HTBLuVA Wiener Neustadt
Höhere Lehranstalt für Informatik

DIPLOMARBEIT

Autonomous universal Mapping and Navigation

Ausgeführt im Schuljahr 2020/21 von:

Themengebiet 1
Lukas Leskovar
5BHIF

Themengebiet 2
Fabian Kleinrad
5BHIF

Betreuer / Betreuerin:

MMag. Dr. Michael Stifter

Wiener Neustadt, am July 9, 2020/21

Abgabevermerk: Übernommen von:

Chapter 1

Eidestattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die im Literaturverzeichnis angegeben Quellen und Hilfsmittel verwendet habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe.

Wiener Neustadt am July 9, 2020/21

Verfasser / Verfasserinnen:

Lukas Leskovar

Fabian Kleinrad

Contents

Chapter 2

Acknowledgement

The authors would like to thank ...

Chapter 3

Kurzfassung

asdf

Chapter 4

Abstract

asdf

Chapter 5

Introduction

Author: Lukas Leskovar

5.1 The Evolution of Robotics

Robotic research has always utilized concepts, processes, and methods of different scientific disciplines such as physics, mathematics, and biology to improve application and aid human needs. Because of this industrial, medical and even agricultural sectors have used technologies and products developed by researchers to improve workflows and alleviate employees from performing exhausting tasks. This relationship ranges back to the early ages of information technology in the 1950s and 1960s in which many developments on production robots and Artificial Intelligence (AI) have been made. Between 1970 and 1990 the public interest in automation and AI has decreased forcing the industry into the so-called AI winter. Despite this recession, research has been continued and the building blocks for another robot boom during the 1990s have been set. Since then the usage of robotic applications has broadened and the industry has proven itself to be a vital aspect of today's economy.

5.2 Robots with human interaction

Nowadays the utilization of robots in workplaces has broadened to almost every branch and is accepted by employees and workers. In countries like Japan, robots are no longer seen as a threat to jobs. Industrial robots are no longer used as simple construction tools, their safety and accuracy have improved so that collaborative robots (Cobots) are capable of working in close cooperation with humans. Surgery robots used in the medical sector not only allow for much more accurate procedures but also enable remote specialists to work on patients without having to be in the same hospital. In developed countries, educational robots are used at school or at home to teach children topics in a playful and interesting way.

[1] Malone, *George Devol: A Life Devoted to Invention, and Robots*

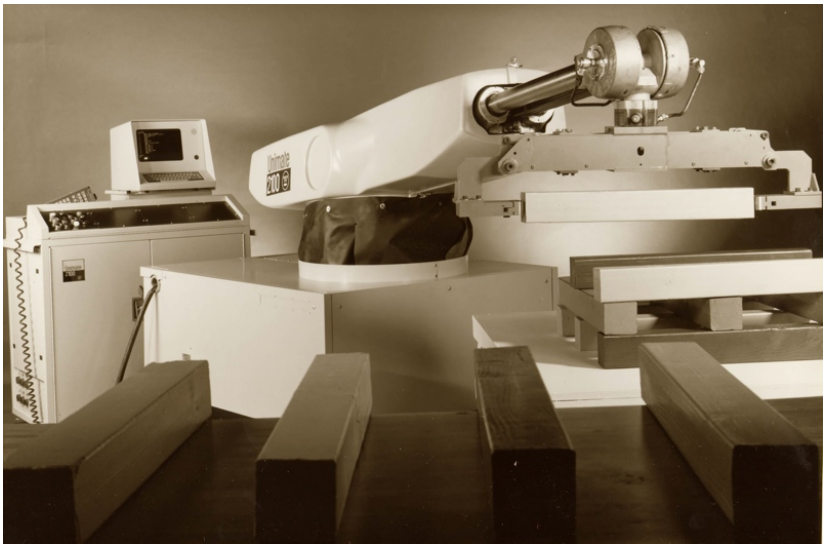


Figure 5.1: Picture of the first industrial robot. The Unimate developed by George Devol and Joseph Engelbert in 1961 was first used for hot die-casting and welding applications.[1]

5.3 Robots in hazardous environments

Robots do not only serve a purpose in a close-to-user work environment, they also ensure human safety by performing dangerous tasks in unsafe surroundings. Remotely controlled robots or drones can be used for inspecting mine shafts, collapsed buildings, pipelines, or overhead power poles. Other applications of such robots are bomb or mine defusion, fire extinction, or avalanche rescue.

5.4 Autonomous robots

Implementing an autonomous robot system is an intricate task that proposes many challenging problems for research or development teams. Autonomy requires a system to continuously work in a dynamic environment without external controlling inputs and utilize perceived information about its surroundings to adapt to environmental change.[2] Despite their complexity in development autonomous systems, mobile or stationary, immensely facilitate the execution of a job for the human user. Such robots can be used to navigate and organize warehouses, constructing parts in an assembly line, or map large areas for comprehensive calculations.

[2]Bekey, *Autonomous robots: from biological inspiration to implementation and control*, Pages 1-2.

Figure 5.2: The Emesent Hovermap mapping a dangerous area inside a mineshaft.[5]

5.5 3D Mapping

5.6 Autonomous 3D Mapping

While 3D Mapping is a well-established and growing economy most applications require human interaction at some point during the mapping process. Products such as the Emesent Hovermap[3] or Exyn Aero[4] utilize the spatial flexibility of drones and high-end light detection and ranging (LiDAR) Sensors to facilitate autonomous mapping in GPS-denied areas without being within sight of the drone pilot. These solutions are used to autonomously map building or explore hazardous environments such as mineshafts in a human-safe manner as seen in Fig. 5.2.

5.7 Goals

The project associated with this thesis aims to implement a 3D Mapping system similar to the aforementioned solutions with limited financial, personnel as well as temporal resources. This goal forces the project team to primarily maintain an open-source approach during development.

The goal of this thesis is to document the challenges encountered and experiences gained by the project team during development to demonstrate how highly sophisticated industrial problems can be solved in a low-budget fashion.

[3]Emesent Hovermap, *Autonomy Level 2 for Emesent Hovermap*.
[4]Exyn Aero, *Exyn Aero - Aerial Mapping Drone*.
[5]Emesent, *Emesent_Hovermap.JPG (JPEG Image, 2018 × 910 pixels)*

5.8 Requirements

In order to declare the project as successful the following criteria have to be implemented:

- The system is capable of generating a 3D Point-cloud of its surroundings
- All required hardware (e.g. sensors, computing boards, etc.) is mounted onto a drone-platform
- All calculations concerning the map-generation are run on a external server in direct communication with the drone
- The drone utilizes the Point-cloud to orientate in its surroundings and navigate one ore multiple waypoints
- The drone is capable of avoiding obstacles as it is moving through a unknown environment

The following criteria can be implemented but have no direct correlation to the success of the project:

- A Web-App facilitating the usage of the system and enabling the user to create waypoints, monitor the drone and mapping algorithm as well as evaluate the Point-cloud
- The system is replicated within the gazebo simulator to simplify future development without direct access to its hardware

Chapter 6

Study of Literature

Author:

Chapter 7

Robot Operating System

Author: Lukas Leskovar
This chapters objective is to describe the basic concepts of the Robot Operating System (ROS) utilized by Autumn. The ROS despite its name is a meta-operating system or middleware providing the utility and services often found in robotics frameworks. It enables the composition of distributed systems by utilizing publisher-subscriber communication between different programs of such systems. Furthermore ROS provides a comprehensive set of tools enabling the compilation, operation as well as testing, visualization and debugging of robotic systems. With its vast amount of libraries and huge open-source community providing useful functionality ROS facilitates the development of robotic applications without having to reimplement standardized technology.[1]

7.1 Conceptual Overview

The Robot Operating System can be divided into three conceptual levels each contributing a integral part to the utility of ROS. These different levels are described in the following sections.

File System

The File System Level mainly provides constraints and best practices for creating and structuring packages and their components.[2] ROS provides appropriate tools to facilitate file-system operations with and within packages.

Computational Graph

The Computational Graph provides crucial functionality to ROS as it refers to the peer-to-peer mesh network of processes (Nodes) each providing data to be utilized within the

[1] Gerkey, *Definition - ROS Answers*.
[2] Open Source Robotics Foundation, *Concepts - ROS Wiki*.

graph by publishing and subscribing to topics.[3] The concepts and technologies powering the computational graph are described in later in this chapter.

Community

The Community preserves the usability of ROS as new and useful packages and tools are created as well as existing functionality is being maintained.

7.2 Packages

Software in ROS is organized in packages containing nodes, libraries or any other piece of software providing functionality.
Since packages are the atomic unit of build and release they aim to be a slim as possible my implementing only a limited set of features. In other words packages should be implemented to provide minimal usability without being too large-scaled.[4] This means that each package is developed to work together with other packages to deliver utility as a connected system.
At file-system level packages simply refer to directories. While most subfolders and files within a package depend on its purpose, every package has to contain a package.xml and a CMakeLists.txt providing meta and build information. Packages can be build by utilizing rosbuild or catkin.[5]

Metapackages

Metapackages are specialized packages only containing a package.xml that logically links multiple related packages.[6] They can be used to conveniently install a group of packages simultaneously.

7.3 Nodes

The goal of ROS is to promote code reusability and decoupling of functionality to aid the versatility and usability of the system. Following this guideline every robotic system utilizing ROS consists of a fine-grained graph of processes called nodes. Each node provides computation on a single feature utilizing a ROS client library to communicate with others over a mesh-like peer-to-peer network.[7]
Exemplary for such as system would be one node running a LiDAR sensor, one responsible for localization, one performing motion planning, one controlling motor drivers and motors as well as one node running the robots main control loop.

[3] Open Source Robotics Foundation, *Concepts - ROS Wiki*.
[4] Open Source Robotics Foundation, *Packages - ROS Wiki*.
[5] Open Source Robotics Foundation, *Building Packages - ROS Wiki*.
[6] Open Source Robotics Foundation, *Metapackages - ROS Wiki*.
[7] Open Source Robotics Foundation, *Nodes - ROS Wiki*.

This architecture allows for a much more fault safe and less complex applications in comparison to monolithic systems.[8] This means that development and debugging are facilitated since errors can be contained within a singular slim node rather than a larger program.
To facilitate the search process each node has a node type consisting of the package name it is located and as well as the executable needed to run such node.

Services

The communication architecture in ROS utilizing a publisher-subscriber model is advantageous in most use-cases, however most distributed systems require remote procedure calls (RPC) which are not supported by default. Services enable communication over RPC by defining a pair of messages, one for requests and one for replies. Such service can then be attached to a node and called by a client using the service name.[9]

7.4 Communication

7.4.1 Topics

7.4.2 Messages

7.5 Master

7.6 Transformation Library

7.7 Simulation

7.7.1 URDF

7.7.2 Gazebo Simulator

[8] Stephens, *Beginning Software Engineering*, Page 94.
[9] Open Source Robotics Foundation, *Services - ROS Wiki*.

Chapter 8

Methodology

Author:

Chapter 9

Implementation

Author:

Chapter 10

Experiment 1

Author:

Chapter 11

Lessons learned

Author:

Chapter 12

Experiment 2

Author:

Chapter 13

Conclusion

Author:

Index

Authors Index

Literature Index

Bibliography

Bekey, George A. *Autonomous robots: from biological inspiration to implementation and control*. en. MIT press, 2005. URL: https://books.google.at/books?hl=de&lr=&id=8MbxCwAAQBAJ&oi=fnd&pg=PR7&dq=autonomous+robots&ots=5EXAZx-UDf&sig=u_CyTA0fa4aMRFAAdnbXpKhaf94&redir_esc=y#v=onepage&q=autonomous%20robots&f=false (visited on 06/30/2021).

Emesent. *Emesent_Hovermap.JPG (JPEG Image, 2018 × 910 pixels)*. URL: https://www.emesent.io/wp-content/uploads/2020/07/DSC_3700-1finalcrop.jpg?id=7339 (visited on 07/04/2021).

Emesent Hovermap. *Autonomy Level 2 for Emesent Hovermap*. URL: https://www.emesent.io/autonomy-level-2/ (visited on 07/04/2021).

Exyn Aero. *Exyn Aero - Aerial Mapping Drone*. URL: https://www.exyn.com/products/exyn-aero-aerial-mapping-drone (visited on 07/04/2021).

Gerkey, Brian. *Definition - ROS Answers*. en. URL: https://answers.ros.org/question/12230/what-is-ros-exactly-middleware-framework-operating-system/ (visited on 07/08/2021).

Malone, Bob. *George Devol: A Life Devoted to Invention, and Robots*. en. 2011. URL: https://spectrum.ieee.org/automaton/robotics/industrial-robots/george-devol-a-life-devoted-to-invention-and-robots (visited on 06/17/2021).

Open Source Robotics Foundation. *Building Packages - ROS Wiki*. en. URL: http://wiki.ros.org/ROS/Tutorials/BuildingPackages (visited on 07/08/2021).

— *Concepts - ROS Wiki*. en. URL: http://wiki.ros.org/ROS/Concepts (visited on 07/08/2021).

— *Metapackages - ROS Wiki*. en. URL: http://wiki.ros.org/Metapackages (visited on 07/08/2021).

— *Nodes - ROS Wiki*. en. URL: http://wiki.ros.org/Nodes (visited on 07/09/2021).

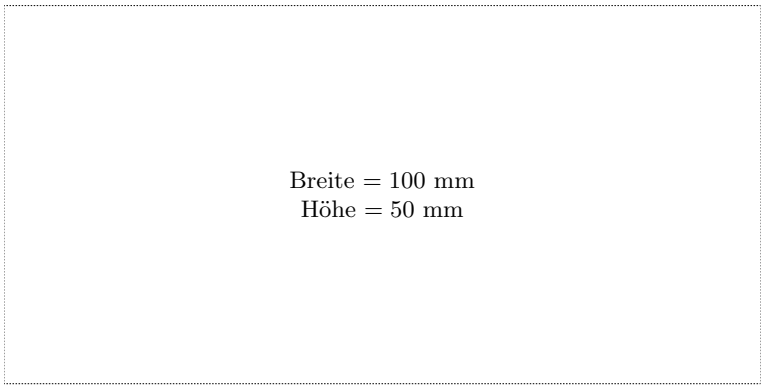
— *Packages - ROS Wiki*. en. URL: http://wiki.ros.org/Packages (visited on 07/08/2021).

— *Services - ROS Wiki*. en. URL: http://wiki.ros.org/Services (visited on 07/09/2021).

Stephens, Rod. *Beginning Software Engineering*. en. Wiley, 2015. ISBN: 9781118969168. URL: https://books.google.at/books?id=SyHWBgAAQBAJ (visited on 07/09/2021).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —