

# Vorlesung Computational Intelligence:

## Teil 1: Einführung und Motivation

**Ralf Mikut, Wilfried Jakob, Markus Reischl**

Karlsruher Institut für Technologie, Institut für Automation und angewandte Informatik

E-Mail: [ralf.mikut@kit.edu](mailto:ralf.mikut@kit.edu), [wilfried.jakob@kit.edu](mailto:wilfried.jakob@kit.edu)

jeden Donnerstag 14:00-15:30 Uhr, Nusselt-Hörsaal

## Organisatorisches

1. Einführung und Motivation
  - 1.1 Warum benötigt man Computational Intelligence?
  - 1.2 Grundlegende Begriffe und Definitionen
  - 1.3 Motivierende Beispiele
  - 1.4 Literaturhinweise
  - 1.5 Überblick über die Vorlesung

# Organisatorisches (1)

- Vorlesung "Computational Intelligence":
  - immer im Wintersemester
  - Inhalt:
    - Fuzzy-Systeme (Mikut/Reischl)
    - Künstliche Neuronale Netze (Mikut/Reischl)
    - Evolutionäre Algorithmen (Jakob/Mikut)
  - 2 SWS / 4 ECTS
  - Prüfung (schriftlich): 20.03.2019 16:00-17:00 Uhr, Gerthsen-HS
- Vorlesung „Datenanalyse für Ingenieure“
  - immer im Sommersemester
  - 3 SWS (2 SWS Vorlesung, 1 SWS Rechnerübungen) / 5 ECTS
  - Prüfung (schriftlich) am 19.02.2019 16:00-17:00 Uhr, Gaede-HS

# Organisatorisches (2)

Schwerpunkte für beide Vorlesungen im Studiengang Maschinenbau:

- Bachelor und Master:
  - SP 05: Berechnungsmethoden im Maschinenbau
  - SP 18: Informationstechnik
  - SP 31: Mechatronik
- nur Master:
  - SP 01: Advanced Mechatronics
  - SP 04: Automatisierungstechnik
  - SP 22: Kognitive Technische Systeme
  - SP 32: Medizintechnik
  - SP 40: Robotik
  - Wahlfach

- Vorlesung wöchentlich, außer
  - 1.11.2018 (Feiertag)
  - 27.12.2018
  - 3.1.2019
- Übungen: Zwei Übungen
  - 1x Fuzzy im Nusselt-HS (auf dem Termin der Vorlesung)
  - 1x Neuronale Netze (praktische Aufgabe mit SciXMiner) im SCC-Pool, Selbststudium mit Angebot für Konsultation, Ankündigung folgt
- Prüfungen:
  - mündlich (bei  $\leq 40$  Teilnehmern) oder schriftlich (bei mehr als 40 Teilnehmern)
  - Dauer: 30 min (mündlich) oder 60 min (schriftlich)
  - Hilfsmittel: keine
  - Schwerpunkte:
    - immer auf letzter Folie (nicht im Skript)
    - besonders wichtig ist Fähigkeit zum Transfer auf praktische Probleme

# Organisatorisches (4)

- ILIAS-Teilnehmerliste zur Information
- Downloads:  
<http://ilias.studium.kit.edu>
  - alle Vorlesungen (Powerpoint)
  - Übungsanleitungen
  - Symbolverzeichnis(können aber u.U. auch während des Semesters ergänzt oder korrigiert werden!)
- Hinweis: nicht alle Informationen enthalten: weitere Beispiele an Tafel, Kurzdemos Software
- dort außerdem: Aktuelle Terminhinweise usw.

**FEEDBACK ERWÜNSCHT!**



The screenshot shows the ILIAS (Institutional Learning Information System) interface. At the top, the KIT logo and 'ILIAS' are visible. Below the navigation bar, the course title 'Computational Intelligence' is displayed, along with the schedule 'Do 14.00-15.30 Nusselt-HS'. A menu bar includes 'Inhalt', 'Info', 'Einstellungen', 'Mitglieder', 'Lernfortschritt', and 'Metadaten'. Below this, there are links for 'Zeigen', 'Verwalten', 'Sortierung', and 'Seite gestalten'. The main content area, titled 'Inhalt', lists three items: 'Linkliste' (with a globe icon), 'Übungen' (with a folder icon) which includes the text 'Hier finden Sie Material zur den Übungen', and 'Vorlesungsmaterial' (with a folder icon) which includes the text 'Hier finden Sie Material zur Vorlesung'.

- PDF mit den wichtigsten Bezeichnern zu allen Folien, dort ist aus Platzgründen nicht jedes Formelzeichen erklärt
- kompatibel zu Buch

MIKUT, R.: Data Mining in der Medizin und Medizintechnik.  
Universitätsverlag Karlsruhe; 2008

- Download möglich
- bitte Feedback, wenn etwas fehlt

## Symbolverzeichnis zur Vorlesung Computational Intelligence

Ralf Mikut, Markus Reischl, Wilfried Jakob  
Karlsruher Institut für Technologie  
E-Mail: ralf.mikut@kit.edu

Beim hier gewählten Bezeichnungsapparat wurde ein Kompromiss zwischen einheitlichen und durchgängigen Bezeichnungen einerseits sowie literaturkonformen und einfachen Bezeichnungen andererseits gewählt. Dabei handelt es sich um eine gekürzte Fassung von [1].

Symbol	Bezeichnung
$A1-3$	Neuronen in Ausgabeschicht
$a_i$	Vektor der Parameter der Zugehörigkeitsfunktionen aller Terme des Merkmals $x_i$
$a_{i,j}$	Parameter der Zugehörigkeitsfunktion des Terms $A_{i,j}$ ( $i = 1$ : rechtes Maximum Trapez-ZGF, $i = m_i$ : linkes Maximum Trapez-ZGF, $i = 2, \dots, m_i - 1$ : Maximum Dreieck-ZGF)
$A_{i,j}$	$i$ -ter linguistischer Term des $i$ -ten Merkmals $x_i$
$A_{i,j,n}$	ODER-Verknüpfung linguistischer Terme des $i$ -ten Merkmals $x_i$ in der Teilprämisse der $n$ -ten Regel
$b, b_i$	Parameter
$B_c$	$c$ -ter linguistischer Term der Ausgangsgröße $y$
$c$	Laufindex für Klassen
$C_r$	Konklusion der $r$ -ten Regel
COG	Schwerpunktmethode (Center of Gravity)
COGS	Schwerpunktmethode für Sineklons (Center of Gravity for Sineklons)

## Organisatorisches

### **1. Einführung und Motivation**

#### **1.1 Warum benötigt man Computational Intelligence?**

#### 1.2 Grundlegende Begriffe und Definitionen

#### 1.3 Motivierende Beispiele

#### 1.4 Literaturhinweise

#### 1.5 Überblick über die Vorlesung



**WSEIO KNÖNEN SIE DEIESN STAZ LSEEN,  
OWHBOHL DIE BCUHSTAEBN NCIHT  
IN DER RITHCIEGN RIEHENFOGLE SHETEN?**

# Warum benötigt man Computational Intelligence?

- Lösen von Aufgaben,
  - bei denen kein wissenschaftlich begründetes mathematisches Modell ("White-Box-Modell") oder Lösungsverfahren vorliegt, die der Mensch aber trotzdem irgendwie kann:
    - durch Lernen von Beispielen und/oder
    - durch Versuch und Fehler und/oder
    - regelbasiertes "Expertenwissen"
  - bei denen andere Verfahren keine befriedigende Lösung finden (schlechte Lösungsqualität oder zu lange Rechenzeiten)
- ACHTUNG!  
Computational Intelligence ist immer nur eine Variante unter anderen, nicht immer die Beste!

## Organisatorisches

- 1. Einführung und Motivation
  - 1.1 Warum benötigt man Computational Intelligence?
  - 1.2 Grundlegende Begriffe und Definitionen**
  - 1.3 Motivierende Beispiele
  - 1.4 Literaturhinweise
  - 1.5 Überblick über die Vorlesung

Begriffsklärung (leo.org):

- Computational:
  - berechenbar
- Intelligence:
  - Intelligenz
  - Einsicht
  - Wissen
  - Auffassungsvermögen
  - Information
  - Klugheit
  - Verstand
  - Geheimdienst
  - Geheimdienstinformationen

- Computational Intelligence:
  - umfasst alle Gebiete, die sich mit der Nachbildung biologischer und menschlicher Problemlösungsstrategien im Computer beschäftigen
  - beinhaltet mehrere Teilgebiete:
    - Fuzzy-Systeme:  
Nachbildung menschlicher regelbasierter Strategien im Computer, Verarbeitung von unscharfen Zugehörigkeitsgraden zwischen 0 und 1
    - Künstliche Neuronale Netze:  
Nachbildung der neuronengebasierten Informationsverarbeitung von Menschen und Tieren in Computern, inkl. Deep Learning
    - Evolutionäre Algorithmen (EA):  
Nachbildung der Veränderungs- und Selektionsprozesse in der Natur
    - Data Mining:  
Finden gültiger, neuer, nützlicher und verständlicher Zusammenhänge (Wissen) aus Daten
  - ACHTUNG! In der Literatur widersprüchliche und überlappende Definitionen, weitere Gebiete werden einbezogen oder nicht, z.T. andere verwandte Begriffe

- Hybride CI-Systeme
  - Neuro-Fuzzy-Systeme
  - Evolutionär optimierte Fuzzy-Systeme
  - Evolutionär optimierte Künstliche Neuronale Netze
- Andere naturanaloge Verfahren ("Metaheuristiken") :
  - Partikelschwarmoptimierung - PSO
  - Ameisenalgorithmen bzw. Ant Colony Optimization ACOsiehe z.B. [Kroll 13]
- Memetische Algorithmen (Kombination aus Evolutionären Algorithmen und lokalen linearen Suchverfahren)

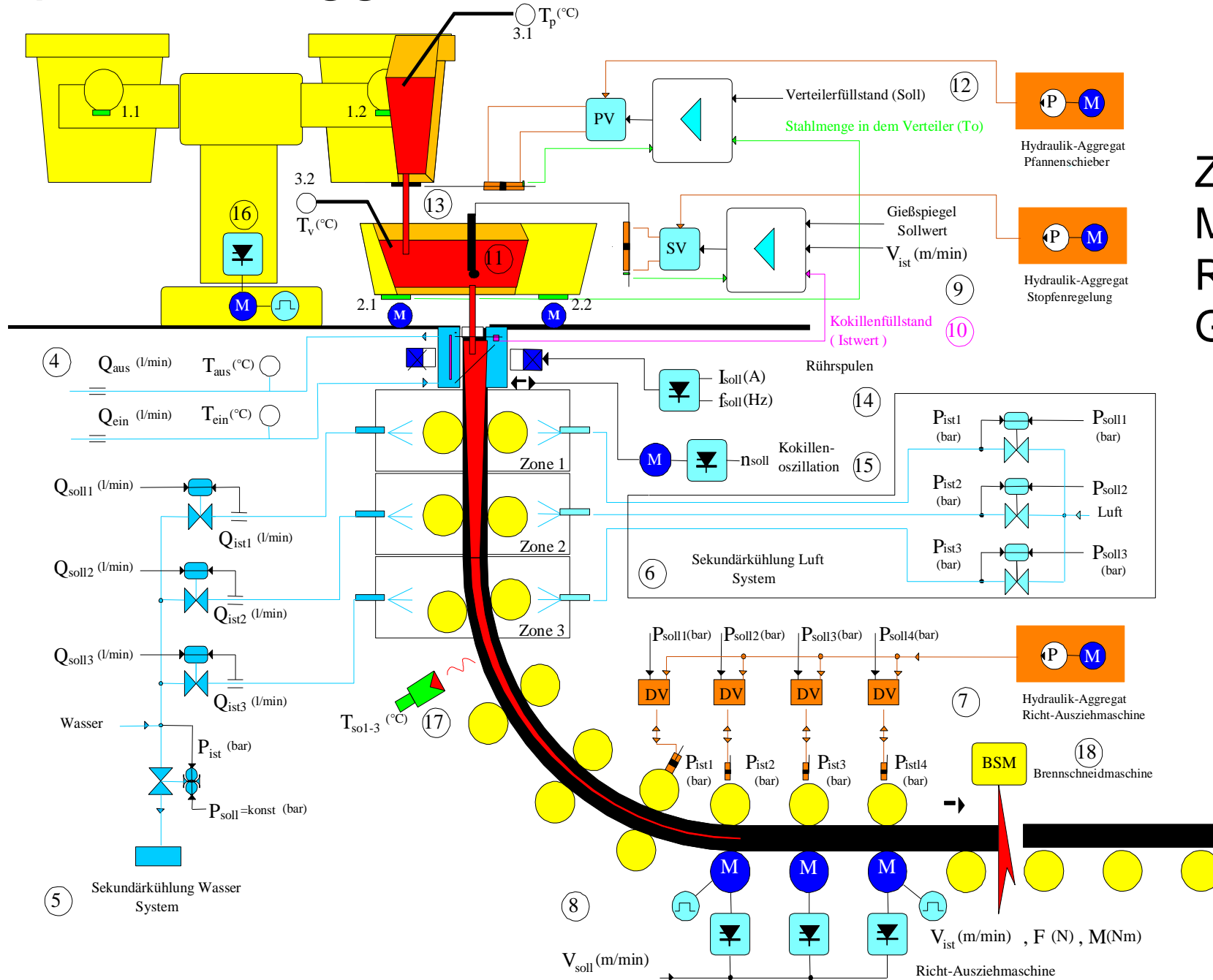
- Soft computing:
  - teilweise Synonym zu Computational Intelligence
  - Aufzählung Fuzzy, Neuronale Netze, Evolutionäre Algorithmen
  - Betonung approximativer Lösung für schwer berechenbare Probleme
- Künstliche Intelligenz (engl. Artificial Intelligence)
  - stärkere Betonung regelbasierter Entscheidungsprozesse
  - verwendet meist Symbole
  - basiert auf klassischer binärer Logik
  - in letzter Zeit wieder allgemeiner interpretiert, insbesondere für autonome Entscheidungen von Maschinen
    - symbolische Welt: Go, Schach usw.
    - subsymbolische Welt: autonomes Fahren, Sprachassistenten usw.

## Organisatorisches

- 1. Einführung und Motivation
  - 1.1 Warum benötigt man Computational Intelligence?
  - 1.2 Grundlegende Begriffe und Definitionen
  - 1.3 Motivierende Beispiele**
  - 1.4 Literaturhinweise
  - 1.5 Überblick über die Vorlesung



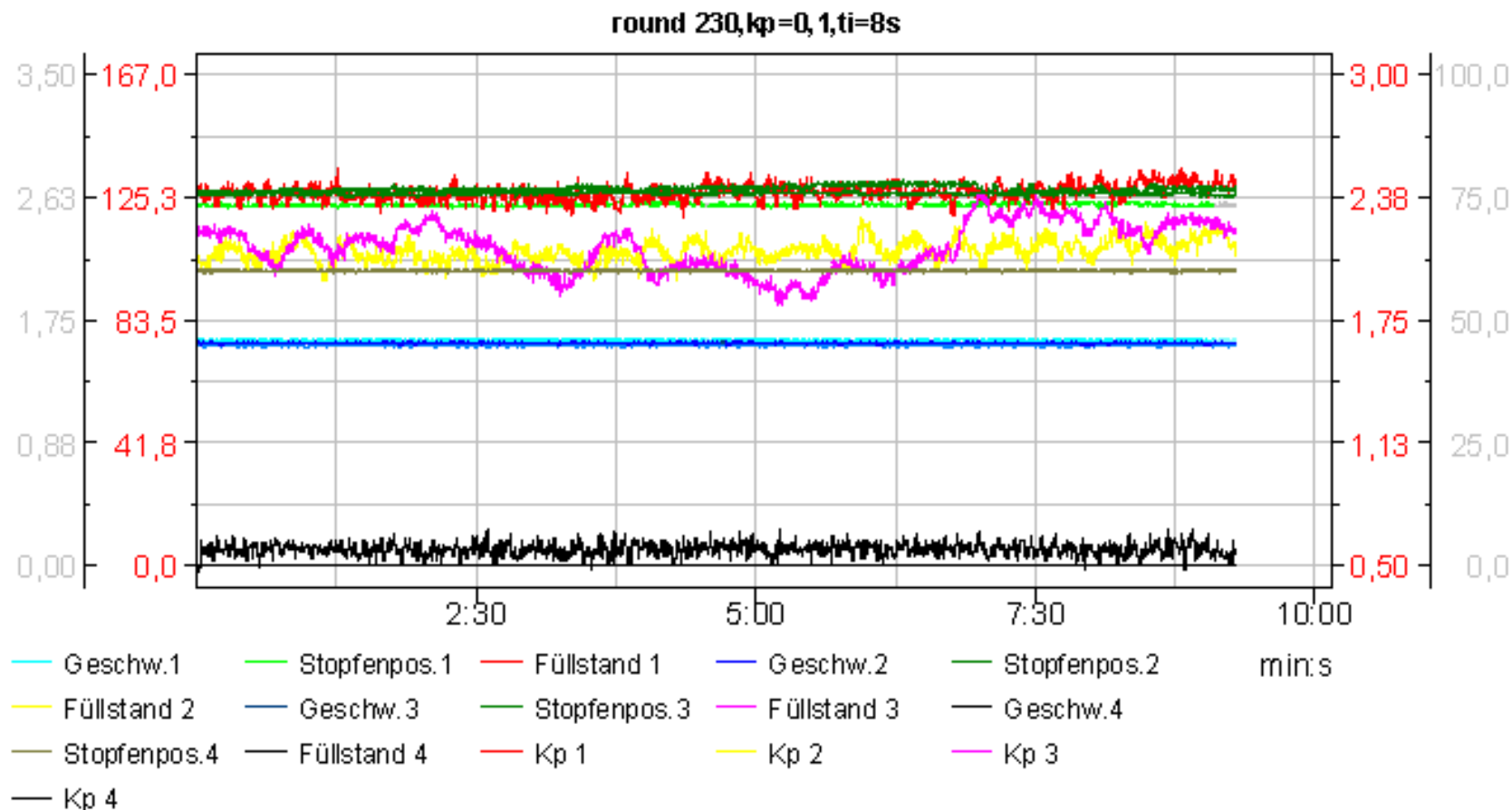
# Beispiel: Stranggießen



Ziel:  
Messung und  
Regelung  
Gießspiegel

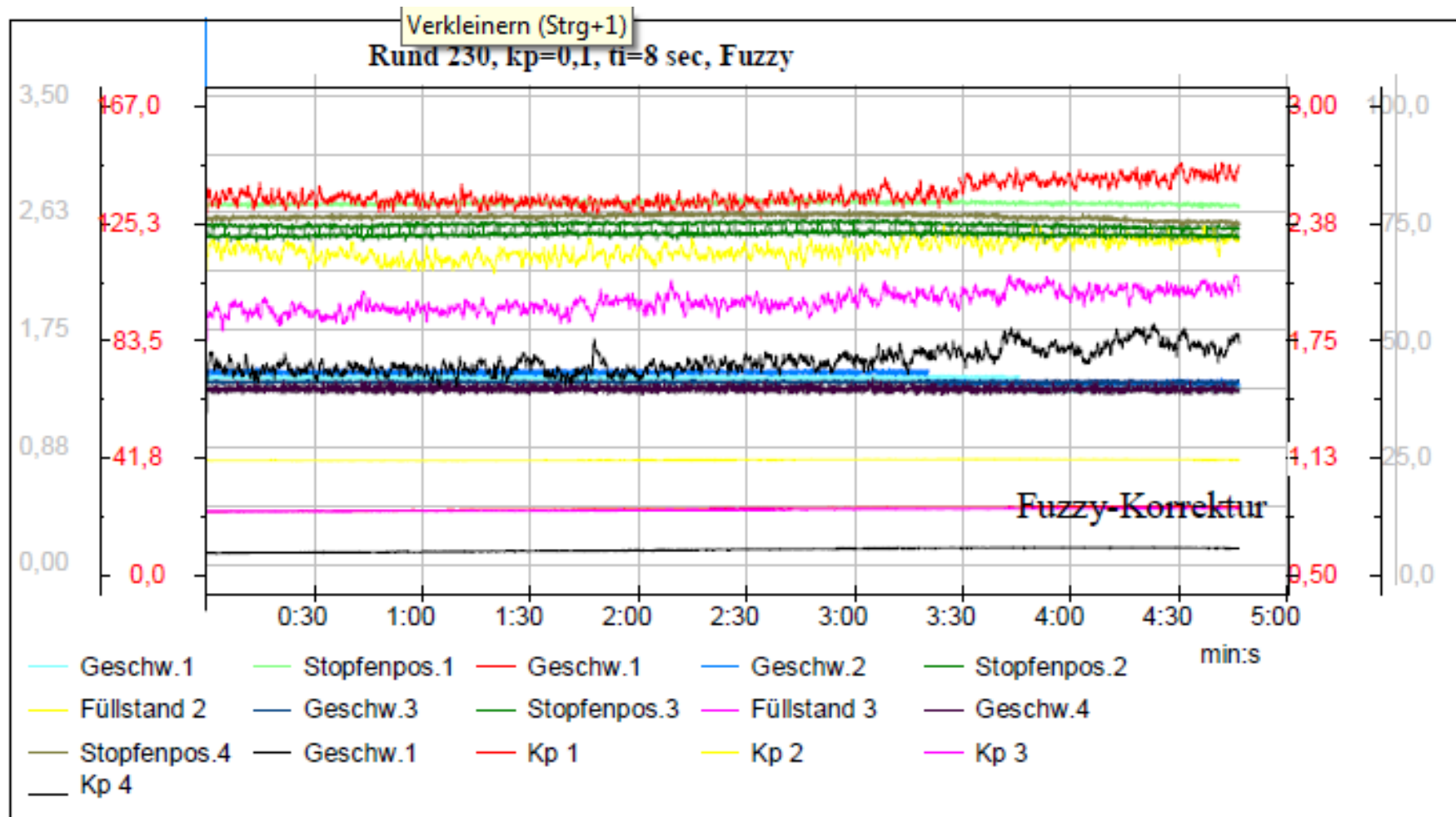
# Erprobung in Baotou, China: PID-Regler

- Verläufe der Gießspiegel und Stopfenpositionen für PID-Regler mit konstanten Parametern
- verschiedene Regelgüten wegen Abnutzungen bzw. Anlagerungen
- starke Schwankungen im Strang 3 (violett)

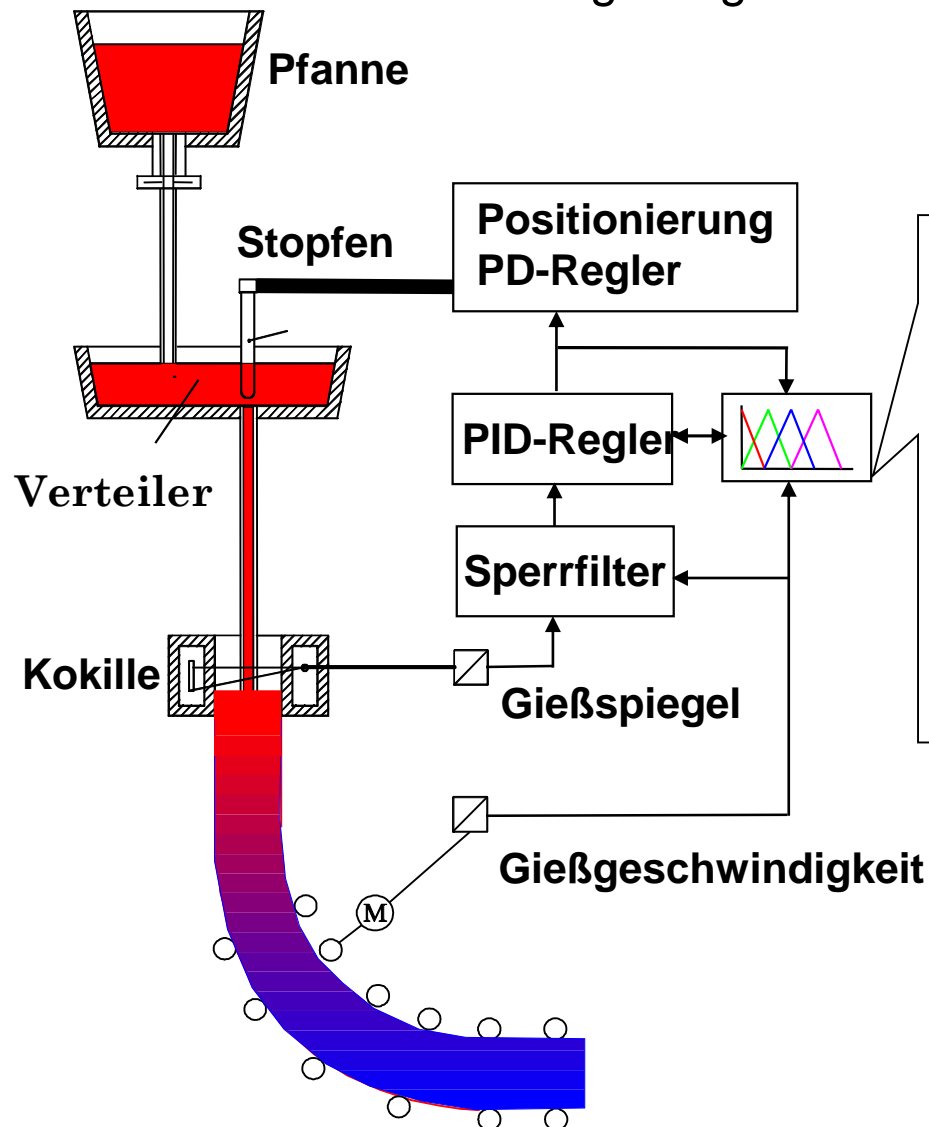
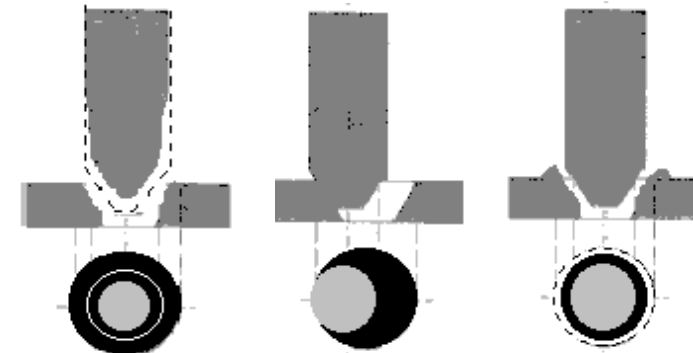


# Erprobung Baotou: Fuzzy-adaptiver PID-Regler

- Verläufe der Gießspiegel und Stopfenpositionen für Fuzzy-adaptive PID-Regler mit variablen Parametern
- vergleichbare Regelgüten und tolerable Schwankungen



Probleme beim Stopfen:  
Abnutzung, Exzentrizität,  
Ablagerungen



- Datenbank (Formate, Stahlsorten)
- Störungsüberwachung
- Start / Stop Gießen
- Fuzzy-Plausibilität
- Fuzzy-Adaption
- Fuzzy-Korrektur Filter
- Identifikation Stopfenkennlinie
- Störgrößenaufschaltung

Adaptionsblock mit Fuzzy-Komponenten

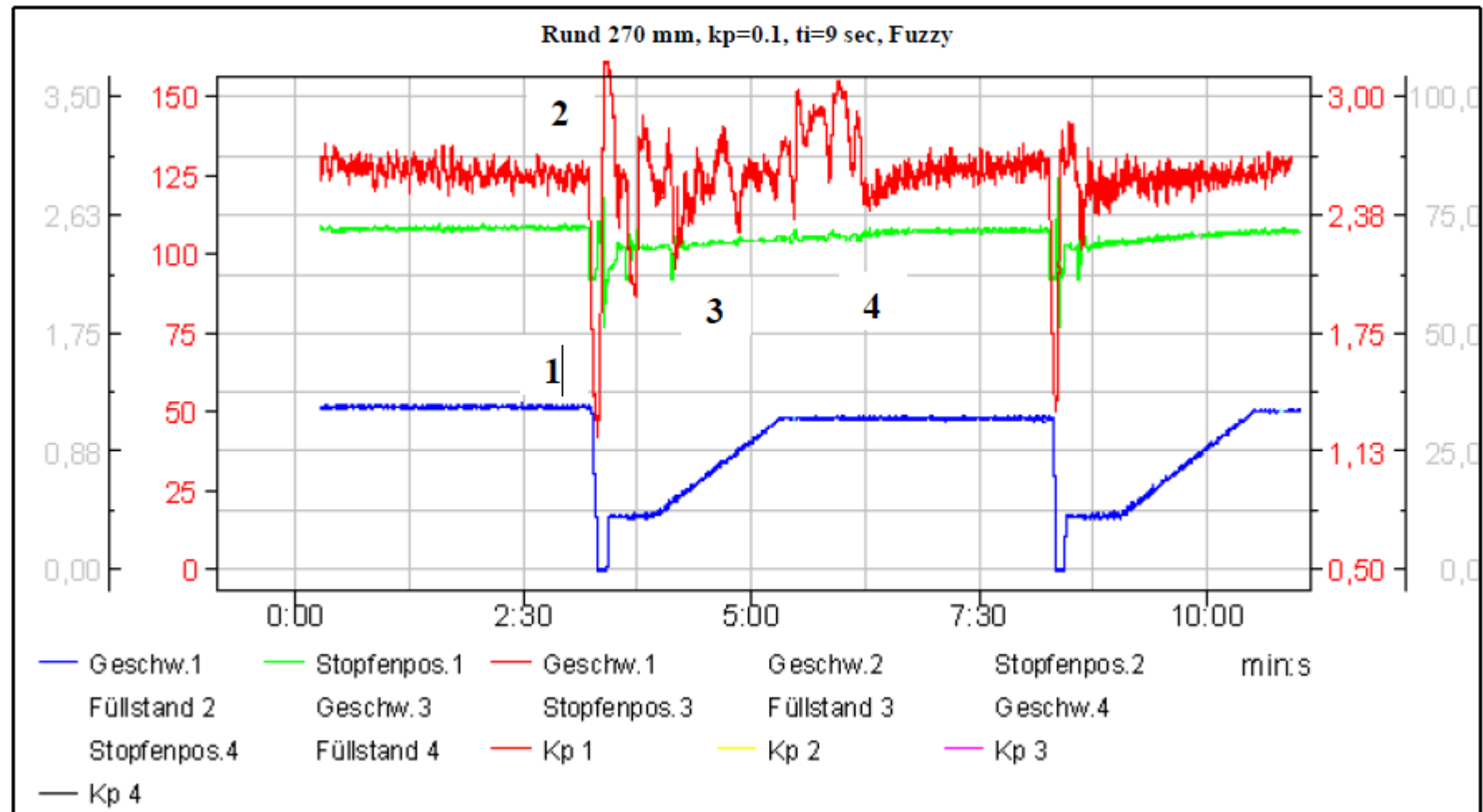
# Automatische Durchbrucherkennung

1 Automatische Erkennung eines drohenden Durchbruchs

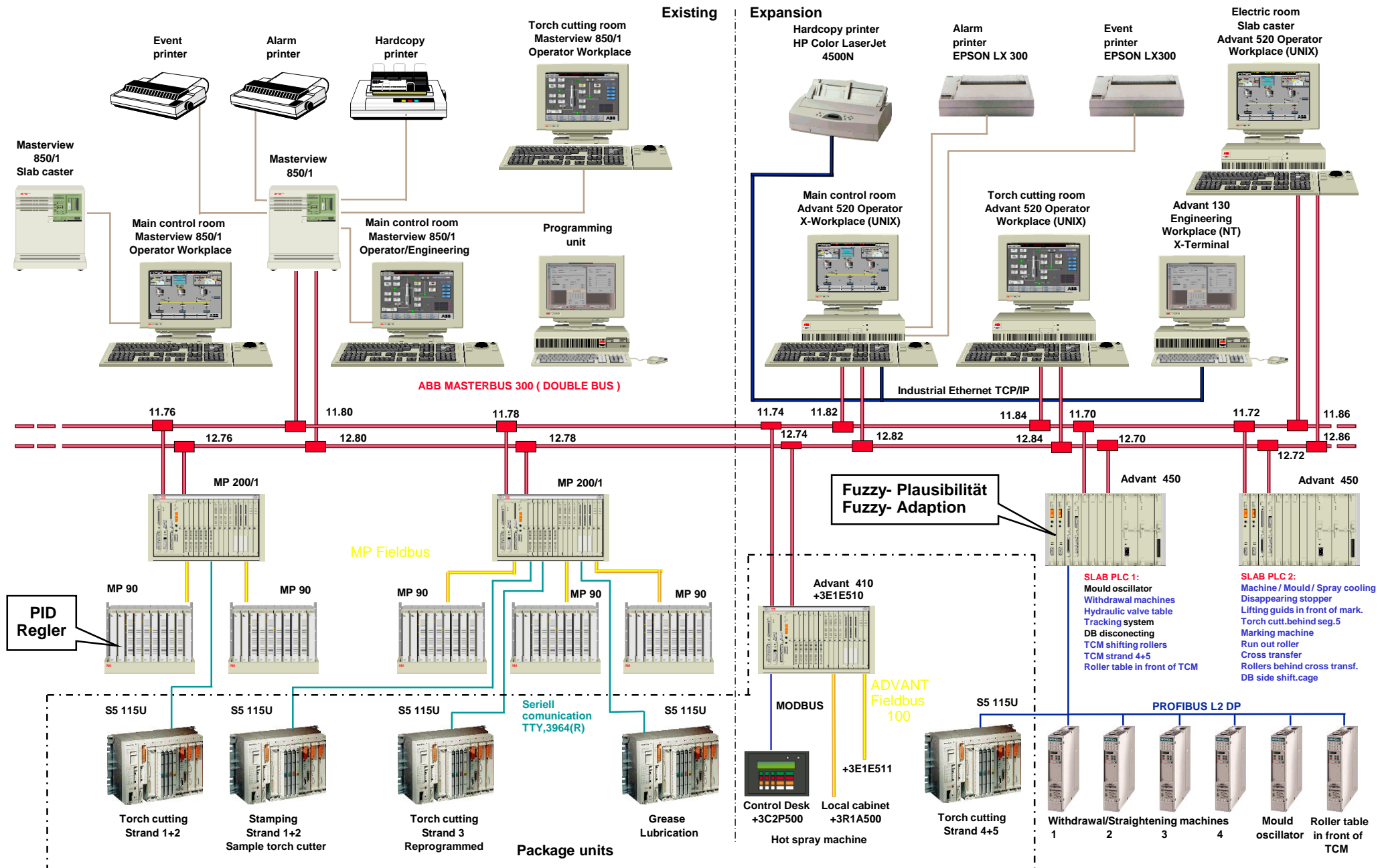
2 Alarm + Umschaltung auf Handbetrieb

3 Manuelle Stabilisierung

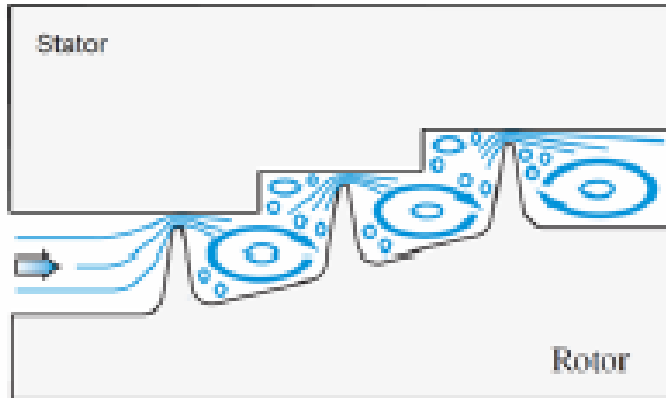
4 Umschaltung auf Automatikbetrieb



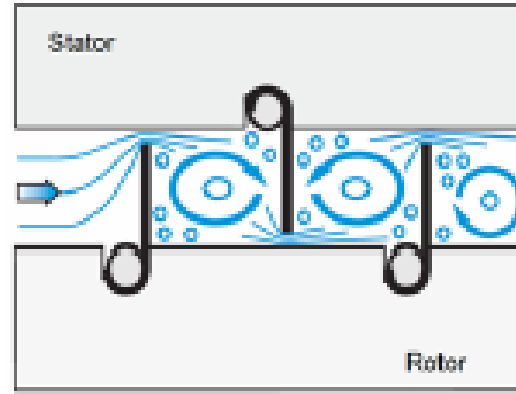
# Leitsystem der Kombianlage, Yieh United, Taiwan



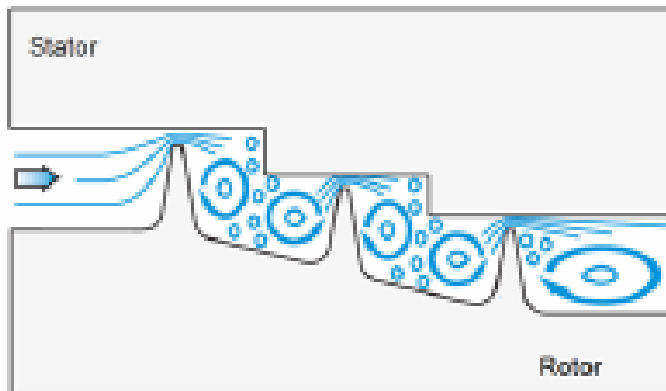
# Beispiel: Modellierung von Labyrinthdichtungen



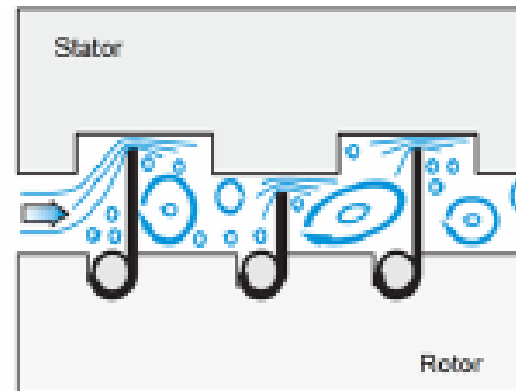
a) Divergentes Stufenlabyrinth



c) « Echtes » Labyrinth



b) Konvergentes Stufenlabyrinth



d) Kammnutlabyrinth

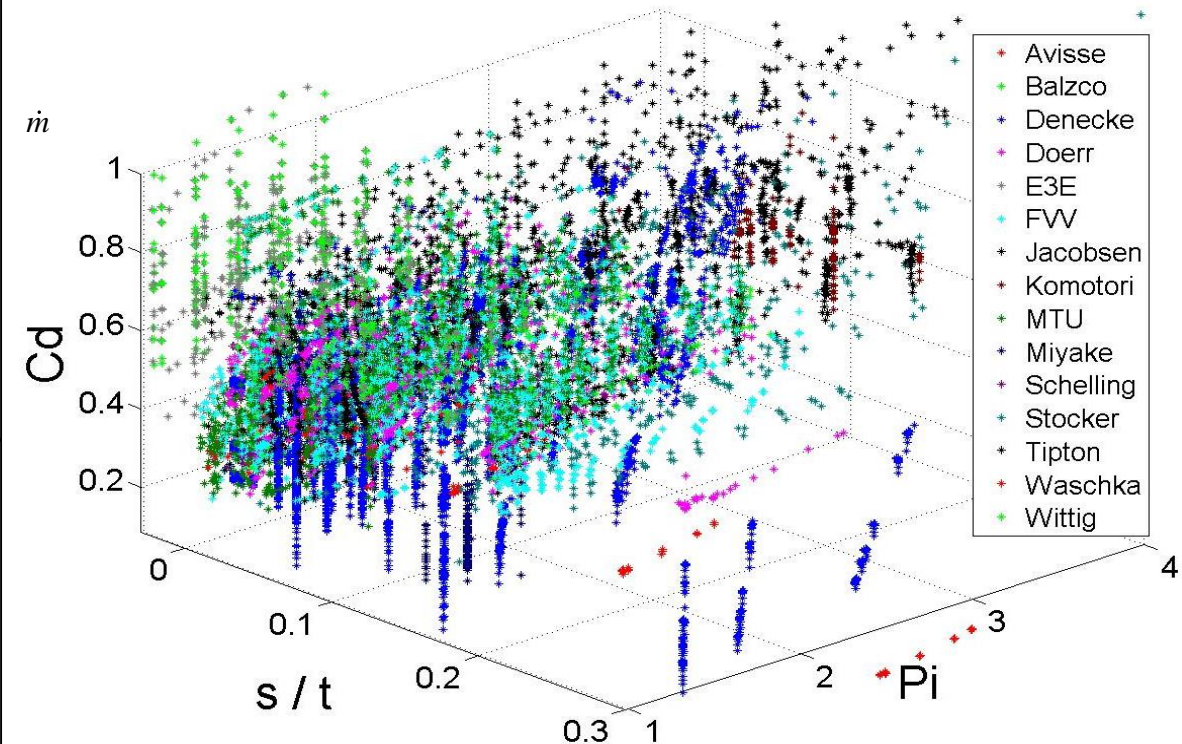
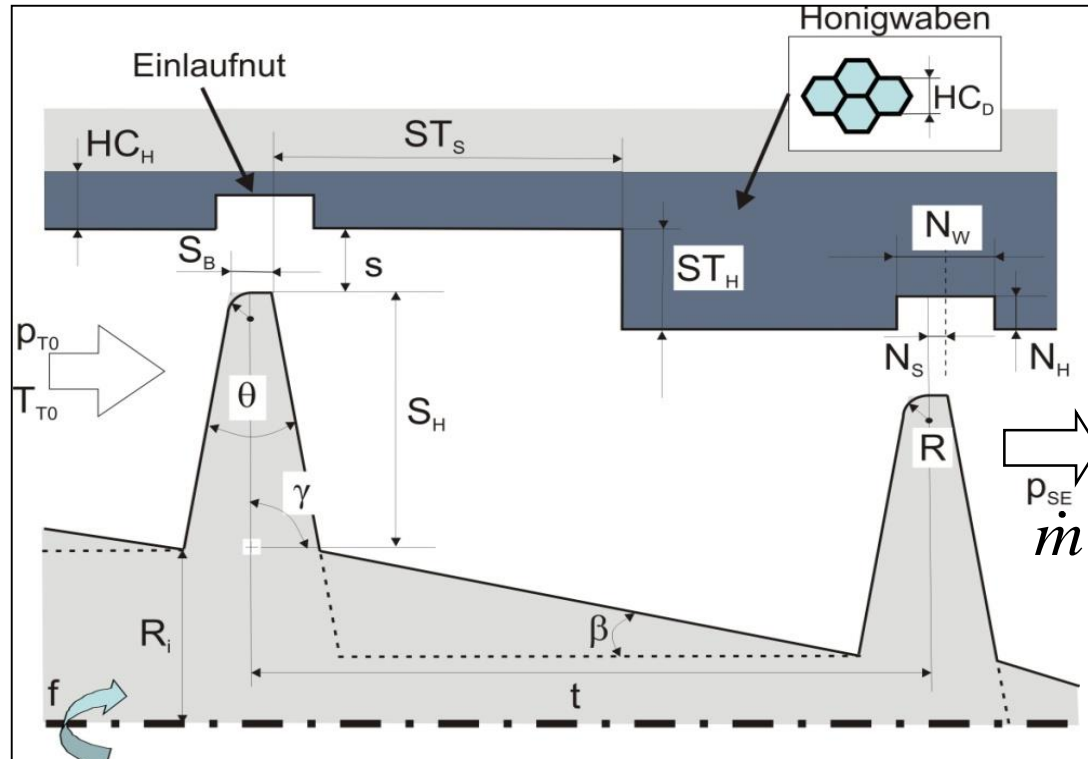
Bildquelle: Denecke, J.: *Dissertation*,  
Universität Karlsruhe (TH), 2007

- Kooperationsprojekt mit Institut für Thermische Strömungsmaschinen, KIT
- gegeben: Mess- und Literaturdaten verschiedener Labyrinthdichtungen
- Ziel: Vorhersage Durchflussbeiwert aus Geometrieparametern
- Vorgehensweise:
  - Datenvorverarbeitung
  - Merkmalsextraktion und -bewertung
  - Regression mit künstlichen Neuronalen Netzen



# Durchblick- und Stufenlabirynthe [Pychynski09,10]

## Datensammlung zum Durchflussverhalten von Durchblick- und Stufenlabirynthen:



**Komplexes System mit 21  
Einflussparametern**

**Datensatz mit 15.297 Datentupeln  
aus 15 Quellen**

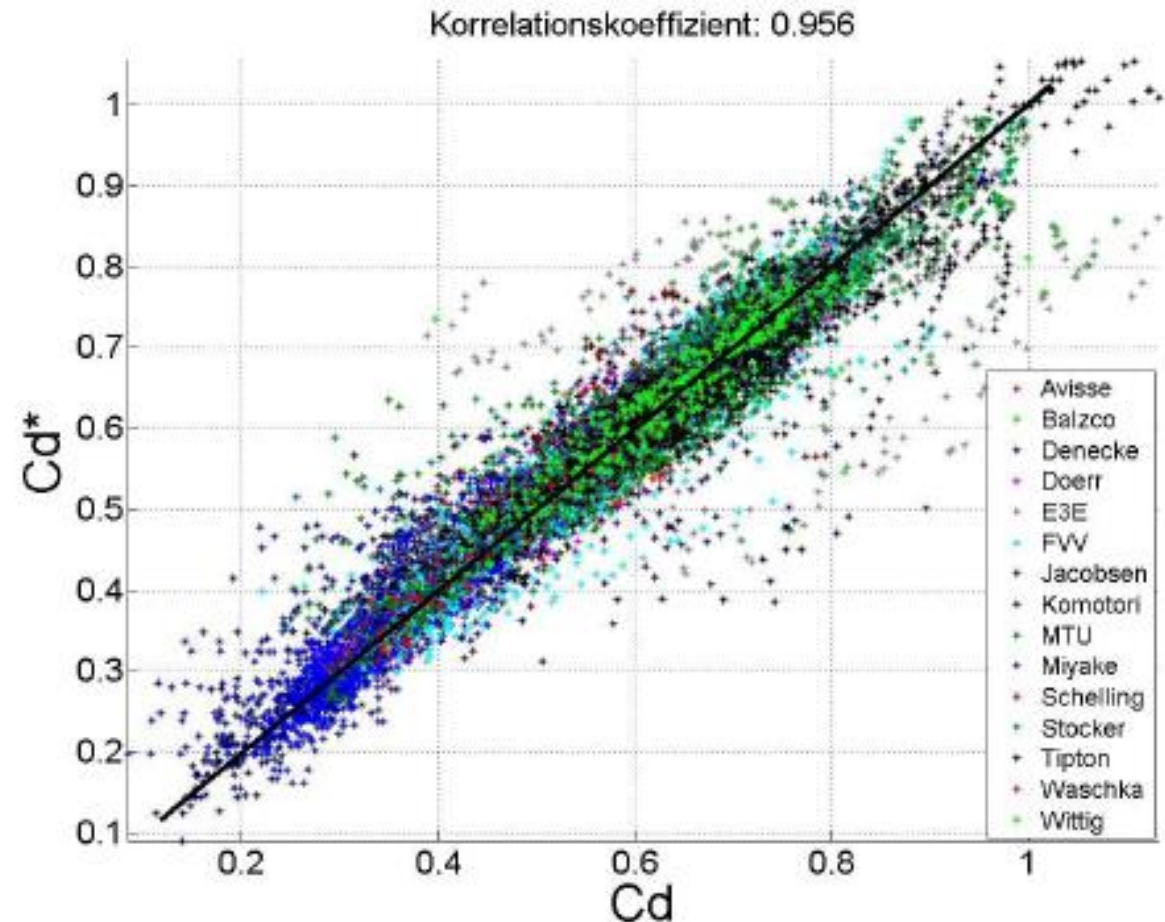
[Pychynski09] Pychynski, T.: Anwendung von Data Mining Methoden zur Analyse von Turbomaschinenkomponenten am Beispiel des Durchflussverhaltens von Labyrinthdichtungen. *Karlsruher Institut für Technologie (KIT)*, **2009**

[Pychynski10] Pychynski, T.; Blesinger, G.; Mikut, R.; Dullenkopf, K. & Bauer., H.-J.: Modelling the Labyrinth Seal Discharge Coefficient Using Data Mining Methods. *Proc., ASME TURBO EXPO; Glasgow*, **2010**



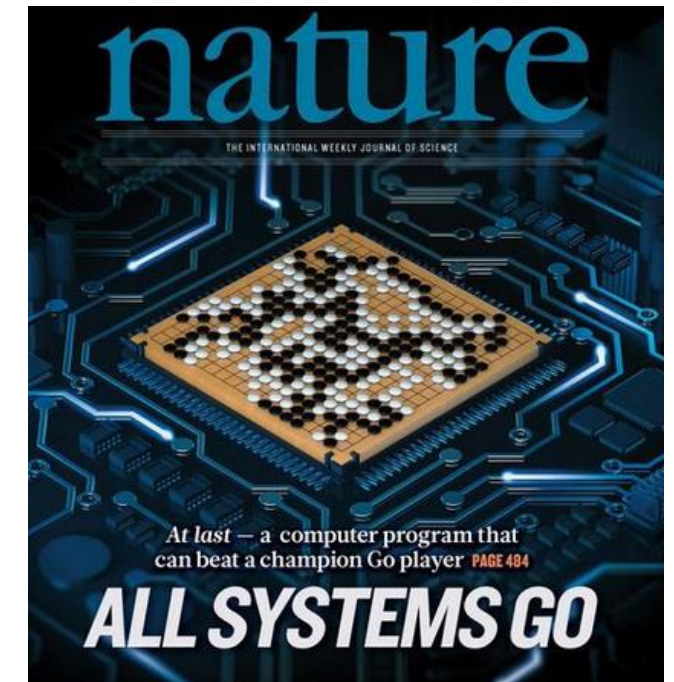
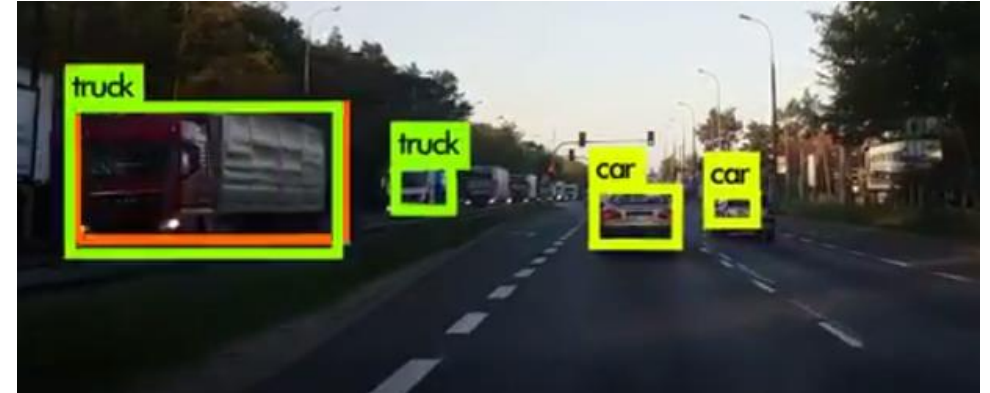
# Lösung [Pychynski09,10]

- Merkmalsauswahl in verschiedenen Varianten, gute Modelle ab 4 Merkmale
- Regressionsmodelle mit Polynomen und Künstlichen Neuronalen Netzen (Multi-Layer-Perceptron mit 30 Neuronen in der verdeckten Schicht )
- Polynome ab Grad 2 o.k., Neuronale Netze besser
- Korrelationskoeffizienten je nach Datensatz Regressionsansatz 0.95 - 0.99
- ACHTUNG! Modellgüte nur in der Nähe von existierenden Datentupeln gut, Probleme in schlecht abgedeckten Bereichen



# Deep Learning

- (zumindest teilweise berechtigter) Hype in den letzten Jahren
- Spektakuläre Erfolge:
  - Bilderkennung, wichtig auch für Autonomes Fahren, Autonome Systeme (Roboter) und Bildgestützte Qualitätskontrolle, siehe z.B. Yolo [Redmon17]  
<https://www.youtube.com/watch?v=yQwfDxBMtXg>
  - AlphaGo (Google DeepMind), Sieg gegen den Weltmeister 2016
  - Viele Versprechungen (Medizintechnik, ...)



# Beispiel: Walzwerkoptimierung

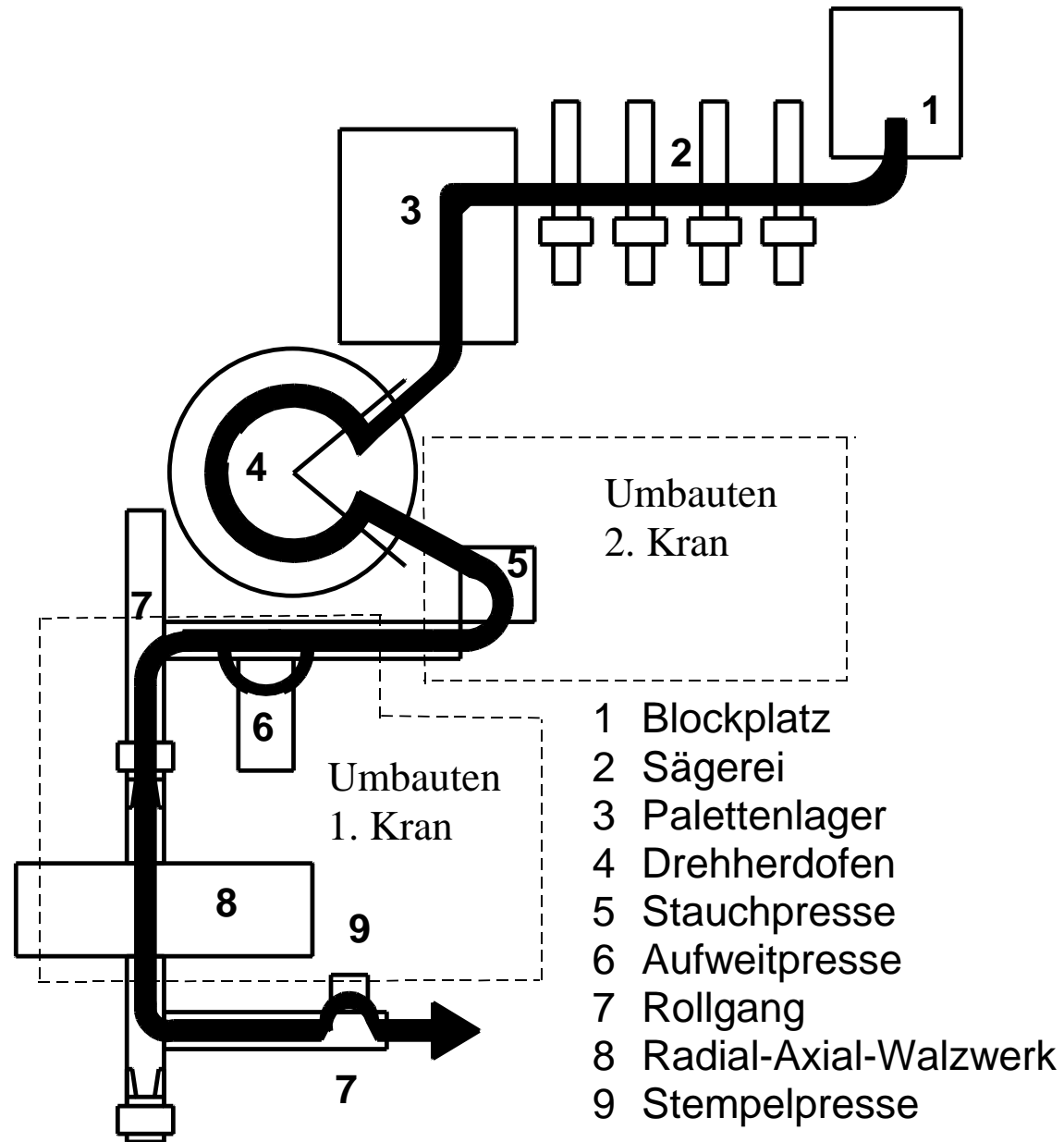
Problemstellung Ringwalzwerk:

- Herstellung von Ringen und Radreifen (reales Industriebeispiel)
- Aufgabe:  
Optimierung der Auftragsreihenfolge für ein Walzprogramm
  - Minimierung der Umbauzeit
  - Einhaltung der (Liefer-) Termine
  - Einhaltung technologischer Restriktionen (Koordinierung Säge, Drehherdofen, Walzen)





# Problemstellung Ringwalzwerk



# Problemstellung Ringwalzwerk

## Praktische Probleme:

- kleine Stückzahlen, damit häufige Umbauten notwendig (z.B. Wochenprogramm >100 Aufträge)
- jeder Auftrag verlangt andere Anlagenkonfiguration (eingebaute Werkzeuge)
- Alternativen zwischen Werkzeugen möglich
- keine Zwischenlagerung in der Anlage möglich (Arbeitstemperatur nach Drehherdofen, durchlaufendes Band)
- häufig zusätzliche Eilaufträge, die flexibel eingeplant werden müssen



# Herangehensweise für eine Optimierung

## Problemverstehen und Formalisieren

- Variante 1 (für Evolutionäre Algorithmen):
  - explizit als Gütekriterium  
(messbare Qualität für eine Lösung  
= häufig > 50% der Arbeit)

$$G_{ges} = \underbrace{\sum_{k=1}^{n-1} u_k(\mathbf{W}(\underline{s}))}_{\text{Umbauzeit}} + \underbrace{\sum_{k=2}^n v_k^{WP}(\underline{s})}_{\text{Strafterme für Restriktionen}} + \underbrace{\sum_{k=2}^n T_k(\underline{s})}_{\text{Terminbewertung}} \stackrel{!}{=} \underset{\underline{s}}{Min}$$

- Formalisierung einer möglichen Lösung  
(hier: Reihenfolge  $\underline{s}$  mit n Aufträgen)
- Variante 2:  
nur Lösungsweg formalisieren  
(z.B. als Experten-Regeln)

# Variante 1: Umbauzeiten

n	Sortiment	Walze	Walzdorn	Zentrierrollen	AWS	AWD	Tast-R.	Ges.	LD
1	Martensite Größe B	glatt	Dorn 02	02.38-(glatt)	ohne	ohne	40er	ohne	270
2	Martensite Größe C	glatt	Dorn 08	02.38-(glatt)	G1	155	40er	ohne	160
3	Martensite Größe C	glatt	Dorn 08	02.38-(glatt)	G2	175	40er	ohne	180
4	Martensite Größe C	glatt	Dorn 07	02.38-(glatt)	G2	175	40er	ohne	180
5	Martensite Größe C	glatt	Dorn 06	02.38-(glatt)	G2	175	40er	ohne	190
6	Martensite Größe C	glatt	Dorn 03	02.38-(glatt)	G2	175	40er SB	ohne	190
7	Martensite Größe A	glatt	Dorn 03	02.38-(glatt)	G2	175	40er SB	ohne	190
8	Martensite Größe C	glatt	Dorn 03	02.38-(glatt)	G2	175	40er SB	ohne	190

Umbau 2->1 Walzdorn (20) = 20 min

Umbau 1->2 Walzdorn (20) + evtl. AWS (20) + evtl. AWD (15) = 20...55 min

(je nach vorherigem Auftrag)

grün: Umbau mit 2. Kran im Schatten der Umbauzeit... (je 10 Minuten)

## Umbauzeiten in Minuten

	n=1	n=2	n=3	n=4	n=5	n=6	n=7
n=1:	0	55	55	55	55	65	65
n=2:	20	0	35	55	55	65	65
n=3:	20	35	0	20	20	30	30
n=4:	20	55	20	0	20	30	30
n=5:	20	55	20	20	0	30	30
n=6:	30	65	30	30	30	0	5
n=7:	30	65	30	30	30	5	0

# Variante 2: Expertenregeln...

hier: Regelbasis aus 24 Regeln

- Regel 1: Fasse als ersten Arbeitsschritt Aufträge mit gleichen Walzen zusammen.

...

- Regel 14: Drängt ein Termin zur Fertigung eines Auftrages, so ist dieser ungeachtet seines Umbauaufwandes einzuplanen.

## Vorteile Expertenregeln:

- spart aufwändige Modellierung
- u. U. höhere Akzeptanz bei Nutzer (Transparenz)

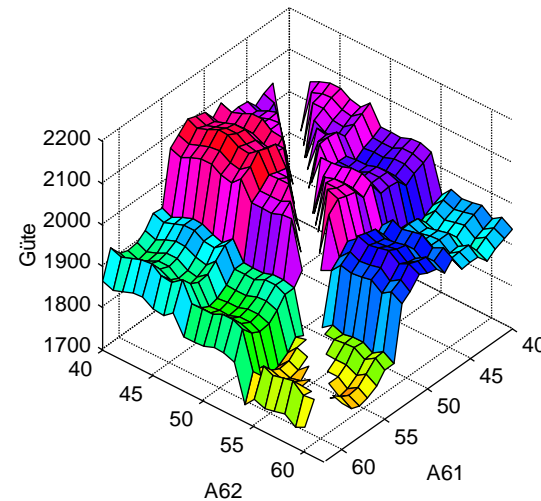
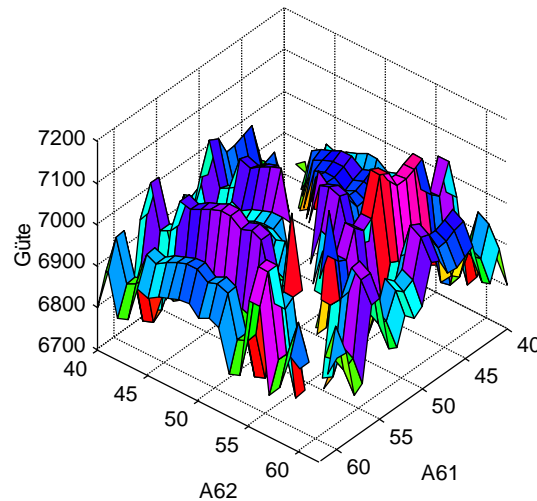
## Nachteile Expertenregeln:

- aufwändig zu bekommen, meist nur ein Teil der Regeln bewusst
- verschenkt meist bessere Ergebnisse



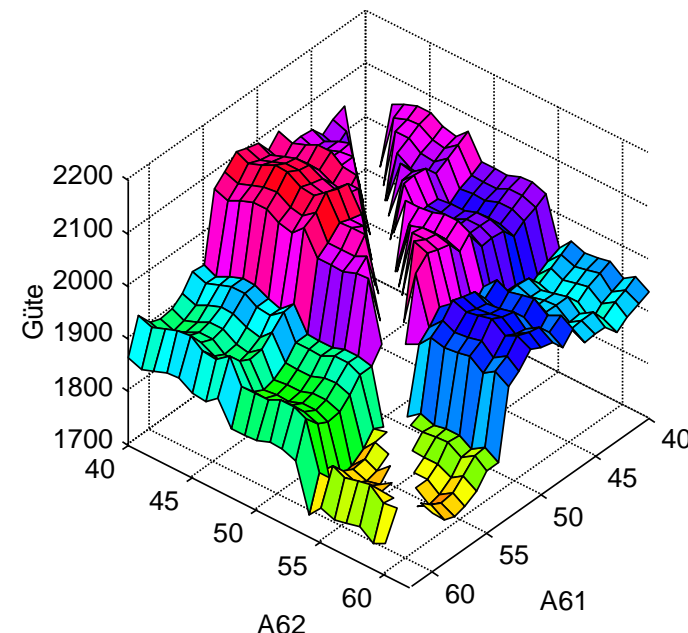
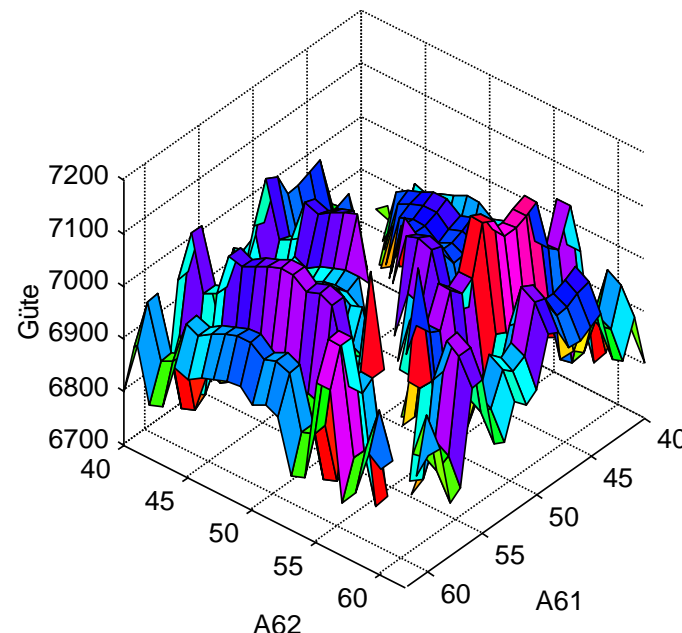
Ausschnitt aus dem Gütegebirge in Abhängigkeit der Positionen zweier Aufträge in einer Walzreihenfolge

- a. Startnäherung zufällige Reihenfolge (links)
- b. Startnäherung Expertenregeln (rechts)



# Ansatzpunkte für eine heuristische Optimierung

- Evolutionäre Algorithmen
- Vertauschen von Aufträgen mit hoher Umbauzeit mit allen anderen Aufträgen
  - neuer Ausgangspunkt: Minimum der Gütefunktion
  - Erweiterung: Einsortieren eines Auftragsblocks von 2, 3 und 4 Aufträgen an verschiedenen Stellen
  - Modifizierung eines Evolutionären Algorithmus: "Gezielte Mutation" (Memetischer Algorithmus, da Kombination mit lokaler Suche)



## Reales Wochenprogramm mit 135 Aufträgen

- beste Lösung: Expertenwissen + modifizierte Evolutionäre Algorithmen
- Rechenzeit: ca. 5 Minuten
- Einsparung gegenüber manueller Planung: ca. 15% Umbauzeit  
Restriktionen werden eingehalten
- andere Lösungen zu langsam und schlechter

## Organisatorisches

- 1. Einführung und Motivation
  - 1.1 Warum benötigt man Computational Intelligence?
  - 1.2 Grundlegende Begriffe und Definitionen
  - 1.3 Motivierende Beispiele
  - 1.4 Literaturhinweise**
  - 1.5 Überblick über die Vorlesung

- Kiendl, H.: Fuzzy Control. Methodenorientiert. Oldenbourg-Verlag, München, 1997
- S. Haykin: Neural Networks: A Comprehensive Foundation. Prentice Hall, 1999
- LeCun, Y.; Bengio, Y. & Hinton, G. Deep Learning Nature, Nature Publishing Group, 2015, 521(7553), 436-444
- Kroll, A. Computational Intelligence: Eine Einführung in Probleme, Methoden und technische Anwendungen. Oldenbourg Verlag, 2013
- Blume, C, Jakob, W: GLEAM - General Learning Evolutionary Algorithm and Method: ein Evolutionärer Algorithmus und seine Anwendungen. KIT Scientific Publishing, 2009 (PDF frei im Internet)
- Schwefel, H.-P.: Evolution and Optimum Seeking. New York: John Wiley, 1995
- Mikut, R.: Data Mining in der Medizin und Medizintechnik. Universitätsverlag Karlsruhe; 2008 (PDF frei im Internet) (<http://digbib.ubka.uni-karlsruhe.de/volltexte/1000008476>)

# Überblick über die Vorlesung

Vorlesung:

1. Einführung und Motivation
2. Fuzzy-Logik
3. Künstliche Neuronale Netze
4. Evolutionäre und Memetische Algorithmen