.

- ODE (Open Dynamic Engine) [11]

Výpočetní model ODE má dvě hlavní součásti, a to dynamiku tuhého tělesa a detekci kolize. Využívá se především v herním průmyslu.

- Newton dynamika [10]

Umožňuje v simulaci interakci dynamických modelů. Věrně vytváří simulace padajících objektů, kolize nebo realistický pohyb vozidla v nerovném terénu.

### 1.3.2 Senzory ve V-Repu

V-Rep umožňuje simulovat základní senzory. Tyto senzory se dělí do dvou základních skupin, kterými jsou senzory měřící vzdálenost a vision senzory.

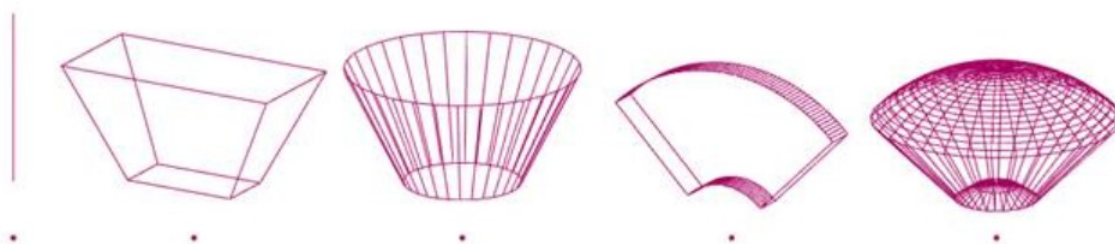
#### **Senzory měřící vzdálenost ve V-Repu:**

- Paprskový

Senzor velice přesně napodobuje reálný IR senzor, který bezkontaktně detekuje překážky pomocí světelného paprsku, a zvládne vypočítat vzdálenost od překážky.

- Pyramidový, válcový, diskový, kuželový

Tyto senzory dokážou detekovat nejbližší detekovatelný bod ve scéně v jejich detekovatelné výšce. Senzory se tedy liší pouze v této oblasti snímání (viz Obr. 7).

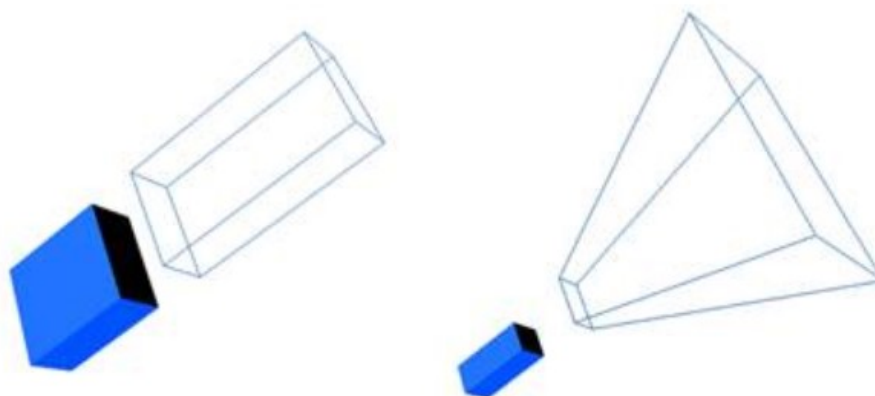


Obr. 7 Typy detekovacích výsečí [14]

Pro všechny tyto moduly senzorů lze nastavit fyzikální způsob snímání: ultrazvukový, infračervený, laser, indukční nebo kapacitní. Díky tomu lze věrně napodobovat reálné senzory.

### Vision senzory:

Tyto senzory se dělí na ortografické a perspektivní senzory (viz Obr. 8).



Obr. 8 Vision senzory [19]

- Ortografický senzor  
Zorné pole tohoto senzoru je pravoúhlé a je vhodné pro simulování infračerveného snímače blízké vzdálenosti nebo laserové detektory.
- Perspektivní senzor  
Zorné pole tohoto senzoru je lichoběžníkové. Tento senzor je vhodný pro simulaci kamer.

Vision senzory detekují jen ty předměty v simulaci, které mají nastavené rendrovatelné vlastnosti.

Všechny základní senzory, kterými disponuje V-Rep se dají rozšířit o script, který umožní vytvořit složitější senzor jako je například lidar. Lidar funguje na principu otáčení IR senzoru a po určitém kroku pootočení změří danou vzdálenost od překážky. Pomocí skriptu tedy lze naprogramovat senzor typu ray aby se otáčel a měřil vzdálenosti v určitých stupních, a tím tak vytvořit simulační model lidarů.

### 1.3.3 Modul pro plánování trajektorie

V-Rep disponuje i modulem „Path and motion planning“ pomocí kterého lze naplánovat trajektorii pohybu [13]. Tento modul využívá knihovnu „Open Motion Planning Library“ (OMPL), která se skládá z mnoha nejmodernějších plánovacích algoritmů.

Pro vygenerování cesty je zapotřebí udělat několik kroků:

- Nadefinovat Start a Cíl objektu
- Vytvořit úlohu plánování cesty
- Zvolit algoritmus výpočtu
- Vytvořit požadovaný prostor
- Nadefinovat kolizní stavy, ke kterým nesmí dojít
- Určit stav Startu a Cíle
- Vypočítat jednu nebo více cest
- Ukončit úlohu plánování

Jelikož algoritmy vypočítají jednu z mnoha možných cest a vyhodnocená cesta není jakkoliv optimalizována ani není zaručená její efektivnost, je doporučeno vypočítat několik cest pro daný Start a Cíl a zvolit tu cestu, která je nejvhodnější.

### 1.3.4 OMPL

Knihovna [18] je navržena tak, aby mohla být snadno integrována do systémů, který poskytuje další potřebné komponenty. Tato knihovna disponuje 25 algoritmy, které se liší v rychlosti výpočtu, způsobu využití, anebo vytváření trajektorie pohybu. Nejvyužívanější algoritmy této knihovny jsou:



- LB-KPIECE [8]

Tento algoritmus je založen na větvení stromů, který využívá diskretizaci pro vedení spojitého prostoru. Tato implementace je zjednodušená a využívá pouze jednu úroveň mřížky.

- RRTs [17]

Tato metoda je asymptoticky optimální přírůstkový algoritmus plánování trajektorie. Algoritmus zajišťuje, že bude konvergovat k optimální variantě na vyhraněném prostoru, který je určen k výpočtu trajektorie.

- RRTConnect [16]

Tento princip využívá k růstu trajektorie dva algoritmy RRTs. Trajektorie prvního roste od začátku, trajektorie druhého roste od konce a tyto dvě cesty se pokouší propojit.

## 1.4 Lokalizace

Pro určení polohy se metody řešící tento problém dají rozdělit do 3 základních skupin. Relativní, absolutní nebo pomocí referenčních bodů. Každá metoda má své výhody a nevýhody, proto se v praxi používá kombinace metod pro zpřesnění polohy. Kombinace metod dále zamezuje chybě v případě selhání jedné z metod [7].

### 1.4.1 Relativní lokalizace

Pomocí dat ze senzorů algoritmy lokalizace odhadují relativní změnu polohy robota vůči jeho předchozí poloze. Zde se vyhodnocuje posunutí a rotace v rovině. Globální poloha se tedy počítá skládáním jednotlivých dílčích změn polohy. Jelikož v každém kroku měření změny pozice dochází k chybě, tak se neustále zvyšuje celková chyba polohy a nelze tedy přesně měřit absolutní polohu robota v prostředí. V praxi se tento způsob lokalizace vyskytuje téměř ve všech lokalizačních systémech, avšak ne samostatně ale v kombinaci s absolutní lokalizací [7].

### 1.4.2 Absolutní lokalizace

Aby nedocházelo k akumulaci chyby polohy z měření, jako je tomu u relativní lokalizace, využívá se v praxi absolutní lokalizace. Ta pomocí jednoho měření vypočítá pravděpodobnost polohy, ve které se právě nachází, aniž by ji ovlivnila chyba z předchozího měření. Pro tuto metodu je zapotřebí znát mapu, nebo aspoň část mapy

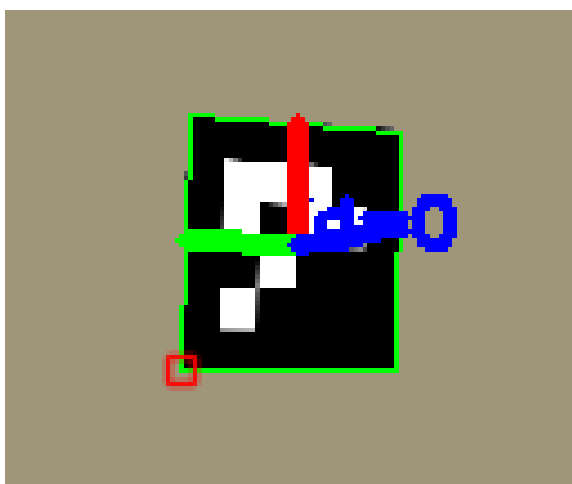
prostředí, ve kterém se robot pohybuje. V opačném případě nastává problém, při kterém robot nedokáže vyhodnotit aktuální pozici. Proto je vhodné kombinovat více metod současně [7].

### 1.4.3 Lokalizace pomocí referenčních bodů

Aby mohl robot určovat svoji aktuální pozici pomocí referenčních bodů, je nutné mít tyto body pevně spjaty s absolutní polohou v prostoru. Body mohou být jakékoliv rysy v prostředí, které může sensorický subsystém rozpoznat. Dále podle naměřených hodnot robot může dopočítat svoji aktuální polohu vůči jednomu nebo více bodů, a tím tak určit absolutní polohu v prostoru. Referenční body mohou být aktivní nebo pasivní. Aktivní body vysílají nebo přijímají data od robotu a jsou následně zpracována. Výsledkem je pak absolutní pozice robotu. Pasivní body žádné data nevysílají ani nepřijímají. Schopnost detekce pasivního referenčního bodu pak závisí na sensorickém subsystému robotu, například strojovém vidění [7].

## 1.5 Emgu CV

Emgu CV [2] je „wrapper“ OpenCV [12] pro zpracovávání obrazu. Knihovna OpenCV jsou postaveny na OpenSource licenci. EmguCV knihovna umožňuje volat funkce z OpenCV z kompatibilních jazyků .NET, jako jsou C#, VB, IronPhyton atd. Knihovny umožňují rozpoznávání základních tvarů, tagů (viz Obr. 9), segmentaci a další.



Obr. 9 Ukázka detekce Aruco tagu

Při rozpoznání tagu je detekováno ID tagu a orientace tagu. Tyto informace jsou přidány do obrazu a následně vykresleny. Orientace se následně vykresluje pomocí osového kříže.

## 2 Specifikace požadavkového listu

Na základě analýzy byly vytvořeny požadavky na systém, podle kterých bude následně navržen tak, aby byly všechny parametry splněny.

### 2.1 Požadavky

Základní parametry požadavků na systém jsou zaznamenány v Tab. 3.

Tab. 3 Parametry požadavků

Robot	Odin
Místo aplikace	Katedra robotiky
Rychlost pohybu	0.5 [m/s]
Senzory pro detekci stěn	Sonary
Senzory pro detekci překážek	Lidary
Senzory pro lokalizaci	Kamery
Komunikace	Wifi
Řízení	Automatické/ Manuální
Funkce	Plánování trajektorie

Další požadavky:

- celý systém bude simulován v prostředí V-Rep
- struktura bude navržena tak, aby byla snadno přenositelná na reálný systém
- systém musí být navržen tak, aby nedošlo ke kolizi
- pro systém budou aplikovány existující knihovny algoritmů

Popis řízení:

- Automatické řízení

Robot autonomně vygeneruje trajektorii pohybu podle zadaného cíle a bude tuto trajektorii následovat. Během jízdy bude kontrolovat vzdálenosti objektů a vytvoří dostatečný prostor, aby nedošlo ke kolizi robotu s neznámým objektem.

- Manuální řízení

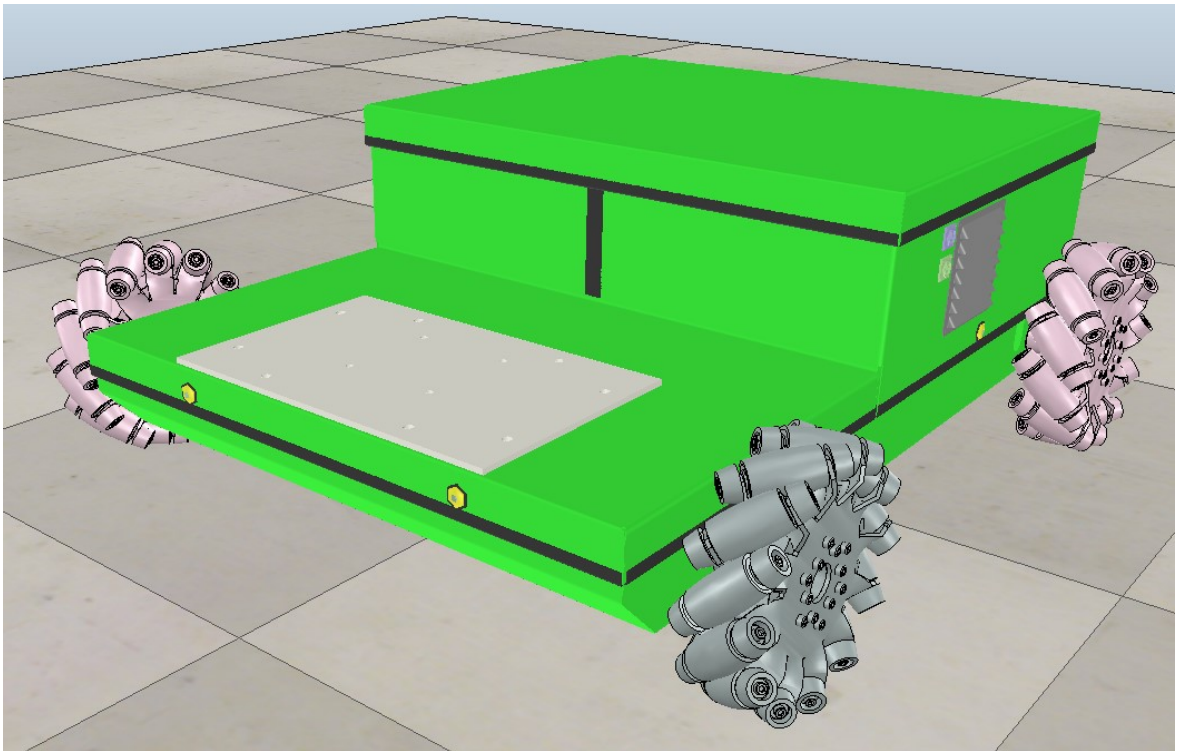
Robot je řízen operátorem, robot nezasahuje do řízení, ale pouze upozorňuje operátora na blížící se objekty a případné kolize.

### 3 Simulační modely

Kapitola obsahuje popis tvorby lokomočního subsystému v simulaci. Dále je v kapitole popsáno vytvoření dynamického modelu z jednoduchých tvarů. Druhá část kapitoly obsahuje popis tvorby simulačního prostředí (dle návrhu) a jeho dynamického modelu.

#### 3.1 Lokomoční subsystém

Simulační model byl vytvořen v prostředí V-Rep (viz Obr. 10). Zde byl importován model z modeláře Creo 4.0 pomocí souboru „stl“. Tento model je však velice náročný na následné výpočty a simulaci, a proto tento model slouží pouze jako pohledová část robotu.

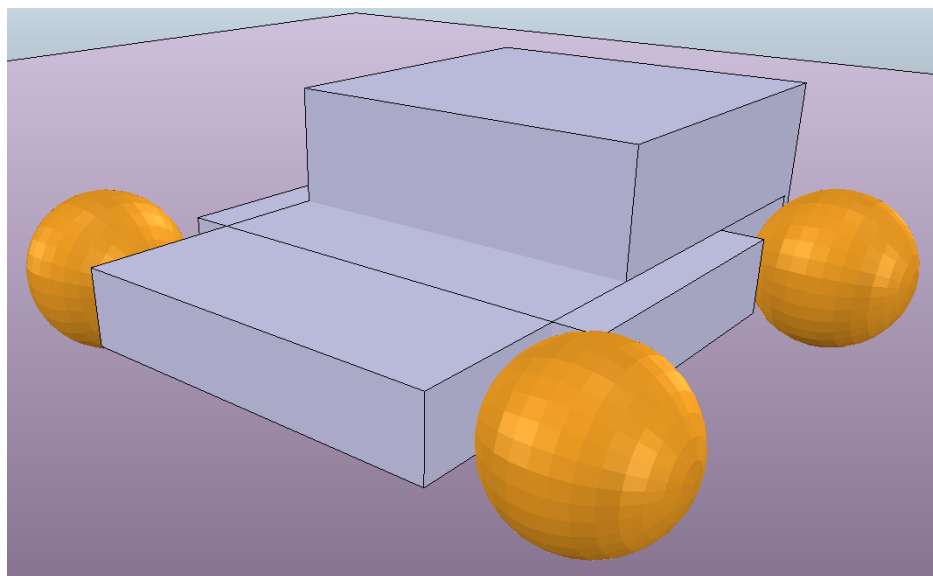


Obr. 10 Robot Odin ve V-Repu

Model byl rozdělen na části, které se mohou vůči sobě pohybovat. Základem je tělo robotu a kola, která jsou oddělena. Ve zjednodušeném dynamickém modelu se provádějí výpočty, které jsou následně reprezentovány pomocí vizuálního modelu. Z důvodu rozdělení vizuálního modelu na části vůči sobě se pohybující, umožňuje simulace reálnou vizualizaci systému.

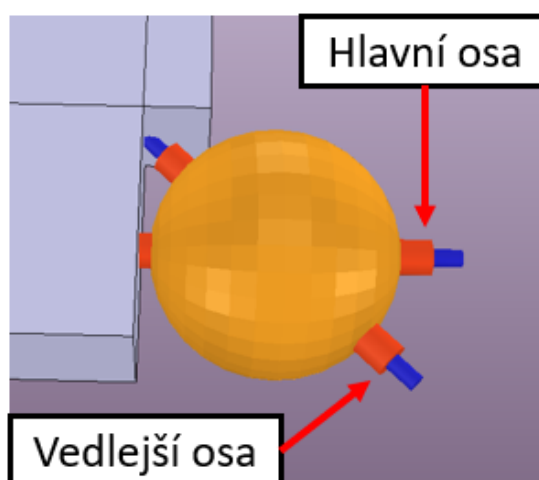
### 3.1.1 Popis dynamického modelu

Výpočetní model robotu byl vytvořen pomocí jednoduchých objektů (viz Obr. 11). Tyto objekty celkový tvar zjednodušují, ale zachovávají důležité parametry robotu, jako je například maximální rozměry nebo momenty setrvačnosti. Pro zjednodušený model byly nastaveny parametry vyplývající z programu Creo, ve kterém byl vytvořen detailní model robotu. Na dynamický model byly následně navazbeny vizuální modely.



Obr. 11 Dynamický model robotu

Všesměrové kolo je simulováno pomocí objektu koule, která má 2 stupně volnosti (viz Obr. 12). Tyto stupně volnosti jsou definovány pomocí hlavní osy, která odpovídá reálné ose kola a pomocné osy, která je pootočená o úhel, který odpovídá úhlu natočení válečků na reálném kole.



Obr. 12 Simulace Mecanum kola

## 3.2 Simulační prostředí

Robot je určen do nově budovaného vnitřního prostoru katedry Robotiky. Podle předběžných návrhů byl vytvořen zjednodušený vizuální model (viz Obr. 13). Aby byla zrychlena simulace slouží tento model pouze pro renderování kamer.

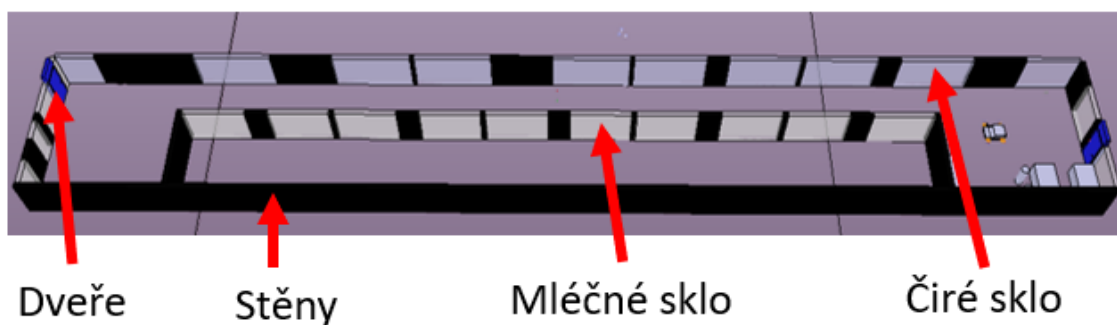


Obr. 13 Vizuální model prostoru

### 3.2.1 Popis dynamického modelu

Chodba, ve které se robot bude pohybovat byla vytvořena jako dynamický model (viz Obr. 14). Tento model je rozdělen podle druhů stěn na číré sklo, mléčné sklo a stěny s dveřmi. Každá z těchto sekcí má nastavené parametry detekce senzory.

Z důvodu vyskytování velkého počtu osob v prostorách chodby, byly vytvořeny dynamické dveře sloužící k průchodu osob. Tímto jsou zajištěny efekty, které mohou rušit systém.



Obr. 14 Dynamický model prostoru

## 4 Senzorický subsystém

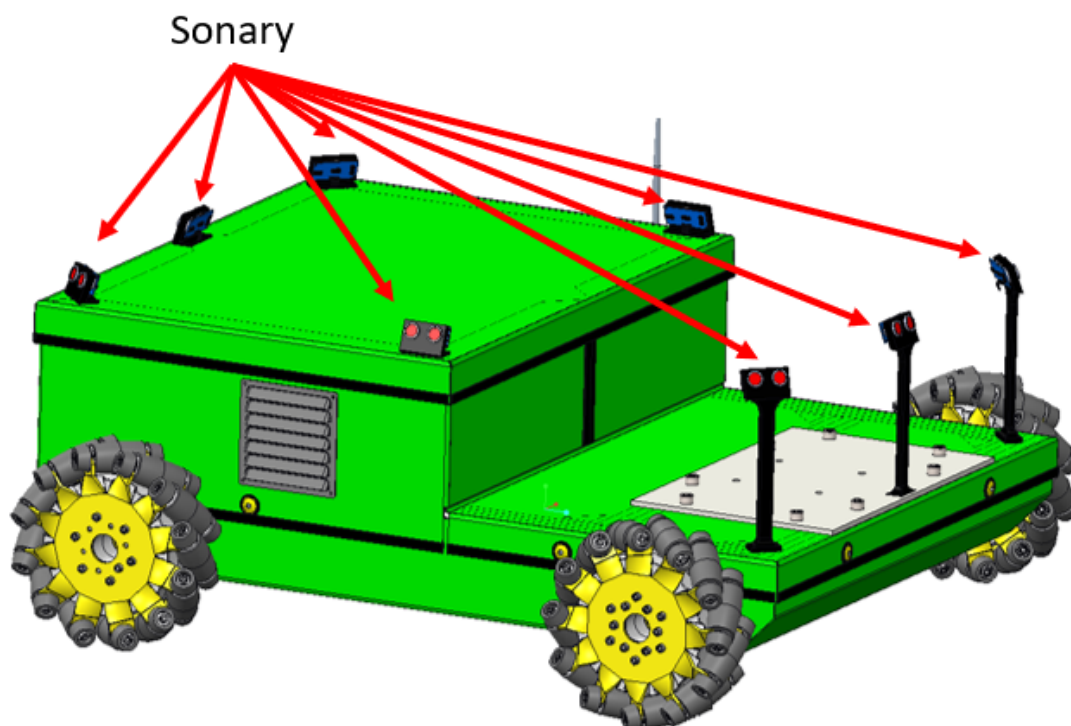
Kapitola pojednává o návrhu senzorů pro potřebný a bezpečný pohyb v daném prostředí. Senzorický subsystém je rozdělen na dvě základní skupiny. První skupina se zabývá detekcí překážek a druhá skupina lokalizací robotu. Pro lokalizaci jsou zde popsány dvě varianty návrhu. V závěru kapitoly jsou vybrány a popsány hardwarové komponenty.

### 4.1 Kolizní subsystém

Pro detekci překážek v definovaném prostředí byl zvolen senzorický subsystém obsahující sonary a lidary.

#### 4.1.1 Sonary

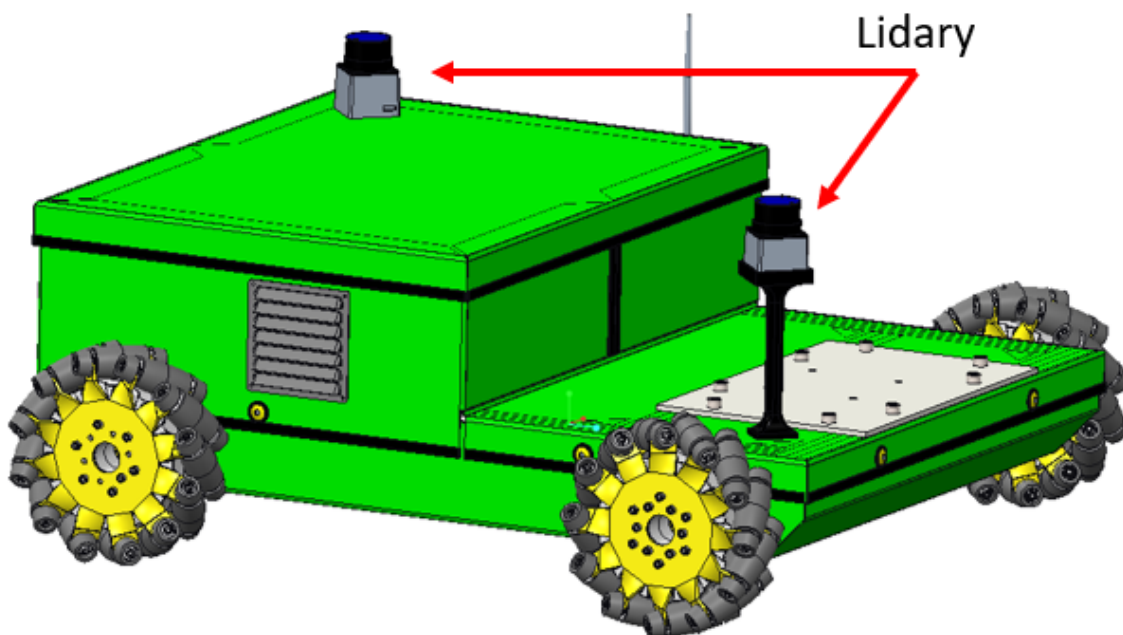
Pomocí sonarů budou detekovány hrubé překážky, kterými jsou například stěny chodby. Rozmístění sonarů je realizováno tak, aby pokrylo rozsah 360° a vzájemně se co nejméně rušily (viz Obr. 15). Ultrazvukové senzory byly zvoleny SRF08. Tento druh senzoru komunikuje přes sběrnici I<sup>2</sup>C. Senzor může využívat až 16 adres. I přes dané adresy lze pro všechny senzory zadat požadavek na měření pomocí základní adresy (0x00) a následně vyčíst jednotlivé hodnoty postupně pomocí reálné adresy senzoru.



Obr. 15 Rozmístění sonarů

### 4.1.2 Lidary

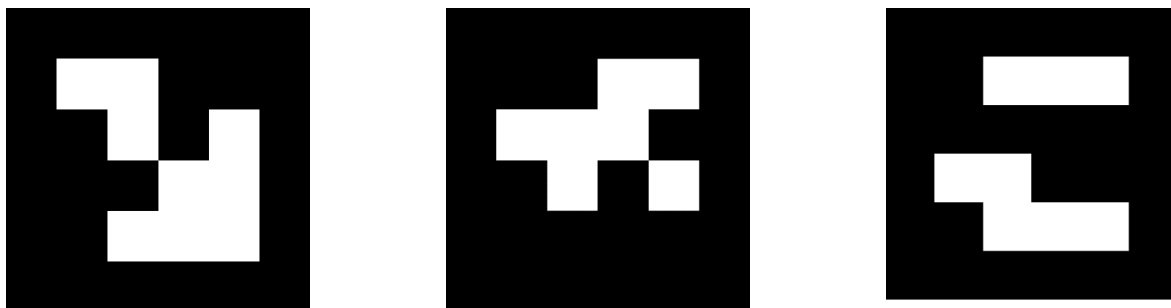
Pomocí lidarů bude realizovaná detekce dynamicky se pohybujících objektů, jako jsou například osoby kráčející v blízkosti robotu. Lidary jsou rozmístěny tak, aby pokrývaly co nejvíce prostoru kolem robotu (viz Obr. 16). Lidary byly zvoleny Hokuyo URG01. Tento senzor lze připojit pomocí USB rozhraní.



Obr. 16 Rozmístění lidarů

### 4.2 Lokalizační subsystém

Lokalizační subsystém byl zvolen na základě pasivních referenčních značek Aruco (viz Obr. 17). V prostředí V-Rep byla vytvořena databáze 70 tagů, které byly rozmístěny po simulačním prostředí tak, aby bylo možné značky rozpoznat pomocí kamer čímž se robot lokalizuje. Pro tento systém byly vytvořeny dvě varianty. Varianty se liší v pozici tagů na mapě a uchycením kamer.

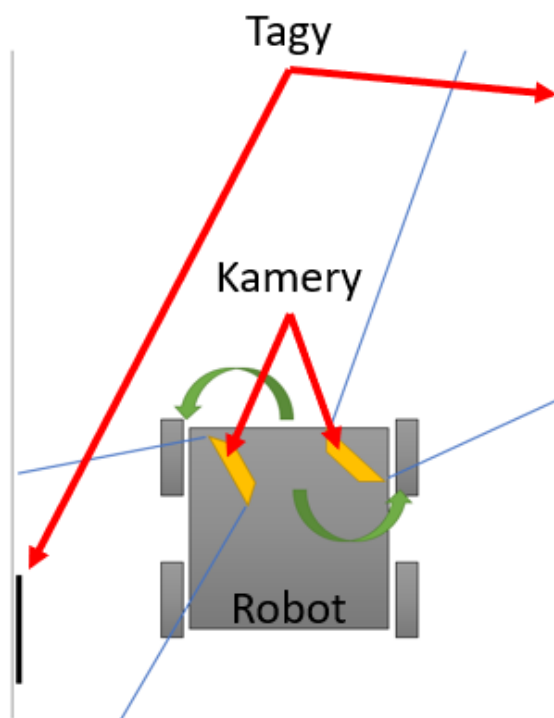


Obr. 17 Aruco tagy



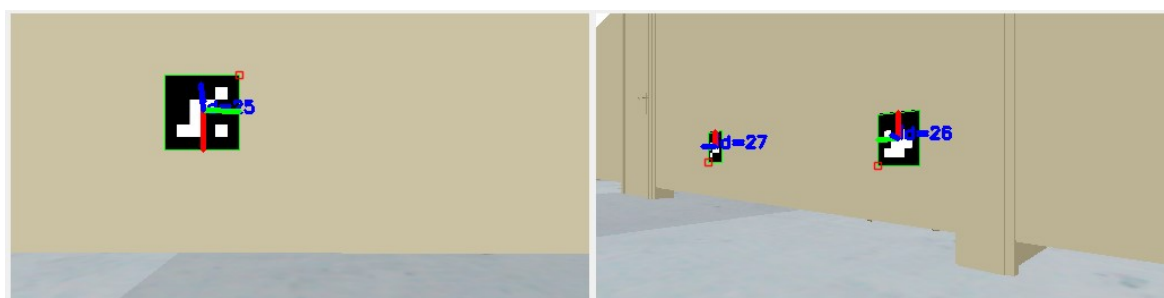
#### 4.2.1 Varianta A

Ve variantě A budou lokalizační kódy rozmístěny v nižších pozicích na stěnách chodby. Aby bylo dostatečně umožněno správnou detekci kódů, bude robot vybaven dvěma kamerami (viz Obr. 18). Každá kamera bude mít vlastní stupeň volnosti, který bude umožňovat natáčení kolem svislé osy kamery, čímž bude umožněno vhodně natáčet kamery, aby vyhodnocování obrazu proběhlo co nejpřesněji.



Obr. 18 Rozmístění Aruco kódů varianta A - schéma

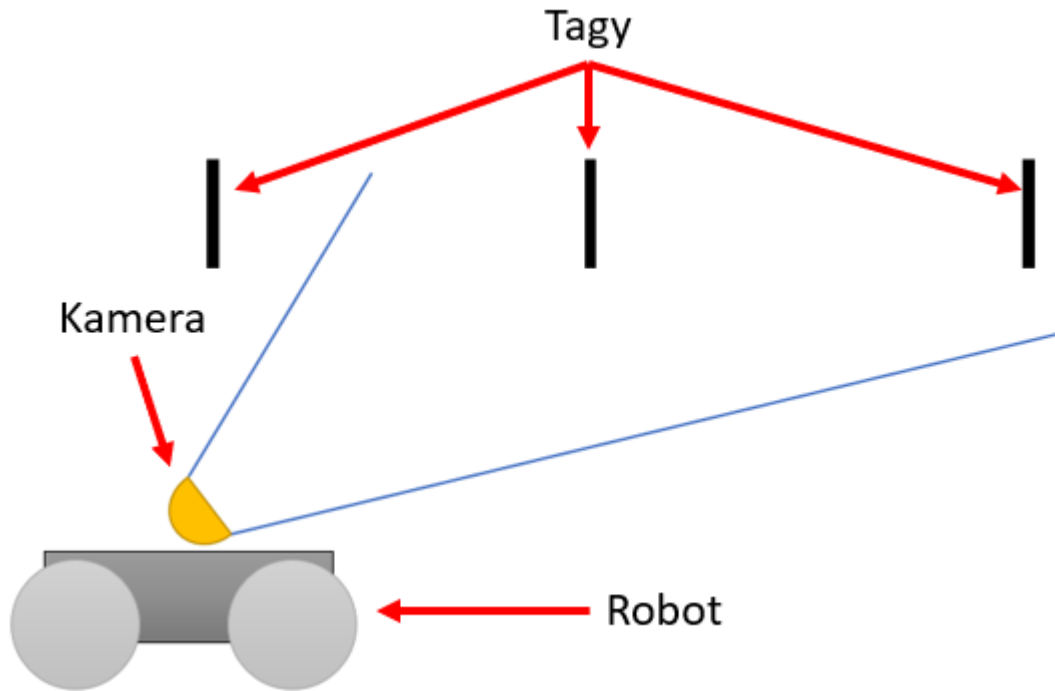
Tato varianta byla simulována (viz Obr. 19) a výsledkem simulace bylo, že i když byly obrazy z kamer dostatečně vyhovující, sledování tagů bylo nevhodné pro následné zpracování. Pozorovací úhel kamer limituje počet sledovaných tagů pro spolehlivou lokalizaci.



Obr. 19 Rozmístění Aruco kódů varianta A - simulace

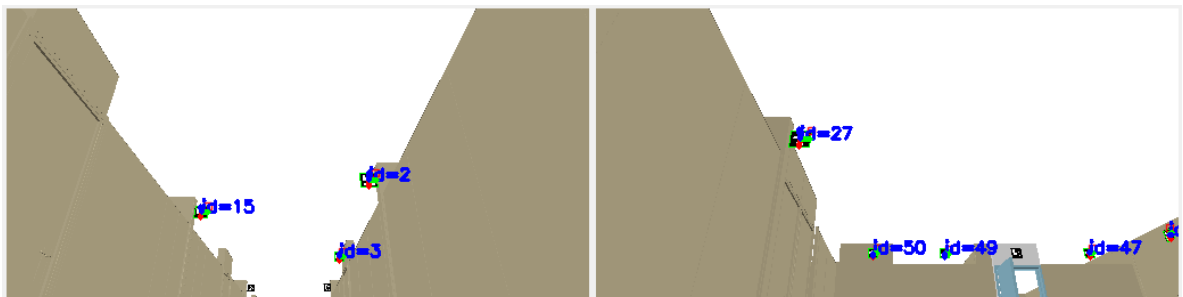
### 4.2.2 Varianta B

Ve variantě B budou tagy rozmístěny kolmo na stěny ve vyšších pozicích. Kamera bude směřovat nahoru tak, aby snímala co nejbližší tagy a zároveň tak, aby tagů bylo detekováno co nejvíce. (viz Obr. 20).



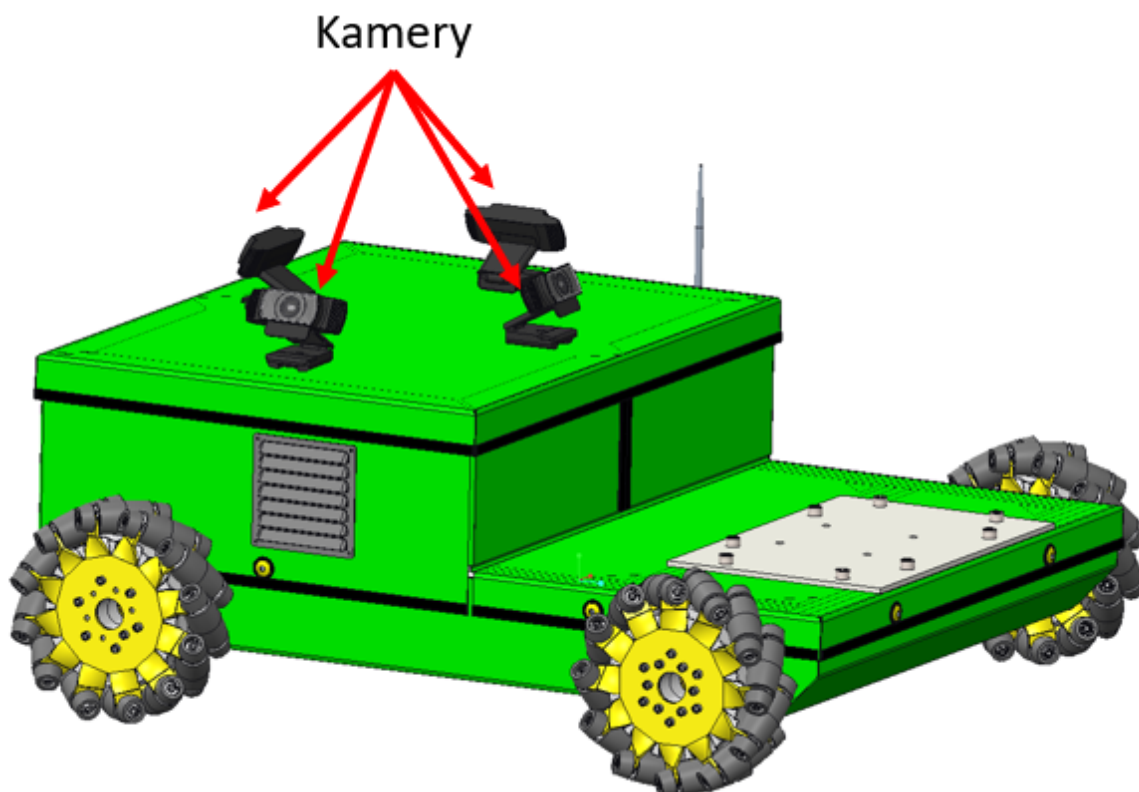
Obr. 20 Rozmístění Aruco kódů varianta B - schéma

Pro tuto variantu byla upravena simulace. Byla provedena změna pozic a orientace tagů, v celém prostoru (viz Obr. 21). Následně zde bylo testováno rozpoznávání tagů a výpočty pozice. Simulace prokázala dostatečný počet rozpoznávaných tagů, pro spolehlivé určení aktuální pozice. Následně zde musel být přidána podmínka, která umožňuje řídit se převážně nejbližšími rozpoznávanými tagy. Toto má za následek přesnější určování pozice z důvodů eliminace chyby detekce vzdálených tagů.



Obr. 21 Rozmístění Aruco kódů varianta B - simulace

Simulace vykazuje s tímto rozpořádáním tagů a kamery výrazně přesnější lokalizaci. Přesto tato varianta neumožňovala lokalizaci ve všech místech a orientaci robotu, protože kamera nenesímala žádný tag. Proto byly přidány na podvozek další 3 kamery, které byly vůči sobě pootočené o 90°, čímž umožňují snímat všechny strany prostředí okolo robotu (viz Obr. 22).



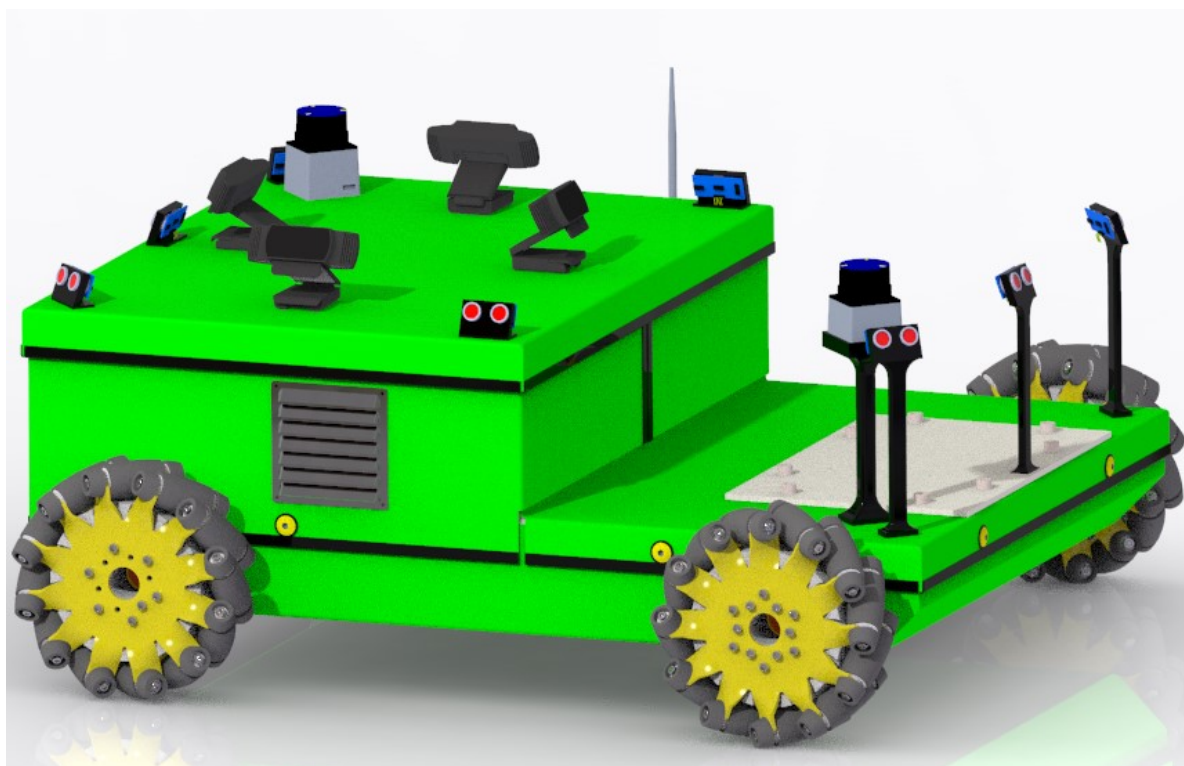
Obr. 22 Umístění kamer

### 4.3 Zhodnocení variant

Aby byla umožněna detekce statických a dynamických překážek, byly na systém navrženy senzory, a to lidary a sonary. Tyto senzory byly vybrány z důvodu odlišné technologie snímání vzdálenosti. Tím je pak eliminována možnost, že z důvodu rozdílného materiálu nebude možno překážku detekovat.

Pro lokalizaci robotu byl zvolen systém na základě referenčních značek. Pro tento systém byly vypracovány dvě varianty. Po simulaci variant A a B, bylo zřejmé že přesnější a kvalitnější rozpořádání kamer je u varianty B. Při snímání více tagů současně lze eliminovat dílčí chyby detekce. Následně je zde zapotřebí vnést do výpočtu parametr vzdálenosti detekovaného tagu. Na základě simulace bylo prokázáno, že s rostoucí vzdáleností tagu od kamery klesá přesnost pozice tagu určené knihovny EmguCV.

Na závěr byl vytvořen celkový 3D model se senzory na robotu. Pro tyto senzory byly vytvořeny zjednodušené držáky. Zde bylo zkontrolováno, zda je rozložení senzorů možné a jejich snímací prostory nebyly narušeny ostatními senzory (viz Obr. 23).



Obr. 23 Celkový senzorický subsystém

Celkový počet senzorů a jejich typ je vypsán v Tab. 4.

Tab. 4 Počty využitých senzorů

Sonary	8 [Ks]
Lidary	2 [Ks]
Kamery	4 [KS]

#### 4.4 Hardwarové komponenty

Tato kapitola se zabývá vlastnostmi a parametry reálných senzorů, které budou aplikovány na podvozek. Dále jsou popsány simulační modely reálných senzorů a jejich nastavené parametry v simulaci. Následně jsou vyhodnoceny rozdíly mezi simulačními a reálnými senzory.

#### 4.4.1 Sonar

Byly zvoleny ultrazvukové senzory SRF08 (viz Obr. 24). Tento druh senzoru komunikuje přes sběrnici I<sup>2</sup>C. Senzor může využívat až 16 adres (viz Tab. 5). I přes dané adresy lze pro všechny senzory zadat požadavek na měření pomocí základní adresy (0x00) a následně vyčíst jednotlivé hodnoty postupně pomocí reálné adresy senzoru.



Obr. 24 Sonar SRF08 [5]

Tyto senzory již byly aplikovány na podvozek, kde ale sloužily pouze jako informativní senzory pro operátora a nepodílely se na žádných autonomních funkcích robotu. Sonary již byly testovány na podvozku a byla zajištěna jejich kompatibilita se systémem.

Tab. 5 Parametry reálného senzoru SRF08

Napětí	5 [V]
Proud	max 15 [mA]
Komunikace	I <sup>2</sup> C bus
Dosah	0,03-6 [m]
Počet adres	16
Pracovní frekvence	40 [Hz]
Rozměry	43x20x16 [mm]

I přesto, že je počet adres omezen, počet senzorů aplikovaných na podvozku nepřekročí danou hranici. Aby se senzory navzájem neovlivňovali při měření, bude se měřit vzdálenost na senzorech v protilehlých směrech v párech. Při změně páru bude dodržena dostatečná prodleva.

#### 4.4.2 Lidar

Pro tuto aplikaci byl vybrán 2D senzor Hokuyo URG-04LX-UG01 (viz Obr. 25), který je k dispozici na naší katedře Robotiky. Pomocí sběrnice USB lze vyčíst data ze senzoru, ale i následně nastavit parametry snímání (viz Tab. 6).



Obr. 25 Hokuyo URG-04LX-UG01 [4]

Lidar je napájen z USB rozhraní, proto není potřeba navrhovat napájecí okruh. Robot disponuje napájecím USB Hubem a tedy nebude stěženo senzory připojit. Řídící jednotka robotu (netduino 2+) prozatím nedisponuje USB rozhraním. Proto bude systém vybaven novou řídicí jednotkou, která zajistí všechny potřebné rozhraní.

Tab. 6 Parametry reálného senzoru Hokuyo URG-04LX-UG01

Napětí	5 [V]
Komunikace	USB bus
Rozsah	240[°]
Dosah	0,02-5,6 [m]
Přesnost	± 3 % měřené vzdálenosti
Délka scanu	100 [ms]
Rozměry	70x50x50 [mm]

### 4.4.3 Kamera

Pro pořizování obrazu byla vybrána kamera Logitech Pro Stream Webcam C922 PRO (viz Obr. 26). Tato kamera je běžně dostupná a umožňuje korekce špatně osvětleného prostředí. Korekce má za následek přizpůsobení se a nedochází ke špatně nahranému obrazu. Parametry kamery jsou shrnuty do Tab. 7.



Obr. 26 Kamera Logitech [9]

Robot prozatím disponuje řídicí jednotkou netduino 2 +, ale tato jednotka nemá USB interface a potřebný výkon pro připojení 4 kamer. Proto bude systém vybaven novou jednotkou s výkonnějším hardwarem a s rozhraním USB.

Tab. 7 Parametry kamery Logitech Pro Stream Webcam C922 PRO

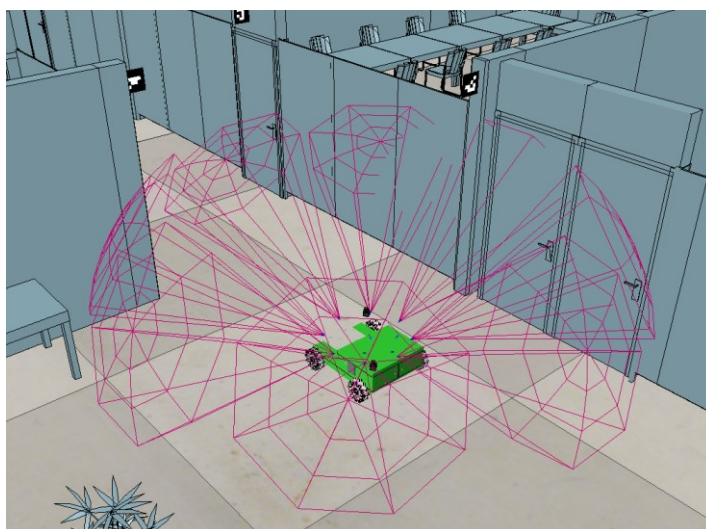
Rozlišení při 30 [fps]	1080p
Rozlišení při 60 [fps]	720p
Technologie objektivu	Skleněný, Full HD
Zorné pole	78 [°]
Hmotnost	162 [g]
Délka kabelu	1.5 [m]
Rozměry	44x95x71 [mm]

## 4.5 Popis sensorických modelů

V kapitole je popsáno nastavení simulačních modelů senzorů, které byly navrženy v předchozí kapitole. Následně jsou v kapitole popsány rozdíly mezi simulačními a reálnými senzory.

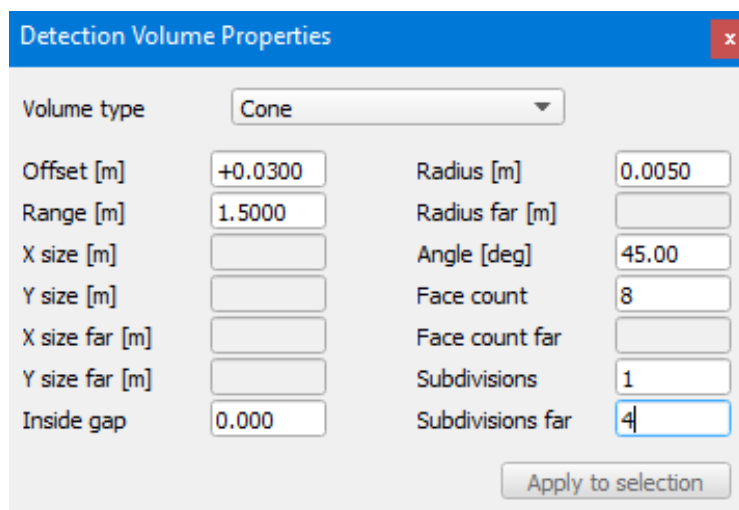
### 4.5.1 Simulační sonar

Pro simulování reálného sonaru byl zvolen „proximity sensor“ typu „Cone“. Tento typ senzoru vhodně napodobuje rozsah snímání reálného senzoru (viz Obr. 27).



Obr. 27 Simulace sonarů

Simulační sonary byly nastaveny podle parametrů reálného sonaru (viz Obr. 28).



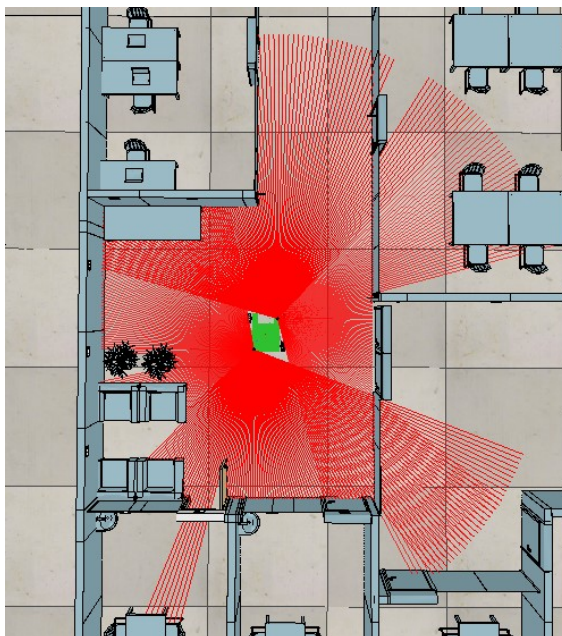
Obr. 28 Parametry sonaru

Rozdíl mezi reálným sonarem je v přesnosti, jelikož simulované prostředí nemá negativní vlivy prostředí a nedochází k ozvěnám jako v reálném případě.



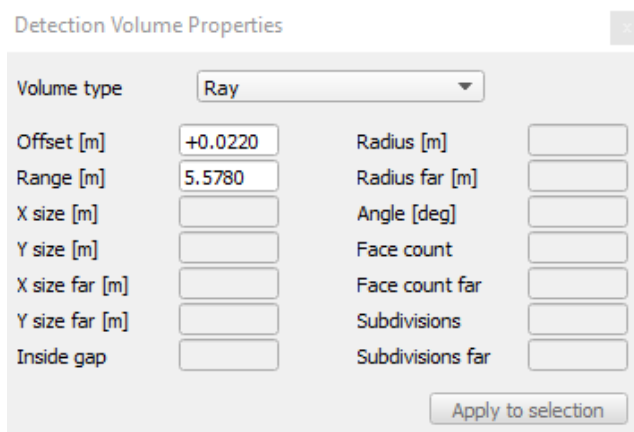
### 4.5.2 Simulační lidar

Tento lidar je k dispozici v základní knihovně, kterou disponuje V-Rep. Princip snímání napodobuje reálný senzor tak, že laserový senzor typu „Ray“ se otáčí kolem svislé osy a měří vzdálenosti v předem definovaném rozsahu (viz Obr. 29).



Obr. 29 Simulace lidarů

Parametry senzoru byly nastaveny podle reálného senzoru (viz Obr. 30). Pro simulační senzor byl nastaven minimální a maximální dosah senzoru a následně omezen rozsah snímání na 240°.

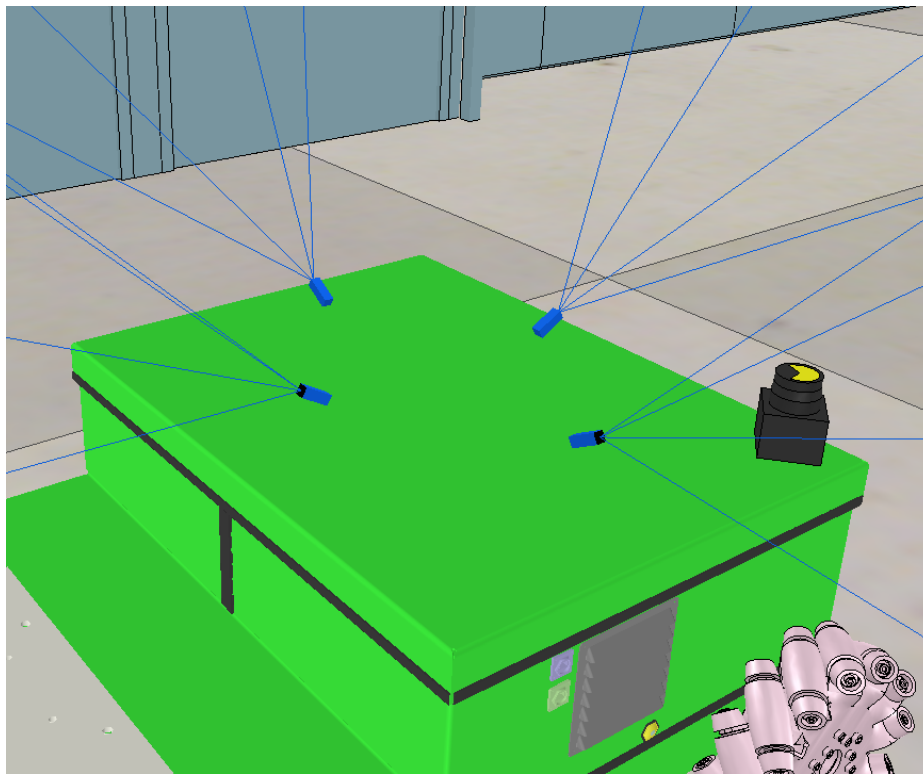


Obr. 30 Parametry lidarů

Hlavní rozdíl mezi reálným lidarem a simulačním lidarem je v přesnosti měření, kde simulační model měří přesně, zatímco reálný senzor měří s procentuální odchylkou vzdálenosti.

### 4.5.3 Simulační kamera

V-Rep disponuje objektem kamera (viz Obr. 31). Objekt umožňuje snímat prostředí, kterému je přednastaveno renderování.



Obr. 31 Simulace kamer

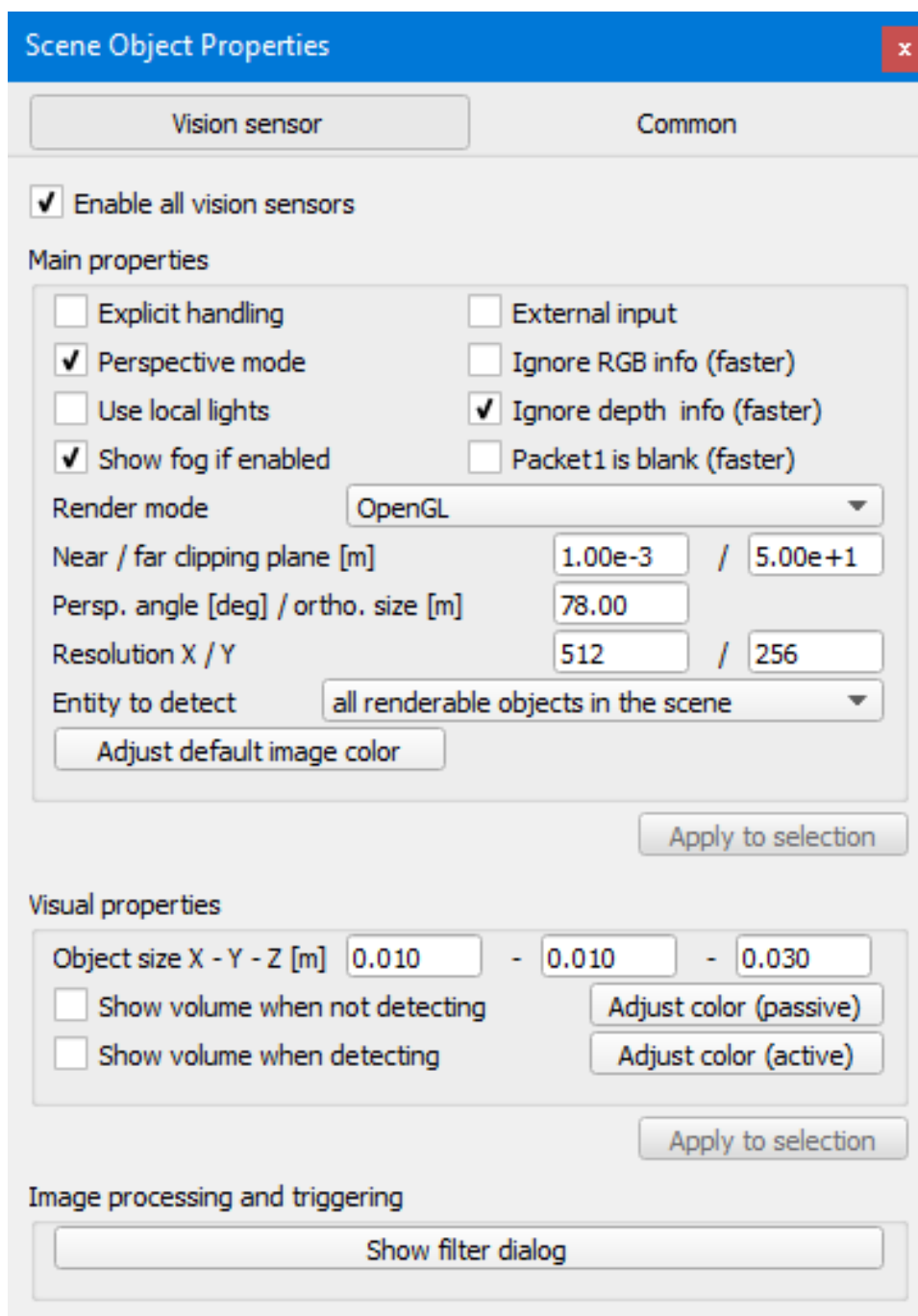
Pro využívání knihoven EmguCV na rozpoznání tagů je zapotřebí nastavit a zkalibrovat kamery. Kalibrace se provádí pomocí snímání tagu a spouštěním kalibračního procesu. Aby kalibrace kamer nemusela probíhat pokaždé, co se systém spustí, byly kalibrační parametry uloženy do souboru, který se při spuštění systému načte a automaticky nastaví kalibraci kamer.

Byly vytvořeny kalibrační soubory pro rozlišení [px]:

- 512x 256
- 640x480
- 1280x960
- 1920x1080

Simulační kamera umožňuje nastavit parametry reálné kamery. Ze simulace vyplynulo, že dostatečně potřebné rozlišení snímání kamer je 512x256 [px]. To má kladný vliv na návrh řídicí jednotky.

Pro všechny kamery na podvozku byly nastaveny parametry zorného pole podle zvolené kamery na 78°. Následně bylo nastaveno rozlišení na 512x256 pixelů (viz Obr. 32).



Obr. 32 Parametry simulační kamery

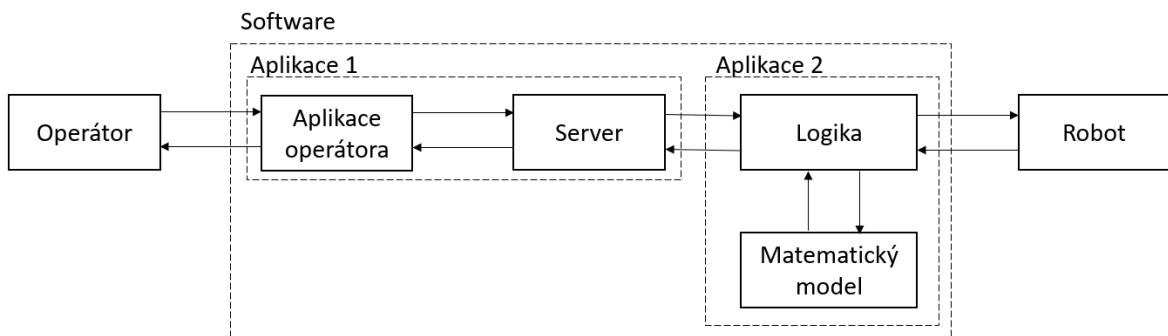
Pro přenesení na reálnou kameru bude zapotřebí vytvořit nové kalibrační soubory. Výsledná kvalita rozpoznávání obrazu nebude ovlivněna, pokud se dodrží přesnost kalibrace s nulovou reprojekční chybou. Tato kalibrace je umožněna prostřednictvím ukázkového příkladu knihoven EmguCV.

## 5 Řídicí systém

Pro testování řídicího systému byla vytvořena simulace, která napodobuje reálný systém. Simulace vychází z reálných parametrů a vlastností materiálů proto, aby byla simulace natolik přesná, aby bylo možné jednoduše aplikovat řídicí systém na reálný robot.

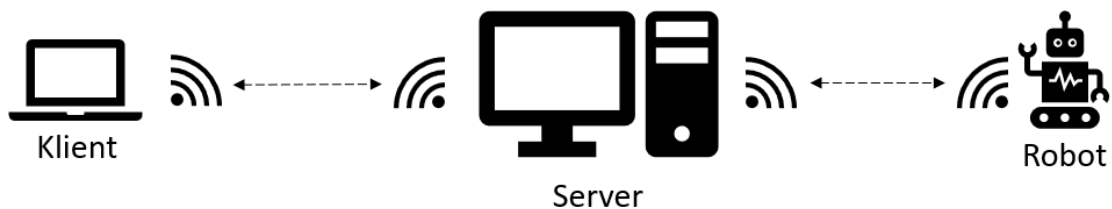
Struktura řídicího systému je vytvořena tak, aby byla jednoduše přenositelná a dala se nakonfigurovat podle potřeby a výkonu výpočetních požadavků.

Celý systém se skládá ze 2 aplikací. Aplikace 1 se skládá z Operátorské části a Serveru, které jsou prozatím na debugování spojeny v jednu aplikaci. Aplikace 2 je složena z Logické části s Matematickým modelem (viz Obr. 33).



Obr. 33 Struktura řídicího systému

Řídicí systém umožňuje řídit právě jednoho robota. Avšak server umožňuje napojit libovolný počet robotů s řídicím systémem, čímž umožňuje operátorovi řídit libovolného robota připojeného na server (viz Obr. 34).



Obr. 34 Schéma sítě

## 5.1 Aplikace 1

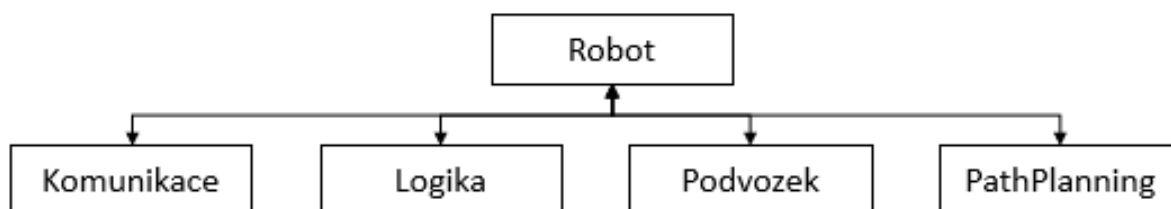
Aplikace 1 je vytvořena tak, aby umožňovala komunikaci robotu s operátorem. Server vytvoří přístupový bod, na který se následně robot s operátorem může připojit. Po připojení robotu na server řídicí systém robotu jako první pošle informace o systému, který se chce připojit na server. Zpráva obsahuje informace o názvu robotu a jeho aktuální nastavení. Tyto informace se pak následně zobrazí v GUI rozhraní pro operátora.

Operátor může nastavit libovolné nastavení a požadavky pro robot, kterými disponuje GUI rozhraní (dále v kapitole 8).

## 5.2 Aplikace 2

Aplikace 2 je vytvořena na to, aby sbírala data od robotu ze senzorů a následně je zpracovávala. Podle aktuálního nastavení robotu, které mu nadefinoval operátor, vykoná dané operace, jako například plánování trajektorie a pohyb do operátorem zadaného místa.

Aplikace obsahuje základní objekt Robot. Tento objekt zaštiťuje celou práci se simulačním nebo reálným robotem. Základní objekty, se kterými objekt Robot pracuje, jsou Komunikace, Logika, Podvozek a PathPlanning (viz Obr. 35).

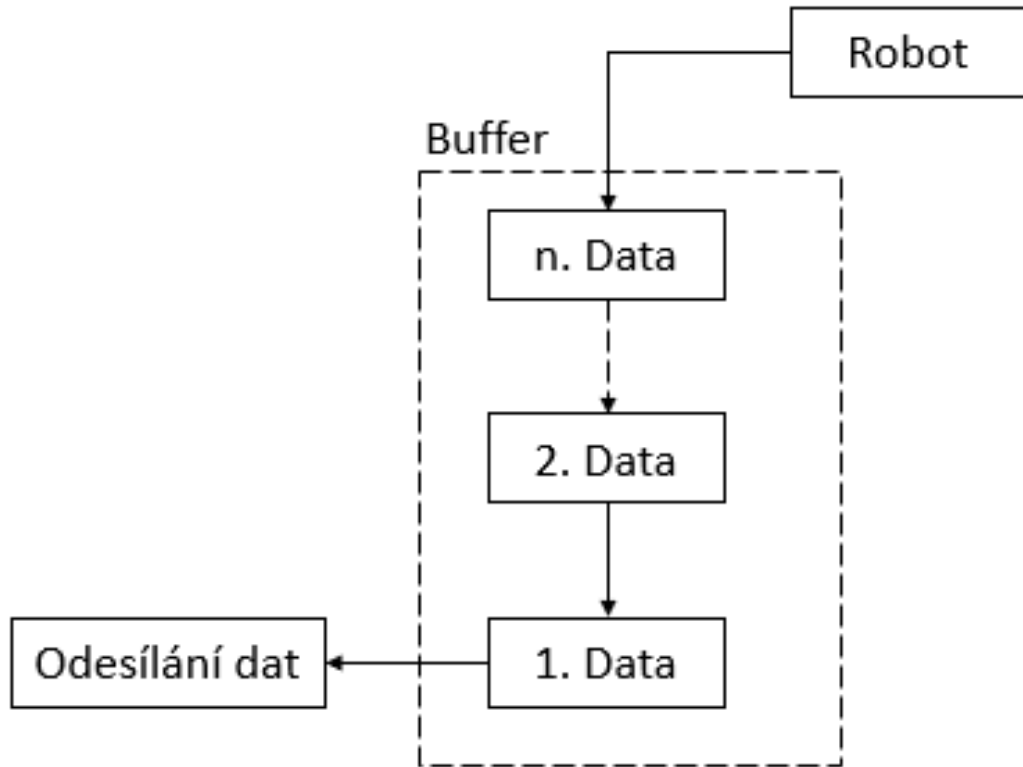


Obr. 35 Blokové schéma aplikace 2

### 5.2.1 Komunikace

Objekt komunikace zprostředkovává komunikaci s operátorem pomocí serveru. Tento objekt zpracovává data k odeslání a přijatá data do předem definovaných šablon. Po založení komunikace se systém začne automaticky připojovat na předem definované místo v síti. Po připojení pošle systém první zprávu o informacích robotu na server. Objekt se dělí na dvě základní vlákna.

První vlákno umožňuje přijímání dat prostřednictvím komunikační linky. Tyto data následně přeloží do objektu, který vykoná dané zásahy do nastavení robotu, které byly požadovány.

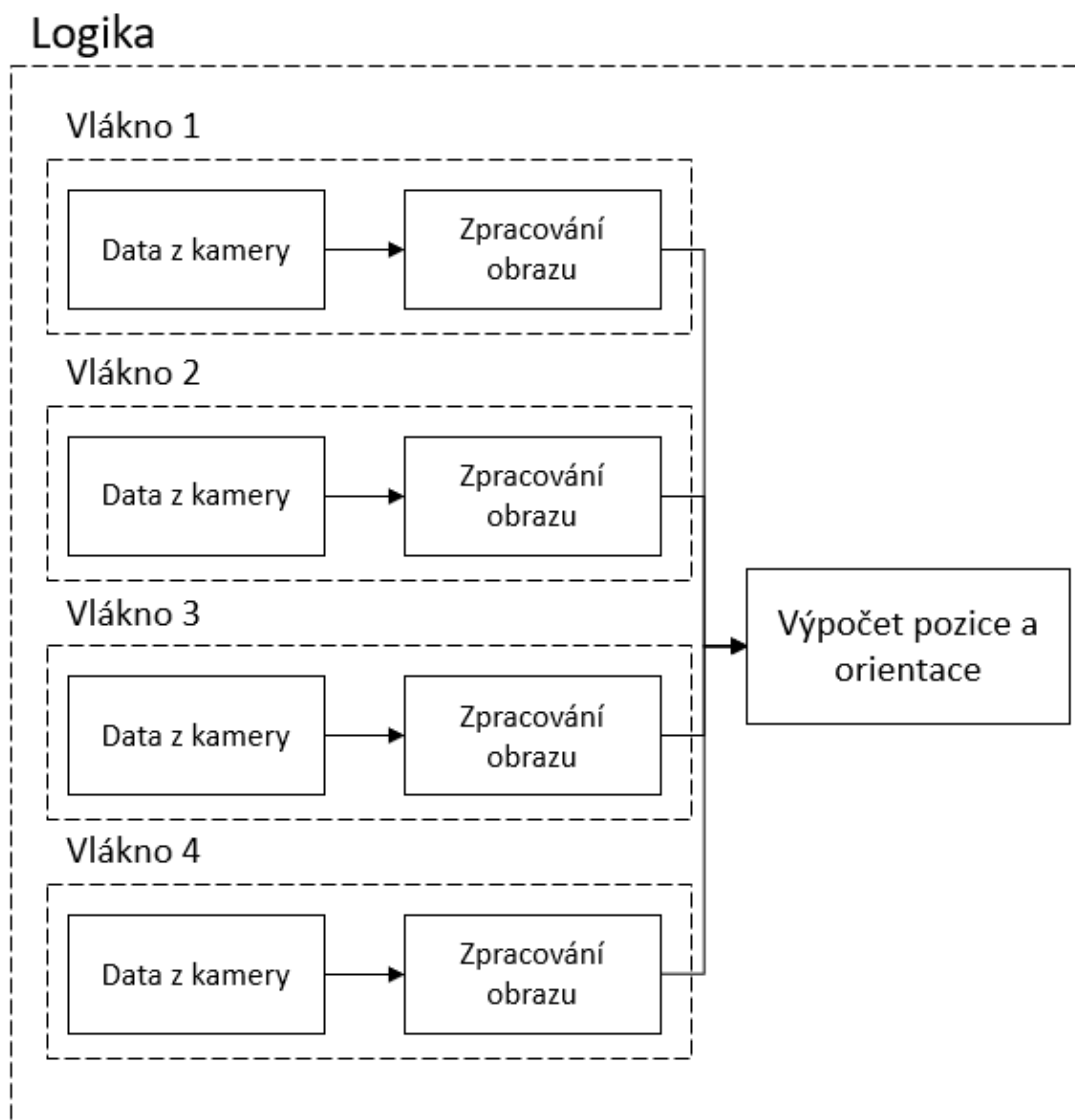


Obr. 36 Blokové schéma odesílání dat

Druhé vlákno naopak umožňuje posílat data operátorovy. Zde je systém bufferu, kde robot ukládá data k odeslání do fronty (viz Obr. 36). Vlákno na odesílání pak prochází frontu a odesílá data v pořadí, v jakém se nashromáždily v bufferu. Data, která byla odeslána jsou následně vymazána.

### 5.2.2 Logika

Tento objekt umožňuje zpracování obrazu a výpočet pozice. Pro každou kameru má objekt vlákno, ve kterém probíhá rozpoznávání tagů (viz Obr. 37). Toto zpracování využívá knihovny EmguCV. Data z rozpoznávaných tagů se ukládají a v pravidelném intervalu se zpracovávají do parametru pozice a orientace robotu.



Obr. 37 Schéma objektu Logika

### 5.2.3 Podvozek

Objekt podvozek obstarává nastavování rychlostí motorů a sbírání dat ze senzorů. Následně zde jsou uloženy základní parametry podvozku. Tento objekt převážně slouží jako přístupový bod k simulačnímu nebo reálnému hardwaru.

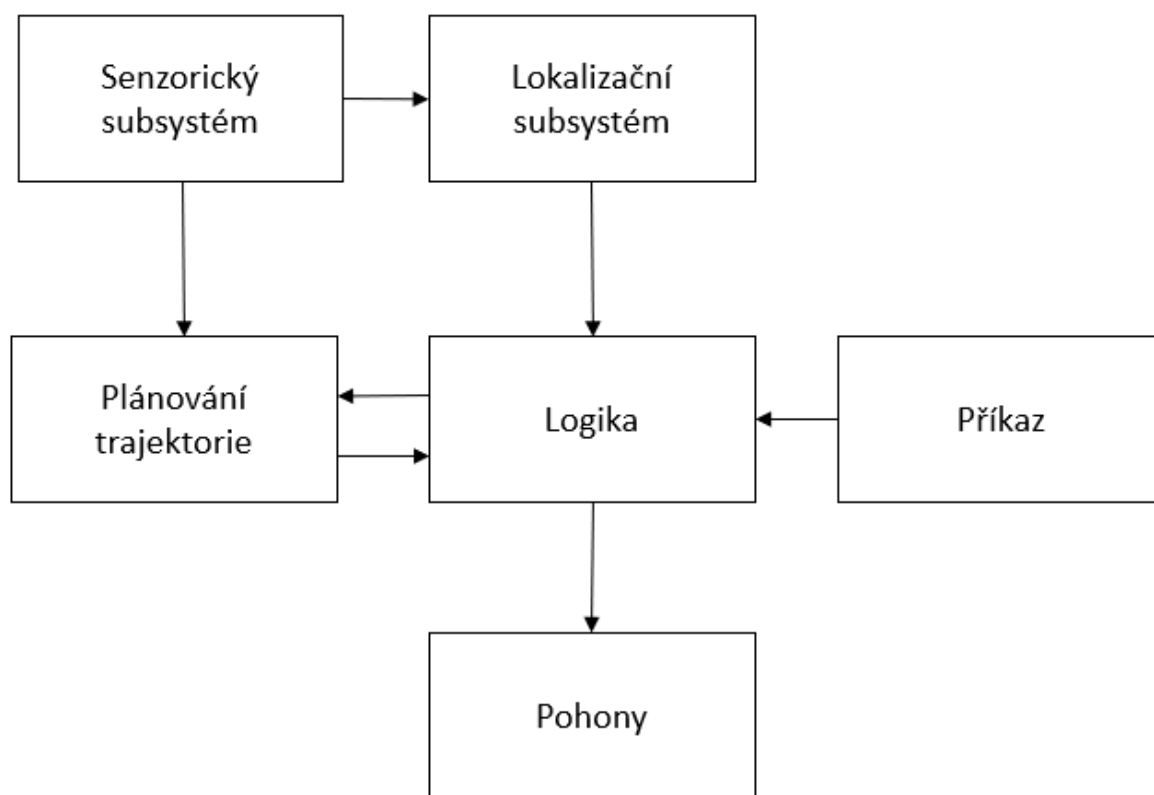
### 5.2.4 PathPlanning

Tento objekt vytváří přístupový bod pro simulaci ve V-Repu s matematickým modelem, ve které se vypočítává trajektorie robotu z aktuálního místa do požadované pozice. Objekt umožňuje nastavovat do simulace data ze senzorů, aktuální polohu a požadovanou polohu. Výsledkem je pak vektor pohybu, kterým se robot má pohybovat, aby postupoval po vypočítané trajektorii a tím tak dorazil do cílové pozice.

## 6 Navigační subsystém

Navigační systém robotu je jednou ze základních funkcí autonomních robotů. Systém umožňuje vypočítat aktuální polohu robotu a nadefinovat teoretickou trajektorii, kterou by měl robot vykonat, aby docílil požadované polohy [6].

Systém, který byl vytvořen, umožňuje robotu vypočítat aktuální pozici a natočení v prostředí, a následně naplánovat trajektorii pohybu na předem definované místo. Během pohybu se robot autonomně vyhýbá neočekávaným překážkám. Základní struktura je vyobrazena na Obr. 38.

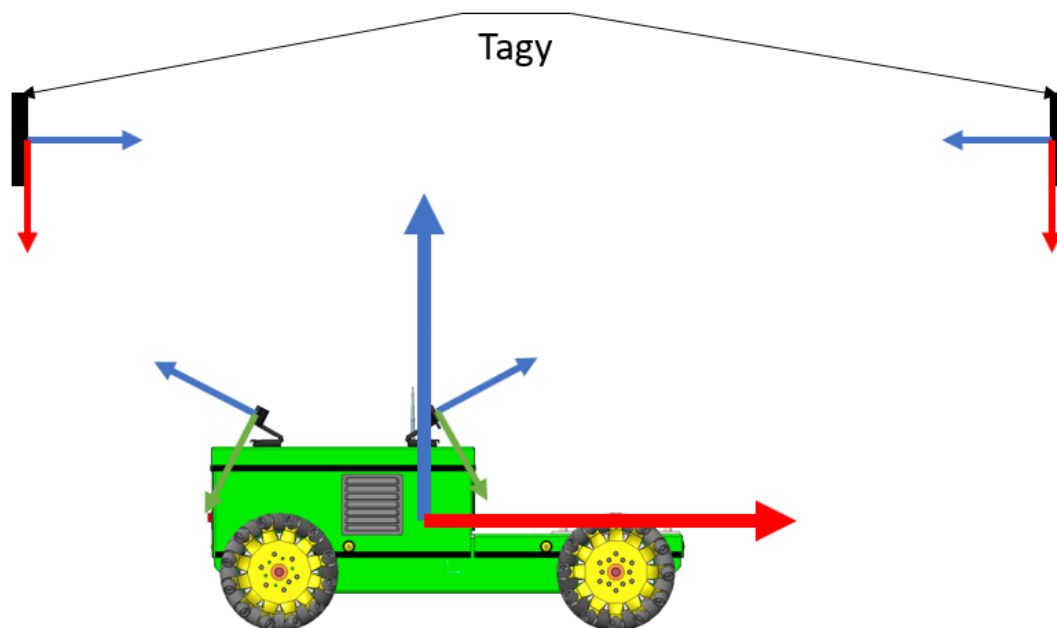


Obr. 38 Blokové schéma navigačního subsystému

### 6.1 Lokalizace

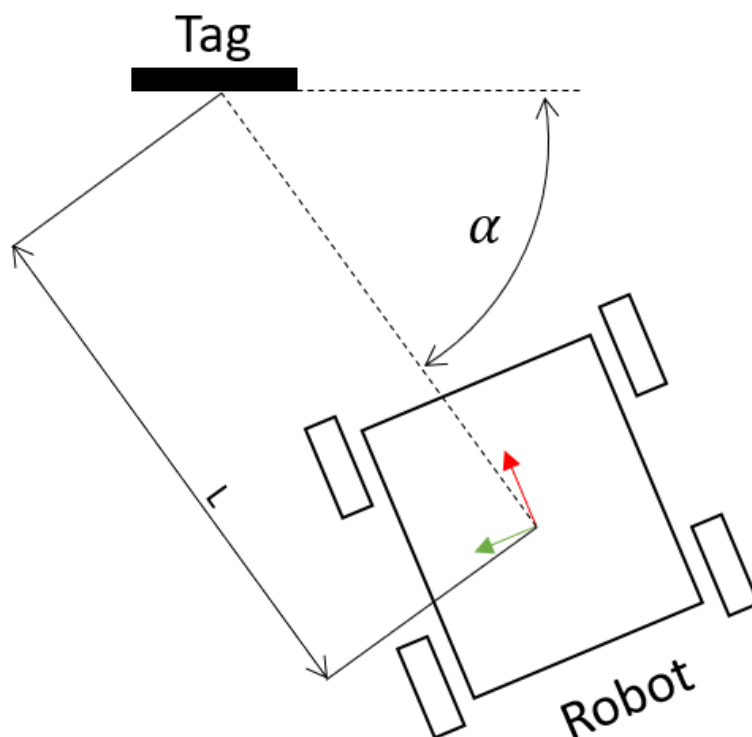
Výpočet aktuální polohy je realizován pomocí rozpoznávání obrazu. Na snímku se vyhledávají Aruco tagy, pomocí knihoven EmguCV. Algoritmy, které rozpoznají tyto tagy a jejich identifikační číslo, umožňují výpočet vzdálenosti a orientaci tagu vůči kameře. Tyto hodnoty jsou klíčové pro následný výpočet polohy robotu vůči detekovaným tagům.





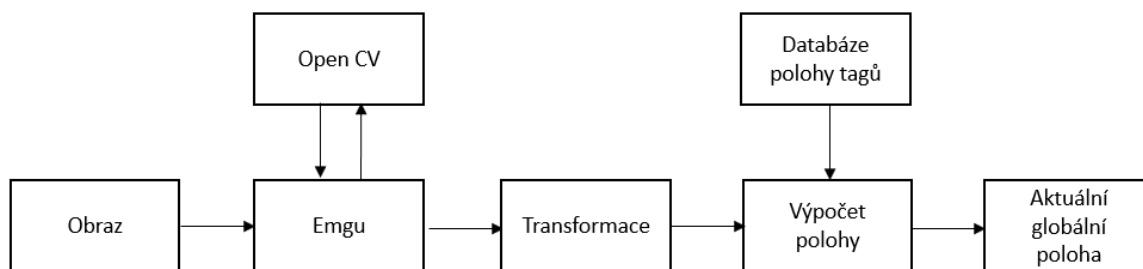
Obr. 39 Souřadnicové systémy výpočtu pozice robotu

Hodnoty jsou pomocí transformačních matic transformovány do souřadnicového systému robotu (viz Obr. 39). Hodnoty jsou přepočítány na vzdálenost  $L$  a úhel  $\alpha$  (viz Obr. 40 ).



Obr. 40 Schéma výpočtových parametrů

Následně je podle známé polohy tagu a známé polohy robotu vůči tagu vypočítána globální poloha robotu (viz Obr. 41).

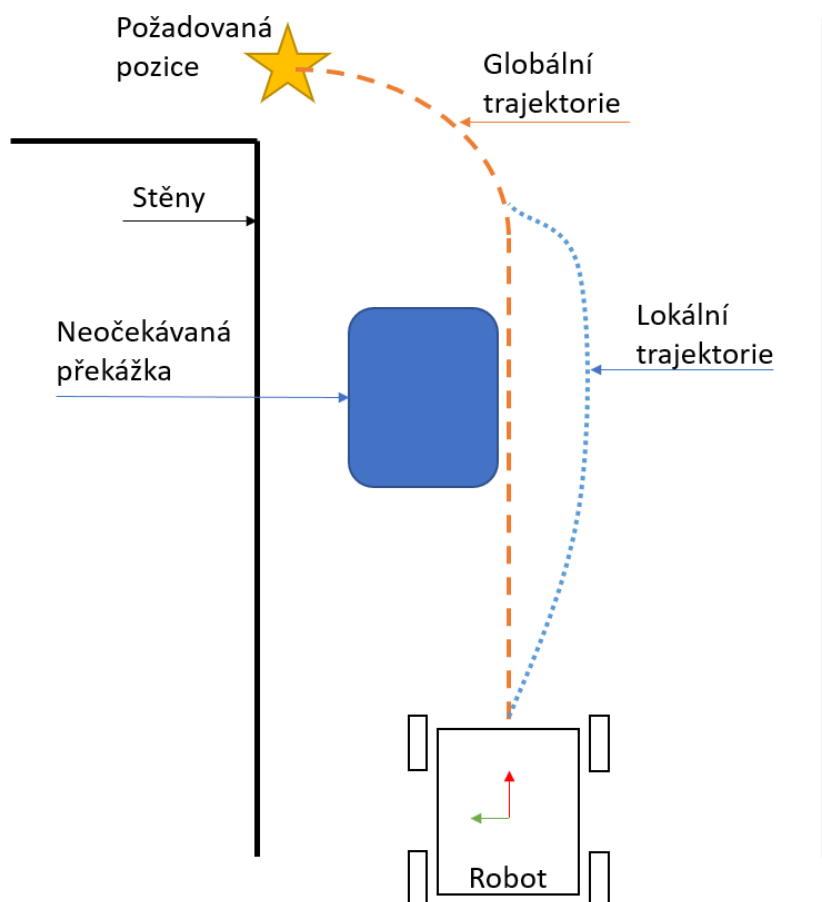


Obr. 41 Blokové schéma lokalizačního subsystému

## 6.2 Plánování trajektorie

Jakmile je známa aktuální poloha robotu, operátor může vyslat požadavek na změnu polohy. Zde je zapotřebí plánování trajektorie tak, aby byl robot schopný autonomně dosáhnout požadované polohy.

Jelikož se robot bude pohybovat v dynamicky se měnícím prostředí, plánování trajektorie je rozděleno na dvě části, a to globální a lokální (viz Obr. 42).



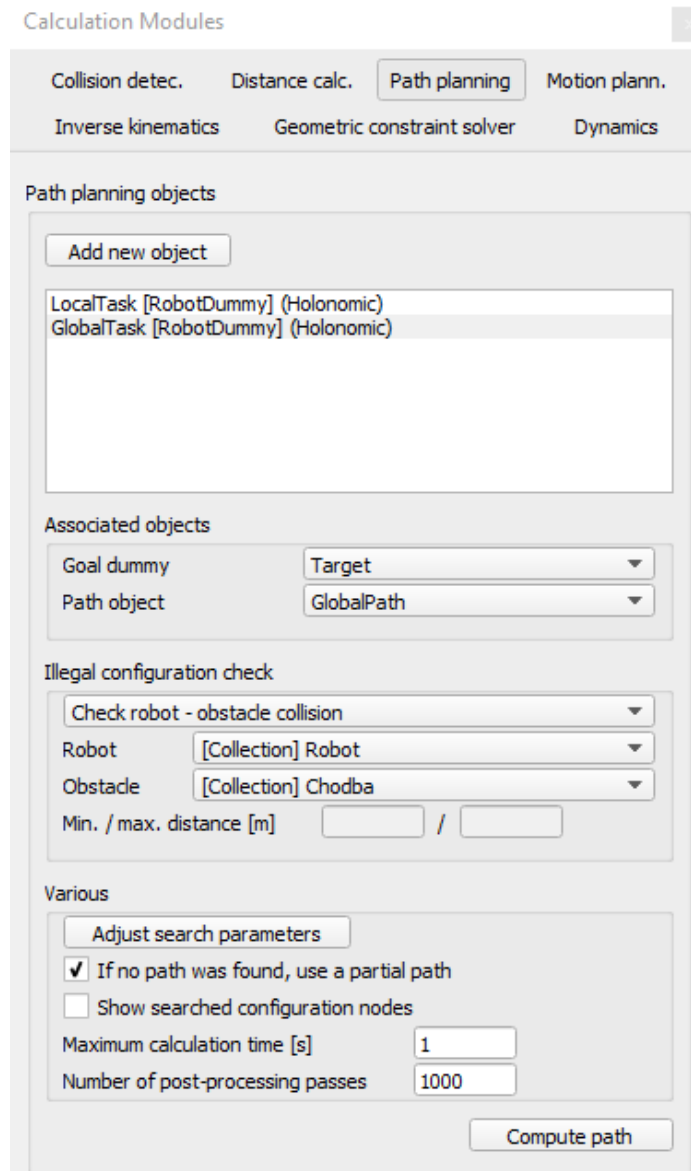
Obr. 42 Schéma plánování trajektorie

## 6.3 Globální trajektorie

Globální trajektorie je vypočítána od aktuální pozice robotu až do cílové pozice. Výpočet globální trajektorie počítá pouze s předem definovaným prostředím. Tato trajektorie slouží pouze pro vytvoření kontrolních bodů. Tyto body pak slouží pro cílové polohy lokální trajektorie.

### 6.3.1 Nastavení algoritmu ve V-Repu

Pro výpočet trajektorie v prostředí V-Rep je zapotřebí nadefinovat počáteční stav, koncový stav a objekt trajektorie, kterou algoritmy přetvoří do finálního vypočítaného tvaru (viz Obr. 43).



Obr. 43 Nastavení globální trajektorie

Pro výpočet globální trajektorie je definován objekt path s názvem GlobalPath, který je využíván pro přetváření na vypočítanou trajektorii globálního plánování. Začátek výpočtu vychází z pozice robotu a končí v cílovém bodu, na který je napažicován objekt dummy s názvem Target. Pozice cílového bodu je definována operátorem. Globální plánování počítá pouze kolize mezi robotem a chodbou.

Path Planning Task Parameters

Holonomic type: X-Y-Delta

Delta axis (x/y/z) [m]: +0.00 / +0.00 / +1.00

Linear [m] / angular [deg] step size: 0.500 / 5.00

Min. turning circle diameter [m]:

Visualize the search area

Search minimum		Search range		Search direction	
X [m]	-4.2e+1	X [m]	+8.4e+1	X	<input checked="" type="checkbox"/> - <input checked="" type="checkbox"/> +
Y [m]	-8.0e+0	Y [m]	+1.6e+1	Y	<input checked="" type="checkbox"/> - <input checked="" type="checkbox"/> +
Z [m]		Z [m]		Z	<input type="checkbox"/> - <input type="checkbox"/> +
Delta [deg]	+1.8e+2	Delta [deg]	+3.6e+2	Delta	<input checked="" type="checkbox"/> - <input checked="" type="checkbox"/> +

Close

Obr. 44 Nastavení parametrů globální trajektorie

Parametry vyhledávací oblasti jsou nastaveny tak, aby pokryly ve všech pozicích na mapě celou mapu, tudíž algoritmus nalezne trajektorii pokaždé. Body trajektorie mají parametry X a Y, které určují pozici a úhel Delta, který určuje natočení okolo osy Z (viz Obr. 44).

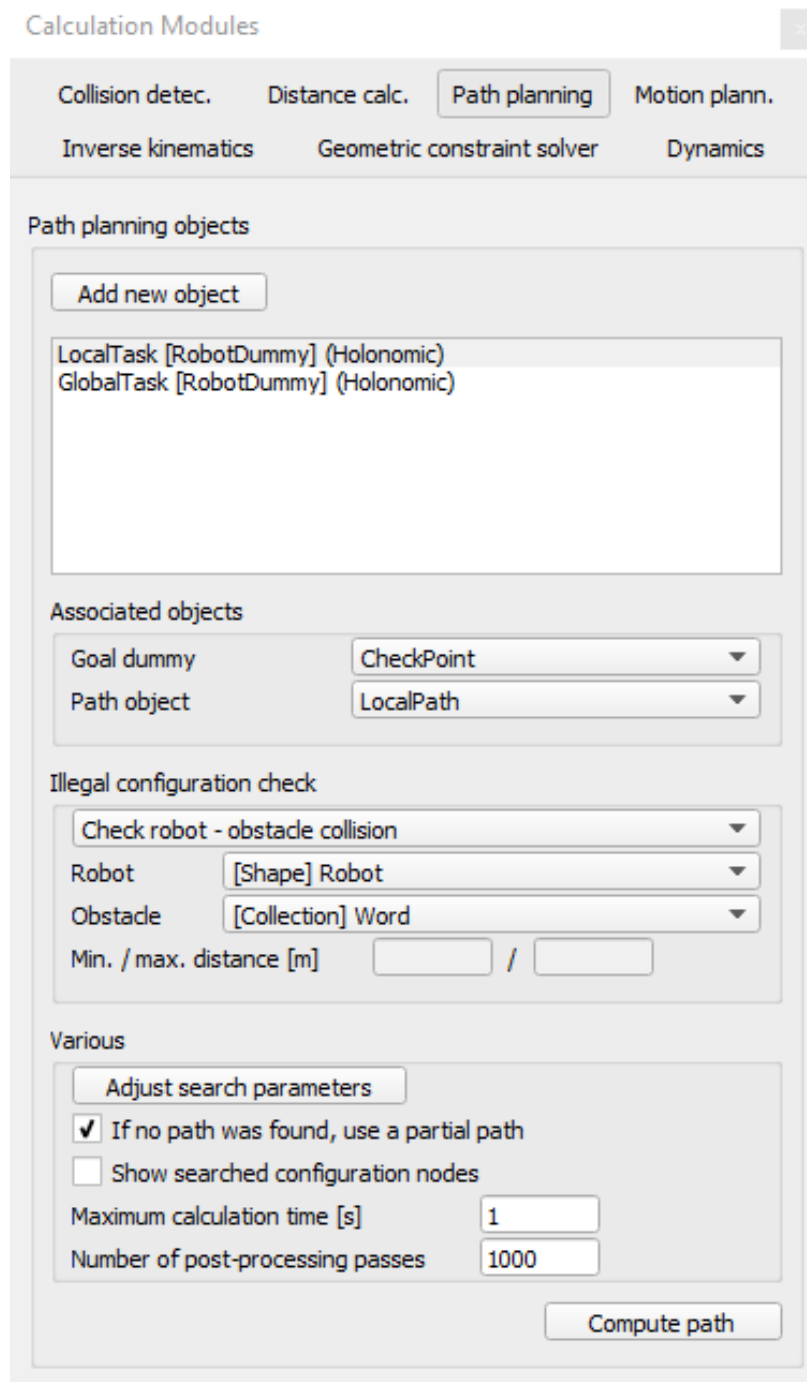
## 6.4 Lokální trajektorie

Výpočet lokální trajektorie následně počítá s daty ze sensorů a kontrolními body z globální trajektorie.

Lokální trajektorie vychází z aktuální pozice a končí v bodě na globální trajektorii. Tato trajektorie se vypočítává mezi kolizí robotu a všech kolizních objemů nastavených v simulaci. Díky tomu tato trajektorie počítá s překážkami a robot se tedy vyhýbá překážkám.

### 6.4.1 Nastavení algoritmu ve V-Repu

Koncový bod, do kterého je počítána trajektorie (viz Obr. 45), se pohybuje po globální trajektorii v závislosti na poloze robotu. Vzdálenost je určena dosahem senzorů, aby byla zajištěna maximální efektivita výpočtu.



Obr. 45 Nastavení lokální trajektorie

Parametry výpočtu lokální trajektorie jsou nastaveny podle maximálních dosahů senzorů (viz Obr. 46).

Path Planning Task Parameters

Holonomic type: X-Y-Delta

Delta axis (x/y/z) [m]: +0.00 / +0.00 / +1.00

Linear [m] / angular [deg] step size: 0.300 / 10.00

Min. turning circle diameter [m]:

Visualize the search area

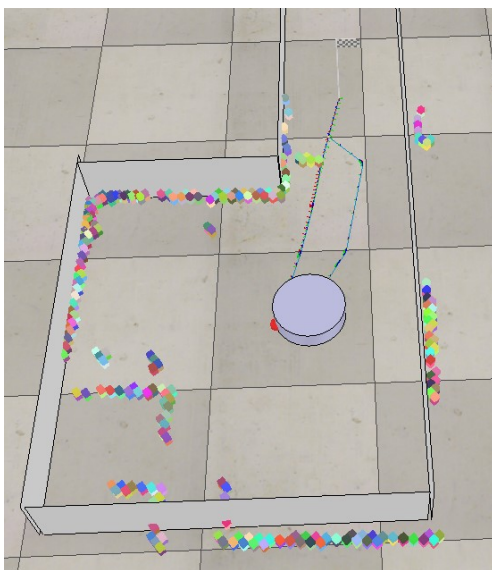
Search minimum	Search range	Search direction
X [m]: -7.0e+0	X [m]: +1.4e+1	X: <input checked="" type="checkbox"/> - <input checked="" type="checkbox"/> +
Y [m]: -3.0e+0	Y [m]: +6.0e+0	Y: <input checked="" type="checkbox"/> - <input checked="" type="checkbox"/> +
Z [m]:	Z [m]:	Z: <input type="checkbox"/> - <input type="checkbox"/> +
Delta [deg]: +1.8e+2	Delta [deg]: +3.6e+2	Delta: <input checked="" type="checkbox"/> - <input checked="" type="checkbox"/> +

Close

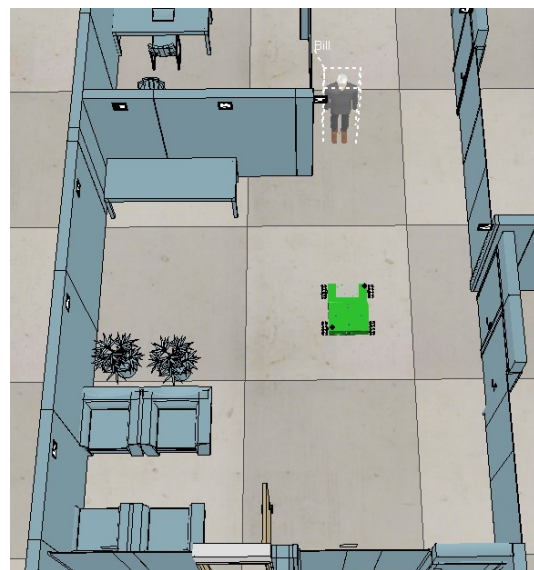
Obr. 46 Nastavení parametrů lokální trajektorie

## 6.5 Simulace výpočtu trajektorie

S nastavením výpočtu trajektorie byla přidána do simulace osoba, která kráčela proti robotu v uličce (viz Obr. 48). Logika systému podle naměřených dat ze senzorů nastavila překážky do simulace matematického modelu, aktuální pozici robotu a přidala cílovou pozici, kterou určil operátor (viz Obr. 47). Podle těchto vstupních dat a předem definovaných algoritmů na výpočet trajektorie pomocí lokální a globální trajektorie systém naplánoval trasu tak, aby se vyhnul překážkám a úspěšně docílil požadované polohy.



Obr. 47 Matematický model



Obr. 48 Simulace systému

## 7 Softwarová část operátora

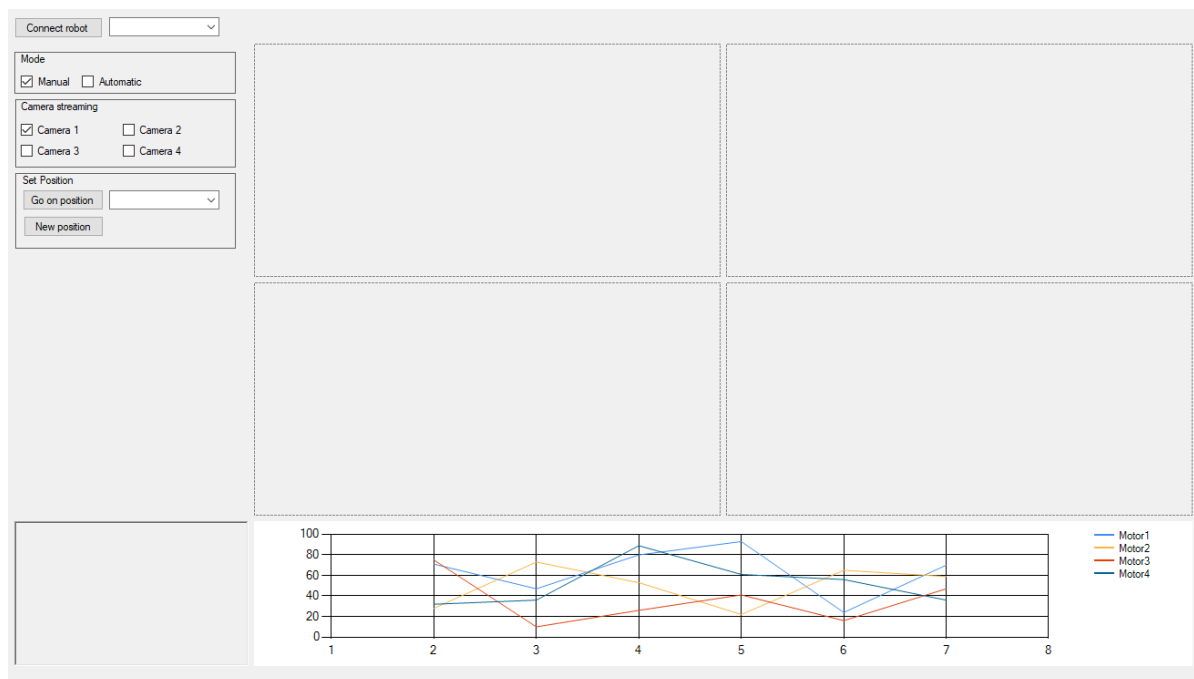
Aplikace pro operátora byla navržena tak, aby umožňovala řídit robota a posílat mu příkazy v reálném čase.

### 7.1 Struktura aplikace

Aplikace je rozdělena do dvou hlavních částí, a to na část operátora a část serveru. Po spuštění aplikace server vytvoří port, na který se může řídicí systém připojit. Jakmile se řídicí systém připojí, pošle serveru zprávu s informacemi o systému. Tyto informace jsou zpracovány a přeposlány do uživatelského rozhraní operátorovy. Pokud by se na server připojovalo více řídicích systémů, operátor může jednoduše vybrat robota z nabídky připojených robotů a ovládat ho.

### 7.2 GUI rozhraní

Rozhraní (viz Obr. 49) umožňuje operátorovy nastavovat parametry a vizualizovat data ze senzorů.



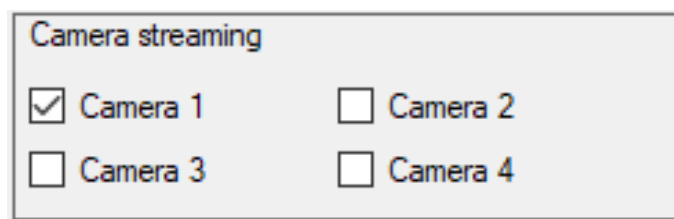
Obr. 49 Rozhraní operátora

Aplikace disponuje přepínáním módu robotu mezi automatickým a manuálním řízením (viz Obr. 50). To je realizováno pomocí prvků „checkbox“. Tyto prvky jsou nastaveny tak, aby nebylo možné mít oba stavy současně zapnuté.



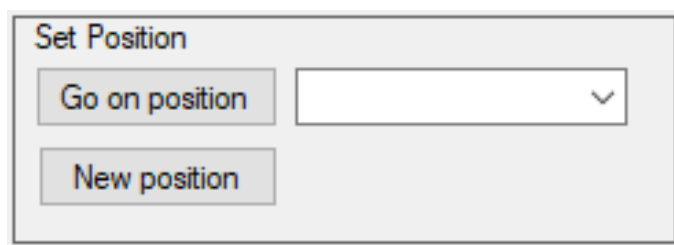
Obr. 50 Nastavování režimů

Následně aplikace umožňuje streamovat data z kamer. Tyto data už jsou přímo zpracované, tudíž jde sledovat zda-li systém funguje tak, jak se předpokládá. Pro vizualizaci těchto streamů jsou připraveny buňky, každá buňka odpovídá právě jedné kameře systému (viz Obr. 51).



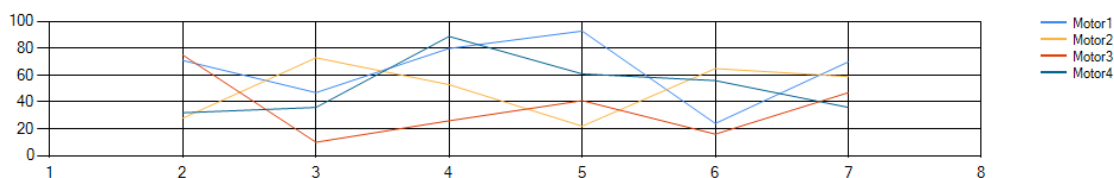
Obr. 51 Nastavování kamer

Pro nastavování cílové pozice robotu je zde sekce, která umožňuje nastavit výslednou pozici, dát příkaz robotu, aby se do dané pozice dostal, anebo vytvořit novou pozici (viz Obr. 52).



Obr. 52 Nastavování pozic

Následně jsou vykreslovány nastavené rychlosti pohonů (viz Obr. 53).

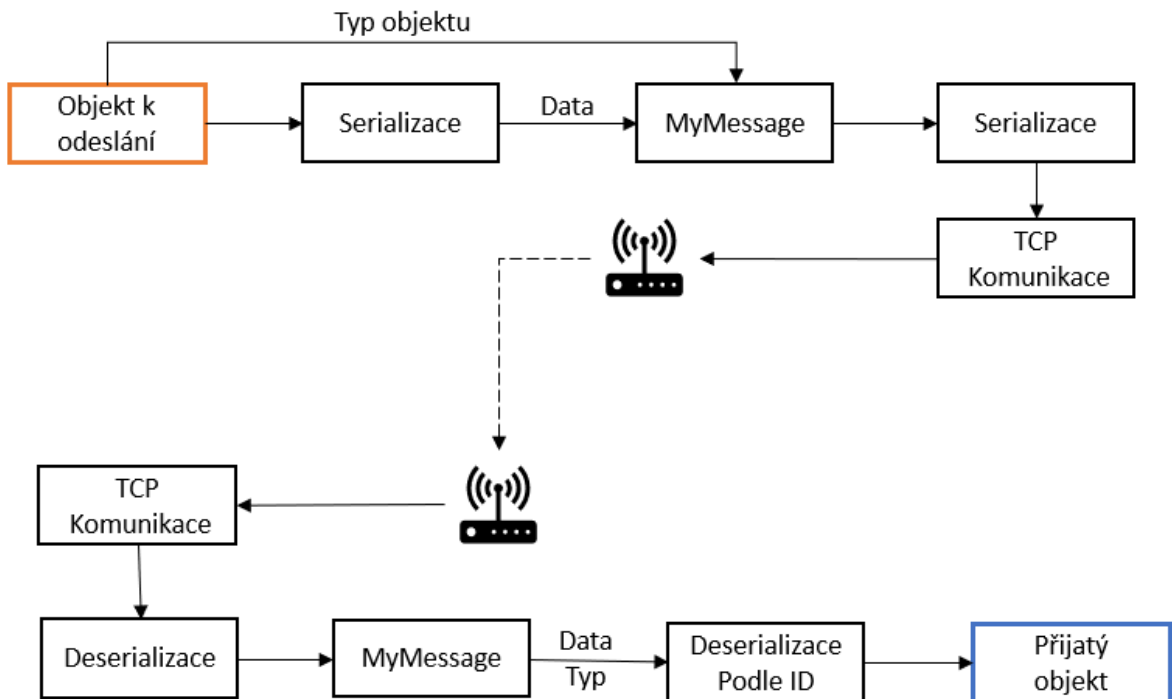


Obr. 53 Graf rychlosti pohonů



### 7.3 Komunikace

Komunikace je vytvořena pomocí TCP spojení (viz Obr. 54). Pomocí této linky jsou posílány zprávy ve formě string. Tento string je serializovaný objekt. Aby bylo možné deserializovat objekt, je potřeba vědět, o jaký typ objektu se jedná. Proto byl vytvořen objekt „MyMessage“, který je prostředníkem k odesílání zprávy. Tedy všechny zaslání a přijaté zprávy jsou typu MyMessage. Tento Objekt je specifický tím, že přeposílá 2 parametry. Prvním parametrem jsou Data. Tyto data jsou serializovaný objekt, který reálně chceme odeslat. Druhým parametrem je ID. Toto ID reprezentuje objekt, který je serializován v parametru data. Každý objekt posílaný přes komunikační linku má svoje specifické ID, tudíž je zaručena správná deserializace objektu.



Obr. 54 Přeposílání objektů

Jakmile lze posílat objekty mezi operátorem a řídicím systémem robotu, lze jednoduše přenést jakákoliv data. Podmínkou je však mít naprosto stejné šablony pro objekty, které se odesílají a přijímají.

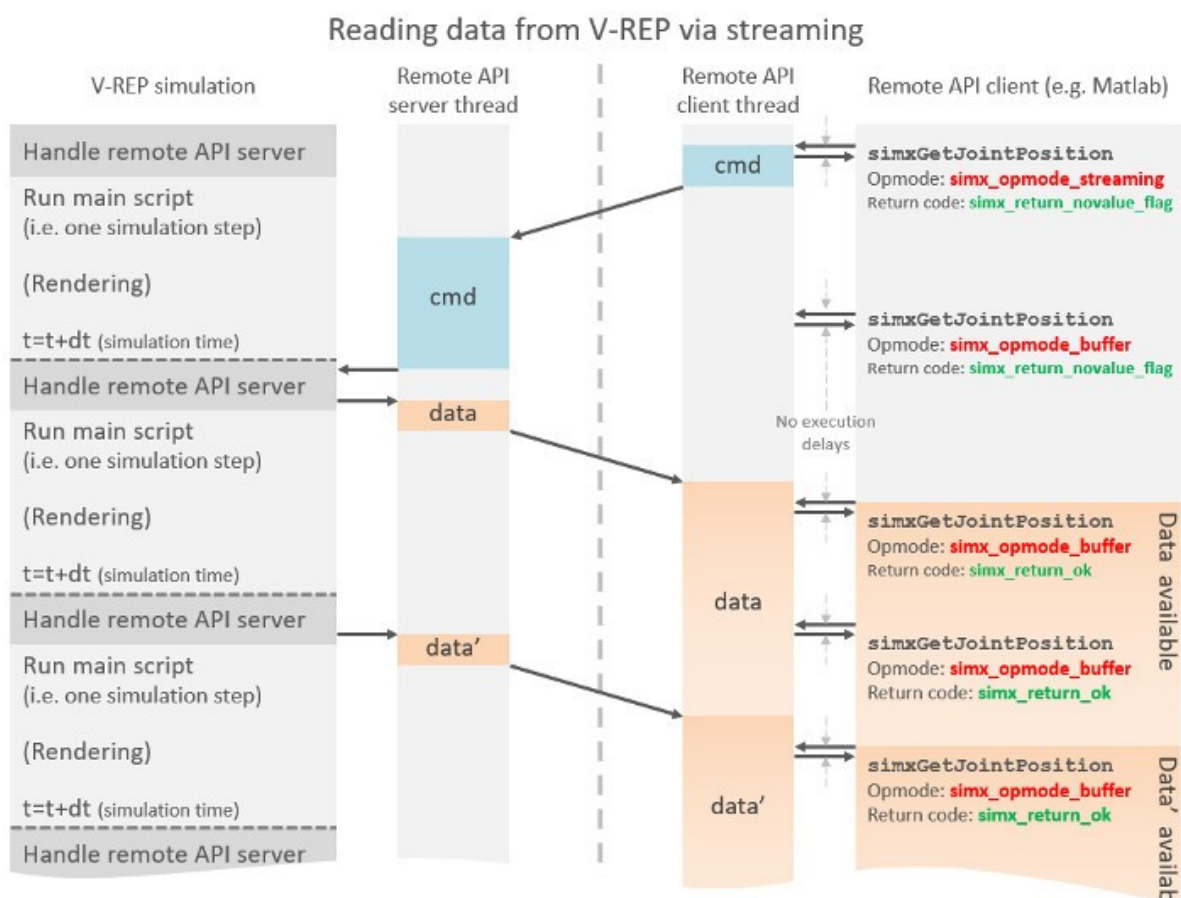
## 8 Softwarová část simulačního modelu

Program, který ovládá simulační model, byl navržen ke snadné aplikaci programu na reálný robot. Simulační model je přizpůsoben tak, aby reálně napodoboval veškeré funkce a vlastnosti, jaké má reálný systém. Simulační model tedy disponuje sběrem dat ze senzorů a možností nastavit rychlosti pohonů, kterými disponuje podvozek.

Simulační model je přizpůsoben tak, aby neustále aktualizoval data ze senzorů a tyto data následně ukládal do bufferu. Tento buffer je neustále k dispozici řídicímu systému, který následně tyto data dále zpracovává.

### 8.1 Komunikace s nadřazeným systémem

Simulační model byl vytvořen tak, aby automaticky sbíral data ze senzorů a kamer. Tyto data jsou ukládána do vyhrazené paměti. Řídicí systém pak následně přistupuje k vyhrazené paměti a čte potřebná data (viz Obr. 55).



Obr. 55 Streamování dat z V-Repu [15]

## 9 Ukázky simulace

Kapitola obsahuje snímky ze simulací v různých situacích. Zde je popsáno, jak reaguje výpočetní model na různé vzdálenosti robotu od požadované polohy. Dále je vyobrazeno, jak výpočetní model reaguje na překážky, které má v cestě nebo poblíž pohybu.

Pro modelovou situaci, kdy robot nemá překážky na krátkém úseku až do zvoleného cíle (viz Obr. 56), má výpočetní model lokální a globální trajektorii totožnou (viz Obr. 57).

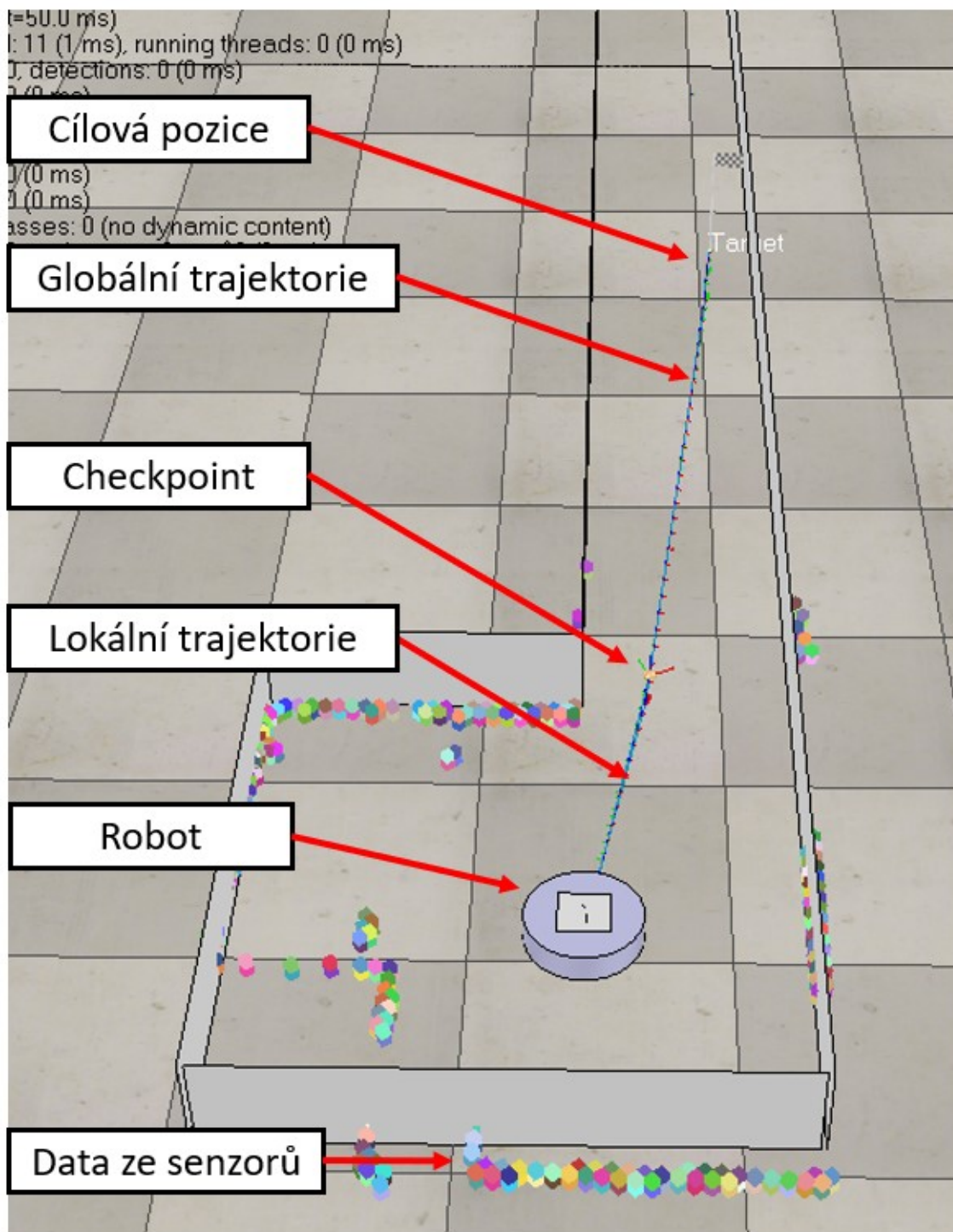


Obr. 56 Simulace bez překážky



Obr. 57 Výpočetní model - bez překážky na krátké vzdálenosti

V případě, kdy robot nemá překážky a pohybuje se po delší trase (větší vzdálenost, než je dosah snímání senzorů), systém umístí checkpoint na globální trajektorii a do tohoto bodu plánuje lokální trajektorii. Jelikož v okolí nejsou žádné překážky, lokální trajektorie je téměř totožná jako část globální trajektorie od robotu po checkpoint (viz Obr. 58).

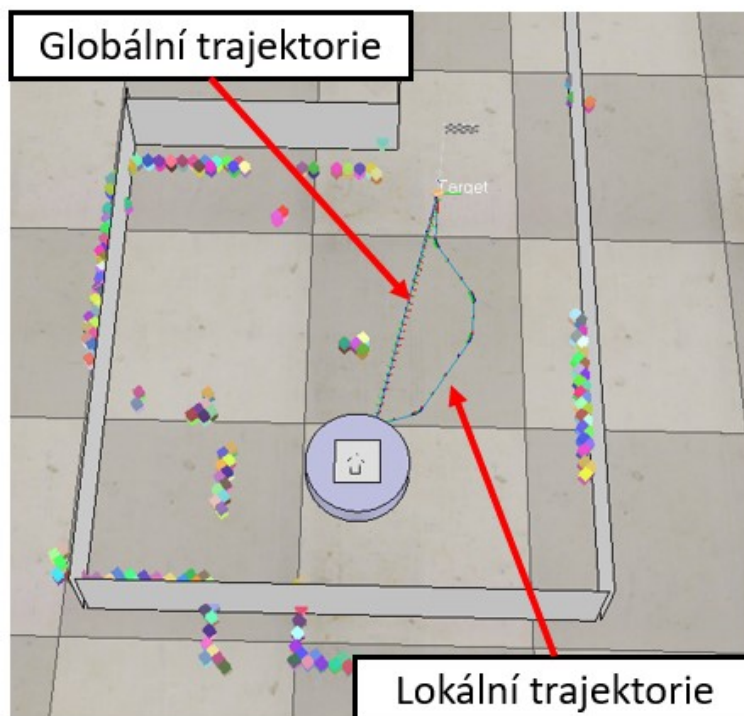


Obr. 58 Výpočetní model - bez překážky na velké vzdálenosti

Další simulovaná situace zachycuje robot dojíždějící do cíle a v cestě stojí osoba (viz Obr. 59). Systém naplánoval lokální trajektorii do cíle tak, aby nedošlo ke kolizi s překážkou (viz Obr. 60). Bezpečná vzdálenost od překážky je volitelná a bude nastavena podle reálných testů systému.



Obr. 59 Simulace s překážkou na krátké vzdálenosti

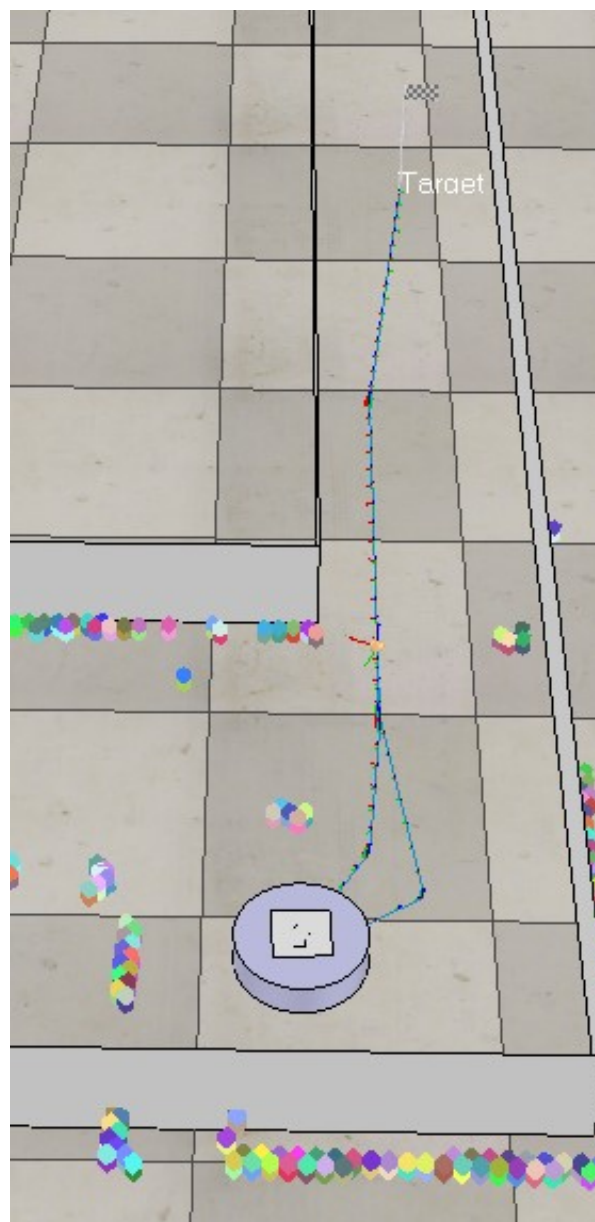


Obr. 60 Výpočetní model – s překážkou na krátké vzdálenosti

Pro simulaci plánování trajektorie na velké vzdálenosti byly přidány do simulace objekty osob. První dvě nejbližší osoby od robotu jsou v dosahu senzorkého subsystému a zbylé dvě jsou za hranicí dosahu senzorů (viz Obr. 61). Systém detekoval první dvě osoby a naplánoval mezi nimi trajektorii tak, aby nedošlo ke kolizi (viz Obr. 62). Od checkpointu je pak dále naplánována pouze globální trajektorie a podle postupu robotu se bude následně měnit trajektorie podle naměřených dat.

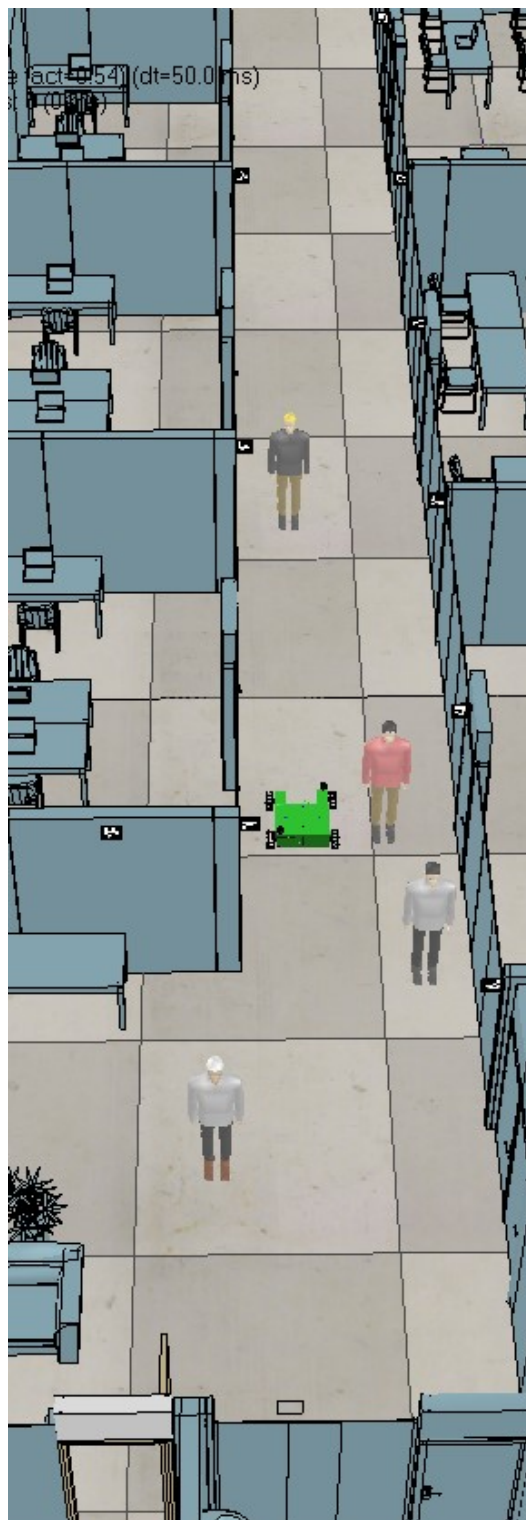


Obr. 61 Simulace s překážkami na velké vzdálenosti



Obr. 62 Výpočetní model – s překážkami na velké vzdálenosti

Na Obr. 63 lze vidět postup robotu v situaci, kdy senzorický subsystém již detekuje všechny objekty osob. Na Obr. 64 jde vidět, jak systém vyřešil danou situaci.



Obr. 63 Simulace s překážkami na velké vzdálenosti



Obr. 64 Výpočetní model – s překážkami na velké vzdálenosti

## 10 Testování systému

Jelikož má katedra robotiky k dispozici prozatím jednu kameru Logitech (viz Kap. 4.4.3), byl otestován pouze lokalizační subsystém na reálném hardwaru.

Řídicí systém byl upraven, aby snímání kamery neprobíhalo ze simulačního modelu, ale z reálné kamery připojené k počítači. Následně byla vytvořena rozšiřující databáze tagů. Tyto rozšiřující tagy byly vytisknuty a rozmístěny v laboratoři (viz Obr. 65). Podle rozmístění byly stanoveny pozice jednotlivých tagů v řídicím systému.



Obr. 65 Testování lokalizace

Po nastavení všech parametrů a spuštění řídicí aplikace, systém spolehlivě určil svoji pozici vůči vytištěným tagům. Zde byla ověřena funkčnost algoritmů a kompatibilita u lokalizačního subsystému.



---

## Závěr

Na základě analýzy dostupných prostředků byly stanoveny požadavky na systém. Prostředí pro pohyb robotu bylo vytvořeno tak, aby odpovídalo návrhu prostor s fyzikálními vlastnosti stěn v prostředí V-Repu. Tím bylo docíleno reálného snímání různými typy senzorů. Také bylo vytvořeno digitální dvojče zvoleného robotu Odin. Tento systém byl osazen senzory, které umožňují detekci překážek a lokalizaci robotu. Pro detekci překážek byly zvoleny sonary a lidary. Pro lokalizaci byl zvolen systém podle referenčních značek. Tyto značky jsou snímány kamerami a pomocí rozpoznání obrazu následně zpracovány. Pro kamerový systém byly vytvořeny dvě varianty, které byly následně testovány v simulacích. Na základě těchto simulací pak byl vytvořen finální sensorický subsystém.

Dále byla vytvořena řídicí aplikace, která umožňuje zpracování obrazu a dat ze senzorů. Tyto data následně slouží k lokalizaci polohy na mapě. Lokalizace byla vyřešena pomocí rozpoznání obrazu a určení polohy tagu vůči kameře. Díky transformačním maticím je pak vypočítána poloha robotu vůči souřadnému systému mapy. Aby bylo možné určovat polohu na každém místě a orientaci robotu na mapě, vyplynulo ze simulace že, systém musí disponovat 4 kamerami, směřující na každou stranu robotu.

Navigace je realizována pomocí simulace ve V-Repu, která má zjednodušené tvary, aby výpočet byl co nejrychlejší. Tato simulace počítá s naměřenými daty ze senzorů, určené pozici robotu na mapě a cílovou pozicí robotu. Pomocí knihoven OMPL je naplánována globální trajektorie. Tato trajektorie vytvoří spojnicí mezi aktuální polohou robotu a požadovanou polohou tak, aby nedošlo ke kolizi se stěnami vyhrazeného prostoru. Následně je generována lokální trajektorie, která počítá s naměřenými daty ze senzorů a s body globální trajektorie. To umožňuje vyhýbat se překážkám a postupovat dále k požadované poloze, a přitom šetřit výpočetní výkon. Výsledkem této simulace je vektor pohybu robotu.

Pro komunikaci s řídicím systémem byl vytvořen server, který zajišťuje spojení operátora a řídicí aplikace. Tím je umožněno řídit robota, nebo mu posílat příkazy, které zpracuje a autonomně vyřeší.

Na závěr byl lokalizační subsystém otestován s reálným hardwarem. Testy prokázaly funkčnost lokalizačního systému.

---

## Seznam použité literatury

- [1] *Bullet Real-Time Physics Simulation* [online]. [cit. 2019-05-15]. Dostupné z: <https://pybullet.org/wordpress/>
- [2] *EmguCV* [online]. [cit. 2019-05-13]. Dostupné z: [http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page)
- [3] GALA, Matěj. *Konstrukční řešení a realizace podvozku se všesměrovými koly* [online]. Ostrava, 2014 [cit. 2019-05-09]. Dostupné z: <http://hdl.handle.net/10084/104778>. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.
- [4] In: *RobotShop* [online]. [cit. 2019-05-13]. Dostupné z: <https://www.robotshop.com/en/hokuyo-urg-04lx-ug01-scanning-laser-rangefinder.html>
- [5] In: *Snail Instruments* [online]. [cit. 2019-05-13]. Dostupné z: <http://www.snailshop.cz/srf/194-srf08-ultrazvukovy-dalkomer-i2c.html>
- [6] KÁRNÍK, Ladislav. *Praktické aplikace servisních robotů: studijní opora*. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, Fakulta strojní, 2011. ISBN isbn978-80-248-2727-8.
- [7] KÁRNÍK, Ladislav. *Využití servisních robotů v nestrojírenských aplikacích: studijní opora*. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, Fakulta strojní, 2011. ISBN 978-80-248-2728-5.
- [8] *LBKPIECE1 Class Reference* [online]. [cit. 2019-05-15]. Dostupné z: [http://ompl.kavrakilab.org/classompl\\_1\\_1geometric\\_1\\_1LBKPIECE1.html#gLBKPIECE1](http://ompl.kavrakilab.org/classompl_1_1geometric_1_1LBKPIECE1.html#gLBKPIECE1)
- [9] *Logitech* [online]. [cit. 2019-05-13]. Dostupné z: <https://www.logitech.com/cs-cz/product/c922-pro-stream-webcam>
- [10] *Newton Dynamics* [online]. [cit. 2019-05-15]. Dostupné z: <http://newtondynamics.com/forum/newton.php>
- [11] *Open Dynamics Engine* [online]. [cit. 2019-05-15]. Dostupné z: <http://www.ode.org>
- [12] *OpenCV* [online]. [cit. 2019-05-15]. Dostupné z: <https://opencv.org>
- [13] Path and motion planning. *Coppelia Robotics* [online]. [cit. 2019-05-14]. Dostupné z: <http://www.coppeliarobotics.com/helpFiles/en/pathAndMotionPlanningModules.htm>
- [14] Proximity sensor types and mode of operation. *Coppelia Robotics* [online]. [cit. 2019-05-14]. Dostupné z: <http://www.coppeliarobotics.com/helpFiles/en/proximitySensorDescription.htm>

- 
- [15] Remote API modus operandi. *Coppelia Robotics* [online]. [cit. 2019-05-14].  
Dostupné z:  
<http://www.coppeliarobotics.com/helpFiles/en/remoteApiModusOperandi.htm>
- [16] *RRTConnect Class Reference* [online]. [cit. 2019-05-15]. Dostupné z:  
[http://ompl.kavrakilab.org/classompl\\_1\\_1geometric\\_1\\_1RRTConnect.html#gRRTC](http://ompl.kavrakilab.org/classompl_1_1geometric_1_1RRTConnect.html#gRRTC)
- [17] *SST Class Reference* [online]. [cit. 2019-05-15]. Dostupné z:  
[http://ompl.kavrakilab.org/classompl\\_1\\_1geometric\\_1\\_1SST.html#gSST](http://ompl.kavrakilab.org/classompl_1_1geometric_1_1SST.html#gSST)
- [18] *The Open Motion Planning Library* [online]. [cit. 2019-05-15]. Dostupné z:  
<http://ompl.kavrakilab.org>
- [19] Vision sensor types and mode of operation. *Coppelia Robotics* [online]. [cit. 2019-05-14]. Dostupné z:  
<http://www.coppeliarobotics.com/helpFiles/en/visionSensorDescription.htm>
- [20] *Vortex Dynamics* [online]. [cit. 2019-05-15]. Dostupné z: <https://www.cm-labs.com/wp-signup.php?new=www.vxsim.com>
- [21] *V-REP User Manual* [online]. Version 3.6.1. [cit. 2019-05-13]. Dostupné z:  
<http://www.coppeliarobotics.com/helpFiles/>

## **Přílohy**

Příloha A: CD médium

S Obsahem:

- diplomová práce v elektronické podobě
- simulační scény
- zdrojové kódy aplikace