



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

# NÁVRH A REALIZACE INTERAKTIVNÍHO GRAFICKÉHO ROZHRANÍ PRO KOMUNIKACI S ROBOTICKÝM KONVOJEM

GRAPHICS COMMUNICATION TOOL DESIGN SUITABLE FOR ROBOTICS CONVOY COMMANDS AND  
MONITORING

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

## AUTOR PRÁCE

AUTHOR

Tomáš Kruták

## VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Stanislav Věchet, Ph.D.

BRNO 2017



# Zadání bakalářské práce

Ústav:	Ústav automatizace a informatiky
Student:	<b>Tomáš Kruták</b>
Studijní program:	Strojírenství
Studijní obor:	Aplikovaná informatika a řízení
Vedoucí práce:	<b>doc. Ing. Stanislav Věchet, Ph.D.</b>
Akademický rok:	2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## **Návrh a realizace interaktivního grafického rozhraní pro komunikaci s robotickým konvojem**

### **Stručná charakteristika problematiky úkolu:**

Navrhněte grafické komunikační rozhraní pro monitorování a obousměrnou komunikaci s konvojem autonomních vozidel. Prostředí musí být vhodné pro integraci ve frameworku ROS.

### **Cíle bakalářské práce:**

- 1) Seznamte se s aktuálním stavem frameworku ROS.
- 2) Prostudujte možnosti zobrazovacího modulu RViz.
- 3) Navrhněte GUI v prostředí RViz vhodné pro autonomní vozidla v konvoji.
- 4) Navržený systém otestujte.

### **Seznam doporučené literatury:**

THRUN Sebastian, BURGARD Wolfram, and FOX Dieter. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series). Intelligent robotics and autonomous agents. The MIT Press, August 2005.

CHOSSET Howie, LYNCH Kevin M., HUTCHINSON Seth, KANTOR George, BURGARD Wolfram, KAVRAKI Lydia E., and THRUN Sebastian. Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents). The MIT Press, June 2005.

ROS.org. ROS.org | Powering the world's robots. [online]. 7.11.2016 [cit. 2016-11-07]. Dostupné z: <http://www.ros.org/>.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

---

doc. Ing. Radomil Matoušek, Ph.D.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Náplní této bakalářské práce je zjištění aktuálního stavu frameworku s názvem ROS (Robot Operating System) a jeho nástroje RViz. Dále vytvoření uživatelského grafického rozhraní pro autonomní konvoj, které bude následně otestováno v praxi. Tato práce je dílčí částí širšího projektu pěti bakalářských prací, který se postupně věnuje různým tématům vývoje autonomního konvoje.

## **ABSTRACT**

The topic of this bachelor thesis is to define actual condition of Framework ROS (Robot Operating System) and its tool RViz. Another topic is developing graphical user interface for autonomous convoy which will be tested in practice. This thesis is part of the wider complex of bachelor theses dealing with different topics of autonomous convoy development.

## **KLÍČOVÁ SLOVA**

Linux, ROS, RViz, GUI pro autonomní konvoj

## **KEYWORDS**

Linux, ROS, RViz, GUI for autonomous convoy



## **BIBLIOGRAFICKÁ CITACE**

KRUTÁK, T. *Návrh a realizace interaktivního grafického rozhraní pro komunikaci s robotickým konvojem*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 37 s. Vedoucí bakalářské práce doc. Ing. Stanislav Věchet, Ph.D..





## **PODĚKOVÁNÍ**

Na tomto místě bych rád vyjádřil poděkování vedoucímu doc. Ing. Stanislavu Věchetovi, Ph.D. za pomoc a věnovaný čas při tvorbě této bakalářské práce. Dále bych chtěl poděkovat celé rodině a blízkému okolí, které mi svou ochotou a trpělivostí vytvořili ideální podmínky pro dokončení práce.



## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením doc. Ing. Stanislava Věchety, Ph.D. a s použitím literatury uvedené v seznamu literatury.

V Brně dne 23. 5. 2017

.....

Tomáš Kruták



## OBSAH

<b>1</b>	<b>ÚVOD.....</b>	<b>15</b>
<b>2</b>	<b>LINUX.....</b>	<b>17</b>
2.1	Ubuntu 16.04 Xenial Xerus.....	17
2.2	Virtual disk .....	17
2.3	Dual boot .....	18
<b>3</b>	<b>ROS .....</b>	<b>21</b>
3.1	Distribuce ROS .....	21
3.2	Instalace ROSu Kinetic .....	22
3.3	Vytvoření pracovního prostředí Catkin .....	24
3.4	Ovládání a orientace v ROS .....	25
<b>4</b>	<b>RVIZ .....</b>	<b>27</b>
4.1	Instalace RViz.....	27
4.2	Seznámení s prostředím .....	27
<b>5</b>	<b>VLASTNÍ ŘEŠENÍ.....</b>	<b>29</b>
5.1	Řízení vodícího auta .....	29
5.2	Broadcaster a Listener .....	30
5.3	Kamera.....	31
<b>6</b>	<b>ZHODNOCENÍ A DISKUZE.....</b>	<b>33</b>
<b>7</b>	<b>ZÁVĚR .....</b>	<b>35</b>
<b>8</b>	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>37</b>



# 1 ÚVOD

Autonomní konvoj je v dnešní době hlavní téma na poli robotiky a umělé inteligence. Velké nadnárodní společnosti, jako je Volvo, Audi či TESLA společně soutěží, kdo navrhne nejspolehlivější systém, který by bylo možno aplikovat do běžného života veřejnosti. Problematika autonomního konvoje je rozsáhlá. Dnes asi nejvíce diskutované oblasti jsou převážně jeho řízení a bezpečnost provozu.

Tato práce je dílčí částí většího projektu, který se snaží vytvořit komplexní systém autonomního konvoje modelu aut na ovládání v měřítku 1:12. Jsou zde postupně řešena témata jako výběr správného hardwaru a jeho následné sestavení, zpracování obrazu a sledování konvoje, jednotlivé řízení každého modelu, jejich komunikace, až ke grafickému výstupu, o kterém pojednává tato bakalářská práce.

Jednou ze základních úvah bylo, na jakém hardwaru a jakých platformách bude systém vytaven. Jako nejrozmumnější variantou se jevílo použít robotický operační systém ROS, který je v dnešní době jedním z nejkvalitnějších a nejuniverzálnějších frameworků pro práci a vývoj robotických aplikací. Vybrána byla nejnovější verze ROS Kinetic, která sice neměla v době vývoje stoprocentní podporu všech knihoven, ale pro tento projekt to byla dostačující volba.

Dalším důležitým rozhodnutím byl výběr správného hardwaru. Zde bylo zvoleno využití jednočipového počítače Raspberry PI 3, který disponuje ARMovým procesorem. Obsahuje také důležité výstupy, jako port USB 2.0, 10/100Mbit Ethernet, HDMI port a WiFi vysílač. Tento jednodeskový počítač byl přilepen k modelu auta, které bylo součástí vybavení laboratoře Ústavu automatizace. Pro vizuální kamerovou komunikaci a následné zpracování obrazu byla zvolena cenově dostupná, přesto kvalitativně dostačující, kamera Microsoft LifeCam HD-3000, která byla rovněž přilepena na model, konkrétně na jeho přední část. Snímaný objekt, sledovaný kamerou, byla výroba jednoho z členů zapojených do projektu, a to papírového kola s konkrétním matematickým symbolem, vhodným pro danou charakteristiku snímání obrazu. Vodící předmět byl samostatná jednotka vedená člověkem. Další jednotky lze umístit na zadní část aut, díky čemuž je možné teoreticky vytvořit nekonečnou řadu autonomního konvoje.

Na základě zmíněného výběru bylo nutné zvolit software. Klíčová byla především podpora a kompatibilita se všemi komponenty, a zároveň podpora frameworku ROS. Byla tedy zvolena linuxová distribuce zastřešená Britskou společností Canonical Ltd, a to Ubuntu 16.04.

Rešeršní část bakalářské práce se věnuje zprovoznění a následného využívání softwaru. Pojednává o možnostech instalace a využití Linuxového operačního systému Ubuntu 16.04 a přiblíží práci v robotickém frameworku ROS, zejména jednoho konkrétního modulu.

Tímto modulem je nástroj zvaný RViz, využívaný pro detailní grafický výstup a jeho obrazovou komunikaci s robotem, zde s robotickým konvojem. Jeho konkrétní využití bude obsahovat kamerové zobrazení a také sledování dráhy konvoje.

Tato část se může v dané problematice jevit jako zbytečná, avšak grafické zobrazování má velký vliv na celkový vývoj a provoz autonomního konvoje. V první řadě je využíváno pro simulování fungujícího systému v praxi. Zde je zahrnuto jak snímání vedoucího vozidla, tedy přímý výstup z kamery, tak také snímání, simulace dráhy a následná kontrola s dráhou reálnou. Grafické zobrazování může také snímat široké okolí a na základě matematických výpočtů určit vhodnou následující trasu. V praxi to může představovat i jediný zdroj zjištění správnosti chování konvoje. Zejména v nebezpečných oblastech, jako válečná zóna, důlní prostředí, nebo pro lidstvo zatím nedostupná zóna při kolonizaci vesmíru.

V této práci bude kladen důraz na kamerové snímání a také na sledování konvoje. To znamená, že grafický výstup bude zobrazovat vodící kruh, respektive první auto konvoje, následované vozidlem v závěsu, které kopíruje jeho přesnou dráhu. Součástí bude také výstup z kamery, zobrazující pohled modelu z první osoby.

Závěrem bude snaha o propojení všech dílčích částí projektů, testovaných v laboratorních podmínkách. Poté proběhne analýza a následné hodnocení.



## 2 LINUX

Stejně jako Windows 7, 8, 10 a MAC OS X, je Linux operační systém. Je to software, který spravuje veškeré hardwarové prostředky spojené se stolním počítačem nebo notebookem. Jednoduše řečeno – řídí veškerou komunikaci mezi nimi. Na vývoji tohoto operačního systému se podílí množství lidí z celého světa, převážně ve svém volném čase, a dávají ho k dispozici na internet v domnění, že bude širokou komunitou dále vylepšován. Díky tomu je veškerá distribuce zdarma, což je jeho hlavní výhoda. Linuxových distribucí existuje široká řada, a proto je důležité, se před instalací důkladně rozmyslet, k čemu vlastně bude využíván. [1]

### 2.1 Ubuntu 16.04 Xenial Xerus

Ubuntu je komunitou vyvíjený operační systém, který zastřešuje britská společnost Canonical. Je vytvořen na Linuxovém jádru a je od základu navržen s ohledem na bezpečnost. Díky tomu se minimalizuje riziko zavirování či jiné narušení běhu počítače. V Ubuntu se nachází kompletní pracovní prostředí s balíkem základních programů, které mohou být rozšířeny o tisíce dalších. [2]

### 2.2 Virtual disk

Virtual disk je jedna ze dvou základních možností, jak nainstalovat Linux. Základní princip je v simulaci hardwarového prostředí. To umožní VirtualBox. „*Je to virtuální stroj, který umožňuje hostitelskému systému (host) simulovat hardwarové prostředí, takže host systém (guest) se domnívá, že pracuje se skutečným hardwarem. Příkladem virtuálního stroje může být, krom VirtualBoxu, například Java Virtual Machine.*“ [3] Ještě před samotným vytvořením virtuálního počítače je nutno vědět, jakou distribuci Linuxu je třeba nainstalovat, a to z důvodu správného fungování. S ohledem na další práci, především ve frameworku ROS, bylo vybráno Ubuntu 16.04. Nezáleží na tom, je-li zvolena 32 nebo 64 bitová verze, protože VirtualBox dokáže pracovat s oběma. Doporučuje se však verze shodná s architekturou procesoru. Následná instalace virtuálního počítače pobíhá intuitivně, ale je nutno důkladně promyslet jeho parametry. Při nastavování operační paměti je třeba dbát na to, aby pro virtuální jednotku nebyla přidělena příliš velká část. Je totiž možné, že výpočetní výkon pro VirtualBox pak nebude dostatečný, což zapříčiní nerozběhnutí virtuální jednotky. Doporučuje se nejméně 768 MB, ale optimální je 1024 až 1536 MB z 4096 MB.

Další věcí ke zvážení je, kolik bude přiděleno místa na fyzickém pevném disku. První možností je dynamické alokování. Tedy paměť se bude dynamicky přizpůsobovat potřebě. Druhou možností je pevná velikost. Doporučuje se minimálně 16 GB. Samotné Ubuntu zabere podle verze velikost přibližně do 7 GB. Doporučuje se také přidělit velikost grafické paměti, alespoň 50 MB, pro plynulejší chod.

Posledním nutným nastavením je pořadí bootování. Zde se obvykle zvolí na první místo optické zařízení. Tímto je instalace virtuálního počítače hotová a lze přejít k instalaci Ubuntu. [4]

## 2.3 Dual boot

Druhá základní varianta, jak nainstalovat Ubuntu, je jeho paralelní instalace k Windows. Znamená to, že na jednom počítači může být nainstalováno více operačních systémů a teprve při zapnutí si uživatel vybere, který bude zrovna používat. Předpoklady instalace jsou následující. Je třeba jej mít uloženo na zvoleném médiu, například CD disk nebo USB flash disk (dále jen USB), a disponovat dostatečným výkonem počítače. Nejnovější distribuce vyžadují nejméně 1 GB RAM, 1 GHz CPU a 5 GB místa na disku. Architektura procesoru může být jak 32, tak 64 bitová. Samotná instalace probíhá následovně. [5]

### Krok 1 - Spuštění Live CD.

Do mechaniky se vloží instalační CD, potažmo USB, a restartuje se počítač. Po načtení se zobrazí prázdná fialová plocha se dvěma ikonami v dolní části obrazovky – Nainstalovat Ubuntu a Vyzkoušet Ubuntu. [5]

### Krok 2 - Zahájení instalace

Po načtení instalátoru se z nabídky vybere požadovaný jazyk a pokračuje se volbou Nainstalovat Ubuntu. Varianta Vyzkoušet Ubuntu slouží například k prohlédnutí systému nebo také pro zprovoznění internetu se statickou IP adresou. [5]

### Krok 3 - Příprava k instalaci

Instalátor ověří, jestli počítač splňuje požadované minimální parametry. To se týká zejména velikosti úložného prostoru, což je nutná podmínka. Dále zjišťuje připojení internetu. To není nutně vyžadováno, ale doporučuje se, především proto, že pro systém jsou neustále vydávány aktualizace, které se ale neprojevují na dostupných instalačních obrazech. Je tedy vhodné, nechat instalátor, společně s Ubuntu, nainstalovat i aktualizace. Pokud připojení k internetu není k dispozici, lze aktualizace nainstalovat i později manuálně. Celkovou instalaci je možné také obohatit o instalaci kodeků pro nesvobodné formáty typu MP3 a jiné, či nesvobodné softwary třetích stran. Jedná se o podporu technologie Flash a zásuvný modul Java. [5]

### Krok 4 - Volba disku pro instalaci

Tato část je velice důležitá, proto je třeba jí věnovat zvýšenou pozornost. Při špatném nastavení může totiž uživatel přijít i o veškerá vlastní data. [5]

Při instalaci Ubuntu vedle jiného operačního systému, zpravidla Windows, existuje možnost zachování veškerých dat, kdy bude odňata pouze prázdná paměť, ta

bude následně přidělena Ubuntu. Poté je při každém startu počítače nabídnuta možnost výběru operačního systému. [5]

Nahradit Windows distribucí Ubuntu je variantou, při které se využije veškeré dostupné místo na disku, a tím pádem jsou smazána veškerá existující data.

Poslední možnost *Něco jiného* představuje ruční nastavení instalace. Pro tuto variantu je třeba mít jisté znalosti a zkušenosti, a proto se doporučuje použít některou ze dvou výše zmíněných variant. [5]

#### *Varianta rozdělení disků – dual boot*

Při volbě této varianty a stisku tlačítka vpřed se objeví okno, kde je možné vybrat na jakém disku, pokud jich existuje víc, bude provedena změna. Je zde také přehledně zobrazeno, v jakém poměru se disk rozdělí. S posuvníkem, reprezentujícím sloupek mezi oběma operačními systémy, lze pohybovat, a tím pádem upravit přidělenou velikost. Po stisku tlačítka *Instalovat nyní* začne instalátor se zmenšováním vybraného oddílu a bude vytvářet místo pro Ubuntu. [5]

#### *Varianta použití celého disku*

Tento způsob odstraní veškerá data, která se na disku nachází. Po stisku tlačítka vpřed se v dalším okně vybere disk, je-li jich více, který bude pro instalaci použit. Po odsouhlasení toho kroku stisknutím tlačítka *Instalovat nyní* začne okamžitě samotná instalace. Tento krok je nevratný! Je proto vhodné ještě opětovně přezkontrolovat, jestli je zvolen správný disk a jestli budou skutečně smazána jen ta data, která smazána být mají. [5]

#### Krok 5 - Nastavení času

Než se systém, běžící na pozadí, nainstaluje, musí být provedeno několik posledních nastavení. První v řadě je nastavení časové zóny. Instalátor zpravidla pozná, v jaké časové zóně se zařízení nachází, a vybere ji sám. Pokud se tak nestane, zadá se časové pásmo manuálně, napsáním města do zobrazeného pole. Instalátor by měl bez problémů rozeznat všechna větší česká města. [5]

#### Krok 6 - Nastavení rozložení klávesnice

Tento krok umožňuje vybrat výchozí rozložení klávesnice. Dle volby jazyka vybraného na začátku instalace, je zde automaticky zvolen také typ klávesnice. To znamená, že když byla vybrána čeština, tak typ klávesnice bude Česko, což je standardní česká klávesnice QWERTZ. Pokud automatická volba není vyhovující, není problém vybrat jakoukoliv jinou, například QWERTY. Správnost vybraného rozložení lze ověřit v testovacím poli v dolní části okna. [5]

#### Krok 7 - vytvoření uživatele

Posledním krokem instalace je vytvoření uživatele. Vyplní se zde jméno, název počítače, dále uživatelské jméno a heslo. Heslo je potřeba si zapamatovat, protože

slouží jak k přihlášení uživatele, tak je vyžadováno při správě systému. Název počítače je důležitý, pokud je připojen ve vnitřní síti s více počítači. Pole může být ponecháno prázdné, případně se dá později změnit. Na závěr existuje ještě možnost nastavit automatické přihlašování nebo variantu, která vyžaduje pro přihlášení heslo. Pro úplné dokončení instalace je potřeba počítač ještě restartovat, poté se už nově nainstalovaný systém spustí. [5]



## 3 ROS

Robotický operační systém (ROS) je flexibilní aplikační rámec neboli framework, sloužící pro psaní robotického softwaru. Je to sbírka nástrojů, knihoven a konvencí, které mají za cíl zjednodušit vytváření komplexního a robustního chování robota skrze širokou škálu robotických platform. Vytvářet skutečně robustní a všeobecně použitelný robotický software je obtížné. Je to dáno tím, že problémy, které se zdají být z pohledu člověka banální, jsou z perspektivy robota komplikované. Proto žádný jednotlivec, laboratoř nebo instituce nemůže doufat, že se do vývoje robota pustí na vlastní pěst.

Výsledkem je ROS, jenž byl od základu postaven pro podporu a spolupráci vývojářů robotického softwaru. Například, jedna laboratoř může mít odborníky na zmapování vnitřního prostředí a může také přispět prvotřídním systémem pro výrobu map. Jiná skupina může mít odborníky na využití map pro navigaci, a další skupina by mohla zjistit přístup počítačového vidění, který dobře pracuje pro rozpoznávání malých objektů v pohybu. ROS byl navržen speciálně pro tyto skupiny, aby spolupracovaly a stavěly na pracích jiných vývojářů, jak je zde popsáno. [6]

### 3.1 Distribuce ROS

Díky mnohaletému vývoji se ROS vyskytuje v několika distribucích. Aktuálně je dostupných deset verzí. Ne všechny však mají aktivní vývojářskou podporu, konkrétně Indigo, Jade a Kinetic. Na květen 2019 je ohlášena zbrusu nová verze Lunar. Je důležité zmínit, že různé distribuce nejsou mezi sebou zcela kompatibilní a každá z nich má jinou podporu pro operační systémy a roboty. Proto se ne vždy v praxi volí nejnovější doporučená verze, jakou je dnes Kinetic, ale nějaká starší verze, například Indigo. [7]

Distro	Release date	Poster	Turtle, turtle in tutorial	EOL date
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame (Recommended)	May 23rd, 2016			May, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)

Obr. 1: Přehled distribucí ROS [7]

### 3.2 Instalace ROSu Kinetic

Instalace se bude provádět pomocí konzolových příkazů v Terminálu, jak je v Ubuntu většinou zvykem. Jde i přímo stáhnout složku s obsahem a instalovat manuálně jednotlivé části, ale je to zdoluhavý a komplikovanější postup. V první řadě je nutné v nastavení Ubuntu povolit některé z možností stahování software z internetu, aby instalace proběhla bez problémů. Konkrétně se musí povolit možnosti restricted, universe a multiverse.

Pokračování probíhá už výhradně v Terminálu. Zde nejprve povolíme počítači přijímat software z packages.ros.org. Ros Kinetic pracuje a je kompatibilní pouze s Wily (Ubuntu 15.10), Xenial (Ubuntu 16.04) a Jessie (Debian 8). Příkaz pro spojení s balíčky vypadá následovně: [8]

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

Hned na to se provede konfigurace kláves.

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAE801FA116
```

Ještě se nutno ujistit, že balíčky jsou aktualizované, případně provést aktualizaci.

```
sudo apt-get update
```

Jelikož ROS obsahuje rozsáhlé množství knihoven a nástrojů, je ve výběru více možností, které se dají stáhnout dle žádaného obsahu. Doporučuje se varianta Desktop-Full. Ta obsahuje prakticky vše potřebné, tedy ROS, rqt, rviz, robot-generic libraries, 2D/3D simulators, navigation and 2D/3D perception.

```
sudo apt-get install ros-kinetic-desktop-full
```

Ne kompletní, ale také poměrně obsáhlá, je verze Desktop. Ta se skládá z ROS, rqt, rviz, and robot-generic libraries.

```
sudo apt-get install ros-kinetic-desktop
```

Třetí možností je ROS-Base. Jedná se o základní verzi, kde je pouze ROS package, build, and communication libraries. Neobsahuje například žádný z GUI nástrojů.

```
sudo apt-get install ros-kinetic-ros-base
```

Jako poslední varianta je možnost stáhnout konkrétní balíček. Provede se to příkazem,

```
sudo apt-get install ros-kinetic-PACKAGE
```

Slovo **PACKAGE** zde bude nahrazeno jménem balíčku, který se plánuje stáhnout. Pro vyhledávání dostupných balíčků slouží příkaz:

```
apt-cache search ros-kinetic
```

Před použitím ROSu je důležité inicializovat rosdep. Rosdep umožňuje instalovat systémové závislosti pro zdroj, který si žádá zkompileovat. Je taktéž velice důležitý pro správný chod některých komponent jádra ROS.

```
sudo rosdep init
rosdep update
```

Další část instalace je nastavení pracovního prostředí. Je to výhodné, když jsou proměnné prostředí ROS automaticky přidány do bash relace pokaždé, když se spustí nový shell.

```
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Pokud je potřeba změnit prostředí pouze aktuálního shellu, lze napsat:

```
source /opt/ros/kinetic/setup.bash
```

Toto byl finální krok, zde instalace končí. Další krok je nastavení pracovní prostředí catkin workspace. [8]

### 3.3 Vytvoření pracovního prostředí Catkin

Pro vytváření vlastních aplikací ve frameworku ROS, je v první řadě potřeba vytvořit uživatelské pracovní prostředí. Od verze Groovy a výš, je Catkin součástí ROSu, pouze se zpřístupní příkazem:

```
sudo apt-get install ros-kinetic-catkin
```

Nyní již k samotnému nastavení prostředí. V první řadě vytvoříme složku `catkin_ws` a přejdeme do ní.

```
$ mkdir -p ~/catkin_ws/src  
$ cd ~/catkin_ws/src
```

I když je pracovní prostor prázdný (v adresáři 'src' nejsou žádné balíčky, pouze jediný odkaz `CMakeLists.txt`), může se stále vytvářet pracovní prostor:

```
$ cd ~/catkin_ws/  
$ catkin_make
```

Příkaz `catkin_make` je pohodlný nástroj pro práci s pracovními prostory typu `catkin`. V aktuálním adresáři, by měla být složka 'build' a 'devel'. Ve složce "devel" je vidno, že existuje nyní několik souborů `*.sh`. Před pokračováním se nechá rozběhnout nový `setup.*sh` soubor.

```
$ source devel/setup.bash
```

Pokud má být pracovní prostor správně překryt pomocí instalačního scriptu, musí být zajištěno, že proměnná prostředí `ROS_PACKAGE_PATH` obsahuje adresář, ve kterém se nacházíte. Funguje-li vše, je instalace úspěšně hotová. [9]

```
$ echo $ROS_PACKAGE_PATH  
/home/youruser/catkin_ws/src:/opt/ros/kinetic/share
```



### 3.4 Ovládání a orientace v ROS

K ovládání systému se používají primárně takzvané *konzolové nástroje*. Lze také sekundárně použít možnost grafického rozhraní RQT, ale tomu zde nebude věnována pozornost.

Konzolové nástroje se spouštějí pomocí příkazů, které jsou zadávány do příkazového řádku přímo v operačním systému. Součástí příkazu je název nástroje a argument. Oba definují činnost nástroje. Každý nástroj, má jasně daný počet argumentů. Je výhodné používat tabulátor, pro automatické doplnění textu. Je to dobrá nápověda při hledání možností nástroje, pomáhá také eliminovat chyby a urychluje práci. [10]

Název nástroje	Popis funkce
roscore	Spuštění jádra ROS.
roslaunch	Spouštění mnoha uzlů zároveň (lokálně, vzdáleně), správa parametr serveru, nastavení systémových proměnných ROS.
roslaunch	Spouštění uzlů jednotlivě a lokálně.
rostopic	Diagnostika témat.
roscd	Diagnostika uzlů.
rosmmsg, rossrv, rosservice	Diagnostika standartních a služebních zpráv.
rosllocate	Lokalizace balíčků.
roscd	Pohyb mezi balíčky.
roscparam	Správa parametr serveru.
roscswtf	Ladění systému ROS.
roscslog	Logování dat.

Obr. 2: Nejvíce používané konzolové nástroje [10]



## 4 RVIZ

Při vývoji robota, nebo jeho ovládání je velice vhodné mít zobrazený grafický výstup. Pro tyto případy existuje RViz. Je to 3D vizualizační nástroj, který zobrazuje data ze snímačů robota a informace o stavu z ROSu.

### 4.1 Instalace RViz

Protože RViz už je součástí ROSu, je instalace triviální záležitostí a provede se příkazem:

```
sudo apt-get install ros-kinetic-rviz
```

Dále se stáhnou zdroje do pracovního prostředí.

```
rosdep install rviz
```

A vytvoří se vizualizátor.

```
rosmake rviz
```

Zde instalace končí. [11]

### 4.2 Seznámení s prostředím

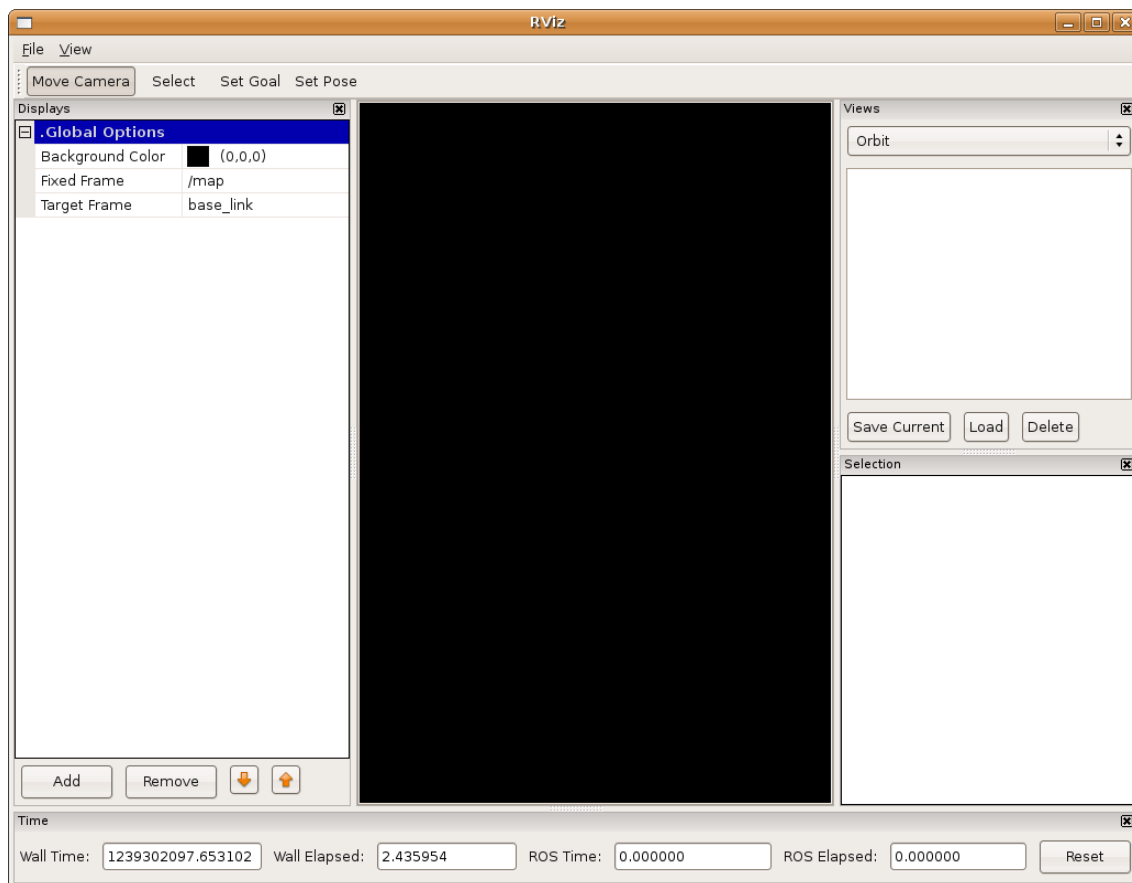
Aby bylo možné RViz spustit, musí být v první řadě spuštěn ROS. Pokud není, zadá se do konzole, samozřejmě s upravenou cestou, toto:

```
source /opt/ros/kinetic/setup.bash  
roscore &
```

Následné otevření RVizu se provede příkazem:

```
roslun rviz rviz
```

Při prvním spuštění RVizu se objeví prázdné ovládací okno.

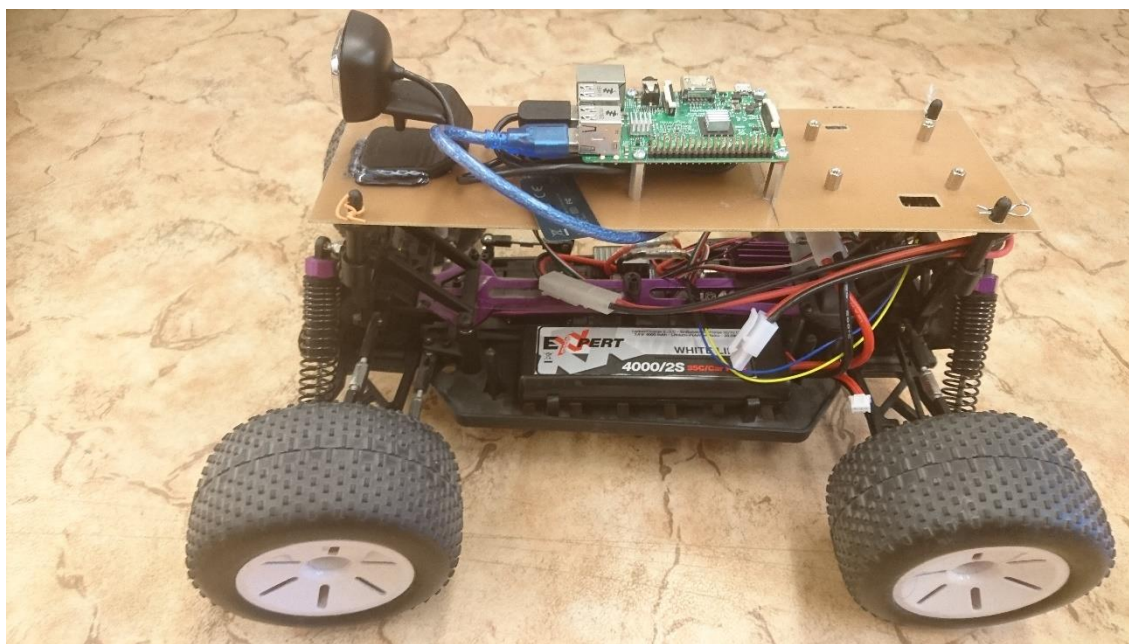


Obr. 3: RViz při prvním spuštění [11]

Velké černé okno uprostřed znázorňuje 3D pohled. Je prázdné, protože nic není připojeno, a tedy se nemá co zobrazit. Levý sloupec znázorňuje seznam zobrazení, který ukáže všechny načtené obrazovky. V tuto chvíli obsahuje pouze globální možnosti. Pravá strana slouží pro zobrazování různých typů kamer. Spodní panel zobrazuje časový přehled, ten je velice důležitý, zejména pokud se pracuje s robotem v reálném čase. [11]

## 5 VLASTNÍ ŘEŠENÍ

Cílem práce je vytvořit GUI pro komunikaci s robotickým konvojem. Úvaha je taková, že se v první řadě vytvoří snímání vodícího auta. Poté bude následovat vytvoření programu, který bude mít za úkol sledovat přesnou dráhu prvního auta. Kamerový výstup má pouze doplňující funkci, a to vizuální kontroly pohybu objektu.



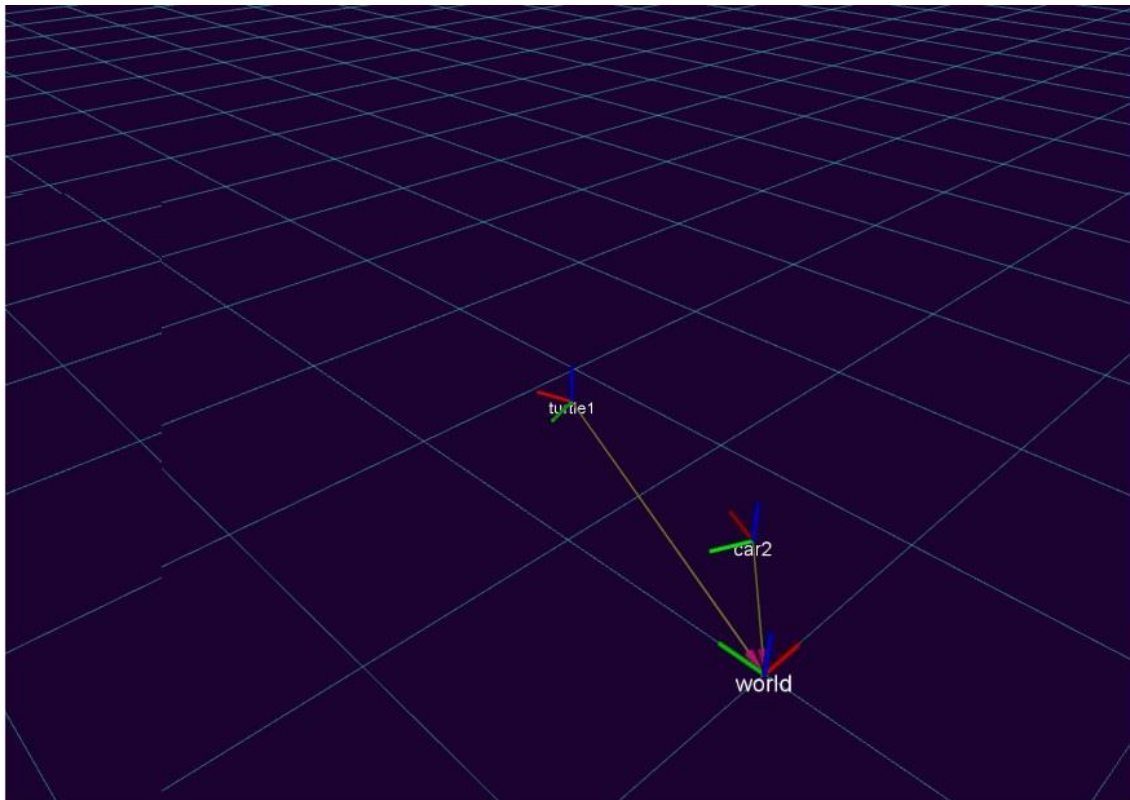
Obr. 4: Model autonomního auta v měřítku 1:12

### 5.1 Řízení vodícího auta

Protože celkový projekt autonomního konvoje má z důvodu omezeného rozpočtu hardware pouze pro jeden model – auto sledující vodící objekt, bylo potřeba vytvořit simulované řízení vedoucího vozidla. Pro tento účel byla ideální volba použít již existující model z kategorie Turtlesim z knihovny. Je to program v databázi ROSu, který umožňuje vytvoření jednoho pohybujícího se objektu. Pokud by bylo fyzicky k dispozici více autonomních modelů, toto první auto by se nemuselo ovládat z počítače, ale přijímalo by WiFi signál o poloze, vysílaný z Raspberry PI. Obsahoval by informace o souřadnicích X, Y a o úhlu natočení. Ty jsou vypočítány pomocí softwaru, který vytvořil jeden z členů projektu. Jde o snímání obrazu pomocí kamery s frekvencí 25 FPS a jeho následné zpracování na potřebné informace. Zde ovšem tato možnost není, a tak se řízení simuluje pomocí počítače. Pro tento úkon je klíčová knihovna teleop-twist-keyboard. Díky ní je možné vytvořený objekt ovládat šipkami na klávesnici.

## 5.2 Broadcaster a Listener

Druhá část vlastního programu obsahuje vytvoření konvoje, respektive objektů, kopírujících dráhu toho vodičího. Jde o vytvoření dvou navzájem komunikujících programů v jazyce PYTHON. První je takzvaný broadcaster, neboli vysílač. Ten snímá aktuální polohu prvního vozidla, konkrétně, jak bylo uvedeno v kapitole 5.1, souřadnici X, Y a úhel. Tyto informace zaznamená, uloží a vysílá je. Poté přijde na řadu listener, jinak řečeno přijímač. Jeho úkolem je spárovat se s vozidlem v závěsu a zpracovat data, jež vysílá broadcaster. Program vyhodnotí informace o poloze, ve které se nachází vedoucí vůz, a navede na ni další vozidlo konvoje. Protože vysílač snímá data v reálném čase zhruba po desetinách sekundy, je chod plynulý a téměř totožný s vozem ve vedení. Lze zde nastavit také časovou prodlevu, ta ale není vhodná, protože v zatáčkách nedochází k přesnému kopírování trasy, tím pádem se stává, že vozidlo v závěsu narazí do překážky.



Obr. 5: Sledování konvoje v RViz

Na obrázku můžeme vidět 3D model, který zobrazuje pohyb dvou objektů. Pro jejich grafickou reprezentaci byla zvolena barevná osa, kde červená linka ukazuje přímý směr auta. Zelená potom zobrazuje jeho levou stranu. První objekt s názvem turtle1 je simulací vedoucího vozidla. Car2 je vozidlo v konvoji. Objekt pod názvem world je výchozí poloha. World je také s každým jedním autem spojen přímkou, udávající jeho vzdálenost od začátku.

### 5.3 Kamera

Přesto, že primární funkce kamery je sledování předchozího auta a zpracování obrazu, může se také použít jako kontrolní výstup pro obsluhu. Umístění na přední části kapoty umožňuje pohled z první osoby. Lze proto vizuálně kontrolovat, zda je konvoj v řadě, nebo sledovat případné překážky. Pro přidání kamery do RVizu, a pro její správný chod slouží knihovna **usb-cam**. Má v sobě několik modů snímání obrazu, hlavně za použití různých barevných spekter, zde ale stačí pouze standardní výstup.



Obr. 6: Kamerový výstup, sledování vodícího vozu



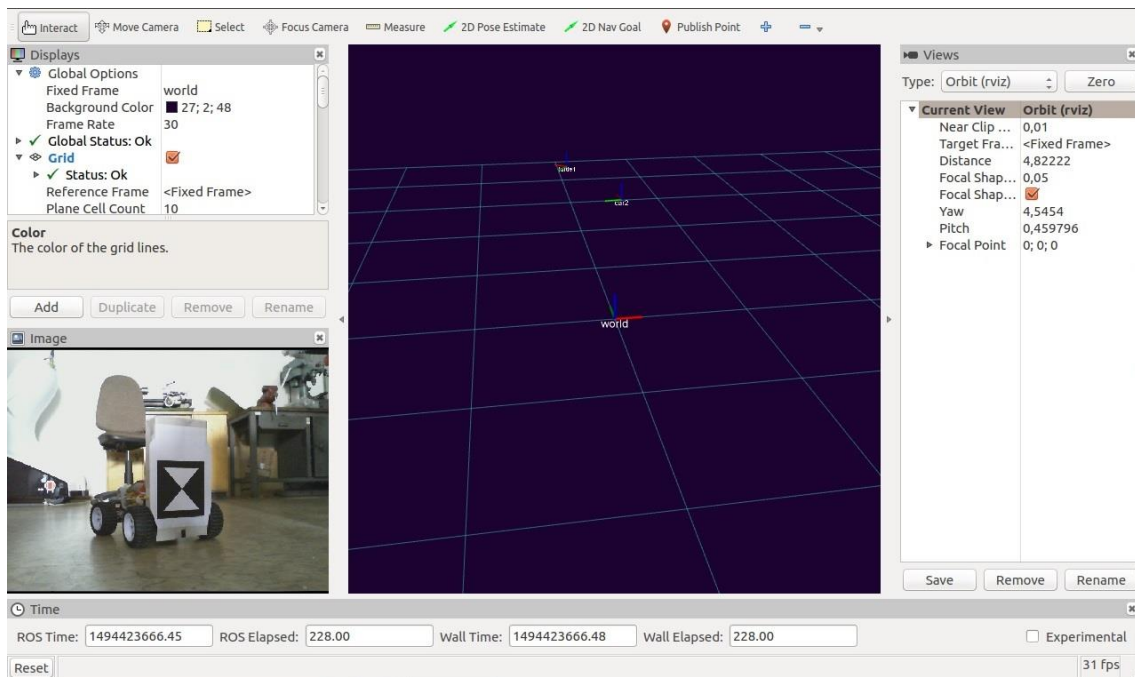


## 6 ZHODNOCENÍ A DISKUZE

Poslední částí celého projektu byla snaha o to, aby se propojily dílčí části všech bakalářských prací s ním spojených. V první řadě se zkoušela funkčnost hardwaru a ovládání auta. To fungovalo velice uspokojivě. V dalším testovacím kroku bylo ověřeno, zda zpracování obrazu plní svoji funkci, což bylo snímání vodícího objektu. Funkce zpracování obrazu byla téměř dokonalá. Problémy však nastaly, když na krátký moment vypadla kamera. Model auta ujel z kurzu a už se nedokázal přizpůsobit dráze vedoucího objektu. Tento problém měl být řešen softwarově, a to způsobem, že auto mělo předpovídat dráhu, kterou se má při výpadku vydat. Bohužel náročnost úkolu a nedostatek času zapříčinil, že se daná úloha nepodařila dokončit.

Jako další vhodné řešení by mohla být synchronizace se zde navrženým grafickým výstupem. Mohlo by to fungovat tím způsobem, že sestava broadcasteru a listeneru, která si umí s kopírováním přesné pomyslné dráhy poradit, by vysílala data o teoretické poloze, a při ztrátě obrazu by měla prioritu, kterou se model bude řídit. Představovalo by to i dobrou kontrolu při chodu. Tento úkol je však na rámec této práce jak obsahově, tak náročností. Problém by nastal také v tom, že je hardware pouze pro jeden model auta, a pro správný chod je potřeba dvou jednočipových počítačů, pro každý model jeden, aby spolu mohly komunikovat.

Omezení, která nastávají při chodu v praxi, jsou i rychlost při pohybování konvoje. Přesněji řečeno, kdyby se konvoj pohyboval příliš rychle, kamerový software by nestíhal zpracovávat informace o vodícím objektu a došlo by k rozdělení. Vytvořené GUI, zobrazené na obr. 7 je, co se týče funkčnosti, dostačující. Dají se zde však do budoucna udělat určitá vylepšení. Místo osového vyobrazení modelů aut lze vymodelovat reálné modely. Mohlo by se také naprogramovat snímání okolí, které by bylo vykreslováno do 3D vizualizace, která by se poté dala analyzovat. Ve výsledku by se pak mohl konvoj stát plně autonomním, kde by vedoucí vozidlo nemuselo mít lidskou obsluhu.



Obr. 7: Výsledné GUI pro autonomní konvoj

## 7 ZÁVĚR

Jak vyplývá z teoretické analýzy, framework ROS je v oblasti robotiky jedním z nejlepších softwarů. Oplývá nepřehledným množstvím knihoven a nástrojů, které otevírají nové možnosti na poli vývoje mobilních robotů. Vytváří globálně nové standardy, kterými se má toto odvětví ubírat. Důkazem toho je, že i jedna z nejlepších světových firem v oblasti robotiky, Boston Dynamics, plně využívá jeho služeb. Jeho nástroj RViz, je dynamicky se rozvíjející grafické rozhraní s velkým množstvím využití při vývoji v robotice.

Projekt pěti bakalářských prací byl velmi ambiciózní, leč při pohledu na dosažené výsledky velmi uspokojivý. Zde navrhovaný a vytvořený systém je až na malé nedostatky, které byly popsány v předchozí kapitole, dostatečně kvalitní. Bylo dosaženo plně funkčního autonomního konvoje. Již od nezákladnějších částí, jako je sestavení a programování hardwaru, zpracování obrazu, celkovou integraci všech dílčích částí až po uživatelské grafické rozhraní.

Mobilní robotika a celkově autonomní konvoje jsou téma současnosti a blízké budoucnosti. Ve světovém měřítku se tímto tématem zabývá nespočet firem, které si uvědomují jeho dopad na poli vývoje dopravy. Tato práce by se mohla v budoucnu rozšířit co do obsahu, jako diplomová či dizertační práce, tak do kvality a následné využití různými firmami v praxi zabývajících se touto problematikou. Možností je tu celá řada, a je jen otázkou času, kdy se autonomní konvoje stanou součástí každodenního života lidí.



## 8 SEZNAM POUŽITÉ LITERATURY

- [1] What is Linux. *Linux.com* [online]. San Francisco: The Linux Foundation, c2016 [cit. 2017-05-08]. Dostupné z: <https://www.linux.com/what-is-linux>
- [2] Co je Ubuntu. *Co je Ubuntu* [online]. Praha: Občanské sdružení Ubuntu, 2008 [cit. 2017-05-08]. Dostupné z: <http://www.ubuntu.cz/>
- [3] VirtualBox. In: *Courses:b0b36prp:tutorials: virtualboxinstallation* [online]. Praha: Váňa, 2016 [cit. 2017-05-17]. Dostupné z: <https://cw.fel.cvut.cz/wiki/courses/b0b36prp/tutorials/virtualboxinstallation>
- [4] Instalace VirtualBoxu a Ubuntu. *Courses:b0b36prp:tutorials: virtualboxinstallation* [online]. Praha: Váňa, 2016 [cit. 2017-05-08]. Dostupné z: <https://cw.fel.cvut.cz/wiki/courses/b0b36prp/tutorials/virtualboxinstallation>
- [5] Instalace Ubuntu: průvodce instalací. *Wiki: Ubuntu* [online]. Praha: Spiras, 2015 [cit. 2017-05-08]. Dostupné z: [http://wiki.ubuntu.cz/instalace/pr%C5%AFvodce\\_instalac%C3%AD?redirect=1](http://wiki.ubuntu.cz/instalace/pr%C5%AFvodce_instalac%C3%AD?redirect=1)
- [6] ROS.org: Powering the world's robots. *ROS.org* [online]. Kalifornie: Garage, 2016 [cit. 2017-05-17]. Dostupné z: <http://www.ros.org/>
- [7] Distributions: ROS Wiki. *ROS.org* [online]. Kalifornie: Garage, 2007 [cit. 2017-05-08]. Dostupné z: <http://wiki.ros.org/Distributions>
- [8] Kinetic: installation. *ROS Wiki* [online]. Kalifornie: Garage, 2017 [cit. 2017-05-17]. Dostupné z: <http://wiki.ros.org/kinetic/Installation/Ubuntu>
- [9] Catkin: create\_a\_workspace. *ROS Wiki* [online]. Kalifornie: Garage, 2017 [cit. 2017-05-17]. Dostupné z: [http://wiki.ros.org/catkin/Tutorials/create\\_a\\_workspace](http://wiki.ros.org/catkin/Tutorials/create_a_workspace)
- [10] TOMÁŠ, P. *Návrh a realizace senzorického systému pro mobilní robot s využitím frameworku ROS*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 85 s. Vedoucí diplomové práce Ing. Stanislav Věchet, Ph.D..
- [11] Rviz: UserGuide. *ROS Wiki* [online]. Kalifornie: Garage, 2015 [cit. 2017-05-17]. Dostupné z: <http://wiki.ros.org/rviz/UserGuide>