

# Organic Computing in Mobilen Arbeitsmaschinen

H. Schmeck, S. Mostaghim, *M. Wünsche*<sup>1)</sup>  
M. Geimer, T. Kautzmann<sup>2)</sup>

Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB),  
Institut für Fahrzeugsystemtechnik (FAST),

Lehrstuhl für Effiziente Algorithmen<sup>1)</sup>  
Lehrstuhl für Mobile Arbeitsmaschinen (Mobima)<sup>2)</sup>



# AIFB

Institut für Angewandte Informatik  
und Formale Beschreibungsverfahren

Prof. Dr. Schmeck  
Dr.-Ing. Mostaghim

Deutsche  
Forschungsgemeinschaft

DFG



**MOBIMA**  
CHAIR OF MOBILE MACHINES

Prof. Dr. Geimer

# Agenda

- Ziele
- Einführung in Organic Computing
- Überblick zur Observer/Controller(O/C)-Architektur
- O/C-Architektur für Mobile Arbeitsmaschinen
- Zusammenfassung und Ausblick



## Organic Computing in Off-highway Machines

- Steigende Treibstoffpreise
- Hoher Anteil Treibstoffkosten an Gesamtkosten
- Wachsende Bedeutung des Umweltschutzes



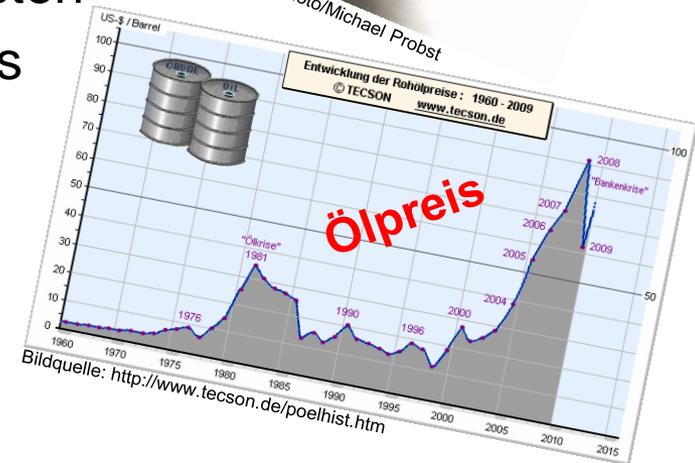
Reduktion des Treibstoffverbrauchs

durch

- Ausnutzung immer größerer Zahl von Freiheitsgraden in der Steuerung
- Systemübergreifende Betrachtung der Maschine
- Entwicklung eines lernfähigen, adaptiven Gesamtmaschinen-Managements



Bildquelle: AP Photo/Michael Probst



Bildquelle: <http://www.tecson.de/poelhist.htm>



# Organic Computing

- Steigende Komplexität technischer Systeme
  - Kommunizierende, selbständig arbeitende Einheiten
  - Unmöglich, a priori alle Eventualitäten vorherzusehen
  - Ziel:  
**adaptive** Systeme, die sich **selbstorganisiert an Veränderungen anpassen** können
  - Voraussetzung:  
**„Lebensähnliche Eigenschaften“ / „organisches Verhalten“**
    - Selbst-organisierend
    - Selbst-konfigurierend
    - Selbst-optimierend
    - Adaptiv
    - Zuverlässig
    - Lernfähig

Problem:  
Systeme „organisch“,  
aber trotzdem **beherrschbar**  
gestalten!

- Von der DFG gefördert von 2005 bis 2011  
<http://www.organic-computing.de/spp>



- Tieferes Verständnis für das Verhalten komplexer Systeme
- Entwicklung von Design-Konzepten für „organische“ technische Systeme
  - Systemübergreifende Betrachtung
  - „Gesteuerte Selbstorganisation“
  - **Observer/Controller-Architektur**

# Observer/Controller(O/C)-Architektur

- System under Observation and Control (SuOC)

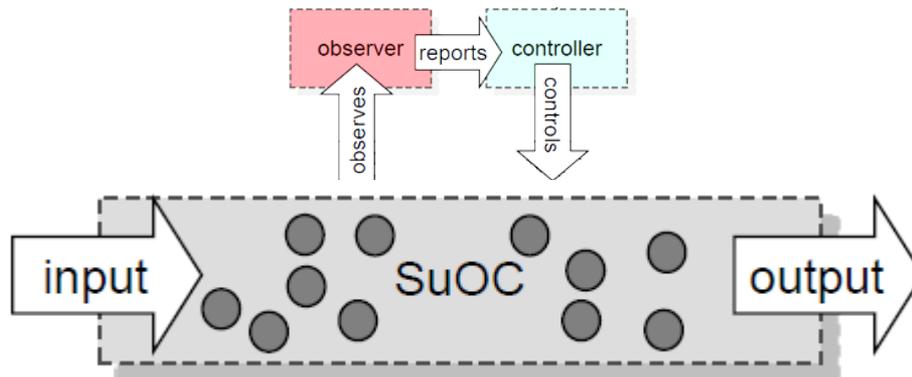
- Komplexes System interagierender Komponenten

- In sich vollständig funktionsfähig

- O/C-Architektur betrachtet Gesamtsystem, greift nur ein, wenn Verbesserungspotential erkannt wird

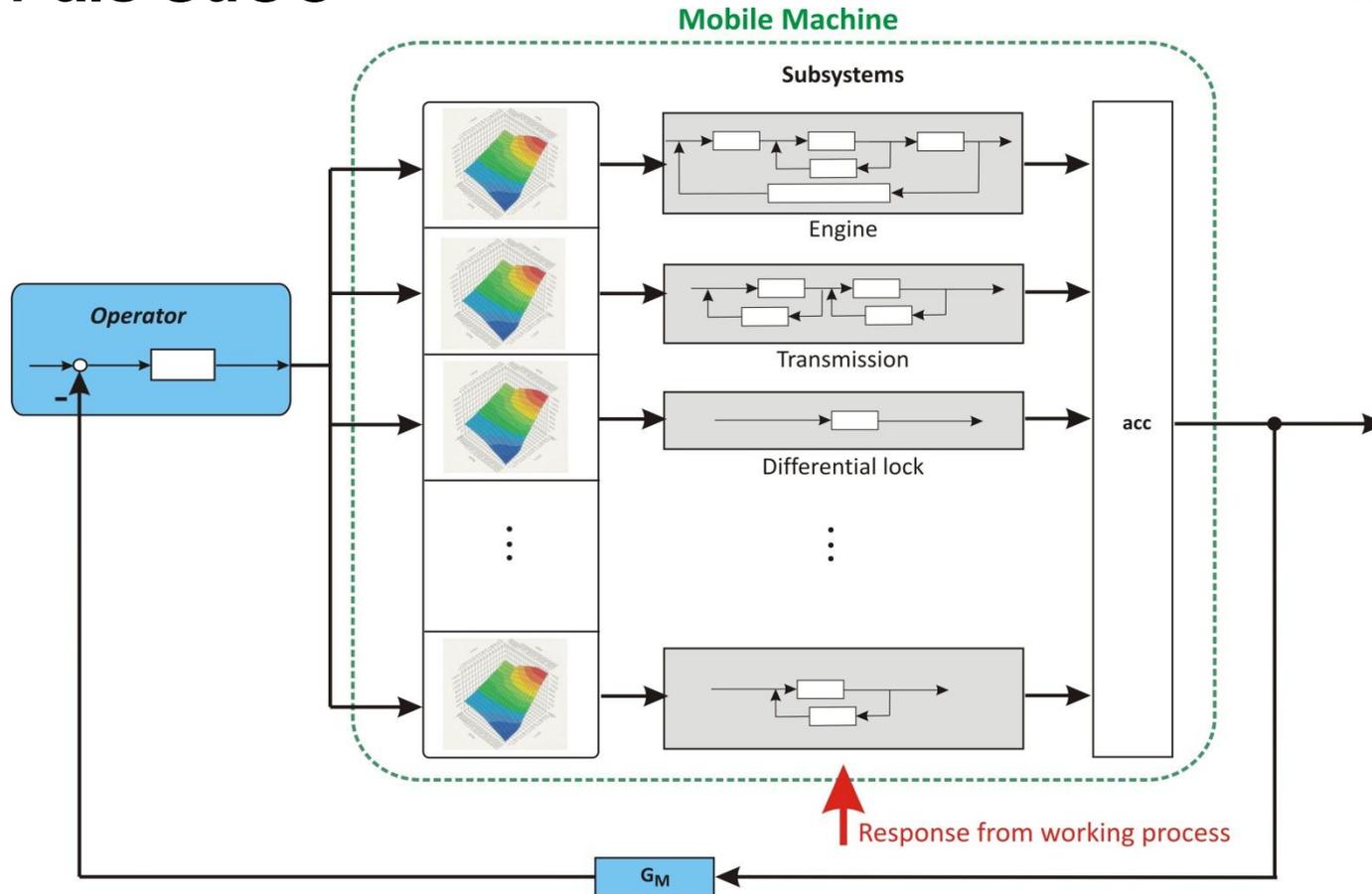
→ Gesteuerte Selbstorganisation!

- Im Projekt OCOM: Fendt Vario 412



U. Richter et al: Towards a generic observer/controller architecture for Organic Computing. In INFORMATIK 2006 - Informatik für Menschen!, Vol. P-93 of LNI, pp. 112–119.

# Traktor als SuOC



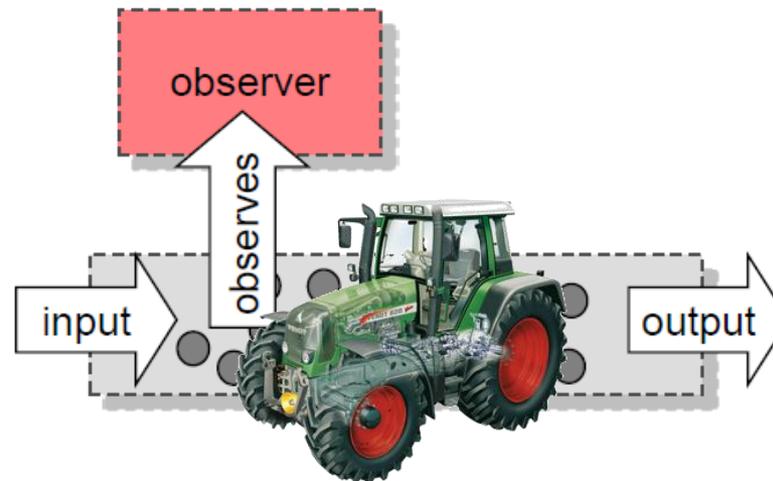
**Co-Simulation**



# Observer/Controller-Architektur

## ■ Observer

- Auswertung aller Messdaten aus dem SuOC
- Charakterisierung des aktuellen Systemzustands (→ Situation)
- Zusammenfassen aller relevanten Werte für den Controller

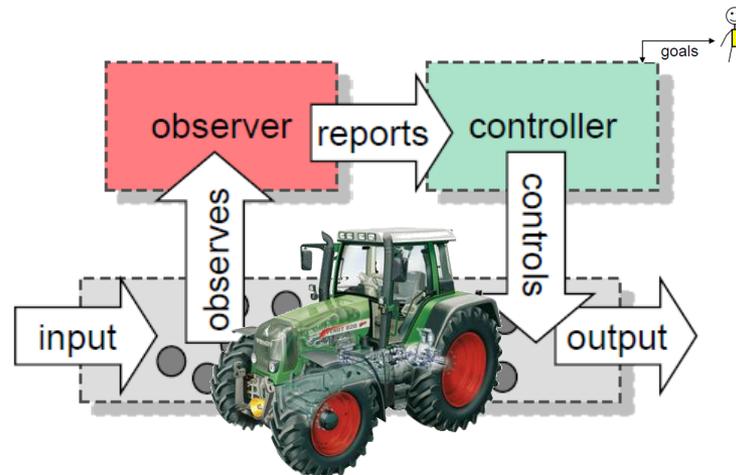


U. Richter et al: Towards a generic observer/controller architecture for Organic Computing.  
 In INFORMATIK 2006 - Informatik für Menschen!, Vol. P-93 of LNI, pp. 112–119.

# Observer/Controller-Architektur

## ■ Controller

- Beurteilt Systemverhalten bzgl. vom Nutzer vorgegebener Ziele
- Kann Systemparameter des SuOC beeinflussen (→ Aktion) (gesteuerte Selbstorganisation)
- Lernt geeignete Aktionen für identifizierte Systemzustände
- Zwei-Ebenen Lernen (**Online / Offline Learning**)



U. Richter et al: Towards a generic observer/controller architecture for Organic Computing.  
 In INFORMATIK 2006 - Informatik für Menschen!, Vol. P-93 of LNI, pp. 112–119.

# Online / Offline Lernen

## ■ Zwei-Ebenen Lernen im Controller:

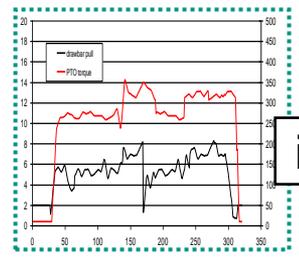
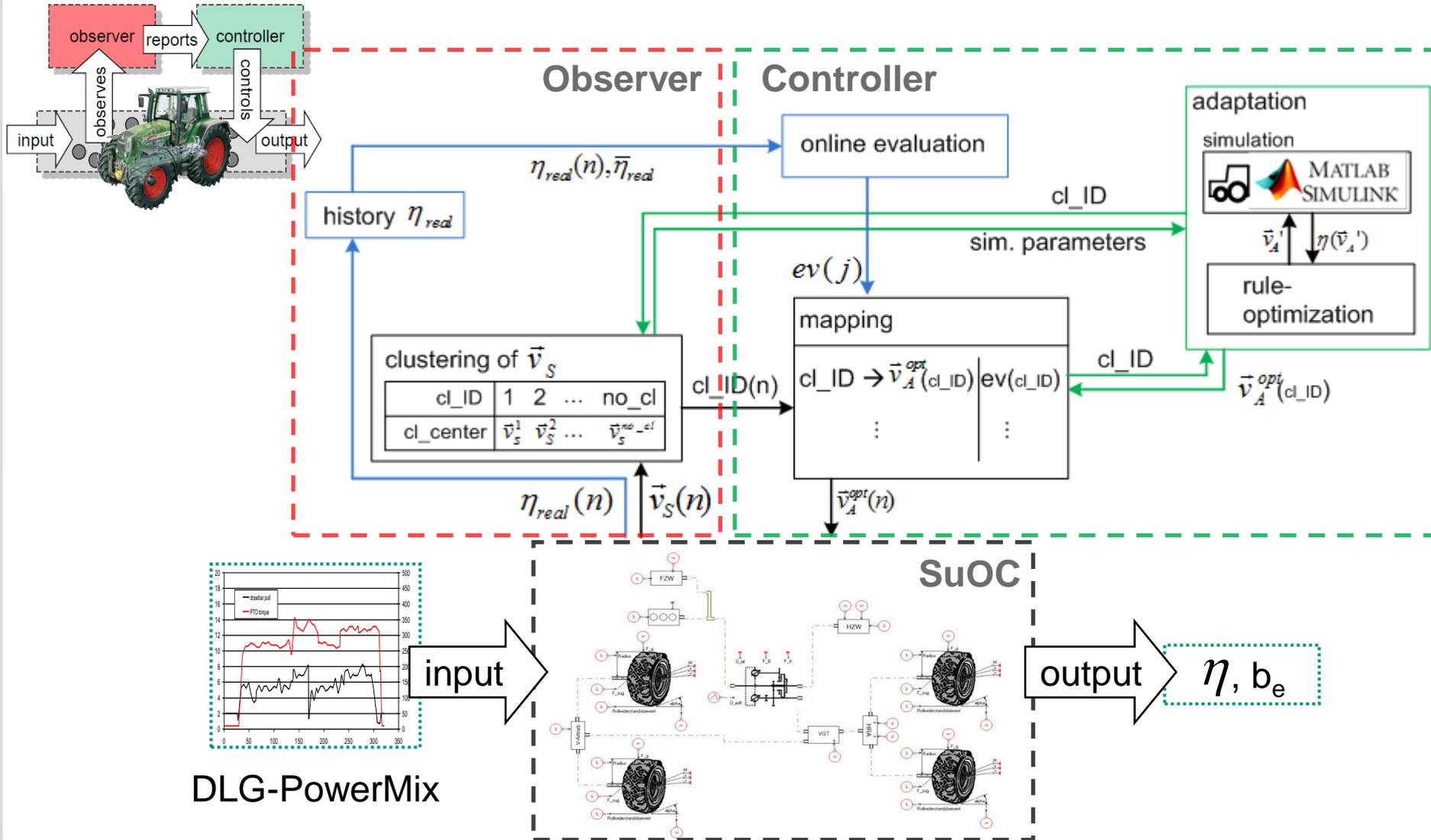
### ■ Offline:

- Erlernen neuer Regeln („Systemzustand→Aktion“) für Situationen des SuOC
- Grundlage:  
Beurteilung von Regeln mittels vereinfachtem *Simulationsmodell*

### ■ Online:

- Anwendung offline erlernter Regeln:  
Aktion ausführen, Wirkung bewerten  
→ Eignung der Aktion für aktuelle reale Situation  
→ „schlechte“ Aktionen mit der Zeit ersetzen
- „*Realitätstest*“ offline gelernter Regeln  
→ Anpassung an dynamische Veränderung der Umgebung

# O/C-Architektur im Projekt OCOM

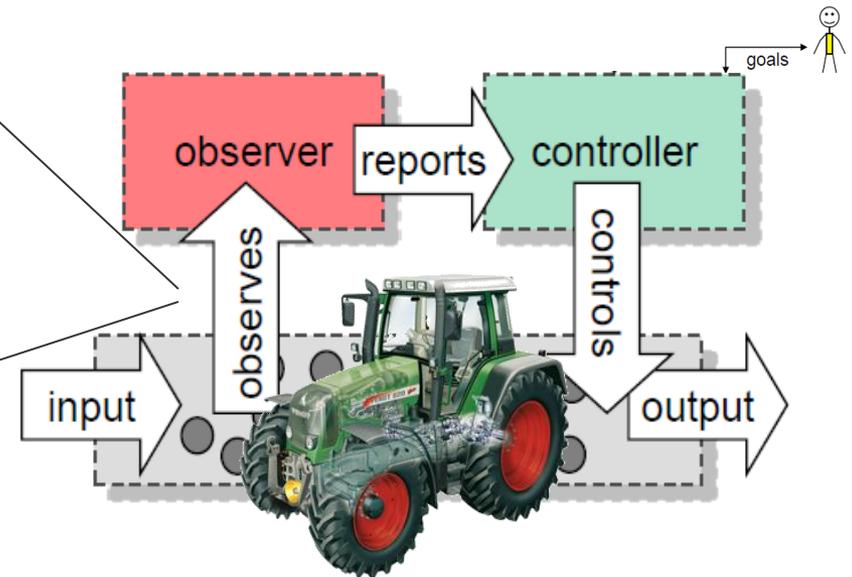


DLG-PowerMix

# Situationsbeschreibung – Parameter

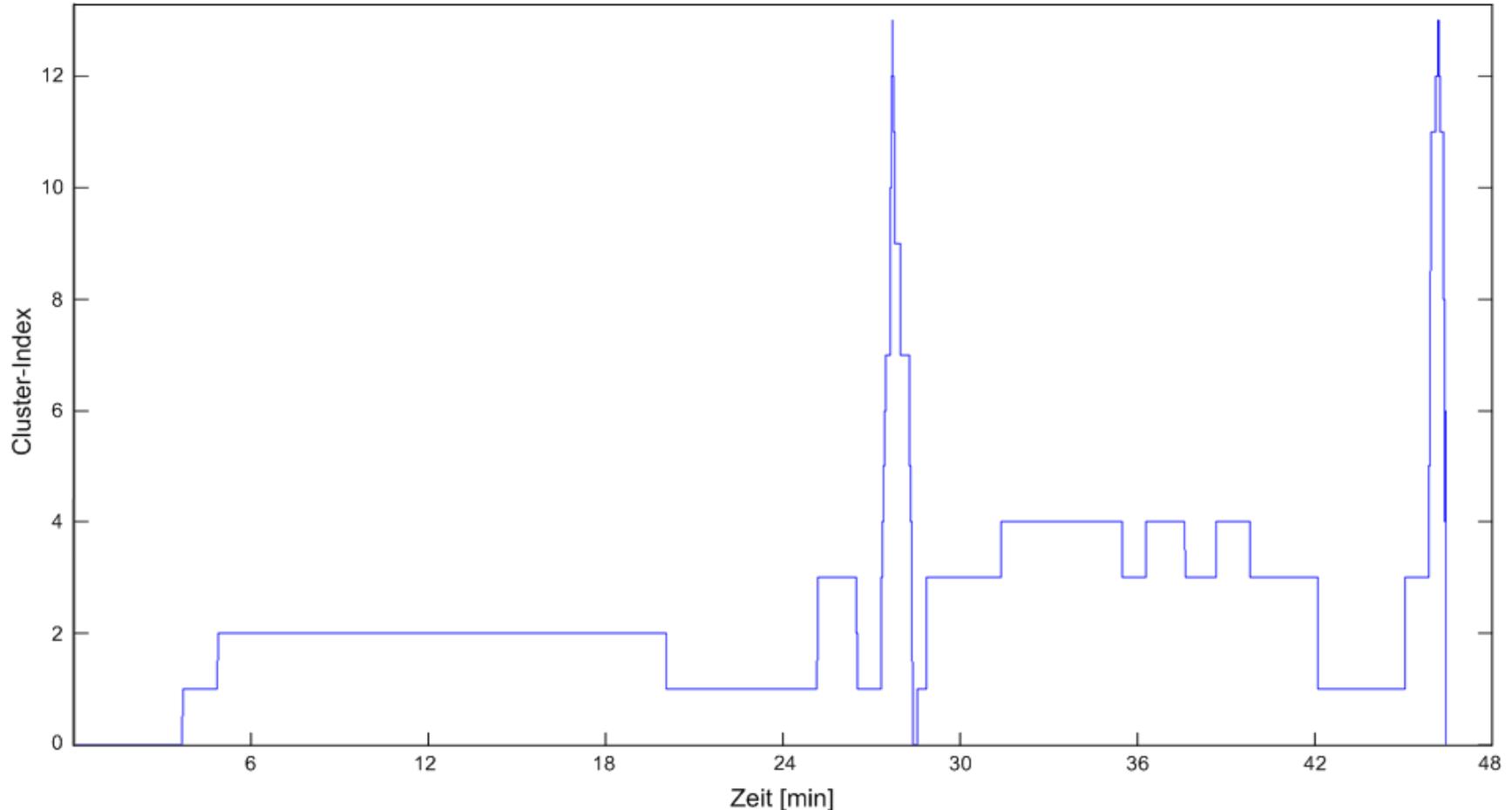
charakterisiert die Anforderungen  
des Arbeitsprozesses an die Maschine

$\vec{v}_S =$	}	$F_{Zug}$	Zugkraft
		$F_{1Z}$	Reifenaufstandskräfte
		$F_{2Z}$	
		$F_{3Z}$	
		$\sigma$	
		$M_{HZW}$	Moment Heckzapfwelle
		$P_{PR}$	Druck Arbeitshydraulik
		$Q_{PR}$	Volumenstrom Arbeitshydraulik
		$P_{PL}$	Druck Lenkpumpe
		$\varphi$	Lenkwinkel

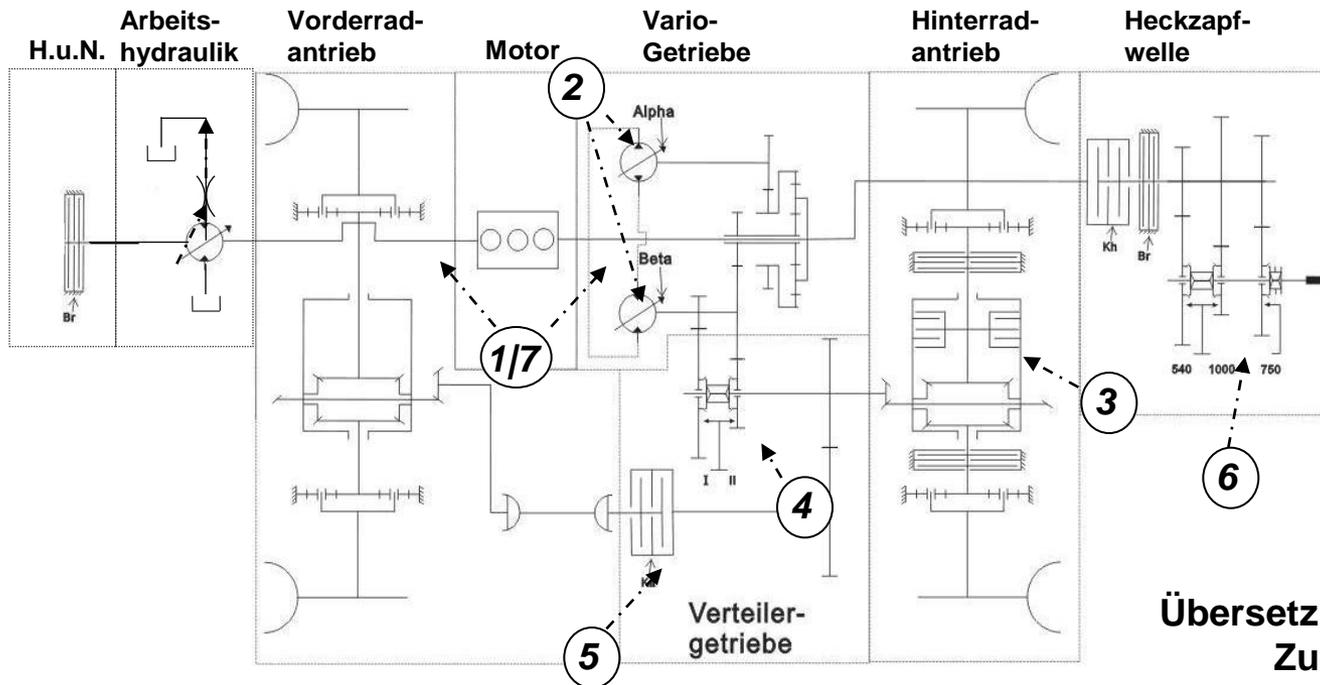


# Situationsbeschreibung – Clustering

- Ähnliche Werte von  $\vec{v}_s$  werden zu einer Situation zusammengefasst
  - Bsp.: DLG Power-Mix Z1G



# Freiheitsgrade im Traktor

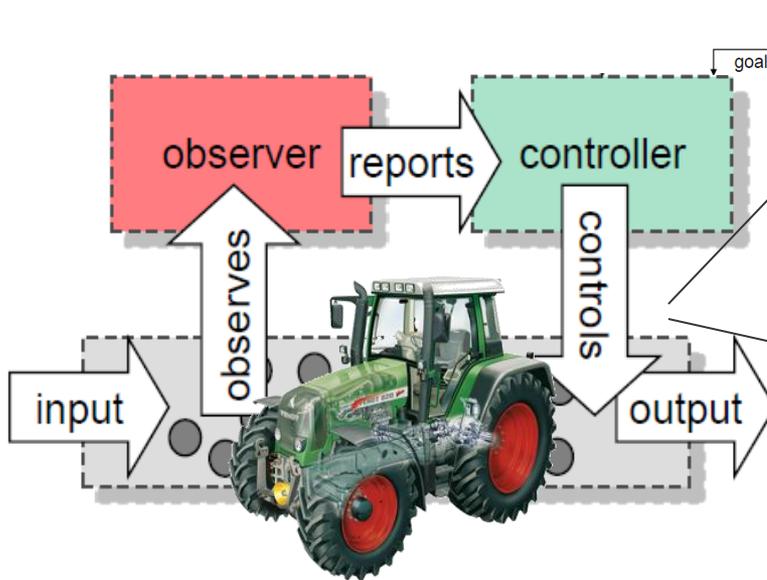


- Drehzahl Kurbelwelle (1)
- Übersetzungsverhältnis Getriebe (2)
- Zustand Differenzialsperre (3)
- Stufe Gruppenschaltung (4)
- Zustand Allradkupplung (5)
- Übersetzungsverhältnis Heckzapfwelle (6)
- Drückungsdrehzahl (7)

→ Aktionsvektor  $\vec{v}_A$

# Aktionsbeschreibung

beinhaltet die  
Einflussmöglichkeiten auf die Maschine



$$\vec{v}_A = \begin{pmatrix} n_{Kurbel} \\ v_{Soll} \\ DS \\ GS \\ AR \\ FHZWe \\ n_{Drück} \end{pmatrix}$$

Drehzahl Kurbelwelle

Sollgeschwindigkeit

Differenzialsperre

Gruppenschaltung

Allrad-Zuschaltung

Spar-Zapfwelle

Drückungsdrehzahl

$$\text{Max } \eta(\vec{v}_A, \vec{v}_S)$$

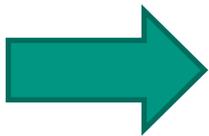
$$\vec{v}_A \in R^A$$

Aktionsvektor, zur Einflussnahme auf den Traktor

$$\vec{v}_S \in R^S$$

Sensorwerte zur Situationsbeschreibung  
( $y$ : teilweise unbekannte Umgebungseinflüsse)

$$\vec{v}_S = f(\vec{v}_A, y)$$



Finde für alle erkannten, wiederkehrenden Situationen  $\vec{v}_S$   
eine Aktion  $\vec{v}_A$ ,  
die zu einem möglichst hohen **Wirkungsgrad  $\eta$  des Gesamtsystems** führt

## Zusammenfassung

- Treibstoffverbrauch reduzieren durch „**organisches**“ Gesamtmaschinen-Management
- Systemübergreifende Betrachtung der Maschine
- Fendt Vario 412 als **System under Observation and Control**
- Klassifizierung des Systemzustands durch Clustering
- **Online** und **offline Lernen** geeigneter Aktionen

## Ausblick

- **O/C Architektur** anhand des Modells verifizieren und weiterentwickeln
- Übertragung der Architektur auf den **realen Traktor**
- Validierung des **Traktormodells**
- Realisierung weiterer Zielvorgaben (**Multikriterielle Optimierung**)

**Vielen Dank für die Aufmerksamkeit.  
Fragen?**