OC for Self-Optimization and Anomaly Handling in Technical Systems

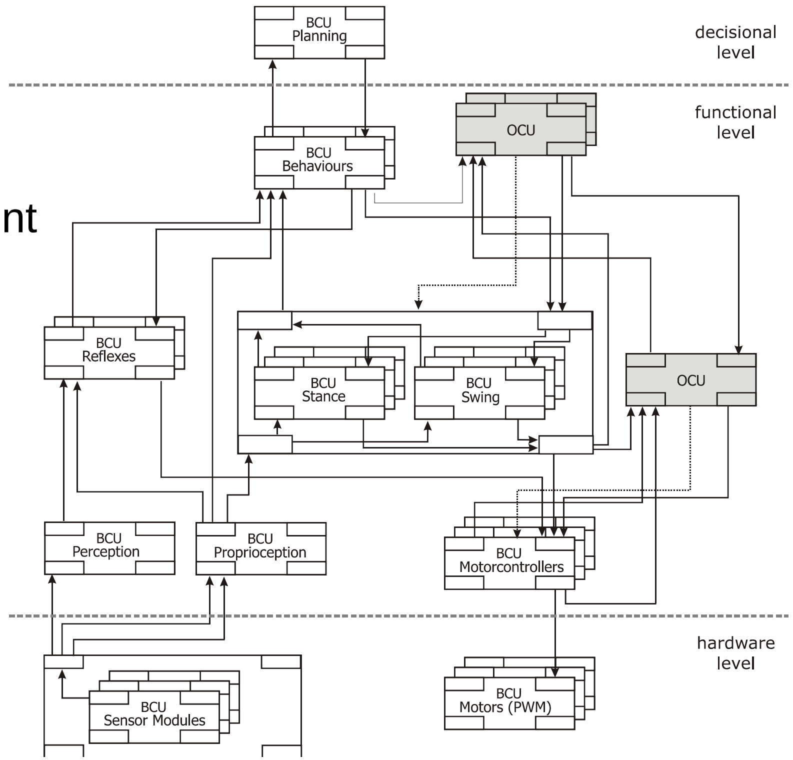
ORCA Project

W. Brockmann, E. Maehle,
S. Krannich, N. Rosemann

OC Colloquium, Hannover, Feb. 2010

OC-Background

- Use OC (self-optimization, self-adaptation, self-organization) to master complexity in technical systems
  - Complexity of engineering
  - Complexity of interaction with the environment
  - Move development time to runtime
  - **But ensure safety and trustworthiness**



- Our architectures and methods are mainly investigated in robotic applications

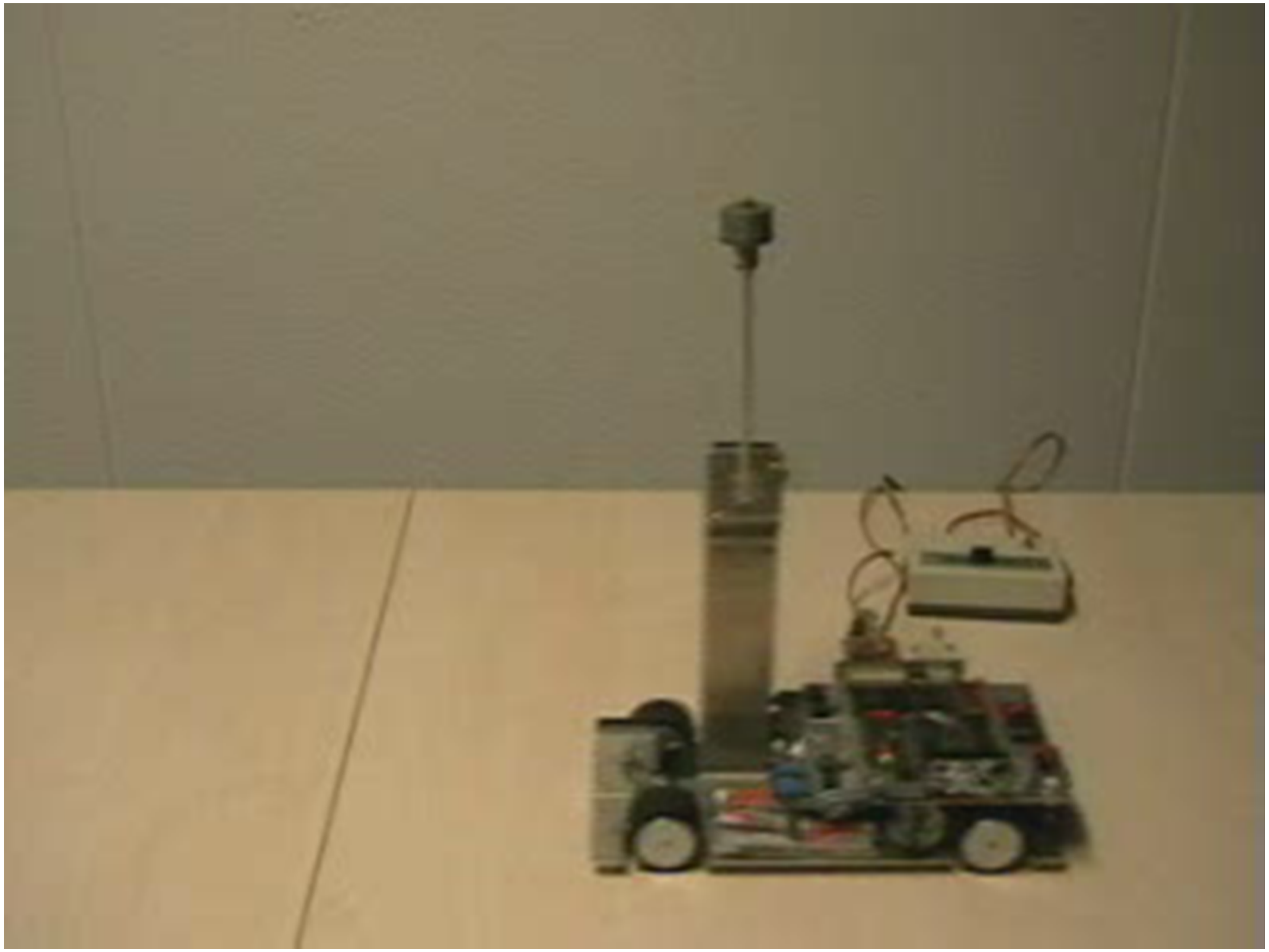
Applications - 1

- Use self-optimization to master complexity in control applications
  - Complexity of engineering
  - Complexity of interaction with the environment
  - Move development time to runtime
  - **But ensure safety and trustworthiness**

- Design of reliably self-optimizing control for real inverted pendulum cart
  - Formal modeling hardly possible
  - Slippage, gear backlash, user interaction
  - Self-optimization
  - **Controlled self-optimization by OC, e.g., the SILKE approach**

Additional O/C-loop to detect and counteract dynamically emerging anomalies/patterns within the self-optimizing system

Applications - 1 (Video)



Applications - 2

- Use self-adaptation to master uncertainty in decision systems
  - Complexity of engineering
  - Complexity of interaction with the environment
  - Move development time to runtime
  - **But ensure safety and trustworthiness**

- Design of high speed classification system with uncertain sensors
  - Formal modeling hardly possible
  - Sensor failures, user interaction
  - Process learned knowledge based on dynamically changing trust signals
  - **Robustness without classical fault tolerance**

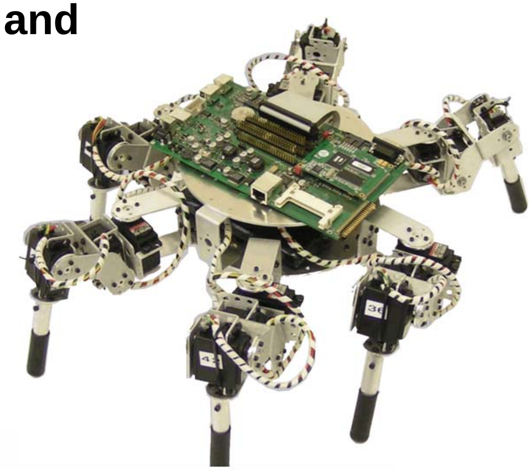
Trust signals

Add. input to classification algorithm in order to protect it from applying learned knowledge in a wrong way and from corruption

Applications - 2 (Video)

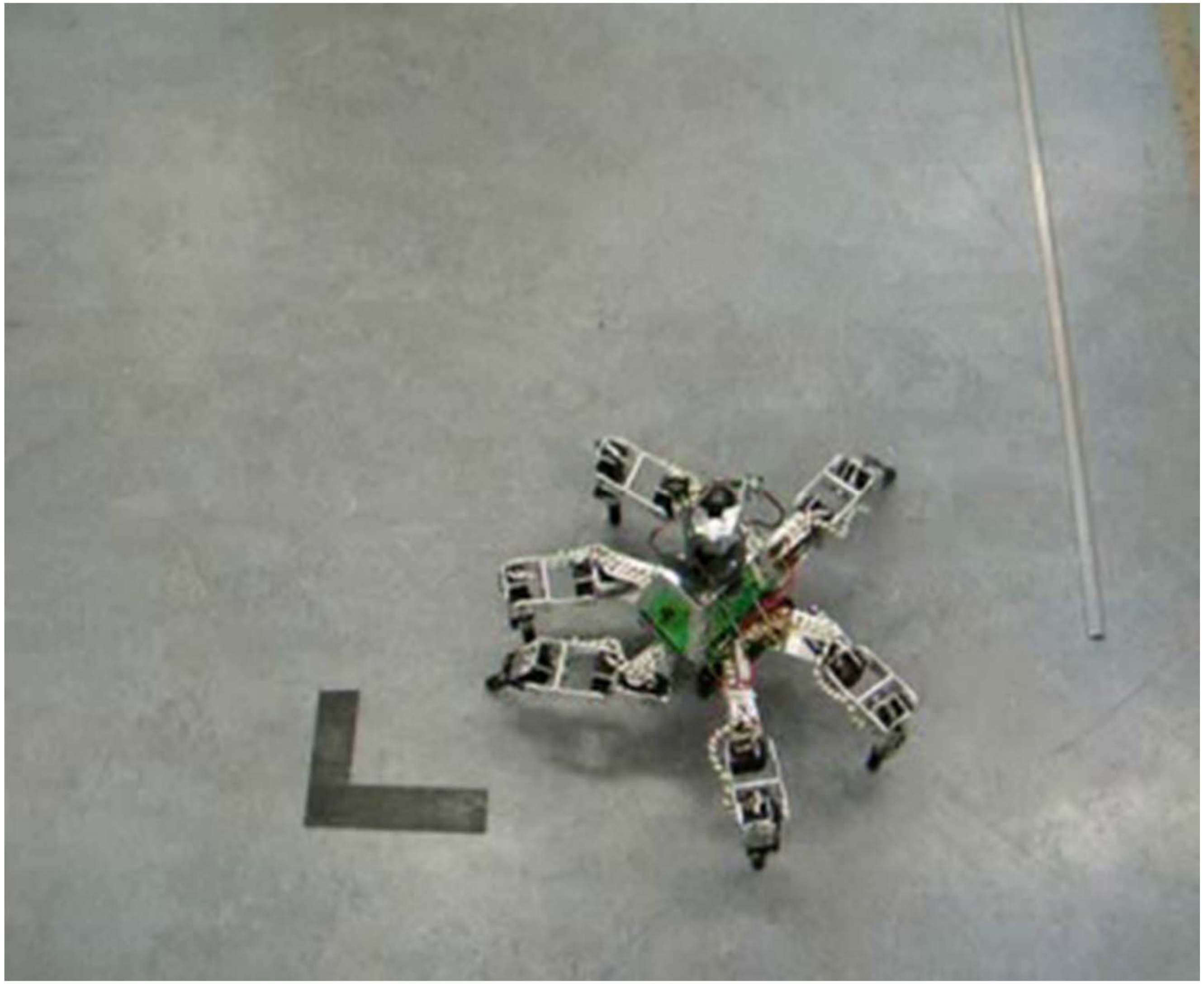
Applications - 3

- Use self-organization to master complexity in robotic systems
  - Complexity of engineering
  - Complexity of interaction with the environment
  - Move development time to runtime
  - **But ensure safety and trustworthiness**
- Design of robust six-legged walking robot OSCAR
  - Formal modeling hardly possible
  - Actor (leg) failures, environment
  - Self-organization for leg coordination and higher behaviors
  - **Appropriate gait patterns emerge dynamically even with lost legs**

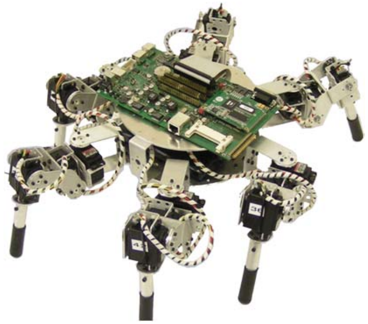


The combination of decentralized control and an actively complying reflex of the alpha joint allows OSCAR to show an emergent gait pattern.
Lost legs can be compensated by a self-organizing leg reconfiguration

Applications - 3 (Video)



Advantages over non-OC techniques



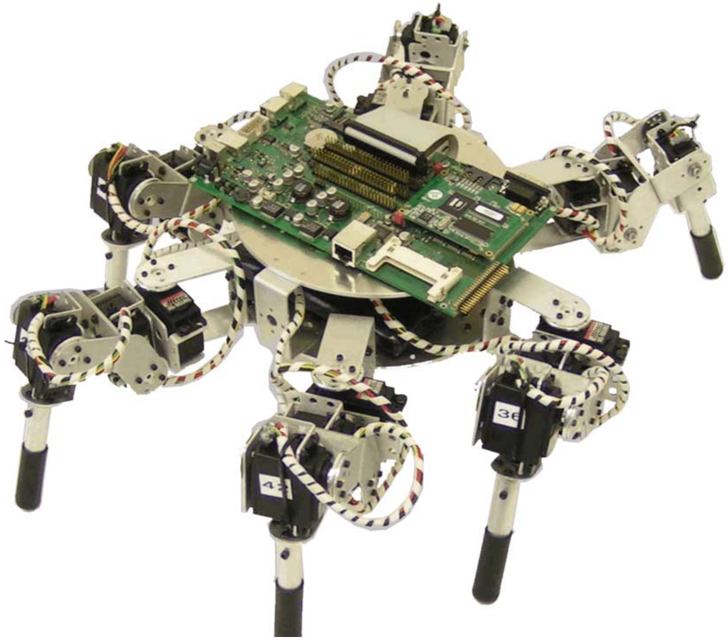
- Shorter engineering time, no formal modeling required
  - For continuous applications, e.g., intelligent control
  - For discrete applications, e.g., gait pattern adaptation
- Lower computational overhead for representing uncertainties
  - For classification systems
  - For anomaly handling in robotic and automation applications
- Higher trust in spite of high flexibility
  - Built-in countermeasures for dynamically emerging anomalous patterns in self-optimizing systems
  - Meta-level control of self-optimization

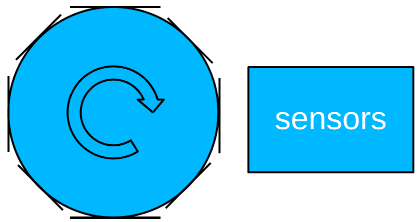
Thanks for your attention

Joint Demonstration Scenario

- Goal for phase 3 is to integrate the different OC-techniques on OSCAR (as an example for complex, hybrid systems)

- Application scenario: environmental monitoring in difficult terrain (obstacles, uneven ground) with unforeseen faults




sensors