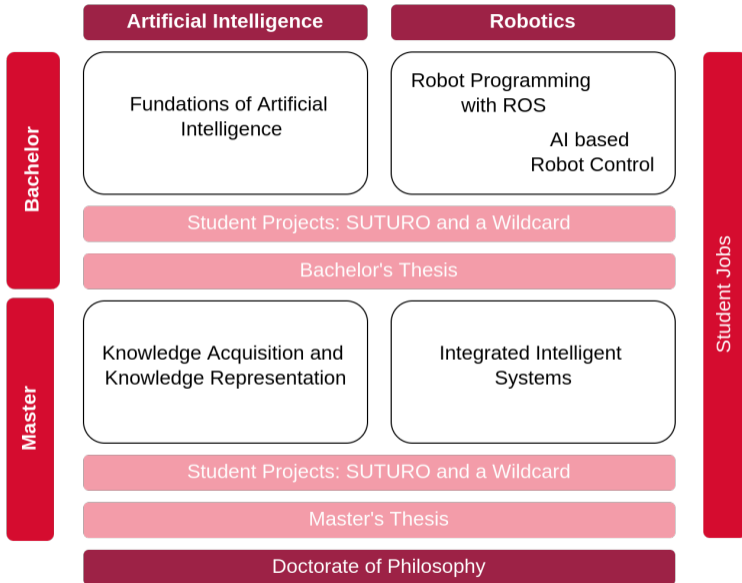


Robot Programming with ROS

1. Introduction, Overview

Arthur Niedźwiecki
19th Oct. 2023





Plan

- 1 Introduction
- 2 Course Overview
- 3 Organizational
- 4 Assignment

General Info

- Lecturers: Arthur, Alina (PhD students at IAI)
- Tutor: Stefan (WiMi at IAI)
- Correspondence: aniedz@cs.uni-bremen.de
- Dates: Thursdays, 14:15 - 15:45, 16:15 - 17:45
- Language: English and German
- Credits: 6 ECTS (4 SWS)
- Course type: practical course
- Course number: 03-IMVP-RPROS (03-BE-710.98b)
- Location: TAB Building, Room 0.30 EG

Plan

- 1 Introduction
- 2 Course Overview
- 3 Organizational
- 4 Assignment

Course Goals

Intended Learning Outcomes

Course Goals

Intended Learning Outcomes

- You can describe the components of a cognitive robot.

Course Goals

Intended Learning Outcomes

- You can describe the components of a cognitive robot.
- You can describe how a robot perceives the world.

Course Goals

Intended Learning Outcomes

- You can describe the components of a cognitive robot.
- You can describe how a robot perceives the world.
- You understand how an autonomous vacuum cleaner navigates.

Course Goal

You will learn / improve your skills in the following:

Course Goal

You will learn / improve your skills in the following:

- Git

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python
- Kinematics

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python
- Kinematics
- Sensors

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python
- Kinematics
- Sensors
- Communication Protocols

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python
- Kinematics
- Sensors
- Communication Protocols
- Coordinate Systems

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python
- Kinematics
- Sensors
- Communication Protocols
- Coordinate Systems
- Base Navigation

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python
- Kinematics
- Sensors
- Communication Protocols
- Coordinate Systems
- Base Navigation
- Task-Level Control

Course Goal

You will learn / improve your skills in the following:

- Git
- Linux
- Python
- Kinematics
- Sensors
- Communication Protocols
- Coordinate Systems
- Base Navigation
- Task-Level Control

...and get to play with a real little robot!

ROS - Robot Operating System

`https://www.ros.org/`

ROS - Robot Operating System

- Middleware for communication of the components of a robotic system

ROS - Robot Operating System

- Middleware for communication of the components of a robotic system
- "Meta-Operating System" for programming robotics software (configuring, starting / stopping, logging etc. software components)

ROS - Robot Operating System

- Middleware for communication of the components of a robotic system
- "Meta-Operating System" for programming robotics software (configuring, starting / stopping, logging etc. software components)
- Powerful build system (based on CMake), with a strong focus on integration and documentation

ROS - Robot Operating System

- Middleware for communication of the components of a robotic system
- "Meta-Operating System" for programming robotics software (configuring, starting / stopping, logging etc. software components)
- Powerful build system (based on CMake), with a strong focus on integration and documentation
- Language-independent architecture: C++, Python, Lisp and more

ROS - Robot Operating System

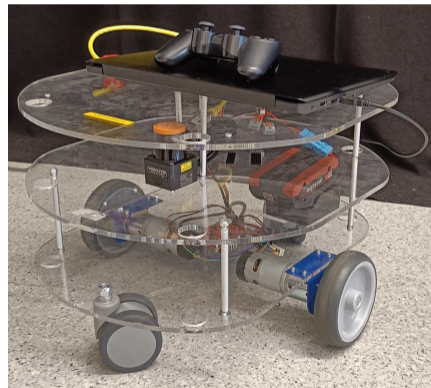
- Middleware for communication of the components of a robotic system
- "Meta-Operating System" for programming robotics software (configuring, starting / stopping, logging etc. software components)
- Powerful build system (based on CMake), with a strong focus on integration and documentation
- Language-independent architecture: C++, Python, Lisp and more
- According to ROS 2020 Community Metrics Report,
 - More than 2 million unique pageviews `wiki.ros.org` a month
 - More than 38 million downloads of `.deb` packages a month

ROS - Robot Operating System

- Middleware for communication of the components of a robotic system
- "Meta-Operating System" for programming robotics software (configuring, starting / stopping, logging etc. software components)
- Powerful build system (based on CMake), with a strong focus on integration and documentation
- Language-independent architecture: C++, Python, Lisp and more
- According to ROS 2020 Community Metrics Report,
 - More than 2 million unique pageviews `wiki.ros.org` a month
 - More than 38 million downloads of `.deb` packages a month
- *De facto* standard in modern robotics

TortugaBot

- 2 controllable wheels
- 2D laser scanner
- Thinkpad E485 PC with bluetooth
- PlayStation joystick



Rough schedule

Until Christmas 2023: Assignments in simulation

Rough schedule

Until Christmas 2023: Assignments in simulation

Jan - Mar 2024: Project in groups

- Controlling TortugaBot
- Heuristic decision-making
- The big day: *competition*

Plan

- 1 Introduction
- 2 Course Overview
- 3 Organizational**
- 4 Assignment

Grading

- Course final grade: 100 points = 50 homework + 50 group project.

Grading

- Course final grade: 100 points = 50 homework + 50 group project.
- You need at least 25 points from homeworks to participate in the project.

Grading

- Course final grade: 100 points = 50 homework + 50 group project.
- You need at least 25 points from homeworks to participate in the project.
- Final grade: 50 of 100 points - 4.0, 100 of 100 points - 1.0.

Grading

- Course final grade: 100 points = 50 homework + 50 group project.
- You need at least 25 points from homeworks to participate in the project.
- Final grade: 50 of 100 points - 4.0, 100 of 100 points - 1.0.
- $Grade = \frac{(100 - P_{your})}{(100 - 50)} * 3 + 1$

Homework assignments

- Filling in the missing gaps in already existing code.

Homework assignments

- Filling in the missing gaps in already existing code.
- New assignments from GitHub

Homework assignments

- Filling in the missing gaps in already existing code.
- New assignments from GitHub
- When ready, download and send the Notebook file.

Homework assignments

- Filling in the missing gaps in already existing code.
- New assignments from GitHub
- When ready, download and send the Notebook file.
- Homework is due in one week.

Homework assignments

- Filling in the missing gaps in already existing code.
- New assignments from GitHub
- When ready, download and send the Notebook file.
- Homework is due in one week.
- Solutions are discussed in the tutorial.

Scheinbedingungen Summary

- Graded homework every week until January, then group project
- Live presentation of the group project, individual grading
- 50 homework + 50 group project = 100 points for final grade
- At least 25 points from the homework to participate in the project
- Final grade: 50 of 100 points - 4.0, 100 of 100 points - 1.0.

- $Grade = \frac{(100 - P_{your})}{(100 - 50)} * 3 + 1$

Links

- This lectures website:

<https://ai.uni-bremen.de/teaching/cs-ros-ws23>

- Git reference book:

<https://git-scm.com/docs/gittutorial>

- Assignments repository:

<https://github.com/artnie/rpwr-assignments>

Info summary

Next class:

- Date: 26.10.
- Time: 14:15
- Place: same room (TAB 0.30)

Assignment:

- Due: 25.10, Wednesday, 23:59
- Points: 3 points
- For questions: write me a mail
or ask your colleagues in the StudIP forum

Plan

- 1 Introduction
- 2 Course Overview
- 3 Organizational
- 4 Assignment

Assignment goals

Set up your workspace



Set up your Git repository



Get comfortable with Jupyter



Install/Navigate a Linux terminal



Linux

Highly recommended to try Linux!

- Ubuntu 20.04 runs all of the institutes robot software
- ROS is best supported for Ubuntu 20.04
- Natively communicate with the TortugaBot in the project
- You can break everything

Ubuntu 20.04 - your options (Recommended)

Release page:

<https://releases.ubuntu.com/focal/>

- Dual boot to multiple OS (most robust)
Prepare boot stick, choose dual-boot during installation

Ubuntu 20.04 - your options (Recommended)

Release page:

<https://releases.ubuntu.com/focal/>

- Dual boot to multiple OS (most robust)
Prepare boot stick, choose dual-boot during installation
- Virtual machine through VirtualBox (least invasive)
<https://www.virtualbox.org/>

Ubuntu 20.04 - your options (Recommended)

Release page:

<https://releases.ubuntu.com/focal/>

- Dual boot to multiple OS (most robust)
Prepare boot stick, choose dual-boot during installation
- Virtual machine through VirtualBox (least invasive)
<https://www.virtualbox.org/>
- WSL2 - Windows Subsystem for Linux (best for Windows)
Full Ubuntu 20.04 CLI under Windows

[https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-10\](https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-10)

Robot Operating System (Recommended)

ROS Noetic runs on Ubuntu 20.04

- Installation guide

<https://wiki.ros.org/noetic/Installation/Ubuntu>

- Open access tutorials

<https://wiki.ros.org/ROS/Tutorials>

Assignments Repository

`https://github.com/artnie/rpwr-assignments`

Q & A

Thanks for your attention!