Mobile robotics
Control systems

Janusz Jakubiak

Wrocław, 22.12.2015

Copyright information

These slides contents base on a book Springer Handbook on Robotics (Eds. B. Siciliano, O. Khatib)

Robot locomotion task

Definition

Determining drives control which will:

- reach the desired goal
- include drive constraints in the plan
- include limited/uncertain information about robot surrounding
- optimize selected task execution criterion

Robot locomotion task

Definition

Determining drives control which will:

- reach the desired goal
- include drive constraints in the plan
- include limited/uncertain information about robot surrounding
- optimize selected task execution criterion

Typical goals

- reaching position
- reaching pose (position, orientation, configuration)
- following a path
- tracking a trajectory

Robot locomotion task

Autonomy of locomotion

- ability to reach a goal without operator intervention
- ability to react to changes in environment
- robustness to sensing errors
- real-time operation

Robot locomotion task

Autonomy of locomotion

- ability to reach a goal without operator intervention
- ability to react to changes in environment
- robustness to sensing errors
- real-time operation

Locomotion types

- Walking: statically stable, dynamic
- Wheeled: holonomic, nonholonomic
- Other: crawling, climbing, jumping, flying, floating

Control architectures

- reactive
- deliberative
- hybrid
- behavioral

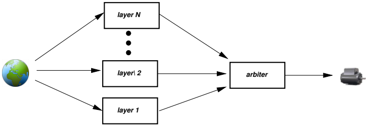
Reactive methods

- biologically inspired, stimulus-reaction scheme
- none (or almost none) world model, no complex reasoning methods
- none or very little memory
- usually in form of a set of simple rules (if you hit an obstacle – stop, if you are stopped – rotate)

Reactive methods

- biologically inspired, stimulus-reaction scheme
- none (or almost none) world model, no complex reasoning methods
- none or very little memory
- usually in form of a set of simple rules (if you hit an obstacle – stop, if you are stopped – rotate)

- fast execution
- little capapilities to store information
- applicable to dynamic, unknown environments

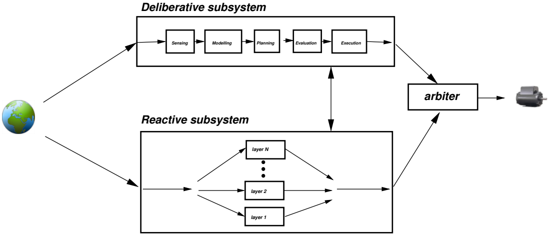


Deliberative methods

- using complex planning – computationally expensive
- internal, symbolic world representation (maps, tasks, memory) required to evaluate effects of action – needs storage and processing
- with sufficient time – allows strategic planning and task optimization
- effective in static, controlled environments

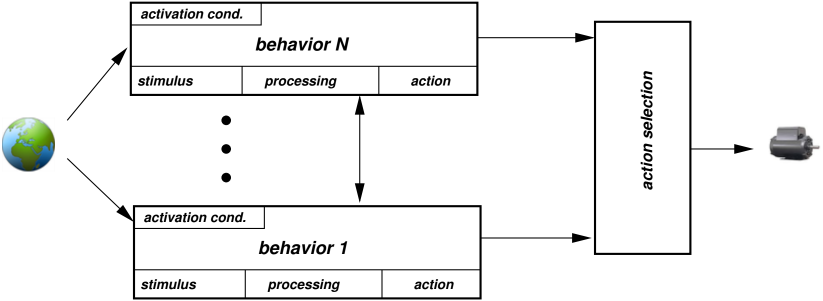
Hybrid architecture

- combination reactive and deliberative subsystems
- reactive subsystem is responsible for “basic needs” (collision avoidance, safety system, etc.)
- deliberative subsystem – tasks which are not time critical (global path planning, optimization)
- requires arbitration to solve contradictory orders
- for full use of subsystems advantages subsystems should communicate and coordinate tasks



Behavioral methods

- evolved from reactive methods
- each behavior solves an own task based on sensors and *other behaviors' outputs*
- behaviors may store complex information (including system history) what allows learning
- systems may use dynamic interactions between behaviors



Applications

- reactive: fast changing environments which require immediate actions; no learning and prediction capabilities
- deliberative: simple environments with small, predictable changes; provide optimal and fully predictive actions, allows strategic plans
- hybrid: environments which require relatively complex planning with world model, but with some actions independent from central planner
- behavioral: environments with relatively high variations which require fast adaptation, but also some planning and learning