Komplexpraktikum Mobile Roboter
Mobile Robotics Lab
Winter 2003/04

Lecture on
Robot Control Architectures and
Motion Control in Saphira

Axel Großmann
Outline

Control approaches for mobile robots

• Requirements for robot behaviours
• Requirements for robot system architectures
• Hierarchical systems
• Behavioural systems

Saphira robot architecture

• Saphira behaviours
• Action selection
• Micro-tasks
• Example application

Practical exercises
Robot Behaviours

A behaviour is a regularity in the interaction dynamics between the agent and the environment.

Example

- Obstacle avoidance

Mechanism to realise a behaviour

- Set of components (sensors, body parts, actuators)
- Control program
Requirements of a Robot Behaviour

Reactivity
  • React to sudden changes in the environment

Robustness
  • Handle imperfect inputs, unexpected events, uncertainties, malfunctions

Reliability
  • Operate without failure or performance degradation

Multiple-goal resolution
  • Responsive to high-priority goals while still servicing low-level goals

Intelligence

Centralised global reasoning
Requirements of a Robot System Architecture

Modularity
- Crucial to incremental design, maintenance, failure detection and correction

Flexibility
- Continuous design changes during the implementation phase

Expandability
- Incremental implementation of robot skills

Adaptability
- Operation in different or changing environments

Multi-sensor integration
- Compensate for limited accuracy
Hierarchical Systems

Separating robot software into functional modules for perception, reasoning, and action

Advantage

- Components controlled by single modules

Disadvantages

- Sluggishness in reacting to rapid changes
- Not very robust due to sequential processing
- Reconfiguration almost impossible
Behavioural Systems

Set of sensing, reasoning, and action loops that operate concurrently to control the robot

Decomposing the robot control problem into the robot’s behaviours

Reason about behaviour of objects

Plan changes to the world

Identify objects

Monitor changes

Build maps

Explore

Wander

Avoid objects

Sensors \[\rightarrow\] Actuators
Saphira Robot Control System

[Diagram of Saphira Client Process with layers labeled: User micro-tasks, Control routines, State reflector, Packet communications, Synchronous micro-tasking OS, User async routines, TTY or TCP/IP connection]
Motion Control in Saphira

Direct control

- monolithic controller

Behavioural control

- set of behavioural actions
- resolver to produce motion
- indirect or partitioned way of controlling robot motion
Instructions for Accessing Saphira (1)

Login at one of our machines:

    ssh mrl03@spock.inf.tu-dresden.de

The first time you login, create your personal project directory using:

    setup-dir YOUR-FRZ-USER-NAME

Right after logging in, setup your working environment using:

    setup-env YOUR-FRZ-USER-NAME

Enter your personal working directory:

    cd $MRLUSER/prj/Giovanna/
Recompile source files if necessary:

    make clean dep all install

Compile and run Saphira demo:

    cd $MRLUSER/prj/Giovanna/sapphira/demo/
    make clean dep all
    ./sapphira.$OSTYPE.bin

Start the Saphira simulator:

    gisimulator

And explore the system ...
Practical Exercises Using Saphira

Compiling and Debugging in Saphira

- Compiling and linking C/C++ source files
- Writing C or C++ client programs
- Debugging C code under UNIX

Examples of direct motion control

Examples of behavioural control

- Behaviour parameters and their effect on the robot
Homework

Programming task

- Place to poles 2 metres apart from each other in the corridor and make the robot go around these poles

Next lecture on Monday, 17th November 2003