

Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 01 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



Bremer Institut für Messtechnik,
Automatisierung und
Qualitätswissenschaft

Motivation

- Kurzvorstellung des BIMAQ -



Methoden

Messsystemtheorie

- Modellierung und Simulation
- Unschärferelationen
- Messbarkeitsgrenzen

Messsystemtechnik

- Optische High-Speed-Messsysteme
- Multi-Sensor-Systeme
- Großkoordinatenmesssysteme

→ modellbasierte, dynamische Messsysteme

Anwendungen

Produktionstechnik & Materialwissenschaft

- Geometrische und Rauheitsmesstechnik
- Optische In-Prozess-Messtechnik
- Thermographie, Randzonenanalyse

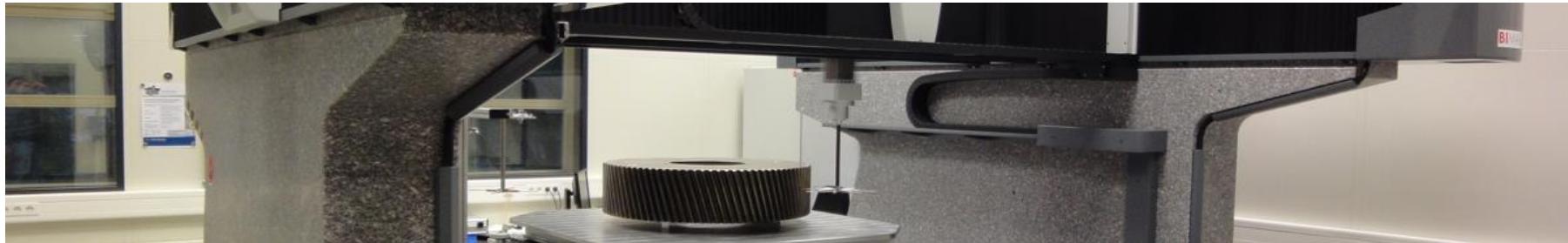
Windenergiesysteme & Strömungsprozesse

- Verzahnungsmesstechnik
- Getriebemesstechnik
- Strömungsmesstechnik



Universität Bremen

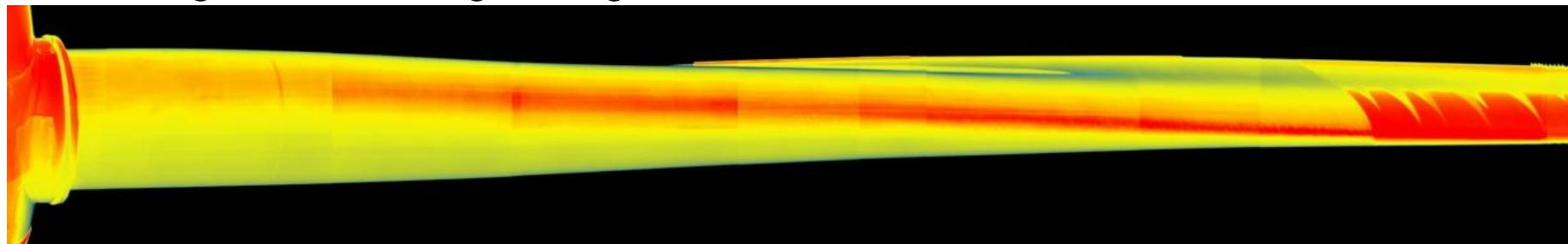
Labor für Großverzahnungen



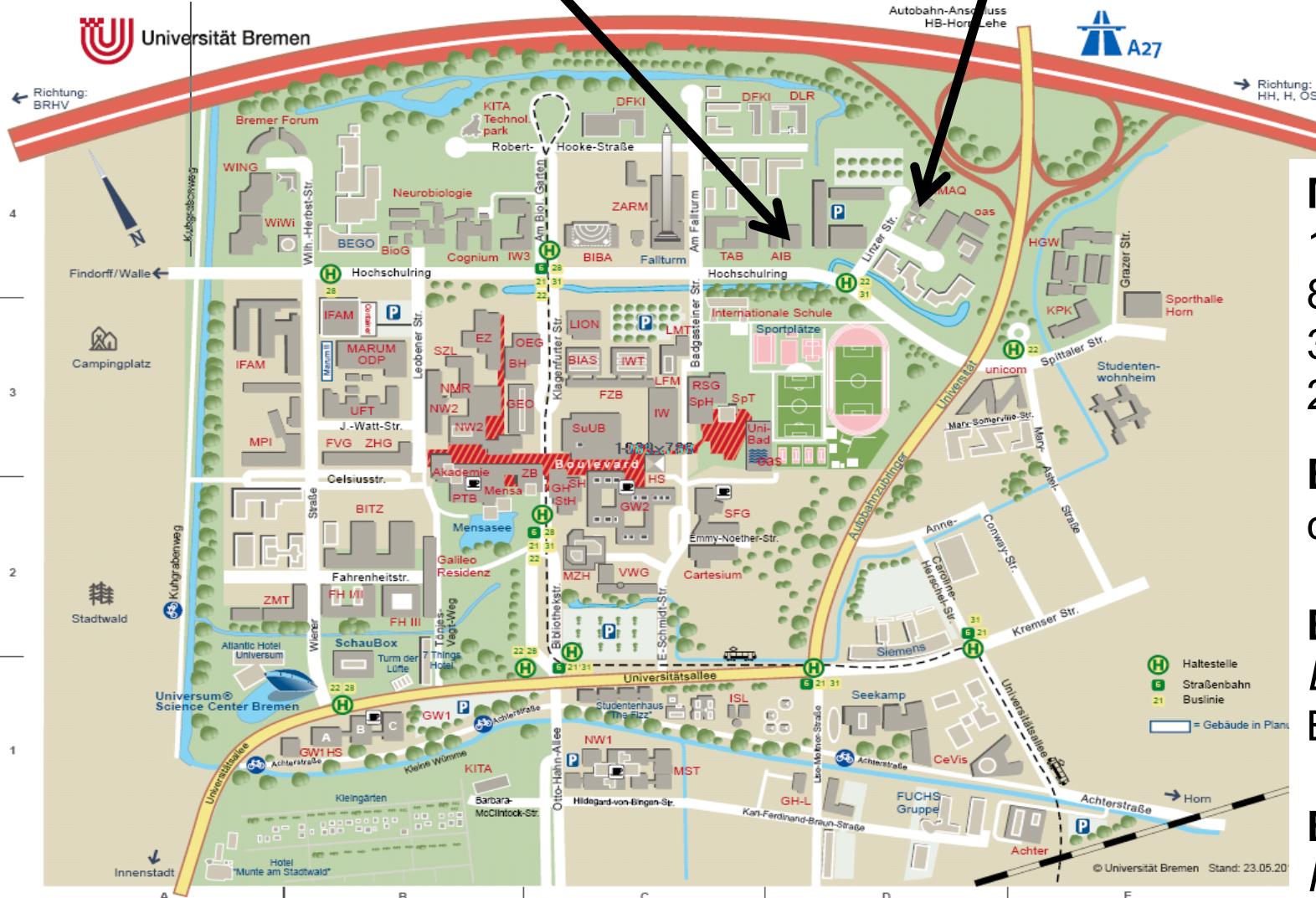
Labore für taktile geometrische Messverfahren und optische In-Prozess-Messtechnik



Forschung zu Windenergieanlagen



BIMAQ Technikum



BIMAQ



Mitarbeiter

15 Wissenschaftler/innen
8 Techniker
3 Verwaltung
20 Studierende

Budget

ca. 2,5 Mio. €/Jahr

BIMAQ-Hauptgebäude
Linzer Str. 13
Büros und Labore

BIMAQ-Technikum
Hochschulring 40
Großlabor und -versuche



Universität Bremen

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Aufbau der Lehrveranstaltung

Vorlesung	Mi 10 Uhr c.t., BIMAQ Auditorium, wöchentlich im Wechsel mit u.a. Übungen, Vorlesungsfolien
Übung	Mi 10 Uhr c.t., BIMAQ Auditorium Ü2 15.11.2017 und Ü6 31.1.2017 Mi 10 Uhr c.t., BIMAQ Linzer Str. 13 Raum 1010 Termine: Ü1 - 1.11.2017, Ü3 6.12., Ü4 20.12., Ü5 - 17.1.2017 Übungsheft mit Kurzlösungen (6 Übungen) , Musterklausur

StudIP:

Hinweis: PABO-Anmeldung

Klausur	120 min,	zugelassene Hilfsmittel: • keine
Modulnote		Konsultation: Termin wird noch bekannt gegeben (Klausurergebnis)

Lehrziele und Gliederung

V1 Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik

V2 Automatisierung in der Produktion

V3 Boolesche Algebra 1

Ü1 Matlab Einführung

V4 Boolsche Algebra 2: Graphen

Ü2 Übung Boolsche Algebra

V5 Fuzzy Logic

Ü3 Fuzzy Logic

V6 Neuronale Netze

Ü4 Neuronale Netze

V7 Automatisiertes Messen und Steuern

Ü5 Automatisiertes Messen und Steuern

V8 Speicherprogrammierbare Steuerungen

Ü6 Übungen und Musterklausuren



Themenfelder der Automatisierungstechnik

- Überblick -

Anwendungen:

- **Autom. von Fertigungskomponenten**
- **Autom. von Fertigungsbereichen**
- **Speicherprogrammierbare Steuerungen**
- Numerische Steuerungen CNC
- Robotersteuerungen
- Die Programmierung der Fertigungseinrichtungen
- **Netzwerke und Netzwerkkomponenten**
- Vernetzte Werkzeugmaschinen und Leitrechnersysteme
- Flex. Vorrichtungs- und Spannelemente
- Retrofitting von Werkzeugmaschinen
- Autom. eines Rapid Prototyping-Verfahrens
- Schaltelemente/ Messwertgeber / Antriebe

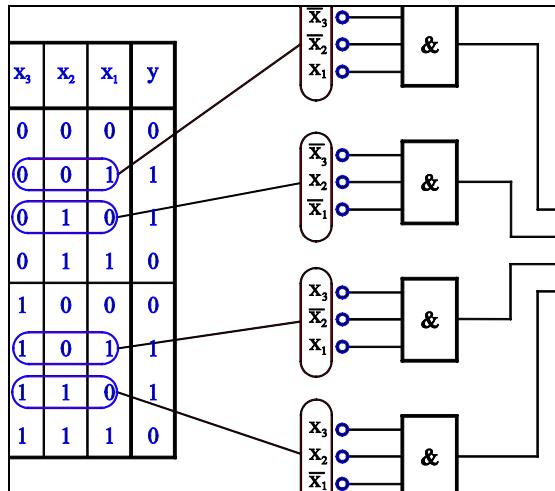
Methoden:

- Prozess- und Informationsdarstellung
- **Boolesche Algebra**
- **Graphen, Automaten**
- **Unscharfe Logik**
- **Neuronale Netze**
- **Automatisiertes Messen und Steuern**
- Rechnertechnik
- Datenmodelle
- Wissensbasierte Systeme
- Robotik, Koordinatentransformation und Kinematik
- Fehlermöglichkeits- und -einflussanalyse FMEA (Qualitätswissenschaft.)
- Diagnosetechniken

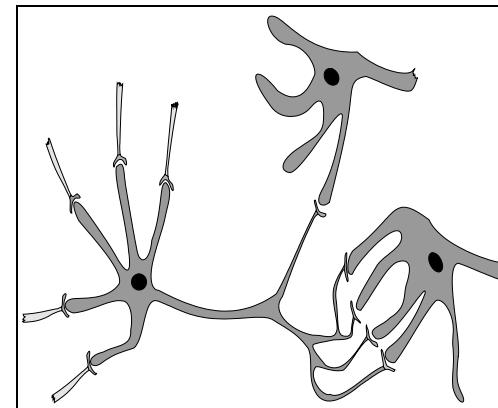
Anwendungsbereiche, Prozesse und Methoden

- Informationstechnische Anwendungen in der Automatisierungstechnik -

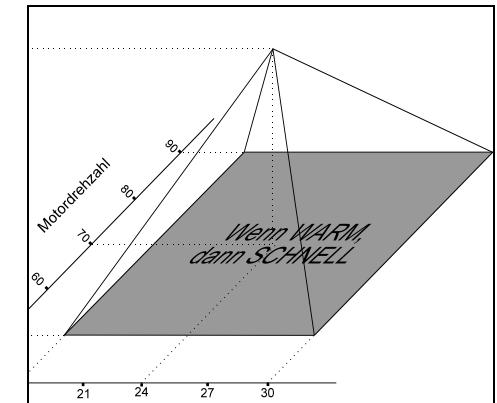
Boolesche Algebra, Graphen



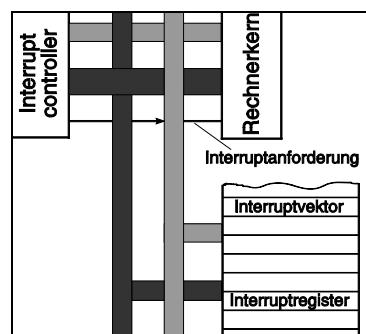
Neuronale Netze



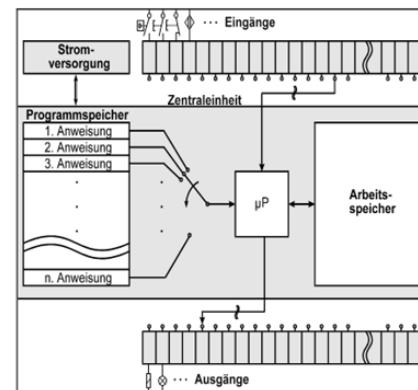
Unscharfe Logik



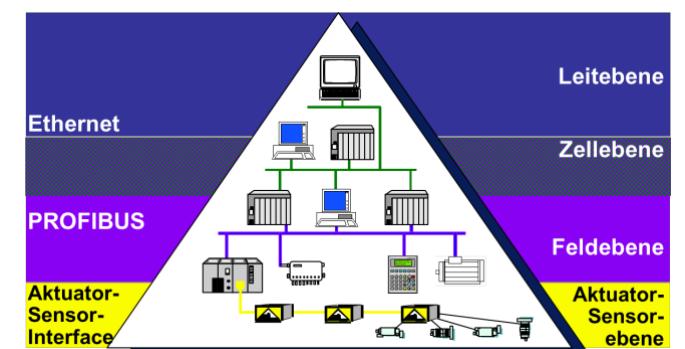
Automatisiertes Messen und Steuern



SPS Technologie



Automatisieren von Fertigungsbereichen



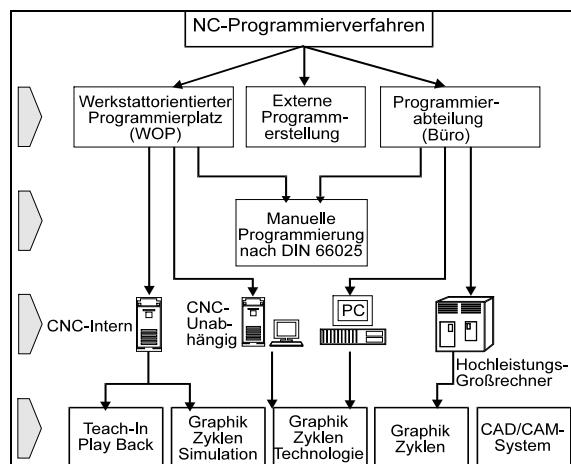
Anwendungsbereiche, Prozesse und Methoden

- Überblick -

Wissensbasierte System

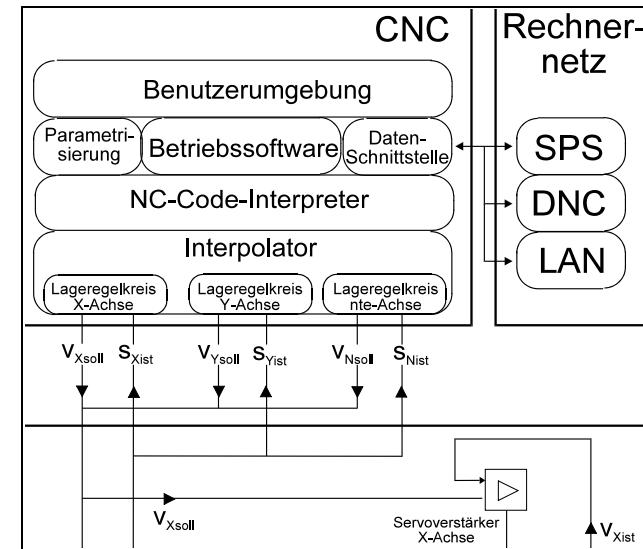


Die Programmierung der Fertigungseinrichtung

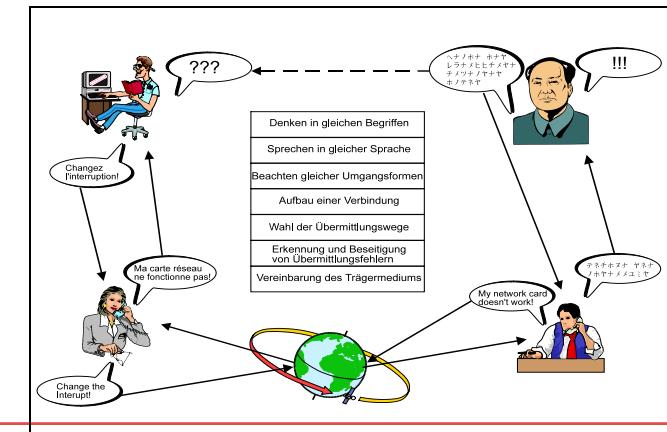
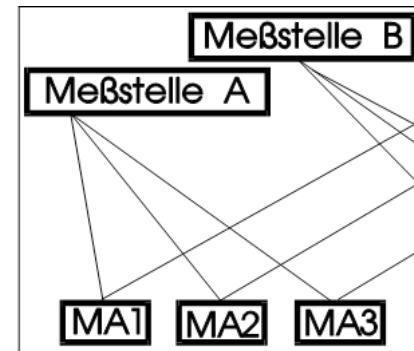


Industrielles Automatisieren mit Steuerungstechnik

Numerische Steuerungen CNC/ Robotik



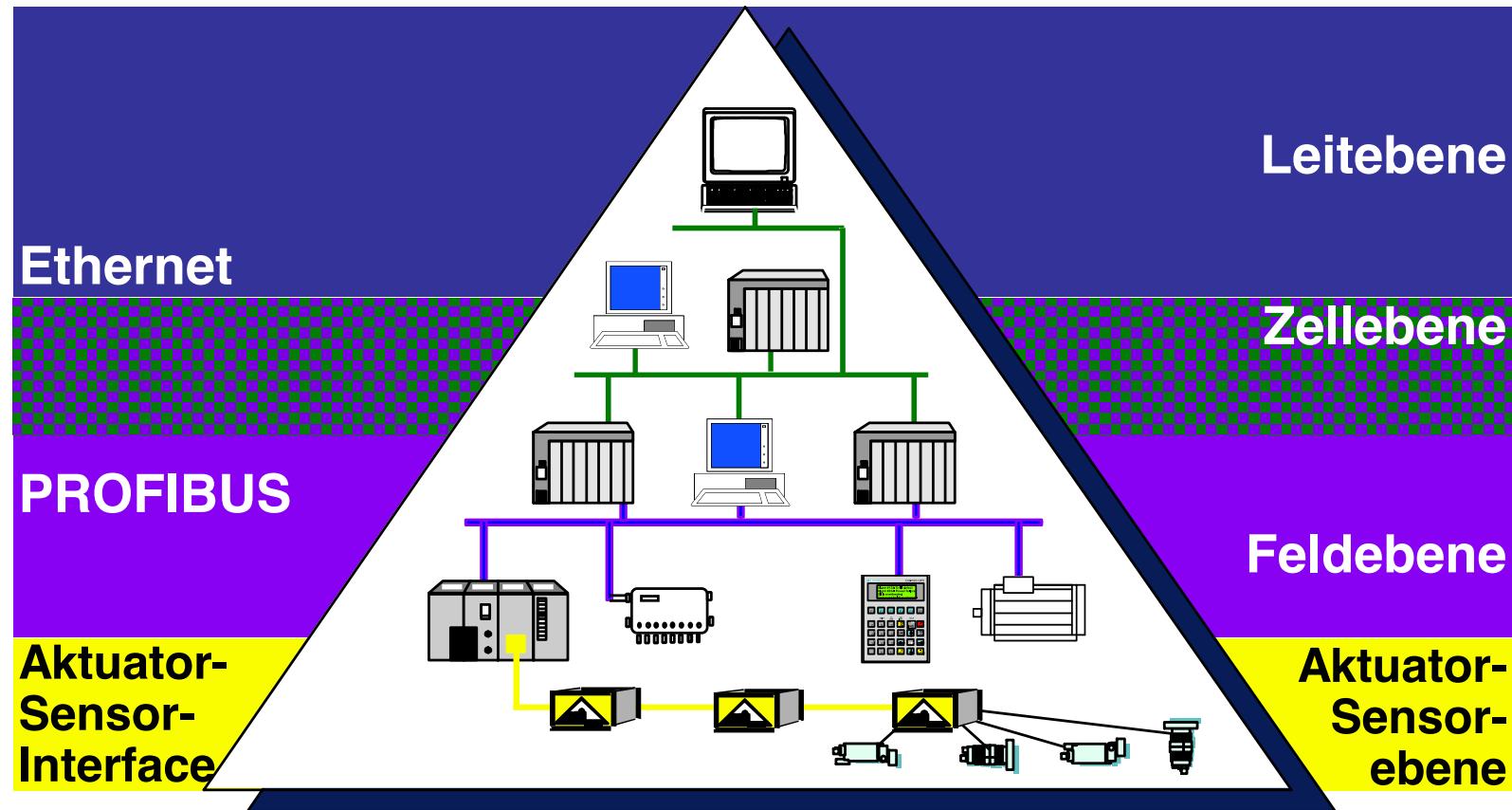
Netzwerk und Netzwerkkomponenten Vernetzte WZM und Leitrechner



Anwendungsbereiche, Prozesse und Methoden

- Automatisierungstechnik in der Produktion -

Automatisieren mit Steuerungstechnik von der Leitebene bis zum industriellen Prozess / Kommunikation in der Produktion



Anwendungsbereiche, Prozesse und Methoden

- Steuerungstechnik in der Produktion - Entwicklung -

Generell werden Steuerungen in der Automatisierungstechnik zum Messen, Steuern, Regeln und Überwachen von Prozessen eingesetzt.

Meilensteine der NC Entwicklung (Numerical Control)

Ideen zur Steuerung einer Konstruktion durch fortlaufende Befehle gehen bis ins 14. Jahrhundert zurück

1800 Webmaschinen - Blechbänder (später Papierstreifen, Lochkarten)

1938 Arbeiten zur binären Verarbeitung von Informationen und Bool'sche Algebra (Claude E. Shannon, M.I.T.)

1946 ENIAC erster elektronischer Digitalrechner (US Militär)

1949 M.I.T NC Entwicklung (US Air Force, Flugzeugentwicklung)

1952 3D-NC Linearinterpolation

1957 NC-Fräsen (und symbolische Programmierung)

1958 Programmiersprache für die Steuerung von NC-Werkzeugmaschinen APT (IBM) auf Mainframe Computer (in Fortran IV, später EXAPT)

1968 IC Technik

1972 NC auf Mini-Computer (IBM), Fortran, Pascal

1980 CNC Technik (Micro-Computer, Mikro-Chips, PC) (CAx-Techniken)

Ergebnis: durchgängige Konstruktion und Bearbeitung von Freiformflächen

2010+ Industrie 4.0 (intelligente verteilte Systeme, Selbststeuerung und Internet der Dinge)



Anwendungsbereiche, Prozesse und Methoden

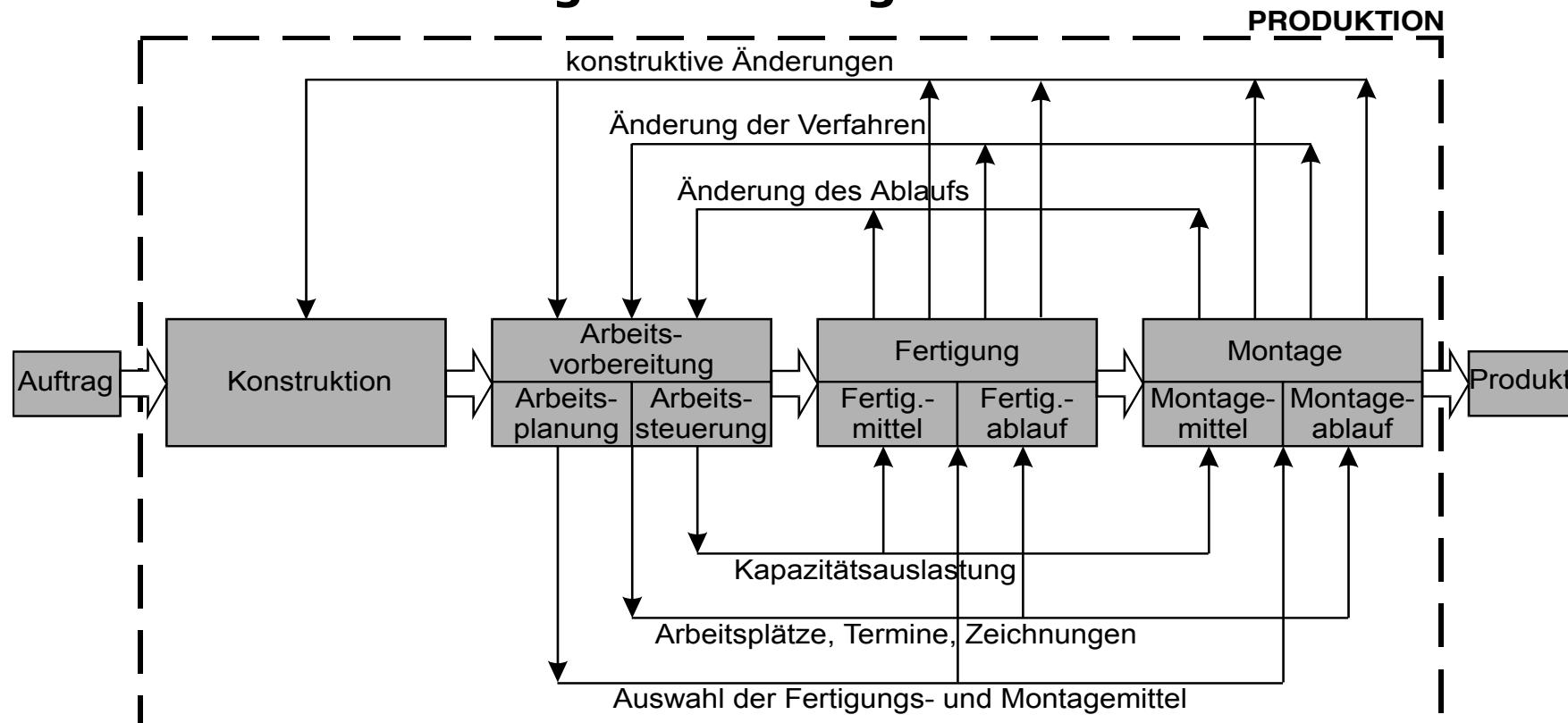
- Entwicklungsaufgaben für die Verbesserung der Steuerungstechnik

- Ablösung der manuellen Steuerung (Drehen, Fräsen, usw.) durch elektrische Schaltungen und Programme (z.B. numerische Steuerung, NC-Technik)
- Ablösung der festen Verdrahtung (freie Programmierung von Verbindungen)
- Echtzeitsteuerung und -programmierung (Software-Werkzeuge)
- Entwicklung von aufgabenorientierten Steuerungen ([Speicherprogrammierbare Steuerungen SPS](#))
- Prozessleittechnik (PLT, PLS)
- Einsatz von modernen Computern und Mikroprozessoren (z.B. CNC Technik)
- freie Programmierung (Assembler, Mnemonik, freie Programmiersprachen)
- komplexe Programme (5- Achsbearbeitung, Freiformflächen)
- Systemkompatibilität (Datenaustausch, Kommunikation und Daten-Formate)
- Informationsfluss, Durchgängigkeit der Informationen

Heute z.B. „Ressourceneffizienz“, „Smart Production“, „Smart Control“ oder „Selbstdiagnose“, „Qualitätsregelkreise“, „Funktionsorientierte Steuerung“, „Materialorientierte oder Prozessorientierte Steuerung“

Anwendungsbereiche, Prozesse und Methoden

- Auftragsabwicklung in der Produktion -



Produktionstechnik dient der Erzeugung von Gütern

- Hauptgruppen *Fertigungstechnik, Verfahrenstechnik und Energietechnik*
- Querschnittstechniken: *Fördertechnik und Informationstechnik*

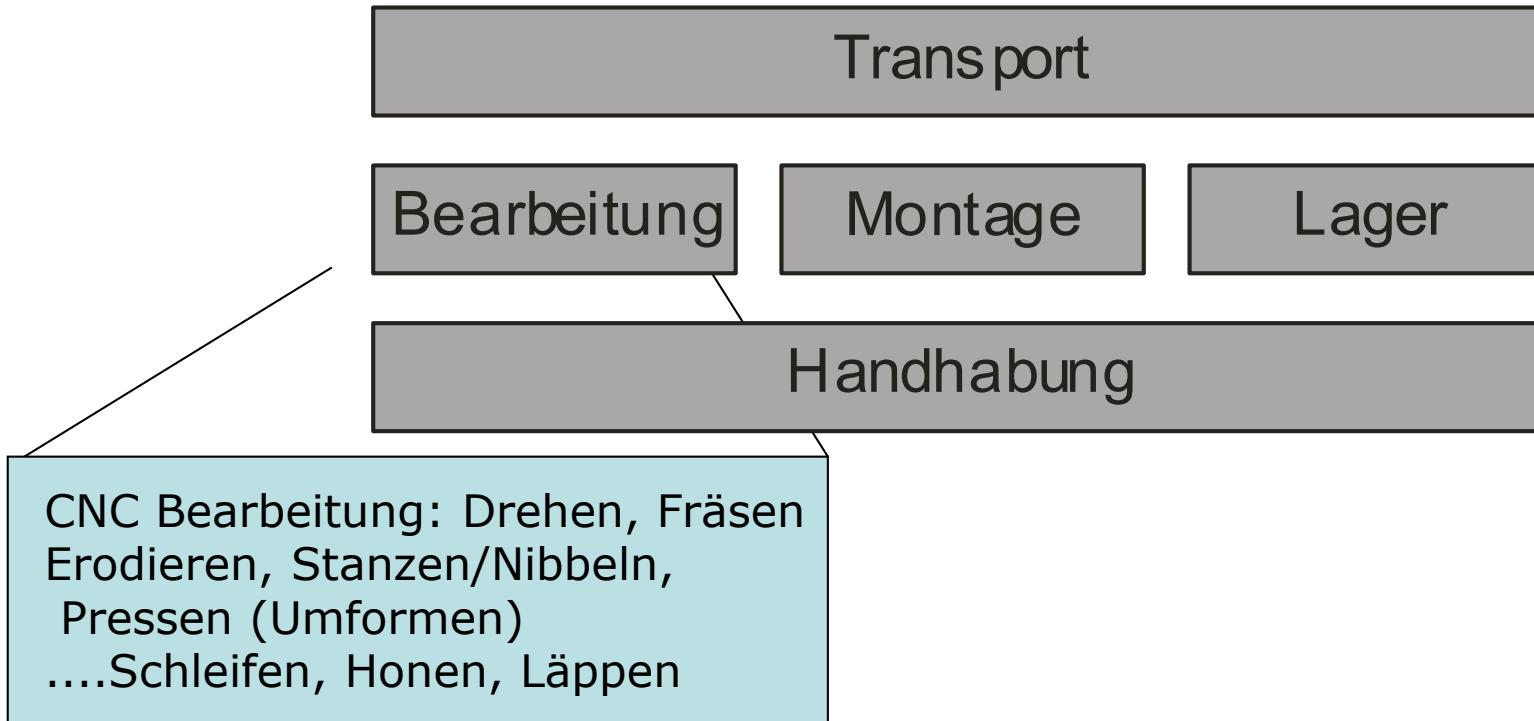
Fertigungstechnik befasst sich mit der Produktion von Stückgütern durch Formherstellung und -veränderung

Bild: Auftragsabwicklung und Informationsfluss in der klassischen Auftragsabwicklung

Literatur: Eversheim, Organisation in der Produktionstechnik, VDI

Anwendungsbereiche, Prozesse und Methoden

- Bereiche der werkstückbehafteten Produktion - Bearbeitung -



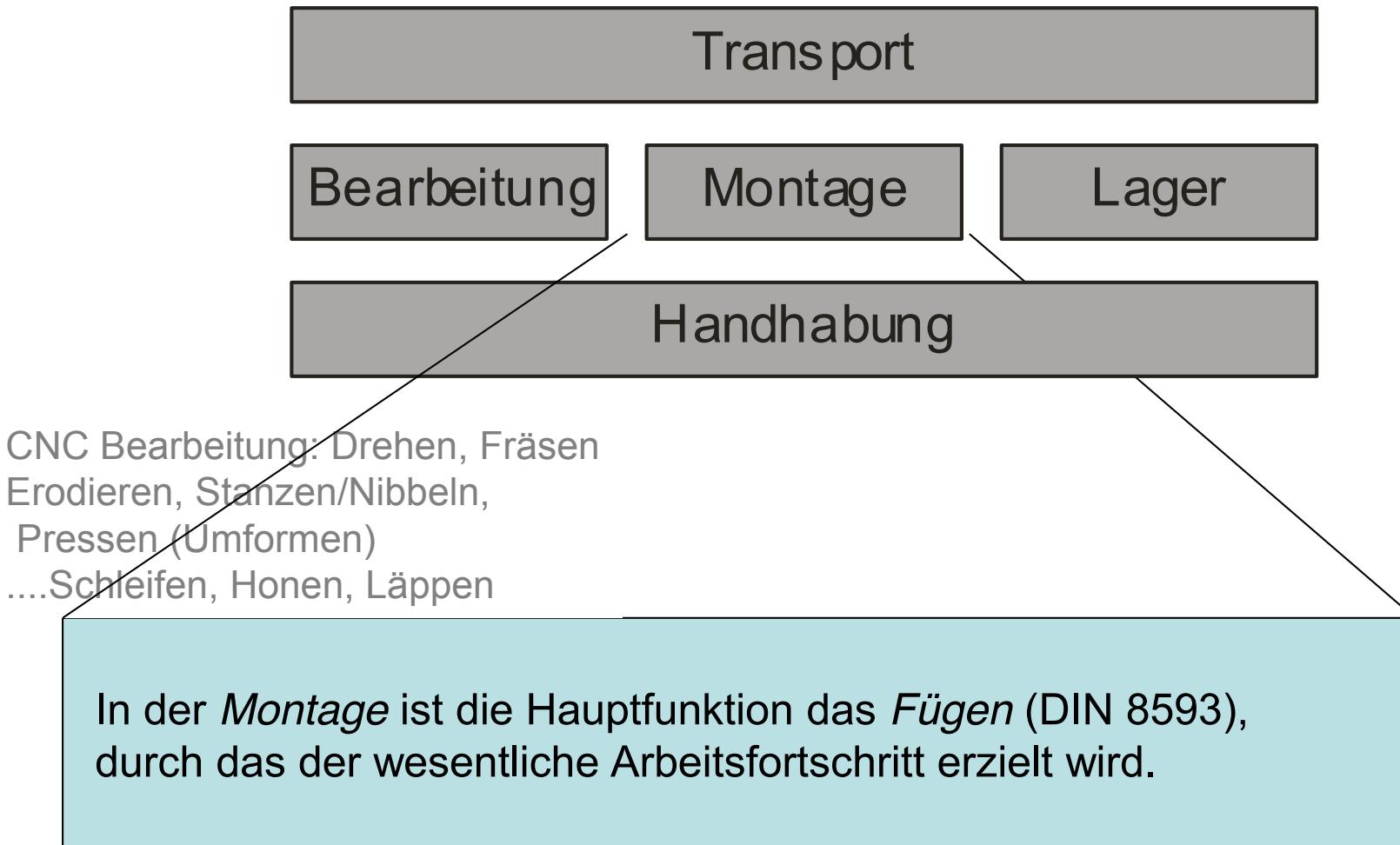
Fertigung wird nach DIN 8580 die Veränderung von Gestalt und Eigenschaft der eingesetzten Rohmaterialien

Literatur: Warnecke, Westerkämper, *Einführung in die Produktionstechnik*, Vieweg
DIN, Deutsches Institut für Normung, Maschinenbau

1.2 Bereiche der
werkstückbehafteten
Produktion

Anwendungsbereiche, Prozesse und Methoden

- Bereiche der werkstückbehafteten Produktion - Montage -



Anwendungsbereiche, Prozesse und Methoden

- Basis- und Integrationstechniken-

Programme

Steuerungen

Automatisierungselemente:
Sensoren / Aktoren

Basistechniken der Automatisierung:

Sensor- und Aktorteknik
Regelungstechnik
Steuerungstechnik
Leittechnik
Robotertechnik

Integrationstechniken der Automatisierung:

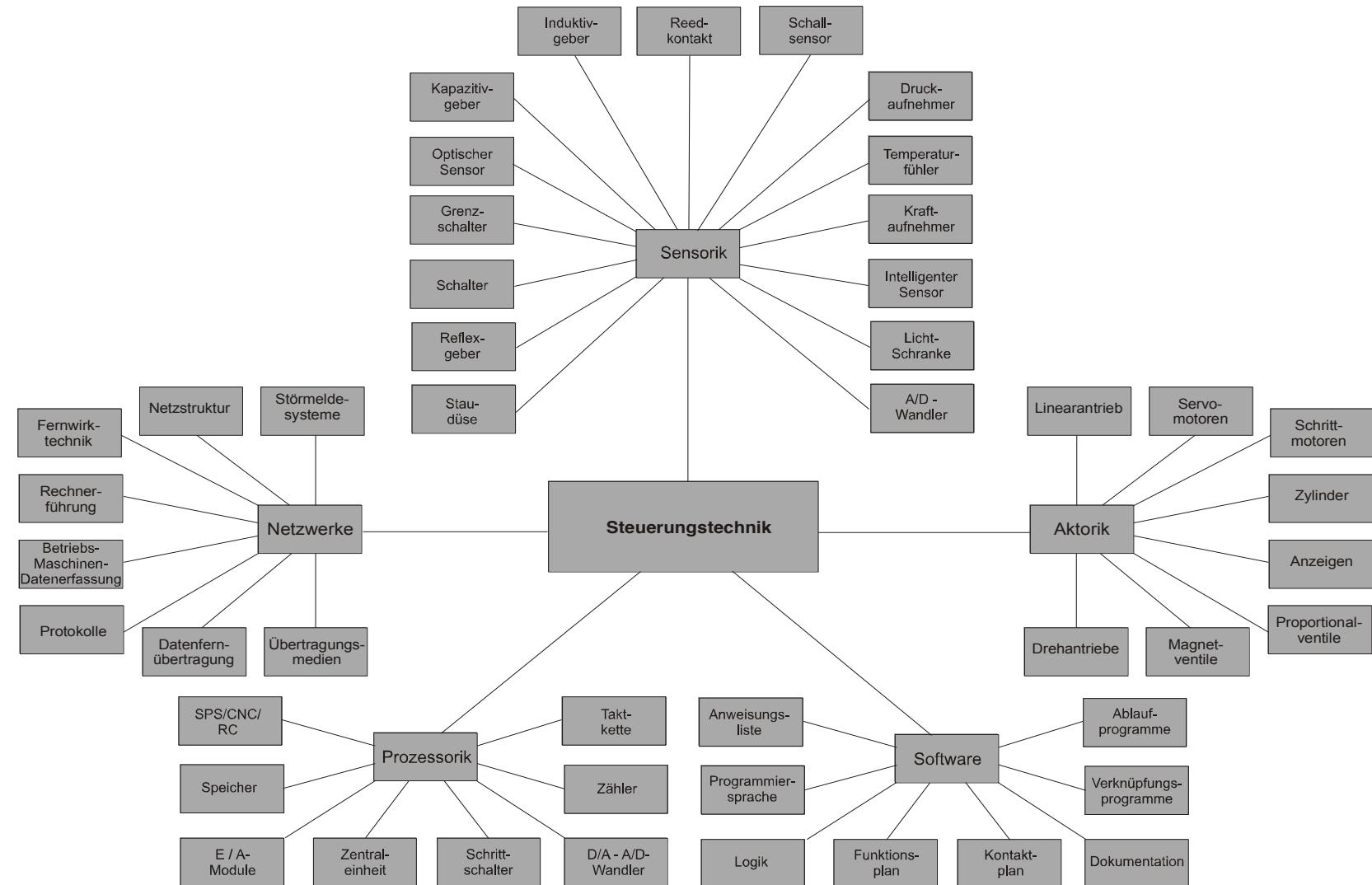
Rechnertechnik
Informationstechnik
Kommunikationstechnik
Mensch-Maschine-Systeme
Systemtechnik
Managementtechniken

Automatisierung von Folgeprozessen (**Steuerungstechnik, Robotertechnik**)

Automatisierung von Fließprozessen (Regelungstechnik, Leittechnik)

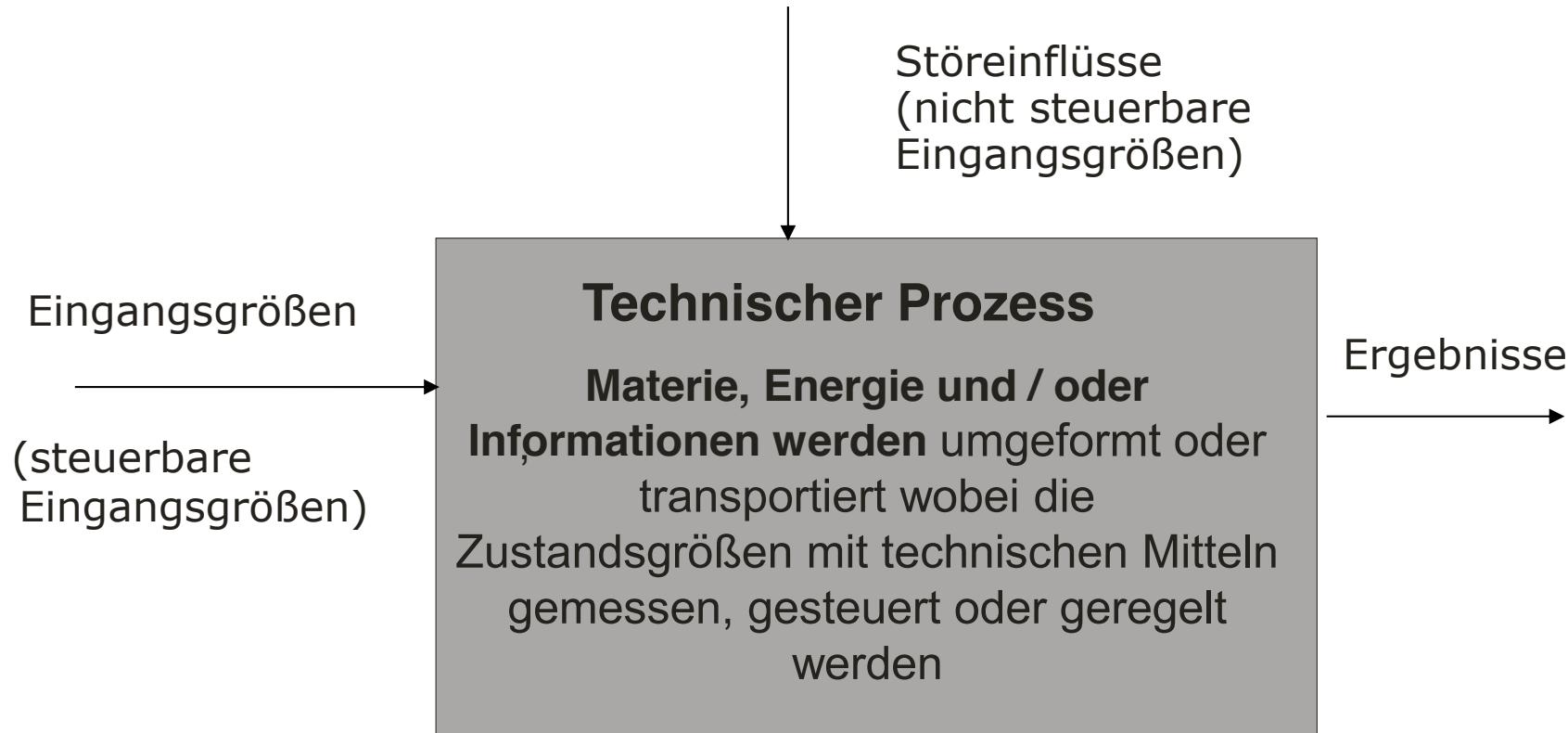
Anwendungsbereiche, Prozesse und Methoden

- Steuerungstechnik in der Produktion - Überblick -



Anwendungsbereiche, Prozesse und Methoden

- Technischer Prozess als Regelstrecke -



Nach DIN 66201 ist ein Prozess „die Gesamtheit von aufeinander einwirkenden Vorgängen in einem System, durch die Materie, Energie oder Informationen umgeformt, transportiert oder gespeichert wird“

Lehrziele und Gliederung

V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik

V2 Automatisierung in der Produktion

V3 Boolesche Algebra 1

Ü1 Matlab Einführung

V4 Boolsche Algebra 2: Graphen

Ü2 Übung Boolsche Algebra

V5 Fuzzy Logic

Ü3 Fuzzy Logic

V6 Neuronale Netze

Ü4 Neuronale Netze

V7 Automatisiertes Messen und Steuern

Ü5 Automatisiertes Messen und Steuern

V8 Speicherprogrammierbare Steuerungen

Ü6 Übungen und Musterklausuren



Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 02 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



Bremer Institut für
Messtechnik, Automatisierung
und Qualitätswissenschaft

Lehrziele und Gliederung

1. Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- 2. Automatisierung in der Produktion**
3. Boolesche Algebra
4. Zustandsvektoren, Graphen, Schaltwerksynthese, Netze
5. Fuzzy Logic
6. Neuronale Netze
7. Automatisieren von Fertigungsbereichen
8. Automatisiertes Messen und Steuern
9. Speicherprogrammierbare Steuerungen

Automatisierung in der Produktion

- Steuerungstechnik in der Produktion - Entwicklung -

Generell werden Steuerungen in der Automatisierungstechnik zum Messen, Steuern, Regeln und Überwachen von Prozessen eingesetzt.

Meilensteine der NC Entwicklung (Numerical Control)

Ideen zur **Steuerung einer Konstruktion** durch fortlaufende Befehle gehen bis ins 14. Jahrhundert zurück

- | | |
|-------|--|
| 1800 | Webmaschinen - Blechbänder (später Papierstreifen, Lochkarten) |
| 1938 | Arbeiten zur binären Verarbeitung von Informationen und Bool'sche Algebra (Claude E. Shannon, M.I.T.) |
| 1946 | ENIAC erster elektronischer Digitalrechner (US Militär) |
| 1949 | M.I.T NC Entwicklung (US Air Force, Flugzeugentwicklung) |
| 1952 | 3D-NC Linearinterpolation |
| 1957 | NC-Fräsen (und symbolische Programmierung) |
| 1958 | Programmiersprache für die Steuerung von NC-Werkzeugmaschinen APT (IBM) auf Mainframe Computer (in Fortran IV, später EXAPT) |
| 1968 | IC Technik |
| 1972 | NC auf Mini-Computer (IBM), Fortran, Pascal |
| 1980 | CNC Technik (Micro-Computer, Mikro-Chips, PC) (CAx-Techniken)
Ergebnis: durchgängige Konstruktion und Bearbeitung von Freiformflächen |
| 2010+ | Industrie 4.0 (intelligente verteilte Systeme, Selbststeuerung und Internet der Dinge) |

Automatisierung in der Produktion

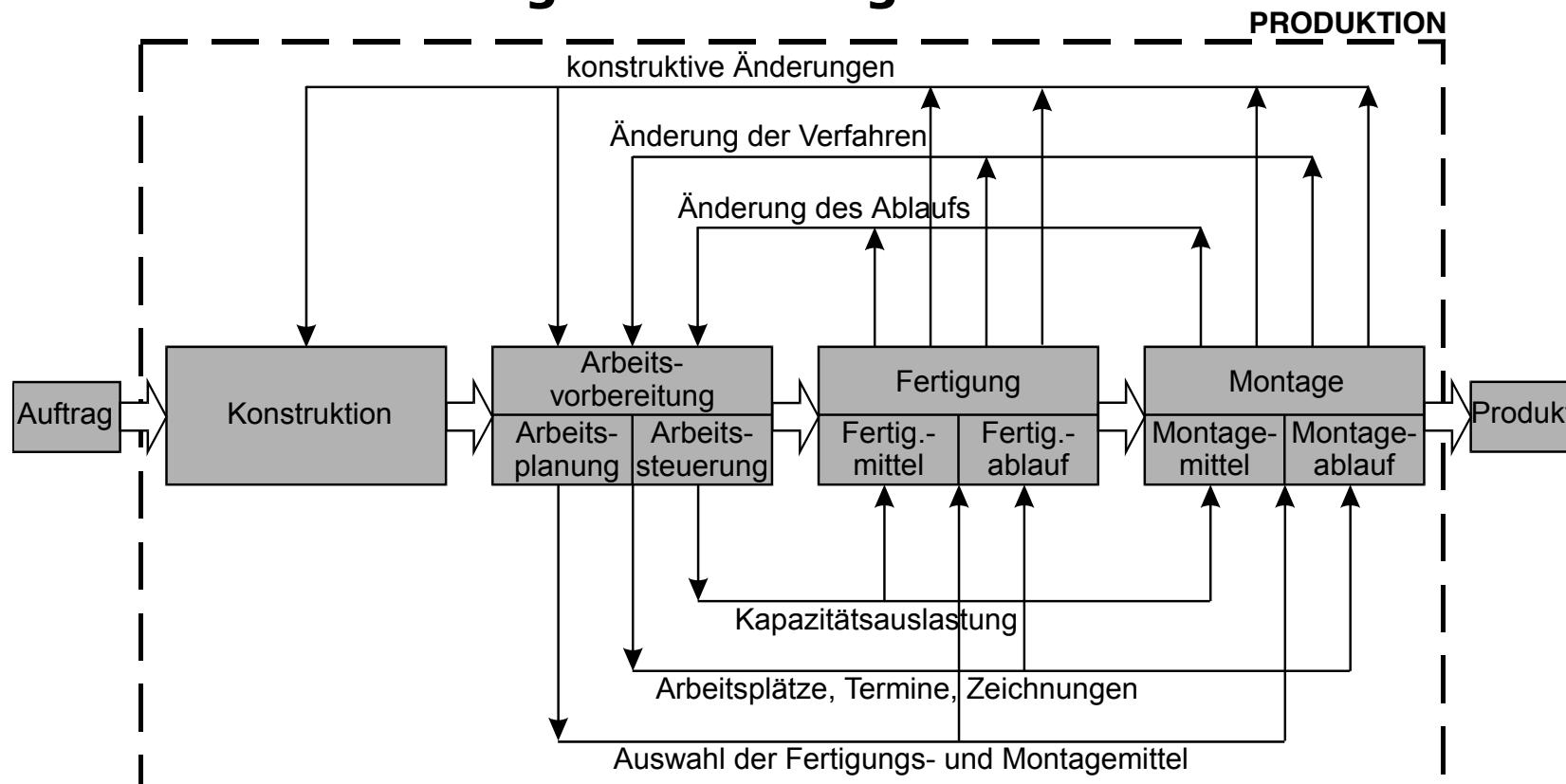
- Entwicklungsaufgaben für die Verbesserung der Steuerungstechnik

- Ablösung der manuellen Steuerung (Drehen, Fräsen, usw.) durch elektrische Schaltungen und Programme (z.B. numerische Steuerung, NC-Technik)
- Ablösung der festen Verdrahtung (freie Programmierung von Verbindungen)
- Echtzeitsteuerung und -programmierung (Software-Werkzeuge)
- Entwicklung von aufgabenorientierten Steuerungen (**Speicherprogrammierbare Steuerungen SPS**)
- Prozessleittechnik (PLT, PLS)
- Einsatz von modernen Computern und Mikroprozessoren (z.B. CNC Technik)
- freie Programmierung (Assembler, Mnemonik, freie Programmiersprachen)
- komplexe Programme (5- Achsbearbeitung, Freiformflächen)
- Systemkompatibilität (Datenaustausch, Kommunikation und Daten-Formate)
- Informationsfluss, Durchgängigkeit der Informationen

Heute z.B. „Ressourceneffizienz“, „Smart Production“, „Smart Control“ oder „Selbstdiagnose“, „Qualitätsregelkreise“, „Funktionsorientierte Steuerung“, „Materialorientierte oder Prozessorientierte Steuerung“

Automatisierung in der Produktion

- Auftragsabwicklung in der Produktion -



Produktionstechnik dient der Erzeugung von Gütern

- Hauptgruppen *Fertigungstechnik, Verfahrenstechnik und Energietechnik*
- Querschnittstechniken: *Fördertechnik und Informationstechnik*

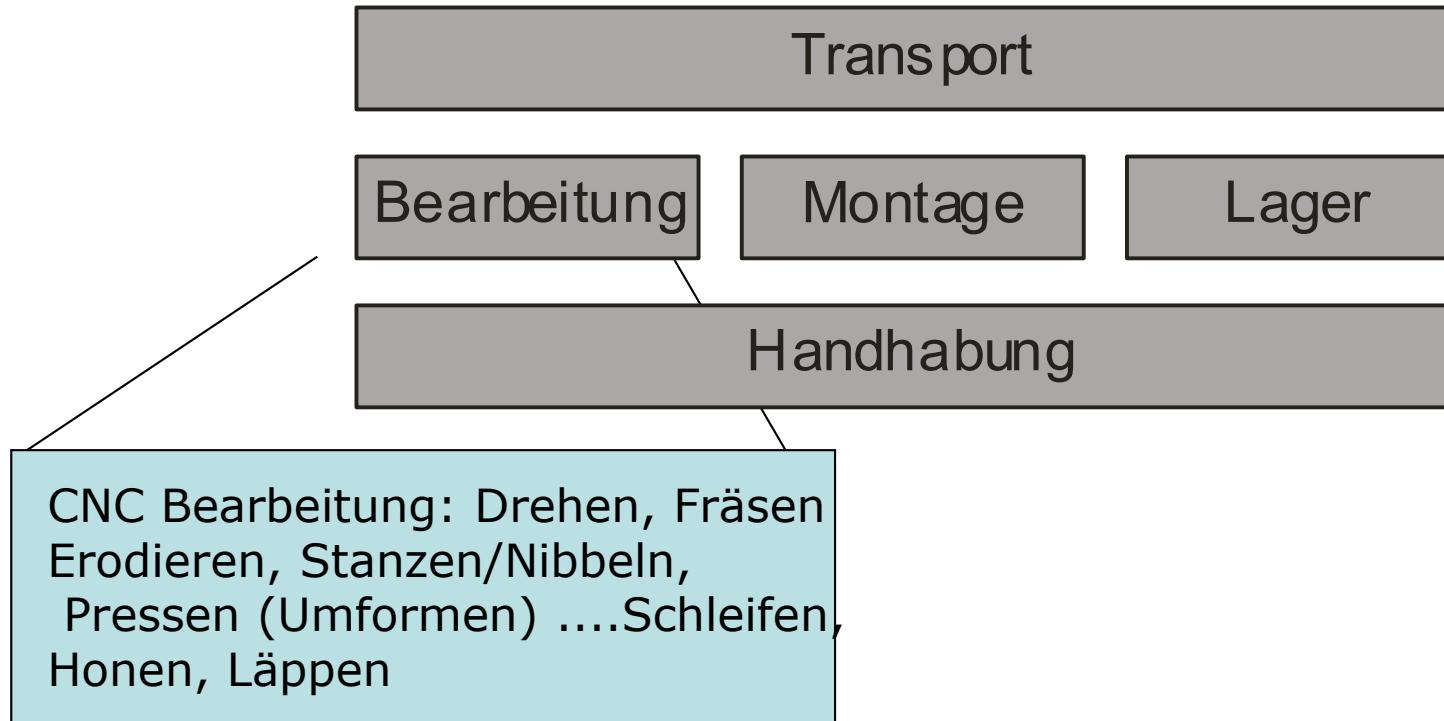
Fertigungstechnik befasst sich mit der Produktion von Stückgütern durch Formherstellung und -veränderung

Bild: Auftragsabwicklung und Informationsfluss in der klassischen Auftragsabwicklung

Literatur: Eversheim, Organisation in der Produktionstechnik, VDI

Automatisierung in der Produktion

- Bereiche der werkstückbehafteten Produktion - Bearbeitung -

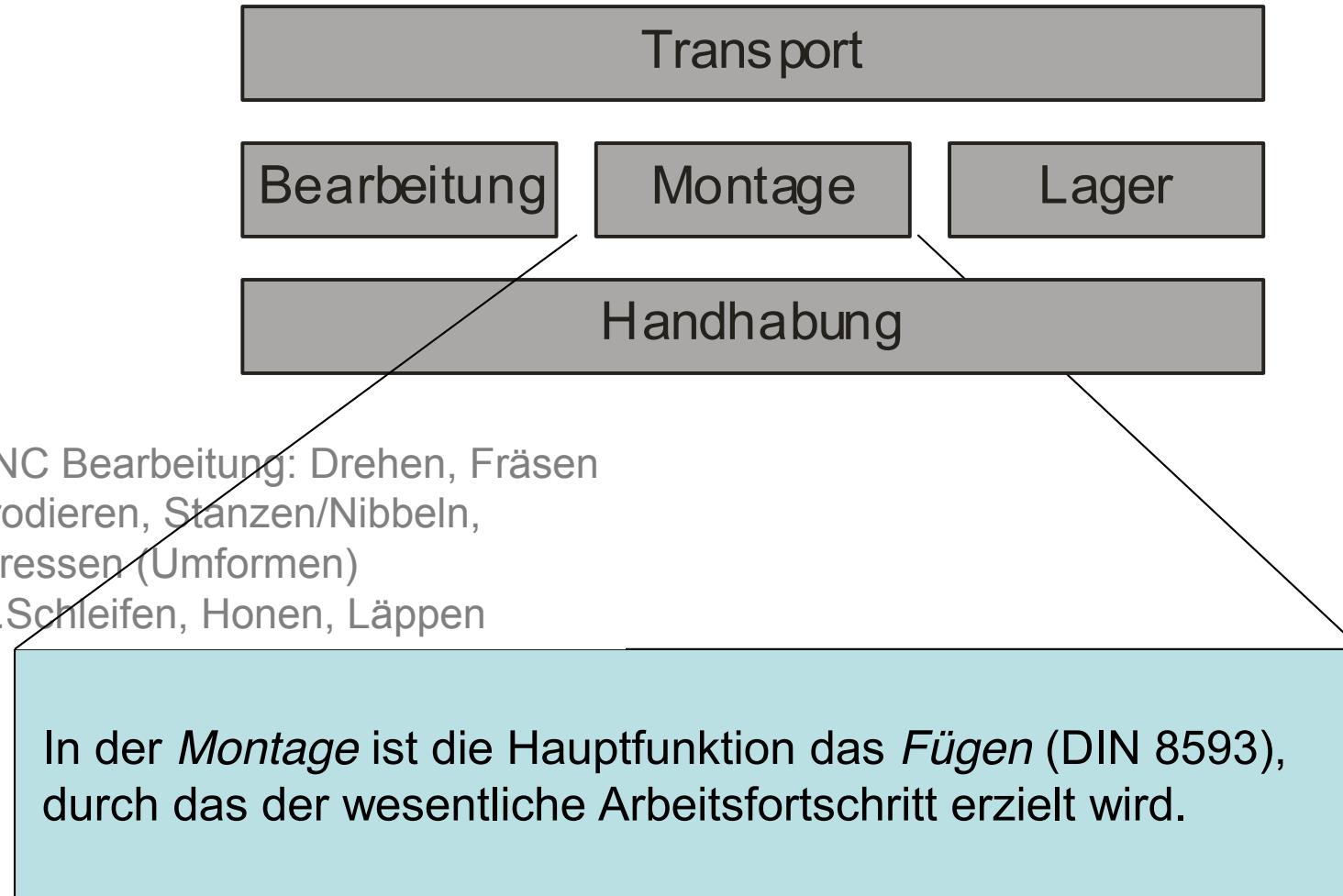


Fertigung wird nach DIN 8580 die Veränderung von Gestalt und Eigenschaft der eingesetzten Rohmaterialien

Literatur: Warnecke, Westerkämper, *Einführung in die Produktionstechnik*, Vieweg
DIN, Deutsches Institut für Normung, Maschinenbau

Automatisierung in der Produktion

- Bereiche der werkstückbehafteten Produktion - Montage -



Automatisierung in der Produktion

- Ziele -

Die Ziele in der Fertigungstechnik sind neben der Herstellung eines „qualitativ hochwertigen Produktes, dass seine Anforderungen optimal erfüllt:

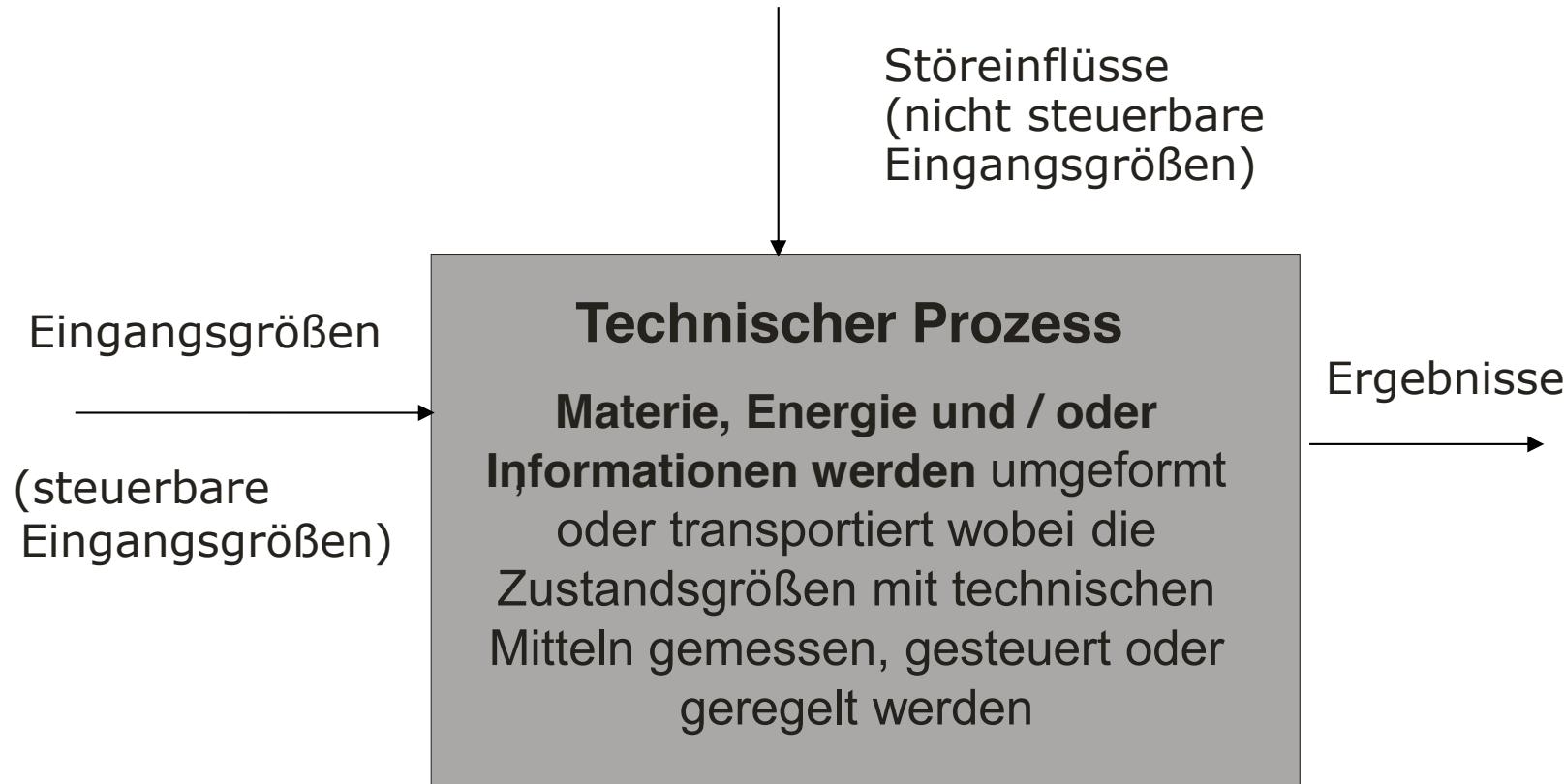
- steigende Mengenleistung
- wachsende Fertigungsgenauigkeit
- Umweltverträglichkeit (bis zur Entsorgung)
- Anpassung der Arbeitsbedingungen an den Menschen
- Senkung der Kosten

Ziele der Automatisierung

- Leistungssteigerung durch Technologie und organisatorischen Verbesserungen (Produktivität steigern, Zeit, Kosten und Energie, Umweltbelastungen sparen),
- Arbeitsbereiche verändern, die „Menschen belasten“ oder durch „Schwankungen der menschlichen Leistungsfähigkeit“ negativ beeinflusst werden
- *Automatisierung in Fertigung, Montage und Transport* bedeutet, dass einzelne Vorgänge oder komplette Produktionsabläufe selbsttätig in programmierte Form durchgeführt werden

Automatisierung in der Produktion

- Technischer Prozess als Regelstrecke -



Nach DIN 66201 ist ein Prozess „die Gesamtheit von aufeinander einwirkenden Vorgängen in einem System, durch die Materie, Energie oder Informationen umgeformt, transportiert oder gespeichert wird“

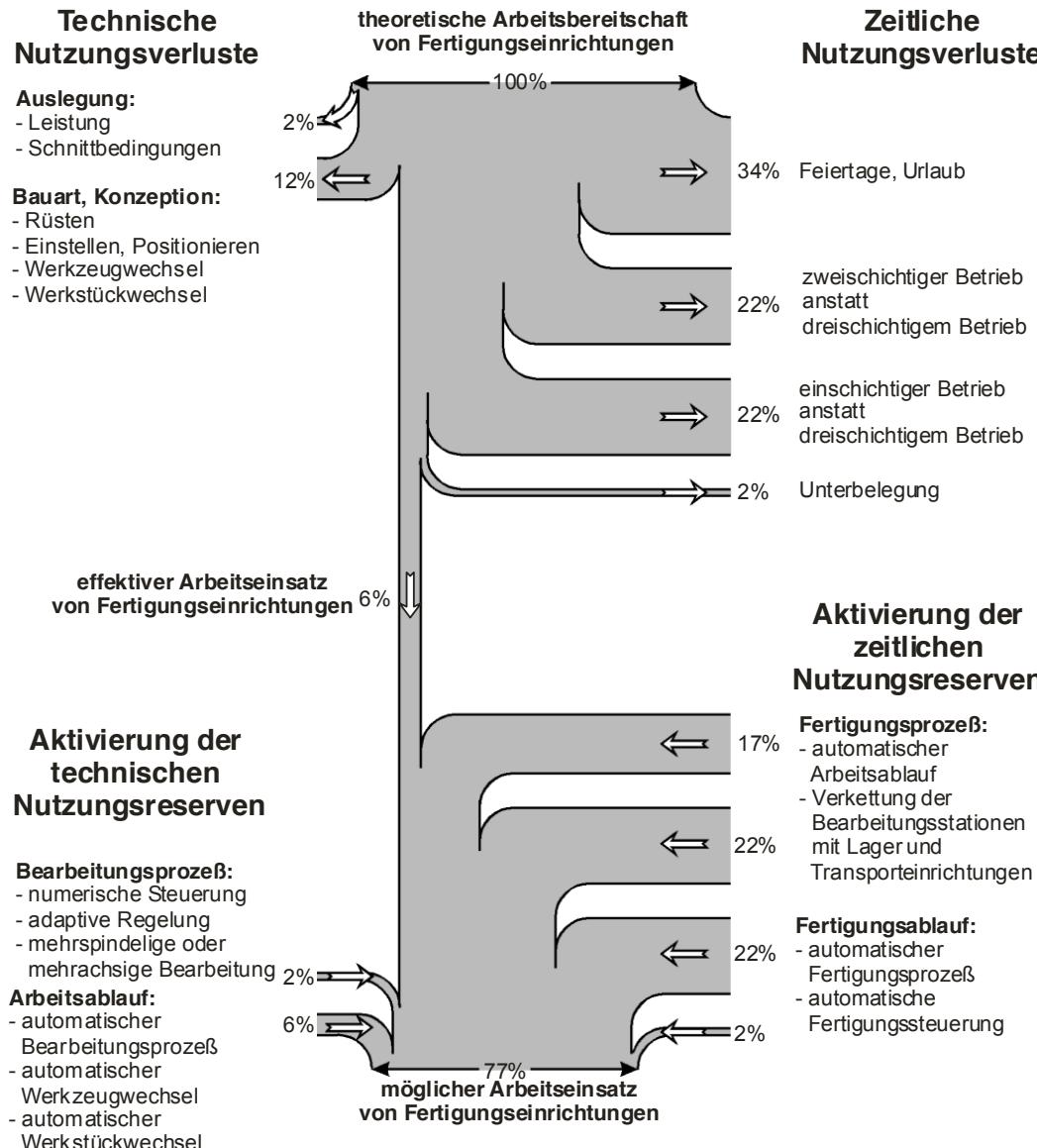
Automatisierung in der Produktion

- Fertigungsmittel und Äquivalent in Montage, Handhabung und Transport -

Einteilung	Beispiel	Äquivalent		
Fertigungsmittel	Fertigung	Montage	Transport	Handhabung
Anlage	Transfer-Straße	Montagelinie	Hängeförderer	Handhabungssystem
Maschine	Bohr-, Fräs-, Dreh-, Erodiermaschine	Industrieroboter Montage-automat	Flurfördersystem	Positioniergerät
Werkzeuge	Spanabhebende Werkzeuge	Greifer	Transporthilfe	Greifer, Schrauber
Vorrichtung	Spannvorrichtung	Fixierstation	Palette	Fixierstation

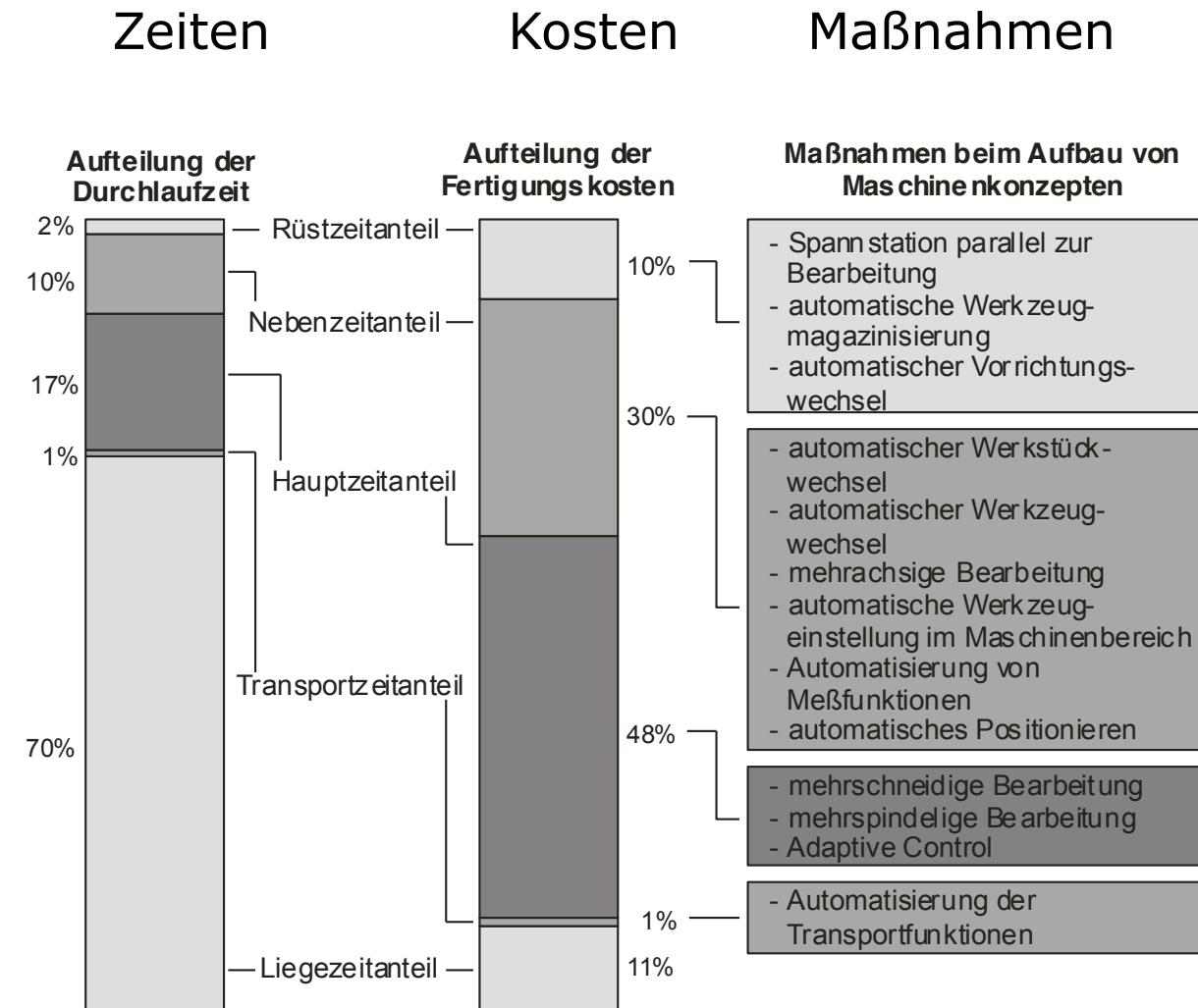
Automatisierung in der Produktion

- Aktivierung der technischen und zeitlichen Nutzungsreserven -



Automatisierung in der Produktion

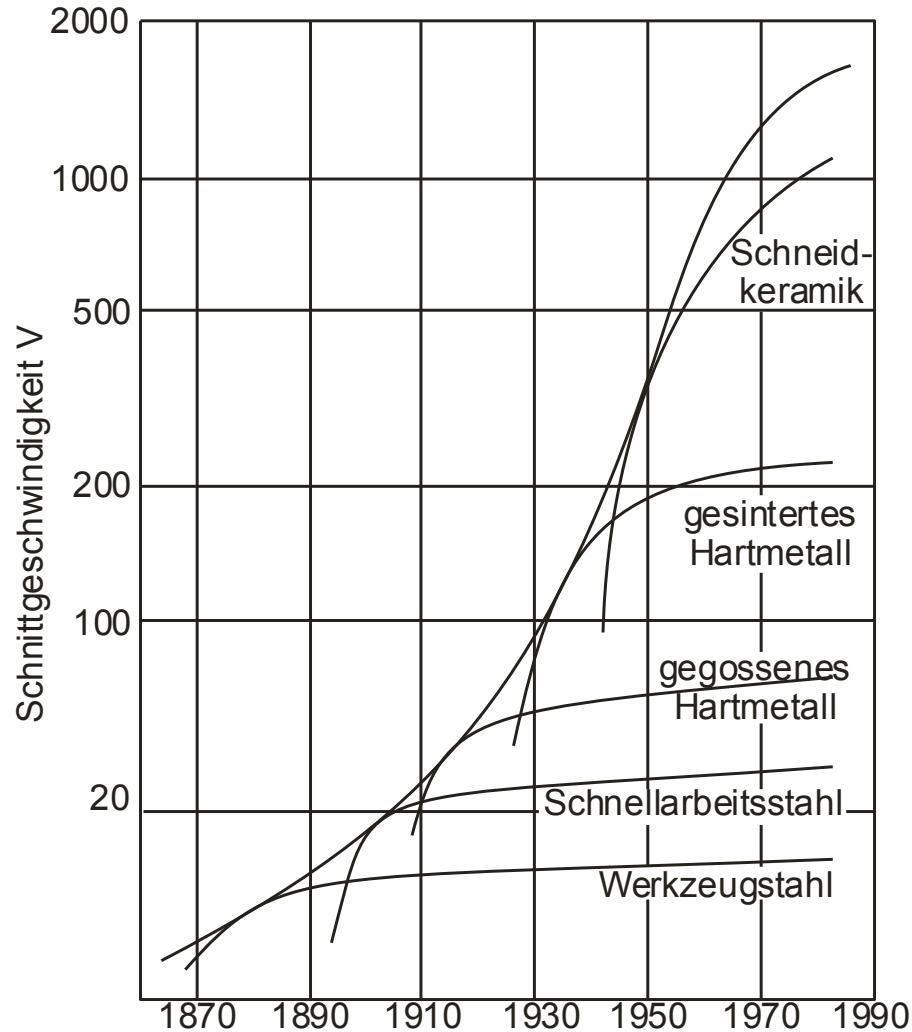
- Zeiten, Kosten Maßnahmen -



[3] Wiedergegeben mit freundlicher Genehmigung des VDI-Verlages, Düsseldorf, aus: Eversheim, Walter: Organisation in der Produktionstechnik. Studium und Praxis. Bd. 4. Düsseldorf 1990, S. 124.

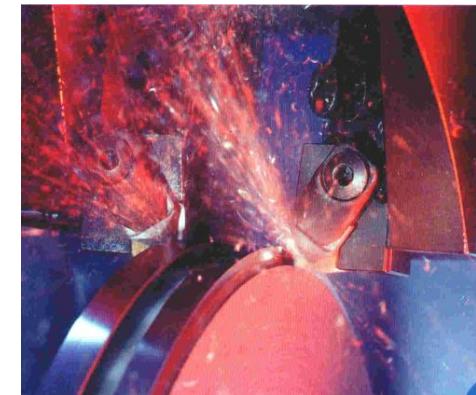
Automatisierung in der Produktion

- Entwicklung der Schnittgeschwindigkeit -



Entwicklung der Schnittgeschwindigkeit (v in m/min) unter dem Einfluss der Schneidewerkzeuge
(z.B. Siliziumnitridkeramik heute v_c 500-1800 m/min)

die Schnittgeschwindigkeit hat sich in den letzten 100 Jahren etwa um den Faktor Hundert verbessert



Hochgeschwindigkeitsbearbeitung (HSC = High Speed Cut) mit Schneidkeramik-Werkstoffen beim Spanen von Gusswerkstücken
Quelle CeramTec

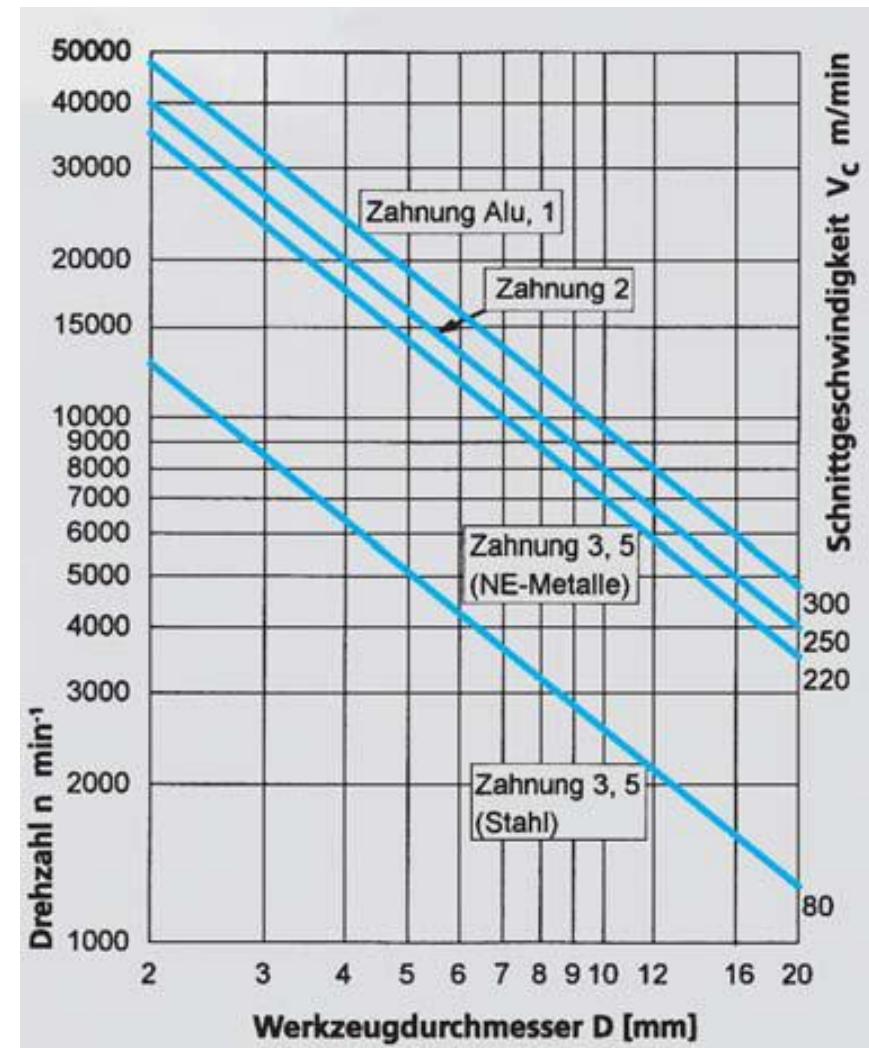
Automatisierung in der Produktion

- Schnittgeschwindigkeit -

- Im Diagramm sind die m/min durch diagonale Geraden dargestellt, und zwar die für HSS gültigen Werte.
- Die dem Werkzeugdurchmesser entsprechende Senkrechte trifft auf die angegebene Schnittgeschwindigkeit (Diagonale).
- Von dort wird in der Waagerechten am linken Rand die entsprechende Maschinendrehzahl in min⁻¹ bestimmt.

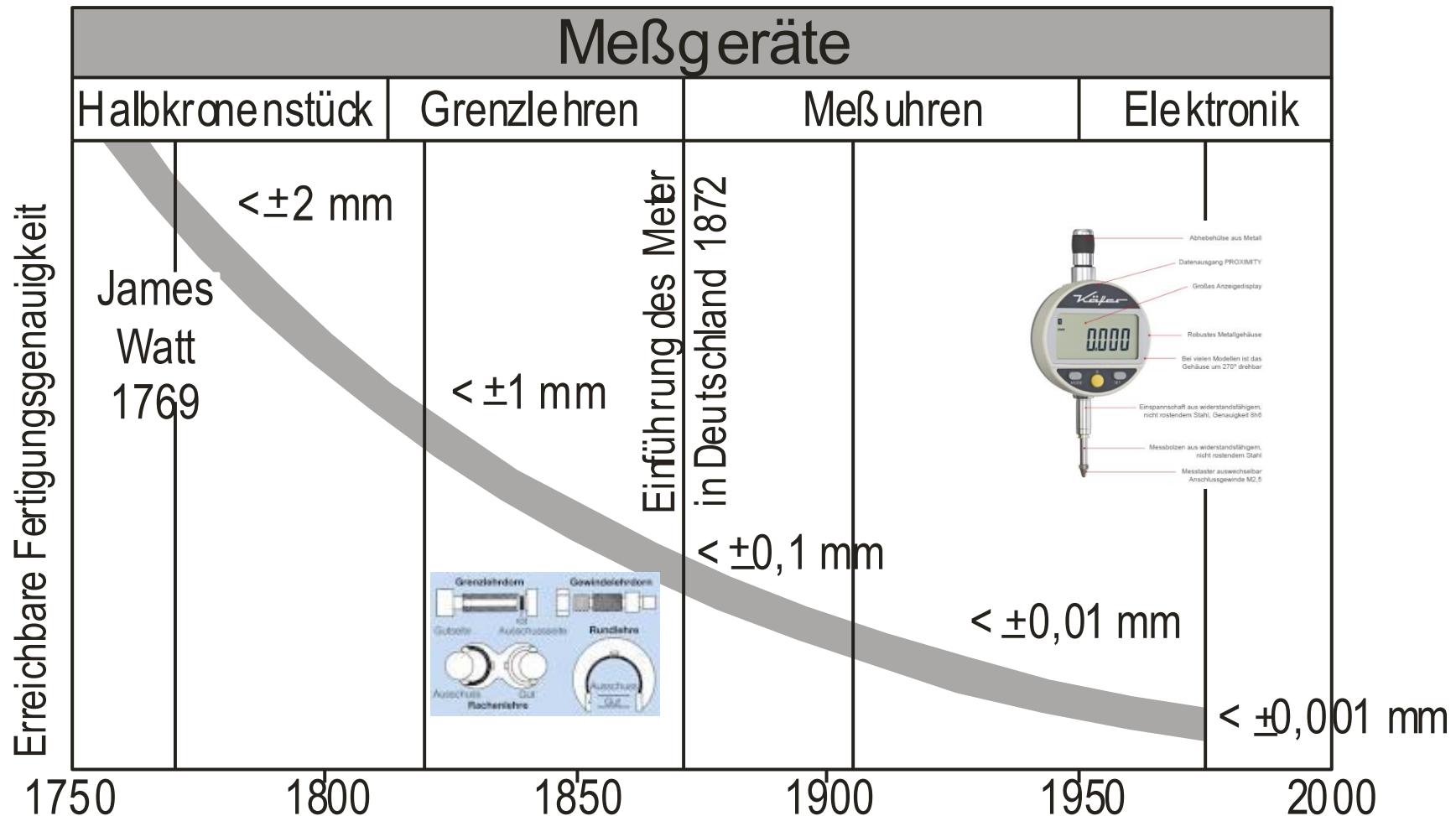
Beispiel:
HSS = Hochleistungsschnellstahl

HSS-Fräser, Zahnung 1
NE-Metallbearbeitung (Alu-Legierungen, Messing, Kupfer, Zink)
Werkzeugdurchmesser 12 mm.
Schnittgeschwindigkeit: 300 m/min.
Maschinendrehzahl: 8000 min⁻¹



Automatisierung in der Produktion

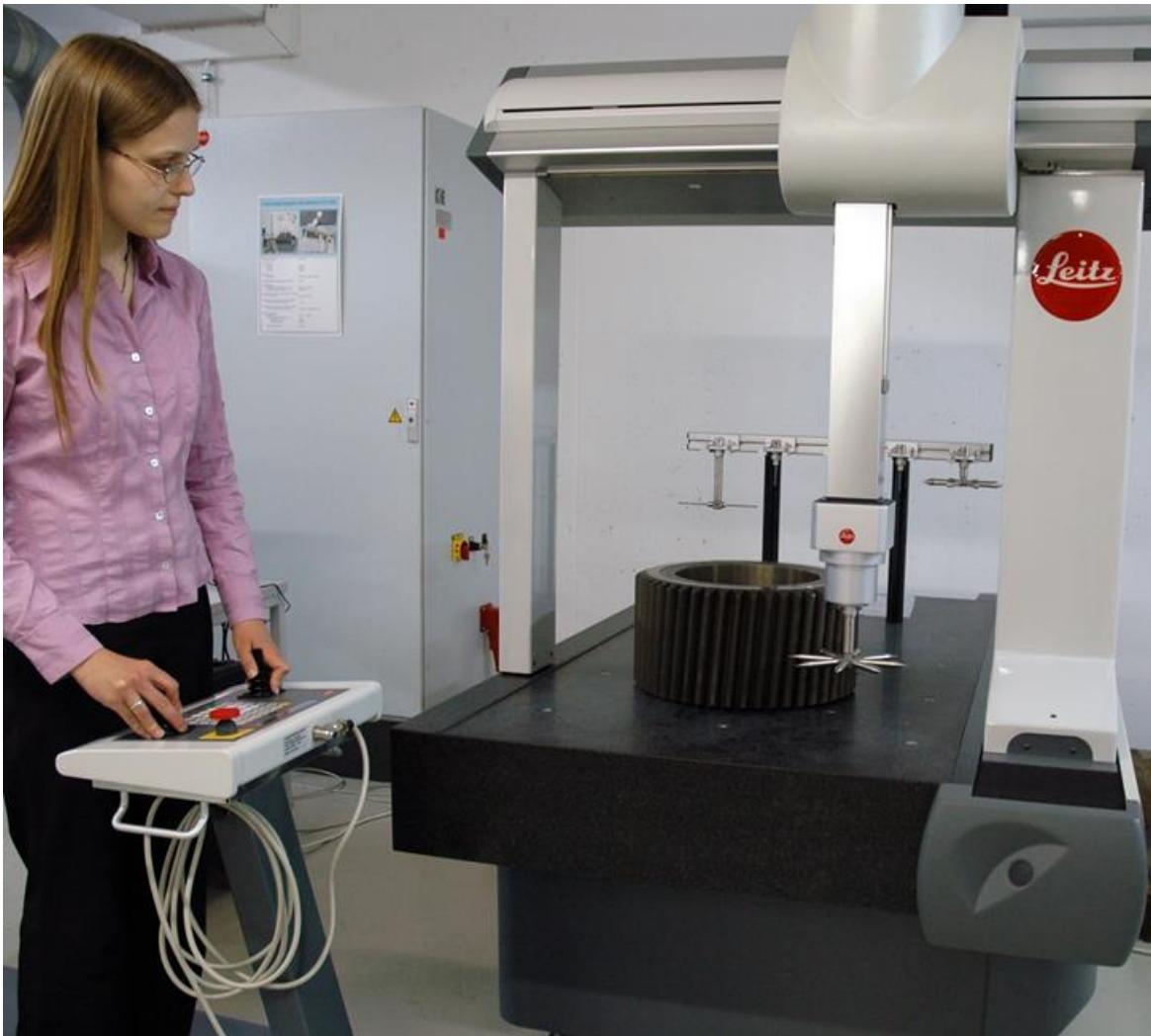
- Fertigungsgenauigkeit und eingesetzte Messtechnik -



- Koordinatenmesstechnik < 1 Micrometer + längenabhängiger Betrag
- Optische Messtechnik - nm Bereiche

Automatisierung in der Produktion

- Taktile Messtechnik -

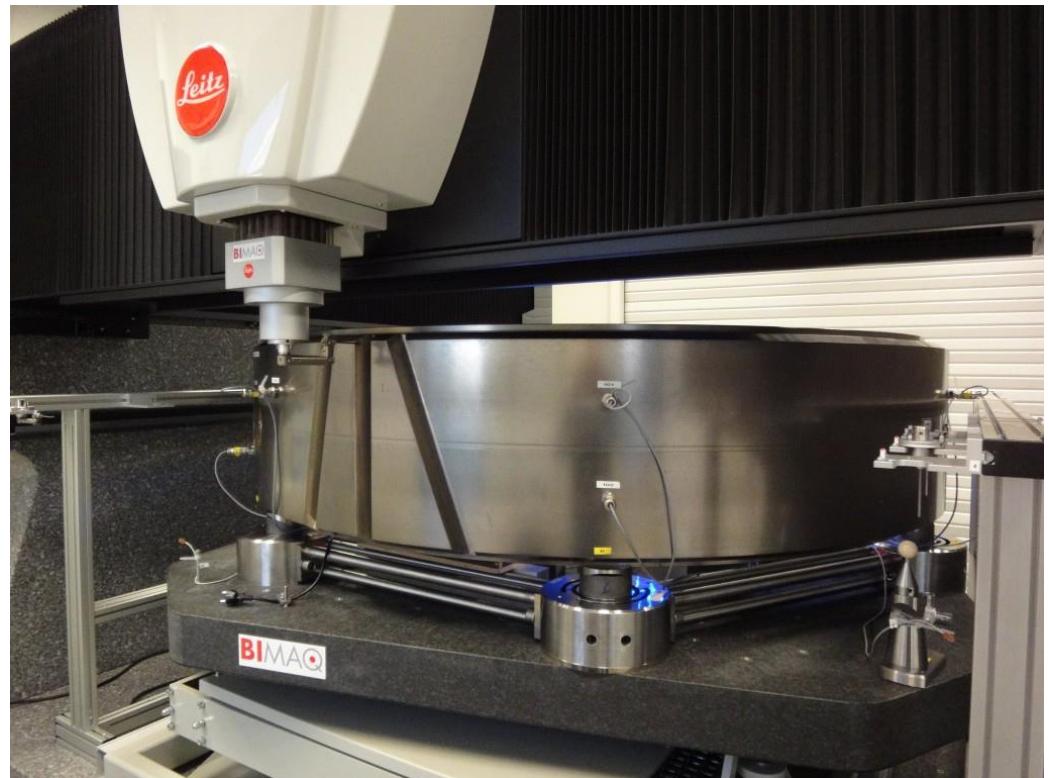


Automatisierung in der Produktion

- Koordinatenmesstechnik -

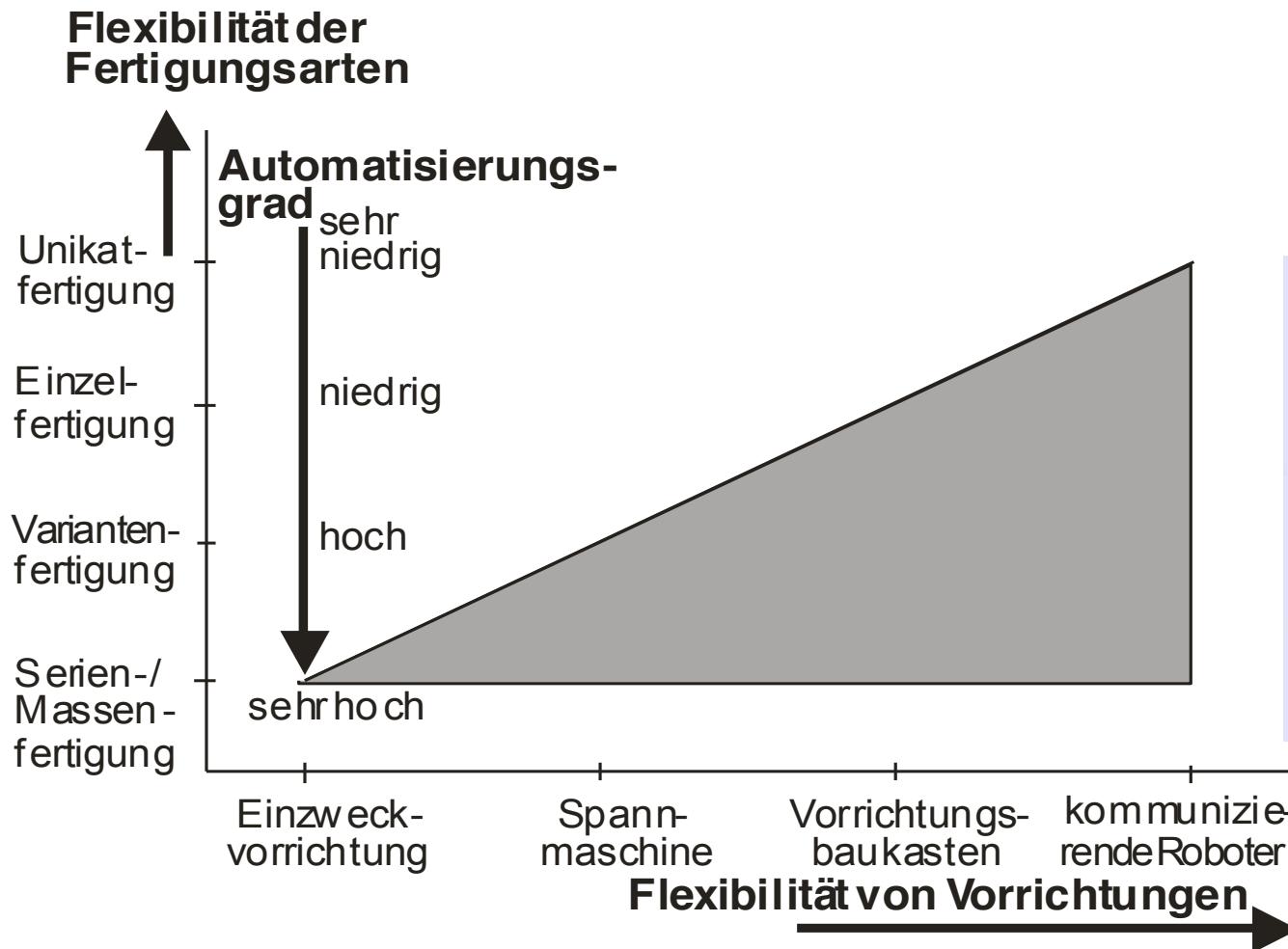


Koordinatenmesstechnik



Automatisierung in der Produktion

- Flexibilität von Vorrichtungen und Automatisierung -



Hydraulische Spannvorrichtung



Automatisierung in der Produktion

- Basis- und Integrationstechniken -

Programme

Steuerungen

Automatisierungselemente:
Sensoren / Aktoren

Basistechniken der Automatisierung:

Sensor- und Aktortechnik

Regelungstechnik

Steuerungstechnik

Leittechnik

Robotertechnik

Integrationstechniken der Automatisierung:

Rechnertechnik

Informationstechnik

Kommunikationstechnik

Mensch-Maschine-Systeme

Systemtechnik

Managementtechniken

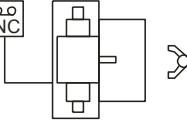
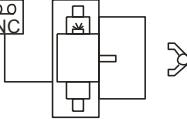
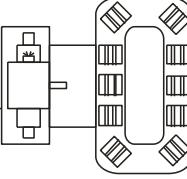
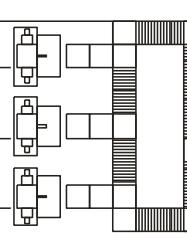
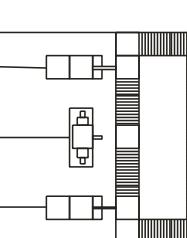
Automatisierung von Folgeprozessen ([Steuerungstechnik](#), [Robotertechnik](#))

Automatisierung von Fließprozessen ([Regelungstechnik](#), Leittechnik)



Automatisierung in der Produktion

- Automatisierungskonzepte -

Automatisierungs-konzepte		Beispiele	Kennzeichnung
Einzelmaschinensysteme	NC-Maschine		<ul style="list-style-type: none"> - Werkzeugmaschine mit einem Verfahren (z.B. Bohren oder Fräsen) - automatischer Fertigungsablauf durch NC-Steuerung
	NC-Bearbeitungs-zentrum		<ul style="list-style-type: none"> - Werkzeugmaschine mit mehreren Verfahren (z.B. Bohren und Fräsen) - automatischer Fertigungsablauf durch NC-Steuerung
	Bearbeitungs-zentrum mit Umlauf-speicher		<ul style="list-style-type: none"> - Werkzeugmaschine mit Werkstückspeicherung - Automatisierung des Fertigungsablaufes und des Werkstückumlaufes durch die NC-Steuerung
verkettete NC-Maschinensysteme	flexibles Fertigungs-system		<ul style="list-style-type: none"> - NC-gesteuerte Maschinen - Transportverkettung mit wahlfreier Verfahr-möglichkeit - Transportsteuerung - Koordination der Substeuerungen durch übergeordnetem Rechner
	flexible Fertigungs-strasse		<ul style="list-style-type: none"> - Verkettung mehrerer NC-Maschinen - feste Arbeits-vorgangsfolge - zentrale Rechner-steuerung nur in Einzelfällen notwendig

Automatisierung in der Produktion

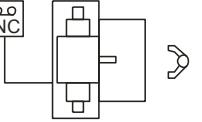
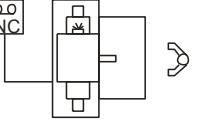
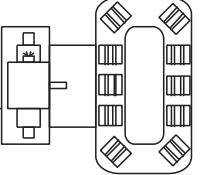
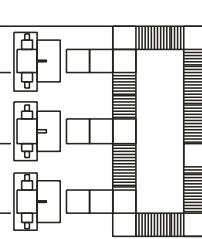
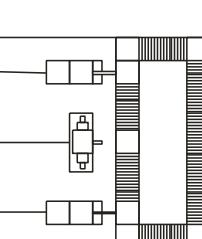
- Fertigungskonzepte und automatisierbare Funktionen -

•Fertigungskonzepte	•Automatisierbare Funktionen
<ul style="list-style-type: none">* CNC-Maschinen und Bearbeitungszentren* CNC-Messgeräte (-maschinen)* DNC – Systeme / Fertigungsleitsysteme* Flexible Fertigungssysteme FFS* FFS mit automatischem Werkzeugsystem* FFS mit integriertem Mess- und Überwachungssystem	<ul style="list-style-type: none">* Bearbeitung* Werkstückhandhabung* Werkzeughandhabung* Werkstücklager und Transport* Werkzeuglager und Transport* Messen von Werkstück- und Werkzeug-geometrie (In Prozess Messtechnik)* Informationsflusssteuerung* Materialflusssteuerung* Fertigungsüberwachung

Fertigungs-, Montage-, Lager- und Transportbereiche sind dadurch gekennzeichnet, dass sie direkt mit den Werkstücken und Werkzeugen arbeiten und repräsentieren damit den werkstückbehafteten Teil der Produktionstechnik.

Automatisierung in der Produktion

- Automatisierungskonzepte -

Automatisierungs-konzepte	Beispiele	Kennzeichnung
Einzelmaschinensysteme	NC-Maschine	 <ul style="list-style-type: none"> - Werkzeugmaschine mit einem Verfahren (z.B. Bohren oder Fräsen) - automatischer Fertigungsablauf durch NC-Steuerung
	NC-Bearbeitungs-zentrum	 <ul style="list-style-type: none"> - Werkzeugmaschine mit mehreren Verfahren (z.B. Bohren und Fräsen) - automatischer Fertigungsablauf durch NC-Steuerung
	Bearbeitungs-zentrum mit Umlauf-speicher	 <ul style="list-style-type: none"> - Werkzeugmaschine mit Werkstückspeicherung - Automatisierung des Fertigungsablaufes und des Werkstückumlaufes durch die NC-Steuerung
verkettete NC-Maschinensysteme	flexibles Fertigungs-system	 <ul style="list-style-type: none"> - NC-gesteuerte Maschinen - Transportverkettung mit wahlfreier Verfahr-möglichkeit - Transportsteuerung - Koordination der Substeuerungen durch übergeordnetem Rechner
	flexible Fertigungs-strasse	 <ul style="list-style-type: none"> - Verkettung mehrerer NC-Maschinen - feste Arbeits-vorgangsfolge - zentrale Rechner-steuerung nur in Einzelfällen notwendig

Automatisierung in der Produktion

- Fertigungskonzepte und automatisierbare Funktionen -

Kriterium Fördermittel	Förderprinzip	Fördergutbewegung	Bewegungs-einleitung	Kraftein-leitung	Führung	Förder-geschw. (m/s)
Induktiv geführtes Flurförderzeug	unstetig - flurgebunden	ohne Relativbewegung zur Auflage	translatorisch	indirekt - dezentral	Auflage	0,3 .. 1,2
Schleppkreis-förderer	stetig - flurfrei	ohne Relativbewegung zur Auflage, formschlüssige Aufnahme	umlaufend	indirekt - zentral	Auflage	0,2 .. 0,5
Regalförderzeug	unstetig - flurgebunden	ohne Relativbewegung zur Auflage, formschlüssige Aufnahme	translatorisch	indirekt - dezentral	Auflage	0,5 .. 1,5
Angetriebener Transportwagen	unstetig - flurfrei oder flurungebunden	ohne Relativbewegung zur Auflage, formschlüssige Aufnahme	translatorisch	indirekt - dezentral	Auflage	0,5 .. 1,5
Schienenhänge-bahn	unstetig - flurfrei	ohne Relativbewegung zur Auflage, formschlüssige Aufnahme	translatorisch	indirekt - dezentral	Auflage	0,5 .. 1,5
Bandförderer	stetig - flurgebunden	ohne Relativbewegung zur Auflage, formschlüssige Aufnahme	umlaufend	direkt - kraftschlüssig	Führungsele-mente	0,1 .. 0,5
Angetriebene Rollerbahn	stetig - flurgebunden	mit Relativbewegung zur Auflage	rotatorisch	direkt - kraftschlüssig	Führungsele-mente	0,4
Handhabungs-gerät (fahrbar + stationär)	unstetig - flurgebunden	ohne Relativbewegung zur Auflage, formschlüssige Aufnahme	translatorisch	indirekt - dezentral	Auflage	0,5 .. 2,5
Ladeportale	unstetig - flurfrei	ohne Relativbewegung zur Auflage, formschlüssige Aufnahme	translatorisch	indirekt - dezentral	Auflage	0,5 .. 2,0

Tabelle 1-1 Fördermittel in FFS

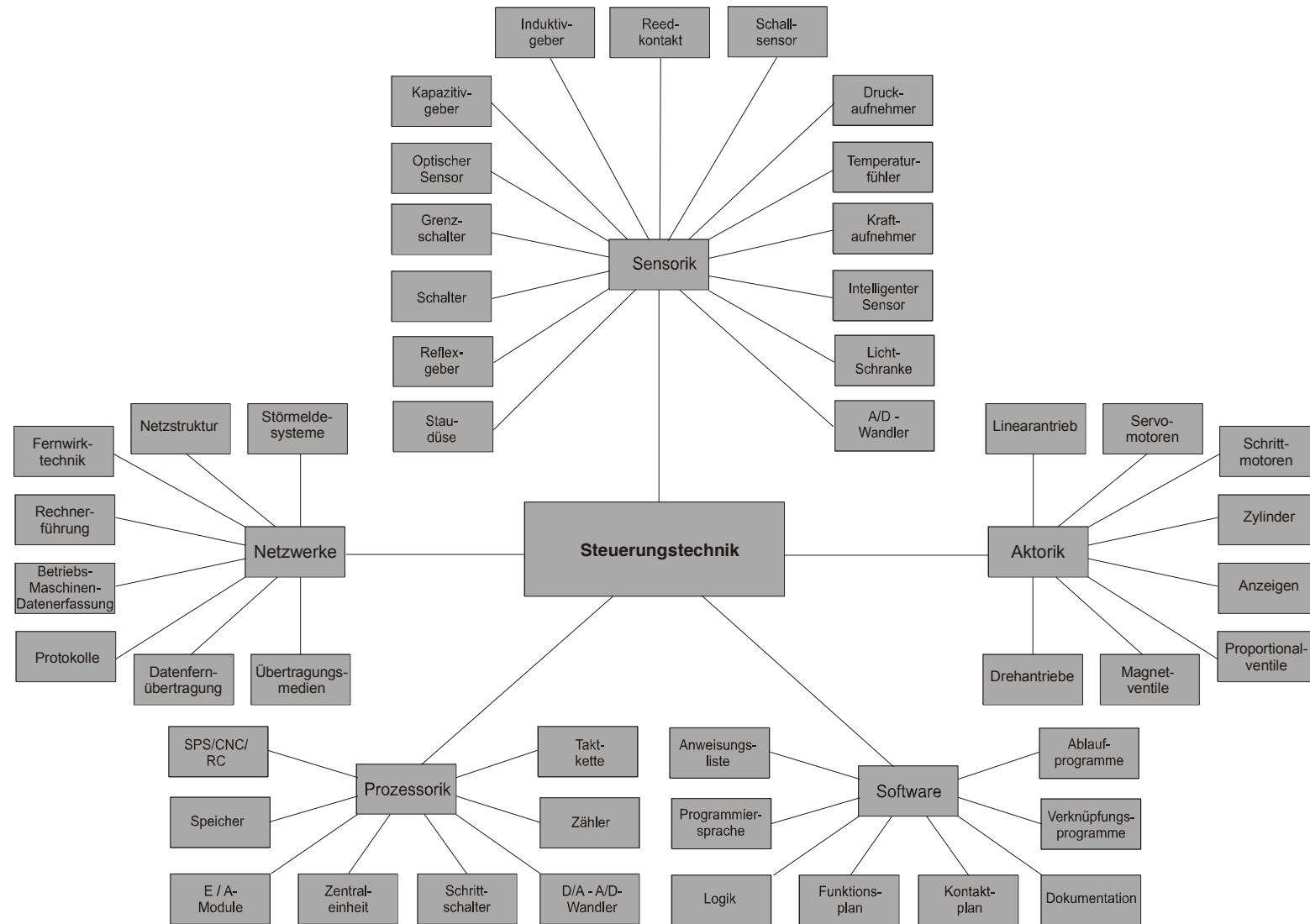
Automatisierung in der Produktion

- Prozesse und Variablen -

•Prozess	•Fließprozesse	•Folgeprozesse	•Stückprozesse
•Variable	•Physikalische Größen mit kontinuierlichem oder stückweise kontinuierlichem Wertebereich (alle Regelstrecken sind technische Systeme, in denen Fließprozesse ablaufen)	•Binäre diskrete Informationen, die Ereignisse melden oder auslösen	•Den Objekten (Stücken) zugeordnete zeitdiskrete oder zeitkontinuierliche Variable mit kontinuierlichem oder diskretem Wertebereich
•Kennzeichen	•Die Variablen sind zeitabhängige oder zeit- und ortsabhängige physikalische Größen (Dynamische Vorgänge)	•Folgen von Einzelereignissen	•Einzelnen identifizierbare Stücke (Objekte), die sich in der Position und/oder ihrem Zustand ändern.
•Beispiele	•- Erzeugungsvorgänge, Umformungsvorgänge, Bewegungsabläufe: •Energieerzeugung, Stahlerzeugung, Zementherstellung, chemische Reaktionen	•- An- und Ablaufvorgänge: •Dampfturbine •- Ablaufvorgänge: •Werkzeugmaschinen, Aufzüge •- Prüfvorgänge: •Geräteprüfung im Prüffeld	•- Fertigungsvorgänge: •Motorenfertigung •- Transportvorgänge: •Schienen- und Straßenverkehr •- Lagervorgänge: Hochregallager
•Mathematische Modelle	•Differentialgleichungen, Übertragungsfunktion	•Boolesche Gleichungen •Ablaufpläne	•Simulationsprogramme, Beschreibung der Systemelemente, Objekte und Fkt.

Automatisierung in der Produktion

- Steuerungstechnik in der Produktion - Überblick -



Automatisierung in der Produktion

- Industrielle Steuerungstechnik - Steuerungen -

• SPS Speicher- programmierbare Steuerungen	• RC Robotersteuerungen	• CNC Computerized Numerical Control	• AS Automatisierungs- systeme (spezialisierte Prozessrechner)
 <ul style="list-style-type: none"> • Steuerung und Überwachung - eines Prozesses - von Maschinen oder Anlagen Echtzeit Bus Systeme (Feldbus, PROFIBUS) Dezentralisierung, WLAN, Mobilfunk, RFID (Industrie 4.0) 	 <ul style="list-style-type: none"> • Bewegungsprogramm • Positionierung • Bahnsteuerung (3-D) • Ablauffolge LAN (Ethernet) Echtzeit Bus Systeme (Feldbus, PROFIBUS) Dezentralisierung, WLAN, Mobilfunk, RFID (Industrie 4.0) 	 <ul style="list-style-type: none"> • Bearbeitungsprogramme • Konturen (2-D) / Bahnen 3D, mehrere Achsen • Bearbeitungskenngrößen (überwachen und steuern) • LAN, Feldbusse, Echtzeitkommunikation 	<ul style="list-style-type: none"> • Prozesssteuerung • Überwachung • Erkennung • Identifizieren • Klassifizieren • LAN Feldbusse Echtzeitkommunikation, WLAN, Mobilfunk, NFC, RFID, (Industrie 4.0)
Einmal- Programmierung SPS Programm	<ul style="list-style-type: none"> • Wiederholte Anwenderprogrammierung • Bewegungsprogramm 	<ul style="list-style-type: none"> • Wiederholte Anwenderprogrammierung • Bearbeitungsprogramm 	<ul style="list-style-type: none"> • Einmal- Programmierung • z.B. Menügeführt • „Funktionsorientiertes“ Programm

Automatisierung in der Produktion

- Vernetzte Automatisierungskomponenten -

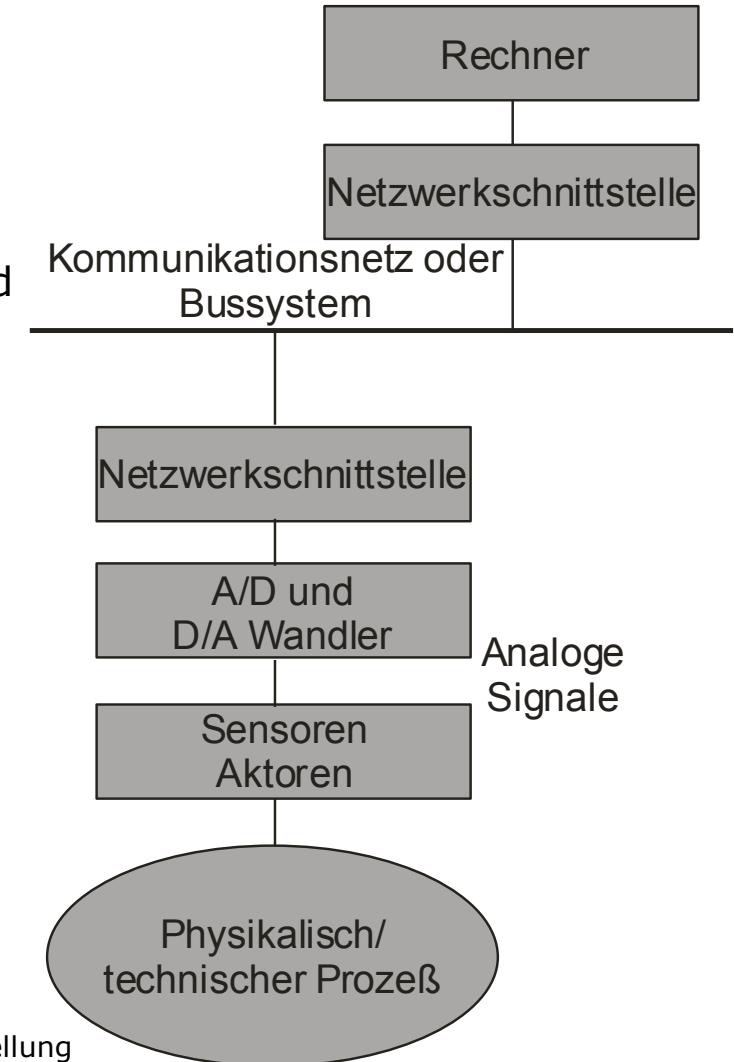
Verteilte Automatisierungskomponenten und Bussysteme:

Industrielle Steuerungen, spezialisierte
Automatisierungssystem (= Rechner)

- Kommunikationskanäle Netzwerkkomponenten,
Bussysteme, z.B. Feldbusse - Profibus, Sensor-Aktor-
Bus; SCADA Systeme) SCADA = Supervisory Control and
Data Acquisition = Überwachen und Steuern technischer
Prozesse durch Computer-Systeme.
- Analog-/Digital-, Digital/Analog-Wandler
- Sensoren
- Aktoren
- der zu steuernde physikalisch-technische Prozess



Bild: Schokoladenherstellung



Automatisierung in der Produktion

- Interne und Externe Sensorik -

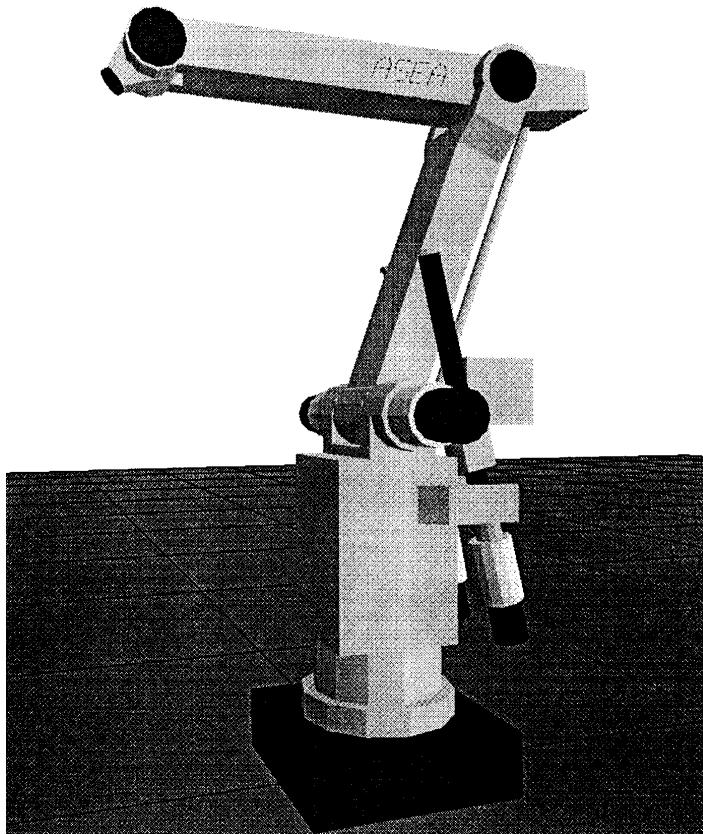


Abb. 1-15

- ASEA-Industrieroboter
erste Simulationsmodelle der 90er
Jahre (Deneb 1993)



KUKA Roboter im Einsatz

Automatisierung in der Produktion

- Interne und Externe Sensorik -

Globale Funktionen der internen Sensorik:

- Positionierung
- Funktionsfähigkeit
- Selbstdiagnose

Interne Sensoren:

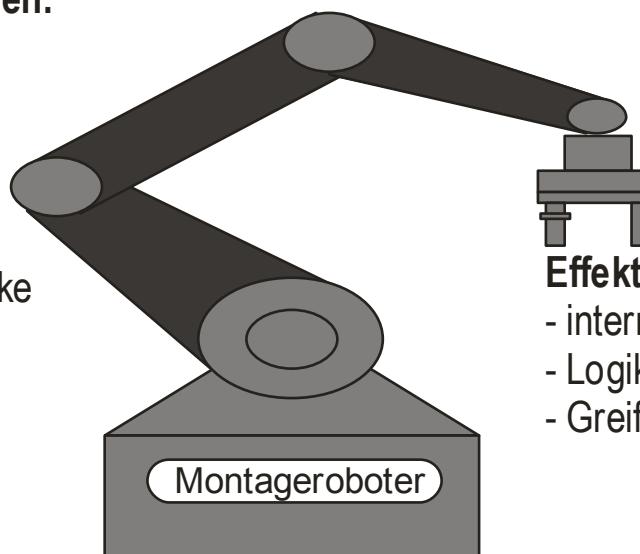
- Meßsysteme
- Weg, Winkel
- Drehzahl
- Stromstärke

Mechanik:

- Achsen, Gelenke
- Antriebe
- Effektoren

Steuerung:

- Programm
- Schnittstellen
- Lageregelung
- Leistung
- Logik



Effektoren:

- interne Sensorik
- Logik
- Greifkraft

Maschineninterne Prozeßparameter:

- Geschwindigkeit
- Beschleunigung
- Greifkraft, etc.
- Funktionsfähigkeit

Automatisierung in der Produktion

- Steuerung und Kaskadenregelung -

Steuerung

- Bewegung der Achsen
- Positionierung
- Überwachung

Leistungselektronik

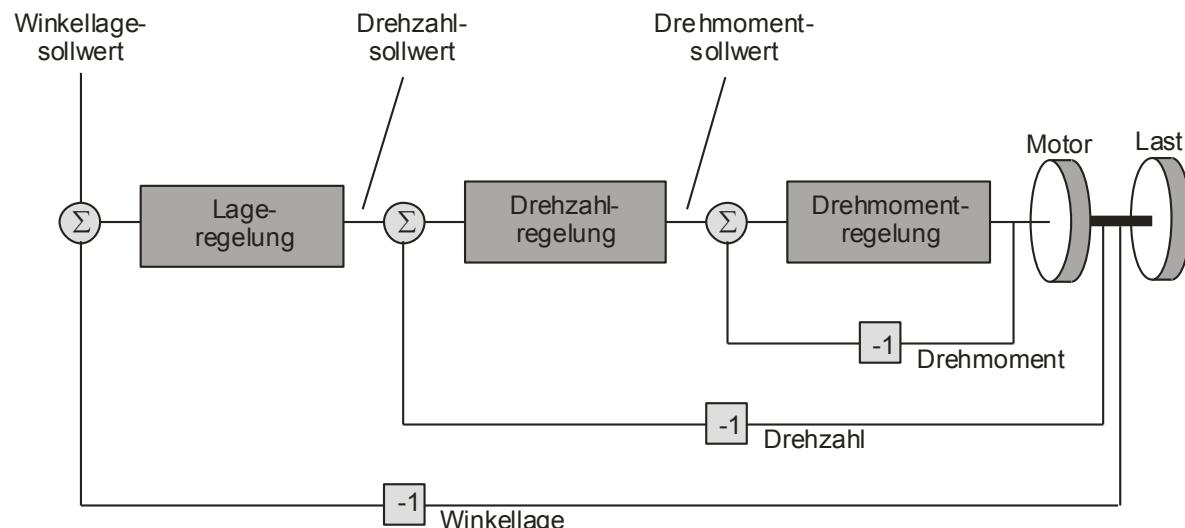
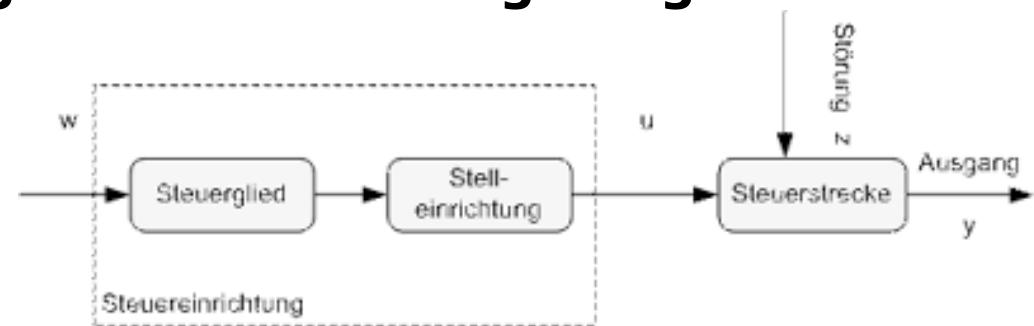
- Bereitstellung ausreichender Antriebsenergie

Antrieb

- Positioniergenauigkeit

Last

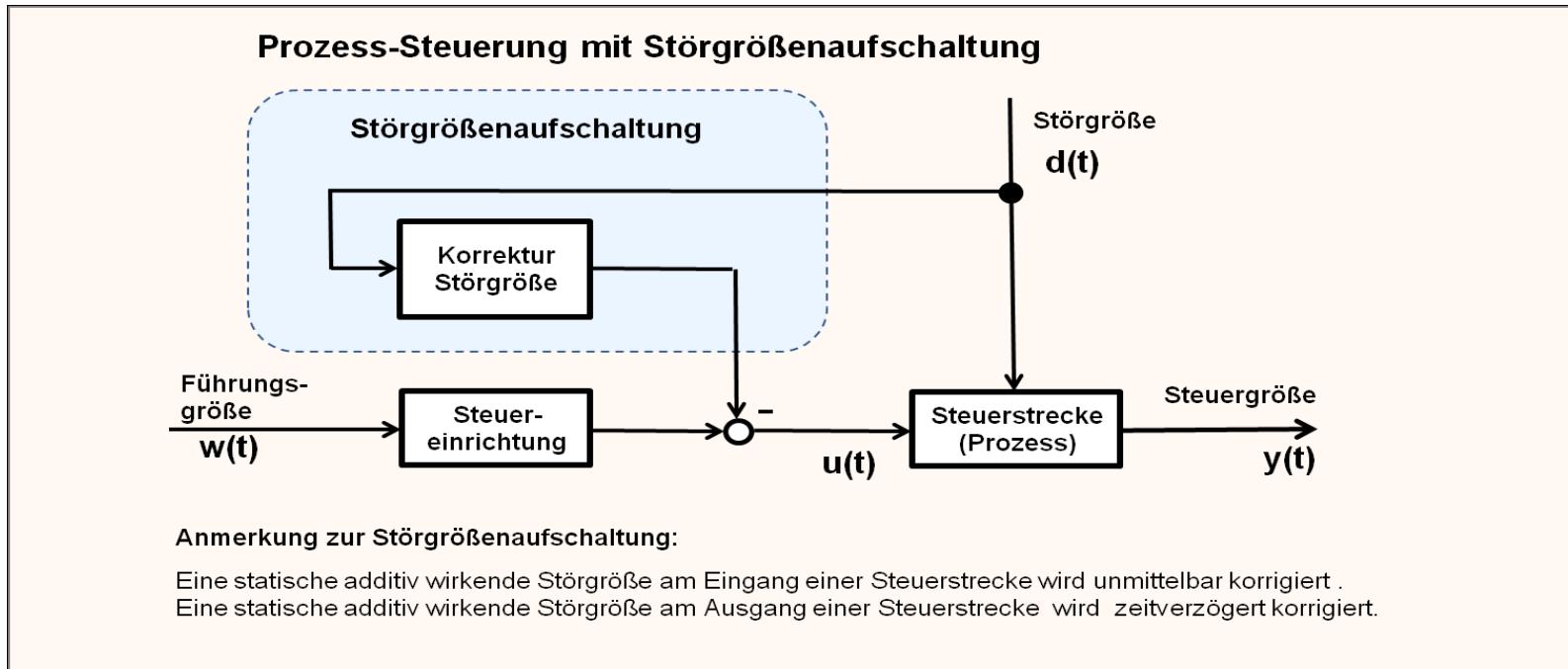
- mechanisches System
- dynamische Belastung



Schematische Darstellung eines Winkellage-Regelkreises

Automatisierung in der Produktion

- Störgrößen -



Automatisierung in der Produktion

- Modelle -

vier Arten mathematischer Modelle:

1. kontinuierlich dynamische Systeme, die durch lineare bzw. nichtlineare Differentialgleichungen beschrieben werden;
2. zeitdiskrete abgetastete dynamische Systeme, die durch lineare oder nichtlineare Differentialgleichungen dargestellt werden;
3. Systeme mit diskreten Ereignissen bzw. Schaltsysteme, die durch diskrete Zustände dargestellt werden (Petri Netze);
4. Systeme mit Ungenauigkeiten, die durch statistische oder linguistische Methoden ausgedrückt werden.

Ein dynamisches System muss in zeitdiskreter Form dargestellt werden, damit ein digitaler Rechner es steuern kann.

Weitere Möglichkeiten der Modellbildung sind beispielsweise heuristische, stochastische und Fehlermodelle.

Lehrziele und Gliederung

1. Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- 2. Automatisierung in der Produktion**
3. Boolesche Algebra
4. Zustandsvektoren, Graphen, Schaltwerksynthese, Netze
5. Fuzzy Logic
6. Neuronale Netze
7. Automatisieren von Fertigungsbereichen
8. Automatisiertes Messen und Steuern
9. Speicherprogrammierbare Steuerungen

Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 03 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



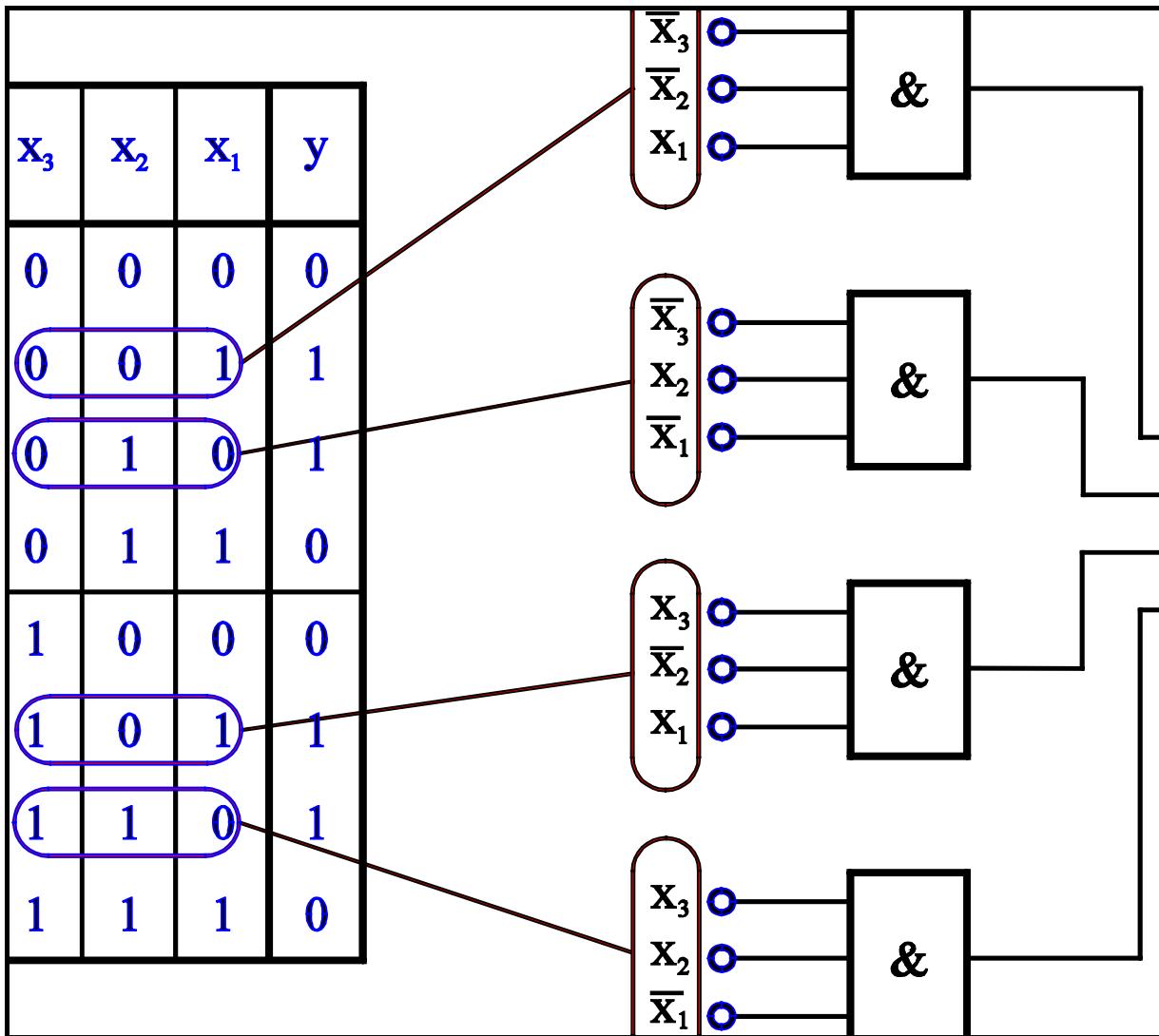
Bremer Institut für
Messtechnik, Automatisierung
und Qualitätswissenschaft

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1**
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Boolesche Algebra

- Boolesche Funktionen -



Boolesche Algebra

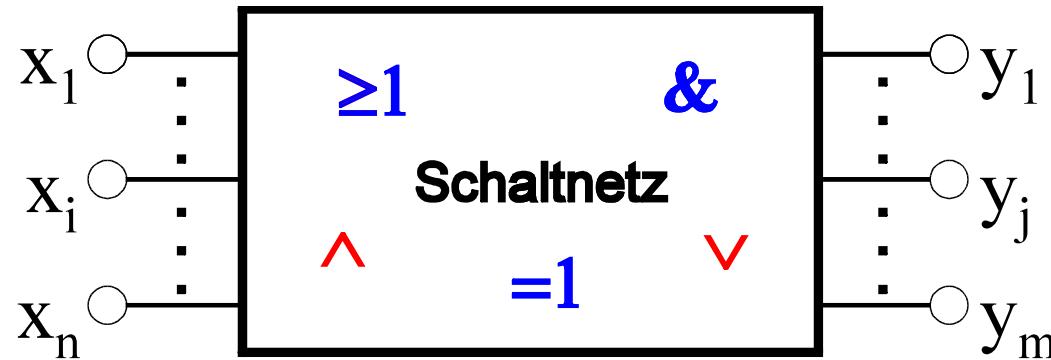
- Boolesche Funktionen -

- Aussagelogik liefert die Grundlage für die Boolesche Algebra
- vergleiche Theorie zur "mathematischen Logik"
- Eine Aussage ist jeder Satz, der entweder wahr oder falsch ist {w,f}
- Schaltfunktionen und Schaltnetze
- f_{Schlt} : Abbildung von $\{0,1\}^n \rightarrow \{0,1\}^m$
- Sie bildet die Menge mit n-Tupel mit n Eingängen in die Menge der m-Tupel mit m Ausgängen ab (Tupel: Liste mathematischer Objekte)

- Literatur: Grundlagen der Informatik/ Math. Logik und Automatentheorie
- z.B. Rembold /Levi: Einführung in die Informatik

Boolesche Algebra

- Definition des Schaltnetzes -



Schaltfunktionen

$$y_j = f(x_1, \dots, x_i, \dots, x_n)$$

$$x_i, y_j \in \{0, 1\}$$

DIN 44300: Ein Schaltnetz ist ein Schaltwerk, dessen Ausgangswert zu irgendeinem Zeitpunkt nur von Wert am Eingang zu diesem Zeitpunkt abhängt

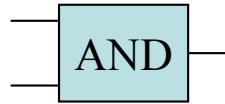
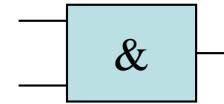
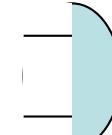
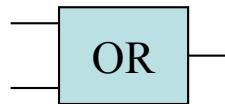
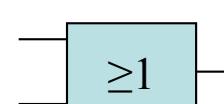
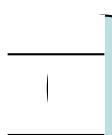
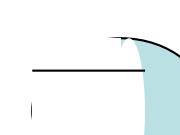
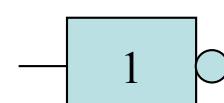
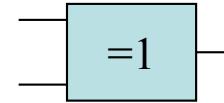
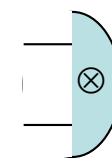
Boolesche Algebra

- Schaltverknüpfungen -

	NEGATION auch NICHT-Verknüpfung	KONJUNKTION auch UND-Verknüpfung	DISJUNKTION auch ODER-Verknüpfung
Logische Darstellung	—	\wedge	\vee
Beispiel	$y = \bar{x}$	$y = x_1 \wedge x_2$	$y = x_1 \vee x_2$
Definition durch Wert der Schaltfunktion	$\begin{array}{l} \bar{0} = 1 \\ \bar{1} = 0 \end{array}$	$\begin{array}{l} 0 \wedge 0 = 0 \\ 0 \wedge 1 = 0 \\ 1 \wedge 0 = 0 \\ 1 \wedge 1 = 1 \end{array}$	$\begin{array}{l} 0 \vee 0 = 0 \\ 0 \vee 1 = 1 \\ 1 \vee 0 = 1 \\ 1 \vee 1 = 1 \end{array}$
Schaltsymbol DIN 40700 seit 1976			
Repräsentation mit Schaltern			

Boolesche Algebra

- Normen und Symbole -

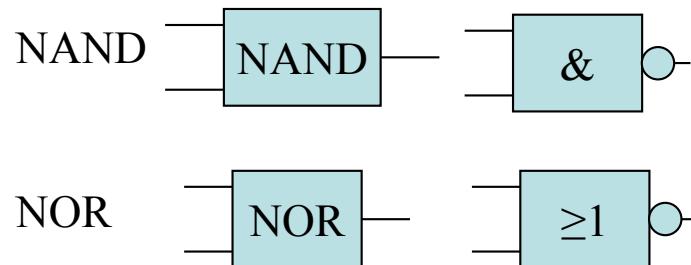
	Symbol (Funktion)	Schaltsymbole DIN 40 700 seit 1976 IEC 60617-12	Schaltsymbole DIN 40 700 bis 1976	Amerikanische Symbole	logische Darstellung
AND					$(x_1 \wedge x_2)$
OR					$(x_1 \vee x_2)$
NOT					\bar{x}_1
XOR					$(x_1 \otimes x_2)$
					$(x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2)$

Boolesche Algebra

- Normen und Symbole -

Symbole
(Funktion)

Schaltsymbole
DIN 40 700
seit 1976
IEC 60617-12



Schaltsymbole
DIN 40 700
bis 1976

Amerikanische
Symbole

logische Darstellung
 $(x_1 \wedge x_2)$

$$\overline{(x_1 \wedge x_2)}$$

$$\overline{(x_1 \vee x_2)}$$

Freie Symbolik (beliebige Schaltfunktion)



$$\begin{aligned} & (x_1 \wedge \bar{x}_1 \wedge x_2) \vee (x_1 \wedge x_2 \wedge \bar{x}_2) \\ & (x_1 \wedge \bar{x}_1 \wedge x_2) \vee (x_1 \wedge x_2 \wedge \bar{x}_2) \\ & (x_1 \wedge \bar{x}_1 \wedge x_2) \vee (x_1 \wedge x_2 \wedge \bar{x}_2) \\ & (\overline{\bar{x}_1 \wedge x_2}) \wedge \neg x_2 \vee \bar{x}_3 \wedge x_4 \vee x \wedge \bar{x}_2 \wedge x \wedge y_2 \end{aligned}$$

Boolesche Algebra

- Die Gesetze der Booleschen Algebra -

	UND	ODER
1.	$x_1 \wedge x_2 = x_2 \wedge x_1$	$x_1 \vee x_2 = x_2 \vee x_1$
2.	$x_1 \wedge x_2 \wedge x_3 = (x_1 \wedge x_2) \wedge x_3$ $= x_1 \wedge (x_2 \wedge x_3)$ $= x_2 \wedge (x_1 \wedge x_3)$	$x_1 \vee x_2 \vee x_3 = (x_1 \vee x_2) \vee x_3$ $= x_1 \vee (x_2 \vee x_3)$ $= x_2 \vee (x_1 \vee x_3)$
3.	$x_1 \wedge (x_2 \vee x_3) = (x_1 \wedge x_2) \vee (x_1 \wedge x_3)$	$x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3)$
4.	$x \wedge 1 = x$	$x \vee 1 = 1$
5.	$x \wedge 0 = 0$	$x \vee 0 = x$
6.	$x \wedge \bar{x} = 0$	$x \vee \bar{x} = 1$
7.	$x_1 \wedge (x_2 \vee x_3) = x_1 \wedge x_4$ mit $x_4 = x_2 \vee x_3$	$x_1 \vee (x_2 \wedge x_3) = x_1 \vee x_4$ mit $x_4 = x_2 \wedge x_3$

1.: Kommutativgesetz
 2.: Assoziativgesetz
 3.: Distributivgesetz
 4.: Einsgesetz
 5.: Nullgesetz
 6.: Komplementgesetz
 7.: Substitution

Boolesche Algebra

- Die Sätze der Booleschen Algebra -

$$1. \quad (\overline{\overline{x}}) = x$$

UND

$$2. \quad x \wedge x = x$$

$$3. \quad x_1 \wedge (x_1 \vee x_2) = x_1$$

$$4. \quad x_1 \wedge (\bar{x}_1 \vee x_2) = x_1 \wedge x_2$$

$$5. \quad (x_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_2) = x_1$$

ODER

$$x \vee x = x$$

$$x_1 \vee (x_1 \wedge x_2) = x_1$$

$$x_1 \vee (\bar{x}_1 \wedge x_2) = x_1 \vee x_2$$

$$(x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) = x_1$$

$$6. \text{ De Morgansches Theorem} \quad y(x_i, \bar{x}_i, 0, 1, \wedge, \vee) = \bar{y}(\bar{x}_i, x_i, 1, 0, \vee, \wedge)$$

Beispiel 1: Satz 3a

Behauptung:

$$x_1 \wedge (x_1 \vee x_2) = x_1$$

$$\begin{aligned} x_1 \wedge (x_1 \vee x_2) &= (x_1 \vee 0) \wedge (x_1 \vee x_2) \\ &= x_1 \vee (0 \wedge x_2) \\ &= x_1 \vee 0 \\ &= x_1 \end{aligned}$$

wg. Nullgesetz

wg. Distributivgesetz

wg. Nullgesetz

qed.

Boolesche Algebra

- De Morgan Theorem -

Der Beweis für das De Morgansche Theorem in seiner allgemeinen Form ist nicht so einfach und auch aufwendig. Wir beschränken uns darum auf den Beweis eines Sonderfalles mit zwei Variablen x_1 und x_2 .

Beispiel 2:

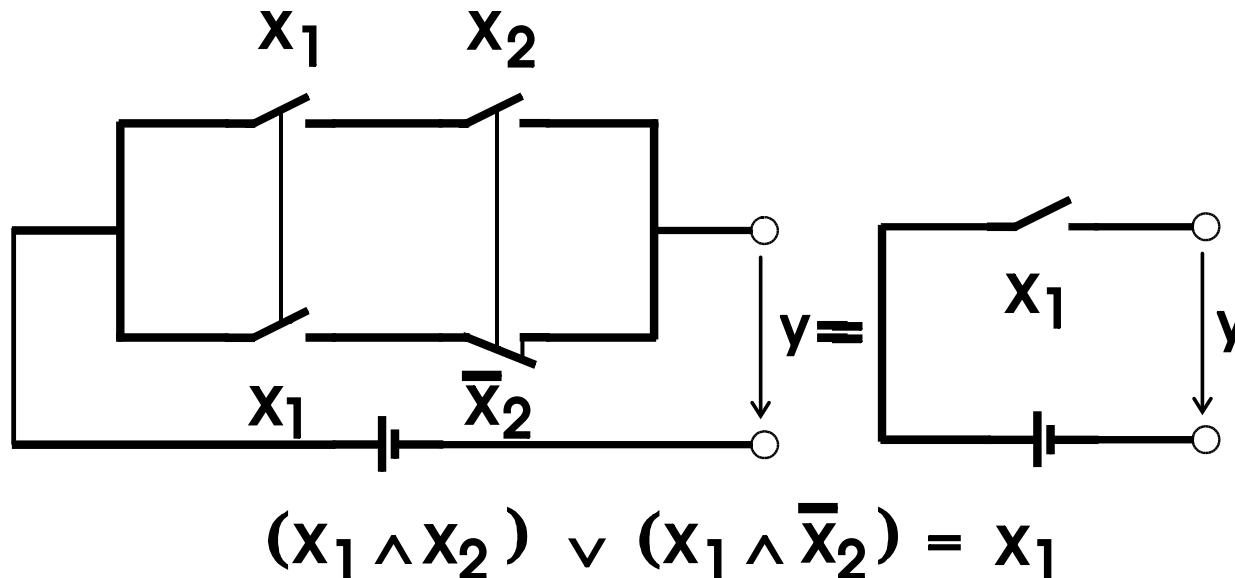
Behauptung:

$$\begin{aligned} x_1 \wedge x_2 = y &\Rightarrow \bar{x}_1 \vee \bar{x}_2 = \bar{y} \\ y \wedge \bar{y} = 0 &= (x_1 \wedge x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) && \text{wg. Komplementgesetz} \\ &= (x_1 \wedge x_2 \wedge \bar{x}_1) \vee (x_1 \wedge x_2 \wedge \bar{x}_2) && \text{wg. Distributivgesetz} \\ &= (x_1 \wedge \bar{x}_1 \wedge x_2) \vee (x_1 \wedge x_2 \wedge \bar{x}_2) && \text{wg. Kommutativgesetz} \\ &= ((x_1 \wedge \bar{x}_1) \wedge x_2) \vee (x_1 \wedge (x_2 \wedge \bar{x}_2)) && \text{wg. Assoziativgesetz} \\ &= (0 \wedge x_2) \vee (x_1 \wedge 0) && \text{wg. Komplementgesetz} \\ &= 0 \vee 0 && \text{wg. Nullgesetz} \\ &= 0 && \text{trivial} \end{aligned}$$

qed.

Boolesche Algebra

- Boolesche Verknüpfungen -



Zur Veranschaulichung eines Satzes der Booleschen Algebra

Boolesche Algebra

- Vollständige Wertetabelle -

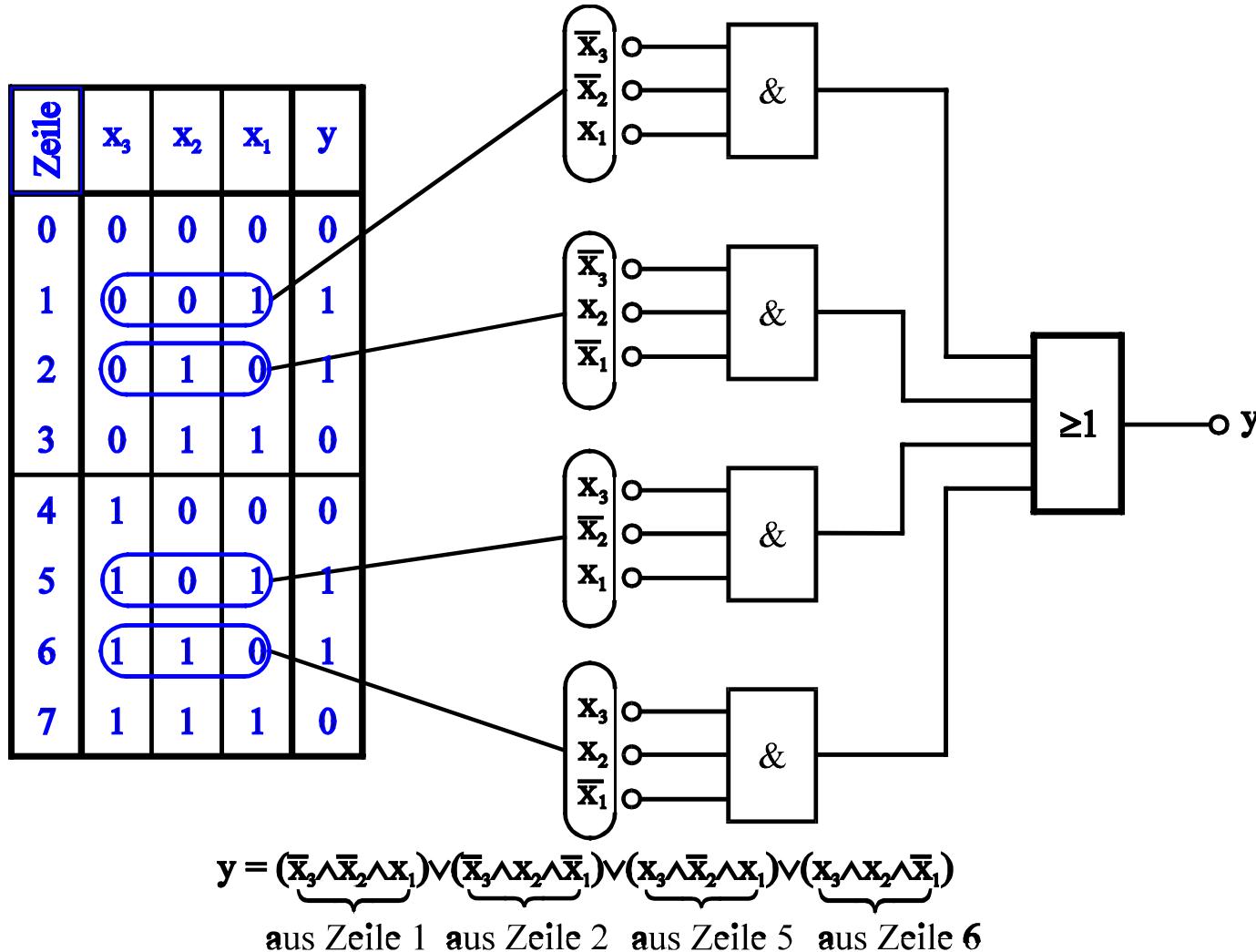
n Eingangsvariable
 $k=2^n$
Zahl der Schaltfunktionen
 $f=2^{2^n}$

Vollständige Wertetabelle für 3 Eingangsvariablen

Zeile	x_3	x_2	x_1	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Boolesche Algebra

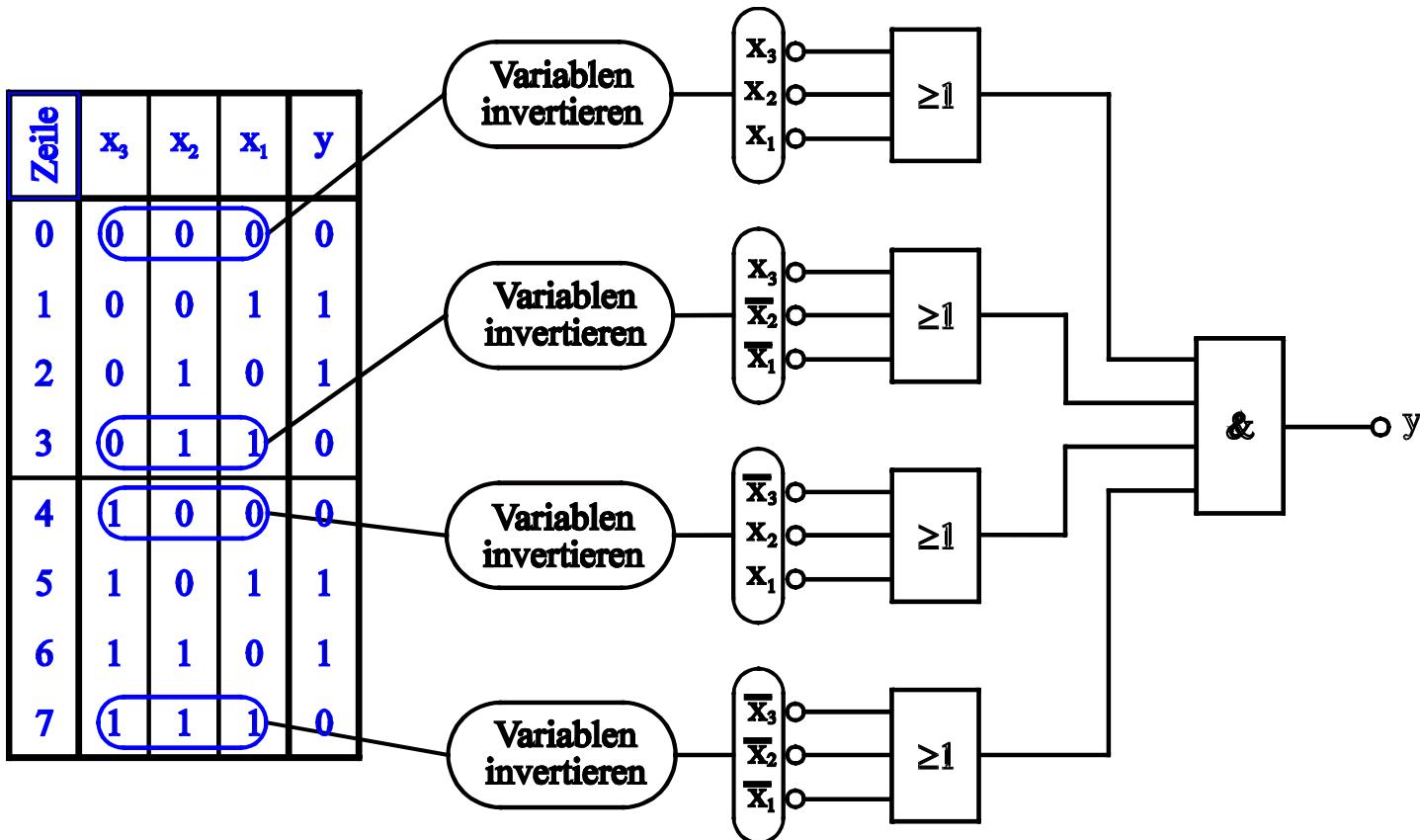
- Disjunktive Normalform -



Die „Wahrheitstabelle“ und die disjunktive Normalform

Boolesche Algebra

- Boolesche Funktionen -



$$y = (\underbrace{x_3 \vee x_2 \vee x_1}_{\text{aus Zeile 0}}) \wedge (\underbrace{x_3 \vee \bar{x}_2 \vee \bar{x}_1}_{\text{aus Zeile 3}}) \wedge (\underbrace{\bar{x}_3 \vee x_2 \vee x_1}_{\text{aus Zeile 4}}) \wedge (\underbrace{\bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1}_{\text{aus Zeile 7}})$$

aus Zeile 0 aus Zeile 3 aus Zeile 4 aus Zeile 7
 Variablen Variablen Variablen Variablen
 invertiert invertiert invertiert invertiert

Die „Wahrheitstabelle“ und die konjunktive Normalform

Boolesche Algebra

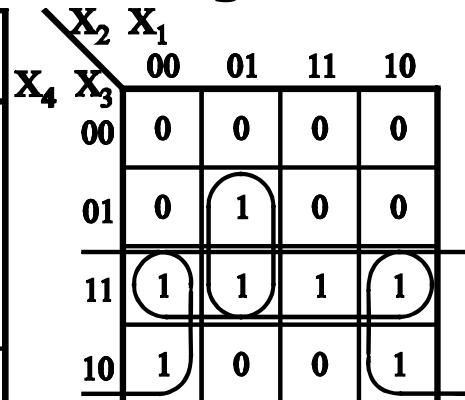
- Boolesche Funktionen -

- Aus der DNF oder der KNF lässt sich immer ein Schaltnetz ableiten, das die Schaltfunktion erfüllt.
- Dieses Schaltnetz ist jedoch nicht optimal bezüglich der Zahl der Verknüpfungen bzw. Bauelemente, die für seine physikalische Abbildung verwendet werden.

Boolesche Algebra

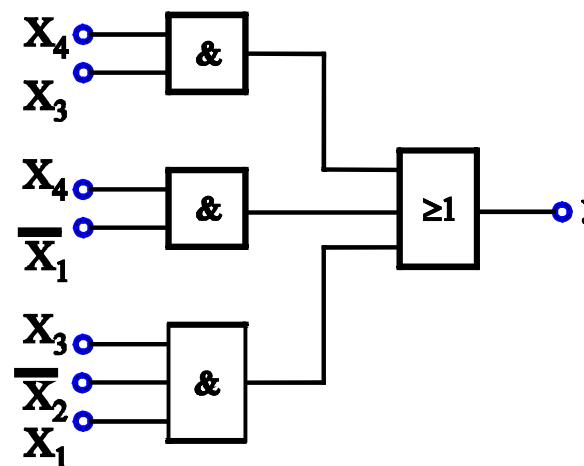
- Karnaugh-Veitch Diagramme -

Zeile	X_4	X_3	X_2	X_1	y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1



Gruppe	erfüllt von
1	$X_4 \wedge X_3$
2	$X_4 \wedge \overline{X}_1$
3	$X_3 \wedge \overline{X}_2 \wedge X_1$

$$y = (X_4 \wedge X_3) \vee (X_4 \wedge \overline{X}_1) \vee (X_3 \wedge \overline{X}_2 \wedge X_1)$$

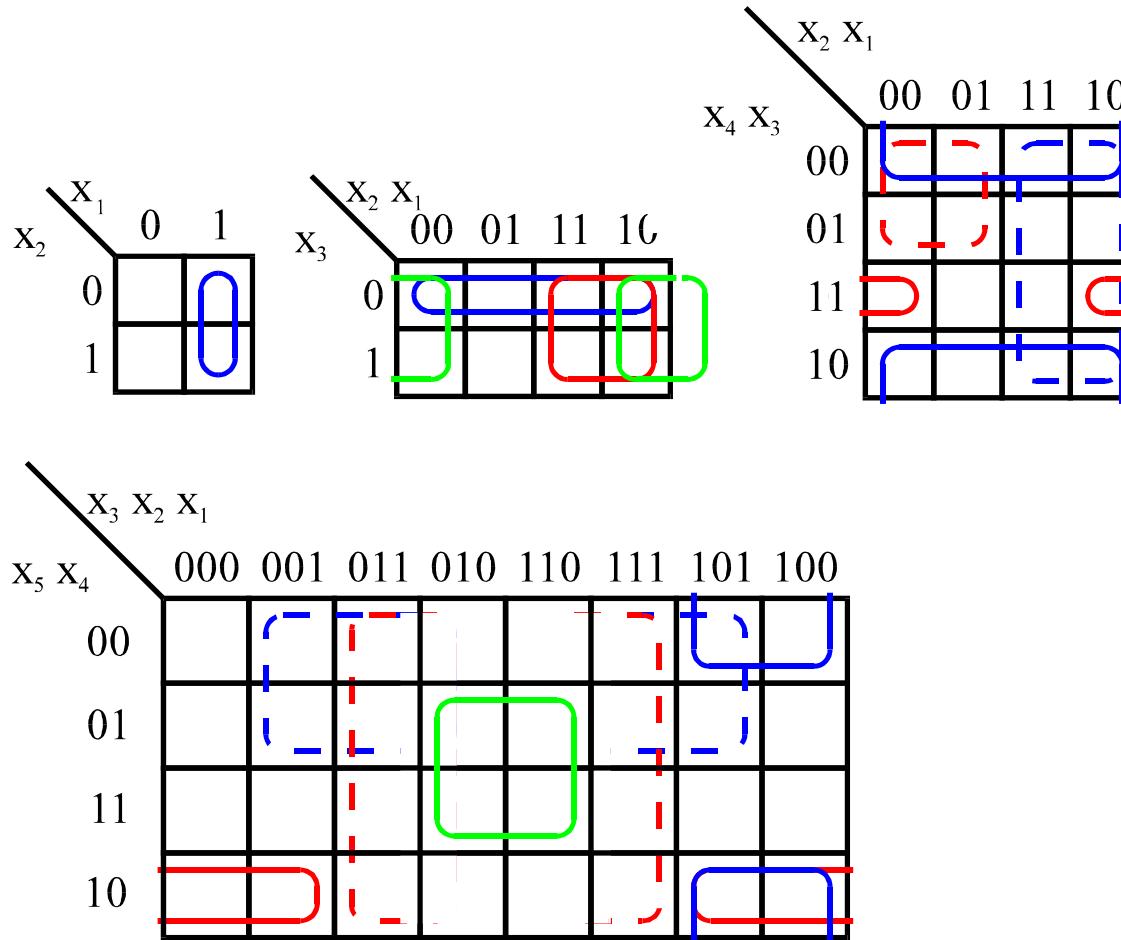


Minimierung mit Karnaugh-Veitch Diagramm

Boolesche Algebra

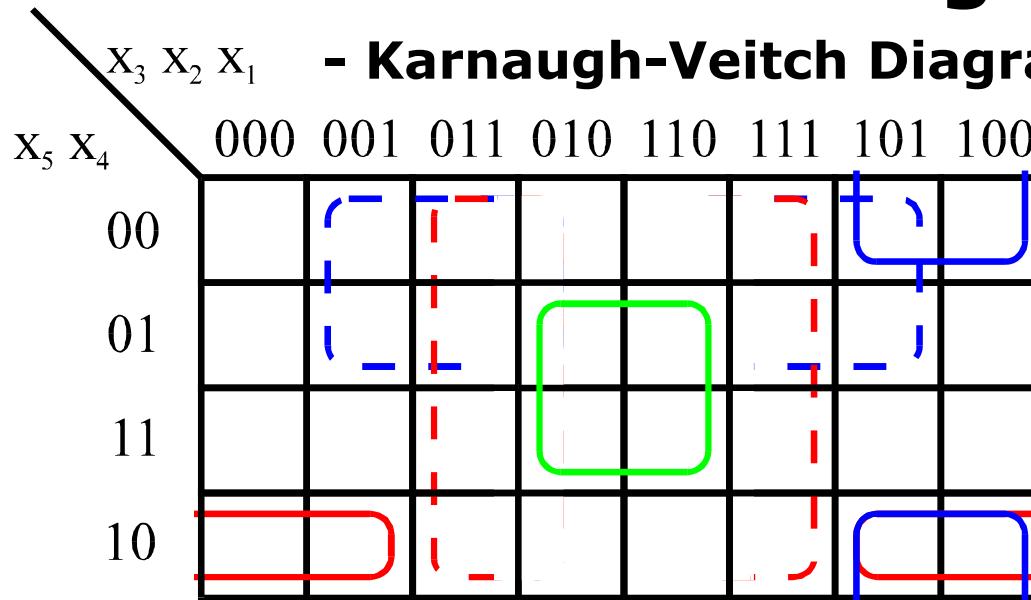
- Karnaugh-Veitch Diagramme -

Beispiele für Gruppenbildung in Karnaugh-Diagrammen



Boolesche Algebra

- Karnaugh-Veitch Diagramme -

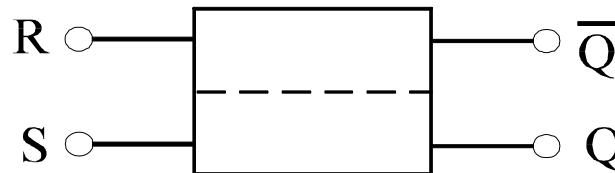


- Ist die Schaltfunktion von fünf Eingangsvariablen abhängig, vergrößern sich die Gruppen durch Berücksichtigung von Symmetrien entlang der in Abb. 2.4-10 senk-rechten Mittelachse. Ist eine **Gruppe n Felder** groß, so sind die in dieser Gruppe enthaltenen Konjunktionen der disjunktiven Normalform **unabhängig** von $m = \text{Id}(n)$ Schaltvariablen. Eine Gruppe ist von einer Schaltvariablen unabhängig, wenn die Schaltvariable innerhalb dieser Gruppe beide möglichen Schaltzustände einnehmen kann.
- Beispiel
- 4 Felder (**Mitte**) $m = \text{Id}(4)=2$ d.h. abhängig von $(\bar{x}_1 \wedge x_2 \wedge x_4)$ unabhängig von $(x_5 \wedge x_3)$
- 8 Felder (**Mitte**) $m = \text{Id}(8)=3$ d.h. abhängig von $(x_1 \wedge x_2)$ unabhängig von $(x_3 \wedge x_4 \wedge x_5)$

Boolesche Algebra

- Speicherelemente -

Symbol



Speicherelemente zum Aufbau von Schaltwerken

- Nachdem zunächst Methoden zur Behandlung von Schaltnetzen vorgestellt wurden, zeigt der folgende Teil grundlegende Bauelemente mit Speicherfunktion.
- Solche Speicherelemente werden Flipflops genannt. Flipflops sind bistabile Kippschaltungen mit zwei stabilen Ausgangssignalzuständen.
- Vergl. EN60617-12

Boolesche Algebra

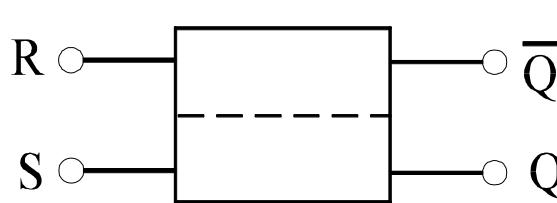
- Speicherelemente, das RS Flipflop -

Wertetabelle

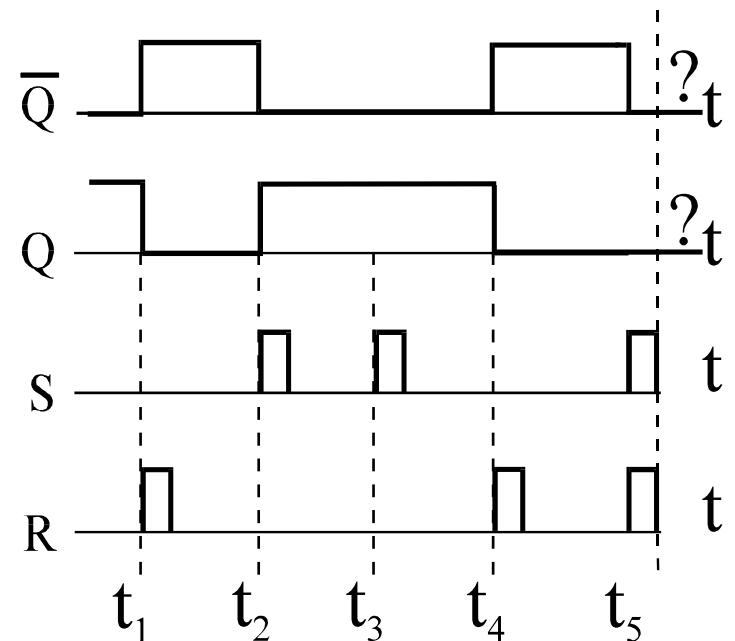
Zeilе	R_{ξ}	S_{ξ}	Q_{ξ}	$Q_{\xi+1}$
1	0	0	0	0
2	0	0	1	1
$t_2:$	3	0	1	0
$t_3:$	4	0	1	1
	5	1	0	0
$t_{1,4}:$	6	1	0	1
$t_5:$	7	1	1	0
$t_5:$	8	1	1	1

unbestimmt

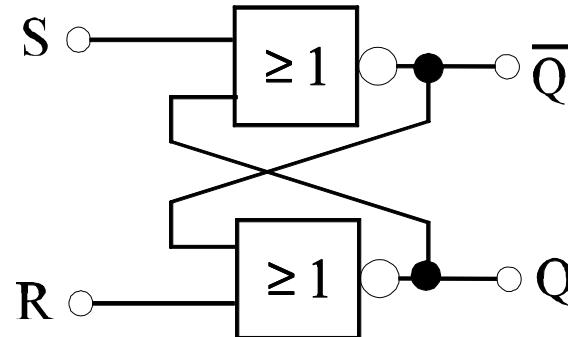
Symbol



Zeitdiagramm



Schaltung



Charakteristische Gleichung

$$Q_{\xi+1} = [S \vee (\bar{R} \wedge Q)]_{\xi}$$

für $R \wedge S = 0$

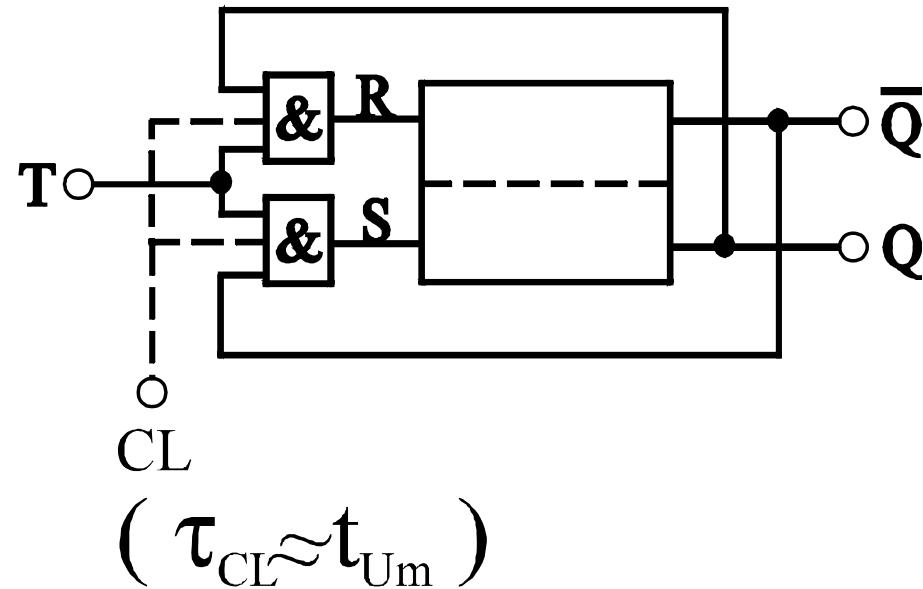
Boolesche Algebra

- Speicherelemente, das T-Flipflop -

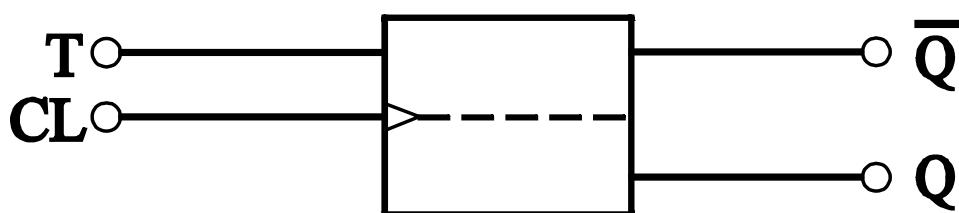
Wertetabelle

Zeile	T_ξ	Q_ξ	$Q_{\xi+1}$
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

T-FF aus RS-FF



Symbol



Charakt. Gleichung

$$Q_{\xi+1} = [(T \wedge \bar{Q}) \vee (\bar{T} \wedge Q)]_\xi$$

Boolesche Algebra

- Speicherelemente, das T-Flipflop -

- Zeitlicher Verlauf T-FF

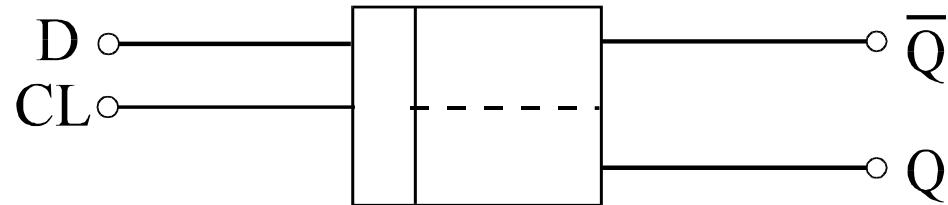
Boolesche Algebra

- Speicherelemente, das D-Flipflop -

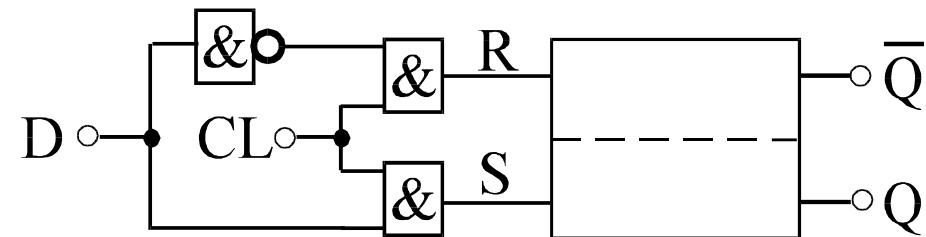
Wertetabelle

Zeile	D_ξ	Q_ξ	$Q_{\xi+1}$
1	0	0	0
2	0	1	0
3	1	0	1
4	1	1	1

Symbol



D-Latch aus RS-FF

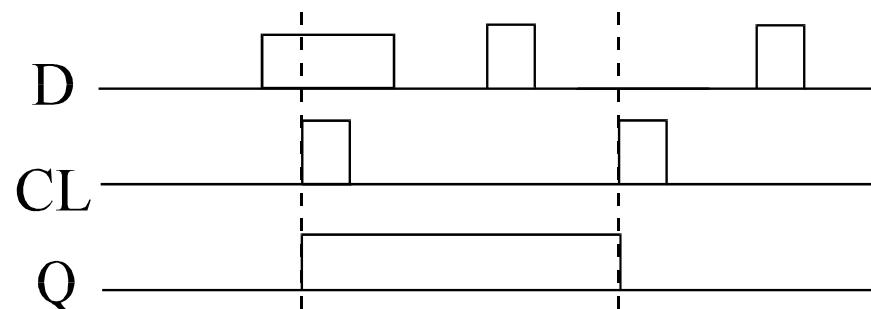


Charakt. Gleichung

$$Q_{\xi+1} = D_\xi$$

(speichert D in Q bis $D=0$ und $CL=1$)

Zeitdiagramm



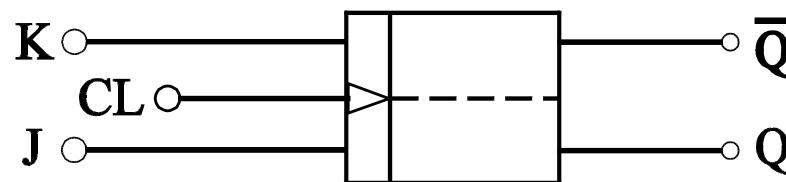
Boolesche Algebra

- Speicherelemente, das JK-Flipflop -

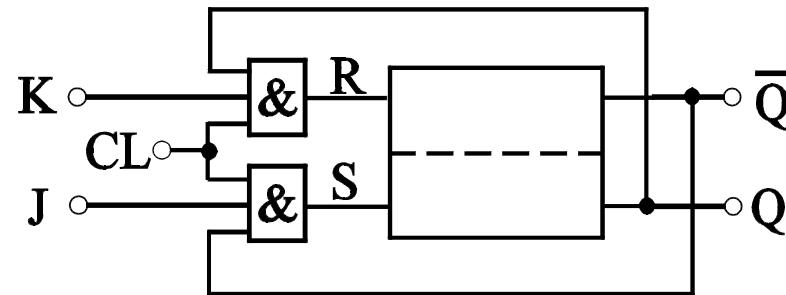
Wertetabelle

Zeile	J_ξ	K_ξ	Q_ξ	$Q_{\xi+1}$
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

Symbol



JK-FF aus RS-FF



Charakt. Gleichung

$$Q_{\xi+1} = [(\bar{K} \wedge Q) \vee (J \wedge \bar{Q})]_\xi$$

Boolesche Algebra

- Speicherelemente, das JK-Flipflop -

CK	J	K	Q_ξ	$Q_{\xi+1}$
X	0	0	0	0
X	0	0	1	1
↑	0	1	X	0
↑	1	0	X	1
↑	1	1	Toggle	

Verkürzte Darstellung der Wertetabelle mit 4 Eingängen
 CK Clock und -> steht für die Flanke (z.B. Vorderflanke)
 X (0 oder 1) Zeile 3 Rücksetzen, Zeile 4 Setzen
 Toggle: 0 -> 1 ; 1 -> 0
 (hin und herschalten, das FF „schwingt“ mit der Frequenz des Clock Signals bei JK=1)

Boolesche Algebra

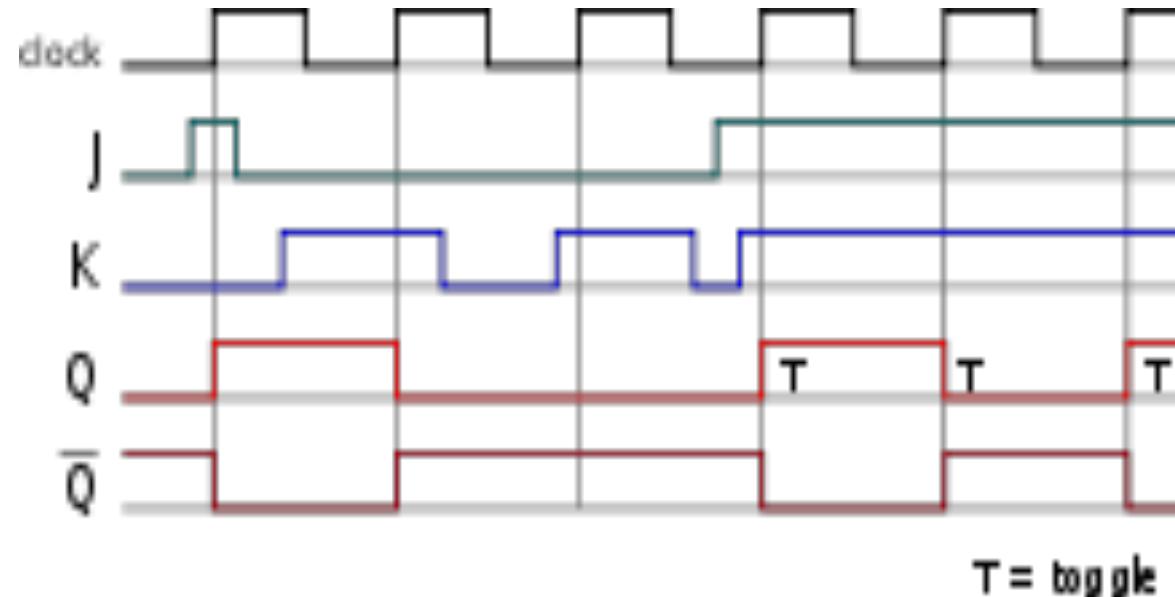
- Speicherelemente, das JK-Flipflop -

Zei le	CK	J	K	Q	Q+ 1	
0	0	0	0	0	0	
1	0	0	0	1	1	
2	0	0	1	0	0	
3	0	0	1	1	1	
4	0	1	0	0	0	
5	0	1	0	1	1	
6	0	1	1	0	0	
7	0	1	1	1	1	
8	1	0	0	0	0	
9	1	0	0	1	1	
10	1	0	1	0	0	
11	1	0	1	1	0	
12	1	1	0	0	1	
13	1	1	0	1	1	
14	1	1	1	0	1	
15	1	1	1	1	0	

Boolesche Algebra

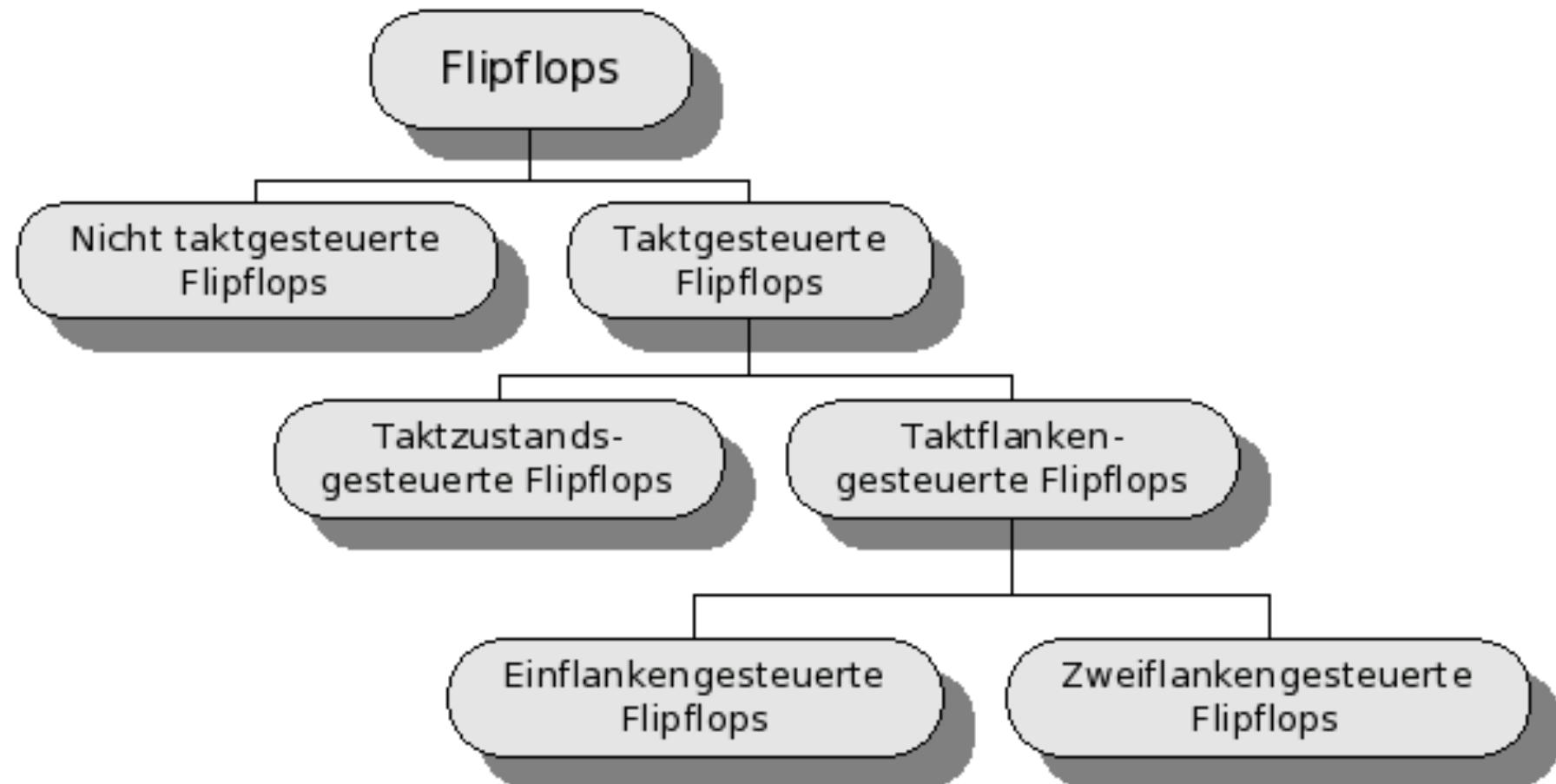
- Speicherelemente, das JK-Flipflop -

- Zeitlicher Verlauf JK-FF



Boolesche Algebra

- Speicherelemente, Flipflops -



Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1**
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 04 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



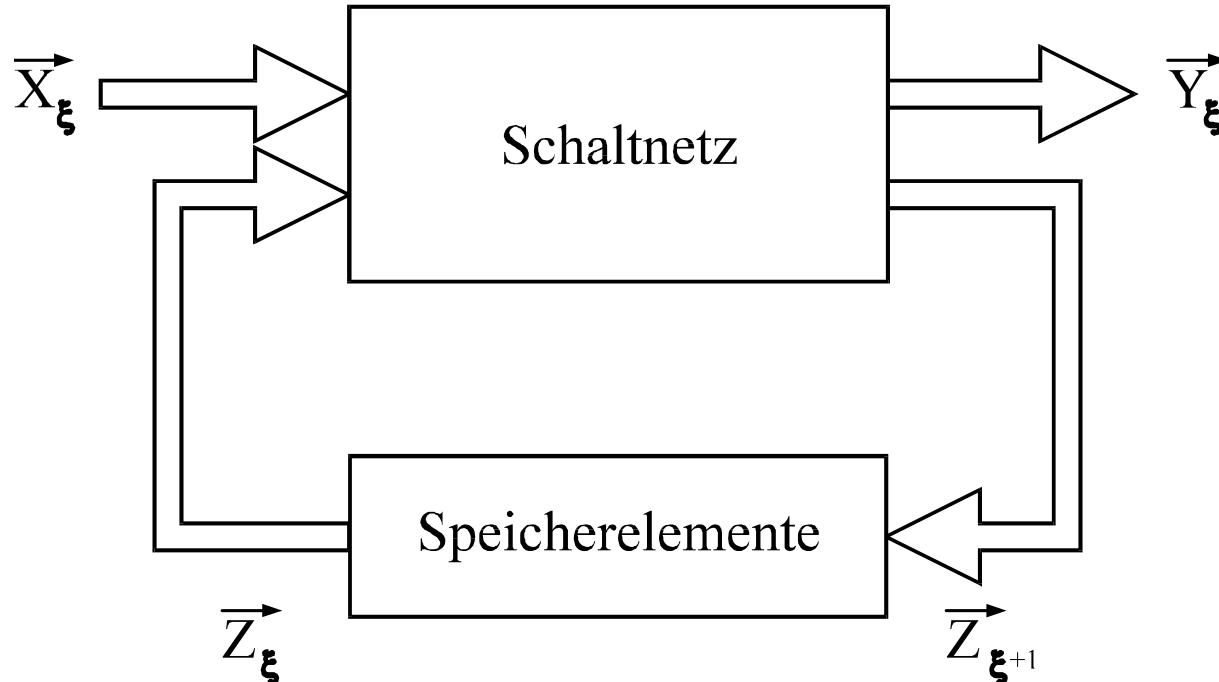
Bremer Institut für
Messtechnik, Automatisierung
und Qualitätswissenschaft

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen**
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Boolesche Algebra

- Struktur von Schaltwerken -



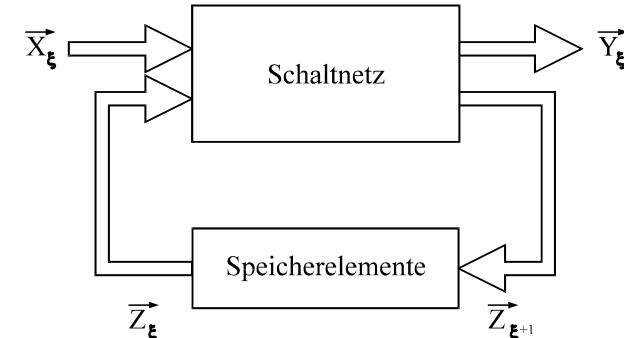
X: Der *Eingangsvektor* des Schaltwerks setzt sich aus den binären Eingangsvariablen zusammen.

Y: Der *Ausgangsvektor* setzt sich aus den binären Ausgangsvariablen zusammen.

Z: Der *Zustandsvektor* des Schaltwerks setzt sich aus den binären Zustandsvariablen zusammen.

Boolesche Algebra

- Schaltwerksgraphen -



Zu einem diskreten Zeitpunkt ξ

befinden sich die Speicherelemente des Schaltwerks in den Zuständen \vec{Z}_ξ

Die Änderung der Eingangsvariablen auf den Wert $\vec{X}_{\xi+1}$

überführt das Schaltwerk in die Zustände $\vec{Z}_{\xi+1}$

Der Zeitindex ξ

bezeichnet also den Zeitpunkt des Zustandsübergangs im Schaltwerk.

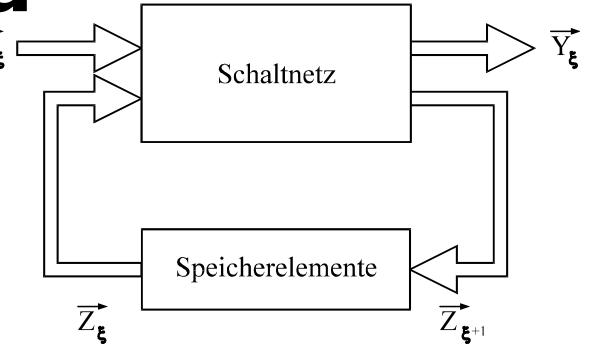
Die Ausgangsvariablen \vec{Y}_ξ

des Schaltwerks werden aus den Vektoren \vec{X}_ξ und \vec{Z}_ξ
abgeleitet. Es gilt also: $\vec{Z}_{\xi+1} = \delta(\vec{X}, \vec{Z})_\xi$ Vektor der binären Übergangsfunktion

$\vec{Y}_\xi = \omega(\vec{X}, \vec{Z})_\xi$ Vektor der binären Ausgangsfunktion

Boolesche Algebra

- Boolesche Funktionen -



δ bezeichnet dabei als den Vektor der binären Übergangsfunktionen und ω als den Vektor der binären Ausgangsfunktionen des Schaltwerks. Schaltwerke, die der allgemeinen Ausgangsfunktionen $\vec{Y}_\xi = \omega(\vec{X}, \vec{Z})_\xi$

genügen, bezeichnet man als **Mealy**-Automaten oder Schaltwerke vom Typ T, da

\vec{Y}_ξ total von \vec{X} und \vec{Z} abhängt.

Schaltwerke, für die gilt:

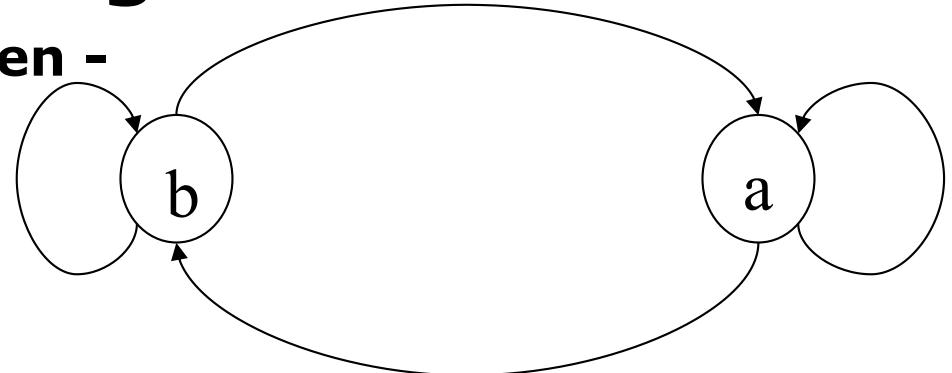
$$\vec{Y}_\xi = \omega(\vec{Z})_\xi$$

werden dagegen als **Moore**-Automaten oder als Schaltwerke vom Typ I bezeichnet,

da \vec{Y}_ξ nur vom internen Zustand abhängt.

Boolesche Algebra

- Graphen -



Graphen (vergl. Automaten-, Graphentheorie)

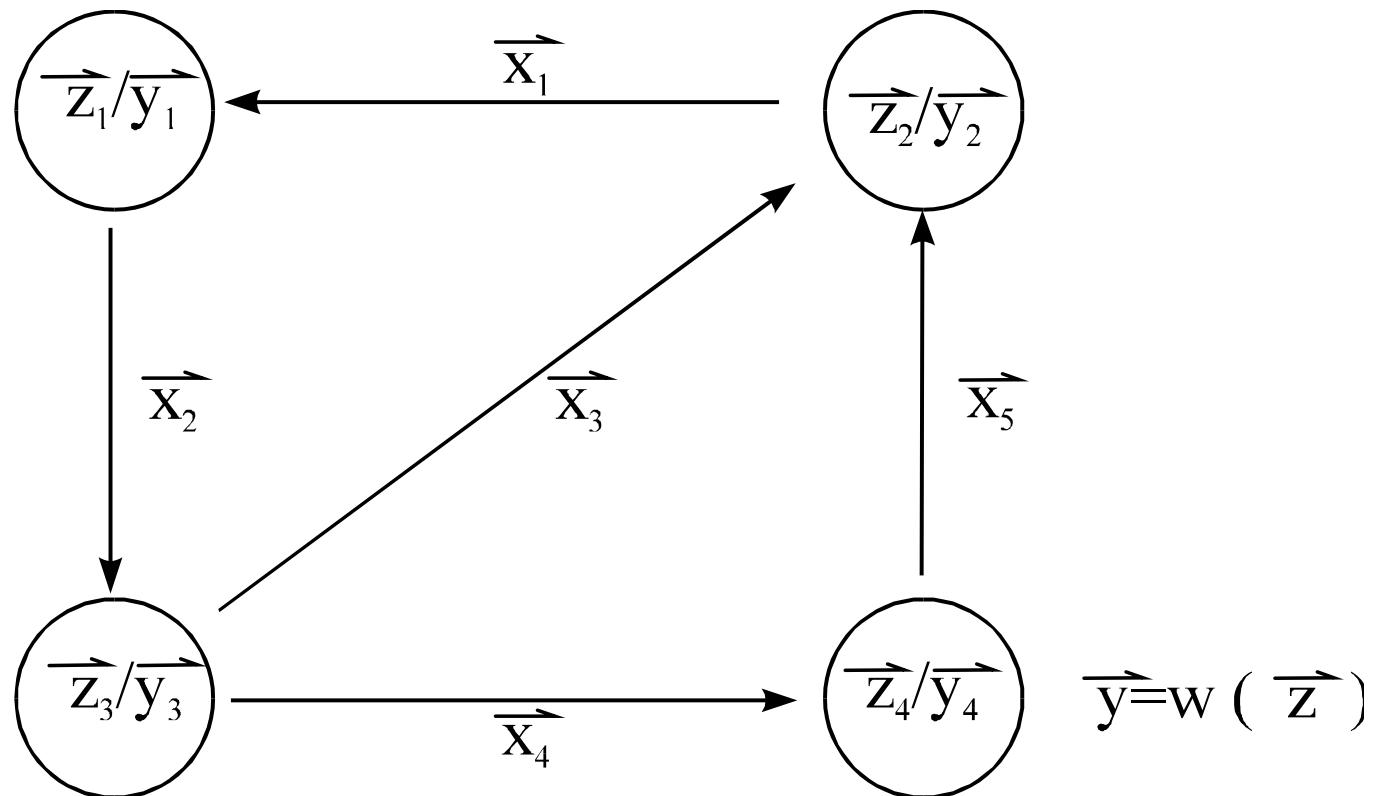
- Graphen dienen der formalen Darstellung von Schaltwerken
- Graphen haben Knoten und Kanten

Zustandsgraphen von Schaltwerken (auch Schaltwerksgraphen):

- Knoten repräsentieren die stabilen Zustände
- Kanten die Übergänge (Transitionen)
- Beispiel Moore- oder Mealy-Automaten

Boolesche Algebra

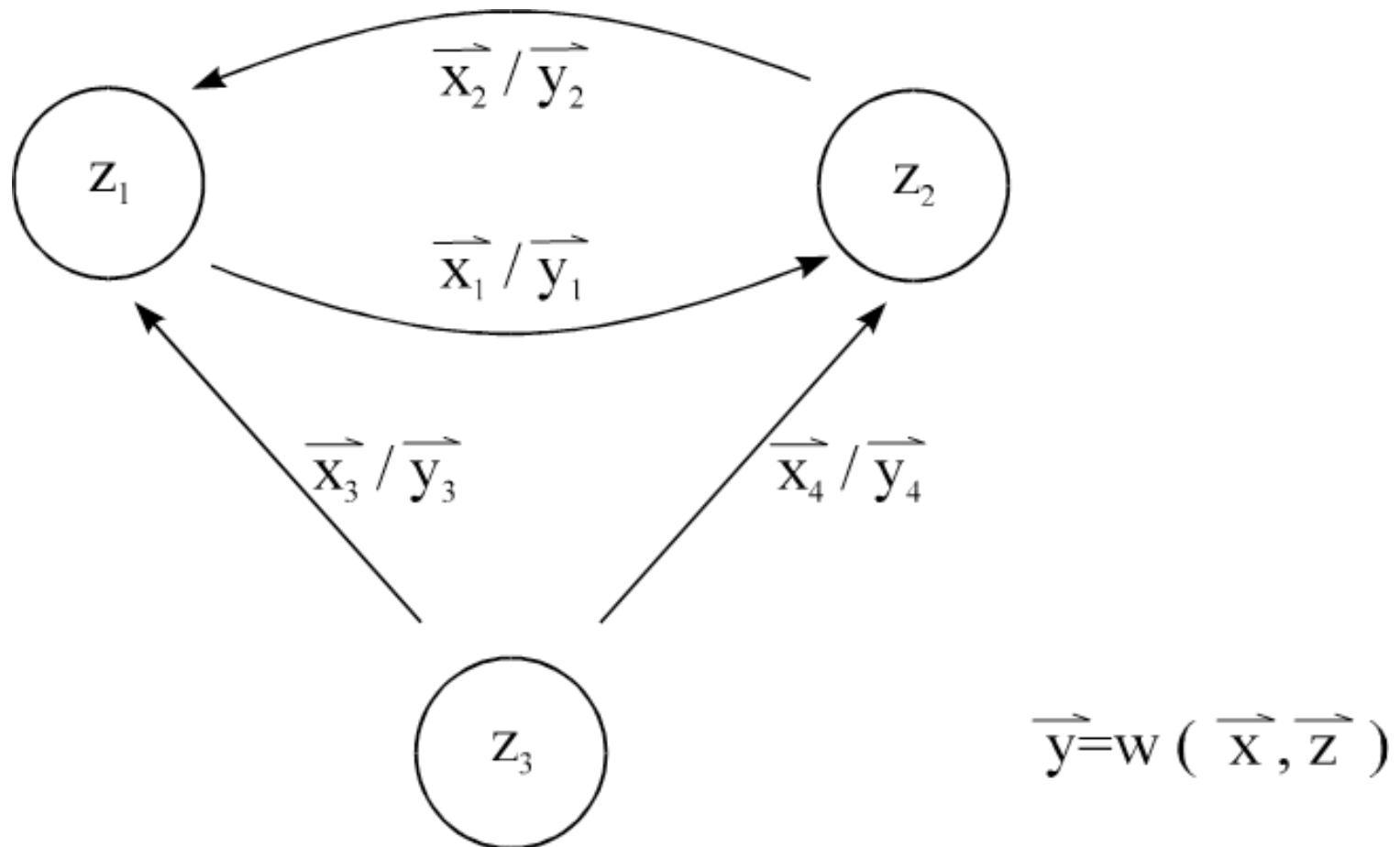
- Moore Automat -



Ausgabewert Y ist direkt einem Zustand Z zugeordnet

Boolesche Algebra

- Mealy Automat -

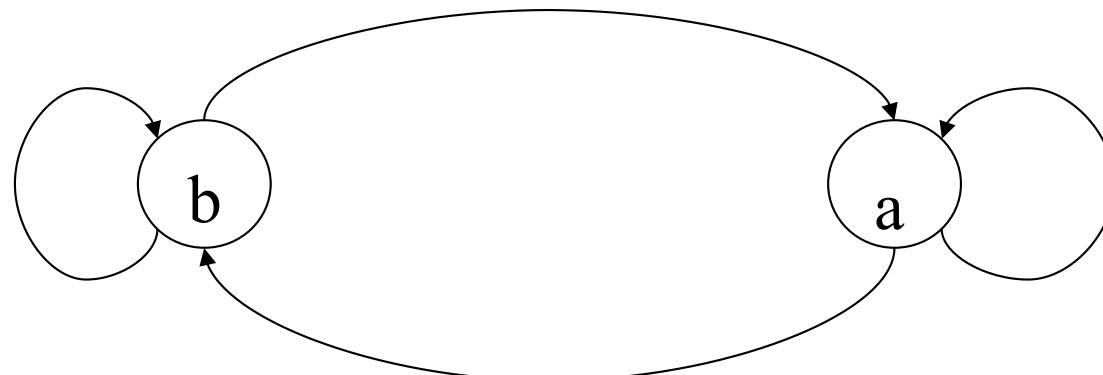


Ausgabewert Y ist abhängig vom Eingang X und dem jeweiligen Zustand Z

Boolesche Algebra

- Boolesche Funktionen -

- Automaten = Maschinen, Bankautomaten, Getränkeautomaten
- Automaten = synchrone Schaltwerke
- => Beschreibung und Verarbeitung formaler Sprachen
- Endliche Automaten => Mengen E A S sind endlich
- Deterministische Automaten => es gibt zu jedem Zustand einen (nicht notwendig verschiedenen) Nachfolgezustand
- Beispiel: Getränkeautomat



Boolesche Algebra

- Endliche Automaten - Beispiel Getränkeautomat -

Endliche Automaten: die Menge der Eingaben, Ausgaben und Zustände endlich, nur ein Anfangszustand s_0

Eingabemenge $E = \{X,K,L,R\}$ (auch Alphabet)

wobei

X = Geldbetrag einwerfen

C= Auswahltaste „Cola“

L = Auswahltaste „Limo“

R = Auswahltaste „Rückgabe“

Zustandsmenge $Z = \{a,b\}$

a = Geldbetrag ausreichend

b = Bereit

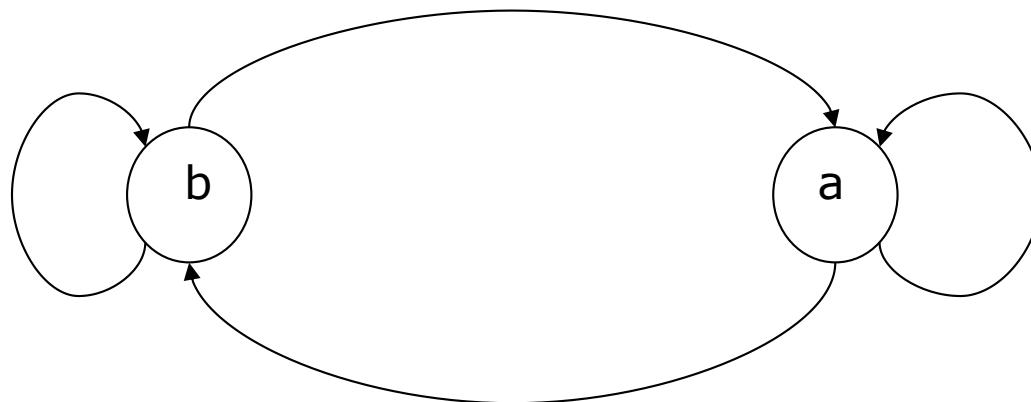
Ausgabemenge $A = \{c, l, x, s\}$

c = Ausgabe Getränk cola

l = Ausgabe Getränk limo

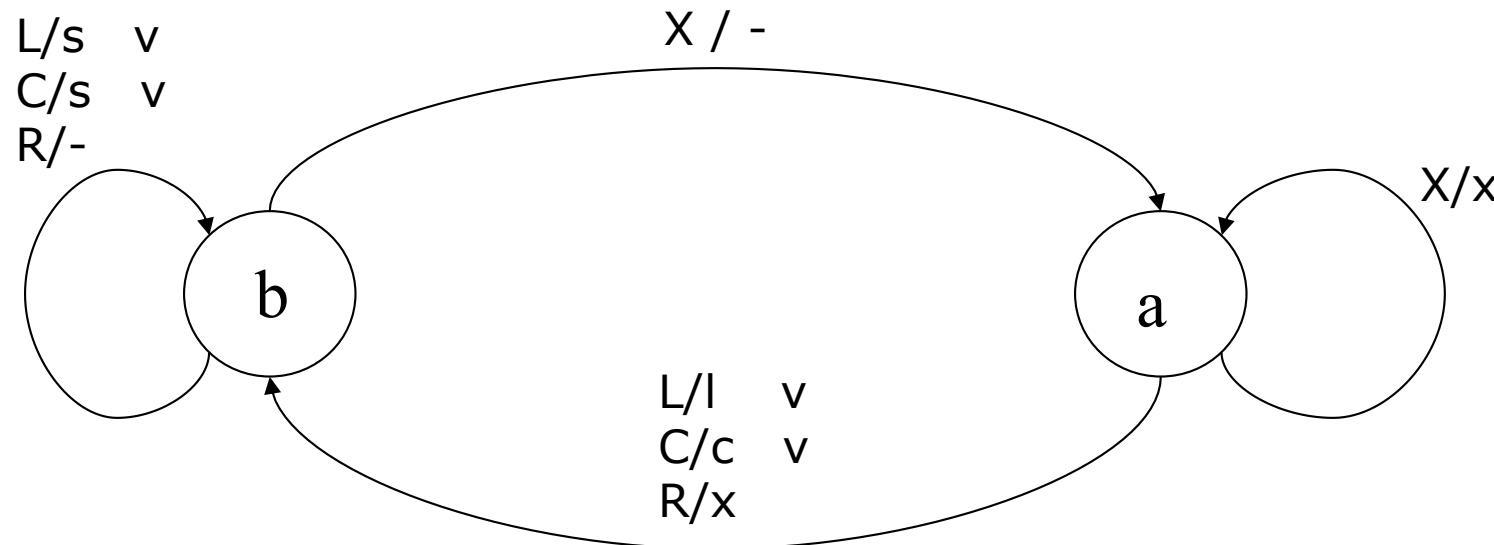
x = Ausgabe Geltbetrag x

s = Ausgabe Signal



Boolesche Algebra

- Endliche Automaten - Beispiel Getränkeautomat -



Endliche Automaten: die Menge der Eingaben, Ausgaben und Zustände endlich, nur ein Anfangszustand s_0

$$E = \{X, K, L, R\}$$

X = Geldbetrag einwerfen

C = Auswahltaste „Cola“

L = Auswahltaste Limo

R = Auswahltaste Rückgabe

Zustandsmenge
a Geldbetrag ausreichend
b Bereit
Ausgabemenge A
c, l, x, s,

Boolesche Algebra

- Schaltwerksanalyse, Synthese -

Aufgabe		Ergebnis	Ziel
Analyse	Formale Beschreibung	Schaltwerksgraph	Anforderung
	Formale Beschreibung	Allgemeine Zustands- / Anregungsmatrix (ZA-Matrix)	Abbildung des Graphen
Synthese	Minimierung (z.B. nach dem Verfahren von Huffmann/ Mealy)	Minimale allg. ZA-Matrix	Stabilitätsuntersuchung
	Codierung der Zustände	Binäre ZA-Matrix	
	Auswahl der Speicherelemente	Anregungsmatrix	Auswahl geeigneter Speicherelemente
	Minimierung der Anregungsfunktion	Minimale Anregungsfunktion	Synthese des Schaltnetzes
	Minimierung der Ausgangsfunktion	Minimale Ausgangsfunktion	
	Zusammensetzen von Speicherelementen, Anregungs- und Ausgangsfunktion	Schaltbild	Ziel der Schaltwerkssynthese

Analyse = Systematische Untersuchung

Synthese = Vereinigung von mehreren Teilen zu einer neuen Einheit

Boolesche Algebra

- Schaltwerkanalyse – Stabilität -

Allgemein gilt: Ein Schaltwerk ist stabil, wenn:

- die Reihe der Folgezustände, die das Schaltwerk für jeden der Eingangsvektoren durchläuft, nicht zyklisch ist, sondern in einem für diesen Eingangsvektor stabilen Zustand endet.
- die Kodierung der Zustände so gewählt werden kann, dass sich beim Übergang der Zustandsvariablen nur ein Bit ändert. Ist dies nicht möglich, kann das Schaltwerk wegen unterschiedlicher Gatterlaufzeiten in fehlerhafte Zustände geraten.

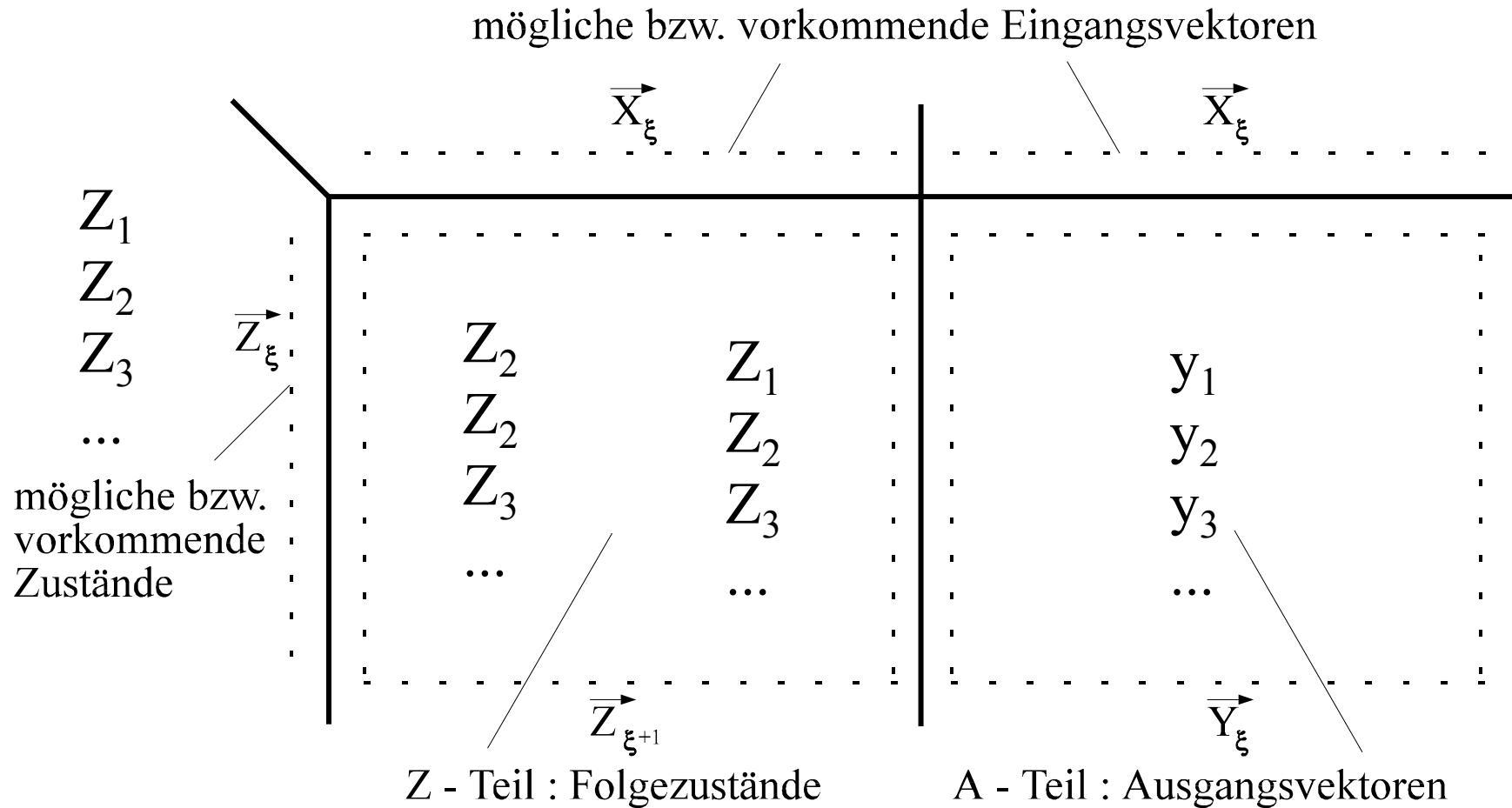
Treten Stabilitätsprobleme auf, können diese durch einen Systemtakt, der einen Zustandsübergang nur zu bestimmten Zeitpunkten ermöglicht, umgangen werden.

Ein solches Schaltwerk bezeichnet man dann als synchrone Schaltwerk, da sich die Zustandsvariablen zum gleichen Zeitpunkt ändern. Dies resultiert jedoch in höheren Kosten für das Schaltwerk.

Ist eine Realisierung des Schaltwerks durch Simulation, z.B. in einer speicherprogrammierbaren Steuerung, vorgesehen, kann man meist von einem Systemtakt ausgehen, der durch den Simulator gegeben ist.

Boolesche Algebra

- Zustands-Anregungs-Matrix (ZA-Matrix) -



Boolesche Algebra

- Umwandlung der Automaten -

Überführung

Mealy in Moore:

- Alle Zustände mit den jeweiligen Ausgaben zeichnen
- Können zu neuen Zustände des Moore Automaten zusammengefasst werden

Moore in Mealy:

- Alle kombinierten Zustände des Moore Automaten müssen wieder zerteilt werden in Einzelzustände des Mealy-Automaten
- Ausgaben auf die Kanten transferieren

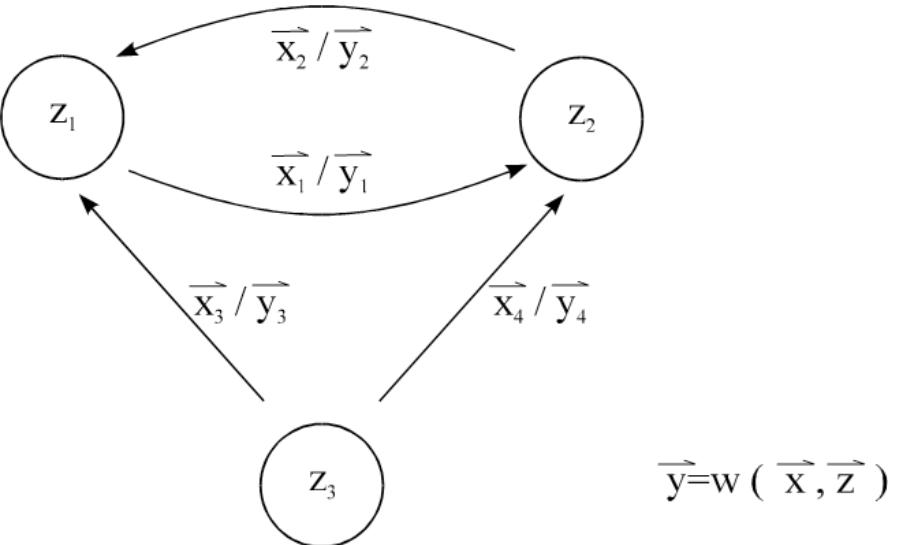
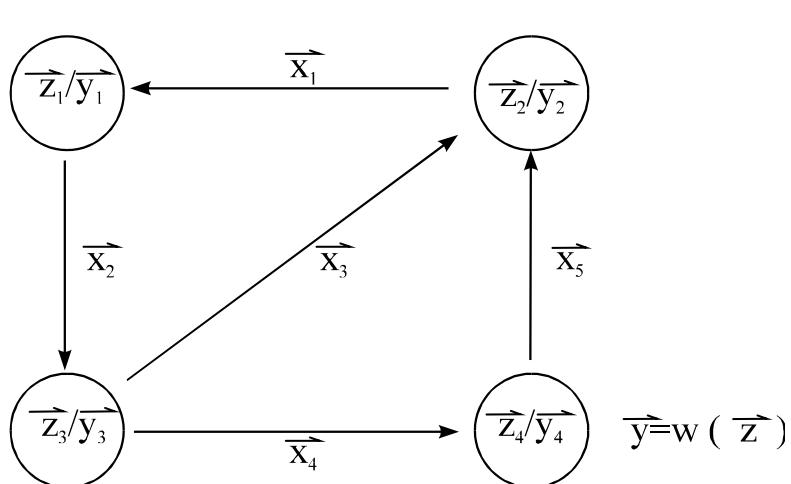
Im allg. ist Mealy leichter zu interpretieren

(Anzahl der Zustände im Moore-Automaten ist größer oder gleich denen im Mealy-Automaten)

s. Rembold S 120

Boolesche Algebra

- Umwandlung der Automaten -



Anzahl der Zustände im Moore-Automaten ist größer oder gleich denen im Mealy-Automaten

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen**
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 05 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



Bremer Institut für
Messtechnik, Automatisierung
und Qualitätswissenschaft

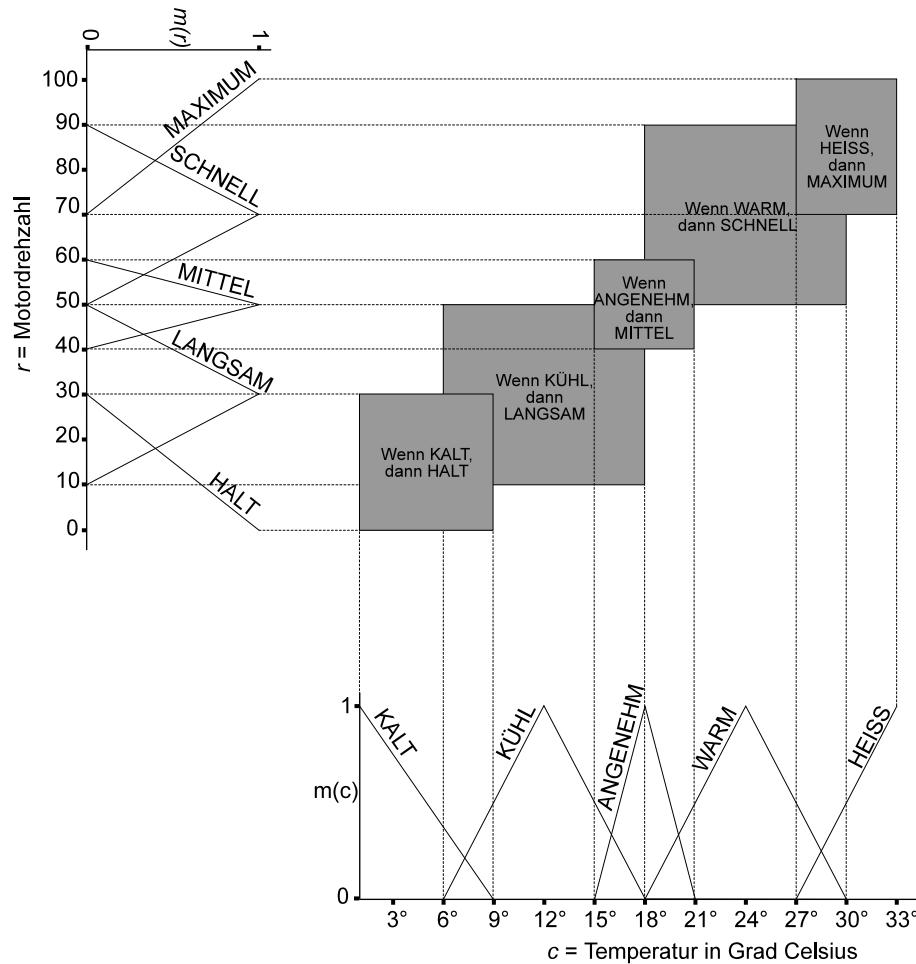
Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 **Fuzzy Logic**
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Fuzzy Logic

- Einführung -

Fuzzy Logic auch ***Unscharfe Logik***



Fuzzy Logic

- Einführung – Definition -

Der Begriff "fuzzy logic", zu deutsch unscharfe Logik, wurde von Lotfi Asker Zadeh geprägt, der in seinem Papier "Fuzzy Sets" (Zadeh 1965, S. 338 ff.) im Jahre 1965 eine Theorie zur Beschreibung und Verknüpfung von unscharfen Mengen vorstellt.

Unscharfe Logik steht für

- Unbestimmtheit,
- für Wagheit,
- Mehrwertigkeit, dafür, dass alles nur zu einem gewissen Grad zutrifft.

Damit stellt die unscharfe Logik eine Erweiterung der klassischen oder binären Logik dar,

mit: ja oder nein, schwarz oder weiß, 1 oder 0, {t, f}

Mit der unscharfen Logik sind qualitative Beschreibungen von Objekten möglich.

z. B. sehr groß oder klein, mittel, für die Größe eines Menschen

Fuzzy Logic

- Einführung – Anwendungen -

Unscharfe Logik wird heute vor allem in der

- Regelungstechnik (nichtlineare Systeme) eingesetzt und eher selten in
- Entscheidungsfindungsprozessen (z.B. technische Diagnose)

Im Bereich der Regelungstechnik werden die Steuerungen, die auf unscharfe Logik basieren, Fuzzy-Steuerungen oder unscharfe Steuerungen genannt.

Unscharfe Steuerungen werden in

- Klimaanlagen,
- Antiblockiersystemen,
- Kopiermaschinen,
- Fernsehern,
- Staubsaugern,
- Camcordern
- Waschmaschinen u.ä. verwendet
- allg. in SPS Systemen

Fuzzy Logic

- Einführung / Historie -

Besonders in Japan wurden in den 1990 Jahren unscharfe Steuerungen in einer Vielzahl von Anwendungen entwickelt und eingesetzt.

- Vor allem lassen sich mit unscharfen Steuerungen **Prozesse regeln**, die sich bislang **nicht oder nur unbefriedigend** mit herkömmlicher Regelungs-technik steuern ließen.

Die Benutzung von Alltagswissen und umgangssprachlichen Formulierungen, wie

"Wenn es regnet, dann spanne den Schirm auf"

oder

"Wenn es dunkel ist, dann schalte das Licht ein",

machen es einfacher, Prozesse zu automatisieren. Dabei bedarf es nicht eines exakten mathematischen Modells des zu regelnden Prozesses.

- Steuerungsvorgänge realisierbar, die von Menschen **nicht mehr geregelt werden können**, etwa das Stabilisieren eines fliegenden Hubschraubers, bei dem ein Rotorblatt fehlt.

Fuzzy Logic

- Theoretischer Hintergrund – Unscharfe Mengen -

Unscharfe Logik befasst sich mit unscharfen Mengen. Jedes Element einer unscharfen Menge gehört zu einem gewissen Grad dazu. Elemente scharfer Mengen (crisp Sets) dagegen zählen entweder zu der Menge oder nicht.

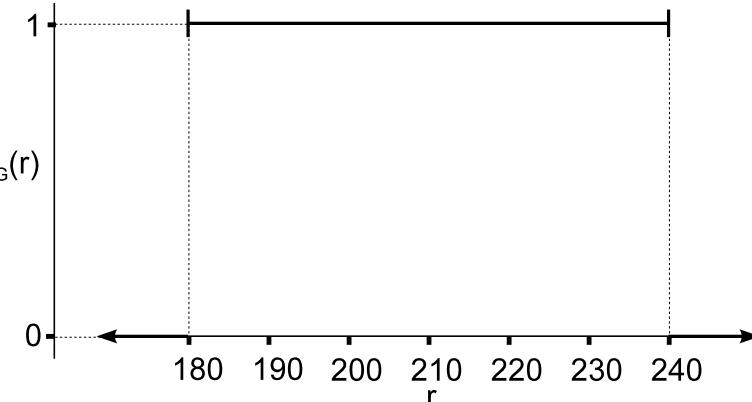
Die Menge G der reellen Zahlen von 180 bis 240 ist scharf und wir schreiben

$$G = \{l \in \mathbb{R} \mid 180 \leq l \leq 240\}.$$

Ebenso kann G durch seine charakteristische zweiwertige Zugehörigkeitsfunktion m_G beschrieben werden:

$$\begin{aligned} m_G(l) &= 1; \quad 180 \leq l \leq 240 \\ &0; \text{ sonst.} \end{aligned}$$

Den Graph von m_G zeigt Bild 1. Jede reelle Zahl l ist entweder in G oder nicht. Weil m_G alle reellen Zahlen $l \in \mathbb{R}$ auf zwei Punkte $\{0, 1\}$ abbildet, zählen scharfe Mengen zur binären Logik



Graph einer Zugehörigkeitsfunktion m_G einer scharfen Menge; verändert nach (Bezdek, Pal 1992, S. 2)

Fuzzy Logic

- Theoretischer Hintergrund – Unscharfe Mengen-

In der Logik werden die Werte von m_G Wahrheitswerte genannt in Hinblick auf die Frage:

"Ist l in G ?"

Die Antwort ist ja dann, und nur dann, wenn $m_G(l) = 1$, andernfalls ist die Antwort nein.

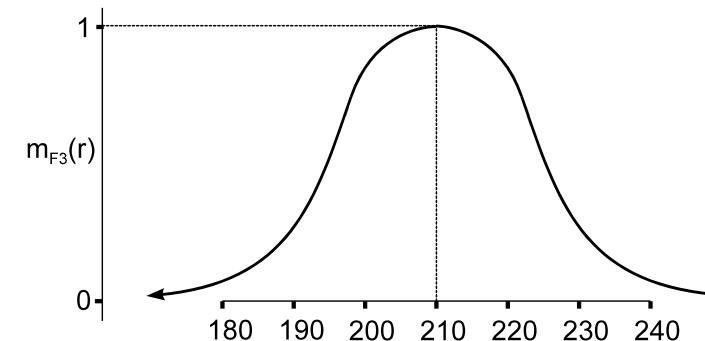
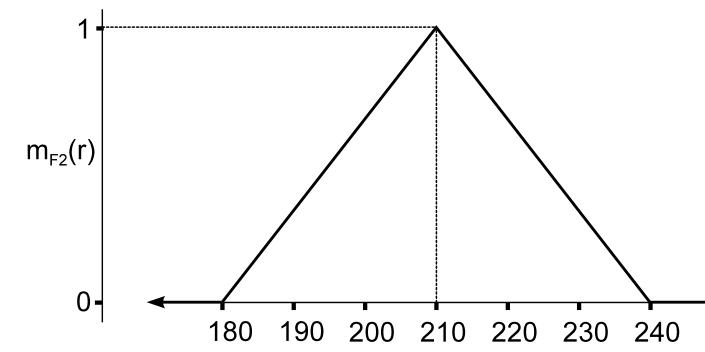
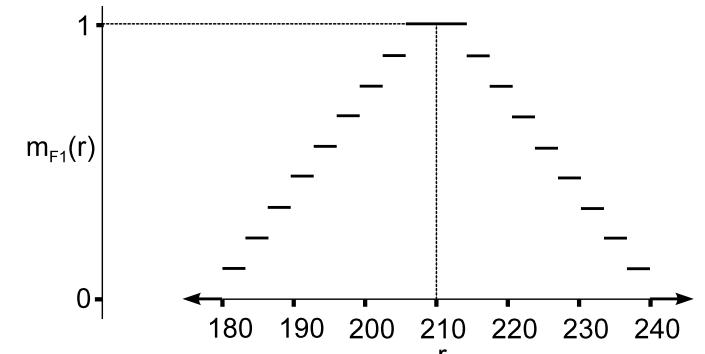
Unscharfe Mengen enthalten Elemente, die unpräzisen Eigenschaften zu einem bestimmten Grad entsprechen, beispielsweise die Menge B der Zahlen, die "nahe 210" sind.

B wird durch seine Zugehörigkeitsfunktion m_B bestimmt, die Wahrheitswerte (truth values) auf das gesamte Einheitsintervall $[0, 1]$ abbildet.

Der Wert $m_B(l)$ ist der Grad der Mitgliedschaft von l in B . Dadurch sind alle Schattierungen von grau zwischen schwarz ($=1$) und weiß ($=0$) möglich.

Weil die Eigenschaft "nahe 210" vage ist, gibt es keine eindeutige Zugehörigkeitsfunktion für B . Je nach beabsichtigter Anwendung ließen sich verschiedene Eigenschaften bestimmen, beispielsweise nach (Bezdek, Pal 1992, S. 2):

- (i) Normalität: $m_B(210) = 1$
- (ii) Monotonie: Je näher l an 210 ist, desto näher ist $m_B(l)$ an 1 und umgekehrt
- (iii) Symmetrie: Zahlen, die gleich weit links und rechts von 210 sind, sollten gleiche Zugehörigkeit besitzen.



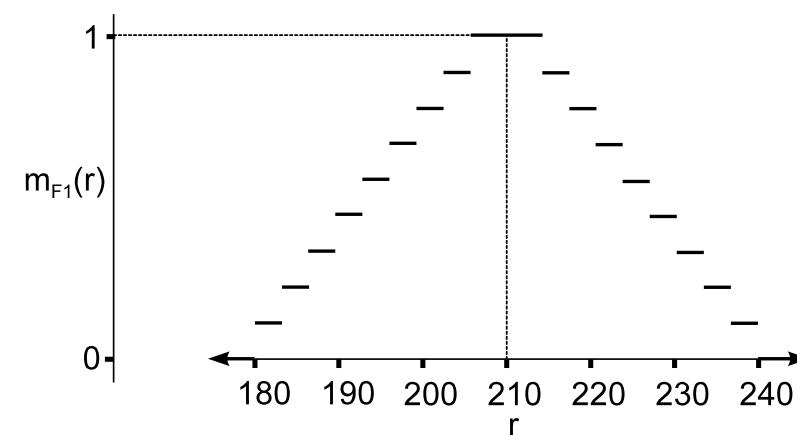
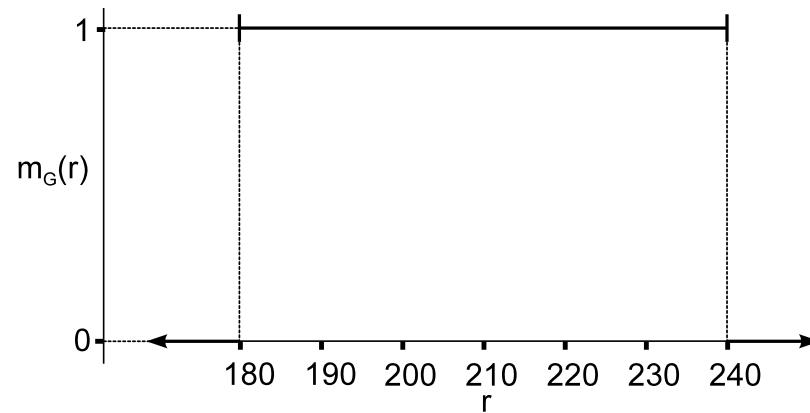
Fuzzy Logic

- Theoretischer Hintergrund – Unscharfe Mengen-

Angenommen, die Zahlen in G und F stehen für die Größen von Basketball spielenden Menschen (b). Wenn jetzt $m_G(b) = 1$, dann ist nur bekannt, dass die Größe zwischen 180 und 240 liegt. Andererseits, wenn $m_B(b) = 0,98$ gilt, dann wissen wir, dass die Größe des basketballspielenden Menschen sehr nahe an 210 ist. Welche der beiden Informationen ist damit sinnvoller bzw. genauer?

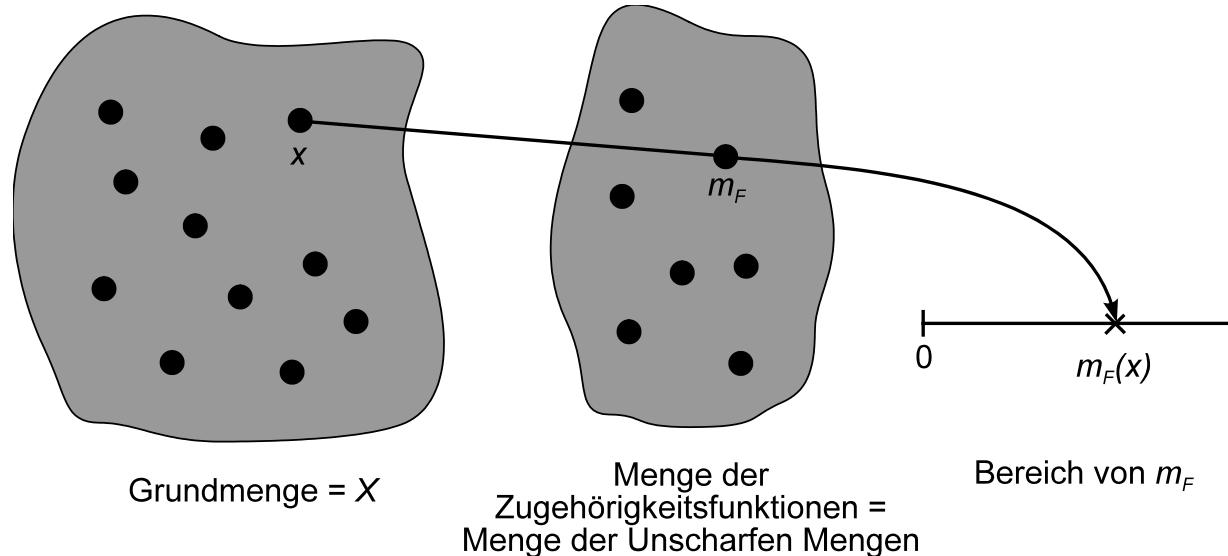
Die Zugehörigkeitsfunktion ist das Grundkonzept in der Theorie von den unscharfen Mengen. Mit ihren Wahrheitswerten wird ausgedrückt, zu welchem Grad Objekte unpräzise definierten Eigenschaften genügen.

Unscharfe Mengen sind immer Funktionen, die aus einer Grundmenge X , beispielsweise die reellen Zahlen, auf das Einheitsintervall $[0, 1]$, dem Bereich der Zugehörigkeitsfunktion von m_F , abbilden (Abb. 19.3)



Fuzzy Logic

- Theoretischer Hintergrund – Unscharfe Mengen-



Unscharfe Mengen sind Funktionen, die aus einer Grundmenge X , beispielsweise die reellen Zahlen, auf das Einheitsintervall $[0, 1]$, dem Bereich der Zugehörigkeitsfunktion von m_F , abbilden.

Fuzzy Logic

- Theoretischer Hintergrund – Unscharfe Mengen-

Es soll gelten: $\mathcal{F}(X) = \text{alle unscharfen Mengen von } X, \text{ d. h.}$

$$m \in \mathcal{F}(X) \Leftrightarrow m : X \rightarrow [0, 1]$$

oder in Worten: *m ist genau dann ein Element der Menge aller unscharfen Mengen* (aller Zugehörigkeitsfunktionen), wenn *m eine Funktion ist, die aus der Grundmenge X auf das Einheitsintervall [0, 1] abbildet.*

Seien nun $m_A, m_B \in \mathcal{F}(X)$, also *zwei unscharfe Mengen (Zugehörigkeitsfunktionen)*, dann soll für jedes x aus der Grundmenge X gelten:

$$(=) \text{ Gleichheit} \quad A = B \Leftrightarrow m_A(x) = m_B(x) \quad (1)$$

In Worten: Zwei unscharfe Mengen A und B sind genau dann gleich, wenn für jedes x ihre Zugehörigkeitsfunktionen den gleichen Wahrheitswert liefern.

$$(\subset) \text{ Enthaltung} \quad A \subset B \Leftrightarrow m_A(x) \leq m_B(x) \quad (2)$$

In Worten: Eine unscharfe Menge A ist enthalten in oder Teilmenge von einer unscharfen Menge B genau dann, wenn für jedes x der Wahrheitswert der Zugehörigkeitsfunktion von A kleiner oder gleich dem Wahrheitswert der Zugehörigkeitsfunktion von B ist.

Fuzzy Logic

- Theoretischer Hintergrund – Unscharfe Mengen -

$$(-) \text{ Komplement} \quad m_{\bar{A}}(x) = 1 - m_A(x) \quad (3)$$

In Worten: Das Komplement einer unscharfen Menge A ist die Menge, für die jeder Wahrheitswert ihrer Zugehörigkeitsfunktion für jedes x gerade die Differenz von 1 und dem Wahrheitswert der Zugehörigkeitsfunktion von A und dem selben x ist.

$$(\cap) \text{ Durchschnitt } m_{A \cap B}(x) = \min \{m_A(x), m_B(x)\} \quad (4)$$

In Worten: Der Durchschnitt zweier unscharfer Mengen A und B ist für jedes x das jeweilige Minimum der zwei Wahrheitswerte der Zugehörigkeitsfunktionen von A und B .

$$(\cup) \text{ Vereinigung } m_{A \cup B}(x) = \max \{m_A(x), m_B(x)\} \quad (5)$$

In Worten: Die Vereinigung zweier unscharfer Mengen A und B ist für jedes x das jeweilige Maximum der zwei Wahrheitswerte der Zugehörigkeitsfunktionen von A und B .

Fuzzy Logic

- Beispiel 1 -

Zugehörigkeitsfunktion zu Beispiel 1; verändert nach (Bezdek, Pal 1992, S. 3)

Beispiel 1

Sei M = Menge der Menschen
und $m \in M$ ein Mensch aus dieser Menge.
Weiter sei

$x = g(m)$ = die Größe eines Menschen
und $X = [0,300]$ = alle möglichen Größen der Menschen in cm.

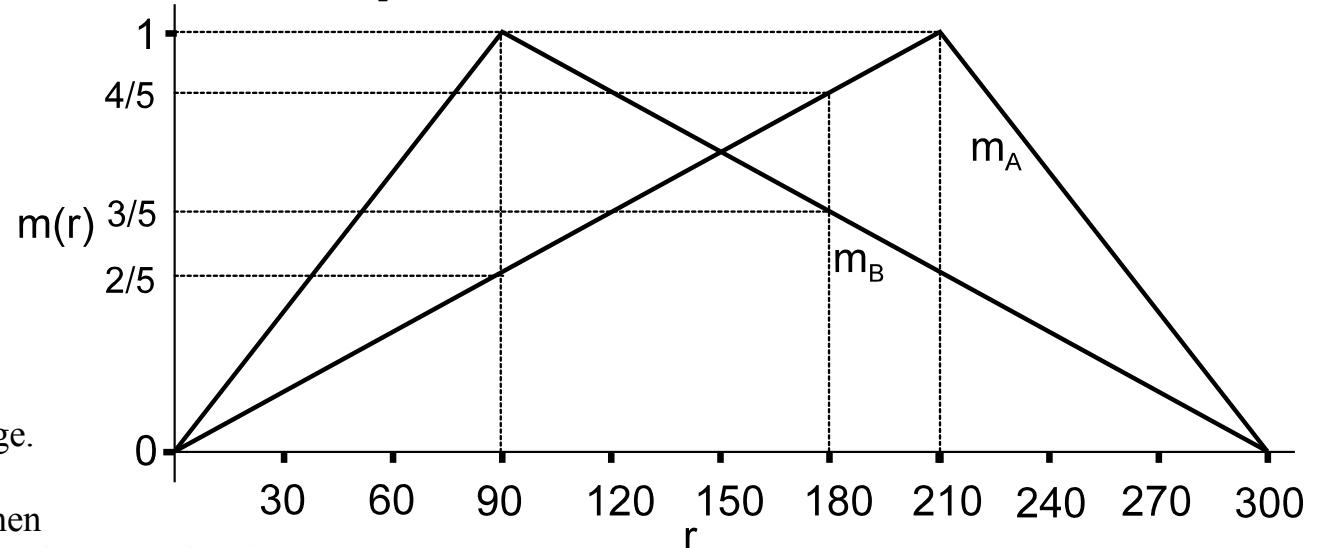


Bild 19.4 zeigt die Zugehörigkeitsfunktionen der unscharfen Mengen m_A , der Menschen, die "nahe 210" cm groß sind,

und

m_B , der Menschen, die "nahe 90" cm groß sind.

Damit gilt:

$$m_A(90) = 2/5$$

Der Grad, zu welchem ein 90 cm Mensch etwa 210 cm groß ist, beträgt also 2/5.

$$m_{\bar{A}}(90) = 1 - (2/5) = 3/5$$

3/5 ist also der Grad, zu welchem ein 90 cm großer Mensch nicht 210 cm groß ist.

$$m_A(180) = 4/5$$

beschreibt den Grad, zu welchem ein 180 cm Mensch etwa 210 cm groß ist.

$$m_B(180) = 3/5$$

ist demnach der Grad, zu welchem ein 180 cm großer Mensch ungefähr 90 cm groß ist.

Fuzzy Logic

- Beispiel 1 -

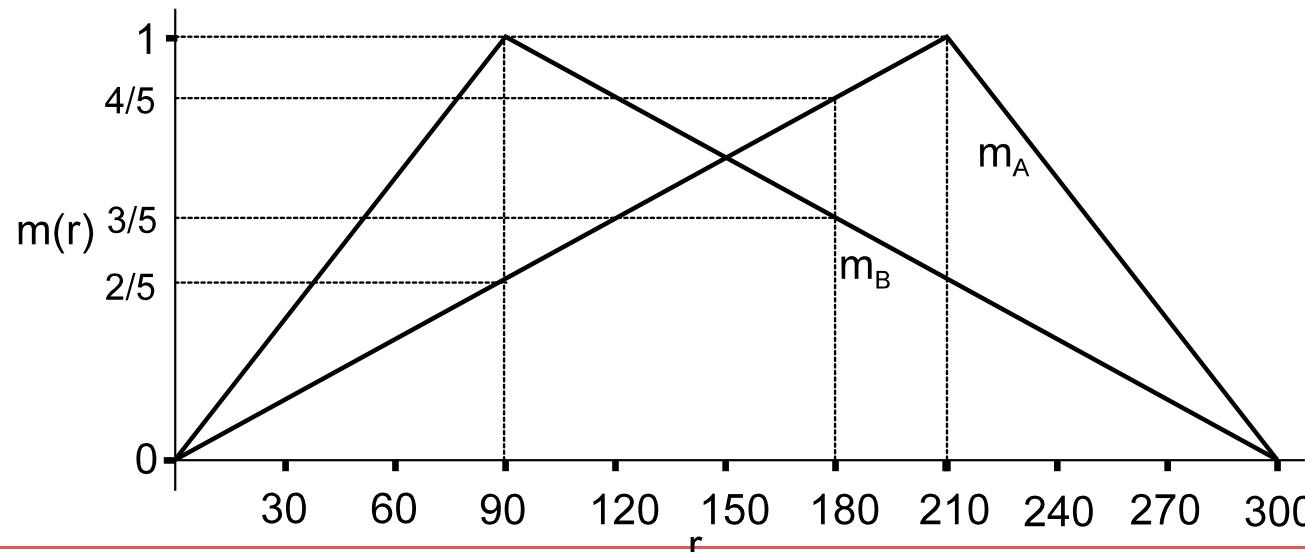
$$\text{Durchschnitt } m_{A \cap B} (180) = \min \{4/5, 3/5\} = 3/5$$

ist demnach der Grad, zu dem er 90 UND 210 cm groß ist.

$$\text{Vereinigung } m_{A \cup B} (180) = \max \{4/5, 3/5\} = 4/5$$

beschreibt, inwieweit er 90 cm ODER 210 cm groß ist.

Die Gleichungen 2 bis 5 führen zu einigen interessanten Situationen. Dazu betrachte die Zugehörigkeitsfunktion m_N im Bild die die Grundlage für folgendes Beispiel liefert.



Fuzzy Logic

- Beispiel 1 -

Sei $X = [a, b]$ und $m_N(x) = 0.5$ $x \in X$

Mit den Gleichungen (3) bis (5) ergibt sich:

$$m_{\bar{N}}(x) = 1 - (0.5) = 0.5 \quad x \in X$$

$$m_{N \cap \bar{N}}(x) = \min \{0.5, 0.5\} = 0.5 \quad x \in X$$

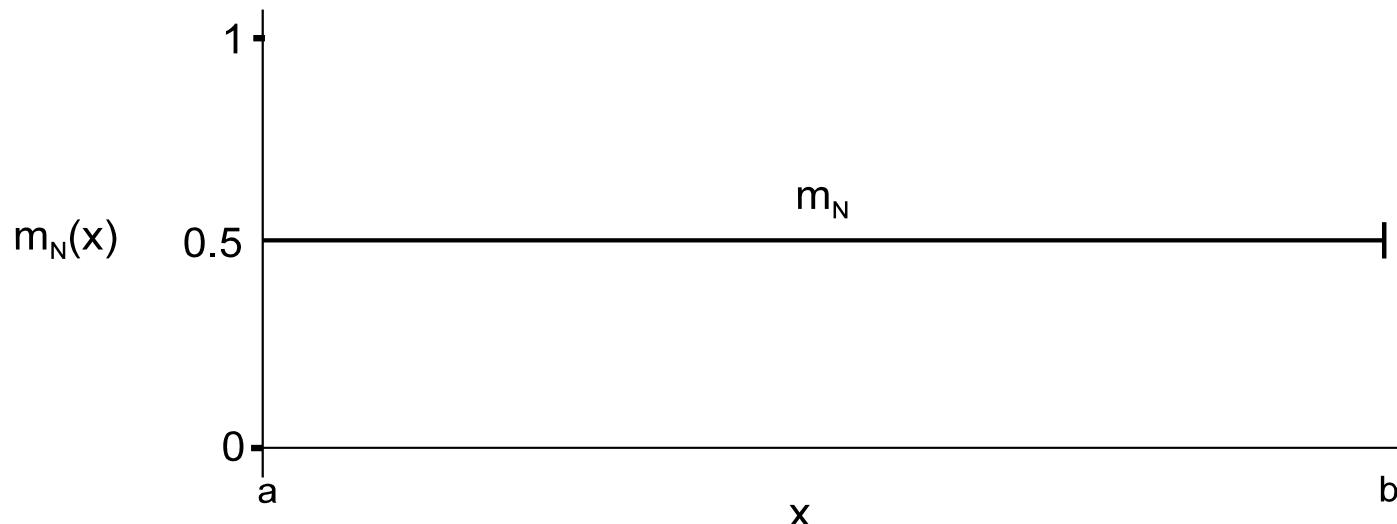
$$m_{N \cup \bar{N}}(x) = \max \{0.5, 0.5\} = 0.5 \quad x \in X$$

Damit gilt für diese unscharfe Menge:

$$N = \bar{N} = N \cap \bar{N} = N \cup \bar{N}.$$

So ist die Vereinigung von N mit seinem Komplement eine Teilmenge von X !

Aber auch $N = \bar{N}$ ist bemerkenswert, denn es ist ein Widerspruch zur klassischen Logik, die auf N ODER \bar{N} basiert und nicht auf N UND \bar{N} .



Fuzzy Logic

- Beispiel 2 -

Stelle Dir ein volles Glas vor. Ist es voll? Die klassische Logik kennt nur ja oder nein. Egal wie voll das Glas ist, wir müssen auf- oder abrunden. Die Antwort lautet also ja.

Mit der unscharfen Logik lautet die Antwort: Das Glas ist zu einem gewissen Grad voll. Wenn es wirklich ganz voll ist, dann ist es zu 100 % voll!

Klassische Logik ist ein Sonderfall der unscharfen Logik:

Wenn das Glas nicht ganz voll ist, dann ist es z. B. zu 93 % voll und wir wissen, dank unscharfer Logik, wie voll das Glas ist, wenigstens ungefähr. Mit unscharfer Logik lässt sich die Realität genauer abbilden.

Nun trinke das Glas aus. Wie viel ist noch im Glas? Die klassische Logik will eine klare Antwort, also: Das Glas ist leer. Wir müssen abrunden. Aber ein kleiner Schluck könnte noch drin sein. Und die unscharfe Antwort: "Es ist zu 2 % voll" ist damit wieder genauer. Was ist aber, wenn das Glas halbvoll ist? Versuche mit der klassischen Logik zu antworten. Ist es voll? Ist es leer? Soll auf- oder abgerundet werden? Es ist keine befriedigende Antwort möglich. Die unscharfe Logik kennt aber die Antwort: Es ist zu 50 % voll (und damit auch zu 50 % leer und zu 50 % nicht voll).

Es gilt nicht:

$N \text{ ODER } \bar{N}$

sondern

$N \text{ UND } \bar{N}$.

Widerspruch-Beispiele: Denke an den Friseur, der auf seinem Schild stehen hat: "Ich rasiere alle Männer dieser Stadt, die sich nicht selbst rasieren". Wer rasiert den Friseur? Oder die Aussage: "Traue mir nicht!" Wenn wir dem Menschen trauen, dann tun wir es laut Aussage nicht und wenn wir ihm nicht trauen, dann tun wir es.

Fuzzy Logic

- Beispielanwendung - Klimaanlage -

X sei die Raumtemperatur in Grad Celsius und Y die Drehzahl des Motors der Klimaanlage

Angenommen, dass bei hoher Drehzahl des Motors die Luft gekühlt wird, dann soll die Drehzahl bei niedriger Temperatur auch niedrig sein und bei hoher Temperatur ebenfalls hoch.

Zweitens gilt es, die unscharfen Mengen für X und Y zu bestimmen. Dazu werden scharfe Werte unscharfen Mengen zugeordnet, was auch Fuzzifizierung genannt wird.

Eine Möglichkeit für X sind die fünf unscharfen Mengen

KALT,
KÜHL,
ANGENEHM,
WARM und
HEISS,

deren Zugehörigkeitsfunktionen graphisch dargestellt sind.

Fuzzy Logic

- Beispielanwendung - Klimaanlage -

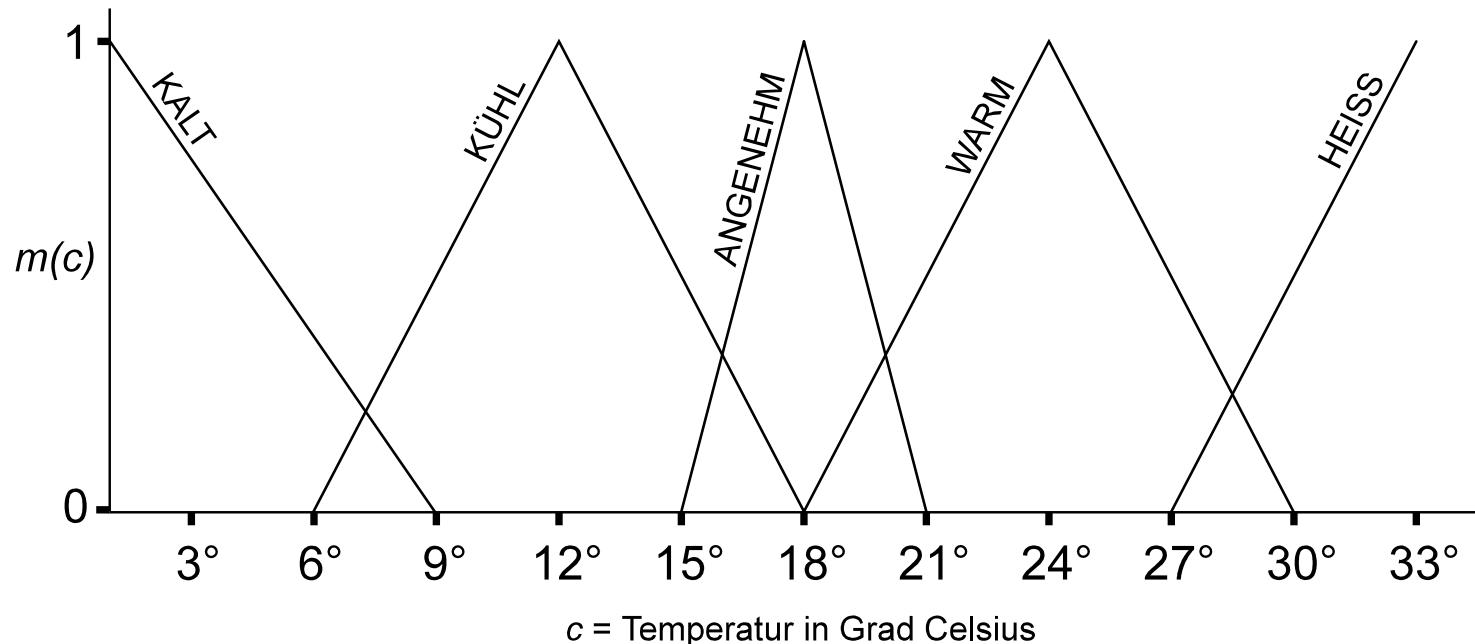


Abb. 19-6 Die Zugehörigkeitsfunktion *KALT*, *KÜHL*, *ANGENEHM*, *WARM* und *HEISS*; verändert nach (Kosko 1993, S. 162)

Statt Dreiecke hätten auch Glockenkurven, Trapezoide oder ähnliches gewählt werden können. In der Praxis sind Dreiecke oftmals ausreichend.

Fuzzy Logic

- Beispielanwendung - Klimaanlage -

5 unscharfen Mengen (*KALT*, *KÜHL*, *ANGENEHM*, *WARM* und *HEISS*)

Die Mengen *KALT* und *HEISS* sind als halbe Dreiecke gezeichnet, da ihre nicht gezeichnete Hälfte nur Temperaturen außerhalb des Bereichs abdecken würde.

Einige Dreiecke, und damit die unscharfen Mengen, sind breiter als andere. Dieses ist der Einfluss gesunden Menschenverstandes in die Regelungstechnik.

Die breitesten unscharfen Mengen sind nicht so entscheidend und eignen sich nur zur groben Kontrolle der Motordrehzahl. Für eine genauere Kontrolle sind "schmalere" unscharfe Mengen notwendig.

Und die Menge *ANGENEHM* steht deshalb nur für einen kleinen Bereich um die eigentlich gewünschte Temperatur von 18 Grad Celsius.

Dadurch soll das Kontrollsysteem der Klimaanlage, unsere unscharfe Steuerung, schnell dafür sorgen, dass die Temperatur in den Bereich kommt, wo es für uns Menschen angenehm ist.

In diesem Bereich allerdings soll die Motordrehzahl mit "Feingefühl" geregelt werden, damit es nicht zu unangenehmen Schwankungen kommt, wie etwa bei herkömmlichen Klimaanlagen, die nicht auf unscharfer Logik basieren.

Fuzzy Logic

- Beispielanwendung - Klimaanlage -

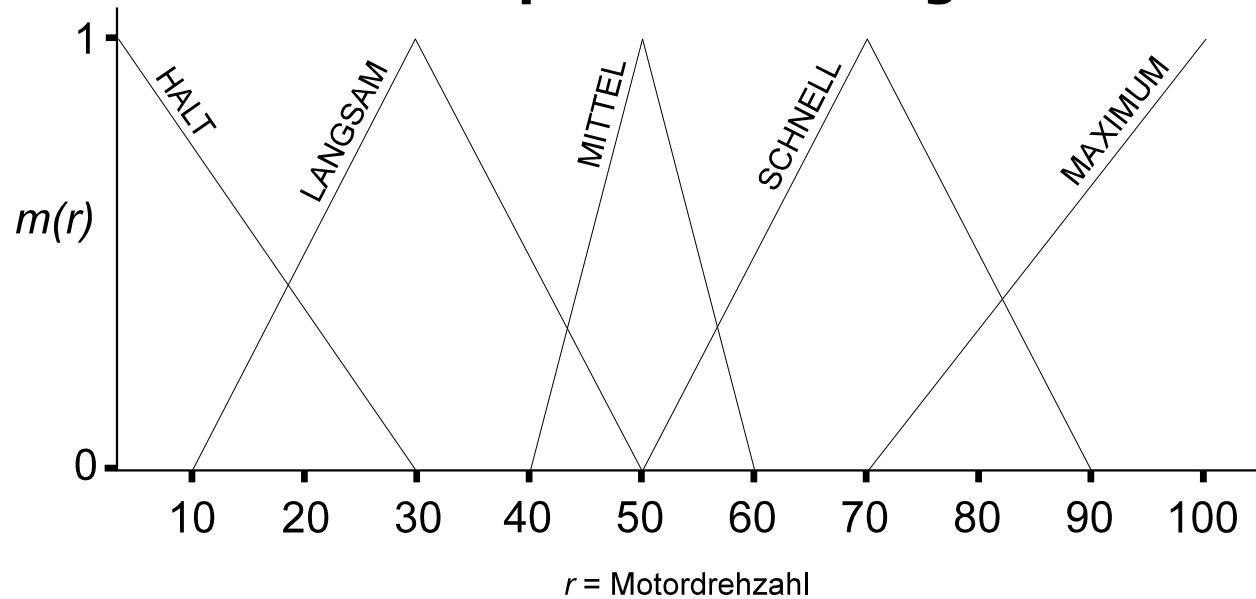


Abb. 19-1 Die Zugehörigkeitsfunktionen *HALT*, *LANGSAM*, *MITTEL*, *SCHNELL* und *MAXIMUM*; verändert nach (Kosko 1993, S. 162)

Auch für die Motordrehzahl werden unscharfe Mengen definiert und wir erhalten
HALT,
LANGSAM,
MITTEL,
SCHNELL und
MAXIMUM

Fuzzy Logic

- Beispielanwendung – Unscharfe Regeln -

Abschließend müssen nun noch die unscharfen Regeln (fuzzy rules) aufgestellt werden.

Es gilt zu verbinden:

die Mengen, die die Motordrehzahl repräsentieren, mit den Mengen, die die Temperaturen repräsentieren.

Jeder Motordrehzahl-Menge wird eine Temperatur-Menge zugewiesen.

Ist die Temperatur im Bereich KALT, dann wollen wir, dass der Motor ausgeschaltet wird, damit es nicht noch kälter wird. Wir wollen ja, dass die Temperatur im angenehmen Bereich bleibt.

Damit haben wir schon die erste Regel:

Wenn X ist KALT, dann Y ist HALT. Der Motor sollte mit mittlerer Drehzahl laufen, wenn die Temperatur angenehm ist:

Wenn X ist ANGENEHM, dann Y ist MITTEL.

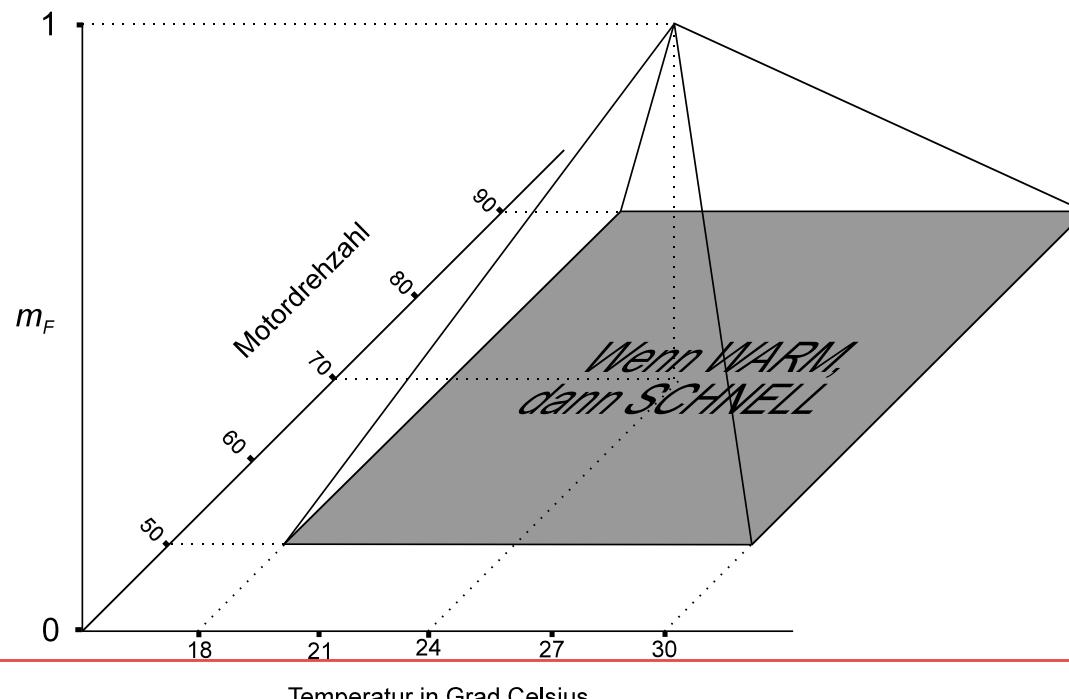
Fuzzy Logic

- Beispielanwendung - Klimaanlage -

So lassen sich fünf Regeln umgangssprachlich bestimmen:

1. Regel: Wenn die Temperatur kalt ist, dann halte den Motor an.
2. Regel: Wenn die Temperatur kühl ist, dann soll der Motor langsam laufen.
3. Regel: Wenn die Temperatur angenehm ist, dann soll der Motor mit mittlerer Drehzahl laufen.
4. Regel: Wenn die Temperatur warm ist, dann soll der Motor schnell laufen.
5. Regel: Wenn die Temperatur heiß ist, dann soll der Motor mit maximaler Drehzahl laufen.

Diese Regeln sind Alltagswissen. Es handelt sich um unscharfe Regeln, denn "kalt" und "schnell" usw. treffen nur zu einem gewissen Grad zu und stehen für unscharfe Mengen. Unscharfe Regeln lassen sich als Patches graphisch darstellen

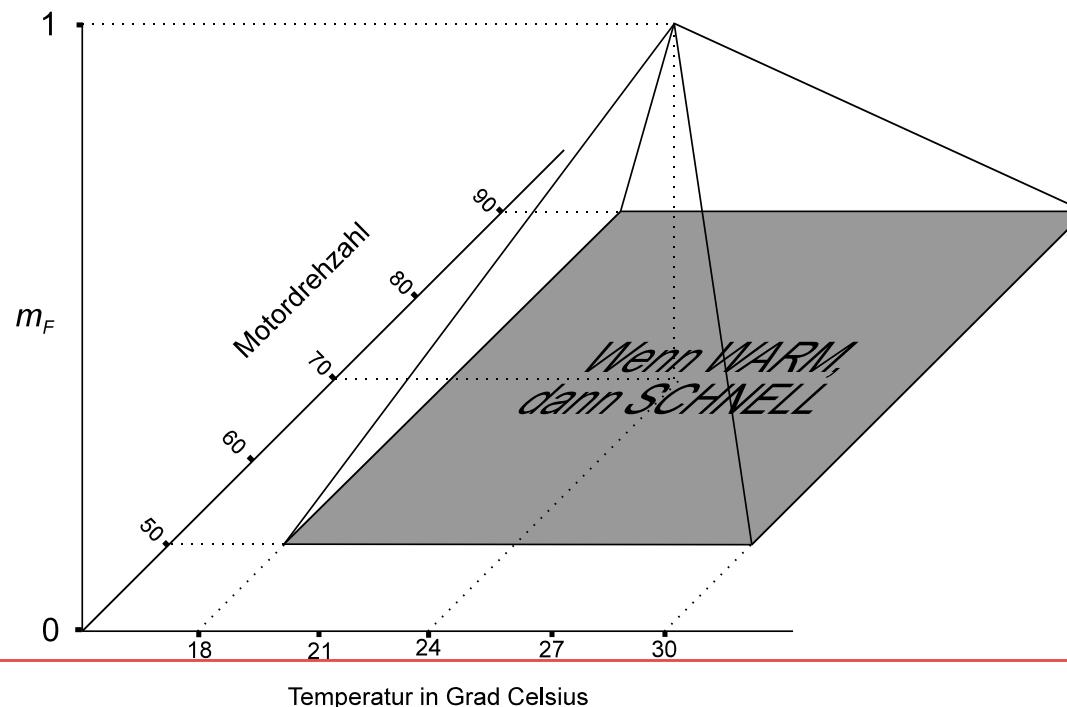


Fuzzy Logic

- Beispielanwendung - Klimaanlage -

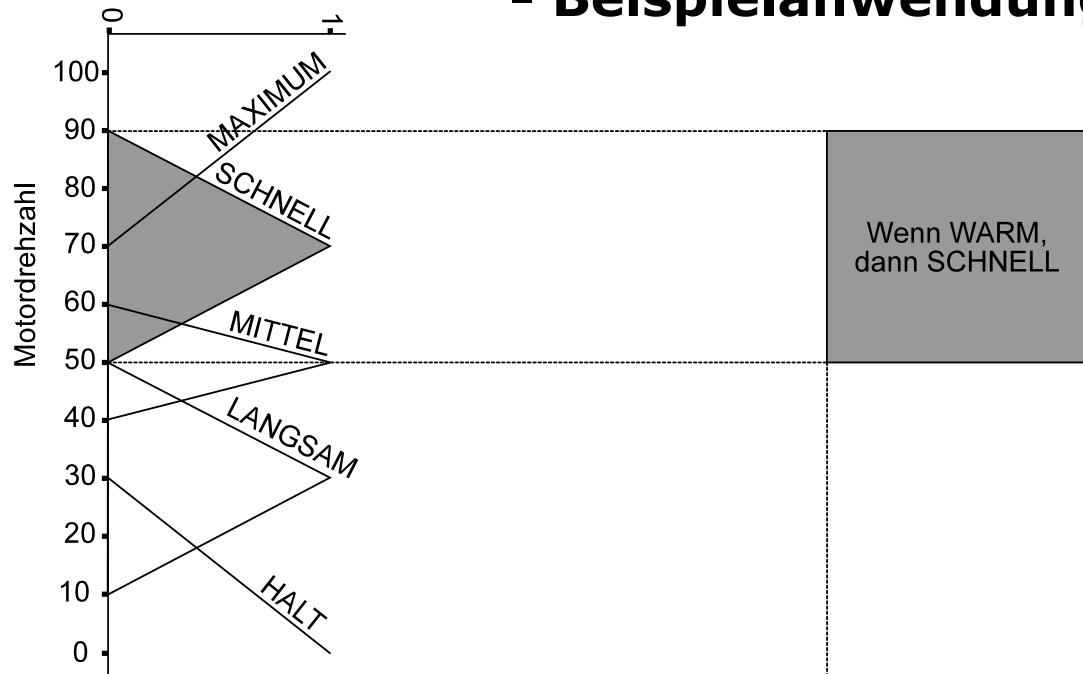
Mathematisch ist das Patch in unserem Fall das Produkt zweier Dreiecke, also eine Pyramide.

Auf dem Pyramidenboden ist **jeder Temperatur eine Motordrehzahl** zugeordnet. Mit der zu jedem Temperatur-Motordrehzahl-Paar gehörenden Höhe der Pyramide lässt sich bestimmen, zu welchem Grad das jeweilige Temperatur-Motordrehzahl-Paar zu dem Regel-Patch gehört. Nur in der Mitte sind es 100 % und außerhalb der Pyramide 0 %.

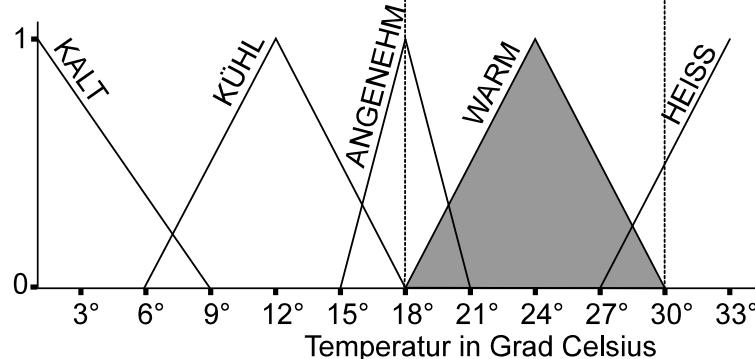


Fuzzy Logic

- Beispielanwendung - Klimaanlage -

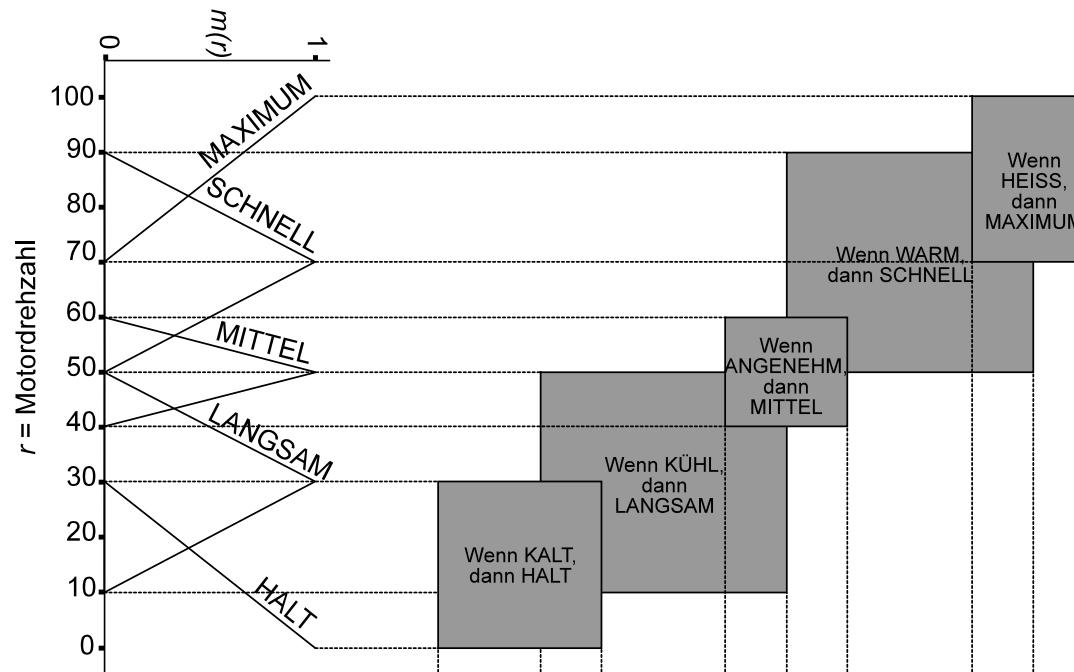


Das Bild ist das „Patch“ der unscharfen Regel „Wenn warm, dann schnell“ als Rechteck gezeichnet, also nur der Pyramidenboden, nämlich das Produkt der Basis des WARM-Dreieckes und der Basis des SCHNELL-Dreieckes. Dies ist ausreichend, um festzulegen, welchem Temperaturbereich welche Motordrehzahlen zugeordnet werden.

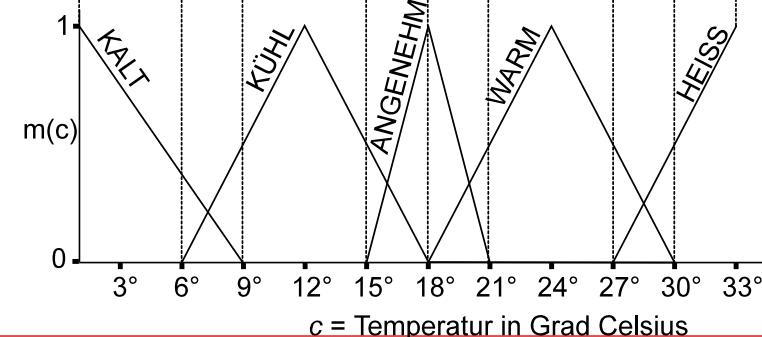


Fuzzy Logic

- Beispielanwendung - Klimaanlage -



Darstellung der Regeln einer unscharfen Steuerung als Rechtecke; verändert nach (Kosko 1993, S. 166)



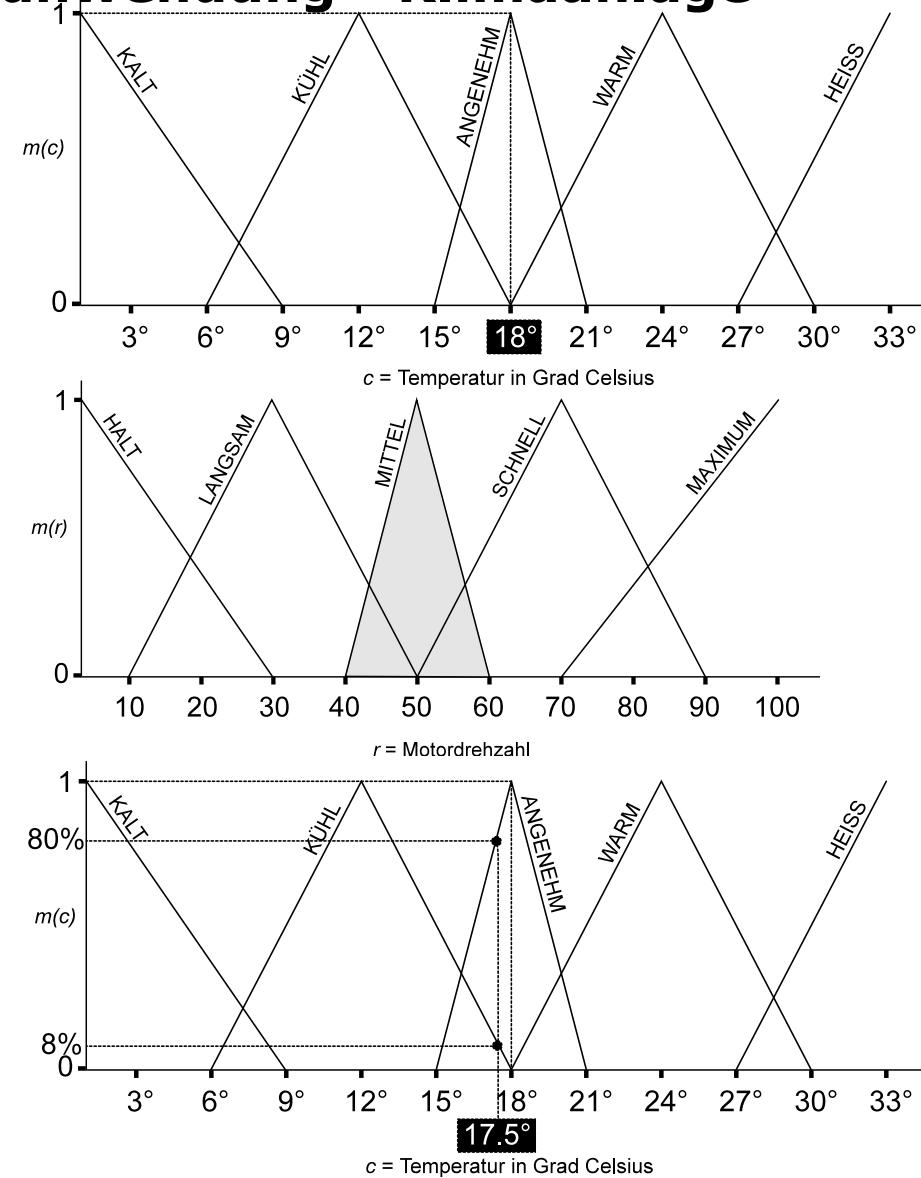
Fuzzy Logic

- Beispielanwendung - Klimaanlage -

Zugehörigkeit von 18° zu den unscharfen Temperatur-Mengen; verändert nach (Kosko 1993, S. 172)

Bei 18° soll die Motorendrehzahl zur unscharfen Menge *MITTEL* gehören; verändert nach (Kosko 1993, S. 172)

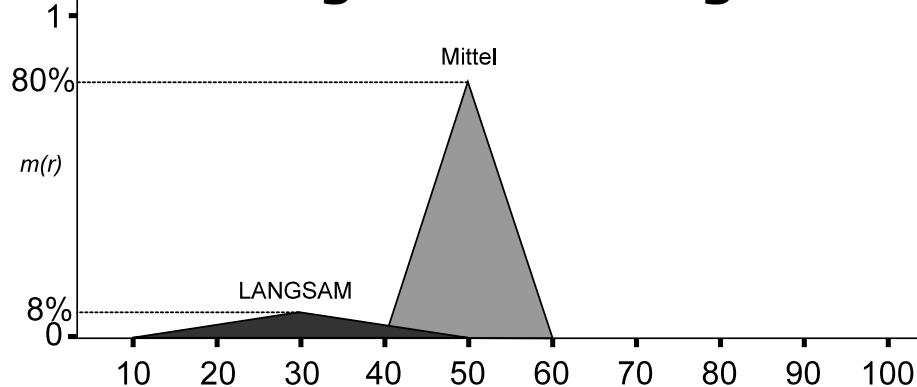
Regel *ANGENEHM* und Regel *KÜHL* feuern; verändert nach (Kosko 1993, S. 173)



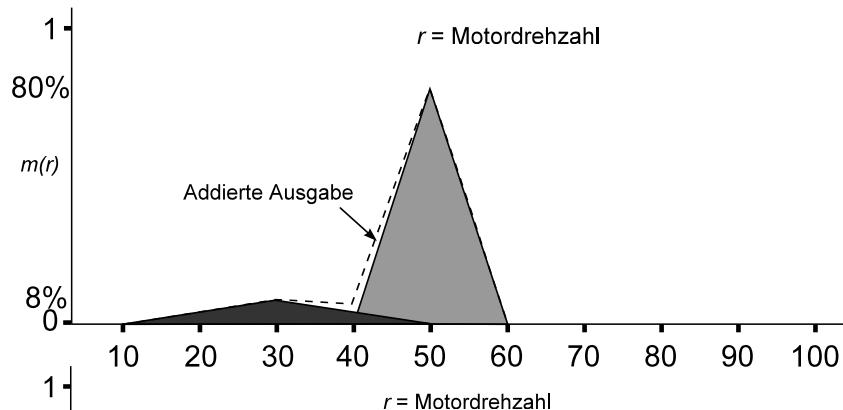
Fuzzy Logic

- Beispielanwendung - Klimaanlage -

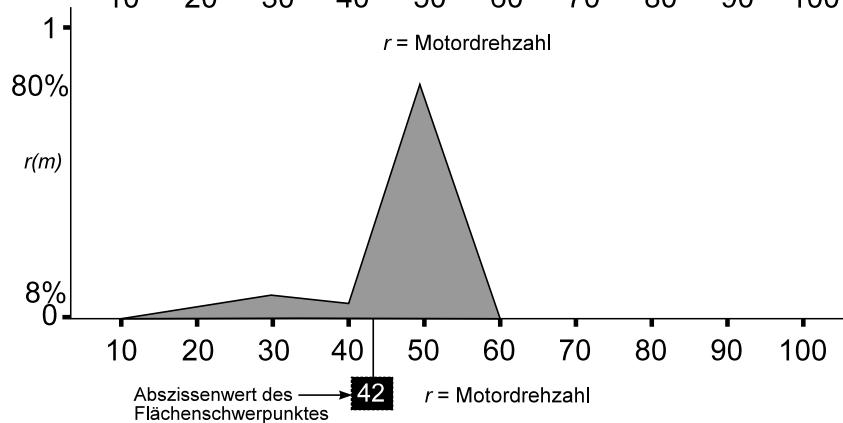
Regel *MITTEL* feuert zu 80 % und Regel *LANGSAM* zu 8 %; verändert nach (Kosko 1993, S. 173)



Addition zweier unscharfer Mengen;
verändert nach (Kosko 1993, Seite 174)



Defuzzifizierung zweier unscharfer
Mengen; verändert nach (Kosko 1993,
S. 174)



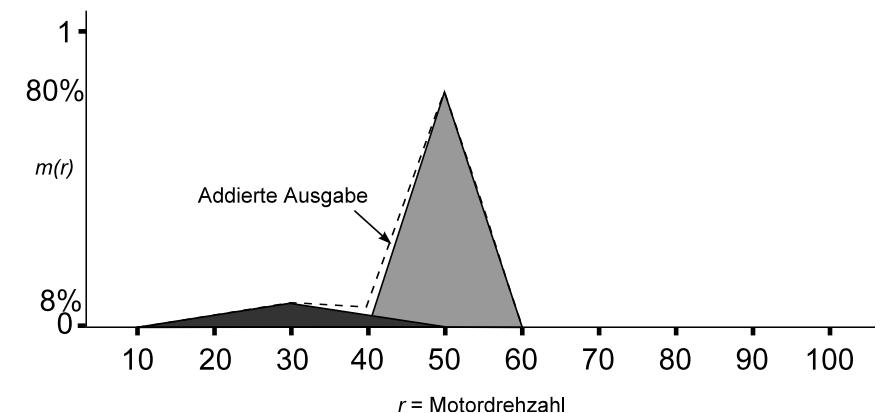
Fuzzy Logic

- Beispielanwendung - Klimaanlage -

Zur Defuzzifizierung der zwei Dreiecke werden diese einfach addiert und der Abszissenwert des Flächenschwerpunktes der Ergebnismenge verwendet.

Diese Defuzzifizierungsmethode wird Schwerpunktmethod (centroid method, center of gravity, center of area) genannt und ist sehr gebräuchlich. Die Berechnung für eine Koordinate x_s des Schwerpunktes $S(x_s, y_s)$ des unter einer Funktion $v = f(x)$ zwischen $x = x_A$ und $x = x_E$ gelegenen Flächenstücks ist aus der Mathematik bekannt (Preuß 1992, S. 183):

$$x_s = \frac{\int_{x_A}^{x_E} xf(x)dx}{\int_{x_A}^{x_E} f(x)dx}$$



Fuzzy Logic

- Beispielanwendung - Klimaanlage -

Wenn die Randkurve der zusammengesetzten Flächen aus Geradenstücken besteht, dann lässt sich die Integration vorab durchführen. Bei m Polygonsegmenten, die durch jeweils 2 Punkte

$$P_k(x_k, y_k)$$

und

$$P_{k+1}(x_{k+1}, y_{k+1})$$

beschrieben sind, lässt sich der Abszissenwert des Schwerpunktes auch folgendermaßen berechnen:

$$x_s = \frac{\sum_{k=1}^m (x_{k+1} - x_k) [(2x_{k+1} + x_k) y_{k+1} + (2x_k + x_{k+1}) y_k]}{3 \sum_{k=1}^m (x_{k+1} - x_k) (y_{k+1} + y_k)}$$

Das ergibt für unsere Steuerung eine Motordrehzahl von 42. Unsere unscharfe Steuerung regelt die Motordrehzahl etwas runter und die Temperatur kann sich so etwas erwärmen.

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 **Fuzzy Logic**
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 06 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



Bremer Institut für
Messtechnik, Automatisierung
und Qualitätswissenschaft

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 **Neuronale Netze**
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Künstliche Neuronale Netze

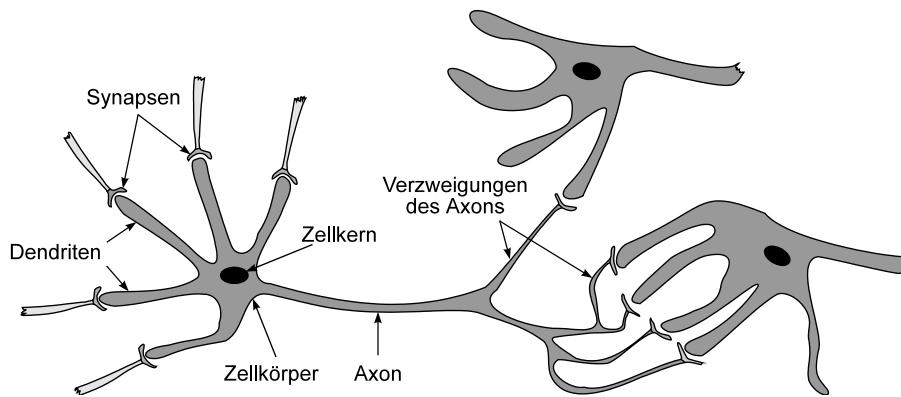
- Historie -

auch „Künstliche Neuronale Netze“ oder (KNN)

Historie - Hintergrund

Grundlagen KNN

Anwendungen



Künstliche Neuronale Netze

- Historie -

Die Realisierung von KNN greift auf Arbeiten von MacCulloch / Pitts um das Jahr 1940 zurück - Beschreibung von logischen Schwellwertelementen

Mit künstlichen neuronalen Netzen wird erstmals versucht, bestimmte Fähigkeiten biologischer Systeme nachzuahmen, das sind Lernfähigkeit und eigenständige Wahl und Gewichtung der neuronalen Verbildungen

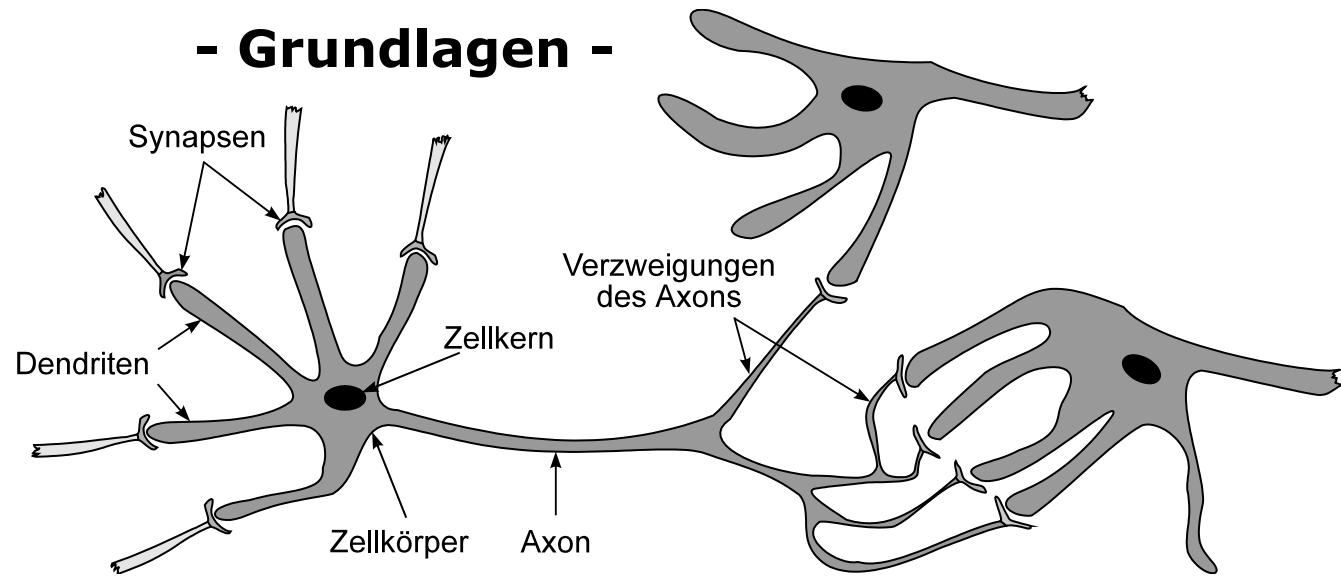


Einige Urgesteine des Fachbereichs der Neuronalen Netze. Von links nach rechts: John von Neumann, Donald O. Hebb, Marvin Minsky, Bernard Widrow, Seymour Papert, Teuvo Kohonen, John Hopfield, "in order of appearance" [Krie05]

Künstliche Neuronale Netze

- Grundlagen -

Schematische Darstellung
einer Nervenzelle (nach Bruns
1990, S. 80)



System aus Nervenzellen (Neuronen) und **Verbindungen** (Synapsen)
Die Struktur eines neuronalen Netzes ist ein **Graph mit Knoten und Kanten**

Lernverfahren bestimmen, wie die am Neuron ankommenden Signale behandelt werden und welche Bedeutung die einzelnen Verbindungen haben

Technisch: System mit Eingängen/ Übertragungsfunktionen und Ausgängen



Künstliche Neuronale Netze

- Grundlagen -

	Gehirn	Computer
Anzahl Recheneinheiten	$\approx 10^{11}$	$\approx 10^9$
Art Recheneinheiten	Neurone	Transistoren
Art der Berechnung	massiv parallel	i.d.R. seriell
Datenspeicherung (assoziativ bzw. adressbasiert): Schaltzeit	$\approx 10^{-3}$ s	$\approx 10^{-9}$ s
Theoretische Schaltvorgänge	$\approx 10^{13}$ 1/s	$\approx 10^{18}$ 1/s
Tatsächliche Schaltvorgänge	$\approx 10^{12}$ 1/s	$\approx 10^{10}$ 1/s

Tabelle 1.1: Der (hinkende) Vergleich zwischen Gehirn und Rechner auf einen Blick. Vorlage: [Zel94] in D. Kriesel, NN,

- **Theorie:** Ein Graph mit Knoten (Neuronen) und Kanten (Synapsen)
- **Synapsen** (Kanten) = gerichtete Signalverbindungen (die Strukturen mit netzartige Verbindungen erlauben)
- **Synonym** zu "künstliches Neuronales Netz" = der Begriff **Konnektionismus**



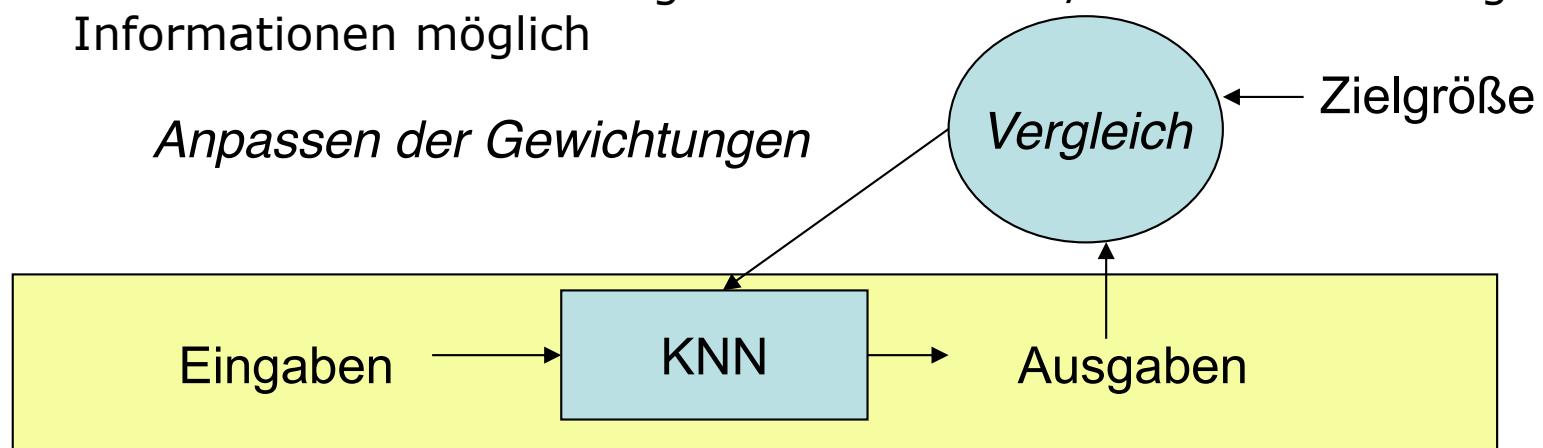
Künstliche Neuronale Netze

- Einsatzgebiete -

Erfolgreiche Einsatzgebiete sind:

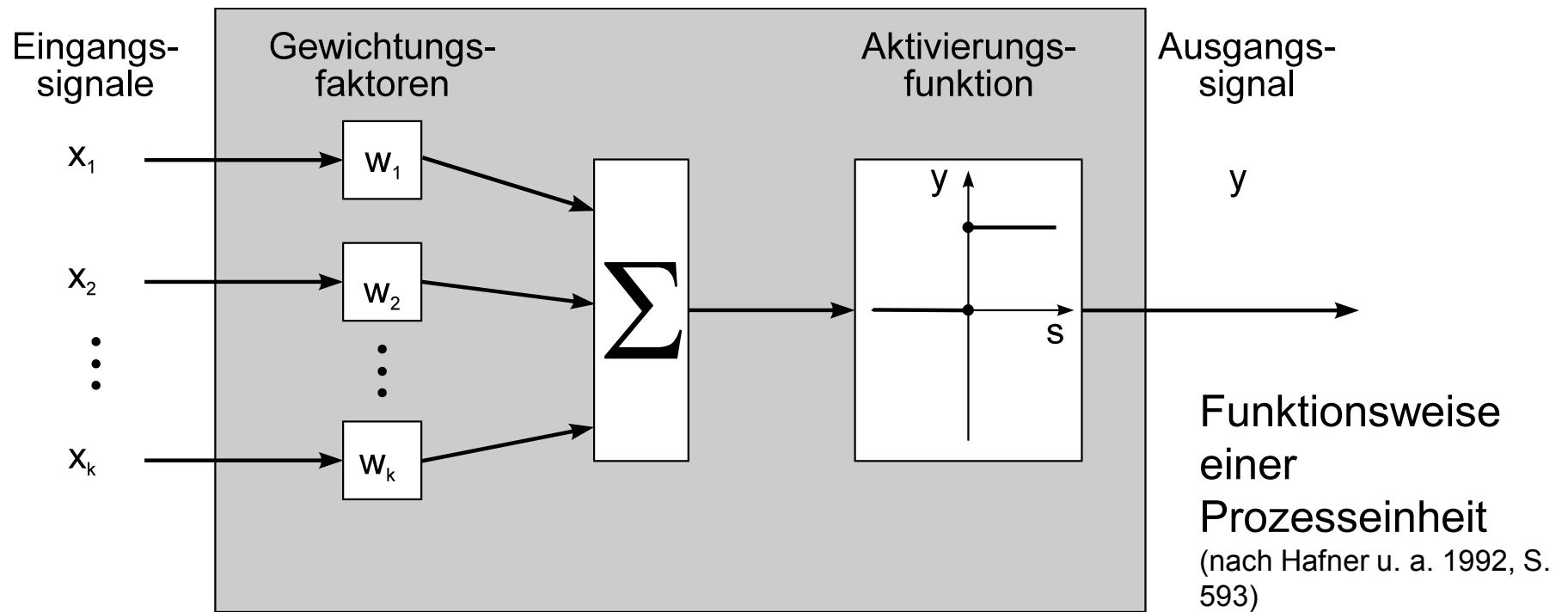
- Modellbildung (trainieren)
- Entscheidungsfindung
- Regelung nichtlinearer Prozesse (Prozess-, Qualitätsregelung)
- Steuerung von Produktionseinrichtungen
- Robotersteuerung (allg. Steuern von Bewegungen)
- Bildverarbeitung (Mustererkennung)
- Technische Diagnose (Mustererkennung, Vorhersage, Prognose)

Vorteil: auch die Verarbeitung fehlerhafter und/oder unvollständiger Informationen möglich

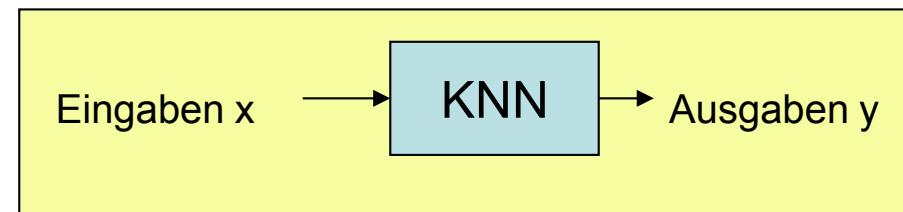


Künstliche Neuronale Netze

- Technische Realisierung -



Eingangssignale x_1, \dots, x_k ,
Gewichtungsfaktoren w_1, \dots, w_k
Addition (oder auch Produktbildung)
Aktivierungsfunktion s
Ausgangssignal y



Künstliche Neuronale Netze

- Historie -

Das Ausgangssignal y einer PE kann nun folgendermaßen berechnet werden:

$$y = f \left(\sum_{k=1}^n (w_k \cdot x_k) \right) \quad [1]$$

Dabei entspricht
 x dem Eingangsvektor,
 w dem Gewichtungsvektor und
 $f(s)$ der Aktivierungsfunktion

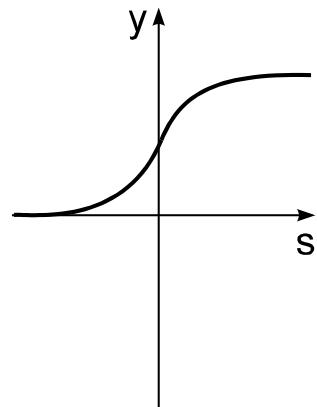
Ein neuronales Netz setzt sich aus mehreren PE zusammen. Üblicherweise werden dazu verschiedene Schichten von PE untereinander vernetzt.

Künstliche Neuronale Netze

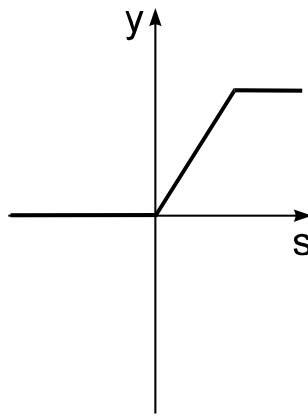
- Aktivierungsfunktionen -

Aktivierungsfunktion

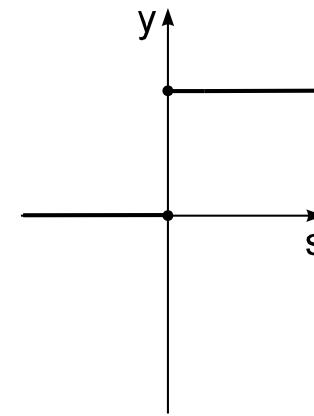
$$y = f(s)$$



(a)



(b)



(c)

Beispiele für Aktivierungs-funktionen einer Prozesseinheit:
sigmoidale (a),
pseudo-lineare (b) und
treppenartige (c)

$$y = f(s)$$

Unter MATLAB

$$a = f(n) \text{ wobei } n = p * w + b$$

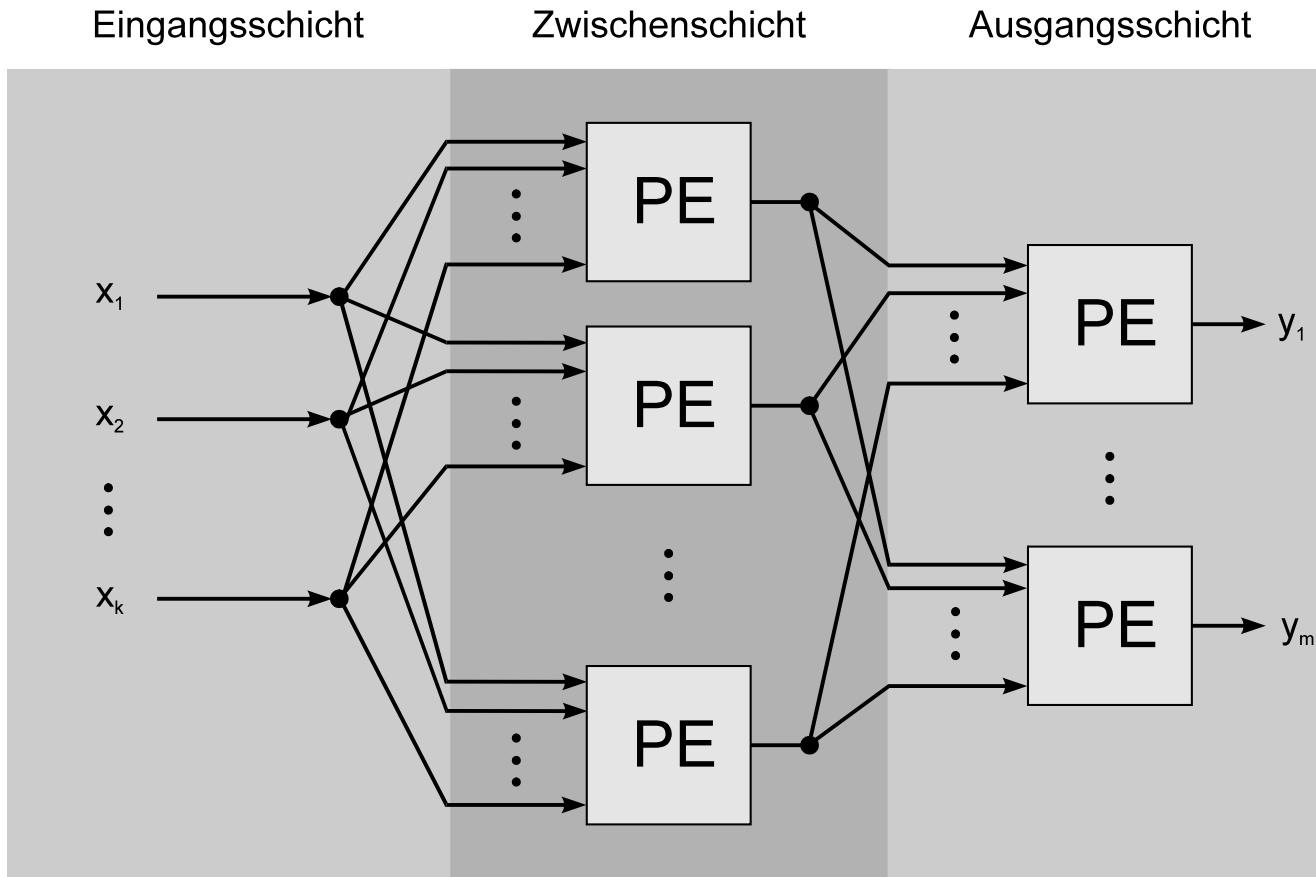
MATLAB:

$a = \text{softmax}(n)$	Compete transfer Function
$a = \text{hardlim}(n)$	Hard Limit Transfer Function
$a = \text{Hardlims}(n)$	Symmetric Hard Limit Func.
$a = \text{logsig}(n)$	Log-Sigmoid Transfer Funktion
$a = \text{poslin}(n)$	Positive Linear Transfer Func.
$a = \text{purelin}(n)$	Linear Transfer Function
$a = \text{radbase}(n)$	Radial Basis Function
$a = \text{satlin}(n)$	Satlin Transfer Function
$a = \text{satlins}(n)$	Symmetric Satlin Function
	Softmaxs, Tan-Sigmoid, Triangular;



Künstliche Neuronale Netze

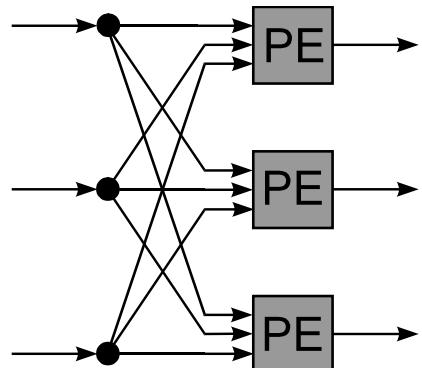
- Netzaufbau -



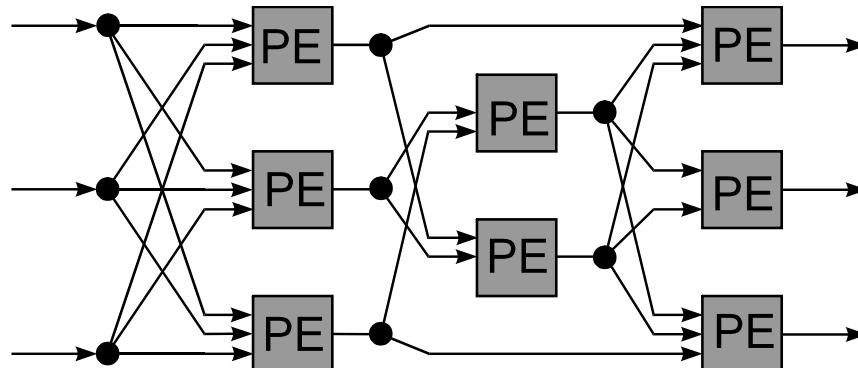
Vernetzung von mehreren Prozesseinheiten

Künstliche Neuronale Netze

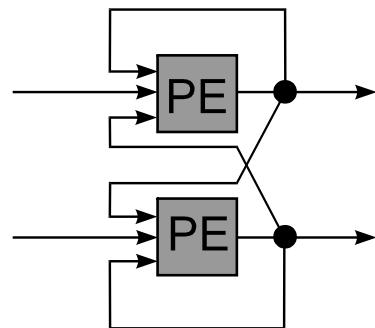
- Netzstrukturen -



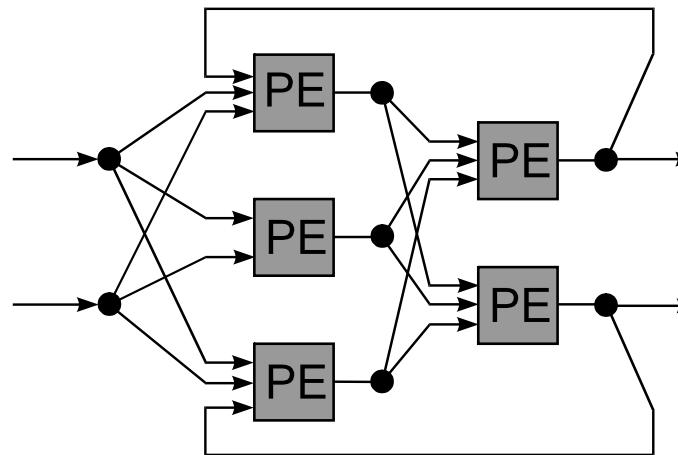
(a)



(b)



(c)



(d)

Verknüpfungsmöglichkeiten neuronaler Netze: (a) und (b) vorwärts gerichtet, (c) und (d) rückgekoppelt (Multi Layer Perceptron)

Künstliche Neuronale Netze

- Implementierung -

- Die Entscheidung, wie die einzelnen PE untereinander zu verknüpfen sind, ist problemabhängig.
- So lassen sich manche Probleme nur von bestimmten Netzwerktypen lösen.
- Die Wahl des Netzwerktyps, also die Bestimmung der PE und deren Verbindungen untereinander, beruht oftmals auf Erfahrungen und erfolgreichen früheren Einsatzes zur Lösung gleicher oder ähnlicher Probleme
- Anwendungsentwicklung = Prototyping -Test -Verifikation
- Anlernen des Netzes (Training)
- Nachteil: Trainingsdaten müssen vorliegen! Zusätzlich Testdaten notwendig
Diese Daten können ev. in der Vergangenheit gesammelt worden sein, ein grundlegendes Verständnis der Zusammenhänge ist notwendig!
- Computerprogramme können diese Vorgehensweise optimal unterstützen

Künstliche Neuronale Netze

- Die Perzeptron Konvergenzregel -

1. Schritt: Initialisierung der Gewichte

Weise den Gewichten $w_k(0)$ ($0 \leq k \leq n$) kleine Zufallswerte zu.

Mit $w_k(t)$ ist das Gewicht der Eingabe x_k zum Zeitpunkt t bzw. Lernschritt t gemeint (hier $t = 0$, also zum Zeitpunkt der Initialisierung).

2. Schritt: Präsentiere eine neue Eingabe und die gewünschte Ausgabe

Eine Eingabe wird dem Netz präsentiert, was für jedes x_k einen entsprechenden Wert bedeutet, und dazu die gewünschte Ausgabe $d(t)$.

3. Schritt: Berechne die eigentliche Ausgabe

Zu den gegebenen Werten wird die tatsächliche Ausgabe berechnet, die sich von der gewünschten Ausgabe unterscheiden kann. Die Ausgabe wird mit der bekannten Formel berechnet:

$$y(t) = f\left(\sum_{k=1}^n (w_k(t) \cdot x_k(t))\right)$$

Dabei ist im Vergleich zu Gleichung 1 nur der Zeitpunkt t als zusätzliche Abhängigkeit hinzugefügt worden.

4. Schritt: Anpassung der Gewichte

$$w_k(t+1) = w_k(t) + l [d(t) - y(t)] x_k(t), \quad 0 \leq k \leq n$$

In dieser Gleichung ist l ein Faktor kleiner 1 und größer 0, auch Lernrate genannt
Wenn die gewünschte Ausgabe $d(t)$ mit der berechneten Ausgabe $y(t)$
übereinstimmt, wird das Gewicht nicht geändert

5. Schritt: Wiederhole durch Gehen zum 2. Schritt

Künstliche Neuronale Netze

- Beispiel 1 -

Einfaches neuronales Netz nach dem Perceptron-Schema (nach Aleksander, Morton 1990, S. 21)

Aufgabe: Unterscheide H und T Buchstaben

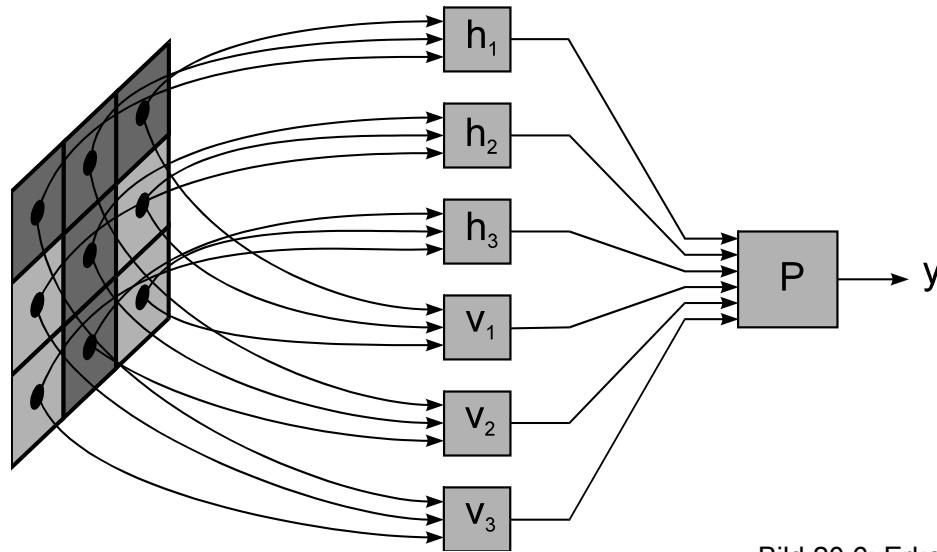


Bild 20.6: Erkennung von H und T Buchstaben

$h_1 - h_3 = 1$ wenn mindestens 2 Punkte schwarz sind, sonst 0;

$v_1 - v_3 = 1$ wenn mindestens 2 Punkte schwarz sind, sonst 0;

P hat Aktivierungsfunktion $f(s)$ und Gewichte w_i

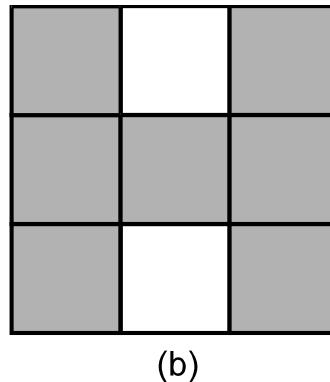
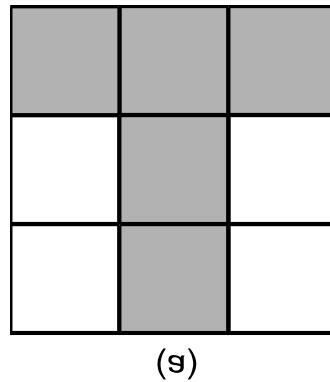
Künstliche Neuronale Netze

- Beispiel 1 -

Einfaches neuronales Netz nach dem Perceptron-Schema (nach Aleksander, Morton 1990, S. 21)

Aufgabe: Unterscheide H und T Buchstaben

Mit einer passenden Aktivierungsfunktion ist nun dafür zu sorgen, dass
 $y = +1$ (das Ergebnis der Summation ist 1) für T-Buchstaben und
 $y = -1$ (das Ergebnis der Summation ist -1) für H-Buchstaben ist.



Trainingsdaten zum Einlernen des Netzes, Buchstabe T (a) und Buchstabe H (b) (nach Aleksander, Morton 1990, S. 23)

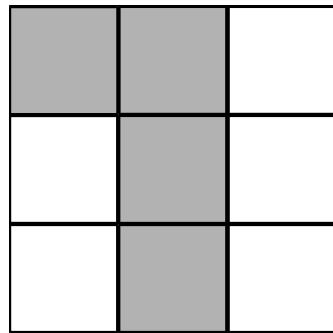
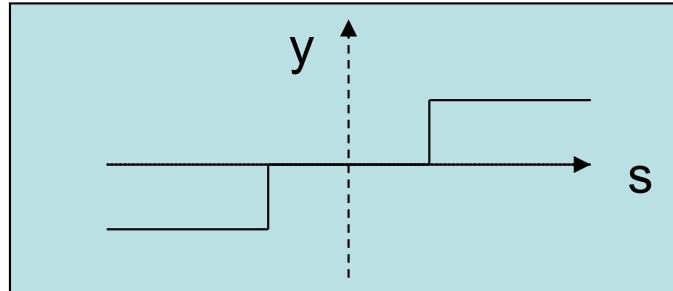
- a) die Gewichte $h1$ und $v2$ sollen positive Werte als Ergebnis haben, z. B. +1,
 - b) die Gewichte von $v1$, $v3$, $h1$, $h2$ und $h3$ müssten dann niedrige Werte haben, beispielsweise -1. Da $h1$ zum Erkennen beider Buchstabentypen benötigt wird, soll das Gewicht einen niedrigen Wert nahe 0 annehmen.
- Daraus ergibt sich für w: $w = (0, -1, -1, -1, +1, -1)$

Künstliche Neuronale Netze

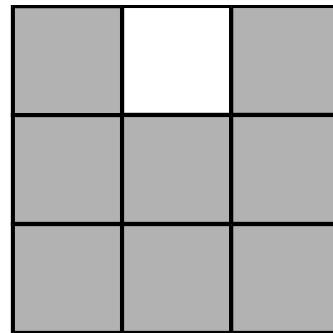
- Beispiel 1 -

Aktivierungsfunktion:

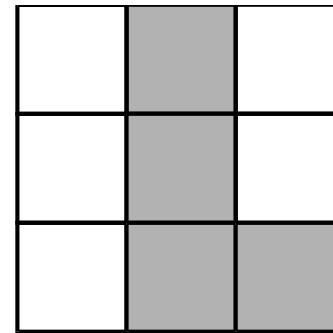
$$\begin{aligned}f(s) &= 1 ; s \geq 1 \\&= 0 ; -1 < s \geq 1 \\&= -1 ; s \leq -1\end{aligned}$$



(a)



(b)



(c)

Erkennung von Mustern als T- und H-Buchstaben (nach Aleksander, Morton 1990, S. 24)

$$w = (0, -1, -1, -1, +1, -1)$$

Daraus ergäbe sich für ein T die Summe: $0+0+0 +0+1+0= 1$

und für H die Summe = $0+(-1)+(-1)+(-1)+0+(-1)= -4$

und für die Beispiele aus Bild 20.8:

a) $0+0+0 +0+1+0 = 1 ;$

T Buchstabe

b) $0+(-1)+(-1) +(-1)+1+(-1) = -3 ;$

H Buchstabe

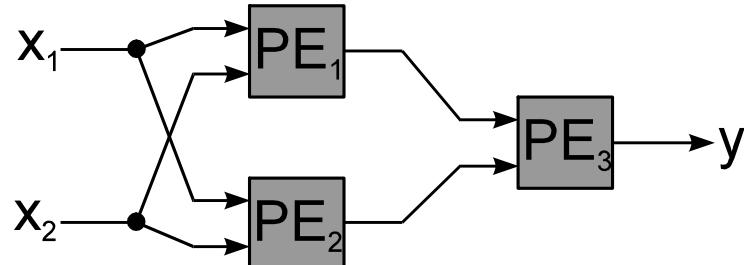
c) $0+0+(-1) +0+1+0= 0 ;$

weder T noch H

Künstliche Neuronale Netze

- Beispiel 2 -

Die Antivalenz oder das Exklusiv Oder (XOR)



Frank Rosenblatt zeigte, dass ein einfaches Perzeptron mit zwei Eingabewerten und einem einzigen Ausgabeneuron zur Darstellung der einfachen logischen Operatoren AND, OR und NOT genutzt werden kann. Marvin Minsky und Seymour Papert wiesen jedoch 1969 nach, dass ein einlagiges Perzeptron den XOR-Operator nicht auflösen kann (Problem der linearen Separierbarkeit). Dies führte zu einem Stillstand in der Forschung der künstlichen neuronalen Netze (KNN).

XOR Eingang x_1, x_2	Summe PE ₁	Ausgang PE ₁	Summe PE ₂	Ausgang PE ₂	Summe PE ₃	Ausgang y
1, 1						
1, 0						
0, 1						
0, 0						

Zu vervollständigende
Tabelle

Künstliche Neuronale Netze

- Beispiel 2 -

Exklusiv Oder (XOR)

PE1 hat die **Gewichte**:

-1 für w1 (Eingang x_1) und **+1 für w2** (Eingang x_2).

Das Ausgangssignal wird wie in Gleichung 1 beschrieben berechnet. Die Aktivierungsfunktion wird so gewählt, dass beim Überschreiten eines **Schwellwertes**, hier **0.5**, die PE als Ergebnis +1 liefert, andernfalls 0.

Für **PE2** gilt für die Gewichte:

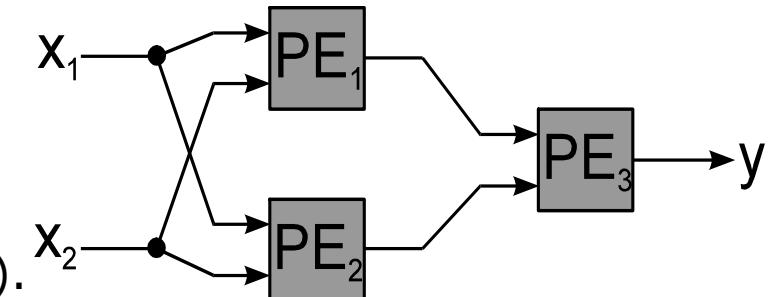
+1 für w1 (Eingang x_1) und -1 für w2 (Eingang x_2),

Schwellwert ist ebenfalls 0.5 und das Ausgangssignal wird auch nach Gleichung 1 berechnet.

Für **PE3** gilt für die Gewichte:

+1 für w1 (Ausgang von PE1) und +1 für w2 (Ausgang von PE2),

Schwellwert ist ebenfalls 0.5 und das Ausgangssignal wird auch nach Gleichung 1 berechnet.

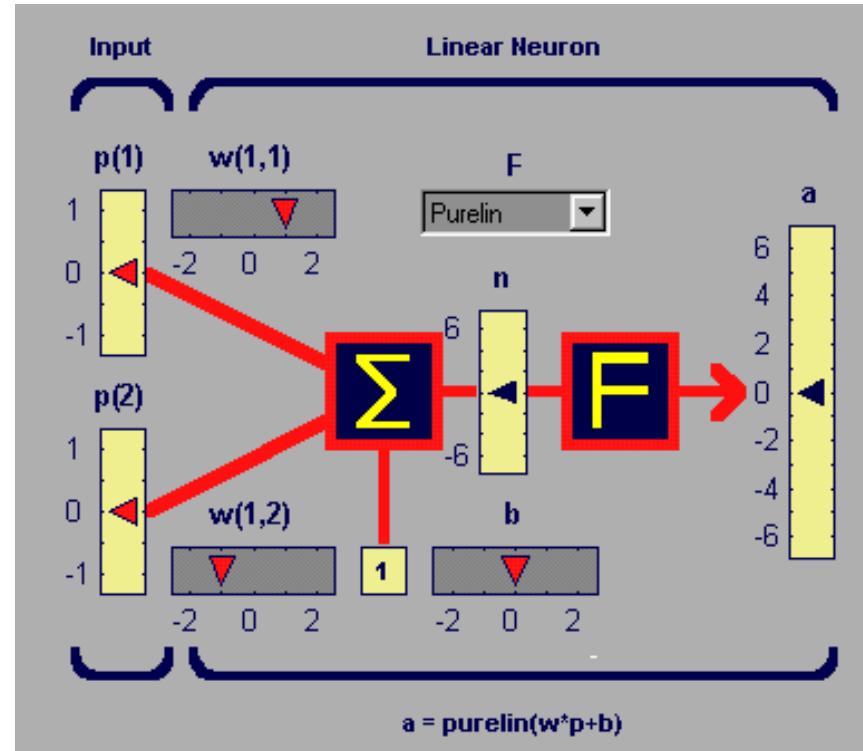


Eingang x_1, x_2	Summe PE ₁	Ausgang PE ₁	Summe PE ₂	Ausgang PE ₂	Summe PE ₃	Ausgang y
1, 1	0	0	0	0	0	0
1, 0	-1	0	1	1	1	1
0, 1	1	1	-1	0	1	1
0, 0	0	0	0	0	0	0

Künstliche Neuronale Netze

- Prozesseinheit unter MatLab -

Demo Single Layer Perceptron (SLP) unter Matlab (NN)

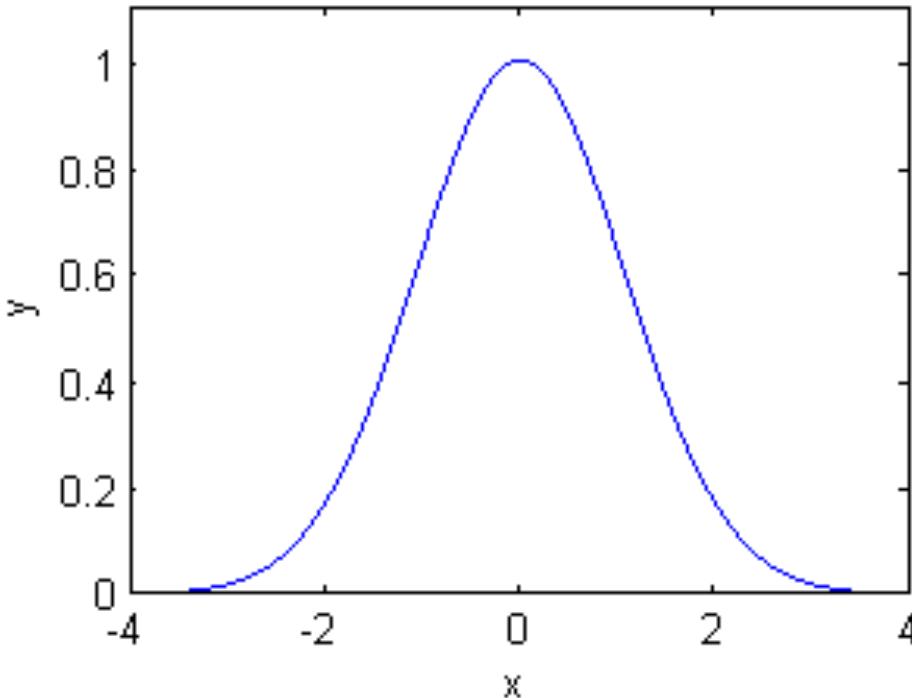


$$\text{Zwischenergebnis } n = w^*p+b$$

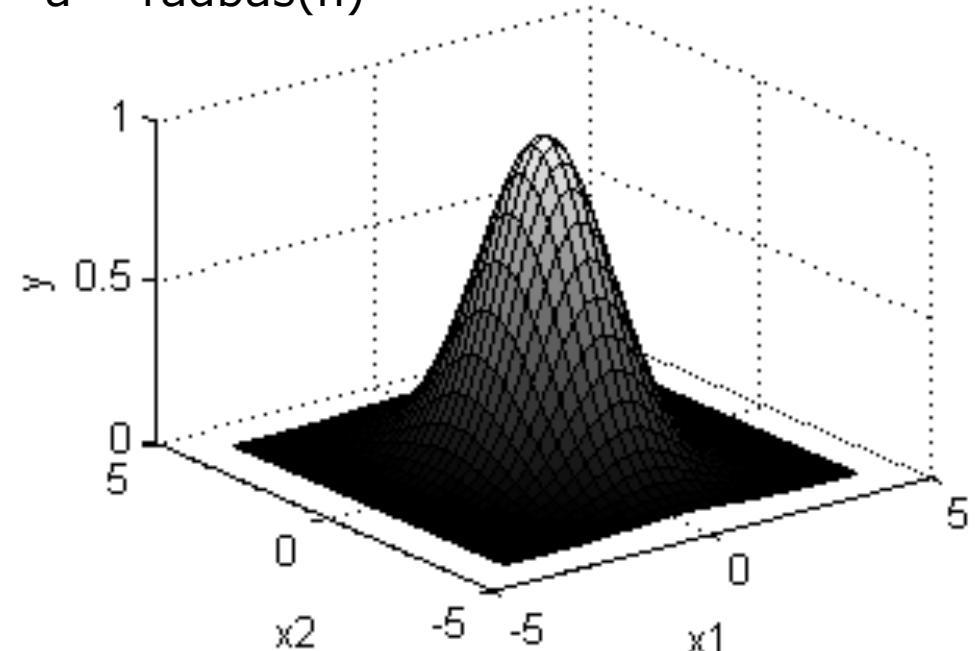
Künstliche Neuronale Netze

- Radiale Basis Funktionen -

Radiale Basis Funktionen (RBF)



$a = \text{radbas}(n)$



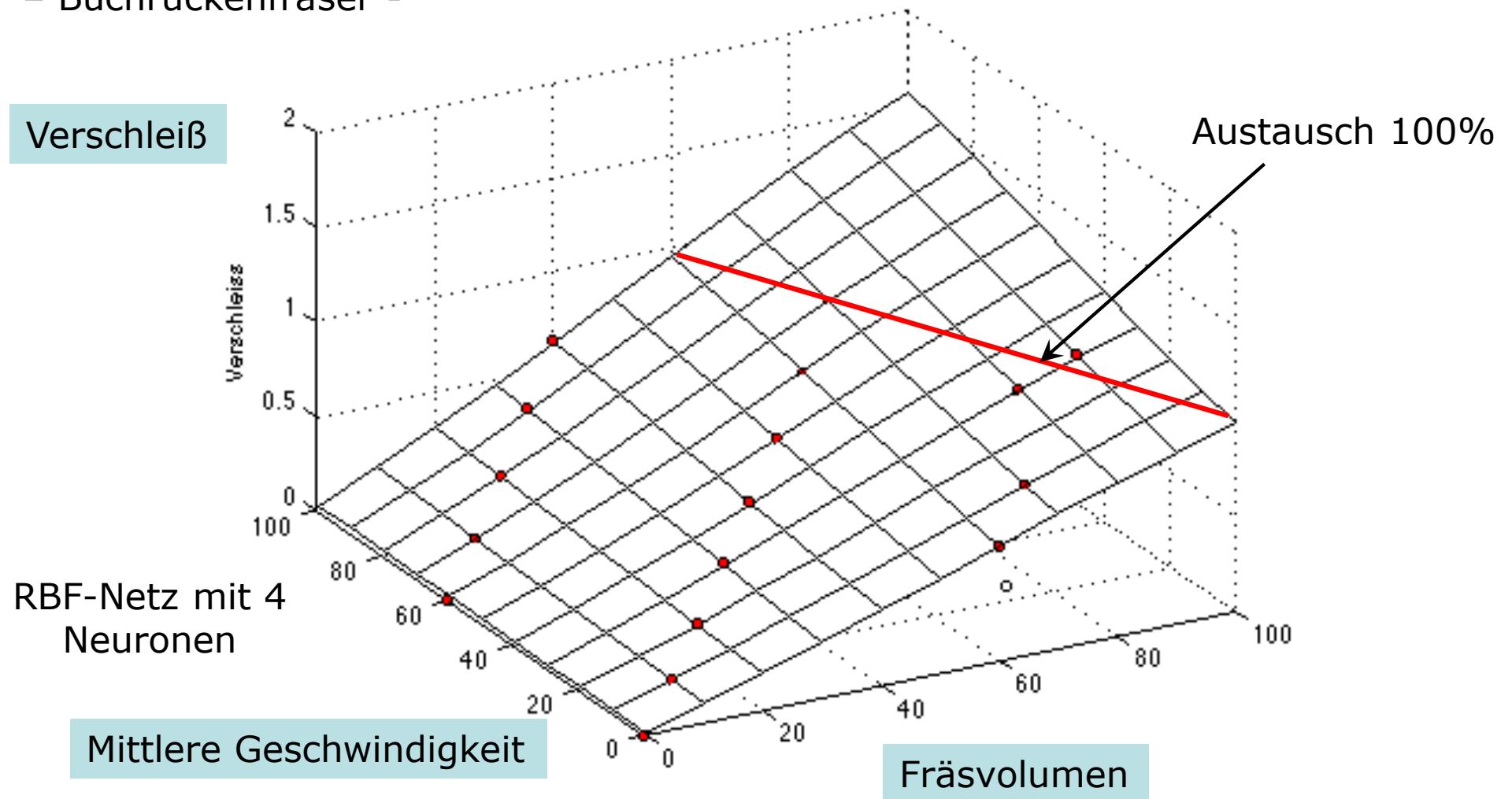
Zwei einzelne Gaußglocken, Ein- und Zweidimensional
In beiden Fällen ist das Zentrum der Gaußglocke im Nullpunkt
Der Abstand r zum Zentrum $(0; 0)$ berechnet sich schlicht aus dem Satz des Pythagoras: $r = \text{Quadratwurzel aus } x_1^2 + x_2^2$.

RBF Netz : Aufsummieren der Gaußglocke

Künstliche Neuronale Netze

- Beispielanwendung – Papierfräser -

Digitale Maschinenakte: Verschleißprognose aufgrund der Prozessdaten, Beispiel
– Buchrückenfräser –



Künstliche Neuronale Netze

- Beispielanwendung 2 -

Laser-chemische Mikroumformung (SFB747-A5)

Ziel: Abtragen von Material im Mikrometerbereich zur Herstellung qualitativ hochwertiger Kleinstwerkzeuge

Motivation:

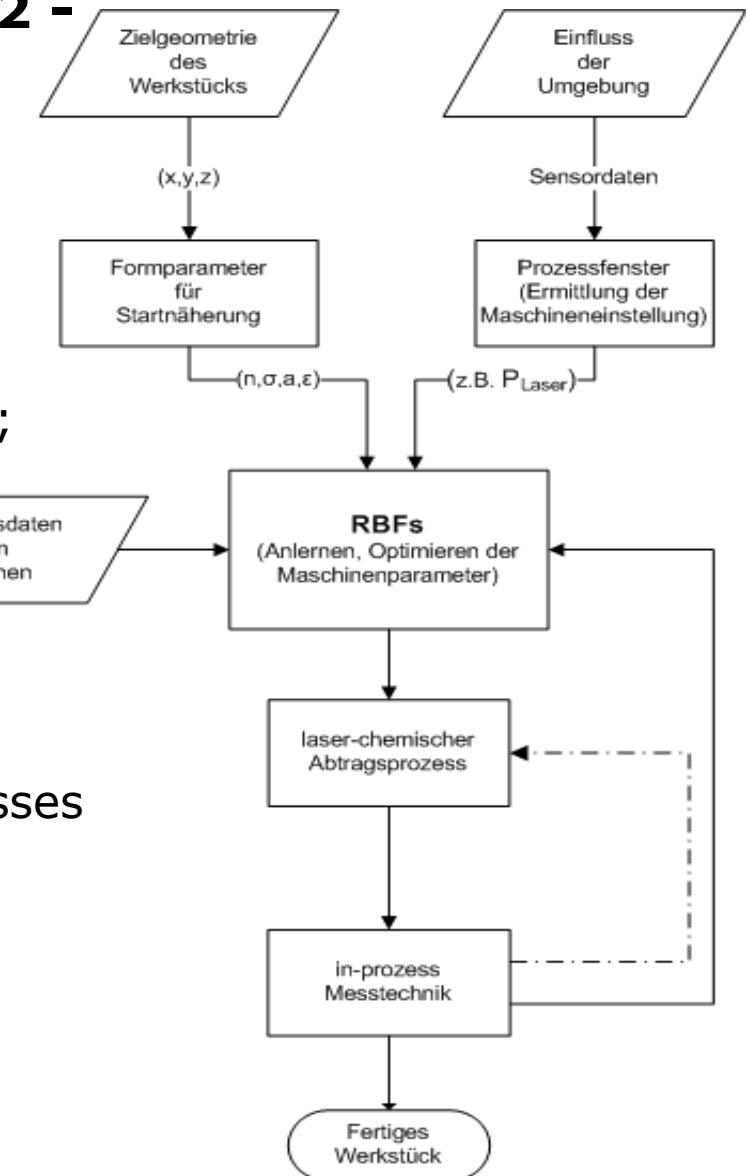
- Prozess unterliegt sehr vielen Parametern
(Anzahl n der Abtragsbahnen; Abtragsgeometrie σ, a ; Qualitätsanforderung, z.B. Rauheit ϵ Umgebungseinflüsse, z.B. Temperatur, Druck, Feuchtigkeit, etc.)
- Erlernen auf Basis teilstatistischer Versuchspläne

Vorteile RBF:

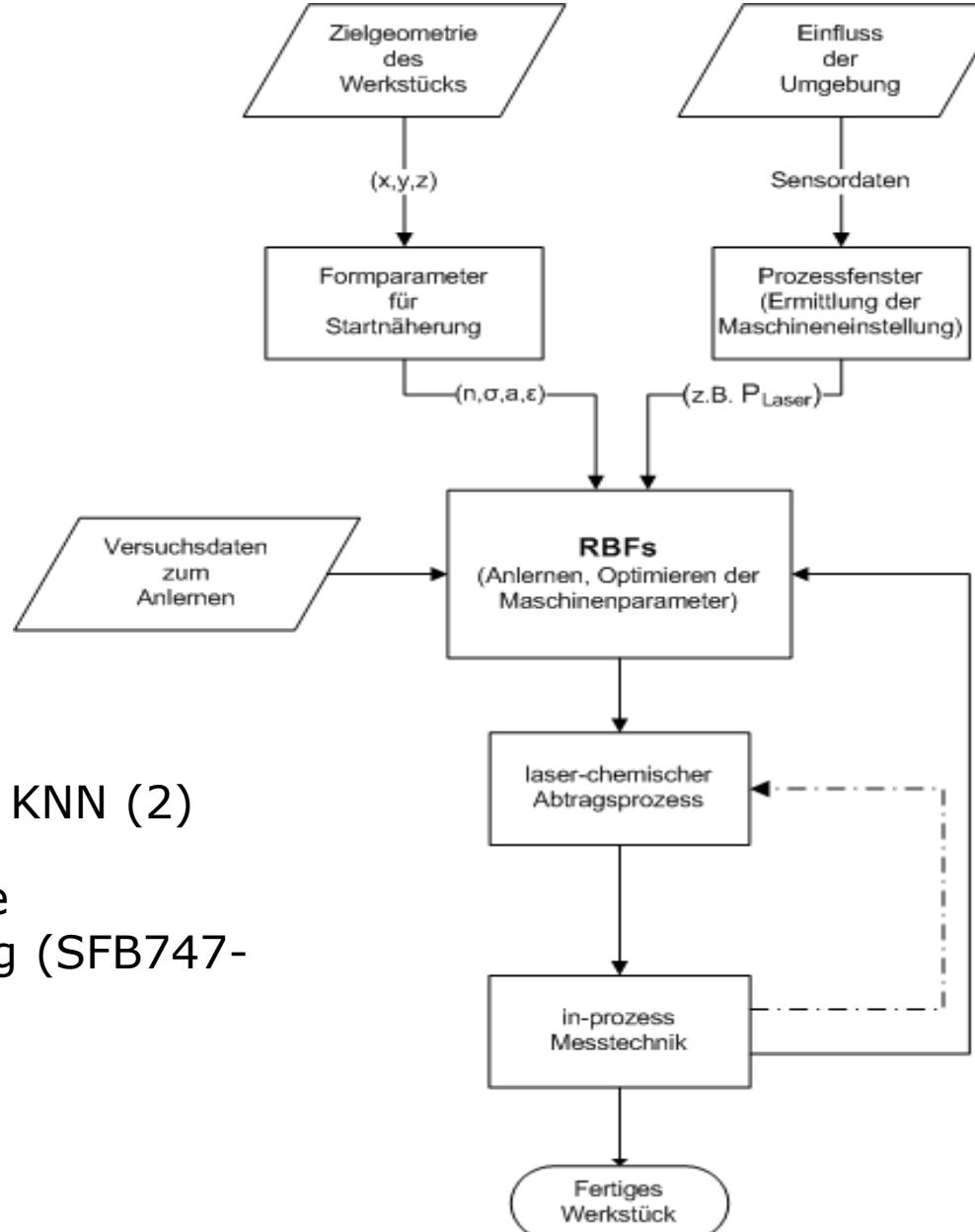
- Hohe Datenverarbeitungsrate, große Flexibilität
- Schnellere und sicherere Konvergenz des Lernprozesses
- Hohe interne Transparenz
(Aufbau und Funktionsweise nachvollziehbar)

Nachteile RBF:

- Vorinformationen erforderlich
(Wissen über Prozessgrenzen erforderlich, da eingeschränkte Extrapolationsmöglichkeit)
- Größerer Stichprobenumfang des Lerndatensatzes



Künstliche Neuronale Netze



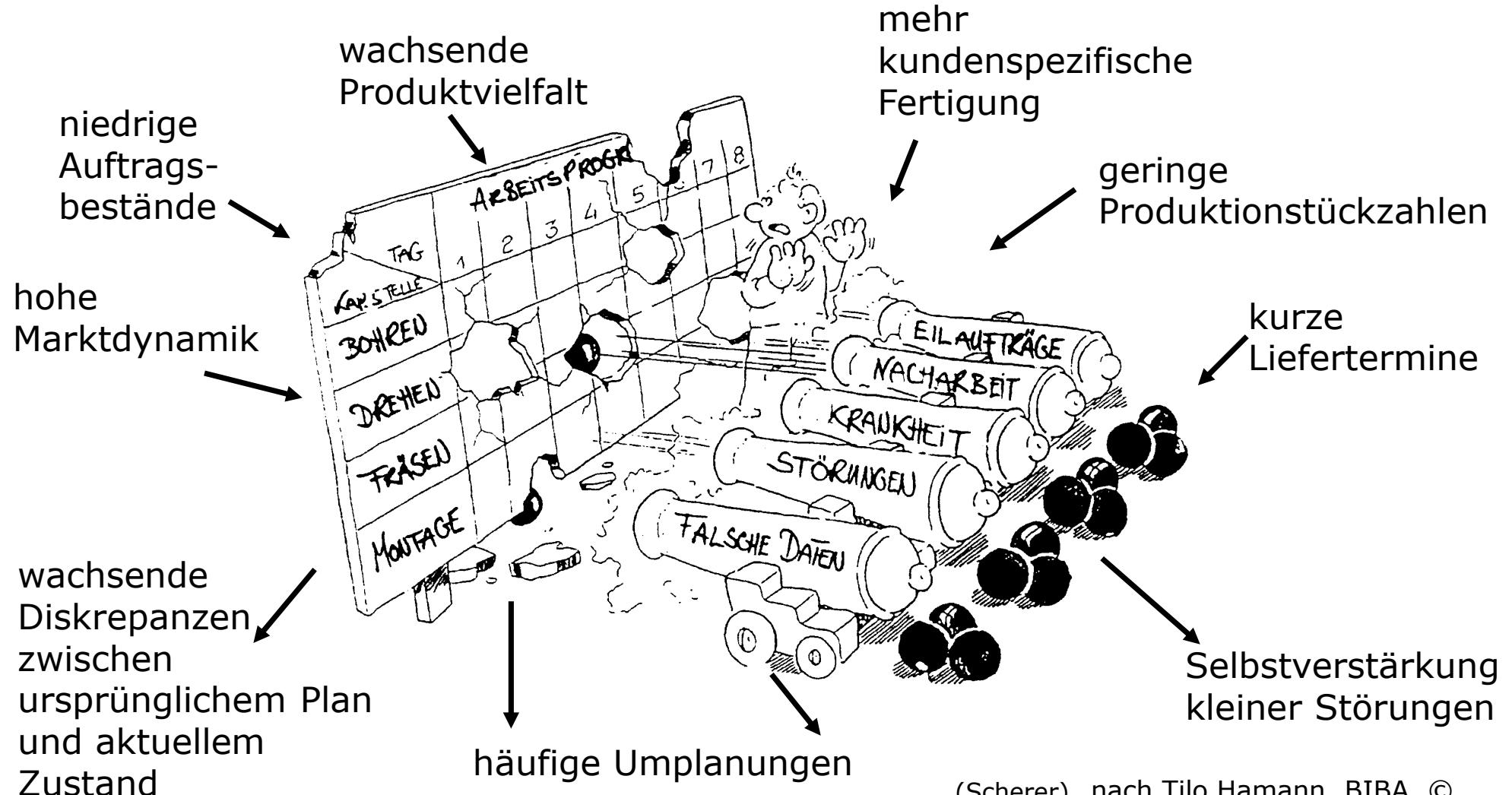
Anwendung von KNN (2)

Laser-chemische
Mikroumformung (SFB747-
A5)

Künstliche Neuronale Netze

- Anwendungsbeispiel 3 -

Modellierung und Steuerung der Produktion mit künstlichen Neuronalen Netzen



(Scherer) nach Tilo Hamann, BIBA ©

Künstliche Neuronale Netze

- Anwendung 3 – Durchlaufzeiten in der Fertigung -

Anwendung von KNN

Aufgabe

- Durchlaufzeit der Lose durch das Fertigungssystem und die Bestände an den Arbeitsplätzen werden geregelt

Erwartete Vorteile der KNN gegenüber konventioneller Regelung

- Lernfähigkeit, Parallelität, Verteilte Wissensrepräsentation
- Höhere Fehlertoleranz, Assoziative Speicherung von Informationen
- Robustheit gegen Störungen oder verrauschte Daten
- Aktive Repräsentation

Erwartete Nachteile der KNN

- Wissenserwerb nur durch Lernen möglich
- keine Introspektion möglich (z.B. durch "Selbstbeobachtung")
- Logisches (sequentielles) Schließen ist schwer zu realisieren
- Lernen ist relativ langsam

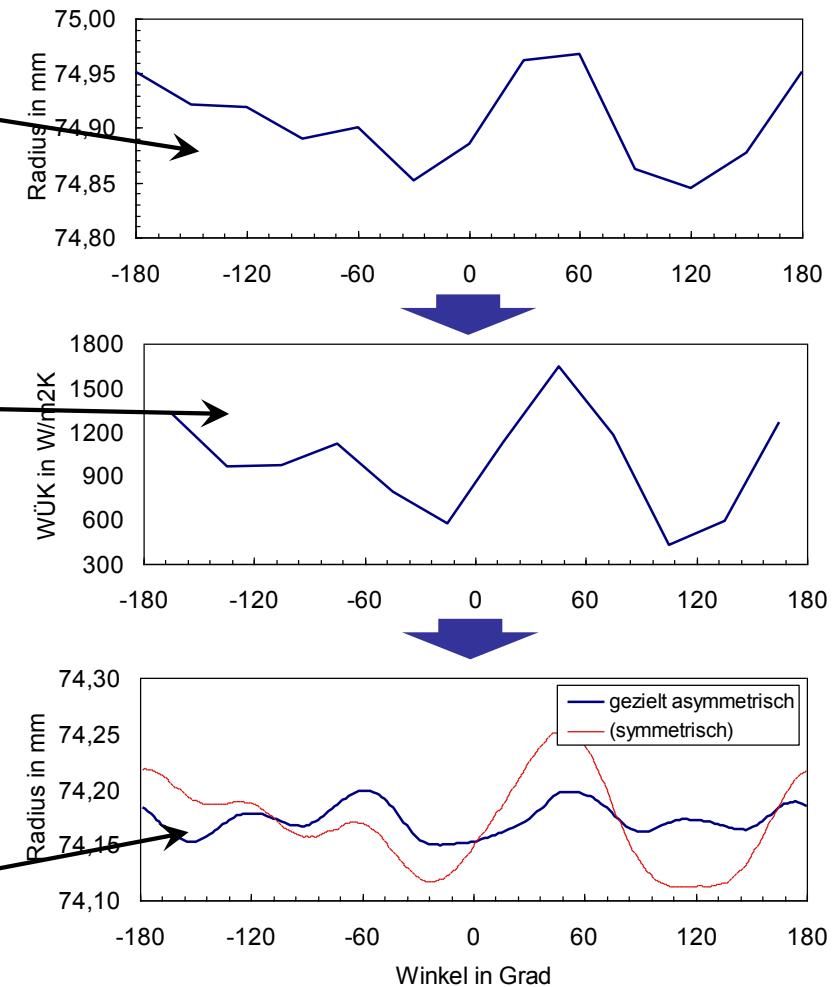
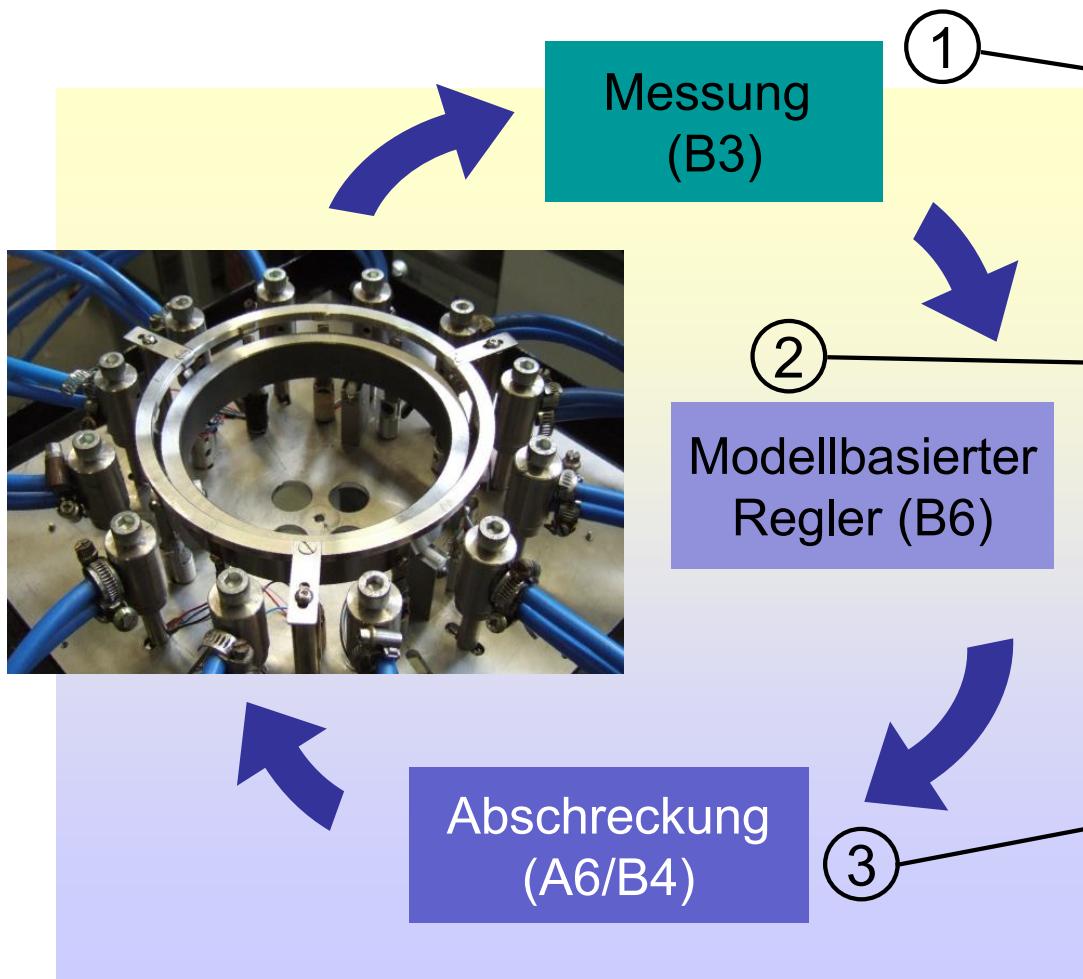
nach Tilo Hamann, BIBA ©

Künstliche Neuronale Netze

- Anwendung 4 -Gasabschreckung -

Anwendung von KNN (4)

Gezielt asymmetrische Gasabschreckung



Künstliche Neuronale Netze

- Weitere Anwendungen und Einsatzgebiete -

Luftfahrt: Autopilot, Steuerungssysteme, Flugroutensimulation, Komponentensimulation, Fehlererkennung in Komponenten

Fahrzeugindustrie: Leitsysteme, Gewährleistungsanalyse, Motor- und Getriebediagnose, Montagefehlererkennung und Diagnose

Militär: Zielverfolgung, Objektseparierung, Steuerung und Regelung, Signal- und Bildverarbeitung in Sonar und Radaranwendungen

Elektronik: Schaltungs- und Chiplayout, Chip-Fehleranalyse, Sprachbe- und -verarbeitung (Digitale Telephonie), Bild- und Stimmerkennung, nichtlineare Modelle

Produktionstechnik: Prozess-Steuerung und Regelung, Produktdesign und Analyse, Prozess und Maschinendiagnose, Vorhersage, Wartung, Bildverarbeitung uvm.

Banken und Versicherungen: Dokumentenerkennung, Kreditvergabeprüfung
... **Unterhaltungsindustrie, Medizin, Öl- und Gasindustrie,** Robotic, Finanzindustr

Zukünftig (BIMAQ): Getriebediagnose in Windenergieanlagen

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 **Neuronale Netze**
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 07 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



Bremer Institut für
Messtechnik, Automatisierung
und Qualitätswissenschaft

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 **Automatisiertes Messen und Steuern**
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Automatisiertes Messen und Steuern

- Überblick -

Messen (Sensoren, Digitalisierung, Vorverarbeitung) und **Steuern** (Verarbeiten der Messergebnisse, der Ergebnisse der Vorverarbeitung, Entscheidung, Berechnung der Ausgaben, Regeln) **Steuern** - Ausgabe (analoger oder digitaler Stellgrößen und Datenübertragung, digital/analog Umsetzung für Aktoren)

Messen

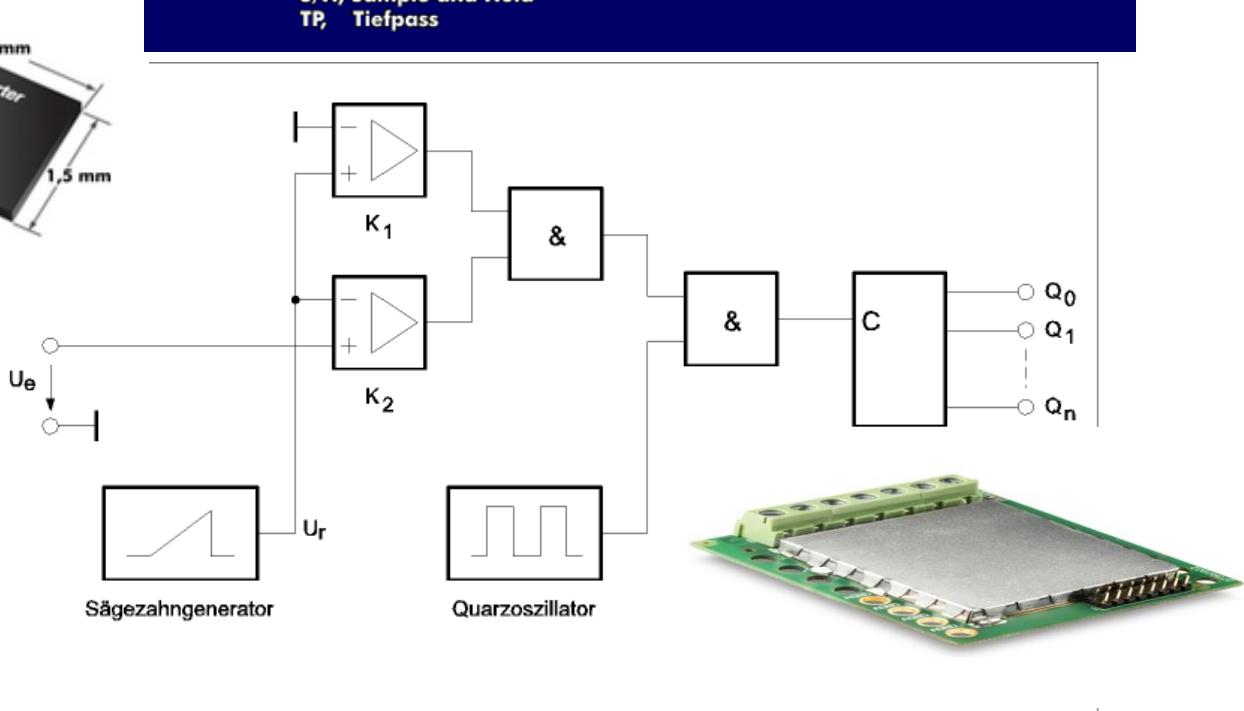
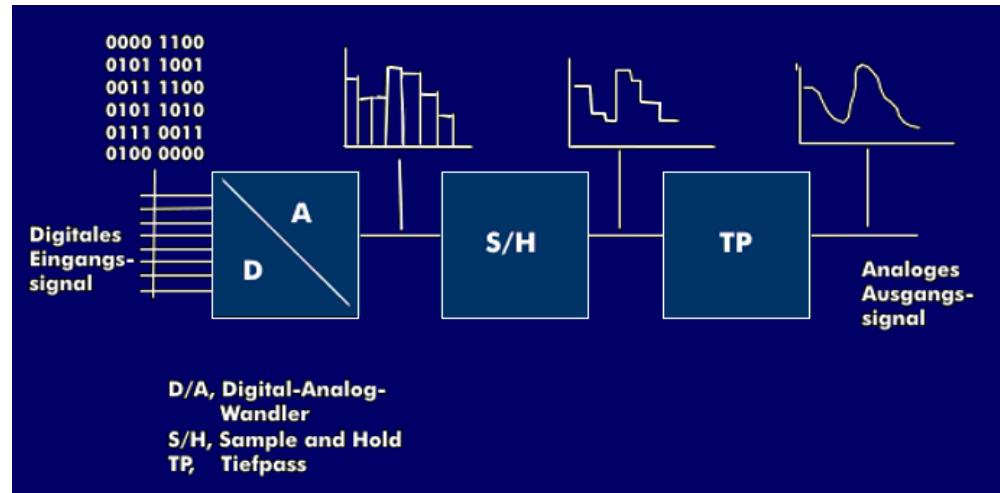
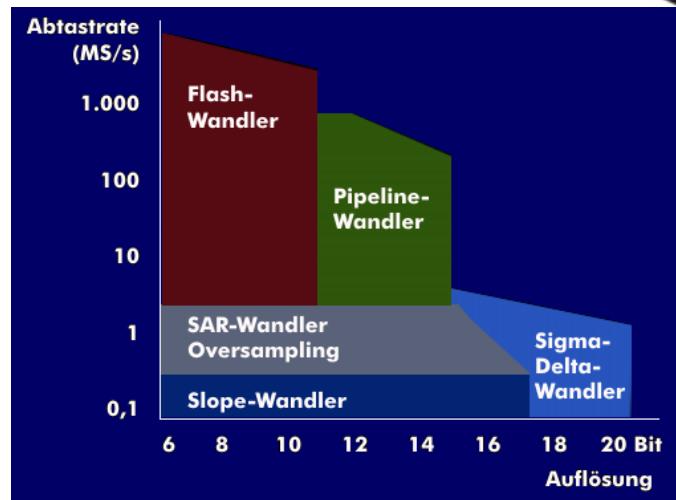
- Messdatenaufnahme Sensoren Wandler A/D Wandler, Sensorbus (Echtzeit)
- Messdatenauswertung – Hardware und Software (-Systeme und Methoden)
- Hardare – FPGAs, DSP, Microcontroller
 - FPGA Field Programmable Gate Array, DSP Digitaler Signalprozessor
- Algorithmen
- Softwarewerkzeuge MatLab, LabView sowie Hardware Targets (XPC Target)
- Programmiersprache, Compiler, „Erzeugersoftware“

Automatisiertes Messen und Steuern

- Messen und Steuern -

- Industrielle Steuerungstechnik
 - SPS, Robotersteuerung/ CNC Steuerungen
 - Spezialisierte Automatisierungssysteme**
 - I/O Karten
 - A/D D/A Wandler
 - Software

16-Bit-A/D-Wandler ADS1115
im QFN-Package von Texas Instruments



Automatisiertes Messen und Steuern

- Wiederholung: Basis und Integrationstechnologien -

Programme

Steuerungen

Automatisierungselemente:
Sensoren / Aktoren

Basistechniken der Automatisierung:

- Sensor- und Aktortechnik
- Sensorsysteme, Mechtronik
- Regelungstechnik
- Steuerungstechnik
- Leittechnik (PLS, PLT)
- Robotertechnik

Integrationstechniken der Automatisierung:

- Rechnertechnik
- Informationstechnik
- Kommunikationstechnik (LAN WAN RFID WLAN)
- Mensch-Maschine-Systeme
- Struktur- und Systemtechnik
- Managementtechniken

Automatisierung von Folgeprozessen ([Steuerungstechnik](#), [Robotertechnik](#))

Automatisierung von Fließprozessen ([Regelungstechnik](#), [Leittechnik](#))

Automatisiertes Messen und Steuern

- Industrielle Steuerungen in der Automatisierung -

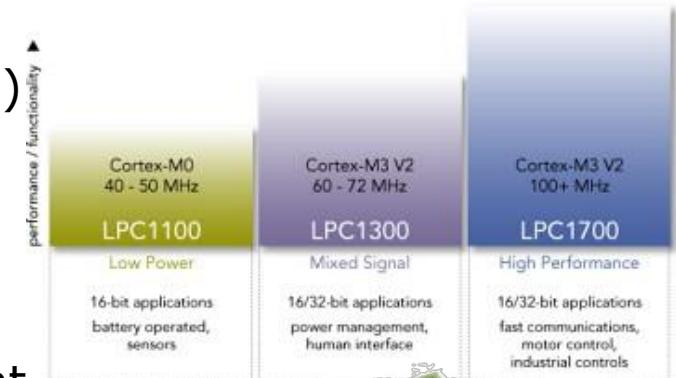
<ul style="list-style-type: none"> • SPS Speicherprogrammierbare Steuerungen 	<ul style="list-style-type: none"> • RC Robotersteuerungen 	<ul style="list-style-type: none"> • CNC Computerized Numerical Control 	<ul style="list-style-type: none"> • AS Automatisierungssysteme • (spezialisierte Prozessrechner)
<ul style="list-style-type: none"> • Steuerung und Überwachung eines Prozesses - von Maschinen oder Anlagen Echtzeit Bus Systeme (Feldbus, PROFIBUS) Dezentralisierung, WLAN, Mobilfunk, RFID (Industrie 4.0) 	<ul style="list-style-type: none"> • Bewegungsprogramm • Positionierung • Bahnsteuerung (3-D) • Ablauffolge LAN (Ethernet) Echtzeit Bus Systeme (Feldbus, PROFIBUS) Dezentralisierung, WLAN, Mobilfunk, RFID (Industrie 4.0) 	<ul style="list-style-type: none"> • Bearbeitungsprogramme • Konturen (2-D) / Bahnen 3D, mehrere Achsen • Bearbeitungskenngrößen (überwachen und steuern) • LAN, Feldbusse, Echtzeitkommunikation 	<ul style="list-style-type: none"> • Prozesssteuerung • Überwachung • Erkennung • Identifizieren • Klassifizieren • LAN Feldbusse Echtzeitkommunikation, WLAN, Mobilfunk, NFC, RFID, (Industrie 4.0)
<ul style="list-style-type: none"> Einmal-Programmierung SPS Programm 	<ul style="list-style-type: none"> • Wiederholte Anwenderprogrammierung • Bewegungsprogramm 	<ul style="list-style-type: none"> • Wiederholte Anwenderprogrammierung • Bearbeitungsprogramm 	<ul style="list-style-type: none"> • Einmal-Programmierung • z.B. Menügeführt • „Funktionsorientiertes“ Programm

Automatisiertes Messen und Steuern

- Messdatenverarbeitung und Steuerung -

- **Micro-Controller:** Programmierung von Micro-Computern (Controllern) als Steuerungen -
 - Direkte Implementierung mit Programmiersystem (PC) oder von ausgetesteten Entwicklungen (MatLab) durch „Compiler“
- **FPGAs:** Field Programmable Gate Array - ein integrierter Schaltkreis für logische Schaltungen
 - der Begriff Programmierung bei FPGAs beschreibt nicht nur die Vorgabe zeitlicher Abläufe, sondern auch die Definition der gewünschten Schaltungsstruktur
 - mittels Hardwarebeschreibungssprache formuliert und von einer Erzeugersoftware in ein Konfigurationsfile übersetzt
 - Beschreibt wie die physikalischen Elemente im FPGA verschaltet werden sollen.
„Konfiguration eines FPGAs“
- **DSP** Digitale Signalprozessoren
 - aufwendige Signalverarbeitungsaufgaben mit hoher Effizienz, spezialisierter hoch-integrierter Microcomputer

NXP ARM Cortex™ Microcontroller
Product Series Overview



Automatisiertes Messen und Steuern

- Messwerterfassung und Vorverarbeitung -

Sensor-Systeme:

z.B.: Laser, Laserdioden / CCD Kameras / Optik / Infrarottechnik
Bildverarbeitungsmethoden (Software, Algorithmen)

Analog Digital Wandler

FPGAs Field Programmable Gate Array

FPGA Einsatz:

- Echtzeit-Verarbeitung von einfachen bis komplexen Algorithmen
- digitalen Signalverarbeitung (digitale Filter, DSP)
- schnellen Fourier-Transformation
- Protokoll-Implementierungen, -Konvertierung
- die Kodierung/Dekodierung von digitalen Videosignalen
- die Ver-/Entschlüsselung von Daten in Echtzeit
- Fehlerkorrekturverfahren

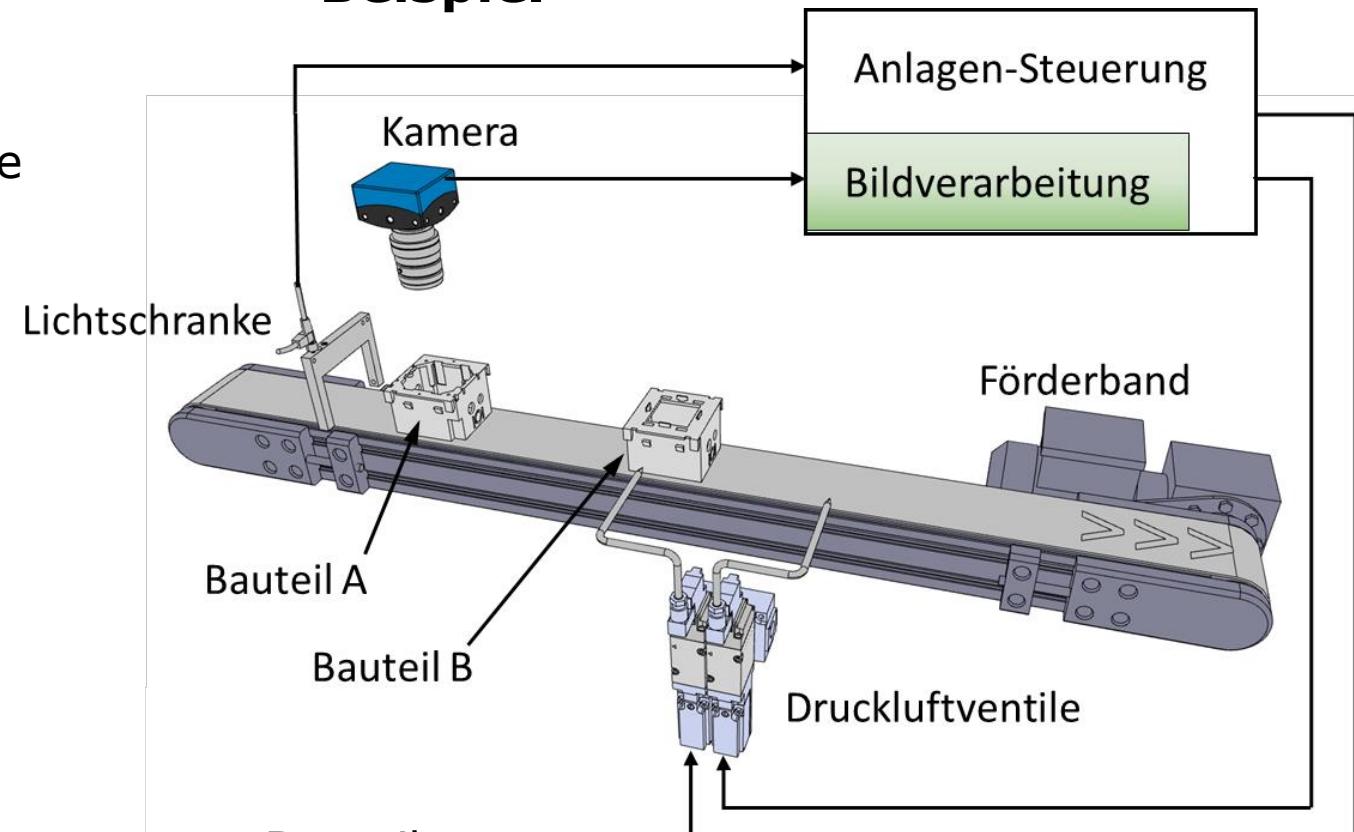
sind Anwendungsgebiete.



Automatisiertes Messen und Steuern

- Beispiel -

Aufbau der Sortieranlage



- Lichtschranke erkennt neues Bauteil
- Foto vom neuen Bauteil wird ausgewertet
- je nach Bauteil wird Ventil 1 oder 2 angesteuert, um das Bauteil vom Förderband in den jeweiligen Sammelbehälter geblasen
- nicht erkannte Bauteile werden am Ende des Förderbandes gesammelt

Automatisiertes Messen und Steuern

- Beispiel Programmiersprachen -

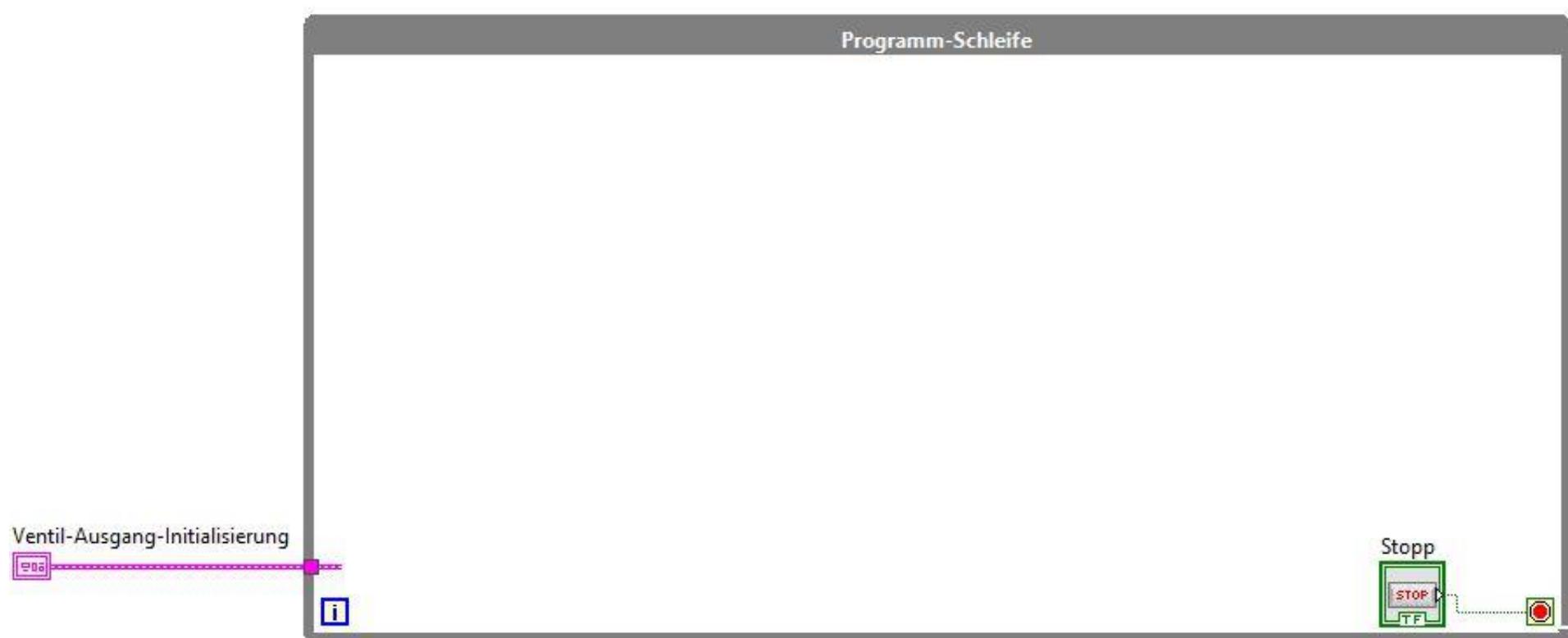
Programmiersprache LabVIEW

- Entwicklungsumgebung speziell für Techniker, Ingenieure, Wissenschaftler
- für Mess-, Steuer- und Regelsysteme
- Grafische Programmiersprache
- Maßgeschneiderte Benutzungsschnittstellen
- Grafisches Programm und Benutzungsschnittstelle bilden immer eine Einheit
- sehr umfangreiche Hardware-Integration
- Embedded-Hardware vom gleichen Hersteller



Automatisiertes Messen und Steuern

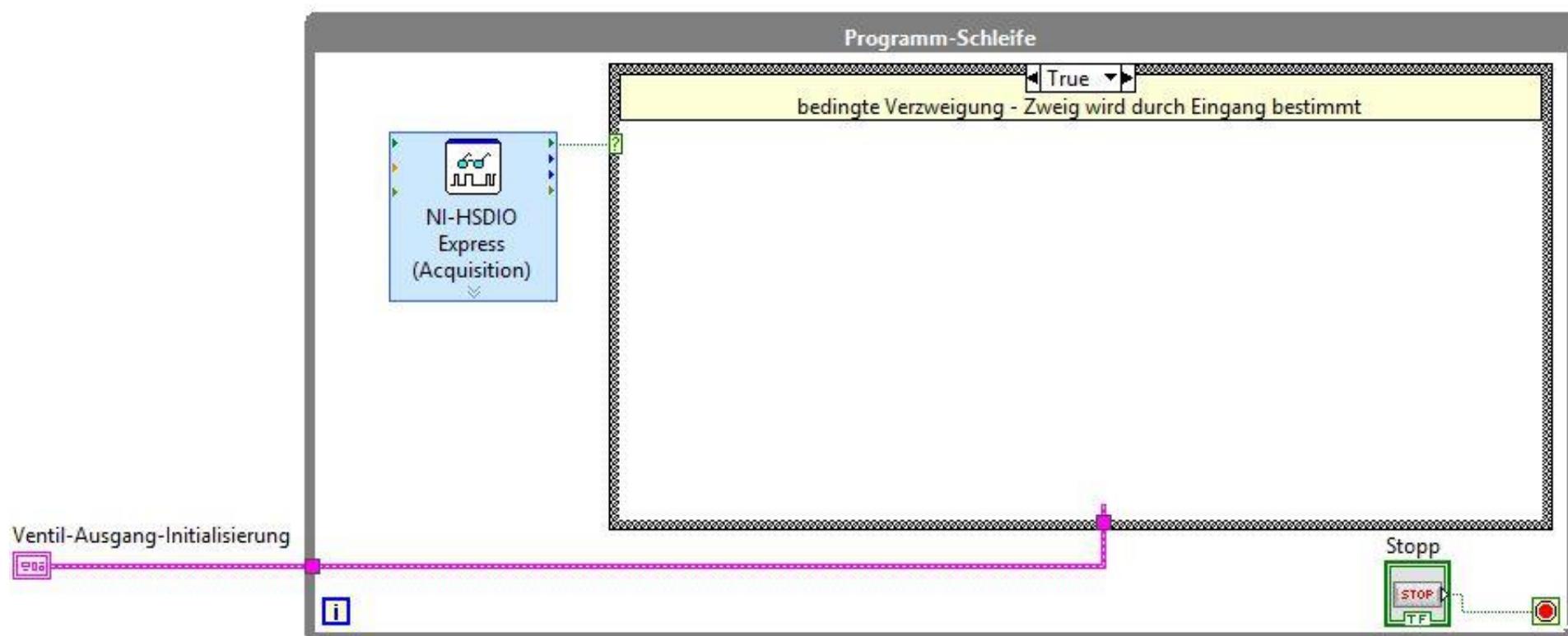
- Softwaresysteme LabView -



- Schleife für kontinuierlich laufendes Programm
- Mehrere Schleifen können parallel zueinander ausgeführt werden (Multitasking)

Automatisiertes Messen und Steuern

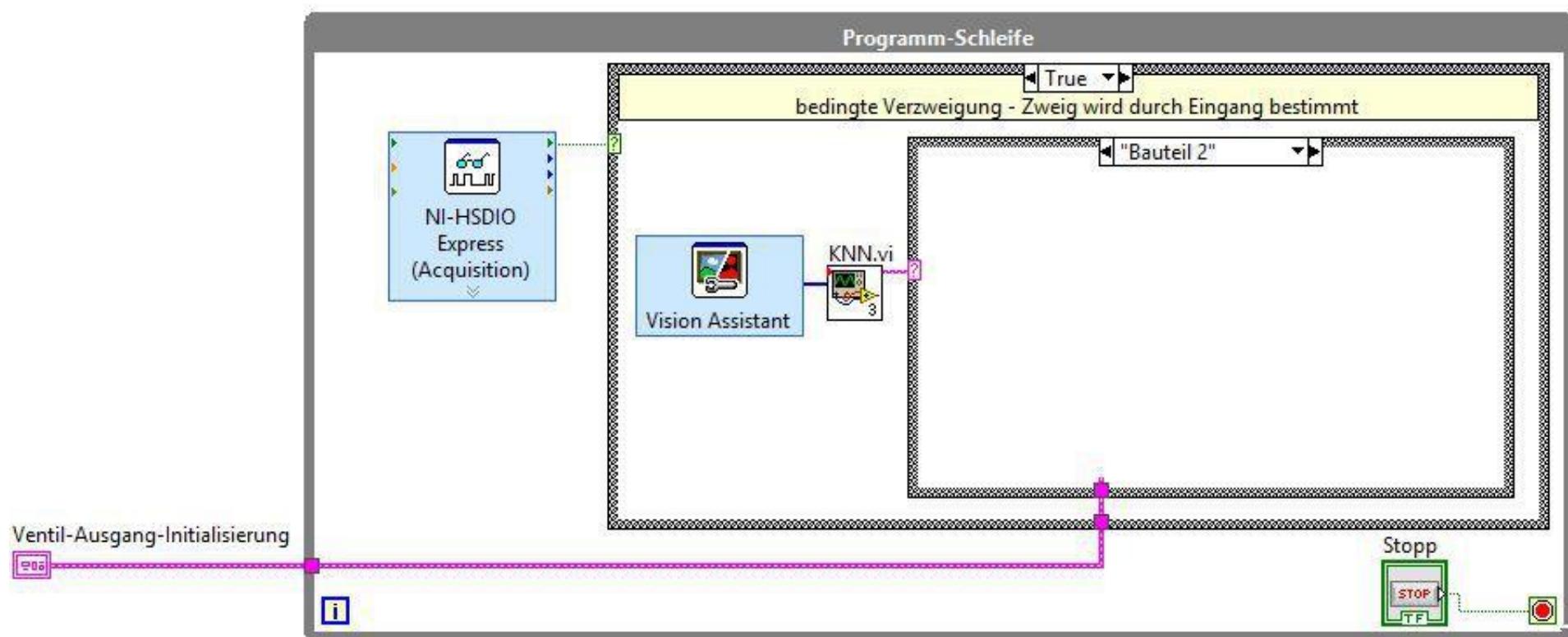
- Softwaresysteme LabView -



- Bedingte Verzweigung hier in Abhängigkeit eines digitalen Eingangs (entspricht IF oder CASE)
- Die einzelnen Fälle werden in jeweils eigenen Rahmen dargestellt.

Automatisiertes Messen und Steuern

- Softwaresysteme LabView -

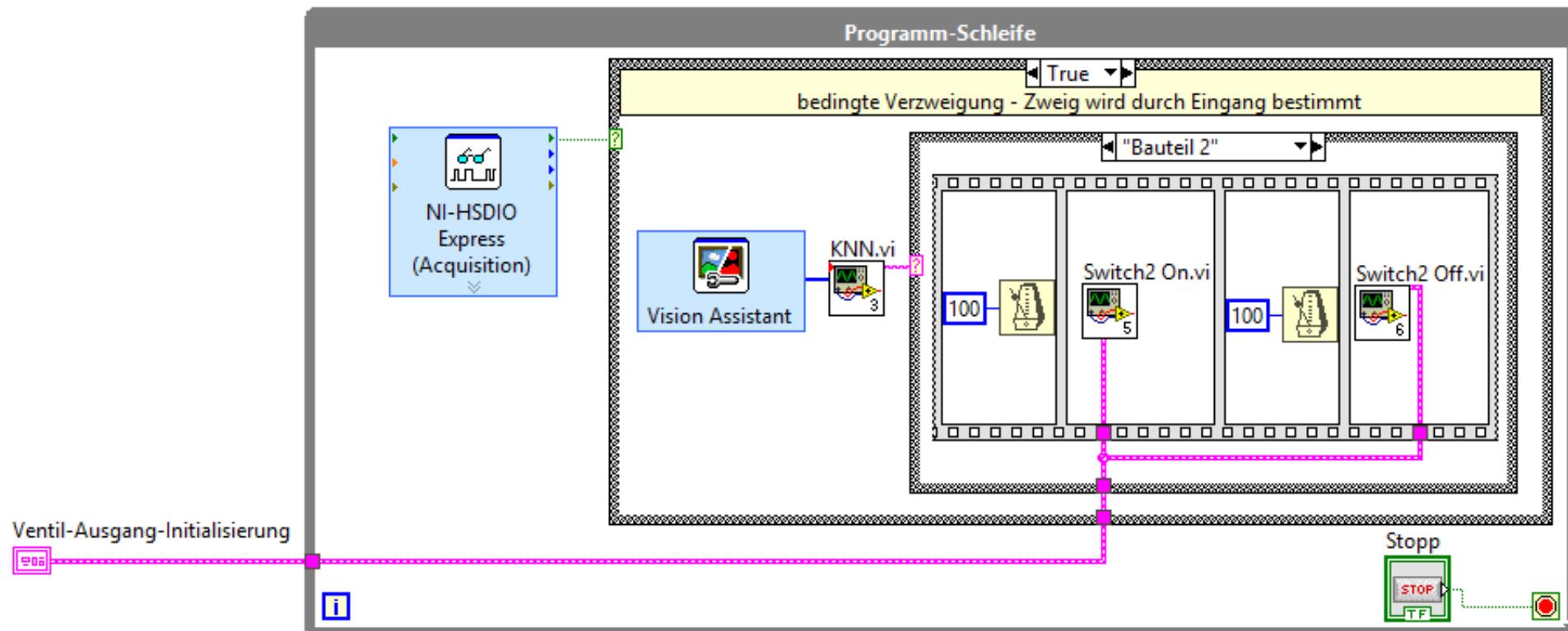


- Verarbeitung eines eingelesenen Bildes in einem Unterprogramm KNN.vi
- (Sub-)Programme werden Virtual Instruments (VI) genannt

LabVIEW - Beispielprogramm

Automatisiertes Messen und Steuern

- Softwaresysteme LabView -

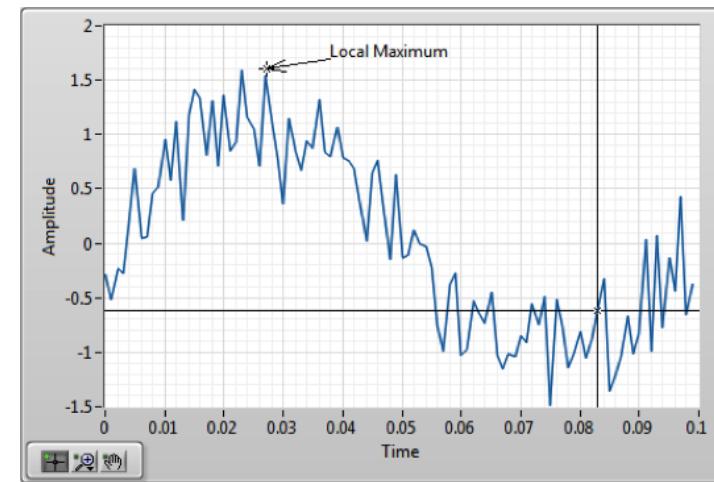
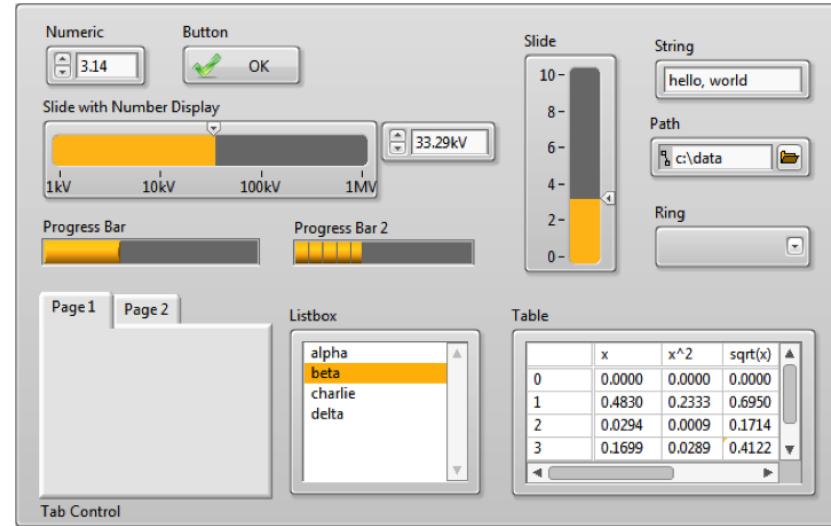
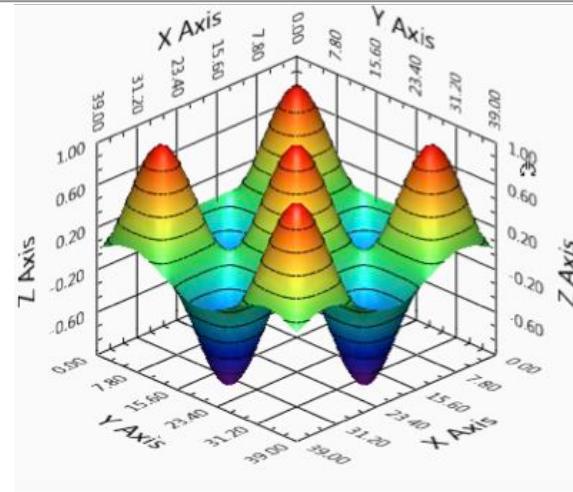
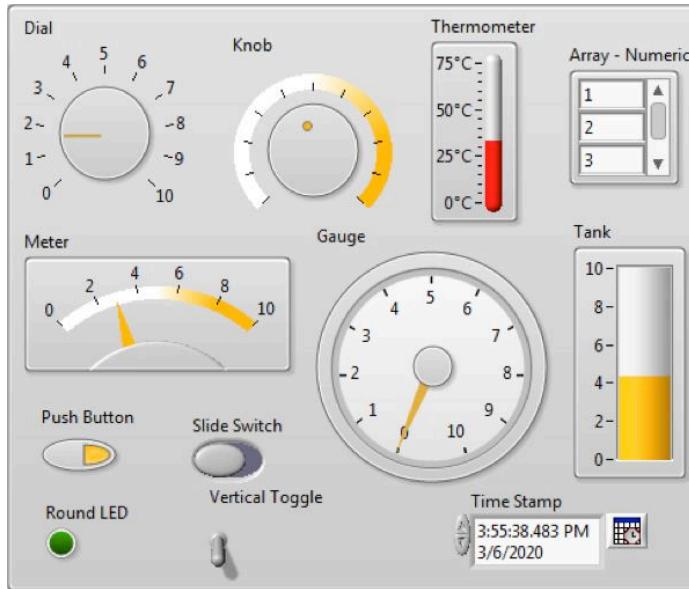


- Reihenfolge der Abarbeitung: alle Eingänge / Vorgänger müssen zuerst berechnet werden
- Besteht keine direkte Abhängigkeit, so können auch Sequenzen definiert werden

Automatisiertes Messen und Steuern

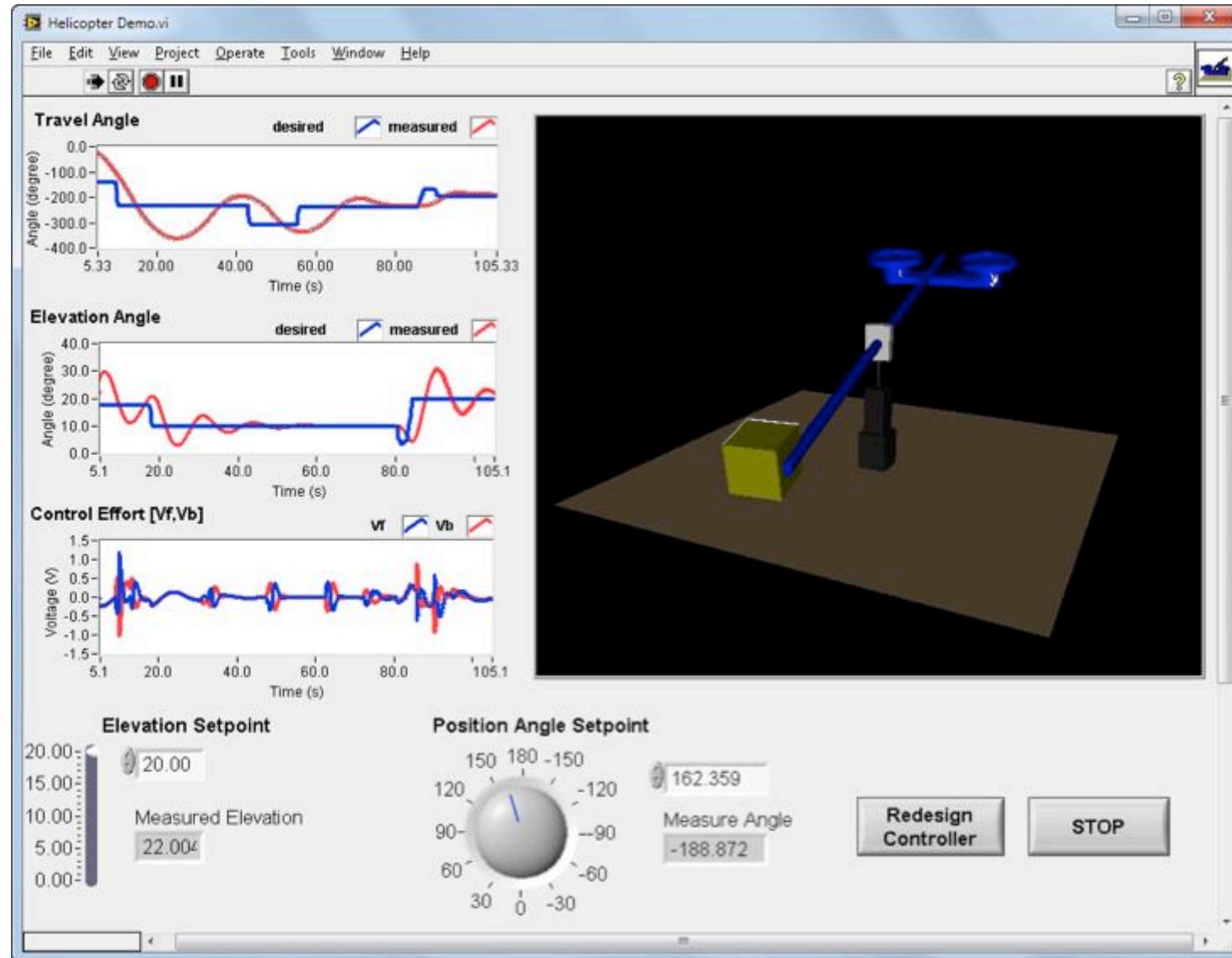
- Softwaresysteme LabView -

Beispiele für Elemente der Benutzungsschnittstelle



Automatisiertes Messen und Steuern

- Beispiel LabView: Helikoptersimulator mit 3D-Bildfunktion -



Automatisiertes Messen und Steuern

- Technologien, Standards, Protokolle, Kommunikation -

SCADA Supervisory control and data acquisition

Level 2 Prozessleitebene in der Automation (entsprechend der Pyramide, OSA Modell)

Aufgabe: z.B. „Bedienen und Beobachten, Rezeptverwaltung und Ausführung, Messwertarchivierung“

Anwendung: Prozesssteuerung, Windenergieanlagen (WEA)

Normen: IEC 62264 führt die Bezeichnungen Level 0 bis 4 ein und beschäftigt sich mit der Integration Ebenen 2,3 und 4.

ISA-88.01 und ISA-95 definieren ein physisches Modell.

Bus Systeme (Kommunikation):

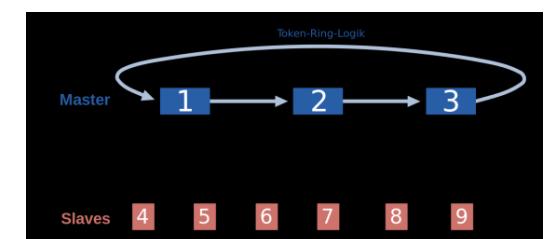
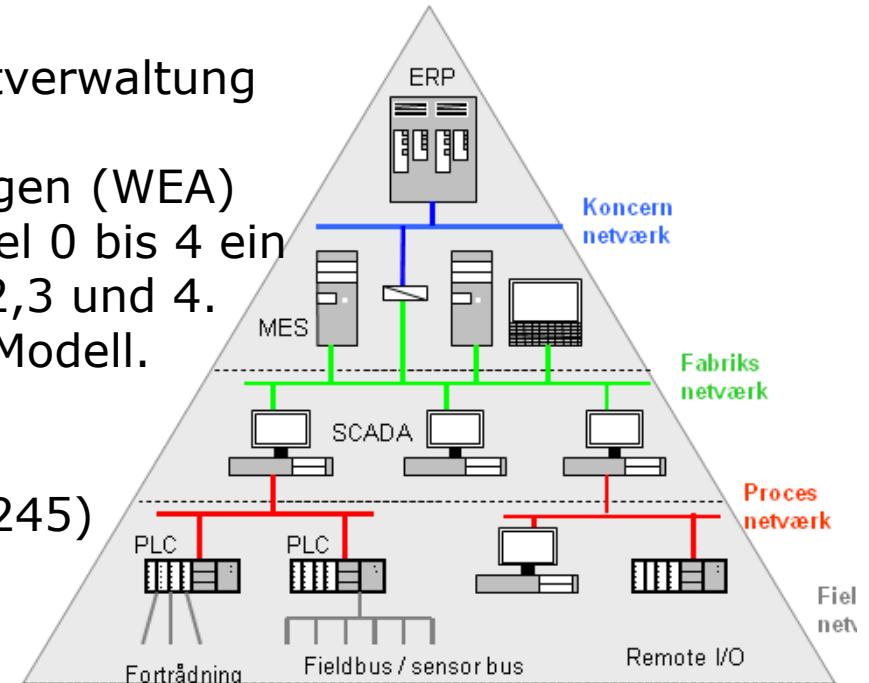
CAN Bus (PKW)

Feldbusse (Process field bus) PROFIBUS (DIN 19245)
seit 1999 in IEC 61158/IEC 61784 festgelegt.

Sensor Bus: AS-Interface ist in den Normen EN 50295 und IEC 62026-2 standardisiert.

Anwendungen

Fertigungsautomatisierung, In-Prozess Messtechnik, Energiesysteme, CMS Condition monitoring (WEA)



PROFIBUS Token Ring Quelle: Cabfdb

Automatisiertes Messen und Steuern

- Beispiel: In-process surface inspection -

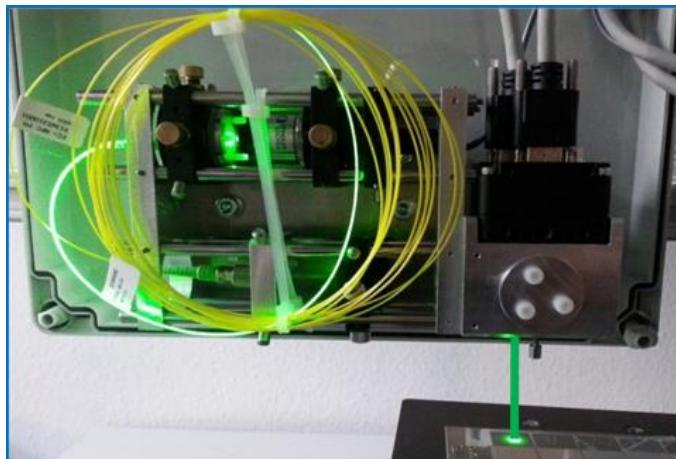
Task

- roughness characterization
- smooth technical surfaces
- running production processes
- surface velocity $< 300 \text{ m/min}$
- plane and cylindrical objects



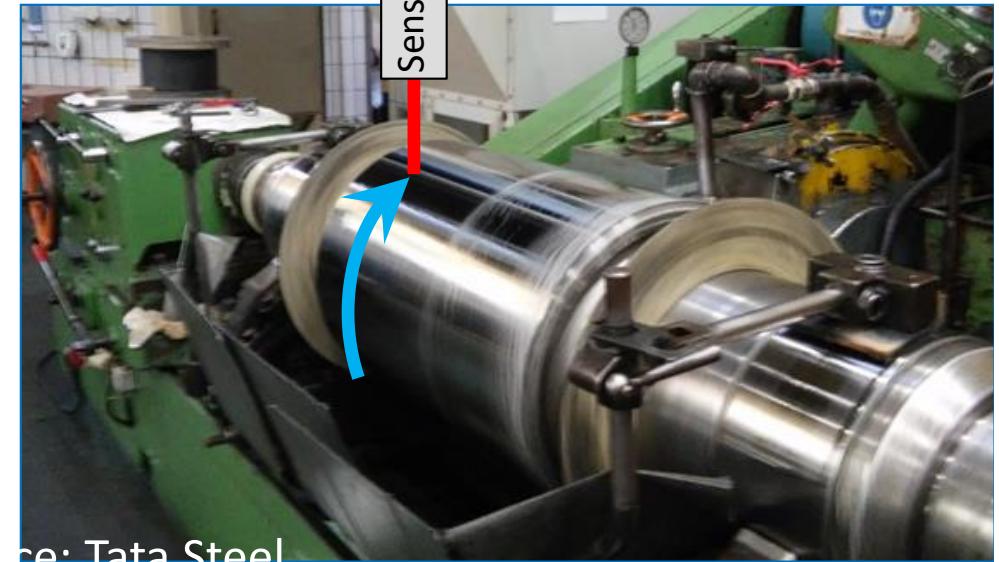
Objectives

- extensive surface coverage
 - large view field $\varnothing 10 \text{ mm}$
 - high measuring rate $> 300 \text{ Hz}$
- real-time measurement



Optical
measuring
setup

Work rolls with optically smooth surfaces

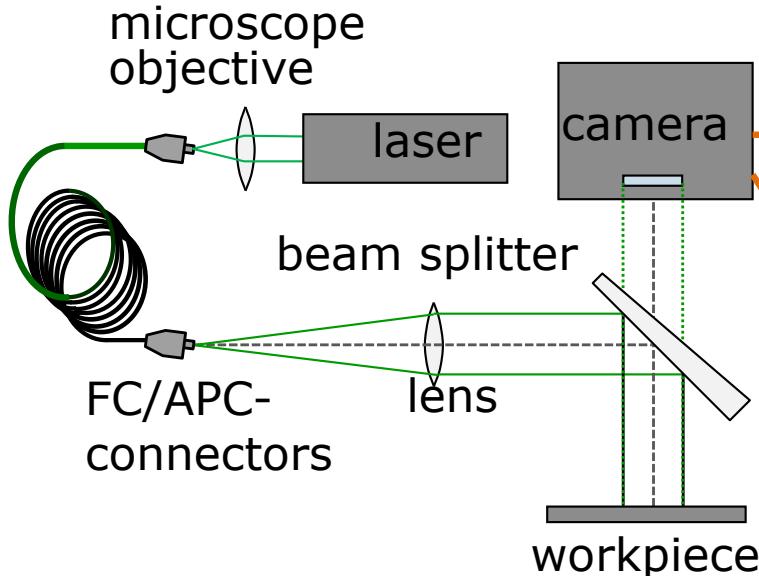


Work roll recycling and measurement

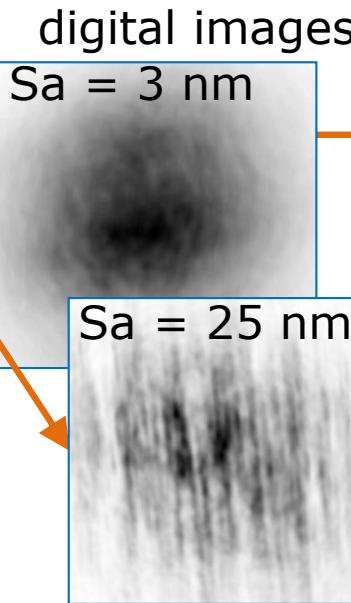
Automatisiertes Messen und Steuern

- Messtechnische Lösung - Rauheitsmessung -

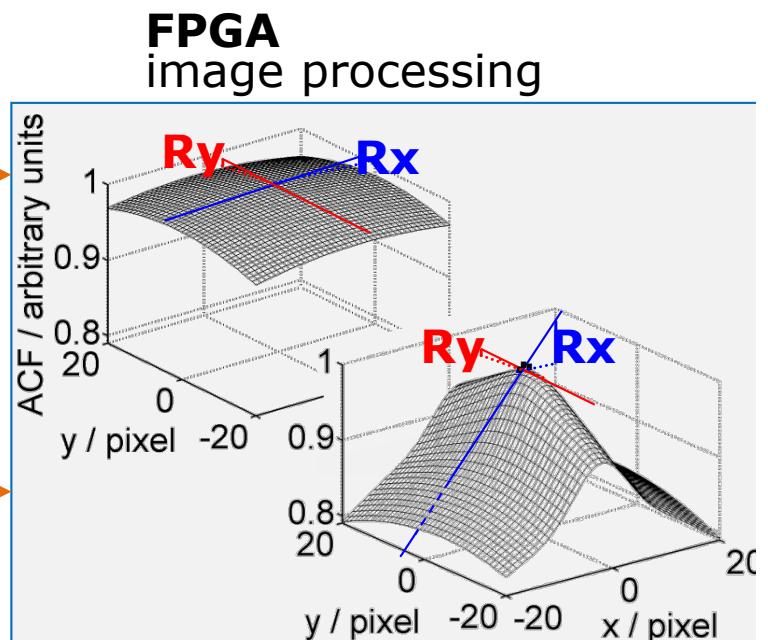
Scattered light measuring setup



- optical fiber as spatial filter
- wedged beam splitter plate
- camera, 2 MPixel, 340 FPS
- roughness dependent partially developed speckle patterns



- FPGA → direct autocorrelation function (ACF)
- optical roughness parameters Rx, Ry (slope of ACF)



Data processing

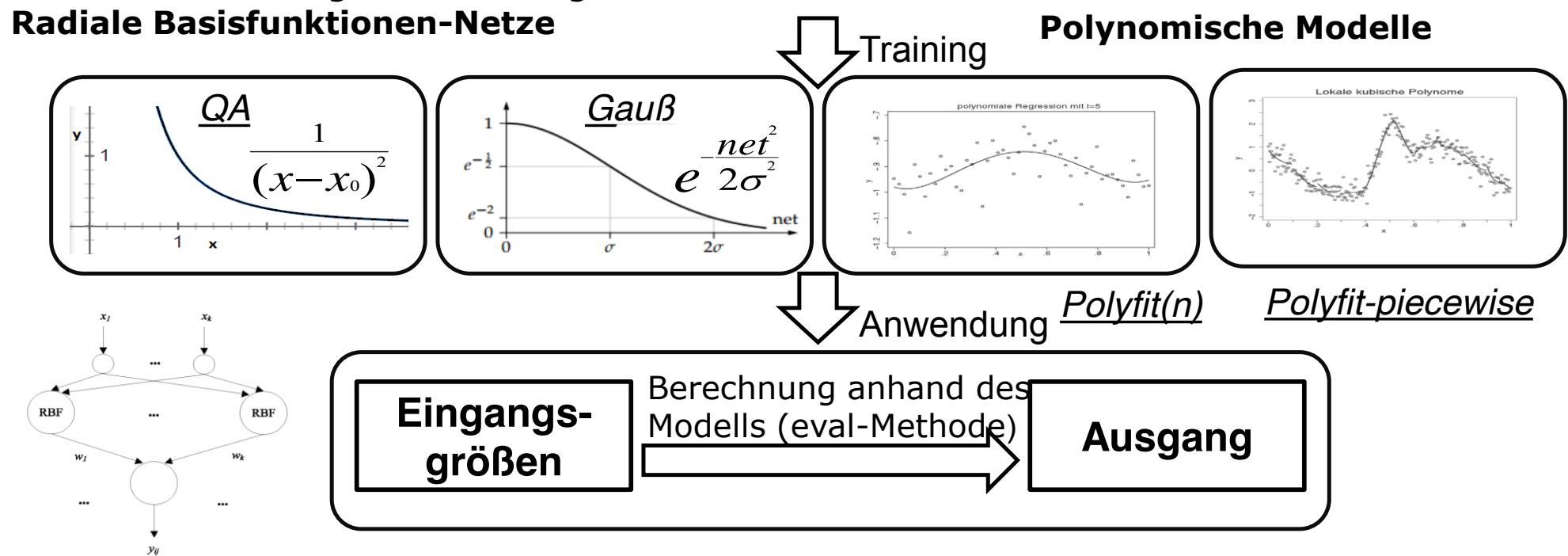
- mean pixel intensity value from previous image
- software-based image processing (fallback, test, ...)

FPGA-based approach

Automatisiertes Messen und Steuern

- Beispiel Laserschmelzen -

- Aufgabe: fast, adaptive control strategy (Schnelle Adaptive Regelung)
- Anwendung: Laserschmelzprozess mit zahlreichen Einflussgrößen
- Lösung: Integration der Regelung mit den Methoden der künstlichen Intelligenz (KI) in eine industrielle Steuerungsumgebung SPS
→ Übertragung der Funktionalität auf ein Reaction Modul (FPGA basiert) zur Verringerung der Zykluszeit, Verbesserung der Ausführungsgeschwindigkeit und Genauigkeit der Ausgaben



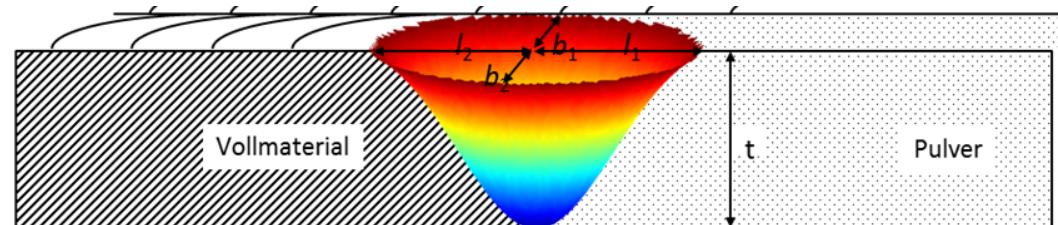
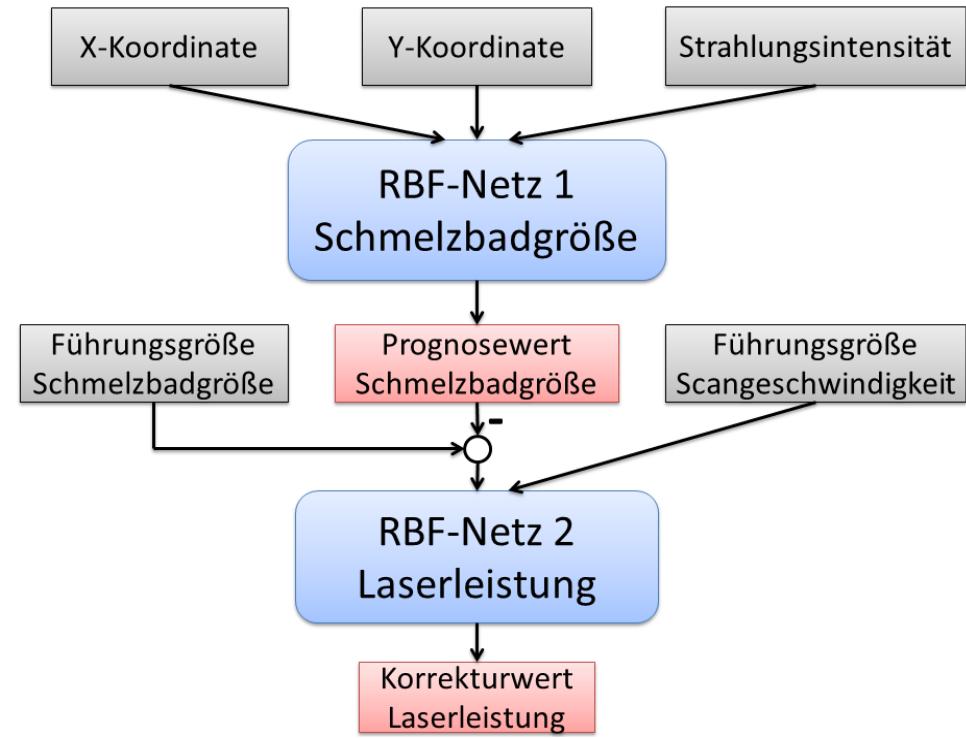
Automatisiertes Messen und Steuern

- Messtechnische Lösung -

Der Laserschmelzprozess

Bestimmung der **Schmelzbadgröße**
(X- und Y-Koordinaten und
Strahlungsintensität)

Bestimmung der **Laserleistung**
(Schmelzbadgröße, Differenz zwischen
Ist-/Sollwert und Scangeschwindigkeit)



Automatisiertes Messen und Steuern

- Messtechnische Lösung -

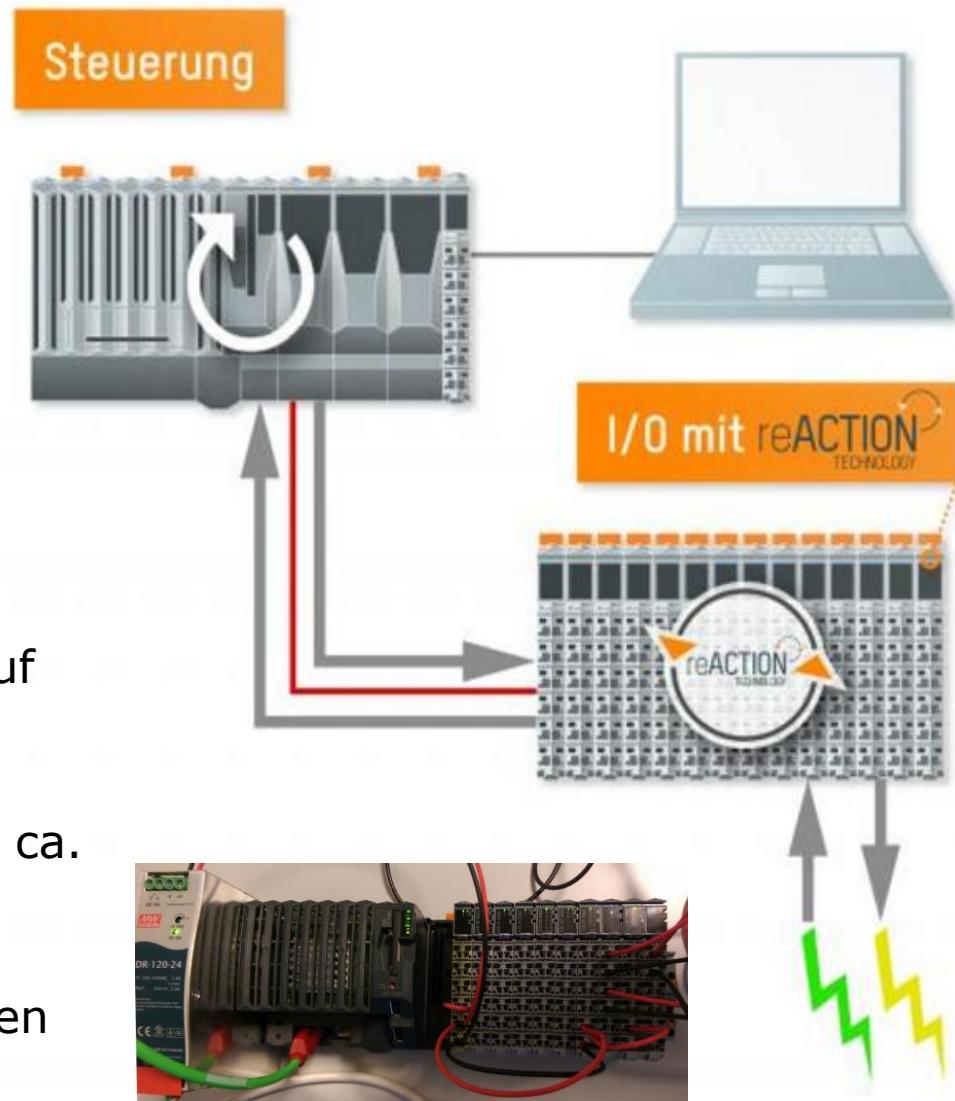
Realisierung der SPS-Programme

SPS + Reaction Modul:

- FPGA-Basis
- Direkte Verbindung mit der SPS
- Eigenständige Abarbeitung des zugeordneten Programms
- Programmierung im Funktionsbaustein-Editor
- Zykluszeit abhängig von der Komplexität (minimal 1µs)

Ziel: Berechnung des Prognosewertes auf dem Reaction Modul

- Die Gesamtzeit zur Generierung des Ausgangssignals sinkt auf insgesamt ca. 15-35µm
- Einschränkungen der Genauigkeit können mit Aufwand minimiert werden



Quelle: B&R

Automatisiertes Messen und Steuern

- Messtechnische Lösung -

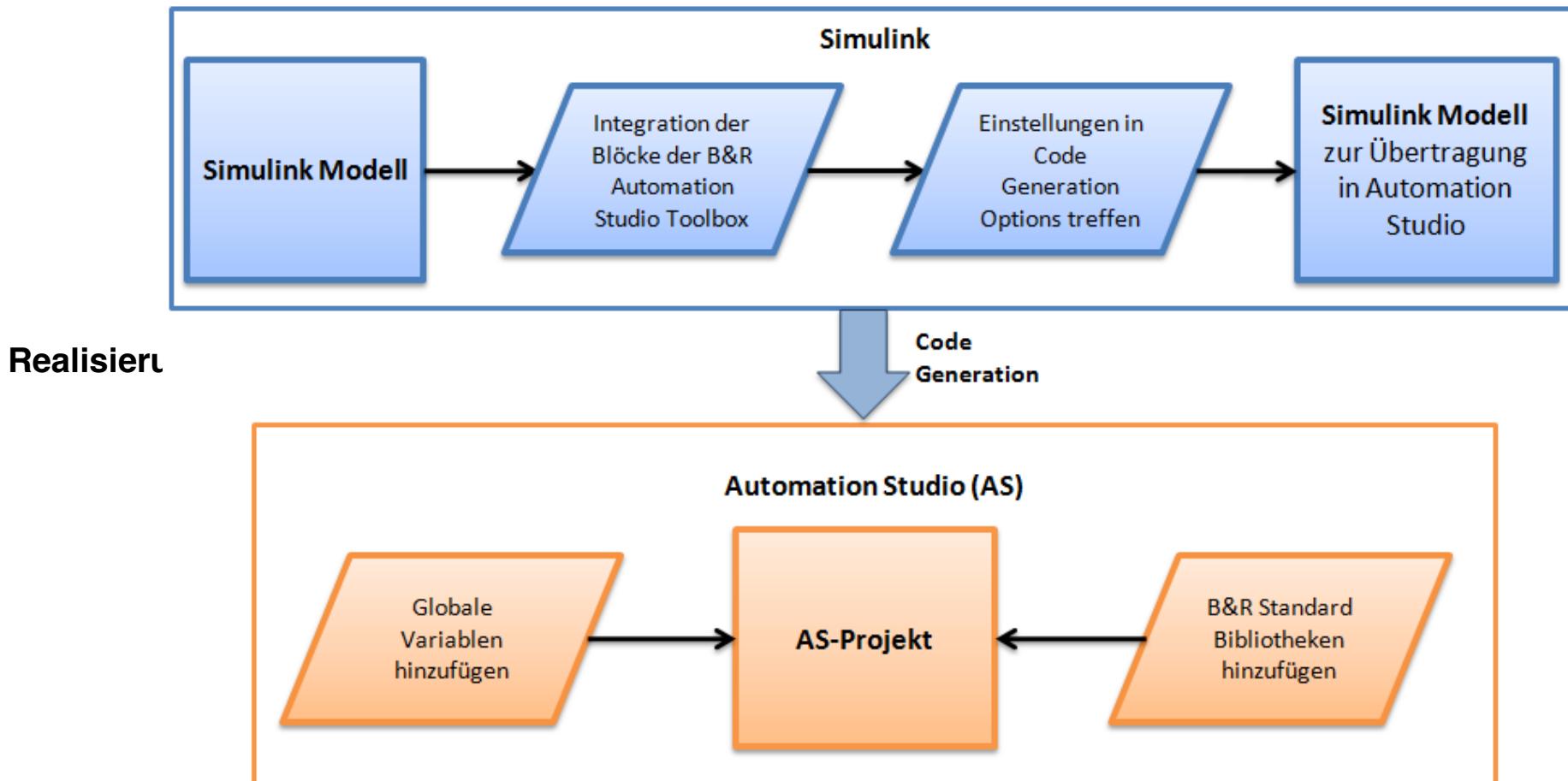
Einschränkungen auf dem Reaction Modul

Einschränkung	Auswirkung	Lösung
Keine Fließkommazahlen	Genauigkeit sinkt	Zusätzliche Divisionsblöcke, Datenbereich optimieren
Eingeschränkte Datenübertragung	Manuelle Einstellungen, kein automatisches Update (3 Eingänge)	Internen Speicher verwenden, Funktionalität aufteilen
Verzögerungen bei Datenübertragungen zwischen zwei Reaction Modulen	Zykluszeit steigt	Reaction Modul mit mehreren Eingängen verwenden
Maximaler Datenbereich DINT	Genauigkeit sinkt	Reaction Programmablauf ändern, zusätzliche Divisionsblöcke verwenden

Automatisiertes Messen und Steuern

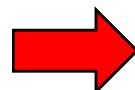
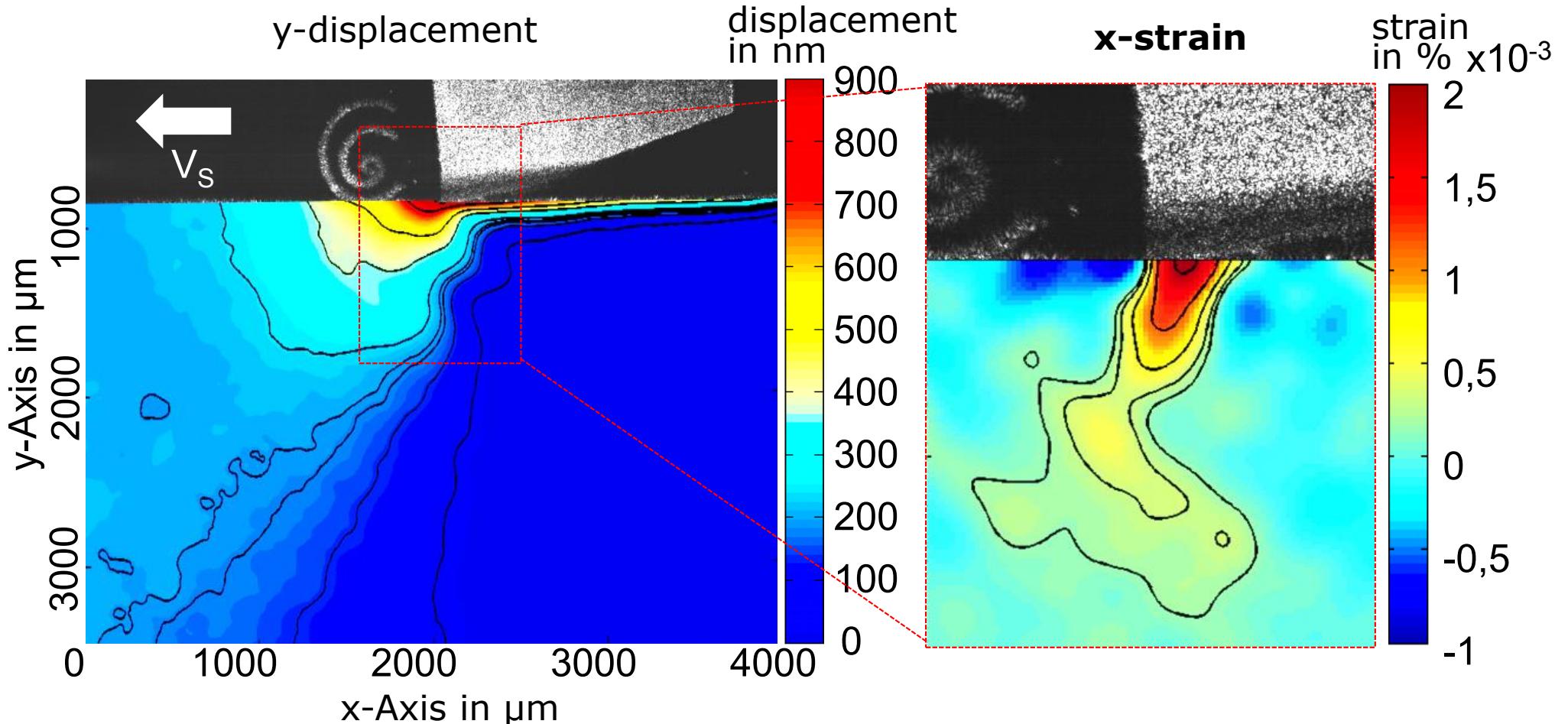
- Messtechnische Lösung -

Überführung der Simulink-Modelle auf die SPS mit der
Toolbox „B&R Automation Studio Target for Simulink“



Automatisiertes Messen und Steuern

- In Process Messtechnik beim Spanabheben -



High spatial and temporal resolution of measured deformation and strain fields even in dynamic processes

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 **Automatisiertes Messen und Steuern**
- Ü5 Automatisiertes Messen und Steuern
- V8 Speicherprogrammierbare Steuerungen
- Ü6 Übungen und Musterklausuren

Einführung in die Automatisierungstechnik

Studiengang: Produktionstechnik, Systems Engineering

- Vorlesung 08 -

Prof. Dr.-Ing. habil. Andreas Fischer
Dr.-Ing. Gerald Ströbel



Bremer Institut für
Messtechnik, Automatisierung
und Qualitätswissenschaft

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 **Speicherprogrammierbare Steuerungen**
- Ü6 Übungen und Musterklausuren

Speicherprogrammierbare Steuerungen

- Wiederholung - Industrielle Steuerungen -

<ul style="list-style-type: none">• SPS Speicherprogrammierbare Steuerungen 	<ul style="list-style-type: none">• RC Robotersteuerungen 	<ul style="list-style-type: none">• CNC Computerized Numerical Control 	<ul style="list-style-type: none">• AS Automatisierungssysteme• (spezialisierte Prozessrechner)
<ul style="list-style-type: none">• Steuerung und Überwachung eines Prozesses- von Maschinen oder AnlagenEchtzeit Bus Systeme (Feldbus, PROFIBUS)Dezentralisierung, WLAN, Mobilfunk, RFID (Industrie 4.0)	<ul style="list-style-type: none">• Bewegungsprogramm• Positionierung• Bahnsteuerung (3-D)• Ablauffolge LAN (Ethernet)Echtzeit Bus Systeme (Feldbus, PROFIBUS)Dezentralisierung, WLAN, Mobilfunk, RFID (Industrie 4.0)	<ul style="list-style-type: none">• Bearbeitungsprogramme• Konturen (2-D) / Bahnen 3D, mehrere Achsen• Bearbeitungskenngrößen (überwachen und steuern)• LAN, Feldbusse, Echtzeitkommunikation	<ul style="list-style-type: none">• Prozesssteuerung• Überwachung• Erkennung• Identifizieren• Klassifizieren• LAN FeldbusseEchtzeitkommunikation, WLAN, Mobilfunk, NFC, RFID, (Industrie 4.0)
Einmal-Programmierung SPS Programm	<ul style="list-style-type: none">• Wiederholte Anwenderprogrammierung• Bewegungsprogramm	<ul style="list-style-type: none">• Wiederholte Anwenderprogrammierung• Bearbeitungsprogramm	<ul style="list-style-type: none">• Einmal-Programmierung• z.B. Menügeführt• „Funktionsorientiertes“ Programm

Speicherprogrammierbare Steuerungen SPS

- Entwicklung -

In den **60er Jahren** wurden für **Automatisierungsaufgaben** systemspezifische elektrische Schaltungen und Programme entwickelt.

Die Funktion wurde entweder mit fester Verdrahtung implementiert (Stromlaufplan) oder durch Rechner gesteuert, die mit aufwendigen Echtzeitprogrammiersprachen programmiert wurden.

Erst seit den **70er Jahren** werden **SPS** für Automatisierungsaufgaben verwendet.

PLC Programmable Logic Controller

Geht zurück auf Odo Struger (US Fa. Allen Bready) und Richard Morley (Fa. Modicon, später AEG-Modicon) der 1969 die Modicon084 als „solid-state sequential logic solver“ = halbleiterbasierendes sequentielles Logiksystem

Im Vordergrund der fortschreitenden Automatisierung der Automobilindustrie in den **80er Jahren** und der rasanten Entwicklung der Computertechnik (Mikroprozessoren) standen: Prozess-Steuerung und -transparenz, Diagnose, Prozessprotokollierung, graphischen Bedienerführung und Dezentralisierung sowie Energieeffizienz

Speicherprogrammierbare Steuerungen SPS

- Normen -

Ein entscheidender Schritt wurde mit der Festlegung und Akzeptanz der Norm **DIN EN 61131** (Programmable Controller) für die Standardisierungs-Bemühungen in der Automatisierung getan.

Der Standard vereint die Erfahrungen der SPS Programmierung, die teilweise in nationalen Standards bereits festgelegt waren. Dazu gehören:

- Grafcet (Frankreich) bzw. IEC 848
- DIN 40719 (D)
- NEMA ICS-3-304 (USA)
- DIN 19239 (D)
- VDI 2880

Speicherprogrammierbare Steuerungen SPS

- Entwicklung -

Eine **Speicherprogrammierbare Steuerung (SPS)** ist eine Steuerung, die flexibel an die Steueraufgabe angepasst werden kann, da ihre Funktion durch ein variables Programm festgelegt wird. Durch die modulare Bauweise und die nachträgliche Austauschbarkeit und Erweiterbarkeit des Programms ist eine SPS einfach an veränderte Anforderungen anzupassen.

Die SPS-Norm **DIN EN 61131-3** hat den Programmierstandard durch Vorgabe von grundlegenden Konzepten definiert. Dazu zählen:

- Datentypen und Variablenkonzeption
- Programmorganisationskonzept
- Programmier-/Fachsprachenkonzept
- Task-Konzept

Als Programmiersprachen stehen zur Verfügung:

Die **Anweisungsliste (AWL)**, eine den Befehlen der speicherprogrammierbaren Steuerung angepasste Assemblersprache.

Strukturierter Text (ST)

Der **Kontaktplan (KOP)**, eine graphische Darstellung, deren Symbole und Anordnung an Stromlaufpläne ähnlich sind.

Funktionsbausteine (FBS), die funktionelle Zusammenhänge der Schaltung unabhängig von ihrer Realisierung darstellt.

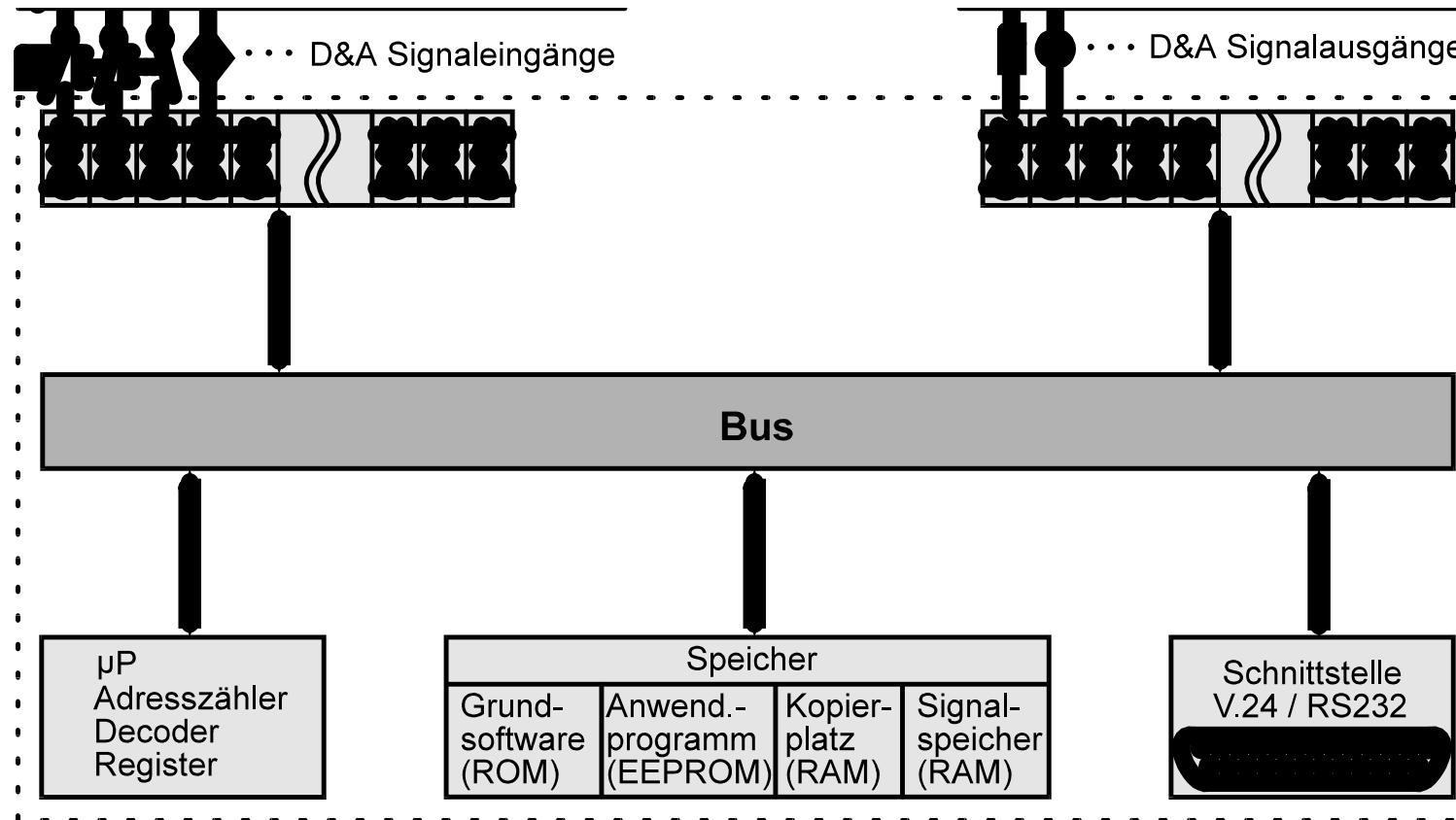
Die **Ablaufsprache (AS)**, die sich an Aufgabenstellungen der Prozesstechnik orientiert.

Speicherprogrammierbare Steuerungen SPS

Englisch		Deutsch		
Abk.	Bezeichnung	Abk.	Bezeichnung	Hinweise
IL	Instruction List	AWL	Anweisungsliste	Vergleichbar mit Assembler
LD	Ladder Diagram	KOP	Kontaktplan	Vergleichbar mit einem Elektro- Schaltplan der um 90° gedreht ist
FBD	Function Block Diagram	FBS	Funktionsbaustein-Sprache	Teilweise (insbesondere bei Siemens STEP 7) auch als FUP (Funktionsplan) bekannt.
SFC	Sequential Function Chart	AS	Ablaufsprache	eine Art Zustandsdiagramm , bei STEP 7 als S7 GRAPH bekannt. Die IEC 61131-3:2003 sieht den SFC als eine Weiterentwicklung von Grafset nach EN 60848 .
ST	Structured Text	ST	Strukturierter Text	angelehnt an Hochsprachen, bei STEP 7 als SCL (Structured Control Language) bezeichnet.
CFC	Continuous Function Chart	CFC	Continuous Function Chart	<i>Nicht Teil der IEC 61131-3:</i> Vollgrafisches Projektierwerkzeug, speziell für Regelungen und Verriegelungen

Speicherprogrammierbare Steuerungen SPS

- Klassisch modulare Aufbau einer SPS Beispiel Siemens S5 -



Die wichtigsten Module sind die Zentraleinheit, das Eingangsmodul, das Ausgangsmodul (Optokoppler), und die Stromversorgung.

Speicherprogrammierbare Steuerungen SPS

- Aufbau und Arbeitsweise einer SPS -

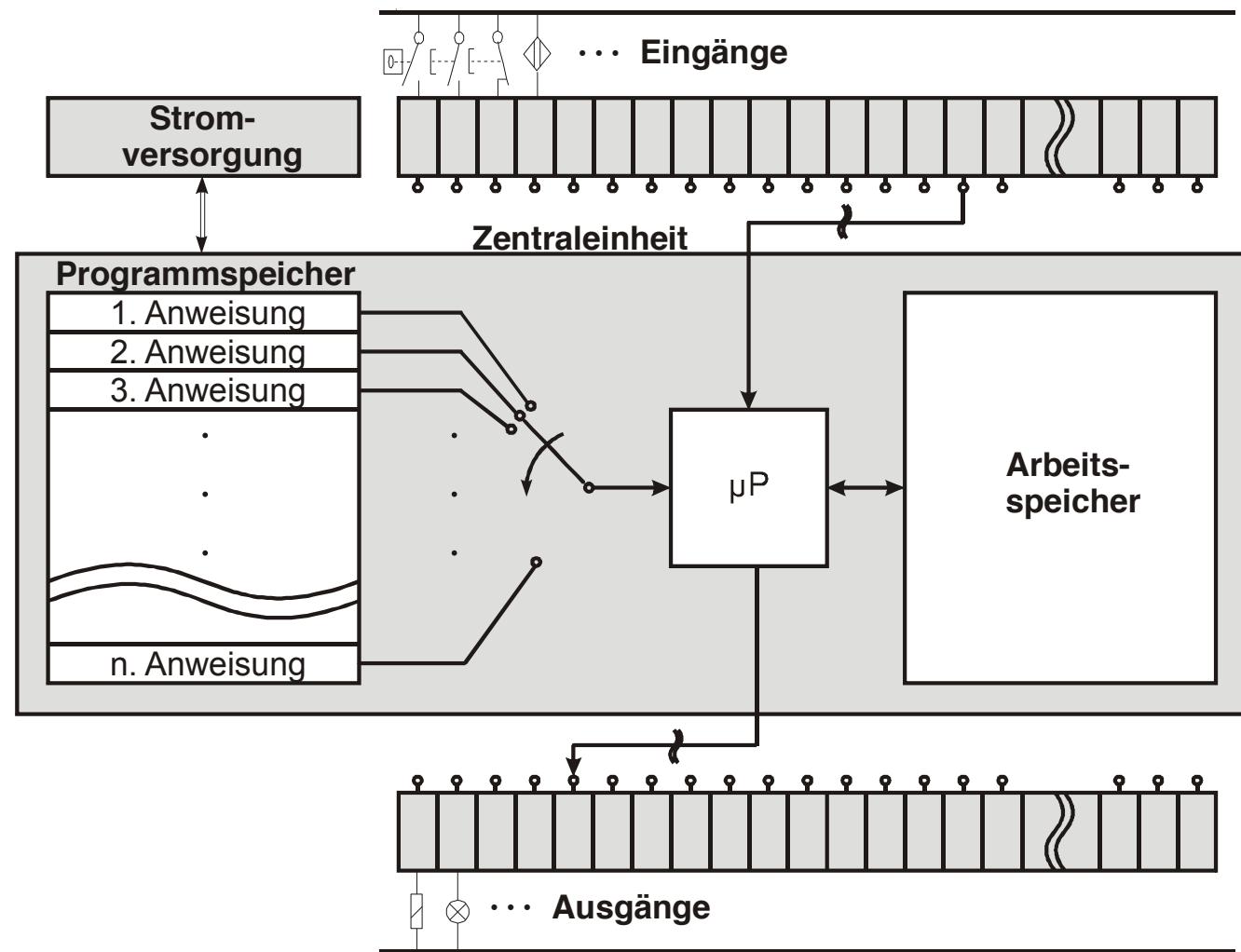
Digitale Eingangssignale:

- Druckknopfschalter, - Geber, - Endschalter,
- digitale Messwertgeber
- weitere:
 - Analoge Eingänge, Zähler, spezielle Signal-Geber

Ausgänge:

- galvanisch getrennt (Optokoppler),
- Relais, Transistoren, Triac-Schaltungen*,

Ein-/Ausgabebaugruppen speziell wählbar nach Leistungsbedarf und Anzahl der Ein-/Ausgänge



Speicherprogrammierbare Steuerungen SPS

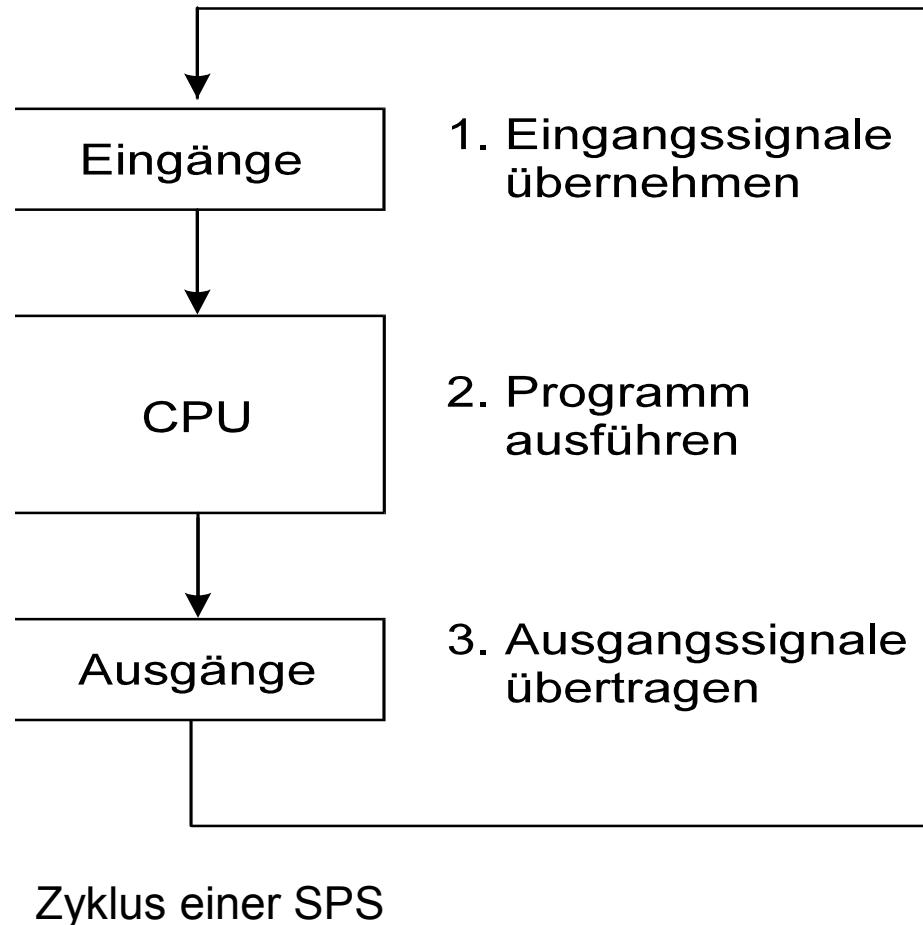
- Funktionsprinzip Zyklische Arbeitsweise der SPS -

Prozessabbild (Eingänge)
speichern

Alarmeingänge überwachen

Anweisungen decodieren
und ausführen
Ergebnissen speichern

Prozessabbild (Ausgänge)
ausgeben



Typische Zykluszeiten von SPS sind 5 bis 10 ms pro 1000 Anweisungen
(schnelle 0,1ms)

Speicherprogrammierbare Steuerungen SPS

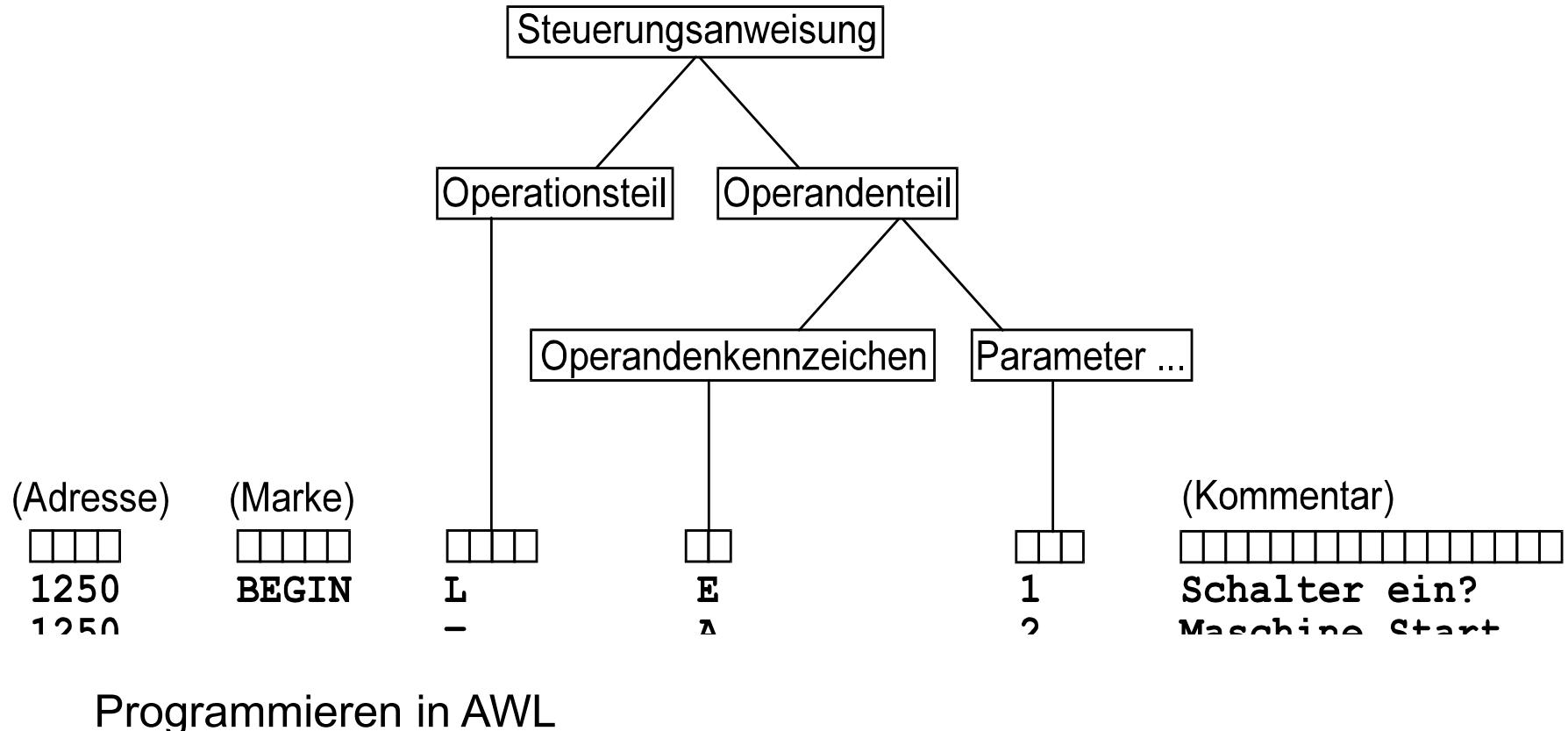
- Leistungsmerkmale -

Die charakteristischen **Prozessmerkmale** und **Zeitkonstanten** bestimmen die notwendige Leistungsfähigkeit der SPS bezüglich der sog. **maximalen Zykluszeit** zur Bearbeitung einer definierten max. Anzahl von Anweisungen und

wiederum bestimmt die **Komplexität der Steuerungsaufgabe** den Umfang des Anwenderprogramms.

Speicherprogrammierbare Steuerungen SPS

- Programmieren in AWL -



Speicherprogrammierbare Steuerungen SPS

- Programmierverfahren -

Fünf Programmierverfahren sind weit verbreitet

- Diese Programmierverfahren wurden entwickelt, um der Denkweise bestimmter Anwender, wie etwa der eines Elektronikmeisters, möglichst nahe zu kommen
- Die meisten Steuerungen lassen sich mit mindestens den ersten drei Verfahren programmieren, die gleichwertig sind
- Eingeteilt werden können diese nach ihrer Darstellungsart in:
 - **alphanumerische Darstellung:** **Anweisungsliste (AWL)**
 - **graphische Darstellung**
 - zustandsabhängig: **Kontaktplan (KOP)**
 - funktionsabhängig: **Funktionsplan (FUP)**
 - (weiterentwickelt zum **FBS**)
Ablaufplan (AP)
(weiterentwickelt zur **AS**)
 - **textueller Darstellung:** **Strukturierter Text (ST)**
Beispiel: E:=(A OR B) & (C OR D) ; Kommentar

Speicherprogrammierbare Steuerungen SPS

- AWL -

- Mnemonische Darstellung der Operationen in der AWL

Operationsteil	Mnemonische Darstellung deutsch	Mnemonische Darstellung englisch
Oder	O	O
Und	U	A
Negation	N	N
Exklusiv Oder	XO	XO
Laden	L	L
Speichern	=	=
Setzen	S	S
Rücksetzen	R	R
Addieren	ADD	ADD
Subtrahieren	SUB	SUB
Multiplizieren	MUL	MUL
Dividieren	DIV	DIV
Nulloperation	NOP	NOP
unbedingter Sprung	SP	JP
bedingter Sprung	SPB	JC
Programmende	PE	EP
Klammer auf	((
Klammer zu))

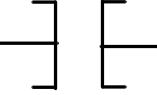
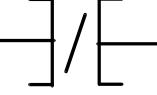
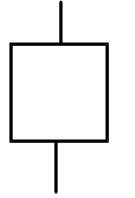
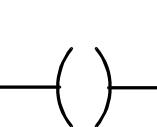
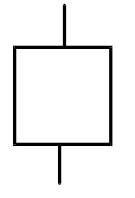
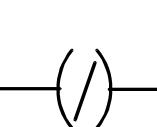
Speicherprogrammierbare Steuerungen SPS

- Mnemonische Darstellung der Operanden in der AWL -

Operandenkennzeichen	Mnemonische Darstellung deutsch	Mnemonische Darstellung englisch
Eingang	E	I
Ausgang	A	O
Merker	M	M
Konstante	K	K
Zeitglied	T	T
Zähler	Z	C
Programmbaustein	P	P
Funktionsbaustein	F	F

Speicherprogrammierbare Steuerungen SPS

- Kontaktplan -

Stromlauf- plan	Kontakt- plan	Beschreibung
		Schließerkontakt, Weitergabe des Eingangssignals
		Öffnerkontakt, Weitergabe eines LOW-Signals
		Relais mit Schließer, Weitergabe des Ausgangssignals
		Relais mit Öffner, Weitergabe eines LOW-Signals

- Die Äquivalenz von Symbolen im Stromlaufplan und im Kontaktplan

Speicherprogrammierbare Steuerungen SPS

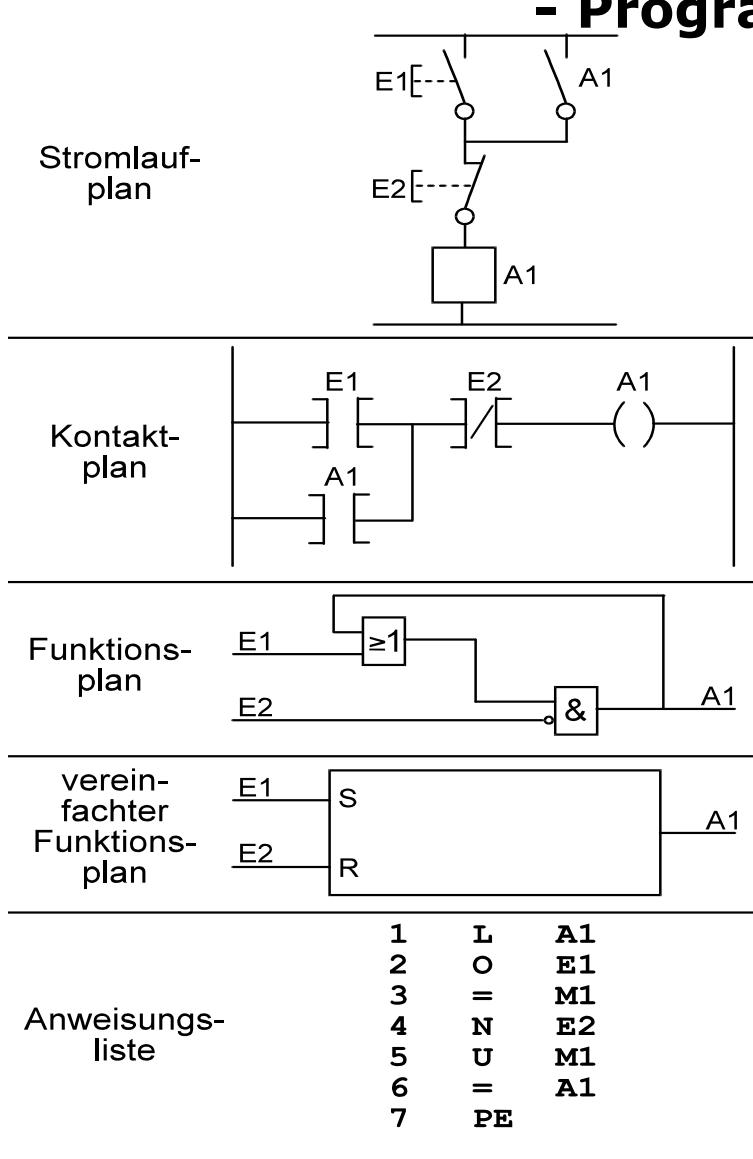
- Funktionsplan I -

Funktion	Symbol	Äquivalent im KOP
UND		Reihenschaltung von Eingängen
ODER		Parallelschaltung von Eingängen
NICHT		Negation des Eingangs

- Die logischen Funktionen und ihr Symbol im Funktionsplan

Speicherprogrammierbare Steuerungen SPS

- Programmieren - Beispiel -

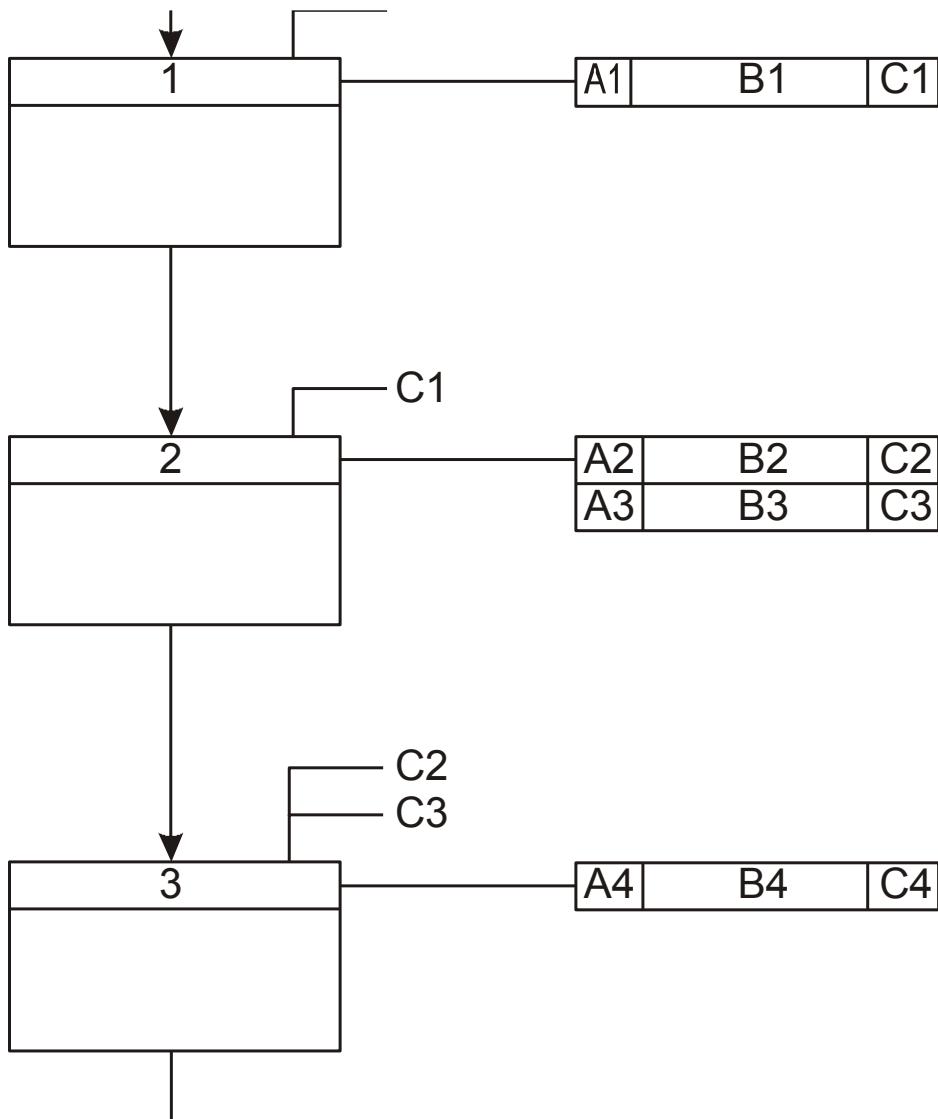


1	L	A1
2	O	E1
3	=	M1
4	N	E2
5	U	M1
6	=	A1
7	PE	

- Verschiedene Programmierverfahren für eine SPS
- E1 setzt M1 und damit A1
- E2 setzt M1 zurück und damit A1

Speicherprogrammierbare Steuerungen SPS

- Aufbau einer Ablaufsteuerung -



Feld A zeigt die Art des Befehls:

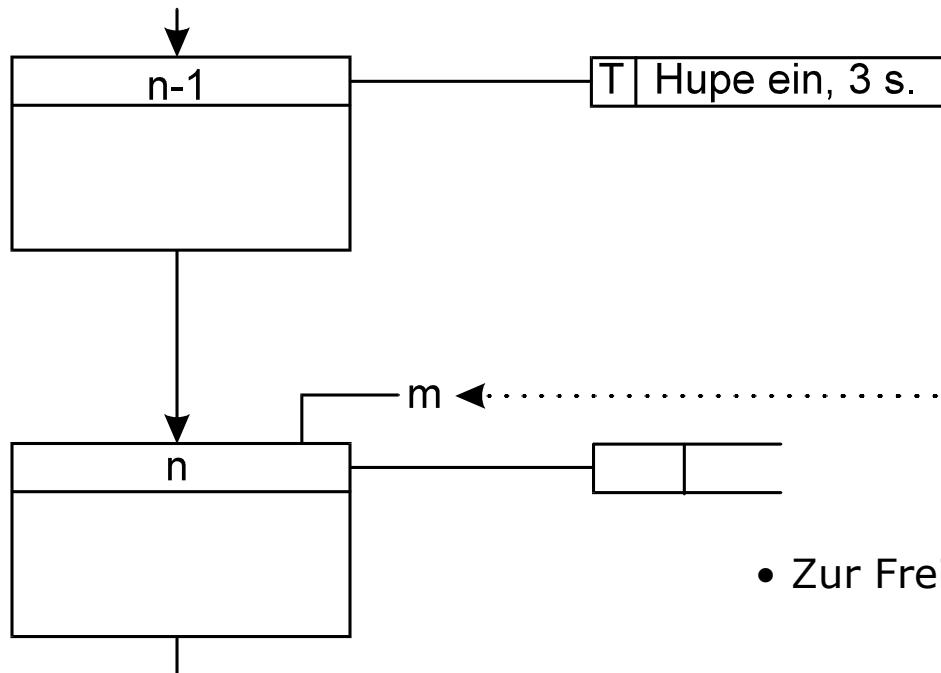
- S gespeichert
- D verzögert
- T zeitlich begrenzt
- SH gespeichert, auch bei Stromausfall
- NS nicht gespeichert
- SD gespeichert und verzögert
- NSD nicht gespeichert und verzögert
- ST gespeichert und zeitlich begrenzt

Feld B beschreibt die Wirkung des Befehls.

Speicherprogrammierbare Steuerungen SPS

- Ablaufsteuerung -

- **Feld C** kennzeichnet durch eine Ziffer das Signal, das bei der Terminierung des Befehls gesetzt wird. Nachfolgende Schritte, die erst ausgeführt werden sollen, wenn dieser Befehl beendet wurde, benutzen dieses als Freigabe-Signal (vgl. Abb. 9-11). Wenn dieses Signal nicht benötigt wird, kann das Feld auch weggelassen werden.
- Schritt n wird erst dann aktiviert, wenn die folgende Bedingung erfüllt ist:



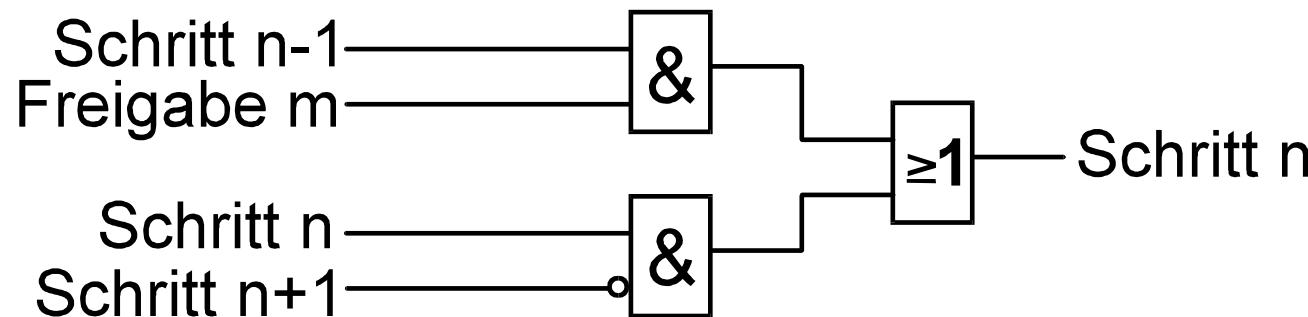
- Zur Freigabe des nachfolgenden Ablaufschritts

Speicherprogrammierbare Steuerungen SPS

- Schrittsymbol -

- Für jedes Schrittsymbol kann es ein oder mehrere Befehlssymbole geben. Das Befehlssymbol kann auch weggelassen werden, z. B. wenn gewartet werden soll. Die Abb. zeigt das Äquivalent eines Schrittsymbols im Funktionsplan.

$$\text{Schr.}_n = \text{Schr.}_{n-1} \wedge \text{Freigabe}_m \vee \text{Schr.}_n \wedge \overline{\text{Schr.}_{n+1}}$$

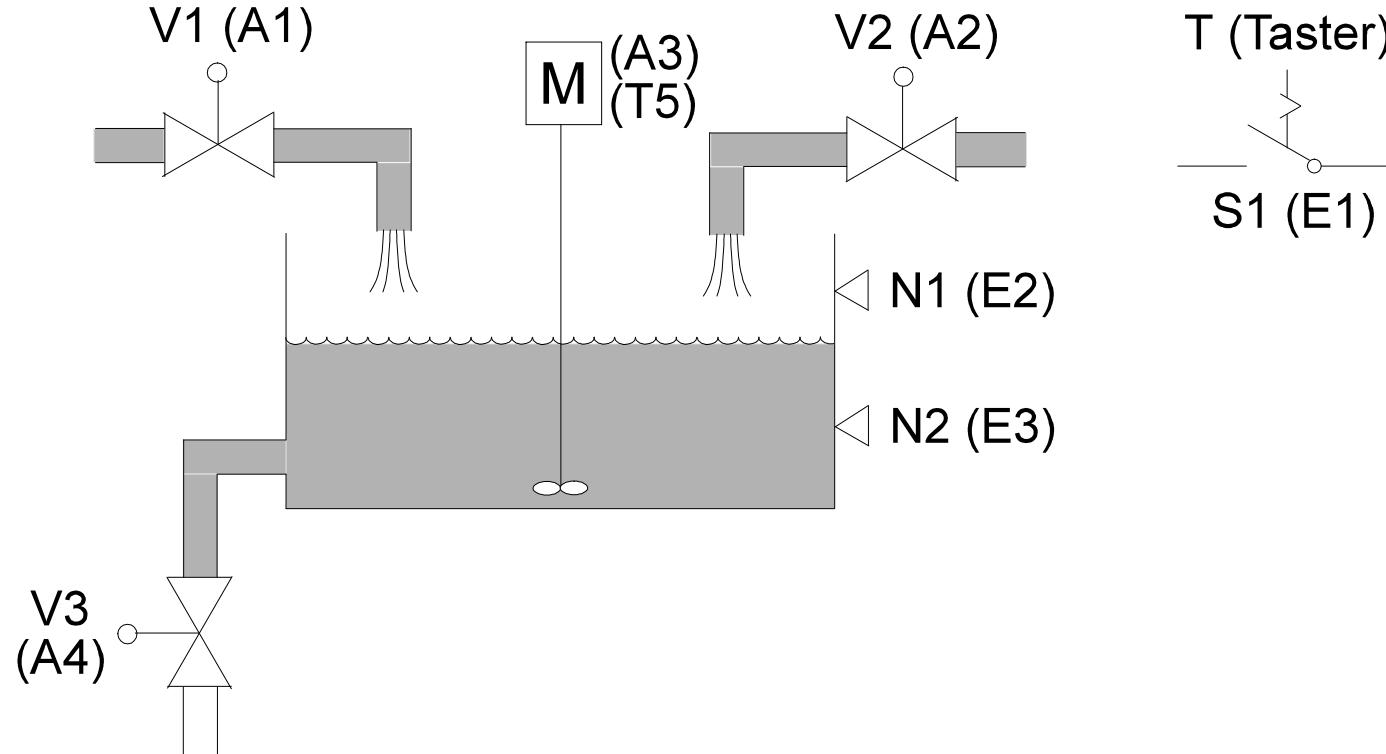


- Äquivalent eines Schrittsymbols im Funktionsplan

Speicherprogrammierbare Steuerungen SPS

- Beispiel -

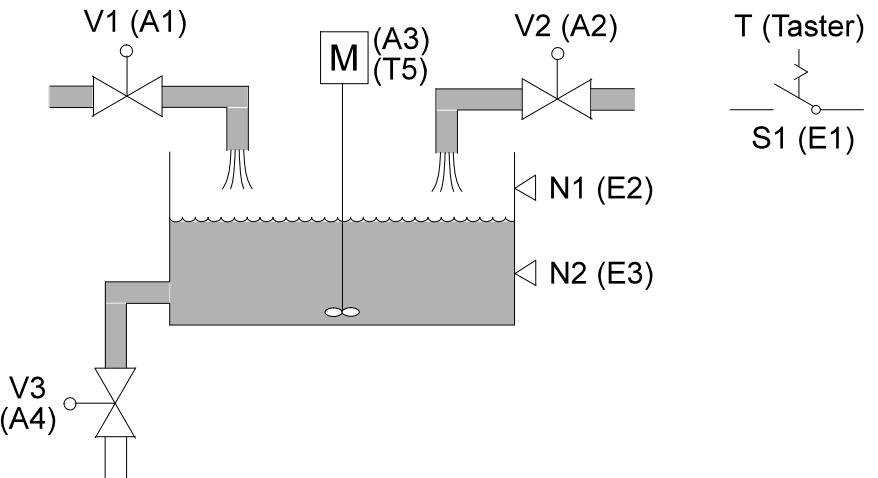
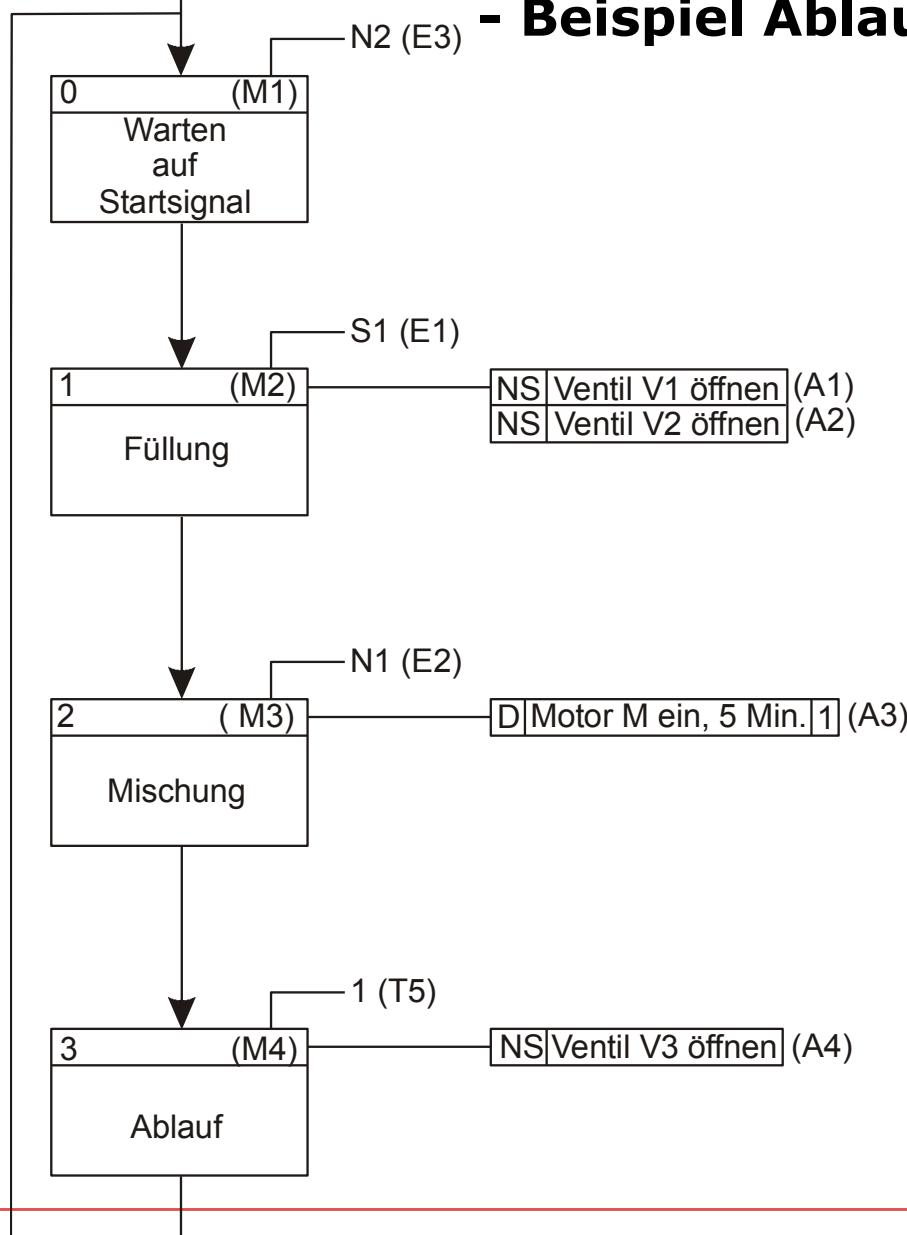
- Beispiel: Mischung zweier Flüssigkeiten in einem Behälter -



- Durch kurze Betätigung des Tasters S1 wird die Apparatur eingeschaltet. Danach wird S1 wieder geöffnet, der Mischvorgang hat begonnen. Der Mischvorgang geschieht nach folgenden Schritten:

Speicherprogrammierbare Steuerungen SPS

- Beispiel Ablaufsteuerung -



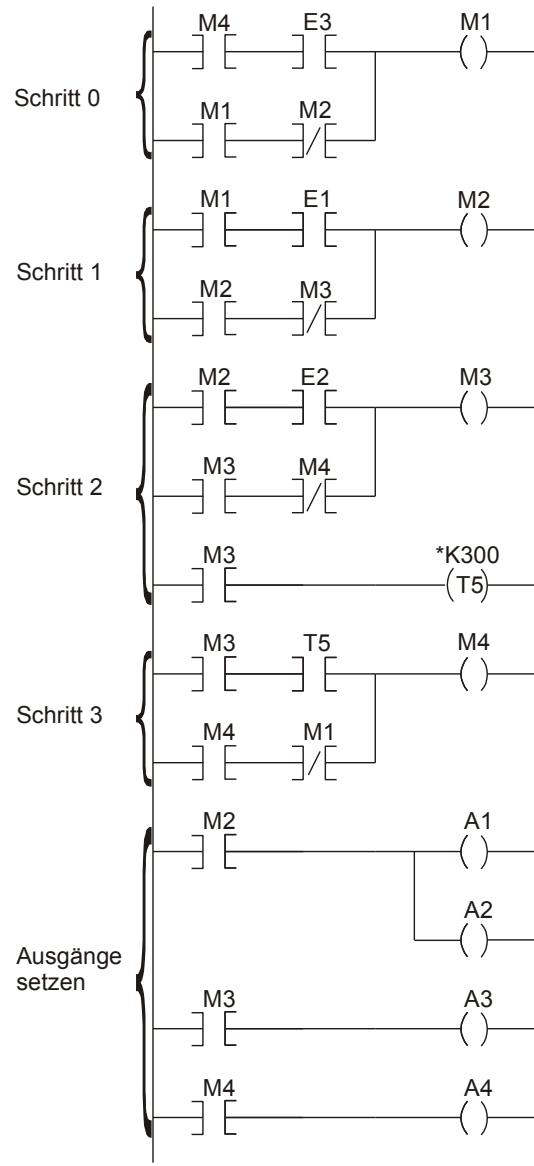
1. Schritt: Die Ventile V1 und V2 öffnen (bis N1)

2. Schritt: Die Mischung durch Maschine M dauert 5 Minuten

3. Schritt: Das Ventil V3 wird geöffnet (bis N2)

Speicherprogrammierbare Steuerungen SPS

- KOP -

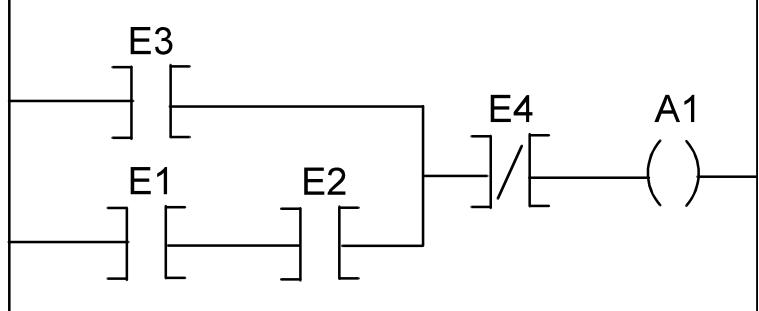


- Kontaktplan für das Mischen zweier Flüssigkeiten

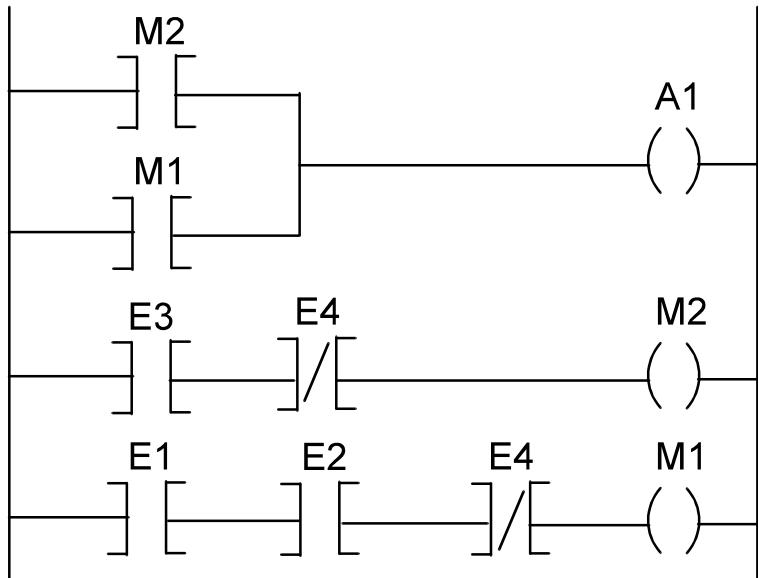
Speicherprogrammierbare Steuerungen SPS

- Merker -

Schaltung ohne Merker



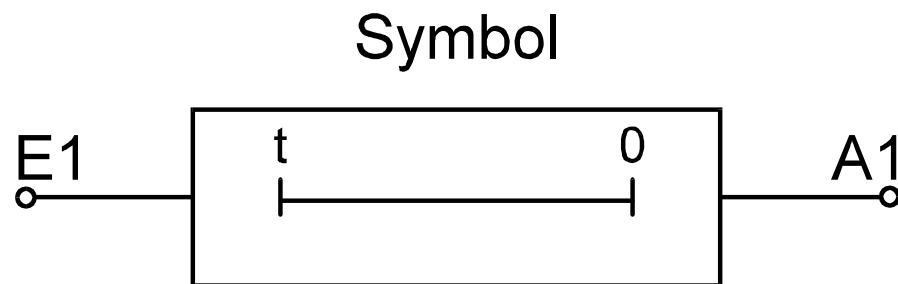
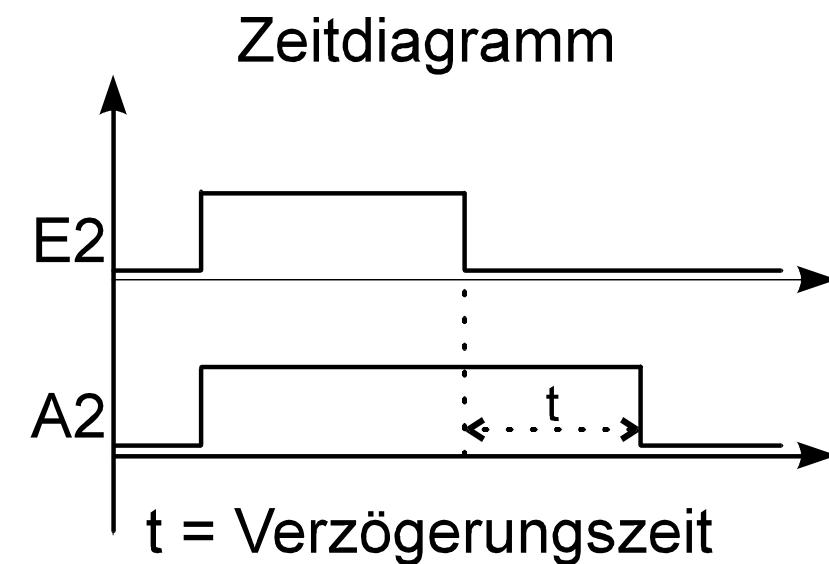
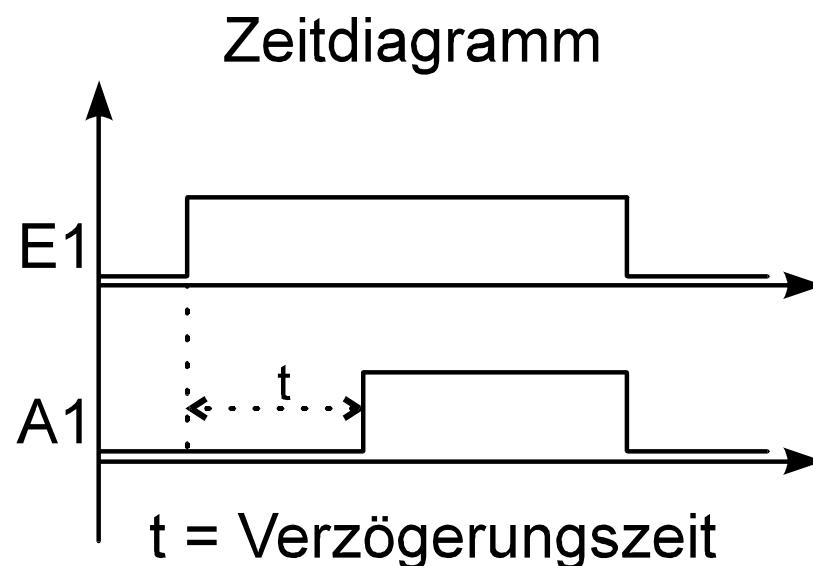
Schaltung mit Merker



Schaltung mit und ohne Merker

Speicherprogrammierbare Steuerungen SPS

- Merker -



•Abb. 9-18 Einschaltverzögerung

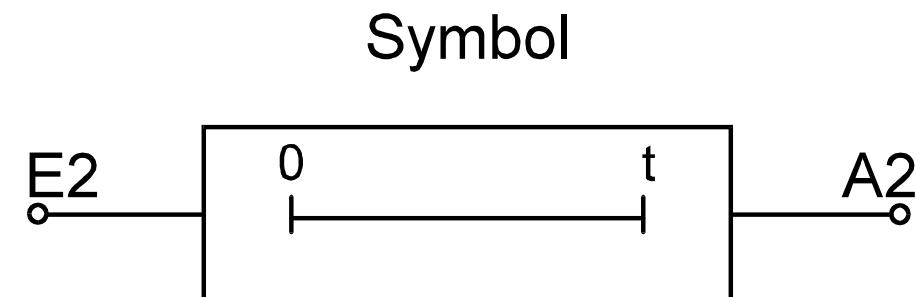
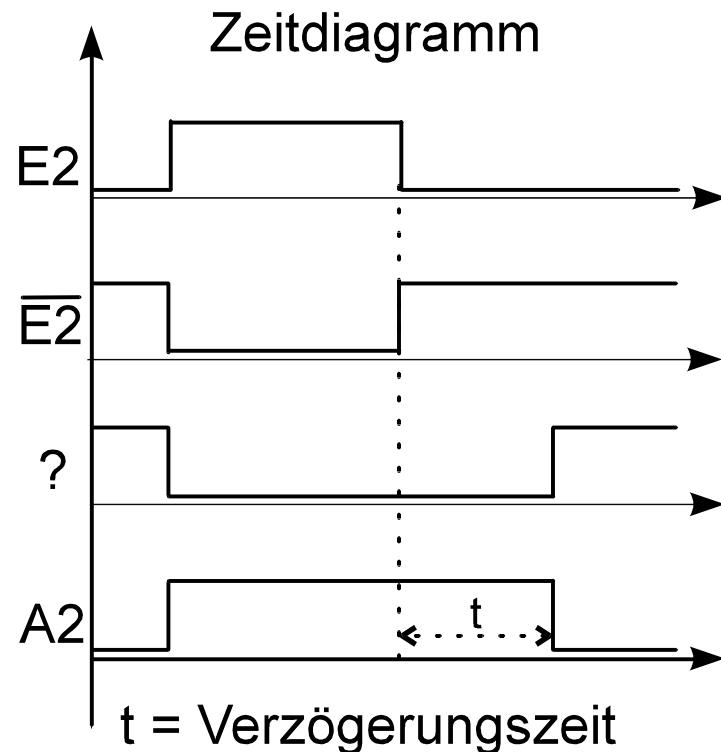


Abb. 9-17 Ausschaltverzögerung

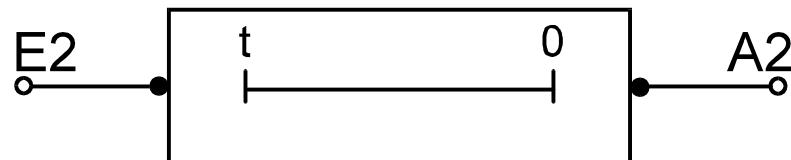
Speicherprogrammierbare Steuerungen SPS

- Ausschaltverzögerung -



Ausschaltverzögerung, realisiert mit einem Einschaltverzögerungsglied

Aufbau



Speicherprogrammierbare Steuerungen SPS

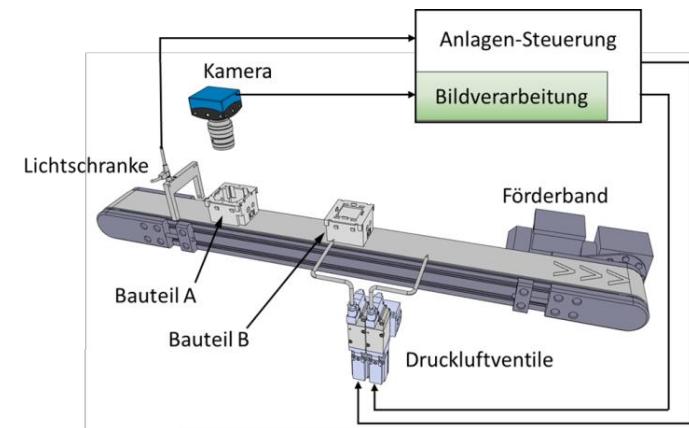
- Zähler -

Zählerfunktionen werden benötigt, um beispielsweise auf einem Förderband liegende Gegenstände zu zählen, die von einem Sensor erfasst werden.

Funktionsweise:

Die vom einem Sensor erzeugten Flanken (auf- oder absteigend) werden von einem Zählmodul gezählt. Sowohl die Zählrichtung als auch der Zielwert "K" (bzw. Startwert bei Rückwärtszählung) können programmiert werden. Wurde der Zielwert erreicht, schaltet der Ausgang "Q" ab.

Spezielle Zählmodule haben Ausgänge für binär kodierte Dezimalwerte (BCD)



Speicherprogrammierbare Steuerungen SPS

- Ausblick IEC 61131-

Stand und Ausblick

Der Standard **IEC 61131** besteht aus 6 Teilen und drei technischen Reports:

Teil 1: Allgemeine Information / General information (Umfeld, Begriffe, weitere relevante Standards Normen)

Teil 2: Ausrüstung und Testanforderungen / Equipment and test requirements (Hardwaremodell, mech. und elektr. Anforderungen)

Teil 3: Programmiersprachen / Programming languages (Datentypen, AWL, KOP, FBS, AS, ST) aktuell **Ausgabe 6-2014**

IEC TR 61131-4 Teil 4: Anwenderrichtlinien / User Guidelines (Systemauswahl, Prüfung vor der Installation, Inbetriebnahme und Wartung)

Teil 5: Kommunikation / Messaging service (Funktionalitäten zur Kommunikation und Mapping auf das MMS Protokoll, identisch mit ISO 9506-5) 11-2001

Teil 6: Funktionale Sicherheit aktuell 10-2013

Teil 7: Fuzzy Regelung / fuzzy control programming 11-2001

Speicherprogrammierbare Steuerungen SPS

- Ausblick IEC 61131 -

- **Technische Reports**
- Neben IEC TR 61131-4: User Guidelines
- IEC TR 61131-8: Implementierung und Anwendung der Programmiersprachen nach
- IEC TR 61131-9: Single-drop digital communication interface for small sensors and actuators (SDCI – oder **IO-Link**)
- Die **IEC 61131** konsolidiert die vorhandenen Dokumente und die seit 1992 entstandenen Ergänzungen

Speicherprogrammierbare Steuerungen SPS

- Forschungsgegenstand – Integrierte Messtechnische Lösung -

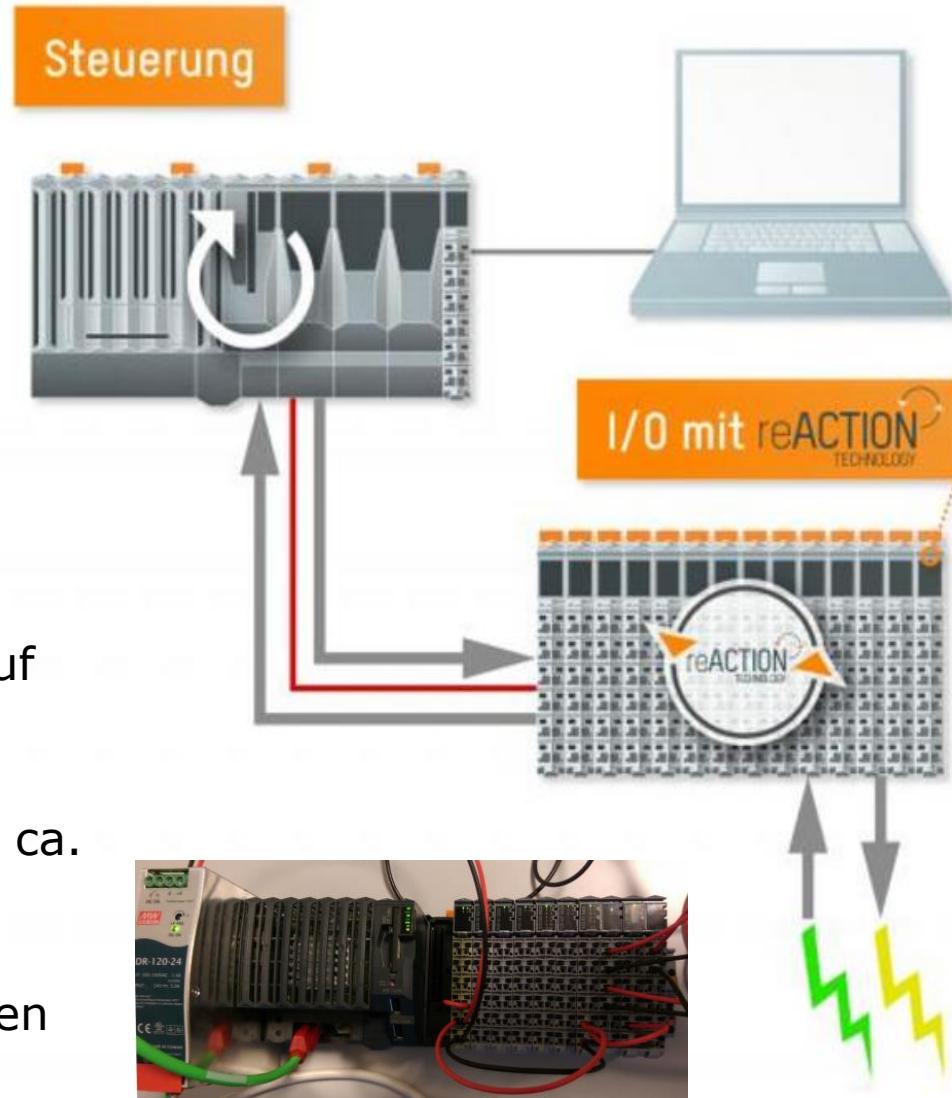
Realisierung der SPS-Programme

SPS + Reaction Modul:

- FPGA-Basis
- Direkte Verbindung mit der SPS
- Eigenständige Abarbeitung des zugeordneten Programms
- Programmierung im Funktionsbaustein-Editor
- Zykluszeit abhängig von der Komplexität (minimal 1µs)

Ziel: Berechnung des Prognosewertes auf dem Reaction Modul

- Die Gesamtzeit zur Generierung des Ausgangssignals sinkt auf insgesamt ca. 15-35µm
- Einschränkungen der Genauigkeit können mit Aufwand minimiert werden

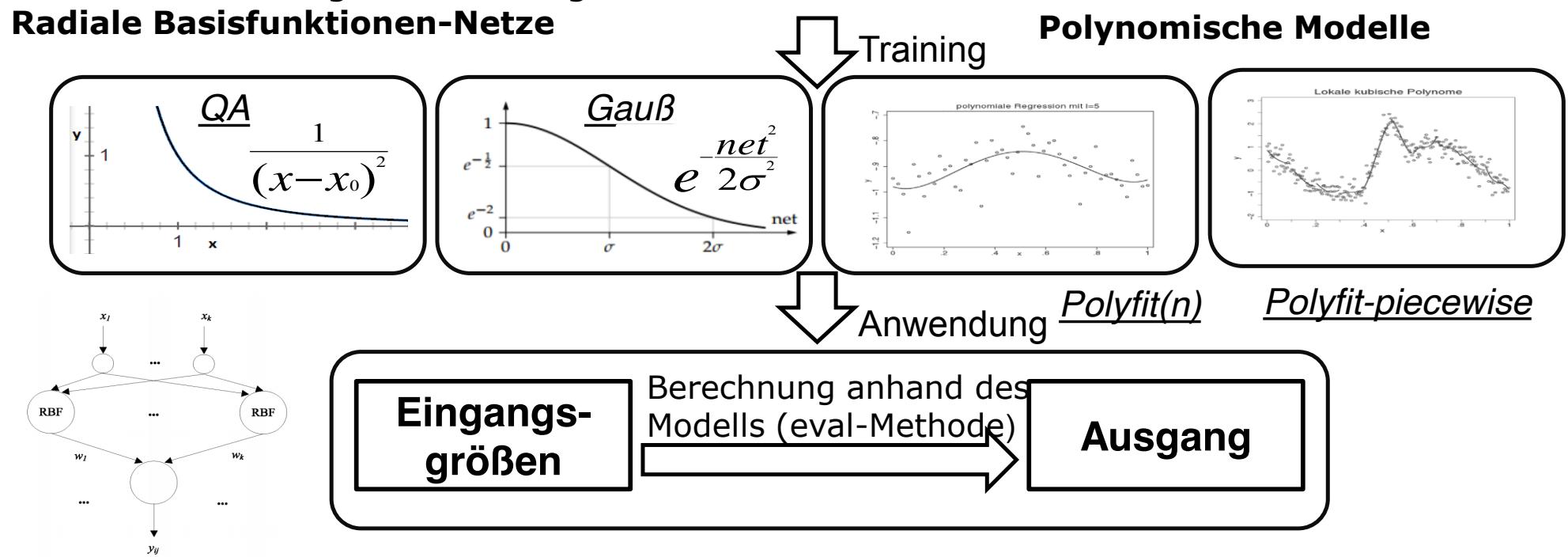


Quelle: B&R

Speicherprogrammierbare Steuerungen SPS

- Beispiel Laserschmelzen -

- Aufgabe: fast, adaptive control strategy (Schnelle Adaptive Regelung)
- Anwendung: Laserschmelzprozess mit zahlreichen Einflussgrößen
- Lösung: Integration der Regelung mit den Methoden der künstlichen Intelligenz (KI) in eine industrielle Steuerungsumgebung SPS
→ Übertragung der Funktionalität auf ein Reaction Modul (FPGA basiert) zur Verringerung der Zykluszeit, Verbesserung der Ausführungsgeschwindigkeit und Genauigkeit der Ausgaben



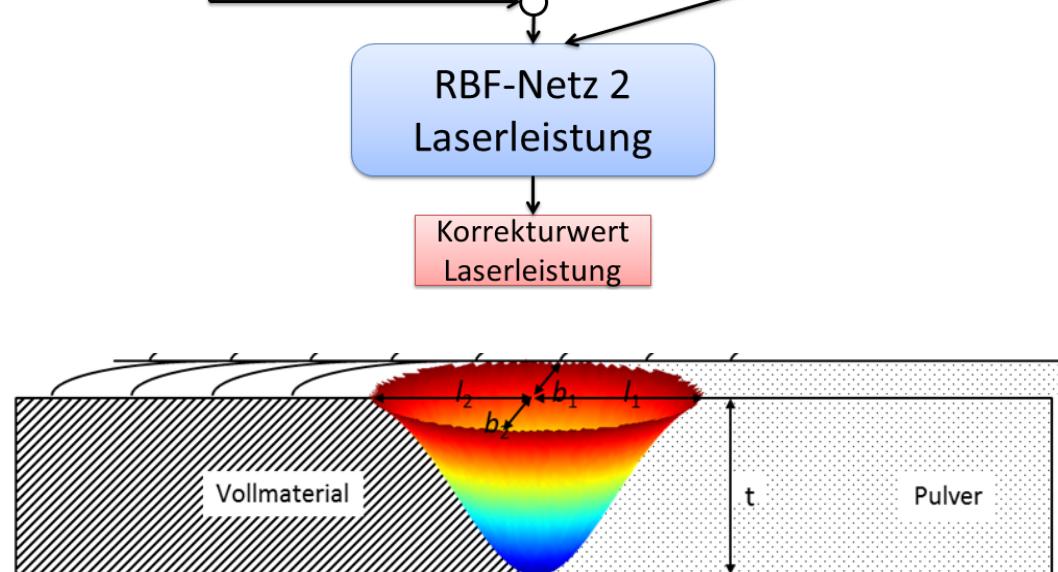
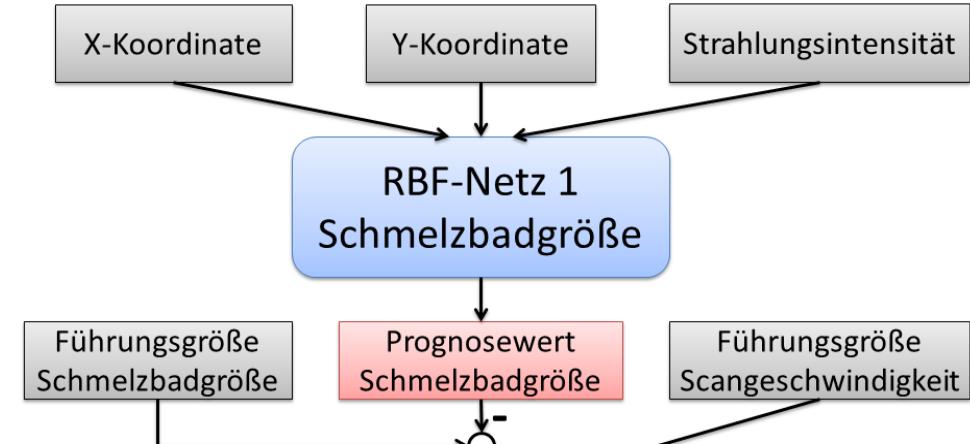
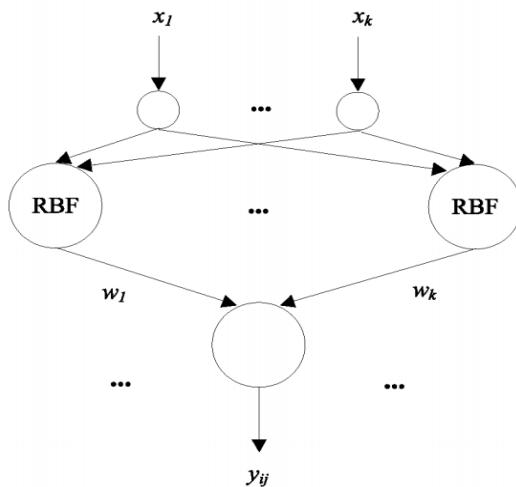
Speicherprogrammierbare Steuerungen SPS

- Messtechnische Lösung -

Der Laserschmelzprozess

Bestimmung der **Schmelzbadgröße**
(X- und Y-Koordinaten und
Strahlungsintensität)

Bestimmung der **Laserleistung**
(Schmelzbadgröße, Differenz zwischen
Ist-/Sollwert und Scangeschwindigkeit)



Speicherprogrammierbare Steuerungen SPS

- Messtechnische Lösung -

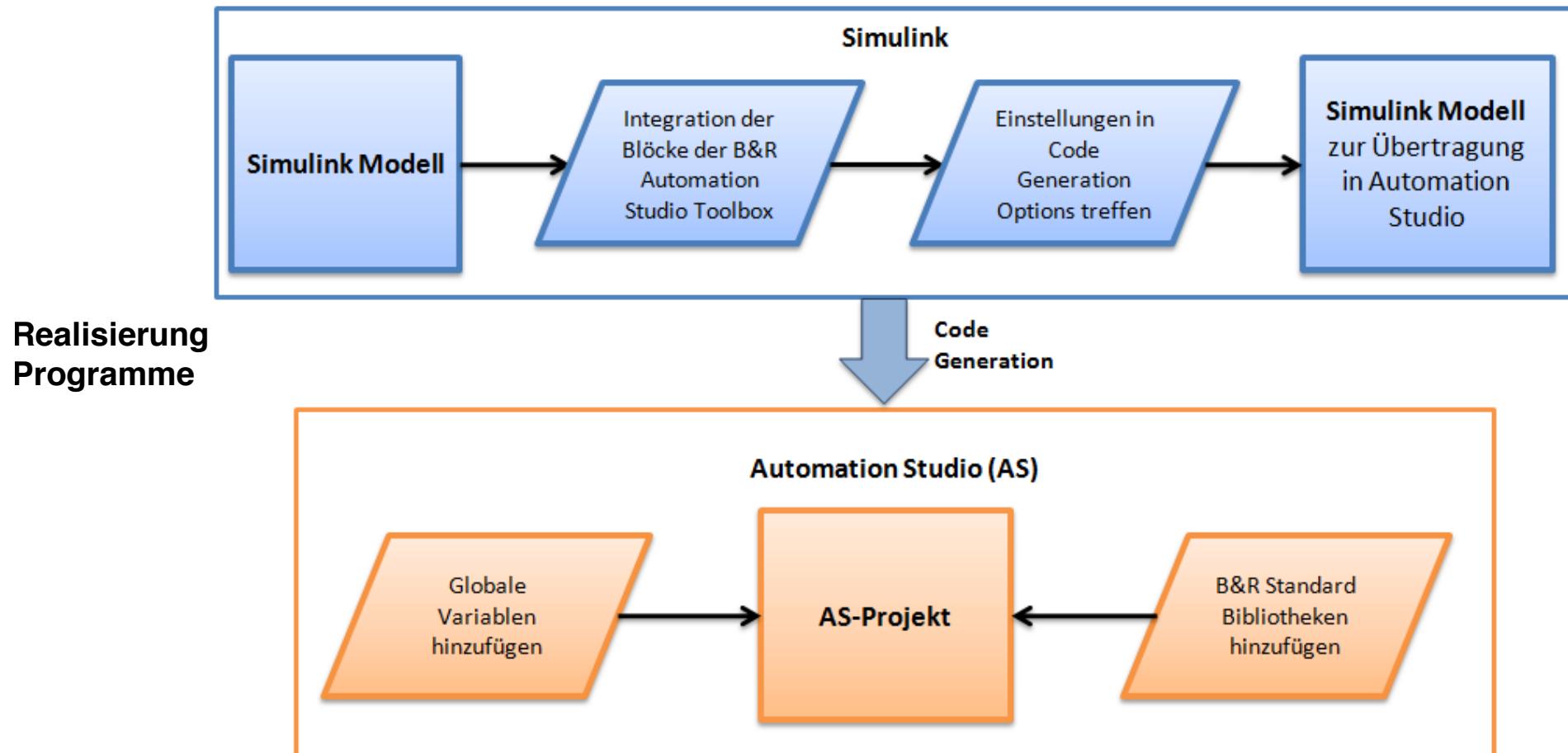
Einschränkungen auf dem Reaction Modul

Einschränkung	Auswirkung	Lösung
Keine Fließkommazahlen	Genauigkeit sinkt	Zusätzliche Divisionsblöcke, Datenbereich optimieren
Eingeschränkte Datenübertragung	Manuelle Einstellungen, kein automatisches Update (3 Eingänge)	Internen Speicher verwenden, Funktionalität aufteilen
Verzögerungen bei Datenübertragungen zwischen zwei Reaction Modulen	Zykluszeit steigt	Reaction Modul mit mehreren Eingängen verwenden
Maximaler Datenbereich DINT	Genauigkeit sinkt	Reaction Programmablauf ändern, zusätzliche Divisionsblöcke verwenden

Automatisiertes Messen und Steuern

- Messtechnische Lösung -

Überführung der Simulink-Modelle auf die SPS mit der
Toolbox „B&R Automation Studio Target for Simulink“



Speicherprogrammierbare Steuerungen SPS

- Literatur -

- Aspern von, Jens: SPS-Softwareentwicklung mit Petrinetzen, IEC-61131-codierte Hochgeschwindigkeitsnetze (Turbo-Netze), VDE VERLAG GMBH (2003), ISBN 3-8007-2728-5
- Aspern von, Jens: SPS-Softwareentwicklung nach IEC 61131, Hüthig Verlag, Heidelberg (2000), ISBN 3-7785-2681-2
- Aspern von, Jens: SPS Grundlagen - Aufbau - Programmierung (IEC61131, S7) - Simulation - Internet - Sicherheit Hüthig Verlag, Heidelberg (2005), ISBN 3-7785-2921-8
- John, Karl-Heinz, Tiegelkamp, Michael: SPS-Programmierung mit IEC 61131-3, Springer Verlag, ISBN 978-3-642-00268-7
- Lepers, Heinrich: SPS-Programmierung nach IEC 61131-3 - Mit Beispielen für CoDeSys und Step7, Franzis Verlag (2007), ISBN 3-7723-5805-5
- Petry, Jochen: SPS Programmierung nach IEC 61131-3 mit Multiprog 4.0, IBP Ingenieurbüro Petry (2007), ISBN 3-0002-2043-7

Lehrziele und Gliederung

- V1 Motivation, Anwendungsbereiche, Prozesse und Methoden der Automatisierungstechnik
- V2 Automatisierung in der Produktion
- V3 Boolesche Algebra 1
- Ü1 Matlab Einführung
- V4 Boolsche Algebra 2: Graphen
- Ü2 Übung Boolsche Algebra
- V5 Fuzzy Logic
- Ü3 Fuzzy Logic
- V6 Neuronale Netze
- Ü4 Neuronale Netze
- V7 Automatisiertes Messen und Steuern
- Ü5 Automatisiertes Messen und Steuern
- V8 **Speicherprogrammierbare Steuerungen**
- Ü6 Übungen und Musterklausuren