

Dynamische Funktionsprüfung hydraulischer Anlagen mit dem wissensbasierten System ^{art}*deco*

R. Lemmen

Forschungsbericht 6/94

Meß-, Steuer- und Regelungstechnik

Übersicht: Die dynamische Funktionsprüfung von ^{art}*deco* besteht im Kern aus einer automatisierten Modellbildung und der automatisierten Bestimmung der zur Modellauswertung verwendeten numerischen Verfahren sowie deren Parameter. Bei den hierfür notwendigen graphisch interaktiven Auswahl- und Präsentationsprozessen findet eine Unterstützung des Anwenders statt. Ziel ist eine möglichst schnelle Simulation bei vorgegebener Genauigkeit, wobei der Anwender durch mathematische Probleme, die nicht zu seinem Spezialgebiet gehören, nicht belastet werden soll.

Gerhard-Mercator-Universität - GH Duisburg

Meß-, Steuer- und Regelungstechnik

Prof. Dr.-Ing. H. Schwarz

Inhaltsverzeichnis

Nomenklatur	II
1 Einleitende Übersicht	1
2 Begriffsdefinitionen	2
3 Dynamische Funktionsprüfung	9
3.1 Poststatische Situation	9
3.2 Anwenderauswahl	10
3.3 Automatisierte Modellerstellung	13
3.4 Auswahl des Simulationsverfahrens	19
3.5 Bewertung	27
4 Zusammenfassung und Ausblick	28
5 Literaturverzeichnis	29
A Zur statischen Funktionsprüfung	32
A.1 Erstellung des Widerstandsnetzwerkes	32
A.2 Detektion nicht propagierbarer Substrukturen	32
B Vorbelegte Richtwerte für Konstanten	34

Formelzeichen und Bezeichnungen

Skalare Größen

A	[mm ²]	Kolbenfläche eines Zylinders
C	[-]	ausführbare Konklusion (Prämissen erfüllt)
e	[-]	globaler Fehler
EW	[-]	Entscheidungswert
h	[-]	Integrationsschrittweite
j	[-]	Konklusionszustand
k	[-]	Anzahl der Statusvariablen einer Komponente
K	[-]	Verstärkung einer Komponente
L	[-]	Lipschitz-Konstante
LC	[-]	lokale Komponente (local component)
m	[-]	Anzahl der Komponenten
MD	[-]	Modelltiefe (model depth)
NL	[-]	Intensität der Nichtlinearität
r	[-]	Variable
S	[-]	Steifheitsgrad
SI	[-]	Simulationszweck
t	[s]	Zeit
T	[-]	lokale Komponentenbeschreibung
TS	[-]	Baumstruktur (tree structure)
v	[-]	Systemversalität
Ven	[-]	Ventil
VV	[-]	Ventil-Verstärker (einschließlich U/I-Wandler)
w	[-]	Variable
x	[m]	Position (des Zylinderkolbens)
\dot{x}	[m/s]	Geschwindigkeit (des Zylinderkolbens)
\ddot{x}	[m/s ²]	Beschleunigung (des Zylinderkolbens)

y	[-]	Position des Ventilschiebers
Z	[-]	statische Statusvariable
$Z(t)$	[-]	dynamischer Statusvariable
ZYL	[-]	Zylinder

Vektorielle Größen

\mathbf{u}	Vektor der Eingangsgrößen
\mathbf{x}	Vektor der inneren Systemzustände
\mathbf{y}	Vektor der Ausgangsgrößen
\mathbf{z}	Vektor der statischen Statusvariablen eines Systems
$\mathbf{z}(t)$	Vektor der dynamischen Statusvariablen eines Systems
\mathbf{z}_∞	Vektor der stationären Statusvariablen eines Systems

Mengen

\mathbb{C}	Menge der ausführbaren Konklusionen (Prämissen erfüllt)
\mathbb{KK}	Menge der knotenparallelen Kanten
\mathbb{LC}	Menge der Komponenten (local components)
\mathbb{MD}	Menge der Modelltiefen
\mathbb{N}	Menge der natürlichen Zahlen
\mathbb{R}	Menge der reellen Zahlen
\mathbb{SI}	Menge der Simulationszwecke
\mathbb{T}	Menge der lokalen Komponentenbeschreibungen
\mathbb{Z}	Menge der statischen Statusvariablen
$\mathbb{Z}(t)$	Menge der dynamischen Statusvariablen

Griechische Buchstaben

β	[-]	Konstante
ε	[-]	lokaler Fehler
λ	[-]	Eigenwert
φ	[-]	Flächenverhältnis
κ	[-]	Bewertungsfaktor eines Einflußfaktors
ω	[1/s]	Frequenz einer Komponente

Indizierung

a	Ausgang
$bezug$	bezogen
$bound$	Schranke (bound)
cut	Abschneide(-wert)
C	Komponente (component)
e	Eingang
end	Ende
$gate$	Wert am Anschluß
gew	gewählter Wert
h	Schrittweite
i, j	Laufindex
k	bekannt (known)
K	Verstärkung
max	maximal
min	minimal
MD	Modelltiefe
NL	Nichtlinearität
o	Eigen(-frequenz)
opt	optimal

<i>rel</i>	relativer Wert (auf [0,1] normiert)
<i>sel</i>	selektierte (Komponente)
<i>self</i>	eigener Wert
<i>sim</i>	Simulation
<i>struc</i>	Struktur (Ersatzgröße; structure)
<i>top</i>	aktuelle (Komponente; topical)
<i>u</i>	unbekannt
<i>v</i>	Systemversalität
*	qualitativer Strukturwert

Funktionen und Operatoren

$\dim(\cdot)$	Anzahl der Elemente (vgl. Schwarz 1991)
$f(\cdot)$	statische Beschreibungsfunktion
$\mathbf{f}(\cdot)$	Vektorfeld der statischen Beschreibung
$g(\cdot)$	dynamische Beschreibungsfunktion
$\mathbf{g}(\cdot)$	Vektorfeld der dynamischen Beschreibung
$\max(\cdot)$	Maximumbildung
$\min(\cdot)$	Minimumbildung
$M(\cdot)$	Übertragungsfunktion des Integrationsverfahrens
$T(\cdot)$	lokale Komponentenbeschreibung
$\vec{T}(\cdot)$	Vektorfeld der lokalen Komponentenbeschreibung
\dashv	Rundungsfunktion
$ \cdot $	Betrag der Menge (Anzahl der Elemente)
$\ \cdot\ $	Norm des Vektors ($\ \cdot\ = \max(\cdot_1 , \cdot_2 , \dots, \cdot_n)$)
\forall	für alle
$\lceil \cdot \rceil$	ceiling (höhere Integerzahl)
$\lfloor \cdot \rfloor$	floor (niedrigere Integerzahl)
$\bar{\cdot}$	arithmetischer Mittelwert

1 Einleitende Übersicht

Die dynamische Funktionsprüfung hydraulischer Anlagen mit dem wissensbasierten System *“^{arg}deco”* soll bei verlangter Genauigkeit möglichst schnell und für den Anwender intuitiv einsichtig sowie weitestgehend graphisch interaktiv zu bedienen sein. Der Anwender ist also von mathematischen und informationstechnischen Problemen, die nicht zu seinem Spezialgebiet gehören und bezüglich derer er häufig kaum Wissen besitzt, zu befreien. Dieses Ziel wird im wissensbasierten System *“^{arg}deco”* dadurch erreicht, daß die Generierung des Modells und dessen mathematische Auswertung automatisiert sind. Hierbei wählt der Anwender lediglich den zu untersuchenden Anlagenteil und somit dessen Komponenten sowie den Simulationszweck aus. Der Simulationszweck repräsentiert dabei ein Maß für die erforderliche Genauigkeit des Simulationsergebnisses und damit des zugrundeliegenden Modells. Die anschließende Bewertung der Ergebnisse bleibt zwangsläufig dem Anwender überlassen, der dabei aber durch zusätzliche Informationen unterstützt wird.

An diese Einführung schließt sich nach einer umfangreichen, aber aufgrund der vielen neuen bzw. speziell verwendeten Begriffe notwendigen Begriffsdefinition (Abschnitt 2) die eigentliche Beschreibung der dynamischen Funktionsprüfung in (Abschnitt 3) an. Diese untergliedert sich in Unterabschnitte, die zunächst die Situation nach der statischen Funktionsprüfung (Abschnitt 3.1) und die Auswahl durch den Anwender (Abschnitt 3.2) beschreiben. Hieran schließt sich die automatisierte Modellbildung (Abschnitt 3.3) an, bei derausgehend von dem gewählten Simulationszweck, der Struktur des zu untersuchenden Anlagenteils und den Komponenten mit deren Kenngrößen (Verstärkung, Eigenkreisfrequenz bzw. Eigenzeitkonstante, Systemversalität und Nichtlinearität) das mathematische Modell bestimmt wird. Bei der Auswahl des Simulationsverfahrens (Abschnitt 3.4) ist sowohl das Verfahren, als Ordnung und die Parameter allein aus dem Simulationszweck und dem nun bestimmten Modell automatisch so festzulegen, daß der Anwender nicht mit den hierbei auftretenden mathematischen Problemstellungen konfrontiert wird. Die Einbeziehung der statischen Ergebnisse und der Fehlerbetrachtungen unterstützt die Bewertung der Ergebnisse (Abschnitt 3.5). Ein zusammenfassender Ausblick (Abschnitt 4) schließt diesen Bericht ab.

2 Begriffsdefinitionen

Dieser Abschnitt erläutert die folgenden Begriffe, wie sie im wissensbasierten System ^{art}deco Verwendung finden.

Definition 2.1 Parameter und Statusvariablen

Parameter beschreiben ein System und sind in der von den Statusvariablen beliebig nichtlinear abhängigen Systembeschreibung immer konstant. Eine Änderung der Parameter bedeutet eine Systemänderung. Die Änderung der Statusvariablen überführt das System dagegen in einen anderen Systemstatus, ändert es aber nicht.

□

Definition 2.2 Vektor der Statusvariablen \mathbf{z} bzw. $\mathbf{z}(t)$

Der Vektor der Statusvariablen \mathbf{z} bzw. $\mathbf{z}(t)$ enthält alle Statusvariablen des Gesamtsystems und setzt sich aus den Vektoren der Statusvariablen \mathbf{z}_C bzw. $\mathbf{z}_C(t)$ der einzelnen Komponenten zusammen. Die aus dem Bereich der Zustandsraumdarstellung bekannten Größen innere Systemzustände \mathbf{x} , Eingangsgrößen \mathbf{u} , Ausgangsgrößen \mathbf{y} sowie Störgrößen \mathbf{z} ¹ sind im Vektor der Statusvariablen \mathbf{z} bzw. $\mathbf{z}(t)$ enthalten. Man unterscheidet zwischen dem Vektor der statischen Statusvariablen \mathbf{z} (Definition 2.4) und dem Vektor der dynamischen Statusvariablen $\mathbf{z}(t)$ (Definition 2.5).

□

Definition 2.3 Hebbare Unstetigkeit

Eine Funktion $g(r)$, $r \in \mathbb{R}$ ist nach (Bronstein und Semendjajew 1985) dann hebbare unstetig, wenn

$$\lim_{r \rightarrow r_o} = \beta, \quad \beta \in \mathbb{R} \quad (2.1)$$

sowohl als linksseitiger und rechtseitiger Grenzwert existiert.

□

Definition 2.4 Statisch

Ein System heißt im Sinne von ^{art}deco statisch, wenn das Zeitverhalten der statischen Statusvariablen \mathbf{z} durch i. allg. nichtlineare Gleichungen beschrieben wird. Dabei sind $\mathbf{f}(\mathbf{z})$ stetige und/oder hebbare unstetige (2.3) Funktionen von $\sum_{j=1}^m k_j$ unabhängigen Statusvariablen

$$\mathbf{z} = \mathbf{f}(\mathbf{z}) \quad , \quad (2.2)$$

wobei \mathbf{z} nicht notwendigerweise unabhängig von der Zeit t sein muß (ausführliche Erklärung s.u.). $\mathbf{f}(\mathbf{z})$ heißt Vektorfeld von \mathbf{z} (Schwarz 1991).

¹ \mathbf{z} wird im folgenden für den Vektor der Statusvariablen und nicht für die Störgröße, die in \mathbf{z} enthalten ist, verwendet.

□

Diese Definition schließt auch Statusänderungen, die von der Zeit abhängen mit ein. Als Beispiel sei hier der mit konstanter Geschwindigkeit ($\dot{x} = \text{const.}$) verfahrende Zylinder genannt, für den $x = \beta \cdot t$ gilt. Weiter beinhaltet diese Definition auch Statusverläufe, die hebbare unstetig und/oder nicht stetig differenzierbar sind, welche in der Praxis häufig vorkommen (Bild 2.1) und somit nicht analytisch² sind.

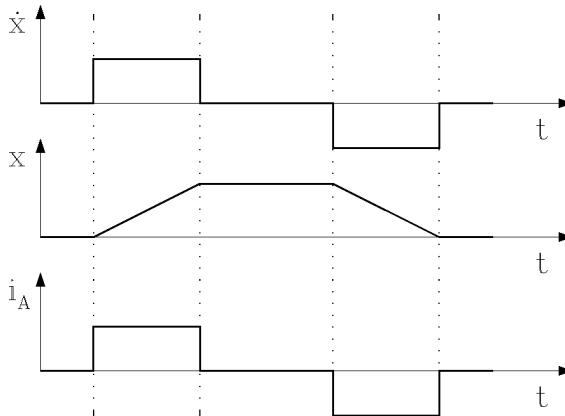


Bild 2.1: Ausschnitt aus dem Funktionsdiagramm eines statischen Statusverlaufs

Eine strengere Definition in dem Sinne, daß alle zeitveränderlichen Größen konstant und deren Ableitungen nach der Zeit t somit Null sind (Föllinger 1992), ist hier nicht zulässig. Die Funktionen f repräsentieren sowohl das Verhalten der Einzelkomponenten als auch deren Kopplungen zum Gesamtsystem.

Definition 2.5 *Dynamisch*

Ein System heißt im Sinne von *ardecō* dynamisch, wenn das Zeitverhalten der dynamischen Statusvariablen $\mathbf{z}(t)$ durch i. allg. nichtlineare Differentialgleichungen beschrieben wird. Zusätzlich können auch (nichtlineare) Gleichungssysteme auftreten. Es gilt für die Differentialgleichungssysteme

$$\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z}(t)) \quad (2.3)$$

und für die Gleichungssysteme

$$\mathbf{z} = \mathbf{f}(\mathbf{z}) \quad (2.4)$$

mit $\mathbf{g}(\mathbf{z}(t))$ und $\mathbf{f}(\mathbf{z})$ wie in Definition 2.4 als Funktionen von $\sum_{j=1}^m k_j$ unabhängigen Statusvariablen $\mathbf{z}(t)$ bzw. \mathbf{z} . Sowohl $\mathbf{g}(\mathbf{z}(t))$ als auch $\mathbf{f}(\mathbf{z})$ sind Vektorfelder nach Schwarz (1991).

□

²Es existiert keine Approximation durch eine konvergente Taylorreihe an der Unstetigkeitsstelle (Schwarz 1991)

Definition 2.6 Stationär

Ein System hat im Sinne von ^artdeco dann seinen stationären Status \mathbf{z}_∞ erreicht, wenn das Zeitverhalten der dynamischen Statusvariablen $\mathbf{z}(t)$ in eine quasistationäre Beschreibung nach Definition 2.4 übergegangen ist.

□

Im Idealfall einer fehlerfreien dynamischen Modellbeschreibung und dynamischen Funktionsprüfung entsprechen die stationären Systemzustände \mathbf{z}_∞ den statischen Systemzuständen \mathbf{z} . Eine strengere Definition des Begriffes „stationär“ wie z. B. von Föllinger (1992), daß die zeitveränderlichen Größen des Systems konstant und daher ihre Ableitungen nach der Zeit t Null sind, ist, wie im Beispiel zu Definition 2.4 dargestellt, hier nicht zulässig.

Definition 2.7 Anwender von ^artdeco

Der Anwender benutzt das System ^artdeco , um mit einem gegebenen Komponentenkatalog hydraulische Anlagen zu konfigurieren und zu prüfen. Er hat keine Änderungsmöglichkeit von Komponenten, insbesondere auf globaler, also der Komponentenkatalogebene. Lediglich alle Statusvariablen sowie ein eingeschränkter Parameterbereich sind für ihn zugänglich.

□

Definition 2.8 Systembeauftragter von ^artdeco

Dem Systembeauftragten obliegt die Wartung der Wissensbasis von ^artdeco . Diese umfaßt die Neuerstellung und Änderung der Bauteile des Komponentenkataloges, Festlegung der Modelltiefen (Definition 2.9), der Modellbeschreibungen, des Simulationszweckes (Definition 2.10), von Regeln zur Bestimmung der Simulationsverfahren und -parameter, usw.

□

Definition 2.9 Modelltiefe

Die Modelltiefe MD ist nach Berger (1993) der Grad der Berücksichtigung von Einflußfaktoren auf die Komponente und deren Umsetzung im Modell.

□

Auch bei den Modelltiefen MD wird die statische und die dynamische Beschreibung unterschieden. Allgemein gilt für die Anzahl der Modelltiefen $|\mathbb{M}\mathbb{D}| \in \mathbb{N}$. Für die statische Beschreibung werden im Gegensatz zur dynamischen Beschreibung keine Modelltiefen von vornherein unterschieden. Diese ergeben sich vielmehr in Abhängigkeit der bekannten Komponentenparameter und -statusvariablen. Bei der dynamischen Beschreibung findet sich in der Modelltiefe 0 die statische Systembeschreibung wieder. Die Modelltiefe 1 beinhaltet üblicherweise ein lineares Systemmodell, das in den folgenden Modelltiefen, nichtlinear erweitert, immer genauer beschrieben wird.

Definition 2.10 *Simulationszweck*

Der Simulationszweck (Intention) $SI \in \mathbb{N}$ ist ein Maß für die vom Anwender geforderte Untersuchungs- und damit Modellierungsgenauigkeit.

□

Die Anzahl der Simulationszwecke $|\mathbb{SI}|$ und die Anzahl der Modelltiefen $|\mathbb{MD}|$ sind miteinander über die Bedingung

$$2 \leq |\mathbb{SI}| \leq |\mathbb{MD}| \quad (2.5)$$

verknüpft.

Die Simulationszwecke können vom Systembeauftragten, in Absprache mit den Anwendern, nach steigender Genauigkeit, beliebig definiert werden. Im folgenden wird eine Vorbelegung von drei Simulationszwecken SI behandelt, wobei das zugehörige Modell einen Eindruck der notwendigen Modelltiefe vermitteln soll:

- | | |
|----------------------|--|
| 1. Systemabschätzung | zugehöriges Modell: grob, linear, evtl. mit Begrenzungen |
| 2. Systemsimulation | zugehöriges Modell: grob, nichtlinear |
| 3. Optimierung | zugehöriges Modell: detailliert, nichtlinear |

Definition 2.11 *Lokale Propagierung*

Die lokale Propagierung ist ein Constraint(Ebner 1992)–Verarbeitungsverfahren, das auf einer Ein–Schritt–Deduktion aufbauend, auch Abhängigkeiten und Alternativen verarbeiten kann.

□

Die lokale Propagierung lässt sich durch folgende Verarbeitungsfolge beschreiben:

Gegeben ist die Menge der globalen bekannten Statusvariablen $\mathbb{Z}_{k,j}$ und die Menge der globalen unbekannten Statusvariablen $\mathbb{Z}_{u,j}$ zum Konklusionszustand j mit $j \in \mathbb{N}$. Diese lassen sich untergliedern in die Mengen der lokalen bekannten Statusvariablen $\mathbb{Z}_{k,i}$ und der lokalen unbekannten Statusvariablen $\mathbb{Z}_{u,i}$ der Komponenten mit $i \in \mathbb{N}$, $i = 1, \dots, m$, $m = \text{Anzahl der Komponenten}$, wobei gilt:

$$\mathbb{Z}_{k,j} = \bigcup_{i=1}^m (\mathbb{Z}_{k,j})_i \quad (2.6)$$

und

$$\mathbb{Z}_{u,j} = \bigcup_{i=1}^m (\mathbb{Z}_{u,j})_i \quad (2.7)$$

mit $j \in \mathbb{N}$.

Für jede lokale Komponentenbeschreibung T_i existiert die Menge \mathbb{C}_i der ausführbaren Konklusionsmöglichkeiten (Prämissen erfüllt), mit

$$|\mathbb{C}_i| = |(\mathbb{Z}_{u,j})_i| \quad , \quad (2.8)$$

die den Konklusionszustand von j nach $j + 1$ überführen. Dies wird solange durchgeführt, bis für ein $j = q$, $q \in \mathbb{N}$, gilt:

$$\mathbb{Z}_{uq} = \emptyset \quad \vee \quad \bigcup_{i=1}^m \mathbb{C}_i = \emptyset \quad . \quad (2.9)$$

Für $\mathbb{Z}_{uq} = \emptyset$ folgt die vollständige, für $\mathbb{Z}_{uq} \neq \emptyset \wedge \bigcup_{i=1}^m \mathbb{C}_i = \emptyset$ folgt die partielle Lösung. Eine Reduzierung der Anzahl der global ausführbaren Konklusionen gegenüber der Summe der lokal ausführbaren Konklusionen ergibt sich aufgrund von strukturellen Abhängigkeiten, die sich mittels der lokalen Propagierung nicht untersuchen lassen. Eine detaillierte Beschreibung der in *artdeco* verwendeten lokalen Propagierung zur statischen Funktionsprüfung liefert Hoffmann (1993).

Definition 2.12 Baumartige und nicht baumartige Strukturen

Eine Struktur, die durch einen Graphen dargestellt werden kann, der aus einer Wurzel und beliebig vielen Ästen so besteht, daß kein Knoten sich selbst erreichen kann, heißt baumartig. Kann sich ein Knoten selbst erreichen, heißt die Struktur nicht baumartig.

□

Bild 2.2 zeigt beispielhaft eine baumartige und eine nicht baumartige Struktur.

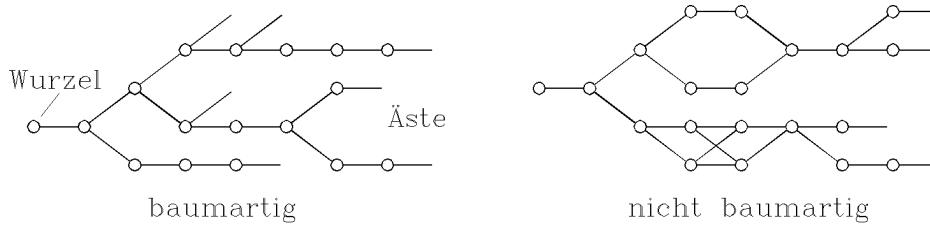


Bild 2.2: Baumartige und nicht baumartige Struktur

Bei der statischen Funktionsprüfung wird ein Widerstandsnetzwerk wie in Bild 2.3 erzeugt und dieser Graph der Strukturuntersuchung unterzogen. Der Graph ist zunächst ungerichtet. Im Anhang A wird dieses genauer erklärt.

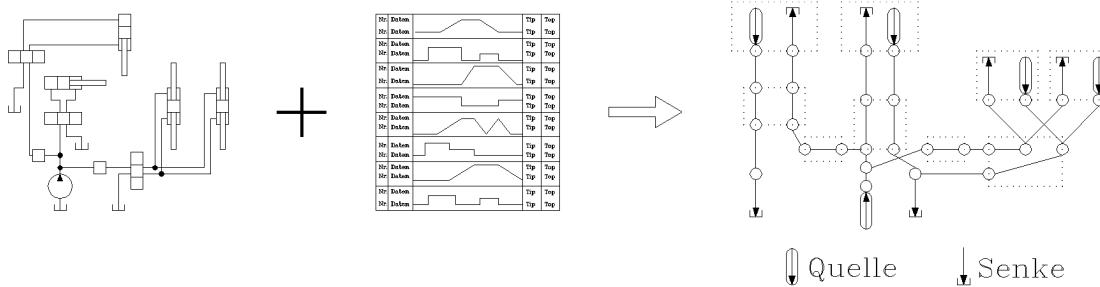


Bild 2.3: Transformation des Hydraulikkreislaufes in ein Widerstandsnetzwerk

Bei der dynamischen Funktionsprüfung wird aus dem Hydraulikschaltplan Bild 2.4 ein gerichteter Informationsgraph wie in Bild 2.5 erzeugt, der strukturell dem Widerstandsnetzwerk der

statischen Funktionsprüfung ähnelt, sich allerdings durch seine Richtung und die verschiedenen lokalen Komponentenbeschreibungen T der noch unbekannten Modelltiefen MD unterscheidet.

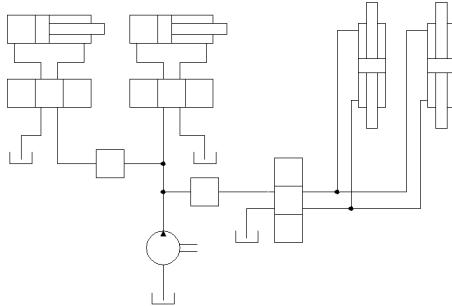


Bild 2.4: Hydraulikkreislauf

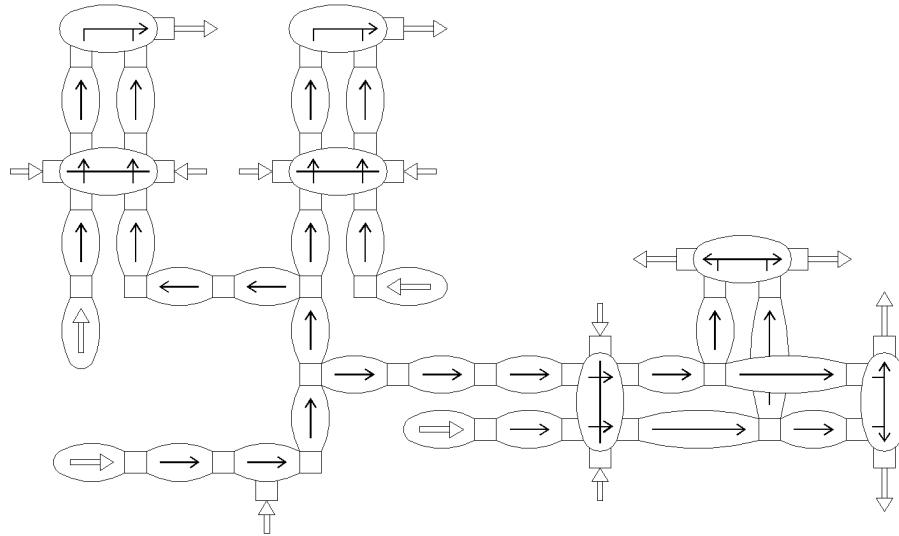


Bild 2.5: Gerichteter Informationsgraph zum Hydraulikkreislauf

Definition 2.13 $\lceil r \rceil$

Die Funktion $\lceil r \rceil$ ergibt das nächst größere $w \in \mathbb{N}$, $w \geq r$, $r \in \mathbb{R}$

□

Definition 2.14 $\lfloor r \rfloor$

Die Funktion $\lfloor r \rfloor$ ergibt das nächst kleinere $w \in \mathbb{N}$, $w \leq r$, $r \in \mathbb{R}$

□

Definition 2.15 Rundungsfunktion $\dashv (|\mathbb{MD}_{top}|; [r])$

Die Funktion \dashv bildet das Argument $r \in [0, 1]$ nach der Vorschrift

$$\left\lceil \frac{\lfloor (2 \cdot (|\mathbb{MD}_{top}| - 1) [r]) \rfloor}{2} \right\rceil \quad (2.10)$$

auf das Intervall $[0, (|\mathbb{MD}_{top}| - 1)] \in \mathbb{N}$ ab.

□

Definition 2.16 Systemversalität v

Gegeben ist ein beliebig nichtlineares dynamisches System nach (2.3) für eine Komponente als

$$\dot{\mathbb{Z}}_C(t) = g_C(\mathbb{Z}_C(t)) \quad . \quad (2.11)$$

Das zugrundeliegende technische System kann als ideal begrenzt angesehen werden und wird für einen Arbeitspunkt bei 50 % der Systembegrenzung – ausgehend von der höchsten Modelltiefe ($|\mathbb{MD}|$) – linearisiert und in ein (im allgemeinen) Mehrgrößenzustandsmodell – mit den Systemmatrizen \mathbb{A} , \mathbb{B} , \mathbb{C} – überführt. Die Systemmatrizen werden nach Svaricek (1987) in die *qualitativen* Strukturmatrizen \mathbb{A}^* , \mathbb{B}^* , \mathbb{C}^* transformiert und der gerichtete Strukturgraph erzeugt. Die Systemversalität³ v bestimmt sich zu

$$v = \dim(\mathbb{Z}_C(t)) + |\mathbb{KK}| \quad , \quad (2.12)$$

der Summe der Dimension des Vektors der Statusvariablen $\mathbb{Z}_C(t)$ und der Anzahl der knotenparallelen Kanten $|\mathbb{KK}|$ des Strukturgraphen.

□

Bild 2.6 zeigt verschiedene knotenparallele Kanten.

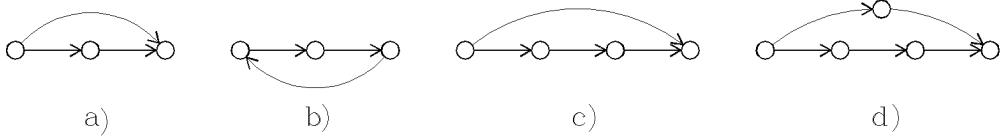


Bild 2.6: Knotenparallele Kanten (a, c und d)

Während der Graph a) eine knotenparallele Kante besitzt, enthält der Graph b) lediglich einen Zyklus. Werden von einer knotenparallelen Kante n Knoten übergangen, so geht diese auch n -fach in die Menge der knotenparallelen Kanten \mathbb{KK} ein; der Graph c) also zweifach. Dabei wird nur die Differenz der Anzahl der übergangenen Knoten gezählt, bei Graph d) also einfach.

Definition 2.17 Nichtlinearität NL

Die Nichtlinearität NL bildet reale technische Nichtlinearitäten (hier hydrostatisch mechatronische Systeme) kontinuierlich auf das Intervall $[0, 1]$ ab, wobei 0 einem vollständig linearen und 1 dem maximal nichtlinearen System entspricht.

□

Definition 2.18 Steifheitsgrad S

Der Steifheitsgrad eines Systems ist gegeben durch

$$S = \left| \frac{\omega_{o,max}}{\omega_{o,min}} \right| \quad , \quad (2.13)$$

der auf die minimale Eigenfrequenz $\omega_{o,min}$ bezogenen maximalen Eigenfrequenz $\omega_{o,max}$.

□

³Die Systemversalität ergibt für ein lineares System ein verallgemeinertes Maß der das System repräsentierenden Pole und Nullstellen.

3 Dynamische Funktionsprüfung

Bei der dynamischen Funktionsprüfung wird genauso wie bei der statischen Funktionsprüfung mit einer komponentenorientierten lokalen Beschreibung sowie mit Topologieinformationen das Verhalten der Gesamtanlage dargestellt (Lemmen 1992b und Hoffmann 1993). Die dynamische Funktionsprüfung baut auf einer erfolgreichen statischen Funktionsprüfung auf. Deshalb wird in Unterabschnitt 3.1 zunächst die Informationssituation nach Abschluß der statischen Funktionsprüfung beschrieben, um anschließend die Teilbereiche der dynamischen Funktionsprüfung darzustellen, die sich in

- Auswahl,
- Modellerstellung,
- Simulationsverfahrensauswahl und
- Bewertung

untergliedern. Im Unterabschnitt 3.2 welcher die Auswahl durch den Anwender erläutert wird die Auswahl des zu simulierenden Anlagenteils, des Untersuchungszeitraumes und des Simulationszweckes behandelt. Für die automatisierte Modellerstellung müssen im Unterabschnitt 3.3 zunächst die dynamische Modellbeschreibung in verschiedenen Modelltiefen, die Modelltiefenbestimmung und die damit eventuell verbundene Modellreduktion beschrieben werden. Die Auswahl des numerischen Simulationsverfahrens im Unterabschnitt 3.4 wird, genauso wie die Auswahl der Simulationsparameter, dem Anwender automatisch vorgeschlagen. Zur Bewertung des Simulationsergebnisses in Unterabschnitt 3.5 können zumindest bezüglich der stationären Endwerte der simulierten Systemzustände die statischen Werte dieser Systemzustände aus der statischen Funktionsprüfung verwendet werden.

3.1 Poststatische Situation

Hier wird die Informationssituation dargestellt, wie sie im Zustand nach erfolgreicher und vollständiger statischer Funktionsprüfung beim wissensbasierten System *artdeco* (Stein, Hoffmann und Lemmen 1992) vorliegt. An dieser Stelle sei direkt darauf hingewiesen, daß die in diesem Bericht schwerpunktmäßig behandelte dynamische Funktionsprüfung nur für den dynamisch untersuchten Anlagenteilbereich eine vollständige statische Funktionsprüfung voraussetzt. Die statische Funktionsprüfung wird mit einem Prüfungsalgorithmus durchgeführt, der die folgenden Module beinhaltet:

- lokale Propagierung,
- Strukturverarbeitung,
- Lösen eines linearen Gleichungssystems und
- Lösen eines nichtlinearen Gleichungssystems.

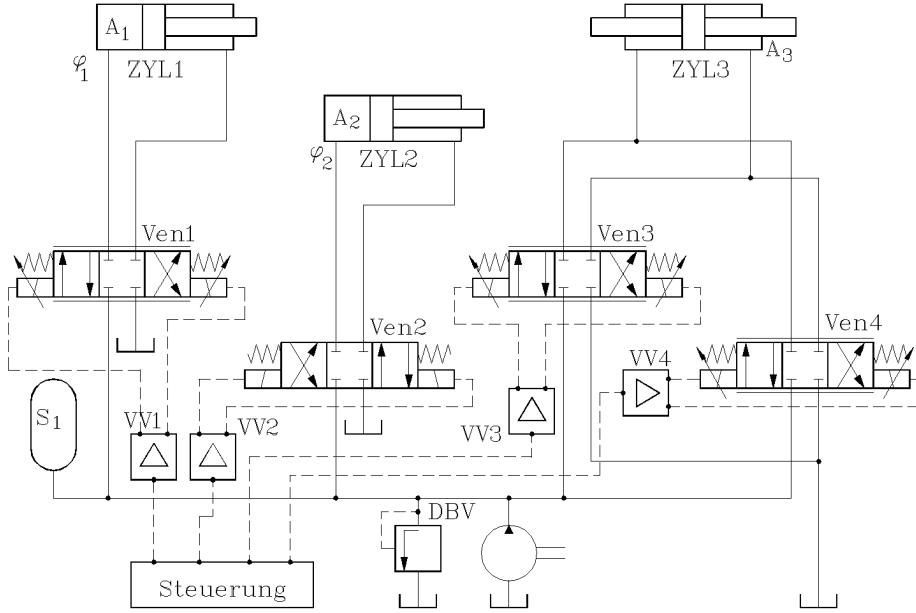


Bild 3.1: Hydraulischer Beispieldrahtzug

Eine detaillierte Beschreibung der Prüfung findet sich in Hoffmann (1993) sowie Lemmen (1992a,b) und wird im folgenden als bekannt vorausgesetzt.

Eine erfolgreiche statische Funktionsprüfung belegt alle Systemzustände Z einer hydraulischen Anlage zu allen Untersuchungszeitpunkten t_j mit Werten. Das Funktionsdiagramm, das die Zustandsinformationen graphisch wiedergibt, wird zwischen den Untersuchungspunkten durch Geradenverläufe ergänzt. Bild 3.1 gibt beispielhaft eine hydraulische Anlage und Bild 3.2 einen Ausschnitt des zugehörigen Funktionsdiagrammes wieder. Die topologischen Informationen aus dem Schaltplan (Bild 3.1), die Informationen über die statischen Zustände Z und die Informationen der lokalen Komponentenbeschreibungen T bilden die statische Gesamtinformation der hydraulischen Anlage.

3.2 Anwenderauswahl

Der Anwender wählt die zu simulierenden Bereiche der hydraulischen Anlage graphisch aus. Sowohl die Gesamtanlage als auch einzelne Anlagenteile können mit Mausunterstützung ausgewählt werden. Wird etwa aus dem Beispieldrahtzug (Bild 3.1) die hydraulische Achse ZYL1 zur Simulation mit dem in Bild 3.3 gezeigten Bereich selektiert, bleiben alle (wenn auch nur teilweise) im Mausfenster enthaltenen Objekte erhalten. Alle anderen Objekte werden entfernt.

Der Anwender kann nun dynamische Eingangsgrößen $Z_e(t)$ und dynamische Ausgangsgrößen $Z_a(t)$ mit

$$Z_e(t) \wedge Z_a(t) \in \mathbb{Z}(t) \quad (3.1)$$

festlegen. Für die dynamischen Eingangsgrößen müssen die Zustandsgrößenzeitverläufe angegeben werden. Die Zustandsgrößenzeitverläufe der dynamischen Ausgangsgrößen dagegen sind

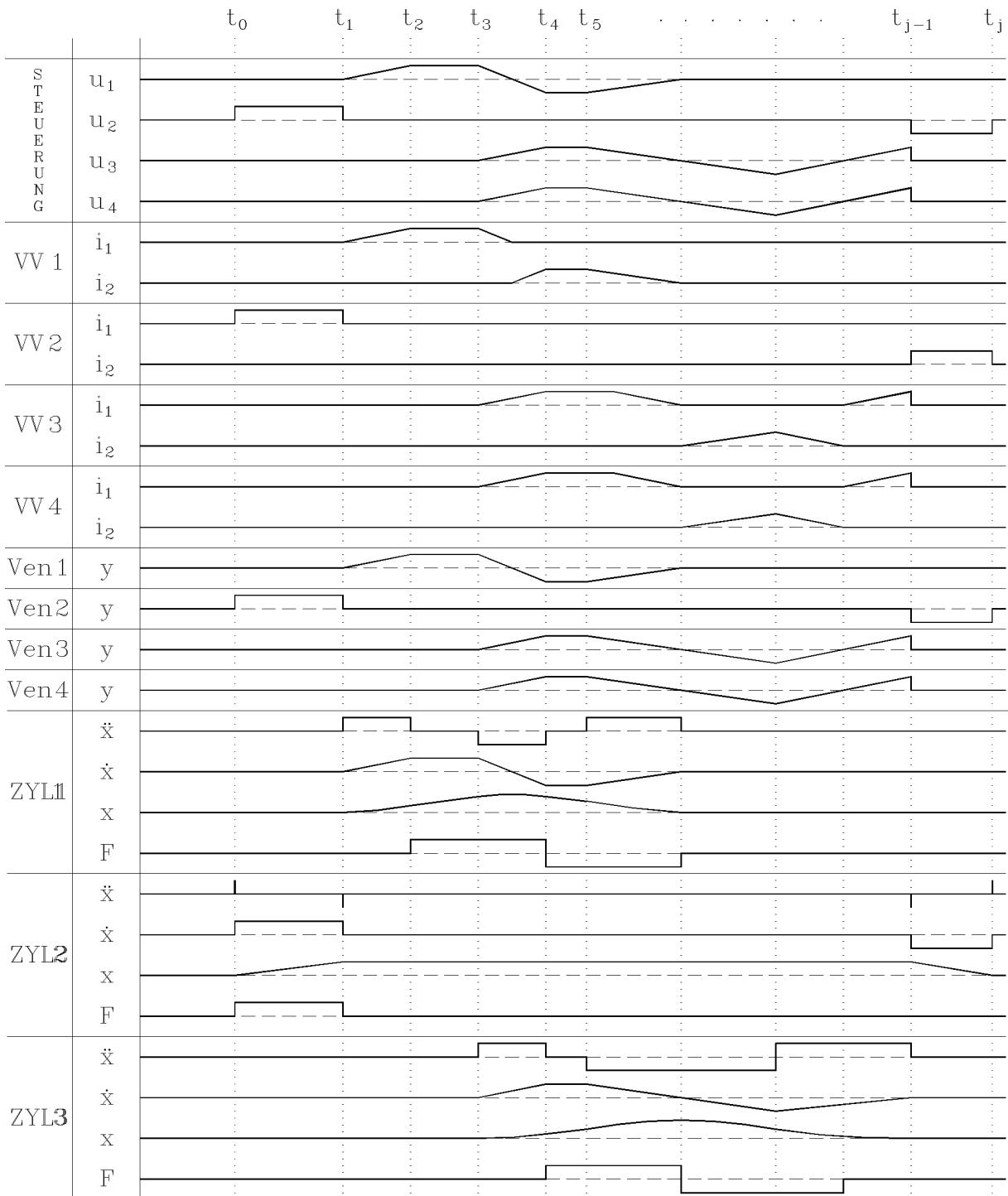


Bild 3.2: Ausschnitt eines Funktionsdiagramms

das Ergebnis der Simulation. Es muß zumindest eine dynamische Eingangsgröße und eine Ausgangsgröße festgelegt werden. Alle anderen Zustandsgrößen an jetzt noch offenen Schnittstellen sind automatisch durch ihre statischen Zustände Z bestimmt. Der Anwender muß also nur die – wie beispielhaft in Bild 3.4 mit einem Doppelpfeil gekennzeichneten – dynamischen Ein- und Ausgangsgrößen definieren. Eine weitere Auswahl, die der Anwender zu tätigen hat, ist die Auswahl des Simulationszweckes $SI_{gew} \in \mathbb{SI}$ (vgl. Definition 2.10).

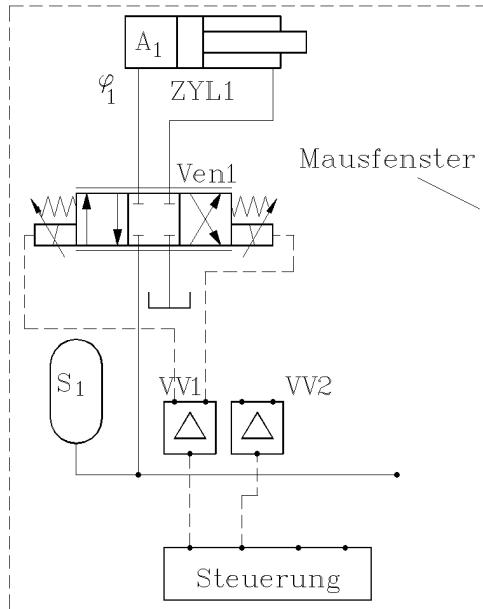


Bild 3.3: Selektierte Objekte aus dem Beispielkreislauf

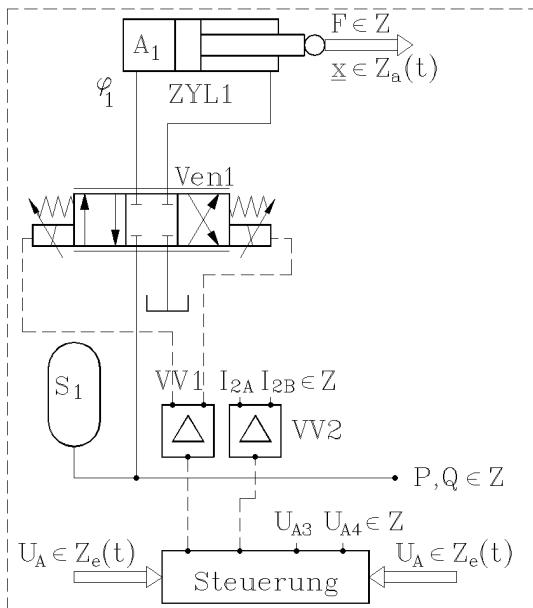


Bild 3.4: Schnitt- sowie Ein- und Ausgangsgrößenfestlegung

3.3 Automatisierte Modellerstellung

Die Aufgabe der automatisierten Modellerstellung besteht darin, ausgehend von der selektierten Struktur, den selektierten Zustandsgrößen (Bild 3.4) sowie den lokalen Komponentenbeschreibungen, ein Gesamtmodell des dynamischen Verhaltens des ausgewählten Anlagenteils zu generieren. Dabei wird die Modellordnung durch das – vom gewählten Simulationszweck SI_{gew} abhängige – Abschneiden höherfrequenter dynamischer Anteile ($\omega_o > \omega_{cut}$) reduziert. Die Modelltiefen MD_{gew} der dynamischen Modellbeschreibungen der Komponenten T_C werden bestimmt und die dynamischen Systemzustände an den Schnittstellen $Z_{gate}(t)$ entsprechend der Struktur unifiziert.

Die Unifikation, welche das Gleichsetzen der Zustände einer Schnittstelle (*gate*) einer Komponente mit den Zuständen der verbundenen Komponente an dieser Schnittstelle bedeutet, stellt kein Problem dar. Die Auswahl der Modelltiefe steht aufgrund ihrer Komplexität im Zentrum der weiteren Betrachtungen. Bei der Modelltiefenbestimmung spielen sechs Einflußfaktoren eine Rolle, deren Einflußnahme wiederum von der Wahl des Simulationszweckes $SI_{gew} \in \mathbb{SI}$ abhängig ist. Diese Einflußfaktoren sind:

- Die Struktur (hier die Baumstruktur TS) beeinflußt stark die Regeln, nach denen das Modell der Anlage aufgestellt wird. Es werden baumartige und nichtbaumartige Strukturen unterschieden.
- Der Mittelwert der relativen Modelltiefen $\overline{MD_{rel}}$ beinhaltet die Information über die schon bearbeiteten Modelle und soll sicherstellen, daß die einzelnen Modelltiefen sinnvoll zusammenpassen.
- Die Eigenfrequenz ω_o bzw. die Eckfrequenz ist bei technischen Systemen das dynamisch bedeutsamste Merkmal.
- Die Verstärkung K spielt speziell bei nichtbaumartigen Strukturen eine Rolle für die Gewichtung der Einflußnahme der einzelnen Komponenten.
- Die Systemversalität v repräsentiert u. a. die Anzahl der Energiespeicher in einer Komponente. Da bei den Hydraulikkomponenten phasenabsenkende Systeme wesentlich sind, kann dieser Einfluß, durch eine vereinfachte Vorgehensweise nach Svaricek (1987) leicht berücksichtigt werden.
- Die Nichtlinearität wird anhand der Kennzahl NL bewertet. Diese wird mittels eines modifizierten Verfahrens beruhend auf Reuter (1992) ermittelt (Vier 1994).

Die folgenden Ausführungen beschreiben die Einflußfaktoren und deren Herleitung eingehend.

Baumstruktur TS

In der praktischen Anwendung ist die Baumstruktur maßgebend. Nichtbaumartige Strukturen ergeben sich in seltenen Fällen. Dies ist hauptsächlich bei Ventilen der Fall, wo äußere Rand-

bedingungen wie z. B. Sicherheitsanforderungen dieses bedingen. Der Grund für die geringe Häufigkeit der nichtbaumartigen Strukturen sind die im Vergleich höheren Kosten.

Eine Baumstruktur bedeutet eine serielle Schaltung der lokalen Komponentenbeschreibungen T_C . Bei dem Fall der nichtbaumartigen Struktur ergeben sich dagegen parallele und serielle Schaltungen. Da bei nichtbaumartigen Strukturen überwiegend Parallelschaltungen auftreten, werden bei der Modelltiefenzuordnung – nicht aber bei der Modellerstellung – die vernetzten Strukturanteile (siehe Anhang A) vernachlässigt und wie parallele Strukturen behandelt.

Komponenten nichtbaumartiger Strukturen müssen vom Verarbeitungsalgorithmus zunächst erkannt und gesondert vorbehandelt werden, um dann wieder wie – und gemeinsam mit – Komponenten der baumartigen Struktur verarbeitet zu werden. Die vom gewählten Simulationszweck SI_{gew} abhängenden Vorbehandlungsmaßnahmen, die von den anderen Einflußfaktoren abhängen und auf diese einwirken, werden bei den jeweiligen Einflußfaktoren erläutert.

Mittelwert der relativen Modelltiefen \overline{MD}_{rel}

Der Mittelwert \overline{MD}_{rel}

$$\overline{MD}_{rel} = \frac{1}{w} \sum_{j=1}^w \frac{MD_{gew,j}}{|\mathbb{MD}_j|} , \quad (3.2)$$

wird aus dem arithmetischen Mittel aller bisher bestimmten relativen Modelltiefen $\frac{MD_{gew}}{|\mathbb{MD}|}$ bestimmt, wobei w die Anzahl der bisher bestimmten Modelltiefen ist. Da die Verrechnung des Mittelwertes der relativen Modelltiefen bei der Modelltiefenbestimmung immer gemeinsam mit der Verrechnung der Frequenz ω geschieht, wird die Verrechnungsvorschrift (3.6) im Abschnitt 3.3 zusammen mit der der Frequenz angegeben.

Eigenfrequenz ω_o

Die Eigenfrequenz tritt bei der Modellreduktion bezüglich der Abschneidefrequenz und bei der Modelltiefenbestimmung in Erscheinung.

Bei der Modellreduktion werden die dynamischen Anteile aller Komponenten, für die

$$\min(\omega_o) \geq \omega_{cut} , \quad (3.3)$$

gilt, vernachlässigt. Bild 3.5 zeigt diesen Zusammenhang. Die Abschneidefrequenz ω_{cut} bestimmt sich aus der Eigenfrequenz der Komponente mit der minimalen Eigenfrequenz $\omega_{o_{min}}$ aller ausgewählten Komponenten zu

$$\omega_{cut} = \beta_{cut} \cdot \omega_{o_{min}} . \quad (3.4)$$

Die von der Anzahl der Simulationszwecke $|\mathbb{SI}|$ und dem ausgewählten Simulationszweck SI_{gew} abhängige Abschneidekonstante β_{cut} ist in Tabelle B.1 im Anhang B aufgeführt und kann vom ^{art}*deco*-Systembeauftragten geändert werden. Die Modelltiefenbestimmung erfolgt für alle Komponenten, deren Struktur baumartig ist. Falls nicht baumartige Strukturen existieren, muß erst die Vorbehandlung zur Bestimmung der Ersatzeinflußgrößen durchgeführt werden.

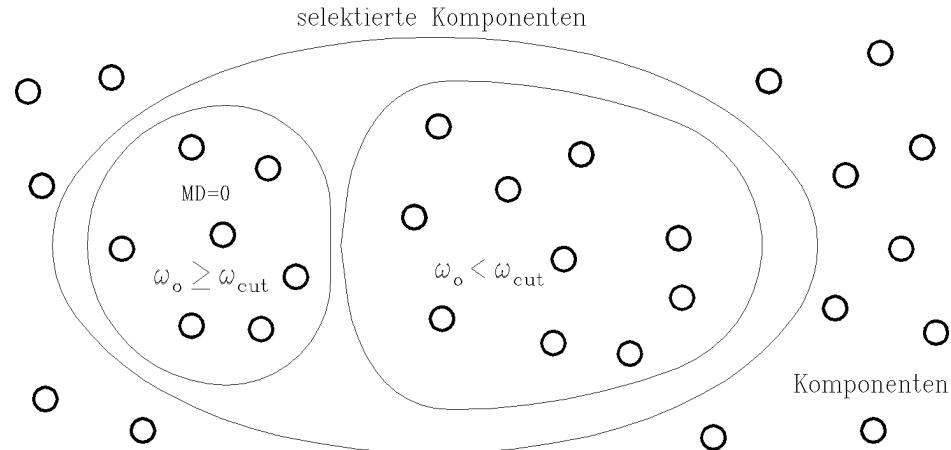


Bild 3.5: Selektierte und dynamisch vernachlässigte Komponenten

Die Eigenfrequenz ω_o bzw. die Ersatzgröße *Strukturfrequenz* ω_{struc} findet für alle gewählten Simulationszwecke

$$SI_{gew} < \lfloor \frac{1}{2} |\mathbb{S}\mathbb{I}| \rfloor \quad (3.5)$$

Berücksichtigung. Die Modelltiefe einer aktuell zu bestimmenden Komponente ist abhängig vom Mittelwert $|MD_{rel}|$ der relativen Modelltiefen und der auf die Abschneidefrequenz ω_{cut} bezogenen Eigenfrequenz ω_{top} der aktuellen Komponente. Für die Modelltiefe der aktuellen Komponente gilt:

$$MD_{top} = \neg \left(|\mathbb{M}\mathbb{D}_{top}| , \left[\kappa_\omega \left(1 - \frac{\omega_{top}}{\omega_{cut}} \right) + \kappa_{MD} \overline{MD_{rel}} \right] \right) \quad (3.6)$$

Dabei sind die κ_i die Bewertungsfaktoren der jeweiligen Einflußfaktoren.

Die Vorbehandlung nichtbaumartiger Strukturen liefert für diese Komponenten eine Strukturersatzfrequenz ω_{struc} , die anstelle der Eigenfrequenz der aktuellen Komponente ω_{top} verarbeitet wird. Die Verarbeitung der Einflußfaktoren unterscheidet sich abhängig von der Struktur nur in diesem Punkt.

Die Verstärkung K

Die Verstärkung K tritt bei der Modelltiefenbestimmung in Erscheinung. Die Vorbehandlung nichtbaumartiger Strukturen liefert – wie bei der Eigenfrequenz – eine Strukturersatzverstärkung K_{struc} , die anstelle der Verstärkung der aktuellen Komponente K_{top} bei der Modelltiefenbestimmung verarbeitet wird.

Die Verstärkung K bzw. die Strukturersatzverstärkung K_{struc} einer Komponente wird für

$$\lfloor \frac{1}{2} |\mathbb{S}\mathbb{I}| \rfloor \leq SI_{gew} < \lfloor \frac{2}{3} |\mathbb{S}\mathbb{I}| + 1 \rfloor \quad (3.7)$$

berücksichtigt. Für die Modelltiefe der aktuellen Komponente MD_{top} wird die Zuordnungsvorschrift (3.6) noch um die auf die Maximalverstärkung K_{max} normierte Verstärkung der aktuellen Komponente K_{top} und dem zugehörigen Bewertungsfaktor κ_K ergänzt. Es gilt

$$MD_{top} = \neg \left(|\mathbb{M}\mathbb{D}_{top}| , \left[\kappa_\omega \left(1 - \frac{\omega_{top}}{\omega_{cut}} \right) + \kappa_K \frac{K_{top}}{K_{max}} + \kappa_{MD} \overline{MD_{rel}} \right] \right) . \quad (3.8)$$

Die Systemversalität v

Analog zur Verstärkung liefert die Vorbehandlung nichtbaumartiger Strukturen eine Strukturersatzsystemversalität v_{struc} , die anstelle der Systemversalität der Komponente v_C bei der Modelltiefenbestimmung verarbeitet wird. Dieser Einflußfaktor wird in Abhängigkeit des gewählten Simulationszweckes SI_{gew} für

$$\lfloor \frac{2}{3}(|\mathbb{SI}| + 1) \rfloor \leq SI_{gew} < |\mathbb{SI}| \quad (3.9)$$

berücksichtigt. Die Zuordnungsvorschrift (3.8) wird um die auf die maximale Systemversalität v_{max} normierte Systemversalität v_{top} der aktuellen Komponente und um den Bewertungsfaktor κ_v zu

$$MD_{top} = \neg \left(|\mathbb{MD}_{top}|, \left[\kappa_\omega \left(1 - \frac{\omega_{otop}}{\omega_{cut}} \right) + \kappa_K \frac{K_{top}}{K_{max}} + \kappa_v \frac{v_{top}}{v_{max}} + \kappa_{MD} \overline{MD_{rel}} \right] \right) \quad (3.10)$$

ergänzt.

Die Nichtlinearität NL

Für die Nichtlinearität erzeugt die Vorbehandlung nichtbaumartiger Strukturen (*TS is .false.*) ebenfalls eine Strukturersatznichtlinearität NL_{struc} . Diese kann dann wie die Nichtlinearität der Komponente NL_C bei der Modelltiefenbestimmung verarbeitet werden. Berücksichtigung bei der Modelltiefenbestimmung findet die Nichtlinearität, wenn die gewählte Modelltiefe mit der Anzahl der Modelltiefen übereinstimmt:

$$SI_{gew} = |\mathbb{SI}|. \quad (3.11)$$

Die Bestimmungsvorschrift (3.10) wird um die Nichtlinearität NL_{top} der aktuellen Komponente und um den Bewertungsfaktor κ_{NL} erweitert. Somit folgt:

$$MD_{top} = \neg \left(|\mathbb{MD}_{top}|, \left[\kappa_\omega \left(1 - \frac{\omega_{otop}}{\omega_{cut}} \right) + \kappa_K \frac{K_{top}}{K_{max}} + \kappa_v \frac{v_{top}}{v_{max}} + \kappa_{NL} NL_{top} + \kappa_{MD} \overline{MD_{rel}} \right] \right). \quad (3.12)$$

Auf die Bestimmung der Nichtlinearität wird in (Vier 1994) detailliert eingegangen.

Algorithmus der Modelltiefenbestimmung

Um die eigentliche Modelltiefenzuordnung durchzuführen, muß zunächst eine initiale Modelltiefenzuordnung für eine erste Komponente stattfinden. Dabei wird aus der Menge aller selektierten Komponenten, für die $\omega_o < \omega_{cut}$ gilt, die Komponente mit der maximalen Eigen- bzw. Strukturfrequenz⁴ ω_{max} bestimmt. Deren Modelltiefe MD ist direkt von dem gewählten Simulationszweck SI_{gew} abhängig:

$$MD_{top} = \neg \left(|\mathbb{MD}_{top}|, \left[\frac{SI_{gew}}{|\mathbb{SI}|} \right] \right). \quad (3.13)$$

⁴Sollten mehrere Komponenten ω_{max} besitzen, so werden aus diesen nacheinander die Arbeits-, dann die Steuer- und danach die Versorgungselemente ausgewählt. Sollten immer noch mehr als eine Komponente vorhanden sein, wird die erste in der Liste stehende gewählt.

Für alle weiteren Komponenten wird die Modelltiefe nach den Zuordnungsvorschriften (3.6) bis (3.12) in Abhängigkeit der Einflußfaktoren und des gewählten Simulationszweckes SI_{gew} bestimmt (Bild 3.6).

Die beschriebene lokale Modelltiefenzuordnung in Abhängigkeit des Simulationszweckes sowie der genannten Einflußfaktoren wird auf alle selektierten Komponenten angewendet. Durch die sich anschließende Unifizierung der Systemzustände der lokalen Modelle wird das globale Modell gebildet. Im einzelnen läßt sich der in Bild 3.7 dargestellte Algorithmus beschreiben.

Die Eingangsgrößen des Algorithmus zur Modelltiefenbestimmung sind die Menge der selektierten Komponenten \mathbb{LC}_{sel} , der gewählte Simulationszweck SI_{gew} und die Anzahl der Simulationszwecke $|\mathbb{SI}|$. Mit diesen Informationen wird ein ω_{cut} bestimmt, wobei

$$MD(LC) = 0 \quad \forall \quad LC \in \mathbb{LC}_{sel} \quad \text{mit} \quad \omega_o(LC) \geq \omega_{cut} \quad (3.14)$$

gilt. Diese Teilmenge der Komponenten \mathbb{LC}_{sel} wird also hinsichtlich ihres zeitverzögernden Verhaltens vernachlässigt (Bild 3.5). Daran anschließend wird für alle Komponenten, die sich in nichtbaumartigen Strukturen befinden (*TS is .true.*), die Strukturvorverarbeitung durchgeführt.

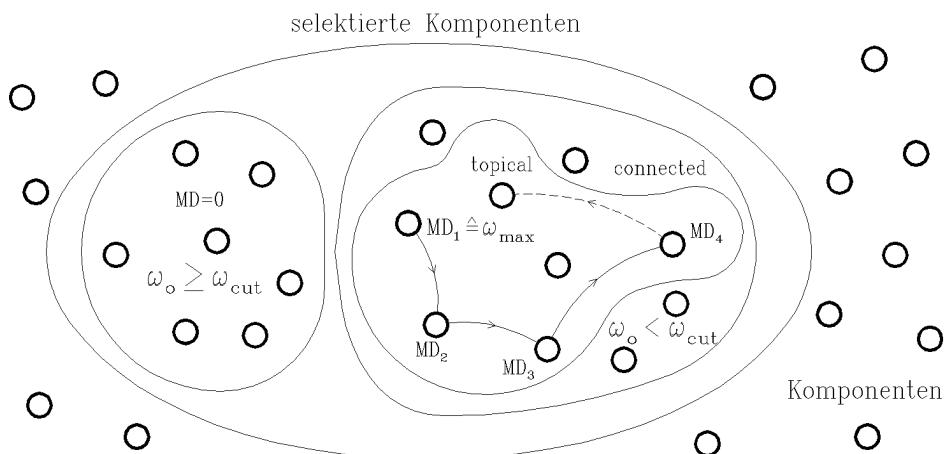


Bild 3.6: Modelltiefenbestimmung

Die Strukturvorverarbeitung liefert in Abhängigkeit des gewählten Simulationszweckes SI_{gew} und der Anzahl der Simulationszwecke $|\mathbb{SI}|$ die Strukturersatzfrequenz ω_{struc} , die Strukturersatzverstärkung K_{struc} , die Strukturersatzversalität v_{struc} und die Strukturersatznichtlinearität NL_{struc} . Diese Ersatzeinflußfaktoren werden dann analog zu den lokalen Einflußfaktoren der Komponenten in baumartigen Strukturen (*TS is .true.*) verarbeitet. Die Ersatzsystembehandlung wird eingehend durch Vier (1994) erläutert.

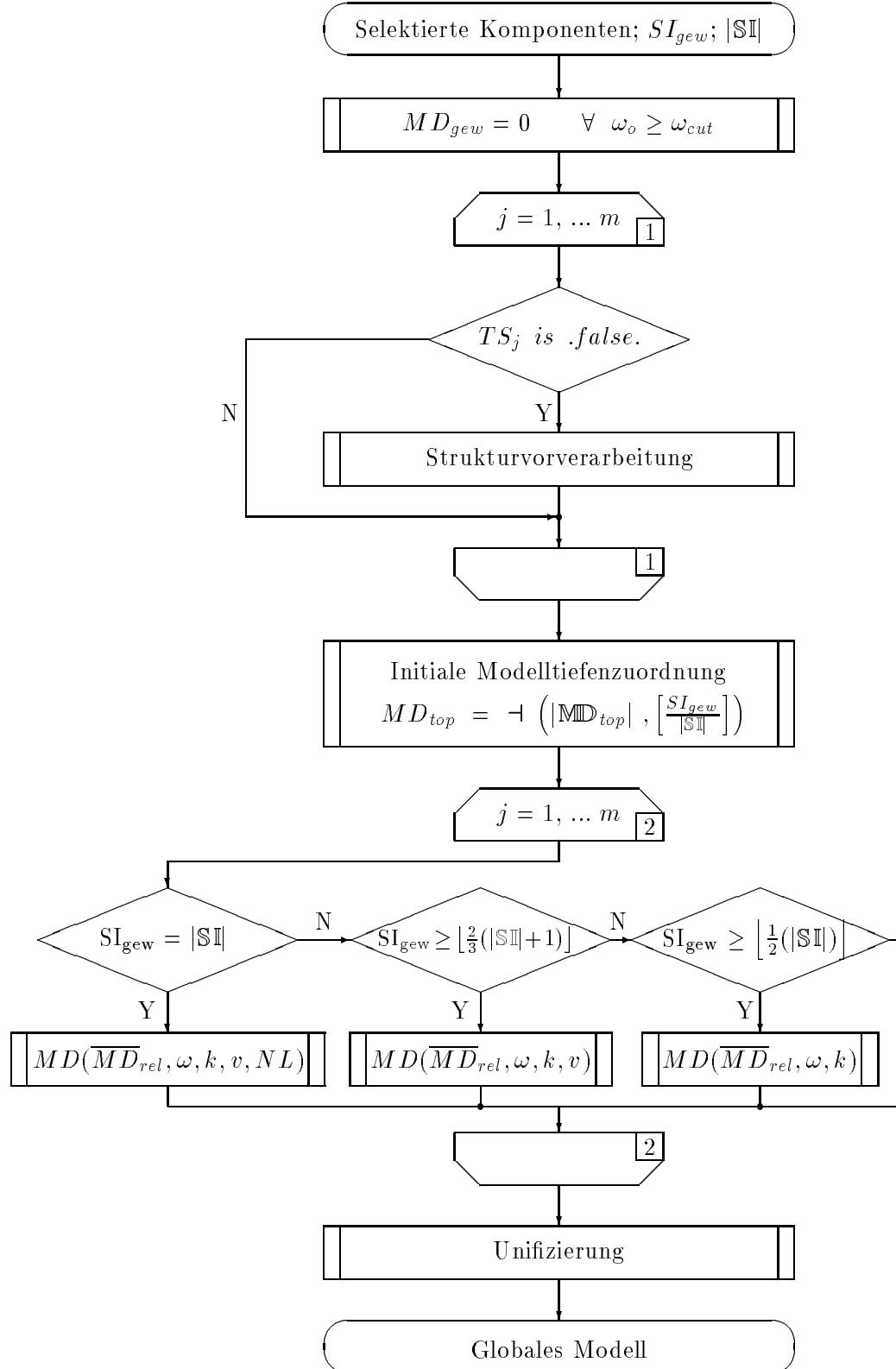


Bild 3.7: Vereinfachter Ablaufplan der Modelltiefenbestimmung

3.4 Auswahl des Simulationsverfahrens

Sollte der Anwender bei in der Praxis üblichen Simulationsprogrammen überhaupt die Möglichkeit haben, zwischen verschiedenen Verfahren zur numerischen Integration des zu lösenden Differentialgleichungssystems auswählen zu können, so erhält er üblicherweise weder bei der Auswahl, noch bei der Festlegung der Verfahrensparameter Unterstützung. Bei den meisten Anwendern ist bezüglich der problemangepaßten Auswahl der numerischen Integrationsverfahren sowie der zugehörigen Parameter (z. B. Fehlerschranken und Schrittweiten) kein oder nur geringes Wissen vorhanden. Dem steht gegenüber, daß in der Mathematik umfangreiches Wissen bezüglich dieser Problemfelder existiert, das dem Anwender geeignet zur Verfügung gestellt werden sollte. Erste Schritte in ähnliche Richtungen gibt Krauth (1989) zusammenfaßend wieder.

Für die Ermittlung des dynamischen Verhaltens eines definierten Modells sollen im folgenden nur numerische Verfahren betrachtet werden. Andere Verfahren, wie sie beispielhaft von Fischwick und Luker (1991) dargestellt sind, behandelt dieser Bericht nicht.

Allgemein unterteilt man die verwendeten numerischen Integrationsverfahren in

- Ein- und Mehrschrittverfahren

sowie in

- implizite und explizite Verfahren.

Bei der numerischen Integration wird prinzipiell von den bekannten Zuständen $Z(t_j)$ mit einer Integrationsschrittweite h auf die unbekannten Zustände $Z(t_{j+1})$ geschlossen. Für ein Einschrittverfahren werden nur die zuletzt berechneten Zustände $Z(t_j)$, für ein Mehrschrittverfahren werden dagegen auch ältere Zustände bei der Berechnung der unbekannten Zustände $Z(t_j)$ verwendet.

Ein implizites Verfahren ist dadurch gekennzeichnet, daß es zur Bestimmung der noch unbekannten Zustände $Z(t_j)$ auch diese selbst verwendet. Dieses bedingt eine Iteration, womit direkt einsichtig ist, daß iterative Verfahren in der Regel aufwendiger sind als explizite Verfahren, die die noch unbekannten Zustände $Z(t_j)$ bei deren Bestimmung nicht miteinbeziehen. Eine Kombination dieser beiden Vorgehensweisen stellen die Prädiktor–Korrektorverfahren dar. Tiefergehende Beschreibungen dieser mathematischen Standardverfahren finden sich u. a. in (Engeln–Müllges und Reutter 1988), (Schmidt 1980), (Clemens 1994) sowie (Suermann 1994).

Die Wahl des numerischen Integrationsverfahrens hängt von dem zu simulierenden System sowie von den an das Simulationsergebnis gestellten Anforderungen ab. Hierbei muß eine kritische Fehlerbetrachtung vorgenommen werden. Die Simulation des Verhaltens eines Systems ist grundsätzlich fehlerbehaftet. Der Gesamtfehler läßt sich nach Bauknecht, Kohlas und Zehnder (1976) in die folgenden Einzelfehler untergliedern, wobei gewährleistet sein sollte, daß diese sinnvoll aufeinander abgestimmt sind und der Aufwand dem Nutzen bei einer Reduktion des Gesamtfehlers vertretbar ist.

- Systemfehler:

Dieser Fehler ist ein typischer Analysefehler eines Menschen, der die Modellbildung eines Systems vorzunehmen hat. Die Erfassung und Untergliederung des Systems sowie die Beschreibung ihrer Beziehungen untereinander und zur Umwelt sind verfälscht. Hier tritt der Vorteil einer synthetisierenden Modellbildung, wie bei *^{art}deco*, im Vergleich zur analysierenden zu Tage. Wird ein Gesamtsystem aus Teilsystemen komponiert, die für sich beschreibbar und damit auch besser validierbar sind, so entfällt dieser Fehler praktisch.

- Modellfehler:

Fehler in der Modellierung, die dazu führen, daß das Verhalten des Modells nicht dem des Originalsystems entspricht, sind bei komplexen Systemen sehr schwer aufzufinden. Abhilfe kann nur eine sorgfältige Modellierung und Validierung bieten. Der damit verbundene Aufwand wächst überproportional mit der Komplexität der zu untersuchenden Systeme. Das Verhältnis von Nutzen zu Aufwand ist dann (wie bei *^{art}deco*) optimal, wenn die Modelle möglichst kleiner Teilsysteme häufig verwendet werden können.

- Datenfehler:

Sind die Werte von Parametern unkorrekt, so bedeutet die Beseitigung dieser Fehler meistens die Notwendigkeit von Messungen, welche teilweise sehr großen Aufwand verursachen, der jedoch durch die oben genannten Gründe gerechtfertigt wird. Vorbelegte Richtwerte für Komponentenklassen und praxisübliche Wertebereiche der betreffenden Parameter, stellen hier schon eine wesentliche Hilfe dar, die krasse Fehler vermeiden hilft.

- Rechenfehler:

Die Rechenfehler eines numerischen Verfahrens setzen sich aus Rundungs- und Abbruchfehlern zusammen, auf die im folgenden noch genauer eingegangen wird. Die Rechenfehler sind im wesentlichen von den Kombinationen des zu simulierenden Modells, des gewählten Verfahrens und der geforderten Genauigkeit abhängig.

- Interpretationsfehler:

Da die Bewertung der Simulationsergebnisse durch den Menschen geschieht, ist eine Fehlerreduktion in diesem Bereich nur eingeschränkt möglich. Neben einer Sensibilisierung des Anwenders, daß und in welchen Größenordnungen mit einem fehlerbehafteten Ergebnis zu rechnen ist, kann eine Unterstützung nur durch zusätzliche Informationen (wie z. B. statische Werte oder praxisübliche Wertebereiche) erfolgen. Im Unterabschnitt 3.5 wird dieser Problembereich eingehender behandelt.

Bei der Auswahl des Simulationsverfahrens sind die Rechenfehler maßgeblich.

Bei der Simulation eines hydrostatischen Systems handelt es sich um die Lösung eines Anfangswertproblems von einem Modell, das gemäß Definition 2.5 durch ein gewöhnliches Differentialgleichungssystem der Form

$$\dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{z}(t)) \quad , \quad \text{mit } \mathbf{z}(t_0) = \mathbf{z}_0 \text{ und } (\mathbf{z}, \dot{\mathbf{z}}, t) \in \mathbb{R}^{\dim(\mathbf{z})} \quad , \quad (3.15)$$

beschrieben ist (vgl. u. a. Suermann 1994).

Um dieses Problem mit Hilfe der numerischen Integration zu lösen, müssen sowohl die Elemente als auch die Komposition der Elemente Modell, Verfahren und Verfahrensparameter den verschiedenen nachfolgenden Bedingungen genügen, die für den Rechenfehler relevant sind. Dieses sind

- die Lipschitz–Bedingung und Stabilität des Anfangswertproblems,
- der Rechenfehler bei einem Integrationsschritt,
- die Stabilität des Integrationsverfahrens,
- die Stetigkeit und
- die Nichtlinearität,

wobei hier die Stetigkeit wesentlich ist.

Für die Existenz und Eindeutigkeit, der Lösung des Anfangswertproblems ist die **Lipschitz–Bedingung** für das Differentialgleichungssystem nach (3.15) eine hinreichende Bedingung. Sie lautet

$$\|\mathbf{g}(t, \mathbf{z}_1(t)) - \mathbf{g}(t, \mathbf{z}_2(t))\| \leq L \|\mathbf{z}_1(t) - \mathbf{z}_2(t)\| \quad (3.16)$$

mit $((t, \mathbf{z}_1(t)), (t, \mathbf{z}_2(t)), L) \in \mathbb{R}$.

Die Lipschitz–Bedingung wird ausführlich in mathematischer Standardliteratur, wie z. B. (Engeln–Müllges und Reutter 1988) erläutert. Eine weitere wichtige Voraussetzung, um eine sinnvollere Lösung des Anfangswertproblems zu erlangen, ist die **Stabilität**, wobei zwischen der Stabilität des Systems und der des numerischen Verfahrens unterschieden wird. Wie in der Systemtheorie allgemein bekannt (z. B. Föllinger 1992), ist ein System dann stabil, wenn alle seine Eigenwerte negativ sind:

$$\operatorname{Re}(\lambda(t, \mathbf{z}(t))) < 0 \quad \forall (t, \mathbf{z}(t)) \in \mathbb{R} \quad . \quad (3.17)$$

Suermann (1994) beschreibt diese Zusammenhänge detaillierter und gibt auch ein Beispiel eines numerisch nicht lösbarer, zwar Lipschitz–beschränkten, aber instabilen Anfangswertproblems.

Der **Rechenfehler bei einem Integrationsschritt** wird wesentlich durch die Integrations schrittweite h bestimmt. Er setzt sich im wesentlichen aus dem Verfahrens- und dem Rundungsfehler, die in Bild 3.8 qualitativ über der Schrittweite aufgetragen sind, zusammen. Der *Rundungsfehler* ist maßgeblich von der Darstellungsgenauigkeit des Digitalrechners und von der Größenordnung der dargestellten Zahl abhängig. Werden die Änderungen der Zahlenwerte eines Zustands infolge der Verringerung der Schrittweite immer kleiner, so wird die bei „großen“ Zahlen zu vernachlässigende Rundungsgenauigkeit maßgebend.

Die hier verwendeten Verfahren der numerischen Integration beruhen darauf, für die nicht–bekannte oder existente exakte Lösungsformel eine Näherungsformel so zu finden, daß die

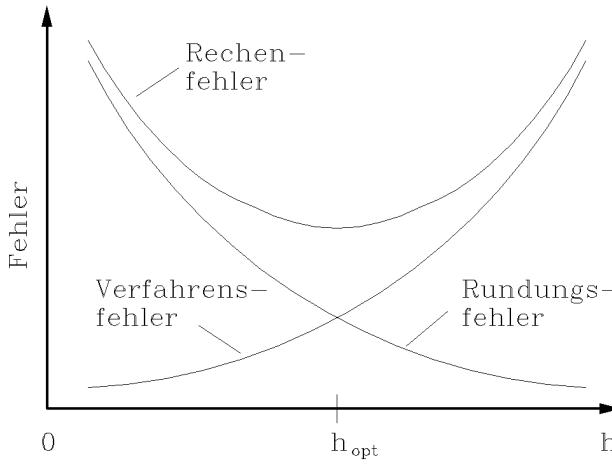


Bild 3.8: Fehler bei einer numerischen Simulation (nach Clemens 1994)

Näherungslösung die exakte Lösung hinreichend genau wiedergibt. Die Abweichung zwischen den Lösungen ist der *Verfahrensfehler*, welcher natürlich mit steigenden Schrittweiten, und damit zunehmendem Abstand der Stützpunkte der Näherungslösung, zunimmt.

Für eine optimale Schrittweite h_{opt} ergibt sich ein lokales Minimum des Rechenfehlers (Bild 3.8). Diese ist u. a. vom verwendeten Digitalrechner, dem numerischen Verfahren und der Lösung selbst abhängig, womit eine Vorausbestimmung schwierig und ineffizient ist. Aus der Praxis (z.B. Clemens 1994, Suermann 1994) ist bekannt, daß die optimale Schrittweite im Hinblick auf die Rechengeschwindigkeit sehr klein ist. So ist statt der Bestimmung der optimalen Schrittweite eine Abschätzung der maximalen Schrittweite zu

$$h_{max} = \frac{0,1}{\omega_{o \ max}} \quad (3.18)$$

praxisüblich. Des Weiteren ist es nicht sinnvoll, mit einer festen Schrittweite zu rechnen, sondern diese dem Verlauf der Lösung anzupassen. Detaillierte Ausführungen zu Schrittweitensteuerungen für explizite und implizite Verfahren finden sich u. a. in (Suermann 1994).

Nach Clemens (1994) läßt sich die **Stabilität eines Integrationsverfahrens** wie folgt beschreiben:

Um eine stabile Simulation zu gewährleisten, müssen auftretende Fehler über den Simulationszeitraum abklingen:

$$\lim_{t \rightarrow t_{end}} < |\beta| \quad (3.19)$$

mit $\beta \in \mathbb{R}$ beliebig, aber beschränkt.

Ist der exakte Verlauf der Lösung $\mathbf{z}(t)$ einer Differentialgleichung mit $\mathbf{z}_j \hat{=} \text{I}$ und $\mathbf{z}_{j+1} \hat{=} \text{II}$ bekannt, läßt sich die Fehlerentwicklung in Bild 3.9 darstellen. Zu dem Zeitpunkt t_j sei III ein durch ein numerisches Integrationsverfahren berechneter fehlerbehafteter Wert \mathbf{z}_j . Bei dem Integrationsschritt von t_j nach t_{j+1} ergibt sich mit einer exakten Integration der Zustand IV,

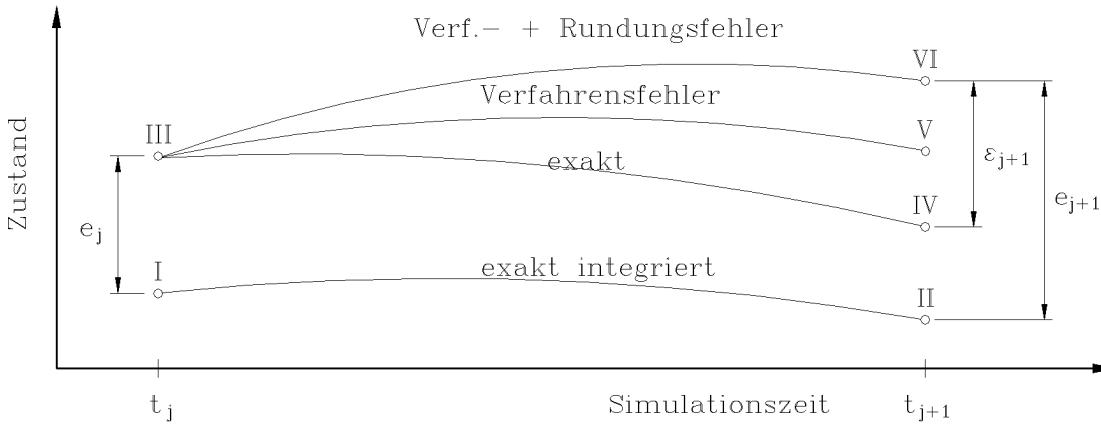


Bild 3.9: Fehlerentwicklung bei der numerischen Simulation

mit einem entsprechenden Verfahrensfehler V und unter zusätzlicher Berücksichtigung des Rundungsfehlers VI.

Bei t_{j+1} setzt sich der globale Fehler

$$e_{j+1} = M(\lambda h_j) e_j + \varepsilon_{j+1} \quad (3.20)$$

aus der Fehlerfortpflanzung $M(\lambda h_j) e_j$ und dem neuen lokalen Fehler ε_{j+1} zusammen. $M(\lambda h_j)$ bezeichnet die diskrete Übertragungsfunktion eines beliebigen Integrationsverfahrens in Abhängigkeit von den Eigenwerten λ , der Jakobimatrix des Systems und der gewählten Schrittweite h_j . Für alle Werte λh_j , die die Bedingung

$$|M(\lambda h_j)| < 1 \quad (3.21)$$

unter der Voraussetzung $\text{Re}(\lambda) < 0$ erfüllen, liegt absolute Stabilität vor. Beinhaltet das Stabilitätsgebiet die gesamte linke λh -Ebene, wird das Verfahren als A-stabil bezeichnet. Diese Verfahren haben für jede Schrittweite abklingende Eigenbewegungen des globalen Fehlers (3.20). Weitergehende Erläuterungen zur Stabilität des Integrationsverfahrens finden sich u. a. bei Schmidt (1980), Engeln-Müllges und Reutter (1988), Suermann (1994) und Clemens (1994).

Von **steifen Systemen** spricht man, wenn für den Steifheitsgrad (Definition 2.18)

$$S = \left| \frac{\omega_{o \max}}{\omega_{o \min}} \right| \gg 1 \quad (3.22)$$

gilt. Ein steifes System besitzt also zumindest einen hochfrequenten Systemanteil, dessen numerische Behandlung eine geringe Schrittweite erfordert, und einen niederfrequenten Systemanteil, der eine hohe Schrittweite bedingt. Aus diesem Grunde haben numerische Verfahren für steife Systeme A-stabil zu sein (Engeln-Müllges und Reutter 1988), womit implizite Verfahren zur Anwendung kommen. In der Praxis wird häufig der Steifheitsgrad $S = 100$ als markante Grenze betrachtet, an und ab der die Integrationsverfahren für steife Systeme (Engeln-Müllges und Reutter 1988) eingesetzt werden. Bei deutlich kleineren Steifheitsgraden, sind explizite Verfahren noch ausreichend effizient. Da in der Praxis der hochfrequente Anteil von steifen

Systemen häufig gegenüber den ohnehin vorhandenen Modellierungsfehlern der niederfrequenten Systemanteile vernachlässigt werden kann, spielen diese Verfahren in der Simulation dieser Systeme normalerweise eine untergeordnete Rolle.

Dem Einfluß von **Nichtlinearitäten**, wobei hier gerade auch die Stetigkeit hervorgehoben werden soll, findet leider noch recht geringe Beachtung. Bei geregelten oder gesteuerten hydrostatischen Systemen sind hebar unstetige (2.3) und/oder nicht stetig differenzierbare Zustandsverläufe relevant. Damit fallen die Mehrschrittverfahren für diese Anwendung praktisch aus (vgl. Clemens 1994). Sollten die genannten Unstetigkeiten häufig auftreten, so bieten Spezialverfahren mit intelligenter Schrittweitensteuerung (Kürten 1993) die exakte Verrechnung dieser Unstetigkeiten.

Auswahlvorschriften

Aus den vorangegangenen Erläuterungen lassen sich die folgenden Auswahlvorschriften für Verfahren zur numerischen Integration von Anfangswertproblemen hydrostatisch mechatronischer Systeme, wie sie im Programm paket *atdeco* behandelt werden, bestimmen. Prinzipiell entfallen aufgrund des üblicherweise variablen bzw. unstetigen Zustandsverlaufes die Mehrschrittverfahren als nicht zweckmäßig für diese Aufgabe (vgl. u. a. Schmidt 1980). Werden als Verfahren Runge–Kutta oder davon abgeleitete Verfahren eingesetzt, so kann aufgrund von Praxiserfahrungen (Suermann 1994) eine gute Abschätzung der Ordnung des einzusetzenden Verfahrens bezüglich der Fehlerschranke e_{bound} gemacht werden. Auf diesen Punkt wird im Bereich der Auswahl der Verfahrensparameter, in dem auch die Fehlerschranke behandelt wird, im Anschluß an die Bereiche *Steifheit* und *Nichtlinearität* eingegangen.

Auswahl bezüglich Steifheit:

Wenn ein System steif ist, ist der Einsatz impliziter Integrationsverfahren effizienter als der von expliziten Integrationsverfahren. Die einfachste Entscheidung, ob ein System als steif anzusehen ist, läßt sich derart finden, daß der Steifheitsgrad (vgl. Definition 2.18) mit einem festen Entscheidungswert EW (hier z. B. 100) verglichen wird. Damit ist eine Betrachtung als steife Systeme nur für diese erforderlich, bei deren automatisierter Modellerstellung für den Abschneidewert β_{cut} (vgl. Anhang B)

$$\beta_{cut} \geq EW \quad (3.23)$$

gilt. Auf alle Systeme, die diese Bedingung nicht erfüllen, sind explizite Verfahren anzuwenden. Bei der noch notwendigen Bestimmung der Verfahrensparameter der die steifen Systeme behandelnden impliziten Verfahren ist bezüglich der Schrittweite h hier schon eine Einordnung anhand der Steifheit möglich. Aufgrund der deutlich unterschiedlichen Eigenfrequenzen sollte die maximale Eigenfrequenz $\omega_{o\ max}$ in der Beziehung (3.18) durch eine Bezugseigenfrequenz $\omega_{o\ bezug}$, die außer der Steifheit S auch den Simulationszweck $|SI|$ bzw. SI_{gew} in ihrer Bildungsvorschrift mit

$$\omega_{o\ bezug} = \omega_{o\ min} + \frac{(S - 1) \omega_{o\ min}}{\exp(1) - \exp(|SI|)} (\exp(1) - \exp(SI_{gew})) \quad (3.24)$$

berücksichtigt, ersetzt werden. Eine elegantere Entscheidung darüber, ob ein System als steif betrachtet werden kann, besteht in dem Vergleich des geschätzten Aufwandes für implizite und für explizite Verfahren. Wenn der geschätzte Aufwand für ein implizites Verfahren geringer ist als für ein explizites, so wird das zu untersuchende System als steif betrachtet. Diese Aufwandsabschätzungen, welche für die Ordnungsbestimmung der Verfahren ohnehin durchgeführt werden, sind für explizite und implizite Runge–Kutta–Verfahren ausführlich in (Suermann 1994) beschrieben. Ist eine Entscheidung bezüglich der Steifheit gefallen, so müssen die Verfahrensparameter bestimmt werden, was nachfolgend im Text behandelt wird.

Auswahl bezüglich Nichtlinearität:

Stetig nichtlineares Verhalten stellt für die Mehrzahl der numerischen Integrationsverfahren kein prinzipielles Problem dar. Anders dagegen ist nicht stetiges bzw. nicht stetig differenzierbares Verhalten zu bewerten. Hierbei führt eine Verletzung der Stetigkeitsvoraussetzung für \mathbf{g} (aus Gl. 3.16) zu einer lokalen Verletzung der Lipschitz–Bedingung (3.16). Trotzdem können – wie die praktische Erfahrung zeigt – zumindest hebbare unstetige Zustandsverläufe mit numerischen Verfahren behandelt werden. So lassen sich solche Systeme mit Hilfe expliziter Verfahren und intelligenter Schrittweitensteuerung (Kürten 1993) trotz lokaler Verletzung der Lipschitz–Bedingung exakt verarbeiten. Ist der Zeitpunkt bzw. der durch einen Systemzustand definierte Zeitpunkt des Auftretens der Nichtlinearität sowie deren Zustandsverlauf bekannt, was für technische Systeme dieser Anwendungsdomäne in der Regel der Fall ist, so wird eine Verfahrensschrittweite h derart bestimmt, daß der genannte Zeitpunkt exakt erreicht wird und kein zusätzlicher Fehler durch die Quasivernachlässigung der Unstetigkeit auftritt. Der sinnvolle Einsatz eines derartigen Spezialverfahrens ist mit vertretbarem Aufwand rechnerisch nicht vorauszubestimmen, weshalb er über eine Nichtlinearitätsschranke NL_{bound} abgeschätzt wird.

Auswahl der Verfahrensparameter:

Neben dem Verfahren selbst hat der Anwender auch die zugehörigen Parameter zu bestimmen. Während die Koeffizienten der Verfahren (Suermann 1994) mit der Wahl des anzuwendenden Verfahrens gegeben sind, bleiben üblicherweise als noch unbekannte Parameter

- die Fehlerschranke ϵ_{bound} ,
- die maximale Schrittweite h_{max} und
- die minimale Schrittweite h_{min} .

Die maximale Schrittweite ist nach (3.18) für nicht steife Systeme zu $h_{max} = \frac{0,1}{\omega_{o max}}$ und für steife Systeme zu $h_{max} = \frac{0,1}{\omega_{o bezug}}$ gegeben. Eine weitere Randbedingung für die Wahl der maximalen Schrittweite ist die Bildschirmschärfe. Eine gute Bildschirmschärfe ergibt sich ausgehend von der praxisüblichen Bildschirmauflösung und dem Simulationszeitraum t_{sim} , für die maximale Schrittweite von

$$h_{max} = \frac{t_{sim}}{640} . \quad (3.25)$$

Die Fehlerschranke e_{bound} wird maßgeblich davon bestimmt, wie genau das Modell ist und welche Genauigkeit beim Ergebnis benötigt wird. Da die Modelltiefe ebenfalls über den gewählten Simulationszweck SI_{gew} und $|SI|$ bestimmt ist, werden die Fehlerschranken e_{bound} in Abhängigkeit von diesen angegeben. Die Default-Werte der Fehlerschranken enthält Anhang B.

Die minimale Schrittweite h_{min} ist sowohl von der maximalen Schrittweite h_{max} als auch von der Fehlerschranke e_{bound} abhängig. Hierzu wird eine Schrittweitenkonstante β_h gebildet, die den Einfluß der Fehlerschranke repräsentiert mit

$$\beta_h = 10^{-\lfloor 2 - \frac{1}{5} \log(e_{bound}) \rfloor} . \quad (3.26)$$

Damit ergibt sich die minimale Schrittweite zu

$$h_{min} = \beta_h \cdot h_{max} . \quad (3.27)$$

Die Ordnung des Verfahrens ist eng mit der Fehlerschranke e_{bound} verknüpft. So haben praktische Tests gezeigt (Suermann 1994), daß die Fehlerordnung des Verfahrens und die positive Potenz der Fehlerschranke ungefähr die gleiche Größenordnung haben sollten. Im System $^{\text{art}}deco$ wird die Ordnung des Verfahrens durch eine Testrechnung mit einer Aufwandsabschätzung nach Suermann (1994) realisiert.

Der Gesamtvergäng der Verfahrensauswahl läßt sich im folgenden Flußdiagramm (Bild 3.10) darstellen.

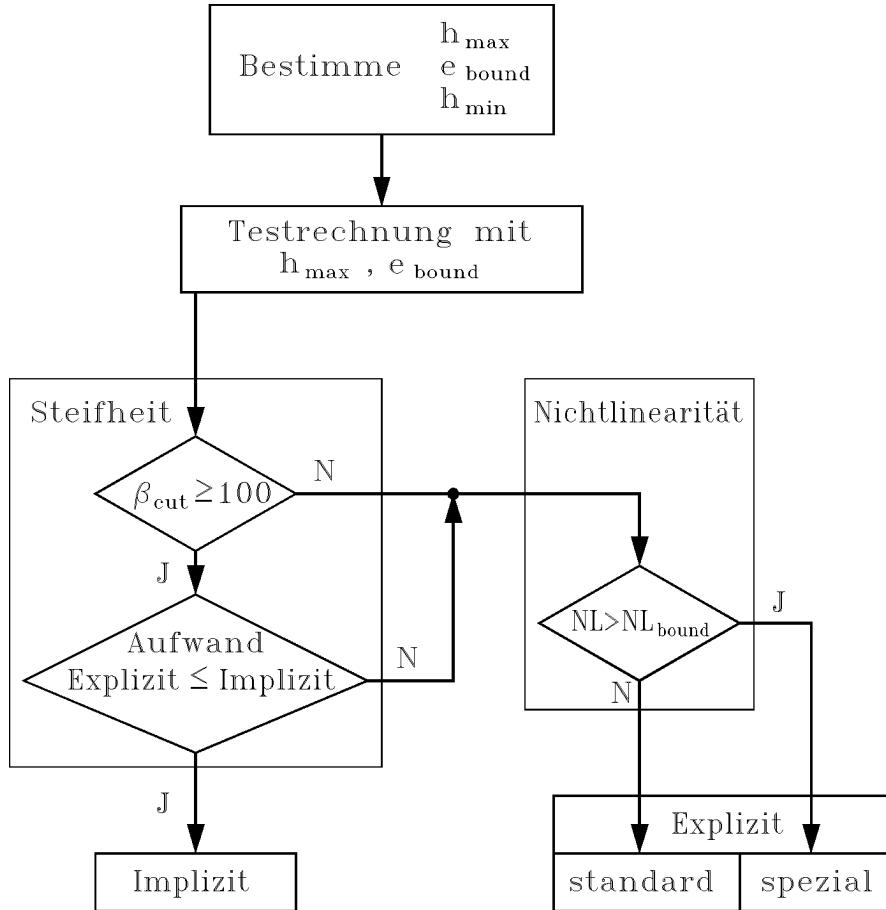


Bild 3.10: Verfahrens- und Parameterbestimmung in $^{\text{art}}deco$

Automatische Verfahrensauswahl und Überprüfung:

Im System *ardeco* findet die Bestimmung der Verfahren sowie deren Ordnung und Parameter automatisch statt. Für jeden Zeitpunkt t_j des Funktionsdiagramms (Bild 3.2) der innerhalb des Simulationszeitraums liegt, wird – wie auch zu Beginn des Simulationszeitraums – nach der Bestimmung der maximalen Schrittweite h_{max} , der Fehlerschranke e_{bound} und der minimalen Schrittweite h_{min} aus dem gewählten Simulationszweck SI_{gew} eine Testrechnung durchgeführt. Hierbei wird ausgehend von der maximalen Schrittweite und der Fehlerschranke der Aufwand aller zur Verfügung stehende implizite und explizite Standardverfahren nach Suermann (1994) bestimmt. Hieran schließt sich abhängig von Aufwand und Nichtlinearität die Auswahl der Verfahren als implizite oder explizite Standard- bzw. Spezialverfahren an. Dieses geschieht für jeden genannten Zeitpunkt t_j , was zur Folge haben kann, daß ein Simulationslauf durchaus mit verschiedenen Verfahren durchgeführt wird. Ziel ist eine möglichst schnelle Simulation bei gegebener Genauigkeit.

3.5 Bewertung

Die Bewertung der Simulationsergebnisse übernimmt üblicherweise der Anwender. Gerade bei der hiermit einhergehenden Modellvalidierung (Schmidt 1980), der Untersuchung also, ob das verwendete Modell die Realität in ausreichendem Maßstab wiedergibt, ist der Anwender auch im Hinblick auf die vielfältigen anderen Einflußfaktoren auf das Simulationsergebnis, häufig überfordert. Die abschließende Bewertung einer dynamischen Funktionsprüfung liegt eindeutig im Arbeitsbereich des Anwenders, aber dieser muß geeignet unterstützt werden.

Durch die Berücksichtigung der Ergebnisse der statischen Funktionsprüfung läßt sich eine grobe Abschätzung der Modellierungs- und Simulationsgüte bezogen auf den stationären Anteil der Simulationsergebnisse einfach durchführen. Fehler aus der Modellierung und dem numerischen Integrationsverfahren treten dabei vermischt auf und erlauben keinen direkten Rückschluß auf den Ursprung der Fehler. Ausgehend von den bekannten Terme der Verfahrens- und Rundungsfehlerabschätzung existiert für die numerischen Integrationsverfahren die Möglichkeit, diese Fehleranteile herauszurechnen. Weiter kann so auf den Fehler der dynamischen bezogen auf die statische Modellbeschreibung geschlossen werden. Dieses kann als Fehlerprotokoll dem interessierten Anwender zur Verfügung gestellt werden, was aber nur optional geschehen sollte, um den Anwender nicht durch zuviel Informationen zu verwirren.

4 Zusammenfassung und Ausblick

Der vorliegende Bericht beschreibt die dynamische Funktionsprüfung hydraulischer Anlagen mit dem wissensbasierten System *^{ar}deco*. Hierzu finden nach einer automatisierten Modellerstellung numerische Verfahren zur Integration der erstellten Differentialgleichungssysteme Verwendung.

Die dynamische Funktionsprüfung baut auf einer erfolgreichen statischen Funktionsprüfung auf.

Dieser Zustand ist im Anschluß an eine umfangreiche Begriffsdefiniton (Abschnitt 2) im Unterabschnitt 3.1 dargestellt. Der Anwender hat nun die Möglichkeit, die zu simulierenden Objekte und den Simulationszweck, also die erwünschte Qualität des Simulationsergebnisses, zu wählen. Für den Anwender soll der gesamte Vorgang möglichst einfach und weitgehend graphisch interaktiv ablaufen. Das Ziel ist eine möglichst schnelle und den Anforderungen entsprechend genaue Modellbildung und Simulation, wobei der Anwender nicht durch mathematische Probleme, die nicht zu seinem Spezialgebiet gehören, belastet werden soll. Durch wissensbasierte Techniken ist es hier möglich, Effizienz, Fehlervermeidung und einfache Bedienbarkeit miteinander zu verknüpfen.

Die Modellbildung ist automatisiert und nur vom gewählten Simulationszweck sowie den lokalen Beschreibungen der Komponenten abhängig. Die wichtigsten Einflußfaktoren der Komponenten sind hierbei die Verstärkung, die Eigenfrequenz bzw. Eigenzeitkonstante, die Systemversalität und die Nichtlinearität. Mit diesen, wie auch den Topologieinformationen, wird das mathematische Systemmodell generiert.

Nach Erstellung des Modells ist das numerische Verfahren, welches die Integration des Differentialgleichungssystems liefern soll, auszuwählen und dessen Parameter zu bestimmen. Dieser Vorgang ist ebenfalls automatisiert. Das für das gegebene Modell quasioptimale Verfahren und dessen Ordnung wird hierbei durch eine Aufwandsabschätzung bestimmt. Die Parameter lassen sich mit heuristischem Wissen (über das der Anwender genauso wie über die Verfahren häufig nicht ausreichend verfügt) in Abhängigkeit des Simulationszweckes ermitteln. Es werden je nach Steifheit des Systems implizite und explizite Einschritt-Integrationsverfahren unterschieden. Abhängig von der Höhe der Nichtlinearität finden bei den expliziten Einschrittverfahren noch Spezial- oder Standardverfahren Anwendung. Allgemein lassen sich verschiedene Ordnungen unterscheiden.

Bei der sich anschließenden Bewertung wird der Anwender durch zusätzliche Informationen unterstützt.

Weiteres Entwicklungspotential ist sowohl bei der Modellierung, als auch bei der Modellauswertung vorhanden. Bei der Modellierung sollte die Möglichkeit der Verbindung unscharfer (z.B. Wertebereiche) und scharfer Größen für Zustände und Parameter untersucht werden. Bei der Modellauswertung können sich durch Methoden der qualitativen Simulation Alternativen zur numerischen Auswertung ergeben. Die prinzipiellen Vorteile dieses Weges zeigt die statische Funktionsprüfung in *^{ar}deco*.

5 Literaturverzeichnis

- Backé, W.** 1973. *Systematik der hydraulischen Widerstandsschaltungen in Ventilen und Regelkreisen*. Mainz: Otto K. Krauskopf.
- Backé, W.** 1992a. *Grundlagen der Ölhydraulik*. Aachen: Vorlesungsumdruck RWTH Aachen. 8. Auflage.
- Backé, W.** 1992b. *Servohydraulik*. Aachen: Vorlesungsumdruck RWTH Aachen. 6. Auflage.
- Bauknecht, K., Kohlas, J. und C. A. Zehnder.** 1976. *Simulationstechnik*. Berlin: Springer Verlag.
- Berger, M.** 1993. *Aufbereitung und Strukturierung der Wissensbasis für die Prüfung und Diagnose der Arbeitselemente einer hydraulischen Anlage*. Studienarbeit MSRT. Universität – GH – Duisburg.
- Bronstein, M. und A. Semendjajew.** 1985. *Taschenbuch der Mathematik*. Berlin: VEB Verlag Leipzig.
- Clemens, D.** 1994. *Zur parallelen Simulation und Regelung großer Systeme mit Transputern*. Dissertation MSRT, Universität – GH – Duisburg. Fortschrittsbericht 8/410 Düsseldorf: VDI.
- Ebertshäuser, H. u. a.** 1973. *Grundlagen der Ölhydraulik*. Mainz: Krausskopf
- Ebner, R.** 1992. *Entwicklung eines Expertensystems zur Konfigurierung hydraulischer Fahrantriebe für Radlader* BMFT Abschlußbericht. MSRT. Universität – GH – Duisburg.
- Engeln-Müllges, G. und F. Reutter.** 1988. *Formelsammlung zur numerischen Mathematik*. Mannheim: BI Wissenschaftsverlag.
- Ewald, R.** 1989. *Der Hydrauliktrainer Band 2*. Lohr a. Main: Mannesmann Rexroth GmbH.
- Faatz, H.** u.a. 1988. *Der Hydrauliktrainer Band 3*. Lohr a. Main: Mannesmann Rexroth GmbH.
- Feigel, H.-J.** 1987a. Nichtlineare Effekte am servoventilgesteuerten Differentialzylinder. *Ö + P 31, S.42-48*
- Feigel, H.-J.** 1987b. Dynamische Kenngrößen eines Differentialzylinders. *Ö + P 31, S.138-148*
- Fischwick, P. A. und P. A. Luker.** 1991. *Qualitative Simulation, Modeling and Analysis*. New York: Springer Verlag.
- Föllinger, O.** 1992. *Regelungstechnik*. Heidelberg: Hüthig Verlag. 7. Auflage.
- Hoffmann, M.** 1993. *Algorithmen zur Verarbeitung von topologischen Informationen in Netzwerken*. Diplomarbeit Mathematik / Informatik, Universität – GH – Duisburg.

- Krauth, J.** 1989. Simulation and Artificial Intelligence: State of the Art in Germany. *Proceedings of the 3rd European Simulation Congress. Edinburgh.*
- Kürten, T.** 1993. *Modellbildung, Simulation und digitale Regelung hydraulischer Antriebe mit elektrohydraulischen Schaltventilen.* Dissertation MSRT, Universität – GH – Universität. Duisburg: F. Seidel.
- Lemmen, R.** 1991. *Akquisition und Analyse von Wissen für die Inbetriebnahme von hydraulischen translatorischen Antrieben.* Diplomarbeit MSRT, Universität – GH – Duisburg
- Lemmen, R.** 1992a. *Konfigurationsprüfung einer hydraulischen Anlage mit einem modellbasierten System.* Forschungsbericht 2/92 MSRT, Universität – GH – Duisburg.
- Lemmen, R.** 1992b. *Modellbasierte Prüfung hydraulischer Anlagen.* Forschungsbericht 7/92 MSRT, Universität – GH – Duisburg.
- Lierschaft K.** 1993. *Zur digitalen Regelung hydraulischer Stetigventile.* Dissertation MSRT, Universität – GH – Duisburg. Fortschrittsbericht 8/355. Düsseldorf: VDI.
- Matthies, K.J.** 1991. *Einführung in die Ölhydraulik.* Stuttgart: Teubner
- Reuter, H.** 1992. *Verfahren zur Untersuchung der Nichtlinearität dynamischer Systeme.* Forschungsbericht 3/92 MSRT, Universität – GH – Duisburg.
- Saffe, P.** 1986. *Optimierung servohydraulischer Antriebe für den Einsatz in Industrierobotern.* Dissertation RWTH, Aachen: IHP.
- Schmidt, G.** 1980. *Simulationstechnik.* München: Oldenbourg Verlag.
- Schmitt, A.** 1986. *Der Hydrauliktrainer Band 1.* Lohr a. Main: Mannesmann Rexroth GmbH.
- Schulte, A.** 1988a. *Grundlagen der Ölhydraulik.* Vorlesungstext MSRT, Universität – GH – Duisburg
- Schulte, A.** 1988b. *Hydraulische Regelkreise und Servosteuerungen.* Vorlesungstext MSRT, Universität – GH – Duisburg
- Schwarz, H. R.** 1988. *Numerische Mathematik.* Stuttgart: Teubner Verlag.
- Schwarz, H.** 1991. *Nichtlineare Regelungssysteme: Systemtheoretische Grundlagen.* München: Oldenbourg.
- Stein, B., M. Hoffmann und R. Lemmen** 1992. Art DeCo: A System which Assists the Checking of Hydraulic Circuits. *Wien: 10th European conference on artificial intelligence.*
- Suermann, M.** 1994. *Wissensbasierte Modellbildung und Simulation von hydraulischen Schaltkreisen.* Diplomarbeit. Praktische Informatik. Universität – GH – Paderborn.
- Svaricek, F.** 1987. *Graphentheoretische Beschreibung und Bestimmung der endlichen und unendlichen Nullstellen von linearen Mehrgrößensystemen.* Dissertation MSRT, Universität – GH – Duisburg. Fortschrittsbericht 8/135 Düsseldorf: VDI.

Symbolics Inc. 1988. *MACSYMA: Reference Manual*. Cambridge Mass.

Vier, E. 1993. *Tankdruckabhängiges Verhalten der Ventil–Differentialzylinderschaltung*. Seminarvortrag MSRT, Universität – GH – Duisburg.

Vier, E. 1994. *Automatisierte Modellerkennung, -strukturierung und -reduzierung einer hydraulischen Anlage*. Studienarbeit MSRT, Universität – GH – Duisburg.

A Zur statischen Funktionsprüfung

A.1 Erstellung des Widerstandsnetzwerkes

Für die statische Funktionsprüfung in *art deco* wird die hydraulische Anlage und deren Anforderungen in ein Widerstandsnetzwerk transformiert, wobei folgende Informationen gegeben sind:

- ein hydraulischer Schaltplan des Kreislaufes (Topologie),
- eine vollständige lokale Beschreibung der Komponenten,
- ein Zustandsdiagramm, das alle Anforderungen beinhaltet,
- Randbedingungen und sonstige Vorgaben (z. B. zulässige Geräusche).

Aus dem Hydraulikplan entsteht mit Hilfe des Zustandsdiagramms ein nichtlineares Widerstandsnetzwerk, das nur Quellen, Senken und (nichtlineare) Widerstände enthält. Bild 2.3 zeigt diesen Zusammenhang.

Es bilden sich mehrere, nur über Komponenten gekoppelte Bäume aus, die hydraulisch voneinander getrennt sind. Die in diesem Graph durch Knoten dargestellten Widerstände lassen sich bezüglich ihres Verhaltens in zwei Gruppen untergliedern:

- Druck–Volumenstromverhalten nach Bernoulli, wobei auch der Fall, daß laminare und keine turbulente Strömung vorliegt, wie turbulente Strömung behandelt wird. Die Unterschiede werden durch entsprechende R_h angepaßt. ($\Delta p = R_h Q^2$)
- sonstiges Druck–Volumenverhalten (z. B. Druckbegrenzungsventile).

A.2 Detektion nicht propagierbarer Substrukturen

Eine reine Baumstruktur (Bild 2.2) läßt sich sofort mit Hilfe der Propagierung (Lemmen 1992) lösen. Es existieren Strukturen, die sich dieser Lösungsmethode entziehen. Das sind alle Strukturen, bei denen ein Knoten sich selbst erreichen kann, ohne Quellen oder Senken zu erreichen.

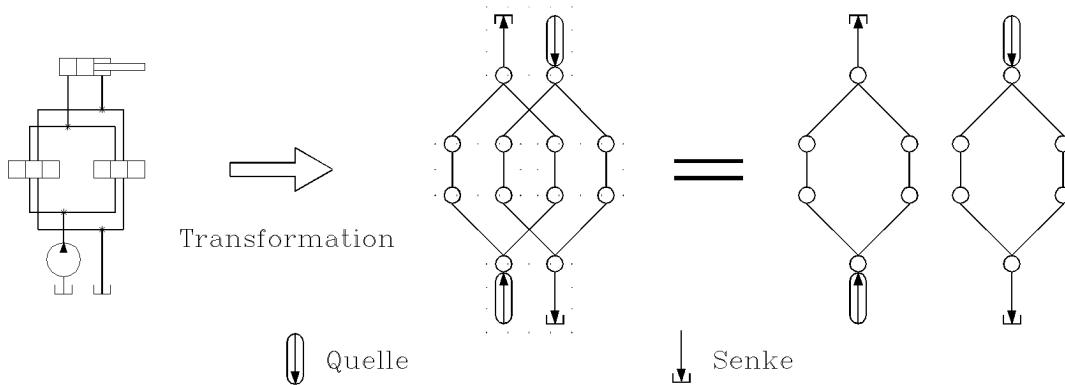


Bild A.1: Einfachste Parallelstruktur (nicht propagierungsfähig)

Solche Strukturen treten meistens als Substrukturen in ansonsten baumartigen Gesamtstrukturen auf. Diese Substrukturen können in parallele und in vernetzte Substrukturen untergliedert werden. Bild A.1 zeigt die einfachste mögliche parallele und Bild A.2 die einfachste mögliche, rein topologiebedingte vernetzte Struktur. Bild A.3 zeigt die einfachste Netzstruktur, die sich

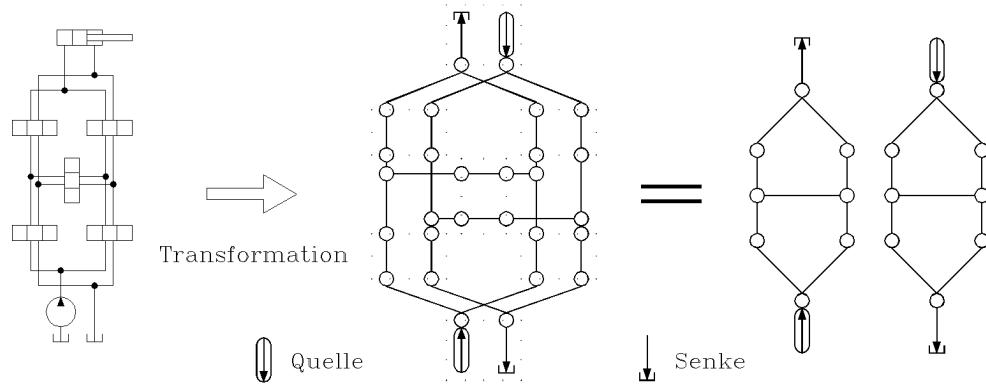


Bild A.2: Einfachste, topologiebedingte vernetzte Struktur (nicht propagierungsfähig)

von der einfachsten Parallelstruktur (Bild A.1) nicht durch die Topologie, sondern durch die lokale Beschreibung T , bzw. mindestens eine lokale Zustandsvariable $\in \mathbb{Z}_C \vee \mathbb{Z}_C(t)$ unterscheidet.

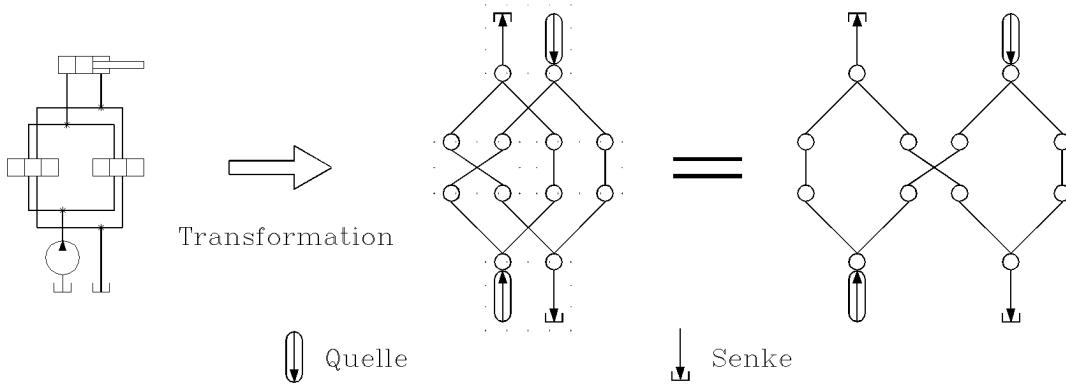


Bild A.3: Einfachste Netzstruktur (nicht propagierungsfähig)

Unterscheidung zwischen Netz- und Parallelstruktur

Eine Parallelstruktur ist gekennzeichnet durch

- eine Subquelle sowie eine Subsenke und
- jeder Knoten muß, um sich selbst erreichen zu können, sowohl die Subquelle als auch die Subsenke berühren.

Werden solche Strukturen entdeckt, so müssen diese gesondert behandelt werden (Hoffmann 1993). Ansonsten kann der baumstrukturierte Graph propagiert werden.

B Vorbelegte Richtwerte für Konstanten

Die Abschneidefrequenz ω_{cut} zur Vernachlässigung dynamisch nicht relevanter Komponenten bestimmt sich zu $\omega_{cut} = \beta_{cut} \omega_{o\ min}$.

SI_{gew}	$ \mathbb{SII} $	2	3	4	5	6	7
1	5	5	5	5	5	5	5
2	30	20	10	10	10	10	10
3	–	50	30	20	20	20	20
4	–	–	100	50	40	30	–
5	–	–	–	100	100	50	–
6	–	–	–	–	500	100	–
7	–	–	–	–	–	–	1000

Tabelle B.1: Abschneidekonstante β_{cut} (Vorbelegung)

Die Bewertungsfaktoren κ_j sind für den Einflußfaktor j von den berücksichtigten Einflußfaktoren abhängig. Tabelle B.2 zeigt diese Abhängigkeit. Es gilt: $\sum \kappa_j = 1$.

Berücksichtigung von:	\overline{MD}_{rel}	$\wedge \omega_o$	$\wedge K$	$\wedge v$	$\wedge NL$
κ_{MD}	0,5	0,5	0,5	0,4	0,4
κ_ω	0,5	0,5	0,2	0,2	0,1
κ_K	–	–	0,3	0,2	0,1
κ_v	–	–	–	0,2	0,1
κ_{NL}	–	–	–	–	0,3

Tabelle B.2: Default-Werte der Bewertungsfaktoren κ_j

Die Fehlerschranke e_{bound} ist nach Suermann (1994) von SI_{gew} und $|\mathbb{S}\mathbb{I}|$ abhängig und als Default-Werte in Tabelle B.3 abgelegt.

SI_{gew}	$ \mathbb{S}\mathbb{I} $					
	2	3	4	5	6	7
1	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
2	10^{-5}	10^{-4}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
	10^{-5}	10^{-4}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
3	—	10^{-6}	10^{-6}	10^{-7}	10^{-7}	10^{-7}
	—	10^{-6}	10^{-6}	10^{-7}	10^{-7}	10^{-7}
4	—	—	10^{-7}	10^{-8}	10^{-9}	10^{-9}
	—	—	10^{-7}	10^{-8}	10^{-9}	10^{-9}
5	—	—	—	10^{-9}	10^{-10}	10^{-11}
	—	—	—	10^{-9}	10^{-10}	10^{-11}
6	—	—	—	—	10^{-11}	10^{-12}
	—	—	—	—	10^{-11}	10^{-12}
7	—	—	—	—	—	10^{-13}
	—	—	—	—	—	10^{-13}

Tabelle B.3: Wertetabelle der Fehlerschranken e_{bound} (nach Suermann 1994)