

EMC 2018

Tooling and Infrastructure

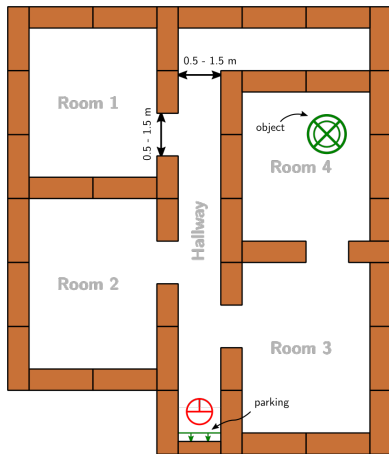
Wouter Kuijpers

Eindhoven University of Technology
Department of Mechanical Engineering

May 2, 2018

The Assignment

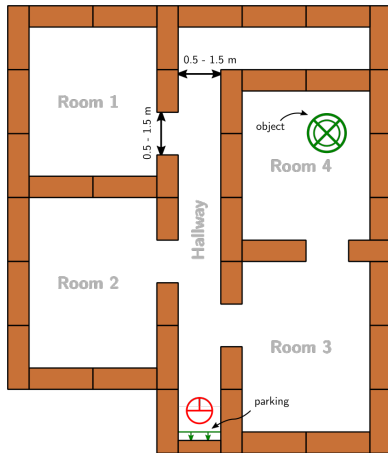
Final Competition: Let a robot map a Hospital and afterwards find an object.



The Assignment

Final Competition: Let a robot map a Hospital and afterwards find an object.

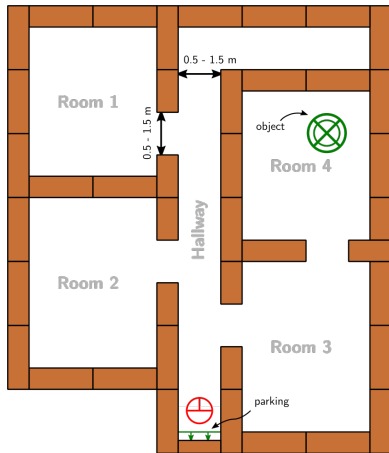
- ▶ You have to:
 - ▶ try to be as fast as possible



The Assignment

Final Competition: Let a robot map a Hospital and afterwards find an object.

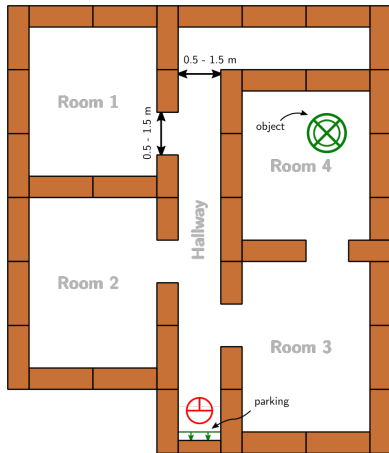
- ▶ You have to:
 - ▶ try to be as fast as possible
- ▶ You can use:
 - ▶ The **Laser Range Finder** to detect walls and doors
 - ▶ The **encoder** data from the wheels
 - ▶ The **control effort** signal to notice touches
 - ▶ The **(high level) hint** we will give you!



The Assignment

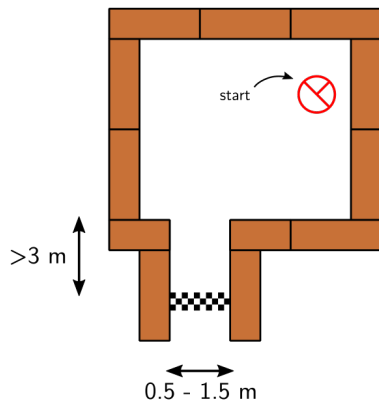
Final Competition: Let a robot map a Hospital and afterwards find an object.

- ▶ You have to:
 - ▶ try to be as fast as possible
- ▶ You can use:
 - ▶ The **Laser Range Finder** to detect walls and doors
 - ▶ The **encoder** data from the wheels
 - ▶ The **control effort** signal to notice touches
 - ▶ The **(high level) hint** we will give you!
- ▶ Important Dates:
 - ▶ Final Presentations: **June 6**
 - ▶ Competition Day: **June 13**



Intermediate Assignment

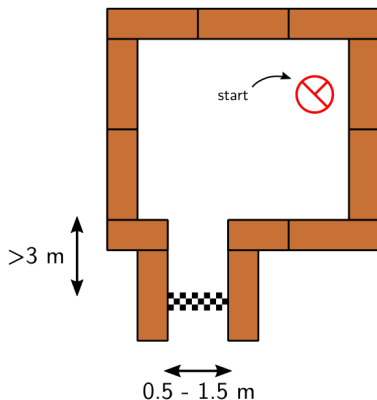
Escape Room Competition: let a robot escape the room through the door.



Intermediate Assignment

Escape Room Competition: let a robot escape the room through the door.

- ▶ You have to:
 - ▶ try to be *as fast as possible*
- ▶ You can use:
 - ▶ The **Laser Range Finder** to detect walls
 - ▶ The **encoder** data from the wheels
 - ▶ The **control effort** signal to notice touches
- ▶ Competition day: **May 23**



Simple, right?

Simple, right?

Don't worry, we supply you with some tools to get you started!

Introducing the Robot: PICO



- ▶ PICO is the robot you have to use!
- ▶ Telepresence Robot from Aldebaran
 - ▶ Robot type: *Jazz*
- ▶ Sensors:
 - ▶ Laser Range Finder (LRF)
 - ▶ Wheel encoders (odometry)
 - ▶ 170° wide-angle camera

Introducing the Robot: PICO



- ▶ PICO is the robot you have to use!
- ▶ Telepresence Robot from Aldebaran
 - ▶ Robot type: *Jazz*
- ▶ Sensors:
 - ▶ Laser Range Finder (LRF)
 - ▶ Wheel encoders (odometry)
 - ▶ 170° wide-angle camera
- ▶ Actuators:
 - ▶ Holonomic base (omni-wheels)
 - ▶ Pan-tilt unit for head

Introducing the Robot: PICO



- ▶ PICO is the robot you have to use!
- ▶ Telepresence Robot from Aldebaran
 - ▶ Robot type: *Jazz*
- ▶ Sensors:
 - ▶ Laser Range Finder (LRF)
 - ▶ Wheel encoders (odometry)
 - ▶ 170° wide-angle camera
- ▶ Actuators:
 - ▶ Holonomic base (omni-wheels)
 - ▶ Pan-tilt unit for head
- ▶ Computer:
 - ▶ Intel I7
 - ▶ Running *Ubuntu 16.04*

ROS

- ▶ Robot Operating System
 - ▶ Open-source meta-operating system for robots

ROS

- ▶ Robot Operating System
 - ▶ Open-source meta-operating system for robots

- ▶ Won't be using it!

ROS

- ▶ Robot Operating System
 - ▶ Open-source meta-operating system for robots
- ▶ Won't be using it!
- ▶ Instead, we will provide our own 'software layer'
 - ▶ It is simpler to understand, and 'cleaner' to use

ROS

- ▶ Robot Operating System
 - ▶ Open-source meta-operating system for robots
- ▶ Won't be using it!
- ▶ Instead, we will provide our own 'software layer'
 - ▶ It is simpler to understand, and 'cleaner' to use
- ▶ However, you are still **allowed** to use ROS!

Ubuntu

Development of PICO's software will be done in Ubuntu.

Ubuntu

Development of PICO's software will be done in Ubuntu.

- ▶ Linux-based operating system
- ▶ Use version **16.04** (not 14.10, 15 or 17!)
- ▶ 32- and 64-bit (64-bit recommended)

Ubuntu

Development of PICO's software will be done in Ubuntu.

- ▶ [Linux](#)-based operating system
- ▶ Use version **16.04** (not 14.10, 15 or 17!)
- ▶ 32- and 64-bit ([64-bit recommended](#))
- ▶ Easy dual boot installation with e.g., Windows
- ▶ Download: [see tutorial!](#)
 - ▶ Any problems? → [Check the wiki.](#)
 - ▶ No info? → Send us an email.



C++

- ▶ We will use C++ as programming language
- ▶ C++ is object-oriented C
 - ▶ “C with Classes”
 - ▶ Encapsulate data and functionality within objects

C++

- ▶ We will use C++ as programming language
- ▶ C++ is object-oriented C
 - ▶ “C with Classes”
 - ▶ Encapsulate data and functionality within objects
- ▶ It is a **powerful** but **complex** programming language.
- ▶ However, we provide you the **EMC framework** to get you started

Creating code: Qt Creator

- ▶ Integrated Development Environment
 - ▶ Advanced code editor
- ▶ Many advantages over 'simple editors':
 - ▶ Syntax highlighting
 - ▶ Code completion
 - ▶ Visual compiler feedback
 - ▶ Static code checking
 - ▶ Refactoring tools
 - ▶ Parenthesis matching
 - ▶ ...



Git Version Control

- ▶ Version Control System:
 - ▶ *'Manages files and directories, and the changes made to them, over time'*
- ▶ Used to **store** and maintain your code on the server
 - ▶ (Like Dropbox)

Git Version Control

- ▶ Version Control System:
 - ▶ *'Manages files and directories, and the changes made to them, over time'*
- ▶ Used to **store** and maintain your code on the server
 - ▶ (Like Dropbox)
- ▶ Maintains **version history**
- ▶ Is **distributed**
 - ▶ You always have the full history on your pc
 - ▶ You can always go back to a version, show differences, even when **off-line**

Git Version Control

- ▶ Version Control System:
 - ▶ *'Manages files and directories, and the changes made to them, over time'*
- ▶ Used to **store** and maintain your code on the server
 - ▶ (Like Dropbox)
- ▶ Maintains **version history**
- ▶ Is **distributed**
 - ▶ You always have the full history on your pc
 - ▶ You can always go back to a version, show differences, even when **off-line**
- ▶ More info on the Wiki

PICO Simulator

- ▶ You will have to work with the **real robot**, but we **only have one**. Therefore:

PICO Simulator

- ▶ You will have to work with the **real robot**, but we **only have one**. Therefore:
- ▶ **Test time is limited and has to be scheduled, see Wiki!**

PICO Simulator

- ▶ You will have to work with the **real robot**, but we **only have one**. Therefore:
- ▶ **Test time is limited and has to be scheduled, see Wiki!**
- ▶ **PICO Simulator:**
 - ▶ Simulates:
 - ▶ Sensors (Laser, odometry)
 - ▶ Actuators (Base)
 - ▶ Environment (maze)

PICO Simulator

- ▶ You will have to work with the [real robot](#), but we [only have one](#). Therefore:
- ▶ **Test time is limited and has to be scheduled, see Wiki!**
- ▶ **PICO Simulator:**
 - ▶ Simulates:
 - ▶ Sensors (Laser, odometry)
 - ▶ Actuators (Base)
 - ▶ Environment (maze)
- ▶ Can easily create test environments using [height maps](#)

PICO Simulator

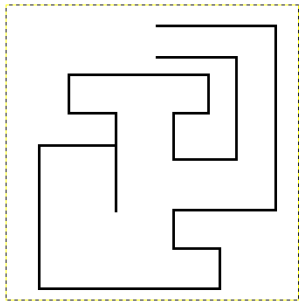
- ▶ You will have to work with the [real robot](#), but we [only have one](#). Therefore:
- ▶ **Test time is limited and has to be scheduled, see Wiki!**
- ▶ **PICO Simulator:**
 - ▶ Simulates:
 - ▶ Sensors (Laser, odometry)
 - ▶ Actuators (Base)
 - ▶ Environment (maze)
- ▶ Can easily create test environments using [height maps](#)
- ▶ [Integrates](#) well with our provided software
 - ▶ If your software runs in the simulator, it runs on the robot

PICO Simulator

- ▶ You will have to work with the **real robot**, but we **only have one**. Therefore:
- ▶ **Test time is limited and has to be scheduled, see Wiki!**
- ▶ **PICO Simulator:**
 - ▶ Simulates:
 - ▶ Sensors (Laser, odometry)
 - ▶ Actuators (Base)
 - ▶ Environment (maze)
- ▶ Can easily create test environments using **height maps**
- ▶ **Integrates** well with our provided software
 - ▶ If your software runs in the simulator, it runs on the robot
 - ▶ This does **not** guarantee that it will also work...

You still need to test on the real system!

PICO Simulator



Example

- ▶ [Full Example](#): from requirements, through Task-Skill-Motion to Software Executable.
- ▶ (far) from perfect!
- ▶ Focus on decoupling parts of functionality, explicitly in the code.
- ▶ Will be released this week! Check the [tutorial page](#)!

Wiki

- ▶ EMC Wiki:
 - ▶ http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control
 - ▶ Info on practical assignment, installation, getting started
 - ▶ Frequently Asked Questions
 - ▶ Log-in: student account

Wiki

- ▶ EMC Wiki:
 - ▶ http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control
 - ▶ Info on practical assignment, installation, getting started
 - ▶ Frequently Asked Questions
 - ▶ Log-in: [student account](#)
- ▶ Group pages on EMC Wiki:
 - ▶ Each group gets its own page
 - ▶ [Update at least weekly](#)

Wiki

- ▶ EMC Wiki:
 - ▶ http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control
 - ▶ Info on practical assignment, installation, getting started
 - ▶ Frequently Asked Questions
 - ▶ Log-in: [student account](#)
- ▶ Group pages on EMC Wiki:
 - ▶ Each group gets its own page
 - ▶ [Update at least weekly](#)
- ▶ Overall use:
 - ▶ Everyone can [edit](#)
 - ▶ If you see a mistake: [correct it](#)
 - ▶ If you had a problem and know how to fix it: [add it](#)

Recap

- ▶ Robot: [PICO](#)
- ▶ OS: [Ubuntu 16.04](#)
- ▶ Programming language: [C++](#)
- ▶ Code editor: [Qt Creator](#)
- ▶ Version control: [git](#)
- ▶ Simulation: [PICO simulator](#)
- ▶ Documentation: [Wiki](#)

That should get you started!

Groups

Each group will be supervised by a tutor:

- | | |
|-------------------|--------------------|
| 1. Yanick Douven | 6. Marzieh |
| 2. Wouter Houtman | 7. Wouter Kuijpers |
| 3. RUVU | 8. Hao |
| 4. Bob | 9. Marzieh |
| 5. Bob & Hao | 10. René & Herman |

It is *your* responsibility to get in touch with your tutor (see Wiki)

What should I do now?

- ▶ Check the Wiki & Finish the Tutorials:
 - ▶ `http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control`

What should I do now?

- ▶ Check the Wiki & Finish the Tutorials:
 - ▶ `http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control`
- ▶ Send an email to your tutor:
 - ▶ to schedule the first meeting,
 - ▶ with one username for access to your Git, (tutorial)

What should I do now?

- ▶ Check the Wiki & Finish the Tutorials:
 - ▶ `http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control`
- ▶ Send an email to your tutor:
 - ▶ to schedule the first meeting,
 - ▶ with one username for access to your Git, (tutorial)
- ▶ With your group:
 - ▶ schedule a try-out test with PICO, next week (7-8-9 May), see test scheme on Wiki!

Groups (1)

Group 1

- ▶ 0914013
- ▶ 0924842
- ▶ 1279491
- ▶ 1031018
- ▶ 0898396
- ▶ 1279602

Group 2

- ▶ 1275801
- ▶ 1037038
- ▶ 0848638
- ▶ 0899061
- ▶ 0843128
- ▶ 0895324

Group 3

- ▶ 0861750
- ▶ 0885514
- ▶ 0883056
- ▶ 0896947
- ▶ 0848904
- ▶ 0909434

Group 4

- ▶ 1286560
- ▶ 0852908
- ▶ 0774811
- ▶ 1032743
- ▶ 0740573
- ▶ 0897675

Groups (2)

Group 5

- ▶ 0847751
- ▶ 0897620
- ▶ 0887636
- ▶ 0903892
- ▶ 0810317
- ▶ 1019851

Group 6

- ▶ 0896965
- ▶ 1036818
- ▶ 0486100
- ▶ 0912153
- ▶ 0778266
- ▶ 1280554

Group 7

- ▶ 1022624
- ▶ 1279483
- ▶ 1279637
- ▶ 1275828
- ▶ 0886654
- ▶ 0833049

Group 8

- ▶ 0817997
- ▶ 1030747
- ▶ 0890579
- ▶ 0892629
- ▶ 0859466
- ▶ 0885734

Groups (3)

Group 9

- ▶ **0892672**
- ▶ 1283251 (?)
- ▶ **1279785**

Group 10

- ▶ 1020646
- ▶ 1283685
- ▶ 1221543
- ▶ 0714775
- ▶ 0767539