

Controlling Robot Manipulators

Alexis Maldonado, Andreas Fedrizzi

Lehrstuhl für Informatik IX
Technische Universität München

Sommersemester 2008

Outline of the Course

- Introduction
 - Robot-Components
 - Manipulators
 - Configuration Space
- Software
 - Player
 - Gazebo
- Position Analysis
 - Vectors in Space
 - Frames
 - Transformations
- Software
 - Roboop
 - GLrooop
- Forward and Inverse Kinematics
- Modeling of Manipulators
 - DH Parameters
- Further topics
 - Trajectory Planning
 - Modern Control Methods

Outline

1 Software

Outline of the meeting today

- Introduce Player
- Introduce Gazebo
- Install kibo package in the computer room
- Run an example program in the Gazebo simulator
- Introduction of the homework

Player

- Player is a **hardware abstraction layer** (HAL) for robotic devices, making it a kind of robot OS.
- Player defines a set of **interfaces** (defined in libplayercore/player.h), which you can use to interact with some class of **devices** (laser sensors, gps sensors, grippers, manipulators, ...).
- Player includes many **drivers**, that control a certain hardware device (like the SICK 200 laser range scanner) and provides the standard Player-interface of that device (here the “laser” interface).
- This allows to write client programs that use the laser interface without knowing what laser scanner the robot actually uses.
- Code written for one robot can be ported easily to another robot platform.

Player

- Player also provides **transport mechanisms** that allow data to be exchanged among drivers and control programs that are executed on different machines.
- Though most Player drivers directly control hardware, a number of **abstract drivers** have been developed recently. An abstract driver uses other drivers, instead of hardware, as sources for data and sinks for commands. The main purpose of abstract drivers is to store algorithms for easy reuse.
- All Player source code is distributed under GPL v2.
- The homepage with documentation regarding Player is <http://playerstage.sourceforge.net/doc/Player-2.0.0/player/>
- A nice overview of Player can be downloaded here <http://robotics.usc.edu/~boyoon/tmp/Player Tutorial.ppt> (note the blank between “Player” and “Tutorial”)

Gazebo

- Gazebo is a **multi-robot simulator** that is capable of simulating robots, sensors and objects in a **3d world**.
- It generates both **realistic sensor feedback** and physically plausible interactions between objects as it includes a simulation of **rigid-body physics** (with ODE).
- Robots and sensors can be controlled through standard Player interfaces.
- When used with Player, Gazebo provides simulates sensor data in the place of real sensor data. Player client programs cannot tell the difference between real devices and the Gazebo simulation of those devices.

Gazebo

- Gazebo is a **multi-robot simulator** that is capable of simulating robots, sensors and objects in a **3d world**.
 - It generates both **realistic sensor feedback** and physically plausible interactions between objects as it includes a simulation of **rigid-body physics** (with ODE).
 - Robots and sensors can be controlled through standard Player interfaces.
 - When used with Player, Gazebo provides simulated sensor data in the place of real sensor data. Player client programs cannot tell the difference between real devices and the Gazebo simulation of those devices.
- **This allows to test new client programs in a simulated Gazebo environment and port them to real robots if the tests are successful.**