Organic Fault-tolerant Robot Control Architecture

E. Maehle,
F. Mösch, M. Litza

University of Lübeck
Institute of Computer Engineering

W. Brockmann,
N. Rosemann

Universität Osnabrück

Institute of Computer Science
Computer Engineering Group

K.-E. Großpietsch

AIS
Institut Autonome Intelligente Systeme

Fraunhofer Institut AIS
Sankt Augustin

5th Colloquium *Organic Computing*
Lübeck

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

ORCA - Organic Robot Control Architecture

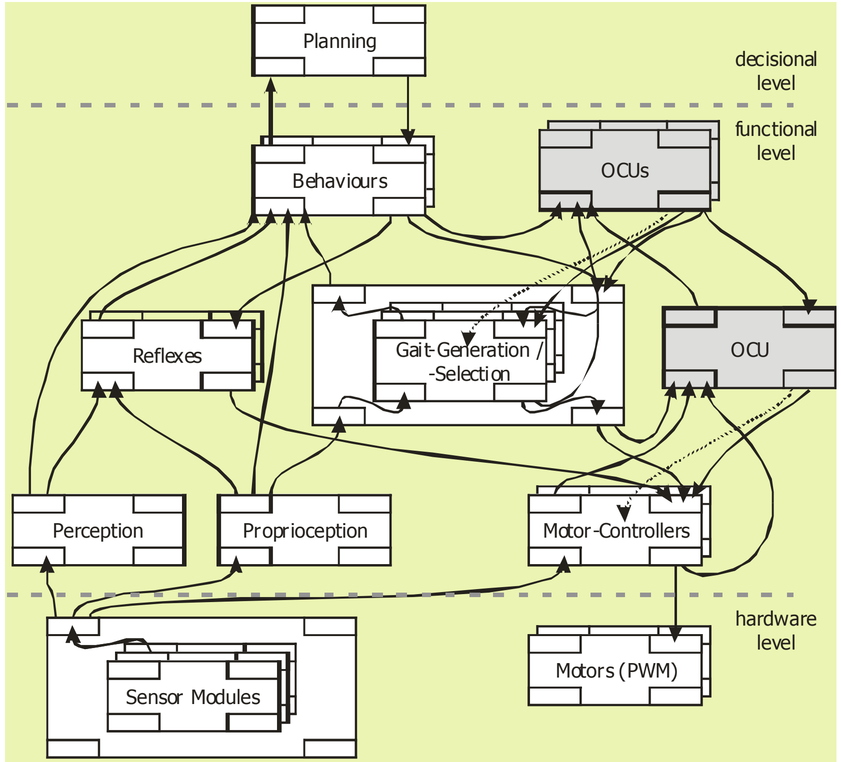
Goals:

- 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

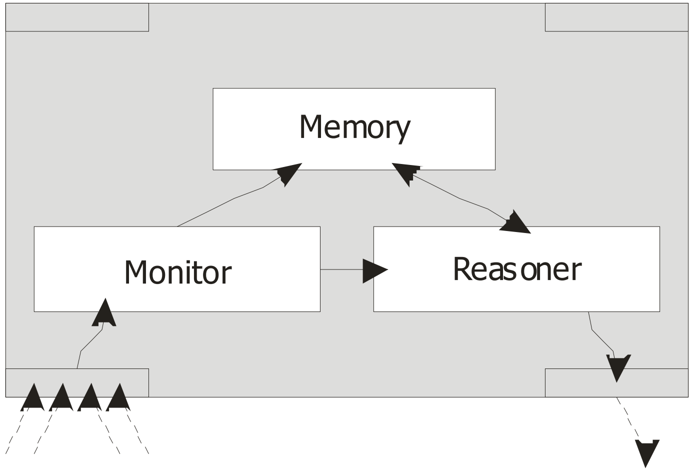
- Approach: controlled emergence on lower functional control levels

ORCA - Organic Robot Control Architecture

- Hierarchic architecture
- Components:
  - BCU = Basic Control Unit
  - OCU = Organic Control Unit
- Observer-Controler related



ORCA – OCU

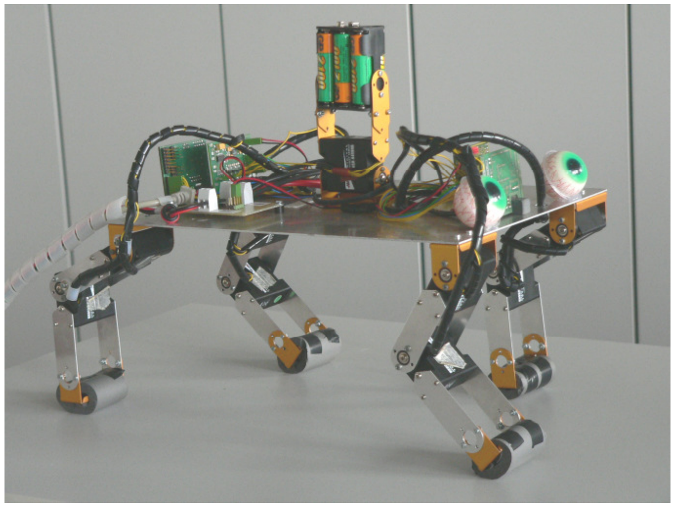


- Monitor: anomaly detection
- Memory: short term history
- Reasoner: hard real-time determination of a counteraction

ORCA Test- and Demonstration-Plattforms

- CARL
- FS-CARL
- WALTER

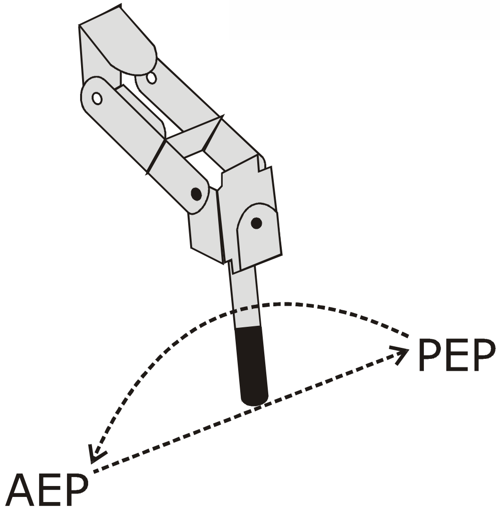
- OSCAR
- OSCAR II
- OSCAR III



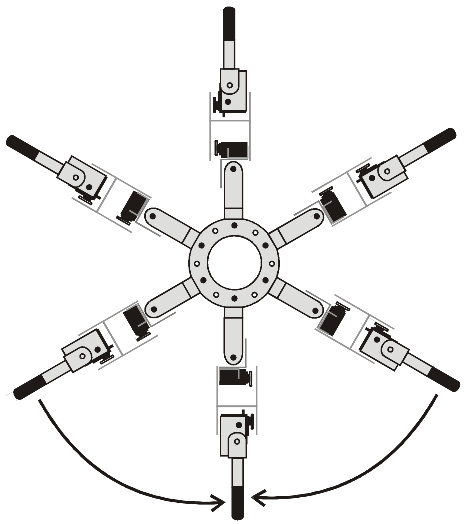
OSCAR

- Hexapod
- 3DOF per leg
- Ground contact sensor

OSCAR – Basic Movement Principles



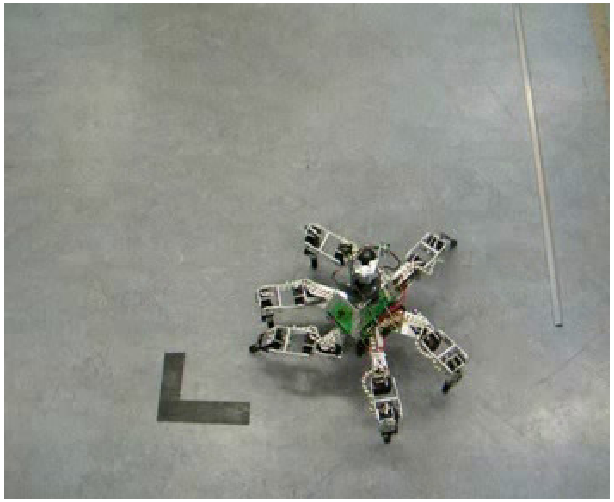
Every leg oscillates periodically between two extreme positions.



One coordination rule:

A leg is allowed to swing only if its neighbors feel ground.

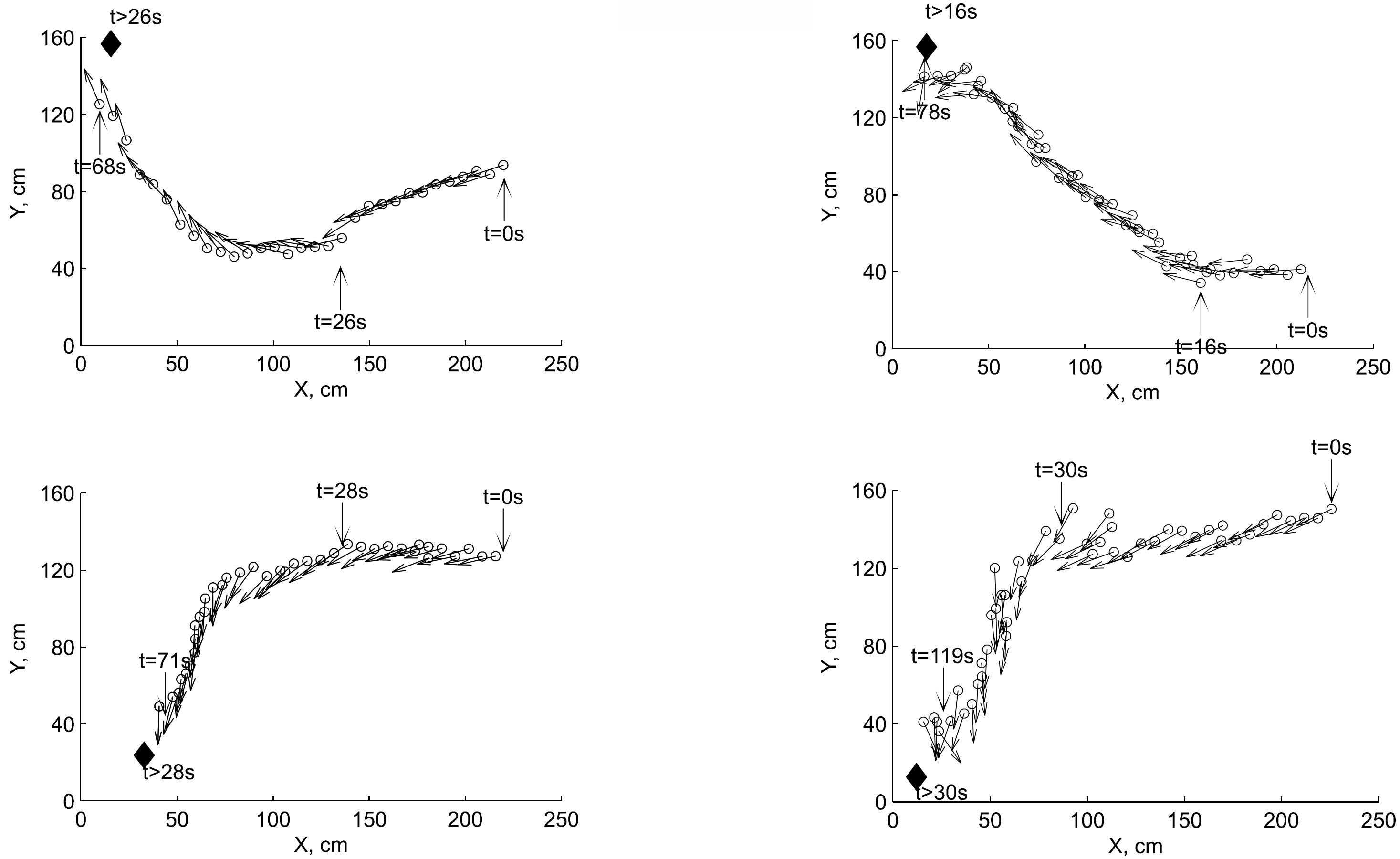
OSCAR - Gait Patterns & Curve Walking



Different gait patterns emerge due to changing few parameters:

- Changing duration of the stance phase change the speed of the pattern.
- Changing distance between the extreme positions allows curve walking.
- Changing relations between legs allows walking with an amputated leg.

OSCAR – Curve Walking Results



Undisturbed

Right middle leg amputated

OSCAR - Anomaly Detection

OSCAR signals:

- 18 servo control signals
- 18 servo position signals
- 18 servo current signals
- 6 ground contact sensors
- 6 independent moving legs (2 possible phases)

Approaches:

- Information Theoretical
- Immune System Inspired (RADE)

Anomaly Detection
– Information Theoretical Approach

- Relation signal (RS) – describes the dependency between 2 signals.

- Dynamic anomaly (DA) – detects if RS is changing significantly relative to its average in the past.

$$DA = \frac{|RS - RSav|}{RSav \Delta RS} 100$$

- Relative anomaly (RA) – detects if RS of a special part (leg) is significantly different from similar parts.

$$RA_i = \frac{\left|RS_i - \overline{RS}_{1:n}\right|}{2(\max(RS_{1:n}) - \min(RS_{1:n}))} 100$$

Anomaly Detection
– Information Theoretical Approach

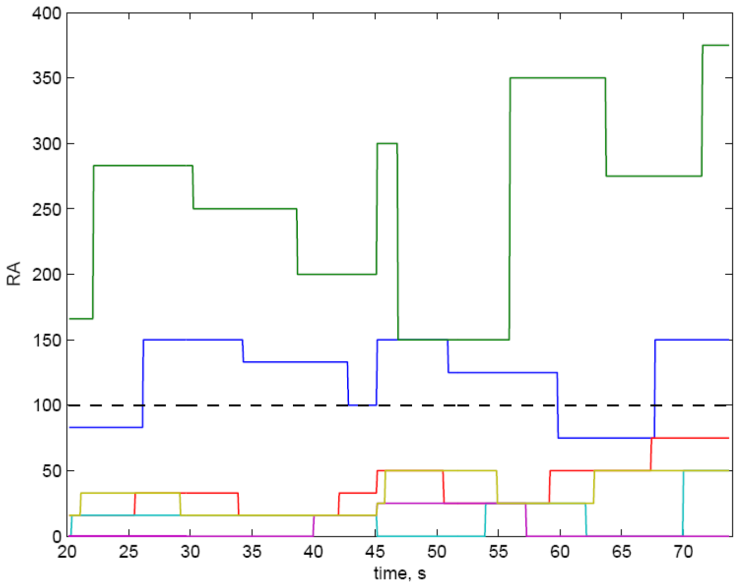
Possible relation signals:

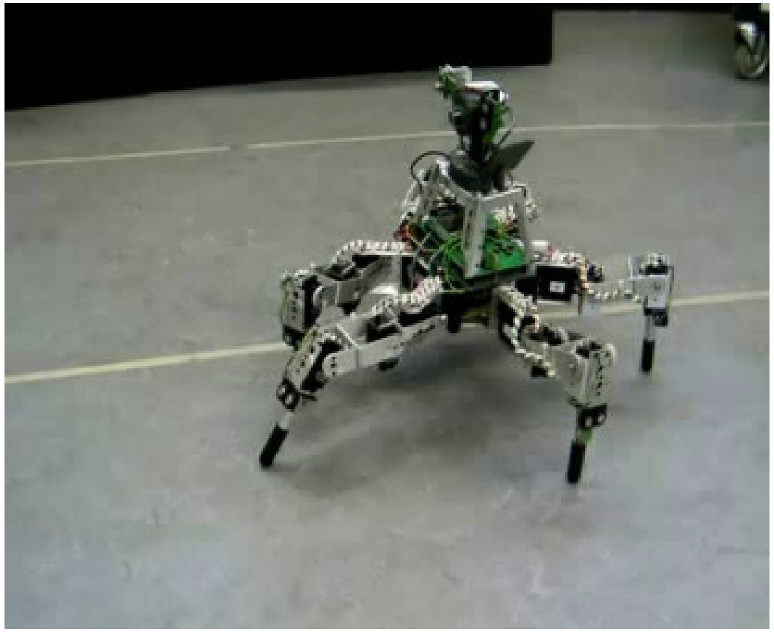
- Mutual Information (MI)
- Correlation

Examples:

- RS(current, real position) allows the detection of a broken gear.
- RS(beta-servo current, ground contact) allows the detection of a missing screw.

Anomaly Detection - Results





Missing screw on one servo (blue) and damaged gear on another (green).

RS is a Mutual Information computed from beta-servo current and ground contact signal.

Health Signal - Concept

Motivation: Engineerability is essential, so we need uniform semantics and representations for anomalies

Approach: Every relevant system entity gets a health signal reflecting its trustworthiness

- System entity: single signal, sensor, subunit, learning unit,…
- Signal reflects the health status of an entity from a **local point of view**
- Other entities (BCU, OCU) can use health signals to calculate own health status
- Health signals can be used to trigger **countermeasures** (OCU)

Health Signal - Semantics

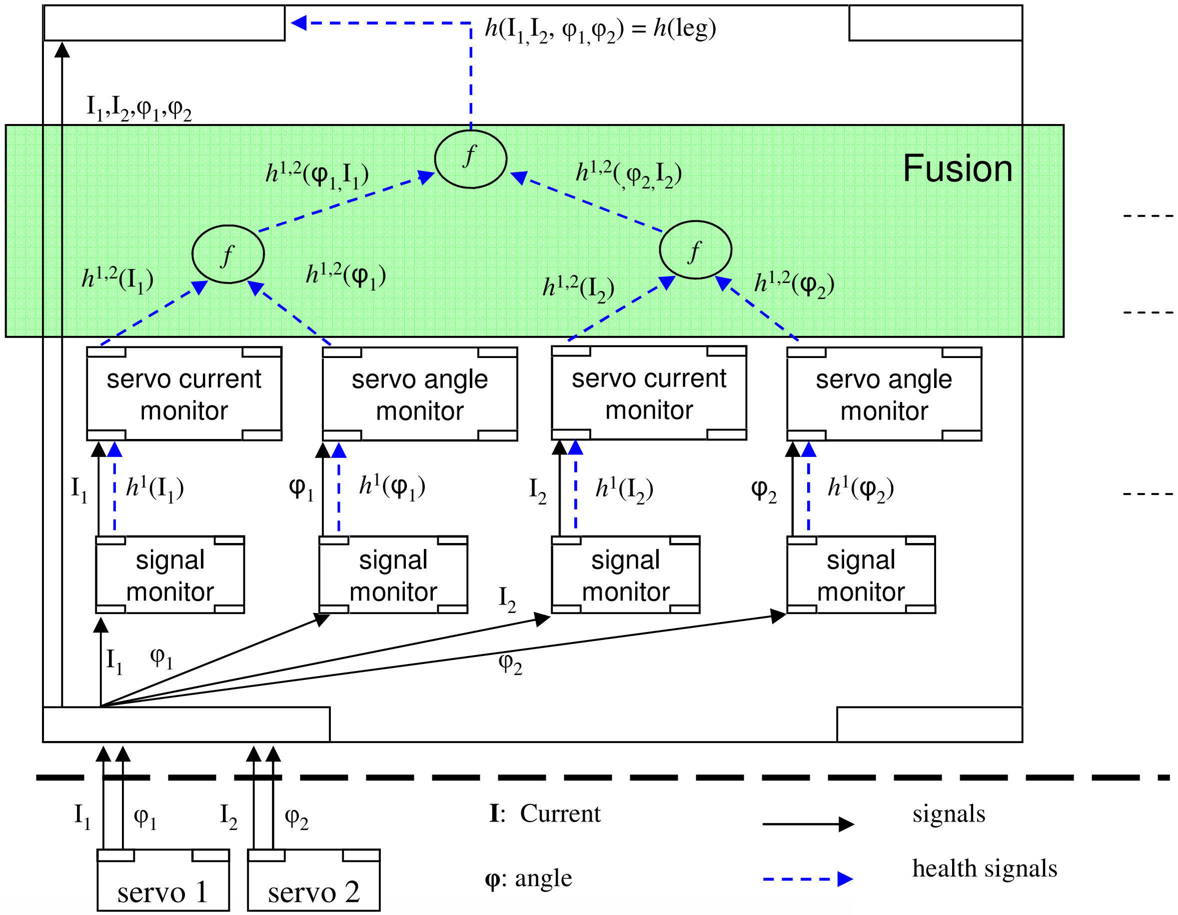
- Health signals (HS) normalized to [0,1]
  - 1: completely healthy (normal) state
  - 0: completely unhealthy (abnormal) state
- Reasons for anomalies (HS<1) are not sent to other entities
- **Advantage**: Uniform treatment for different types of anomalies

  throughout the system hierarchy:
  - Signal quality
  - Hardware faults
  - Interaction faults
  - Design gaps
  - Unlearnt areas
  - …

Health Signal - Fusion

- General approach for fusion: T-norm (minimum, product, …)

  (Other operators for special cases, e.g., for alternative sub-units)



Leg health

Joint health

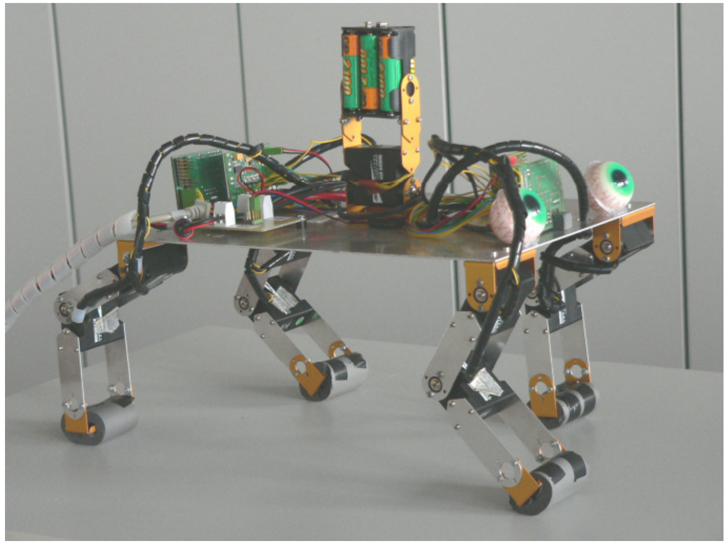
Interaction faults

Signal quality

Health Signal – Closing the Loop

- First example: Reaction to dropping of health signal when stepping against obstacles with robot WALTER

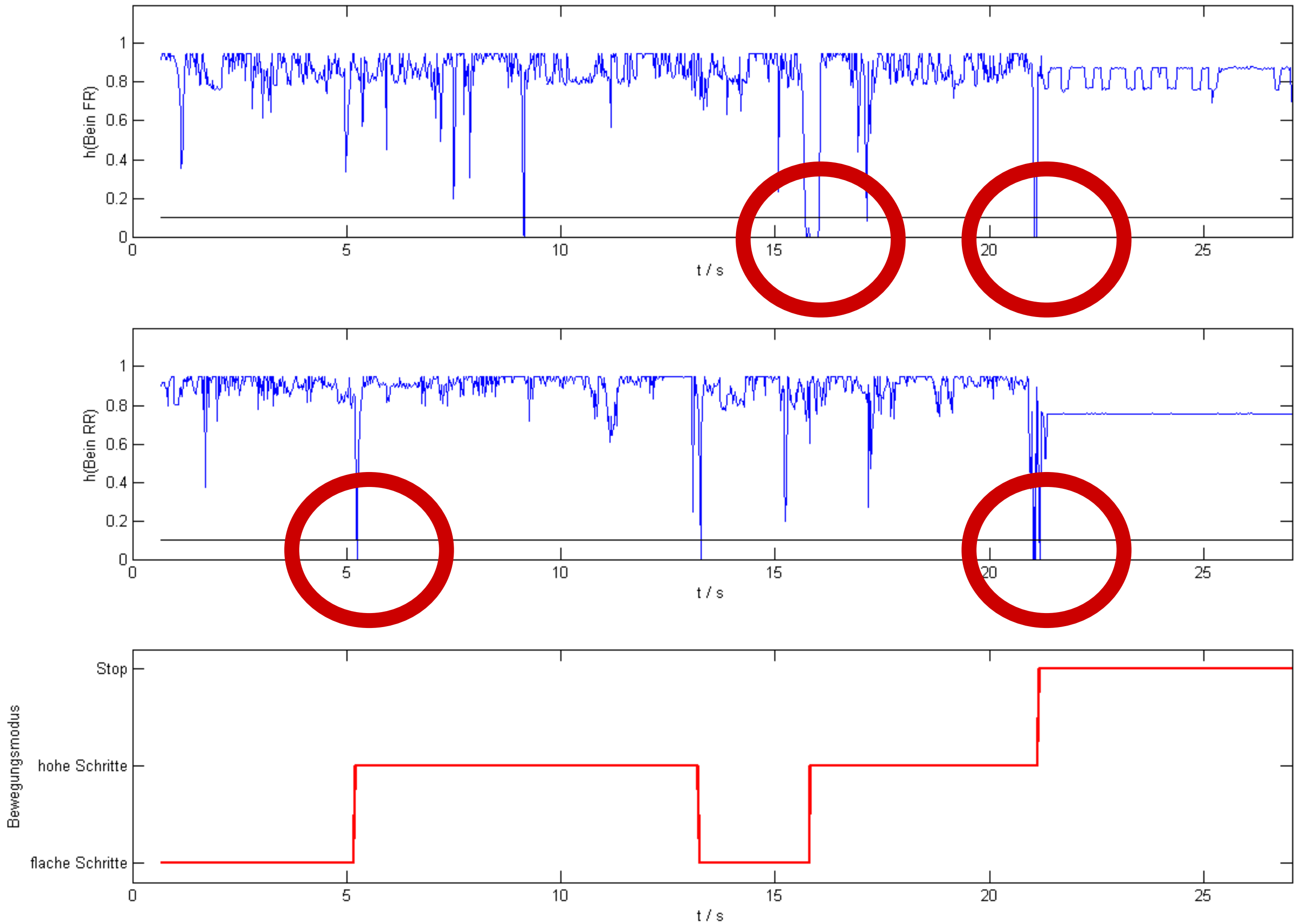
→ step against obstacle
→ anomalies in current/position
→ dropping health signal
→ increase step height
→ overcome obstacle
→ keep walking
→ decrease step height



Health Signal – Closing the Loop

Small Obstacle

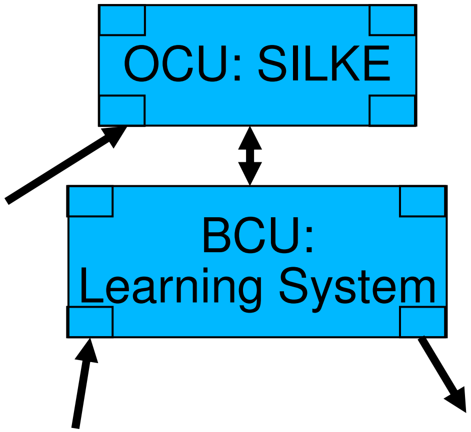
Health Signal – Closing the Loop



Safe Learning - Concept

Motivation: Learning is essential for complex self-x systems, but safety and controllability have to be guaranteed

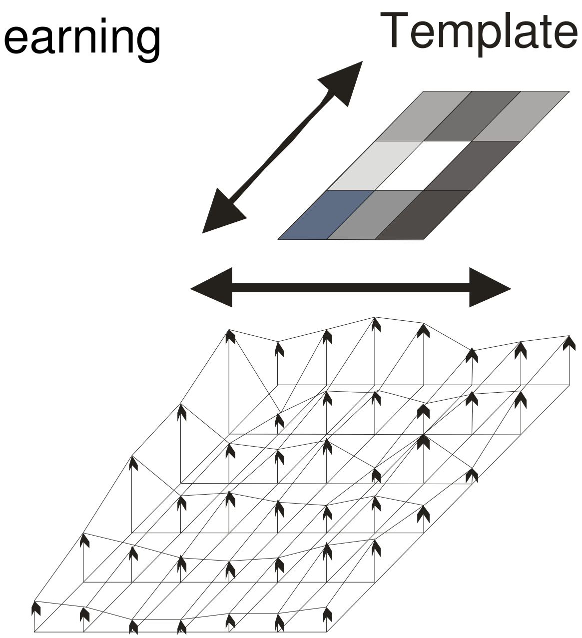
Approach:
- *Adaptive Filters* for low level adaptation
- *Controlled self-optimization* by determining and correcting local characteristics of the dynamically learnt behavior (*monitoring* and/or *guidance*)



- Inspired by the immune system
  - → Detection of anomalies/antigenes by local characteristics through local templates
- Incremental learning in a local learning system
  - → Although a **local** approach, there is a **global** effect

Safe Learning – SILKE
(System to immunize learning knowledge-based elements)

- Extension of zero-order Takagi-Sugeno fuzzy systems
  - Rules form a fixed lattice in the input space
  - Each node of the lattice is a conclusion
  - Conclusions (i.e., numbers) are changed by learning
- After each learning step: look at local neighborhood of the input vector where learning took place
- Template determines characteristics of neighborhood



SILKE - Foundations

- Apply template H to n-dim rulebase S at rule $i_1,\ldots,i_n$ by normalized **matrix convolution** to obtain SILKE-template-values S'

$$S'_{i_1,\ldots,i_n} = \frac{1}{N_H \cdot r^n} \sum_{u_1=0}^{r-1} \cdots \sum_{u_n=0}^{r-1} S_{i_1+u_1-k,\ldots,i_n+u_n-k} \cdot H_{u_1+1,\ldots,u_n+1}$$

- Template size r >1 (odd number), normalization N_H, $k = (r-1)/2$

Examples for 2-dim templates:

$$H_{average} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad H_{gradient} = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

SILKE - Example

Simulated pole-balancing cart

- SILKE-*Monitoring* by template for intended gradient
- SILKE-*Guidance* by template for average rule conclusions

Learning step

Apply template

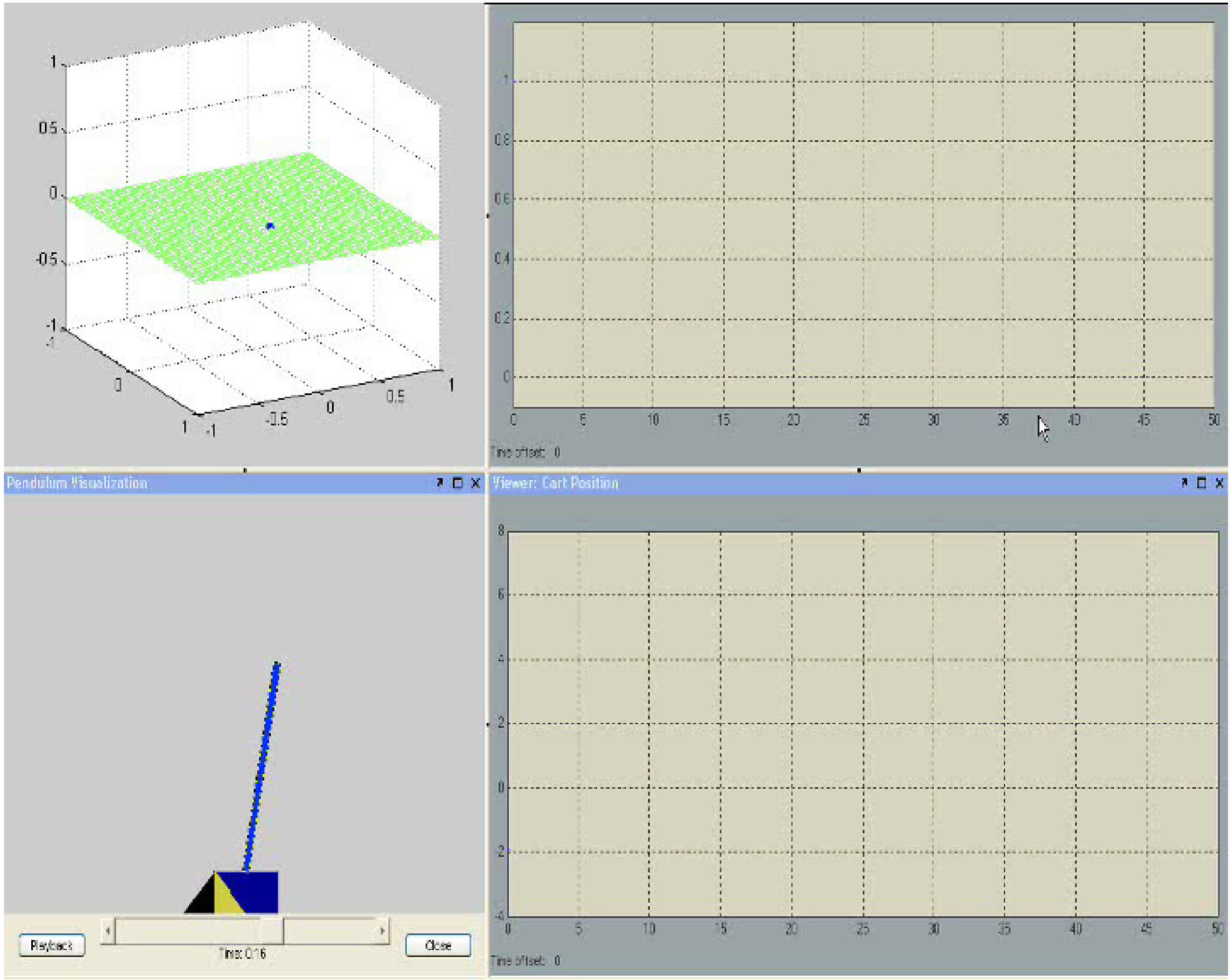
Correct rule base

The SILKE algorithm with a sample averaging template applied to the current state *s* and a local fuzzy learner *L* with adjustment rate λ:

> *active_rules* ← all rules of *L* which are applicable to state *s*
> **for each** rule *a* **in** *active_rules* **do**
>   *conclusion(a)* ← PerformLearning(*L*, *s*)
> **end for**
> **for each** rule *a* **in** *active_rules* **do**
>   *silke(a)* ← apply template to rule *a*
> **end for**
> **for each** rule *a* **in** *active_rules* **do**
> $conclusion(a) \leftarrow (1-\lambda)\cdot conclusion(a) + \lambda\cdot silke(a)$
> **end for**

- Optimize guidance effect by tuning the **adjustment rate** λ

SILKE – Example without Correction

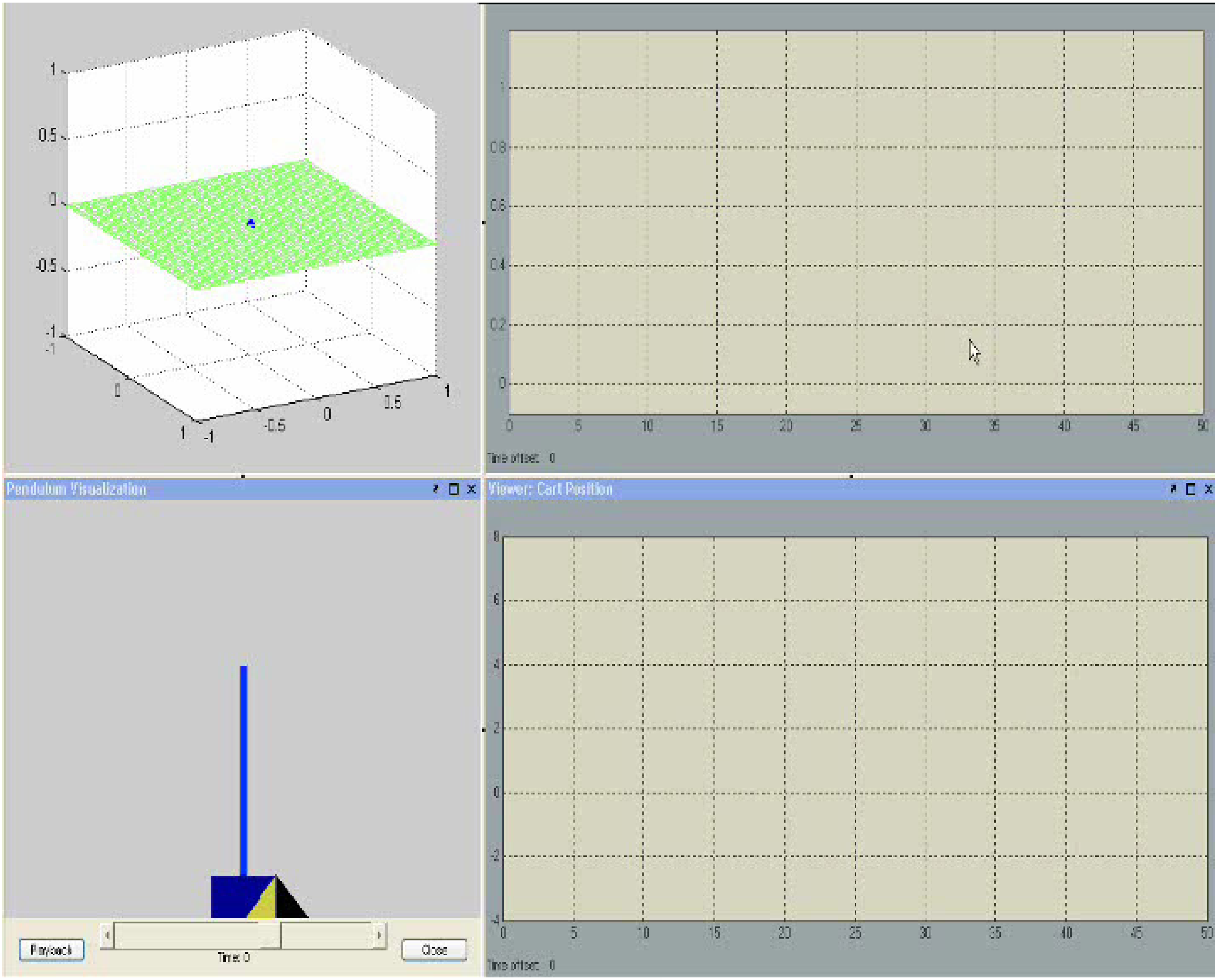


Monitoring:

$$\begin{pmatrix} -1 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

No correction with SILKE

SILKE – Example with Correction



Monitoring:

$$\begin{pmatrix} -1 & -1 & 1 \\ -1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Correcting:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

ORCA-Project

Issues

	Emergence	Health Signal	Adaptive Filters	Safe Learning
PHASE 1	Gait Pattern with Walknet Cope with leg detachment Emergent curve walking	Semantics Generation Fusion SILKE for monitoring	Simulation of relevant faults Fault compensation by adaptive filters	SILKE templates Adjustment rate Mathematical foundations
PHASE 2	Self-opt leg movement Adaptation to internal anomalies Self-org high level behavior	Health signal as feedback to control Health signal for whole system	Implementation of AF for low level fault compensation in Real Robots	Interplay of health signal and learning Disrupted learning Extension of SILKE-approach

OSCAR I | OSCAR II | OSCAR III | Simulations | WALTER | FS-CARL | CARL

Platforms