EMC 2013
Practical Assignment
A-maze-ing PICO

Sjoerd van den Dries
Eindhoven University of Technology
Department of Mechanical Engineering

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

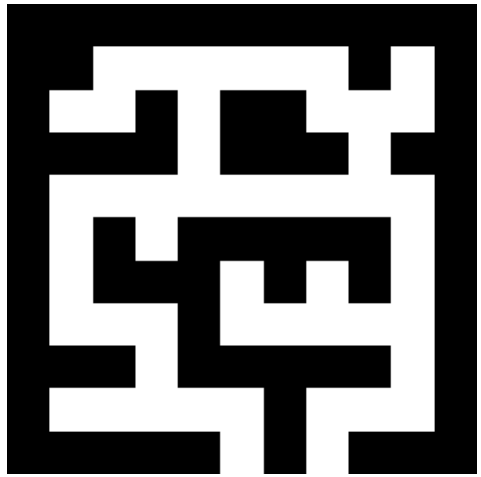
September 4, 2013

Introducing PICO

- Telepresence Robot from Aldebaran
  - Robot type: Jazz
- Sensors:
  - Laser Range Finder (LRF)
  - 170° wide-angle camera
  - Sonar
- Actuators:
  - Differential wheels
  - Pan-tilt unit for head
- Computers:
  - Internal: Intel Atom
  - External: Laptop, I7
  - Running Ubuntu
  - ROS (Robot Operating System)

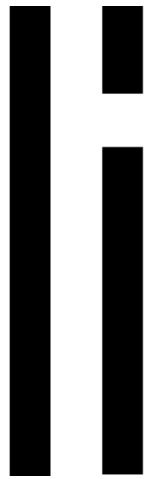
The Assignment

- Let PICO navigate through a maze and find and go to the exit.
- You have to:
  - try to be as fast as possible
  - but avoid hitting obstacles at all cost!
- You can use:
  - The Laser Range Finder to detect walls
  - (Optionally: use camera to detect arrows pointing to the exit)
- Competition day: October 23rd



Intermediate Assignment

- Corridor Competition: Let PICO drive through a corridor and go through the side exit.
- You have to:
  - try to be as fast as possible
  - but avoid hitting obstacles at all cost!
- You can use:
  - The Laser Range Finder to detect walls
- Competition day: September 25th



Ubuntu

- Linux-based operating system
- Use version 12.04 (Long Term Support release)
- 32- and 64-bit (64-bit recommended)
- Easy dual boot installation with *e.g.*, Windows
- Download: `www.ubuntu.com`
  - Any problems? → Check the wiki.
  - No info? → Ask the ICT Servicedesk or contact us.

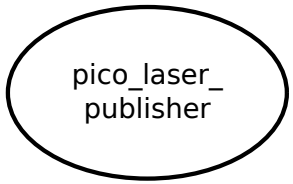
ROS

- Robot Operating System
- Open-source meta-operating system for robots
- Primary goal: support code reuse in robotics R&D
- Implemented in C++, Python
- Allows running code on multiple computers
- We will use ROS Fuerte
- www.ros.org

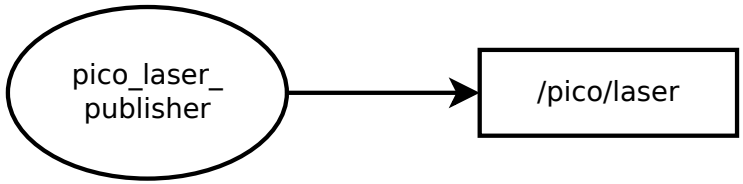
ROS Concepts

- Node: process that performs computation
- Master: provides name registration and lookup
- Messages: nodes communicate with each other by passing messages
- Topics: named buses over which nodes exchange messages
- Services: request/reply communication
- Parameter Server: allows data storage by key in a central location

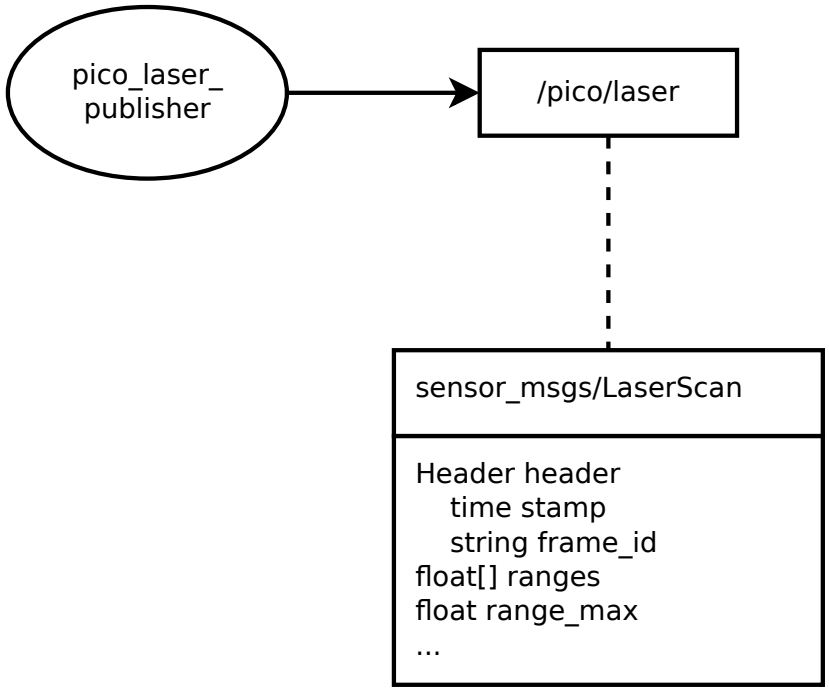
ROS Communication Example



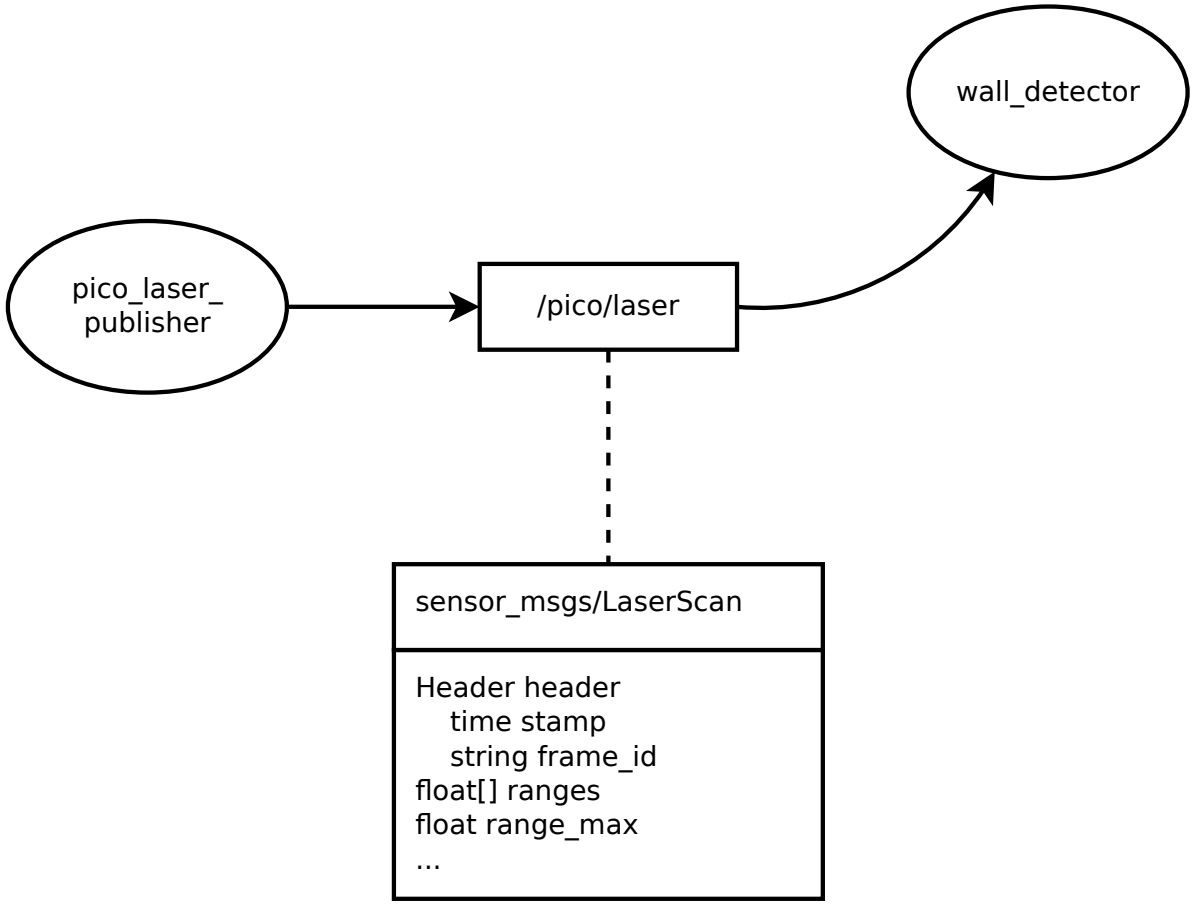
ROS Communication Example



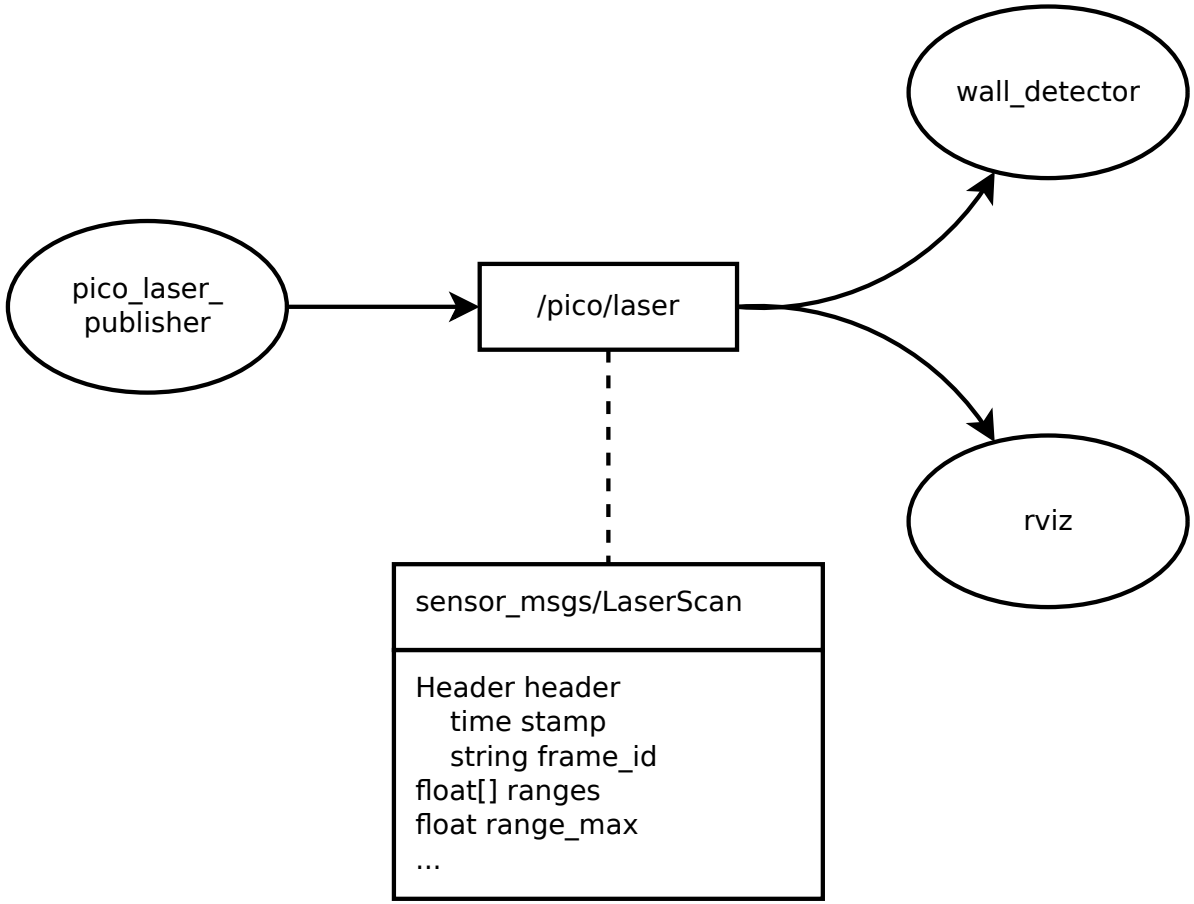
ROS Communication Example



ROS Communication Example



ROS Communication Example



ROS Filesystem

- Package: contains one chunk of functionality with clear input and output
  - A module, library, set of tools, ...
  - *E.g.*, `wall_detector`
- Manifest: contains meta data about a package
  - Dependencies
  - Author
  - ...

ROS Filesystem

- Package: contains one chunk of functionality with clear input and output
  - A module, library, set of tools, ...
  - *E.g.*, `wall_detector`
- Manifest: contains meta data about a package
  - Dependencies
  - Author
  - ...
- Stack: collection of packages that provides functionality as a whole
  - *E.g.*, `navigation`
- Stack manifest: contains meta data about the stack

ROS Community

- Enormous amount of open-source libraries and tools available!
  - Approx. 2000 packages and counting!
  - Perception
  - Navigation
  - Manipulation
  - Visualization
  - Etc . . . Etc . . .

ROS Community

- Enormous amount of open-source libraries and tools available!
  - Approx. 2000 packages and counting!
  - Perception
  - Navigation
  - Manipulation
  - Visualization
  - Etc . . . Etc . . .
- Thriving community
  - Approx. 100 research groups contributing (registered!)
  - Mailing lists, forums, . . .
  - Spin-offs:
    - ROS industrial

ROS Tutorials

- http://wiki.ros.org/ROS/Tutorials
- Skip tutorials with Catkin in the name
- Whenever possible, select the rosbuild version

C++

- We will use C++ as programming language
- One of the two core ROS languages
  - Packages roscpp and roslib
- C++ is object-oriented C
  - “C with Classes”
  - Encapsulate data and functionality within objects
- More on C++ and ROS next week

C++

- We will use C++ as programming language
- One of the two core ROS languages
  - Packages roscpp and roslib
- C++ is object-oriented C
  - “C with Classes”
  - Encapsulate data and functionality within objects
- More on C++ and ROS next week
- In the meantime: many good tutorials available, *e.g.*:
  - http://www.cplusplus.com/doc/tutorial

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
  - ...

Subversion (SVN)

- Version Control System:
  - *'Manages files and directories, and the changes made to them, over time'*
- Use to store and maintain your code on the server

Subversion (SVN)

- Version Control System:
  - *'Manages files and directories, and the changes made to them, over time'*
- Use to store and maintain your code on the server
- Basic commands:
  - `svn checkout <URL>`
  - `svn add <FILENAME / DIRECTORY>`
  - `svn commit -m '...message...'`
  - `svn status`

Subversion (SVN)

- We created a dedicated SVN repository for EMC
- SVN-account per group

Subversion (SVN)

- We created a dedicated SVN repository for EMC
- SVN-account per group
- https://roboticssrv.wtb.tue.nl/svn/emc/2013
  - general *(accessible to all)*
    - jazz_gazebo
    - jazz_example
    - ...
  - groups *(accessible per group)*
    - emc01
    - emc02
    - ...

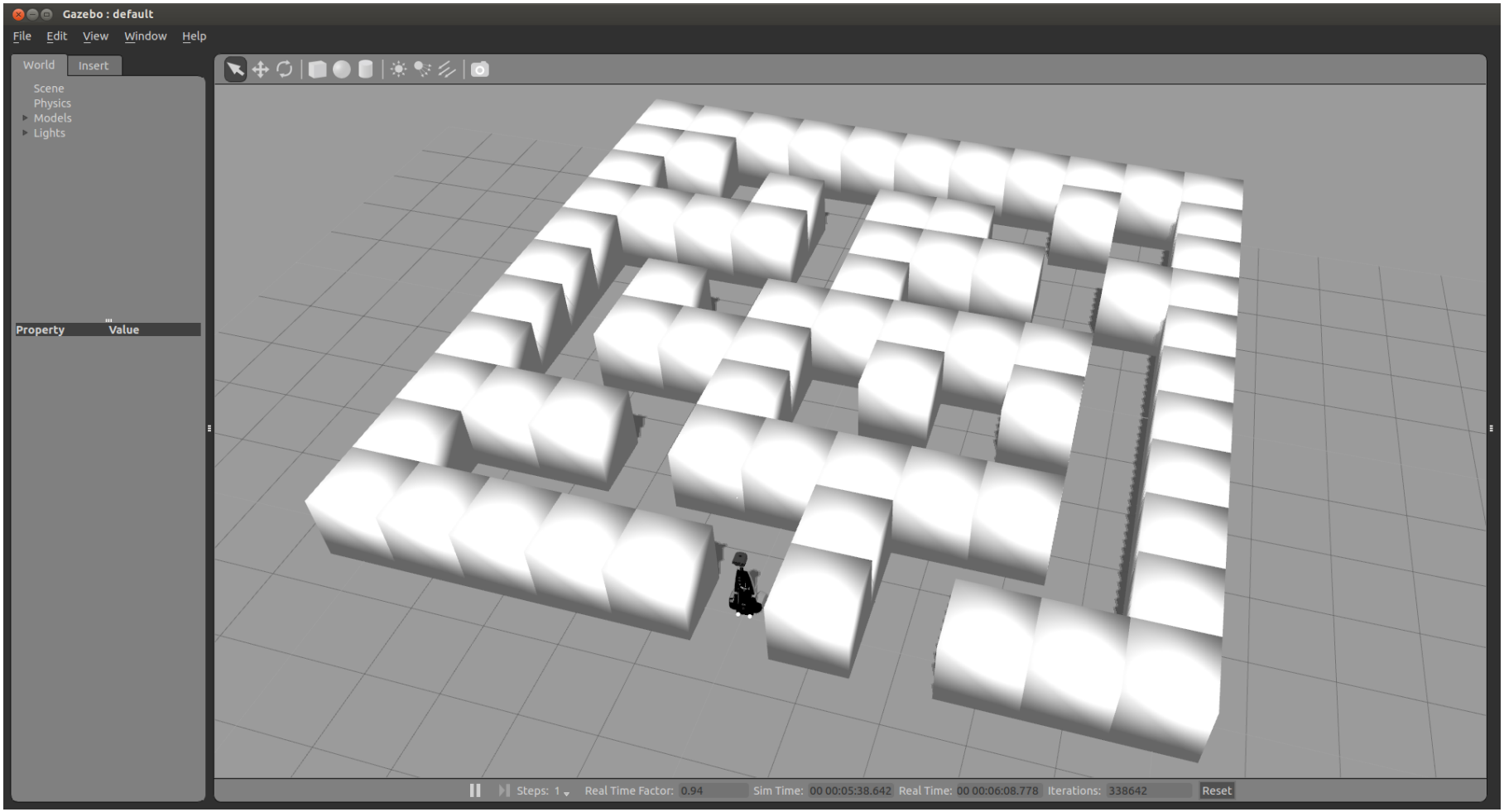
PICO Simulator

- 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:
- PICO Simulation:
  - Build in Gazebo simulator
  - Simulates:
    - Sensors
    - Actuators
    - Environment (maze)
    - Physics
  - Integrates well with ROS

PICO Simulator



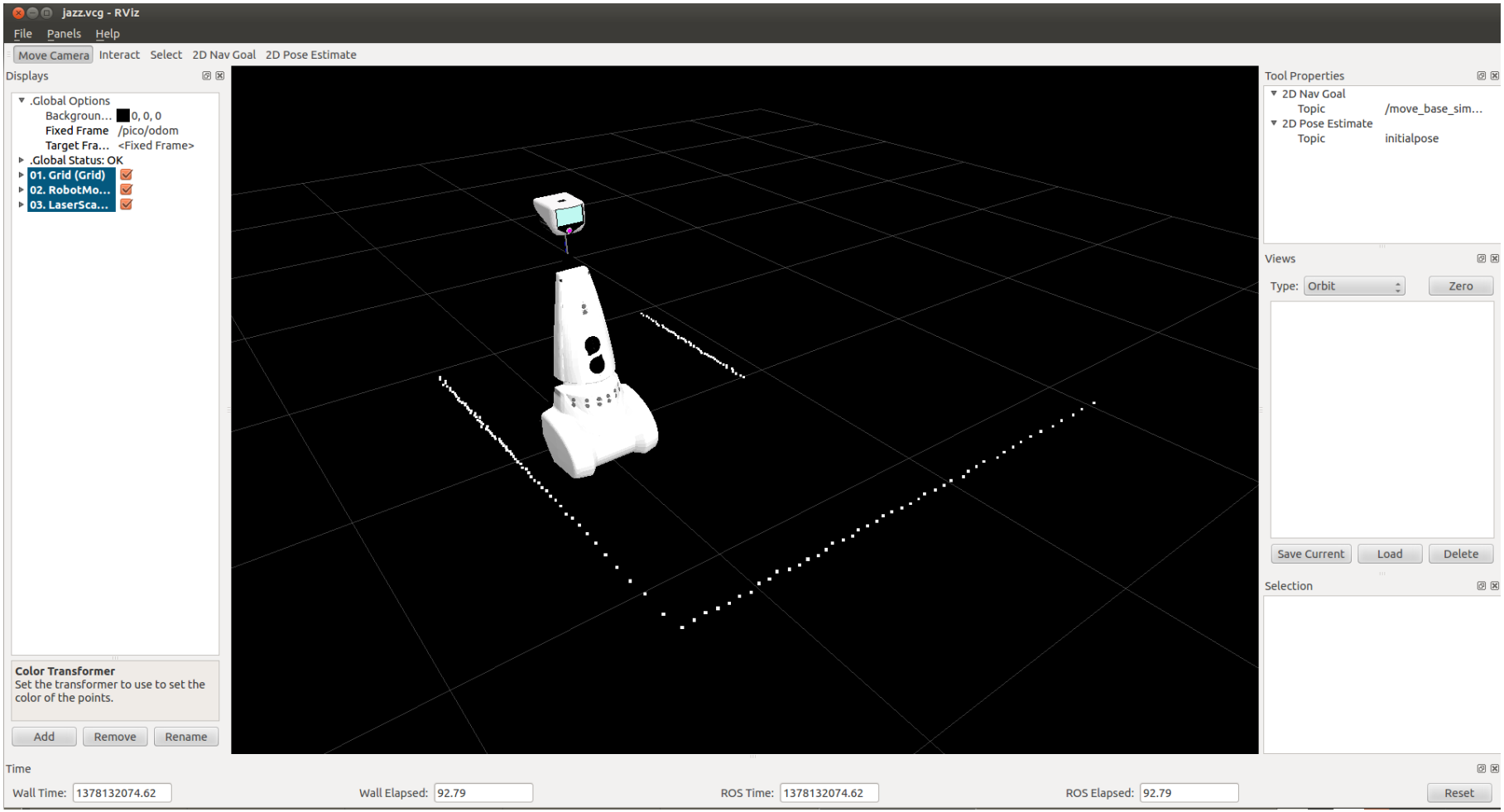
RViz - Visualization

- RViz
  - 3D visualization tool for ROS
  - Visualizes:
    - Robot model
    - Sensor data
    - Odometry information
    - Markers
    - ...

RViz - Visualization

- RViz
  - 3D visualization tool for ROS
  - Visualizes:
    - Robot model
    - Sensor data
    - Odometry information
    - Markers
    - ...
- Gazebo + RViz:
  - Gazebo determines how the world is
  - RViz shows how the robot perceives it

RViz


jazz.vcg - RViz
File Panels Help
Move Camera Interact Select 2D Nav Goal 2D Pose Estimate
Displays
.Global Options
Backgroun... 0, 0, 0
Fixed Frame /pico/odom
Target Fra... <Fixed Frame>
.Global Status: OK
01. Grid (Grid)
02. RobotMo...
03. LaserSca...
Color Transformer
Set the transformer to use to set the color of the points.
Add Remove Rename
Tool Properties
2D Nav Goal
Topic /move_base_sim...
2D Pose Estimate
Topic initialpose
Views
Type: Orbit Zero
Save Current Load Delete
Selection
Time
Wall Time: 1378132074.62
Wall Elapsed: 92.79
ROS Time: 1378132074.62
ROS Elapsed: 92.79
Reset

Wiki

- EMC Wiki:
  - http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control
  - Info on practical assignment, installation, getting started
  - 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
  - Log-in: student account
- Group pages on EMC Wiki:
  - Each group 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
  - Log-in: student account
- Group pages on EMC Wiki:
  - Each group 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

- Assignment: solve maze with PICO robot

Recap

- Assignment: solve maze with PICO robot
- OS: Ubuntu 12.04
- Platform: ROS
- Programming language: C++
- Code editor: Qt Creator
- Version control: SVN
- Simulation: Gazebo
- Visualization: RViz
- Documentation: Wiki

Getting Started

1. Create groups
   - 4-5 students per group
2. Check the wiki:
   - http://cstwiki.wtb.tue.nl /index.php?title=Embedded_Motion_Control
3. Add your group info to the wiki
4. Follow the instructions on the wiki:
   - Installation
   - C++ tutorials
   - ROS tutorials
   - PICO simulator tutorials

SVN-account information and tutor name will be sent to you
It is *your* responsibility to get in touch with your tutor