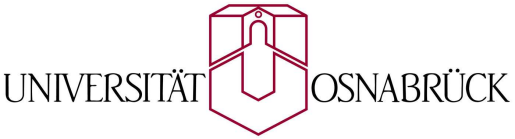
ORCA – Organic Robot Control Architecture

Universität zu Lübeck
Institut für Technische Informatik

Erik Maehle
Marek Litza



Institut für Informatik

Werner Brockmann

Karl-Erwin Großpietsch

Institut
Intelligente Analyse- und
Informationssysteme

Overview

- **Motivation**
- **ORCA-Architecture**
- **Demonstrator OSCAR**
- **Adaptive Filters**
- **Health-Signal**
- **Learning**
- **SILKE-Approach**
- **Outlook**

Motivation

Autonomous mobile robots in human environments

unstructured, dynamically changing environment

no explicit model of the environment

-> fault-tolerance, safety

no explicit fault model

complex control systems

-> engineering bottleneck

Organic Computing can help

ORCA - Organic Robot Control Architecture

Goals:

- Self-organization: adapt to malfunctions without a formal model
- Learning: online and in-situ under hard real-time constraints
- Safety: avoid critical system states at any time
- Goal-Directedness: stable improvement of behavior
- Low Cost: overhead should be as low as possible

Approach: **controlled emergence** on lower functional control levels

ORCA - Organic Robot Control Architecture

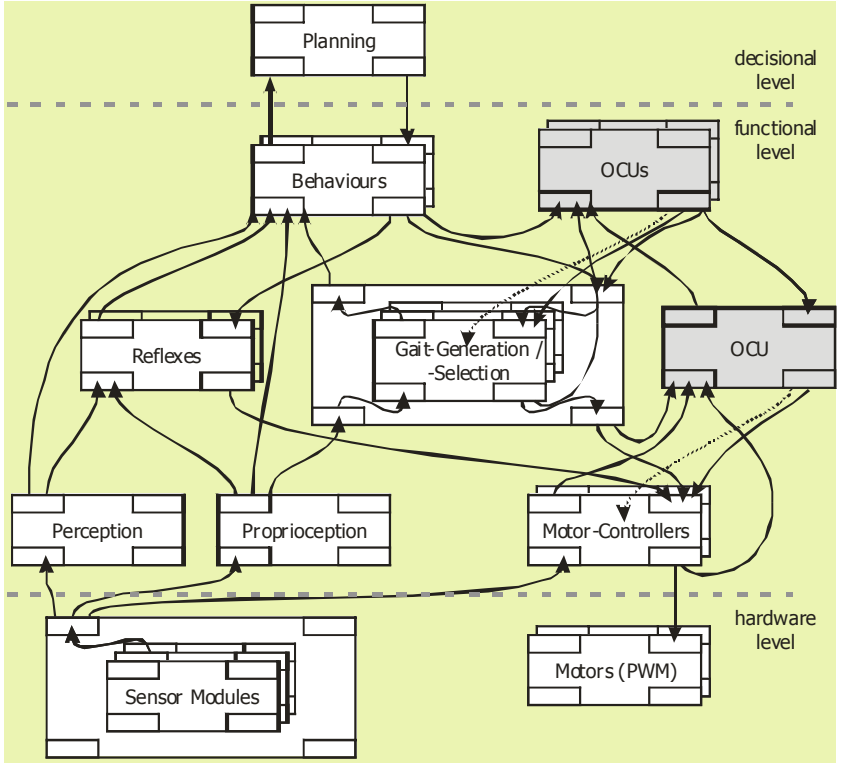
Hierarchic architecture

Components are selfdescribing

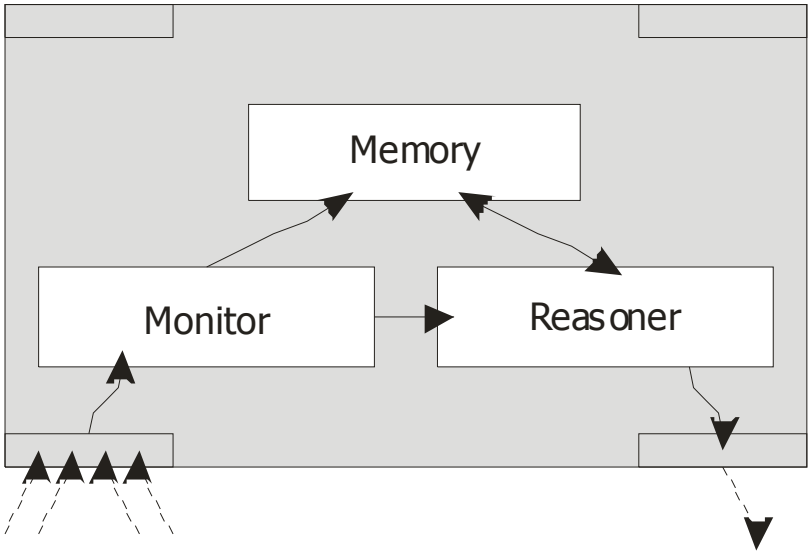
Within one level components combine each other autonomously (like p2p)

Components:

- BCU = Basic Control Unit
- OCU = Organic Control Unit



OCU



- Monitor: anomaly detection
- Memory: short term history
- Reasoner: hard real-time determination of a counteraction

OSCAR
Organic Self-Configuring and Adapting Robot

Hexapod with 18 DOF
(Servos)

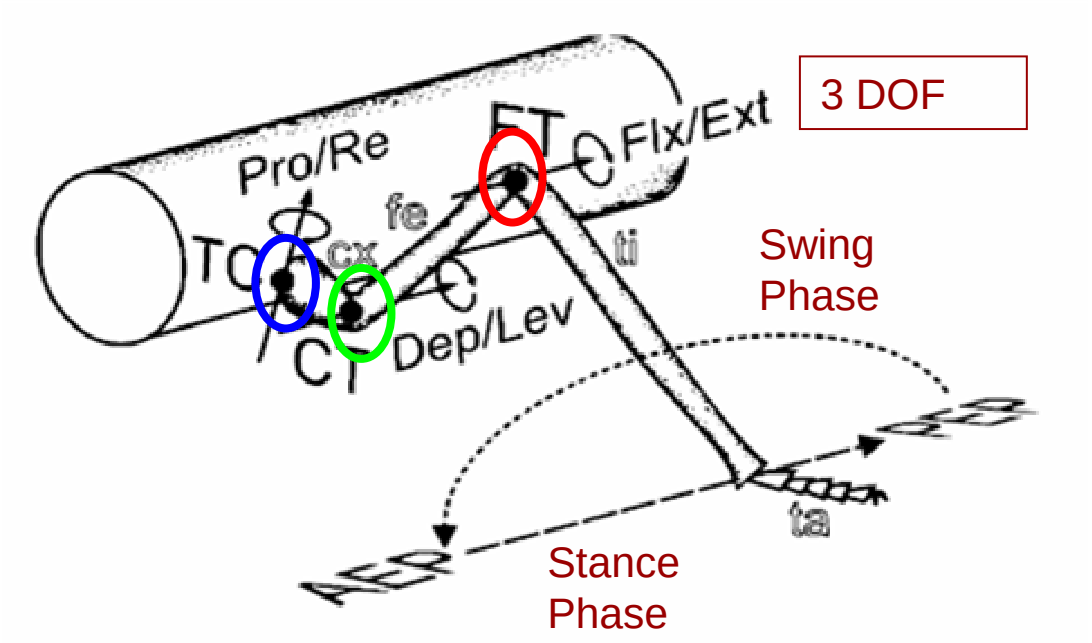
Ground contact Sensor

Control Computer:

- JControl/Smartdisplay
- Servo Driver Modul
- I2C-Bus
- PC (BlueTooth)

Programming Language:
JAVA

Basic Approaches for Insect Walking

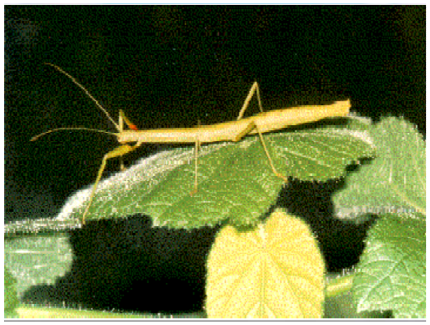


3 Joints: α,β,γ

AEP: Anterior Extreme Position
PEP: Posterior Exreme Position

Distributed Walking Control in OSCAR

Insects produce **global** walking patterns with **local** coordination rules. [Cruse et. al. 90]

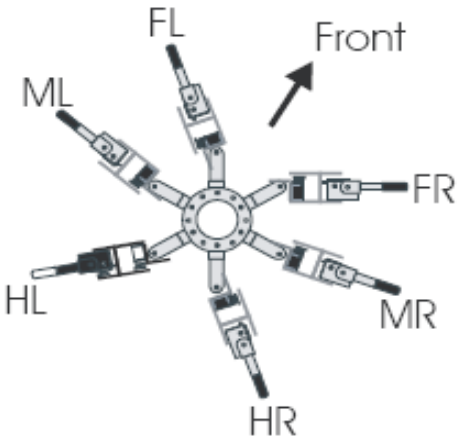


```
if (leg[i-1].groundContact() && (leg[i+1].groundContact()){
            leg[i].swing();
}
else
            leg[i].stance();
}
```

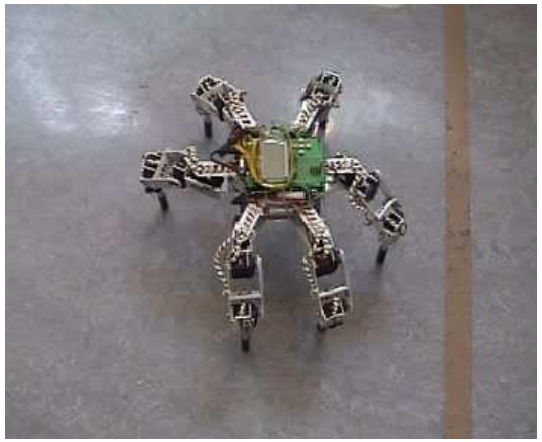
So far only one single rule implemented with circular neighborhood relation for all legs.

Swing phase has constant length.

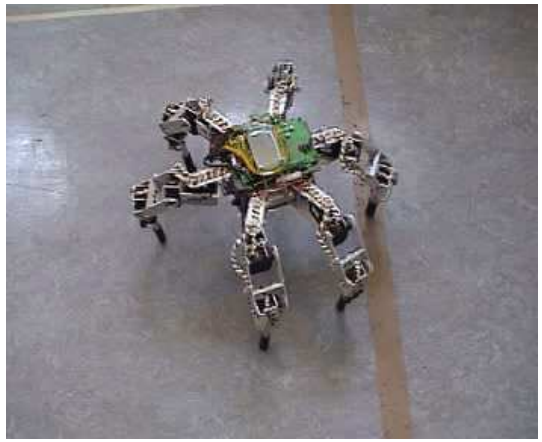
Duration of stance phase determines velocity.



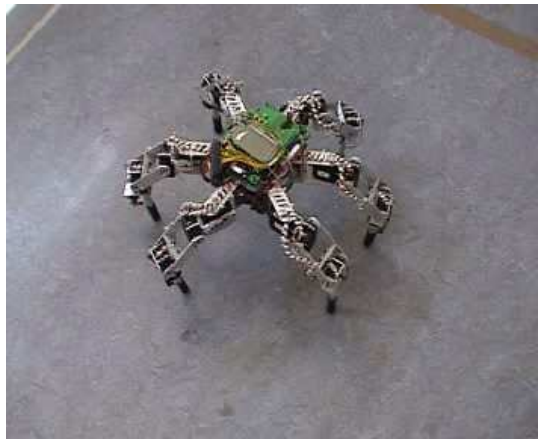
Self-Organizing Gait Patterns in OSCAR



Slow Gait



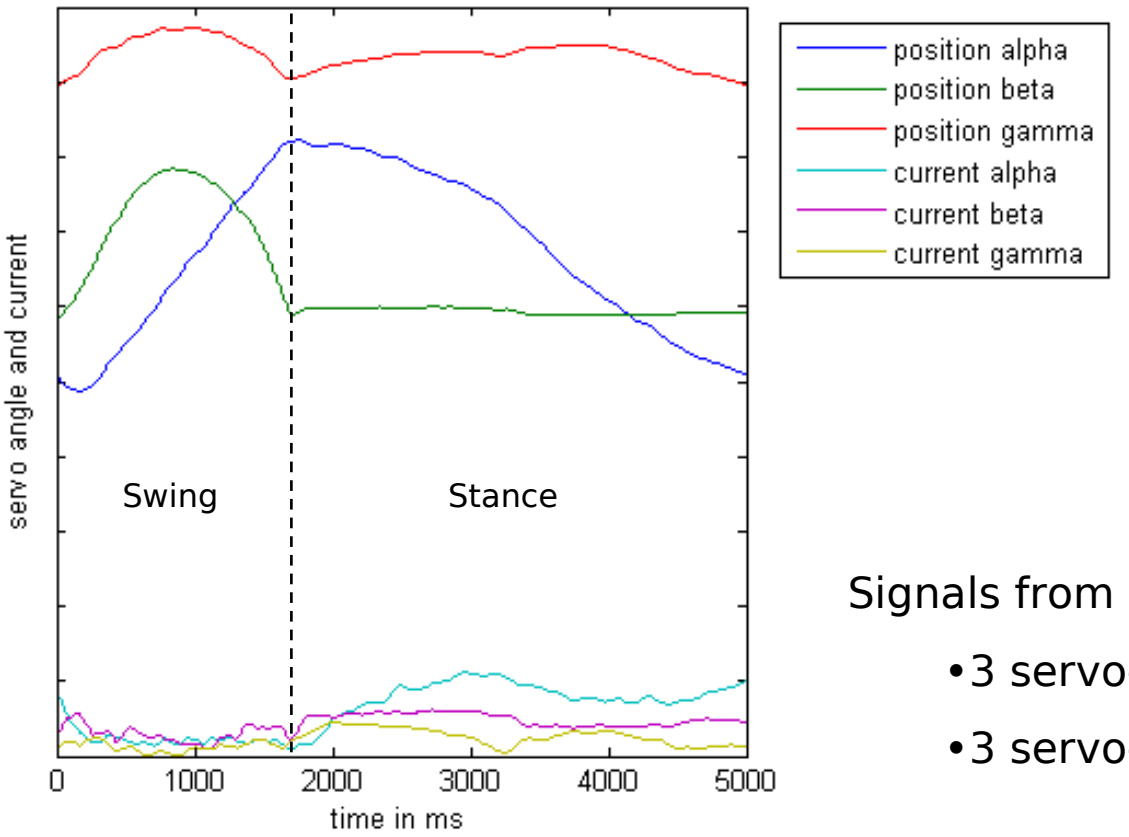
Tetrapod Gait



Tripod Gait

Emergent Gaits with Increasing Speed!

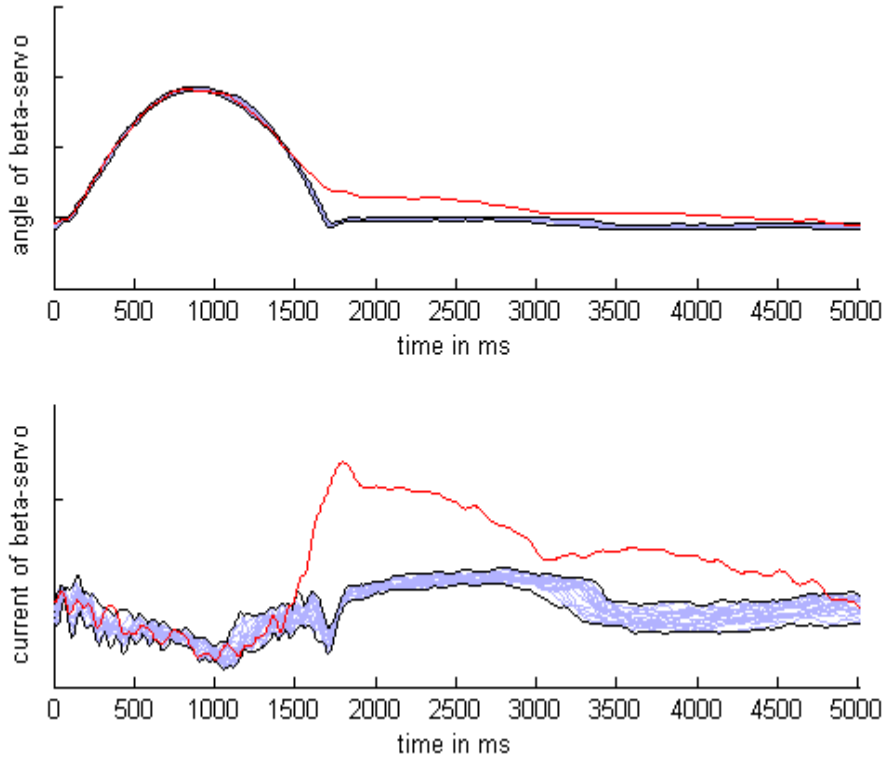
Monitoring of one Leg



Signals from one Leg:

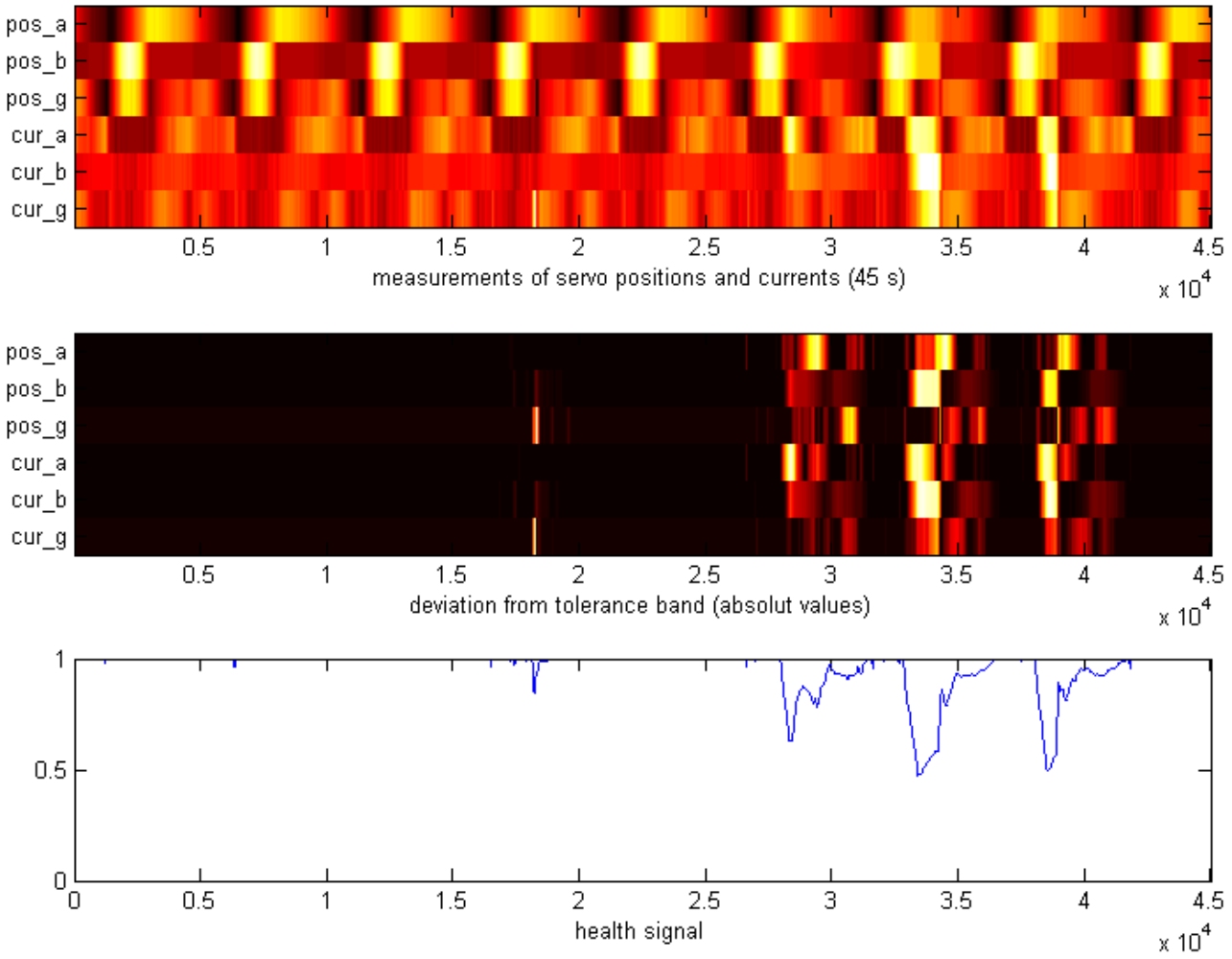
- 3 servo-positions (α,β,γ)
- 3 servo-currents (α,β,γ)

Monitoring – tolerance band

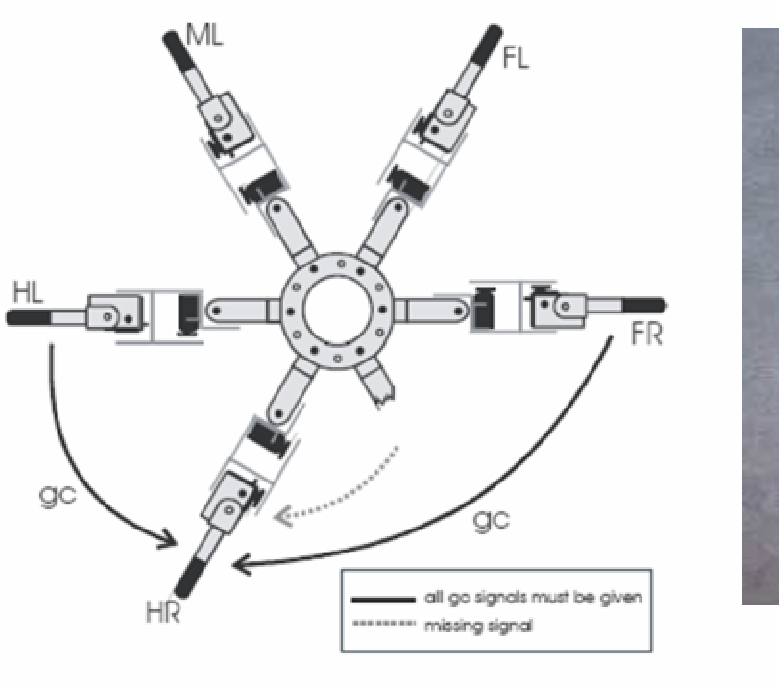


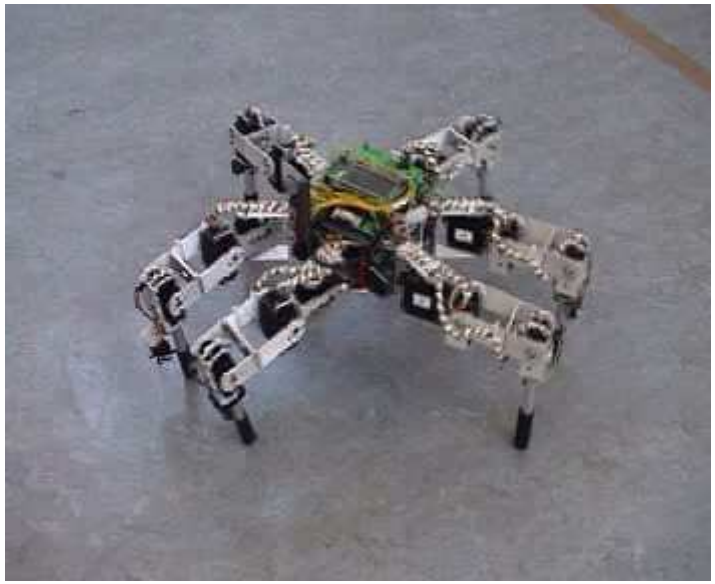
beta-servo:
position and current

Monitoring - Health-Signal



OSCAR Walking with Lost Middle Leg





Same single rule, only
neighborhood relation changes.

Treatment of Local Component Faults in OSCAR by Means of Adaptive Filters

At FhG-IAIS, component faults in OSCAR and their effect on its trajectories were exhaustively simulated, using the robot simulator Gazebo.

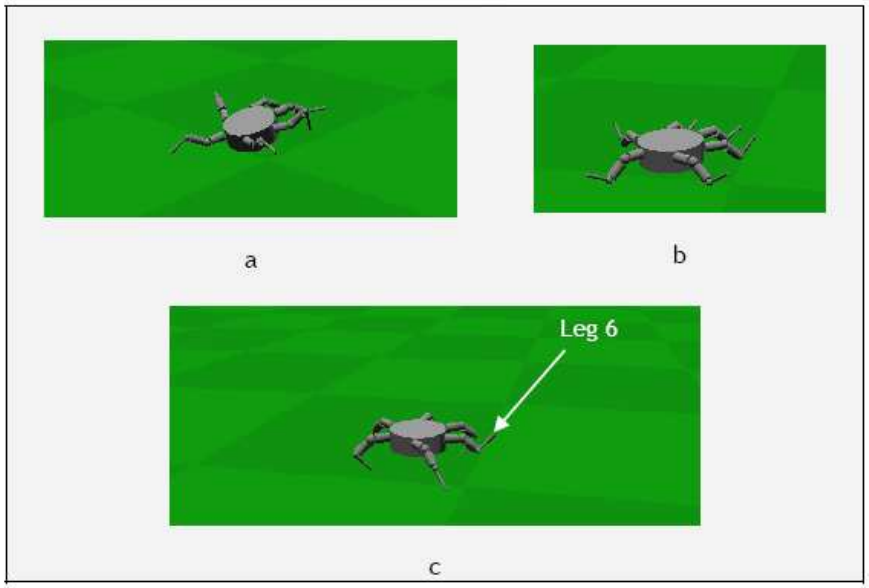
For compensating fault effects, the components as e.g. joints and servo motors are controlled by means of adaptive filter algorithms, i.e output variables y are estimated by weighted sums of successively sensed values of a control variable x:

$$y(n) = \sum_{i=0}^{N-1} w_i(n) * x(n-i)$$

The weights w_i are trained to memorize the experience to compensate deviations of y.

The algorithms dealing with the fault effects inside a roboter leg were integrated to form a homogeneous Leg Fault Treatment Layer, i.e this layer can autonomously compensate the different fault effects inside a robot leg.

Compensation of Fault Effect



OSCAR getting healed after suffering from dislocated joints during motion

(a) OSCAR falling down while walking.

(b) Beta-joints healed. Controller will now heal gamma-joints.

(c) OSCAR's pose after correction. Note that all legs are properly oriented except leg 6. This is because gamma-joint of this leg was already off the ground and therefore the controller skipped healing it.

´Health-Signal´

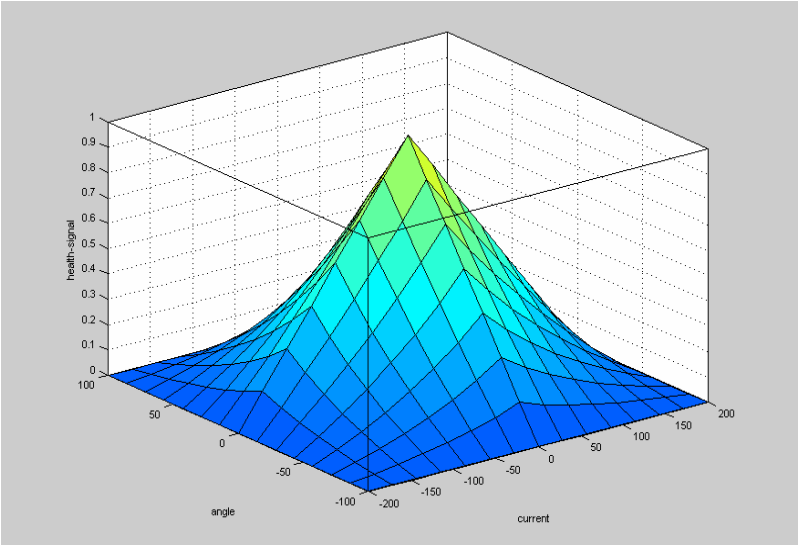
Specific aim: rate anomalies, e.g. irregular sensor or behaviour patterns

- at any system level (signals, modules, system itself)
- without a formal model
- potentially capable of learning

Approach:

- fusion and abstraction of data, states and module activities
- gradual severity index: 1 = completely normal
  0 = totally abnormal behavior

-> (neuro-) **fuzzy-system** to generate the health-signals



Example:

Health-signal versus current and angle errors

Demonstrator CARL

(Creeping Autonomous Robot for Learning demonstrations)

Motivation:

- generalization of investigations on OSCAR
- simpler, but scalable testbed for
  - architectural issues
  - health-signal
  - emergent behavior
  - online learning
  - self-organization

Current state:

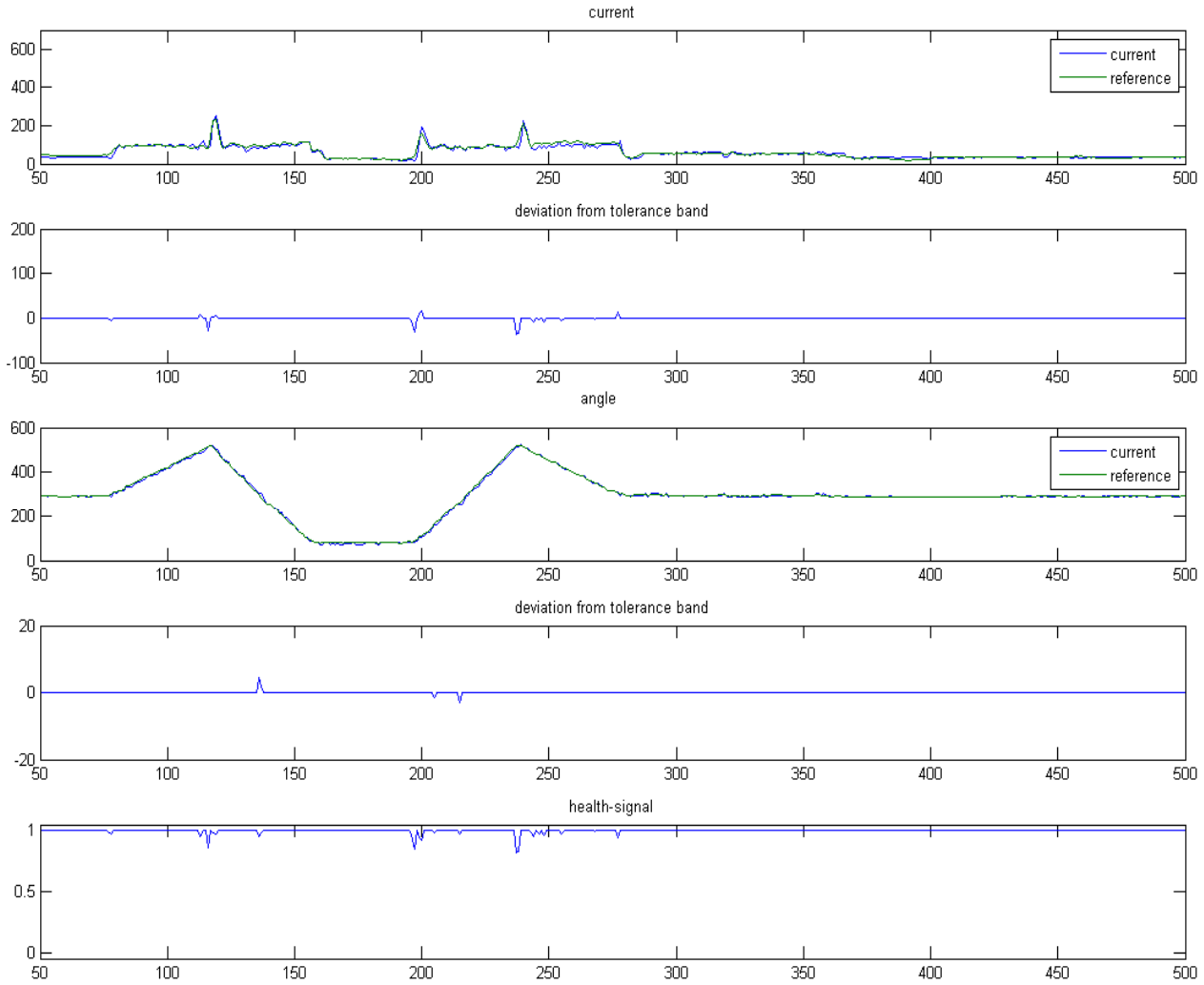
- health-signal
- emergent global behavior out of local rules
- first steps of learning

Results

Motor Current:

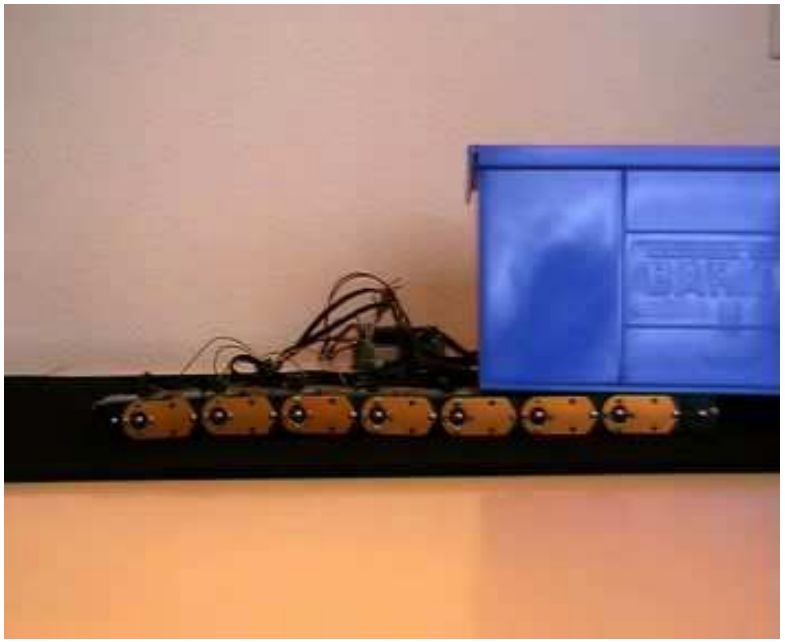
Joint Angle:

Health-Signal:

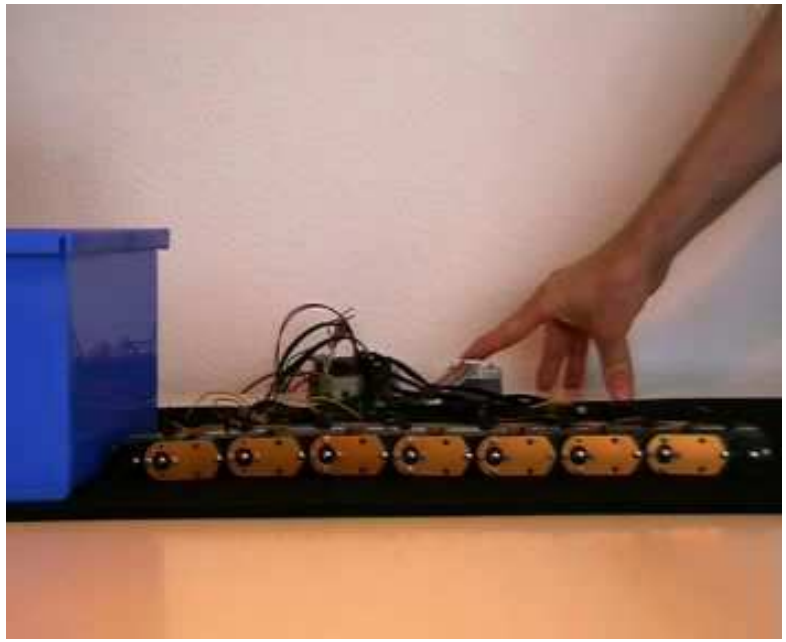


Normal behavior of CARL

Test Scenarios



Blocked upward movement



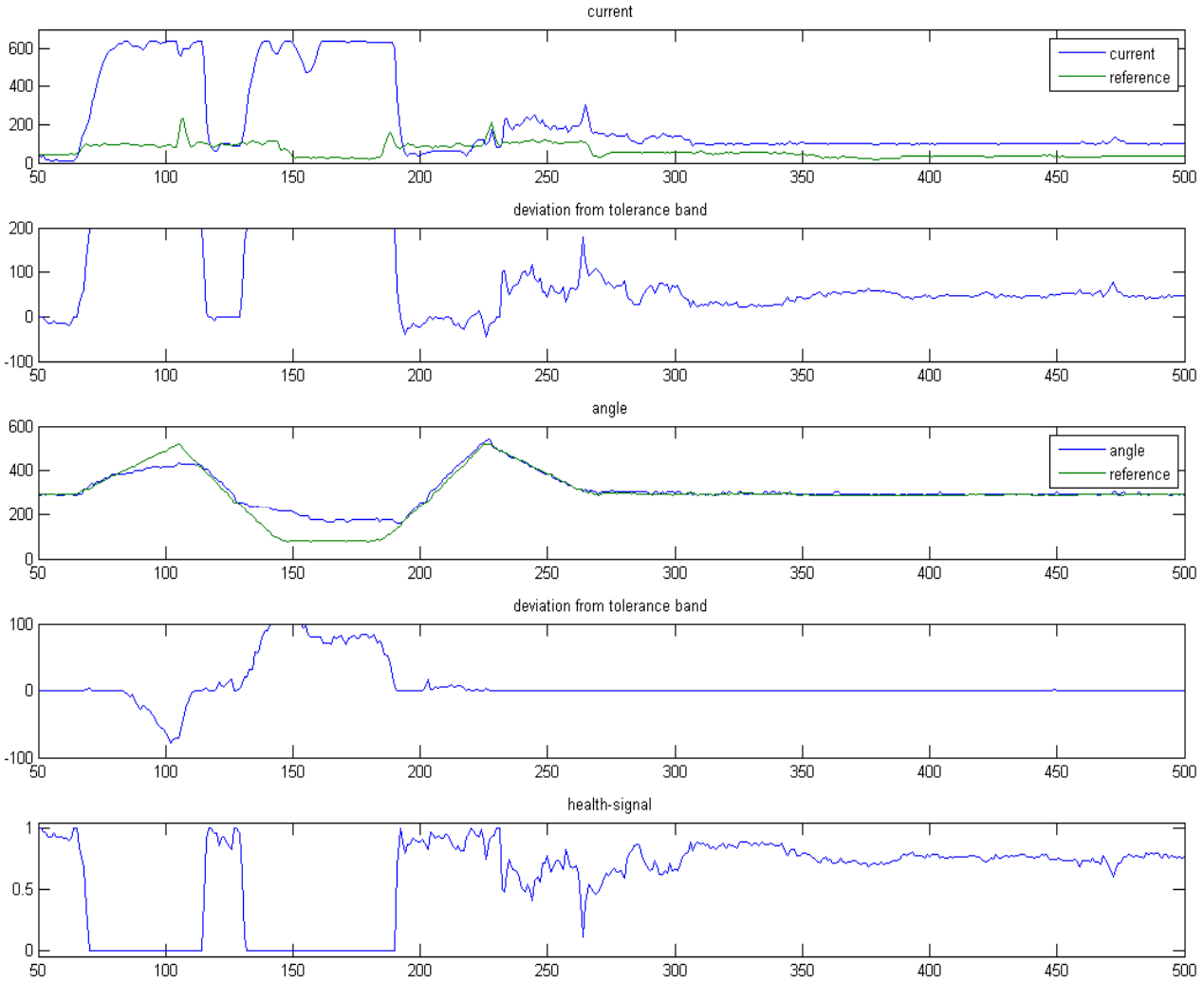
Blocked forward movement

Results

Motor Current:

Joint Angle:

Health-Signal:



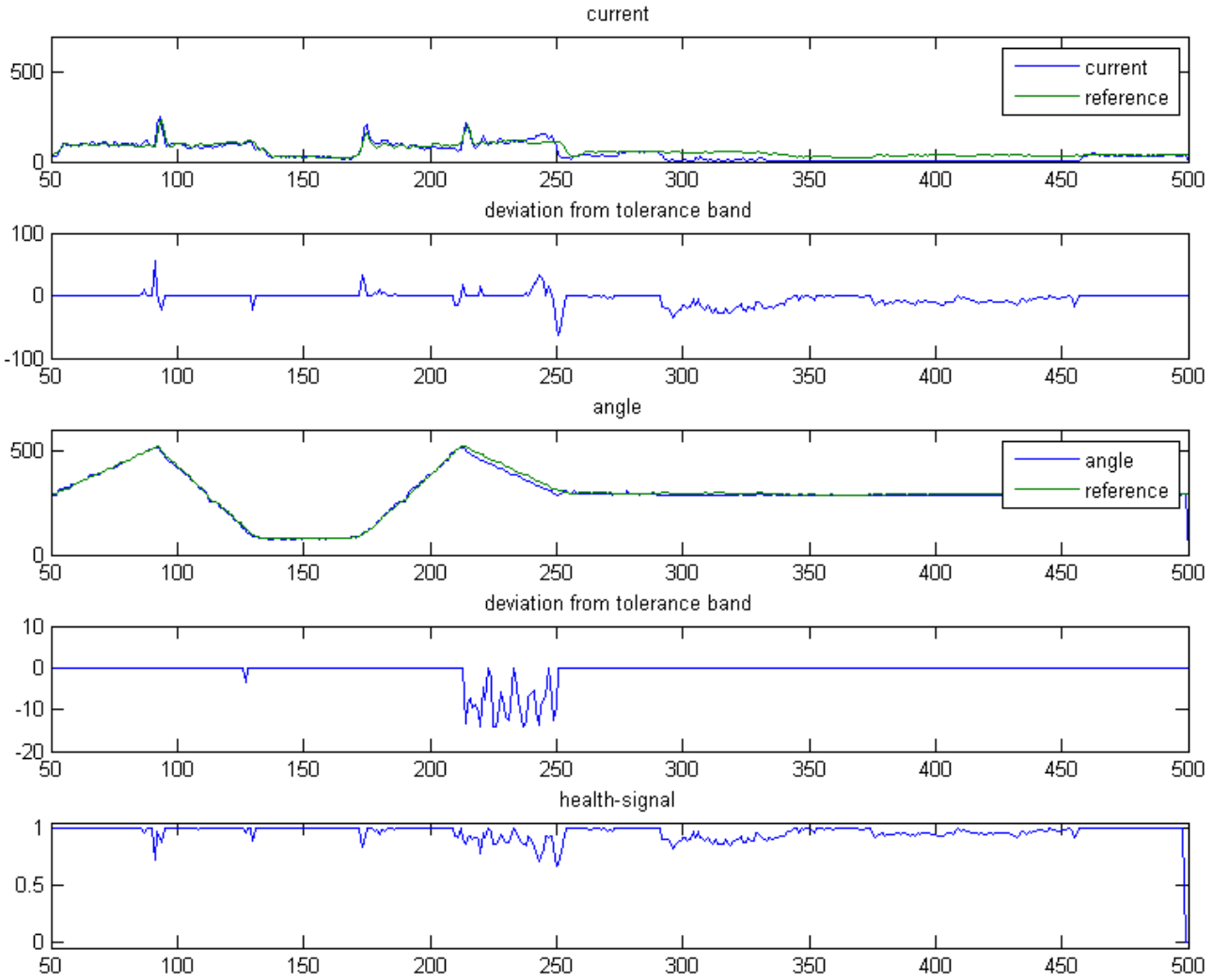
Blocked upward movement

Results

Motor Current:

Joint Angle:

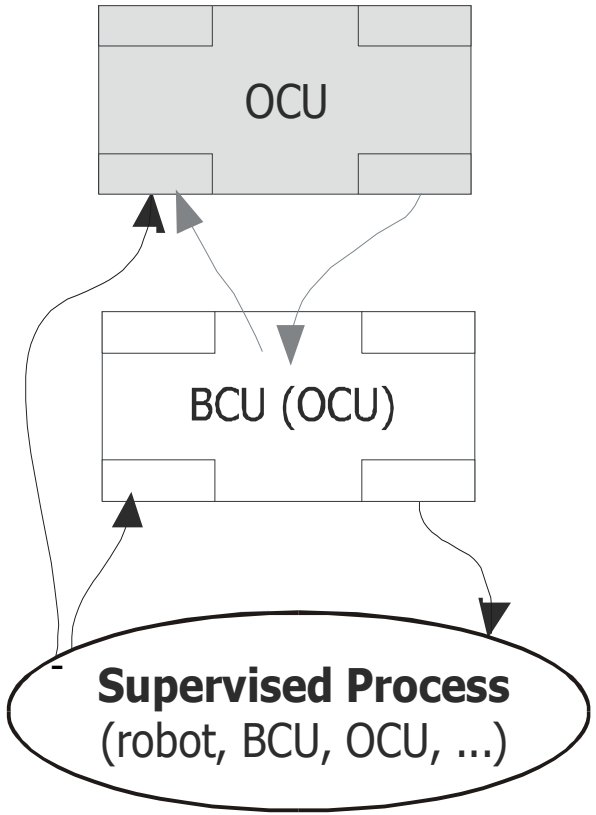
Health-Signal:



Blocked forward movement

Learning-Basis

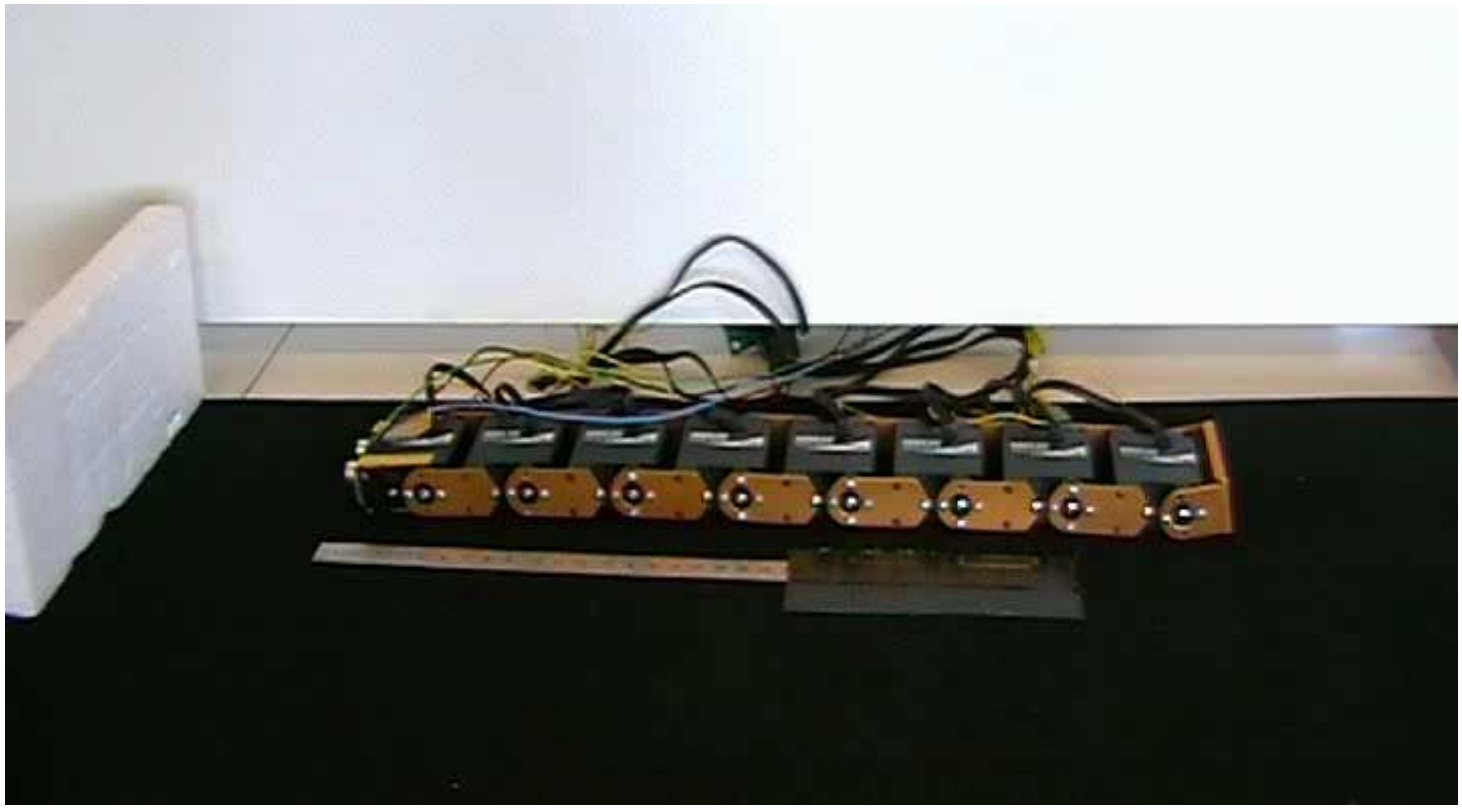
Basic approach:



guarded online-learning

- online and in-situ
- under hard real-time constraints
- hybrid crisp-fuzzy systems as learning paradigm
  -> partitioning of the input space
- avoidance of safety-critical states at any time
  -> safety warranty by tag-mechanism

First Learning Experiment



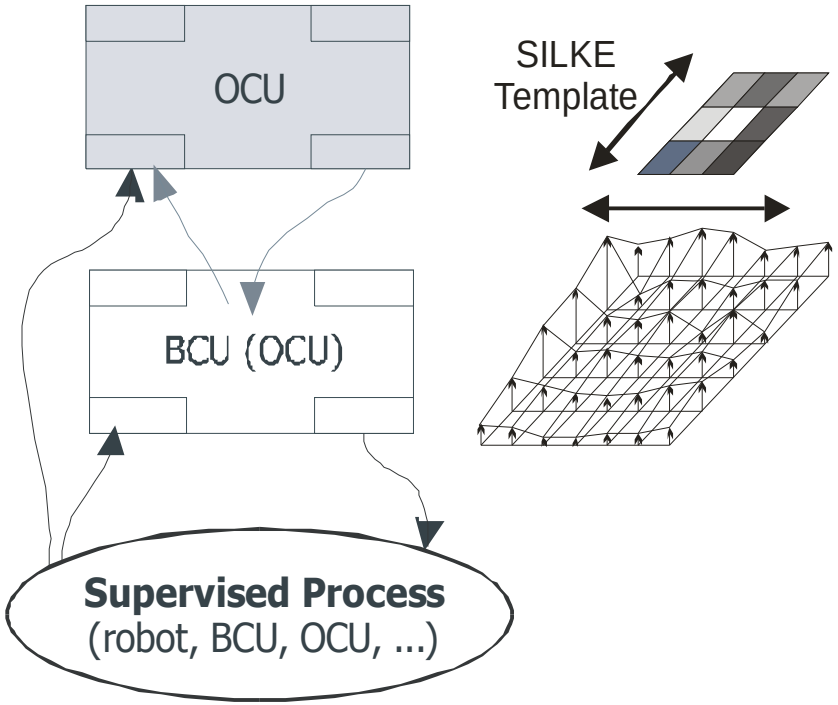
Aim:	optimizing the step width
Sensory input:	change in distance to block
Tuning parameter:	joint angle of wave

SILKE-Approach

(System to Immunize Learning Knowledge-based Elements)

Aim: (meta-level) control of learning dynamics

- avoid harsh control actions
- stabilize convergence
- avoid uninitialized learning situations



Approach:

- locally operating template
- operation on ´cell-level´
- complementary to
  - safety-rules
  - OCUs

Operators:

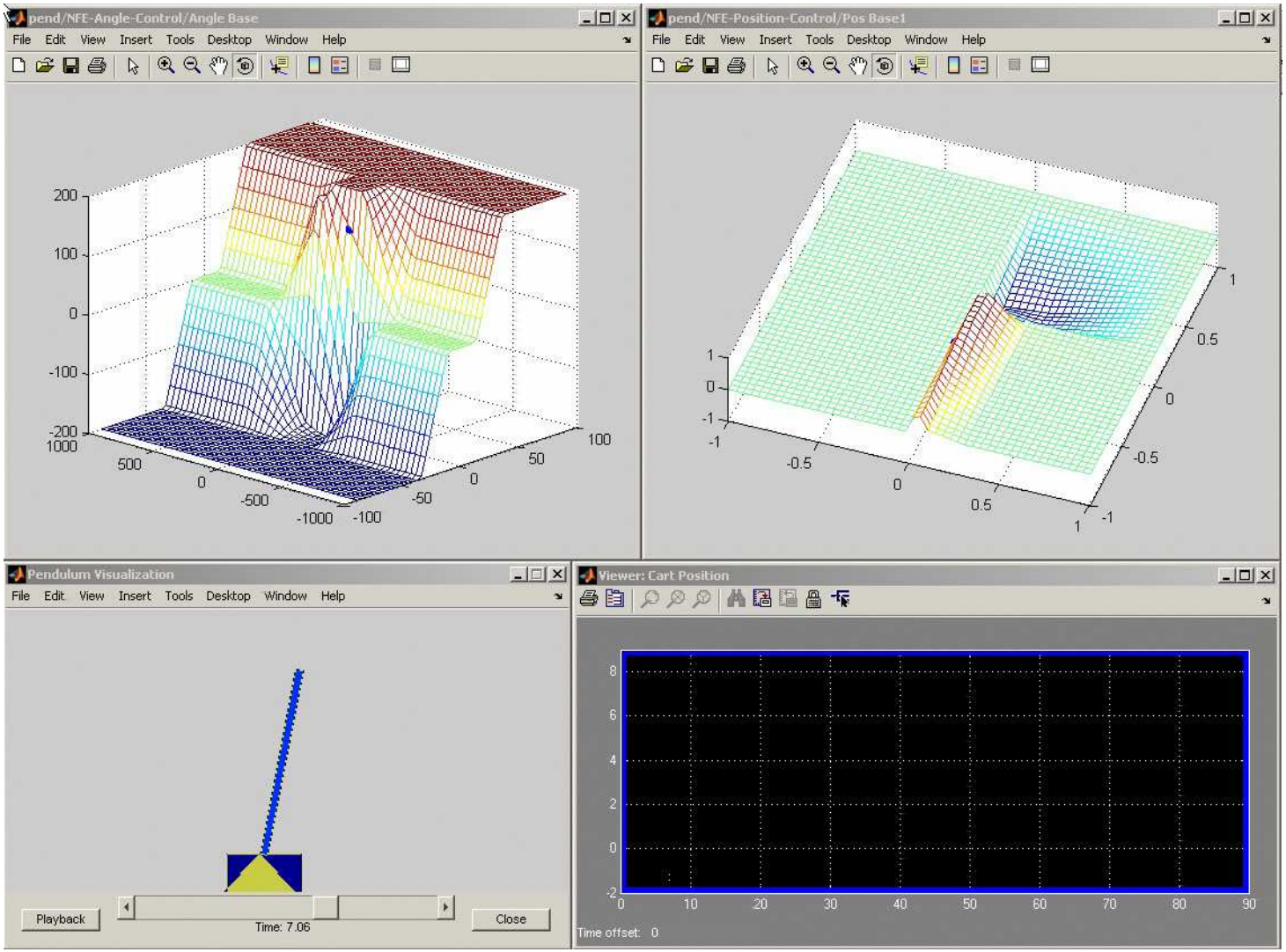
- delimiting the steepness
- averaging
- . . .

Background

Class of application: - interacting learning systems
- learning control of potentially instable, ´unsafe´ systems

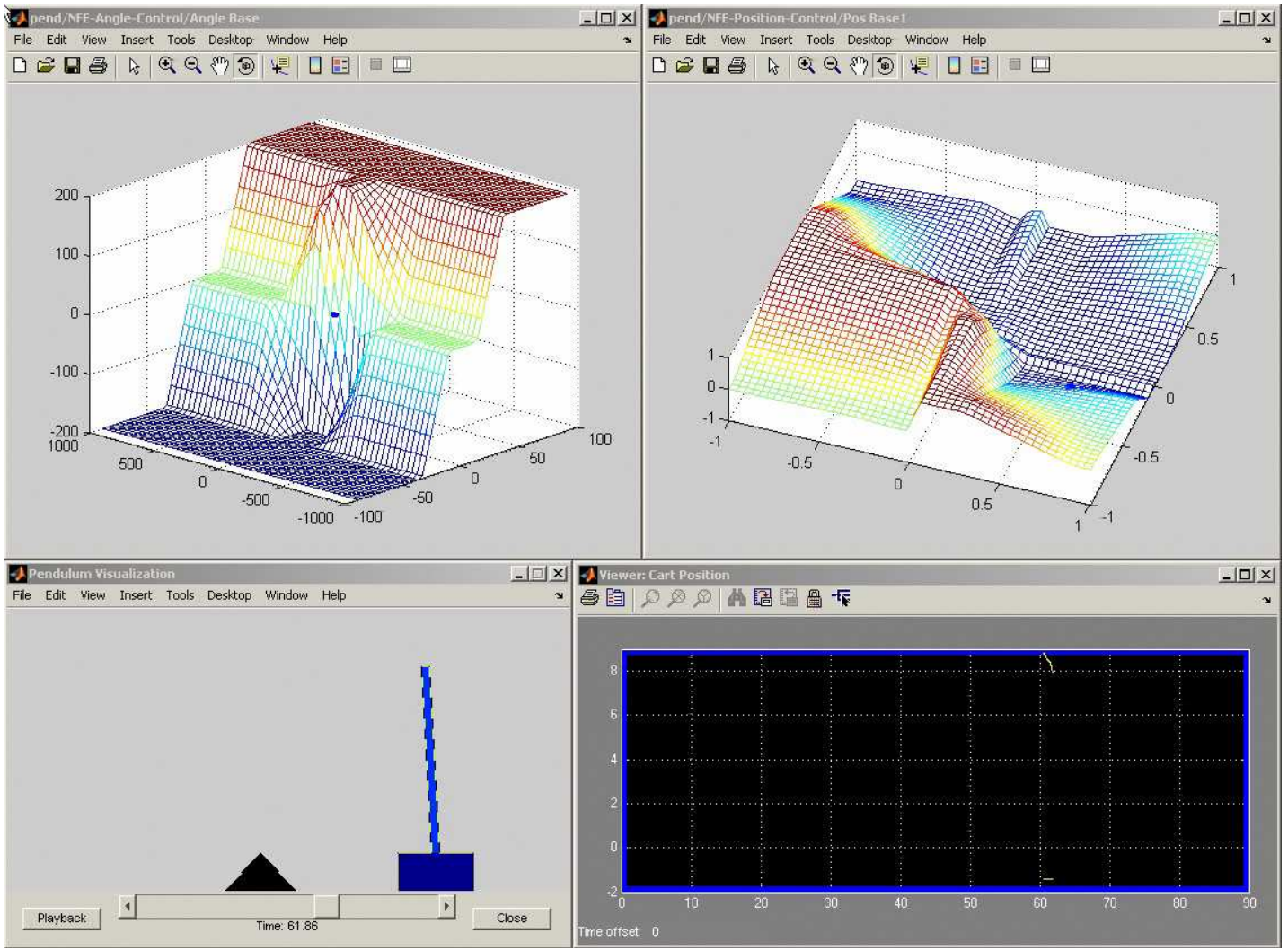
Concrete investigation: interacting learning balance and position controllers (in simulations)

Learning Behavior without SILKE



Interacting learning balance and position controllers

Learning Behavior with SILKE



Interacting learning balance and position controllers

Outlook

ORCA:	- architectural issues (e.g. separate health-signal hierarchy?) - self-organizing components
Health-signal:	- learning of (more diverse) reference behaviors - abstraction and fusion of health-signals
Learning:	- investigations on OSCAR / CARL and in faulty situations - control of interacting learning systems
SILKE:	- investigations under real-life conditions (OSCAR / CARL) - development of new template operators
OSCAR / CARL:	- additional sensors - complex environment & fault scenarios - simulation results verified on real machines

Publications

1. Brockmann, W.; Maehle, E.; Mösch, F.; *Organic Fault-Tolerant Control Architecture for Robotic Applications*. 4th IARP/IEEE-RAS/EURON Workshop on Dependable Robots in Human Environments, Nagoya, Japan, Juni 16-18, 2005, Nagoya University/Japan 2005
2. Großpietsch, K.-E.; *Adaptive Filters for the Dependable Control of Autonomous Robot Systems.* Proc. DPDNS 05 Workshop, Denver 2005
3. Brockmann, W.; Großpietsch, K.-E.; Maehle, E.; Mösch, F.; *ORCA – Eine Organic Computing-Architektur für Fehlertoleranz in autonomen mobilen Robotern*. Mitteilungen der GI/ITG-Fachgruppe Fehlertolerierende Rechensysteme, Nr. 33, März 2006
4. El Sayed Auf, A.; Mösch, F.; Litza, M.; *How the Six-Legged Walking Machine OSCAR Handle Leg Amputation*, To appear at 9th Int. Conf. Simulation of Adaptive Behaviour – SAB'06, Sept 2006.
5. Großpietsch, K.-E.; Silayeva, T.A.; *Organic Computing - A New Paradigm for Achieving Self-Organized Dependable Behaviour in Complex IT Systems.* To appear in: Proc. IDIMT 2006 Workshop, Ceske Budejivice (Czech Republik), September 13-15, 2006
6. Raman Agarwal: *Intelligent Algorithms for Controlling an Autonomous Robot system.* Master Thesis, Fraunhofer IAIS, Sept 2006
7. Brockmann, W.; Großpietsch, K.-E.; Kleinlützum, K.; Maehle, E.; Meyer, D.M.; Mösch, F.: *Concept for a Fault-Tolerant Control Architecture for CLAWAR Machines*. To appear: 9. Int. Conf. Climbing and Walking Robots and the Support Technologies for Mobile Machines, Brussels, Belgium, Sept. 2006
8. Mösch, F.; Litza, M.; El Sayed Auf, A.; Maehle, E.; Großpietsch, K.-E.; Brockmann, W.: *ORCA – Towards an Organic Robotic Control*. To Appear: IWSOS - International Workshop on Self-Organizing Systems, Sept. 2006
9. Jakimovski B.; Litza, M.; Mösch, F.; El Sayed Auf, A.; *Development of an Organic Computing Architecture for Robot Control*. To Appear: GI-Workshop on Organic Computing – Status and Outlook, Dresden, Oct. 2006
10. Brockmann, W.; Meyer, D.M.; *Ein immunsystem-inspirierter Ansatz zum Überwachen des Lernens in Neuro-Fuzzy-Systemen*. To appear: 16. Workshop „Computational Intelligence", Bommerholz, Nov. 2006

ITI

UNIVERSITÄT OSNABRÜCK

IAIS