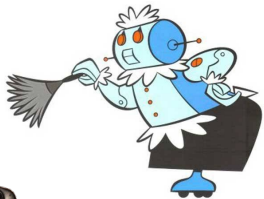
COMPUTER ENGINEERING PROJECT I

PROGRAMMING FOR ROBOTICS

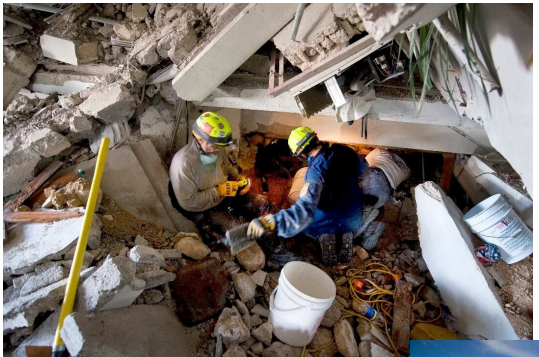
Course Introduction

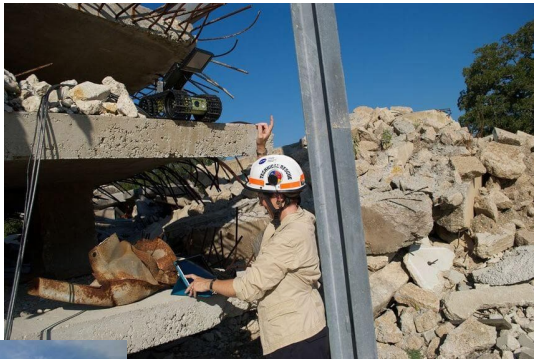
Jalil Boudjadar, Mirgita Frasheri

MOTIVATION



MOTIVATION: SEARCH AND RESCUE





Video

https://www.youtube.com/watch?v=FrgEbx6esYE

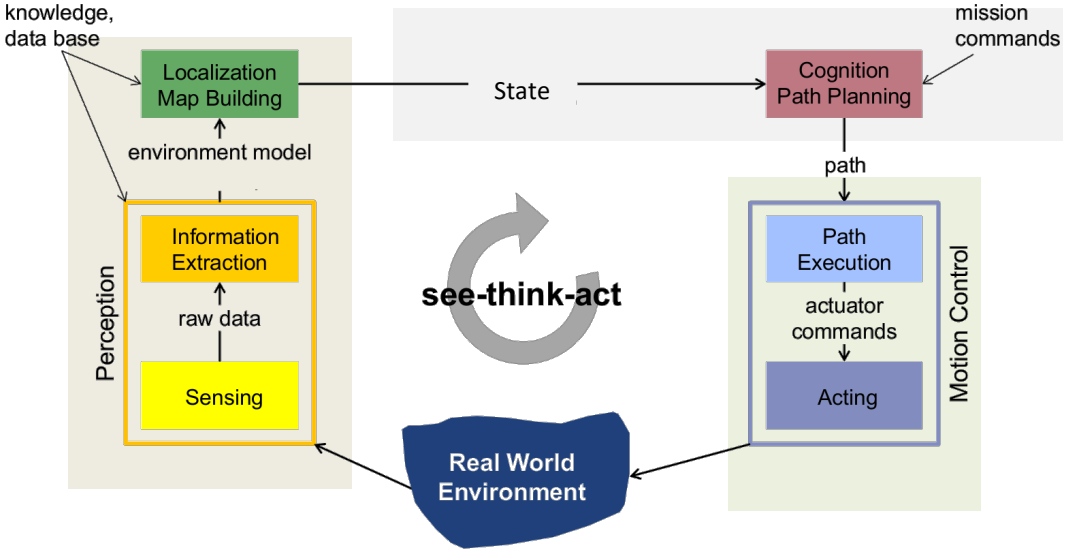
Demos from previous years:

1- Nav1

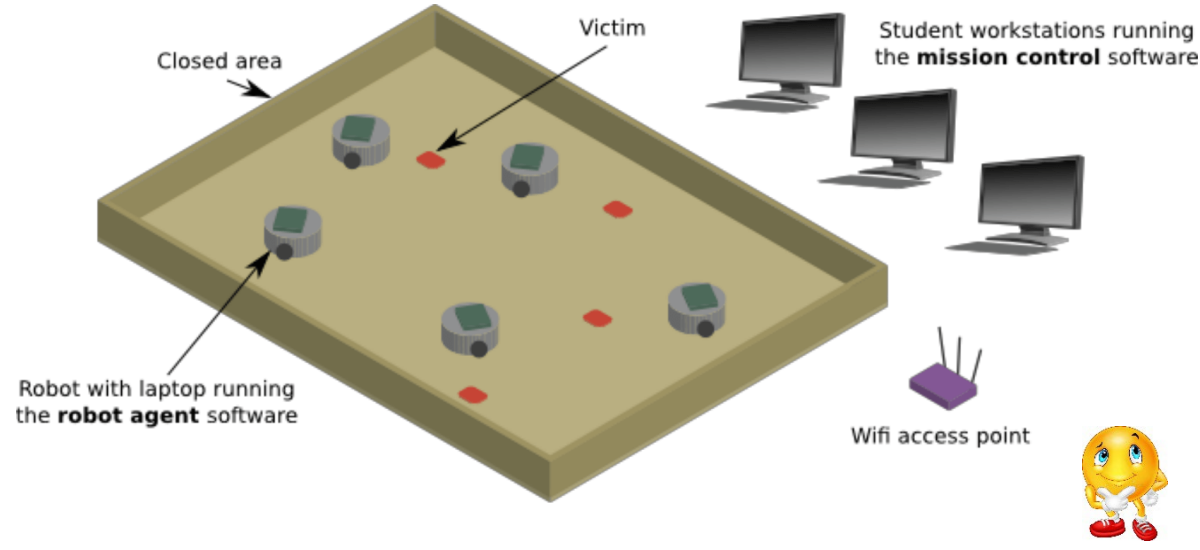
2- Nav2

Course Setup

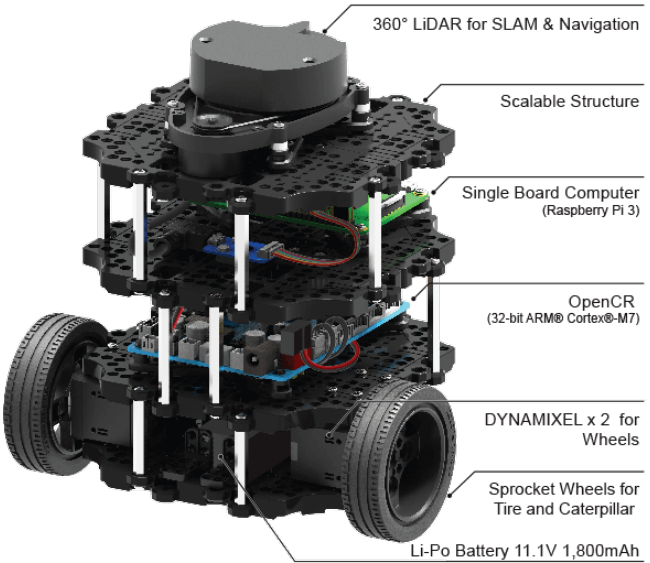
THIS COURSE: THE SEE-THINK-ACT CYCLE



SEARCH AND RESCUE: ARENA



SEARCH AND RESCUE: TURTLEBOT3 BURGER ROBOT

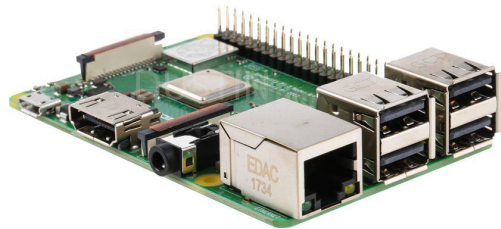


Sensors and computation boards

RGB sensor

Lidar 360 degree

Raspberry PI 3

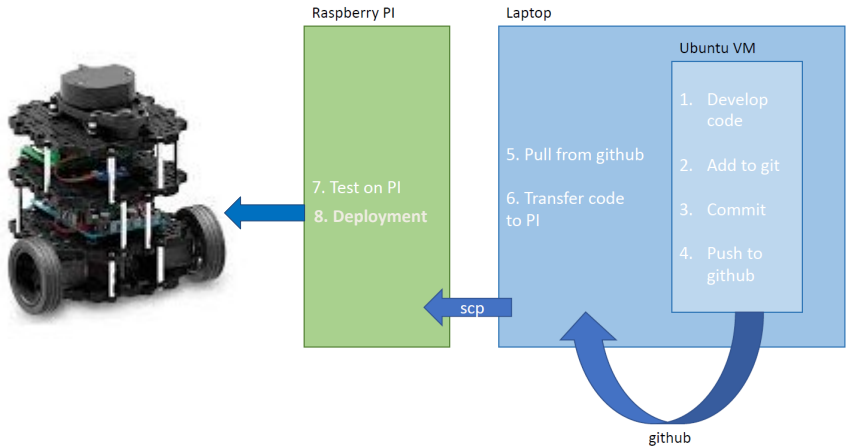


Programming & Environments

Actual robot

Embedded board

Computer systems



GOALS OF THE COURSE

1. Mounting TurtleBot3 robots as *sensor/actuator* system
2. Digital and combinatorial *circuits*,
3. *Embedded* programming on Raspberry PI,
4. Design of *real-time control* software
5. Working with Linux and ROS
6. Document and report a *system design*
7. Optimization
8. Team work
9. Have fun!!

Course Administrative Information

COURSE INFORMATION

- Hands-on course
- 5 ECTS Course -> 150 hours -> 14 Weeks
- evaluated with report and presentations
- 7-point grading scale
- Jalil Boudjadar (5123-413) -> jalil@ece.au.dk
- Mirgita Frasheri (5123-424) -> mirgita.frasheri@ece.au.dk

FORMAT

- Lectures:
  - Short lectures of about 30 minutes
- Projects:
  - Individual work with our assistance
  - Weekly project updates to the reports
  - Progress presentation

Course format

Put your engineering hats on

Trial & error

Google is your friend

Share experience with each other

We'll help you if you're stuck

What to expect

Working with robots can be frustrating ☹

- Sd cards get corrupted
- Sensor readings are not perfect
- New errors always on the corner
- Things simply break sometimes

It is also fun ☺

- Pure joy when things start working
- Learn from errors, the more you see the better in the long run

Your presentations

Each group presents at least once

Last 15min of every session dedicated to this

- Max 2 presentations per session
- Start week 3

Format

- Share what you learned that day or before
- 5-7 min tops
- slides not mandatory – your choice ☺

Book your group in the slot (find the time sheet in BS in week 1)

- First come first serve

Report prep

Week 6

- Send us an initial report structure
  - Main headings and subheading

Week 11

- More elaborated
- Bullet points after each subheadings

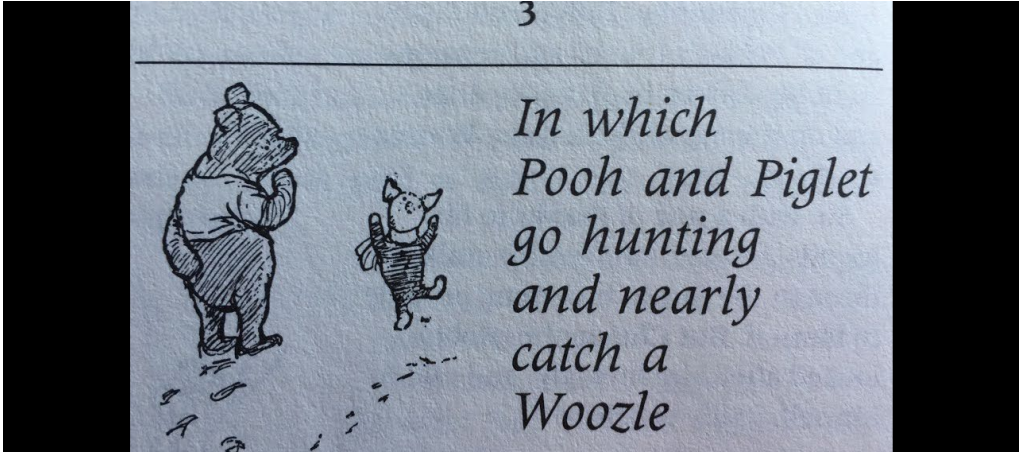
In each case we will provide feedback to you ☺

The selection of the template is free but we recommend a scientific article template available both in MS words and latex, e.g.

https://www.ieee.org/conferences/publishing/templates.html

Hints: Title, names, Introduction (Aim and Objectives), Specifications, Design, Implementation, Experiment set up and results, Discussion, Personal Contributions, References

The Winnie the Pooh method



Padlet

Report issues errors and their solutions

Find the link in Brightspace (week 1)

Intro to **Sense-Plan-Act**

Dive into **Robot Navigation**

Go deeper **Optimisation**

Bringing it home **Demo &Competition**

Platform: **Turtlebots**

Topics
1. **Raspberry PI architecture**
2. **Communicating with the PI**
3. **RGB sensor Setup**

We'll be using
1. PI
2. RGB sensor

Topics
Simple navigation:
1. **Read from lidar**
2. **Actuate robot (back, forward, left, right)**
3. **Count wall collisions**

Topics
1. **Single objective optimization**
   a. **Wrt speed**
2. **Multi-objective optimization**
   a. **Wrt speed + collision count**
   b. **Tag count**

To the ARENA!

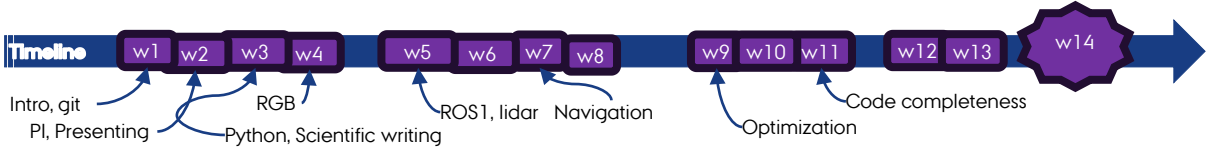
We'll be using
1. Lidar and RGB sensors on the Turtlebot

Foundation

How do we work?
Goals:
1. **Compatibility**
2. **Code reuse &sharing**

Tools:
1. **Version Control: git**
2. **Virtual Machines (VM)**

Progamming tools
OS: Ubuntu (Linux)
Language: **Python3**
Communication middleware: **Robot Operating System ROS1**



PLAN

Part I: Basics to work with Raspberry PI
Week 1: Course introduction: structure, goals, assignments, reports. gitHub
Week 2: Introduction to programming with Raspberry PI
Week 3: Programming with Python.
Week4: Programming a RGB sensor
Part II: ROS Programming
Week 5-6: Working with ROS for Robots programming
Week 7-8: Robots navigation
Part III: Optimization
Week 9: Optimization of the robots navigation speed
Week10: Optimization with respect to both speed and collision avoidance
Week11: Advanced optimization
Part IV: Competition
Week12-13: Finalizing implementation
Week 14: Competition, demo and examination

COURSE OPERATION

- Do
  - Participate in class
  - Ask questions (you will be doing others a favour)
  - Discuss course material, readings, assignments with other students
  - Cite any work you use in your report and project

- Don't
  - Cheat or plagiarize
  - Uncited use of any material from anywhere
  - Share / steal any material with/from former or current students

LEARNING OUTCOMES

By the end of the course the student will be able to:

- Conduct a design of computer system from a problem description.
- Evaluate a hardware-software configuration
- Explore design space alternatives
- Document the design of a computer system
- Discuss the strengths and weaknesses of the project

TODAY

- Form your groups
- Choose presentation spots (Padlet)
- Few slides about github, then install Git.
- Mini-lecture about Raspberry PI
  - PI architecture
  - How to access
  - Tutorial and exercise

TOOLS

- versioning -> git
  https://git-scm.com/
- Latex
  https://www.xmlmath.net/texmaker/
- Ubuntu
  https://ubuntu.com/
- ROS
  http://www.ros.org/
- Python
  https://www.python.org/
- Spyder
  https://www.spyder-ide.org/