Practical Course WS 2012

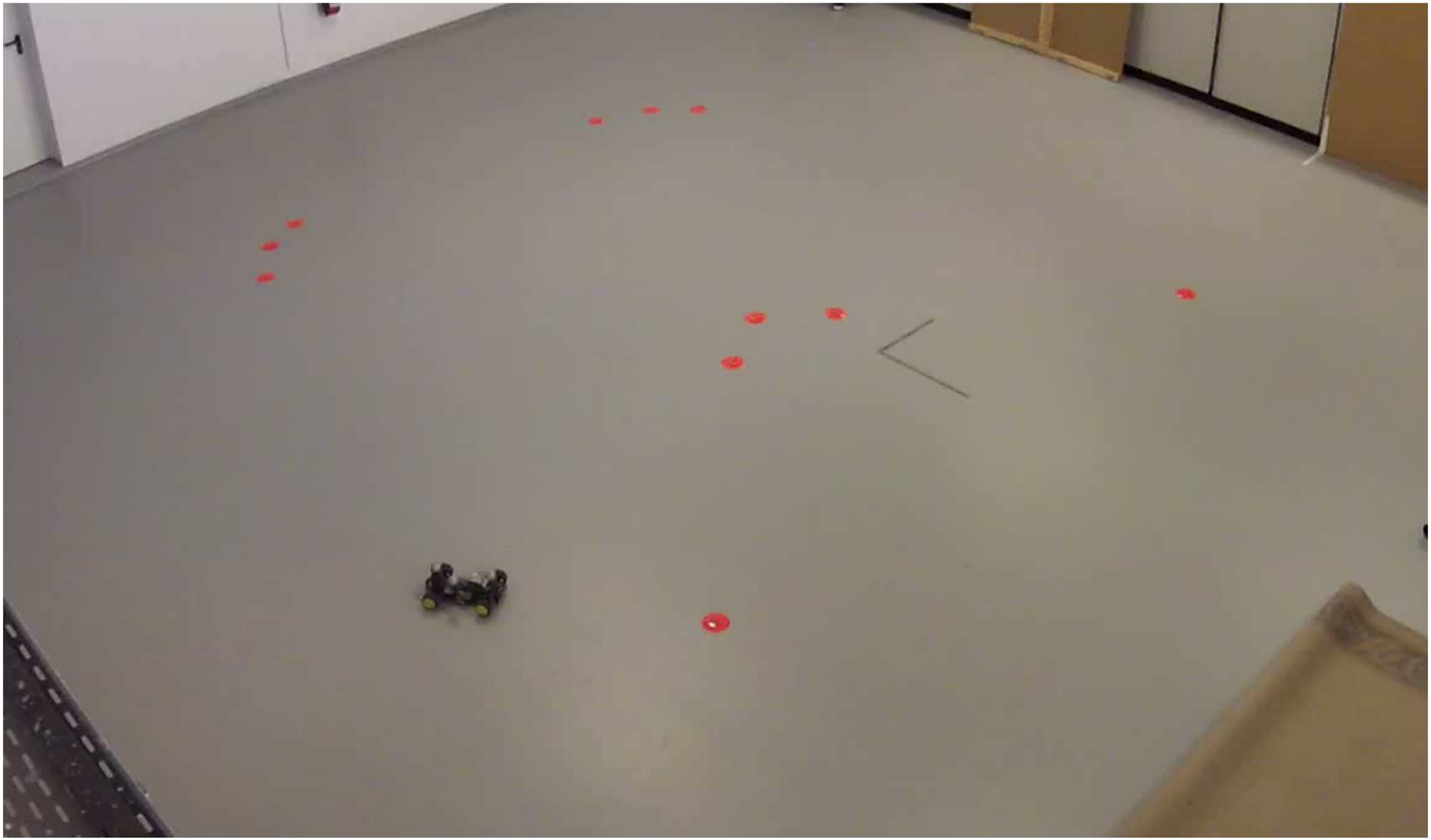
Model Car Racing

Wolfram Burgard

Jörg Müller

University of Freiburg, Germany

Topics of this Course



Topics of this Course

- Racing car control
  - What is the current state of the car, i.e., what is its current position and velocity relative to the race track?
  - What is the optimal control command given the current state of the car?
  - How to deal with high speeds and drift?
- Trajectory optimization
  - What is the optimal trajectory given the race track and its width?
- Real world racing with a R/C model car

Goal of this Course

- Hands-on development of a robotic planning and control system
- Deeper understanding of planning and control
- Practical programming experience
- Team work
- Experience in contributing to a software project

Project Structure

- Teams of three people
- Everyone has an own task/component to develop within the team
- Team members are supposed to help each other (tasks may not be equally difficult)
- Components interact via predefined interfaces

Requirements

- Programming skills are essential (C/C++)
- Ability to work in a team
- Knowledge of "Introduction to Mobile Robotics" is useful but not essential
  - Important topics are
    Robot control paradigms, wheeled locomotion, path planning and collision avoidance
- Development under Linux
  (tested with Ubuntu 12.04)
- Use of versioning with Subversion

Versioning Tool: Subversion

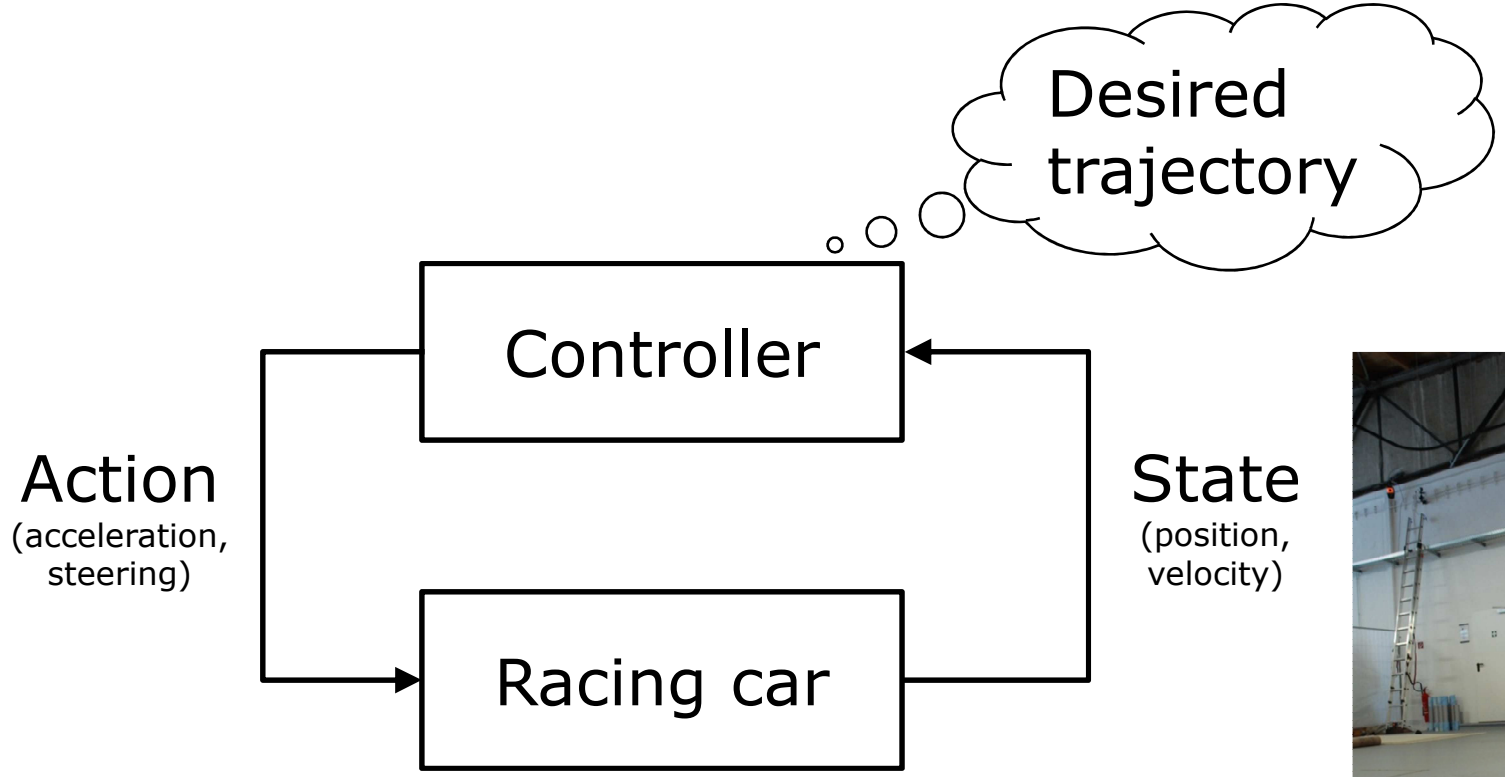
- Extremely useful for cooperative development and version tracking
- Stores every change made to the code
- Allows to go back to any intermediate revision
- Supports to merge different versions
- Inherently multi-user
- In this course, Subversion has to be used
- See the website of this course for tutorials

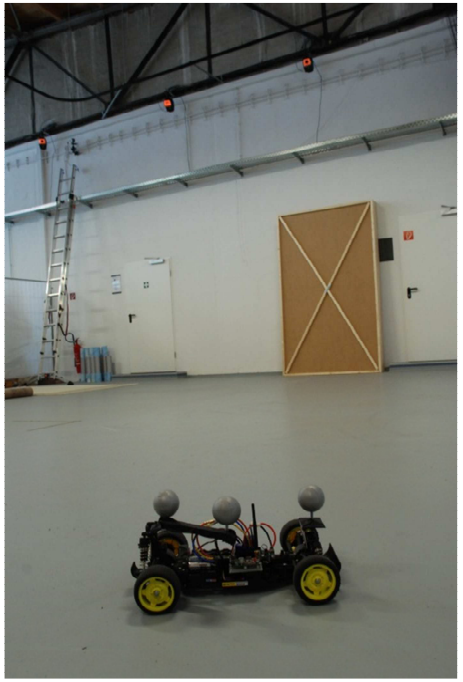
Meetings

- Weekly meetings:
  Tue 14h-16h (ct), building 101, room 01-018
- Each group has to provide a short report presentation (3-7 min) at each meeting
- Each group has to write a brief, informal summary (to be stored in the SVN)
- Each group should present its current project plan and evaluate the own progress

Topics of this Course in more Detail

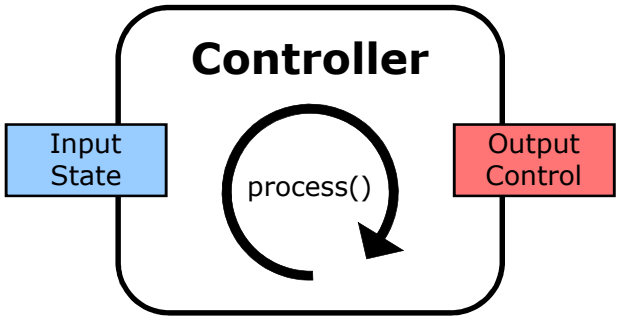
- Car racing as a control problem





Modular Software Development

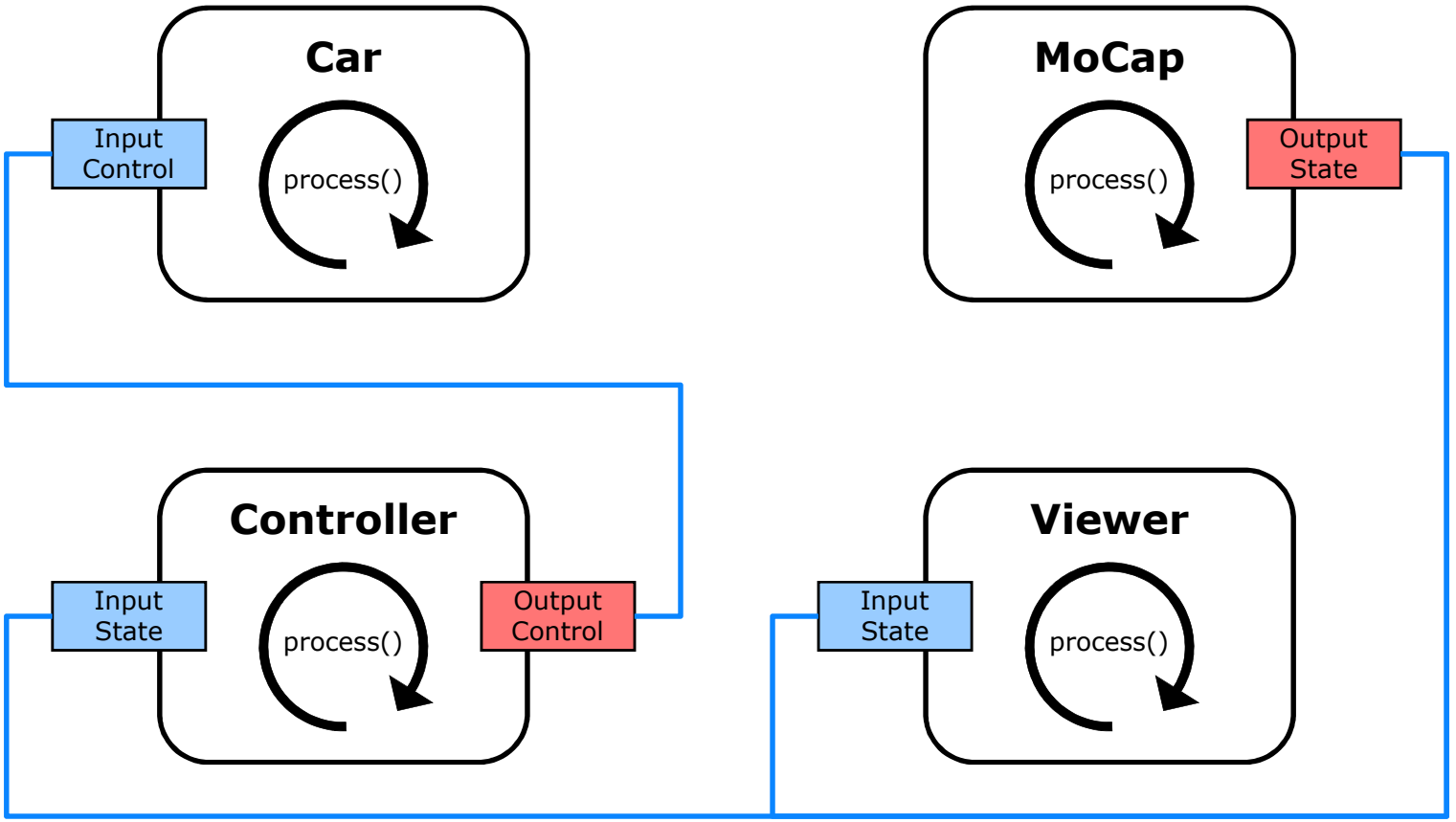
- Input/Output architecture
- Communication between modules through connection of Inputs to Outputs
- Optional Input for parameters
- Process function
  - Processes available input data and generates output data




```
Class Controller : public Module {
public:
  Input<State> inState;
  Output<Control> outControl;
  void process();
};
```

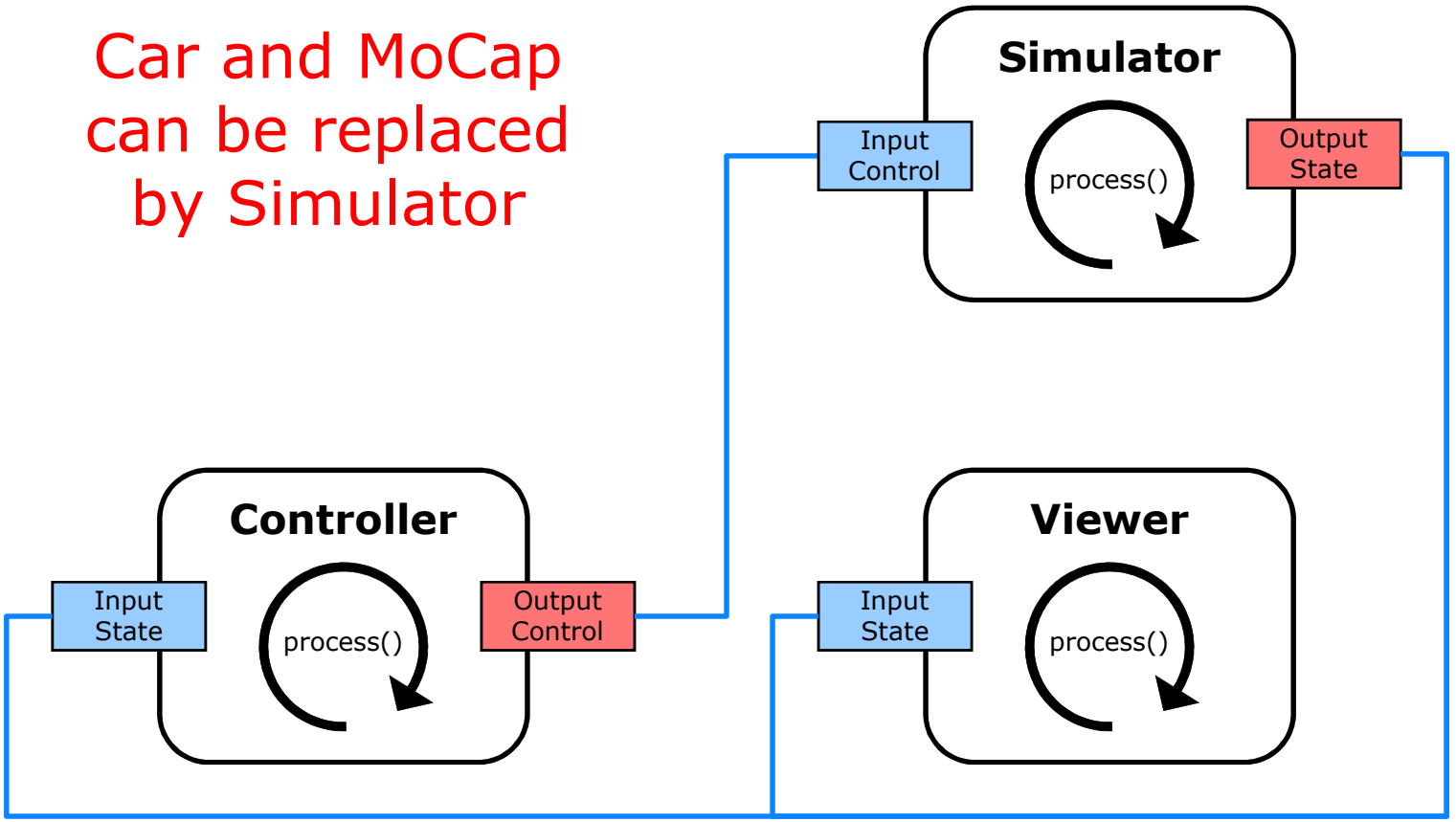
Software Architecture

- Encapsulate functionality in **modules**



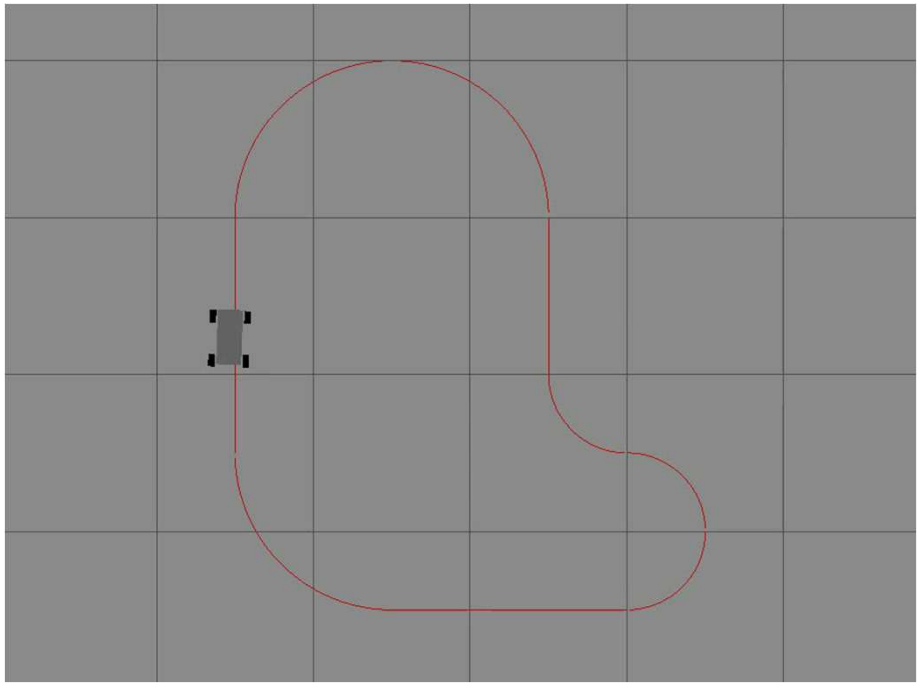
Software Architecture

- Encapsulate functionality in **modules**



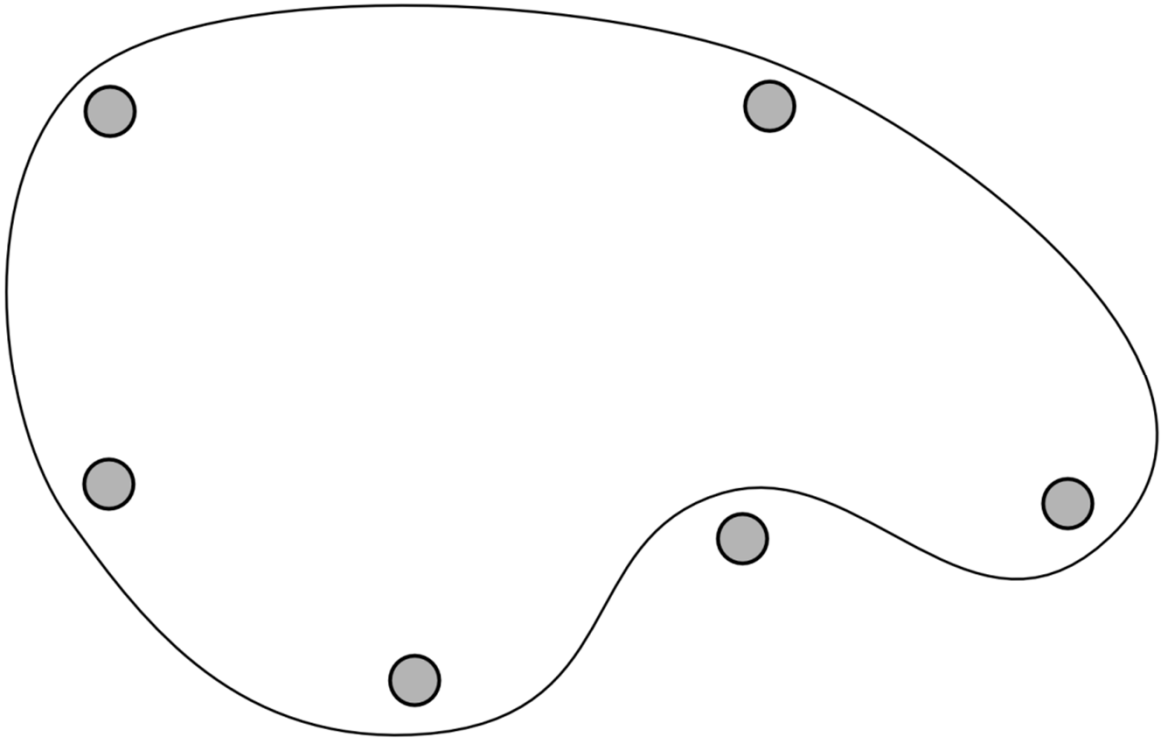
Racing Car Control

- Assignment 1: Change the controller so that it runs the car in an endless loop
- Assignment 2: Improve the speed command values in the controller module.



Topics of this Course in more Detail

- Car racing as a planning problem



Team Setup

- Find your team mates
- Decide on the task assignment
- Get familiar with your task (read!)
- Define your own milestones
- Breakdown milestones into tasks

Framework Setup

- Website: http://ais.informatik.uni-freiburg.de/teaching/ws12/practicalA/
- Download the basic profile setup
- Create the ‘code’ folder in your home directory
- Checkout and compile the library fern
  https://aissvn.informatik.uni-freiburg.de/svn/projects-fern
- Checkout and compile the robular framework
  https://aissvn.informatik.uni-freiburg.de/svn/projects-robular/branches/practical
- Read the file README in the framework for details on compiling and running the code

Repository Setup

- Set up your own repository on
  https://aissvn.informatik.uni-freiburg.de
- Copy the controller from the framework
- Create a CMake project
  - Include the headers of the framework
  - Link against the libraries of the framework

Contact

- Contact us whenever you have problems, questions, or ideas.
- Best is via E-Mail:
  muellerj@informatik.uni-freiburg.de
- Office: Building 079, ground floor
- If you have serious problems, contact us as soon as possible (the other team members depend on you).