



# **Beitrag zur Inbetriebnahme von Antrieben mechatronischer Produktionsmaschinen als internetbasierte Dienstleistung**

Von der Fakultät für Maschinenbau der  
Technischen Universität Chemnitz

genehmigte

Dissertation  
zur Erlangung des akademischen Grades

Doktor-Ingenieur  
- Dr.-Ing. -

vorgelegt

von Dipl.-Ing. Kai Michael Gäbel  
geboren am 02. April 1976 in Karl-Marx-Stadt  
eingereicht am 09. Dezember 2005

Gutachter: Prof. Dr.-Ing. habil. Karl Heß  
Prof. Dr.-Ing. habil. Peter Löber  
Prof. Dr.-Ing. Dietmar Röhmer

Chemnitz, den 08. Juni 2006



# **Bibliographische Beschreibung**

**Gäbel, Kai Michael**

**Thema:**

## **Beitrag zur Inbetriebnahme von Antrieben mechatronischer Produktionsmaschinen als internetbasierte Dienstleistung**

**Dissertation an der Fakultät für Maschinenbau der Technischen Universität Chemnitz, Institut für Werkzeugmaschinen und Produktionsprozesse, Professur Steuerungs- und Regelungstechnik, Chemnitz, 2005**

- 127 Seiten
- 88 Bilder
- 12 Tabellen
- 80 Literaturverweise

Referat:

Mechatronische Produktionsmaschinen werden nicht mehr nur über ihren Leistungsumfang bewertet, sondern auch über die mitgebrachten Tools für Engineeringaufgaben. Wie erste Ansätze zeigen, erweitert die Integration von TCP/IP in die Steuerungsebene die Möglichkeiten zur Unterstützung des Anwenders nicht nur im Fertigungsprozess, sondern nahezu über den gesamten Lebenszyklus.

Die Dissertation beschreibt einen E-Service zur Inbetriebnahme von Lagereglern an Bewegungssteuerungen. Im Rahmen der prototypischen Realisierung erfolgt der Aufbau eines Dienstleistungsportals in einem Computernetz. Es übernimmt eine zentrale Rolle in der Kommunikationsstruktur. Aufgaben rund um die Dienstverwaltung, Dienstausführung und Dienstkommunikation von verteilten Programmen bilden hier die Schwerpunkte. Die Einordnung der realisierten und favorisierten Reglerinbetriebnahme erfolgt in die Gruppe der Offline-Verfahren. Auf dem gewählten Zielsystem, der Bewegungssteuerung SIMOTION von Siemens, läuft dabei der Echtzeitteil ab. Die Offline-Komponente mit den Identifikations- und Reglerberechnungsprogrammen wird im Anwender-Browser ausgeführt. Das System bildet zusammen einen internetbasierten, automatischen Service.

Schlagworte:

Achse, Bewegungssteuerung, Dienstleistung, E-Service, Identifikation, Inbetriebnahme, Internet, Lageregler, Mechatronik, Motion-Control, Produktionsmaschinen, Siemens, SIMOTION, Steuerung, Portal, Regelungstechnik, verteilt, Web-Service



Gewidmet meinen Eltern.



# Inhaltsverzeichnis

Verzeichnis der verwendeten Abkürzungen . . . . .	XI
<b>1 Einleitung</b>	<b>15</b>
<b>2 Inbetriebnahme von Maschinen</b>	<b>17</b>
2.1 Motion-Control . . . . .	17
2.2 Inbetriebnahme . . . . .	18
2.2.1 Hardwarekonfiguration . . . . .	19
2.2.2 Achsparametrierung . . . . .	19
2.2.3 Lageregler . . . . .	21
2.2.4 Technologieobjekte . . . . .	22
2.2.5 Parametrieren des Ablaufsystems . . . . .	23
2.2.6 Unterstützung der Inbetriebnahme . . . . .	25
2.3 Inbetriebnahme als internetbasierte Dienstleistung . . . . .	26
<b>3 Inbetriebnahme von Lageregelkreisen</b>	<b>29</b>
3.1 Analyse der Regelstrecke . . . . .	30
3.1.1 Theoretische Analyse . . . . .	30
3.1.2 Experimentelle Analyse . . . . .	31
3.1.2.1 Testfunktionen . . . . .	31
3.1.2.2 Modellstruktur . . . . .	35
3.1.2.3 Bestimmung der Modellparameter . . . . .	36
3.1.2.4 Identifikation mittels Simulation . . . . .	38
3.1.2.5 Nichtlinearitäten . . . . .	50
3.2 Reglerauswahl . . . . .	51
3.3 Bestimmen der Reglerparameter . . . . .	51
3.3.1 Gütemaß von Regelkreisen . . . . .	51
3.3.2 Berechnung der Reglerparameter mittels Simulation . . . . .	54
3.4 Validierung . . . . .	54
<b>4 Dienstleistungssysteme</b>	<b>55</b>
4.1 Reglerinbetriebnahme im Netz . . . . .	55

4.2	Grundlagen für E-Services . . . . .	56
4.2.1	Internet . . . . .	56
4.2.1.1	7-Schichten-Modell . . . . .	56
4.2.1.2	IP-Adressen . . . . .	58
4.2.1.3	Kommunikationsprotokolle . . . . .	59
4.2.1.4	Internet-Dienste . . . . .	60
4.2.1.5	Netze . . . . .	61
4.2.1.6	Kommunikationsmodelle . . . . .	61
4.2.1.7	Übertragungsarten . . . . .	62
4.2.1.8	Kommunikationsgeräte . . . . .	62
4.2.2	Programmiersprachen für Web-Applikationen . . . . .	64
4.2.2.1	HTML und XML . . . . .	64
4.2.2.2	Perl . . . . .	65
4.2.2.3	PHP . . . . .	65
4.2.2.4	Java . . . . .	65
4.2.2.5	SOAP, UDDI, WSDL – Web-Service . . . . .	66
4.3	E-Service – Dienstleistungen für die Produktion . . . . .	67
4.3.1	Teleservice . . . . .	67
4.3.2	Neue Ansätze . . . . .	68
4.3.3	Das E-Industrial-Services-Portal . . . . .	69
4.3.4	E-Industrial-Services . . . . .	70
4.3.4.1	E-Simulation . . . . .	71
4.3.4.2	E-Online-Diagnose . . . . .	71
4.3.5	ePS Network Services . . . . .	72
4.4	Sicherheitsbetrachtungen . . . . .	73
<b>5</b>	<b>Entwurf und Implementierung</b>	<b>75</b>
5.1	Konzept der Implementierung . . . . .	76
5.2	Systemstruktur . . . . .	78
5.3	Der Web-Service im System . . . . .	78
5.3.1	Portal . . . . .	78
5.3.1.1	Webserver-Funktionalitäten . . . . .	79
5.3.1.2	Verwaltungsaufgaben . . . . .	80
5.3.1.3	Schnittstellen . . . . .	80
5.3.1.4	Expertenzuschaltung . . . . .	84
5.3.1.5	Datensicherheit und Zugriffsbeschränkung . . . . .	85
5.3.1.6	Abrechnungssystem . . . . .	85
5.3.1.7	Optimierung der Struktur . . . . .	85
5.3.1.8	Dienstkonfigurator . . . . .	86



5.3.1.9	Dienststarter . . . . .	87
5.3.2	Die Anwenderschnittstelle/Offline-Komponente . . . . .	87
5.3.2.1	Programmierung und Benutzer-Oberfläche . . . . .	87
5.3.2.2	Ablaufsteuerung . . . . .	87
5.3.2.3	Kommunikationsmodul . . . . .	88
5.3.2.4	Programmierung . . . . .	90
5.3.2.5	Oberfläche . . . . .	91
5.3.3	Maschinenschnittstelle/Online-Komponente . . . . .	91
5.3.4	Resultierende Struktur . . . . .	92
5.4	Ausführung einer Online-Regelkreisinbetriebnahme . . . . .	93
5.4.1	P-Regler . . . . .	93
5.4.2	PV-Regler . . . . .	95
5.5	Ausführung einer Offline-Regelkreisinbetriebnahme . . . . .	98
5.5.1	Identifikation . . . . .	98
5.5.1.1	Simulator und Regelungsbibliothek . . . . .	98
5.5.1.2	Beaufschlagen der Strecke mit einem Testsignal . . . . .	101
5.5.1.3	Modellbestimmung . . . . .	102
5.5.1.4	Parameterberechnung . . . . .	104
5.5.1.5	Ergebnisse . . . . .	107
5.5.2	Reglerberechnung . . . . .	109
5.5.2.1	P-Regler . . . . .	109
5.5.2.2	PV-Regler mit Symmetrierfilter . . . . .	111
5.6	Sicherheitsmechanismen . . . . .	113
5.7	Bewertung der Verfahren . . . . .	113
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>115</b>
	<b>Literaturverzeichnis</b>	<b>117</b>
	<b>Abbildungsverzeichnis</b>	<b>123</b>
	<b>Tabellenverzeichnis</b>	<b>125</b>
	<b>Anhang</b>	<b>127</b>
<b>A</b>	<b>Algorithmen</b>	<b>127</b>
<b>B</b>	<b>Buszugriffsverfahren</b>	<b>129</b>
<b>C</b>	<b>Nichtlinearitäten</b>	<b>131</b>



# Verzeichnis der verwendeten Abkürzungen

## Formelzeichen

$A_{lin}$	Lineare Regelfläche
$A_{abs}$	Betragsregelfläche
$A_{sqr}$	Quadratische Regelfläche
$A_{sqr_t}$	Zeitgewichtete Quadratische Regelfläche
$A_{sqr_t^2}$	Zeitquadratgewichtete Quadratische Regelfläche
$A_{Zabs_t}$	Zeitgewichtete Betragsregelfläche auf Störsignal
$D$	Dämpfung
$E$	Schranken
$K_0$	Kreisverstärkung
$K_S$	Statischer Übertragungsfaktor der Regelstrecke
$K_P$	Reglerverstärkung P-Regler
$K_{PV}$	Reglerverstärkung P-Regler mit Vorsteuerung
$T, T_1 \dots T_m$	Zeitkonstanten mit $m = 1, 2, 3 \dots$
$T_a$	Abtastzeit
$T_{ANR}$	Anregelzeit
$T_{aPRNS}$	Taktlänge eines PRNS-Takts
$T_{AUS}$	Ausregelzeit
$T_{Filter}$	Zeitkonstante des Filters
$T_R$	Zeit der Rampenfunktion
$T_\Sigma$	Summenzeitkonstante
$T_t$	Totzeit
$e(t)$	Regelabweichung/Schleppabstand
$a$	Freier Parameter in der Optimierung
$b$	Freier Parameter in der Optimierung
$g(t)$	Gewichtsfunktion
$h(t)$	Sprungantwort
$h_m$	Maximale Überschwingweite
$h_{\ddot{u}}$	Überschwingweite
$P_m$	Punkt $(t_m, h_m)$
$t_m$	Zeitpunkt der maximalen Überschwingweite
$u(t)$	Eingangssignal eines Systems
$u_s(t)$	Sprungfunktion
$u_{s0}(t)$	Sprunghöhe
$x(t)$	Ausgangssignal eines Systems
$\delta(t)$	Einheitsimpulsfunktion
$\sigma(t)$	Einheitssprungfunktion

$\mu_G$	Gleitreibungskoeffizient
$\mu_H$	Haftreibungskoeffizient
$\nu$	Verstärkungsfaktor für Schranken
$\phi$	Zielfunktion
$\Phi$	Gütekriterium

## Abkürzungen

CGI	Common Gateway Interface
CIDR	Classless Interdomain Routing
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DP	Dezentrale Peripherie
DSC	Dynamic Stiffness Control
ePS	electronic Production Services
ES	Engineeringsystem
FhG	Fraunhofer Gesellschaft
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet-Protokoll
IPC	Industrie-Personal-Computer
IPO	Interpolator, Wegesteuerung
IPv4	IP Version 4
IPv6	IP Version 6
IRC	Internet Relay Chat
ISO	International Organization for Standardization
ITAE	Integral of time multiplied absolute error
JVM	Java Virtual Machine
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MC	Motion-Control
MPI	Multi Point Interface
MSR	Mutative Schrittweitenregelung
NXOR	Negierter Exclusive-Oder-Operator
OPC	Openness, Productivity and Collaboration
OSI	Open Systems Interconnection

PCU	Personal Computer Unit
PDA	Personal Digital Assistant
Perl	Practical Extraction and Report Language
PG	Programmiergerät
PHP	Hypertext Preprocessor
PRBS	Pseudo-Random Binary Signal
PRNG	Pseudo-Random Number Generators
PRNS	Pseudo-Random Number Signal
PV	P-Regler mit Vorsteuerung und Symmetrierfilter
RPC	Remote Procedure Call – ferngesteuerter Prozeduraufruf
SessID	Session-Identifikationsnummer
SGML	Standard Generalized Markup Language
SS	Schnittstelle
SSI	Server Side Includes
SSL	Secure Sockets Layer
TA	Testautomat
TCP	Transport Control Protocol
TO	Technologieobjekt
UDDI	Universal Service Description, Discovery and Integration
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
URL	Unified Ressource Locator
VDI	Verein Deutscher Ingenieure
VDMA	Verband Deutscher Maschinen- und Anlagenbau
VM	Virtual Machine
WAP	Wireless Application Protocol
WLAN	Wireless Area Network
WSDL	Web-Service Description Language
XHTML	Extensible Hypertext Markup Language
XOR	Exklusiv-Oder-Operator

Nicht aufgeführte Abkürzungen und Formelzeichen sind an den entsprechenden Stellen im Text definiert.



# Kapitel 1

## Einleitung

Maschinenbau und Automatisierungstechnik werden in zunehmendem Maße miteinander verwoben. Die Steuerungs- und Antriebstechnik nimmt in nahezu jeder Produktionsmaschine einen erheblichen Anteil an der Funktionserfüllung wahr. Während noch vor wenigen Jahren Nocken, Getriebe oder Kurvengetriebe durch mechanische Bauelemente ausgeführt wurden, dominieren gegenwärtig schon softwaregesteuerte Bewegungsabläufe, vor allen Dingen in komplizierten Zusammenhängen. Das Gebiet wird seit längerer Zeit als Mechatronik mit Verbindung zwischen mechanischen, elektronischen und informationstechnischen Komponenten vorangetrieben.

Die Beherrschung mechatronischer Systeme setzt modernste Methoden und Werkzeuge voraus. Hier geht es schon lange nicht mehr nur um Programmierunterstützung, sondern um einen Werkzeugkasten für verschiedenste Aufgaben. Die Tools werden in einem umfangreichen Engineeringsystem gebündelt. Die Inbetriebnahme der Gesamtanlage gelangt hier in das Blickfeld. Den wissenschaftlichen Mittelpunkt stellt dabei eine Reglerinbetriebnahme dar. Die Anforderungen werden vor allen Dingen durch Regelungen von Antrieben in einem Vielachssystem gestellt. Obwohl Inbetriebnahmemechanismen seit Jahren erforscht werden, stellen praktische Arbeiten die damit betrauten Ingenieure vor ernst zu nehmende Schwierigkeiten. Auch automatische Reglerinbetriebnahmetools verlangen nicht selten ein Nachoptimieren von Hand. Oftmals wird ein Experte hinzugezogen, was sich aufgrund des weltweiten Einsatzes als zeitaufwändig und unrentabel erweist.

Die Entwicklungen in der modernen Informations- und Kommunikationstechnik führen zu einer fast lückenlosen Vernetzung von Steuerungs- und Rechentechnik. In diesem Umfeld etabliert sich ein effektives Dienstleistungsgewerbe. Erste Ansätze sind realisiert und vor allen Dingen als Teleservice, Ferndiagnose, Visualisierung und Programmierung eingeführt. Diesem Trend folgend ist ein lohnendes Vorhaben, die Inbetriebnahme von Steuerungs-, Regelungs- und Antriebstechnik an der Maschine über das Internet durch Dienstleistungsrechner auszuführen.

Diese Dissertation stellt ein Konzept zur internetbasierten Inbetriebnahme von Antrieben an mechatronischen Produktionsmaschinen auf der Grundlage ausgewählter Strategien vor. Während bisherige Ansätze Tools in Form von Programmen auf Rechnern bereitstellen, wird eine neue Qualität durch die Ansiedlung der Dienstleistung in einem Netz angestrebt. Die Wahl des Netzes wird unterstützt

durch das Einbeziehen von Ethernet und TCP/IP in die Steuerungsebene. In der Dissertation wird eine Online- und eine Offline-Reglerinbetriebnahme prototypisch realisiert.

Zur Lösung der Aufgabe werden Grundlagen der Inbetriebnahme von Maschinen und Lageregelkreisen abgehandelt. Einzelne Methoden sind in der Regelungstechnik auch bekannt, werden jedoch auf den speziellen Fall angepasst und ausgestaltet. Darüber hinaus wird die Vernetzung der beteiligten Komponenten dargestellt und erste Lösungen von Web-Services der Automatisierungstechnik vorgestellt. Ein E-Service wird als verteiltes Programm zur Lagereglerinbetriebnahme implementiert und beschrieben.



# Kapitel 2

## Inbetriebnahme von mechatronischen Produktionsmaschinen

### 2.1 Motion-Control

In einem mechatronischen Gesamtsystem werden die elektronischen und informationstechnischen Komponenten durch Bewegungssteuerungen (Motion-Control, MC) realisiert. Der Einsatz von Bewegungssteuerungen verfolgt das Ziel, mechanische Bauelemente wie die Königswelle, Getriebe, Kurvenscheiben und Nocken durch Elektronik und Software zu ersetzen. Bewegungssteuerungen stellen die zentrale Lösung zur Realisierung mechatronischer Systeme dar. Sie vereinen Funktionalitäten für logische Operationen, für die Bewegung und die Regelung von physikalischen Größen wie Temperatur, Druck und Kraft.

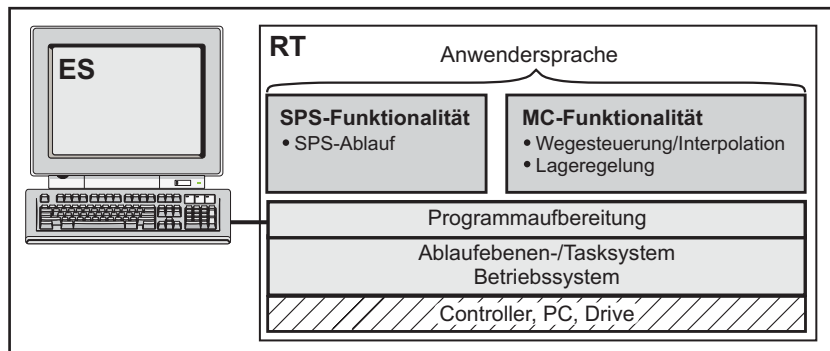


Abbildung 2.1: Aufbau mechatronischer Systeme

Bewegungssteuerungen stellen die zentrale Lösung zur Realisierung mechatronischer Systeme dar. Sie vereinen Funktionalitäten für logische Operationen, für die Bewegung und die Regelung von physikalischen Größen wie Temperatur, Druck und Kraft.

MC-Steuerungen bestehen aus einem Runtimesystem (RT) und einem Engineeringsystem (ES) entsprechend Abbildung 2.1. Im Engineeringsystem werden dem Anwender Tools für die Konstruktion, Programmierung, Inbetriebnahme und Diagnose bereitgestellt. Es deckt eine breite Palette von Aufgaben der Automatisierungstechnik ab. Das Engineeringsystem wird auf einem PC zur Ausführung gebracht. Das Runtimesystem läuft auf Controller, PC und Drive ab. Diese drei Ausführungsformen sind in Abbildung 2.2 dargestellt.

Controller-based beruht auf dem Prinzip einer eigenen Hardwareplattform, auf der die Steuerungs- und MC-Funktionalität läuft. Dabei steht die Flexibilität beim Anschluss unterschiedlicher Antriebe sowie ein breites Spektrum an verarbeitbaren Prozesssignalen im Vordergrund. Die Variante PC-based integriert Steuerungsfunktionen mit Motion-Control- und B&B-Funktionen auf einem IPC mit Echtzeitkern. Es können weitere Applikationen zur Prozessdatenverarbeitung und -auswertung parallel ablaufen. Besondere Vorteile bietet diese Plattform auch bei der Ferndiagnose und Fernbedienung. Die Vereinigung von Antrieb und MC wird als Drive-based bezeichnet. Vorteile ergeben sich in der kompakten Bauweise und bei der dezentralen Anwendung. Die Verlagerung der Berechnungen in den Antrieb ermöglicht zeitkritische und hochdynamische Achsankopplungen.

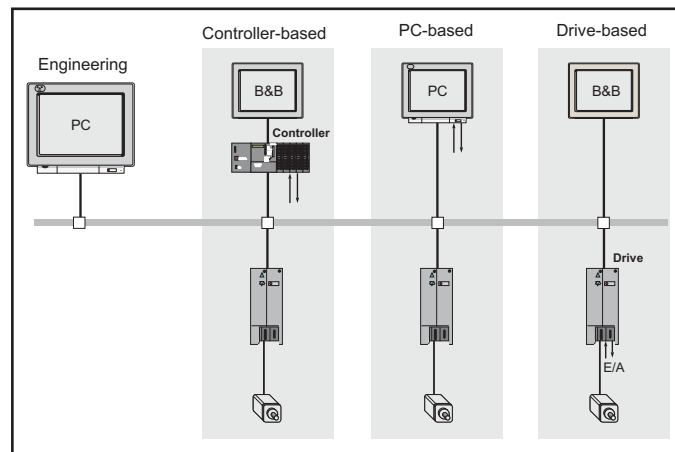


Abbildung 2.2: Ausführungen von Bewegungssteuerungen [24]

In Abbildung 2.3 ist eine Controller-based-MC in einem Automatisierungsnetz dargestellt. Die Verbindung der Steuerung in die Anwenderebene erfolgt hauptsächlich über Ethernet oder Profibus. Die Aktor-Sensor-Ebene kommuniziert über echtzeitfähige Bussysteme mit der Steuerung. Gegenwärtig findet vor allem Profibus, Sercos und zunehmend auch eine Mischung aus Profibus und Ethernet als Profinet Anwendung [39, 66].

Binäre Signale werden über Zusatzbaugruppen, den digitalen Ein- und Ausgangsbaugruppen, an die Steuerung angeschlossen. Analoge Signale, wie die der Sensoren zur Druck-, Kraft- und Temperaturmessung, werden dementsprechend über Analogbaugruppen in die Steuerung geführt.

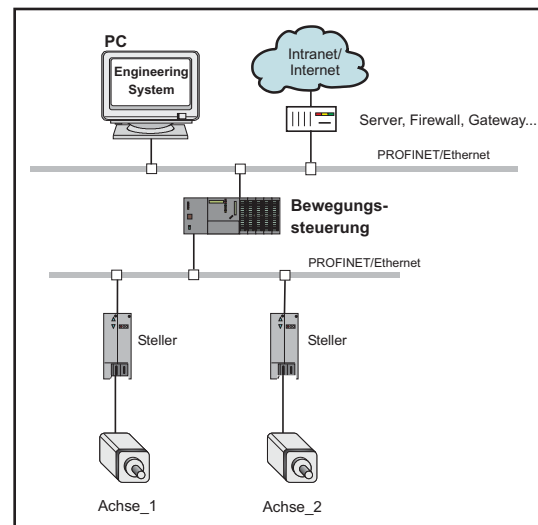


Abbildung 2.3: Netzaufbau in mechatronischen Systemen [24]

## 2.2 Inbetriebnahme

Die neuen Eigenschaften mechatronischer Systeme ermöglichen nicht nur hochwertigere technische Lösungen, sie konfrontieren Ingenieure und Techniker auch mit neuen Aufgaben. Das setzt in der Regel ein völlig neues anspruchsvolles Handling einer Vielzahl von Werkzeugen im Rahmen eines

Engineering systems voraus. Entwurf und Inbetriebnahme von Bewegungssteuerungen an Maschinen werden im Kern durch:

- Programmierung
- Konfiguration
- Parametrierung
- spezielle Inbetriebnahmemechanismen für Antriebe und Regelungen

ausgeführt.

Einige Schritte der Inbetriebnahme von Bewegungssteuerungen werden im Anschluss näher betrachtet.

### 2.2.1 Hardwarekonfiguration

In einem ersten Schritt ist die Hardwarekonfiguration von Steuerung und Maschine vorzugeben. Sie wird über eine grafische Benutzeroberfläche erstellt und in einer Datei abgespeichert. Diese enthält alle Geräte, die mit der Steuerung direkt verbunden sind. Hierzu zählen neben der Steuerung selbst die Stromversorgung, E/A-Baugruppen und die Steller der Antriebe. Sie können aus einem Katalog ausgewählt werden, der zu jedem Gerät eine Standardkonfiguration enthält. Die Konfiguration ist einzeln anzupassen.

In einem weiteren Schritt werden die Adressen dieser Peripheriegeräte vergeben und der Netzaufbau wird vorgenommen. Die weitere Parametrierung des Netzes ist abhängig von der Wahl des Netztyps. Wird Profibus zugrunde gelegt, sind der Master und die Slaves zu definieren sowie die Buszyklen festzulegen. Sind Antriebe am Bus angeschlossen, so ist zusätzlich ein Kommunikationstelegramm auszuwählen. Weitere Notwendigkeiten ergeben sich im Einzelfall.

Ist die Hardwarekonfiguration erstellt, stehen dem Anwender Mechanismen für die Konsistenzprüfung zur Verfügung. Anschließend kann sie auf die Steuerung geladen werden. Beim Wiederanlauf wird die Konfiguration mit dem Ist-Zustand verglichen und eventuelle Fehler werden an den Anwender gemeldet.

### 2.2.2 Achsparametrierung

Bevor mit der Programmierung begonnen werden kann, sind Technologieobjekte (TO) in das Steuerungsprogramm einzufügen. Technologieobjekte enthalten Funktionen, die den ersetzten Maschinenbauelementen entsprechen. Die folgenden TOs sind beispielsweise realisiert:

- Achse

- Getriebe
- Kurvenscheibe
- Nocken
- Externer Geber.

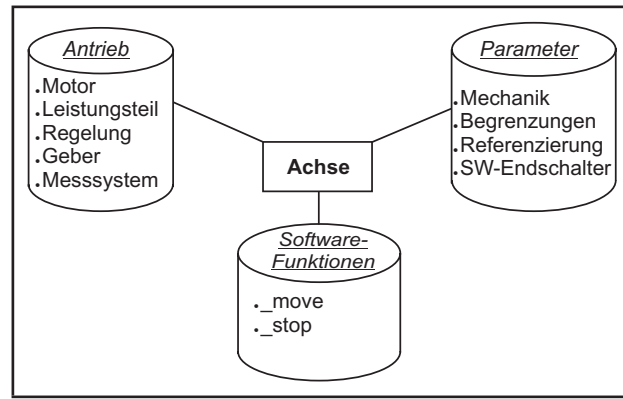


Abbildung 2.4: Achsdefinition in mechatronischen Systemen

Die Parametrierung der TOs soll stellvertretend mit dem TO Achse abgehandelt werden. Die Abbildung 2.4 zeigt die Definition einer "Mechatronischen Achse". Sie zeichnet sich durch einen Teil Hardware (dem Antriebsstrang), einen Teil Software (der Funktionalität) sowie den zugehörigen Parametern aus.

Der angeschlossene Antrieb kann unterteilt werden in:

- den Steller mit der Regelungsfunktionalität für die Strom- und Drehzahlregelung
- den Motor (Schritt-, Servo- und Linearmotor)
- die Mechanik (Getriebe, Spindel).

Der Antriebsstrang ist schematisch in Abbildung 2.5 dargestellt. Das Bild zeigt die Anbindung des Antriebs an die Steuerung über den Lageregelkreis. Er ist dem Drehzahl- und Stromregelkreis überlagert.

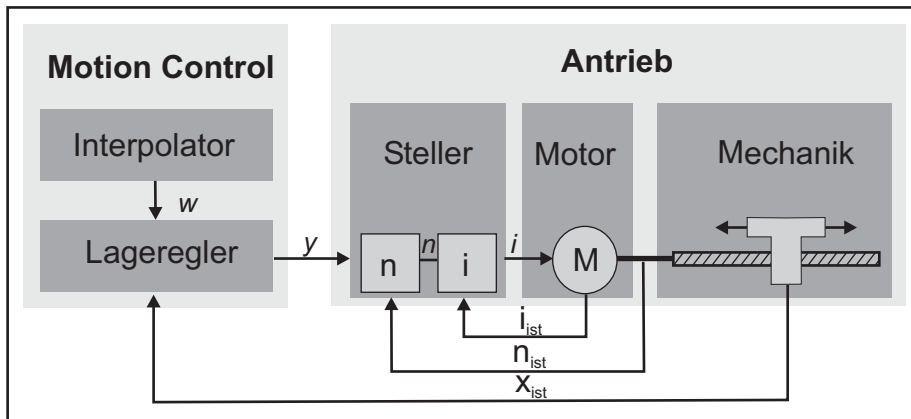


Abbildung 2.5: Antriebsstrang mit MC-Anbindung

Die Funktionalität wird in allgemeine (\_enableAxis, \_stop, \_disableAxis) und achsspezifische (\_move, \_pos, \_enableCamming) Funktionen unterschieden. Daraus ergeben sich die folgenden verfügbaren Achstechnologien:

- Drehzahl-Achse



Aufgrund der transportbedingten Totzeiten (siehe Kapitel 2.2.5) der Soll- und Ist-Werte ist das Regelverhalten für schnelle, dynamische Prozesse nicht optimal.

Um diesen Nachteil zu kompensieren, wird in einer weiteren Variante der Lageregler im Antrieb gerechnet. Außerdem erfolgt eine Erweiterung mit einem Vorfilter und einer Vorsteuerung. Die Vorsteuerung schaltet den Drehzahl Sollwert direkt auf den Drehzahlregler auf. Die Berechnung des Lagereglers im Antrieb hat den Vorteil, dass kommunikationsbedingte Totzeiten entfallen. Besitzt der Antrieb eine kleinere Zykluszeit, wird zusätzlich das Regelverhalten verbessert. Diese Erweiterung wird als DSC (*Dynamic Stiffness Control*) bezeichnet (Abbildung 2.7).

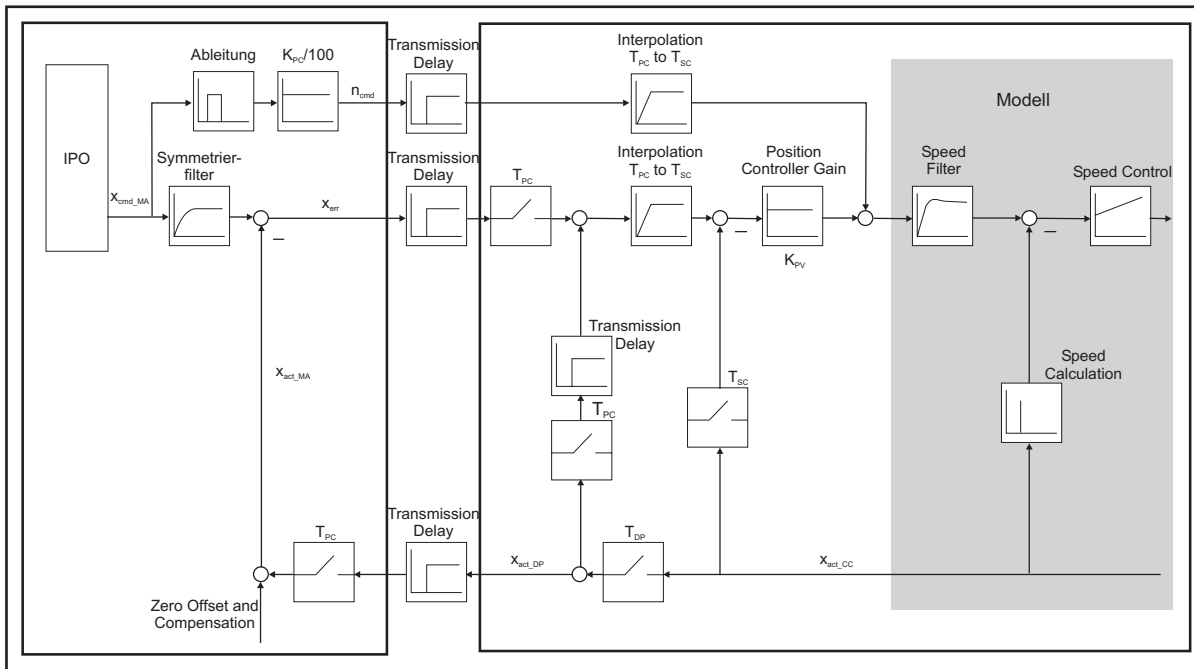


Abbildung 2.7: Geschwindigkeitssollwertschnittstelle mit Vorsteuerung [65]

In der Inbetriebnahmephase des Regelkreises wird eine dieser Reglerstrukturen gewählt. In beiden Fällen sind die Reglerparameter zu bestimmen. Bei einer Verwendung der Vorsteuerung wird zusätzlich die Summenzeitkonstante  $T_{\Sigma}$  des Antriebsstrangs bestimmt. Sie geht in die Parametrierung des Symmetrierfilters ein.

Die genannten Schritte sind für jede weitere Achse zu wiederholen. In Vielachssystemen erfordert das manuelle Vorgehen viel Zeit.

## 2.2.4 Technologieobjekte

Sind alle Achsen parametrierung, werden die weiteren Technologieobjekte (Getriebe, Kurvenscheiben, Nocken) hinzugefügt. Die Reihenfolge orientiert sich an den Abhängigkeiten der Objekte untereinander.

### 2.2.5 Parametrieren des Ablaufsystems

Bewegungssteuerungen verfügen über ein Multi-Task-Ablaufsystem. Es legt die zeitliche Reihenfolge und Häufigkeit der Abarbeitung von Programmen fest. Hierzu zählen Systemprogramme ebenso wie Anwenderprogramme. Die Einteilung erfolgt in Ablafebene. Jede Ebene wird durch ein spezielles Ablaufverhalten und eine eigene Systempriorität charakterisiert. Die Ablafebene werden in Tasks/Programme unterteilt, die einen zyklischen, azyklischen, sequenziellen oder ereignisgesteuerten Charakter haben.

Die Abbildung 2.8 zeigt einen Auszug des Ablaufsystems einer Bewegungssteuerung. Das "global-control"-Signal ist der Taktgeber auf dem Profibus<sup>1</sup>. Darauf synchronisieren sich alle Kommunikationspartner. Das Ablaufsystem arbeitet die verwendeten zyklischen Tasks in einem festen Verhältnis zu diesem Takt ab.

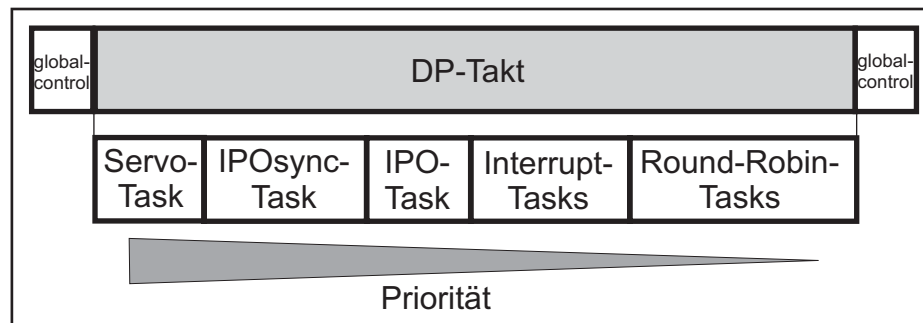


Abbildung 2.8: Reihenfolge der Tasks im Ablaufsystem

Im folgenden sollen einige für die Programmierung wichtige Tasks charakterisiert werden.

**Servo-Task** – läuft synchron zum (Profibus-)Takt in einem einstellbaren Verhältnis ab. In ihr wird unter anderem der Lageregler berechnet.

**IPOSynchronous-Task** – läuft zyklisch vor dem IPO-Takt einmalig durch. Das Verhältnis von Servo-Takt zu IPO-Takt ist in bestimmten Grenzen einstellbar. Wie alle hier folgenden Tasks ist sie vom Anwender zu programmieren. Die benötigte Zeit wird überwacht, bei einer Zeitüberschreitung wird eine vom Anwender zu definierende Funktion ausgeführt.

**Interrupt-Tasks** – sind ereignisgesteuerte Tasks. Die Auslösung kann über Anwenderfunktionen (wait-for-condition, user-interrupt) oder durch Systemfunktionen oder -fehler (Ausfall von Hardware) erfolgen.

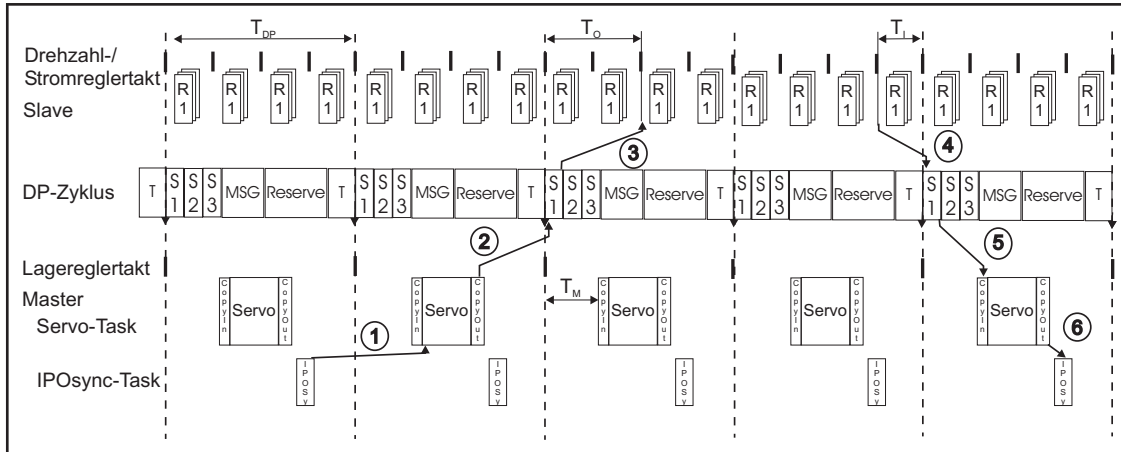
**Background-Task** – wird im Hintergrund zyklisch über ein Round-Robin-Verfahren abgearbeitet. Eine Zeitüberwachung ist vom Anwender konfigurierbar.

**Motion-Tasks** – werden wie die Background-Task über Round-Robin abgearbeitet, jedoch sequenziell und ohne Zeitüberwachung.

<sup>1</sup> Wird kein Profibus verwendet, so wird das Signal durch einen eigenen Taktgeber der Steuerung generiert.

**Startup- und Shutdown-Task** – werden beim STOP/RUN- bzw. beim RUN/STOP-Übergang durchlaufen.

Aufgrund des in Abbildung 2.8 dargestellten Ablaufsystems und der Verwendung des Profibuses ergibt sich bei der Kommunikation mit den Slaves eine Totzeit, die in der Identifikation und Reglerdimensionierung beachtet werden muss. In Abbildung 2.9 ist das Taktzeitdiagramm mit dem Signalweg für einen speziellen Fall – Schreiben des Drehzahlollwertes in der IPOsynchronous-Task und dem frühesten Sichtbarwerden der Reaktion darauf in selbiger Task – dargestellt.



**Abbildung 2.9: Totzeiten durch Profibus DP, optimiertes Protokoll**

- $T_I$  – Synchronisationszeitpunkt der Istwerterfassung
- $T_O$  – Synchronisationszeitpunkt der Sollwertübernahme
- $T_M$  – Verschiebung des Lagereglertaktes des Masters
- $T$  – Taktgeber für Profibustakt (global-control)
- $S1 \dots Sn$  – Zyklischer Datenaustausch mit Slave 1 ... n
- $MSG$  – Nichtzyklischer Kommunikationsanteil
- $R1 \dots Rm$  – Regler 1 ... m

Beschreibung zu Abbildung 2.9:

1. Setzen einer Variable<sup>2</sup> in der IPOsynchronous-Task.
2. Bearbeiten in der Servo-Task im nächsten Takt und anschließendes Schreiben durch Servo-Task auf den Bus und Transport.
3. Übernahme des Sollwertes durch den Slave und Verarbeiten im Antrieb.
4. Schreiben des aktuellen Istwertes auf den Bus.
5. Lesen vom Bus durch Master.
6. Servo-Daten stehen der IPOsynchronous-Task zum Lesen zur Verfügung.

<sup>2</sup> Achse\_1.servosettings.additionalsetpointvalue[0]



Bei  $T_{DP} = T_{Servo} = T_{IPO}$  berechnet sich die Zeit, in der beim Anwender frühestens eine Reaktion des Antriebs auf eine Änderung eines Sollwertes sichtbar wird, wie folgt:

$$T_{totLS} = 3T_{DP} + T_O + T_I \quad (2.1)$$

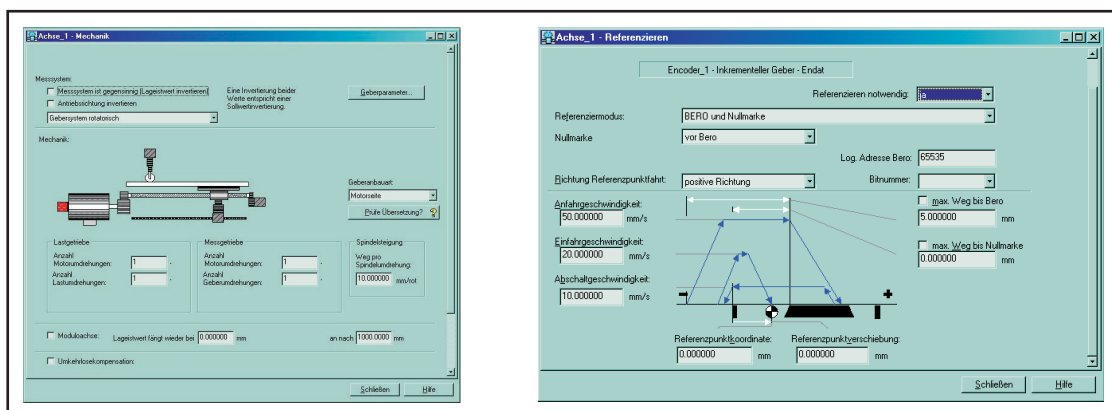
Für Sensor-Daten und für vom Interpolator generierte Daten gilt dies mit einem IPO-Takt mehr.<sup>3</sup>

Die Totzeiten für den Regler berechnen sich nach:

$$T_{totRegler} = T_O + T_I + T_{Servo} \quad (2.2)$$

### 2.2.6 Unterstützung der Inbetriebnahme

Die Inbetriebnahme und Projekterstellung ist eine komplexe Aufgabe, die einen entsprechenden Kenntnisstand des Anwenders voraussetzt. Die Möglichkeiten von Computern lassen eine Unterstützung/Hilfestellung in verschiedenen Formen zu. Grafische Benutzeroberflächen in Verbindung mit Eingabemasken eignen sich zur Verbesserung des Verständnisses von Maschinenparametern. So erfolgt beispielsweise die Parametrierung der am Antrieb angeschlossenen Mechanik über derartige Eingabemasken. Sie stellen dem Anwender eine Anzahl von Möglichkeiten der Mechanikstruktur zur Verfügung, die je nach Achstyp vorgegeben werden. Zwei Beispiele zeigt Abbildung 2.10. Links ist die Parametrierung des Mechanikstranges dargestellt und rechts die Parametrierung der Funktion "Referenzieren der Achse".



**Abbildung 2.10: Inbetriebnahmemasken der Mechanik (links) und zum Referenziervorgang (rechts) einer Achse [64]**

Komplexere Probleme werden mittels Benutzerführung über eine Anzahl von abzuarbeitenden Schritten, mit so genannten Wizards, gelöst. Ein Wizard begleitet den Anwender durch den Parametriervorgang und trifft dabei selbstständig Schlussfolgerungen auf Eingaben. So hat das Erstellen einer Rundachse mittels eines Wizards beispielsweise als Folge, dass Einheiten im Winkelmaß und nicht metrisch angegeben werden.

<sup>3</sup> Bei einem Verhältnis DP : Servo : IPO = 1 : 1 : 1

Nach Abschluss einer Aufgabe prüfen Diagnosefunktionen die eingestellten Werte anhand von Tests auf ihre Richtigkeit.

## 2.3 Inbetriebnahme als internetbasierte Dienstleistung

Wie schon die begrenzten Ausführungen zeigen, stellt die Inbetriebnahme von Antrieben einen aufwändigen Prozess dar. Lösungswege sind auch teilweise bekannt. Nachfolgende Erweiterungen werden in der Arbeit vorgestellt:

- Inbetriebnahmesystem mit Echtzeitkomponenten und Reglerberechnung
- internetbasierte Dienstleistung.

In dieser Dissertation wird eine automatische, internetbasierte Lagereglerinbetriebnahme für Servoantriebe vorgestellt. Die Reglerinbetriebnahme ist eine wissenschaftlich anspruchsvolle Aufgabe [68]. Tools zur Unterstützung beschränken sich hauptsächlich auf konkrete Anwendungsfälle. Weichen die Vorgaben ab, werden Experten vor Ort angefordert. Kann diese Unterstützung aus der Ferne erfolgen, ergeben sich für den Anwender Zeit- und Kostenvorteile.

Systeme zur Reglerinbetriebnahme sollten die folgenden Module beinhalten:

- Echtzeitfunktionen zur Sollwertvorgabe und Istwertspeicherung von Größen
- Visualisierungsfunktionen
- Methoden zur Reglerauswahl
- Methoden zur Reglerdimensionierung
- Aufbau der Reglerstruktur im Zielsystem.

Der Einzug von Internettechnologien in die Steuerungsebene bereitet die Voraussetzungen für TCP/IP-basierte Dienstleistungen. Erste Ansätze sind unter dem Begriff E-Service bekannt [2, 9, 15, 37, 67, 73, 74]. Realisiert sind Tools zur Auswertung von Maschinendaten, zur Generierung von Anwenderprogrammen, zur Konstruktion [35] sowie zur Optimierung von Abläufen. Ein typisches Beispiel für Dienstleistungen mit Expertenunterstützung ist der Teleservice. Derartige Dienste schaffen eine neuartige Hersteller- und Kundenbeziehung [23].

Das Bereitstellen und die Ausführung von Dienstleistungen in Netzen erfordert besondere Rechnerstrukturen. Im Internet werden diese vorzugsweise als Client-Server-Architektur ausgeführt. Der Server erfüllt Aufgaben rund um die Dienstleistung. Dabei steht nicht nur die Koordination der Ausführung der Dienstleistungen im Mittelpunkt, sondern auch die Vermittlung der Kommunikation, das Buchen und Abrechnen der Dienste sowie die Verwaltung von Benutzern. Diese Funktionen stehen dem Anwender unter einer zentralen Adresse – dem Portal – zur Verfügung. Das Portal kann

beispielsweise Dienste zur Visualisierung, Programmierung, Diagnose und Optimierung enthalten. Die Dienste setzen sich aus modularen, verteilten Programmen zusammen, deren Funktionen sich über die Kommunikationspartner erstrecken. Kommunikationspartner sind Rechner im Portal und die Clients. Clients sind im Fall der Reglerinbetriebnahme ein Anwender-PC und die Steuerung. Der PC führt dabei den Dienstablauf aus und enthält die Methoden zur Berechnung. Die Steuerung erfüllt Echtzeitfunktionen (Funktionsgenerator, Trace, Überwachungen). Ein Dienst Visualisierung setzt sich beispielsweise aus Echtzeittrace (Steuerung), Kommunikation (Portal) und der Grafikoberfläche (PC) zusammen. Abbildung 2.11 zeigt ein erstes Schema zur internetbasierten Dienstleistung.

Neben den Anforderungen an die Reglerinbetriebnahme ergeben sich für die Realisierung weitere Anforderungen in Bezug auf die Vernetzung:

- durchgängige Kommunikation vom Wirk- bis zum Ausführungsort des Dienstes.
- Aufbau einer Netzarchitektur für Dienstleistungen (Portal und Schnittstellen).
- Anpassung der Verfahren zur Reglerinbetriebnahme an E-Service-Strukturen.
- Sicherheit für Mensch, Daten und Maschine.

Die folgenden Vorteile für den Anwender werden beim Einsatz internetbasierter Dienstleistungen erwartet:

- Es wird automatisch mit der aktuellen, auf dem Server gespeicherten Software gearbeitet. Update-Funktionen sind nicht nötig.
- Die Unterstützung durch Experten über ein Netz ist jederzeit gewährleistet.
- Weltweiter Zugriff auf die Ergebnisse des Dienstes sind möglich.

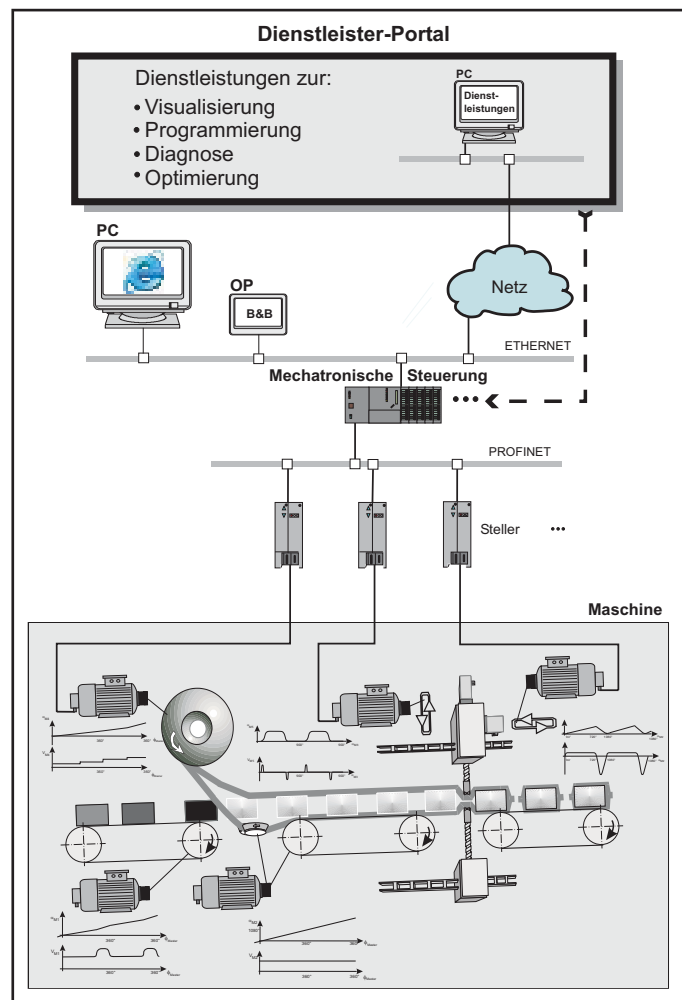


Abbildung 2.11: Schema zur internetbasierten Dienstleistung

Auch für die Anbieter dieser Dienstleistungen ergeben sich Vorteile gegenüber der Dienstleistung vor Ort:

- Multiplikation des Know-hows durch Implementation in Algorithmen und der weltweit mögliche Zugriff rund um die Uhr.
- Gesonderte Abrechnung der Dienste nach Nutzung.
- Die Auswertung der Dienstergebnisse bringt einen Einblick in die Nutzungsgewohnheiten des Maschinenbetreibers.

# Kapitel 3

## Inbetriebnahme von Lageregelkreisen

Die Reglerinbetriebnahme ist seit Jahren ein aktuelles Thema in der Regelungstechnik. Sie zielt auf eine optimale Anpassung von Regelkreisen an die jeweiligen Zielsysteme ab. Betrachtete Zielsysteme sind in dieser Arbeit vorrangig Antriebe mechatronischer Systeme. Sie werden hauptsächlich drehzahl- oder lagegeregelt betrieben. Für deren Reglerentwurf sind eine Anzahl von Strategien bekannt, von denen einige in diesem Kapitel näher betrachtet werden.

Reglerinbetriebnahmesysteme unterteilen sich in Online- und Offline-Verfahren [20].

Online-Verfahren sind dadurch gekennzeichnet, dass ihre zugrunde liegenden Informationen (Soll-Ist-Verläufe) während des Betriebes der Steuerung erfasst und verarbeitet werden. Die Reglerparametrierung erfolgt auf der Basis von zumeist adaptiven Algorithmen, ausgehend von einer Startinitialisierung. Der Vorteil, die Regleranpassung im laufenden Betrieb durchführen zu können, ist mit hohen Anforderungen an die Ressourcen der Steuerung verbunden.

In der Abbildung 3.1 ist das Prinzip dargestellt.

Eine Alternative für Systeme mit zeitinvarianten Prozessparametern stellen die Offline-Verfahren nach Abbildung 3.2 dar. Sie benötigen einen Inbetriebnahme-PC, auf dem die Offline-Toolkette ausgelagert wird. Auf der Steuerung (online) sind nur Funktionen, die Echtzeitbedingungen zu erfüllen haben, angesiedelt. Hierzu zählen der Signalgenerator und Funktionen zum Messen und Aufzeichnen von Informationen. Über eine Schnittstelle erfolgt die Datenübergabe zwischen Steuerung und Inbetriebnahme-PC. Die Offline-Toolkette umfasst die folgenden Schritte:

1. Analyse der Regelstrecke
2. Reglerauswahl

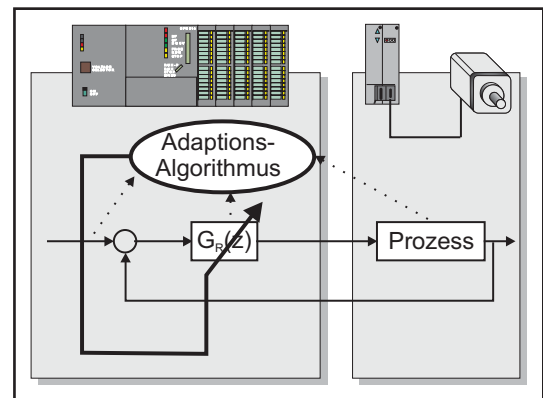


Abbildung 3.1: Online-Inbetriebnahme [20]

3. Berechnung der Reglerparameter
4. Validierung der Ergebnisse.

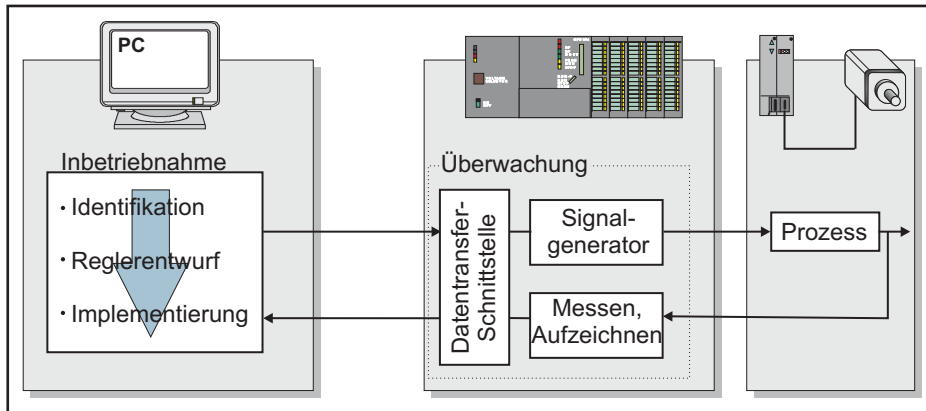


Abbildung 3.2: Offline-Inbetriebnahme [20]

In der Inbetriebnahmephase der Maschine werden vorzugsweise Offline-Verfahren angewendet. Einige werden in diesem Kapitel näher betrachtet.

## 3.1 Analyse der Regelstrecke

Bei der Identifikation von Strecken ist das Ziel die Beschreibung des Systems mittels eines mathematischen Modells. Dieses Modell enthält typische charakteristische Eigenschaften der Strecke. Die Modellbildung kann auf dem Weg der theoretischen oder experimentellen Analyse erfolgen. Beide Formen haben ihre Vor- und Nachteile, welche hier kurz betrachtet werden sollen [32, 33, 41, 62].

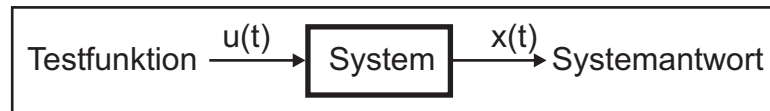
### 3.1.1 Theoretische Analyse

Wichtige Voraussetzungen bei der theoretischen Analyse sind das Wissen über die physikalischen Vorgänge und über den Aufbau der Regelstrecke. Diese werden mit den physikalischen Bilanzgleichungen zum Kräfte- und Momentengleichgewicht sowie den Erhaltungssätzen der Masse, der Energie und des Impulses beschrieben. Es wird mit Hilfe der Kennwerte ein Modell oft höherer Ordnung erstellt. Es ist üblich, die Modellordnung zu verringern und so einfachere, übersichtlichere Modelle zu erhalten, welche dem System aber dennoch hinreichend entsprechen.

Die Vorteile der theoretischen Analyse liegen darin, dass die innere Struktur des Systems vor der eigentlichen Realisierung erfasst werden kann und so Kenntnisse über die inneren Zusammenhänge gewonnen werden. Das steigert das Verständnis über das System und lässt Schwachstellen frühzeitig erkennen. Nachteilig sind die oft komplexen Modelle, die sich ergebenden Ungenauigkeiten bei der Reduzierung der Ordnung und unzureichendes Wissen über die Kenn- und Einflussgrößen [31, 41].

### 3.1.2 Experimentelle Analyse

Die experimentelle Analyse von Regelstrecken zielt auf die Strukturermittlung und die Bestimmung der Parameter dieser Struktur ab. Grundprinzip der Identifikation ist das Aufschalten eines Testsignals auf das System. Die darauf folgende Reaktion wird als Systemantwort bezeichnet, wobei diese die Charakteristiken des Systems und der Testfunktion enthält [41]. Abbildung 3.3 stellt dieses Prinzip dar.



*Abbildung 3.3: Prinzip der experimentellen Analyse*

Aus der Systemantwort lässt sich das Modell bestimmen. Die Methoden sind abhängig von der aufgeschalteten Testfunktion und dem Verlauf der Systemantwort. Die Wahl des Testsignals und der Methode charakterisiert das Identifikationsverfahren. Um eine ideale Kombination für den untersuchten Fall zu finden, werden die Anforderungen an Identifikationsverfahren wie folgt benannt:

1. parametrisches Modell der Regelungstechnik als Ergebnis
2. Abdecken eines breiten Spektrums von Systemen
3. keine Überlastung des Systems
4. Determiniertheit des Ergebnisses
5. Identifikation im Bereich des Arbeitspunkts.

Aus den definierten Zielsystemen (Antriebe für mechatronische Maschinen) ergeben sich weitere spezielle Anforderungen in Bezug auf die Gewährleistung der Sicherheit von Mensch und Maschine während der gesamten Inbetriebnahme. Diese Kriterien sind noch zu entwerfen und vom Verfahren zu erfüllen.

#### 3.1.2.1 Testfunktionen

Ein wichtiger Gesichtspunkt für eine erfolgreiche Identifikation ist die Wahl der Testfunktion. Im Anschluss werden folgende Eingangssignale vorgestellt:

- Impulsfunktion
- Sprungfunktion
- Rampenfunktion
- Rausch- und Sinusfunktion.

Die Testfunktionen müssen das System über einen möglichst großen Frequenzbereich erregen, um erfolgreich zu sein.

### Impulsfunktion

Die Einheitsimpulsfunktion  $\delta(t)$  ist ein Impuls als Eingangsgröße  $u(t)$  mit der Fläche eins, wobei gilt:

$$\delta(t) = \begin{cases} 0 & \text{für } t < 0 \text{ und } t > 0 \\ \infty & \text{für } t = 0 \end{cases}, \int \delta(t) dt = 1. \quad (3.1)$$

Am Ausgang des Systems liegt nach dem Anlegen des Impulses die Gewichtsfunktion  $g(t)$  als Impulsantwort an. Abbildung 3.4 zeigt die Impulsantwort eines Proportionalgliedes erster Ordnung.

Beim Einsatz in der Technik ist die Impulsfunktion in ihrer Höhe und zeitlichen Beschränkung jedoch aufgrund der jeweiligen physikalischen Gesetze und der eingesetzten Technik beschränkt.

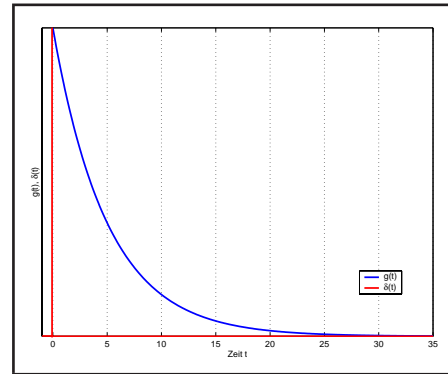


Abbildung 3.4: Gewichtsfunktion eines  $PT_1$ -Gliedes

### Sprungfunktion

Die Sprungfunktion  $u_s(t)$  als Eingangsgröße  $u(t)$ , ist eine einfach zu realisierende wichtige Funktion bei der Identifikation von Regelstecken. Sie kann bezogen auf eine Einheitssprungfunktion  $\sigma(t)$  durch

$$u_s(t) = u_{s0} \cdot \sigma(t) \quad (3.2)$$

$$\sigma(t) = \begin{cases} 0 & \text{für } t \leq 0 \\ 1 & \text{für } t > 0 \end{cases} \quad (3.3)$$

definiert werden.

Die Antwort auf eine Sprungfunktion ist die Sprungantwort  $h(t)$  (Abbildung 3.5). Typisch bei der Identifikation von Antrieben ist die Aufschaltung eines Drehzahl Sprunges auf den Drehzahlregelkreis.

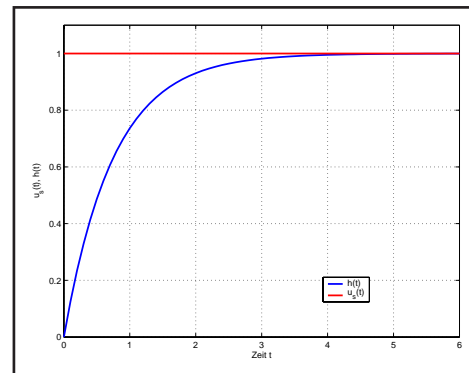


Abbildung 3.5: Sprungantwort eines  $PT_1$ -Gliedes

Treppenfunktionen sind mehrere aneinandergesetzte Sprungfunktionen. Wird die in der Abbildung 3.6 dargestellte Funktion auf Antriebe angewendet, bietet sie folgende Vorteile gegenüber einzelnen Sprüngen:

- Fehler durch Reibung oder Lose werden minimiert, da nur die Stufen im Arbeitspunkt ausgewertet werden.



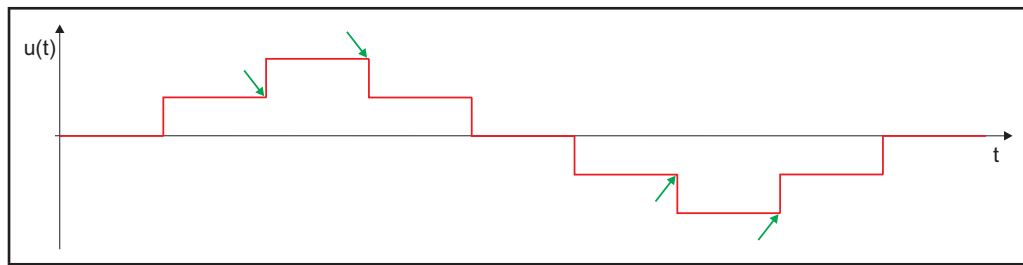


Abbildung 3.6: Treppenfunktion

- Der Rückweg wird ebenfalls zur Identifikation genutzt.

Im Fall der Aufschaltung auf den Drehzahlregelkreis werden nur die Stufen, die in Abbildung 3.6 mit einem Pfeil markiert sind, ausgewertet.

### Rampenfunktion

Antriebe für mechatronische Maschinen werden oft mit Rampenfunktionen beaufschlagt. In der Praxis resultieren sie im Geschwindigkeitsverlauf aus der Begrenzung der Beschleunigung.

Das Integrieren zweier entgegengesetzter, um die Zeit  $T_R$  verschobener Sprungfunktionen hat eine Rampenfunktion als Ergebnis. Abbildung 3.7 zeigt diese Schaltung.

Mittels der Rampenfunktion lässt sich über den Schleppabstand im eingeschwungenen Zustand

die Kreisverstärkung  $K_0$  des Lagereglers nach Gleichung 3.4 ermitteln [71].

$$K_0 = \frac{1}{e} \quad (3.4)$$

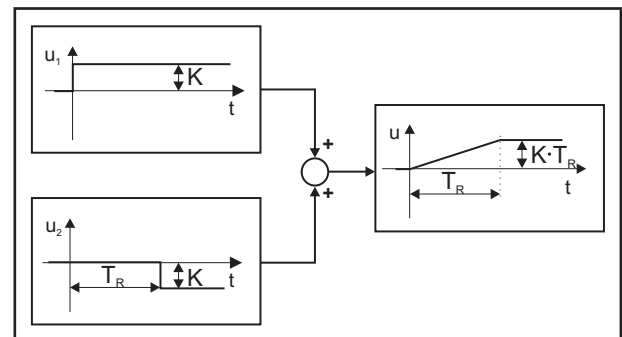


Abbildung 3.7: Generierung einer Rampenfunktion [71]

### Rauschsignal

Rauschsignale erregen das System mit einem Spektrum an Frequenzen. Die Erzeugung von Rauschsignalen ist durch unterschiedliche mathematische Methoden möglich. Im Anschluss wird eine Methode nach Paton [51] vorgestellt. Das Verfahren ist in der Abbildung 3.8 dargestellt.

Es beruht auf einem rückgekoppelten Schieberegister aus binären Speicherelementen. Mit der richtigen Rückkopplung erhält man einen Zyklus aus  $2^n - 1$  Takten. Dabei ist  $n$  die Anzahl der genutzten Speicherelemente. Für  $n = 8$  sind die Elemente 4 und 5 sowie 6 und 8 XOR und diese Ergebnisse NXOR zu verknüpfen. Das resultierende Ergebnis ist auf das erste Speicherelement zurückzuführen. Am Ausgang des 8. Speicherelementes kann ein *Pseudo-random binary signal* (PRBS) abgegriffen werden, dessen erste Takte in Abbildung 3.9 (a) dargestellt sind. Die PRBS-Folge erscheint über einen kurzen Zeitraum zufällig, wiederholt sich jedoch nach einem Zyklus von  $2^n - 1$  Takten. Jedes Bitmuster des Registers ist in einem Zyklus einmalig. Mittels eines Analog-/digital-Wandlers wird

dieses Bitmuster in eine PRNS-Folge (*Pseudo-random number signal*) gewandelt. Abbildung 3.9 (b) zeigt diesen Verlauf mit einem zusätzlichen Offset. Die PRNS-Folge kann als Testfunktion dienen. Die Taktlänge  $T_{aPRNS}$  sowie die Amplitude des Signals dienen als variable Parameter der Funktion.

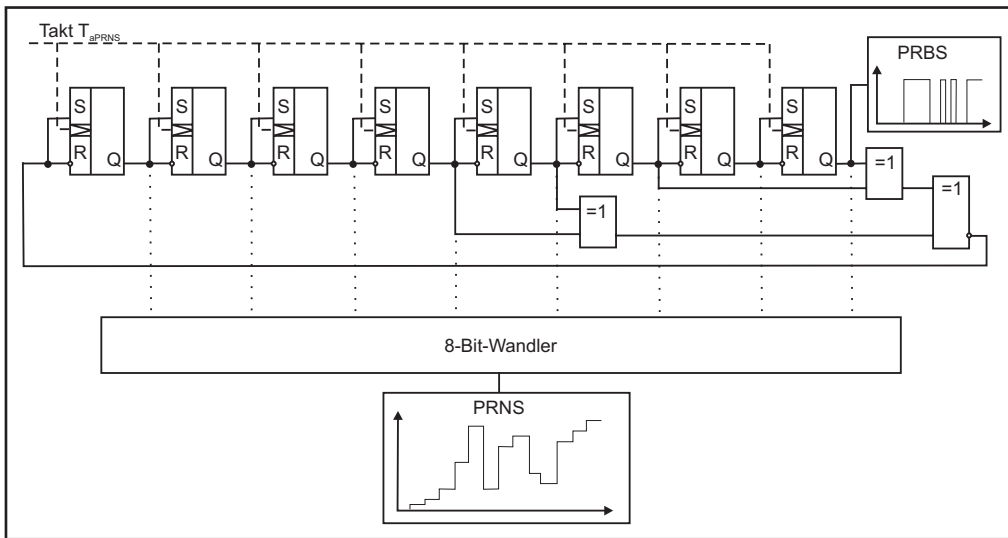


Abbildung 3.8: Pseudo-random number generator

Der *Pseudo-random number generator* (PRNG) bietet den Vorteil, dass er bei gleicher Initialisierung<sup>1</sup> immer die gleiche Zahlenfolge liefert. Ein Aufzeichnen ist so nicht nötig.

Die System- und Parameterbestimmung erfolgt über die ab Seite 38 beschriebene Simulation.

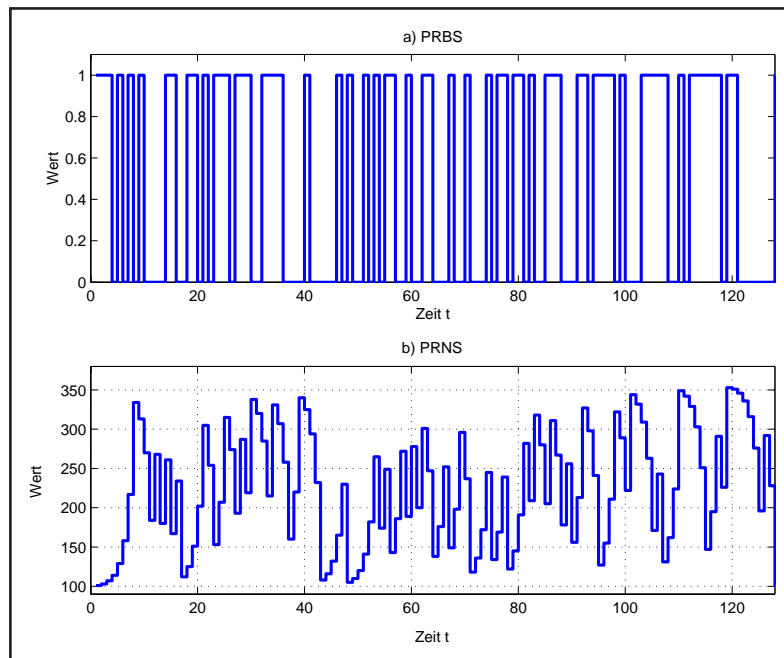


Abbildung 3.9: Signalverläufe am PRBS-Ausgang (a) und am PRNS-Ausgang (b)

<sup>1</sup> Im Normalfall werden zum Zeitpunkt  $t = 0$  alle Speicherelemente auf null gesetzt.

### Sinusfunktion

Der Sinus eignet sich ebenfalls zur Identifikation von Systemen. Anhand der Phasenverschiebung und des Amplitudengangs zwischen Testsignal und Ausgangssignal kann das System identifiziert werden. Die Verläufe von Phase und Amplitude – bei kleiner werdenden Frequenzen – werden im Bode-Diagramm dargestellt. Die Bestimmung der Parameter erfolgt über die Auswertung der Kurvenverläufe im Bode-Diagramm [19]. In Abbildung 3.10 ist ein Auszug eines Sinussignals mit zugehöriger Systemantwort dargestellt.

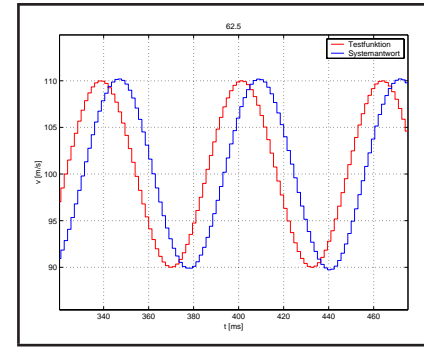


Abbildung 3.10: Sinussignal

#### 3.1.2.2 Modellstruktur

Über den Verlauf der Systemantwort bei bekannter Testfunktion lässt sich das System klassifizieren. Unterschieden wird dabei in Proportional-, Integral- und Differenzialverhalten. Für die Modellbestimmung eignet sich die Anregung der Regelstrecke über eine Sprungfunktion.

Untersucht wird hierbei der statische Endwert  $x(t \rightarrow \infty)$  der Sprungantwort. Läuft dieser Wert gegen unendlich, so liegt ein integrierendes Verhalten vor. Läuft der Wert für  $x(t \rightarrow \infty)$  gegen einen neuen statischen Endwert, ungleich null, so hat das System ein proportionales Verhalten. Nähert sich  $x(t \rightarrow \infty)$  dem Wert null an, so ist das System von differenzierendem Charakter. Die drei Grundformen sind in Abbildung 3.11 mit Verzögerung zweiter Ordnung dargestellt. Auch Zusammensetzungen sind möglich.

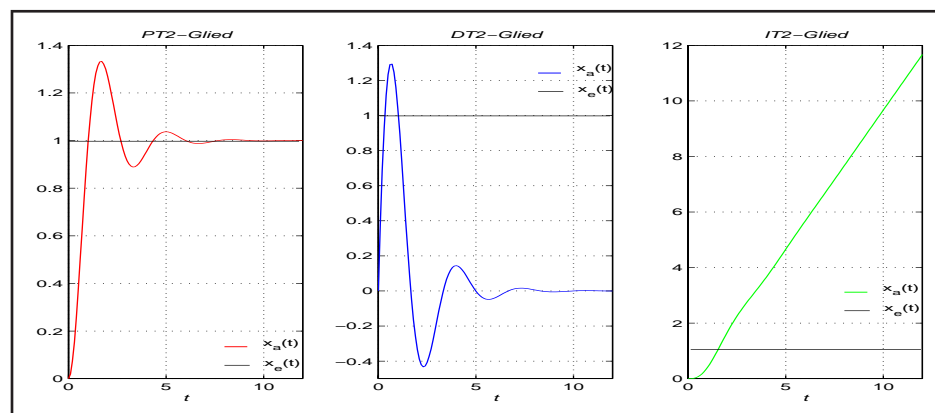


Abbildung 3.11: Darstellung der Grundsysteme

Die Ordnung des Systems lässt sich auf theoretischem Weg näherungsweise über den Anstieg der Antwort zum Zeitpunkt  $t = 0$  bestimmen. Ist der Anstieg ungleich null, so handelt es sich um ein System mit Verzögerung erster Ordnung, andernfalls um ein System höherer Ordnung.

Ist die Struktur ermittelt, so sind im Anschluss deren Parameter zu bestimmen.

### 3.1.2.3 Ausgewählte mathematische Methoden zur Bestimmung der Modellparameter

Die Bestimmung der Parameter erfolgt in der Praxis bevorzugt nach heuristischen Verfahren. Im Anschluss soll auf drei klassische, kontinuierliche Verfahren zur Bestimmung der Modellparameter eingegangen werden. Alle setzen eine Anregung des Systems mit der Sprungfunktion voraus. Die Verfahren sind auch für diskrete Systeme geeignet, bei denen die Bedingung 3.5 erfüllt ist.

$$T_a \leq \frac{T_\Sigma}{2} \quad (3.5)$$

Sie gilt für die Identifikation ebenso wie für die Reglerberechnung [41]. Bei der Auswahl der Methoden wurde Wert auf eine einfache Implementierung in Rechnern gelegt.

Da in der Antriebstechnik die Zeitkonstanten im Bereich der Zykluszeit von Steuerungen liegen können, werden darauf folgend Methoden zur Berechnung der Parameter diskreter Modelle vorgestellt.

Die Bestimmung des statischen Übertragungsfaktors  $K_S$  erfolgt durch Anregung mittels Sprung nach:

$$K_S = \frac{x(t \rightarrow \infty)}{u(t > 0)} \quad (3.6)$$

Regelstrecken des Lagereglers sind vorrangig drehzahlregelt. Sie haben proportionalen Charakter. Die ausgewählten Methoden berücksichtigen dies.

#### 63 %-Methode

Die 63 %-Methode wird zur Berechnung der Zeitkonstante  $T$  von Proportionalgliedern mit Verzögerung 1. Ordnung eingesetzt. Voraussetzung ist, dass es sich bei der zu untersuchenden Kurve um die Sprungantwort  $h(t)$  handelt und dass sich das System im eingeschwungenen Zustand befindet.

Für kontinuierliche Systeme liefert die Methode in Abbildung 3.12 gute Ergebnisse. Der Algorithmus ist einfach und leicht zu implementieren. Für diskrete Systeme muss zwischen den Punkten interpoliert werden um den 63 %-Wert zu ermitteln. Zweckmäßig ist hier die Geradeninterpolation. Auch für Strecken höherer Ordnung liefert diese Methode schnell überschlägige Werte für die Summenzeitkonstante.

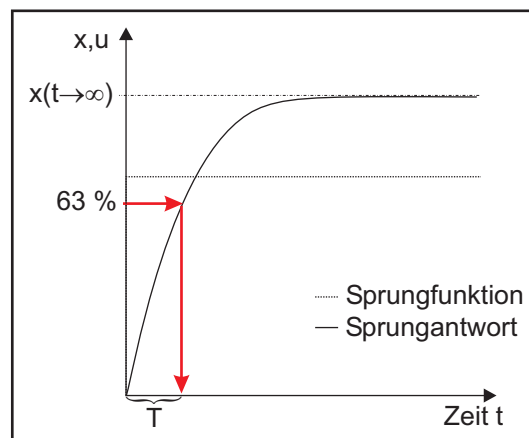


Abbildung 3.12: 63 %-Methode

#### Proportionalglieder mit Verzögerung höherer Ordnung

Es ist in der Praxis üblich,  $PT_n$ -Glieder mit  $n \geq 2$  in ihrer Ordnung zu reduzieren. Die Herkunft der Zeitkonstanten eines Antriebsstrangs kann bei Reduzierung auf ein  $PT_2$ -Modell wie folgt zugeordnet werden. Die kleine Zeitkonstante entspricht der Summe der elektrischen Verzögerungen im Antrieb. Die größere wird durch Verzögerungen im Mechanikstrang hervorgerufen. Da diese Systeme

schwingfähig sein können, wird hier in Systeme mit der Dämpfung  $D \geq 1$ , dies entspricht asymptotischem Systemcharakter, und in Systeme mit einer Dämpfung  $0 < D < 1$ , den schwingfähigen Systemen, unterschieden.

### Strecken mit asymptotischem Verhalten

Für eine asymptotische Übergangsfunktion lautet die Übertragungsfunktion:

$$G(s) = \frac{K_s}{T_1 T_2 s^2 + (T_1 + T_2)s + 1} \quad (3.7)$$

Die Normierungen im Aktor-Sensor-System erfolgen so, dass sich ein Streckenübertragungsfaktor von  $K_s = 1$  ergibt. Die Differenzialgleichung lautet:

$$(T_1 T_2) \frac{d^2 x(t)}{dt^2} + (T_1 + T_2) \frac{dx(t)}{dt} + 1 = u(t) \quad (3.8)$$

Für die analytische Prozessantwort auf einen Einheitssprung nach 3.3 gilt:

$$h(t) = 1 - \frac{T_1}{T_1 - T_2} e^{-\frac{1}{T_1} t} + \frac{T_2}{T_1 - T_2} e^{-\frac{1}{T_2} t} \quad (3.9)$$

Die freien Parameter  $T_1$  und  $T_2$  können durch zwei signifikante Punkte des Übergangsvorganges des Prozesses ermittelt werden. Zu diesem Zweck werden Punkte  $P_n(t_n, h_n)$  der normierten Sprungantwort nach Abbildung 3.13 definiert. Das Einsetzen der Punkte ergibt das Gleichungssystem:

$$\begin{aligned} h_{20} &= 0.2 = 1 - \frac{T_1}{T_1 - T_2} e^{-\frac{1}{T_1} t_{20}} + \frac{T_2}{T_1 - T_2} e^{-\frac{1}{T_2} t_{20}} \\ h_{80} &= 0.8 = 1 - \frac{T_1}{T_1 - T_2} e^{-\frac{1}{T_1} t_{80}} + \frac{T_2}{T_1 - T_2} e^{-\frac{1}{T_2} t_{80}} \end{aligned} \quad (3.10)$$

Dieses Gleichungssystem ist nicht trivial lösbar. Die Lösung kann numerisch erfolgen und tabellarisch dargestellt werden [20]. Für dieses Verfahren stehen unter dem Begriff "Zeitprozentmethode" bereits Diagramme zur Verfügung [25], aus denen sich die Werte der Zeitkonstanten herauslesen lassen.

### Schwach gedämpfte Strecken

Für schwach gedämpfte Strecken lautet die Übertragungsfunktion:

$$G(s) = \frac{K_s}{T^2 s^2 + 2DTs + 1} \quad (3.11)$$

Gesucht werden die Zeitkonstante  $T$  und die Dämpfung  $D$ . Dazu wird das System mit einem Einheitssprung angeregt. Die Bestimmung dieser Parameter wird im Punkt  $P_m(t_m, h_m)$  durchgeführt, Abbildung 3.14. Wird der Punkt  $P_m$  in den analytischen Kurvenverlauf des Systems

$$h(t) = 1 - e^{-\frac{D}{T} t} \left( \cos\left(\frac{\sqrt{1-D^2}}{T} t\right) + \frac{D}{\sqrt{1-D^2}} \sin\left(\frac{\sqrt{1-D^2}}{T} t\right) \right) \quad (3.12)$$

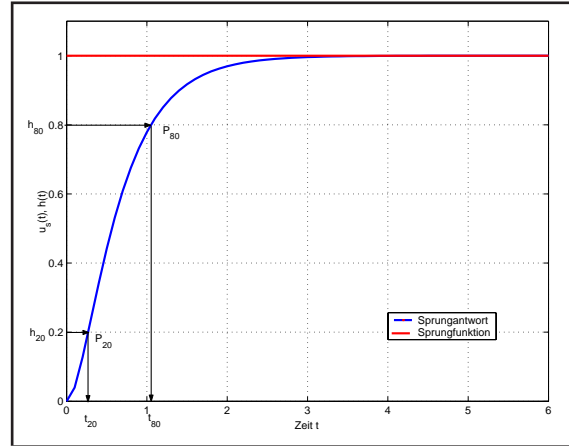


Abbildung 3.13: Asymptotische Übergangsfunktion

mit einem Übertragungsfaktor  $K_s = 1$  eingesetzt, so ergibt sich ein Gleichungssystem 2. Ordnung. Die Lösung entspricht den freien Parametern  $T$  und  $D$ . Nach einmaligem Ableiten der Übergangsfunktion im Punkt  $P_m$  wird (3.12) zu:

$$\frac{dh(t_m)}{dt} = g(t_m) = 0 = e^{-\frac{D}{T}t_m} \frac{1}{T\sqrt{1-D^2}} \sin\left(\frac{\sqrt{1-D^2}}{T}t_m\right) \quad (3.13)$$

Wenn der periodische Teil der Gleichung gleich null und

$$\frac{\sqrt{1-D^2}}{T} \cdot t_m = \pi \quad (3.14)$$

ist, ist die Gleichung erfüllt. Die erste Bestimmungsgleichung heißt dann:

$$T = \frac{\sqrt{1-D^2}}{\pi} \cdot t_m. \quad (3.15)$$

Mit der Bedingung (3.14) lässt sich die Übergangsfunktion im Punkt  $P_m$  beschreiben:

$$h_m = 1 + e^{-\frac{D}{T}t_m}. \quad (3.16)$$

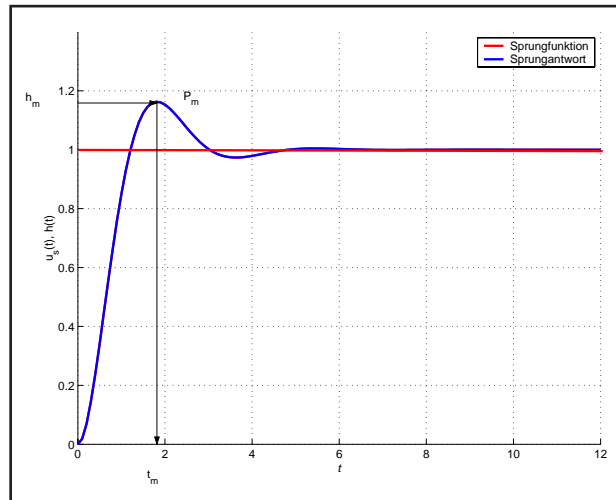


Abbildung 3.14: Schwach gedämpfte Systeme

Anstatt der absoluten Amplitude  $h_m$  wird nun nur die Überschwingweite  $h_{\ddot{u}} = h_m - 1$  betrachtet. Mit dem Einsetzen von (3.14) und anschließendem Zusammenfassen entsteht eine zweite Lösungsgleichung zur Berechnung der Dämpfungskonstanten  $D$ :

$$D = \sqrt{\frac{\ln^2 h_{\ddot{u}}}{\pi^2 + \ln^2 h_{\ddot{u}}}}. \quad (3.17)$$

Mit den Parametern  $D$  und  $T$  ist die Strecke identifiziert.

### 3.1.2.4 Identifikation mittels Simulation

Die Simulation hat aufgrund der stetig wachsenden Rechenleistung von Computern für die Regelungstechnik zunehmend an Bedeutung gewonnen [42]. Immer komfortabler werdende Werkzeuge unterstützen diesen Trend. Als Beispiel ist MATLAB Simulink zu nennen. Durch Simulation können komplizierte technische Prozesse nachgebildet werden. Das erhöht das Verständnis und lässt Fehler im System frühzeitig erkennen. Es ist schon in der Konstruktionsphase möglich, ein Modell zu erstellen und das Verhalten/Zeitverhalten in kritischen Situationen zu analysieren. Für die Regelungstechnik heißt das beispielsweise, dass kritische Reglereinstellungen erst simuliert werden, bevor diese auf die Maschine angewendet werden [40, 80].

Der Grundgedanke der Identifikation mittels Simulation ist, ein reales System über ein parametrisches, mathematisches Modell nachzubilden. In einem iterativen Prozess nach Abbildung 3.15 wird der Parametersatz so verändert, dass Modellausgang und Streckenausgang im zeitlichen Verlauf nach Anregung mit einer Testfunktion hinreichend übereinstimmen. Das parametrische Modell entspricht dann annähernd der Strecke für diese Testfunktion. Die Identifikation ist abgeschlossen, wenn die Fehlerbewertung eine Schwelle unterschreitet. In der sich anschließenden Reglerdimensionierung wird auf die Ergebnisse zurückgegriffen.

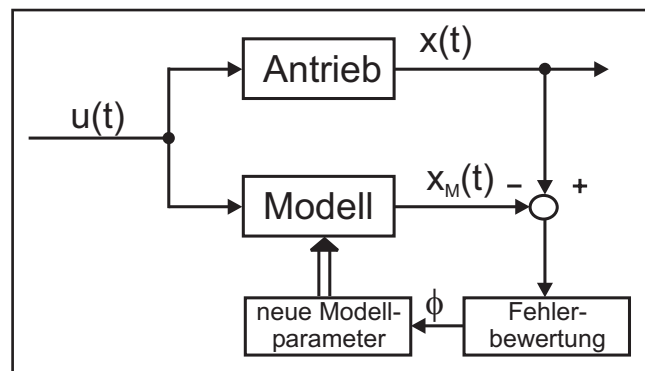


Abbildung 3.15: Identifikation mittels Simulation

Der Suchprozess entspricht einer Optimierungsaufgabe. Das Optimum ist in einem kleinen Fehler zwischen der Reaktion von Antrieb und Modell auf ein vorgegebenes Testsignal zu suchen.

Der Ablauf einer Identifikation über Simulation kann wie folgt beschrieben werden:

1. Anregen des Systems mit einer in Kapitel 3.1.2.1 vorgestellten Testfunktion.
2. Strukturermittlung und Aufstellen eines mathematischen Modells.
3. Simulieren der Reaktion des Modells auf die Testfunktion aus Schritt 1.
4. Vergleichen der Reaktionen/Prozessantworten, Bewerten des Fehlers und daraufhin:
  - (a) Verwerfen oder Ändern der Modellstruktur bei wiederholt ungenügenden Ergebnissen, anschließend mit Punkt 2 fortsetzen.
  - (b) Ändern der Parameter des Modells als Optimierungsaufgabe und zurück zu 3.
  - (c) Beenden bei Erfolg.

Für einen modularen Aufbau eines Simulationssystems zur Identifikation sind die folgenden Komponenten nötig:

- Simulator
- Modelldatenbank [46]
- Funktionsgenerator
- Optimieralgorithmus
- Gütebewertung/Fehlerbewertung/Zielfunktion.

Der Simulator bildet die Grundlage für die Simulation. Er interpretiert die Verschaltungsinformationen der Modellstruktur und realisiert den Signaldurchlauf sowie die Berechnung. Hierin liegt ein großer Teil des Know-how von Simulationssystemen. Für die Einbindung der Modelle wird eine Schnittstelle bereitgestellt.

Die Modellbibliothek enthält grundlegende allgemeine mathematische Funktionen, zum Beispiel den Summationspunkt und Strukturelemente wie direkte Verbindungen oder Verzweigungen zwischen den Elementen. Speziell für die Identifikation sind Elemente der Regelungstechnik und der Mechanik gefordert.

Ein besonderes Augenmerk ist auf den Algorithmus zur Parametervorgabe zu legen. Er muss den Anforderungen

- determiniert
- Finden des globalen Minimums/Maximums [48]
- robust gegen Rechenungenauigkeiten [50]
- Vermeiden von Abbruchfehlern [58]
- Fehlerabschätzung des Ergebnisses [58]

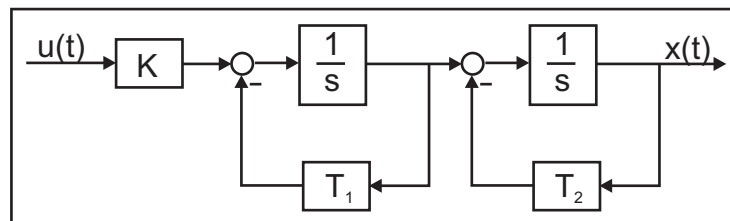
genügen. Die Wahl des Verfahrens orientiert sich an der Erfüllung der genannten Kriterien und am jeweiligen Anwendungsfall. Einige dieser Verfahren werden ab Seite 46 vorgestellt.

Die Zielfunktion bewertet die Übereinstimmung von Modell und Strecke. Sie ist nach dem speziellen Anwendungsfall aufzustellen und hat einen direkten Einfluss auf den Verlauf der Suche.

### ***Digitale Simulation von Systemen der Regelungstechnik mittels Integration***

Der Simulation kontinuierlicher Systeme mittels digitaler Rechentechnik ist zu Eigen, dass nur diskrete Zeitpunkte betrachtet werden. Die Zeitabstände zwischen den Zeitpunkten entsprechen der Taktzeit  $T_a$ . Der Kurvenverlauf dazwischen bleibt unbekannt. Es gilt jedoch, je kleiner  $T_a$  gewählt wird, desto genauer wird der Kurvenzug erfasst. Ziel ist das Ermitteln des zeitlichen Verlaufes eines Ausgangssignals  $x(t)$  auf ein Eingangssignal  $u(t)$ . Die Berechnung kann über Integrationsverfahren erfolgen. In Abbildung 3.16 ist das Blockschaltbild für ein  $PT_2$ -Element dargestellt.

Die Wahl des Integrationsalgorithmus ist ebenfalls ein entscheidender Faktor für die Genauigkeit des Verfahrens. Aus diesem Grund wird im Anschluss kurz auf die Integration eingegangen.



**Abbildung 3.16: Simulation eines  $PT_2$ -Gliedes**



Integratoren

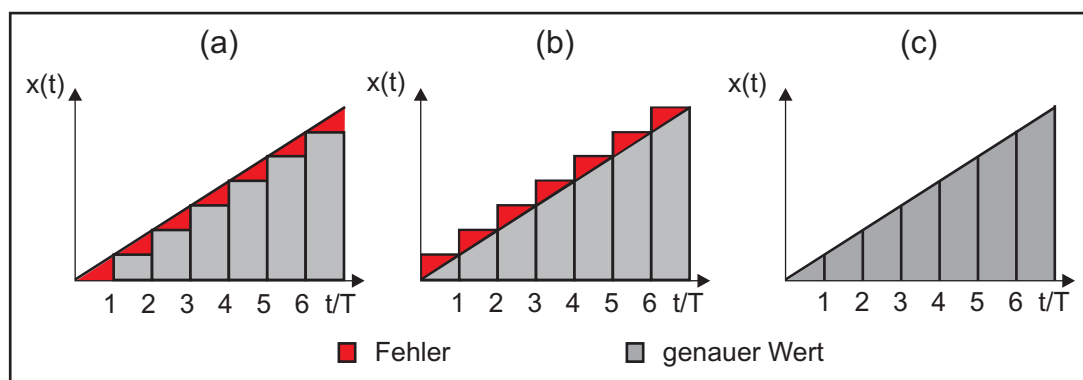
Das Integral eines analogen Signals  $x(t)$  ist die Fläche unter der Kurve von  $x(t)$ .

$$\begin{array}{ccc}
 x(t) \rightarrow & \boxed{\mathbf{I}} & \rightarrow y(t) \\
 \int & & \int
 \end{array}
 \qquad
 y(t) = K_I \int x(t) dt$$

Aufgrund der Abtastzeit ergibt sich bei diskreten Systemen für das Integral ein Fehler. Die Integration kann auf unterschiedliche Arten gelöst werden. Typisch sind die Rechteck- und Trapeznäherung. Für die Rechtecknäherung wird die Riemann-Summe [50] über alle diskreten Messwerte mit dem Zeitabstand  $T_a$  gebildet. Die Rechtecknäherung Typ I (3.18) integriert von der linken Intervallgrenze aus, während die Rechtecknäherung Typ II (3.19) von der rechten Intervallgrenze ausgeht.

$$y_k = K_I \sum_{i=0}^{k-1} x_i \cdot T_a \quad (3.18)$$

$$y_k = K_I \sum_{i=1}^k x_i \cdot T_a \quad (3.19)$$



**Abbildung 3.17:** (a) Rechteck-Integrator Typ I, (b) Rechteck-Integrator Typ II, (c) Trapez-Integrator Typ III aus [41, 56]

Wie in Abbildung 3.17 (a) und 3.17 (b) zu sehen ist, liefert der Typ I zu kleine Werte und Typ II zu große Werte für die angegebene monoton steigende Funktion. Zur Minimierung des Fehlers kann der Algorithmus durch die Trapeznäherung (3.20) ersetzt werden.

$$y_k = \frac{K_I \cdot T_a}{2} \sum_{i=1}^k (x_i - x_{i-1}) \quad (3.20)$$

Wie Abbildung 3.17 (c) zeigt, ist die Trapeznäherung für die dargestellte Anstiegsfunktion genau.

Differenziatoren

Die Differenziation wird bei digitalen Systemen über die Differenzenbildung realisiert.



Differenziatoren unterscheiden sich in Typ I (3.21) – rückwärts –, in Typ II (3.22) – vorwärts – und in Algorithmen mit Mittelwertbildung.

$$y_k = \frac{K_D}{T_a} \cdot (x_k - x_{k-1}) \quad (3.21)$$

$$y_{k-1} = \frac{K_D}{T_a} \cdot (x_k - x_{k-1}) \quad (3.22)$$

Algorithmen mit Mittelung mehrerer Differenzenquotienten blenden Störungen aus. Ein Beispiel ist der Differenziations-Algorithmus nach Typ IV [41].

Wird eine qualitative Verbesserung des Integrals angestrebt, so besteht eine Möglichkeit in der Aufnahme zusätzlicher Schätzwerte pro Zeitschritt. Diese Aufgabe erfüllen Mehrschrittverfahren, wie das *Runge-Kutta-Verfahren*, auf welches hier nicht weiter eingegangen werden soll.

Die Simulation über Integration ist für die Regelungstechnik untypisch, da der Charakter eines Übertragungsgliedes keine Rolle spielt [42]. Die Integrationsvorschrift ist für ein I-Glied die gleiche wie für ein P-Glied. Eine andere Möglichkeit der Simulation ist der Weg über die Differenzengleichung, die für jedes Glied speziell hergeleitet werden muss. Dieses Vorgehen wird in weiteren Verfahren für Systeme 1. und 2. Ordnung beschrieben.

***Digitale Simulation von Systemen der Regelungstechnik mittels Differenzengleichung***

Ziel dieser Verfahren ist die Berechnung der Reaktion  $x(t_{i+1})$  eines Systems auf eine beliebige Eingangsfunktion  $u(t_i)$  über die Differenzengleichung. Sie wird aus der z-transformierten Übertragungsfunktion

$$G(z) = \frac{X(z)}{U(z)} = \frac{b_0 + b_1 z + \dots + b_n z^{-n}}{1 + a_1 z + \dots + a_n z^{-n}} \quad (3.23)$$

abgeleitet.

Die analoge Übertragungsfunktion eines  $PT_1$ -Gliedes lautet:

$$G_S(s) = \frac{K_S}{1 + T_S s} \quad (3.24)$$

Mit der z-Transformation und einem Halteglied 1. Ordnung

$$G_S(z) = \left[ \frac{z-1}{z} \right] \cdot Z \left( \frac{G_S(s)}{s} \right) \quad (3.25)$$

ergibt sich die z-Übertragungsfunktion [41]:

$$G(z) = K_S \cdot \frac{1 - e^{-\frac{T_a}{T}}}{z - e^{-\frac{T_a}{T}}} \quad (3.26)$$

Umgewandelt in die diskrete Übertragungsfunktion der Form (3.23) durch Multiplikation mit  $z^{-1}$  erhält man:

$$G(z) = K_S \cdot \frac{(1 - e^{-\frac{T_a}{T}}) \cdot z^{-1}}{1 \cdot z^0 - e^{-\frac{T_a}{T}} \cdot z^{-1}} = \frac{X(z)}{U(z)} = \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}}. \quad (3.27)$$

Daraus leiten sich die diskreten Zeitkonstanten  $a_0, a_1$  und  $b_0, b_1$  ab:

$$\begin{aligned} a_0 &= 1 \\ a_1 &= -e^{-\frac{T_a}{T}} \\ b_0 &= 0 \\ b_1 &= 1 - e^{-\frac{T_a}{T}} \end{aligned}$$

Aus (3.27) lässt sich die Differenzgleichung für ein diskretes System 1. Ordnung ableiten:

$$x(k) = b_1 \cdot u[k-1] - a_1 \cdot x[k-1]. \quad (3.28)$$

Für ein System 2. Ordnung erfolgen die Schritte äquivalent.

$PT_2$  mit  $D \geq 1$ :

$$G_S(s) = \frac{K_S}{(1 + sT_1)(1 - sT_2)} \quad (3.29)$$

Mit (3.25) ergibt sich nach der z-Transformation die z-Übertragungsfunktion [41]:

$$G_S(z) = \frac{(x_1 T_1 - x_2 T_2) z^{-1} - \{x_1 x_2 (T_1 - T_2) - (x_1 T_1 - x_2 T_2)\} z^{-2}}{z^0 - \left(e^{\frac{T_a}{T_1}} + e^{\frac{T_a}{T_2}}\right) z^{-1} - e^{\frac{T_a}{T_1}} e^{\frac{T_a}{T_2}} z^{-2}} \cdot \frac{K_S}{T_1 - T_2} \quad (3.30)$$

mit

$$\begin{aligned} x_1 &= 1 - e^{-\frac{T_a}{T_1}} \\ x_2 &= 1 - e^{-\frac{T_a}{T_2}}. \end{aligned}$$

Wird die Differenzgleichung wie in (3.28) gebildet, so lassen sich aus (3.30) die diskreten Zeitkonstanten  $a_0 \dots a_2$  und  $b_0 \dots b_2$  ablesen.

$$\begin{aligned} a_0 &= 1 \\ a_1 &= -e^{\frac{T_a}{T_1}} - e^{\frac{T_a}{T_2}} \\ a_2 &= e^{\frac{T_a}{T_1}} e^{\frac{T_a}{T_2}} \\ b_0 &= 0 \\ b_1 &= (x_1 T_1 - x_2 T_2) \cdot \frac{K_S}{T_1 - T_2} \\ b_2 &= \left(x_1 x_2 (T_1 - T_2) - (x_1 T_1 - x_2 T_2)\right) \cdot \frac{K_S}{T_1 - T_2} \end{aligned}$$

Für schwingfähige  $PT_2$ -Glieder ( $0 < D < 1$ ):

$$G_S(s) = \frac{K_S \cdot \omega_0^2}{s^2 + 2D\omega_0 s + \omega_0^2} = \frac{K_S}{\frac{1}{w_0^2} s^2 + 2D\frac{1}{w_0} + 1} \quad (3.31)$$

mit  $T = \frac{1}{w_0}$ . Z-transformiert und dividiert durch  $z^2$  erhält man:

$$G_S(z) = K_S \frac{(1 - \beta_1)z^{-1} + (\alpha_1^2 + \beta_2)z^{-2}}{z^0 - 2\alpha_1 \cos(\omega_e T_a)z^{-1} + \alpha_1^2 z^{-2}} \quad (3.32)$$

mit

$$\alpha_1 = e^{-D\frac{1}{T}T_a} \quad (3.33)$$

$$\beta_1 = \alpha_1 \frac{\sin(\omega_e T_a + \phi)}{\sqrt{1 - D^2}} \quad (3.34)$$

$$\beta_2 = \alpha_1 \frac{\sin(\omega_e T_a - \phi)}{\sqrt{1 - D^2}} \quad (3.35)$$

$$\phi = \arccos(D) \quad (3.36)$$

$$\omega_e = \frac{1}{T} \sqrt{1 - D^2}. \quad (3.37)$$

Die Differenzengleichung lautet:

$$x[0] = b_1 u[k - 1] + b_2 u[k - 2] - a_1 x[k - 1] - a_2 x[k - 2] \quad (3.38)$$

mit

$$a_0 = 1 \quad (3.39)$$

$$a_1 = 2\alpha_1 \cdot \cos(\omega_e \cdot T_a) \quad (3.40)$$

$$a_2 = \alpha_1^2 \quad (3.41)$$

$$b_0 = 0 \quad (3.42)$$

$$b_1 = K_S \cdot (1 - \beta_1) \quad (3.43)$$

$$b_2 = K_S \cdot (\alpha_1^2 + \beta_2). \quad (3.44)$$

### ***Ausgewählte Optimierungsverfahren***

In der Identifikation mittels Simulation stellt die Parametervorgabe eine besondere Herausforderung dar. Die Effizienz und Güte der Identifikation ist maßgeblich von diesem Punkt abhängig.

Hierfür gibt es eine Vielzahl von Optimierungsverfahren. Aus der historischen Entwicklung begründet, zielen sie auf differenzierte Einsatzgebiete ab. Ein universelles Verfahren für alle Aufgabenstellungen steht zur Zeit nicht zur Verfügung [16]. Stets muss der Algorithmus auf den jeweiligen Fall angepasst werden.

Allen untersuchten Verfahren ist zu Eigen, dass sie die Qualität des Ergebnisses mit einer Zielfunktion messen, die einen numerischen Wert als Ergebnis hat. Der erfolgreiche Einsatz eines jeden Verfahrens

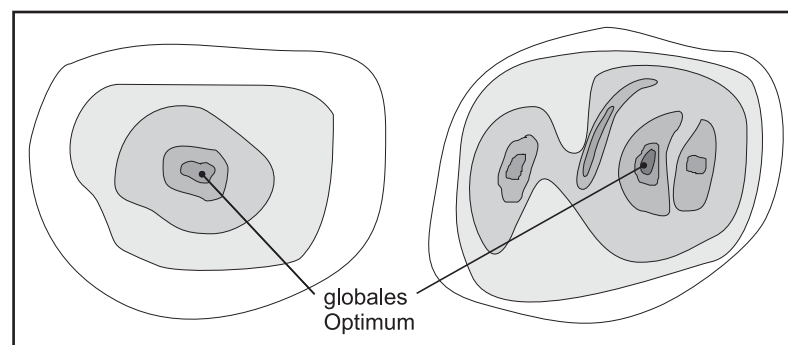
ist ebenfalls von den Eigenschaften dieser Zielfunktion abhängig. Die Zielfunktion ist problemspezifisch. Durch sie wird bestimmt, welcher Parametersatz beispielsweise der "bessere" ist. Werden bei einer Optimierung mehrere Ziele gleichzeitig verfolgt, so muss mit dem Auftreten von Pareto-Flächen gerechnet werden.

Die Zielfunktion lässt sich für zwei gesuchte Parameter  $a, b$  als eine Fläche vorstellen, die für den Wertebereich von  $a$  und  $b$  definiert ist. Der Wert der Zielfunktion über die Parameter abgetragen bildet die Gütefläche. Sind mehr als zwei Parameter an der Suche beteiligt, so handelt es sich um eine Fläche auf einer Hyperkugel.

Ziel der Optimierung ist es, das Minimum oder Maximum dieser Fläche zu finden. Welches von beiden als Optimalkriterium genutzt wird, ist nicht ausschlaggebend. Es gilt:

$$\text{Max}\{F(a, b)\} = -\text{Min}\{-F(a, b)\}. \quad (3.45)$$

Wird nach dem Minimum gesucht, entspricht die tiefste Stelle der Fläche dem Optimum. Flächen wie die in der Abbildung 3.18 links sind in der Praxis eher selten. Es treten häufig lokale Minima, Schluchten, Plateaus, Senken, Täler und Grate auf, ähnlich der Abbildung 3.18 rechts. Aufgabe des Verfahrens ist es, mit seltenen Formationen umgehen zu können und das globale Minimum zu finden. Eine weitere Schwierigkeit besteht im Auftreten der bereits erwähnten Pareto-Flächen. Sie haben die Eigenschaft, dass alle auf ihr befindlichen Punkte den gleichen Wert der Zielfunktion annehmen. Eine Verbesserung eines Parameters bewirkt die Verschlechterung eines anderen. In der Literatur [16, 61] wird die Einführung von Randbedingungen (Restriktionen) und der Wichtung der Parameter vorgeschlagen, um so ein für den jeweiligen Anwendungsfall optimales Ergebnis zu erzielen. Bei der Nutzung von Restriktionen kann es sein, dass sich das Optimum innerhalb der Einschränkung befindet, dann wird nur das Suprema, die kleinste obere Grenze oder entsprechend die größte untere Grenze – Infimum – gefunden.



**Abbildung 3.18: Fläche zweier Zielfunktionen**

Ist das Optimum mit einer zeitlichen Abhängigkeit behaftet, so reicht eine statische Optimierung nicht aus, es muss auf dynamische Optimierungsverfahren zurückgegriffen werden. Angewendet auf die Identifikation von Antrieben, liegt dieser Fall bei der Änderung der bewegten Massen, zum Beispiel durch einen Bearbeitungsvorgang, vor.

Statische Optimierungsverfahren lassen sich unter anderem in die Gruppen

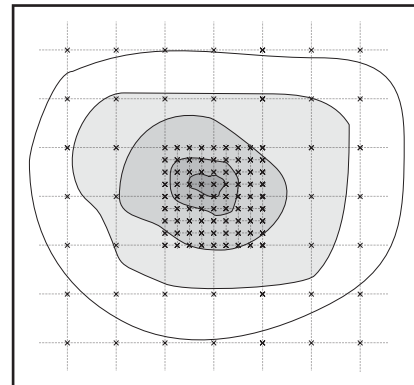
- Rastersuche
- Gradientenverfahren
- stochastische Verfahren
- Evolutionsstrategien

einteilen.

### Rastersuche

Bei der Rastersuche wird die Parametervorgabe so geändert, dass ein Parameterraum in bestimmten Schrittweiten vollständig durchsucht wird. Die vorgegebenen Werte haben diskreten Charakter. Es besteht die Gefahr, das globale Minimum bei zu groß gewählter Schrittweite zu überspringen.

Als Startbedingungen sind die Rasterweite und ein Zielgebiet, in dem das Optimum vermutet wird, anzugeben. Je größer das Zielgebiet, die Dimension, und je feiner die Schrittweite ist, desto zeitaufwändiger ist die Suche. Die Rasterweite ist umgekehrt proportional zur Wahrscheinlichkeit, mit der das Minimum getroffen wird. Erweiterungen dieser Methode suchen in nachfolgenden Schritten mit kleiner werdender Schrittweite und eingeschränktem Zielgebiet in den gefundenen Senken weiter. Abbildung 3.19 stellt das Prinzip dar.



**Abbildung 3.19: Rastersuche**

### Monte-Carlo-Methode

Die Monte-Carlo-Methode oder auch das "Trail-and-Error-Verfahren" gibt mittels Zufallszahlen Parameter eines vorher festzulegenden Zufallsraumes vor [63, 76]. Monte-Carlo-Verfahren werden beispielsweise zur Lösung schwieriger Differenzialgleichungen oder zur Berechnung von Integralen genutzt. Der Zufallsraum ist als der Zahlenbereich zu sehen, in dem die gesuchten Parameter nach heuristischen Kenntnissen vermutet werden. Von jedem vorgegebenen Parametersatz wird die Qualität bestimmt. Der beste Parametersatz wird gespeichert. Das Verfahren ist nach einer vorzugebenen Schranke beendet. Schranken können die Anzahl der Proben oder der Qualitätswert sein. Nach Schwefel [63] schneiden die Rastersuche und Zufallssuche bei einer Trefferwahrscheinlichkeit von 63% gleich ab. Bei einer geforderten Trefferwahrscheinlichkeit von 90% benötigen Rastermethoden jedoch weniger Proben. Ursache hierfür ist die dichte Nachbarschaft oder bei diskreten Systemen die mögliche Wiederholung von Proben im Verlauf der Suche bei der Zufallsstrategie.

Einige Quellen [30, 43, 45] beschreiben verschiedene Erweiterungen der Monte-Carlo-Verfahren durch Informationsspeicherung und/oder sequenzielle Erweiterung. So wird beispielsweise, je nach

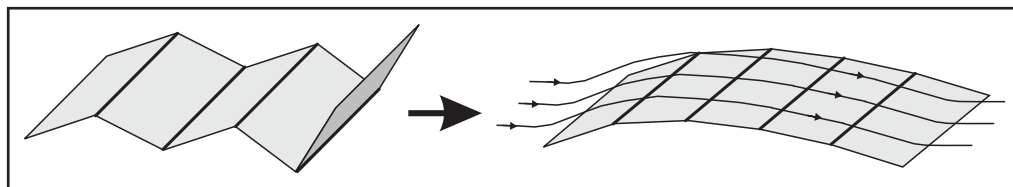
Apriorikennntnis über das Minimum, der Raum in verschieden große Volumina aufgeteilt und für jedes nur eine Probe vergeben. In vermuteter Nähe des Minimums werden die Volumina kleiner gewählt. Damit wird die Gleichverteilung der Proben aufgehoben. Nach mehreren Durchläufen wird die Suche in dem Gebiet mit den besten Werten fortgesetzt. Dieses Vorgehen erlaubt weniger Proben bei gleicher Trefferwahrscheinlichkeit.

### Gradientenverfahren

Gradientenverfahren sind Verfahren, die die Abstiegsrichtung der Gütefläche bestimmen und in diese Richtung nach einem Minimum suchen. Sie müssen in jedem Schritt eine Verbesserung des Gütewertes erreichen. Damit sind sie für Güteflächen mit Plateaus, lokalen Minima oder gar chaotischem Verhalten nicht geeignet [48, 61]. Einige Varianten einer ganzen Menge von unterschiedlichen Gradientenverfahren, auf die hier nicht weiter eingegangen werden soll, werden in [58] vorgestellt und bewertet.

### Evolutionsstrategie

Eine Alternative zu den genannten Verfahren bilden die Evolutionsstrategien nach Rechenberg [54, 55]. Rechenberg geht davon aus, dass die natürliche Evolution der Natur ein gut entwickeltes Optimierungsverfahren ist, welches auch auf technische Probleme angewendet werden kann. In Experimenten hat er verschiedene Evolutionsstrategien auf Probleme der Formoptimierung von Körpern angewendet. Dazu nutzte er beispielsweise eine Gelenkplatte, die er im Windkanal Strömungen aussetzte. Optimalkriterien waren zum Beispiel ein geringer Strömungswiderstand oder ein maximaler Auftrieb der Platte. In Abbildung 3.20 ist links die Ausgangssituation und rechts das Ergebnis der Optimierung dargestellt. Die Ergebnisse finden sich in Natur und Technik wieder.



**Abbildung 3.20: Formoptimierungs-Experiment von Rechenberg (Auftriebsprofil)**

Evolutionsstrategien sind Algorithmen, die die natürliche Evolution in unterschiedlichen Abstraktionsebenen nachbilden. Es wird davon ausgegangen, dass der Organismus mit der besten Anpassung an seine Umwelt, mit dem besten Wert seiner Zielfunktion durchschnittlich mehr Nachkommen zeugt als weniger angepasste Individuen/Parameterpaare. Ihr Erbmaterial wird sich mit der Zeit durchsetzen. Dieser Vorgang wird als Selektion bezeichnet. Um eine Entwicklung – also eine Änderung der Eigenschaften von Generation zu Generation – zu erreichen, ist das Erbmaterial zu modifizieren. Änderungen erfolgen durch das "Vermischen" der Gene der Eltern untereinander<sup>2</sup> oder auch über eine zufällige Änderung des Erbgutes durch äußere Einflüsse, wie Strahlung oder Chemikalien in der

<sup>2</sup>Rekombination – zufällige Auswahl der Gene der Eltern

Natur. Die Stärke der Änderung des Erbgutes wird durch die Mutationsrate bestimmt, in der Evolutionsstrategie wird hierbei von der Schrittweite, oder besser der Streuung, gesprochen. Mit der Streuung werden die Gene, in der Evolutionsstrategie die Objektparameter, zufällig verändert. Ob es sich dabei um eine Verbesserung der Parameter handelt, ist zufällig. Es ist nachvollziehbar, dass sich die Schrittweite im Laufe der Suche anpassen muss. Je besser das Optimum erreicht wird, desto kleiner muss die Schrittweite sein und umgekehrt. Andernfalls liegt ein schlechtes Konvergenzverhalten vor.

Die Anpasstheit/Vitalität der Individuen wird über eine Zielfunktion bestimmt.

Das Verhältnis von Eltern zu Nachkommen beschreibt den Evolutionsdruck. Er steigt, je kleiner das Verhältnis gewählt wird. Ein zu hoher Evolutionsdruck wirkt sich jedoch ungünstig auf die Artenvielfalt aus. Rechenberg empfiehlt ein Verhältnis von  $\frac{15}{100}$ .

Eine weitere Eigenschaft des Verfahrens ist die des Alterns der Eltern. Die Entscheidung, ob die Elternparametersätze nur eine Generation oder mehrere Generationen leben sollen, muss bedacht werden. Gibt es eine Verschlechterung der Zielfunktion der Kinder, so können die Eltern in den Folgegenerationen zum globalen Minimum führen. Stehen jedoch die Eltern in einer "Sackgasse", so hat das Weiterleben der Eltern negative Auswirkung auf den Optimierungsprozess. Als Lösung wird eine Alterungsfunktion der Eltern über die Vitalfunktion von Generation zu Generation vorgeschlagen.

Ein entscheidender Vorteil von Evolutionsstrategien ist, dass sie mit vielen Flächenformen der Zielfunktion umgehen können. Die Anforderungen an Optimierungsstrategien können hierbei erweitert werden durch:

- schnelle Konvergenz zum globalen Optimum und
- große Artenvielfalt über viele Generationen, um einen möglichst großen Bereich der Fläche abzusuchen.

Da die Möglichkeiten der Realisierung so vielfältig sind wie die natürliche Evolution selbst, soll im Anschluss die My- $(\mu, \lambda)$ -Strategie mit mutativer Schrittweitenregelung (MSR) nach Rechenberg [54] vorgestellt werden. Sie besitzt eine Vielzahl an Einstell- sowie Erweiterungsmöglichkeiten, die ein breites Einsatzgebiet zulassen.

Sie erzeugt aus  $\mu$  Eltern ( $E$ ) genau  $\lambda$  Nachkommen ( $N$ ). Aus den Nachkommen werden die  $\mu$  besten ( $NB_i$ ) zu Eltern selektiert und innerhalb einer Schrittweite willkürlich zu neuen Nachkommen mutiert.

Die MSR wird über einen Vektor  $\mathbf{z}$  aus normalverteilten Zufallszahlen  $z_1, z_2, \dots, z_n$  mit dem Zentralwert null, der Streuung  $\sigma = 1/\sqrt{n}$  und den Schrittweiten  $\delta_{N1}$  bis  $\delta_{N\lambda}$  realisiert. Die Schrittweiten sind Bestandteil des Parametersatzes und werden von Generation zu Generation mutiert:

$$\delta_{N1}^g = \delta_{Ei}^g \cdot \xi_1, \quad \delta_{N2}^g = \delta_{Ej}^g \cdot \xi_2, \quad \dots \quad \delta_{N\lambda}^g = \delta_{Ek}^g \cdot \xi_\lambda \quad (3.46)$$

mit

$$i, j, k = \text{ran}[1, 2, \dots, \mu]. \quad (3.47)$$



Für die Größen  $\xi_1$  bis  $\xi_\lambda$  schlägt Rechenberg im Programmiermodus folgende Berechnung vor:

$$IF \ RND < 0,5 \ THEN \ \xi_i = \alpha \ ELSE \ \xi_i = 1/\alpha. \quad (3.48)$$

Für  $\alpha$  wird ein Wert von 1,3 empfohlen.

Die Mutation des Parameter-Vektors  $\mathbf{x}$  der Nachkommen erfolgt dann so, dass die Werte der Eltern mit dem Produkt aus Schrittweite und normalverteilter Zufallszahl addiert werden.

$$\mathbf{x}_{N1}^g = \mathbf{x}_{Ei}^g + \delta_{N1}^g \cdot \mathbf{z}_1, \quad \mathbf{x}_{N2}^g = \mathbf{x}_{Ej}^g + \delta_{N2}^g \cdot \mathbf{z}_2, \quad \dots \quad \mathbf{x}_{N\lambda}^g = \mathbf{x}_{Ek}^g + \delta_{N\lambda}^g \cdot \mathbf{z}_\lambda \quad (3.49)$$

Die Auswahl der neuen Eltern aus den Nachkommen erfolgt über die Ergebnisse der Zielfunktion. Die  $\mu$  besten Nachkommen werden Eltern. Vererbt wird der Parametervektor  $\mathbf{x}$

$$\mathbf{x}_{E1}^{g+1} = \mathbf{x}_{NB1}^g, \quad \mathbf{x}_{E2}^{g+1} = \mathbf{x}_{NB2}^g, \quad \dots \quad \mathbf{x}_{E\mu}^{g+1} = \mathbf{x}_{NB\mu}^g \quad (3.50)$$

und die Schrittweite  $\delta$ :

$$\delta_{E1}^{g+1} = \delta_{NB1}^g, \quad \delta_{E2}^{g+1} = \delta_{NB2}^g, \quad \dots \quad \delta_{E\mu}^{g+1} = \delta_{NB\mu}^g. \quad (3.51)$$

$B1$  ist der Index des Nachkommen mit der besten Zielfunktion,  $B2$  der mit der zweiten und so weiter.

Initialisiert werden die Eltern mit zufälligen Werten. Ist der Zielbereich für eine Aufgabe bekannt, so können sie auch grob in dieses Gebiet gesetzt werden.

### **Abbruchkriterien**

Bei allen diesen iterativen Verfahren ist das Abbruchkriterium, dessen Erfüllung die Suche als erfolgreich oder nicht erfolgreich kennzeichnet, ein entscheidendes Problem [16, 61, 58, 63]. Wird die Suche zu früh beendet, so ist der Fehler zum Optimum recht groß. Wird die Suche hingegen später beendet, so wird wertvolle Rechenzeit verschwendet.

Friedgen [16] und Schwefel [63] brechen die Evolutionsstrategie ab, wenn sich die Steuergrößen nicht mehr entscheidend ändern. Eine alternative Möglichkeit bei der Evolutionsstrategie ist das nahe Beieinanderliegen der Nachkommen, so dass die Artenvielfalt gering ist. Dies ist der Fall, wenn alle Zielfunktionen der Elterngeneration und die jeweiligen Parameter innerhalb einer Spanne sind.

### **Zielfunktionen**

Wie bereits erwähnt, erfolgt die Bewertung des Ergebnisses über eine Zielfunktion. Sie liefert einen numerischen Wert, der die Güte des Modells bestimmt. Die Zielfunktion orientiert sich an den zu erreichenden Zielen und muss speziell dafür aufgestellt werden. Für den vorliegenden Fall besteht das Ziel, dass die Kurvenverläufe der Systemantwort und des Modells gut übereinstimmen. Die Zielfunktion berechnet den Fehler zwischen den Kurven.

$$\Phi = \sum_{i=1}^n |v_{Antrieb_i} - v_{Modell_i}| \quad (3.52)$$

Dies entspricht einem Aufsummieren der Fehler über alle Punkte der Kurve. Sollen größere Fehler stärker bewertet werden, ist die Anwendung der absoluten Fehlerquadratsumme sinnvoll:

$$\Phi = \sum_{i=1}^n \left\{ v_{Antrieb_i} - v_{Modell_i} \right\}^2 \quad (3.53)$$

Nachteilig bei (3.52) und (3.53) ist die Abhängigkeit des Fehlers von der Anzahl der Wertepaare. Ein direkter Vergleich ist so bei unterschiedlichen Tracelängen nicht möglich. Um diesen Einfluss zu minimieren, ist durch die Anzahl  $n$  der Werte zu dividieren.

$$\Phi = \frac{\sum_{i=1}^n \left\{ v_{Antrieb_i} - v_{Modell_i} \right\}^2}{n} \quad (3.54)$$

Eine prozentuale Aussage der Güte, beispielsweise die Kurven entsprechen sich zu 80 %, ist damit jedoch auch nicht möglich.

Andere Untersuchungen [18] zu einer weiteren Fehlerberechnungsmethode beziehen sich speziell auf die normierte Sprungantwort eines Proportionalgliedes. Bei dieser Methode werden Fehler im Modell bei 60 %–95 % des statischen Endwertes besonders streng bestraft.

### 3.1.2.5 Nichtlinearitäten

Nichtlinearitäten ändern das ideale Systemverhalten ab und erschweren eine mathematische Beschreibung. Im Fall elektrischer Antriebe mit verbundener Mechanik werden Nichtlinearitäten unterteilt in antriebsseitige und mechanische Nichtlinearitäten. Zu denen im Antrieb gehören vor allem vom Anwender vorgegebene Begrenzungen sowie der Drehzahl-Momentenverlauf. Zu den mechanischen Nichtlinearitäten werden Lose und Reibung gezählt. Im Anhang C werden sie kurz betrachtet. Weitere, jedoch hier vernachlässigte Größen sind:

- die Zeitvarianz von Streckenparametern<sup>3</sup>
- eingeschleppte Nichtlinearitäten von Messsystemen [59]
- der Einfluss von Temperatur
- der Jitter bei Transportzeiten und der Programmausführung.

#### ***Begrenzungen im Antrieb***

Begrenzungen im Antrieb werden vor allem zur Vermeidung von Überlasten und zur Schonung der Mechanik eingesetzt. Sie dienen der Anpassung von Motor und Maschine. Zur Anwendung kommen in der Steuerung unter anderen die folgenden Begrenzungen:

- Drehmomentenbegrenzung

---

<sup>3</sup> Änderung von angetriebenen Massen, zum Beispiel bei der Zerspanung

- Leistungsbegrenzung
- Strombegrenzung
- Drehzahlbegrenzung
- Beschleunigungsbegrenzung durch Hochlaufgeber.

### ***Substitution von nichtlinearen Effekten während der Identifikation***

Nichtlineare Effekte sollen vor allem in der Phase der Identifikation minimiert werden. Hierzu werden im Folgenden einige Anweisungen formuliert.

Begrenzungen im Antrieb sind an der Leistungsgrenze des Antriebs aktiv. Eine Faustregel besagt, dass Sollwertänderungen von 10% des maximalen Stellbereiches nicht in Begrenzungen enden. Im Einzelfall ist dies jedoch zu prüfen.

Der Einfluss von Reibung auf die Ergebnisse lässt sich minimieren, indem die Messwerte nur in Bereichen linearer Reibung ausgewertet werden. Dies gilt im Allgemeinen im Arbeitspunkt. Lose sind konstruktionsbedingt und wirken sich auf Regelungen generell nachteilig aus. Sie sind in der Identifikation zu ermitteln [6, 41, 62].

Erfahrungswerte zeigen, dass die Dynamik der mechatronischen Maschine nicht mehr hauptsächlich nur durch die Mechanik bestimmt wird, sondern von Motor, Antrieb und Regelung. Lose und Reibung fallen weniger ins Gewicht.

## **3.2 Reglerauswahl**

Die Reglerauswahl beschränkt sich auf die in der industriellen Steuerung implementierten Verfahren nach Kapitel 2. Zur Auswahl stehen ein P-Regler, ein P-Regler mit Vorsteuerung und DSC.

Der Einsatz des einfachen P-Reglers orientiert sich auf simple Bewegungsabläufe, die keine hohen Anforderungen an die Genauigkeit stellen. Die zweite Variante, ein P-Regler mit Vorsteuerung wird oft genutzt. Die Reglerinbetriebnahme konzentriert sich also auf die Parametrierung von Reglern.

## **3.3 Bestimmen der Reglerparameter**

### **3.3.1 Gütemaß von Regelkreisen**

Auch bei der Einstellung von Regelkreisen müssen Zielkriterien – Gütemaße – definiert und erreicht werden. Es kann für Führungsverhalten und für Störverhalten bestimmt werden. Das Optimum orientiert sich an den technischen Anforderungen des Prozesses. Die Betrachtungen können im Zeit- oder im Frequenzbereich erfolgen. Die Werte:

- Anregelzeit  $T_{ANR}$

- Ausregelzeit  $T_{AUS}$
- Schleppabstand
- Überschwingweite  $h_{\ddot{u}}$
- Kreisverstärkung  $K_0$

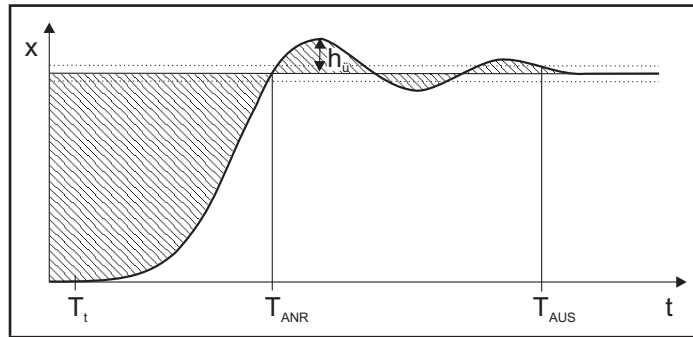


Abbildung 3.21: Gütemaße für Führungsverhalten

bestimmen das Gütemaß im Zeitbereich nach Erregung mit einem Sprung. Sie liefern jedoch keine Aussage über die Dynamik des Regelprozesses. Integrative Verfahren bewerten hingegen den gesamten Ausgleichsvorgang. Grundlage der Berechnung ist die Regelfläche. Die Regelfläche ist das Integral der Regelabweichung, genannt Lineare Regelfläche (3.55). In Abbildung 3.21 ist sie schraffiert zu erkennen. Ziel des Reglerentwurfs ist die Minimierung des angewendeten Integralkriteriums.

$$\Phi_1 = A_{lin}(t \rightarrow \infty) = \int_0^{\infty} [x_d(t) - x_d(t \rightarrow \infty)] \cdot dt \stackrel{!}{=} Min \quad (3.55)$$

Der Nachteil, dass sich positive und negative Flächen aufheben, wird durch die Betragsregelfläche behoben.

$$\Phi_2 = A_{abs}(t \rightarrow \infty) = \int_0^{\infty} |x_d(t) - x_d(t \rightarrow \infty)| \cdot dt \stackrel{!}{=} Min \quad (3.56)$$

Wird die Lineare Regelfläche minimiert, so ergibt sich für einen Standardregelkreis 2. Ordnung eine Dämpfung  $D = 0,659$  und eine Überschwingweite von  $\ddot{u} = 6,35\%$  [41]. Sollen große Regelabweichungen als besonders schlecht gewertet werden, so ist das Integral der quadratischen Regelabweichung, die Quadratische Regelfläche, zu bilden.

$$\Phi_3 = A_{sqr}(t \rightarrow \infty) = \int_0^{\infty} [x_d(t) - x_d(t \rightarrow \infty)]^2 \cdot dt \stackrel{!}{=} Min \quad (3.57)$$

$$D = 0,5$$

$$\ddot{u} = 16\%$$

Da der Ausgleichsvorgang so schnell wie möglich erfolgen soll, ist es sinnvoll, diese Flächen mit der Zeit zu gewichten.

$$\Phi_4 = A_{sqr_t}(t \rightarrow \infty) = \int_0^{\infty} [x_d(t) - x_d(t \rightarrow \infty)]^2 \cdot t \cdot dt \stackrel{!}{=} Min \quad (3.58)$$

$$D = 0,595$$

$$\ddot{u} = 9,8\%$$

Auch eine quadratische Zeitgewichtung ist möglich.

$$\Phi_5 = A_{sqrt}(t \rightarrow \infty) = \int_0^{\infty} [x_d(t) - x_d(t \rightarrow \infty)]^2 \cdot t^2 \cdot dt \stackrel{!}{=} Min \quad (3.59)$$

$$D = 0,667$$

$$\ddot{u} = 5,99\%$$

Für diskrete Systeme sind die Integrale in (3.55) bis (3.62) durch eine diskrete Operation aus Kapitel 3.1.2.4 (Seite 41) zu ersetzen. Üblich ist die Rechteck-Integration.

Ein weiteres Gütemaß für die Beurteilung von Lagereglern ist der Schleppabstand. Die Berechnung ergibt sich aus (3.4) zu:

$$e = \frac{1}{K_0} \quad (3.60)$$

Der Schleppabstand wird um so kleiner, je größer die Kreisverstärkung ist. Das daraus zu formulierende Gütemaß lautet:

$$\frac{1}{K_0} \stackrel{!}{=} Min \quad (3.61)$$

Neben der Bewertung des Führungsverhaltens kann auch das Störverhalten bewertet werden. Da ein Störsignal so wenig Einfluss wie möglich haben soll, ist eine Bewertung der Abweichung der Regelgröße von der Nullposition sinnvoll. Gewichtet mit der Zeit wird eine schnelle Ausregelung positiv bewertet.

$$\Phi_6 = A_{Zabs}(t \rightarrow \infty) = \int_0^{\infty} |0 - x_d(t \rightarrow \infty)| \cdot t \cdot dt \stackrel{!}{=} Min \quad (3.62)$$

In der Praxis werden die Reglerparameter nicht nur auf ein einzelnes Gütemaß eingestellt, sondern auf einen Kompromiss aus einer Menge von Forderungen. Bretschneider stellt in [8] eine Möglichkeit der Beurteilung bei Auswahl mehrerer Kriterien vor. Zusätzlich führt er die Schranken aus Tabelle 3.1 ein. Sie setzen Grenzen, bei deren Überschreitung das Gütemaß exponentiell verschlechtert wird.

$E_1 = h \leq h_{tol}$	Anforderung nach überschwingfreiem bzw. definiertem Überschwingen beim Positionieren
$E_4 = \epsilon \leq \epsilon_{max}$	Begrenzen der Nulldurchgänge des Geschwindigkeitsverlaufs zur Beurteilung des Störverhaltens

**Tabelle 3.1: Schranken [8]**

Die Berechnung der Gesamtgüte erfolgt dann nach

$$\phi = \Phi \cdot E$$

$$\Phi = \prod \Phi_i = \Phi_1 \cdot \Phi_2 \cdot \Phi_3 \cdot \Phi_4 \cdot \Phi_5 \cdot \Phi_6$$

$$E = \prod f_i(\text{Wert}, \text{Schranke}) \quad (3.63)$$

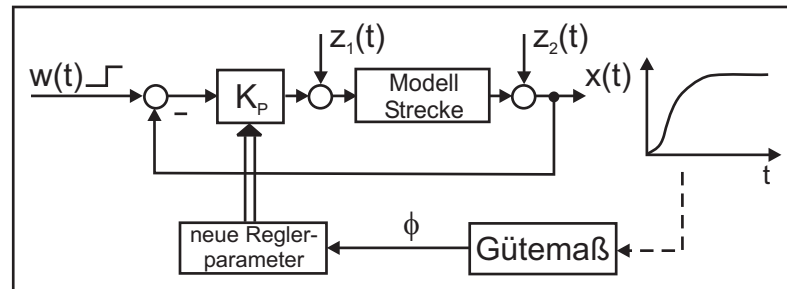
mit

$$f_i(\text{Wert}, \text{Schranke}) = \begin{cases} 1 & \text{für Wert} \leq \text{Schranke} \\ e^{\nu \cdot (\frac{\text{Wert}}{\text{Schranke}} - 1)} & \text{für Wert} > \text{Schranke} \end{cases} \quad (3.64)$$

wobei  $\nu$  als ein Verstärkungsfaktor eingeführt wird, der das Einhalten der Schranke sichern soll. Über die Festlegung des numerischen Wertes für  $\nu$  werden keine Aussagen getroffen.

### 3.3.2 Berechnung der Reglerparameter mittels Simulation

Unbehauen empfiehlt in [71] die Simulation als eine weit verbreitete Methode zur Berechnung der Reglerparameter. Insofern Startwerte benötigt werden, können diese nach Ziegler und Nichols oder nach Chien, Hrones und Reswick berechnet werden.



**Abbildung 3.22: Berechnung von Reglerparametern mittels Simulation**

Die Berechnung der Reglerparameter erfolgt hierbei ähnlich der Identifikation mittels Simulation als iterative Optimierung.

In einem ersten Schritt ist die im Reglerentwurf erstellte Reglerstruktur zu modellieren. Die Regelstrecke entspricht dem in der Identifikation erhaltenen parametrischen Modell. Aus der technischen Verwendung des Reglers ergibt sich das Gütemaß und die Anforderung nach Führungs- oder Störverhalten. Für Führungsverhalten wird nach Abbildung 3.22 die Führungsgröße  $w(t)$  in Form eines Sollwert-Sprunges auf den Regelkreis geschaltet. Ist der Regler auf Störverhalten auszulegen, so ist die Störgröße je nach erwartetem Angriffspunkt beispielsweise auf  $z_1(t)$  oder  $z_2(t)$  aufzuschalten.

## 3.4 Validierung

Sind Reglerparameter gefunden, so sind diese in einem weiteren Schritt am realen System zu testen. Es sind spezielle Sicherheitsvorschriften nach Kapitel 5.6 zu beachten, da fehlerhafte Reglereinstellungen schnell zu Gefahren für Mensch und Maschine führen.

In der Validierung ist der entworfene Kreis im Zielsystem aufzubauen und mit Testfunktionen anzuregen. Wird das geforderte Zielverhalten in engen Grenzen nicht erreicht, so ist die Modellstruktur im Reglerentwurf zu ändern und im Anschluss die Parameterberechnung erneut auszuführen.

# Kapitel 4

## Dienstleistungssysteme im Internet

### 4.1 Reglerinbetriebnahme im Netz

Das manuelle Vorgehen der Reglerinbetriebnahme konzentriert sich auf das Beobachten von Variablen und das Ziehen von Schlüssen auf Grundlage heuristischen Wissens. Ein einfaches Hilfsmittel stellt für den Inbetriebnehmer die Visualisierung dieser Variablen dar (Trace-Funktionalität). In der Visualisierung kann zwischen der Darstellung charakteristischer Größen (Überschwingweite, Anregelzeit usw.) und in die Darstellung von zeitlichen Verläufen unterschieden werden. Aus diesen Aufzeichnungen ermittelt der Anwender die Kennwerte des Antriebs grafisch oder rechnerisch. Unterstützungen bieten automatische Tools zur Inbetriebnahme an. Eine Voraussetzung ist auch hier die Übermittlung der aufgezeichneten Traces an das Visualisierungs- und Inbetriebnahmetool, wie es in der Offline-Inbetriebnahme bereits der Fall ist.

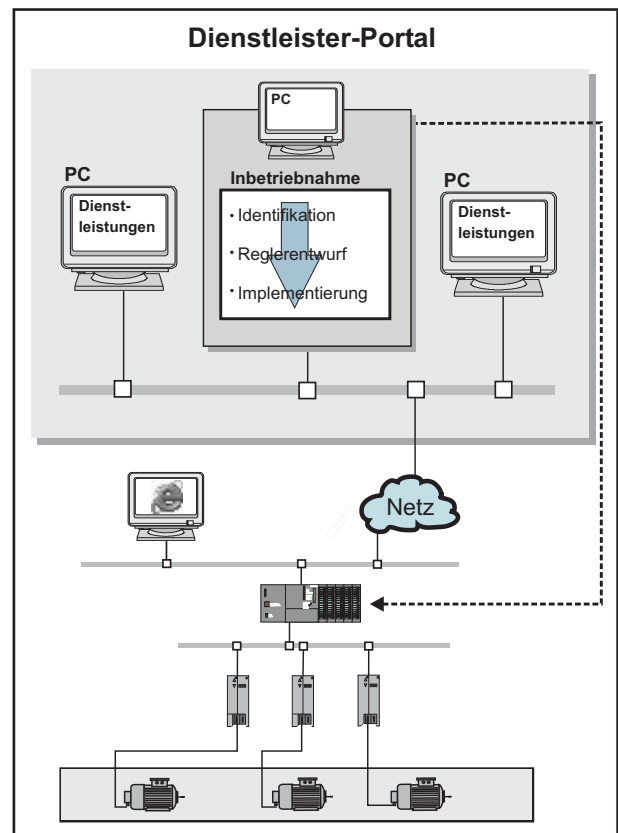


Abbildung 4.1: Reglerinbetriebnahme im System

Bisher musste zwischen Offline- und Online-Teil eine Direktverbindung hergestellt werden, die räumlich begrenzt war. Der Einzug von Ethernet und damit auch dem Internet in die Steuerungsebene, hebt diese Barriere auf. Der Transport der Soll-Istwert-Verläufe zum Auswertungsrechner kann nun weltweit erfolgen. Daraus ergeben sich völlig neue Dienstleistungsansätze, die zudem vollkommen automatisch ablaufen können. Grundgedanke ist hierbei die Verlagerung des Algorithmus/Dienstes auf Rechner im Netz. Diese Rechner gehören ei-

nem Portal an. Bisher werden die Programme auf jedem Rechner lokal abgelegt, auf dem sie benötigt werden. Der administrative Aufwand der Pflege ist hoch. Für die Reglerinbetriebnahme bietet es sich nun an, den Offline-Teil in das Netz zu verlagern. Abbildung 4.1 zeigt die Einbindung des Offline-Algorithmus in das Portal. Die Anbindung an die Echtzeitkomponente auf dem Controller erfolgt über das Internet. Zum Buchen und Ausführen des Dienstes ist ein Webbrowser auf einem Rechner nötig.

Anbieter derartiger Dienstleistungen können Steuerungshersteller, Maschinenhersteller oder externe Dienstleister sein, die dem Inbetriebnahmepersonal über diesen Weg ihr Know-how bereitstellen.

In diesem Kapitel wird ein Überblick über Verbindungen zwischen Rechnern und über Möglichkeiten der Bereitstellung von Software in TCP/IP-basierten Netzen gegeben. Anschließend erfolgt die Betrachtung typischer, realisierter E-Services für Maschinen.

## 4.2 Grundlagen für E-Services

### 4.2.1 Internet

Als das "Netz der Netze" ist das Internet bekannt. Der Begriff "Internet" steht für ein weltweites Rechnernetz. Schätzungsweise 317,6 Mio. Hosts waren zu Beginn des Jahres 2005 an diesem angeschlossen [79]<sup>1</sup>. Im Internet können Rechner unterschiedlicher Betriebssysteme miteinander kommunizieren. Dies ist zurückzuführen auf das standardisierte OSI-Kommunikationsmodell der *International Organization for Standardization* (ISO).

#### 4.2.1.1 7-Schichten-Modell

Das OSI-Referenzmodell teilt die Kommunikationsaufgabe nach der Funktionalität in die sieben Schichten aus Tabelle 4.1 auf. Diese sind bei jedem Kommunikationspartner vorhanden, müssen jedoch nicht alle genutzt werden. Das Modell beschreibt die Inhalte und Schnittstellen dieser Schichten zueinander, so dass eine Kommunikation zwischen ihnen in beiden Richtungen erfolgen kann. Nicht festgelegt ist, wie deren Inhalte umgesetzt werden. Mit der festen Vorgabe dieser Schnittstellen ist es möglich, unterschiedliche Rechnersysteme und -netze miteinander zu verbinden, ganz gleich von welchem Hersteller und mit welchem Betriebssystem.

Das Referenzmodell unterliegt einer Zweiteilung. Die Schichten 1–4 dienen ausschließlich der Übertragung der Daten zwischen Kommunikationspartnern. Den Weg zum Anwender finden die Daten über die Schichten 5–7 und die jeweiligen Applikationen.

Die **Physikalische Schicht** realisiert die Übertragung und den Empfang der einzelnen Bits über ein Medium, beispielsweise Glasfaserkabel, Luft oder Koaxialkabel. Es legt die Codierung, Spannungspegel, Signallängen der Bits und die Zuordnung zur Pinbelegung des Steckers fest.

---

<sup>1</sup> Artikel: Chronologie des Internets



Schicht	Bezeichnung	Erläuterung
7	Anwendungsschicht (Application Layer)	stellt die auf dem Netzwerk basierenden Dienste für die Programme des Endanwenders bereit (Datenübertragung, elektronische Post usw.)
6	Darstellungsschicht (Presentation Layer)	legt die Anwenderdaten-Strukturen fest und konvertiert die Daten, bevor sie zur Sitzungsbzw. Anwendungsschicht gegeben werden (Formatierung, Verschlüsselung, Zeichensatz)
5	Sitzungsschicht	definiert eine Schnittstelle für den Auf- und Abbau von Sitzungen, d. h. zur Benutzung der logischen Kanäle des Transportsystems
4	Transportschicht (Transport Layer)	stellt fehlerfreie logische Kanäle für den Datentransport zwischen den Teilnehmern bereit
3	Netzwerkschicht (Network Layer)	transportiert die Daten von der Quelle zum Ziel und legt die Wege der Daten im Netz fest
2	Datenverbindungsschicht (Data Link Layer)	legt die Datenformate für die Übertragung fest und definiert die Zugriffsart zum Netzwerk. Sie wird in die "Zugriffssteuerung für das Medium" (MAC) und die "Logische-Ankopplungs-Steuerung" (LLC) unterteilt.
1	Physikalische Schicht (Physical Layer)	definiert die elektrischen und mechanischen Eigenschaften der Leitung, Pegeldefinition

*Tabelle 4.1: OSI-Referenzmodell (Open Systems Interconnection) [60]*

Für den sicheren Transport der Daten ist die **Datenverbindungsschicht** verantwortlich. In ihr findet der Buszugriff statt. Im Anhang B sind einige ausgewählte Verfahren kurz beschrieben. Weiterhin erfolgt das Aufteilen der Daten in Pakete von einigen hundert Bytes. Diese Pakete enthalten die Nutzdaten sowie Sicherheitsinformationen, die von dieser Schicht hinzugefügt werden. Diese Zusatzinformationen dienen dem Erkennen von Übertragungsfehlern und zum Quittieren bereits erhaltener Pakete. Die Schicht realisiert weiterhin einen Überlaufschutz, falls mehr Daten empfangen als weiterverarbeitet werden können.

Die **Netzwerkschicht** übernimmt das Routing. Darunter wird die Festlegung des Weges, auf dem die Daten im Netz übertragen werden, verstanden. Hierfür existieren verschiedene Algorithmen, die jeweils unterschiedliche Anforderungen an die Wegewahl stellen (wenig Zwischenstationen, kürzester Weg).

In der **Transportschicht** werden bei verbindungsorientiertem Transport die Verbindungen aufgebaut und die Verbindungs- und Übertragungsarten realisiert. Auf diese wird in den Kapiteln 4.2.1.6 und 4.2.1.7 näher eingegangen. Weiterhin erfolgt die Flusssteuerung und Fehlerbehebung sowie eine Aufteilung der Daten der Sitzungsschicht in kleinere Pakete.

Dienste zum Aufbau und Abbau von Sitzungen werden durch die **Sitzungsschicht** bereitgestellt. Sie steuert den Zugriff eines oder mehrerer Prozesse auf den fehlerfreien logischen Kanal der Transportschicht.

Die **Darstellungsschicht** realisiert die einheitliche Darstellung der Daten auf dem Bildschirm oder Drucker. Analysiert wird hier zum Beispiel der verwendete Zeichensatz der empfangenen Daten.

Auf die **Anwendungsschicht** wird die Applikation aufgesetzt. Sie stellt die Funktionen für die Kommunikation bereit. Jeder Prozess, der Funktionen der Schicht sieben nutzt, spricht diese über eine eigene Anwenderinstanz an.

#### 4.2.1.2 IP-Adressen

Alle Rechner, die unmittelbar an das Internet angeschlossen sind, sind unter einer weltweit eindeutigen Adresse ansprechbar. Diese Adresse besteht, beim Internet Protokoll *IP – Internet suit Protocol* Version 4 (IPv4), aus einer 32-Bit-Zahl. Sie wird über vier durch Punkte voneinander getrennte Zahlen zwischen 0 und 255 dargestellt. Die Adresse

141.6.200.118

entspricht der Bitfolge

10001101 00000110 11001000 01110110.

Die Bitfolge setzt sich aus einem Netz-Teil und einem Host-Teil zusammen. Die Anzahl der Netz-Bits variiert zwischen 14 und 28 Bit. Sie geben an, in welchem Netz sich der Rechner befindet. Es gibt unterschiedliche Klassen dieser Netze. Sie unterscheiden sich durch die maximale Anzahl anschließbarer Hosts und ihrer Funktionalität. Die Netze Class-A bis Class-C sind für die allgemeine Verwendung, Class-D-Netze hingegen nur für Multicast-Anwendungen reserviert. Testzwecken dient das Class-E-Netz. Die Unterscheidung der Netze erfolgt jeweils durch die ersten Bits der IP-Adresse nach Abbildung 4.2.

Class A	0	7 Bit Network Prefix	24 Bit Host ID
Class B	10	14 Bit Network Prefix	16 Bit Host ID
Class C	110	21 Bit Network Prefix	8 Bit Host ID
Class D	1110	28 Bit Network Prefix	
Class E	1111	28 Bit Network Prefix	

Abbildung 4.2: Netzklassen im IPv4-Protokoll

Da dieses System sehr ineffektiv ist (Class-A-Netze sind zu groß, Class-C-Netze zu klein) und das Wachstum des Internets in den vergangenen Jahren überproportional gestiegen ist, wurde das System

erweitert. Eine dieser Erweiterungen ist der Wegfall der Netzklassen (*Classless Interdomain Routing* – CIDR). Mit Hilfe der dafür eingeführten Subnetzmaske können (fast) beliebig große Subnetze entstehen. Die Subnetzmaske ist gleich der IP-Adresse aufgebaut. Binär kann, von links nach rechts gelesen, die Bitfolge jedoch nur von "1" auf "0" wechseln. Die Anzahl der 1er Bits legt fest, wie viele Bits der IP die Netzadresse angeben. So sind die ersten 24 Bit der IP durch die Subnetzmaske 255.255.255.0 dem Netzwerk zugeordnet und die restlichen 8 Bit können im unterlagerten LAN vergeben werden. Von den 256 möglichen Hostadressen sind zwei durch die Netzwerkadresse (alle Host-Bits sind "0") und den Multicast-Dienst (alle Bits "1") belegt. Die Berechnung der freien IP-Adressen erfolgt nach:

$$n_{Hosts} = 2^{Anzahl\_Host\_Bits} - 2. \quad (4.1)$$

Für die Subnetzmaske 255.255.255.0 heißt dies, die Bitschreibweise

11111111 11111111 11111111 00000000

enthält acht Nullen. Nach (4.1) können  $n_{Hosts} = 254$  Hosts im Netz angeschlossen werden.

Das exponentielle Wachstum der Anzahl an das Internet angeschlossener Rechner lässt erahnen, dass der Vorrat an freien und nutzbaren IP-Adressen fast erschöpft ist. Vor allem im Asiatischen Raum ist das ein Problem. In der Erweiterung – dem IPv6 – wird der Adressraum von  $2^{32}$  auf  $2^{128}$  erweitert. Die Darstellung erfolgt nicht mehr binär, sondern hexadezimal. Gegenwärtig überwiegt noch die Nutzung des IPv4.

Zum einfacheren Merken der Adressen von Rechnern können diesen Namen zugeordnet werden. Die Zuordnung erfolgt über DNS (*Domain Name Service*). Die Adresse <http://www.tu-chemnitz.de>, der Webserver der TU Chemnitz, ist so leichter im Gedächtnis zu behalten als die IP <http://134.109.132.162>. Beide Varianten führen zur Homepage der TU Chemnitz. Realisiert wird dies mittels im Netz verteilter DNS-Server, die den Domainnamen ([www.tu-chemnitz.de](http://www.tu-chemnitz.de)) in die IP-Adresse auflösen.

### 4.2.1.3 Kommunikationsprotokolle

Die Kommunikation im Internet basiert auf dem verbindungsorientierten Dienst TCP (*Transmission Control Protocol*) oder dem verbindungslosen Dienst UDP (*User Datagram Protocol*). Verbindungslos heißt in diesem Fall, die Datenpakete werden mit Angabe der Zieladresse versendet, ohne zu wissen, wann und in welcher Reihenfolge die Pakete den Empfänger erreichen. Genutzt wird es zum Beispiel bei Video-Streams. Verbindungsorientierte Dienste (TCP) hingegen öffnen eine Verbindung zum Zielrechner, senden die Daten, warten auf eine positive Quittung und schließen erst dann wieder den Kanal. Im Fehlerfall werden die Pakete wiederholt gesendet.

Beide Protokolle werden in Zusammenhang mit IP verwendet. Im OSI-Referenzmodell lagert sich UDP und TCP in der Transportschicht an und IP in der Netzwerkschicht. Da TCP/IP älter sind als

das OSI-Referenzmodell, bilden sie nicht den Standard ab und sind nur begrenzt auf die einzelnen Schichten anwendbar.

#### 4.2.1.4 Internet-Dienste

Aufgesetzt auf das TCP/IP-Protokoll<sup>2</sup> werden dem Nutzer weitere Dienste in Protokollform bereitgestellt [70, 49]. Die Dienste erfüllen Aufgaben unterschiedlicher Natur.

Der Dienst *E-Mail* versendet elektronische Post an einen Empfänger. Dieser muss ein Postfach besitzen, welches mit einem herkömmlichen Postfach grob vergleichbar ist. Vorteil bei dieser Art der Kommunikation ist, dass beide Partner nicht gleichzeitig online sein müssen. Der Empfänger kann zu einem späteren Zeitpunkt die E-Mail abrufen.

Das *File Transfer Protocol* (FTP) ist ein Datenübertragungsprotokoll, mit dem sich von einem Rechner zu einem Server in beide Richtungen Dateien übertragen lassen. Es können Verzeichnisse angezeigt, angelegt und gelöscht werden.

Das *Hypertext Transfer Protocol* (HTTP) ist für den Zugriff auf den Multimediabereich des *World Wide Web* (WWW) zuständig. Der Datenaustausch erfolgt hier zwischen Server und Client mit einem Browser und einem Webserver.

Einige weitere Dienste sind kurz in Tabelle 4.2 beschrieben.

Dienst	Beschreibung
Telnet	Dienst zum Arbeiten auf entfernten Rechnern
Newsgroups	Diskussionsforum einer Interessengemeinschaft zu einem vorgegebenen Thema
IRC	Protokoll zum Chatten
Gopher	Dienst zum Auffinden von Informationen, hat an Bedeutung verloren
WAP	HTTP ähnlicher Dienst für Handys und PDA

**Tabelle 4.2: Weitere Internet-Dienste**

In diesem Zusammenhang ist der Begriff der URL (*Unified Resource Locator*) zu nennen. Eine URL ist eine Ortsangabe einer Ressource im Internet. Sie setzt sich aus:

$\langle \text{Dienst} \rangle : // \langle \text{server} \rangle : \langle \text{Port} \rangle / \langle \text{Pfad} \rangle ? \langle \text{Anfrage} \rangle$

zusammen [79]<sup>3</sup>. Alle Dateien und Dienste werden über dieses System genau lokalisiert. Ein Beispiel ist der folgende Link:

$http : //www.google.de/search?hl = de\&q = TU + Chemnitz.$

<sup>2</sup> UDP/IP ebenfalls

<sup>3</sup> Begriff: Uniform Resource Identifier

#### 4.2.1.5 Netze

Das **LAN** (*Local Area Network*) ist ein örtlich begrenztes, auf dem OSI-Schichtenmodell beruhendes Netz. Es stellt, wie das Internet, ein Netzwerk von Rechnern dar, muss aber nicht ans Internet angeschlossen sein. Das Computer-Netzwerk einer Firma an einem Standort ist zum Beispiel ein LAN.

Ein **WAN** (*Wide Area Network*) unterscheidet sich vom LAN nur durch seine größere Ausdehnung. So entsteht zum Beispiel durch die Verbindung zweier Firmennetzwerke in verschiedenen Städten mittels einer Standleitung ein WAN.

Der Begriff **Intranet** wurde von Netscape geprägt. Eine eindeutige Definition existiert dafür jedoch nicht. Es kann gesagt werden, dass es sich hierbei um ein internes Firmennetz handelt, das auf dem TCP/IP-Protokoll beruht. Es dient vor allem der Information der Mitarbeiter mit unternehmensspezifischen Inhalten. So können hier zum Beispiel Informationen über aktuelle News, Produktneuheiten, Telefon- und Mailverzeichnisse abgerufen werden. Der Unterschied zum LAN besteht hauptsächlich in der Darstellung von Hypertexten. Der Anwender benötigt zur Ansicht einen Browser.

Das **Extranet** ist zumeist ein Teil des Intranets von Unternehmen, dessen Inhalt über das Internet einer bestimmten Nutzergruppe zur Verfügung gestellt wird. Solche Nutzergruppen können Mitarbeiter im Außendienst, Kunden oder Lieferanten sein. Der Schutz vor unberechtigter Nutzung kann über Passwortabfragen oder über die Freischaltung bestimmter IP-Address-Bereiche erfolgen.

#### 4.2.1.6 Kommunikationsmodelle

Kommunikationsmodelle beschreiben die Hierarchie und Aufgabenverteilung der beteiligten Kommunikationspartner im Netz. Hier soll in Modelle aus Tabelle 4.3 unterschieden werden.

Modell	Kommunikationspartner	Verbindung
Client-Server	Server stellt Informationen bereit und Client fordert diese Infos an	besteht nur für Datenaustausch
Peer-to-Peer	gleichberechtigte Partner, die Client- und Server-Funktionen übernehmen	nur für Datenaustausch
Producer-Consumer	Producer generiert Daten, die vom Consumer angefordert werden	immer wenn der Producer Daten bereitstellt und der Consumer sie aufnehmen kann
Publisher-Subscriber	Publisher stellt kontinuierlich Informationen bereit, die vom Subscriber verarbeitet werden	zyklisch durch Timer oder durch externe Anforderung; Vermittlung erfolgt teilweise über Agenten

*Tabelle 4.3: Kommunikationsmodelle*

#### 4.2.1.7 Übertragungsarten

Die Übertragungsart bestimmt die Anzahl der Empfänger. Eine Unicast-Übertragung entspricht einer Point-to-Point-Verbindung, bei der Nachrichten nur zwischen den zwei Kommunikationspartnern übertragen werden.

Das Senden von Nachrichten an alle Teilnehmer eines Netzwerkes entspricht einem Broadcast. Hierbei wird die Nachricht an die letzte Subnetzadresse adressiert.

Ein Multicast sendet die Nachrichten an eine festgelegte Gruppe von Teilnehmern.

#### 4.2.1.8 Kommunikationsgeräte

Die Übermittlung von Datenpaketen in oder zwischen Netzen erfordert Geräte, die

- die Signale verstärken
- Netze unterschiedlicher Standards verbinden
- die Wegeplanung der Datenpakete übernehmen.

Die folgenden Geräte realisieren jeweils Teile dieser Funktionen.

Repeater verstärken die Signale, um die Reichweite von Netzen zu erhöhen.

Bridges leiten die Daten von einem Subnetz in ein anderes Subnetz weiter. Sie sind schneller als Router, dafür aber nicht so vielseitig einsetzbar. Sie bereiten die Daten bis zur 2. Schicht auf.

Router leiten Datenpakete in Richtung Empfänger weiter. Sie übernehmen die Planung der Wegwahl. Dabei durchlaufen die Daten die ersten drei Schichten des OSI-Referenzmodells.

Ein Gateway dient als Verbindung zwischen zwei grundsätzlich verschiedenen Rechnernetzen. Es werden alle Schichten des Referenzmodells durchlaufen.

Ein Proxy ist ein Rechner, der das LAN mit dem Internet verbindet. Er speichert aus dem Internet angeforderte Daten. Werden diese Seiten erneut durch einen Nutzer angefordert, so können diese gespeicherten Seiten dem Anwender zur Verfügung gestellt werden. Das hat den Vorteil, dass die Verbindung ins Internet, die oft eine stark begrenzte Bandbreite und hohe Kosten aufweist, nicht in Anspruch genommen werden muss. Zudem kann ein Proxy-Cache die Daten schneller liefern als der Webserver. Nachteile ergeben sich allerdings, wenn die gespeicherten Seiten im Internet geändert wurden. Der Proxy muss das erkennen und dem Anwender die neue Seite laden. Das ist nicht immer sofort der Fall und muss bei der Programmierung beachtet werden.

Hubs verteilen Daten in Sternstrukturen entweder passiv (nur weiterleiten) oder aktiv (mit Verstärkung).

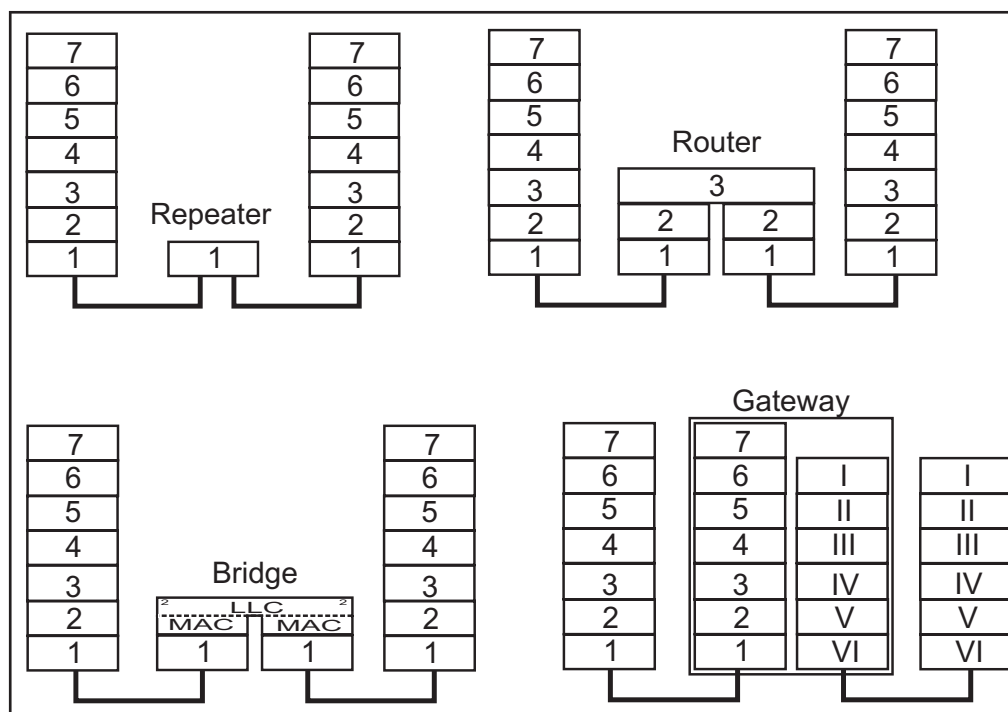


Abbildung 4.3: Geräte der Netzwerktechnik im OSI-Referenzmodell

Switches sind eine Weiterentwicklung von Hubs. Sie vermitteln jedoch gezielt Datenpakete in den Sternzweig, in denen sich der Empfänger befindet. Switches müssen nicht konfiguriert werden, sie lernen selbstständig. Im Vergleich zu Hubs realisieren sie höhere Datenraten.

Zu den Kommunikationsgeräten zählen auch Geräte, die Informationen im Netz bereitstellen. Bekannt sind hier vor allem Webserver. Sie stellen multimediale Informationen in Form von Internetseiten zur Verfügung.

Ein Webserver ist ein Rechner mit einer speziellen Software und einer Anbindung an ein Netz. Dieses Netz kann ein Intra-, Extra- oder das Internet sein.

Webserver arbeiten nach dem Client-Server-Prinzip. Sie hören den Port 80 ständig auf Anforderungen von Dokumenten ab. Solche Dokumente sind im Allgemeinen HTML-Seiten, Java-Applets, CGI-Dateien oder Bild-Dateien. Gegenstationen zum Server sind Internet-Clients, die mit einer Browser-Software ausgestattet sind. Browser stehen von verschiedenen Herstellern zur Verfügung. Einige bekannte sind der Internet Explorer, Netscape Mozilla und Opera. Sie unterstützen alle den HTML-Standard, jedoch in unterschiedlichen Ausprägungen. Verschiedene Medien-Dateien können über Plug-Ins dargestellt werden. Die Anforderung der Daten erfolgt über die URL.

Für dynamische Inhalte bieten viele Webserver eine CGI-Schnittstelle (*Common Gateway Interface*) an. Darüber können Scripte und Programme aus dem HTML-Code heraus mittels der Methode GET aufgerufen werden und eigenständig den HTML-Code erzeugen. Um die Scripte auszuführen, muss auf dem Server ein Interpreter vorhanden sein, der das Script übersetzt und ausführt. Für jede Anfrage wird eine neue Programm-Instanz gestartet, die das Script interpretiert und ausführt. Dies verlangt dem Server viel Rechenleistung ab und wird als Nachteil von CGI betrachtet. Eine häufig eingesetzte



Scriptsprache ist Perl. Alternativen stellen die Sprachen PHP, ASP, JSP/Servlet, FastCGI und SSI sowie Servlets dar. Auf Programmiersprachen wird im Anschluss näher eingegangen.

Für die Übermittlung von Daten des Clients an den Server stehen die Methoden POST und PUT zur Verfügung.

Die in diesem Abschnitt behandelten Protokolle, Geräte und Algorithmen stellen keine vollständige Abhandlung des Themengebietes "Internet" dar. Die Inhalte beziehen sich auf die später auszuwählenden Algorithmen in der Implementierungsphase.

## 4.2.2 Programmiersprachen für Web-Applikationen

Programmiersprachen für Internet-Dienste sind nach folgenden Kriterien auszuwählen:

**IT-Sicherheit** – Die Programme dürfen keine Daten an Dritte preisgeben. Realisiert wird dies bei der Übertragung mittels Verschlüsselungsverfahren. Eine weitere Forderung ist, keinem der Kommunikationspartner zu schaden [36].

**Netzwerkfähigkeit** – Für die Übermittlung von Daten und den Zugriff auf verteilte Ressourcen werden Mechanismen zum Auf- und Abbauen von Netzwerkverbindungen benötigt.

**Ort der Ausführung** – Die Möglichkeit der Ausführung dieser Programme besteht auf der Server- und der Client-Seite. Beide Varianten haben ihre Vor- und Nachteile, die je nach Einsatzfall abgewogen werden müssen.

**Dynamik der Oberfläche** – Hier steht die Frage der dynamischen Inhalte der Oberfläche im Vordergrund. Wird die Seite erst beim Neuladen aktualisiert, oder sofort bei Eintreten eines Ereignisses. Im zweiten Fall schließen sich die meisten serverseitigen Sprachen aus [36].

Die folgende Vorstellung einiger ausgewählter Sprachen soll eine Hilfestellung für die Auswahl in der Implementierung geben.

### 4.2.2.1 HTML und XML

HTML ist eine Metasprache, abgeleitet von SGML. Sie dient der Auszeichnung von Hypertexten und besteht aus Strukturanweisungen, die es erlauben, Texte strukturiert in einem Webbrowser darzustellen. Im weitesten Sinne bereitet sie damit den Weg der Visualisierung von Texten im Browser [49, 69].

XML ist eine Auszeichnungssprache speziell für Daten. XML hat sich bereits beim Austausch von Daten zwischen unterschiedlichen Programmen bewährt. Die Daten werden losgelöst von der Art der Darstellung in Strukturen abgelegt. Darüber hinaus existieren eine Vielzahl von Mischformen wie zum Beispiel XHTML.



#### 4.2.2.2 Perl

Perl ist eine Scriptsprache, die die CGI-Schnittstelle unterstützt. Sie eignet sich besonders gut zum Verarbeiten von Texten. Mit Zusatzmodulen sind Aufgaben unterschiedlichster Art lösbar, so zum Beispiel die Datenbankprogrammierung. Auch in der Systemadministration wird Perl gern angewendet [22, 79].

Nachteilig ist die generell langsame Verarbeitung von Scriptsprachen. Die Ausführung auf dem Webserver erfordert sehr performante Rechentechnik.

#### 4.2.2.3 PHP

PHP ist eine leicht erlernbare serverseitige Scriptsprache. Die Syntax orientiert sich an Perl und C. Vor allem Funktionen zur Datenbankanbindung und Protokollauswahl sind für die Programmierung dynamischer Webseiten hilfreich [79].

#### 4.2.2.4 Java

Die Sprache Java vertritt ein weiteres Konzept. Java ist

- objektorientiert
- plattformunabhängig
- modular.

Das Java-Programm kann über die Laufzeitumgebung, der JVM (*Java Virtual Machine*), in Webbrowsern integriert und ausgeführt werden. Das heißt, Java-Programme belasten nicht die Rechenzeit der Webserver, sondern werden auf dem Client ausgeführt. Für Webbrowser geschriebene Programme nennen sich Applets. Sie beinhalten Sicherheitsrestriktionen, die den Internet-Client vor Schaden schützen sollen. So können sie Internetverbindungen nur zu dem Server aufbauen, von dem sie geladen werden. Zugriffe auf das lokale Dateisystem sind nicht möglich [11, 12, 53, 79].

Schlagworte, die für Java sprechen sind "[...] das dynamische Binden (Internet-Bezug) sowie Sicherheitsaspekte, Ausnahmebehandlung, automatische Speicherverwaltung (Garbage Collection), mehrfädige Ausführung und vor allem große Unterstützung im Bereich der verteilten Programmierung [...]" [14] (S. 87) sowie die Objektorientiertheit. Eine große Menge an vorgefertigten Bibliotheken ist bereits im Standardumfang enthalten. Auf dem lokalen Rechner wird die JVM benötigt, die den Byte-Code ausführt. Die JVM wird von unterschiedlichen Herstellern in unterschiedlichen Ausführungen angeboten. Die bekannteste ist die von Sun Microsystems. Sie ist kostenlos, aber nicht Bestandteil von Windows. Microsoft liefert zum Betriebssystem eine eigene JVM aus, die aber zur JVM von Sun Microsystems nicht in allen Funktionen kompatibel ist. Der Support von Microsoft ist Anfang 2005 eingestellt worden<sup>4</sup>.

---

<sup>4</sup> siehe [www.heise.de](http://www.heise.de), Artikel vom 07.10.2003: Microsoft verlängert Unterstützung von Java

#### 4.2.2.5 SOAP, UDDI, WSDL – Web-Service

SOAP (*Simple Object Access Protocol*) ist ein Protokoll zur Kommunikation von Web-Services. Eine Definition durch die Gesellschaft für Informatik ist in [17] zu finden:

"Web-Services sind selbstbeschreibende, gekapselte Software-Komponenten, die eine Schnittstelle anbieten, über die ihre Funktionen entfernt aufgerufen und die lose durch den Austausch von Nachrichten miteinander gekoppelt werden können. Zur Erreichung universeller Interoperabilität werden für die Kommunikation die herkömmlichen Kanäle des Internets verwendet. Web-Services basieren auf den drei Standards WSDL, SOAP und UDDI: Mit WSDL wird die Schnittstelle eines Web-Services spezifiziert, via SOAP werden Prozedurfernaufrufe übermittelt und mit UDDI, einem zentralen Verzeichnisdienst für angebotene Web-Services, können andere Web-Services aufgefunden werden."<sup>5</sup>

Das World Wide Web Consortium und Intel vertreten ähnliche Definitionen in [21] und [1].

Nach [38] ist, wie die Abbildung 4.4 zeigt, ein Web-Service hingegen nur die Schnittstelle zwischen einem Anwendungsprogramm und dem Anwender.

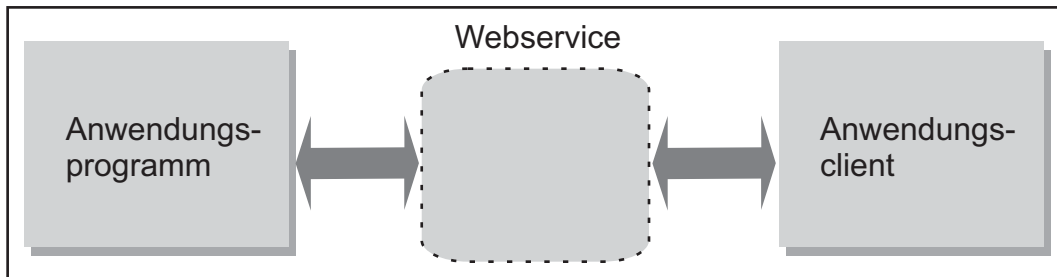


Abbildung 4.4: Web-Service [38]

Diese Arbeit folgt der Definition der Gesellschaft für Informatik. Ein Web-Service ist ein Programm/eine Funktion mit einer Schnittstelle, die über Computernetze und internet-typische Protokolle (HTTP, TCP/IP) angesprochen werden kann, ähnlich RPC (*Remote Procedure Call*) [60]. Der Gegenpart zu diesen Funktionen sind Programme, die diese Web-Services nutzen. Die Schnittstelle wird in WSDL (*Web-Services Description Language*) über XML beschrieben. Der Anwendungsclient findet mittels des Verzeichnisdienstes UDDI (*Universal Service Description, Discovery and Integration*), ähnlich den Gelben Seiten, für ein bestehendes, definiertes Problem einen Service, der eine Lösungsstrategie bereithält. Über die WSDL-Beschreibung und die gelieferte URL können die Funktionen mit den benötigten Parametern angesprochen und genutzt werden.

Die Web-Services haben bereits auch in der Automatisierungstechnik Einzug gehalten. Die OPC berichtet in [26], dass der "[...] Wechsel von der COM/DCOM zur Web-Service-basierten OPC-Technologie[...]" vollzogen wurde. Mit SOAP ist "[...] endlich der Sprung zu einer wirklich offenen Architektur gelungen [...]" [4].

<sup>5</sup> Die Schreibweise für *Web-Service* wird in der Literatur uneinheitlich vorgefunden.

## 4.3 E-Service – Dienstleistungen für die Produktion

Mit dem Siegeszug des Internets und den nach und nach sichtbar werdenden Vorteilen, die vor allem in der weltumspannenden Ausdehnung und dem nahezu unbeschränkten Informationspotenzial liegen, wurde das Internet in Aufgaben der Automatisierungstechnik mit einbezogen [37, 57]. Die Globalisierung der Märkte und steigende Serviceanforderungen der Kunden an die Hersteller von Maschinen lassen enorme Kosten auf die Dienstleister zukommen. Strategien für automatische Dienstleistungen aus der Ferne sind zur Zeit nicht in dem erforderlichen Maß oder gar nicht verfügbar. Eine erste Entwicklung einer Dienstleistung auf der Basis des Internets ist der Teleservice.

### 4.3.1 Teleservice

Teleservice gibt dem Anwender Hilfestellung in Ausnahmesituationen zur Fehlereingrenzung und Fehlerbehebung. Über eine Internetleitung kann sich der Maschinenhersteller auf die Steuerung der Maschine aufschalten, Daten auslesen und sich sogar über installierte Kameras ein Bild vom optischen Zustand der Maschine verschaffen. Daraus ergeben sich Reparaturanweisungen, Ersatzteillisten und wenn nötig ein Auftrag an einen Servicetechniker.

Das folgende Beispiel der Firma Zeiss beschreibt den Ablauf bei der Nutzung von Teleservice recht allgemeingültig in Abbildung 4.5.

1. Maschinenbediener meldet über Telefon den Störfall an die Zeiss-Hotline.
2. Weiterleiten an Serviceteam im Support-Center.
3. Einwählen der Maschine über LAN oder Modem zum Support-Center und Freischalten der Maschine für Serviceteam durch Bediener.
4. Untersuchen der Maschine und Beheben des Fehlers, falls nicht möglich, Speichern der ermittelten Daten und Übergabe an lokalen Servicetechniker.

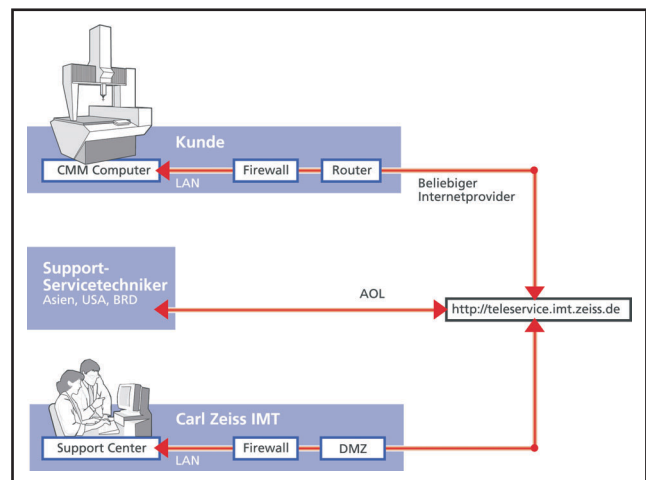


Abbildung 4.5: Teleservice der Firma Zeiss [73]

Die Einbeziehung des Menschen auf beiden Seiten und die Inanspruchnahme des Dienstes in Ausnahmesituationen ist typisch für Teleservice-Anwendungen. Dies hat jedoch zur Folge, dass Ausfallzeiten durch Fehlerbehebung, Reparatur und Ersatzteilbeschaffung entstehen.

### 4.3.2 Neue Ansätze

Neue Ansätze von Servicedienstleistungen zielen auf eine Reduzierung der ungeplanten Ausfallzeiten und auf eine Optimierung von Produktionsabläufen auf strategischer als auch auf technischer Seite. So sind Dienste in den Bereichen:

- kontinuierliche Überwachung,
- Optimierung
- Programmierung
- Konstruktion
- Inbetriebnahme

denkbar und teilweise realisiert. Aber auch der Nutzen multimedialer Internettechnologien darf nicht aus dem Blickfeld geraten. Die ständige Verfügbarkeit und Aktualität von Informationen jeglicher Art bringt einen ungeheuren Nutzen. Den "Lebenslauf" von Maschinen mit allen durchgeführten Umbauten, Wartungen und Bauplänen dem Servicepersonal an der Maschine zur Verfügung stellen zu können, ist von großem Wert, da es die Stillstandszeiten minimieren kann [9].

Anbieter dieser Services können, wie im Fall von Zeiss, Maschinenhersteller sein, die ihren Kundenservice über den gesamten Produktlebenszyklus garantieren. Für Maschinenhersteller wird das immer mehr zu einem wichtigen Wettbewerbsvorteil auf dem Markt [9]. Damit werden neue Geschäftsfelder in Zeiten der drohenden Deindustrialisierung eröffnet. Ein weiterer Nutzen der Dienstleister aus E-Services ist die Vervielfältigung ihres Expertenwissens mittels Implementation des Know-hows in Algorithmen. Vollständig automatisierte E-Services können selbstständig, ohne Zutun des Menschen auf der Dienstleister-Seite, 24 Stunden am Tag ablaufen.

Mit der Frage der Zahlungsbereitschaft der Nutzer für E-Services hat sich der Verein Deutscher Ingenieure in einer Studie befasst. In Tabelle 4.3.2<sup>6</sup> ist ein Auszug aus den Ergebnissen dargestellt.

E-Services bauen auf die in Kapitel 4.2.2.5 vorgestellten Netzwerk-Grundlagen auf. In dieser Arbeit sind E-Services für die Produktion Gegenstand der Untersuchungen. Das Spektrum an vorstellbaren Diensten reicht von der Aktor-Sensor-Ebene über die Steuerungsebene, der Produktionsplanungsebene bis ins Management.

---

<sup>6</sup> Originalveröffentlichung in [3] und [10]

Katalog möglicher E-Service-Leistungen	Kunde: Zahlungsbereitschaft	Hersteller: Kostspflicht
Internetbasierte Fehlerbehebung	45 %	71 %
Internetbasierte Instandhaltung	34 %	67 %
Internetbasierte Fehlerdiagnose	31 %	61 %
Internetbasierte Störungsmeldung	31 %	58 %
Internetbasiertes Software-Update	24 %	52 %
Internetbasierte Prozessführung/Teleanufacturing	21 %	76 %
Internetbasierte Maschinen- und Prozessüberwachung	21 %	72 %
Internetbasierte Inbetriebnahme	15 %	70 %
Internetbasierte Engineering/Entwicklung	13 %	46 %
Elektronischer Zugriff auf Störfalldatenbank	13 %	29 %
Internetbasierte Einsatzanforderung	11 %	33 %
Internetbasiertes Projektmanagement	10 %	56 %

*Tabelle 4.4: Auszug aus [37] zur Zahlungsbereitschaft für E-Services*

### 4.3.3 Das E-Industrial-Services-Portal

Die Fraunhofer-Gesellschaft hat in den letzten Jahren ein Internet-Portal – E-Industrial Services – für Mehrwertdienste an Maschinen aufgebaut. Als Zielgruppe dieser Dienste werden Maschinenhersteller und deren Kunden gleichermaßen genannt. Den Herstellern von Maschinen wird ein System für die Implementation eigener Dienste zur Verfügung gestellt. Diese können durch die Kunden gebucht werden. Der Mehrwert für den Maschinenhersteller ergibt sich, wenn er diese Dienstdaten des Kunden auswertet und darin gezielt nach Schwachstellen an seinen Maschinen sucht.

Das Portal<sup>7</sup> bildet die technische Grundlage für die Erbringung von Dienstleistungen. Verglichen mit dem herkömmlichen Dienst "Auto-Reparatur" entspricht es der Werkstatt, in der Reparaturen durchgeführt werden. Es ist das Fundament zukünftiger Dienstleistungen und das Tor der Dienste ins Internet. Fraunhofer hat es als offene Umgebung gestaltet. Es integriert:

- Werkzeuge
- Anwendungen
- Dienstleistungen.

Wie in Abbildung 4.6 zu erkennen, besitzt das Portal modularen Charakter. Es greift auf bereits vorhandene Techniken zurück, um den Entwicklungsaufwand so gering wie möglich zu halten. Besonderes Augenmerk wird auf breite Schnittstellen zwischen den Modulen gelegt, um eine Anpassung an Kundensysteme leichter zu ermöglichen.

Die betrachteten Sicherheitsrisiken in E-Industrial-Services werden in [15] benannt mit:

<sup>7</sup> Ein Portal ist eine Sammlung von Internetseiten zu einem Thema.

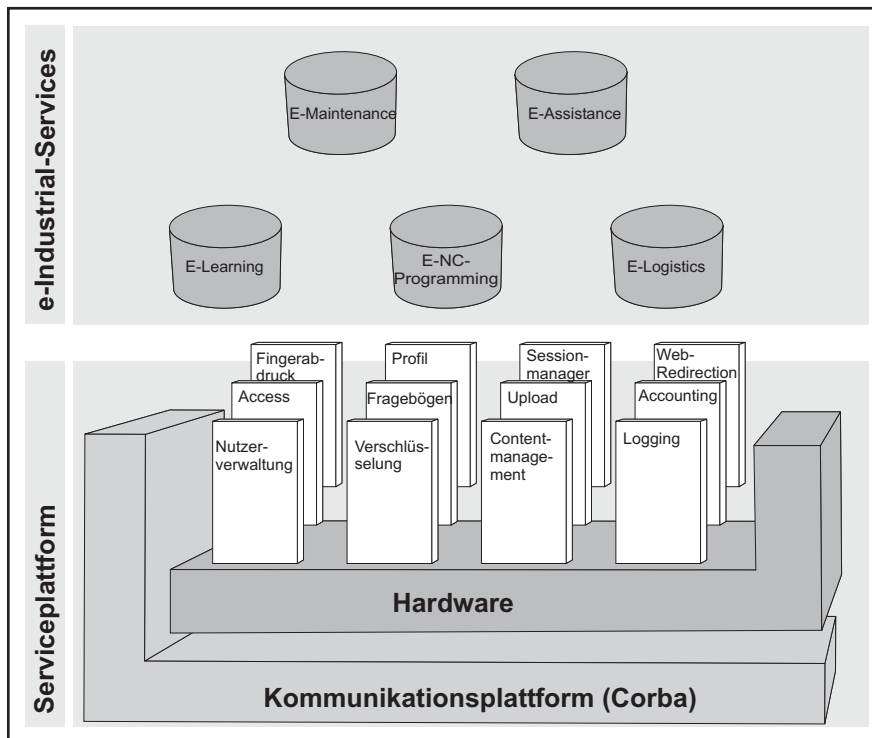


Abbildung 4.6: Portal für E-Industrial-Services nach Fraunhofer [15]

- Gefahr für Mensch und Maschine
- Spionage von Kundendaten und Rezepturen
- Missbrauch der Service-Resultate.

Gefahr droht vor allem von direkten Konkurrenten, Mitarbeitern des Unternehmens und Hackern.

Als Lösungsansätze werden Rollendefinitionen, Rechtevergabe sowie eine Begrenzung der erlaubten Zugriffsbereiche genannt. Die Übertragung der Daten wird durch Verschlüsselungsverfahren geschützt. Um unerwünschte Zugriffe auf sensible Bereiche zu unterbinden, wird ein Service-Agent eingesetzt. Dieser abgerüstete PC befindet sich im Kundennetzwerk und baut nur Verbindungen nach außen auf. Verbindungsanforderungen von außen lehnt er ab. Der Kunde hat jederzeit die Möglichkeit, seine Steuerung oder Datenbank vom Dienst zu trennen.

In [15] wird sehr deutlich, dass das Thema (Daten-)Sicherheit auf Anwender- und Dienstleistungsebene nicht zu unterschätzen ist, ein gesundes Verhältnis von Aufwand und Nutzen aber schnell überschritten wird.

#### 4.3.4 E-Industrial-Services

Die Fraunhofer-Gesellschaft entwickelt in einem Forschungsverbund eigene E-Industrial-Services, um Erfahrungen zu sammeln und das Angebotsspektrum an Dienstleistungen für die Industrie zu erhöhen. Im Folgenden werden zwei ausgewählte Services näher beschrieben [74].

#### 4.3.4.1 E-Simulation

Der Service E-Simulation gliedert sich in den Bereich der Optimierung von Produktionsprozessen ein. Mittels Simulationen können Abläufe in der Produktion grafisch dargestellt werden. Dies ermöglicht die Optimierung von Ressourcen vor oder parallel zum Produktionsprozess.

Anwendung in der Wirtschaft findet die Simulation vor allem bei der Planung oder Umstrukturierung von Produktionsabläufen. Der Vorteil einer Simulation über das Internet ist die globale Erreichbarkeit und ständige Aktualität des Modells. Für den Nutzer ergeben sich weitere Vorteile in der Kosteneinsparung durch Lizenzen von Simulationssystemen, der Unterstützung beim Aufbau der Simulation durch Experten und das Nutzen fremder Rechenleistung.

Der Service E-Simulation der FhG beginnt mit dem Aufbau des Modells durch Simulations-Experten. Ist das Modell erstellt, können die Auftraggeber Prozess- und Steuerparameter an den Simulator senden. Nach einer Zeit<sup>8</sup> stehen dem Anwender Ergebnisse in Form von Tabellen zur Verfügung, aus denen sich Schwachstellen im Prozess ableiten lassen. Inwieweit sich daraus Optimierungsvorschläge ergeben, wird nicht erörtert.

#### 4.3.4.2 E-Online-Diagnose

Für Maschinen mit hohen Anforderungen an die Verfügbarkeit bietet sich eine ständige Überwachung von aussagekräftigen Parametern wie Schwingungen, Temperaturen oder Gütekriterien des Reglers (Verlauf des Schleppabstandes) an. Mit der Beobachtung dieser Werte lassen sich Fehler an Bauteilen bereits vor dem Bruch erkennen. Anzeichen dafür ist das Überschreiten von Grenzen, zum Beispiel der Temperatur eines Fräasers.

Ein Vorteil für den Anwender ist die steigende Verfügbarkeit seiner Maschinen sowie eine bessere Qualität seiner Produkte. Der Maschinenhersteller kann aufgrund der ausgelesenen Daten den Defekt online feststellen und vor dem Einsatz eines Technikers die Ersatzteile beschaffen. Zumeist sei ein Einsatz eines Technikers jedoch gar nicht mehr nötig, so [74]. Weiterhin kann er über die Analyse der aufgenommenen Daten ein Anforderungsprofil des Maschinenbetreibers an die Maschine erstellen und seine zukünftigen Maschinen den Marktbedingungen anpassen. Hieraus ergibt sich ein Mehrzweck für beide Seiten.

Weitere von der Fraunhofer-Gesellschaft entwickelte Services sind:

- bedarfsgerechte Personalqualifizierung
- E-Logistic
- elektronisches Serviceheft für Maschinen
- E-NC Programming.

---

<sup>8</sup> ca. 30 Minuten, nach [74]



### 4.3.5 ePS Network Services

Die ePS Network Services von Siemens bieten ebenfalls elektronische Dienstleistungen an. Sie stellen eine industrielle Lösung für Steuerungstechnik dar. Aufgebaut ist ein Portal mit den Diensten:

- eP-Dynamic
- eP-Performance
- eP-Access

sowie Grundfunktionen:

- Sicherheit bei der Datenübertragung mittels SSL-Verschlüsselung
- Benutzer- und Maschinenverwaltung
- Installation und Speicherung der Dienste auf dem ePS-Server
- Speichern und Archivieren von Daten der angeschlossenen Maschinen.

Grundlage ist der ePS Network Server. Dieser dient als Bindeglied zwischen Maschinensteuerungen, Maschinenherstellern und Dienstleistern. Der Zugriff auf die Steuerung durch den Hersteller erfolgt ausschließlich über den ePS Network Server analog zu Abbildung 4.7<sup>9</sup>.

Die angebotenen Dienste beziehen sich hauptsächlich auf Services für PC-based NC Steuerungen von Siemens (SINUMERIK 810D/840Di/840D mit PCU 50) mit den Zielen:

- Performance-Steigerung
- optimierte Störfalldiagnose
- Fernzugriff auf die Steuerung.

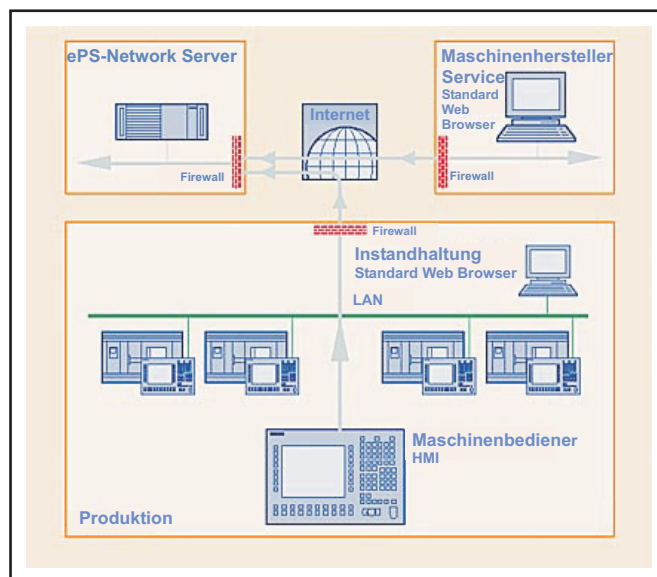


Abbildung 4.7: ePS Network Services – System [2]

Wucherer stellt in [78] die folgenden möglichen Services kurz vor:

**Machine Services** – unterstützen die Inbetriebnahme und bieten Störungsanalysen an. Sie enthalten Funktionen wie Fernbedienung, Audio Conferencing, Chat Mode oder File Transfer.

<sup>9</sup> Quelle: [2] -> Dienstleistungen -> ePS Network Services



**Machine Performance** – erkennen von Verschleißerscheinungen durch periodische Tests.

**Workpiece Related Services** – Echtzeiterfassung von Prozessdaten zur frühzeitigen Erkennung von Schwachstellen an der Maschine.

**Data Management** – Web-basierter Dienst zum Verwalten und Archivieren von Steuerungsdaten wie Parametern, Maschinendaten und NC-Programmen.

**E-Shop** – Online-Ersatzteilbeschaffung.

ePS Network Services sind E-Services für ein auf Siemens Simatic spezialisiertes Portal. Eine allgemeine Lösung für alle webfähigen Steuerungen scheitert unter anderem an der fehlenden genormten Schnittstelle für das Abfragen von Variablen und den Zugriff auf Steuerungsfunktionen. Die E-Industrial-Services behaupten, mit ihren "breiten" Schnittstellen für alle Systeme/Steuerungen geeignet zu sein.

Es ist auffällig, dass die vorgestellten Dienstleistungen eher einen passiven Charakter haben. Ein Bewegen von Maschinenteilen durch einen E-Service ist bisher nicht bekannt. Gründe dafür können fehlende Sicherheitsrichtlinien für diese Art des Zugriffs sein.

## 4.4 Sicherheitsbetrachtungen

Sicherheitsanforderungen für aktive Fernzugriffe liegen vor allem in der Datensicherheit und in der Sicherheit von Mensch und Maschine, wobei beide voneinander abhängen. Wickert gibt in [75] Anregungen zur Maschinensicherheit während des Teleservice. Für ihn ist "die Steuerung der Maschine ohne Sichtverbindung" und der unerwünschte Verbindungsaufbau zur Steuerung problematisch. Dieser kann durch Verwechslung oder durch Sabotage zustande kommen. Zur Vermeidung derartiger Probleme wird die Trennung der Internetverbindung, wenn gerade kein Teleservice erwünscht ist, vorgeschlagen. Bei modernen mechatronischen Steuerungen in einem Netz wird empfohlen, diese aus dem eigentlichen Netz herauszulösen oder den Teleservice generell nur über extra Verbindungen zuzulassen. Weiterhin ist eine Verbindungsaufnahme von außen als problematisch anzusehen. Erfolgt der Verbindungsaufbau von der Steuerung aus, werden Fremdzugriffe weitestgehend verhindert.

Gegenwärtig ist abzusehen, dass im IT-Sicherheitssektor eine stetige Weiterentwicklung voranschreiten wird.

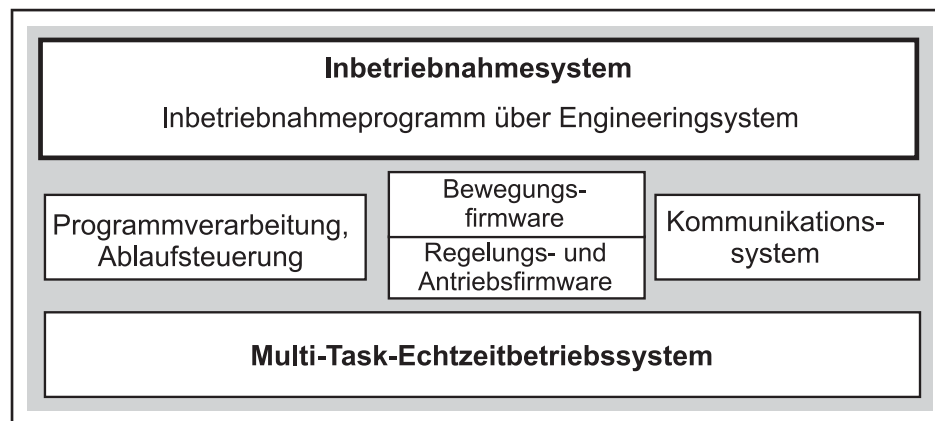
Wickert weist weiter darauf hin, dass der Teleservice nicht an Betriebszustände gebunden ist. Das heißt, er kann auch im Einrichtebetrieb ausgeführt werden. Hier werden die Schutzeinrichtungen der Maschine meist nicht abgefragt. Ist der Maschinenraum offen, kann er dem Personal zugänglich sein. Besteht kein Sichtkontakt des Operators zur Maschine, sind Gefährdungen von Personen nicht auszuschließen. In jüngster Zeit betrifft das nicht mehr nur den Teleservice. Engineeringsysteme können mittlerweile über Ethernet mit der Steuerung verbunden werden. Prinzipiell kann auch hier ein Verfahren von Maschinenteilen ohne Sichtkontakt erfolgen, ganz unbeachtet von einem versehentlichen

Verbinden auf einen benachbarten Controller durch eine falsche IP-Adresse. Eine Absicherung über maschinenindividuelle Passwörter scheint vielversprechend. Der Zugriff auf die Maschine soll dem Personal visuell angezeigt werden. Meldeleuchten oder farbig blinkende Bedienpanels sind ebenfalls eine Möglichkeit. Letztendlich wird von Wickert ein eigener Betriebszustand Teleservice gefordert. Außer der EG-Maschinenrichtlinie sind derzeit keine offiziellen Dokumente in Deutschland verfügbar, die speziell den aktiven Fernzugriff (mit Eingriff in den Programmablauf) auf Funktionsgruppen der Maschine regeln.

# Kapitel 5

## Entwurf und Implementierung

Die Implementierung einer prototypischen Lösung erfolgt auf einer industriellen Bewegungssteuerung mit einem Multi-Task-Echtzeitbetriebssystem. Für die Untersuchungen wird das System SIMOTION von Siemens gewählt. Eingesetzt wird diese Bewegungssteuerung in der Automatisierung von Produktionsmaschinen. Die Architektur der implementierten Software ist in Abbildung 5.1 dargestellt.



*Abbildung 5.1: Softwarearchitektur*

Die Grundlage bildet das Multi-Task-Echtzeitbetriebssystem. Auf diesem läuft die Steuerungsfirmware sowie das Anwenderprogramm auf unterschiedlichen Taskebenen ab. Die Ebenen unterscheiden sich in der Häufigkeit und Determiniertheit ihrer Abarbeitung. Die Steuerung des Ablaufes der Tasks übernimmt das Ablaufsystem mit Ablaufsteuerung. Der Anwender kann unter zyklischen Tasks, ereignisgesteuerten Tasks und Tasks der Round-Robin-Ebene unterscheiden. Aus der Programmverarbeitung werden zur Laufzeit die Technologiefunktionen an die Technologieobjekte weitergereicht. Beim Befehl "Positionieren einer Achse" beispielsweise generiert die Bewegungsplanung eine Bahn. Die Bahn-Sollwerte werden an den Regler übergeben, der die Stellgrößen berechnet. Sie werden über das Kommunikationssystem und ein Medium an den Antrieb gesendet. Dieses Medium muss Echtzeitbedingungen erfüllen. Profibus hat sich hier etabliert. Das reine Ethernet dagegen besitzt diese Echtzeiteigenschaften nicht. Es wird daher eher für die Verbindung zum Engineeringsystem und zu

B&B-Panels genutzt. Die Erweiterung Profinet genügt dem Einsatz im Sensor-Aktorbereich, ist aber noch nicht durchgängig verfügbar.

Bisherige Ansätze zur automatischen Reglerinbetriebnahme legen die Algorithmen in zum Antrieb oder zur Steuerung gehörenden Engineeringsystemen ab. Sie stehen dem Anwender über die gesamte Lizenzlaufzeit zur Verfügung. Es hat sich etabliert, diese Tools unentgeltlich bereitzustellen, was dem mitunter hohen Programmieraufwand und dem enthaltenen Know-how widerspricht.

Im Rahmen dieser Arbeit wird ein neuer Ansatz eines Reglerinbetriebnahme-Tools für digitale Antriebe an Bewegungssteuerungen in prototypischer Ausfertigung beschrieben und erstellt. Die typische Funktionalität bestehender Tools wird beibehalten und die Grundlage zur Beseitigung existierender Schwachpunkte gelegt.

### **Lösungsansatz 1:**

Die herkömmliche Lösung der Programmierung eines Reglerinbetriebnahme-Tools ist die Einbindung in das gewählte Engineeringsystem. Durch dessen modularen Aufbau kann das Tool in einer Hochsprache implementiert werden. Der sich daraus ergebende Vorteil ist, dass bereits vorhandene Funktionen/Schnittstellen zur Kommunikation, Programmierung, Aufzeichnung (Trace) und zur Ablaufsteuerung genutzt und nicht neu erstellt werden müssen.

### **Lösungsansatz 2:**

Eine andere Möglichkeit stellt die Implementation als E-Service dar. Die Literaturanalyse zu diesem Thema in Kapitel 4.3 hat bereits erste positive Ergebnisse bei der Umsetzung von Dienstleistungen für die Produktion als E-Service aufzeigen können. Die Besonderheit eines Dienstes zur Reglerinbetriebnahme besteht in dem selbstständigen Bewegen von Maschinenteilen.

Zur Reduzierung des Programmieraufwandes wird nach einer Möglichkeit gesucht, Funktionen des Engineeringsystems zu nutzen. Der Fernzugriff auf die Steuerungen erfordert besondere, noch festzulegende Sicherheitsvorkehrungen.

## **5.1 Konzept der Implementierung**

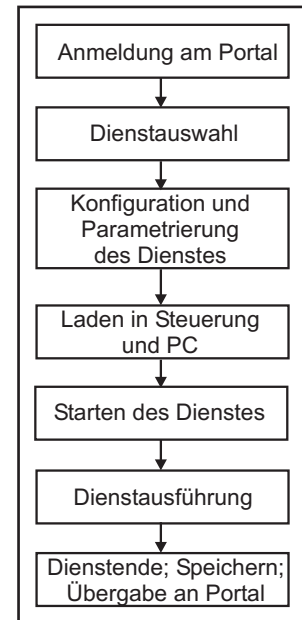
Für die folgende Implementierung wird ein Grobkonzept nach Lösungsansatz 2 erstellt, an dem sich die weiteren Entwicklungen orientieren.

Die Softwarestruktur setzt sich aus drei verteilten Programmen zusammen. Im Internet befindet sich ein Portal zur Koordination der Dienstleistungen und der Kommunikation. Der Dienst selbst

teilt sich in die Online- und Offline-Komponente<sup>1</sup>. Die Online-Komponenten mit Echtzeitfunktionalität (Funktionsgenerator, Trace, Sicherheitsfunktionen) werden auf der Steuerung angesiedelt, die Offline-Funktionen zur Ablaufsteuerung und Berechnung werden im Portal bereitgestellt und auf dem Anwender-PC zur Ausführung gebracht. Die Verbindung der drei Komponenten erfolgt über Kommunikationsbeziehungen.

Der Standardablauf von Diensten wird entsprechend Abbildung 5.2 wie folgt festgelegt:

Der Anwender verbindet sich über einen handelsüblichen Browser mit dem Portal-Prototypen für E-Services der Produktionstechnik. Nach einer Anmeldung über ein noch zu definierendes Verfahren wählt er den von ihm gewünschten Dienst aus einer Dienstdatenbank aus. Die Dienstdatenbank beinhaltet die Adresse des Dienstes im Netz. Sie verweist auf Rechner im Internet/Portal. Bevor der Dienst gestartet werden kann, ist er vom Anwender zu konfigurieren. Hierfür sind im Portal Strategien abgelegt. Nach dem Start des Dienstes auf dem Anwender-PC und der Steuerung wird eine Verbindung zwischen beiden aufgebaut. Diese Verbindung darf aus Sicherheitsgründen nur über das Portal möglich sein [44]. Die Reglerinbetriebnahme wird nach den bereits beschriebenen Schritten ausgeführt. Nach erfolgreichem Abschluss des Dienstes sind dem Anwender die Ergebnisse zu übermitteln und portalseitig zu speichern. Dieser Schritt beendet den Service.



**Abbildung 5.2:**  
**Dienstablauf**

Aus dem dargestellten Szenario können die notwendigen Kommunikationsbeziehungen entsprechend Abbildung 5.3 abgeleitet werden. Der Anwender tritt mit dem Portal und dem Offline-Teil des Dienstes in Interaktion. Im Portal wird der Dienst parametrierung. Über die Offline-Komponente steuert er den Dienstablauf. Außerdem kommt dem Portal eine zentrale Vermittler- und Verwalterrolle zu. Es realisiert die Kommunikation zwischen den genannten Komponenten und verwaltet die Dienste und die Dienstausführung. Der Dienst selbst enthält Informationen und Anweisungen zur Diensterbringung, die Bedienung erfolgt durch den Anwender. Das Target ist dienstespezifisch, es entspricht im Fall der Reglerinbetriebnahme der Steuerung C230-2 und enthält die Online-Komponente.

Während der Ausführung von Diensten werden allgemeine Anforderungen gestellt. Im Vordergrund steht immer die Sicherheit von Mensch und Maschine. Ein wichtiger Aspekt ist ebenfalls der Schutz sensibler Daten vor Dritten. Hierzu zählen nicht nur die Anwenderdaten, sondern auch die bereitgestellten Dienstalgorithmen. Neben diesen grundlegenden Funktionen ist eine generelle Hilfestellung bei der Dienstausführung wünschenswert.

<sup>1</sup> Die Begriffe Online- und Offline-Komponente werden nach ihrer Funktionalität aus der Offline-Reglerinbetriebnahme abgeleitet, vgl. Kapitel 3.

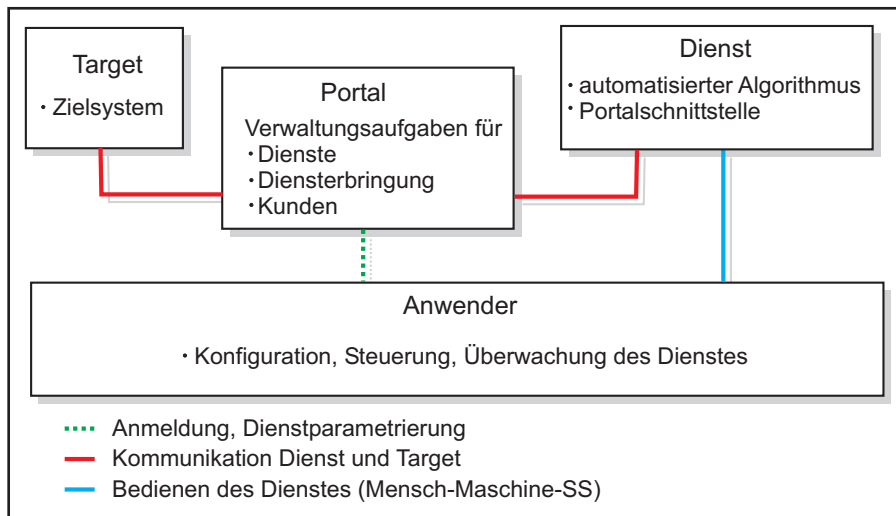


Abbildung 5.3: Kommunikationsbeziehungen im System aus Anwendersicht

## 5.2 Systemstruktur

Die praktische Realisierung geht von der Struktur nach Abbildung 5.4 aus. Es sind die folgenden Komponenten enthaltenen:

- ein PC mit dem Betriebssystem Win2000
- eine Steuerung C230-2 ohne Webserver des Entwicklungsstandes *May 31.03.00*
- ein Steller SIMODRIVE 611 für zwei Achsen
- zwei Motoren.

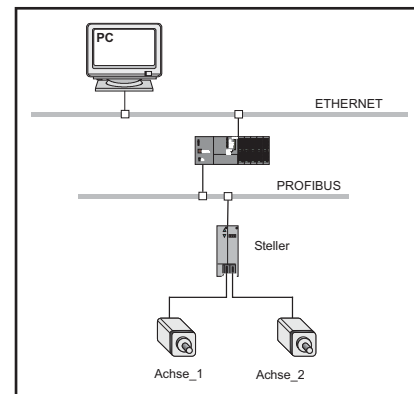


Abbildung 5.4: Systemaufbau

Der PC und die Steuerung sind mit dem LAN des Firmennetzwerkes verbunden. Die Kommunikation von Steuerung und Steller erfolgt über Profibus DP. Alle weiteren Betrachtungen beziehen sich auf diesen Versuchsaufbau.

## 5.3 Der Web-Service im System

### 5.3.1 Portal

Für die Bereitstellung der Dienste wird in einem Netz ein Portal aufgebaut. Das Portal entspricht einer Homepage mit erweitertem Funktionsumfang. Im folgenden Abschnitt werden diese Funktionen und der Aufbau des realisierten Prototyps vorgestellt.

Folgender Funktionsumfang wird vom Portal gefordert:

1. Webserver-Funktionalität

2. Verwaltung von Diensten, Nutzern und Sitzungen
3. Vermittlung der Kommunikation Service–Steuerung
4. Expertenzuschaltung
5. Datensicherheit und Zugriffsbeschränkung
6. Abrechnung der genutzten Dienste.

#### 5.3.1.1 Webserver-Funktionalitäten

Diese beziehen sich auf die in Kapitel 4.2.1.8, S.63 genannten Eigenschaften. Webserver werden in unterschiedlichen Ausprägungen und von verschiedenen Herstellern angeboten. Ein typischer ist der Apache-Webserver. Das Bundesamt für Sicherheit in der Informationstechnik empfiehlt ihn in [47] aus den folgenden Gründen:

- Er ist frei verfügbar unter der Apache Public Licence.
- Er ist mit über 50% der weltweit am häufigsten eingesetzte Webserver.
- Durch einen modularen Aufbau und die daraus resultierende Scalierbarkeit der Funktionsgruppen wird die Serverfunktionalität optimal auf die Bedürfnisse eingestellt.
- Er enthält bereits wichtige sicherheitsrelevante Funktionen.

Um eine Entscheidung zum Einsatz eines Webserver zu treffen, muss auf Anforderungen bezüglich des Portals vorgegriffen werden:

- Unterstützung des Common Gateway Interfaces (Perl)
- Anbindung von Datenbanken
- Erfüllen der hohen Sicherheitsanforderungen hinsichtlich User-Daten
- Betriebssystem Windows XP
- Einrichten von Benutzergruppen und einer Zugriffskontrolle
- Software sollte für den Aufbau des Prototypen kostenlos sein.

Diese Anforderungen werden vom Apache-Webserver unterstützt. Zum Einsatz kommt die Version 2.0.43 (Win32). Andere Alternativen werden nicht untersucht. Im Laufe der Arbeiten wurden keine Schwachstellen erkannt, die einen Wechsel auf eine andere Serversoftware begründen.

### 5.3.1.2 Verwaltungsaufgaben

Für Verwaltungsaufgaben wird eine Datenbank bereitgestellt. Auch hier hat sich im Laufe der vergangenen Jahre eine durchgesetzt, die für Forschungszwecke unentgeltlich zur Verfügung steht – MySQL. Die Anforderungen an die Datenbank werden als nicht außergewöhnlich eingeschätzt. Entscheidungskriterien sind:

- die Kosten
- die Schnittstellen zu Perl
- die Lauffähigkeit unter dem Betriebssystem Windows.

Diesen Anforderungen wird MySQL gerecht.

In der Datenbank werden Informationen über Nutzer und Dienste gespeichert. Zu den Verwaltungsaufgaben des Portals gehört auch die Verwaltung der Sitzung, auch Session genannt. Immer wenn sich der Kunde mit seinem Passwort auf dem Portal anmeldet, bekommt er eine einmalige Session-Identifikationsnummer (SessID)<sup>2</sup> zugewiesen. Diese ID dient der Identifikation des Nutzers bei jeder seiner Aktivitäten. Der Verlauf der Sitzung wird in der Datenbank protokolliert, um die Schritte des Anwenders auch später nachvollziehen zu können. Nützlich ist das vor allem bei der Abrechnung von Leistungen und im Fehlerfall.

### 5.3.1.3 Schnittstellen

Grundlage der Funktionsweise von E-Services ist eine Kommunikation zwischen den verteilten Funktionen auf PC, Portal und Steuerung. Es ergibt sich das in Abbildung 5.5 dargestellte Kommunikationsdreieck Steuerung/Portal/Service.

Die Kommunikationsaufgabe wird in die drei Bereiche:

- Dienst/Portal
- Portal/Steuerung
- Verbindung im Portal

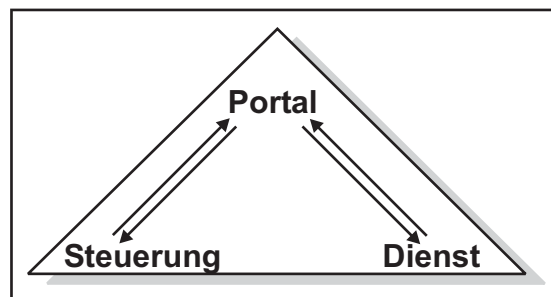


Abbildung 5.5: Kommunikationsdreieck

gegliedert. Eine durchgängige Kommunikation kann nur stattfinden, wenn alle benötigten Schnittstellen und die zugehörigen Protokolle bekannt und umgesetzt sind.

<sup>2</sup>Der Algorithmus zur Berechnung wurde aus [22] entnommen.



**Schnittstelle: Portal/Steuerung**

Charakteristisch ist der Zugriff auf die Steuerung. Die Schnittstellenbeschreibung ist bereits durch die Steuerungssoftware festgelegt. Folgende Funktionen der Kommunikation sind für die Dienstleistung zu erfüllen:

1. Download eines Anwenderprogramms
2. Umschalten der Betriebszustände (RUN und STOP)
3. Lesen von Variablen der Steuerung
4. Schreiben von Variablen auf die Steuerung

Diese Funktionen sind bereits im Engineeringssystem abgelegt. Das Engineeringssystem ist ein über Jahre gewachsenes Softwarepaket. Der "Nachbau" einzelner Teile ist unpraktikabel und mit hohem Zeit- und Kostenaufwand verbunden.

Wucherer nennt in [77] anzustrebende Ziele der Automatisierungstechnik. Eines beschreibt ein Automatisierungssystem mit "[...] Produktkomponenten, die über offene innere Schnittstellen in unterschiedlichen Strukturen zur Anwendung kommen". Die Mehrfachverwendung einzelner Softwarekomponenten wird somit als erstrebenswert betrachtet.

Das Engineeringssystem besteht aus drei Schichten. Die Grundlage bildet Step 7. Darauf setzt das Basis-System des ES auf. Im Kern ist die Funktionalität implementiert. Mittels einer grafischen Benutzeroberfläche (dritte Ebene) werden diese Funktionen vom Anwender bedient.

Das Basis-ES und Step 7 sind über die COM-Technologie realisiert. Die enthaltenen COM-Objekte werden von der Oberfläche "bedient". Perl kann diese COM-Objekte ebenfalls ansprechen. Wird die obere Schicht durch Perl-Scripte ersetzt, kann die Funktionalität des ES genutzt werden.

In der Softwareentwicklung wird ein derartiges Script bereits für automatisierte Funktionstests genutzt und im Folgenden mit Testautomat (TA) bezeichnet.

Auf der Portalseite ist das Engineeringssystem zusammen mit

- Step 7
- Perl
- den Dateien des Testautomaten

installiert. Die beiden Komponenten TA und Webserver verbinden das Portal in Richtung Steuerung und in Richtung Anwender. Abbildung 5.6 zeigt die dabei entstandene zusätzliche Schnittstelle TA/Webserver.

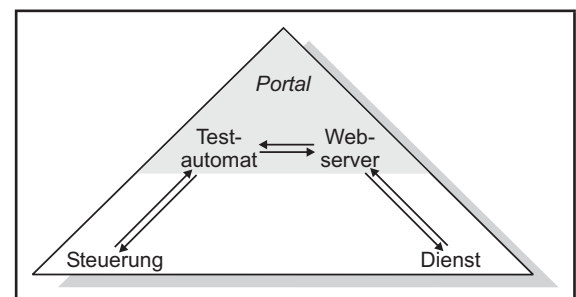


Abbildung 5.6: Erweiterung der SS

Funktion	Parameter	Beschreibung
readVar	VarName	Lesen von Variablen
writeVar	VarName, VarValue	Schreiben von Variablen
readArray	VarName, AnzArrayElemnts	Lesen eines Arrays
deviceSTOP	keine	Steuerung geht in STOP
deviceRUN	keine	Steuerung geht in RUN
projectDownload	keine	Download des Projektes

**Tabelle 5.1: Funktionen des Testautomaten**

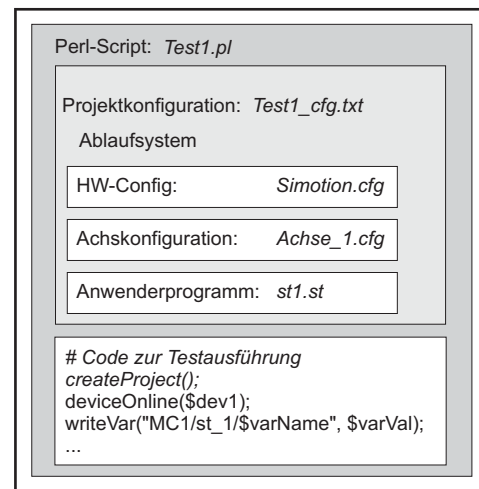
Um eine Verbindung zwischen TA und Steuerung herstellen zu können, ist vom Testautomaten ein Projekt zu generieren. Es enthält die Online-Komponenten der Inbetriebnahme.

Wie bei allen S7-Projekten bildet auch bei SIMOTION die Hardwarekonfiguration (HW-Konfig) die Grundlage eines jeden Projektes. Sie liegt dem Anwender zumeist schon als Datei vor und kann vom Testautomaten nach Upload auf den Server direkt verarbeitet werden.

Die Projekterstellung erfolgt anschließend über Konfigurationsdateien in Textform für:

- das Ablaufsystem
- die Achskonfiguration
- das Anwenderprogramm.

Die Konfiguration des Ablaufsystems zusammen mit dem Anwenderprogramm ist dienstspezifisch, die Achskonfiguration und HW-Konfig dagegen maschinenspezifisch. Wie in Abbildung 5.7 dargestellt, werden die genannten Dateien in ein Perl-Script eingebunden, aus dem ein Projekt generiert wird. Erst mit dem Download dieses Projektes auf die Steuerung besteht eine durchgängige Kommunikation.



**Abbildung 5.7: Testautomat**

Zur Kommunikation zwischen Dienst und Steuerung stehen die Funktionen aus Tabelle 5.1 zur Verfügung. Ein Beispiel eines Lesevorgangs einer globalen Variable des Anwenderprogramms auf der Steuerung verdeutlicht den Zugriff auf die Schnittstelle. Der Befehl

```
$varInh = readVar(MC1/st_1/count);
```

schreibt in die Perl-Variable `$varInh` den Wert der Variable `count`, die sich im aktuellen Projekt auf der Steuerung `MC1` in der Quelle `st_1` als globaler Bezeichner befindet.

Es wird davon ausgegangen, dass das Netz, in dem sich die Steuerung befindet, über Firewall und Proxy vom Internet getrennt ist. Da die Kommunikation mit der Steuerung nur über das Portal statt-

findet – also über eine feste IP-Adresse – kann diese als Ausnahme in die Firewall eingetragen werden. Damit ist eine Verbindung zur Steuerung mit dem TA möglich, andere Hosts im Internet werden weiterhin abgewehrt. Profinet unterstützt diese Funktion [34].

#### **Schnittstelle: Dienst/Portal**

Diese Schnittstelle verbindet den Dienst im Anwenderbrowser mit dem Portal, im Speziellen mit dem Webserver. Aus den Betrachtungen zur Schnittstelle Portal/Steuerung können Aufgaben dieser Schnittstelle abgeleitet werden:

1. Dienstauswahl
2. Dienstparametrierung
3. Steuern der Betriebszustände
4. Auftrag zum Lesen des Fehlerstatus
5. Auftrag zum Lesen, Schreiben sowie Darstellen von Variablen.

Bei diesen Aufgaben handelt es sich um die Darstellung von Informationen und die Weiterleitung von Anwenderaktionen. Die Darstellung erfolgt in einem Browser. Hier ist das HTTP-Protokoll üblich. HTTP greift standardmäßig auf den Port 80 zu. Dieser ist bei der Firewall für die Internetverbindung geöffnet. Bezogen auf die Dynamik der darzustellenden Daten ist festzustellen, dass die Aufgaben 1 und 2 Reaktionen auf Nutzereingaben auszuführen haben. Diese sind derart beschaffen, dass beispielsweise auf die Nutzereingabe der Achsanzahl auch die entsprechende Anzahl Achsdialoge geöffnet wird. Nach [36] handelt es sich hierbei um serverseitige Dynamik. Einsatzfähig sind CGI-Scripte. Aufgrund der Perl-Verwendung im Testautomat wird auch hier auf Perl zurückgegriffen.

Die weiteren Punkte 3 bis 5 erfordern hingegen eine clientseitige Dynamik. Für die Anzeige eines Fehlers kann nicht gewartet werden, bis die Seite neu aufgerufen wird. Als clientseitige Programmiersprache hat sich Java etabliert. Die Abfrage von Statuswerten der Steuerung erfolgt über zyklisches Pollen. Die Kommunikation mit dem Webserver wird über eine *URLConnection* aufgebaut. Diese nutzt das HTTP-Protokoll und kann so den bereits geöffneten Port in der Firewall nutzen. Zu beachten ist, dass auch hier nach jeder Antwort vom Webserver die Verbindung abgebaut wird. Ziel der *URLConnection* ist die CGI-Datei *reDirect.cgi*, die die Anforderung vom Dienst auswertet. Die Auswertung erfolgt auf Grundlage eines vereinbarten Protokolls, das die Übermittlung der SessID sowie die Dienstanforderung, beispielsweise "Lese eine Variable!", vorschreibt. Vom Webserver muss die Anforderung an den Testautomaten weitergeleitet werden.

#### **Schnittstelle: Testautomat/Webserver**

Da sich der Testautomat und der Webserver in einem Intranet befinden<sup>3</sup>, kann hier eine Kommunikation ohne störende Firewall vorausgesetzt werden. Die Kommunikation über Sockets ist auf beliebigen<sup>4</sup>

<sup>3</sup> Im Prototyp ist dies auf einem Rechner realisiert.

<sup>4</sup> Ausgenommen sind Ports, die durch andere Prozesse belegt oder reserviert sind.

Ports möglich. Nach dem Start öffnet der Testautomat als Server einen Socket auf einem freien Port und wartet auf eingehende Verbindungen vom Webserver aus. Die IP-Adresse und der Port des TA sind auf dem Webserver unter der SessID des Nutzers abgespeichert. Der Verbindungsaufbau erfolgt über einen Socket. Der Befehlshandler im TA analysiert die Anforderung und übermittle sie an die Steuerung. Deren Antwort wird entpackt und an den Webserver zurückgesendet. Die Verbindung TA/Webserver wird beendet. Der Webserver leitet den Inhalt der Antwort automatisch an den Offline-Teil und beendet die Verbindung zum Dienst. Die Abbildung 5.8 verdeutlicht den Vorgang des Lesens einer Variable nochmals grafisch.

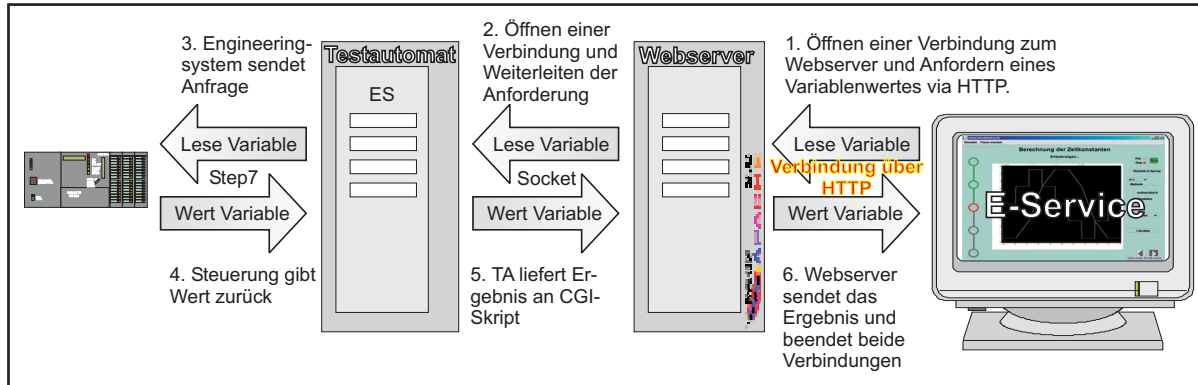


Abbildung 5.8: Lesen einer Variable

Zu beachten ist, dass der TA nicht multitaskfähig ist. In Perl existiert die Möglichkeit, mit dem Befehl *fork* einen quasi-parallelen Zweig in der Programmverarbeitung vom Hauptprozess abzuzweigen. Fork kopiert dabei das Speicherabbild in den Child-Prozess. Da der Testautomat über 100 MB Arbeitsspeicher belegt, ist dies kein praktikabler Weg. Der Nachteil, nur eine serielle Kommunikation zur Verfügung zu haben, muss in Kauf genommen werden. Als Konsequenz ist in der Kommunikationskette ein Befehlspuffer (Befehls-Queue) nötig. Möglich ist eine Implementierung im Webserver oder im Dienst. Die Funktionsweise des Webserver erscheint für diese Aufgabe ungeeignet – die Implementierung im Dienst wird bevorzugt. Zeitkritische Funktionen können aus diesem Grund und der mangelnden Echtzeitfähigkeit des Internets nicht auf diesem Weg zur Verfügung gestellt werden. Sicherheitsfunktionen sind nicht über Internet integrierbar. Es besteht die Anforderung diese auf der Steuerung und über Anwender-Hardware zu realisieren.

#### 5.3.1.4 Expertenzuschaltung

Bei Versagen der automatischen Inbetriebnahme wird dem Anwender die Konsultation eines Experten angeboten.

Die Architektur des Portals stellt dieser Funktion nichts in den Weg. Der Nutzer nimmt Kontakt zum Testautomaten über den Webserver (*reDirect.pl*) auf, indem er seine *sessID* übermittelt. Ist einem Dritten diese ID ebenfalls bekannt, kann er über den Webserver Verbindung zur Steuerung aufnehmen. Abbildung 5.9 verdeutlicht das.

Ein Sicherheitsrisiko stellt diese Möglichkeit nicht dar, da bei jedem Verbindungsaufbau zum Server Username und Passwort überprüft werden. Hat der Experte keine Zugriffsrechte auf dem Portal, so kommt die Verbindung nicht zustande.

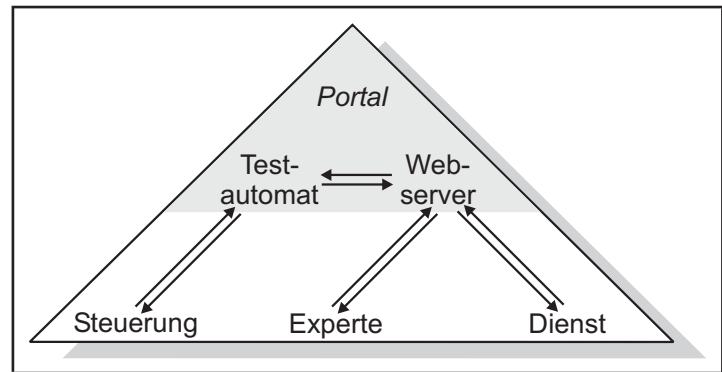


Abbildung 5.9: Experteneinbindung

Die Umsetzung erfolgt im Rahmen dieser Arbeit nicht.

Für den Experten ist ebenfalls eine Oberfläche zu entwerfen, die ihm einen erweiterten Funktionsumfang bereitstellt. Diese Funktionen sollen das detailliertere Wissen des Experten unterstützen. Denkbar sind Darstellungen wie Frequenzgang, Amplitudengang oder Wurzelortskurve.

### 5.3.1.5 Datensicherheit und Zugriffsbeschränkung

Im Prototypen ist die Verwendung der vom Apache-Webserver mitgelieferten Zugriffsbeschränkung als ausreichend zu betrachten. Sie kann für komplette Verzeichnisse eingeführt werden und ist nur mittels eines auf dem Server gespeicherten Logins und Passwortes zu betreten.

Die Datenübertragung wird über eine verschlüsselte Verbindung mittels HTTPS geschützt. HTTPS ist HTTP mit einer Verschlüsselung nach SSL in der Darstellungsschicht des OSI-Referenzmodells. In [15] werden weitere Hinweise gegeben.

### 5.3.1.6 Abrechnungssystem

Abrechnungssysteme für E-Commerce stehen in hoher Anzahl zur Verfügung. Aus diesem Grund soll an dieser Stelle nicht weiter darauf eingegangen werden.

### 5.3.1.7 Optimierung der Struktur

Die Tatsache, dass für jeden Nutzer, der einen Dienst startet, eine eigene Instanz des Engineeringsystems angelegt wird, beschränkt die gleichzeitig aktive Anzahl der Nutzer pro PC. Nach subjektivem Empfinden ist im realisierten Prototypen die obere Grenze bei maximal zwei Instanzen zu setzen. Um eine Anwendung in der Praxis zu ermöglichen, ist hier eine Erweiterung einzuführen, die die aktive Nutzeranzahl steigert. Dafür ist der Aufbau eines Netzes einer Gruppe von Dienstleistungsrechnern "hinter" dem Webserver analog zu Abbildung 5.10 nötig. Soll ein Dienst gestartet werden, so wird ein freier Rechner gesucht und auf ihm der Testautomat gestartet. Nebeneffekt ist eine gesteigerte Flexibilität in Bezug auf Software-Versionen. Es wird für jede Steuerung der passende Dienstleistungsrechner gefunden, ohne dass der Anwender eine Neuinstallation vornehmen muss. Die Aus-

wahl des passenden Rechners erfolgt über das Web-Service-Protokoll UDDI und die Beschreibung der Schnittstellen dementsprechend über WSDL.

Prinzipiell sind der Rechneranzahl nach oben nahezu keine Grenzen gesetzt. Es steht ein beliebig erweiterbares System zur Verfügung.

### 5.3.1.8 Dienstkonfigurator

Die Konfiguration des Dienstes beschränkt sich weitestgehend auf die Anpassung des Dienstes an die Maschine. Solche vom Anwender anzugebenden Daten sind Adressen und Eigenschaften der Antriebe, wie deren maximaler Verfahrbereich. Welche Parameter benötigt werden, ist wiederum dienstspezifisch und wird in die Module

- HW-Konfig,
- Technologieobjekte und
- Dienstparameter

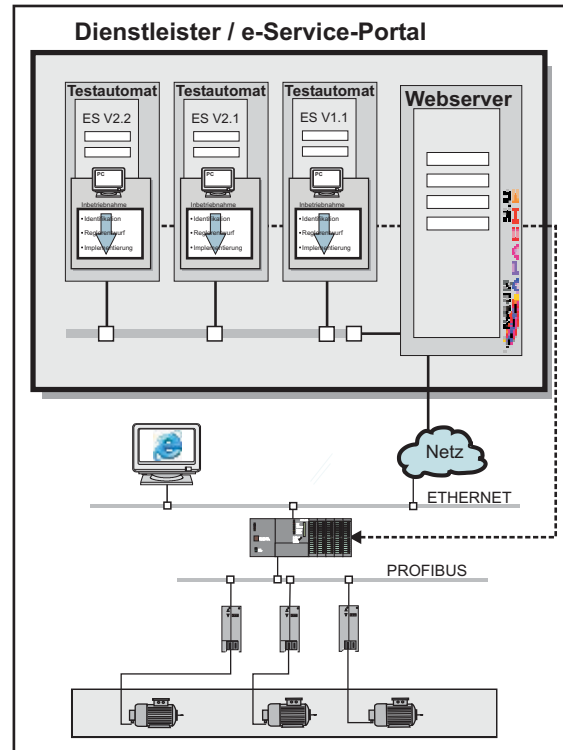


Abbildung 5.10: Optimierte Portalstruktur

unterteilt. Im Konfighandler sind für jedes Modul die Parametersätze zu hinterlegen, so dass bei Aufruf eines Dienstes nur die benötigten Parameter in den Parametrierungsvorgang einbezogen werden. Sie werden vom Webserver unter der Session-ID abgespeichert. Optimal kann dieses "Hinterlegen" ausgeführt werden, wenn auch hier auf die Methoden des Engineeringssystems zurückgegriffen wird. Im ES sind bereits TO-Parameter gespeichert und Plausibilitätsprüfungen abgelegt. Eine Schnittstelle zu diesen Funktionen ist gegenwärtig jedoch nicht vorhanden.

Die Programmierung erfolgt in Perl auf Basis der CGI-Schnittstelle des Servers. Die Parameter werden aus einer Datei eingelesen und mittels Templates zu Formularen im Anwenderbrowser generiert. Die Oberfläche beim Anwender teilt sich wie in Abbildung 5.11 in drei Frames. Im Anwenderbereich (2) erfolgt die Parametrierung des Dienstes. Der Fortschritt sowie Fehlermeldungen werden in der

Navigation

Dienst ParameterDB  
Entwicklungsstand

Parametrierung der Hardwareendschalter der Achse

Bei der Reglerbetriebnahme wird Ihr Antrieb mit im offenen Regelkreis betrieben. Dabei sind die Softwareendschalter nicht wirksam. Bitte geben Sie hierfür die Hardwareadressen der Hardwareendschalter ein.

Parameter	Wert	Erklärung
Einschalten	YES	Abblinderung der Endlagenschalter
Logische Adresse (neg)	228	Logische Adresse - negative Seite
Bitnummer	0	Bitnummer negative Seite
Logische Adresse (pos)	228	Logische Adresse - positive Seite
Bitnummer	1	Bitnummer positive Seite
Lage HV-Endlagenschalter	END_MOUNTED_SWITCH	Lage der HV-Endlagenschalter zum Achsverfahrbereich

Anfrage senden Zurücksetzen

[Thu Oct 21 08:54:09 2004] -> Homingparameter erfolgreich geladen.  
 [Thu Oct 21 08:54:14 2004] -> AxisType erfolgreich geladen.  
 [Thu Oct 21 08:55:23 2004] -> Upload-HWC-Ordg erfolgreich, starte Upload der Achse.  
 [Thu Oct 21 08:55:31 2004] -> Dienst Reglerbetriebnahme wurde gestartet.  
 [Thu Oct 21 08:55:33 2004] -> Starte persönliche Session auf Service-Server mit Log...

Abbildung 5.11: Oberfläche für die Parametrierung des Dienstes



Statuszeile (3) übermittelt. Das Navigationsfenster (1) dient zur Orientierung und zur Auswahl der Dienste.

### 5.3.1.9 Dienststarter

Er veranlasst das Laden der Anwenderoberfläche und das Starten des Testautomaten im Portal. Der Befehl zum Starten wird nach der vollständigen Parametrierung des Dienstes von einem CGI-Script ausgelöst. Aufgrund der begrenzten Rechte des CGI-Scripts ist ein direkter Aufruf des Testautomaten nicht erlaubt. Zu diesem Zweck wurde ein weiteres Script – *Connector.cgi* – erstellt. Mit erweiterten Rechten ausgestattet, lauscht es an einem festen Port auf eingehende Verbindungen vom CGI-Script. Beim Start wird die SessID übermittelt. Eine Ausgabe des Dienststarters ist in Abbildung 5.12 zu sehen.

```

CCCC  OOOO  NN  N  NN  N  EEEEE  CCCC  TTTTT  OOOO  RRRRR
C   C   O   O  NN  N  NN  N  E       C   C   TT   O   O  R   R
C   O   O   O  NN  N  NN  N  E       C       TT   O   O  R   R
C   O   O   N  N  N  N  N  N  EEEEE  C       TT   O   O  RRRRR
C   O   O   N  N  N  N  N  N  E       C       TT   O   O  RRR
C   C   O   O  N  N  N  N  N  N  E     C   C   TT   O   O  R   R
CCCC  OOOO  N   NN  N  NN  EEEEE  CCCC  TT   OOOO  R   R

Connector gestartet. Warte auf Verbindungsaufruf.

Received a connection from pc18.ist, Port 1536
Receiving:
  SessionID:  JUsMnVsXJUV-RmxTtgsFYA
  Service:    RegIIbN
  Port:      12345
Starte Dienst RegIIbN
Antwort an USER = OK
*****
Warte auf die naechste Verbindung

```

**Abbildung 5.12: Ausgabe des Connectors bei Verbindungsaufnahme**

Der Start des Automaten ist nach ca. einer Minute<sup>5</sup> abgeschlossen, worauf das Laden der Anwenderoberfläche freigegeben wird.

## 5.3.2 Die Anwenderschnittstelle/Offline-Komponente

### 5.3.2.1 Programmierung und Benutzer-Oberfläche

Die Anwenderschnittstelle ist auf einem handelsüblichen PC mit einem Browser lauffähig und als Java-Applet in eine HTML-Seite eingebunden. Um die Oberfläche flexibler gestalten zu können, sind die Bedienelemente in ein eigenes Fenster eingebunden.

Die Anwenderschnittstelle enthält Funktionen zum Steuern des Dienstablaufes und zur Kommunikation sowie den Offline-Teil des Dienstes.

### 5.3.2.2 Ablaufsteuerung

Die Funktionalität zur Ablaufsteuerung ist auf beide Seiten des Dienstes verteilt. Auf der Seite der Steuerung wird sie für den Ablauf der Echtzeitfunktionalität benötigt. Mittel zur Beeinflussung des Ablaufes sind:

- Starten und Beenden von Tasks

<sup>5</sup> Auf einem Pentium 4 mit 2,6 GHz und 516 MB RAM

von/nach	OFFLINE	STOP	RUN	ERROR	Controllerstatus
OFFLINE	-	X	-	X	STOP
STOP	X	-	X	X	STOP
RUN	-	X	-	X	RUN
ERROR	X	X	X	-	RUN/STOP

*Tabelle 5.2: Mögliche erlaubte (X) Dienstzustandsübergänge*

- Setzen von Freigaben
- Events des Fehlerhandlers
- Steuerung der Betriebszustände

Unterschieden werden Dienstzustände und Betriebszustände. Die Dienstzustände orientieren sich an den Teilschritten im Dienstablauf. Sie sind an bestimmte Betriebszustände der Steuerung gebunden. Die Betriebszustände spiegeln den Zustand der Kommunikation und den Zustand der Steuerung wieder. Ein Übergang von einem Zustand zu einem anderen ist an Konventionen gebunden und kann von Anwenderseite oder von der Steuerung aus erfolgen. Tabelle 5.2 stellt eine Übersicht dar.

### 5.3.2.3 Kommunikationsmodul

Die Kommunikation mit dem Webserver wird über vier Stufen abgewickelt. Der Oberfläche stehen die Grundfunktionen:

- `readVar(String varName)`
- `readSysVar(String varName)`
- `readArray(String varName, int varAnz)`
- `writeVar(String varName, double varVal)`

zur Verfügung. Mit ihnen können Variablen auf der Steuerung gelesen und geschrieben werden. Da ein "Hängenbleiben" der Oberfläche nicht erwünscht ist, werden diese Funktionen nur über eigene Prozesse aufgerufen. Der Ablauf der Kommunikation ist in Abbildung 5.13 dargestellt. Die Befehlsaufbereitung generiert aus den Funktionsparametern und dem Befehl (bspw. `readVar` oder `readSysVar`) einen String, getrennt durch einen String-Delimiter (1).

```
readSysVar_._ipoclock
```

Dieser String wird an die Befehlsqueue übergeben (2), die ihn an den Ausgangs-Puffer weiterreicht (3a). Bei der Übergabe wird ein freier numerierter Eingangs-Puffer reserviert. Die Nummer erhält der aufrufende Prozess (3b) und der Befehlsstring. Im Anschluss überwacht der aufrufende Prozess



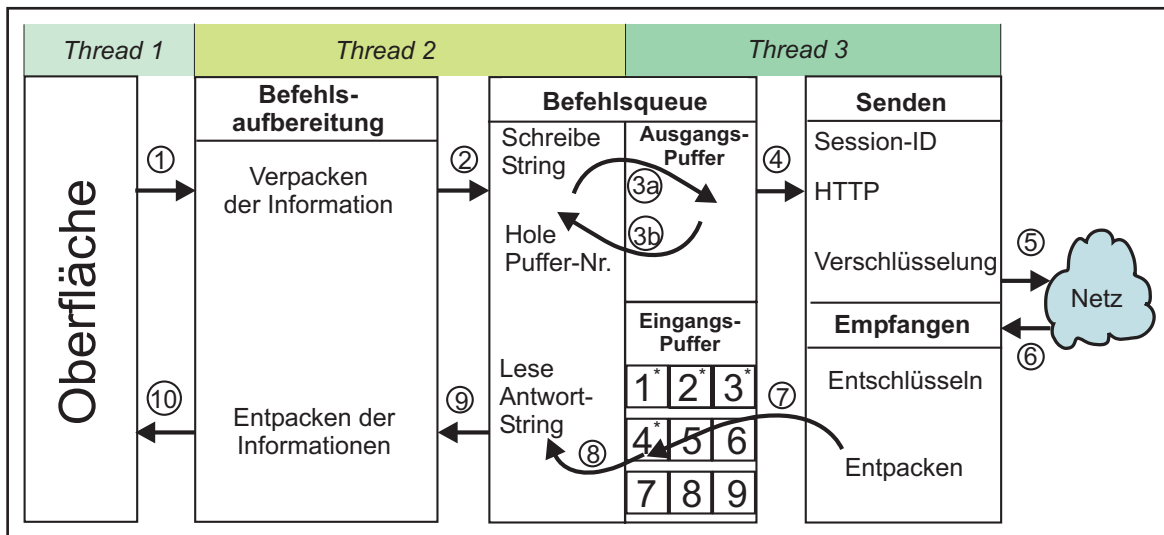


Abbildung 5.13: Kommunikation im Applet über die Befehlsqueue

"seinen" zugewiesenen Eingangspuffer auf eine Antwort (8). Der erste Befehl des Ausgangspuffers wird in einem separaten Prozess nach dem FIFO-Prinzip an die unterlagerte Schicht weitergereicht (4). Dort wird der String mit der SessID markiert und per HTTP versendet (5). Die erhaltene Antwort (6) wird entschlüsselt und in den reservierten Puffer abgelegt (7). Nach dem Entpacken (8/9) steht die Antwort zur Weiterverarbeitung zur Verfügung (10).

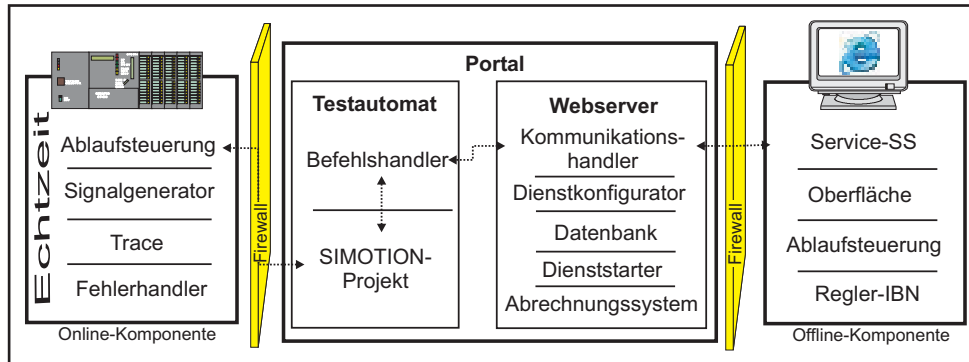


Abbildung 5.14: Module im Gesamtsystem

Dieses Vorgehen entspricht dem FIFO-Prinzip, nach dem die Befehle in der zeitlichen Reihenfolge ihres Absetzens abgearbeitet werden. Zeitkritische Funktionen sind nicht realisierbar, da keinerlei Sicherstellung der Verarbeitung innerhalb einer Zeitspanne erfolgt.

Abbildung 5.14 zeigt zusammenfassend das Portal mit der Offline- und Online-Komponente und die darin enthaltenen Funktionsmodule. Sie sind im Internet verteilt und sehr wahrscheinlich über Firewall voneinander getrennt. Das Portal beinhaltet die Schnittstelle zur Steuerung und übernimmt eine zentrale Vermittlerrolle im gesamten Dienstablauf. Dienstspezifische Funktionen sind verteilt in der Offline- und Online-Komponente. Die Steuerung übernimmt dabei vor allem zeitkritische Funktionen (Trace, Überwachungen). Der Offline-Teil enthält das Rechenwerk zur Erfüllung des Dienstes.

### 5.3.2.4 Programmierung

Der Dienst wird nach Abbildung 3.2 in eine Offline-Toolkette und eine Online-Toolkette unterschieden. Die Online-Toolkette ist auf der Steuerung angesiedelt und in ein SIMOTION-Projekt eingebunden, der Offline-Teil wird auf der Anwenderseite in Java ausgeführt.

Die Programmierung erfolgt objektorientiert. Der Byte-Code wird vom Portal geladen und als Applet in den Webbrowser eingebettet. In

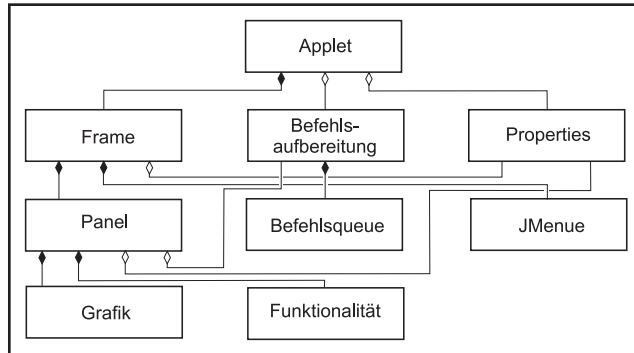


Abbildung 5.15: Objektbeziehungen im Applet

das Applet werden alle Objekte eingebunden. Die Abbildung 5.15 zeigt die wesentlichen Objekte für die Erbringung allgemeiner Aufgaben. Dazu gehört die Kommunikationsschnittstelle mit Befehlsaufbereitung und Befehlsqueue. Sie bildet die Grundlage zur Kommunikation und baut nach dem Dienststart das Kommunikationsdreieck Applet/Server/TA/Steuerung auf. Diese Funktionsgruppen realisieren allgemeine Aufgaben. Um die Dienste leicht austauschen zu können, erfolgt des Weiteren eine Unterteilung in dienstspezifische Funktionen. Sie werden zusammen mit der Oberfläche in Panels eingebunden, die sich in einem Frame mit den allgemeinen Funktionen vereinigen. Zusammen bilden sie den Offline-Teil.

Die Trennung von allgemeinen und spezifischen Funktionen erlaubt ein sehr flexibles Hinzufügen weiterer Dienste. Unterstützung gibt das Java-typische dynamic-binding, das ein Laden der Dienste sogar zur Laufzeit ermöglicht.

Die Abbildung 5.16 zeigt wichtige Objekte des Dienstes Reglerinbetriebnahme. In der obersten Ebene befindet sich die Ablaufsteuerung. Sie koordiniert den Dienst teils eigenständig und teils mit Hilfe von Benutzereingaben über die Oberfläche. Die zweite Ebene ist nach den funktionellen Einzelschritten Referenzierung, Identifikation, Reglerberechnung und Validierung aufgeteilt. Die

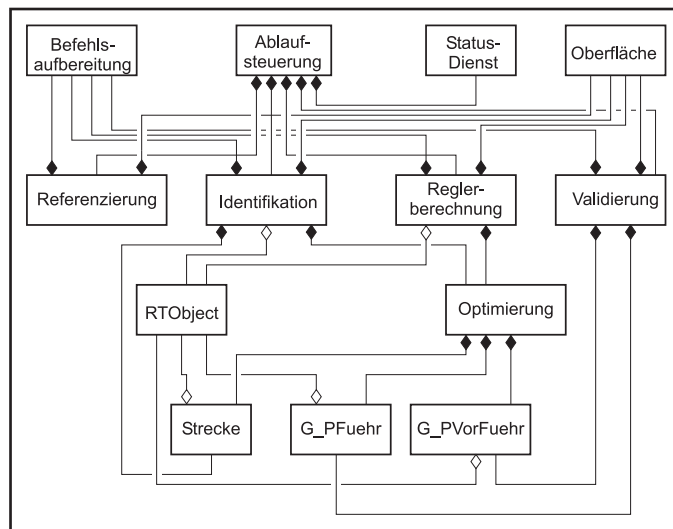


Abbildung 5.16: Objekte im Reglerinbetriebnahmedienst

Die Kommunikation mit der Steuerung erfolgt über die von der Befehlsaufbereitung zur Verfügung gestellten Funktionen. Die unteren Ebenen zeigen auszugsweise die Objekte der Regelungstechnik zur Identifikation und Reglerberechnung. Die Struktur ist sehr kompliziert. Der Abschnitt 5.5.1.1 geht näher darauf ein.

### 5.3.2.5 Oberfläche

In Abbildung 5.17 ist die Oberfläche in der Identifikation beispielhaft dargestellt. Sie ist Wizard-ähnlich aufgebaut und enthält Elemente zur Ablaufsteuerung, Visualisierung, Parametrierung und für Hilfsfunktionen. Sie enthält Elemente allgemeiner Funktionen (Menü) und dienstspezifischer Funktionen.

Zentrales Element ist die Visualisierung des Trace. Schaltflächen parametrieren die Randbedingungen der Identifikation. Über die Vor- und Zurück-Buttons steuert der Anwender den Ablauf. Kleinste Schritte sind die Einzelschritte Referenzierung, Identifikation, Reglerberechnung und Validierung, wobei in der Identifikation der aufgezeichnete Trace vom Anwender vor der Berechnung kontrolliert werden kann. Der innere Ablauf dieser Schritte wird von der Ablaufsteuerung automatisch durchgeführt.

Hilfen werden dem Anwender über das Menü und über Tooltips gegeben. Tooltips wie in Abbildung 5.18 beschreiben die Funktionsweise von Buttons und werden sichtbar, wenn der Mauszeiger über einem Grafikelement verweilt. Die Menüfunktionen zur Hilfe verweisen auf HTML-Seiten mit Hilfetexten auf dem Portal.

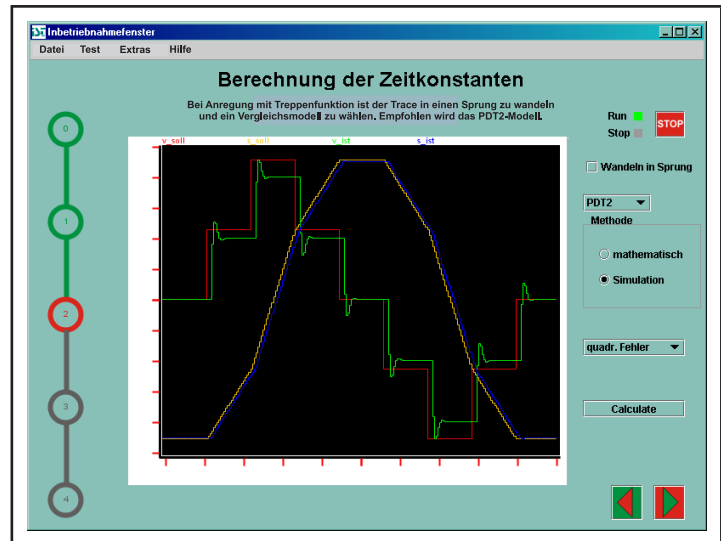


Abbildung 5.17: Oberfläche in der Identifikation

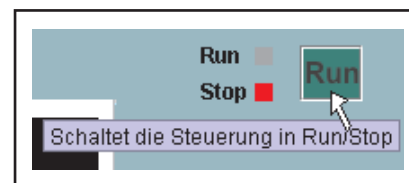


Abbildung 5.18: Tooltips geben dem Anwender Hilfestellung

### 5.3.3 Maschinschnittstelle/Online-Komponente

Der Anspruch des Dienstes an das SIMOTION-Projekt wird auf die Achsen und die Einbindung der Funktionsgruppen in das Tasksystem begrenzt. Der Projektaufbau wird so einfach wie möglich gehalten, um das Projekt für ein breites Maschinenspektrum zu nutzen. Die Parametrierung der Achsen umfasst hauptsächlich die Achseigenschaften:

- Achstyp
- Referenzierart
- Hardwareendlagenschalter
- Hardware-Adressen der Achse und des Gebers.

Task	Programm	Funktion
StartupTask	st_1.StartUp	Initialisieren des Programms
BackgroundTask	st_1.schrittWeiter st_1.startMotTask	steuerungsseitige Ablaufsteuerung Starten der zum Schritt gehörenden Motion-Tasks
Motion-Tasks_1	st_1.referAxis	Referenzieren der Achse
Motion-Tasks_2	st_1.testNorm	Überprüfen der Normierung der Achse
Motion-Tasks_3	st_1.prepareIdent	Vorbereiten der Achse auf Identifikation
Motion-Tasks_4	st_1.validate	Validierung
Motion-Tasks_5	st_1.online	Online-Verfahren
TechnologicalFaultTask	st_1.TechFault	Behandlung technologischer Fehler
PeripheralFaultTask	st_1.PeriFault	Behandlung von Ausbaufehlern der Peripherie
IPOsynchronousTask	st_1.fktGen_1 st_1.fktGen_2 st_1.traceMe st_1.errHandl	Funktionsgenerator für Treppenfunktionen Funktionsgenerator für PRNS Aufzeichnung der Verläufe Überwachung der Istwertverläufe und Verriegelungen
ShutdownTask	st_1.ShutDown	Behandlung RUN-STOP-Übergang

*Tabelle 5.3: Belegung des Tasksystems*

Deren Parametrierung erfolgt im Dienstkonfigurator des Webservers durch den Anwender.

Das Ablaufsystem wird fest vorgegeben und ist in Tabelle 5.3 dargestellt. Die "schnellste" vom Anwender zu programmierende Task ist die IPOsynchronous-Task, die im Projekt synchron zum Servo-Takt im Verhältnis 1 : 1 betrieben wird. In ihr werden alle Funktionen, die Echtzeitbedingungen erfordern, eingebunden. Signalgenerator und Tracefunktion stellen ebenso die Anforderung an einen zeitlich determinierten Zyklus mit kleinen Zykluszeiten wie der Fehlerhandler.

### 5.3.4 Resultierende Struktur

Die Struktur des Systems zur Dienstleistung ist in Abbildung 5.19 dargestellt. Sie zeigt das Dienstleistungs-Portal mit der enthaltenen Dienstbibliothek. Die Beschreibungen der vorhandenen Dienste werden in einer Datenbank gespeichert. Darin wird ebenfalls jedem Dienst ein Link, der zum Ablageort zeigt, zugeordnet. Der Ablageort kann sich auf dem gleichen Rechner, im gleichen Netz oder auf einem entfernten Rechner in einem angeschlossenen Netz befinden. Weiterhin sind in der Datenbank Parameter und Strategien zur Initialisierung des Dienstes hinterlegt.

Der Start eines Dienstes beginnt mit seiner Parametrierung und Konfiguration durch den Anwender. Anschließend wird der Testautomat gestartet und ein dienstspezifisches SIMOTION-Projekt generiert. Gleichzeitig werden die Routinen der Offline-Komponente auf den Anwender-PC geladen und im Browser zur Ausführung gebracht. Mit dem Laden des SIMOTION-Projektes in die Steuerung ist

der Dienst bereit.

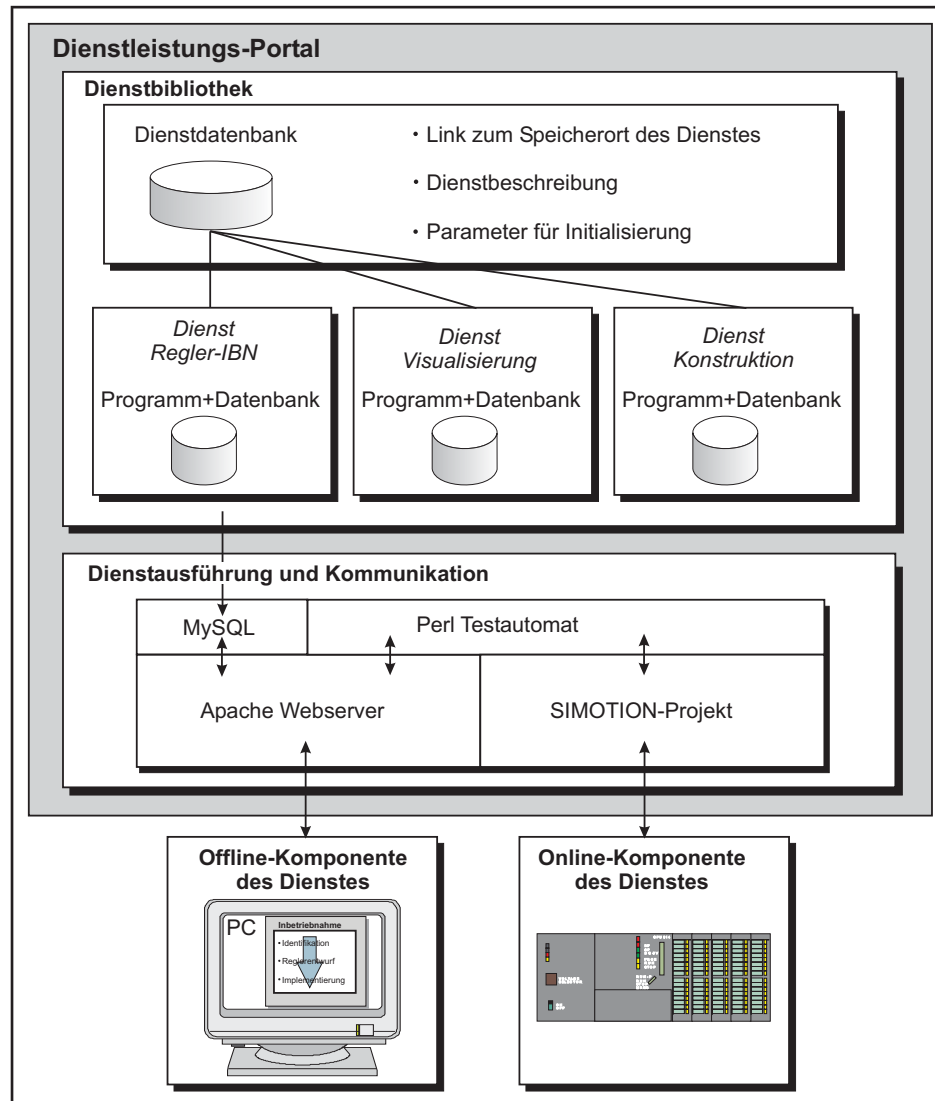


Abbildung 5.19: Resultierende Struktur

## 5.4 Ausführung einer Online-Regelkreisinbetriebnahme

Die Inbetriebnahme des P-Reglers hat das Ziel, einen Verstärkungsfaktor  $K_P$  zu ermitteln, mit dem der Regelkreis ein bestimmtes dynamisches Verhalten aufweist. Das Verhalten wird mittels einer Zielfunktion bewertet. Typische Gütekriterien von Zielfunktionen sind in Kapitel 3.3 betrachtet. Im Folgenden wird eine Online-Reglerinbetriebnahme realisiert.

### 5.4.1 P-Regler

Die Einstellung der Reglerparameter erfolgt auf heuristischem Wege. Ausgehend von einem Startwert wird sich dem gewünschten Verhalten schrittweise angenähert. Hierbei muss der Zusammen-

hang zwischen Regelkreis-Dynamik und Reglerparametern bekannt sein. Bei einem oder zwei Reglerparametern ist das ein praktikabler Weg. Ist eine höhere Anzahl von Parametern gesucht, sind die Zusammenhänge nicht mehr trivial und eine gerichtete Suche nur mit sehr viel Erfahrung des Inbetriebnehmers möglich.

Vor Beginn sind die unterlagerten Regelkreise einzustellen und die Normierung des Antriebs zu überprüfen. Die Prüfung erfolgt mittels eines geschwindigkeitsgeregelten `_move`-Befehls. Erreicht die Achse die vorgegebene Geschwindigkeit in engen Grenzen, so ist die Umsetzung der Stellgrößen richtig. Andernfalls ist das Übersetzungsverhältnis vom Anwender zu korrigieren. Anschließend wird ein Referenzvorgang ausgeführt, der eine definierte Position der Achse im Verfahrbereich herstellt. In der Steuerung wird die Reglerstruktur implementiert. Erfahrungsgemäß sind Verstärkungsfaktoren von  $K_P > 40 \frac{1}{s}$  typisch für "kleine" Servoantriebe. Als Startwert für die Suche wird für  $K_P$  vom Anwender ein Startwert gewählt. Auf den Lageregelkreis werden rampenförmige Sollwerte eingeschaltet. Zur Beurteilung des Regelverhaltens können die Lageist- und Lagesollwertverläufe, sowie die zugehörigen Geschwindigkeitsverläufe aufgezeichnet und anschließend bewertet werden. Gewählte Zielkriterien des P-Reglers sind ein schnelles Erreichen des Geschwindigkeitssollwertes, sowie eine maximale Überschwingweite. Beide Kriterien stehen in engem Zusammenhang. Die Strategie orientiert sich an der Überschwingweite. Die Suche verläuft nach folgendem Prinzip: Ist die Überschwingweite mit dem Startwert kleiner als die gewünschte, wird die Reglerverstärkung erhöht. Die Schrittweiten der Erhöhung oder Verkleinerung werden prozentual angegeben. In Experimenten wird festgestellt, dass Erhöhungen des aktuellen Wertes um 20% schnell Fortschritte erzielen. Die Erhöhung wird ausgehend vom Startwert so lange durchgeführt, bis die festgelegte Schwelle (Überschwingweite) überschritten wird. Danach wird in 10%-Schritten nach unten gesucht, bis die Schwelle unterschritten wird. Daraufhin erfolgt eine Erhöhung um 5%, bis eine Überschreitung der Schwelle stattfindet. Die Suchrichtung wird dann umgekehrt und die Suche mit halbiertes Schrittweite fortgesetzt, bis das Ergebnis im Zielbereich liegt. Ein beispielhafter Verlauf der Suche ist in Abbildung 5.20 links für eine 3%-Überschwingweite dargestellt. Das resultierende Verhalten des Regelkreises ist im rechten Bild zu sehen.

Für die automatische Inbetriebnahme ergeben sich einige Erweiterungen zur Wahrung der Sicherheit von Mensch und Maschine. Hierfür wird im Fehlerhandler eine zyklische Überwachung implementiert. Gefahr für die Maschine besteht besonders bei stark schwingenden oder instabilen Reglereinstellungen. Diesen wird mit einem vom Anwender als sicher befundenen Reglerdatensatz begegnet. Auf der Steuerung befindet sich immer der sichere (passiv) und der berechnete Datensatz (aktiv). Während der Verfahrbewegung wird die Geschwindigkeit ständig in der IPOsynchronous-Task überwacht. Überschreitet sie eine Grenze, so wird auf den sicheren Reglerdatensatz umgeschaltet und die Bewegung gestoppt. In Abbildung 5.21 ist links ein stark schwingender Geschwindigkeitsverlauf dargestellt. Im rechten Bild ist der Verlauf mit eingeschalteter Stabilitätskontrolle zu sehen. Nach Einwechseln des Anwenderdatensatzes wird in dem Beispiel die Bewegung gestoppt und die Rampe fortgesetzt. Der Geschwindigkeitsverlauf ist stabil. Alternativ kann beim Auftreten von übermäßigen

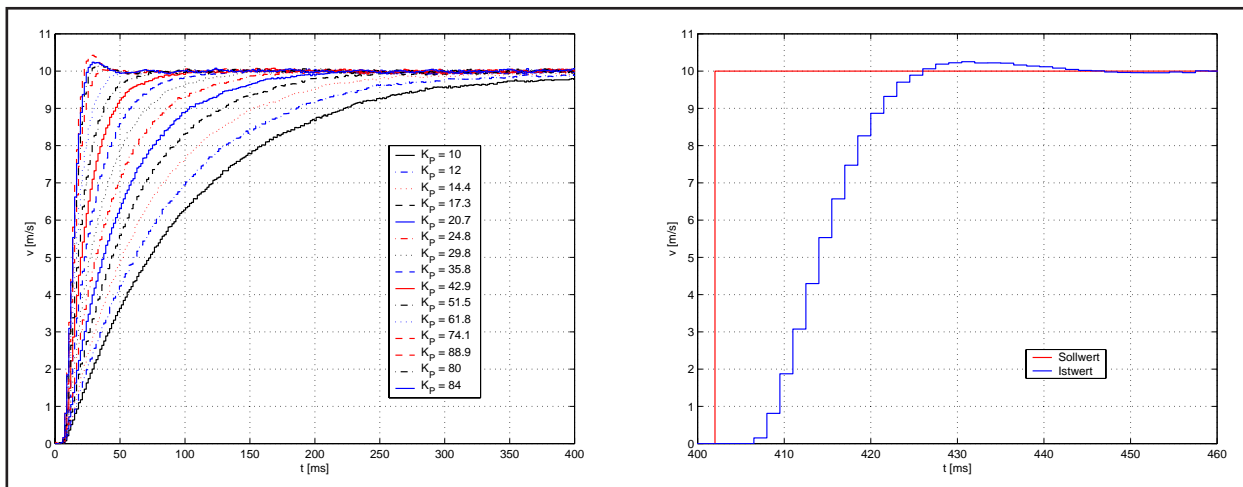


Abbildung 5.20: Verlauf (links) und Ergebnis (rechts) der Suche des Verstärkungsfaktors beim P-Regler

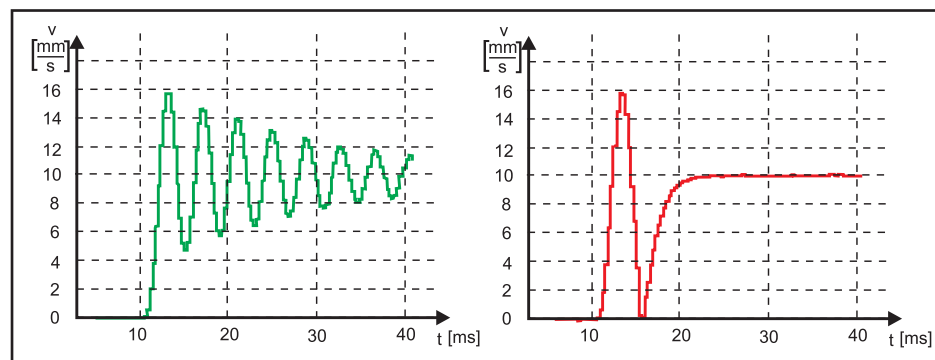


Abbildung 5.21: Erkennung stark schwingender Reglerdaten

Schwingungen der Verstärkungsfaktor des Reglers so oft halbiert werden, bis kein Schwingen mehr auftritt. In [18] werden damit Erfolge erzielt.

Weitere Bemerkungen zur Sicherheit während der Inbetriebnahme, speziell aus der Ferne, werden in Kapitel 5.6 dargelegt.

## 5.4.2 PV-Regler

Soll der Regelkreis mit Vorsteuerung betrieben werden, so ist der Vorfilter auf die Regelstrecke abzustimmen. Das heißt, die Summenzeitkonstante  $T_\Sigma$  der Strecke ist abzüglich der Totzeit in den  $PT_1$ -Vorfilter einzusetzen. Die Totzeit für den Regler berechnet sich nach Formel (2.2). Bei der Aufzeichnung sind jedoch andere durch das Tasksystem bedingte Totzeiten enthalten. Die Gleichung ist dem Einzelfall auf Grundlage von Abbildung 2.9 anzupassen. Für die Ersatzzeit des Filters gilt:

$$T_{Filter} = T_\Sigma - T_{tot}. \quad (5.1)$$

Da der Regelkreis bereits für ein definiertes Überschwingen eingestellt ist, kann aus dem Verstärkungsfaktor  $K_P$  und dem Punkt  $P_m = (h_m, t_m)$  die Zeitkonstante der Regelstrecke ermittelt



werden. Der Filter in SIMOTION ist mittels eines  $PT_1$ -Gliedes umgesetzt. In der folgenden Berechnung wird für die Regelstrecke ein Proportionalelement erster Ordnung verwendet, so dass sich eine Reglerstruktur nach Abbildung 5.22 ergibt.

Die Übertragungsfunktion des Regelkreises für Führungsverhalten ist

$$G_F(s) = \frac{K_P \cdot \frac{1}{T_\Sigma \cdot s + 1} \cdot \frac{1}{s}}{1 + \frac{K_P}{s(T_\Sigma \cdot s + 1)}} \quad (5.2)$$

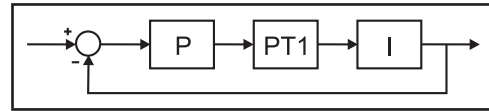


Abbildung 5.22: P-Regler

und multipliziert mit  $\frac{s \cdot (T_\Sigma \cdot s + 1)}{K_P}$  erhält man

$$G_F(s) = \frac{1}{\frac{T_\Sigma}{K_P} s^2 + \frac{1}{K_P} s + 1}. \quad (5.3)$$

Für den Lageregelkreis wird ein periodisches System (3.11) angesetzt, weil der Lageregelkreis mit P-Regler schwingfähig werden kann. Ein Koeffizientenvergleich bringt die Zusammenhänge

$$T^2 = \frac{T_\Sigma}{K_P} \quad \text{und} \quad 2DT = \frac{1}{K_P}. \quad (5.4)$$

Die Berechnung der Dämpfung  $D$  des Regelkreises wurde bereits in Kapitel 3.1.2.3 aus der Übergangsfunktion (3.12) und deren Ableitung (3.13) für den Punkt  $P_m$  hergeleitet. Gleichung (3.15) stellt hierbei eine Bestimmungsgleichung für  $T$  des Regelkreises dar. Durch geeignetes Umformen von (5.4) erhält man die Summenzeitkonstante  $T_\Sigma$  der Regelstrecke. Die Summenzeitkonstante enthält die Totzeit des Regelkreises, welche sich nach Gleichung (2.2) mit  $T_i = 0,433 \text{ ms}$  und  $T_o = 0,150 \text{ ms}$  berechnet.

In Tabelle 5.4 sind Experimente für unterschiedliche Überschwingweiten des Lageregelkreises aufgelistet.

$h_{\ddot{u}}$	$h_m$	$t_m$	$K_p$	D	$T$ nach (3.15)	$T_\Sigma$ nach (5.4)
ca. <b>3%</b>	1,025	29,5 ms	84	0,76	6,1 ms	3,1 ms
ca. <b>7%</b>	1,068	25,0 ms	98	0,65	6,0 ms	3,5 ms
ca. <b>10%</b>	1,110	22,5 ms	105	0,57	5,8 ms	3,5 ms
ca. <b>15%</b>	1,140	20,5 ms	115	0,53	5,2 ms	3,1 ms
ca. <b>20%</b>	1,190	19,0 ms	125	0,46	5,3 ms	3,5 ms

Tabelle 5.4: Identifikation im Online-Verfahren für  $T_a = 1,5 \text{ ms}$

Zu erkennen ist, dass die Zeitkonstante  $T$  des Regelkreises mit zunehmendem  $K_P$  kleiner wird. Die daraus berechneten Summenzeitkonstanten sind – wie zu erwarten – nahezu konstant. Nach Mittelung der Werte für die untersuchten Überschwingweiten ergibt sich eine Summenzeitkonstante der Regelstrecke von  $T_\Sigma = 3,34 \text{ ms}$ . Abzüglich der Totzeit von  $T_t = 2,08 \text{ ms}$  erhält man eine Ersatzzeit für den Filter  $T_{\text{Filter}} = 1,26 \text{ ms}$ .

Wird die Vorsteuerung aktiviert und die ermittelte Ersatzzeitkonstante eingesetzt, so erhält man den Kurvenverlauf in Abbildung 5.23 auf eine rampenförmige Lagesollwertaufschaltung. Ist der Geschwindigkeitsverlauf wie in Abbildung 5.23 bei Anregung mit einer Rampe überschwingfrei und



stetig mit einem Wendepunkt, so ist die Reglerparametrierung/Identifikation erfolgreich. Zur weiteren Verbesserung des Übergangsverhaltens ist in SIMOTION DSC einzuschalten. Damit wird der Regler im Antrieb mit Zykluszeiten von gegenwärtig  $T_{aDSC} = 0,125\text{ms}$  gerechnet. Das Resultat ist in Abbildung 5.24 zu sehen.

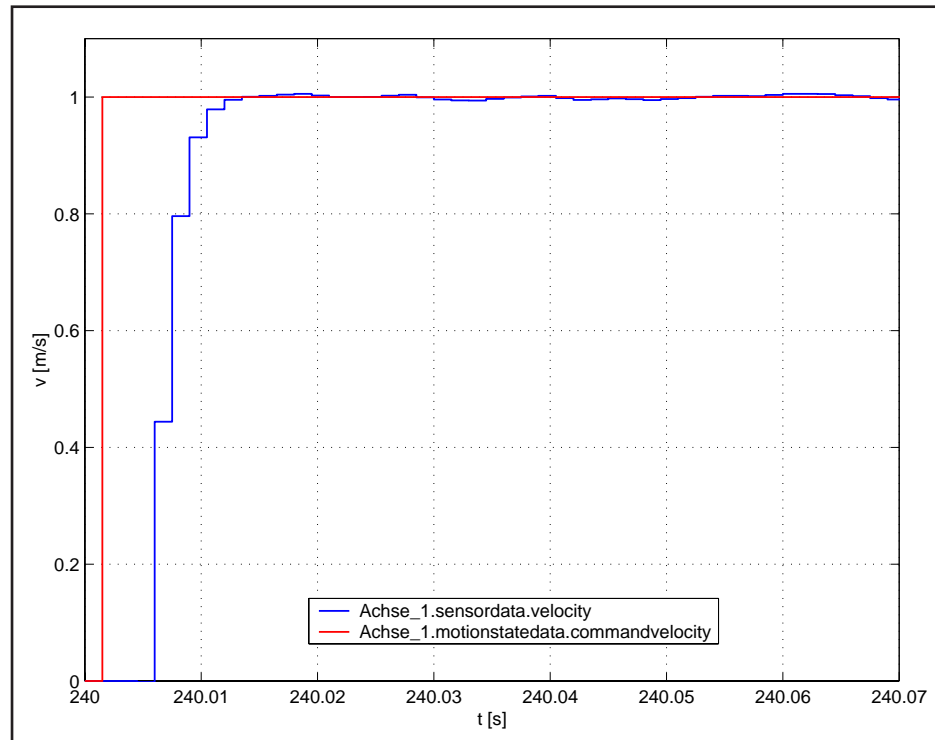


Abbildung 5.23: P-Regler mit Vorsteuerung nach der Online-Inbetriebnahme

### Bemerkungen zum Vorfilter

Das dynamische Verhalten des Vorfilters soll möglichst gut dem Verhalten der Strecke entsprechen. Ist die Zeitkonstante  $T_{Filter}$  identisch mit der Verzögerung des Antriebs, muss der Lageregler nur den Modellfehler zwischen Antrieb und  $PT_1$ -Filter ausregeln. Um diesen so klein wie möglich zu halten, ist der Drehzahlregelkreis nahezu überschwingfrei einzustellen. Ist der Fehler der Zeitkonstante des Filters zu groß zum Antrieb, resultieren die Geschwindigkeitsverläufe aus Abbildung 5.25.

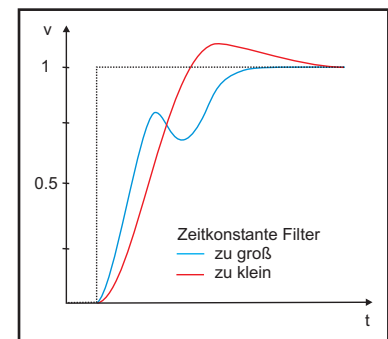


Abbildung 5.25: Auswirkungen falscher Ersatzzeitkonstanten

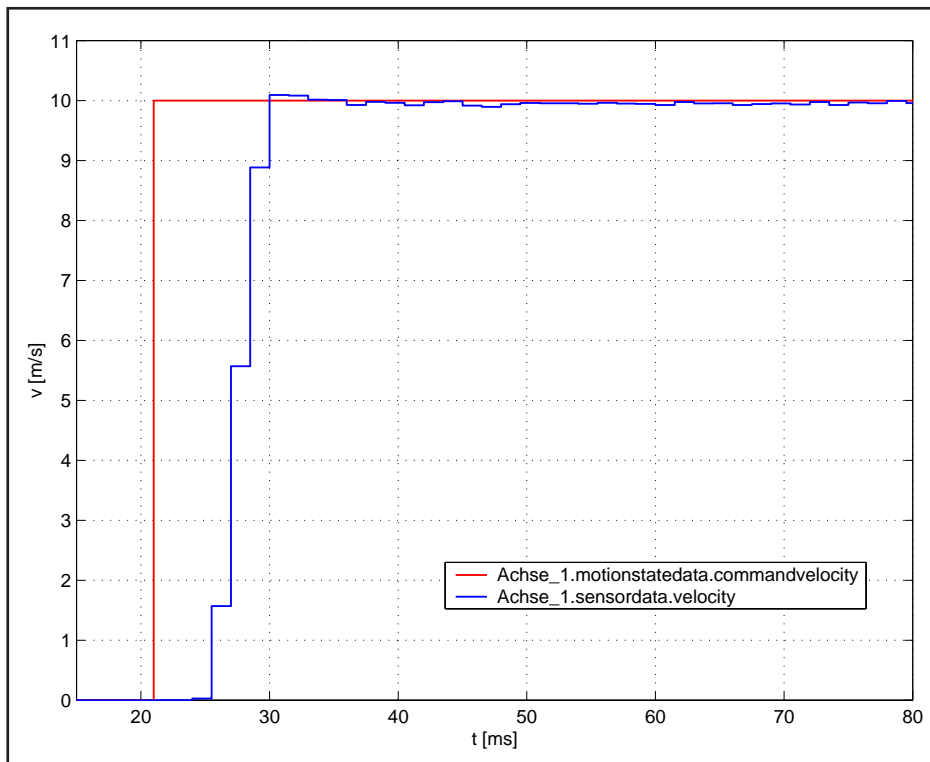


Abbildung 5.24: P-Regler mit Vorsteuerung und DSC

## 5.5 Ausführung einer Offline-Regelkreisinbetriebnahme

Der Ablauf ist angelehnt an das beschriebene Vorgehen in Kapitel 3 und der VDI-Richtlinie 3685 [72].

Vor der Inbetriebnahme des Lagereglers sind wie bei der Online-Inbetriebnahme die unterlagerten Regelkreise einzustellen. Die Überprüfung der Normierung sowie die Referenzierung der Achsen schließen sich daran an.

### 5.5.1 Identifikation

Untersuchte grafische und arithmetische Verfahren zur Identifikation haben in Experimenten nicht den erwünschten Erfolg erzielt. Probleme bereitet die geringe Anzahl von Werten, die den charakteristischen Kurvenzug beschreiben. Zudem wird ein Verfahren gesucht, dass die Totzeiten ermittelt. Im Rahmen der Offline-Reglerinbetriebnahme wird für die Identifikation die Methode der Simulation gewählt.

In die Softwarestruktur des E-Service ist dazu ein Simulationsmodul implementiert.

#### 5.5.1.1 Simulator und Regelungsbibliothek

Im Dienst Reglerinbetriebnahme nimmt das Simulationsmodul einen bedeutenden Stellenwert ein. Gegenstand der Simulation sind Strukturen der Regelungstechnik in offener und geschlossener Form.

Anwendung findet die Simulation in der Identifikation und Reglerparametrierung.

### Objekte der Regelungstechnik

Eine typische Struktur ist in Abbildung 5.26 dargestellt. In der Simulation werden die im Signaldurchlauf enthaltenen Objekte nach ihrer Wirkung auf das Signal betrachtet. In diesem Sinne wird in die Objekte

- Sources – Signalgeneratoren
- Branches – Signalverzweigungen
- Merges – Signalzusammenführung
- Signalmanipulatoren
- Sinks – Signalaufzeichnern

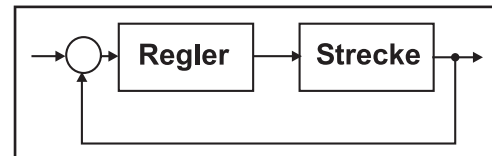


Abbildung 5.26: Typische Reglerstruktur

unterschieden. Deren Symboldarstellung zeigt Abbildung 5.27. Mit Hilfe der gewählten Objekte kann ein großer Teil der regelungstechnisch relevanten Strukturen mit linearem Verhalten nachgebildet werden.

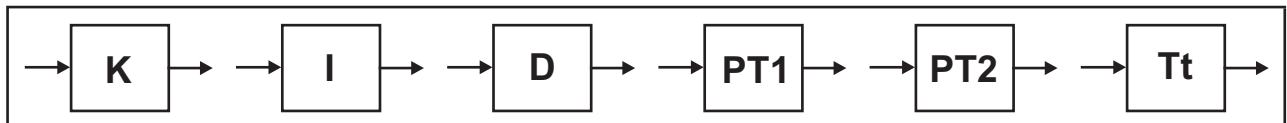


Abbildung 5.27: Objekte zur Simulation

Signalgeneratoren speisen Signale in das Modell ein. Die vorgestellte Auswahl wird um die Möglichkeit, aufgezeichnete Zahlenreihen auszugeben, erweitert und in Abbildung 5.28 als Array bezeichnet.

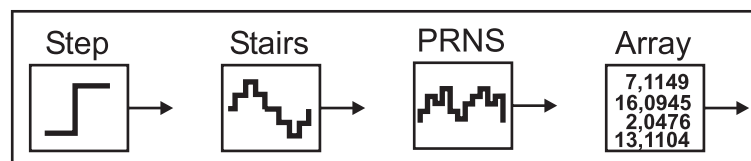


Abbildung 5.28: Signalquellen

Manipulatoren bilden Teile der simulierten Prozesse/Funktionen nach und ändern auf diese Weise das Signal. Unterschieden werden Übertragungsgleichungen

- der Regelungstechnik nach Abbildung 5.29
- der Mechanik
- der Mathematik.

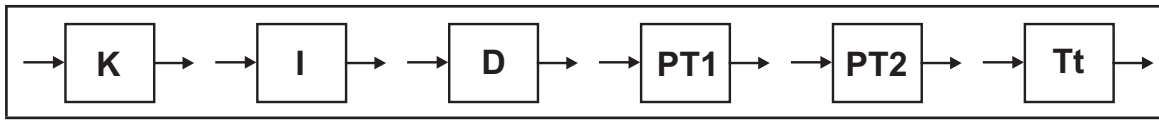


Abbildung 5.29: Einige Manipulatoren der Regelungstechnik

Die mathematischen Funktionen finden Einsatz in der Gütebewertung der Teilaufgaben Identifikation und Reglerberechnung. In Abbildung 5.30 sind vier typische Manipulatoren zur Gütebewertung dargestellt.

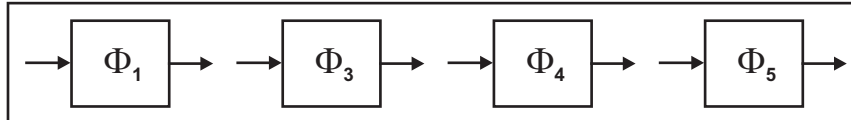


Abbildung 5.30: Manipulatoren zur Gütebewertung

### Verschaltung der Objekte

Die programmtechnische Umsetzung bildet diese vorgestellte Strukturierung in Java-Klassen nach. Alle Objekte leiten sich von einer Klasse *RObject* ab, die eine Schnittstelle zum Berechnen des Objektausgangs auf vorgegebene Eingangsgrößen besitzt. Alle abgeleiteten Klassen werden über diese Funktion angesprochen. Die Funktionalität darunter ist objektspezifisch.

```
out = RObject.getOutput(input);
```

Die Verschaltung der *RObject* zu Strukturen der Regelungstechnik erfolgt in der Klasse *RTSystems*. Sie nimmt die einzelnen Objekte auf. Die Verschaltung erfolgt zweigweise über dynamische Array-Listen. Unter einem Zweig wird die Reihenschaltung von *RObject*en verstanden, die in einer Senke enden. Durch eine Aufspaltung (Connector) mittels eines Branches entsteht ein neuer Zweig. Die Vereinigung (Merge)

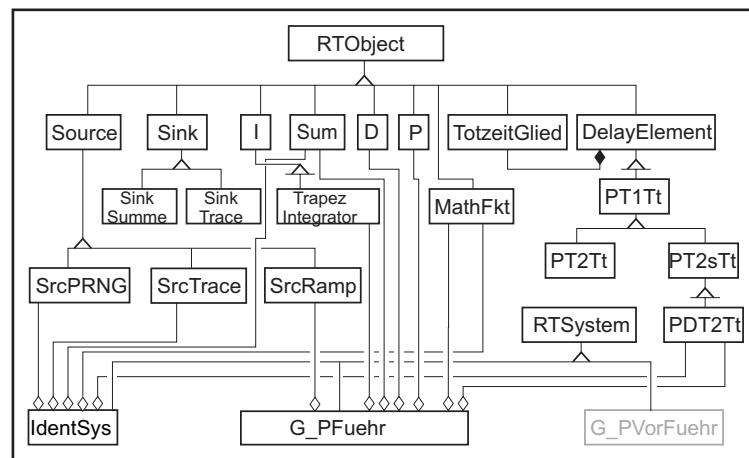


Abbildung 5.31: Auszug der Simulator-Objektzusammenschaltung

zweier Zweige erfolgt hauptsächlich über mathematische Funktionen. Die vorzeichenabhängige Summation der Signale ist hier typisch. Zur Identifikation und Reglerberechnung der beiden Regelkreise wurde jeweils eine Struktur aufgebaut. In Abbildung 5.31 sind die Objektstrukturen in der Identifikation (*IdentSys*) und für das Führungsverhalten des P-Reglers in der Objektzusammenschaltung dargestellt. Abbildung 5.32 zeigt die Umwandlung eines Zweiges in eine Liste.

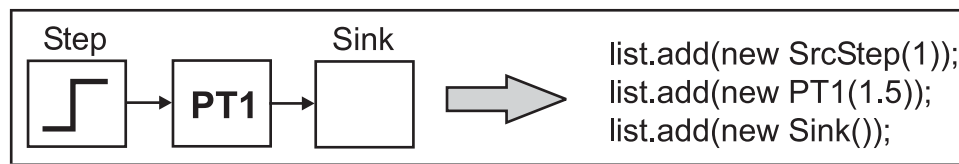


Abbildung 5.32: Umwandlung der Struktur in eine Liste

Der Berechnungsalgorithmus ruft nacheinander jedes Objekt der Liste über die Schnittstelle *getOutput(input)* auf, wobei der Input der Output des vorgeschalteten Objekts ist. Werden die Zweige aufgespalten, so sind beide Zweige bis zum Merge-Knoten zu berechnen. Ein kompletter Durchlauf entspricht einem Simulationstakt.

Die Strukturen sind vom Anwender nicht dynamisch erstellbar, sondern werden ihm vorgegeben. Das vorgestellte Prinzip mit dynamischen Array-Listen ist dafür aber geeignet. Mit einer Erweiterung kann der beschriebene Algorithmus auch für eine dynamische Struktur verwendet werden. Zu ergänzen sind die Module:

**Engineeringoberfläche** – grafische Unterstützung zum Aufbau der Strukturen über Symbole.

**Wandler** – liest die grafische Struktur ein und konvertiert sie in Listen.

### 5.5.1.2 Beaufschlagen der Strecke mit einem Testsignal

Die Struktur der Strecke ist in Abbildung 5.33 im rechten Teil dargestellt. Sie beinhaltet Steller, Motor und Mechanik. Zur Anregung mit einem Testsignal wird sie im Vorwärtszweig vom Lageregler getrennt und nachfolgend über die Variable mit der Testfunktion beaufschlagt.

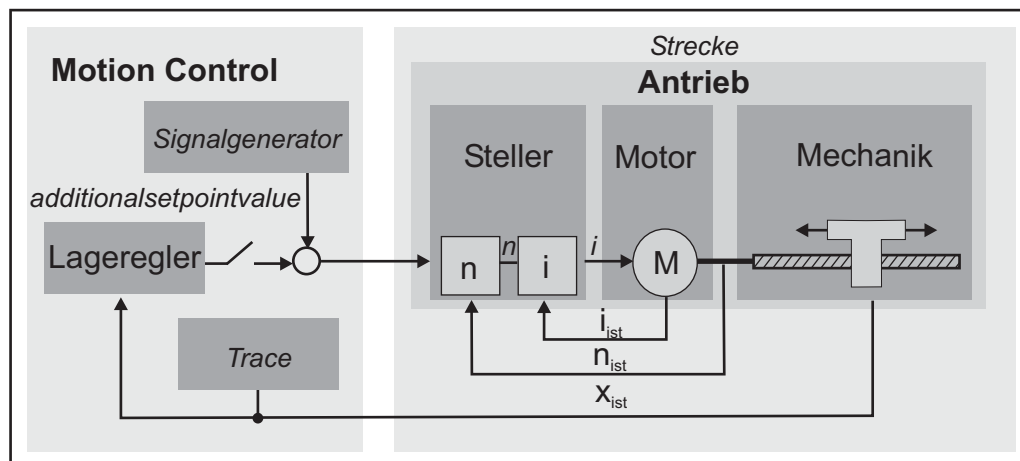


Abbildung 5.33: Anregen der Strecke

Als Testfunktionen werden Treppenfunktionen und PRNS untersucht. Sie sind speziell auf die Regelstrecke abzustimmen. Die Einstellmöglichkeiten der Treppenfunktion unterliegen folgenden Grenzen:

- Stufenanzahl (Begrenzung durch  $v_{max}$ )

- Stufenlänge (min  $2 \cdot T_{\Sigma} + T_t$ ; max speicherabhängig)
- Stufenhöhe (ca. 5–10% von  $v_{max}$ )
- Wiederholungen (keine).

PRN-Signale können ebenfalls über Parameter modifiziert und damit der Strecke angepasst werden:

- Taktzeit muss ein Vielfaches von  $T_a$  sein
- Höhe des Signals ist über Faktoren verstellbar
- Offset.

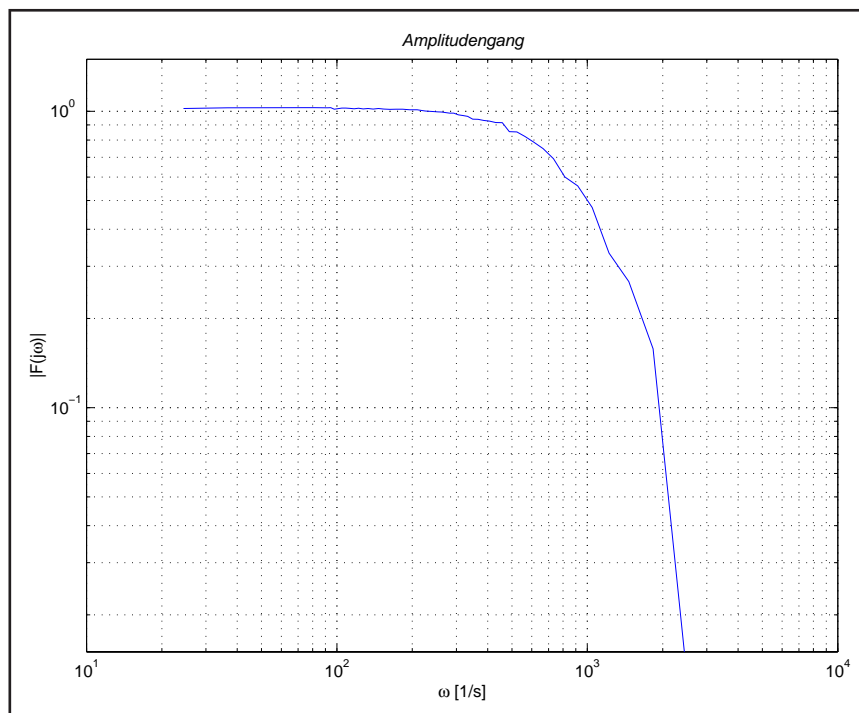
Da der Lageregelkreis in einem geöffneten Zustand ist, sind in dieser Phase Überwachungen und Funktionen wie Softwareendlagenschalter inaktiv. Aus Sicherheitsgründen werden diese im Anwenderprogramm auf der Steuerungsseite nachempfunden. Zusätzlich sind Hardwaresicherungen – Hardwareendlagenschalter – vom Anwender einzusetzen.

Aufgezeichnet werden der Geschwindigkeits- und Positionsverlauf in einem Trace. Wichtig ist, dass Servodatensätze für die Verarbeitung genutzt werden, da vom IPO generierte Variablen nach Abbildung 2.9 einen Totzeittakt mehr aufweisen. Die Traces werden nach Beendigung an den Offline-Teil übertragen und bilden die Grundlage für die weiteren Berechnungen.

### 5.5.1.3 Modellbestimmung

Der nächste Schritt ist die Einordnung des Antriebs in ein regelungstechnisches Modell. Isermann empfiehlt in [32], die experimentelle Identifikation auf der Grundlage theoretischer Betrachtungen zur Regelstrecke aufzubauen. Das Ergebnis soll ein parametrisches Modell sein, dessen Parameter experimentell ermittelt werden.

Für die Inbetriebnahme des Lageregelkreises ist der Drehzahlregelkreis zu identifizieren. Der Amplitudengang eines weitgehend überschwingfrei eingestellten Antriebs zeigt proportionales Verhalten mit Verzögerung höherer Ordnung, wie in Abbildung 5.34 dargestellt.



**Abbildung 5.34:** Amplitudengang eines weitgehend überschwingfrei eingestellten Geschwindigkeitsregelkreises

Üblich ist die Reduzierung auf ein Modell zweiter Ordnung. Die Zeitkonstanten setzen sich aus einer zusammengefassten elektrischen Zeitkonstante und einer mechanischen Zeitkonstante zusammen. Die elektrische Verzögerung hat ihren Ursprung im Antrieb/Steller. Im Vergleich zur mechanischen ist sie in der Regel um ein Vielfaches kleiner. Massenträgheitsmomente von Motor und Mechanikstrang dominieren die Dynamik. Die Erfahrungen zeigen, dass Drehzahlregelkreise nicht überschwingfrei eingestellt werden. Es ist zu untersuchen, ob schwingfähige Modelle besser die Eigenschaften der Strecke widerspiegeln. Es werden  $PT_2$ -Modelle mit und ohne Dämpfung angesetzt. Zur Vereinfachung der Identifikation wurde untersucht, ob die Zusammenfassung der elektrischen und mechanischen Zeitkonstanten sinnvoll ist. Ein weiterer Modellansatz mit einem  $PT_1$ -Modell soll das zeigen. Es ist bekannt, dass einfache Modelle

- die Parametersuche erleichtern
- das Verständnis über das System erhöhen
- die Rechenzeit verkürzen
- die Fehlerwahrscheinlichkeit reduzieren.

Demgegenüber steht der Modellfehler, der sich in den berechneten Reglerparametern niederschlägt. Im Gegenzug soll ein komplizierteres Modell untersucht werden. Antriebe werden oft nach dem Betragsoptimum eingestellt. Das Betragsoptimum ergibt eine 3–4%-Überschwingweite. Bei dem schwingfähigen  $PT_2$ -Modell ist  $D$  ein Maß für die Dämpfung und die Zeitkonstante  $T$  ein Maß

der Periodendauer einer Schwingung. Wird ein D-Anteil in das Modell eingefügt, steht ein weiterer Freiheitsgrad zur Anpassung des Modells an das reale Verhalten des Antriebs zur Verfügung.

In den Arbeiten zur Simulation von Antrieben [6, 27] wurde gezeigt, dass aufwändigere Modelle nicht immer eine bessere Annäherung an das reale Systemverhalten als Folge haben. In [40] wird diese Aussagen unterstützt.

Zusammengefasst ergeben sich die folgenden Eigenschaften:

- Verzögerung höherer Ordnung
- Dämpfung  $D$  kleiner/gleich eins
- D-Anteil des Drehzahlregelkreises
- Totzeit  $T_t$  aus Task- und Kommunikationssystem.

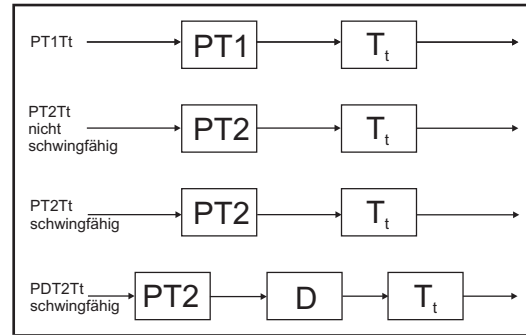


Abbildung 5.35: Übersicht der verwendeten Modelle

Für das weitere Vorgehen werden die Modelle nach Abbildung 5.35 angesetzt. Als Modellparameter stehen die Zeitkonstante  $T$ , die Dämpfung  $D$  und die Vorhaltzeit  $T_V$  zur Berechnung aus. In der Prüfung der Normierung wird ein Übertragungsfaktor  $K$  der Strecke von eins festgestellt.

#### 5.5.1.4 Parameterberechnung

Die Bestimmung der Parameter des Modells erfolgt auf der Grundlage des aufgezeichneten Geschwindigkeitsverlaufs. In Abbildung 5.36 ist links die Systemantwort auf eine Treppenfunktion dargestellt. Das rechte Diagramm stellt die normierte Sprungantwort dar, die nach der Mittelung aus der Treppe extrahiert wird.

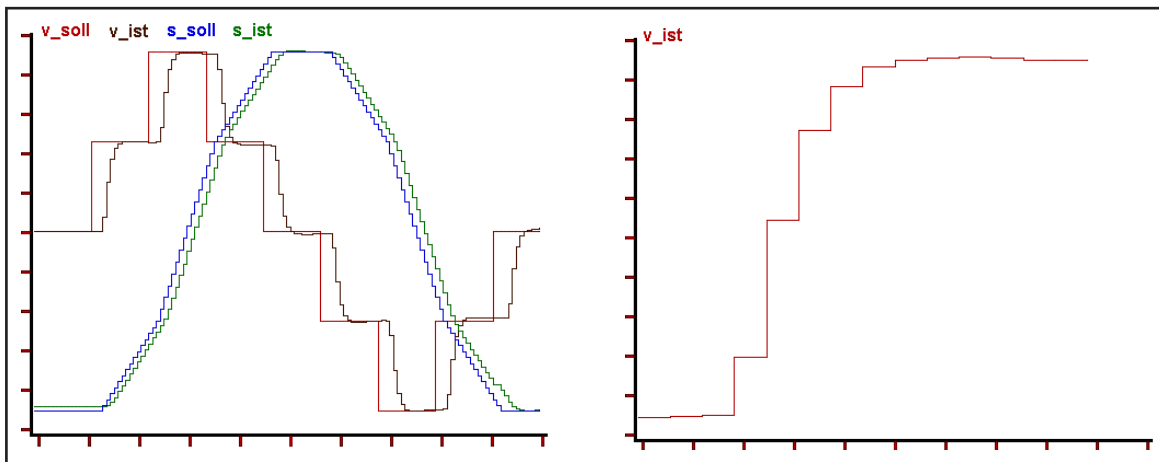


Abbildung 5.36: Trace des Antriebs auf eine Treppenfunktion – Taktzeit  $T_a = 1,5 \text{ ms}$

Die Parameterberechnung wird nach dem Prinzip der Simulation gelöst. Die in einem Array aufgezeichnete Sprungantwort wird mit der simulierten Modellantwort verglichen. Je nach



Übereinstimmung (Wert der Zielfunktion) werden die Modellparameter geändert. Die Zielfunktion berechnet den quadratischen Fehler zwischen Modell und Strecke. Die Abbildung 5.37 stellt den Signalflussplan für die entworfene Simulationsumgebung nach Kapitel 5.5.1.1 und das Prinzip der Identifikation mittels Simulation nach Abbildung 3.15 dar. Im rechten Teil ist die Umsetzung in Java notiert.

In der Datensenke, gekennzeichnet durch das Summenzeichen, wird der quadratische Fehler in jedem Takt aufsummiert. Er geht in den anschließenden Optimierungsprozess ein.

Vor der Wahl der Strategie zur Optimierung wird die Oberfläche der Zielfunktion untersucht. Um eine Visualisierung im dreidimensionalen Raum zu ermöglichen wird ein

$PT_1T_t$ -Glied mit den freien Parametern  $T$  und  $T_t$  als Vergleichsmodell genutzt. In der Abbildung 5.38 spannen die freien Parameter die x-y-Ebene auf, die Zielfunktion belegt die dritte Dimension.

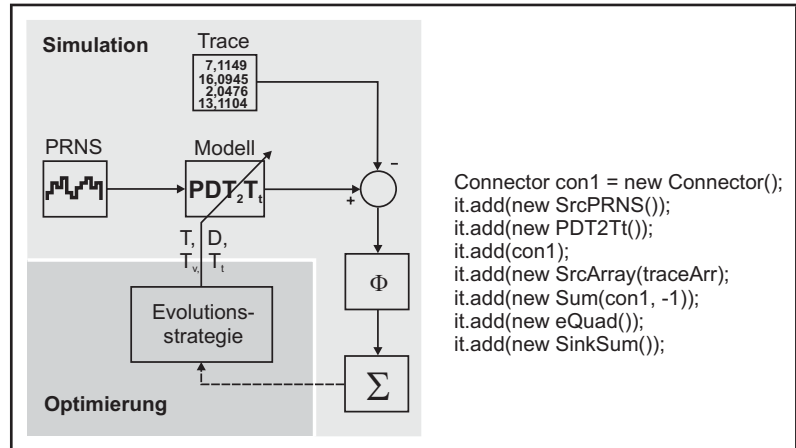


Abbildung 5.37: Identifikation mittels Simulation in Java

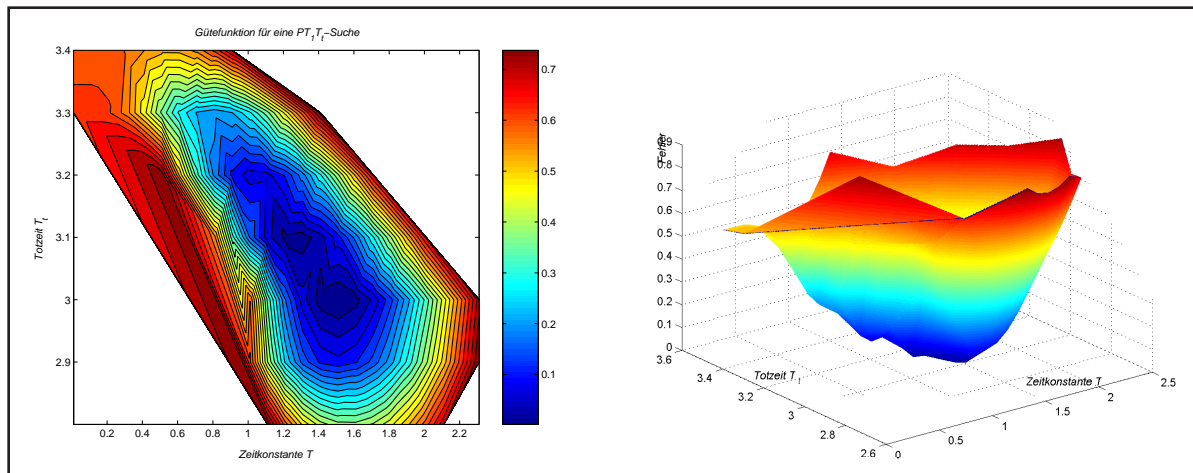


Abbildung 5.38: Zielfunktion bei der simulierten Suche der Parameter eines  $PT_1T_t$ -Elements

Deutlich sind lokale Minima sowie Schluchten zu erkennen. Nach den Kriterien zur Wahl der Optimierungsstrategie (Kapitel 3.1.2.4, S. 44 ff.) scheiden die Verfahren Rastersuche und Gradientenverfahren aus. Gewählt wird die auf Seite 47 beschriebene Evolutionstrategie.

### Erweiterung der Evolutionstrategie

Für die  $(\mu, \lambda)$ -Strategie wird die Anzahl der generierten Nachkommen mit  $\lambda = 20$  und die daraus gewählten besten Nachkommen mit  $\mu = 5$  festgesetzt. Zur Bewertung der angewendeten Evolutionstrategie sind Diagramme zur Konvergenz und Artenvielfalt in der Identifikation mit einem  $PT_2T_t$ -

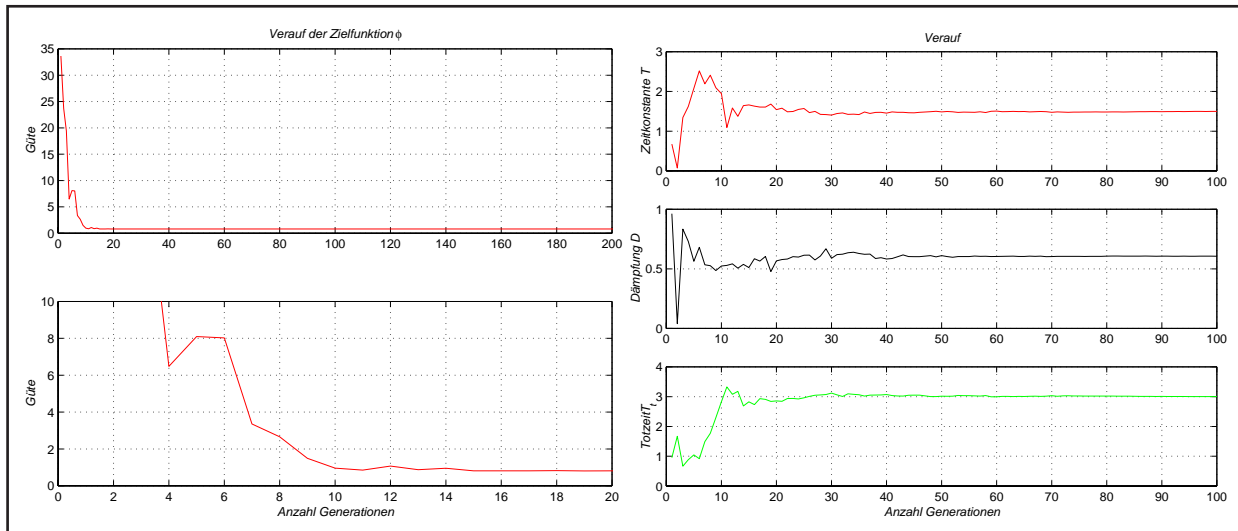


Abbildung 5.39: Konvergenz von Parametern und Zielfunktion

Modell hinterlegt. Die Abbildung 5.39 rechts zeigt, dass bereits nach ca. 20 Schritten gute Werte der Zielfunktion erreicht werden. Die gesuchten Parameter in Abbildung 5.39 links stabilisieren sich nach weiteren 20 Generationen.

Die genaue Betrachtung der Zielfunktion zeigt jedoch ein Ausschlagen nach oben bei Generation sechs. An dieser Stelle haben die Nachkommen eine schlechtere Zielfunktion als die Eltern. Ein solches Verhalten kann nötig sein, um eine Senke zu verlassen, es kann aber auch zum "Verlaufen" führen. In Experimenten ist das oft zu beobachten. Im ungünstigsten Fall sind die Ergebnisse nicht verwendbar. Um das zu verhindern, wird die in 3.1.2.4 beschriebene Strategie modifiziert, indem die Lebensdauer der Eltern nicht nur auf eine Generation beschränkt wird. Im Gegenzug wird deren Vitalfunktion von Generation zu Generation gealtert. Die Alterungsrate in Abbildung 5.40 (links) wird auf 1% festgesetzt.

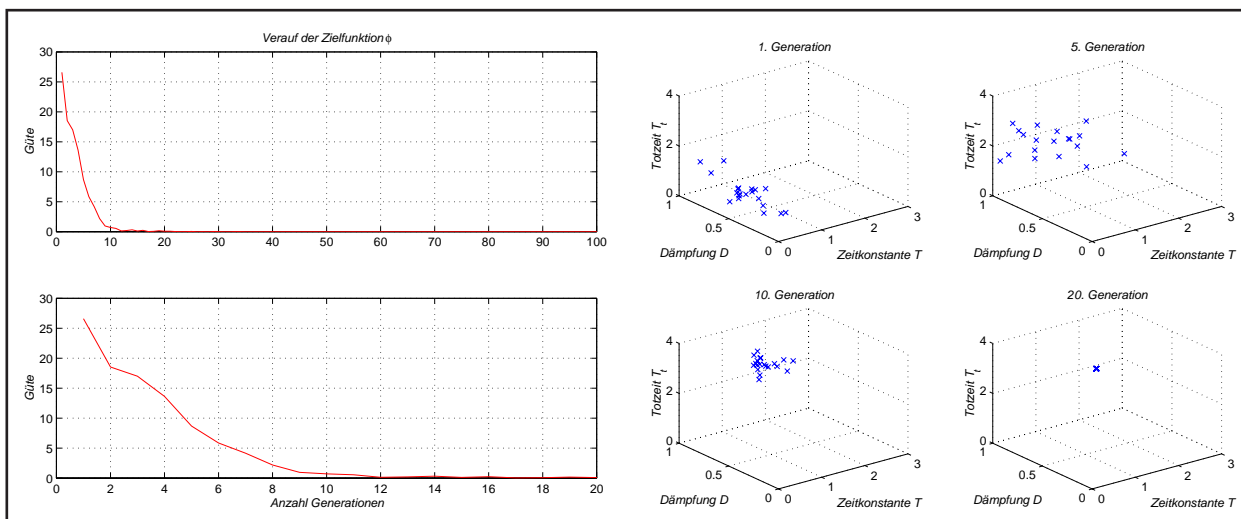


Abbildung 5.40: Konvergenz (links) und Entwicklung der Artenvielfalt (rechts)

Die Entwicklung der Artenvielfalt ist ebenfalls ein Indikator für die Güte der Optimierungsstrategie. Wie die Abbildung 5.40 zeigt, wird die Forderung nach einer breiten Verteilung im Raum zu Beginn der Suche und einer geringen Verteilung bei Erreichen des Optimums erfüllt.

### 5.5.1.5 Ergebnisse

In diesem Abschnitt werden nur Ergebnisse zum Einfluss der Testfunktionen und der Vergleichsmodele auf den Identifikationsverlauf betrachtet. Die letztendlich entscheidenden Auswirkungen der Ergebnisse auf die Reglerparameter werden im Kapitel 5.5.2 ausgewertet.

Bei Vorgabe zufälliger Startwerte aller Parameter im Bereich von null und eins steigt die Wahrscheinlichkeit des "Verlaufens" mit der Zunahme der Anzahl von Parametern des Vergleichsmodells – trotz der Erweiterung – an. Zu beobachten ist das bereits beim  $PT_2T_t$ -Modell. Als Grund sind die zufälligen Startwerte der Simulation zu nennen. Tests mit  $PT_1T_t$ -Modellen stellen sich als robust heraus. Der Modellfehler im Vergleich zu umfangreicheren Modellen ist jedoch groß. Aus dieser Feststellung folgt, dass eine erste Identifikation mittels  $PT_1T_t$ -Modell sinnvoll ist. Diese ermittelten Parameter dienen in einem weiteren Schritt als Startwerte für kompliziertere Modelle. Ein "Verlaufen" wird danach nicht mehr festgestellt. Die schrittweise Erhöhung der Modellordnung im Iterationsprozess ist erfolgreich.

Die Identifikationsergebnisse bei einer Anregung mit der Treppenfunktion in Tabelle 5.5 und bei einer Anregung mit einem PRNS in Tabelle 5.6 zeigen eine Totzeit von  $3,6 \cdot T_a$ . Zu erkennen ist, dass bei beiden Anregungsformen die Gütewerte, mit denen die Übereinstimmung der Kurvenverläufe zwischen Antrieb und Modell bewertet sind, mit zunehmender Komplexität des Modells besser werden. Dieses Ergebnis wird auch in den Abbildungen 5.41 und 5.42 bestätigt.

	$PT_1T_t$	$PT_2T_t$	$PT_2sT_t$	$PDT_2T_t$
$T_1$	1,30 ms	0,74 ms	1,50 ms	2,80 ms
$T_2$	-	0,75 ms	-	-
$D$	-	-	0,72	0,99
$T_v$	-	-	-	4,5 ms
$T_t$	$3,6 T_a$	$3,6 T_a$	$3,6 T_a$	$3,6 T_a$
Güte	0,0124	0,0108	0,0080	0,0070

*Tabelle 5.5: Ergebnisse der Identifikation mit Treppenfunktion*

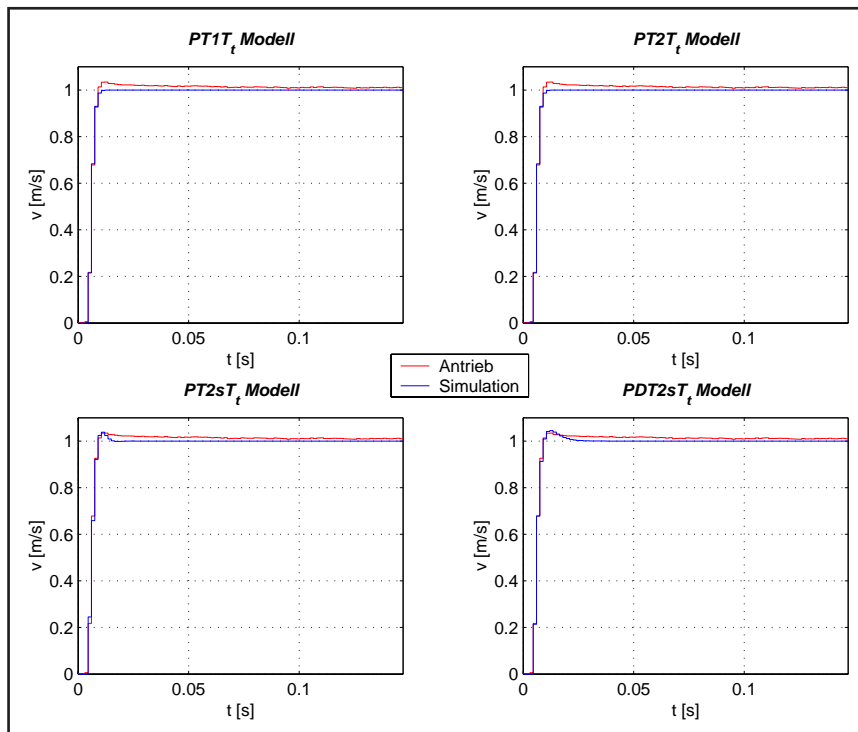


Abbildung 5.41: Ergebnis der Identifikation mit unterschiedlichen Modellen auf Basis der Treppenfunktion

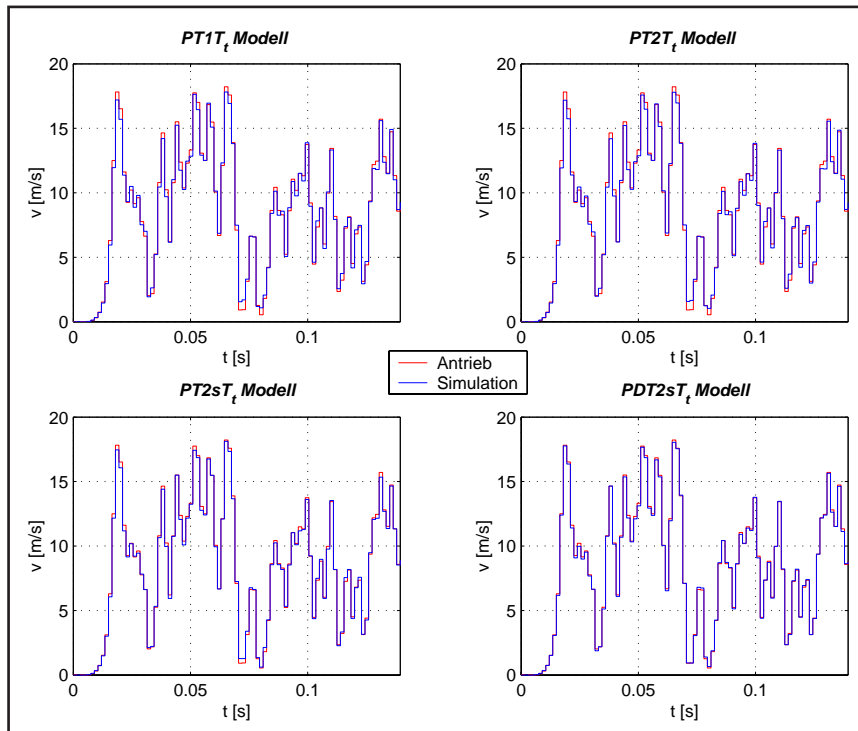


Abbildung 5.42: Ergebnis der Identifikation mit unterschiedlichen Modellen auf Basis von PRNS

	$PT_1T_t$	$PT_2T_t$	$PT_2sT_t$	$PDT_2T_t$
$T_1$	1,50 ms	0,84 ms	0,78 ms	1,70 ms
$T_2$	-	0,74 ms	-	-
$D$	-	-	0,99	0,83
$T_v$	-	-	-	1,5 ms
$T_t$	$3,6 T_a$	$3,6 T_a$	$3,6 T_a$	$3,6 T_a$
<b>Güte</b>	14	4,8	4,6	1,4

*Tabelle 5.6: Ergebnis der Identifikation mit PRNS-Signal*

Ein Vergleich der ermittelten Zeitkonstanten zwischen den Modellen ist nicht trivial. Das beste Modell wird nach den Ergebnissen der Reglerberechnung ausgewählt. Festzustellen ist schon an dieser Stelle, dass die Ergebnisse der  $PT_1$ -Modelle nah an denen des Online-Verfahrens liegen.

## 5.5.2 Reglerberechnung

Zur Minimierung des Programmieraufwands wird die Berechnung der Reglerverstärkung  $K_P$  ebenfalls über die Simulation durchgeführt.

### 5.5.2.1 P-Regler

Die Berechnung von Startwerten, zum Beispiel nach Samal [56]

$$K_P = \frac{\pi}{4} \cdot \frac{1}{K_S \cdot (T_t + T_u)}, \quad (5.5)$$

stellt sich als nicht notwendig heraus. Der Suchalgorithmus ist für einen Parameter stabil und schnell.

Die Struktur während der Simulation ist in Abbildung 5.43 dargestellt. Sie enthält den Regelkreis und die Funktionen zur Berechnung der Güte des Regelverhaltens. Der Regelkreis wurde der SIMOTION-Beschreibung in Abbildung 2.6 nachempfunden. Diese Darstellung ist bereits stark vereinfacht. Der reale Regelkreis enthält eine Vielzahl weiterer Funktionen, die das Regelverhalten verbessern und die Stabilität erhöhen. Sie sind verteilt zwischen Steuerung und Motor. Welche Funktionen das sind und wie deren Funktionsweise ist, ist für die Öffentlichkeit unbekannt und gehört zum Know-how des Herstellers.

Zum Erreichen des gewünschten Übertragungsverhaltens werden die folgenden Gütekriterien genutzt:

- Integralkriterium der Zeitquadratgewichteten Quadratischen Regelfläche des Geschwindigkeits-Istwertes und -Sollwertes
- Forderung nach maximaler Rückführverstärkung
- Beschränkung der Überschwingweite  $h_{\ddot{u}}$  durch eine Schranke nach Bretschneider
- (oder) Forderung nach minimalem Schleppabstand.

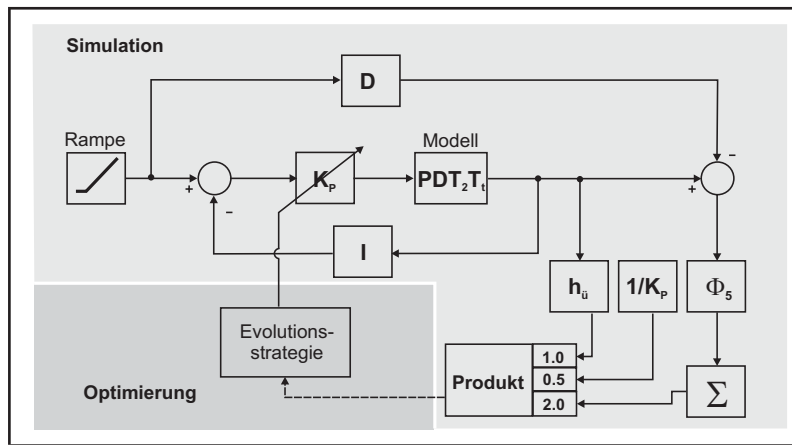


Abbildung 5.43: Struktur zur Berechnung des  $K_P$ -Faktors

Jedes Gütekriterium kann extra gewichtet werden. Die Werte der Gütefunktionen werden am Ende eines Simulationsdurchlaufes nach Gleichung (3.63) für jeden Parametersatz zusammengefasst. Ziel der Simulation ist ein minimaler Gütewert.

Der Evolutionsalgorithmus zur Optimierung ist der gleiche wie in der Identifikation. Es ist keine Umstellung der Struktur oder Arbeitsweise nötig.

In der Tabelle werden die ermittelten Verstärkungsfaktoren für unterschiedliche Überschwingweiten mit dem realen Wert des Regelkreises verglichen. Die linke Spalte enthält jeweils den Wert des Modells, welches mit der Treppenfunktion ermittelt ist, die rechte Spalte den des mit PRNS identifizierten Modells.

$h_{\ddot{u}}$	$PT_1T_t$		$PT_2T_t$		$PT_2sT_t$		$PDT_2T_t$		Real P
1 %	73	73	56	56	63	58	62	61	78
3 %	81	81	62	62	69	64	66	66	85
7 %	92	92	71	71	77	73	74	74	95
10 %	98	98	77	77	83	79	78	80	102
15 %	109	109	86	86	91	89	87	88	113
20 %	119	119	94	94	100	97	95	97	125

Tabelle 5.7: Ermittelte Verstärkungsfaktoren  $K_P$  in  $\frac{1}{s}$  für gegebene Überschwingweiten  $h_{\ddot{u}}$   
 Die linke Spalte gibt den Wert des mit dem Treppensignal identifizierten Modells an, die rechte Spalte den PRNS-Wert

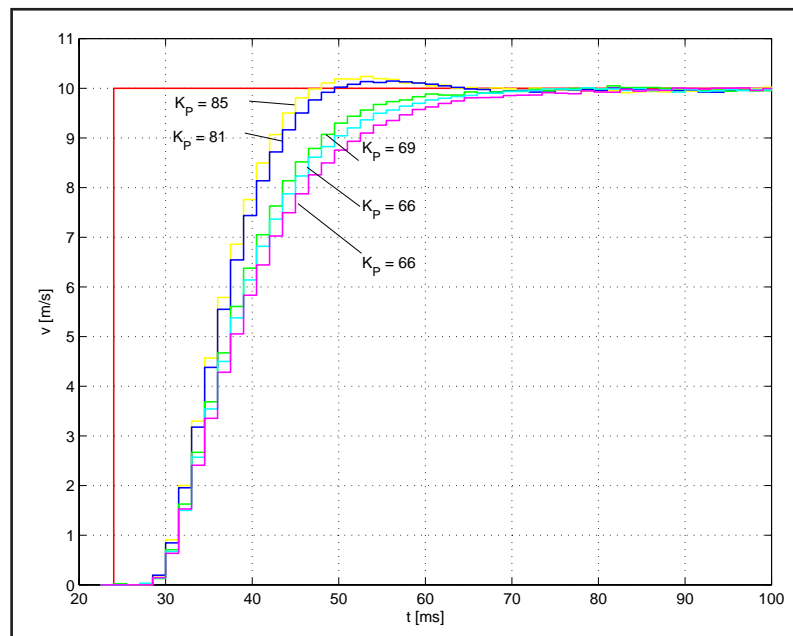
Zu erkennen ist, dass die ermittelten  $K_P$ -Werte etwas kleiner sind als die des realen Regelkreises. Die Auswirkung des Identifikationssignals auf die Reglerverstärkung ist als gering einzuschätzen. Auffällig ist, dass das Regelkreismodell mit dem einfachen  $PT_1$ -Modell in den Experimenten näher am Verhalten des Regelkreises ist als kompliziertere Modelle. Da die Optimierung/Parametersuche mit einfachen Modellen aufgrund der geringeren Parameteranzahl robuster und schneller ist, wird das  $PT_1$ -Modell favorisiert.

Die Tabelle 5.7 zeigt den sehr geringen Einfluss der Anregungssignale in der Identifikation auf die berechnete Reglerverstärkung. Die Vorteile der Treppenfunktion gegenüber PRNS sind

- die Mittelung der Werte
- der kleine, einfach berechenbare Verfahrensweg
- der Zeitvorteil durch Reduzierung auf einen Sprung in der Simulation.

Aufgrund dieser Vorteile wird die Treppenfunktion favorisiert.

In der automatischen Reglerinbetriebnahme erfolgt anschließend ein Prüfen der ermittelten Werte auf der Steuerung (Abbildung 5.44). Die gewählte Reglerstruktur wird in der Steuerung implementiert. Die Validierung erfolgt mit rampenförmigem Sollwertverlauf. Das Verhalten des Regelkreises kann wieder gut am Geschwindigkeitsverlauf ermittelt werden. In dieser Phase ist die Schwingungsüberwachung des Online-Verfahrens aktiv. Entsprechen die geforderten Eigenschaften in Grenzen denen des Lagereglers, so ist die Reglerberechnung erfolgreich. Die ermittelten Parameter werden dem Anwender bereitgestellt.

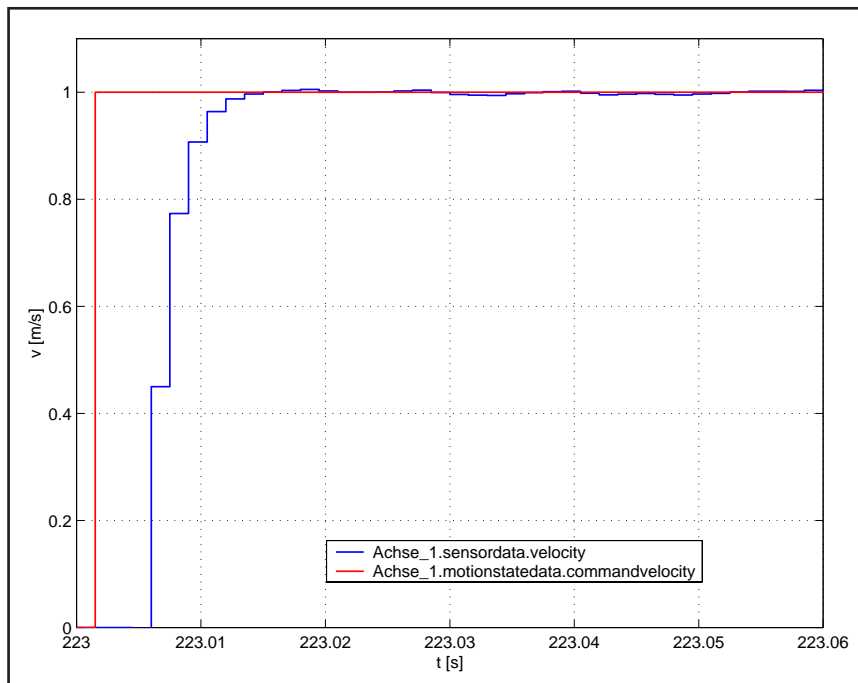


**Abbildung 5.44: Validierung der Ergebnisse in der P-Reglerberechnung**

### 5.5.2.2 PV-Regler mit Symmetrierfilter

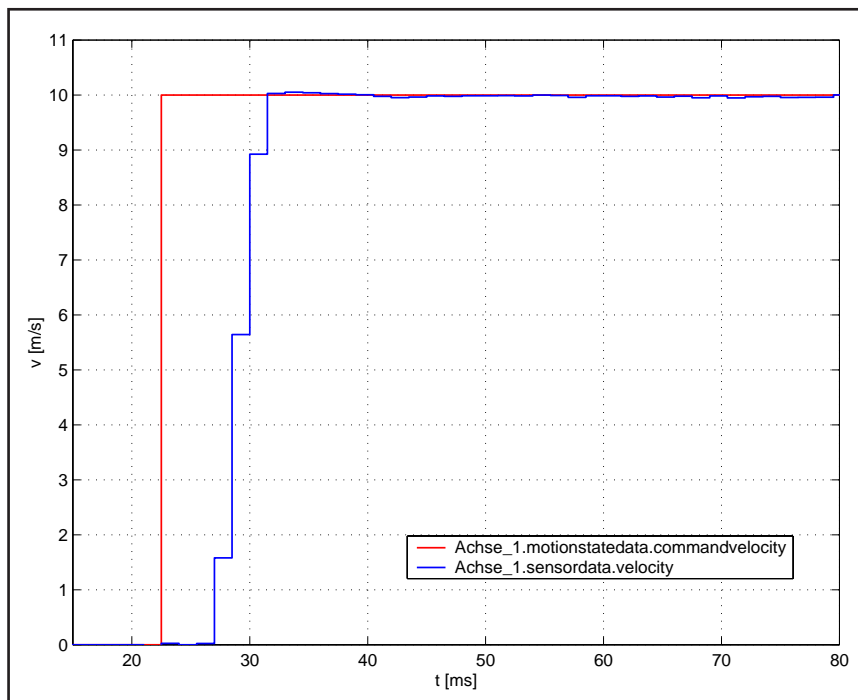
Auch das Offline-Verfahren kann den Regelkreis mit Vorsteuerung parametrieren.

Der PV-Regler wird mit der Reglerverstärkung  $K_{PV} = K_P$  und der erhaltenen Zeitkonstante aus der Identifikation mit dem  $PT_1$ -Modell für den Symmetrierfilter in der Steuerung implementiert. Ist das



**Abbildung 5.45: P-Regler mit Vorsteuerung nach der Offline-Inbetriebnahme**

resultierende Regelverhalten überschwingfrei, so ist die Identifikation erfolgreich. Abbildung 5.45 zeigt das nach Anlegen einer Rampe an den Sollwertzweig des Lagereglers.



**Abbildung 5.46: P-Regler mit Vorsteuerung und DSC nach der Offline-Inbetriebnahme**

Um das Regelverhalten zu verbessern wird im Anschluss DSC zugeschaltet. Wie Abbildung 5.46



zeigt, ist der Lageregler daraufhin schneller.

## 5.6 Sicherheitsmechanismen

Sicherheitmechanismen dienen dem Schutz von Mensch und Maschine. Der vorgestellte Service unterscheidet sich zu bestehenden Diensten darin, dass er aktiv die Maschine bedient. Da dies prinzipiell keine Anwesenheit von Personal vor Ort erfordert, sind Warnungen an der Maschine optisch oder akustisch anzuzeigen.

Gefahren können in der Identifikationsphase und in der Validierung der Reglerparameter auftreten. Während der Identifikation ist der Lageregler geöffnet. Sämtliche Überwachungen (Softwareendschalter, Schleppabstand) sind nicht aktiv. Um sicherzustellen, dass die zur Verfügung stehenden Wege nicht überschritten werden, sind extra Überwachungen zu integrieren. Da diese die Forderung nach Echtzeit stellen, werden sie auf der Steuerung in der IPO-Synchronous-Task integriert. Hinweise zur Sicherheit in der Validierungsphase werden in [72] erteilt.

Rechenfehler können die Maschine bei zu groß gewählten Verstärkungsfaktoren in der Validierung zum Schwingen bringen. Ein Aufschwingen der Maschine ist möglich. Abhilfe schafft eine Geschwindigkeits-Istwert-Überwachung, die im Falle des Überschreitens der vorgegebenen Geschwindigkeit über eine Marke eine Not-Stop-Strategie aufruft.

## 5.7 Bewertung der Verfahren

Zur Bewertung der Verfahren werden die Genauigkeit und Robustheit, die Einsatzmöglichkeiten in der automatischen Inbetriebnahme sowie Zeitfaktoren und die Sicherheit in der Durchführung als objektive Kriterien herangezogen. Die Anwenderakzeptanz fließt als subjektive Größe mit ein.

Die Hauptkriterien zur Bewertung stellen die Genauigkeit und Robustheit des Verfahrens dar. Die Online-Methode stellt den  $K_P$ -Faktor direkt auf der Steuerung ein. Die Genauigkeit ist hier recht hoch. Die Berechnung der Ersatzzeit hingegen beruht einzig auf einem Punkt  $P_m$ . Ist dieser durch Rauschen fehlerhaft, so wirkt sich das direkt auf die Ergebnisse aus. Die Tabelle 5.4 zeigt, dass die Überschwingweite einen Einfluss auf das Ergebnis hat. In der Identifikation über Simulation wird dagegen der gesamte Kurvenzug bewertet, einzelne Störungen fallen nicht derartig ins Gewicht. Hier steht die Güte der Modelle im Vordergrund.

Ein weiteres wichtiges Kriterium bei der Auswahl des Verfahrens ist die Sicherheit während der Inbetriebnahme. Das Online-Verfahren wird nur im geschlossenen Regelkreis betrieben. Das heißt, die Überwachungen der Softwareendlagen, die Positionsüberwachung und auch die Stellgrößenüberwachung sind aktiv. Gegen den Einsatz der heuristischen Suche spricht die Schätzung des Startwertes der Reglerverstärkung im ersten Durchlauf. Wird sie zu groß gewählt, kann eine Beschädigung des Antriebsstrangs nicht ausgeschlossen werden. Im Offline-Verfahren wird der La-

geregler dagegen geöffnet, was ein Abschalten der Überwachungen zur Folge hat. Aus diesem Grund sind im Dienst eigene Überwachungen implementiert, die die Sicherheit der Maschine gewährleisten. Der vom Verfahren benötigte Weg kann ein Ausschlusskriterium beim Einsatz sein. Er ist durch den Verfahrbereich der Achse begrenzt. In der Online-Methode muss die Verfahrbewegung nur so weit durchgeführt werden, bis die Höhe des ersten Überschwinges  $P_m$  bekannt ist. Der benötigte Verfahrweg ist sehr stark vom Verstärkungsfaktor des P-Reglers abhängig. Bei zu kleinen Verstärkungsfaktoren wird dieser Punkt erst sehr spät oder auch gar nicht erreicht. In diesem speziellen Fall ist der Wegbedarf höher als beim Offline-Verfahren. Generell ist der geschlossene P-Regelkreis langsamer als die Regelstrecke. Da im Offline-Verfahren aber nicht nur der Punkt  $P_m$  für die Berechnung genutzt, sondern der gesamte Einschwingvorgang betrachtet wird, ist die Aufzeichnung über  $P_m$  hinaus notwendig und die zurückgelegte Wegstrecke oft länger. In den durchgeführten Experimenten stand der benötigte Verfahrweg allerdings stets zur Verfügung.

Die zeitliche Dauer der Reglerberechnung ist bei der Inbetriebnahme der Maschine möglichst klein zu halten und ein letztes Kriterium bei der Auswahl eines Verfahrens. Im Online-Verfahren muss die Maschine für die gesamte  $K_P$ -Suche und  $T_{Filter}$ -Berechnung zur Verfügung stehen. Im Offline-Verfahren wird dagegen nur einmal ein Trace aufgenommen. Die Parameterberechnung erfolgt anschließend auf einem separaten Rechner. In dieser Zeit steht die Maschine für weitere Aufgaben zur Verfügung. Zudem spart die Auswertung des Hin- und Rückwegs Zeit, da für eine Wegstrecke zwei Experimente durchgeführt werden. Außerdem werden auftretende Richtungseffekte gemittelt.

Nicht zuletzt ist die Akzeptanz des Inbetriebnahmetools beim Anwender wichtig für dessen Erfolg. Der Anwender beurteilt hauptsächlich nach den oben genannten Kriterien. Oft wird nicht so stark nach den verwendeten Algorithmen entschieden, sondern danach, wie sich ein Tool in eine Toolkette einreicht. Gewünscht wird beispielsweise die Weiterverwendung von ermittelten Daten, ohne diese von Hand in ein weiteres Programm übertragen zu müssen. Solche Daten der Reglerinbetriebnahme sind die Modell- und Reglerparameter. Wünschenswert ist eine Einbeziehung dieser in den Prozess der Bewegungsplanung der sich anschließenden Phase der Programmierung. Mit einer geeigneten Erweiterung der vorhandenen Simulationsumgebung der Offline-Reglerinbetriebnahme auch mit Interpolations- und Kurvenscheibenfunktionalität steht ein Tool zur Simulation der Bewegungsabläufe von Vielachssystemen zur Verfügung. Hier wird vor allem die Optimierung der Bewegungsführung im Achsverbund als weitere Aufgabe anstehen.

Die Bewertung nach den fünf Kriterien legt die Verwendung des Offline-Verfahrens nahe.

# Kapitel 6

## Zusammenfassung und Ausblick

Mechatronische Systeme werden mit einer steigenden Anzahl an Achsen ausgeführt. In Produktionsmaschinen sind zwanzig, fünfzig oder hundert Achsen keine Seltenheit. Auf mehrere hundert Achsen in verschiedenen Gleichlaufverbänden wird man sich einstellen müssen. Die Beherrschung von Maschinen und Anlagen mit einer Kompliziertheit aus derartigen Multiachsverbänden setzt geeignete Inbetriebnahmestrategien voraus. Zur Inbetriebnahme der einzelnen Lageregelkreise sind Verfahren bekannt. In der Arbeit werden die Verfahren nach den definierten Anforderungen ausgewählt und für eine automatisierte Inbetriebnahme aufbereitet. Die Ausführung erfolgt als Ferninbetriebnahme über das Internet. Der Inbetriebnahmeprozess beschränkt sich auf die in der Steuerung vorgegebenen Regelkreisstrukturen.

Mit dem Einzug von Ethernet in die Steuerungsebene und der damit möglichen Anbindung an das Internet ergeben sich vielfältige Anwendungen auf Basis von TCP/IP. Im Fokus stehen internetbasierte Dienstleistungen. Für Steuerungs- und Maschinenhersteller bieten diese die Möglichkeit, ihr Know-how in Rechnern abzulegen und dem Anwender rund um die Uhr zur Verfügung zu stellen. Der industrielle Einsatz von Dienstleistungen hat sich an naheliegenden Aufgaben durchgesetzt. Zu nennen sind hier vor allem Visualisierungs-, Diagnose- und Programmierleistungen.

Im Rahmen dieser Arbeit wird eine komplizierte internetbasierte Dienstleistung untersucht. Gegenstand ist die Inbetriebnahme von Lagereglern für Servoantriebe an mechatronischen Produktionsmaschinen. Als Zielsystem wird die Bewegungssteuerung SIMOTION der Siemens AG genutzt.

Das Vorgehen entspricht einer Offline-Inbetriebnahme, die auch Online-Komponenten enthält. Die Verbindung beider Komponenten erfolgt über einen Vermittler im Internet, dem Portal. Das so entstehende Kommunikationsdreieck bietet einige Vorteile gegenüber einer direkten Kommunikation:

- Die Kommunikation in die sensible Automatisierungsebene erfolgt ausschließlich über das Portal.
- Das Portal nutzt die Funktionalität des bestehenden Engineeringssystems und geht damit einen weiteren Schritt in Richtung offener, dezentraler Automatisierungssysteme.

- Die Zuschaltung eines Experten zum Inbetriebnahmevorgang ist nach einer noch erforderlichen Software-Erweiterung möglich.
- Die Auslagerung von Rechenzeit auf den Anwender-PC, um Ressourcen einzusparen.

Besondere Beachtung findet die Sicherheit von Daten sowie von Mensch und Maschine während des Inbetriebnahmevorgangs. Zur Datensicherheit stehen Funktionen wie Firewall, Authentifizierung und Datenverschlüsselung zur Verfügung. Richtlinien zu Sicherheitsfunktionen bei der Fernsteuerung von Maschinen sind in der Literatur und in Normen bisher nur in eingeschränktem Maße betrachtet worden.

Neben Recherchen und Untersuchungen zur Theorie wurden die Ergebnisse an einem prototypischen, verteilten Steuerungssystem nachgewiesen. Die Reglerinbetriebnahme ist nun auch ohne Anwesenheit des Inbetriebnehmers vor Ort technisch weltweit möglich. Die Funktionalität ist unter labormäßigen Bedingungen demonstriert worden. Es zeigt sich, dass

- Eine sichere Kommunikation über vorgestellte Strukturen durchführbar und sinnvoll ist.
- Die Reglerinbetriebnahme ohne lokal installiertes Engineeringssystem ausgeführt werden kann.
- Die Simulation im Zusammenhang mit einem Optimierungsverfahren ein Verfahren zur Identifikation von Strecken mit unbekanntem Totzeiten darstellt.

Für eine industrielle Breitenutzung sind weitere Aspekte aufzubereiten. Dazu gehören einerseits Inbetriebnahmestrategien für eine Vielzahl von dezentralen Regelkreisen oder Antrieben und andererseits Sicherheitsvorkehrungen, die beim Zugriff auf Maschinen ohne Sichtkontakt einzuhalten sind. Die Arbeit gibt darüber ausgewählte Anregungen. Der Vorteil des Anwenders, kein Engineeringssystem vor Ort zu benötigen, wird mit einem hohen Rechenaufwand im Portal bezahlt. Die maximale gleichzeitige Nutzerzahl ist dadurch nach oben stets beschränkt. Sie kann aber über technische Erweiterungen gesteigert werden.

# Literaturverzeichnis

- [1] Intel AG. A perspective on web services. <http://www.mywebservices.org/index.php/article/articleprint/113/-1/24/>, 2002.
- [2] Siemens AG. eps-network services. *Online: www.siemens.de/simatic*, 11/2004.
- [3] K. Backhaus. *Die Zahlungsbereitschaft des Kunden für produktbegleitende Dienstleistungen – Ergebnisse einer Kundenbefragung*. VDMA Verlag Frankfurt, 1999.
- [4] Oliver Baumann. Die neue OPC-Freiheit. *Computer & Automation*, September 2005.
- [5] S. Bocklisch. *Script "Kennwertermittlung linearer Glieder mit determinierten Testsignalen"*. Script, 1993.
- [6] Rene Böhm. *Beitrag zur Simulation von mechatronischen Produktionsmaschinen mit Mehrkörpermodellen*. Diplomarbeit, TU Chemnitz, 2004.
- [7] Claudia Brand. *Neuronale Identifikation von Totzeiten*. Dissertation, TU München, 2002.
- [8] J. Bretschneider. Automatisierte Inbetriebnahme elektromechanischer Vorschubsysteme. In *Stuttgarter Impulse. Innovation durch Technik und Organisation*. Fertigungstechnisches Kolloquium, Axel-Springer-Verlag, 1997.
- [9] Stephan Bürkner. Internetbasierter Service im Lebenszyklus komplexer Produkte. In *Fortschritt-Berichte VDI*. VDI-Verlag, 2001.
- [10] EBC Consulting. *E-Services in der Investitionsgüterindustrie, Eine Befragung von Anbietern und Anwendern*. VDI Verlag Düsseldorf, 2001.
- [11] Cay S. Horstmann; Gary Cornell. *core JAVA 2 – Band 1 - Grundlagen*. Markt und Technik, 2003.
- [12] Cay S. Horstmann; Gary Cornell. *core JAVA 2 – Band 2 - Expertenwissen*. Markt und Technik, 2003.
- [13] Deutsche Elektronische Kommission im DIN und VDE (DKE). *DIN 19226 – Regelungstechnik und Steuerungstechnik*, 2 1994.

- [14] Philipp Drum. *Effiziente Methoden für global verteiltes wissenschaftliches Rechnen*. Dissertation, TU München, 2001.
- [15] R. Berger E. Hohwieler, G. Schreck. Bereitstellung elektronischer Dienstleistungen für Produktionssysteme. *Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik*, 2002.
- [16] Michael Friedgen. *Optimierung der Bahn und des Zeitpunktes des Wiedereintritts geflügelter Raumtransporter mit Gradientenverfahren und Evolutionsstrategien*. Wissenschaftsverlag Aachen, first edition, 1995.
- [17] Gesellschaft für Informatik Arbeitskreis „Entwicklung Web-service-basierter Anwendungen“. Flyer zu Symposium „Entwicklung Web-Service-basierter Anwendungen“. <http://www.winf.tu-darmstadt.de/arbeitskreis/symposium.htm>, Oktober 2003.
- [18] Kai Gäbel. *Beitrag zum e-Service in der Automatisierungstechnik am Beispiel einer Regelkreisinbetriebnahme*. Diplomarbeit, Technische Universität Chemnitz, 2001.
- [19] Klaus Göldner. *Mathematische Grundlagen der Systemanalyse*, volume I. Verlag Harry Deutsch, 1981.
- [20] Kay Grammatke. *Beitrag zur Einordnung regelungstechnischer Funktionalitäten in ein flexibles, verteilbares Steuerungssystem*. Dissertation, TU Chemnitz, 1999.
- [21] World Wide Web Consortium (Web Service Architecture Working Group). Web Services Architecture Requirements W3C Working Group. <http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211>, Februar 2004.
- [22] Bierznieks Guelich, Gundavaram. *CGI-Programmierung mit Perl*. O'Reilly, 2001.
- [23] Karl Heß. Die Funktion ist wichtig. *motion world*, Juli 2004.
- [24] Karl Heß. *Script "Industrielle Steuerungstechnik"*. TU Chemnitz, 2004.
- [25] Karl Heß. *Script "Steuerungs- und Regelungstechnik"*. TU Chemnitz, 2004.
- [26] Steffen Himstedt. Die neue OPC-Architektur. *Computer & Automation*, 08 2004.
- [27] Stefan Hofman. *Beitrag zur Simulation von Produktionsmaschinen mit Bewegungssteuerungen*. Studienarbeit, TU Chemnitz, 2004.
- [28] Prof. Dr. W. Höger. *Script "Mechatronik"*. FH München, 2004.
- [29] W. Beitz; K.-H. Küttner (Hrsg.). *Dubbel*. Axel-Springer-Verlag, 18. edition, 1995.
- [30] P Hupfer. *Optimierung von Baukonstruktionen*. Teubner, 1970.

- [31] Rolf Iserman. *Mechatronic systems: fundamentals*. Axel–Springer–Verlag, first edition, 2003.
- [32] Rolf Isermann. *Identifikation dynamischer Systeme*, volume I. Axel–Springer–Verlag, erste edition, 1988.
- [33] Rolf Isermann. *Identifikation dynamischer Systeme*, volume II. Axel–Springer–Verlag, erste edition, 1988.
- [34] Frank Iwanitz. Webintegration und IT-Security. *A&D Newsletter*, April 2004.
- [35] Ulrich Kertscher. *Entwicklung eines webbasierten Berechnungs- und Gestaltungssystems für Kurvengetriebe zur Unterstützung der verteilten Produktentwicklung*. Dissertation, TU Chemnitz, 2002.
- [36] Frithjof Klasen. Webtechnologien im Vergleich. *A&D Kompendium*, 2005.
- [37] Mike Körner. *E-Service-Support im Maschinen- und Anlagenbau*. Dissertation, Universität St.Gallen, 2002.
- [38] J. Snell; D. Tidwell; P. Kulchenko. *Webservice-Programmierung mit SOAP*. O'REILLY-Verlag, first edition, 2002.
- [39] Jürgen Jasperneite; Gunnar Lessman. Integrationsplattform Profinet. *Computer & Automation*, 06 2005.
- [40] Kreuzer; Lugtenburg. *Industrieroboter*. Axel–Springer–Verlag Berlin, 1. edition, 1994.
- [41] Wendt; Lutz. *Taschenbuch der Regelungstechnik*. Verlag Harry Deutsch, vierte edition, 2002.
- [42] Anatoli Makarov. *Regelungstechnik und Simulation*. Vieweg–Verlag, second edition, 1998.
- [43] D.S. McArthur. *Strategie in research: alternative methods for design of experiments*. IRE Trans., 1961.
- [44] Arnd Lüder; Ralf Messerschmidt. Sicheres Web Based Management. *A&D Kompendium*, 2005.
- [45] I.B. Motskus. *Some experiments related to the capabilities of man in solving multiextremal problems heuristically*. Engng. Cybern., 1965.
- [46] P. C. Müller. Modellbildung, Identifikation und Simulation mittels Deskriptorsysteme. In *VDI/VDE-GMA-Kongress 2003 – Baden-Baden*. GMA, 03./04. Juni 2003.
- [47] Isabel Münch. Apache Webserver - Sicherheitsstudie. Technical report, Bundesamt für Sicherheit in der Informationstechnik, November 2002.
- [48] Markus Deppe; Eckehard Münch. *Das Mehrgrößen-Optimierungsverfahren MOPO*. Institut für Mechatronik, Universität Paderborn, April 2001.



- [49] Stefan Münz. *SelbHTML – HTML-Dateien selbst erstellen*, 8. edition, 10 2001.
- [50] Walther Oevel. *Einführung in die Numerische Mathematik*. Spektrum - Akademischer Verlag, 1996.
- [51] Barry Paton. *Grundlagen der digitalen Elektronik*. National Instruments Corporation, 6504 Bridge Point Parkway, Austin, Texas 78730-5039, 1998.
- [52] Manfred Peschel. *Rechnergestützte Analyse regelungstechnischer Systeme*. Akademie Verlag Berlin, erste edition, 1993.
- [53] Virginia H. Philipp. *CW-Studie: JAVA: Einsatz - Entwicklung - Perspektiven*. Computerwoche Verlag GmbH, Brabanter Straße 4, 80805 München, first edition, 1998.
- [54] Ingo Rechenberg. *Evolutionstrategie '94*. frommann-holzboog, first edition, 1994.
- [55] Ingo Rechenberg. Kurzfassung - Vorlesung Evolutionstrategie. [www.bionik.tu-berlin.de/institut/skript/vorle1.htm](http://www.bionik.tu-berlin.de/institut/skript/vorle1.htm), März 2004.
- [56] Erwin Samal. *Grundriß der praktischen Regelungstechnik*. Oldenbourg Verlag München, 1996.
- [57] Swen Schmeisser. *Beitrag zu offenen Automatisierungsnetzen am Beispiel eines Web-basierten Labors Automatisierungstechnik*. Dissertation, TU Chemnitz, 2004.
- [58] Oliver Schmid. *Implementierung und Optimierung impliziter iterativer Lösungsalgorithmen zur Simulation komplexer Strömungen in Turbomaschinen*. Dissertation, Universität GH Essen, 2001.
- [59] Gießler; Schmidt. *Vom Prozeß zur Regelung*. Siemens AG, 1990.
- [60] Gerhard Schnell. *Bussysteme in der Automatisierungstechnik*. Vieweg Verlag, 2. edition, 1996.
- [61] Konrad Schumacher. *Untersuchungen zur Theorie der Gradientverfahren*. Dissertation, Eberhard-Karls-Universität Tübingen, 1974.
- [62] Frank Schütte. *Automatisierte Reglerinbetriebnahme für elektrische Antriebe mit schwingungsfähiger Mechanik*. Shaker Verlag, 2003.
- [63] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionstrategie*. Birkhäuser Verlag, first edition, 1977.
- [64] Siemens AG, Automation & Drives. *SIMOTION – Getting Started*, 1/2004.
- [65] Siemens AG, Automation & Drives. *SIMOTION - Motion Control Technologieobjekt Achse*, 1/2004.



- [66] Herbert Barthel; Wolfgang Stripf. Profisave up to date. *Computer & Automation*, 06 2005.
- [67] Peter-Michael Synek. Möglichkeiten der Mechatronik nutzen. *A&D*, April 2005.
- [68] Rene Simon; Thomas Kleegrewe; Thorsten Szczepanski. Die Inbetriebnahme. *Computer & Automation*, 03 2005.
- [69] Robert Tolksdorf. *HTML & XHTML die Sprachen des Web*. dpunkt.Verlag, 5. edition, 2003.
- [70] Christian Ullenboom. *Java ist auch eine Insel*. Galileo Computing, 2003.
- [71] Heinz Unbehauen. *Regelungstechnik I*. Vieweg, elfte edition, 2001.
- [72] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA). *VDI/VDE 3685 Blatt 3 – Inbetriebnahmesysteme für Regelungen*, 09 2001.
- [73] Michael Vetter. Mit Teleservice Kosten sparen – Servicetechniker muss nicht immer vor Ort sein. *Control 2003*, 2003.
- [74] D. Schraft; E. Westkämper. e-Simulation - ein Mehrwertdienst des Verbunds. *Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA*, 2001.
- [75] Karl Wickert. Sicherheitstechnische Risiken bei der Fernwartung von Maschinen. <http://www.pz.bgn.de>, Juni 2002.
- [76] W. Wood. *In Physics of Simple Liquids*. North Holland: Amsterdam, 1968.
- [77] Klaus Wucherer. *Beitrag zur Gestaltung offener industrieller Automatisierungssysteme*. Dissertation, TU Chemnitz, 1998.
- [78] Klaus Wucherer. Motion Control: Integration von Antriebs- und Steuerungstechnik. In R. Neugebauer, editor, *Offensivkonzepte Wirtschaftlicher Produktionstechnik*. Fraunhofer Institut Werkzeugmaschinen und Umformtechnik, Verlag Wissenschaftliche Scripten, 09 2001.
- [79] [www.Wikipedia.de](http://www.Wikipedia.de). *Wikipedia Herbst 2004 CD*, 09 2004.
- [80] Oliver Zirn. *Modellbildung und Simulation mechatronischer Systeme*. expert verlag, first edition, 2002.



# Abbildungsverzeichnis

2.1	Aufbau mechatronischer Systeme . . . . .	17
2.2	Ausführungen von Bewegungssteuerungen [24] . . . . .	18
2.3	Netzaufbau in mechatronischen Systemen [24] . . . . .	18
2.4	Achsdefinition in mechatronischen Systemen . . . . .	20
2.5	Antriebsstrang mit MC-Anbindung . . . . .	20
2.6	Geschwindigkeitssollwertschnittstelle [65] . . . . .	21
2.7	Geschwindigkeitssollwertschnittstelle mit Vorsteuerung [65] . . . . .	22
2.8	Ablaufsystem . . . . .	23
2.9	Totzeiten durch Profibus DP . . . . .	24
2.10	Masken zur Inbetriebnahme . . . . .	25
2.11	Schema zur internetbasierten Dienstleistung . . . . .	27
3.1	Online-Inbetriebnahme [20] . . . . .	29
3.2	Offline-Inbetriebnahme [20] . . . . .	30
3.3	Prinzip der experimentellen Analyse . . . . .	31
3.4	Gewichtsfunktion eines PT1-Gliedes . . . . .	32
3.5	Sprungantwort eines PT1-Gliedes . . . . .	32
3.6	Treppenfunktion . . . . .	33
3.7	Generierung einer Rampenfunktion [71] . . . . .	33
3.8	Pseudo-random number generator . . . . .	34
3.9	PRBS- und PRNS-Signalverläufe . . . . .	34
3.10	Sinussignal . . . . .	35
3.11	Darstellung der Grundsysteme . . . . .	35
3.12	63 %-Methode . . . . .	36
3.13	Asymptotische Übergangsfunktion . . . . .	37
3.14	Schwach gedämpfte Systeme . . . . .	38
3.15	Identifikation mittels Simulation . . . . .	39
3.16	Simulation eines PT2-Gliedes über Integratoren . . . . .	40
3.17	Integratoren . . . . .	41
3.18	Fläche zweier Zielfunktionen . . . . .	45

3.19	Rastersuche . . . . .	46
3.20	Formoptimierungs-Experiment von Rechenberg (Auftriebsprofil) . . . . .	47
3.21	Gütemaße für Führungsverhalten . . . . .	52
3.22	Berechnung von Reglerparametern mittels Simulation . . . . .	54
4.1	Reglerinbetriebnahme im System . . . . .	55
4.2	Netzklassen im IPv4-Protokoll . . . . .	58
4.3	Geräte der Netzwerktechnik im OSI-Referenzmodell . . . . .	63
4.4	Web-Service [38] . . . . .	66
4.5	Teleservice der Firma Zeiss [73] . . . . .	67
4.6	Portal für E-Industrial-Services nach Fraunhofer [15] . . . . .	70
4.7	ePS Network Services – System . . . . .	72
5.1	Softwarearchitektur . . . . .	75
5.2	Dienstablauf . . . . .	77
5.3	Kommunikationsbeziehungen im System aus Anwendersicht . . . . .	78
5.4	Systemaufbau . . . . .	78
5.5	Kommunikationsdreieck . . . . .	80
5.6	Erweiterung der SS . . . . .	81
5.7	Testautomat . . . . .	82
5.8	Lesen einer Variable . . . . .	84
5.9	Experteneinbindung . . . . .	85
5.10	Optimierte Portalstruktur . . . . .	86
5.11	Oberfläche für die Parametrierung des Dienstes . . . . .	86
5.12	Ausgabe des Connectors bei Verbindungsaufnahme . . . . .	87
5.13	Kommunikation im Applet über die Befehlsqueue . . . . .	89
5.14	Module im Gesamtsystem . . . . .	89
5.15	Objektbeziehungen im Applet . . . . .	90
5.16	Objekte im Reglerinbetriebnahmedienst . . . . .	90
5.17	Oberfläche in der Identifikation . . . . .	91
5.18	Tooltips geben dem Anwender Hilfestellung . . . . .	91
5.19	Resultierende Struktur . . . . .	93
5.20	Verlauf (links) und Ergebnis (rechts) der Suche des Verstärkungsfaktors beim P-Regler	95
5.21	Erkennung stark schwingender Reglerdaten . . . . .	95
5.22	P-Regler . . . . .	96
5.23	P-Regler mit Vorsteuerung nach der Online-Inbetriebnahme . . . . .	97
5.25	Auswirkungen falscher Ersatzzeitkonstanten . . . . .	97
5.24	P-Regler mit Vorsteuerung und DSC . . . . .	98
5.26	Typische Reglerstruktur . . . . .	99

5.27	Objekte zur Simulation . . . . .	99
5.28	Signalquellen . . . . .	99
5.29	Einige Manipulatoren der Regelungstechnik . . . . .	100
5.30	Manipulatoren zur Gütebewertung . . . . .	100
5.31	Auszug der Simulator-Objektzusammenschaltung . . . . .	100
5.32	Umwandlung der Struktur in eine Liste . . . . .	101
5.33	Anregen der Strecke . . . . .	101
5.34	Amplitudengang eines weitgehend überschwingfrei eingestellten Geschwindigkeitsregelkreises . . . . .	103
5.35	Übersicht der verwendeten Modelle . . . . .	104
5.36	Trace des Antriebs auf eine Treppenfunktion . . . . .	104
5.37	Identifikation mittels Simulation in Java . . . . .	105
5.38	Darstellung einer Zielfunktion . . . . .	105
5.39	Konvergenz von Parametern und Zielfunktion . . . . .	106
5.40	Konvergenz (links) und Entwicklung der Artenvielfalt (rechts) . . . . .	106
5.41	Ergebnis der Identifikation mit unterschiedlichen Modellen auf Basis der Treppenfunktion . . . . .	108
5.42	Ergebnis der Identifikation mit unterschiedlichen Modellen auf Basis von PRNS . . . . .	108
5.43	Struktur zur Berechnung des KP-Faktors . . . . .	110
5.44	Validierung der Ergebnisse in der P-Reglerberechnung . . . . .	111
5.45	P-Regler mit Vorsteuerung nach der Offline-Inbetriebnahme . . . . .	112
5.46	P-Regler mit Vorsteuerung und DSC nach der Offline-Inbetriebnahme . . . . .	112
C.1	Darstellung von Lose . . . . .	131
C.2	Kennlinien von Lose . . . . .	131

# Tabellenverzeichnis

3.1	Schranken [8] . . . . .	53
4.1	OSI-Referenzmodell (Open Systems Interconnection) [60] . . . . .	57
4.2	Weitere Internet-Dienste . . . . .	60
4.3	Kommunikationsmodelle . . . . .	61
4.4	Auszug aus [37] zur Zahlungsbereitschaft für E-Services . . . . .	69
5.1	Funktionen des Testautomaten . . . . .	82
5.2	Mögliche erlaubte (X) Dienstzustandsübergänge . . . . .	88
5.3	Belegung des Tasksystems . . . . .	92
5.4	Identifikation im Online-Verfahren für $T_a = 1,5$ ms . . . . .	96
5.5	Ergebnisse der Identifikation mit Treppenfunktion . . . . .	107
5.6	Ergebnis der Identifikation mit PRNS-Signal . . . . .	109
5.7	Ermittelte Verstärkungsfaktoren . . . . .	110

# Anhang A

## Algorithmen

### $PT_2$ -Differenzgleichung mit $D < 1$

Die Differenzgleichung für ein  $PT_2$ -System mit einer Dämpfung kleiner eins lautet:

$$x(k) = b_1u(k-1) + b_2u(k-2) - a_1x(k-1) - a_2x(k-2) \quad (\text{A.1})$$

mit

$$a_0 = 1 \quad (\text{A.2})$$

$$a_1 = -2\alpha \cdot \cos(\omega_0 \cdot T_a) \quad (\text{A.3})$$

$$a_2 = \alpha^2 \quad (\text{A.4})$$

$$b_0 = 0 \quad (\text{A.5})$$

$$b_1 = K_s \cdot (1 - \beta_1) \quad (\text{A.6})$$

$$b_2 = K_s \cdot (\alpha^2 + \beta - 2) \quad (\text{A.7})$$

und den Koeffizienten

$$\alpha = e^{-\frac{D}{T}T_a} \quad (\text{A.8})$$

$$\beta_1 = \alpha \cdot \frac{\sin(\omega_0 T_a + \arccos D)}{\sqrt{1 - D^2}} \quad (\text{A.9})$$

$$\beta_2 = \alpha \cdot \frac{\sin(\omega_0 T_a - \arccos D)}{\sqrt{1 - D^2}} \quad (\text{A.10})$$

$$\omega_0 = \frac{1}{T} \sqrt{1 - D^2}. \quad (\text{A.11})$$

**$PDT_2$ -Differenzgleichung**

Analog zu Kapitel A ist die Differenzgleichung für ein System 2.Ordnung:

$$x(k) = b_1 u(k-1) + b_2 u(k-2) - a_1 x(k-1) - a_2 x(k-2) \quad (\text{A.12})$$

mit

$$a_0 = 1 \quad (\text{A.13})$$

$$a_1 = -2\alpha \quad (\text{A.14})$$

$$a_2 = \gamma \quad (\text{A.15})$$

$$b_0 = 0 \quad (\text{A.16})$$

$$b_1 = K_s \cdot (1 - \alpha + \beta\delta) \quad (\text{A.17})$$

$$b_2 = K_s \cdot (\gamma - \alpha - \beta\delta) \quad (\text{A.18})$$

und den Koeffizienten

$$\alpha = e^{-\frac{D}{T}T_a} \cdot \cos(\omega_0 \cdot T_a) \quad (\text{A.19})$$

$$\beta = e^{-\frac{D}{T}T_a} \cdot \sin(\omega_0 \cdot T_a) \quad (\text{A.20})$$

$$\gamma = e^{-2\frac{D}{T}T_a} \quad (\text{A.21})$$

$$\delta = \frac{T_V - DT}{T\sqrt{1 - D^2}} \quad (\text{A.22})$$

$$\omega_0 = \frac{1}{T}\sqrt{1 - D^2}. \quad (\text{A.23})$$



# Anhang B

## Buszugriffsverfahren

Aufgrund der Kommunikation mehrerer Kommunikationspartner über "eine" gemeinsame Datenleitung kann es zu Kollisionen der Datenpakete kommen. Die Daten sind dann unbrauchbar. Es ist ein Verfahren vorzugeben, an das sich alle Teilnehmer halten. Solche Verfahren können in kontrollierte und in zufällige Buszugriffsverfahren unterteilt werden.

Im kontrollierten Verfahren bekommt jeder Sender eine Zeit vorgegeben, in der er die alleinigen Zugriffsrechte auf den Bus besitzt. Diese Vorgabe kann über einen Master erfolgen oder über die so genannten Token, ähnlich dem Staffelstab beim Staffellauf.

Vom zufälligen Buszugriff wird gesprochen, wenn der Sender nur dann sendet, wenn er auch wirklich Daten versenden will. Hierbei kann nicht ausgeschlossen werden, dass der Bus nicht gleichzeitig von zwei Sendern genutzt wird. Es muss erkannt werden, dass die Übertragung gestört wurde und es ist nach einer vom Verfahren abhängigen Zeit erneut zu senden. Eine erneute Kollision kann jedoch ebenso wenig ausgeschlossen werden, eine Aussage über die Garantiezeit des Empfangs der Daten ist nicht möglich.

### Master-Slave-Prinzip

Das Master-Slave-Verfahren ist kontrolliert. Es existiert ein Master, der in einem Zyklus die angeschlossenen Slaves nacheinander abfragt (*polling*), welche sofort antworten. Prioritäten können über die Anzahl der polls pro Slave und Zyklus vergeben werden.

### Token-Prinzip

Beim Token-Prinzip ist jeder Teilnehmer, der den Token besitzt, der Master. Nach dem Senden wird der Token, in Form einer festgelegten Bitfolge, an den nächsten Teilnehmer weitergegeben. Da die Zeit, in der der Master den Token hält, fest vorgegeben wird, kann von einer echtzeitfähigen Kommunikation gesprochen werden.

## CSMA/CD

Die Teilnehmer greifen bei diesem Verfahren nur auf den Bus zu, wenn sie Daten zum Senden haben. Dabei hören sie den Bus ab und wenn dieser frei ist, beginnen sie ihre Übertragung. Sendet ein Teilnehmer zur gleichen Zeit, kommt es zu einer Kollision und beide Sender unterbrechen für eine zufällige Zeit, bevor sie erneut den Bus abhören und versuchen zu senden.

Um die Effektivität des Verfahrens zu steigern, wurde es mit einem Mechanismus zum Erkennen von Kollisionen erweitert (*Collision Detection – CD*). Dabei hört der Sender beim Senden gleichzeitig den Bus ab. Wenn das gesendete Signal nicht mit dem empfangenen übereinstimmt, hat eine Kollision stattgefunden. Ohne CD hätte der Sender die Kollision erst nach einer negativen Quittung des Empfängers bemerkt.

# Anhang C

## Nichtlinearitäten

### Lose

Lose treten in Produktionsmaschinen zum Beispiel an Spindeln, Getrieben und mechanischen Verbindungen auf. Ihr Einfluss fällt vor allem bei Richtungsänderungen, aber auch bei sprunghaften Geschwindigkeitsänderungen ins Gewicht. Abbildung C.1 zeigt deren Darstellung.

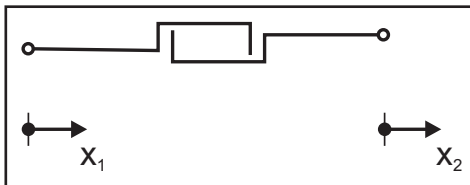


Abbildung C.1: Darstellung von Lose [28]

Die Kennlinie für Lose zeigt Abbildung C.2.

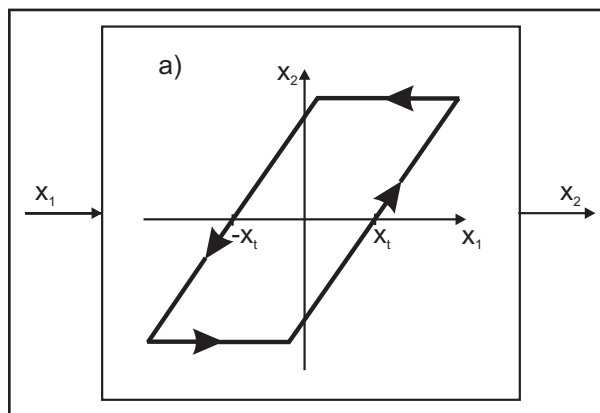


Abbildung C.2: Kennlinie von Lose [28]

## Reibung

In mechanischen Systemen spielt die Reibung eine entscheidende Rolle. Isermann unterscheidet in [31] drei Arten von auftretender Reibung:

**Festkörperreibung** – Reibung zwischen nicht geschmierten Flächen von Körpern.

**Flüssigkeitsreibung** – Reibung zwischen Flächen, die auf einem Flüssigkeits- oder Schmierfilm gleiten.

**Mischreibung** – Wenn das Verhältnis ( $\lambda = h/\sigma$ ) der Dicke  $h$  des Schmierfilmes zur Oberflächenrauheit  $\sigma$  gleich 1...3 beträgt, treten Festkörperreibung und Flüssigkeitsreibung gleichzeitig auf.

Festkörperreibung wird in Gleitreibung sowie Haftreibung unterteilt. Haftreibung ist bei Flächen anzuwenden, die sich zueinander nicht bewegen, Gleitreibung bei Flächen, die sich zueinander bewegen. Für beide Reibungsarten stehen Faktoren, die die Stärke der Reibung beschreiben. Es gilt: Die Haftreibung  $\mu_H$  ist größer als die Gleitreibung  $\mu_G$ . Dieser Unterschied ist die Ursache für den so genannten Stick-Slip-Effekt bei beginnender Relativbewegung zwischen zwei Flächen.

Die Werte der Reibungskoeffizienten, das Verhältnis aus Reibungskraft  $F_R$  zu Normalkraft  $F_N$ , liegen bei:

$$\mu_H \approx 0,15 \dots 0,8$$

$$\mu_G \approx 0,10 \dots 0,6.$$

Die Reibungskräfte werden aus dem Reibungskoeffizienten und der Normalkraft berechnet:

$$\begin{aligned} F_H &= \mu_H * F_N \\ F_G &= \mu_G * F_N \end{aligned} \tag{C.1}$$

Der Zusammenhang zwischen Geschwindigkeit und Reibkraft bei Flüssigkeitsreibung ist

$$F \sim v, \tag{C.2}$$

bei kleinen Geschwindigkeiten und laminarer Strömung im Schmiermittel. Bei hohen Geschwindigkeiten und turbulenten Strömungen im Schmiermittel gilt hingegen:

$$F \sim v^2. \tag{C.3}$$

# Lebenslauf

Name	Kai Michael Gäbel	
Geboren am	02.04.1976	
in	Karl-Marx-Stadt	
Wehrersatzdienst	Sept. 1996- Juni 1997	Grundwehrdienst
Studium	1995-1998	Grundstudium Maschinenbau an der TU Chemnitz
	1998-2001	Hauptstudium mit dem Hauptfach "Steuerungstechnik im Maschinenbau" und dem Nebenfach "Prozessdatenkommunikation"
	Juli 2001- Sept. 2001	Diplomarbeit bei der IST GmbH Chemnitz Abschluss am 12.09.2001
Praktika	11.10.1999- 10.03.2000	Siemens AG Chemnitz
Beruflicher Werdegang	01.10.2001- 24.11.2003	Wissenschaftlicher Mitarbeiter an der Professur Steuerungs- und Regelungstechnik der TU Chemnitz
	seit 25.11.2003	Wissenschaftlicher Assistent an der Profes- sur Steuerungs- und Regelungstechnik der TU Chemnitz

Chemnitz, den 08. Juni 2006